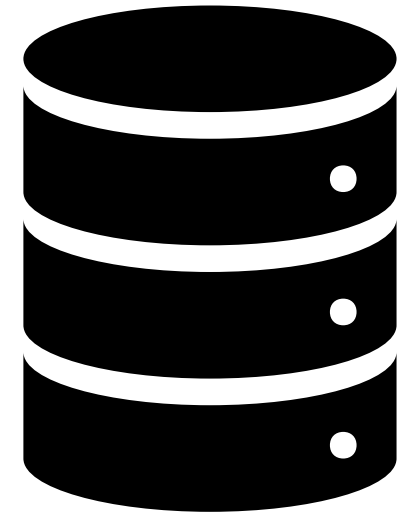


Базы данных

Лекция 4. Транзакции. Проектирование базы данных.



Меркурьева Надежда

✉ merkurievanad@gmail.com

📧 [@merkurievanad](https://www.instagram.com/merkurievanad)

ФПМИ МФТИ, 2021



ТРАНЗАКЦИЯ

Транзакция – группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными, гарантированно переводящая БД из одного непротиворечивого состояния в другое.



TCL (TRANSACTION CONTROL LANGUAGE)

- COMMIT
 - Применяет транзакцию, т.е. сохраняет изменения, произведенные в процессе выполнения транзакции
- ROLLBACK
 - Откатывает все изменения, произведенные в процессе выполнения транзакции
- SAVEPOINT
 - Создает так называемую точку останова

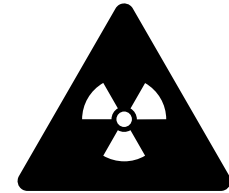


ТОЧКА ОСТАНОВА

- Точка останова – промежуточный участок в транзакции, на который можно откатиться в случае необходимости
 - Позволяет дробить транзакцию на части
 - Позволяет реализовать «вложенные» транзакции

СВОЙСТВА ТРАНЗАКЦИИ (ACID)

- **Atomicity** (Атомарность):
 - Выполнены либо все подоперации, либо никакие
- **Consistency** (Согласованность)
 - Каждая успешная транзакция фиксирует только допустимые результаты
- **Isolation** (Изолированность)
 - Параллельные транзакции не влияют на результаты друг друга
- **Durability** (Устойчивость)
 - Вне зависимости от сбоев системы результаты успешных транзакций сохраняются в системе



ATOMICITY (АТОМАРНОСТЬ)

- Никакая транзакция не будет зафиксирована в системе частично
- Выполнены либо все подоперации, либо никакая
- На практике одновременное и атомарное выполнение транзакций невозможно
- На практике «атомарность» реализуется с использованием «отката» (rollback)
- В процессе отката отменяются все уже произведенные операции

CONSISTENCY (СОГЛАСОВАННОСТЬ)

- Фиксируются только допустимые результаты операций
- Согласованность применяется не только в контексте БД.

Пример: банковские транзакции

- В ходе выполнения операции согласованность не требуется
- Вследствие атомарности промежуточная несогласованность остается скрытой

ISOLATION (ИЗОЛИРОВАННОСТЬ)

- Параллельные транзакции не оказывают влияния на друг друга
- В реальных БД полная изолированность не поддерживается
- Уровень изолированности – характеристика соответствия БД свойству изолированности

ПРОБЛЕМЫ ПОДДЕРЖКИ ИЗОЛИРОВАННОСТИ

Потерянное обновление

- Изменение одного блока данных несколькими транзакциями

«Грязное» чтение

- Чтение данных, измененных впоследствии откатившейся транзакцией

Неповторяющееся чтение

- Повторное чтение измененных данных одной и той же транзакцией

Чтение «фантомов»

- Взаимосвязанные критерии изменения данных двумя транзакциями

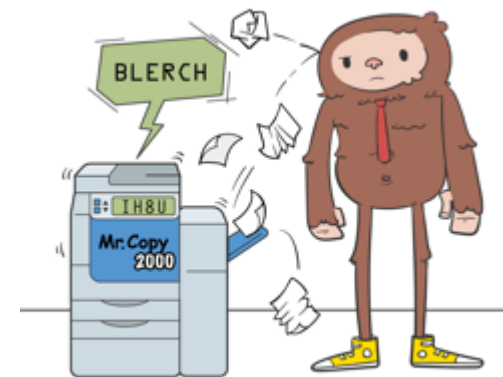
ПОТЕРЯННОЕ ОБНОВЛЕНИЕ

Транзакция 1	Транзакция 2
<pre>UPDATE table_1 SET attr_2 = attr_2 + 20 WHERE attr_1 = 1;</pre>	<pre>UPDATE table_1 SET attr_2 = attr_2 + 25 WHERE attr_1 = 1;</pre>

Что получим в итоге?

```
1.attr_2 = attr_2 + 20
2.attr_2 = attr_2 + 25
```

Результат однозначно не определен

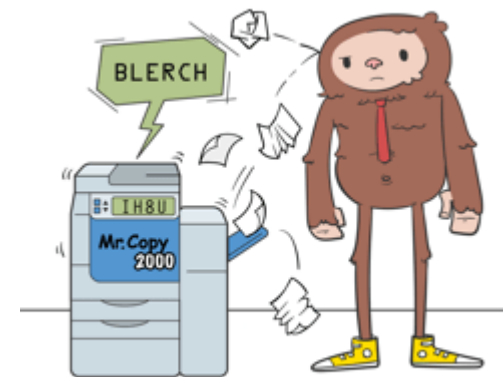


ПОТЕРЯННОЕ ОБНОВЛЕНИЕ

Транзакция 1	Транзакция 2
<pre>UPDATE table_1 SET attr_2 = attr_2 + 5000 WHERE attr_1 = 1;</pre>	<pre>UPDATE table_1 SET attr_2 = attr_2 + 50 WHERE attr_1 = 1;</pre>

