AMERICAN INTERNATIONAL UNIVERSITY - BANGLADESH

INTRODUCTION TO DATABASE

FINAL-TERM PROJECT

Section: [A]

Title: Botanical Farm Management System

Submitted By:

Group no - 02

Name	ID	Contribution (%)
MD. FARHAN SADIK	21-44403-1	50
OTHOYE PROMETE	21-44509-1	5
Tahseen Habib Maliha	21-44402-1	30
FABLIHA HASNIN DUTI	21-44396-1	15

Table of contents:

1.Project Description:

Main objective: Providing a management system for a botanical farm where pharmaceutical companies search and purchase plants from the botanical farm for various needs

Description: Plants for commercial sale are grown on a botanical farm. The primary goal of this database is to assist businesses in locating and purchasing plants for purposes such as genetically modified plants and fruits, herbal medicine production, clinical trials for experimental medicines, textile work, and research. The database would be able to choose various plants based on the demands of the user (business). The farm would also be able to keep track of inventory and sales for all of the plants in the garden.

As it is seen that a mechanism must exist within the system to provide a relationship between the farm and the plants, the entity 'plants detail' is formed to show that a farm may grow one plant or many plants while holding all their details such as plantID (primary key), plantName, Quantity, GM

The farm also has to adhere to their customer or the companies they are dealing with, therefore a relationship is created between these two entities 'farm' and 'Customer' respectively where many companies can inquire for the plants they require. The attributes such as CustomerID (primary key), CompanyName, Address are used for the 'Customer' entity

In order to function, the farm requires the aid of its workers or formally its employees (alongside their details) and so a relationship between the entity's 'farm' and 'Employees' is created where many employees would work to allow
the farm to function. The 'Employees' entity would contain EmpID (primary key), EmpName, EmpSalary, EmpPhoneNumber

Finally, the 'farm' is created to centralize all activities and is addressed by multiple other instances of other entities for inquiries whenever the need arises, whether it's from a customer or in their plant production or the employees who are working there. The ‘farm’ entity would contain BatchNo (primary key), CustomerID (foreign key), EmpID (foreign key), plantID (foreign key)

These table are still in pre-normalized format and will have to be revised to remove any redundancies

Entity	Attributes	Description
Plant Detail	plantID (primary key)	Uniquely identifies a plant in the farm which a customer may be looking for
	pName	Contains the name of the plant
	Quantity	Shows the number of plants available for a specific specimen
	GM	Shows the percentage of genetical modification done on the plant
	Price	Show the cost of purchase of a plant
	soldAmount	The number of units sold of a specific plant

Entity	Attributes	Description
Employees	EmpID (primary key)	Can uniquely identify an employee for the farm
	EmpName	Contains the name of the employee
	EmpSalary	Contains the employee’s salary
	EmpPhoneNumber,	Contains the mobile/telephone number of the employee
	hiredDate,	Shows the date of when the employee was hired

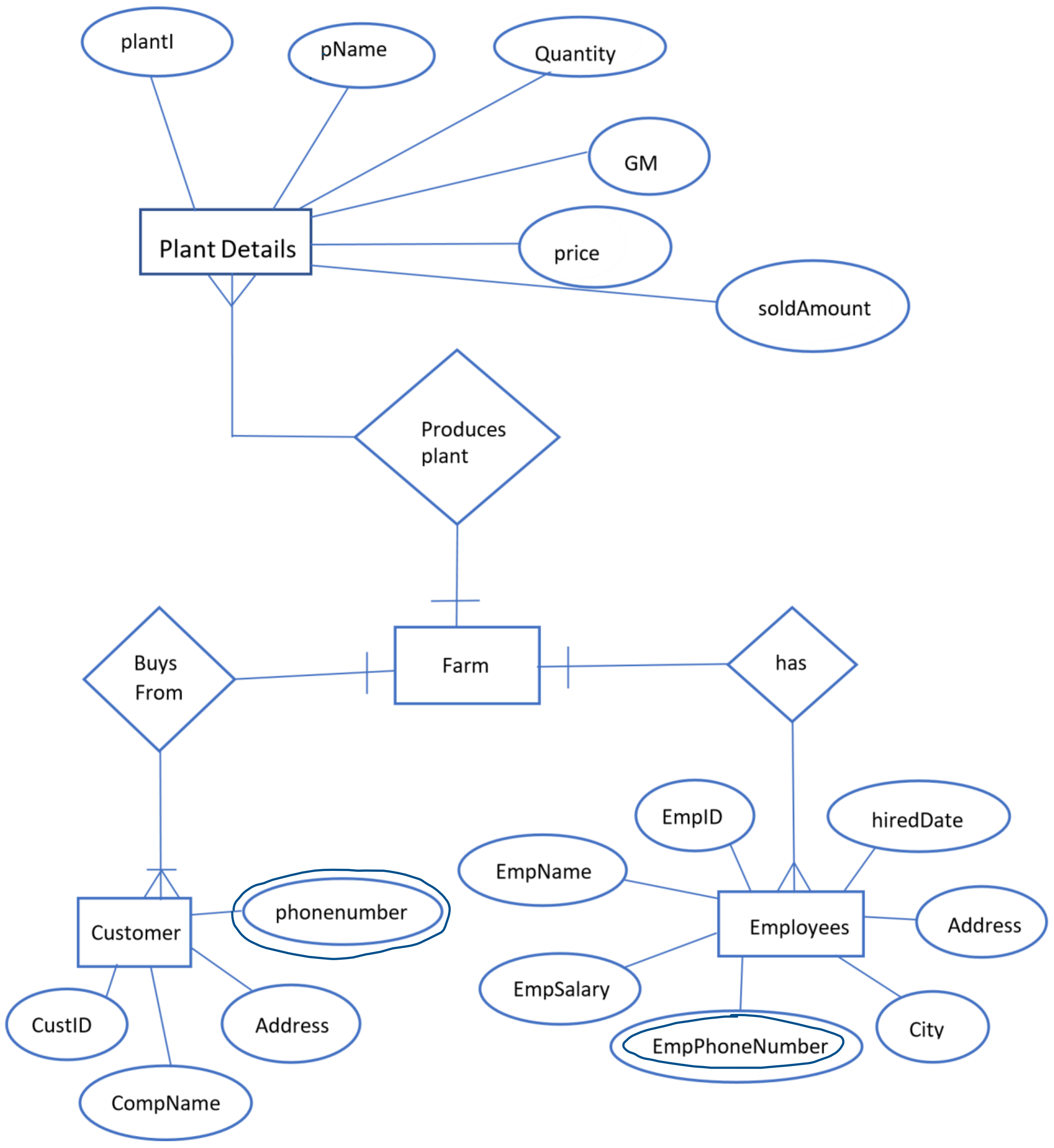
	Emptype,	The employee’s specific job in the farm
	City	Shows the city where the employee is from
	Address	Contains the location of the employee

Entity	Attributes	Description
Customer	CustID (primary key)	Contains a value to uniquely identify each customer (company in this case)
	CompName	Contains the name of the company
	Address	Shows the location of the company
	phoneNumber	Contains contact number of the company

Entity	Attributes	Description
Farm	BatchNo (primary key)	Contains the number of current batches of production and is used to uniquely identify each record involving plants, employees and customers
	CustID (foreign key)	Contains a value to uniquely identify each customer (company in this case)
	EmpID (foreign key)	Can uniquely identify an employee for the farm

	plantID (foreign key)	Uniquely identifies a plant in the farm which a customer may be looking for

2.ER-Diagram



3.Normalization:

Pre-normalized :

Entity: Plant detail

Attributes: plantID (primary key), pName, Quantity, GM, price, soldAmount

1NF- No multi-valued attributes

2NF- Partial Dependency found

Entities
Sales (SalesID(primary key), Quantity, GM, price, soldAmount)

Plant detail (plantID(primary key), plantName, SaleID (foreign key)) 3NF-

No transitive dependency present

Entity: Customer

Attributes: CustID, compName, Address, phonenumber

1NF- phonenumber is multivalued

Resolved by restricting it to contain only one number as
'telephoneNumber'

2NF- Partial dependency found

Resolved entities

Customer (CustID (primary key), CusEmail (foreign key) ,CompName)

Customer Detail (CusEmail (primary key), CompName, Address, telephoneNumber)

3NF- No transitive dependency present

Entity: Employees

Attributes: EmpID (primary key), EmpName, EmpSalary, EmpPhoneNumber, hiredDate, Emptype, City, Address

1NF- EmpPhoneNumber is multivalued

Attribute is then separated into 'EmpNumber' and 'EmergencyNumber'

Employees (EmpID (primary key), EmpName, EmpSalary, EmpNumber, 'EmergencyNumber', hiredDate, Emptype, City, Address)

2NF- Partial Dependency found

Resolved Entities

Employees (EmpID(primary key), WorkID (foreign key), EmpName, EmpNumber)

Employee Detail (WorkID(primary key), EmpSalary, EmergencyNumber, hiredDate, Emptype, City, Address)

3NF- Transitive Dependency found

Resolved entities

(Attribute: City is omitted)

Employee Detail (WorkID(primary key), EmpSalary, EmergencyNumber, hiredDate, Emptype, Address)

Entity: Farm

Attribute: BatchNo (primary key), CustID (foreign key), EmpID (foreign key), plantID (foreign key)

1NF- No multivalued attribute

2NF- No partial dependency found

3NF- No transitive dependency found

Post-Normalization:

