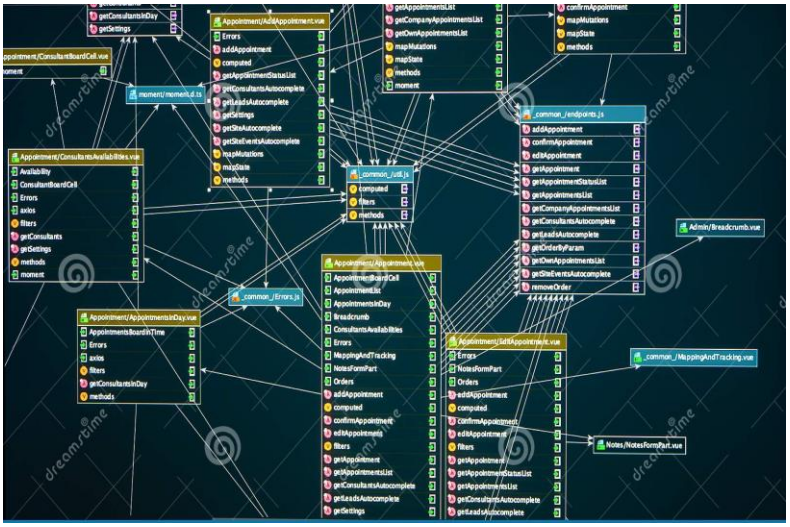


CIS 5430 – Database and Data Warehousing Project
Company Database Design

Professor Ming Wang

Submitted by:

Group 4

Shailja Pandit (Team Leader)
Pooja Madhup (Developer)
Anish Omprakash Pandey (Developer)
Carlos Madrigal Alvarez (Developer)

Contents

Company Database- Need

This database will provide an improved method of keeping important information organized and properly stored, in order to use it when needed. At this moment, the Database Management System in use is presenting a significant number of errors, and since these errors can carry more problems, the decision to migrate all the critical data was taken.

With the implementation of this new Database System, the company will be able to keep track of the different departments that the organization has, also, it will provide a method to store information related to projects, either new or current ones. As in all Database System, the option to save and query details about the company's employees is available, and with this, not only the person who does a job will be in the system, but also their dependents, this to provide coverage for their health insurance.

As mentioned before, there is a need for the company to maintain their data centralized, and with this new database that will be possible. Databases are very powerful tools and are used to organize data, create databases, and control data using query languages. One of the main benefits of a database is that it makes storage of information quick and easy. Some of the other reasons why a database management system is important are improved security, efficiency, and are accessible to multiple users. Also, databases can be scaled according to future use.

Company Database Requirements

1. The company is organized into departments. For these departments we have the following business rules:
2. Each department has a unique name.
3. Each department has a unique number.
4. Each department may have several locations.
5. Each department has a particular employee who manages the department.
6. We keep track of the start date when the manager began managing the department.
7. A department controls a number of projects.
8. Each project has a unique name.
9. Each project has a unique number.
10. Each project may have more than one category.
11. One project must be either an in-house project or an out-source project.
12. We keep track of the number of hours per week that an employee works on each in-house project.
13. For out-source projects we store:
    a. Out-source project ID
    b. Company Name
    c. Rate
14. For employees we store each employee's:
    a. Name
    b. Social security number
    c. Address
    d. Salary

    e. Sex
    f. Birthday
15. An employee is assigned to one department but may work on several in-house projects, which are not necessarily controlled by the same department.
16. We also keep track of the direct supervisor of each employee.
17. We want to keep track of the dependents of each employee for insurance purposes. From these we keep each dependent's:
    a. Name
    b. Sex
    c. Birthday
    d. Relationship to the employees

Company Database Function

During the planning step, we'll use a variety of technologies to assist us in creating a well-structured database. ERD plus wall used for creating EER diagram for conceptual design. The group members collaborated via teams. All the group meetings were set through Teams. The group was led by Shailja Pandit, she was also responsible for Functional Dependencies and Normalization form of the tables. Referential Integrity Constraints and Relational Database tables were done by Pooja. Analyzing Business Rules and Creating Relationship Matrix was done by Carlos. Conceptual Design using ERD plus was initiated by Anish. This transactional data, as well as associated tables, must be kept and organized in a way that allows analysts to gain insight and business intelligence while making well-informed business decisions. We all collaboratively worked on Creating Tables , DML commands ,PL/SQL queries and ORDBMS.

Project Purpose

The purpose of having Company Database is to communicate information related to Employees and departments effectively. It can be used for Updating, Storing, Deleting and Retrieval of data that can be used for business visualization and reports.

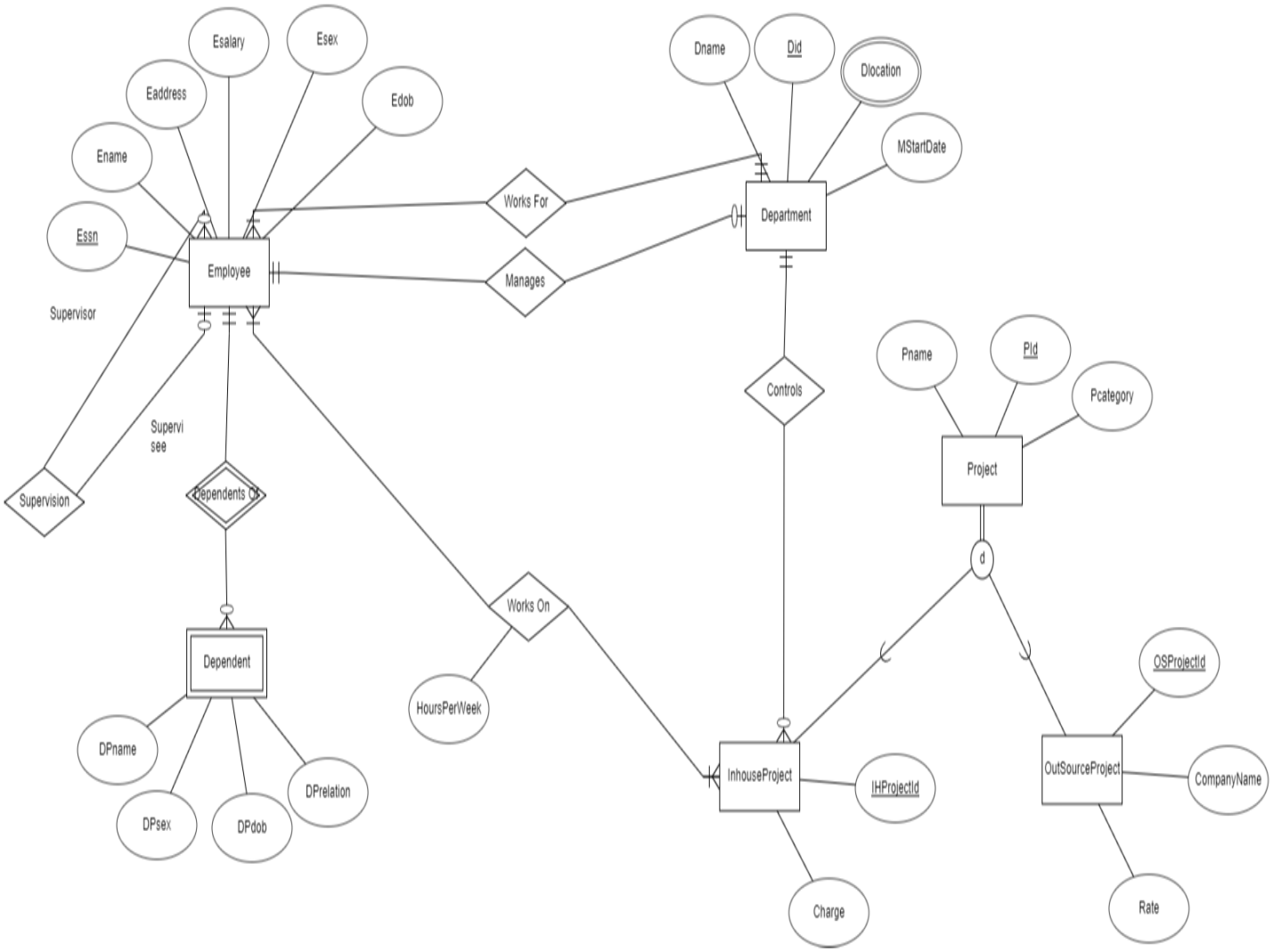
Relationship Matrix

	Employee	**Department**	**Dependent**	**Project (InHouse/OutSource)**
Employee	May Supervise/ May be supervised	Must Have/Must Be Managed	Must belong	Must Have (InHouse)
Department	Must Work/May Manage	--	--	Must be Controlled (InHouse)
Dependent	May Have		--	--

Project (InHouse/Out Source)	Must Work (InHouse)	May Control (InHouse)	--	Must be Either Inhouse or Outsource

ERD Diagram

ER/EER diagram showing keys and attributes of entity types, relationship types, cardinality, and participation is displayed below:



Relational Database

Table name: **Dependent/DEPEND**

EID (FK)	DPname	DPsex	DPdob	DPrelation

Table name: **Employee/EMP**

EID(PK)	Essn	EFName	ELName	EAddress	ECity	EState	Ezip	Esalary	Esex	Edob	**Did(FK)**	**Supervisor(FK**

Table name: **Department/DEPT**

Did(PK)	Dname	**DManager(FK)**	MStartDate

Table name: **DeptLocation/DEPTLOC**

Did(FK)	**DLocID(PK)**	DLocAddress	DLocPhone

Table name: **Project/PROJ**

Pid(PK)	Pname	Pcategory	PLocation

Table name: **InHouseProject/IHPROJ**

IHProjId(PK,FK)	Charge	**Did(FK)**	

Table name: **OutSourceProject/OSPROJ**

OHProjId(PK,FK)	CompanyName	Rate	

Table name: **EmployeeInhouse/ EMPIHPROJ**

EmpIHProjID (PK)	**EID(FK)**	**IHProjId (FK)**	HoursPerWeek

Referential Integrity

Set up all the referential integrity constraints in the relational schema after the transformation.
F.K. TableName.ColumnName -> P.K. TableName.ColumnName

F.K. Department.DManager→P.K. Employee.EID

F.K. Employee.Did → P.K. Department.Did
F.K. Employee.Supervisor →P.K. Employee.EID
F.K. Dependent.EID →P.K. Employee.EID
F.K. InHouseProject.Did →P.K. Department.Did
F.K. InHouseProject.IHProjId →P.K. Project.Pid
F.K. EmployeeInhouse.Pid →P.K. InHouseProject.IHProjId
F.K. EmployeeInhouse.EID →P.K. Employee.EID
F.K. DeptLocation.Did →P.K. Department.Did
F.K OutSourceProject.OSProjId→ PK Project.Pid

Functional Dependency

Show functional dependency analysis for all the tables in the relational schema.

