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 02: Creation of Database

1. Create a new database named "MusicLibraryDB".

2. Create 3 tables:
    - Songs: song_id (PK), title, artist, album, genre.

      ```
      CREATE TABLE Songs (
        song_id INT PRIMARY KEY,
        title VARCHAR(255),
        artist VARCHAR(255),
        album VARCHAR(255),
      ```

```
   genre VARCHAR(255)
);
```

- Playlists: playlist_id (PK), playlist_name.

```
CREATE TABLE Playlists (
   playlist_id INT PRIMARY KEY,
   playlist_name VARCHAR(255)
);
```

- Playlist_Songs: playlist_song_id (PK), playlist_id (FK - Playlists), song_id (FK - Songs), added_date.

```
CREATE TABLE Playlist_Songs (
   playlist_song_id INT PRIMARY KEY,
   playlist_id INT,
   song_id INT,
   added_date DATE,
   FOREIGN KEY (playlist_id) REFERENCES Playlists(playlist_id),
   FOREIGN KEY (song_id) REFERENCES Songs(song_id)
);
```

3. Insert the following data into these tables:

   - **Songs Table:**

```
INSERT INTO Songs (song_id, title, artist, album, genre)

VALUES (1, 'Shape of You', 'Ed Sheeran', '÷', 'Pop'),

(2, 'Bohemian Rhapsody', 'Queen', 'A Night at the Opera', 'Rock'),

(3, 'Hallelujah', 'Jeff Buckley', 'Grace', 'Rock');
```

   - **Playlists Table:**

```
INSERT INTO Playlists (playlist_id, playlist_name)

VALUES (1, 'Chill Vibes'),

(2, 'Rock Classics');
```

   - **Playlist_Songs Table:**

```
INSERT INTO Playlist_Songs (playlist_song_id, playlist_id, song_id, added_date)

VALUES (1, 1, 1, '2024-04-01'),
(2, 1, 3, '2024-04-01'),

(3, 2, 2, '2024-04-01'),

(4, 2, 3, '2024-03-27');
```

4. Apply the following queries:
    - **Query # 1:** Retrieve all records from the "Songs" table where the genre is 'Rock'.
    - **Query # 2:** Retrieve all records from the "Playlist_Songs" table where the added date is today.
    - **Query # 3**: Retrieve the playlist names and their total number of songs from the "Playlists" and "Playlist_Songs" tables using joins.
    - **Query # 4:** Retrieve the average number of songs per playlist using sub query.
    - **Query # 5:** Retrieve the playlist names and their corresponding songs' titles from the "Playlists", "Songs", and "Playlist_Songs" tables using joins.
    - **Query # 6:** Retrieve all songs that are not in any playlist yet, including details from the "Songs" and "Playlist_Songs" tables using joins.