Principles of Data Management

Alvin Lin

August 2018 - December 2018

Databases

A **database management system (DBMS)** consists of:

- a collection of interrelated data
- programs to access that data
- and an environment that makes it easy to use

Things that use databases:

- social networks
- retail
- university systems
- banks
- manufacturing

Before databases, things were stored in flat files. This had a few issues:

- data redundancy
- data access was difficult
- data isolation, data is in different formats
- integrity problems

- lack of atomicity
- problems with concurrency
- no built-in security

Database levels of extraction:

- Physical - how it's stored
- Logical - relationships of data
- View - limit of the data

Instances and schemas:

- Physical schema - how the data is physically stored
- Logical schema - restrictions applied to the data
- Instance - value of the data at a point in time
- Physical data dependence - we do not care how we store the data, changing the physical schema should not affect the logical schema.

Data Model

A data model is a collection of tools to describe data, the relationships in the data, the semantics of the data, and the constraints applied to the data. There are a few key models:

- Relational Model - data represents objects from the program
- Entity Relationship Model
- Semi-Structured Model - includes things like XML, there is an overall structure to the document, but it is not concrete
- Object Based Model

Spreadsheets are similar to relational models, the columns represent the type of the data stored and the rows hold related data.

A **data definition language (DDL)** is used to define the database structure, and is stored in the data dictionary. A data manipulation language (DML) is used to access and modify data in the database, often called the query language. Data manipulation languages come in two classes:

- Pure (relational algebra, domain relational calculus, tuple relational calculus)
- Commercial (SQL)

SQL is the most common query language. It is not a turing complete language, but there exist extensions to make it so. Generally, a higher level language is used for complex operations.

Designing a database

- Logical - what is a good collection of data and how does it relate?
- Business Decisions - what do we need to record?
- Technical - where does that data go and how do we represent it?
- Physical Design - database engineers decide how stored data will be written to the disk.

Database Engine

1. **Storage Manager**: handles file organization, indexing/hashing, and manages where the data is physically stored.
2. **Query Processor**: takes your query and sends it to a parser and translator, which is then converted to relational algebra. This is sent to an optimizer, which pulls metadata from the data and uses it to optimize the query. After the query it optimized, it is sent to an execution plan and then executed by pulling data from the appropriate databases. From there, we get output (most often in the form of a table).
3. **Transaction Manager**: deals with worst case scenarios if the system fails. A transaction is defined by multiple physical operations treated as a one logical operation. This part handles transaction and concurrency management.

Users

- Naive users: uses an application that uses a database, without knowledge of it.
- Application programmers: know about the database and the tables that exist to some degree. They may not have access to all data or understand the underlying infrastructure.

- Sophisticated user: uses tools to manipulate the database.
- Administrator: creates and manages the structure and indices of the database.

Database Architecture

- Centralized: all data is localized in one central database.
- Client-Server: a single point of entry into the database, with multiple backends to manage queries.
- Parallel: can be either centralized or client-server, running on multiple processors.
- Distributed: data is distributed among multiple servers.

Timeline of database usage:

- 1950s: data processing done with tapes, input on punch cards, access was sequential.
- 1960-70s: hard disks allowed for random access, network models and the relational data model were invented, as well as transaction processing.
- 1980s: SQL was invented, parallel and distributed systems and object oriented databases were conceptualized.
- 1990s: data mining systems and decision support systems became popularized, web commerce grew, and databases approached the scale of terabytes.
- 2000s: XML and XQuery were heavily used for data, document storage, and document sending.
- Recently: Big Data, BigTable, large scale data analytics into the scale of petabytes and exabytes.

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech