МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОСТОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ
«РОСТОВСКИЙ-НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ»

Фонд оценочных средств

по учебной дисциплине

ОП.07 «Основы проектирования баз данных»

программы подготовки специалистов среднего звена

по специальности

09.02.13 «Интеграция решений с применением технологий искусственного интеллекта»

(базовой подготовки)

г. Ростов-на-Дону, 2026 г.

РАССМОТРЕНО
на заседании цикловой комиссии
Программирование
Протокол № 1 от 31.08.2026 года
Председатель ЦК Сулавко С.Н.

УТВЕРЖДАЮ:
Зам. директора по УМР
И.В. Подцатова
«16» ноября 2026 г.

Фонд оценочных средств разработан на основе рабочей программы дисциплины ОП.07 «Основы проектирования баз данных», разработанной по специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта».

Разработчик: Кротенко Е.М., Малая М.А. – преподаватели государственного бюджетного профессионального образовательного учреждения Ростовской области «Ростовский-на-Дону колледж связи и информатики».

СОДЕРЖАНИЕ

1. Паспорт фонда оценочных средств 4
2. Результаты освоения дисциплины, подлежащие проверке 6
3. Оценка освоения учебной дисциплины 10
4. Контрольно-оценочные материалы для промежуточной аттестации по учебной дисциплине. 101

1. Паспорт фонда оценочных средств

В результате освоения учебной дисциплины ОП.07 «Основы проектирования баз данных» обучающийся должен обладать предусмотренными ФГОС СПО по специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта» следующими умениями, знаниями и владеть навыками:

Код ОК, ПК	**Уметь**	**Знать**	**Владеть навыками**
ОК.01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	Методы и подходы решения задач профессиональной деятельности	-
ОК.02	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии	Основы информационных технологий, методы анализа и интерпретации данных	-
ОК.03	Планировать и реализовывать профессиональное и личностное развитие, использовать знания правовой и финансовой грамотности	Основы предпринимательства, правовой и финансовой грамотности, подходы к личностному развитию	-
ОК.04	Эффективно взаимодействовать и работать в коллективе и команде	Основы командной работы, принципы эффективного взаимодействия	-
ОК.05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации	Особенности государственного языка Российской Федерации, правила деловой коммуникации	-
ОК.06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение	Основы духовно-нравственных ценностей, принципы антикоррупционного поведения	-
ОК.07	Содействовать сохранению окружающей среды, эффективно действовать в чрезвычайных ситуациях	Основы экологии, принципы бережливого производства, методы действий в ЧС	-
ОК.08	Использовать средства физической культуры для поддержания здоровья	Основы физической культуры и здоровья, методы поддержания физической формы	-
ОК.09	Пользоваться профессиональной документацией на государственном и иностранном языках	Основы ведения профессиональной документации на разных языках	-

ПК 2.1.	– Производить идентификацию проблем, связанных с нормальным функционированием базы данных; – Принимать решения по локализации проблем, связанных с нормальным функционированием базы данных; – Документировать внештатные ситуации связанные с нормальным функционированием базы данных;	– Основные коды ошибок при работе с базой данных; Методы и средства устранения ошибок, возникающих при работе с базой данных;	– Идентификации проблем, связанных с нормальным функционированием базы данных; Восстановления системы.
ПК 2.4.	Производить формирование требований к обработке данных и их извлечению;	Формы документов, необходимых для формирования, ведения и использования банка данных	Подготовки документации по формированию требований хранилищ банка данных
ПК 2.5.	– Добавлять, удалять и изменять данные в базе данных; Производить операции по импорту и экспорту данных в различных форматах	Типы данных хранения информации в базе данных	Проектирования, разработки и эксплуатации баз данных
ПК 2.7	– Выбирать критерии оптимизации выполнения запросов к БД; – Производить статистический анализ запросов к БД; – Классифицировать запросы к БД по видам и по результатам статистического анализа; – Применять языки и системы программирования БД для оптимизации выполнения запросов	– Модели и структуры данных, физические модели БД; – Особенности реализации структуры данных и управления данными в БД; – Языки и системы программирования БД; – Основы статистического анализа – Виды поисковых запросов	– Назначение приоритета запросов к БД; – Оптимизация выполнения статистически значимых запросов к БД
ПК 2.8	– Осуществлять коммуникации с заинтересованными сторонами; – Использовать выбранную среду программирования и средства системы управления базами данных; Применять выбранные	– Методологии и технологии проектирования и использования баз данных; – Особенности выбранной среды программирования и системы управления базами данных; Синтаксис выбранного	– Создание программного кода в соответствии с техническим заданием (готовыми спецификациями); – Оптимизация программного кода с использованием специализированных программных средств

	языки программирования для написания программного кода	языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования	

Формой аттестации в соответствии с рабочей программой по учебной дисциплине ОП.07 «Основы проектирования баз данных» является экзамен.

2. Результаты освоения учебной дисциплины, подлежащие проверке

2.1. В результате аттестации по учебной дисциплине ОП.07 «Основы проектирования баз данных» осуществляется комплексная проверка следующих умений и знаний:

Таблица 1

Результаты обучения	Показатели освоенности компетенций	Методы оценки
Знает: − Методы и подходы решения задач профессиональной деятельности − Основы информационных технологий, методы анализа и интерпретации данных − Основы предпринимательства, правовой и финансовой грамотности, подходы к личностному развитию − Основы командной работы, принципы эффективного взаимодействия − Особенности государственного языка Российской Федерации, правила деловой коммуникации − Основы духовно-нравственных ценностей, принципы антикоррупционного поведения − Основы экологии, принципы бережливого производства, методы действий в ЧС − Основы физической культуры и здоровья, методы	Демонстрирует знания: − методов и подходов решения задач профессиональной деятельности; − основ информационных технологий, методов анализа и интерпретации данных; − основ предпринимательства, правовой и финансовой грамотности, подходов к личностному развитию; − основ командной работы, принципов эффективного взаимодействия; − особенностей государственного языка Российской Федерации, правил деловой коммуникации; − основ духовно-нравственных ценностей, принципов антикоррупционного поведения; − основ экологии, принципов бережливого производства, методов действий в чрезвычайных ситуациях; − основ физической культуры и здоровья, методов поддержания физической формы; − основ ведения	*Экспертное наблюдение выполнения практических работ и видов работ по практике* *Диагностика (письменная проверочная работа)* *Экзамен*

поддержания физической формы − Основы ведения профессиональной документации на разных языках − Основные коды ошибок при работе с базой данных; − Методы и средства устранения ошибок, возникающих при работе с базой данных; − Формы документов, необходимых для формирования, ведения и использования банка данных − Типы данных хранения информации в базе данных − Модели и структуры данных, физические модели БД; − Особенности реализации структуры данных и управления данными в БД; − Языки и системы программирования БД; − Основы статистического анализа − Виды поисковых запросов − Методологии и технологии проектирования и использования баз данных; − Особенности выбранной среды программирования и системы управления базами данных; − Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования Умеет: − Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам − Использовать современные средства поиска, анализа и интерпретации информации, и	профессиональной документации на разных языках; − основных кодов ошибок при работе с базами данных; − методов и средств устранения ошибок, возникающих при работе с базами данных; − форм документов, необходимых для формирования, ведения и использования банка данных; − типов данных, используемых для хранения информации в базах данных; − моделей и структур данных, физических моделей баз данных; − особенностей реализации структур данных и управления данными в базах данных; − языков и систем программирования баз данных; − основ статистического анализа; − видов поисковых запросов; − методологий и технологий проектирования и использования баз данных; − особенностей выбранной среды программирования и системы управления базами данных; − синтаксиса выбранного языка программирования, особенностей программирования на данном языке и его стандартных библиотек. Демонстрирует умения: − выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; − использовать современные средства поиска, анализа и интерпретации информации, а также информационные технологии; − планировать и реализовывать профессиональное и личностное развитие, применять знания правовой и финансовой	

информационные технологии − Планировать и реализовывать профессиональное и личностное развитие, использовать знания правовой и финансовой грамотности − Эффективно взаимодействовать и работать в коллективе и команде − Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации − Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение − Содействовать сохранению окружающей среды, эффективно действовать в чрезвычайных ситуациях − Использовать средства физической культуры для поддержания здоровья − Пользоваться профессиональной документацией на государственном и иностранном языках − Производить идентификацию проблем, связанных с нормальным функционированием базы данных; − Принимать решения по локализации проблем, связанных с нормальным функционированием базы данных; − Документировать внештатные ситуации связанные с нормальным функционированием базы данных; − Производить формирование требований к обработке данных и их извлечению; − Добавлять, удалять и изменять данные в базе данных; − Производить операции	грамотности; − эффективно взаимодействовать и работать в коллективе и команде; − осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации; − проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение; − содействовать сохранению окружающей среды, эффективно действовать в чрезвычайных ситуациях; − использовать средства физической культуры для поддержания здоровья; − пользоваться профессиональной документацией на государственном и иностранном языках; − выявлять и идентифицировать проблемы, связанные с нормальным функционированием баз данных; − принимать решения по локализации проблем, возникающих при функционировании баз данных; − документировать внештатные ситуации, связанные с функционированием баз данных; − формировать требования к обработке и извлечению данных; − добавлять, удалять и изменять данные в базах данных; − выполнять операции импорта и экспорта данных в различных форматах; − выбирать критерии оптимизации выполнения запросов к базам данных; − проводить статистический анализ запросов к базам данных; − классифицировать запросы к базам данных по видам и результатам статистического анализа; − применять языки и системы программирования баз данных для	

по импорту и экспорту данных в различных форматах − Выбирать критерии оптимизации выполнения запросов к БД; − Производить статистический анализ запросов к БД; − Классифицировать запросы к БД по видам и по результатам статистического анализа; − Применять языки и системы программирования БД для оптимизации выполнения запросов − Осуществлять коммуникации с заинтересованными сторонами; − Использовать выбранную среду программирования и средства системы управления базами данных; − Применять выбранные языки программирования для написания программного кода	оптимизации выполнения запросов; − осуществлять коммуникацию с заинтересованными сторонами; − использовать выбранную среду программирования и средства системы управления базами данных; − применять выбранные языки программирования для написания программного кода.	

3. Оценка освоения учебной дисциплины:
3.1. Формы и методы оценивания

Предметом оценки служат умения и знания, предусмотренные рабочей программой по дисциплине ОП.07 «Основы проектирования баз данных».

Технология оценки знаний и умений по дисциплине соответствует Положению о формах, периодичности и порядке проведения текущего контроля успеваемости и промежуточной аттестации обучающихся в ГБПОУ РО «РКСИ».

Контроль и оценка освоения учебной дисциплины по темам (разделам)

Таблица 2

Элемент учебной дисциплины	**Текущий контроль**		**Промежуточная аттестация**	
	Форма контроля (по темам)	**Проверяемые ОК, ПК**	**Форма контроля (**по разделам**)**	**Проверяемые ОК, ПК**
Раздел 1 Основы проектирования и создания баз данных				
Тема 1.1 Введение в базы данных	*– письменная проверочная работа;* *– практическая проверка (защита практических работ №1-2).*	*ОК.01 – 09* *ПК 2.1,* *ПК 2.4 – 2.5,* *ПК 2.7 – 2.8*		
Тема 1.2 Концептуальное и логическое проектирование баз данных	*– письменная проверочная работа;* *– практическая проверка (защита практических работ №3-6).*	*ОК.01 – 09,* *ПК 2.1,* *ПК 2.4 – 2.5,* *ПК 2.7 – 2.8*		
Тема 1.3 Физическое проектирование баз данных	*– письменная проверочная работа;* *– практическая проверка (защита практических работ №7-9).*	*ОК.01 – 09,* *ПК 2.1,* *ПК 2.4 – 2.5,* *ПК 2.7 – 2.8*		
Раздел 2 Основы работы с запросами и оптимизация баз данных				
Тема 2.1 Основы SQL и работа с запросами	*– письменная проверочная работа;* *практическая проверка (защита практических работ №10-12).*	*ОК.01 – 09,* *ПК 2.1,* *ПК 2.4 – 2.5,* *ПК 2.7 – 2.8*		
Тема 2.2 Оптимизация запросов	*– письменная проверочная работа;* *практическая проверка (защита практических работ №13-16).*	*ОК.01 – 09,* *ПК 2.1,* *ПК 2.4 – 2.5,* *ПК 2.7 – 2.8*		

Тема 2.3 Разрабатывать объекты баз данных NoSQL	*– письменная проверочная работа;* *практическая проверка (защита практических работ №17).*	*ОК.01 – 09,* *ПК 2.1,* *ПК 2.4 – 2.5,* *ПК 2.7 – 2.8*		
			Экзамен	*ОК.01 – 09,* *ПК 2.1,* *ПК 2.4 – 2.5,* *ПК 2.7 – 2.8*

3.2. Задания для оценки освоения учебной дисциплины

3.2.1. Задания для оценки знаний и умений в ходе проведения текущего контроля

Темы 1.1
Основные понятия баз данных

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Текст задания по теоретической части 1

Ответьте на вопросы.

Вариант 1

1 Что такое база данных?

2 Что такое объект базы данных? Назовите виды и состояния объектов.

3 Что такое ключевое поле? Дайте определения существующим видам ключей.

4 Что подразумевается под целостностью данных?

5 Что такое нормализация отношений? Расскажите о трёх нормальных формах

Вариант 2

1 Что такое таблица базы данных?

2 Что содержит класс объектов базы данных?

3 В чём назначение индексов? Какие существуют типы индексов?

4 В чем назначение администрирования баз данных? (описать задачи и функции администрирования базы данных)

5 Технология обслуживания базы данных: обеспечение работоспособности и восстановление данных. Основные методы и средства защиты данных в базах данных.

Вариант 3

1 Что такое атрибут объекта базы данных?

2 Что такое СУБД?

3 Какие существуют типы отношений между таблицами? Дайте определение каждому отношению и опишите алгоритм его построения.

4 Методы и технологии защиты информации в базах данных.

5 Организация непротиворечивости и целостности данных.

Вариант 4

1 В чём назначение домена в базах данных?

2 CASE-средства для проектирования баз данных (определение, назначение, примеры).

3 Какие существуют типы моделей данных? Опишите их.

4 Способы контроля доступа к данным и управления привилегиями.

5 Дайте определение и назначение следующим понятиям: форма, отчёты и запросы в СУБД.

Текст задания по теоретической части 2

Ответьте на вопросы.

Вариант 1

1) первичный ключ (определение, назначение);

2) суррогатный ключ (определение, назначение);

3) тип отношения «один к одному»(определение, алгоритм построения, как графически обозначается это отношение, какие ключи используются при построении);

4) тип отношения «многие ко многим»(определение, алгоритм построения, как графически обозначается это отношение, какие ключи используются при построении);

5) по каким ключам строятся связи в реляционной БД?

Вариант 2

1) внешний ключ (определение, назначение);
2) альтернативный (вторичный) ключ (определение, назначение);
3) тип отношения «один ко многим»(определение, алгоритм построения, как графически обозначается это отношение, какие ключи используются при построении);
4) индекс / индексация (определение, назначение);
5) сколько связей может быть у таблицы? От чего это зависит?

Критерий оценки

Каждый полный правильный ответ оценивается 1 баллом.
0-2 правильных ответа – «2»
3 правильных ответа – «3»
4 правильных ответа – «4»
5 правильных ответов – «5»

Практическое занятие № 1

Анализ структуры базы данных на примере реальной системы

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

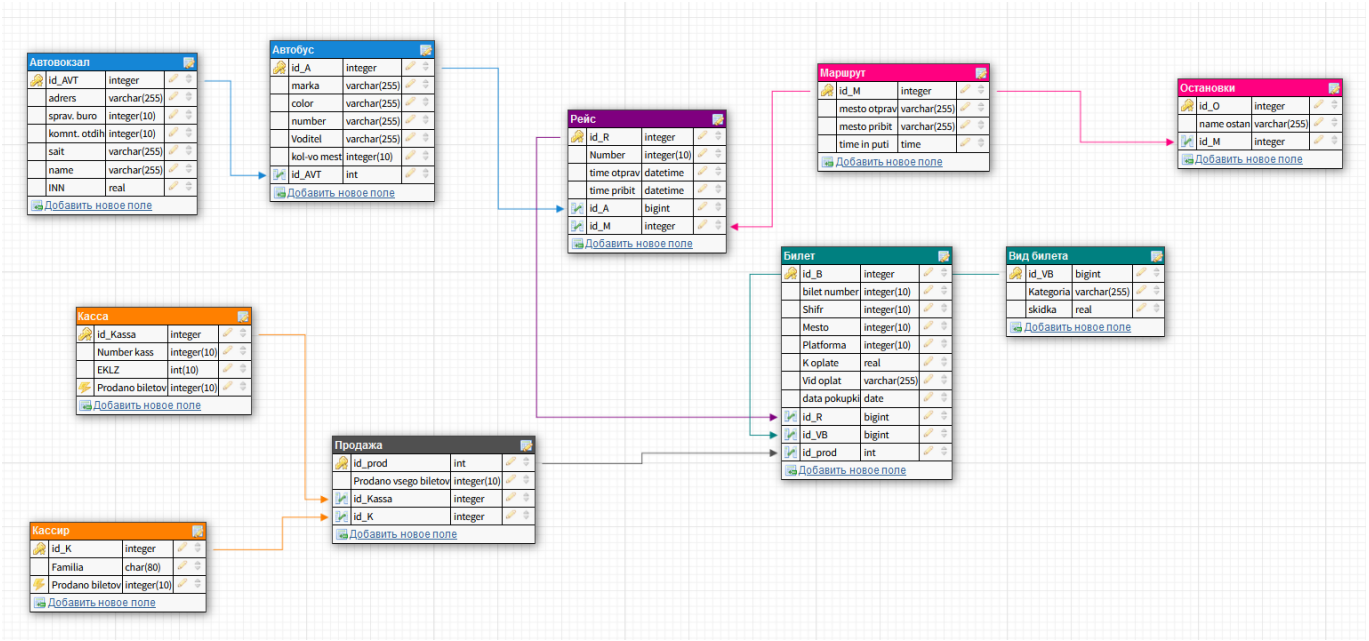
ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: Сделать анализ структуры базы данных «Автовокзал».

Этапы выполнения задания:

- определите в ER-модели базы данных родительские и дочерние объекты;
- опишите структуру каждого объекта с их атрибутами;
- выделите имеющиеся в объектах ключи и опишите их;
- опишите связи между таблицами с указанием типа связи.



Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №2

Построение ER-диаграммы для простой предметной области.

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: Изучить предметную область деятельности компании и сделать ER-диаграмму необходимой базы данных.

Описание предметной области:

Компания «Фортуна» занимается оптовой и розничной продажей автомобильных запчастей. Она имеет один большой склад, на который поступают запчасти и идёт поставка в розничный магазин для продажи.

Необходимо создать базу данных, которая позволит хранить следующую информацию:

- о самой компании;
- о складе компании;
- о магазине, в котором ведётся продажа;
- о категории товаров;
- о товарах;
- о поставках товара;
- о продажах товара.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Тема 1.2

Концептуальное и логическое проектирование баз данных

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Текст задания по теоретической части 1

Дайте определение каждой Нормальной Форме. Укажите аномалии этой формы. Опишите алгоритм устранения аномалии.

Вариант 1

1) 1 НФ
2) 2 НФ
3) 3НФ
4) 4 НФ
5) ДК НФ

Вариант 2

1) 1 НФ
2) 2 НФ
3) НФ Бойса-Кодда
4) НФ
5) НФ

Текст задания по теоретической части 2

Ответьте на вопросы.

Вариант 1

1) Дайте определение понятию целостность данных.
2) Дайте определение понятию непротиворечивости данных;
3) Опишите правила реализации целостности данных.
4) Что такое транзакции в БД?
5) Что подразумевает под собой ссылочная целостность на внешних ключах?

Вариант 2

1)
2) Дайте определение понятию целостность данных.
3) Что подразумевается под ограничением целостности данных?
4) Классификация ограничений целостности данных (перечислить эти ограничения).
5) Что подразумевает под собой ссылочная целостность на триггерах?

Критерий оценки

Каждый полный правильный ответ оценивается 1 баллом.
0-2 правильных ответа – «2»
3 правильных ответа – «3»
4 правильных ответа – «4»
5 правильных ответов – «5»

Текст задания по практической части

Практическое занятие №3

Определение сущностей и атрибутов для заданной предметной области

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: Определить сущности и атрибуты, с полной характеристикой, для предметной области «Фортуна».

Уточнённое описание предметной области «Фортуна»:

– о самой компании (название, адрес регистрации, ИНН, ФИО директора);

– о складе компании (адрес, телефон, ФИО начальника);

– о магазине, в котором ведётся продажа (адрес, телефон, ФИО директора);

– о товарах (название категории запчасти (использовать: **двигатель, дворники, кузо**в), процент надбавки к цене (зависит от категории товара), название запчасти, цена, Количество конкретной запчасти на складе;

– о поставках товара (дата поставки, количество каждого поставленного товара, общее количество поставленных товаров в поставке;

– о продажах товара (дата продажи, количество каждого проданного товара, общее количество проданных товаров в продаже).

База данных должна отвечать всем требованиям нормализации реляционной модели данных.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №4

Определение ключей и построение отношений

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: У сформированных в практическом задании 3 объектов определить ключевые поля необходимые для построения отношений и построить данные отношения.

Пошаговое выполнение работы:

- построить логические связи, т.е. определить на логическом уровне какие объекты и как будут между собой связаны согласной предметной области;
- при необходимости устранить аномалии в отношениях;
- в каждом объекте определить столбец, которому будет дана роль первичного ключа;
- добавить в необходимые объекты внешние ключи для связей;
- построить связи между объектами.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №5

Нормализация таблиц и отношений

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: Проведите нормализацию объектов из предметной области «Фортуна».

Для этого:

- выявить все имеющиеся аномалии в объектах;
- устранить аномалии, применяя правила нормализации;
- привести к 3НФ вновь полученные после нормализации объекты;
- связать сформированные после нормализации объекты с их родительскими объектами;
- определить есть ли аномалии в связях;
- устранить аномалии связей.

Работа выполняется по вариантам.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №6

Построение ER-диаграммы с учётом нормализации в CASE-средстве

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: Сформированную модель предметной области «Фортуна» реализовать в онлайн case–средстве dbdesigner.net.

Для этого:

− загрузить браузер и открыть сайт dbdesigner.net;

− создать в case–средстве базу данных и её таблицы, используя исходные данные из полученного раздаточного материала;

− выявить аномалии избыточности данных в таблицах;

− устранить аномалии;

− установить в таблицах необходимые ключи для реализации связей;

− связать таблицы между собой;

− определить есть ли аномалии в связях;

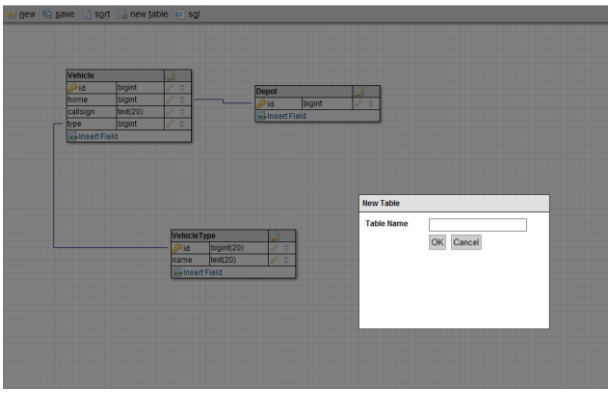
− если аномалии найдены – устранить их, если нет – показать работу преподавателю.

Работа выполняется по вариантам. Пример раздаточного материала смотрите выше.

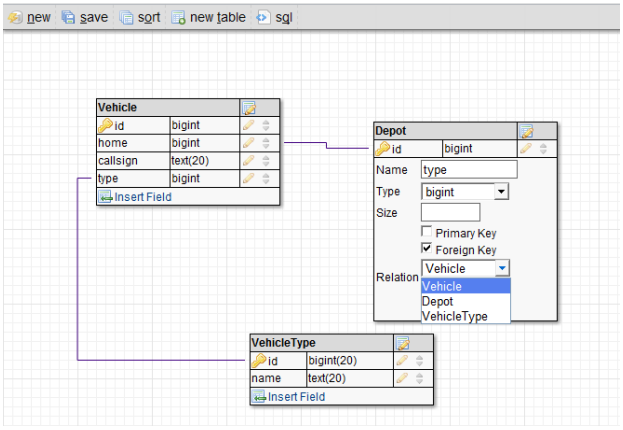
Пошаговая работа:

Создадим схему базы данных, используя генератор схем dbdesigner.net. Для начала нужно залогиниться.

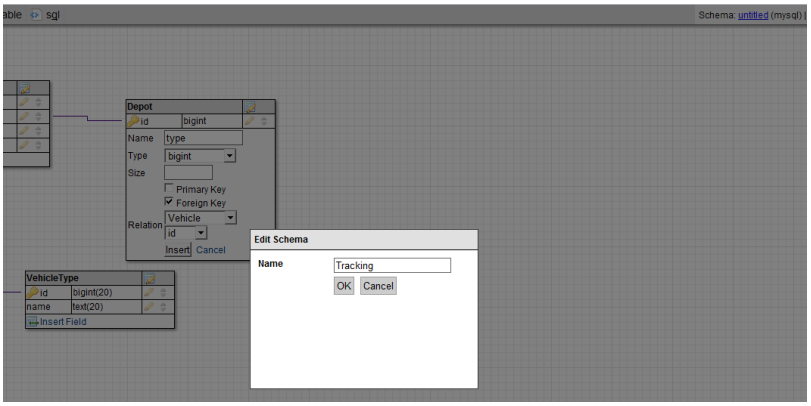
Затем задать имя базе данных. Затем создаем таблицу, нажав на соответствующую кнопку и задав ей имя.



Далее создаются поля в таблице. Поле может быть как простого типа, так и ссылкой на другую таблицу, если у нас их уже несколько.



Когда все готово задается схеме имя и сохраняем ее.



Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Тема 1.3
Физическое проектирование баз данных

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Текст задания по теоретической части

1) Что подразумевает под собой оптимизация структуры базы данны?
2) Для чего необходимо указывать тип данных атрибутам объекта?
3) Какие типы данных используется в SQL-СУБД?
4) Для чего создаются индексы в СУБД?
5) Как настроить хранилище данных?

Практическое занятие №7

Создание базы данных в СУБД на основе логической модели.

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

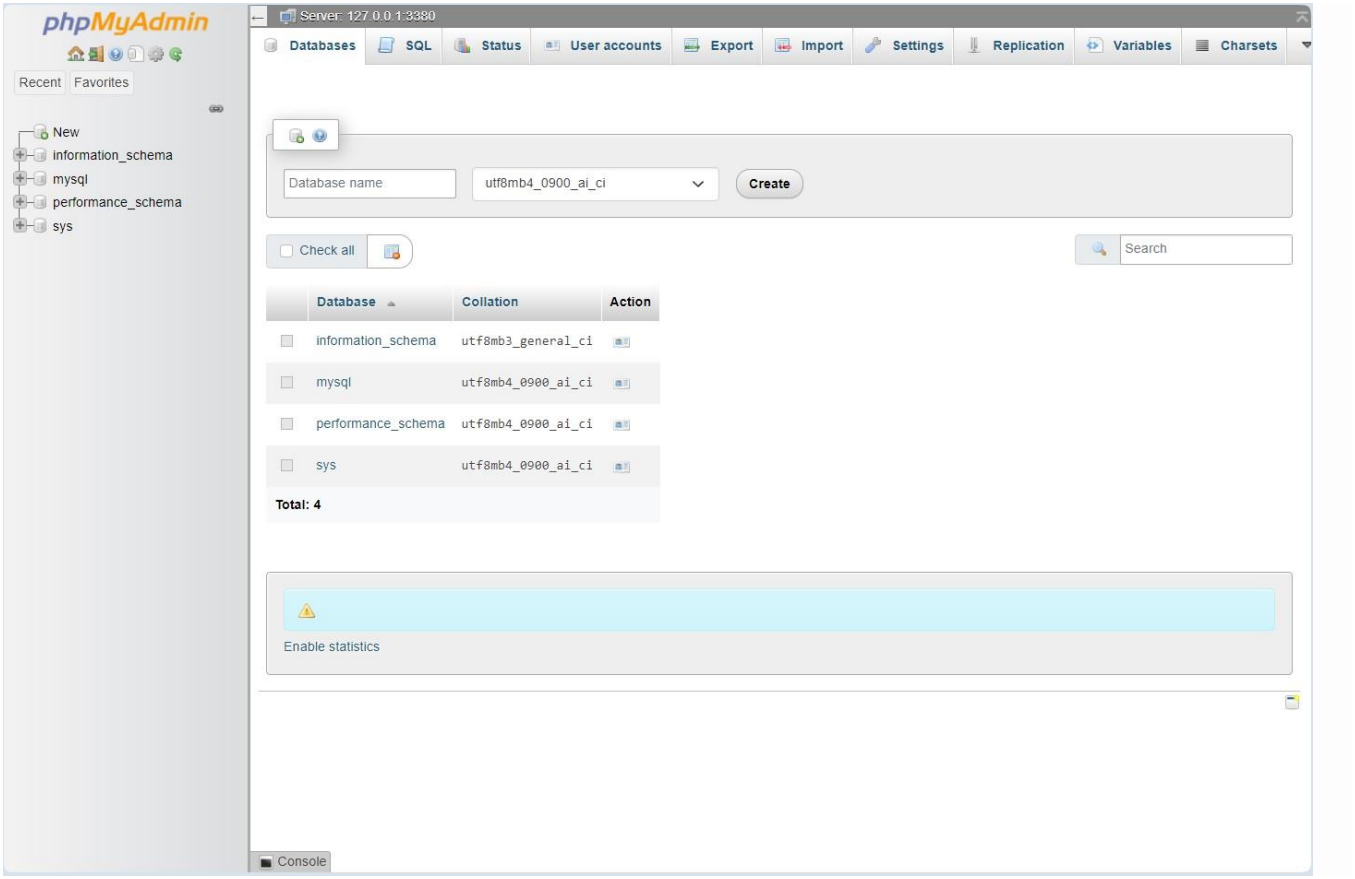
ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: Содать базу данных предметной области «Фортуна» в СУБД MySQL используя PHPMyAdmin.

