**Queries (**Simran**)**

1. Sequences:

(to add departmentId into pp\_department table in new row using nextvalue)

---------------SEQ (using sequence to add departmentId)

CREATE SEQUENCE PP\_SEQ\_SIMRAN

START WITH 7007

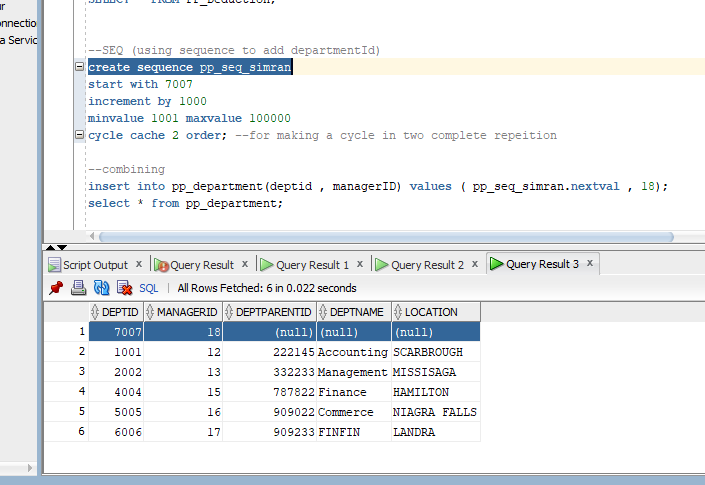
INCREMENT BY 1000

MINVALUE 1001 MAXVALUE 100000

CYCLE CACHE 2 ORDER; --FOR MAKING A CYCLE IN TWO COMPLETE REPEITION

-------------COMBINING

INSERT INTO PP\_DEPARTMENT (DEPTID , MANAGERID) VALUES ( PP\_SEQ\_SIMRAN.NEXTVAL , 18);



2.) Views

(To check for payment having less than $500 bonus)

-------views

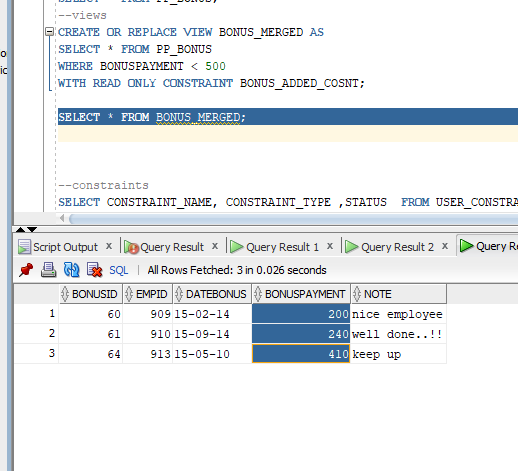
CREATE OR REPLACE VIEW BONUS\_MERGED AS

SELECT \* FROM PP\_BONUS

WHERE BONUSPAYMENT < 500

WITH READ ONLY CONSTRAINT BONUS\_ADDED\_COSNT;

SELECT \* FROM BONUS\_MERGED;



3. Viewing constraints in tables

SELECT CONSTRAINT\_NAME, CONSTRAINT\_TYPE ,STATUS FROM USER\_CONSTRAINTS WHERE TABLE\_NAME IN ('pp\_department');

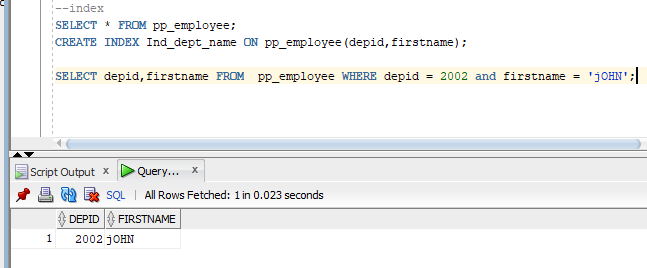
4. Indexes

(to identify the deptid by firstname from pp\_employee)

SELECT \* FROM PP\_EMPLOYEE;

CREATE INDEX IND\_DEPT\_NAME ON PP\_EMPLOYEE(DEPID,FIRSTNAME);

SELECT DEPID,FIRSTNAME FROM PP\_EMPLOYEE WHERE DEPID = 2002 AND FIRSTNAME = 'jOHN';



5. pl/sql command {anchor} (TO print payroll information using few variable)

SET SERVEROUTPUT ON;

DECLARE

LV\_PAYROLLID PP\_PAYROLL.PAYROLLID%TYPE;

LV\_EMPID PP\_PAYROLL.EMPID%TYPE;

LV\_GROSSPAY PP\_PAYROLL.GROSSPAY%TYPE;

BEGIN

SELECT PAYROLLID , EMPID, GROSSPAY

INTO LV\_PAYROLLID, LV\_EMPID,LV\_GROSSPAY

FROM PP\_PAYROLL WHERE SALARYID = '56';

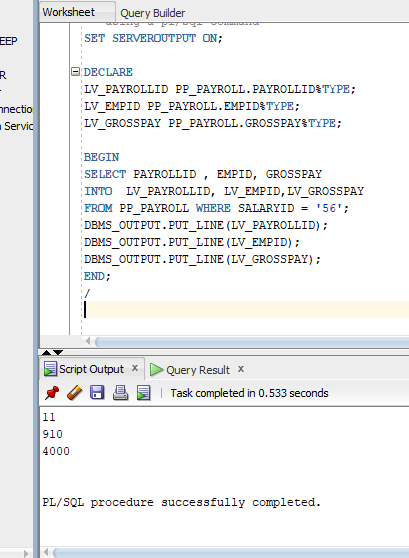
DBMS\_OUTPUT.PUT\_LINE(LV\_PAYROLLID);

