A picture containing calendar

Description automatically generated

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

**DEPARTMENT OF Computer Science and Engineering**

##### INTRODUCTION TO DATABASE

**Section: B**

##### Project: Superstore Management System

**Supervised By**

**JUENA AHMED NOSHIN**

**Submitted By**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Contributions** |
| **Sakib Al Mahamud** | **21-45655-3** | **Scenario Description, Table Creation, Conclusion** |
| **Nimur Islam Joy** | **21-45656-3** | **ER Diagram, Query Writing** |
| **Istyak Ahmed** | **21-45658-3** | **Normalization, Schema Diagram, Introduction** |
| **Md.Sakibul Haque Tanmoy** | **21-45659-3** | **Data Insertion, Relational Algebra** |

**Contents**

**Page no.**

**1. Introduction** ---------------------------------------------------------------------- 03

**2. Scenario Description** ----------------------------------------------------------- 03

3. **ER Diagram** ----------------------------------------------------------------------- 04

**4. Normalization** -------------------------------------------------------------------- 05

**5. Schema Diagram** ---------------------------------------------------------------- 13

**6. Table Creation** ------------------------------------------------------------------- 14

**7. Data Insertion** -------------------------------------------------------------------- 26

**8. Query Writing** -------------------------------------------------------------------- 35

**9. Relational Algebra** --------------------------------------------------------------- 42

**10. Conclusion** ------------------------------------------------------------------------ 43

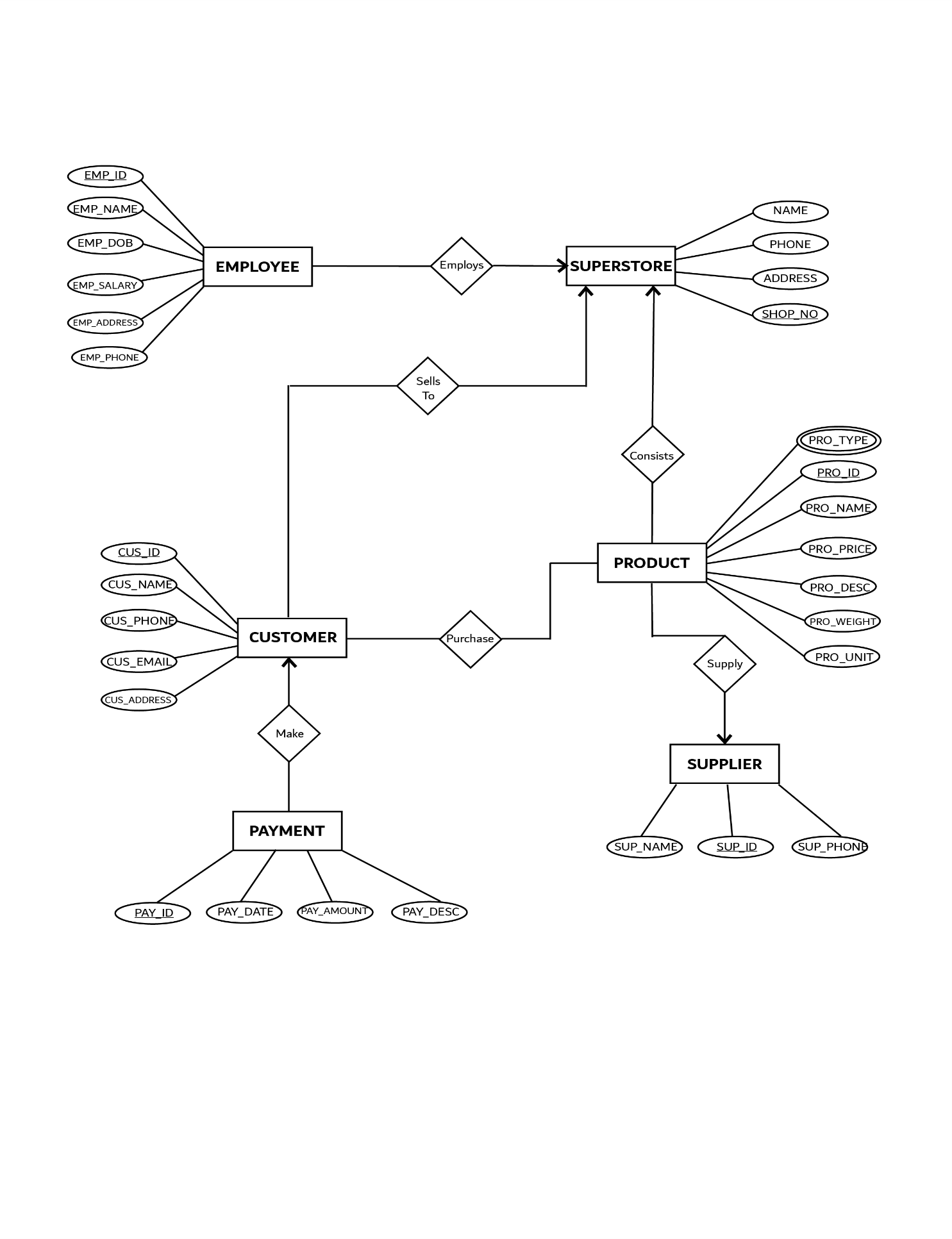
**Introduction:**

Detailed SQL queries of our project start after the system’s ER Diagram is completed. This phase aims to develop the internal relationship between each of the entities identified during the designing of the ER Diagram. In the ER Diagram, a more detailed specification is given by explaining the types of information in attributes which entities are supposed to carry.

**Scenario Description:**

In a Superstore management system, a superstore sells various products to the customers who visit the superstore to buy their needs. A superstore has its unique shop no. It also has the name, phone, and address. A superstore has suppliers who are identical by their supplier id. A supplier also has more attributes, such as name and phone number. One specific supplier supplies just one particular product to the store. Each product has its unique id and type. They also have their name, price, description, weight, and unit. Customers are identical to their customer ID and have their usual name, phone, email, and geographical address. The customer makes a payment to purchase any product. Each payment gets recorded in the store database through their unique id and a few more details like Payment date, amount, and description. There are also employees who work in the superstore to keep all the activities going. Each delivery man is recognized by their distinct employee id. They also have a few recorded details like their name, date of birth, salary, address, and phone number. This is how a superstore management system is built.

**ER Diagram:**



**Normalization:**

**Make**

**UNF**

Make(Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address, Pay\_ID, Pay\_Date, Pay\_Amount, Pay\_Desc)

**1NF**

There is no multi valued attribute , Realtion already in 1NF.

1. Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address, Pay\_ID, Pay\_Date, Pay\_Amount, Pay\_Desc

**2NF**

1. Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address.

2. Pay\_ID, Pay\_Date, Pay\_Amount, Pay\_Desc.

**3NF**

There is no transitive dependency. Relation already in 3NF

1. Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address.

2. Pay\_ID, Pay\_Date, Pay\_Amount, Pay\_Desc.

**TABLE CREATION**

1. Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address.

2. Pay\_ID, Pay\_Date, Pay\_Amount, Pay\_Desc, **Cus\_id**

**Purchase**

**UNF**

Purchase(Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address, Pro\_ID, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit)

**1NF**

Pro\_Type is a multi valued attribute.

1. Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address, Pro\_ID, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit

**2NF**

1. Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address.

2. Pro\_ID, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit

**3NF**

There is no transitive dependency. Relation already in 3NF

1. Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address.

2. Pro\_ID, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit

**TABLE CREATION**

1. Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address.

2. Pro\_id, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit

3. Cus\_id, Pro\_id

**Supply**

**UNF**

Supply(Sup\_ID, Sup\_Name, Sup\_Phone, Pro\_hID, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit)

**1NF**

Pro\_Type is a multi valued attribute.

1. Sup\_ID, Sup\_Name, Sup\_Phone, Pro\_ID, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit

**2NF**

1. Sup\_ID, Sup\_Name, Sup\_Phone

2. Pro\_ID, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit

**3NF**

There is no transitive hdependency. Relation already in 3NF

1. Sup\_ID, Sup\_Name, Sup\_Phone .

2. Pro\_ID, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit .

**TABLE CREATION**

1. Sup\_ID, Sup\_Name, Sup\_Phone .

2. Pro\_id, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit, **Sup\_id**

**Consists**

**UNF**

Consists(Name,Phone,Address,Shop\_No,Pro\_Type,Pro\_id,Pro\_Name,Pro\_Price,Pro\_Dese,Pro\_Weight,Pro\_unit)

**1NF**

Pro\_Type is a multi valued attribute.

1.Name,Phone,Address,Shop\_No,Pro\_Type,Pro\_id,Pro\_Name,Pro\_Price,Pro\_Dese,Pro\_Weight,Pro\_unit

**2NF**

1. Name,Phone,Address,Shop\_No

2. Pro\_Type,Pro\_id,Pro\_Name,Pro\_Price,Pro\_Dese,Pro\_Weight,Pro\_unit

**3NF**

There is no transitive dependency. Relation already in 3NF.

1. Name,Phone,Address,Shop\_No

2. Pro\_Type, Pro\_id, Pro\_Name,Pro\_Price,Pro\_Dese,Pro\_Weight,Pro\_unit

**Table Creation**

1. Name,Phone,Address,Shop\_No

2. Pro\_Type,Pro\_id,Pro\_Name,Pro\_Price,Pro\_Dese,Pro\_Weight,Pro\_unit, **Shop\_No**

3. Shop\_No , Pro\_id.

**Employs**

**UNF**

employs(Emp\_id, Emp\_Name, Emp\_Dob, Emp\_Salary, Emp\_Address, Emp\_Phone, Name, Phone, Address, Shop\_No)

**1NF**

There is no multivalued attribute, Realtion already in 1NF

1. Emp\_id, Emp\_Name, Emp\_Dob, Emp\_Salary, Emp\_Address, Emp\_Phone, Name, Phone, Address, Shop\_No

**2NF**

1. Emp\_id, Emp\_Name, Emp\_Dob, Emp\_Salary, Emp\_Address, Emp\_Phone

2. Name, Phone, Address, Shop\_No

**3NF**

There is no transitive dependency. Relation already in 3NF

1. Emp\_id, Emp\_Name, Emp\_Dob, Emp\_Salary, Emp\_Address, Emp\_Phone

2. Name, Phone, Address, Shop\_No

**Table Creation**

1. Emp\_id, Emp\_Name, Emp\_Dob, Emp\_Salary, Emp\_Address, Emp\_Phone, **Shop\_No**

2. Name, Phone, Address, Shop\_No

**Sells To**

**UNF**

Sells to(Cus\_id,Cus\_Name,Cus\_Phone,Cus\_Email,Cus\_Address,Name,Phone,Address,Shop\_No)

**1NF**

There is no multivalued attribute.

1.Cus\_id, Cus\_Name, Cus\_Phone, Cus\_Email, Cus\_Address, Name, Phone, Address, Shop\_No

**2NF**

1. Cus\_id, Cus\_Name, Cus\_Phone, Cus\_Email, Cus\_Address

2. Name, Phone, Address, Shop\_No

**3NF**

There is no transitive dependency. Relation already in 3NF

1. Cus\_id, Cus\_Name, Cus\_Phone, Cus\_Email, Cus\_Address

2. Name, Phone, Address, Shop\_No

**Table Creation**

1.Cus\_id, Cus\_Name, Cus\_Phone, Cus\_Email, Cus\_Address ,Shop\_No

2.Name, Phone, Address, Shop\_No.

