**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОСТОВСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССТОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ**

**«РОСТОВСКИЙ-НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ»**

|  |  |
| --- | --- |
|  |  |

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

**по профессиональному модулю**

**ПМ.01 «РАЗРАБОТКА, АДМИНИСТРИРОВАНИЕ И ЗАЩИТА БАЗ ДАННЫХ»**

**по специальности 09.02.11 «Разработка и управление программного обеспечения»**

**(базовой подготовки)**

г. Ростов-на-Дону

2025 г.

|  |  |
| --- | --- |
| **РАССМОТРЕНО**  На заседании цикловой комиссии  программирования  Протокол № 1 от 01.09.2025 года  Председатель ЦК \_\_\_\_\_\_\_\_\_\_\_ А.С.Сулавко | **УТВЕРЖДАЮ**  Зам. директора по УМР  \_\_\_\_\_\_\_\_\_\_\_\_И.В.Подцатова  02.09. 2025 г. |

Комплект контрольно-оценочных средств по ПМ.01 «Разработка, администрирование изащита баз данных» разработан в соответствии с рабочей программой по ПМ, разработанной в 2025 году, по специальности 09.02.11 «Разработка и управление программного обеспечения».

Организация-разработчик: Государственное бюджетное профессиональное образовательное учреждение Ростовской области «Ростовский-на-Дону колледж связи и информатики»

Разработчик:

Малая М.А. преподаватель государственного бюджетного профессионального образовательного учреждения Ростовской области «Ростовский-на-Дону колледж связи и информатики».

# СОДЕРЖАНИЕ

1.[Общие положения ………………………………………………………………..4](#_Toc167656260)

[2. Результаты освоения модуля, подлежащие проверке …………………………5](#_Toc167656261)

[3. Оценка освоения теоретического курса профессионального модуля ПМ.11 «Разработка, администрирование и защита баз данных»………..……………..10](#_Toc167656263)

[4. Контроль приобретения практического опыта. Оценка по учебной и (или) производственной практике (по профилю специальности) …………………..136](#_Toc167656286)

[5. Контрольно-оценочные материалы для ПМ.11 «Разработка, администрирование и защита баз данных» …………………………………….138](#_Toc167656287)

# **1.Общие положения**

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности: «РАЗРАБОТКА, АДМИНИСТРИРОВАНИЕ И ЗАЩИТА БАЗ ДАННЫХ» и сформированность профессиональных и общих компетенций.

Формой аттестации по профессиональному модулю является экзамен по модулю.

**1.Формы промежуточной аттестации по профессиональному модулю**

Таблица 1

|  |  |
| --- | --- |
| **Элементы модуля, профессиональный модуль** | **Формы промежуточной аттестации** |
| **1** | **2** |
| МДК.01.01 «Проектирование и разработка баз данных» | Дифференцированный зачет |
| МДК.01.02 «Управление базами данных» | Дифференцированный зачет |
| УП.01.01 Учебная практика | Зачет |
| ПП.01.01 | Зачет |
| ПM.01.ЭК | Квалификационный экзамен |

# **2. Результаты освоения модуля, подлежащие проверке**

# **2.1. Профессиональные и общие компетенции**

В результате контроля и оценки по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Таблица 1

| **Профессиональные компетенции** | **Показатели оценки результата** |
| --- | --- |
| ПK 1.1. Проектировать базы данных. | * использование современныхcase-средств проектирования баз данных; * выполнение проектирования логической и физической схемы базы данных; * создание хранимых процедур и триггеров на базах данных; * применение стандартных методов для защиты объектов базы данных; * выполнение стандартных процедур резервного копирования и мониторинга выполнения этой процедуры; * выполнение процедур восстановления базы данных и ведение мониторинга выполнения этой процедуры; * организация информационной безопасности на уровне базы данных * *выполнение нормализации баз данных;* * *разработка схемы баз данных в различных СУБД;* * *выполнение репликации БД;* |
| ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области. |
| ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных. |
| ПK 1.4. Администрировать базы данных. |
| ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

Таблица 2

| **Общие компетенции** | **Показатели оценки результата** |
| --- | --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам | * выполнение постановки цели, выбора и применения методов и способов решения профессиональных задач; * выполнение адекватной оценки и самооценки эффективности и качества выполнения профессиональных задач |
| ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; | * выбор и использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач |
| ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях; | * демонстрация ответственности за принятые решения; * выполнение самоанализа и коррекция результатов собственной работы; |
| ОК 4 Эффективно взаимодействовать и работать в коллективе и команде; | * взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; * обоснованность анализа работы членов команды (подчиненных) |
| ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста; | * анализ эффективности использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке. |
| ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках. | * выполнение постановки цели, выбора и применения методов и способов решения профессиональных задач; * анализ адекватной оценки и самооценки эффективности и качества выполнения профессиональных задач |

**2.2. Требования к портфолио**

**Тип портфолио:** портфолио работ

**Проверяемые результаты обучения:**

**Основные требования**

Требования к структуре и оформлению портфолио:

1. В портфолио входят практические работы, входящие в МДК.01.01 «Технология разработки и защиты баз данных».
2. Практические работы оформлены в виде отчета.
3. Оформление отчета соответствует требованиям документа «Правила оформления текстовых документов в учебном процессе»

**Показатели оценки портфолио**

Таблица

|  |  |  |
| --- | --- | --- |
| **Коды и наименования проверяемых компетенций или их сочетаний** | **Показатели оценки результата** | **Оценка**  **(да / нет)** |
| ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. | - использование современных case-средств проектирования баз данных;  - выполнение проектирования логической и физической схемы базы данных;  - создание хранимых процедур и триггеров на базах данных;  - применение стандартных методов для защиты объектов базы данных;  - выполнение стандартных процедур резервного копирования и мониторинга выполнения этой процедуры;  - выполнение процедур восстановления базы данных и ведение мониторинга выполнения этой процедуры;  - организация информационной безопасности на уровне базы данных  выполнение нормализации баз данных;   * разработка схемы баз данных в различных СУБД; * выполнение репликации БД; |  |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках. | * выполнение постановки цели, выбора и применения методов и способов решения профессиональных задач; * выполнение адекватной оценки и самооценки эффективности и качества выполнения профессиональных задач * выбор и использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач * демонстрация ответственности за принятые решения; * выполнение самоанализа и коррекция результатов собственной работы; * взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; * обоснованность анализа работы членов команды (подчиненных) * анализ эффективности использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке. * выполнение постановки цели, выбора и применения методов и способов решения профессиональных задач; * анализ адекватной оценки и самооценки эффективности и качества выполнения профессиональных задач |  |

# **3.Оценка освоения теоретического курса профессионального модуля ПМ.01 «Разработка, администрирование и защита баз данных»**

**3.1. Задания для оценки освоения МДК**

**3.3.1. Задания для оценки освоения МДК.01.01 «Технология разработки и защиты баз данных**

**Тема 1.1 Основные положения теории баз данных**

**Задание 1**.

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
* оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

Вопросы к устному опросу по теме:

1. Что такое база данных?

*Ответ: База данных представляет собой организованную коллекцию данных, которая хранится и управляется в компьютерной системе.*

1. Что такое таблица в базе данных?

*Ответ: Таблица в базе данных представляет собой структурированное представление данных в виде строк и столбцов.*

3. Что такое ключевое поле или ключ в базе данных?  
  *Ответ: Ключевое поле или ключ в базе данных является уникальным идентификатором для каждой записи в таблице, позволяющим быстро получать доступ к нужным данным.*

4.Какие типы баз данных существуют?  
   *Ответ: Существуют реляционные, иерархические, сетевые, объектно-ориентированные и др.*

5. Каковы основные принципы реляционной модели данных?  
*Ответ: Основные принципы реляционной модели данных: сущности хранятся в виде таблиц, связи между ними устанавливаются с помощью ключевых полей, данные нормализуются для избегания избыточности и проблем согласованности.*

1. Что такое информационная система?  
   *Ответ: Информационная система - это система, предназначенная для сбора, хранения, обработки и передачи данных с целью управления или принятия решений.*
2. Какие основные типы информационных систем существуют?  
      *Ответ: Существуют оперативные, тактические и стратегические информационные системы, а также автоматизированные информационные системы и информационные системы поддержки принятия решений.*
3. Что такое модель информационной системы?  
   *Ответ: Модель информационной системы представляет собой абстрактное представление системы, включающее ее основные компоненты, процессы и взаимосвязи.*
4. Какие бывают структуры информационных систем?  
      *Ответ: Структуры информационных систем могут быть иерархическими, сетевыми, реляционными, объектно-ориентированными и т.д.*
5. Чем отличаются информационные системы от информационных технологий?  
   *Ответ: Информационная система - это организованная совокупность компонентов, включая информационные технологии, для сбора, хранения и обработки данных с целью обеспечения поддержки бизнес-процессов, в то время как информационные технологии представляют собой средства и методы обработки информации.*
6. Что такое структура баз данных?  
      *Ответ: Структура баз данных определяет организацию данных в базе, включая формат хранения, связи между данными и способы доступа к ним.*
7. Какие основные компоненты включает в себя структура баз данных?  
   *Ответ: Структура баз данных включает в себя таблицы (сущности), связи между таблицами, индексы для ускорения доступа к данным, а также кластеры, определяющие способ физического хранения данных.*
8. Что такое представления в базе данных?  
     *Ответ: Представление в базе данных - это виртуальная таблица, которая представляет данные из одной или нескольких таблиц базы данных в определенном формате для удобства запросов.*
9. Чем отличаются индексы от таблиц в базе данных?  
     *Ответ: Индексы в базе данных служат для ускорения поиска и сортировки данных, в то время как таблицы хранят сами данные.*
10. Что такое кластеры в базе данных?  
    *Ответ: Кластеры в базе данных определяют способ физического хранения данных на диске, оптимизируя доступ к ним и ускоряя выполнение запросов.*
11. Что такое нормализация базы данных?
12. *Ответ: Нормализация базы данных - это процесс организации данных в базе для уменьшения избыточности и обеспечения целостности и эффективности хранения.*
13. Какие нормальные формы существуют в процессе нормализации баз данных?  
       *Ответ: Существуют нормальные формы от первой (1NF) до пятой (5NF), а также дополнительные формы, такие как нормальная форма Бойса-Кодда (BCNF) и третья нормальная форма (3NF).*
14. Какие основные принципы нормализации баз данных?  
      *Ответ: Основные принципы нормализации: устранение повторения данных, разделение таблиц на логически связанные сущности, минимизация избыточной информации.*
15. Зачем нужна нормализация баз данных?  
       *Ответ: Нормализация баз данных позволяет избежать аномалий при вставке, обновлении и удалении данных, обеспечивает более эффективное использование памяти и улучшает производительность запросов к базе данных.*
16. Какие преимущества и недостатки нормализации баз данных?  
       *Ответ: Преимущества нормализации: повышение целостности данных, уменьшение избыточности информации, более эффективное хранение данных. Недостатки: усложнение запросов, возможное увеличение количества соединений таблиц.*
17. Что такое целостность данных в контексте базы данных?  
       *Ответ: Целостность данных в базе данных означает правильность и согласованность данных, их защищенность от ошибок и некорректных изменений.*
18. Какие методы обеспечения целостности данных существуют?  
       *Ответ: Существуют ограничения целостности данных, триггеры, хранимые процедуры, а также использование внешних ключей и уникальных индексов*.
19. Что такое ограничения целостности данных и какие виды ограничений существуют?  
     *Ответ: Ограничения целостности данных - это правила, которые накладываются на данные, чтобы обеспечить их корректность. Видыограничений: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK.*
20. Что представляют собой триггеры в базе данных и как они обеспечивают целостность данных?

*Ответ: Триггеры - это хранимые процедуры, которые автоматически выполняются при определенных событиях (например, при вставке, обновлении или удалении данных), что позволяет контролировать и обеспечивать целостность данных*.

1. Каким образом использование внешних ключей способствует обеспечению целостности данных?  
      *Ответ: Внешний ключ - это поле или комбинация полей, которые связывают одну таблицу с другой. Использование внешних ключей позволяет поддерживать связи между таблицами и обеспечивать целостность данных при выполнении операций в базе.*

**Тема 1.2. Проектирование баз данных**

**Задание 1**.

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
* оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

Вопросы к устному опросу по теме:

1. Что такое концептуальная модель данных?

*Ответ:* Концептуальная модель данных является высокоуровневым описанием структуры данных и их взаимосвязей без учета спецификаций баз данных.  
2. Какие принципы лежат в основе построения концептуальной модели данных?

*Ответ:* Принципы построения концептуальной модели данных включают абстракцию, модульность, иерархичность и независимость от конкретной реализации.  
3. Что такое логическая модель данных?

*Ответ:* Логическая модель данных описывает структуру данных и их связи в контексте конкретной системы управления базами данных.  
4. Какие принципы следует соблюдать при построении логической модели данных?

*Ответ:* При построении логической модели данных необходимо учитывать нормализацию, целостность и согласованность данных

5. Что такое физическая модель данных?

*Ответ:* Физическая модель данных описывает способы физического хранения и доступа к данным в базе данных, такие как таблицы, индексы, ключи и т.д.  
6. На каких принципах базируется построение физической модели данных?

*Ответ:* При построении физической модели данных необходимо учитывать производительность, эффективность доступа к данным, резервирование и масштабируемость базы данных.

7*.*Что такое инструментальные средства проектирования схемы базы данных?

*Ответ:* Инструментальные средства проектирования схемы базы данных представляют собой программные средства, которые облегчают разработку и моделирование баз данных.  
8. Какие основные функции предоставляют современные инструментальные средства проектирования схемы базы данных?

*Ответ:* Современные инструментальные средства проектирования схемы базы данных предоставляют функциональности для создания ER-диаграмм, генерации SQL кода, проверки нормализации, визуализации структуры данных и т.д.

9. Какие популярные инструменты используются для проектирования схемы базы. данных?  
*Ответ:* Некоторые популярные инструменты для проектирования схемы базы данных включают MySQL Workbench, Microsoft Visio, ERwin Data Modeler, Oracle SQL Developer и др.

10. Какие преимущества предоставляют инструментальные средства проектирования схемы базы данных?

*Ответ:* Инструментальные средства проектирования схемы базы данных позволяют ускорить процесс разработки, облегчить совместную работу над проектом, автоматизировать генерацию кода и обеспечить более высокую надежность создаваемых баз данных.

1. Какие характеристики следует учитывать при выборе инструментальных средств проектирования схемы базы данных?  
   *Ответ:* При выборе инструментальных средств проектирования схемы базы данных следует учитывать поддерживаемые типы баз данных, функциональные возможности, удобство использования, стоимость лицензирования и доступность необходимых дополнительных функций.
2. Что представляет собой концептуальная модель данных?  
   *Ответ:* Концептуальная модель данных представляет собой высокоуровневое описание структуры данных и их взаимосвязей, не зависящее от конкретной технологии хранения данных.
3. Какие принципы лежат в основе построения концептуальной модели данных?  
   *Ответ:*Принципы построения концептуальной модели данных включают абстракцию, модульность, иерархичность и независимость от физической реализации.
4. Что включает в себя логическая модель данных?  
   *Ответ:*Логическая модель данных включает в себя описание структуры данных с учетом конкретной технологии хранения, нормализацию таблиц, ключи, отношения и другие аспекты разработки базы данных.
5. Каковы основные принципы построения логической модели данных?  
   *Ответ:* Основные принципы логической модели данных включают нормализацию, соблюдение целостности данных, оптимизацию запросов и эффективность доступа к данным.
6. Что описывает физическая модель данных?  
   *Ответ:*Физическая модель данных описывает способы хранения данных в реальной базе данных, например, таблицы, индексы, хранимые процедуры, оптимизацию для конкретной СУБД.
7. На какие аспекты следует обратить внимание при построении физической модели данных?  
   *Ответ:* При построении физической модели данных следует обращать внимание на производительность базы данных, способы оптимизации запросов, выбор типов данных и настройки индексов для эффективного доступа к данным.

**Задание 2**. Практические занятия

Практическое занятие №1 «Сбор и анализ информации».

Цель работы: сформировать у студентов практические знания по сбору и анализу информации по приведенной предметной области.

Проверяемые результаты обучения:

* основные положения теории баз данных, хранилищ данных, баз знаний;
* основные принципы структуризации и нормализации базы данных;
* методы организации целостности данных;
* нормальные формы и правила Кодда;
* нормализация базы данных;

Формирование компетенций (указываются коды ОК и ПК):

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоапться профессиональной документацией на государственном и иностранном языках. Пользоваться профессиональной документацией на государственном и иностранном языках. (п. 3.2 в ред. Приказа Минпросвещения России от 01.09.2022 N 796)  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

Задание:

Изучить предметную область «Грузоперевозки». Определить информацию, которую необходимо хранить в БД.

**БД: Грузоперевозки**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Перевозка** | | Дата отправления  Адрес «откуда»  Адрес «куда»  Дата прибытия  Дата формирования заявки  Название груза  Вес груза  Характеристика груза  Общая стоимость поездки | | |  | | --- | | **Перевозчик** | | Название компании перевозчика  Адрес компании перевозчика  Телефон компании перевозчика  Платёжные реквизиты компании перевозчика |  |  | | --- | | **Автомобиль** | | Марка автомобиля  Модель автомобиля  ГОС номер автомобиля | |
| |  | | --- | | **Заказчик** | | Название заказчика  ФИО контактного лица заказчика  Номер телефона заказчика | | |  | | --- | | **Водитель** | | ФИО водителя | |
| |  | | --- | | **Категорияперевозки** | | Категория перевозки (по городу, по городу в выходные дни, по области, по России и т.д.)  Стоимость поездки в зависимости от категории  Единица измерения перевозки (км., часы – зависит от категории перевозки) | | |

Критерии оценки

|  |  |
| --- | --- |
| **Этап задания** | **Балл** |
| Создание схемы базы данных со всеми сущностями | 2 – все сущности  0 – не все сущности |
| Проведена нормализация и устранены аномалии | 1 – нормализация проведена, аномалий нет  0,5 – есть незначительные ошибки (не более двух)  0 – нормализация не проведена или ошибок более двух |
| Определение «ключей» в сущностях | 1 – ключи установлены правильно и позволяют обеспечить целостность  0,5 – ключи определены не везде, но не более двух ошибок (это не относиться к «первичным ключам» – сущности должны соответствовать первой нормальной форме)  0 – частично отсутствуют первичные ключи или более двух ошибок во «внешних ключах». |
| Реализованы связи между сущностями | 1 – связи, для реализации целостности данных, установлены все.  0,5 – отсутствует одна связь  0 – отсутствует более одной связи |
| Максимальное кол-во баллов | 5 |

Таблица – Критерии оценки

|  |  |
| --- | --- |
| **Оценка** | **Количество баллов** |
| Отлично – « 5 » | 5 баллов |
| Хорошо – « 4 » | 4 балла |
| Удовлетворительно – « 3 » | 2 – 3 баллов |
| Неудовлетворительно – « 2 » | 1 и менее 1 баллов |

Практическое занятие №2 «Проектирование реляционной схемы базы данных в CASE-средстве»

Цель работы: сформировать у студентов практические знанияпо проектированию баз данных

Проверяемые результаты обучения:

* нормализация базы данных;
* работа с современными case-средствами проектирования баз данных;

проектирование логической и физической схемы базы данных

* знание основных положений теории баз данных, хранилищ данных, баз знаний;
* знание основных принципов структуризации и нормализации базы данных;
* знание основных принципов построения концептуальной, логической и физической модели данных;
* знание методов организации целостности данных;
* знание нормальных форм и правил Кодда;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках. (п. 3.2 в ред. Приказа Минпросвещения России от 01.09.2022 N 796)  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

1. Выполнить проектирование концептуальной, логической моделей базы данных.

*Описание предметной области*

Вашей задачей является разработка информационной системы для Отдела технического контроля ООО «СПбЦ «ЭЛМА».

Санкт-Петербургский Центр «ЭЛМА» - российская научно-производственная компания. С 1991 г. компанией разрабатываются, изготавливаются и внедряются технологические процессы, химические растворы и оборудование для производства печатных плат.

Разработанные «СПбЦ «ЭЛМА» технологические процессы введены в ОСТ 107.460092.028-96 «Платы печатные. Технические требования к технологии изготовления».

Независимо от выпускаемой продукции, на производстве обязательно присутствуют работники, которые постоянно контролирует технический процесс и его соответствие ГОСТу. Это – специалисты Отдела Технического контроля. ОТК следит за всеми этапами производства, начиная от поставки сырья и заканчивая складированием готовой продукции.

ОТК отвечает за некачественную продукцию, за ее возврат на производство.

ОТК «СПбЦ «ЭЛМА» работает на прием материала с собственного производства, проводит специализированные исследования с помощью новейшего оборудования, ведет базу данных исследований, предоставляет отчеты по проведенным исследованиям и контролю качества выпускаемой продукции.  Кроме того, ОТК проводит контрольные проверки на контроль качества большинства предприятий химической и нефтехимической промышленности Санкт-Петербурга, таких как ПК «Полимилл», ООО «Промышленные технологии», ЗАО «ТехПластТехнологии», ОАО «Эластоком» и другие.

В силу того, что оборудование для лабораторных исследований в ОТК ООО «СПбЦ «ЭЛМА» является уникальным, отдел предоставляет услуги независимой технической экспертизы состава полимерных материалов частным лицам.

Вам необходимо разработать информационную систему для ОТК, с которой будут работать сотрудники отдела, а также клиенты – сторонние предприятия для контрольных проверок или частные лица.

**Ход работы:**

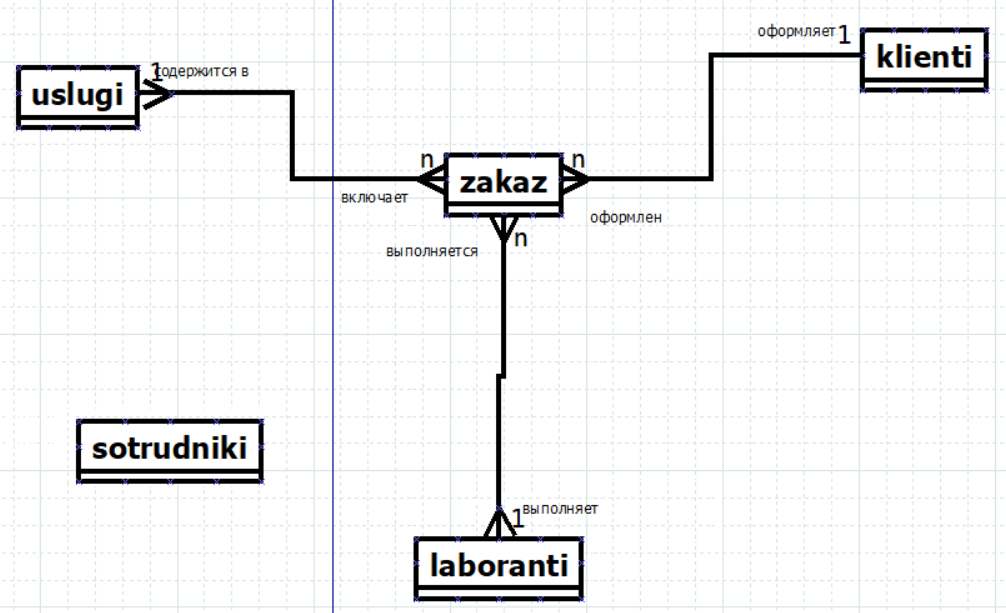
* + - 1. Изучите описание предметной области. На основе описания предметной области и задания экзамена Вам необходимо спроектировать ER-диаграммы для информационной системы. Обязательна 3 нормальная форма с обеспечением ссылочной целостности.

1. Обеспечьте хранение в базе данных:

* услуги ОТК (наименование, стоимость, код услуги, срок выполнения, среднее отклонение);
* данные клиентов (ЮЛ – название компании, адрес, ИНН, р/с, БИК, ФИО руководителя, ФИО контактного лица, телефон контактного лица, e-mail, пароль; для ФЛ – e-mail, пароль, ФИО, дата рождения, серия и номер паспорта, телефон);
* заказ (дата создания, услуги, которые входят в заказ, статус заказа, статус услуги в заказе, время выполнения заказа (в часах), стоимость заказа);
* оказанная услуга (услуга, когда и кем были проведены исследования);
* данные лаборантов и контролеров ОТК (логин, пароль, ФИО, последняя дата и время входа, набор услуг, которые он может оказывать, проведенные исследования);
* данные других сотрудников ОТК (логин, пароль, ФИО, последняя дата и время входа).

1. Выполните проектирование концептуальной модели

* Выделите необходимые сущности.
* Определите типы связей между сущностями.
* Составьте матрицу связей.
* Используя программу Dia, выполните создание концептуальной модели БД.

**Результат работы приведен ниже:**

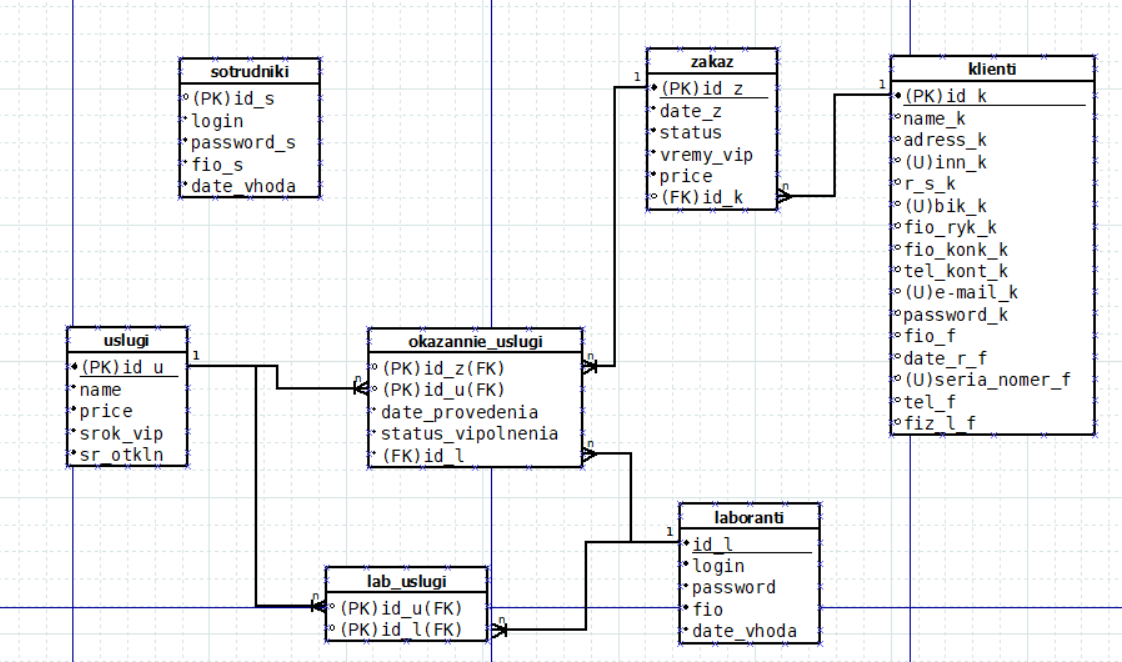
1. Выполните проектирование логической модели БД

* Выделите необходимые атрибуты в сущностях.
* Определите первичные и внешние кличи.
* Проведите нормализацию БД (решите связь многие ко многим, если это необходимо)
* Постройте матрицусвязей.
* Определите типы связей.
* Определите ограничения уникальности и поля обязательные к заполнению.
* Удалите все расчетные поля в сущностях, а также повторяющиеся поля.
* Заполните таблицы сопоставления.
* Используя программу Dia, выполните создание логической модели БД.

Пример заполнения документа сопоставления таблиц.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя таблицы** | | **Краткое имя таблицы** | | | |
| uslugi | |  | |  |  |
| **Key Type** | | **Optionality** |  |  | **Column Name** |
| pk | | \* |  |  | id\_u |
| uk | | \* |  |  | name |
|  |  | \* |  |  | price |
|  |  | \* |  |  | srok\_vip |
|  |  | \* |  |  | sr\_otkl |

Результат построения логической модели приведен ниже:



**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №3. Проектирование физической модели данных. Разработка словаря данных.

Цель работы: сформировать у студентов практические знания по проектированию физической модели и разработке словаря данных.

Проверяемые результаты обучения:

* основные положения теории баз данных, хранилищ данных, баз знаний;
* основные принципы структуризации и нормализации базы данных;
* методы организации целостности данных;
* нормальные формы и правила Кодда;
* нормализация базы данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

1. Заполнить таблицы сопоставления.
2. Выполнить проектирование физической модели данных.
3. Разработать словарь данных.

**Ход работы**

1. Заполнение таблиц сопоставления.

В качестве примера для вас заполнена таблица okazannie\_uslugi.

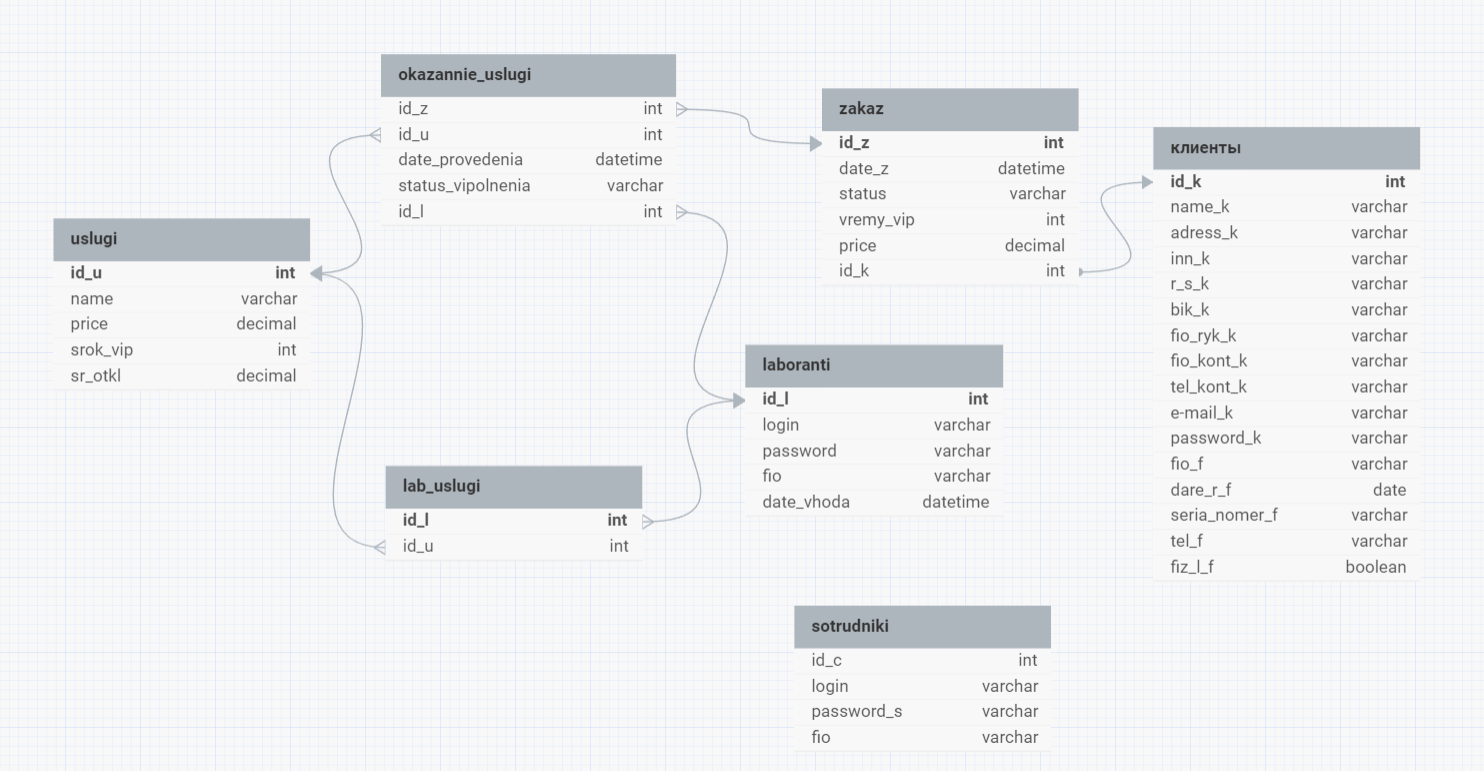
**Решение:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя таблицы** | **Краткое имя таблицы** | |  |  |
| okazannie\_uslugi |  |  |  |  |
| **Key Type** | **Optionality** | **Column Name** | **Data type** | **size** |
|  |  |  |  |  |
| pk, fk0 | \* | id\_z | int |  |
| pk, fk1 | \* | id\_u | int |  |
|  | \* | date\_provedenia | date |  |
|  | \* | status\_vipolnenia | varchar | 15 |
| fk2 | \* | id\_l | int |  |

1. Проектирование физической модели данных.

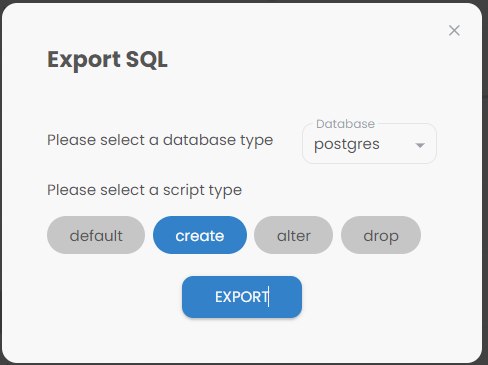
Запустите CASE-систему DBDESIGNER<https://erd.dbdesigner.net>

Выполните создание физической схемы.

Результат приведен ниже:

1. Экспортируйте файл скрипта

Export ⭢Export SQL



1. Выполните создание словаря данных.

Пример заполнения словаря данных приведен ниже:

**Словарь данных**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Имя таблицы** | **Краткое имя таблицы** | |  | | | |
| услуги/ uslugi | ctr |  |  | | | |
| Key Type  (Тип ключа) | Optionality  (Обязательность/ необязательность) | ColumnName (Название столбца) | Datatype  (Тип данных) | Size  (Размер поля) | Constraints (Ограничения) | Comment (Комментарий) |
|  |  |  |  |
| pk | \* | id\_u | serial |  | "uslugi\_pkey" PRIMARY KEY, btree (id\_u) | Номер услуги |
|  | \* | name | varchar | 50 |  | Название услуги |
|  | \* | price | numeric | 10,2 |  | Цена услуги |
|  | \* | srok\_vip | int |  |  | Срок выполнения услуги |
|  | \* | sr\_otkl | numeric | 10,2 |  | Срок отклонения |

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

**Тема 1.3 Разработка и администрирование баз данных**

**Задание 1**.

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
* оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

Вопросы к устному опросу по теме:

1. Что такое база данных в PostgreSQL?

*Ответ*: База данных в PostgreSQL- это набор взаимосвязанных данных, организованных в соответствии с определенной схемой. Каждая база данных в PostgreSQL является изолированным логическим пространством данных.

2. Что такое схема в PostgreSQL?

*Ответ*: Схема в PostgreSQL - это логическая структура для организации объектов базы данных, таких как таблицы, представления, функции, последовательности и др. Схема позволяет группировать и организовывать эти объекты.

3. Какая схема создается по умолчанию в PostgreSQL?

*Ответ*: По умолчанию в PostgreSQL создается схема public. Все новые объекты базы данных, созданные без указания схемы, помещаются в эту схему.

4. Как создать новую схему в PostgreSQL?

*Ответ*: Чтобы создать новую схему в PostgreSQL, используется оператор CREATE SCHEMA. Например: CREATE SCHEMA новая\_схема;

5. Как удалить схему в PostgreSQL?

*Ответ*: Для удаления схемы в PostgreSQL используется оператор DROP SCHEMA. Например: DROP SCHEMA новая\_схема;

6. Как указать схему при создании объекта базы данных?

*Ответ*: При создании объекта базы данных (таблицы, представления, функции и т.д.) можно указать схему, в которой этот объект должен быть создан, используя следующий синтаксис: схема. объект. Например: CREATE TABLE новая\_схема. таблица (...);

7. Как получить список всех схем в базе данных PostgreSQL?

*Ответ*: Чтобы получить список всех схем в базе данных PostgreSQL, можно выполнить следующий SQL-запрос: SELECT schema\_name FROM information\_schema. schemata;

1. Что такое системный каталог в PostgreSQL?

*Ответ*: Системный каталог в PostgreSQL- это набор системных таблиц, содержащих метаданные (описательную информацию) о базе данных, ее объектах и параметрах.

1. Где находится системный каталог в PostgreSQL?

*Ответ*: Системный каталог в PostgreSQL находится в схеме information\_schema и в некоторых системных схемах, таких как pg\_catalog.

1. Зачем нужен системный каталог?

*Ответ*: Системный каталог необходим для хранения информации о структуре базы данных, объектах (таблицах, представлениях, функциях и т.д.), правах доступа, типах данных и многом другом. Эта информация используется самой СУБД для управления базой данных.

1. Какие системные таблицы входят в системный каталог PostgreSQL?

*Ответ*: В системный каталог PostgreSQL входят такие таблицы, как pg\_class, pg\_attribute, pg\_type, pg\_database, pg\_user, pg\_namespace и многие другие.

1. Можно ли изменять данные в системном каталоге напрямую?

*Ответ*: Напрямую изменять данные в системном каталоге не рекомендуется, так как это может привести к повреждению базы данных и нарушению ее целостности. Для изменения метаданных следует использовать соответствующие SQL-операторы.

1. Как получить информацию из системного каталога?

*Ответ*: Для получения информации из системного каталога можно использовать SQL-запросы к соответствующим системным таблицам. Например, для получения списка всех таблиц в базе данных можно выполнить запрос SELECT \* FROM pg\_catalog.pg\_tables;.

1. Какие преимущества дает знание системного каталога?

*Ответ*: Знание системного каталога позволяет лучше понимать внутреннее устройство PostgreSQL, получать различную метаинформацию о базе данных, объектах и параметрах, а также выполнять более сложные и специфические операции с базой данных.

1. Что такое табличные пространства в PostgreSQL?

*Ответ*: ВPostgreSQL табличные пространства (tablespaces) представляют собой логические контейнеры для хранения данных, которые позволяют разделять и организовывать физическое распределение данных по различным директориям или файловым системам.

1. Как создать табличное пространство в PostgreSQL?

*Ответ*: Для создания табличного пространства в PostgreSQL используется оператор CREATE TABLESPACE. Например: CREATE TABLESPACE имя\_пространства LOCATION '/путь/к/директории';

1. Зачем использовать табличные пространства в PostgreSQL?

*Ответ*: Использование табличных пространств позволяет разгружать дисковую подсистему, управлять распределением данных для оптимизации производительности, а также улучшать организацию хранения данных и бэкапов.

1. Как просмотреть список всех табличных пространств в PostgreSQL?

*Ответ*: Для просмотра списка всех табличных пространств в PostgreSQL можно выполнить запрос к системной таблице pg\_tablespace. Например: SELECT \* FROM pg\_tablespace;

1. Как присоединить таблицу к определенному табличному пространству?

*Ответ*: Чтобы присоединить таблицу к определенному табличному пространству в PostgreSQL, при создании таблицы следует указать параметр TABLESPACE. Например: CREATE TABLE имя\_таблицы (...) TABLESPACE имя\_пространства;

1. Можно ли изменить местоположение табличного пространства в PostgreSQL?

*Ответ*: Да, в PostgreSQL можно изменить местоположение табличного пространства с помощью оператора ALTER TABLESPACE. Например: ALTER TABLESPACE имя\_пространства LOCATION '/новый/путь/к/директории';

1. Как удалить табличное пространство в PostgreSQL?

*Ответ*: Для удаления табличного пространства в PostgreSQL используется оператор DROP TABLESPACE. Например: DROP TABLESPACE имя\_пространства;

1. Как создать новую таблицу в PostgreSQL?

*Ответ*: Для создания новой таблицы в PostgreSQL используется оператор CREATE TABLE. Например: CREATE TABLE имя\_таблицы (столбец1 тип\_данных, столбец2 тип\_данных, ...);

1. Как добавить новый столбец в существующую таблицу?

*Ответ*: Для добавления нового столбца в существующую таблицу в PostgreSQL используется оператор ALTER TABLE ADD COLUMN. Например: ALTER TABLE имя\_таблицы ADD COLUMN новый\_столбецтип\_данных;

1. Как удалить столбец из таблицы в PostgreSQL?

*Ответ*: Для удаления столбца из таблицы в PostgreSQL используется оператор ALTER TABLE DROP COLUMN. Например: ALTER TABLE имя\_таблицы DROP COLUMN имя\_столбца;

1. Как переименовать таблицу в PostgreSQL?

*Ответ*: Для переименования таблицы в PostgreSQL используется оператор ALTER TABLE RENAME TO. Например: ALTER TABLE старое\_имя\_таблицы RENAME TO новое\_имя\_таблицы;

1. Как удалить таблицу в PostgreSQL?  
   *Ответ*: Для удаления таблицы в PostgreSQL используется оператор DROP TABLE. Например: DROP TABLE имя\_таблицы;
2. Как добавить первичный ключ к таблице в PostgreSQL?

*Ответ*: Для добавления первичного ключа к таблице в PostgreSQL используется оператор ALTER TABLE ADD PRIMARY KEY. Например: ALTER TABLE имя\_таблицы ADD PRIMARY KEY (столбец1, столбец2);

1. Как добавить внешний ключ к таблице в PostgreSQL

*Ответ*: Для добавления внешнего ключа к таблице в PostgreSQL используется оператор ALTER TABLE ADD FOREIGN KEY. Например: ALTER TABLE имя\_таблицы ADD FOREIGN KEY (столбец1) REFERENCES другая\_таблица(столбец1);

1. Как просмотреть структуру таблицы в PostgreSQL?

*Ответ*: Для просмотра структуры таблицы в PostgreSQL можно использовать операторы \d имя\_таблицы в интерактивном режиме, либо запрос SELECT \* FROM information\_schema. columns WHERE table\_name = 'имя\_таблицы';

1. Что такое индексы в PostgreSQL и для чего они используются?

*Ответ*: Индексы в PostgreSQL - это структуры данных, которые ускоряют поиск и доступ к данным в таблицах. Они используются для оптимизации производительности запросов, особенно при выполнении операций SELECT.

1. Как создать индекс в PostgreSQL?

*Ответ*: Для создания индекса в PostgreSQL используется оператор CREATE INDEX. Например: CREATE INDEX имя\_индекса ON имя\_таблицы (столбец);

1. Как удалить индекс в PostgreSQL?

*Ответ*: Для удаления индекса в PostgreSQL используется оператор DROP INDEX. Например: DROP INDEX имя\_индекса;

1. Как проверить наличие индексов на таблице в PostgreSQL?

*Ответ*: Для проверки наличия индексов на таблице в PostgreSQL можно выполнить запрос к системной таблице pg\_indexes. Например: SELECT \* FROM pg\_indexes WHERE tablename = 'имя\_таблицы';

1. Как изменить данные в таблице PostgreSQL?

*Ответ*: Для изменения данных в таблице PostgreSQL используется оператор UPDATE. Например: UPDATE имя\_таблицы SET столбец = новое\_значение WHERE условие;

1. Как удалить данные из таблицы в PostgreSQL?

*Ответ*: Для удаления данных из таблицы PostgreSQL используется оператор DELETE. Например: DELETE FROM имя\_таблицы WHERE условие;

1. Как добавить данные в таблицу PostgreSQL?

*Ответ*: Для добавления данных в таблицу PostgreSQL используется оператор INSERT INTO. Например: INSERT INTO имя\_таблицы (столбец1, столбец2) VALUES (значение1, значение2);

1. Как пересоздать индекс в PostgreSQL?

*Ответ*: Для пересоздания индекса в PostgreSQL необходимо удалить существующий индекс и создать новый. Например:

SQL

DROP INDEX имя\_индекса;

CREATE INDEX имя\_индекса ON имя\_таблицы (столбец);

1. Что такое процедура и для чего она используется?  
   *Ответ*: Процедура - это набор инструкций, который выполняется в определенном порядке для выполнения определенной задачи.
2. Какие преимущества имеют процедуры в программировании?  
   *Ответ*: Процедуры позволяют разделить программу на более мелкие подзадачи, повторно использовать код, улучшить понимание программы, облегчить отладку и управление программой.
3. Что такое событие и какие типы событий существуют?  
   *Ответ*: Событие - это произошедшее изменение в программе, на которое программа должна реагировать. Существует множество типов событий, таких как клики мыши, нажатия клавиш, загрузка страницы и др.
4. Какие методы можно использовать для обработки событий?  
   *Ответ*: Для обработки событий можно использовать методы, такие как обработчики событий, слушатели событий и методы, привязанные к событиям.
5. Как можно связать процедуры с событиями в программировании?  
   *Ответ*: В программировании процедуры могут быть связаны с событиями с помощью обработчиков событий. Когда происходит определенное событие, вызывается соответствующий обработчик, который выполняет определенную процедуру.
6. Что такое триггеры в PostgreSQL и для чего они используются?  
   *Ответ*: Триггеры в PostgreSQL - это хранимые процедуры, которые автоматически срабатывают при выполнении определенного события на таблице. Они используются для реализации бизнес-логики, обеспечения целостности данных, аудита и других задач.
7. Какие типы триггеров поддерживаются в PostgreSQL?  
   *Ответ*: ВPostgreSQL поддерживаются триггеры ДО (BEFORE) и ПОСЛЕ (AFTER) на операции INSERT, UPDATE и DELETE, а также триггеры на уровне таблицы или столбца.
8. Как создать триггер в PostgreSQL?  
   *Ответ*: Для создания триггера в PostgreSQL используется команда CREATE TRIGGER, в которой указывается имя триггера, событие, на которое он должен реагировать, таблица, на которую он применяется, и тело триггера - хранимая процедура на PL/pgSQL.
9. Какие могут быть применения триггеров в PostgreSQL?  
   *Ответ*: Применения триггеров в PostgreSQL могут быть разнообразными: от автоматического обновления связанных данных и проверки целостности данных до аудита изменений, сбора статистики или реализации бизнес-логики.
10. Как удалить триггер в PostgreSQL?  
    *Ответ*: Для удаления триггера в PostgreSQL используется команда DROP TRIGGER, в которой указывается имя триггера и таблица, на которую он применяется. Также можно использовать команду ALTER TABLE для удаления триггера, если известно его имя.
11. Что такое транзакция в PostgreSQL?  
       *Ответ*: Транзакция в PostgreSQL представляет собой последовательность операций базы данных, которые выполняются как единое целое.
12. Как начать новую транзакцию в PostgreSQL?  
       *Ответ*: Для начала новой транзакции в PostgreSQL используется команда BEGIN или START TRANSACTION.
13. Что такое COMMIT в PostgreSQL и какова его роль в транзакции?  
       *Ответ*: COMMIT в PostgreSQL используется для завершения транзакции и сохранения всех изменений, внесенных в базу данных в рамках данной транзакции.
14. Что происходит, если внутри транзакции произойдет ошибка?  
       *Ответ*: В случае возникновения ошибки внутри транзакции в PostgreSQL, все изменения, внесенные до момента возникновения ошибки, будут отменены (ROLLBACK).
15. Как откатить (отменить) транзакцию в PostgreSQL?  
       *Ответ*: Для отката (отмены) транзакции в PostgreSQL используется команда ROLLBACK.
16. Какие существуют уровни изоляции транзакций в PostgreSQL?  
       *Ответ*: В PostgreSQL поддерживаются четыре уровня изоляции транзакций: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ и SERIALIZABLE.
17. Можно ли вложить одну транзакцию в другую в PostgreSQL?  
       *Ответ*: Да, в PostgreSQL можно вложить одну транзакцию в другую при помощи сохранения точек сохранения (SAVEPOINT).
18. Как проверить статус транзакции в PostgreSQL?  
       *Ответ*: Для проверки статуса текущей транзакции в PostgreSQL можно использовать функцию currenttransactioninfo().
19. Что такое блокировка в PostgreSQL?  
       *Ответ*: Блокировка в PostgreSQL - это механизм, который предотвращает одновременное изменение данных несколькими параллельными транзакциями.
20. Какие типы блокировок поддерживает PostgreSQL?  
       *Ответ*: PostgreSQL поддерживает различные типы блокировок, такие как блокировки на чтение (sharedlocks), блокировки на запись (exclusivelocks), а также другие уровни блокировки.
21. В чем разница между блокировкой на чтение и блокировкой на запись в PostgreSQL?

*Ответ*: Блокировка на чтение (sharedlock) позволяет нескольким транзакциям читать данные одновременно, в то время как блокировка на запись (exclusivelock) предотвращает другие транзакции от одновременного чтения и записи данных.

1. Что такое deadlock и как его можно предотвратить в PostgreSQL?  
      *Ответ*: Deadlock - это ситуация, когда две или более транзакции взаимно блокируют друг друга и не могут завершиться. В PostgreSQLdeadlock может быть предотвращен путем правильного использования блокировок и управления транзакциями.
2. Как можно определить текущие блокировки в PostgreSQL?  
      *Ответ*: Для определения текущих блокировок в PostgreSQL можно использовать системный каталог pg\_locks или функцию pg\_blocking\_pids().
3. Как можно разрешить блокировку в PostgreSQL?  
     *Ответ*: Блокировку можно разрешить путем завершения транзакции, которая удерживает блокировку, либо путем изменения уровня изоляции транзакции или запроса.
4. Какие меры можно предпринять для снижения конфликтов блокировок в PostgreSQL?  
      *Ответ*: Для снижения конфликтов блокировок в PostgreSQL можно использовать короткие и одновременные транзакции, правильно выбирать типы блокировок, делать запросы экономными, а также управлять транзакциями и блокировками с умом.
5. Что такое транзакция в PostgreSQL?  
   *Ответ*: Транзакция в PostgreSQL — это набор операций, который выполняется как единое целое. В случае успешного выполнения всех операций в транзакции, изменения сохраняются в базе данных. В случае ошибки, все изменения отменяются.
6. Как начать транзакцию в PostgreSQL?  
   *Ответ*: Для начала транзакции в PostgreSQL используется команда "BEGIN;".
7. Как завершить транзакцию в PostgreSQL?  
   *Ответ*: Для завершения транзакции в PostgreSQL можно использовать команды "COMMIT;", чтобы сохранить изменения, или "ROLLBACK;", чтобы отменить изменения и вернуть базу данных к предыдущему состоянию.
8. Что такое сохранная точка (savepoint) в PostgreSQL?  
   *Ответ*: Сохранная точка в PostgreSQL позволяет установить метку внутри транзакции, чтобы можно было отменить изменения, произошедшие после этой точки, с помощью команды "ROLLBACK TO SAVEPOINT".
9. Можно ли выполнять параллельные транзакции в PostgreSQL?  
   *Ответ*: Да, в PostgreSQL можно выполнять параллельные транзакции. Система управления транзакциями обеспечивает изоляцию между транзакциями для предотвращения конфликтов и соблюдения целостности данных.
10. Что такое блокировка (lock) в PostgreSQL?  
    *Ответ*: Блокировка в PostgreSQL — это механизм, который предотвращает одновременное изменение одних и тех же данных из разных транзакций для обеспечения целостности данных.
11. Какие типы блокировок поддерживает PostgreSQL?  
    *Ответ*: PostgreSQL поддерживает различные типы блокировок, такие как SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE, ACCESS SHARE, ROW EXCLUSIVE и другие, каждый из которых предназначен для определенных целей и контроля доступа к данным.
12. Что такое блокировка чтения и блокировка записи в PostgreSQL?  
    *Ответ*: Блокировка чтения позволяет одному процессу читать данные без блокировки других запросов на чтение, но блокирует запросы на запись. Блокировка записи предотвращает одновременное изменение данных из разных транзакций.
13. Какие проблемы могут возникнуть из-за неправильного использования блокировок в PostgreSQL?  
    *Ответ*: Неправильное использование блокировок в PostgreSQL может привести к блокировкам-тупикам (deadlocks), когда две или более транзакции блокируют друг друга и не могут завершиться, или к долгим ожиданиям (longwaits) из-за блокировок других транзакций.
14. Как можно избежать проблем с блокировками в PostgreSQL?  
    *Ответ*: Чтобы избежать проблем с блокировками в PostgreSQL, следует правильно выбирать типы блокировок, минимизировать время удержания блокировок, использовать индексы для ускорения поиска данных и тщательно планировать работу с блокировками в своем приложении.
15. Что такое импорт данных в PostgreSQL?  
    *Ответ*: Импорт данных в PostgreSQL - процесс загрузки данных из внешних источников в базу данных PostgreSQL.
16. Какой командой можно выполнить импорт данных в PostgreSQL?  
    *Ответ*: Для импорта данных в PostgreSQL можно использовать команду COPY или утилиту pgrestore.
17. Какие форматы файлов поддерживает PostgreSQL для импорта данных? *Ответ*: PostgreSQL поддерживает импорт данных из файлов в форматах CSV, TSV, SQL, и других.
18. Какой командой можно экспортировать данные из PostgreSQL? *Ответ*: Для экспорта данных из PostgreSQL можно использовать команду COPY или утилиту pgdump.
19. Какие параметры можно указать при импорте данных в PostgreSQL?  
    *Ответ*: При импорте данных в PostgreSQL можно указать параметры такие как разделитель полей, кодировка файлов, формат данных и многие другие.
20. Можно ли импортировать данные из других типов баз данных в PostgreSQL?  
    *Ответ*: Да, можно импортировать данные из других типов баз данных в PostgreSQL, используя промежуточные форматы данных или специальные утилиты.
21. Каким образом можно автоматизировать процесс импорта и экспорта данных в PostgreSQL?  
    *Ответ*: Для автоматизации процесса импорта и экспорта данных в PostgreSQL можно написать скрипты на языке программирования, использовать планировщик задач или специализированные инструменты управления данными.
22. Что такое мониторинг сервера PostgreSQL?  
    *Ответ*: Мониторинг сервера PostgreSQL- это процесс отслеживания и анализа различных метрик и параметров работы PostgreSQL для обеспечения его корректной и эффективной работы.
23. Какие основные метрики следует отслеживать при мониторинге PostgreSQL?

*Ответ*: Косновным метрикам, которые необходимо отслеживать, относятся: использование ЦП, использование памяти, использование дискового пространства, количество активных соединений, время выполнения запросов, уровень репликации и т.д.

1. Какие инструменты можно использовать для мониторинга PostgreSQL?  
   *Ответ*: Для мониторинга PostgreSQL можно использовать встроенные средства, такие как pg\_stat\_statements, pg\_stat\_activity, а также сторонние инструменты, например, pgBadger, pgMetrics, Zabbix, Prometheus и др.
2. Что такое pg\_stat\_statements и как его использовать для мониторинга?  
   *Ответ*: pg\_stat\_statements- это встроенный модуль PostgreSQL, который собирает статистику о выполнении SQL-запросов. Его можно использовать для оптимизации медленных запросов и выявления проблемных участков в приложении.
3. Как можно настроить оповещения о критических ситуациях в PostgreSQL?  
   *Ответ*: Для настройки оповещений о критических ситуациях в PostgreSQL можно использовать системы мониторинга, такие как Zabbix или Prometheus, которые позволяют настраивать правила оповещений и способы их доставки (email, SMS и т.д.).
4. Какие действия следует предпринять при обнаружении проблем во время мониторинга PostgreSQL?  
   *Ответ*: При обнаружении проблем во время мониторинга PostgreSQL необходимо проанализировать журналы ошибок, оптимизировать медленные запросы, настроить параметры конфигурации PostgreSQL, очистить неиспользуемые индексы и таблицы, а также при необходимости увеличить ресурсы сервера.
5. Как можно централизованно собирать данные мониторинга с нескольких серверов PostgreSQL?  
   *Ответ*: Для централизованного сбора данных мониторинга с нескольких серверов PostgreSQL можно использовать специализированные системы мониторинга, такие как Zabbix, Prometheus или решения для мониторинга баз данных, например, OmniDB, pgAdmin и др.
6. Что такое управление доступом в PostgreSQL?  
   *Ответ*: Управление доступом в PostgreSQL- это процесс контроля доступа пользователей к базе данных, схемам, таблицам и другим объектам.
7. Какие основные уровни доступа поддерживает PostgreSQL?  
   *Ответ*: PostgreSQL поддерживает уровни доступа к объектам базы данных, такие как SELECT, INSERT, UPDATE, DELETE, а также возможность управления доступом к объектам на уровне ролей и групп пользователей.
8. Как создать нового пользователя в PostgreSQL?  
   *Ответ*: Для создания нового пользователя в PostgreSQL используется команда CREATE USER, например:

SQL

CREATE USER new\_user WITH PASSWORD 'password';

1. Как назначить права доступа пользователю в PostgreSQL?  
   Права доступа назначаются с помощью команды GRANT, например:

*Ответ*: SQL

GRANT SELECT, INSERT ON table\_name TO user\_name;

1. Как можно отозвать права доступа у пользователя в PostgreSQL?  
   ПравадоступаотзываютсяспомощьюкомандыREVOKE, например:

*Ответ*: SQL

REVOKE SELECT, INSERT ON table\_name FROM user\_name;

1. Какие системные роли существуют в PostgreSQL и как они используются?  
   *Ответ*: В PostgreSQL существуют различные системные роли, такие как SUPERUSER, CREATEDB, CREATEROLE, которые предоставляют различные привилегии пользователю.
2. Как ограничить доступ к конкретным объектам базы данных в PostgreSQL?  
   *Ответ*: Для ограничения доступа к конкретным объектам, таким как таблицы, схемы или столбцы, в PostgreSQL используются различные механизмы, такие как роли, правила, триггеры, а также уровень схемы и строки.
3. Что такое репликация в PostgreSQL?  
      *Ответ*: Репликация в PostgreSQL- это процесс создания и поддержания копий данных в реальном времени для обеспечения отказоустойчивости и повышения производительности.
4. Какие типы репликации поддерживает PostgreSQL?  
      *Ответ*: PostgreSQL поддерживает несколько типов репликации, включая мастер-мастер, мастер-слейв и каскадную репликацию.
5. В чем разница между мастер-мастер и мастер-слейв репликацией?  
      *Ответ*: В мастер-мастер репликации оба сервера могут принимать записи и синхронизировать их друг с другом, в то время как в мастер-слейв репликации только один сервер принимает записи, а другие сервера получают данные от него.
6. Как настроить репликацию в PostgreSQL?  
      *Ответ*: Для настройки репликации в PostgreSQL необходимо создать резервные копии базы данных, настроить файл конфигурации и настроить параметры репликации на серверах.
7. Как проверить состояние репликации в PostgreSQL?  
      *Ответ*: Для проверки состояния репликации в PostgreSQL можно использовать команды "pg\_stat\_replication" для просмотра активных реплик и "pg\_stat\_database" для просмотра состояния базы данных на репликах.