After normalization, total table number increases to 7

Color Codes
Entities
Primary Key
Foreign Key

1. **Entity:** Farm

   Attributes: BatchNo, CustID, EmpID,

   plantID

2. **Entity:** Plant detail

   Attributes: plantID, plantName, SaleID

3. **Entity:** Sales

   Attributes: SalesID, Quantity, GM, price, soldAmount

4. **Entity:** Employees

   Attributes: EmpID, WorkID, EmpName, EmpNumber

5. **Entity:** Employee detail

   Attributes: WorkID, EmpSalary, EmergencyNumber, hiredDate,

Emptype, Address

6. **Entity:** Customer

   Attributes: CustID, CusEmail ,CompName

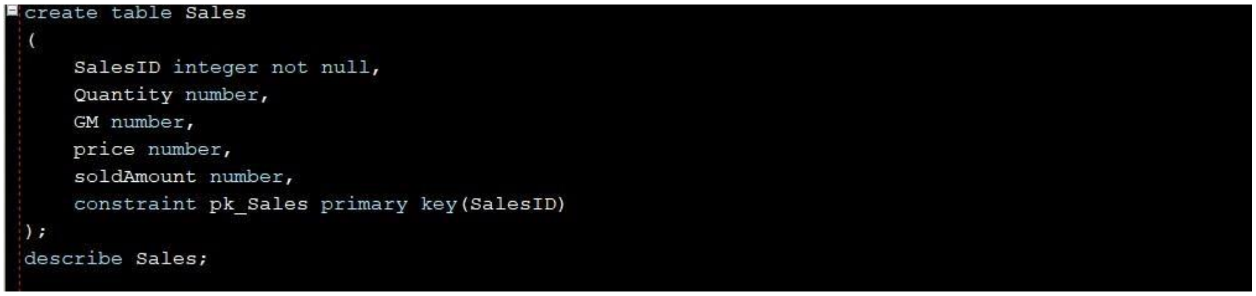
7. **Entity:** Customer Detail

   Attributes: CusEmail, CompName, Address, telephoneNumber

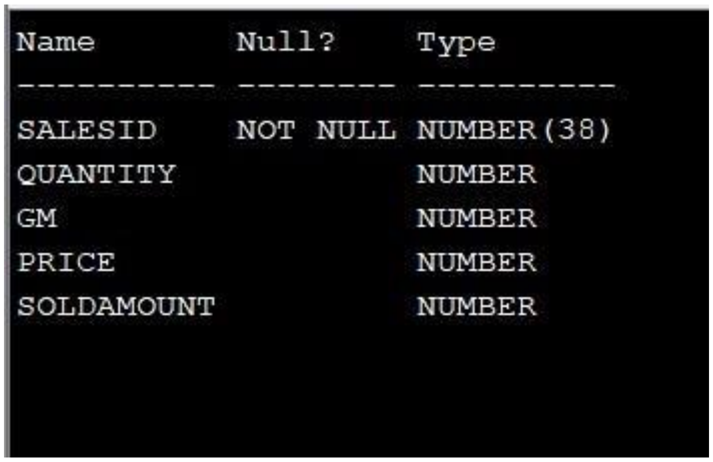
4.Code conversion:

Sales table:

Creation:

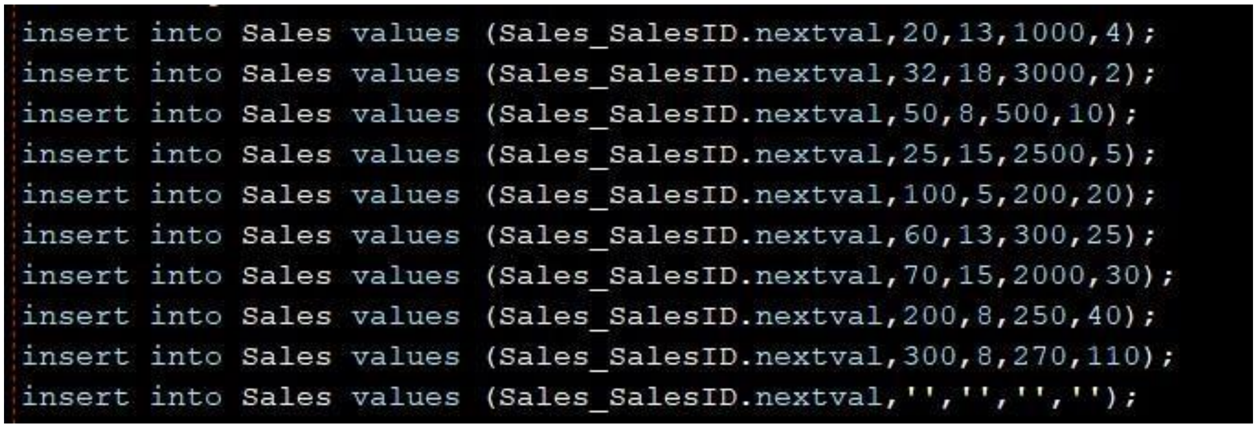


```
create table Sales
(
    SalesID integer not null,
    Quantity number,
    GM number,
    price number,
    soldAmount number,
    constraint pk_Sales primary key(SalesID)
);
describe Sales;
```

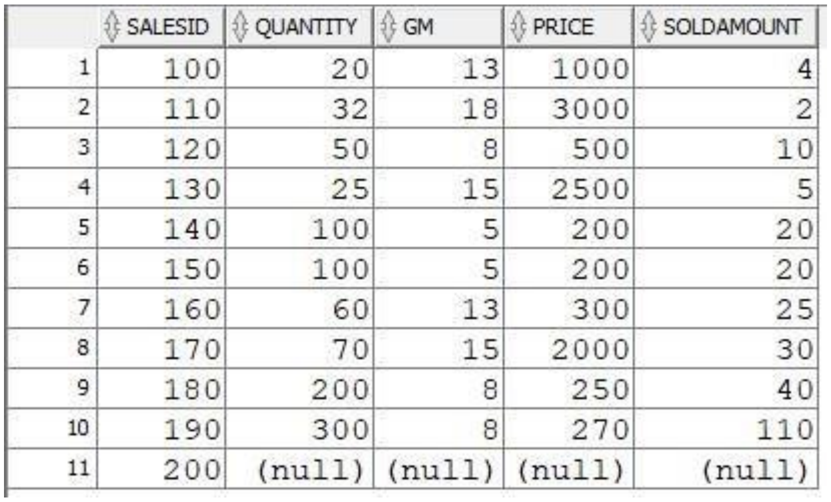


```
Name       Null?    Type
---------- -------- ----------
SALESID    NOT NULL NUMBER(38)
QUANTITY            NUMBER
GM                  NUMBER
PRICE               NUMBER
SOLDAMOUNT          NUMBER
```

Insertion:



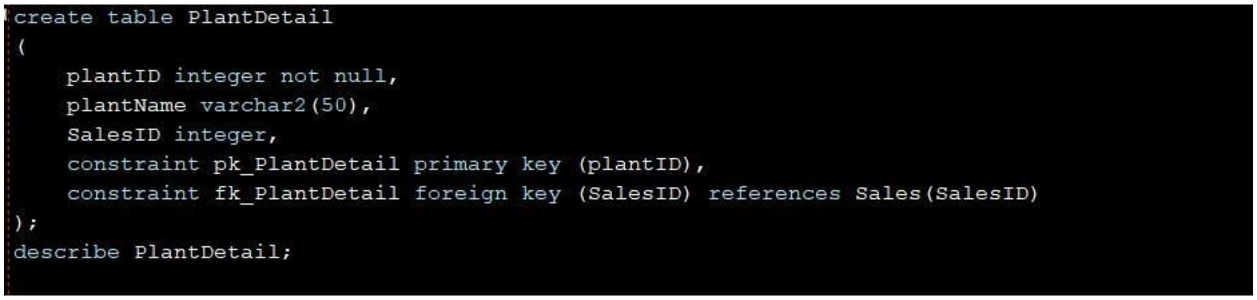
```
insert into Sales values (Sales_SalesID.nextval,20,13,1000,4);
insert into Sales values (Sales_SalesID.nextval,32,18,3000,2);
insert into Sales values (Sales_SalesID.nextval,50,8,500,10);
insert into Sales values (Sales_SalesID.nextval,25,15,2500,5);
insert into Sales values (Sales_SalesID.nextval,100,5,200,20);
insert into Sales values (Sales_SalesID.nextval,60,13,300,25);
insert into Sales values (Sales_SalesID.nextval,70,15,2000,30);
insert into Sales values (Sales_SalesID.nextval,200,8,250,40);
insert into Sales values (Sales_SalesID.nextval,300,8,270,110);
insert into Sales values (Sales_SalesID.nextval,'','','','');
```



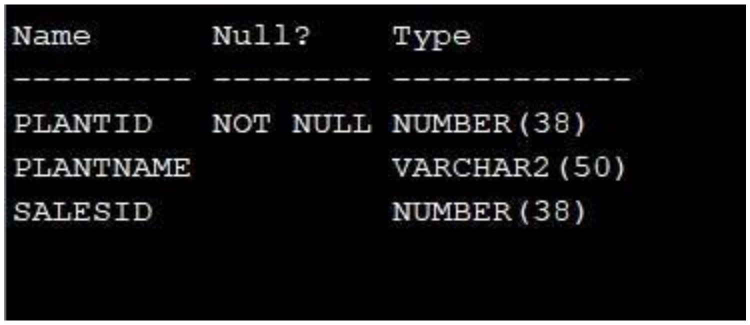
	SALESID	QUANTITY	GM	PRICE	SOLDAMOUNT
1	100	20	13	1000	4
2	110	32	18	3000	2
3	120	50	8	500	10
4	130	25	15	2500	5
5	140	100	5	200	20
6	150	100	5	200	20
7	160	60	13	300	25
8	170	70	15	2000	30
9	180	200	8	250	40
10	190	300	8	270	110
11	200	(null)	(null)	(null)	(null)

Plant Detail table:

Creation:

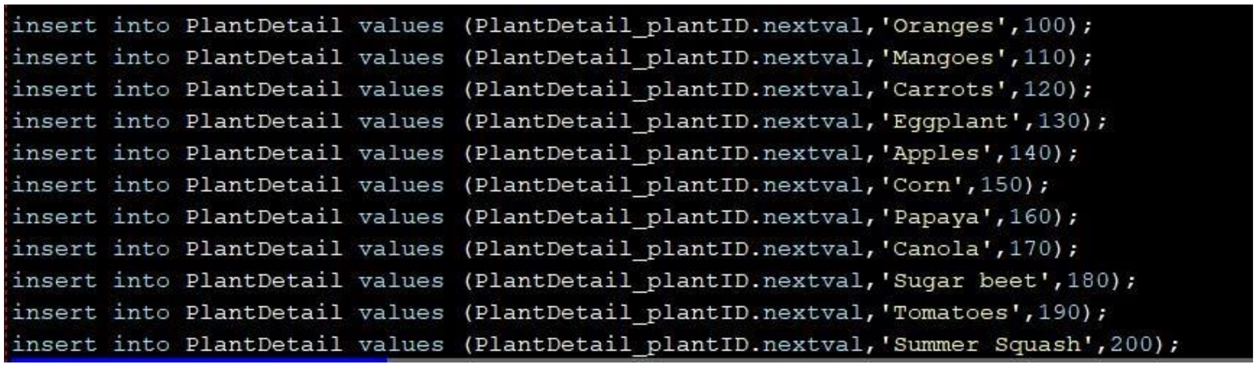


```
create table PlantDetail
(
    plantID integer not null,
    plantName varchar2(50),
    SalesID integer,
    constraint pk_PlantDetail primary key (plantID),
    constraint fk_PlantDetail foreign key (SalesID) references Sales(SalesID)
);
describe PlantDetail;
```

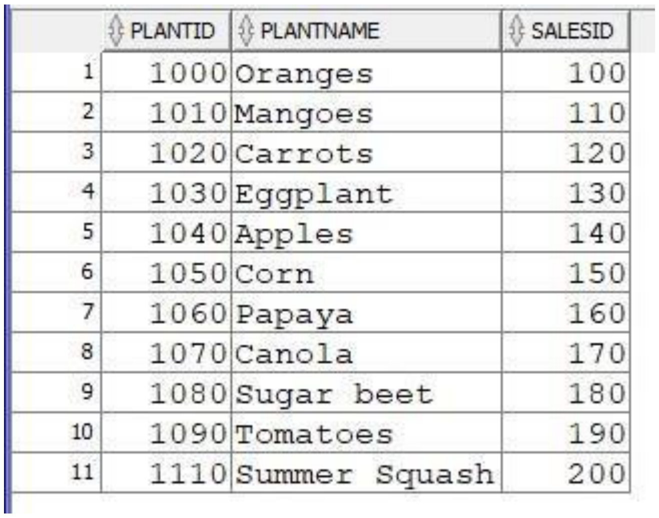


```
Name       Null?    Type
---------- -------- ------------
PLANTID    NOT NULL NUMBER(38)
PLANTNAME           VARCHAR2(50)
SALESID             NUMBER(38)
```

Insertion:



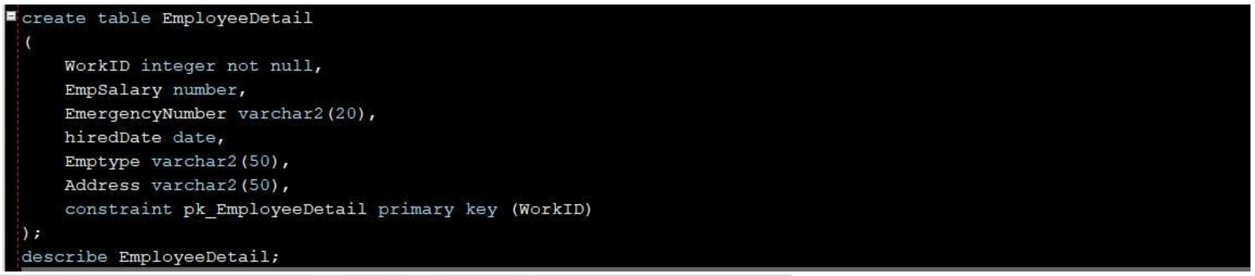
```
insert into PlantDetail values (PlantDetail_plantID.nextval,'Oranges',100);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Mangoes',110);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Carrots',120);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Eggplant',130);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Apples',140);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Corn',150);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Papaya',160);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Canola',170);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Sugar beet',180);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Tomatoes',190);
insert into PlantDetail values (PlantDetail_plantID.nextval,'Summer Squash',200);
```



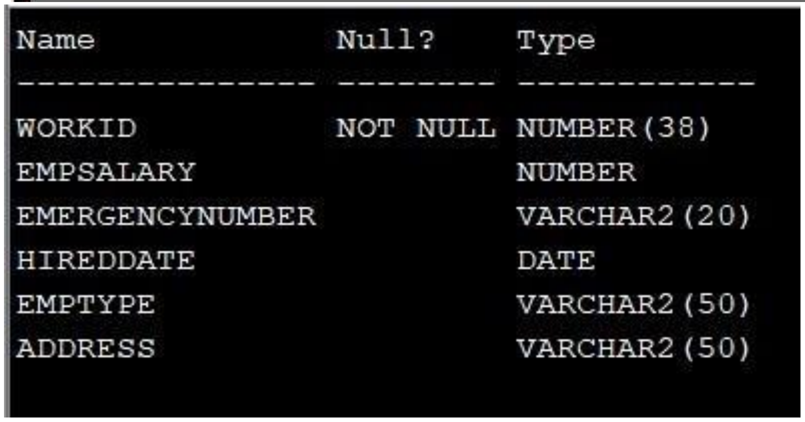
	PLANTID	PLANTNAME	SALESID
1	1000	Oranges	100
2	1010	Mangoes	110
3	1020	Carrots	120
4	1030	Eggplant	130
5	1040	Apples	140
6	1050	Corn	150
7	1060	Papaya	160
8	1070	Canola	170
9	1080	Sugar beet	180
10	1090	Tomatoes	190
11	1110	Summer Squash	200

Employees Detail table:

Creation:

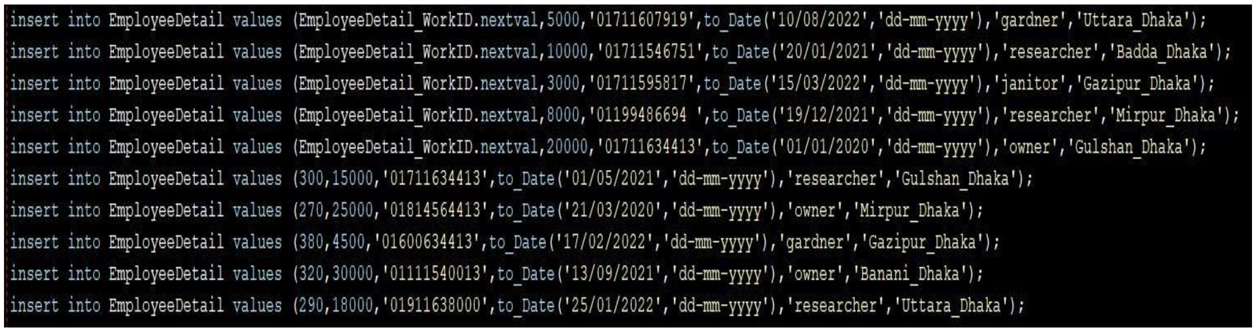


```
create table EmployeeDetail
(
    WorkID integer not null,
    EmpSalary number,
    EmergencyNumber varchar2(20),
    hiredDate date,
    Emptype varchar2(50),
    Address varchar2(50),
    constraint pk_EmployeeDetail primary key (WorkID)
);
describe EmployeeDetail;
```

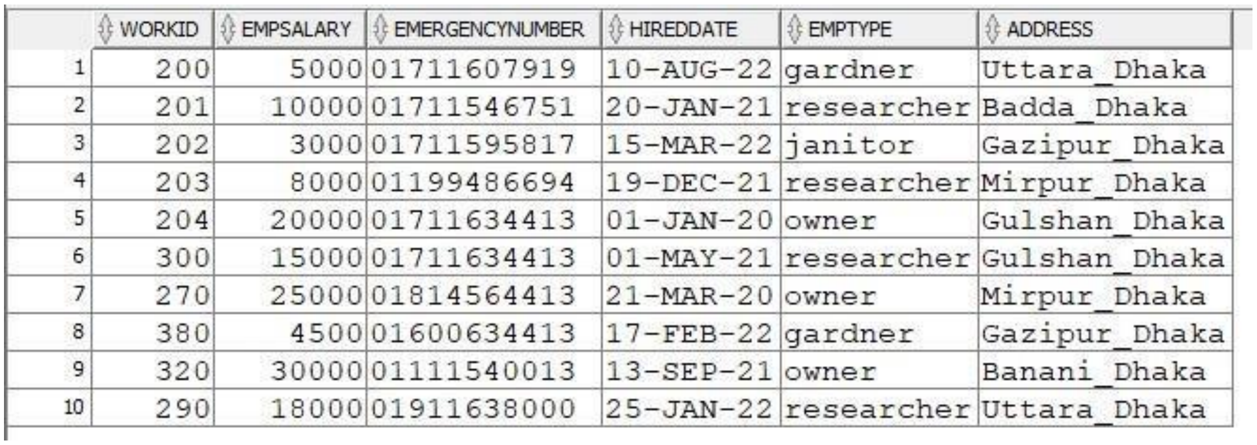


```
Name            Null?    Type
--------------- -------- ------------
WORKID          NOT NULL NUMBER(38)
EMPSALARY                NUMBER
EMERGENCYNUMBER          VARCHAR2(20)
HIREDDATE                DATE
EMPTYPE                  VARCHAR2(50)
ADDRESS                  VARCHAR2(50)
```

Insertion:



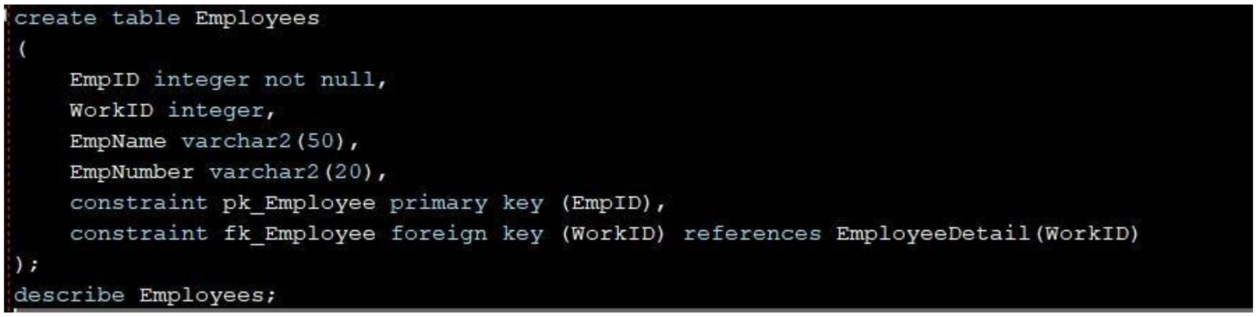
```
insert into EmployeeDetail values (EmployeeDetail_WorkID.nextval,5000,'01711607919',to_Date('10/08/2022','dd-mm-yyyy'),'gardner','Uttara_Dhaka');
insert into EmployeeDetail values (EmployeeDetail_WorkID.nextval,10000,'01711546751',to_Date('20/01/2021','dd-mm-yyyy'),'researcher','Badda_Dhaka');
insert into EmployeeDetail values (EmployeeDetail_WorkID.nextval,3000,'01711595817',to_Date('15/03/2022','dd-mm-yyyy'),'janitor','Gazipur_Dhaka');
insert into EmployeeDetail values (EmployeeDetail_WorkID.nextval,8000,'01199486694 ',to_Date('19/12/2021','dd-mm-yyyy'),'researcher','Mirpur_Dhaka');
insert into EmployeeDetail values (EmployeeDetail_WorkID.nextval,20000,'01711634413',to_Date('01/01/2020','dd-mm-yyyy'),'owner','Gulshan_Dhaka');
insert into EmployeeDetail values (300,15000,'01711634413',to_Date('01/05/2021','dd-mm-yyyy'),'researcher','Gulshan_Dhaka');
insert into EmployeeDetail values (270,25000,'01814564413',to_Date('21/03/2020','dd-mm-yyyy'),'owner','Mirpur_Dhaka');
insert into EmployeeDetail values (380,4500,'01600634413',to_Date('17/02/2022','dd-mm-yyyy'),'gardner','Gazipur_Dhaka');
insert into EmployeeDetail values (320,30000,'01111540013',to_Date('13/09/2021','dd-mm-yyyy'),'owner','Banani_Dhaka');
insert into EmployeeDetail values (290,18000,'01911638000',to_Date('25/01/2022','dd-mm-yyyy'),'researcher','Uttara_Dhaka');
```



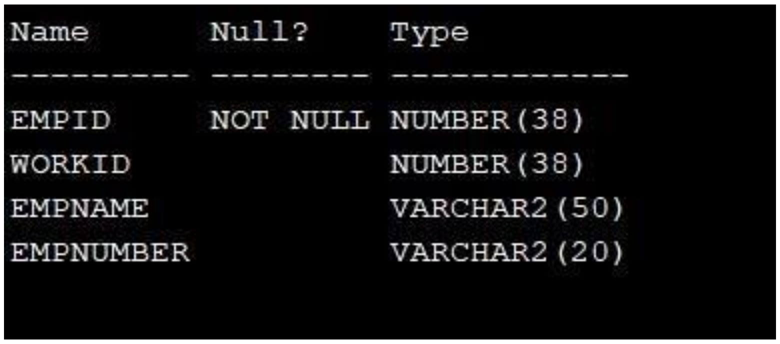
	WORKID	EMPSALARY	EMERGENCYNUMBER	HIREDDATE	EMPTYPE	ADDRESS
1	200	5000	01711607919	10-AUG-22	gardner	Uttara_Dhaka
2	201	10000	01711546751	20-JAN-21	researcher	Badda_Dhaka
3	202	3000	01711595817	15-MAR-22	janitor	Gazipur_Dhaka
4	203	8000	01199486694	19-DEC-21	researcher	Mirpur_Dhaka
5	204	20000	01711634413	01-JAN-20	owner	Gulshan_Dhaka
6	300	15000	01711634413	01-MAY-21	researcher	Gulshan_Dhaka
7	270	25000	01814564413	21-MAR-20	owner	Mirpur_Dhaka
8	380	4500	01600634413	17-FEB-22	gardner	Gazipur_Dhaka
9	320	30000	01111540013	13-SEP-21	owner	Banani_Dhaka
10	290	18000	01911638000	25-JAN-22	researcher	Uttara_Dhaka

Employees Table:

Creation:

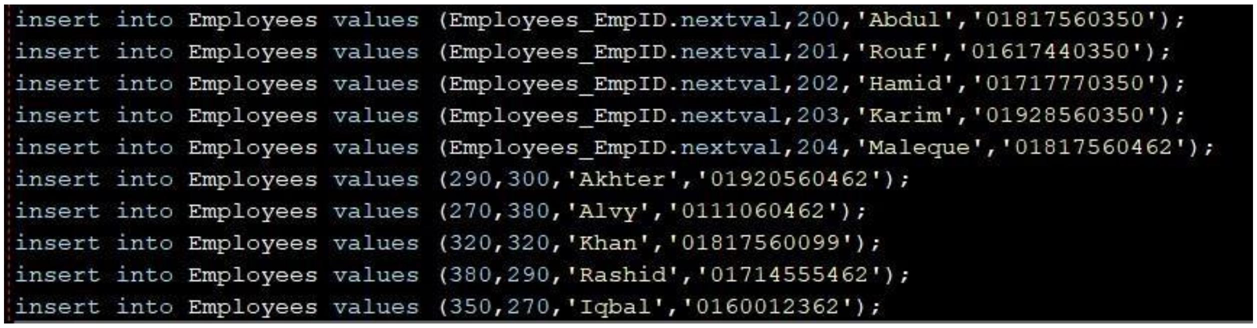


```
create table Employees
(
    EmpID integer not null,
    WorkID integer,
    EmpName varchar2(50),
    EmpNumber varchar2(20),
    constraint pk_Employee primary key (EmpID),
    constraint fk_Employee foreign key (WorkID) references EmployeeDetail(WorkID)
);
describe Employees;
```

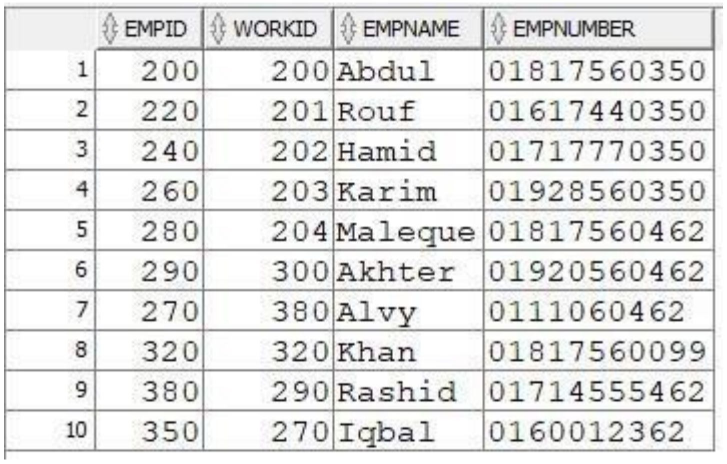


```
Name      Null?    Type
--------- -------- ------------
EMPID     NOT NULL NUMBER(38)
WORKID             NUMBER(38)
EMPNAME            VARCHAR2(50)
EMPNUMBER          VARCHAR2(20)
```

Insertion:



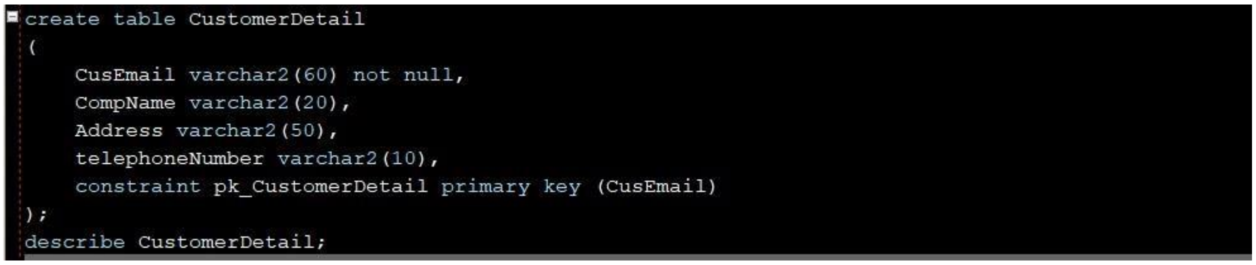
```
insert into Employees values (Employees_EmpID.nextval,200,'Abdul','01817560350');
insert into Employees values (Employees_EmpID.nextval,201,'Rouf','01617440350');
insert into Employees values (Employees_EmpID.nextval,202,'Hamid','01717770350');
insert into Employees values (Employees_EmpID.nextval,203,'Karim','01928560350');
insert into Employees values (Employees_EmpID.nextval,204,'Maleque','01817560462');
insert into Employees values (290,300,'Akhter','01920560462');
insert into Employees values (270,380,'Alvy','0111060462');
insert into Employees values (320,320,'Khan','01817560099');
insert into Employees values (380,290,'Rashid','01714555462');
insert into Employees values (350,270,'Iqbal','0160012362');
```