3. Cus\_id, Shop\_No

**Temporary Table:**

1. ~~Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address.~~

2. Pay\_ID, Pay\_Date, Pay\_Amount, Pay\_Desc, **Cus\_id**

3. ~~Cus\_id, Cus\_Name , Cus\_phone, Cus\_Email, Cus\_Address.~~

4. ~~Pro\_id, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight, Pro\_Unit~~

5. Cus\_id, Pro\_id

6. Cus\_id, Cus\_Name, Cus\_Phone, Cus\_Email, Cus\_Address

7.~~Name, Phone, Address, Shop\_No.~~

8. Cus\_id, Shop\_No

9. Shop\_No , Pro\_id.

10. ~~Name, Phone, Address, Shop\_No~~

11. ~~Pro\_Type,Pro\_id,Pro\_Name,Pro\_Price,Pro\_Dese,Pro\_Weight,Pro\_unit~~

12. Emp\_id, Emp\_Name, Emp\_Dob, Emp\_Salary, Emp\_Address, Emp\_Phone, **Shop\_No**

13. Name, Phone, Address, Shop\_No

14. Sup\_ID, Sup\_Name, Sup\_Phone .

15. Pro\_id, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight,

Pro\_Unit, **Sup\_Id**

**Final Tables:**

1. Name, Phone, Address, Shop\_No

2. Emp\_id, Emp\_Name, Emp\_Dob, Emp\_Salary, Emp\_Address, Emp\_Phone, **Shop\_No**

3. Pro\_id, Pro\_Type, Pro\_Name, Pro\_Price, Pro\_Desc, Pro\_Weight,

Pro\_Unit, **Sup\_Id**

4. Cus\_id, Cus\_Name, Cus\_Phone, Cus\_Email, Cus\_Address

5. Sup\_ID, Sup\_Name, Sup\_Phone.

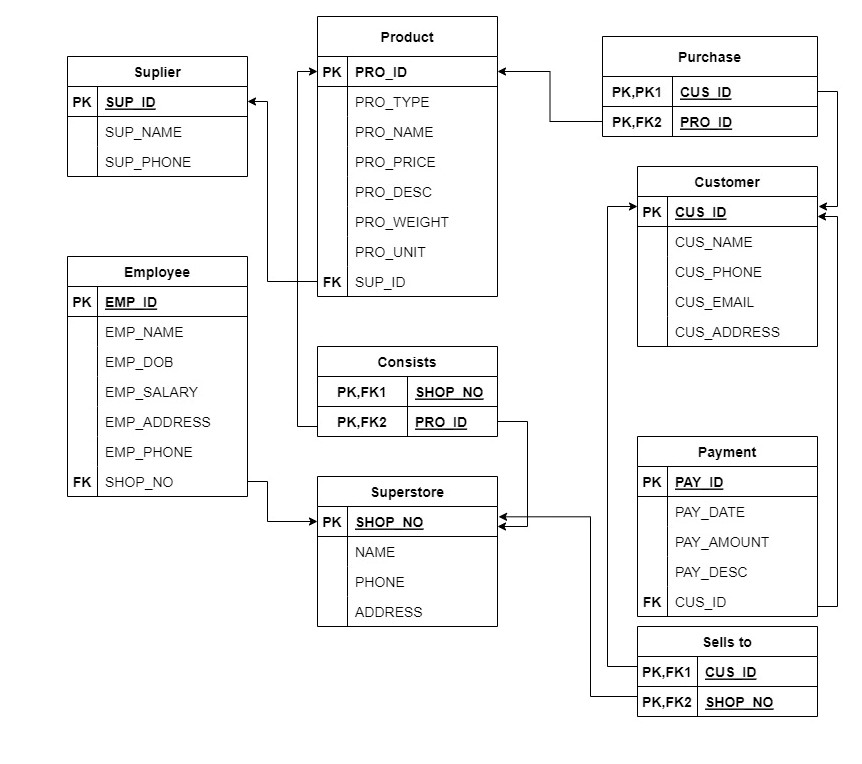
6. Pay\_ID, Pay\_Date, Pay\_Amount, Pay\_Desc, **Cus\_id**

7.Cus\_id, Pro\_id

8. Cus\_id, Shop\_No

9. Shop\_No , Pro\_id

**Schema Diagram:**

****

**Table Creation:**

CREATE TABLE SUPERSTORE(

NAME Varchar2(20),

PHONE Number(10),

ADDRESS Varchar2(20),

SHOP\_NO Number(10)

);

ALTER TABLE Superstore

ADD CONSTRAINT SHO\_PK

PRIMARY KEY (SHOP\_NO);

desc employee

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'SUPERSTORE';

drop TABLE SUPERSTORE

CREATE TABLE EMPLOYEE(

EMP\_ID Number(10),

EMP\_NAME varchar2(10),

EMP\_DOB Date,

EMP\_SALARY Number(7),

EMP\_ADDRESS varchar2(10),

EMP\_PHONE number(10),

SHOP\_NO number(10)

);

ALTER TABLE EMPLOYEE

ADD CONSTRAINT EMP\_PK

PRIMARY KEY (EMP\_ID);

