МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Смоленский казачий институт промышленных технологий и бизнеса (филиал) федерального государственного бюджетного образовательного учреждения «Московский государственный университет технологий и управления имени К.Г. Разумовского (Первый казачий университет)»

Смоленский казачий институт промышленных технологий и бизнеса (филиал)

УТВЕРЖДАЮ
Директор СКИПТБ (филиал) ФГБОУ ВО
«МГУТУ им. К.Г. Разумовского (ПКУ)
_____ Н.С. Коржикова
«21» апреля 2026 г.

Рабочая программа дисциплины (модуля)

Б1.О.04.06 ОБЩЕПРОФЕССИОНАЛЬНЫЙ МОДУЛЬ Базы данных

Кафедра:	Системной автоматизации и информационных технологий
Направление подготовки:	09.03.01 Информатика и вычислительная техника
Направленность (профиль):	Проектирование и разработка программного обеспечения
Квалификация выпускника:	Бакалавр
Форма обучения:	очная
Год набора:	2026
Общая трудоемкость:	288 часов/8 з.е.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 00F5 665F CCDB 5046 05DD 6E08 EBCB D00A DC
Действителен: с 21 октября 2025 г. по 14 января 2027 г.
Владелец: Коржикова Надежда Сергеевна
Должность: Директор

Вязьма, 2026 г.

Программу составил(и):
Старший преподаватель Долотовский Алексей Игоревич

Рабочая программа дисциплины (модуля)
"Базы данных"
разработана на основании учебного плана, утвержденного ученым советом 26 марта 2026 г. протокол № 9 в соответствии с ФГОС ВО Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929),

Руководитель ОПОП
канд. пед. наук, доцент Кузьмин К.А._____

Рабочая программа обсуждена на заседании обеспечивающей кафедры
Системной автоматизации и информационных технологий
Протокол от 10 апреля 2026 г. № 9
Зав. кафедрой Кузьмин Константин Анатольевич_____

СОДЕРЖАНИЕ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Цели:

Цель освоения дисциплины «Базы данных» (далее – «дисциплина») состоит в формировании у студентов компетенций в соответствии с ФГОС ВО, учебным планом и основной профессиональной образовательной программой (далее ОПОП).

1.2. Задачи:

Задачами освоения дисциплины являются:
1) формирование теоретических знаний по предмету дисциплины (в т.ч. освоение основных концепций и понятий, необходимой терминологии);
2) приобретение практических умений и навыков в соответствии с формируемыми компетенциями в рамках предмета дисциплины, необходимые в т.ч. для последующего самообразования;
3) ознакомление с типичными программными средствами, используемыми в рамках предметной области дисциплины;
4) формирование навыков самостоятельной работы, в том числе поиска и анализа информации в рамках предметной области дисциплины.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

Цикл (раздел) ОП: "Базы данных"

Дисциплина относится к обязательной части ОПОП и обязательна для освоения.

Связь с предшествующими дисциплинами (модулями), практиками

№ п/п	Наименование	Семестр	Шифр компетенции
1	Инженерная графика	3	ОПК-1.1, ОПК-1.2, ОПК-1.3
2	Ознакомительная практика	2	УК-7.1, УК-7.2, УК-7.3, ОПК-8.1, ОПК-8.2, ОПК-8.3, УК-4.1, УК-4.2, УК-4.3
3	Программирование	3	ОПК-2.1, ОПК-2.2, ОПК-2.3
4	Сети и телекоммуникации	4	ОПК-7.1, ОПК-7.2, ОПК-7.3
5	Специализированные пакеты профессиональной деятельности	0	
6	Основы информационных технологий	1	УК-1.1, УК-1.2, УК-1.3, ОПК-2.1, ОПК-2.2, ОПК-2.3
7	Архитектура вычислительных систем	3	ОПК-2.1, ОПК-2.2, ОПК-2.3
8	Пакеты прикладных программ для профессиональной деятельности	2	ОПК-2.1, ОПК-2.2, ОПК-2.3
9	Модуль Информационные технологии и программирование	0	
10	Модуль Введение в информационные технологии	0	

Связь с последующими дисциплинами (модулями), практиками

№ п/п	Наименование	Семестр	Шифр компетенции
1	Технологическая (проектно-технологическая) практика	4	УК-1.1, УК-1.2, УК-1.3, УК-2.1, УК-2.2, УК-2.3, УК-3.1, УК-3.2, УК-3.3, УК-3.4, УК-3.5, УК-3.6, УК-5.1, УК-5.2, УК-5.3, УК-8.1, УК-8.2, УК-8.3, УК-8.4, УК-8.5, УК-8.6, УК-9.1, УК-9.2, УК-9.3, УК-10.1, УК-10.2, УК-10.3, УК-10.4, УК-10.5, УК-10.6, ОПК-1.1, ОПК-1.2, ОПК-1.3, ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-3.1, ОПК-3.2, ОПК-3.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-5.1, ОПК-5.2, ОПК-5.3, ОПК-6.1, ОПК-6.2, ОПК-6.3, ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-9.1, ОПК-9.2, ОПК-9.3, УК-4.4, УК-4.5, УК-4.6, УК-6.1, УК-6.2, УК-6.3
2	Интеллектуальные информационные системы	5	ОПК-2.1, ОПК-2.2, ОПК-2.3
3	Управление информационными системами	6	ОПК-8.1, ОПК-8.2, ОПК-8.3, ОПК-4.1, ОПК-4.2, ОПК-4.3
4	Основы технологии хлеба, кондитерских и макаронных изделий	0	

Распределение часов дисциплины

Семестр (<Курс>.<Семестр на курсе>)	3 (2.1)		4 (2.2)		Итого	
Недель	17 2/6		17 1/6			
Вид занятий	УП	РП	УП	РП	УП	РП
Лекции	16	16	16	16	32	32
Лабораторные	48	48	48	48	96	96
Итого ауд.	64	64	64	64	128	128
Контактная работа	64	64	64	64	128	128
Сам. работа	80	80	53	53	133	133
Часы на контроль			27	27	27	27
Итого	144	144	144	144	288	288

Вид промежуточной аттестации:

ЗаО 3 семестр

Экзамен 4 семестр

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины (модуля) направлен на формирование следующих компетенций и индикаторов их

ОПК-5:Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

ОПК-5.1: Знает основы системного администрирования, администрирования систем управления базами данных, современные методы информационного взаимодействия информационных и автоматизированных систем

ОПК-5.2: Умеет выполнять подключение, установку и проверку аппаратных, программно-аппаратных и программных средств

ОПК-5.3: Владеет методами установки системного и прикладного программного обеспечения

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименования разделов, тем, их краткое содержание и результаты освоения /вид занятия/	Семестр	Часы	Интеракт.	Прак. подг.	Индикаторы достижения компетенции	Виды контроля
	Раздел 1.Базы данных						
1.1	**Тема 1 Вводные понятия Краткое содержание: понятие, состав и назначение компонентов ИС; понятие БД и СУБД; архитектуры современных ИС; модели данных: понятие, преимущества и недостатки. Предполагаемые результаты обучения: Знать: теоретические основы баз данных, структуру баз данных, системы управления базами данных для информационных систем различного назначения; архитектуру баз данных, физические и логические уровни представления данных; основы моделей данных /Лек/**	3	8	0	0	ОПК-5.1	Вопросы для устного опроса
1.2	**Тема 1 Вводные понятия Краткое содержание: понятие, состав и назначение компонентов ИС; понятие БД и СУБД; архитектуры современных ИС; модели данных: понятие, преимущества и недостатки. Предполагаемые результаты обучения: Уметь: создавать запросы к базам данных для выполнения различных операций обработки информации; выделять и**	3	24	0	0	ОПК-5.2,ОПК-5.3	Отчет по лабораторным работам

	классифицировать объекты предметной области, строить диаграммы потоков данных Владеть: навыками работы с базами данных; навыками описания функциональных требований с использованием графических нотаций и CASE-инструментов /Лаб/						
1.3	**Тема 1 Вводные понятия Краткое содержание: понятие, состав и назначение компонентов ИС; понятие БД и СУБД; архитектуры современных ИС; модели данных: понятие, преимущества и недостатки. Предполагаемые результаты обучения: Знать: теоретические основы баз данных, структуру баз данных, системы управления базами данных для информационных систем различного назначения; архитектуру баз данных, физические и логические уровни представления данных; основы моделей данных Уметь: создавать запросы к базам данных для выполнения различных операций обработки информации; выделять и классифицировать объекты предметной области, строить диаграммы потоков данных Владеть: навыками работы с базами данных; навыками описания функциональных требований с использованием графических нотаций и CASE-инструментов /Ср/**	3	40	0	0	ОПК-5.1,ОПК-5.2,ОПК-5.3	вопросы для самоподготовки
1.4	**Тема 2 Проектирование реляционных баз данных Краткое содержание: Концептуальное (инфологическое) проектирование. Модель «сущность -связь», ER-диаграмма. Логическое (даталогическое) проектирование реляционных БД. Нормализация и денормализация. Хронология изменений в БД. Моделирование иерархических данных в реляционных СУБД. Интернационализация и локализация данных. Объекты в реляционных СУБД. Проектирование физического хранения данных. Индексация и секционирование Предполагаемые результаты обучения: Знать: основы проектирования баз данных ; способы описания предметной области; назначение и виды СУБД, основные модели данных; основные способы хранения данных в ИС; способы повышения производительности**	3	8	0	0	ОПК-5.1	Вопросы для устного опроса

	систем БД и восстановления БД; интерфейсы прикладного программирования БД, языковые средства СУБД /Лек/						
1.5	**Тема 2 Проектирование реляционных баз данных** **Краткое содержание:** **Концептуальное (инфологическое) проектирование. Модель «сущность -связь», ER-диаграмма. Логическое (даталогическое) проектирование реляционных БД. Нормализация и денормализация. Хронология изменений в БД. Моделирование иерархических данных в реляционных СУБД. Интернационализация и локализация данных. Объекты в реляционных СУБД. Проектирование физического хранения данных. Индексация и секционирование** **Предполагаемые результаты обучения:** **Уметь: проектировать базы данных информационных систем различного назначения; выполнять инфологическое и даталогическое проектирование БД, осуществлять проверку полученных моделей и их оптимизацию, составлять запросы; переносить данные и метаданные в другую СУБД** **Владеть: навыками разработки приложений на основе систем управления базами данных; навыками проектирования моделей БД с использованием графических нотаций и CASE-инструментов; навыками и средствами экспорта/импорта данных; навыками и средствами администрирования БД; навыками подключения приложений к БД, используя универсальные среды разработки /Лаб/**	3	24	0	0	ОПК-5.2,ОПК-5.3	Отчет по лабораторным работам
1.6	**Тема 2 Проектирование реляционных баз данных** **Краткое содержание:** **Концептуальное (инфологическое) проектирование. Модель «сущность -связь», ER-диаграмма. Логическое (даталогическое) проектирование реляционных БД. Нормализация и денормализация. Хронология изменений в БД. Моделирование иерархических данных в реляционных СУБД. Интернационализация и локализация данных. Объекты в реляционных СУБД. Проектирование физического хранения данных. Индексация и секционирование** **Предполагаемые результаты обучения:** **Знать: основы проектирования баз данных ; способы описания**	3	40	0	0	ОПК-5.1,ОПК-5.2,ОПК-5.3	вопросы для самоподготовки

	предметной области; назначение и виды СУБД, основные модели данных; основные способы хранения данных в ИС; способы повышения производительности систем БД и восстановления БД; интерфейсы прикладного программирования БД, языковые средства СУБД **Уметь: проектировать базы данных информационных систем различного назначения; выполнять инфологическое и даталогическое проектирование БД, осуществлять проверку полученных моделей и их оптимизацию, составлять запросы; переносить данные и метаданные в другую СУБД** **Владеть: навыками разработки приложений на основе систем управления базами данных; навыками проектирования моделей БД с использованием графических нотаций и CASE-инструментов; навыками и средствами экспорта/импорта данных; навыками и средствами администрирования БД; навыками подключения приложений к БД, используя универсальные среды разработки /Ср/**						
1.7	**Зачет с оценкой** **Знать: теоретические основы баз данных, структуру баз данных, системы управления базами данных для информационных систем различного назначения; архитектуру баз данных, физические и логические уровни представления данных; основы моделей данных; основы проектирования баз данных ; способы описания предметной области; назначение и виды СУБД, основные модели данных; основные способы хранения данных в ИС; способы повышения производительности систем БД и восстановления БД; интерфейсы прикладного программирования БД, языковые средства СУБД** **Уметь: проектировать базы данных информационных систем различного назначения; создавать запросы к базам данных для выполнения различных операций обработки информации; выделять и классифицировать объекты предметной области, строить диаграммы потоков данных; выполнять инфологическое и даталогическое проектирование БД, осуществлять проверку полученных моделей и их оптимизацию, составлять запросы; переносить данные и метаданные в другую СУБД** **Владеть: навыками разработки**	3	0	0	0	ОПК-5.1,ОПК-5.2,ОПК-5.3	Вопросы к зачету с оценкой, итоговое тестирование

	приложений на основе систем управления базами данных; навыками работы с базами данных; навыками описания функциональных требований с использованием графических нотаций и CASE-инструментов; навыками проектирования моделей БД с использованием графических нотаций и CASE-инструментов; навыками и средствами экспорта/импорта данных; навыками и средствами администрирования БД; навыками подключения приложений к БД, используя универсальные среды разработки /ЗаО/						
1.8	**Тема 3 Реализация и администрирование БД.** **Краткое содержание: Язык запросов SQL: понятие, история, стандарты, диалекты. SQL: создание БД и таблиц. SQL: поиск, выборка и модификация данных. Триггер. Администрирование БД средствами реляционной СУБД. Пользователи и роли. Механизм транзакций. Понятие NoSQL. Понятие big data. Предполагаемые результаты обучения:** **Знать: основы языка запросов SQL /Лек/**	4	16	0	0	ОПК-5.1	Вопросы для устного опроса
1.9	**Тема 3 Реализация и администрирование БД.** **Краткое содержание: Язык запросов SQL: понятие, история, стандарты, диалекты. SQL: создание БД и таблиц. SQL: поиск, выборка и модификация данных. Триггер. Администрирование БД средствами реляционной СУБД. Пользователи и роли. Механизм транзакций. Понятие NoSQL. Понятие big data.** **Уметь: создавать таблицы и базы данных, осуществлять поиск, выборку и модификацию данных с помощью языка SQL** **Владеть: понятиями NoSQL и BigData /Лаб/**	4	48	0	0	ОПК-5.2,ОПК-5.3	Отчет по лабораторным работам
1.10	**Тема 3 Реализация и администрирование БД.** **Краткое содержание: Язык запросов SQL: понятие, история, стандарты, диалекты. SQL: создание БД и таблиц. SQL: поиск, выборка и модификация данных. Триггер. Администрирование БД средствами реляционной СУБД. Пользователи и роли. Механизм транзакций. Понятие NoSQL. Понятие big data.** **Знать: основы языка запросов SQL** **Уметь: создавать таблицы и базы данных, осуществлять поиск, выборку и модификацию данных с**	4	53	0	0	ОПК-5.1,ОПК-5.2,ОПК-5.3	вопросы для самоподготовки

	помощью языка SQL Владеть: понятиями NoSQL и BigData /Ср/						
1.11	**Экзамен. Подготовка и проведение экзамена. Знать: основы системного администрирования, администрирования систем управления базами данных, современные методы информационного взаимодействия информационных и автоматизированных систем Уметь: выполнять подключение, установку и проверку аппаратных, программно-аппаратных и программных средств Владеть: методами установки системного и прикладного программного обеспечения /Экзамен/**	4	27	0	0	ОПК-5.1,ОПК-5.2,ОПК-5.3	итоговое тестирование, вопросы к экзамену

Перечень применяемых активных и интерактивных образовательных технологий:

Информационные технологии

Личностно ориентированная технология, способ организации самостоятельной деятельности учащихся, направленный на решение задачи учебного проекта

Компьютерная технология обучения

Основана на использовании информационных технологий в учебном процессе. Реализация данной технологии осуществляется посредством компьютера и иных мультимедийных средств. Использование компьютерных технологий делает учебный процесс не только современным и познавательным, но интересным для обучающихся

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная (внеаудиторная) работа обучающегося (далее - СРС) – это планируемая учебная, практическая, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время (свободное от аудиторных учебных занятий) по заданию и при методическом руководстве научно-педагогического работника (далее – преподаватель) и (или) лиц, привлекаемых к реализации основных профессиональных образовательных программ на иных условиях, но без их непосредственного участия.

СРС по заданию и при методическом руководстве преподавателя и (или) лиц, привлекаемых к реализации основных профессиональных образовательных программ на иных условиях, реализуется во время групповых консультаций и (или) индивидуальной работы обучающихся с преподавателями Университета и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях (в том числе индивидуальных консультаций), а также во время текущего контроля выполнения заданий, отнесенных к самостоятельной работе обучающихся.

Целью СРС является овладение формирование компетенций через овладение знаниями, умениями и навыками профессиональной деятельности по направлению подготовки (специальности). Самостоятельная работа является обязательной для каждого обучающегося.

Формы самостоятельной работы обучающихся определяются преподавателями кафедр Университета при разработке рабочих программ дисциплин (модулей), рабочих программ практик, программ государственной итоговой (итоговой) аттестации, методических указаний по выполнению практических, лабораторных работ, написанию курсовых работ/проектов и ВКР в соответствии с их содержанием.

