**Лекция №2. Введение в базы данных**

**Что такое данные?**

Данные — это набор фактов в виде слов, чисел или изображений. Данные — один из важнейших активов любого бизнеса. Его используют и собирают практически повсеместно. Ваш банк хранит данные о вас, ваше имя, адрес, номер телефона, номер счета и т. д. Компания, выпустившая вашу кредитную карту, и ваши учетные записи PayPal также хранят данные о вас. Все у вас есть смартфоны, которые используют базы данных контактов. Данные очень важны, и они должны быть в безопасности. Для этих целей используются базы данных.

**Что такое база данных?**

**База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.**

Базы данных повсюду и используются каждый день. База данных — это хранилище данных, программа, которая хранит данные. База данных также предоставляет функциональные возможности для добавления, изменения и запроса этих данных.

Существуют различные виды баз данных. Данные могут храниться в различных формах.

**Модели данных**

Модель данных — это концептуальное описание некоторой предметной области, формальная теория представления и обработки данных в системе управления базами данных (СУБД), включающая как минимум три аспекта:

* аспект структуры: методы описания типов и логических структур данных в базе данных;
* аспект манипулирования: методы манипулирования данными;
* аспект целостности: методы описания и поддержания целостности базы данных.

Различают как минимум следующие модели данных:

* Логические модели:
  + Иерархическая модель
  + Сетевая модель
  + Реляционная модель
  + Объектно-связная модель (ER)
  + Модель «сущность-атрибут-значение» (EAV)
  + Объектно-ориентированная модель (из ООП)
  + Модель документа
  + Модель звезды и модель снежинки
* Физические модели:
  + Плоская модель
  + Табличная модель
  + Перевернутая модель
* Другие модели:
  + Ассоциативная модель
  + Корреляционная модель
  + Семантическая модель
  + XML-модель
  + Многозначность
  + Семантическая сеть и именованные графы
  + Склад троек

**Классификация базы данных**

Базы данных можно классифицировать по модели данных, количеству пользователей, организации системы хранения и так далее.

* По модели данных:
  + Иерархический
  + Сеть
  + Относительный
  + Модель сущность-связь (ER)
  + Модель сущность-атрибут-значение[en] (EAV)
  + Объектно-ориентированная модель (из ООП)
  + Модель документа
  + Модель звезды и модель снежинки
* По количеству пользователей:
  + Личный (рабочий стол)
  + Уровень рабочей группы
  + масштаб предприятия
  + Корпоративный
* По системной организации:
  + Распределенный
  + Централизованный

В настоящее время почти все широко используемые базы данных являются реляционными. О них мы поговорим далее.

**Примеры реляционных баз данных**

SQL/DS, DB2 Universal Database - IBM;

Oracle;

MS SQL-сервер

Postgres SQL;

MySQL;

SQLite.

Думаю, многие из вас слышали об этих базах данных или даже использовали их.

Следующий важный термин:

**СУБД (системы управления базами данных)**

**СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ (СУБД) — набор языковых и программных средств, предназначенных для создания, обслуживания и совместного использования базы данных многими пользователями.**

Набор программных инструментов в базе данных, называемый системой управления базой данных или сокращенно СУБД. Для реляционных баз данных это называется системой управления реляционными базами данных. СУБД — это набор программных инструментов, которые управляют такими аспектами, как доступ, организация и хранение данных. РСУБД служит основой приложений во многих отраслях, включая банковское дело, транспорт, здравоохранение и т. д. Примерами систем управления реляционными базами данных являются SQL Server Management Studio, MySQL, SQLite, Oracle Database, DB2 Warehouse, DB2 on Cloud и т. д.

Рассмотрим, каким требованиям должны соответствовать современные СУБД.

**ТРЕБОВАНИЯ К СОВРЕМЕННЫМ СУБД**

* Масштабируемость - отсутствие существенного снижения скорости выполнения пользовательских запросов при пропорциональном увеличении количества запросов и аппаратных ресурсов, используемых данной СУБД (таких как объем оперативной памяти, количество процессоров и серверов);
* Доступность – возможность всегда выполнить запрос;
* Надежность - минимальная вероятность сбоев, наличие средств восстановления данных после сбоев, средств резервного копирования и дублирования данных, обеспечивающих возможность выполнения этих операций без прерывания работы пользователей;
* Управляемость - простота администрирования, наличие средств автоматической настройки (включает в себя: средства создания баз данных и их объектов, средства описания правил репликации данных, утилиты управления пользователями, средства мониторинга событий, утилиты миграции с других СУБД и т.д.);
* Наличие средств защиты данных от потери и несанкционированного доступа;
* Поддержка доступа к данным с помощью веб-сервисов (веб-сервисов);
* Поддержка стандартных механизмов доступа к данным: ODBC, JDBC, OLE, DB, ADO .NET и др.

**Использование БАЗ ДАННЫХ**

Область применения баз данных очень обширна. Они используются в различных областях, таких как:

* Сбор данных
* Планирование потребности в материалах
* Производство планирования ресурсов
* Enterprise Resource Planning - планирование производственных ресурсов + управление финансами и человеческими ресурсами предприятия
* Обработка корпоративных ресурсов и отношений - планирование ресурсов предприятия + отношения с клиентами и поставщиками, управление цепочками поставок
* Системы поддержки принятия решений
* Анализ данных
* Хранилище данных
* Пространственные и географические базы данных
* Мультимедийные базы данных
* Мобильность и персональные базы данных
* Информационно-поисковые системы
* Распределенные информационные системы
* Всемирная паутина (WWW)

**ОСНОВНЫЕ ФУНКЦИИ СУБД**

Основные функции систем управления базами данных следующие:

* управление данными во внешней памяти
* управление буфером памяти
* управление транзакциями
* логирование и восстановление БД после сбоев
* поддержка языков баз данных

В системах управления реляционными базами данных данные представлены в виде таблиц, определяющих взаимосвязь записей. Реляционные СУБД отличаются простотой, гибкостью и точностью. Каждый из них одновременно работает с данными, размещенными в нескольких таблицах. Поэтому реляционные базы данных ориентированы на быстрый доступ к небольшим объемам данных.

Реляционная база данных представляет собой набор взаимосвязанных двумерных таблиц (отношений). Сотрудник IBM Эдгар Кодд предложил эту модель в 1970 году.

Каждая таблица соответствует одному объекту и состоит из фиксированного количества столбцов (доменов или полей) и строк (кортежей или записей, соответствующих экземплярам объекта). Значения в одном столбце относятся к одному типу. Реляционная модель может описывать иерархические и сетевые отношения.

Преимущества реляционных баз данных:

- Простота (вместо файлов самой разной структуры с разным количеством полей в записях используются простые двумерные таблицы) и

- Гибкость (Не является жесткой, так как связь между объектными таблицами может быть установлена ​​не до выполнения прикладных программ, а в процессе их выполнения. То есть можно не ждать окончания проектирования логической модели, а разрабатывать прикладные программы одновременно с процессом создания логической модели (изменение логической модели не влечет за собой необходимости настройки всех прикладных программ).

Недостаток: Изначально реляционные базы данных характеризовались довольно низкой скоростью доступа к данным. Однако сегодня скорость компьютеров и процессоров растет очень быстро, поэтому этот недостаток уже не является существенным.

Практически все современные СУБД — Oracle, MySQL, MS Access, MS SQL Server, Interbase (Borland) и т.п. - реляционны.

**Что такое SQL?**

**SQL расшифровывается как структурированный язык запросов** - универсальный компьютерный язык, используемый для создания, изменения и управления данными в реляционных базах данных.

SQL — это в первую очередь информационно-логический язык для описания хранимых данных, извлечения из них и изменения данных. SQL не является языком программирования.