Full Dependency

Dependent/DEPEND
EID, DPname → DPrelation, DPdob, DPsex

Employee/EMP
EID → Essn, EFName, ELName,EAddress, EState, ECity, Ezip, Esalary,Esex ,Edob, Supervisor
Department
Did → Dname, Dmanager, MStartDate

DeptLocation/DEPTLOC
Did,DLocID → DLocAddress,DLocPhone

Project/PROJ
Pid → Pname, Plocation,Pcategory

InHouseProject/IHPROJ
IHProjId → Charge,Did

OutSourceProject/OHPROJ
OSProjId → CompanyName, Rate

EmployeeInHouse/EMPIHPROJ
EmpIHProjID → HourPerWeek, Eid,Pid

Transitive Dependency

Department
DManager → MStartDate

Employee
Ezip → Eaddress,Estate,Ecity

Relational Schema after Normalization

Indicate what normalization form each table is in.

Table Name	1NF	2NF	3NF
Employee	X	X	
Department	X	X	
DeptLocation	X	X	X
Dependent	X	X	X
Project	X	X	X
InHouseProject	X	X	X
OutSourceProject	X	X	X
EmployeeInHouse	X	X	X

Normalization Form

Normalized the tables in the relational schema.

Table name: **DeptLocation/DEPTLOC** **(1 NF, 2 NF, 3 NF)**

Did(FK)	DLocID(PK)	DLocAddress	DLocPhone

Table name: **Project/PROJ** **(1 NF, 2 NF, 3 NF)**

Pid(PK)	Pname	Pcategory	PLocation

Table name: **InHouseProject/IHPROJ** **(1 NF, 2 NF, 3 NF)**

IHProjId(PK,FK)	Charge	Did(FK)	

Table name: **OutSourceProject/OSPROJ** **(1 NF, 2 NF, 3 NF)**

OHProjId(PK,FK)	CompanyName	Rate	

Table name: **EmployeeInhouse/EMPIHPROJ** **(1 NF, 2 NF, 3 NF)**

EmpIHProjId(PK)	EID(FK)	Pid(FK)	HoursPerWeek	

Table name: **Dependent/DEPEND** **(1 NF, 2 NF, 3 NF)**

EID (FK)	DPname	DPsex	DPdob	DPrelation

Table name: **Department/DEPT**
(1 NF, 2 NF)

Did(PK)	Dname	DManager (FK)	MStartDate

Department Table is not in 3rd normal form, since it contains Transitive dependencies

Did(PK)	Dname	DManager(FK)

DManager(PK)	MStartDate

Table name: **Employee/EMP** **(1 NF, 2 NF)**

EID(PK)	Essn	EFName	ELName	EAddress	ECity	EState	Ezip	Esalary	Esex	Edob	Did(FK)	Supervisor(FK)

Employee Table is not in 3rd normal form, since it contains Transitive dependencies

EID(PK)	Essn	EFName	ELName	Ezip(FK)	Esalary	Esex	Edob	Did(FK)	Supervisor(FK)

Ezip(PK)	ECity	EState	EAddress

Database Creation

Project **(Pooja Madhup)**

```
CREATE TABLE Proj
(
  Pname VARCHAR2(30) NOT NULL,
  PId NUMBER(5) NOT NULL,
  Pcategory VARCHAR2(7),
  PLocation VARCHAR2(15),
  CONSTRAINT proj_pid_pk PRIMARY KEY (PId),
  CONSTRAINT proj_pname_uk unique (pname)
);
```

```
CREATE TABLE PROJ
(
  Pname VARCHAR2(30) NOT NULL,
  Pld NUMBER(5) NOT NULL,
  Pcategory VARCHAR2(7),
  PLocation VARCHAR2(15),
  CONSTRAINT proj_pid_pk PRIMARY KEY (Pld),
  CONSTRAINT proj_pname_uk unique (pname)
);
```

Execute Load Script Save Script Cancel

Table created.

OutSourceProject (Anish Omprakash Pandey)

```
CREATE TABLE OSPROJ
(
  OSProjId NUMBER NOT NULL,
  CompanyName VARCHAR2(15),
  Rate NUMBER(3),
  CONSTRAINT osproj_osprojid_pk PRIMARY KEY (OSProjId),
  CONSTRAINT osproj_osprojid_fk FOREIGN KEY (OSProjId) REFERENCES PROJ(PId)
);
```

```
CREATE TABLE OSPROJ
(
  OSProjId NUMBER NOT NULL,
  CompanyName VARCHAR2(15),
  Rate NUMBER(3),
  CONSTRAINT osproj_osprojid_pk PRIMARY KEY (OSProjId),
  CONSTRAINT osproj_osprojid_fk FOREIGN KEY (OSProjId) REFERENCES
PROJ(PId)
);
```

Execute Load Script Save Script Cancel

Table created.

InhouseProject (Anish Omprakash Pandey)

```
CREATE TABLE IHPROJ
(
  IHProjId NUMBER(5) NOT NULL,
  Charge NUMBER(5,2),
  Did NUMBER(3) NOT NULL,
  CONSTRAINT ihproj_ihprojid_pk PRIMARY KEY (IHProjId),
  CONSTRAINT ihproj_ihprojid_fk FOREIGN KEY (IHProjId) REFERENCES PROJ(PId),
  CONSTRAINT ihproj_did_fk FOREIGN KEY (Did) REFERENCES DEPT(Did)
);
```

```
CREATE TABLE IHPROJ
(
  IHProjId NUMBER(5) NOT NULL,
  Charge NUMBER(5,2),
  Did NUMBER NOT NULL,
  CONSTRAINT ihproj_ihprojid_pk PRIMARY KEY (IHProjId),
  CONSTRAINT ihproj_ihprojid_fk FOREIGN KEY (IHProjId) REFERENCES
PROJ(Pld),
  CONSTRAINT ihproj_did_fk FOREIGN KEY (Did) REFERENCES DEPT(Did)
);
```

Execute Load Script Save Script Cancel

Table created.

Employee (Shailja Pandit)

```
CREATE TABLE EMP
(
  EID NUMBER(4) NOT NULL,
  Esex CHAR(1),
  Esalary NUMBER(7,2),
  Eaddress VARCHAR2(20),
  Ecity VARCHAR2(10),
  Estate VARCHAR2(3),
  Ezip NUMBER(6),
  Edob DATE,
  Essn NUMBER(9),
  Did NUMBER(3),
  Supervisor NUMBER(5),
  ELname VARCHAR2(10),
  EFname VARCHAR2(10) NOT NULL,
  CONSTRAINT emp_eid_pk PRIMARY KEY (EID),
  CONSTRAINT emp_supervisor_fk FOREIGN KEY (Supervisor) REFERENCES EMP(EID)
);
```

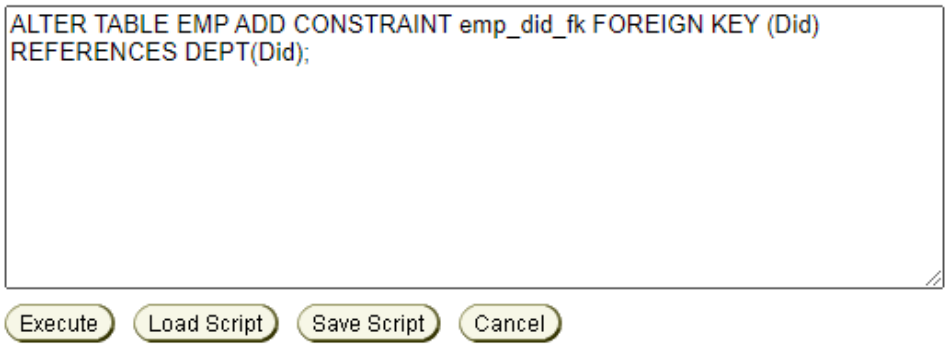
```
CREATE TABLE EMP
(
  EID NUMBER(4) NOT NULL,
  Ename VARCHAR2(10),
  Esex CHAR(1),
  Esalary NUMBER(7,2),
  Eaddress VARCHAR2(20),
  Ecity VARCHAR2(10),
  Estate VARCHAR2(3),
  Ezip NUMBER(6),
```

Execute Load Script Save Script Cancel

Table created.

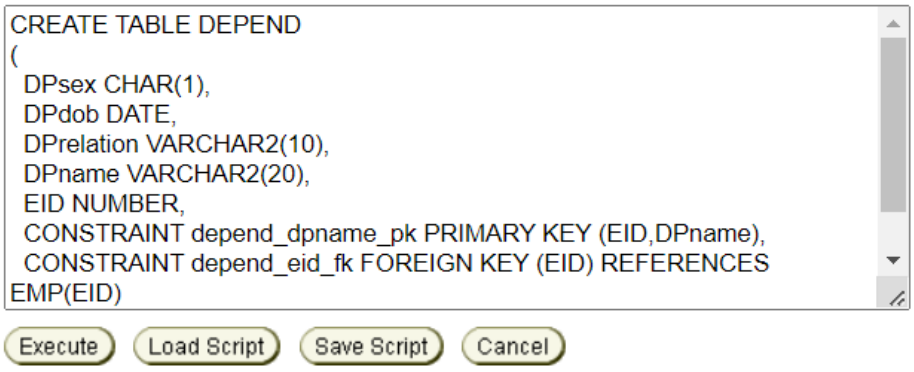
ALTER QUERY

ALTER TABLE EMP ADD CONSTRAINT emp_did_fk FOREIGN KEY (Did) REFERENCES DEPT(Did);



Dependent (Shailja Pandit)

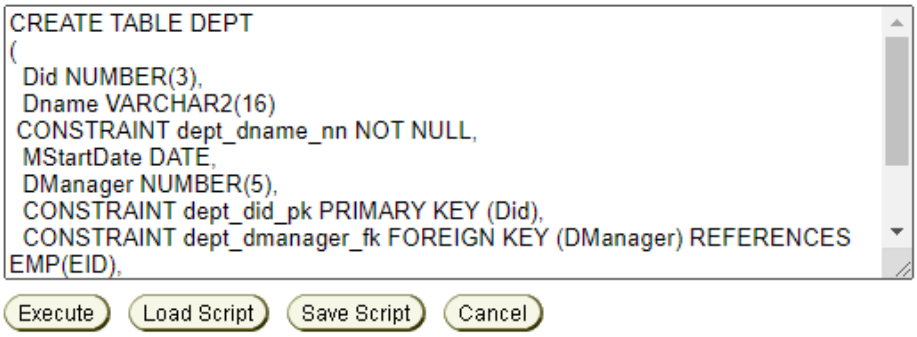
```
CREATE TABLE DEPEND
(
  DPsex CHAR(1),
  DPdob DATE,
  DPrelation VARCHAR2(10),
  DPname VARCHAR2(20),
  EID NUMBER,
  CONSTRAINT depend_dpname_pk PRIMARY KEY (EID,DPname),
  CONSTRAINT depend_eid_fk FOREIGN KEY (EID) REFERENCES EMP(EID)
);
```



Department (Pooja Madhup)

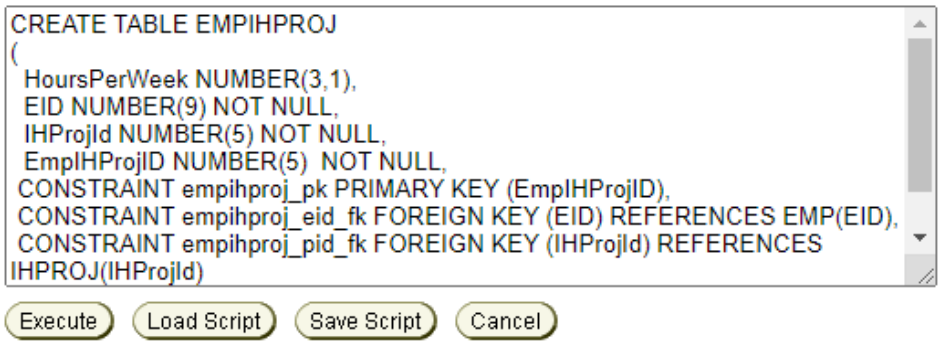
```
CREATE TABLE DEPT
(
  Did NUMBER(3),
  Dname VARCHAR2(16)
  CONSTRAINT dept_dname_nn NOT NULL,
```

```
 MStartDate DATE,
 DManager NUMBER(5),
 CONSTRAINT dept_did_pk PRIMARY KEY (Did),
 CONSTRAINT dept_dmanager_fk FOREIGN KEY (DManager) REFERENCES EMP(EID),
 CONSTRAINT   dept_did_ck check (did between 10 and 99),
 CONSTRAINT dept_dname_uk unique (Dname)
);
```



EmployeeInhouse (Carlos M Mardrigal)

```
CREATE TABLE EMPIHPROJ
(
 HoursPerWeek NUMBER(3,1),
 EID NUMBER(9) NOT NULL,
 IHProjId NUMBER(5) NOT NULL,
 EmpIHProjID NUMBER(5)  NOT NULL,
CONSTRAINT empihproj_pk PRIMARY KEY (EmpIHProjID),
CONSTRAINT empihproj_eid_fk FOREIGN KEY (EID) REFERENCES EMP(EID),
CONSTRAINT empihproj_pid_fk FOREIGN KEY (IHProjId) REFERENCES IHPROJ(IHProjId)
);
```



DeptLocation (Carlos M Mardrigal)

```
CREATE TABLE DEPTLOC
(
```

```
  DLocID NUMBER(3) NOT NULL,
  DLocAddress VARCHAR2(40),
  DLocPhone NUMBER(10),
  Did NUMBER(3) NOT NULL,
  CONSTRAINT deptloc_pk PRIMARY KEY (DLocID),
  CONSTRAINT deptloc_did_fk FOREIGN KEY (Did) REFERENCES DEPT(Did)
);
```

```
CREATE TABLE DEPTLOC
(
  DLocID NUMBER(3) NOT NULL,
  DLocAddress VARCHAR2(15),
  DLocPhone NUMBER(10),
  Did NUMBER(3) NOT NULL,
  CONSTRAINT deptloc_pk PRIMARY KEY (DLocID),
  CONSTRAINT deptloc_did_fk FOREIGN KEY (Did) REFERENCES DEPT(Did)
);
```

Execute | Load Script | Save Script | Cancel

Table created.

Database Structure

Project (Pooja Madhup)

DESC PROJ;

Name	Null?	Type
PNAME	NOT NULL	VARCHAR2(30)
PID	NOT NULL	NUMBER(5)
PCATEGORY		VARCHAR2(7)
PLOCATION		VARCHAR2(15)

OutSourceProject (Anish Omprakash Pandey)

DESC OSPROJ;

Name	Null?	Type
OSPROJID	NOT NULL	NUMBER
COMPANYNAME		VARCHAR2(15)
RATE		NUMBER(3)

InhouseProject (Anish Omprakash Pandey)

DESC IHPROJ;

Name	Null?	Type
IHPROJID	NOT NULL	NUMBER(5)
CHARGE		NUMBER(5,2)
DID	NOT NULL	NUMBER

Employee (Shailja Pandit)

DESC EMP;

Name	Null?	Type
EID	NOT NULL	NUMBER(4)
ESEX		CHAR(1)
ESALARY		NUMBER(7,2)
EADDRESS		VARCHAR2(20)
ECITY		VARCHAR2(10)
ESTATE		VARCHAR2(3)
EZIP		NUMBER(6)
EDOB		DATE
ESSN		NUMBER(9)
DID		NUMBER(3)
SUPERVISOR		VARCHAR2(9)
ELNAME		VARCHAR2(10)
EFNAME	NOT NULL	VARCHAR2(10)

Dependent (Shailja Pandit)

DESC DEPEND;

Name	Null?	Type
DPSEX		CHAR(1)
DPDOB		DATE
DPRELATION		VARCHAR2(10)
DPNAME	NOT NULL	VARCHAR2(20)
EID	NOT NULL	NUMBER

Department (Pooja Madhup)

DESC DEPT;

Name	Null?	Type
DID	NOT NULL	NUMBER(3)
DNAME	NOT NULL	VARCHAR2(16)
MSTARTDATE		DATE
DMANAGER		NUMBER(5)

EmployeeInhouse (Anish Omprakash Pandey)

DESC EMPIHPROJ;

Name	Null?	Type
HOURSPERWEEK		NUMBER(3,1)
EID	NOT NULL	NUMBER(9)
IHPROJID	NOT NULL	NUMBER(5)
EMPIHPROJID	NOT NULL	NUMBER(5)

DeptLocation (Anish Omprakash Pandey)

DESC DEPTLOC;

Name	Null?	Type
DLOCID	NOT NULL	NUMBER(3)
DLOCADDRESS		VARCHAR2(15)
DLOCPHONE		NUMBER(10)
DID	NOT NULL	NUMBER(3)

Database Insertion

Project (Pooja Madhup)

INSERT INTO PROJ VALUES ('Smart Sensors',10001, 'IHPROJ ', 'CA');
INSERT INTO PROJ VALUES ('Tracking Technology',10002,'IHPROJ','CA');
INSERT INTO PROJ VALUES ('Computer Wi-Fi',10003,'IHPROJ','CA');
INSERT INTO PROJ VALUES ('New Hardware Install',10004,'IHPROJ','CA');

INSERT INTO PROJ VALUES ('Network Update',20001,'OSPROJ','WA');
INSERT INTO PROJ VALUES ('AI Deployment',20002,'OSPROJ','FL');
INSERT INTO PROJ VALUES ('IS Software',20003,'OSPROJ','FL');

Enter SQL, PL/SQL and SQL*Plus statements.

```
INSERT INTO PROJ VALUES ('Smart Sensors',10001, 'IHPROJ ', 'CA');
```

Execute Load Script Save Script Cancel

1 row created.

OutSourceProject (Anish Omprakash Pandey)

INSERT INTO spandit3.OSPROJ VALUES (20001,'YZT Tech',250);
INSERT INTO spandit3.OSPROJ VALUES (20002,'Orion Intl',250);
INSERT INTO spandit3.OSPROJ VALUES (20003,'WASP IT',175);

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
INSERT INTO spandit3.OSPROJ VALUES (20001,'YZT Tech',250);
INSERT INTO spandit3.OSPROJ VALUES (20002,'Orion Intl',250);
INSERT INTO spandit3.OSPROJ VALUES (20003,'WASP IT',175);
```

Execute Load Script Save Script Cancel

1 row created.

1 row created.

1 row created.

InhouseProject (Anish Omprakash Pandey)

INSERT INTO spandit3.IHPROJ VALUES (10001,'45',10);
INSERT INTO spandit3.IHPROJ VALUES (10002,'50',20);
INSERT INTO spandit3.IHPROJ VALUES (10003,'55',30);
INSERT INTO spandit3.IHPROJ VALUES (10004,'60',40);

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
INSERT INTO spandit3.IHPROJ VALUES (10001,'45',10);
INSERT INTO spandit3.IHPROJ VALUES (10002,'50',20);
INSERT INTO spandit3.IHPROJ VALUES (10003,'55',30);
INSERT INTO spandit3.IHPROJ VALUES (10004,'60',40);
```

Execute Load Script Save Script Cancel

1 row created.

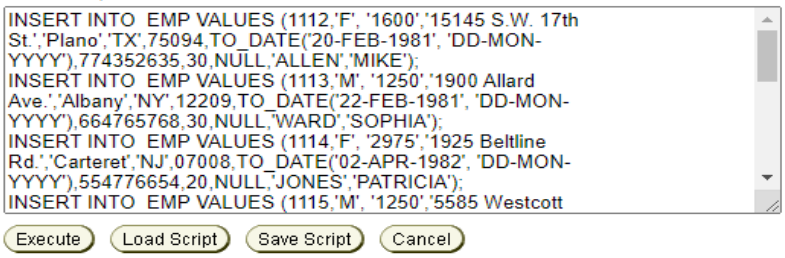
1 row created.

1 row created.

1 row created.