В Университете оборудованы специальные помещения для самостоятельной работы обучающихся. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к информационно-телекоммуникационной сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

6.1. Перечень компетенций с указанием уровней их сформированности

ОПК-5:Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

Недостаточный уровень:

Не знает основы системного администрирования, администрирования систем управления базами данных, современные методы информационного взаимодействия информационных и автоматизированных систем

Не умеет выполнять подключение, установку и проверку аппаратных, программно-аппаратных и программных средств

Не владеет методами установки системного и прикладного программного обеспечения

Пороговый уровень:

Знает основы администрирования систем управления базами данных

Умеет выполнять подключение и установку аппаратных и программных средств

Владеет методами установки программного обеспечения

Продвинутый уровень:

Знает основы системного администрирования, администрирования систем управления базами данных

Умеет выполнять подключение, установку и проверку аппаратных и программных средств

Владеет методами установки системного программного обеспечения

Высокий уровень:

Знает основы системного администрирования, администрирования систем управления базами данных, современные методы информационного взаимодействия информационных и автоматизированных систем

Умеет выполнять подключение, установку и проверку аппаратных, программно-аппаратных и программных средств

Владеет методами установки системного и прикладного программного обеспечения

6.2. Описание критериев оценивания успеваемости

НЕДОСТАТОЧНЫЙ УРОВЕНЬ	ПОРОГОВЫЙ УРОВЕНЬ	ПРОДВИНУТЫЙ УРОВЕНЬ	ВЫСОКИЙ УРОВЕНЬ
Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; - непонимание сущности дополнительных вопросов в рамках заданий билета; - отсутствие умения выполнять практические задания, предусмотренные программой дисциплины (модуля); - отсутствие готовности (способности) к дискуссии и низкая степень контактности.	Обучающийся демонстрирует: - знания теоретического материала; - неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; - неуверенные и неточные ответы на дополнительные вопросы; - недостаточное владение литературой, рекомендованной программой дисциплины (модуля); - умение без грубых ошибок решать практические задания, которые следует выполнить.	Обучающийся демонстрирует: - знание и понимание основных вопросов контролируемого объема программного материала; - твердые знания теоретического материала; - способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития; - правильные и конкретные, без грубых ошибок ответы на поставленные вопросы; - умение решать практические задания, которые следует выполнить; - владение основной литературой, рекомендованной программой дисциплины (модуля); - наличие собственной обоснованной позиции по обсуждаемым вопросам; - незначительные оговорки и неточности в раскрытии отдельных положений вопросов билета, присутствует неуверенность в ответах на дополнительные вопросы.	Обучающийся демонстрирует: - глубокие, всесторонние и аргументированные знания программного материала; - полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий в рамках обсуждаемых заданий; - способность устанавливать и объяснять связь практики и теории; - логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора; - умение решать практические задания; - свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.
Оценка «незачет», «неудовлетворительно»	**Оценка «зачтено/удовлетворительно», «удовлетворительно»**	**Оценка «зачтено/хорошо», «хорошо»**	**Оценка «зачтено/отлично», «отлично»**

6.3. Оценочные средства текущего контроля

Вопросы для устного опроса:

Тема 1 Вводные понятия

1. Что такое база данных?
2. Какие бывают типы баз данных?
3. Что такое реляционная модель данных?
4. Что такое SQL?
5. Какие бывают типы запросов в SQL?
6. Что такое индекс в базе данных?
7. Что такое транзакция?

8. Какие бывают виды ограничений (constraints) в базах данных?
9. Что такое хранилище данных (data warehouse)?
10. Какие бывают виды репликации данных в базах данных?

Тема 2 Проектирование реляционных баз данных
1. Что такое проектирование баз данных?
2. Какие шаги включает в себя процесс проектирования баз данных?
3. Что такое нормализация базы данных?
4. Какие бывают виды связей между таблицами?
5. Какие бывают виды ключей в таблицах?
6. Что такое внешний ключ?
7. Какие бывают типы атрибутов в таблицах?
8. Что такое ER-модель?
9. Что такое денормализация базы данных?
10. Какие бывают типы индексов в базах данных?

Тема 3 Реализация и администрирование БД
1. Что такое реализация базы данных?
2. Как происходит установка СУБД на сервер?
3. Какие действия необходимо выполнить для создания базы данных в СУБД?
4. Что такое индексы в реляционных базах данных?
5. Как выполняются запросы к базе данных?
6. Что такое транзакция в реляционных базах данных?
7. Какие виды резервного копирования базы данных существуют?
8. Что такое оптимизация запросов в реляционных базах данных?
9. Какие задачи решает администратор баз данных?
10. Как осуществляется защита данных в реляционных базах данных?

Вопросы для самоподготовки:

Тема 1 Вводные понятия
1. Что такое база данных и для чего она используется?
2. Какие типы баз данных существуют и чем они отличаются друг от друга?
3. Что такое СУБД и какие функции она выполняет?
4. Какие модели данных используются для описания информации в базах данных?
5. Что такое нормализация и зачем она нужна при проектировании баз данных?
6. Что такое индексы в базах данных и как они помогают ускорить выполнение запросов?
7. Что такое транзакции и зачем они нужны в базах данных?
8. Что такое реляционная алгебра и как она связана с работой с реляционными базами данных?
9. Какие виды связей между таблицами в реляционных базах данных существуют?
10. Что такое SQL и как он используется для работы с базами данных?
11. Какие проблемы могут возникнуть при работе с базами данных и как их можно решить?
12. Какие требования обычно предъявляются к базам данных в организациях?

Тема 2 Проектирование реляционных баз данных
1. Что такое реляционная модель данных?
2. Какие сущности используются в реляционной модели данных?
3. Что такое таблица в реляционной модели данных?
4. Какие свойства имеет отношение в реляционной модели данных?
5. Что такое первичный ключ?
6. Какие типы связей могут быть между таблицами в реляционной модели данных?
7. Что такое нормализация базы данных?
8. Какие формы нормализации существуют?
9. Что такое денормализация базы данных?
10. Какие преимущества и недостатки имеет денормализация базы данных?
11. Что такое ER-модель и как она связана с реляционной моделью данных?
12. Какие инструменты могут быть использованы для проектирования реляционных баз данных?

Тема 3 Реализация и администрирование БД
1. Что такое индексы в реляционных базах данных и для чего они используются?
2. Что такое транзакции и как они используются для обеспечения целостности данных?
3. Что такое ключевые ограничения в БД и как они могут помочь обеспечить целостность данных?
4. Какие методы существуют для резервного копирования и восстановления данных в БД?
5. Что такое SQL-инъекции и как они могут повлиять на безопасность БД?
6. Каковы преимущества и недостатки нормализации баз данных?
7. Какие виды репликации БД существуют и как они могут быть полезны в различных сценариях?
8. Что такое масштабирование БД и как оно может быть реализовано?
9. Что такое профилирование БД и какие инструменты можно использовать для его реализации?

10. Как можно оптимизировать запросы к БД для повышения производительности?
11. Какие методы существуют для обнаружения и устранения проблем производительности в БД?
12. Каковы типичные задачи, с которыми сталкиваются администраторы БД и как они могут быть решены?

Задание к лабораторным работам см. Приложение 1

6.4. Оценочные средства промежуточной аттестации

Вопросы к зачету с оценкой
Индикатор Знать
1) Прочитайте текст и установите правильную последовательность этапов проектирования базы данных:
1. Создание ER-диаграммы
2. Нормализация отношений
3. Определение сущностей и атрибутов
4. Физическое проектирование
2) Прочитайте текст и установите правильную последовательность этапов жизненного цикла БД:
1. Создание резервной копии
2. Проектирование структуры
3. Физическая реализация
4. Тестирование и отладка
3) Прочитайте текст и установите правильную последовательность уровней архитектуры ИС:
1. Физический уровень
2. Логический уровень
3. Внешний уровень
4. Концептуальный уровень
4) Прочитайте текст и установите правильную последовательность этапов нормализации:
1. Первая нормальная форма
2. Третья нормальная форма
3. Вторая нормальная форма
4. Бойцевская нормальная форма
5) Прочитайте текст и установите правильную последовательность операций при работе с данными:
1. Создание индекса
2. Добавление данных
3. Обновление данных
4. Создание таблицы
6) Установите соответствие между типами моделей данных и их характеристиками:
Модель
1. Иерархическая
2. Сетевая
3. Реляционная
4. Постреляционная
Характеристика
А) Данные представлены в виде таблиц
Б) Данные организованы в виде древовидной структуры
В) Данные могут иметь множественные связи
Г) Поддерживает многомерные структуры
Д) Использует только однонаправленные связи
7) Установите соответствие между компонентами ИС и их назначением:
Компонент
1. СУБД
2. БД
3. Приложения
4. Пользователи
Назначение
А) Хранит данные
Б) Обеспечивает доступ к данным
В) Используют данные
Г) Создают и обрабатывают данные
Д) Управляют производительностью
8) Установите соответствие между типами нормализации и их целями:
Нормализация
1. 1НФ
2. 2НФ
3. 3НФ
4. BCNF
Цель
А) Устранение частичных зависимостей
Б) Устранение дублирования данных

В) Устранение транзитивных зависимостей
Г) Устранение избыточных функциональных зависимостей
Д) Устранение всех видов зависимостей
9) Установите соответствие между типами индексов и их особенностями:
Индекс
1. B-дерево
2. Хеш-индекс
3. Кластерный
4. Некластерный
Особенность
А) Сортирует данные физически
Б) Использует хеш-функцию
В) Сортирует данные логически
Г) Оптимизирует поиск по диапазону
Д) Требует меньше места
10) Установите соответствие между этапами проектирования и их содержанием:
Этап
1. Инфологический
2. Даталогический
3. Физический
4. Концептуальный
Содержание
А) Создание таблиц и связей
Б) Создание ER-диаграммы
В) Выбор СУБД и оборудования
Г) Определение требований
Д) Нормализация отношений
Индикатор Уметь
11) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какая модель данных использует древовидную структуру для организации информации?
1. Реляционная
2. Иерархическая
3. Сетевая
4. Постреляционная
12) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие операции относятся к DML (языку обработки данных)?
1. CREATE
2. INSERT
3. UPDATE
4. DROP
13) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какой тип связи требует создания промежуточной таблицы?
1. 1:1
2. 1:M
3. M:M
4. N:1
14) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие преимущества дает нормализация базы данных?
1. Уменьшение избыточности данных
2. Упрощение физического хранения
3. Повышение целостности данных
4. Ускорение операций вставки
15) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какой метод индексации оптимален для поиска по диапазону значений?
1. B-дерево
2. Hash-индекс
3. Bitmap
4. Кластерный индекс
16) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие компоненты входят в архитектуру клиент-сервер?
1. Клиентское приложение
2. Сервер приложений
3. Сервер базы данных
4. Сетевой протокол
17) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какой тип данных используется для хранения изображений?
1. VARCHAR
2. BLOB
3. TEXT

4. CLOB
18) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие основные факторы влияют на производительность базы данных?
1. Тип индекса
2. Объем оперативной памяти
3. Количество пользователей
4. Нормализация таблиц
19) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какой этап проектирования идет после инфологического?
1. Физическое проектирование
2. Даталогическое проектирование
3. Концептуальное проектирование
4. Нормализация
20) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие основные типы целостности существуют в БД?
1. Целостность сущностей
2. Целостность ссылок
3. Доменная целостность
4. Транзакционная целостность
Индикатор Владеть
21) Прочитайте текст и запишите развернутый ответ:
В компании возникла необходимость хранения информации о студентах, преподавателях и предметах. Опишите, какие компоненты информационной системы потребуются для реализации этой задачи, и объясните их назначение.
22) Рассчитайте:
В базе данных хранится 10 миллионов записей. Среднее время доступа к одной записи без индекса составляет 0.001 секунды. После создания индекса время доступа сократилось до 0.0001 секунды. Определите, сколько времени экономится при выполнении запроса на выборку 1000 записей.
23) Определите:
В реляционной базе данных есть таблица “Заказы” с полями: ID заказа, дата, сумма, ID клиента, ID товара. Проанализируйте, к какой нормальной форме относится таблица, и обоснуйте свой ответ.
24) Прочитайте текст и запишите развернутый обоснованный ответ:
Компания планирует перейти на многоязычную поддержку в своей информационной системе. Опишите, какие изменения потребуются в структуре базы данных и какие проблемы могут возникнуть при реализации.
25) Вычислите:
В таблице “Клиенты” 1 миллион записей. Создается кластерный индекс по полю “Дата регистрации”. Оцените, какой объем дискового пространства потребуется для хранения индекса, если размер ключа составляет 8 байт.
26) Определите:
В системе управления складом необходимо реализовать иерархическую структуру хранения товаров (категория -> подкатегория -> товар). Предложите способ моделирования такой иерархии в реляционной базе данных и обоснуйте выбор.
27) Прочитайте текст и запишите развернутый ответ:
В компании возникла проблема с производительностью базы данных при выполнении сложных запросов. Предложите способы оптимизации, включая использование индексов и секционирования, и обоснуйте их эффективность.
28) Рассчитайте:
В таблице “Продажи” 100 миллионов записей. Средняя длина записи составляет 1000 байт. Определите оптимальный размер страницы для хранения данных, учитывая, что размер блока в файловой системе составляет 4 КБ.
29) Определите:
В базе данных необходимо реализовать систему версионности для отслеживания изменений в таблице “Сотрудники”. Опишите, какие подходы к хранению истории изменений существуют, и выберите наиболее подходящий для данной задачи.
30) Прочитайте текст и запишите развернутый обоснованный ответ:
Компания планирует переход на облачную СУБД. Проанализируйте преимущества и недостатки такого решения, учитывая следующие аспекты: производительность, безопасность, стоимость владения, масштабируемость и возможность восстановления после сбоев.

Вопросы к экзамену
Индикатор Знать
1) Прочитайте текст и установите правильную последовательность этапов проектирования:
1. Инфологическое проектирование
2. Даталогическое проектирование
3. Физическое проектирование
4. Концептуальное проектирование
2) Прочитайте текст и установите правильную последовательность операций при создании БД:
1. Создание таблиц
2. Создание схемы
3. Создание связей
4. Создание ограничений
3) Прочитайте текст и установите правильную последовательность этапов обработки данных:
1. Преобразование данных
2. Извлечение данных
3. Загрузка данных
4. Хранение данных
4) Прочитайте текст и установите правильную последовательность операций при оптимизации:
1. Создание индексов

Тип данных
1. Числовые
2. Строковые
3. Дата/время
4. Бинарные
Пример
А) 2025-05-07
Б) 12345
В) ‘Привет’
Г) Изображение
Д) Логическое значение
7) Установите соответствие между операциями DML и их назначением:
Операция
1. INSERT
2. UPDATE
3. DELETE
4. SELECT
Назначение
А) Извлечение данных
Б) Добавление данных
В) Удаление данных
Г) Изменение данных
Д) Создание структуры
8) Установите соответствие между типами связей и их характеристиками:
Связь
1. 1:1
2. 1:M
3. M:M
4. N:M
Характеристика
А) Используется редко в реальных системах
Б) Один ко многим
В) Многие ко многим
Г) Требует промежуточной таблицы
Д) Один к одному
9) Установите соответствие между методами индексации и их особенностями:
Метод
1. B-дерево
2. B+ дерево
3. Bitmap
4. Hash
Особенность
А) Использует битовые карты
Б) Оптимизирован для диапазона значений
В) Прямое хеширование
Г) Эффективен для редких значений
Д) Требует много памяти
10) Установите соответствие между типами целостности данных и их описанием:
Тип
1. Целостность сущностей
2. Целостность ссылок
3. Доменная целостность
4. Бизнес-правила
Описание
А) Ограничения на значения атрибутов
Б) Уникальность записей
В) Соответствие внешним ключам
Г) Специфические правила организации
Д) Контроль за транзакциями

Индикатор Уметь
11) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какие из перечисленных особенностей характерны для современного SQL?
1. Поддержка только реляционных операций
2. Возможность работы с JSON-данными
3. Отсутствие встроенных функций
4. Только статический анализ данных
12) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие команды относятся к DDL (языку определения данных)?

1. CREATE
2. UPDATE
3. DROP
4. SELECT
13) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какой оператор используется для сортировки результатов в SQL?
1. SORT
2. ORDER BY
3. ARRANGE
4. RANK
14) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие типы триггеров можно создать в SQL?
1. BEFORE INSERT
2. AFTER DELETE
3. DURING UPDATE
4. ON CHANGE
15) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какая роль имеет максимальный уровень привилегий в СУБД?
1. DBA
2. USER
3. ADMIN
4. ROOT
16) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие свойства входят в принцип ACID?
1. Atomicity
2. Complexity
3. Isolation
4. Dynamic
17)Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Какая модель данных используется в NoSQL базах?
1. Реляционная
2. Документ-ориентированная
3. Иерархическая
4. Сетевая
18) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие характеристики относятся к big data?
1. Volume (объем)
2. Velocity (скорость)
3. Validity (достоверность)
4. Value (ценность)
19) Выберите правильный ответ из предложенных вариантов и обоснуйте свой выбор:
Что такое транзакция в СУБД?
1. Единица работы с данными
2. Команда создания таблицы
3. Тип данных
4. Вид индекса
20) Выберите правильные ответы и запишите аргументы, обосновывающие выбор ответа:
Какие операции относятся к DML (языку манипуляции данными)?
1. INSERT
2. CREATE
3. UPDATE
4. DROP
Индикатор Владеть
21) Рассчитайте:
В интернет-магазине хранится информация о 2 миллионах товаров. Каждый товар имеет 15 атрибутов, средний размер записи составляет 1 КБ. Определите общий объем данных и предложите оптимальную схему хранения с учетом индексации для быстрого поиска по категориям и ценам.
22) Определите:
В системе бронирования авиабилетов необходимо хранить информацию о рейсах, пассажирах и бронированиях. Составьте ER-диаграмму для этой системы и опишите связи между сущностями. Укажите первичные и внешние ключи.
23) Вычислите:
В базе данных социальной сети 50 миллионов пользователей. Каждый пользователь в среднем имеет 200 друзей. Рассчитайте объем данных для хранения связей между пользователями и предложите способ оптимизации хранения этих данных.
24) Прочитайте текст и запишите развернутый ответ:
В системе онлайн-банкинга возникла проблема с производительностью при выполнении массовых операций перевода средств. Опишите, как можно оптимизировать процесс обработки транзакций с учетом требований целостности данных.
25) Определите:
В базе данных электронной библиотеки необходимо хранить книги, авторы которых могут писать как на русском, так и на иностранных языках. Предложите решение для корректного хранения и поиска интернациональных данных с учетом особенностей разных языков.
26) Рассчитайте:

29) Вычислите:
В системе учета заказов среднее время обработки запроса без индекса составляет 0.5 секунды, с индексом - 0.05 секунды. При этом создание и поддержание индекса требует дополнительных 0.01 секунды на каждую операцию записи. Определите, при каком количестве операций чтения в день оправдано использование индекса, если количество операций записи составляет 1000 в день.
30) Прочитайте текст и запишите развернутый ответ:
В системе управления отелем необходимо реализовать бронирование номеров с учетом различных типов номеров и дополнительных услуг. Опишите структуру базы данных и составьте SQL-запрос для поиска свободных номеров с определенными параметрами на заданный период.

