The School of Engineering Technology and Applied Science



**COMP214**

**Advanced Database Concepts**

**Term project**

**Submitted to:**

Professor Vijayalakshmi Tiruchengode Angamuthu (Viji)

Salary Management System

Submitted by:

**Kelly Tan(300991647) Simranjeet Kaur(300988063) Tariqule Hasan Khan(300883293)**

Table of Contents

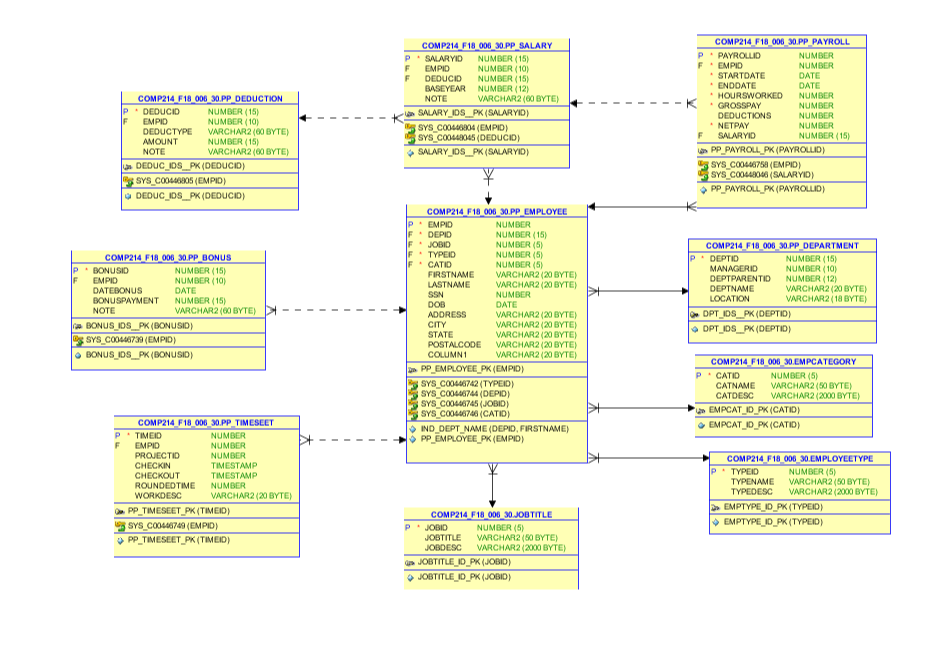
1 ER MODEL (Salary Management System)........................................................................................................................................................ 3

2 Business Rules.................................................................................................................................................................................................. 4

3 DESCRIPTION OF TABLES GENERATED!!!.......................................................................................................................................................... 4

4 SQL code for generating table and queries:!!!............................................................................................................................................... 24

**ER MODEL (Salary Management System)**



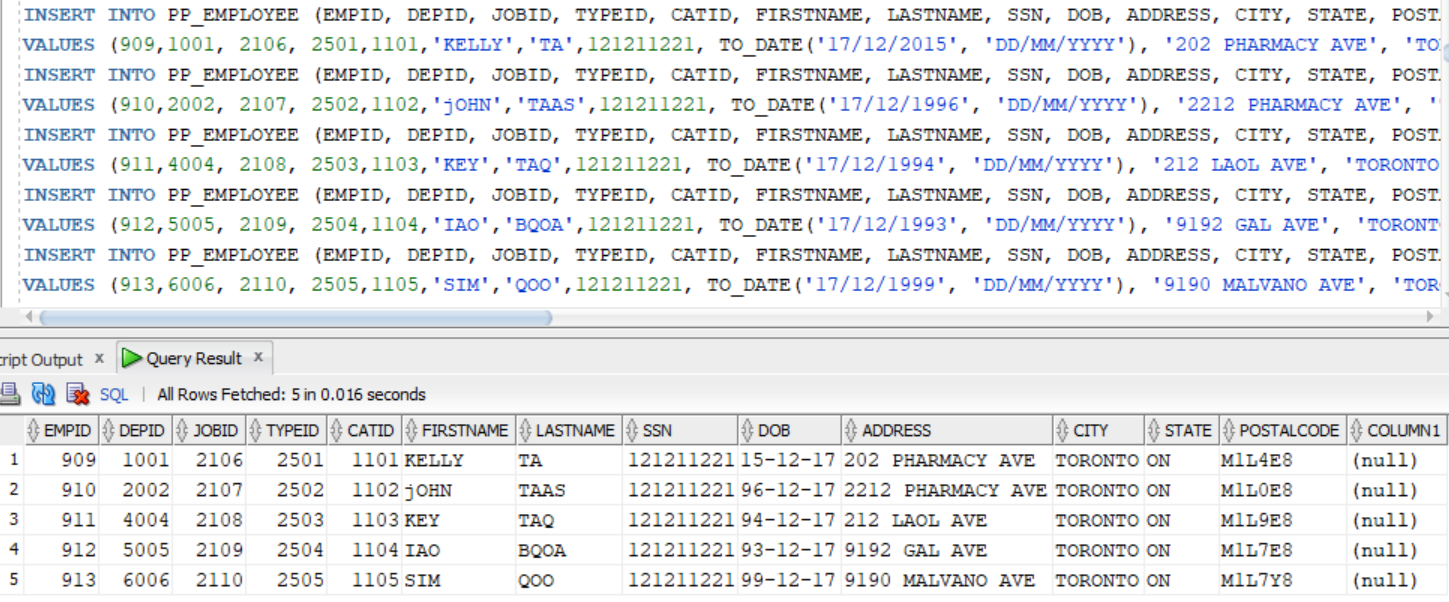
**BUSINESS RULES:**

* Each employee can receive one or more salaries and bonuses. Each particular salary belongs to one and only one employee. Each bonus belongs to one and only one employee.
* Each salary must receive one and only one deductions. Each deduction is associated with one or more salaries.
* Each salary belongs to one or more payrolls. Each payroll can receive one and only one salary and one employee. Each employee can receive one or more payrolls.
* Each employee can be associated with one or more time seeds. Each time seed is associated to one and only one employee.
* Each job title can be associated to one or more employees. Each job title is associated to one and only one employee.
* Each employee is in one and only one department. Each department can have one or more employee.
* Each employee belongs to one and only one employee category. Each employee category can have one or more employee.
* Each employee belongs to one and only one employee type. Each employee type can shave one or more employee.

**DESCRIPTION OF TABLES GENERATED!!!**

Table name – Employee (Tariq)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Key Type/  Constraint | Data Type | Filed length |
| user\_id | Used to login into the online system | Primary Key | number | 10 |
| First\_Name | First name of the user |  | Varchar2 | 15 |
| last\_name | Last name of the user |  | Varchar2 | 15 |
| gender | Male or female user |  | char | 1 |
| age | Age of the user |  | number | 3 |
| Mobile-number | Contact number if user |  | number | 10 |
| email | Contact mail of the user |  | Varchar2 | 50 |
| City | Locational city of the user |  | Varchar2 | 20 |
| Pin\_code | Postal code for the location |  | number | 6 |



CREATE TABLE PP\_EMPLOYEE

(

EMPID NUMBER NOT NULL

, DEPID NUMBER NOT NULL

, JOBID NUMBER NOT NULL

, TYPEID NUMBER NOT NULL

, CATID NUMBER NOT NULL

, FIRSTNAME VARCHAR2(20)

, LASTNAME VARCHAR2(20)

, SSN NUMBER

, DOB DATE

, ADDRESS VARCHAR2(20)

, CITY VARCHAR2(20)

, STATE VARCHAR2(20)

, POSTALCODE VARCHAR2(20)

, COLUMN1 VARCHAR2(20)

, CONSTRAINT PP\_EMPLOYEE\_PK PRIMARY KEY

(

EMPID

)

ENABLE

);

INSERT INTO PP\_EMPLOYEE (EMPID, DEPID, JOBID, TYPEID, CATID, FIRSTNAME, LASTNAME, SSN, DOB, ADDRESS, CITY, STATE, POSTALCODE)

VALUES (909,1001, 2106, 2501,1101,'KELLY','TA',121211221, TO\_DATE('17/12/2015', 'DD/MM/YYYY'), '202 PHARMACY AVE', 'TORONTO','ON','M1L4E8' );

INSERT INTO PP\_EMPLOYEE (EMPID, DEPID, JOBID, TYPEID, CATID, FIRSTNAME, LASTNAME, SSN, DOB, ADDRESS, CITY, STATE, POSTALCODE)

VALUES (910,2002, 2107, 2502,1102,'jOHN','TAAS',121211221, TO\_DATE('17/12/1996', 'DD/MM/YYYY'), '2212 PHARMACY AVE', 'TORONTO','ON','M1L0E8' );

INSERT INTO PP\_EMPLOYEE (EMPID, DEPID, JOBID, TYPEID, CATID, FIRSTNAME, LASTNAME, SSN, DOB, ADDRESS, CITY, STATE, POSTALCODE)

VALUES (911,4004, 2108, 2503,1103,'KEY','TAQ',121211221, TO\_DATE('17/12/1994', 'DD/MM/YYYY'), '212 LAOL AVE', 'TORONTO','ON','M1L9E8' );

INSERT INTO PP\_EMPLOYEE (EMPID, DEPID, JOBID, TYPEID, CATID, FIRSTNAME, LASTNAME, SSN, DOB, ADDRESS, CITY, STATE, POSTALCODE)

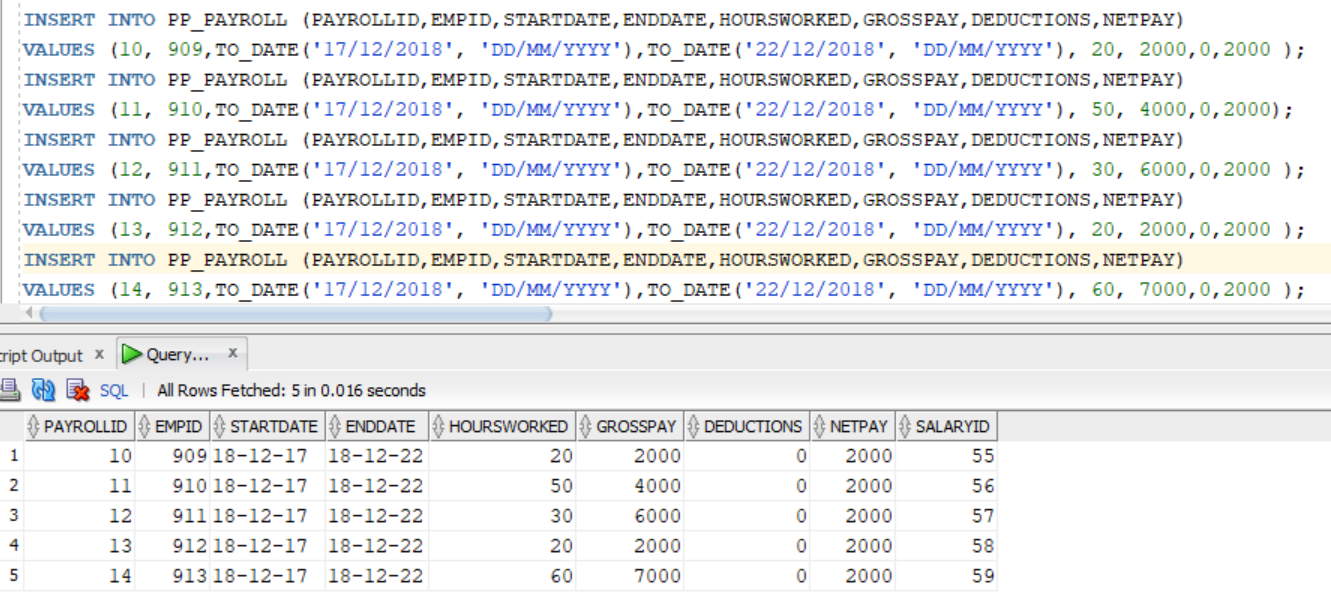
VALUES (912,5005, 2109, 2504,1104,'IAO','BQOA',121211221, TO\_DATE('17/12/1993', 'DD/MM/YYYY'), '9192 GAL AVE', 'TORONTO','ON','M1L7E8' );

INSERT INTO PP\_EMPLOYEE (EMPID, DEPID, JOBID, TYPEID, CATID, FIRSTNAME, LASTNAME, SSN, DOB, ADDRESS, CITY, STATE, POSTALCODE)

VALUES (913,6006, 2110, 2505,1105,'SIM','QOO',121211221, TO\_DATE('17/12/1999', 'DD/MM/YYYY'), '9190 MALVANO AVE', 'TORONTO','ON','M1L7Y8' );

Table name – Payroll (Tariq)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/**  **Constraint** | **Data Type** | **Field length** |
| Passenger\_id | A unique identification number for passenger | Primary key | Number | 10 |
| User\_id | Used to login into the online system | Foreign key | Number | 10 |
| Age | Age of the passenger |  | Number | 3 |
| Reservation\_status | Information of reservation |  | Varchar2 | 2 |
| Seat\_number | Seat information |  | Number | 3 |
| Ticket\_id | Information regarding the ticket |  | number | 10 |



CREATE TABLE PP\_PAYROLL

(

PAYROLLID NUMBER NOT NULL

, EMPID NUMBER NOT NULL

, STARTDATE DATE NOT NULL

, ENDDATE DATE NOT NULL

, HOURSWORKED NUMBER NOT NULL

, GROSSPAY NUMBER NOT NULL

, DEDUCTIONS NUMBER

, NETPAY NUMBER NOT NULL

, CONSTRAINT PP\_PAYROLL\_PK PRIMARY KEY

(

PAYROLLID

)

ENABLE

);

INSERT INTO PP\_PAYROLL (PAYROLLID,EMPID,STARTDATE,ENDDATE,HOURSWORKED,GROSSPAY,DEDUCTIONS,NETPAY)

VALUES (10, 909,TO\_DATE('17/12/2018', 'DD/MM/YYYY'),TO\_DATE('22/12/2018', 'DD/MM/YYYY'), 20, 2000,0,2000 );

INSERT INTO PP\_PAYROLL (PAYROLLID,EMPID,STARTDATE,ENDDATE,HOURSWORKED,GROSSPAY,DEDUCTIONS,NETPAY)

VALUES (11, 910,TO\_DATE('17/12/2018', 'DD/MM/YYYY'),TO\_DATE('22/12/2018', 'DD/MM/YYYY'), 50, 4000,0,2000);

INSERT INTO PP\_PAYROLL (PAYROLLID,EMPID,STARTDATE,ENDDATE,HOURSWORKED,GROSSPAY,DEDUCTIONS,NETPAY)

VALUES (12, 911,TO\_DATE('17/12/2018', 'DD/MM/YYYY'),TO\_DATE('22/12/2018', 'DD/MM/YYYY'), 30, 6000,0,2000 );

INSERT INTO PP\_PAYROLL (PAYROLLID,EMPID,STARTDATE,ENDDATE,HOURSWORKED,GROSSPAY,DEDUCTIONS,NETPAY)

VALUES (13, 912,TO\_DATE('17/12/2018', 'DD/MM/YYYY'),TO\_DATE('22/12/2018', 'DD/MM/YYYY'), 20, 2000,0,2000 );

INSERT INTO PP\_PAYROLL (PAYROLLID,EMPID,STARTDATE,ENDDATE,HOURSWORKED,GROSSPAY,DEDUCTIONS,NETPAY)

VALUES (14, 913,TO\_DATE('17/12/2018', 'DD/MM/YYYY'),TO\_DATE('22/12/2018', 'DD/MM/YYYY'), 60, 7000,0,2000 );

ALTER TABLE PP\_PAYROLL ADD SALARYID NUMBER(15);

DESC PP\_PAYROLL;

UPDate pp\_payroll set salaryid ='55' where payrollid= '10';

UPDate pp\_payroll set salaryid ='56' where payrollid= '11';

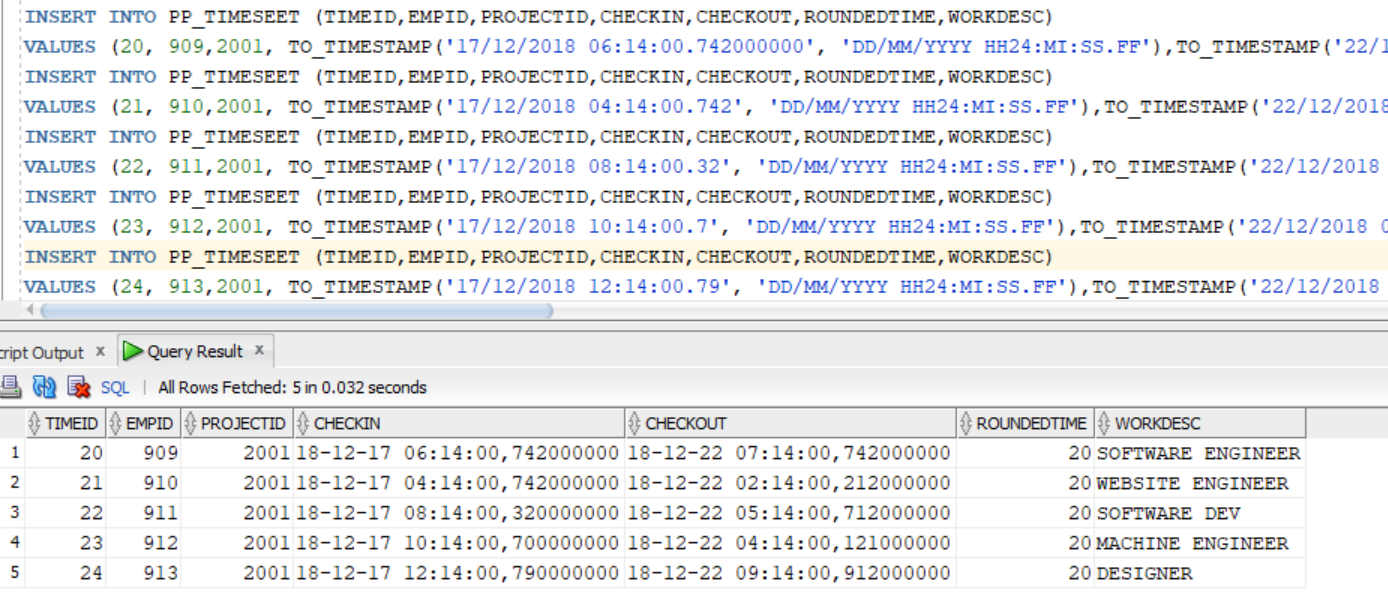
UPDate pp\_payroll set salaryid ='57' where payrollid= '12';

UPDate pp\_payroll set salaryid ='58' where payrollid= '13';

UPDate pp\_payroll set salaryid ='59' where payrollid= '14';

Table name - TimeSheet (Tariq)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/**  **Constraint** | **Data Type** | **Field length** |
| Train\_number | Number of the train | Primary key | number | 10 |
| Train\_name | Name of the train |  | Varchar2 | 20 |
| Arrival\_time | Time at which train arrives |  | Time | 10 |
| Departure\_time | Time at which train departs |  | Time | 10 |
| Availability\_of\_seats | Available seats |  | number | 3 |
| Fare | Cost of the ticket |  | Number | 10 |
| Date | day of the bookings |  | Date | 10 |

****

CREATE TABLE PP\_TIMESEET

(

TIMEID NUMBER NOT NULL

, EMPID NUMBER

, PROJECTID NUMBER

, CHECKIN TIMESTAMP

, CHECKOUT TIMESTAMP

, ROUNDEDTIME NUMBER

, WORKDESC VARCHAR2(20)

, CONSTRAINT PP\_TIMESEET\_PK PRIMARY KEY

(

TIMEID

)

ENABLE

);

INSERT INTO PP\_TIMESEET (TIMEID,EMPID,PROJECTID,CHECKIN,CHECKOUT,ROUNDEDTIME,WORKDESC)

VALUES (20, 909,2001, TO\_TIMESTAMP('17/12/2018 06:14:00.742000000', 'DD/MM/YYYY HH24:MI:SS.FF'),TO\_TIMESTAMP('22/12/2018 07:14:00.742000000', 'DD/MM/YYYY HH24:MI:SS.FF'), 20, 'SOFTWARE ENGINEER');

INSERT INTO PP\_TIMESEET (TIMEID,EMPID,PROJECTID,CHECKIN,CHECKOUT,ROUNDEDTIME,WORKDESC)

VALUES (21, 910,2001, TO\_TIMESTAMP('17/12/2018 04:14:00.742', 'DD/MM/YYYY HH24:MI:SS.FF'),TO\_TIMESTAMP('22/12/2018 02:14:00.212000000', 'DD/MM/YYYY HH24:MI:SS.FF'), 20, 'WEBSITE ENGINEER');

INSERT INTO PP\_TIMESEET (TIMEID,EMPID,PROJECTID,CHECKIN,CHECKOUT,ROUNDEDTIME,WORKDESC)

VALUES (22, 911,2001, TO\_TIMESTAMP('17/12/2018 08:14:00.32', 'DD/MM/YYYY HH24:MI:SS.FF'),TO\_TIMESTAMP('22/12/2018 05:14:00.712', 'DD/MM/YYYY HH24:MI:SS.FF'), 20, 'SOFTWARE DEV');

INSERT INTO PP\_TIMESEET (TIMEID,EMPID,PROJECTID,CHECKIN,CHECKOUT,ROUNDEDTIME,WORKDESC)

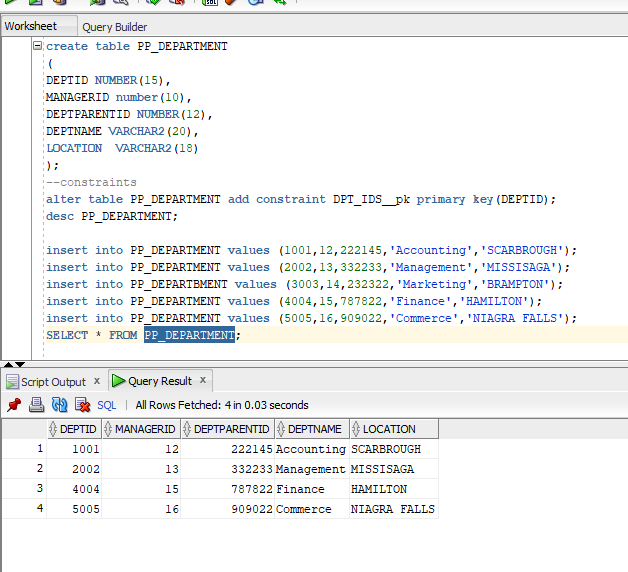
VALUES (23, 912,2001, TO\_TIMESTAMP('17/12/2018 10:14:00.7', 'DD/MM/YYYY HH24:MI:SS.FF'),TO\_TIMESTAMP('22/12/2018 04:14:00.121', 'DD/MM/YYYY HH24:MI:SS.FF'), 20, 'MACHINE ENGINEER');

INSERT INTO PP\_TIMESEET (TIMEID,EMPID,PROJECTID,CHECKIN,CHECKOUT,ROUNDEDTIME,WORKDESC)

VALUES (24, 913,2001, TO\_TIMESTAMP('17/12/2018 12:14:00.79', 'DD/MM/YYYY HH24:MI:SS.FF'),TO\_TIMESTAMP('22/12/2018 09:14:00.912', 'DD/MM/YYYY HH24:MI:SS.FF'), 20, 'DESIGNER');

Table name – department(Simranjeet)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Key Type/  Constraint | Data Type | Field length |
| DEPTID | Used to login into the online system | Primary Key | number | 15 |
| MANAGERID | First name of the user |  | number | 10 |
| DEPTPARENTID | Last name of the user |  | number | 12 |
| DEPTNAME | Male or female user |  | varchar2 | 20 |
| LOCATION | Age of the user |  | varchar | 18 |



create table PP\_DEPARTMENT

(

DEPTID NUMBER(15),

MANAGERID number(10),

DEPTPARENTID NUMBER(12),

DEPTNAME VARCHAR2(20),

LOCATION VARCHAR2(18)

);

--constraints

alter table PP\_DEPARTMENT add constraint DPT\_IDS\_\_pk primary key(DEPTID);

desc PP\_DEPARTMENT;

insert into PP\_DEPARTMENT values (1001,12,222145,'Accounting','SCARBROUGH');

insert into PP\_DEPARTMENT values (2002,13,332233,'Management','MISSISAGA');

insert into PP\_DEPARTBMENT values (3003,14,232322,'Marketing','BRAMPTON');

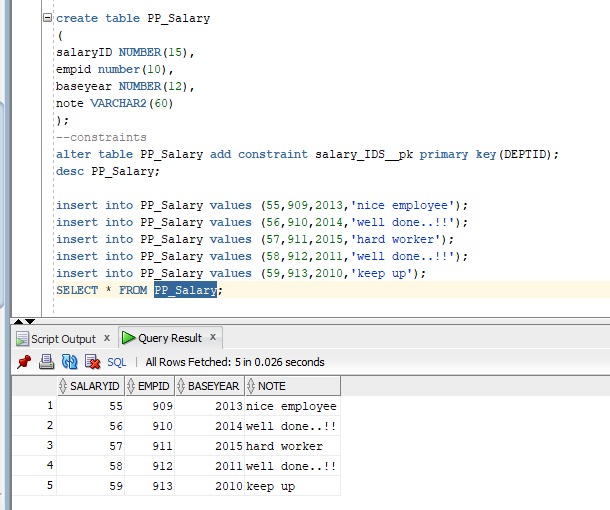
insert into PP\_DEPARTMENT values (4004,15,787822,'Finance','HAMILTON');

insert into PP\_DEPARTMENT values (5005,16,909022,'Commerce','NIAGRA FALLS');

SELECT \* FROM PP\_DEPARTMENT;

Table name – Salary (Simranjeet)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/**  **Constraint** | **Data Type** | **Field length** |
| salaryID | Unique identification for the ticket | Primary key | Number | 15 |
| empid | Used to login into the online system |  | Number | 10 |
| baseyear | Information update regarding the ticket |  | number | 12 |
| note | Number of the train |  | varchar2 | 60 |

****

create table PP\_Salary

(

salaryID NUMBER(15),

empid number(10),

baseyear NUMBER(12),

note VARCHAR2(60)

);

--constraints

alter table PP\_Salary add constraint salary\_IDS\_\_pk primary key(DEPTID);

desc PP\_Salary;

insert into PP\_Salary values (55,909,2013,'nice employee');

insert into PP\_Salary values (56,910,2014,'well done..!!');

insert into PP\_Salary values (57,911,2015,'hard worker');

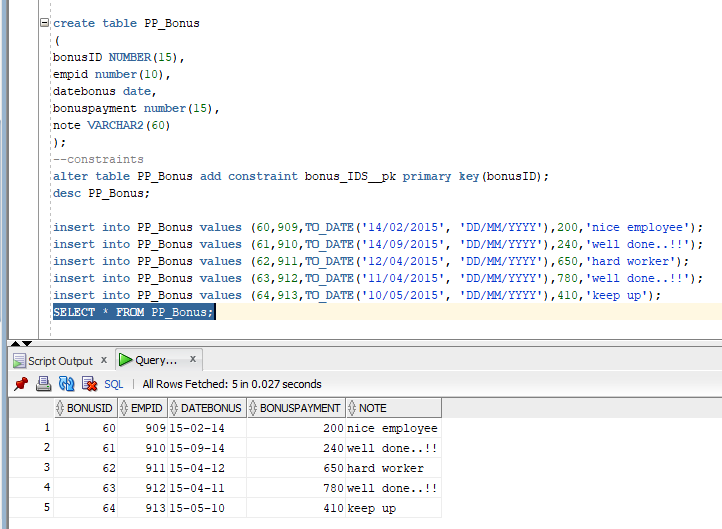
insert into PP\_Salary values (58,912,2011,'well done..!!');

insert into PP\_Salary values (59,913,2010,'keep up');

SELECT \* FROM PP\_Salary;

Table name – Bonus(Simranjeet)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Key Type/  Constraint | Data Type | Field length |
| user\_id | Used to login into the online system | Primary Key | number | 10 |
| empid | First name of the user |  | Varchar2 | 15 |
| datebonus | Last name of the user |  | Varchar2 | 15 |
| bonuspayment | Male or female user |  | char | 1 |
| note | Age of the user |  | number | 3 |



create table PP\_Bonus

(

bonusID NUMBER(15),

empid number(10),

datebonus date,

bonuspayment number(15),

note VARCHAR2(60)

);

--constraints

alter table PP\_Bonus add constraint bonus\_IDS\_\_pk primary key(bonusID);

desc PP\_Bonus;

insert into PP\_Bonus values (60,909,TO\_DATE('14/02/2015', 'DD/MM/YYYY'),200,'nice employee');

insert into PP\_Bonus values (61,910,TO\_DATE('14/09/2015', 'DD/MM/YYYY'),240,'well done..!!');

insert into PP\_Bonus values (62,911,TO\_DATE('12/04/2015', 'DD/MM/YYYY'),650,'hard worker');

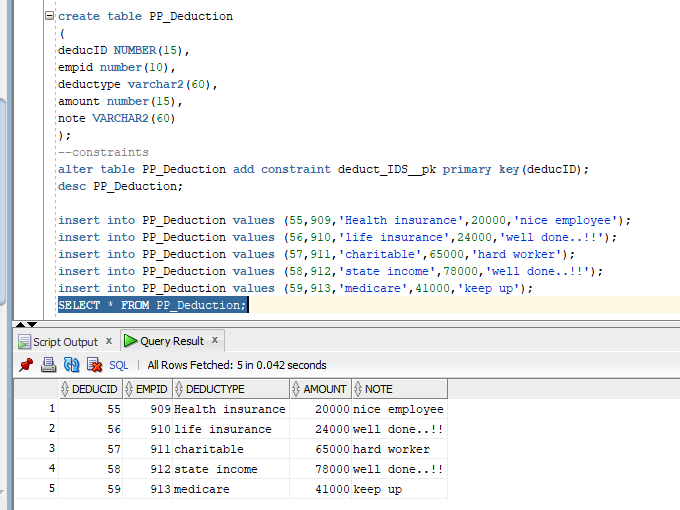
insert into PP\_Bonus values (63,912,TO\_DATE('11/04/2015', 'DD/MM/YYYY'),780,'well done..!!');

insert into PP\_Bonus values (64,913,TO\_DATE('10/05/2015', 'DD/MM/YYYY'),410,'keep up');

SELECT \* FROM PP\_Bonus;

Table name – Deductions(Simranjeet)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Key Type/  Constraint | Data Type | Field length |
| deducID | Used to login into the online system | Primary Key | number | 15 |
| empID | First name of the user |  | number | 10 |
| deductype | Last name of the user |  | number | 60 |
| amount | Male or female user |  | varchar | 15 |
| note | Age of the user |  | varchar2 | 60 |

****

SELECT \* FROM PP\_Bonus;

create table PP\_Deduction

(

deducID NUMBER(15),

empID number(10),

deductype NUMBER(60),

amount VARCHAR2(15),

note VARCHAR2(60)

);

--constraints

alter table PP\_Deduction add constraint Deduc\_IDS\_\_pk primary key(DEPTID);

desc PP\_Deduction;

insert into PP\_Deduction values (55,909,'Health Insurance',20000,'nice employee');

insert into PP\_Deduction values (56,910,'life insurance',24000,'well done..!!');

insert into PP\_Deduction values (57,911,'charitable',65000,'hard worker');

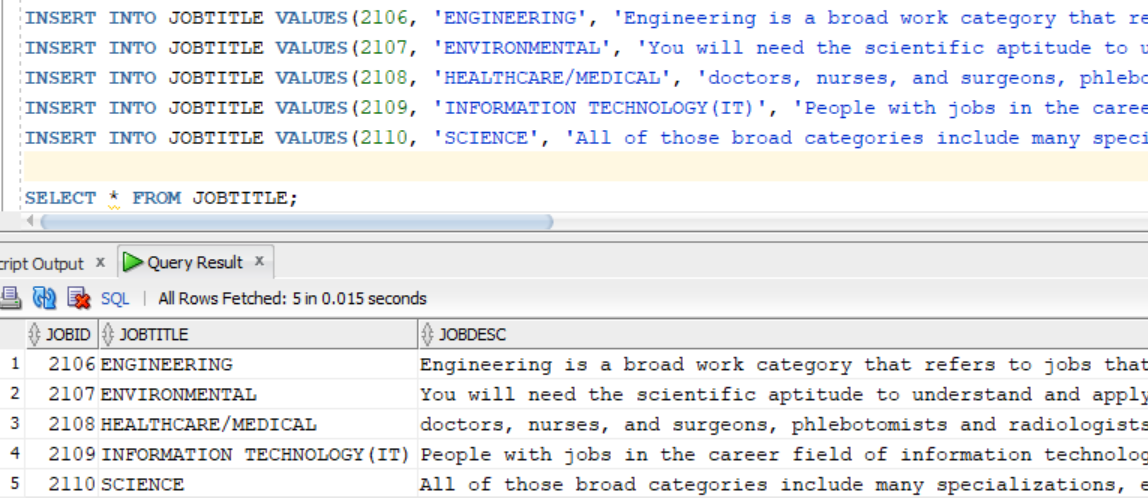
insert into PP\_Deduction values (58,912,'state income',78000,'well done..!!');

insert into PP\_Deduction values (59,913,'medicare',41000,'keep up');

SELECT \* FROM PP\_Deduction;

Table name – JobTitle(Kelly)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/**  **Constraint** | **Data Type** | **Field length** |
| JobId | Used to identify the job id | Primary key | Number | 5 |
| JobTitle | Used to identify the job title |  | Varchar2 | 50 |
| JobDesc | Used to understand the job |  | Varchar2 | 2000 |



-- Create Job Title Table

CREATE TABLE JOBTITLE (

JOBID NUMBER(5),

JOBTITLE VARCHAR2(50),

JOBDESC VARCHAR2(2000),

CONSTRAINT JOBTITLE\_ID\_PK PRIMARY KEY(JOBID)

);

--inserting data

INSERT INTO JOBTITLE VALUES(2106, 'ENGINEERING', 'Engineering is a broad work category that refers to jobs that use science and mathematics to solve a variety of problems.');

INSERT INTO JOBTITLE VALUES(2107, 'ENVIRONMENTAL', 'You will need the scientific aptitude to understand and apply complex concepts in biology, chemistry, geology, and physics and will learn to apply both quantitative and qualitative analytical skills to solving problems and interpreting research data.');

INSERT INTO JOBTITLE VALUES(2108, 'HEALTHCARE/MEDICAL', 'doctors, nurses, and surgeons, phlebotomists and radiologists.');

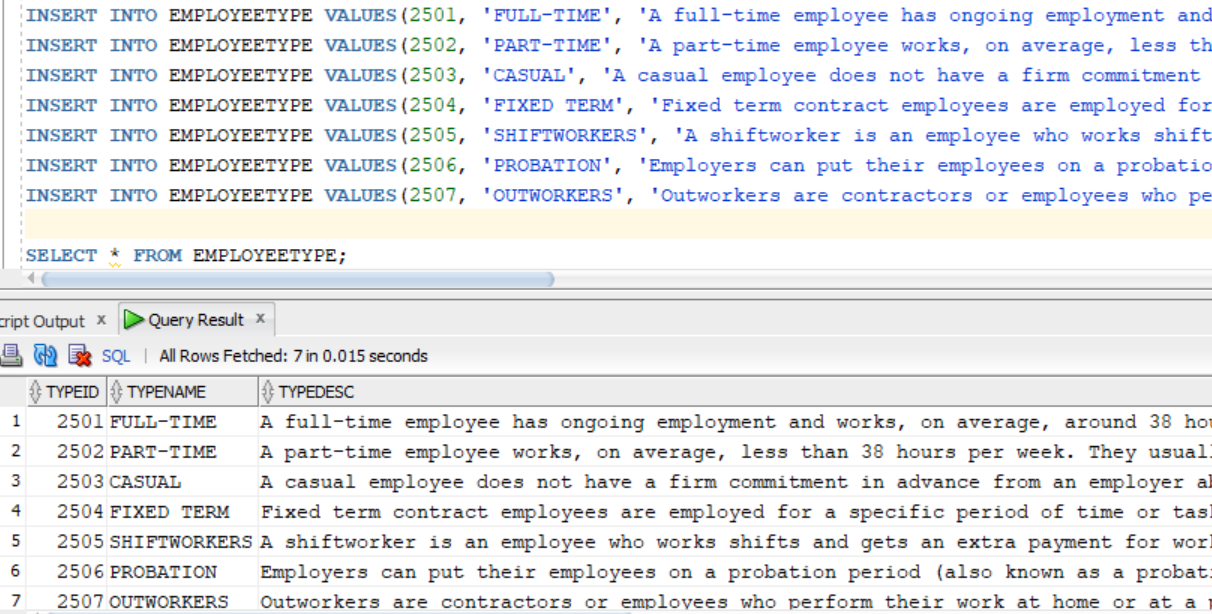
INSERT INTO JOBTITLE VALUES(2109, 'INFORMATION TECHNOLOGY(IT)', 'People with jobs in the career field of information technology (IT) use computers, software, networks, servers, and other technology to manage and store data.');

INSERT INTO JOBTITLE VALUES(2110, 'SCIENCE', 'All of those broad categories include many specializations, each with its own set of technical skills, knowledge, and educational requirements.');

SELECT \* FROM JOBTITLE;

Table name – EmployeeType(Kelly)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/**  **Constraint** | **Data Type** | **Field length** |
| TypeId | Used to identify the type id | Primary key | Number | 5 |
| TypeName | used to identify the type name |  | Varchar2 | 50 |
| TypeDesc | used to understand the type |  | Varchar2 | 2000 |



-- Create Employee Type Table

CREATE TABLE EMPLOYEETYPE (

TYPEID NUMBER(5),

TYPENAME VARCHAR2(50),

TYPEDESC VARCHAR2(2000),

CONSTRAINT EMPTYPE\_ID\_PK PRIMARY KEY(TYPEID)

);

DROP TABLE EMPLOYEETYPE;

INSERT INTO EMPLOYEETYPE VALUES(2501, 'FULL-TIME', 'A full-time employee has ongoing employment and works, on average, around 38 hours each week. The actual hours of work for an employee in a particular job or industry are agreed between the employer and the employee and/or set by an award or registered agreement.');

INSERT INTO EMPLOYEETYPE VALUES(2502, 'PART-TIME', 'A part-time employee works, on average, less than 38 hours per week. They usually work regular hours each week and is entitled to the same benefits as a full-time employee, but on a pro rata basis. They are a permanent employee or on a fixed-term contract.');

INSERT INTO EMPLOYEETYPE VALUES(2503, 'CASUAL', 'A casual employee does not have a firm commitment in advance from an employer about how long they will be employed for, or the days (or hours) they will work. A casual employee also does not commit to all work an employer might offer.');

INSERT INTO EMPLOYEETYPE VALUES(2504, 'FIXED TERM', 'Fixed term contract employees are employed for a specific period of time or task. For example a 6 month contract where employment ends after 6 months.');

INSERT INTO EMPLOYEETYPE VALUES(2505, 'SHIFTWORKERS', 'A shiftworker is an employee who works shifts and gets an extra payment for working shift hours.');

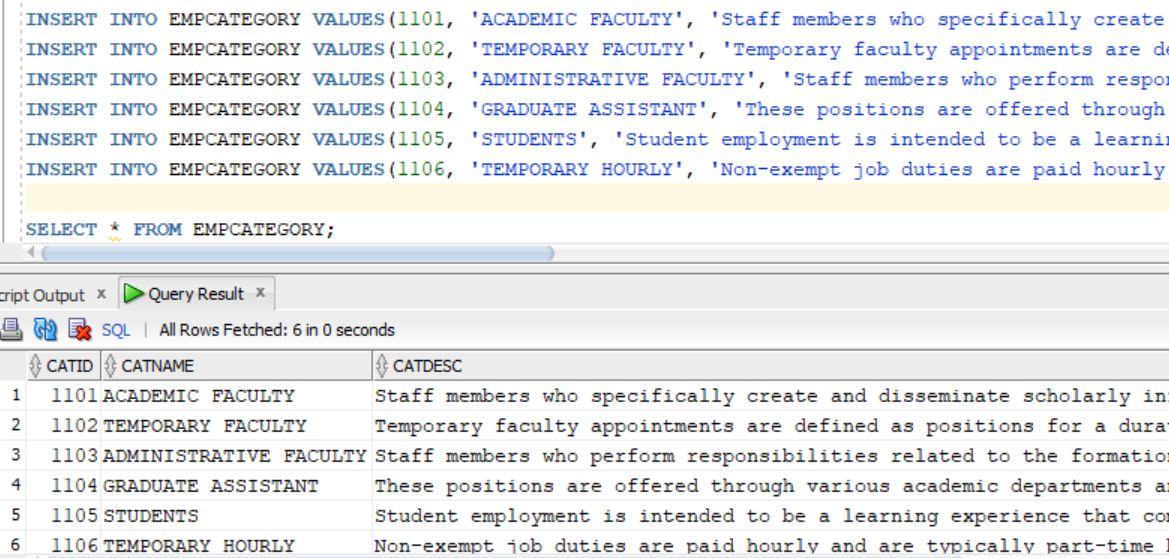
INSERT INTO EMPLOYEETYPE VALUES(2506, 'PROBATION', 'Employers can put their employees on a probation period (also known as a probationary period) to assess if employees are suitable for the role and business.');

INSERT INTO EMPLOYEETYPE VALUES(2507, 'OUTWORKERS', 'Outworkers are contractors or employees who perform their work at home or at a place that wouldn’t normally be thought of as a business premises. Outworkers are common in the textile, clothing or footwear industry.');

SELECT \* FROM EMPLOYEETYPE;

Table name – EmpCategory (Kelly)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Key Type/**  **Constraint** | **Data Type** | **Field length** |
| CatId | used to identify the category id | Primary key | Number | 5 |
| CatName | used to identify the category name |  | Varchar2 | 50 |
| CatDesc | used to understand the category |  | Varchar2 | 2000 |



-- Create Employee Category Table

CREATE TABLE EMPCATEGORY (

CATID NUMBER(5),

CATNAME VARCHAR2(50),

CATDESC VARCHAR2(2000),

CONSTRAINT EMPCAT\_ID\_PK PRIMARY KEY(CATID)

);

INSERT INTO EMPCATEGORY VALUES(1101, 'ACADEMIC FACULTY', 'Staff members who specifically create and disseminate scholarly information through teaching, research and library services.');

INSERT INTO EMPCATEGORY VALUES(1102, 'TEMPORARY FACULTY', 'Temporary faculty appointments are defined as positions for a duration of one year or less.');

INSERT INTO EMPCATEGORY VALUES(1103, 'ADMINISTRATIVE FACULTY', 'Staff members who perform responsibilities related to the formation and implementation of managerial policies or general operations of the university.');

INSERT INTO EMPCATEGORY VALUES(1104, 'GRADUATE ASSISTANT', 'These positions are offered through various academic departments and are paid by grants or state funds.');

