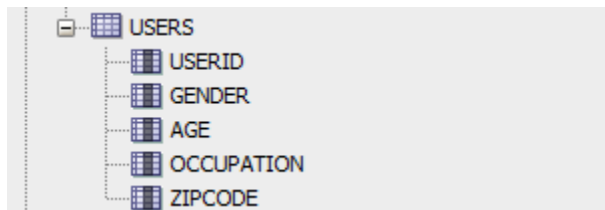
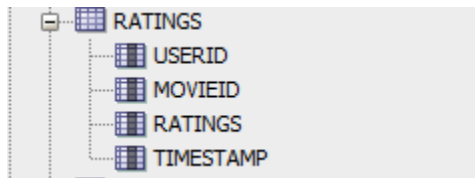
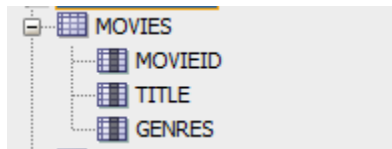


Homework #4

CSC 452

Zehong Zhuang

Table & Variable Names



Age_Info Procedure

```
3 CREATE OR REPLACE PROCEDURE Age_Info
4 AS
5     sqlstring VARCHAR2 (1000);
6 BEGIN
7     BEGIN
8         EXECUTE IMMEDIATE 'DROP TABLE AgeTable';
9     EXCEPTION
10        WHEN OTHERS THEN
11            IF SQLCODE != -942 THEN
12                RAISE;
13            END IF;
14    END;
15    sqlstring := 'CREATE TABLE AgeTable(
16    AgeNum NUMBER PRIMARY KEY,
17    AgeFact VARCHAR2 (30))';
18    EXECUTE IMMEDIATE sqlstring;
19    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (1, ''Under 18'')';
20    EXECUTE IMMEDIATE sqlstring;
21    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (18, ''1-24'')';
22    EXECUTE IMMEDIATE sqlstring;
23    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (25, ''25-34'')';
24    EXECUTE IMMEDIATE sqlstring;
25    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (35, ''35-44'')';
26    EXECUTE IMMEDIATE sqlstring;
27    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (45, ''45-49'')';
28    EXECUTE IMMEDIATE sqlstring;
29    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (50, ''50-55'')';
30    EXECUTE IMMEDIATE sqlstring;
31    sqlstring := 'INSERT INTO AgeTable (AgeNum, AgeFact) VALUES (56, ''56+'')';
32    EXECUTE IMMEDIATE sqlstring;
33    COMMIT;
34 END Age_Info;
--
37 DECLARE
38     CURSOR AGE IS
39         SELECT*
40         FROM AgeTable;
41     ATable Age%ROWTYPE;
42 BEGIN
43     OPEN AGE;
44     LOOP
45         FETCH Age INTO ATable;
46         EXIT WHEN Age%NOTFOUND;
47         DBMS_OUTPUT.PUT_LINE('Age Code: ' || ATable.AgeNum || ' Age Description: ' || ATable.AgeFact);
48     END LOOP;
49     CLOSE AGE;
50 END;
```