Итоговое тестирование (ЗАЧЕТ С ОЦЕНКОЙ)

1. Что такое ER-модель в проектировании баз данных?
a) Модель, описывающая сущности и их связи
b) Модель, описывающая таблицы и поля в базе данных
c) Модель, описывающая запросы к базе данных
d) Модель, описывающая пользовательский интерфейс
2. Что такое схема базы данных?
a) описание данных, хранимых в БД
b) список таблиц в БД
c) тип БД
d) набор SQL-запросов
3. Что такое первичный ключ?
a) уникальный идентификатор строки в таблице
b) уникальный идентификатор таблицы
c) первая колонка в таблице
d) любой индекс в таблице
4. Что такое индекс в БД?
a) механизм для ускорения поиска данных в таблице
b) специальный тип данных, используемый для хранения дат
c) уникальный идентификатор таблицы
d) способ шифрования данных
5. Что такое проектирование базы данных?
a) процесс создания схемы БД
b) процесс наполнения БД данными
c) процесс удаления данных из БД
d) процесс написания SQL-запросов
6. Какой тип индекса хранит ссылки на строки таблицы в отсортированном порядке?
a) кластерный индекс
b) не кластерный индекс
c) полнотекстовый индекс
d) хеш-индекс
7. Какие типы связей могут существовать между таблицами в БД?
a) Один-к-одному, многие-ко-многим, многие-к-одному
b) Один-к-одному, многие-ко-многим, один-ко-многим
c) Один-ко-многим, многие-к-одному, многие-ко-многим
d) Линейные, круговые, древовидные
8. Какие ограничения могут быть установлены на столбцы таблицы?
a) NOT NULL, CHECK, UNIQUE, PRIMARY KEY
b) NOT NULL, CHECK, UNIQUE, PRIMARY, KEY, FOREIGN KEY
c) NOT NULL, CHECK, UNIQUE, PRIMARY KEY, FOREIGN KEY, DEFAULT
d) MIN, MAX, AVG, SUM
9. Как называется таблица, которая связывает две или более таблицы в реляционной базе данных?
A) Граф таблиц
B) Мета-таблица
C) Ссылочная таблица
D) Промежуточная таблица
E) Связующий массив
10. Какое свойство в реляционной базе данных определяет уникальность каждой строки в таблице?
A) Идентификатор
B) Первичный ключ
C) Внешний ключ
D) Индекс
E) Временная метка
11. Какой тип связи используется, если каждому элементу из одной таблицы соответствует несколько элементов из другой таблицы, и наоборот?
a) Один-к-одному
b) Один-ко-многим
c) Многие-ко-многим

C. Ключи должны быть уникальными, но могут содержать значения, которые повторяются в таблице.
D. Ключи могут быть пустыми (NULL).
13. Какое из следующих утверждений относительно нормализации баз данных верно?
A. Нормализация используется для увеличения объема данных в таблице.
B. Нормализация используется для уменьшения объема данных в таблице.
C. Нормализация не оказывает влияния на объем данных в таблице.
D. Нормализация используется для шифрования данных.
14. Какие из следующих утверждений относительно первичных ключей верны?
A. Первичные ключи могут содержать значения, которые повторяются в таблице.
B. Первичные ключи должны быть уникальными и не могут содержать значения, которые повторяются в таблице.
C. Первичные ключи должны быть уникальными, но могут содержать значения, которые повторяются в таблице.
D. Первичные ключи могут быть составными из нескольких полей.
15. Как называется процесс установки и настройки базы данных на сервере?
a) Развертывание
b) Удаление
c) Обновление
d) Миграция

Итоговое тестирование (ЭКЗАМЕН):

1. Какие языки программирования могут использоваться для работы с базами данных?
a) Python, Java, C++
b) HTML, CSS, JavaScript
c) SQL, PL/SQL, T-SQL
d) Pascal, Basic, Assembly
2. Что такое ER-модель?
a) Модель для описания бизнес-процессов
b) Модель для описания реляционных баз данных
c) Модель для описания графических интерфейсов
d) Модель для создания компьютерных игр
3. Какие типы связей могут быть в ER-модели?
a) Один к одному
b) Один ко многим
c) Многие ко многим
d) Все вышеперечисленные
e) Иерархические связи
4. Что такое нормализация в реляционных базах данных?
a) Процесс разбиения таблиц на более мелкие с целью уменьшения повторений данных
b) Процесс объединения таблиц с целью уменьшения потерь данных
c) Процесс создания дубликатов таблиц для более быстрого доступа к данным
d) Процесс шифрования данных в базе
5. Что такое SQL?
a) Язык программирования для создания веб-приложений
b) Язык запросов для работы с реляционными базами данных
c) Язык для создания мультимедийного контента
d) Язык для написания компьютерных вирусов
6. Какие из перечисленных объектов могут быть таблицами в реляционной базе данных?
a) Строки
b) Столбцы
c) Ячейки
d) Все перечисленные
e) Графики и диаграммы
7. Какие типы связей могут существовать между таблицами в реляционной базе данных?
a) Один-к-одному
b) Один-ко-многим
c) Многие-ко-многим
d) Все перечисленные
e) Циклические связи
8. Какой язык используется для работы с реляционными базами данных?
a) SQL
b) HTML
c) CSS
d) JavaScript
e) PHP
9. Как называется процесс создания структуры таблиц в реляционной базе данных?
a) Проектирование
b) Реализация

c) Администрирование
d) Оптимизация
e) Форматирование
10. Какой тип данных можно использовать для хранения текстовой информации в реляционной базе данных?
a) INT
b) VARCHAR
c) BOOLEAN
d) FLOAT
e) IMAGE
11. Какой уровень изоляции транзакций гарантирует наивысшую степень защиты от параллельного доступа к данным?
a) READ COMMITTED
b) REPEATABLE READ
c) SERIALIZABLE
d) FULL ACCESS
12. Каким образом можно обеспечить резервное копирование базы данных?
a) Ручным копированием файлов
b) С помощью специализированных программ для резервного копирования
c) Невозможно сделать резервную копию базы данных
d) Через социальные сети
13. Какие задачи выполняет администратор баз данных?
a) Проектирование баз данных
b) Разработка программного обеспечения
c) Управление и поддержка баз данных
d) Создание компьютерных игр
14. Какова основная задача SQL Server Agent?
a) Управление реляционной моделью данных
b) Планирование и выполнение заданий на сервере баз данных
c) Оптимизация производительности БД
d) Создание антивирусной защиты
15. Какой параметр конфигурации определяет максимальный объем памяти, который может быть использован SQL Server?
a) max_server_memory
b) max_db_memory
c) max_query_memory
d) max_user_memory

6.5. Примерная тематика курсовых работ (проектов)
Учебным планом не предусмотрено

6.6. Методические указания для обучающихся по освоению дисциплины (модуля)
Методические рекомендации по подготовке к лекционным занятиям

Просмотрите конспект сразу после занятий. Пометьте материал конспекта лекций, который вызывает затруднения для понимания. Попытайтесь найти ответы на затруднительные вопросы, используя предлагаемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь на текущей консультации или на ближайшей лекции за помощью к преподавателю. Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.
Работа с рекомендованной литературой:
При работе с основной и дополнительной литературой целесообразно придерживаться такой последовательности. Сначала прочитать весь заданный текст в быстром темпе. Цель такого чтения заключается в том, чтобы создать общее представление об изучаемом материале, понять общий смысл прочитанного. Затем прочитать вторично, более медленно, чтобы в ходе чтения понять и запомнить смысл каждой фразы, каждого положения и вопроса в целом. Чтение приносит пользу и становится продуктивным, когда сопровождается записями. Это может быть составление плана прочитанного текста, тезисы или выписки, конспектирование и др. Выбор вида записи зависит от характера изучаемого материала и целей работы с ним. Если содержание материала несложное, легко усваиваемое, можно ограничиться составлением плана. Если материал содержит новую и трудно усваиваемую информацию, целесообразно его законспектировать. План – это схема прочитанного материала, перечень вопросов, отражающих структуру и последовательность материала. Конспект – это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов: - план-конспект – это развернутый детализированный план, в котором по наиболее сложным вопросам даются подробные пояснения, - текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника, - свободный конспект – это четко и кратко изложенные основные положения в результате глубокого изучения материала, могут присутствовать

выписки, цитаты, тезисы; часть материала может быть представлена планом, - тематический конспект – составляется на основе изучения ряда источников и дает ответ по изучаемому вопросу. В процессе изучения материала источника и составления конспекта нужно обязательно применять различные выделения, подзаголовки, создавая блочную структуру конспекта. Это делает конспект легко воспринимаемым и удобным для работы.

Методические рекомендации по подготовке к практическим занятиям

Практические занятия представляют особую форму сочетания теории и практики. Их назначение – углубление проработки теоретического материала предмета путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к практическим занятиям включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение практического занятия предполагает, например:
- индивидуальные выступления студентов с сообщениями по какому-либо вопросу изучаемой темы;
- фронтальное обсуждение рассматриваемой проблемы, обобщения и выводы;
- решение задач и упражнений по образцу;
- решение вариантных задач и упражнений;
- решение ситуационных производственных (профессиональных) задач;
- проектирование и моделирование разных видов и компонентов профессиональной деятельности;
- выполнение контрольных работ;
- работу с тестами.
При подготовке к практическим занятиям студентам рекомендуется: внимательно ознакомиться с тематикой практического занятия; прочесть конспект лекции по теме, изучить рекомендованную литературу; составить краткий план ответа на каждый вопрос практического занятия; проверить свои знания, отвечая на вопросы для самопроверки; если встретятся незнакомые термины, обязательно обратиться к словарю и зафиксировать их в тетради. Все письменные задания выполнять в рабочей тетради. Практические занятия развивают у студентов навыки самостоятельной работы по решению конкретных задач.

Методические рекомендации по подготовке к лабораторным работам

Лабораторные работы представляют одну из форм освоения теоретического материала с одновременным формированием практических навыков в изучаемой дисциплине (модуле). Их назначение – углубление проработки теоретического материала, формирование практических навыков путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к лабораторным работам включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение лабораторной работы предполагает:
- изучение теоретического материала по теме лабораторной работы (по вопросам изучаемой темы);
- выполнение необходимых расчетов и экспериментов;
- оформление отчета с заполнением необходимых таблиц, построением графиков, подготовкой выводов по проделанным экспериментам и теоретическим расчетам;
- по каждой лабораторной работе проводится контроль: проверяется содержание отчета, проверяется усвоение теоретического материала.
Контроль усвоения теоретического материала является индивидуальным.

Методические указания по выполнению отчёта к лабораторным работам

Основным требованием по выполнению лабораторных и практических работ является полное исчерпывающее описание всей проделанной работы, позволяющее судить о полученных результатах, степени выполнения и профессиональной подготовки студентов.
Методические указания обеспечивают комплексный подход в учебной работе студентов, единство и преемственность требований к оформлению результатов работы на разных этапах обучения. С единых позиций приведены основные требования к структуре, оформлению и содержанию отчета по лабораторным и практическим работам.
Структура отчёта:
- цель работы;
- краткие теоретические сведения;
- ход выполнения работы;
- выводы.
Дополнительные элементы:
- приложения;
- библиографический список.
Требования к содержанию отчёта:
1. Титульный лист
В верхнем поле листа указывают полное наименование учебного заведения.
В среднем поле указывается вид работы, в данном случае лабораторная или практическая работа с указанием курса, по которому она выполнена, и ниже ее название. Название работы приводится без слова "тема" и в кавычки не заключается.
Далее ближе к правому краю титульного листа указывают фамилию, инициалы и группу учащегося, выполнившего работу, а также фамилию, инициалы преподавателя, принявшего работу.
В нижнем поле листа указывается место выполнения работы и год ее написания (без слова год).
2. Цель работы должна отражать тему работы, а также конкретные задачи, поставленные студенту на период выполнения работы. По объему цель работы в зависимости от сложности и многозадачности работы составляет от нескольких строк до

0,5 страницы.
3. Краткие теоретические сведения. В этом разделе излагается краткое теоретическое описание изучаемой в работе темы. Материал раздела не должен копировать содержание методического пособия или учебника по данной теме, а ограничивается изложением основных понятий, требующихся для дальнейшей обработки полученных результатов. Объем литературного обзора не должен превышать 1/3 части всего отчета.
4. Ход выполнения работы. В данном разделе подробно излагается методика выполнения работы, процесс получения данных и способ их обработки. Если используются стандартные пакеты компьютерных программ для обработки экспериментальных результатов, то необходимо обосновать возможность и целесообразность их применения, а также подробности обработки данных с их помощью.
5. Выводы по работе - кратко излагаются результаты работы, полученные в результате выполнения работы, а также краткий анализ полученных результатов.
Отчет по лабораторной работе оформляется на листе формата А4. Допускается оформление отчета по лабораторной работе в электронном виде средствами Microsoft Office. Текст работы должен быть напечатан через полтора интервала шрифтом Times New Roman, кегль – 12. Поля должны оставаться по всем четырем сторонам печатного листа: левое – не менее 30 мм, правое – не менее 10, нижнее – не менее 20 и верхнее – не 15 мм.
Для защиты лабораторной работы студент должен подготовить отчет, провести самостоятельную работу, иметь отметку о проверенном отчете.
Результаты определяются по пятибалльной системе оценок.

Методические рекомендации по выполнению реферата

Реферат – письменная работа объемом 8–10 страниц. Это краткое и точное изложение сущности какого-либо вопроса, темы. Тему реферата студент выбирает из предложенных преподавателем или может предложить свой вариант. В реферате нужны развернутые аргументы, рассуждения, сравнения. Содержание темы излагается объективно от имени автора. Функции реферата: информативная, поисковая, справочная, сигнальная, коммуникативная. Степень выполнения этих функций зависит от содержательных и формальных качеств реферата и для каких целей их использует. Требования к языку реферата: должен отличаться точностью, краткостью, ясностью и простотой.
Структура реферата:
1. Титульный лист
2. Оглавление (на отдельной странице). Указываются названия всех разделов (пунктов плана) реферата и номера страниц, указывающие начало этих разделов в тексте реферата.
3. Введение. Аргументируется актуальность исследования, т.е. выявляется практическое и теоретическое значение данного исследования. Далее констатируется, что сделано в данной области предшественниками, перечисляются положения, которые должны быть обоснованы. Обязательно формулируются цель и задачи реферата.
4. Основная часть. Подчиняется собственному плану, что отражается в разделении текста на главы, параграфы, пункты. План основной части может быть составлен с использованием различных методов группировки материала. В случае если используется чья-либо неординарная мысль, идея, то обязательно нужно сделать ссылку на того автора, у кого взят данный материал.
5. Заключение. Последняя часть научного текста. В краткой и сжатой форме излагаются полученные результаты, представляющие собой ответ на главный вопрос исследования.
6. Приложение. Может включать графики, таблицы, расчеты.
7. Библиография (список литературы). Указывается реально использованная для написания реферата литература. Названия книг располагаются по алфавиту с указанием их выходных данных. Общие требования к построению, содержанию и оформлению».
При проверке реферата оцениваются:
- знание фактического материала, усвоение общих представлений, понятий, идей;
- характеристика реализации цели и задач исследования;
- степень обоснованности аргументов и обобщений;
- качество и ценность полученных результатов;
- использование литературных источников;
- культура письменного изложения материала;
- культура оформления материалов работы.
Правила написания научных текстов (реферат):
Здесь приводятся рекомендации по консультированию студентов относительно данного вида самостоятельной работы. Во время консультаций руководителю следует предложить к обсуждению следующие вопросы.
• Какова истинная цель Вашего научного текста – это поможет Вам разумно распределить свои силы и время.
• Важно разобраться, кто будет «читателем» Вашей работы.
• Начинать писать серьезную работу следует не раньше, чем возникнет ощущение, что по работе с источниками появились идеи, которыми можно поделиться.
• Должна быть идея, а для этого нужно научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного).
• Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно, а также стремясь структурировать свой текст.
• Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Методические рекомендации по выполнению контрольных работ

Контрольная работа выполняется по вариантам. На бланке указывается факультет, курс, группа, ФИО студента. Вопросы строятся на основе тестовых и ситуативных заданий. В тестовых заданиях, выбирается правильный(ые) ответ(ы). При решении ситуативных заданий выбирается правильная последовательность действий в рассматриваемой ситуации. Проверка контрольной работы позволяет выявить и исправить допущенные студентами ошибки, указать, какие вопросы дисциплины (модуля) ими недостаточно усвоены и требуют доработки. Студент должен внимательно ознакомиться с письменными замечаниями преподавателя и приступить к их исправлению, для чего еще раз повторить соответствующий материал.