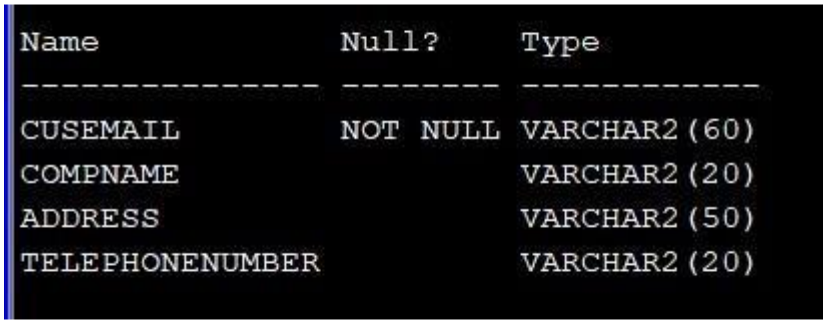
	EMPID	WORKID	EMPNAME	EMPNUMBER
1	200	200	Abdul	01817560350
2	220	201	Rouf	01617440350
3	240	202	Hamid	01717770350
4	260	203	Karim	01928560350
5	280	204	Maleque	01817560462
6	290	300	Akhter	01920560462
7	270	380	Alvy	0111060462
8	320	320	Khan	01817560099
9	380	290	Rashid	01714555462
10	350	270	Iqbal	0160012362

Customer Detail table:

Creation:

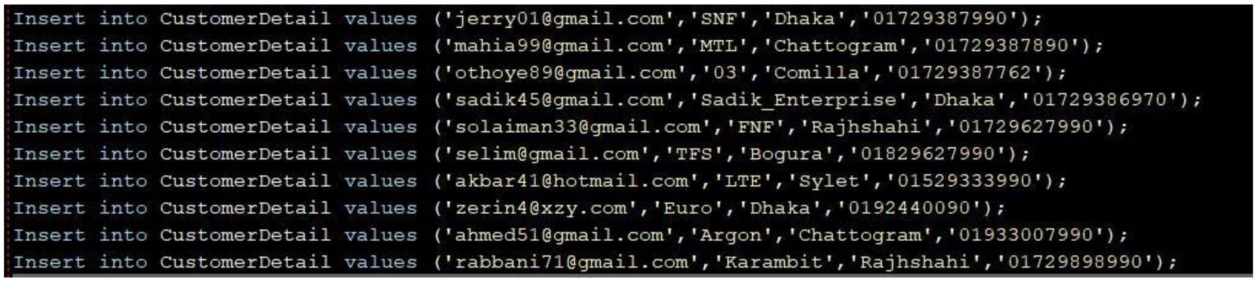


```
create table CustomerDetail
(
    CusEmail varchar2(60) not null,
    CompName varchar2(20),
    Address varchar2(50),
    telephoneNumber varchar2(10),
    constraint pk_CustomerDetail primary key (CusEmail)
);
describe CustomerDetail;
```

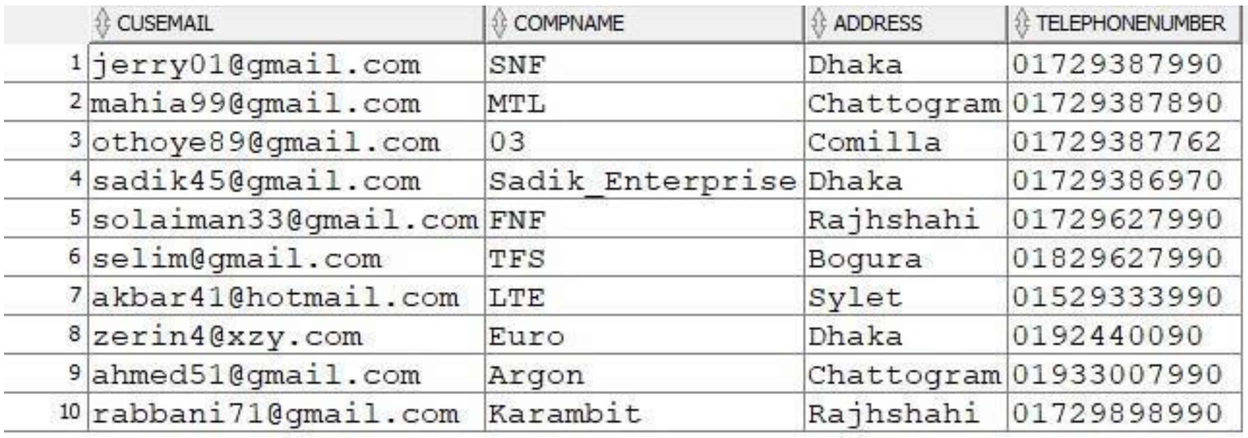


```
Name            Null?    Type
--------------- -------- ------------
CUSEMAIL        NOT NULL VARCHAR2(60)
COMPNAME                 VARCHAR2(20)
ADDRESS                  VARCHAR2(50)
TELEPHONENUMBER          VARCHAR2(20)
```

Insertion:



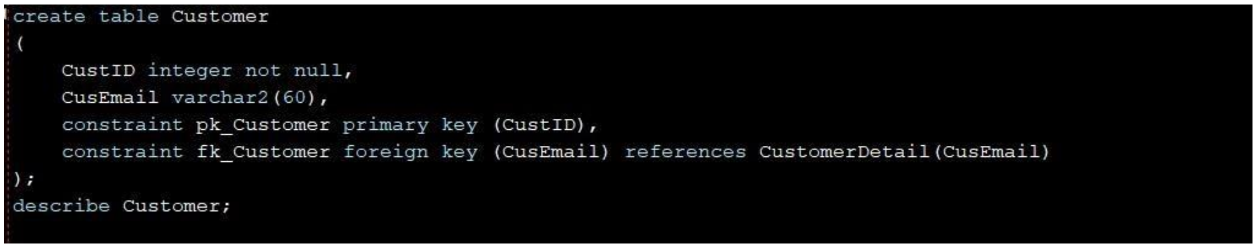
```
Insert into CustomerDetail values ('jerry01@gmail.com','SNF','Dhaka','01729387990');
Insert into CustomerDetail values ('mahia99@gmail.com','MTL','Chattogram','01729387890');
Insert into CustomerDetail values ('othoye89@gmail.com','03','Comilla','01729387762');
Insert into CustomerDetail values ('sadik45@gmail.com','Sadik_Enterprise','Dhaka','01729386970');
Insert into CustomerDetail values ('solaiman33@gmail.com','FNF','Rajhshahi','01729627990');
Insert into CustomerDetail values ('selim@gmail.com','TFS','Bogura','01829627990');
Insert into CustomerDetail values ('akbar41@hotmail.com','LTE','Sylet','01529333990');
Insert into CustomerDetail values ('zerin4@xzy.com','Euro','Dhaka','0192440090');
Insert into CustomerDetail values ('ahmed51@gmail.com','Argon','Chattogram','01933007990');
Insert into CustomerDetail values ('rabbani71@gmail.com','Karambit','Rajhshahi','01729898990');
```



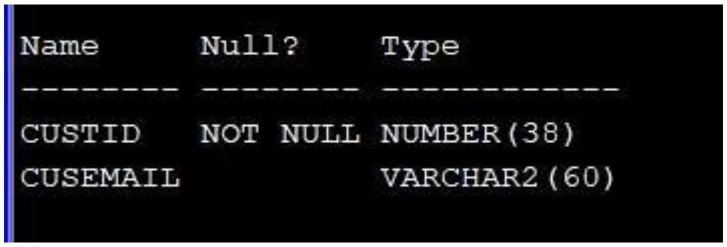
	CUSEMAIL	COMPNAME	ADDRESS	TELEPHONENUMBER
1	jerry01@gmail.com	SNF	Dhaka	01729387990
2	mahia99@gmail.com	MTL	Chattogram	01729387890
3	othoye89@gmail.com	03	Comilla	01729387762
4	sadik45@gmail.com	Sadik_Enterprise	Dhaka	01729386970
5	solaiman33@gmail.com	FNF	Rajhshahi	01729627990
6	selim@gmail.com	TFS	Bogura	01829627990
7	akbar41@hotmail.com	LTE	Sylet	01529333990
8	zerin4@xzy.com	Euro	Dhaka	0192440090
9	ahmed51@gmail.com	Argon	Chattogram	01933007990
10	rabbani71@gmail.com	Karambit	Rajhshahi	01729898990

Customer table:

Creation:

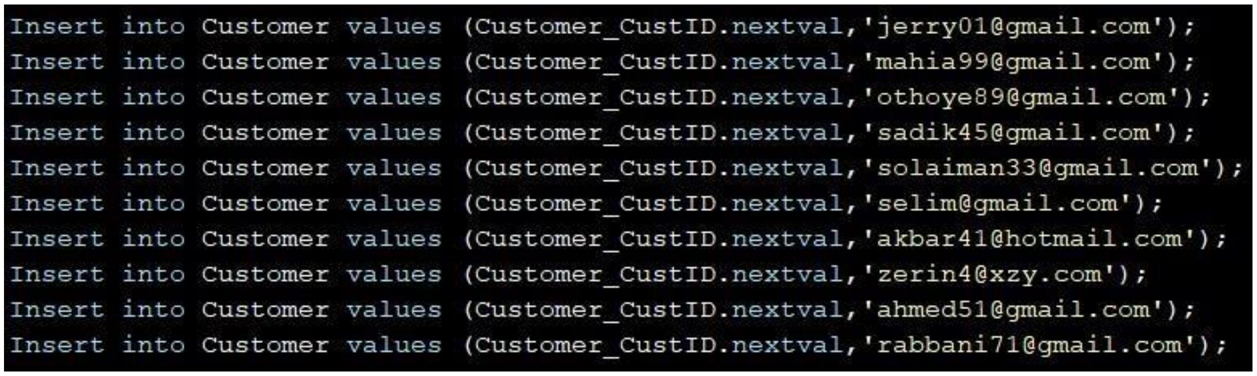


```
create table Customer
(
    CustID integer not null,
    CusEmail varchar2(60),
    constraint pk_Customer primary key (CustID),
    constraint fk_Customer foreign key (CusEmail) references CustomerDetail(CusEmail)
);
describe Customer;
```