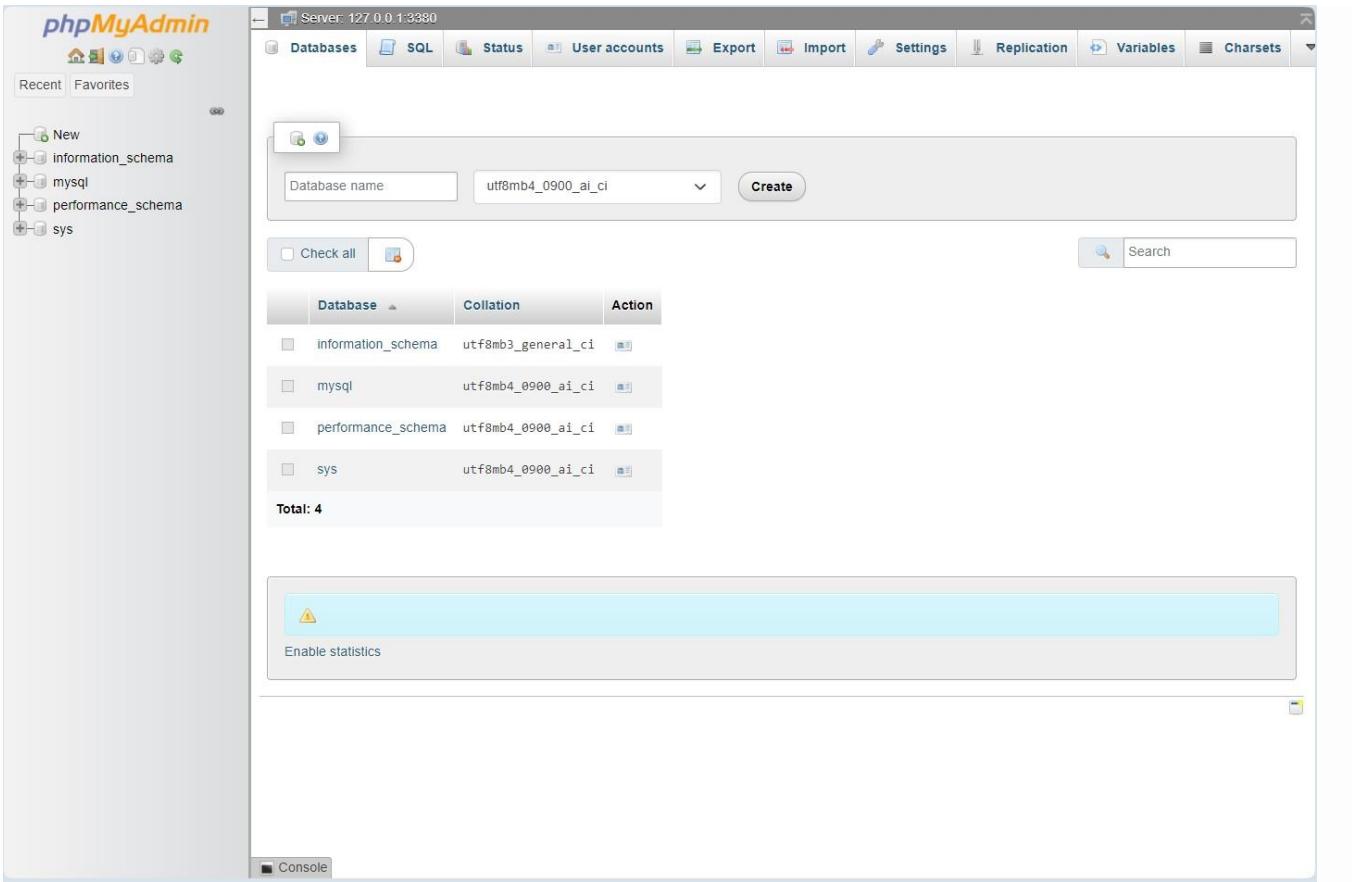
Ход выполнения.

Запустите PHPMyAdmin.

Перейдите на вкладку Databases.

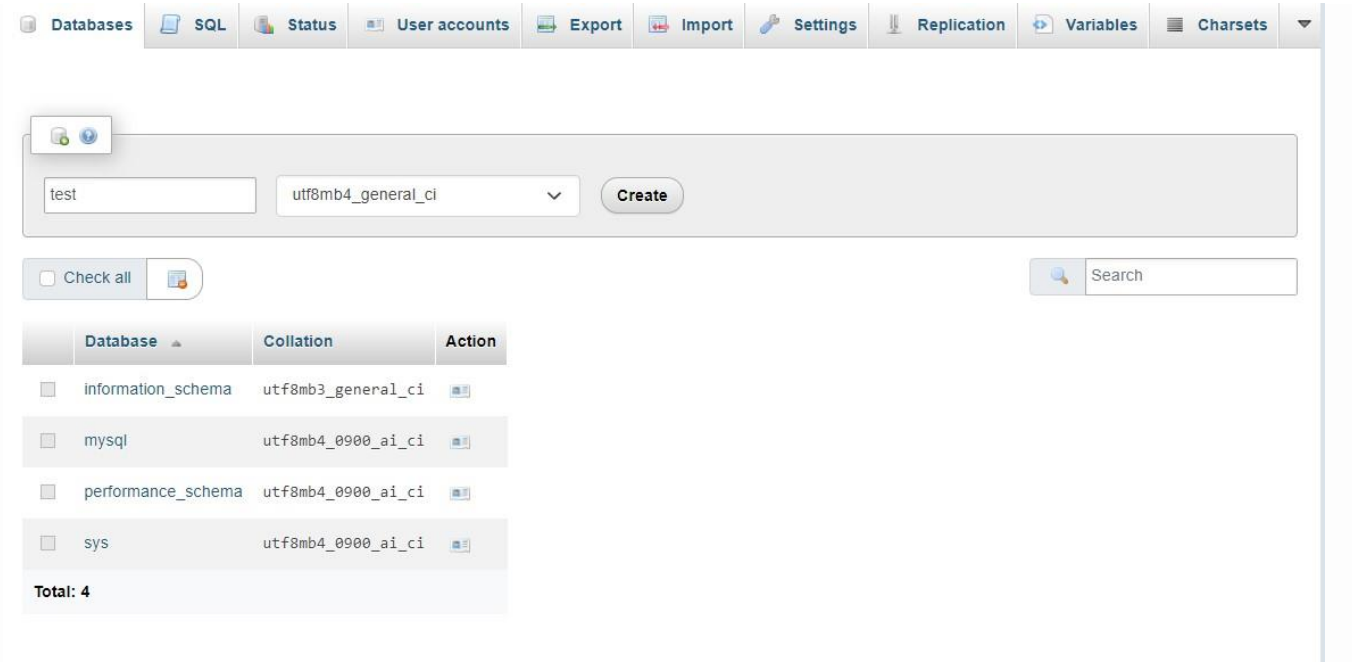


В этой вкладке находим пункт Database name.

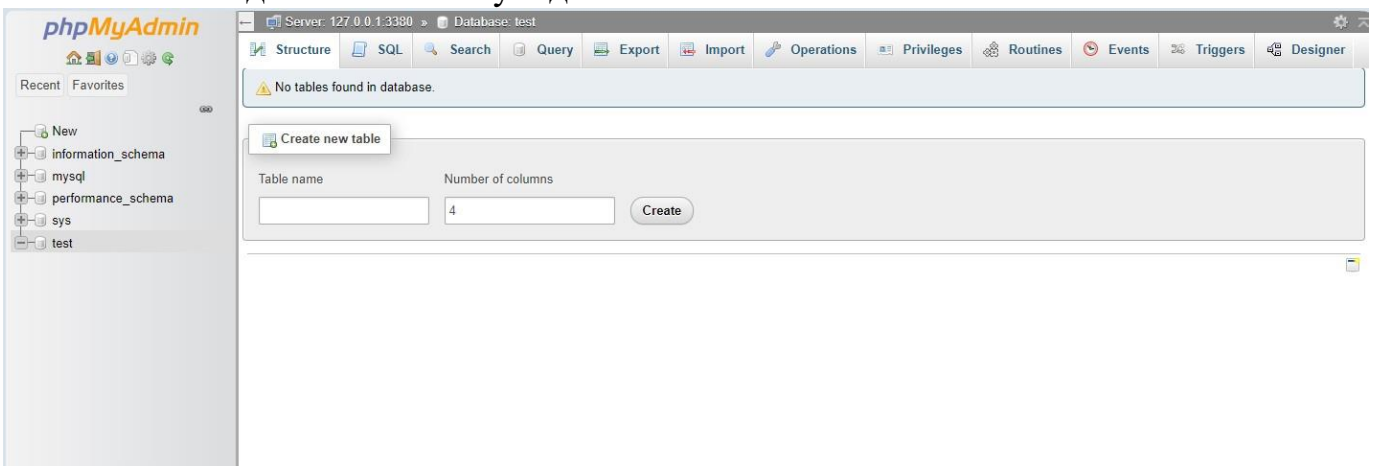


В инпут вводим имя нашей БД. Далее в соседнем оуне выбираем кодировку. По умолчанию в нем прописана кодировка utf8mb4_0900_ai_ci. Однако

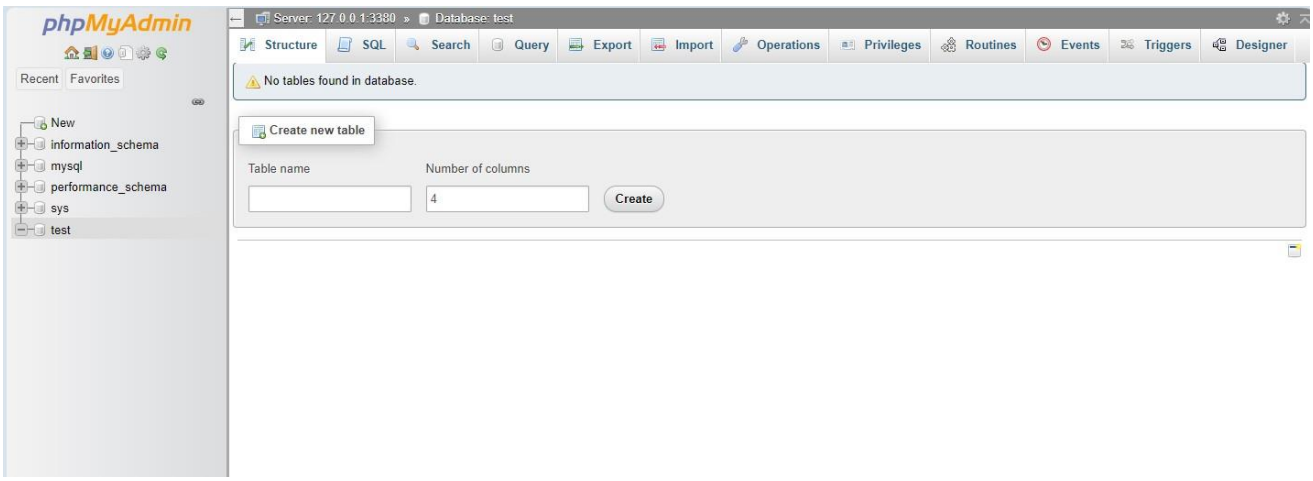
рекомендуется установить кодировку utf8mb4_general_ci, позволяющую работать не только с латиницей, но и с кириллицей. После чего нажимаем на кнопку create, находящуюся справа.



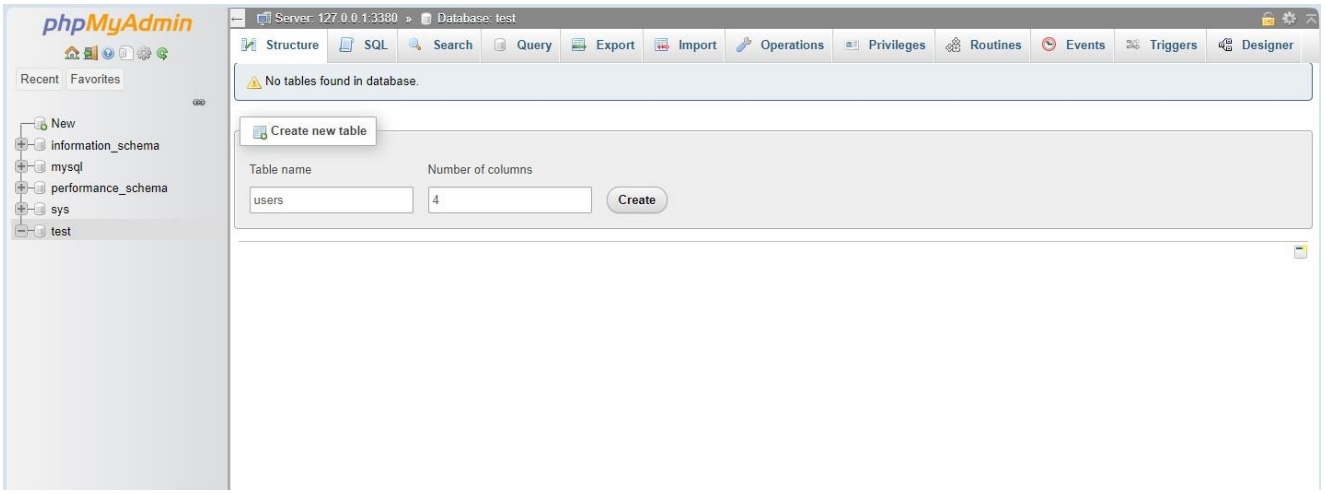
База создана. Слева вы увидите её имя.



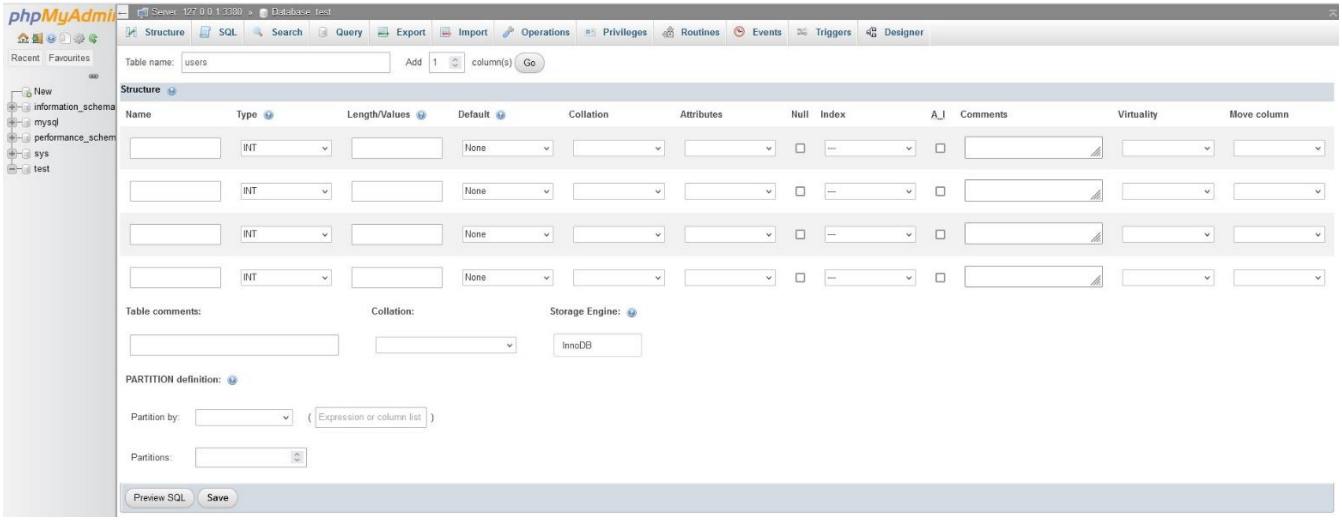
Теперь создадим в нашей БД users. Кликните по имени базы данных в левом сайдбаре, чтобы перейти в эту БД. Затем выберите вкладку Structure.



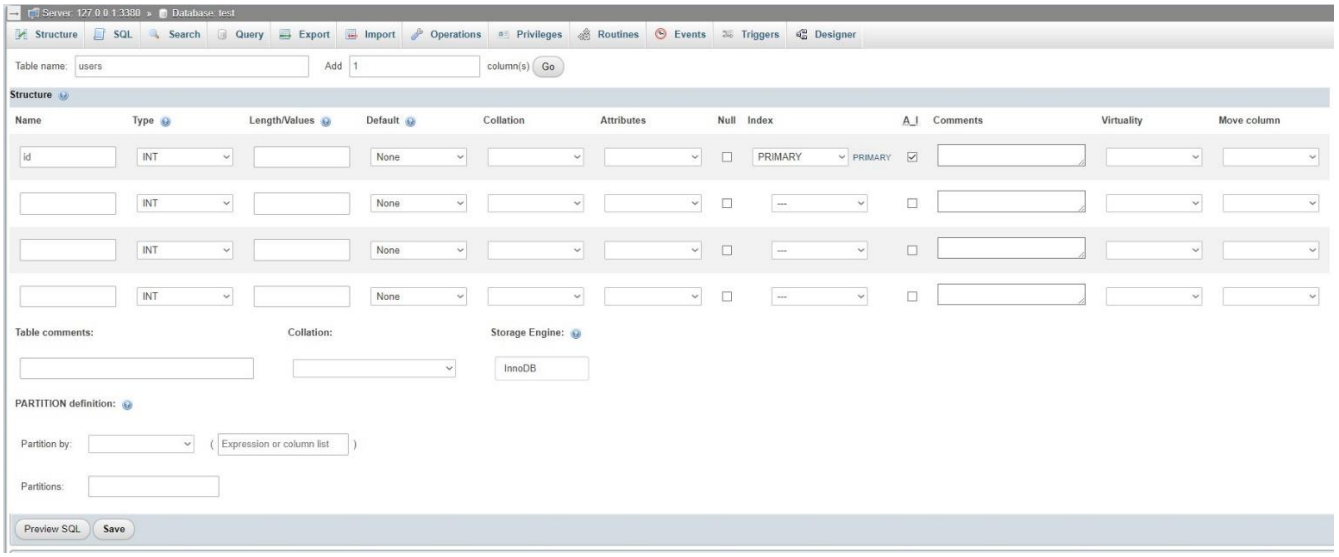
Далее переходим в секцию Create new table. В инпуте Table name вводим название таблицы - users. В инпуте справа можно задать количество колонок (полей) в таблице. Пока что нас устроит их дефолтное количество - 4. Для завершения создания таблицы нажимаем на кнопку Create.



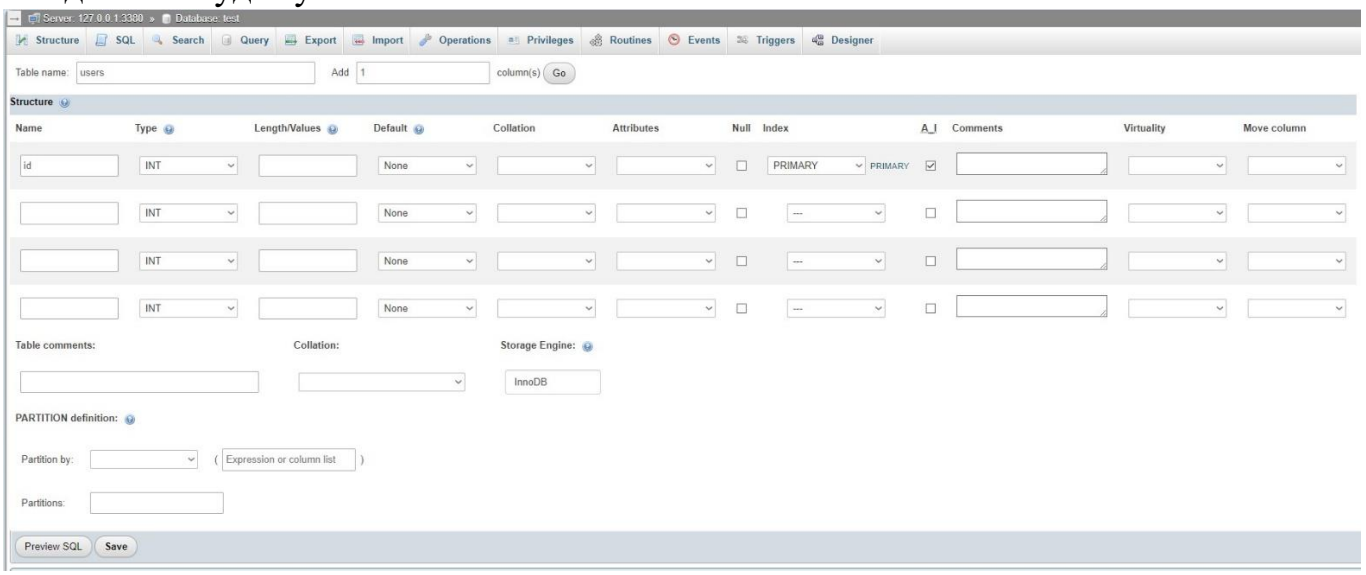
Теперь нам откроется окошко со структурой таблицы, в котором можно назначать имя и тип данных для полей.



Далее формируем столбцы будущей таблицы.
Первым полем будет `id`. Для него подходит числовой тип `INT`.

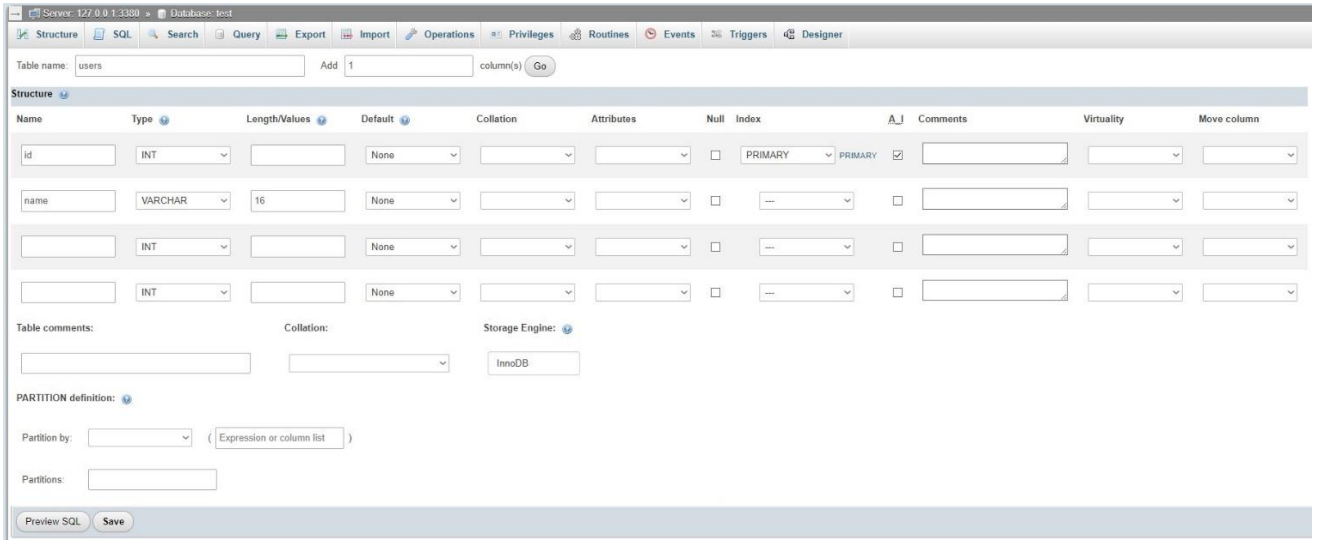


Далее ставим галочку в поле `AI` (`AUTO_INCREMENT`). Это нужно для автозаполнения поля `id` при вставке записей в таблицу. Благодаря этому нам не нужно будет прописывать `id` для каждой записи в таблице. Кроме того, отмеченный `AI` автоматически назначает для `id` тип `Index` в значении `PRIMARY`. Теперь каждое `id` будет уникальным.



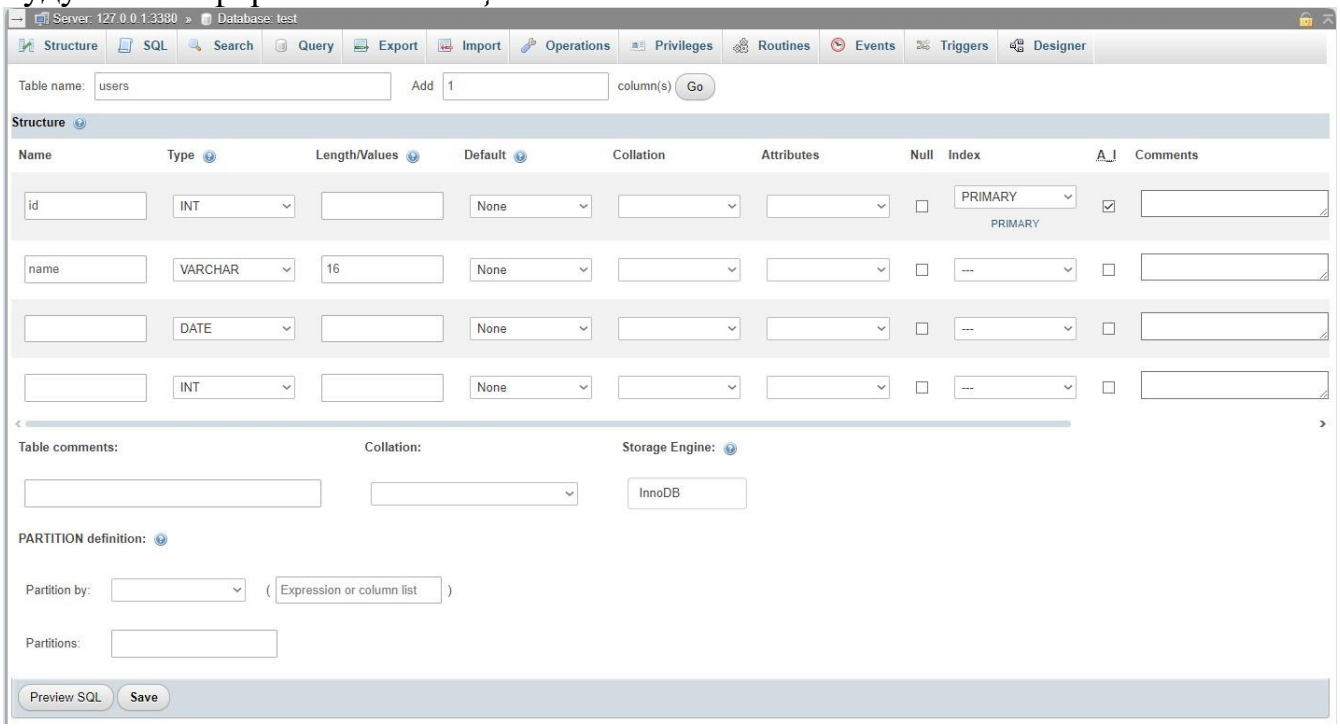
Пример: Тип данных VARCHAR

Теперь перейдем к полю `name`. Для него назначим тип `VARCHAR`, предназначенный для работы со строками. Для этого типа нужно обязательно указать длину строки. Пусть она будет равна `16` символам.



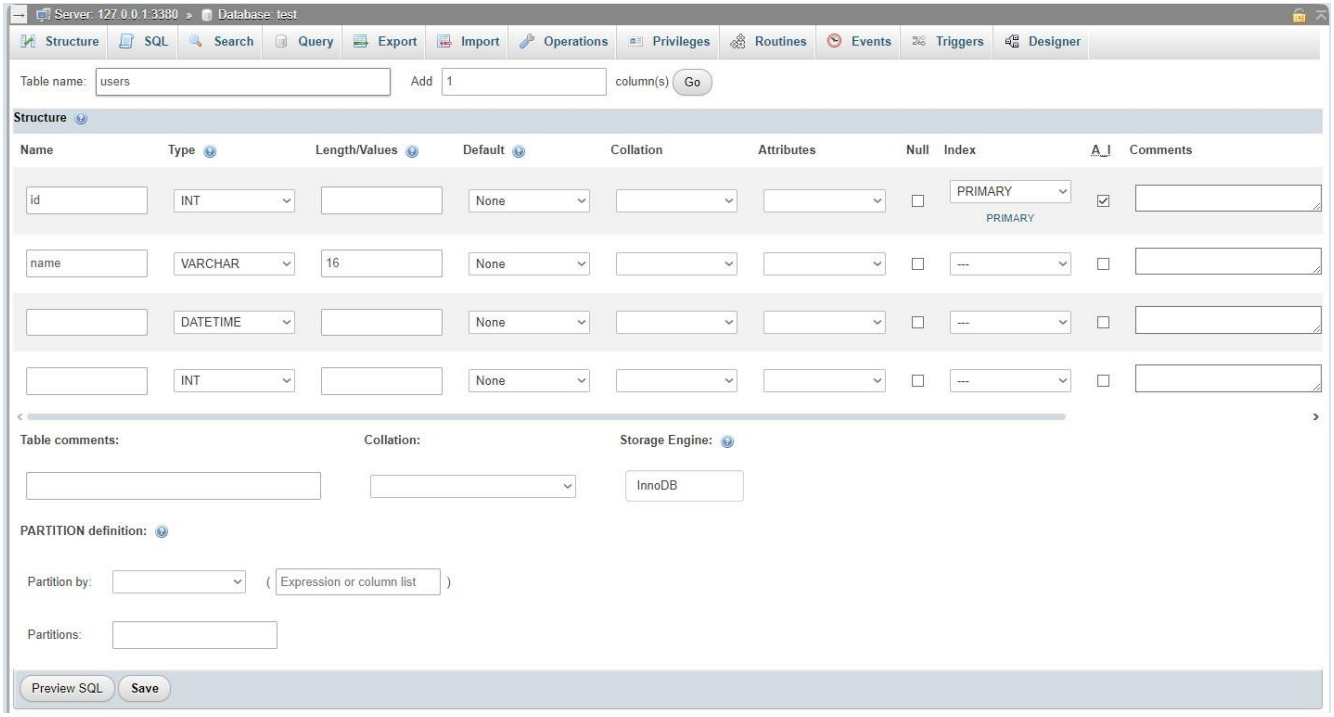
Пример: Тип данных DATE

Для работы с датами предусмотрен тип `DATE`. Все значения для данного типа будут иметь формат *год-месяц-день*.



