

Проектирование БД

Проектирование базы данных является неотъемлемой частью ее. При проектировании базы за атомарную единицу считают сущность. Не стоит путать сущность и таблицу, это не одно и то же. Сущностью называют какую-то единицу бизнес процесса, а таблица не всегда ее отражает. Существуют таблицы справочники, настроечные таблицы и т. п.

Типы связей

Всего разделяют три типа связей

Один к одному (1:1)

Проще всего будет объяснить суть этой связи на примере.

Сущности **Паспорт** и **Человек**. У одного человека связан с одним паспортом и один паспорт связан с одним человеком. Не может у человека быть больше одного паспорта и один паспорт не может принадлежать нескольким людям.

Один ко многим (1:M)

Примером в данном случае может являться **Класс** и **Ученик**.

Ученик может быть только в одном классе, однако в одном классе может быть множество учеников.

Много ко многим (M:M)

Примером в данном случае может являться **Магазин** и **Покупатель**.

Покупатель может ходить в несколько магазинов, а в магазин могут ходить сразу несколько покупателей.

Для реализации данных типов связей в SQL необходимо придерживаться следующих архитектурных решений.

Один к одному (1:1)

У одной из таблиц добавляется поле (или набор полей), которое хранит в себе первичный ключ другой таблицы. В данном типе связи совершенно не важно, какая именно таблицы будет содержать в себе это поле (или набор полей). Его называют **внешний ключ**.

Пример:

Таблица **Паспорт** (**id**, номер паспорта, кем выдан, год выдачи, **id_пользователя**)

Таблица **Человек** (**id**, имя, фамилия)

В данном случае **id_пользователя** является внешним ключом.

Один ко многим (1:M)

У таблицы, которая соответствует сущности, которая может иметь множество связей с иной сущностью, добавляется поле с внешним ключом.

Пример:

Таблица **Класс** (**id**, номер класса, **id_учителя**)

Таблица **Ученик** (**id**, имя, фамилия, **id_класса**)

В данном случае **id_класса** является внешним ключом.

Много ко многим (M:M)

В данном типе связей не обойтись обычным добавлением поля. Для реализации связи M:M необходимо добавить дополнительную таблицу связи, которая будет содержать в себе только пары соединяемых записей.

Пример:

Таблица **Магазин** (**id**, адрес, название)

Таблица **Покупатель** (**id**, имя, фамилия)

Таблица **Магазин2Покупатель** (**id_магазина**, **id_покупателя**)

Типы ключей

Ранее мы поверхностно касались темы ключей, однако на данном этапе давайте дадим им четкое определение и определим, по каким критериям они делятся.

Первичный ключ - поле, или набор полей, которые могут однозначно определить запись в таблице.

Ключ кандидата — ключ кандидата может быть любым столбцом или комбинацией столбцов, которые могут квалифицироваться как первичный ключ

Внешний ключ — это столбец или комбинация столбцов, значения которых соответствуют Первичному ключу в другой таблице. Связь между двумя таблицами задается через соответствие Первичного ключа в одной из таблиц внешнему ключу во второй.

По количеству полей ключи делятся на: **составные и простые**.

По информативности:

Естественный Ключ (ЕК) — набор атрибутов описываемой записью сущности, уникально её идентифицирующий (например, номер паспорта для человека)

Суррогатный Ключ (СК) — автоматически сгенерированное поле, никак не связанное с информационным содержанием записи.

ER-диаграмма

Данный тип диаграммы позволяет отображать сущности и связи между ними. Это, пожалуй, основной артефакт, позволяющий быстро познакомиться с базой или ее областью на необходимом уровне детализации.

Небольшой пример ER- диаграммы:



Задание

Постройте ER диаграмму службы авиаперевозок и обозначьте типы связей между ними.

Нормальные формы

Существует множество нормальных форм, однако на практике чаще всего приносятся первые три (выделены жирным).

1. **Первая нормальная форма (1NF)**
2. **Вторая нормальная форма (2NF)**
3. **Третья нормальная форма (3NF)**
4. Нормальная форма Бойса — Кодда (BCNF)
5. Четвёртая нормальная форма (4NF)
6. Пятая нормальная форма (5NF)
7. Доменно-ключевая нормальная форма (DKNF)
8. Шестая нормальная форма (6NF)

Отношение находится в первой нормальной форме (сокращённо 1НФ), если все его атрибуты атомарны, то есть если ни один из его атрибутов нельзя разделить на более простые атрибуты, которые соответствуют каким-то другим свойствам описываемой сущности.

Отношение находится во второй нормальной форме (сокращённо 2НФ) тогда и только тогда, когда оно находится в первой нормальной форме и каждый его неключевой атрибут неприводимо зависит от первичного ключа.

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Задание

Существует таблица **User** со следующими атрибутами:

1. Имя Фамилия
2. Дата рождения
3. Номер телефона

4. Мобильный оператор
5. Знак зодиака (не спрашивайте зачем, так надо)
6. Город проживания
7. Страна проживания

Необходимо привести ее к 3 нормальной форме