# Формализация информации и Big Data

«02.03.03 - Математическое обеспечение и администрирование информационных систем направленность разработка и администрирование информационных систем»

http//vikchas.ru

## Тема 1. Формализация информации

## Лекция 3 «Базы данных»

Часовских Виктор Петрович
д-р техн. наук, профессор кафедры ШИиКМ

ФГБОУ ВО «Уральский государственный экономический университет»

Екатеринбург 2022

Кристофер Дейт (Christopher J. Date) — один из крупнейших специалистов в области баз данных определяет:

1. Система **баз данных** — это, по сути, не что иное, как компьютеризированная система хранения однотипных записей. Саму же базу данных можно рассматривать как подобие электронной картотеки, т.е. хранилище или контейнер для некоторого набора файлов данных, занесенных в компьютер. Пользователям этой системы предоставляется возможность выполнять (или передавать системе запросы на выполнение) множество различных операций над такими файлами, например:»

- добавлять новые пустые файлы в базу данных;
- вставлять новые данные в существующие файлы;
- получать данные из существующих файлов;
- удалять данные из существующих файлов;
- изменять данные в существующих файлах;
- удалять существующие файлы из базы данных.

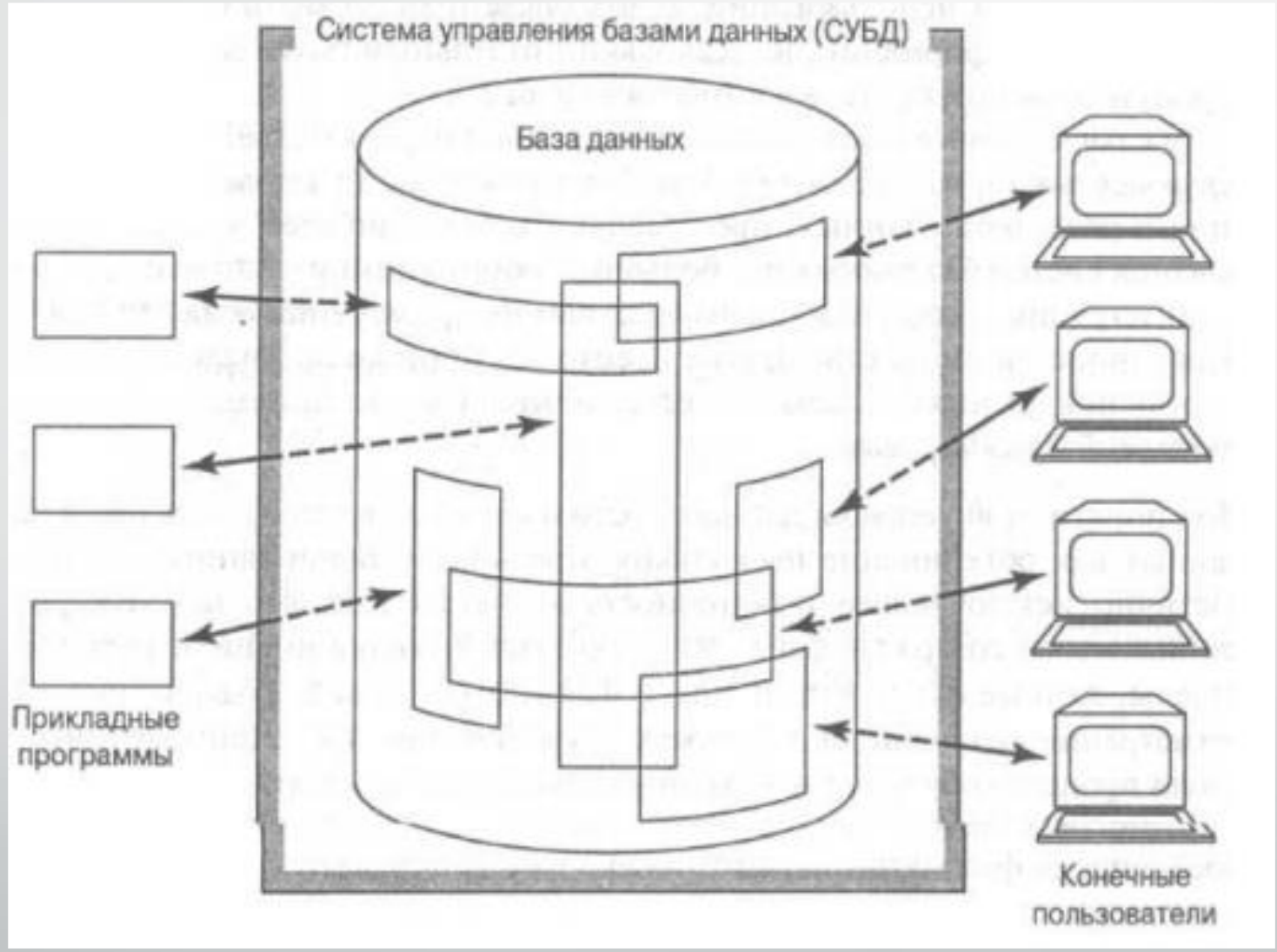
*2.Система баз данных — это компьютеризированная система хранения записей, т.е. компьютеризированная система, основное назначение которой — хранить информацию, предоставляя пользователям средства ее извлечения и модификации.*

# История баз данных

- Середина 1960-х гг. Иерархические СУБД (IMS компании IBM)
- Середина 1960-х гг. Сетевые СУБД (IDMS/R компании Computer Associates)
- 1970 г. Доктор E. F. Codd предложил реляционную модель.
- 1970-е гг. Предложены первые реляционные СУБД (Ingres в Berkeley и System R в компании IBM, язык SQL).
- 1971 СУБД Adabas, сокращение от “адаптируемая система баз данных"
- 1976 г. П. Чен предложил модель «сущность–связь» (entity–relation) – ERмодель.
- 1979 г. Появление коммерческих реляционных СУБД Oracle, Ingres, DB2
- 1987 г. Стандарт ISO языка SQL (последующие выпуски стандарта: 1989, 1992 (SQL2), 1999 (SQL:1999), 2003 (SQL:2003), 2008 (SQL:2008), 2011 (SQL:2011)

- 1990-е гг. Появление объектно-ориентированных СУБД и объектно реляционных СУБД.
- 1990-е гг. Появление хранилищ данных (data warehousing).
- Середина 1990-х гг. Интеграция web с базами данных.
- Середина 1990-х гг. – по настоящее время open source СУБД PostgreSQL. 2005 год – первый выпуск СУБД **Greenplum**, реляционная СУБД с архитектурой MPP (*массивно-параллельная архитектура*).
- В 2015 году исходный код СУБД Greenplum выпущен под свободной лицензией.
- В 2018 год – российская компания «Аренадата Софтвер», разработчик первого отечественного дистрибутива Аренадата.
- 2019 СУБД SQL 2019 с кластером больших данных SQL Server 2019, которые обеспечивают полномасштабную среду для работы с BigData, в том числе с использованием машинного обучения и возможностей искусственного интеллекта.

# Упрощенная схема СУБД



# *Архитектуры СУБД*

*Симметричная мультипроцессорная архитектура (SMP),*

Некоторое время назад основными системами для аналитической обработки больших объемов данных являлись smp-системы или системы симметричные мультипроцессорной архитектуры.

## *Архитектуры СУБД*

*Симметричная мультипроцессорная архитектура (SMP),*

Главная особенностью систем с такой архитектурой является наличие общих физических ресурсов, которые разделяются между несколькими процессорами сравнимой производительности. Память служит, в частности, для передачи сообщений между процессорами и этом все вычислите текстура называется симметричной.

Большинство популярных СУБД таких как Oracle, Sybase и PostgreSQL , реализованы именно на такой архитектуре.

**SMP-системы** обладают рядом преимуществ, таких как высокая скорость обмена данными между процессорами за счет наличия общей памяти, простота и универсальность обслуживания и относительно невысокая цена.

Однако существенным недостатком таких систем является плохая масштабируемость.

Каждый раз, когда необходимо увеличить скорость обработки данных, нам необходимо увеличивать мощность сервера за счет увеличения числа процессоров. Как правило, данная операция является дорогостоящей, а в некоторых случаях и вовсе невозможно из-за физических ограничений в конфигурации сервера.

## *Массивно-параллельной обработки данных (MPP).*

Для решения проблемы масштабируемости в системах аналитической обработки данных стали применять архитектуру **MPP**, или архитектуру массивно-параллельной обработки данных.

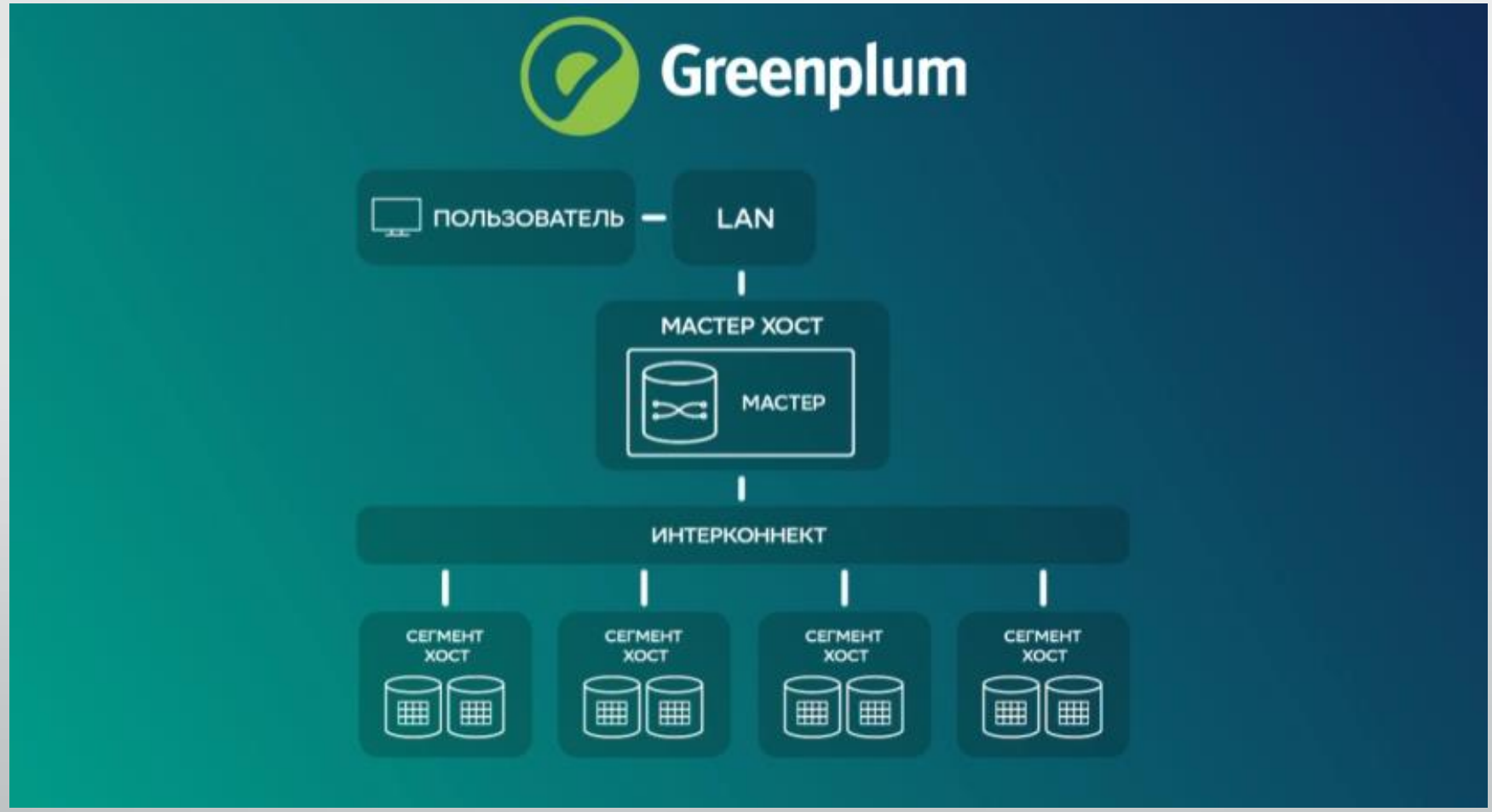
В такой архитектуре система состоит из нескольких независимых узлов, соединенных по сети.

При этом в каждом вычислительном узле процессор обладает своими собственными физическими ресурсами, такими как память и диски, которые не разделяются с другими узлами. Именно поэтому такая архитектура также называется Shared Nathing (*ничего общего*).

В **MPP** -системах вычислительная мощность и объем хранения данных увеличиваются за счет добавления в систему дополнительных вычислительных узлов.

Данный подход позволяет достигнуть линейный рост производительности и в зависимости от количества узлов в системе.

**Greenplum** - *реляционная СУБД с архитектурой MPP без разделения ресурсов.*



**Greenplum** предназначена до хранения и обработки больших объемов данных методом распределения данных и обработки запросов на нескольких серверах. Данная СУБД лучше всего подходит для построения корпоративных хранилищ данных, решение аналитических задач и задач машинного обучения и искусственного интеллекта.

