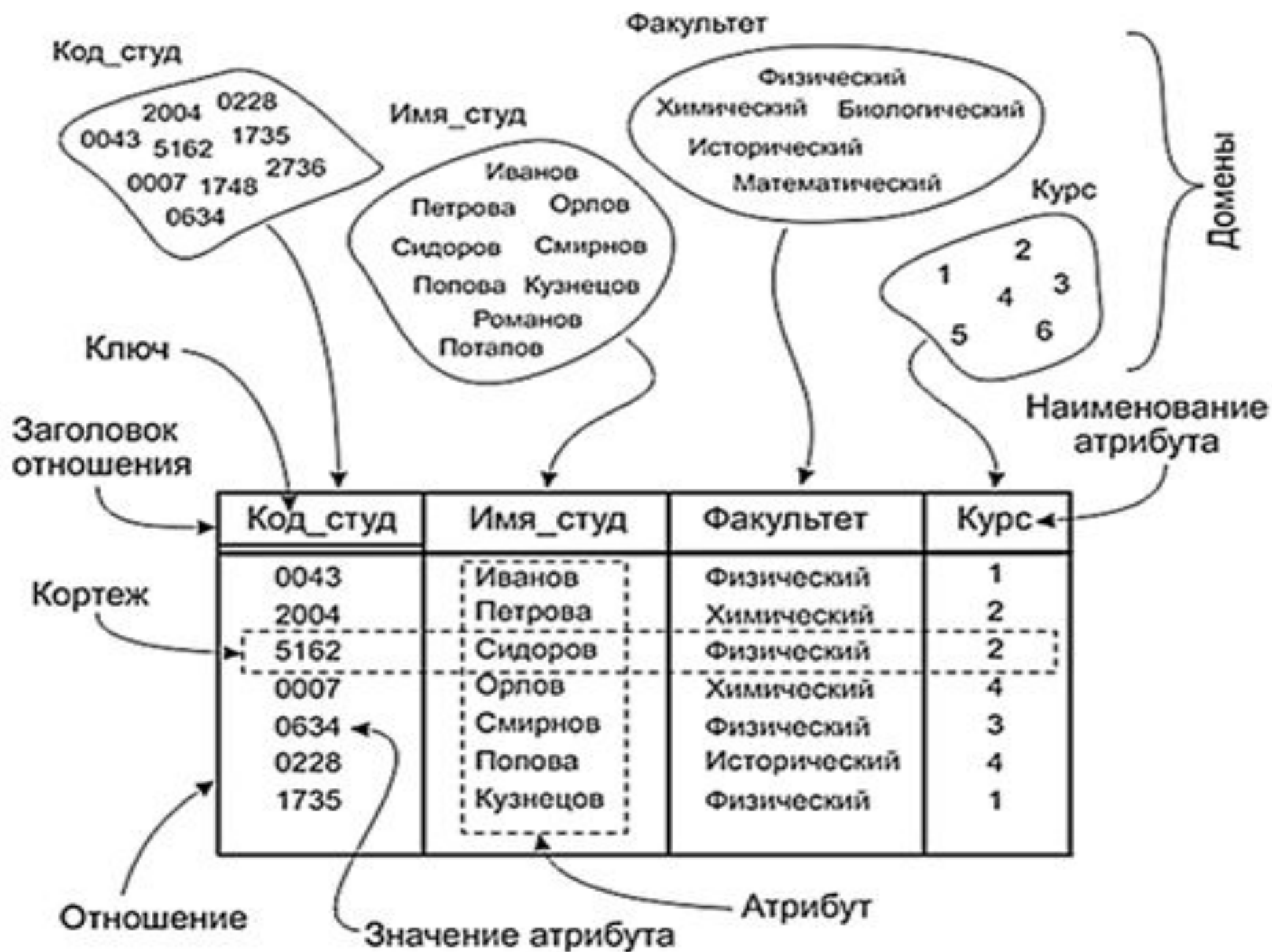


Основные понятия реляционной модели данных.
Нормальные формы.