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'EMPLOYEE';

alter table EMPLOYEE

add constraint EMP\_SHO\_FK foreign key(SHOP\_NO) references Superstore(SHOP\_NO);

desc employee

drop table EMPLOYEE

CREATE TABLE Product(

PRO\_ID Number(10),

PRO\_TYPE Varchar2(20),

PRO\_NAME Varchar2(20),

PRO\_PRICE Number(10),

PRO\_DESC Varchar2(20),

PRO\_WEIGHT Number(10,2),

PRO\_UNIT Number(10),

SUP\_ID Number(10)

);

ALTER TABLE PRODUCT

ADD CONSTRAINT PRO\_PK

PRIMARY KEY (PRO\_ID);

alter table PRODUCT

add constraint PRO\_SUP\_FK foreign key(SUP\_ID) references SUPPLIER(SUP\_ID);

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'PRODUCT';

drop table PRODUCT

CREATE TABLE CUSTOMER(

CUS\_ID NUMBER(10),

CUS\_NAME Varchar2(20),

CUS\_PHONE Number(10),

CUS\_EMAIL Varchar2(10),

CUS\_ADDRESS Varchar2(10)

);

ALTER TABLE CUSTOMER

ADD CONSTRAINT CUS\_PK

PRIMARY KEY (CUS\_ID);

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'CUSTOMER';

desc customer

drop table CUSTOMER

CREATE TABLE SUPPLIER(

SUP\_ID NUMBER(10),

SUP\_NAME Varchar2(20),

SUP\_PHONE Number(10)

);

ALTER TABLE SUPPLIER

ADD CONSTRAINT SUP\_PK

PRIMARY KEY (SUP\_ID);

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'SUPPLIER';

desc

drop table SUPPLIER

CREATE TABLE PAYMENT(

PAY\_ID NUMBER(10),

PAY\_DATE date,

PAY\_AMOUNT Number(10),

PAY\_DESC Varchar2(20),

CUS\_ID Number(10)

);

ALTER TABLE PAYMENT

ADD CONSTRAINT PAY\_PK

PRIMARY KEY (PAY\_ID);

alter table PAYMENT

add constraint PAY\_CUS\_FK foreign key(CUS\_ID) references CUSTOMER(CUS\_ID);

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'PAYMENT';

desc

drop table PAYMENT

desc purchase

CREATE TABLE PURCHASE(

CUS\_ID Number(10),

PRO\_ID Number(10),

constraint PUR\_PK primary key (CUS\_ID, PRO\_ID)

);

alter table PURCHASE

add constraint PUR\_CUS\_FK foreign key(CUS\_ID) references CUSTOMER(CUS\_ID);

alter table PURCHASE

add constraint PUR\_PRO\_FK foreign key(PRO\_ID) references PRODUCT(PRO\_ID);

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'PURCHASE';

drop table PURCHASE

CREATE TABLE SELLSTO(

CUS\_ID NUMBER(10),

SHOP\_NO Number(10),

constraint SELLS\_PK primary key (CUS\_ID, SHOP\_NO)

);

alter table SELLSTO

add constraint SELLS\_CUS\_FK foreign key(CUS\_ID) references CUSTOMER(CUS\_ID);

alter table SELLSTO

add constraint SELLS\_SHO\_FK foreign key(SHOP\_NO) references SUPERSTORE(SHOP\_NO);

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'SELLSTO';

drop table SELLSTO

desc consists

CREATE TABLE CONSISTS(

SHOP\_NO Number(10),

PRO\_ID Number(10),

constraint CONS\_PK primary key (SHOP\_NO,PRO\_ID)

);

alter table CONSISTS

add constraint CONS\_SHO\_FK foreign key(SHOP\_NO) references SUPERSTORE(SHOP\_NO);

alter table CONSISTS

add constraint CONS\_PRO\_FK foreign key(PRO\_ID) references PRODUCT(PRO\_ID);

select constraint\_name, column\_name

from user\_cons\_columns

where table\_name = 'CONSISTS';

drop table CONSISTS

INSERT INTO SUPERSTORE VALUES ('FARM PICKED',789564, 'NEVADA', 2254);

INSERT INTO SUPERSTORE VALUES ('MAX GROCERY',758925, 'DENVER', 1734);

INSERT INTO SUPERSTORE VALUES ('STOP & SHOP',284524, 'KANSAS', 9872);

INSERT INTO SUPERSTORE VALUES ('GROCERY DEPOT',631510, 'NASHVILLE', 1342);

INSERT INTO SUPERSTORE VALUES ('FRESH FOODS',851265, 'OREGON', 6452);

SELECT \*

FROM SUPERSTORE

INSERT INTO EMPLOYEE VALUES (73124,'NATHAN', '11-NOV-1978', 1000, 'NEVADA', 567812, 2254);

INSERT INTO EMPLOYEE VALUES (75435,'DRAKE', '07-JUN-1984', 800, 'DENVER', 786421, 1734);

INSERT INTO EMPLOYEE VALUES (98767,'AURTHUR', '13-JAN-1987', 1400, 'KANSAS', 167598, 9872);

INSERT INTO EMPLOYEE VALUES (90762,'MORGAN', '26-SEP-1978', 1100, 'NASHVILLE', 753892, 1342);

INSERT INTO EMPLOYEE VALUES (69875,'RHEA', '27-JUN-1978', 1700, 'OREGON', 432114, 6452);

SELECT \*

FROM EMPLOYEE

INSERT INTO SUPPLIER VALUES (67481,'JAKE', 875168);

INSERT INTO SUPPLIER VALUES (89571,'GYLLENHAAL', 567122);

INSERT INTO SUPPLIER VALUES (97056,'ROBERT', 751123);

INSERT INTO SUPPLIER VALUES (76233,'NIRO', 880751);

INSERT INTO SUPPLIER VALUES (57122,'JACKMAN', 751282);