Script Output x Explain Plan x Query Result x Query Result 1 x

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

```
53 CREATE OR REPLACE PROCEDURE Occupation_Info
54 AS
55     sqlstring varchar2 (1000);
56 BEGIN
57     BEGIN
58         EXECUTE IMMEDIATE 'DROP TABLE OccupationTable';
59     EXCEPTION
60         WHEN OTHERS THEN
61             IF SQLCODE != -942 THEN
62                 RAISE;
63             END IF;
64     END;
65     sqlstring := 'CREATE TABLE OccupationTable(
66     OccupationID NUMBER PRIMARY KEY,
67     JobDescription VARCHAR2(30))';
68     EXECUTE IMMEDIATE sqlstring;
69     sqlstring := 'INSERT INTO OccupationTable VALUES (0, ''other'')';
70     EXECUTE IMMEDIATE sqlstring;
71     sqlstring := 'INSERT INTO OccupationTable VALUES (1, ''academic/educator'')';
72     EXECUTE IMMEDIATE sqlstring;
73     sqlstring := 'INSERT INTO OccupationTable VALUES (2, ''artist'')';
74     EXECUTE IMMEDIATE sqlstring;
75     sqlstring := 'INSERT INTO OccupationTable VALUES (3, ''clerical/admin'')';
76     EXECUTE IMMEDIATE sqlstring;
77     sqlstring := 'INSERT INTO OccupationTable VALUES (4, ''college/grad student'')';
78     EXECUTE IMMEDIATE sqlstring;
79     sqlstring := 'INSERT INTO OccupationTable VALUES (5, ''customer service'')';
80     EXECUTE IMMEDIATE sqlstring;
81     sqlstring := 'INSERT INTO OccupationTable VALUES (6, ''doctor/health care'')';
82     EXECUTE IMMEDIATE sqlstring;
83     sqlstring := 'INSERT INTO OccupationTable VALUES (7, ''executiv/managerial'')';
84     EXECUTE IMMEDIATE sqlstring;
85     sqlstring := 'INSERT INTO OccupationTable VALUES (8, ''farmer'')';
86     EXECUTE IMMEDIATE sqlstring;
87     sqlstring := 'INSERT INTO OccupationTable VALUES (9, ''homemaker'')';
88     EXECUTE IMMEDIATE sqlstring;
89     sqlstring := 'INSERT INTO OccupationTable VALUES (10, ''K-12 Student'')';
90     EXECUTE IMMEDIATE sqlstring;
91     sqlstring := 'INSERT INTO OccupationTable VALUES (11, ''lawyer'')';
92     EXECUTE IMMEDIATE sqlstring;
93     sqlstring := 'INSERT INTO OccupationTable VALUES (12, ''programmer'')';
94     EXECUTE IMMEDIATE sqlstring;
95     sqlstring := 'INSERT INTO OccupationTable VALUES (13, ''rtired'')';
96     EXECUTE IMMEDIATE sqlstring;
97     sqlstring := 'INSERT INTO OccupationTable VALUES (14, ''sales/marketing'')';
98     EXECUTE IMMEDIATE sqlstring;
99     sqlstring := 'INSERT INTO OccupationTable VALUES (15, ''scientist'')';
100    EXECUTE IMMEDIATE sqlstring;
101    sqlstring := 'INSERT INTO OccupationTable VALUES (16, ''self-employed'')';
102    EXECUTE IMMEDIATE sqlstring;
103    sqlstring := 'INSERT INTO OccupationTable VALUES (17, ''technician/engineer'')';
104    EXECUTE IMMEDIATE sqlstring;
105    sqlstring := 'INSERT INTO OccupationTable VALUES (18, ''tradesman/craftsman'')';
106    EXECUTE IMMEDIATE sqlstring;
107    sqlstring := 'INSERT INTO OccupationTable VALUES (19, ''unemployed'')';
108    EXECUTE IMMEDIATE sqlstring;
109    sqlstring := 'INSERT INTO OccupationTable VALUES (20, ''writer'')';
110    EXECUTE IMMEDIATE sqlstring;
111 END;

113 DECLARE
114     CURSOR OI IS
115         SELECT*
116         FROM OccupationTable;
117     OI2 OI%ROWTYPE;
118 BEGIN
119     OPEN OI;
120     LOOP
121         FETCH OI INTO OI2;
122         EXIT WHEN OI%NOTFOUND;
123         DBMS_OUTPUT.PUT_LINE('Occupation Number: ' || OI2.OccupationID || ' Job Description: ' || OI2.JobDescription);
124     END LOOP;
125     CLOSE OI;
126 END;
127
```

