CS 220

Database Systems

Fall 2019

Course Information

Credit Hours : 3+1

Instructor : Mr. Burhan Abbasi
Email : babbasi.mscs16seecs@seecs.edu.pk

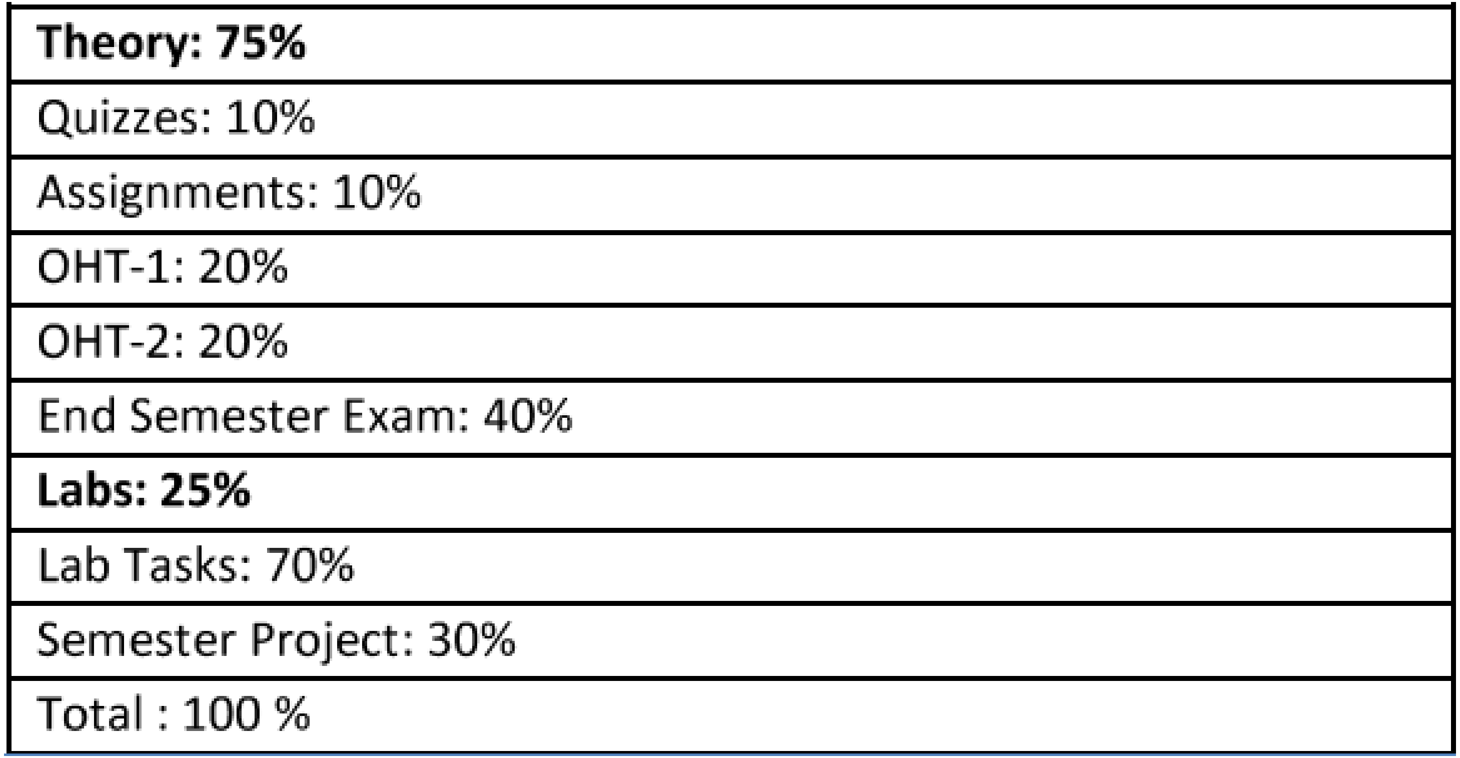
Lab Engineer : TBD

Grading Components:

