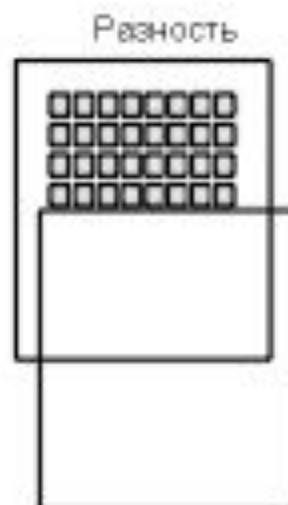
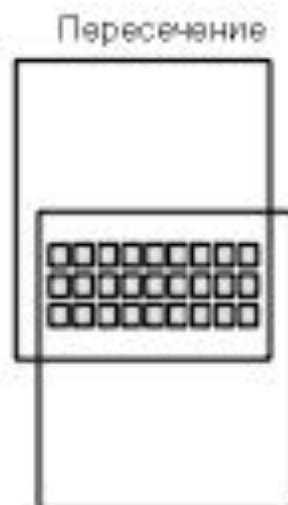
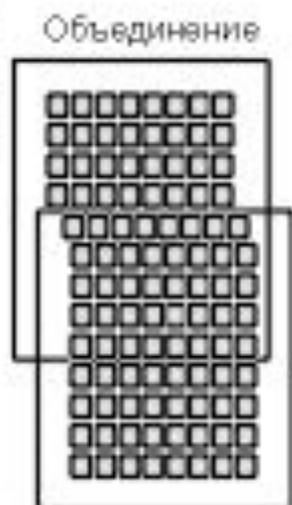
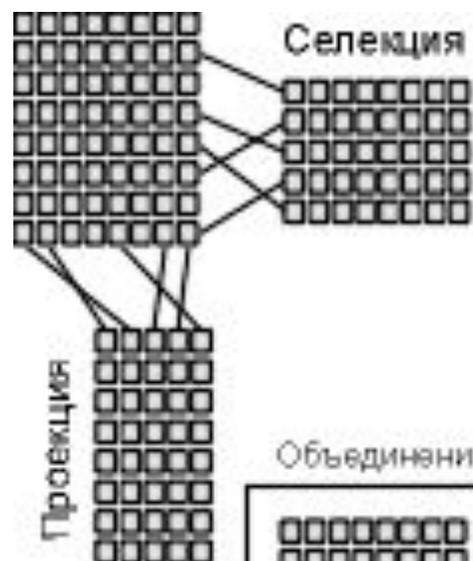


Традиционные операции реляционной алгебры

профессор Федин Ф.О.



Традиционные операции:

- * **Объединение** - возвращает отношение, содержащее все кортежи, принадлежащие или одному из двух определенных отношений, или обоим
- * **Пересечение** – возвращает отношение, содержащее все кортежи, принадлежащие одновременно двум определенным отношениям
- * **Вычитание** – возвращает отношение, содержащее все кортежи, которые принадлежат первому из двух определенных отношений и не принадлежат второму
- * **Расширенное декартово произведение** – возвращает отношение, содержащее всевозможные кортежи, являющиеся сочетанием двух кортежей, принадлежащих соответственно двум определенным отношениям

Специальные операции:

* **Выборка (ограничение)** – возвращает отношение, содержащее все кортежи из определенного отношения, удовлетворяющие определенным условиям

* **Проекция** – возвращает отношение, содержащее все кортежи (называемые как подкортежи) определенного отношения после исключения из него некоторых атрибутов

* **Соединение (естественное)** - возвращает отношение, кортежи которого – это сочетание двух кортежей (принадлежащих соответственно двум определенным отношениям), имеющих общее значение для одного или нескольких атрибутов этих двух отношений (и такие общие значения в результирующем кортеже появляются только один раз)

* **Деление** - для двух отношений, бинарного и унарного, возвращает отношение, содержащее все значения одного атрибута бинарного отношения, соответствующее (в другом атрибуте) всем значениям в унарном отношении

Операции объединения, пересечения и вычитания требуют от операндов совместимости по типу.

Два отношения совместимы по типу, если:

1. каждое из них имеет одно и то же множество атрибутов
2. возможно такое упорядочение атрибутов в схемах, что на одинаковых местах будут находиться сравнимые атрибуты

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПЕОЮЛ Сидоров А.С.	Находка
P ₄	ОАО «Владхлеб»	Владивосток

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример: Имеются следующие отношения:

Отношение **Продукты1** содержит продукты, имеющиеся в магазине

Отношение **Продукты2** содержит продукты, поставляемые поставщиком P₂

Отношение **Поставщики** содержит поставщиков продуктов

Отношение **ВидПродукта** содержит виды продуктов

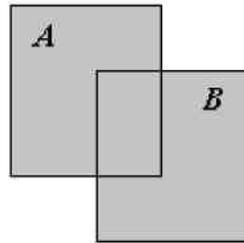
Первые три отношения имеют одинаковую степень, т.е. выполняется первое условие совместимости по типу.

Второе условие выполняется только для отношений **Продукты1** и **Продукты2**, т.е. только эти отношения совместимы по типу, а значит с ними можно выполнять операции объединения, пересечения и вычитания.

Теоретико-множественные операции реляционной алгебры

ОБЪЕДИНЕНИЕ

Объединением двух совместимых по типу отношений A и B называется отношение с тем же заголовком, как в исходных отношениях, и с телом, состоящим из множества всех кортежей, принадлежащих A или B или обоим отношениям (за исключением повторяющихся).



Пусть заданы два отношения $A = \{a\}$, $B = \{b\}$, где a и b – соответственно кортежи отношений A и B , то объединение

$$A \cup B = \{c \mid c \in A \vee c \in B\},$$

где c – кортеж нового отношения,

\vee – операция логического сложения «ИЛИ».

ОБЪЕДИНЕНИЕ

Пример:

Объединим отношения **Продукты1** (содержащее продукты, имеющиеся в магазине) и **Продукты2** (содержащее продукты, поставляемые поставщиком P_2).

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P_1
2	Соль	P_1
13	Мука	P_1
26	Рис	P_2
58	Гречка	P_2
130	Крупа манная	P_2
162	Пшено	P_2
474	Молоко	P_3
891	Кефир	P_3

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P_2
35	Перловка	P_2
58	Гречка	P_2
130	Крупа манная	P_2
162	Пшено	P_2
200	Крупа ячневая	P_2

Поставщики

КодП	Наименование	Город
P_1	ООО «Восток»	Владивосток
P_2	ОАО «Приморье»	Уссурийск
P_3	ПБЮЛ Сидоров А.С.	Находка
P_4	ОАО «Владхлеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Результатом объединения станет отношение R_1 , содержащее продукты, которые или имеются в магазине или поставляются поставщиком P_2 (либо и то и другое).

R_1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P_1
2	Соль	P_1
13	Мука	P_1
26	Рис	P_2
58	Гречка	P_2
130	Крупа манная	P_2
162	Пшено	P_2
474	Молоко	P_3
891	Кефир	P_3
35	Перловка	P_2
200	Крупа ячневая	P_2

Внимание! Дублирующие кортежи исключены из результирующего отношения R_1 .

$$R_1 = \text{Продукты1} \cup \text{Продукты2}$$

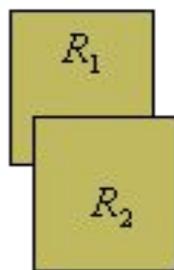
ОБЪЕДИНЕНИЕ

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

Объединение / UNION /

<u>Обозначение</u>	<u>Определение</u>	<u>LEAP</u>
$R_1 \cup R_2$	$\{r: r \in R_1 \vee r \in R_2\}$	$r = (R1) \text{ union } (R2)$

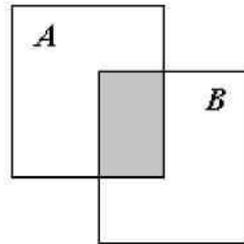
Пример:



$$R[Q,T] \cup S = \begin{bmatrix} 5 & a \\ 3 & a \\ 9 & a \\ 1 & b \\ 2 & b \\ 4 & b \end{bmatrix} \cup \begin{bmatrix} 5 & a \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \\ 1 & b \end{bmatrix} = \begin{bmatrix} 5 & a \\ 3 & a \\ 9 & a \\ 1 & b \\ 2 & b \\ 4 & b \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \end{bmatrix}$$

ПЕРЕСЕЧЕНИЕ

Пересечением двух совместимых по типу отношений A и B называется отношение с тем же заголовком, как в исходных отношениях, и с телом, состоящим из множества всех кортежей, принадлежащих одновременно обоим отношениям A и B .



$$A \cap B = \{c \mid c \in A \wedge c \in B\},$$

где \wedge – операция логического умножения (логическое «И»).

ПЕРЕСЕЧЕНИЕ

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПБООЛ Сидоров А.С.	Находка
P ₄	ОАО «Владхлеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример:

Пересечением отношений *Продукты1* и *Продукты2* станет отношение *R2*, содержащее продукты, имеющиеся в магазине и поставляемые поставщиком *P₂*.

$$R2 = \text{Продукты1} \cap \text{Продукты2}$$

R2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂

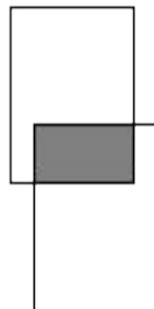
ПЕРЕСЕЧЕНИЕ

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

Пересечение / INTERSECT /

<u>Обозначение</u>	<u>Определение</u>	<u>LEAP</u>
$R_1 \cap R_2$	$\{r: r \in R_1 \wedge r \in R_2\}$	$r = (R1) \text{ intersect } (R2)$

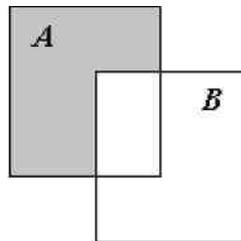
Пример:



$$R[Q,T] \cap S = \begin{bmatrix} 5 & a \\ 3 & a \\ 9 & a \\ 1 & b \\ 2 & b \\ 4 & b \end{bmatrix} \cap \begin{bmatrix} 5 & a \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \\ 1 & b \end{bmatrix} = \begin{bmatrix} 5 & a \\ 1 & b \end{bmatrix}$$

ВЫЧИТАНИЕ

Вычитанием двух совместимых по типу отношений A и B называется отношение с тем же заголовком, как в исходных отношениях, и с телом, состоящим из множества всех кортежей, принадлежащих отношению A и не принадлежащих отношению B .



$$A \setminus B = \{c \mid c \in A \wedge c \notin B\}$$

ВЫЧИТАНИЕ

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПБОЮЛ Сидоров А.С.	Находка
P ₄	ОАО «Владклеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример:

При вычитании отношения *Продукты2* из отношения *Продукты1* получится отношение R_3 , содержащее продукты, имеющиеся в магазине, кроме тех продуктов, которые поставяет поставщик P_2 .

При вычитании отношения *Продукты1* из отношения *Продукты2* получится другое отношение R_4 (поскольку операция вычитания не коммутативная). Отношение R_4 будет содержать продукты, поставяемые поставщиком P_2 , кроме тех продуктов, которые имеются в магазине.

$$R_3 = \text{Продукты1} \setminus \text{Продукты2}$$

R3

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
474	Молоко	P ₃
891	Кефир	P ₃

$$R_4 = \text{Продукты2} \setminus \text{Продукты1}$$

R4

КодПродукта	Продукт	КодПоставщика
35	Перловка	P ₂
200	Крупа ячневая	P ₂

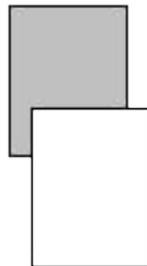
ВЫЧИСЛЕНИЕ

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

Разность / SET DIFFERENCE /

<u>Обозначение</u>	<u>Определение</u>	<u>LEAP</u>
$R_1 - R_2$	$\{r: r \in R_1 \wedge r \notin R_2\}$	$r = (R1) \text{ difference } (R2)$ $r = (R1) \text{ minus } (R2)$

Пример:



$$R[Q,T] - S = \begin{bmatrix} \cancel{5} & \cancel{a} \\ 3 & a \\ 9 & a \\ 1 & \cancel{b} \\ 2 & b \\ 4 & b \end{bmatrix} \cap \begin{bmatrix} \boxed{5} & \boxed{a} \\ 10 & b \\ 15 & c \\ 2 & d \\ 6 & a \\ \boxed{1} & \boxed{b} \end{bmatrix} = \begin{bmatrix} 3 & a \\ 9 & a \\ 2 & b \\ 4 & b \end{bmatrix}$$

РАСШИРЕННОЕ ДЕКАРТОВО ПРОИЗВЕДЕНИЕ

Прежде чем определить саму операцию, введем дополнительно понятие конкатенации, или сцепления, кортежей.