Script Output x Explain Plan x Query Result x Query Result 1 x

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

```
184 DECLARE
185     CURSOR GI IS
186         SELECT*
187         FROM GenreTable;
188     GI2 GI%ROWTYPE;
189 BEGIN
190     OPEN GI;
191     LOOP
192         FETCH GI INTO GI2;
193         EXIT WHEN GI%NOTFOUND;
194         DBMS_OUTPUT.PUT_LINE('Genre ID: ' || GI2.GenreID || ' Genre: ' || GI2.Genre);
195     END LOOP;
196     CLOSE GI;
197 END;
```

Script Output x Explain Plan x Query Result x Query Result 1 x

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.

```
129 CREATE OR REPLACE PROCEDURE Genre_Info
130 AS
131     sqlstring varchar2 (1000);
132 BEGIN
133     BEGIN
134         EXECUTE IMMEDIATE 'DROP TABLE GenreTable';
135     EXCEPTION
136         WHEN OTHERS THEN
137             IF SQLCODE != -942 THEN
138                 RAISE;
139             END IF;
140     END;
141     sqlstring := 'CREATE TABLE GenreTable (
142         GenreID NUMBER PRIMARY KEY,
143         Genre VARCHAR2 (20))';
144     EXECUTE IMMEDIATE sqlstring;
145     sqlstring := 'INSERT INTO GenreTable VALUES (1, ''Action'')';
146     EXECUTE IMMEDIATE sqlstring;
147     sqlstring := 'INSERT INTO GenreTable VALUES (2, ''Adventure'')';
148     EXECUTE IMMEDIATE sqlstring;
149     sqlstring := 'INSERT INTO GenreTable VALUES (3, ''Animation'')';
150     EXECUTE IMMEDIATE sqlstring;
151     sqlstring := 'INSERT INTO GenreTable VALUES (4, ''Childrens'')';
152     EXECUTE IMMEDIATE sqlstring;
153     sqlstring := 'INSERT INTO GenreTable VALUES (5, ''Comedy'')';
154     EXECUTE IMMEDIATE sqlstring;
155     sqlstring := 'INSERT INTO GenreTable VALUES (6, ''Crime'')';
156     EXECUTE IMMEDIATE sqlstring;
157     sqlstring := 'INSERT INTO GenreTable VALUES (7, ''Documentary'')';
158     EXECUTE IMMEDIATE sqlstring;
159     sqlstring := 'INSERT INTO GenreTable VALUES (8, ''Drama'')';
160     EXECUTE IMMEDIATE sqlstring;
161     sqlstring := 'INSERT INTO GenreTable VALUES (9, ''Fantasy'')';
162     EXECUTE IMMEDIATE sqlstring;
163     sqlstring := 'INSERT INTO GenreTable VALUES (10, ''Film-Noir'')';
164     EXECUTE IMMEDIATE sqlstring;
165     sqlstring := 'INSERT INTO GenreTable VALUES (11, ''Horror'')';
166     EXECUTE IMMEDIATE sqlstring;
167     sqlstring := 'INSERT INTO GenreTable VALUES (12, ''Musical'')';
168     EXECUTE IMMEDIATE sqlstring;
169     sqlstring := 'INSERT INTO GenreTable VALUES (13, ''Mystery'')';
170     EXECUTE IMMEDIATE sqlstring;
171     sqlstring := 'INSERT INTO GenreTable VALUES (14, ''Romance'')';
172     EXECUTE IMMEDIATE sqlstring;
173     sqlstring := 'INSERT INTO GenreTable VALUES (15, ''Sci-Fi'')';
174     EXECUTE IMMEDIATE sqlstring;
175     sqlstring := 'INSERT INTO GenreTable VALUES (16, ''Thriller'')';
176     EXECUTE IMMEDIATE sqlstring;
177     sqlstring := 'INSERT INTO GenreTable VALUES (17, ''War'')';
178     EXECUTE IMMEDIATE sqlstring;
179     sqlstring := 'INSERT INTO GenreTable VALUES (18, ''Western'')';
180     EXECUTE IMMEDIATE sqlstring;
181     COMMIT;
182 END;
```

User Info Procedure

```
201 CREATE OR REPLACE PROCEDURE User_Info
202 AS
203     sqlstring varchar2 (1000);
204 BEGIN
205     BEGIN
206         EXECUTE IMMEDIATE 'DROP TABLE UserTable';
207     EXCEPTION
208         WHEN OTHERS THEN
209             IF SQLCODE != -942 THEN
210                 RAISE;
211             END IF;
212     END;
213     sqlstring := 'CREATE TABLE userTable(
214         UserID NUMBER PRIMARY KEY,
215         Gender varchar2 (5),
216         AgeCode NUMBER,
217         OccupationCode NUMBER,
218         ZipCode VARCHAR2 (20))';
219     EXECUTE IMMEDIATE sqlstring;
220     sqlstring := 'INSERT INTO UserTable SELECT* FROM Users';
221     EXECUTE IMMEDIATE sqlstring;
222     COMMIT;
223 END User_Info;
```

```
225 DECLARE
226     CURSOR US IS
227         SELECT*FROM UserTable;
228     US2 US%ROWTYPE;
229 BEGIN
230     OPEN US;
231     LOOP
232         FETCH US INTO US2;
233         EXIT WHEN US%NOTFOUND;
234         DBMS_OUTPUT.PUT_LINE ('User ID: ' || US2.UserID || ' Gender: ' || US2.Gender || ' AgeCode: ' || US2.AgeCode || ' Occupation: ' || US2.OccupationCode || ' ZipCode: ' || US2.ZipCode);
235     END LOOP;
236     CLOSE US;
237 END;
```

Script Output x Explain Plan x Query Result x Query Result 1 x

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

```
240 CREATE OR REPLACE PROCEDURE Ratings_Info
241 AS
242     sqlstring varchar2 (1000);
243 BEGIN
244     BEGIN
245         EXECUTE IMMEDIATE 'DROP TABLE RatingsTable';
246     EXCEPTION
247         WHEN OTHERS THEN
248             IF SQLCODE != -942 THEN
249                 RAISE;
250             END IF;
251     END;
252     sqlstring := 'CREATE TABLE RatingsTable (
253     UserID NUMBER,
254     MovieID NUMBER,
255     Ratings NUMBER,
256     Timestamp VARCHAR2 (50))';
257     EXECUTE IMMEDIATE sqlstring;
258     sqlstring := 'INSERT INTO RatingsTable SELECT* FROM Ratings';
259     EXECUTE IMMEDIATE sqlstring;
260     COMMIT;
261 END;
```

```
263 DECLARE
264     CURSOR RT IS
265     SELECT*
266     FROM RatingsTable;
267     RT2 RT%ROWTYPE;
268 BEGIN
269     OPEN RT;
270     LOOP
271         FETCH RT INTO RT2;
272         EXIT WHEN RT%NOTFOUND;
273         DBMS_OUTPUT.PUT_LINE('User ID: ' || RT2.UserID || ' Movie ID: ' || RT2.MOVIEID || ' Ratings: ' || RT2.Ratings || ' Timestamp: ' || RT2.Timestamp);
274     END LOOP;
275     CLOSE RT;
276 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

```
279 CREATE OR REPLACE PROCEDURE Movies_Info
280 AS
281     sqlstring varchar2 (1000);
282 BEGIN
283     BEGIN
284         EXECUTE IMMEDIATE 'DROP TABLE MoviesTable';
285     EXCEPTION
286         WHEN OTHERS THEN
287             IF SQLCODE != -942 THEN
288                 RAISE;
289             END IF;
290     END;
291     sqlstring := 'CREATE TABLE MoviesTable(
292     MovieID NUMBER PRIMARY KEY,
293     Title VARCHAR2 (100),
294     Years VARCHAR2 (10))';
295     EXECUTE IMMEDIATE sqlstring;
296     sqlstring := 'INSERT INTO MoviesTable (MovieID, Title, Years) SELECT DISTINCT MOVIEID, SUBSTR(TITLE, 0, LENGTH(TITLE)-6), substr(Title, -6)
297     FROM MOVIES';
298     EXECUTE IMMEDIATE sqlstring;
299     COMMIT;
300 END;