- Quiz (3)
- Assignments (3)
- OHT 1+OHT 2+ ESE
- Lab (Lab Work + Semester Project)

Attendance Requirements : Above 75%

Assessment Criteria



Theory: 75%
Quizzes: 10%
Assignments: 10%
OHT-1: 20%
OHT-2: 20%
End Semester Exam: 40%
Labs: 25%
Lab Tasks: 70%
Semester Project: 30%
Total : 100 %

Course Resources

Text Book :

R. Elmasri, S.B. Navathe (2016): Fundamentals of Database Systems, 7/E, Addison-Wesley

Reference Books :

T. M. Connolly, C. E. Begg (2015): Database Systems, 6th Edition, Addison-Wesley

J. A. Hoffer, V. Ramesh, and H. Topi (2013): Modern Database Management 11/E, Pearson

Silberschatz, Korth and Sudarshan (2010): Database System Concepts 6/E, McGraw-Hill

What is the End Goal?

Knowledge & Skill

Lecture 1

Traditional Approach to Organize Data

Store the data in files

Files may be of different formats i.e. xls, doc, ppt, txt

Write application specific code to manage it

Could there be any challenges?

Database

Database : **Collection of data**

- ➔ collection of logically related data for a particular domain
- ➔ may consist of **Entities** & their **Relationships**

Can you think of an example of a Database?

*Data : Known facts that can be recorded and have meaning

How to Manage a Database ?

- Database Management System (DBMS)
- DBMS is software designed to assist in maintaining and utilizing large collections of data
- Examples : Access, Oracle, DB2, MySQL, SQL Server, SQLite

Question:
Can different applications interact with the same DBMS?

Can there be different users?

Database Systems Allow for following:

Efficient : handle complex queries

Convenient : easier to write queries

Massive : huge size gigabytes, terabytes, petabytes

Persistent : data stored even after program ends

Multi-user : consistent information even in case of multiple users

Database Systems - Architecture

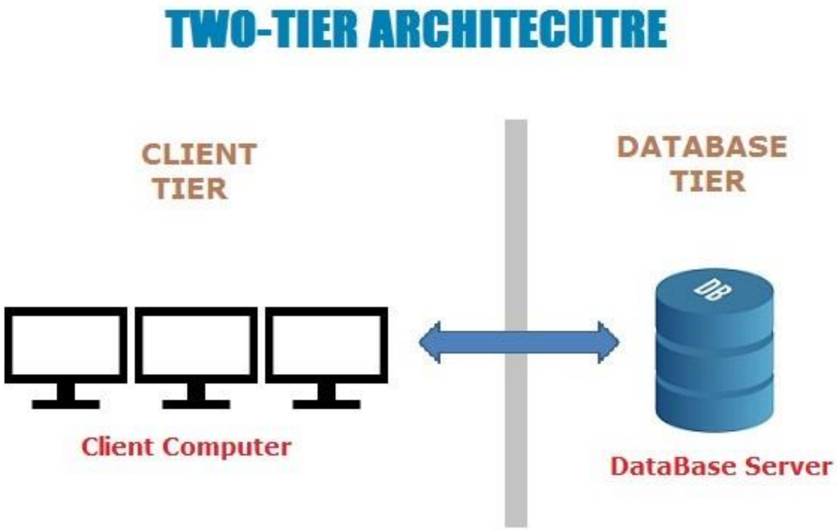
- Functionality is distributed between two types of modules
  - Client Module
    - Handles User Friendly Interface
    - Can run on mobiles and PCs
  - Server Module
    - Handles data storage, access, request processing

Database Systems- Architecture

- Different variations in capabilities of Server & Client
- File Server Architecture
  - Files are shared on server
  - Clients do the processing
  - SQLite, Microsoft Access

Architecture - Two-Tier Client-Server Architecture

- Dedicated Machine running DBMS
- Clients only access information
- SQL Server



Advantages:

Easier to maintain

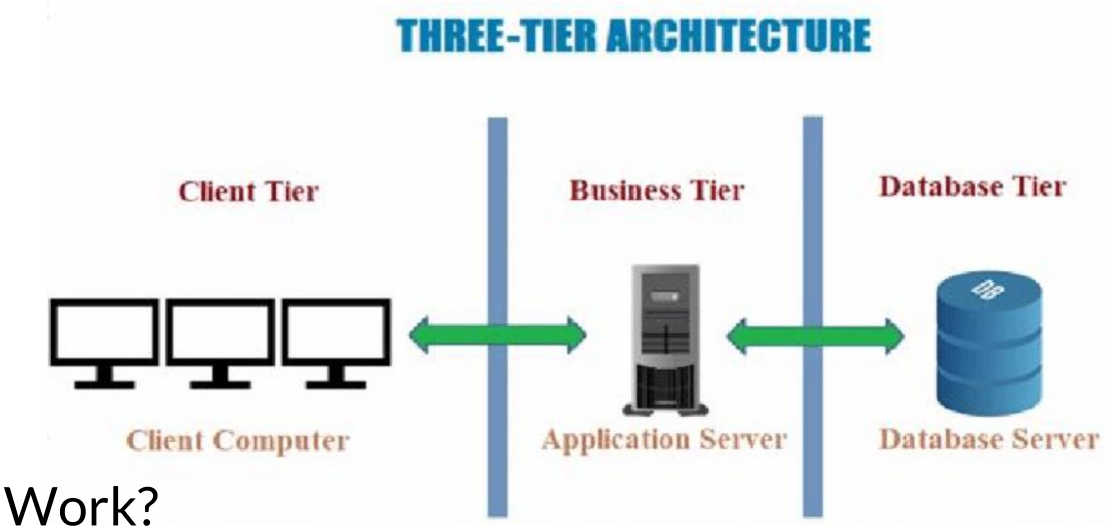
Reduced Hardware & Communication cost

Architecture- Three-Tier Client-Server Architecture

- Client
- Application Server
  - Business Logic
- Database Server
  - DBMS

Example

How Does **Google** Work?



Advantage :

Easier to Scale. **Why?**

Database Users

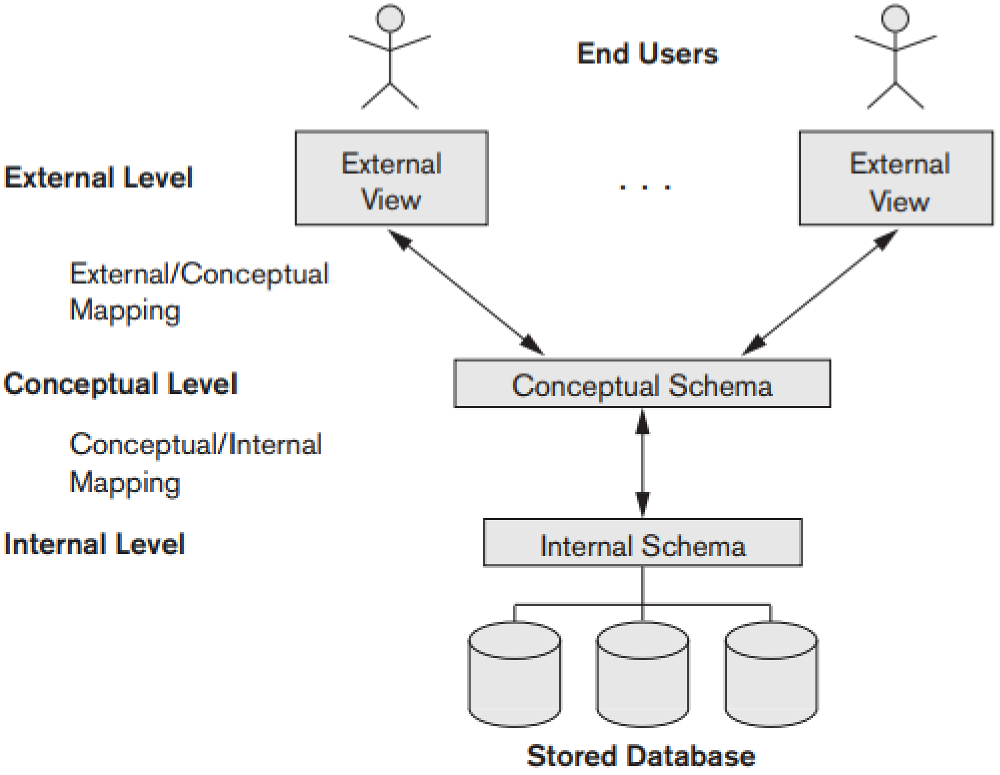
- People can interact with a database in different capacities
  - User
  - Application Developer
  - DBMS Developer
  - Database Designer
    - Logical Design : Requirement gathering, business rules
    - Physical Design : Physical storage policies, security constraints
  - Data Administrator
  - Database Administrator

