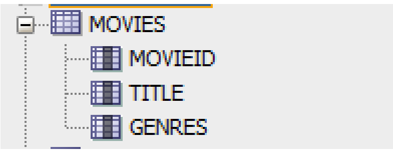
Homework #4

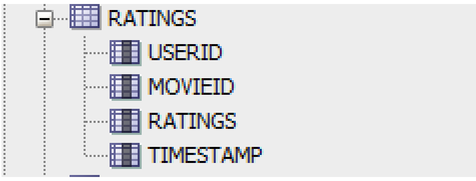
CSC 452

Zehong Zhuang

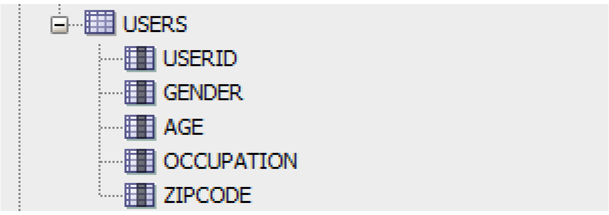
Table & Variable Names



- MOVIES
  - MOVIEID
  - TITLE
  - GENRES

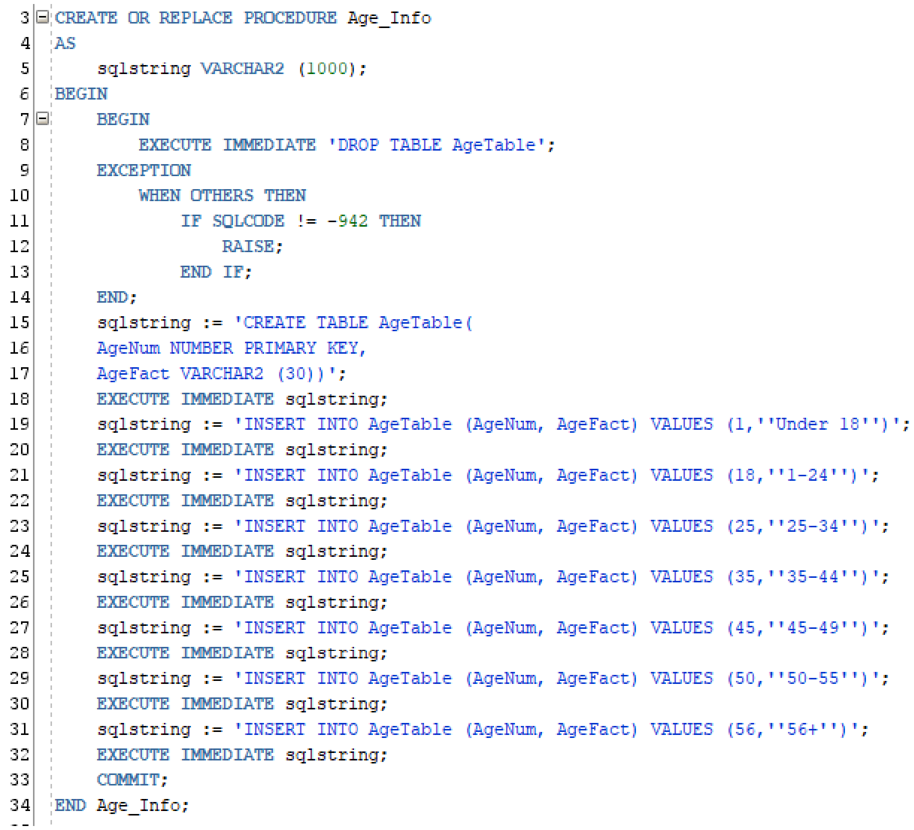


- RATINGS
  - USERID
  - MOVIEID
  - RATINGS
  - TIMESTAMP

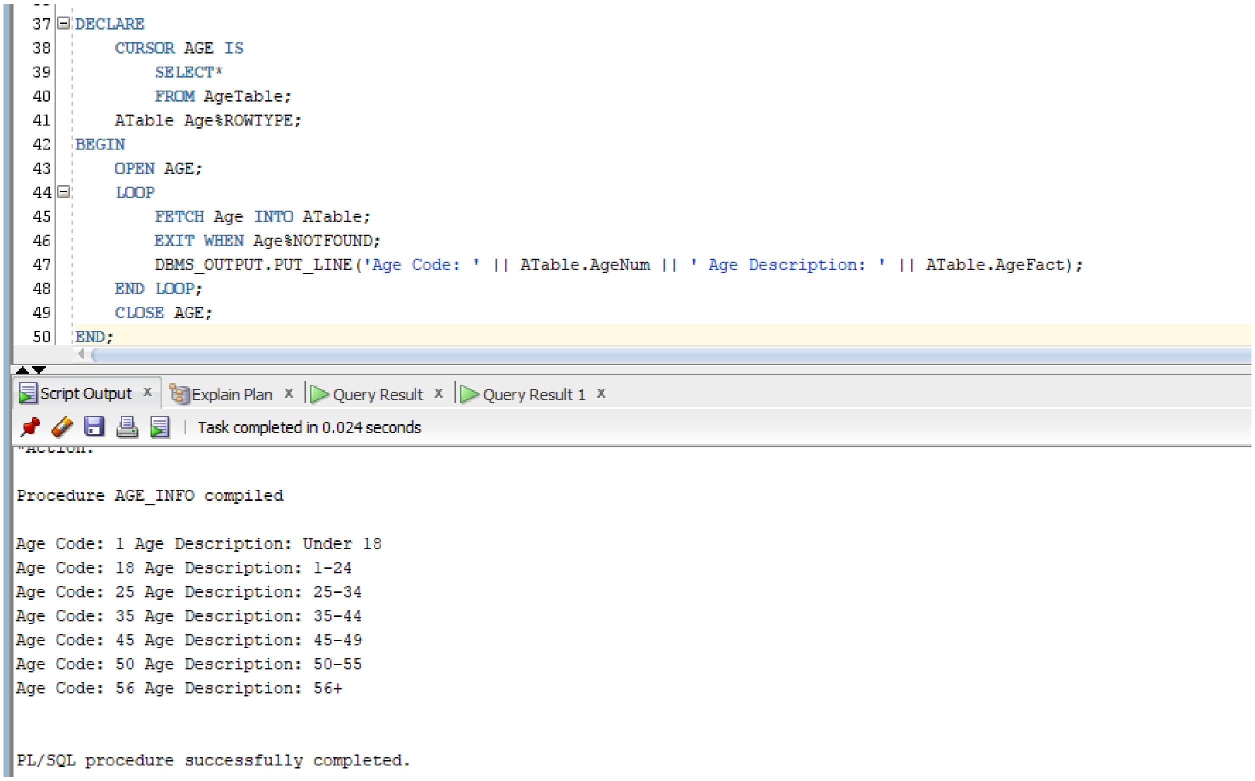


- USERS
  - USERID
  - GENDER
  - AGE
  - OCCUPATION
  - ZIPCODE

Age_Info Procedure



```
CREATE OR REPLACE PROCEDURE Age_Info
AS
    sqlstring VARCHAR2 (1000);
BEGIN
    BEGIN
        EXECUTE IMMEDIATE 'DROP TABLE AgeTable';
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE != -942 THEN
                RAISE;
            END IF;
    END;
    sqlstring := 'CREATE TABLE AgeTable(
    AgeNum NUMBER PRIMARY KEY,
    AgeFact VARCHAR2 (30))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (1,''Under 18'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (18,''1-24'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (25,''25-34'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (35,''35-44'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (45,''45-49'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (50,''50-55'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (56,''56+'')';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END Age_Info;
```

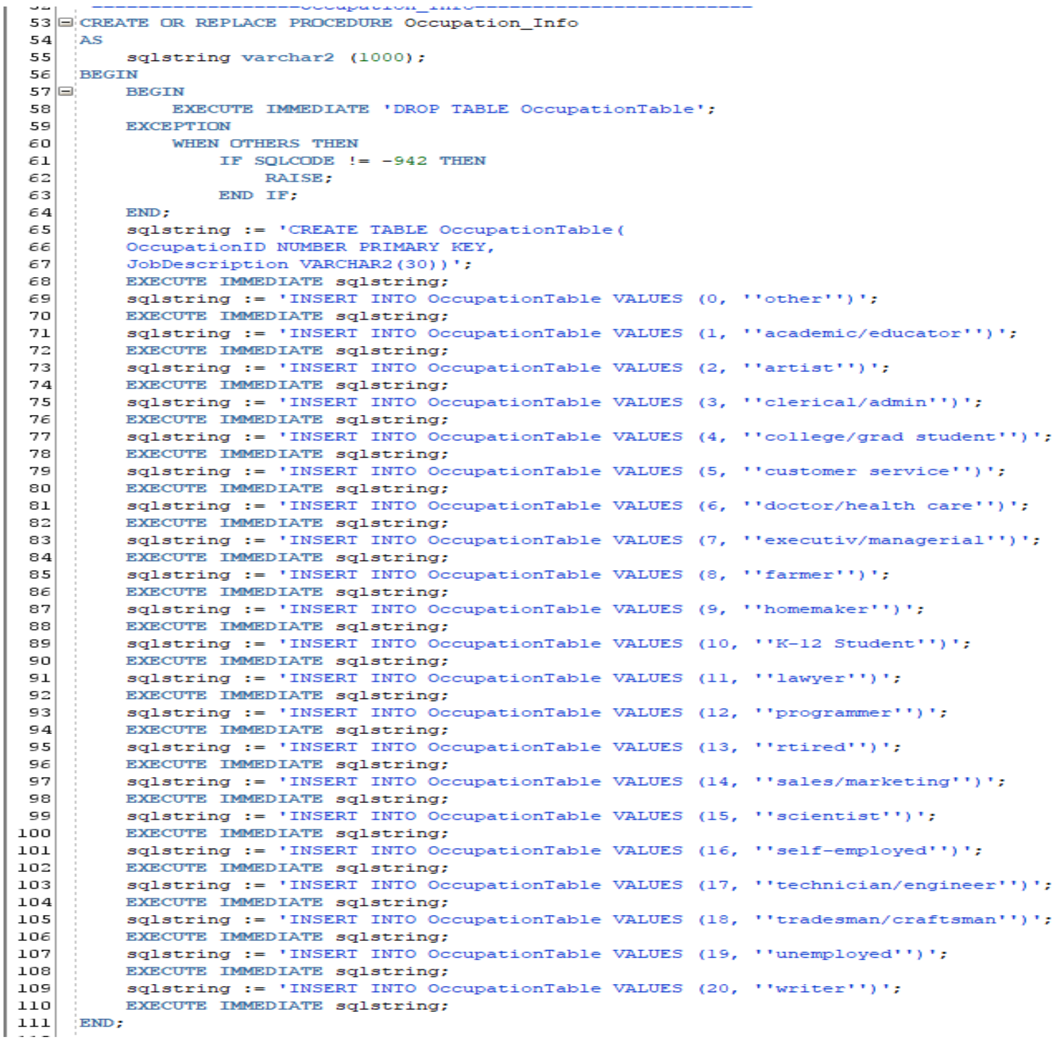


```
DECLARE
    CURSOR AGE IS
        SELECT*
        FROM AgeTable;
    ATable Age%ROWTYPE;
BEGIN
    OPEN AGE;
    LOOP
        FETCH Age INTO ATable;
        EXIT WHEN Age%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Age Code: ' || ATable.AgeNum || ' Age Description: ' || ATable.AgeFact);
    END LOOP;
    CLOSE AGE;
END;
```

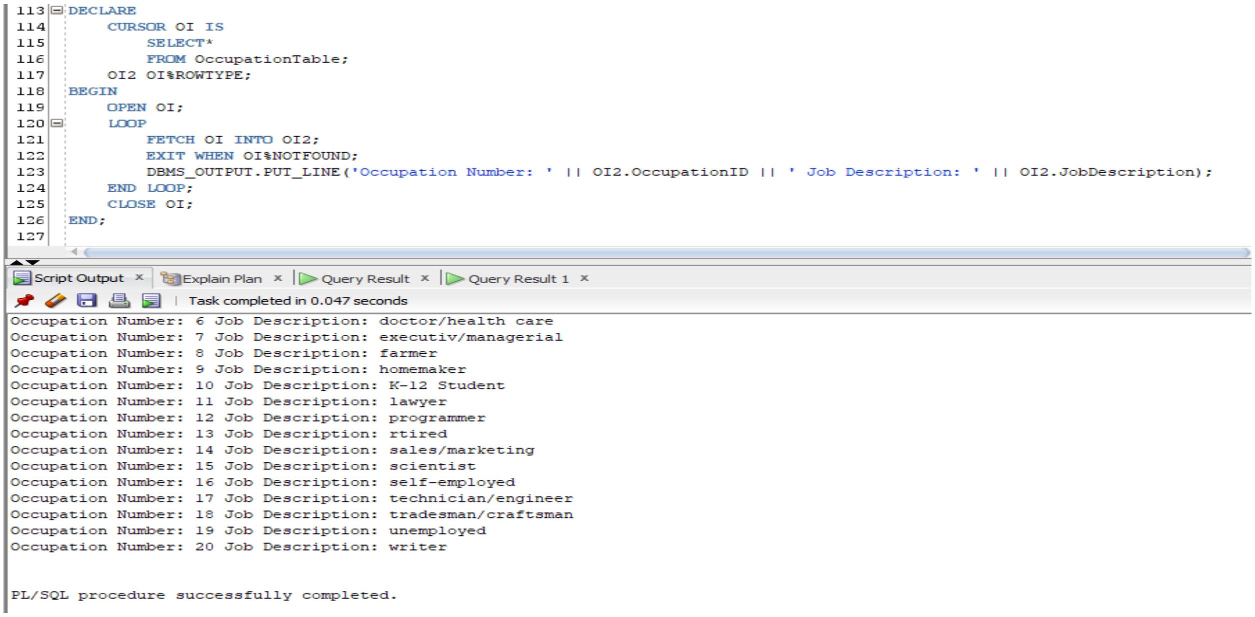
Task completed in 0.024 seconds

```
*Action:

Procedure AGE_INFO compiled

Age Code: 1 Age Description: Under 18
Age Code: 18 Age Description: 1-24
Age Code: 25 Age Description: 25-34
Age Code: 35 Age Description: 35-44
Age Code: 45 Age Description: 45-49
Age Code: 50 Age Description: 50-55
Age Code: 56 Age Description: 56+


PL/SQL procedure successfully completed.
```

Occupation Info Procedure



```
--------------------Occupation_Info----------------------------
CREATE OR REPLACE PROCEDURE Occupation_Info
AS
    sqlstring varchar2 (1000);
BEGIN
    BEGIN
        EXECUTE IMMEDIATE 'DROP TABLE OccupationTable';
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE != -942 THEN
                RAISE;
            END IF;
    END;
    sqlstring := 'CREATE TABLE OccupationTable(
    OccupationID NUMBER PRIMARY KEY,
    JobDescription VARCHAR2(30))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (0, ''other'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (1, ''academic/educator'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (2, ''artist'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (3, ''clerical/admin'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (4, ''college/grad student'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (5, ''customer service'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (6, ''doctor/health care'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (7, ''executiv/managerial'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (8, ''farmer'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (9, ''homemaker'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (10, ''K-12 Student'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (11, ''lawyer'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (12, ''programmer'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (13, ''rtired'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (14, ''sales/marketing'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (15, ''scientist'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (16, ''self-employed'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (17, ''technician/engineer'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (18, ''tradesman/craftsman'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (19, ''unemployed'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (20, ''writer'')';
    EXECUTE IMMEDIATE sqlstring;
END;
```

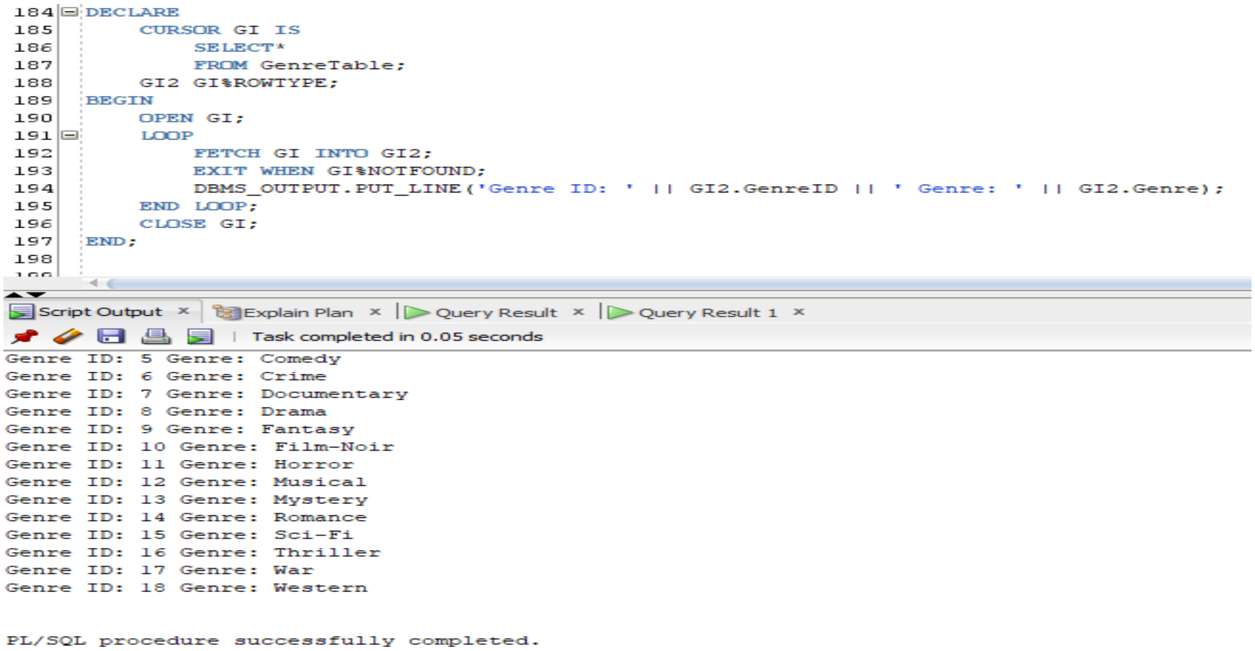


```
DECLARE
    CURSOR OI IS
        SELECT*
        FROM OccupationTable;
    OI2 OI%ROWTYPE;
BEGIN
    OPEN OI;
    LOOP
        FETCH OI INTO OI2;
        EXIT WHEN OI%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Occupation Number: ' || OI2.OccupationID || ' Job Description: ' || OI2.JobDescription);
    END LOOP;
    CLOSE OI;
END;
```