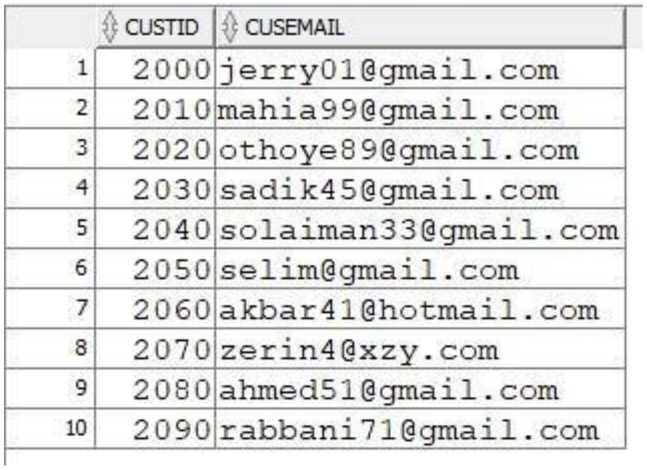


```
Name      Null?    Type
--------- -------- ------------
CUSTID    NOT NULL NUMBER(38)
CUSEMAIL           VARCHAR2(60)
```

Insertion:



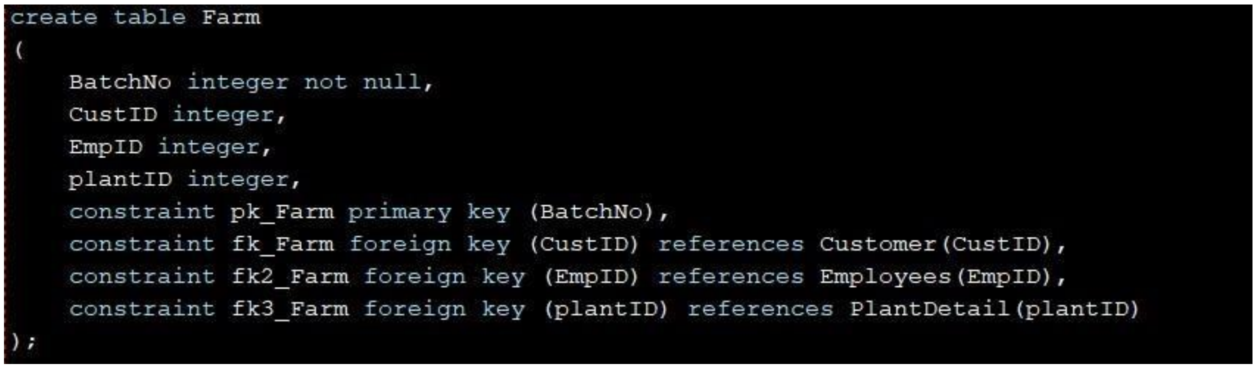
```
Insert into Customer values (Customer_CustID.nextval,'jerry01@gmail.com');
Insert into Customer values (Customer_CustID.nextval,'mahia99@gmail.com');
Insert into Customer values (Customer_CustID.nextval,'othoye89@gmail.com');
Insert into Customer values (Customer_CustID.nextval,'sadik45@gmail.com');
Insert into Customer values (Customer_CustID.nextval,'solaiman33@gmail.com');
Insert into Customer values (Customer_CustID.nextval,'selim@gmail.com');
Insert into Customer values (Customer_CustID.nextval,'akbar41@hotmail.com');
Insert into Customer values (Customer_CustID.nextval,'zerin4@xzy.com');
Insert into Customer values (Customer_CustID.nextval,'ahmed51@gmail.com');
Insert into Customer values (Customer_CustID.nextval,'rabbani71@gmail.com');
```



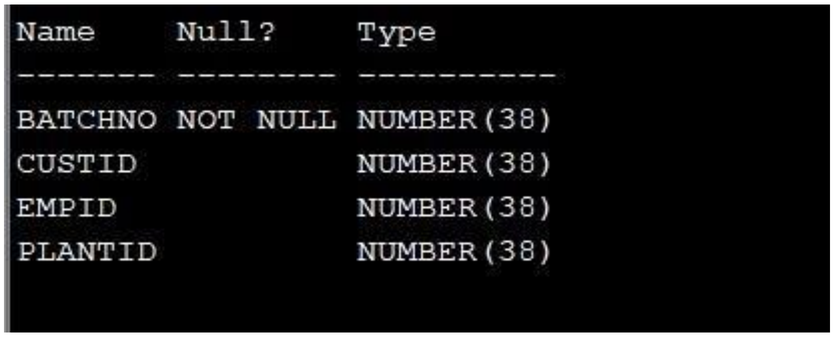
	CUSTID	CUSEMAIL
1	2000	jerry01@gmail.com
2	2010	mahia99@gmail.com
3	2020	othoye89@gmail.com
4	2030	sadik45@gmail.com
5	2040	solaiman33@gmail.com
6	2050	selim@gmail.com
7	2060	akbar41@hotmail.com
8	2070	zerin4@xzy.com
9	2080	ahmed51@gmail.com
10	2090	rabbani71@gmail.com

Farm table:

Creation:

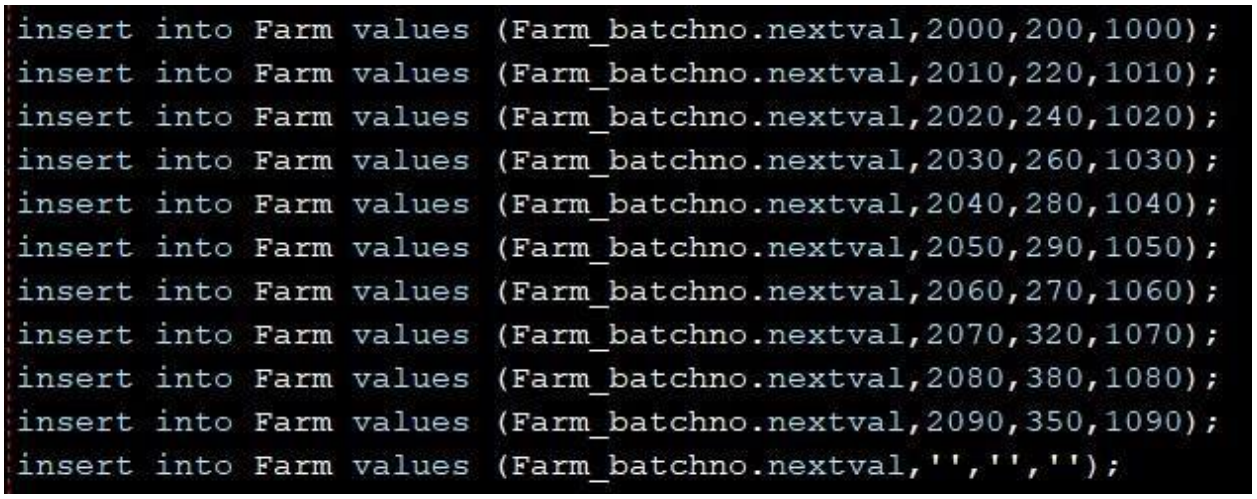


```
create table Farm
(
    BatchNo integer not null,
    CustID integer,
    EmpID integer,
    plantID integer,
    constraint pk_Farm primary key (BatchNo),
    constraint fk_Farm foreign key (CustID) references Customer(CustID),
    constraint fk2_Farm foreign key (EmpID) references Employees(EmpID),
    constraint fk3_Farm foreign key (plantID) references PlantDetail(plantID)
);
```

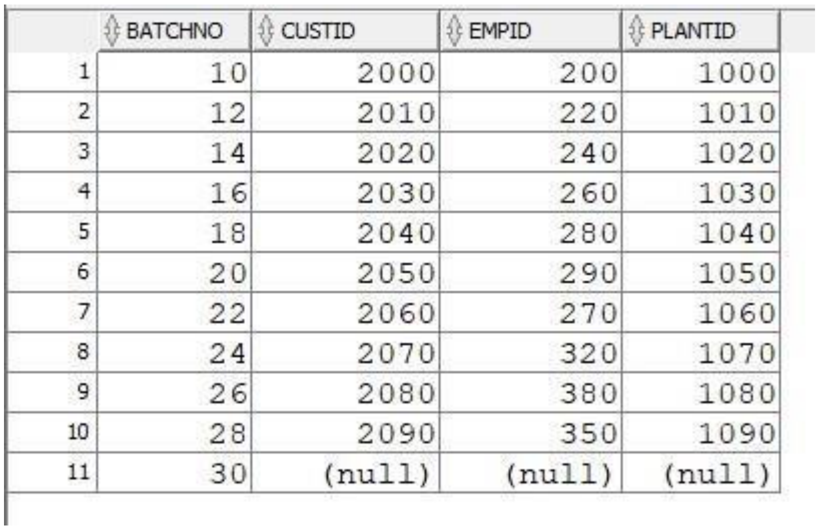


```
Name     Null?    Type
-------- -------- ----------
BATCHNO  NOT NULL NUMBER(38)
CUSTID            NUMBER(38)
EMPID             NUMBER(38)
PLANTID           NUMBER(38)
```

Insertion:



```
insert into Farm values (Farm_batchno.nextval,2000,200,1000);
insert into Farm values (Farm_batchno.nextval,2010,220,1010);
insert into Farm values (Farm_batchno.nextval,2020,240,1020);
insert into Farm values (Farm_batchno.nextval,2030,260,1030);
insert into Farm values (Farm_batchno.nextval,2040,280,1040);
insert into Farm values (Farm_batchno.nextval,2050,290,1050);
insert into Farm values (Farm_batchno.nextval,2060,270,1060);
insert into Farm values (Farm_batchno.nextval,2070,320,1070);
insert into Farm values (Farm_batchno.nextval,2080,380,1080);
insert into Farm values (Farm_batchno.nextval,2090,350,1090);
insert into Farm values (Farm_batchno.nextval,'','','');
```

