
# Введение в реляционные базы данных

Лекция 1: в SQL

Артем Толканев

September 25, 2024

# Вопросы и ответы

- Кто лектор? - Артем, tg:@ArtemDrowMag, email: artem.tolkanev@phystech.edu
- github курса? - тык
- рбд сложно? - и да, и нет
- как записаться на курс? - см. чат

# На этой лекции поговорим о...

... таком явлении, как базы данных

... реляционной модели

... реляционной алгебре

# База данных

**Данные** (data) -

1. элементы, составляющие ввод или вывод;
2. последовательность одного или более символов;
3. истинное высказывание об описываемом мире.

**База данных** (database) -

1. множество истинных высказываний

WHAT A DATABASE *REALLY* IS:
PREDICATES AND PROPOSITIONS

*An open letter to Open University database students*

Hugh Darwen

ABSTRACT

Despite the name, a database is best thought of as a repository not just for data, but rather for facts—that is, for *true propositions*. The article that follows explains this remark in lay terms and begins to explore some of its many implications.

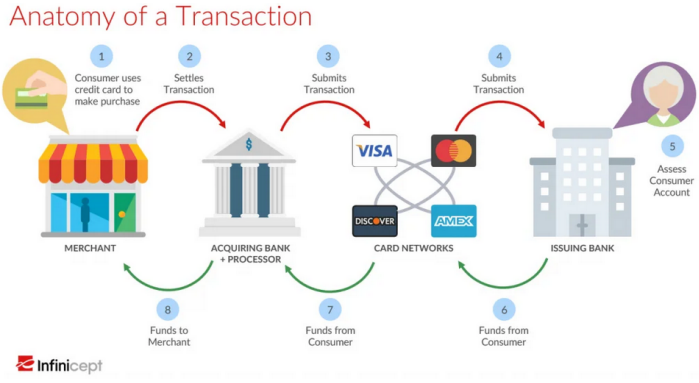
2. некоторых набор пермаментных данных;
3. организованая коллекция связанных данных, которая моделирует какую-то часть реального мира.

Базы данных - ключевой компонент большинства приложений.

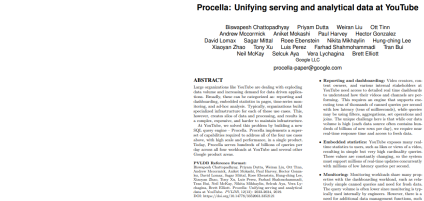


girafe ai

# Примеры из жизни



# Примеры из жизни

**Procella: Unifying serving and analytical data at YouTube**

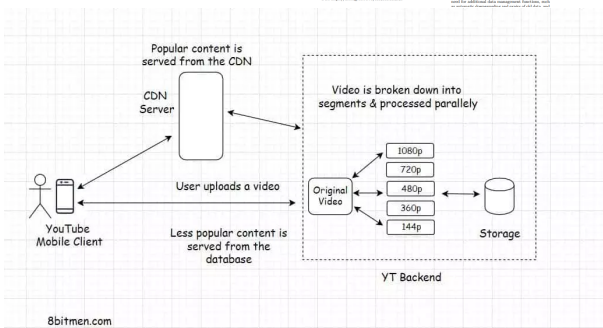
Biswapesh Chattopadhyay Priyam Dutta Weiran Liu Ott Tinn
Andrew Mccormick Aniket Mokashi Paul Harvey Hector Gonzalez
David Lomax Sagar Mittal Roee Ebenstein Nikita Mikhaylin Hung-ching Lee
Xiaoyan Zhao Tony Xu Luis Perez Farhad Shahmohammadi Tran Bui
Neil McKay Selcuk Aya Vera Lychagina Brett Elliott
Google LLC
procella-paper@google.com

**ABSTRACT**

Large organizations like YouTube are dealing with exploding data volume and increasing demand for data-driven applications. Rec...

**PVLDB Reference Format:**

Biswapesh Chattopadhyay, Priyam Dutta, Weiran Liu, Ott Tinn, Andrew Mccormick, Aniket Mokashi, Paul Harvey, Hector Gonzalez, David Lomax, Sagar Mittal, Roee Ebenstein, Hung-ching Lee, Xiaoyan Zhao, Tony Xu, Luis Perez, Farhad Shahmohammadi, Tran Bui, Neil McKay, Selcuk Aya, Vera Lychagina, Brett Elliott. Procella: Unifying serving and analytical data at YouTube. PVLDB, 12(12): 2022-2034, 2019.
DOI: https://doi.org/10.14778/3352063.3352135



# Примеры из жизни


## Core

Stack Overflow uses a **WISC** stack ~~via BizSpark~~ (we graduated!):

- **Operating System** Microsoft Windows Server 2019 x64
- **Web Server** IIS 10
- **Database** SQL Server 2019 running Microsoft Windows Server 2016 x64
- **Language** C#

## Software Development Tools

- **IDE** Visual Studio 2019
- **Framework** Microsoft .NET 6.0
- **Web Framework** ASP.NET Core 6.0 with MiniProfiler
- **View Engine** Razor
- **Browser Framework** jQuery 1.12.4
- **Data Access Layer** Entity Framework Core 2.2 and Dapper
- **Cache / Additional Data** redis 4.0.7 via StackExchange.Redis, with serialization via protobuf-net
- **Source Control** Git using a GitHub Enterprise instance hosted by GitHub (previously self-hosted GitHub, Mercurial from 2010–2014, Subversion from 2008–2010)
- **Compare Tool** Beyond Compare 4



girafe ai

# Пример учебный

Придумаем модель данных для отслеживания публикакий статей нужных вам авторов.

Нужная нам информация, как минимум:

- Информация об авторах
- Информация о статьях, которые эти авторы написали

# CSV-файлик

Будем держать нашу базу данных в CSV формате и управлять данными в отдельном приложении.

- Для авторов отдельный файлик, для статей отдельный файлик;
- Приложение должно обрабатывать файл каждый раз, когда нужно прочитать или обновить запись;

Автор(Имя, Альма-матер, Дата рождения)

| | | |
|---|---|---|
| Jürgen Schmidhuber | Technical University of Munich | 17 January 1963 |
| Арнольд, Владимир Игоревич | МГУ | «12 июня 1937» |

Статья(Автор,Название, Год)

| | | |
|---|---|---|
| Jürgen Schmidhuber | Deep learning in neural networks: An overview | 2014 |
| Арнольд, Владимир Игоревич | On additive semigroups starting parts | 2008 |
| Арнольд, Владимир Игоревич | Smooth functions statistics | 2007 |

# CSV-файлик

Будем держать нашу базу данных в CSV формате и управлять данными в отдельном приложении.

- Для авторов отдельный файлик, для статей отдельный файлик;
- Приложение должно обрабатывать файл каждый раз, когда нужно прочитать или обновить запись;

Автор(Имя, Альма-матер, Дата рождения)

| | | |
|---|---|---|
| Jürgen Schmidhuber | Technical University of Munich | 17 January 1963 |
| Арнольд, Владимир Игоревич | МГУ | «12 июня 1937» |

Статья(Автор,Название, Год)

| | | |
|---|---|---|
| Jürgen Schmidhuber | Deep learning in neural networks: An overview | 2014 |
| Арнольд, Владимир Игоревич | On additive semigroups starting parts | 2008 |
| Арнольд, Владимир Игоревич | Smooth functions statistics | 2007 |

Пример: найти год, в котором Шмидхубер публиковал работу ”Smooth functions statistics”

# CSV-файлик

Что делать если мы ошиблись в имени автора?
Как мы можем быть уверены, что это один и тот же автор?
Как работать в случае, когда у статьи несколько соавторов?
Что случится, если мы удалим автора, когда мы уже дополнили статьями другой CSV?