**Задание 2**. Практические занятия

Практическое занятие №4. «Установка ОС Linux.»

Цель работы: сформировать у студентов практические знания по установке ОС.

Проверяемые результаты обучения:

* приемы администрирования, сопровождения и отказоустойчивости базы данных.

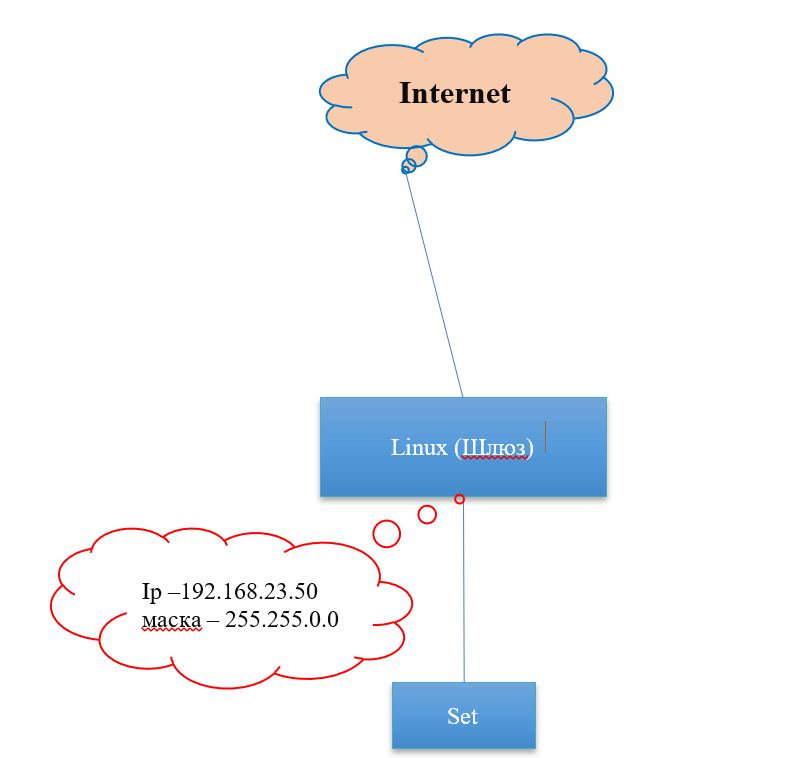
Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

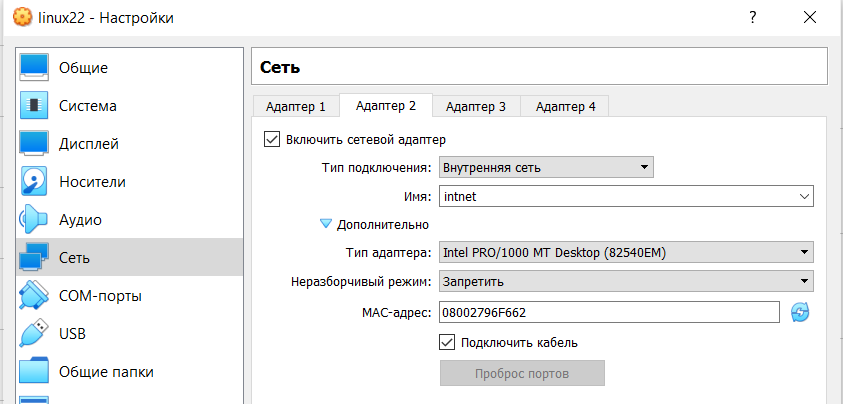
1. Выполнить установке ОС

**Ход работы**

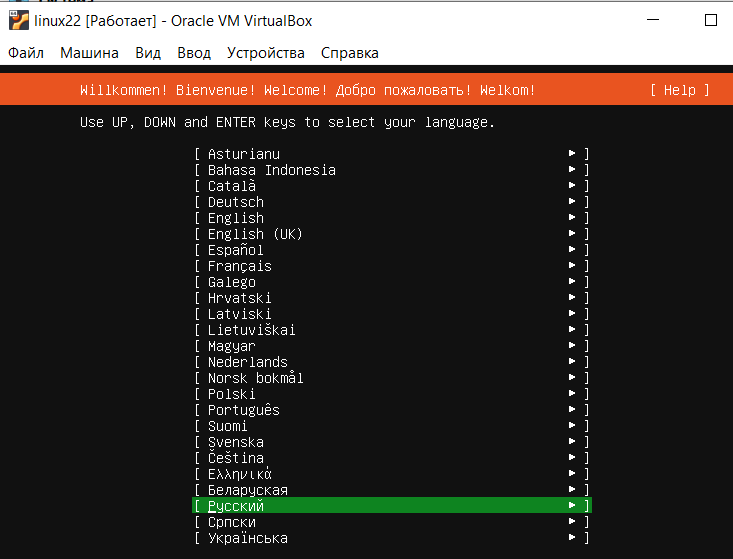


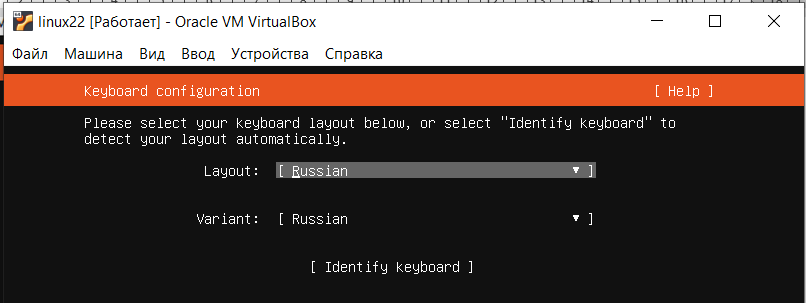
1. Создать виртуальную машину с именем **Linux\_ (номер группы)**

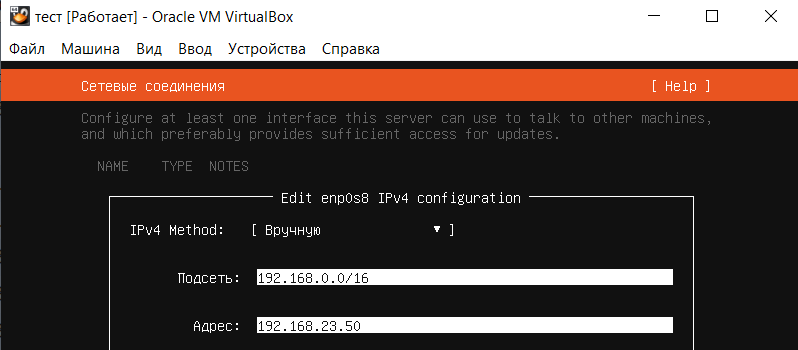
* Двухъядерный процессор 2 ГГц
* 2 ГБ RAM (системная память)
* 15 ГБ места на жестком диске.
* Добавить вторую сетевую карту



1. Включить виртуальную машину Linux (номер группы).



1. Выбрать язык установки Русский
2. Продолжить без обновления.
3. Выбрать русскую раскладку клавиатуры.
4. Выбрать вариант переключения раскладки клавиатуры.
5. Настроить сетевые параметры.



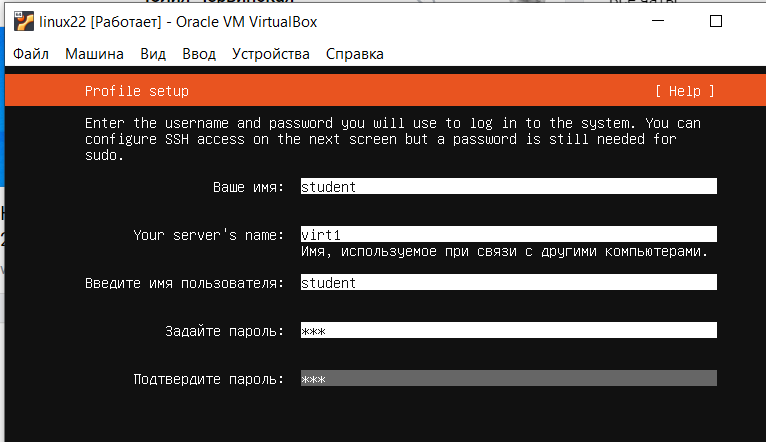
Вычислить номер сети для ip-адреса 192.168.23.50 и маски сети 255.255.0.0

Полученный номер сети внести в сетевые настройки.

1. Согласиться с предложенными настройками.
2. Указать настройки

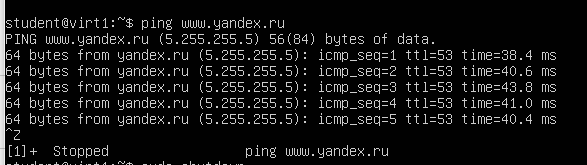
Пароль – 111

Дополнительные компоненты не устанавливать.



1. После перезагрузки компьютера и ввода пароля проверить наличие Internet.

Ввести команду pingya.ru и проверить пинг от сервера.

Если Internet работает, то вы увидите следующий экран

Прервать выполнение команды можно сочетанием клавиш Ctrl+Z

1. Выполнить установку обновлений.

* sudo apt update
* sudo apt upgrade

1. Завершить работу виртуальной машины командой

sudoshutdown -h now

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №5. Установка и управление сервером.

Цель работы: сформировать у студентов практические знания по установке и управлению сервером БД.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.

Формирование компетенций (указываются коды ОК и ПК):

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

4

Исходные коды

Установка из исходных кодов

стабильная версия сервера

можно использовать нестандартные параметры

или собрать на специфичной архитектуре

Установка из репозитория git

текущая версия сервера

необходимо в первую очередь для разработчиков ядра;

требует более широкого набора инструментов

Полезно иметь представление о том, как происходит установка из

исходных кодов, чтобы в случае необходимости можно было собрать

PostgreSQL с нестандартными параметрами или на специфичной

архитектуре.

По адресу <https://www.postgresql.org/ftp/source/v13.7/>доступен

исходный код версии 13.7 в двух вариантах (gzip и bzip2).

В виртуальной машине необходимый файл уже находится в домашнем

каталоге пользователя student

4

Исходные коды

Установка из исходных кодов

стабильная версия сервера

можно использовать нестандартные параметры

или собрать на специфичной архитектуре

Установка из репозитория git

текущая версия сервера

необходимо в первую очередь для разработчиков ядра;

требует более широкого набора инструментов

Полезно иметь представление о том, как происходит установка из

исходных кодов, чтобы в случае необходимости можно было собрать

PostgreSQL с нестандартными параметрами или на специфичной

архитектуре.

По адресу <https://www.postgresql.org/ftp/source/v13.7/>доступен

исходный код версии 13.7 в двух вариантах (gzip и bzip2).

В виртуальной машине необходимый файл уже находится в домашнем

каталоге пользователя student

4

Исходные коды

Установка из исходных кодов

стабильная версия сервера

можно использовать нестандартные параметры

или собрать на специфичной архитектуре

Установка из репозитория git

текущая версия сервера

необходимо в первую очередь для разработчиков ядра;

требует более широкого набора инструментов

Полезно иметь представление о том, как происходит установка из

исходных кодов, чтобы в случае необходимости можно было собрать

PostgreSQL с нестандартными параметрами или на специфичной

архитектуре.

По адресу <https://www.postgresql.org/ftp/source/v13.7/>доступен

исходный код версии 13.7 в двух вариантах (gzip и bzip2).

В виртуальной машине необходимый файл уже находится в домашнем

каталоге пользователя student

4

Исходные коды

Установка из исходных кодов

стабильная версия сервера

можно использовать нестандартные параметры

или собрать на специфичной архитектуре

Установка из репозитория git

текущая версия сервера

необходимо в первую очередь для разработчиков ядра;

требует более широкого набора инструментов

Полезно иметь представление о том, как происходит установка из

исходных кодов, чтобы в случае необходимости можно было собрать

PostgreSQL с нестандартными параметрами или на специфичной

архитектуре.

По адресу <https://www.postgresql.org/ftp/source/v13.7/>доступен

исходный код версии 13.7 в двух вариантах (gzip и bzip2).

В виртуальной машине необходимый файл уже находится в домашнем

каталоге пользователя student

4

Исходные коды

Установка из исходных кодов

стабильная версия сервера

можно использовать нестандартные параметры

или собрать на специфичной архитектуре

Установка из репозитория git

текущая версия сервера

необходимо в первую очередь для разработчиков ядра;

требует более широкого набора инструментов

Полезно иметь представление о том, как происходит установка из

исходных кодов, чтобы в случае необходимости можно было собрать

PostgreSQL с нестандартными параметрами или на специфичной

архитектуре.

По адресу <https://www.postgresql.org/ftp/source/v13.7/>доступен

исходный код версии 13.7 в двух вариантах (gzip и bzip2).

В виртуальной машине необходимый файл уже находится в домашнем

каталоге пользователя student

4Исходные кодыУстановка из исходных кодовстабильная версия сервераможно использовать нестандартные параметрыили собрать на специфичной архитектуреУстановка из репозитория gitтекущая версия серверанеобходимо в первую очередь для разработчиков ядра;требует более широкого набора инструментовПолезно иметь представление о том, как происходит установка из исходных кодов, чтобы в случае необходимости можно было собрать PostgreSQL с нестандартными параметрами или на специфичной архитектуре.По адресу https://www.postgresql.org/ftp/source/v13.7/ доступен исходный код версии 13.7 в двух вариантах (gzip и bzip2).В виртуальной машине необходимый файл уже находится в домашнем каталоге пользователя student

4

Исходные коды

Установка из исходных кодов

стабильная версия сервера

можно использовать нестандартные параметры

или собрать на специфичной архитектуре

Установка из репозитория git

текущая версия сервера

необходимо в первую очередь для разработчиков ядра;

требует более широкого набора инструментов

Полезно иметь представление о том, как происходит установка из

исходных кодов, чтобы в случае необходимости можно было собрать

PostgreSQL с нестандартными параметрами или на специфичной

архитектуре.

По адресу <https://www.postgresql.org/ftp/source/v13.7/>доступен

исходный код версии 13.7 в двух вариантах (gzip и bzip2).

В виртуальной машине необходимый файл уже находится в домашнем

каталоге пользователя student

4

Исходные коды

Установка из исходных кодов

стабильная версия сервера

можно использовать нестандартные параметры

или собрать на специфичной архитектуре

Установка из репозитория git

текущая версия сервера

необходимо в первую очередь для разработчиков ядра;

требует более широкого набора инструментов

Полезно иметь представление о том, как происходит установка из

исходных кодов, чтобы в случае необходимости можно было собрать

PostgreSQL с нестандартными параметрами или на специфичной

архитектуре.

По адресу <https://www.postgresql.org/ftp/source/v13.7/>доступен

исходный код версии 13.7 в двух вариантах (gzip и bzip2).

В виртуальной машине необходимый файл уже находится в домашнем

каталоге пользователя student

По адресу <https://www.postgresql.org/ftp/source/v13.7/>доступен исходный код версии 13.7 в двух вариантах (gzip и bzip2).

В виртуальной машине необходимый файл уже находится в домашнем каталоге пользователя student.

**Распаковка архива**

Архив с исходными кодами находится в домашнем каталоге пользователя student. Распаковываем архив и переходим в каталог с исходным кодом:

**student$ tarxzf /home/student/postgresql-13.6.tar.gz**

**student$ ls -ld /home/student/postgresql-13.6**

ls: cannot access '/home/student/postgresql-13.6': No such file or directory

**student$ cd /home/student/postgresql-13.6**

**Создание конфигурации**

Если требуется повторно выполнить конфигурацию, например, с другими параметрами, то предварительно нужно очистить результаты предыдущего запуска:

**student$ makedistclean**

В команде configure можно указать различные параметры конфигурации. Например:

* --prefix — каталог установки, по умолчанию /usr/local/pgsql;
* --enable-debug — для включения отладочной информации.

Установим сервер в домашний каталог пользователя student. Кроме того, будем использовать порт 5555 вместо порта 5432 по умолчанию:

**student$./configure --prefix=/home/student/pgsql13 --with-pgport=5555**

Вывод этой и следующих команд занимает очень много места, мы его не будем показывать.

**Сборка и установка PostgreSQL**

Возможные варианты:

* make — сборка только сервера;
* makeworld — сборка сервера, всех расширений и документации.

Собираем сервер. В зависимости от характеристик компьютера команда make может работать достаточно долго (минуты и даже десятки минут).

**student$ make**

Теперь устанавливаем сервер.

**student$ makeinstall**

Для сборки расширений нужно повторить последние две команды, перейдя в подкаталог contrib.

**student$ cd /home/student/postgresql-13.6/contrib**

**student$ make**

**student$ make install**

**Созданиекластера**

Теперь необходимо создать каталог для данных. Для промышленного использования владельцем каталога обычно назначают отдельного пользователя ОС. В нашем случае, пусть это будет каталог пользователя student внутри /home/student/pgsql13

**student$ mkdir /home/student/pgsql13/data**

Этот каталог часто называют PGDATA по имени переменной окружения, которую удобно использовать при работе с утилитами сервера.

**student$ export PGDATA=/home/student/pgsql13/data**

Все утилиты сервера установлены в подкаталоге bin, который стоит добавить в переменную PATH:

**student$ export PATH=/home/student/pgsql13/bin:$PATH**

Для инициализации кластера базы данных используется утилита initdb.

* В ключе -U можно указать имя суперпользователя СУБД. Если его не задать, то используется имя пользователя ОС, в нашем случае student. Обычно таким пользователем является postgres, так что зададим это имя явно.
* Ключ -k включает подсчет контрольных суммы страниц, что позволяет своевременно обнаруживать повреждение данных.
* Если переменная PGDATA не установлена, то следует в ключе -D указать каталог для данных. В нашем случае этот ключ можно было бы не указывать.

**student$ initdb -U postgres -k -D /home/student/pgsql13/data**

**Управление сервером**

Теперь все готово к запуску сервера.

* В ключе -l укажем файл журнала сообщений сервера.
* Ключ -D опускаем, поскольку задана переменная PGDATA.

**student$ pg\_ctl start -l /home/student/logfile**

Для проверки работы можно вызвать утилиту psql, которая подключится к серверу и вернет текущее время:

**student$ psql -U postgres -p 5555 -c 'SELECT now();'**

now

-------------------------------

2024-01-16 10:28:42.094348+03

(1 row)

Для останова сервера используется команда:

**student$ pg\_ctl stop -m fast**

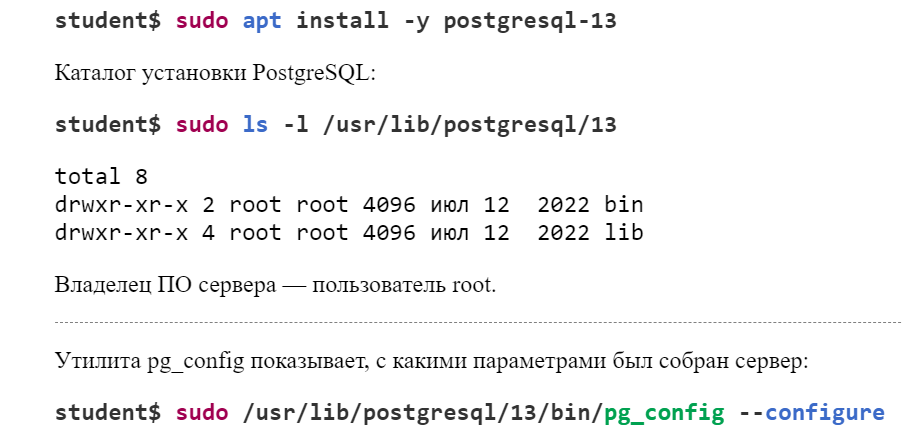
В ключе -m можно указать один из трех режимов останова:

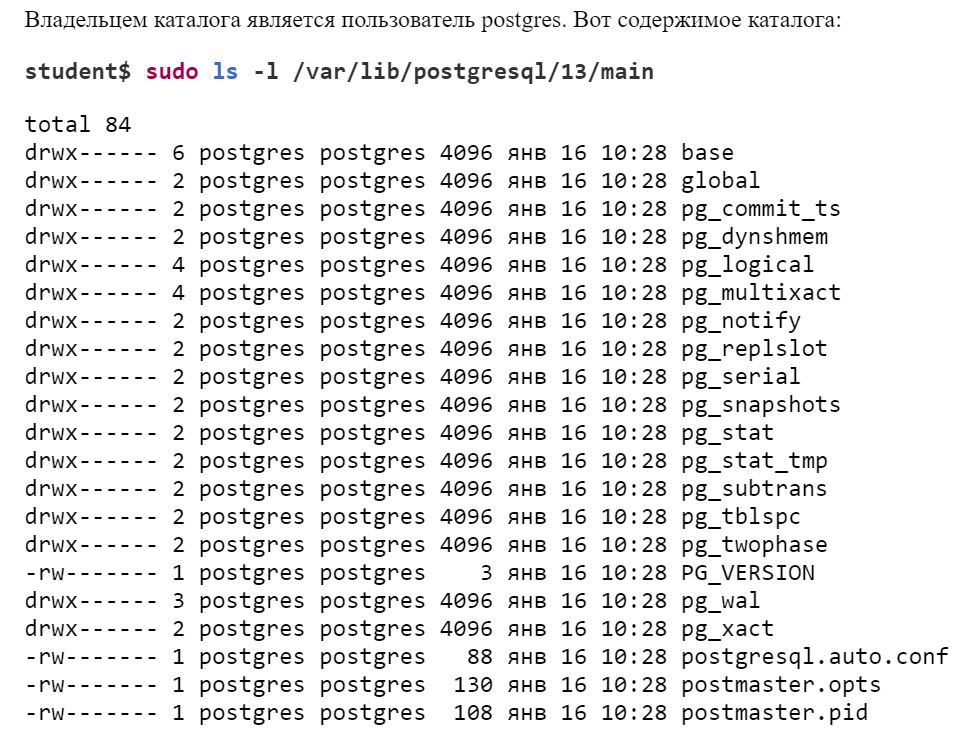
* fast — принудительно завершает сеансы и записывает на диск изменения из оперативной памяти;
* smart — ожидает завершения всех сеансов и записывает на диск изменения из оперативной памяти;
* immediate — принудительно завершает сеансы, при запуске потребуется восстановление.

По умолчанию используется режим fast.

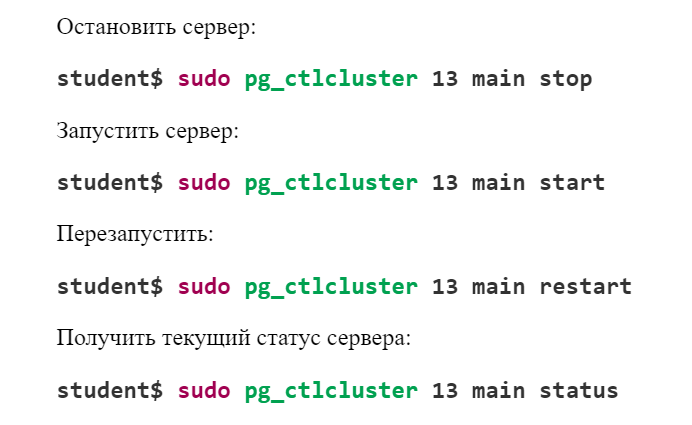
1. **Установка пакета в Ubuntu**

# Создание кластера в Ubuntu





Управление сервером в Ubuntu.



pg\_ctl: server is running (PID: 3035)

/usr/lib/postgresql/13/bin/postgres "-D" "/var/lib/postgresql/13/main" "-c" "config\_file=/etc/postgresql/13/main/postgresql.conf"

Перечитатьфайлыконфигурации:

**student$ sudopg\_ctlcluster 13 main reload**

Утилита pg\_ctlcluster позволяет управлять несколькими серверами разных версий. В качестве параметров утилите передаются:

* 13 — номер версии сервера;
* main — название сервера.

Журнал сообщений сервера при пакетной установке находится здесь:

**student$ ls -l /var/log/postgresql/postgresql-13-main.log**

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №6 «Создание базы данных в СУБД.»

Цель работы: сформировать у студентов практические знания по созданию баз данных в СУБД.

Проверяемые результаты обучения:

* создавать схемы баз данных в различных СУБД;
* основные положения теории баз данных, хранилищ данных, баз знаний;
* методы описания схем баз данных в современных системах управления базами данных;
* структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

1. Ознакомится с шаблонами БД.
2. Выполнить создание БД.
3. Выполнить изменение созданной БД.

**Ход работы:**

Список доступных БД

~~~~~~~~~~~~~~~~~~~

=> \l

Listofdatabases

Name| Owner | Encoding | Collate | Ctype | Accessprivileges

-----------+----------+----------+-------------+-------------+-----------------------

postgres |postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |

template0 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +

| | | | | postgres=CTc/postgres

(3 rows)

**Подключениек template1**

~~~~~~~~~~~~~~~~~~~~~~~

=> \c template1 postgres

You are now connected to database "template1" as user "postgres".

Для создания новой базы данных служит команда CREATE DATABASE:

=>create database test;

CREATE DATABASE

=>\c test

You are now connected to database "test" as user "postgres".

=>select datname, datistemplate, datallowconn, datconnlimit from pg\_database;

datname |datistemplate | datallowconn | datconnlimit

-----------+---------------+--------------+--------------

template1 | t | t | -1

template0 | t | f | -1

postgres | f | t | -1

test | f | t | -1

(4 rows)

-----------------------------------------------------------------------

А если создать базу данных из шаблона template0?

=>create database test0 template template0 connection limit 20;

CREATE DATABASE

=>\c test0

You are now connected to database "test0" as user "postgres".

Здесь мы указали не только имя шаблона, но и максимальное количество соединений:

=>select datname, datistemplate, datallowconn, datconnlimit from pg\_database;

datname |datistemplate | datallowconn | datconnlimit

-----------+---------------+--------------+--------------

template1 | t | t | -1

template0 | t | f | -1

postgres | f | t | -1

test | f | t | -1

test0 | f | t | 20

(5 rows)

-----------------------------------------------------------------------

-----------------------------------------------------------------------

Рассмотрим несколько полезных функций для вычисления занимаемого объема.

Размербазыданных:

=>select pg\_database\_size('test0');

pg\_database\_size

------------------

6434132

(1 row)

Чтобы не считать разряды, можно вывести размер в читаемом виде:

=>select pg\_size\_pretty(pg\_database\_size('test0'));

pg\_size\_pretty

----------------

6283 kB

(1 row)

-----------------------------------------------------------------------

Создадимтаблицуииндекс:

=>create table t(n numeric);

CREATE TABLE

=>create index t\_idx on t(n);

CREATE INDEX

=>insert into t select \* from generate\_series(1, 10000);

INSERT 0 10000

-----------------------------------------------------------------------

Теперь можно вывести размер, занимаемый таблицей:

=>select pg\_size\_pretty(pg\_table\_size('t'));

pg\_size\_pretty

----------------

392 kB

(1 row)

-----------------------------------------------------------------------

А также размер индексов таблицы:

=>select pg\_size\_pretty(pg\_indexes\_size('t'));

pg\_size\_pretty

----------------

240 kB

(1 row)

И размер таблицы вместе с индексами:

=>select pg\_size\_pretty(pg\_total\_relation\_size('t'));

pg\_size\_pretty

----------------

632 kB

(1 row)

-----------------------------------------------------------------------

Созданную базу данных можно переименовать:

=>alter database test rename to db;

ALTER DATABASE

=>select datname, datistemplate, datallowconn, datconnlimit from pg\_database;

datname |datistemplate | datallowconn | datconnlimit

-----------+---------------+--------------+--------------

template1 | t | t | -1

template0 | t | f | -1

postgres | f | t | -1

test0 | f | t | 20

db | f | t | -1

(5 rows)

-----------------------------------------------------------------------

Изменитьколичествосоединений:

=> alter database db connection limit 10;

ALTER DATABASE

=>select datname, datistemplate, datallowconn, datconnlimit from pg\_database;

datname |datistemplate | datallowconn | datconnlimit

-----------+---------------+--------------+--------------

template1 | t | t | -1

template0 | t | f | -1

postgres | f | t | -1

test0 | f | t | 20

db | f | t | 10

(5 rows)

-----------------------------------------------------------------------

Базу данных можно удалить (если к ней нет активных подключений).

Поскольку мы подключены к другой базе, то удаление пройдет успешно:

=>\conninfo

You are connected to database "test0" as user "postgres" via socket in "/tmp" at port "5432".

=> drop database db;

DROP DATABASE

=>select datname, datistemplate, datallowconn, datconnlimit from pg\_database;

datname |datistemplate | datallowconn | datconnlimit

-----------+---------------+--------------+--------------

template1 | t | t | -1

template0 | t | f | -1

postgres | f | t | -1

test0 | f | t | 20

(4 rows)

-----------------------------------------------------------------------

Удалим и вторую базу.

=>\c postgres

You are now connected to database "postgres" as user "postgres".

=>drop database test0;

DROP DATABASE

=>select datname, datistemplate, datallowconn, datconnlimit from pg\_database;

datname |datistemplate | datallowconn | datconnlimit

-----------+---------------+--------------+--------------

template1 | t | t | -1

template0 | t | f | -1

postgres | f | t | -1

(3 rows)

-----------------------------------------------------------------------

Если база данных не существует, команда DROP DATABASE будет ругаться:

=>drop database db;

ERROR: database "db" does not exist

Поэтому можно воспользоваться другой формой (она применима и к удалению других объектов):

=>drop database if exists db;

NOTICE: database "db" does not exist, skipping

DROP DATABASE

Конец демонстрации.

