EXERCISE 12: VIEWS, SYNONYMS & SEQUENCE

Faculty Name: Dr. Balasundaram A
Slot: L43+L44
Class Number: CH2021221000708
Date: 10/11/2021

DBMS LAB ASSIGNMENT 12 MEHER SHRISHTI NIGAM 20BRS1193

VIEWS, SYNONYMS & SEQUENCE

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_20BRS1193(mID number, title varchar2(60),
release_year varchar2(10),
director varchar2(30), CONSTRAINT Pkey_movie PRIMARY KEY(mID));
CREATE TABLE REVIEWER_20BRS1193(rID number, rname varchar2(50),
CONSTRAINT Pkey_review PRIMARY KEY(rID));
CREATE TABLE RATING_20BRS1193(rID number, mID number, stars number,
rDate varchar2(30), CONSTRAINT Pkey_rating PRIMARY KEY(rID,mID));
```

```
SQL> CREATE TABLE MOVIE_20BRS1193(mID number, title varchar2(60), release_year varchar2(10),
 director varchar2(30), CONSTRAINT Pkey_movie PRIMARY KEY(mID));

Table created.

SQL> CREATE TABLE REVIEWER_20BRS1193(rID number, rname varchar2(50), CONSTRAINT Pkey_review
PRIMARY KEY(rID));

Table created.

SQL> CREATE TABLE RATING_20BRS1193(rID number, mID number, stars number, rDate varchar2(30),
 CONSTRAINT Pkey_rating PRIMARY KEY(rID,mID));

Table created.

SQL>
```

```
SQL>
SQL> INSERT ALL
  2      INTO MOVIE_20BRS1193 VALUES(1, 'Home Alone', '1990', 'Chris Columbus')
  3      INTO MOVIE_20BRS1193 VALUES(2, 'Men in Black', '1997', 'Barry Sonnenfeld')
  4      INTO MOVIE_20BRS1193 VALUES(3, 'Mr. Beans Holiday', '2007', 'Steve Bendelack')
  5      INTO MOVIE_20BRS1193 VALUES(4, 'Cars', '2006', 'John Lasseter')
  6      INTO MOVIE_20BRS1193 VALUES(5, 'Venom', '2018', 'Ruben Fleischer')
  7      INTO MOVIE_20BRS1193 VALUES(6, 'Bad Boys', '1995', 'Michael Bay')
  8  SELECT * FROM DUAL;

6 rows created.

SQL>
SQL> INSERT ALL
  2      INTO REVIEWER_20BRS1193 VALUES(1, 'Sophie Kane')
  3      INTO REVIEWER_20BRS1193 VALUES(2, 'Liam Pate')
  4      INTO REVIEWER_20BRS1193 VALUES(3, 'Harley Wynn')
  5  SELECT * FROM DUAL;

3 rows created.

SQL>
SQL> INSERT ALL
  2      INTO RATING_20BRS1193 VALUES(1, 2, 0, '4-Dec-2010')
  3      INTO RATING_20BRS1193 VALUES(3, 1, 3, '1-Dec-2009')
  4      INTO RATING_20BRS1193 VALUES(2, 2, 4, '4-Dec-2011')
  5      INTO RATING_20BRS1193 VALUES(2, 3, 1, '10-Jan-2021')
  6      INTO RATING_20BRS1193 VALUES(1, 5, 4, '4-Feb-2017')
  7      INTO RATING_20BRS1193 VALUES(3, 4, 2, '16-Dec-2011')
  8      INTO RATING_20BRS1193 VALUES(1, 1, 5, '24-Mar-2008')
  9      INTO RATING_20BRS1193 VALUES(2, 4, 0, '31-Apr-2013')
 10  SELECT * FROM DUAL;

8 rows created.
```

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

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

```
CREATE TABLE ALL_MOVIES AS (select * from MOVIE_20BRS1193);
select * from ALL_MOVIES;
```

```
SQL> CREATE TABLE ALL_MOVIES AS (select * from MOVIE_20BRS1193);

Table created.

SQL> select * from ALL_MOVIES;

       MID TITLE
---------- --------------------------------------------------------------
RELEASE_YE DIRECTOR
---------- ------------------------------
         1 Home Alone
1990       Chris Columbus

         2 Men in Black
1997       Barry Sonnenfeld

         3 Mr. Beans Holiday
2007       Steve Bendelack


       MID TITLE
---------- --------------------------------------------------------------
RELEASE_YE DIRECTOR
---------- ------------------------------
         4 Cars
2006       John Lasseter

         5 Venom
2018       Ruben Fleischer

         6 Bad Boys
1995       Michael Bay


6 rows selected.
```

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

```
CREATE TABLE MOVIE_TITLE AS (select mID , title from
MOVIE_20BRS1193);
select * from MOVIE_TITLE;
```

```
SQL>
SQL> CREATE TABLE MOVIE_TITLE AS (select mID , title from MOVIE_20BRS1193);

Table created.

SQL> select * from MOVIE_TITLE;

       MID TITLE
---------- --------------------------------------------------------------
         1 Home Alone
         2 Men in Black
         3 Mr. Beans Holiday
         4 Cars
         5 Venom
         6 Bad Boys

6 rows selected.
```

3. Create any new table from existing table (MOVIE) with all attributes and the directors name starts with 'M'.