- Вы пополняете свою карту на 5000 рублей
- Кто-то переводит вам 50 рублей

Итоговая сумма на вашем счету неизвестна



«ГРЯЗНОЕ» ЧТЕНИЕ

Транзакция 1	Транзакция 2
UPDATE table_1 SET attr_2 = attr_2 + 20 WHERE attr_1 = 1;	
	SELECT attr_2 FROM table_1 WHERE attr_1 = 1;
ROLLBACK WORK;	



«ГРЯЗНОЕ» ЧТЕНИЕ

Транзакция 1	Транзакция 2
UPDATE table_1 SET attr_2 = attr_2 + 20 WHERE attr_1 = 1;	
	SELECT attr_2 FROM table_1 WHERE attr_1 = 1;
ROLLBACK WORK;	

- Вы оплачиваете дорогостоящую покупку со счета родственника
- Ваш родственник проверяет остаток -> впадает в ступор
.....
- Списание не проходит, транзакция откатывается, но родственник уже в шоке



НЕПОВТОРЯЮЩЕЕСЯ ЧТЕНИЕ

Транзакция 1	Транзакция 2
	SELECT attr_2 FROM table_1 WHERE attr_1 = 1;
UPDATE table_1 SET attr_2 = attr_2 + 20 WHERE attr_1 = 1;	
COMMIT;	
	SELECT attr_2 FROM table_1 WHERE attr_1 = 1;



НЕПОВТОРЯЮЩЕЕСЯ ЧТЕНИЕ

Транзакция 1	Транзакция 2
	SELECT attr_2 FROM table_1 WHERE attr_1 = 1;
UPDATE table_1 SET attr_2 = attr_2 + 20 WHERE attr_1 = 1;	
COMMIT;	
	SELECT attr_2 FROM table_1 WHERE attr_1 = 1;

- Вы сформировали корзину на сайте и жмете на кнопку «Оплатить»
- Стоимость товаров на сайте увеличивается
- С карты списывается бОльшая сумма, чем вы ожидали



ЧТЕНИЕ ФАНТОМОВ

Транзакция 1	Транзакция 2
	SELECT <i>sum</i> (attr_2) FROM table_1;
INSERT INTO table_1 (attr_1, attr_2) VALUES (15, 20);	
COMMIT;	
	SELECT <i>sum</i> (attr_2) FROM table_1;



ЧТЕНИЕ ФАНТОМОВ

Транзакция 1	Транзакция 2
	SELECT <i>sum</i> (attr_2) FROM table_1;
INSERT INTO table_1 (attr_1, attr_2) VALUES (15, 20);	
COMMIT;	
	SELECT <i>sum</i> (attr_2) FROM table_1;

- Начальник попросил вас составить отчет трат общий по итогам текущего месяца и в разрезе недель того же месяца
- Вы рассчитываете общую сумму транзакций за месяц
- Происходит новое списание
- Вы рассчитываете сумму транзакций в разрезе недель → учитываете новую транзакцию → суммы не бьются



УРОВНИ ИЗОЛИРОВАННОСТИ ТРАНЗАКЦИЙ

- Read uncommitted (чтение незафиксированных данных)
- Read committed (чтение фиксированных данных)
- Repeatable read (повторяемость чтения)
- Serializable (упорядочиваемость)

READ UNCOMMITTED

- Первый уровень изоляции
 - Гарантирует отсутствие потерянных обновлений
 - Итоговое значение – результат выполнения каждой транзакции
 - Возможно считывание незафиксированных изменений
-
- Данные блокируются на время внесения изменений
 - На время чтения данных блокировка отсутствует

READ COMMITTED

- Второй уровень изоляции
- Используется в большей части СУБД
- Защита от «грязного» чтения
- В процессе выполнения одна из транзакций успешно завершается, тогда остальные работают с измененными данными
- Реализация IRL на усмотрение разработчиков СУБД:
 - Блокирование читаемых и изменяемых данных
 - Сохранение нескольких версий параллельно изменяемых строк

READ COMMITTED: БЛОКИРОВКА

- Блокирование читаемых и изменяемых данных
- *Пишущая* транзакция блокирует данные для *читающих* транзакций уровня read committed или выше
- Отсутствует «грязное» чтение
- Возможно неповторяющееся чтение

READ COMMITTED: ВЕРСИОННОСТЬ

- Сохранение нескольких версий параллельно изменяемых строк
- СУБД создает новую версию строки, с которой работает «изменяющие данные» транзакция
- Для «читающей» транзакции доступна последняя **зафиксированная** версия

REPEATABLE READ

- Третий уровень изоляции
- Читающая транзакция игнорирует изменения в данных, которые были ей ранее прочитаны
- Никакая транзакция не может изменять данные, читаемые текущей транзакцией, пока чтение не завершено
- Спасает от эффекта неповторяющегося чтения

SERIALIZABLE

- Четвертый (самый высокий) уровень изоляции
- Транзакции **полностью** изолированы друг от друга
- Параллельных транзакций как будто бы не существует вовсе
- Транзакции не подвержены эффекту «фантомного чтения»

