Introduction

- What is a database?
- Persistent vs. non-persistent data.
- What is a DBMS?
- Why use a database?
  - Redundancy, shareability, inconsistency, standards, integrity, data independence
  - Advantages of databases over traditional file systems.
- When not to use a DBMS?
- Database Models (relational, hierarchical, network and object oriented)
- Data Models (set of concepts used to describe the structure of a database)
  1) high level or conceptual (entities, relationships, attributes) ER
  2) Representational or Implementation data model (relational hierarchical, network, OO)
  3) Low level or physical (record format, access mechanism)
- Database Schema (structure or intension of the database)
- Database State (occurrences, instances or the extension of the database)
- DBMS architecture (ANSI/SPARC model)
  1) External view (individual user or programmer's view)
  2) Conceptual view (representation of the entire database as known to user community)
  3) Internal view (storage view)
- Physical and logical data independence
- Data sub-languages (DDL and DML)
- Database query facilities (procedural and non-procedural)
  1) Form based (QBE)
  2) Menu based
  3) Natural language
  4) Query language (SQL)

ER Modeling

- ER Modeling (a conceptual design tool for database design)
  1) Entities
  2) Relationships
  3) Attributes
- Ability to produce an ER based on a description of an organization.
  Types of attributes (simple, composite, single and multi valued, stored, derived and NULL)
- Keys, composite keys, domains, weak entities, owner or identifying entities
- Recursive relationships, degree of a relationship
- Cardinality and ordinality.
Relational Model
- Relational Data Model and Relational Algebra
- Definitions for database, relation, attribute, tuple, domain under the relational model.
- Relation Schema
- Degree of a relation
  - Cardinality of a relation (# of values in the domain, depth of table, varies over time)
- Basic properties of relations: (no duplicate tuples, unordered, all attributes are atomic)
- Relational model constraints:
  1) Domain constraints
  2) Key constraints
  3) Entity integrity constraints
  4) Referential integrity constraints
  5) Semantic integrity constraints
- Keys, super keys, candidate key, primary key, alternate key, foreign key
- Relational Algebra Operations on Relations:
  1) Update operations:
     - (insert, delete, modify)
  2) Retrieval operations:
     - (select, project, join, union, intersection, difference, cartesian product, divide)
- Steps in defining a database.
  1) Data model
  2) Data Definition Language
  3) Data Manipulation Language
- ER to Relational Mapping

SQL
- SQL - Structured Query Language
- History and Standard
- Data definition:
  - Create schema, create table, create domain,
  - Alter
  - Drop
  - Primary key and foreign key, not null and default constraints.
- Data manipulation language.
  - Insert / into / values
  - Delete / from / where
  - Update / set / where
  - Select / distinct / From / Where / groupby / having / orderby
- Aggregate functions:
  - Count, sum, max, min, avg

Normalization and Formal Methods
- Functional Dependencies and Normalization
- Informal Design guidelines: (semantics, of attributes, reduce redundancy, reduce null values, disallow spurious tuples.)
- Formal Design Guidelines: (Functional dependencies, Normal forms.)
- Update Anomalies
  1) Insertion anomalies
  2) Deletion anomalies
  3) Modification anomalies
- Definitions for FD, FFD, 1NF, 2NF, 3NF
- Ability to take a relation and apply the FD’s and Normal forms to them, showing the resulting relation(s) after each transformation.