1. **Отношение** это таблица, состоящая из строк и столбцов и имеющая вверху строку, называемую заголовком отношения.
2. Строки таблицы-отношения называются **кортежами** (tuple), а столбцы **атрибутами** (attribute).
3. Количество кортежей в отношении называется **кардинальным числом отношения**, а количество атрибутов называется **степенью отношения**.
4. Каждый атрибут в отношении имеет наименование, которое указывается в заголовочной части отношения.
5. **Ключ отношения** – это атрибут или набор атрибутов отношения такие, что в любой момент времени в отношении не существует строк, для которых значение или комбинация значений ключевых атрибутов являются одинаковыми. Ключ, таким образом, является уникальным идентификатором кортежей отношения.
6. **Домен отношения** – это множество значений, из которого могут браться значения конкретного атрибута. То есть конкретный набор значений атрибута в любой момент времени должен быть подмножеством множества значений домена, на котором определен этот атрибут. Значения атрибута, которые отсутствуют в множестве, задаваемом доменом, являются недопустимыми.



Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам, называется ***нормализацией***. Нормализация предназначена для приведения структуры БД к виду, обеспечивающему ***минимальную логическую избыточность***

общее назначение процесса нормализации заключается в следующем:

1. исключение некоторых типов избыточности;
2. устранение некоторых аномалий обновления;
3. разработка проекта базы данных, который является достаточно «качественным» представлением реального мира, интуитивно понятен и может служить хорошей основой для последующего расширения;
4. упрощение процедуры применения необходимых ограничений целостности.

Устранение избыточности производится, как правило, за счёт декомпозиции отношений таким образом, чтобы в каждом отношении хранились только первичные факты (то есть факты, не выводимые из других хранимых фактов).

Первая нормальная форма (1NF)

Переменная отношения находится в первой нормальной форме (1НФ) тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

Таблица должна удовлетворять следующим пяти условиям:

1. Нет упорядочивания строк сверху вниз (другими словами, порядок строк не несет в себе никакой информации).
2. Нет упорядочивания столбцов слева направо (другими словами, порядок столбцов не несет в себе никакой информации).
3. Нет повторяющихся строк.
4. Каждое пересечение строки и столбца содержит ровно одно значение из соответствующего домена (и больше ничего).
5. Все столбцы являются обычными.

Исходная ненормализованная (то есть не являющаяся правильным представлением некоторого отношения) таблица:

<u>Сотрудник</u>	<u>Номер телефона</u>
Иванов И. И.	283-56-82 390-57-34
Петров П. П.	708-62-34

Таблица, приведённая к 1НФ, являющаяся правильным представлением некоторого отношения:

<u>Сотрудник</u>	<u>Номер телефона</u>
Иванов И. И.	283-56-82
Иванов И. И.	390-57-34
Петров П. П.	708-62-34

Вторая нормальная форма (2NF)

Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо (функционально полно) зависит от её потенциального ключа.

Если потенциальный ключ является простым, то есть состоит из единственного атрибута, то любая функциональная зависимость от него является неприводимой (полной). Если потенциальный ключ является составным, то согласно определению второй нормальной формы в отношении не должно быть неключевых атрибутов, зависящих от части составного потенциального ключа.

Вторая нормальная форма по определению запрещает наличие неключевых атрибутов, которые вообще не зависят от потенциального ключа. Таким образом, вторая нормальная форма в том числе запрещает создавать отношения как несвязанные (хаотические, случайные) наборы атрибутов.

Пусть в следующем отношении **первичный ключ** образует пара атрибутов { *Филиал компании, Должность* }:

R

Филиал компании	Должность	Зарплата	Наличие компьютера
Филиал в Томске	Уборщик	20000	Нет
Филиал в Москве	Программист	40000	Есть
Филиал в Томске	Программист	25000	Есть

Допустим, что зарплата зависит от филиала и должности, а наличие компьютера зависит только от должности.

