Stored Procedures and Functions

1. ***Procedures:***

**/\*EXE1\*/**

CREATE OR REPLACE PROCEDURE Remove\_pilot(v\_pilot in pilot.nopilot%type) IS

BEGIN

DELETE FROM pilot

WHERE nopilot=v\_pilot;

COMMIT;

END Remove\_pilot;

1. Accept nopilot prompt ‘please enter the number of pilot to be deleted:’

EXECUTE Remove\_pilot (&nopilot) ; ---3465

1. SELECT \* FROM pilot;

**/\*EXE2\*/**

create or replace procedure comm\_pilot

(v\_pilot IN pilot.nopilot%type, txcomm IN float)

is

v\_comm Pilot.comm%type;

begin

select comm

Into v\_comm

From pilot

Where nopilot=v\_pilot;

DBMS\_OUTPUT.PUT\_LINE(‘The commission value before update: ’||v\_comm);

If v\_comm is null then

DBMS\_OUTPUT.PUT\_LINE(‘Commission is null’);

Else

Update pilot

Set comm=comm\*txcomm

Where nopilot= v\_pilot;

select comm

Into v\_comm

From pilot

Where nopilot=v\_pilot;

DBMS\_OUTPUT.PUT\_LINE(‘The commission value after update : ’||v\_comm);

End if;

COMMIT;

End comm\_pilot;

1. EXECUTE comm\_pilot(’6589’, 1.1) ;

/\*2nd solution\*/

---THE RATE ENTERED BY THE USER

Accept txtrate prompt ‘please enter the rate:’

EXECUTE comm\_pilot(’6589’, &txtrate) ;

1. EXECUTE comm\_pilot('3452', 1.2) ;

/\*2nd solution\*/

---THE RATE AND YHE NUMBER OF THE PILOT ARE ENTERED BY THE USER

Accept txtrate prompt ‘please enter the rate:’

Accept nopilot prompt ‘please enter the number of the pilot:’

EXECUTE comm\_pilot(&nopilot, &txtrate) ;

1. SELECT \* FROM pilot;

**/\*EXE3\*/**

select text from user\_source where name = 'REMOVE\_PILOT';

**/\*EXE4\*/**

select \* from user\_objects;

**/\*EXE5\*/**

select object\_name from user\_objects where object\_type = 'TABLE';

**/\*EXE6\*/**

select object\_name from user\_objects where object\_type = 'PROCEDURE';

**/\*EXE7\*/**

select object\_name from all\_objects where object\_type = 'PROCEDURE';

1. ***Functions:***

**/\*EX1\*/**

CREATE OR REPLACE FUNCTION max\_hour\_flight\_type (v\_type Device.codetype%type)

RETURN NUMBER IS

V\_nbhflight plane.nbhflight%type;

BEGIN

Select max(nbhflight)

Into V\_ nbhflight

FROM plane

WHERE type = v\_type;

RETURN(V\_ nbhflight);

END max\_hour\_flight\_type;

/

1. variable Max\_nbhflight NUMBER;
2. execute :Max\_nbhflight := max\_hour\_flight\_type ('734');
3. print Max\_nbhflight;

/\*2nd solution\*/

codetype entered by the user

ACCEPT CODETYPE PROMPT ‘PLEASE ENTER THE TYPE OF THE PLANE :’

variable Max\_nbhflight NUMBER;

execute :Max\_nbhflight := max\_hour\_flight\_type (&CODETYPE);

print Max\_nbhflight;

**/\*EX2\*/**

CREATE OR REPLACE FUNCTION pilot\_plane\_driven (v\_nopilot pilot.nopilot%type)

RETURN NUMBER IS

Number\_plane integer;

BEGIN

Select COUNT(Distinct plane)

Into Number\_plane

FROM assignation

Where pilot= v\_nopilot;

RETURN(Number\_plane);

END pilot\_plane\_driven;

/

1. variable nbrplane number;
2. execute :nbrplane:= pilot\_plane\_driven('6723'); --&v\_nopilot
3. print nbrplane;

**/\*EX3\*/**

select object\_name from all\_objects where object\_type = 'FUNCTION';

**/\*EX4\*/**

select object\_name from user\_objects where object\_type = 'FUNCTION';

1. ***Triggers:***

**/\*EXE 1\*/**

CREATE OR REPLACE TRIGGER Insert\_pilot

BEFORE INSERT ON pilot

FOR EACH ROW

BEGIN

IF (:NEW.sal < 300.00)

THEN

:NEW.sal := 700.00;

END IF;

END;

/

**/\*EXE 2\*/**

CREATE OR REPLACE TRIGGER Hiring\_pilot

AFTER INSERT OR UPDATE OF hir\_date ON pilot

FOR EACH ROW

BEGIN

IF (:new. hir\_date > sysdate) THEN

RAISE\_APPLICATION\_ERROR(-20000, ‘The date is over the system date’);

END IF;

END;

/