

## Group Project – 1

### Databases and Data warehousing-5430

#### Introduction: (Chandan Basavaraju)

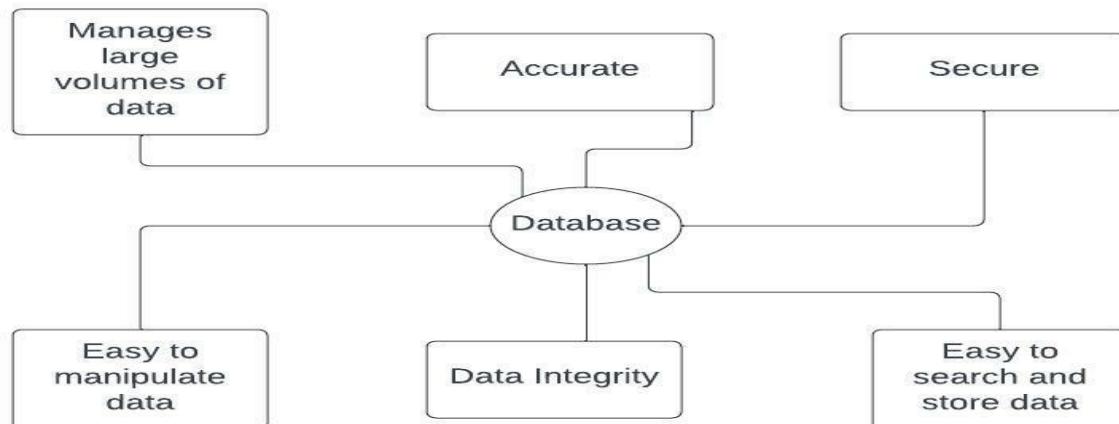
##### Database Scenario

The COMPANY database keeps track of a company's employees, departments, and projects

1. The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when the manager began managing the department. A department may have several locations.
2. A department controls several projects. Each project has a unique name, a unique number. One project must be either an in-house project or an out-source project. We keep track of the number of hours per week that an employee works on each in-house project.
3. We store each employee's name, social security number, address, salary, sex, and birthday. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department. We also keep track of the direct supervisor of each employee.
4. We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's name, sex, birthday, and relationship to the employee.

Manages	1:1	Works For	1: N
Controls	1: N	Supervision	1: N
Worked on	M: N	Dependents Of	1: N

Now we must develop a sustainable database for the company. A database must be accurately built, for collection of data, which are stored in various forms, files, or information. Database needs to couple both storage and accessibility of information. They are vital for company or organization, where it stores and manages all the relevant details about a company which is a combination of employees, departments, projects, inhouse or outhouse project, employees dependent.



**Needs – Requirements – Function – and purpose of development.** (Chandan Basavaraju)

For expressing database queries and alterations, a database system contains both a data definition language and a data manipulation language. One of the most important advantages of using DBMSs is that they provide you complete control over your data and the programs that use it. Databases are organized collections of data that can be easily accessed, maintained, and updated. Database systems are crucial to your organization because they communicate information about your sales transactions, product inventory, customer profiles, and marketing initiatives.

**Need:** (Chandan Basavaraju)

Spreadsheet are fantastic for crunching numbers. However, if you have a lot of data, such as information about customers, employees, or goods, you might benefit from a more effective data management tool. Here are some reasons why you should consider using databases instead of spreadsheets to help your organization thrive.

When your company expands and you hire employees to help you run it, keeping track of the expanding volume of data can be difficult. To improve your chances of success, good database systems can help you handle all of your business-critical data centrally, safely, and securely - whenever you need it.

Managing employee records with an HR database can save you time and money. It can automate regular HR activities and speed up the processing of data such as employee hours, leave, benefits, payroll, and so on. This can free up time for you to focus on building your company.

If customers are the lifeblood of your company, a good customer relationship management (CRM) database should be at the center of any expansion strategy. CRM databases with all of the bells and whistles are usually capable of storing and processing anything from customer contact information, interaction history, and accounts to new prospects, leads, and business opportunities. Some CRM systems can even assist you with marketing efforts like as email newsletters, by allowing you to launch and track them.

Managing your inventory well can occasionally feel like a balancing act. It's easy to have too much - sitting on a shelf risking wastage, or too little - disappointing your customers and damaging your reputation. If you're tracking your inventory manually, it's also easy to miscount, make data entry errors or misplace spreadsheets and notes. By using an inventory tracking database, especially alongside electronic data interchange and barcode scanning, you can avoid those risks and minimise lost sales while maximising your opportunities for growth.

Most business databases have some form of reporting capabilities - from analysing input data and productivity tracking, to anticipating future trends and customers' needs. If you're planning a strategy for growth, a robust database system can be your business' most valuable resource.

**Types of Databases:**

1. Hierarchical databases.
2. Network databases.
3. Object-oriented databases.
4. Relational databases.
5. NoSQL databases.

**Requirements:** (Chandan Basavaraju)

Asking the proper questions is the best way to go about selecting a database management solution. Before choosing a database program, every institution should ask the following questions.

What type of data do we want to keep, and why do we want to keep it?

What is the significance of the relationship between each data set we are attempting to store?

Is the data kept in a dynamic or static format?

What information do we wish to arrange and how will we get access to it?

How much data do we secure and how do we do it?

Who will have access to this information?

**Functions:** (Chandan Basavaraju)

Management of data dictionaries, data storage, data transformation, and data presentation.

Data Integrity Management, Security Management, Multi-User Access Control, Backup and Recovery Management.

Database Communication Interface, Database Access Languages and Application Interface.

The features and functionalities of a product are defined by functional requirements.

Nonfunctional requirements are the characteristics of a system that are not functional. They're sometimes referred to as quality characteristics.

**Business Rules**

One employee must work for one and only one department.

One department must have 4 or more employees.

One employee may manage one and only one department.

One department must be managed by one and only employee.

One department may control one or more in-house projects.

One in-house project must be controlled by one and only one department.

One employee must work on one or many projects.

One project must have one or many employees to work on.

One employee may have one or more dependents.

One dependent must belong to one and only one employee.

One employee may supervise many other employees.

One employee may be supervised by another employee.

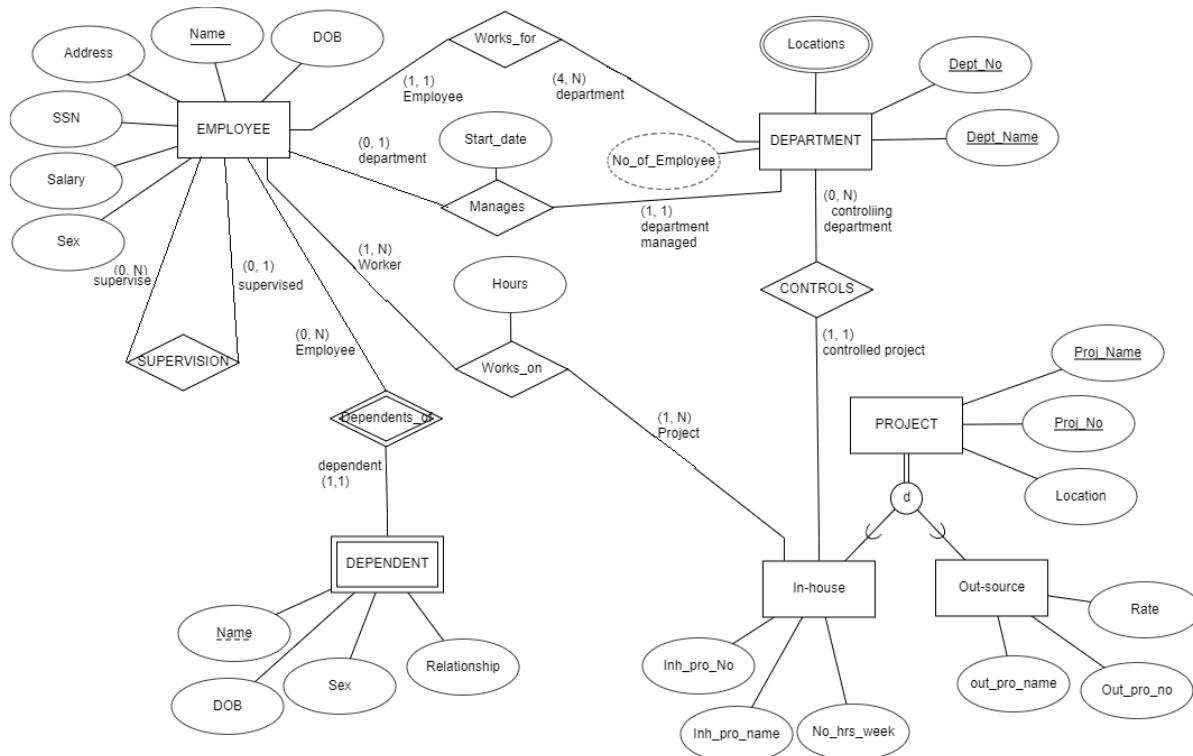
One project must be either an in-house project or out-source project.

**I. Given the above database scenario and business rules, do the following things:**

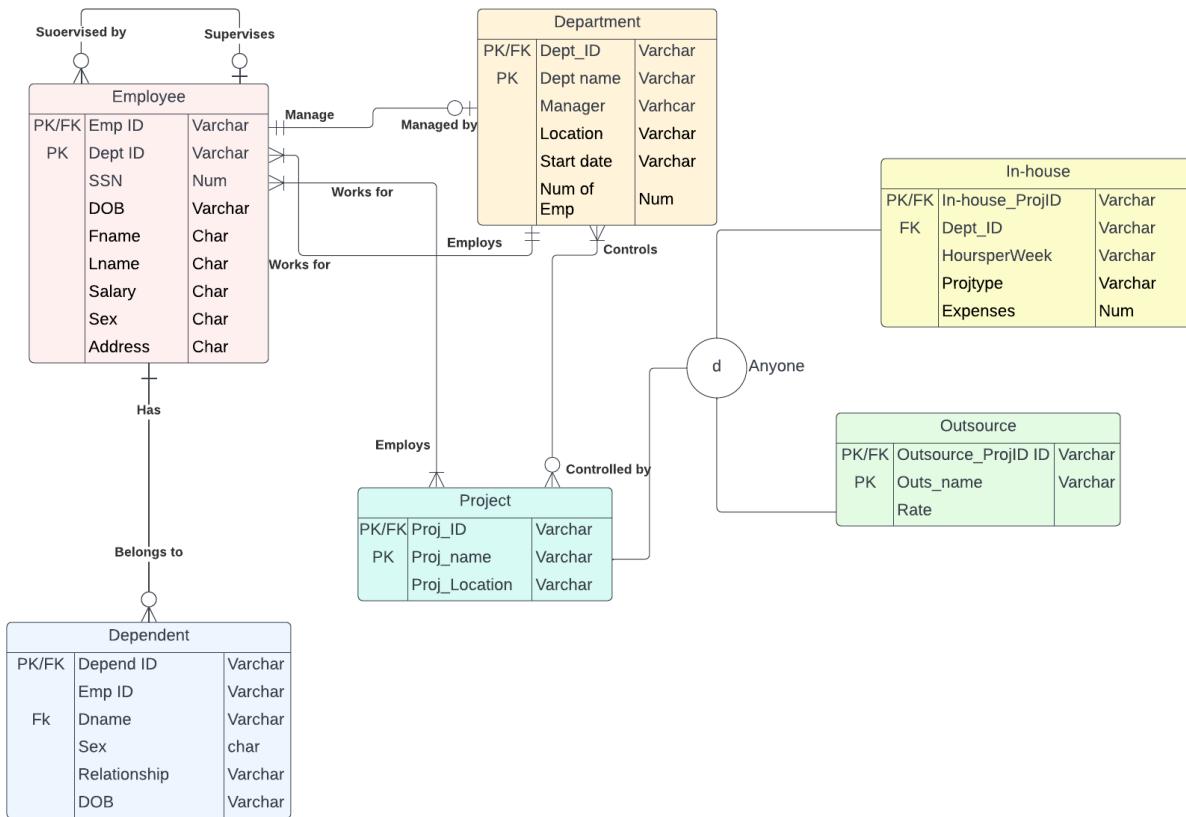
1. Identify entity types and relationship types using relationship matrix (**RUTA ANTALIYA**)

	Employee	Department	Project	In-house	Out-house	Dependent
employee	May supervise	Must be managed	Must have	==	==	May have
department	Must work	==	==	Must be controlled	==	==
project	Must work	==	==	==	==	==
in-house	==	may control	Must be	==	==	==
out-source	==	==	Must be	==	==	==
dependent	Must be belong	==	==	==	==	==

2. Draw an ER diagram with 1) entity types, 2) relationship types, 3) keys, 4) attributes, and 5) cardinality constraint using ERD Plus <https://erdplus.com/create-account> (**RUTA ANTALIYA**)



ER -Diagram – ( Chandan Basavaraju )



Group Number – 1. Basavaraju, Chandan. Chantaphakul, Kato. Antaliya, Ruta. Akkineni, Snehit  
 5430- Dr. Ming Wang

3. Map either one of the above ER diagrams to a relational database indicating the relation's name, attribute name and primary key for each relation. Add appropriate additional field according to scenario.

(Kato Chantaphakul)

Table name: Employee

Emp_ID (PK)	Dept_ID. (FK)	SSN	DOB	FName
LName	Salary	Sex	Address	

Table name: Dependent

Depend_ID (PK)	Emp_ID(FK)	D_Name	Sex	Relationship
DOB				

Table name: Department

Dept_ID. (PK)	Dept_Name	Manager	Location	Start_Date
NumOfEmp				

Table name: Location

Loc_ID (PK)`	Branch	Address	City	State
ZipCode				

Table name: Project

Proj_ID (PK)	Proj_Name	Proj_Location		

Table name: In-House

InHouse_ProjID (PK)	Dept_ID (FK)	HoursPerWeek	ProjType	Expenses

Table name: Out-Source

OutSource_ProjID (PK)	OutS_Name	Rate

Table name: Works-On

Work_ID (PK)	Emp-ID (FK)	Proj_ID (FK)		

4. Establish join paths for the above relational database using referential integrity by drawing arrow lines between the above tables. Indicate all the foreign keys (FK). (Kato Chantaphakul)

Table name: Employee

Emp_ID (PK)	Dept_ID (FK)	SSN	FName	LName
Address	Salary	Sex	DOB	

Table name: Dependent

Depend_ID (PK)	Emp_ID(FK)	FName	LName	Relation
DOB	Gender			

Table name: Department

Dept_ID. (PK)	Dept_Name	Manager	Location	Start_Date
NumOfEmp				

Table name: Location

Loc_ID (PK)	Branch	Address	City	State
ZipCode				

Table name: Project

Proj_ID (PK)	Proj_Name	Proj_Location		

Table name: In-House

InHouse_ProjID (PK)	Dept_ID (FK)	HoursPerWeek	ProjType	Expenses

Table name: Out-Source

OutSource_ProjID (PK)	OutS_Name	Rate	

Table name: Works-On

Work_ID (PK)	Emp-ID (FK)	Proj_ID (FK)		

5. Do functional dependency analysis (**Snehith**)

### **Functional Dependency**

#### **Full Dependency**

$\text{EmpID} \rightarrow \text{SSN, EmpName, EmpAdd, Salary, Sex}$

$\text{EmpID, DepLName, DepFName} \rightarrow \text{Relation, DOB, Gender}$

$\text{DepID} \rightarrow \text{DepName, Manager, Location, Start\_date, NumofEmp}$

$\text{LocID} \rightarrow \text{Branch}$

$\text{Project.ProjID} \rightarrow \text{ProjName, ProjLocation}$

$\text{InHouse.ProjID} \rightarrow \text{HoursPerWeek, ProjType, Expenses}$

$\text{OutSource.ProjID} \rightarrow \text{Rate}$

### **Transitive Dependency**

$\text{Manager} \rightarrow \text{Start\_date}$

6. Show the relational schema after normalization (**Ruta Antaliya**)

ANS: - Department, Employee table are not in the 1NF Form.

Decompose the relation schema to perform the normalization

- Employee(Emp\_ID, Dept\_ID, FName, LName, Sex, DOB, Address)
- EMP In-House Project(InHouse\_ProjID, Dept\_ID, HoursPerWeek , ProjType )
- DEPARTMENT( Dept\_ID., Dept\_Name, Manager, Location, Start\_Date, NumOfEmp)
  - Department\_Locatins(Dept\_ID., Loc\_ID)
  - Location(Loc\_ID, Address, City, State, ZipCode, Bramch)
- PROJECT( Proj\_ID, Proj\_Name, Proj\_Location)
  - In-House Project(InHouse\_ProjID, Dept\_ID, HoursPerWeek , ProjType )
  - Out-House Project(OutSource\_ProjID, OutS\_Name , Rate)
- DEPENDENT(Depend\_ID, Emp\_ID, FName, LName, Relation, DOB, Sex)

Two relations have been created

- EMP In-House Project
- Department\_Locatins

Now functional dependency is maintained.

7. Indicate what normalization form each table is in. (**Snehith**)

### Table Normalization

	1NF	2NF	3NF
Employee	X	X	
Department	X	X	
Department Location	X	X	X
Dependent	X	X	X
Project	X	X	X
In-House	X	X	X
Out-Source	X	X	X
Employee Project	X	X	X



# Company database

5430- Group 1

## **TABLE OF CONTENTS:**

<b>Introduction</b>	<b>3</b>
<b>Database Creation Script</b>	<b>4</b>
<b>Describe Table</b>	<b>9</b>
<b>Performing Insertion, Update, Delete, Create view</b>	<b>15</b>
<b>Select (where, group by, having) Query</b>	<b>18</b>
<b>PL/SQL &amp; ORDBMS – (Procedures, Functions, Type, Object)</b>	<b>23</b>
<b>UML DIAGRAM</b>	<b>36</b>

# **Introduction** (**Chandan Basavaraju**)

Our database design model is developed for a company to enrich their working capabilities by keeping track of employees, department, and models, the business is divided into departments. Each department has a distinct name, a distinct number, and a distinct employee in charge. We keep records of when the manager took over management of the department, there could be multiple locations for a department. A department oversees several initiatives. Each project has its own name and number. Either an in-house or an out-source project must be completed. We keep track of how many hours each employee works on each in-house project per week. We keep track of each employee's name, SSN, address, salary, gender, and birthday. Although an individual is allocated to one department, he or she may work on multiple projects that are not all managed by the same department. We also keep track of each employee's direct supervisor. We need to maintain track of each employee's dependents for insurance purposes. Name, sex, birthday, and relationship of each dependent to the employee are all kept on file. Large volumes of data are used by businesses. Database management system (DBMS) is a piece of software that allows you to arrange information in a database is the usual abbreviation for database management system, and you'll often see it instead of the full word. Database management system's goal is to store and transform data into information that can be used to make choices. A database management system (DBMS) is made up of three components: The physical database consists of a collection of files containing data. The database engine is the software that allows you to access and edit the database's contents. The database schema is a description of the data in the database's logical structure.

## **Functionalities:**

Include management of the data dictionary management of data storage transformation and display of data management of security backup and recovery management with multi-user access control management of data integrity application programming interfaces and database access languages communication interfaces for databases.

## **Users:**

End Users / Naive / Parametric Database Administrator (DBA) Sophisticated Users Data Base Designers System Analysts Casual Users / Temporary Users Application Program.

## **Team leaders:**

Database team oversees the company's enterprise database systems' strategic planning and routine maintenance.

## **Developers:**

Database programmers, often known as database developers, oversee designing and implementing computer databases. They select the most appropriate database management system for a given client, as well as test database programs for efficiency and performance, and troubleshoot and resolve issues.

# Database Creation Script

```
CREATE TABLE EMPLOYEE      (Chandan Basavaraju)
(
    Name VARCHAR(100) NOT NULL,
    SSN INT NOT NULL,
    Address VARCHAR(100) NOT NULL,
    Salary INT NOT NULL,
    Sex CHAR(20) NOT NULL,
    DOB CHAR(30) NOT NULL,
    EmpID INT NOT NULL,
    Dept_No INT NOT NULL,
    SUPERVISION_EmpID INT,
    CONSTRAINT Emp_Gender check(Sex in('Male','Female','M','F')),
    CONSTRAINT Emp_salary check(salary>0),
    CONSTRAINT EMPLOYEE_PK PRIMARY KEY (EmpID),
    CONSTRAINT EMPLOYEE_UK UNIQUE (Ssn),
    CONSTRAINT EMPLOYEE_FK FOREIGN KEY (SUPERVISION_EmpID) REFERENCES
EMPLOYEE(EmpID)
);
INSERT INTO EMPLOYEE VALUES ('Ruta','786987567','355 S Hines','2000','Female','27-07-
87','1','1','1');
INSERT INTO EMPLOYEE VALUES ('Chandan','986987567','5145
S.W.17th','1000','Male','07-07-81','2','2','2');
INSERT INTO EMPLOYEE VALUES ('Kato','346987567','355 Hines Blvd','1500','Male','27-
09-87','3','3','3');
INSERT INTO EMPLOYEE VALUES ('Snehit','906987567','355 Hines Blvd','2500','Male','15-
08-85','4','4','4');
INSERT INTO EMPLOYEE VALUES ('Kanoh','126987567','312 Kiowai St','2000','Female','27-
07-80','5','5','5');
INSERT INTO EMPLOYEE VALUES ('Mountain','126987567','5689 College
Ave','1000','Male','07-07-81','6','6','6');
INSERT INTO EMPLOYEE VALUES ('Kalo','346990567','45 Capitol Ave','1500','Male','27-
09-87','7','7','7');
INSERT INTO EMPLOYEE VALUES ('hit','906567567','424 Missouri Ave','2500','Male','15-
08-85','8','8','8');
INSERT INTO EMPLOYEE VALUES ('tara','709987567','4 Rocky Point
Dr','2000','Female','27-07-80','9','9','9');
INSERT INTO EMPLOYEE VALUES ('nadan','986987598','45 S 17th St','1000','Male','07-07-
81','10','10','10');
```

```

CREATE TABLE DEPARTMENT ((Chandan Basavaraju))
(
    Dept_No INT NOT NULL,
    Dept_Name VARCHAR(20) NOT NULL,
    EmpID INT NOT NULL,
    CONSTRAINT DEPARTMENT_UK UNIQUE (Dept_Name),
    CONSTRAINT DEPARTMENT_PK PRIMARY KEY (Dept_No),
    CONSTRAINT DEPARTMENT_FK FOREIGN KEY (EmpID) REFERENCES
EMPLOYEE(EmpID)
);
INSERT INTO DEPARTMENT VALUES ('1','IT Division','1');
INSERT INTO DEPARTMENT VALUES ('2','Account Division','2');
INSERT INTO DEPARTMENT VALUES ('3','Helthcare Division','3');
INSERT INTO DEPARTMENT VALUES ('4','HR Division','4');
INSERT INTO DEPARTMENT VALUES ('5','Production Division','5');
INSERT INTO DEPARTMENT VALUES ('6','Engineering Division','6');
INSERT INTO DEPARTMENT VALUES ('7','Networking Division','7');
INSERT INTO DEPARTMENT VALUES ('8','Medical Division','8');
INSERT INTO DEPARTMENT VALUES ('9','Biology Division','9');
INSERT INTO DEPARTMENT VALUES ('10','Architecture Division','10');

```

```

CREATE TABLE DEPENDENT ((Kato Chantaphakul))
(
    Name CHAR(20) NOT NULL,
    Sex CHAR(20) NOT NULL,
    DOB VARCHAR(50) NOT NULL,
    Relationship CHAR(50) NOT NULL,
    depdID INT NOT NULL,
    EmpID INT NOT NULL,
    CONSTRAINT DEPENDENT_PK PRIMARY KEY (depdID),
    CONSTRAINT DEPEMPID_FK FOREIGN KEY (EmpID) REFERENCES
EMPLOYEE(EmpID)
);
INSERT INTO DEPENDENT VALUES ('Mountain','Male','07-07-81','Employee','6','6');
INSERT INTO DEPENDENT VALUES ('Kalo','Male','27-09-87','Customer','7','7');
INSERT INTO DEPENDENT VALUES ('hit','Male','15-08-85','Employer','8','8');
INSERT INTO DEPENDENT VALUES ('tara','Female','27-07-80','Employee','9','9');
INSERT INTO DEPENDENT VALUES ('nadan','Male','07-07-81','Employer','10','10');

```

```

CREATE TABLE PROJECT (Kato Chantaphakul)
(
    Proj_Name VARCHAR(100) NOT NULL,
    Proj_No INT NOT NULL,
    Location VARCHAR(100) NOT NULL,
    Dep_No INT NOT NULL,
    CONSTRAINT PROJECT_PK PRIMARY KEY (Proj_No)
);
INSERT INTO PROJECT VALUES ('Android','1','USA','1');
INSERT INTO PROJECT VALUES ('Apple','2','UK','2');
INSERT INTO PROJECT VALUES ('Healthcare','3','USA','3');
INSERT INTO PROJECT VALUES ('Medical','4','India','4');
INSERT INTO PROJECT VALUES ('Finance','5','Canada','5');
INSERT INTO PROJECT VALUES ('Engineering','6','USA','6');
INSERT INTO PROJECT VALUES ('Product','7','Russia','7');
INSERT INTO PROJECT VALUES ('Designing','8','USA','8');
INSERT INTO PROJECT VALUES ('cyber security','9','Australia','9');
INSERT INTO PROJECT VALUES ('Construction','10','Thailand','10');

```

```

CREATE TABLE In_house (Ruta Antaliya)
(
    Inh_pro_No INT NOT NULL,
    No_hrs_week INT NOT NULL,
    Inh_pro_name VARCHAR(100) NOT NULL,
    Proj_No INT NOT NULL,
    Dept_No INT NOT NULL,
    EmpID INT NOT NULL,
    CONSTRAINT In_house_no_PK PRIMARY KEY (inh_pro_No),
    CONSTRAINT In_house_pro_FK1 FOREIGN KEY (Proj_No) REFERENCES
PROJECT(Proj_No),
    CONSTRAINT In_house_FK2 FOREIGN KEY (Dept_No) REFERENCES
DEPARTMENT(Dept_No),
    CONSTRAINT In_house_FK3 FOREIGN KEY (EmpID) REFERENCES
EMPLOYEE(EmpID)
);
INSERT INTO In_house VALUES ('7','45','healthcare project', '3','3','3');
INSERT INTO In_house VALUES ('5','45','legendary project', '1','1','1');
INSERT INTO In_house VALUES ('4','40','logo design project', '8','8','8');

