CSC 370 – Database Systems
Introduction

Instructor: Alex Thomo

(http://webhome.cs.uvic.ca/~thomo **)**

What’s a database?

- Database (DB): a collection of information that exists over a long period of time.

- Database Management System (DBMS): a complex software for handling
  - Large data efficiently and safely.

Database [management] system

1. Allows users to easily create new databases and specify their schema.

2. Enables users to easily query and modify the data using a high level language.

Database [management] system

3. Supports intelligent storage of large amounts of data.

   – Protects data from accidental or not proper use.

     Example: We can require from the DBMS to not allow the insertion of two different employees with the same SIN.

   – Allows efficient access to the data for queries and modifications.

     Example: Indexes over specified fields

Database [management] system

4. Controls access to data from many users at once (concurrency), without allowing “bad” interactions that can corrupt the data accidentally.

   – **Example**: a travel agent sees an empty seat in a flight and proceeds to book it. Another agent in another part of the world books the same seat at that very moment.

Database [management] system

5. Recovers from software failures and crashes

   – **Example**: Banking system. A user gets money from an ATM machine and the power fails or software crashes.

     - Bank shouldn’t record debit if it hasn’t given the money to the client.
     - Or, it should be able to record the debit if the money has already been delivered. Errors not tolerated.

Early database systems and file syst.

- First commercial database systems evolved from file systems.
  - File systems allow the storage of big amounts of data.

- But, no query language for the data in files.

  Need programs for extracting elementary information from a set of files.

  Example:

  File Employees with records having fields

  (emp_code, name, dept_code)

  File Departments with records having fields:

  (dept_code, dept_name, dept_manager, dept_location)

  Given an employee "Smith", which department he is working for?

Continued

In the absence of a query language we have to write a program which will:

1. open the file Employees
2. declare a variable of the same type as the records stored in the file
3. scan the file:

   while the end of the file is not yet encountered, assign the current record to above variable.

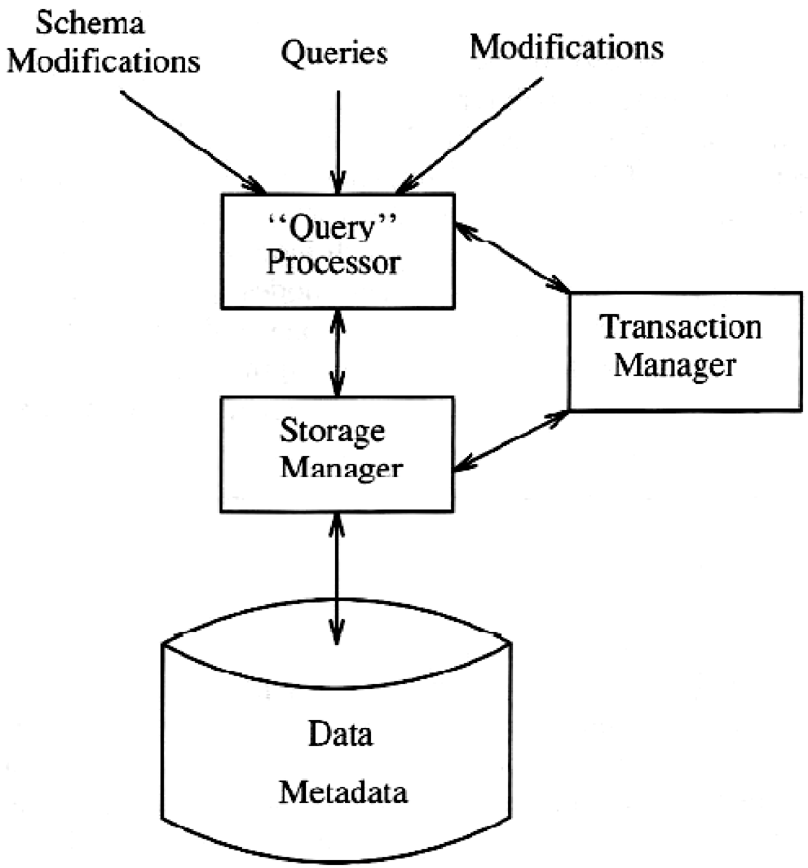
4. If the value of the name field is “Smith” then get the value of the dept_code field. Suppose it is “100”
5. Search in a similar way for a record with “100” for the dept_code in the Department file.
6. Print the dept_name when successfully finding the dept_code.