DBMS\_OUTPUT.PUT\_LINE(LV\_EMPID);

DBMS\_OUTPUT.PUT\_LINE(LV\_GROSSPAY);

END;

/



6. Deleting and handling exception

SELECT \* FROM PP\_DEPARTMENT;

SET SERVEROUTPUT ON;

BEGIN

DELETE FROM PP\_DEPARTMENT WHERE DEPTID = 7007 ;

IF SQL%NOTFOUND THEN

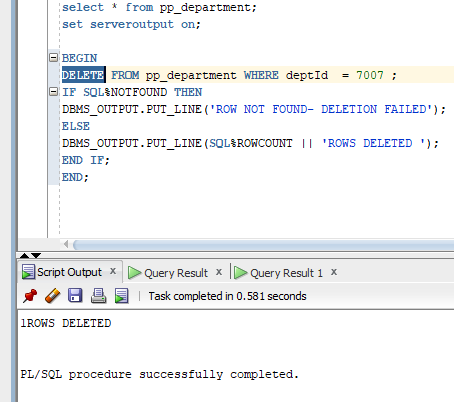
DBMS\_OUTPUT.PUT\_LINE('ROW NOT FOUND- DELETION FAILED');

ELSE

DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT || 'ROWS DELETED ');

END IF;

END;



7. Cursor {IMPLICIT METHOD}

SET SERVEROUTPUT ON;

DECLARE

LV\_PAYROLLID PP\_PAYROLL.PAYROLLID%TYPE;

LV\_EMPID PP\_PAYROLL.EMPID%TYPE;

LV\_GROSSPAY PP\_PAYROLL.GROSSPAY%TYPE;

CURSOR C12 IS

SELECT PAYROLLID , EMPID, GROSSPAY FROM PP\_PAYROLL;

BEGIN

OPEN C12;

LOOP

FETCH C12 INTO LV\_PAYROLLID, LV\_EMPID,LV\_GROSSPAY;

EXIT WHEN C12%NOTFOUND;

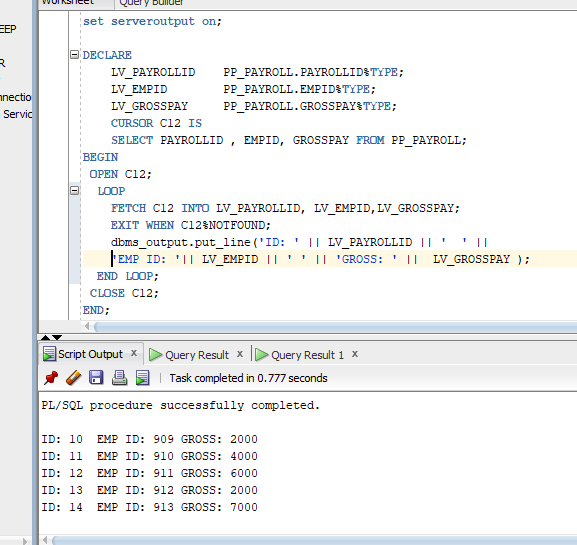
DBMS\_OUTPUT.PUT\_LINE('ID: ' || LV\_PAYROLLID || ' ' ||

'EMP ID: '|| LV\_EMPID || ' ' || 'GROSS: ' || LV\_GROSSPAY );

END LOOP;

CLOSE C12;

END;



8.) PROCEDURE

CREATE OR REPLACE PROCEDURE MY\_PAYROLL\_PRODEDURE IS --PROCEDURE

CURSOR C11 IS ---------------USING CURSOR

select \* FROM PP\_SALARY;

R\_CUR C11%ROWTYPE;

BEGIN

OPEN C11;

DBMS\_OUTPUT.PUT('CUSTOMER'||' '||'CITY');

LOOP

FETCH C11 INTO R\_CUR;

EXIT WHEN C11%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(R\_CUR.SALARYID||' '||R\_CUR.BASEYEAR);

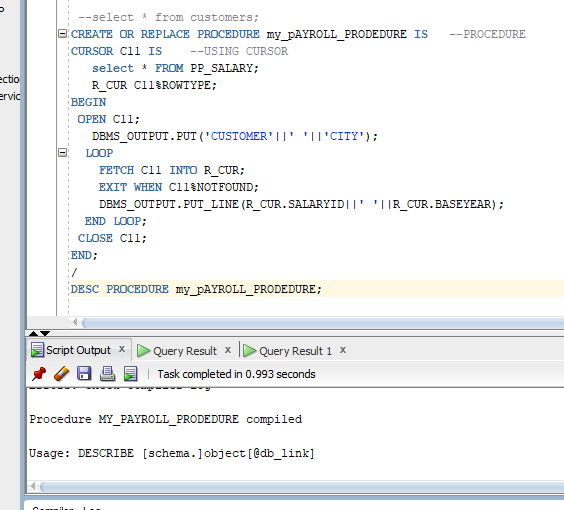
END LOOP;

CLOSE C11;

END;

/

DESC PROCEDURE my\_pAYROLL\_PRODEDURE;



9.) Package

10) Triggers