

# Определение РМД

Основная идея реляционной модели данных заключается в том, чтобы представить любой набор данных в виде двумерного массива - **таблицы**. В простейшем случае реляционная модель описывает единственную двумерную таблицу, но чаще всего эта модель описывает структуру и взаимоотношения между несколькими различными таблицами. Реляционные модели данных, или реляционные базы данных, являются в настоящее время основным способом в проектировании и организации информационных систем в производстве и бизнесе. Методы и алгоритмы обработки информации в реляционных базах данных основываются на теории **реляционной алгебры** (алгебра логики, булева алгебра).

Основоположником теории реляционных баз данных считается сотрудник фирмы IBM доктор Э.Ф. Кодд, опубликовавший 6 июня 1960 г. статью «Реляционная модель данных для больших коллективных банков данных» («A Relational Model of Data for Large Shared Data Banks»). В этой статье впервые и был использован термин «реляционная модель данных», что и положило начало реляционным базам данных.

Теория опиралась на математический аппарат теории множеств. Э.Ф.Кодд доказал, что любой набор данных можно представить в виде двумерных таблиц особого вида, известных в математике как отношения. От английского слова «relation» («отношение») и произошло название «реляционная модель данных».

Таким образом:

**реляционная база данных**

представляет собой информацию (данные) об объектах, представленную в виде двумерных массивов - таблиц, объединенных определенными связями.



**Таблица базы данных** - двумерный массив, содержащий информацию об одном классе объектов, называемый отношением.

Поскольку не всякой таблице можно поставить в соответствие отношение, приведем **условия, выполнение которых позволяет таблицу считать отношением.**

1. Все строки таблицы должны быть уникальны.
2. Имена столбцов таблицы должны быть различны, а значения их простыми, то есть недопустима группа значений в одном столбце одной строки.
3. Все строки одной таблицы должны иметь одну структуру, соответствующую именам и типам столбцов.
4. Порядок размещения строк в таблице может быть произвольным.

**Кортеж** представляет собой строку таблицы, последовательность атрибутов, связанных между собой отношениями AND (И). Множество кортежей называется **телом отношения**.

**Атрибуты** представляют собой свойства, характеризующие сущность (объект любой природы, данные о котором хранятся в базе данных, данные о сущности хранятся в отношении). В структуре таблицы каждый атрибут именуется и ему соответствует заголовок некоторого столбца таблицы.

**Домен** представляет собой множество всех возможных значений определенного атрибута отношения. Каждый домен образует значения одного типа данных, например, числовые или символьные.

# Термины реляционных баз данных

Теория БД	Реляционные СУБД(FoxPro, Microsoft Access)	SQL Server 7.0
Отношение (Relation)	Таблица (Table)	Таблица (Table)
Атрибут (Attribute)	Поле (Field)	Колонка (Column)
Кортеж (Tuple)	Запись (Record)	Строка (Row)



Одним из важных понятий, необходимых для построения оптимальной структуры реляционных баз данных, является понятие ключа.

**Ключом** считается поле, значения которого однозначно определяют значения всех остальных полей в таблице. Например, поле «номер паспорта», или «ИНН», однозначно определяет характеристики любого физического лица (при составлении баз данных для отделов кадров или бухгалтерии предприятия).

Каждое поле, не входящее в состав ключа, называется **не ключевым** полем таблицы.

Пусть в отношении  $R_1$  имеется не ключевой атрибут  $A$ , значения которого являются значениями ключевого атрибута  $B$  другого отношения  $R_2$ . Тогда говорят, что атрибут  $A$  отношения  $R_1$  есть **внешний ключ**. С помощью внешних ключей устанавливаются связи между отношениями.

Ключом таблицы может быть не одно, а сочетание нескольких полей - **составной ключ**. Каждое отношение имеет обязательно комбинацию атрибутов, которая может служить ключом. Ее существование гарантируется тем, что в отношении нет повторяющихся кортежей, а это значит, что по крайней мере вся совокупность атрибутов обладает свойством однозначной идентификации кортежей отношения. В этом случае множество полей может быть возможным ключом таблицы только тогда, когда удовлетворяются два независимых от времени условия: **уникальность и минимальность**.



**Уникальность** ключа означает, что в любой момент времени таблица базы данных не может содержать никакие две различные записи, имеющие одинаковые значения ключевых полей. Выполнение условия уникальности является обязательным.

**Минимальность** ключевых полей означает, что только сочетание значений всех выбранных полей отвечает требованиям уникальности записей таблицы базы данных. То есть, это означает, что ни одно из входящих в ключ полей не может быть исключено из него без нарушения уникальности.

Каждая таблица должна иметь, по крайней мере, один возможный ключ, который выбирается в качестве главного (**первичного**) ключа (поля «номер паспорта» или «ИНН» могут быть главными ключами). Обычно первичным ключом назначается тот ключ, которым проще всего пользоваться при повседневной работе. Остальные возможные ключи, если они есть, называются **альтернативными**. Ключи обычно используют в качестве идентификатора и организации связывания таблиц.

К отношениям можно применять систему операций, позволяющую получить одни отношения из других. Например, результатом запроса к реляционной БД может быть новое отношение, вычисленное на основе имеющихся отношений. Поэтому можно разделить обрабатываемые данные на хранимую и вычисляемую части. Основной единицей обработки данных в реляционных БД является отношение, а не отдельные его кортежи (записи).

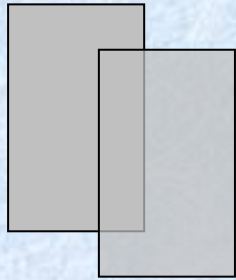


# Операции реляционной алгебры.

В реляционной алгебре операнды и результаты всех действий являются **отношениями**. Языки реляционной алгебры являются процедурными, так как отношение, являющееся результатом запроса к реляционной БД, вычисляется при выполнении последовательности реляционных операторов, применяемых к отношениям. Операторы состоят из **операндов**, в роли которых выступают отношения, и **реляционных операций**. Результатом реляционной операции является отношение. Вариант реляционной алгебры, предложенный Коддом, включает в себя следующие основные операции: объединение, разность (вычитание), пересечение, декартово (прямое) произведение (или произведение), выборка (селекция, ограничение), проекция, деление и соединение.

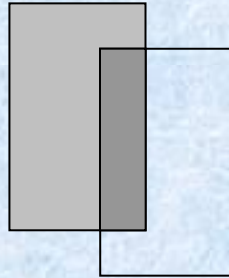
# Упрощенное представление операций:

ОБЪЕДИНЕНИЕ



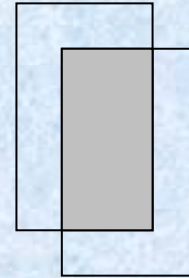
Подробно

РАЗНОСТЬ



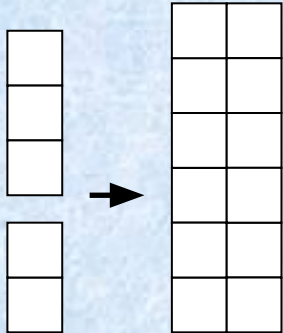
Подробно

ПЕРЕСЕЧЕНИЕ



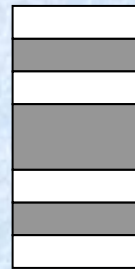
Подробно

ДЕКАРТОВО ПРОИЗВЕДЕНИЕ



Подробно

ВЫБОРКА



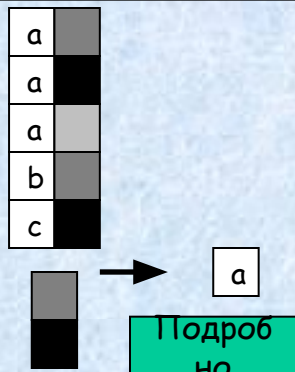
Подробно

ПРОЕКЦИЯ



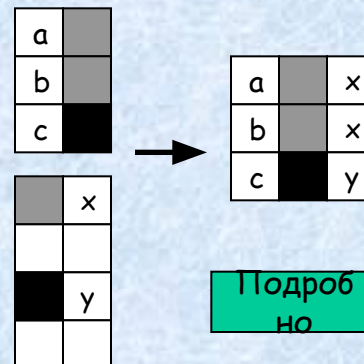
Подробно

ДЕЛЕНИЕ



Подробно

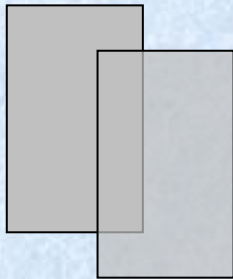
СОЕДИНЕНИЕ  
(ЕСТЕСТВЕННОЕ)



