```sql
-- create table 1 Users
CREATE TABLE Users (
    user_id INTEGER PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    join_date DATE,
    Limit_Time VARCHAR(20),
    subscription_type VARCHAR(50),
    Location VARCHAR(50)
);

-- insert
INSERT INTO Users (user_id, name, email, join_date, Limit_Time,
subscription_type, Location) VALUES
(1, 'Ali Khan', 'ali@example.com', '2024-01-02', '1 Month', 'Premium',
'USA'),
(2, 'Maria Gomez', 'gomez@example.com', '2024-01-10', '1 Week',
'Basic', 'Canada'),
(3, 'Ahmed Bulbul', 'bulbul@example.com', '2024-01-20', '6 Month',
'Free', 'UK'),
(4, 'Sara Ali', 'sara@example.com', '2024-02-01', '1 Year', 'Premium',
'Germany'),
(5, 'Mohammed Rahman', 'rahman@example.com', '2024-02-10', '1 Week',
'Basic', 'France'),
(6, 'Nazma Begum', 'nazma@example.com', '2024-02-15', '6 Month',
'Premium', 'Australia'),
(7, 'Minhaz Rahman', 'minhaz@example.com', '2024-02-20', '1 Month',
'Free', 'USA'),
(8, 'Shakil Ahmed', 'shakil@example.com', '2024-02-25', '6 Month',
'Premium', 'Canada'),
(9, 'Nazmul Islam', 'nazmul@example.com', '2024-03-01', '1 Month',
'Premium', 'UK'),
(10, 'Hasan Ali', 'hasan@example.com', '2024-03-10', '1 Year', 'Basic',
'Germany'),
(11, 'Masudul Islam', 'masud@example.com', '2024-03-12', '6 Month',
'Free', 'France'),
(12, 'Probal Ray', 'ray@example.com', '2024-03-15', '1 Month',
'Premium', 'Australia'),
(13, 'Elias Hossain', 'elias@example.com', '2024-03-20', '1 Week',
'Basic', 'USA'),
(14, 'Shubasish Dutta', 'dutta@example.com', '2024-03-25', '6 Month',
'Free', 'Canada'),
(15, 'Deloar Hossain', 'hossain@example.com', '2024-03-30', '1 Year',
'Premium', 'UK'),
(16, 'Ashik Chowdhury', 'ashik@example.com', '2024-04-01', '1 Month',
'Basic', 'Germany'),
(17, 'Ataur Rahman', 'ataur@example.com', '2024-04-03', '6 Month',
'Premium', 'France'),
(18, 'Shafin Ahmed', 'shafin@example.com', '2024-04-05', '1 Week',
'Free', 'Australia'),
(19, 'Hamidur Rahman', 'hamidur19@example.com', '2024-04-10', '6
Month', 'Premium', 'USA'),
(20, 'Robayet Hasan Masfi', 'masfi@example.com', '2024-04-12', '1
Year', 'Premium', 'Canada');
```

```
-- UPDATE:
UPDATE Users
SET Location = CASE
    WHEN user_id IN (1, 3, 5, 17, 19) THEN 'USA'
    WHEN user_id IN (2, 4, 6, 8, 10, 20) THEN 'Canada'
    WHEN user_id IN (7, 9, 11, 13, 15) THEN 'Australia'
    WHEN user_id IN (12, 14, 16, 18) THEN 'South Africa'
    ELSE Location
END
WHERE user_id BETWEEN 1 AND 20;



 -- Delete
DELETE FROM Users
WHERE name LIKE '%Rahman%' AND Limit_Time = '1 Week';

SELECT * FROM Users;







-- table 2 Genres
CREATE TABLE Genres (
    genre_id INTEGER PRIMARY KEY,
    genre_name VARCHAR(300)
);

INSERT INTO Genres (genre_id, genre_name) VALUES
(1, 'Action'), (2, 'Comedy'), (3, 'Drama'), (4, 'Sci-Fi'), (5,
'Thriller'),
(6, 'Romantic'), (7, 'Horror'), (8, 'Crime'), (9, 'Adventure'), (10,
'Historical'),
(11, 'Fantasy'), (12, 'Animation'), (13, 'Mystery'), (14,
'Documentary'), (15, 'Musical');


SELECT * FROM Genres;








-- table 3 Movies
CREATE TABLE Movies (
    movie_id INTEGER PRIMARY KEY,
    title VARCHAR(100),
    release_year INT,
    genre_id INT,
    duration_mins INT,
    director VARCHAR(100),
    FOREIGN KEY (genre_id) REFERENCES Genres(genre_id)
```

```sql
);

INSERT INTO Movies (movie_id, title, release_year, genre_id,
duration_mins, director) VALUES
(1, 'The Matrix', 1999, 4, 136, 'Lana Wachowski'),
(2, 'Titanic', 1997, 6, 195, 'James Cameron'),
(3, 'Avatar', 2009, 4, 162, 'James Cameron'),
(4, 'The Godfather', 1972, 3, 175, 'Francis Ford Coppola'),
(5, 'Forrest Gump', 1994, 3, 142, 'Robert Zemeckis'),
(6, 'The Avengers', 2012, 1, 143, 'Joss Whedon'),
(7, 'Iron Man', 2008, 1, 126, 'Jon Favreau'),
(8, 'Gladiator', 2000, 10, 155, 'Ridley Scott'),
(9,'Fight Club', 1997, 5, 157, 'David Fincher'),
(10,'S7VEN', 1996, 5,185, 'David Fincher'),
(11, 'The Lion King', 1994, 12, 88, 'Roger Allers'),
(12,'The Truman Show', 1998, 4,103, 'Peter Woir'),
(13, 'Black Panther', 2018, 1, 134, 'Ryan Coogler'),
(14, 'Get Out', 2017, 5, 104, 'Jordan Peele'),
(15,'Pulp Fiction', 1994, 5, 154, 'Quentin Tarantiono'),
(16,'Django UnChained', 2016, 1, 165, 'Quentin Tarantiono'),
(17,'Inglorious *****', 2009, 1, 153, 'Quentin Tarantiono'),
(18,'Frankie and Johnny', 1991, 6, 118, 'Garry Marshal'),
(19,'Memories of a murderer', 2003, 5, 232, 'Bong Joon-ho'),
(20,'Oceans Eleven', 2001, 8, 116, 'Steven Soderbergh');

ALTER TABLE Movies ADD Box_office_revenue DECIMAL(12, 2);

UPDATE Movies
SET Box_office_revenue = CASE
    WHEN movie_id = 1 THEN 838.38
    WHEN movie_id IN (4, 6, 7, 10, 11, 12, 14, 15, 16, 18) THEN 124.45
    WHEN movie_id IN (2, 5, 8, 13, 17, 19) THEN 571.56
    WHEN movie_id IN (3, 9, 20) THEN 250.55
    ELSE Box_office_revenue
END
WHERE movie_id BETWEEN 1 AND 20;



SELECT * FROM Movies;












-- Table 4 WatchHistory

CREATE TABLE WatchHistory (
    watch_id INTEGER PRIMARY KEY,
    user_id INT,
    movie_id INT,
    watch_date DATE,
    device_used VARCHAR(50),
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (movie_id) REFERENCES Movies(movie_id)
);

INSERT INTO WatchHistory (watch_id, user_id, movie_id, watch_date,
device_used) VALUES
(1, 1, 3, '2024-03-15', 'Mobile'),
(2, 2, 5, '2024-03-17', 'Laptop'),
(3, 3, 6, '2024-03-20', 'Tablet'),
(4, 4, 7, '2024-03-21', 'Laptop'),
(5, 13, 8, '2024-03-22', 'Mobile'),
(6, 6, 9, '2024-03-23', 'Laptop'),
(7, 7, 10, '2024-03-25', 'Tablet'),
(8, 8, 11, '2024-03-26', 'Laptop'),
(9, 9, 12, '2024-03-27', 'Mobile'),
(10, 10, 13, '2024-03-29', 'Laptop'),
(11, 11, 14, '2024-03-30', 'Tablet'),
(12, 12, 15, '2024-04-01', 'Mobile'),
(13, 13, 16, '2024-04-02', 'Laptop'),
(14, 14, 17, '2024-04-03', 'Tablet'),
(15, 15, 18, '2024-04-05', 'Laptop'),
(16, 16, 19, '2024-04-06', 'Mobile'),
(17, 17, 20, '2024-04-07', 'Laptop'),
(18, 18, 1, '2024-04-08', 'Tablet'),
(19, 19, 2, '2024-04-09', 'Laptop'),
(20, 20, 4, '2024-04-10', 'Mobile');



SELECT * FROM WatchHistory;









-- Table 5 Ratings
CREATE TABLE Ratings (
    rating_id INTEGER PRIMARY KEY,
    user_id INT,
    movie_id INT,
    rating INT CHECK (rating BETWEEN 1 AND 5),
    review TEXT,
    review_date DATE,
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (movie_id) REFERENCES Movies(movie_id)
);

INSERT INTO Ratings (rating_id, user_id, movie_id, rating, review,
review_date) VALUES
(1, 1, 3, 5, 'Amazing visuals!', '2024-03-15'),
(2, 2, 5, 4, 'Heartwarming story.', '2024-03-17'),
(3, 3, 6, 3, 'Could be better.', '2024-03-20'),
(4, 4, 7, 5, 'Masterpiece.', '2024-03-21'),
(5, 13, 8, 2, 'Not impressed.', '2024-03-22'),
(6, 6, 9, 5, 'Family favorite!', '2024-03-23'),
(7, 7, 10, 4, 'Emotional.', '2024-03-25'),
(8, 8, 11, 5, 'Classic Disney.', '2024-03-26'),
(9, 9, 12, 3, 'Too musical.', '2024-03-27'),
(10, 10, 13, 4, 'Great representation.', '2024-03-29'),
(11, 11, 14, 5, 'Chilling and smart.', '2024-03-30'),
(12, 12, 15, 3, 'Weird but cool.', '2024-04-01'),
(13, 13, 16, 5, 'Beautiful visuals.', '2024-04-02'),
(14, 14, 17, 4, 'Touching story.', '2024-04-03'),
(15, 15, 18, 4, 'Well animated.', '2024-04-05'),
(16, 16, 19, 2, 'Not my type.', '2024-04-06'),
(17, 17, 20, 5, 'Loved every second.', '2024-04-07'),
(18, 18, 1, 4, 'Neo is iconic!', '2024-04-08'),
(19, 19, 2, 5, 'Best love story.', '2024-04-09'),
(20, 20, 4, 5, 'Legendary film.', '2024-04-10');



-- Update


UPDATE Ratings
SET review = 'Truly a visual masterpiece.', rating = 5
WHERE user_id = 18 AND movie_id = 1;


UPDATE Ratings
SET review = 'Total suspense, not recommend for week hearted.', rating
= 5
WHERE user_id = 6 AND movie_id = 9;



UPDATE Ratings
SET review = 'One of the romantic films I have seen!', rating = 5
WHERE user_id = 20 OR movie_id = 18;

SELECT * FROM Ratings;




-- Easy Queries

-- 1) 1. Fetch all users who have a 'Premium' subscription and whose
name contains 'Ali
 SELECT * FROM Users
WHERE subscription_type = 'Premium' AND name LIKE '%Ali%' ;

-- 2) count all genres
 SELECT COUNT(*) FROM Genres;



-- 3) fetch all movies of Quentin tarantino before the year 2000
```

```
 SELECT * FROM Movies WHERE release_year>2000 AND director = 'Quentin
Tarantiono';


 -- 4. Fetch the rating of movie_id 3
SELECT rating FROM Ratings WHERE movie_id = 3;


-- 5) fetch Users name who joined after 20 th february 2024 and locate
in USA
SELECT name FROM Users WHERE  join_date> '2024-02-20' AND Location =
'USA' ;


--  6) Delete a genre
DELETE FROM Genres WHERE genre_id = 15;
SELECT * FROM Genres;


-- 7) Update the matrix movie duration to 150 minute
UPDATE Movies
SET duration_mins = 150
WHERE title = 'The Matrix';
 SELECT * FROM Movies;


-- 8) update the device to smart tv for id 5
UPDATE WatchHistory
SET device_used = 'Smart TV'
WHERE watch_id = 5;
SELECT * FROM WatchHistory;






-- Moderate queries


-- 1) fetch all movie titles user-id 9 has watched
SELECT title FROM Movies WHERE movie_id IN (SELECT movie_id FROM
WatchHistory WHERE user_id = 9);






-- 2) fetch movie title which duration are over 150 minute and over 500
million Box_office_revenue
SELECT title, duration_mins, Box_office_revenue
FROM Movies
WHERE duration_mins > 150 AND Box_office_revenue > 500;
```

```
-- 3)Cross product between title of movies and rating of Ratings table