-----------------------------------------------------------------------

=> \q

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №7 «Табличные пространства»

Цель работы: сформировать у студентов практические знания по работе с табличными пространствами.

Проверяемые результаты обучения:

* основные положения теории баз данных, хранилищ данных, баз знаний;
* применение стандартных методов для защиты объектов базы данных;
* основные методы и средства защиты данных в базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Изучить работу с табличными пространствами в СУБД.

При создании кластера создаются два табличных пространства:

**=> select \* from pg\_tablespace;**

spcname | spcowner | spcacl | spcoptions

------------+----------+--------+------------

pg\_default | 10 | |

pg\_global | 10 | |

(2 rows)

\* pg\_global - общие объекты кластера,

\* pg\_default - табличное пространство по умолчанию.

-----------------------------------------------------------------------

Для нового табличного пространства нужен пустой каталог,

владельцем которого является пользователь postgres (*выйти из psql*).

**cd /home/postgres**

**mkdir ts1\_dir**

**ls -lts1\_dir- вывод информации о файлах и каталогах**

total 0

-----------------------------------------------------------------------

Теперь можно создать табличное пространство:

**=> create tablespace ts1 location '/home/postgres/ts1\_dir';**

CREATE TABLESPACE

Список можно получить и командой psql:

**=> \db**

List of tablespaces

Name | Owner | Location

------------+----------+------------------------

pg\_default | postgres |

pg\_global |postgres |

ts1 | postgres | /home/postgres/ts1\_dir

(3 rows)

-----------------------------------------------------------------------

Табличное пространство может использоваться несколькими базами кластера.

При этом у каждой базы есть табличное пространство по умолчанию.

Создадим БД и назначим ей ts1 в качестве пространства по умолчанию:

**=> create database test tablespace ts1;**

CREATE DATABASE

-----------------------------------------------------------------------

Теперь все создаваемые таблицы и индексы будут попадать в ts1,

если явно не указать другое.

Подключимсякбазе:

**=> \c test**

You are now connected to database "test" as user "postgres".

Создадимтаблицу:

**=> create table t1(id serial, name text);**

CREATE TABLE

=> select \* from pg\_tables where tablename='t1';

schemaname | tablename | tableowner | tablespace | hasindexes | hasrules | hastriggers

------------+-----------+------------+------------+------------+----------+-------------

public | t1 | postgres | | f | f | f

(1 row)

**Столбцы pg\_tables**

| **Имя** | **Тип** | **Ссылки** | **Описание** |
| --- | --- | --- | --- |
| schemaname | name | [pg\_namespace](https://postgrespro.ru/docs/postgresql/9.5/catalog-pg-namespace).nspname | Имя схемы, содержащей таблицу |
| tablename | name | [pg\_class](https://postgrespro.ru/docs/postgresql/9.5/catalog-pg-class).relname | Имя таблицы |
| tableowner | name | [pg\_authid](https://postgrespro.ru/docs/postgresql/9.5/catalog-pg-authid).rolname | Имя владельца таблицы |
| tablespace | name | [pg\_tablespace](https://postgrespro.ru/docs/postgresql/9.5/catalog-pg-tablespace).spcname | Имя табличного пространства, содержащего таблицу (NULL, если это пространство по умолчанию) |
| hasindexes | boolean | [pg\_class](https://postgrespro.ru/docs/postgresql/9.5/catalog-pg-class).relhasindex | True, если эта таблица имеет (или недавно имела) индексы |
| hasrules | boolean | [pg\_class](https://postgrespro.ru/docs/postgresql/9.5/catalog-pg-class).relhasrules | True, если для таблицы определены (или были определены) правила |
| hastriggers | boolean | [pg\_class](https://postgrespro.ru/docs/postgresql/9.5/catalog-pg-class).relhastriggers | True, если для таблицы определены (или были определены) триггеры |
| rowsecurity | boolean | [pg\_class](https://postgrespro.ru/docs/postgresql/9.5/catalog-pg-class).relrowsecurity | True, если для таблицы включена защита строк |

Пустое поле tablespace указывает на табличное пространство по умолчанию.

-----------------------------------------------------------------------

Можно изменить табличное пространство по умолчанию, но при этом все таблицы

физически переносятся из одного пространства в другое.

Предварительно надо отключиться от базы.

**=> \c postgres**

You are now connected to database "postgres" as user "postgres".

**=> alter database test set tablespace pg\_default;**

ALTER DATABASE

**=> \c test**

You are now connected to database "test" as user "postgres".

-----------------------------------------------------------------------

При создании объекта можно явно указать табличное пространство:

**=> create table t2(n numeric) tablespace ts1;**

CREATE TABLE

**=> select \* from pg\_tables where tablename like 't\_';**

schemaname | tablename | tableowner | tablespace | hasindexes | hasrules | hastriggers

------------+-----------+------------+------------+------------+----------+-------------

public | t1 | postgres | | f | f | f

public | t2 | postgres | ts1 | f | f | f

(2 rows)

Уже созданную таблицу (или индекс) можно переместить в другое табличное пространство.

Создадим еще одно табличное пространство ts2 и переместим в него таблицу t1.

**mkdir ts2\_dir**

**=> create tablespace ts2 location '/home/postgres/ts2\_dir';**

CREATE TABLESPACE

**=> alter table t1 set tablespace ts2;**

ALTER TABLE

=> select \* from pg\_tables where tablename like 't\_';

schemaname | tablename | tableowner | tablespace | hasindexes | hasrules | hastriggers

-----------+-----------+------------+------------+------------+----------+-------------

public | t1 | postgres | ts2 | t | f | f

public | t2 | postgres | ts1 | f | f | f

(2 rows)

-----------------------------------------------------------------------

Можно переместить и все таблицы сразу:

**=> alter table all in tablespace ts1 set tablespace ts2;**

ALTER TABLE

**=> select \* from pg\_tables where tablename like 't\_';**

schemaname | tablename | tableowner | tablespace | hasindexes | hasrules | hastriggers

-----------+-----------+------------+------------+------------+----------+-------------

public | t1 | postgres | ts2 | t | f | f

public | t2 | postgres | ts2 | f | f | f

(2 rows)

-----------------------------------------------------------------------

Табличное пространство можно переименовать:

**=> alter tablespace ts1 rename to test\_tablespace;**

ALTER TABLESPACE

Атакжеудалить...

**=> drop tablespace test\_tablespace;**

DROP TABLESPACE

**=> drop tablespace ts2;**

ERROR: tablespace "ts2" is not empty

...но только, если оно пусто.

-----------------------------------------------------------------------

**=> drop table t1;**

DROP TABLE

**=> drop table t2;**

DROP TABLE

**=> drop tablespace if exists ts2;**

DROP TABLESPACE

Конец демонстрации.

-----------------------------------------------------------------------

=> \q

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №8 «Работа с системным каталогом»

Цель работы: сформировать у студентов практические знания по работе с системным каталогом СУБД.

Проверяемые результаты обучения:

* основные положения теории баз данных, хранилищ данных, баз знаний;
* применять стандартные методы для защиты объектов базы данных;
* структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Изучить работу с системным катогом.

Системный каталог

~~~~~~~~~~~~~~~~~

Создадим БД и тестовые объекты

psql

CREATE DATABASE dba1\_07\_demo;

CREATE DATABASE

psql dba1\_07\_demo

=>create table master\_table (id int primary key, c text not null);

CREATE TABLE

=>comment on table master\_table is 'Тестоваятаблица';

COMMENT

=>create table detail\_table (id int references master\_table(id), c text);

CREATE TABLE

=>create view my\_view as select \* from detail\_table;

CREATE VIEW

=>comment on column my\_view.id is 'Идентификатор';

COMMENT

-----------------------------------------------------------------------

Список таблиц системного словаря:

=>\dt pg\_catalog.\*

List of relations

Schema | Name | Type | Owner

------------+-------------------------+-------+----------

pg\_catalog | pg\_aggregate | table | postgres

pg\_catalog | pg\_am | table | postgres

pg\_catalog | pg\_amop | table | postgres

pg\_catalog | pg\_amproc | table | postgres

pg\_catalog | pg\_attrdef | table | postgres

pg\_catalog | pg\_attribute | table | postgres

pg\_catalog | pg\_auth\_members | table | postgres

pg\_catalog | pg\_authid | table | postgres

pg\_catalog | pg\_cast | table | postgres

pg\_catalog | pg\_class | table | postgres

pg\_catalog | pg\_collation | table | postgres

pg\_catalog | pg\_constraint | table | postgres

pg\_catalog | pg\_conversion | table | postgres

pg\_catalog | pg\_database | table | postgres

pg\_catalog | pg\_db\_role\_setting | table | postgres

pg\_catalog | pg\_default\_acl | table | postgres

pg\_catalog | pg\_depend | table | postgres

pg\_catalog | pg\_description | table | postgres

pg\_catalog | pg\_enum | table | postgres

pg\_catalog | pg\_event\_trigger | table | postgres

pg\_catalog | pg\_extension | table | postgres

pg\_catalog | pg\_foreign\_data\_wrapper | table | postgres

pg\_catalog | pg\_foreign\_server | table | postgres

pg\_catalog | pg\_foreign\_table | table | postgres

pg\_catalog | pg\_index | table | postgres

pg\_catalog | pg\_inherits | table | postgres

pg\_catalog | pg\_language | table | postgres

pg\_catalog | pg\_largeobject | table | postgres

pg\_catalog | pg\_largeobject\_metadata | table | postgres

pg\_catalog | pg\_namespace | table | postgres

pg\_catalog | pg\_opclass | table | postgres

pg\_catalog | pg\_operator | table | postgres

pg\_catalog | pg\_opfamily | table | postgres

pg\_catalog | pg\_pltemplate | table | postgres

pg\_catalog | pg\_proc | table | postgres

pg\_catalog | pg\_range | table | postgres

pg\_catalog | pg\_rewrite | table | postgres

pg\_catalog | pg\_seclabel | table | postgres

pg\_catalog | pg\_shdepend | table | postgres

pg\_catalog | pg\_shdescription | table | postgres

pg\_catalog | pg\_shseclabel | table | postgres

pg\_catalog | pg\_statistic | table | postgres

pg\_catalog | pg\_tablespace | table | postgres

pg\_catalog | pg\_trigger | table | postgres

pg\_catalog | pg\_ts\_config | table | postgres

pg\_catalog | pg\_ts\_config\_map | table | postgres

pg\_catalog | pg\_ts\_dict | table | postgres

pg\_catalog | pg\_ts\_parser | table | postgres

pg\_catalog | pg\_ts\_template | table | postgres

pg\_catalog | pg\_type | table | postgres

pg\_catalog | pg\_user\_mapping | table | postgres

(51 rows)

-----------------------------------------------------------------------

Список представлений системного словаря:

=>\dv pg\_catalog.\*

List of relations

Schema | Name | Type | Owner

------------+---------------------------------+------+----------

pg\_catalog | pg\_available\_extension\_versions | view | postgres

pg\_catalog | pg\_available\_extensions | view | postgres

pg\_catalog | pg\_cursors | view | postgres

pg\_catalog | pg\_group | view | postgres

pg\_catalog | pg\_indexes | view | postgres

pg\_catalog | pg\_locks | view | postgres

pg\_catalog | pg\_matviews | view | postgres

pg\_catalog | pg\_prepared\_statements | view | postgres

pg\_catalog | pg\_prepared\_xacts | view | postgres

pg\_catalog | pg\_replication\_slots | view | postgres

pg\_catalog | pg\_roles | view | postgres

pg\_catalog | pg\_rules | view | postgres

pg\_catalog | pg\_seclabels | view | postgres

pg\_catalog | pg\_settings | view | postgres

pg\_catalog | pg\_shadow | view | postgres

pg\_catalog | pg\_stat\_activity | view | postgres

pg\_catalog | pg\_stat\_all\_indexes | view | postgres

pg\_catalog | pg\_stat\_all\_tables | view | postgres

pg\_catalog | pg\_stat\_archiver | view | postgres

pg\_catalog | pg\_stat\_bgwriter | view | postgres

pg\_catalog | pg\_stat\_database | view | postgres

pg\_catalog | pg\_stat\_database\_conflicts | view | postgres

pg\_catalog | pg\_stat\_replication | view | postgres

pg\_catalog | pg\_stat\_sys\_indexes | view | postgres

pg\_catalog | pg\_stat\_sys\_tables | view | postgres

pg\_catalog | pg\_stat\_user\_functions | view | postgres

pg\_catalog | pg\_stat\_user\_indexes | view | postgres

pg\_catalog | pg\_stat\_user\_tables | view | postgres

pg\_catalog | pg\_stat\_xact\_all\_tables | view | postgres

pg\_catalog | pg\_stat\_xact\_sys\_tables | view | postgres

pg\_catalog | pg\_stat\_xact\_user\_functions | view | postgres

pg\_catalog | pg\_stat\_xact\_user\_tables | view | postgres

pg\_catalog | pg\_statio\_all\_indexes | view | postgres

pg\_catalog | pg\_statio\_all\_sequences | view | postgres

pg\_catalog | pg\_statio\_all\_tables | view | postgres

pg\_catalog | pg\_statio\_sys\_indexes | view | postgres

pg\_catalog | pg\_statio\_sys\_sequences | view | postgres

pg\_catalog | pg\_statio\_sys\_tables | view | postgres

pg\_catalog | pg\_statio\_user\_indexes | view | postgres

pg\_catalog | pg\_statio\_user\_sequences | view | postgres

pg\_catalog | pg\_statio\_user\_tables | view | postgres

pg\_catalog | pg\_stats | view | postgres

pg\_catalog | pg\_tables | view | postgres

pg\_catalog | pg\_timezone\_abbrevs | view | postgres

pg\_catalog | pg\_timezone\_names | view | postgres

pg\_catalog | pg\_user | view | postgres

pg\_catalog | pg\_user\_mappings | view | postgres

pg\_catalog | pg\_views | view | postgres

(48 rows)

-----------------------------------------------------------------------

Список системных функций для определения размера объектов:

=>\x

Expanded display is on.

=>\df pg\_catalog.pg\_\*size

List of functions

-[ RECORD 1 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_column\_size

Result data type | integer

Argument data types | "any"

Type | normal

-[ RECORD 2 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_database\_size

Result data type | bigint

Argument data types | name

Type | normal

-[ RECORD 3 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_database\_size

Result data type | bigint

Argument data types | oid

Type | normal

-[ RECORD 4 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_indexes\_size

Result data type | bigint

Argument data types | regclass

Type | normal

-[ RECORD 5 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_relation\_size

Result data type | bigint

Argument data types | regclass

Type | normal

-[ RECORD 6 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_relation\_size

Result data type | bigint

Argument data types | regclass, text

Type | normal

-[ RECORD 7 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_table\_size

Result data type | bigint

Argument data types | regclass

Type | normal

-[ RECORD 8 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_tablespace\_size

Result data type | bigint

Argument data types | name

Type | normal

-[ RECORD 9 ]-------+-----------------------

Schema | pg\_catalog

Name | pg\_tablespace\_size

Result data type | bigint

Argument data types | oid

Type | normal

-[ RECORD 10 ]------+-----------------------

Schema | pg\_catalog

Name | pg\_total\_relation\_size

Result data type | bigint

Argument data types | regclass

Type | normal

=>\x

Expanded display is off.

-----------------------------------------------------------------------

Получение описания функции:

=>\sfpg\_table\_size

CREATE OR REPLACE FUNCTION pg\_catalog.pg\_table\_size(regclass)

RETURNS bigint

LANGUAGE internal

STRICT

AS $function$pg\_table\_size$function$

Редактирование функции: \effunction\_name

-----------------------------------------------------------------------

Получение размера баз данных кластера:

=>select pd.datname as database\_name

=> ,pg\_database\_size (pd.oid) as db\_size\_bytes

=> ,pg\_size\_pretty(

=>pg\_database\_size (pd.oid)

=> ) as db\_size\_text

=> from pg\_database pd;

database\_name | db\_size\_bytes | db\_size\_text

---------------+---------------+--------------

template1 | 6466900 | 6315 kB

template0 | 6332932 | 6185 kB

postgres | 6442324 | 6291 kB

dba1\_07\_demo | 6532436 | 6379 kB

test | 6589780 | 6435 kB

(5 rows)

-----------------------------------------------------------------------

Команды \d\* позволяют быстро получить информацию

из таблиц системного словаря.

Получить список таблиц:

=>\dt

List of relations

Schema | Name | Type | Owner

--------+--------------+-------+----------

public | detail\_table | table | postgres

public | master\_table | table | postgres

(2 rows)

-----------------------------------------------------------------------

Получить список представлений:

=>\dv

Listofrelations

Schema | Name | Type | Owner

--------+---------+------+----------

public | my\_view | view | postgres

(1 row)

-----------------------------------------------------------------------

Одной командой можно получить список нескольких типов объектов:

\dtvismE

t - таблицы

v - представления

i - индексы

s - последовательности

m - материализованные представления

E - внешниетаблицы

=>\dtvi

List of relations

Schema | Name | Type | Owner | Table

--------+-------------------+-------+----------+--------------

public | detail\_table | table | postgres |

public | master\_table | table | postgres |

public | master\_table\_pkey | index | postgres | master\_table

public | my\_view | view |postgres |

(4 rows)

-----------------------------------------------------------------------

Модификатор + позволяет получить дополнительную информацию.

Состав дополнительной информации предопределен и зависит от типа объекта

=>\dt+

List of relations

Schema | Name | Type | Owner | Size | Description

--------+--------------+-------+----------+------------+------------------

public | detail\_table | table | postgres | 8192 bytes |

public | master\_table | table | postgres | 8192 bytes | Тестоваятаблица

(2 rows)

-----------------------------------------------------------------------

Модификатор S выводит системные объекты в дополнении к пользовательским.

=>\dtS \*table\*

List of relations

Schema | Name | Type | Owner

------------+------------------+-------+----------

pg\_catalog | pg\_foreign\_table | table | postgres

pg\_catalog | pg\_tablespace | table | postgres

public | detail\_table | table | postgres

public | master\_table | table | postgres

(4 rows)

-----------------------------------------------------------------------

Для получения информации об одном объекте

(таблица, представление, индекс, последовательность, внешняя таблица):

=>\d master\_table

Table "public.master\_table"

Column | Type | Modifiers

--------+---------+-----------

id | integer | not null

c | text | not null

Indexes:

"master\_table\_pkey" PRIMARY KEY, btree (id)

Referenced by:

TABLE "detail\_table" CONSTRAINT "detail\_table\_id\_fkey" FOREIGN KEY (id) REFERENCES master\_table(id)

-----------------------------------------------------------------------

Модификатор + работает и для \d:

=>\d+ my\_view

View "public.my\_view"

Column | Type | Modifiers | Storage | Description

--------+---------+-----------+----------+---------------

id | integer | | plain | Идентификатор

c | text | | extended |

View definition:

SELECT detail\_table.id,

detail\_table.c

FROM detail\_table;

-----------------------------------------------------------------------

Использование REG\* типов.

Напишем запрос, который выводит столбцы представления my\_view.

Нам потребуются таблицы системного каталога:

pg\_class - список объектов, включая представления

pg\_attribute - списокстолбцов

-----------------------------------------------------------------------

=>select oid, relname from pg\_class limit 5;

oid |relname

-------+----------------------

2619 | pg\_statistic

1247 | pg\_type

24752 | pg\_toast\_24749

24754 | pg\_toast\_24749\_index

2840 | pg\_toast\_2619

(5 rows)

=>select attrelid, attname from pg\_attribute limit 5;

attrelid | attname

----------+--------------

1255 | proname

1255 | pronamespace

1255 | proowner

1255 | prolang

1255 | procost

(5 rows)

-----------------------------------------------------------------------

Дветаблицысвязаныпоpg\_class.OID - pg\_attribute.attrelid

Запрос на соединение двух таблиц:

=>select c.attrelid, c.attname

=>from pg\_attribute c, pg\_class t

=>wherec.attrelid = t.oid

=>and t.relname = 'my\_view';

attrelid | attname

----------+---------

24768 | c

24768 | id

(2 rows)

-----------------------------------------------------------------------

Запроссиспользованиемregclass:

=>select attrelid, attname

=>from pg\_attribute

=>whereattrelid = 'my\_view'::regclass;

attrelid | attname

----------+---------

24768 | c

24768 | id

(2 rows)

-----------------------------------------------------------------------

=>\q

Посмотрим на запрос, которые выполняет psql для команды \dt

psql dba1\_07\_demo -E -c '\dt'

\*\*\*\*\*\*\*\*\* QUERY \*\*\*\*\*\*\*\*\*\*

SELECT n.nspname as "Schema",

c.relname as "Name",

CASE c.relkind WHEN 'r' THEN 'table' WHEN 'v' THEN 'view' WHEN 'm' THEN 'materialized view' WHEN 'i' THEN 'index' WHEN 'S' THEN 'sequence' WHEN 's' THEN 'special' WHEN 'f' THEN 'foreign table' END as "Type",

pg\_catalog.pg\_get\_userbyid(c.relowner) as "Owner"

FROM pg\_catalog.pg\_class c

LEFT JOIN pg\_catalog.pg\_namespace n ON n.oid = c.relnamespace

WHERE c.relkind IN ('r','')

AND n.nspname<> 'pg\_catalog'

AND n.nspname<> 'information\_schema'

AND n.nspname !~ '^pg\_toast'

AND pg\_catalog.pg\_table\_is\_visible(c.oid)

ORDER BY 1,2;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

List of relations

Schema | Name | Type | Owner

--------+--------------+-------+----------

public | detail\_table | table | postgres

public | master\_table | table | postgres

(2 rows)

-----------------------------------------------------------------------

psqlpostgres -c "DROP DATABASE dba1\_07\_demo;"

DROP DATABASE

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №9 Создание, изменение таблиц.

Цель работы: сформировать у студентов практические знания по созданию и изменению таблиц в БД.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* создавать схемы баз данных в различных СУБД;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

* + - 1. Выполнить создание таблиц в СУБД
      2. Выполнить изменение таблиц.

**Ход работы**

* + - 1. Cоздание таблиц в СУБД

CREATE TABLE IF NOT EXISTS "uslugi" (

"id\_u" serial NOT NULL,

"name" varchar(50) NOT NULL,

"price" numeric(10,2) NOT NULL,

"srok\_vip" INTEGER NOT NULL,

"sr\_otkl" numeric(10,2) NOT NULL,

PRIMARY KEY ("id\_u")

);

CREATE TABLE IF NOT EXISTS "klienti" (

"id\_k"serial NOT NULL,

"name\_k" text,

"adress\_k" text,

"inn\_k" text UNIQUE,

"r\_s\_k" text,

"bik\_k" text,

"fio\_ryk\_k" text,

"fio\_kont\_k" text,

"tel\_kont\_k" text,

"e-mail\_k" text UNIQUE,

"password\_k" text,

"fio\_f" text,

"dare\_r\_f" date,

"seria\_nomer\_f" text UNIQUE,

"tel\_f" text,

"fiz\_l\_f" boolean,

PRIMARY KEY ("id\_k")

);

CREATE TABLE IF NOT EXISTS "zakaz" (

"id\_z" serial NOT NULL,

"date\_z" timestamp without time zone DEFAULT '0',

"status" text,

"vremy\_vip" bigint,

"price" numeric(10,2),

"id\_k" bigint,

ADD CONSTRAINT "zakaz\_fk5" FOREIGN KEY ("id\_k") REFERENCES "klienti"("id\_k"),

PRIMARY KEY ("id\_z")

);

CREATE TABLE IF NOT EXISTS "laboranti" (

"id\_l" serial NOT NULL,

"login" text NOT NULL,

"password" text NOT NULL,

"fio" text NOT NULL,

"date\_vhoda" timestamp without time zone DEFAULT '0',

PRIMARY KEY ("id\_l")

);

CREATE TABLE IF NOT EXISTS "okazannie\_uslugi" (

"id\_z" int,

"id\_u" int,

"date\_provedenia" timestamp without time zone,

"status\_vipolnenia" text,

"id\_l" int,

ADD CONSTRAINT "okazannie\_uslugi\_fk0" FOREIGN KEY ("id\_z") REFERENCES "zakaz"("id\_z"),

ADD CONSTRAINT "okazannie\_uslugi\_fk1" FOREIGN KEY ("id\_u") REFERENCES "uslugi"("id\_u"),

ADD CONSTRAINT "okazannie\_uslugi\_fk4" FOREIGN KEY ("id\_l") REFERENCES "laboranti"("id\_l")

);

CREATE TABLE IF NOT EXISTS "lab\_uslugi" (

"id\_l" int NOT NULL,

"id\_u" int,

ADD CONSTRAINT "lab\_uslugi\_fk0" FOREIGN KEY ("id\_l") REFERENCES "laboranti"("id\_l"),

ADD CONSTRAINT "lab\_uslugi\_fk1" FOREIGN KEY ("id\_u") REFERENCES "uslugi"("id\_u"),

PRIMARY KEY ("id\_l"));

CREATE TABLE IF NOT EXISTS "sotrudniki" (

"id\_с" serial NOT NULL,

"login" text NOT NULL,

"password\_s" text NOT NULL,

"fio" text NOT NULL

);

1. Изменение таблиц.

* Изменить тип поляname\_kтаблицы klienti с text на varhar(50).
* Сделать поле date\_vhoda таблицы laboranti обязательным к заполнению.
* Создать составной первичный ключ на таблицу okazannie\_uslugi по полям id\_z, id\_u.

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №10 «Создание генераторов, индексов, представлений»

Цель работы: сформировать у студентов практические знания по созданию генераторов, индексов, представлений.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

* + - 1. Изучить работу с генераторами, индексами и представлениями.

**Ход работы:**

**ПОСЛЕДОВАТЕЛЬНОСТИ**

~~~~~~~~~~~~~~~~~~

Создадим последовательность:

=>**createsequencesstartwith 2 cache 10;**

CREATE SEQUENCE

Ее можно использовать, например, для генерации уникального ключа:

=>**insert into people values (nextval('s'), 'Петров', false);**

INSERT 0 1

=>**select currval('s');**

currval

---------

2

(1 row)

|  |  |  |
| --- | --- | --- |
| currval(regclass) | bigint | Выдаёт значение заданной последовательности, которое было возвращено при последнем вызове функции nextval |

-----------------------------------------------------------------------

Поскольку при создании последовательности мы указали cache 10, сеансы будут кэшировать

диапазоны значений. Это приведет к тому, что два сеанса могут выдавать номера не по порядку:

| psql

| => select nextval('s');

| nextval

| ---------

| 12

| (1 row)

|

=>**select nextval('s');**

nextval

---------

3

(1 row)

-----------------------------------------------------------------------

| =>**select nextval('s');**

| nextval

| ---------

| 13

| (1 row)

|

=>**select nextval('s');**

nextval

---------

4

(1 row)

| => \q

---------------------------------------------------------------------

1. Создайте таблицу, в которой будет использоваться последовательность:

CREATE TABLE employees (

id SERIAL PRIMARY KEY,

nameVARCHAR(50),

departmentVARCHAR(50)

);

1. Создайте последовательность с помощью оператора CREATE SEQUENCE:

CREATE SEQUENCE employee\_id\_seqSTART WITH 1INCREMENT BY 1NO MAXVALUECACHE 1;

1. Свяжите последовательность с полем таблицы:

ALTER SEQUENCE employee\_id\_seqOWNED BY employees.id;

1. Вставьте данные в таблицу, и значения id будут автоматически генерироваться из последовательности:

INSERT INTO employees (name, department)VALUES ('John Doe', 'Sales'),('Jane Smith', 'Marketing');

1. Проверьте, что последовательность работает:

SELECT \* FROM employees;

**ИНДЕКСЫ**

~~~~~~~

Создадим большую таблицу:

=>create table bigtable(id serial, t text);

CREATE TABLE

=>insert into bigtable

=>select s.id, 'This is a line #' || s.id from generate\_series(1,1000000) as s(id);

INSERT 0 1000000

=>select count(\*) from bigtable;

count

---------

1000000

(1 row)

-----------------------------------------------------------------------

Сколько времени занимает поиск одного значения в такой таблице?

=>\timing on

Timing is on.

=>select \* from bigtable where id=42;

id | t

----+--------------------

42 | This is a line #42

(1 row)

Time: 115,043 ms

-----------------------------------------------------------------------

Теперь проиндексируем таблицу:

=>create unique index bigtable\_id on bigtable(id);

CREATE INDEX

Time: 411,549 ms

=>select \* from bigtable where id=42;

id | t

----+--------------------

42 | This is a line #42

(1 row)

Time: 2,753 ms

=>\timing off

Timing is off.

Индекс не является универсальным средством увеличения производительности. В общем случае

индекс может быть полезен, если из таблицы требуется выбрать существенно меньше данных,

чем в ней находится. Кроме того, надо учитывать накладные расходы на обновление индекса

при изменении таблицы.

ПРЕДСТАВЛЕНИЯ

~~~~~~~~~~~~~

Создадим представление для суммы дохода по городам за все время.

=>create view revenue\_by\_city as

=>select city, sum(amount) from revenue group by city;

CREATE VIEW

Запрос будет выполняться при обращении к представлению:

=>\timing on

Timing is on.