# CSV-файлик

Что делать, если мы хотим сделать еще одно приложение, которое использует ту же самую базу данных?
Что делать, если несколько человек пытается писать в один и тот же CSV в одно и тоже время?
Что случится, если во время работы нашей программы выключат свет?

# Системы управления базами данных

**Система управления базой данных** (СУБД) представляет собой программное обеспечение, которое управляет всем доступом к базе данных

1. Пользователь выдает запрос на доступ к данным, применяя определенный подъязык данных (обычно это язык SQL)
2. СУБД перехватывает этот запрос и анализирует его
3. СУБД просматривает внешнюю схему (ее объектную версию) для этого пользователя, соответствующее отображение ”внешний—концептуальный”, концептуальную схему, отображение ”концептуальный—внутренний” и определения структур хранения
4. СУБД выполняет необходимые операции в хранимой базе данных.

# Системы управления базами данных

**Система управления базой данных** (СУБД) представляет собой программное обеспечение, которое управляет всем доступом к базе данных

Цель СУБД поддерживать работу с базой данных в соответствии с определенной **моделью данных**
БД и СУБД это разные вещи.

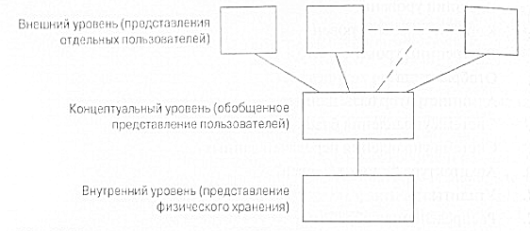
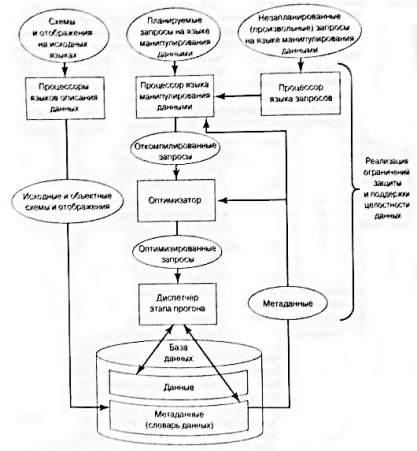



Рис. 2.1. Три уровня архитектуры ANSI/SPARC

# Системы управления базами данных

**Система управления базой данных** (СУБД) представляет собой программное обеспечение, которое управляет всем доступом к базе данных



# Системы управления базами данных

Зачем нужны СУБД 11 января 2023 года Федеральное управление гражданской авиации (FAA) остановило все полеты в США из-за сбоя системы NOTAM (можно тыкнуть).
NOTAM - система для извещейний пилотов воздушных судов о потенциальных угрозах.
Судя по всему данные хранились в плоских таблцах и вообще система не изменялась с 1980 годов (можно тыкнуть).

# Модели данных

- Реляционная <- Большая часть СУБД
- Ключ-значение
- Графовая
- Документная
- Векторная
- Другие...

# Модели данных

- Реляционная
- Ключ-значение <- кеширование
- Графовая
- Документная
- Векторная
- Другие...

# Модели данных

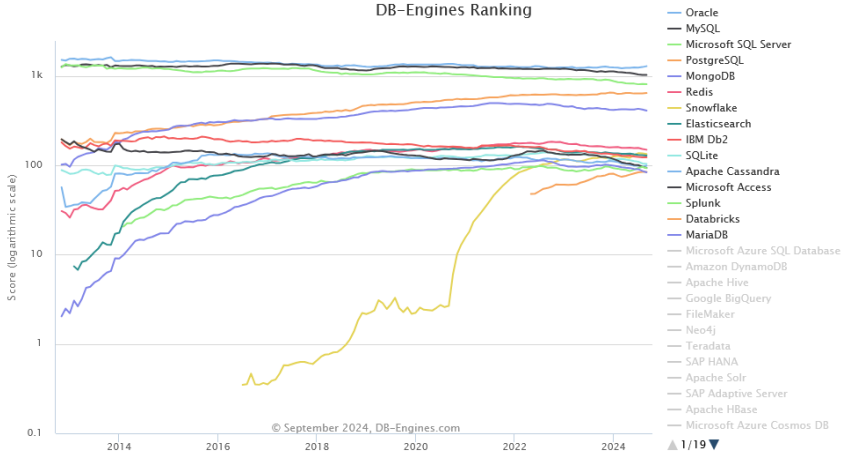
- Реляционная
- Ключ-значение
- Графовая <- NoSQL
- Документная <- NoSQL
- Векторная
- Другие...