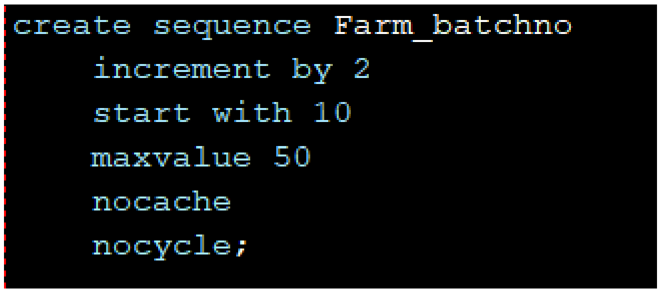


	BATCHNO	CUSTID	EMPID	PLANTID
1	10	2000	200	1000
2	12	2010	220	1010
3	14	2020	240	1020
4	16	2030	260	1030
5	18	2040	280	1040
6	20	2050	290	1050
7	22	2060	270	1060
8	24	2070	320	1070
9	26	2080	380	1080
10	28	2090	350	1090
11	30	(null)	(null)	(null)

5.Reporting Queries:

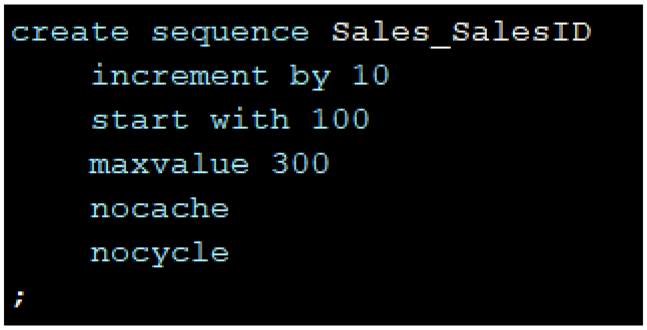
Creating Sequences:

For Farm Table:



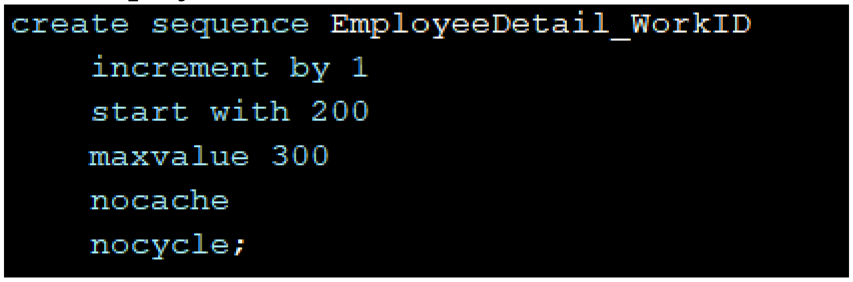
```
create sequence Farm_batchno
    increment by 2
    start with 10
    maxvalue 50
    nocache
    nocycle;
```

For Sales Table:



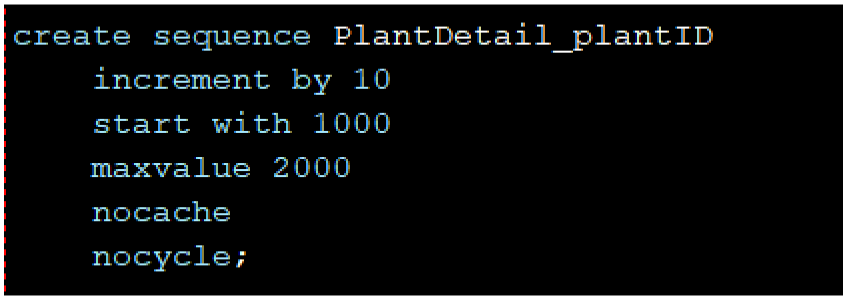
```
create sequence Sales_SalesID
    increment by 10
    start with 100
    maxvalue 300
    nocache
    nocycle
;
```

For Employee Detail Table:



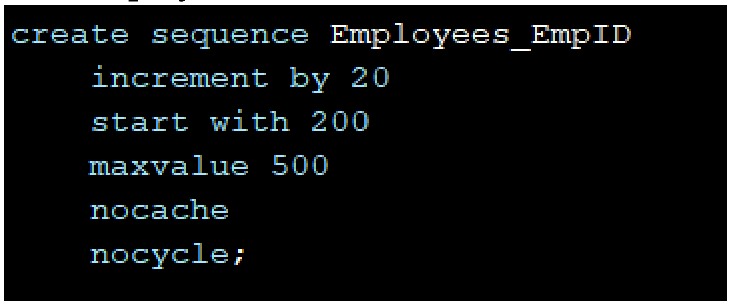
```
create sequence EmployeeDetail_WorkID
    increment by 1
    start with 200
    maxvalue 300
    nocache
    nocycle;
```

For Plant Detail Table:



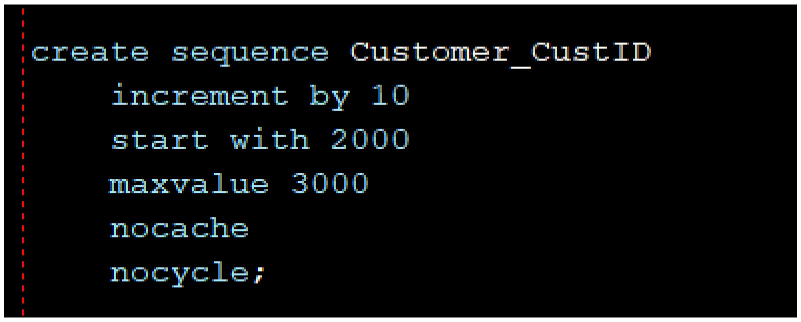
```
create sequence PlantDetail_plantID
    increment by 10
    start with 1000
    maxvalue 2000
    nocache
    nocycle;
```

For Employees Table:



```
create sequence Employees_EmpID
    increment by 20
    start with 200
    maxvalue 500
    nocache
    nocycle;
```

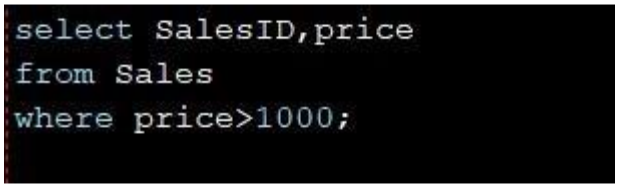
For Customer Table:



```
create sequence Customer_CustID
    increment by 10
    start with 2000
    maxvalue 3000
    nocache
    nocycle;
```

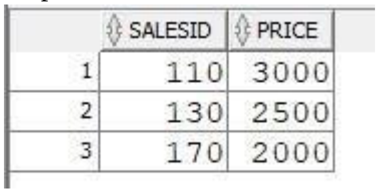
Simple queries:

1. **Finding the IDs of the plants which have a price over 1000 :**



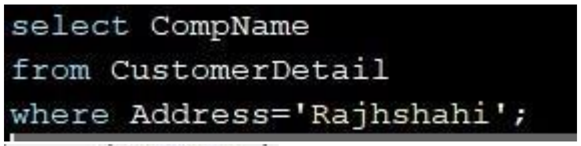
```
select SalesID,price
from Sales
where price>1000;
```

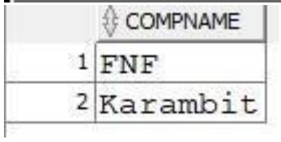
Output:



	SALESID	PRICE
1	110	3000
2	130	2500
3	170	2000

2. **Finding the names of companies who are based in ‘Rajshahi’ :**

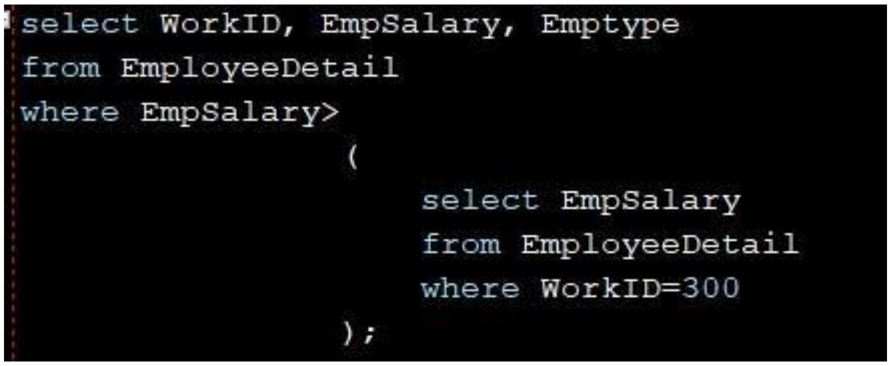

```
select CompName
from CustomerDetail
where Address='Rajhshahi';
```



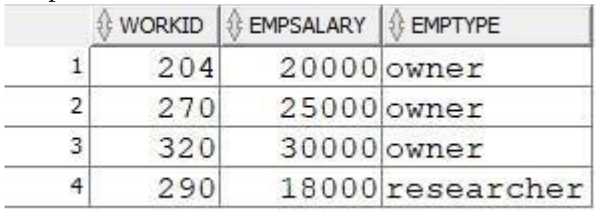
	COMPNAME
1	FNF
2	Karambit

Sub-queries:

1. **Find the WorkIDs, salary and jobs of employees working whose salary is greater than the employee with the who's WorkID is 300 :**

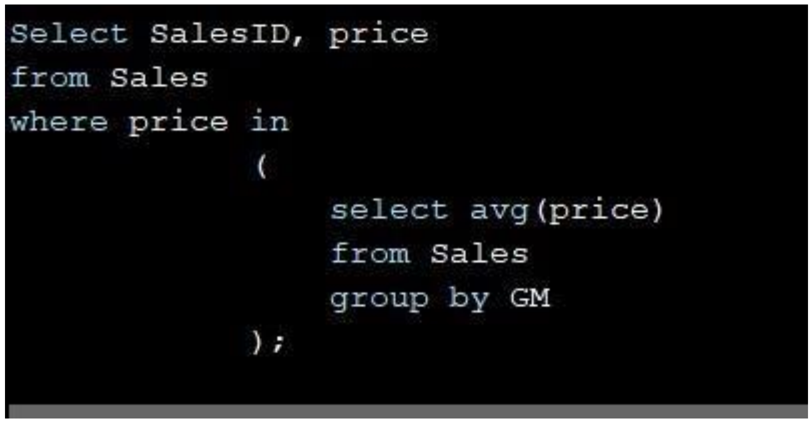

```
select WorkID, EmpSalary, Emptype
from EmployeeDetail
where EmpSalary>
                (
                    select EmpSalary
                    from EmployeeDetail
                    where WorkID=300
                );
```