How Database Systems Make Life Easier

Database Systems provide **Data Abstraction**

- Data Abstraction means hiding unnecessary information from user.
- There are three levels of abstraction:
    - External (View) level abstraction
        - End user can access data without worrying about anything
        - Multiple views of same database can exist
    - Conceptual (Logical) level abstraction
        - Programmer can access data and see logical relationships between data elements
    - Internal (Physical) level abstraction
        - Database Administrators manage actual storage, size and performance of database

Data Abstraction in Three Schema Architecture



Data Abstraction Leads to Data Independence

The capacity to change the schema at one level of a database system without having to change the schema at the next higher level.

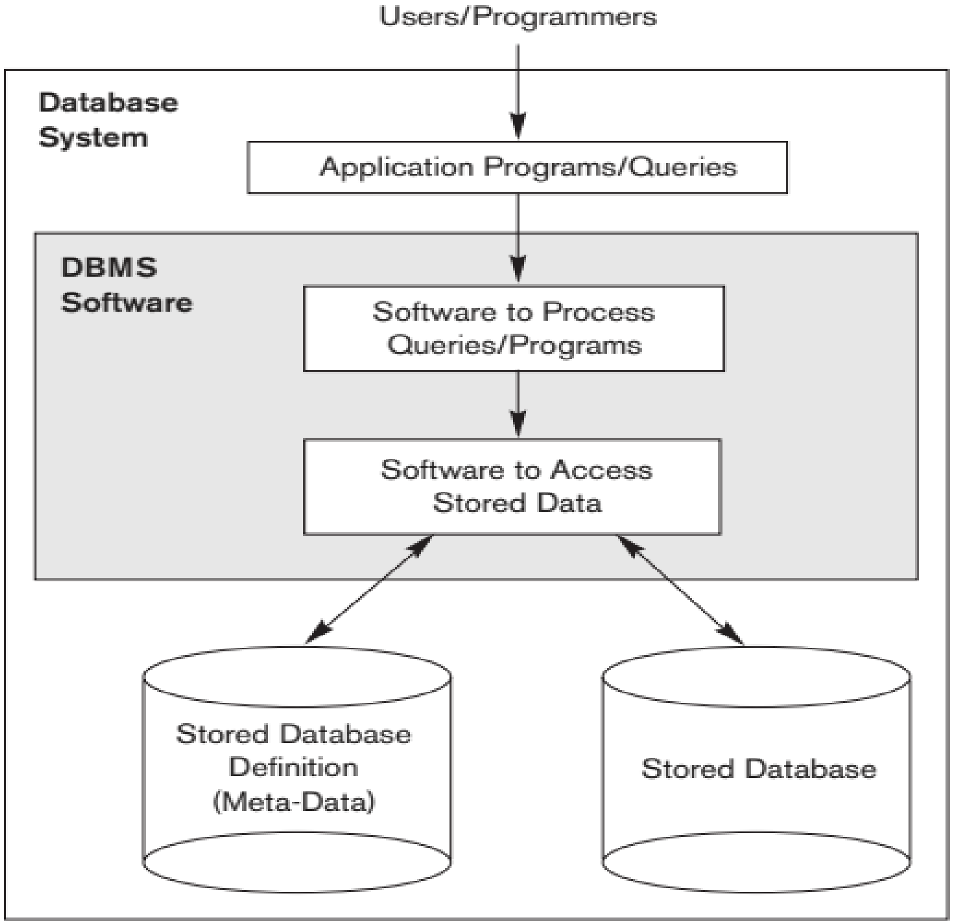
Logical data independence

Change the conceptual schema without having to change external schemas or application programs e.g. Add columns, apply or remove constraints