Employee (Shailja Pandit)

INSERT INTO Employee VALUES (1111,'JAMES SMITH','M',800,1355 S Hines Blvd,Gainesville,FL,32601-2871,17-Dec-80,775093455,20,7902);
INSERT INTO EMP VALUES (1112,'F', '1600','15145 S.W. 17th St.','Plano','TX',75094,TO_DATE('20-FEB-1981', 'DD-MON-YYYY'),774352635,30,NULL,'ALLEN','MIKE');
INSERT INTO EMP VALUES (1113,'M', '1250','1900 Allard Ave.','Albany','NY',12209,TO_DATE('22-FEB-1981', 'DD-MON-YYYY'),664765768,30,NULL,'WARD','SOPHIA');
INSERT INTO EMP VALUES (1114,'F', '2975','1925 Beltline Rd.','Carteret','NJ',07008,TO_DATE('02-APR-1982', 'DD-MON-YYYY'),554776654,20,NULL,'JONES','PATRICIA');
INSERT INTO EMP VALUES (1115,'M', '1250','5585 Westcott Ct.','Sacramento','CA',94206,TO_DATE('28-SEP-1981', 'DD-MON-YYYY'),554336475,30,NULL,'MARTIN','TRACY');
INSERT INTO EMP VALUES (1116,'F', '2850','325 Flatiron Dr.','Boulder','CO',80514,TO_DATE('01-MAY-1985', 'DD-MON-YYYY'),776443322,30,NULL,'BLAKE','MARY');
INSERT INTO EMP VALUES (1117,'M', '2450','394 Rainbow Dr.','Seattle','WA',97954,TO_DATE('09-JUN-1981', 'DD-MON-YYYY'),998221143,10,NULL,'CLARK','KENT');
INSERT INTO EMP VALUES (1118,'M', '3000','816 Peach Rd.','SantaClara','CA',96915,TO_DATE('09-DEC-1982', 'DD-MON-YYYY'),773279988,20,NULL,'SCOTT','DOUG');
INSERT INTO EMP VALUES (1119,'M', '5000','3709 First Street','Clearwater','FL',34620,TO_DATE('17-MAR-1981', 'DD-MON-YYYY'),883774588,10,NULL,'KING','STEVEN');
INSERT INTO EMP VALUES (1120,'F', '1500','2400 Rocky Point Dr.','Seminole','FL',34646,TO_DATE('08-SEP-1981', 'DD-MON-YYYY'),886532435,30,NULL,'TURNER','TINA');



1 row created.

1 row created.

1 row created.

1 row created.

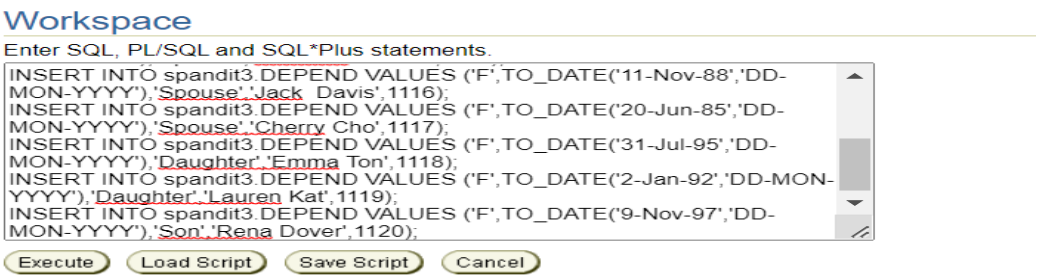
1 row created.

1 row created.

Dependent (Shailja Pandit)

INSERT INTO spandit3.DEPEND VALUES ('F',TO_DATE('12-Oct-1886','DD-MON-YYYY')'Spouse',Bob Winson',1111);
INSERT INTO spandit3.DEPEND VALUES ('M',TO_DATE('3-Aug-1999','DD-MON-YYYY'),'Son','Jay Roy',1112);

```
INSERT INTO spandit3.DEPEND VALUES ('F',TO_DATE('5-Sep-97','DD-MON-
YYYY'),'Daughter','Merry  Le',1113);
INSERT INTO spandit3.DEPEND VALUES ('F',TO_DATE('8-Jan-60','DD-MON-
YYYY'),'Mother','Micheal Joy',1114);
INSERT INTO spandit3.DEPEND VALUES ('M',TO_DATE('9-Mar-77','DD-MON-
YYYY'),'Spouse','Stefanee Jessee',1115);
INSERT INTO spandit3.DEPEND VALUES ('F',TO_DATE('11-Nov-88','DD-MON-
YYYY'),'Spouse','Jack  Davis',1116);
INSERT INTO spandit3.DEPEND VALUES ('F',TO_DATE('20-Jun-85','DD-MON-
YYYY'),'Spouse','Cherry Cho',1117);
INSERT INTO spandit3.DEPEND VALUES ('F',TO_DATE('31-Jul-95','DD-MON-
YYYY'),'Daughter','Emma Ton',1118);
INSERT INTO spandit3.DEPEND VALUES ('F',TO_DATE('2-Jan-92','DD-MON-
YYYY'),'Daughter','Lauren Kat',1119);
INSERT INTO spandit3.DEPEND VALUES ('F',TO_DATE('9-Nov-97','DD-MON-YYYY'),'Son','Rena
Dover',1120);
```




```
1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.
```

Department (Pooja Madhup)

```
INSERT INTO DEPT VALUES (10,'Administration',TO_DATE('17-NOV-2018','DD-MON-
YYYY'),1001)
INSERT INTO DEPT VALUES (20,'Marketing',TO_DATE('07-SEP-2019','DD-MON-YYYY'),1002),
INSERT INTO DEPT VALUES (30,'IT',TO_DATE('22-OCT-2018','DD-MON-YYYY'),1003)
INSERT INTO DEPT VALUES (40,'Human Resources',TO_DATE('26-APR-2015','DD-MON-
YYYY'),1004),
INSERT INTO DEPT VALUES (50,'Finance',TO_DATE('22-DEC-2017','DD-MON-YYYY'),1005),
```

```
INSERT INTO DEPT(Did, Dname,MStartdate)  VALUES
(20,'Marketing',TO_DATE('07-SEP-2019','DD-MON-YYYY'));
```

Execute Load Script Save Script Cancel

1 row created.

EmployeeInhouse (Carlos M Madrigal)

INSERT INTO spandit3.EMPIHPROJ VALUES(40,1111,10001,12);
INSERT INTO spandit3.EMPIHPROJ VALUES(20,1112,10001,22);
INSERT INTO spandit3.EMPIHPROJ VALUES(20,1115,10002,32);
INSERT INTO spandit3.EMPIHPROJ VALUES(40,1117,10003,42);
INSERT INTO spandit3.EMPIHPROJ VALUES(20,1118,10003,52);
INSERT INTO spandit3.EMPIHPROJ VALUES(30,1119,10003,62);
INSERT INTO spandit3.EMPIHPROJ VALUES(40,1120,10004,

```
INSERT INTO spandit3.EMPIHPROJ VALUES(40,1111,10001,12);
INSERT INTO spandit3.EMPIHPROJ VALUES(20,1112,10001,22);
INSERT INTO spandit3.EMPIHPROJ VALUES(20,1115,10002,32);
INSERT INTO spandit3.EMPIHPROJ VALUES(40,1117,10003,42);
INSERT INTO spandit3.EMPIHPROJ VALUES(20,1118,10003,52);
INSERT INTO spandit3.EMPIHPROJ VALUES(30,1119,10003,62);
INSERT INTO spandit3.EMPIHPROJ VALUES(40,1120,10004,72);
```

Execute Load Script Save Script Cancel

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

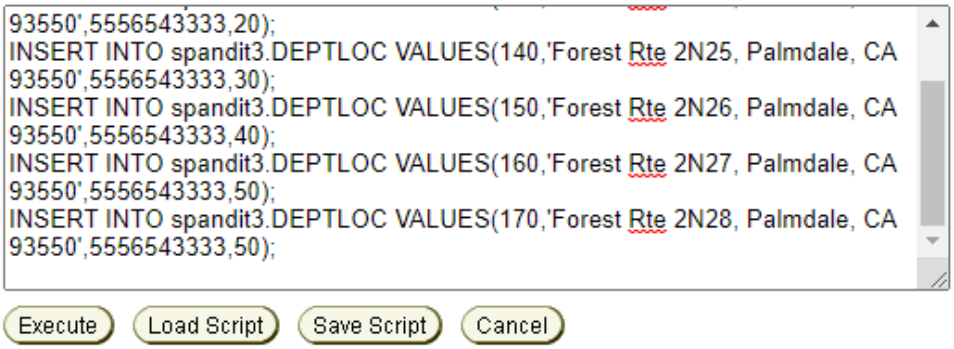
1 row created.

1 row created.

DeptLocation (Carlos M Madrigal)

INSERT INTO spandit3.DEPTLOC VALUES(120,'Corcoran Ct 882, Bernica, CA 94510',5556547857,10);

```
INSERT INTO spandit3.DEPTLOC VALUES(130,'Forest Rte 2N23, Palmdale, CA
93550',5556543333,20);
INSERT INTO spandit3.DEPTLOC VALUES(140,'Forest Rte 2N25, Palmdale, CA
93550',5556543333,30);
INSERT INTO spandit3.DEPTLOC VALUES(150,'Forest Rte 2N26, Palmdale, CA
93550',5556543333,40);
INSERT INTO spandit3.DEPTLOC VALUES(160,'Forest Rte 2N27, Palmdale, CA
93550',5556543333,50);
INSERT INTO spandit3.DEPTLOC VALUES(170,'Forest Rte 2N28, Palmdale, CA
93550',5556543333,50);
```



1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

Database Instance

PROJECT (Pooja Madhup)

SELECT * FROM proj;

PNAME	PID	PCATEGORY	PLOCATION
Smart Sensors	10001	IHPROJ	CA
Tracking Technology	10002	IHPROJ	CA
Computer Wi-Fi	10003	IHPROJ	CA
New Hardware Install	10004	IHPROJ	CA

Network Update	20001	OSPROJ	WA
AI Deployment	20002	OSPROJ	FL
IS Software	20003	OSPROJ	FL

OutSourceProject (Anish Omprakash Pandey)

SELECT * FROM OSPROJ;

OSPROJID	COMPANYNAME	RATE
20001	YZT Tech	250
20002	Orion Intl	250
20003	WASP IT	175

InhouseProject (Anish Omprakash Pandey)

select * from spandit3.IHProj

IHPROJID	CHARGE	DID
10001	45	10
10002	50	20
10003	55	30
10004	60	40

Employee (Shailja Pandit)

SELECT * FROM emp;

EID	ESEX	ESALARY	EADDRESS	ECITY	ESTATE	EZIP	EDOB	ESSN	DID	SUPERVISOR	ELNAME	EFNAME
1111	M	800	1335 S Hines Blvd	Orlando	FL	32601	24-MAR-77	775093455	20		SMITH	JAMES
1112	F	1600	15145 S.W. 17th St.	Plano	TX	75094	20-FEB-81	774352635	30		ALLEN	MIKE
1113	M	1250	1900 Allard Ave.	Albany	NY	12209	22-FEB-81	664765768	30	1118	WARD	SOPHIA
1114	F	2975	1925 Beltline Rd.	Carteret	NJ	7008	02-APR-82	554776654	20		JONES	PATRICIA
1115	M	1250	5585 Westcott Ct.	Sacramento	CA	94206	28-SEP-81	554336475	30		MARTIN	TRACY

1116	F	2850	325 Flatiron Dr.	Boulder	CO	80514	01-MAY-85	776443322	30	1115	BLAKE	MARY
1117	M	2450	394 Rainbow Dr.	Seattle	WA	97954	09-JUN-81	998221143	10		CLARK	KENT
1119	M	5000	3709 First Street	Clearwater	FL	34620	17-MAR-81	883774588	10	1120	KING	STEVEN
1120	F	1500	2400 Rocky Point Dr.	Seminole	FL	34646	08-SEP-81	886532435	30		TURNER	TINA
1118	M	3000	816 Peach Rd.	SantaClara	CA	96915	09-DEC-82	773279988	20	1115	SCOTT	DOUG

Dependent (Shailja Pandit)

Select * from DEPEND;

DPS	DPDOB	DPRELATION	DPNAME	EID
F	05-SEP-97	Daughter	Merry Le	1113
F	08-JAN-60	Mother	Micheal Joy	1114
M	09-MAR-77	Spouse	Stefanee Jessee	1115
F	11-NOV-88	Spouse	Jack Davis	1116
F	20-JUN-85	Spouse	Cherry Cho	1117
F	31-JUL-95	Daughter	Emma Ton	1118
F	02-JAN-92	Daughter	Lauren Kat	1119
F	09-NOV-97	Son	Rena Dover	1120

8 rows selected.

Department (Pooja Madhup)

SELECT* FROM dept;

DID	DNAME	MSTARTDATE	DMANAGER
10	Administration	17-NOV-18	1111
20	Marketing	07-SEP-19	1113
30	IT	22-OCT-18	1114
40	Human Resources	26-APR-15	1115
50	Finance	22-DEC-17	1120

EmployeeInhouse (Carlos M Madrigal)

SELECT * FROM spandit3.EMPIHPROJ;

HOURSPERWEEK	EID	IHPROJID	EMPIHPROJID
40	1111	10001	12
20	1112	10001	22
20	1115	10002	32
40	1117	10003	42
20	1118	10003	52
30	1119	10003	62
40	1120	10004	72

7 rows selected.

DeptLocation (Carlos M Madrigal)

SELECT * FROM spandit3.DEPTLOC;

DLOCID	DLOCADDRESS	DLOCPHONE	DID
110	Forest Rte 2N24, Palmdale, CA 93550	5556543333	10
120	Corcoran Ct 882, Bernica, CA 94510	5556547857	10
130	Forest Rte 2N23, Palmdale, CA 93550	5556543333	20
140	Forest Rte 2N25, Palmdale, CA 93550	5556543333	30
150	Forest Rte 2N26, Palmdale, CA 93550	5556543333	40
160	Forest Rte 2N27, Palmdale, CA 93550	5556543333	50
170	Forest Rte 2N28, Palmdale, CA 93550	5556543333	50

7 rows selected.

SQL DML Commands

UPDATE COMMAND (Shailja Pandit)

Updating the value of CHARGE for IHPROJID=10002 from 50 to 65

IHPROJID	CHARGE	DID
10001	40	10
10002	50	20
10003	55	30
10004	60	40

UPDATE IHPROJ SET CHARGE=65 WHERE IHPROJID = 10002;

```
UPDATE IHPROJ SET CHARGE=65 WHERE IHPROJID = 10002;
```

Execute Load Script Save Script Cancel

1 row updated.

IHPROJID	CHARGE	DID
10001	40	10
10002	65	20
10003	55	30
10004	60	40

DELETE COMMAND (Carlos M Madrigal)

Deleting the DLOCAddress for DLOCID= 180

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
SELECT * FROM spandit3.DEPTLOC
```

Execute Load Script Save Script Cancel

DLOCID	DLOCADDRESS	DLOCPHONE	DID
180	Forest Rte 2M78, Palmdale, CA 93748	7777387774	50
110	Forest Rte 2N24, Palmdale, CA 93550	5556543333	10
120	Corcoran Ct 882, Bernica, CA 94510	5556547857	10
130	Forest Rte 2N23, Palmdale, CA 93550	5556543333	20
140	Forest Rte 2N25, Palmdale, CA 93550	5556543333	30
150	Forest Rte 2N26, Palmdale, CA 93550	5556543333	40
160	Forest Rte 2N27, Palmdale, CA 93550	5556543333	50
170	Forest Rte 2N28, Palmdale, CA 93550	5556543333	50

DELETE FROM spandit3.DEPTLOC WHERE DLocid =180;

```
DELETE FROM spandit3.DEPTLOC WHERE dlocid = 180;
```

Execute Load Script Save Script Cancel

1 row deleted.

```
SELECT * FROM spandit3.DEPTLOC

--DELETE FROM spandit3.DEPTLOC WHERE dlocid = 180;
```

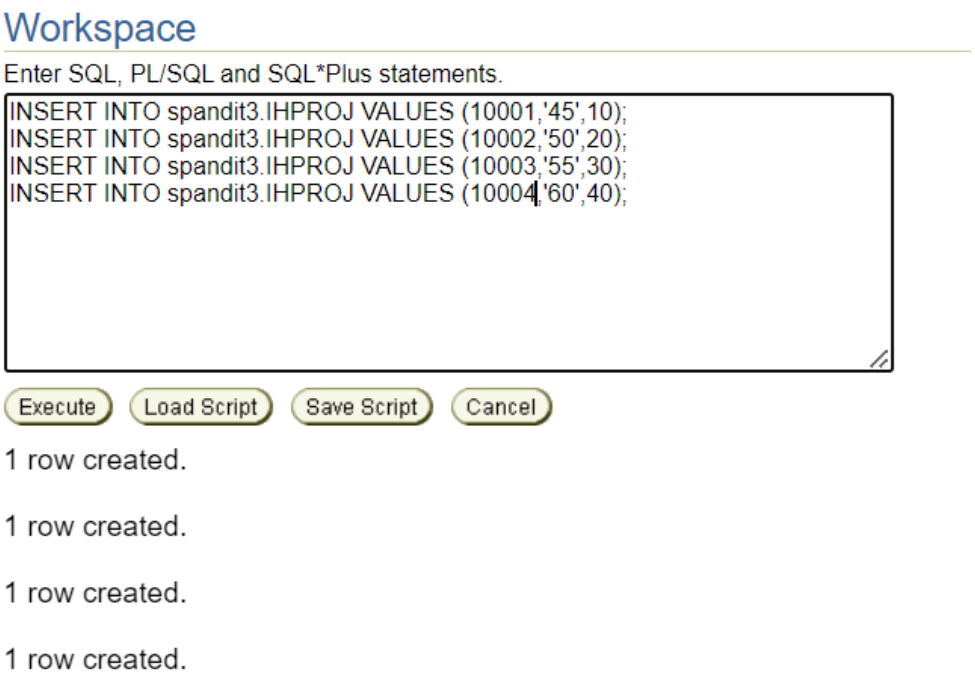
Execute | Load Script | Save Script | Cancel

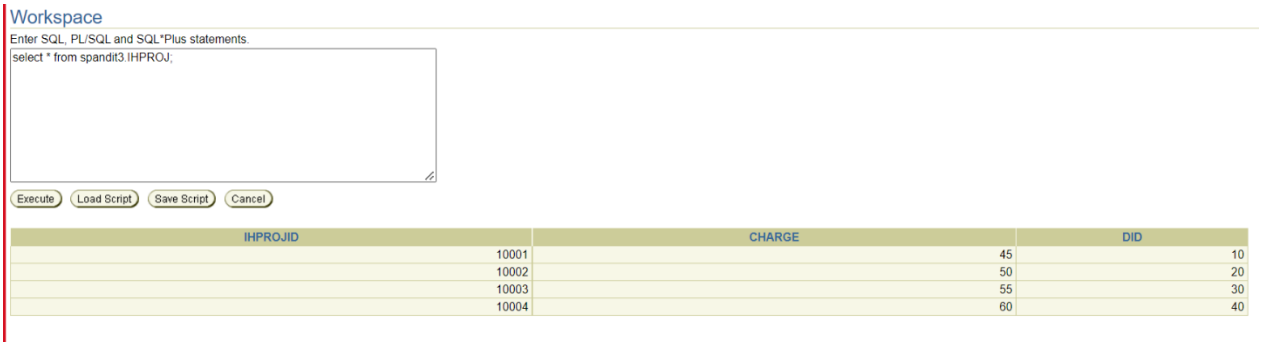
DLOCID	DLOCADDRESS	DLOCPHONE	DID
110	Forest Rte 2N24, Palmdale, CA 93550	5556543333	10
120	Corcoran Ct 882, Bernica, CA 94510	5556547857	10
130	Forest Rte 2N23, Palmdale, CA 93550	5556543333	20
140	Forest Rte 2N25, Palmdale, CA 93550	5556543333	30
150	Forest Rte 2N26, Palmdale, CA 93550	5556543333	40
160	Forest Rte 2N27, Palmdale, CA 93550	5556543333	50
170	Forest Rte 2N28, Palmdale, CA 93550	5556543333	50

INSERT COMMAND (Anish Omprakash Pandey)

Inserting new Values into IHPROJ table.

INSERT INTO IHPROJ VALUES (10001,'45',10);





IHPROJID	CHARGE	DID
10001	45	10
10002	50	20
10003	55	30
10004	60	40

SQL VIEW (Shailja Pandit)

Created a view to display Firstname,Lastname and State of Employees from EMP Table

```
CREATE OR REPLACE VIEW CA_FL_Vu
AS SELECT EFname, ELname, Estate
FROM EMP
WHERE Estate IN ('CA', 'FL') ;

SELECT * FROM CA_FL_Vu;
```

```
CREATE OR REPLACE VIEW CA_FL_Vu
AS SELECT EFname, ELname, Estate
FROM EMP
WHERE Estate IN ('CA', 'FL') ;

SELECT * FROM CA_FL_Vu;
```

Execute Load Script Save Script Cancel

View created.

EFNAME	ELNAME	ESTATE
JAMES	SMITH	FL
TRACY	MARTIN	CA
STEVEN	KING	FL
TINA	TURNER	FL
DOUG	SCOTT	CA

JOIN (Anish Omprakash Pandey)

Joined EMP and DEPT table

```
SELECT Emp.EID, EFName, Estate,DName,MStartdate
FROM spandit3.Emp, spandit3.Dept
WHERE Emp.EID = Dept.Dmanager;
```

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
SELECT Emp.EID, EFName, Estate,DName,MStartdate
FROM spandit3.Emp, spandit3.Dept
WHERE Emp.EID = Dept.Dmanager;
```

Execute Load Script Save Script Cancel

EID	EFNAME	ESTATE	DNAME	MSTARTDATE
1111	JAMES	FL	Administration	17-NOV-18
1113	SOPHIA	NY	Marketing	07-SEP-19
1114	PATRICIA	NJ	IT	22-OCT-18
1115	TRACY	CA	Human Resources	26-APR-15
1120	TINA	FL	Finance	22-DEC-17

SUBQUERY (Pooja Madhup)

Finding maximum salary for an employee from EMP table

```
SELECT Eid, ESalary, EFname,ELname
FROM spandit3.Emp
WHERE ESalary = (Select MAX (Esalary)
From spandit3.Emp
);
```

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
SELECT Eid, ESalary, EFname,ELname
FROM spandit3.Emp
WHERE ESalary = (Select MAX (Esalary)
From spandit3.Emp
);
```

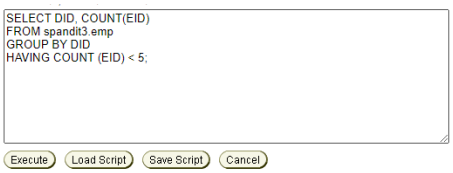
Execute Load Script Save Script Cancel

EID	ESALARY	EFNAME	ELNAME
1119	5000	STEVEN	KING

GROUP BY HAVING (Pooja Madhup)

Grouping by Department ID to have count by Employee ID

```
SELECT DID, COUNT(EID)
FROM spandit3.emp
GROUP BY DID
HAVING COUNT (EID) < 5;
```



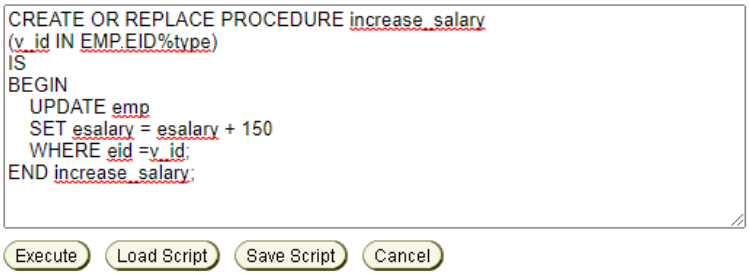
DID	COUNT(EID)
20	3
10	2

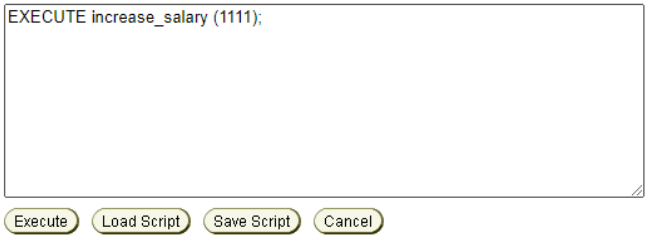
PL/SQL STATEMENT BLOCKS AND ORDBMS

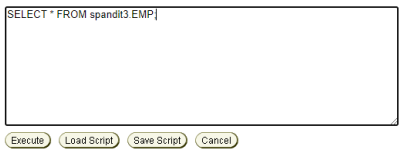
STORED PROCEDURE (Carlos M Madrigal)

Creating a Procedure to Increase the salary of a particular employee

```
CREATE OR REPLACE PROCEDURE increase_salary
(v_id IN EMP.EID%type)
IS
BEGIN
   UPDATE emp
   SET esalary = esalary + 150
   WHERE eid =v_id;
END increase_salary;
```







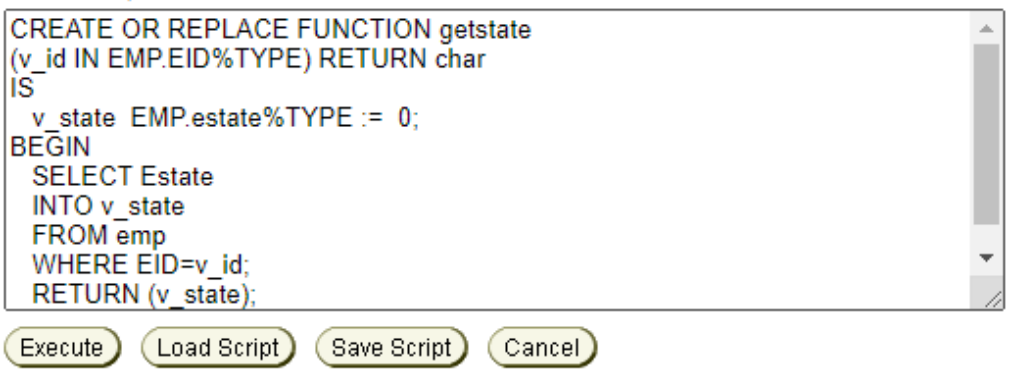
EID	ESE	ESALARY	EADDRESS	ECITY	ESTATE	EZIP	EDOB	ESSN	DID	SUPERVISOR	ELNAME	EFNAME
1111	M	800	1335 S Hines Blvd	Orlando	FL	32601	24-MAR-77	775093455	20		SMITH	JAMES

EID	ESE	ESALARY	EADDRESS	ECITY	ESTATE	EZIP	EDOB	ESSN	DID	SUPERVISOR	ELNAME	EFNAME
1111	M	950	1335 S Hines Blvd	Orlando	FL	32601	24-MAR-77	775093455	20		SMITH	JAMES

PL/SQL FUNCTIONS (Carlos M Madrigal)

Created a function to get state for a particular employee

```
CREATE OR REPLACE FUNCTION getstate
(v_id IN EMP.EID%TYPE) RETURN char
IS
  v_state  EMP.estate%TYPE :=  0;
BEGIN
  SELECT Estate
  INTO v_state
  FROM emp
  WHERE EID=v_id;
  RETURN (v_state);
END getstate;
```




```
VARIABLE g_state varchar2(50);
exec :g_state := getstate(1120);
PRINT g_state;
```

```
VARIABLE g_state varchar2(50);
exec :g_state := getstate(1120);
PRINT g_state;
```

Execute | Load Script | Save Script | Cancel

PL/SQL procedure successfully completed.

G_STATE
FL

ORDBMS SYNTAX (Carlos M Madrigal)

```
CREATE OR REPLACE TYPE EDUCATION_TY AS OBJECT
(career VARCHAR2(50),
university VARCHAR2(50),
degree VARCHAR2(25),
year NUMBER);
```

Workspace

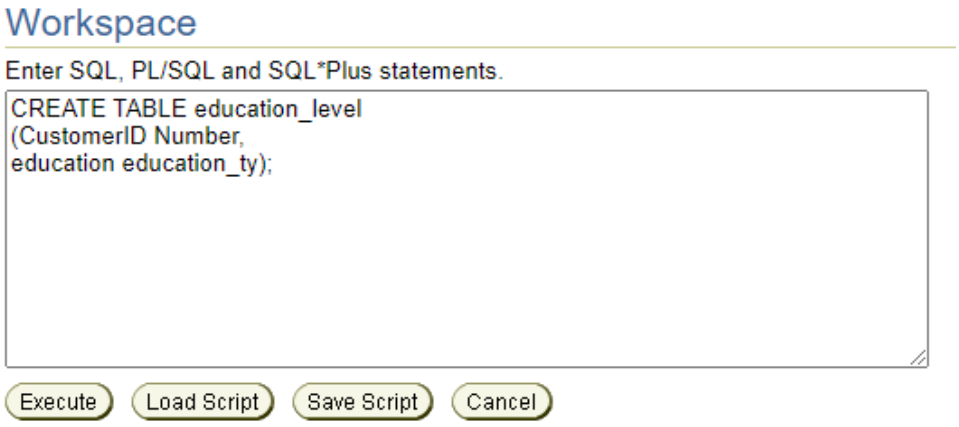
Enter SQL, PL/SQL and SQL*Plus statements.

```
CREATE OR REPLACE TYPE EDUCATION_TY AS OBJECT
(career VARCHAR2(50),
 university VARCHAR2(50),
 degree VARCHAR2(25),
 year NUMBER);
```

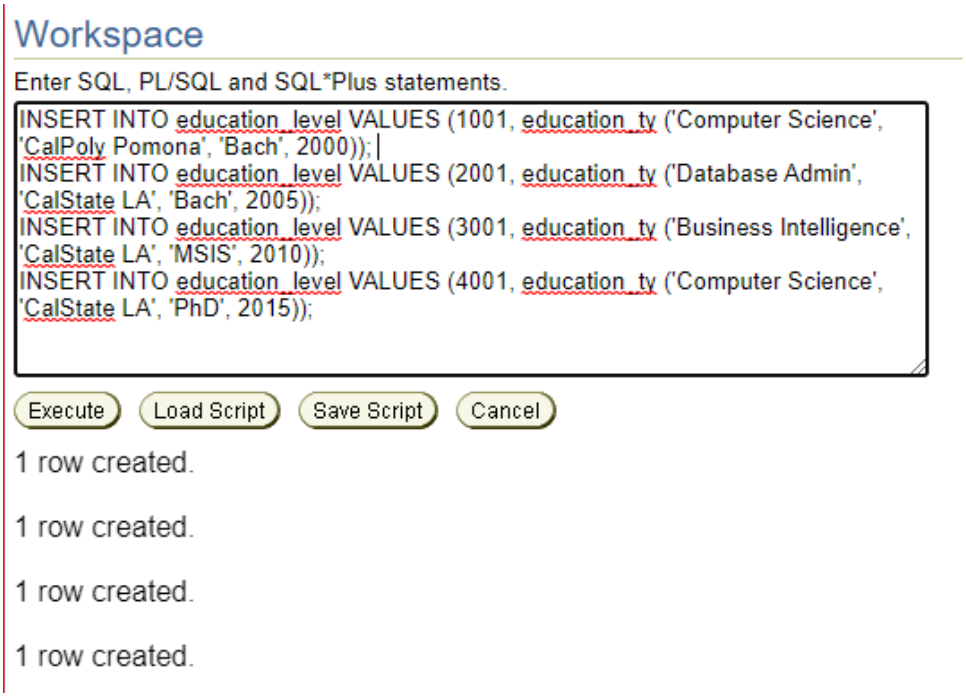
Execute | Load Script | Save Script | Cancel

Type created.

```
CREATE TABLE education_level
(CustomerID Number,
education education_ty);
```



INSERT INTO education_level VALUES (1001, education_ty ('Computer Science', 'CalPoly Pomona', 'Bach', 2000));
INSERT INTO education_level VALUES (2001, education_ty ('Database Admin', 'CalState LA', 'Bach', 2005));
INSERT INTO education_level VALUES (3001, education_ty ('Business Intelligence', 'CalState LA', 'MSIS', 2010));
INSERT INTO education_level VALUES (4001, education_ty ('Computer Science', 'CalState LA', 'PhD', 2015));



SELECT CustomerID, o.education.career FROM education_level o;

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
SELECT CustomerID,  o.education.career FROM education_level o;
```

Execute | Load Script | Save Script | Cancel

CUSTOMERID	EDUCATION.CAREER
1001	Computer Science
2001	Database Admin
3001	Business Intelligence
4001	Computer Science

STORED PROCEDURE (Anish Omprakash Pandey)

Procedure is created to change the charge for IHPROJID =10001 from 45 to 40

Before Creating procedure:

select * from spandit3.IHProj

IHPROJID	CHARGE	DID
10001	45	10
10002	50	20
10003	55	30
10004	60	40

```
CREATE OR REPLACE PROCEDURE CHARGE (NEW_ CHARGE IN NUMBER)
AS
Begin
update IHPROJ
set
CHARGE= NEW CHARGE
where IHPROJID= 10001;
END;
```

Procedure Created:

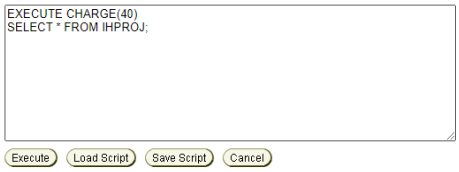
```
CREATE OR REPLACE PROCEDURE CHARGE (NEW_CHARGE IN NUMBER)
AS
Begin
update IHPROJ
set
CHARGE= NEW_CHARGE
where IHPROJID= 10001;
END;
```

Execute | Load Script | Save Script | Cancel

Procedure created.

After Procedure Created:

EXECUTE CHARGE(40)
SELECT * FROM IHPROJ;

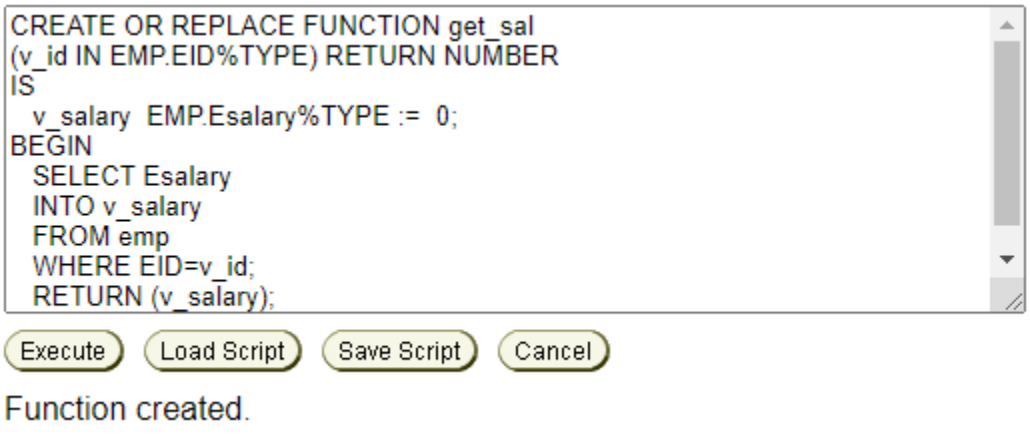


IHPROJID	CHARGE	DID
10001	40	10
10002	50	20
10003	55	30
10004	60	40

PL/SQL FUNCTIONS (Anish Omprakash Pandey)

Function is created to retrive salary of a particular employee

```
CREATE OR REPLACE FUNCTION get_sal
(v_id IN EMP.EID%TYPE) RETURN NUMBER
IS
  v_salary  EMP.Esalary%TYPE :=  0;
BEGIN
  SELECT Esalary
  INTO v_salary
  FROM emp
  WHERE EID=v_id;
  RETURN (v_salary);
END get_sal;
```




```
VARIABLE g_salary number;
exec :g_salary := get_sal(1115);
PRINT g_salary;
```

```
VARIABLE g_salary number
exec :g_salary := get_sal(1115)
PRINT g_salary
```

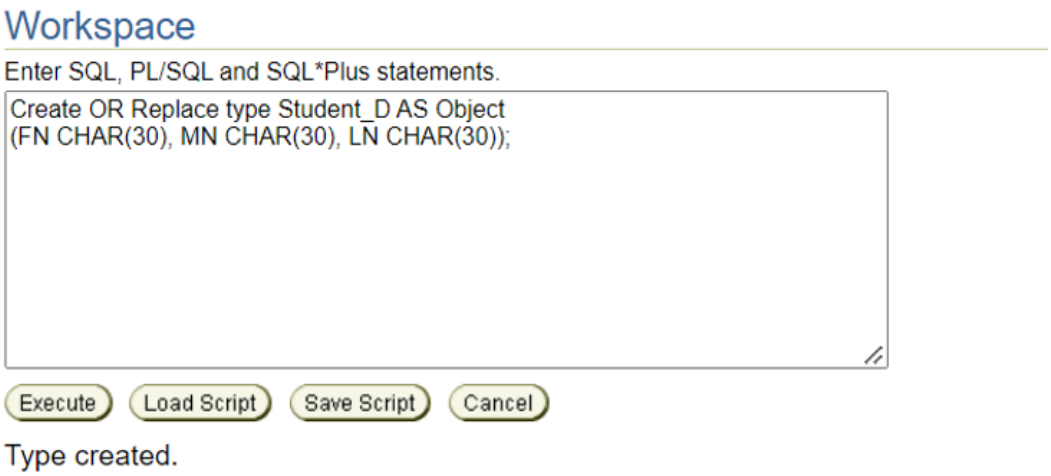
Execute | Load Script | Save Script | Cancel

PL/SQL procedure successfully completed.

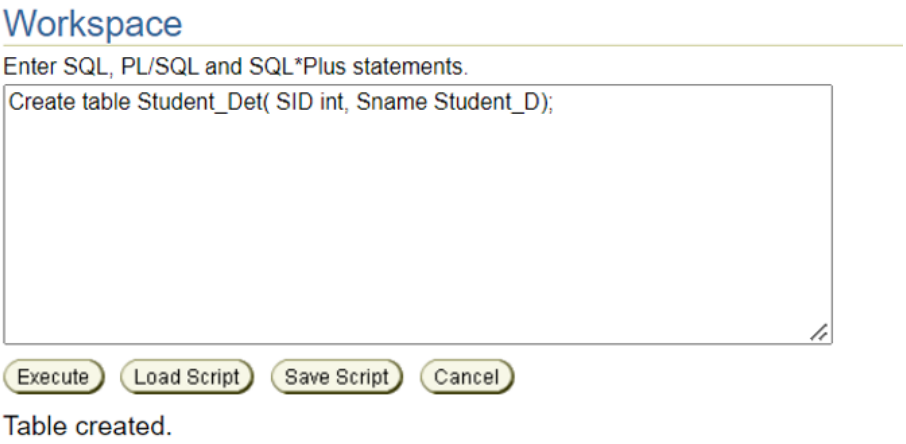
G_SALARY
1250

ORDBMS SYNTAX (Anish Omprakash Pandey)

Create OR Replace type Student_D AS Object
(FN CHAR(30), MN CHAR(30), LN CHAR(30));



Create table Student_Det(SID int, Sname Student_D);



INSERT INTO Student_Det VALUES (1, Student_D('Daniel', 'Robert' , 'Radcliff'));
INSERT INTO Student_Det VALUES (2, Student_D('Emma', 'Charlotte' , 'Watson'));
INSERT INTO Student_Det VALUES (3, Student_D('Rupert', 'Chris' , 'Grint'));

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
INSERT INTO Student_Det VALUES (1, Student_D('Daniel', 'Robert' , 'Radcliff'));
INSERT INTO Student_Det VALUES (2, Student_D('Emma', 'Charlotte' ,
'Watson'));
INSERT INTO Student_Det VALUES (3, Student_D('Rupert', 'Chris' , 'Grint'));
```

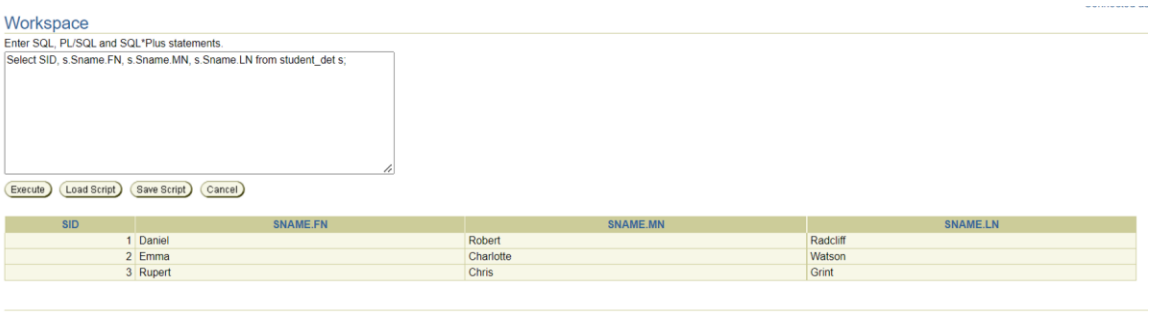
Execute | Load Script | Save Script | Cancel

1 row created.

1 row created.

1 row created.

Select SID, s.Sname.FN, s.Sname.MN, s.Sname.LN from student_det s;



SID	SNAME.FN	SNAME.MN	SNAME.LN
1	Daniel	Robert	Radcliff
2	Emma	Charlotte	Watson
3	Rupert	Chris	Grint

STORED PROCEDURE (Shailja Pandit)

Procedure is created to to update Relation for a particular employee

Before Creating Procedure:

DPS	DPDOB	DPRELATION	DPNAME	EID
F	05-SEP-97	Daughter	Merry Le	1113
F	08-JAN-60	Mother	Micheal Joy	1114
M	09-MAR-77	Spouse	Stefanee Jessee	1115
F	11-NOV-88	Spouse	Jack Davis	1116
F	20-JUN-85	Spouse	Cherry Cho	1117
F	31-JUL-95	Daughter	Emma Ton	1118
F	02-JAN-92	Daughter	Lauren Kat	1119
F	09-NOV-97	Son	Rena Dover	1120

Procedure Created:

```
CREATE OR REPLACE PROCEDURE UPDATE_RELATION (NEW_RELATION IN char)
IS
Begin
update DEPEND
set
DPRELATION= NEW_RELATION
```

where EID= 1120;
END;