SELECT \*

FROM SUPPLIER

INSERT INTO PRODUCT VALUES (8934,'NICHE', 'T-Shirt', 10, 'Made In VIETNAM', 100.00, 6, 67481);

INSERT INTO PRODUCT VALUES (4122,'LUXURY', 'PERFUME', 200, 'Made In FRANCE', 500.00, 8, 89571);

INSERT INTO PRODUCT VALUES (5791,'CONSUMER', 'BUTTER', 5, 'Made In INDIA', 500.00, 12 , 97056);

INSERT INTO PRODUCT VALUES (7589,'COMMODITIES', 'OIL', 2, 'Made In BRAZIL', 1000.00, 10, 76233);

INSERT INTO PRODUCT VALUES (8591,'CONVENIENCE', 'SOFT DRINK', 3, 'Made In AUSTRIA', 25.00, 7, 57122);

SELECT \*

FROM PRODUCT

INSERT INTO CUSTOMER VALUES (3612,'AEGON', 12412, 'aegon@.com', 'KANSAS');

INSERT INTO CUSTOMER VALUES (5142,'ALLEN', 41242, 'allen@.com', 'DENVER');

INSERT INTO CUSTOMER VALUES (7491,'BEN', 63463, 'ben@.com', 'NEVADA');

INSERT INTO CUSTOMER VALUES (9755,'DAVID', 76518, 'david@.com', 'NASHVILLE');

INSERT INTO CUSTOMER VALUES (9763,'RABIOT', 75912, 'rabit@.com', 'OREGON');

SELECT \*

FROM CUSTOMER

INSERT INTO PAYMENT VALUES (4124, '15-JUN-2009', 7, 'PAID', 7491);

INSERT INTO PAYMENT VALUES (6741, '21-JUN-2009', 17, 'PAID', 3612);

INSERT INTO PAYMENT VALUES (8791, '08-JUN-2009', 90, 'DUE', 5142);

INSERT INTO PAYMENT VALUES (4742, '19-JUN-2009', 67, 'PAID', 9763);

INSERT INTO PAYMENT VALUES (8091, '29-JUN-2009', 45, 'DUE', 9755);

SELECT \*

FROM PAYMENT

INSERT INTO PURCHASE VALUES (3612,8934);

INSERT INTO PURCHASE VALUES (5142,4122);

INSERT INTO PURCHASE VALUES (7491,5791);

INSERT INTO PURCHASE VALUES (9755,7589);

INSERT INTO PURCHASE VALUES (9763,8591);

SELECT \*

FROM PURCHASE

INSERT INTO SELLSTO VALUES (3612,2254);

INSERT INTO SELLSTO VALUES (5142,1734);

INSERT INTO SELLSTO VALUES (7491,9872);

INSERT INTO SELLSTO VALUES (9755,1342);

INSERT INTO SELLSTO VALUES (9763,6452);

SELECT \*

FROM SELLSTO

INSERT INTO CONSISTS VALUES (2254,8934);

INSERT INTO CONSISTS VALUES (1734,4122);

INSERT INTO CONSISTS VALUES (9872,5791);

INSERT INTO CONSISTS VALUES (1342,7589);

INSERT INTO CONSISTS VALUES (6452,8591);

SELECT \*

FROM CONSISTS

CREATE USER t

IDENTIFIED BY abcd;

**DATA INSERTION**

**Superstore Table:**

1. INSERT INTO SUPERSTORE VALUES ('FARM PICKED',789564, 'NEVADA', 2254);

2. INSERT INTO SUPERSTORE VALUES ('MAX GROCERY',758925, 'DENVER', 1734);

3. INSERT INTO SUPERSTORE VALUES ('STOP & SHOP',284524, 'KANSAS', 9872);

4. INSERT INTO SUPERSTORE VALUES ('GROCERY DEPOT',631510, 'NASHVILLE', 1342);

5. INSERT INTO SUPERSTORE VALUES ('FRESH FOODS',851265, 'OREGON', 6452);



**Employee Table:**

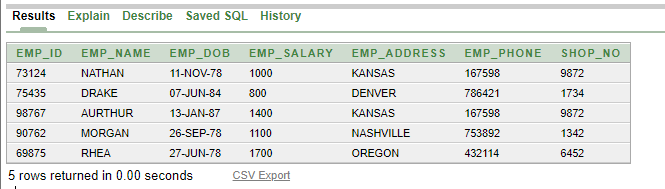
1. INSERT INTO EMPLOYEE VALUES (73124,'NATHAN', '11-NOV-1978', 1000, 'NEVADA', 567812, 2254);

2. INSERT INTO EMPLOYEE VALUES (75435,'DRAKE', '07-JUN-1984', 800, 'DENVER', 786421, 1734);

3. INSERT INTO EMPLOYEE VALUES (98767,'AURTHUR', '13-JAN-1987', 1400, 'KANSAS', 167598, 9872);

4. INSERT INTO EMPLOYEE VALUES (90762,'MORGAN', '26-SEP-1978', 1100, 'NASHVILLE', 753892, 1342);

5. INSERT INTO EMPLOYEE VALUES (69875,'RHEA', '27-JUN-1978', 1700, 'OREGON', 432114, 6452);



**Supplier Table:**

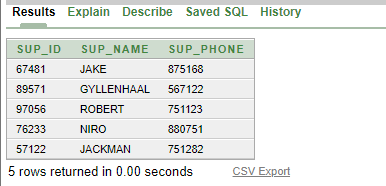
1. INSERT INTO SUPPLIER VALUES (67481,'JAKE', 875168);