Существует функциональная зависимость *Должность* → *Наличие компьютера*, в которой левая часть (детерминант) является лишь частью первичного ключа, что нарушает условие второй нормальной формы.

Для приведения к 2NF исходное отношение следует декомпозировать на два отношения:

R1

Филиал компании	Должность	Зарплата
Филиал в Томске	Уборщик	20000
Филиал в Томске	Программист	25000
Филиал в Москве	Программист	40000

R2

Должность	Наличие компьютера
Уборщик	Нет
Программист	Есть

Третья нормальная форма (3NF)

Переменная отношения находится в третьей нормальной форме тогда и только тогда, когда она находится во второй нормальной форме, и отсутствуют транзитивные функциональные зависимости неключевых атрибутов от ключевых.

Запоминающееся и, по традиции, наглядное резюме определения 3NF Коуда было дано Биллом Кентом: каждый неключевой атрибут «должен предоставлять информацию о ключе, полном ключе и ни о чём, кроме ключа».

Условие зависимости от «полного ключа» неключевых атрибутов обеспечивает то, что отношение находится во второй нормальной форме; а условие зависимости их от «ничего, кроме ключа» — то, что они находятся в третьей нормальной форме.

R1

Сотрудник	Отдел	Телефон
Гришин	Бухгалтерия	11-22-33
Васильев	Бухгалтерия	11-22-33
Петров	Снабжение	44-55-66

Каждый сотрудник относится исключительно к одному отделу; каждый отдел имеет единственный телефон. Атрибут *Сотрудник* является первичным ключом. Личных телефонов у сотрудников нет, и телефон сотрудника зависит исключительно от отдела.

В примере существуют следующие функциональные зависимости: *Сотрудник* → *Отдел*, *Отдел* → *Телефон*, *Сотрудник* → *Телефон*.

Переменная отношения R1 находится во **второй нормальной форме**, поскольку каждый атрибут имеет неприводимую функциональную зависимость от потенциального ключа *Сотрудник*.

Зависимость *Сотрудник* → *Телефон* является транзитивной, следовательно, отношение не находится в третьей нормальной форме.

В результате разделения R1 получаются две переменные отношения, находящиеся в 3NF:

R2

Отдел	Телефон	Сотрудник	Отдел
Бухгалтерия	11-22-33	Гришин	Бухгалтерия
Снабжение	44-55-66	Васильев	Бухгалтерия
		Петров	Снабжение

R3

Исходное отношение R1 при необходимости легко получается в результате **операции соединения** отношений R2 и R3.

Нормальная форма Бойса — Кодда (BCNF)

Переменная отношения находится в нормальной форме Бойса — Кодда (иначе — в усиленной третьей нормальной форме) тогда и только тогда, когда каждая её нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ.

Пусть R является переменной отношения, а X и Y — произвольными подмножествами множества атрибутов переменной отношения R . Y *функционально зависимо* от X тогда и только тогда, когда для любого допустимого значения переменной отношения R , если два кортежа переменной отношения R совпадают по значению X , они также совпадают и по значению Y . Подмножество X называют детерминантом, а Y — зависимой частью.

Функциональная зависимость тривиальна тогда и только тогда, когда её правая (зависимая) часть является подмножеством её левой части (детерминанта).

Функциональная зависимость называется неприводимой слева, если ни один атрибут не может быть опущен из её детерминанта без нарушения зависимости (иными словами, детерминант избыточен).

Ситуация, когда отношение будет находиться в 3NF, но не в BCNF, возникает, например, при условии, что отношение имеет два (или более) потенциальных ключа, которые являются составными, и между отдельными атрибутами таких ключей существует функциональная зависимость. Поскольку описанная зависимость не является транзитивной, то такая ситуация под определение 3NF не подпадает. На практике такие отношения встречаются достаточно редко, для всех прочих отношений 3NF и BCNF эквивалентны.

Предположим, рассматривается отношение, представляющее данные о бронировании теннисных кортов на день:

Бронирования кортов

Номер корта	Время начала	Время окончания	Тариф
1	09:30	10:30	«Корт 1 для членов клуба»
1	11:00	12:00	«Корт 1 для членов клуба»
1	14:00	15:30	«Корт 1 для не членов клуба»
2	10:00	11:30	«Корт 2 для не членов клуба»
2	11:30	13:30	«Корт 2 для не членов клуба»
2	15:00	16:30	«Корт 2 для членов клуба»

Таким образом, возможны следующие составные первичные ключи: {**Номер корта, Время начала**}, {**Номер корта, Время окончания**}, {**Тариф, Время начала**}, {**Тариф, Время окончания**}.

Отношение соответствует второй (2NF) и третьей (3NF) нормальной форме. Требования второй нормальной формы выполняются, так как все атрибуты входят в какой-то из потенциальных ключей, а неключевых атрибутов в отношении нет. Также нет и транзитивных зависимостей, что соответствует требованиям третьей нормальной формы. Тем не менее, существует функциональная зависимость **Тариф** → **Номер корта**, в которой левая часть (детерминант) не является потенциальным ключом отношения, то есть отношение *не находится в нормальной форме Бойса — Кодда*.

Недостатком данной структуры является то, что, например, по ошибке можно приписать тариф «Корт 1 для членов клуба» к бронированию второго корта, хотя он может относиться только к первому корту.

Можно улучшить структуру с помощью декомпозиции отношения на два, получив отношения, удовлетворяющие BCNF (подчёркнуты атрибуты, входящие в первичный ключ). Для большей наглядности к информации о тарифах добавлен атрибут **Для членов клуба:**

Тарифы

<u>Тариф</u>	<u>Номер корта</u>	<u>Для членов клуба</u>
«Корт 1 для членов клуба»	1	Да
«Корт 1 для не членов клуба»	1	Нет
«Корт 2 для членов клуба»	2	Да
«Корт 2 для не членов клуба»	2	Нет

Бронирования

<u>Тариф</u>	<u>Время начала</u>	<u>Время окончания</u>
«Корт 1 для членов клуба»	09:30	10:30
«Корт 1 для членов клуба»	11:00	12:00
«Корт 1 для не членов клуба»	14:00	15:30
«Корт 2 для не членов клуба»	10:00	11:30
«Корт 2 для не членов клуба»	11:30	13:00
«Корт 2 для членов клуба»	15:00	16:30

Четвёртая нормальная форма (4NF)

Переменная отношения находится в четвёртой нормальной форме, если она находится в нормальной форме Бойса — Кодда и не содержит нетривиальных многозначных зависимостей.

Предположим, что рестораны производят разные виды пиццы, а службы доставки ресторанов работают только в определенных районах города.

Составной первичный ключ соответствующей переменной отношения включает три атрибута: {Ресторан, Вид пиццы, Район доставки}.

Такая переменная отношения не соответствует 4НФ, так как существует следующая многозначная зависимость:

{Ресторан} \twoheadrightarrow {Вид пиццы}
{Ресторан} \twoheadrightarrow {Район доставки}

То есть, например, при добавлении нового вида пиццы придется внести по одному новому кортежу для каждого района доставки. Возможна логическая аномалия, при которой определенному виду пиццы будут соответствовать лишь некоторые районы доставки из обслуживаемых рестораном районов.

Для предотвращения аномалии нужно декомпозировать отношение, разместив независимые факты в разных отношениях. В данном примере следует выполнить декомпозицию на {Ресторан, Вид пиццы} и {Ресторан, Район доставки}.

Указанные выше многозначные зависимости в данном случае называются внедрёнными зависимостями.

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.

Доменно-ключевая нормальная форма (DKNF)

Переменная отношения находится в ДКНФ тогда и только тогда, когда каждое наложенное на неё ограничение является логическим следствием ограничений доменов и ограничений ключей, наложенных на данную переменную отношения.