Сцеплением, или конкатенацией, кортежей $c = \langle c_1, c_2, \dots, c_n \rangle$ и $q = \langle q_1, q_2, \dots, q_m \rangle$ называется кортеж, полученный добавлением значений второго в конец первого. Сцепление кортежей c и q обозначается как (c, q) .

$$(c, q) = \langle c_1, c_2, \dots, c_n, q_1, q_2, \dots, q_m \rangle,$$

где n – число элементов в первом кортеже c ,

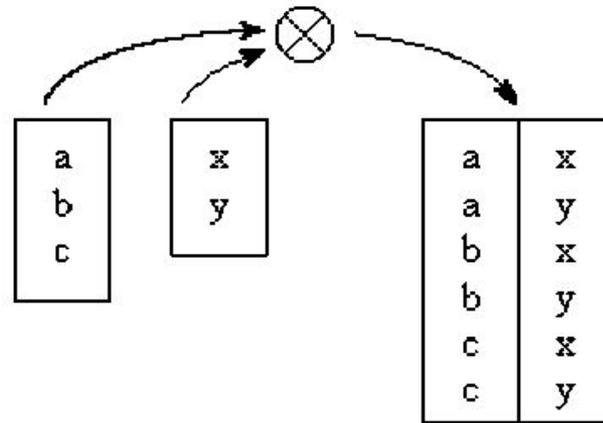
m – число элементов во втором кортеже q .

Все предыдущие операция не меняли степени или арности отношений – это следует из определения эквивалентности схем отношений.

Операция расширенного декартова произведения меняет степень результирующего отношения.

Расширенное декартово произведение двух отношений A и B , где A и B не имеют общих атрибутов, определяется как отношение с заголовком, который представляет собой сцепление двух заголовков исходных отношений A и B , и телом, состоящим из множества всех кортежей s , таких, что s представляет собой сцепление кортежа a , принадлежащего отношению A , и кортежа b , принадлежащего отношению B .

РАСШИРЕННОЕ ДЕКАРТОВО ПРОИЗВЕДЕНИЕ



$$A \times B = \{c \mid c \in A \wedge c \in B\}$$

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

Пример:

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

РАСШИРЕННОЕ ДЕКАРТОВО ПРОИЗВЕДЕНИЕ

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПБООЛ Сидоров А.С.	Находка
P ₄	ОАО «Владхлеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример:

Декартовым произведением отношений *Поставщики* и *ВидПродукта* будет отношение R_5 . Отношение R_5 соответствует ситуации, когда **все** поставщики поставляют **все** виды продуктов.

$R_5 = \text{Поставщики} \otimes \text{ВидПродукта}$

R_5

КодП	Наименование	Город	КодВида	Вид
P ₁	ООО «Восток»	Владивосток	1	Молочная
P ₁	ООО «Восток»	Владивосток	2	Мясная
P ₁	ООО «Восток»	Владивосток	3	Хлебопродукция
P ₂	ОАО «Приморье»	Уссурийск	1	Молочная
P ₂	ОАО «Приморье»	Уссурийск	2	Мясная
P ₂	ОАО «Приморье»	Уссурийск	3	Хлебопродукция
P ₃	ПБООЛ Сидоров А.С.	Находка	1	Молочная
P ₃	ПБООЛ Сидоров А.С.	Находка	2	Мясная
P ₃	ПБООЛ Сидоров А.С.	Находка	3	Хлебопродукция
P ₄	ОАО «Владхлеб»	Владивосток	1	Молочная
P ₄	ОАО «Владхлеб»	Владивосток	2	Мясная
P ₄	ОАО «Владхлеб»	Владивосток	3	Хлебопродукция

$$A \setminus B = \{c \mid c \in A \wedge c \notin B\}$$

Специальные операции реляционной алгебры

профессор Федин Ф.О.

ВЫБОРКА (ограничение, горизонтальное подмножество)

Пусть α - булевское выражение, составленное из термов сравнения с помощью связок И (\wedge), ИЛИ (\vee), НЕ ($-$) и, возможно, скобок.

В качестве термов сравнения допускаются:

1. терм $A \theta a$,

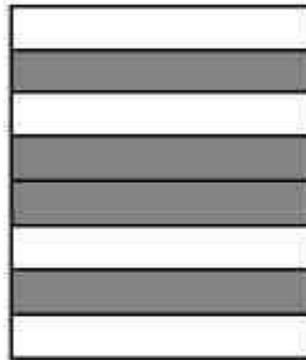
где A - имя некоторого атрибута, принимающего значения из домена D ;

a - константа, взятая из того же домена D , $a \in D$;

θ - одна из допустимых для данного домена D операций сравнения ($=$, \neq , $<$, \leq , $>$, \geq);

2. терм $A \theta B$,

где A , B - имена некоторых θ -сравнимых атрибутов, то есть атрибутов, принимающих значение из одного и того же домена D .



ВЫБОРКА

$$A \setminus B = \{c \mid c \in A \wedge c \notin B\}$$

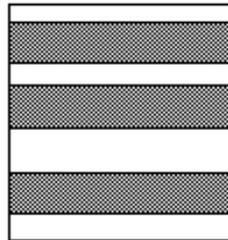
ВЫБОРКА

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

Пример:



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

ВЫБОРКА

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПБОЮЛ Сидоров А.С.	Находка
P ₄	ОАО «Владхлеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример:

- Результатом выборки продуктов, поставляемых поставщиком P₃, из отношения *Продукты1* будет отношение R₆.
- Результатом выборки Владивостокских поставщиков из отношения *Поставщики* будет отношение R₇.

а) R₆ = Продукты1 [КодПоставщика = "P₃"]

R6

КодПродукта	Продукт	КодПоставщика
474	Молоко	P ₃
891	Кефир	P ₃

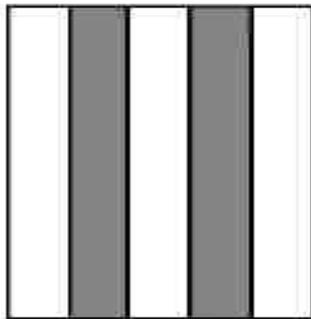
б) R₇ = Поставщики [Город = "Владивосток"]

R7

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₄	ОАО «Владхлеб»	Владивосток

ПРОЕКЦИЯ (вертикальное подмножество)

Проекцией отношения A по атрибутам X, Y, \dots, Z , где каждый из атрибутов принадлежит отношению $A(A[X, Y, \dots, Z])$, называется отношение с заголовком $\{X, Y, \dots, Z\}$ и телом, содержащим множество всех кортежей $\{X:x, Y:y, \dots, Z:z\}$, таких, для которых в отношении A значение атрибута X равно x , атрибута Y равно y , ..., атрибута Z равно z .



С помощью оператора проекции получено «вертикальное» подмножество данного отношения, т.е. подмножество, получаемое исключением всех атрибутов, не указанных в списке атрибутов, и последующим исключением дублирующих кортежей (подкортежей) из того, что осталось.

Никакой атрибут не может быть указан в списке атрибутов более одного раза.

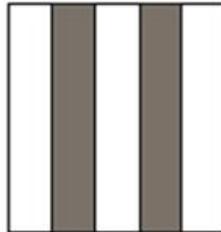
ПРОЕКЦИЯ

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:



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

ПРОЕКЦИЯ

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПБООЛ Сидоров А.С.	Находка
P ₄	ОАО «Владжлеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример:

- Проекцией отношения *Продукты1* по атрибуту *КодПоставщика* будет отношение R_8 . Обратите внимание, что дублирующие кортежи исключены из отношения R_8 .
- Проекцией отношения *Поставщики* по атрибуту *Город* будет отношение R_9 .

a) $R_8 = \text{Продукты1} [\text{КодПоставщика}]$

R_8

КодПоставщика
P ₁
P ₂
P ₃

b) $R_9 = \text{Поставщики} [\text{Город}]$

R_9

Город
Владивосток
Уссурийск
Находка

- Довольно часто операция проекции используется в сочетании с другими операциями. Например, нужно выбрать названия поставщиков из Владивостока (на основе отношения *Поставщики*). Сначала выполняется операция выборки, а затем – проекции.

c) $R_7 = \text{Поставщики} [\text{Город} = \text{“Владивосток”}]$

$R_{10} = R_7 [\text{Наименование}]$

или $R_{10} = (\text{Поставщики} [\text{Город} = \text{“Владивосток”}]) [\text{Наименование}]$

R_7

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₄	ОАО «Владжлеб»	Владивосток

R_{10}

Наименование
ООО «Восток»
ОАО «Владжлеб»

СОЕДИНЕНИЕ (естественное, условное)

Операция соединения имеет несколько разновидностей. Однако наиболее важным является естественное соединение, поэтому часто для обозначения именно естественного соединения используют общий термин «соединение».

Пусть отношения A и B имеют заголовки:

$\{X_1, X_2, \dots, X_m, Y_1, Y_2, \dots, Y_n\}$

и

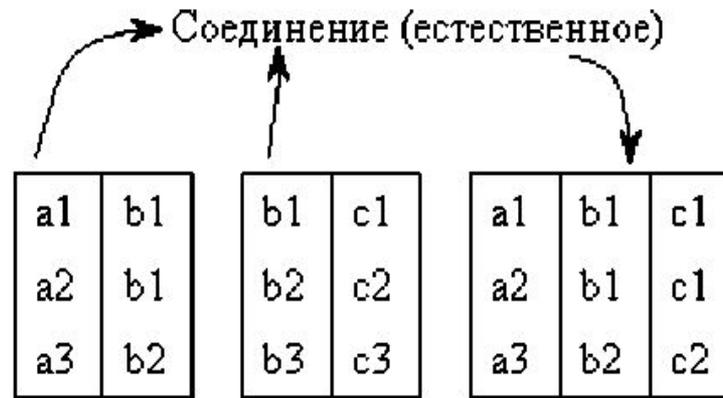
$\{Y_1, Y_2, \dots, Y_n, Z_1, Z_2, \dots, Z_p\}$ соответственно;

т.е. атрибуты Y_1, Y_2, \dots, Y_n (и только они) – общие для двух отношений;

X_1, X_2, \dots, X_m – остальные атрибуты отношения A ; Z_1, Z_2, \dots, Z_p – остальные атрибуты отношения B . Предположим также, что соответствующие атрибуты (т.е. атрибуты с одинаковыми именами) определены на одном и том же домене. Будем рассматривать выражения $\{X_1, X_2, \dots, X_m\}$, $\{Y_1, Y_2, \dots, Y_n\}$, $\{Z_1, Z_2, \dots, Z_p\}$ как три составных атрибута X, Y, Z соответственно.

СОЕДИНЕНИЕ

Естественным соединением отношений A и B называется отношение с заголовком $\{X, Y, Z\}$ и телом, содержащим множество всех кортежей $\{X:x, Y:y, Z:z\}$, таких, для которых в отношении A значение атрибута X равно x , а атрибута Y равно y , и в отношении B значение атрибута Y равно y , а атрибута Z равно z .



Если отношения A и B не имеют общих атрибутов, то естественное соединение превращается в декартово произведение.

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.

Соединение / JOIN /

<u>Обозначение</u>	<u>Определение</u>
$R_1 [A_1 \theta A_2] R_2$	$\{(r_1 r_2) : r_1 \in R_1 \wedge r_2 \in R_2 \wedge (r_1[A_1] \theta r_2[A_2])\}$

LEAP : $r = \text{join } (R1) (R2) ((\text{cond}) \text{ bool } (\text{cond}))$

Пример:

$$P[D_3 = D_4]Q = \begin{bmatrix} 1 & 11 & x & x & 1 \\ 1 & 11 & x & x & 2 \\ 2 & 11 & y & y & 1 \\ 4 & 12 & x & x & 1 \\ 4 & 12 & x & x & 2 \end{bmatrix} = \begin{bmatrix} 1 & 11 & x & 1 \\ 1 & 11 & x & 2 \\ 2 & 11 & y & 1 \\ 4 & 12 & x & 1 \\ 4 & 12 & x & 2 \end{bmatrix}$$

СОЕДИНЕНИЕ

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 $СОТР_НОМ = ОТДЕЛ_НАЧ$.

$C1 = СОТРУДНИКИ [СОТР_НОМ = ОТД_НАЧ] ОТДЕЛЫ$

(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.

СОЕДИНЕНИЕ

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПЕОЮЛ Сидоров А.С.	Находка
P ₄	ОАО «Владхлеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример:

Рассмотрим отношения *Продукты1* и *Поставщики*. Атрибуты *КодПоставщика* и *КодП* определены на одном и том же домене кодов поставщиков. Поскольку при естественном соединении также требуется, чтобы общие атрибуты соединяемых отношений имели одинаковые имена, переименуем атрибут *КодП* отношения *Поставщики* в *КодПоставщика*. Тогда естественным соединением отношений *Продукты1* и *Поставщики* по атрибуту *КодПоставщика* будет отношение R11.

R11 = Продукты1 [Продукты1. КодПоставщика = Поставщики.КодПоставщика]
Поставщики

R11

КодПродукта	Продукт	КодПоставщика	Наименование	Город
1	Сахар	P ₁	ООО «Восток»	Владивосток
2	Соль	P ₁	ООО «Восток»	Владивосток
13	Мука	P ₁	ООО «Восток»	Владивосток
26	Рис	P ₂	ОАО «Приморье»	Уссурийск
58	Гречка	P ₂	ОАО «Приморье»	Уссурийск
130	Крупа манная	P ₂	ОАО «Приморье»	Уссурийск
162	Пшено	P ₂	ОАО «Приморье»	Уссурийск
474	Молоко	P ₃	ПЕОЮЛ Сидоров А.С.	Находка
891	Кефир	P ₃	ПЕОЮЛ Сидоров А.С.	Находка

СОЕДИНЕНИЕ

Условное соединение (или θ -соединение) используется, когда необходимо соединить два отношения на основе некоторых условий, отличных от эквивалентности.

Пусть отношения A и B не имеют общих имен атрибутов, и θ определяется как в операции выборки. Тогда **условным соединением** отношения A по атрибуту X с отношением B по атрибуту Y называется отношение с заголовком, который представляет собой сцепление двух заголовков исходных отношений A и B (как и при операции декартова произведения), и с телом, содержащим множество кортежей t , таких что t принадлежит этому декартову произведению и вычисление условия « $X \theta Y$ » дает значение «истина» для этого кортежа.

Атрибуты X и Y должны быть определены на одном и том же домене, а операция должна иметь смысл для этого домена.

СОЕДИНЕНИЕ

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПЕОЮЛ Сидоров А.С.	Находка
P ₄	ОАО «Владхлеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример:

Получить названия продуктов (отношение *Продукты1*), поставляемых поставщиками из Владивостока (отношение *Поставщики*). По сути, в этом примере необходимо использовать две операции: условного соединения – для получения непосредственно списка продуктов, поставляемых Владивостокскими поставщиками (*R12*); и проекции – для получения только названий продуктов (*R13*).

$R12 = \text{Продукты1} [(\text{Продукты1.КодПоставщика} = \text{Поставщики.КодП}) \wedge \text{Поставщики.Город} = \text{«Владивосток»}]$

ИЛИ

$R13 = (\text{Продукты1} [(\text{Продукты1.КодПоставщика} = \text{Поставщики.КодП}) \wedge \text{Поставщики.Город} = \text{«Владивосток»}] \text{Продукты1} [\text{Продукт}])$

R12

КодПродукта	Продукт	КодПоставщика	КодП	Наименование	Город
1	Сахар	P ₁	P ₁	ООО «Восток»	Владивосток
2	Соль	P ₁	P ₁	ООО «Восток»	Владивосток
13	Мука	P ₁	P ₁	ООО «Восток»	Владивосток

R13

Продукт
Сахар
Соль
Мука

ДЕЛЕНИЕ

Пусть отношения A и B имеют заголовки:

$\{X_1, X_2, \dots, X_m, Y_1, Y_2, \dots, Y_n\}$

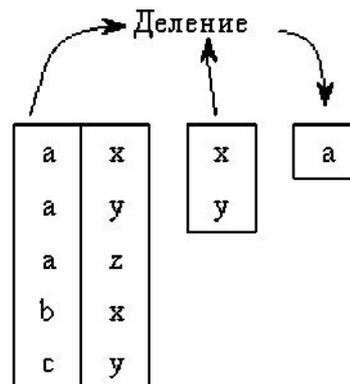
и

$\{Y_1, Y_2, \dots, Y_n\}$ соответственно;

т.е. атрибуты Y_1, Y_2, \dots, Y_n – общие для двух отношений, и отношение A имеет дополнительные атрибуты X_1, X_2, \dots, X_m , а отношение B не имеет дополнительных атрибутов.

Отношения A и B представляют соответственно делимое и делитель.

Предположим также, что соответствующие атрибуты (т.е. атрибуты с одинаковыми именами) определены на одном и том же домене. Пусть выражения $\{X_1, X_2, \dots, X_m\}$ и $\{Y_1, Y_2, \dots, Y_n\}$ обозначают два составных атрибута X и Y соответственно.



ДЕЛЕНИЕ

Делением отношений A и B называется отношение с заголовком $\{X\}$ и телом, содержащим множество всех кортежей $\{X:x\}$ таких, что существует кортеж $\{X:x, Y:y\}$, который принадлежит отношению A для всех кортежей $\{Y:y\}$, принадлежащих отношению B .

Нестрого это можно сформулировать так: результат содержит такие X -значения из отношения A , для которых соответствующие Y -значения (из A) включают все Y -значения из отношения B .

Если запрос на естественном языке включает слово «все» («получить поставщиков, поставляющих все виды продуктов»), то почти наверняка потребуется операция деления.

ДЕЛЕНИЕ

Пусть отношение R , называемое делимым, содержит атрибуты (A_1, A_2, \dots, A_n) . Отношение S – делитель содержит подмножество атрибутов $A: (A_1, A_2, \dots, A_k)$ ($k < n$). Результирующее отношение C определено на атрибутах отношения R , которых нет в S , т.е. $A_{k+1}, A_{k+2}, \dots, A_n$. Кортежи включаются в результирующее отношение C только в том случае, если его декартово произведение с отношением S содержится в делимом R .

Деление / DIVISION /

Обозначение	Определение	LEAP:
$R_1 [A_1 \div A_2] R_2$	$\{r[A_1] : r \in R_1 \wedge R_2[A_2] \subseteq g_R(r[A_1])\}$	не поддерживается

Пример:

пусть

$$R_1(A_1, A_2) = \begin{bmatrix} a & x \\ a & y \\ a & z \\ b & x \\ c & y \end{bmatrix}$$

$$R_2(A_3, A_4) = \begin{bmatrix} 1 & x \\ 2 & x \\ 1 & y \end{bmatrix}$$

тогда

$$R_1[A_2 \div A_4] = \begin{bmatrix} a & x \\ a & y \\ a & z \\ b & x \\ c & y \end{bmatrix} \div \begin{bmatrix} x \\ y \end{bmatrix} = [a]$$

ДЕЛЕНИЕ

Продукты1

КодПродукта	Продукт	КодПоставщика
1	Сахар	P ₁
2	Соль	P ₁
13	Мука	P ₁
26	Рис	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
474	Молоко	P ₃
891	Кефир	P ₃

Продукты2

КодПродукта	Продукт	КодПоставщика
26	Рис	P ₂
35	Перловка	P ₂
58	Гречка	P ₂
130	Крупа манная	P ₂
162	Пшено	P ₂
200	Крупа ячневая	P ₂

Поставщики

КодП	Наименование	Город
P ₁	ООО «Восток»	Владивосток
P ₂	ОАО «Приморье»	Уссурийск
P ₃	ПЕОЮЛ Сидоров А.С.	Находка
P ₄	ОАО «Владхлеб»	Владивосток

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

Пример:

Пусть отношение R_{14} содержит поставщиков и виды поставляемых ими продуктов, а отношение *ВидПродукта* содержит виды продуктов. Тогда, чтобы получить поставщиков поставляющих **ВСЕ** виды продуктов, необходимо отношение R_{14} разделить на отношение *ВидПродукта* по атрибуту *КодВида*.

R14

КодП	Наименование	КодВида
P ₁	ООО «Восток»	1
P ₁	ООО «Восток»	2
P ₂	ОАО «Приморье»	1
P ₂	ОАО «Приморье»	2
P ₂	ОАО «Приморье»	3
P ₃	ПЕОЮЛ Сидоров А.С.	3
P ₄	ОАО «Владхлеб»	1
P ₄	ОАО «Владхлеб»	2
P ₄	ОАО «Владхлеб»	3

ВидПродукта

КодВида	Вид
1	Молочная
2	Мясная
3	Хлебопродукция

R15

КодП	Наименование
P ₂	ОАО «Приморье»
P ₄	ОАО «Владхлеб»

INDICACIONES

Примеры, демонстрирующие возможности операций реляционной алгебры:

И. Даны следующие отношения:

R_1 (Таб№, ФИО, Должность, Отдел) — список сотрудников по отделам

R_2 (Должность, Оклад) — должностные оклады

R_3 (Должность, Отдел) — штатные должности по отделам, т.е. какие должности должны быть в каждом отделе.

Сотрудник может занимать несколько должностей в одном и том же или разных отделах. Составить формулы для следующих запросов:

1. Табельные номера и ФИО сотрудников отдела “Бухгалтерия”

2. $(R_1[\text{Отдел} = \text{«Бухгалтерия»}])[\text{Таб№}, \text{ФИО}]$

3. ФИО сотрудников с окладом больше 3000 руб

4. $(R_1[R_1.\text{Должность} = R_2.\text{Должность} \wedge R_2.\text{Оклад} > 3000]R_2)[\text{ФИО}]$

5. Список вакантных должностей в отделах

6. $R_3 \setminus (R_1[\text{Должность}, \text{Оклад}])$

7. Отделы, не имеющие по штату должность “Мастер”

8. $R_3 \setminus R_3[\text{Должность} = \text{«Мастер»}]$

9. ФИО сотрудников, занимающих больше одной должности

$(R_1[R_1.\text{ФИО} = R_1'.\text{ФИО} \wedge R_1.\text{Должность} \neq R_1'.\text{Должность}]R_1')[\text{ФИО}]$

$R_4(R_1[\text{ФИО}]) \setminus R_4$

Примеры, демонстрирующие возможности операций реляционной алгебры:

И. Даны следующие отношения:

R_1 (Таб№, ФИО, Должность, Отдел) — список сотрудников по отделам

R_2 (Должность, Оклад) — должностные оклады

R_3 (Должность, Отдел) — штатные должности по отделам, т.е. какие должности должны быть в каждом отделе.

Сотрудник может занимать несколько должностей в одном и том же или разных отделах. Составить формулы для следующих запросов:

1. Табельные номера и ФИО сотрудников отдела “Бухгалтерия”

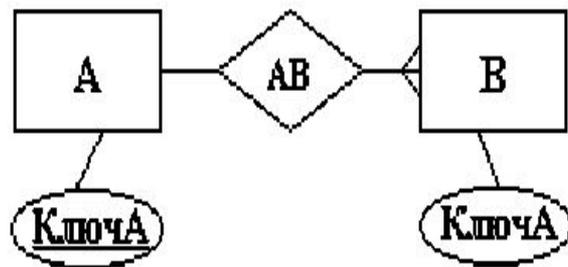
$(R_1[\text{Отдел} = \text{«Бухгалтерия»}]) \text{ [Таб№, ФИО]}$

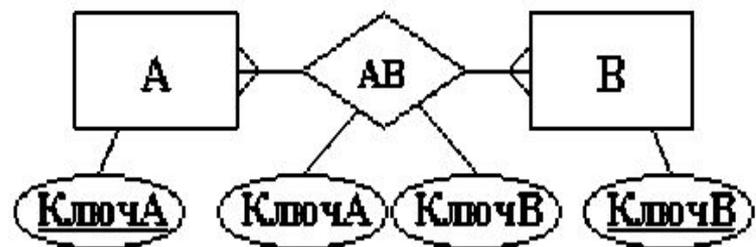
1. ФИО сотрудников с окладом больше 3000 руб

$(R_1[R_1.\text{Должность} = R_2.\text{Должность} \wedge R_2.\text{Оклад} > 3000]R_2) \text{ [ФИО]}$

1. Список вакантных должностей в отделах

$R_3 \setminus (R_1[\text{Должность, Оклад}])$







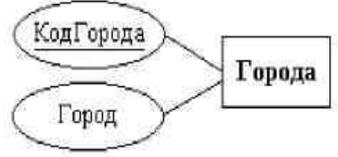
Продукты
<u>КодПрод</u>
Продукт
ЕдИзм
СрокХран(дни)
УсловияХран