# Модели данных

- Реляционная
- Ключ-значение
- Графовая
- Документная
- Векторная <- Научные/ML
- Другие...

# Модели данных

Картинка из The Part of PostgreSQL I Hate the Most., by Bohan Zhang



# Модели данных

- Реляционная <- Тема этого курса
- Ключ-значение
- Графовая
- Документная
- Векторная
- Другие...

# Реляционная модель

**Структура:** Данные в базе воспринимаются пользователем, как таблицы

| | book_id | name | theme_id | publishing_house_id | year_of_publishing | language_id | reading_hall_only |
|---|---|---|---|---|---|---|---|
| 1 | 200 | Война и Мир 1,2 том | 1 | 11 | 1998 | 1 | 0x00 |
| 2 | 201 | Война и Мир 3,4 том | 1 | 11 | 1998 | 1 | 0x00 |
| 3 | 202 | Othello | 2 | 12 | 2005 | 2 | 0x01 |
| 4 | 203 | Курс аналитической геометрии | 5 | 13 | 2005 | 1 | 0x00 |
| 5 | 204 | Обломов | 1 | 14 | 2005 | 1 | 0x00 |
| 6 | 205 | Капитанская Дочка | 1 | 15 | 2005 | 1 | 0x00 |
| 7 | 206 | Общая физика | 5 | 13 | 2005 | 1 | 0x00 |
| 8 | 207 | Дубровский | 1 | 15 | 2005 | 1 | 0x00 |
| 9 | 208 | Анна Каренина | 1 | 11 | 2005 | 1 | 0x01 |
| 10 | 209 | 1984 | 2 | 12 | 2005 | 2 | 0x00 |
| 11 | 210 | Мартин Иден | 2 | 16 | 2005 | 1 | 0x00 |
| 12 | 211 | Сердца Трех | 2 | 16 | 2007 | 1 | 0x00 |
| 13 | 212 | Белый Клык | 2 | 16 | 2007 | 1 | 0x00 |
| 14 | 213 | Три Сестры | 1 | 14 | 2007 | 1 | 0x00 |
| 15 | 214 | Русские сказки | 4 | 14 | 2007 | 1 | 0x00 |

# Реляционная модель

**Целостность:** таблицы отвечают определенным условиям целостности

Наличие первичного ключа

| | book_id | name | theme_id | publishing_house_id | year_of_publishing | language_id | reading_hall_only |
|---|---|---|---|---|---|---|---|
| 1 | 203 | Курс аналитической геометрии | 5 | 13 | 2005 | 1 | 0x00 |
| 2 | 215 | Курс аналитической геометрии | 5 | 13 | 2006 | 1 | 0x00 |

Наличие внешнего ключа

| | language_id | language |
|---|---|---|
| 1 | 0 | русский |
| 2 | 1 | английский |
| 3 | 2 | французский |
| 4 | 3 | немецкий |

# Реляционная модель

**Обработка:** В распоряжении пользователя имеются операторы манипулирования таблицами, которые генерируют новые таблицы на основании уже имеющихся и среди которых есть, по крайней мере, операторы сокращения, проекции и объединения

```
SELECT COUNT([b].[name]) AS 'Количество книг на английском языке'
    FROM
        [library].[dbo].[book] AS b
        INNER JOIN
        [library].[dbo].[language] AS l
        ON [b].[language_id] = [l].[language_id]
    WHERE
        [l].[language] LIKE 'английский'
```

