**Assignment 2a CSC 401**

**Name: Varunan Asokan**

**Part 1**

SQL> desc mm\_movie

Name Null? Type

------------------------------------------------------------------------------------------

MOVIE\_ID NOT NULL NUMBER(4)

MOVIE\_TITLE VARCHAR2(40)

MOVIE\_CAT\_ID NOT NULL NUMBER(2)

MOVIE\_VALUE NUMBER(5,2)

MOVIE\_QTY NUMBER(2)

SQL> select \* from mm\_movie;

MOVIE\_ID MOVIE\_TITLE MOVIE\_CAT\_ID MOVIE\_VALUE MOVIE\_QTY

------------------------------------------------------------------------------------------------------------------

1 Alien 1 10 5

2 Bladerunner 1 8 3

3 Star Wars 1 15 11

4 Texas Chainsaw Masacre 2 7 2

5 Jaws 2 7 1

6 The good, the bad and the ugly 3 7 2

7 Silverado 3 7 1

8 Duck Soup 4 5 1

9 Planes, trains and automobiles 4 5 3

10 Waking Ned Devine 4 12 4

11 Deep Blue Sea 5 14 3

12 The Fifth Element 5 15 5

12 rows selected.

**Altering table to include the STK\_FLAG column**

SQL> alter table mm\_movie add STK\_FLAG char(1);

Table altered

**Structure of table after adding the column**

SQL> desc mm\_movie

Name Null? Type

------------------------------------------------------------------------------------------- ------------------------

MOVIE\_ID NOT NULL NUMBER(4)

MOVIE\_TITLE VARCHAR2(40)

MOVIE\_CAT\_ID NOT NULL NUMBER(2)

MOVIE\_VALUE NUMBER(5,2)

MOVIE\_QTY NUMBER(2)

STK\_FLAG CHAR(1)

SQL> $notepad calculate.sql;

**Code inside Calculate.sql (Anonymous SQL block with Cursor For Loop)**

declare

cursor c1 is select \* from mm\_movie;

n number;

begin

dbms\_output.put\_line('hello');

for i in c1

loop

n:=i.movie\_value\*i.movie\_qty;

if n>=75 then

update mm\_movie set STK\_FLAG='\*' where movie\_id=i.movie\_id;

end if;

end loop;

end;

SQL> @calculate;

/

PL/SQL procedure successfully completed.

SQL> select \* from mm\_movie;

MOVIE\_ID MOVIE\_TITLE MOVIE\_CAT\_ID MOVIE\_VALUE MOVIE\_QTY STK\_FLAG

-------------------------------------------------------------------------------------------------------------------------

1 Alien 1 10 5

2 Bladerunner 1 8 3

3 Star Wars 1 15 11 \*

4 Texas Chainsaw Masacre 2 7 2

5 Jaws 2 7 1

6 The good, the bad and the ugly 3 7 2

7 Silverado 3 7 1

8 Duck Soup 4 5 1

9 Planes, trains and automobiles 4 5 3

10 Waking Ned Devine 4 12 4

11 Deep Blue Sea 5 14 3

12 The Fifth Element 5 15 5 \*

12 rows selected.

**Part 2**

Initial code block **without** any exception handler

SET SERVEROUTPUT ON

VARIABLE g\_movie\_id NUMBER

BEGIN

:g\_movie\_id := 4;

END;

/

DECLARE

v\_count NUMBER;

v\_title mm\_movie.movie\_title%TYPE;

BEGIN

SELECT m.movie\_title, COUNT(r.rental\_id)

INTO v\_title, v\_count

FROM mm\_movie m, mm\_rental r

WHERE m.movie\_id = r.movie\_id

AND m.movie\_id = :g\_movie\_id

GROUP BY m.movie\_title;

DBMS\_OUTPUT.PUT\_LINE(v\_title || ': ' || v\_count);

END;

/

**Output:**

Texas Chainsaw Masacre: 2

**Code block with exception handlers**

SET SERVEROUTPUT ON

VARIABLE g\_movie\_id char(3)

BEGIN

:g\_movie\_id := 'abc';

END;

/

DECLARE

v\_count NUMBER;

v\_title mm\_movie.movie\_title%TYPE;

BEGIN

SELECT m.movie\_title, COUNT(r.rental\_id)

INTO v\_title, v\_count

FROM mm\_movie m, mm\_rental r

WHERE m.movie\_id = r.movie\_id

AND m.movie\_id = :g\_movie\_id

GROUP BY m.movie\_title;

DBMS\_OUTPUT.PUT\_LINE(v\_title || ': ' || v\_count);

**EXCEPTION**

**WHEN INVALID\_NUMBER THEN**

**dbms\_output.put\_line('Invalid Input. Movie Id should be a number');**

**WHEN NO\_DATA\_FOUND THEN**

**dbms\_output.put\_line('Movie not found');**

**WHEN OTHERS THEN**

**dbms\_output.put\_line('Unexpected Error');**

end;

**Output:**

Invalid Input. Movie Id should be a number

**Another set of input**

SET SERVEROUTPUT ON

VARIABLE g\_movie\_id number

BEGIN

:g\_movie\_id := 98;

END;

/

DECLARE

v\_count NUMBER;

v\_title mm\_movie.movie\_title%TYPE;

BEGIN

SELECT m.movie\_title, COUNT(r.rental\_id)

INTO v\_title, v\_count

FROM mm\_movie m, mm\_rental r

WHERE m.movie\_id = r.movie\_id

AND m.movie\_id = :g\_movie\_id

GROUP BY m.movie\_title;

DBMS\_OUTPUT.PUT\_LINE(v\_title || ': ' || v\_count);

EXCEPTION

WHEN INVALID\_NUMBER THEN

dbms\_output.put\_line('Invalid Input. Movie Id should be a number');

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Movie not found');

WHEN OTHERS THEN

dbms\_output.put\_line('Unexpected Error');

end;

**Output**

Movie not found