2. INSERT INTO SUPPLIER VALUES (89571,'GYLLENHAAL', 567122);

3. INSERT INTO SUPPLIER VALUES (97056,'ROBERT', 751123);

4. INSERT INTO SUPPLIER VALUES (76233,'NIRO', 880751);

5. INSERT INTO SUPPLIER VALUES (57122,'JACKMAN', 751282);



**Product Table:**

1. INSERT INTO PRODUCT VALUES (8934,'NICHE', 'T-Shirt', 10, 'Made In VIETNAM', 100.00, 6, 67481);

2. INSERT INTO PRODUCT VALUES (4122,'LUXURY', 'PERFUME', 200, 'Made In FRANCE', 500.00, 8, 89571);

3. INSERT INTO PRODUCT VALUES (5791,'CONSUMER', 'BUTTER', 5, 'Made In INDIA', 500.00, 12 , 97056);

4. INSERT INTO PRODUCT VALUES (7589,'COMMODITIES', 'OIL', 2, 'Made In BRAZIL', 1000.00, 10, 76233);

5. INSERT INTO PRODUCT VALUES (8591,'CONVENIENCE', 'SOFT DRINK', 3, 'Made In AUSTRIA', 25.00, 7, 57122);



**Customer Table:**

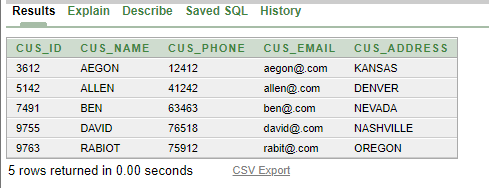
1. INSERT INTO CUSTOMER VALUES (3612,'AEGON', 12412, 'aegon@.com', 'KANSAS');

2. INSERT INTO CUSTOMER VALUES (5142,'ALLEN', 41242, 'allen@.com', 'DENVER');

3. INSERT INTO CUSTOMER VALUES (7491,'BEN', 63463, 'ben@.com', 'NEVADA');

4. INSERT INTO CUSTOMER VALUES (9755,'DAVID', 76518, 'david@.com', 'NASHVILLE');

5. INSERT INTO CUSTOMER VALUES (9763,'RABIOT', 75912, 'rabit@.com', 'OREGON');



**Payment Table:**

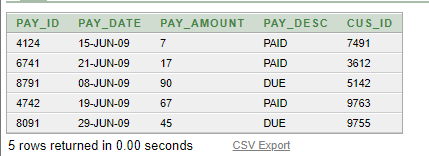
1. INSERT INTO PAYMENT VALUES (4124, '15-JUN-2009', 7, 'PAID', 7491);

2. INSERT INTO PAYMENT VALUES (6741, '21-JUN-2009', 17, 'PAID', 3612);

3. INSERT INTO PAYMENT VALUES (8791, '08-JUN-2009', 90, 'DUE', 5142);

4. INSERT INTO PAYMENT VALUES (4742, '19-JUN-2009', 67, 'PAID', 9763);

5. INSERT INTO PAYMENT VALUES (8091, '29-JUN-2009', 45, 'DUE', 9755);



**Purchase Table:**

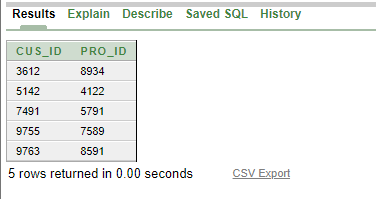
1.INSERT INTO PURCHASE VALUES (3612,8934);

2. INSERT INTO PURCHASE VALUES (5142,4122);

3.INSERT INTO PURCHASE VALUES (7491,5791);

4. INSERT INTO PURCHASE VALUES (9755,7589);

5. INSERT INTO PURCHASE VALUES (9763,8591);



**Sells to Table:**

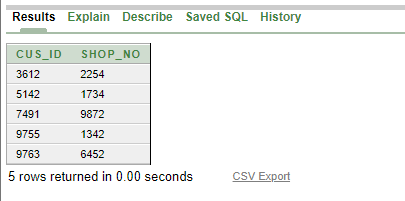
1. INSERT INTO SELLSTO VALUES (3612,2254);

2. INSERT INTO SELLSTO VALUES (5142,1734);

3. INSERT INTO SELLSTO VALUES (7491,9872);

4.INSERT INTO SELLSTO VALUES (9755,1342);

5.INSERT INTO SELLSTO VALUES (9763,6452);



**Consists Table:**

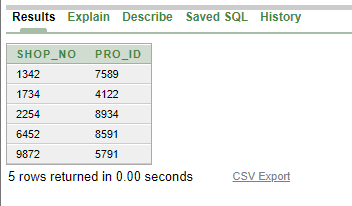
1. INSERT INTO CONSISTS VALUES (2254,8934);

2. INSERT INTO CONSISTS VALUES (1734,4122);

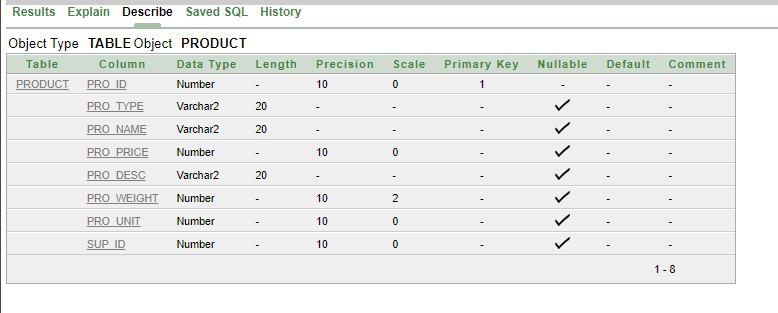
3. INSERT INTO CONSISTS VALUES (9872,5791);

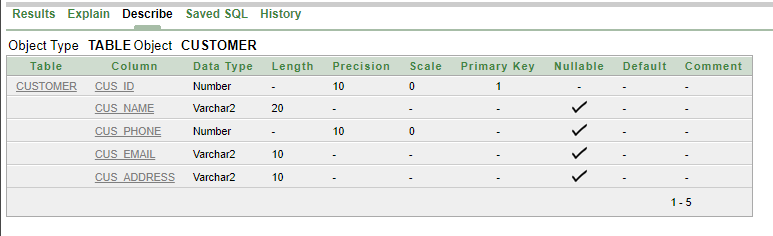
4. INSERT INTO CONSISTS VALUES (1342,7589);

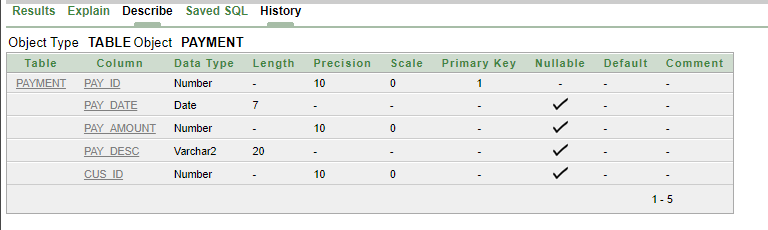
5. INSERT INTO CONSISTS VALUES (6452,8591);

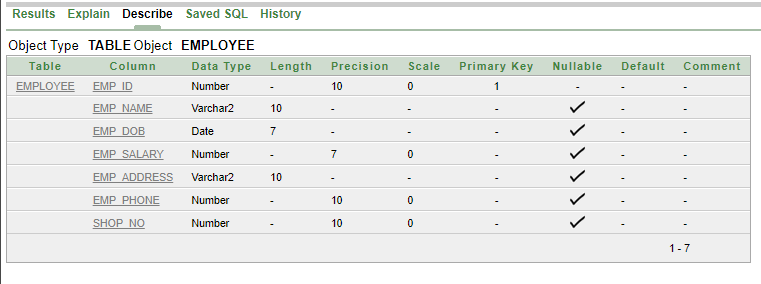


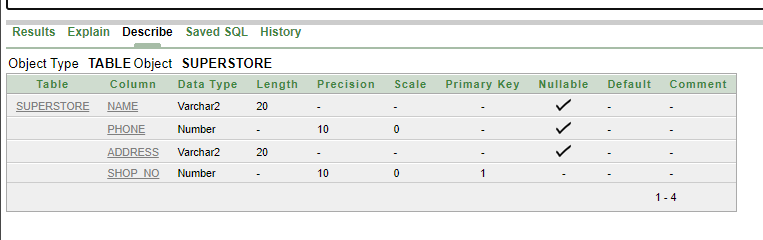
**Screenshots After Inserting Data:**

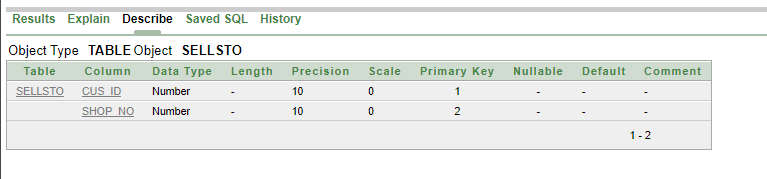


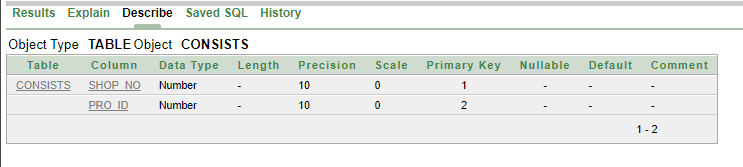


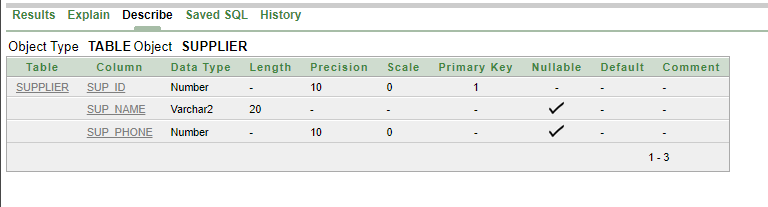


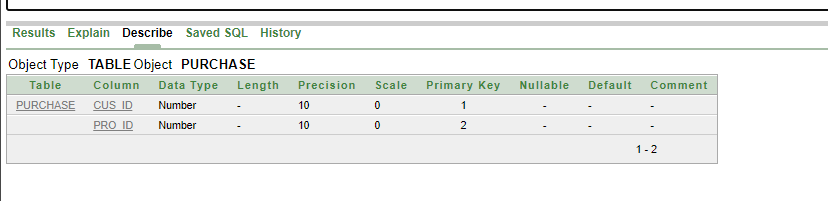












**Query Writing:**

**SINGLE ROW FUNCTION**

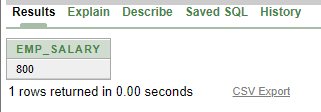
**QUES 1.** Display only the column Employee salary from EMPLOYEE table for employee “DRAKE”

**Query:**

SELECT EMP\_SALARY

FROM EMPLOYEE

WHERE EMP\_NAME = 'DRAKE';

