БАЗЫ ДАННЫХ И СУБД

Глава 1. Введение

Современный мир тонет в огромном количестве постоянно появляющейся информации. В 2017 году объем мировой информации оценивался в 2.7 Зеттабайта (1 Зеттабайт = 10^3 Эксабайта = 10^6 Петабайта = 10^9 Терабайта = 10^{12} Гигабайта). В 2019 году эта оценка выросла до 4.4 Зеттабайт. Чтобы справиться с таким количеством информации нужны соответствующие инструменты, о которых и пойдёт речь. В частности мы познакомимся с базами данных (БД) и системами управления базами данных (СУБД).

В широком смысле понятие "база данных"обобщается до истории любых средств, с помощью которых человечество хранило и обрабатывало данные. В узком же смысле, применяемом в современном понимании, история баз данных начинается с 1955 года, когда появилось первое программируемое оборудование обработки записей. В 1965 году была сформирована Data Base Task Group (DBTG) – рабочая группа, разработавшая в дальнейшем язык описания данных (Data Definition Language) и манипулирования данными (Data Manipulation Language). Но даже спустя более чем 50 лет общепризнанной единой формулировки не существует, поэтому приведем определение на основе международных стандартов ISO/IEC:

Определение 1.1. *База данных – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.*

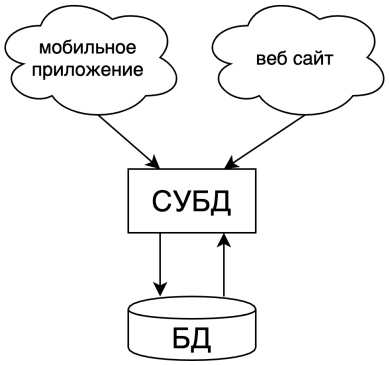

Рисунок 1. Что такое СУБД?

К сожалению наличия одних только баз данных недостаточно, чтобы обеспечить удобную работу с данными. Скажем, данные, хранящиеся в БД, могут

быть необходимы для работы веб сайта, который в свою очередь порождает данные, задействованные для работы мобильного приложения. Для того, чтобы облегчить запись и получение данных используются системы управления базами данных.

Определение 1.2. *СУБД – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.*

Глава 2. Базы данных

2.1. Мотивация

Рассмотрим на примере завода, для чего нам нужны БД. Допустим, вы являетесь директором завода и хотите организовать своих работников, чтобы достичь большей эффективности. Завод является довольно большой организацией, так что в любой момент времени где-то что-то происходит (многие люди записывают или получают информацию). Вам понадобится хранить информацию об:

- людях, которые работают на вас (инженеры, менеджеры и т.д.)
- организации, с которыми вы взаимодействуете (поставщики, заказчики)
- законченные и текущие операции:
  - выплаты зарплат
  - контроль производственных процессов
  - закупки материалов
- распределение плана по цехам
- и т.д...

Вам будет необходимо делиться некоторой информацией с другими организациями, с которым вы занимаетесь бизнесом, а также защищать часть этой информации. Таким образом вам понадобится:

- директор, замдиректора и начальники должны иметь доступ ко всей информации и иметь возможность производить множество операций (закупать необходимые материалы, организовывать поставки готовых товаров и т.д.).
- бухгалтеры должны иметь доступ для контроля денежных потоков (выплата премий и зарплат, оплата контрактов с партнерами).
- рабочие должны иметь возможность запросить необходимые им ресурсы и отчитываться о выполнение поставленных задач.
- полу-публичный интерфейс доступный для партнеров, через который они смогут отслеживать выполнение общих задач.

