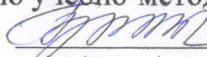


**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ НОВОСИБИРСКОЙ ОБЛАСТИ
«НОВОСИБИРСКИЙ ЭЛЕКТРОМЕХАНИЧЕСКИЙ КОЛЛЕДЖ»
(ГБПОУ НСО «Новосибирский электромеханический колледж»)**

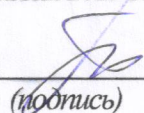
УТВЕРЖДАЮ
Заместитель директора
по учебно-методической работе
 Е.О. Кравченко
(подпись) (ФИО)
«16» июня 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.08 ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

Специальность 09.02.07 Информационные системы и программирование

Рассмотрено
на заседании кафедры общеобразовательных дисциплин
Протокол № 10 от «16» июня 2025 г.

Председатель кафедры  Белимова В.Г.
(подпись) (Ф.И.О.)

2025 г.

Фонд оценочных средств учебного предмета разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) по профессии 09.02.07 Информационные системы и программирование

Организация-разработчик: ГБПОУ НСО «НЭК»

Разработчик:

Комарова А.В., преподаватель высшей квалификационной категории

Согласовано:

Методист _____ Рахметова Т.Ю.

« ____ » _____ 20__ г.

Оглавление

1. Перечень компетенций
2. Паспорт фонда оценочных средств
 - 2.1. Структура фонда оценочных средств для текущего контроля и промежуточной аттестации
 - 2.2. Критерии оценивания компетенций
 - 2.3. Типовые контрольные задания
3. Экзаменационные билеты
 - 3.2. Эталонные ответы экзаменационных билетов. Практическая часть
 - 3.3. Эталонные ответы экзаменационных билетов. Теоретическая часть
4. Перечень материалов, оборудования и информационных источников
5. Методические материалы, определяющие процедуры оценивания знаний

1. Перечень компетенций

1.1 Перечень общих компетенций

Код	Наименование общих компетенций
ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языке

1.2 Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ПК 11.1	Осуществлять сбор, обработку и анализ информации для проектирования баз данных.
ПК 11.2	Проектировать базу данных на основе анализа предметной области.
ПК 11.3	Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.
ПК 11.4	Реализовывать базу данных в конкретной системе управления базами данных.
ПК 11.5	Администрировать базы данных.
ПК 11.6	Защищать информацию в базе данных с использованием технологии защиты информации

В результате освоения учебной дисциплины студент должен:

Код ПК, ОК	Умения	Знания
ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 11.1- 11.6	Проектировать реляционную базу данных. Использовать язык запросов для программного извлечения сведений из баз данных.	Основы теории баз данных. Модели данных. Особенности реляционной модели и проектирование баз данных. Изобразительные средства, используемые в ER- моделировании. Основы реляционной алгебры. Принципы проектирования баз данных. Обеспечение непротиворечивости и целостности данных. Средства проектирования структур баз данных. Язык запросов SQL.

**Матрица покрытия компетенций разделами курса
«Основы проектирования баз данных»**

Компетенция	Раздел 1	Раздел 2	Раздел 3	Раздел 4	Раздел 5
ОК 1	✓	✓	✓	✓	✓
ОК 2	✓	✓	✓	✓	✓
ОК 4	✓	✓	✓	✓	✓
ОК 5	✓	✓	✓	✓	✓
ОК 9	✓	✓	✓	✓	✓
ОК 10	✓	✓	✓	✓	✓
ПК 11.1		✓	✓	✓	✓
ПК 11.2	✓	✓	✓	✓	✓
ПК 11.3	✓		✓		✓
ПК 11.4		✓	✓	✓	✓
ПК 11.5			✓	✓	✓
ПК 11.6		✓	✓	✓	

2. Паспорт фонда оценочных средств

2.1. Структура фонда оценочных средств для текущего контроля и промежуточной аттестации

№ п/п	Результаты освоения УД/М (код и наименование)	Код контролируемой компетенции	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Входное тестирование		Контрольная работа «Входное тестирование»	
2	Тема 1. Основные понятия баз данных	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 11.1 - ПК 11.6	Контрольная работа №1 «Основные понятия баз данных»	Вопросы и задания экзамена
3	Тема 2. Взаимосвязи в моделях и реляционный подход к построению моделей	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 11.1 - ПК 11.6	Контрольная работа №2 «Взаимосвязи в моделях и реляционный подход к построению моделей»	Вопросы и задания экзамена
4	Тема 3. Этапы проектирования баз данных	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 11.1 - ПК 11.6	Контрольная работа №3 «Этапы проектирования баз данных»	Вопросы и задания экзамена
5	Тема 4. Проектирование структур баз данных	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 11.1 - ПК 11.6	Контрольная работа №4 «Проектирование структур баз данных»	Вопросы и задания экзамена
6	Тема 5. Организация запросов SQL	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 11.1 - ПК 11.6	Контрольная работа №5 «Организация запросов SQL»	Вопросы и задания экзамена

2.2. Критерии оценивания компетенций

2.3. Типовые контрольные задания

Контрольная работа №1 «Основные понятия баз данных»

Время: 45 минут

Часть I. Тестовое задание по теории (1 вариант для всех)

Инструкция: Выберите один или несколько правильных ответов.

1. **База данных - это:**
 - a) Программа для хранения данных
 - b) Совокупность организованных данных
 - c) Таблица в Excel
 - d) Упорядоченная совокупность данных, хранящаяся в памяти компьютера
2. **СУБД предназначена для:**
 - a) Создания баз данных
 - b) Удаления ненужных данных
 - c) Обеспечения целостности данных
 - d) Выполнения запросов к данным
3. **Атрибут сущности - это:**
 - a) Название базы данных
 - b) Характеристика объекта
 - c) Связь между таблицами
 - d) Тип данных в таблице
4. **Первичный ключ должен обладать свойствами:**
 - a) Уникальности
 - b) Минимальной длины
 - c) Непустоты
 - d) Простоты запоминания

Часть II. Практическое задание (2 варианта)

Вариант 1

Задание: Разработать структуру БД «Школьная библиотека». Определите:

1. Основные сущности (не менее 3)
2. Атрибуты для каждой сущности (3-5 атрибутов)
3. Выделите первичные ключи
4. Опишите связи между сущностями

Вариант 2

Задание: Разработать структуру БД «Спортивная секция». Определите:

1. Основные сущности (не менее 3)
2. Атрибуты для каждой сущности (3-5 атрибутов)
3. Выделите первичные ключи
4. Опишите связи между сущностями

Контрольная работа №2 «Взаимосвязи в моделях и реляционный подход»