Поставщики
<u>КодПост</u>
Поставщик
КодГорода
Адрес
ФИОдиректора
Телефон
Факс



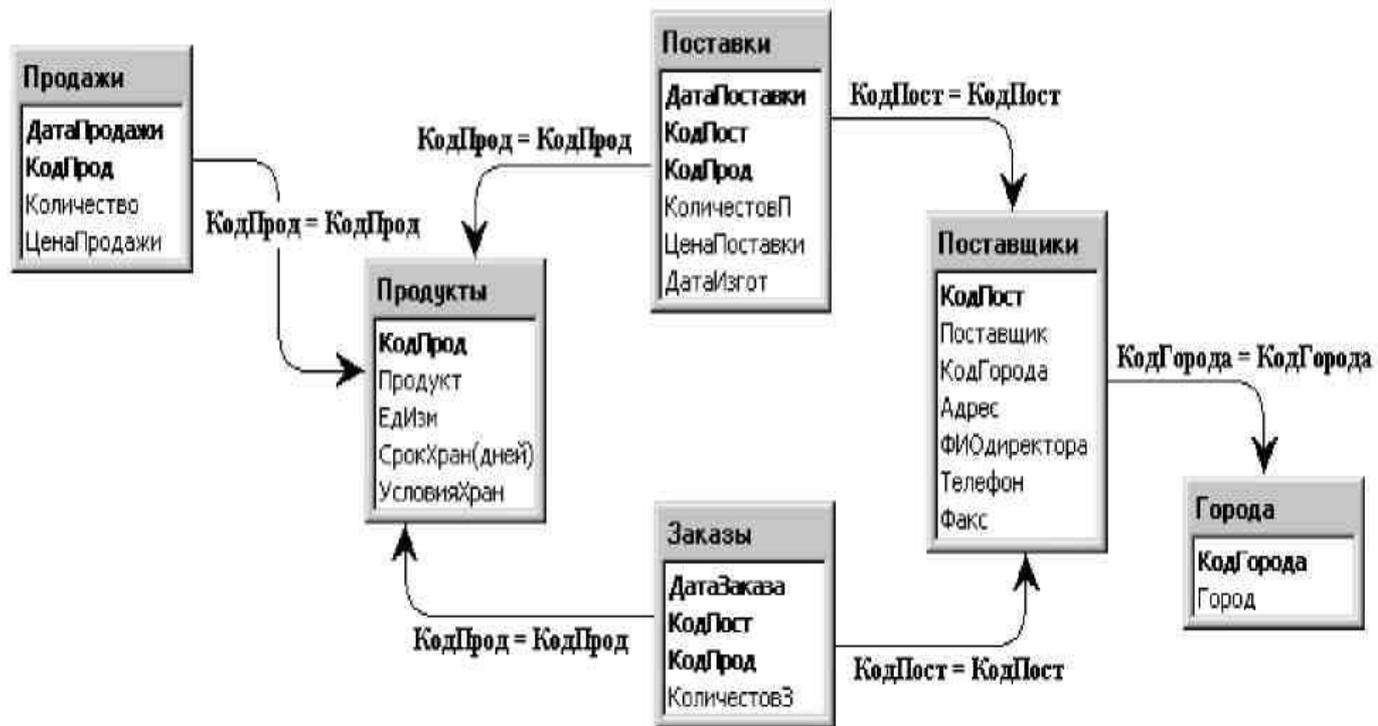
Продажи
<u>ДатаПродажи</u>
КодПрод
Количество
ЦенаПродажи



Города
<u>КодГорода</u>
Город

Атрибут	Тип данных (СУБД Access)	Допустимость Null-значений	Первичный ключ	Внешний ключ
Отношение Продукты				
<i>КодПрод</i>	Целое	Нет	√	
<i>Продукт</i>	Текстовый (30)	Нет		
<i>ЕдИзм</i>	Текстовый (5)	Да		
<i>СрокХран(дней)</i>	Целое	Да		
<i>УсловияХран</i>	Текстовый (200)	Да		
Отношение Поставщики				
<i>КодПост</i>	Целое	Нет	√	
<i>Поставщик</i>	Текстовый (50)	Нет		
<i>КодГорода</i>	Целое	Да		√
<i>Адрес</i>	Текстовый (100)	Да		
<i>ФИОдиректора</i>	Текстовый (50)	Да		
<i>Телефон</i>	Текстовый (15)	Да		
<i>Факс</i>	Текстовый (15)	Да		
Отношение Продажи				
<i>ДатаПродажи</i>	Дата/время	Нет	√	
<i>КодПрод</i>	Целое	Нет		√
<i>Количество</i>	Одинарное с плавающей точкой	Да		
<i>ЦенаПродажи</i>	Денежный	Да		
Отношение Города				
<i>КодГорода</i>	Целое	Нет	√	
<i>Город</i>	Текстовый (30)	Нет		

Атрибут	Тип данных (СУБД Access)	Допустимость Null-значений	Первичный ключ	Внешний ключ
Отношение Поставки				
<i>ДатаПоставки</i>	Дата/время	Нет	V	
<i>КодПост</i>	Целое	Нет		V
<i>КодПрод</i>	Целое	Нет		V
<i>КоличествоП</i>	Одинарное с плавающей точкой	Да		
<i>ЦенаПоставки</i>	Денежный	Да		
<i>ДатаИзгот</i>	Дата/время	Да		
Отношение Заказы				
<i>ДатаЗаказа</i>	Дата/время	Нет	V	
<i>КодПост</i>	Целое	Нет		V
<i>КодПрод</i>	Целое	Нет		V
<i>КоличествоЗ</i>	Одинарное с плавающей точкой	Да		



НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ

Профессор кафедры 31
Федин Федор Олегович

После составления концептуальной (логической) схемы БД необходимо проверить её на отсутствие аномалий модификации данных.

Различают три вида аномалий:

- *аномалии обновления,*
- *аномалии удаления,*
- *аномалии добавления.*

Аномалия обновления может возникнуть в том случае, когда информация дублируется.

Другие аномалии возникают тогда, когда две и более сущности объединены в одно отношение.

Пример:

53

Рассмотрим аномалии на примере отношения (таблицы) **ПОСТАВКИ** (атрибуты, входящие в ключ, выделены подчёркиванием):

ПОСТАВКИ (Номер поставки, Название товара, Цена товара, Количество, Дата поставки, Название поставщика, Адрес поставщика)

Аномалия обновления: может возникнуть, если у какого-либо поставщика изменился адрес. Изменения должны быть внесены во все кортежи, соответствующие поставкам этого поставщика; в противном случае данные будут противоречивы.

Аномалия удаления: при удалении записей обо всех поставках определённого поставщика все данные об этом поставщике будут утеряны.

Аномалия добавления: возникнет, если с поставщиком заключен договор, но поставок от него ещё не было. Сведения о таком поставщике нельзя внести в таблицу ПОСТАВКИ, т.к. для него не определён ключ (номер поставки и название товара) и другие обязательные атрибуты.

Декомпозиция схемы отношения R – замена её совокупностью схем отношений A_i таких, что

$$R = \boxtimes A_i$$

и не требуется, чтобы отношения A_i были непересекающимися.

Декомпозиция отношения не должна приводить к потере зависимостей между атрибутами сущностей.

Для декомпозиции должна существовать операция реляционной алгебры, применение которой позволит восстановить исходной отношение.

Схема базы данных – совокупность схем отношений, адекватно моделирующих абстрактные объекты предметной области и семантические связи между этими объектами.

Функциональная зависимость. Атрибут Y некоторого отношения функционально зависит от X (атрибуты могут быть составными), если в любой момент времени каждому значению X соответствует одно значение Y .

Функциональная зависимость обозначается $X \rightarrow Y$.

Избыточная функциональная зависимость – это зависимость, заключающая в себе такую информацию, которая может быть получена на основе других зависимостей, имеющихся в базе данных.

Полная функциональная зависимость. Неключевой атрибут функционально полно зависит от составного ключа если он функционально зависит от всего ключа в целом, но не находится в функциональной зависимости от какого-либо из входящих в него атрибутов.

Транзитивная функциональная зависимость.

Пусть X , Y , Z – три атрибута некоторого отношения.

При этом $X \rightarrow Y$ и $Y \rightarrow Z$, но обратное соответствие отсутствует, т.е. $Z \not\rightarrow Y$ и $Y \not\rightarrow X$.

Тогда Z транзитивно зависит от X .

Многозначная зависимость.

Пусть X , Y , Z – три атрибута отношения R .

В отношении R существует многозначная зависимость

$$R.X \twoheadrightarrow R.Y$$

только в том случае, если множество значений Y , соответствующее паре значений X и Z , зависит только от X и не зависит от Z .

В общем случае необходимо проводить нормализацию к пятой нормальной форме (5НФ).

На практике зачастую оказывается достаточным приведение к третьей нормальной форме (3НФ).

Первая нормальная форма (1НФ): Отношение находится в 1НФ, если значения всех его атрибутов атомарны.

Атомарность – степень структурирования и детализации информации в БД.

Глубина структурирования определяется практической необходимостью при манипулировании данными.

Пример:

Пусть атрибут предназначен для хранения всего адреса (город, улица, дом, квартира).

Данный атрибут будет атомарным, если нет необходимости манипулировать отдельными городами или улицами.

В противном случае этот атрибут не является атомарным и необходимо его дальнейшее разбиение на отдельные атрибуты (город), (улица, дом, квартира).

Пример 1

R_1 – Ненормализованное отношение

КодПоставщика	КодПродукта	Продукт
P_1	1	Сахар
	2	Соль
	13	Мука
P_2	26	Рис
	58	Гречка
	130	Крупа манная
	162	Пшено
P_3	474	Молоко
	891	Кефир

R_2 – Нормализованное отношение

КодПоставщика	КодПродукта	Продукт
P_1	1	Сахар
P_1	2	Соль
P_1	13	Мука
P_2	26	Рис
P_2	58	Гречка
P_2	130	Крупа манная
P_2	162	Пшено
P_3	474	Молоко
P_3	891	Кефир

Вторая нормальная форма (2НФ): Отношение (таблица) находится во 2НФ, если оно находится в 1НФ, и каждый неключевой атрибут функционально полно зависит от всего ключа.

Пусть имеется отношение R , находящееся в 1НФ: $R(\underline{k_1}, \underline{k_2}, a_1, a_2)$.

В этом отношении:

- $\underline{k_1}, \underline{k_2}$ – составной первичный ключ;
- a_1 и a_2 – неключевые атрибуты.

Пусть также имеют место функциональные зависимости:

- $\underline{k_1}, \underline{k_2} \rightarrow a_1$ (атрибут a_1 функционально полно зависит от первичного ключа $\underline{k_1}, \underline{k_2}$);
- $\underline{k_1} \rightarrow a_2$ (атрибут a_2 зависит от части первичного ключа k_1 , т.е. имеется неполная функциональная зависимость).

Для приведения отношения R к 2НФ его необходимо декомпонировать на два отношения:

- $R_1(\underline{k_1}, a_2)$;
- $R_2(\underline{k_1}, \underline{k_2}, a_1)$.

Отношения R_1 и R_2 будут иметь связь один-ко-многим по атрибуту k_1 .

Пример:

Дано отношение

Поставки(КодПоставщика, КодПродукта, ЕдиницаИзмерения).

Поставщик может поставлять различные продукты, один и тот же продукт может поставляться разными поставщиками.

Тогда первичным ключом отношения будут атрибуты

КодПоставщика и КодПродукта.

Значит, существует функциональная зависимость:

КодПоставщика, КодПродукта → ЕдиницаИзмерения.

Какой бы поставщик не поставил продукт, единица измерения от этого не изменится (например, цельное молоко измеряется литрами независимо от поставщика, а соль – килограммами). Следовательно, существует еще одна функциональная зависимость (неключевой атрибут зависит от части первичного ключа):

КодПродукта → ЕдиницаИзмерения.

Для исключения неполной функциональной зависимости выполним деление на два отношения:

Поставки(КодПоставщика, КодПродукта)

и

Продукты(КодПродукта, ЕдиницаИзмерения).

При неполной функциональной зависимости возникают **аномалии**:

- *включения* (пока поставщиком не будет поставлен продукт, нельзя указать единицу измерения);
- *удаления* (исключение поставщика может привести к потере единицы измерения продукта);
- *обновления* (при изменении единицы измерения продукта, приходится менять данные везде, где встречается данный продукт).