Painful procedure.

Compare it to the short and elegant SQL query

```
SELECT dept_name
FROM Employees, Department
WHERE Employees.name="Smith" AND
    Employees.dept_code = Department.dept_code
```

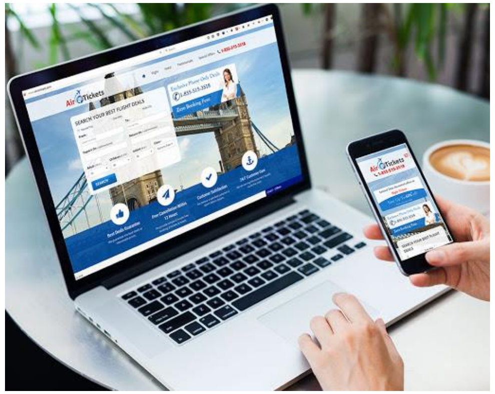
Architecture of a DBMS



First important applications of DBMS’s

Examples

- Airline reservation systems
- Banking systems
- Corporate records



Airline Reservation Systems

- Data items include:
  - **Reservations** by a single customer on a single flight, including such information as assigned seat…
  - **Flight information** –airport they fly from and to, their departure and arrival times…
  - **Ticket information** – prices, requirements, and availability.

- Typical queries ask for:
  - Flights leaving about a certain time from one given city to another, seats available, prices, etc.

- Typical data modifications include:
  - Making a reservation in a flight for a customer, assigning a seat, etc.

- **Many agents** access parts of the data at any given time.
  - DBMS must allow **concurrent access** and prevent problems such as two agents assigning the same seat simultaneously.
- DBMS should also **protect against loss of records** if the system suddenly fails.

Banking Systems

- Data items include:
  - **Customers**, their names, addresses etc.
  - **Accounts**, and their balances
  - **Loans**, and their balances
  - **Connections** between customers and their accounts and loans.

- Typical queries are those for account and loan balances. Typical modifications are those representing a *payment from* or *deposit to* an account.

- In banking systems **failures cannot be tolerated**.

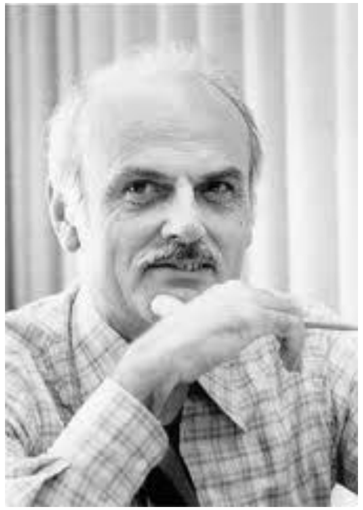
Early DBMS's (1960's)

- Encouraged the user to view the data much as it was stored.
- Chief models were the **Hierarchical** and **Network** models.

- Main characteristic: easy jumping or navigating from one object to another through pointers.
  - E.g. From one employee to his department.

- However these models didn't provide a high-level query language for the data.
- Also they didn't allow on-line schema modifications.

Relational databases



Codd (1970)

- Database system should present the user with a view of data organized as tables (also called relations).

- Queries expressed in high-level language
  - **S**tructured **Q**uery **L**anguage (**SQL**)

- Behind the scene there are complex data structures and algorithms that allow rapid response to queries.
  - But the user would not be concerned with these things.

Trivia

SQL was initially developed at IBM in the early 1970s.

This version, was initially called SEQUEL (Structured English Query Language)

