

Реляционная модель данных

Эта модель данных *реализована во многих существующих СУБД*, причем на сегодняшний день она является *наиболее распространенной*.

Основное *достоинство* реляционного подхода – это *небольшой набор простых и точных понятий*, которые позволяют моделировать *разнообразные предметные области*.

Структурная организация данных в реляционной модели

Основа реляционной модели – математическое понятие **отношения** (англ. – relation).

Физическим представлением отношения является обычная **двумерная таблица**.

В отдельной таблице обычно хранятся данные для некоторого **информационного объекта** (ИО).

При таком способе структурирования данных

Примеры информационных объектов

Название	Реквизиты ИО
Студент	Фамилия, имя, отчество, пол, год рождения, факультет, курс, группа, оценки, ...
Преподаватель	Фамилия, имя, отчество, пол, год рождения, кафедра, должность, стаж работы, научные труды, ...
Пациент	Фамилия, имя, отчество, пол, год рождения, место работы (учебы), должность,

В таблице реляционной БД столбцы называются **полями** и они соответствуют реквизитам ИО, для которого предназначена рассматриваемая таблица.

Каждому полю обычно дают содержательное **название**, причем в отдельной таблице названия полей не должны повторяться.

Строки таблицы для хранения данных называются **записями** (или **кортежами**).

В полях отдельной записи хранятся

Пример таблицы для хранения данных

<i>Поле1</i>	<i>Поле2</i>	<i>....</i>	<i>ПолеN</i>
Фамилия	Имя	Возраст
Иванов	Василий	21
Петрова	Ольга	19

Заголовок
таблицы

Запись1

Запись2

При формировании заголовка таблицы порядок расположения столбцов значения не имеет.

Записи в таблице тоже могут размещаться в произвольном порядке.

Фундаментальные (базовые) свойства отношения (таблицы)

1. Каждая ячейка отношения содержит только одно элементарное (атомарное, неделимое) значение.
2. Каждая запись является уникальной, т.е. дублирование записей не допускается.
Это следует из определения таблицы как множества записей, а каждое множество по определению состоит из различных элементов.
3. Порядок размещения записей не имеет никакого значения, что также вытекает из понятия «множество».

Целостность данных в реляционной модели

Эти требования, гарантирующие корректность данных, включают в себя два условия:

- целостность отдельных таблиц (отношений);
- требование ***целостности таблицы*** по ссылкам (ссылочная целостность).
состоит в том, что любая запись в рассматриваемой таблице должна быть ***отличимой от любой другой записи.***

Минимальный набор атрибутов, позволяющий однозначно идентифицировать каждую запись рассматриваемого отношения, называется **потенциальным ключом**. Ключ называют **простым**, если он состоит из одного атрибута (поля).

Например, по номеру налогоплательщика (ИНН) можно однозначно определить его адрес, фамилию и другие персональные данные.

Ключ называется **составным**, если он образован из нескольких атрибутов.

Отношение всегда имеет хотя бы один ключ, т. к. в крайнем случае для этой роли можно использовать все множество атрибутов.

Тот потенциальный ключ, который выбран для однозначной идентификации записей таблицы, называют **первичным ключом** (Primary Key — РК).

В составе первичного ключа ни один атрибут не может содержать пустых значений (NULL).

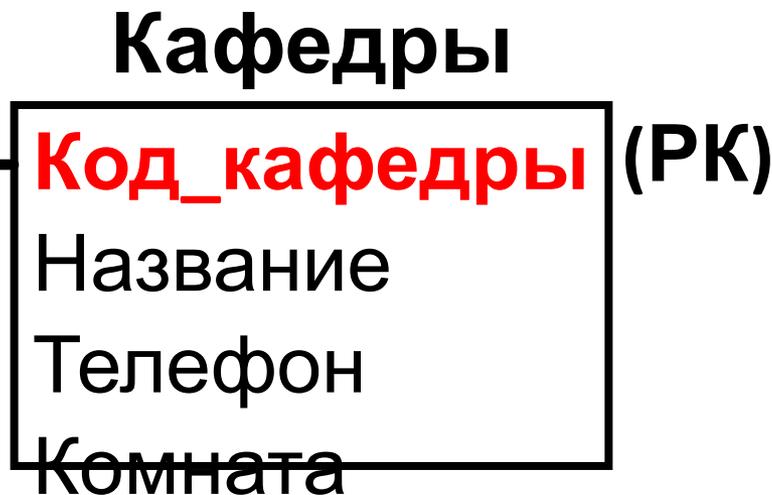
Остальные потенциальные ключи становятся **альтернативными ключами** (Alternate Key — АК).

Для первичного ключа лучше всего подходит

Целостность по ссылкам

Требование *ссылочной целостности* обусловлено тем, что очень часто данные для взаимосвязанных информационных объектов (ИО) хранятся в разных таблицах.

Преподаватели



Рассматриваемые таблицы полностью определяют ИО «Кафедра», включая перечень преподавателей по каждой

кафедре. В данном случае необходимо, чтобы каждая запись из таблицы «Преподаватели» ссылалась на **конкретную запись** в таблице «Кафедры».

Ссылки должны идти на **первичный ключ** таблицы «Кафедры», т.к. именно этот атрибут **однозначно идентифицирует** конкретную кафедру.

При этом в таблице «Преподаватели» атрибут «Кафедра», содержащий адреса ссылок, выполняет роль **внешнего ключа** (Foreign Key — FK)

В данной ситуации говорят, что две таблицы ***связаны по ссылкам***.

Одной записи из таблицы «Кафедры» обычно соответствует несколько записей в таблице «Преподаватели», поэтому первую таблицу называют ***главной***, а вторую – ***подчиненной***.

Требование целостности по ссылкам

закljučается в следующем:

для каждого значения FK в подчиненной таблице должна существовать запись с таким же значением PK в главной таблице.

Если это требование не выполняется, то появляются «висячие» (или «дохлые») ссылки