

# Реляционная модель

**Реляционная модель** основывается на понятии отношение (relation). Отношение представляет собой множество элементов, называемых кортежами. Наглядной формой представления отношения является двумерная таблица. Таблица имеет строки (записи) и столбцы (колонки). Каждая строка таблицы имеет одинаковую структуру и состоит из полей. Строкам таблицы соответствуют кортежи, а столбцам — атрибуты отношения.

## ОТДЕЛЫ

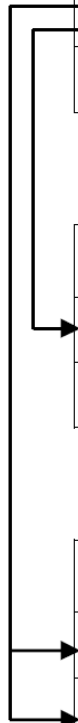
Отд_Номер	Отд_Размер	Отд_Зарп
3	10	50 000
4	15	75 000

## НАЧАЛЬНИКИ

Отд_Номер	Нач_Номер	Нач_Имя	Нач_Телеф
3	31	Иванов	0-11
4	41	Васильев	0-12

## СОТРУДНИКИ

Отд_Номер	Сотр_Номер	Сотр_Имя	Сотр_Зарп
3	32	Петров	4 000
3	33	Сидоров	4 000



# Характеристики реляционных баз данных

- Использование ключей
- Отсутствие избыточности данных
- Ограничение ввода
- Поддержание целостности данных
- Назначение прав
- Структурированный язык запросов (SQL)
- Переносимость

### ОТДЕЛЫ

Отд_Номер	Отд_Размер	Отд_Зарп
3	10	50 000
4	15	75 000

### НАЧАЛЬНИКИ

Отд_Номер	Нач_Номер	Нач_Имя	Нач_Телеф
3	31	Иванов	0-11
4	41	Васильев	0-12

### СОТРУДНИКИ

Отд_Номер	Сотр_Номер	Сотр_Имя	Сотр_Зарп
3	32	Петров	4 000
3	33	Сидоров	4 000

Таблица **ОТДЕЛЫ** содержит данные о номере отдела (Отд\_Номер), количестве сотрудников в отделе (Отд\_Размер) и суммарной зарплате сотрудников отдела (Отд\_Зарп), причем столбец Отд\_Номер является **первичным ключом**. В таблице **НАЧАЛЬНИКИ** содержатся данные о личном номере начальников (Нач\_Номер), их фамилии и телефоны, первичным ключом является столбец Нач\_Номер. В таблице **СОТРУДНИКИ** соответственно содержатся номера, фамилии и зарплата сотрудников каждого отдела; первичный ключ — Сотр\_Номер. Таблицы НАЧАЛЬНИКИ и СОТРУДНИКИ связаны с таблицей ОТДЕЛЫ по полю Отд\_Номер, т.е. это поле является еще и **внешним ключом**.

➤ Сущности

➤ Связи

Виды связей:

☐ Многие ко многим

☐ Один ко многим

- с обязательной связью
- с необязательной связью

☐ Один к одному

- с обязательной связью
- с необязательной связью

## Многие ко многим

- Слева указаны работники (их id), справа — должности (их id). Работники и должности на этой таблице указываются с помощью id.

Employeeid	PositionId
1	1
1	2
2	3
3	3

Таким образом, мы говорим, что работник с id 1 находится на должность с id 1. При этом обратите внимание на то, что в этой таблице работник с id 1 имеет две должности: 1 и 2. Т.е., каждому работнику слева соответствует некая должность справа.

Мы также можем сказать, что должности с id 3 принадлежат пользователи с id 2 и 3. Т.е., каждой роли справа принадлежит некий работник слева.

## Один ко многим

PhoneId	PersonId	PhoneNumber
1	5	11 091-10
2	5	19 124-66
3	17	21 972-02

Данная таблица содержит три номера телефона. При этом номера телефона с id 1 и 2 принадлежат пользователю с id 5. Номер с id 3 принадлежит пользователю с id 17.

## Один к одному

DisabledPersonId	EmployeeId
1	159
2	722
3	937

Но это еще не связь один к одному. Дело в том, что в такую таблицу работник может быть вписан более одного раза, соответственно, мы получили отношение один ко многим: работник может быть несколько раз инвалидом. Нужно сделать так, чтобы работник мог быть вписан в таблицу только один раз, соответственно, мог быть инвалидом только один раз. Для этого нам нужно указать, что столбец EmployeeId может хранить только уникальные значения. Нам нужно просто наложить на столбец EmployeeId ограничение unique. Это ограничение сообщает, что атрибут может принимать только уникальные значения. Выполнив это, мы получили связь один к одному.

# Обязательные и необязательные связи

## Один ко многим

- Один ко многим с обязательной связью:  
К одному полку относятся многие бойцы. Один боец относится только к одному полку. Обратите внимание, что любой солдат обязательно принадлежит к одному полку, а полк не может существовать без солдат.
- Один ко многим с необязательной связью:  
На планете Земля живут все люди. Каждый человек живет только на Земле. При этом планета может существовать и без человечества. Соответственно, нахождение нас на Земле не является обязательным

Одну и ту же связь можно рассматривать как обязательную и как необязательную. Рассмотрим вот такой пример:

**У одной биологической матери может быть много детей. У ребенка есть только одна биологическая мать.**

- А) У женщины необязательно есть свои дети. Соответственно, связь необязательна.  
Б) У ребенка обязательно есть только одна биологическая мать – в таком случае, связь обязательна.



## Один к одному

- Один к одному с обязательной связью:  
У одного гражданина определенной страны обязательно есть только один паспорт этой страны. У одного паспорта есть только один владелец.
- Один к одному с необязательной связью:  
У одной страны может быть только одна конституция. Одна конституция принадлежит только одной стране. Но конституция не является обязательной. У страны она может быть, а может и не быть, как, например, у Израиля и Великобритании.

Одну и ту же связь можно рассматривать как обязательную и как необязательную:

**У одного человека может быть только один загранпаспорт. У одного загранпаспорта есть только один владелец.**

А) Наличие загранпаспорта необязательно – его может и не быть у гражданина. Это необязательная связь.

Б) У загранпаспорта обязательно есть только один владелец. В этом случае, это уже обязательная связь.

## Многие ко многим

Любая связь многие ко многим является необязательной. Например:

**Человек может инвестировать в акции разных компаний (многих). Инвесторами какой-то компании являются определенные люди (многие).**

А) Человек может вообще не инвестировать свои деньги в акции.

Б) Акции компании мог никто не купить.

# Проектирование реляционных баз данных

- Цели проектирования БД
  - Прикладные БД
  - Предметные БД
- Процедура проектирования

Процесс проектирования информационных систем является достаточно сложной задачей. Он начинается с идентификации сущностей. Затем необходимо выполнить следующие шаги процедуры проектирования даталогической модели:

1. Представить каждый стержень (независимую сущность) таблицей базы данных (базовой таблицей) и специфицировать первичный ключ этой базовой таблицы.
2. Представить каждую ассоциацию (связь вида "многие-ко-многим" или "многие-ко-многим-ко-многим" и т.д. между сущностями) как базовую таблицу. Использовать в этой таблице внешние ключи для идентификации участников ассоциации и специфицировать ограничения, связанные с каждым из этих внешних ключей.

# Нормализация реляционных баз данных

## Ненормализованная форма или нулевая нормальная форма (UNF)

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

## Первая нормальная форма (1NF)

Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

<u>Модель</u>	<u>Фирма</u>	Цена	Скидка
M5	BMW	5500000	5%
X5M	BMW	6000000	5%
M1	BMW	2500000	5%
GT-R	Nissan	5000000	10%

(1NF)

Вторая нормальная форма (2NF)

<u>Модель</u>	<u>Фирма</u>	Цена
M5	BMW	5500000
X5M	BMW	6000000
M1	BMW	2500000
GT-R	Nissan	5000000

<u>Фирма</u>	Скидка
BMW	5%
Nissan	10%

<u>Модель</u>	Магазин	Телефон
BMW	Риал-авто	87-33-98
Audi	Риал-авто	87-33-98
Nissan	Некст-Авто	94-54-12

(2NF)

Третья нормальная форма (3NF)

<u>Магазин</u>	Телефон
Риал-авто	87-33-98
Некст-Авто	94-54-12

<u>Модель</u>	Магазин
BMW	Риал-авто
Audi	Риал-авто
Nissan	Некст-Авто

Приведение к нормальной форме Бойса-Кодда (НФБК / BCNF)