Третья нормальная форма (3НФ): Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

$$A \setminus B = \{c \mid c \in A \wedge c \notin B\}$$

Пример:

Дано отношение *Группы*(Группа, *Специальность*, *Факультет*) с первичным ключом Группа.

Группа однозначно определяет специальность, а специальность однозначно определяет факультет. То есть существуют следующие функциональные зависимости:

Группа → *Специальность* (и наоборот, *Специальность* ↗ *Группа*);
Специальность → *Факультет* (*Факультет* ↗ *Специальность*).

После исключения транзитивной функциональной зависимости получим отношения:

Группы(Группа, *Специальность*)
 И
Специальности(Специальность, *Факультет*).

- 2.1. Введение в консолидацию данных
- 2.2. Общая характеристика OLTP-систем
- 2.3. Предпосылки появления систем поддержки принятия решений

Ситуации, с которыми сталкиваются аналитики:

Данные расположены в различных источниках самых разнообразных форматов и типов

Данные могут быть избыточными или, наоборот, недостаточными

Данные являются «грязными», то есть содержат факторы, мешающие их правильной обработке и анализу (пропуски, аномальные значения, дубликаты и противоречия)

Прежде чем приступать к анализу данных, необходимо выполнить ряд процедур, цель которых – «доведение» данных до приемлемого уровня качества и информативности, а также организовать их интегрированное хранение в структурах, обеспечивающих их целостность, непротиворечивость, высокую скорость и гибкость выполнения аналитических запросов.

Консолидация – комплекс методов и процедур, направленных на извлечение данных из различных источников, обеспечение необходимого уровня их информативности и качества, преобразование в единый формат, в котором они могут быть загружены в хранилище данных или аналитическую систему.

Критерии оптимальности с точки зрения консолидации данных:

➤ Обеспечение высокой скорости доступа к данным

➤ Компактность хранения

➤ Автоматическая поддержка целостности структуры данных

➤ Контроль непротиворечивости данных

Задачи консолидации данных

➤ 1. Выбор источников данных

➤ 2. Разработка стратегии консолидации

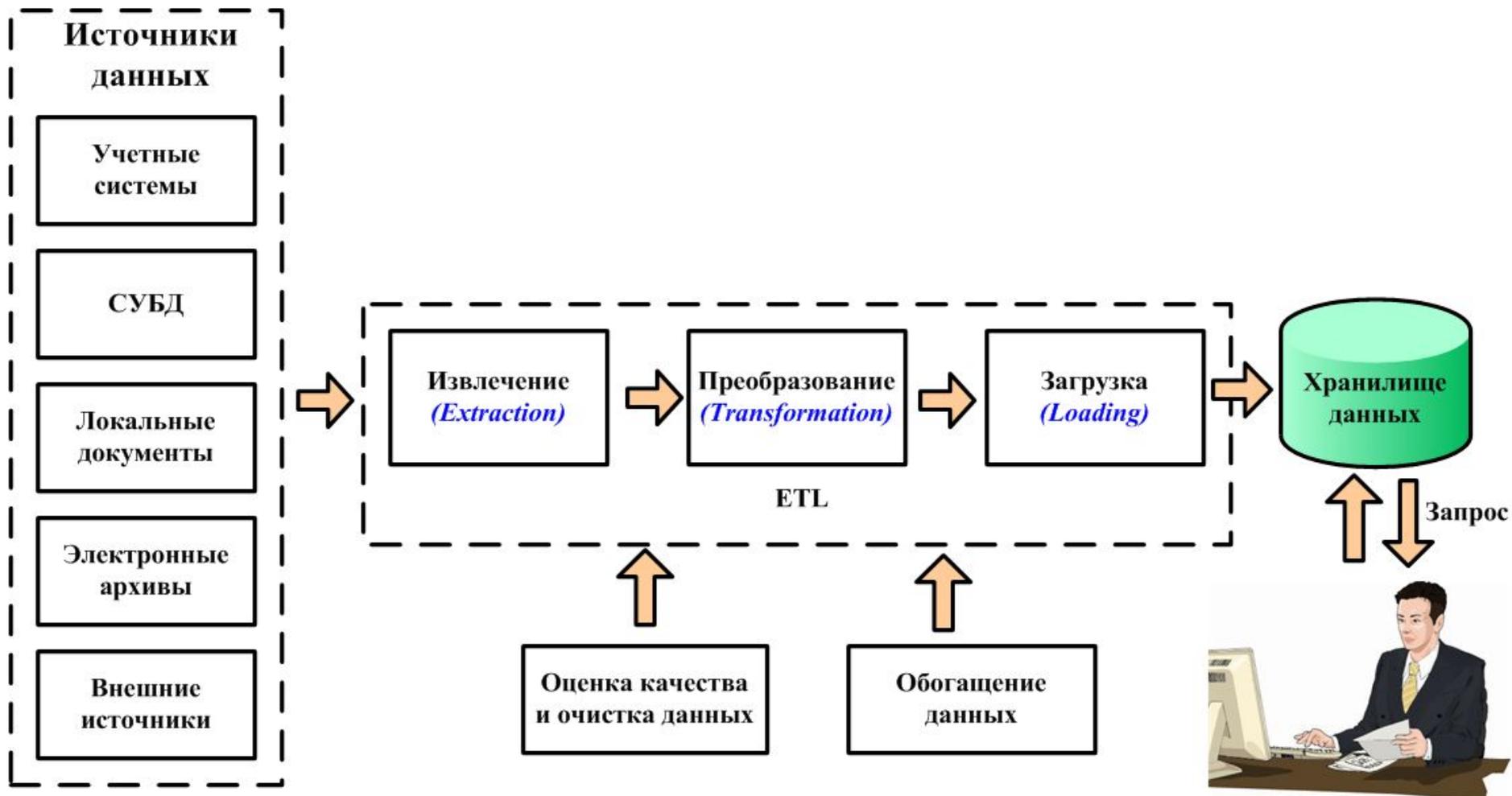
➤ 3. Оценка качества данных

➤ 4. Очистка данных

➤ 5. Обогащение данных

➤ 6. Перенос данных в хранилище данных

Обобщенная схема процесса консолидации данных



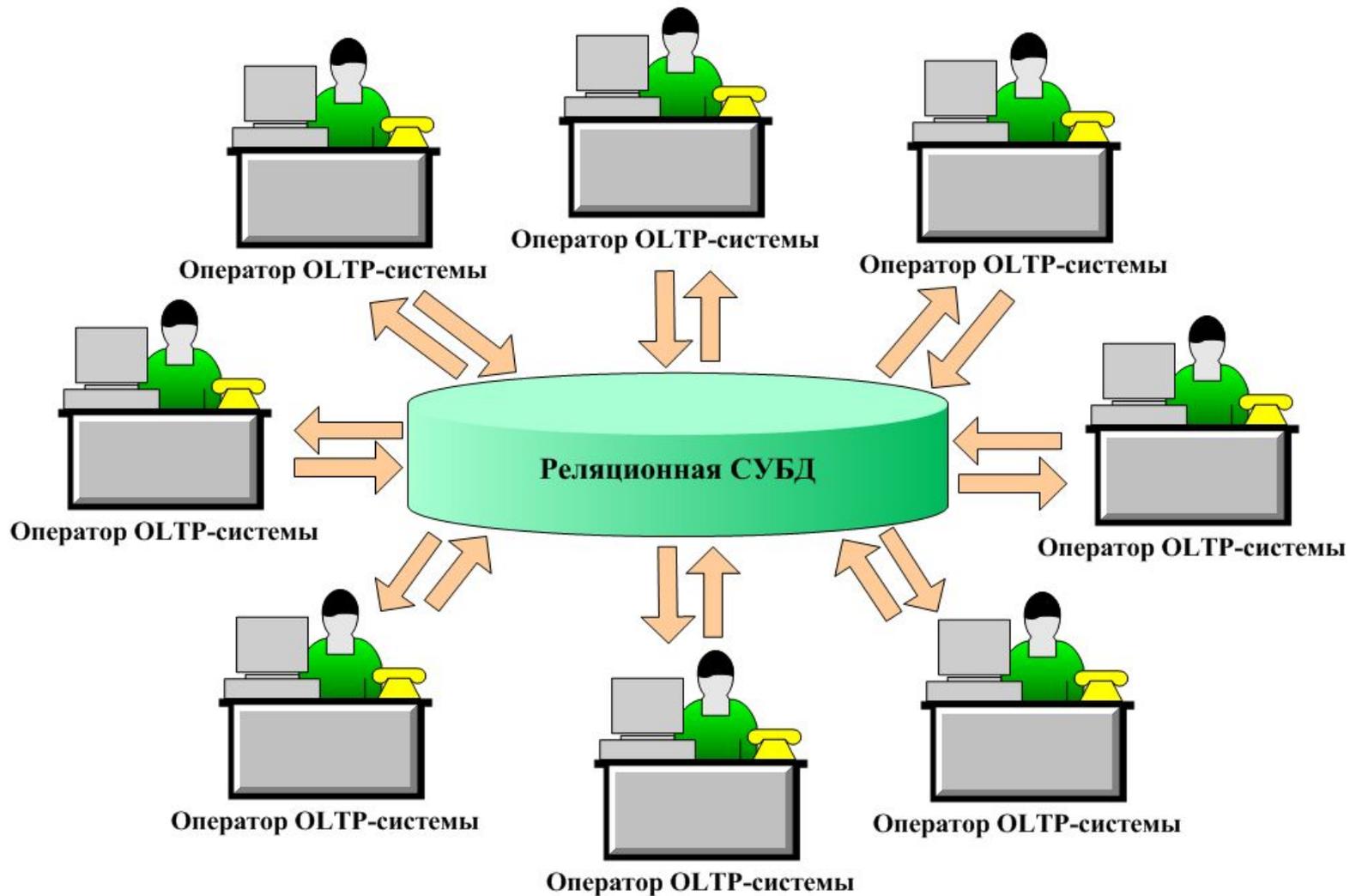
ETL (Extraction, Transformation, Loading) – процесс решающий задачу извлечения данных из разнотипных источников, их преобразования к виду, пригодному для хранения в определенной структуре, а также загрузки в соответствующую базу или хранилище данных.

OLTP (On-Line Transaction Processing) – это системы оперативной, то есть в режиме реального времени, обработки транзакций

Транзакция – некоторый набор операций над базой данных, который рассматривается как единое завершённое, с точки зрения пользователя, действие над некоторой информацией, обычно связанное с обращением к базе данных

Главное требование к OLTP-системам – быстрое обслуживание относительно простых запросов большого числа пользователей, при этом время ожидания выполнения типового запроса не должно превышать несколько секунд

Обобщенная структура системы OLTP



Характерные черты, свойственные всем OLTP-системам:

- запросы и отчеты полностью регламентированы;

- информация об обслуживании данного клиента теряет смысл, становится неактуальной и подлежит удалению по прошествии определенного времени (то есть исторические данные не поддерживаются);

- операции производятся над данными с максимальным уровнем детализации.

Постепенно появилась потребность в системах, которые могли бы выполнять не только простейшие действия над данными (подсчитывать суммы, средние, максимальные и минимальные значения), а позволяли бы проводить глубокую аналитическую обработку имеющихся данных с целью:

- поиска скрытых структур и закономерностей в массивах данных;
- вывода из них правил, которым подчиняется данная предметная область;
- стратегического и оперативного планирования;
- формирования нерегламентированных запросов;
- принятия решений и прогнозирования их последствий.

В связи с этим появился новый класс информационных систем – **системы поддержки принятия решений (СППР)**, ориентированные на аналитическую обработку данных с целью получения знаний, необходимых для разработки решений в области управления.

Отличия СППР и OLTP-систем

Свойство	OLTP-система	СППР
Цели использования данных	Быстрый поиск, простейшие алгоритмы обработки	Аналитическая обработка с целью поиска скрытых закономерностей, построения прогнозов и моделей и т. д.
Уровень обобщения (детализации) данных	Детализированные	Как детализированные, так и обобщенные (агрегированные)
Требования к качеству данных	Возможны некорректные данные (ошибки регистрации, ввода и т. д.)	Ошибки в данных не допускаются, поскольку могут привести к некорректной работе аналитических алгоритмов
Формат хранения данных	Данные могут храниться в различных форматах в зависимости от приложения, в котором они были созданы	Данные хранятся и обрабатываются в едином формате
Время хранения данных	Как правило, не более года (в пределах отчетного периода)	Годы, десятилетия
Изменение данных	Данные могут добавляться, изменяться и удаляться	Допускается только пополнение; ранее добавленные данные изменяться не должны, что позволяет обеспечить их хронологию
Периодичн. обновления	Часто, но в небольших объемах	Редко, но в больших объемах
Доступ к данным	Должен быть обеспечен доступ ко всем текущим (оперативным) данным	Должен быть обеспечен доступ к историческим (то есть накопленным за достаточно длительный период времени) данным с соблюдением их хронологии
Характер выполняемых запросов	Стандартные, настроенные заранее	Нерегламентированные, формируемые аналитиком «на лету» в зависимости от требуемого анализа
Время выполнения запроса	Несколько секунд	До нескольких минут

Хранилище данных (Data Warehouse) – разновидность систем хранения данных, ориентированная на поддержку процесса анализа данных, обеспечивающая целостность, непротиворечивость и хронологию данных, а также высокую скорость выполнения аналитических запросов.