# Реляционная модель

**Обработка:** В распоряжении пользователя имеются операторы манипулирования таблицами, которые генерируют новые таблицы на основании уже имеющихся и среди которых есть, по крайней мере, операторы сокращения, проекции и объединения

```
SELECT COUNT([b].[name]) AS 'Количество книг на английском языке'
    FROM
        [library].[dbo].[book] AS b
        INNER JOIN
        [library].[dbo].[language] AS l
        ON [b].[language_id] = [l].[language_id]
    WHERE
        [l].[language] LIKE 'английский'
```

| | Количество книг на английском языке |
|---|---|
| 1 | 14 |

# Реляционная модель

**SQL** - один (почти) язык, который можно использовать для разных СУБД.

Ключевые вещи:

- → Содержит базу данных в простых структурах данных (отношениях)
- → Физическое хранилище отеделено
- → Доступ к дате осуществляется путем высокоуровнего языка, наиболее эфективный способ извелечения данных перекладывается на СУБД

# Реляционная модель

**Тип** - именованное множество значений данных (где - значение представляет собой “отдельно взятую константу”; переменная - некоторая позиция, в которой размещается конкретное проявление определенного значения) ,наряду со связанным множеством операторов, которые могут применяться к значениям и переменным рассматриваемого типа.

# Реляционная модель

**Тип** - именованное множество значений данных ,наряду со связанным множеством операторов, которые могут применяться к значениям и переменным рассматриваемого типа. Свойства типов:

- Каждый конкретный тип может быть определен системой или пользователем
- В состав определения типа входим спецификация множества допустимых значений
- Физическое представление типа скрыто от пользователя, но каждый тип имеет минимум одно возможное представление
- Наличие специальных операторов: присваивания, селектора, проверка на равенство

# Реляционная модель

**Тип** - именованное множество значений данных ,наряду со связанным множеством операторов, которые могут применяться к значениям и переменным рассматриваемого типа. Некоторые типы являются подтипами других типов. Полиморфизм операторов.
Базовые типы SQL:

- Точные числа (integer, numeric, smallinteger, decimal)
- Приблизительные числа (real, float)
- Строковое типы (char, varchar, character large object)
- Двоичные строки (bit, varbinary, binary large object)
- Временные (date, time, timestamp, timestamp with zone, interval)
- Логический (true, false, unknown)
- Массивы (array)

# Кортежи

Если дана коллекция типов $Ti(i = 1, i = 2, ..., n)$, которые не обязательно все должны быть разными, то значением кортежа, определенным с помощью этих типов (обозначим за *t*), является множество упорядоченных троек в форме $\langle A_i, T_i, v_i \rangle$, где $A_i$ - имя аттрибута, $T_i$ - имя типа и $v_i$ - значение типа $T_i$.

$$TUPLE\ \{A_1 T_1, A_2, T_2, ..., A_n, T_n\}$$

# Кортежи

Если дана коллекция типов $Ti(i = 1, i = 2, ..., n)$, которые не обязательно все должны быть разными, то значением кортежа, определенным с помощью этих типов (обозначим за $t$), является множество упорядоченных троек в форме $\langle A_i, T_i, v_i \rangle$, где $A_i$ - имя аттрибута, $T_i$ - имя типа и $v_i$ - значение типа $T_i$.

$$TUPLE\,\{A_1T_1, A_2, T_2, ..., A_n, T_n\}$$

| | | | | | | |
|---|---|---|---|---|---|---|
| 200 | Война и Мир 1.2 том | 1 | 11 | 1998 | 1 | 0x00 |

# Кортежи

**Кортежи**

Кроме того кортеж t должен соответствовать требованиям:

- Значение $n$ имеет степень или арность $t$
- Упорядоченная тройка $\langle Ai, Ti, vi \rangle$ является компонентой $t$
- Упорядоченная пара $\langle Ai, Ti \rangle$ представляет собой атрибут $t$ и однозначно определяется именем атрибута $A_i$ (имена атрибутов совпадают, только если $i = j$). Значение $v_i$ - значение атрибута, соответствующего имени атрибута $A_i$ кортежа $t$. Тип $T_i$ - это соответствующий тип атрибута.
- Полное множество атрибутов составляет заголовок $t$.
- Тип кортежа $t$ определен заголовком $t$, а сам заголовок и этот тип кортежа имеют такие же атрибуты и такуе же степень, как $t$.