Output:

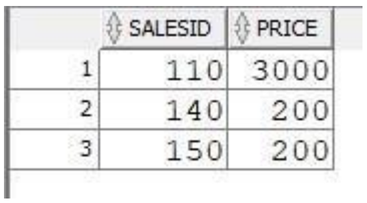


	WORKID	EMPSALARY	EMPTYPE
1	204	20000	owner
2	270	25000	owner
3	320	30000	owner
4	290	18000	researcher

2. **Find the IDs and prices of those plants who have a price similar to the average price of plants (according to GM percentages) in the farm :**


```
Select SalesID, price
from Sales
where price in
            (
                select avg(price)
                from Sales
                group by GM
            );
```

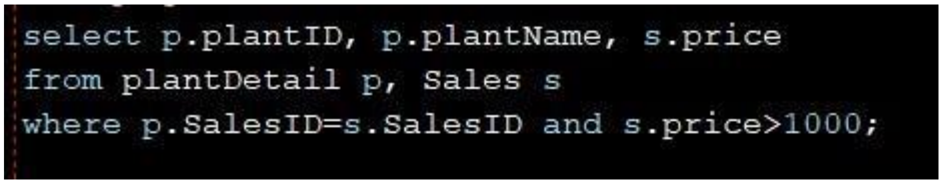
Output:



	SALESID	PRICE
1	110	3000
2	140	200
3	150	200

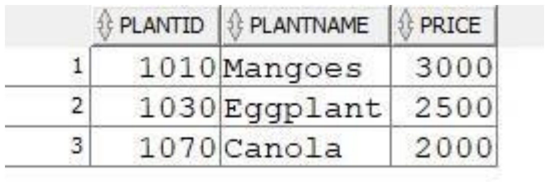
Equijoins:

Find the names and plantID of those plants who have a price above 1000 :



```
select p.plantID, p.plantName, s.price
from plantDetail p, Sales s
where p.SalesID=s.SalesID and s.price>1000;
```

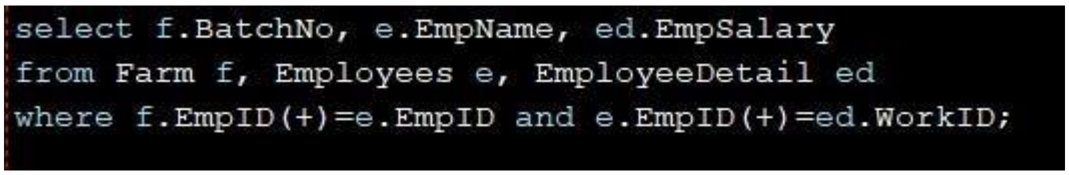
Output



	PLANTID	PLANTNAME	PRICE
1	1010	Mangoes	3000
2	1030	Eggplant	2500
3	1070	Canola	2000

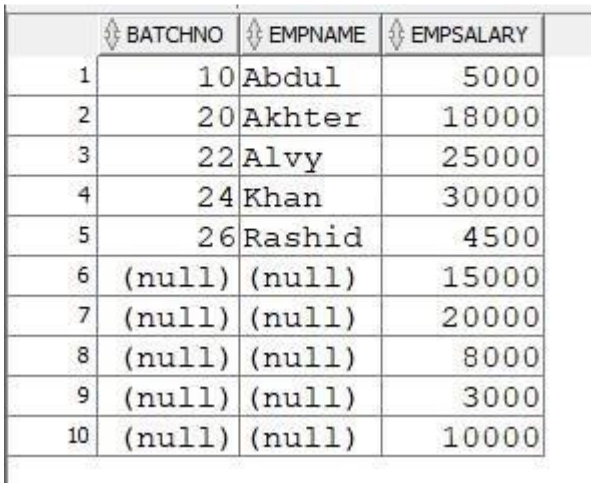
Outer-joins:

Find the batch no. and the names and salaries of the employees in the farm :



```
select f.BatchNo, e.EmpName, ed.EmpSalary
from Farm f, Employees e, EmployeeDetail ed
where f.EmpID(+)=e.EmpID and e.EmpID(+)=ed.WorkID;
```

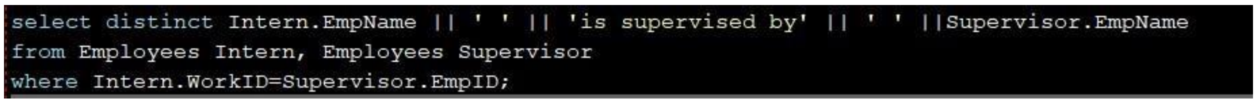
Output:



	BATCHNO	EMPNAME	EMPSALARY
1	10	Abdul	5000
2	20	Akhter	18000
3	22	Alvy	25000
4	24	Khan	30000
5	26	Rashid	4500
6	(null)	(null)	15000
7	(null)	(null)	20000
8	(null)	(null)	8000
9	(null)	(null)	3000
10	(null)	(null)	10000

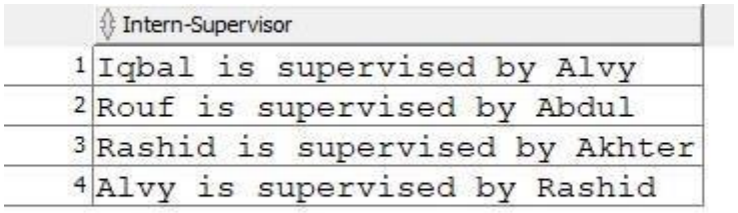
Self-joins:

Find all those employees and their names who are being supervised by another employee :



```
select distinct Intern.EmpName || ' ' || 'is supervised by' || ' ' ||Supervisor.EmpName
from Employees Intern, Employees Supervisor
where Intern.WorkID=Supervisor.EmpID;
```

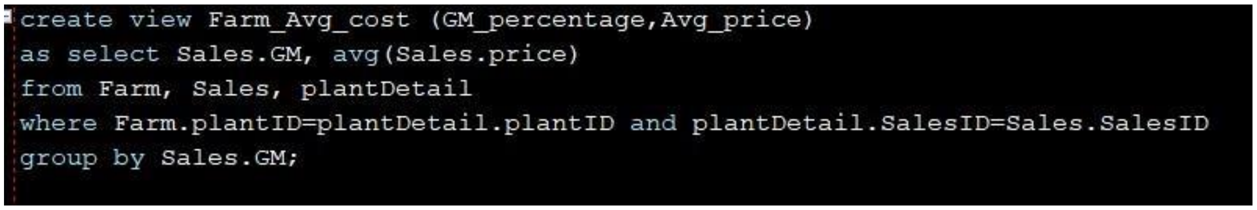
Output:



	Intern-Supervisor
1	Iqbal is supervised by Alvy
2	Rouf is supervised by Abdul
3	Rashid is supervised by Akhter
4	Alvy is supervised by Rashid

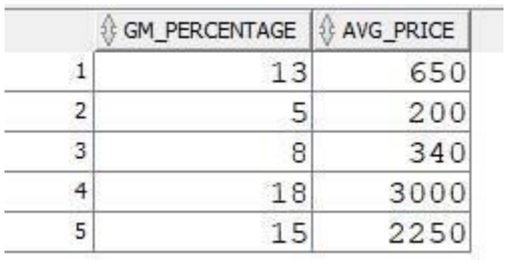
View:

Creating a view that contains the average cost of plants in the farm according to their GM percentages :



```
create view Farm_Avg_cost (GM_percentage,Avg_price)
as select Sales.GM, avg(Sales.price)
from Farm, Sales, plantDetail
where Farm.plantID=plantDetail.plantID and plantDetail.SalesID=Sales.SalesID
group by Sales.GM;
```

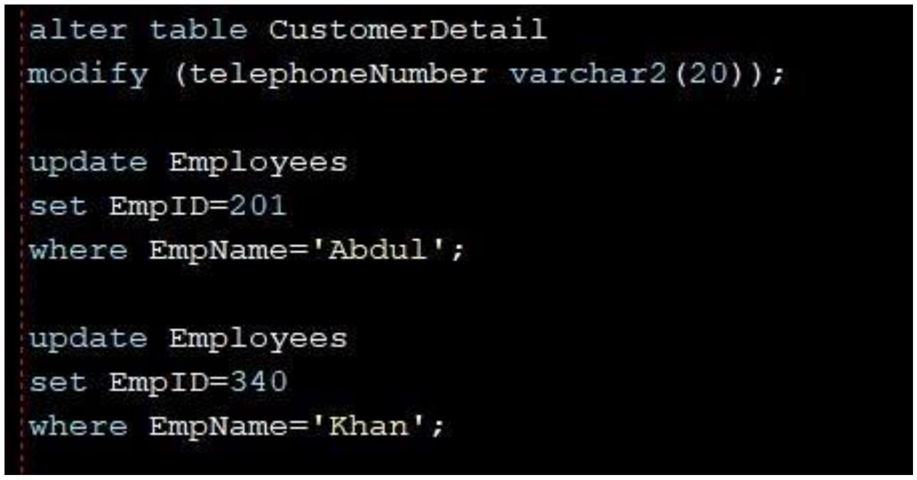
Output:



	GM_PERCENTAGE	AVG_PRICE
1	13	650
2	5	200
3	8	340
4	18	3000
5	15	2250

Updates:

Some alterations were done to find valid outputs from the database:



```
alter table CustomerDetail
modify (telephoneNumber varchar2(20));

update Employees
set EmpID=201
where EmpName='Abdul';

update Employees
set EmpID=340
where EmpName='Khan';
```

Thank you