INSERT INTO EMPCATEGORY VALUES(1105, 'STUDENTS', 'Student employment is intended to be a learning experience that complements a student’s academic and career goals and provides valuable work experience.');

INSERT INTO EMPCATEGORY VALUES(1106, 'TEMPORARY HOURLY', 'Non-exempt job duties are paid hourly and are typically part-time but may be full-time for a few months.');

SELECT \* FROM EMPCATEGORY;

**SQL code for generating table and QUERIES:**

**Queries (by --**KELLY---**)**

**-- FIND NUMBER OF DATA OF EACH PROJECT ID**

SELECT PROJECTID, COUNT(\*) FROM PP\_TIMESEET GROUP BY PROJECTID;



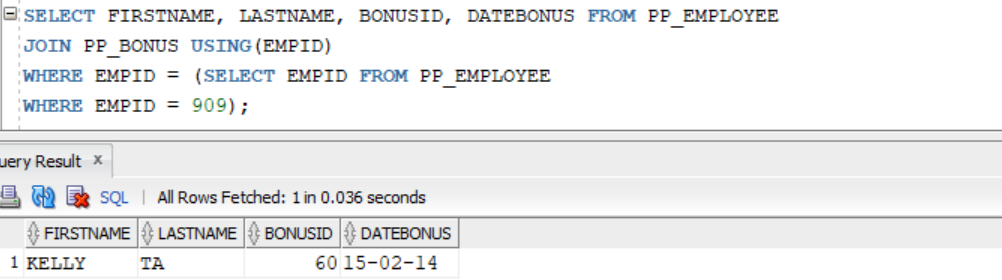
-- **USING JOIN TO DISPLAY THE BONUS ID FROM EMP ID WITH FIRST NAME AS KELLY**

SELECT FIRSTNAME, LASTNAME, BONUSID, DATEBONUS FROM PP\_EMPLOYEE

JOIN PP\_BONUS USING(EMPID)

WHERE EMPID = (SELECT EMPID FROM PP\_EMPLOYEE

WHERE EMPID = 909);



**-- TO FIND EMPID WHERE DEDUCID = 570 IN A PL/SQL BLOCK**

SET SERVEROUTPUT ON;

DECLARE

LV\_EMPID PP\_DEDUCTION.EMPID%TYPE;

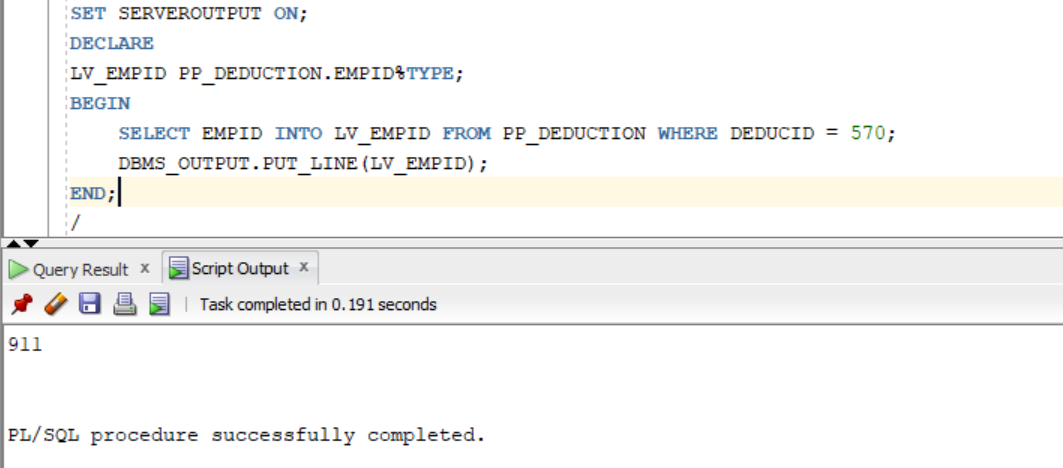
BEGIN

SELECT EMPID INTO LV\_EMPID FROM PP\_DEDUCTION WHERE DEDUCID = 570;

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

END;

/



**-- DISPLAY JOBID AND JOB DESCRIPTION WHERE JOB ID = 2109 USING PL/SQL BLOCK %ROWTYPE**

SET SERVEROUTPUT ON;

DECLARE

L\_JOBTITLE JOBTITLE%ROWTYPE;

BEGIN

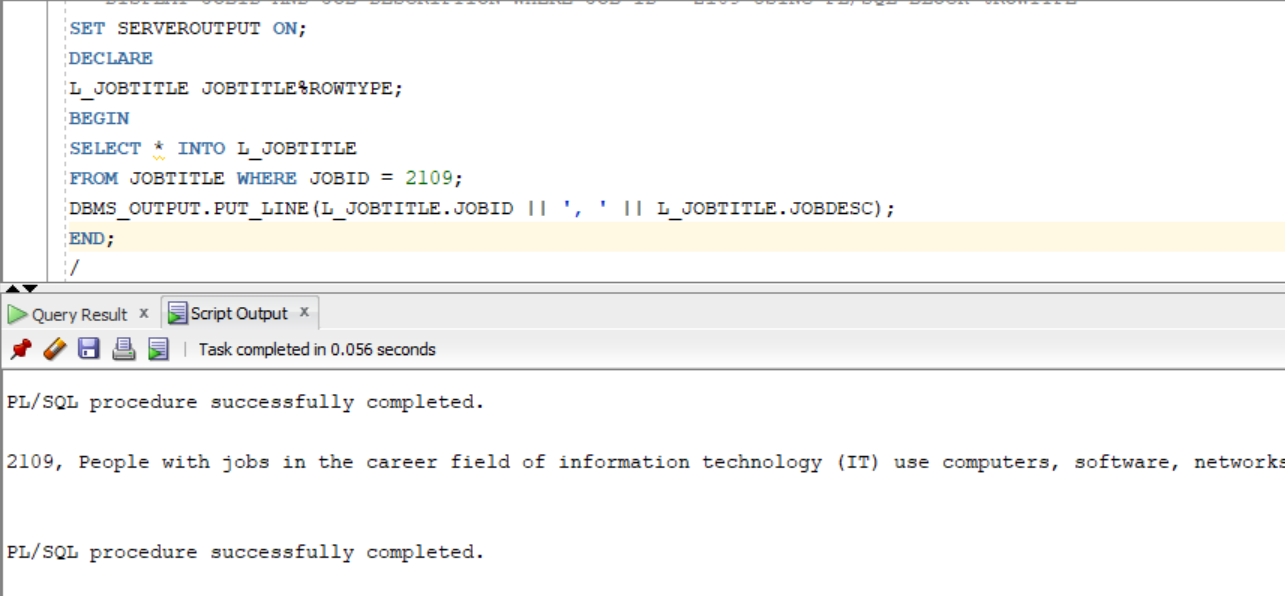
SELECT \* INTO L\_JOBTITLE

FROM JOBTITLE WHERE JOBID = 2109;

DBMS\_OUTPUT.PUT\_LINE(L\_JOBTITLE.JOBID || ', ' || L\_JOBTITLE.JOBDESC);

END;

/



**-- DISPLAY EMPLOYEE ID, PAYROLL ID AND START DATE USING CURSOR IN PL/SQL BLOCK**

SET SERVEROUTPUT ON;

DECLARE

C\_EMPID NUMBER(5);

C\_PAYROLLID NUMBER(5);

C\_STARTDATE DATE;

CURSOR C\_PAYROLL IS

SELECT EMPID, PAYROLLID, STARTDATE FROM PP\_PAYROLL;

BEGIN

OPEN C\_PAYROLL;

LOOP

FETCH C\_PAYROLL INTO C\_EMPID, C\_PAYROLLID, C\_STARTDATE;

EXIT WHEN C\_PAYROLL%NOTFOUND;

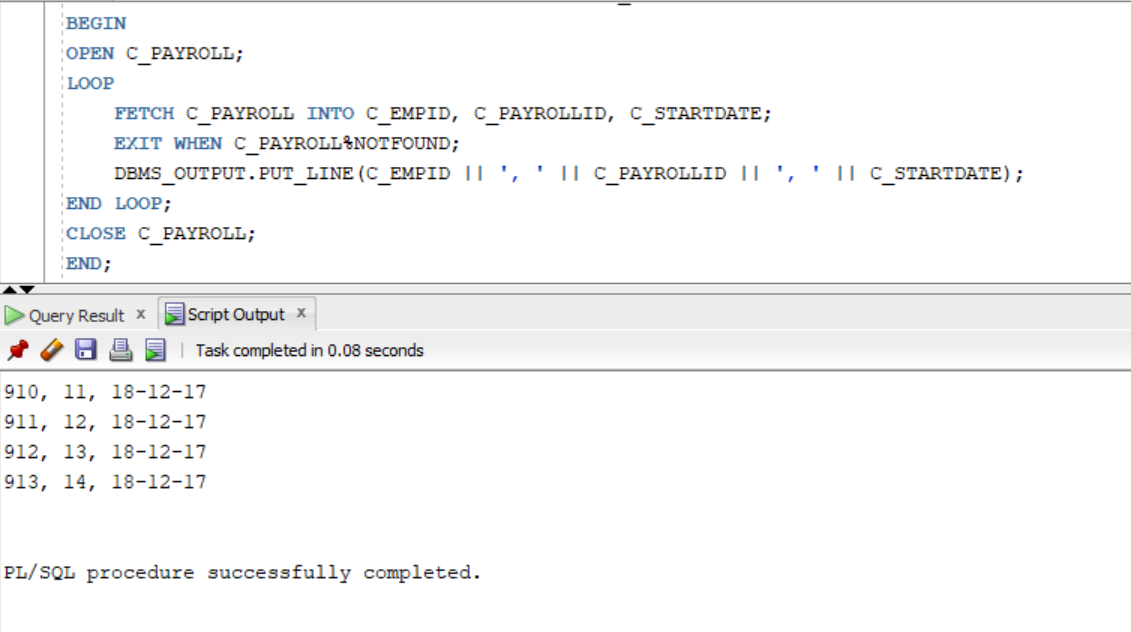
DBMS\_OUTPUT.PUT\_LINE(C\_EMPID || ', ' || C\_PAYROLLID || ', ' || C\_STARTDATE);

END LOOP;

CLOSE C\_PAYROLL;

END;

/



-**- PACKAGE TO FIND EMPLOYEE TYPE**

CREATE OR REPLACE PACKAGE EMPLOYEE\_TYPE AS

PROCEDURE FIND\_EMPLOYEE

(P\_TYPEID IN NUMBER,

P\_TYPENAME OUT VARCHAR2,

P\_TYPEDESC OUT VARCHAR2);

END EMPLOYEE\_TYPE;

/

--**PACKAGE BODY**

CREATE OR REPLACE PACKAGE BODY EMPLOYEE\_TYPE AS

PROCEDURE FIND\_EMPLOYEE

(P\_TYPEID IN NUMBER,

P\_TYPENAME OUT VARCHAR2,

P\_TYPEDESC OUT VARCHAR2)