=>select \* from revenue\_by\_city;

city | sum

------+-------------------

SPB | 48 796.46 руб

MSK | 49 972 664.21 руб

(2 rows)

Time: 342,959 ms

-----------------------------------------------------------------------

Материализованное представление сохраняет результат запроса.

=>create materialized view revenue\_by\_city\_m as

=>select city, sum(amount) from revenue group by city;

SELECT 2

Time: 354,552 ms

=>select \* from revenue\_by\_city\_m;

city | sum

------+-------------------

SPB | 48 796.46 руб

MSK | 49 972 664.21 руб

(2 rows)

Time: 1,470 ms

=>\timing off

Timing is off.

-----------------------------------------------------------------------

Изменение в данных будет сразу же отражено в представлении:

=>insert into revenue values (1000000.00, current\_date, 'SPB');

INSERT 0 0

=>select \* from revenue\_by\_city;

city | sum

------+-------------------

SPB | 1 048 796.46 руб

MSK | 49 972 664.21 руб

(2 rows)

Но не в материализованном представлении.

=>select \* from revenue\_by\_city\_m;

city | sum

------+-------------------

SPB | 48 796.46 руб

MSK | 49 972 664.21 руб

(2 rows)

-----------------------------------------------------------------------

Изменения появятся только после обновления материализованного представления:

=>refresh materialized view revenue\_by\_city\_m;

REFRESH MATERIALIZED VIEW

=>select \* from revenue\_by\_city\_m;

city | sum

------+-------------------

SPB | 1 048 796.46 руб

MSK | 49 972 664.21 руб

(2 rows)

----------------------------------------------------------------------------------------------------------

Задание 1. Создание представления

Создайте таблицу "Сотрудники" со следующими полями:

* id (первичный ключ, serial)
* имя (varchar)
* фамилия (varchar)
* должность (varchar)
* зарплата (integer)

Заполните таблицу тестовыми данными.

Создайте представление "ВсеСотрудники" на основе таблицы "Сотрудники", включающее все поля:

CREATE VIEW ВсеСотрудники ASSELECT \* FROM "Сотрудники";

Выполните запрос к представлению:

SELECT \* FROM ВсеСотрудники;

Задание 2. Изменение представления

Создайте новое представление "СотрудникиИнженеры", включающее только сотрудников с должностью "Инженер":

CREATE VIEW "СотрудникиИнженеры" ASSELECT \* FROM "Сотрудники"WHERE "Должность" = 'Инженер';

Выполните запрос к представлению:

SELECT \* FROM "СотрудникиИнженеры";

Задание 3. Удаление представления

Удалите представление "ВсеСотрудники":

DROP VIEW "ВсеСотрудники";

Убедитесь, что представление удалено, выполнив запрос:

SELECT \* FROM "ВсеСотрудники";

Вы должны получить ошибку, так как представление больше не существует.

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №11 «Каскадные операции»

Цель работы: сформировать у студентов практические знания по работе с каскадными операциями для управления связями между таблицами.

Проверяемые результаты обучения:

* применять стандартные методы для защиты объектов базы данных;
* обеспечивать информационную безопасность на уровне базы данных;
* методы организации целостности данных;
* основные методы и средства защиты данных в базах данных;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

1. Научиться работать с каскадными операциями для управления связями между таблицами СУБД.

**Ход работы**

**Задание 1.** Создание таблиц.

Создайте таблицу "Категории" со следующими полями:

* id (первичный ключ, serial)
* название (varchar)

Создайте таблицу "Товары" со следующими полями:

* id (первичный ключ, serial)
* название (varchar)
* категория (integer, внешний ключ, ссылается на id в таблице "Категории")
* цена (integer)

**Задание 2.** Создание каскадных операций.

1. Создайте внешний ключ в таблице "Товары" с каскадным удалением:

ALTER TABLE "Товары"

ADD CONSTRAINT "fk\_Категория"

FOREIGN KEY ("Категория")

REFERENCES "Категории" (id)

ON DELETE CASCADE;

1. Создайте внешний ключ в таблице "Товары" с каскадным обновлением:

ALTER TABLE "Товары"

ADD CONSTRAINT "fk\_Категория"

FOREIGN KEY ("Категория")

REFERENCES "Категории" (id)

ON UPDATE CASCADE;

**Задание 3**. Тестирование каскадных операций.

1. Вставьте несколько категорий и товаров в таблицы.
2. Удалите категорию, которая имеет связанные товары:

DELETE FROM "Категории" WHERE id = <id\_категории>;

Проверьте, что товары, связанные с удаленной категорией, также были удалены.

Обновите id категории, которая имеет связанные товары:

UPDATE "Категории" SET id = <новый\_id> WHERE id = <старый\_id>;

Проверьте, что id категории в таблице "Товары" также было обновлено.

**Задание 4.** Удаление каскадных операций.

Удалите внешний ключ с каскадным удалением:

ALTER TABLE "Товары"DROP CONSTRAINT "fk\_Категория";

Убедитесь, что каскадное удаление больше не работает, выполнив запрос удаления категории.

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №12 «Импорт, экспорт данных»

Цель работы: научиться импортировать и экспортировать данные в PostgreSQL с использованием утилит и команд.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* выполнять стандартные процедуры резервного копирования и мониторинга выполнения этой процедуры;
* выполнять процедуру восстановления базы данных и вести мониторинг выполнения этой процедуры;
* основные методы и средства защиты данных в базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Выполнить экспорт и импорт данных в таблицы БД.

**Ход работы:**

1. Создать БД с названием test.
2. Подключиться к созданной БД.
3. Создать таблицыtovari и category.

сreate table category(

id\_c int primary key,

name varchar (50) not null);

сreate table tovari (

id\_tint primary key,

name varchar (50) not null,

id\_cint,

foreign key(id\_c) references category(id\_c)

);

1. Заполнить таблицы. Внести не менее 5 записей.
2. Задание 1. Экспорт данных
3. Используя команду pg\_dump, экспортируйте данные из таблицы «tovari» в файл формата SQL

pg\_dump -h <хост> -U <пользователь> -d <база\_данных> -t tovari>tovari.sql

1. Проверьте содержимое экспортированного файла «tovari.sql»
2. Задание 2. Импорт данных

Создайте новую базу данных или очистите существующую базу данных для импорта данных.

Используя команду psql, импортируйте данные из файла "tovari.sql" в базу данных:

psql -h <хост> -U <пользователь> -d <база\_данных> -f tovari.sql

1. Проверьте, что данные успешно импортированы, выполните запрос к таблице "Товары" в базе данных.
2. Задание 3. Использование форматов данных

Используйте формат CSV для экспорта данных из таблицы «category»:

COPY categoryTO '/путь/category.csv' DELIMITER ',' CSV HEADER;

Проверьте содержимое файла «category.csv».

Импортируйте данные из файла «category.csv» обратно в базу данных, используя команду COPY.

1. Задание 4. Расширенный импорт и экспорт

Используйте утилиту pg\_restore для импорта данных из дампа базы данных:

pg\_restore -h <хост> -U <пользователь> -d <база\_данных><дамп\_базы\_данных>

Используйте параметры --data-only и --table для экспорта только данных определенной таблицы.

Задание 1. Экспорт базы данных

Подключитесь к базе данных PostgreSQL, которую необходимо экспортировать.

Используйте утилиту pg\_dump для создания дампа всей базы данных:

bash

pg\_dump -h <хост> -U <пользователь> -d <база\_данных>>база\_данных.sql

Создать резервую копию БДtest.

Задание 1. Экспорт базы данных

Подключитесь к базе данных PostgreSQL, которую необходимо экспортировать.

Используйте утилиту pg\_dump для создания дампа всей базы данных:

pg\_dump -h <хост> -U <пользователь> -d <база\_данных>>база\_данных.sql

pg\_dump -h localhost -U postgres-d test>test.sql

Задание 2. Импорт базы данных

Создайте новую базу данных в PostgreSQL, куда будет импортирован дамп:

CREATEDATABASEcopy\_test;

psql -hlocalhost -Upostgres -dcopy\_test -ftest.sql

Этот запрос импортирует все объекты и данные из файла база\_данных.sql в новую базу данных.

Проверьте, что импорт прошел успешно, подключившись к новой базе данных и выполнив запросы к различным таблицам.

Задание 3. Экспорт и импорт с использованием форматов

Экспортируйте базу данных в формате custom (архивный формат):

pg\_dump -Fc -h<хост> -U<пользователь> -d<база\_данных>>база\_данных.custom

Импортируйте базу данных из архивного формата:

pg\_restore -h<хост> -U<пользователь> -d<новая\_база\_данных>база\_данных.custom

Формат custom позволяет сохранять дополнительную информацию, такую как индексы, триггеры и т.д.

**Критерииоценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №13. Оконные функции в SQL.

Цель работы: Научиться использовать оконные функции в PostgreSQL для выполнения аналитических расчетов.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

**Ход работы:**

Задание 1. Создание таблицы и заполнение данными

* Создайте новую БД с именем test\_f.
* Подключитесь к созданной базе.
* Создайте таблицу "prodaji" со следующими полями:
* id (первичный ключ, serial)
* tovar (varchar (50))
* category (varchar(20))
* price (decimal (7,2))
* kol (integer)
* date\_p (date)
* Заполните таблицу тестовыми данными о продажах.

Задание 2. Использование оконной функции ROW\_NUMBER()

Выведите список продаж с порядковым номером для каждой строки в пределах категории, отсортированной по дате:

SELECT

ROW\_NUMBER() OVER (PARTITION BY categoryORDER BY date\_p) AS

Номер,

Товар,

Категория,

Цена,

Количество,

Дата

FROMprodaji

ORDER BY category,date\_p;

Задание 3. Использование оконной функции RANK()

Выведите список продаж с рангом для каждой строки в пределах категории, отсортированной по дате. Ранг учитывает повторяющиеся значения:

SELECT

RANK() OVER (PARTITION BY categoryORDER BY date\_p AS Ранг,

Товар,

Категория,

Цена,

Количество,

Дата

FROM prodaji

ORDER BY category, date\_p;

Задание 4. Использование оконной функции DENSE\_RANK()

Выведите список продаж с плотным рангом для каждой строки в пределах категории, отсортированной по дате. Плотный ранг пропускает повторяющиеся значения:

SELECT

DENSE\_RANK() OVER (PARTITIONBYcategoryORDERBYdate\_p) AS "Плотныйранг",

Товар,

Категория,

Цена,

Количество,

Дата

FROM prodaji

ORDER BY category, date\_p;

Задание 5. Использование оконной функции SUM()

Выведите список продаж с накопительной суммой количества для каждой строки в пределах категории, отсортированной по дате:

SELECT

Товар,

Категория,

Цена,

Количество,

SUM(kol) OVER (PARTITIONBYcategoryORDERBYdate\_p) AS "Накопительнаясумма"

FROM prodaji

ORDER BY category, date\_p;

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №14 «Хранимые функции»

Цель работы: сформировать у студентов практические знания по работе с хранимыми процедурами.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* создание хранимых процедур и триггеров на базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Изучить работу с хранимыми процедурами.

**Ход работы:**

**ХРАНИМЫЕ ФУНКЦИИ**

~~~~~~~~~~~~~~~~

Простой пример функции на SQL:

=>create function random\_text(len integer) returns text as $$

=> select string\_agg(chr(trunc(65+random()\*26)::integer),'') from generate\_series(1,$1);

=>$$ languagesql;

CREATE FUNCTION

Теперь функцию можно использовать в запросах:

=>select random\_text(10);

random\_text

-------------

FVDBKHPHBU

(1 row)

------------------------------------------------------------------------

Задание 1. Создание таблицы и заполнение данными

* Создайте новую БД.
* Подключитесь к созданной БД.
* Создайте таблицу sotrudniki со следующими полями:
* id (первичный ключ, serial)
* ima (varchar(20))
* familia (varchar(30))
* doljnost (varchar(20))
* salary (integer).
* Заполните таблицу тестовыми данными о сотрудниках.

Задание 2. Создание хранимой процедуры для вставки данных в таблицу sotrudniki.

CREATE OR REPLACE PROCEDURE insert\_sotr(

p\_imaVARCHAR(20),

p\_familiaVARCHAR(30),

p\_doljnostVARCHAR(20),

p\_salaryINTEGER

)

AS $$

BEGIN

INSERT INTO sotrudniki(ima, familia, doljnost, salary )

VALUES (p\_ima, p\_familia, p\_doljnost, p\_salary);

END;

$$

LANGUAGE plpgsql;

Вызовите процедуру для добавления сотрудника

CALL insert\_sotr('Иван', 'Иванов', 'Менеджер', 5000);

## Задание 3. Создание хранимой процедуры для обновления данных

1. Создайте хранимую процедуру для обновления данных сотрудника по id:

CREATE OR REPLACE PROCEDURE update\_sotr(

p\_id INTEGER,

p\_imaVARCHAR(20),

p\_familiaVARCHAR(30),

p\_doljnostVARCHAR(20),

p\_salaryINTEGER)

AS $$

BEGIN

UPDATE sotrudniki

SET ima = p\_ima,

familia = p\_familia,

doljnost= p\_doljnost,

salary = p\_salary

WHERE id = p\_id;

END;

$$

LANGUAGE plpgsql;

1. Вызовите процедуру для обновления данных сотрудника:

CALL update\_sotr(1, 'Иван', 'Иванов', 'Директор', 10000);

## Задание 4. Создание хранимой процедуры с возвратом результата

1. Создайте хранимую процедуру, возвращающую список сотрудников по должности:

CREATE OR REPLACE PROCEDURE select\_sotr\_dolj(

p\_doljnostVARCHAR(20),

OUT p\_ima VARCHAR(20),

OUTp\_familia VARCHAR(30),

OUT p\_salaryINTEGER

)

AS $$

BEGIN

FOR p\_id, p\_ima, p\_familia, p\_salary IN

SELECT id, ima, familia, salary

FROMsotrudniki

WHERE doljnost = select\_sotr\_dolj.p\_doljnost

LOOP

RETURN;

END LOOP;

END;

$$

LANGUAGE plpgsql;

1. Вызовите процедуру для получения списка сотрудников по должности:

CALL select\_sotr\_dolj('Менеджер');

**Критерииоценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №15 «Триггеры»

Цель работы: Научиться создавать и использовать триггеры в PostgreSQL для автоматизации обработки данных.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* создавать хранимые процедуры и триггеры на базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Изучить работу с триггерами.

**Ход работы:**

ТРИГГЕРЫ

~~~~~~~~

Продемонстрируем триггер для сохранения времени последнего изменения строк.

Добавим поле в таблицу people:

=>alter table people add column last\_update timestamp;

ALTER TABLE

Создадим функцию, которая будут использоваться в триггере:

=>create function set\_last\_update() returns trigger as $$

=>begin

=>new.last\_update := current\_timestamp;

=> return new;

=>end;

=>$$ language plpgsql;

CREATE FUNCTION

-----------------------------------------------------------------------

И собственно триггер:

=>create trigger people\_last\_update

=> before update or insert on people

=> for each row

=> execute procedure set\_last\_update();

CREATE TRIGGER

-----------------------------------------------------------------------

Проверим работоспособность вставки:

=>insert into people(id, name) values (3, 'Сидоров');

INSERT 0 1

=>select \* from people;

id | name | employed | last\_update

----+---------+----------+----------------------------

2 | Петров | f |

3 | Сидоров | | 2016-06-16 17:23:14.703131

(2 rows)

-----------------------------------------------------------------------

И обновления:

=>update people set employed = true where id = 2;

UPDATE 1

=>select \* from people;

id | name | employed | last\_update

----+---------+----------+----------------------------

3 | Сидоров | | 2016-06-16 17:23:14.703131

2 | Петров | t | 2016-06-16 17:23:14.711634

(2 rows)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание 1. Создание таблицы и заполнение данными

* Создайте БД с именем test\_t.
* Подключитесь к созданной БД.
* Создайте таблицу tovari со следующими полями:
* id (первичный ключ, serial)
* name (varchar(50))
* price (integer)
* kol (integer)
* Заполните таблицу тестовыми данными о товарах.

Задание 2. Создание триггера для проверки цены.

Создайте функцию, которая будет проверять цену товара при вставке или обновлении:

CREATE OR REPLACE FUNCTION proverit\_price()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.price<= 0 THEN

RAISE EXCEPTION 'Цена товара должна быть больше 0';

END IF;

RETURN NEW;

END;

$$

LANGUAGE plpgsql;

1. Создайте триггер, который будет вызывать функцию "ПроверитьЦену" при вставке или обновлении данных в таблице "Товары":

CREATE TRIGGER oroverka\_price

BEFORE INSERT OR UPDATE ON tovari

FOR EACH ROW

EXECUTE FUNCTION proverit\_price();

## Задание 3. Создание триггера для ведения истории изменений

1. Создайте таблицу для хранения истории изменений товаров:

CREATE TABLE history\_tovarov (

id SERIAL PRIMARY KEY,

id\_tINTEGER,

name VARCHAR,

price INTEGER,

kol INTEGER,

deistvieVARCHAR(10),

date\_h TIMESTAMP DEFAULT CURRENT\_TIMESTAMP);

1. Создайте функцию, которая будет записывать изменения в таблицу истории:

CREATE OR REPLACE FUNCTION insert\_history()

RETURNS TRIGGER AS $$

BEGIN

IF TG\_OP = 'DELETE' THEN

INSERT INTO history\_tovarov (id\_t, name, price,kol, deistvie)

VALUES (OLD.id, OLD.name, OLD.price, OLD.kol, 'Удаление');

ELSIF TG\_OP = 'UPDATE' THEN

INSERT INTO history\_tovarov(id\_t, name, price, kol, deistvie)

VALUES (NEW.id, NEW. Name, NEW.price, NEW.kol, 'Обновление');

ELSIF TG\_OP = 'INSERT' THEN

INSERT INTO history\_tovarov(id\_t, name, price, kol, deistvie)

VALUES (NEW.id, NEW. name, NEW.price, NEW.kol, 'Вставка');

END IF;

RETURN NULL;

END;

$$

LANGUAGE plpgsql;

1. Создайте триггер, который будет вызывать функцию insert\_history() при вставке, обновлении или удалении данных в таблице "tovari":

CREATE TRIGGER "history\_tovarov"

AFTER INSERT OR UPDATE OR DELETE ON tovari

FOR EACH ROW

EXECUTE FUNCTION insert\_history();

1. Выполните операции вставки, обновления и удаления товаров. Проверьте, что изменения записываются в таблицу истории.

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №16. Транзакции.

Цель работы: научиться использовать транзакции в PostgreSQL для обеспечения целостности данных.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* применять стандартные методы для защиты объектов базы данных;
* обеспечивать информационную безопасность на уровне базы данных;
* методы организации целостности данных;
* основные методы и средства защиты данных в базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Изучить работу с транзакциями.

**Ход работы:**

## Задание 1. Создание таблицы и заполнение данными

1. Создайте новую БД.
2. Подключитесь к созданной БД.
3. Создайте таблицу "scheta" со следующими полями:
   * id (первичный ключ, serial)
   * ima (varchar)
   * balance (decimal)
4. Заполните таблицу тестовыми данными о счетах.

## Задание 2. Выполнение транзакции для перевода средств

1. Начните транзакцию:

BEGIN;

1. Выполните операции по переводу средств с одного счета на другой:

UPDATE scheta SET balance = balance - 100.00 WHERE ima = 'Борис';

UPDATE scheta SET balance = balance + 100.00 WHERE ima = 'Олег';

1. Завершите транзакцию, зафиксировав изменения:

COMMIT;

## Задание 3. Отмена транзакции при ошибке

1. Начните новую транзакцию:

BEGIN;

1. Выполните операции по переводу средств, но с ошибкой:

UPDATE schetaSETbalance = balance - 200.00 WHERE "Имя" = 'Борис';

UPDATE schetaSETbalance = balance + 100.00 WHERE "Имя" = 'Олег';

1. Отмените транзакцию, так как баланс Бориса стал отрицательным:

ROLLBACK;

## Задание 4. Использование уровней изоляции транзакций

1. Установите уровень изоляции транзакций "Повторяемое чтение":

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

BEGIN;

1. Выполните запрос, который будет читать данные из таблицы "Счета":

SELECT \* FROM scheta;

1. В другом сеансе выполните операцию обновления данных в таблице "Счета":

UPDATE scheta SET balance = balance + 50.00 WHERE ima = 'Борис';

1. Вернитесь в первый сеанс и выполните тот же запрос чтения. Убедитесь, что данные не изменились, несмотря на обновление во втором сеансе.
2. Завершите транзакцию в первом сеансе:

COMMIT;

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №17. События.

Цель работы: научиться создавать и использовать события в PostgreSQL для автоматизации регулярных задач.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* применять стандартные методы для защиты объектов базы данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Изучить работу с событиями.

**Ход работы:**

## Задание 1. Создание таблицы и заполнение данными

1. Создайте новую БД.
2. Подключитесь к новой БД.
3. Создайте таблицу orders со следующими полями:
   * id (первичный ключ, serial)
   * item (varchar(50))
   * kol (integer)
   * date\_o (date)
4. Заполните таблицу тестовыми данными о заказах.

## Задание 2. Создание события для удаления старых заказов.

1. Создайте функцию, которая будет удалять заказы, созданные более 30 дней назад:

CREATE OR REPLACE FUNCTION del\_old\_orders()

RETURNS VOID AS $$

BEGIN

DELETE FROM orders

WHERE date\_o< (CURRENT\_DATE - INTERVAL 30 DAY);

END;

$$

LANGUAGE plpgsql;

1. Создайте событие, которое будет вызывать функцию del\_old\_orders() каждый день в 00:00:

CREATE EVENT SCHEDULE every 1 day

STARTS '2024-05-27 00:00:00'

ON EVENT del\_old\_orders;

1. Убедитесь, что событие создано, выполнив запрос:

SELECT \* FROM pg\_events;

## Задание 3. Создание события для отправки отчета по заказам.

1. Создайте функцию, которая будет формировать отчет по заказам за последние 7 дней:

CREATE OR REPLACE FUNCTION otchet\_orders()

RETURNS TEXT AS $$

DECLARE

report TEXT;

BEGIN

report := 'Отчет по заказам за последние 7 дней:' || E'\n\n';

SELECT

string\_agg(

'Товар: ' || item || ', Количество: ' || kol || ', Дата: ' || date\_o,

E'\n'

)

INTO report

FROM orders

WHERE date\_o>= (CURRENT\_DATE - INTERVAL '7 DAY');

RETURN report;

END;

$$

LANGUAGE plpgsql;

1. Создайте событие, которое будет вызывать функцию otchet\_orders()каждое воскресенье в 21:00 и отправлять отчет на электронную почту:

**SELECT cron.schedule('0 21 \* \* 0', $$**

**SELECT pg\_send\_email('stenly@yandex.ru', 'report\_o', otchet\_orders();**

**$$**

**);**

1. Убедитесь, что событие создано, выполнив запрос:

SELECT \* FROM pg\_events;

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №18 «Мониторинг работы сервера (сбор статистики, журнал сообщений»

Цель работы: сформировать у студентов практические знания по мониторингу работы сервера.

Проверяемые результаты обучения:

* приемы администрирования, сопровождения и отказоустойчивости базы данных.

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Ознакомиться с процессами мониторинга сервера.

**Ход работы:**

Сбором статистики занимается фоновый процесс statscollector.

Количеством собираемой информации управляют несколько

параметров сервера, так как чем больше информации собирается, тем

больше и накладные расходы

Сбором статистики занимается фоновый процесс statscollector. Количеством собираемой информации управляют несколько параметров сервера, так как чем больше информации собирается, тем больше и накладные расходы

**Статистикавнутрибазы**

=> CREATE DATABASE admin\_monitoring;

CREATE DATABASE

=> \c admin\_monitoring

You are now connected to database "admin\_monitoring" as user "student".

Вначале включим сбор статистики ввода-вывода:

=> ALTER SYSTEM SET track\_io\_timing=on;

ALTER SYSTEM

=> SELECT pg\_reload\_conf();

pg\_reload\_conf

----------------

t

(1 row)

Смотреть на активности сервера имеет смысл, когда какие-то активности на самом деле есть. Чтобы сымитировать нагрузку, воспользуемся pgbench — штатной утилитой для запуска эталонных тестов.

Сначала утилита создает набор таблиц и заполняет их данными.

**student$ pgbench -i admin\_monitoring**

dropping old tables...

NOTICE: table "pgbench\_accounts" does not exist, skipping

NOTICE: table "pgbench\_branches" does not exist, skipping

NOTICE: table "pgbench\_history" does not exist, skipping

NOTICE: table "pgbench\_tellers" does not exist, skipping

creating tables...

generating data (client-side)...

100000 of 100000 tuples (100%) done (elapsed 0.14 s, remaining 0.00 s)

vacuuming...

creating primary keys...

done in 0.49 s (drop tables 0.00 s, create tables 0.02 s, client-side generate 0.25 s, vacuum 0.08 s, primary keys 0.12 s).

Сбросим все накопленные ранее статистики:

=>**SELECT pg\_stat\_reset();**

pg\_stat\_reset

---------------

(1 row)

=>**SELECT pg\_stat\_reset\_shared('bgwriter');**

pg\_stat\_reset\_shared

----------------------

(1 row)

Теперь запускаем тест TPC-B на несколько секунд:

**student$ pgbench -T 10 admin\_monitoring**

starting vacuum...end.

transaction type: <builtin: TPC-B (sort of)>

scaling factor: 1

query mode: simple

number of clients: 1

number of threads: 1

duration: 10 s

number of transactions actually processed: 1548

latency average = 6.462 ms

tps = 154.756948 (including connections establishing)

tps = 154.948442 (excluding connections establishing)

Теперь мы можем посмотреть статистику обращения к таблицам в терминах строк:

=> SELECT \*

FROM pg\_stat\_all\_tables

WHERE relid = 'pgbench\_accounts'::regclass \gx

-[ RECORD 1 ]-------+-----------------

relid | 16546

schemaname | public

relname | pgbench\_accounts

seq\_scan | 0

seq\_tup\_read | 0

idx\_scan | 3096

idx\_tup\_fetch | 3096

n\_tup\_ins | 0

n\_tup\_upd | 1548

n\_tup\_del | 0

n\_tup\_hot\_upd | 551

n\_live\_tup | 0

n\_dead\_tup | 1382

n\_mod\_since\_analyze | 1548

n\_ins\_since\_vacuum | 0

last\_vacuum |

last\_autovacuum |

last\_analyze |

last\_autoanalyze |

vacuum\_count | 0

autovacuum\_count | 0

analyze\_count | 0

autoanalyze\_count | 0

И в терминах страниц:

=> SELECT \*

FROM pg\_statio\_all\_tables

WHERE relid = 'pgbench\_accounts'::regclass \gx

-[ RECORD 1 ]---+-----------------

relid | 16546

schemaname | public

relname | pgbench\_accounts

heap\_blks\_read | 16

heap\_blks\_hit | 8686

idx\_blks\_read | 273

idx\_blks\_hit | 7925

toast\_blks\_read |

toast\_blks\_hit |

tidx\_blks\_read |

tidx\_blks\_hit |

Существуют аналогичные представления для индексов:

**=>SELECT \***

**FROMpg\_stat\_all\_indexes**

**WHERErelid = 'pgbench\_accounts'::regclass\gx**

**=>SELECT \***

**FROMpg\_statio\_all\_indexes**

**WHERErelid = 'pgbench\_accounts'::regclass\gx**

Можно посмотреть глобальную статистику по всей базе данных:

**=>SELECT \***

**FROMpg\_stat\_database**

**WHEREdatname = 'admin\_monitoring'\gx**

# **Текущие активности**

Воспроизведем сценарий, в котором один процесс блокирует выполнение другого, и попробуем разобраться в ситуации с помощью системных представлений.

Создадим таблицу с одной строкой:

**=>CREATE TABLEt(n integer);**

CREATE TABLE

**=>INSERT INTO t VALUES(42);**

INSERT 0 1

Запустим два сеанса, один из которых изменяет таблицу и ничего не делает:

**student$ psql -d admin\_monitoring**

**=>BEGIN;**

BEGIN

**=>UPDATE t SET n = n + 1;**

UPDATE 1

А второй пытается изменить ту же строку и блокируется:

**student$ psql -d admin\_monitoring**

**=>UPDATE t SET n = n + 2;**

Посмотрим информацию об обслуживающих процессах:

**=>SELECTpid, query, state, wait\_event, wait\_event\_type, pg\_blocking\_pids(pid)**

**FROMpg\_stat\_activity**

**WHEREbackend\_type = 'client backend'\gx**

-[ RECORD 1 ]----+-----------------------------------------------------------------------------

pid | 16666

query | UPDATE t SET n = n + 1;

state | idle in transaction

wait\_event | ClientRead

wait\_event\_type | Client

pg\_blocking\_pids | {}

-[ RECORD 2 ]----+-----------------------------------------------------------------------------

pid | 15896

query | SELECT pid, query, state, wait\_event, wait\_event\_type, pg\_blocking\_pids(pid)+

| FROM pg\_stat\_activity +

| WHERE backend\_type = 'client backend'

state | active

wait\_event |

wait\_event\_type |

pg\_blocking\_pids | {}

-[ RECORD 3 ]----+-----------------------------------------------------------------------------

pid | 16746

query | UPDATE t SET n = n + 2;

state | active

wait\_event | transactionid

wait\_event\_type | Lock

pg\_blocking\_pids | {16666}

А мы покажем, как завершить блокирующий сеанс вручную. Сначала узнаем номер заблокированного процесса при помощи функции pg\_blocking\_pids:

**=>SELECTpidASblocked\_pid**

**FROMpg\_stat\_activity**

**WHEREbackend\_type = 'client backend'**

**ANDcardinality(pg\_blocking\_pids(pid)) >0;**

blocked\_pid

-------------

16746

(1 row)

Блокирующий процесс можно вычислить и без функции pg\_blocking\_pids, используя запросы к таблице блокировок. Запрос покажет две строки: одна транзакция получила блокировку (granted), другая — нет и ожидает.