Шестая нормальная форма (6NF)

Переменная отношения находится в шестой нормальной форме тогда и только тогда, когда она удовлетворяет всем нетривиальным зависимостям соединения. Из определения следует, что переменная находится в 6НФ тогда и только тогда, когда она неприводима, то есть не может быть подвергнута дальнейшей декомпозиции без потерь. Каждая переменная отношения, которая находится в 6НФ, также находится и в 5НФ.

Введена К. Дейтом в его книге, как обобщение пятой нормальной формы для хронологической базы данных.

Реляционная алгебра

Реляционная алгебра — замкнутая система операций над отношениями в реляционной модели данных. Операции реляционной алгебры также называют реляционными операциями.

Реляционная алгебра представляет собой набор таких операций над отношениями, что результат каждой из операций также является отношением. Это свойство алгебры называется **замкнутостью**.

Операции над одним отношением называются унарными, над двумя отношениями — бинарными, над тремя — тернарными (таковые практически неизвестны).

Пример унарной операции — проекция, пример бинарной операции — объединение.

Ограничения на операции

Некоторые реляционные операции, в частности, операции объединения, пересечения и вычитания, требуют, чтобы отношения имели совпадающие (одинаковые) заголовки (схемы). Это означает, что совпадают количество атрибутов, названия атрибутов и тип (домен) одноимённых атрибутов.

Некоторые отношения формально не являются совместимыми из-за различия в названиях атрибутов, но становятся таковыми после применения операции переименования атрибутов.

Переименование

Результатом применения операции переименования атрибутов является отношение с изменёнными именами атрибутов.

Синтаксис:

```
R RENAME Atr1, Atr2, ... AS NewAtr1, NewAtr2, ...
```

где

R — отношение

Atr1, Atr2, ... — исходные имена атрибутов

NewAtr1, NewAtr2, ... — новые имена атрибутов.

Пример 1.

Следующий оператор возвращает неименованное отношение, в котором атрибут *City_Num* переименован в *CityId* :

```
City RENAME City_Num AS CityId
```

Традиционно, определяют восемь реляционных операторов, объединенных в две группы.

Теоретико-множественные операторы:

- Объединение
- Пересечение
- Вычитание
- Декартово произведение

Специальные реляционные операторы:

- Выборка
- Проекция
- Соединение
- Деление

Будем называть отношения *совместимыми по типу*, если они имеют идентичные заголовки, а именно,

- Отношения имеют одно и то же множество имен атрибутов, т. е. для любого атрибута в одном отношении найдется атрибут с таким же наименованием в другом отношении,
- Атрибуты с одинаковыми именами определены на одних и тех же доменах.

Объединение

Отношение с тем же заголовком, что и у совместимых по типу отношений A и B, и телом, состоящим из кортежей, принадлежащих или A, или B, или обоим отношениям.

Синтаксис:

A UNION B

Пример 2. Пусть даны два отношения *A* и *B* с информацией о сотрудниках:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 1 Отношение A

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Пушников	2500
4	Сидоров	3000

Таблица 2 Отношение B

Объединение отношений *A* и *B* будет иметь вид:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000
2	Пушников	2500
4	Сидоров	3000

Таблица 3 Отношение A UNION B

Пересечение

Отношение с тем же заголовком, что и у отношений A и B, и телом, состоящим из кортежей, принадлежащих одновременно обоим отношениям A и B.

Синтаксис:

A INTERSECT B

Пример 3. Для тех же отношений A и B , что и в предыдущем примере пересечение имеет вид:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000

Таблица 4 Отношение $A \text{ INTERSECT } B$

Вычитание

Отношение с тем же заголовком, что и у совместимых по типу отношений A и B , и телом, состоящим из кортежей, принадлежащих отношению A и не принадлежащих отношению B .