of both's movie id 9 and 10
SELECT Movies.title, Ratings.rating
FROM Movies
CROSS JOIN Ratings
WHERE Movies.movie_id IN (9, 10) AND Ratings.movie_id IN (9, 10) ;
```

```
 -- 4) Fetch the movie titles watched by user ID 1
SELECT m.title
FROM Movies m, WatchHistory wh
WHERE m.movie_id = wh.movie_id
AND wh.user_id = 1;
```

```
-- 5) Fetch all users who have rated "The Matrix" higher than 4
SELECT name
FROM Users
WHERE user_id IN (
    SELECT user_id FROM Ratings WHERE movie_id = 1 AND rating > 4
);
```

```
-- 6)Update user location in 3 countries USA, AUs and south africa
according to their user-id
UPDATE Users
SET Location = CASE
    WHEN user_id IN (1, 3, 5, 17, 19) THEN 'USA'
    WHEN user_id IN (2, 4, 6, 8, 10, 20) THEN 'Canada'
    WHEN user_id IN (7, 9, 11, 13, 15) THEN 'Australia'
    WHEN user_id IN (12, 14, 16, 18) THEN 'South Africa'
    ELSE Location
END
WHERE user_id BETWEEN 1 AND 20;
SELECT name, Location FROM Users;
```

```
-- 7) find name where contains ma
SELECT genre_id, genre_name
FROM Genres
WHERE genre_name LIKE '%ma%';
```

```
-- 8) List watches with movie titles
SELECT wh.watch_date, m.title, wh.device_used
FROM WatchHistory wh, Movies m
WHERE wh.movie_id = m.movie_id;
```

```sql
-- Advanced queries

-- 1) List of Users Who Joined After the First User from the USA
SELECT * FROM Users
WHERE join_date > (
    SELECT join_date
    FROM Users
    WHERE Location = 'USA'
    ORDER BY join_date ASC
    LIMIT 1
)
ORDER BY join_date ASC;







-- 2) Users with the same Limit_Time as any Free user
SELECT name FROM Users
WHERE Limit_Time IN (
    SELECT Limit_Time FROM Users WHERE subscription_type = 'Free'
);




-- 3) Genres that are linked to at least one movie directed by someone
who directed â□□Titanicâ□□
SELECT * FROM Genres
WHERE genre_id IN (
    SELECT DISTINCT genre_id FROM Movies
    WHERE director = (
        SELECT director FROM Movies WHERE title = 'Titanic'
    )
);


-- 4) Movie that has highest duration
SELECT title  FROM Movies
WHERE release_year = (
    SELECT release_year FROM Movies
    ORDER BY duration_mins DESC LIMIT 1
);



-- 5) Ratings for movies that the user has also watched on Mobile
SELECT rating_id, review_date FROM Ratings
WHERE user_id IN (
    SELECT user_id FROM WatchHistory WHERE device_used = 'Mobile'
);
```

```
-- 6) List of Movie Genres and Their Movies Released Between 1990 and
2020
SELECT g.genre_name, m.movie_id AS total_movies
FROM Genres g, Movies m
WHERE g.genre_id = m.genre_id
AND m.release_year BETWEEN 1990 AND 2020;
```

```
-- 7)  Find Users Who Watched a Movie But Never Rated It
SELECT DISTINCT u.name
FROM Users u
JOIN WatchHistory w ON u.user_id = w.user_id
LEFT JOIN Ratings r ON w.user_id = r.user_id AND w.movie_id =
r.movie_id
WHERE r.rating IS NULL;
SELECT name FROM Users;
```

```
-- 8) Find all movies directed by someone whose name contains 'James'
and have at least one rating
SELECT DISTINCT m.title
FROM Movies m, Ratings r
WHERE m.movie_id = r.movie_id
  AND m.director LIKE '%James%';
```