Task completed in 0.047 seconds

```
Occupation Number: 6 Job Description: doctor/health care
Occupation Number: 7 Job Description: executiv/managerial
Occupation Number: 8 Job Description: farmer
Occupation Number: 9 Job Description: homemaker
Occupation Number: 10 Job Description: K-12 Student
Occupation Number: 11 Job Description: lawyer
Occupation Number: 12 Job Description: programmer
Occupation Number: 13 Job Description: rtired
Occupation Number: 14 Job Description: sales/marketing
Occupation Number: 15 Job Description: scientist
Occupation Number: 16 Job Description: self-employed
Occupation Number: 17 Job Description: technician/engineer
Occupation Number: 18 Job Description: tradesman/craftsman
Occupation Number: 19 Job Description: unemployed
Occupation Number: 20 Job Description: writer


PL/SQL procedure successfully completed.
```

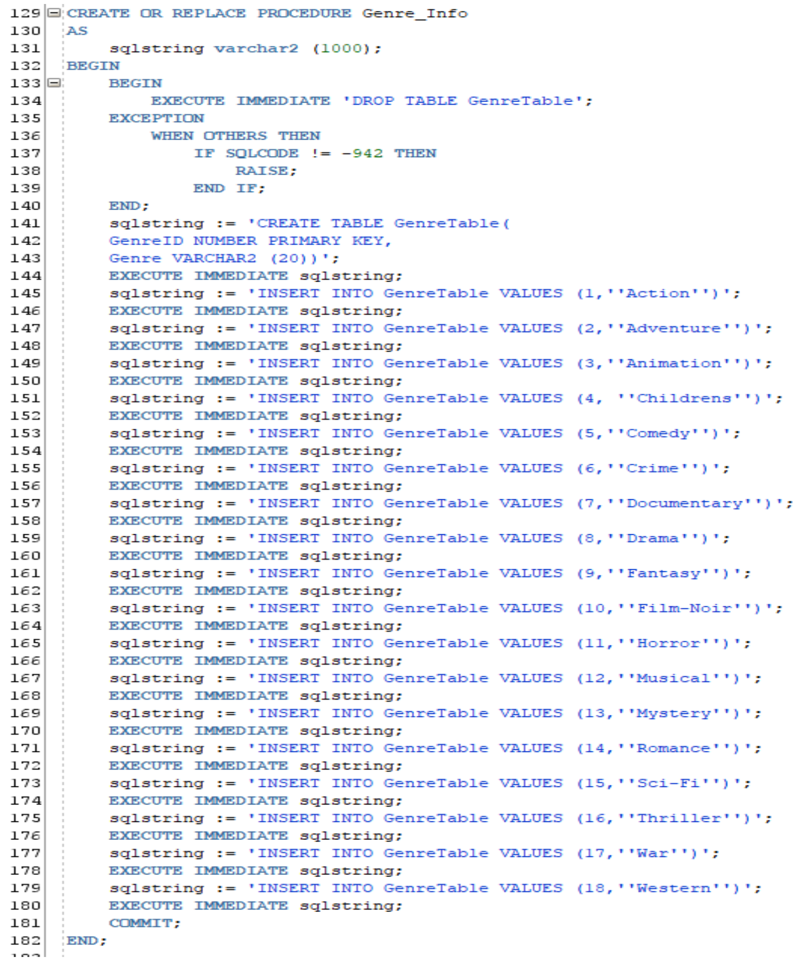
Genre Info Procedure



```
DECLARE
    CURSOR GI IS
        SELECT*
        FROM GenreTable;
    GI2 GI%ROWTYPE;
BEGIN
    OPEN GI;
    LOOP
        FETCH GI INTO GI2;
        EXIT WHEN GI%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Genre ID: ' || GI2.GenreID || ' Genre: ' || GI2.Genre);
    END LOOP;
    CLOSE GI;
END;
```

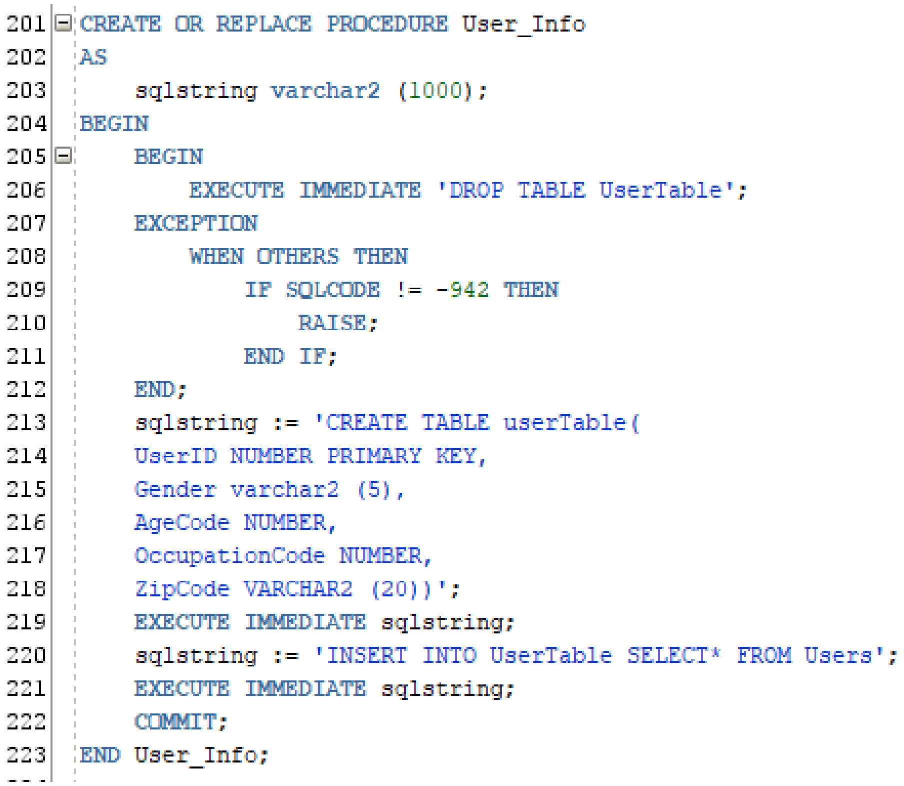
Task completed in 0.05 seconds

```
Genre ID: 5 Genre: Comedy
Genre ID: 6 Genre: Crime
Genre ID: 7 Genre: Documentary
Genre ID: 8 Genre: Drama
Genre ID: 9 Genre: Fantasy
Genre ID: 10 Genre: Film-Noir
Genre ID: 11 Genre: Horror
Genre ID: 12 Genre: Musical
Genre ID: 13 Genre: Mystery
Genre ID: 14 Genre: Romance
Genre ID: 15 Genre: Sci-Fi
Genre ID: 16 Genre: Thriller
Genre ID: 17 Genre: War
Genre ID: 18 Genre: Western


PL/SQL procedure successfully completed.
```

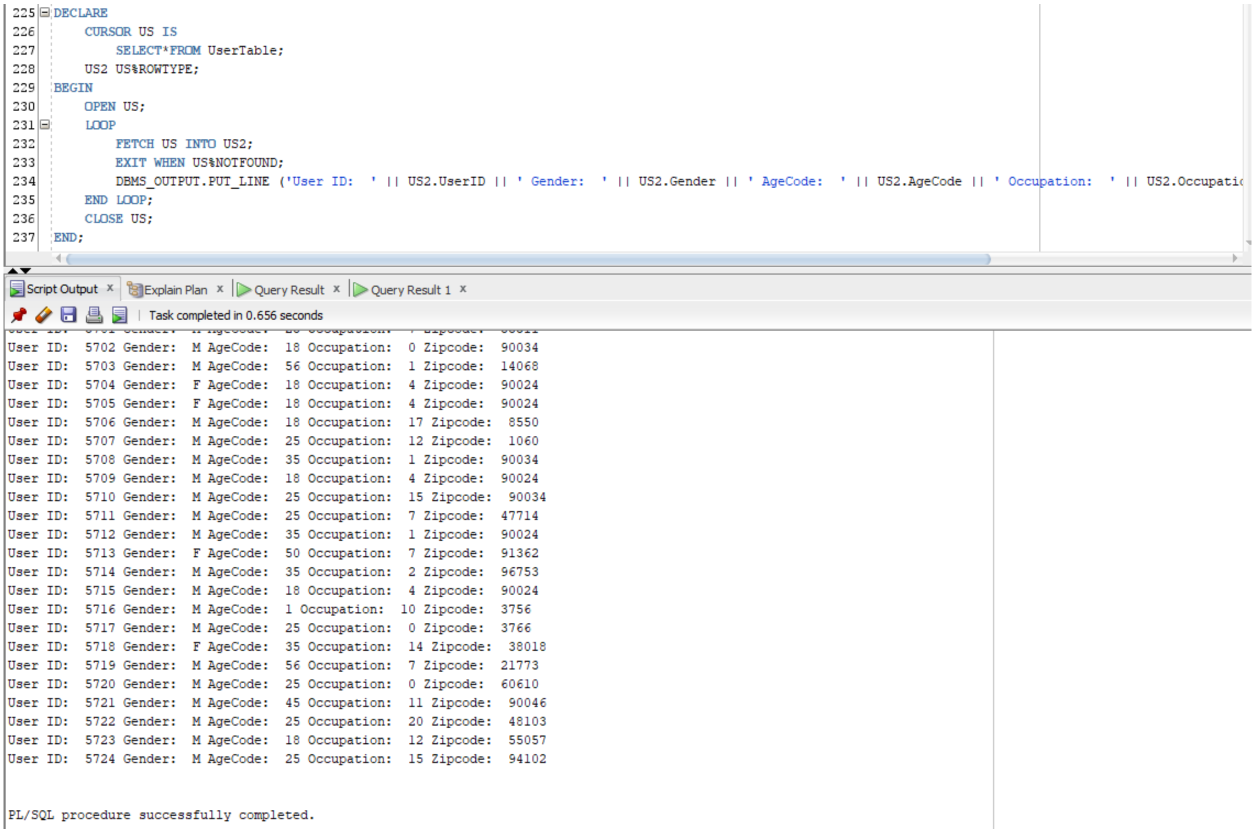


```
CREATE OR REPLACE PROCEDURE Genre_Info
AS
    sqlstring varchar2 (1000);
BEGIN
    BEGIN
        EXECUTE IMMEDIATE 'DROP TABLE GenreTable';
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE != -942 THEN
                RAISE;
            END IF;
    END;
    sqlstring := 'CREATE TABLE GenreTable(
    GenreID NUMBER PRIMARY KEY,
    Genre VARCHAR2 (20))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (1,''Action'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (2,''Adventure'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (3,''Animation'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (4, ''Childrens'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (5,''Comedy'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (6,''Crime'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (7,''Documentary'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (8,''Drama'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (9,''Fantasy'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (10,''Film-Noir'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (11,''Horror'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (12,''Musical'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (13,''Mystery'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (14,''Romance'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (15,''Sci-Fi'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (16,''Thriller'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (17,''War'')';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (18,''Western'')';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END;
```

User Info Procedure



```
CREATE OR REPLACE PROCEDURE User_Info
AS
    sqlstring varchar2 (1000);
BEGIN
    BEGIN
        EXECUTE IMMEDIATE 'DROP TABLE UserTable';
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE != -942 THEN
                RAISE;
            END IF;
    END;
    sqlstring := 'CREATE TABLE userTable(
    UserID NUMBER PRIMARY KEY,
    Gender varchar2 (5),
    AgeCode NUMBER,
    OccupationCode NUMBER,
    ZipCode VARCHAR2 (20))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO UserTable SELECT* FROM Users';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END User_Info;
```

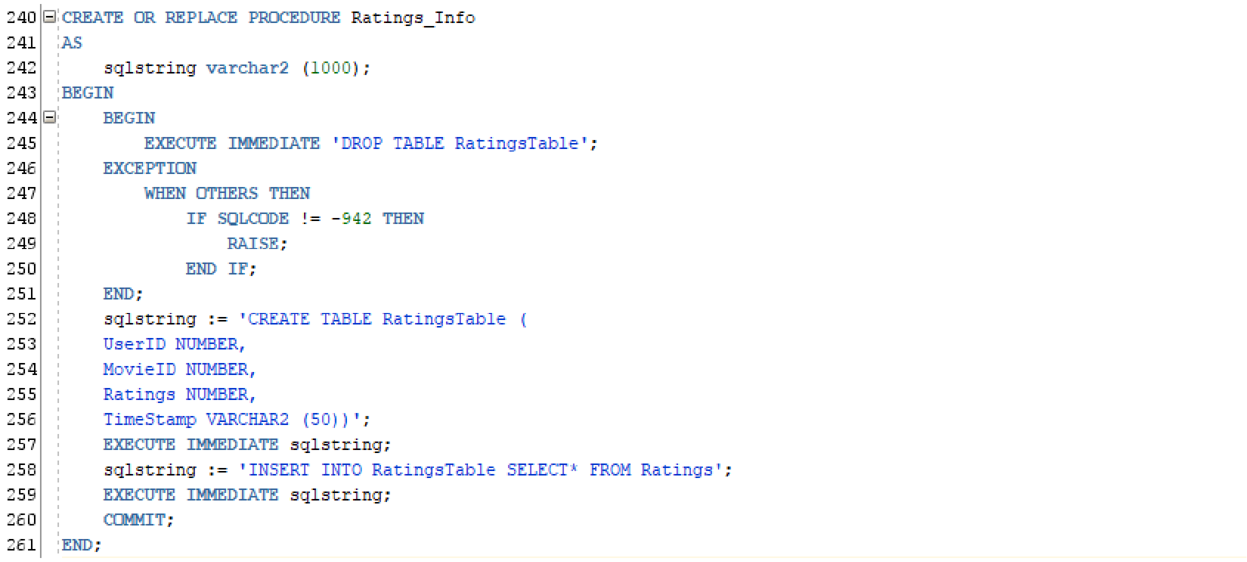


```
DECLARE
    CURSOR US IS
        SELECT*FROM UserTable;
    US2 US%ROWTYPE;
BEGIN
    OPEN US;
    LOOP
        FETCH US INTO US2;
        EXIT WHEN US%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE ('User ID:  ' || US2.UserID || ' Gender:  ' || US2.Gender || ' AgeCode:  ' || US2.AgeCode || ' Occupation:  ' || US2.Occupati
    END LOOP;
    CLOSE US;
END;
```

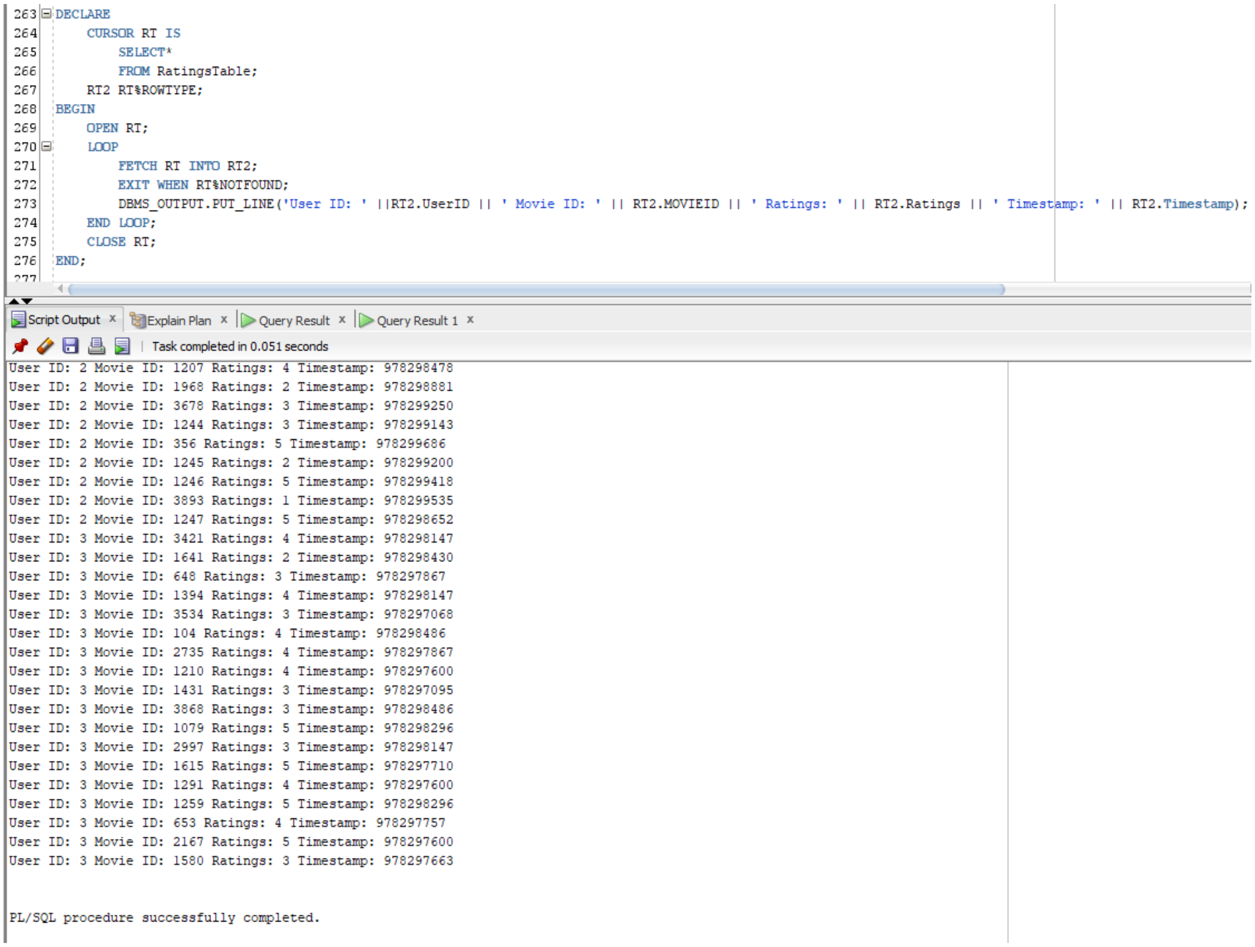
Task completed in 0.656 seconds

```
User ID:  5702 Gender:  M AgeCode:  18 Occupation:  0 Zipcode:  90034
User ID:  5703 Gender:  M AgeCode:  56 Occupation:  1 Zipcode:  14068
User ID:  5704 Gender:  F AgeCode:  18 Occupation:  4 Zipcode:  90024
User ID:  5705 Gender:  F AgeCode:  18 Occupation:  4 Zipcode:  90024
User ID:  5706 Gender:  M AgeCode:  18 Occupation:  17 Zipcode:  8550
User ID:  5707 Gender:  M AgeCode:  25 Occupation:  12 Zipcode:  1060
User ID:  5708 Gender:  M AgeCode:  35 Occupation:  1 Zipcode:  90034
User ID:  5709 Gender:  M AgeCode:  18 Occupation:  4 Zipcode:  90024
User ID:  5710 Gender:  M AgeCode:  25 Occupation:  15 Zipcode:  90034
User ID:  5711 Gender:  M AgeCode:  25 Occupation:  7 Zipcode:  47714
User ID:  5712 Gender:  M AgeCode:  35 Occupation:  1 Zipcode:  90024
User ID:  5713 Gender:  F AgeCode:  50 Occupation:  7 Zipcode:  91362
User ID:  5714 Gender:  M AgeCode:  35 Occupation:  2 Zipcode:  96753
User ID:  5715 Gender:  M AgeCode:  18 Occupation:  4 Zipcode:  90024
User ID:  5716 Gender:  M AgeCode:  1 Occupation:  10 Zipcode:  3756
User ID:  5717 Gender:  M AgeCode:  25 Occupation:  0 Zipcode:  3766
User ID:  5718 Gender:  F AgeCode:  35 Occupation:  14 Zipcode:  38018
User ID:  5719 Gender:  M AgeCode:  56 Occupation:  7 Zipcode:  21773
User ID:  5720 Gender:  M AgeCode:  25 Occupation:  0 Zipcode:  60610
User ID:  5721 Gender:  M AgeCode:  45 Occupation:  11 Zipcode:  90046
User ID:  5722 Gender:  M AgeCode:  25 Occupation:  20 Zipcode:  48103
User ID:  5723 Gender:  M AgeCode:  18 Occupation:  12 Zipcode:  55057
User ID:  5724 Gender:  M AgeCode:  25 Occupation:  15 Zipcode:  94102


PL/SQL procedure successfully completed.
```

Ratings Info Procedure



```
CREATE OR REPLACE PROCEDURE Ratings_Info
AS
    sqlstring varchar2 (1000);
BEGIN
    BEGIN
        EXECUTE IMMEDIATE 'DROP TABLE RatingsTable';
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE != -942 THEN
                RAISE;
            END IF;
    END;
    sqlstring := 'CREATE TABLE RatingsTable (
    UserID NUMBER,
    MovieID NUMBER,
    Ratings NUMBER,
    TimeStamp VARCHAR2 (50))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO RatingsTable SELECT* FROM Ratings';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END;
```

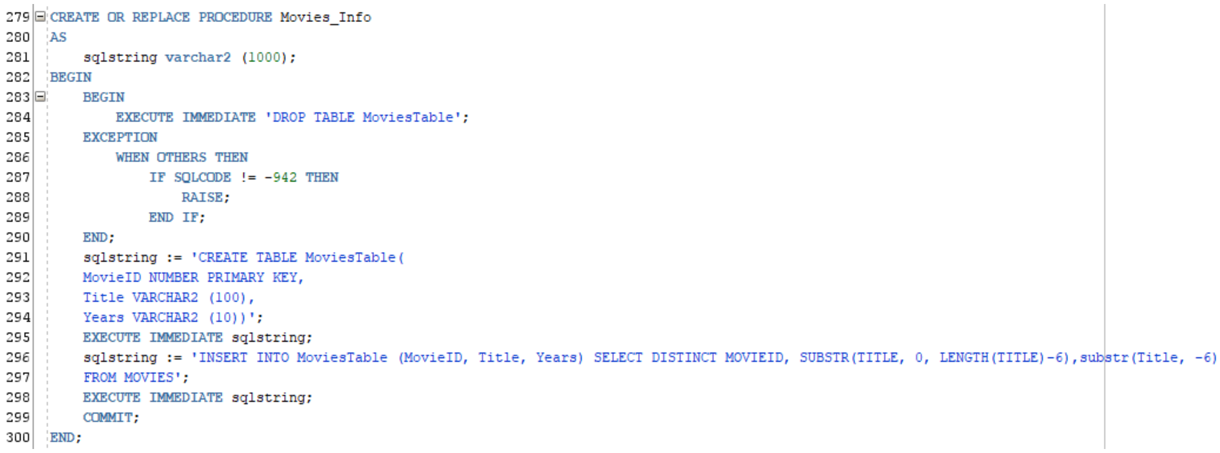


```
DECLARE
    CURSOR RT IS
        SELECT*
        FROM RatingsTable;
    RT2 RT%ROWTYPE;
BEGIN
    OPEN RT;
    LOOP
        FETCH RT INTO RT2;
        EXIT WHEN RT%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('User ID: ' ||RT2.UserID || ' Movie ID: ' || RT2.MOVIEID || ' Ratings: ' || RT2.Ratings || ' Timestamp: ' || RT2.Timestamp);
    END LOOP;
    CLOSE RT;
END;
```

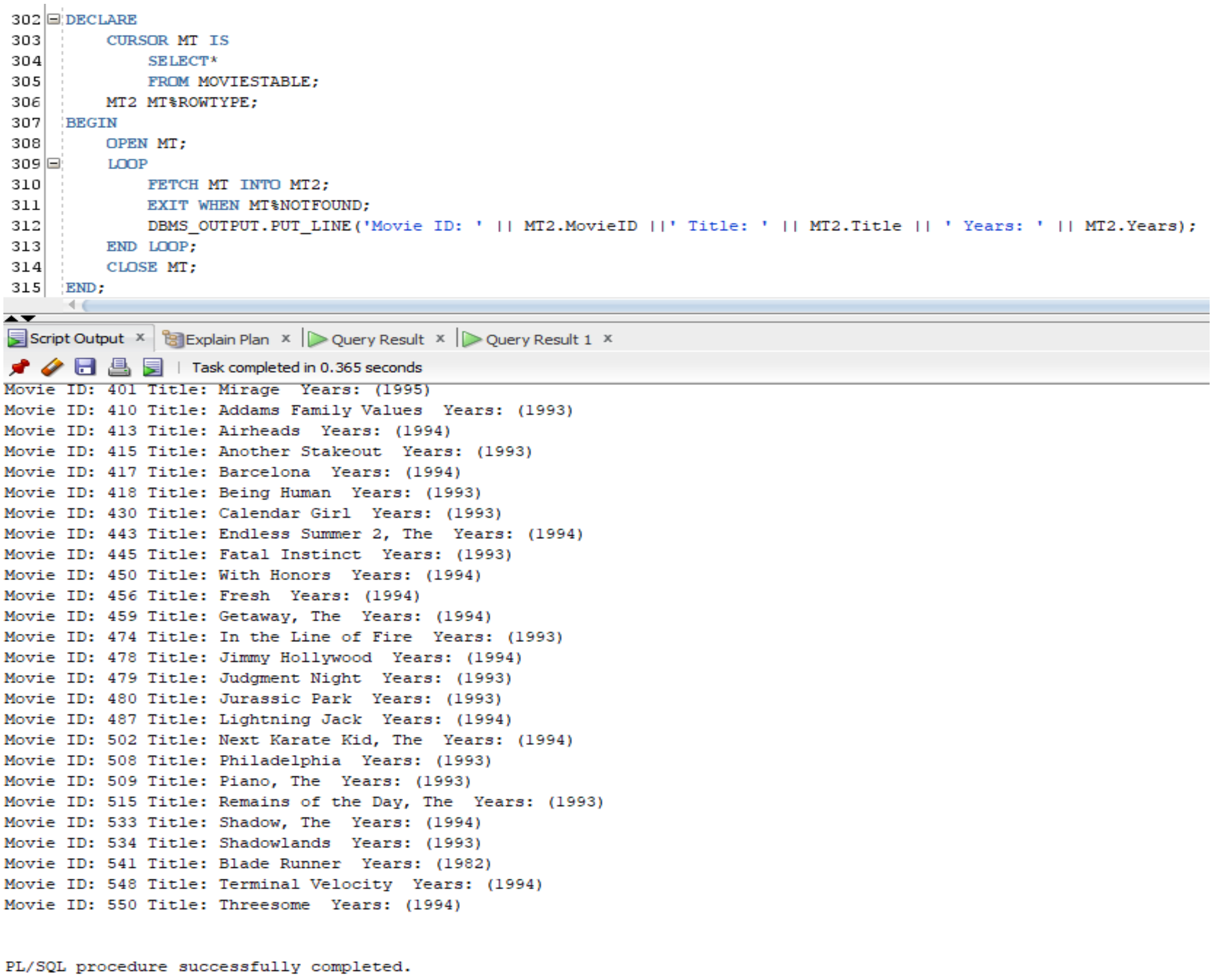
Task completed in 0.051 seconds

```
User ID: 2 Movie ID: 1207 Ratings: 4 Timestamp: 978298478
User ID: 2 Movie ID: 1968 Ratings: 2 Timestamp: 978298881
User ID: 2 Movie ID: 3678 Ratings: 3 Timestamp: 978299250
User ID: 2 Movie ID: 1244 Ratings: 3 Timestamp: 978299143
User ID: 2 Movie ID: 356 Ratings: 5 Timestamp: 978299686
User ID: 2 Movie ID: 1245 Ratings: 2 Timestamp: 978299200
User ID: 2 Movie ID: 1246 Ratings: 5 Timestamp: 978299418
User ID: 2 Movie ID: 3893 Ratings: 1 Timestamp: 978299535
User ID: 2 Movie ID: 1247 Ratings: 5 Timestamp: 978298652
User ID: 3 Movie ID: 3421 Ratings: 4 Timestamp: 978298147
User ID: 3 Movie ID: 1641 Ratings: 2 Timestamp: 978298430
User ID: 3 Movie ID: 648 Ratings: 3 Timestamp: 978297867
User ID: 3 Movie ID: 1394 Ratings: 4 Timestamp: 978298147
User ID: 3 Movie ID: 3534 Ratings: 3 Timestamp: 978297068
User ID: 3 Movie ID: 104 Ratings: 4 Timestamp: 978298486
User ID: 3 Movie ID: 2735 Ratings: 4 Timestamp: 978297867
User ID: 3 Movie ID: 1210 Ratings: 4 Timestamp: 978297600
User ID: 3 Movie ID: 1431 Ratings: 3 Timestamp: 978297095
User ID: 3 Movie ID: 3868 Ratings: 3 Timestamp: 978298486
User ID: 3 Movie ID: 1079 Ratings: 5 Timestamp: 978298296
User ID: 3 Movie ID: 2997 Ratings: 3 Timestamp: 978298147
User ID: 3 Movie ID: 1615 Ratings: 5 Timestamp: 978297710
User ID: 3 Movie ID: 1291 Ratings: 4 Timestamp: 978297600
User ID: 3 Movie ID: 1259 Ratings: 5 Timestamp: 978298296
User ID: 3 Movie ID: 653 Ratings: 4 Timestamp: 978297757
User ID: 3 Movie ID: 2167 Ratings: 5 Timestamp: 978297600
User ID: 3 Movie ID: 1580 Ratings: 3 Timestamp: 978297663


PL/SQL procedure successfully completed.
```

Movies Info Procedure



```
CREATE OR REPLACE PROCEDURE Movies_Info
AS
    sqlstring varchar2 (1000);
BEGIN
    BEGIN
        EXECUTE IMMEDIATE 'DROP TABLE MoviesTable';
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE != -942 THEN
                RAISE;
            END IF;
    END;
    sqlstring := 'CREATE TABLE MoviesTable(
    MovieID NUMBER PRIMARY KEY,
    Title VARCHAR2 (100),
    Years VARCHAR2 (10))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO MoviesTable (MovieID, Title, Years) SELECT DISTINCT MOVIEID, SUBSTR(TITLE, 0, LENGTH(TITLE)-6),substr(Title, -6)
    FROM MOVIES';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END;
```

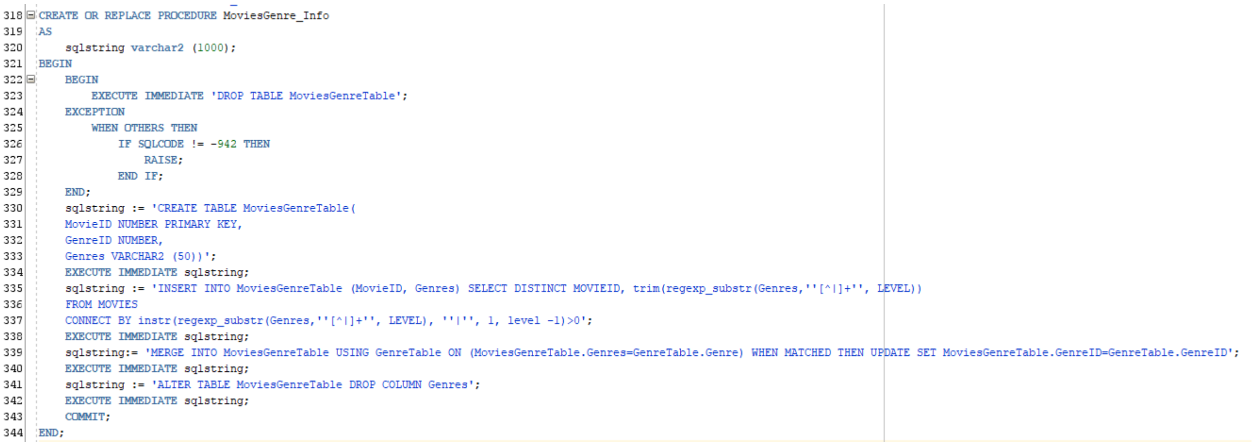


```
DECLARE
    CURSOR MT IS
        SELECT*
        FROM MOVIESTABLE;
    MT2 MT%ROWTYPE;
BEGIN
    OPEN MT;
    LOOP
        FETCH MT INTO MT2;
        EXIT WHEN MT%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Movie ID: ' || MT2.MovieID ||' Title: ' || MT2.Title || ' Years: ' || MT2.Years);
    END LOOP;
    CLOSE MT;
END;
```

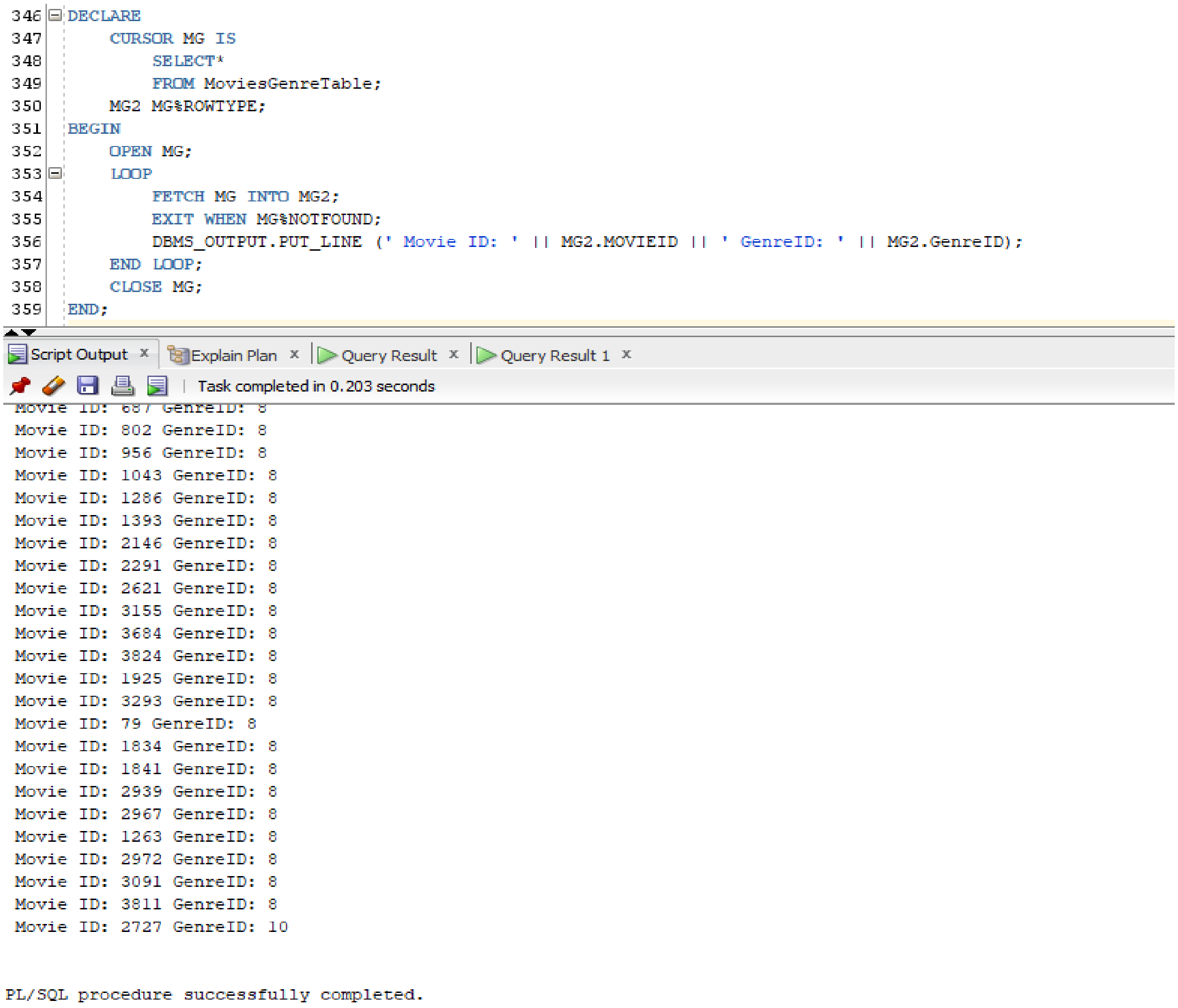
Task completed in 0.365 seconds

```
Movie ID: 401 Title: Mirage  Years: (1995)
Movie ID: 410 Title: Addams Family Values  Years: (1993)
Movie ID: 413 Title: Airheads  Years: (1994)
Movie ID: 415 Title: Another Stakeout  Years: (1993)
Movie ID: 417 Title: Barcelona  Years: (1994)
Movie ID: 418 Title: Being Human  Years: (1993)
Movie ID: 430 Title: Calendar Girl  Years: (1993)
Movie ID: 443 Title: Endless Summer 2, The  Years: (1994)
Movie ID: 445 Title: Fatal Instinct  Years: (1993)
Movie ID: 450 Title: With Honors  Years: (1994)
Movie ID: 456 Title: Fresh  Years: (1994)
Movie ID: 459 Title: Getaway, The  Years: (1994)
Movie ID: 474 Title: In the Line of Fire  Years: (1993)
Movie ID: 478 Title: Jimmy Hollywood  Years: (1994)
Movie ID: 479 Title: Judgment Night  Years: (1993)
Movie ID: 480 Title: Jurassic Park  Years: (1993)
Movie ID: 487 Title: Lightning Jack  Years: (1994)
Movie ID: 502 Title: Next Karate Kid, The  Years: (1994)
Movie ID: 508 Title: Philadelphia  Years: (1993)
Movie ID: 509 Title: Piano, The  Years: (1993)
Movie ID: 515 Title: Remains of the Day, The  Years: (1993)
Movie ID: 533 Title: Shadow, The  Years: (1994)
Movie ID: 534 Title: Shadowlands  Years: (1993)
Movie ID: 541 Title: Blade Runner  Years: (1982)
Movie ID: 548 Title: Terminal Velocity  Years: (1994)
Movie ID: 550 Title: Threesome  Years: (1994)


PL/SQL procedure successfully completed.
```

Movie Genre Info



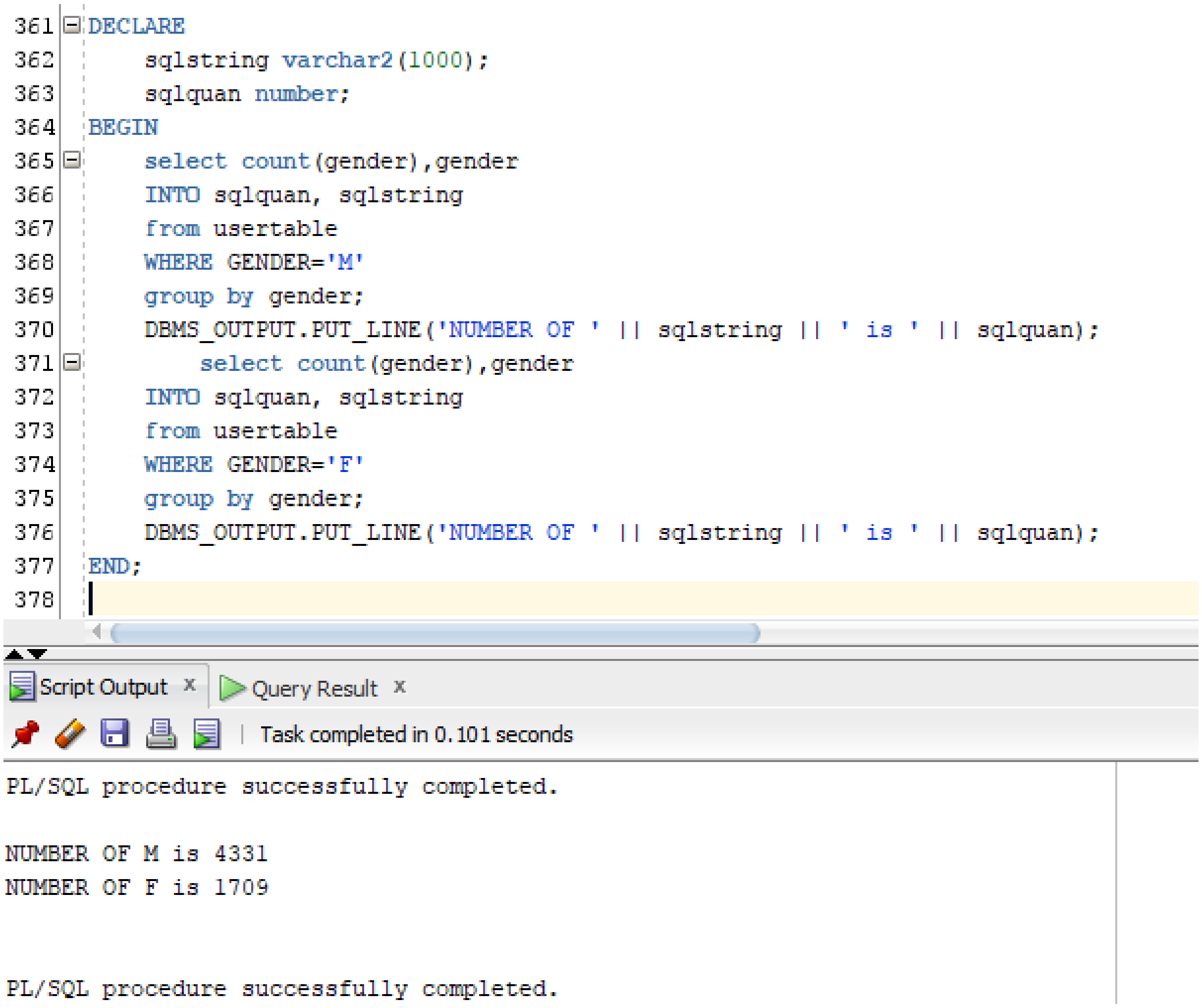
```
CREATE OR REPLACE PROCEDURE MoviesGenre_Info
AS
    sqlstring varchar2 (1000);
BEGIN
    BEGIN
        EXECUTE IMMEDIATE 'DROP TABLE MoviesGenreTable';
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE != -942 THEN
                RAISE;
            END IF;
    END;
    sqlstring := 'CREATE TABLE MoviesGenreTable(
    MovieID NUMBER PRIMARY KEY,
    GenreID NUMBER,
    Genres VARCHAR2 (50))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO MoviesGenreTable (MovieID, Genres) SELECT DISTINCT MOVIEID, trim(regexp_substr(Genres,''[^|]+'', LEVEL))
    FROM MOVIES
    CONNECT BY instr(regexp_substr(Genres,''[^|]+'', LEVEL), ''|'', 1, level -1)>0';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring:= 'MERGE INTO MoviesGenreTable USING GenreTable ON (MoviesGenreTable.Genres=GenreTable.Genre) WHEN MATCHED THEN UPDATE SET MoviesGenreTable.GenreID=GenreTable.GenreID';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'ALTER TABLE MoviesGenreTable DROP COLUMN Genres';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END;
```



```
DECLARE
    CURSOR MG IS
        SELECT*
        FROM MoviesGenreTable;
    MG2 MG%ROWTYPE;
BEGIN
    OPEN MG;
    LOOP
        FETCH MG INTO MG2;
        EXIT WHEN MG%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (' Movie ID: ' || MG2.MOVIEID || ' GenreID: ' || MG2.GenreID);
    END LOOP;
    CLOSE MG;
END;
```

```
 Movie ID: 687 GenreID: 8
 Movie ID: 802 GenreID: 8
 Movie ID: 956 GenreID: 8
 Movie ID: 1043 GenreID: 8
 Movie ID: 1286 GenreID: 8
 Movie ID: 1393 GenreID: 8
 Movie ID: 2146 GenreID: 8
 Movie ID: 2291 GenreID: 8
 Movie ID: 2621 GenreID: 8
 Movie ID: 3155 GenreID: 8
 Movie ID: 3684 GenreID: 8
 Movie ID: 3824 GenreID: 8
 Movie ID: 1925 GenreID: 8
 Movie ID: 3293 GenreID: 8
 Movie ID: 79 GenreID: 8
 Movie ID: 1834 GenreID: 8
 Movie ID: 1841 GenreID: 8
 Movie ID: 2939 GenreID: 8
 Movie ID: 2967 GenreID: 8
 Movie ID: 1263 GenreID: 8
 Movie ID: 2972 GenreID: 8
 Movie ID: 3091 GenreID: 8
 Movie ID: 3811 GenreID: 8
 Movie ID: 2727 GenreID: 10


PL/SQL procedure successfully completed.
```

Query Question



```
DECLARE
    sqlstring varchar2(1000);
    sqlquan number;
BEGIN
    select count(gender),gender
    INTO sqlquan, sqlstring
    from usertable
    WHERE GENDER='M'
    group by gender;
    DBMS_OUTPUT.PUT_LINE('NUMBER OF ' || sqlstring || ' is ' || sqlquan);
        select count(gender),gender
    INTO sqlquan, sqlstring
    from usertable
    WHERE GENDER='F'
    group by gender;
    DBMS_OUTPUT.PUT_LINE('NUMBER OF ' || sqlstring || ' is ' || sqlquan);
END;
```

Task completed in 0.101 seconds

```
PL/SQL procedure successfully completed.

NUMBER OF M is 4331
NUMBER OF F is 1709


PL/SQL procedure successfully completed.
```

Females are the majority of users who play important roles in movie ratings. Movie producer should consider and evaluate female audience’s opinions if they want their movies receiving higher rates.

```
SET SERVEROUT ON;
========================Age Info==============================
CREATE OR REPLACE PROCEDURE Age_Info
AS
   sqlstring VARCHAR2 (1000);
BEGIN
   BEGIN
      EXECUTE IMMEDIATE 'DROP TABLE AgeTable';
   EXCEPTION
      WHEN OTHERS THEN
         IF SQLCODE != -942 THEN
            RAISE;
         END IF;
   END;
   sqlstring := 'CREATE TABLE AgeTable(
   AgeNum NUMBER PRIMARY KEY,
   AgeFact VARCHAR2 (30))';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (1,''Under 18'')';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (18,''1-24'')';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (25,''25-34'')';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (35,''35-44'')';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (45,''45-49'')';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (50,"50-55")';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (56,"56+")';
   EXECUTE IMMEDIATE sqlstring;
   COMMIT;
END Age_Info;



DECLARE
   CURSOR AGE IS
      SELECT*
      FROM AgeTable;
   ATable Age%ROWTYPE;
BEGIN
   OPEN AGE;
   LOOP
      FETCH Age INTO ATable;
      EXIT WHEN Age%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Age Code: ' || ATable.AgeNum || ' Age Description: ' || ATable.AgeFact);
   END LOOP;
   CLOSE AGE;
END;

==================Occupation_Info========================
CREATE OR REPLACE PROCEDURE Occupation_Info
AS
   sqlstring varchar2 (1000);
BEGIN
```

```
BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE OccupationTable';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -942 THEN
            RAISE;
        END IF;
END;
sqlstring := 'CREATE TABLE OccupationTable(
OccupationID NUMBER PRIMARY KEY,
JobDescription VARCHAR2(30))';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO OccupationTable VALUES (0, "other")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO OccupationTable VALUES (1, "academic/educator")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO OccupationTable VALUES (2, "artist")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO OccupationTable VALUES (3, "clerical/admin")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO OccupationTable VALUES (4, "college/grad student")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO OccupationTable VALUES (5, "customer service")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO OccupationTable VALUES (6, "doctor/health care")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO OccupationTable VALUES (7, "executiv/managerial")';
EXECUTE IMMEDIATE sqlstring;
```

```
    sqlstring := 'INSERT INTO OccupationTable VALUES (8, "farmer")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (9, "homemaker")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (10, "K-12 Student")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (11, "lawyer")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (12, "programmer")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (13, "rtired")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (14, "sales/marketing")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (15, "scientist")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (16, "self-employed")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (17, "technician/engineer")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (18, "tradesman/craftsman")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (19, "unemployed")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO OccupationTable VALUES (20, "writer")';
    EXECUTE IMMEDIATE sqlstring;
END;
```

```
DECLARE
   CURSOR OI IS
      SELECT*
      FROM OccupationTable;
   OI2 OI%ROWTYPE;
BEGIN
   OPEN OI;
   LOOP
      FETCH OI INTO OI2;
      EXIT WHEN OI%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Occupation Number: ' || OI2.OccupationID || ' Job Description: ' || OI2.JobDescription);
   END LOOP;
   CLOSE OI;
END;
```

==================Genre Info============================

```
CREATE OR REPLACE PROCEDURE Genre_Info
AS
   sqlstring varchar2 (1000);
BEGIN
   BEGIN
      EXECUTE IMMEDIATE 'DROP TABLE GenreTable';
   EXCEPTION
      WHEN OTHERS THEN
         IF SQLCODE != -942 THEN
            RAISE;
         END IF;
   END;
```

```
sqlstring := 'CREATE TABLE GenreTable(
GenreID NUMBER PRIMARY KEY,
Genre VARCHAR2 (20))';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (1,"Action")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (2,"Adventure")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (3,"Animation")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (4, "Childrens")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (5,"Comedy")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (6,"Crime")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (7,"Documentary")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (8,"Drama")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (9,"Fantasy")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (10,"Film-Noir")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (11,"Horror")';
EXECUTE IMMEDIATE sqlstring;
sqlstring := 'INSERT INTO GenreTable VALUES (12,"Musical")';
EXECUTE IMMEDIATE sqlstring;
```

```
    sqlstring := 'INSERT INTO GenreTable VALUES (13,"Mystery")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (14,"Romance")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (15,"Sci-Fi")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (16,"Thriller")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (17,"War")';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO GenreTable VALUES (18,"Western")';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END;

DECLARE
    CURSOR GI IS
        SELECT*
        FROM GenreTable;
    GI2 GI%ROWTYPE;
BEGIN
    OPEN GI;
    LOOP
        FETCH GI INTO GI2;
        EXIT WHEN GI%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Genre ID: ' || GI2.GenreID || ' Genre: ' || GI2.Genre);
    END LOOP;
    CLOSE GI;
```

```
END;



==================User Info============================
CREATE OR REPLACE PROCEDURE User_Info
AS
   sqlstring varchar2 (1000);
BEGIN
   BEGIN
      EXECUTE IMMEDIATE 'DROP TABLE UserTable';
   EXCEPTION
      WHEN OTHERS THEN
         IF SQLCODE != -942 THEN
            RAISE;
         END IF;
   END;
   sqlstring := 'CREATE TABLE userTable(
   UserID NUMBER PRIMARY KEY,
   Gender varchar2 (5),
   AgeCode NUMBER,
   OccupationCode NUMBER,
   ZipCode VARCHAR2 (20))';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO UserTable SELECT* FROM Users';
   EXECUTE IMMEDIATE sqlstring;
   COMMIT;
END User_Info;
```

```
DECLARE
   CURSOR US IS
      SELECT*FROM UserTable;
   US2 US%ROWTYPE;
BEGIN
   OPEN US;
   LOOP
      FETCH US INTO US2;
      EXIT WHEN US%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE ('User ID:  ' || US2.UserID || ' Gender:  ' || US2.Gender || ' 
AgeCode:  ' || US2.AgeCode || ' Occupation:  ' || US2.OccupationCode || ' Zipcode:  ' || 
US2.Zipcode);
   END LOOP;
   CLOSE US;
END;


================Ratings Info========================
CREATE OR REPLACE PROCEDURE Ratings_Info
AS
   sqlstring varchar2 (1000);
BEGIN
   BEGIN
      EXECUTE IMMEDIATE 'DROP TABLE RatingsTable';
   EXCEPTION
      WHEN OTHERS THEN
         IF SQLCODE != -942 THEN
            RAISE;
         END IF;
   END;
```

```
    sqlstring := 'CREATE TABLE RatingsTable (
    UserID NUMBER,
    MovieID NUMBER,
    Ratings NUMBER,
    TimeStamp VARCHAR2 (50))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO RatingsTable SELECT* FROM Ratings';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END;

DECLARE
    CURSOR RT IS
        SELECT*
        FROM RatingsTable;
    RT2 RT%ROWTYPE;
BEGIN
    OPEN RT;
    LOOP
        FETCH RT INTO RT2;
        EXIT WHEN RT%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('User ID: ' ||RT2.UserID || ' Movie ID: ' || RT2.MOVIEID || ' Ratings: ' || RT2.Ratings || ' Timestamp: ' || RT2.Timestamp);
    END LOOP;
    CLOSE RT;
END;

================Movies Info==============================
CREATE OR REPLACE PROCEDURE Movies_Info
```

```
AS
    sqlstring varchar2 (1000);
BEGIN
    BEGIN
        EXECUTE IMMEDIATE 'DROP TABLE MoviesTable';
    EXCEPTION
        WHEN OTHERS THEN
            IF SQLCODE != -942 THEN
                RAISE;
            END IF;
    END;
    sqlstring := 'CREATE TABLE MoviesTable(
    MovieID NUMBER PRIMARY KEY,
    Title VARCHAR2 (100),
    Years VARCHAR2 (10))';
    EXECUTE IMMEDIATE sqlstring;
    sqlstring := 'INSERT INTO MoviesTable (MovieID, Title, Years) SELECT DISTINCT
MOVIEID, SUBSTR(TITLE, 0, LENGTH(TITLE)-6),substr(Title, -6)
    FROM MOVIES';
    EXECUTE IMMEDIATE sqlstring;
    COMMIT;
END;

DECLARE
    CURSOR MT IS
        SELECT*
        FROM MOVIESTABLE;
    MT2 MT%ROWTYPE;
BEGIN
```

```
   OPEN MT;
   LOOP
      FETCH MT INTO MT2;
      EXIT WHEN MT%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Movie ID: ' || MT2.MovieID ||' Title: ' || MT2.Title || ' Years: ' || MT2.Years);
   END LOOP;
   CLOSE MT;
END;

======================Movie_Genre Info=====================
CREATE OR REPLACE PROCEDURE MoviesGenre_Info
AS
   sqlstring varchar2 (1000);
BEGIN
   BEGIN
      EXECUTE IMMEDIATE 'DROP TABLE MoviesGenreTable';
   EXCEPTION
      WHEN OTHERS THEN
         IF SQLCODE != -942 THEN
            RAISE;
         END IF;
   END;
   sqlstring := 'CREATE TABLE MoviesGenreTable(
   MovieID NUMBER PRIMARY KEY,
   GenreID NUMBER,
   Genres VARCHAR2 (50))';
   EXECUTE IMMEDIATE sqlstring;
   sqlstring := 'INSERT INTO MoviesGenreTable (MovieID, Genres) SELECT DISTINCT
MOVIEID, trim(regexp_substr(Genres,''[^|]+'', LEVEL))

   FROM MOVIES

   CONNECT BY instr(regexp_substr(Genres,''[^|]+'', LEVEL), ''|'', 1, level -1)>0';

   EXECUTE IMMEDIATE sqlstring;

   sqlstring:= 'MERGE INTO MoviesGenreTable USING GenreTable ON
(MoviesGenreTable.Genres=GenreTable.Genre) WHEN MATCHED THEN UPDATE SET
MoviesGenreTable.GenreID=GenreTable.GenreID';

   EXECUTE IMMEDIATE sqlstring;

   sqlstring := 'ALTER TABLE MoviesGenreTable DROP COLUMN Genres';

   EXECUTE IMMEDIATE sqlstring;

   COMMIT;

END;


DECLARE

   CURSOR MG IS

      SELECT*

      FROM MoviesGenreTable;

   MG2 MG%ROWTYPE;

BEGIN

   OPEN MG;

   LOOP

      FETCH MG INTO MG2;

      EXIT WHEN MG%NOTFOUND;

      DBMS_OUTPUT.PUT_LINE (' Movie ID: ' || MG2.MOVIEID || ' GenreID: ' ||
MG2.GenreID);

   END LOOP;

   CLOSE MG;

END;
```

=======================Queris================================

```
DECLARE
    sqlstring varchar2(1000);
    sqlquan number;
BEGIN
    select count(gender),gender
    INTO sqlquan, sqlstring
    from usertable
    WHERE GENDER='M'
    group by gender;
    DBMS_OUTPUT.PUT_LINE('NUMBER OF ' || sqlstring || ' is ' || sqlquan);
        select count(gender),gender
    INTO sqlquan, sqlstring
    from usertable
    WHERE GENDER='F'
    group by gender;
    DBMS_OUTPUT.PUT_LINE('NUMBER OF ' || sqlstring || ' is ' || sqlquan);
END;
```