MC 302 DBMS: Unit 1 – Basic Concepts

Goonjan Jain

Department of Applied Mathematics

Delhi Technological University

Outline

- Introduction to DBMSs
- The Entity Relationship model
- The Relational Model
- SQL: the commercial query language
- DB design: FD, 3NF, BCNF
- Indexing
- Transaction Processing
- Concurrency control

Books

- Fundamentals of Database Systems. Ramez Elmasari and Shamkant B. Navathe
- Database System Concepts. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
- Database Management Systems. Raghu Ramakrishnan and Johannes Gehrke

What is the goal of DBMS?

- Electronic record keeping
- **Fast** and **convenient** access of information
- DBMS – Database Management System
  - Commercial systems like – Oracle, SQL Server, MySQL etc.
- Database System – DBMS + data + application programs
- For example – students, taking classes, calculate grades.

One solution: Paper Based

- Advantages –
  - Cheap, easy to use
  - E.g. student folders etc.
- Disadvantages –
  - No ‘ad-hoc’ queries
  - No sharing
  - Large carbon footprint

Next possible solution

- Computer based –
  - flat files + C (JAVA) programs to access them
- Layout –
  - comma separated values (csv)
    - Rohan, 123, A
    - Amit, 239, A+
- Problems?

Problems with File System

- **Data redundancy and inconsistency –**
  - Repetition of data i.e. each data may have more than a single copy.
  - The file system cannot control redundancy of data
  - Each user defines and maintains the needed files for a specific application to run.
  - Changes made by one user does not reflect in files used by second users, which leads to inconsistency of data.
- **Data sharing –**
  - File system does not allow sharing of data or sharing is too complex.
- **Data concurrency –**
  - Concurrent access to data means more than one user is accessing the same data at the same time.
  - Anomalies occur when changes made by one user gets lost because of changes made by other user.
  - File system does not provide any procedure to stop anomalies.

Problems with File System

- **Data searching –**
  - For every search operation, a different application program has to be written.
- **Data integrity –**
  - some constraints need to be applied on the data before inserting it in database.
  - The file system does not provide any procedure to check these constraints automatically.
- **System crashing –**
  - systems might have crashes due to various reasons.
  - In file systems, once the system crashes, there will be no recovery of the data that’s been lost.
- **Data security –**
  - A file system provides a password mechanism to protect the database but how longer can the password be protected?

Why problems?

- Two main reasons-
  - File layout description is buried within the C programs
  - There is no support of concurrency

DBMS handles exactly these two problems

DBMS

- Commercial/freeware DBMS
- Main vendors/products

- In labs we will be using MySQL.
- However, one can opt any RDBMS.

Commercial	Open source
Oracle	MySQL
IBM/DB2	Postgres
MS SQL server	miniBase
Sybase	sqlite
MS Access	

Advantages over flat files

- Logical and physical data independence
  - Data layout, security info etc. stored explicitly on the disk
- Concurrent access
- Transaction processing

Disadvantages over flat files

- Price
- Additional expertise

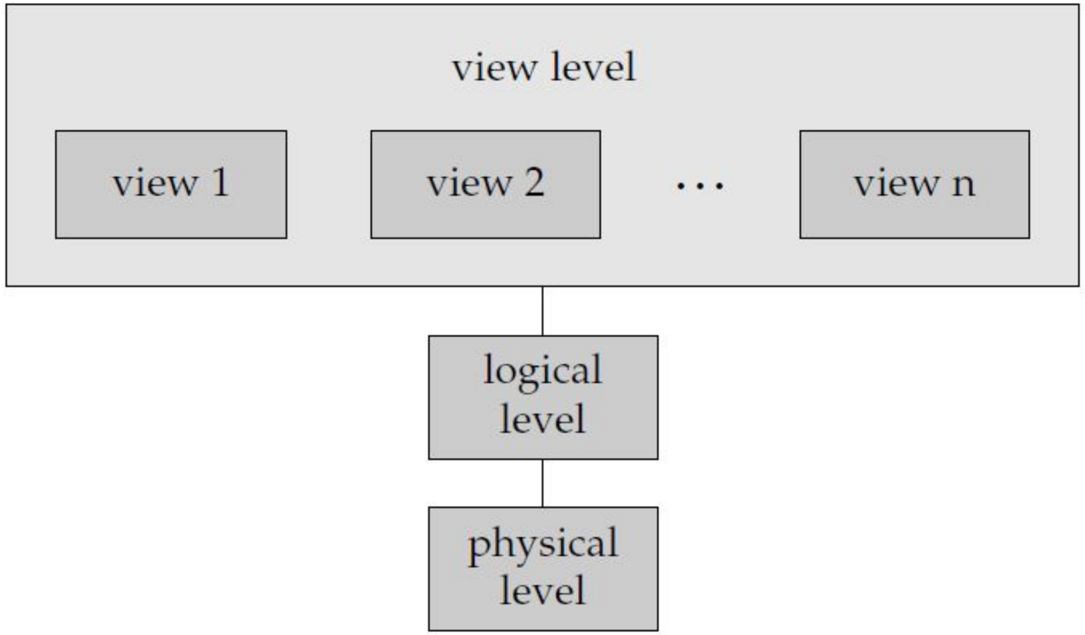
- Hence overkill for small, single-user data sets

Fundamental concepts

- Data Abstraction
- Logical data independence
- Physical data independence

Data Abstraction

- Hiding the irrelevant details
- Several levels of abstraction
  - View level
  - Logical level
  - Physical level



3-level architecture

- View level
  - Describes only a part of DB
  - V1: select roll_no from student
  - V2: Select fac_id, name from faculty
- Logical level
  - What data stored in the DB, and what relationships exist among data. Eg. tables
  - Student(roll_no, name)
  - Faculty(fac_id, name, deptNo)
- Physical level
  - How are these tables stored, how many bytes/attributes etc.

Schema and Instances

- Schema
  - overall design of DB
  - Changes infrequently
- Instance
  - Collection of information stored in DB at a particular moment

DB schema corresponds to variable declaration.

Value in variables at a point of time correspond to an instance of a DB schema

DB Schemas

- Several schemas partitioned according to levels of abstraction
- Physical schema
  - schema at physical level
  - Can be changed without affecting application programs
- Logical Schema
  - schema at logical level
  - Programs construct applications using logical schema
- Several schemas at view level, called subschemas

3-level architecture

- Logical data independence
  - Can add/drop column; add/drop table
- Physical data independence
  - Can add index; change record order

Data Models

- Collection of conceptual tools
  - Describe data
  - data relationships
  - data semantics
  - consistency constraints
- Relational model
- Entity-Relationship model
- Object-Based data model
- Sem-structured data model

Relational Model

- Collection of tables to represent
  - Data
  - Relationship among data
- Each table has multiple columns
- Each column has a unique name
- Tables also called relations

Entity-Relationship model

- Collection of entities and relationships objects
- Entity – ‘thing’ or ‘object’ in real world, distinguishable from other objects

Database Languages

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Query Language (DQL)
- Data Control Language (DCL)
- Transaction Control Language (TCL)

Data Definition Language (DDL)

- define the database structure or schema
- CREATE - to create objects in the database
- ALTER - alters the structure of the database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- RENAME - rename an object

Data Manipulation Language (DML)

- managing data within schema objects
- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain
- LOCK TABLE - control concurrency

Data Query Language

- get some schema relation based on the query passed
- SELECT
  - retrieve or fetch data from a database
  - fetch either the entire table or according to some specified rules
  - data returned is stored in a result table, called result-set

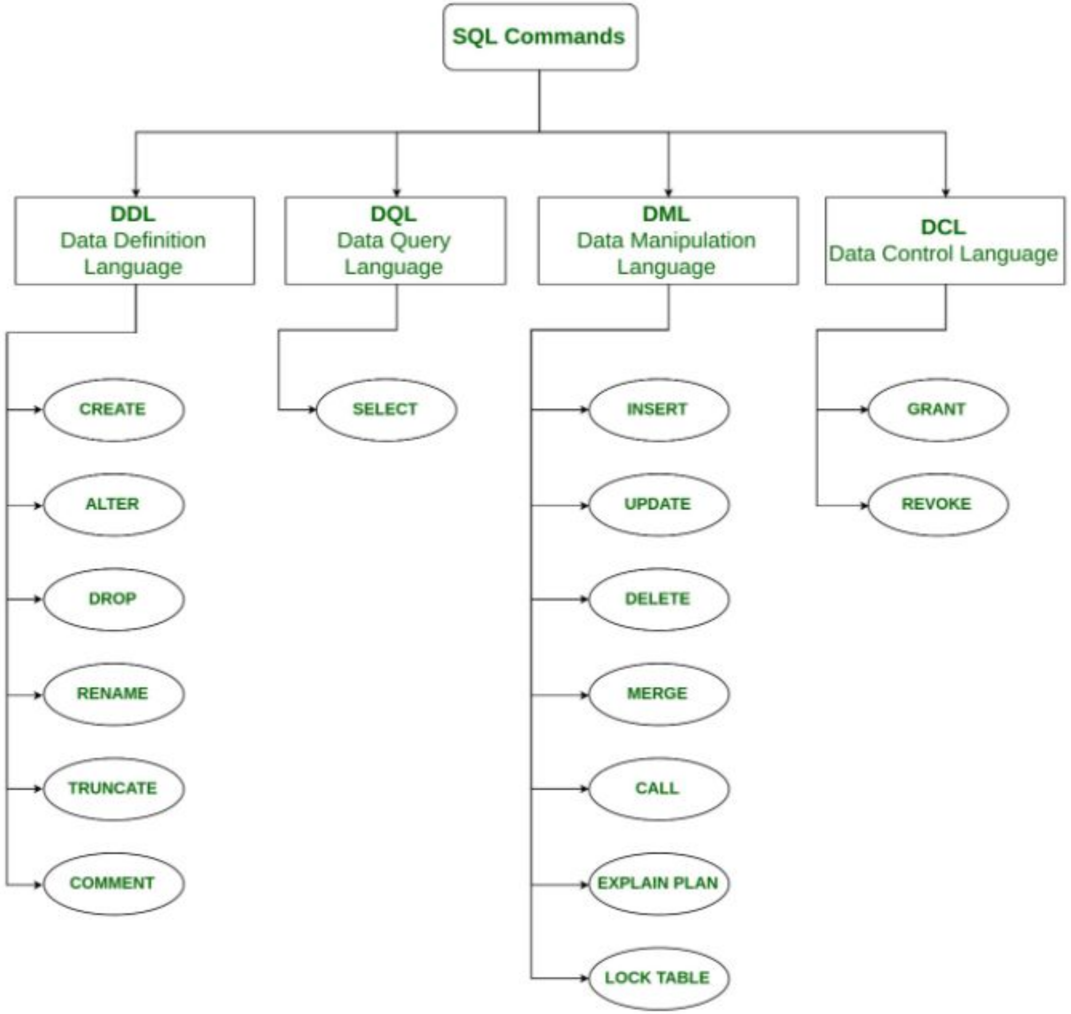
Data Control Language (DCL)

- Deals with the rights, permissions and other controls of the database system.
- GRANT - gives user's access privileges to database
- REVOKE - withdraw access privileges given with the GRANT command

Transaction Control Language (TCL)

- allows statements to be grouped together into logical transactions.
- COMMIT - save work done
- SAVEPOINT - identify a point in a transaction to which you can later roll back
- ROLLBACK - restore database to original since the last COMMIT
- SET TRANSACTION - Change transaction options like isolation level and what rollback segment to use

Types of SQL Commands



Database Users

- ‘naïve’ users
- Casual users
- Application programmers
- DBA – database administrators

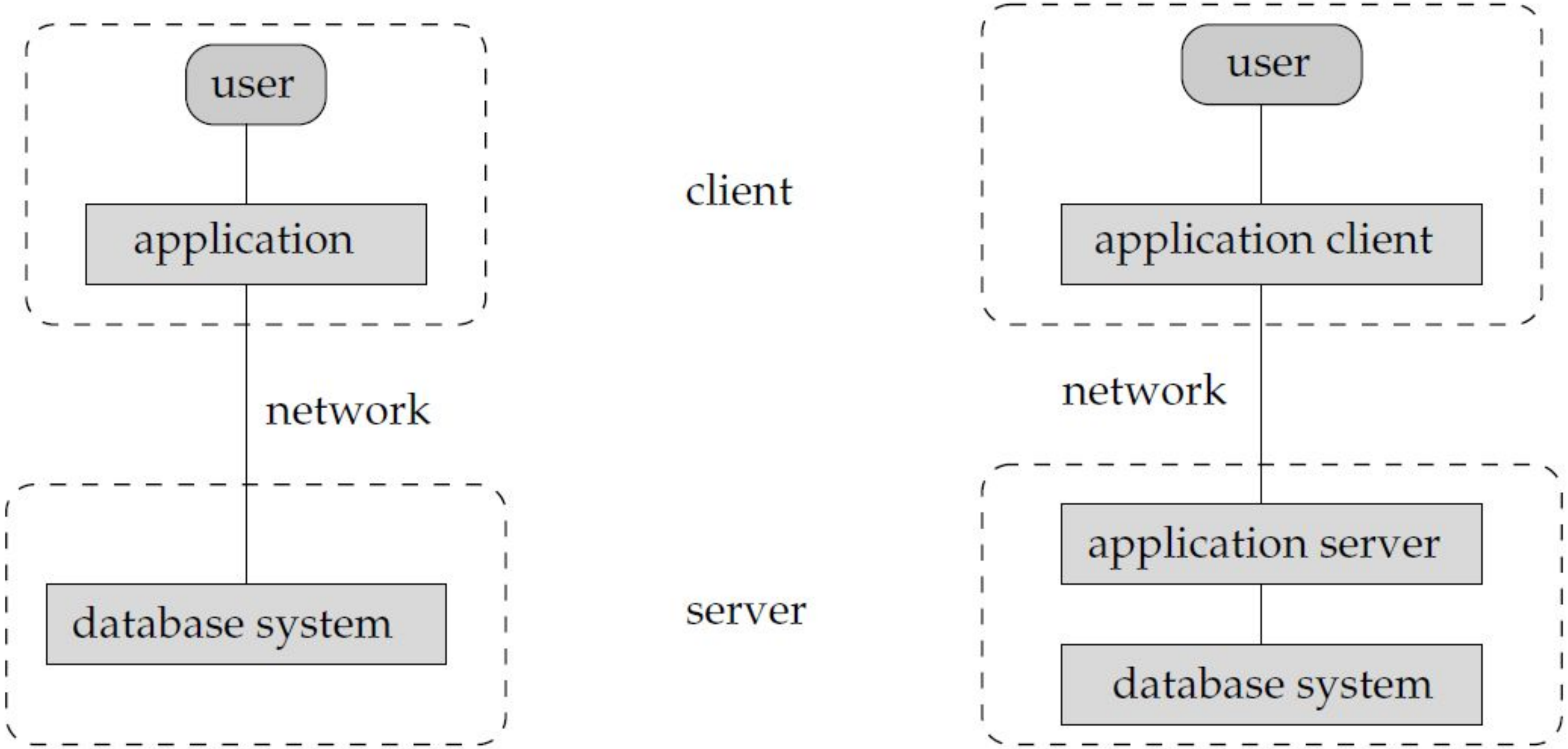
Database Administrator (DBA)

- Schema definition (logical level)
- Physical schema (storage structure, access methods)
- Schema modifications
- Granting authorizations
- Integrity constraint specification

Database Architecture

- DB systems can be
  - Centralized
  - Client-server
- Distributed DBs span multiple geographically separated machines

Application Architectures



Conclusions

- (Relational)DBMSs – electronic record keeper
- Customize them with CREATE table commands
- Ask SQL queries to retrieve data
- Advantages over flat file systems
  - Logical + physical data independence
  - Concurrency control and recovery