Introduction to Database Systems

CS 4750 Database Systems

[A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, Ch. 1]

Outline

- Database vs. database Management System (DBMS)
- Why use database and DBMS
- Why learn about database and DBMS
- Roles in DB environment
- Common features of database system

Database

What is a database?

- Organized collection of related data

Database may be as simple as a text file or a csv file or may be as complex as a large relational, integrated collection of data

Give examples of databases

- Bank account database; payroll database; UVA's student database; Amazon's product database; Hotel reservation database; your notes for this class

Why do we need databases (in general)?

- Contain details about the organization or domain application
- Manage large amounts of data – deal with "big data"

Types of Databases

Operational databases

- Collect, modify, maintain data
- Backbone of companies
- Store dynamic data (i.e., change constantly, reflect up-to-the-minute info)

Our focus

Analytical databases

- Store and track historical and time-dependent data
- Asset for tracking trends, viewing statistical data over a long period, making strategic business projections
- Store static data (i.e., never or very rarely change, reflect a point-in-time snapshot of the data, not up to date)

Example: Online Bookseller

1. Purposes/services of the app

- Sell books
- Recommend books
- …

2. Data that should be maintained in a database

- Data about books, customers, pending orders, complete orders, canceled orders, order histories, trends, preferences, …
- Data during one sitting (user session)

3. Functionalities to provide the services

- Insert or remove books
- Find books by author(s), title, …
- Analyze past order history
- …

Let’s Brainstorm

Refer to the brainstorm-scheduler activity on the schedule page

Imagine we are working on a carpool management system to help registered users share car rides with other users traveling on the same route. The user may share a car or ride with another user who is willing to share.

(Note: We are not implementing anything; we will only consider functionalities or services the users may expect from this system and data needed to provide the services. Feel free to explore further ☺)

Go to https://padlet.com/UpsornP/4750-brainstorm-9c77krp5uf0vnkx5 and share ideas:

- What functionalities/services users may expect?
- What data do we need?
- Based on the data we just list, can you think of how they are related? What data should be put together? How should we connect between sets of data?

Needs for Data management

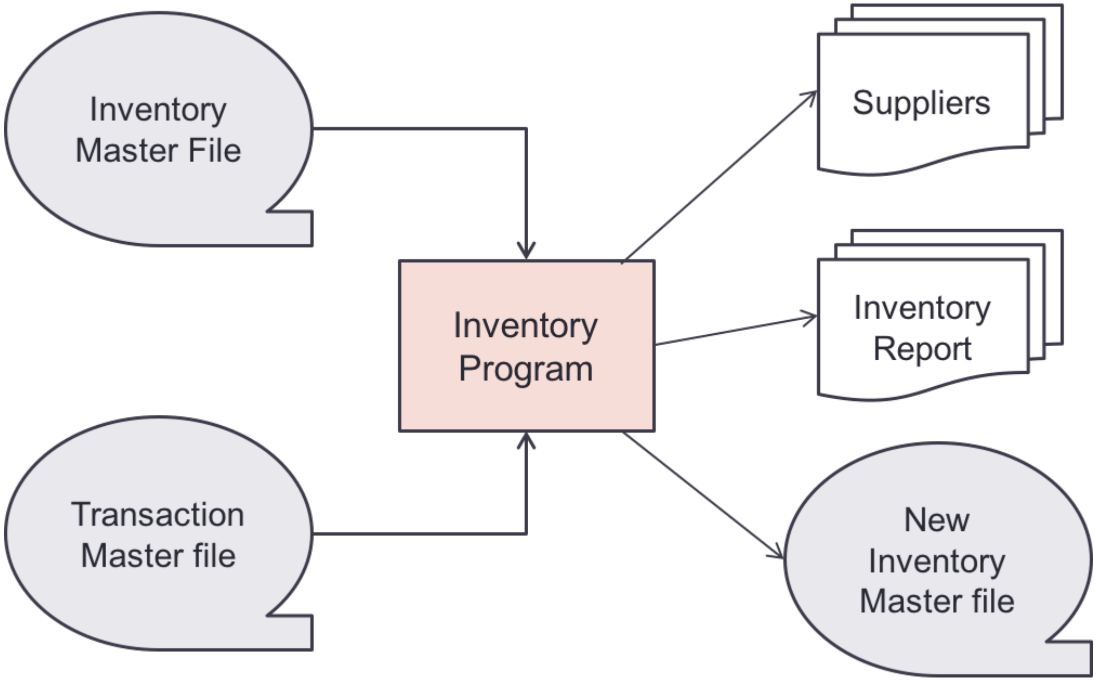
Data must be persistent.
However, data are large – can’t fit all in memory

- Describe real-world entities in terms of stored data
- Persist large datasets
- Efficiently query and update data
- Change structure of data stored (add, update, remove attributes)
- Simultaneously updates
- Recover from failures
- Ensure security and integrity
- Minimize redundancy

Why do we want to minimize redundancy in databases?

- Save space, save cost
- Avoid updating/deleting in multiple places -- consistency
- Prevent data from being corrupt

Early Attempt … to Relational Model



- Rely on file systems
- Programmers must visualize data much as they were stored
- Use several different data models for describing the structure of information in a DB

Ted Codd introduced database systems based on ”**relations**”

[Codd, E.F., “A relational model for large shared data banks,” Comm. ACM, 13:6, pp. 377-387, 1970]

Database Management System

What is a DBMS?

- Software to create, manage, maintain, persist databases over long periods of time

- Manage: data, database engine, database schema

database engine: Allow data to be accessed, locked, and modified

database schema: Define the database's logical structure

Provide concurrency, security, data integrity and uniform administration procedures

Examples:

MySQL, SQLite, MongoDB, PostgreSQL, Oracle, DB2, MS-SQL, Derby

DBMS Properties

- Queryable: Provide a way to ask DB questions and retrieve data
- Durable: Ensure the safety of information stored (data persists)
  - In-memory DB trade durability for speed?
- Have schema: Define structure for storage of information
  - What about Semi-structured DB?
- No redundancy: Reduce space
  - Indexes trade space for speed?
- Optimizes queries: Make query run faster
  - What about complex queries? NoSQL DB has a “WYSIWYG” flavor
- Handle concurrent transactions: Manage database engine
  - Turn off serialization for speed?

Difficult to achieve all – balance and tradeoff

Key Roles in DB Environment

- DB Administrator (DBA): load data, tune system, keep thing running
- DB designers: specify structure (schema) of data to be stored
- Application developers: write programs that access and manipulate data
- Data analyst: clean and correct, mine, integrate data
- DBMS implementer: build DBMS

[Ricardo C.M. and Urban S.D., Databases Illuminated, Ch.1]

Common Features / Key Concepts

- Data models: how to describe real-world data
- Schema: description of tables
- Instance: snapshot of data stored in DB at a given time
- Data Definition Language (DDL): effect schema
- Data Manipulation language (DML): effect instance
- Physical data independence: change how data are stored on disk w/o affecting apps
- Logical data independence: change schema w/o affecting apps
- Query processing and cost estimation: estimate cost of execution, choose the plan with the least estimated cost
- Transactions: atomicity, consistency, isolation, and durability

Wrap-Up

- Database vs. database Management System (DBMS)
- Why use database and DBMS
- Why learn about database and DBMS
- Roles in DB environment
- Common features of database system

What's next?

- DB architecture
- Data model
- Relational data model

- Start thinking about your project, form a team of 3-4 members