**SQL Scripts**

1. **Tables:**

CREATE TABLE Users (

UserID NUMBER PRIMARY KEY,

Username VARCHAR2(50),

Email VARCHAR2(100),

Password VARCHAR2(50),

SubscriptionStatus VARCHAR2(20)

);



CREATE TABLE Content (

ContentID NUMBER PRIMARY KEY,

Title VARCHAR2(100),

Genre VARCHAR2(50),

Rating NUMBER(3,1)

);



CREATE TABLE Watchlist (

WatchlistID NUMBER PRIMARY KEY,

UserID NUMBER,

ContentID NUMBER,

FOREIGN KEY (UserID) REFERENCES Users(UserID),

FOREIGN KEY (ContentID) REFERENCES Content(ContentID)

);



CREATE TABLE Subscriptions (

SubscriptionID NUMBER PRIMARY KEY,

UserID NUMBER,

PlanName VARCHAR2(50),

StartDate DATE,

EndDate DATE,

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);



CREATE TABLE SubscriptionPlans (

PlanID NUMBER PRIMARY KEY,

PlanName VARCHAR2(50),

Price NUMBER(10, 2),

DurationMonths NUMBER,

Description VARCHAR2(255)

);



1. **Data:**

INSERT INTO Users VALUES (1, 'saif\_m', 'saif@example.com', 'pass123', 'Active');

INSERT INTO Users VALUES (2, 'maria\_m', 'maria@example.com', 'maria2024', 'Active');

INSERT INTO Users VALUES (3, 'karim\_s', 'karim@example.com', 'karimpass', 'Active');

INSERT INTO Users VALUES (4, 'michael\_m', 'michael@example.com', 'mic1234', 'not Active');

INSERT INTO Users VALUES (5, 'alex\_p', 'alex@example.com', 'alex1234', 'Active');

INSERT INTO Users VALUES (6, 'jerry\_t', 'jerry@example.com', 'passjerry', 'Active');

INSERT INTO Content VALUES (1, 'Avengers: Endgame', 'Action', 9.0);

INSERT INTO Content VALUES (2, ' How I Met Your Mother', 'Comedy', 8.5);

INSERT INTO Content VALUES (3, 'YOU', 'Drama', 7.8);

INSERT INTO Content VALUES (4, 'Grown Ups', 'Comedy', 6.0);

INSERT INTO Content VALUES (5, ' Breaking Bad', 'Drama', 9.5);

INSERT INTO Content VALUES (6, 'Welad Rizk', 'Action', 6.7);

INSERT INTO Content VALUES (7, 'The Whale', 'Drama', 7.6);

INSERT INTO Content VALUES (8, 'Al Atawla', 'Drama', 6.3);

INSERT INTO Content VALUES (9, 'Free Guy', 'Comedy', 7.1);

INSERT INTO Content VALUES (10, 'Deadpool', 'Comedy', 8.0);

INSERT INTO Content VALUES (11, 'The Crow', 'Action', 4.7);

INSERT INTO Content VALUES (12, 'John Wick', 'Action', 7.5);

INSERT INTO Watchlist VALUES (1, 1, 1);

INSERT INTO Watchlist VALUES (2, 2, 2);

INSERT INTO Watchlist VALUES (3, 3, 3);  
INSERT INTO Watchlist VALUES (4, 3, 2);

INSERT INTO Watchlist VALUES (5, 4, 7);

INSERT INTO Watchlist VALUES (6, 5, 9);

INSERT INTO Watchlist VALUES (7, 6, 10);

INSERT INTO Watchlist VALUES (8, 2, 5);

INSERT INTO Watchlist VALUES (9, 4, 2);

INSERT INTO Watchlist VALUES (10, 1, 11);

INSERT INTO Watchlist VALUES (11, 5, 3);

INSERT INTO Watchlist VALUES (12, 6, 2);

INSERT INTO Watchlist VALUES (13, 2, 12);

INSERT INTO Watchlist VALUES (14, 4, 6);

INSERT INTO Watchlist VALUES (15, 3, 5);

INSERT INTO SubscriptionPlans VALUES (1, 'Basic', 9.99, 1, 'Basic plan with limited features');

INSERT INTO SubscriptionPlans VALUES (2, 'Premium', 19.99, 1, 'Premium plan with all features');

INSERT INTO SubscriptionPlans VALUES (3, 'Family', 29.99, 3, 'Family plan with access for up to 4 users');

ALTER TABLE Subscriptions

ADD PlanID NUMBER;