**Greenplum** предоставляет широкий выбор инструментов для решения аналитических задач, например, поддержка различных сложных типов данных библиотек для задач Data Science.

В **Greenplum** можно разрабатывать хранимые процедуры не только на встроенном процедурном языке SQL, но и на многих других популярных языках программирования, которые будут компилироваться и выполняться внутри базы данных.

# Реляционная модель данных

Реляционные модели широко используются при построении баз данных. Реляционная модель данных была разработана Э. Коддом в 1969 – 1979.

На формирование современного представления о реляционной модели большое влияние оказали результаты практической реализации реляционных систем управления базами данных, включающих средства представления данных на физическом уровне, средства логического представления данных и поддержки реляционной модели данных, а также средства доступа к данным и манипулирования данными на пользовательском и административном уровнях.

В процессе эволюции были введены ограничения на структурирование и тип хранимых данных, разработаны способы реализации полной реляционной модели и создан инструментарий для манипулирования данными, не зависящими от их физического представления

# Понятие модели данных

Модель данных можно определить как средство описания логического представления физических данных.

Модель данных задает некоторый базовый набор понятий и свойств, которым должны удовлетворять все базы данных этой модели.

Модель данных состоит из трех основных компонент.

1. Структура данных.
2. Допустимые операции, выполняемые на структуре данных и составляющие основу языка данных рассматриваемой модели.
3. Ограничения для контроля целостности.

# Реляционная структура данных

рнепаотроироиь

Единственной структурой данных реляционной модели является нормализованное n-арное отношение.

Термин “отношение” используется здесь в общепринятом математическом смысле, как подмножество декартова произведения.

Пусть $D_1, D_2, \dots, D_n$ - произвольные (не обязательно различные) конечные множества.

Декартовым произведением этих множеств $D_1 \text{x} D_2 \text{x} \dots D_n$ называется множество элементов вида $\text{d} = < d_1, d_2, \dots, d_n >$, где $d_i \in D_i$, $\text{i} = 1, 2, \dots, \text{n}$.

Элементы декартова произведения называются кортежами.

Число компонент кортежа n называется степенью или арностью кортежа

# Декартово произведение

Рассмотрим пример. Пусть первое множество $D_1$ состоит из двух элементов, $D_1=\{s_1, s_2\}$, второе множество $D_2$ состоит из трех элементов $D_2=\{p_1, p_2, p_3\}$. Тогда декартово произведение этих множеств есть множество $D_1xD_2=\{<s_1p_1>, <s_1p_2>, <s_1p_3>, <s_2p_1>, <s_2p_2>, <s_2p_3>\}$, состоящее из 6 = 2*3 элементов.

Данный пример можно проиллюстрировать рисунком:

| $D_1$ |
|---|
| $s_1$ |
| $s_2$ |
| |

X

| $D_2$ |
|---|
| $p_1$ |
| $p_2$ |
| $p_3$ |

→

| $D_1$ | $D_2$ |
|---|---|
| $s_1$ | $p_1$ |
| $s_1$ | $p_2$ |
| $s_1$ | $p_3$ |
| $s_2$ | $p_1$ |
| $s_2$ | $p_2$ |
| $s_2$ | $p_3$ |

Декартово произведение позволяет получить все возможные комбинации элементов исходных множеств.

Отношением R, определенным на множествах $D_1, D_2, \dots, D_n$ называется подмножество декартова произведения $D_1 \text{x} D_2 \text{x} \dots D_n$

Множества $D_1, D_2, \dots, D_n$ называются доменами отношения R.

Степень или арность кортежей (в рассматриваемом случае n) определяет степень или арность отношения. Отношения арности 1 - унарные, арности 2 - бинарные, арности 3 - тернарные, арности k - k-арные.

Число кортежей в отношении R называется кардинальным числом или мощностью отношения.

Кардинальное число отношения может изменяется, в отличие от его степени.

Порядок кортежей в отношении несуществен.

Напротив, домены должны быть упорядочены внутри отношения (отношение есть множество кортежей, где j-й элемент каждого кортежа взят из их j-го домена).

**Отношения** удобно представлять в форме **таблиц**, где каждая строка есть кортеж, а каждый столбец - атрибут, определенный на некотором домене.

Данный неформальный подход к понятию отношения дает более привычную для разработчиков и пользователей форму представления, где реляционная база данных представляет собой конечный набор таблиц

Наименьшей единицей данных в реляционной модели является отдельное значение данных.

Такие значения рассматриваются как атомарные, т.е. более не детализируемые в рамках данной модели.

При описании реляционной модели базы данных в качестве **доменов** будем рассматривать множества таких значений одного и того же типа.

Таким образом, домены представляют собой пулы значений, из которых берутся фактические значения, появляющиеся в атрибутах (столбцах) таблицы. В этом смысле домен можно рассматривать, как синоним термина “**область определения**” для столбца таблицы.

Если атрибут определен на некотором домене, то значения этого атрибута должны удовлетворять условиям, наложенным на данный домен

# Реляционная алгебра

Реляционная алгебра (от английского слова relation - отношение) базируется на классической теории множеств и в ее основе лежит совокупность операций над отношениями.

Реляционная алгебра содержит две группы операций.

Первая группа операций. Поскольку отношения являются множествами, то к ним применимы следующие обычные теоретико-множественные операции:

- объединение отношений;
- пересечение отношений;
- разность (вычитание) отношений;
- декартово произведение отношений

Вторая группа операций содержит следующие специальные реляционные операции:

- селекция (ограничение) отношения;
- проекция отношений;
- соединение отношений;
- деление отношений.

Заметим, что данный набор операций несколько расширен относительно варианта первоначально предложенного Коддом.

Отметим также, что реляционная алгебра является замкнутой относительно понятия отношения, т.е. исходными операндами и результатами операций являются отношения

# Теоретико-множественные операции

Для всех теоретико-множественных операций, кроме декартова произведения, отношения-операнды должны быть совместимы по объединению, т.е. должны быть одной и той же степени (арности).

Более того, в заголовках обоих отношений должен содержаться один и тот же набор атрибутов, и одноименные атрибуты должны быть определены на одних и тех же доменах.

**Объединение отношений R = $S_1 \cup S_2$**

Результатом выполнения операции объединения двух отношений $S_1$ и $S_2$ является отношение R, включающее все кортежи, входящие хотя бы в одно из отношений-операндов.

Пример.

$S_1$

| к | л | м |
|---|---|---|
| у | ф | х |
| *а* | *б* | *в* |

$S_2$

| ж | з | и |
|---|---|---|
| *а* | *б* | *в* |
| о | п | р |
| э | ю | я |

$R = S_1 \cup S_2$

| к | л | м |
|---|---|---|
| у | ф | х |
| *а* | *б* | *в* |
| ж | з | и |
| о | п | р |
| э | ю | я |

*Пример операции объединения отношений*

***2. Пересечение отношений*** $\mathbf{R = S_1 \cap S_2}$

Результатом выполнения операции пересечения двух отношений $S_1$ и $S_2$ является отношение R, включающее все кортежи, входящие в оба отношения-операнды.

Пример.

$S_1$

| к | л | м |
|---|---|---|
| у | ф | х |
| *а* | *б* | *в* |

$S_2$

| ж | з | и |
|---|---|---|
| *а* | *б* | *в* |
| о | п | р |
| э | ю | я |

$R = S_1 \cap S_2$

| *а* | *б* | *в* |
|---|---|---|

*Пример операции пересечения отношений*

***Разность (вычитание) отношений*** $\mathbf{R = S_1 - S_2}$

Результатом выполнения операции разности (вычитания) двух отношений $S_1$ и $S_2$ является отношение R, включающее все кортежи, входящие в отношение $S_1$-первый операнд, и не входящие в отношение $S_2$-второй_операнд.

Пример.

$S_1$

| к | л | м |
|---|---|---|
| у | ф | х |
| *а* | *б* | *в* |

$S_2$

| ж | з | и |
|---|---|---|
| *а* | *б* | *в* |
| о | п | р |
| э | ю | я |

$R = S_1 - S_2$

| к | л | м |
|---|---|---|
| у | ф | х |

## *Декартово произведение отношений* $R = S_1 \times S_2$

Отличительной особенностью данной операции от предыдущих является то, что операнды могут представлять из себя отношения, построенные по разным схемам (разной степени). Если отношение $S_1$ имеет арность $k_1$, а отношение $S_2$ имеет арность $k_2$, то декартовым произведением отношений $S_1$ и $S_2$ является множество кортежей арности $(k_1 + k_2)$. Причем первые $k_1$ элементов образуют кортеж их отношений $S_1$, а последние $k_2$ элементов - из отношения $S_2$.

Пример.

$S_1$

| | | |
|---|---|---|
| к | л | м |
| у | ф | х |
| а | б | в |

$S_2$

| | |
|---|---|
| о | п |
| э | ю |

$R = S_1 \times S_2$

| | | | | |
|---|---|---|---|---|
| к | л | м | о | п |
| к | л | м | э | ю |
| у | ф | х | о | п |
| у | ф | х | э | ю |
| а | б | в | о | п |
| а | б | в | э | ю |

*Пример операции декартова произведения отношений*

# Специальные реляционные операции

## *Селекция (ограничение) отношения*

Селекция (ограничение) R отношения S по формуле F есть подмножество всех кортежей для которых истинна формула F:

$$R = \sigma_F (S),$$

где F - формула, образованная:

1. операндами, являющимися номерами столбцов;
2. логическими операторами $\wedge$ (И), $\vee$ (ИЛИ), $\neg$ (НЕ);
3. арифметическими бинарными отношениями сравнения: $<, =, >, \leq, \neq, \geq$

Операция селекции имеет одно отношение на входе и одно отношение на выходе. Результирующее отношение состоит из подмножества кортежей исходного отношения.

Пример.

| к | л | м |
|---|---|---|
| у | ф | х |
| а | б | в |

S

| к | л | м |
|---|---|---|
| а | б | в |

$R_1 = \sigma_{1=к \vee 1=а}(S)$

| у | ф | х |
|---|---|---|

$R_2 = \sigma_{3=х}(S)$

*Примеры операции селекции отношения*

## *Проекция отношений*

Операция взятия проекции позволяет получить такое подмножество исходного отношения, которое получается выбором заданных атрибутов с последующим исключением, избыточных кортежей-дубликатов.

Операция требует наличия двух операндов - проецируемого отношения S и списка имен атрибутов, входящих в заголовок отношения. Таким образом, проекция позволяет получать вертикальное подмножество заданного отношения, путем выборки заданных столбцов и компоновки их в указанном порядке:

$$R = \pi_{i1, i2, ..., in} (S)$$

где $i_1, i_2, ..., i_n$ - номера столбцов отношения S.

Пример.

| к | л | м |
|---|---|---|
| у | ф | х |
| а | б | в |

S

| к | л |
|---|---|
| у | ф |
| а | б |

$R_1 = \pi_{1,2} (S)$

| м | к |
|---|---|
| х | у |
| в | а |

$R_2 = \pi_{3,1} (S)$

## *Соединение отношений*

Пусть θ - арифметический оператор сравнения; n - арность отношения $S_1$; m – арность отношения $S_2$, i, j - номера (имена) столбцов в отношениях $S_1$ и $S_2$ соответственно.

*Соединением* R отношений $S_1$ и $S_2$ называется множество всех кортежей r таких, что r является конкатенацией (сцеплением) какого-либо кортежа $s_1$ из $S_1$ и какого-либо кортежа $s_2$ из $S_2$ с условием, что выражение i θ j истинно:

$$R = S_1 \underset{i\theta j}{\times} S_2 = \sigma_{i\theta\,(n+j)}(S_1 \times S_2)$$

Как видно из определения, операция соединения имеет сходство с декартовым произведением. Однако при соединении в результирующее отношение включаются только кортежи, удовлетворяющие определенному соотношению между атрибутами соединения соответствующих отношений.

Пример.

$S_1$

| А | Б | В |
|---|---|---|
| к | л | м |
| у | ф | х |
| а | б | в |

$S_2$

| Г | Д |
|---|---|
| л | а |
| о | п |

$$R = S_1 \underset{Б=Г}{\times} S_2$$

| | | | | |
|---|---|---|---|---|
| к | л | м | л | а |

*Пример операции соединения отношений*

## *Деление отношений*

В простейшей форме операция деления делит отношение степени два (делимое) на отношение степени один (делитель) и продуцирует отношение степени один (частное). Пусть делимое $S_1$ имеет атрибуты А и Б, а делитель $S_2$ - атрибут А. Результатом деления $S_1$ на $S_2$ является отношение R с единственным атрибутом Б таким, что каждое *б* этого атрибута S.*б* появляется как значение $S_1$.Б и пара значений (*а*, *б*) входит в $S_1$ для всех значений *а*, входящих в $S_2$.

Пример.

$S_1$

| | | | |
|---|---|---|---|
| п | л | о | в |
| э | ф | я | м |
| в | к | о | в |
| п | л | я | м |
| в | к | я | м |
| п | л | к | в |

$S_2$

| | |
|---|---|
| я | м |
| о | в |

$R = S_1 / S_2$

| | |
|---|---|
| п | л |
| в | к |

*Пример операции деления отношений*