

ЗАНЯТИЕ Основы баз данных



Александр Джумурат, онлайн кинотеатр «ivi.ru»



adzhumurat@ivi.ru



adzhumurat

Джумурат Александр

Разработчик рекомендательной системы, онлайн-кинотеатр ivi.ru

Цель занятия:

Получить общее понятие о БД: схемы данных, примеры БД. Понятие SQL.

В КОНЦЕ ЗАНЯТИЯ СМОЖЕТЕ:

- Различать различные СУБД по целям использования, выбирать БД исходя из задачи и проектировать архитектуру
- Развернуть Docker-контейнер с БД Postgres
- Научатся создавать в Postgres пользователя БД, схему таблиц а так же сами таблицы с учётом нормализации

О ЧЁМ ПОГОВОРИМ И ЧТО СДЕЛАЕМ

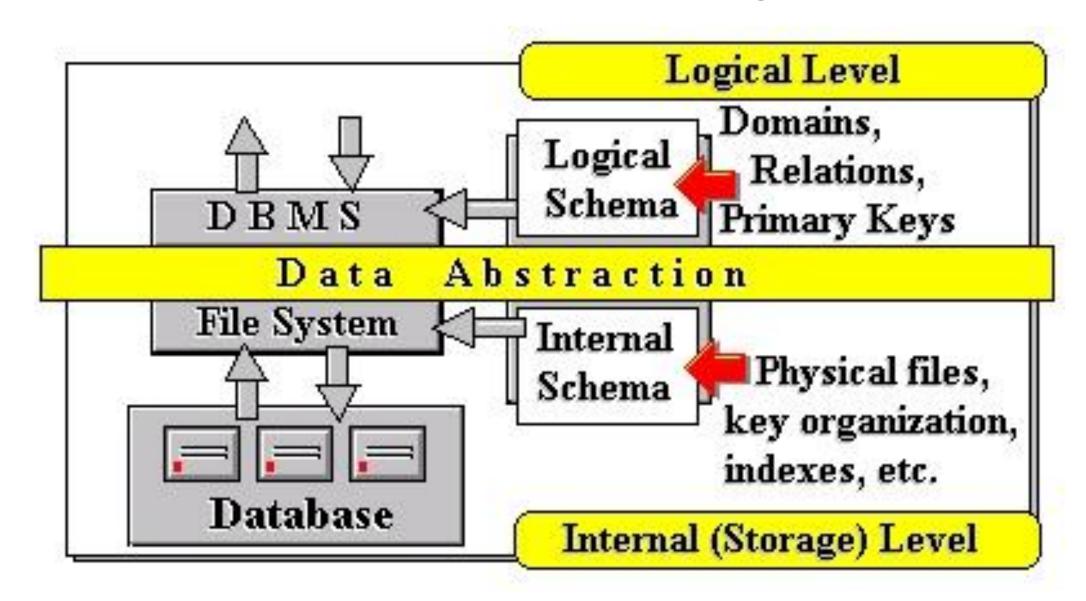
- 1. Базы данных: зачем они нужны
- 2. Выбор БД в зависимости от цели.
- 3. Работа с БД из командной строки.

О чём не будем говорить?

- движки таблиц (InnoDB, MyISAM ...)
- настройка кеша под конкретное железо
- план выполнения запросов
- правильные индексы
- компромисс между нормализацией, дубликацией данных и скоростью

• ...

Часть 1 Базы данных: основные определения **База данных -** совокупность информации, к которой можно легко получить доступ и модифицировать



Логический слой содержит бизнес-логику доменной области (например, онлайн-кинтотеатр)

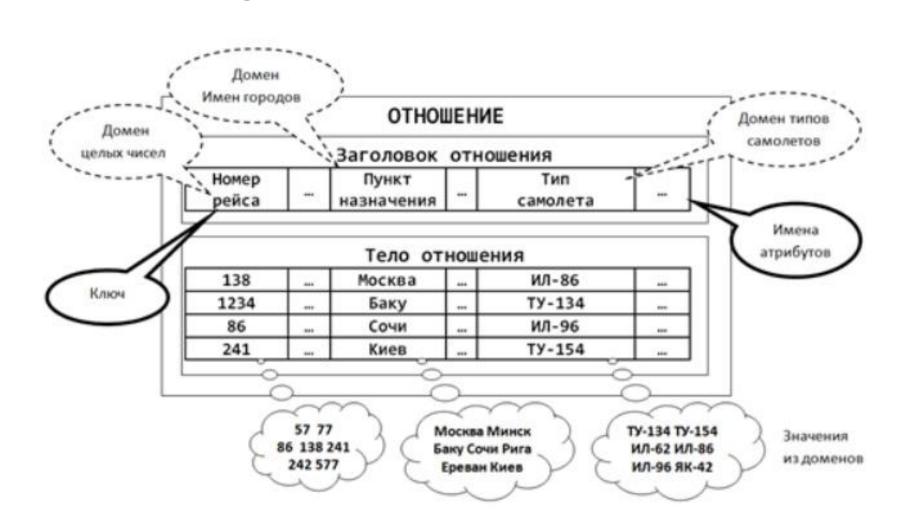
Внутренний слой это работа с сетью, процессами ФС, памятью, уровнями доступа и тд....

Зачем нужны БД в DataScience?

- хранилище исторических данных: обучение моделей, ad-hoc анализ
- хранение моделей ML (например, факторы logistic regression)
- сохранение результатов работы моделей (посчитали эмбеддинги видео-контента сохранили для потомков

Структура данных БД

Данные в хранилищах представлены с помощью отношений (relationship). У отношения есть заголовок (table schema) и ключи (relationship keys). Заголовок состоит из атрибутов.



ключи служат для связи между таблицами.

Схема БД: "звезда" (star)

В центре схемы одна таблица фактов (fact table) — центр «звезды» — и нескольких таблиц измерений (dimension table)

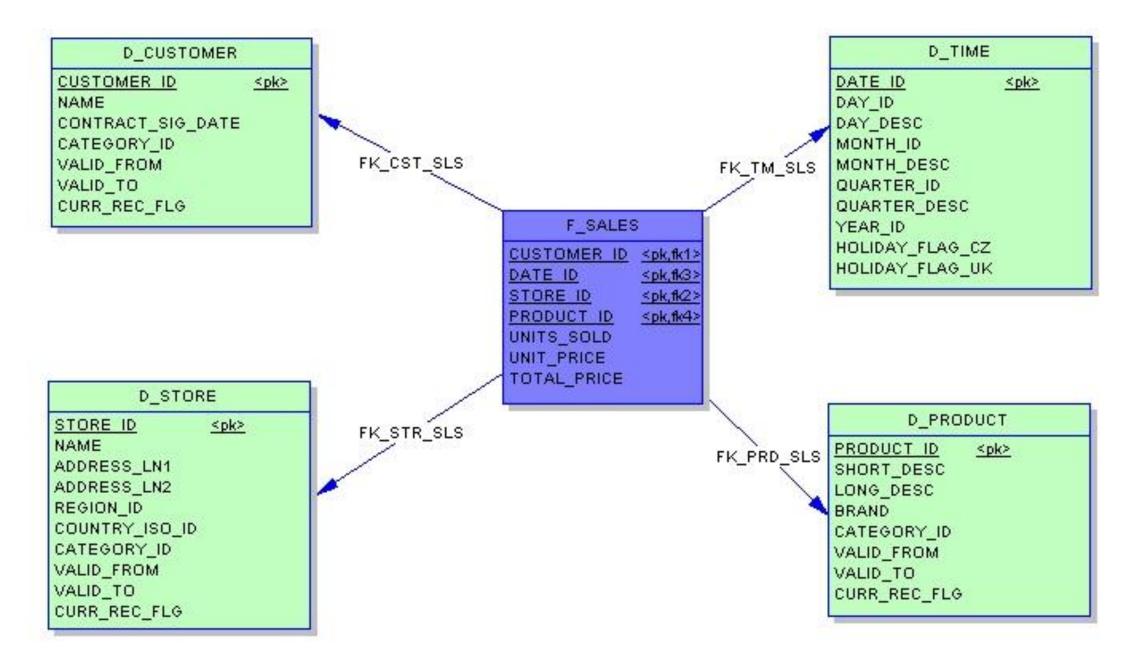
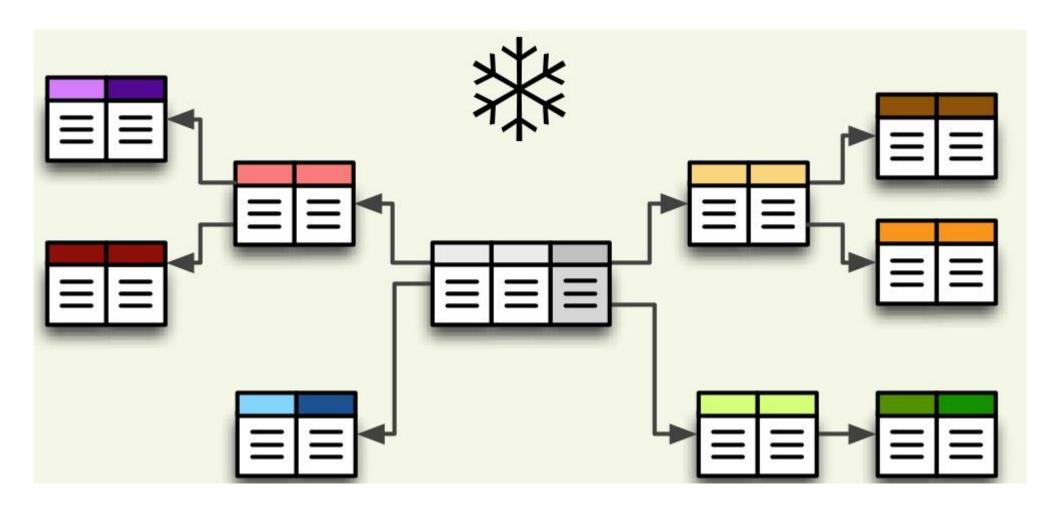