Часть I. Тестовое задание по теории

1. **Связь «один-ко-многим» означает:**
 - a) Одной записи в одной таблице соответствует несколько записей в другой
 - b) Каждой записи соответствует только одна запись в другой таблице
 - c) Любой записи может соответствовать любое количество записей в другой таблице
2. **Внешний ключ необходим для:**
 - a) Ускорения работы базы данных
 - b) Организации связей между таблицами
 - c) Обеспечения уникальности записей
 - d) Создания резервных копий
3. **Реляционная модель данных основана на:**
 - a) Древовидной структуре
 - b) Сетевой структуре
 - c) Табличной структуре
 - d) Иерархической структуре

Часть II. Практическое задание (2 варианта)

Вариант 1

Задание: Даны сущности:

- **Пациенты** (id, ФИО, дата_рождения, адрес)
- **Врачи** (id, ФИО, специальность, категория)
- **Приемы** (id, дата_приема, пациент_id, врач_id, диагноз)

Определите типы связей и постройте схему связей между таблицами.

Вариант 2

Задание: Даны сущности:

- **Студенты** (id, ФИО, группа, курс)
- **Предметы** (id, название, часы, преподаватель)
- **Оценки** (id, студент_id, предмет_id, дата, оценка)

Определите типы связей и постройте схему связей между таблицами.

Контрольная работа №3 «Этапы проектирования БД»

Часть I. Тестовое задание по теории

1. **Первым этапом проектирования БД является:**
 - a) Нормализация
 - b) Определение предметной области
 - c) Создание физической модели
 - d) Написание запросов

2. **ER-диаграмма используется для:**
 - a) Описания физической структуры БД
 - b) Визуализации концептуальной модели
 - c) Написания SQL-запросов
 - d) Тестирования БД
3. **Нормализация необходима для:**
 - a) Ускорения работы БД
 - b) Устранения избыточности данных
 - c) Упрощения программирования
 - d) Создания резервных копий

Часть II. Практическое задание (2 варианта)

Вариант 1

Задание: Спроектируйте БД «Автосервис» по этапам:

1. Описание предметной области
2. Выделение сущностей и атрибутов
3. Построение концептуальной модели (ER-диаграмма)
4. Определение типов связей

Вариант 2

Задание: Спроектируйте БД «Туристическое агентство» по этапам:

1. Описание предметной области
2. Выделение сущностей и атрибутов
3. Построение концептуальной модели (ER-диаграмма)
4. Определение типов связей

Контрольная работа №4 «Проектирование структур БД»

Часть I. Тестовое задание по теории

1. **Таблица находится в 1NF, если:**
 - a) Все атрибуты атомарны
 - b) Нет повторяющихся групп
 - c) Все записи уникальны
 - d) Верно все вышеперечисленное
2. **Аномалия удаления проявляется в:**
 - a) Потере связи между данными
 - b) Невозможности удалить данные без потери информации
 - c) Удалении лишних данных
 - d) Автоматическом удалении дублей

Часть II. Практическое задание (2 варианта)

Вариант 1

Задание: Дана ненормализованная таблица «Заказы»:

text

Заказ № | Дата | Клиент | Товар1 | Цена1 | Товар2 | Цена2

Нормализуйте таблицу до 1NF и 2NF. Объясните каждое действие.

Вариант 2

Задание: Дана таблица "Сотрудники_Проекты":

text

Код_сотрудника | ФИО | Должность | Код_проекта | Название_проекта | Роль

Нормализуйте таблицу до 2NF. Объясните выполненные действия.

Контрольная работа №5 «Организация запросов SQL»

Часть I. Тестовое задание по теории

1. **Оператор SELECT** используется для:
 - a) Выбора данных из таблицы
 - b) Обновления данных
 - c) Удаления таблиц
 - d) Создания БД
2. **Оператор WHERE** применяется для:
 - a) Группировки данных
 - b) Фильтрации записей
 - c) Сортировки результатов
 - d) Объединения таблиц
3. **JOIN** используется для:
 - a) Объединения данных из нескольких таблиц
 - b) Создания новых таблиц
 - c) Удаления дублирующихся записей
 - d) Изменения структуры таблиц

Часть II. Практическое задание (2 варианта)

Общая схема БД для обоих вариантов:

- **Students** (student_id, name, group)
- **Subjects** (subject_id, title, teacher)
- **Grades** (grade_id, student_id, subject_id, date, grade)

Вариант 1

Задание: Напишите SQL-запросы:

1. Вывести список всех студентов группы «ИТ-21»
2. Найти количество студентов у каждого преподавателя
3. Вывести средний балл по каждому предмету

Вариант 2

Задание: Напишите SQL-запросы:

1. Вывести студентов, имеющих оценки выше 4
2. Найти преподавателя с наибольшим количеством студентов
3. Вывести список предметов с количеством полученных оценок

Ответы на контрольные работы

Контрольная работа №1 «Основные понятия баз данных»

Часть I. Тестовое задание (Эталон ответов)

Вопрос	Правильные ответы	Баллы
1	b, d (Совокупность организованных данных; Упорядоченная совокупность данных, хранящаяся в памяти компьютера)	1
2	a, c, d (Создания баз данных; Обеспечения целостности данных; Выполнения запросов к данным)	1
3	b (Характеристика объекта)	1
4	a, c (Уникальности; Непустоты)	1

Часть II. Практическое задание (Эталон оценки)

Вариант 1 «Школьная библиотека»

sql

-- Сущности и атрибуты:

Книги (book_id, название, автор, год_издания, издательство, id_полки)

Читатели (reader_id, ФИО, класс, дата_регистрации)

Выдачи (issue_id, book_id, reader_id, дата_выдачи, дата_возврата)

-- Связи:

-- Книги (1) - Выдачи(М) - один экземпляр книги выдается много раз

-- Читатели (1) - Выдачи(М) - один читатель делает много выдач

Вариант 2 «Спортивная секция»

sql

-- Сущности и атрибуты:

Спортсмены (sportsman_id, ФИО, возраст, разряд, дата_вступления)

Тренеры (trainer_id, ФИО, специализация, стаж)

Тренировки (training_id, sportsman_id, trainer_id, дата, результат)

-- Связи:

-- Тренеры (1) - Спортсмены(М) - один тренер тренирует многих спортсменов

-- Спортсмены (1) - Тренировки(М) - один спортсмен проходит много тренировок

Критерии оценки практической части:

- Сущности (3 шт.) - 1.5 балла (0.5 за каждую)
- Атрибуты (3-5 на сущность) - 2 балла
- Первичные ключи - 1 балл
- Связи между сущностями - 1.5 балла
- **Итого: 6 баллов**

