

## Ограничения целостности

Целостность данных - это механизм поддержания соответствия базы данных предметной области. В реляционной модели данных определены два базовых требования обеспечения целостности:

- **целостность ссылок**
- **целостность сущностей.**

**Целостность сущностей.**

**Объект реального мира представляется в реляционной базе данных как кортеж некоторого отношения. Требование целостности сущностей заключается в следующем: **каждый кортеж любого отношения должен отличаться от любого другого кортежа этого отношения (т.е. любое отношение должно обладать первичным ключом).****

Вполне очевидно, что если данное требование не соблюдается (т.е. кортежи в рамках одного отношения не уникальны), то в базе данных может храниться противоречивая информация об одном и том же объекте. Поддержание целостности сущностей обеспечивается средствами системы управления базой данных (СУБД). Это осуществляется с помощью двух ограничений:

- **при добавлении записей в таблицу проверяется уникальность их первичных ключей**
- **не допускается изменение значений атрибутов, входящих в первичный ключ.**

## Целостность ссылок

Сложные объекты реального мира представляются в реляционной базе данных в виде кортежей нескольких нормализованных отношений, связанных между собой. При этом:

1. Связи между данными отношениями описываются в терминах функциональных зависимостей.
2. Для отражения функциональных зависимостей между кортежами разных отношений используется дублирование первичного ключа одного отношения (родительского) в другое (дочернее). Атрибуты, представляющие собой копии ключей родительских отношений, называются внешними ключами.

Требование целостности по ссылкам состоит в следующем:

**Для каждого значения внешнего ключа, появляющегося в дочернем отношении, в родительском отношении должен найтись кортеж с таким же значением первичного ключа.**

Пусть, например, даны отношения **ОТДЕЛ** (**N\_ОТДЕЛА**, **ИМЯ\_ОТДЕЛА**) и **СОТРУДНИК** (**N\_СОТРУДНИКА**, **N\_ОТДЕЛА**, **ИМЯ\_СОТРУДНИКА**), в которых хранятся сведения о работниках предприятия и подразделениях, где они работают. Отношение **ОТДЕЛ** в данной паре является родительским, поэтому его первичный ключ "*N\_отдела*" присутствует в дочернем отношении **СОТРУДНИК**.

Требование целостности по ссылкам означает здесь, что в таблице **СОТРУДНИК** не может присутствовать кортеж со значением атрибута "*N\_отдела*", которое не встречается в таблице **ОТДЕЛ**. Если такое значение в отношении **ОТДЕЛ** отсутствует, значение внешнего ключа в отношении **СОТРУДНИК** считается неопределенным.

**Как правило, поддержание целостности ссылок также возлагается на систему управления базой данных. Например, она может не позволить пользователю добавить запись, содержащую внешний ключ с несуществующим (неопределенным) значением.**

**Часто вместо выражения "целостность по ссылкам" употребляют его синонимы "ссылочная целостность", "целостность связей" или "требование внешнего ключа".**

## **Реляционное исчисление.**

В реляционной модели определяются два базовых механизма манипулирования данными:

- **основанная на теории множеств реляционная алгебра**
- **основанное на математической логике реляционное исчисление.**

## Операции над данными (реляционная алгебра).

### Операции обработки кортежей.

Эти операции связаны с изменением состава кортежей в каком-либо отношении.

- **ДОБАВИТЬ** - необходимо задать имя отношения и ключ кортежа.
- **УДАЛИТЬ** - необходимо указать имя отношения, а также идентифицировать кортеж или группу кортежей, подлежащих удалению.
- **ИЗМЕНИТЬ** - выполняется для названного отношения и может корректировать как один, так и несколько кортежей.