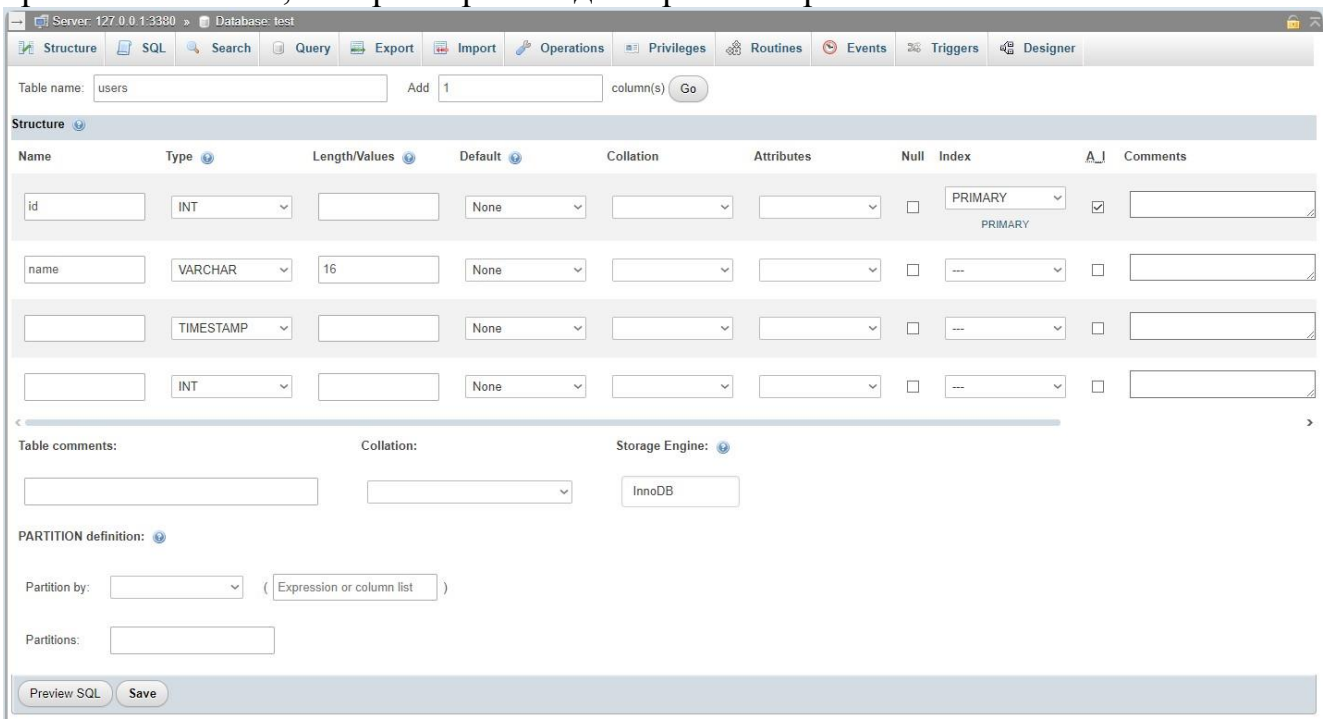
Пример: Тип данных DATETIME

Также для сохранения значений в формате *год-месяц-день часы:минуты:секунды* можно использовать тип `DATETIME`.



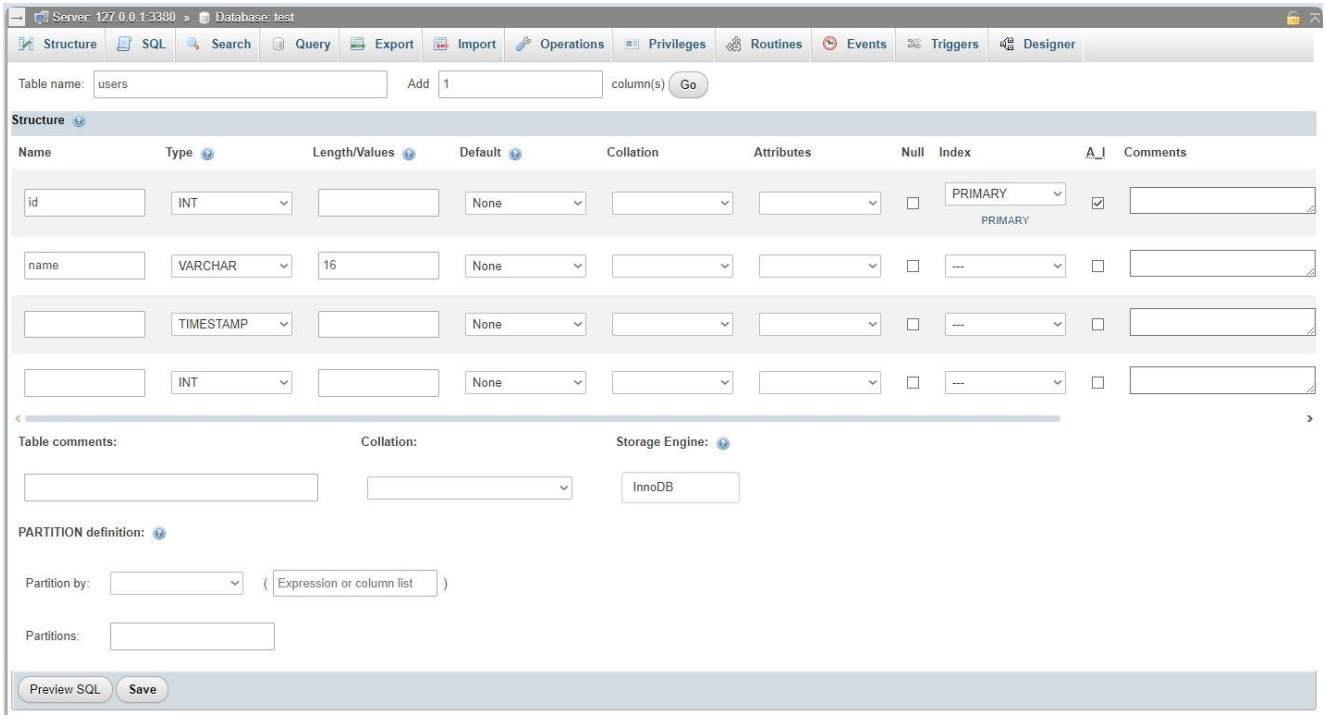
Пример: Тип данных TIMESTAMP

Тип `TIMESTAMP` хранит время как количество секунд, прошедших с 1970-01-01 00:00:00 UTC (так называемая Unix Epoch). Используется для отслеживания времени событий, которые происходят в разных временных зонах.

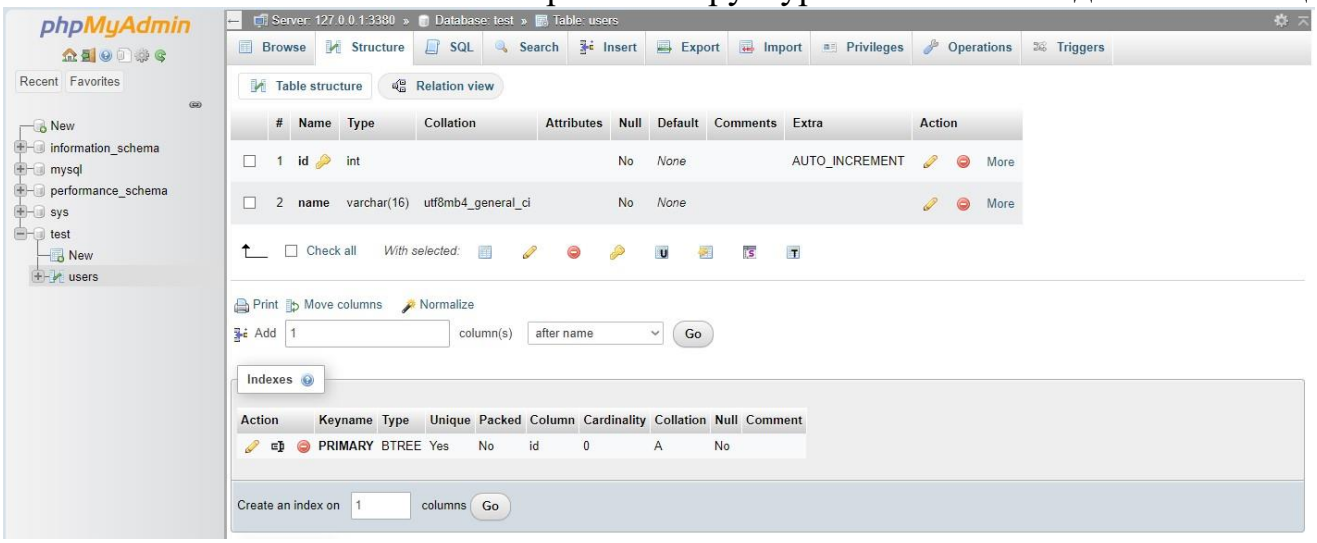


Пример: Завершаем создание таблицы

Теперь осталось завершить создание нашей таблицы. Для этого нажимаем на кнопку `Save`.

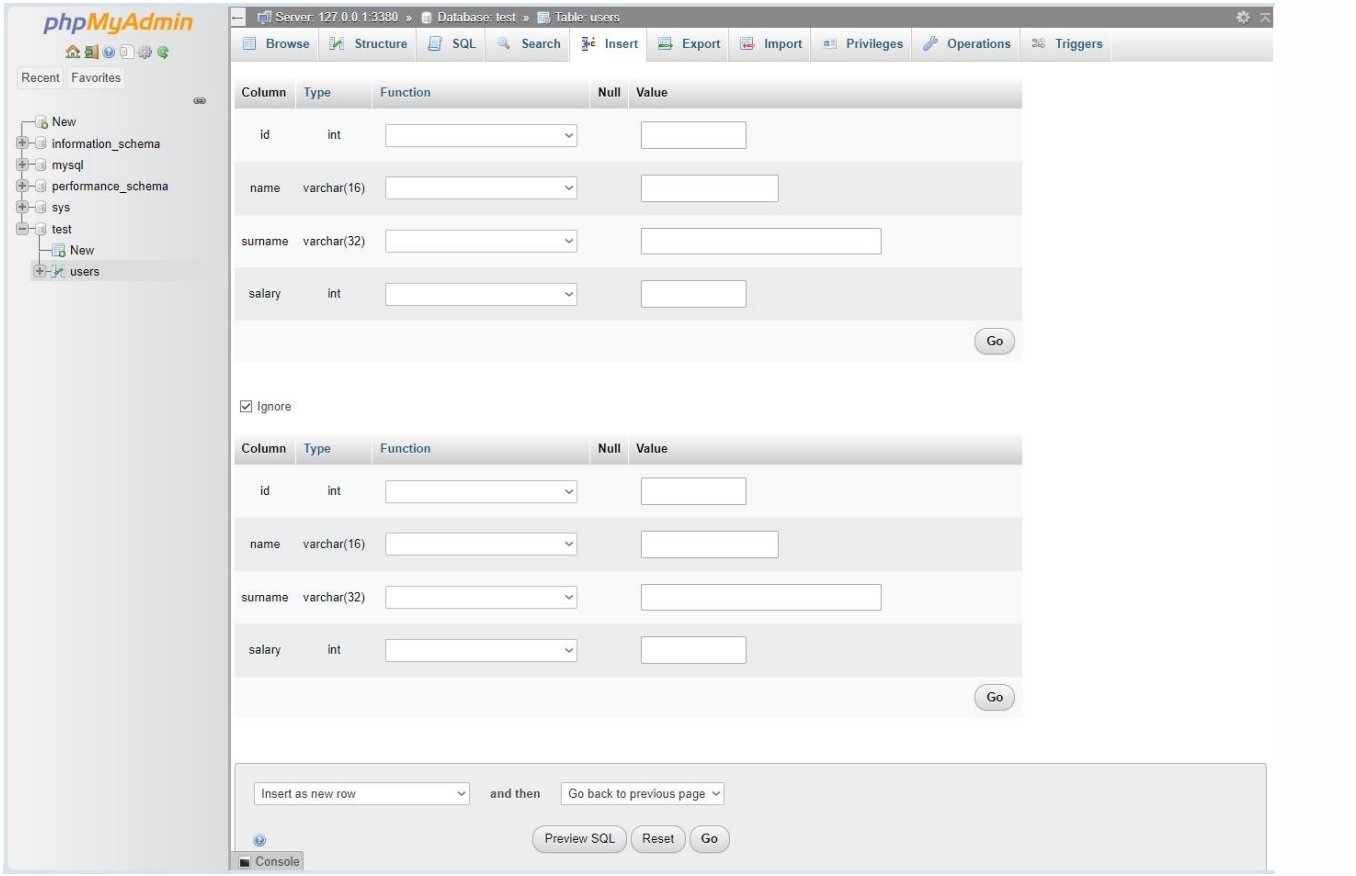


После этого в окошке отобразится структура только что созданной таблицы.

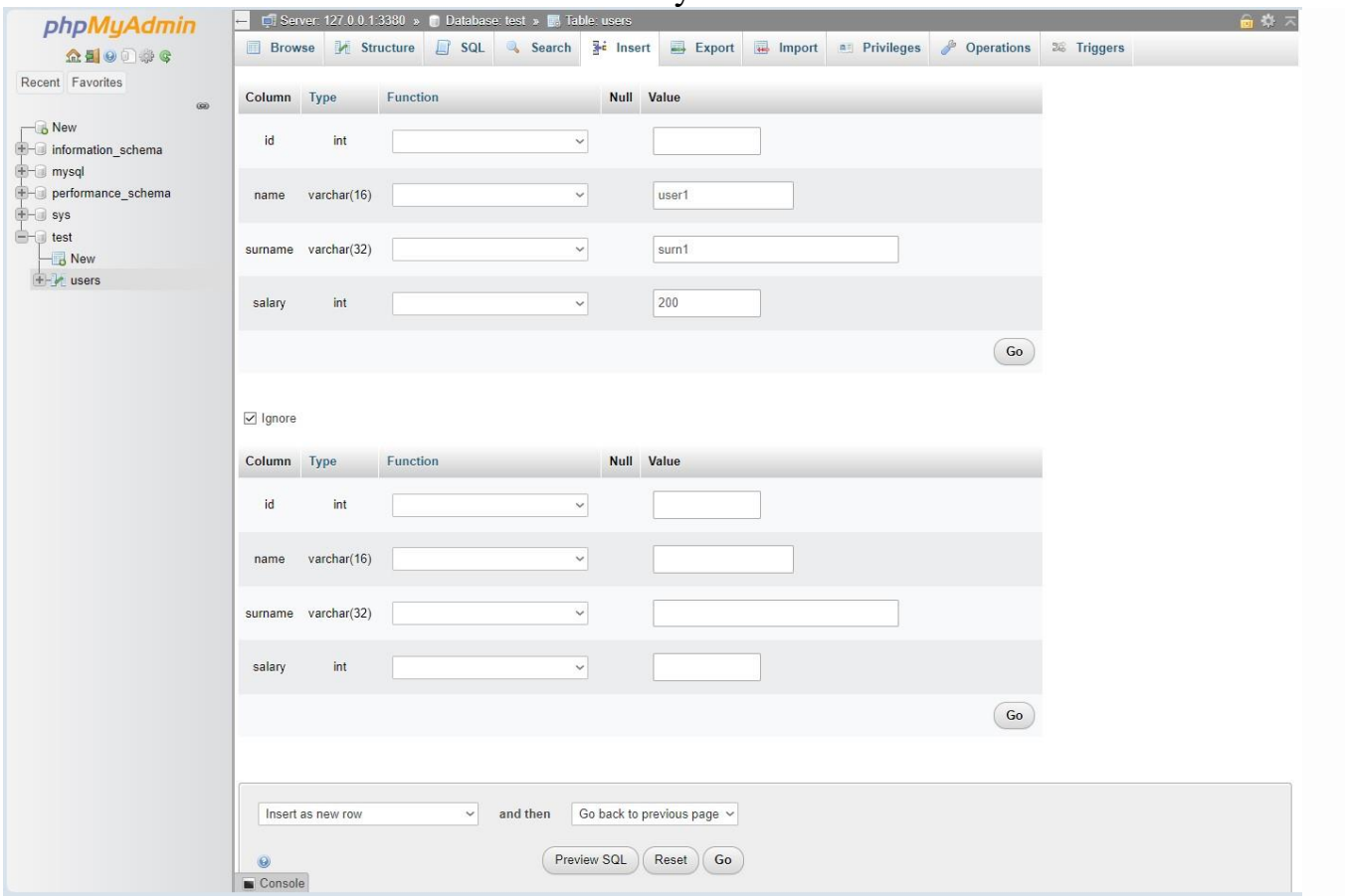


Добавление записей в таблицу в PhpMyAdmin

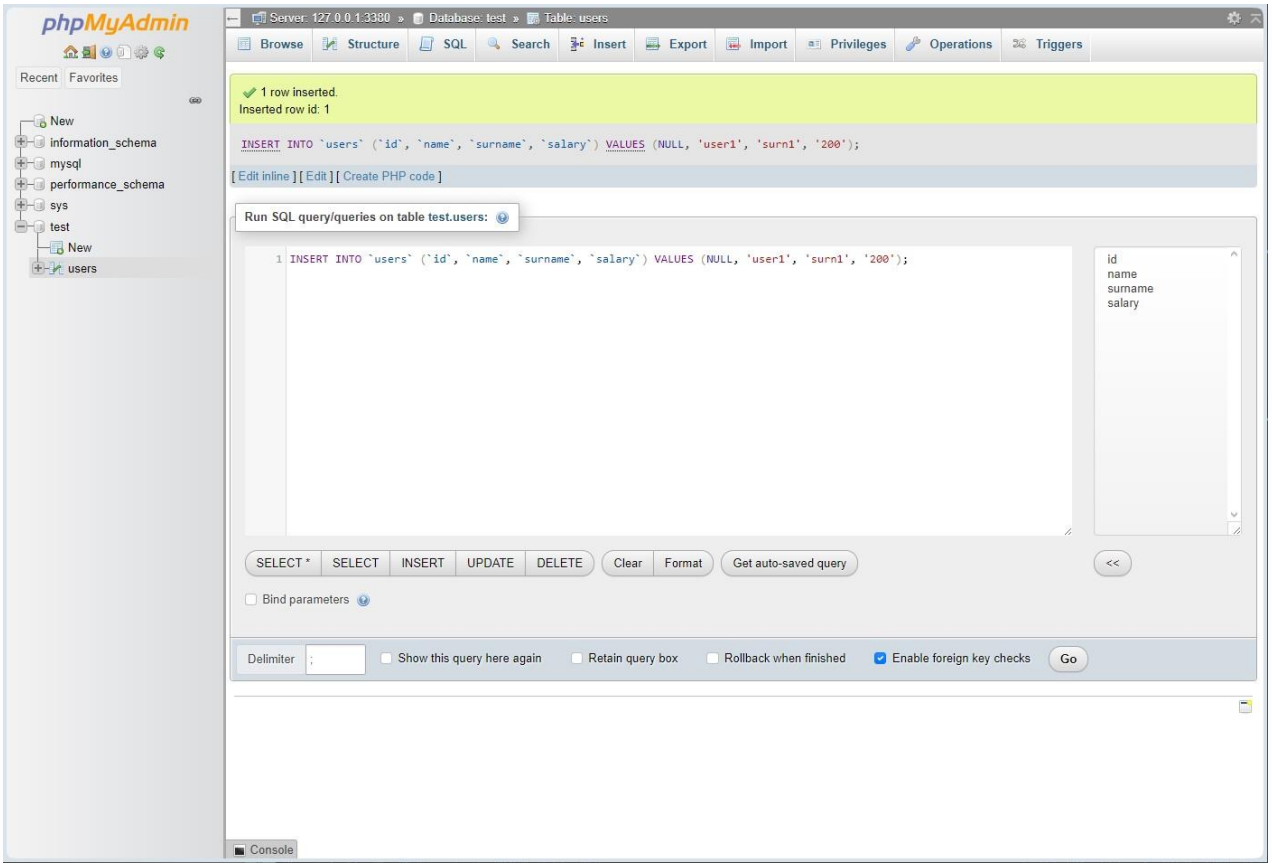
Давайте теперь добавим в нашу таблицу запись. Для этого переходим во вкладку `Insert`.



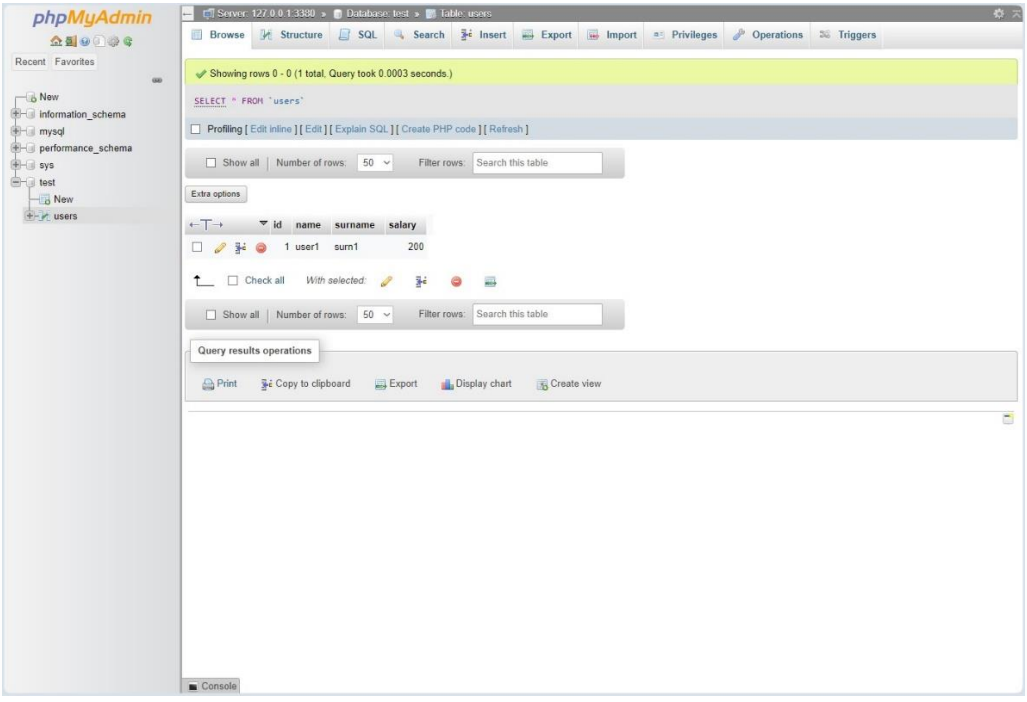
В этой вкладке заполняем инпуты. При этом инпут с id можно пропустить, поскольку все его значения будут добавляться автоматически. Для подтверждения внесенных изменений нажимаем на кнопку Go.



После добавления записи вы увидите страничку с выполненным SQL запросом.



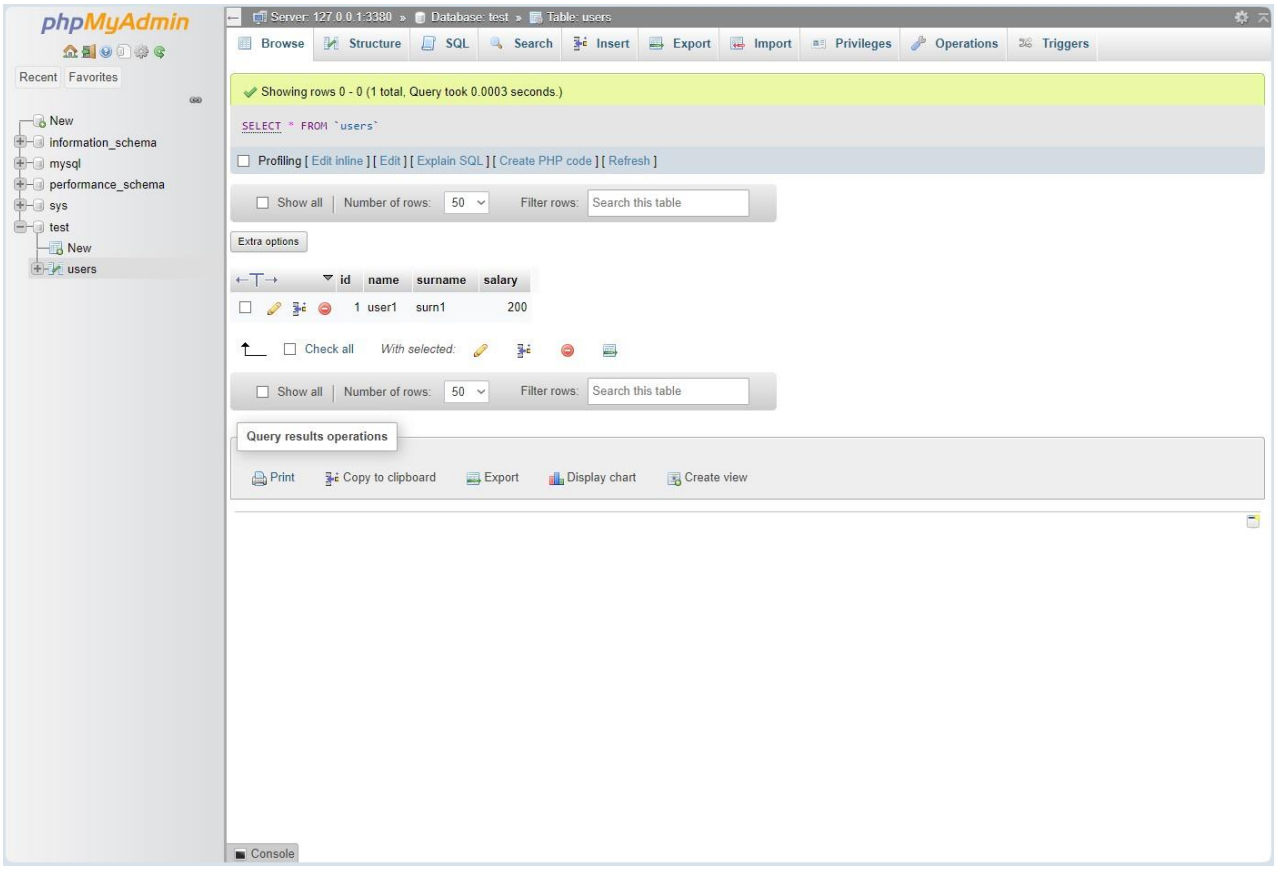
Давайте посмотрим на содержимое нашей таблицы во вкладке `Browse`.



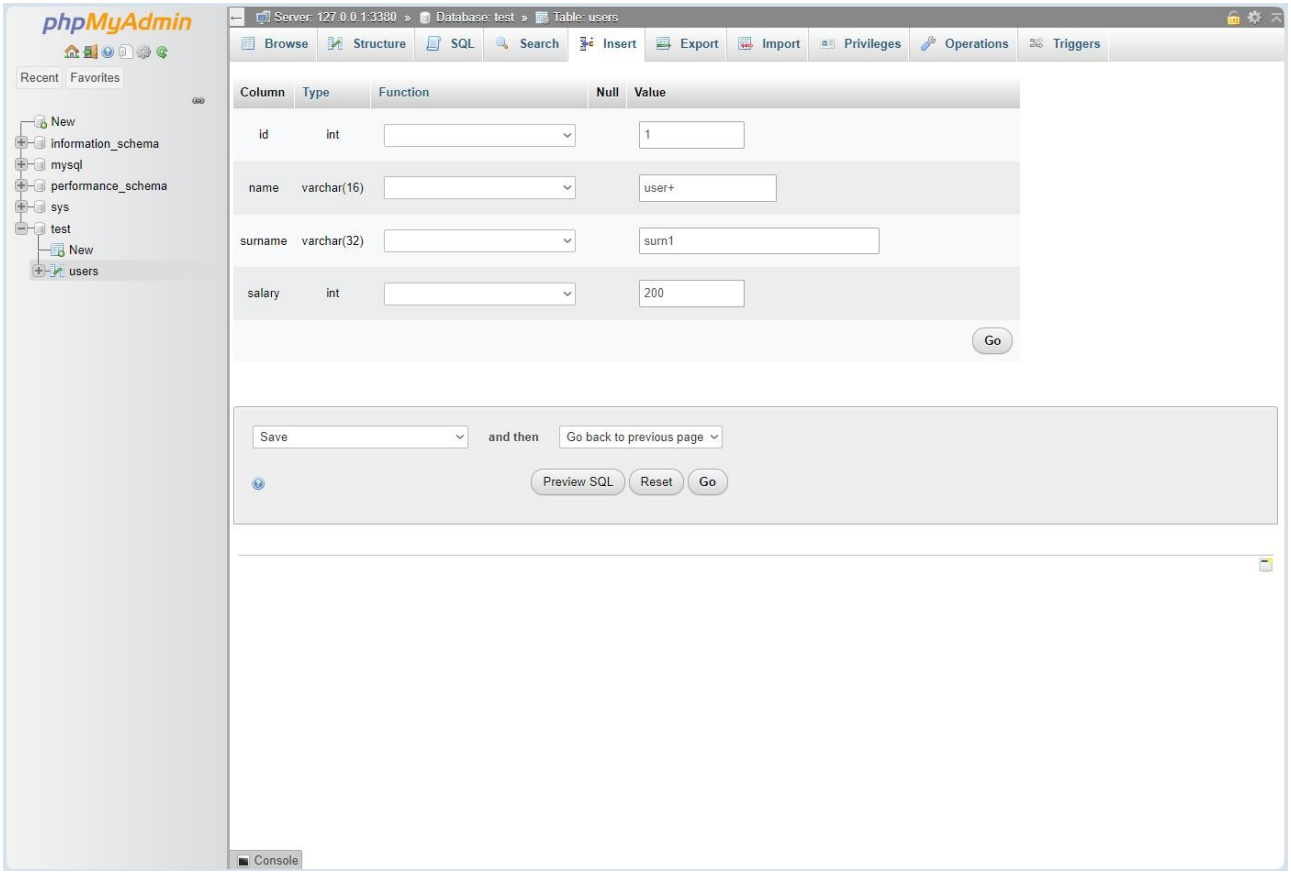
Редактировать записи в PMA можно использовать с помощью двух вариантов. Ниже рассмотрим их подробнее.