Изначально SQL был основным способом взаимодействия пользователя с базой данных и существовал небольшой набор команд (операторов), которые позволяли создавать таблицы, добавлять новые записи, извлекать записи из таблиц (по заданному условию), удалять записи, изменять структура таблиц.

Из-за возрастающей сложности язык SQL стал более прикладным языком программирования, и пользователи получили возможность использовать визуальные конструкторы запросов.

Язык SQL представляет собой набор операторов, которые используются для написания запросов.

Операторы SQL делятся на несколько групп в зависимости от задачи:

− операторы определения данных (Data Definition Language, DDL);

В том числе: Create (создание объекта в базе данных), Alter (изменение объекта), Drop (удаление объекта);

− операторы манипулирования данными (Data Manipulation Language, DML);

В том числе: Select, Insert, Update, Delete.

− операторы определения доступа к данным (Data Control Language, DCL);

В том числе: Grant (предоставление доступа), Revoke (отзыв предоставленного доступа), Deny (запрет доступа).

− Операторы языка управления транзакциями (TCL).

Транзакции представляют собой набор команд, которые выполняются одна за другой. Если все команды были выполнены, транзакция считается успешной. Если во время выполнения возникает ошибка, транзакция откатывается.

К таким операторам относятся:

- begin transaction – команда запуска транзакции;

- commit transaction – изменение команд внутри транзакции;

- rollback transaction - откат транзакции, если что-то пошло не так;

- save transaction - промежуточная точка сохранения внутри транзакции.

***Пример.*** Платежи онлайн. В этой операции банк сначала просит ввести сумму и получателя, затем проверяет и подтверждает операцию, а затем вводит смс-код. Во время выполнения этой цепочки операций вы можете отменить платеж, и транзакция будет откатана.

**SQL-запросы работают с таблицами данных.**

Основные понятия реляционной модели данных:

**Связь (Relation)** — сама таблица, она двумерная и состоит из столбцов и строк.

**Атрибут (Attribute** ) – столбец в таблице, содержащий один конкретный параметр.

**Домен (Domain** ) –валидное значения для каждого атрибута.

**Запись (Record)** (или строка, кортеж) - табличная строка с порядковым номером с информацией об одном конкретном объекте.

**Значение (Value**) – элемент таблицы, расположенный на пересечении столбцов и строк.

**Ключ (Key)** самый важный столбец в таблице. Это обязательно. С его помощью таблицы связаны друг с другом.

**Ключи** могут быть нескольких видов:

- **Primary key** – идентификатор, как индекс или статья;

- **Potential key** – другое уникальное значение, которое может быть идентификатором;

- **Foreign key** – ссылочный столбец для соединения двух таблиц. С помощью внешнего ключа система управления базой данных понимает, как таблицы в базе данных связаны друг с другом.

Далее кратко рассмотрим примеры написания запросов к базе данных. Для демонстрации я буду использовать бесплатный инструмент SQLite Studio. Вы можете скачать его по этой ссылке: <https://sqlitestudio.pl/>.

Документацию по базам данных SQLite можно посмотреть на официальном сайте <https://www.sqlite.org/index.html>.

Мы рассмотрим пять простых команд в SQL для работы с базами данных:

* создать таблицу
* вставить данные для заполнения таблицы,
* выбрать данные из таблицы
* обновить данные в таблице
* и удалить данные из таблицы.

**Заключение**

На ресурсе <https://www.w3schools.com/sql/sql_join.asp> вы можете найти примеры различных операторов и функций SQL с результирующими таблицами.

Другие ресурсы по изучению баз данных:

<https://docs.oracle.com/en-us/iaas/mysql-database/doc/getting-started.html>

<https://docs.microsoft.com/en-us/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver16>

<https://www.sqltutorial.org/>

<https://sqlitestudio.pl/>

<https://www.dbdesigner.net/>