

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

“Санкт-Петербургский национальный исследовательский университет  
информационных технологий механики и оптики”

Мегафакультет: трансляционных информационных технологий

Факультет: информационных технологий и программирования

**Лабораторная работа №1**

**По дисциплине: “Проектирование баз данных”**

**Тема: “Проектирование архитектуры БД”**

Выполнила студент группы №М3216:

*Шевцов Роман Сергеевич*

**САНКТ-ПЕТЕРБУРГ**

**2025**

### **Задачи:**

1. Провести анализ функционала сайта или портала в выбранной предметной области с позиции работы с данными, выделить сущности, их атрибуты и связи между сущностями.
2. Спроектировать архитектуру БД для выбранной темы в виде модели сущность-связь (ERM) в нотации Мартина (Crow's Foot, «воронья лапка»).
3. Преобразовать ERM в физическое представление (PDM) путём добавления нужных соединительных таблиц и внешних ключей – столбцов в существующие таблицы.
4. Проверить, что модель находится минимум в третьей нормальной форме (3НФ).

### **Порядок выполнения работы:**

1. Выбрать один из предложенных ниже вариантов предметных областей или согласовать с преподавателем другой вариант.
2. Провести анализ функционала сайта или портала в выбранной предметной области с позиции работы с данными, выделить сущности, их атрибуты и связи.
3. Спроектировать архитектуру БД для выбранной темы в виде ERM в нотации Мартина.
4. Преобразовать ERM в PDM, указать правила, которые использовались при преобразовании (для связей один-ко-многим, многие-ко-многим).
5. Предоставить отчёт, включить в него следующие данные:
  - краткое описание предметной области и портала, в том числе пояснение, охватывает ли моделирование все данные или только те, которые относятся к некоторым бизнес-процессам (указать к каким);
  - ERM в нотации Мартина, PDM, процесс преобразования;
  - обоснование нахождения модели данных в 3НФ (или выше по желанию).

### **Требования к архитектуре БД:**

1. Минимум 3 НФ.
2. Минимум 6 сущностей, максимум 10 сущностей.
3. Должны быть указаны ключи.
4. Должны быть разные связи, в том числе, один-ко-многим и многие-ко-многим.
5. Должны быть указаны типы атрибутов.
6. Должна существовать сущность Пользователь (необходимо для следующих лабораторных).

### **Решение:**

Выбранная тема: оператор сотовой связи <https://spb.shop.megafon.ru/>

## Краткое описание предметной области и портала:

### Предметная область

Предметной областью для данной лабораторной работы является оператор сотовой связи. Операторы сотовой связи предоставляют услуги мобильной связи, интернета, SMS, а также дополнительные услуги, такие как роуминг, антивирусы, подписки на контент и т.д. Основные бизнес-процессы оператора включают:

- Управление клиентами (регистрация, авторизация, управление личными данными).
- Управление тарифами (создание, изменение, подключение тарифов).
- Управление услугами (подключение и отключение дополнительных услуг).
- Управление платежами (оплата услуг, пополнение счёта).
- Управление устройствами (учёт устройств, подключённых к сети оператора).

### Портал

В качестве примера портала был выбран сайт оператора сотовой связи Мегафон (<https://spb.shop.megafon.ru/>). Портал предоставляет клиентам возможность:

- Выбирать и подключать тарифы.
- Управлять дополнительными услугами.
- Оплачивать услуги и просматривать историю платежей.
- Просматривать информацию о подключённых устройствах.
- Получать информацию о текущих подписках и их статусе.

### Охват моделирования

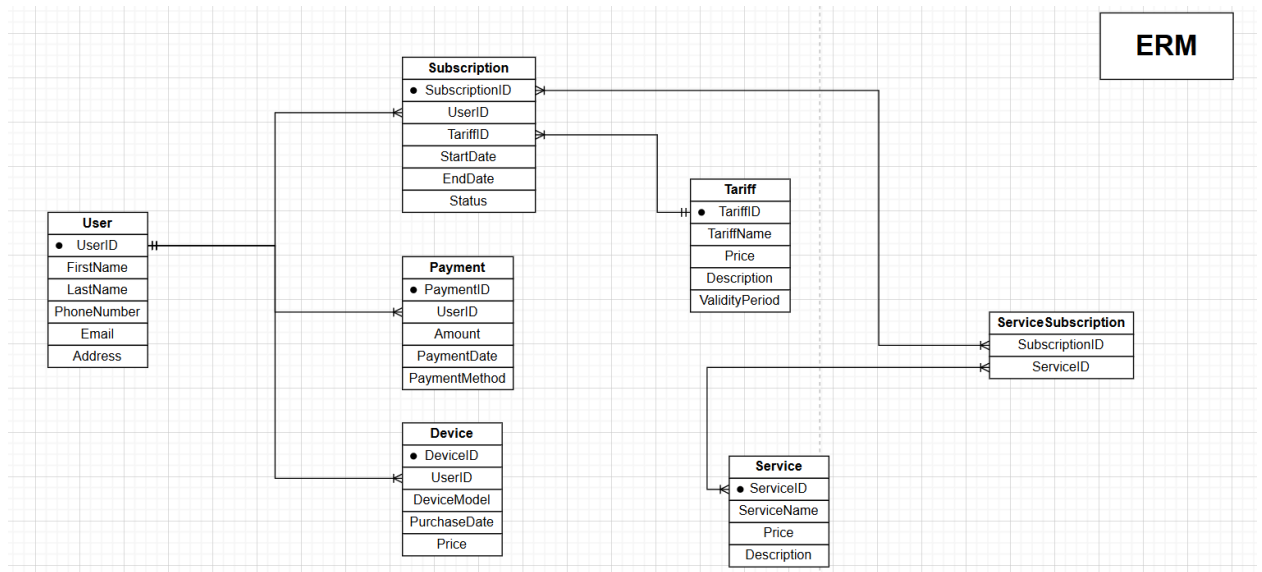
Моделирование охватывает основные данные и бизнес-процессы, связанные с управлением клиентами, тарифами, услугами, платежами и устройствами. В частности, модель включает:

1. Управление клиентами:
  - Хранение данных о пользователях (ФИО, номер телефона, email, адрес).
  - Учёт подключённых тарифов и услуг.
2. Управление тарифами:
  - Хранение информации о доступных тарифах (название, стоимость, описание, срок действия).
  - Подключение тарифов пользователями.
3. Управление услугами:
  - Хранение информации о дополнительных услугах (название, стоимость, описание).
  - Подключение услуг к подпискам пользователей.
4. Управление платежами:
  - Хранение информации о платежах пользователей (сумма, дата, способ оплаты).

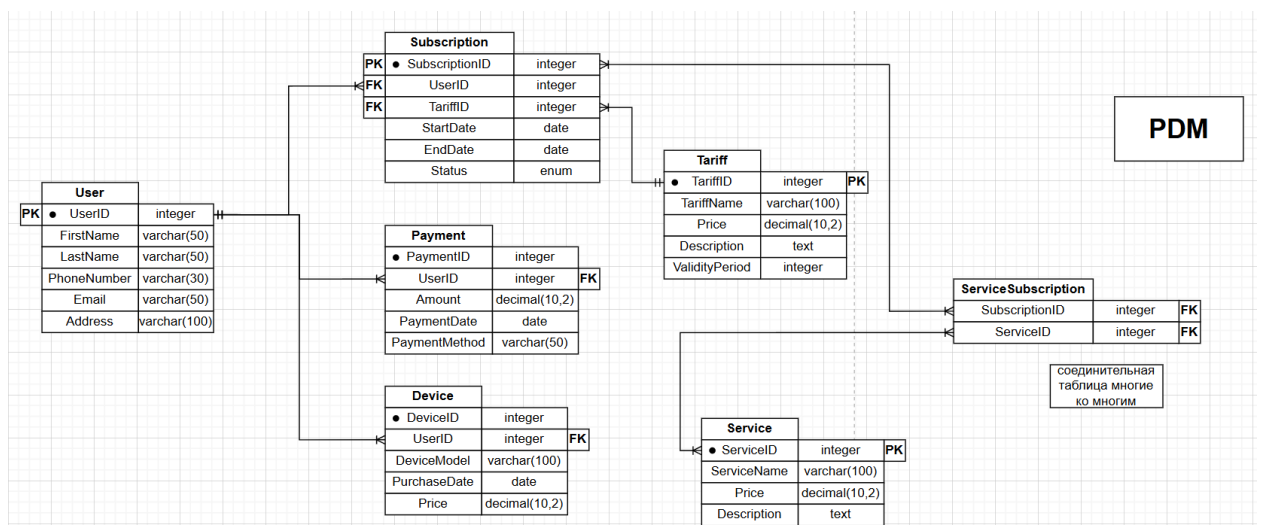
## 5. Управление устройствами:

- Учёт устройств, подключённых к сети оператора (модель, дата покупки, стоимость).

### ERM в нотации Мартина:



### PDM:



### Процесс преобразования ERM в PDM:

#### 1. Определение сущностей и их атрибутов

На первом этапе создается ERM-диаграмма (Entity-Relationship Model), которая отражает основные сущности предметной области, их атрибуты и связи между ними.

В моем случае были выделены следующие сущности:

- User – пользователь системы
- Subscription – подписка
- Tariff – тарифный план
- Service – услуга

- Payment – платеж
- Device – устройство
- ServiceSubscription – связывающая таблица для отношений "многие ко многим" между подписками и услугами

## 2. Определение первичных ключей (PK) и связей

На ERM-диаграмме связи между сущностями обозначены линиями, указывающими на тип отношений (1:1, M:N).

- Для каждого объекта был определен уникальный идентификатор (PK):
  - UserID для User
  - SubscriptionID для Subscription
  - TariffID для Tariff
  - ServiceID для Service
  - PaymentID для Payment
  - DeviceID для Device
- Для связи "многие ко многим" между Service и Subscription была создана промежуточная таблица ServiceSubscription, содержащая составной первичный ключ (SubscriptionID, ServiceID).

## 3. Добавление внешних ключей (FK)

Для сохранения связей между таблицами в PDM (Physical Data Model) были добавлены внешние ключи (FK):

- В Subscription → UserID (пользователь подписки)
- В Subscription → TariffID (тариф подписки)
- В Payment → UserID (чей платеж)
- В Device → UserID (чье устройство)
- В ServiceSubscription → SubscriptionID и ServiceID

## 4. Выбор типов данных

При переходе к PDM для каждого атрибута был определен конкретный тип данных в зависимости от его назначения:

- INTEGER – для идентификаторов (PK, FK)
- VARCHAR(N) – для строковых данных (имена, описания, статусы)
- DECIMAL(10,2) – для сумм платежей и стоимости тарифов
- DATE – для хранения дат (начало/окончание подписки, дата платежа)
- TEXT – для длинных описаний

## 5. Нормализация данных

Была проведена нормализация, чтобы:

- Избежать дублирования данных
- Оптимизировать структуру хранения
- Сохранить логические связи между сущностями

В результате получена PDM-диаграмма, полностью отражающая логическую модель ERM в виде структуры таблиц реляционной базы данных.

### **Обоснование нахождения модели данных в третьей нормальной форме (3НФ)**

При проектировании реляционной базы данных важно привести ее структуру к третьей нормальной форме (3НФ), чтобы минимизировать дублирование данных и избежать аномалий при их обновлении. Рассмотрим, как моя модель удовлетворяет требованиям 3НФ.

#### 1. Первая нормальная форма (1НФ)

Таблицы находятся в первой нормальной форме (1НФ), если:

- Все атрибуты содержат только атомарные (неделимые) значения.
- В таблицах нет повторяющихся групп данных или списков значений.

В моей модели каждая колонка содержит только одно значение (например, PhoneNumber в User содержит один номер, а не список номеров). Каждая таблица имеет уникальный первичный ключ (PK), идентифицирующий строки.

#### 2. Вторая нормальная форма (2НФ)

Таблицы находятся во второй нормальной форме (2НФ), если:

- Они соответствуют 1НФ.
- Все неключевые атрибуты зависят только от первичного ключа (PK), а не от его части.

В моей модели нет составных первичных ключей, кроме таблицы ServiceSubscription, где PK = (SubscriptionID, ServiceID). В таблицах нет частичных зависимостей, так как каждый атрибут зависит только от полного PK.

Пример:

В Subscription атрибуты StartDate, EndDate, Status зависят только от SubscriptionID, а не, например, от UserID или TariffID.

#### 3. Третья нормальная форма (3НФ)

Таблицы находятся в третьей нормальной форме (3НФ), если:

- Они соответствуют 2НФ.
- В них нет транзитивных зависимостей, то есть каждый неключевой атрибут зависит только от первичного ключа и не зависит от других неключевых атрибутов.

В моей модели отсутствуют транзитивные зависимости. Например, в User атрибут Email зависит только от UserID, а не от FirstName или LastName. В Subscription информация о тарифах (TariffName, Price) вынесена в отдельную таблицу Tariff, а не хранится в Subscription, что предотвращает дублирование данных.

### **Вывод**

Преобразование модели данных в 3НФ позволило:

✓ Избежать избыточности и дублирования данных.

- ✓ Уменьшить вероятность аномалий при обновлении и удалении записей.
- ✓ Обеспечить целостность данных за счет использования внешних ключей (FK).

Таким образом, предложенная реляционная модель соответствует третьей нормальной форме (3НФ) и является оптимальной для хранения данных.