```

```

CREATE TABLE Out_source  (Ruta Antalya)
(
    Out_pro_name VARCHAR(100) NOT NULL,
    Out_pro_no INT NOT NULL,
    Rate INT NOT NULL,
    Proj_No INT NOT NULL,
    CONSTRAINT Out_source_PK PRIMARY KEY (Out_pro_no),
    CONSTRAINT Out_source_FK FOREIGN KEY (Proj_No) REFERENCES
PROJECT(Proj_No)
);
INSERT INTO Out_source VALUES ('application project','1','4','2');
INSERT INTO Out_source VALUES ('medical project','2','5','4');
INSERT INTO Out_source VALUES ('finance project','3','5','5');
INSERT INTO Out_source VALUES ('product development project','6','4','7');
INSERT INTO Out_source VALUES ('cyber security project','9','5','9');
INSERT INTO Out_source VALUES ('building project','10','5','10');

```

```

CREATE TABLE DEPARTMENT_Locations  (Snehith Akkineni)
(
    Location_ID VARCHAR(100) NOT NULL,
    Location_str VARCHAR(100) NOT NULL,
    Location_zip INT NOT NULL,
    Dept_No INT NOT NULL,
    CONSTRAINT DEPARTMENT_Locations_PK PRIMARY KEY (Location_ID),
    CONSTRAINT DEPARTMENT_Locations_FK FOREIGN KEY (Dept_No) REFERENCES
DEPARTMENT(Dept_No)
);
Insert into department_locations values('1','San Diego','94563','3');
Insert into department_locations values('2','Santa Monica','78563','2');
Insert into department_locations values('3','San Francisco','90563','8');
Insert into department_locations values('4','Los Angeles','92563','1');

```

```
CREATE TABLE works_on (Snehith Akkineni)
(
Work_id INT NOT NULL,
Hrsperwk INT NOT NULL,
EmpID INT NOT NULL,
Inh_pro_No NOT NULL,
CONSTRAINT Works_on_PK PRIMARY KEY (work_id),
CONSTRAINT Works_on_FK FOREIGN KEY (EmpID) REFERENCES Employee(EmpID),
CONSTRAINT Works_on_FK1 FOREIGN KEY (inh_pro_No) REFERENCES
In_house(inh_pro_No)
);
INSERT INTO works_on VALUES ('1','45','3','7');
INSERT INTO works_on VALUES ('2','45','1','5');
INSERT INTO works_on VALUES ('3','40','8','4');
```

## DESC TableName

DESCRIBE EMPLOYEE; (Chandan Basavaraju)

Connected as RANTALI@mwang3

**Workspace**  
Enter SQL, PL/SQL and SQL\*Plus statements.  
describe Employee;

Name	Null?	Type
NAME	NOT NULL	VARCHAR2(100)
SSN	NOT NULL	NUMBER(38)
ADDRESS	NOT NULL	VARCHAR2(100)
SALARY	NOT NULL	NUMBER(38)
SEX	NOT NULL	CHAR(20)
DOB	NOT NULL	CHAR(30)
EMPID	NOT NULL	NUMBER(38)
DEPT_NO	NOT NULL	NUMBER(38)
SUPERVISION_EMPID		NUMBER(38)

DESCRIBE DEPARTMENT; (Chandan Basavaraju)

Connected as RANTALI@mwang3

**Workspace**  
Enter SQL, PL/SQL and SQL\*Plus statements.  
DESCRIBE DEPARTMENT;

Name	Null?	Type
DEPT_NO	NOT NULL	NUMBER(38)
DEPT_NAME	NOT NULL	VARCHAR2(20)
EMPID	NOT NULL	NUMBER(38)

DESCRIBE DEPENDENT; (Kato Chantaphakul)

Connected as VCHANTA@mwang3

**Workspace**  
Enter SQL, PL/SQL and SQL\*Plus statements.  
DESC DEPENDENT

Name	Null?	Type
NAME	NOT NULL	CHAR(20)
SEX	NOT NULL	CHAR(20)
DOB	NOT NULL	VARCHAR2(50)
RELATIONSHIP	NOT NULL	CHAR(50)
DEPDID	NOT NULL	NUMBER(38)
EMPID	NOT NULL	NUMBER(38)

## DESCRIBE PROJECT; (Kato Chantaphakul)

Connected as VCHANTA@mwang3

Workspace  
Enter SQL, PL/SQL and SQL\*Plus statements.  
DESC PROJECT

Clear

Execute Load Script Save Script Cancel

Name	Null?	Type
PROJ_NAME	NOT NULL	VARCHAR2(100)
PROJ_NO	NOT NULL	NUMBER(38)
LOCATION	NOT NULL	VARCHAR2(100)
DEP_NO	NOT NULL	NUMBER(38)

## DESCRIBE IN\_HOUSE; (Ruta Antaliya)

Connected as RANTALI@mwang3

Workspace  
Enter SQL, PL/SQL and SQL\*Plus statements.  
DESCRIBE In\_house;

Clear

Execute Load Script Save Script Cancel

Name	Null?	Type
INH_PRO_NO	NOT NULL	NUMBER(38)
NO_HRS_WEEK	NOT NULL	NUMBER(38)
INH_PRO_NAME	NOT NULL	VARCHAR2(100)
PROJ_NO	NOT NULL	NUMBER(38)
DEPT_NO	NOT NULL	NUMBER(38)
EMPID	NOT NULL	NUMBER(38)

## DESCRIBE OUT\_SOURCE; (Ruta Antaliya)

Connected as RANTALI@mwang3

Workspace  
Enter SQL, PL/SQL and SQL\*Plus statements.  
DESCRIBE OUT\_SOURCE;

Clear

Execute Load Script Save Script Cancel

Name	Null?	Type
OUT_PRO_NAME	NOT NULL	VARCHAR2(100)
OUT_PRO_NO	NOT NULL	NUMBER(38)
RATE	NOT NULL	NUMBER(38)
PROJ_NO	NOT NULL	NUMBER(38)

## DESCRIBE DEPARTMENT\_LOCATIONS; (Snehith Akkineni)

Connected as SAKKINE@mwang3

**Workspace**  
Enter SQL, PL/SQL and SQL\*Plus statements.  
Desc department\_locations;

Name	Null?	Type
LOCATION_ID	NOT NULL	VARCHAR2(100)
LOCATION_STR	NOT NULL	VARCHAR2(100)
LOCATION_ZIP	NOT NULL	NUMBER(38)
DEPT_NO	NOT NULL	NUMBER(38)

## DESCRIBE WORKS\_ON; (Snehith Akkineni)

Connected as SAKKINE@mwang3

**Workspace**  
Enter SQL, PL/SQL and SQL\*Plus statements.  
Desc works\_on;

Name	Null?	Type
WORK_ID	NOT NULL	NUMBER(38)
HRSWERWK	NOT NULL	NUMBER(38)
EMPID	NOT NULL	NUMBER(38)
INH_PRO_NO	NOT NULL	NUMBER(38)

## select \* from TableName

select \* from EMPLOYEE; (Chandan Basavaraju)

Connected as RANTALI@mwang3

**Workspace**

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

```
select * from EMPLOYEE;
```

NAME	SSN	ADDRESS	SALARY	SEX	DOB	EMPID	DEPT_NO	SUPERVISION_EMPID
Ruta	786987567	355 S Hines	2000	Female	27-07-87	1	1	1
Chandan	986987567	5145 S.W.17th	1000	Male	07-07-81	2	2	2
Kato	346987567	355 Hines Blvd	1500	Male	27-09-87	3	3	3
Snehit	906987567	355 Hines Blvd	2500	Male	15-08-85	4	4	4
Kanoh	126987567	312 Kiowai St	2000	Female	27-07-80	5	5	5
Kalo	346990567	45 Capitol Ave	1500	Male	27-09-87	7	7	7
hit	906567567	424 Missouri Ave	2500	Male	15-08-85	8	8	8
tara	709987567	4 Rocky Point Dr	2000	Female	27-07-80	9	9	9
nadan	986987598	45 S 17th St	1000	Male	07-07-81	10	10	10

select \* from DEPARTMENT; (Chandan Basavaraju)

Connected as RANTALI@mwang3

**Workspace**

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

```
select * from DEPARTMENT;
```

DEPT_NO	DEPT_NAME	EMPID
1	IT Division	1
2	Account Division	2
3	Healthcare Division	3
4	HR Division	4
5	Production Division	5
7	Networking Division	7
8	Medical Division	8
9	Biology Division	9

SELECT \* FROM Dependent; (Kato Chantaphakul)

Connected as VCHANTA@mwang3

**Workspace**

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

```
SELECT * FROM Dependent;
```

NAME	SEX	DOB	RELATIONSHIP	DEPDID	EMPID
Kaden	Male	27-09-87	Customer	7	7
hit	Male	15-08-85	Employer	8	8
tara	Female	27-07-80	Employee	9	9
nadan	Male	07-07-81	Employer	10	10

SELECT \* FROM Project; (Kato Chantaphakul)

Connected as VCHANTA@mwang3

Workspace

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

SELECT \* FROM Project;

Execute Load Script Save Script Cancel Clear

PROJ_NAME	PROJ_NO	LOCATION	DEP_NO
Android	1 USA		1
Apple	2 UK		2
Healthcare	3 USA		3
Medical	4 India		4
Finance	5 Canada		5
Engineering	6 USA		6
Product	7 Russia		7
Designing	8 USA		8
cyber security	9 Australia		9
Construction	10 Thailand		10

10 rows selected.

select \* from IN\_HOUSE; (Ruta Antaliya)

Connected as RANTALI@mwang3

Workspace

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

select \* from IN\_HOUSE;

Execute Load Script Save Script Cancel Clear

INH_PRO_NO	NO_HRS_WEEK	INH_PRO_NAME	PROJ_NO	DEPT_NO	EMPID
7	45	healthcare project	3	3	3
5	45	legendary project	1	1	1
4	40	logo design project	8	8	8

select \* from OUT\_SOURCE; (Ruta Antaliya)

Connected as RANTALI@mwang3

Workspace

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

select \* from OUT\_SOURCE;

Execute Load Script Save Script Cancel Clear

OUT_PRO_NAME	OUT_PRO_NO	RATE	PROJ_NO
application project	1	4	2
medical project	2	5	4
finance project	3	5	5
product development project	6	4	7
cyber security project	9	5	9
building project	10	5	10

Select \* from DEPARTMENT\_LOCATIONS; (Snehith Akkineni)

Connected as SAKKINE@mwang3

Workspace

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

```
select * from department_locations;
```

Clear

Execute Load Script Save Script Cancel

LOCATION_ID	LOCATION_STR	LOCATION_ZIP	DEPT_NO
1	San Diego	94563	3
2	Santa Monica	78563	2
3	San Francisco	90563	8
4	Los Angeles	92563	1
5	Artesia	90701	4
6	Monterey Park	91754	7
7	Chicago	60007	9
8	New York	10009	6

8 rows selected.

Select \* from WORKS\_ON; (Snehith Akkineni)

Connected as SAKKINE@mwang3

Workspace

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

```
select * from works_on;
```

Clear

Execute Load Script Save Script Cancel

WORK_ID	HRSPERWK	EMPID	INH_PRO_NO
1	45	3	7
2	45	1	5
3	40	8	4

## INSERTION (Ruta Antalya)

### INSERT Value in Employee Table

NAME	SSN	ADDRESS	SALARY	SEX	DOB	EMPID	DEPT_NO	SUPERVISION_EMPID
Ruta	786987567	355 S Hines	2000	Female	27-07-87	1	1	1
Chandan	986987567	5145 S.W.17th	1000	Male	07-07-81	2	2	2
Kato	346987567	355 Hines Blvd	1500	Male	27-09-87	3	3	3
Snehit	906987567	355 Hines Blvd	2500	Male	15-08-85	4	4	4
Kanoh	126987567	312 Kiowai St	2000	Female	27-07-80	5	5	5
Kalo	346990567	45 Capitol Ave	1500	Male	27-09-87	7	7	7
hit	906567567	424 Missouri Ave	2500	Male	15-08-85	8	8	8
tara	709987567	4 Rocky Point Dr	2000	Female	27-07-80	9	9	9
nadan	986987598	45 S 17th St	1000	Male	07-07-81	10	10	10

### INSERT INTO EMPLOYEE VALUES

```
('Anan','909987596','45 S 17th St','1500','Male','07-07-71','11','11','11');
```

Connected as RANTALI@mwang3

#### Workspace

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

```
select * from EMPLOYEE;
```

NAME	SSN	ADDRESS	SALARY	SEX	DOB	EMPID	DEPT_NO	SUPERVISION_EMPID
Ruta	786987567	355 S Hines	2000	Female	27-07-87	1	1	1
Chandan	986987567	5145 S.W.17th	1000	Male	07-07-81	2	2	2
Kato	346987567	355 Hines Blvd	1500	Male	27-09-87	3	3	3
Snehit	906987567	355 Hines Blvd	2500	Male	15-08-85	4	4	4
Kanoh	126987567	312 Kiowai St	2000	Female	27-07-80	5	5	5
Kalo	346990567	45 Capitol Ave	1500	Male	27-09-87	7	7	7
hit	906567567	424 Missouri Ave	2500	Male	15-08-85	8	8	8
tara	709987567	4 Rocky Point Dr	2000	Female	27-07-80	9	9	9
nadan	986987598	45 S 17th St	1000	Male	07-07-81	10	10	10
Anan	909987596	45 S 17th St	1500	Male	07-07-71	11	11	11

## UPDATE (Ruta Antalya)

### UPDATE value in In\_house Table

INH_PRO_NO	NO_HRS_WEEK	INH_PRO_NAME	PROJ_NO	DEPT_NO	EMPID
7	45	healthcare project	3	3	3
5	45	legendary project	1	1	1
4	40	logo design project	8	8	8

### UPDATE In\_house

```
SET Inh_pro_name = 'Finance_Project'
```

```
WHERE Inh_pro_No = 7;
```

Connected as RANTALI@mwang3

#### Workspace

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

```
select * from In_house;
```

INH_PRO_NO	NO_HRS_WEEK	INH_PRO_NAME	PROJ_NO	DEPT_NO	EMPID
7	45	Finance_Project	3	3	3
5	45	legendary project	1	1	1
4	40	logo design project	8	8	8

## DELETE (Ruta Antalya)

### DELETE Value in Employee Table

NAME	SSN	ADDRESS	SALARY	SEX	DOB	EMPID	DEPT_NO	SUPERVISION_EMPID
Ruta	786987567	355 S Hines	2000	Female	27-07-87	1	1	1
Chandan	986987567	5145 S.W.17th	1000	Male	07-07-81	2	2	2
Kato	346987567	355 Hines Blvd	1500	Male	27-09-87	3	3	3
Snehit	906987567	355 Hines Blvd	2500	Male	15-08-85	4	4	4
Kanoh	126987567	312 Kiowai St	2000	Female	27-07-80	5	5	5
Kalo	346990567	45 Capitol Ave	1500	Male	27-09-87	7	7	7
hit	906567567	424 Missouri Ave	2500	Male	15-08-85	8	8	8
tara	709987567	4 Rocky Point Dr	2000	Female	27-07-80	9	9	9
nadan	986987598	45 S 17th St	1000	Male	07-07-81	10	10	10
Anan	909987598	45 S 17th St	1500	Male	07-07-71	11	11	11

DELETE FROM EMPLOYEE

WHERE EMPID = 11;

Connected as RANTALI@mwang3

#### Workspace

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

SELECT \* FROM EMPLOYEE;

NAME	SSN	ADDRESS	SALARY	SEX	DOB	EMPID	DEPT_NO	SUPERVISION_EMPID
Ruta	786987567	355 S Hines	2000	Female	27-07-87	1	1	1
Chandan	986987567	5145 S.W.17th	1000	Male	07-07-81	2	2	2
Kato	346987567	355 Hines Blvd	1500	Male	27-09-87	3	3	3
Snehit	906987567	355 Hines Blvd	2500	Male	15-08-85	4	4	4
Kanoh	126987567	312 Kiowai St	2000	Female	27-07-80	5	5	5
Kalo	346990567	45 Capitol Ave	1500	Male	27-09-87	7	7	7
hit	906567567	424 Missouri Ave	2500	Male	15-08-85	8	8	8
tara	709987567	4 Rocky Point Dr	2000	Female	27-07-80	9	9	9
nadan	986987598	45 S 17th St	1000	Male	07-07-81	10	10	10

## CREATE VIEW (Ruta Antalya)

### CREATE VIEW for city

CREATE VIEW Cty\_Vw

AS SELECT name AS EMP\_NAME, Location\_str AS city1

FROM EMPLOYEE, DEPARTMENT\_LOCATIONS

WHERE EMPLOYEE.Dept\_No=DEPARTMENT\_LOCATIONS.Dept\_No;

select \* from Cty\_Vw;

Connected as RANTALI@mwang3

#### Workspace

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

select \* from Cty\_Vw;

EMP_NAME	CITY1
Ruta	Los Angeles
Chandan	Santa Monica
Kato	San Diego
hit	San Francisco

## Create View - (Snehith Akkineni)

//Creating a View to display employee ID's who work more than or equal to 45 hours.