Основная задача ХД – поддержка процесса анализа данных

Требования к ХД:

Высокая скорость получения данных из хранилища

Автоматическая поддержка внутренней непротиворечивости данных

Возможность получения и сравнения срезов данных

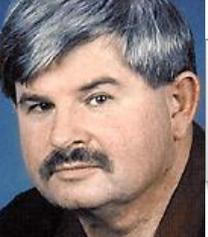
Наличие удобных средств для просмотра данных в хранилище

Обеспечение целостности и достоверности хранящихся данных

Положения, лежащие в основе концепции ХД:

Интеграция и согласование данных из различных источников, расположенных как внутри предприятия, так и во внешнем окружении

Разделение наборов данных, используемых системами выполнения транзакций и СППР



У истоков концепции ХД стоял технический директор компании *Prism Solutions* **Билл Инмон**.

Определение ХД (Б. Инмон): **предметно-ориентированный, интегрированный, неизменяемый и поддерживающий хронологию** набор данных, предназначенный для обеспечения принятия управленческих решений.

Предметная ориентированность: ХД должно разрабатываться с учетом специфики конкретной предметной области, а не аналитических приложений, с которыми его предполагается использовать.

Интегрированность: должна быть обеспечена возможность загрузки в ХД информации из источников, поддерживающих различные форматы данных и созданных в различных приложениях.

Неизменчивость: в отличие от обычных систем оперативной обработки данных, в ХД данные после загрузки не должны подвергаться каким-либо изменениям, за исключением добавления новых данных.

Поддержка хронологии: соблюдение порядка следования записей, для чего в структуру ХД вводятся ключевые атрибуты **Дата** и **Время**.

Использование концепции ХД в СППР и анализе данных способствует достижению следующих целей:

Своевременное обеспечение аналитиков и руководителей всей информацией, необходимой для выработки обоснованных и качественных управленческих решений

Автоматическая поддержка внутренней непротиворечивости данных создание единой модели представления данных в организации

Создание интегрированного источника данных, предоставляющего удобный доступ к разнородной информации и гарантирующего получение одинаковых ответов на одинаковые запросы из различных аналитических приложений

Данные в ХД хранятся:

- в **детализированном** виде;
- в **агрегированном** виде;
- в виде **метаданных**.

Данные в **детализированном виде** поступают непосредственно из источников данных и соответствуют элементарным событиям, регистрируемым OLTP-системами.

Процесс обобщения детализированных данных называется **агрегированием**, а сами обобщенные данные – **агрегированными**.

Метаданные – данные о данных.

Метаданные хранятся отдельно от данных в **репозитории метаданных**

Два уровня метаданных

Технический (статистика загрузки данных и их использования, описание модели данных и т.д.)

Бизнес-уровень (бизнес-термины и определения, которыми привык оперировать пользователь)

Бизнес-метаданные образуют **семантический слой**.

Пользователь оперирует близкими ему терминами предметной области (товар, клиент, продажи, покупки и т. д.), а семантический слой транслирует бизнес-термины в низкоуровневые запросы к данным в хранилище.

Три основных подхода к использованию ХД:

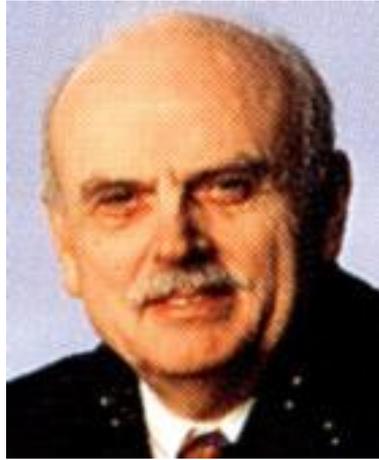
нерегламентированные запросы – возможность получать ответы на нестандартные, сформированные «по требованию» вопросы

регулярные отчеты – подготовка отчетов стандартных форм, получаемых многократно с определенной периодичностью

интеллектуальный анализ данных – поддержка процесса интеллектуального анализа больших массивов данных с целью выявления скрытых закономерностей, структур и объектов, построения моделей, прогнозов и т. д.

Чтобы приблизить ХД к условиям и специфике конкретной организации, в настоящее время разработано несколько архитектур хранилищ данных – **реляционные, многомерные, гибридные и виртуальные**.

OLAP (*On-Line Analytical Processing*) – технология оперативного извлечения нужной информации из больших массивов данных и формирования соответствующих отчетов.



Реляционную модель организации хранимых данных разработал в начале 1970-х годов англо-американский ученый **Эдгар Кодд** (Edgar Codd).

В основе реляционных хранилищ данных (ROLAP) лежит разделение данных на две группы – **измерения** и **факты**.

Измерения – это категориальные атрибуты, наименования и свойства объектов, участвующих в некотором бизнес-процессе.

Примеры измерений: наименования товаров, названия фирм-поставщиков и покупателей, ФИО людей, названия городов и т. д.

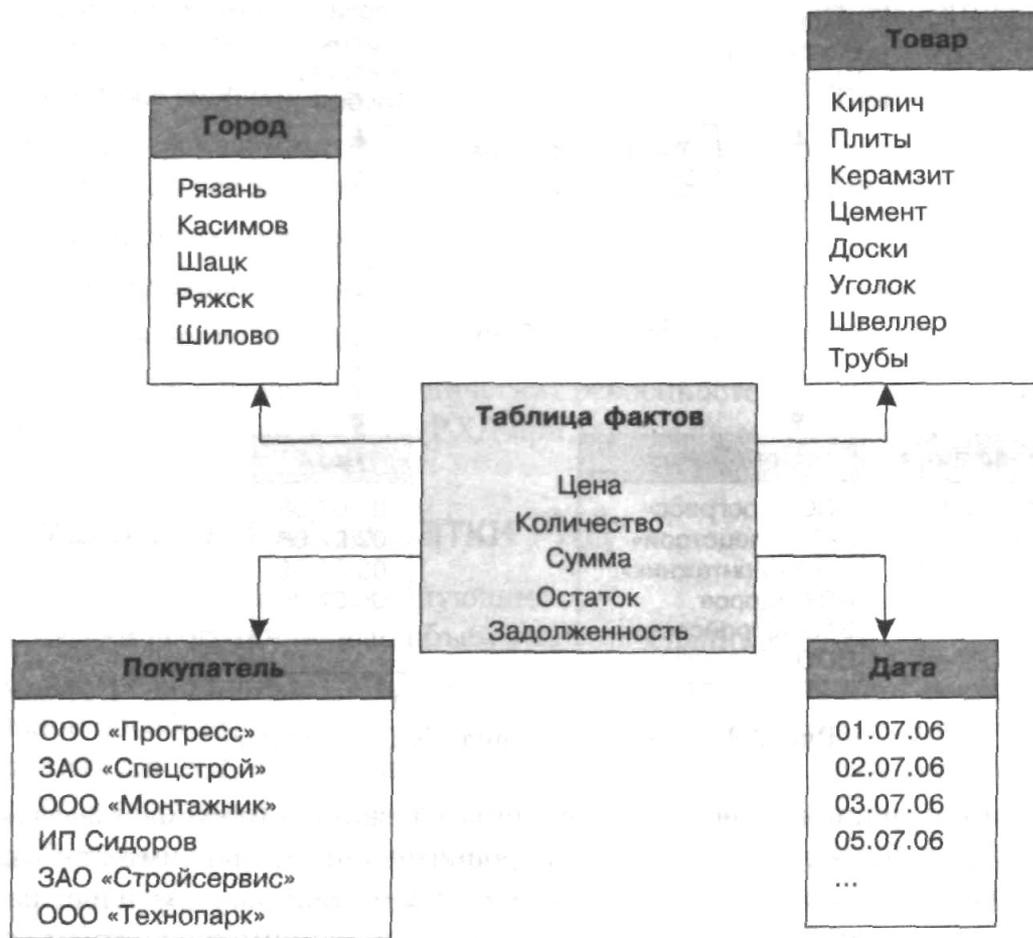
Измерения качественно описывают исследуемый бизнес-процесс.

Факты – это непрерывные по своему характеру данные (могут принимать бесконечное множество значений).

Примеры фактов: цена товара или изделия, их количество, сумма продаж или закупок, зарплата сотрудников, сумма кредита и т. д.

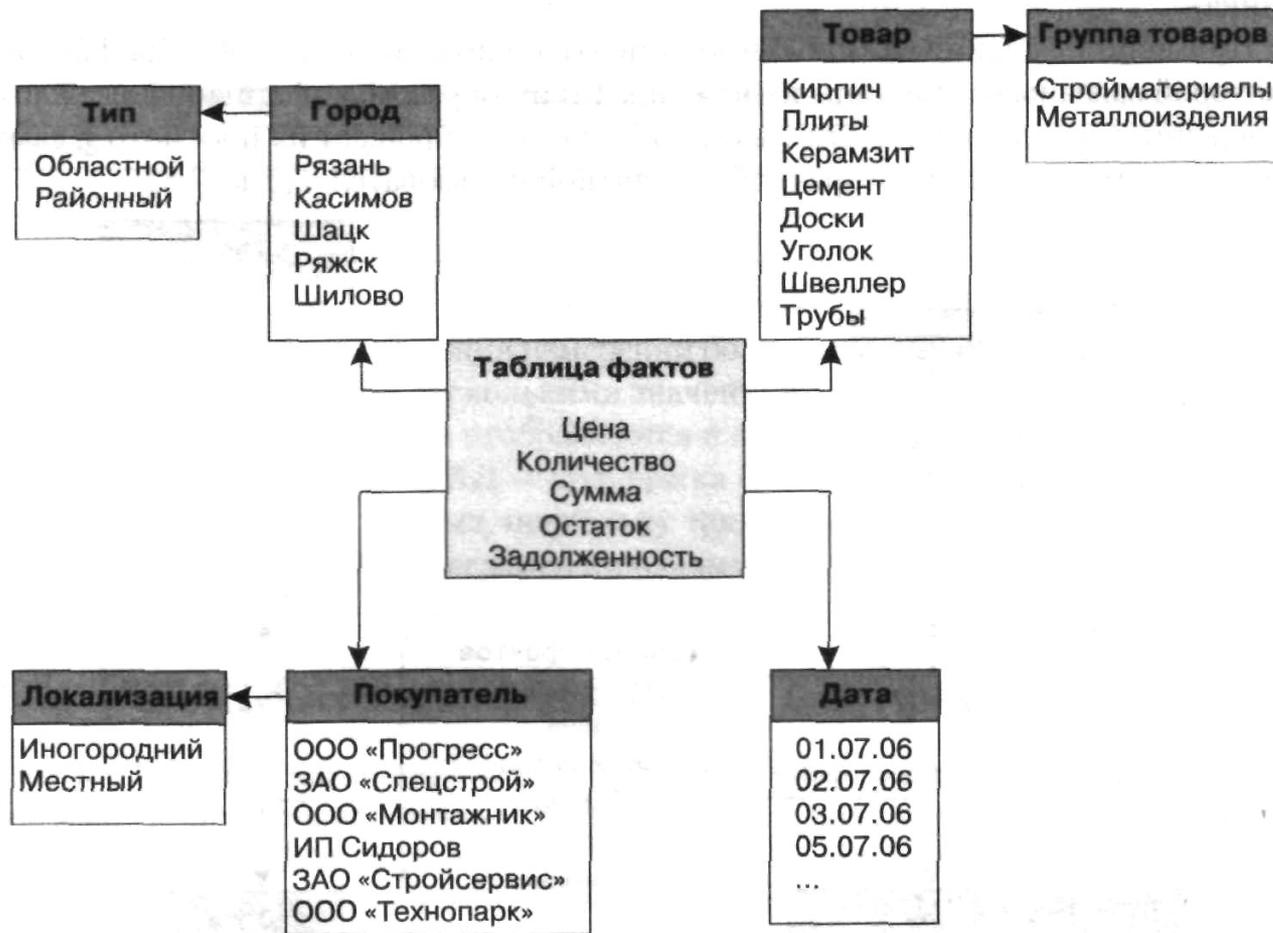
Факты количественно описывают бизнес-процесс.

