# Quiz # 01

## Task 01: Consider the following table "Students":

| RNO | NAME | FatherName | CITY | MARKS |
|---|---|---|---|---|
| 1 | Ali | Waqar | Multan | 450 |
| 2 | Kashif | Farhad | Lahore | 370 |
| 3 | Ijaz | Asif | Peshawar | 430 |

Write down SQL queries to perform the following tasks:

1. Create a new table "Courses" with the following columns: CourseID (int, auto-increment), CourseName (varchar), InstructorID (int), StartDate (date), EndDate (date), and Capacity (int).
2. Insert data into the "Courses" table for at least three courses, ensuring that the InstructorID corresponds to an existing instructor in an "Instructors" table.
3. Add a new column "Description" (varchar) to the "Courses" table with a default value of 'TBD' (To Be Determined).
4. Update the "Description" column for one of the courses to a specific description.
5. Delete the "Description" column from the "Courses" table, ensuring that the deletion does not impact any existing data or relationships.

## Task: Creation of Database for a Movie Library

Create a new database named "MovieLibraryDB".

**Create 3 tables:**

- Movies: movie_id (PK), title, director, release_year, genre.
- Actors: actor_id (PK), actor_name.
- Movie_Cast: movie_cast_id (PK), movie_id (FK - Movies), actor_id (FK - Actors), role.

**-- Create 3 tables:**

**-- Movies table**

```
CREATE TABLE IF NOT EXISTS Movies (
    movie_id INT PRIMARY KEY,
    title VARCHAR(255),
    director VARCHAR(255),
    release_year INT,
    genre VARCHAR(255)
);
-- Actors table
CREATE TABLE IF NOT EXISTS Actors (
    actor_id INT PRIMARY KEY,
    actor_name VARCHAR(255)
);
-- Movie_Cast table
CREATE TABLE IF NOT EXISTS Movie_Cast (
    movie_cast_id INT PRIMARY KEY,
    movie_id INT,
    actor_id INT,
    role VARCHAR(255),
    FOREIGN KEY (movie_id) REFERENCES Movies(movie_id),
    FOREIGN KEY (actor_id) REFERENCES Actors(actor_id)
);
```

Insert the following data into these tables:

**Movies Table:**

```
INSERT INTO Movies (movie_id, title, director, release_year, genre) VALUES (1, 'Inception', 'Christopher Nolan', 2010, 'Sci-Fi'), (2, 'The Shawshank Redemption', 'Frank Darabont', 1994, 'Drama'), (3, 'Pulp Fiction', 'Quentin Tarantino', 1994, 'Crime');
```

**Actors Table:**

```
INSERT INTO Actors (actor_id, actor_name) VALUES (1, 'Leonardo DiCaprio'), (2, 'Morgan Freeman'), (3, 'Tim Robbins'), (4, 'John Travolta');
```

**Movie_Cast Table:**

```
INSERT INTO Movie_Cast (movie_cast_id, movie_id, actor_id, role) VALUES (1, 1, 1, 'Cobb'), (2, 1, 2, 'Red'), (3, 2, 2, 'Ellis Boyd "Red" Redding'), (4, 2, 3, 'Andy Dufresne'), (5, 3, 1, 'Vincent Vega'), (6, 3, 4, 'Jules Winnfield');
```

Apply the following queries:

- **Query #1:** Retrieve all records from the "Movies" table where the genre is 'Drama'.
- **Query #2:** Retrieve all records from the "Movie_Cast" table where the role is 'Cobb'.
- **Query #3**: Retrieve the actor names and the movies they have acted in from the "Actors" and "Movie_Cast" tables using joins.
- **Query #4:** Retrieve the average release year of movies.
- **Query #5:** Retrieve the movie titles and their corresponding actors from the "Movies", "Actors", and "Movie_Cast" tables using joins.
- **Query #6:** Retrieve all actors who have not been cast in any movie yet, including details from the "Actors" and "Movie_Cast" tables using joins.
- **Query #7:** Retrieve the top 3 actors who have appeared in the most number of movies, along with the count of movies they have appeared in.
- **Query #8:** Retrieve the movie titles along with the number of actors who have appeared in each movie, sorted in descending order of the count of actors.