**Project**

**Employee Payroll Management System**

**Name:** Akshaya Ganesh

**Name of Project:** Employee Payroll Management System

**Summary:**

1. Employee Information

Employee data is very essential in order to maintain a proper record of the employees and there personal information for various purposes like contacting them for inviting for certain summit, feedback of the company from the employee data

1. Maintaining Salary

Very important to keep this data which will help not only the managers and the HR to keep a track of the employee salaries but also help the company or its board to analyze what amount they are spending on a particular employee of a particular company

1. Work Location

It is very much important for an organization small or big to have a record of all the work locations they operate from to see how they can develop in that particular region and also increase the hiring in that region so that the organization can increase there Market Outreach that area.

1. Projects

In order to be successful company should be involved in various projects, so they also need to maintain the record of the salaries each employee is being paid for a particular type of project he/she is working on

**PL/SQL features used in the project:**

1. Created Explicit Cursors which shows the hourly pay of the employees associated with there Accounts and Ref cursor showing the employees who are a part of a particular department
2. Create a CDB and a PDB with users to manage the data according to the area of interest
3. Implement pre-defined exception cursor\_already\_open to demonstrate the understanding of the exceptional handling concept which shows what error will populate when we try to open a cursor which is already open
4. Also, created Relational, Inline and Materialized Views satisfying various business requirements
5. Created Index on AccountDetails table
6. Built an E-R Diagram to know how the entities are related in the payroll management system for any company

**List of Entities:**

**Employee**

Employee table will include all the personal details of the employee and would be very much cover overall information of that particular employee

**Salary**

Salary Table will cover all the current and previous salaries an employee had or currently has. This table will help a manager/ an HR to analyze which employee has been given promotion on which date or when did his salary grade changed

**Department**

Department Table maintains the data of the all the possible departments an employee can belong to

**Account Details**

Account Details Table will maintain the data regarding the accounts which the employee has connected with the company for his/her salary to be credited

**Attendance**

This table includes all the data of the employees attendance which includes the number of hours an employee has worked in a week

**Project**

This table includes the data of all the projects a particular company is working on or the projects on which the company is going to work in the future

**Education**

The Education Table keeps the track of the education of the employee including his degrees achieved until now

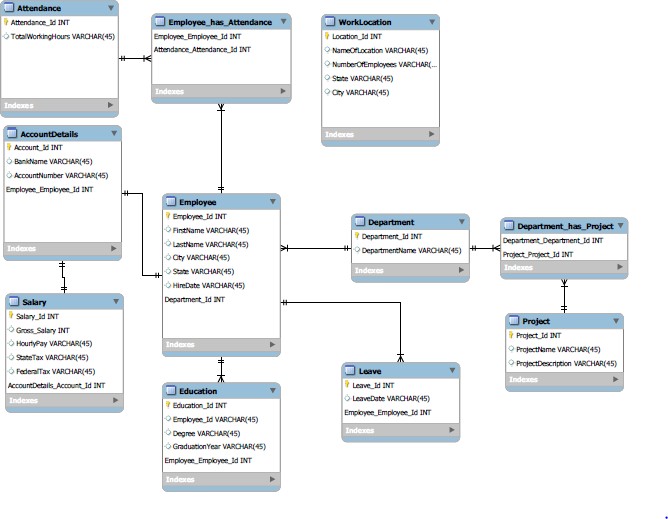
Work Location

The name of the table tells you most of the things. This table includes the location of the office, which city is it located, which state it is in and also tracks the number of employees in a particular location

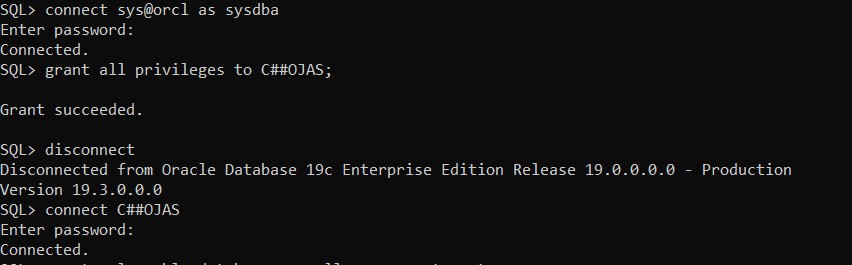
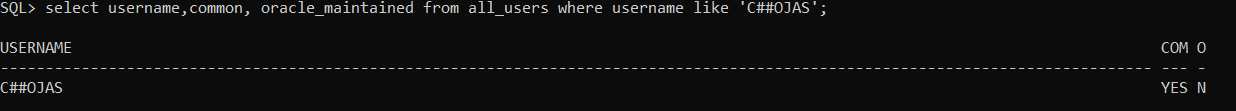
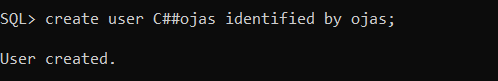
**Leave**

Leave table keeps the record of the number of leaves an employee takes or has taken over the course of any month or an year