Схема построения РХД «звезда»



Центральной является **таблица фактов** (Fact table), с которой связаны **таблицы измерений** (Dimension tables).

Схема построения РХД «снежинка» (модификация схемы «звезда»)



Основное функциональное отличие схемы «снежинка» от схемы «звезда» – это возможность работы с иерархическими уровнями, определяющими степень детализации данных.

Достоинства схемы «звезда»:

Простота и логическая прозрачность модели

Более простая процедура пополнения измерений, поскольку приходится работать только с одной таблицей

Недостатки схемы «звезда»:

Медленная обработка измерений, поскольку одни и те же значения измерений могут встречаться несколько раз в одной и той же таблице

Высокая вероятность возникновения несоответствий в данных (в частности, противоречий), например, из-за ошибок ввода

Достоинства схемы «снежинка»:

Она ближе к представлению данных в многомерной модели

Процедура загрузки из РХД в многомерные структуры более эффективна и проста, поскольку загрузка производится из отдельных таблиц

Намного ниже вероятность появления ошибок, несоответствия данных

Большая, по сравнению со схемой «звезда», компактность представления данных, поскольку все значения измерений упоминаются только один раз

Недостатки схемы «снежинка»

Достаточно сложная для реализации и понимания структура данных

Усложненная процедура добавления значений измерений

Преимущества РХД:

Практически неограниченный объем хранимых данных

Поскольку реляционные СУБД лежат в основе построения многих систем оперативной обработки (OLTP), которые обычно являются главными источниками данных для ХД, использование реляционной модели позволяет упростить процедуру загрузки и интеграции данных в хранилище

При добавлении новых измерений данных нет необходимости выполнять сложную физическую реорганизацию хранилища, в отличие, например, от многомерных ХД

Обеспечиваются высокий уровень защиты данных и широкие возможности разграничения прав доступа

Главный недостаток РХД

При использовании высокого уровня обобщения данных и иерархичности измерений в таких хранилищах начинают «размножаться» таблицы агрегатов. В результате скорость выполнения запросов реляционным хранилищем замедляется

Выбор реляционной модели при построении ХД целесообразен в следующих случаях:

Значителен объем хранимых данных (многомерные ХД становятся неэффективными)

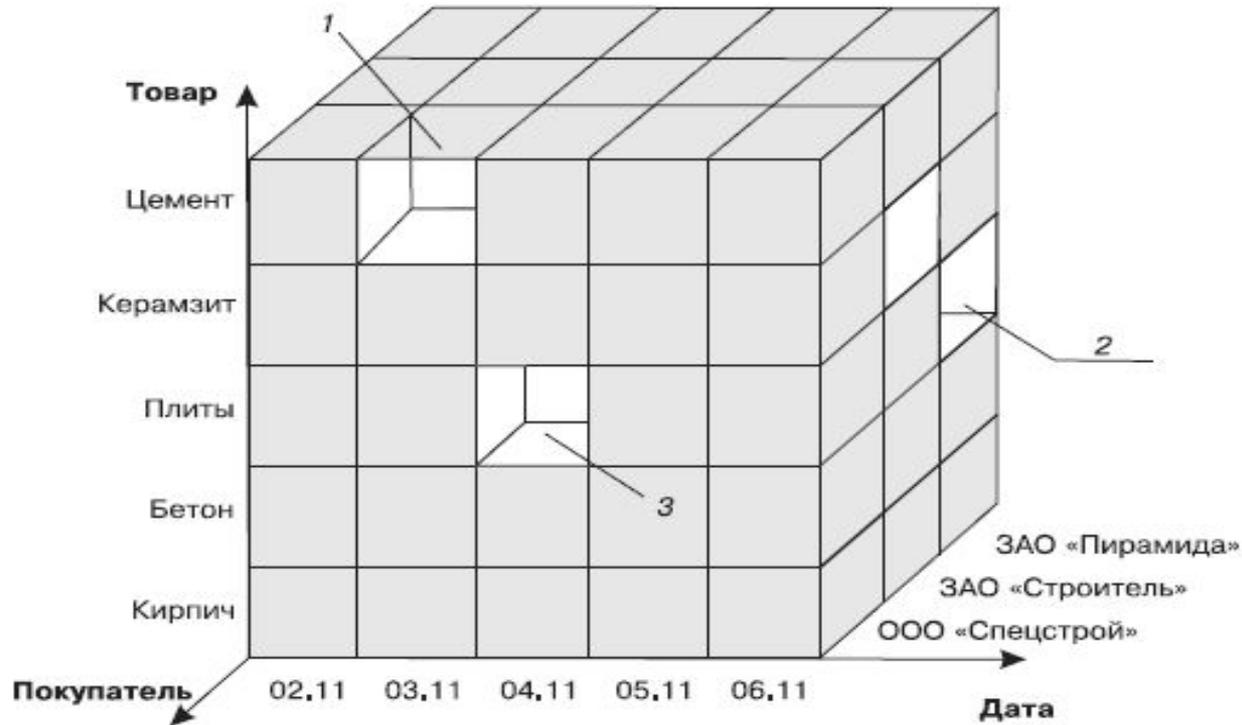
Иерархия измерений несложная (другими словами, немного агрегированных данных)

Требуется частое изменение размерности данных

Многомерная модель данных, лежащая в основе построения многомерных хранилищ данных (**MOLAP**), опирается на концепцию **многомерных кубов**, или **гиперкубов**.

Кубы представляют собой упорядоченные многомерные массивы.

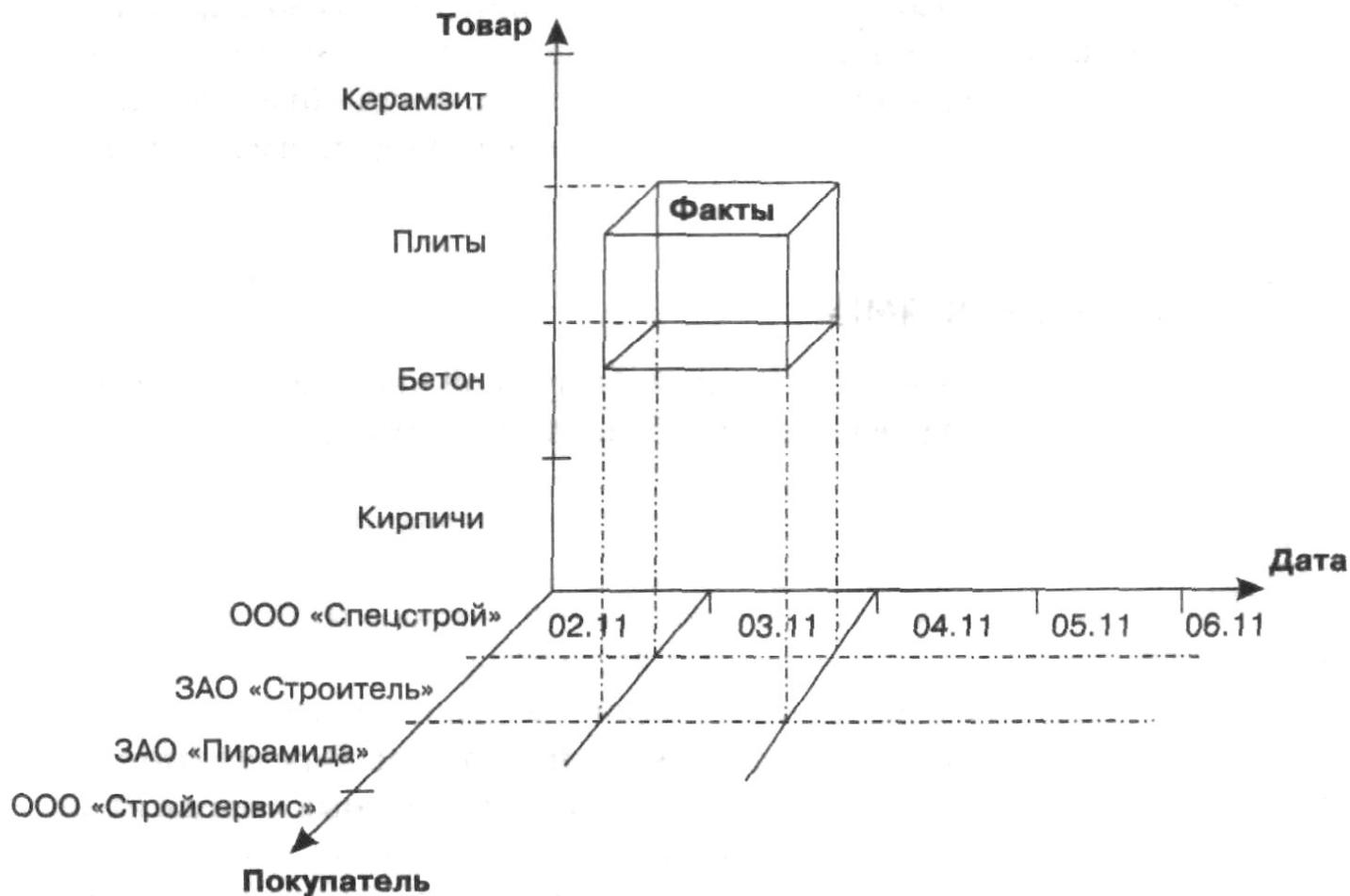
Многомерный куб можно рассматривать как систему координат, осями которой являются измерения (например, **Дата**, **Товар**, **Покупатель**). По осям будут откладываться значения измерений



В ячейке 1 будут располагаться факты, относящиеся к продаже цемента ООО «Спецстрой» 3 ноября, в ячейке 2 – к продаже плит ЗАО «Пирамида» 6 ноября, а в ячейке 3 – к продаже плит ООО «Спецстрой» 4 ноября.

Многомерный взгляд на измерения *Дата*, *Товар* и *Покупатель*

Выделенный сегмент будет содержать информацию о том, сколько плит, на какую сумму и по какой цене приобрела фирма ЗАО «Строитель» 3 ноября.



Преимущества многомерного подхода

Представление данных в виде многомерных кубов более наглядно, чем совокупность нормализованных таблиц реляционной модели, структуру которой представляет только администратор БД

Возможности построения аналитических запросов к системе, использующей МХД, более широки

В некоторых случаях использование многомерной модели позволяет значительно уменьшить продолжительность поиска в МХД, обеспечивая выполнение аналитических запросов практически в режиме реального времени

Недостатки использования многомерной модели

Для ее реализации требуется большой объем памяти

Многомерная структура труднее поддается модификации

Применение систем хранения, в основе которых лежит многомерное представление данных, целесообразно только в тех случаях, когда объем используемых данных сравнительно невелик, а сама многомерная модель имеет стабильный набор измерений.