УРОВНИ ИЗОЛЯЦИИ

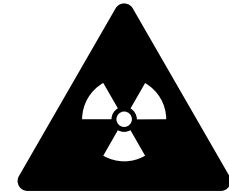
	Read uncommitted	Read committed	Repeatable read	Serializable
Потерянное обновление	+	+	+	+
«Грязное» чтение	-	+	+	+
Неповторяющееся чтение	-	-	+	+
Фантомное чтение	-	-	-	+

DURABILITY (УСТОЙЧИВОСТЬ)

- Вне зависимости от сбоев системы результаты успешных транзакций сохраняются в системе
- Если пользователь получил подтверждение об успешности транзакции, гарантируется сохранность результатов

СВОЙСТВА ТРАНЗАКЦИИ (ACID)

- **Atomicity** (Атомарность):
 - Выполнены либо все подоперации, либо никакие
- **Consistency** (Согласованность)
 - Каждая успешная транзакция фиксирует только допустимые результаты
- **Isolation** (Изолированность)
 - Параллельные транзакции не влияют на результаты друг друга
- **Durability** (Устойчивость)
 - Вне зависимости от сбоев системы результаты успешных транзакций сохраняются в системе



TRANSACTION CONTROL LANGUAGE (TCL)

- `BEGIN TRANSACTION transaction_mode [, ...]`
 - `ISOLATION LEVEL { SERIALIZABLE | REPEATABLE READ | READ COMMITTED | READ UNCOMMITTED }`
 - `READ WRITE | READ ONLY`
 - `[NOT] DEFERRABLE`
- `BEGIN / START`
 - `COMMIT`
 - `ROLLBACK`
- `SAVEPOINT name`
 - `ROLLBACK TO SAVEPOINT name`
 - `RELEASE SAVEPOINT name`

ПРИМЕР ИСПОЛЬЗОВАНИЯ

```
BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED;  
    INSERT INTO table1 VALUES (1);  
    SAVEPOINT my_savepoint;  
    INSERT INTO table1 VALUES (2);  
    ROLLBACK TO SAVEPOINT my_savepoint;  
    INSERT INTO table1 VALUES (3);  
COMMIT;
```



ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

– процесс создания детализированной модели данных* БД, а также необходимых ограничений целостности.

- Модель данных – абстрактная модель, которая:
 - Организует элементы данных
 - Описывает, как они взаимодействуют друг с другом
 - Описывает, как они взаимодействуют с объектами внешнего мира

ОСНОВНЫЕ ЗАДАЧИ ПРОЕКТИРОВАНИЯ БД



хранение всей
необходимой
информации



получение нужных
данных по запросу



сокращение
избыточности и
дублирования данных



обеспечение
целостности базы
данных

ОСНОВНЫЕ ШАГИ ПРОЕКТИРОВАНИЯ

Определение предметной области и данных, которые будут храниться в БД

Определение взаимосвязей между различными элементами данных

Наложение логической структуры на данные на основе определенных соотношений

Создание спроектированной БД с учетом особенностей используемой СУБД

ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ

Концептуальное проектирование
(инфологическое)

Логическое проектирование
(дatalogическое)

Физическое проектирование



ШАГ 1. ОПРЕДЕЛЕНИЕ ДАННЫХ

- Ожидание:
 - Найти эксперта по проектированию БД
 - Найти эксперта в предметной области БД
 - Делегировать задачу этим людям
- Реальность:
 - Разобраться во всем самостоятельно



ШАГ 1. ОПРЕДЕЛЕНИЕ ДАННЫХ

- Какую предметную область описываем?
 - Полная база товаров магазинов «Пятерочка»
 - Полная клиентская база банка «Тинькофф»
 - Учебные процессы «МФТИ»
 - И т.д.

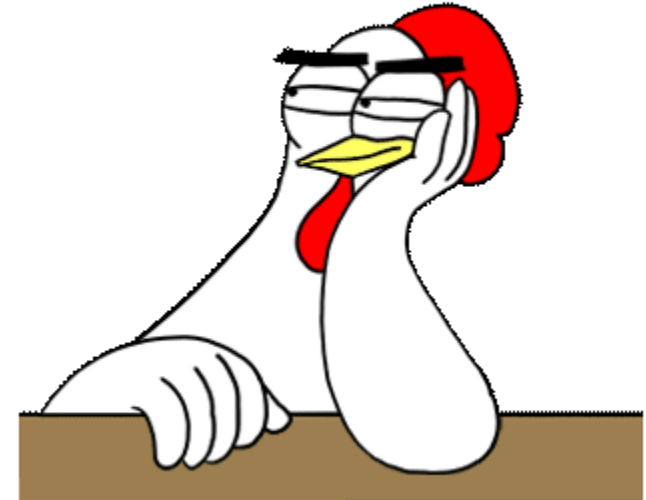
ШАГ 1. ОПРЕДЕЛЕНИЕ ДАННЫХ

- Как будем описывать? Как будем детализировать?
 - Полная база товаров магазина «Пятерочка»:
 - Каталог товаров
 - Каталог складов
 - Актуальные характеристики наполнения склада
 - Планы поставок
 - И т.д.



ШАГ 1. ОПРЕДЕЛЕНИЕ ДАННЫХ

- Как будем описывать? Как будем детализировать?
- Полная клиентская база банка «Тинькофф»:
 - Клиент
 - Дебетовые продукты клиента
 - Кредитные продукты клиента
 - Транзакции
 - И т.д.



ШАГ 1. ОПРЕДЕЛЕНИЕ ДАННЫХ

- Как будем описывать? Как будем детализировать?
 - Учебные процессы «МФТИ»
 - Студент
 - Группа
 - Преподаватель
 - Расписание пар
 - Расписание экзаменов
 - Распределение аудиторий
 - И т.д.



КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Концептуальная модель данных – описание основных объектов и отношений между ними.

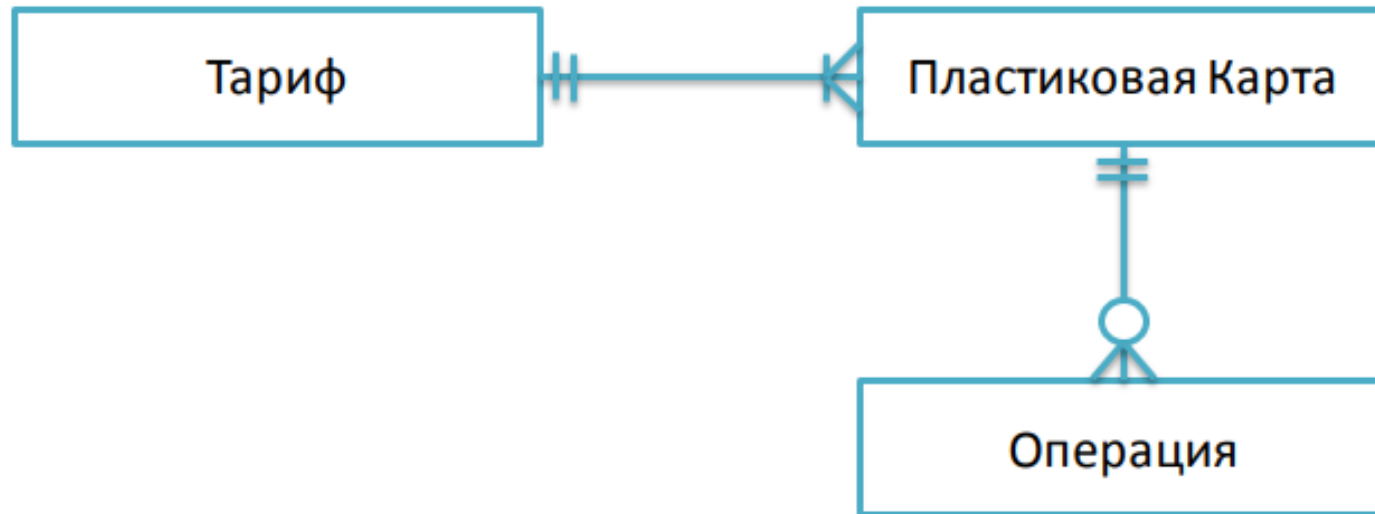
- Про описание объектов поговорили. Что насчет связей?

ER-МОДЕЛЬ

ER-модель (entity-relationship model) – модель данных, позволяющая описывать концептуальные и логические схемы.

- Задаются:
 - Сущности
 - Связи

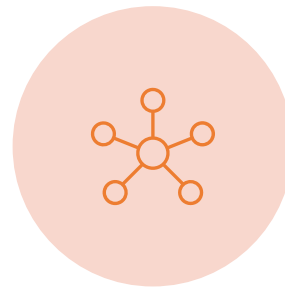
ER-МОДЕЛЬ



ШАГ 2. ОПРЕДЕЛЕНИЕ ВЗАИМОСВЯЗЕЙ



Определили
предметную область и
выделили сущности



Необходимо установить
взаимосвязи

НОТАЦИЯ «СУЩНОСТЬ»

Сущность

НОТАЦИЯ «ВОРОНЬЯ ЛАПКА»

В оригинале Crow's Foot notation

✓ Многие (строго больше одного)



✓ Один и только один



✓ Один или ноль



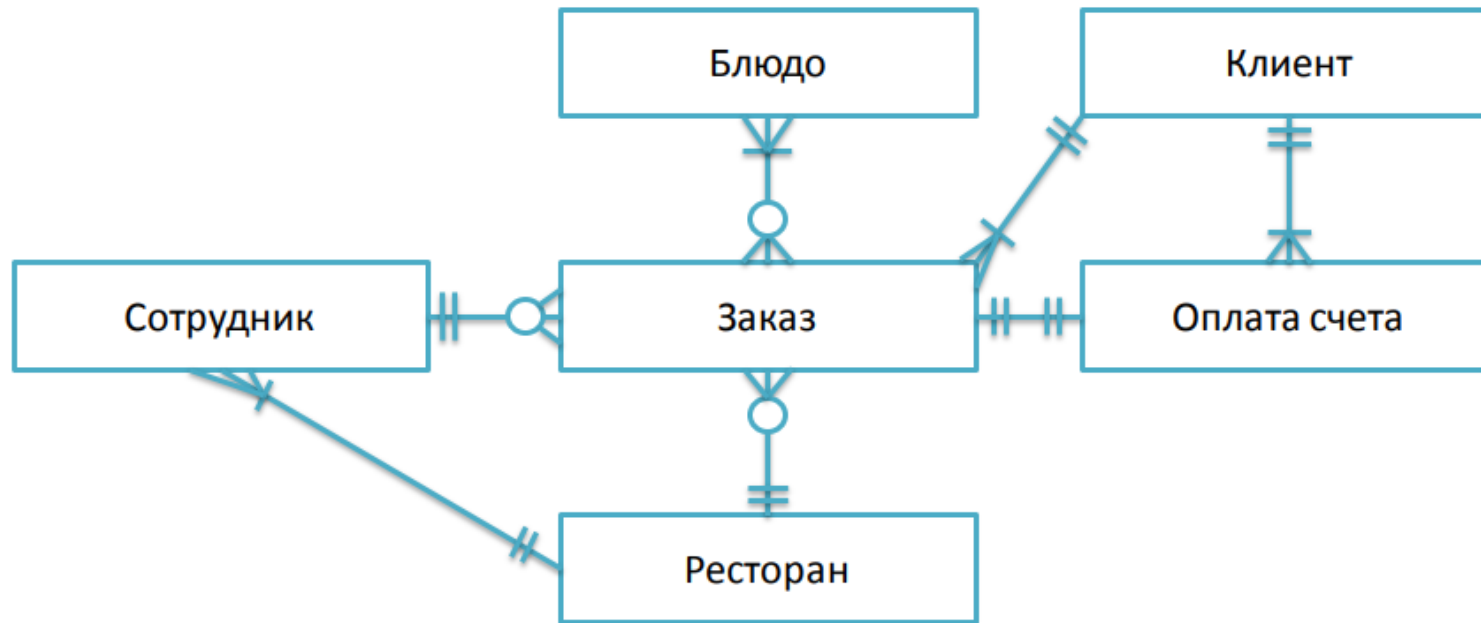
✓ Многие или один



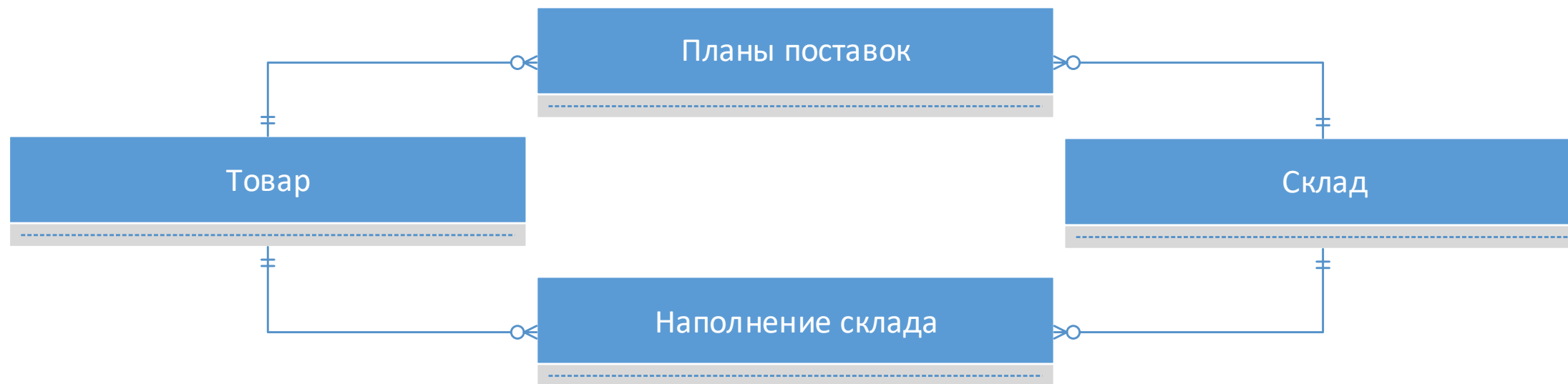
✓ Многие или один или ноль



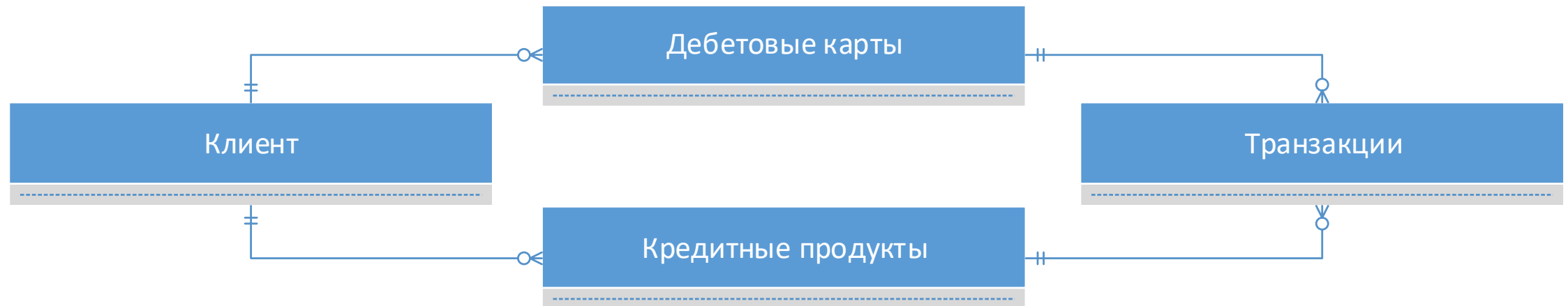
КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ БД РЕСТОРАНА



ПРИМЕРЫ ER-ДИАГРАММ: «ПЯТЕРОЧКА»



ПРИМЕРЫ ER-ДИАГРАММ: «ТИНЬКОФФ»



ШАГ 3. СОЗДАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

- Уже имеем концептуальную схему
- Хотим получить больший уровень детализации:
 - Уточнение атрибутивного состава
 - Уточнение ограничений, накладываемых на атрибутивный состав
 - Нормализация отношений
 - В некоторых случаях допускается уточнение типа атрибута

ПОТЕНЦИАЛЬНЫЙ КЛЮЧ

– это подмножество атрибутов отношения, удовлетворяющее требованиям уникальности и минимальности

- *Уникальность*: нет и не может быть двух кортежей данного отношения, в которых значения этого подмножества атрибутов совпадают
- *Минимальность*: в составе потенциального ключа отсутствует меньшее подмножество атрибутов, удовлетворяющее условию уникальности

ПОТЕНЦИАЛЬНЫЙ КЛЮЧ

- Из свойства отношения: потенциальный ключ существует **всегда**, даже если он включает **все** атрибуты отношения
- Допустимо наличие нескольких потенциальных ключей в отношении

КЛАССИФИКАЦИЯ ПО ПРИЗНАКУ ОБЩНОСТИ



ПЕРВИЧНЫЙ КЛЮЧ

– это один из потенциальных ключей отношения, выбранный в качестве основного (Primary key, PK)

- Если в отношении имеется лишь один потенциальный ключ, он и будет первичным ключом
- Если потенциальных ключей несколько, то:
 - Один из них выбирается в качестве первичного
 - Остальные ключи называются ***альтернативными***
- Недопустимо отсутствие значения

ПЕРВИЧНЫЙ КЛЮЧ

- Теоретически, все потенциальные ключи одинаково пригодны для использования в качестве первичного ключа
- На практике, используют тот потенциальный ключ, который:
 - Занимает меньше места при хранении
 - Не утратит свою уникальность со временем

СУРРОГАТНЫЙ КЛЮЧ

– это дополнительное служебное поле, которое добавляется к уже имеющимся информационным полям таблицы, единственное предназначение которое – служить первичным ключом

- Значение такого поля генерируется искусственно
- Не стоит искать в нем какой-то глубинный смысл
- Ключ, который основан на уже существующем поле, называется **естественным**
- Ключ, который основан на естественном ключе путем добавления дополнительного поля, называется **интеллектуальным**

СУРРОГАТНЫЙ КЛЮЧ

- Обычно суррогатный ключ – числовое поле
- Значения суррогатного ключа генерируется арифметической прогрессией с шагом 1
- В ряде СУБД существует специальный тип данных, автоматически генерирующий такую последовательность:
 - PostgreSQL – SERIAL
 - MySQL – AUTO_INCREMENT

ДОСТОИНСТВА СУРРОГАТНЫХ КЛЮЧЕЙ

- **Неизменность:** заполнили одним значением раз и навсегда (кроме экстраординарных ситуаций)
- **Гарантированная уникальность:** т.к. значение генерируется автоинкрементом, повторение значений исключено
- **Гибкость:** т.к. такой ключ не несет никакой информативной нагрузки, его можно свободно заменить
- **Проще программировать:** позволяет не завязывать на структуре конкретной БД. Особенно удобно для языков со статической типизацией
- **Эффективность:** удобнее при создании ссылок на другие таблицы

НЕДОСТАТКИ СУРРОГАТНЫХ КЛЮЧЕЙ

- **Уязвимость генераторов:** по номерам ключей возможно узнать число новых записей за определенный период времени
- **Неинформативность:** усложняется ручная проверка БД
- **Склоняет администратора пропустить нормализацию:** вместо того, чтобы разбить отношение на несколько отношений и аккуратно учесть все связи, велик соблазн просто создать суррогатный ключ
- **Вопросы оптимизации:** необходимость поддержания и суррогатного, и естественного ключей (об этом поговорим на лекции 6)
- **Невольная привязка разработчика к поведению генератора ключей в конкретной СУБД**

ВНЕШНИЙ КЛЮЧ

Пусть R_1 и R_2 – две переменные отношения, не обязательно различные. **Внешним ключом FK** в R_2 является подмножество атрибутов переменной R_2 такое, что выполняются следующие требования:

- В переменной отношения R_1 имеется потенциальный ключ РК такой, что РК и FK совпадают с точностью до переименования атрибутов
- В любой момент времени каждое значение FK в текущем значении R_2 идентично значению РК в некотором кортеже в текущем значении R_1 . Иными словами, в любой момент времени множество всех значений FK в R_2 является подмножеством значений РК в R_1 .

ВНЕШНИЙ КЛЮЧ

- Отношение R_1 , содержащее потенциальный ключ, называется **главным, целевым** или **родительским**
- Отношение R_2 , содержащее внешний ключ, называется **подчиненным** или **дочерним**

ВНЕШНИЙ КЛЮЧ

ID (PK)	CITY_NM
1	Москва
2	Санкт-Петербург
3	Владивосток

ID (PK)	STREET_NM	CITY_ID (FK)
181	Малая Бронная	1
182	Тверской бульвар	1
183	Невский проспект	2
184	Пушкинская	2
185	Светланская	3
186	Пушкинская	3

НОРМАЛЬНАЯ ФОРМА

- **Нормальная форма** — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных
- Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение
- Приведение БД к нормальной форме – нормализация
 - Каждая следующая форма включает в себя ограничения предыдущих

НОРМАЛИЗАЦИЯ БД

- Предназначена:
 - Минимизация логической избыточности
 - Уменьшение потенциальной противоречивости
- Не предназначена:
 - Уменьшение / увеличение производительности БД
 - Уменьшение / увеличение физического объема БД

НОРМАЛИЗАЦИЯ БД

- Исключение некоторых типов избыточности
- Устранение некоторых аномалий* обновления
- Разработка проекта БД, который является:
 - Качественным представлением реального мира
 - Интуитивно понятен
 - Легко расширяем в дальнейшем
- Упрощение процедуры применения необходимых ограничений целостности

АНОМАЛИИ ДАННЫХ

- Ситуация в таблице БД такая, что:
 - Существенно осложнена работа с БД
 - В БД присутствует противоречия
- Причина:
 - Излишнее дублирование данных в таблице