Контрольная работа №2 «Взаимосвязи в моделях и реляционный подход»

Часть I. Тестовое задание (Эталон ответов)

Вопрос	Правильные ответы	Баллы
1	a (Одной записи в одной таблице соответствует несколько записей в другой)	1
2	b (Организации связей между таблицами)	1
3	c (Табличной структуре)	1

Часть II. Практическое задание (Эталон оценки)

Вариант 1 «Поликлиника»

sql

-- Связи:

-- Пациенты (1) ↔ Приемы(М) - один пациент может иметь много приемов

-- Врачи (1) ↔ Приемы(М) - один врач проводит много приемов

-- Схема: Пациенты ← Приемы → Врачи

Вариант 2 «ВУЗ»

sql

-- Связи:

-- Студенты (1) $\leftrightarrow$ Оценки(M) - один студент получает много оценок

-- Предметы (1) $\leftrightarrow$ Оценки(M) - по одному предмету ставят много оценок

-- Схема: Студенты $\leftarrow$ Оценки $\rightarrow$ Предметы

Критерии оценки практической части:

- Правильное определение всех связей - 2 балла
- Построение схемы связей - 2 балла
- Обоснование выбора типа связи - 2 балла
- **Итого: 6 баллов**

Контрольная работа №3 «Этапы проектирования БД»

Часть I. Тестовое задание (Эталон ответов)

Вопрос	Правильные ответы	Баллы
1	б (Определение предметной области)	1
2	б (Визуализации концептуальной модели)	1
3	б (Устранения избыточности данных)	1

Часть II. Практическое задание (Эталон оценки)

Вариант 1 "Автосервис"

sql

-- 1. Предметная область: учет клиентов, автомобилей, заказов на ремонт

-- 2. Сущности:

-- Клиенты (*client_id*, ФИО, телефон)

-- Автомобили (*car_id*, *client_id*, марка, модель, госномер)

-- Заказы (*order_id*, *car_id*, дата_заказа, описание_работ, стоимость)

-- 3. ER-диаграмма: Клиенты (1) $\rightarrow$ Автомобили(M) $\rightarrow$ Заказы(M)

-- 4. Типы связей: 1:M между всеми сущностями

Вариант 2 «Турагентство»

sql

-- 1. Предметная область: продажа туров, клиенты, менеджеры

-- 2. Сущности:

-- Туры (tour_id, направление, стоимость, длительность)

-- Клиенты (client_id, ФИО, паспорт)

-- Продажи (sale_id, tour_id, client_id, менеджер, дата_продажи)

-- 3. ER-диаграмма: Туры (1) → Продажи(M) ← Клиенты (1)

-- 4. Типы связей: 1:M от Туров и Клиентов к Продажам

Критерии оценки практической части:

- Описание предметной области - 1 балл
- Выделение сущностей и атрибутов - 2 балла
- Построение ER-диаграммы - 2 балла
- Определение типов связей - 1 балл
- Итого: 6 баллов

Контрольная работа №4 «Проектирование структур БД»

Часть I. Тестовое задание (Эталон ответов)

Вопрос	Правильные ответы	Баллы
1	d (Верно все вышеперечисленное)	1
2	b (Невозможности удалить данные без потери информации)	1

Часть II. Практическое задание (Эталон оценки)

Вариант 1 "Заказы"

sql

-- 1NF: Убираем повторяющиеся группы (Товар1, Цена1, Товар2, Цена2)

Заказы (заказ_№, дата, клиент)

Позиции_заказа (id, заказ_№, товар, цена)

-- 2NF: Все атрибуты зависят от полного ключа (уже выполнено)

-- Объяснение: Разделили на две таблицы для устранения повторяющихся групп

Вариант 2 «Сотрудники_Проекты»

sql

-- 2NF: Устраняем частичные зависимости

Сотрудники (код_сотрудника, ФИО, должность)

Проекты (код_проекта, название_проекта)

Участие_в_проектах (id, код_сотрудника, код_проекта, роль)

-- Объяснение: ФИО и должность зависят только от кода сотрудника,

-- название проекта зависит только от кода проекта

Критерии оценки практической части:

- Правильная нормализация до требуемой формы - 3 балла
- Объяснение выполненных действий - 2 балла
- Устранение аномалий - 1 балл
- **Итого: 6 баллов**

Контрольная работа №5 «Организация запросов SQL»

Часть I. Тестовое задание (Эталон ответов)

Вопрос	Правильные ответы	Баллы
1	a (Выбора данных из таблицы)	1
2	b (Фильтрации записей)	1
3	a (Объединения данных из нескольких таблиц)	1

Часть II. Практическое задание (Эталон оценки)

Вариант 1

sql

-- 1. Вывести список всех студентов группы "ИТ-21"

```
SELECT * FROM Students WHERE group = 'ИТ-21';
```

-- 2. Найти количество студентов у каждого преподавателя

```
SELECT teacher, COUNT(DISTINCT s.student_id) as student_count
```

```
FROM Subjects sub
JOIN Grades g ON sub.subject_id = g.subject_id
JOIN Students s ON g.student_id = s.student_id
GROUP BY teacher;
```

-- 3. Вывести средний балл по каждому предмету

```
SELECT title, AVG(grade) as avg_grade
FROM Subjects sub
JOIN Grades g ON sub.subject_id = g.subject_id
GROUP BY title;
```

Вариант 2

sql

-- 1. Вывести студентов, имеющих оценки выше 4

```
SELECT DISTINCT s.name, s.group
FROM Students s
JOIN Grades g ON s.student_id = g.student_id
WHERE g.grade > 4;
```

-- 2. Найти преподавателя с наибольшим количеством студентов

```
SELECT teacher, COUNT(DISTINCT s.student_id) as student_count
FROM Subjects sub
JOIN Grades g ON sub.subject_id = g.subject_id
JOIN Students s ON g.student_id = s.student_id
GROUP BY teacher
ORDER BY student_count DESC
LIMIT 1;
```

-- 3. Вывести список предметов с количеством полученных оценок

```
SELECT title, COUNT(grade) as grades_count
FROM Subjects sub
LEFT JOIN Grades g ON sub.subject_id = g.subject_id
GROUP BY title;
```

Критерии оценки практической части:

- Правильный синтаксис каждого запроса - 1 балл ($\times 3 = 3$ балла)
 - Корректная логика запроса - 1 балл ($\times 3 = 3$ балла)
 - **Итого: 6 баллов**
-

Общая система оценивания:

- **Теоретическая часть:** 3 балла (по 1 баллу за вопрос)
- **Практическая часть:** 6 баллов (по критериям выше)
- **Всего за работу:** 9 баллов

Шкала перевода в оценки:

- 8-9 баллов = **5 (отлично)**
- 6-7 баллов = **4 (хорошо)**
- 4-5 баллов = **3 (удовлетворительно)**
- 0-3 балла = **2 (неудовлетворительно)**

4. Экзаменационные билеты

I. ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ

Тема 1. Основные понятия баз данных

1. Классификация и сравнительная характеристика СУБД
2. Архитектуры баз данных (двух- и трёхзвенная структуры, клиент-сервер, файл-сервер)
3. Понятия банка данных и базы данных. Пользователи банков данных. Преимущества использования БД

Тема 2. Взаимосвязи в моделях и реляционный подход

4. Понятие модели данных. Классификация моделей данных. Даталогические и физические модели данных
5. Основные понятия реляционной модели: декартово произведение, отношение, схема отношения, домен, кортеж, первичный ключ, внешний ключ
6. Инфологическое моделирование предметной области. Модель «сущность-связь». Понятия сущность, атрибут, ключевой атрибут, связь. Типы связей
7. Ограничения целостности реляционной модели данных

Тема 3. Этапы проектирования баз данных

8. Жизненный цикл БД
9. Концептуальная, логическая и физическая модели данных
10. Разработка схемы данных и задание системы взаимосвязей между таблицами

11. Создание и перемещение файла БД. Разработка и описание структур таблиц данных

Тема 4. Проектирование структур баз данных

12. Первая нормальная форма. Функциональная зависимость и вторая нормальная форма. Полная функциональная зависимость, транзитивная зависимость, третья нормальная форма
13. Нормальная форма Бойса-Кодда
14. Четвертая нормальная форма. Теорема Фейджина. Пятая нормальная форма. Особые свойства бинарных отношений. Физическая организация базы данных

Тема 5. Организация запросов SQL

15. Язык SQL. Синтаксис оператора SELECT. Применение агрегатных функций и вложенных запросов в операторе выбора. SQL-функции. Вложенные подзапросы
16. Вычисляемые поля. Группировка данных. Использование агрегатных функций

II. ПРАКТИЧЕСКИЕ ЗАДАНИЯ

1. Разработать программу БД Гостиница. Предусмотреть ввод данных на форме и запись данных в две таблицы: в первую вносятся паспортные данные клиента, во вторую дата, номер и стоимость проживания.
2. В БД Стоматология создайте запрос для отбора пациентов с определенным видом заболевания. Условие отбора задается через форму ввода.
3. Создать БД Книжный магазин. Данные заполняются через форму ввода, импортируются в файл Excel.
4. Создать программу, удаляющую записи в одной из таблиц БД Стоматология в файл Excel, согласно некоторому условию. Условие отбора задается через форму ввода.
5. Разработать программу БД Отдел кадров. Данные заполняются через форму ввода, записываются в две таблицы. В первую записываются фамилия, имя, отчество, дата рождения, специальность, во вторую – должность, оклад.
6. Создать программу для фильтрации данных в таблице Товар БД Заказы, согласно одному из двух выбранных условий (выбор осуществляется пользователем с помощью Inputbox).
7. Создать программу для просмотра данных о товарах, записанных в таблице БД Заказы (выбор осуществляется пользователем с помощью Inputbox).
8. В БД Стоматология создать на форме кнопку, обновляющую записи в одной из таблиц, согласно некоторому условию (выбор осуществляется пользователем с помощью Inputbox).
9. Разработать БД Клиенты. Предусмотреть ввод данных на форме и запись данных в две таблицы: в первую вносятся паспортные данные, во вторую дата, номер договора и стоимость услуг. Организовать запрос по некоторому условию (выбор осуществляется пользователем с помощью Inputbox).
10. В БД Стоматология создать на форме кнопку, добавляющую в одну из таблиц записи из внешнего файла (в качестве разделителя используется пробел). Создать кнопку распечатки данных из таблицы.
11. Создать БД Отдел кадров. Данные заполняются через форму ввода, записываются в файл Excel и отображаются в таблице.
12. В БД Стоматология создать на форме кнопку, обновляющую записи в одной из таблиц, согласно некоторому условию (выбор осуществляется пользователем с помощью Inputbox).
13. Создать программу для просмотра данных о товарах, записанных в таблице БД Заказы (выбор осуществляется пользователем с помощью Inputbox).

14. Разработать программу БД Отдел кадров. Данные заполняются через форму ввода, записываются в две таблицы. В первую записываются фамилия, имя, отчество, дата рождения, специальность, во вторую – должность, оклад. Организовать запрос по условию (выбор осуществляется пользователем с помощью Inputbox).

15. В БД имеются две связанные таблицы: в первой указаны ФИО, Специальность, Должность, Число отработанных часов, Заработная плата; во второй – справочник тарифов почасовой оплаты (Должность, Почасовая оплата). Начислить заработную плату (в виде отчета). При формировании отчета плата за час должна быть получена из справочника.

16. Создать базу данных Автомагазин, состоящую из таблиц Склад {Марка, Объем двигателя, Цвет, Тип кузова, Год выпуска, Номер кузова, Поставщики}, Поставщики {Фирма, ФИО, Телефон, Адрес}. Создать связь этих таблиц. Создать форму с подчинением. Добавить кнопку для вывода отчета по данным запроса о поставках (выбор осуществляется пользователем с помощью Inputbox).

КРИТЕРИИ ОЦЕНКИ ЭКЗАМЕНА

Итоговая проверка знаний предусматривает ответы на вопросы определенного уровня сложности по разделам дисциплины, изучаемым в течение всего курса, что составляет базовую подготовку студентов, предусмотренную стандартом.

Систематизация теоретических заданий по сложности вводится для дифференциации знаний и умений студентов при рубежном и итоговом контроле.

При проведении экзамена по дисциплине, кроме проверки теоретических знаний, предусматривается проверка практических знаний и умений студентов, а именно умение выделять основные сущности и атрибуты сущностей предметной области, выделять связи сущностей и отображать их на конкретную модель данных, работать в системе управления базами данных, строить запросы, формировать отчеты.

Критерии оценки:

Оценка 5 (отл) ставится при выполнении всех вышеперечисленных требований по заданию без ошибок и замечаний.

При наличии небольших недочетов на одном – двух этапах решения задачи ставится оценка 4 (хор).