Physical data independence

Change the internal schema without having to change the conceptual schema e.g. change in storage scheme

Example of a Database System

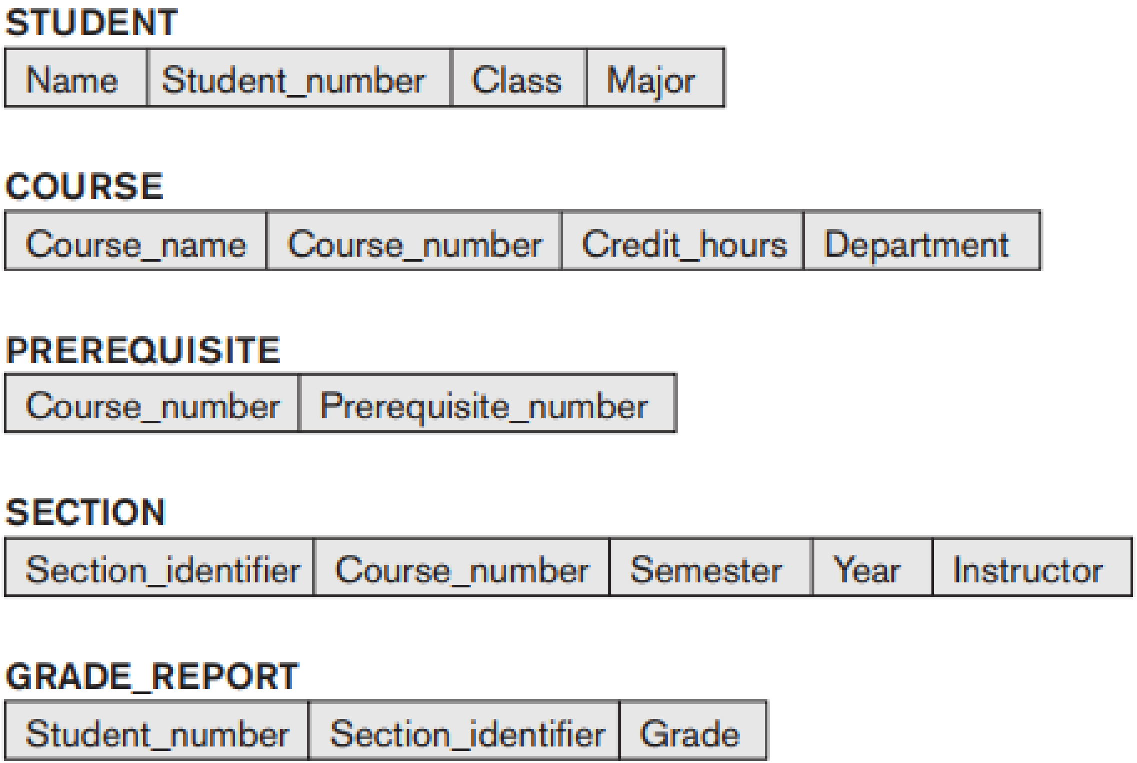

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

QUESTIONS ???

Schema & Meta Data

- The description of a database is called the database schema
- The information about the data is called meta-data.
    - Structures - How?
        - Tables & Columns in each table
    - Data Types - What?
        - Data type of each column
    - Constraints - What Not?
        - Limitations & Rules to be applied on data