Пример

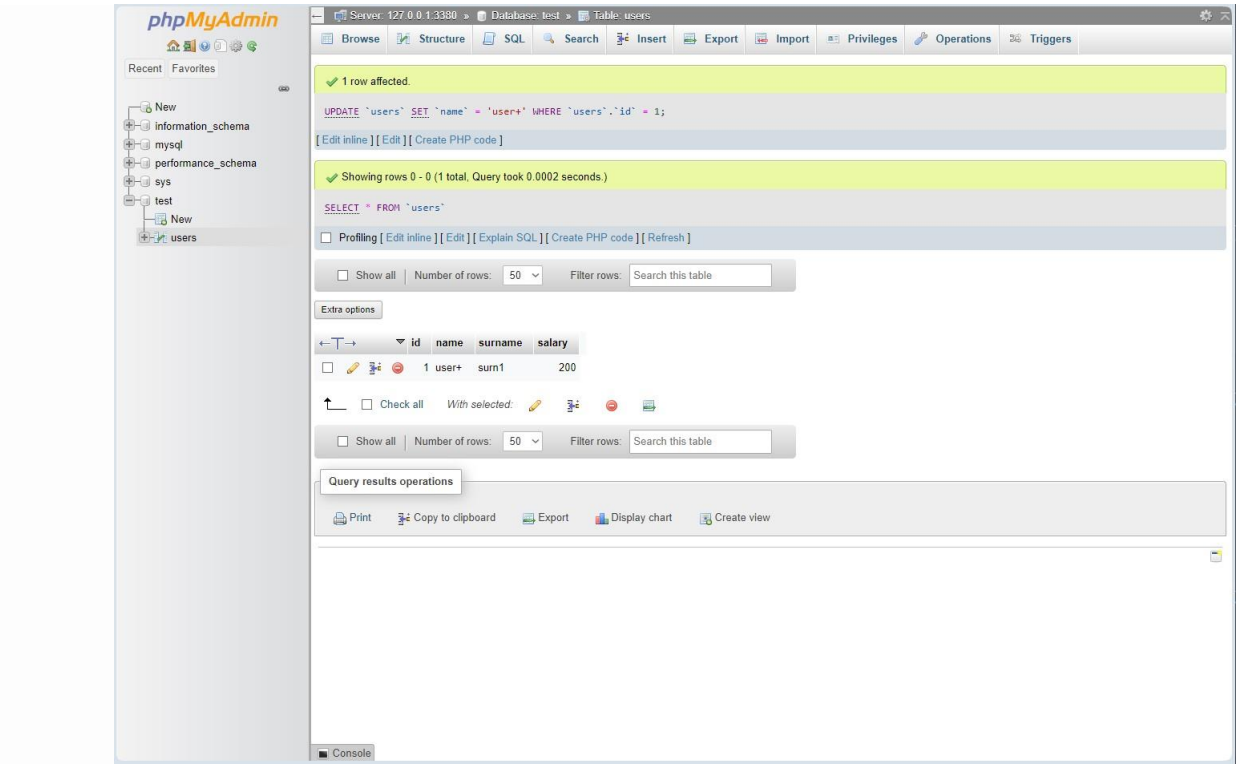
Давайте изменим имя нашему пользователю. Для этого перейдем во вкладку `Browse` и нажмем на иконку с карандашом.



После этого откроется вкладка Insert с инпутами, содержащими данные нашей записи. Давайте изменим имя на 'user+'. После чего нажмем на кнопку Go.

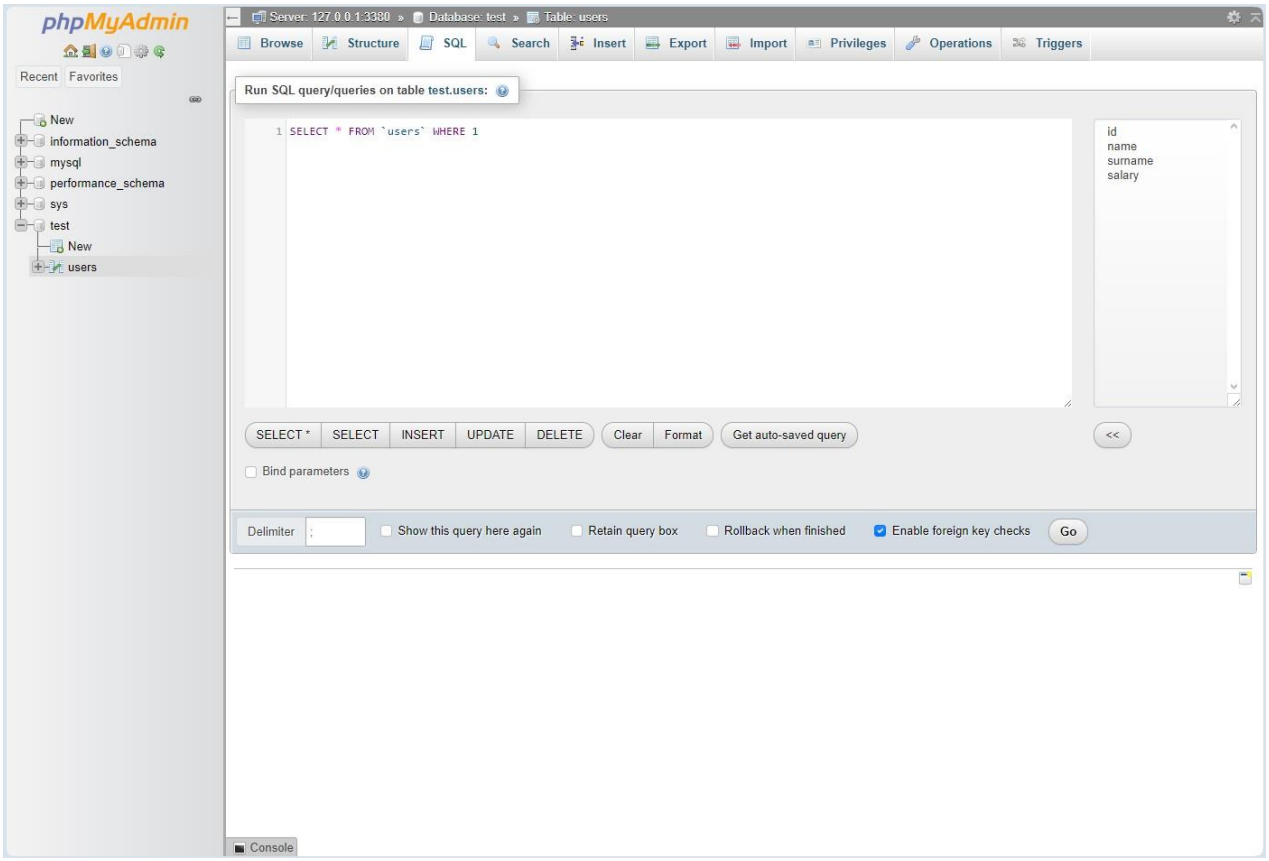


Если изменения применились, то вас перенаправит на страницу с содержимым таблицы. Также в ее верхней части вы увидите сработавший SQL запрос.

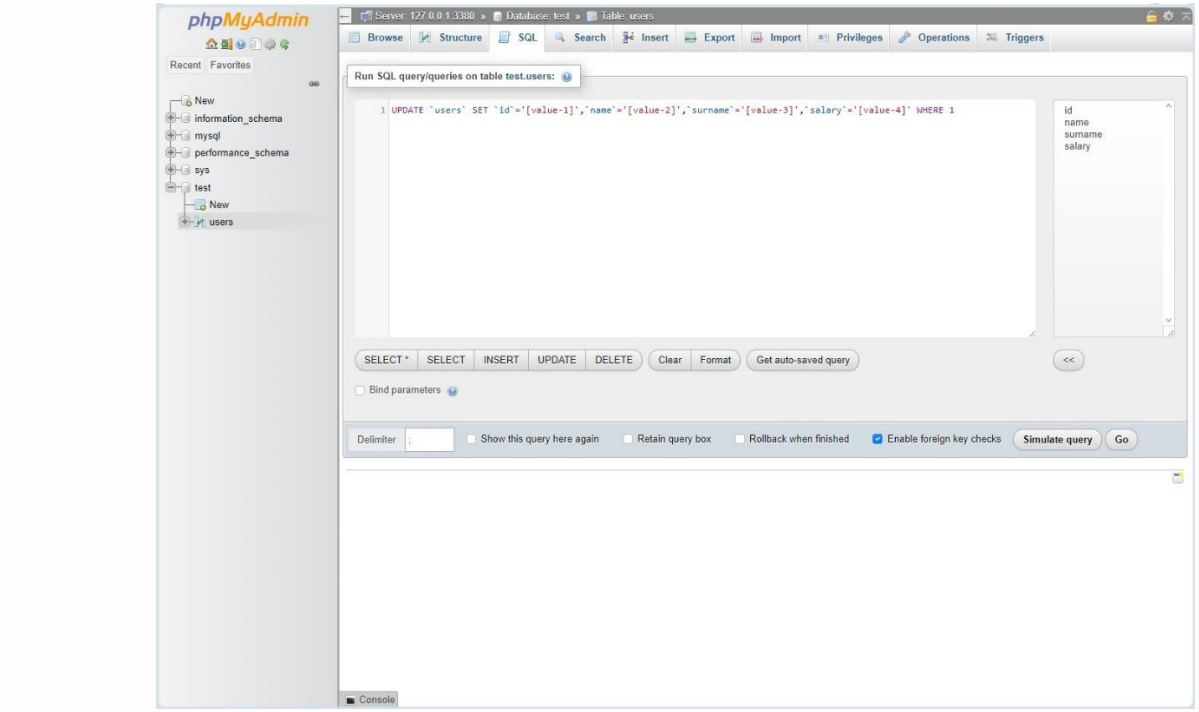


Пример

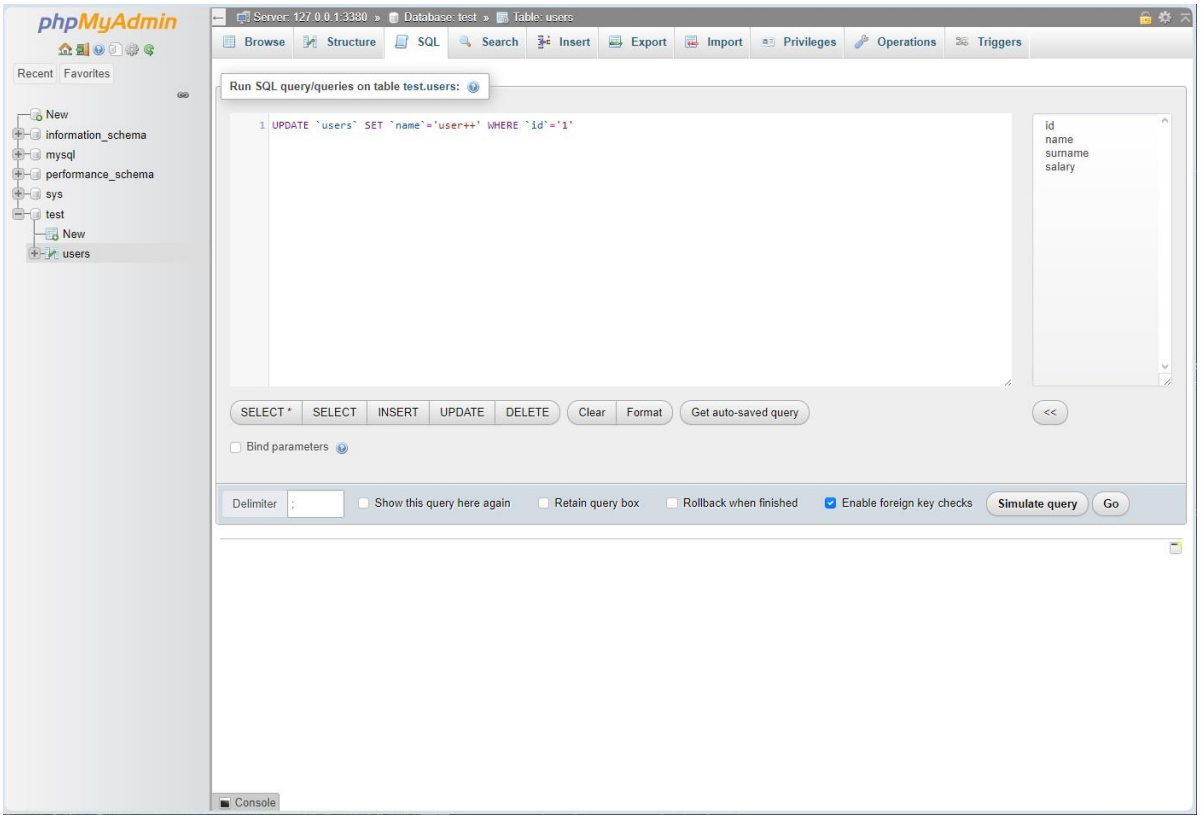
Изменения можно внести и через прямой SQL запрос во вкладке `SQL`. Вначале для удобства можно воспользоваться кнопкой-подсказкой `UPDATE`.



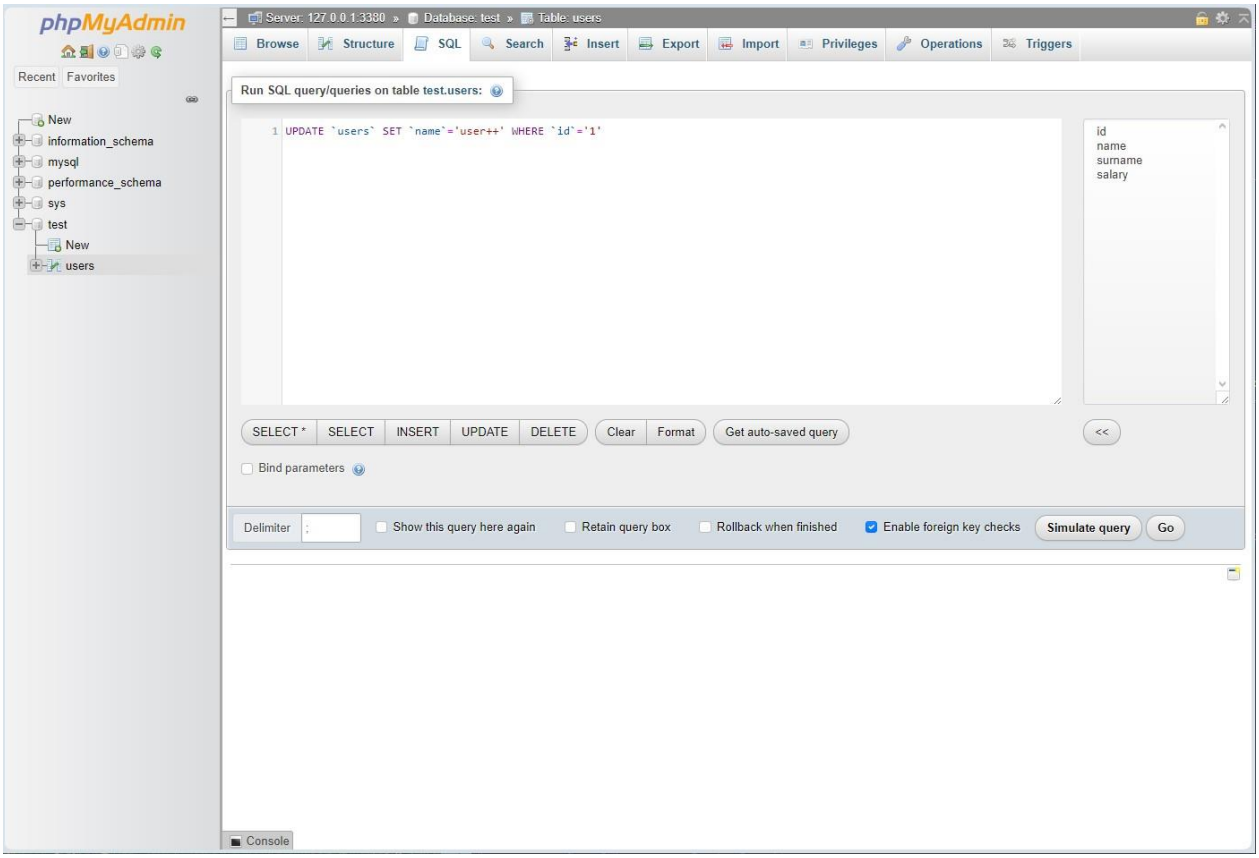
Теперь в поле для запроса появится шаблон, в который можно внести нужные изменения.



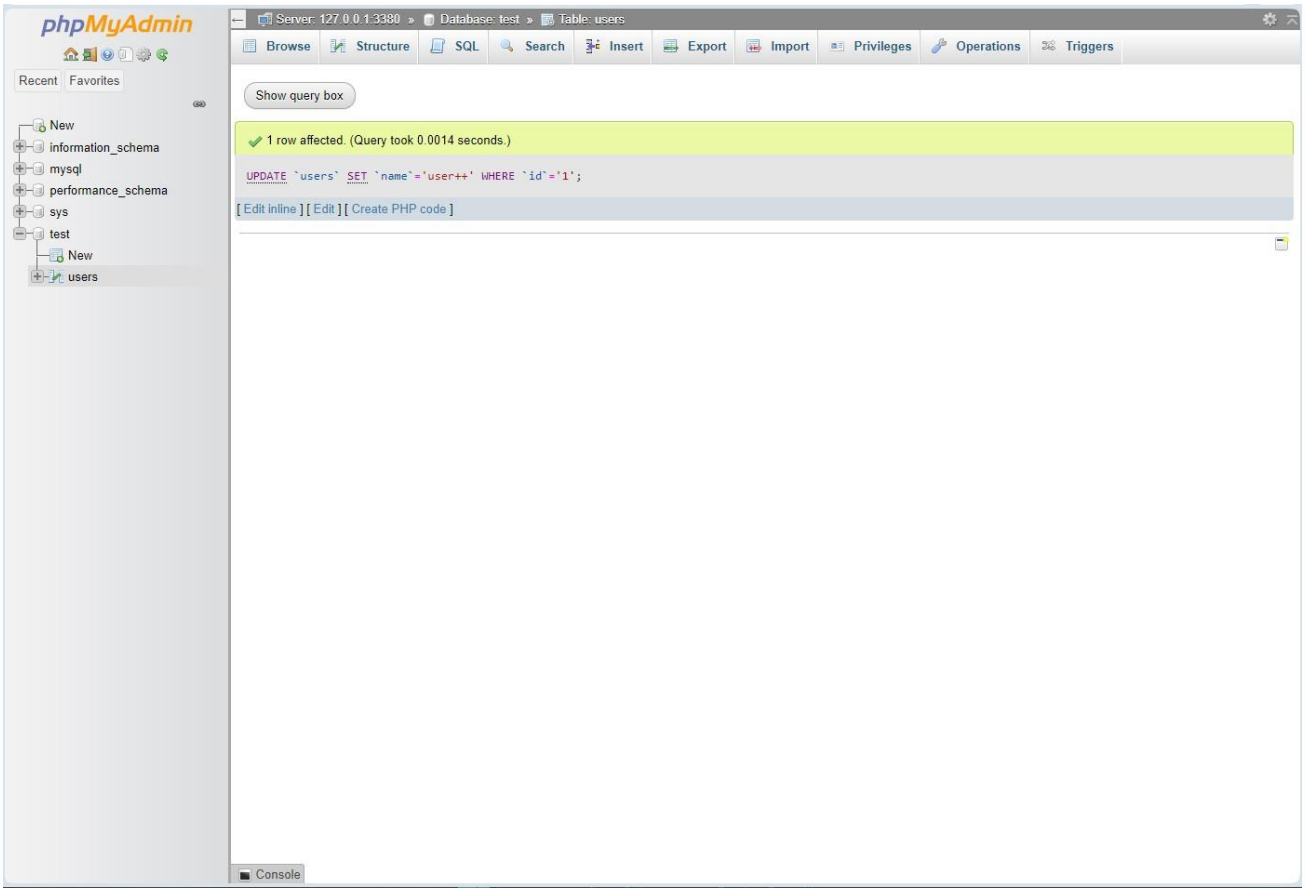
Давайте исправим шаблон для редактирования записи и заменим имя пользователя на user++.



Для подтверждения изменений нажимаем на кнопку Go.



После успешного внесения изменений вы увидите страничку с примененным SQL запросом.



Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №8

Настройка индексов для ускорения запросов

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

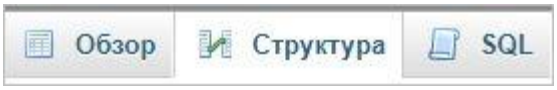
ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: Настроить индексы в таблицах базы данных.

Индексы позволяют оптимизировать работу с базой – данные сортируются и СУБД не приходится искать значения среди всех записей. На практике, это может сократить время обработки запросов на несколько порядков.

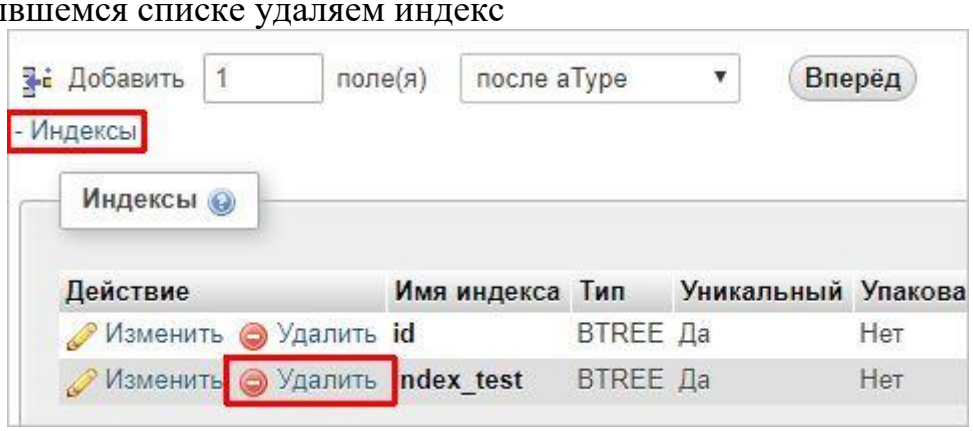
Заходим в phpMyAdmin - выбираем базу данных и таблицу – переходим на вкладку **Структура**



В правой части кликаем по необходимому индексу для его создания



Для удаления в том же разделе **Структура** кликаем по **Индексы** и в раскрывшемся списке удаляем индекс



Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №9

Модификация базы данных в СУБД

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления

здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

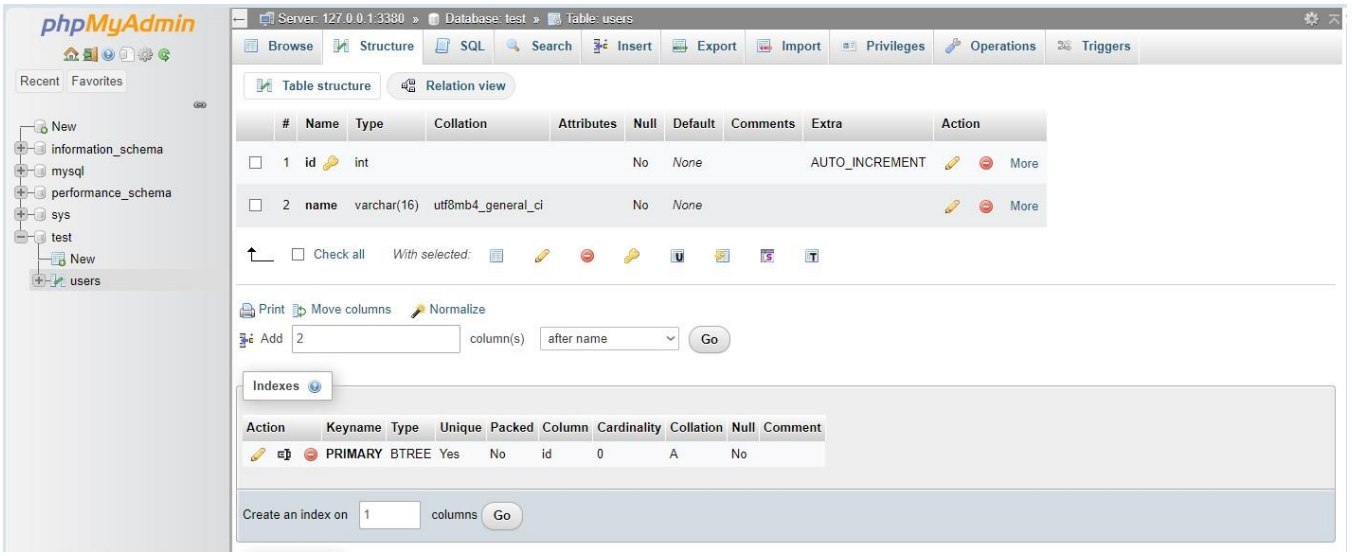
ПК 2.8 Разрабатывать объекты баз данных NoSQL

Задание: Произвести модификацию созданной базы данных

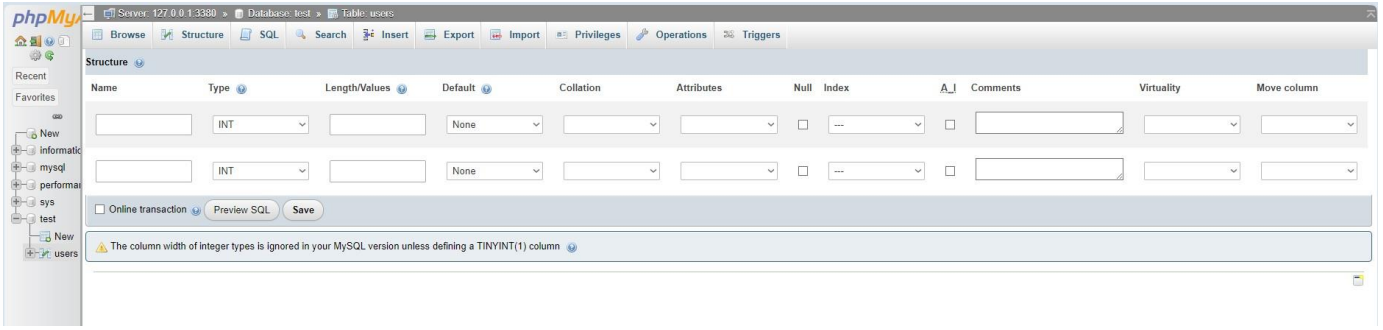
Ход выполнения

В PMA можно редактировать структуру таблицы. Давайте вставим в нее два поля – `surname` и `salary`.

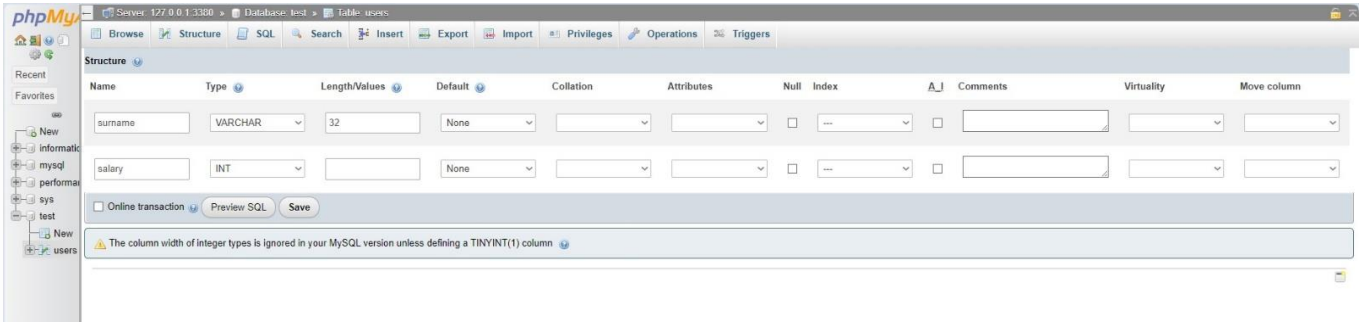
Для этого внизу таблицы найдите инпут `Add`. В нем нужно указать число полей – `2`. Справа от него находится выпадающий список, в котором можно выбрать после каких полей будут вставлены новые. Пусть они вставятся, как и указано по дефолту после поля `name`. Теперь нажимаем кнопку `Go`.



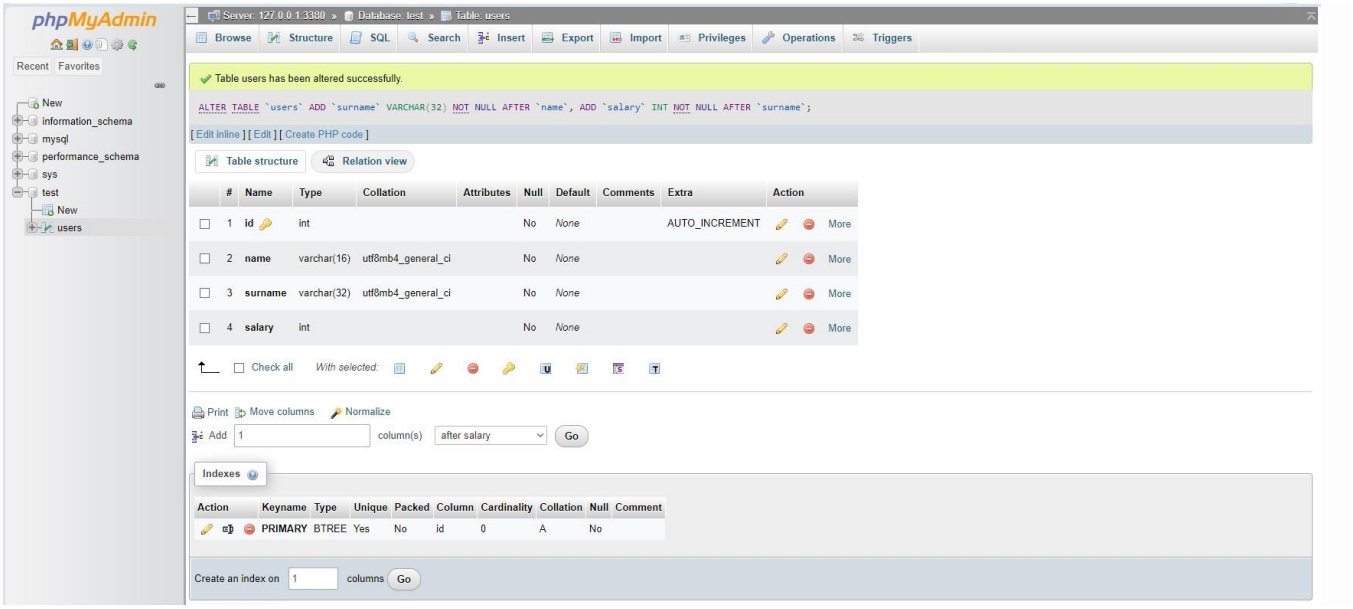
Нам откроется поле, в котором можно назначить имя и тип данных для новых строк.



Заполняем наши поля surname и salary, назначая каждому из них нужный тип данных. Для того, чтобы внесенные изменения сохранились нажимаем на кнопку Save.



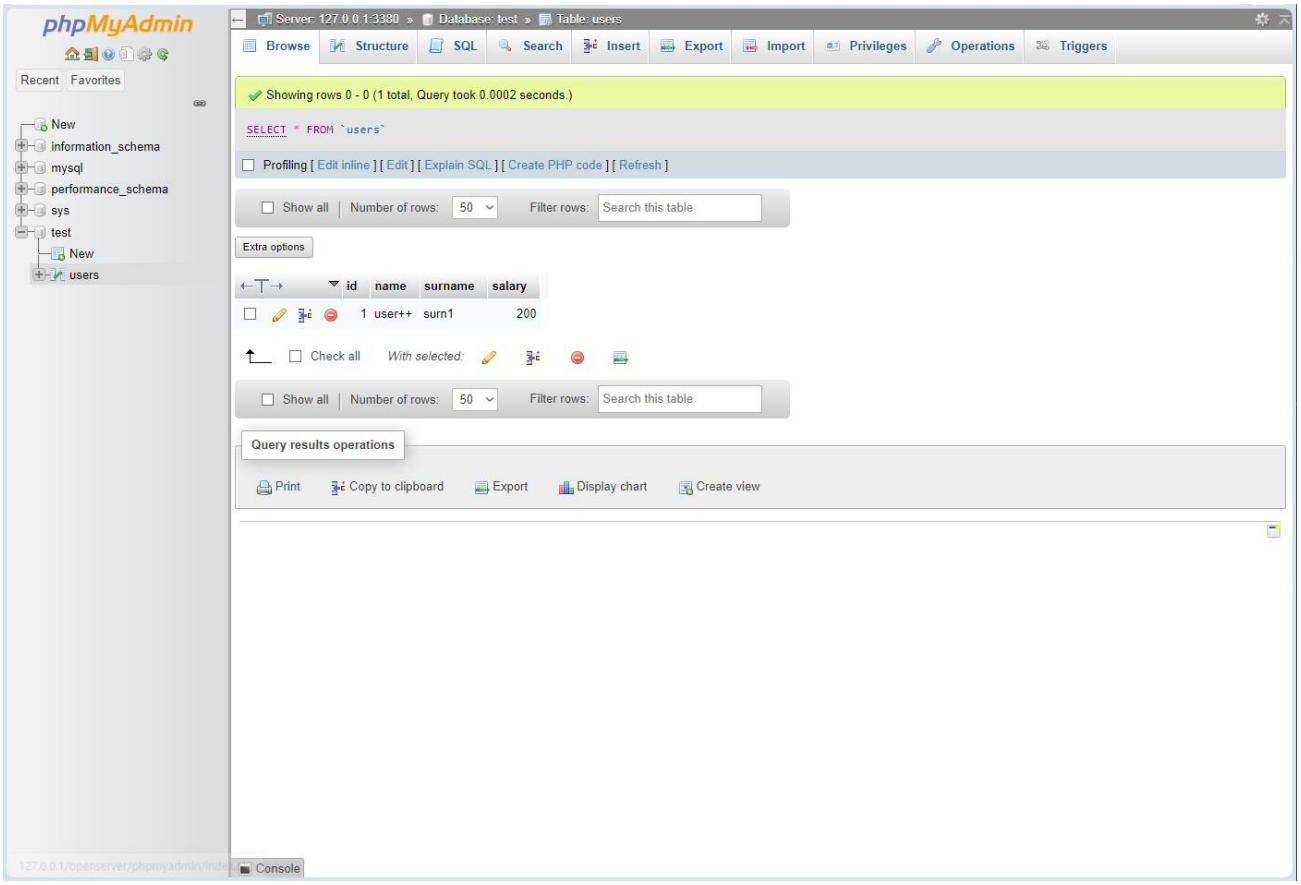
После успешного сохранения новых данных выведется общая структура таблицы. При этом над ней будет отображен SQL запрос, который отработал при вставке новых строк.



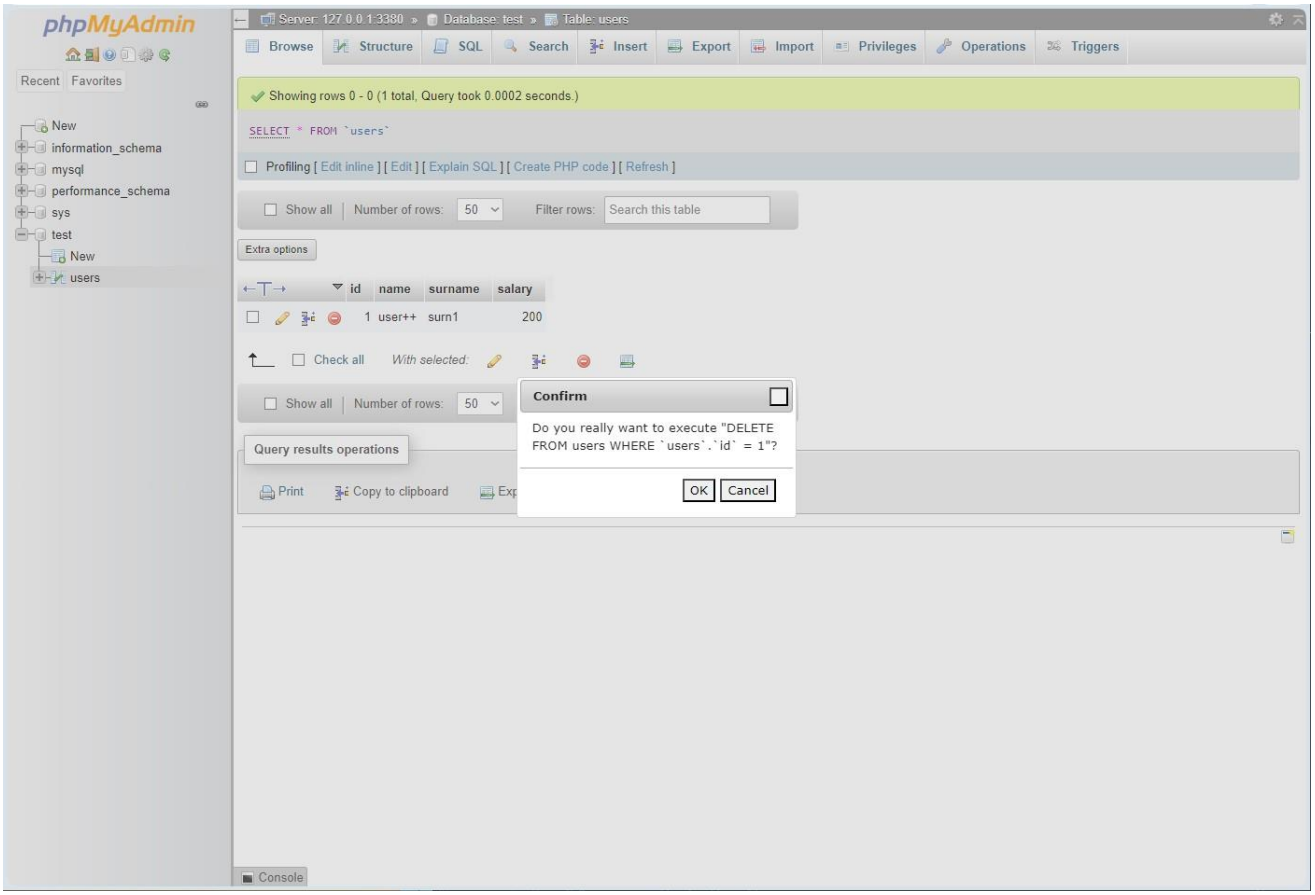
Удалить запись из таблицы можно также с помощью иконки PMA и через прямой SQL запрос. Давайте рассмотрим эти способы.

Пример

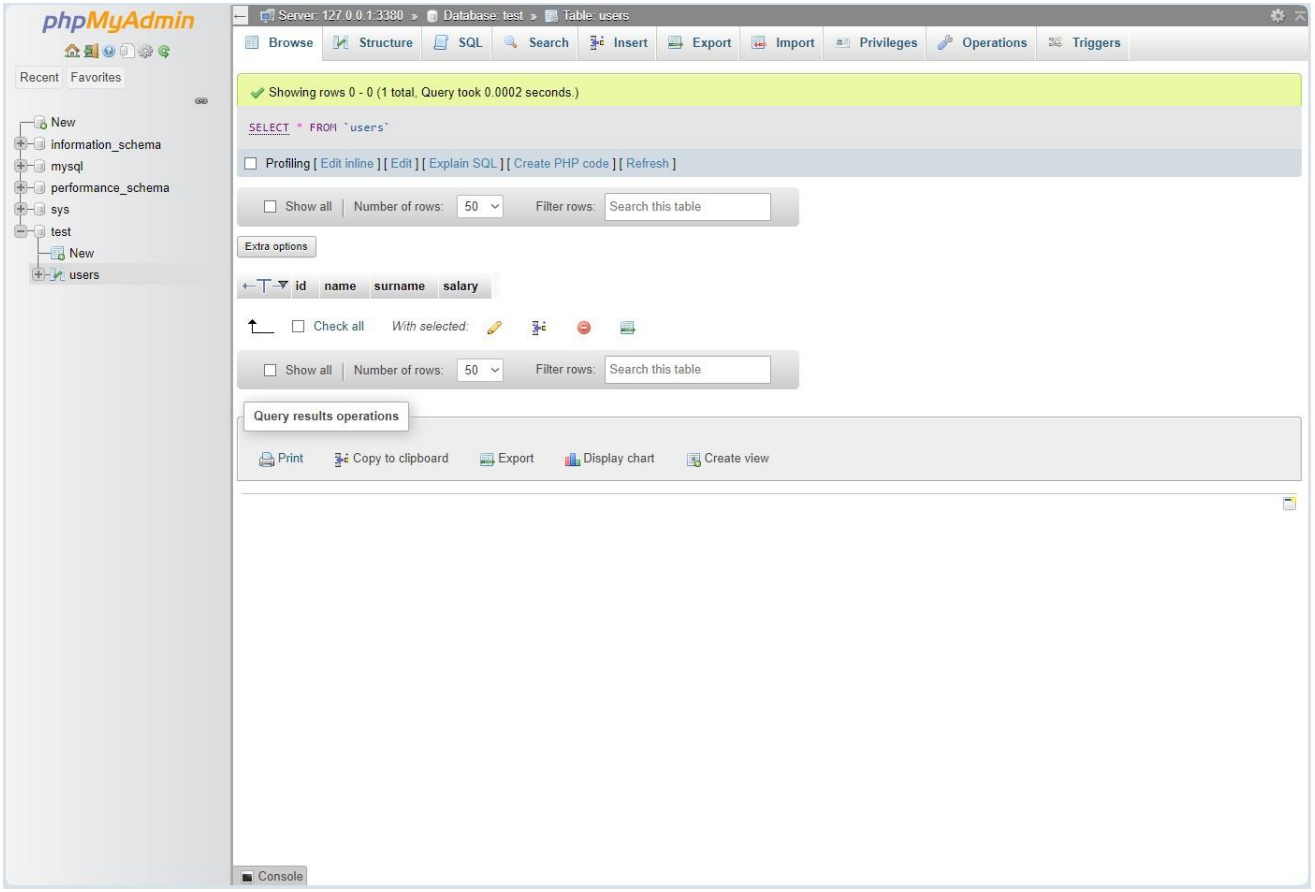
Для удаления нашего пользователя нужно перейти во вкладку Browse и нажать на иконку со значком стоп.



После этого появится окошко для подтверждения удаления

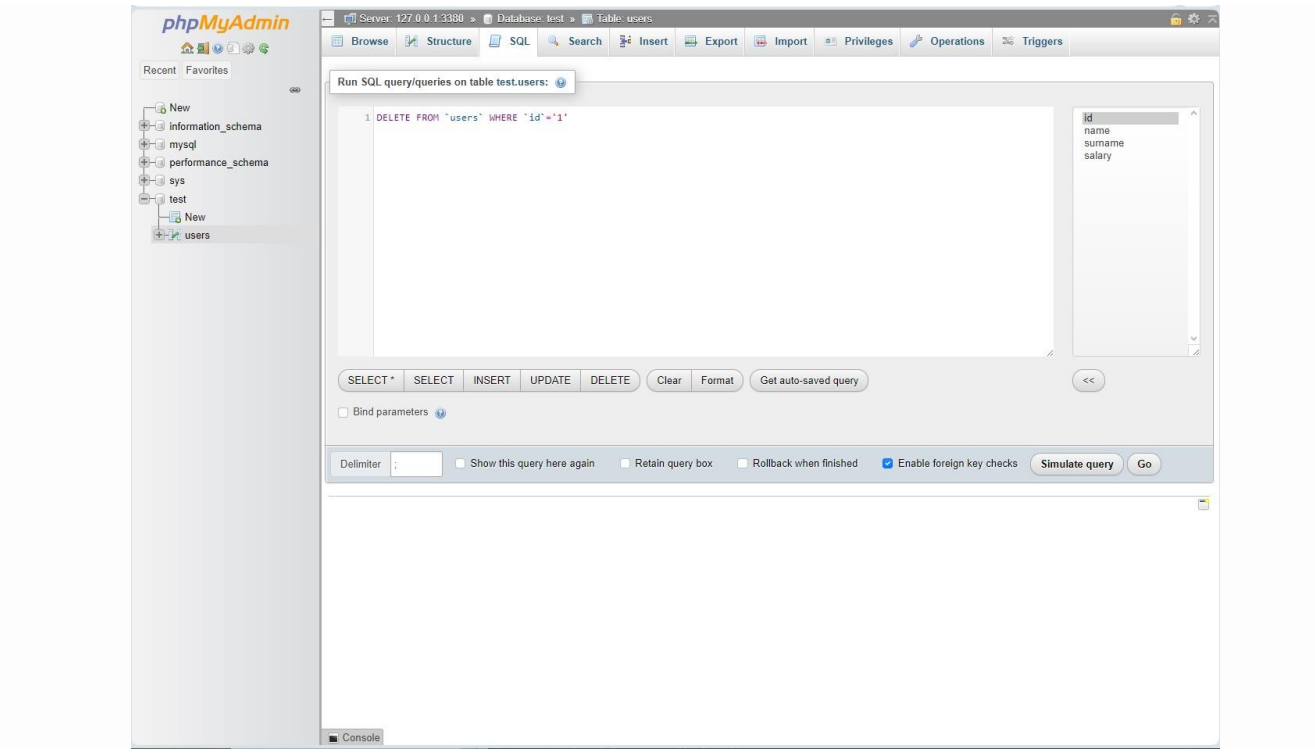


Нажимаем на кнопку OK и нам вернется пустая таблица.

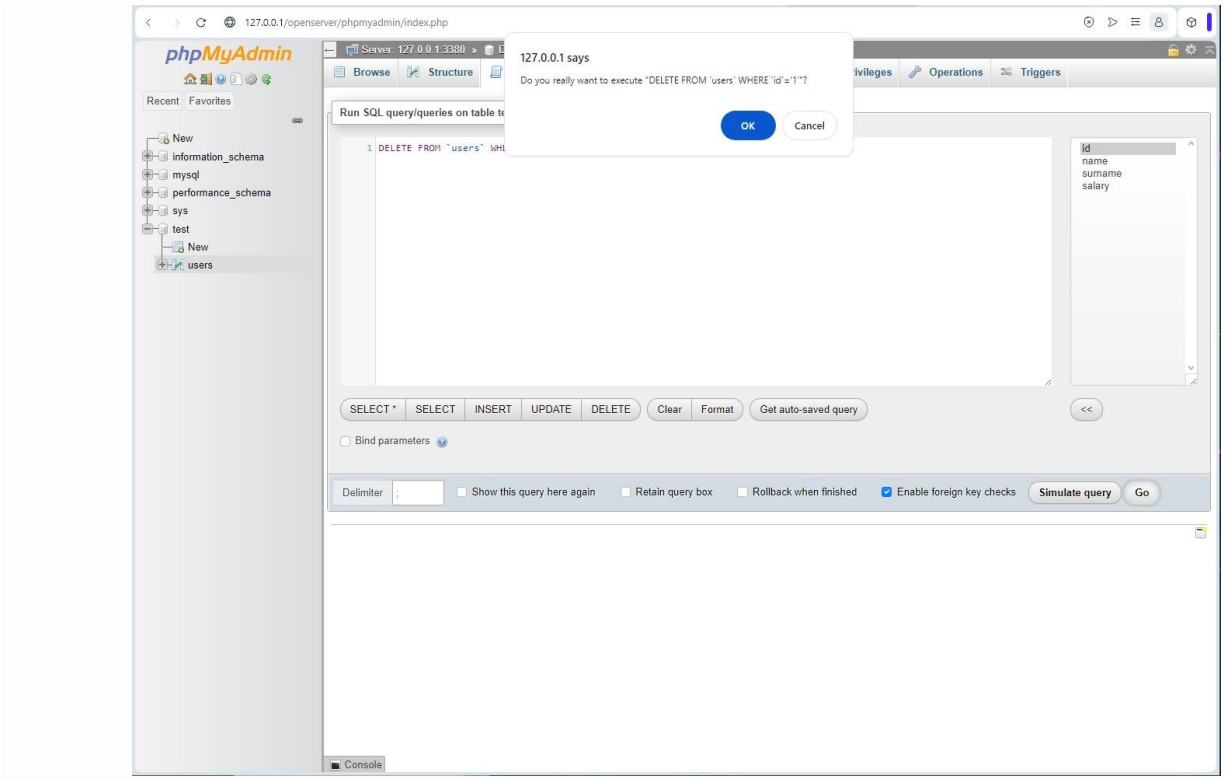


Пример

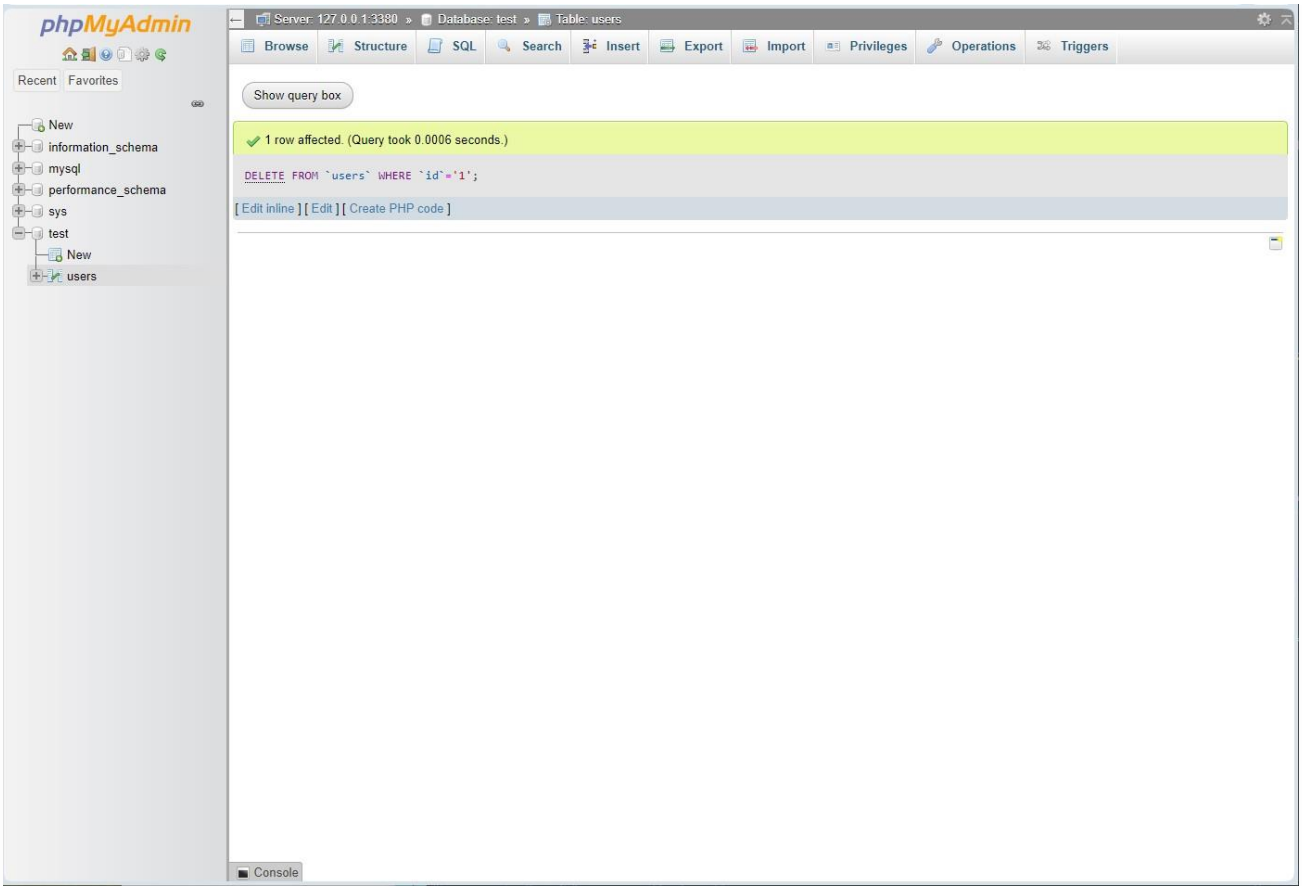
Теперь давайте удалим нашу запись с помощью SQL запроса. Во вкладке SQL нажимаем кнопку-подсказку DELETE и подставляем в шаблон нужные данные.



Подтверждаем удаление нажатием на кнопку Go. После этого появится окошко для подтверждения удаления.



Нажимаем на кнопку OK и нам вернется сработавший SQL запрос.



Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Тема 2.1
Основы SQL и работа с запросами

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Текст задания по теоретической части 1

Ответьте на вопросы.

Вариант 1

1) Создание баз данных
2) Просмотр имеющихся (созданных) таблиц в выбранной базе данных
3) Вставка записей в таблицу
4) Добавление нового столбца
5) Удаление таблицы
6) SELECT * FROM Student
7) ALTER TABLE Marka RENAME MarkaTS
8) DELETE FROM cust WHERE city='London'
9) ALTERTABLE gruppa DROP PRIMARY KEY
10) UPDATE ts SET gos='o333oo' WHERE id_ts=15

Вариант 2

1) Просмотр имеющихся (созданных) баз данных

2) Просмотр структуры созданной таблицы
3) Добавление первичного ключа к уже имеющейся таблице
4) Удаление столбца
5) Переименование таблицы
6) ALTER TABLE forums CHANGE Gruppa GruppaIT char(20)
7) UPDATE Zakaz SET oplata='Карта' WHERE oplata='Безнал'
8) ALTER TABLE forums DROP new_test
9) ALTERTABLEpredmet ADDFOREIGNKEY (id_pr) REFERENCESprepod(id_pr)
10) describe forums

Вариант 3
1) Создание таблиц
2) Удаление баз данных
3) Добавление внешнего ключа к уже имеющейся таблице
4) Переименование столбца и изменение его типа данных
5) Изменение содержимого таблицы
6) ALTER TABLE forums ADD test int(10) AFTER name
7) drop table Student
8) ALTERTABLEprepod ADD PRIMARY KEY id_pr
9) UPDATE bilet SET cena=1500 WHERE cena=1000
10) ALTER TABLE forums CHANGE test test int(15)

Текст задания по теоретической части 2

Напишите SQL-команду на поставленный вопрос.

Вариант 1
1 В какие даты были сделаны заказы
2 Из какого города заказчик *Liu*
3 Кто из заказчиков проживает в *Berlin* и *Rome*
4 Кто из заказчиков с номерами 2002, 2004, 2006 имеет имя на «С»
5 Сколько заказов сделали каждый из продавцов с номерами 1001 и 1007

Вариант 2
1 В каких городах проживают заказчики
2 Из какого города продавец *Peel*
3 Кто из продавцов проживает в *London*с комиссией больше 11%
4 Какие заказы были сделаны летом на сумму больше 1000 руб.
5 Какая средняя стоимость заказов за весну

Вариант 3
1 Какие вариации комиссий продавцов используются в таблице «Продавцы»
2 Какие заказы сделаны после 10 марта 2021 года
3 Какие заказы оформлены 10 марта и 10 июня
4 Кто из заказчиков проживает в городах на «R»и «S»
5 Какой минимальный рейтинг заказчиков из города *SanJose*

Вариант 4

1 Какие вариации рейтингов заказчиков используются в таблице «Заказчики»
2 Кто из заказчиков проживает в *Paris*
3 Кто из заказчиков проживает в *SanJose*с рейтингом больше 150
4 Кто из продавцов проживает в *London(Paris)* с именем на «М» (Мотика)
5 Покажите самый дорогой заказ сделанный 10 марта

Вариант 5

1 В каких городах проживают заказчики
2 Кто из продавцов имеет комиссионные ниже 15%
3 Какие заказы были оформлены 10 апреля с суммой больше 1000 руб.
4 Кто из заказчиков проживает в *Rome* с именем на «Р» (Перейра)
5 Какой общий рейтинг заказчиков из *Rome*

Текст задания по теоретической части 3

Напишите SQLкоманду на поставленный вопрос.

1 Вариант

1) Показать самые большие комиссионные продавцов в каждом городе. Результат упорядочите по названию города.

2) Какой общий рейтинг заказчиков в каждом городе? Результат упорядочите по названию города.

3) Покажите сумму заказов за каждую дату. Результат упорядочите по дате.

4) Какая средняя стоимость заказа по каждой дате? Результат упорядочите по дате.

5) Покажите среднюю стоимость заказов каждого продавца за каждый день

2 Вариант

1) Сколько заказчиков в каждом городе? Результат упорядочите по названию города.

2) Сколько заказчиков обслужил каждый продавец? Результат упорядочите по номеру продавца.

3) Покажите самый дорогой заказ за каждую дату. Результат упорядочите по дате.

4) Покажите самую дорогую покупку, оформленную каждый продавцом.

5) Покажите количество заказчиков за каждый день

Критерий оценки

Каждый полный правильный ответ оценивается в1 балл.
0-2 правильных ответа – «2»
3 правильных ответа – «3»
4 правильных ответа – «4»
5 правильных ответов – «5»

Практическое занятие №10

Создание и модификация базы данных средствами языка SQL

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Часть 1 – Создание базы данных

Краткие теоретические сведения

Подключение к серверу базы данных

mysql -u root -p

1) Создание баз данных

create database <имябазыданных>

create database Practika;

1.1) Создание базы данных с установкой кодировки (русский язык)

create database <имябд>CHARACTER SET utf8 COLLATE utf8_unicode_ci

create database st CHARACTER SET utf8 COLLATE utf8_unicode_ci

2) Просмотр имеющихся (созданных) баз данных

show databases;

show Practika;

3) Выбор базы данных для работы с ней, т.е. для создания в ней таблиц
use<имябазыданных>;

usePractika;

4) Удаление баз данных **drop database <имя базы данных>;**

drop database Practika;

5) Созданиетаблиц
createtable<имятаблицы>
(название_столбца1 типам_данных(длина_поля) notnull,
название_столбца2 типам_данных(длина_поля),
…
primarykey (имя_поля_первичного_ключа)
);

// создаётся таблица forums с первичным ключём id_forum. Для определения первичного ключа используется оператор PRIMARY KEY

```
CREATE TABLE forums
(
  id_forum int(5) NOT NULL auto_increment,
  name char(38),
  pos float(4,2),
  hide int(3),
PRIMARY KEY (id_forum)
);
```

// создаётсятаблица Customers спервичнымключём cnum ивнешнимключём snum. Для определения внешнего ключа используется оператор FOREIGN KEY. Оператор REFERENCES указывает имя таблицы содержащей первичный ключ и имя самого первичного ключа. В данном случае столбцу внешнего ключа snum определили NOTNULL – это будет означать, что при заполнении таблицы Customers нельзя будет оставить не заполненным столбец snum. Если у столбца внешнего ключа не стоит ограничение NOTNULL, то значение внешнего ключа не обязательно и можно будет его оставить пустым при заполнении таблицы данными.

```
CREATE TABLE Customers
(
   cnum integer NOT NULL,
   cname char(10),
   city char(10),
   snum integer NOT NULL,
PRIMARY KEY (cnum),
FOREIGN KEY (snum) REFERENCES Salespeople(snum)
);
```

5.1) Установка значений по умолчанию

Когда вы вставляете строку в таблицу без указания значений в ней для каждого поля, SQL должен иметь *значение по умолчанию* для включения его в определенное поле, или же команда будет отклонена. Наиболее общим значением по умолчанию является **NULL**. Это – значение по умолчанию для любого столбца, которому не было дано ограничение NOT NULL или который имел другое назначение *по умолчанию*.

Значение **DEFAULT** (ПО УМОЛЧАНИЮ) указывается в команде CREATE TABLE тем же способом, что и ограничение столбца, хотя, с технической точки зрения, значение DEFAULT не ограничительного свойства — оно не ограничивает значения, которые вы можете вводить, а просто определяет, что может случиться, если вы не введете любое из них.

Предположим, что вы работаете в офисе Нью-Йорка, и подавляющее большинство ваших продавцов живут в Нью-Йорке. Вы можете указать Нью-Йорк в качестве значения поля city, по умолчанию, для вашей таблицы Продавцов:

```
CREATE TABLE Salespeople
(
  snum integer NOT NULL UNIQUE,
  sname char(10) NOT NULL UNIQUE,
  city char(10) DEFAULT = 'New York',
comm  decimal CHECK (comm < 1)
);
```

Конечно, вводить значение Нью-Йорк в таблицу каждый раз, когда назначается новый продавец, не такая уж необходимость, и можно просто пренебречь им (не вводя его) даже если оно должно иметь некоторое значение. *Значение по умолчанию* такого типа, более предпочтительно, чем, например, длинный конторский номер, указывающий на ваше собственное ведомство, в таблице Заказов.

Длинные числовые значения более расположены к ошибке, поэтому если подавляющее большинство (или все) ваших Заказов должны иметь ваш собственный конторский номер, желательно устанавливать для них значение *по умолчанию*.

5.2) установка уникального значения

UNIQUE – это ограничение устанавливает, что все значения данного столбца будут уникальны в пределах таблицы, и создает индекс. Можно применять к столбцам с поддержкой NULL, но так как NULL будет считаться уникальным значением, возможна только одна NULL-запись.

phoneVARCHAR(20) **UNIQUE**NOTNULL

5.3) установка проверки данных

CHECK – позволяет установить дополнительную проверку данных для столбца или набора столбцов. Это тоже CONSTRAINT, так как накладывает ограничение.

Примеры:

Ограничение даты рождения сотрудника –

birthday DATE NOT NULL **CHECK** (birthday > ‘1900-01-01’)

Ограничениепола –

pol char(10) NOT NULL **CHECK** (pol IN ('М', 'Ж'))

Ограничениетипатовара –

tip_tovara VARCHAR(30) **CHECK** (VALUE IN ('Электроника', 'Одежда', 'Еда'));

При вставке данных, не прошедших проверку, будет сообщение об ошибке **Check constraint 'staff_chk_1' is violated**. Ситуация усложняется, когда установлено несколько **CHECK,** поэтому рекомендуется давать понятное имя.

Воспользуемся полной командой для создания **CHECK** и определим не только ограничение даты рождения, но и допустимые форматы телефона через регулярное выражение.

```
CREATE TABLE Staff
(
   id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(255) NOT NULL,
   position VARCHAR(30),
   birthday DATE NOT NULL,
   has_child BOOLEAN DEFAULT(0) NOT NULL,
   phone VARCHAR(20) UNIQUE NOT NULL,
   CONSTRAINT staff_chk_birthday CHECK (birthday > '1900-01-01'),
   CONSTRAINT staff_chk_phone CHECK (phone REGEXP '[+]?[0-9]{1,3}
?\\(?[0-9]{3}\\)? ?[0-9]{2}[0-9 -]+[0-9]{2}')
);
```

Для добавления ограничений используем оператор **CONSTRAINT**, при этом, все названия уникальны, как и имена таблиц. Учитывая, что по умолчанию названия включают в себя и имя таблицы, рекомендуем придерживаться этого правила. Если используется **CONSTRAINT**, мы обязаны дать имя ограничению, которое вводим.