**=>SELECTlocktype, transactionid, pid, mode, granted**

**FROMpg\_locks**

**WHEREtransactionidIN (**

**SELECTtransactionidFROMpg\_locksWHEREpid = 16746AND NOT granted**

**);**

locktype | transactionid| pid | mode | granted

---------------+---------------+-------+---------------+---------

transactionid | 2130 | 16746 | ShareLock | f

transactionid | 2130 | 16666 | ExclusiveLock | t

(2 rows)

В общем случае нужно аккуратно учитывать тип блокировки.

Выполнение запроса можно прервать функцией pg\_cancel\_backend. В нашем случае транзакция простаивает, так что просто прерываем сеанс, вызвав pg\_terminate\_backend:

**=>SELECTpg\_terminate\_backend(b.pid)**

**FROMunnest(pg\_blocking\_pids(16746)) ASb(pid);**

pg\_terminate\_backend

----------------------

t

(1 row)

Функция unnest нужна, поскольку pg\_blocking\_pids возвращает массив идентификаторов процессов, блокирующих искомый обслуживающий процесс. В нашем примере блокирующий процесс один, но в общем случае их может быть несколько.

Подробнее о блокировках рассказывается в курсе DBA2.

Проверим состояние обслуживающих процессов.

**=>SELECTpid, query, state, wait\_event, wait\_event\_type**

**FROMpg\_stat\_activity**

**WHEREbackend\_type = 'client backend'\gx**

-[ RECORD 1 ]---+------------------------------------------------------

pid | 15896

query | SELECT pid, query, state, wait\_event, wait\_event\_type+

| FROM pg\_stat\_activity +

| WHERE backend\_type = 'client backend'

state | active

wait\_event |

wait\_event\_type |

-[ RECORD 2 ]---+------------------------------------------------------

pid | 16746

query | UPDATE t SET n = n + 2;

state | idle

wait\_event | ClientRead

wait\_event\_type | Client

Осталось только два, причем заблокированный успешно завершил транзакцию.

Представление pg\_stat\_activity показывает информацию не только про обслуживающие процессы, но и про служебные фоновые процессы экземпляра:

**=>SELECTpid, backend\_type, backend\_start, state**

**FROMpg\_stat\_activity;**

pid |backend\_type | backend\_start | state

-------+------------------------------+-------------------------------+--------

10039 | logical replication launcher | 2024-01-16 10:29:57.34861+03 |

10037 | autovacuum launcher | 2024-01-16 10:29:57.349716+03 |

17059 | autovacuum worker | 2024-01-16 10:31:07.221673+03 |

15896 | client backend | 2024-01-16 10:30:52.0689+03 | active

16746 | client backend | 2024-01-16 10:31:05.596307+03 | idle

10035 | background writer | 2024-01-16 10:29:57.351916+03 |

10034 | checkpointer | 2024-01-16 10:29:57.352908+03 |

10036 | walwriter | 2024-01-16 10:29:57.351165+03 |

(8 rows)

Сравним с тем, что показывает операционная система:

**student$ sudohead -n 1 /var/lib/postgresql/13/main/postmaster.pid**

10032

**student$ sudops -o pid,command --ppid 10032**

PID COMMAND

10034 postgres: 13/main: checkpointer

10035 postgres: 13/main: background writer

10036 postgres: 13/main: walwriter

10037 postgres: 13/main: autovacuum launcher

10038 postgres: 13/main: stats collector

10039 postgres: 13/main: logical replication launcher

15896 postgres: 13/main: student admin\_monitoring [local] idle

16746 postgres: 13/main: student admin\_monitoring [local] idle

Можно заметить, что в pg\_stat\_activity не попадает процесс statscollector.

**Критерииоценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №19 «Резервное копирование и восстановление бд»

Цель работы: Научиться создавать и восстанавливать резервные копии базы данных PostgreSQL.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* выполнение стандартных процедур резервного копирования и мониторинга выполнения этой процедуры;
* выполнение процедуры восстановления базы данных и вести мониторинг выполнения этой процедуры;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Изучить операции резервного копирования и восстановления из резервной копии базы.

**Ход работы:**

Шаг 1: Создание резервной копии базы данных

1. Создание резервной копии с использованием pg\_dump:

**pg\_dump -U username -d dbname>backup.sql**

1. Создание сжатой резервной копии с использованием pg\_dump и gzip:

**pg\_dump -U username -d dbname | gzip> backup.sql.gz**

Шаг 2: Восстановление базы данных из резервной копии

1. Восстановление базы данных из резервной копии без сжатия:

**psql -U username -d dbname<backup.sql**

1. Восстановление базы данных из сжатой резервной копии:

**gunzip -c backup.sql.gz | psql -U username -d dbname**

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №20. Управление доступом. Роли и атрибуты.

Цель работы: научитьсяуправлять доступом. Создавать роли.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* обеспечивать информационную безопасность на уровне базы данных;
* способы контроля доступа к данным и управления привилегиями;
* основные методы и средства защиты данных в базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Завести роль creator без права входа в систему, но с правом создания баз данных и ролей.

2. Завести пользователя weak с правом входа в систему.

3. Убедиться, что weak не может создать базу данных.

4. Включить пользователя weak в группу creator.

5. Создать базу данных DB9 под пользователем weak.

6. Удалить базу данных и роли.

**Ходработы:**

=>create role creator with createdbcreaterole;

CREATE ROLE

Роль weak

~~~~~~~~~

=>create role weak with login;

CREATE ROLE

Проверка возможности создания БД

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

=>\c - weak

You are now connected to database "postgres" as user "weak".

=>create database db;

ERROR: permission denied to create database

Включениевгруппу

~~~~~~~~~~~~~~~~~~

=>\c - postgres

You are now connected to database "postgres" as user "postgres".

=>grant creator to weak;

GRANT ROLE

СозданиеБД

~~~~~~~~~~~

=> \c - weak

You are now connected to database "postgres" as user "weak".

=>set role creator;

SET

=>create database db9;

CREATE DATABASE

Удаление базы и ролей

~~~~~~~~~~~~~~~~~~~~~

=>\c - postgres

You are now connected to database "postgres" as user "postgres".

=>drop database db9;

DROP DATABASE

=>drop role weak;

DROP ROLE

=>drop role creator;

DROPROLE

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №21. Управление доступом. Привилегии.

Цель работы: научиться управлять доступом. Определять привилегии пользователям.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* обеспечивать информационную безопасность на уровне базы данных;
* способы контроля доступа к данным и управления привилегиями;
* основные методы и средства защиты данных в базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

1. Создать базу данных DB11 и две роли: writer и reader.

2. Отозвать у роли public все привилегии на схему public, выдать эти привилегии роли writer и только usage — роли reader.

3. Настроить привилегии по умолчанию так, чтобы роль reader получала доступ на чтение к таблицам, принадлежащим writer в схеме public.

4. Создать пользователя w1 в группе writer и r1 в группе reader.

5. Под пользователем writer создать таблицу.

6. Убедиться, что r1 имеет доступ только на чтение, а w1 имеет полный доступ к таблице, включая удаление.

**Ходработы:**

База данных и роли

~~~~~~~~~~~~~~~~~~

=>createdatabasedb11;

CREATE DATABASE

=>create user writer;

CREATE ROLE

=>create user reader;

CREATE ROLE

Привилегии

~~~~~~~~~~

=>\c db11

You are now connected to database "db11" as user "postgres".

=>revoke all on schema public from public;

REVOKE

=>grant all on schema public to writer;

GRANT

=>grant usage on schema public to reader;

GRANT

Привилегиипоумолчанию

~~~~~~~~~~~~~~~~~~~~~~~

=>alter default privileges for role writer in schema public

=>grant select on tables to reader;

ALTER DEFAULT PRIVILEGES

Пользователи

~~~~~~~~~~~~

=>create user w1 in role writer;

CREATE ROLE

=>create user r1 in role reader;

CREATE ROLE

Таблица

~~~~~~~

=>\c - writer

You are now connected to database "db11" as user "writer".

=>create table t(n numeric);

CREATE TABLE

Проверка

~~~~~~~~

=>\c - w1

You are now connected to database "db11" as user "w1".

=>insert into t values (42);

INSERT 0 1

=>\c - r1

You are now connected to database "db11" as user "r1".

=>select \* from t;

=>update t set n=n+1;

ERROR: permission denied for relation t

=>\c - w1

You are now connected to database "db11" as user "w1".

=>drop table t;

DROP TABLE

=>\q

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №22. Политика защиты строк.

Цель работы: научиться настраивать политики строк.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных;
* применять стандартные методы для защиты объектов базы данных;
* обеспечивать информационную безопасность на уровне базы данных;
* способы контроля доступа к данным и управления привилегиями;
* основные методы и средства защиты данных в базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

1.Настроить политику защиты строк.

**Ходработы:**

# Пример политики защиты строк

**=>CREATE DATABASEaccess\_rls;**

CREATE DATABASE

**=>\caccess\_rls**

You are now connected to database "access\_rls" as user "student".

Алиса и Боб работают в разных отделах одной компании.

**student=# CREATE ROLEaliceLOGIN;**

CREATE ROLE

**student=# CREATE ROLE bob LOGIN;**

CREATE ROLE

**student=# CREATE TABLEusers\_depts(**

**login text,**

**department text**

**);**

CREATE TABLE

**student=# INSERT INTOusers\_deptsVALUES ('alice','PR'), ('bob','Sales');**

INSERT 0 2

Они обращаются к одной таблице, содержащей информацию обо всех отделах. При этом и Алиса, и Боб должны видеть данные только своего отдела.

**student=# CREATE TABLErevenue(**

**department text,**

**amount numeric(10,2)**

**);**

CREATE TABLE

**student=# INSERT INTO revenue SELECT'PR', -random()\* 100.00FROMgenerate\_series(1,100000);**

INSERT 0 100000

**student=# INSERT INTO revenue SELECT'Sales', random()\*1000.00FROMgenerate\_series(1,10000);**

INSERT 0 10000

Определим соответствующую политику и включим ее:

**student=# CREATE POLICY departments ON revenue**

**USING (department = (SELECT department FROMusers\_deptsWHERE login = current\_user));**

CREATE POLICY

**student=# ALTER TABLE revenue ENABLE ROW LEVEL SECURITY;**

ALTER TABLE

И нужно выдать Алисе и Бобу привилегии:

**student=# GRANT SELECT ONusers\_depts, revenue TOalice, bob;**

GRANT

Суперпользователь (он же владелец в данном случае) видит все строки независимо от политики:

**student=# SELECT department, sum(amount) FROM revenue GROUP BY department;**

department | sum

------------+-------------

PR | -5011384.62

Sales | 5014871.94

(2 rows)

А что увидят Алиса и Боб?

**student=# \c - alice**

You are now connected to database "access\_rls" as user "alice".

**alice=>SELECT department, sum(amount) FROM revenue GROUP BY department;**

department | sum

------------+-------------

PR | -5011384.62

(1 row)

**=>\caccess\_rls bob**

You are now connected to database "access\_rls" as user "bob".

**bob=>SELECT department, sum(amount) FROM revenue GROUP BY department;**

department | sum

------------+------------

Sales | 5014871.94

(1 row)

# Несколько политик

Разрешим теперь Бобу добавлять строки в таблицу, но только для своего отдела и только в пределах 100 рублей:

* первое требование будет выполнено автоматически (единственный предикат работает и для существующих, и для новых строк);
* для второго создадим новую ограничительную политику.

**alice=>\c - student**

You are now connected to database "access\_rls" as user "student".

**student=# CREATE POLICY amount ON revenue AS RESTRICTIVE**

**USING (true) -- видны все существующие строки**

**WITH CHECK (abs(amount) <= 100.00); -- новые строки должны удовлетворять**

CREATE POLICY

**student=# GRANT INSERT ON revenue TO bob;**

GRANT

Проверим:

**bob=>INSERT INTO revenue VALUES ('Sales', 42.00);**

INSERT 0 1

**bob=>INSERT INTO revenue VALUES ('PR', 42.00);**

ERROR: new row violates row-level security policy for table "revenue"

**bob=>INSERT INTO revenue VALUES ('Sales', 1000.00);**

ERROR: new row violates row-level security policy "amount" for table "revenue"

Политики, созданные для таблицы, показывают команды psql \d (описание объекта) и \dp (описание привилегий), например:

**student=# \d revenue**

Table "public.revenue"

Column | Type | Collation | Nullable | Default

------------+---------------+-----------+----------+---------

department | text | | |

amount | numeric(10,2) | | |

Policies:

POLICY "amount" AS RESTRICTIVE

USING (true)

WITH CHECK ((abs(amount) <= 100.00))

POLICY "departments"

USING ((department = ( SELECTusers\_depts.department

FROM users\_depts

WHERE (users\_depts.login = CURRENT\_USER))))

Эту информацию можно получить и из представления pg\_policies системного каталога.

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №23. Подключение и аутентификация.

Цель работы: научиться управлять подключением и аутентификацией.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* обеспечивать информационную безопасность на уровне базы данных;
* способы контроля доступа к данным и управления привилегиями;
* основные методы и средства защиты данных в базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

1.Измените конфигурационные файлы (предварительно

сохранив оригиналы) таким образом, чтобы:

безусловно разрешить локальное соединение суперпользователям

student и postgres;

разрешить сетевые подключения всем пользователям к любым базам

данных с аутентификацией по паролю с использованием MD5.

2.Создайте роль alice с паролем, зашифрованным MD5,

и роль bob с паролем, зашифрованным SCRAM-SHA-256.

3.Проверьте возможность подключения под созданными

ролями.

4.Под суперпользовательской ролью посмотрите пароли

пользователей alice и bob в системном каталоге.

5.Восстановите исходные конфигурационные файлы

Измените конфигурационные файлы (предварительно сохранив оригиналы) таким образом, чтобы:

* безусловно разрешить локальное соединение суперпользователям student и postgres;
* разрешить сетевые подключения всем пользователям к любым базам данных с аутентификацией по паролю с использованием MD5.2.

Создайте роль alice с паролем, зашифрованным MD5,и роль bob с паролем, зашифрованным SCRAM-SHA-256.3.

Проверьте возможность подключения под созданными ролями.

4. Под суперпользовательской ролью посмотрите пароли пользователей alice и bob в системном каталоге.

5. Восстановите исходные конфигурационные файлы.

**Ходработы:**

# 1. Настройка аутентификации

Сохраним исходный файл настроек:

**student$ sudocp -n /etc/postgresql/13/main/pg\_hba.conf ~/pg\_hba.conf.orig**

Теперь перезапишем pg\_hba.conf с нуля:

**student$ sudotee /etc/postgresql/13/main/pg\_hba.conf<< EOF**

**local allpostgres trust**

**local all student trust**

**host all allall md5**

**EOF**

local allpostgres trust

local all student trust

host all allall md5

**student$ sudopg\_ctlcluster 13 main reload**

# 2. Создание ролей

**=>SHOWpassword\_encryption;**

password\_encryption

---------------------

md5

(1 row)

**=>CREATE ROLEaliceLOGIN PASSWORD'alice';**

CREATE ROLE

**=>SETpassword\_encryption='scram-sha-256';**

SET

**=>CREATE ROLE bob LOGIN PASSWORD'bob';**

CREATE ROLE

# 3. Проверка подключения

Поскольку настройки требуют ввода пароля, мы укажем его явно в строке подключения.

При выполнении этого задания лучше ввести пароль вручную, чтобы убедиться в том, что он запрашивается.

**=>\c"dbname=student user=alice host=localhost password=alice"**

You are now connected to database "student" as user "alice" on host "localhost" (address "127.0.0.1") at port "5432".

**=>\c"dbname=student user=bob host=localhost password=bob"**

You are now connected to database "student" as user "bob".

# 4. Просмотр паролей

**=>\q**

**student$ psql**

**=>SELECTrolname, rolpasswordFROMpg\_authidWHERErolnameIN ('alice','bob') \gx**

-[ RECORD 1 ]--------------------------------------------------------------------------------------------------------------------------------------

rolname | alice

rolpassword | md5579e43b423b454623383471aeb85cd87

-[ RECORD 2 ]--------------------------------------------------------------------------------------------------------------------------------------

rolname | bob

rolpassword | SCRAM-SHA-256$4096:v+LjzoVkKe1tgqnzG/D9qA==$Mof8EisspWTRRqnG4ut39aa6ojlnYxeixX/2ZulsYno=:mJWBAN1Z6255P2M7tZUZ5p2zzqyka8bJOHr5x3Oyx94=

Пароли хранятся как значение хеш-функции, не допускающее расшифровки. Сервер всегда сравнивает между собой зашифрованные значения — введенный пароль и значение из pg\_authid.

# 5. Восстановление исходных настроек

**student$ sudocp ~/pg\_hba.conf.orig /etc/postgresql/13/main/pg\_hba.conf**

**student$ sudopg\_ctlcluster 13 main reload**

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №24. Репликация и миграция баз данных

Цель работы: Научиться настраивать репликацию и миграцию баз данных в PostgreSQL для обеспечения отказоустойчивости и масштабируемости.

Проверяемые результаты обучения:

* знание структур данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
* приемы администрирования, сопровождения и отказоустойчивости базы данных.
* обеспечивать информационную безопасность на уровне базы данных;
* настраивать миграции бд;
* настраивать репликации БД;
* основные методы и средства защиты данных в базах данных;

Формирование компетенций (указываются коды ОК и ПК)

|  |
| --- |
| ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;  ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;  ОК 4 Эффективно взаимодействовать и работать в коллективе и команде;  ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;  ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.  ПK 1.1. Проектировать базы данных.  ПK 1.2. Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области.  ПK 1.3. Реализовывать базу данных в конкретной системе управления базами данных.  ПK 1.4. Администрировать базы данных.  ПK 1.5. Защищать информацию в базе данных с использованием технологии защиты информации. |

**Задание:**

Изучить механизмы репликации и миграции БД в СУБД.

**Ходработы:**

Настройки по умолчанию позволяют использовать протокол репликации:

* wal\_level = replica,
* max\_wal\_senders = 10,
* разрешение на подключение в pg\_hba.conf.

Создадим автономную резервную копию. С ключом -R утилита сформирует необходимые для репликации конфигурационные параметры.

**student$ sudorm -rf /home/student/basebackup**

**student$ pg\_basebackup --pgdata=/home/student/basebackup -R**

Убеждаемся, что второй сервер остановлен, и выкладываем автономную копию в его каталог данных. Владельцем файлов должен быть пользователь postgres.

**student$ sudopg\_ctlcluster 13 replicastatus**

pg\_ctl: no server running

**student$ sudorm -rf /var/lib/postgresql/13/replica**

**student$ sudomv /home/student/basebackup/ /var/lib/postgresql/13/replica**

**student$ sudochown -R postgres:postgres /var/lib/postgresql/13/replica**

# Реплика

Утилита pg\_basebackup добавила в postgresql.auto.conf параметры соединения с основным сервером:

**student$ sudocat /var/lib/postgresql/13/replica/postgresql.auto.conf**

# Do not edit this file manually!

# It will be overwritten by the ALTER SYSTEM command.

primary\_conninfo = 'user=student passfile=''/home/student/.pgpass'' channel\_binding=prefer host=''/var/run/postgresql'' port=5432 sslmode=prefer sslcompression=0 sslsni=1 ssl\_min\_protocol\_version=TLSv1.2 gssencmode=prefer krbsrvname=postgrestarget\_session\_attrs=any'

А также создала пустой файл standby.signal, который даст указание серверу войти в режим постоянного восстановления.

**student$ sudols -l /var/lib/postgresql/13/replica/standby.signal**

-rw------- 1 postgrespostgres 0 янв 16 10:32 /var/lib/postgresql/13/replica/standby.signal

Можнозапускатьсервер.

**student$ sudopg\_ctlcluster 13 replica start**

Посмотримнапроцессыреплики.

**student$ sudohead -n 1 /var/lib/postgresql/13/replica/postmaster.pid**

24784

**student$ sudops -o pid,command --ppid 24784**

PID COMMAND

24785 postgres: 13/replica: startup recovering 000000010000000000000008

24787 postgres: 13/replica: checkpointer

24788 postgres: 13/replica: background writer

24789 postgres: 13/replica: stats collector

24790 postgres: 13/replica: walreceiver streaming 0/8000060

Процесс walreceiver принимает поток журнальных записей, процесс startup применяет изменения.

И сравним с процессами мастера.

**student$ sudohead -n 1 /var/lib/postgresql/13/main/postmaster.pid**

10032

**student$ sudops -o pid,command --ppid 10032**

PID COMMAND

10034 postgres: 13/main: checkpointer

10035 postgres: 13/main: background writer

10036 postgres: 13/main: walwriter

10037 postgres: 13/main: autovacuum launcher

10038 postgres: 13/main: stats collector

10039 postgres: 13/main: logical replication launcher

24473 postgres: 13/main: student student [local] idle

24791 postgres: 13/main: walsender student [local] streaming 0/8000060

Здесь добавился процесс walsender.

# Проверка репликации

Состояние репликации можно смотреть в представлении на мастере:

**student$ psql -p 5432**

**=>SELECT \* FROMpg\_stat\_replication\gx**

-[ RECORD 1 ]----+------------------------------

pid | 24791

usesysid | 16384

usename | student

application\_name | 13/replica

client\_addr |

client\_hostname |

client\_port | -1

backend\_start | 2024-01-16 10:34:02.471843+03

backend\_xmin |

state | streaming

sent\_lsn | 0/8000060

write\_lsn | 0/8000060

flush\_lsn | 0/8000060

replay\_lsn | 0/8000060

write\_lag |

flush\_lag |

replay\_lag |

sync\_priority | 0

sync\_state | async

reply\_time | 2024-01-16 10:34:02.61037+03

Выполним несколько команд на мастере:

**=>CREATE DATABASEreplica\_overview;**

CREATE DATABASE

**=>\creplica\_overview;**

You are now connected to database "replica\_overview" as user "student".

**=>CREATE TABLEtest(id integerPRIMARY KEY, descrtext);**

CREATE TABLE

Проверимреплику:

**student$ psql -p 5433 -d replica\_overview**

**=>SELECT \* FROM test;**

id | descr

----+-------

(0 rows)

**=>INSERT INTO test VALUES (1, 'Раз');**

INSERT 0 1

**=>SELECT \* FROM test;**

id | descr

----+-------

1 | Раз

(1 row)

При этом изменения на реплике не допускаются:

**=>INSERT INTO test VALUES (2, 'Два');**

ERROR: cannot execute INSERT in a read-only transaction

# Переключение на реплику

Теперь переведем реплику из режима восстановления в обычный режим.

**student$ sudopg\_ctlcluster 13 replica promote**

**=>INSERT INTO test VALUES (2, 'Два');**

INSERT 0 1

Получили два самостоятельных, никак не связанных друг с другом сервера.

**Критерии оценки.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен

# **Контроль приобретения практического опыта. Оценка по учебной и (или) производственной практике (по профилю специальности)**

**4.1. Общие положения**

Целью оценки по учебной и (или) производственной практике (по профилю специальности) является оценка:

1) профессиональных и общих компетенций;

2) практического опыта и умений.

Оценка по учебной и (или) производственной практике выставляется на основании данных аттестационного листа (характеристики профессиональной деятельности обучающегося/студента на практике) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и (или) требованиями организации, в которой проходила практика.

**4.2. Виды работ практики и проверяемые результаты обучения по частично вариативному профессиональному модулю ПМ.01«Технология разработки и защиты баз данных»**

Таблица 7

|  |  |  |
| --- | --- | --- |
| **Иметь практический опыт** | **Виды и объем работ на учебной практике, требования к их выполнению и/ или условия выполнения** | **Документ, подтверждающий качество выполнения работ** |
| **1** | **2** | **3** |
| * в работе с объектами базы данных в конкретной системе управления базами данных; * использовании стандартных методов защиты объектов базы данных; * работе с документами отраслевой направленности | * сбор, обработка и анализ информации для проектирования баз данных; * проектирование базы данных на основе анализа предметной области; * разработка объектов базы данных в соответствии с результатами анализа предметной области; * реализация базы данных в конкретной системе управления базами данных; * администрирование базы данных;   защита информации в базе данных с использованием технологии защиты информации. | * аттестационный лист о прохождении практики или справка с места работы. |

Таблица 8

|  |  |  |
| --- | --- | --- |
| **Иметь практический опыт** | **Виды и объем работ на производственной практике (по профилю специальности), требования к их выполнению и/ или условия выполнения** | **Документ, подтверждающий качество выполнения работ** |
| **1** | **2** | **3** |
| * В работе с объектами базы данных в конкретной системе управления базами данных; * Использовании стандартных методов защиты объектов базы данных; * В работе с документами отраслевой направленности | * Получение задания, сбор и описание требований к разработке ПП по заданию предприятия и согласованию с руководителем практики или собственная тема, если выбрана и утверждена с дипломным руководителем. * Проектирование базы данных, соответствующей требованиям, с использованием современных case-средств: * Разработка концептуальной, логической и физической модели базы данных. * Формирование схемы данных. * Реализация базы данных в конкретной системе управления базами данных (СУБД): * Описание СУБД, в которой реализована база данных, обоснование выбора СУБД. * Описание способов обеспечения достоверности информации в базе данных. * Разработка объектов базы данных. * Применение стандартных методов для защиты объектов базы данных. * Выполнение процедур резервного копирования и восстановления базы данных. * Обеспечение информационной безопасности на уровне базы данных. * Создание хранимых процедур и триггеров в базе данных. | * аттестационный лист о прохождении практики или справка с места работы. |

# **Контрольно-оценочные материалы для ПМ.01 «Разработка, администрирование и защита баз данных»**

# **5.1. Контрольно-оценочные материалы для дифференцированного зачета по МДК 01.01 «Технология разработки и защиты баз данных»**

**Перечень вопросов и заданий к дифференцированному зачету:**

**Вопросы:**

1. Дайте определение следующим понятиям: таблица, база данных, домен, атрибут, информационная система, объект, класс объектов, СУБД.
2. Что такое ключевое поле? Дайте определения существующим видам ключей.
3. Какие существуют типы отношений между таблицами? Дайте определение каждому отношению.
4. Какие существуют типы моделей данных? Опишите их.
5. Что такое нормализация отношений? Расскажите о шести нормальных формах
6. Что такое индексация, индексы? ER-модели базы данных (определение, назначение, методы описания).
7. Какие способы контроля доступа к данным вы знаете?
8. Дайте определение целостности данных.
9. Дайте определение и назначение следующим понятиям: форма, отчёты и запросы в СУБД.
10. Приведитеформаткоманды SELECT (IN, BETWEEN, LIKE, DISTINCT).
11. JOIN (структура, примеры применения).
12. Опишите отличие WHERE от HAVING (приведите примеры применения).
13. Опишите отличие GROUPBY от ORDERBY (приведите примеры применения).
14. Какие основные принципы технологии разработки баз данных.
15. Какие инструменты обеспечивают защиту данных в базе данных?
16. Что такое резервное копирование данных и для чего оно нужно?
17. Какие методы мониторинга баз данных вы знаете?  
    15. Что такое SQL и какие операторы в нем используются?
18. Какие типы индексов существуют в базах данных?
19. Какие виды транзакций поддерживаются в базах данных?
20. Что такое репликация данных и как она используется для обеспечения доступности баз данных?
21. Какие роли пользователей могут существовать в СУБД?  
    безопасности баз данных вы знаете и как они применяются?
22. Что такое СУБД и какую роль она играет в разработке баз данных?
23. Какие преимущества предоставляет PostgreSQL как СУБД?
24. Какие основные типы данных поддерживает PostgreSQL?
25. Как создать новую таблицу в PostgreSQL с помощью SQL запроса?
26. Какие ограничения целостности данных можно применить к таблицам в PostgreSQL?
27. Как добавить новую запись в таблицу PostgreSQL?
28. Как обновить существующую запись в таблице PostgreSQL?
29. Как удалить запись из таблицы PostgreSQL?
30. Какие операторы сравнения поддерживает PostgreSQL?
31. Что такое хранимые процедуры (storedprocedures) и как их создать в PostgreSQL?
32. Как проверить производительность запросов в PostgreSQL с помощью EXPLAIN?
33. Как управлять схемами (schemas) в PostgreSQL?
34. Как использовать функции окон для аналитических запросов в PostgreSQL?
35. Как добавить комментарии к таблицам и столбцам в PostgreSQL?

**Практические задания:**

**Задание 1:** Создайте базу данных "Интернет-магазин" со следующими таблицами:

1. Таблица "Товары":
   * id (первичныйключ, auto\_increment)
   * название
   * описание
   * цена
   * количество на складе
2. Таблица "Пользователи":
   * id (первичныйключ, auto\_increment)
   * имя
   * email (уникальный)
   * пароль
   * роль (покупатель/администратор)
3. Таблица "Заказы":
   * id (первичныйключ, auto\_increment)
   * id\_пользователя (внешний ключ, ссылается на таблицу "Пользователи")
   * дата\_заказа
   * статус (новый/оплачен/доставлен)
4. Таблица "Корзина":
   * id (первичныйключ, auto\_increment)
   * id\_пользователя (внешний ключ, ссылается на таблицу "Пользователи")
   * id\_товара (внешний ключ, ссылается на таблицу "Товары")
   * количество

## Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 5 записей в каждую таблицу.

## Задание 3. Выполнение запросов

1. Выведите список всех товаров, отсортированных по цене в порядке убывания.
2. Найдите товары, количество которых на складе меньше 10.
3. Выведите список пользователей, зарегистрированных в качестве покупателей.
4. Найдите все заказы, сделанные пользователем с email "[example@example.com](mailto:example@example.com)".
5. Выведите общее количество товаров в корзине для пользователя с id = 3.
6. Обновите статус заказа с id = 2 на "оплачен".
7. Удалите товар с id = 4 из корзины пользователя с id = 2.

## Задание 4. Создание представления

Создайте представление "Популярные\_товары", которое выводит 3 самых популярных товара (с наибольшим количеством в корзине).

## Задание 5. Создание триггера

Создайте триггер, который при добавлении нового товара в корзину проверяет, есть ли достаточное количество товара на складе. Если количество товара в корзине превышает количество на складе, триггер должен выдавать ошибку.

## Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Оформить\_заказ", которая принимает id пользователя в качестве параметра. Процедура должна:

1. Создать новый заказ для указанного пользователя с текущей датой и статусом "новый".
2. Перенести товары из корзины пользователя в таблицу "Заказы".
3. Очистить корзину пользователя.
4. Уменьшить количество товаров на складе в соответствии с заказанным количеством.

Выполните процедуру для пользователя с id = 2.

**Задание 2.** Создайте базу данных "Библиотека" со следующими таблицами:

1. Таблица "Книги":
   * id (первичныйключ, auto\_increment)
   * название
   * автор
   * издательство
   * год\_издания
   * количество\_экземпляров
2. Таблица "Читатели":
   * id (первичныйключ, auto\_increment)
   * фамилия
   * имя
   * email (уникальный)
   * телефон
3. Таблица "Абонементы":
   * id (первичныйключ, auto\_increment)
   * id\_читателя (внешний ключ, ссылается на таблицу "Читатели")
   * id\_книги (внешний ключ, ссылается на таблицу "Книги")
   * дата\_выдачи
   * дата\_возврата
   * статус (выдана/возвращена)

## Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 10 записей в таблицу "Книги", 5 записей в таблицу "Читатели" и 8 записей в таблицу "Абонементы".

## Задание 3. Выполнение запросов

1. Выведите список всех книг, отсортированных по названию в алфавитном порядке.
2. Найдите книги, изданные после 2010 года.
3. Выведите список читателей, у которых есть активные абонементы (статус "выдана").
4. Найдите все абонементы, выданные читателю с фамилией "Иванов".
5. Выведите количество книг, находящихся в библиотеке (без учета выданных).
6. Обновите статус абонемента с id = 3 на "возвращена" и увеличьте количество экземпляров соответствующей книги на 1.
7. Удалите книгу с id = 7 из базы данных.

## Задание 4. Создание представления

Создайте представление "Популярные\_книги", которое выводит 5 самых популярных книг (с наибольшим количеством выданных экземпляров).

## Задание 5. Создание триггера

Создайте триггер, который при добавлении нового абонемента проверяет, есть ли достаточное количество экземпляров книги на складе. Если количество выданных экземпляров превышает количество имеющихся, триггер должен выдавать ошибку.

## Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Выдать\_книгу", которая принимает id читателя и id книги в качестве параметров. Процедура должна:

1. Создать новый абонемент для указанного читателя и книги с текущей датой выдачи и датой возврата через 30 дней.
2. Уменьшить количество экземпляров книги на 1.

Выполните процедуру для читателя с id = 3 и книги с id = 5.Это практическое задание охватывает основные операции с базой данных, такие как создание таблиц, заполнение данными, выполнение запросов, создание представлений, триггеров и хранимых процедур. Оно поможет закрепить знания и навыки работы с СУБД MySQL.

**Задание3.**Создайте базу данных "Университет" со следующими таблицами:

1. Таблица "Студенты":
   * id (первичныйключ, auto\_increment)
   * фамилия
   * имя
   * дата\_рождения
   * курс
   * группа
2. Таблица "Преподаватели":
   * id (первичныйключ, auto\_increment)
   * фамилия
   * имя
   * кафедра
   * должность
3. Таблица "Предметы":
   * id (первичныйключ, auto\_increment)
   * название
   * кафедра
   * часы
4. Таблица "Оценки":
   * id (первичныйключ, auto\_increment)
   * id\_студента (внешний ключ, ссылается на таблицу "Студенты")
   * id\_предмета (внешний ключ, ссылается на таблицу "Предметы")
   * оценка

## Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 15 записей в таблицу "Студенты", 5 записей в таблицу "Преподаватели", 8 записей в таблицу "Предметы" и 20 записей в таблицу "Оценки".

## Задание 3. Выполнение запросов

1. Выведите список всех студентов, отсортированных по курсу и группе.
2. Найдите студентов, у которых средний балл выше 4.5.
3. Выведите список предметов, преподаваемых на кафедре "Информатики".
4. Найдите оценки студента с фамилией "Петров".
5. Выведите количество студентов на каждом курсе.
6. Обновите оценку студента с id = 3 по предмету с id = 2 на 5.
7. Удалите преподавателя с id = 4 из базы данных.

## Задание 4. Создание представления

Создайте представление "Лучшие\_студенты", которое выводит топ-5 студентов с самыми высокими средними баллами.

## Задание 5. Создание триггера

Создайте триггер, который при добавлении новой оценки проверяет, что оценка находится в диапазоне от 2 до 5. Если оценка не соответствует этому диапазону, триггер должен отклонить операцию.

## Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Выдать\_оценку", которая принимает id студента, id предмета и оценку в качестве параметров. Процедура должна:

1. Добавить новую оценку для указанного студента и предмета.
2. Посчитать средний балл студента и обновить его в таблице "Студенты".

Выполните процедуру для студента с id = 5, предмета с id = 3 и оценки 4. Это практическое задание охватывает основные операции с базой данных, такие как создание таблиц, заполнение данными, выполнение запросов, создание представлений, триггеров и хранимых процедур. Оно поможет закрепить знания и навыки работы с СУБД MySQL.

**Задание 4.**

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 10 записей в таблицу "Товары", 5 записей в таблицу "Покупатели", 8 записей в таблицу "Заказы" и 20 записей в таблицу "Детали\_заказа".

Задание 3. Выполнение запросов

Выведите список всех товаров, отсортированных по цене в порядке возрастания.

Найдите товары, у которых количество на складе меньше 5.

Выведите список покупателей, у которых есть оформленные заказы.

Найдите все заказы, сделанные покупателем с email "[example@example.com](mailto:example@example.com)".

Выведите общую сумму каждого заказа.

Обновите статус заказа с id = 2 на "обрабатывается".

Удалите детали заказа с id = 5.

Задание 4. Создание представления

Создайте представление "Суммарные\_заказы", которое выводит общую сумму каждого заказа.

Задание 5. Создание триггера

Создайте триггер, который при добавлении новой детали заказа автоматически обновляет количество товара на складе.

Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Оформить\_заказ", которая принимает id покупателя и id товара в качестве параметров. Процедура должна:

Создать новый заказ для указанного покупателя с текущей датой и статусом "новый".

Добавить детали заказа для выбранного товара.

Уменьшить количество товара на складе соответственно.

Выполните процедуру для покупателя с id = 3 и товара с id = 7.

**Задание 5.**

Создайте базу данных "Автосервис" со следующими таблицами:

Таблица "Автомобили":

id (первичный ключ, auto\_increment)

марка

модель

год\_выпуска

гос\_номер

владелец

Таблица "Клиенты":

id (первичный ключ, auto\_increment)

фамилия

имя

телефон

email (уникальный)

Таблица "Услуги":

id (первичный ключ, auto\_increment)

название

описание

стоимость

Таблица "Заказы":

id (первичныйключ, auto\_increment)

id\_автомобиля (внешний ключ, ссылается на таблицу "Автомобили")

id\_клиента (внешний ключ, ссылается на таблицу "Клиенты")

дата\_приема

дата\_выдачи

статус (новый/в\_работе/выполнен)

Таблица "Детали\_заказа":

id (первичный ключ, auto\_increment)

id\_заказа (внешний ключ, ссылается на таблицу "Заказы")

id\_услуги (внешний ключ, ссылается на таблицу "Услуги")

стоимость

количество

Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 10 записей в таблицу "Автомобили", 5 записей в таблицу "Клиенты", 8 записей в таблицу "Услуги", 12 записей в таблицу "Заказы" и 20 записей в таблицу "Детали\_заказа".

Задание 3. Выполнение запросов

Выведите список всех автомобилей, отсортированных по году выпуска в порядке убывания.

Найдите автомобили, принадлежащие клиенту с фамилией "Иванов".

Выведите список клиентов, у которых есть активные заказы (статус "в\_работе" или "выполнен").

Найдите все заказы, выполненные для автомобиля с гос. номером "А123БС".

Выведите общую стоимость каждого заказа.

Обновите статус заказа с id = 5 на "выполнен".

Удалите деталь заказа с id = 8.

Задание 4. Создание представления

Создайте представление "Популярные\_услуги", которое выводит топ-3 самых популярных услуг (с наибольшим количеством в заказах).

Задание 5. Создание триггера

Создайте триггер, который при добавлении новой детали заказа автоматически рассчитывает общую стоимость детали.

Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Принять\_заказ", которая принимает id клиента, id автомобиля и массив id услуг в качестве параметров. Процедура должна:

Создать новый заказ для указанного клиента и автомобиля с текущей датой приема и статусом "новый".

Добавить детали заказа для выбранных услуг.

Рассчитать общую стоимость заказа.

Выполните процедуру для клиента с id = 3, автомобиля с id = 7 и услуг с id = 2, 4,

**Задание 6.**

Создайте базу данных "Туристическое агентство" со следующими таблицами:

Таблица "Туры":

id (первичный ключ, auto\_increment)

название

направление

дата\_начала

дата\_окончания

стоимость

количество\_мест

Таблица "Клиенты":

id (первичныйключ, auto\_increment)

фамилия

имя

телефон

email (уникальный)

Таблица "Бронирования":

id (первичный ключ, auto\_increment)

id\_тура (внешний ключ, ссылается на таблицу "Туры")

id\_клиента (внешний ключ, ссылается на таблицу "Клиенты")

дата\_бронирования

количество\_мест

статус (новое/подтверждено/отменено)

Таблица "Отзывы":

id (первичный ключ, auto\_increment)

id\_тура (внешний ключ, ссылается на таблицу "Туры")

id\_клиента (внешний ключ, ссылается на таблицу "Клиенты")

оценка

текст

Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 10 записей в таблицу "Туры", 15 записей в таблицу "Клиенты", 12 записей в таблицу "Бронирования" и 8 записей в таблицу "Отзывы".

Задание 3. Выполнение запросов

Выведите список всех туров, отсортированных по направлению и дате начала.

Найдите туры, у которых осталось меньше 5 свободных мест.

Выведите список клиентов, сделавших бронирование.

Найдите все бронирования, сделанные клиентом с email "[example@example.com](mailto:example@example.com)".

Выведите среднюю оценку для каждого тура.

Обновите статус бронирования с id = 6 на "подтверждено".

Удалите отзыв с id = 4.

Задание 4. Создание представления

Создайте представление "Популярные\_направления", которое выводит топ-3 самых популярных направлений (с наибольшим количеством туров).

Задание 5. Создание триггера

Создайте триггер, который при добавлении нового бронирования проверяет, что количество забронированных мест не превышает количество свободных мест в туре. Если количество превышено, триггер должен отклонить операцию.

Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Забронировать\_тур", которая принимает id клиента, id тура и количество мест в качестве параметров. Процедура должна:

Создать новое бронирование для указанного клиента и тура с текущей датой и статусом "новое".

Уменьшить количество свободных мест в туре соответственно.

Выполните процедуру для клиента с id = 5, тура с id = 3 и 2 мест

**Задание 7.**

Создайте базу данных "Медицинская клиника" со следующими таблицами:

Таблица "Пациенты":

id (первичный ключ, auto\_increment)

фамилия

имя

дата\_рождения

телефон

email (уникальный)

Таблица "Врачи":

id (первичный ключ, auto\_increment)

фамилия

имя

специальность

стаж

Таблица "Приемы":

id (первичныйключ, auto\_increment)

id\_пациента (внешний ключ, ссылается на таблицу "Пациенты")

id\_врача (внешний ключ, ссылается на таблицу "Врачи")

дата\_приема

диагноз

лечение

Таблица "Анализы":

id (первичныйключ, auto\_increment)

id\_приема (внешний ключ, ссылается на таблицу "Приемы")

название

результат

дата\_сдачи

Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 15 записей в таблицу "Пациенты", 8 записей в таблицу "Врачи", 20 записей в таблицу "Приемы" и 12 записей в таблицу "Анализы".

Задание 3. Выполнение запросов

Выведите список всех пациентов, отсортированных по фамилии в алфавитном порядке.

Найдите пациентов, у которых день рождения в текущем месяце.

Выведите список врачей, специализирующихся на "Терапия".

Найдите все приемы, назначенные пациенту с фамилией "Иванов".

Выведите среднее количество приемов на одного врача.

Обновите диагноз приема с id = 7 на "ОРВИ".

Удалите анализ с id = 5.

Задание 4. Создание представления

Создайте представление "Популярные\_врачи", которое выводит топ-3 самых популярных врачей (с наибольшим количеством приемов).

Задание 5. Создание триггера

Создайте триггер, который при добавлении нового анализа автоматически устанавливает текущую дату сдачи.

Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Записать\_на\_прием", которая принимает id пациента, id врача и дату приема в качестве параметров. Процедура должна:

Создать новый прием для указанного пациента и врача с указанной датой.

Проверить, что у врача нет других приемов в это время.

Выполните процедуру для пациента с id = 3, врача с id = 5 и даты "2023-05-15".

**Задание 8.**

Создайте базу данных "Интернет-магазин электроники" со следующими таблицами:

Таблица "Товары":

id (первичный ключ, auto\_increment)

название

категория

производитель

цена

количество\_на\_складе

Таблица "Покупатели":

id (первичный ключ, auto\_increment)

фамилия

имя

email (уникальный)

телефон

адрес

Таблица "Заказы":

id (первичный ключ, auto\_increment)

id\_покупателя (внешний ключ, ссылается на таблицу "Покупатели")

дата\_заказа

статус (новый/в\_обработке/доставлен)

Таблица "Детали\_заказа":

id (первичный ключ, auto\_increment)

id\_заказа (внешний ключ, ссылается на таблицу "Заказы")

id\_товара (внешний ключ, ссылается на таблицу "Товары")

количество

цена\_за\_единицу

Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 15 записей в таблицу "Товары", 10 записей в таблицу "Покупатели", 12 записей в таблицу "Заказы" и 20 записей в таблицу "Детали\_заказа".

Задание 3. Выполнение запросов

Выведите список всех товаров, отсортированных по цене в порядке возрастания.

Найдите товары, относящиеся к категории "Смартфоны" и имеющие более 10 единиц на складе.

Выведите список покупателей, сделавших хотя бы один заказ.

Найдите все заказы, сделанные покупателем с email "[example@example.com](mailto:example@example.com)".

Выведите общую сумму каждого заказа.

Обновите статус заказа с id = 7 на "в\_обработке".

Удалите деталь заказа с id = 12.

Задание 4. Создание представления

Создайте представление "Топ\_продаваемые\_товары", которое выводит топ-5 самых продаваемых товаров (с наибольшим количеством в заказах).

Задание 5. Создание триггера

Создайте триггер, который при добавлении новой детали заказа автоматически обновляет количество товара на складе.

Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Оформить\_заказ", которая принимает id покупателя и массив id товаров в качестве параметров. Процедура должна:

Создать новый заказ для указанного покупателя с текущей датой и статусом "новый".

Добавить детали заказа для выбранных товаров.

Уменьшить количество товаров на складе соответственно.

Рассчитать общую стоимость заказа.

Выполните процедуру для покупателя с id = 5 и товаров с id = 3, 7, 11.

**Задание 9.**

Создайте базу данных "Прокат автомобилей" со следующими таблицами:

Таблица "Автомобили":

id (первичный ключ, auto\_increment)

марка

модель

год\_выпуска

гос\_номер

стоимость\_проката\_в\_день

Таблица "Клиенты":

id (первичныйключ, auto\_increment)

фамилия

имя

телефон

email (уникальный)

паспортные\_данные

Таблица "Аренда":

id (первичный ключ, auto\_increment)

id\_автомобиля (внешний ключ, ссылается на таблицу "Автомобили")

id\_клиента (внешний ключ, ссылается на таблицу "Клиенты")

дата\_начала

дата\_окончания

стоимость

статус (новая/активная/завершена)

Таблица "Штрафы":

id (первичный ключ, auto\_increment)

id\_аренды (внешний ключ, ссылается на таблицу "Аренда")

причина

сумма

Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 10 записей в таблицу "Автомобили", 15 записей в таблицу "Клиенты", 12 записей в таблицу "Аренда" и 8 записей в таблицу "Штрафы".

Задание 3. Выполнение запросов

Выведите список всех автомобилей, отсортированных по стоимости проката в порядке убывания.

Найдите автомобили, выпущенные после 2015 года.

Выведите список клиентов, у которых есть активные аренды (статус "активная").

Найдите все аренды, совершенные клиентом с фамилией "Иванов".

Выведите общую сумму штрафов для каждой аренды.

Обновите статус аренды с id = 6 на "завершена".

Удалите штраф с id = 4.

Задание 4. Создание представления

Создайте представление "Популярные\_модели", которое выводит топ-3 самых популярных моделей автомобилей (с наибольшим количеством в арендах).

Задание 5. Создание триггера

Создайте триггер, который при добавлении новой аренды автоматически рассчитывает стоимость аренды на основе стоимости проката и количества дней.

Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Оформить\_аренду", которая принимает id клиента, id автомобиля, дату начала и дату окончания в качестве параметров. Процедура должна:

Создать новую аренду для указанного клиента и автомобиля с указанными датами и статусом "новая".

Рассчитать стоимость аренды на основе стоимости проката и количества дней.

Выполните процедуру для клиента с id = 5, автомобиля с id = 3, даты начала "2023-05-01" и даты окончания "2023-05-07".

**Задание 10.**

Создайте базу данных "Фитнес-клуб" со следующими таблицами:

Таблица "Члены\_клуба":

id (первичный ключ, auto\_increment)

фамилия

имя

дата\_рождения

телефон

email (уникальный)

Таблица "Тренеры":

id (первичный ключ, auto\_increment)

фамилия

имя

специализация

стаж

Таблица "Занятия":

id (первичныйключ, auto\_increment)

название

описание

id\_тренера (внешний ключ, ссылается на таблицу "Тренеры")

дата\_и\_время

Таблица "Посещения":

id (первичныйключ, auto\_increment)

id\_члена\_клуба (внешний ключ, ссылается на таблицу "Члены\_клуба")

id\_занятия (внешний ключ, ссылается на таблицу "Занятия")

дата\_посещения

Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 20 записей в таблицу "Члены\_клуба", 8 записей в таблицу "Тренеры", 12 записей в таблицу "Занятия" и 30 записей в таблицу "Посещения".

Задание 3. Выполнение запросов

Выведите список всех членов клуба, отсортированных по фамилии в алфавитном порядке.

Найдите членов клуба, родившихся в текущем году.

Выведите список тренеров, специализирующихся на "Йога".

Найдите все посещения, сделанные членом клуба с фамилией "Петров".

Выведите среднее количество посещений на одного члена клуба.

Обновите специализацию тренера с id = 4 на "Пилатес".

Удалите занятие с id = 7.

Задание 4. Создание представления

Создайте представление "Популярные\_занятия", которое выводит топ-3 самых популярных занятий (с наибольшим количеством посещений).

Задание 5. Создание триггера

Создайте триггер, который при добавлении нового посещения автоматически устанавливает текущую дату посещения.

Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Записаться\_на\_занятие", которая принимает id члена клуба и id занятия в качестве параметров. Процедура должна:

Создать новое посещение для указанного члена клуба и занятия с текущей датой.

Проверить, что у члена клуба нет других посещений в это же время.

Выполните процедуру для члена клуба с id = 5 и занятия с id = 3.

**Задание 11.**

Создайте базу данных "Онлайн кинотеатр" со следующими таблицами:

Таблица "Фильмы":

id (первичный ключ, auto\_increment)

название

жанр

режиссер

год\_выпуска

рейтинг

Таблица "Пользователи":

id (первичный ключ, auto\_increment)

имя

email (уникальный)

пароль

дата\_регистрации

Таблица "Просмотры":

id (первичный ключ, auto\_increment)

id\_пользователя (внешний ключ, ссылается на таблицу "Пользователи")

id\_фильма (внешний ключ, ссылается на таблицу "Фильмы")

дата\_просмотра

продолжительность\_просмотра

Задание 2. Заполнение таблиц данными

Заполните таблицы данными, используя оператор INSERT. Добавьте не менее 10 записей в таблицу "Фильмы", 5 записей в таблицу "Пользователи", и 15 записей в таблицу "Просмотры".

Задание 3. Выполнение запросов

Выведите список всех фильмов, отсортированных по рейтингу в порядке убывания.

Найдите фильмы определенного жанра (например, "комедия").

Выведите список пользователей, зарегистрированных в определенный день.

Найдите все просмотры, сделанные определенным пользователем.

Выведите общее время просмотра для каждого фильма.

Обновите рейтинг фильма с id = 3 на новое значение.

Удалите просмотр с id = 8.

Задание 4. Создание представления

Создайте представление "Лучшие\_фильмы", которое выводит топ-5 фильмов с самым высоким рейтингом.

Задание 5. Создание триггера

Создайте триггер, который при добавлении нового просмотра автоматически обновляет продолжительность просмотра для соответствующего фильма.

Задание 6. Создание хранимой процедуры

Создайте хранимую процедуру "Добавить\_просмотр", которая принимает id пользователя, id фильма и продолжительность просмотра в качестве параметров. Процедура должна:

Добавить новый просмотр для указанного пользователя и фильма с текущей датой.

Обновить продолжительность просмотра для фильма.

Выполните процедуру для пользователя с id = 2, фильма с id = 5 и продолжительности просмотра 120 минут.

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
* оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Критерии оценки практического задания**

|  |  |
| --- | --- |
| **Этап задания** | **Балл** |
| Создание базы данных со всеми таблицами | 1 – все таблицы  0 – не полная БД |
| Нормализация таблиц и устранение аномалий | 1 – нормализация проведена, аномалий нет  0,5 – есть незначительные ошибки (не более двух)  0 – нормализация не проведена или ошибок более двух |
| Определение «ключей» в таблицах | 1 – ключи установлены правильно и позволяют обеспечить целостность базы данных  0,5 – ключи определены не везде, но не более двух ошибок (это не относиться к «первичным ключам» – таблицы должны соответствовать первой нормальной форме)  0 – частично отсутствуют первичные ключи или более двух ошибок во «внешних ключах». |
| Реализованы связи между таблицами | 1 – связи, для реализации целостности данных базы данных, установлены все.  0,5 – отсутствует одна связь  0 – отсутствует более одной связи |
| Таблицы заполнены в соответствии с предметной область. | 1 – таблицы заполнены  0 – таблицы не заполнены или заполнены частично |
| Запрос реализован правильно | Каждый правильно реализованный запрос 2 балла  Запрос является правильным, если выданный результат соответствует с заданием |
| Представление реализовано правильно | 2 балла за реализованное представление |
| Триггер реализован правильно | 2 балла за реализованный триггер |
| Хранимая процедура реализована правильно | 3 балла за реализованную хранимую процедуру |
| Максимальное кол-во баллов | 26 |

Таблица – Критерии оценки

|  |  |
| --- | --- |
| **Оценка** | **Количество баллов** |
| Отлично – « 5 » | 22 – 26 баллов |
| Хорошо – « 4 » | 18 – 21 баллов |
| Удовлетворительно – « 3 » | 11 – 17 баллов |
| Неудовлетворительно – « 2 » | 10 и менее 10 баллов |

**5.2. Контрольно-оценочные материалы для квалификационного экзамена**

# **по ПМ.01«Разработка, администрирование и защита баз данных»**

**Перечень экзаменационных вопросов:**

1. Дайте определение следующим понятиям: таблица, база данных, домен, атрибут, информационная система, объект, класс объектов, СУБД.
2. Что такое ключевое поле? Дайте определения существующим видам ключей.
3. Какие существуют типы отношений между таблицами? Дайте определение каждому отношению.
4. Какие существуют типы моделей данных? Опишите их.
5. Что такое нормализация отношений? Расскажите о шести нормальных формах
6. Что такое индексация, индексы? ER-модели базы данных (определение, назначение, методы описания).
7. Какие способы контроля доступа к данным вы знаете?
8. Дайте определение целостности данных.
9. Дайте определение и назначение следующим понятиям: форма, отчёты и запросы в СУБД.
10. Приведитеформаткоманды SELECT (IN, BETWEEN, LIKE, DISTINCT).
11. JOIN (структура, примеры применения).
12. Опишите отличие WHERE от HAVING (приведите примеры применения).
13. Опишите отличие GROUPBY от ORDERBY (приведите примеры применения).
14. Какие основные принципы технологии разработки баз данных.
15. Какие инструменты обеспечивают защиту данных в базе данных?
16. Что такое резервное копирование данных и для чего оно нужно?
17. Какие методы мониторинга баз данных вы знаете?  
    15. Что такое SQL и какие операторы в нем используются?
18. Какие типы индексов существуют в базах данных?
19. Какие виды транзакций поддерживаются в базах данных?
20. Что такое репликация данных и как она используется для обеспечения доступности баз данных?
21. Какие роли пользователей могут существовать в СУБД?  
    безопасности баз данных вы знаете и как они применяются?
22. Что такое СУБД и какую роль она играет в разработке баз данных?
23. Какие преимущества предоставляет PostgreSQL как СУБД?
24. Какие основные типы данных поддерживает PostgreSQL?
25. Как создать новую таблицу в PostgreSQL с помощью SQL запроса?
26. Какие ограничения целостности данных можно применить к таблицам в PostgreSQL?
27. Как добавить новую запись в таблицу PostgreSQL?
28. Как обновить существующую запись в таблице PostgreSQL?
29. Как удалить запись из таблицы PostgreSQL?
30. Какие операторы сравнения поддерживает PostgreSQL?
31. Что такое хранимые процедуры (storedprocedures) и как их создать в PostgreSQL?
32. Как проверить производительность запросов в PostgreSQL с помощью EXPLAIN?
33. Как управлять схемами (schemas) в PostgreSQL?
34. Как использовать функции окон для аналитических запросов в PostgreSQL?
35. Как добавить комментарии к таблицам и столбцам в PostgreSQL?

**Перечень экзаменационных заданий:**

1. Выполнить проектирование базы данных **«Прокат автомобилей».**  
   В Ваш автопарк входит некоторое количество автомобилей различных марок (Ваз, Форд, Ниссан и т.д.), стоимостей и типов кузова (cедан, универсал, хэтчбек). Каждый автомобиль имеет свою стоимость проката. В пункт проката обращаются клиенты.

Все клиенты проходят обязательную регистрацию, при которой о них собирается стандартная информация (фамилия, имя, отчество, адрес, телефон). Каждый клиент может обращаться в пункт проката несколько раз. Все обращения клиентов фиксируются, при этом по каждой сделке запоминаются дата выдачи и ожидаемая дата возврата.

**Задания:**

1. Выбрать из таблицы информацию по прокату авто и вывести вычисляемое поле «Стоимость проката». Стоимость зависит от количества дней, которые авто находится в прокате у клиента.
2. Используя запрос, выбрать автомобили, которые находились в прокате у определенного клиента.
3. Используя параметрический запрос, выбрать все автомобили, которые были выданы в марте 2016 г.
4. Используя запрос, удалить информацию по клиенту.
5. Выполнить проектирование базы данных **«Библиотека».**

В библиотеке хранятся книги. Каждая книга характеризуется следующими параметрами: название автор издательство год издания количество страниц количество экземпляров стоимость книги.

В библиотеке ведется картотека читателей. О них собирается стандартная информация (номер читательского билета, ФИО, адрес и телефон читателя).

Каждый читатель может обращаться в библиотеку несколько раз. В случае выдачи экземпляра книги читателю в библиотеке хранится специальный вкладыш, в котором должны быть записаны следующие сведения:

* номер билета читателя, который взял книгу
* код книги
* дата выдачи книги
* дата возврата
* Отметка о возврате.

Книга имеет много экземпляров и поэтому может быть выдана многим читателям.

В библиотеке работают сотрудники, которые выдают книги читателям. О сотрудниках должна храниться информация.

**Задания:**

1. Выбрать книгу, c максимальным количеством экземпляров.
2. Выбрать читателей, которые имеют задолженность.
3. Определить количество книг по каждому автору.
4. Добавить информацию о новой книге.
5. Выполнить проектирование базы данных **«Ремонт компьютеров».**

Фирма выполняет ремонт компьютеров. Требуется разработать базу данных для хранения информации о выполнении ремонтных работ сотрудниками фирмы. При оформлении заказа фиксируется дата выполнения заказа, выполненная работа, исполнитель работы. Каждый исполнитель получает фиксированный процент вознаграждения от стоимости выполнения работы. Этот процент устанавливается персонально каждому исполнителю при заключении трудового договора между фирмой и работником. Исполнитель получает вознаграждение, которое вычисляется как Стоимость выполнения заказа \* Фиксированный процент вознаграждения.