Методические рекомендации по подготовке к коллоквиуму

Коллоквиумом называется собеседование преподавателя и студента по заранее определенным контрольным вопросам. Целью коллоквиума является формирование у студента навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы. На коллоквиум выносятся крупные, проблемные, нередко спорные теоретические вопросы. Упор делается на монографические работы профессора-автора данного спецкурса. От студента требуется:
- владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;
- знание разных точек зрения, высказанных в научной литературе по соответствующей проблеме, умение сопоставлять их между собой;
- наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.
Коллоквиум - это не только форма контроля, но и метод углубления, закрепления знаний студентов, так как в ходе собеседования преподаватель разъясняет сложные вопросы, возникающие у студента в процессе изучения данного источника. Однако коллоквиум не консультация и не экзамен. Его задача добиться глубокого изучения отобранного материала, пробудить у студента стремление к чтению дополнительной социологической литературы. Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует литературу для изучения и объясняет процедуру проведения коллоквиума. Как правило, на самостоятельную подготовку к коллоквиуму студенту отводится 3-4 недели. Методические указания состоят из рекомендаций по изучению источников и литературы, вопросов для самопроверки и кратких конспектов ответа с перечислением основных фактов и событий, относящихся к пунктам плана каждой темы. Это должно помочь студентам целенаправленно организовать работу по овладению материалом и его запоминанию. При подготовке к коллоквиуму следует, прежде всего, просмотреть конспекты лекций и практических занятий и отметить в них имеющиеся вопросы коллоквиума. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.
Коллоквиум проводится в форме индивидуальной беседы преподавателя с каждым студентом или беседы в небольших группах (2-3 человека). Обычно преподаватель задает несколько кратких конкретных вопросов, позволяющих выяснить степень добросовестности работы с литературой, проверяет конспект. Далее более подробно обсуждается какая-либо сторона проблемы, что позволяет оценить уровень понимания. По итогам коллоквиума выставляется дифференцированная оценка по пятибалльной системе.

Методические рекомендации по устному опросу/самоподготовке

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, используя лист опорных сигналов,
воспроизвести по памяти определения, выводы формул, формулировки основных положений и доказательств. В случае необходимости следует рекомендовать еще раз внимательно разобраться в материале. Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала – умение решать задачи или пройти тестирование по пройденному материалу. Однако преподавателю следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

Методические рекомендации по подготовке к семинарским занятиям

Одним из видов внеаудиторной самостоятельной работы является подготовка к семинарским занятиям. Семинар – форма учебно-практических занятий, при которой студенты обсуждают сообщения, доклады и рефераты, выполненные ими по результатам учебных или научных исследований под руководством преподавателя. Преподаватель в этом случае является координатором обсуждений темы семинара, подготовка к которому является обязательной. Поэтому тема семинара и основные источники обсуждения предъявляются до обсуждения для детального ознакомления, изучения. Цели обсуждений направлены на формирование навыков профессиональной полемики и закрепление обсуждаемого материала. Семинар – это такая форма организации обучения, при которой на этапе подготовки доминирует самостоятельная работа учащихся с учебной литературой и другими дидактическими средствами над серией вопросов, проблем и задач, а в процессе семинара идут активное обсуждение, дискуссии и выступления учащихся, где они под руководством преподавателя делают обобщающие выводы и заключения. Семинар предназначен для углубленного изучения дисциплины, овладения методологией научного познания, то главная цель семинарских занятий – обеспечить студентам возможность овладеть навыками и умениями использования теоретического знания применительно к особенностям изучаемой отрасли.

Методические рекомендации по подготовке к эссе

Одним из видов самостоятельной работы студентов является написание творческой работы по заданной либо

согласованной с преподавателем теме. Творческая работа (эссе) представляет собой оригинальное произведение объемом 500-700 слов, посвященное какой-либо значимой классической либо современной проблеме в определенной теоретической и практической области. Творческая работа не является рефератом и не должна носить описательный характер, большое место в ней должно быть уделено аргументированному представлению своей точки зрения студентами, критической оценке рассматриваемого материала и проблематики, что должно способствовать раскрытию творческих и аналитических способностей. Цели написания эссе – научиться логически верно и аргументировано строить устную и письменную речь; работать над углублением и систематизацией своих философских знаний; овладеть способностью использовать основы знаний для формирования мировоззренческой позиции. Приступая к написанию эссе, изложите в одном предложении, что именно вы будете утверждать и доказывать (свой тезис).

Методические рекомендации по подготовке к докладу

Для подготовки доклада необходимо выбрать актуальную тему. Желательно, чтобы тема была интересна докладчику и вызывала желание качественно подготовить материалы. Подготовка доклада предполагает: определение цели доклада; подбор необходимого материала, определяющего содержание доклада; составление плана доклада, распределение собранного материала в необходимой логической последовательности.
Композиция доклада имеет вступление, основную часть и заключение.
Вступление должно содержать: название доклада; сообщение основной идеи; современную оценку предмета изложения; краткое перечисление рассматриваемых вопросов; интересную для слушателей форму изложения. Основная часть, в которой необходимо раскрыть суть темы, обычно строится по принципу отчёта. Задача основной части: представить достаточно данных для того, чтобы слушатели заинтересовались темой.
Заключение – чёткое обобщение и краткие выводы по излагаемой теме.

Методические рекомендации по подготовке к собеседованию

Собеседование – средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.
Цель собеседования: проверка усвоения знаний; умений применять знания; сформированности профессионально значимых личностных качеств.
Подготовка к собеседованию предполагает повторение пройденного материала и приобретение навыка свободного владения терминологией и фактическими данными по определенному разделу дисциплины (модуля).

Методические рекомендации по подготовке к тестированию/ итоговому тестированию

Тестирование/итоговое тестирование – это система стандартизированных заданий, определенного содержания и упорядоченных в рамках определенной стратегии предъявления, позволяющая качественно оценить структуру и эффективно измерить уровень знаний, умений и навыков по дисциплине (модулю). Тестирование позволяет оценить у студентов уровень знаний фактического материала, умения логически мыслить, способности к рефлексии и творческому подходу к решению поставленной задачи на каждом этапе их обучения.
При самостоятельной подготовке к тестированию студенту необходимо:
- четко выяснить все условия тестирования заранее (сколько тестов будет предложено, сколько времени отводится на тестирование, какова система оценки результатов);
- проконсультироваться с преподавателем по вопросу выбора учебной литературы;
- проработать информационный материал по разделу дисциплине;
- конспективно выписать из соответствующих разделов учебной дисциплины основные понятия, ключевые идеи, теоретические концепции и подходы;
- повторить теоретический материал лекций, самостоятельно изученной учебной и научной литературы, семинарских занятий.
При выполнении тестовых заданий рекомендуется:
- внимательно полностью прочитать вопрос и предлагаемые варианты ответов, выбрать правильные из них (их может быть несколько);
- при затруднении при ответе на сложный для студента вопрос целесообразно переходить к другим вопросам теста. По завершению ответов на все последующие вопросы теста необходимо вернуться к трудному вопросу и дать на него продуманный ответ;
- по окончании тестирования следует проверить правильность данных ответов на вопросы теста, чтобы избежать механических ошибок.

Методические рекомендации по подготовке к экзамену

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной дисциплине (модулю). Экзаменационная сессия – это серия экзаменов, установленных учебным планом. Между экзаменами интервал 2-4 дня, в течение студент систематизирует уже имеющиеся знания. На консультации перед экзаменом студенты должны быть ознакомлены с основными требованиями и получить ответы на возникающие в процессе подготовки
вопросы. Необходимо ориентировать студентов на систематическую подготовку к занятиям в течение семестра, что

позволит использовать время экзаменационной сессии для систематизации знаний.

Методические рекомендации по подготовке к зачету/зачету с оценкой

В ходе подготовки к зачету/зачету с оценкой студент, в первую очередь, должен систематизировать знания, полученные в ходе изучения дисциплины. К зачету/зачету с оценкой необходимо готовиться целенаправленно, регулярно, систематически и с первых дней обучения по данной дисциплине (модулю). В самом начале учебного курса познакомьтесь со следующей учебно-методической документацией:
- программой дисциплины (модуля);
- перечнем знаний, умений и навыков, которыми студент должен владеть;
- тематическими планами лекций, семинарских занятий;
- учебниками, учебными пособиями по дисциплине, а также электронными ресурсами;
- перечнем вопросов к зачету/зачету с оценкой.

После этого у обучающихся должно сформироваться четкое представление об объеме и характере знаний и умений, которыми надо будет овладеть по дисциплине (модулю). Систематическое выполнение учебной работы на лекциях и лабораторных занятиях позволит успешно освоить дисциплину (модуль) и создать хорошую базу для сдачи зачета/зачета с оценкой.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Рекомендуемая литература	
7.1.1. Основная литература	
Л.1.1	Кондрашов Ю.Н. Язык SQL. Сборник ситуационных задач по дисциплине «Базы данных» [Электронный ресурс]:Учебно-практическое пособие. - Москва: Русайнс, 2023. - 125 с. – Режим доступа: https://book.ru/book/947081
Л.1.2	Мамедли Р. Э. Базы данных. Лабораторный практикум [Электронный ресурс]:учебное пособие для вузов. - Санкт-Петербург: Лань, 2023. - 152 с. – Режим доступа: https://e.lanbook.com/book/319400
Л.1.3	Васильева М. А., Филипченко К. М., Балакина Е. П. Информационное обеспечение систем управления. Проектирование базы данных с заданиями [Электронный ресурс]:учебник для вузов. - Санкт-Петербург: Лань, 2023. - 200 с. – Режим доступа: https://e.lanbook.com/book/339740
Л.1.4	Волк В. К. Базы данных. Проектирование, программирование, управление и администрирование [Электронный ресурс]:. - Санкт-Петербург: Лань, 2022. - 244 с. – Режим доступа: https://e.lanbook.com/book/193373
Л.1.5	Ратушняк Г.Я., Золкин А.Л., Никитин А.Л. Базы данных [Электронный ресурс]:Учебное пособие. - Москва: Русайнс, 2022. - 127 с. – Режим доступа: https://book.ru/book/947106
Л.1.6	Кумскова И.А. Базы данных [Электронный ресурс]:Учебник. - Москва: КноРус, 2022. - 400 с. – Режим доступа: https://book.ru/book/943244
7.1.2. Дополнительная литература	
Л.2.1	Волк В. К. Базы данных. Проектирование, программирование, управление и администрирование [Электронный ресурс]:. - Санкт-Петербург: Лань, 2021. - 244 с. – Режим доступа: https://e.lanbook.com/book/176670
Л.2.2	Кондрашов Ю.Н. Язык SQL. Сборник ситуационных задач по дисциплине «Базы данных» [Электронный ресурс]:Учебно-практическое пособие. - Москва: Русайнс, 2021. - 125 с. – Режим доступа: https://book.ru/book/942020
7.2. Лицензионное и свободно распространяемое программное обеспечение в том числе отечественного производства	
7.2.1	1С:Предприятие 8. Комплект для обучения в высших и средних учебных заведениях
7.2.2	СПС КонсультантПлюс. Версия Проф
7.2.3	Информационно-поисковая правовая справочная система «Гарант»
7.2.4	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition
7.2.5	Microsoft Office 2013 Pro Plus
7.2.6	Microsoft Windows 10 Pro
7.3. Перечень профессиональных баз данных, информационных справочных систем и ресурсов сети Интернет	
7.3.1	Электронно-библиотечная система "Лань". Режим доступа: https://e.lanbook.com/
7.3.2	Электронно-библиотечная система "Университетская библиотека онлайн". Режим доступа: https://biblioclub.ru/
7.3.3	Электронно-библиотечная система "BOOK.ru". Режим доступа: https://book.ru/
7.3.4	Сайт национального открытого университета "ИНТУИТ". Режим доступа: https://intuit.ru/
7.3.5	ПЛАТФОРМА ОНЛАЙН-ОБРАЗОВАНИЯ «РАЗУМ». Режим доступа: https://razoom.mgutm.ru/

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1	Адрес: 215119, Смоленская область, Вяземский район, город Вязьма, ул. Ленина, д. 77: аудитория 18-1312 - Лаборатория Учебная аудитория для проведения занятий лекционного типа, лабораторных и практических занятий; занятий семинарского типа; для курсового проектирования (выполнения курсовых работ); для проведения групповых и индивидуальных консультаций; для текущего контроля и промежуточной аттестации : Рабочие места обучающихся; Рабочее место преподавателя; Маркерная доска; 15 рабочих мест обучающихся оснащенные ПЭВМ с подключением к сети интернет и обеспечением доступа в электронную информационно-образовательную среду; Учебно-наглядные пособия
8.2	Адрес: 215119, Смоленская область, Вяземский район, город Вязьма, ул. Ленина, д. 77: аудитория 18-1310 - Лаборатория Учебная аудитория для проведения занятий лекционного типа; занятий семинарского типа, лабораторных и практических занятий; для курсового проектирования (выполнения курсовых работ); для проведения групповых и индивидуальных консультаций; для текущего контроля и промежуточной аттестации : Рабочие места обучающихся; Рабочее место преподавателя; Классная доска; 15 рабочих мест обучающихся оснащенные ПЭВМ с подключением к сети интернет и обеспечением доступа в электронную информационно-образовательную среду; Ноутбук переносной; Переносной проектор; Переносной экран; Учебно-наглядные пособия

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Организация образовательного процесса для лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Методическими рекомендациями по организации образовательного процесса для инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса» Министерства образования и науки РФ от 08.04.2014г. № АК-44/05вн. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Студенты с ограниченными возможностями здоровья, в отличие от остальных студентов, имеют свои специфические особенности восприятия, переработки материала. Подбор и разработка учебных материалов производится с учетом индивидуальных особенностей. Предусмотрена возможность обучения по индивидуальному графику, при составлении которого возможны различные варианты проведения занятий: в академической группе и индивидуально, на дому с использованием дистанционных образовательных технологий.

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Системной автоматизации и информационных технологий
Протокол от __ _____ 2027 г. № __
Зав. кафедрой Кузьмин Константин Анатольевич_____

Рабочая программа согласована на заседании выпускающей кафедры
Системной автоматизации и информационных технологий
Протокол от __ _____ 2027 г. № __
Зав. кафедрой Кузьмин Константин Анатольевич _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Системной автоматизации и информационных технологий
Протокол от __ _____ 2028 г. № __
Зав. кафедрой Кузьмин Константин Анатольевич_____

Рабочая программа согласована на заседании выпускающей кафедры
Системной автоматизации и информационных технологий
Протокол от __ _____ 2028 г. № __
Зав. кафедрой Кузьмин Константин Анатольевич _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Системной автоматизации и информационных технологий
Протокол от __ _____ 2029 г. № __
Зав. кафедрой Кузьмин Константин Анатольевич_____

Рабочая программа согласована на заседании выпускающей кафедры
Системной автоматизации и информационных технологий
Протокол от __ _____ 2029 г. № __
Зав. кафедрой Кузьмин Константин Анатольевич _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
_____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры
Системной автоматизации и информационных технологий
Протокол от __ _____ 2030 г. № __
Зав. кафедрой Кузьмин Константин Анатольевич_____

Рабочая программа согласована на заседании выпускающей кафедры
Системной автоматизации и информационных технологий
Протокол от __ _____ 2030 г. № __
Зав. кафедрой Кузьмин Константин Анатольевич _____

Тема 1 Вводные понятия

1.1. Установка сервера MySQL 5

Для создания баз данных и работы с ними потребуется установить сервер MySQL и утилиты, которые поставляются вместе с сервером. Дистрибутив сервера можно взять с сайта **dev.mysl.com.** Это должен быть zip-архив из раздела **Windows downloads** (Windows (x86)). Далее в этом разделе описывается установка, настройка и работа с MySQL 5.

Разархивируйте пакет и запустите на выполнение файл с расширением exe, затем выберите типичный способ установки. После установки сконфигурируйте сервер с помощью мастера (wizard), выбрав вначале стандартный вариант конфигурации сервера.



Рис. 1. Выбор режима конфигурации сервера MySQL.

Выберите установку сервера MySQL (см. рис. 1) как сервиса Windows, тогда MySQL будет запускаться при старте системы автоматически. Выберите имя для сервиса, например, MySQL 5. Под этим именем вы в дальнейшем найдете его в списке сервисов в окне Администрирование|Службы.

Укажите, что необходимо включить путь до каталога bin в переменную PATH. Переменная PATH содержит имена всех каталогов, в которых система ищет программу, запускаемую пользователем. Если окажется, что программа находится в каталоге, не

указанном в переменной PATH, то Windows не сможет ее найти и сообщит о невозможности запуска приложения.

В каталоге bin сервера MySQL содержатся те файлы, которые потребуется запускать при старте сервера, а также утилиты, которые позволят вам связываться с сервером напрямую из командной строки.

На следующем этапе задайте пароль для суперпользователя root (рис. 2).



Рис. 2. Конфигурирование доступа к серверу MySQL

ПРИМЕЧАНИЕ

Суперпользователь root — это владелец системы, он может выполнять любые действия по управлению сервером и администрированию баз данных.