```
CREATE TABLE M_DIRECTOR AS (select * from MOVIE_20BRS1193 where director like 'M%');
select * from M_DIRECTOR;
```

```
SQL>
SQL> CREATE TABLE M_DIRECTOR AS (select * from MOVIE_20BRS1193 where director like 'M%');

Table created.

SQL> select * from M_DIRECTOR;

       MID TITLE
---------- --------------------------------------------------------------
RELEASE_YE DIRECTOR
---------- ------------------------------
         6 Bad Boys
1995       Michael Bay


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 RATING_20BRS1193 R Inner Join MOVIE_20BRS1193 M on R.mID = M.mID where R.rDate > '20-JAN-2011' );
```

```
SQL> CREATE VIEW LateRating AS (select M.mID, M.title, R.stars, R.rDate from RATING_20BRS1193 R Inner Join
MOVIE_20BRS1193 M on R.mID = M.mID where R.rDate > '20-JAN-2011' );

View created.
```

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 RATING_20BRS1193 R Inner Join MOVIE_20BRS1193 M on R.mID = M.mID where stars > 3);
```

```
SQL>
SQL> CREATE VIEW HighRating AS (select M.mID,M.title from RATING_20BRS1193 R Inner Join MOVIE_20BRS1193 M
on R.mID = M.mID where stars > 3);

View created.
```

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.title,r.mID from RATING_20BRS1193
R Inner Join MOVIE_20BRS1193 M on R.mID = M.mID where stars=0);
```

```
SQL>
SQL> CREATE VIEW NoRating AS (select m.title,r.mID from RATING_20BRS1193 R Inner Join MOVIE_20BRS1193 M on
 R.mID = M.mID where stars=0);

View created.
```

7. Display all the views generated.

```
select * from LateRating;
select * from HighRating;
select * from NoRating;
```

```
SQL>
SQL> select * from LateRating;

       MID TITLE
---------- --------------------------------------------------------------
     STARS RDATE
---------- -----------------------------
         1 Home Alone
         5 24-Mar-2008

         2 Men in Black
         0 4-Dec-2010

         2 Men in Black
         4 4-Dec-2011


       MID TITLE
---------- --------------------------------------------------------------
     STARS RDATE
---------- -----------------------------
         4 Cars
         0 31-Apr-2013

         5 Venom
         4 4-Feb-2017


SQL> select * from HighRating;

       MID TITLE
---------- --------------------------------------------------------------
         2 Men in Black
         5 Venom
         1 Home Alone

SQL> select * from NoRating;

TITLE                                                              MID
------------------------------------------------------------ ----------
Men in Black                                                        2
Cars                                                                4

SQL>
```

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

```
DELETE HighRating where title = 'Home Alone';
```

```
SQL> DELETE HighRating where title = 'Home Alone';

1 row deleted.

SQL>
```

9. Drop any view.

```
DROP VIEW HighRating;
```

```
SQL> DROP VIEW HighRating;

View dropped.
```

Exercise on Synonyms

10. Create a synonym for any table.

```
CREATE SYNONYM good_movies for MOVIE_20BRS1193;
```

```
SQL> CREATE SYNONYM good_movies for MOVIE_20BRS1193;

Synonym created.
```

11. Drop the synonym.

```
DROP SYNONYM good_movies;
```

```
SQL>
SQL> DROP SYNONYM good_movies;

Synonym dropped.
```

Exercises on Sequence

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

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

```
SQL> CREATE SEQUENCE seq1 INCREMENT BY 1 START WITH 1 MAXVALUE 100 MINVALUE 1;

Sequence created.
```

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

```
select seq1.nextval Sequence_Number, rID, rname from
REVIEWER_20BRS1193;
```

```
SQL>
SQL> select seq1.nextval Sequence_Number, rID, rname from REVIEWER_20BRS1193;

SEQUENCE_NUMBER        RID RNAME
--------------- ---------- --------------------------------------------------
              1          1 Sophie Kane
              2          2 Liam Pate
              3          3 Harley Wynn
```

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

```
CREATE SEQUENCE seq2 INCREMENT BY 1 START WITH 14 MAXVALUE 30
MINVALUE 14;
INSERT INTO MOVIE_20BRS1193 VALUES(seq2.nextval,'Sherlock
Holmes','2009','Guy Ritchie');
INSERT INTO MOVIE_20BRS1193 VALUES(seq2.nextval,'The Edge of
Seventeen','2016','Kelly Fremon Craig');
```

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

Sequence created.

SQL> INSERT INTO MOVIE_20BRS1193 VALUES(seq2.nextval,'Sherlock Holmes','2009','Guy Ritchie');

1 row created.

SQL> INSERT INTO MOVIE_20BRS1193 VALUES(seq2.nextval,'The Edge of Seventeen','2016','Kelly Fremon Craig');


1 row created.
```

15. Drop the sequence seq1.

```
DROP SEQUENCE seq1;
```

```
SQL>
SQL> DROP SEQUENCE seq1;

Sequence dropped.
```