Синтаксис:

$A \text{ MINUS } B$

Пример 4. Для тех же отношений A и B , что и в предыдущем примере вычитание имеет вид:

<i>Табельный номер</i>	<i>Фамилия</i>	<i>Зарплата</i>
2	Петров	2000
3	Сидоров	3000

Таблица 5 Отношение A MINUS B

Декартово произведение

Отношение $(A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_m)$, заголовок которого является сцеплением заголовков отношений $A(A_1, A_2, \dots, A_m)$ и $B(B_1, B_2, \dots, B_m)$, а тело состоит из кортежей, являющихся сцеплением кортежей отношений A и B :

$$(a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_m)$$

таких, что

$$(a_1, a_2, \dots, a_m) \in A,$$
$$(b_1, b_2, \dots, b_m) \in B.$$

Синтаксис:

A TIMES B

Пример 5. Пусть даны два отношения *A* и *B* с информацией о поставщиках и деталях:

Номер поставщика	Наименование поставщика
1	Иванов
2	Петров
3	Сидоров

Таблица 6 Отношение A (Поставщики)

Номер детали	Наименование детали
1	Болт
2	Гайка
3	Винт

Таблица 7 Отношение B (Детали)

Декартово произведение отношений *A* и *B* будет иметь вид:

Номер поставщика	Наименование поставщика	Номер детали	Наименование детали
1	Иванов	1	Болт
1	Иванов	2	Гайка
1	Иванов	3	Винт
2	Петров	1	Болт
2	Петров	2	Гайка
2	Петров	3	Винт
3	Сидоров	1	Болт
3	Сидоров	2	Гайка
3	Сидоров	3	Винт

Выборка (ограничение)

Отношение с тем же заголовком, что и у отношения *A*, и телом, состоящим из кортежей, значения атрибутов которых при подстановке в условие *c* дают значение ИСТИНА. *c* представляет собой логическое выражение, в которое могут входить атрибуты отношения *A* и/или скалярные выражения.

Синтаксис:

A WHERE *c*

Пример 6. Пусть дано отношение A с информацией о сотрудниках:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 9 Отношение A

Результат выборки A WHERE Зарплата < 3000 будет иметь вид:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000

Таблица 10 Отношение A WHERE Зарплата < 3000

Смысл операции выборки очевиден - выбрать кортежи отношения, удовлетворяющие некоторому условию. Таким образом, операция выборки дает "горизонтальный срез" отношения по некоторому условию.

Проекция

При выполнении проекции выделяется «вертикальная» вырезка отношения-операнда с естественным уничтожением потенциально возникающих кортежей-дубликатов.

Синтаксис:

$A[X, Y, \dots, Z]$

или

PROJECT A {x, y, ..., z}

Пример 7. Пусть дано отношение A с информацией о поставщиках, включающих наименование и месторасположение:

Номер поставщика	Наименование поставщика	Город поставщика
1	Иванов	Уфа
2	Петров	Москва
3	Сидоров	Москва
4	Сидоров	Челябинск

Таблица 11 Отношение A (Поставщики)

Проекция $A[\text{Город поставщика}]$ будет иметь вид:

Город поставщика
Уфа
Москва
Челябинск

Таблица 12 Отношение A[Город поставщика]

Соединение

Операция соединения отношений A и B по предикату P логически эквивалентна последовательному применению операций декартового произведения A и B и выборки по предикату P . Если в отношениях имеются атрибуты с одинаковыми наименованиями, то перед выполнением соединения такие атрибуты необходимо переименовать.

Синтаксис:

$(A \text{ TIMES } B) \text{ WHERE } P$

Деление

Отношение с заголовком (X_1, X_2, \dots, X_n) и телом, содержащим множество кортежей (x_1, x_2, \dots, x_n) , таких, что для всех кортежей $(y_1, y_2, \dots, y_m) \in B$ в отношении $A(X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_m)$ найдется кортеж $(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$.

Синтаксис:

$A \text{ DIVIDE } B$