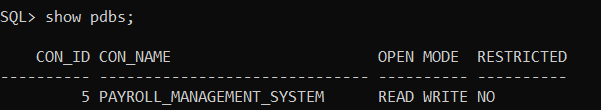
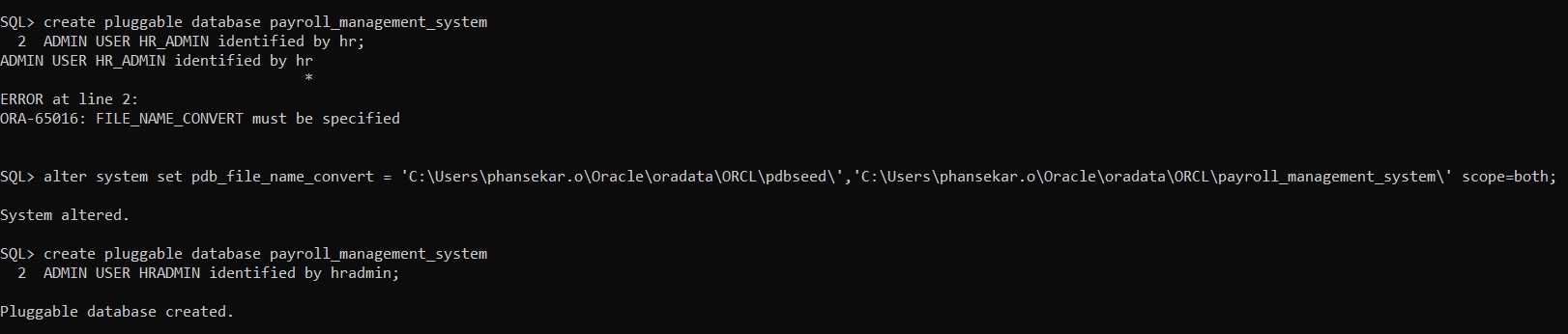
E-R Diagram

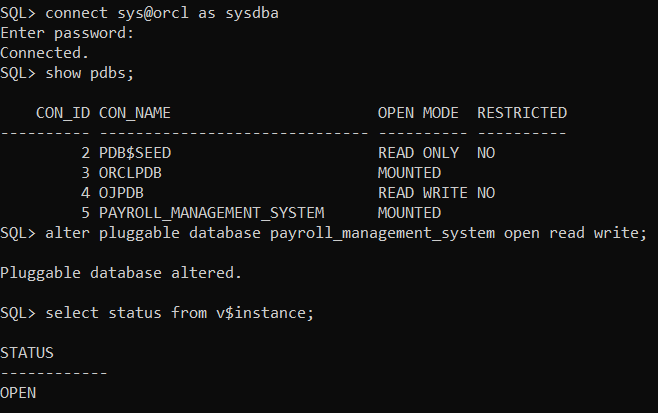


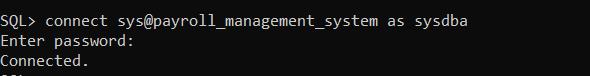
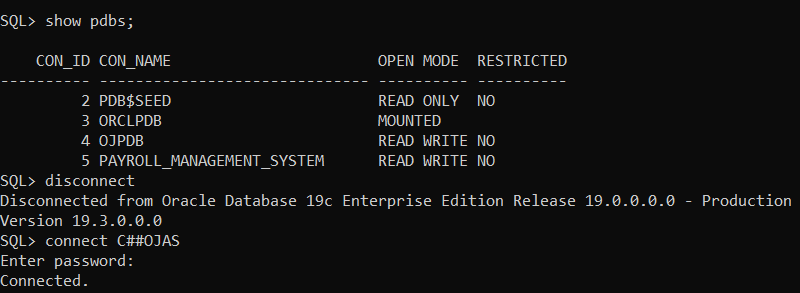
1. Created Common User on sysdba



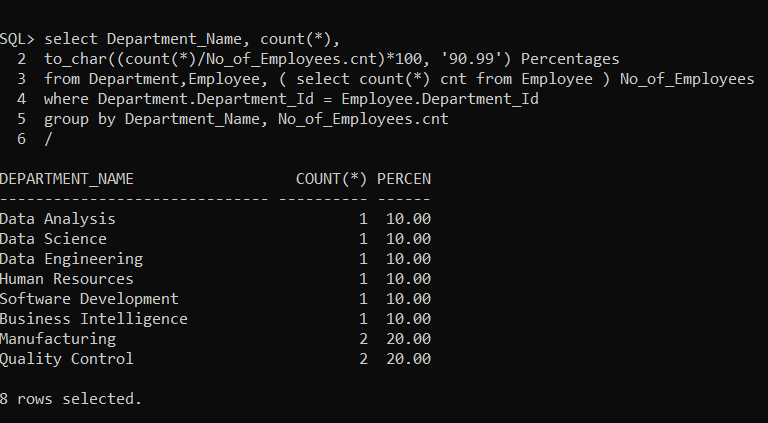
1. Create Pluggable Database





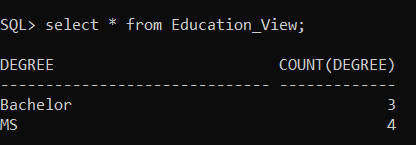


1. Inline View

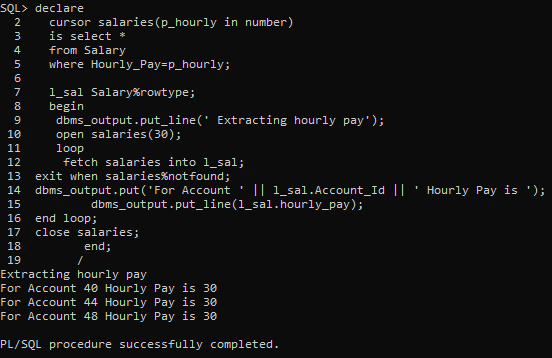


1. Materialized Views

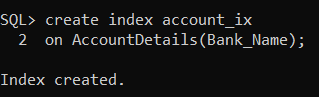
-- Number of Employees with different degrees



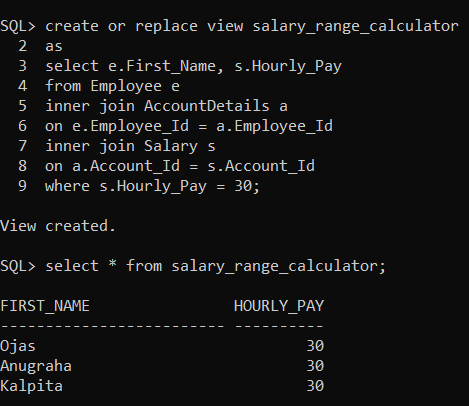
1. Explicit Cursor



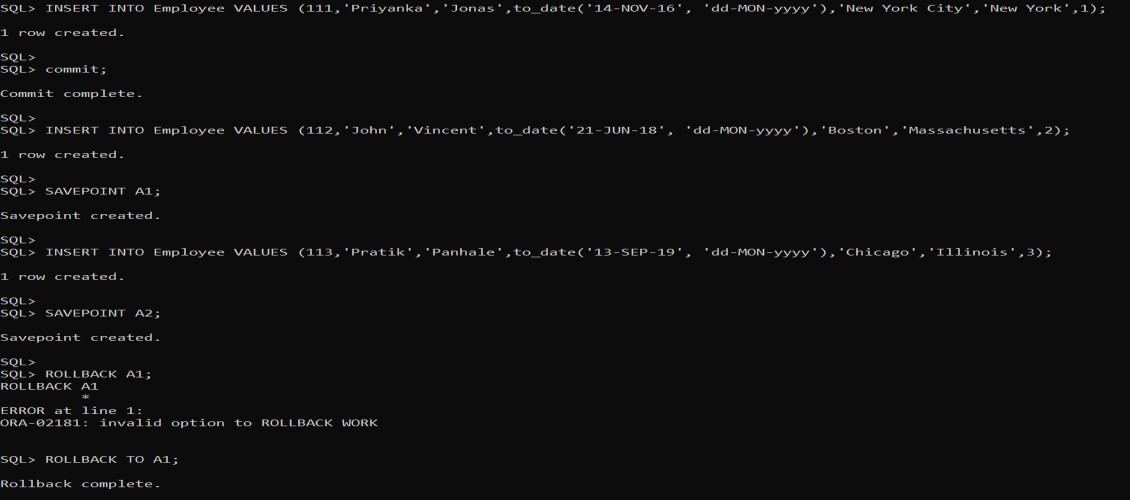
1. Index



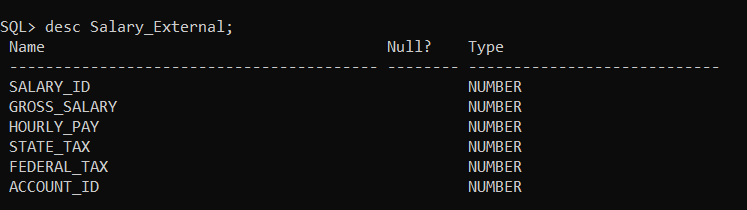
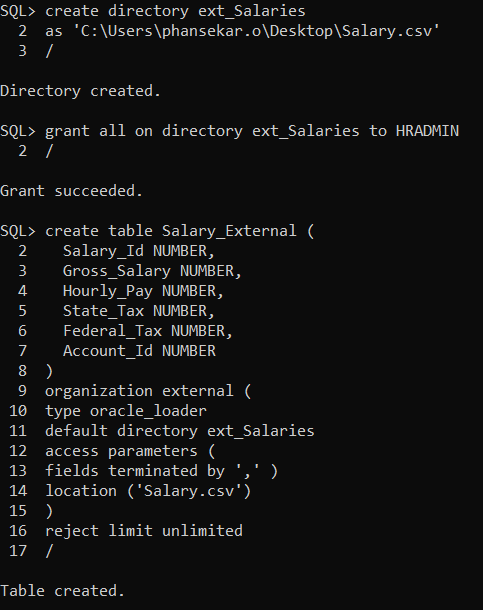
1. Relational Views



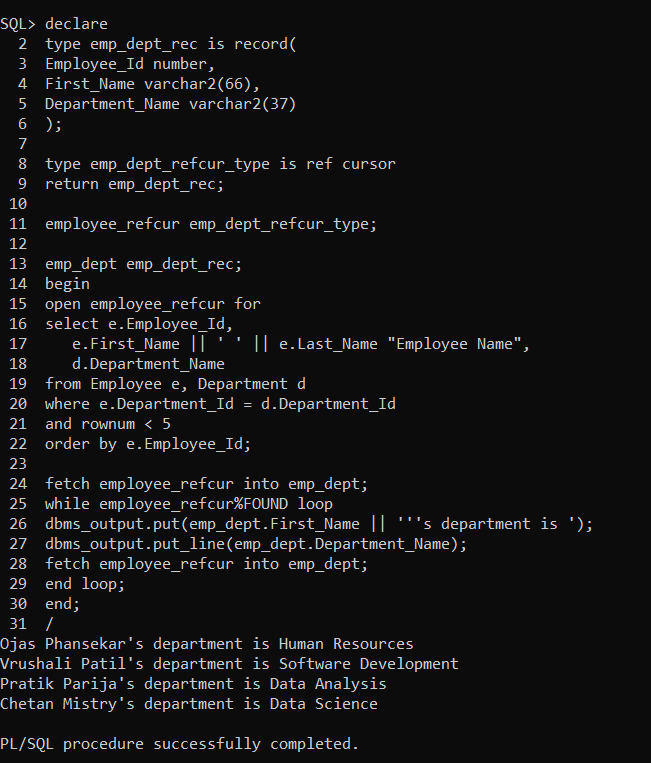
1. Transaction



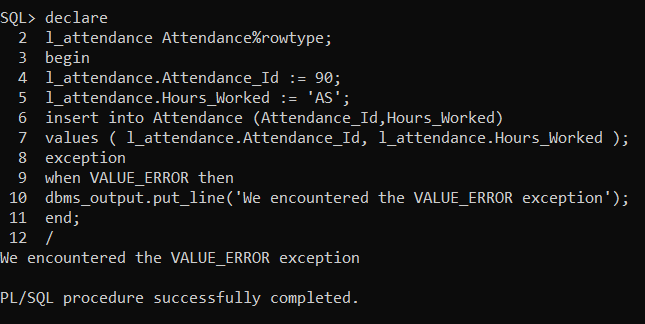
1. External Table

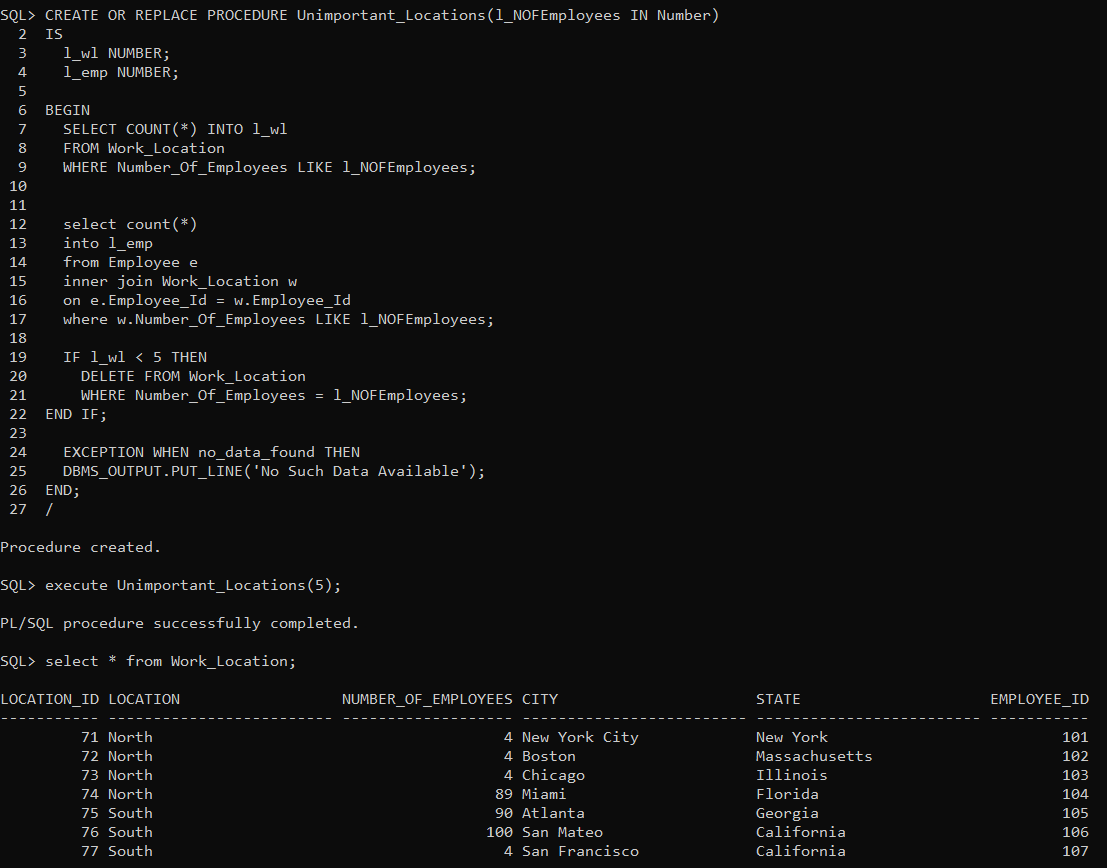


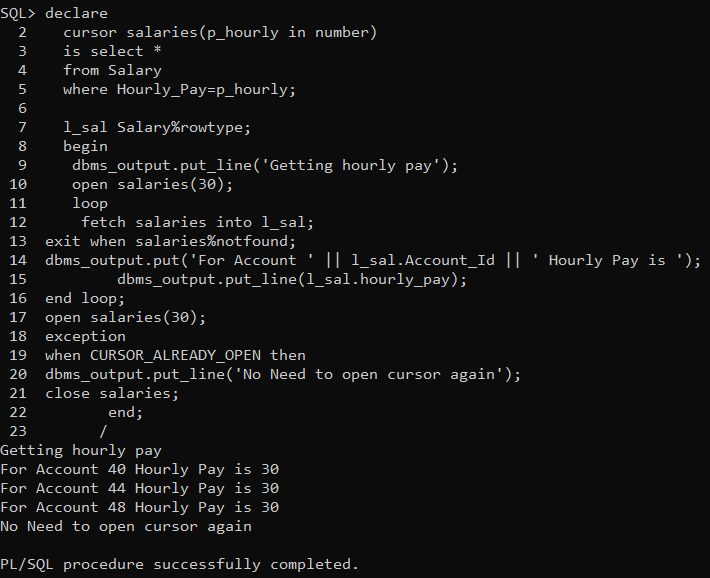
1. Ref cursor



1. Pre-defined Exception



1. Procedure
2. Predefined Exception and Explicit Cursor



**Appendix:**

# Create Table Statements

Employee

---------------------------------------------------

CREATE TABLE Employee( Employee\_Id NUMBER(6), First\_Name VARCHAR2(25), Last\_Name VARCHAR2(25), Hire\_Date DATE,

City VARCHAR2(25), State VARCHAR2(25),

CONSTRAINT EMPLOYEE\_PK PRIMARY KEY (Employee\_Id));

---------------------------------------------------

Department

---------------------------------------------------

CREATE TABLE Department( Department\_Id NUMBER, Department\_Name VARCHAR2(30),

CONSTRAINT DEPARTMENT\_PK PRIMARY KEY (Department\_Id)

);

-------------------------------------------------

Salary

-------------------------------------------------

CREATE TABLE Salary(

Salary\_Id NUMBER,

Gross\_Salary NUMBER, Hourly\_Pay NUMBER, State\_Tax NUMBER, Federal\_Tax NUMBER, Account\_Id NUMBER,

CONSTRAINT SALARY\_PK PRIMARY KEY (Salary\_Id),

FOREIGN KEY (Account\_Id)

REFERENCES ACCOUNTDETAILS(Account\_Id)

);

-------------------------------------------------

DepartmentProject Bridge

-------------------------------------------------

CREATE TABLE DepartmentProject( Department\_Id NUMBER, Project\_Id NUMBER,

CONSTRAINT DEPTPROJECT\_PK PRIMARY KEY (Department\_Id,Project\_Id), FOREIGN KEY (Department\_Id)

REFERENCES Department(Department\_Id), FOREIGN KEY (Project\_Id)

REFERENCES Project(Project\_Id)

);

--------------------------------------------------

Project

--------------------------------------------------

CREATE TABLE Project( Project\_Id NUMBER, Project\_Name VARCHAR2(50),

Project\_Description VARCHAR2(50),

CONSTRAINT Project\_PK PRIMARY KEY (Project\_Id)

);

---------------------------------------------------

AccountDetails

----------------- -

- - - -

- - -

CREATE TABLE AccountDetails( Account\_Id NUMBER, Bank\_Name VARCHAR2(50), Account\_Number VARCHAR2(50), Employee\_Id NUMBER,

CONSTRAINT Account\_PK PRIMARY KEY (Account\_Id), FOREIGN KEY (Employee\_Id)

REFERENCES Employee(Employee\_Id)

);

---------------------------------------------------

Education

---------------------------------------------------

CREATE TABLE Education( Education\_Id NUMBER, Employee\_Id NUMBER, Degree VARCHAR(30),

Graduation\_Year NUMBER(4),

CONSTRAINT Location\_PK PRIMARY KEY (Education\_Id), FOREIGN KEY (Employee\_Id)

REFERENCES Employee(Employee\_Id)

);

---------------------------------------------------

Leave

---------------------------------------------------

CREATE TABLE Leave(

Leave\_Id NUMBER, Employee\_Id NUMBER, Leave\_date DATE,

CONSTRAINT Leave\_PK PRIMARY KEY (Leave\_Id),

FOREIGN KEY (Employee\_Id) REFERENCES Employee(Employee\_Id)

);

----------------------------------------------------

EmployeeAttendance Bridge

----------------------------------------------------

CREATE TABLE Employee\_Attendance( Employee\_Id NUMBER, Attendance\_Id NUMBER,

CONSTRAINT DEPARTMENTPROJECT\_PK PRIMARY KEY (Employee\_Id,Attendance\_Id), FOREIGN KEY (Employee\_Id)

REFERENCES Employee(Employee\_Id), FOREIGN KEY (Attendance\_Id)

REFERENCES Attendance(Attendance\_Id)

);

----------------------------------------------------

Attendance

----------------------------------------------------

CREATE TABLE Attendance( Attendance\_Id NUMBER, Hours\_Worked NUMBER,

CONSTRAINT Attendance\_PK PRIMARY KEY (Attendance\_Id)

);

----------------------------------------------------

WorkLocation

----------------------------------------------------

CREATE TABLE Work\_Location( Location\_Id NUMBER, Location VARCHAR2(25),

Number\_Of\_Employees NUMBER, City VARCHAR2(25),

State VARCHAR2(25),

CONSTRAINT Loc\_PK PRIMARY KEY (Location\_Id)

);

# Insert Statements

INSERT INTO Employee VALUES (101,'Ojas','Phansekar',to\_date('14-APR-16', 'dd-MON-yyyy'),'New York City','New York',1);

INSERT INTO Employee VALUES (102,'Vrushali','Patil',to\_date('21-JUN-18', 'dd-MON- yyyy'),'Boston','Massachusetts',2);

INSERT INTO Employee VALUES (103,'Pratik','Parija',to\_date('13-SEP-19', 'dd-MON- yyyy'),'Chicago','Illinois',3);

INSERT INTO Employee VALUES (104,'Chetan','Mistry',to\_date('12-APR-11', 'dd-MON- yyyy'),'Miami','Florida',4);

INSERT INTO Employee VALUES (105,'Anugraha','Varkey',to\_date('16-AUG-17', 'dd-MON- yyyy'),'Atlanta','Georgia',5);

INSERT INTO Employee VALUES (106,'Rasagnya','Reddy',to\_date('25-JUL-18', 'dd-MON-yyyy'),'San Mateo','California',6);

INSERT INTO Employee VALUES (107,'Aishwarya','Boralkar',to\_date('18-DEC-10', 'dd-MON-yyyy'),'San Francisco','California',7);

INSERT INTO Employee VALUES (108,'Shantanu','Savant',to\_date('27-NOV-15', 'dd-MON- yyyy'),'Seattle','Washington',8);

INSERT INTO Employee VALUES (109,'Kalpita','Malvankar',to\_date('24-APR-16', 'dd-MON- yyyy'),'Boston','Massachusetts',8);

INSERT INTO Employee VALUES (110,'Saylee','Bhagat',to\_date('21-MAY-14', 'dd-MON-yyyy'),'San Francisco','California',7);

INSERT INTO Department VALUES (1,'Human Resources'); INSERT INTO Department VALUES (2,'Software Development'); INSERT INTO Department VALUES (3,'Data Analysis');

INSERT INTO Department VALUES (4,'Data Science');

INSERT INTO Department VALUES (5,'Business Intelligence'); INSERT INTO Department VALUES (6,'Data Engineering'); INSERT INTO Department VALUES (7,'Manufacturing'); INSERT INTO Department VALUES (8,'Quality Control');

INSERT INTO Project VALUES (21,'Dev','Whatever'); INSERT INTO Project VALUES (22,'Prod','do something'); INSERT INTO Project VALUES (23,'Test','focus');

INSERT INTO Project VALUES (24,'Nothing','do nothing');

INSERT INTO Project VALUES (25,'Research','focus on everything'); INSERT INTO Project VALUES (26,'Next Steps','find some way out');

INSERT INTO AccountDetails VALUES (40,'Santander','S12344',101); INSERT INTO AccountDetails VALUES (41,'Santander','S12345',102);

INSERT INTO AccountDetails VALUES (42,'Santander','S12346',103); INSERT INTO AccountDetails VALUES (43,'Santander','S12347',104); INSERT INTO AccountDetails VALUES (44,'Chase','C12344',105); INSERT INTO AccountDetails VALUES (45,'Chase','C12345',106); INSERT INTO AccountDetails VALUES (46,'Chase','C12347',107); INSERT INTO AccountDetails VALUES (47,'Chase','C12334',108); INSERT INTO AccountDetails VALUES (48,'BOFA','C12378',109); INSERT INTO AccountDetails VALUES (49,'BOFA','C12390',110);

INSERT INTO Education VALUES (10,101,'MS',2017); INSERT INTO Education VALUES (11,102,'MS',2019); INSERT INTO Education VALUES (12,104,'MS',2011); INSERT INTO Education VALUES (13,108,'MS',2015); INSERT INTO Education VALUES (14,109,'Bachelor',2013); INSERT INTO Education VALUES (15,107,'Bachelor',2008); INSERT INTO Education VALUES (16,106,'Bachelor',2007);

INSERT INTO Leave VALUES (51,104,to\_date('1-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (52,108,to\_date('2-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (53,109,to\_date('3-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (54,107,to\_date('4-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (55,106,to\_date('5-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (56,104,to\_date('6-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (57,108,to\_date('7-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (58,109,to\_date('7-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (59,107,to\_date('8-DEC-19', 'dd-MON-yyyy')); INSERT INTO Leave VALUES (60,106,to\_date('9-DEC-19', 'dd-MON-yyyy'));

INSERT INTO Attendance VALUES (90,10); INSERT INTO Attendance VALUES (91,20); INSERT INTO Attendance VALUES (92,30); INSERT INTO Attendance VALUES (93,40); INSERT INTO Attendance VALUES (94,45); INSERT INTO Attendance VALUES (95,56); INSERT INTO Attendance VALUES (96,58);

INSERT INTO Work\_Location VALUES (71,'North',4,'New York City','New York',101); INSERT INTO Work\_Location VALUES (72,'North',4,'Boston','Massachusetts',102); INSERT INTO Work\_Location VALUES (73,'North',4,'Chicago','Illinois',103);

INSERT INTO Work\_Location VALUES (74,'North',89,'Miami','Florida',104); INSERT INTO Work\_Location VALUES (75,'South',90,'Atlanta','Georgia',105);

INSERT INTO Work\_Location VALUES (76,'South',100,'San Mateo','California',106); INSERT INTO Work\_Location VALUES (77,'South',4,'San Francisco','California',107); INSERT INTO Work\_Location VALUES (78,'South',2,'Seattle','Washington',108); INSERT INTO Work\_Location VALUES (79,'South',25,'Alpharetta','Georgia',109); INSERT INTO Work\_Location VALUES (80,'South',20,'Keene','New Hampshire',110);

INSERT INTO Work\_Location VALUES (81,'South',22,'Hampton','New Hampshire',109);

INSERT INTO Employee\_Attendance VALUES (101,90); INSERT INTO Employee\_Attendance VALUES (102,91); INSERT INTO Employee\_Attendance VALUES (103,92); INSERT INTO Employee\_Attendance VALUES (104,93); INSERT INTO Employee\_Attendance VALUES (105,94); INSERT INTO Employee\_Attendance VALUES (106,95); INSERT INTO Employee\_Attendance VALUES (107,96); INSERT INTO Employee\_Attendance VALUES (108,91);

INSERT INTO Employee\_Attendance VALUES (109,92); INSERT INTO Employee\_Attendance VALUES (110,93);

INSERT INTO DepartmentProject VALUES (1,21); INSERT INTO DepartmentProject VALUES (2,22); INSERT INTO DepartmentProject VALUES (3,23); INSERT INTO DepartmentProject VALUES (4,24); INSERT INTO DepartmentProject VALUES (5,25); INSERT INTO DepartmentProject VALUES (6,26); INSERT INTO DepartmentProject VALUES (7,21); INSERT INTO DepartmentProject VALUES (8,24);

INSERT INTO Salary VALUES (1,57600,30,200,1000,40); INSERT INTO Salary VALUES (2,76800,40,300,1300,41); INSERT INTO Salary VALUES (3,96000,50,400,1500,42); INSERT INTO Salary VALUES (4,115200,60,500,1700,43); INSERT INTO Salary VALUES (5,57600,30,200,1000,44); INSERT INTO Salary VALUES (6,76800,40,300,1300,45); INSERT INTO Salary VALUES (7,96000,50,400,1500,46); INSERT INTO Salary VALUES (8,115200,60,500,1700,47); INSERT INTO Salary VALUES (9,57600,30,200,1000,48); INSERT INTO Salary VALUES (10,76800,40,300,1300,49);

# Inline View

select Department\_Name, count(\*), to\_char((count(\*)/No\_of\_Employees.cnt)\*100, '90.99') Percentages

from Department,Employee, ( select count(\*) cnt from Employee ) No\_of\_Employees where Department.Department\_Id = Employee.Department\_Id

group by Department\_Name, No\_of\_Employees.cnt

/

# Materialized View

*Number of Employees with different degrees*

---------------------------------------------

create materialized view Education\_View build immediate

refresh on commit as

select Degree, count(Degree) from Education

group by Degree;