**Минимальный список характеристик:**

ФИО исполнителя работы, Процент вознаграждения (может различаться для разных исполнителей), Наименование работы, Стоимость работы (фиксированная для каждого наименования работы), Дата выполнения заказа.

Один исполнитель может выполнять различные работы. Одна и та же работа может быть выполнена разными исполнителями.

**Задания:**

1. Внести информацию по новому исполнителю.

2. Выбрать все заказы, в которых содержится работа с названием «Замена вентилятора».

3. Вывести информацию по заказам с вычисляемым полем Стоимость выполнения заказа.

4. Вывести вознаграждения исполнителя по всем заказам. Используя параметрический запрос.

1. Выполнить проектирование базы данных **«Страховая компания».**

Компания имеет различные филиалы по всей стране. Каждый филиал характеризуется названием, адресом и телефоном. В компанию обращаются различные лица с целью заключения договора о страховании. Договор заключается по определенному виду страхования (например, страхование автотранспорта от угона, страхование домашнего имущества, добровольное медицинское страхование). При заключении договора фиксируется дата заключения, страховая сумма, вид страхования, тарифная ставка.

Нужно учесть, что договора заключают страховые агенты. Помимо информации об агентах, нужно еще хранить филиал, в котором работают агенты.

**Задание.**

1. Используя параметрический запрос, выбрать всех агентов, работающих в филиале №1.
2. Выбрать все договора, оформленные сотрудниками, работающими в филиале №2.
3. Подсчитать количество договоров по каждому агенту.
4. Вывести договор с самой минимальной Страховой суммой.
5. Выполнить проектирование базы данных **«Гостиница».**

Гостиница предоставлят номера клиенам на определенный срок. Каждый номер характеризутсявмстимостью комфортностью (люкс, полулюкс, обычный) и ценой. Клиентами являются различные лица, о которых должна храниться определенная информация (ФИО, Паспортные данные, Адрес, Телефон). При заселении фиксируется дата поселения. При выезде из гостиницы запоминается дата освобождения.

**Задания.**

1. Выбрать все номера, которые заняты. Дата освобождения не заполнена.

2. Посчитать количество номеров с комфортностью люкс.

3.Подсчитать количество клиентов, заехавших в гостиницу 11.09.2016г

4. Вывести список клиентов, которые проживали в гостинице в течение месяца.

5. Вывести всех клиентов, из Ростова-на-Дону.

1. Выполнить проектирование базы данных **«Аэропорт».**

**Минимальный список характеристик:**

ФИО пассажира, паспортные данные, название авиакомпании, № рейса, откуда и куда, время в пути, тип рейса (внутренний/международный), дата вылета, время вылета.

**Задания:**

1. Произведите выборку информации обо всех пассажирах, летавших рейсом 2

2. Подсчитайте количество рейсов за сегодняшнюю дату.

3.Составьте список всех пассажиров, улетавших сегодня из Москвы. Отсортируйте список по алфавиту.

4. Измените время вылета рейса 5.

1. Выполнить проектирование базы данных **«Больница».**

Перед приемом в больницу пациент проходит регистрацию. Если пациент уже был на приеме в данной больнице, тогда проводится поиск по базе данного пациента, проводится запись на прием и выдаются амбулаторный лист. В амбулаторном листе указывается, в какое время и на который день назначен прием, к какому специалисту, в каком кабинете будет осуществляться прием, стоимость приема и пометка оплачен прием или нет. Если же пациент впервые в данной больнице, тогда ему присваивается уникальный регистрационный номер, заносятся в базу данных (фамилия, имя и отчество, адрес и дата рождения), а затем уже происходит запись на прием к конкретному врачу и на конкретное время и выдается амбулаторный лист. Один пациент может быть записан на прием к нескольким врачам, каждому приему назначается уникальный номер.

**Задания**

1. Используя запросы, выбрать информацию о пациенте;

2. Вывести информацию о приемах (Код записи, Дата приема, Фамилия пациента, Фамилия доктора, Кабинет);

3. Выдать информацию о количестве пациентов для каждого врача (Использовать группировку по фамилии доктора;

4.Просмотреть расписания врачей в зависимости от назначенных приемов;

Параметрический запрос по фамилии врача и по дате приема. Выбирать необходимо за указанный диапазон дат.

1. Выполнить проектирование базы данных **«Университет».**

**Минимальный список характеристик:**

* ФИО преподавателя, адрес преподавателя ;
* код, название, количество часов, тип контроля;
* номер аудитории, дата, время, группа

Один преподаватель может вести несколько дисциплин и одна дисциплина может вестись несколькими преподавателями.

**Задания**

1. Вывести всех преподавателей, которые ведут дисциплину биология.
2. Вывести все предметы, преподаваемые в группе 37ТО.
3. Используя запрос, вывести расписание по группе за любой период.
4. Выбрать все группы, у которых 11.09.2016 занятия проходили во 2 аудитории.
5. Добавить информацию по новому преподавателю.
6. Выполнить проектирование базы данных **«Зоопарк».**

**Минимальный список характеристик:**

* Вид животного, суточное потребление корма, континент обитания.
* Название комплекса, номер помещения, наличие водоема, отопления, количество животных в помещении.

Один и тот же вид животного в зоопарке может находиться в разных помещениях и в одном помещении может находиться несколько видов животных.

***Вид животного*** (Насекомые, Млекопитающие, Птицы, Земноводные, Рыбы, Пресмыкающиеся)

***Континент обитания*** (Евразия, Северная и Южная Америка, Африка, Австралия, Антарктида)

**Задание.**

1. Вывести всех животных, обитающих в помещении с отоплением.
2. Подсчитать количество животных по каждому виду.
3. Изменить название помещения «Выставочный комплекс» на «Приматы»
4. Выбрать все случаи размножения животных в помещении без водоема.
5. Выполнить проектирование базы данных **«Ломбард».**

Описание предметной области Вы работаете в ломбарде. Вашей задачей является отслеживание финансовой стороны его работы. Деятельность компании организована следующим образом: к вам обращаются различные лица с целью получения денежных средств под залог определенных товаров. У каждого из приходящих к вам клиентов вы запрашиваете фамилию, имя, отчество и другие паспортные данные. После оценивания стоимости принесенного в качестве залога товара вы определяете сумму, которую готовы выдать на руки клиенту, а также свои комиссионные. Кроме того, определяете срок возврата денег. Если клиент согласен, то ваши договоренности фиксируются в виде документа, деньги выдаются клиенту, а товар остается у вас. В случае если в указанный срок не происходит возврата денег, товар переходит в вашу собственность.

Примечание.

Вид товара (браслет, кольцо, картина, серьги)

В документе может присутствовать только один вид товара и к нему описание.

**Задания**

1. Вывести всех клиентов, которые сдавали браслеты в мае 2016 г
2. Вывести все сделки по клиенту Иванов Иван Иванович
3. Вывести все сделки, по которым сумма находится в интервале 8000 и 10000.
4. Использую запрос, удалить клиента.
5. **Выполнить проектирование базы данных «Реализация готовой продукции».**

Описание предметной области. Вы работаете в компании, занимающейся оптово-розничной продажей различных товаров.

Деятельность компании организована следующим образом: компания торгует товарами из определенного спектра. Каждый из этих товаров характеризуется наименованием, оптовой ценой, розничной ценой и справочной информацией. В вашу компанию обращаются покупатели. Для каждого из них вы запоминаете в базе данных стандартные данные (наименование, адрес, телефон, контактное лицо) и составляете по каждой сделке документ, запоминая наряду с покупателем количество купленного им товара и дату покупки.

Обычно покупатели в рамках одной сделки покупают не один товар, а сразу несколько.

**Задания**

1. Вывести клиентов, проживающих в Ростове-на-Дону
2. Выбрать все сделки, оформленные 01.10.2015г.
3. Выбрать все товары, включенные в заказ №1.
4. Добавить информацию о новом товаре
5. **Выполнить проектирование базы данных «Бюро по трудоустройству».**

Вы работаете в бюро по трудоустройству. Деятельность бюро организована следующим образом: бюро готово искать работников для различных работодателей и вакансии для ищущих работу специалистов различного профиля. При обращении к вам клиента-работодателя его стандартные данные (название, вид деятельности, адрес, телефон) фиксируются в базе данных. При обращении к вам клиента-соискателя его стандартные данные (фамилия, имя, отчество, квалификация, профессия, иные данные) также фиксируются в базе данных. По каждому факту удовлетворения интересов обеих сторон составляется документ. В документе указываются соискатель, работодатель, должность и комиссионные (доход бюро).

Необходимо вести справочник «Виды деятельности».

**Задания**

1. Выбрать всех соискателей на должность Бухгалтер.
2. Подсчитать количество работодателей с видом деятельности оптовая торговля.
3. Выбрать сделку с минимальным комиссионным вознаграждением.
4. Используя запрос, удалить соискателя.

DELETE \* FROM Работодатель WHERE Название=[Введите название];

1. **Выполнить проектирование базы данных «Театр».**

* Минимальный список характеристик:
* Актер, ФИО, пол;
* Дата назначения на роль, дата снятия с роли, режиссер, тип роли
* Название роли, название пьесы, дата и время постановки.

На одну и ту же роль могут назначаться разные актеры.

Тип роли: главная, вторая, эпизод, статист т.д.

Жанр (Драма, комедия, фарс).

Пьесы (Гамлет, Ромео и Джульетта, Горе от ума, Недоросль).

**Задания:**

1. Вывести всех актеров, играющих в пьесе Недоросль.
2. Вывести всех актеров, играющих вторую роль.
3. Вывести спектакли (пьесы) прошедшие в мае 2016.
4. Вывести все пьесы, поставленные режиссером Аверченко М.И.
5. **Выполнить проектирование базы данных «Автотранспортное предприятие».**

**Минимальный список характеристик:**

* госномер, марка автомобиля, его техническое состояние, пробег, грузоподъемность, расход топлива,
* табельный номер водителя, ФИО, дата рождения, стаж работы, оклад, категори;
* дата выезда, дата прибытия, место назначения, расстояние, расход горючего, масса груза.

Один и тот же автомобиль может использоваться разными водителями и  один водитель может использовать разные автомобили.

**Задания**

1. Используя запрос, выбрать водителей, использующих заданную марку автомобиля.
2. Подсчитать количество автомобилей, имеющих плохое техническое  состояние.
3. Выбрать водителей, которые выехали в Москву 11.09.2016.
4. Изменить пробег автомобиля с госномером Х007ЕЕ на 1789.
5. **Выполнить проектирование базы данных «Нотариальная контора».**

Вы работаете в нотариальной конторе. Деятельность нотариальной конторы организована следующим образом: фирма готова предоставить клиенту определенный комплекс услуг. Для наведения порядка вы формализовали эти услуги, составив их список с описанием каждой услуги. При обращении к вам клиента его стандартные данные (название, вид деятельности, адрес, телефон) фиксируются в базе данных. По каждому факту оказания услуги клиенту составляется документ. В документе указываются комиссионные (доход конторы), описание сделки.

В рамках одной сделки клиенту может быть оказано несколько услуг. Стоимость каждой услуги фиксирована.

*Услуги - удостоверение сделки, доверенности, завещания, договора (ренты, брачный договор и др.); выдача свидетельства о праве на наследство и тд.*

1. Посчитать сумму каждой услуги в таблице УслугиСделки.
2. Посчитать сумму услуг по каждой сделке.

**3.** Используя параметрический запрос, выбрать все сделки, заключенные с клиентом. Запрос должен содержать название клиента и название услуг.

**4**. Добавить нового клиента.

1. **Выполнить проектирование базы данных «Фирма по продаже запчастей».**

Вы работаете в фирме, занимающейся продажей запасных частей для автомобилей. Основная часть деятельности, находящейся в вашем ведении, связана с работой с поставщиками. Фирма имеет определенный набор поставщиков, по каждому из которых известны название, адрес и телефон. У этих поставщиков вы приобретаете детали. Каждая деталь наряду с названием характеризуется артикулом и ценой (считаем цену постоянной). Каждый факт покупки запчастей у поставщика фиксируется в базе данных, причем обязательными для запоминания являются дата покупки и количество приобретенных деталей.

Покупка может включать как одно наименование детали, так и несколько.

**Задания**

**1.** Посчитать сумму деталей в каждой поставке.

**2**.Используя параметрические запросы, выбрать поставку по определенному поставщику.

**3**. Выбрать детали, которые были поставлены до 01.01.2016г

**4**. Вывести все детали, цена которых больше 5000.

1. **Выполнить проектирование базы данных «Курсы повышения квалификации».**

Вы работаете в учебном заведении и занимаетесь организацией курсов повышения квалификации. В вашем распоряжении имеются сведения о сформированных группах студентов. Группы формируются в зависимости от специальности и отделения. В каждую из них включено определенное количество студентов. Проведение занятий обеспечивает штат преподавателей. Для каждого из них у вас в базе данных зарегистрированы стандартные анкетные данные (фамилия, имя, отчество, телефон) и стаж работы. В результате распределения нагрузки вы получаете информацию о том, сколько часов занятий проводит каждый преподаватель с соответствующими группами. Кроме того, хранятся сведения о типе проводимых занятий (лекции, практика), предмете и оплате за 1 час. Размер почасовой оплаты зависит от предмета и типа занятия.

**Задание**

1. Вывести из нагрузки все предметы с типом занятия Практика.

2. Используя запрос, вывести все занятия, которые вел определенный преподаватель

3.Вывести стоимость проведенных занятий, ФИО преподаватели, название дисциплины и номер группы.

4. Добавить новый предмет.

1. **Выполнить проектирование базы данных «Коммерческая компания».**

Вы являетесь сотрудником коммерческого отдела компании, продающей различные товары через Интернет. Работа компании организована следующим образом: на Интернет - сайте представлены (выставлены на продажу) некоторые товары. Каждый из них имеет некоторое название, цену и единицу измерения (штуки, килограммы, литры). Для проведения исследований и оптимизации работы магазина вы пытаетесь собирать данные с клиентов. При этом для вас определяющее значение имеют стандартные анкетные данные, а также телефон и адрес электронной почты для связи. В случае приобретения товаров на сумму свыше 5000 р. клиент переходит в категорию постоянных. По каждому факту продажи вы автоматически фиксируете клиента, товары, количество, дату продажи, дату доставки.

Обычно клиенты в рамках одного заказа покупают не один вид товара, а несколько видов.

**Задания**

1. Выяснить, есть ли среди клиентов те, которых необходимо перевести в категорию постоянных.

2. Вывести список постоянных клиентов.

3. Вывести продажу с самой максимальной стоимостью

1. **Выполнить проектирование базы данных «Ювелирная мастерская».**

Вы работаете в ювелирной мастерской, осуществляющей изготовление ювелирных изделий для частных лиц на заказ. Вы работаете с определенными материалами (платина, золото, серебро, различные драгоценные камни и т.д.). При обращении потенциального клиента вы определяете, какое именно изделие ему необходимо. Все изготавливаемые изделия принадлежат к некоторому типу (серьги, кольца, броши, браслеты), выполнены из определенного материала, имеют некоторый вес и цену (включающую стоимость материалов и работы).

Ювелирное изделие может состоять из нескольких материалов. В заказе может фигурировать только одно изделие.

**Задания**

1. Выбрать все продажи за месяц апрель 2016

2. Выбрать все изделия, в состав которых входит платина.

3. Подсчитать количество заказов по каждому клиенту

4. Удалить из справочника изделие с весом 0,1

1. **Выполнить проектирование базы данных «Парикмахерская».**

Вы работаете в парикмахерской, обслуживающей клиентов в соответствии с их пожеланиями и некоторым каталогом различных видов стрижки. Так, для каждой стрижки определены название, принадлежность полу (мужская, женская), стоимость работы. Для наведения порядка вы, по мере возможности, составляете базу данных клиентов, запоминая их анкетные данные (фамилия, имя, отчество). Начиная с пятой стрижки, клиент переходит в категорию постоянных. После того как закончена очередная работа, документом фиксируются стрижка, клиент и дата производства работ.

**Задания**

1. Выяснить, есть ли среди клиентов те, которых необходимо перевести в категорию постоянных.

* Посчитать количество стрижек по каждому клиенту
* Если у клиента больше 5 стрижек, в карточке клиента установить признак постоянного клиента.

2.Вывести список постоянных клиентов.

3. Вывести список клиентов, которые посетили парикмахерскую 01.05.2016 г.

4. Добавить в каталог новую стрижку.

1. **Выполнить проектирование базы данных «Химчистка».**

Вы работаете в химчистке, осуществляющей прием у населения вещей для выведения пятен. Для наведения порядка вы, по мере возможности, составляете базу данных клиентов, запоминая их анкетные данные (фамилия, имя, отчество). Все оказываемые услуги подразделяются на виды, имеющие название, тип и стоимость. Работа с клиентом первоначально состоит в определении вида услуги и, соответственно, ее стоимости. Если клиент согласен, он оставляет вещь (при этом фиксируются услуга, клиент и дата приема, наименование изделия) и забирает ее после обработки (при этом фиксируется дата возврата). Справочник изделий хранить не требуется.

Виды услуг:

* чистка текстильных изделий;
* чистка кожаных, меховых, замшевых вещей;
* чистка ковров, пледов, покрывал, подушек;
* чистка мягких игрушек;
* химчистка спецодежды;
* услуги прачечной и портного.

Тип услуг:

* чистка
* химчистка
* пошив
* стирка

**Задания**

1. Выбрать все услуги, оказанные в мае 2016г
2. Вывести все услуги с типом «Чистка».
3. Выбрать всех клиентов, которые обращались в химчистку в 2016 г.
4. Выбрать оказанные услуги, после выполнения которых, клиенты не забрали свои вещи.
5. **Выполнить проектирование базы данных «Занятость актеров».**

Вы являетесь коммерческим директором театра, и в ваши обязанности входит вся организационно-финансовая работа, связанная с привлечением актеров и заключением контрактов. Вы организовали дело следующим образом: каждый год театр осуществляет постановку различных спектаклей. Каждый спектакль имеет определенный бюджет. Для участия в конкретных постановках в определенных ролях привлекаются актеры. С каждым из актеров вы заключаете персональный контракт на определенную сумму. Каждый из актеров имеет некоторый стаж работы, некоторые из них удостоены различных наград и званий.

В рамках одного спектакля на одну и ту же роль привлекается несколько актеров.

**Задания**

1. Вывести всех актеров, у которых стоимость годового контракта выше 500 000.
2. Используя параметрический запрос, вывести всех актеров, играющих в спектакле.
3. Вывести все спектакли, с годом постановки 2015.
4. Используя параметрический запрос, удалить информацию об актере.
5. **Выполнить проектирование базы данных «Выдача банком кредитов».**

В зависимости от условий получения кредита, процентной ставки и срока возврата все кредитные операции делятся на несколько основных видов. Каждый из этих видов имеет свое название. Кредит может получить клиент, при регистрации предоставивший следующие сведения: название, вид собственности, адрес, телефон, контактное лицо. Каждый факт выдачи кредита регистрируется банком, при этом фиксируются сумма кредита, клиент и дата выдачи.

**Виды кредитов** – потребительский, ипотечный, кредитные карты, банковский кредит, автокредитование.

**Вид собственности** – [личная](http://dic.academic.ru/dic.nsf/ruwiki/1008406), частная, коллективная, государственная, [общественная,](http://dic.academic.ru/dic.nsf/ruwiki/665733) смешанная

**Задания**

1. Выбрать всех клиентов, которым были выданы потребительские кредиты в размере более 50 000.
2. Используя запрос, удалить клиента.
3. Добавить новый вид кредита.
4. Вывести все непогашенные кредиты.
5. **Выполнить проектирование базы данных «Учет телефонных разговоров».**

Абонентами компании являются юридические лица, имеющие телефонную точку, ИНН, расчетный счет в банке. Стоимость переговоров зависит от города, в который осуществляется звонок, и времени суток (день, ночь). Каждый звонок абонента автоматически фиксируется в базе данных. При этом запоминаются город, дата, длительность разговора и время суток.

**Задания.**

1. Выбрать все переговоры, которые проводились в ночное время суток. И вывести стоимость ночного тарифа. Посчитать стоимость переговоров.

2. Подсчитать количество звонков по каждому клиенту.

3. Вывести звонок с самой максимальной стоимостью.

4.Используя запрос, удалить клиент

**Пример экзаменационного билета**

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ

«РОСТОВСКИЙ НА-ДОНУ КОЛЛЕДЖСВЯЗИ И ИНФОРМАТИКИ»

|  |  |
| --- | --- |
| РАССМОТРЕНО  на заседании цикловой комиссии  программирования  «\_\_\_» \_\_\_\_ 202**\_г**., протокол № \_\_  Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ *(подпись) (Фамилия, И.О.)* | УТВЕРЖДАЮ  Зам. директора по НМР  \_\_\_\_\_\_\_\_\_\_\_\_ /Подцатова И.В./  *(подпись) (Фамилия, И.О.)*  «\_\_» \_\_\_\_\_\_\_\_ 202\_ г. |

**ПРОФЕССИОНАЛЬНОЕ (ОЦЕНОЧНОЕ) ЗАДАНИЕ № \_\_\_**

**ДЛЯ КВАЛИФИКАЦИОННОГО ЭКЗАМЕНА ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

**ПМ.01 Разработка, администрирование и защита баз данных**

**Специальность 09.02.11 «Разработка и управление программным обеспечением»**

**Учебная группа** \_\_\_\_\_\_\_\_\_\_\_

*Практическая часть:*

**Продукт деятельности**

Выполнить проектирование базы данных **«Прокат автомобилей».**  
В Ваш автопарк входит некоторое количество автомобилей различных марок (Ваз, Форд, Ниссан и т.д.), стоимостей и типов кузова (cедан, универсал, хэтчбек). Каждый автомобиль имеет свою стоимость проката. В пункт проката обращаются клиенты.

Все клиенты проходят обязательную регистрацию, при которой о них собирается стандартная информация (фамилия, имя, отчество, адрес, телефон). Каждый клиент может обращаться в пункт проката несколько раз. Все обращения клиентов фиксируются, при этом по каждой сделке запоминаются дата выдачи и ожидаемая дата возврата.

**Использование базы данных:**

1. Выполнить проектирование БД.
2. Реализовать базу данных в СУБД PostgreSql.
3. Выполнить необходимые запросы согласно предметной области (3 запроса на выборку информации более чем из 2-х таблиц, 1 запрос на удаление данных, 1 запрос на обновление данных в таблице)

*Теоретическая часть* **(профессионально значимая информация):**

1. Основные термины баз данных (предметная область, сущность, атрибут, ключевой элемент и т.д.)

|  |  |
| --- | --- |
| Преподаватели: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | *(Подпись) (Фамилия, И.О)* |
|  | « \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 202\_ г. |

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ

«РОСТОВСКИЙ НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ»

|  |  |
| --- | --- |
| РАССМОТРЕНО  на заседании цикловой комиссии  программирования  «\_\_» \_\_\_\_\_\_\_\_ 202**\_г**., протокол № \_\_  Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_/  *(подпись) (Фамилия, И.О.)* | УТВЕРЖДАЮ  Зам. директора по НМР  \_\_\_\_\_\_\_\_\_\_\_\_ /Подцатова И.В./  *(подпись) (Фамилия, И.О.)*  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 202\_ г. |

**ПРОФЕССИОНАЛЬНОЕ (ОЦЕНОЧНОЕ) ЗАДАНИЕ № \_\_\_**

**ДЛЯ КВАЛИФИКАЦИОННОГО ЭКЗАМЕНА ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

**ПМ.01 Разработка, администрирование и защита баз данных**

**Специальность 09.02.11 «Разработка и управление программным обеспечением» Учебная группа** \_\_\_\_\_\_\_\_\_\_\_

*Практическая часть:*

**Продукт деятельности**

Выполнить проектирование базы данных **«Библиотека».** В библиотеке хранятся книги. Каждая книга характеризуется следующими параметрами: название автор издательство год издания количество страниц количество экземпляров стоимость книги.

В библиотеке ведется картотека читателей. О них собирается стандартная информация (номер читательского билета, ФИО, адрес и телефон читателя).

Каждый читатель может обращаться в библиотеку несколько раз. В случае выдачи экземпляра книги читателю в библиотеке хранится специальный вкладыш, в котором должны быть записаны следующие сведения: номер билета читателя, который взял книгу; код книги; дата выдачи книги; дата возврата; Отметка о возврате.

Книга имеет много экземпляров и поэтому может быть выдана многим читателям.

В библиотеке работают сотрудники, которые выдают книги читателям. О сотрудниках должна храниться информация.

**Использование базы данных:**

1. Выполнить проектирование БД.
2. Реализовать базу данных в СУБД PostgreSql.
3. Выполнить необходимые запросы согласно предметной области (3 запроса на выборку информации более чем из 2-х таблиц, 1 запрос на удаление данных, 1 запрос на обновление данных в таблице)

*Теоретическая часть* **(профессионально значимая информация):**

1. Сетевая модель данных (понятия модели, обзор моделей)

|  |  |
| --- | --- |
| Преподаватели: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | *(Подпись) (Фамилия, И.О)* |
|  | « \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 202\_ г. |

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ

«РОСТОВСКИЙ НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ»

|  |  |
| --- | --- |
| РАССМОТРЕНО  на заседании цикловой комиссии  программирования  «\_\_» \_\_\_\_\_\_ 202**\_г**., протокол № \_  Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/  *(подпись) (Фамилия, И.О.)* | УТВЕРЖДАЮ  Зам. директора по НМР  \_\_\_\_\_\_\_\_\_\_\_\_ /Подцатова И.В./  *(подпись) (Фамилия, И.О.)*  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_ г. |

**ПРОФЕССИОНАЛЬНОЕ (ОЦЕНОЧНОЕ) ЗАДАНИЕ № \_\_\_**

**ДЛЯ КВАЛИФИКАЦИОННОГО ЭКЗАМЕНА ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

**ПМ.01 Разработка, администрирование и защита баз данных**

**Специальность 09.02.11 «Разработка и управление программным обеспечением»**

**Учебная группа** \_\_\_\_\_\_\_\_\_\_\_

*Практическая часть:*

**Продукт деятельности**

Выполнить проектирование базы данных **«Ремонт компьютеров».**

Фирма выполняет ремонт компьютеров. Требуется разработать базу данных для хранения информации о выполнении ремонтных работ сотрудниками фирмы. При оформлении заказа фиксируется дата выполнения заказа, выполненная работа, исполнитель работы. Каждый исполнитель получает фиксированный процент вознаграждения от стоимости выполнения работы. Этот процент устанавливается персонально каждому исполнителю при заключении трудового договора между фирмой и работником. Исполнитель получает вознаграждение, которое вычисляется как Стоимость выполнения заказа \* Фиксированный процент вознаграждения.

**Минимальный список характеристик:**

ФИО исполнителя работы, Процент вознаграждения (может различаться для разных исполнителей), Наименование работы, Стоимость работы (фиксированная для каждого наименования работы), Дата выполнения заказа.

Один исполнитель может выполнять различные работы. Одна и та же работа может быть выполнена разными исполнителями.

**Использование базы данных:**

1. Выполнить проектирование БД.
2. Реализовать базу данных в СУБД PostgreSql.
3. Выполнить необходимые запросы согласно предметной области (3 запроса на выборку информации более чем из 2-х таблиц, 1 запрос на удаление данных, 1 запрос на обновление данных в таблице)

*Теоретическая часть* **(профессионально значимая информация):**

1. Иерархическая (понятия модели, обзор моделей)

|  |  |
| --- | --- |
| Преподаватели: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | *(Подпись) (Фамилия, И.О)* |
|  | « \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 202\_ г. |

Критерии оценки выполнения экзаменационного задания

Таблица – Соответствие баллов этапу выполненного задания

|  |  |
| --- | --- |
| **Этап задания** | **Балл** |
| Создание базы данных со всеми таблицами | 1 – все таблицы  0 – не полная БД |
| Нормализация таблиц и устранение аномалий | 1 – нормализация проведена, аномалий нет  0,5 – есть незначительные ошибки (не более двух)  0 – нормализация не проведена или ошибок более двух |
| Определение «ключей» в таблицах | 1 – ключи установлены правильно и позволяют обеспечить целостность базы данных  0,5 – ключи определены не везде, но не более двух ошибок (это не относиться к «первичным ключам» – таблицы должны соответствовать первой нормальной форме)  0 – частично отсутствуют первичные ключи или более двух ошибок во «внешних ключах». |
| Реализованы связи между таблицами | 1 – связи, для реализации целостности данных базы данных, установлены все.  0,5 – отсутствует одна связь  0 – отсутствует более одной связи |
| Таблицы заполнены в соответствии с предметной область. | 1 – таблицы заполнены  0 – таблицы не заполнены или заполнены частично |
| Запрос реализован правильно | Каждый правильно реализованный запрос 2 балла  Запрос является правильным, если выданный результат соответствует с заданием |
| Максимальное кол-во баллов | 15 |

Таблица – Критерии оценки

|  |  |
| --- | --- |
| **Оценка** | **Количество баллов** |
| Отлично – « 5 » | 15 – 13 баллов |
| Хорошо – « 4 » | 12 – 11 баллов |
| Удовлетворительно – « 3 » | 10 – 8 баллов |
| Неудовлетворительно – « 2 » | 7 и менее 7 баллов |