```
CREATE OR REPLACE PROCEDURE UPDATE_RELATION (NEW_RELATION IN
char)
IS
Begin
update DEPEND
set
DPRELATION= NEW_RELATION
where EID= 1120;
END;
```

Execute | Load Script | Save Script | Cancel

Procedure created.

After Procedure Created:

EXECUTE UPDATE_RELATION('Brother')
SELECT * FROM DEPEND;

PL/SQL procedure successfully completed.

DPS	DPDOB	DPRELATION	DPNAME	EID
F	05-SEP-97	Daughter	Merry Le	1113
F	08-JAN-60	Mother	Micheal Joy	1114
M	09-MAR-77	Spouse	Stefanee Jessee	1115
F	11-NOV-88	Spouse	Jack Davis	1116
F	20-JUN-85	Spouse	Cherry Cho	1117
F	31-JUL-95	Daughter	Emma Ton	1118
F	02-JAN-92	Daughter	Lauren Kat	1119
F	09-NOV-97	Brother	Rena Dover	1120

8 rows selected.

PL/SQL FUNCTIONS (Shailja Pandit)

Function is created to retrive the Name of Dependent for a particular employee

Function Created:

```
CREATE OR REPLACE FUNCTION getname (v_id IN EMP.EID%TYPE) RETURN char
IS
  v_name  DEPEND.DPNAME%TYPE :=0;
BEGIN
  SELECT DPNAME INTO v_name
  FROM DEPEND
  WHERE EID=v_id;
  RETURN (v_name);
END getname;
```

```
CREATE OR REPLACE FUNCTION getname (v_id IN EMP.EID%TYPE) RETURN
char
IS
  v_name  DEPEND.DPNAME%TYPE :=0;
BEGIN
  SELECT DPNAME INTO v_name
  FROM DEPEND
  WHERE EID=v_id;
  RETURN (v_name);
END getname;
```

Execute | Load Script | Save Script | Cancel

Function created.

After Creating Function:

```
VARIABLE d_name varchar2(50);
exec :d_name := getname(1115);
PRINT d_name;
```

```
VARIABLE d_name varchar2(50);
exec :d_name := getname(1115);
PRINT d_name;
```

Execute | Load Script | Save Script | Cancel

PL/SQL procedure successfully completed.

D_NAME
Stefanee Jessee

ORDBMS SYNTAX (Shailja Pandit)

```
CREATE OR REPLACE TYPE ADDRESS_TY AS OBJECT
        (street VARCHAR2(50),
         city VARCHAR2(25),
         state CHAR(2),
         zip  NUMBER);
```

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
CREATE OR REPLACE TYPE ADDRESS_TY AS OBJECT
(street VARCHAR2(50),
 city VARCHAR2(25),
 state CHAR(2),
 zip  NUMBER);
```

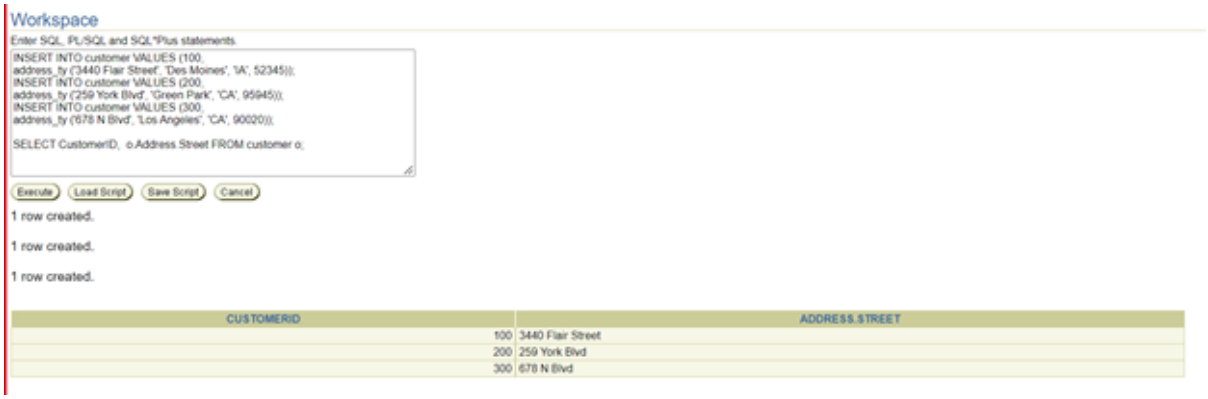
Execute | Load Script | Save Script | Cancel

Type created.

```
CREATE TABLE customer
(CustomerID Number,
Address address_ty);

INSERT INTO customer VALUES (100,
address_ty ('3440 Flair Street', 'Des Moines', 'IA', 52345));
INSERT INTO customer VALUES (200,
address_ty ('259 York Blvd', 'Green Park', 'CA', 95945));
INSERT INTO customer VALUES (300,
address_ty ('678 N Blvd', 'Los Angeles', 'CA', 90020));

SELECT CustomerID,  o.Address.Street FROM customer o;
```



STORED PROCEDURE (Pooja Madhup)

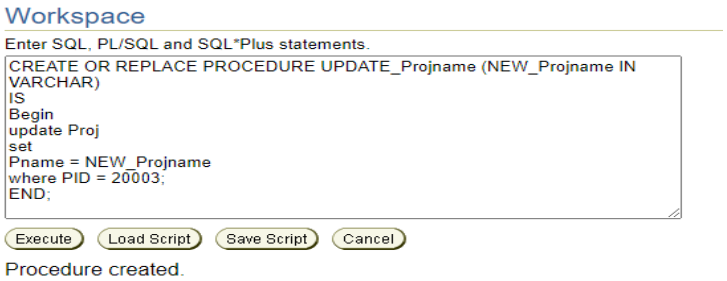
Procedure is created to change the Pname of PID = '20003' from IS SOFTWARE to IT SOFTWARE

Before Procedure Created:

SELECT * FROM proj;

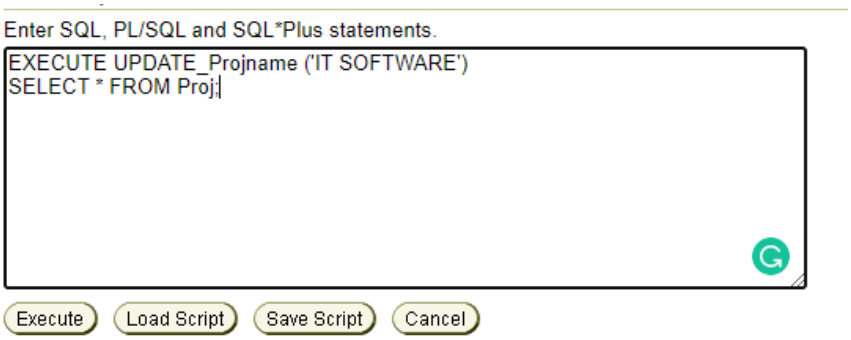
PNAME	PID	PCATEGORY	PLOCATION
Smart Sensors	10001	IHPROJ	CA
Tracking Technology	10002	IHPROJ	CA
Computer Wi-Fi	10003	IHPROJ	CA
New Hardware Install	10004	IHPROJ	CA
Network Update	20001	OSPROJ	WA
AI Deployment	20002	OSPROJ	FL
IS Software	20003	OSPROJ	FL

```
CREATE OR REPLACE PROCEDURE UPDATE_Projname (NEW_Projname IN VARCHAR)
IS
Begin
update Proj
set
Pname = NEW_Projname
where PID = 20003;
END;
```



After Procedure Created:

After update(PName from 'IS Software' to 'IT Software')
EXECUTE UPDATE_Projname ('IT SOFTWARE')
SELECT * FROM Proj;



PNAME	PID	PCATEGORY	PLOCATION
Tracking Technology	10002	IHPROJ	CA
Computer Wi-Fi	10003	IHPROJ	CA
New Hardware Install	10004	IHPROJ	CA
Network Update	20001	OSPROJ	WA
AI Deployment	20002	OSPROJ	FL
IT SOFTWARE	20003	OSPROJ	FL
Smart Sensors	10001	IHPROJ	CA

7 rows selected.

PL/SQL FUNCTIONS (Pooja Madhup)

Function is created to retrieve the name of a Project for a particular Project ID

Function created

CREATE OR REPLACE FUNCTION Func_name (Proj_ID in number) RETURN VARCHAR
IS
Projname varchar(50);
BEGIN
SELECT Pname into Projname
FROM Proj
WHERE PID = Proj_ID;
RETURN Projname;
END Func_name;

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
CREATE OR REPLACE FUNCTION Func_name (Proj_ID in number) RETURN
VARCHAR
IS
Projname varchar(50);
BEGIN
SELECT Pname into Projname
FROM Proj
WHERE PID = Proj_ID;
RETURN Projname;
END Func_name;
```

Execute | Load Script | Save Script | Cancel

Function created.

After creating function:

Variable Projname VARCHAR2(50);
exec:Projname := Func_name(10001);
print Projname;

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
Variable Projname VARCHAR2(50);
exec:Projname := Func_name(10001);
print Projname;
```

Execute | Load Script | Save Script | Cancel

PL/SQL procedure successfully completed.

PROJNAME
Smart Sensors

ORDBMS SYNTAX (Pooja Madhup)

Create OR Replace type Class_D AS Object
(Serial_No INT, Name VARCHAR2(30), CIN NUMBER(9));

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
Create OR Replace type Class_D AS Object
(Serial_No INT, Name VARCHAR2(30), CIN NUMBER(9));
```

Execute | Load Script | Save Script | Cancel

Type created.

CREATE TABLE Class_Det
(Serial_No INT, Name VARCHAR2(30), CIN NUMBER(9));

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
CREATE TABLE Class_Det
(Serial_No INT, Name VARCHAR2(30), CIN NUMBER(9));
```

Execute Load Script Save Script Cancel

Table created.

```
INSERT INTO Class_Det VALUES (1, 'Robin', '411977573');
INSERT INTO Class_Det VALUES (2, 'Bob', '411977584');
INSERT INTO Class_Det VALUES (3, 'Erik', '411977595');
```

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
INSERT INTO Class_Det VALUES (1, 'Robin', '411977573');
INSERT INTO Class_Det VALUES (2, 'Bob', '411977584');
INSERT INTO Class_Det VALUES (3, 'Erik', '411977595');
```

Execute Load Script Save Script Cancel

1 row created.

1 row created.

1 row created.

```
SELECT c.Serial_No, c.Name, c.CIN FROM Class_Det c;
```

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
SELECT c.Serial_No, c.Name, c.CIN FROM Class_Det c;
```

Execute Load Script Save Script Cancel

SERIAL_NO	NAME	CIN
1	Robin	411977573
2	Bob	411977584
3	Erik	411977595

UML DIAGRAM

The company Database is illustrated in the UML class diagram below. In the UML diagram below, Each of the classes is displayed as a rectangle that includes a top section that displays the class name, the middle

section which displays the attributes of the class and the last section displays the methods that operate on the data in the object. The associations between the classes are denoted with multiplicity and Inheritance is shown with the help of an empty diagram. In this case, the ‘InHouse Project’ and the ‘OutSource Project’ inherit the properties of the ‘Project’ Class.