Advantages of DBMS:

- Data & execution abstraction
  - Data Model
    - A collection of high level data description constructs that hide low-level storage details
  - The relational model
    - Most common data model
    - Main construct is relation: table of records
    - Every relation has a schema
      - Relation name
      - Name of fields
      - Types of fields
- Reliability
  - Enforcing Integrity Constraints, such as
    - Data types
    - Value Ranges
    - Certain rules on records
  - Backup and recovery
  - Restricting unauthorised access

- Efficiency & Performance
  - Performance problems with files
    - Redundancy
    - Inconsistency-waste of effort
  - No performance problems with DB
  - Space efficiency
    - Minimises data redundancy by storing data only once
  - Time efficiency
    - Eliminates the need for multiple updates to keep the replicas consistent and up to date
    - Enhances query performance by means of optimisations and access methods
    - Allows many users to access and share database concurrently

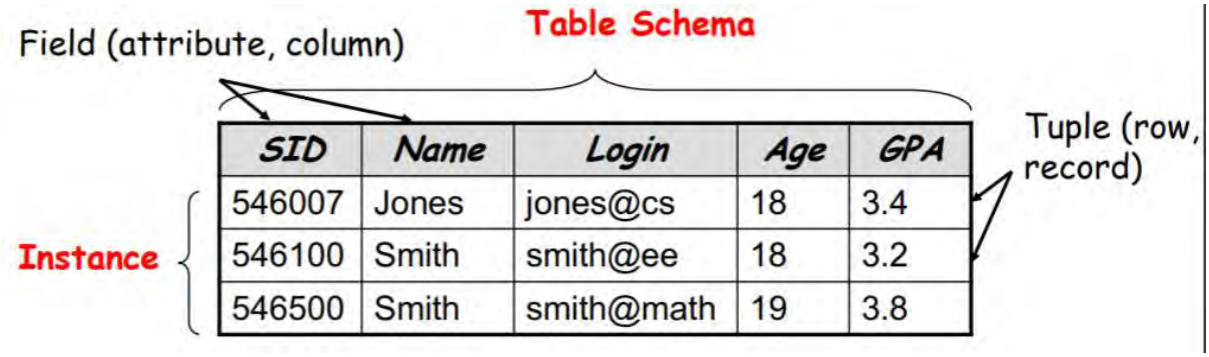
Database Languages:

- Data Definition Language
  - Define Schemas
  - Define Integrity Constraints i.e. unique SID's
- Data Manipulation Language(DML):
  - To ask questions = Query i.e which students have GPA >3.
  - To insert, delete and update data
  - SQL most used database language

When an SQL-DBMS is Inappropriate?

- Disadvantages:
  - Price to buy
  - Additional expertise(SQL/DBA)
- Hence, it is over-kill when
  - The database has simple structure and/or its size is small

- The application is simple, special purpose and is not expected to change
- Concurrent, multiple user access is not required
- Can tolerate failures



Schema vs Instance:

- Instance: the data in the database at a particular moment in time
  - Aka database state or snapshot
- Database changes frequently:
  - Each insert, delete, or update → change from one state of the database into another state
- Schema is fairly static:
  - Changes occur as the application requirements change

```
SELECT   S.name
FROM     Sailors S
WHERE    S.rating > 8
```

Find all sailors with rating > 8

```
SELECT  S.sname
FROM    Sailors S, Reservations R
WHERE   S.sid = R.sid
        AND R.BID = 103
```

Joins

DISTINCT is an optional keyword indicating that the answer should not contain duplicates.

Conceptual Evaluation Strategy:

- Semantics of an SQL query defined in terms of the following conceptual evaluation strategy
  1. Compute the cross-product of from-list
  2. Discard resulting tuples that fail qualifications
  3. Delete attributes that are not in select-list
  4. If DISTINCT is specified, eliminate duplicate rows

Aggregate Operators:

1. COUNT ([DISTINCT] A):
The number of (unique) values in column A

2. SUM ([DISTINCT] A):
The sum of all (unique) values in column A

3. AVG ([DISTINCT] A):
The average of all (unique) values in column A

- Find the average age of all sailors with rating 10:

```
SELECT AVG (S.age)
FROM Sailors S
WHERE S.rating = 10
```

4. MAX (A):
The maximum value in column A

5. MIN (A):
The minimum value in column A

Consider: Find the number of sailors at each rating level (1,2, …) that has at least 2 sailors

```
SELECT S.rating, COUNT(*)
FROM Sailors S
GROUP BY S.rating
HAVING COUNT (*) > 1
```

Conceptual Evaluation:

1. Compute cross-product of tables in from-list

2. Apply the qualifications in the WHERE clause

3. Eliminate unwanted columns

❖ Only columns in SELECT, GROUP BY, or HAVING are necessary

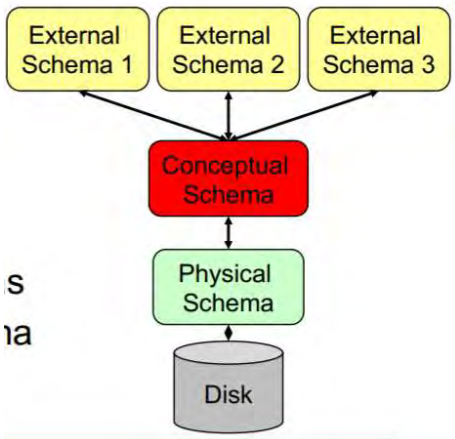
4. Sort table according to GROUP BY clause

5. Apply group-qualification in the HAVING clause

6. Generate one answer for each remaining group

Levels of abstraction in DBMS:

The data in a DBMS is described at three levels of abstraction:



Conceptual Schema:

❖ the process of designing a conceptual schema
❖ Describes data in terms of data model

Physical Schema: