**Zhumagali Kanat IT2-2003**

**Lab4. Creating Procedures**

1. Create at least 4 procedures for your database. You may use the anonymous blocks you created in Lab 2 as a basis.
2. At least 3 procedures must be with parameters.
3. During the defense you must show the execution of the procedures.

create or replace procedure print\_movies is

begin

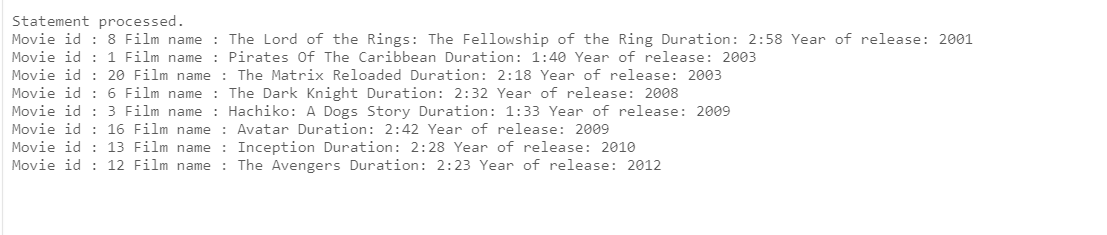
for sal\_record in (select movie\_id, film\_name, duration, year\_of\_release from movie where YEAR\_OF\_RELEASE > 2000 order by year\_of\_release asc) loop

dbms\_output.put\_line('Movie id : ' || sal\_record.movie\_id || ' Film name : ' || sal\_record.film\_name || ' Duration: ' || sal\_record.duration || ' Year of release: ' || sal\_record.year\_of\_release);

end loop;

end;

execute print\_movies;



create or replace procedure mov\_id (

m\_id in movie.movie\_id%type,

m\_name out movie.film\_name%type,

m\_duration out movie.duration%type,

m\_year out movie.year\_of\_release%type

) is

begin

select film\_name, duration, year\_of\_release into m\_name, m\_duration, m\_year from movie where movie\_id = m\_id;

end mov\_id;

declare

m\_name movie.film\_name%type;

m\_duration movie.duration%type;

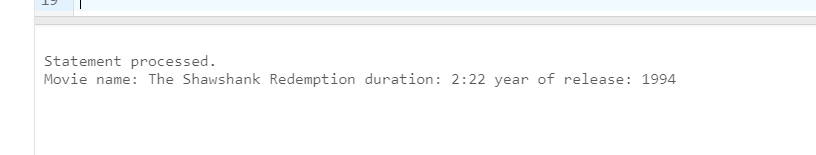
m\_year movie.year\_of\_release%type;

begin

mov\_id(4, m\_name,m\_duration,m\_year);

dbms\_output.put\_line('Movie name: ' || m\_name || ' duration: ' || m\_duration || ' year of release: ' || m\_year);

end;



create or replace fetch\_genre\_movies (my\_genre in genre.genre\_name%type ) IS

cursor genre\_cursor is select genre\_id, genre\_name from genre where genre\_name = my\_genre;

cursor movie\_g\_info(g\_id NUMBER) IS

select m.movie\_id, m.film\_name, m.year\_of\_release from movie m join movie\_genre mg on mg.movie\_id = m.movie\_id where mg.genre\_id = g\_id;

genre\_rec genre\_cursor%ROWTYPE;

movie\_genre\_info\_rec movie\_g\_info%ROWTYPE;

begin

for genre\_rec in genre\_cursor loop

DBMS\_OUTPUT.PUT\_LINE(genre\_rec.genre\_id || ' Genre name: ' || genre\_rec.genre\_name);

foor movie\_genre\_info\_rec IN movie\_g\_info(genre\_rec.genre\_id) LOOP

DBMS\_OUTPUT.PUT\_LINE(

'Movie id: ' || movie\_genre\_info\_rec.movie\_id ||

' Film name: ' || movie\_genre\_info\_rec.film\_name ||

' Year of release: ' || movie\_genre\_info\_rec.year\_of\_release

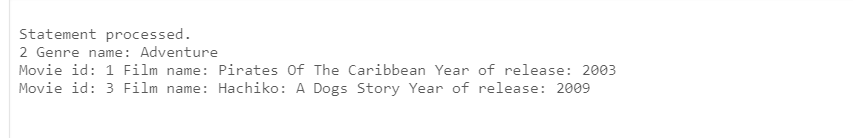
);

end loop;

end loop;

end fetch\_genre\_movies;

execute fetch\_genre\_movies('Adventure')



create or replace procedure get\_bb(my\_b in finance.budget%type) is

type movie\_fin\_info is record(

movie\_id movie.movie\_id%type,

film\_name movie.film\_name%type,

budget finance.budget%type,

income finance.income%type

);

m\_info movie\_fin\_info;

cursor fin\_cursor is

select m.movie\_id, m.film\_name, f.budget, f.income

from movie\_fin mf

join movie m on mf.movie\_id = m.movie\_id

join finance f on mf.fin\_id = f.fin\_id;

begin

for fin\_record in fin\_cursor loop

m\_info := fin\_record;

if to\_number(m\_info.budget) > my\_b then

dbms\_output.put\_line('Film name: ' || m\_info.film\_name ||

' has a budget: ' || m\_info.budget ||

' and its income: ' || m\_info.income);

else

dbms\_output.put\_line(m\_info.film\_name ||

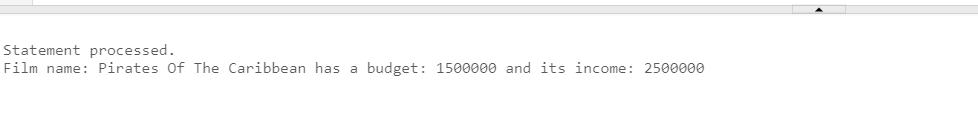
' has a budget less than ' || my\_b);

end if;

end loop;

end;

execute get\_bb(15000);



Questions:

1. Define a procedure in PL/SQL
2. Describe the functionality of procedures
3. What types of parameters exist?
4. Explain the functionality of IN parameters
5. Explain the functionality of OUT parameters
6. Explain the functionality of IN OUT parameters
7. What are formal parameters?
8. What are actual parameters?
9. What types of notations for passing actual parameters exist in PL/SQL?
10. Describe positional notation for passing actual parameters
11. Describe named notation for passing actual parameters
12. Describe mixed notation for passing actual parameters