ТЕМА 1  
Конспект по теме "Проектирование баз данных (БД)"

1. Проектирование БД

Проектирование БД — процесс создания структуры базы данных, который включает:

* Определение требований к данным.
* Организацию данных в логическую модель.
* Оптимизацию для эффективного хранения и обработки.

2. Стадии проектирования БД

1. Сбор и анализ требований – изучение предметной области.
2. Концептуальное проектирование – создание ER-диаграмм.
3. Логическое проектирование – преобразование модели в схему (реляционная, NoSQL и др.).
4. Физическое проектирование – оптимизация для СУБД (индексы, партиционирование).
5. Реализация и тестирование – создание БД, наполнение данными, проверка.

3. Основные задачи проектирования БД

* Обеспечение целостности данных.
* Минимизация избыточности.
* Оптимизация производительности.
* Удобство поддержки и масштабирования.

4. Кто занимается проектированием БД?

* Аналитики данных – сбор требований.
* Архитекторы БД – проектирование структуры.
* Разработчики БД – реализация и оптимизация.
* Администраторы БД (DBA) – настройка и обслуживание.

5. Стек разработчика БД (проектировщика)

* Языки: SQL, PL/SQL, T-SQL.
* Инструменты: ERwin, Lucidchart, Draw.io, MySQL Workbench, pgAdmin.
* СУБД: PostgreSQL, MySQL, Oracle, MS SQL Server, MongoDB (для NoSQL).
* Методологии: Нормализация, ER-моделирование.

Концептуальная модель и ER-диаграммы

6. Концептуальная модель

Абстрактное описание данных без привязки к СУБД. Основные элементы:

* Сущность (Entity) – объект предметной области (например, Студент).
* Атрибут – свойство сущности (например, Имя, ID).
* Связь (Relationship) – взаимодействие сущностей (например, Студент учится в Группе).

7. Терминология

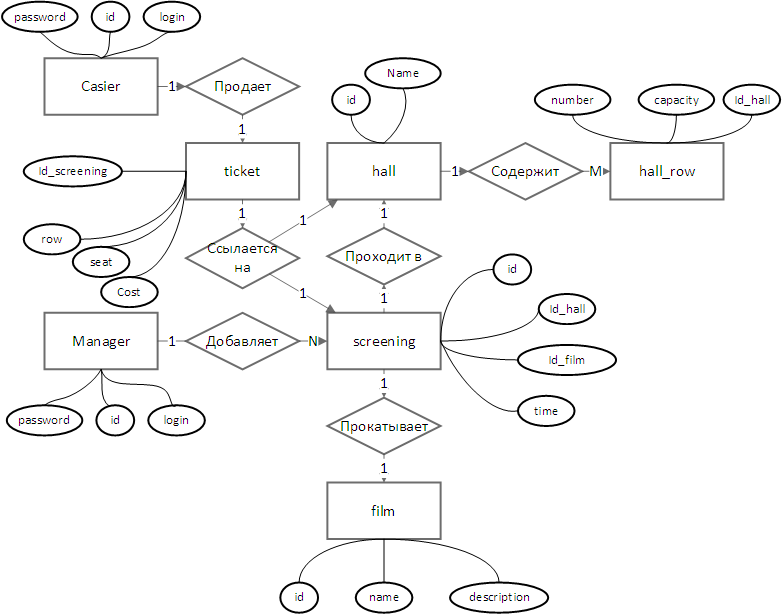
* Сущность – таблица в БД (например, Книга).
* Сильная (правильная) сущность – существует независимо (например, Автор).
* Слабая сущность – зависит от сильной (например, Глава книги без книги не существует).
* Конкретная сущность – имеет физические экземпляры (например, Сотрудник).
* Связь – отношение между сущностями (например, Работает в между Сотрудник и Отдел).

8. Виды связей в ER-диаграммах

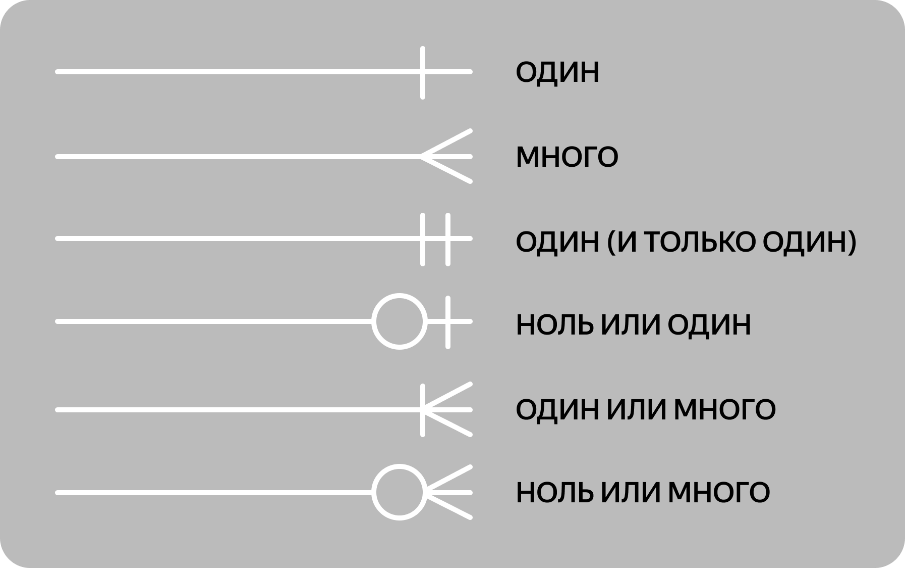
* 1:1 (Один к одному) – например, Паспорт ↔ Человек.
* 1:N (Один ко многим) – например, Автор → Книги.
* M:N (Многие ко многим) – например, Студенты ↔ Курсы (реализуется через связующую таблицу).

9. Нотации ER-диаграмм

* Нотация Чена:
  + Сущности – прямоугольники.
  + Связи – ромбы.
  + Атрибуты – овалы.



* Нотация Мартина (Crow’s Foot):
  + Сущности – прямоугольники.
  + Связи – линии с обозначениями (| – один, > – многие).



10. Этапы построения концептуальной модели

1. Выделение сущностей (например, Заказ, Товар).
2. Определение атрибутов (для Заказ: номер, дата).
3. Установление связей (Заказ содержит Товар).
4. Определение типов связей (1:N, M:N).
5. Проверка на избыточность и нормализация.

Итог: Проектирование БД – ключевой этап разработки, требующий понимания предметной области, умения моделировать данные и выбирать оптимальные решения для хранения информации.

ТЕМА 2

Конспект по теме "Аномалии в данных. Нормализация реляционной БД"

1. Аномалии в данных

Аномалии — это проблемы, возникающие при некорректной структуре БД, ведущие к избыточности, противоречивости и сложностям в поддержке данных.

Типы аномалий:

1. Аномалии вставки (Insertion Anomaly)
   * Невозможность добавить данные без наличия связанных данных.
   * Пример: Нельзя добавить нового студента, пока он не записан на курс.
2. Аномалии обновления (Update Anomaly)
   * Необходимость изменять одни и те же данные в нескольких местах.
   * Пример: Если у преподавателя меняется имя, его нужно обновить во всех связанных записях.
3. Аномалии удаления (Deletion Anomaly)
   * Потеря данных при удалении связанных записей.
   * Пример: Удаление последнего студента с курса может случайно удалить и сам курс.

2. Нормализация реляционной БД

Нормализация — процесс устранения аномалий путем разбиения таблиц на более простые и логически связанные.

Задачи нормализации:

✅ Уменьшение избыточности данных.  
✅ Обеспечение целостности данных.  
✅ Упрощение поддержки и модификации БД.

3. Нормальные формы (1НФ, 2НФ, 3НФ, BCNF, 4НФ, 5НФ)

1. Первая нормальная форма (1НФ)

Условия:

* Все атрибуты атомарны (неделимы).
* Нет повторяющихся групп.

Пример нарушения 1НФ:

| Student\_ID | Name | Courses |
| --- | --- | --- |
| 1 | Alex | Math, Physics |

Исправление (1НФ):

| Student\_ID | Name | Course |
| --- | --- | --- |
| 1 | Alex | Math |
| 1 | Alex | Physics |

2. Вторая нормальная форма (2НФ)

Условия:

* Таблица уже в 1НФ.
* Нет частичных зависимостей (все неключевые атрибуты зависят от всего первичного ключа).

Пример нарушения 2НФ:

| Order\_ID | Product\_ID | Product\_Name | Quantity |
| --- | --- | --- | --- |
| 1001 | P001 | Laptop | 2 |

🔹 Проблема: Product\_Name зависит только от Product\_ID, а не от всего ключа (Order\_ID, Product\_ID).

Исправление (2НФ):  
Таблица Orders:

| Order\_ID | Product\_ID | Quantity |
| --- | --- | --- |
| 1001 | P001 | 2 |

Таблица Products:

| Product\_ID | Product\_Name |
| --- | --- |
| P001 | Laptop |

3. Третья нормальная форма (3НФ)

Условия:

* Таблица уже в 2НФ.
* Нет транзитивных зависимостей (неключевые атрибуты не зависят от других неключевых атрибутов).

Пример нарушения 3НФ:

| Student\_ID | Name | Department | HOD |
| --- | --- | --- | --- |
| 101 | Alice | CS | Dr. Smith |

🔹 Проблема: HOD (зав. кафедрой) зависит от Department, а не напрямую от Student\_ID.

Исправление (3НФ):  
Таблица Students:

| Student\_ID | Name | Department |
| --- | --- | --- |
| 101 | Alice | CS |

Таблица Departments:

| Department | HOD |
| --- | --- |
| CS | Dr. Smith |

4. Нормальная форма Бойса-Кодда (BCNF)

Условия:

* Уже в 3НФ.
* В таблице нет зависимостей, где детерминант (определяющий атрибут) не является потенциальным ключом.

Пример нарушения BCNF:

| Course | Lecturer | Student |
| --- | --- | --- |
| Math | Prof. A | Alice |

🔹 Проблема: Если лектор ведет только один курс, то Course → Lecturer, но Lecturer не ключ.

Исправление (BCNF):  
Таблица Student\_Courses:

| Student | Course |
| --- | --- |
| Alice | Math |

Таблица Course\_Lecturers:

| Course | Lecturer |
| --- | --- |
| Math | Prof. A |

4. Виды нормализации (этапы с примерами)

1. Денормализация (обратный процесс) – намеренное дублирование данных для ускорения запросов.
2. Постепенная нормализация (1НФ → 2НФ → 3НФ → BCNF → 4НФ → 5НФ).

Пример полной нормализации (1НФ → 3НФ):  
📌 Исходная таблица (аномалии):

| Order\_ID | Customer | Product | Price | Category |
| --- | --- | --- | --- | --- |
| 1001 | Alice | Laptop | 1000 | Electronics |

📌 После 1НФ: (разделение составных данных)  
📌 После 2НФ: (отделение Products от Orders)  
📌 После 3НФ: (отделение Categories от Products)

Итоговые таблицы:

* Orders (Order\_ID, Customer\_ID)
* Order\_Details (Order\_ID, Product\_ID, Price)
* Products (Product\_ID, Product\_Name, Category\_ID)
* Categories (Category\_ID, Category\_Name)

Вывод

Нормализация устраняет аномалии, но может усложнять запросы. Часто применяют частичную нормализацию, оставляя некоторые дублирования для производительности.