# Procedure

*Locations with less number of employees*

CREATE OR REPLACE PROCEDURE Unimportant\_Locations(l\_NOFEmployees IN Number) IS

l\_wl NUMBER; l\_emp NUMBER;

BEGIN

SELECT COUNT(\*) INTO l\_wl

FROM Work\_Location

WHERE Number\_Of\_Employees LIKE l\_NOFEmployees;

select count(\*) into l\_emp

from Employee e

inner join Work\_Location w

on e.Employee\_Id = w.Employee\_Id

where w.Number\_Of\_Employees LIKE l\_NOFEmployees;

IF l\_wl < 5 THEN

DELETE FROM Work\_Location

WHERE Number\_Of\_Employees = l\_NOFEmployees; END IF;

EXCEPTION WHEN no\_data\_found THEN DBMS\_OUTPUT.PUT\_LINE('No Such Data Available'); END;

# Explicit Cursor

declare

cursor salaries(p\_hourly in number) is select \*

from Salary

where Hourly\_Pay=p\_hourly;

l\_sal Salary%rowtype; begin

dbms\_output.put\_line(' Extracting hourly pay'); open salaries(30);

loop

fetch salaries into l\_sal;

exit when salaries%notfound;

dbms\_output.put('For Account ' || l\_sal.Account\_Id || ' Hourly Pay is '); dbms\_output.put\_line(l\_sal.hourly\_pay);

end loop; close salaries;

end;

/

# Pre-Defined Exception

declare

l\_attendance Attendance%rowtype; New\_Exception exception;

begin

l\_attendance.Attendance\_Id := 90; l\_attendance.Hours\_Worked := 'AS';

insert into Attendance (Attendance\_Id,Hours\_Worked)

values ( l\_attendance.Attendance\_Id, l\_attendance.Hours\_Worked ); exception

when VALUE\_ERROR then

dbms\_output.put\_line('We encountered the VALUE\_ERROR exception');

end;

/

# Explicit Cursor and Pre-Defined Cursor Together

declare

cursor salaries(p\_hourly in number) is select \*

from Salary

where Hourly\_Pay=p\_hourly;

l\_sal Salary%rowtype; begin

dbms\_output.put\_line('Getting hourly pay'); open salaries(30);

loop

fetch salaries into l\_sal;

exit when salaries%notfound;

dbms\_output.put('For Account ' || l\_sal.Account\_Id || ' Hourly Pay is '); dbms\_output.put\_line(l\_sal.hourly\_pay);

end loop;

open salaries(30); exception

when CURSOR\_ALREADY\_OPEN then dbms\_output.put\_line('No Need to open cursor again'); close salaries;

end;

/

# External Table

create table Salary\_External ( Salary\_Id NUMBER, Gross\_Salary NUMBER, Hourly\_Pay NUMBER, State\_Tax NUMBER, Federal\_Tax NUMBER, Account\_Id NUMBER

)

organization external (

type oracle\_loader

default directory ext\_Salaries access parameters (

fields terminated by ',' ) location ('Salary.csv')

)

reject limit unlimited

/

# Ref Cursor

declare

type emp\_dept\_rec is record( Employee\_Id number, First\_Name varchar2(66), Department\_Name varchar2(37)

);

type emp\_dept\_refcur\_type is ref cursor return emp\_dept\_rec;

employee\_refcur emp\_dept\_refcur\_type;

emp\_dept emp\_dept\_rec;

begin

open employee\_refcur for

select e.Employee\_Id,

e.First\_Name || ' ' || e.Last\_Name "Employee Name", d.Department\_Name

from Employee e, Department d

where e.Department\_Id = d.Department\_Id and rownum < 5

order by e.Employee\_Id;

fetch employee\_refcur into emp\_dept; while employee\_refcur%FOUND loop

dbms\_output.put(emp\_dept.First\_Name || '''s department is '); dbms\_output.put\_line(emp\_dept.Department\_Name);

fetch employee\_refcur into emp\_dept; end loop;

end;

/

# Transaction

INSERT INTO Employee VALUES (111,'Priyanka','Jonas',to\_date('14-NOV-16', 'dd-MON-yyyy'),'New York City','New York',1);

commit;

INSERT INTO Employee VALUES (112,'John','Vincent',to\_date('21-JUN-18', 'dd-MON- yyyy'),'Boston','Massachusetts',2);

SAVEPOINT A1;

INSERT INTO Employee VALUES (113,'Pratik','Panhale',to\_date('13-SEP-19', 'dd-MON- yyyy'),'Chicago','Illinois',3);

SAVEPOINT A2;

ROLLBACK A1;

# Relational View

create or replace view salary\_range\_calculator as

select e.First\_Name, s.Hourly\_Pay from Employee e

inner join AccountDetails a

on e.Employee\_Id = a.Employee\_Id inner join Salary s

on a.Account\_Id = s.Account\_Id where s.Hourly\_Pay = 30;