Подробно

К следующему разделу

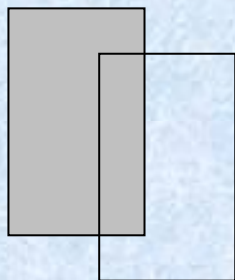
**Объединением** двух отношений  $R_1$  и  $R_2$  одинаковой размерности ( $R_1 \text{ UNION } R_2$ ) является отношение  $R$ , содержащее все элементы исходных отношений (с исключением повторений).



К  
таблице

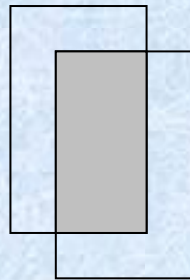


**Вычитание** отношений  $R1$  и  $R2$  одинаковой размерности ( $R1 \text{ MINUS } R2$ ) есть отношение, тело которого состоит из множества кортежей, принадлежащих  $R1$ , но не принадлежащих  $R2$ . Результат операции вычитания зависит от порядка следования операндов, то есть  $R1 \text{ MINUS } R2$  и  $R2 \text{ MINUS } R1$  - не одно и то же.



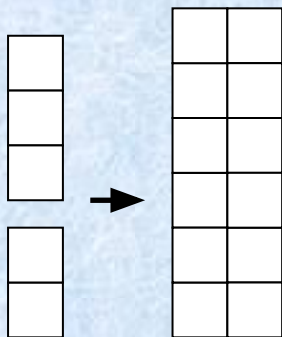
К  
таблице

**Пересечение** двух отношений  $R_1$  и  $R_2$  одинаковой размерности ( $R_1 \text{ INTERSECT } R_2$ ) порождает отношение  $R$  с телом, включающим в себя кортежи, одновременно принадлежащие обоим исходным отношениям.



К  
таблице

**Декартово произведение** отношения R1 степени k1 и отношения R2 степени k2 (R1 TIMES R2) (степень отношения определяется количеством атрибутов, которое в нем присутствует), которые не имеют одинаковых имен атрибутов, есть такое отношение R степени (k1+k2), заголовок которого представляет сцепление заголовков отношений R1 и R2, а тело имеет кортежи такие, что первые k1 элементов кортежей принадлежат множеству R1, а последние k2 элементов - множеству R2. При необходимости получить произведение двух отношений, имеющих одинаковые имена одного или нескольких атрибутов, применяется операция переименования RENAME.



К  
таблице



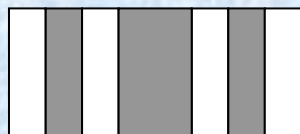
**Выборка** ( $R \text{ WHERE } f$ ) отношения  $R$  по формуле  $f$  представляет собой новое отношение с таким же заголовком и телом, состоящим из таких же кортежей отношения  $R$ , которые удовлетворяют истинности логического выражения, заданного формулой  $f$ . Для записи формулы используются операнды - имена атрибутов (или номера столбцов), константы, логические операции ( $AND$  - И,  $OR$  - ИЛИ,  $NOT$  - НЕ), операции сравнения и скобки.


К  
таблице

**Проекция** отношения  $A$  на атрибуты  $X, Y, \dots, Z$  ( $A[X, Y, \dots, Z]$ ), где множество  $\{X, Y, \dots, Z\}$  является подмножеством полного списка атрибутов заголовка отношения  $A$ , представляет собой отношение с заголовком  $X, Y, \dots, Z$  и телом, содержащим кортежи отношения  $A$ , за исключением повторяющихся кортежей. Повторение одинаковых атрибутов в списке  $X, Y, \dots, Z$  запрещается.

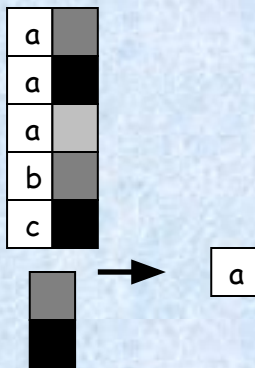
Операция проекции допускает следующие дополнительные варианты записи:

- ✓ отсутствие списка атрибутов подразумевает указание всех атрибутов (операция тождественной проекции)
- ✓ выражение вида  $R[ ]$  означает пустую проекцию, результатом которой является пустое множество
- ✓ операция проекции может применяться к произвольному отношению, в том числе и к результату выборки.



К  
таблице

Результатом *деления* отношения R1 с атрибутами A и B на отношение R2 с атрибутом B ( $R1 \text{ DIVIDEBY } R2$ ), где A и B простые или составные атрибуты, причем атрибут B - общий атрибут, определенный на одном и том же домене (множестве доменов составного атрибута), является отношение R с заголовком A и телом, состоящим из кортежей r таких, что в отношении R1 имеются кортежи (r, s), причем множество значений s включает множество значений атрибута B отношения R2.

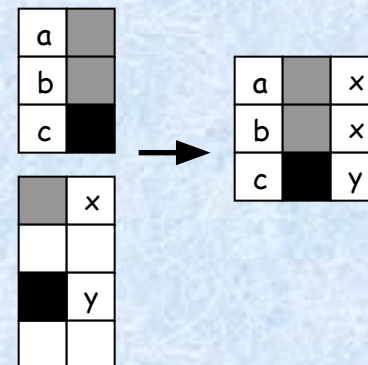
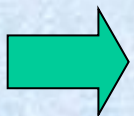


К  
таблице



**Соединение**  $C_f(R1, R2)$  отношений  $R1$  и  $R2$  по условию, заданному формулой  $f$ , представляет собой отношение  $R$ , которое можно получить путем Декартова произведения отношений  $R1$  и  $R2$  с последующим применением к результату операции выборки по формуле  $f$ . Правила записи формулы  $f$  такие же, как и для операции селекции.

Другими словами, соединением отношения  $R1$  по атрибуту  $A$  с отношением  $R2$  по атрибуту  $B$  (отношения не имеют общих имен атрибутов) является результат выполнения операции вида:  $(R1 \text{ TIMES } R2) \text{ WHERE } A \text{ } Q \text{ } B$ , где  $Q$  - логическое выражение над атрибутами, определенными на одном (нескольких - для составного атрибута) домене. Соединение  $C_f(R1, R2)$ , где формула  $f$  имеет произвольный вид (в отличие от частных случаев), называют также  $Q$ -соединением.



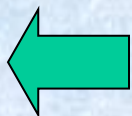


Входят в соединение только те элементы, которые удовлетворяют условию. Это называется условием соединения.

Важными с практической точки зрения частными случаями соединения являются **эквисоединение** и **естественное соединение**.

Если условие соединения является условием равенства, то такое соединение называется эквисоединением.

Если столбцы отношения поименованы и условие равенства накладывается на одноименные столбцы, то такое соединение называется естественным соединением.



# Основные виды связи таблиц.

Логическое связывание таблиц происходит с помощью внешнего ключа. Суть связывания состоит в установлении соответствия полей таблиц. То есть, поле, содержащееся в данной таблице и являющееся главным ключом в другой таблице, есть внешний ключ, связывающий эти таблицы.