Требования к качеству пароля здесь нежесткие, можете выбрать любое словарное слово, записав его латиницей. Не разрешайте удаленный доступ к серверу для суперпользователя.

ПРИМЕЧАНИЕ

Разрешение удаленного доступа к серверу существенно ослабит его безопасность и может быть разрешено, только если доступ осуществляется с шифрацией передаваемых данных.

Создайте учетную запись для анонимного пользователя.

ПРИМЕЧАНИЕ

Анонимный пользователь заходит в систему и при этом получает ограниченные права, в частности, он не имеет доступа к системным базам данных.

Завершите установку и удостоверьтесь, что сервис MySQL запущен. Найдите среди установленных на вашем компьютере программ группу MySQL и запустите клиентскую программу доступа к серверу MySQL **Command Line Client** из командной строки.

Если установка завершилась благополучно, то сервер у вас уже запущен, клиентская утилита mysql установит соединение с сервером и предложит ввести пароль. Введите пароль суперпользователя. Таким образом, вы соединитесь с сервером MySQL под именем суперпользователя root, т. е. будете иметь все возможные права для работы. Для завершения сеанса работы с сервером MySQL достаточно дать команду quit в окне клиентского приложения, и его окно закроется.

1.2. Настройка сервера MySQL 5

После установки сервера не стоит сразу создавать базы с данными, записанными в какой-либо кодировке русского языка. Начиная с 4.1 версии MySQL, задать кодировку можно не только для всего сервера, но и для отдельной базы, таблицы и даже столбца. Причем в вашей базе данных столбцы могут иметь как различную кодировку, так и разные правила сравнения данных.

Сравнение или сопоставление (от англ, *collation*) – это способ упорядочивания строк, например, по алфавиту. При этом можно учитывать регистр (строчная или заглавная попалась буква), а можно игнорировать его. При сопоставлении можно задать порядок в соответствии с кодировкой символов. Кроме того, вы можете указать, что сравнивать придется двоичный код или текст. Это все вместе и называется сопоставлением.

При установке сервера MySQL в Windows оказывается необходимым выполнить настройку сервера с учетом кодировки и способа сравнения данных в базе, иначе русские слова будут отображаться в виде строк из вопросительных знаков.

Для настройки MySQL откройте каталог, в который был установлен сервер, и найдите файл my.ini. В этот конфигурационный файл и необходимо внести приведенные далее директивы. Найдите в файле my.ini раздел, касающийся настройки самого сервера, он помечен символами [mysqld]. Строки, начинающиеся с символа решетки (#), являются комментариями. Найдите директиву default-character-set (использовать по умолчанию указанную кодировку), измените ее и добавьте еще 3 директивы так, чтобы этот фрагмент файла имел вид:

```
default-character-set=cp1251
```

character-set-server=cp1251

init-connect="SET NAMES cp1251"

skip-character-set-client-handshake

Две последние строки принудительно устанавливают кодировку cp1251 для всех запросов. Эти директивы позволяют создавать таблицы в кодировке cp1251 и благополучно выполнять запросы, используя русскоязычные строки.

ПРИМЕЧАНИЕ

Не стоит создавать столбцы, таблицы и базы с русскими именами. Иначе в самых неожиданных местах вас подстерегают ошибки и некорректная обработка данных.

1.3. Контрольные вопросы

1. Для чего нужна переменная PATH?
2. Кто такой «Суперпользователь root»?
3. Какие права у анонимного пользователя?
4. Как определить, что установка MySQL завершилась удачно?
5. Для каких элементов СУБД MySQL можно задавать кодировку?
6. Какой файл является конфигурационным в СУБД MySQL?

Тема 2 Проектирование реляционных баз данных

2.1. Описание предметной области

Рассмотрим работу фирмы, владеющей несколькими такси и нанимающей водителей для работы на них. Руководство фирмы решило хранить информацию об автомобилях и водителях, работающих в компании, так, чтобы информация была полной, не повторялась многократно в разных файлах, и легко было найти нужные данные.

Такую информацию целесообразно хранить в базе данных. Реляционная база данных состоит из именованных таблиц. Таблица состоит из строк (записей) и столбцов (полей). Первая таблица cars содержит сведения об автомобилях:

- модель;

- год выпуска;
- государственный регистрационный номер;
- цвет;
- учетный номер автомобиля в автопарке.

Вторая таблица, назовем ее drivers, предназначена для информации о водителях:

- имя;
- отчество;
- фамилия;
- дата рождения;
- домашний адрес;
- дата приема на работу;
- учетный номер водителя.

Таблица timetable содержит данные о расписании использования того или иного автомобиля:

- дата;
- учетный номер автомобиля в автопарке;
- учетный номер водителя.

Таблица cars связана с таблицей timetable. Схема базы данных такой фирмы представлена на рис. 3.

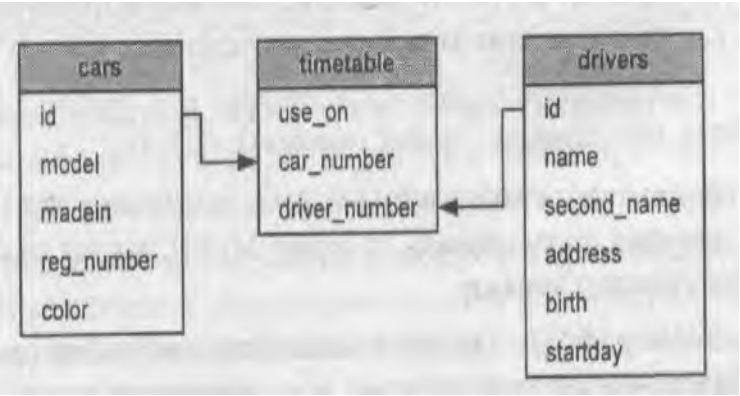

Рис. 3. Схема базы данных таксопарка

Одна и та же машина, описанная в таблице cars, может несколько раз упоминаться в таблице timetable. Такая связь называется отношением "один-ко-многим" (1:N). Аналогично каждый водитель из таблицы drivers может несколько раз упоминаться в таблице timetable. Отношения 1:N являются основными отношениями, поддерживаемыми серверами баз данных. Кроме этого, может встречаться отношение "один-к-одному" (1 : 1). При поступлении на работу водитель представил рекомендации. Их тоже можно занести в базу, заведя для этой цели новый столбец. Но рекомендации – это длинный текст, и есть они не у всех, значит, несколько полей останутся пустыми, это

будет необоснованной тратой памяти. Можно завести отдельную таблицу driver_ref, содержащую два столбца – учетный номер водителя и сам столбец с текстом рекомендации. Отношение между таблицами drivers и driver_ref будет отношением "один-к-одному", ведь у каждого водителя есть только один текст рекомендаций.

Обычно, если вы проектируете базу и обнаруживаете, что между таблицами существует связь "один-к-одному", это означает, что таблицы можно объединить. Проверяйте, не напрасно ли вы создаете дополнительные таблицы, но в данном примере создание таблицы оправдано.

В реляционной базе данных нельзя напрямую создать отношение "многие-ко-многим" *(M:N*). Его необходимо преобразовать в два отношения 1:N, устанавливаемых с промежуточной таблицей.

Если значения одного поля таблицы представлены в поле другой таблицы, то говорят, что первое поле ссылается на второе. Когда один столбец в таблице ссылается на другой, он называется *внешним ключом;* а столбец, на который он ссылается, называется *родительским ключом.* Ключи служат связующим звеном между таблицами. Внешний ключ (foreign key) – это столбец, который соответствует первичному ключу другой таблицы.

В нашем примере учетный номер автомобиля в автопарке и учетный номер водителя являются первичными ключами в таблицах cars и drivers и внешними ключами в таблице timetable. Сервер проверяет соответствие значений в столбцах, по которым происходит связь. Вы не сможете ввести такое значение в поле внешнего ключа, которое отсутствует в поле первичного ключа. То есть надо сначала принять водителя на работу и присвоить ему номер, а уж потом использовать этот номер в расписании автопарка.

Значения в поле первичного ключа должны быть уникальными.

Полный набор таблиц для базы данных называется *схемой базы данных.* Схема должна показывать таблицы вместе с их столбцами, типы данных столбцов, указывать первичный ключ каждой таблицы и все внешние ключи.

2.2. Работа с клиентской программой mysql

Утилита mysql позволяет осуществить соединение с сервером MySQL и давать команды SQL для создания баз, записи в них данных и выборки данных из баз. В версии MySQL 5 реализован оконный интерфейс для работы с mysql. Его можно вызвать, выбрав пункт меню MySQL **Command Line Client** в группе программ MySQL.

При запуске утилита mysql установит соединение с сервером и предложит ввести пароль. Введите пароль суперпользователя root.

Команда на языке SQL обычно состоит из выражения, за которым следует точка с запятой. Когда пользователь вводит команду, mysql отправляет ее серверу для выполнения и выводит на экран сначала результаты, а затем — новую строку mysql>, что означает готовность к выполнению новых команд.

Строка mysql> называется *приглашением командной строки.*

Утилита mysql выводит результаты работы запроса, если таковые есть, в виде таблицы (строк и столбцов). В первой строке этой таблицы содержатся заголовки столбцов, а в следующих строках — собственно результаты. Обычно заголовками столбцов становятся имена, полученные из таблиц базы (рис. 4).

Утилита mysql также сообщает количество возвращаемых строк и время выполнения запроса, что позволяет в некоторой степени составить представление о производительности сервера.

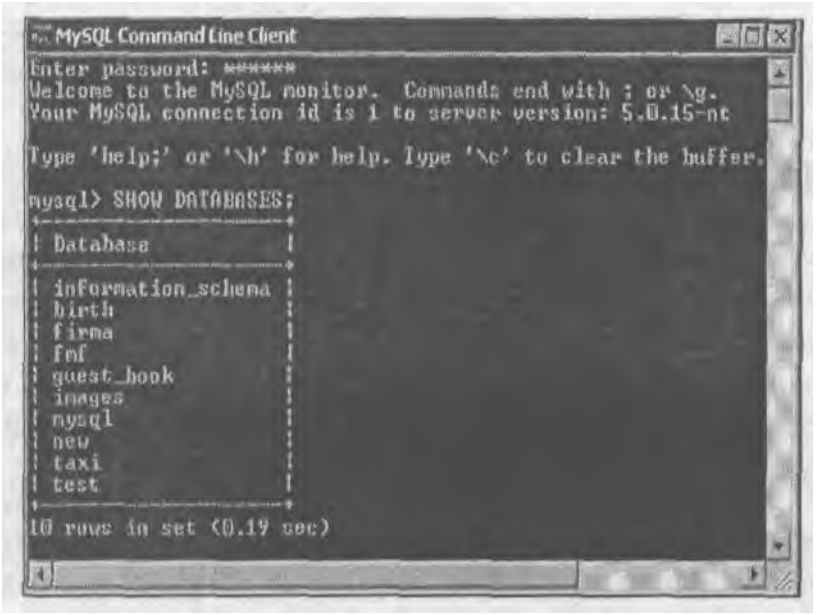

Рис. 4. Окно утилиты mysql

Для mysql признаком завершения команды является точка с запятой, а не конец строки. Команды собираются, но не исполняются до тех пор, пока программа не обнаружит точку с запятой. Вот пример команды, требующей распечатать список баз данных сервера и занимающей несколько строк:

```
mysql>    SHOW
->    DATABASES;
```

Обратите внимание на то, как изменилась метка командной строки (с mysql > на –>) после ввода первой строки этого запроса. Таким образом mysql показывает, что завершенного выражения она пока что не получила и ожидает его полного ввода. Эта

метка очень полезна, т. к. предоставляет весьма ценную информацию о состоянии программы. С ее помощью всегда можно узнать, чего ждет mysql.

В табл. 1 приведены все возможные варианты вида метки командной строки и соответствующие им состояния mysql.

Таблица 1. *Метка командной строки MySQL*

Метка	Значение
mysql >	Ожидание новой команды
->	Ожидание следующей строки многострочной команды
' >	Ожидание следующей строки, сбор строкового выражения, начинающегося с одиночной кавычки
" >	Ожидание следующей строки, сбор строкового выражения, начинающегося с двойной кавычки

2.3. Типы данных MySQL

Перед созданием базы данных надо подумать о том, **какие** типы данных будут использованы.

2.3.1. Строковые типы

В MySQL определены следующие строковые типы (табл. 2).

Таблица 2. *Тип столбца и требуемый объем памяти*

Тип столбца	Необходимый объем памяти
CHAR(M)	M байтов, 0≤M≤255
VARCHAR(M)	L+1 байтов, где L≤M и 0≤M≤65 535
BINARY (M)	M байтов, 0≤M≤255
VARBINARY(W)	L+1 байтов, где L≤M и 0≤M≤255
TINYBLOB, TINYTEXT	L+1 байтов, где $L<2^8$

BLOB, TEXT	L+2 байтов, где L < 2^{16}
MEDIUMBLOB, MEDIUMTEXT	L +3 байтов, где L < 2^{24}
LONGBLOB, LONGTEXT	L +4 байтов, где L < 2^{32}
ENUM(*'value1 ' , 'value2 ',...)*	1 или 2 байта в зависимости от числа значений (65 535 значений максимум)
SET(*'value1 ' , 'value2* ',...)	1 , 2, 3, 4 или 8 байтов в зависимости от числа значений (64 значения максимум)

Типы CHAR (строки фиксированной длины) и VARCHAR (строки произвольной длины) – это типы, предназначенные для записи коротких фрагментов текста. Длину каждого из них можно регулировать числом M. Записи в столбцы типа CHAR будут дополняться пробелами до максимальной длины. При этом длина строки не зависит от размеров данных, в то время как в столбцах с типом VARCHAR размер поля зависит от размеров данных. MySQL удаляет пробелы в конце текстовых строк у char во время извлечения и у VARCHAR во время сохранения.

Тип VARCHAR позволяет экономить память, но при хранении данных приходится указывать, сколько памяти занимает введенное значение. Это значение сервер использует для того, чтобы определить, с какой позиции начнется запись следующего значения. При записи данных типа CHAR серверу не приходится указывать в каждом поле реальную длину строки, а при извлечении данных считывать это значение для того, чтобы определить, где начнется следующее значение поля, поэтому сервер работает с такими строками быстрее, зато используется несколько больший объем памяти.

Если вы при определении таблицы зададите тип данных в поле фамилии водителя как CHAR (10), а потом попробуете ввести более длинную фамилию, то вводимый текст будет усечен до указанной в определении таблицы длины.

Тип BLOB (Binary Large Object, большой двоичный объект) используется для хранения двоичных данных.

Данные типа ENUM могут принимать одно из нескольких заранее заданных значений: ENUM (*value1, value2,*...). Например, предусмотрено, что в некотором столбце должен храниться ответ на вопрос, а допустимыми значениями ответа являются "да" и "нет". Тогда *value1* будет "да", а *value2* – "нет".

Тип SET предусматривает, что в поле одновременно может содержаться несколько из заранее заданного списка значений.

2.3.2. Форматы записи даты и времени

Дату и время можно хранить в одном из следующих форматов (табл.3).

Таблица 3. *Типы значений даты и времени*

Тип столбца	Описание	Диапазон значений
DATETIME	YYYY-MM-DDHH : MM : SS	'1000-01-01 00:00:00' — '9999-12-31 23:59:59'
DATE	YYYY-MM-DD	'1000-01-01' — '9999-12-31'
TIMESTAMP	YYYY-MM-DD HH:MM:SS	'1970-01-01 00:00:00' — '2037-12-31 23:59:59'
TIME	HH:MM:SS	'-838:59:59' — '838:59:59'
YEAR	YY ИЛИ YYYY	1970-2069 или 1901-2155

Отсчет времени в MySQL осуществляется с начала эпохи UNIX – 1 января 1970г.

2.3.3. Хранение числовых значений

Целые числа могут храниться со знаком (signed) и без (unsigned). Допустимые форматы целых чисел представлены в табл. 4.

Таблица 4. *Целые числа*

Тип	Память в байтах	Минимальное значение		Максимальное значение	
		Signed	Unsigned	Signed	Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32 768	0	32767	65535
MEDIUMINT	3	-8 388 608	0	8 388 607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-2^{63}	0	$2^{63}-1$	$2^{64}-1$

Числа с плавающей точкой могут храниться в столбцах следующих типов (табл. 5).

Таблица. 5. *Числа с плавающей точкой*

Тип	Память в байтах
FLOAT (p)	4, если 0≤p≤24; 8, если 25≤p≤53
FLOAT	4
DOUBLE [размерность]	8
DECIMAL *(M, D)* , NUMERIC *(M, D)*	Переменная

Данные типа FLOAT или DOUBLE округляются при записи до указанной в определении столбца точности.

Тип DECIMAL предназначен для хранения данных, которые не могут быть округлены, например, денежных величин. В MySQL версии 5.0.2 и более ранних данные этого типа хранятся в виде текста. Начиная с версии 5.0.3, данные типа DECIMAL хранятся в двоичном виде, в котором десять двоичных цифр записываются в 4-х байтах.

2.4. Создание базы данных

Теперь создадим базу данных описанной выше фирмы с использованием клиентской программы ***mysql***. Создадим базу данных taxi таксопарка командой CREATE DATABASE:

mysql> CREATE DATABASE taxi;

Команда должна завершаться точкой с запятой (этот символ называется разделителем – delimiter), получив команду, сервер выполняет ее и выдает сообщение (при успехе – Query OK, 1 row affected).

При создании базы данных она автоматически не выбирается, выбирать ее нужно отдельно. Для этого надо дать команду использовать только что созданную базу данных: mysql>USE taxi;

Создавать базу нужно только однажды, но выбирать ее приходится в каждом сеансе работы с mysqi.