Create View work\_V

As

Select work\_id, EmpID, hrsperwk, inh\_pro\_no

From works\_on

Where hrsperwk <= 45;

Select from work\_V;

The screenshot shows the Oracle SQL Developer workspace interface. At the top right, it says "Connected as SAKKINE@mwang3". Below that is a toolbar with "Clear" and other buttons. The main area is titled "Workspace" and contains a text editor with the following SQL code:

```
Create View work_V
as
Select Work_Id, EmpID, hrsperwk, inh_pro_no
from works_on
where hrsperwk <= 45;

Select * from work_V;
```

Below the code are four buttons: "Execute", "Load Script", "Save Script", and "Cancel". A message "View created." is displayed below the buttons. To the right of the workspace is a preview pane showing the results of the query:

WORK_ID	EMPID	HRSPERWK	INH_PRO_NO
1	3	45	7
2	1	45	5
3	8	40	4

**List all the EMPLOYEES in the company that salary between \$1300 and \$2000. (Ruta Antaliya)**

Ans: SELECT name AS EMP\_NAME, salary AS Salary  
FROM EMPLOYEE  
WHERE salary BETWEEN 1300 and 2000;

#### Workspace

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

```
--1. List all the employees in the company that salary between $1300 and $2000.  
SELECT name AS EMP_NAME, salary AS Salary FROM EMPLOYEE WHERE  
salary BETWEEN 1300 and 2000;
```

Connected as RANTALI@mwang3

[Clear](#)

[Execute](#) [Load Script](#) [Save Script](#) [Cancel](#)

EMP_NAME	SALARY
Ruta	2000
Kato	1500
Kanoh	2000
Kalo	1500
tara	2000

## SUBQUERY

**Retrieve Project Name from project table where department number equal 4. (Ruta Antaliya)**

```
SELECT Proj_No, Proj_Name  
FROM PROJECT  
WHERE Dep_No =  
(SELECT Dept_No  
FROM DEPARTMENT  
WHERE Dept_No=4);
```

#### Workspace

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

```
SELECT Proj_No, Proj_Name  
FROM PROJECT  
WHERE Dep_No =  
(SELECT Dept_No  
FROM DEPARTMENT  
WHERE Dept_No=4);
```

Connected as RANTALI@mwang3

[Clear](#)

[Execute](#) [Load Script](#) [Save Script](#) [Cancel](#)

PROJ_NO	PROJ_NAME
4	Medical

// To fond the zip code of location by using subquery (Snehith Akkineni)

Select location\_zip  
From department\_locations  
Where  
dept\_no = (select dept\_no from department where dept\_no = 4);

Connected as SAKKINE@mwang3

Workspace

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

```
select location_zip
from department_locations
where
dept_no = (select dept_no from department where dept_no = 4);
```

Clear

Execute Load Script Save Script Cancel

LOCATION_ZIP
90701

//Finding minimum salary (Kato Chantaphakul)

SELECT Empid, Salary, Name  
FROM Employee  
WHERE Salary = (Select MIN (salary)  
FROM Employee);

Connected as VCHANTA@mwang3

Workspace

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

```
SELECT Empid, Salary, Name
FROM Employee
WHERE Salary = (Select MIN (salary)
FROM Employee);
```

Clear

Execute Load Script Save Script Cancel

EMPID	SALARY	NAME
2	1000	Chandan
10	1000	nadan

## **GROUP BY**

**No of Project in USA.** (Ruta Antaliya)

```
SELECT Proj_Name, COUNT(Proj_No) AS Proj_No  
FROM PROJECT where Location='USA'  
GROUP BY Proj_Name  
HAVING COUNT (Proj_No) <3;
```

—

Connected as RANTALI@mwang3

**Workspace**

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

```
SELECT Proj_Name, COUNT(Proj_No) AS Proj_No  
FROM PROJECT where Location='USA'  
GROUP BY Proj_Name  
HAVING COUNT (Proj_No) <3;
```

(G)

Execute Load Script Save Script Cancel

PROJ_NAME	PROJ_NO
Healthcare	1
Android	1
Designing	1
Engineering	1

**To display employee ID count using GROUPBY.** (Snehith Akkineni)

```
Select Empid, count(work_id)  
From works_on  
GROUP BY empid  
Having count(work_id) < 2;
```

—

Connected as SAKKINE@mwang3

**Workspace**

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

```
SELECT EmpID, COUNT(work_id)  
FROM Works_on  
GROUP BY EmpID  
HAVING COUNT(work_ID) < 2;
```

Execute Load Script Save Script Cancel

EMPID	COUNT(WORK_ID)
8	1
3	1
1	1

--Display Employee who has work less than 40 Hours per week using GROUP BY  
(Kato Chantaphakul)

```
SELECT Emp1.Name, count(Works_on.HrsPerWk) AS HoursPerWeek
FROM Employee Emp1, Works_on
WHERE Emp1.EmpID = Works_on.EmpID
GROUP BY Emp1.Name
HAVING COUNT (Works_on.HrsPerWk) < 40;
```

Connected as VCHANTA@mwang3

Workspace

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

```
SELECT Emp1.Name, count(Works_on.HrsPerWk) AS HoursPerWeek
FROM Employee Emp1, Works_on
WHERE Emp1.EmpID = Works_on.EmpID
GROUP BY Emp1.Name
HAVING COUNT (Works_on.HrsPerWk) < 40;
```

Clear

Execute Load Script Save Script Cancel

NAME	HOURS PER WEEK
Ruta	1
hit	1
Kato	1

## JOIN

In house Project for different Employee. (Ruta Antaliya)

```
SELECT EMPLOYEE.Name, EMPLOYEE.SSN, In_house.Inh_pro_name
FROM EMPLOYEE INNER JOIN In_house ON
EMPLOYEE.EmpID = In_house.EmpID
ORDER BY EMPLOYEE.EmpID;
```

Connected as RANTALI@mwang3

Workspace

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

```
SELECT EMPLOYEE.Name, EMPLOYEE.SSN, In_house.Inh_pro_name
FROM EMPLOYEE INNER JOIN In_house ON
EMPLOYEE.EmpID = In_house.EmpID
ORDER BY EMPLOYEE.EmpID;
```

Clear

Execute Load Script Save Script Cancel

NAME	SSN	INH_PRO_NAME
Ruta	786987567	legendary project
Kato	346987567	Finance_Project
hit	906567567	logo design project

## Joining 3 tables Employee, In\_house and works\_on. (Snehith Akkineni)

Select works\_on.work\_id, work\_on.hrsperwk, employee.empid, in\_house.inh\_pro\_no from works\_on, employee, in\_house where works\_on.empid = employee.empid and works\_on.inh\_pro\_no = in\_house.ihn\_pro\_no;

Connected as SAKKINE@mwang3

Workspace

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

```
select works_on.work_id, work_on.hrsperwk, employee.empid,
in_house.inh_pro_no from works_on, employee, in_house where
works_on.empid = employee.empid and works_on.inh_pro_no =
in_house.ihn_pro_no;
```

Clear

Execute Load Script Save Script Cancel

WORK_ID	HRSPERWK	EMPID	INH_PRO_NO
1	45	3	7
2	45	1	5
3	40	8	4

## //Joining data from Employee Table and Dependent Table (Kato Chantaphakul)

```
SELECT Employee.EmpID, Employee.Name, Employee.DOB, Employee.sex
FROM Employee, Dependent
WHERE Employee.EmpID = Dependent.EmpID;
```

Connected as VCHANTA@mwang3

Workspace

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

```
SELECT Employee.EmpID, Employee.Name, Employee.DOB,
Employee.sex
FROM Employee, Dependent
WHERE Employee.EmpID = Dependent.EmpID;
```

Clear

Execute Load Script Save Script Cancel

EMPID	NAME	DOB	SEX
7	Kalo	27-09-87	Male
8	hit	15-08-85	Male
9	tara	27-07-80	Female
10	nadan	07-07-81	Male

## PL/SQL Statement block

```
CREATE OR REPLACE PROCEDURE query_In_house (Ruta Antalya)
  (v_id      IN In_house.Inh_pro_No%TYPE,
   v_Projname OUT In_house.Inh_pro_name%TYPE,
   v_hrs_week OUT In_house.No_hrs_week %TYPE)
IS
BEGIN
  SELECT Inh_pro_name, No_hrs_week
  INTO v_Projname, v_hrs_week
  from In_house
  WHERE Inh_pro_No =v_id;
END query_In_house;
```

Connected as RANTALI@mwang3

Clear

**Workspace**

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

```
CREATE OR REPLACE PROCEDURE query_In_house
  (v_id      IN In_house.Inh_pro_No%TYPE,
   v_Projname OUT In_house.Inh_pro_name%TYPE,
   v_hrs_week OUT In_house.No_hrs_week %TYPE)
IS
BEGIN
  SELECT Inh_pro_name, No_hrs_week
  INTO v_Projname, v_hrs_week
  from In_house
  WHERE Inh_pro_No =v_id;
```