АНОМАЛИИ МОДИФИКАЦИИ

- Изменение данных одной записи влечет за собой необходимость изменения аналогичных данных еще некоторых записей

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

- Хотим изменить адрес поставщика 1:
 - Придется менять адрес во всех строках

АНОМАЛИИ УДАЛЕНИЯ

- Удаление определенных записей несет потерю информации, которую удалять не хотели

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

- Хотим удалить записи о поставках от поставщика 2:
 - Теряем всю информации о поставщике 2, включая его адрес

АНОМАЛИИ ДОБАВЛЕНИЯ

- Не можем добавить новую запись, если неизвестны значения первичных ключей

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

- Заключили контракт с поставщиком 3:
 - Не можем добавить информацию о нем в таблицу, т.к. еще не было поставок

НОРМАЛИЗАЦИЯ БД

- Нормализация БД производится за счет декомпозиции отношения
- Декомпозиция – разложение исходной переменной отношения на несколько эквивалентных
- Декомпозиция обратна соединению
- Декомпозиция называется **декомпозицией без потерь** или **правильной**, если она обратима

НОРМАЛЬНЫЕ ФОРМЫ

- Первая нормальная форма (1NF)
- Вторая нормальная форма (2NF)
- Третья нормальная форма (3NF)
- Нормальная форма Бойса-Кодда (BCNF)
- Четвертая нормальная форма (4NF)
- Пятая нормальная форма / Нормальная форма проекции-соединения (5NF / PJNF)
- Доменно-ключевая нормальная форма (DKNF)
- Шестая нормальная форма (6NF)

НОРМАЛЬНЫЕ ФОРМЫ

- **Первая нормальная форма (1NF)**
- **Вторая нормальная форма (2NF)**
- **Третья нормальная форма (3NF)**
- **Нормальная форма Бойса-Кодда (BCNF)**
- **Четвертая нормальная форма (4NF)**
- **Пятая нормальная форма / Нормальная форма проекции-соединения (5NF / PJNF)**
- **Доменно-ключевая нормальная форма (DKNF)**
- **Шестая нормальная форма (6NF)**

ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА

- Переменная отношения находится в ***первой нормальной форме*** тогда и только тогда, когда значения всех атрибутов отношения атомарны.

ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА

- Переменная отношения находится в ***первой нормальной форме*** тогда и только тогда, когда значения всех атрибутов отношения атомарны.
- Отношение находится в 1NF, если все его атрибуты являются простыми. Все используемые домены содержат только скалярные значения.

ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА

Семинарист	Группа
Халяпов Александр	911, 921, 924
Меркурьева Надежда	912, 932
Мавлютов Максим	922, 923
Лукьянов Денис	925, 926
Митюрин Максим	927
Тюрюмина Элла	931, 952
Роздухова Нина	951

Таблица не находится в 1NF, т.к. все группы каждого семинариста записаны в одной ячейке

ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА

Семинарист	Группа
Халяпов Александр	911
Меркурьева Надежда	912
Халяпов Александр	921
Мавлютов Максим	922
Мавлютов Максим	923
Халяпов Александр	924
Лукьянов Денис	925
Лукьянов Денис	926
Митюрин Максим	927
Тюрюмина Элла	931
Меркурьева Надежда	932
Роздухова Нина	951
Тюрюмина Элла	952

ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

- Переменная отношения находится во **второй нормальной форме** тогда и только тогда, когда она находится в первой нормальной форме, и каждый неключевой атрибут минимально функционально зависит от потенциального ключа
- **Функциональная зависимость** между множествами атрибутов X и Y означает, что для любого допустимого набора кортежей в данном отношении верно следующее: если два кортежа совпадают по значению X , то они совпадают по значению Y
- **Минимальная функциональная зависимость** означает, что в составе первичного ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость

ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

- Любая переменная отношения, находящаяся в 1NF, но не находящаяся в 2NF, может быть приведена к набору переменных отношений, находящихся в 2NF
- В результате декомпозиции получим набор проекций исходной переменной отношения, причем обратимый

ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

Название группы	Название CD-диска	Название песни	Автор слов	Композитор
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

Название группы	Название CD-диска	Название песни	Автор слов	Композитор
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

Данная таблица находится в 1NF, но не во 2NF, т.к. автор слов и композитор зависят только от полей «Название группы» и «Название песни», т.е. от того, что песня включена на другой CD-диск, значения этих полей никак не изменятся

ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

Название группы	Название CD-диска	Название песни
Scorpions	World Wide Live	Countdown
Scorpions	World Wide Live	Coming Home
Scorpions	World Wide Live	Blackout
Scorpions	Blackout	Blackout
The Big City	Blackout	Blackout

Название группы	Название песни	Автор слов	Композитор
Scorpions	Countdown	Klaus Meine	Matthias Jabs
Scorpions	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Rudolf Schenker	Klaus Meine

ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

Название группы	Название CD-диска	Название песни
Scorpions	World Wide Live	Countdown
Scorpions	World Wide Live	Coming Home
Scorpions	World Wide Live	Blackout
Scorpions	Blackout	Blackout
The Big City	Blackout	Blackout

Название группы	Название песни	Автор слов	Композитор
Scorpions	Countdown	Klaus Meine	Matthias Jabs
Scorpions	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Rudolf Schenker	Klaus Meine

ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА

- Переменная отношения находится в **третьей нормальной форме** (3NF) в том и только в том случае, когда она находится во второй нормальной форме, и каждый неключевой атрибут нетранзитивно функционально зависит от первичного ключа
- Иными словами, каждое неключевое поле должно содержать информацию о ключе, полном ключе и ни о чём, кроме ключа

ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА

Сотрудник	Отдел	Телефон
Иванов	Бухгалтерия	11-22-334
Петров	Бухгалтерия	11-22-334
Сидоров	Снабжение	22-33-445

Данная таблица находится в 2NF, но не во 3NF, т.к. поле «Телефон» зависит от поля «Отдел»

Функциональные зависимости:

- Сотрудник -> Отдел
- Отдел -> Телефон
- Сотрудник -> Телефон

ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА

Отдел	Телефон
Бухгалтерия	11-22-334
Снабжение	22-33-445

Сотрудник	Отдел
Иванов	Бухгалтерия
Петров	Бухгалтерия
Сидоров	Снабжение

ЗАЧЕМ ВООБЩЕ ЭТО НУЖНО?