ALTER TABLE Subscriptions

ADD CONSTRAINT fk\_plan\_id FOREIGN KEY (PlanID) REFERENCES SubscriptionPlans(PlanID);

INSERT INTO subscriptions VALUES (subscriptions\_seq.NEXTVAL, 1,'Premium' ,TO\_DATE('2025-04-28', 'YYYY-MM-DD'), TO\_DATE('2026-04-28', 'YYYY-MM-DD'),2);

INSERT INTO subscriptions VALUES (subscriptions\_seq.NEXTVAL, 2,'Premium' ,TO\_DATE('2025-04-28', 'YYYY-MM-DD'), TO\_DATE('2026-04-28', 'YYYY-MM-DD'),2);INSERT INTO subscriptions VALUES (subscriptions\_seq.NEXTVAL, 3,'Basic' ,TO\_DATE('2025-04-28', 'YYYY-MM-DD'), TO\_DATE('2026-04-28', 'YYYY-MM-DD'),1);

INSERT INTO subscriptions VALUES (subscriptions\_seq.NEXTVAL, 5,'Family' ,TO\_DATE('2025-04-28', 'YYYY-MM-DD'), TO\_DATE('2026-04-28', 'YYYY-MM-DD'),3);

INSERT INTO subscriptions VALUES (subscriptions\_seq.NEXTVAL, 6,'Family' ,TO\_DATE('2025-04-28', 'YYYY-MM-DD'), TO\_DATE('2026-04-28', 'YYYY-MM-DD'),3);

1. **Procedures:**

CREATE OR REPLACE PROCEDURE UpdateContentRating(

p\_ContentID IN NUMBER,

p\_NewRating IN NUMBER

)

IS

BEGIN

UPDATE Content

SET Rating = p\_NewRating

WHERE ContentID = p\_ContentID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Content rating updated successfully!');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No content found with the specified ID.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END;



create or replace PROCEDURE SubscribeUser (

p\_UserID IN NUMBER,

p\_PlanID IN NUMBER

) IS

v\_Price NUMBER(10, 2);

v\_DurationMonths NUMBER;

v\_PlanName VARCHAR2(100);

v\_StartDate DATE := SYSDATE;

v\_EndDate DATE;

BEGIN

BEGIN

SELECT Price, DurationMonths, PlanName

INTO v\_Price, v\_DurationMonths, v\_PlanName

FROM SubscriptionPlans

WHERE PlanID = p\_PlanID;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error: PlanID ' || p\_PlanID || ' does not exist.');

RETURN;

END;

v\_EndDate := ADD\_MONTHS(v\_StartDate, v\_DurationMonths);

BEGIN

INSERT INTO Subscriptions (SubscriptionID, UserID, PlanID, PlanName, StartDate, EndDate)

VALUES (Subscriptions\_SEQ.NEXTVAL, p\_UserID, p\_PlanID, v\_PlanName, v\_StartDate, v\_EndDate);

UPDATE Users

SET SubscriptionStatus = 'Active'

WHERE UserID = p\_UserID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Subscription for user ' || p\_UserID || ' to plan "' || v\_PlanName || '" has been successful.');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error occurred while processing subscription for user ' || p\_UserID);

END;

END SubscribeUser;



CREATE OR REPLACE PROCEDURE GetContentData (

input IN VARCHAR2,

selected\_genre IN VARCHAR2,

select\_result OUT SYS\_REFCURSOR

) AS

BEGIN

IF input = 'genres' THEN

OPEN select\_result FOR

SELECT DISTINCT Genre FROM Content;

ELSIF input = 'titles' THEN

IF selected\_genre IS NULL OR selected\_genre = 'All' THEN

OPEN select\_result FOR

SELECT Title FROM Content;

ELSE

OPEN select\_result FOR

SELECT Title FROM Content WHERE Genre = selected\_genre;

END IF;

END IF;

END;



CREATE OR REPLACE PROCEDURE GetShowRating (

selected\_title IN VARCHAR2,

result\_rating OUT NUMBER

)

IS

BEGIN

SELECT rating

INTO result\_rating

FROM Content

WHERE title = selected\_title;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

result\_rating := NULL;

WHEN OTHERS THEN

result\_rating := NULL;

END;

1. **Sequences:**

CREATE SEQUENCE Subscriptions\_SEQ

START WITH 1

INCREMENT BY 1

NOCACHE;



CREATE SEQUENCE user\_seq START WITH 7 INCREMENT BY 1;