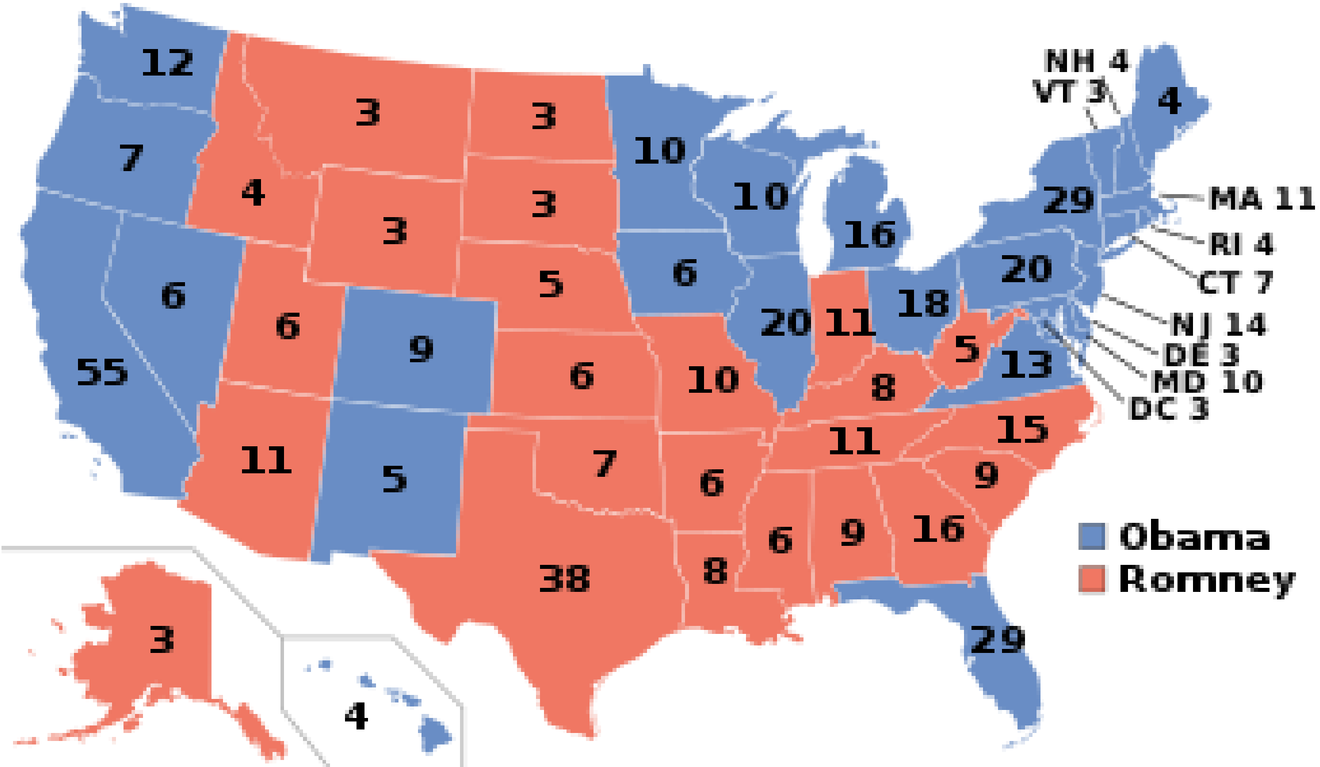
Acronym SEQUEL was later changed to SQL because "SEQUEL" was a trademark of the UK-based Hawker Siddeley aircraft company.

Pronunciation:

SQL ***ESS**-kew-**EL*** or ***SEE**-kwəl*

Databases for Data Science

US Elections 2012



- **Nate Silver** of the **538 blog**, was able to predict this map perfectly before the election.
  - **Media**: talked a lot about, what a genius Nate Silver was … sophisticated mathematics he was using … he's sort of a whiz with these things.
  - **Nate Silver**: methods used to make this prediction were actually **pretty simple**.

- "…the biggest win came from **good old SQL** on a Vertica data warehouse and from providing access to data to dozens of analytic staffers who could **follow their own curiosity** and distill and analyze data as they needed"

Dan Woods

Jan 13, 2013, CITO Research

Bruce Lindsay Speaks Out on System R, Benchmarking, Life as an **IBM Fellow**
https://sigmod.org/publications/interviews/pdf/p71-column-winslet.pdf

*Bruce, I've heard you quoted as saying that relational databases are **the foundation of western civilization**. Can you expand on this point?*

I believe that their adoption in business, government, and education has **enabled the progress we've made in productivity**, and just made things generally better.

I believe that **if the relational databases were to stop right now, you would be stuck in San Diego until they got going again**.

Without these tools to manage the **complex affairs** of **government**, **business**, **education**, and **science** our progress would be really much slower.

Database Studies

- Design of databases.
  - How to structure information?
  - How to connect data items?
  - What constraints should the data satisfy?

- Database programming.
  - How to query and modify the database?
  - How is database programming combined with conventional programming?

- Database system implementation.
  - How does one build a DBMS?
    - query processing,
    - transaction processing
    - organizing storage for efficient access

Main RDBMS’es

- Oracle
- Microsoft SQL Server
- IBM DB2

- PostgreSQL
- MySQL
- SQLite