При ошибках на этапе разработки алгоритма решения задачи и / или этапе реализации средствами СУБД (ошибка в алгоритме влечет за собой ошибку в реализации) ставится оценка 3 (удовл).

Оценку 2 (неуд) получает студент, не справившийся с заданием.

Итоговая экзаменационная оценка складывается из оценки ответа на теоретические вопросы и оценки выполнения задания.

Исключение – студент не справился с заданием, т.е. по этапу решения задачи имеет неудовлетворительную оценку.

Устный опрос – это 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.

Критерии оценки устного опроса студентов:

Оценка «отлично»:

- глубокое и прочное усвоение материала темы или раздела;
- полные, последовательные, грамотные, логически излагаемые аргументированные ответы;
- демонстрация обучающимся знаний в объеме пройденной программы и дополнительно рекомендованной литературы;
- воспроизведение учебного материала с требуемой степенью точности.

Оценка «хорошо»:

- наличие несущественных ошибок, недостаточно аргументированные ответы на вопросы;
- демонстрация обучающимся знаний в объеме пройденной программы;
- четкое изложение учебного материала.

Оценка «удовлетворительно»:

- наличие несущественных ошибок в ответе, отсутствие аргументации, но достаточно грамотное и логичное изложение;
- демонстрация обучающимся недостаточно полных знаний по пройденной программе, отсутствие аргументации;
- не структурированное, не грамотное и не логичное изложение учебного материала при ответе.

Оценка «неудовлетворительно»:

- незнание материала темы или раздела;
- серьезные ошибки при ответе.

3.2. Эталонные ответы экзаменационных билетов. Практическая часть

ЗАДАНИЕ 1: БД «ГОСТИНИЦА»

Эталон решения:

sql

-- 1. Создание таблиц

```
CREATE TABLE Клиенты (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    паспортные_данные VARCHAR(100) NOT NULL,  
    ФИО VARCHAR(100) NOT NULL,  
    телефон VARCHAR(20)  
);
```

```
CREATE TABLE Проживание (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    id_клиента INT,  
    дата_заезда DATE,  
    дата_выезда DATE,  
    номер INT,
```

```

стоимость DECIMAL(10,2),
FOREIGN KEY (id_клиента) REFERENCES Клиенты(id)
);

```

-- 2. Код для формы (C#/VB.NET)

```

private void btnСохранить_Click(object sender, EventArgs e)
{
    // Сохранение в таблицу Клиенты
    string sqlКлиент = "INSERT INTO Клиенты (паспортные_данные, ФИО, телефон)
VALUES (" +
        txtПаспорт.Text + ", " + txtФИО.Text + ", " + txtТелефон.Text + ")";
    ExecuteSQL(sqlКлиент);

    // Получение ID последнего добавленного клиента
    string sqlId = "SELECT MAX(id) FROM Клиенты";
    int idКлиента = GetLastId(sqlId);

    // Сохранение в таблицу Проживание
    string sqlПроживание = "INSERT INTO Проживание (id_клиента, дата_заезда,
дата_выезда, номер, стоимость) VALUES (" +
        idКлиента + ", " + dtЗаезд.Value + ", " + dtВыезд.Value + ", " + txtНомер.Text + ", " +
txtСтоимость.Text + ")";
    ExecuteSQL(sqlПроживание);

    MessageBox.Show("Данные сохранены!");
}

```

ЗАДАНИЕ 2: ЗАПРОС ДЛЯ БД «СТОМАТОЛОГИЯ»

Эталон решения:

sql

-- Структура таблицы Пациенты

```
CREATE TABLE Пациенты (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    ФИО VARCHAR(100),  
    заболевание VARCHAR(100),  
    возраст INT,  
    дата_посещения DATE  
);
```

-- Код для формы

```
private void btnНайти_Click(object sender, EventArgs e)  
{  
    string заболевание = txtЗаболевание.Text;  
  
    string sql = "SELECT * FROM Пациенты WHERE заболевание = '" + заболевание + "'";  
  
    // Заполнение DataGridView  
    DataTable dt = GetDataTable(sql);  
    dataGridView1.DataSource = dt;  
  
    if (dt.Rows.Count == 0)  
        MessageBox.Show("Пациентов с таким заболеванием не найдено");  
}
```

ЗАДАНИЕ 3: БД «КНИЖНЫЙ МАГАЗИН»

Эталон решения:

sql

-- Создание таблицы

```
CREATE TABLE Книги (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    название VARCHAR(150),  
    автор VARCHAR(100),  
    жанр VARCHAR(50),  
    цена DECIMAL(8,2),  
    количество INT  
);
```

-- Код экспорта в Excel

```
private void btnЭкспорт_Click(object sender, EventArgs e)  
{  
    string sql = "SELECT * FROM Книги";  
    DataTable dt = GetDataTable(sql);  
  
    // Создание Excel файла  
    Excel.Application excelApp = new Excel.Application();  
    Excel.Workbook workbook = excelApp.Workbooks.Add();  
    Excel.Worksheet worksheet = workbook.ActiveSheet;  
  
    // Заголовки  
    for (int i = 0; i < dt.Columns.Count; i++)  
        worksheet.Cells[1, i+1] = dt.Columns[i].ColumnName;  
  
    // Данные  
    for (int i = 0; i < dt.Rows.Count; i++)  
        for (int j = 0; j < dt.Columns.Count; j++)  
            worksheet.Cells[i+2, j+1] = dt.Rows[i][j];
```

```

workbook.SaveAs("Книги_экспорт.xlsx");
excelApp.Quit();

MessageBox.Show("Данные экспортированы в Excel!");
}

```

ЗАДАНИЕ 4: УДАЛЕНИЕ ЗАПИСЕЙ ИЗ БД «СТОМАТОЛОГИЯ»

Эталон решения:

```

sql
-- Код для формы удаления
private void btnУдалить_Click(object sender, EventArgs e)
{
    string условие = txtУсловие.Text; // Например: "возраст > 60"

    // Сначала экспортируем удаляемые записи
    string sqlЭкспорт = "SELECT * FROM Пациенты WHERE " + условие;
    DataTable dt = GetDataTable(sqlЭкспорт);

    if (dt.Rows.Count > 0) {
        // Экспорт в Excel
        ExportToExcel(dt, "Удаленные_записи.xlsx");

        // Удаление записей
        string sqlУдалить = "DELETE FROM Пациенты WHERE " + условие;
        ExecuteSQL(sqlУдалить);

        MessageBox.Show($"Удалено {dt.Rows.Count} записей. Данные сохранены в Excel.");
    } else {
        MessageBox.Show("Записей по указанному условию не найдено");
    }
}