AS

BEGIN

SELECT TYPENAME, TYPEDESC INTO P\_TYPENAME, P\_TYPEDESC FROM EMPLOYEETYPE

WHERE TYPEID = P\_TYPEID;

DBMS\_OUTPUT.PUT\_LINE('TYPE NAME: ' || P\_TYPENAME);

DBMS\_OUTPUT.PUT\_LINE('TYPE DESC: ' || P\_TYPEDESC);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('INVALID TYPE ID');

END FIND\_EMPLOYEE;

END EMPLOYEE\_TYPE;

/

-- **TESTING PACKAGE**

SET SERVEROUTPUT ON;

DECLARE

LV\_NAME VARCHAR2(50);

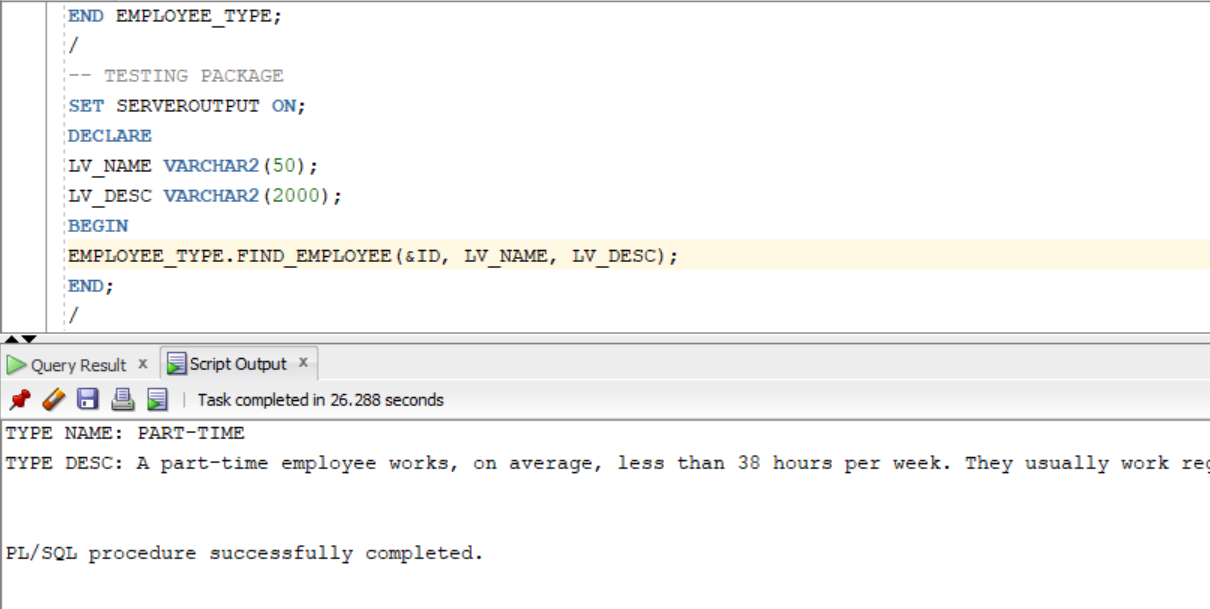
LV\_DESC VARCHAR2(2000);

BEGIN

EMPLOYEE\_TYPE.FIND\_EMPLOYEE(&ID, LV\_NAME, LV\_DESC);

END;

/



**Queries (by --**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

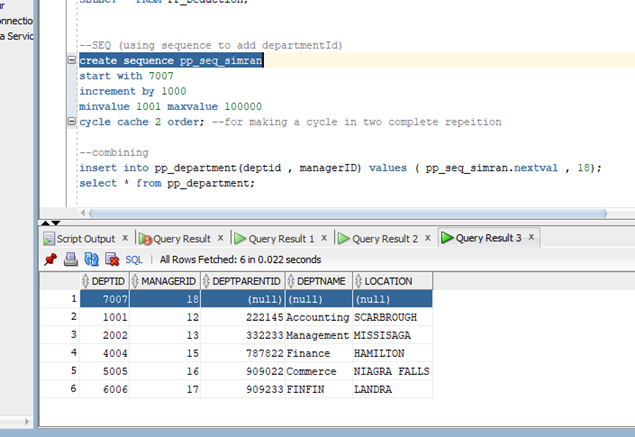
INCREMENT BY 1000

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

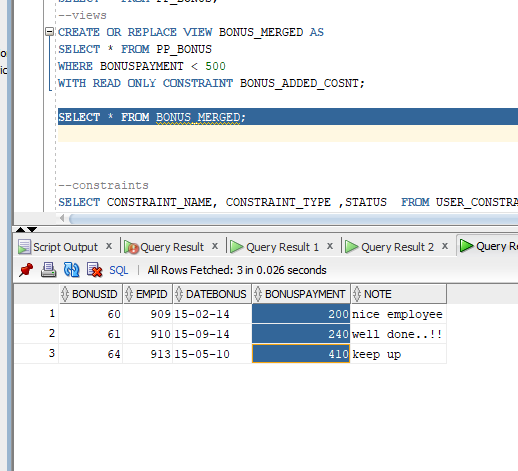
CREATE OR REPLACE VIEW BONUS\_MERGED AS

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');

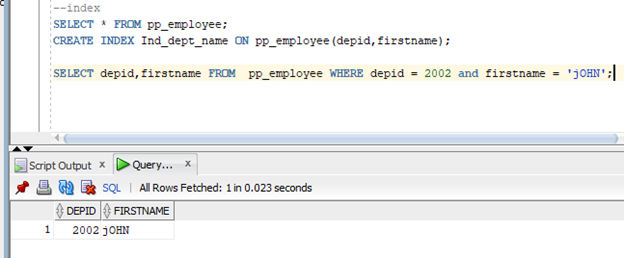
4. Indexes

(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';

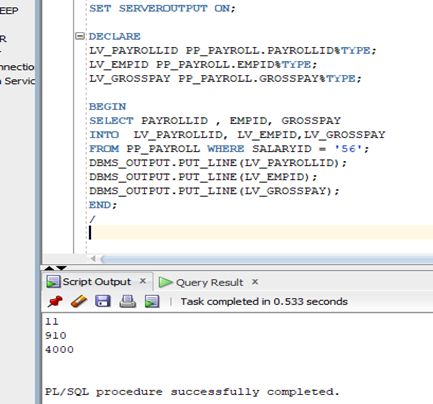
DBMS\_OUTPUT.PUT\_LINE(LV\_PAYROLLID);

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

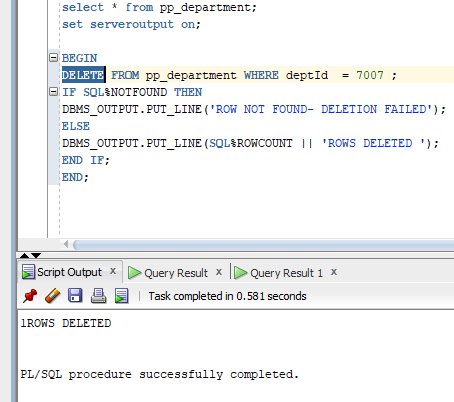
DBMS\_OUTPUT.PUT\_LINE('ROW NOT FOUND- DELETION FAILED');

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;

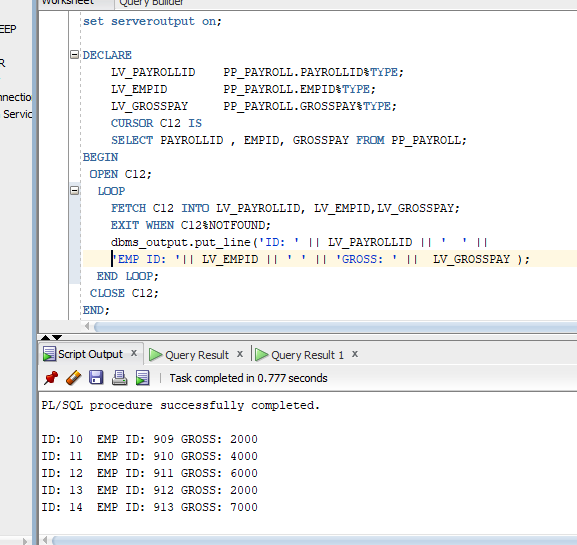
DBMS\_OUTPUT.PUT\_LINE('ID: ' || LV\_PAYROLLID || ' ' ||

'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);

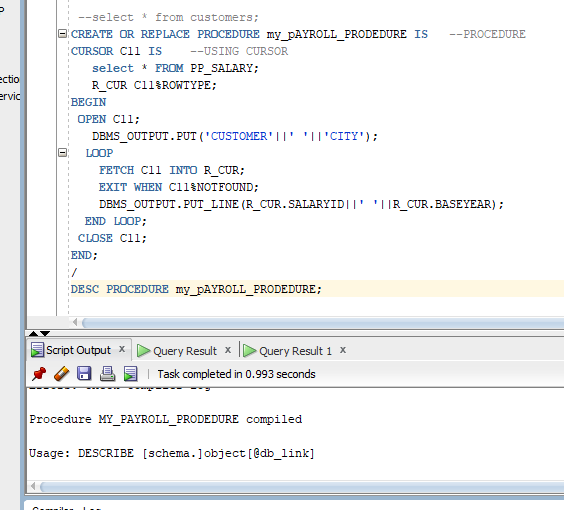
END LOOP;

CLOSE C11;

END;

/

DESC PROCEDURE my\_pAYROLL\_PRODEDURE;



9.) Package

I.) PACKAGE HAVING CURSOR

--Declaring the cursor in package

CREATE OR REPLACE PACKAGE employee\_pay\_pkg IS

CURSOR emp\_cur IS --USING CURSOR

select \* FROM PP\_SALARY;

R1\_CUR emp\_cur%ROWTYPE;

end;

/

SET SERVEROUTPUT ON;

create or replace package body employee\_pay\_pkg is

procedure EMP\_PRO is

R1\_CUR emp\_cur%ROWTYPE;

begin

open emp\_cur;

loop

fetch emp\_cur into R1\_CUR;

exit when emp\_cur%NOTFOUND;

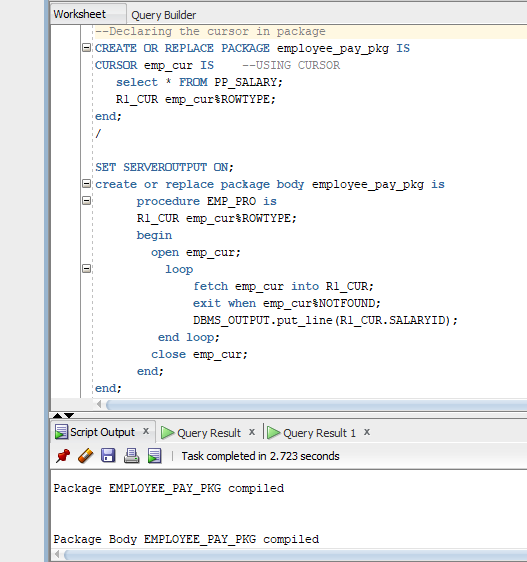
DBMS\_OUTPUT.put\_line(R1\_CUR.SALARYID);

end loop;

close emp\_cur;

end;

end;



10.) TRIGGERS (SIMPLE)

CREATE OR REPLACE TRIGGER pay\_trig

AFTER INSERT OR DELETE OR UPDATE ON PP\_payroll

FOR EACH ROW

BEGIN

if inserting then

update pp\_salary set SALARYID =: New.SALARYID where ;EMPID = :new.EMPID

elsif

UPDATING THEN

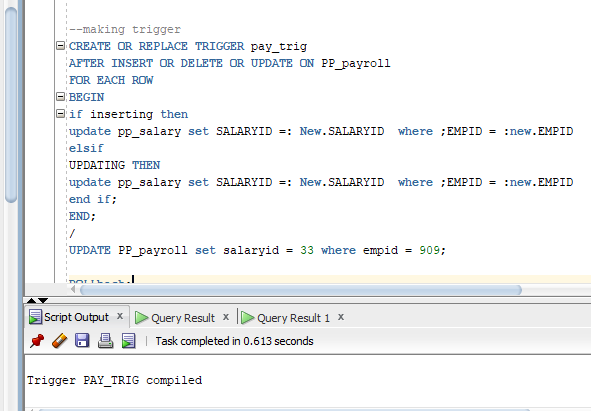
update pp\_salary set SALARYID =: New.SALARYID where ;EMPID = :new.EMPID

end if;

END;

/

UPDATE PP\_payroll set salaryid = 33 where empid = 909;



**Queries (by --**TARIQ---**)**

**--PACKAGE SPECIFICATION-------**

create OR REPLACE PACKAGE EMP\_QUERY\_PKG IS

PROCEDURE EMP\_INFO\_PP

(P\_ID IN PP\_EMPLOYEE.EMPID%TYPE,

P\_FIRSTNAME OUT PP\_EMPLOYEE.FIRSTNAME%TYPE,

P\_CITY OUT PP\_EMPLOYEE.CITY%TYPE,

P\_STATE OUT PP\_EMPLOYEE.STATE%TYPE);

PROCEDURE EMP\_INFO\_PP

(P\_ID IN PP\_EMPLOYEE.LASTNAME%TYPE,

P\_FIRSTNAME OUT PP\_EMPLOYEE.FIRSTNAME%TYPE,

P\_CITY OUT PP\_EMPLOYEE.CITY%TYPE,

P\_STATE OUT PP\_EMPLOYEE.STATE%TYPE);

END;

/

SELECT \* FROM PP\_EMPLOYEE;

**--CREATING THE PACKAGE BODY--------------------**

CREATE OR REPLACE PACKAGE BODY EMP\_QUERY\_PKG IS

PROCEDURE EMP\_INFO\_PP

(P\_ID IN PP\_EMPLOYEE.EMPID%TYPE,

P\_FIRSTNAME OUT PP\_EMPLOYEE.FIRSTNAME%TYPE,

P\_CITY OUT PP\_EMPLOYEE.CITY%TYPE,

P\_STATE OUT PP\_EMPLOYEE.STATE%TYPE)

IS

BEGIN

SELECT FIRSTNAME, CITY, STATE INTO P\_FIRSTNAME, P\_CITY, P\_STATE

FROM PP\_EMPLOYEE

WHERE EMPID = P\_ID ;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('ID DOES NOT EXIST');

END;

PROCEDURE EMP\_INFO\_PP

(P\_ID IN PP\_EMPLOYEE.LASTNAME%TYPE,

P\_FIRSTNAME OUT PP\_EMPLOYEE.FIRSTNAME%TYPE,

P\_CITY OUT PP\_EMPLOYEE.CITY%TYPE,

P\_STATE OUT PP\_EMPLOYEE.STATE%TYPE)

IS

BEGIN

SELECT FIRSTNAME, CITY, STATE INTO P\_FIRSTNAME, P\_CITY, P\_STATE

FROM PP\_EMPLOYEE

WHERE LASTNAME= P\_ID;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('LASTNAME DOES NOT FOUND');

END;

END;

/

**--TESTING THE PACKAGE------------------------**

**--FIRST ATTEMPT-----------------------------------------**

DECLARE

LV\_ID\_NUM PP\_EMPLOYEE.EMPID%TYPE :=&EMPLOYEE\_ID;

LV\_FIRSTNAME\_TXT PP\_EMPLOYEE.FIRSTNAME%TYPE;

LV\_CITY\_TXT PP\_EMPLOYEE.CITY%TYPE;

LV\_STATE\_TXT PP\_EMPLOYEE.STATE%TYPE;

BEGIN

EMP\_QUERY\_PKG.EMP\_INFO\_PP (LV\_ID\_NUM, LV\_FIRSTNAME\_TXT, LV\_CITY\_TXT, LV\_STATE\_TXT);

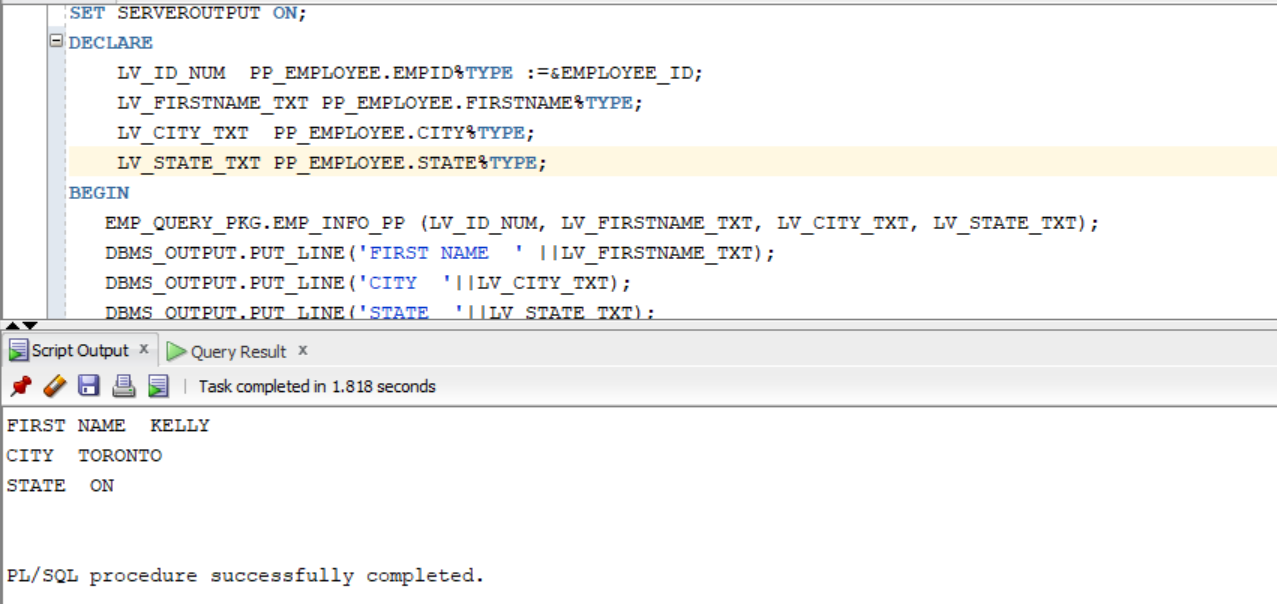
DBMS\_OUTPUT.PUT\_LINE('FIRST NAME ' ||LV\_FIRSTNAME\_TXT);

DBMS\_OUTPUT.PUT\_LINE('CITY '||LV\_CITY\_TXT);

DBMS\_OUTPUT.PUT\_LINE('STATE '||LV\_STATE\_TXT);

END;

/



SET SERVEROUTPUT ON;

CREATE OR REPLACE FUNCTION DISPLAYNAME

RETURN varchar2

IS

c\_ID pp\_employee.empid%type;

c\_FIRST pp\_employee.firstname%type;

c\_LASTNAME pp\_employee.lastname%type;

BEGIN

SELECT FIRSTNAME,LASTNAME

INTO

c\_FIRST,c\_LASTNAME

FROM PP\_EMPLOYEE

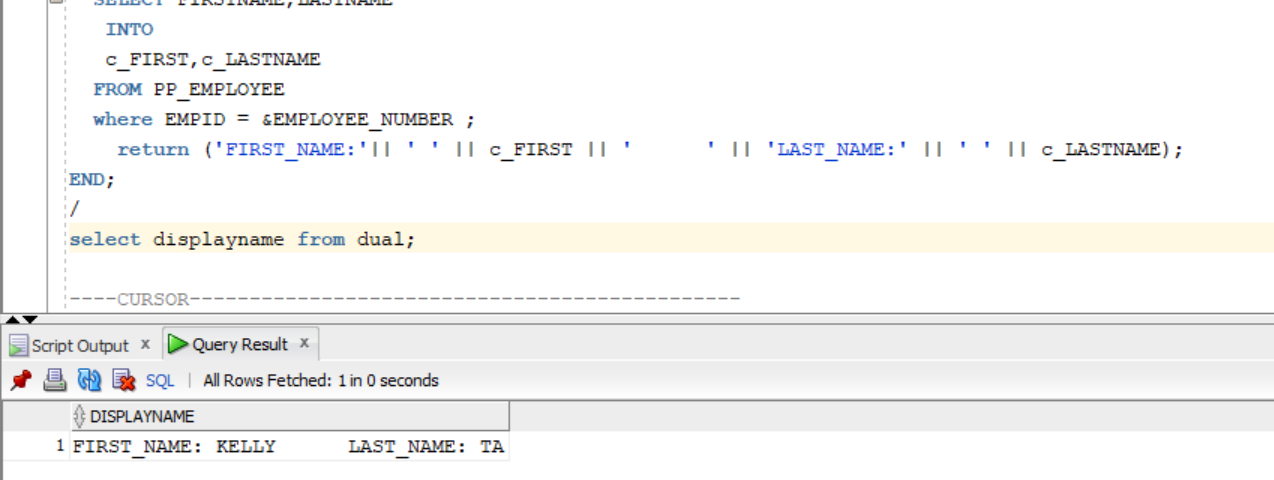
where EMPID = &EMPLOYEE\_NUMBER ;

return ('FIRST\_NAME:'|| ' ' || c\_FIRST || ' ' || 'LAST\_NAME:' || ' ' || c\_LASTNAME);

END;

/

select displayname from dual;



**----CURSOR----------------------------------------------**

SET SERVEROUTPUT ON;

DECLARE

TYPE type\_EMPLOYEE is TABLE OF PP\_EMPLOYEE%ROWTYPE

INDEX BY PLS\_INTEGER;

C1 type\_EMPLOYEE ;

begin

select \* BULK COLLECT into C1

from PP\_EMPLOYEE where EMPID=&n;