# Отношения

**Значение отношения** (обозначим за *r*) - состоит из заголовка и тела, где:

- Заголовок отношения *r* - представляют собой заголовок кортежа
- Отношение *r* имеет такие же атрибуты и такую же степень, как и заголовок.
- Тело отношения *r* представляет собой множество кортежей, имеющих один и тот же заголовок.

**Кардинальность** отношения *r* определяется как количество элементов этого множества.

# Отношения

**Значение отношения** (обозначим за *r*) - состоит из заголовка и тела, где:

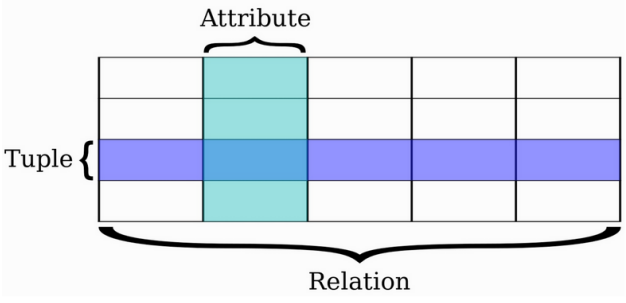
- Заголовок отношения *r* - представляют собой заголовок кортежа
- Отношение *r* имеет такие же атрибуты и такую же степень, как и заголовок.
- Тело отношения *r* представляет собой множество кортежей, имеющих один и тот же заголовок.

| Relation | | | | |
|---|---|---|---|---|
| Имя первого аттрибута $\overbrace{A_1}$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ |
| $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
| $v_2$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ |

# Отношения

**Значение отношения** (обозначим за *r*) - состоит из заголовка и тела, где:

- Заголовок отношения *r* - представляют собой заголовок кортежа
- Отношение *r* имеет такие же атрибуты и такую же степень, как и заголовок.
- Тело отношения *r* представляет собой множество кортежей, имеющих один и тот же заголовок.



# Отношения

Тип отношения *r* определяется заголовком *r* и имеет такие же атрибуты и степень, как и сам заголовок *RELATION* $\langle A_1 T_1, A_2 T_2, ..., A_n T_n \rangle$:

# Отношения

Тип отношения *r* определяется заголовком *r* и имеет такие же атрибуты и степень, как и сам заголовок *RELATION* $\langle A_1 T_1, A_2 T_2, \ldots, A_n T_n \rangle$:

| language_id# : integer# | language: varchar(20) |
|---|---|
| 0 | русский |
| 1 | английский |
| 2 | французский |
| 3 | немецкий |

Имеет тип *RELATION* $\langle language\_id : integer, language : varchar(20) \rangle$

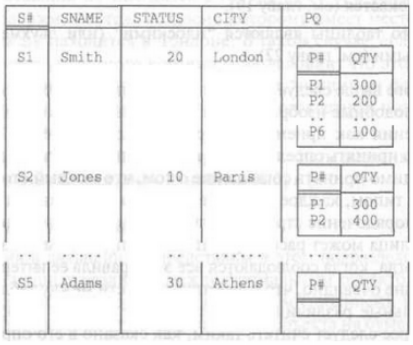
# Отношения

**Отношение** - абстрактный объект
**Таблица** - конкретное изображение такого абстрактного объекта. Тело таблицы в контексте SQL является мультимножеством строк.