Schema



STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

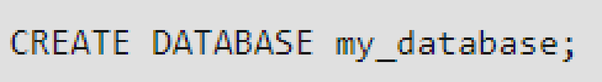
Student_number	Section_identifier	Grade

Structured Query Language - SQL

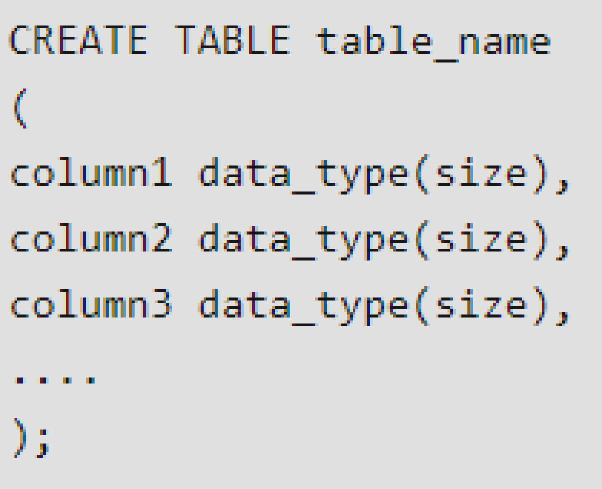
- SQL lets you access and manipulate databases
- 3 types of commands
  - DDL - Data Definition Language
  - DML - Data Manipulation Language
  - DCL - Data Control Language

DDL - Data Definition Language

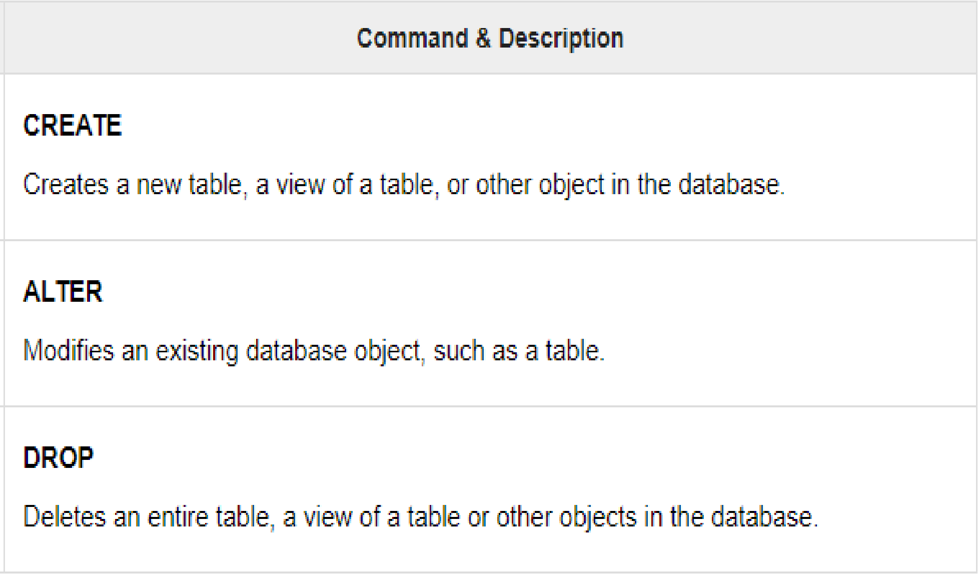
The DDL allows the user to create data structures in the data model used by the database



```
CREATE DATABASE my_database;
```



```
CREATE TABLE table_name
(
column1 data_type(size),
column2 data_type(size),
column3 data_type(size),
....
);
```



Command & Description
CREATE Creates a new table, a view of a table, or other object in the database.
ALTER Modifies an existing database object, such as a table.
DROP Deletes an entire table, a view of a table or other objects in the database.