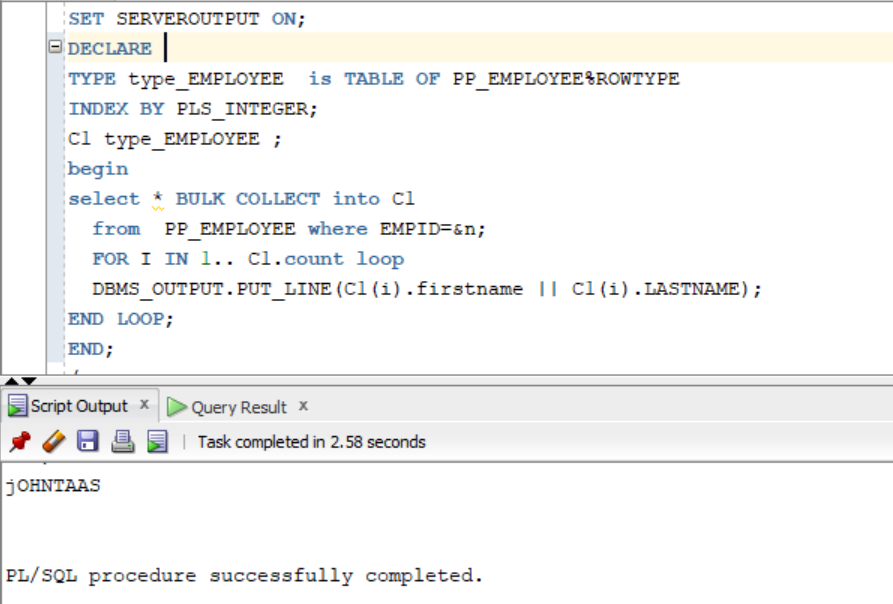
FOR I IN 1.. C1.count loop

DBMS\_OUTPUT.PUT\_LINE(C1(i).firstname || C1(i).LASTNAME);

END LOOP;

END;

/



**-- cursor based record variable**

set serveroutput on;

Declare

CURSOR C1 IS

SELECT EMPID,FIRSTNAME FROM PP\_EMPLOYEE

WHERE EMPID>909;

var\_cus c1%rowtype;

BEGIN

OPEN C1;

LOOP

FETCH C1 INTO var\_cus;

EXIT WHEN C1%NOTFOUND;

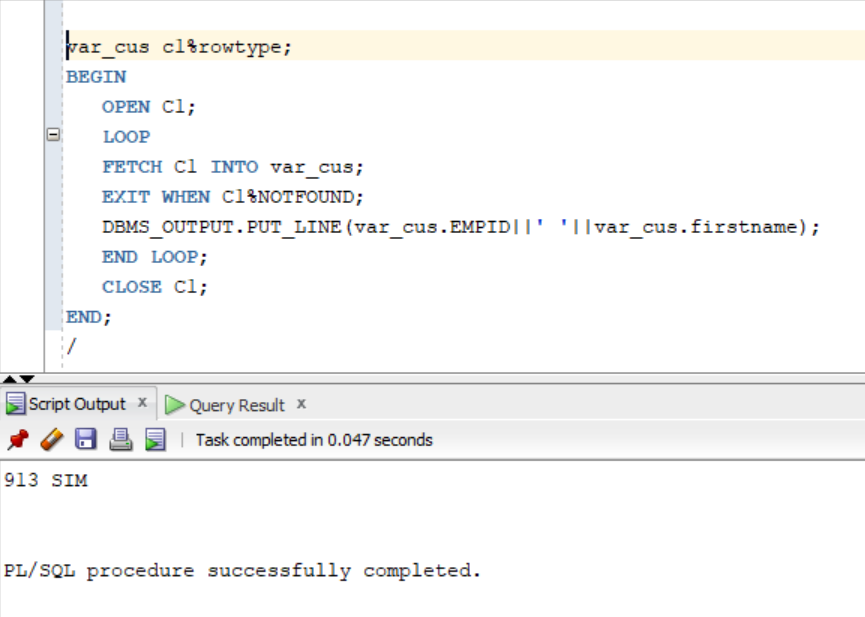
DBMS\_OUTPUT.PUT\_LINE(var\_cus.EMPID||' '||var\_cus.firstname);

END LOOP;

CLOSE C1;

END;

/



**--TRIGGER FOR EMPLOYEE----**

CREATE OR REPLACE TRIGGER EMP\_ONE\_trg

AFTER UPDATE OF SSN ON EMP\_EMPLOYEE

BEGIN

DBMS\_OUTPUT.PUT\_LINE('TRIGGER NUMBER 1 IS FIRED!');

END;

create or replace

trigger EMP\_TWO\_trg

AFTER UPDATE OF SSN ON PP\_EMPLOYEE

FOLLOWS EMP\_ONE\_trg

BEGIN

DBMS\_OUTPUT.PUT\_LINE('TRIGGER NUMBER TWO IS FIRED!');

END;

/

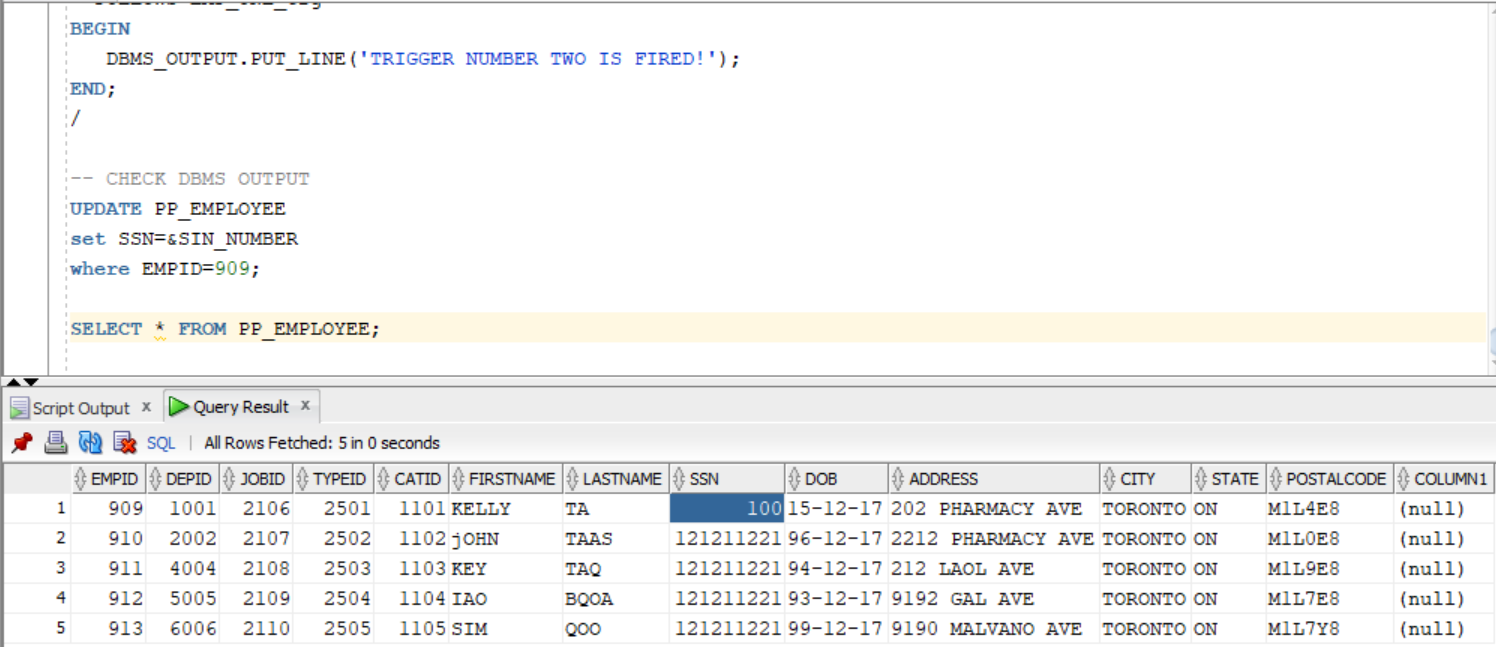
**-- CHECK DBMS OUTPUT**

UPDATE PP\_EMPLOYEE

set SSN=&SIN\_NUMBER

where EMPID=909;

SELECT \* FROM PP\_EMPLOYEE;



**TO DISPLAY TABLE CONTENT**

SELECT \* FROM EMPCATEGORY;

SELECT \* FROM EMPLOYEETYPE;

SELECT \* FROM JOBTITLE;

SELECT \* FROM PP\_BONUS;

SELECT \* FROM PP\_DEDUCTION;

SELECT \* FROM PP\_DEPARTMENT;

SELECT \* FROM PP\_EMPLOYEE;

SELECT \* FROM PP\_PAYROLL;

SELECT \* FROM PP\_SALARY;

SELECT \* FROM PP\_TIMESEET;

**DROPPING THE TABLES**

DROP TABLE PP\_EMPLOYEE CASCADE CONSTRAINTS;

DROP TABLE PP\_PAYROLL CASCADE CONSTRAINTS;

DROP TABLE PP\_TIMESEET CASCADE CONSTRAINTS;

DROP TABLE PP\_BONUS CASCADE CONSTRAINTS;

DROP TABLE PP\_DEPARTMENT CASCADE CONSTRAINTS;

DROP TABLE PP\_DEDUCTION CASCADE CONSTRAINTS;

DROP TABLE PP\_SALARY CASCADE CONSTRAINTS;

DROP TABLE EMPLOYEETYPE CASCADE CONSTRAINTS;

DROP TABLE JOBTITLE CASCADE CONSTRAINTS;

DROP TABLE EMPCATEGORY CASCADE CONSTRAINTS;

**DESCRIBING THE TABLE**

DESC PP\_EMPLOYEE;

DESC PP\_PAYROLL;

DESC PP\_TIMESEET;

DESC PP\_BONUS;

DESC PP\_DEPARTMENT;

DESC PP\_DEDUCTION;

DESC PP\_SALARY;

DESC EMPLOYEETYPE;

DESC EMPCATEGORY;

DESC JOBTITLE;

**DISPLAYING CONSTRAINTS**

SELECT \* FROM user\_constraints WHERE table\_name = 'PP\_EMPLOYEE';

SELECT \* FROM user\_constraints WHERE table\_name = 'PP\_PAYROLL';

SELECT \* FROM user\_constraints WHERE table\_name = 'PP\_TIMESEET';

SELECT \* FROM user\_constraints WHERE table\_name = 'PP\_BONUS';

SELECT \* FROM user\_constraints WHERE table\_name = 'PP\_DEPARTMENT';

SELECT \* FROM user\_constraints WHERE table\_name = 'PP\_DEDUCTION';

SELECT \* FROM user\_constraints WHERE table\_name = 'PP\_SALARY';

SELECT \* FROM user\_constraints WHERE table\_name = 'EMPLOYEETYPE';

SELECT \* FROM user\_constraints WHERE table\_name = 'EMPCATEGORY';

SELECT \* FROM user\_constraints WHERE table\_name = 'JOBTITLE';

**X\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*X**