Execute Load Script Save Script Cancel

Procedure created.

```
variable g_Projname varchar2(100)
variable g_hrs_week number
execute query_In_house(40, :g_Projname, :g_hrs_week)
print g_Projname g_hrs_week
```

G\_PROJNAME

logo design project

G\_HRS\_WEEK

40

## (Ruta Antaliya)

DESC query\_In\_house;

Connected as RANTALI@mwang3

### Workspace

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

DESC query\_In\_house;

PROCEDURE query\_In\_house

Argument Name	Type	In/Out	Default?
V_ID	NUMBER(38)	IN	
V_PROJNAME	VARCHAR2(100)	OUT	
V_HRS_WEEK	NUMBER(38)	OUT	

## (Snehith Akkineni)

CREATE OR REPLACE PROCEDURE SA(social in number, sal in number)

AS

BEGIN

UPDATE EMPLOYEE

SET

SALARY = SAL

WHERE SSN = social;

END;

Connected as SAKKINE@mwang3

### Workspace

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

```
CREATE OR REPLACE PROCEDURE SA(social in number, sal in number)
AS
BEGIN
UPDATE EMPLOYEE
SET
SALARY = SAL
WHERE SSN = social;
END;
```

Procedure created.

## PROCEDURE CALL: (Snehith Akkineni)

Previously Employee table

Connected as SAKKINE@mwang3

**Workspace**

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

```
select * from employee;
```

NAME	SSN	ADDRESS	SALARY	SEX	DOB	EMPID	DEPT_NO	SUPERVISION_EMPID
Ruta	786987567	355 S Hines	2000	Female	27-07-87	1	1	1
Chandan	986987567	5145 S.W.17th	1000	Male	07-07-81	2	2	2
Kato	346987567	355 Hines Blvd	1500	Male	27-09-87	3	3	3
Snehit	906987567	355 Hines Blvd	2500	Male	15-08-85	4	4	4
Kanoh	126987567	312 Kiowai St	2000	Female	27-07-80	5	5	5
Mountain	126987567	5689 College Ave	1000	Male	07-07-81	6	6	6
Kalo	346990567	45 Capitol Ave	1500	Male	27-09-87	7	7	7
hit	906567567	424 Missouri Ave	2500	Male	15-08-85	8	8	8
tara	709987567	4 Rocky Point Dr	2000	Female	27-07-80	9	9	9
nadan	986987598	45 S 17th St	1000	Male	07-07-81	10	10	10

10 rows selected.

## After updating the employee table:

Exec SA (906987567, 3200)

Connected as SAKKINE@mwang3

**Workspace**

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

```
EXEC SA(906987567, 3200)
```

PL/SQL procedure successfully completed.

## Employee table after updating procedure:

NAME	SSN	ADDRESS	SALARY	SEX	DOB	EMPID	DEPT_NO	SUPERVISION_EMPID
Ruta	786987567	355 S Hines	2000	Female	27-07-87	1	1	1
Chandan	986987567	5145 S.W.17th	1000	Male	07-07-81	2	2	2
Kato	346987567	355 Hines Blvd	1500	Male	27-09-87	3	3	3
Snehit	906987567	355 Hines Blvd	3200	Male	15-08-85	4	4	4
Kanoh	126987567	312 Kiowai St	2000	Female	27-07-80	5	5	5
Mountain	126987567	5689 College Ave	1000	Male	07-07-81	6	6	6
Kalo	346990567	45 Capitol Ave	1500	Male	27-09-87	7	7	7
hit	906567567	424 Missouri Ave	2500	Male	15-08-85	8	8	8
tara	709987567	4 Rocky Point Dr	2000	Female	27-07-80	9	9	9
nadan	986987598	45 S 17th St	1000	Male	07-07-81	10	10	10

10 rows selected.

## FUNCTION

```
CREATE OR REPLACE FUNCTION get_hrs_week (Ruta Antaliya)
(v_id IN In_house.Inh_pro_No%TYPE) RETURN NUMBER
IS
    v_hrs_week In_house.No_hrs_week%TYPE := 0;
BEGIN
    SELECT No_hrs_week
    INTO v_hrs_week
    FROM In_house
    WHERE Inh_pro_No=v_id;
    RETURN (v_hrs_week);
END get_hrs_week;
/
```

Connected as RANTALI@mwang3

### Workspace

```
Enter SQL, PL/SQL and SQL*Plus statements.  
Connected as RANTALI@mwang3  
CREATE OR REPLACE FUNCTION get_hrs_week  
(v_id IN In_house.Inh_pro_No%TYPE) RETURN NUMBER  
IS  
    v_hrs_week In_house.No_hrs_week%TYPE := 0;  
BEGIN  
    SELECT No_hrs_week  
    INTO v_hrs_week  
    FROM In_house  
    WHERE Inh_pro_No=v_id;  
    RETURN (v_hrs_week);  
END get_hrs_week;
```

Function created.

```
VARIABLE g_hrs_week number
exec :g_hrs_week := get_hrs_week(45)
PRINT g_hrs_week
```

G_HRS_WEEK
45

--Counting total number of projects (Kato Chantaphakul)

```
CREATE OR REPLACE FUNCTION totalProjects
RETURN number
IS total number(4) := 0;
BEGIN
    SELECT COUNT(*) into total
    FROM project;
    RETURN total;
END;
```

## Workspace

Connected as VCHANTA@mwang3

```
Enter SQL, PL/SQL and SQL*Plus statements.  
CREATE OR REPLACE FUNCTION totalProjects  
RETURN number  
IS total number(4) := 0;  
BEGIN  
    SELECT COUNT(*) into total  
    FROM project;  
    RETURN total;  
END;
```

Clear

Execute Load Script Save Script Cancel

Function created.

## --Call Function (Kato Chantaphakul)

```
SET SERVEROUTPUT ON  
DECLARE  
    totalProj number(4);  
BEGIN  
    totalProj := totalProjects();  
    dbms_output.put_line('Total Number of Projects: ' || totalProj);  
END;
```

Connected as VCHANTA@mwang3

## Workspace

```
Enter SQL, PL/SQL and SQL*Plus statements.  
SET SERVEROUTPUT ON  
DECLARE  
    totalProj number(4);  
BEGIN  
    totalProj := totalProjects();  
    dbms_output.put_line('Total Number of Projects: ' || totalProj);  
END;
```

Clear

Execute Load Script Save Script Cancel

Total Number of Projects: 10

PL/SQL procedure successfully completed.

PROJ_NAME	PROJ_NO	LOCATION	DEP_NO
Android	1	USA	1
Apple	2	UK	2
Healthcare	3	USA	3
Medical	4	India	4
Finance	5	Canada	5
Engineering	6	USA	6
Product	7	Russia	7
Designing	8	USA	8
cyber security	9	Australia	9
Construction	10	Thailand	10

10 rows selected.

## (Snehith Akkineni)

Create or Replace function location (zip in number) return char IS name varchar(50);

BEGIN

Select location\_str into name

from department\_locations

where location\_zip = zip;

```
RETURN name;
```

```
END;
```

Connected as SAKKINE@mwang3

Workspace

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

```
create or replace function location (zip in number) return char IS name
varchar(50);
BEGIN
Select location_str into name
from department_locations
where location_zip = zip;
RETURN name;
END;
```

Clear

Execute Load Script Save Script Cancel

Function created.

### Function\_call (Snehit AKkineni)

```
Variable d number(10);
```

Connected as SAKKINE@mwang3

Workspace

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

```
Variable d varchar2(50);
```

Clear

Execute Load Script Save Script Cancel

SP2-0863: SQL\*Plus processing completed

```
Exec:d:= location(90701);
```

Connected as SAKKINE@mwang3

Workspace

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

```
Exec:d:= location(90701);
```

Clear

Execute Load Script Save Script Cancel

PL/SQL procedure successfully completed.

```
Print d();
```

Connected as SAKKINE@mwang3

Workspace

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

```
Print d();
```

Clear

Execute Load Script Save Script Cancel

D

Artesia

## **Object type :** (Snehith Akkineni)

```
CREATE OR REPLACE TYPE Address_new AS Object  
  
(street Varchar2(50),  
  
City varchar2(25),  
  
State char(2),  
  
Zip number);
```

Connected as SAKKINE@mwang3

**Workspace**

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

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

**Execute** **Load Script** **Save Script** **Cancel**

Type created.

CREATE TABLE empAddress (Snehith Akkineni)

```
(Emid number,  
  
Address address_new);
```

Connected as SAKKINE@mwang3

**Workspace**

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

```
CREATE TABLE empAddress  
(EmID Number,  
Address address_new);
```

**Execute** **Load Script** **Save Script** **Cancel**

Table created.