В реляционной алгебре определены следующие операции обработки отношений:

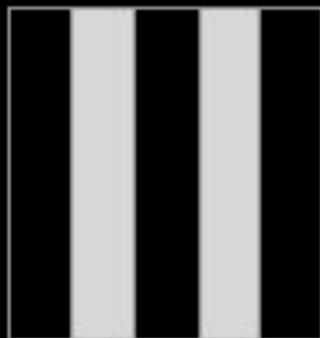
• **ПРОЕКЦИЯ (ВЕРТИКАЛЬНОЕ ПОДМНОЖЕСТВО).**

Операция проекции представляет из себя выборку из каждого кортежа отношения значений атрибутов, входящих в список A, и удаление из полученного отношения повторяющихся строк.

**Проекция / PROJECT /**

<u>Обозначение</u>	<u>Определение</u>	<u>LEAP</u>
$R[A]$	$\{r[A] : r \in R\}$	$r = \text{project}(R) (A_1, A_2, \dots, A_n)$

Пример:



$$R[M, T] = \begin{bmatrix} x & a \\ y & a \\ z & a \\ w & b \\ \hline w & b \\ \hline w & b \end{bmatrix} = \begin{bmatrix} x & a \\ y & a \\ z & a \\ w & b \end{bmatrix}$$

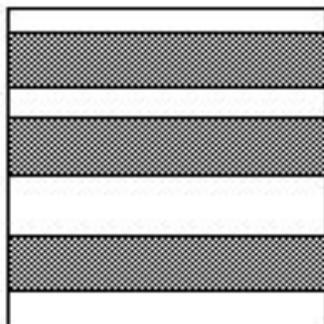
• **ВЫБОРКА (ОГРАНИЧЕНИЕ, ГОРИЗОНТАЛЬНОЕ ПОДМНОЖЕСТВО).**

На входе используется одно отношение, результат - новое отношение, построенное по той же схеме, содержащее подмножество кортежей исходного отношения, удовлетворяющих условию выборки.

**Выборка / SELECT /**

<u>Обозначение</u>	<u>Определение</u>	<u>LEAP</u>
$R[A \ \theta \ v]$	$\{r : r \in R \wedge (r[A] \ \theta \ v)\}$	$r = \text{select } (R) \ ((\text{cond}) \ \text{bool} \ (\text{cond}))$
$R[A_1 \ \theta \ A_2]$	$\{r : r \in R \wedge (r[A_1] \ \theta \ r[A_2])\}$	

