EXERCISE 8: VIEWS, SYNONYMS & SEQUENCES

Name: Mahesh Jagtap
Reg No. 24MCS1017
Date: 27.09.2024

Create the following tables:
Movie (mID, title, release_year, director)
Reviewer (rID, rname)
Rating (rID, mID, stars, rDate)
Description: reviewer rID, movie mID, a number of stars rating (1-5) and rating Date rDate.

```
CREATE TABLE Movie (
  mID NUMBER PRIMARY KEY,
  title VARCHAR2(24) NOT NULL,
  release_year NUMBER(4),
  director VARCHAR2(24)
);
```

```
SQL> CREATE TABLE Movie (
  2      mID NUMBER PRIMARY KEY,
  3      title VARCHAR2(24) NOT NULL,
  4      release_year NUMBER(4),
  5      director VARCHAR2(24)
  6  );

Table created.
```

```
CREATE TABLE Reviewer (
  rID NUMBER PRIMARY KEY,
  rname VARCHAR2(255) NOT NULL
);
```

```
SQL> CREATE TABLE Reviewer (
  2      rID NUMBER PRIMARY KEY,
  3      rname VARCHAR2(255) NOT NULL
  4  );

Table created.
```

```
CREATE TABLE Rating (
  rID NUMBER,
  mID NUMBER,
  stars NUMBER(1) CHECK (stars BETWEEN 1 AND 5),
  rDate DATE,
  PRIMARY KEY (rID, mID),
  FOREIGN KEY (rID) REFERENCES Reviewer(rID),
```

FOREIGN KEY (mID) REFERENCES Movie(mID)
);

```
SQL> CREATE TABLE Rating (
  2      rID NUMBER,
  3      mID NUMBER,
  4      stars NUMBER(1) CHECK (stars BETWEEN 1 AND 5),
  5      rDate DATE,
  6      PRIMARY KEY (rID, mID),
  7      FOREIGN KEY (rID) REFERENCES Reviewer(rID),
  8      FOREIGN KEY (mID) REFERENCES Movie(mID)
  9  );

Table created.
```

-- Insert sample data into Movie table
INSERT INTO Movie VALUES (101, 'Inception', 2010, 'Christopher Nolan');
INSERT INTO Movie VALUES (102, 'The Matrix', 1999, 'Roman reigns');
INSERT INTO Movie VALUES (103, 'Interstellar', 2014, 'Rock');

-- Insert sample data into Reviewer table
INSERT INTO Reviewer VALUES (1, 'Ronaldo');
INSERT INTO Reviewer VALUES (2, 'Messi');
INSERT INTO Reviewer VALUES (3, 'John cena');

-- Insert sample data into Rating table
INSERT INTO Rating VALUES (1, 101, 5, TO_DATE('2023-09-01', 'YYYY-MM-DD'));
INSERT INTO Rating VALUES (2, 102, 4, TO_DATE('2023-09-02', 'YYYY-MM-DD'));
INSERT INTO Rating VALUES (3, 103, 5, TO_DATE('2023-09-03', 'YYYY-MM-DD'));
INSERT INTO Rating VALUES (1, 102, 3, TO_DATE('2023-09-04', 'YYYY-MM-DD'));
INSERT INTO Rating VALUES (2, 103, 4, TO_DATE('2023-09-05', 'YYYY-MM-DD'));
INSERT INTO Rating VALUES (3, 101, 5, TO_DATE('2023-09-06', 'YYYY-MM-DD'));

Exercise on Create tables from Existing Tables (Sub-tables)

1. Create any new table from the existing table (MOVIE) with all attributes

```
CREATE TABLE Movie_Copy AS
SELECT *
FROM Movie;
```

```
SQL> CREATE TABLE Movie_Copy AS
  2  SELECT *
  3  FROM Movie;

Table created.
```

2. Create any new table from the existing table (MOVIE) with two attributes

```
CREATE TABLE Movie_Two_Attributes AS
SELECT mID, title
FROM Movie;
```

```
SQL> CREATE TABLE Movie_Two_Attributes AS
  2  SELECT mID, title
  3  FROM Movie;

Table created.
```

3. Create a new table from the existing table (MOVIE) with all attributes and the directors name starts with ‘R’.

```
CREATE TABLE Movie_Director_R AS
SELECT *
FROM Movie
WHERE director LIKE 'R%';
```

```
SQL> CREATE TABLE Movie_Director_R AS
  2  SELECT *
  3  FROM Movie
  4  WHERE director LIKE 'R%';

Table created.

SQL>
```

Exercise on Views

4. Create a View called **LateRating** which contains movie ratings after January 20, 2011. The view contains the movie ID, movie title, number of stars, and rating date.

```
CREATE VIEW LateRating AS
SELECT m.mID, m.title, r.stars, r.rDate
FROM Movie m
JOIN Rating r ON m.mID = r.mID
WHERE r.rDate > TO_DATE('2011-01-20', 'YYYY-MM-DD');
```

```
SQL> CREATE VIEW LateRating AS
  2  SELECT m.mID, m.title, r.stars, r.rDate
  3  FROM Movie m
  4  JOIN Rating r ON m.mID = r.mID
  5  WHERE r.rDate > TO_DATE('2011-01-20', 'YYYY-MM-DD');

View created.

SQL> select * from LateRating;

       MID TITLE                     STARS RDATE
---------- ---------------------- ---------- ---------
       101 Inception                       5 01-SEP-23
       102 The Matrix                      4 02-SEP-23
       103 Interstellar                    5 03-SEP-23
       102 The Matrix                      3 04-SEP-23
       103 Interstellar                    4 05-SEP-23
       101 Inception                       5 06-SEP-23

6 rows selected.
```

5. Create a View **HighRating** which contains movies with rating above 3 stars. The view contains the movie ID and movie title.

```
CREATE VIEW HighRating AS
SELECT m.mID, m.title
FROM Movie m
JOIN Rating r ON m.mID = r.mID
WHERE r.stars > 3;
```

```
SQL> CREATE VIEW HighRating AS
  2  SELECT m.mID, m.title
  3  FROM Movie m
  4  JOIN Rating r ON m.mID = r.mID
  5  WHERE r.stars > 3;

View created.

SQL> select * from HighRating;

       MID TITLE
---------- ------------------------
       101 Inception
       102 The Matrix
       103 Interstellar
       103 Interstellar
       101 Inception
```

6. Create a View **NoRating** which contains movies with no ratings. The view contains the movie ID and movie title.

```
CREATE VIEW NoRating AS
SELECT m.mID, m.title
FROM Movie m
LEFT JOIN Rating r ON m.mID = r.mID
WHERE r.mID IS NULL;
```

```
SQL> CREATE VIEW NoRating AS
  2  SELECT m.mID, m.title
  3  FROM Movie m
  4  LEFT JOIN Rating r ON m.mID = r.mID
  5  WHERE r.mID IS NULL;

View created.
```

7. Display all the views generated.

```
SELECT view_name
FROM user_views;
```

```
VIEW_NAME
--------------------------------
MVIEW_RECOMMENDATIONS
MVIEW_WORKLOAD
NORATING
PRODUCT_PRIVS
```

8. Execute UPDATE/DELETE commands on the view created.

```
UPDATE Rating r
SET stars = 5
WHERE r.mID IN (SELECT mID FROM HighRating);
```

```
SQL> UPDATE Rating r
  2  SET stars = 5
  3  WHERE r.mID IN (SELECT mID FROM HighRating);

6 rows updated.

SQL> SELECT mID, stars
  2  FROM Rating
  3  WHERE stars = 5;

       MID      STARS
---------- ----------
       101          5
       102          5
       103          5
       102          5
       103          5
       101          5

6 rows selected.
```

DELETE FROM Movie
WHERE mID IN (SELECT mID FROM NoRating);

```
SQL> DELETE FROM Movie
  2  WHERE mID IN (SELECT mID FROM NoRating);

0 rows deleted.

SQL> SELECT mID, title
  2  FROM Movie;

       MID TITLE
---------- ------------------------
       101 Inception
       102 The Matrix
       103 Interstellar
```

9. Drop any view.

DROP VIEW LateRating;

```
SQL> DROP VIEW LateRating;

View dropped.
```

Exercise on Synonyms

10. Create a synonym for any table.

```
SQL> CREATE SYNONYM mv FOR Movie;

Synonym created.
```

11. Drop the synonym.

DROP SYNONYM mv;

```
SQL> DROP SYNONYM mv;

Synonym dropped.
```

Exercises on Sequence

12. Create a sequence named seq1 start with min value 1 and max value 100.

CREATE SEQUENCE seq1
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 100
CYCLE;

```
SQL> CREATE SEQUENCE seq1
  2    START WITH 1
  3    INCREMENT BY 1
  4    MINVALUE 1
  5    MAXVALUE 100
  6    CYCLE;

Sequence created.
```

13. Connect the sequence with any table and display the content with sequence no.

```
CREATE TABLE Movie_Records (
   ID NUMBER PRIMARY KEY,
   Title VARCHAR2(100),
   Release_Year NUMBER
);
INSERT INTO Movie_Records (ID, Title, Release_Year) VALUES (seq1.NEXTVAL,
'Inception', 2010);
INSERT INTO Movie_Records (ID, Title, Release_Year) VALUES (seq1.NEXTVAL,
'The Matrix', 1999);
INSERT INTO Movie_Records (ID, Title, Release_Year) VALUES (seq1.NEXTVAL,
'Interstellar', 2014);
SELECT ID, Title, Release_Year FROM Movie_Records;
```

```
SQL> SELECT ID, Title, Release_Year FROM Movie_Records;

        ID TITLE                RELEASE_YEAR
---------- -------------------- ------------
         1 Inception                    2010
         2 The Matrix                   1999
         3 Interstellar                 2014
```

14. Create a sequence named seq2 start with min value 14 and max value 30 for the MID in the Movie table.

```
CREATE SEQUENCE seq2
 START WITH 14
 INCREMENT BY 1
 MINVALUE 14
 MAXVALUE 30
 NOCYCLE;
```

```
SQL> CREATE SEQUENCE seq2
  2    START WITH 14
  3    INCREMENT BY 1
  4    MINVALUE 14
  5    MAXVALUE 30
  6    NOCYCLE;

Sequence created.
```

15. Drop the sequence seq1.

DROP SEQUENCE seq1;

```
SQL> DROP SEQUENCE seq1;

Sequence dropped.
```