Insert into empAddress values

```
(100, address_new, '4127 Atlantic Blvd', 'Monterey Park', 'CA', '91754');
```

Insert into empAddress values

```
(200, address_new, '4325 Cracck Blvd', 'Hellman', 'CA', '98655');
```

Insert into empAddress values

```
(300, address_new, '18612 wear Blvd', 'Scotch', 'CA', '90654');
```

Connected as SAKKINE@mwang3

**Workspace**

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

```
INSERT INTO empAddress VALUES (100,
address_new ('4127 Atlantic Blvd', 'Monterey park', 'CA', 91754));

INSERT INTO empAddress VALUES (200,
address_new ('4325 Craack Blvd', 'Hellman', 'CA', 98655));

INSERT INTO empAddress VALUES (300,
address_new ('18612 Wear Blvd', 'Scotch', 'CA', 90654));
```

1 row created.  
1 row created.  
1 row created.

Select Emid, o.Address.Street from empAddress o;

Connected as SAKKINE@mwang3

**Workspace**

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

```
SELECT EmID, o.Address.Street FROM empAddress o;
```

EMID	ADDRESS.STREET
100	4127 Atlantic Blvd
200	4325 Craack Blvd
300	18612 Wear Blvd

### ORDMS – (Snehith Akkineni)

CREATE OR REPLACE TYPE Employee\_details AS Object  
(FN char(30), MN char(30), LN char(30));

Connected as SAKKINE@mwang3

**Workspace**

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

```
CREATE OR REPLACE TYPE Employee_details AS Object
(FN char(30), MN char(30), LN char(30));
```

Type created.

Create table Employ\_De(E\_ID int, Ename Employee\_details);

Connected as SAKKINE@mwang3

**Workspace**

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

```
Create table Employ_De(E_ID int, Ename Employee_details);
```

Table created.

```
Insert into Employ_De values(2, Employee_details('Andrew', 'Bero', 'Cary'));
```

```
Insert into Employ_De values(3, Employee_details('Der', 'Hre', 'Hyt'));
```

```
Insert into Employ_De values(4, Employee_details('Tre', 'Uyr', 'Jpe'));
```

Connected as SAKKINE@mwang3

**Workspace**

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

```
Insert into Employ_De values(2, Employee_details('Andrew', 'Bero', 'Cary'));
Insert into Employ_De values(3, Employee_details('Der', 'Hre', 'Hyt'));
Insert into Employ_De values(4, Employee_details('Tre', 'Uyr', 'Jpe'));
```

1 row created.  
1 row created.  
1 row created.

```
Select e_id, e.ename.fn, e.ename.mn, e.ename.ln from employ_de e;
```

Connected as SAKKINE@mwang3

**Workspace**

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

```
Select e_id, e.ename.fn, e.ename.mn, e.ename.ln from employ_de e;
```

E_ID	ENAME.FN	ENAME.MN	ENAME.LN
2	Andrew	Bero	Cary
3	Der	Hre	Hyt
4	Tre	Uyr	Jpe

## **OBJECT TYPE** (Ruta Antalya)

```
CREATE TYPE Projc_Name As OBJECT  
(Projc_No NUMBER,  
EMPlly_Name VARCHAR2(100),  
Projc_Location VARCHAR2(100))
```

Connected as RANTALI@mwang3

### Workspace

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

```
CREATE TYPE Projc_Name As OBJECT  
(Projc_No NUMBER,  
EMPlly_Name VARCHAR2(100),  
Projc_Location VARCHAR2(100))
```

Type created.

```
CREATE TABLE EMPL2
```

```
(EMPL_ID NUMBER, Project1 Projc_Name)
```

```
INSERT INTO EMPL2 VALUES ('1',Projc_Name('1','abc','USA'));
```

```
INSERT INTO EMPL2 VALUES ('2',Projc_Name('2','xyz','INDIA'));
```

```
SELECT EMPL_ID, p.Project1.EMPlly_Name from EMPL2 p;
```

Connected as RANTALI@mwang3

### Workspace

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

```
SELECT EMPL_ID, p.Project1.EMPlly_Name from EMPL2 p;
```

### PROJECT1.EMPL\_NAME

EMPL_ID	PROJECT1.EMPL_NAME
1	abc
2	xyz

```
select * from EMPL2;
```

Connected as RANTALI@mwang3

### Workspace

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

```
select * from EMPL2;
```

### PROJECT1(PROJC\_NO, EMPL\_NAME, PROJC\_LOCATION)

EMPL_ID	PROJECT1(PROJC_NO, EMPL_NAME, PROJC_LOCATION)
1	PROJC_NAME(1, 'abc', 'USA')
2	PROJC_NAME(2, 'xyz', 'INDIA')

## VARRAY (Ruta Antaliya)

```
CREATE TYPE varray_proj_empl AS VARRAY(3) OF VARCHAR2(14);
ALTER TABLE EMPL2 ADD (EMPL_Name varray_proj_empl);
UPDATE EMPL2
SET EMPL_Name = (varray_proj_empl('bajaj',
'honda','cvc'))
WHERE EMPL_ID = 1;
```

```
Select * from EMPL2;
```

Connected as RANTALI@mwang3

Workspace

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

select \* from EMPL2;

Clear

Execute Load Script Save Script Cancel

EMPL_ID	PROJECT1(PROJC_NO, EMPLY_NAME, PROJC_LOCATION)	EMPLY_NAME
1	PROJC_NAME(1, 'abc', 'USA')	VARRAY_PROJ_EMPL('bajaj', 'honda', 'cvc')
2	PROJC_NAME(2, 'xyz', 'INDIA')	

## Object Views on a Relational Table (Ruta Antaliya)

### In\_house Table:

```
CREATE TABLE In_house
```

```
(
```

```
Inh_pro_No INT NOT NULL,
No_hrs_week INT NOT NULL,
Inh_pro_name VARCHAR(100) NOT NULL,
Proj_No INT NOT NULL,
Dept_No INT NOT NULL,
EmpID INT NOT NULL,
CONSTRAINT In_house_PK PRIMARY KEY (inh_pro_No),
CONSTRAINT In_house_FK1 FOREIGN KEY (Proj_No) REFERENCES
PROJECT(Proj_No),
CONSTRAINT In_house_FK2 FOREIGN KEY (Dept_No) REFERENCES
DEPARTMENT(Dept_No),
CONSTRAINT In_house_FK3 FOREIGN KEY (EmpID) REFERENCES
EMPLOYEE(EmpID));
```

```
INSERT INTO In_house VALUES ('7','45','healthcare project', '3','3','3');
INSERT INTO In_house VALUES ('5','45','legendary project', '1','1','1');
INSERT INTO In_house VALUES ('4','40','logo design project', '8','8','8');
```

### In\_house Relational Table:

```
CREATE TYPE In_house_Project1_type AS OBJECT(
InhProj_No NUMBER(10),
Hrper_week NUMBER(10),
Dept_No NUMBER(10),
Projt_No NUMBER(10),
EmplID NUMBER(10));
```

```
CREATE VIEW IN_house_Proj_view OF In_house_Project1_type
WITH OBJECT IDENTIFIER (InhProj_No) AS
SELECT o.Inh_pro_No, o.No_hrs_week, o.Dept_No, o.Proj_No, o.EmpID
FROM In_house o
WHERE o.Inh_pro_No = 7;
select * from IN_house_Proj_view1;
```

Connected as RANTALI@mwang3

Workspace

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

```
select * from IN_house_Proj_view1;
```

Clear

Execute Load Script Save Script Cancel

INHPROJ_NO	HRPER_WEEK	DEPTT_NO	PROJT_NO	EMPLID
7	45	3	3	3

### Create or Replace Function Employee (Chandan)

Set ServerOutput on

Declare

```
salary();
```

Begin

```
salary <= 1500;
```

```
dbms_output.put_line ( total number of employee salary <=1500 // employee);
```

```
END;
```

Connected as CBASAVA@mwang3

Workspace

```
Enter SQL, PL/SQL and SQL*Plus statements.  
Set ServerOutput on  
Declare  
salary()  
Begin  
salary <= 1500;  
dbms_output.put_line ( total number of employee salary <=1500 // employee);  
END;
```

Connected as CBASAVA@mwang3

Workspace

```
Enter SQL, PL/SQL and SQL*Plus statements.  
Create or Replace function employee  
Return salary  
IS salary()  
Begin = select employee into salary  
from employee  
return EID  
END;
```

## Show ERRORS (Kato Chantaphakul)

Show ERRORS PROCEDURE Proc\_name

Show ERRORS FUNCTION Func\_name

Connected as VCHANTA@mwang3

Workspace

```
Enter SQL, PL/SQL and SQL*Plus statements.  
Show ERRORS  
Show ERRORS PROCEDURE Proc_name  
Show ERRORS FUNCTION Func_name
```

No errors.

No errors.

No errors.

## **UML DIAGRAM** (Ruta Antalya)