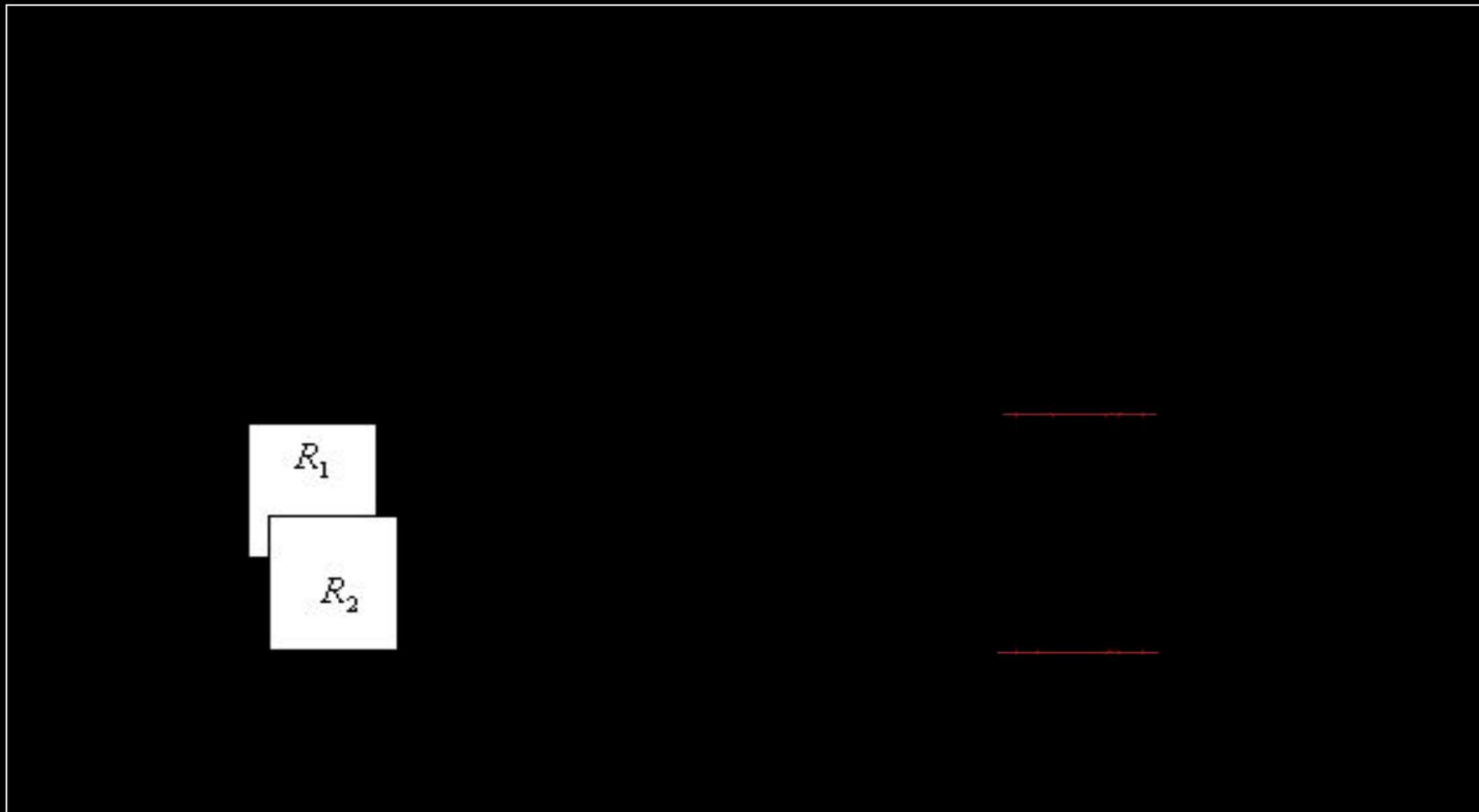
Пример:



$$P[D_2 = 11] = \begin{bmatrix} 1 & 11 & x \\ 2 & 11 & y \\ 3 & 11 & z \end{bmatrix}$$

## • ОБЪЕДИНЕНИЕ.

Отношения-операнды в этом случае должны быть определены по одной схеме. Результирующее отношение содержит все строки операндов за исключением повторяющихся.



## • ПЕРЕСЕЧЕНИЕ.

На входе операции два отношения, определенные по одной схеме. На выходе - отношение, содержащие кортежи, которые присутствуют в обоих исходных отношениях.



## •РАЗНОСТЬ.

Операция во многом похожая на ПЕРЕСЕЧЕНИЕ, за исключением того, что в результирующем отношении содержатся кортежи, присутствующие в первом и отсутствующие во втором исходных отношениях.



# •ДЕКАРТОВО ПРОИЗВЕДЕНИЕ

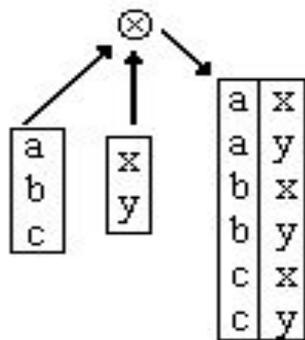
Входные отношения могут быть определены по разным схемам. Схема результирующего отношения включает все атрибуты исходных. Кроме того:

- степень результирующего отношения равна сумме степеней исходных отношений
- мощность результирующего отношения равна произведению мощностей исходных отношений.