302 DECLARE
303     CURSOR MT IS
304         SELECT*
305         FROM MOVIESTABLE;
306     MT2 MT%ROWTYPE;
307 BEGIN
308     OPEN MT;
309     LOOP
310         FETCH MT INTO MT2;
311         EXIT WHEN MT%NOTFOUND;
312         DBMS_OUTPUT.PUT_LINE('Movie ID: ' || MT2.MovieID || ' Title: ' || MT2.Title || ' Years: ' || MT2.Years);
313     END LOOP;
314     CLOSE MT;
315 END;
```

Script Output x Explain Plan x Query Result x Query Result 1 x

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

```
318 CREATE OR REPLACE PROCEDURE MoviesGenre_Info
319 AS
320     sqlstring varchar2 (1000);
321 BEGIN
322     BEGIN
323         EXECUTE IMMEDIATE 'DROP TABLE MoviesGenreTable';
324     EXCEPTION
325         WHEN OTHERS THEN
326             IF SQLCODE != -942 THEN
327                 RAISE;
328             END IF;
329     END;
330     sqlstring := 'CREATE TABLE MoviesGenreTable(
331     MovieID NUMBER PRIMARY KEY,
332     GenreID NUMBER,
333     Genres VARCHAR2 (50))';
334     EXECUTE IMMEDIATE sqlstring;
335     sqlstring := 'INSERT INTO MoviesGenreTable (MovieID, Genres) SELECT DISTINCT MOVIEID, trim(regexp_substr(Genres, '[^|]+' , LEVEL))
336     FROM MOVIES
337     CONNECT BY instr(regexp_substr(Genres, '[^|]+' , LEVEL), '|') = 1, level -1 > 0';
338     EXECUTE IMMEDIATE sqlstring;
339     sqlstring:= 'MERGE INTO MoviesGenreTable USING GenreTable ON (MoviesGenreTable.Genres=GenreTable.Genre) WHEN MATCHED THEN UPDATE SET MoviesGenreTable.GenreID=GenreTable.GenreID';
340     EXECUTE IMMEDIATE sqlstring;
341     sqlstring := 'ALTER TABLE MoviesGenreTable DROP COLUMN Genres';
342     EXECUTE IMMEDIATE sqlstring;
343     COMMIT;
344 END;
```

```
346 DECLARE
347     CURSOR MG IS
348         SELECT*
349         FROM MoviesGenreTable;
350     MG2 MG%ROWTYPE;
351 BEGIN
352     OPEN MG;
353     LOOP
354         FETCH MG INTO MG2;
355         EXIT WHEN MG%NOTFOUND;
356         DBMS_OUTPUT.PUT_LINE (' Movie ID: ' || MG2.MOVIEID || ' GenreID: ' || MG2.GenreID);
357     END LOOP;
358     CLOSE MG;
359 END;
```

Script Output x Explain Plan x Query Result x Query Result 1 x

Task completed in 0.203 seconds

```
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

```
361 DECLARE
362     sqlstring varchar2(1000);
363     sqlquan number;
364 BEGIN
365     select count(gender),gender
366     INTO sqlquan, sqlstring
367     from usertable
368     WHERE GENDER='M'
369     group by gender;
370     DBMS_OUTPUT.PUT_LINE('NUMBER OF ' || sqlstring || ' is ' || sqlquan);
371     select count(gender),gender
372     INTO sqlquan, sqlstring
373     from usertable
374     WHERE GENDER='F'
375     group by gender;
376     DBMS_OUTPUT.PUT_LINE('NUMBER OF ' || sqlstring || ' is ' || sqlquan);
377 END;
378
```

Script Output x Query Result x
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;

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(regex_substr(Genres,"[^|]+" , LEVEL))

FROM MOVIES

CONNECT BY instr(regex_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;