6) Просмотр имеющихся (созданных) таблиц в выбранной базе данных

show tables;

7) Просмотр структуры созданной таблицы

describe<имятаблицы>;

```
describeforums;
```

8) Вставка записей в таблицу

INSERTINTO<имя_таблицы>VALUES
(значение_столбца1, значение_столбца2,…),
…
(значение_столбца1, значение_столбца2,…);

После оператора VALUES в скобках через запятую перечисляются значения соответствующих полей таблицы в соответствии с их типами.

```
INSERT INTO authors VALUES
(1, 'Maks', 123, 'maks@mail.ru ', 'www.softtime.ru', 'программист', 0, 0),
(2, 'Igor', 123, 'igor@mail.ru', 'http://www.softtime.ru', 'Программист', 407, 0);
```

8.1) Вставка результатов запроса

Вы можете также использовать команду INSERT, чтобы получать или выбирать значения из одной таблицы и помещать их в другую, чтобы использовать их вместе с запросом. Чтобы сделать это, вы просто заменяете предложение VALUES (из предыдущего примера) на соответствующий запрос:

```
INSERT INTO Londonstaff
SELECT *
FROM Salespeople
WHERE city = 'London';
```

Здесь выбираются все значения, произведенные запросом — то есть, все строки из таблицы Продавцов со значениями city = "London" — и помещаются в таблицу называемую Londonstaff. Чтобы это работало, таблица Londonstaff должна отвечать следующим условиям:

* Она должна уже быть создана командой CREATE TABLE.

* Она должна иметь четыре столбца, которые совпадают с таблицей Продавцов в терминах типа данных; то есть первый, второй, и так далее, столбцы каждой таблицы должны иметь одинаковый тип данных (причем они не обязаны иметь одинаковых имен).

Общее правило то, что вставляемые столбцы таблицы, должны совпадать со столбцами выводимыми подзапросом, в данном случае, для всей таблицы Продавцов.

Londonstaff — это теперь независимая таблица, которая получила некоторые значения из таблицы Продавцов (Salespeople). Если значения в таблице Продавцов будут вдруг изменены, это никак не отразится на таблице Londonstaff (хотя вы могли бы создать такой эффект с помощью Представления (VIEW), описанного в Главе 20).

Так как или запрос или команда INSERT могут указывать столбцы по имени, вы можете, если захотите, переместить только выбранные столбцы, а также переупорядочить только те столбцы, которые вы выбрали.

Предположим, например, что вы решили сформировать новую таблицу с именем Daytotals, которая просто будет следить за общим количеством долларов сумм приобретений упорядоченных на каждый день. Вы можете ввести эти данные независимо от таблицы Заказов, но сначала вы должны заполнить таблицу Daytotals информацией ранее представленной в таблице Заказов.

Понимая, что таблица Заказов охватывает последний финансовый год, а не только несколько дней, как в нашем примере, вы можете видеть преимущество использования следующего условия INSERT в подсчете и вводе значений

```
INSERT INTO Daytotals (date, total)
SELECT odate, SUM (amt)
FROM Orders
GROUP BY odate;
```

Обратите внимание что, как предложено ранее, имена столбцов таблицы Заказов и таблицы Daytotals — не должны быть одинаковыми. Кроме того, если дата приобретения и общее количество — это единственные столбцы в таблице, и они находятся в данном Заказе, их имена могут быть исключены из вывода из-за их очевидной простоты.

8.2) Использование подзапросов с INSERT

INSERT — это самый простой случай. Вы уже видели, как вставлять результаты запроса в таблицу. Вы можете использовать подзапросы внутри любого запроса, который генерирует значения для команды INSERT тем же самым способом, которым вы делали это для других запросов — т.е. внутри предиката или предложения HAVING.

Предположим, что мы имеем таблицу с именем SJpeople, столбцы которой

совпадают со столбцами нашей таблицы Продавцов. Вы уже видели, как заполнять таблицу подобно этой, заказчиками в городе, например, в San Jose:

```
INSERT INTO SJpeople
SELECT *
FROM Salespeople
WHERE city = 'San Jose';
```

Теперь мы можем использовать подзапрос, чтобы добавить к таблице SJpeople всех продавцов, которые имеют заказчиков в San Jose, независимо от того, находятся ли там продавцы или нет:

```
INSERT INTO Sjpeople
SELECT *
FROM Salespeople
WHERE snum = ANY (SELECT snum
        FROM Customers
        WHERE city = 'San Jose');
```

Оба запроса в этой команде функционируют так же, как если бы они не являлись частью выражения INSERT. Подзапрос находит все строки для заказчиков в San Jose и формирует набор значений snum. Внешний запрос выбирает строки из таблицы Salespeople, где эти значения snum найдены. В этом примере, строки для продавцов Rifkin и Serres, которые назначены заказчикам в San Jose — Liu и Cisneros, будут вставлены в таблицу SJpeople.

Индивидуальное задание

Задание: создайте базу данных из раздаточного материала

Предусмотреть требования и ограничения к таблицам.
Продавцы:
– все поля кроме «comm» обязательные;
– нельзя ввести комиссионные ниже 0.10 и выше 0.20.

Заказчики:
– все поля кроме «snum» обязательные;
– по умолчанию в столбце «city» должно быть значение «SanJose».

Заказы:
– все поля кроме «snum» и «cnum» обязательные.

Часть 2 – Модификация базы данных и структуры таблиц

Краткие теоретические сведения

Изменение структуры таблицы
ALTERTABLE<имя_таблицы><значение>;

Синтаксис	Описание команды

ALTERTABLE<имя_таблицы> **ADD**<имя столбца><тип столбца> **FIRST**<имя столбца, перед которым добавлять новый столбец> **AFTER**<имя столбца, после которого добавлять новый столбец>	Добавление нового столбца **FIRST** добавляет новый столбец перед указанным столбцом. **AFTER** добавляет новый столбец после указанного столбца. Если место добавления не указано, по умолчанию столбец добавляется в конец таблицы.
ALTER TABLE <имя_таблицы> **ADD INDEX** [index_name] (index_col_name,...)	Добавлениеиндекса index_name длястолбца index_col_name. Если имя индекса index_name не указывается, ему присваивается имя совпадающее с именем столбца index_col_name.
ALTERTABLE<имя_таблицы> **ADD PRIMARY KEY**<имя столбца, которому устанавливается первичный ключ> **ADDFOREIGNKEY**<имя столбца, которому устанавливается первичный ключ>**REFERENCES**<имя таблицы, на которую ссылаетесь, т.е. главную><имя столбца, на который ссылаетесь>	Делает указанный столбец первичным ключом таблицы. Делает указанный столбец внешним ключом таблицы.
ALTERTABLE<имя_таблицы> **CHANGE**<старое_имя_столбца> <новое _имя_столбца><тип_данных_столбца> **Пример:** **ALTERTABLE**<имя_таблицы> **CHANGE** old_col_name new_col_name type	Изменение имени столбца с обязательным указанием типа данных для нового столбца!!! Пример: Изменение столбца с именем old_col_name на столбец с именем new_col_name и типом type.
ALTERTABLE<имя_таблицы> **DROP**<имя_столбца> **Пример:** **ALTERTABLE**<имя_таблицы> **DROP** col_name	Удаление столбца Пример: Удаление столбца с именем col_name.
ALTER TABLE <имя_таблицы> **DROP PRIMARY KEY**	Удаление первичного ключа таблицы.
ALTERTABLE<имя_таблицы> **DROP INDEX**<имя_индексного_столбца> **Пример:** **ALTERTABLE**<имя_таблицы> **DROP INDEX** index_name	Удаление индексного столбца Пример: Удаление индекса index_name.

1) Добавление нового столбца

Добавим в таблицу forums новый столбец test, поместив его после столбца name.

```
ALTER TABLE forums ADD test int(10) AFTER name;
```

2) Удаление столбца

Теперь удалим столбец new_test:

```
ALTER TABLE forums DROP new_test;
```

3) Удаление таблицы

drop table <имя таблицы>;

4) Переименование таблицы

```
ALTER TABLE t1 RENAME t2;
```

5) Переименование столбца и изменение его типа данных

```
ALTER TABLE forums CHANGE test   test   int(15);
```

6) Удаление строк из таблиц

```
DELETE FROM Salespeople;
```

Обычно, вам нужно удалить только некоторые определенные строки из таблицы. Чтобы определить какие строки будут удалены, вы используете предикат, так же как вы это делали для запросов. Например, чтобы удалить продавца Axelrod из таблицы, вы можете ввести

```
DELETE FROM Salespeople
WHERE snum = 1003;
```

Конечно, вы можете также использовать DELETE с предикатом, который бы выбирал группу строк, как показано в этом примере:

```
DELETE FROM Salespeople
WHERE city = 'London';
```

7) Изменение содержимого таблицы

UPDATE <имя_таблицы> SET <имя_столбца> = <новое_значение>
WHERE <имя_столбца> = <старое_значение>;

Следующий пример изменит название города в записях пользователей с "*Ryazan*" на "*Рязань*":

```
UPDATE users SET city='Рязань' WHERE city='Ryazan';
```

7.1) Команда UPDATE для многих столбцов

Однако вы не должны ограничивать себя модифицированием единственного столбца с помощью команды UPDATE. Предложение SET может назначать любое число столбцов, отделяемых запятыми. Все указанные назначения могут быть сделаны для любой табличной строки, но только для одной в каждый момент времени. Предположим, что продавец Motika ушел на пенсию, и мы хотим переназначить его номер новому продавцу:

```
UPDATE Salespeople
SET sname = 'Gibson', city = 'Boston', comm = .10
```

```
WHERE snum = 1004;
```

Эта команда передаст новому продавцу Gibson, всех текущих заказчиков бывшего продавца Motika и Заказы, в том виде, в котором они были скомпонованы для Motika с помощью поля snum.

Вы не можете, однако, модифицировать сразу много таблиц в одной команде, частично потому, что вы не можете использовать префиксы таблицы со столбцами измененными предложением SET. Другими словами, вы не можете сказать — "SET Salespeople.sname = Gibson" в команде UPDATE, вы можете сказать только так — "SET sname = Gibson".

Задание:

1 Законспектировать все команды по данной практической работе.

2 Используя команды SQL выполните следующее задание:

– переименовать таблицу Продавцы, к текущему имени добавьте цифру 1;

– в таблице Продавцы переименуйте столбцы «sname» и «city» – в конец текущего имени добавьте цифру 1;

– в таблицу Продавцы добавить столбец **Naprav после столбца Sname** (направление деятельности продавца). Столбцу указать тип данных и размер поля;

– внести **данные** в столбец **Naprav** (например: продукты, молочная продукция, алкогольная продукция, табачные изделия, колбасная продукция, мясная продукция, соки-воды);

– в таблицу Заказы добавить столбец **Dostavka** (стоимость доставки, числовое значение) **после столбца Amt**. Столбцу указать тип данных и размер поля;

– внести **данные** в столбец **Dostavka** (цифры, означающие стоимость доставки);

– в таблицу Заказы добавить столбец **Oplata** (вид оплаты заказа) **после столбца Dostavka**. Столбцу указать тип данных и размер поля;

– внести **данные** в столбец **Oplata** (например: наличный, безналичный);

– в таблицу Заказчики добавить столбец **pol**(пол заказчика) **после столбца cname**. Столбцу указать тип данных и размер поля;

– внести **данные** в столбец **pol** («м» или «ж»).

– в таблице Заказчики изменить London на Paris;

– в таблице Продавцы изменить всю комиссию которая **больше** 0,12 на 0,16;

– удалите строку из таблицы Заказы с номером заказа 3011.

3 Покажите работу преподавателю.

4 Верните прежние названия таблиц и столбцов.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №11

Написание SQL-запросов для выборки данных из базы.

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Краткие теоретические сведения

Выборка данных осуществляется с помощью оператора **SELECT.** Формат команды приведён ниже.

SELECT [**DISTINCT**] список_выбираемых_полей
FROMсписок_таблиц
[**WHERE**условие_выборки]
[**GROUPBY**условие_группировки]
[**ORDERBY**условие_упорядочивания]

Пример выборки столбцов snum, sname, city, comm из таблицы Salespeople.

```
SELECT snum, sname, city, comm FROM Salespeople;
```

Для удаления избыточных данных команда **SELECT**использует оператор **DISTINCT**

```
SELECT DISTINCT snum FROM Orders;
```

В выборке данных могут применяться условные операторы. Например:

```
SELECT * FROM Customers WHERE rating = 100;
```

Операторы Буля и как они работают:

1) **AND** (логическое И) берет два Буля (в форме A AND B) как аргументы и оценивает их по отношению к истине, верны ли они оба;

2) **OR** (логическое ИЛИ) берет два Буля (в форме A OR B) как аргументы и оценивает на правильность, верен ли один из них;

3) **NOT(!)** (логическое НЕ) берет одиночный Булев (в форме NOT A) как аргументы и заменяет его значение с неверного на верное или верное на неверное.

```
SELECT * FROMCustomersWHEREcity = 'SanJose' ANDrating> 200;
SELECT * FROMCustomersWHEREcity = 'SanJose' ORrating> 200;
SELECT * FROM Customers WHERE city = 'San Jose' OR NOT rating > 200;
```

Уловныеоператоры IN, BETWEEN (REGEXP), LIKE:

1) оператор **IN** определяет набор значений, в который данное значение может или не может быть включено. В соответствии с нашей учебной базой данных, на которой вы обучаетесь по настоящее время, если вы хотите найти всех продавцов, которые размещены в Barcelona или в London, вы должны использовать следующий запрос :

```
SELECT * FROM SalespeopleWHERE city = 'Barcelona' OR city = 'London';
```

Имеется и более простой способ получить ту же информацию:

```
SELECT * FROM Salespeople WHERE city IN ('Barcelona', 'London');
```

2) оператор **BETWEEN** похож на оператор IN. В отличие от определения по номерам из набора, как это делает IN, BETWEEN определяет диапазон, значения которого должны уменьшаться, что делает предикат верным. Вы должны ввести ключевое слово BETWEEN с начальным значением, ключевое **AND** и конечное значение. В отличие от IN, BETWEEN чувствителен к порядку, и первое значение в предложении должно быть первым в алфавитном или числовом порядке. Обратите внимание, что, в отличие от английского языка, SQL не говорит "*значение находится между (BETWEEN) значением и значением*", а просто "*значение BETWEEN значение AND значение*". Следующий пример будет извлекать из таблицы Продавцов всех продавцов с комиссионными между .10 и .12:

```
SELECT * FROM Salespeople WHERE comm BETWEEN .10 AND .12;
```

Обратите внимание, что данный оператор **не включает** в результат вывода нижнюю границу символьного выражения (таков уж его алгоритм работы).

В SQLимеется оператор **REGEXP**, который может сделать то, чего не делает **BETWEEN.**

3) оператор **LIKE** применим только к полям типа CHAR или VARCHAR, с которыми он используется, чтобы находить подстроки. То есть, он ищет поле символа, чтобы видеть, совпадает ли с условием часть его строки. В качестве

условия он использует групповые символы (wildcards) — специальные символы которые могут соответствовать чему-нибудь. Имеются два типа групповых символов используемых с LIKE:

– символ подчеркивания (_) замещает любой одиночный символ. Например, '**b_t**' будет соответствовать словам '**bat**' или '**bit**', но не будет соответствовать 'brat';

– знак процента (**%**) замещает последовательность любого числа символов (включая символы нуля). Например '**%p%t**' будет соответствовать словам '**put',** '**posit',** или '**opt',** но не '**spite**'. Давайте найдем всех заказчиков, чьи имена начинаются с G :

```
SELECT * FROM Customers WHERE cname LIKE 'G%';
```

Индивидуальное задание:

1 Изучите основные операторы и перепишите их назначение в тетрадь.

2 Используя команды SQL выполните задания.

Часть 1:

1) выведите города Заказчиков без повторений;
2) выведите имеющиеся рейтинги Заказчиков без повторений;
3) выведите имеющиеся комиссионные Продавцов без повторений;
4) выведите все поля из таблицы Продавцы при условии, что продавец из Лондона;
5) выведите все поля из таблицы Продавцы при условии, что комиссия продавца больше 0.12%;
6) выведите все поля из таблицы Заказчики при условии, что заказчик из Рима;
7) выведите все поля из таблицы Заказчики при условии, что рейтинг заказчика 100 и 300;
8) выведите все поля из таблицы Заказы при условии, что заказ меньше 1000;
9) выведите все поля из таблицы Заказчики при условии, что заказчик из Рима с рейтингом 200;
10) выведите все поля из таблицы Заказчики при условии, что заказчик или из Берлина или с рейтингом 300.

Часть 2:

1) выберите все поля из таблицы Заказчики, в которых заказчиков обслуживали продавцы 1001, 1002 и 1007;
2) выберите все поля из таблицы Заказы, в которых заказы оформляли заказчики 2001, 2004 и 2008;
3) выберите из таблицы Продавцы все записи, где номер продавца от 1001 до 1005;
4) выберите из таблицы Продавцы все записи, у которых коэффициент продавца от 10% до 13%;
5) выберите из таблицы Заказчики все записи где имя заказчика от буквы A до G;
6) выберите из таблицы Заказчики все записи где заказчики проживают в городах от буквы A до L;
7) выберите из таблицы Заказы все заказы, которые были оформлены с 10.03.2021 по 10.05.2021;
8) выведите все данные о заказчиках, имя которых начинается с буквы 'C';
9) выведите все данные о заказчиках, имя которых начинается с буквы 'G';
10) выведите все данные о заказчиках, имя которых начинается с буквы 'G' и заканчивается 's'.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №12

Использование агрегатных функций, операторов группировки и упорядочивания данных в SQL

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Часть 1 – Использование агрегатных функций

Краткие теоретические сведения

Агрегатные (групповые) функции производят одиночное значение для всей группы таблицы. Имеется список этих функций:

Функция	Назначение
COUNT	считает число значений в данном столбце, или число строк в

	таблице. Когда она считает значения столбца, она используется с DISTINCT, чтобы производить счет чисел различных значений в данном поле Чтобы подсчитать общее число строк в таблице, используйте функцию COUNT созвездочкойвместоимениполя SELECT COUNT(*) FROM Customers
SUM	производит арифметическую сумму всех выбранных значений данного поля
AVG	производит усреднение всех выбранных значений данного поля
MIN	производит наименьшее из всех выбранных значений данного поля
MAX	производит наибольшее из всех выбранных значений данного поля
ROUND	Функция округления результата работы агрегатной функции **round (агрегатная функция(столбец_таблицы), n)** n – число, до которого нужно округлять дробную часть

Все эти функции возвращают единственное значение. При этом функции **COUNT, MIN** и **MAX** применимы к любым типам данных, в то время как **SUM** и **AVG** используются только для числовых полей. Разница между функцией **COUNT(*)** и **COUNT(<имя поля>)** состоит в том, что вторая при подсчете не учитывает NULL-значения.

SELECT SUM(amt) FROM Orders;

Переименование столбцов в запросе

В результате выборки данных с использованием агрегатных функций, столбец с результатом имеет имя такое же как и сама агрегатная функция, которая была использщована в этом запросе. Функция AS позволяет переименовать столбец в заголовке которого было название с агрегатной функцией. Иными словами AS даёт псевдоним столбцу. Предыдущий пример изменим с использованием оператора AS.

SELECT SUM(amt) AS Суммаприобретений FROM Orders;

Аргумент ALL

Агрегатные функции могут также (в большинстве реализаций) использовать аргумент **ALL**, который помещается перед именем поля, подобно DISTINCT, но означает противоположное — включать дубликаты.

Следующая команда подсчитает (**COUNT**) число не NULL значений в поле rating в таблице Заказчиков (включая повторения):

SELECT COUNT(ALL rating) FROM Customers;

Агрегаты, построенные на скалярном выражении

Возможно использовать агрегатные функции с аргументами, которые состоят из скалярных выражений, включающих одно или более полей. Предположим, что таблица Заказов имеет еще один столбец, который хранит предыдущий неуплаченный баланс (поле blnc) для каждого заказчика. Вы должны найти этот текущий баланс, добавлением суммы приобретений к предыдущему балансу. Вы можете найти наибольший неуплаченный баланс следующим образом:

```
SELECT MAX(blnc + amt) FROM Orders;
```

Скалярное выражение на основе выбираемых полей

SQL позволяет вам помещать скалярные выражения и константы среди выбранных полей. Эти выражения могут дополнять или замещать поля в предложениях SELECT, и могут включать в себя одно или более выбранных полей.

```
SELECT snum, sname, city, comm * 100 FROM Salespeople;
```

```
==============  SQL Execution Log ====
| SELECT snum, sname, city, comm * 100
| FROM Salespeople;
| ======================================
|  snum     sname      city
| ------   ---------  -----------   ---------
|  1001     Peel       London      12.000000
|  1002     Serres     San Jose    13.000000
|  1004     Motika     London      11.000000
|  1007     Rifkin     Barcelona   15.000000
|  1003     Axelrod    New York    10.000000
 ======================================
```

Рисунок – Помещение выражения в вашем запросе

Столбцы вывода

Столбцы вывода – это столбцы данных созданные запросом способом, иным чем просто извлечение их из таблицы. Вы создаете их всякий раз, когда вы используете агрегатные функции, константы, или выражения в предложении SELECT запроса.

Помещение текста в вашем выводе запроса

Вы можете вставлять константы в предложение SELECT запроса, включая и текст. Однако символьные константы, в отличие от числовых констант, не могут использоваться в выражениях.

Тем ни менее, возможность вставлять текст в вывод ваших запросов очень удобная штука.

```
SELECT snum, sname, city, ' % ', comm * 100 FROM Salespeople;
```

```
==============  SQL Execution Log =======
| SELECT snum, sname, city, '%', comm * 100
| FROM Salespeople;
| ====================================
|  snum  sname     city
```

```
|------  --------  -----------  ----  ---------
|  1001   Peel      London       %   12.000000
|  1002   Serres    San Jose     %   13.000000
|  1004   Motika    London       %   11.000000
|  1007   Rifkin    Barcelona    %   15.000000
|  1003   Axelrod   New York     %   10.000000
 ========================================
```

Рисунок – Вставка символов в ваш вывод

```
SELECT ' For ', odate, ', there are ',COUNT(DISTINCT onum), 'orders.'
FROM Orders GROUP BY odate;
```

```
==============  SQL Execution Log ==============
| SELECT 'For', odate, ', ' there are ' ,
| COUNT (DISTINCT onum), ' orders '
| FROM Orders
| GROUP BY odate;
| ============================================== |
|         odate
|------  ----------   ---------  ------  -------
|  For   10/03/1990 , there are       5  orders.
|  For   10/04/1990 , there are       2  orders.
|  For   10/05/1990 , there are       1  orders.
|  For   10/06/1990 , there are       2  orders.
================================================
```

Рисунок – Комбинация текста, значений поля, и агрегатов

Индивидуальное задание

Теоретическая часть:

1 Перечислите и расшифруйте агрегатные функции.
2 Чем отличаются функции COUNT(*) и COUNT(<имя поля>)?
3 Какая функция позволяет переименовать столбец-результат?
4 В чём назначение аргумента **ALL**?
5 Как можно использовать скалярные выражения в запросах?
6 Как строиться скалярное выражение на основе выбираемых полей?
7 Что подразумевают собой столбцы выводы?
8 Каким образом происходит помещение текста в вашем выводе запроса?

Практическая часть:

С помощью команд выполните следующие задания:

1 Покажите сколько продавцов в таблице Продавцы? Столбец с результатом назовите «Количество продавцов»;
2 Покажите из скольки городов продавцы? Столбец с результатом назовите «Города продавцов»;
3 Выведите сколько строк в таблице Заказчики? Столбец с результатом назовите «Количество строк»;
4 Выведите общий рейтинг заказчиков? Столбец с результатом назовите «Общий рейтинг заказчиков»;
5 Выведите на какую сумму сделаны все заказы? Столбец с результатом назовите «Сумма заказов»;
6 Выведите средний рейтинг заказчиков? Столбец с результатом назовите

«Средний рейтинг заказчиков»;
7 Выведите среднюю стоимость заказов? Столбец с результатом назовите «Средняя стоимость заказа»;
8 Выведите минимальный рейтинг заказчиков? Столбец с результатом назовите «Минимальный рейтинг заказчика»;
9 Выведите максимальную комиссию продавцов? Столбец с результатом назовите «Максимальная комиссия продавца»;
10 Посчитайте общую стоимость заказа с учётом доставки. Столбец с результатом назовите «Общая стоимость заказа»;
11 Выведитеиз таблицы Заказчики поля: Имя, Город и Рейтинг, уменьшенный в 50 раз;
12 Выведитеиз таблицы Продавцы поля: Имя, Город, синус комиссионных, увеличенный в 10 раз;
13 Выведитестроковую константу, поясняющую поле 'cnum' из таблицы Заказчики, Номер заказчика, Имя заказчика и Город;
14 Выведите строковые константы, поясняющие поля 'snum' и 'comm' из таблицы Продавцы, Номер, Имя продавца, Город, Комиссионные.

Часть 2 – Группировка и упорядочивание данных в SQL

Часть 2.1 – Группировка

Предложение **GROUP BY** используется для группирования результата одного или нескольких столбцов.

Например, вы хотите найти наибольшую сумму приобретений, полученную каждым продавцом. Вы можете сделать раздельный запрос для каждого из них, выбрав MAX (amt) из таблицы Заказов для каждого значения поля snum. GROUP BY, однако, позволит Вам поместить их все в одну команду:

```
SELECT snum, MAX(amt) FROM Orders GROUP BY snum;
```

Вывод для этого запроса показывается ниже

```
SQLExecutionLog
```

Вы можете также использовать GROUP BY с многочисленными полями. Совершенствуя вышеупомянутый пример далее, предположим, что вы хотите

```
SELEC
```

увидеть наибольшую сумму приобретений получаемую каждым продавцом каждый день. Чтобы сделать это, вы должны сгруппировать таблицу Заказов по продавцам и датам, и применить функцию MAX к каждой такой группе, подобно этому:

```
SELECTsnum, odate, MAX (amt)
FROM Orders GROUP BY snum, odate;
```

Вывод для этого запроса показывается ниже

```
SQLExecutionLog

SELECT snum, odate, MAX(amt)
FROM Orders GROUP BY snum, odate;
```

HAVING

Если предложение **WHERE** определяет предикат **для фильтрации строк**, то предложение **HAVING** применяется **после группировки** для определения аналогичного предиката, фильтрующего группы по значениям агрегатных функций. Это предложение необходимо для проверки значений, которые получены с помощью **агрегатной функции** не из отдельных строк источника записей, определенного в предложении **FROM**, а из **групп таких строк**. Поэтому такая проверка не может содержаться в предложении **WHERE**.

Заметим, что в предложении HAVING нельзя использовать **псевдоним**, используемый для именования значений **агрегатной функции**. Дело в том, что **предложение SELECT**, формирующее выходной набор запроса, выполняется предпоследним перед **предложением ORDER BY**.

Предположим, что вы хотели бы увидеть только максимальную сумму приобретений, значение которой выше $3000.00. Но исходя из выше написанного Вы не сможете сделать что-нибудь подобно следующему:

```
SELECT snum, odate, MAX(amt)
FROM Oreders
WHERE MAX(amt) > 3000.00
GROUP BY snum, odate;
```

Это будет отклонением от строгой интерпретации ANSI. Чтобы увидеть максимальную стоимость приобретений свыше $3000.00, вы можете использовать предложение HAVING. Предложение HAVING определяет критерии, используемые чтобы удалять определенные группы из вывода, точно также как предложение WHERE делает это для индивидуальных строк. Правильнойкомандойбудетследующая:

```
SELECT snum, odate, MAX(amt)
FROM Orders
GROUP BY snum, odate
HAVING MAX(amt) > 3000.00;
```

Вывод для этого запроса показывается ниже.

```
=============== SQL Execution Log =======
| SELECT snum, odate, MAX (amt)
| FROM Orders
| GROUP BY snum, odate
| HAVING MAX (amt) > 3000.00;
| ====================================
|  snum       odate
| ------    ----------    --------
|  1001     10/05/1990     4723.00
|  1001     10/06/1990     9891.88
|  1002     10/03/1990     5160.45
 ====================================
```

Аргументы в предложении HAVING должны иметь одно значение на группу вывода. Следующаякомандабудетзапрещена:

```
SELECT snum, MAX(amt)
FROM Orders
GROUP BY snum
HAVING odate = 10/03/1988;
```

Поле odate не может быть вызвано предложением HAVING, потому что оно может иметь (и действительно имеет) больше чем одно значение на группу вывода. Чтобы избегать такой ситуации, предложение HAVING должно ссылаться только на агрегаты и поля выбранные GROUP BY. Имеется правильный способ сделать вышеупомянутый запрос:

```
SELECT snum, MAX(amt)
FROM Orders
WHERE odate = 10/03/1990
GROUP BY snum;
```

Следующий пример показывает наибольшие Заказы для Serres и Rifkin:

```
SELECT snum, MAX(amt)
FROM Orders
GROUP BY snum
HAVING snum IN (1002,1007);
```

```
=============== SQL Execution Log =======
| SELECT snum, MAX (amt)
| FROM Orders
| GROUP BY snum
| HAVING snum IN (1002, 1007);
| ====================================
|   snum
| ------     --------
|   1002       5160.45
|   1007       1098.16
 ====================================
```

Рисунок – Использование HAVING с полями GROUPBY

Часть 2.2 – Упорядочивание

Таблицы — это неупорядоченные наборы данных, и данные, которые выходят из них, не обязательно появляются в какой-то определенной последовательности. SQL использует команду **ORDER BY**, чтобы позволять вам упорядочивать ваш вывод. Эта команда упорядочивает вывод запроса согласно значениям в том или ином количестве выбранных столбцов. Многочисленные столбцы упорядочиваются один внутри другого, также как с GROUP BY, и вы можете определять возрастание (ASC) или убывание (DESC) для каждого столбца. По умолчанию установлено — возрастание.

```
SELECT * FROM Orders ORDER BY cnum DESC;
```

```
SQL Execution Log

SELECT * FROM Orders ORDER BY cnum DESC;
```

Упорядочение по нескольким столбцам

Мы можем также упорядочивать таблицу по другому столбцу.

```
SELECT * FROM Orders ORDER BY cnum DESC, amt DESC;
```

```
=============== SQL Execution Log ===========
| SELECT *
| FROM Orders
| ORDER BY cnum DESC, amt DESC;
|============================================
|  onum      amt      odate      cnum     snum
| ------  --------  ----------  -----    -----
|  3006    1098.16  10/03/1990   2008     1007
|  3001      18.69  10/03/1990   2008     1007
|  3002    1900.10  10/03/1990   2007     1004
|  3011    9891.88  10/06/1990   2006     1001
|  3008    4723.00  10/05/1990   2006     1001
|  3010    1309.95  10/06/1990   2004     1002
|  3007      75.75  10/04/1990   2004     1002
|  3005    5160.45  10/03/1990   2003     1002
|  3009    1713.23  10/04/1990   2002     1003
|  3003     767.19  10/03/1990    2001     1001
 ============================================
```

Вы можете использовать ORDER BY таким же способом сразу с любым числом столбцов. Обратите внимание что, во всех случаях, столбцы которые упорядочиваются должны быть указаны в выборе SELECT.

Упорядочение агрегатных групп

ORDER BY может использоваться с GROUP BY для упорядочения групп.

```
SELECT snum, odate, MAX (amt)
FROM Orders GROUP BY snum, odate
ORDER BY snum;
```

```
=============== SQL Execution Log ============
| SELECT snum, odate, MAX (amt)
| FROM Orders
| GROUP BY snum, odate
| ORDER BY snum ;
|============================================
|   snum      odate       amt
|  -----    ----------  --------
|   1001    10/06/1990    767.19
|   1001    10/05/1990   4723.00
|   1001    10/05/1990   9891.88
|   1002    10/06/1990   5160.45
|   1002    10/04/1990     75.75
|   1002    10/03/1990   1309.95
|   1003    10/04/1990   1713.23
|   1004    10/03/1990   1900.10
|   1007    10/03/1990   1098.16
 ============================================
```

Упорядочение вывода по номеру столбца

Вместо имен столбцов вы можете использовать их порядковые номера для указания поля, используемого в упорядочении вывода. Эти номера могут ссылаться не на порядок столбцов в таблице, а на их порядок в выводе.

```
SELECT sname, comm FROM Salespeople GROUP BY 2 DESC;
```

```
=============== SQL Execution Log ==========
| (SELECT sname, comm
| FROM Salespeople
| ORDER BY 2 DESC;
|=======================================
|   sname      comm
|  --------  --------
|  Peel         0.17
|  Serres       0.13
|  Rifkin       0.15
 =======================================
```

Одна из основных целей этой возможности ORDER BY — дать вам возможность использовать GROUP BY со столбцами вывода также как и со столбцами таблицы. Столбцы, производимые агрегатной функцией, константы или выражения в предложении SELECT запроса, абсолютно пригодны для использования с GROUP BY, если они ссылаются к ним с помощью номера.

```
SELECT snum, COUNT (DISTINCT onum)
FROM Orders GROUP BY snum
ORDER BY 2 DESC;
```

```
=============== SQL Execution Log ============
| SELECT snum, odate, MAX (amt)               |
| FROM Orders                                 |
| GROUP BY snum                               |
| ORDER BY 2 DESC;                            |
|=============================================
|   snum                                      |
|  -----    ----------                        |
|   1001          3                           |
|   1002          3                           |
|   1007          2                           |
|   1003          1                           |
|   1004          1                           |
 =============================================
```

В этом случае, вы должны использовать номер столбца, так как столбец вывода не имеет имени; и вы не должны использовать саму агрегатную функцию.

Упорядочение с помощью оператора NULL

Если имеются пустые значения (NULL) в поле, которое вы используете для упорядочивания вашего вывода, они могут или следовать, или предшествовать каждому другому значению в поле.

Индивидуальное задание

Теоретическая часть:

4) Каково назначение GROUP BY?
5) Для чего необходимо предложение HAVING?
6) С помощью какого оператора производится упорядочивание данных?
7) Что означают операторы ASC и DESC?

8) Каким образом можно производить упорядочивание по нескольким столбцам?
9) Каким образом можно производить упорядочивание групп?
10) Как производить упорядочение вывода по номеру столбца?

Практическая часть 1:

1 Напишите запрос, который сосчитал бы число различных, отличных от NULL значений поля city в таблице Заказчиков.

2 Напишите запрос, который выбрал бы наименьшую сумму заказа для каждого заказчика.

3 Напишите запрос, который выбрал бы наименьшую сумму заказа каждого продавца за каждый день.

4 Напишите запрос, который показывает наименьшую стоимость заказа для заказчиков Cisneros, Grass и Clemens.

5 Напишите запрос, который выведет имена заказчиков в алфавитном порядке, но только тех, чьи имена начинаются с буквы G.

6 Напишите запрос, который выведет высшийрейтинг в каждом городе.

7 Напишите запрос, который сосчитал бы число Продавцов, регистрирующих свои Заказы, за каждый день (если продавец имел более одного Заказа в данный день, он должен учитываться только один раз).

8 Напишите запрос, который сосчитал бы число Заказов за каждый день для каждого Продавца.

Практическая часть2:

1 Выведите названия всех заказчиков относительно рейтинга, т. е. по каждому рейтингу должны быть выведены те заказчики, которые к нему относятся.

2 Выведите заказчиков, которых обслуживает конкретный продавец, т. е. по каждому продавцу должен быть выведен список заказчиков, с которыми он работает.

3 Вывести из таблицы Продавцы поля Номер, Имя, Город упорядоченные по убыванию по полю 'sname'.

4 Вывести из таблицы Заказчики поля Имя, Город, Рейтинг упорядоченные по возрастанию по полям 'rating' и 'cname'.

5 Вывести из таблицы Продавцы поля Имя, Город, сгруппированные по полю Город, и упорядоченные по полю Имя.

6 Вывести из таблицы Заказчики поля Имя, Город, Рейтинг, сгруппированные по 2-му столбцу.

7 Вывести из таблицы Заказчики поля Город и количество рейтингов каждого города упорядоченных по возрастанию.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Тема 2.2
Оптимизация запросов

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Текст задания по теоретической части 1

Напишите SQLкоманду на поставленный вопрос.

1 Вариант

1) Покажите все заказы (номер заказа и его стоимость), которые оформил продавец *Peel*(имя Peel так же выведите)
2) Покажите заказчиков(номер и имя), с которыми работали продавцы (номер и имя) с комиссионными 0.10 и 0.15 (0.16) (комиссионные так же выведите)
3) Покажите общую стоимость заказов заказчиков с рейтингом 200
4) Вывести имена всех продавцов и среднюю стоимость их заказов
5) Покажите имя продавца и имя заказчика, чьи заказы сделаны в марте на сумму свыше 2000р.

2 Вариант

1. Покажите все заказы, которые оформилзаказчик*Hoffman*(вывести имя продавца, сумму его заказа и имя заказчика)
2. Покажите имена продавцов, которые работали с заказчиками чьи имена на букву

«С»
3. Покажите максимальную стоимость заказа продавца *Peel*(имя продавца так же выведите)
4. Выведите имена тех заказчиков, которые оформляли заказы летом.
5. Покажите общую стоимость заказов в каждом городе продавцов (город тоже вывести)

3 Вариант

1) Покажите все заказы (номер заказа и его стоимость), которые оформил продавец с комиссией 0.10 (комиссию тоже вывести)
2) Покажите заказчиков (номер заказчика и его имя), с которыми работали продавцы либо из *Barcelona* либо с комиссией 0.13 (город и комиссионные так же вывести)
3) Покажите среднюю стоимость заказов для каждого заказчика (имя так же вывести) из городов *Rome*и *Berlin*(название города так же вывести)
4) Вывести имена всех заказчиков и максимальную стоимость их заказов
5) Покажите заказчика (имя заказчика), который работал с продавцом из *London* в июне месяце (город и дату так же вывести)

4 Вариант

1) Покажите все заказы (номер заказа), которые оформилзаказчикс рейтингом 300
2) Покажите продавцов (номер продавца и его имя), которые работали с заказчиками из *Rome* с рейтингом меньше 200 (город и рейтинг так же вывести)
3) Покажите заказ (номер заказа) с минимальной суммой приобретения, который оформил продавец с комиссионными 0.12 (комиссионные так же вывести)
4) Покажите имена тех продавцов, которые оформляли заказы (номер заказа) на сумму от 1000р. до 5000р. Летом (сумму заказа и даты та же вывести).
5) Покажите продавца (имя), который работал с заказчиком с рейтингом больше 200, весной (рейтинг и даты так же вывести)

Текст задания по теоретической части 2

Напишите SQLкоманду на поставленный вопрос.

1 Вариант

1. Покажите продавцов (номер и имя), которые оформили больше двух заказов (кол-во заказов так же вывести)
2. Покажите заказчиков (номер и имя), которые оформили больше одного заказа в каждом месяце (кол-во заказов и дату так же вывести)
3. Покажите город заказчика, у которого самый большой суммарный рейтинг (вывести название города и его максимальный суммарный рейтинг)
4. Покажите заказы (номер и суму) заказчиков, которые проживают в городе с максимальным суммарным рейтингом.
5. Покажите имя заказчика и стоимость его заказа, которые проживают в городе с максимальным суммарным рейтингом

2 Вариант

1. Покажите продавца, у которого самая большая суммарная стоимость заказа (вывести имя продавца и общую сумму его покупок)
2. Покажите имена продавцов, которые работали с заказчиками чьи имена на букву «С»
3. Покажите максимальную стоимость заказа продавца *Peel* (имя продавца так же выведите)
4. Выведите имена тех заказчиков, которые оформляли заказы летом.
5. Покажите общую стоимость заказов в каждом городе продавцов при условии, что эта общая стоимость выше средней стоимости всех заказов (город тоже вывести)

Критерий оценки

Каждый полный правильный ответ оценивается в1 балл.
0-2 правильных ответа – «2»
3 правильных ответа – «3»
4 правильных ответа – «4»
5 правильных ответов – «5»

Практическое занятие №13

Реализация сложных запросов. Join.

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

Часть 1 – Реализация сложных запросов

Теоретические сведения

Одна из наиболее важных особенностей запросов SQL — это их способность определять связи между различными таблицами и выводить информацию из них в терминах этих связей, всю внутри одной команды.

Этот вид операции называется — *объединением*, которое является одним из видов операций в реляционных базах данных.

Имена таблиц и столбцов

Полное имя столбца таблицы фактически состоит из имени таблицы, сопровождаемого точкой и затем именем столбца <имя таблицы>.<имя столбца>. Имеются несколько примеров имен

```
Salespeople.snum
Salespeople.city
Orders.odate
```

Создание объединения

Предположим, что вы хотите поставить в соответствии вашему продавцу ваших заказчиков в городе, в котором они живут, поэтому вы увидите все комбинации продавцов и заказчиков для этого города. Вы будете должны брать каждого продавца и искать в таблице Заказчиков всех заказчиков того же самого города. Вымоглибысделатьэто, введясследующуюкоманду:

```
SELECT Customers.cname, Salespeople.sname, Salespeople.city
FROM Salespeople, Customers
WHERE Salespeople.city = Customers.city;
```

```
================  SQL Execution Log ============
| SELECT Customers.cname, Salespeople.sname,
| Salespeople.city
| FROM Salespeople, Customers
| WHERE Salespeople.city = Customers.city
| ============================================== |
|  cname      cname          city
| -------    --------       ----
| Hoffman    Peel           London
| Clemens    Peel           London
| Hoffman    Motika         London
| Clemens    Motika         London
| Liu        Serres         San Jose
| Cisneros   Serres         San Jose
 ================================================
```

Объединение таблиц через справочную целостность

Эта особенность часто используется просто для эксплуатации связей встроенных в базу данных. В предыдущем примере, мы установили связь между двумя таблицами в объединении. Это прекрасно. Но эти таблицы уже были соединены через поле snum. Эта связь называется *состоянием справочной целостности*. Используя объединение, можно извлекать данные в терминах этой связи. Например, чтобы показать имена всех заказчиков соответствующих

продавцам, которые их обслуживают, мы будем использовать такой запрос:

```
SELECT Customers.cname, Salespeople.sname
FROM Customers, Salespeople
WHERE Salespeople.snum = Customers.snum;
```

```
=============== SQL Execution Log ============
| SELECT Customers.cname, Salespeople.sname,
| FROM Salespeople, Customers
| WHERE Salespeople.snum = Customers.snum
|=============================================
|  cname       sname
| -------     --------                    |
| Hoffman     Peel
| Giovanni    Axelrod
| Liu         Serres
| Grass       Serres
| Clemens     Peel
| Cisneros    Rifkin
| Pereira     Motika
 =============================================
```

Объединения таблиц по равенству значений в столбцах и другие виды объединений

Объединения, которые используют предикаты основанные на равенствах называются — *объединениями по равенству*. Все наши примеры относились именно к этой категории, потому что все условия в предложениях WHERE базировались на математических выражениях использующих знак равно (=). Строки **'city='London'** и **'Salespeople.snum=Orders.snum'** — примеры таких типов равенств, найденных в предикатах.

Объединения по равенству — это вероятно наиболее общий вид объединения, но имеются и другие. Вы можете, фактически, использовать любой из реляционных операторов в объединении. Здесьпоказанпримердругоговидаобъединения:

```
SELECT sname, cname
FROM Salespeople, Customers
WHERE sname < cname AND rating < 200;
```

```
=============== SQL Execution Log ===========
| SELECT sname, cname
| FROM Salespeople, Customers
| WHERE sname < cname
| AND rating < 200;
|============================================
|    sname       cname
|   --------    -------
|   Peel         Pereira
|   Motika       Pereira
|   Axelrod      Hoffman
|   Axelrod      Clemens
|   Axelrod      Pereira
 ============================================
```

Объединение более двух таблиц

Вы можете также создавать запросы, объединяющие более двух таблиц. Предположим, что мы хотим найти все Заказы заказчиков не находящихся в тех городах где находятся их продавцы. Для этого необходимо связать все три наши типовые таблицы:

```
SELECT onum, cname, Orders.cnum, Orders.snum
FROM Salespeople, Customers, Orders
WHERE Customers.city <> Salespeople.city
AND Orders.cnum = Customers.cnum
AND Orders.snum = Salespeople.snum;
```

```
========== SQL Execution Log ==========
| SELECT onum, cname, Orders.cnum, Orders.snum
| FROM Salespeople, Customers, Orders
| WHERE Customers.city <> Salespeople.city
| AND Orders.cnum = Customers.cnum
| AND Orders.snum = Salespeople.snum;
| ========================================
|  onum   cname      cnum    snum
| ------ -------    -----   -----
|  3001  Cisneros    2008    1007
|  3002  Pereira     2007    1004
|  3006  Cisneros    2008    1007
|  3009  Giovanni    2002    1003
|  3007  Grass       2004    1002
|  3010  Grass       2004    1002
 ========================================
```

Индивидуальное задание

Теоретическая часть:

1) Что подразумевает операция объединения таблиц?
2) Из чего формируется полное имя столбца?
3) Каким образом создаётся объединение?
4) Что означает объединение по равенству?

Практическая часть

1) Вывести фамилию продавца, фамилию покупателя и сумму покупки между 700 и 2000,.

2) Вывести фамилию продавцаи сумму его покупки без комиссии.
3) Вывести фамилию продавцов и соответствующих им фамилии покупателей.
4) Вывести фамилию продавцов и фамилию покупателей, которые из одного города (поле Название города так же вывести).
5) Вывести поле дата, где комиссия продавца меньше 0.12 (поле Комиссия так же вывести).
6) Вывести поле Сумма заказа покупателя, у которого рейтинг больше 100 (поле Рейтинг так же вывести).
7) Вывести имя заказчика, у которого заказ совершен 10.03.2021 (поле Дата так же вывести).

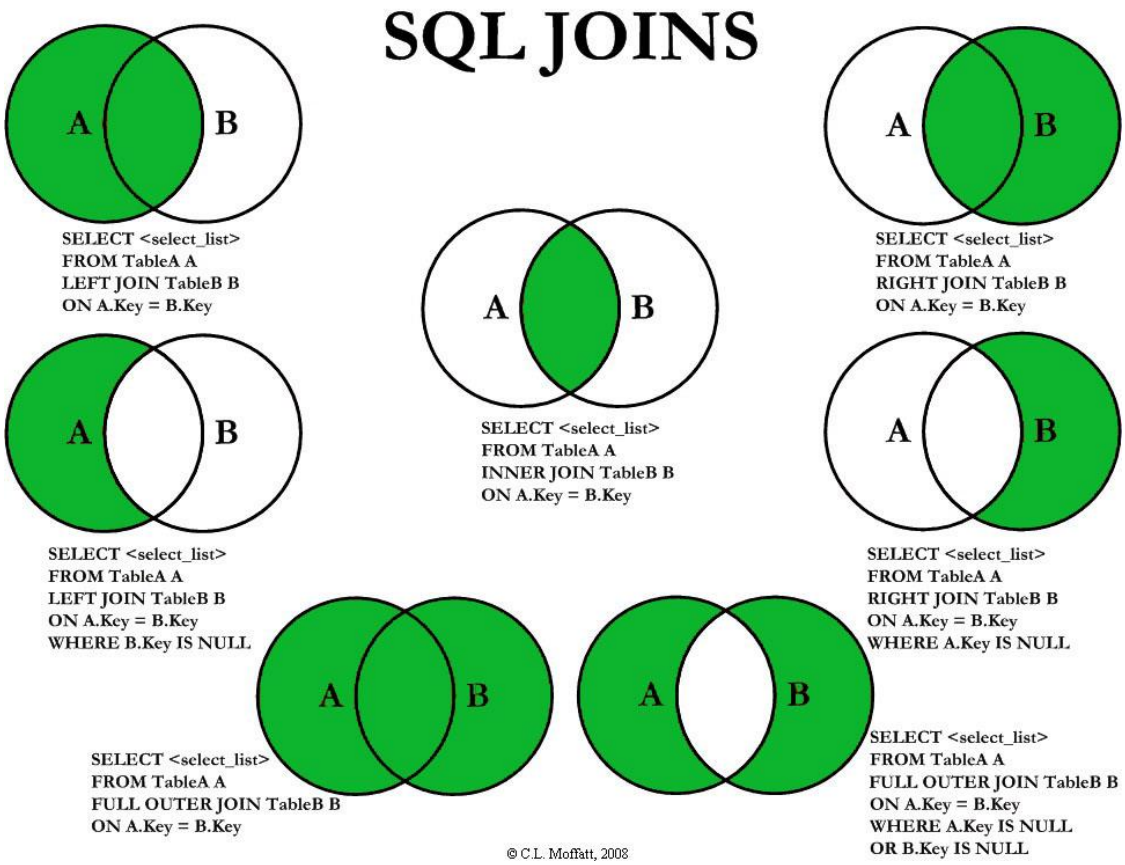
8) Вывести имя покупателя, сумму его заказа и комиссионные продавца за 10.04.2021 (поле Дата так же вывести).

9) Вывести имя и рейтинг заказчиков, у которых сумма заказа больше 3000 и покупка совершена 10.03.2021 (поля Сумма заказа и Дата так же вывести).

10) Вывести номера заказов, продавцы и заказчики которых живут в Сан Джосе (поля ФИО заказчика, ФИО продавца и Название города так же вывести).

Покажите работу преподавателю.

Часть 2 – Работа с JOIN в MySQL



Запросы без JOIN	Запросы с JOIN
Раунд 1. Вывести все заказы с информацией по их заказчикам	
SELECT * FROM orders,customers WHERE orders.cnum = customers.cnum	SELECT * FROM orders INNER JOIN customers ON orders.cnum = customers.cnum ЛИБО ЕЩЁ ПРОЩЕ SELECT * FROM orders INNER JOIN customers USING (cnum)
(никакой особой разницы)	

<table>
<tr><td colspan="2">Раунд 2. Вывести номер, имя продавца и имя заказчика для каждого заказа, у которого сумма больше 2000</td></tr>
<tr><td>SELECT onum,sname,cname FROM orders,customers,salespeople WHERE orders.snum = salespeople.snum AND orders.cnum = customers.cnum AND amt > 2000</td><td>SELECT onum,sname,cname FROM orders INNER JOIN salespeople USING (snum) INNER JOIN customers USING (cnum) WHERE amt > 2000</td></tr>
<tr><td colspan="2">(небольшая разница, просто без JOIN удобнее добавить таблицы, но неудобнее их связать)</td></tr>
<tr><td colspan="2">Раунд 3. Вывести имя каждого заказчика и сумму его заказов</td></tr>
<tr><td>SELECT cname, (SELECT SUM(amt) FROM orders WHERE orders.cnum = customers.cnum) as 'CustomerTotal' FROM customers</td><td>SELECT cname, SUM(amt) as 'CustomerTotal' FROM customers LEFT OUTER JOIN orders USING (cnum) GROUP BY cnum</td></tr>
<tr><td colspan="2">(без JOIN придется делать дополнительные подзапросы для каждого запроса, затратно)</td></tr>
<tr><td colspan="2">Раунд 4. Вывести таблицу с соотношениями продавцов и заказчиков, также включающую продавцов и заказчиков, не имеющих соотношения</td></tr>
<tr><td>SELECT sname, cname FROM salespeople,customers WHERE salespeople.snum = customers.snum UNION SELECT sname, NULL as 'cname' FROM salespeople WHERE snum NOT IN (SELECT snum FROM customers) UNION SELECT NULL as 'sname', cname FROM customers WHERE snum IS NULL</td><td>*не поддерживается в MySQL* SELECT sname,cname FROM salespeople FULL OUTER JOIN customers USING (snum)</td></tr>
<tr><td colspan="2">(нормально достигается только через JOIN, но в MySQL не поддерживается)</td></tr>
<tr><td colspan="2">Итоги: JOINы победили</td></tr>
</table>

Задание для выполнения практической работы необходимо взять из предыдущей работы.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №14

Вложенные (коррелированные) запросы

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Краткие теоретические сведения

Рассмотрим работу подзапроса. Предположим, что мы знаем имя продавца: Motika, но не знаем значение его поля snum, и хотим извлечь все заказы из таблицы Заказы. Запросиегорезультатсмотритениже.

```
SELECT *
FROMOrders
WHEREsnum = (SELECTsnum
FROMSalespeople
WHEREsname = 'Motika');
```

SQL Execution Log

При использовании подзапросов в предикатах основанных на уравнениях или неравенствах, вы должны убедиться, что использовали подзапрос который будет выдавать одну и только одну строку вывода.

Если вы используете подзапрос, который не выводит никаких значенийвообще, команда выполнится, но основной запрос не выведет никаких значений.

DISTINCT с подзапросами

Вы можете, в некоторых случаях, использовать DISTINCT, чтобы вынудить подзапрос генерировать одиночное значение. Предположим что мы хотим найти все заказы кредитований для тех продавцов которые обслуживают Hoffmana (cnum = 2001).

```
SELECT *
FROM Orders
WHERE snum = ( SELECT DISTINCT snum
               FROM Orders
               WHERE cnum = 2001 );
```

```
=============  SQL Execution Log ==============
| SELECT *                                      |
| FROM Orders                                   |
| WHERE snum =                                  |
| (SELECT DISTINCT snum                         |
| FROM Orders                                   |
| Where cnum = 2001);                           |
|=================================================|
|  onum      amt      odate      cnum     snum  |
|  -----   ---------  ---------   ------  -------|
|  3003     767.19  10/03/1990   2001     1001  |
|  3008    4723.00  10/05/1990   2006     1001  |
|  3011    9891.88  10/06/1990   2006     1001  |
|=================================================
```

Подзапрос установил, что значение поля snum совпало с Hoffman — 1001, и затем основной запрос выделил все порядки с этим значением snum из таблицы Порядков (не разбирая, относятся они к Hoffman или нет). Так как каждый заказчик назначен к одному и только этому продавцу, мы знаем, что каждая строка в таблице Порядков с данным значением cnum должна иметь такое же значение snum. Однако, так как тамможет быть любое число таких строк, подзапрос мог бы вывести много (хотя и идентичных) значений snum для данного поля cnum. Аргумент DISTINCT предотвращает это.

Предикаты с подзапросами являются необратимыми

Вы должны обратить внимание, что предикаты, включающие подзапросы, используют выражение <скалярная форма><оператор><подзапрос>, а не <подзапрос><оператор><скалярное выражение> или, <подзапрос><оператор><подзапрос>.

Другими словами, вы не должны записывать предыдущий пример так:

```
SELECT *
FROM Orders
WHERE ( SELECT DISTINCT snum
        FROM Orders
        WHEREcnum = 2001 ) = snum;
```

Индивидуальноезадание

Теоретическая часть:

1 В чём преимущество подзапросов?

2 Какого назначение оператора DISTINCT в подзапросах?

Практическая часть:

1 Вывести из таблицы Заказы все значения полей у заказчика Hoffman.

2 Вывести заказы продавца из New York.

3 Вывести заказчиков, которыеоформляли заказу продавца Motika.

4 Вывести продавцов, которые работали с заказчиком Liu.

5 Вывести покупки заказчика с рейтингом 100 и городом Rome.

Покажите работу преподавателю.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №15

Использование агрегатных функций в подзапросах. Операторы ANY и ALL.

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Часть 1 – Агрегатных функций в подзапросах

Краткие теоретические сведения

Вы хотите увидеть все Заказы, имеющие сумму приобретений выше средней на 4-е Октября

```
SELECT *
FROM Orders
WHERE amt > (SELECT AVG (amt)
             FROM Orders
             WHERE odate = 10/04/1990);
```

```
===============  SQL Execution Log ==============
| SELECT *                                       |
| FROM Orders                                    |
| WHERE amt >                                    |
| (SELECT AVG (amt)                              |
|  FROM Orders                                   |
|  WHERE odate = 01/04/1990);                   |
| ============================================== |
|   onum       amt       odate      cnum     snum  |
|  -----    --------  ----------  -----    -----  |
|   3002     1900.10  10/03/1990   2007     1004  |
|   3005     5160,45  10/03/1990   2003     1002  |
|   3006     1098.19  10/03/1990   2008     1007  |
|   3009     1713.23  10/04/1990   2002     1003  |
|   3008     4723.00  10/05/1990   2006     1001  |
|   3010     1309.95  10/06/1990   2004     1002  |
|   3011     9891.88  10/06/1990   2006     1001  |
================================================
```

Средняя сумма приобретений на 10Апреля — 1788.98 (1713.23 + 75.75) делится пополам, что в целом равняется = 894.49. Все строки со значением в поле amt выше этого — являются выбранными.

Имейте в виду, что сгруппированные агрегатные функции, которые являются

агрегатными функциями, определенными в терминах предложения GROUP BY, могут производить многочисленные значения. Они, следовательно, не позволительны в подзапросах такого характера. Даже если GROUP BY и HAVING используются таким способом, что только одна группа выводится с помощью подзапроса, команда будет отклонена в принципе. Вы должны использовать одиночную агрегатную функцию с предложением WHERE, что устранит нежелательные группы. Например, следующий запрос, который должен найти среднее значение комиссионных продавца в Лондоне

```
SELECT AVG (comm)
FROM Salespeople
GROUP BY city
HAVING city = 'London;
```

не может использоваться в подзапросе! Во всяком случае, это не лучший способ формировать запрос. Другимспособомможетбыть

```
SELECT AVG (comm)
FROM Salespeople
WHERE city = 'London';
```

Использование подзапросов, которые выдают много строк с помощью оператора IN

Вы можете использовать подзапросы, которые производят любое число строк, если вы используете специальный оператор IN (операторы BETWEEN, LIKE, и IS NULL не могут использоваться с подзапросами). Например, мы можем найти все атрибуты таблицы Заказов для продавца в Лондоне

```
SELECT *
FROM Orders
WHERE snum IN (SELECT snum
               FROM Salespeople
               WHERE city = 'London');
```

```
==============  SQL Execution Log ==============
| SELECT *                                      |
| FROM Orders                                   |
| WHERE snum IN                                 |
| (SELECT snum                                  |
|  FROM Salespeople                             |
|  WHERE city = 'London');                      |
| ============================================= |
|   onum       amt       odate      cnum    snum  |
|  -----    --------  ----------  -----   ------  |
|   3003      767.19  10/03/1990   2001     1001  |
|   3002     1900.10  10/03/1990   2007     1004  |
|   3006     1098.19  10/03/1990   2008     1007  |
|   3008     4723.00  10/05/1990   2006     1001  |
|   3011     9891.88  10/06/1990   2006     1001  |
================================================
```

Подзапросы выбирают одиночные столбцы

Смысл всех подзапросов обсужденных в этой главе тот, что все они выбирают одиночный столбец. Это обязательно, поскольку выбранный вывод сравнивается с одиночным значением. Подтверждением этому то, что SELECT * не может использоваться в подзапросе.

Использование выражений в подзапросах

Вы можете использовать выражение, основанное на столбце, а не просто сам столбец, в предложении SELECT подзапроса. Это может быть выполнено или с помощью реляционных операторов, или с IN. Например, следующий запрос использует реляционный оператор = (вывод показывается в рисунке):

```
SELECT *
FROM Customers
WHERE cnum = (SELECT snum + 1000
                FROM Salespeople
               WHERE sname = 'Serres');
```

```
===============  SQL Execution Log ============
| SELECT *                                      |
| FROM Customers                                |
| WHERE cnum =                                  |
| (SELECT snum + 1000                           |
| WHERE Salespeople                             |
| WHERE sname = 'Serres'                        |
| ============================================= |
|   cnum     cname     city    rating    snum   |
|  -----    --------   ----    ------   -----   |
|   2002    Giovanni   Rome       200    1003   |
 ===============================================
```

Подзапросы в предложении HAVING

Вы можете также использовать подзапросы внутри предложения HAVING. Эти подзапросы могут использовать свои собственные агрегатные функции, если они не производят многочисленных значений или использовать GROUP BY или HAVING. Следующийзапросявляетсяэтомупримером (выводпоказываетсяврисунке):

```
SELECT rating, COUNT (DISTINCT cnum)
FROM Customers
GROUP BY rating
HAVING rating > (SELECT AVG (rating)
                   FROM Customers
                   WHERE city = 'San Jose');
```

```
================ SQL Execution Log ============
| SELECT rating,count (DISTINCT cnum)          |
| FROM Customers                               |
| GROUP BY rating                              |
| HAVING rating >                              |
| (SELECT AVG (rating)                         |
| FROM Customers                               |
| WHERE city = 'San Jose'                      |
|==============================================|
|  rating                                      |
| --------    --------                         |
|   300            2                           |
 ==============================================
```

Эта команда подсчитывает заказчиков с оценками выше среднего в San Jose. Так как имеются другие оценки отличные от 300, они должны быть выведены с числом номеров заказчиков, которые имели эту оценку.

Индивидуальное задание

Теоретическая часть

1 Для чего используют агрегатные функции в подзапросах?

Практическая часть

1 Напишите запрос, который вывел бы имена Заказчиков, их рейтинг и среднюю стоимость заказа.

2 Напишите запрос, который вывел бы имена Заказчиков из Рима, рейтинг которых ниже среднего.

3 Напишите запрос, который выводит комиссионные (comm) всех Продавцов, обслуживавших Заказчиков в Париже.

4 Написать запрос, который подсчитывает количество Заказов с суммой приобретения ниже среднего, совершенных 10 марта 2021.

5 Написать запрос, который выводит Продавца, уникальный номер которого на 1000 меньше уникального номера Заказчика, проживающего в Берлине.

6 Напишите запрос, который бы выбрал общую сумму всех приобретений в Заказах для каждого продавца, у которого эта общая сумма больше, чем сумма наибольшего Заказа в таблице.

Покажите работу преподавателю.

Часть 2 – Операторы ANY и ALL в подзапросах

Операторы ANY и ALL используются с подзапросами, когда подзапрос выдаёт больше одного значения (именно в таких случаях не всегда справляется оператор IN). В отличии от оператора IN, ANY и ALL могут использовать не только оператор « = », но и операторы « > », « < » и т.д.

Оператор ANY возвращает true, если какое-либо из значений подзапроса соответствует условию.

Оператор ALL возвращает true, если все значения подзапроса удовлетворяют условию.

Синтаксис ANY

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY
(SELECT column_name FROM table_name WHERE condition);
```

Синтаксис ALL

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ALL
(SELECT column_name FROM table_name WHERE condition);
```

Пример ANY

Вывести имена товаров, соответствующие ЛЮБЫМ записям в таблице info, с количеством = 15:

```
SELECT name
FROM product
WHERE product_id = ANY (SELECT product_id FROM info WHERE counts = 15);
```

Вывести имена товаров, соответствующие ЛЮБЫМ записям в таблице info, с количеством > 15:

```
SELECT name
FROM product
WHERE product_id = ANY (SELECT product_id FROM info WHERE counts > 15);
```

Пример ALL

Вывести имена товаров, соответствующие ВСЕМ записям в таблице info с количество = 7:

```
SELECT name
FROM product
WHERE product_id = ALL (SELECT product_id FROM info WHERE couns =7);
```

Пример сравнения результатов одного и того же за проса с операторами ALL и ANY.

Вывести все данные о Заказчиках, у которых рейтинг выше, чем рейтинг у заказчиков из Rome.

У заказчиков из Rome рейтинг равен 100 и 200.

cnum	cname	pol	city	rating	snum
2002	Giovanni	м	Rome	200	1003
2007	Pereira	м	Rome	100	1004

ANY выдаст и тех заказчиков, у которых рейтинг больше 100, и тех, у кого рейтинг больше 200, т.е. или 200 или 300.

```
SELECT * FROM Cust
WHERE rating > ANY (SELECT rating FROM Cust WHERE city ='Rome')
```

cnum	cname	pol	city	rating	snum
2002	Giovanni	м	Rome	200	1003
2003	Liu	ж	San Jose	200	1002
2004	Grass	м	Berlin	300	1002
2008	Cisneros	м	San Jose	300	1007
2009	Sara	ж	London	300	1001

ALL имеет более жёсткие условия, он выдаст только тех заказчиков, у которых рейтинг 300. По условию 200 > 100, но 200 < 300. Таким образом он выдаст только те, кто именно больше самого максимального из предложенных.

```
SELECT * FROM Cust
WHERE rating > ALL (SELECT rating FROM Cust WHERE city ='Rome')
```

cnum	cname	pol	city	rating	snum
2004	Grass	м	Berlin	300	1002
2008	Cisneros	м	San Jose	300	1007
2009	Sara	ж	London	300	1001

Практическое задание:

1 Показать тех Продавцов и их города, которые проживают в тех же городах, что и Заказчики.

2 Показать продавцов, у которых комиссионные выше чем у продавцов из Лондона.

3 Покажите заказы, которые дешевле стоимости заказов за 2021-04-10. (запрос должен показать только те значения, которые ниже amt самого дешёвого за эту дату)

4 Покажите номер продавца, у которого самая высокая суммарная стоимость заказов (суммарную стоимость заказа так же вывести).

5 Покажите имя и номер продавца, у которого самая высокая суммарная стоимость заказов (суммарную стоимость заказа так же вывести).

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Практическое занятие №16

Настройка параметров производительности базы данных. Импорт, экспорт данных.

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Импорт данных базы в документы пользователя

Импортировать данные в таблицы СУБД MySQL можно из разных источников. Самые распространённые – это.xlsx (Excel) и .txt (Блокнот).

Ниже рассмотрим реализацию из двух источников.

Рассмотрим алгоритм импорта

1.1 Создайте в MySQL таблицу с тем же количеством столбцов, как и в таблице, из которой будет происходить импорт.
// Обратите внимание, что типы данных столбцов обеих таблиц должны совпадать!

1.2 Преобразуем .xlsxили .txt в .csv
«Сохранить как» – «Другие форматы» – «Тип файла» – CSV(разделители – запятые) (Текстовый файл (с разделителем табуляции) – для преобразования в текстовый формат .txt)

1.3 Разместить файл импорта в доступный для импортирования системный каталог MySQL.

Посмотреть название каталога, разрешённого для импорта/экспорта можно двумя способами:

1) в конфигурационном файле …my.ini в параметре secure-file-priv;
2) узнать этот каталог с помощью команды

SHOW VARIABLES LIKE "secure_file_priv".

MySQL даёт возможность импорта/экспорта из файлов, только при условии, что эти файлы размещены в разрешённом сервером каталоге. Соответственно вам необходимо перенести файл для импорта именно туда.

1.4 Используем команду loaddatainfile для реализации импорта данных.

Оператор**LOAD DATA INFILE**позволяет считывать данные из текстового файла и очень быстро импортировать их в таблицу базы данных.

Формат команды:

LOADDATAINFILE «'полный путь к файлу импорта'»

// обратите внимание, при написании полного пути к файлу необходимо использовать как разделитель имён именно « / »

INTOTABLE «имя таблицы базы данных»

// указывает имя таблицы базы данных, в которую будет произведёт импорт

CHARACTER SET cp1251

// кодировка самого загружаемого CSV файла (очень важно для кириллицы)

FIELDSTERMINATEDBY ';'

// указываем, что является разделителем колонок/столбцов (**,** или **;**), посмотреть это можно открыв csv-файл с помощью любого редактора кода, в нашем случае .csv имеет разделитель « **;** », поэтому это мы и указываем в формате команды

ENCLOSEDBY '"'

// еслив исходном файле значения текстового поля заключены в двойные кавычки, то необходимо это указать команде

LINES TERMINATED BY '\r\n'

// разделение линий в фале импорта, \r\n – для табличного формата csv, /n – если для импорта используется блокнот, т.е. txt-файл

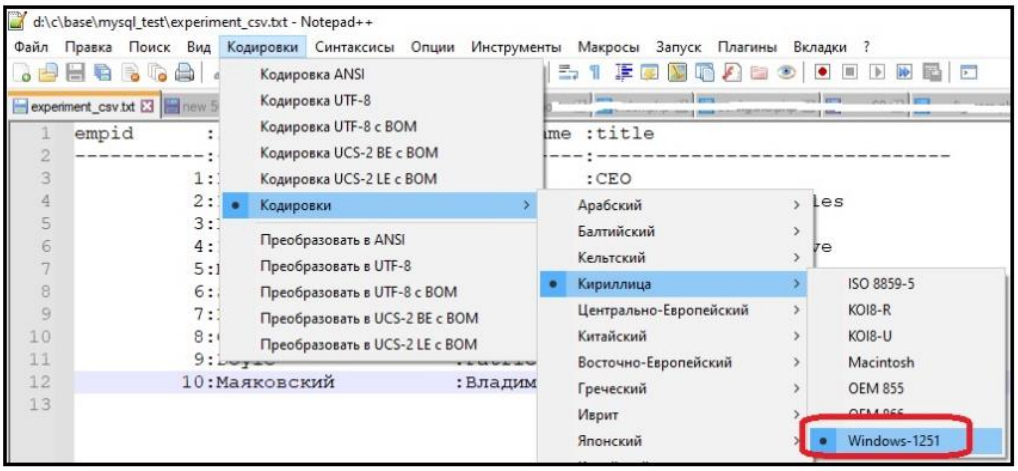
IGNORE 1 LINES

// пропускает первую строку в файле, т.к. она является заголовком

(«название столбца 1», «название столбца 2», «название столбца n»);

Для того, чтобы узнать кодировку вашего CSV файла нужно сделать так, как показано на рисунке.

Если кодировка иная, измените (так вы измените «кракозябры» на читаемый русский текст).



Если у вас таблица в MySQL создана с использованием параметра AUTO_INCREMENT в поле первичного ключа, то о его заполнении можно не беспокоится, MySQL его сам заполнит. Чтобы игнорировать импорт первого столбца исходной таблицы Вам необходимо явно прописать столбцы (колонки), в которые нужно сделать запись.

Дополнительная информация

Также сразу лучше всего проверить возможность использовать оператор LOAD DATA. Дляэтого необходимо выполнить следующий запрос:

SHOWVARIABLESLIKE 'local_infile';

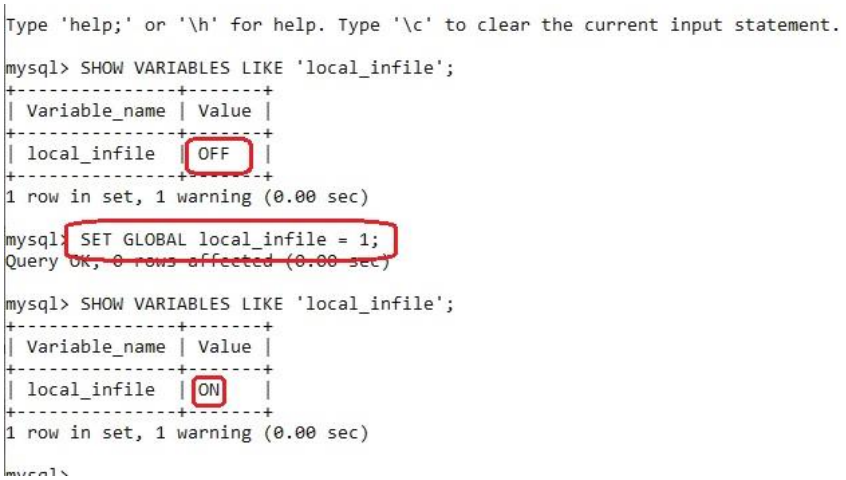
```
mysql> SHOW VARIABLES LIKE 'local_infile';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| local_infile  | OFF   |
+---------------+-------+
1 row in set, 1 warning (0.00 sec)

mysql>
```

Еслив колонке Value значение OFF, то при попытке выполнить оператор LOAD DATA... MySQLвыдастошибку:

*ERROR 1148: The used command **isnot** allowed **withthis** MySQL version*

Чтобыисправить данное недоразумение достаточно выполнить следующий запрос:

SETGLOBAL local_infile = 1;



Минусданного решения в том, что необходимы права администратора чтобы изменить данную глобальную переменную.

Экспорт данных базы в документы пользователя

Для экспорта данных мы должны считать данные из таблицы базы данных (select * from«имя таблицы») и перенаправить их в необходимый файл (так же используя полный путь к имени файла) используя команду **intooutfile**, которая схода по синтаксису с **LOAD DATA INFILE.**

select * from XXX
into outfile 'путькфайлу.csv'
fields terminated by ';'
enclosed by ""
lines terminated by '\n';

Пример:
select * from comics into outfile 'c:/ospanel/userdata/temp/upload/rr.csv'
fields terminated by ';'
enclosed by ""
lines terminated by '\n';

Выполнить выгрузку данных из таблиц базы данных в csv файл

SELECT * FROM таблица_базы_данных
INTO OUTFILE 'путь_и_имя_файла.csv'
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n';

И в XLS файл

SELECT * FROM таблица_базы_данных INTO OUTFILE 'путь_и_имя_файла.xls';

ENCLOSEDBY – указывает, что двойная кавычка " окружает значения. FIELDSTERMINATEDBY',' – Используйте**эту**строку, чтобы указать что строка заканчивается **,** . IGNORE 1 ROWS; - Многие CSV-файлы экспортируются с метками столбцов в первой строке.**Эта**команда говорит**MySQL**игнорировать первую строку, так как вы уже создали свою таблицу с соответствующими заголовками столбцов. Конструкция TERMINATED **BY '\n'** говорит о том, что каждая строка заканчивается значением указанным в**TERMINATED BY**.

CSV

```
mysql> select * from klient into outfile '/var/lib/mysql-files/import4.csv'
 fields terminated by ','  lines terminated by '\n';
Query OK, 6 rows affected (0,00 sec)
```

Провести выгрузку в файл с расширением txt.

XLS

```
mysql> select * from klient into outfile '/var/lib/mysql-files/import2.xls';
Query OK, 6 rows affected (0,00 sec)
```

Провести выгрузку в файл с расширением doc.

Практическое задание

1 Произвести импорт данных из .xlsxфайла в таблицу MtSQL (таблицу создайте самостоятельно, она должна содержать не мене 5 столбцов с типами данных даты, числового целого, числового дробного, символьного, содержимое таблицы должно быть на русском языке).

2 Произвести импорт данных из.txtфайла в таблицу MtSQL (таблицу создайте самостоятельно, она должна содержать не мене 5 столбцов с типами данных даты, числового целого, числового дробного, символьного, содержимое таблицы должно быть на русском языке).

3 Произвести экспорт данных таблицы «Заказы» в .xlsx файл

4 Произвести экспорт данных таблицы «Заказчики юр.лица» в .txt файл

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

Тема 2.3
Разрабатывать объекты баз данных NoSQL

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Практическое занятие № 17.
Создание базы данных NoSQL и выполнение основных запросов.

Проверяемые компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. 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 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в

чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;

ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных

ПК 2.4 Формировать требования хранилищ банка данных для обучения

ПК 2.5 Подготавливать данные для базы знаний

ПК 2.7 Осуществлять автоматизацию баз данных

ПК 2.8 Разрабатывать объекты баз данных NoSQL

Создание коллекции

Чтобы создать новую таблицу, можно воспользоваться методов create_collection базы данных:

```
import pymongo
from pymongo.server_api import ServerApi

client = pymongo.MongoClient("mongodb+srv://megabax:<Пароль>@cluster0.vwremdq.mongodb.net/?retryWrites=true&w=majority", server_api=ServerApi('1'))
db = client.test

try:
   print("MongoDB version is %s" % client.server_info()['version'])
except pymongo.errors.OperationFailure as error:
   print(error)
   quit(1)

db.create_collection("MyTable")
```

После выполнения программы в списке таблиц базы данных появится наша таблица.

Заполнение коллекции данными

В общих чертах, как работать с базой данных mongoDB. Как видим из примеров кода, сначала мы создаем подключение (клиента). Через него получаем объект базы данных (это поле объекта клиента, имя которого совпадает с именем базы данных). Сам объект базы данных содержит в себе поля, соответствующие таблицам. Эти таблицы представляют собой коллекцию элементов таблицы. Эти коллекции содержат в себе методы по манипулированию данными, такие как перебор элементов, поиск, добавление и удаление. Давайте, например, создадим тестовую базу данных, путем добавления в цикле множества элементов методом insert_one:

```
import pymongo
from pymongo.server_api import ServerApi
```

```
import random
import string

def generate_random_string(length):
    letters = string.ascii_lowercase
    rand_string = ''.join(random.choice(letters) for i in range(length))
    print("Random string of length", length, "is:", rand_string)
    return rand_string

def get_random_features():
    dict={}
    ln=random.randint(1,10)
    for i in range(ln):
        dict["feature"+str(i)]=generate_random_string(random.randint(3,15))
    return dict

client = pymongo.MongoClient("mongodb+srv://megabax:<Пароль>@cluster0.vwremdq.mongodb.net/?retryWrites=true&w=majority", server_api=ServerApi('1'))
db = client.test

try:
    print("MongoDB version is %s" % client.server_info()['version'])
except pymongo.errors.OperationFailure as error:
    print(error)
    quit(1)


mongo_collection = db.goods


for i in range(100):
    mongo_collection.insert_one({
        "_id": i,
        "name": generate_random_string(random.randint(3,15)),
        "price": round(random.random()*100,2),
        "features": get_random_features(),
        "quantity": random.randint(1,10000)
    })
```

Перебор и фильтрация документов

Теперь попробуем перебрать элементы таблицы:

```
import pymongo
from pymongo.server_api import ServerApi
```

```
client = pymongo.MongoClient("mongodb+srv://megabax:<Пароль>@cluster0.vwremdq.mongodb.net/?retryWrites=true&w=majority", server_api=ServerApi('1'))
db = client.test

try:
    print("MongoDB version is %s" % client.server_info()['version'])
except pymongo.errors.OperationFailure as error:
    print(error)
    quit(1)

my_collection = db.goods
my_cursor = my_collection.find()

for item in my_cursor:
    print(item["name"])
```

Этот код переберет все элементы, и выведет значения их полей name. Можно перебрать по фильтру:

```
my_cursor = my_collection.find({
    "name": "jgdpspvonpune"
})
```

Вложенные коллекции в документах

Стоит заметить, что в качестве полей у элементов коллекции может быть и коллекция, и какой-либо другой объект. Например, у элемента коллекции goods есть элемент features, который сам по себе является объектом, а именно словарем:

А сейчас мы сделаем следующее. Очистим коллекцию goods, а потом снова перезаполним ее, но немного по-другому. Для этого можно вызвать метод delete_many(), а в качестве параметров указать пустой фильтр:

```
import pymongo
from pymongo.server_api import ServerApi

client = pymongo.MongoClient("mongodb+srv://megabax:<Пароль>@cluster0.vwremdq.mongodb.net/?retryWrites=true&w=majority", server_api=ServerApi('1'))
db = client.test

try:
    print("MongoDB version is %s" % client.server_info()['version'])
except pymongo.errors.OperationFailure as error:
    print(error)
```

```
        quit(1)

    my_collection = db.goods
    my_collection.delete_many({})
```

Результатом работы скрипта будет полная очистка коллекции.

Теперь поменяем в нашем скрипте формирования демо базы всего лишь одну функцию:

```
    def get_random_features():
        list=[]
        ln=random.randint(1,10)
        for i in range(ln):

list.append(generate_random_string(random.randint(3,15)))
        return list
```

Теперь вместо объекта в поле features у нас коллекция, а не объект:

И теперь мы можем осуществить поиск по элементу коллекции:

```
    my_cursor = my_collection.find({
        "features": {"$eq": "hpwfslrwfsuqmue"}
    })
```

Запрос найдет все документы, в которых в коллекции features встретится строка «hpwfslrwfsuqmue».

Операторы сравнения в фильтрах

Выражение $eq в фильтре означает «равно». Можно сравнивать на больше меньше, например, «меньше или равно» будет кодироваться как $lt:

```
    import pymongo
    from pymongo.server_api import ServerApi

    client                                                    =
pymongo.MongoClient("mongodb+srv://megabax:<Пароль>@cluster
0.vwremdq.mongodb.net/?retryWrites=true&w=majority",
server_api=ServerApi('1'))
    db = client.test

    try:
        print("MongoDB      version      is      %s"      %
client.server_info()['version'])
    except pymongo.errors.OperationFailure as error:
        print(error)
```

```
    quit(1)

my_collection = db.goods

my_cursor = my_collection.find({
    "price": {"$lt": 10}
})

for item in my_cursor:
    print(item["name"], item["price"])
```

Эта программа выдаст список товаров, цена которых не превышает 10.

Критерии оценивания:

Оценка «5» (отлично) — работа выполнена полностью и правильно, все задания решены, ошибки отсутствуют.

Оценка «4» (хорошо) — работа выполнена полностью, допущены незначительные ошибки, не влияющие на общий результат.

Оценка «3» (удовлетворительно) — работа выполнена частично, допущены ошибки, но основные задачи решены.

Оценка «2» (неудовлетворительно) — работа не выполнена или содержит грубые ошибки, основные задачи не решены.

4. Контрольно-оценочные материалы для промежуточной аттестации по учебной дисциплине – ОП.07 «Основы проектирования баз данных».

Теоретические вопросы на экзамен:

1. Понятие базы данных и системы управления базами данных (СУБД). Назначение, основные функции и компоненты СУБД.
2. Классификация баз данных. Основные особенности, преимущества и недостатки реляционных и нереляционных баз данных.
3. Основные понятия реляционной модели данных: отношение, атрибут, домен, кортеж, ключ и связь.
4. Модели данных и структуры информационных систем. Иерархическая, сетевая и реляционная модели данных.
5. Однопользовательские и многопользовательские системы баз данных. Особенности организации доступа к данным.
6. Основные этапы проектирования баз данных. Концептуальное, логическое и физическое проектирование.
7. ER-модель данных. Основные элементы ER-диаграмм: сущности, атрибуты, связи и их характеристики.
8. Типы связей между сущностями в реляционных базах данных. Реализация связей «один к одному», «один ко многим» и «многие ко многим».
9. Первичные, внешние и составные ключи в реляционных базах данных. Назначение и особенности их использования.
10. Нормализация реляционных баз данных. Первая, вторая и третья нормальные формы. Цели и принципы нормализации.
11. Переход от концептуальной модели базы данных к логической. Построение реляционной схемы базы данных.
12. CASE-средства проектирования баз данных. Назначение, возможности и преимущества использования CASE-технологий.
13. Целостность данных в реляционных базах данных. Основные виды ограничений целостности и способы их реализации.
14. Физическое проектирование базы данных. Создание базы данных и таблиц в СУБД.
15. Типы данных в СУБД. Правила выбора типов данных при проектировании структуры таблиц.
16. Индексы в базах данных. Назначение, виды и влияние индексов на производительность запросов.
17. Основные операции управления данными в СУБД: добавление, изменение и удаление записей. Оптимизация структуры таблиц.
18. Язык SQL: назначение, основные группы команд и области их применения.
19. Создание и изменение структуры таблиц средствами SQL. Операторы CREATE TABLE, ALTER TABLE и DROP TABLE.
20. Операторы SQL для работы с данными: INSERT, UPDATE и DELETE. Назначение и синтаксис.
21. Выборка данных с помощью оператора SELECT. Использование

условий отбора WHERE, логических операторов и сортировки ORDER BY.

22. Агрегатные функции SQL: COUNT, SUM, AVG, MIN и MAX. Примеры использования.

23. Группировка данных в SQL. Операторы GROUP BY и HAVING, их назначение и особенности применения.

24. Соединение таблиц в SQL. Виды JOIN (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN) и особенности их использования.

25. Подзапросы в SQL. Вложенные и коррелированные подзапросы, их назначение и примеры применения.

26. Использование агрегатных функций в подзапросах. Операторы ANY и ALL в SQL.

27. Оптимизация SQL-запросов. Основные методы повышения производительности запросов к базе данных.

28. Планы выполнения SQL-запросов. Анализ планов выполнения и выявление причин низкой производительности.

29. Настройка производительности СУБД. Основные способы импорта и экспорта данных.

30. Нереляционные базы данных NoSQL. Основные типы, особенности организации хранения данных, создание объектов и выполнение основных запросов.

Практические задания на экзамен:

1. Проектирование структуры базы данных. Разработайте структуру базы данных «Библиотека». Определите основные сущности, их атрибуты, первичные и внешние ключи. Установите связи между таблицами.

2. Построение ER-диаграммы. Постройте ER-диаграмму базы данных «Интернет-магазин», включающую сущности «Клиенты», «Товары», «Заказы» и «Состав заказа». Укажите типы связей между сущностями.

3. Нормализация базы данных. Дана таблица «Заказы», содержащая поля: НомерЗаказа, ДатаЗаказа, ФИОКлиента, ТелефонКлиента, КодТовара, НазваниеТовара, Цена, Количество. Выполните нормализацию таблицы до третьей нормальной формы (3НФ).

4. Создание базы данных средствами SQL. Создайте базу данных «Учебное заведение». Разработайте и создайте таблицы «Студенты», «Группы» и «Специальности». Определите первичные и внешние ключи.

5. Создание таблицы с ограничениями целостности. Создайте таблицу «Сотрудники» с полями: КодСотрудника, ФИО, ДатаРождения, Должность, Зарплата. Используйте ограничения PRIMARY KEY, NOT NULL, UNIQUE и CHECK.

6. Изменение структуры таблицы. Создайте таблицу «Товары» с полями КодТовара, Название, Цена. С помощью SQL добавьте поле «КоличествоНаСкладе», измените тип данных одного из полей и удалите ненужный столбец.

7. Добавление данных в таблицу. Создайте таблицу «Студенты» с полями

КодСтудента, ФИО, Группа, СреднийБалл. Добавьте в таблицу не менее 10 записей с помощью оператора INSERT INTO.

8. Изменение и удаление записей. Создайте таблицу «Товары», содержащую не менее 10 записей. Напишите SQL-запросы для увеличения цены товаров определённой категории на 10% и удаления товаров с нулевым остатком.

9. Выборка данных с условием. Создайте таблицу «Сотрудники» с полями КодСотрудника, ФИО, Отдел, Должность, Зарплата. Напишите SQL-запрос для вывода сотрудников определённого отдела, зарплата которых превышает 50 000 рублей.

10. Сортировка и фильтрация данных. Создайте таблицу «Студенты» с полями КодСтудента, ФИО, Группа, СреднийБалл. Выведите студентов со средним баллом выше 4,0, отсортировав результат по убыванию среднего балла.

11. Использование агрегатных функций. Создайте таблицу «Продажи» с полями КодПродажи, ДатаПродажи, КодТовара, Количество, Сумма. Напишите SQL-запросы для определения общего количества продаж, общей суммы выручки, средней суммы продажи, минимальной и максимальной суммы продажи.

12. Группировка данных. Создайте таблицу «Сотрудники» с полями КодСотрудника, ФИО, Отдел, Зарплата. Напишите SQL-запрос для определения средней зарплаты по каждому отделу. Выведите только отделы, в которых средняя зарплата превышает 60 000 рублей.

13. Соединение таблиц с помощью INNER JOIN. Создайте таблицы «Студенты» и «Группы», связанные внешним ключом. Напишите SQL-запрос, выводящий ФИО студента, название группы и специальность.

14. Использование LEFT JOIN. Создайте таблицы «Клиенты» и «Заказы». Напишите SQL-запрос, который выводит список всех клиентов и количество оформленных ими заказов, включая клиентов, не совершивших ни одного заказа.

15. Использование вложенных запросов. Создайте таблицу «Сотрудники» с полями КодСотрудника, ФИО, Отдел, Зарплата. Напишите SQL-запрос для поиска сотрудников, зарплата которых превышает среднюю зарплату по организации.

16. Использование коррелированных подзапросов. Создайте таблицу «Сотрудники» с полями КодСотрудника, ФИО, Отдел, Зарплата. Напишите SQL-запрос для вывода сотрудников, зарплата которых превышает среднюю зарплату по их отделу.

17. Использование операторов ANY и ALL. Создайте таблицу «Товары» с полями КодТовара, Название, Категория, Цена. Напишите SQL-запросы для поиска товаров, цена которых выше цены хотя бы одного товара выбранной категории, и товаров, цена которых выше цены всех товаров выбранной категории.

18. Создание индексов и оптимизация запросов. Создайте таблицу «Заказы», содержащую не менее 100 записей. Напишите запрос для поиска заказов по дате и идентификатору клиента. Проанализируйте план выполнения запроса, создайте соответствующий индекс и сравните планы выполнения до и после создания индекса.

19. Импорт и экспорт данных. Создайте таблицу «Товары» с полями

КодТовара, Название, Категория, Цена, Количество. Выполните импорт данных из CSV-файла в таблицу базы данных. Экспортируйте содержимое таблицы в CSV-файл и проверьте корректность данных.

20. Создание и использование базы данных NoSQL. Создайте базу данных «Каталог товаров» в MongoDB. Создайте коллекцию «Товары», добавьте не менее 10 документов с полями Название, Категория, Цена, Количество. Выполните запросы для поиска товаров определённой категории, сортировки по цене, изменения цены выбранного товара и удаления документа по заданному условию.

Пример экзаменационного билета

ОДОБРЕНО Цикловой комиссией Программирования Протокол №____ от _____ _____ С.Н. Сулавко	**Экзаменационный билет** **по дисциплине** **«Основы проектирования баз данных»** **для специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта»** **4 семестр** **Вариант №1**	**УТВЕРЖДАЮ** Заместитель директора по УМР И.В. Подцатова _____ «_____» _____ 2026г.

Теоретическая часть

1. Графические методы численного решения алгебраических и трансцендентных уравнений
2. Понятие интервала изоляции корня уравнения

Практическая часть
Исходные данные:

Перевозка(8 записей)
Дата отправления
Адрес «откуда»
Адрес «куда»
Дата прибытия
Дата формирования заявки
Название груза
Общая стоимость поездки
Категория перевозки («Город», «Город выходные», «Область», «Россия»)
Стоимость поездки в зависимости от категории
Единица измерения перевозки (км., часы – зависит от категории перевозки)

Перевозчик(1 запись)
Название компании перевозчика
Адрес компании перевозчика
Телефон компании перевозчика
Платёжные реквизиты компании перевозчика

Водитель(3 записи)
ФИО водителя
Телефон

Заказчик(5 записей)
Название заказчика
ФИО контактного лица заказчика
Номер телефона заказчика

Автомобиль(3 записи)
Марка автомобиля
Модель автомобиля
ГОС номер автомобиля

Задание:

1. Создать базу данных
2. Провести нормализацию таблиц
3. Создать таблицы
4. Определить «ключи» в таблицах исходя из логической модели базы данных
5. Реализовать связи между таблицами
6. Заполнить таблицы
7. Реализовать запросы к базе данных и проверить правильность их выполнения
   7.1 Показать ФИО заказчика, который осуществлял поезду из категории «Область» (название категории так же вывести)
   7.2 Показать сколько поездок осуществил каждый водитель (ФИО водителя так же вывести)
   7.3 Показать заказчика, который заказывал такси больше всех (ФИО заказчика так же показать)
   7.4 Показать водителя, который принёс большую прибыль компании (ФИО водителя так же вывести)
   7.5 Показать заказчика «А», которого перевозил автомобиль «Б» (ФИО заказчика и ГОС номер автомобиля так же вывести)

«____»_____ 2026г. Преподаватель _____

Лист согласования

Дополнения и изменения к комплекту ФОС на учебный год

Дополнения и изменения к комплекту ФОС на _____ учебный год по дисциплине _____ комплект ФОС внесены следующие изменения:

_____

_____

_____

_____

_____

Дополнения и изменения в комплекте ФОС обсуждены на заседании ЦК _____ Протокол № _____ от «_____» _____ 20_____г.

Председатель ЦК