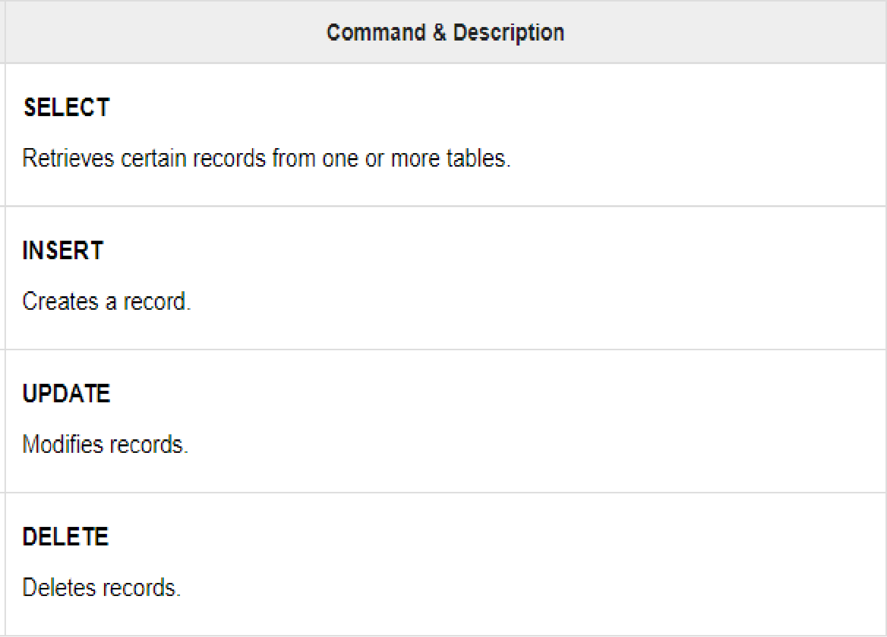
DML - Data Manipulation Language

Manipulations could be any of following:

Adding new data (Create)
Reading (Read)
Updating existing data (Update)
Delete existing data (Delete)

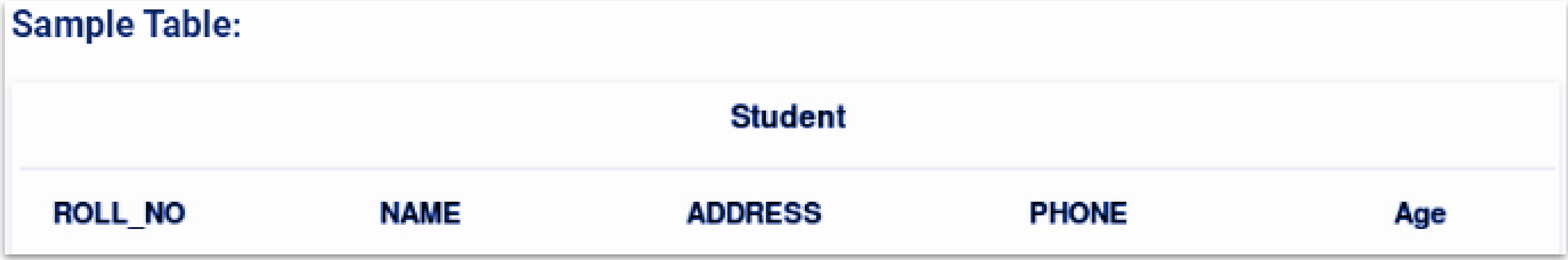
*How to remember :

CRUD Operations



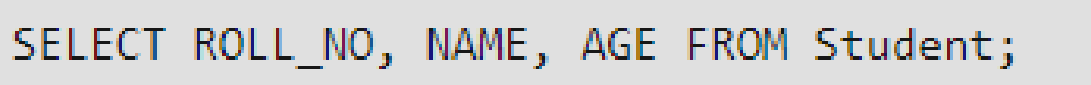
Command & Description
SELECT Retrieves certain records from one or more tables.
INSERT Creates a record.
UPDATE Modifies records.
DELETE Deletes records.

DML - Data Manipulation Language

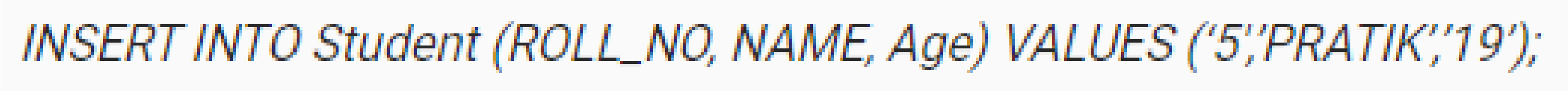


Sample Table:

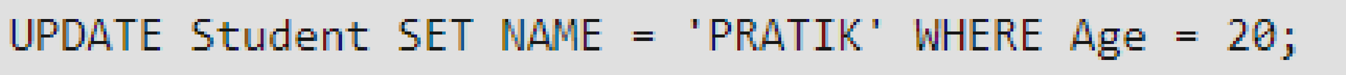
Student				
ROLL_NO	NAME	ADDRESS	PHONE	Age



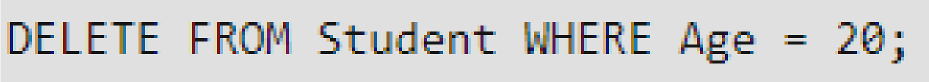
```
SELECT ROLL_NO, NAME, AGE FROM Student;
```



INSERT INTO Student (ROLL_NO, NAME, Age) VALUES (‘5’,’PRATIK’,’19’);



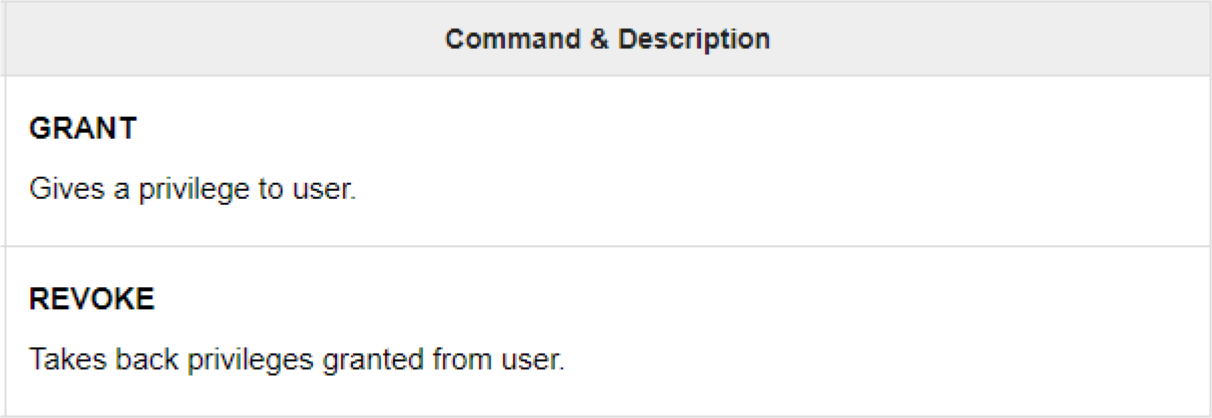
```
UPDATE Student SET NAME = 'PRATIK' WHERE Age = 20;
```



```
DELETE FROM Student WHERE Age = 20;
```

DCL - Data Control Language

Deals with the rights, permissions and other controls of the database system.



Command & Description
GRANT Gives a privilege to user.
REVOKE Takes back privileges granted from user.

Data Models

Collection of concepts that can be used to describe the structure of a database

There are different types of data models to address the level of detail

- Representational or Implementation data models
    - Used in commercial DBMSs
    - Relational Data Model is an implementation model
    - They explain data in terms of entities, relationships and attributes
- Physical Data Models
    - They explain the details of how data is stored on the computer

Relational Data Models

Three main components in conceptual data model

1. **Entity**

   A real world object or a concept

1. **Attribute**

   Property of an entity that can be used to describe an entity

1. **Relationship**

   How are two or more entities related to each other

Relational Data Model is the most widely adopted model.

Relational Data Model

Some key concepts in a relational model

A **relation** is a table with columns and rows.

An **attribute** is a named column of a relation.

A **tuple** is a row of a relation.

A **domain** is a set of allowable values for one or more attributes.

The **degree** of a relation is the number of attributes it contains.

Relational Data Model

Some key concepts in a relational model

The **cardinality** of a relation is the number of tuples it contains.

A **relational database** is a collection of normalized relations with distinct relation names.

The **intension** of a relation is the structure of the relation including its domains.

The **extension** of a relation is the set of tuples currently in the relation.