Схема БД: "снежинка" (snowflake)

Снежинка имеет более высокую степень нормализации - каждая таблица "измерений" может иметь свои собственный измерения



Нормальная форма -

требование к структуре таблиц в для устранения из базы избыточных функциональных зависимостей между атрибутами (полями). Всего существует 6 НФ.

Пример 1НФ - все атрибуты являются скалярами, не повторяющихся строк.

Сырые логи обычно не соответствуют требованиям 1НФ.

2-я нормальная форма -

каждый неключевой атрибут неприводимо зависит от ключа.

Пример 2НФ: наличие премии "Оскар" зависит от режиссёра и фильма, а вот рейтинг IMDB зависит только от режиссёра - т.е. 2НФ не выполняется.

Фильм	Режиссёр	Оскар	Рейтинг IMBD
Энни Холл	Вуди Аллен	"+"	8
Быть Джоном Малковичем	Спайк Джонс	"+"	7
Любовь и смерть	Вуди Аллен	"_"	8

Задача на дом

Привести таблицу ко 2-ой нормальной форме Пример

3-я Нормальная форма -

отношение находится в 2НФ

ни один неключевой атрибут R не находится в транзитивной функциональной зависимости от ключа.

Пример: 3НФ не выполняется - по стране действия можно определить наличие "Оскара" (транзитивность)

Фильм	Оскар	Страна действия
Энни Холл	"+"	США
Быть Джоном Малковичем	"+"	США
Любовь и смерть	"_"	Россия

Задача на дом

Привести таблицу ко 3-ой нормальной форме Пример

Самостоятельное изучение
Нормальные формы БД
Статья на Хабре

Часть 2 Базы данных: CLI

Основная БД курса - PostgreSQL

Путь джедая: установить Docker и Docker-Compose

(https://docs.docker.com/docker-for-windows/install/

https://docs.docker.com/compose/install/)

Так же нужно клонировать репозиторий:

https://github.com/Dju999/flask_docker_app

И скачать данные с Kaggle:

https://www.kaggle.com/rounakbanik/the-movies-dataset/version/7#

Либо скачать и установить Postgres для Windows, Mac, Linux: https://www.postgresgl.org/download/

Базы данных: CLI

SQL

Появился в 1974

Structured query language — «язык структурированных запросов») — декларативный язык программирования, применяемый для создания, модификации и управления данными

Пример

Интерфейс командной строки PostgreSQL Пример

Задание

Создать таблички в Postgres со слайда про нормальные формы

Часть 3

Базы данных: обзор

SQL и NO-SQL базы данных

Реляционные БД хранят структурированные данные и отвечают ACID (Atomicity, Consistency, Isolation, Durability — атомарность, непротиворечивость, изолированность, долговечность). Примеры: MySQL, Postgres, Firebird.

Нереляционные (NotOnly SQL DB) хранят объекты с произвольным набором атрибутов - например, нет ограничений на типы данных и кол-во полей. Примеры: MongoDB, CouchDB, Redis. Подробнее тут.

Основые отличия реляционных и нереляционных БД.

Базы данных: обзор

Реляционные БД

Microsoft SQL Server, Oracle Database - монстры энтерпрайза

MySQL и PostgreSQL - аналоги с открытым кодом MariaDB - форк MySQL

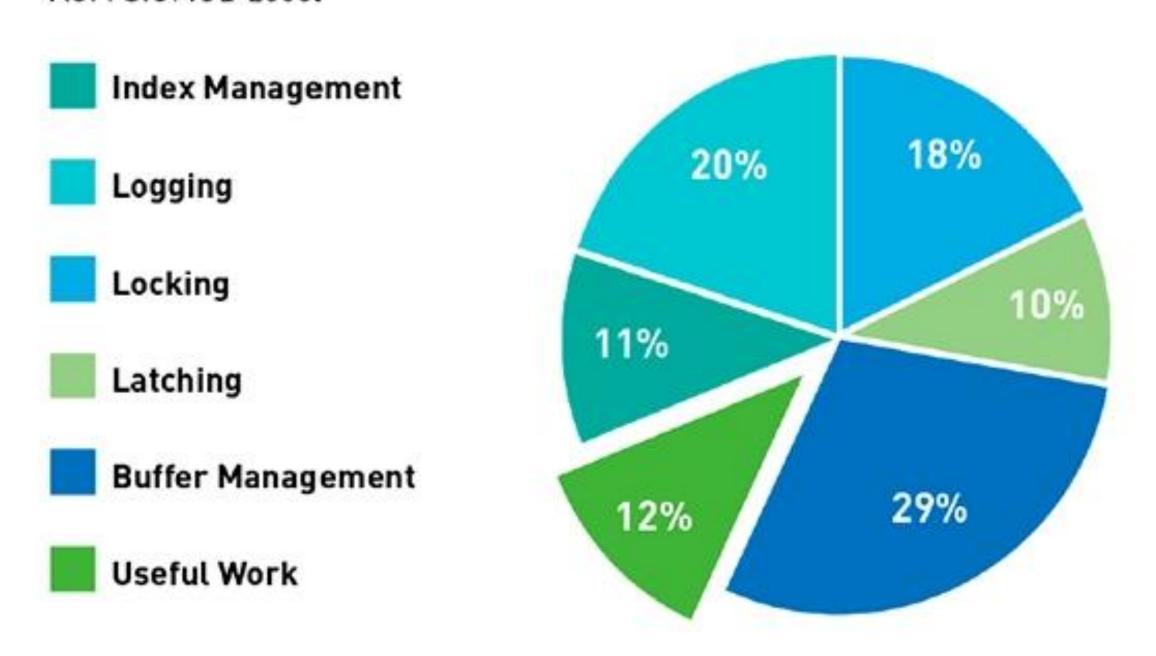
Преимущества: транзакционность

Недостатки: сложность масштабирования

Плата за ACID

General Purpose RDBMS Processing Profile

OLTP Through the Looking Glass, and What We Found There Stavros Harizopoulos, Daniel Abadi, Samuel Madden, and Michael Stonebraker ACM SIGMOD 2008.



Вывод: большая часть ресурсов тратится не на выполнение SQL!

NoSQL базы данных

MongoDB (doc) - вложенные документы

Cassandra (col) - почти транзацкции за счёт доступности

CouchDB (doc) - как Mongo, но на Erlang

Tarantool (key-value) - быстро, но не для любых данных

Memcache, Redis - распределённый кеш

Преимущества: масштабируемость, легко настроить

Недостатки: САР-теорема (невозможно одновременно обеспечить Consistency (непротиворечивость), Availa-bility (доступность), Partition Tolerance (устойчивость к разделению)

NoSQL базы данных

MongoDB:

Коллекция — именованное множество документов, при этом один документ принадлежит лишь одной коллекции. **Документ** — совокупность свойств, включая уникальный идентификатор _id.

BigData

Базы распределённые хранилища поверх HDFS

Hive - есть SQL. Медленно, но надёжно.

Pig - свой процедурный язык, почти MapReduce.

Impala, Presto - не используют Hadoop для вычислений - только возможность HDFS.

ДОМАШНЕЕ ЗАДАНИЕ Ссылка на github

ПОЛЕЗНЫЕ MATEPИAЛЫ

1. NoSQL от Техносферы



Спасибо за внимание!

Джумурат Александр