Таблицы создаются командой CREATE TABLE.

При создании нужно указать не только имя таблицы, но и ее полное определение, состоящее из определений отдельных полей. Таблица cars должна содержать поля:

- model – с названием модели автомобиля;
- madein — года ее выпуска;
- reg_number – государственного регистрационного номера;
- color – цвета.

Кроме того, вам понадобится столбец id, однозначно идентифицирующий запись об автомобиле — первичный ключ. Значения этого поля не должны повторяться и не могут быть пустыми, желательно, чтобы эти значения были целыми числами, увеличивающимися на единицу в каждой новой записи.

```
mysql> CREATE TABLE cars (
model CHAR(50)    NOT NULL,
madein YEAR(4)    NOT NULL,
reg_number CHAR(12)    NOT NULL,
color CHAR(15)    NOT NULL,
id SMALLINT AUTO_INCREMENT,
PRIMARY    KEY(id) ) ;
```

Атрибут NOT NULL означает, что все строки таблицы должны иметь значение в этом столбце. Если NOT NULL не указано, поле может быть пустым (NULL).

PRIMARY KEY после имени столбца определяет, что этот столбец является первичным ключом для таблицы. Данные в этом столбце должны быть уникальными.

AUTO_INCREMENT – атрибут, используемый для создания столбца с уникальными значениями. Если при вставке строк в таблицу оставлять такое поле пустым, MySQL автоматически генерирует уникальное значение идентификатора. Это значение будет на единицу больше максимального значения, уже существующего в столбце. В каждой таблице может быть не больше одного такого поля.

MySQL использует следующий алгоритм для инициализации счетчика для столбца id, имеющего атрибут AUTO_INCREMENT: после запуска сервера MySQL при первом запросе на добавление данных в таблицу cars сервер высчитывает максимальное значение в столбце id. Полученное значение увеличивается на единицу, заносится в новую запись и в счетчик. Если таблица была пуста, то счетчик устанавливается в единицу.

Таблица drivers содержит столбцы для хранения имени, отчества и фамилии водителя – name, second_name, family_name; даты его рождения birth, домашнего адреса address и даты поступления на работу startdate.

```
mysql> CREATE TABLE drivers (
name CHAR(30) NOT NULL,
second_name CHAR(30),
```

```
family_name CHAR(30) NOT NULL,
birth DATE NOT NULL,
address CHAR(150) NOT NULL DEFAULT 'unkonwn',
startdate DATE NOT NULL DEFAULT '2002-01-01',
id SMALLINT AUTO_INCREMENT,
PRIMARY KEY(id) );
```

Кроме этого, при создании таблицы для некоторых полей могут применяться дополнительные ключевые слова, уточняющие диапазон возможных значений. Ключевое слово UNSIGNED может применяться после указания целочисленного типа и означает, что его значение может быть либо положительным, либо нулевым.

DEFAULT – после этого ключевого слова указывается значение по умолчанию для данного столбца. Например:

```
DEFAULT      'no description'
```

Таблица timetable состоит из полей use_on с датой, номером машины car_number и идентификатором водителя driver_number.

```
mysql> CREATE TABLE    timetable (
use_on DATE DEFAULT NULL,
car_number SMALLINT,
driver_number SMALLINT,
id    SMALLINT AUTO_INCREMENT,
PRIMARY KEY(id),
INDEX      (car_number),
INDEX      (driver_number),
FOREIGN KEY (car_number) REFERENCES cars (id),
FOREIGN KEY (driver_number) REFERENCES drivers (id) );
```

MySQL поддерживает ссылочную целостность базы с помощью ограничения внешнего ключа foreign key. Назначение этого ключа — проверять соответствие значений в столбцах родительского и внешнего ключей. Эти столбцы должны иметь одинаковый тип данных, и только те значения, которые встречаются в родительском ключе, могут использоваться во внешнем ключе. Родительский ключ должен содержать неповторяющиеся значения и не содержать значений NULL.

Последние два столбца таблицы timetable являются внешними ключами, и MySQL 5 требует, чтобы перед созданием внешних ключей на этих столбцах были созданы индексы.

Индекс формируется из значений одного или нескольких столбцов таблицы и позволяет находить нужную строку по заданному значению. Для ускорения запросов индексы обычно создаются на тех столбцах таблицы, которые часто используются в запросах.

После создания таблиц можно просмотреть их список и структуру. Посмотреть список таблиц в базе можно командой.

```
mysql> SHOW TABLES;
```

а структуру таблицы –

```
mysql> DESCRIBE table_name;
```

Здесь tabie_name – имя таблицы. Эта команда позволит посмотреть типы данных столбцов и дополнительные атрибуты, указанные при создании таблицы.

2.5. Самостоятельная работа №1

Для защиты лабораторной работы №2 необходимо выполнить самостоятельную работу №1 и ответить на контрольные вопросы.

1. Создайте учебную базу, используя утилиту mysql, в соответствии со схемой на рис.5 и атрибутами и типами данных, указанных под ней.

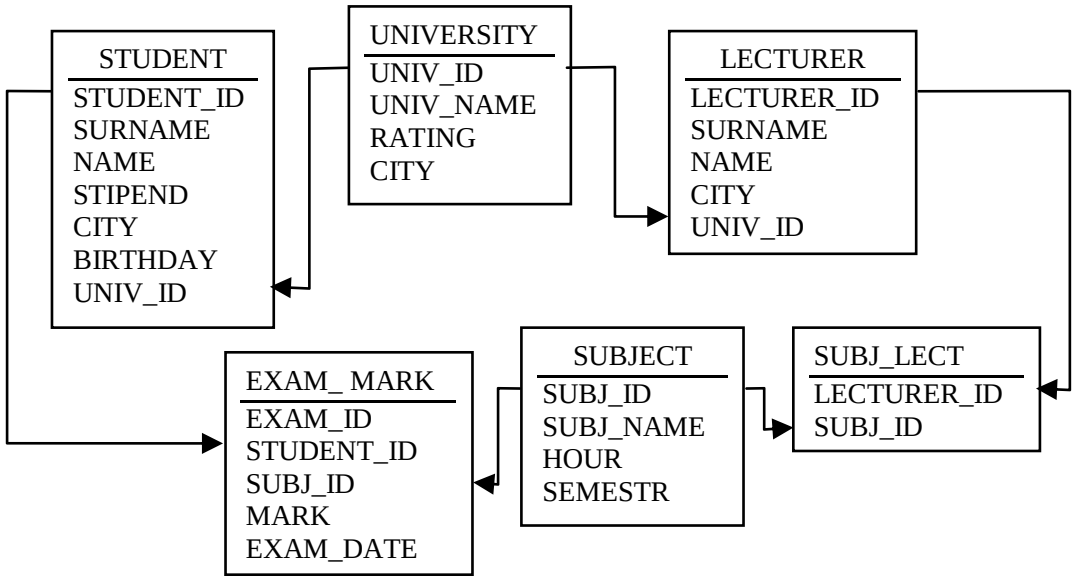

Рис.5. Схема учебной базы данных «Университет»

Структура учебной базы данных:

Сущность UNIVERSITY (Университет). Атрибуты:

UNIV_ID – идентификатор университета (тип данных SMALLINT – ключевое поле)

UNIV_NAME – название университета (тип данных CHAR(10))

RATING – рейтинг (тип данных SMALLINT)

CITY – город в котором расположен университет (тип данных CHAR(30))

Сущность SUBJECT (предмет обучения). Атрибуты:

SUBJ_ID – числовой код предмета обучения (тип данных SMALLINT – ключевое поле)

SUBJ_NAME – наименование предмета (тип данных CHAR(50))

HOUR – количество часов отводимых на изучение предмета (тип данных SMALLINT)

SEMESTR – семестр в котором изучается предмет (тип данных TINYINT)

Сущность STUDENT (Студент). Атрибуты:

STUDENT_ID – числовой код идентифицирующий студента (тип данных SMALLINT – ключевое поле)

SURNAME – Фамилия студента (тип данных CHAR(30))

NAME – Имя студента (тип данных CHAR(30))

STIPEND – размер получаемой стипендии (тип данных SMALLINT)

CITY – город, в котором живет студент (тип данных CHAR(30))

BIRTHDAY – день рождения студента (тип данных DATE)

UNIV_ID - числовой код идентифицирующий университет, в котором учится студент (тип данных SMALLINT – внешний ключ от сущности UNIVERSITY)

Сущность LECTURER (Преподаватель). Атрибуты:

LECTURER_ID – числовой код идентифицирующий преподавателя (тип данных SMALLINT – ключевое поле)

SURNAME – фамилия преподавателя (тип данных CHAR(30))

NAME – имя преподавателя (тип данных CHAR(30))

CITY – город, в котором живет преподаватель (тип данных CHAR(30))

UNIV_ID – числовой код идентифицирующий университет в котором работает преподаватель (тип данных SMALLINT – внешний ключ от сущности UNIVERSITY)

Сущность EXAM_ MARK (Экзаменационные оценки). Атрибуты:

EXAM_ID – идентификатор экзамена (тип данных SMALLINT – ключевое поле)

STUDENT_ID – идентификатор студента (тип данных SMALLINT – внешний ключ от сущности STUDENT)

SUBJ_ID – идентификатор предмета (тип данных SMALLINT – внешний ключ от сущности SUBJECT)

MARK– экзаменационная оценка (тип данных TINYINT)

EXAM_DATE – дата экзамена (тип данных DATE)

Сущность SUBJ_LECT (Преподаватель предмета). Атрибуты:

LECTURER_ID – числовой код идентифицирующий преподавателя (тип данных SMALLINT – внешний ключ от сущности LECTURER)

SUBJ_ID – идентификатор предмета (тип данных SMALLINT – внешний ключ от сущности SUBJECT)

Тема 3 Реализация и администрирование БД.

3.1. Клиентские утилиты СУБД MySQL 5

Интерфейс MySQL **Command line client**, с которым вы работали в предыдущем разделе, удобен, но не позволяет использовать функциональность утилиты mysql полностью. Кроме того, в состав MySQL входит еще ряд полезных утилит, с одной из которых, mysqldump, необходимо познакомиться даже начинающим.

3.1.1. Утилита командной строки *mysql*

Клиентская утилита ***mysql*** запускается в интерактивном режиме в DOS-окне, когда вы выбираете в меню пункт **Command line client**. До MySQL 5 в комплекте поставки сервера не было такой возможности запуска mysql, и приходилось самим вызывать окно для работы с командной строкой. Как всегда в Windows, за удобство работы с готовым интерфейсом приходится платить непониманием сущности происходящего, при этом еще и лишаясь некоторых возможностей, благополучно использовавшихся при работе со старым интерфейсом. Поэтому, налюбовавшись на сервис **Command Line Client** из MySQL 5, вернемся в юность информационных технологий и откроем DOS-окно для ввода команд. В Windows это делается так: щелкаем по кнопке **Пуск,** выбираем пункт меню **Выполнить** и в появившемся окне **Запуск программы** набираем cmd. Появляется черное DOS-окно со строкой, содержащей имя текущего каталога, – это приглашение ввести команду. Курсор мигает в конце этой строки.

Установленный сервер mysqld-nt.exe, клиентские утилиты mysql, mysqidump и mysqlimport находятся в том каталоге, в который вы установили сервер, а именно в C:\Program Files\MySQL\MySQL Server 5.0\bin. Поскольку при установке MySQL вы добавили путь до этого каталога в переменную PATH, то смело можете запускать утилиту сразу после открытия DOS-окна.

Утилиты предназначены для работы с сервером и при запуске потребуют указать имя пользователя, под которым вы хотите подключиться к серверу, и пароль. При конфигурировании сервера MySQL вы указали пароль для суперпользователя root, теперь снова им пора воспользоваться. Обычно для получения доступа к серверу MySQL необходимо сообщить клиентской программе и хост, с которым вы хотите соединиться. Например, клиент mysql можно запустить следующим образом:

>mysql -h *host_name* -u *user_name* -p*your_pas*

Здесь ключ -h указывает на то, что дальше последует имя хоста, к которому вы хотите подключиться. По умолчанию подключение осуществляется к серверу localhost. Ключ -u

означает, что за ним следует имя пользователя, от имени которого осуществляется соединение, а после ключа -p указывается пароль. Обратите внимание, что пробела перед паролем нет.

Формат вызова утилиты mysql в нашем случае будет таким:

> mysql -u root -p

После ключа -p не указан пароль, мы знаем, что он нужен, но не хотим давать его в командной строке, а дождемся приглашения и введем так, что его у нас из-за спины никто не прочитает. Приглашение появится, если сервер MySQL запущен, а если сервер не доступен, будет выведено сообщение о невозможности к нему подключиться. Если ключи набраны неверно, например, пропущены пробелы, вы получите справку по утилите. После ввода пароля вы должны увидеть приветствие и приглашение командной строки уже самой утилиты ***mysql*** (рис. 6).

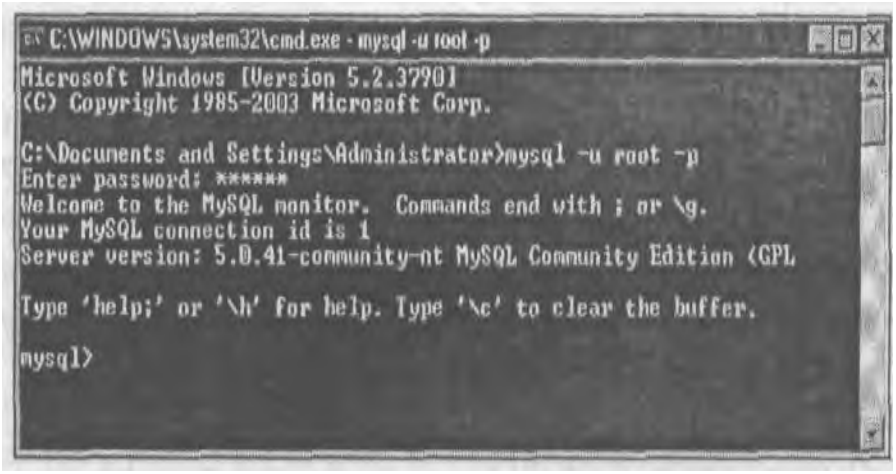

Рис. 6. Запуск утилиты *mysql*

Похоже на **MySQL Command Line Client?** Что и требовалось доказать.

Получив приглашение утилиты mysql, можно давать команды MySQL для создания баз и таблиц. Завершение сеанса работы происходит по команде:

mysql>quit

После этого в DOS-окне снова появится приглашение операционной системы. Учтите, что и при работе с mysql, и с операционной системой существует возможность повтора ранее данных команд. Просмотреть историю команд можно щелчками по кнопкам со стрелками вверх и вниз. По щелчку после приглашения командной строки появляются команды, запустить их можно, нажав клавишу <Enter>.

Утилита mysql может работать в двух режимах: интерактивном, который вы только что инициировали, и пакетном.

Для работы с утилитой в пакетном режиме следует написать сценарий, состоящий из последовательности команд SQL, которые предстоит исполнить для создания базы и таблиц, входящих в базу. Сценарий следует сохранить в тот же каталог bin сервера MySQL, из которого вы запускали утилиту mysql. Принято хранить сценарии в файлах с расширением sql. Кроме уже рассмотренных команд в скрипте надо предусмотреть возможность того, что в сценарии создания таблицы возможны ошибки. При исправлении ошибок в сценарии вы запускаете его несколько раз, но уже при повторном запуске получаете сообщение о том, что нельзя создать уже существующую таблицу. Значит, надо предварять создание таблицы командой

```
mysql> DROP TABLE IF EXISTS cars;
```

и удалить таблицу, если она уже существует.

Тогда сценарий создания и заполнения таблицы cars будет выглядеть так:

```
USE taxi;
DROP TABLE IF EXISTS cars;
CREATE TABLE    cars (
model char(50) NOT NULL,
madein year(4) NOT NULL,
reg_number char (12) NOT NULL,
color char(15) NOT NULL,
id smallint AUTO_INCREMENT PRIMARY KEY
) ;

INSERT INTO cars (model, madein, reg_number, color) VALUES
('Volga','2002','A788YC23','yellow'),
('Volkswagen', '2003' , 'A789YC23', 'red'),
('Renault','2005','A790YC23','white');
```

Скрипт создается в любом текстовом редакторе в кодировке Windows-1251. В этой кодировке данные и записываются. И хотя в окне утилиты ***mysql*** русские буквы будут нечитаемыми, но скрипты (например PHP) передадут символы правильно.

При использовании в пакетном режиме сценарии с запросами для создания базы надо копировать в каталог bin и передавать их на вход утилите mysql при запуске:

>mysql -u root -p < script.sql

Здесь script.sql — это сценарий, содержащий команды SQL для создания таблиц, записи в них данных и т. п.

3.1.2. Утилита *mysqldump*

Утилита mysqldump позволяет получить дамп ("моментальный снимок") содержимого базы данных или совокупности баз для создания резервной копии или пересылки данных на другой SQL-сервер баз данных. Дамп будет содержать набор команд SQL для создания и заполнения таблиц. Откройте окно. Если вы из командной строки дадите такую команду:

> mysqldump *имя_базы*

то распечатаете весь этот набор команд. Чтобы сохранить дамп в файле, дайте команду перенаправить вывод не на экран, а в файл backup-file.sql:

>mysqldump *имя_базы* > backup-file.sql

Если не указывать имена таблиц или использовать параметр --databases или --all-databases, то будет получен дамп базы данных в целом (всех баз данных сервера).

Перенеся файл дампа на другой сервер MySQL, можно этот файл передать на вход утилиты MySQL для создания таблиц в базах данных.

3.1.3. Утилита *mysqlimport*

Утилита mysqlimport импортирует текстовые записи в таблицы. На основании имени файла утилита определяет имя таблицы, в которую импортируются данные.

> mysqlimport *параметры имя_базы_данных имя_текстового_файла*

Для каждого текстового файла, указанного в командной строке, mysqlimport удаляет расширение в имени файла и использует его, чтобы определить, в какую таблицу занести содержимое. Например, файлы с именами 'patient.txt', 'patient. text' и 'patient' должны быть все занесены в таблицу с именем 'patient'.

Например, для того чтобы импортировать данные из файла new.txt (одноименного с таблицей базы) в базу данных our_base, надо дать следующую команду, а затем ввести пароль:

```
>  mysqlimport -u root -p -local our_base new.txt
password:******
```

Каждая строка в файле образует запись в таблице и должна содержать значения полей, разделенные запятыми. Конец записи распознается по символу конца строки текстового файла.

3.2. Запись, обновление и удаление данных в таблицах

3.2.1. Запись данных в таблицы

После создания базы данных необходимо внести данные в таблицы. Для внесения данных в базу данных можно использовать оператор INSERT. Самый простой вариант – внесение данных во все поля таблицы:

```
mysql> INSERT INTO cars VALUES
('Volga', '2003', 'A786YC23', 'white', '','1');
```

Аналогично можно добавлять данные только в избранные поля, при этом надо указать имена полей явно:

```
mysql> INSERT INTO cars (model, madein, reg_number, color) VALUES
-> ('Volga', '2002 ' , 'A788YC23', 'yellow');
mysql>INSERT INTO cars (model, madein, reg_number, color) VALUES
-> ('Volkswagen','2003','A789YC23','red');
```

```
mysql>INSERT INTO cars (model, madein, reg_number, color) VALUES
-> ('Renault', '2005' , 'A790YC23', 'white');
```

Добавим еще несколько значений в созданные ранее таблицы:

```
mysql> INSERT INTO drivers VALUES (
-> 'Stanislav', 'Petrovitch', 'Nikolaev',
-> '1970-01-22' , 'Kurortny prospect 11, flat 54', '2004-02-30', 1) ;
mysql> INSERT INTO drivers (name, second_name, family_name,
-> birth, address, startdate)
-> VALUES ('Fedor', 'Sergeevitch', 'Sentsov',
-> '1960-11-02', 'Donskaya 92, flat 112', '2003-12-30');
mysql> INSERT INTO timetable VALUES ('2008-08-09', 3,1,1),
-> ('2008-07-21', 2,2,2), ('2008-07-20', 2,2,3);
```

Последняя запись наверняка делается диспетчером автопарка каждый день с утра. Чтобы упростить его работу, можно использовать встроенную в MySQL функцию curdate(), которая возвращает текущую дату как раз в нужном формате:

```
mysql> INSERT INTO timetable VALUES (curdate(), 2,2,4);
```

Если вы попытаетесь оставить пустыми поля, отмеченные NOT NULL и не имеющие значения по умолчанию, то сервер MySQL выдаст сообщение об ошибке и не выполнит команду.

Кроме того, сервер MySQL 5 проверяет связи между таблицами при внесении записей. То есть вам не удастся внести в поле, являющееся внешним ключом, значение, отсутствующее в связанной таблице.

Утилита mysql может создавать таблицы и заполнять их данными и в пакетном режиме. Для этого необходимо создать текстовый сценарий с командами SQL, обычно такой сценарий сохраняют в файле с расширением sql. Затем необходимо запустить утилиту mysql в пакетном режиме и передать ей имя сценария. Рассмотрим эту процедуру подробнее в следующем разделе.

3.2.2. Обновление и удаление данных в таблицах

Если необходимо изменить значение поля, то следует использовать команду UPDATE. В команде надо указать имя используемой таблицы и предложение SET, определяющее изменение, которое нужно сделать для некоторого столбца. Исправим дату выпуска автомобиля "Renault":

```
mysql>UPDATE cars SET madein='2004' WHERE model='Renault';
```

Можно изменить значение всех полей столбца, например, при переоценке товаров. Увеличим цены на все товары на 10% с учетом инфляции:

```
mysql>UPDATE goods SET price = price * 1.1;
```

Удалить целые строки из таблицы можно с помощью команды DELETE. Удалим некоторые строки из таблицы cars в связи с продажей старых машин, изготовленных до 2006 г.:

```
mysql>DELETE FROM cars where madein<=2005;
```

Можно удалить все записи из таблицы:

```
mysql>DELETE FROM cars;
```

При этом счетчик автомобилей в столбце id не сбрасывается, и при вставке новых данных MySQL автоматически присвоит новой записи следующий после последнего удаленного номер.

Для того чтобы очистить таблицу и сбросить счетчик, можно применить команду:

```
mysql>TRUNCATE cars;
```

Удалить таблицу можно командой DROP TABLE:

```
mysql>DROP TABLE timetable;
```

При этом надо иметь права на удаление, а также учитывать, что удалить таблицу, имеющую первичный ключ, по которому осуществляется связь с другой таблицей, MySQL не позволит, пока не удалена таблица с соответствующим внешним ключом. Удалить можно и всю базу целиком:

```
mysql>DROP DATABASE taxi;
```

3.3. Работа со структурой таблицы. Индексы

3.3.1. Изменение структуры таблицы

Оператор ALTER TABLE используется для изменения структуры таблицы. Преобразования, осуществляемые с помощью этого оператора, представлены в табл. 6.

Таблица 6. *Преобразования таблицы с помощью оператора ALTER TABLE*

Синтаксис	Описание
ADD [COLUMN] column_description [FIRST | AFTER column]	Добавить новый столбец в указанное место
ADD INDEX *[index] (column,* ...)	Добавить индекс в указанный столбец (столбцы) таблицы
ADD PRIMARY KEY (column,…)	Сделать указанный столбец (столбцы) первичным ключом таблицы
ALTER [COLUMN] column {SET DEFAULT value | DROP DEFAULT}	Добавить или удалить значение по умолчанию определенного

	столбца
CHANGE [COLUMN] column new_column_description	Изменить столбец с именем *column* так, чтобы он получил указанное описание. Это можно использовать для изменения имени столбца, поскольку *column_description* включает в себя имя
MODIFY [COLUMN] column_description	Похоже на CHANGE. Используется для изменения типов столбцов, но не имен
DROP [COLUMN] column	Удалить указанный столбец
DROP PRIMARY KEY	Удалить первичный индекс (не столбец!)
DROP INDEX index	Удалить указанный индекс
RENAME table tbl to *new_tbl*	Переименовать таблицу

Допустим, вы решили добавить отсканированные фотографии водителей в базу, чтобы легче было печатать пропуска или другие документы. Ссылки на файлы с фотографиями можно поместить в новый столбец, добавив его в таблицу командой alter table:

```
mysql> ALTER TABLE drivers ADD COLUMN photo CHAR(50);
```

В данном примере столбец добавляется в конец таблицы, но можно было бы добавить ключевое слово FIRST, тогда столбец стал бы первым, а можно указать after birth, поместив его тем самым после указанного столбца.

Допустим, вы обнаружили, что поле для указания адреса оказалось мало для вводимых данных: последние символы обрезаются, т. к. адрес длиннее 150 выделенных для него символов. Изменим определение столбца:

```
mysql> ALTER TABLE drivers MODIFY address CHAR(200);
```

3.3.2. Создание индексов

Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному значению путем последовательного просмотра таблицы строка за строкой может занимать много времени. Индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и, таким образом, позволяет находить нужную строку по заданному значению.

Для оптимальной производительности запросов индексы обычно создаются на тех столбцах таблицы, которые часто используются в запросах. Для одной таблицы может быть создано несколько индексов.

Когда столбец помечается как ключ, создается индекс. Индекс хранит список значений ключа и указатели на записи, содержащие эти значения. При создании индекса таблица уже должна содержать столбцы, по которым создается индекс.

В базе таксопарка в таблице cars первичным ключом является учетный номер машины. Запросы же к базе могут строиться на поиске по государственному регистрационному номеру. Для того чтобы повысить скорость выполнения таких запросов, следует создать индекс по столбцу reg_number, в котором записан регистрационный номер:

```
mysql>CREATE INDEX nomer ON cars (reg_number) ;
```

При создании индекса на сервере формируется упорядоченный список значений этого поля, который MySQL использует при выполнении запросов. Если для создания индекса используется несколько полей, то значения второго поля упорядочиваются внутри первого, третьего – внутри второго и т. д.

Имя индекса должно быть уникальным в базе данных. После создания индекса MySQL использует индекс для поиска данных автоматически, т. е. в запросах не требуется указывать индекс каким-либо образом.

3.4. Самостоятельная работа №2

Для защиты лабораторной работы №3 необходимо выполнить самостоятельную работу №2 и ответить на контрольные вопросы.

1. Используя утилиту mysql, заполните таблицы учебной базы в соответствии с таблицами 7-12.

Таблица 7. *Сущность UNIVERSITY*

UNIV_ID	UNIV_NAME	RATING	CITY
22	МГУ	606	Москва
10	ВГУ	296	Воронеж
11	НГУ	345	Новосибирск
32	РГУ	416	Ростов
14	БГУ	326	Белгород
15	ТГУ	368	Томск
18	ВГМА	327	Воронеж

Таблица 8. *Сущность SUBJECT*

SUBJ_ID	SUBJ_NAME	HOUR	SEMESTR
10	Информатика	56	1
22	Физика	34	1
43	Математика	56	2
56	История	34	4
94	Английский	56	3
73	Физика	34	5

Таблица 9. *Сущность STUDENT*

STUDENT_ID	SURNAME	NAME	STI-PEND	CITY	BIRTHDAY	UNIV_ID
1	Иванов	Иван	600	Орел	3.12.1989	11
3	Петров	Петр	900	Курск	1.12.1987	10
6	Сидоров	Вадим	600	Москва	7.06.1986	22
10	Кузнецов	Борис	0	Брянск	8.12..1988	32
12	Зайцева	Ольга	1000	Липецк	1.05.1988	11
265	Павлов	Андрей	0	Воронеж	5.11.1986	18

32	Котов	Павел	600	Белгород	NULL	14
654	Лукин	Артем	900	Воронеж	1.12.1997	10
274	Петров	Антон	900	NULL	5.08.1986	22
55	Белкин	Вадим	1000	Воронеж	7.01.1986	15
8	Пупкин	Василий	600	Ростов	17.10.1990	32
111	Косинцева	Рима	600	Курск	27.11.1986	14
222	Фокин	Федор	1000	Томск	5.05.1988	15
543	Аверин	Авакум	0	Тамбов	11.03.1987	18

Таблица 10. *Сущность LECTURER*

LECTURER_ID	SURNAME	NAME	CITY	UNIV_ID
24	Колесников	Борис	Воронеж	10
46	Никонов	Иван	Воронеж	10
74	Лагутин	Павел	Москва	22
108	Струков	Николай	Москва	22
276	Николаев	Виктор	Новосибирск	11
328	Сорокин	Андрей	Орел	11
17	Воронин	Валентин	Ростов	32
5	Доронина	Дарья	Ростов	32
33	Ёлкин	Евгений	Белгород	14
38	Жижикин	Жорес	Тула	14
57	Злова	Зинаида	Томск	15
101	Ильин	Илья	Томск	15
213	Юрков	Юрий	Воронеж	18
273	Яснова	Яна	Россошь	18

Таблица 11. *Сущность EXAM_ MARK*

EXAM_ID	STUDENT_ID	SUBJ_ID	MARK	EXAM_DATE
145	3	10	5	12.01.07
34	265	10	4	23.01.07
75	8	10	5	05.01.07
238	8	22	3	22.06.06
639	10	22	NULL	22.06.06
43	654	22	4	18.01.07
4	274	43	4	19.01.07
3	111	43	3	15.01.07

53	543	43	3	18.01.07
48	6	56	5	18.06.06
63	32	56	4	10.01.07
79	111	56	5	09.01.07
412	12	94	4	06.01.07
436	222	94	NULL	18.06.06
123	55	94	4	18.06.06
120	1	73	5	21.01.07
208	55	73	4	20.01.07
175	12	73	4	21.01.07

Таблица 12. *Сущность SUBJ_LECT*

LECTURER_ID	SUBJ_ID
24	10
46	22
74	43
108	56
276	94
328	73
17	10
5	22
33	43
38	56
57	94
101	73
213	10
273	43

3.5. Контрольные вопросы по ведению базы данных

1. Что такое «приглашением командной строки»?
2. Что в MySQL является признаком завершения команды?
3. О чем говорит метка командной строки «’>» (апостроф-больше)?
4. Какие строковые типы данных определены MySQL?
5. Какие типы данных записи даты и времени определены MySQL?
6. Какие числовые типы данных определены MySQL?
7. Какая команда используется в MySQL для создания базы данных? Приведите структуру команды создания таблицы.
8. Какие команды и как задают в таблице первичный и внешний ключи?
9. Каким образом в MySQL можно посмотреть существующие базы данных, таблицы и структуру таблиц?
10. Перечислите утилиты MySQL и их назначения.
11. Какова структура записи данных в таблицы в MySQL?
12. Какая команда и как меняет структуру таблиц в MySQL?

Приложение 2. Задание к практическим работам

Тема 1 Вводные понятия

1.1. Назначение и виды запросов

Запросы являются одним из основных инструментов выборки, обновления и обработки данных в таблицах базы данных. Запрос позволяет сформировать пользовательское представление о данных, не обязательно отвечающее требованиям нормализации. Результат выполнения запроса — это новая, чаще всего временная, таблица, которая существует до закрытия запроса. Структура такой таблицы определяется выбранными из одной или нескольких таблиц полями. Записи формируются путем объединения записей таблиц, участвующих в запросе. Способ объединения записей различных таблиц указывается при определении их связи. Условия отбора, сформулированные в запросе, позволяют фильтровать записи, составляющие результат объединения таблиц.

Запрос строится на основе одной или нескольких взаимосвязанных таблиц, позволяя объединять данные, содержащиеся в них. При этом могут использоваться таблицы базы данных, а также сохраненные таблицы, полученные в результате выполнения других запросов. Кроме того, запрос может строиться непосредственно на другом запросе с использованием его временной таблицы с результатами.

Запрос позволяет выбрать необходимые данные из одной или нескольких взаимосвязанных таблиц, произвести вычисления и получить результат в виде виртуальной таблицы. Через запрос можно производить обновление данных в таблицах, добавление и удаление записей

Запросы наиболее часто отправляются с помощью команды SELECT, которая дает инструкцию базе данных извлечь информацию из одной или нескольких таблиц. Для выполнения запроса вначале необходимо соединиться с сервером MySQL и выбрать базу данных.

Для этого запустим клиентскую программу **MySQL Command Line Client**, а затем дадим команду выбора базы данных:

```
mysql>USE taxi
```

Запросим данные о модели и цвете автомобилей автопарка:

```
mysql>SELECT model, color FROM cars;
```

Результат представлен на рис. 7.

SELECT – ключевое слово, которое сообщает базе данных, что эта команда – запрос. Все запросы начинаются этим словом, за которым следует пробел.

model, color – список столбцов, которые выбираются из таблицы и помещаются в результирующую выборку в памяти.

FROM – ключевое слово, которое должно быть в каждом запросе. За ним должен идти пробел, а затем имя таблицы, используемой в качестве источника информации.



Рис. 7. Запрос к базе с помощью команды select

Если данных в таблице немного, то можно запросить их все:

```
mysql>SELECT * FROM table_name;
```

Здесь символ звездочки означает, что следует выбрать значения из всех столбцов. Но чаще запрашивают значения отдельных полей, поскольку обработка такого запроса требует меньших ресурсов.

Выясним, какие водители работали за то время, пока ведется учет с помощью базы. Нам нужен только список водителей из таблицы timetable, без повторов.

DISTINCT – параметр в запросе SELECT, дающий возможность исключить повторяющиеся данные из результатов запроса.

```
mysql>SELECT DISTINCT driver_number FROM timetable;
```

Результат запроса будет таким:

```
+ ------------------+
| driver_number |
+ ------------------+
|          1          |
|          2          |
+ ------------------+
```

Результат получился действительно без повторений, но не слишком понятный — указаны номера, а не фамилии водителей. Но фамилии находятся в другой таблице. Как выбрать данные из двух и более таблиц, вы узнаете, рассмотрев способы объединения таблиц.

2.2. Использование запросов с условием

2.2.1. Запросы с указанием критерия отбора данных

Таблицы бывают очень большими, а нас могут интересовать только определенные строки, SQL дает возможность устанавливать критерии для отбора нужных строк. Предложение WHERE позволяет определить критерий отбора строк. Операции, допустимые в предложении WHERE, даны в табл. 13.