Действия над измерениями гиперкуба:

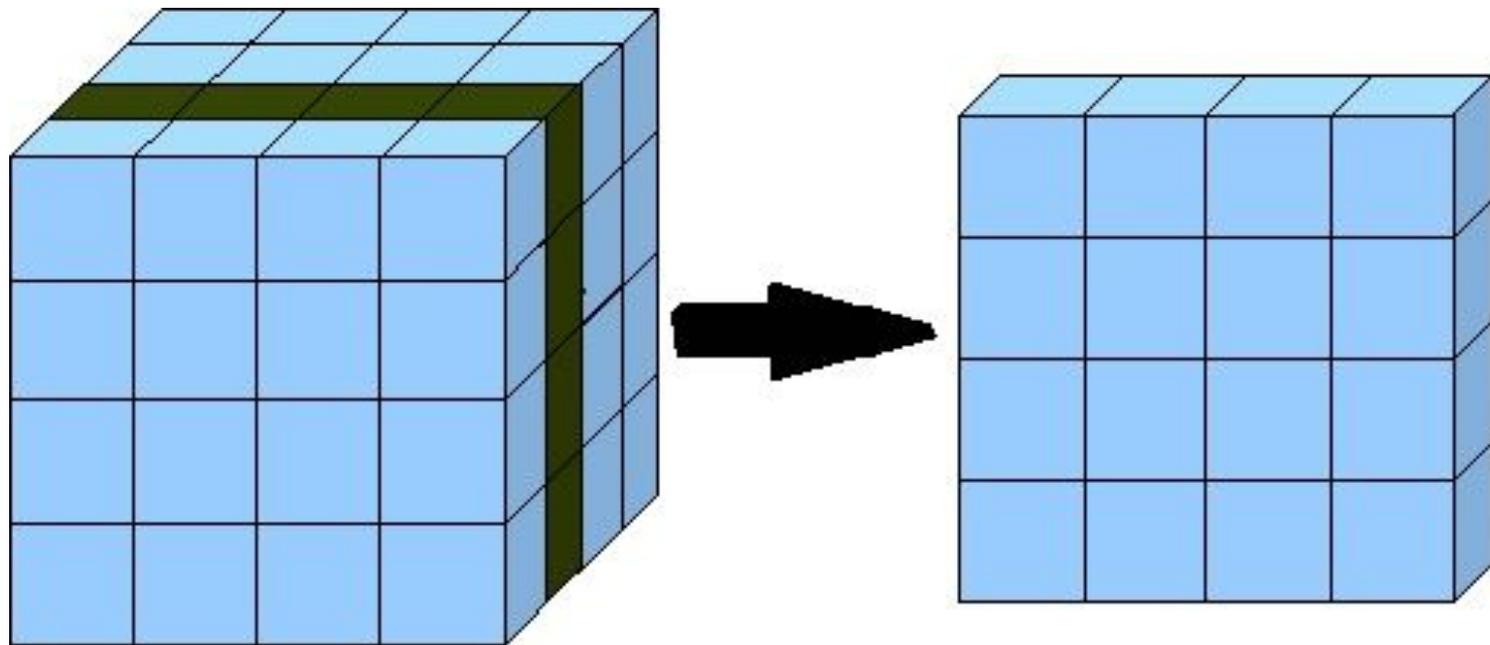
Сечение

Транспонирование (вращение)

Свертка (группировка)

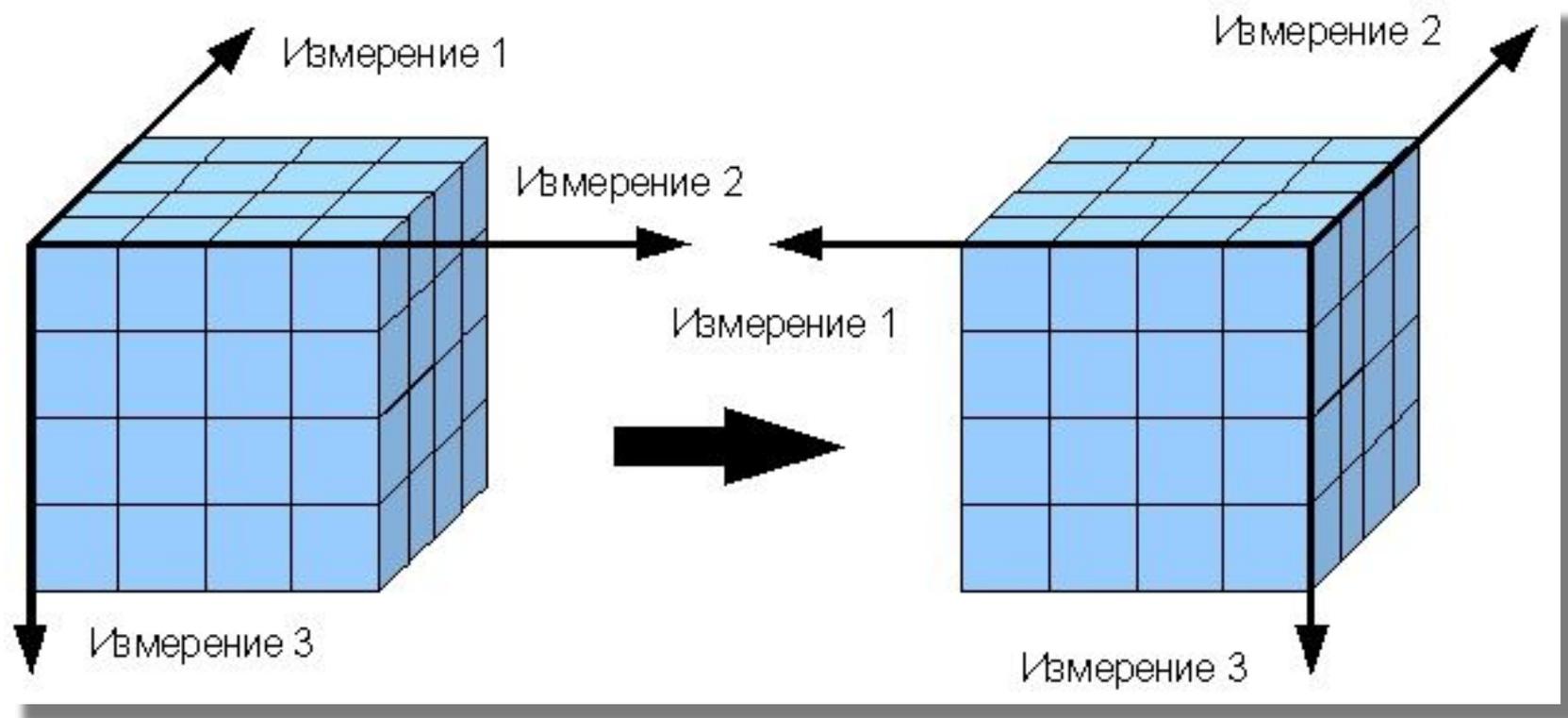
Детализация (декомпозиция)

Сечение (срез) - формируется подмножество многомерного массива данных, соответствующее единственному значению одного или нескольких элементов измерений, не входящих в это подмножество.

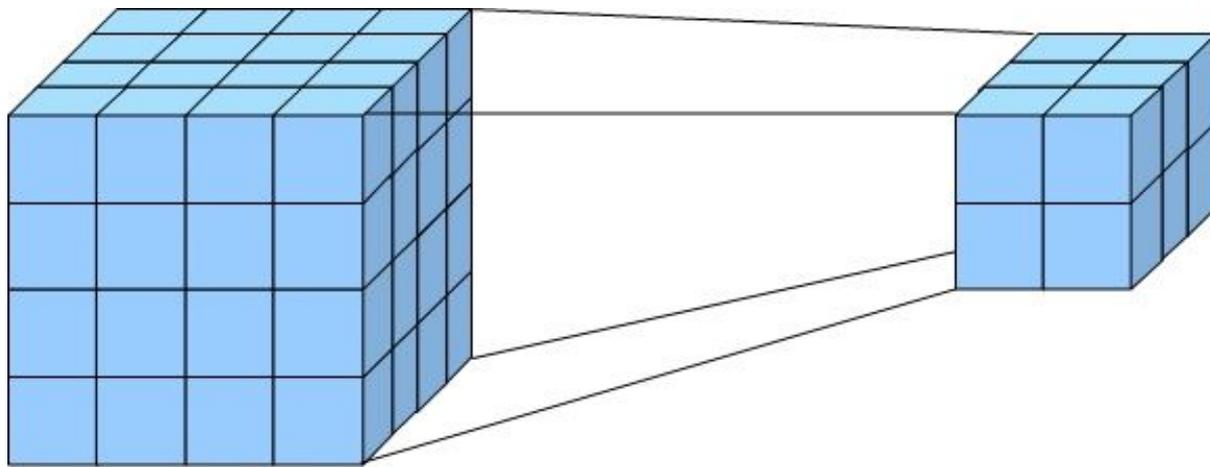


В результате сечения получается срез или несколько срезов, каждый из которых содержит информацию, связанную со значением измерения, по которому он был построен.

Вращение – изменение расположения измерений, представленных в отчете или на отображаемой странице.



Свертка – замена одного или нескольких подчиненных значений измерений теми значениями, которым они подчинены (уровень обобщения данных увеличивается)



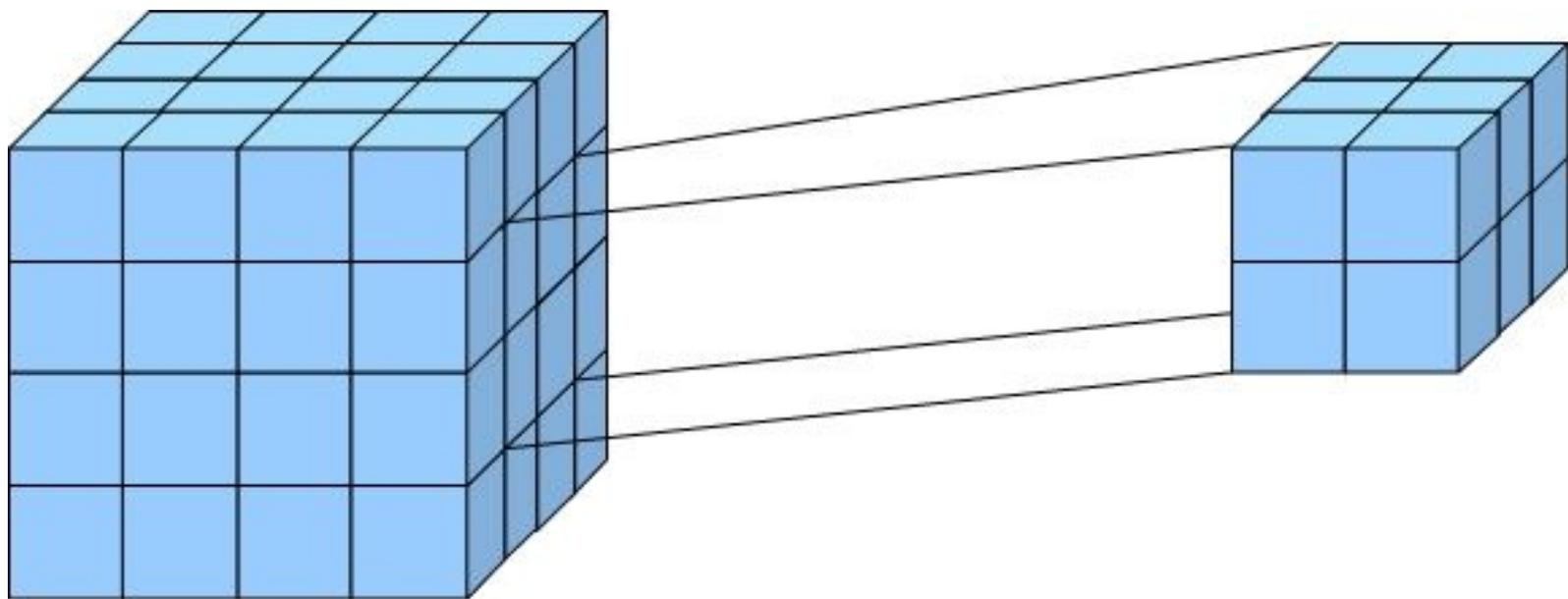
Исходная таблица

Группа	Товар	Сумма
Стройматериалы	Кирпич	22000
	Цемент	12000
	Керамзит	4500
	Доска	7400
Инструмент	Отвертка	1200
	Электропила	7600
	Дрель	2450
	Шпатель	780

Результат свертки исходной таблицы по измерению «Товар»

Группа	Сумма
Стройматериалы	45900
Инструмент	12030

Детализация - процедура обратная свертке (уменьшает уровень обобщения данных)



Гибридные хранилища данных (*Hybrid OLAP*, HОLAP) – хранилища данных, построенные на основе MOLAP и ROLAP.

Позволяют сочетать **высокую производительность**, характерную для многомерных ХД (MOLAP), и возможность хранить сколь угодно большие массивы данных, присущую **реляционным ХД (ROLAP)**.

Преимущества гибридной модели

Хранение данных в реляционной структуре делает их в большей степени системно-независимыми, что особенно важно при использовании в управлении предприятием экономической информации (показателей).

Реляционная структура формирует устойчивые и непротиворечивые опорные точки для многомерного хранилища.

Поскольку реляционное хранилище поддерживает актуальность и корректность данных, оно обеспечивает очень надежный транспортный уровень для доставки информации в многомерное хранилище.

Недостаток гибридной модели

Усложнение администрирования ХД из-за более сложного регламента его пополнения, поскольку при этом необходимо согласовывать изменения в реляционной и многомерной структурах

Витрина данных (*data mart*) – специализированное локальное тематическое хранилище, подключенное к централизованному ХД и обслуживающее отдельное подразделение организации или определенное направление ее деятельности.

Концепция витрины данных заключается в выделении профильных данных, чаще всего используемых по определенному направлению деятельности, в отдельный набор и в организации его хранения в отдельной многомерной БД, подключенной к централизованному РХД.