1. Каждый кортеж содержит точно одно значение (соответствующего типа) для каждого атрибута.
2. Атрибуты не характеризуются каким-либо упорядочением (например, слева на право).
3. Кортежи не характеризуются каким-либо упорядочением (например, сверху вниз).
4. В отношении отсутствуют дубликаты кортежей

# Отношения

**Отношение** - абстрактный объект
**Таблица** - конкретное изображение такого абстрактного объекта. Тело таблицы в контексте SQL является мультимножеством строк. Каждый кортеж содержит точно одно значение (соответствующего типа) для каждого атрибута.



| S# | SNAME | STATUS | CITY | PQ |
|---|---|---|---|---|
| S1 | Smith | 20 | London | P# QTY<br>P1 300<br>P2 200<br>.. ...<br>P6 100 |
| S2 | Jones | 10 | Paris | P# QTY<br>P1 300<br>P2 400 |
| .. | ..... | .. | ..... | ..... |
| S5 | Adams | 30 | Athens | P# QTY |

# Отношения

**Отношение** - абстрактный объект
**Таблица** - конкретное изображение такого абстрактного объекта. Тело таблицы в контексте SQL является мультимножеством строк. Атрибуты не характеризуются каким-либо упорядочением (например, слева на право).

| language_id# : integer# | language: varchar(20) |
|---|---|
| 0 | русский |
| 1 | английский |
| 2 | французский |
| 3 | немецкий |

=

| language: varchar(20) | language_id# : integer# |
|---|---|
| русский | 0 |
| английский | 1 |
| французский | 2 |
| немецкий | 3 |

# Отношения

**Отношение** - абстрактный объект
**Таблица** - конкретное изображение такого абстрактного объекта. Тело таблицы в контексте SQL является мультимножеством строк. Кортежи не характеризуются каким-либо упорядочением (например, сверху вниз)

| language_id# : integer# | language: varchar(20) |
|---|---|
| 0 | русский |
| 1 | английский |
| 2 | французский |
| 3 | немецкий |

=

| language_id# : integer# | language: varchar(20) |
|---|---|
| 1 | английский |
| 3 | немецкий |
| 0 | русский |
| 2 | французский |

# Отношения

**Отношение** - абстрактный объект
**Таблица** - конкретное изображение такого абстрактного объекта. Тело таблицы в контексте SQL является мультимножеством строк. В отношении отсутствуют дубликаты кортежей.

| language_id# : integer# | language: varchar(20) |
|---|---|
| 0 | русский |
| 1 | английский |
| 2 | французский |
| 3 | немецкий |
| 3 | немецкий |

- Формально не отношение

# Работа с данными

СУБД предоставляет специальное API для осуществления хранения и манипулирования данными.
В основном мы будем пользоваться процедруным языком, использующим в своей основе реляционную алгебру.

# Реляционная алгебра

- набор операций. которые принимают отношения в качестве операндов и возвращают отношения в качестве результата.

# Реляционная алгебра

- набор операций. которые принимают отношения в качестве операндов и возвращают отношения в качестве результата.
Алгебру можно можно поделить на две группы:

- Традиционные операция со множествами - объединение, пересечение, разность и декартово произведение

# Реляционная алгебра

- набор операций. которые принимают отношения в качестве операндов и возвращают отношения в качестве результата.
Алгебру можно можно поделить на две группы:

- Традиционные операция со множествами - объединение, пересечение, разность и декартово произведение
- Специальные реляционные операции - сокращение, проекция, соединение, деление.

# Реляционная алгебра

- набор операций. которые принимают отношения в качестве операндов и возвращают отношения в качестве результата.
Алгебру можно можно поделить на две группы:

- Традиционные операция со множествами - объединение, пересечение, разность и декартово произведение
- Специальные реляционные операции - сокращение, проекция, соединение, деление.

Операции применимы ко всем отношениям. Данные операции предназначены только для чтения, в том смысле, что они читают, но не обновляют свои операнды.



girafe ai