Таблица 13. *Знаки операции сравнения для конструкции WHERE*

Оператор	Название	Описание
=	Равенство	Проверяет, являются ли два значения равными
>	Больше	Проверяет, больше ли одно значение

		другого
<	Меньше	Проверяет, меньше ли одно значение другого
>=	Больше или равно	Проверяет, больше или равно одно значение по отношению к другому
< =	Меньше или равно	Проверяет, меньше или равно одно значение по отношению к другому
!= ИЛИ <>	Не равно	Проверяет, не равны ли два значения
IS NOT NULL		Проверяет, имеет ли поле значение
IS NULL		Проверяет, не имеет ли поле значение
BETWEEN	Величина между	Проверяет, значение больше или равно минимальному и меньше или равно максимальному
IN	Значение содержится	Проверяет, содержится ли значение в определенном множестве
NOT IN	Значение не содержится	Проверяет, не содержится ли значение в определенном множестве
LIKE	Соответствие	Проверяет, отвечает ли значение образцу, используя простые механизмы соответствия SQL
NOT LIKE	Соответствие	Проверяет, не соответствует ли значение образцу
REGEXP	Регулярное выражение	Проверяет, соответствует ли значение регулярному выражению

Узнаем цвет всех автомобилей "Волга":

```
SELECT color FROM cars WHERE model='Volga';
```

Допустим, нам надо узнать, когда выходил на работу водитель с учетным номером 1:

```
mysql>SELECT use_on FROM timetable WHERE driver_number = 1;
```

Определим номера всех автомобилей "Волга" желтого цвета:

```
mysql>    SELECT reg_number FROM cars
-> WHERE model    = 'Volga' AND color = 'yellow';
```

Можно проверить несколько критериев сразу, объединяя их операциями AND или OR:

```
mysql> SELECT birth FROM drivers
-> WHERE name='Fedor' AND family_name='Sentsov';
```

Оператор IN определяет множество значений, которому данное значение может принадлежать или не принадлежать. Чтобы найти номерные знаки всех автомобилей "Волга" и "Volkswagen", дайте такую команду:

```
mysql> SELECT reg_number FROM cars WHERE model
-> IN ('Volga', 'Volkswagen');
```

Оператор BETWEEN задает границы, в которые должно попасть значение, чтобы условие запроса выполнялось. Вы должны ввести ключевое слово between с начальным значением, ключевое слово AND и конечное значение. Выясним, кто из водителей устроился на работу в период с 1 января 2001 г. до 1 января 2004 г.:

```
mysql> SELECT family_name FROM drivers
-> WHERE startdate BETWEEN '2001-01-01' AND '2004-01-01';
```

Оператор LIKE использует простой механизм соответствия SQL. Образец может состоять из обычного текста и знака процента (%) для указания совпадения с любым количеством символов. В MySQL соответствия не чувствительны к регистру. Например, шаблон ' Fed%' соответствует любой строке, которая начинается с ' Fed'.

```
mysql>SELECT * FROM drivers WHERE name LIKE 'Fed%';
```

Ключевое слово REGEXP служит для указания регулярных выражений. MySQL использует регулярные выражения в стиле POSIX.

Найдем данные об автомобиле, в номере которого встречается число 788:

```
mysql>SELECT * FROM cars where reg_number='[A-Z]+788[A-Z0-9]+';
```

2.2.2. Группировка данных и агрегатные функции

Группировка и агрегирование данных совершаются с помощью функций, определенных в MySQL. Описание некоторых функций дано в табл.14.

Таблица 14. *Функции MySQL*

Название	Описание
AVG (столбец)	Средняя величина значений в определенном
COUNT (элементы)	При указании столбца выдается число ненулевых значений в этом столбце. Если перед именем столбца указать DISTINCT, то выдается только количество неодинаковых значений
MIN (столбец)	Минимальное значение в столбце
MAX (столбец)	Максимальное значение в столбце
SUM (столбец)	Сумма значений в столбце

Агрегатные функции указываются подобно именам полей в запросе SELECT, но сами они используют имена поля в качестве аргументов. SUM и AVG могут обрабатывать только числовые поля. С функциями COUNT, MAX и MIN могут использоваться и числовые, и символьные поля. Например, чтобы определить количество водителей в автопарке, можно запросить

```
mysql>SELECT COUNT(id) FROM drivers;
```

Если надо узнать, сколько водителей работало за отчетное время, то подойдет команда

```
mysql>SELECT COUNT (DISTINCT driver_number) FROM timetable;
```

Функция count (*) посчитает число строк в таблице, но если указать count (field) — посчитать значения в поле *field*, в котором есть значения NULL, то пустые поля учтены не будут.

Предложение GROUP BY позволяет определять подмножество значений в особом поле в терминах другого поля и применять функцию агрегата к подмножеству. Это дает вам возможность объединять поля и агрегатные функции в едином предложении SELECT.

Посчитаем, сколько автомобилей есть в таксопарке, сгруппировав данные по моделям:

```
mysql>SELECT model, COUNT(model) FROM cars GROUP BY model;
```

Каждая группа состоит из строк с одинаковым значением поля model. Результат запроса представлен на рис. 8.

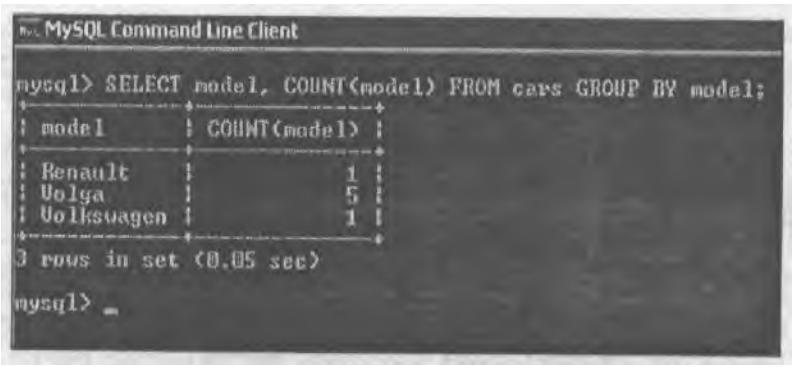

Рис. 8. Упорядочение результатов запроса

Таблицы – это неупорядоченные наборы данных, и выборка из них не обязательно появляется в какой-то определенной последовательности. Для извлечения данных в определенном порядке используется конструкция ORDER BY.

Упорядочим сведения о водителях по дате рождения:

```
mysql>SELECT * FROM drivers ORDER BY birth;
```

В полученной выборке список откроют старшие водители. Если требуется упорядочить данные по убыванию, следует использовать ключевое слово DESC (от англ, *descending* — по убыванию):

```
mysql>SELECT * FROM drivers ORDER BY birth DESC;
```

Конструкция LIMIT служит для указания, сколько строк результата следует отображать. Если надо найти двух самых пожилых водителей, то можно применить такую команду:

```
mysql>SELECT * FROM drivers ORDER BY birth LIMIT 2;
```

Можно использовать два параметра: номер строки, с которой следует начать, и количество строк.

```
mysql>SELECT family_name FROM drivers LIMIT 2,3;
```

Последний запрос можно интерпретировать так: "Выбрать фамилии водителей, в результате отобразить три строки, начиная со строки 2". Нумерация строк начинается с нуля.

4.3. Самостоятельная работа №3

Для защиты лабораторной работы №4 необходимо выполнить самостоятельную работу №3 и ответить на контрольные вопросы.

В соответствии с номером по журналу, используя простые запросы, получить информацию в утилите mysql из учебной базы данных созданной в лабораторных работах №2-3.

1. Посчитать количество предметов с объемом часов <50

2. Вывести количество преподавателей предмета 43.
3. Вывести количество городов, в которых учатся студенты.
4. Напишите запрос для определения количества изучаемых предметов в каждом семестре.
5. Напишите запрос для определения количества студентов, не сдавших каждый экзамен.
6. Напишите запрос для определения количества студентов сдавших каждый экзамен.
7. Напишите запрос для получения среднего балла каждого студента.
8. Составьте запрос позволяющий выбрать студентов и выдать их максимальные оценки.
9. Составьте запрос позволяющий выбрать студентов и выдать их минимальные оценки.
10. Составьте запрос для таблицы EXAM_MARKS позволяющий подсчитать количество различных предметов.
11. Составьте запрос, для подсчета количества студентов, сдавших экзамен по предмету с идентификатором 10.
12. Выберите из таблицы экзаменационные оценки записи, в которых присутствуют записи об оценке.
13. Выберите из таблицы экзаменационные оценки записи, в которых отсутствуют оценки.
14. Вывести список студентов имена которых начинаются на И или С.
15. Вывести фамилию и имя всех преподавателей, проживающих в Воронеже.
16. Вывести фамилию и имя преподавателей, фамилия которых начинается на букву И.
17. Вывести наименование предметов, название которых начинается на букву И.
18. Вывести студентов и предметы, экзамены по которым сдавались между 10 и 20 января 2007 г.
19. Выбрать студентов и даты сдачи экзаменов по предмету 94.
20. Выбрать предметы и даты по экзаменам, которые сданы студентами с номерами 12 и 111.
21. Вывести студентов моложе 20 лет.
22. Выберите студентов, указав их имя, фамилию и курс, тех, которые проживают в Белгороде и их стипендия 1000 р.
23. Выберите студентов, указав их имя, фамилию и курс, тех, которые проживают в Воронеже и их стипендия ≤ 600 р.
24. Выберите университеты, рейтинг которых больше 300 баллов.
25. Выберите студентов, указав их имя, фамилию и курс, тех, которые не получают стипендию.
26. Выберите студентов, указав их имя, фамилию и курс, тех, которые получают стипендию > 600.
27. Выберите из таблицы Экзаменационные оценки список всех дат, исключить повторяющиеся.
28. Выберите из таблицы Экзаменационные оценки список всех оценок, исключить повторяющиеся.

29. Вывести наименование предмета, количество часов для каждого предмета в 1-м семестре.
30. Вывести все строки таблицы Экзаменационные оценки, в которых предмет имеет номер 12.

Тема 2 Проектирование реляционных баз данных

2.1. Запросы к двум и более таблицам

Многотабличные запросы позволяет сформировать записи путем объединения взаимосвязанных записей таблиц из таблиц базы данных и выбора из них нужных записей. Например, при объединении двух нормализованных связанных одно-многозначными отношениями таблиц, для которых обеспечивается связная целостность, результирующая запись образуется на основе записи подчинённой таблицы, в которую добавляются поля из связанной записи в главной таблице. Подобное объединение формирует ненормализованную таблицу, в которой число записей равно числу записей в подчиненной таблице. При этом данные главной таблицы дублируются в различных записях результирующей таблицы.

При проектировании и конструировании запроса важнейшим условием является правильное представление о том, как идет объединение записей таблиц при формировании результата.

Запрос, который обращен к нескольким записям одной таблицы или к нескольким таблицам одновременно, называется *объединением.* В теории баз данных определено несколько типов запросов, чаще других применяется внутреннее объединение (inner join), которое и будет описано в этом разделе.

Рассмотрим запрос, при котором надо выбрать данные из двух таблиц так, чтобы выбирались только те строки, значения которых по соединяемым одноименным столбцам совпадают. Например, нам требуется узнать фамилию водителя, который позавчера водил синюю машину. Для того чтобы получить все необходимые данные, вам потребуется запросить все три таблицы базы, только тогда вы соберете все нужные сведения.

Вам придется оперировать именами столбцов из разных таблиц. Следует учесть, что полное имя столбца состоит из имени базы данных, таблицы и собственно имени столбца, например, taxi.cars.color.

При работе с одной базой данных из запросов можно исключить упоминание имени этой базы. Если запросы адресуются к одной таблице, то имя таблицы также опускается. Но при объединении необходимо указывать полностью имена таблиц и столбцов.

Часто имена таблиц заменяют короткими псевдонимами (alias), указывая их в запросе после ключевого слова FROM. Псевдонимы для таблиц являются стандартной частью языка SQL и обычно состоят из одной буквы. Задать псевдоним можно с помощью ключевого слова AS:

```
mysql> SELECT color FROM cars AS c;
```

Псевдонимы удобно использовать в случае, когда имя результирующего столбца слишком длинное или выводятся данные из двух таблиц, в которых есть одноименные столбцы.

Между таблицами существуют связи по определенным столбцам. Столбец cars.id связывает таблицу cars с таблицей timetable через столбец timetable.car_number. Аналогично строится связь между таблицей drivers и timetable.

Объединение осуществляется по столбцам, образующим связь. При этом выбираются одинаковые значения полей, по которым осуществляется связь между таблицами:

```
timetable.car_number=cars.id
и
timetable.driver_number=drivers.id
```

Определим дату (данные из столбца timetable.use_on) и фамилию водителя (drivers. family_name), который водил машину определенного цвета (cars.color):

```
mysql> SELECT t.use_on, d. family_name, c.color
-> FROM timetable AS t, cars AS c, drivers AS d
-> WHERE t.car_number=c.id AMD t .driver_number=d. id;
```

Выборка данных происходит из таблиц timetable, cars и drivers, для которых определены псевдонимы, сокращающие запись запроса и облегчающие его редактирование. Выбираются только те строки из таблиц timetable и cars, в которых

указан одинаковый учетный номер автомобиля. А из таблиц timetable и drivers выбираются строки, в которых поля в столбце, содержащем учетный номер водителя, одинаковы.

2.2. Вложенные запросы

Поддержка вложенных запросов появилась в MySQL относительно недавно, начиная с версии 4.1. Вложенный запрос создает результирующую таблицу, данные которой используются внешним запросом. Вложенный запрос помещается в скобках после выражения WHERE.

Создадим вложенный запрос с целью выяснить, на каком автомобиле работал определенный водитель 30 мая 2006 г.:

```
mysql> SELECT model, reg_number FROM cars
-> WHERE id = (SELECT id
-> FROM timetable
-> WHERE date = '2006-05-30' AND driver_number=l);
```

Сначала выполняется команда SELECT внутри скобок. MySQL получает результат вложенного запроса и использует его в условии в выражении where во внешнем запросе SELECT. При этом результаты внутреннего запроса не отображаются. Конструкция запроса в приведенном только что примере подразумевает, что результат внутреннего запроса должен представлять собой единственное значение. В противном случае в результате выполнения этого запроса появится сообщение об ошибке: "Error 1241: Subquery returns more than 1 row". Вложенные запросы могут содержать команды SELECT, INSERT и UPDATE, а также SET.

В случае, когда результатом вложенного запроса является не одно значение, а таблица, синтаксис запроса несколько меняется. Например, определим, на каких автомобилях работали водители 8 и 12 августа 2006г.:

```
SELECT model, reg_number
FROM cars
WHERE id IN
```

```
(
   (SELECT car_number FROM timetable
   WHERE date = '2006-08-08'),

   (SELECT car_number FROM timetable
   WHERE date = '2006-08-12')
);
```

Внешний запрос выбирает модель и регистрационный номер тех автомобилей, учетные номера которых указаны в таблице timetable в записях, относящихся к 8 и 12 августа 2006 г. Перечень условий для выражения in пишется через запятую и помещается во внешние скобки. Вначале выполняются вложенные запросы, результаты которых формируют набор значений для выражения in.

3.3. Самостоятельная работа №4

Для защиты лабораторной работы №5 необходимо выполнить самостоятельную работу №4 и ответить на контрольные вопросы.

В соответствии с номером по журналу, используя вложенные подзапросы или соединение таблиц, получить информацию в утилите mysql из учебной базы данных созданной в лабораторных работах №2-3.

1. Составьте запрос, с подзапросом для получения данных об оценках предмета студента Иванова.
2. Составьте запрос, выбирающий данные об именах и фамилиях студентов, имеющих по предмету с идентификатором 56 балл выше общего среднего.
3. Составьте запрос, выбирающий данные об именах и фамилиях студентов, имеющих по предмету с идентификатором 43 балл ниже общего среднего.
4. Составьте запрос, выполняющий вывод фамилию студентов и количества предметов, по которым они экзаменовались.
5. Напишите команду SELECT, использующую связные подзапросы и выполняющую вывод, имен студентов у которых стипендия совпадает с максимальной стипендией по городу.
6. Напишите запрос, который выводит имена и идентификаторы студентов, для которых известно, что они проживают в городе в котором нет университета.
7. Напишите запрос, который позволяет вывести имена и идентификаторы всех студентов, для которых известно, что они проживают не в том городе, где находится их университет. Используйте соединение таблиц.

8. Напишите запрос, который позволяет вывести имена и идентификаторы всех студентов, для которых известно, что они проживают не в том городе, где находится их университет. Используйте связный подзапрос.
9. Вывести на экран фамилии студентов с оценкой по предметам и фамилией преподавателя принимавшего экзамен.
10. Вывести все экзамены, которые были приняты в ВУЗе с идентификатором 10.
11. Вывести экзамены и фамилий экзаменаторов на дату 18.06.06
12. Составьте запрос для таблицы, который выводит минимальные оценки каждого студента и его город.
13. Составьте запрос для таблицы, который выводит максимальные оценки каждого студента и его город.
14. Вывести всех преподавателей с нагрузкой >50ч и их город.
15. Вывести все предметы ВУЗов по семестрам.
16. Вывести результаты экзаменов ВУЗов с рейтингом >350.
17. Вывести ВУЗы, в которых средний бал студентов выше общего среднего бала.
18. Вывести средний бал студентов по всем предметам.
19. Вывести средний бал студентов за период1.01.07 – 15.01.07.
20. Вывести фамилии преподавателей, у которых на экзаменах средний бал ниже общего среднего бала.
21. Напишите команду SELECT, использующую связные подзапросы и выполняющую вывод, имен студентов у которых стипендия совпадает с максимальной стипендией по ВУЗу.
22. Вывести студентов с повышенной стипендией (>600) сдавших экзамены на тройку.
23. Вывести студентов со стипендией ≤600 сдавших экзамены без троек.
24. Вывести студентов и преподавателей, проживающих в одном городе.
25. Вывести всех преподавателей, работающих не в своем городе.
26. Вывести молодых студентов (<20 лет) со стипендией 1000 сдавших экзамены не на отлично.
27. Вывести предметы, которые никогда не сдавали студенты.
28. Вывести ВУЗы, в которых не проходил экзамен по информатике.
29. Вывести списки преподавателей с предметами, которые они ведут, объемом нагрузки этого предмета и номера семестра.
30. Вывести все отметки по предметам в ВУЗе с минимальным рейтингом.