Три основных вида связи:

- ✓ один к одному 1:1
- ✓ один ко многим 1:M
- ✓ многие ко многим M:M

**Связь вида 1:1** образуется в случае, когда все поля, по которым связываются таблицы, являются ключевыми. Поскольку значения в ключевых полях обеих таблиц не повторяются, обеспечивается взаимно-однозначное соответствие записей этих таблиц. На практике связи вида 1:1 используется сравнительно редко, так как хранимую информацию легко объединить в одну таблицу. Возможны случаи, когда удобнее иметь не одну, а две и более таблиц. Причиной этого может быть необходимость ускорить обработку, повысить удобство работы нескольких пользователей с общей информацией, обеспечить более высокую степень защиты информации и т.д. Пример

**Связь 1:M** имеет место в случае, когда одной записи одной таблицы соответствует несколько записей другой таблицы. Пример

**Связь M:M** - самый общий вид связи, возникает в случаях, когда нескольким записям одной таблицы соответствует несколько записей другой таблицы. Из перечисленных видов связи чаще используется связь вида 1:M. Пример

К  
следующему  
разделу

## Пример связи 1:1

<i>Код</i>	<i>Цена</i>
<i>а</i>	<i>4000</i>
<i>б</i>	<i>1500</i>

Назад

<i>Код</i>	<i>Название</i>
<i>а</i>	<i>Стол</i>
<i>б</i>	<i>Стул</i>



## Пример связи 1:М

<i>Код</i>	<i>Вид устройства</i>
<i>а</i>	<i>CD-ROM</i>
<i>б</i>	<i>CD-Recorder</i>
<i>в</i>	<i>Sound Blaster</i>

Назад

<i>Код</i>	<i>Фирма-производитель</i>	<i>Наличие на складе</i>
<i>а</i>	<i>Acer</i>	<i>да</i>
<i>а</i>	<i>Mitsumi</i>	<i>нет</i>
<i>а</i>	<i>NEC</i>	<i>да</i>
<i>а</i>	<i>Panasonic</i>	<i>да</i>
<i>а</i>	<i>Sony</i>	<i>да</i>
<i>б</i>	<i>Philips</i>	<i>нет</i>
<i>б</i>	<i>Sony</i>	<i>нет</i>
<i>б</i>	<i>Yamaha</i>	<i>да</i>
<i>в</i>	<i>Creative Labs</i>	<i>да</i>

## Пример связи М:М

<i>Рабочий</i>	<i>Станок</i>
<i>Иванов</i>	<i>1</i>
<i>Иванов</i>	<i>2</i>
<i>Петров</i>	<i>1</i>
<i>Петров</i>	<i>3</i>
<i>Сидоров</i>	<i>2</i>

<i>Обслуживающий</i>	<i>Станок</i>
<i>Голубев</i>	<i>1</i>
<i>Голубев</i>	<i>3</i>
<i>Зыков</i>	<i>2</i>
<i>Зыков</i>	<i>3</i>

[Назад](#)

# Целостность.

**Целостность данных** - это непрерывное во времени соответствие совокупности данных общепринятым стандартам хранения данных, а также некоторым правилам, которые определяются заранее.

**Ограничения целостности** - это условия, накладываемые на данные с целью обеспечения их непротиворечивости. Ограничения целостности делятся на внутренние, которые накладываются самой базой данных, и явные, которые явно выписываются проектировщиками. Выделяют также ограничения целостности объектов, связей и приложений.

Ограничение *целостности объектов* накладывает ограничение на ключ, который не может содержать пустых значений. Смысл такого ограничения – идентификация объектов.

Ограничение *целостности связей* предполагает, что связь может быть установлена только с существующим объектом, т.е. поля, используемые в одних отношениях и одновременно являющиеся ключевыми в других должны выбираться из реально существующих.

Ограничение *целостности приложений* – это те ограничения, которые вытекают из сути предметной области.



Для обеспечения целостности данных необходимо выполнять следующие действия:

- ✓ при удалении из отношения записи нужно проверить отсутствие ссылок на удаляемую запись во всех связанных с ним отношениях. Причем прежде всего необходимо удалить в связанных отношениях все записи, которые ссылаются на удаляемую запись, и только после этого – саму запись
- ✓ необходимо внимательно относиться к изменению значений связующих полей
- ✓ при добавлении новой записи в отношение, в котором осуществляется ссылка на значение в другом отношении, необходимо проверить наличие связующих полей в отношении, на которое осуществляется ссылка.

Как правило, определение условия целостности данных осуществляется при установлении взаимосвязи между отношениями; при этом пользователю часто предоставляется возможность самому решить, каким путем сохранять целостность базы данных и насколько жестко должно соблюдаться условие целостности при различных операциях изменения данных.