```

ЗАДАНИЕ 5: БД «ОТДЕЛ КАДРОВ»

Эталон решения:

sql

-- Создание таблиц

```
CREATE TABLE Сотрудники (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    ФИО VARCHAR(100),  
    дата_рождения DATE,  
    специальность VARCHAR(100)  
);  
  
CREATE TABLE Должности (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    id_сотрудника INT,  
    должность VARCHAR(100),  
    оклад DECIMAL(10,2),  
    FOREIGN KEY (id_сотрудника) REFERENCES Сотрудники(id)  
);
```

-- Код сохранения

```
private void btnСохранить_Click(object sender, EventArgs e)  
{  
    // Сохранение в Сотрудники  
    string sqlСотрудник = "INSERT INTO Сотрудники (ФИО, дата_рождения,  
    специальность) VALUES ('" +  
        txtФИО.Text + "', '" + dtРождения.Value + "', '" + txtСпециальность.Text + "')";  
    ExecuteSQL(sqlСотрудник);  
  
    // Сохранение в Должности  
    int idСотрудника = GetLastId("SELECT MAX(id) FROM Сотрудники");  
    string sqlДолжность = "INSERT INTO Должности (id_сотрудника, должность, оклад)  
    VALUES (" +  
        idСотрудника + ", '" + txtДолжность.Text + "', " + txtОклад.Text + ")";  
    ExecuteSQL(sqlДолжность);
```

```
}
```

ЗАДАНИЕ 6: ФИЛЬТРАЦИЯ ТОВАРОВ

Эталон решения:

```
sql
```

```
-- Код фильтрации
```

```
private void btnФильтровать_Click(object sender, EventArgs e)
```

```
{
```

```
    string выбор = Microsoft.VisualBasic.Interaction.InputBox(
```

```
        "Выберите тип фильтрации:\n1 - По цене\n2 - По категории", "Фильтрация", "1");
```

```
    string sql = "";
```

```
    if (выбор == "1") {
```

```
        string цена = Microsoft.VisualBasic.Interaction.InputBox("Введите минимальную цену:",  
"Фильтр по цене", "1000");
```

```
        sql = "SELECT * FROM Товары WHERE цена >= " + цена;
```

```
    } else if (выбор == "2") {
```

```
        string категория = Microsoft.VisualBasic.Interaction.InputBox("Введите категорию:",  
"Фильтр по категории", "Электроника");
```

```
        sql = "SELECT * FROM Товары WHERE категория = '" + категория + "'";
```

```
    }
```

```
    DataTable dt = GetDataTable(sql);
```

```
    dataGridView1.DataSource = dt;
```

```
}
```

ЗАДАНИЕ 7: ПРОСМОТР ТОВАРОВ

Эталон решения:

```
sql
private void btnПросмотреть_Click(object sender, EventArgs e)
{
    string критерий = Microsoft.VisualBasic.Interaction.InputBox(
        "Введите критерий:\nвсе - все товары\nдешевые - до 1000 руб\nдорогие - от 5000 руб",
        "Просмотр товаров", "все");

    string sql = "SELECT * FROM Товары";

    switch (критерий.ToLower()) {
        case "дешевые": sql += " WHERE цена < 1000"; break;
        case "дорогие": sql += " WHERE цена > 5000"; break;
        // "все" - оставляем без изменений
    }

    dataGridView1.DataSource = GetDataTable(sql);
}
```

ЗАДАНИЕ 8: ОБНОВЛЕНИЕ ЗАПИСЕЙ В СТОМАТОЛОГИИ

Эталон решения:

```
sql
private void btnОбновить_Click(object sender, EventArgs e)
{
    string условие = Microsoft.VisualBasic.Interaction.InputBox(
        "Введите условие WHERE (например: диагноз = 'кариес'):", "Условие обновления");

    string новыеЗначения = Microsoft.VisualBasic.Interaction.InputBox(
        "Введите новые значения (например: диагноз = 'пульпит', статус = 'вылечен'):",
        "Новые значения");
}
```

```

if (!string.IsNullOrEmpty(условие) && !string.IsNullOrEmpty(новыеЗначения)) {
    string sql = "UPDATE Пациенты SET " + новыеЗначения + " WHERE " + условие;
    int количество = ExecuteSQL(sql);
    MessageBox.Show($"Обновлено записей: {количество}");
}
}

```

ЗАДАНИЕ 9: БД «КЛИЕНТЫ» С ЗАПРОСОМ

Эталон решения:

```

sql
-- Таблицы
CREATE TABLE Клиенты (
    id INT PRIMARY KEY AUTO_INCREMENT,
    паспортные_данные VARCHAR(100),
    ФИО VARCHAR(100)
);

CREATE TABLE Договоры (
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_клиента INT,
    дата DATE,
    номер_договора VARCHAR(50),
    стоимость_услуг DECIMAL(10,2),
    FOREIGN KEY (id_клиента) REFERENCES Клиенты(id)
);

-- Код поиска
private void btnНайти_Click(object sender, EventArgs e)
{
    string условие = Microsoft.VisualBasic.Interaction.InputBox(
        "Введите ФИО клиента или номер договора:", "Поиск");
}

```

```

        string sql = @"SELECT к.ФИО, к.паспортные_данные, д.номер_договора, д.дата,
        д.стоимость_услуг
        FROM Клиенты к
        JOIN Договоры д ON к.id = д.id_клиента
        WHERE к.ФИО LIKE '%" + условие + "%' OR д.номер_договора LIKE '%" +
        условие + "%'";

        dataGridView1.DataSource = GetDataTable(sql);
    }

```

ЗАДАНИЕ 10: ИМПОРТ И ПЕЧАТЬ

Эталон решения:

```

sql
private void btnИмпорт_Click(object sender, EventArgs e)
{
    OpenFileDialog dialog = new OpenFileDialog();
    if (dialog.ShowDialog() == DialogResult.OK) {
        string[] lines = File.ReadAllLines(dialog.FileName);
        foreach (string line in lines) {
            string[] данные = line.Split(' ');
            if (данные.Length >= 3) {
                string sql = "INSERT INTO Пациенты (ФИО, заболевание, возраст) VALUES ('" +
                    данные[0] + "', '" + данные[1] + "', " + данные[2] + ")";
                ExecuteSQL(sql);
            }
        }
        MessageBox.Show("Данные импортированы!");
    }
}

private void btnПечать_Click(object sender, EventArgs e)
{
    PrintDocument doc = new PrintDocument();
    doc.PrintPage += new PrintPageEventHandler(PrintPageHandler);
}