## Декартово произведение / CARTESIAN PRODUCT /

<u>Обозначение</u>	<u>Определение</u>	<u>LEAP</u>
$R_1 \otimes R_2$	$\{(r_1    r_2) : r_1 \in R_1 \wedge r_2 \in R_2\}$	$r = (R1) \text{ product } (R2)$

Пример:



$$R[M, T] \otimes (R[Q, T] \cap S) = \begin{bmatrix} x & a \\ y & a \\ z & a \\ w & b \end{bmatrix} \otimes \begin{bmatrix} 5 & a \\ 1 & b \end{bmatrix} = \begin{bmatrix} x & a & 5 & a \\ x & a & 1 & b \\ y & a & 5 & a \\ y & a & 1 & b \\ z & a & 5 & a \\ z & a & 1 & b \\ w & b & 5 & a \\ w & b & 1 & b \end{bmatrix}$$

## •СОЕДИНЕНИЕ

Данная операция имеет сходство с ДЕКАРТОВЫМ ПРОИЗВЕДЕНИЕМ. Однако, здесь добавлено условие, согласно которому вместо полного произведения всех строк в результирующее отношение включаются только строки, удовлетворяющие определенному соотношению между атрибутами соединения  $(A_1, A_2)$  соответствующих отношений.



Также как и выражения реляционной алгебры формулы реляционного исчисления определяются над отношениями реляционных баз данных, и результатом вычисления также является отношение.

Эти механизмы манипулирования данными различаются уровнем процедурности:

- запрос, представленный на языке реляционной алгебры, может быть вычислен на основе вычисления элементарных алгебраических операций с учетом их старшинства и возможных скобок
- формула реляционного исчисления только устанавливает условия, которым должны удовлетворять кортежи результирующего отношения. Поэтому языки реляционного исчисления являются более непроедурными или декларативными.

Пример: Пусть даны два отношения: СОТРУДНИКИ  
(СОТР\_НОМЕР, СОТР\_ИМЯ, СОТР\_ЗАРПЛ, ОТД\_НОМЕР)  
ОТДЕЛЫ(ОТД\_НОМЕР, ОТД\_КОЛ, ОТД\_НАЧ)

Мы хотим узнать имена и номера сотрудников, являющихся начальниками отделов с количеством работников более 10. Выполнение этого запроса средствами реляционной алгебры распадается на четко определенную последовательность шагов:

(1).выполнить соединение отношений СОТРУДНИКИ и ОТДЕЛЫ по условию СОТР\_НОМ = ОТДЕЛ\_НАЧ.

$S1 = \text{СОТРУДНИКИ} [\text{СОТР\_НОМ} = \text{ОТД\_НАЧ}] \text{ОТДЕЛЫ}$

**(2).из полученного отношения произвести выборку по условию  
ОТД\_КОЛ > 10**

**C2 = C1 [ОТД\_КОЛ > 10].**

**(3).спроецировать результаты предыдущей операции на  
атрибуты СОТР\_ИМЯ, СОТР\_НОМЕР**

**C3 = C2 [СОТР\_ИМЯ, СОТР\_НОМЕР]**

**Заметим, что порядок выполнения шагов может повлиять на эффективность выполнения запроса. Так, время выполнения приведенного выше запроса можно сократить, если поменять местами этапы (1) и (2).**

**В этом случае сначала из отношения СОТРУДНИКИ будет сделана выборка всех кортежей со значением атрибута ОТДЕЛ\_КОЛ > 10, а затем выполнено соединение результирующего отношения с отношением ОТДЕЛЫ.**

**Машинное время экономится за счет того, что в операции соединения участвуют меньшие отношения.**

**На языке реляционного исчисления данный запрос может быть записан как:**

**Выдать СОТР\_ИМЯ и СОТР\_НОМ для СОТРУДНИКИ таких,  
что**

**существует ОТДЕЛ с таким же, что и СОТР\_НОМ значением  
ОТД\_НАЧ**

**и значением ОТД\_КОЛ большим 50.**

**Здесь мы указываем лишь характеристики результирующего отношения, но не говорим о способе его формирования. СУБД сама должна решить какие операции и в каком порядке надо выполнить над отношениями СОТРУДНИКИ и ОТДЕЛЫ. Задача оптимизации выполнения запроса в этом случае также ложится на СУБД.**