Номер стоянки	Время начала	Время окончания	Тариф
1	09:30	10:30	Бережливый
1	11:00	12:00	Бережливый
1	14:00	15:30	Стандарт
2	10:00	12:00	Премиум-В
2	12:00	14:00	Премиум-В
2	15:00	18:00	Премиум-А

(3NF)

Нормальная форма Бойса-Кодда (НФБК / BCNF) (частная форма третьей нормальной формы)

<u>Тариф</u>	Номер стоянки	Имеет льготы
Бережливый	1	Да
Стандарт	1	Нет
Премиум-А	2	Да
Премиум-В	2	Нет

<u>Тариф</u>	<u>Время начала</u>	Время окончания
Бережливый	09:30	10:30
Бережливый	11:00	12:00
Стандарт	14:00	15:30
Премиум-В	10:00	12:00
Премиум-В	12:00	14:00
Премиум-А	15:00	18:00

## Четвертая нормальная форма (4NF)

Требование четвертой нормальной формы (4NF) заключается в том, чтобы в таблицах отсутствовали нетривиальные многозначные зависимости. В таблицах многозначная зависимость выглядит следующим образом.

Таблица должна иметь как минимум три столбца, допустим А, В и С, при этом В и С между собой никак не связаны и не зависят друг от друга, но по отдельности зависят от А, и для каждого значения А есть множество значений В, а также множество значений С.

В данном случае многозначная зависимость обозначается вот так:

$$A \twoheadrightarrow B$$

$$A \twoheadrightarrow C$$

Если подобная многозначная зависимость есть в таблице, то она не соответствует четвертой нормальной форме.



Студент	Курс	Хобби
Иванов И.И.	SQL	Футбол
Иванов И.И.	Java	Хоккей
Сергеев С.С.	SQL	Волейбол
Сергеев С.С.	SQL	Теннис
John Smith	Python	Футбол
John Smith	Java	Теннис

(3NF / BCNF)

Первичный ключ здесь также составной и состоит он из всех трех столбцов. Курс и хобби никак не связаны и не зависят друг от друга, но по отдельности зависят от студента.

Студент ->-> Курс

Студент ->-> Хобби

Данная таблица не находится в четвертой нормальной форме. Допустим, нам необходимо получить информацию о хобби студентов, которые посещают курс по SQL. Очевидным действием станет выборка с условием Курс = SQL, в результате мы получим 3 хобби: футбол, волейбол и теннис. Однако в исходной таблице можно увидеть, что Иванов И.И. посещает курс по SQL и имеет хобби «Хоккей», но в нашей выборке этого хобби нет.

Чтобы нормализовать исходную таблицу, мы должны разбить ее на две.

Студент	Курс
Иванов И.И.	SQL
Иванов И.И.	Java
Сергеев С.С.	SQL
John Smith	Python
John Smith	Java

Нормализация до четвертой нормальной формы, как и до последующих (5 и 6), в современном мире на реальных данных практически не встречается.

Студент	Хобби
Иванов И.И.	Футбол
Иванов И.И.	Хоккей
Сергеев С.С.	Волейбол
Сергеев С.С.	Теннис
John Smith	Футбол
John Smith	Теннис

## Пятая нормальная форма (5NF)

Переменная отношения находится в пятой нормальной форме (иначе – в проекционно-соединительной нормальной форме) тогда и только тогда, когда каждая нетривиальная зависимость соединения в ней определяется потенциальным ключом (ключами) этого отношения.

Требование пятой нормальной формы (5NF) заключается в том, чтобы в таблице каждая нетривиальная зависимость соединения определялась потенциальным ключом этой таблицы.

Декомпозиция без потерь – процесс разбиения одной таблицы на несколько, при условии, что в случае соединения таблиц, которые были получены в результате декомпозиции, будет формироваться ровно та же самая информация, что и в исходной таблице до декомпозиции.