- На первых порах позволяет проектировать неплохие БД
- Помогает избегать типичных ошибок при проектировании БД людям без опыта
- Формирует привычку делать _нормально_ сразу, не оставляя на потом
- Постепенно вырабатывает навык проектирования, без прочной завязки на нормальных формах

ШАГ 3. СОЗДАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

- Уже имеем концептуальную схему
- Хотим получить больший уровень детализации:
 - Уточнение атрибутивного состава
 - Уточнение ограничений, накладываемых на атрибутивный состав
 - В некоторых случаях допускается уточнение типа атрибута

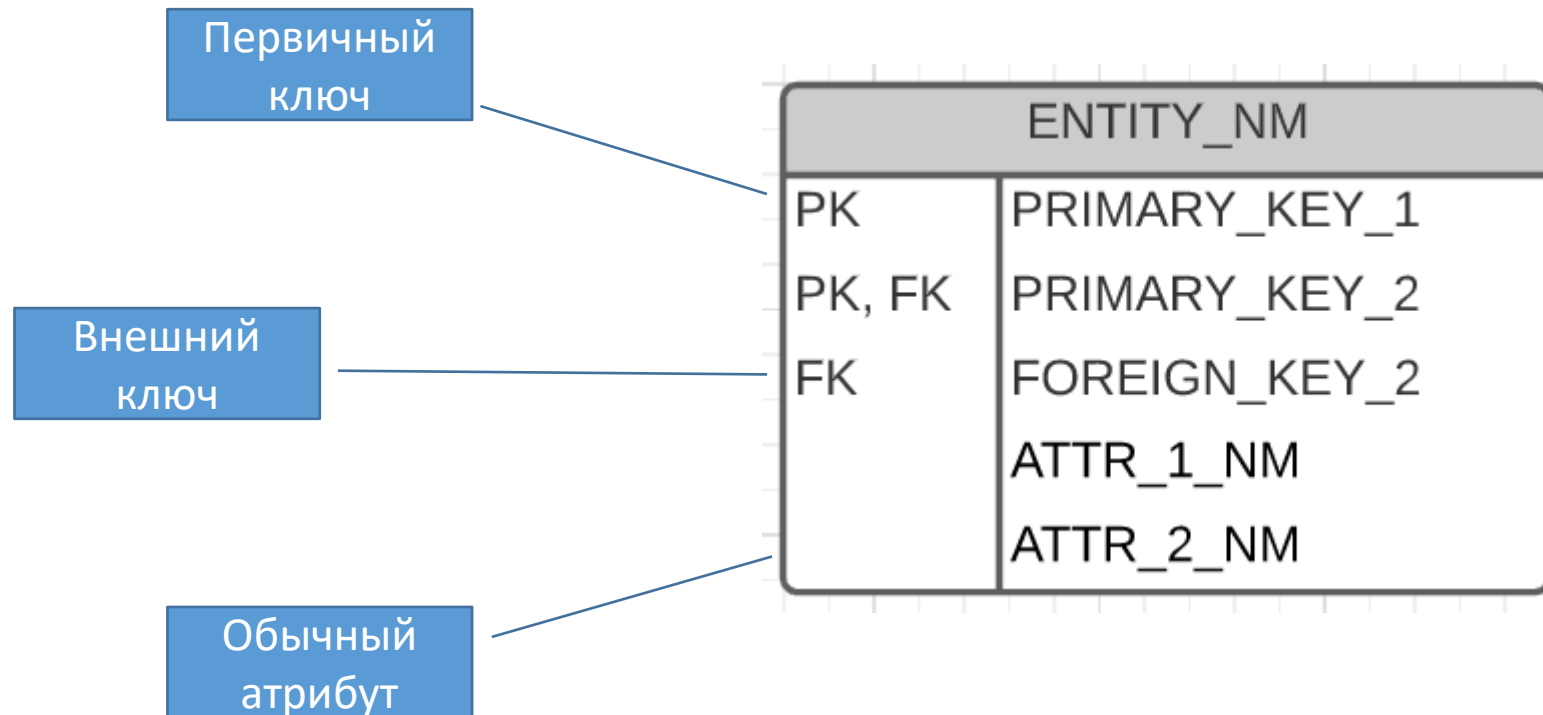
ЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

- ***Логическая модель данных*** – расширение концептуальной модели данных путем определения для сущностей их атрибутов, описаний и ограничений, частично уточняет состав сущностей и взаимосвязи между ними.

ЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

- Разрешает выход за рамки концептуальной модели
- Является прототипом будущей физической модели
- Не учитывает специфику какой-либо конкретной СУБД
- Для иллюстрации также используется ER-нотация

ER-НОТАЦИЯ «СУЩНОСТЬ»



ЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ РЕСТОРАН