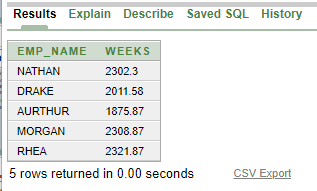


**QUES 2.** Determine age of all employess in Weeks.

**Query:**

SELECT EMP\_NAME, ROUND((SYSDATE-EMP\_DOB)/7,2) WEEKS

FROM EMPLOYEE



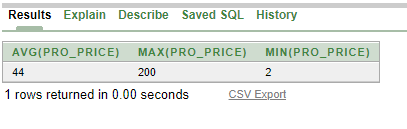
**GROUP FUNCTION**

**QUES 1.** Display average product price, max product price, minimum product price from Product table

**Query:**

SELECT AVG(PRO\_PRICE), MAX(PRO\_PRICE), MIN(PRO\_PRICE)

FROM PRODUCT



**QUES 2**. Display employee name, employee salary, employee address who have salary more than 1000, in descending order

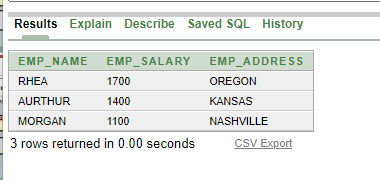
**Query:**

SELECT EMP\_NAME, EMP\_SALARY, EMP\_ADDRESS

FROM EMPLOYEE

WHERE EMP\_SALARY> 1000

ORDER BY EMP\_SALARY DESC



**SUB-QUERY:**

**QUES 1.** Select product name from Product table which has more price than Butter using subquery.

**Query:**

SELECT PRO\_NAME

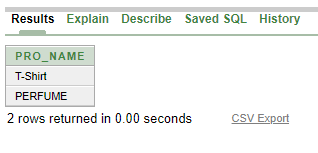
FROM PRODUCT

WHERE PRO\_PRICE>

(SELECT PRO\_PRICE

FROM PRODUCT

WHERE PRO\_NAME='BUTTER');



**Ques 2:**

Display customer name from customer table where customer id is 9763 using subquey.

SELECT CUS\_NAME

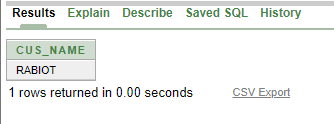
FROM CUSTOMER

WHERE CUS\_ADDRESS =

(SELECT CUS\_ADDRESS

FROM CUSTOMER

WHERE CUS\_ID= 9763);



**JOINING:**

**QUES 1**

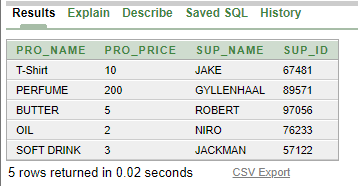
Display Product name, Product price, supplier name, supplier id from Product and supplier table using equijoins.

**Query:**

SELECT PRODUCT.PRO\_NAME, PRODUCT.PRO\_PRICE, SUPPLIER.SUP\_NAME, SUPPLIER.SUP\_ID

FROM SUPPLIER, PRODUCT

WHERE SUPPLIER.SUP\_ID=PRODUCT.SUP\_ID;



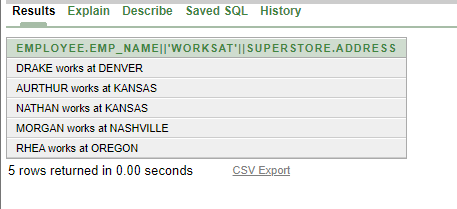
**QUES 2.**

Query Display workplace of all employee using joining method.

SELECT EMPLOYEE.EMP\_NAME||' works at '||SUPERSTORE.ADDRESS

FROM EMPLOYEE, SUPERSTORE

WHERE EMPLOYEE.SHOP\_NO = SUPERSTORE.SHOP\_NO;



**Creating View:**

**Ques 1 .**

Create a view Payment10, which contains Payment amount, payment id,customer id for due payment description,

Query

CREATE VIEW Payment10

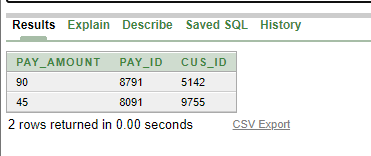
AS SELECT PAY\_Amount, PAY\_ID, CUS\_ID

FROM PAYMENT

WHERE PAY\_DESC ='DUE';

select \*

from Payment10



**Ques 2.**

Create a view Employee10, which contains Employee name, Employee Date of Birth, Employee Address for employees who have salary more than 1500.

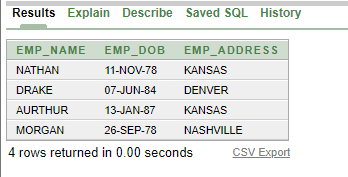
**Query:**

CREATE VIEW employee10 AS SELECT EMP\_NAME,EMP\_DOB,EMP\_ADDRESS FROM EMPLOYEE

WHERE EMP\_SALARY <1500;

select \*

from employee10



**Relational Algebra:**

1. Fine the employee’s name where EMP\_ID = 73124

**Ans: emp\_name (emp\_id=”Nathan”())**

2. Fine the product type where PRO\_NAME = Butter

**Ans: PRO\_TYPE (PRO\_TYPE=””(CONSUMER))**

3. Find the supplier phone number where SUP\_NAME = ROBERT

**Ans: SUP\_PHONE (SUP\_PHONE=””(751123))**

4. Find the payment date where PAY\_ID = 4742

**Ans: PAY\_DATE (PAY\_DATE=””(19-JUN-09))**

5. Find the superstore name where SHOP\_NO = 1342

**Ans: NAME (NAME=””(GROCERY DEPOT))**

**Conclusion**

This project helped us understand how to efficiently handle large amounts and multiple types of data. We have also learned how SQL works, structure information, and analyze them ourselves. If we're looking to the future, we obviously want to invest our time in looking deeper into database management that's still widely used and will continue to be in the years to come.