Сформулируем некоторые требования к информационной системе, которая бы могла удовлетворить наши запросы:

1. **Что описывать:** какие ключевые вещи из реального мира нам нужно описать? Насколько подробно?
2. **Как хранить данные:** можем ли мы использовать текстовые файлы: люди.txt, организации.txt, деньги.txt? Если да, то в каком виде можно записывать информацию?
3. **Контроль над доступом:** как организовать доступ к данным, чтобы бухгалтеры знали о движениях денег на заводе, но не домашние адреса рабочих. А рабочие бы знали о производственных процессах, но не о денежных переводах.
4. **Сбор данных:** каким образом можно получить интересующие нас данные?
5. **Быстродействие доступа:** некоторые данные нам будут нужны мгновенно, а некоторые можно ожидать в течение длительного времени.
6. **Атомарность:** когда бухгалтер переводит деньги из одного места в другое, нам нужны гарантии, что либо деньги взяты из места А и положены в место Б, либо ничего не произошло. В противном случае мы рискуем не досчитаться денег.
7. **Согласованность:** в любой момент времени данные должны быть верными (например, не должно быть двух работников с одним паспортом).
8. **Изолированность:** несколько одновременных заказов материалов не должны перезаписать друг друга.
9. **Устойчивость:** даже если выключится компьютер, у нас должна быть возможность восстановить все данные.

Взглянув на все требования, становится понятно, что попытки вручную заниматься манипуляциями с данными обречены на неудачи. К счастью, БД позволяют удовлетворить все эти запросы.

2.2. Фундаментальные концепты

Базы данных являются микромиром в мире компьютерных наук; их изучение включает в себя: языки, теория, операционные системы, параллельное программирование, пользовательские интерфейсы, оптимизация, алгоритмы, искусственный интеллект, системный дизайн, параллельные и распределенные си-

стемы, статистика, динамическое программирование. Некоторые концепты, на которых мы заострим внимание:

Представление данных

Нам нужен стабильный и структурированный способ представления данных для согласованности и эффективности совместного доступа к данным. Нужные нам концепты:

- **Модель данных:** набор конструктов (или же парадигма) описывающая организацию данных. Например, таблицы, графы, иерархии, объекты, и т.д.
- **Логическая схема:** описание определенных наборов данных, использующее данную модель данных.
- **Физическая схема:** физическая организация данных, т.е. то, как данные и метаданные лежат на дисках.

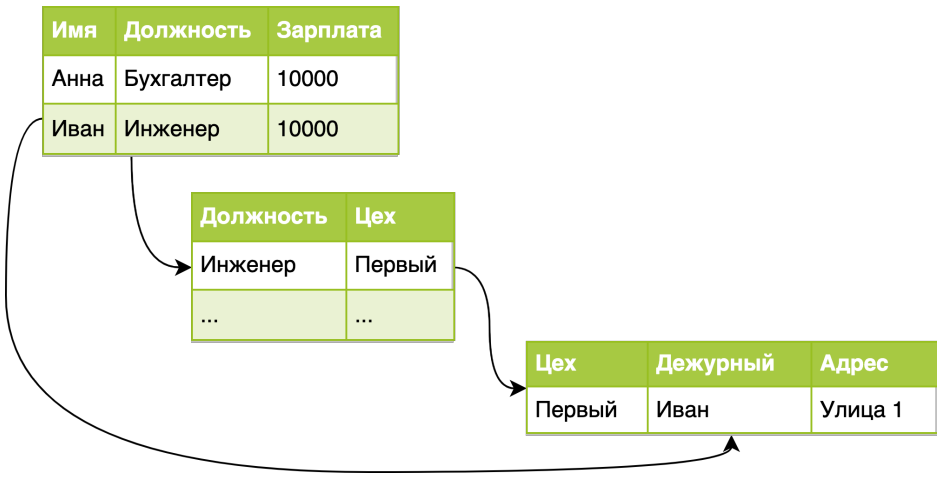

Рисунок 2. Логическая схема части БД завода

Декларативность запросов и обработка запросов

Высокоуровневый язык для описания операций над данными. Цель заключается в том, чтобы гарантировать независимость от данных, отделяя “что” вы хотите сделать с данными от того, “как” это будет достигнуто.

- Высокоуровневый язык для доступа к данным.
- Независимость от данных (логически и физически)
- Оптимизационные приёмы для эффективного доступа к данным.

Транзакции

Базовый блок для доступа и манипуляций с данными.

- способ группировки действий, которые должны произойти атомарно (всё, либо ничего).
- гарантирует переход БД из одного верного состояния в другое
- изолирует от параллельного исполнения других действий / транзакций.
- восстаналиваемы в случае проблем (например, пропадет электричество).

Глава 3. Системы управления базами данных

Задачей СУБД является обеспечение библиотеки изощренных методов и стратегий для хранения, доступа и обновления данных, которые также гарантируют быстродействие, атомарность, согласованность, изолированность и устойчивость. СУБД автоматически компилирует пользовательские декларативные запросы в план исполнения (стратегия выполнения различных шагов для исполнения пользовательских запросов), ищет эквивалентные и более эффективные способы получить тот же самый результат (оптимизация запросов) и исполняет их.

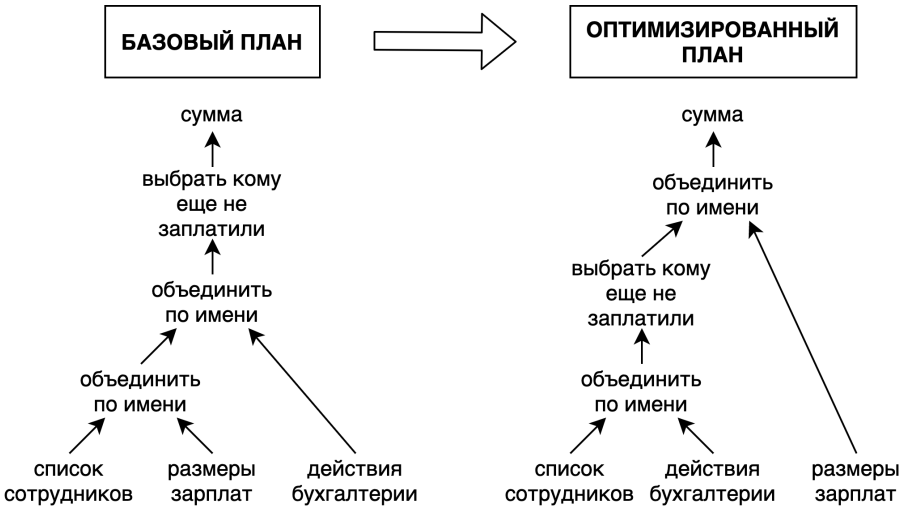

Рисунок 3. Два эквивалентных плана

3.1. Состав СУБД

Обычно современная СУБД содержит следующие компоненты:

- **ядро**, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию;
- **процессор языка базы данных**, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода;
- **подсистему поддержки времени исполнения**, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД;

- **сервисные программы** (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

3.2. Классификация СУБД

В мире существует огромное разнообразие различных СУБД, их можно условно разделить:

- По модели данных
  - иерархические. Данные представляются в виде древовидной структуры.
  - сетевые. Данные представляются в виде графа, т.е. в отличие от иерархической модели у каждой записи-потомка может быть несколько предков.
  - реляционные. Отношения между данными опираются на математическом понятии отношение.
  - объектно-ориентированные. Данные представляются в виде объектов, наделенных свойствами и использующие методы взаимодействия с другими объектами.
  - объектно-реляционные. Сочетает подходы реляционных и объектно-ориентированных СУБД.
- По степени распредленности
  - локальные (все части локальной СУБД размещаются на одном компьютере)
  - распределенные (части СУБД могут размещаться не только на одном, но на двух и более компьютерах)
- По способу доступа к БД
  - файл-серверные. СУБД располагается на каждом клиентском компьютере, а доступ к данным осуществляется через локальную сеть.
  - клиент-серверные. СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно.
  - встраиваемые. СУБД может быть частью некоторого программного продукта, не требуя процедуры самостоятельной установки.