

Islamic University of Technology

CSE-4308

Database Management Systems Lab

Lab-9 Report

Name : Mukit Mahdin

ID : 200042170

Department : CSE

Program : SWE

Group : B

Task - 1 :

Write PL/SQL statements to perform the following tasks

- a. Print Your Name.

```
SET SERVEROUTPUT ON SIZE 1000000
BEGIN
DBMS_OUTPUT . PUT_LINE ('Mukit Mahdin');
END ;
/
```

- b. Take your student ID as input and print its length

```
DECLARE
studentid VARCHAR2 (10);
BEGIN
studentid := '&studentid';
DBMS_OUTPUT . PUT_LINE (length(studentid));
END ;
/
```

- c. Take two numbers as input and print their product

```
DECLARE
num1 number;
num2 number;
BEGIN
num1 := '&num1';
num2 := '&num2';
DBMS_OUTPUT . PUT_LINE (num1*num2);
END ;
/
```

- d. Print the current system time in 12 hour format

```
DECLARE
D DATE := SYSDATE ;
BEGIN
DBMS_OUTPUT . PUT_LINE ( TO_CHAR (D, 'HH12 :MI:SS'));
END ;
/
```

- e. Take a number as input and print whether it is a whole number or a fraction (with and without CASE statement).

Without Case :

```
DECLARE
num1 number;
BEGIN
    num1 := '&num1';
    IF (num1-ROUND(num1) = 0) THEN
        DBMS_OUTPUT.PUT_LINE ('The number is a whole number');
    ELSE
        DBMS_OUTPUT.PUT_LINE ('The number is a fraction');
    END IF;
END ;
/
```

With Case :

```
DECLARE
num1 number;
BEGIN
    num1 := '&num1';
    CASE num1-ROUND(num1)
        WHEN 0 THEN
            DBMS_OUTPUT.PUT_LINE ('The number is a whole number');
        ELSE
            DBMS_OUTPUT.PUT_LINE ('The number is a fraction');
    END CASE ;
END ;
/
```

- f. Write a procedure that takes a number as argument and prints whether it is a composite number or not.

```
CREATE OR REPLACE
PROCEDURE FIND_COMPOSITE ( NUM IN NUMBER , RESULT OUT VARCHAR2)
AS
BEGIN
    RESULT := 'Number is composite';
    FOR i IN 2 ..NUM
        LOOP
            IF MOD(NUM, i) != 0 THEN
```

```

                                RESULT := 'Number is not composite';
                                EXIT;
                            END IF;
                        END LOOP;
END;
/

```

Calling the procedure :

```

DECLARE
RESULT VARCHAR2(100);
num number;
BEGIN
num := '&num';
FIND_COMPOSITE(num,RESULT);
DBMS_OUTPUT . PUT_LINE (RESULT);
END;
/

```

Description:

- Task-a is done only using the right Syntax to give output
- In Task-b,I found out the length using the built-in length() function
- Task-c is a simple multiplication using proper PL/SQL syntax
- Took Sysdate in a variable and printed that in a 12-hour format
- Used round() function to convert unrounded number and after performing subtraction from the rounded number if we get 0, the number will be a whole number otherwise it will be a fractional number

Task-2 :

(a) Write a procedure to find the N top-rated movies and their details. The procedure will take N as input and print the details up to N movies. If N is greater than the number of movies, then it will print an error message.

```
create or replace PROCEDURE findmovie(num in number)
AS
rownums number(10);
cursor movieinfo is
    select *
    from (select mov_id,mov_title,max(mov_year) as mov_year,max(mov_language) as
mov_language,max(mov_releasedate) as mov_releasedate,max(country) as country,
avg(rev_stars) as average_rating
from movie natural join rating
group by mov_id,mov_title
order by average_rating desc)
where rownums<=num;
BEGIN
    select max(rownums) into rownums
    from (select mov_id,mov_title,max(mov_year) as mov_year,max(mov_language) as
mov_language,max(mov_releasedate) as mov_releasedate,max(country) as country,
avg(rev_stars) as average_rating
from movie natural join rating
group by mov_id,mov_title
order by average_rating desc);

    case true
    when(num>rownums) then
        DBMS_OUTPUT.PUT_LINE('Out of bound')
    else
        for i in movieinfo LOOP
            DBMS_OUTPUT.PUT_LINE(i.mov_title||chr(9)||i.mov_year||chr(9)||i.mov_l
            anguage||chr(9)||i.mov_releasedate||chr(9)||i.country);
        end LOOP;
    end case;
end;
/
DECLARE
num number;
BEGIN
    num:='&number';
    findmovie(num);
end;
/
```

(b) Write a function to find the movie status ("Solo", "Ensemble"). If the total number of actors/actresses in a movie is 1, then the status should be "Solo", else it should be "Ensemble". The function will take the title of the movie as input and return the status.

```
create or replace
PROCEDURE insert_status(movietitle in VARCHAR2)
as
cursor movieinfo is
    select mov_id,mov_title,count(act_id) as numberofactors
    from movie natural join casts
    where mov_title=movietitle;

BEGIN
    for i in movieinfo LOOP
        case true
            when(i.numberofactors<=1) then
                DBMS_OUTPUT.PUT_LINE('solo');
                DBMS_OUTPUT.PUT_LINE(i.numberofactors);
            else
                DBMS_OUTPUT.PUT_LINE("ensemble");
                DBMS_OUTPUT.PUT_LINE(i.numberofactors);
            end case;
        end LOOP;
    end;
/

DECLARE
    movietitle VARCHAR2;
BEGIN
    movietitle := '&title';
    insert_status(movietitle);
end;
/
```

(c) Write a procedure to find the possible nominees for the Oscars. A director is eligible for Oscar if at least one of their movies has an average rating of at least 7. Also, the movie should be reviewed by more than 10 reviewers.

```
create or replace
PROCEDURE oscarwinner
as
cursor movieinfo is
    select dir_id,max(dir_firstname) as dir_firstname,max(dir_lastname) as dir_lastname
```

```

        from (select mov_id,mov_title,avg(rev_stars) as averagerating
        count(rev_stars) as reviews
        from movie natural join rating
        group by mov_id,mov_title
        order by average_rating desc) natural join director
        where average_rating >=7 and reviews>=10
        group by dir_id
        order by dir_id;
BEGIN
    for i in movieinfo loop
        DBMS_OUTPUT.PUT_LINE(i.dir_id|| chr(9) || i.dir_firstname || chr(9)
||i.dir_lastname);
    end loop;
end;
/
BEGIN
    oscarwinner();
end;
/

```

(d) Write a function that will take the title of the movie as input and find the movie category based on Table 1.

```

create or replace
PROCEDURE moviecatagory(movie_title in VARCHAR2)
as
year VARCHAR2(20);
cursor movieinfo is
    select mov_id,mov_title,max(mov_releasedate) as mov_releasedate,avg(rev_stars)as rating
    from movie natural join rating
    where mov_title=movie_title
    group by mov_id,mov_title
    order by average_rating desc;

BEGIN
    for i in movieinfo loop
        year:=TO_CHAR(i.mov_releasedate,'YYYY');
        if i.rating>6.5 and year>=1950 and year<=1959 then
            DBMS_OUTPUT.PUT_LINE('Fantastic Fifties');
        elsif i.rating>6.7 and year>=1960 and year<=1969 then
            DBMS_OUTPUT.PUT_LINE('Sweet Sixties');
        elsif i.rating>6.9 and year>=1970 and year<=1979 then
            DBMS_OUTPUT.PUT_LINE('Super seventies');
        elsif i.rating>7.1 and year>=1980 and year<=1989 then
            DBMS_OUTPUT.PUT_LINE('Elastic Eighties');
        elsif i.rating>7.3 and year>=1990 and year<=1999 then

```

```
        DBMS_OUTPUT.PUT_LINE('Neat Nineties');
    else
        DBMS_OUTPUT.PUT_LINE('Garbage');
    end if;
end loop;
end;
/
```

```
DECLARE
    movie_title VARCHAR2(20);
BEGIN
    movie_title:='&title';
    moviecatagory(title);
end;
/
```