## Пятая нормальная форма (5NF)

Существует таблица Т (C1, C2, C3) где C1, C2, C3 – столбцы и вместе они являются составным первичным ключом. Таблица находится в четвертой нормальной форме. В соответствии с требованиями предметной области у нас проявляется зависимость соединения:

$$\{C1, C2\}, \{C1, C3\}, \{C2, C3\}$$

Чтобы привести данную таблицу к пятой нормальной форме, необходимо декомпозировать ее на следующие три таблицы:

T1 (C1, C2)

T2 (C1, C3)

T3 (C2, C3)

При этом, если мы соединим (JOIN) эти три новые таблицы (T1, T2, T3) и получим исходную таблицу (Т), то это будет означать, что декомпозицию мы выполнили без потерь.

Сотрудник	Проект	Направление
Иванов И.И.	Интернет магазин	Разработка
Сергеев С.С.	Интернет магазин	Бухгалтерия
Сергеев С.С.	Новый офис	Реализация
John Smith	Личный кабинет	Бухгалтерия
Иванов И.И.	Личный кабинет	Разработка
Иванов И.И.	Информационная система	Разработка

- Иванов И.И. может работать только в направлении «Разработка»
- Сергеев С.С. может работать в любом направлении, за исключением «Разработка»
- Иванов И.И. может участвовать в большом количестве проектов
- John Smith может участвовать только в одном проекте

Чтобы выполнить декомпозицию без потерь, нам нужно разбить данную таблицу на три проекции:

$$\{\text{Сотрудник, Проект}\}, \{\text{Сотрудник, Направление}\}, \{\text{Проект, Направление}\}$$

с условием, что в случае обратного соединения, мы получим те же самые данные, что у нас были и до декомпозиции.

Пятая нормальная форма (5NF)

Сотрудник	Проект
Иванов И.И.	Интернет магазин
Сергеев С.С.	Интернет магазин
Сергеев С.С.	Новый офис
John Smith	Личный кабинет
Иванов И.И.	Личный кабинет
Иванов И.И.	Информационная система

Сотрудник	Направление
Иванов И.И.	Разработка
Сергеев С.С.	Бухгалтерия
Сергеев С.С.	Реализация
John Smith	Бухгалтерия

Проект	Направление
Интернет магазин	Разработка
Интернет магазин	Бухгалтерия
Новый офис	Реализация
Личный кабинет	Бухгалтерия
Личный кабинет	Разработка
Информационная система	Разработка

Если выполнить следующий запрос, который соединяет эти три таблицы, и он вернет нам точно такие же данные, что и в исходной таблице, то зависимости соединения у нас нет, и наши таблицы находятся в 5NF.

```
SELECT СП.Сотрудник, ПН.Проект, СН.Направление
FROM СотрудникПроект СП
JOIN ПроектНаправление ПН ON СП.Проект = ПН.Проект
JOIN СотрудникНаправление СН ON СП.Сотрудник = СН.Сотрудник AND ПН.Направление = СН.Направление
```

	Сотрудник	Проект	Направление
1	Иванов И.И.	Интернет магазин	Разработка
2	Сергеев С.С.	Интернет магазин	Бухгалтерия
3	Сергеев С.С.	Новый офис	Реализация
4	John Smith	Личный кабинет	Бухгалтерия
5	Иванов И.И.	Личный кабинет	Разработка
6	Иванов И.И.	Информационная система	Разработка

	Сотрудник	Проект	Направление
1	Иванов И.И.	Интернет магазин	Разработка
2	Сергеев С.С.	Интернет магазин	Бухгалтерия
3	Сергеев С.С.	Новый офис	Реализация
4	John Smith	Личный кабинет	Бухгалтерия
5	Иванов И.И.	Личный кабинет	Разработка
6	Иванов И.И.	Информационная система	Разработка

Данные точно такие же. Наши таблицы находятся в пятой нормальной форме.

## Нормальные формы баз данных

- Ненормализованная форма или нулевая нормальная форма (UNF)
- Первая нормальная форма (1NF)
- Вторая нормальная форма (2NF)
- Третья нормальная форма (3NF)
- Нормальная форма Бойса-Кодда (BCNF)
- Четвертая нормальная форма (4NF)
- Пятая нормальная форма (5NF)
- Доменно-ключевая нормальная форма (DKNF)
- Шестая нормальная форма (6NF)