```

```

PrintDialog printDialog = new PrintDialog();
printDialog.Document = doc;
if (printDialog.ShowDialog() == DialogResult.OK) {
    doc.Print();
}
}

```

ЗАДАНИЕ 11: ОТДЕЛ КАДРОВ С EXCEL И ТАБЛИЦЕЙ

Эталон решения:

```

sql
private void Form_Load(object sender, EventArgs e)
{
    // Загрузка данных в DataGridView
    string sql = "SELECT с.ФИО, с.дата_рождения, с.специальность, д.должность, д.оклад "
+
    "FROM Сотрудники с JOIN Должности д ON с.id = д.id_сотрудника";
    dataGridView1.DataSource = GetDataTable(sql);
}

private void btnЭкспорт_Click(object sender, EventArgs e)
{
    ExportToExcel((DataTable)dataGridView1.DataSource, "Сотрудники.xlsx");
}

```

ЗАДАНИЕ 15: РАСЧЕТ ЗАРПЛАТЫ

Эталон решения:

sql

-- Таблицы

```
CREATE TABLE Сотрудники (  
    ФИО VARCHAR(100),  
    Специальность VARCHAR(100),  
    Должность VARCHAR(100),  
    Часы_отработки INT,  
    Зарплата DECIMAL(10,2)  
);
```

```
CREATE TABLE Тарифы (  
    Должность VARCHAR(100),  
    Почасовая_оплата DECIMAL(8,2)  
);
```

-- Запрос для отчета

```
string sqlОтчет = @"SELECT  
    с.ФИО,  
    с.Должность,  
    с.Часы_отработки,  
    т.Почасовая_оплата,  
    (с.Часы_отработки * т.Почасовая_оплата) as Начислено  
FROM Сотрудники с  
JOIN Тарифы т ON с.Должность = т.Должность";
```

-- Обновление зарплаты в таблице

```
string sqlОбновить = @"UPDATE Сотрудники с  
JOIN Тарифы т ON с.Должность = т.Должность  
SET с.Зарплата = с.Часы_отработки * т.Почасовая_оплата";
```

ЗАДАНИЕ 16: БД «АВТОМАГАЗИН»

Эталон решения:

sql

-- Таблицы со связью

```
CREATE TABLE Поставщики (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    Фирма VARCHAR(100),  
    ФИО_контактного_лица VARCHAR(100),  
    Телефон VARCHAR(20),  
    Адрес VARCHAR(200)  
);
```

```
CREATE TABLE Склад (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    Марка VARCHAR(50),  
    Объем_двигателя DECIMAL(3,1),  
    Цвет VARCHAR(30),  
    Тип_кузова VARCHAR(30),  
    Год_выпуска INT,  
    Номер_кузова VARCHAR(50),  
    id_поставщика INT,  
    FOREIGN KEY (id_поставщика) REFERENCES Поставщики(id)  
);
```

-- Код для отчета

```
private void btnОтчет_Click(object sender, EventArgs e)  
{  
    string условие = Microsoft.VisualBasic.Interaction.InputBox(  
        "Введите условие для отчета (например: Год_выпуска > 2020):", "Отчет по  
поставкам");  
  
    string sql = @"SELECT  
        п.Фирма, п.ФИО_контактного_лица, п.Телефон,
```

```

        с.Марка, с.Цвет, с.Тип_кузова, с.Год_выпуска
FROM Склад с
JOIN Поставщики п ON с.id_поставщика = п.id";

if (!string.IsNullOrEmpty(условие))
    sql += " WHERE " + условие;

dataGridView1.DataSource = GetDataTable(sql);
}

```

ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ:

```

csharp
// Выполнение SQL-запроса
private int ExecuteSQL(string sql)
{
    using (MySqlConnection conn = new MySqlConnection(connectionString))
    {
        conn.Open();
        MySqlCommand cmd = new MySqlCommand(sql, conn);
        return cmd.ExecuteNonQuery();
    }
}

// Получение DataTable
private DataTable GetDataTable(string sql)
{
    DataTable dt = new DataTable();
    using (MySqlConnection conn = new MySqlConnection(connectionString))
    {
        conn.Open();
        MySqlDataAdapter da = new MySqlDataAdapter(sql, conn);
        da.Fill(dt);
    }
}

```

```

        return dt;
    }

    // Получение последнего ID
    private int GetLastId(string sql)
    {
        using (MySQLConnection conn = new MySQLConnection(connectionString))
        {
            conn.Open();
            MySqlCommand cmd = new MySqlCommand(sql, conn);
            return Convert.ToInt32(cmd.ExecuteScalar());
        }
    }
}

```

3.3. Эталонные ответы экзаменационных билетов. Теоретическая часть

Тема 1. Основные понятия баз данных

1. Классификация и сравнительная характеристика СУБД

Классификация СУБД:

1. По модели данных:

- Реляционные (MySQL, PostgreSQL, Oracle)
- Иерархические (IMS)
- Сетевые (IDMS)
- Объектно-ориентированные (Cache)
- NoSQL (MongoDB, Cassandra)

2. По степени распределенности:

- Локальные (Access, SQLite)
- Распределенные (Oracle RAC, MySQL Cluster)

3. По способу доступа:

- Файл-серверные (Access, FoxPro)
- Клиент-серверные (Oracle, MySQL, PostgreSQL)
- Встраиваемые (SQLite)

Сравнительная характеристика:

Параметр	Файл-сервер	Клиент-сервер
Производительность	Низкая	Высокая
Безопасность	Слабая	Сильная
Масштабируемость	Ограничена	Хорошая
Стоимость	Низкая	Средняя-высокая

2. Архитектуры баз данных

Двухзвенная архитектура (клиент-сервер):

- **Клиент:** Представление + бизнес-логика
- **Сервер:** Хранение данных + СУБД
- **Преимущества:** Простота реализации
- **Недостатки:** Высокая нагрузка на клиента

Трёхзвенная архитектура:

- **Клиент:** Только представление (UI)
- **Сервер приложений:** Бизнес-логика
- **Сервер БД:** Хранение данных + СУБД
- **Преимущества:** Масштабируемость, безопасность

Файл-серверная архитектура:

- Файлы БД хранятся на файловом сервере
- Каждый клиент самостоятельно обрабатывает данные
- **Проблемы:** Блокировки, целостность, безопасность

Клиент-серверная архитектура:

- Централизованная обработка на сервере
- Клиенты только отправляют запросы
- **Преимущества:** Безопасность, производительность

3. Банк данных и база данных

Банк данных (БнД) - это система, включающая:

- Базу данных
- СУБД
- Административные средства
- Методы и языки доступа

База данных (БД) - это организованная совокупность данных, хранящаяся в памяти компьютера.

Пользователи БнД:

1. **Конечные пользователи:** Работают через приложения
2. **Разработчики приложений:** Создают программы для работы с БД
3. **Администраторы БД (DBA):** Управление, безопасность, резервное копирование

Преимущества использования БД:

- Устранение избыточности данных
 - Обеспечение целостности и согласованности
 - Совместный доступ к данным
 - Независимость данных и программ
 - Стандартизация данных
-

Тема 2. Взаимосвязи в моделях и реляционный подход

4. Модели данных

Модель данных - совокупность правил описания и обработки данных.

Классификация моделей данных:

1. **Концептуальные (инфологические):** ER-модели, семантические сети
2. **Логические (дatalogические):** Реляционная, иерархическая, сетевая
3. **Физические:** Описание физического хранения на носителях

Дatalogическая модель - описывает логическую структуру данных для СУБД.

Физическая модель - описывает способ хранения данных на физических носителях.

5. Основные понятия реляционной модели

Декартово произведение множеств А и В - множество всех упорядоченных пар (a,b), где $a \in A$, $b \in B$.

Отношение - подмножество декартова произведения доменов.

Схема отношения - заголовок отношения (названия атрибутов).

Домен - множество допустимых значений атрибута.

Кортеж - строка таблицы, элемент отношения.

Первичный ключ (Primary Key) - уникальный идентификатор кортежа.

Внешний ключ (Foreign Key) - атрибут для связи с первичным ключом другого отношения.

6. Инфологическое моделирование. Модель «сущность-связь»

Сущность - реальный или абстрактный объект, информация о котором хранится.

Атрибут - характеристика сущности.

Ключевой атрибут - уникально идентифицирует экземпляр сущности.

Связь - ассоциация между сущностями.

Типы связей:

- **Один-к-одному (1:1)** - Один сотрудник → Одно рабочее место
 - **Один-ко-многим (1:M)** - Один отдел → Много сотрудников
 - **Многие-ко-многим (M:N)** - Много студентов → Много курсов
-

7. Ограничения целостности реляционной модели

Целостность сущности: Первичный ключ не может быть NULL и должен быть уникальным.

Целостность ссылок: Значение внешнего ключа должно существовать в связанной таблице или быть NULL.

Пользовательская целостность: Бизнес-правила, задаваемые разработчиком.

Тема 3. Этапы проектирования баз данных

8. Жизненный цикл БД

1. **Планирование:** Анализ требований, выбор СУБД
2. **Проектирование:** Концептуальная, логическая, физическая модели
3. **Реализация:** Создание БД, написание приложений
4. **Эксплуатация:** Работа пользователей, сопровождение
5. **Развитие:** Модификация, оптимизация
6. **Завершение:** Архивация, миграция на новую систему

9. Концептуальная, логическая и физическая модели

Концептуальная модель:

- Независима от СУБД
- ER-диаграммы
- Описание предметной области

Логическая модель:

- Зависит от типа СУБД
- Нормализованные таблицы
- Связи, ключи

Физическая модель:

- Конкретная реализация в СУБД
- Типы данных, индексы
- Размещение на диске

10. Разработка схемы данных и связей

Этапы:

1. Выделение сущностей и атрибутов
2. Определение первичных ключей
3. Установление связей между сущностями
4. Определение типов связей (1:1, 1:M, M:N)
5. Нормализация схемы

11. Создание и перемещение файла БД

Создание БД:

CREATE DATABASE имя_базы;

Создание таблиц:

```
CREATE TABLE имя_таблицы (  
    поле1 тип_данных,  
    поле2 тип_данных,  
    PRIMARY KEY (поле1)  
);
```

Перемещение БД: Экспорт/импорт данных, резервное копирование и восстановление.

Тема 4. Проектирование структур баз данных

12. Нормальные формы

Первая нормальная форма (1NF):

- Все атрибуты атомарны
- Нет повторяющихся групп
- Все кортежи уникальны

Вторая нормальная форма (2NF):

- Находится в 1NF
- Нет частичных зависимостей от ключа

Третья нормальная форма (3NF):

- Находится в 2NF
- Нет транзитивных зависимостей

Функциональная зависимость $X \rightarrow Y$: Значение Y однозначно определяется значением X .

Полная функциональная зависимость: Y зависит от всего составного ключа.

Транзитивная зависимость: $X \rightarrow Y$ и $Y \rightarrow Z$, тогда $X \rightarrow Z$.

13. Нормальная форма Бойса-Кодда (BCNF)

Условия:

- Отношение в 3NF
- Каждая детерминанта - потенциальный ключ

Детерминант - атрибут или группа атрибутов, от которых функционально зависит другой атрибут.

14. Четвертая и пятая нормальные формы

Четвертая нормальная форма (4NF):

- Отношение в BCNF
- Нет многозначных зависимостей

Теорема Фейджина: Отношение находится в 4NF тогда и только тогда, когда оно является соединением своих проекций без потерь.

Пятая нормальная форма (5NF):

- Отношение в 4NF
- Нет зависимостей соединения

Бинарные отношения: Связи между двумя сущностями.

Тема 5. Организация запросов SQL

15. Язык SQL. Оператор SELECT

Базовый синтаксис:

sql

SELECT [DISTINCT] столбцы

FROM таблицы

[WHERE условие]

[GROUP BY столбцы]

[HAVING условие_группы]

[ORDER BY столбцы [ASC|DESC]];

Агрегатные функции:

- COUNT() - количество записей
- SUM() - сумма
- AVG() - среднее значение
- MAX()/MIN() - максимальное/минимальное значение

Вложенные подзапросы:

sql

SELECT имя FROM сотрудники

WHERE отдел_id IN (SELECT id FROM отделы WHERE название = 'ИТ');

SQL-функции:

- Строковые: UPPER(), LOWER(), SUBSTRING()
 - Числовые: ROUND(), ABS(), MOD()
 - Дата/время: NOW(), DATE(), YEAR()
-

16. Вычисляемые поля и группировка

Вычисляемые поля:

sql

SELECT название, цена, количество, цена * количество AS сумма

FROM товары;

Группировка данных:

sql

SELECT отдел, AVG(зарплата) AS средняя_зарплата

FROM сотрудники

GROUP BY отдел

HAVING AVG(зарплата) > 50000;

Использование агрегатных функций с GROUP BY:

- Группировка по одному или нескольким столбцам
- Агрегатные функции применяются к каждой группе
- HAVING для фильтрации групп (аналог WHERE для строк)