AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Semester: Spring 2022-23
Topic: Computer Shop Management System
Group no: 1

ID	NAME
21-45721-3	Mushfequr Rahman
22-46323-1	Md. Nazmul Islam Rahat
22-46354-1	Maruf Ahmed
22-46432-1	Milton Neogi

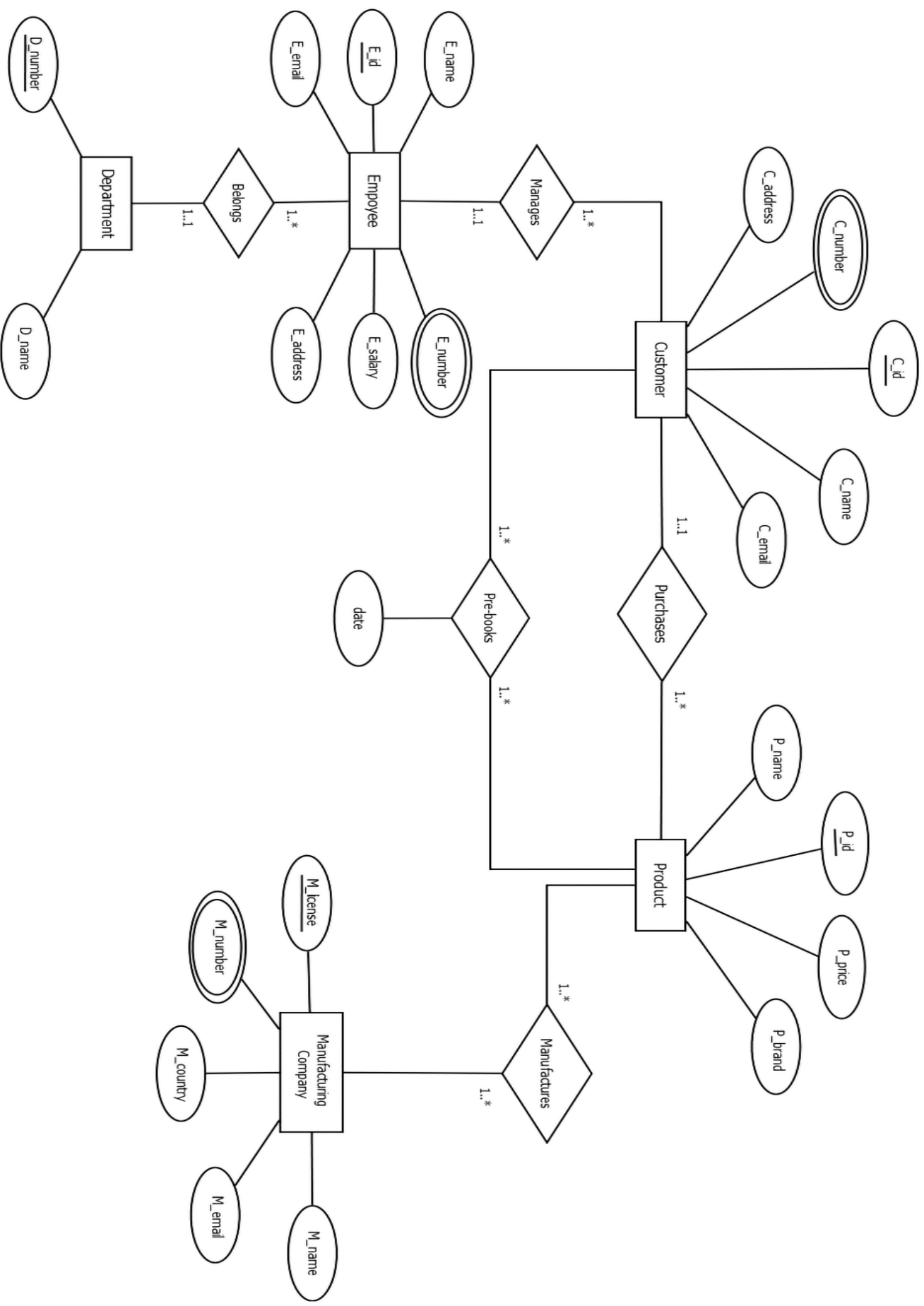| |
|---|
| Submitted to: Md Sajid Bin-Faisal |
| Section: E |
| Course: Introduction to Database |

Table of Contents

Introduction

The name of our project is Computer Shop Management System.
In today's digital age, computer shops have become an integral part of our lives, providing essential products to customers. However, managing a computer shop can be a difficult task. To simplify the complexities, computer shop owners can find this computer management system very handy. This system has been created using oracle database 10g software. The Computer Shop Management System is a software designed to assist computer shop owners in effectively organizing their day-to-day task. They can keep track of customers, products, sales, employees etc. Hence, run their business smoothly.

Case study for Computer Shop Management System

In a Computer Shop Management System, a customer may purchase many products. One product may be purchased by exactly one customer. A customer is identified by a customer ID. The system also stores customer name, email, phone number and address. A product is identified by product name, product ID, product price and brand name. A customer may also pre-book many products. A product may be pre-booked by many customers. To find the priority of the pre-booking the date of pre-booking is also stored. A product is manufactured by at least one manufacturing company. A manufacturing company may manufacture many products. The system stores manufacturing company name, email, country, contact number and it is uniquely identified by license number. The customers are managed by employees. An employee may manage many customers, but a customer is managed by one employee. The system also stores employee name, employee ID, email, address, salary and phone number. An employee belongs to exactly one department, but a department may have many employees. Each department has a name, and the unique property of each department is a department number.

ER-Diagram

Department — D_number, D_name

Belongs (1..1 — 1..*)

Empoyee — E_email, E_id, E_name, E_address, E_salary, E_number

Manages (1..1 — 1..*)

Customer — C_address, C_number, C_id, C_name, C_email

Purchases (1..1 — 1..*)

Pre-books (1..* — 1..*) — date

Product — P_name, P_id, P_price, P_brand

Manufactures (1..* — 1..*)

Manufacturing Company — M_number, M_license, M_country, M_name, M_email

Normalization:

UNF (Purchases): C_address, C_number, <u>C_id</u>, C_name, C_email, P_name, <u>P_id</u>, P_price, P_brand.
1NF: C_address, C_number, <u>C_id</u>, C_name, C_email, P_name, <u>P_id</u>, P_price, P_brand.
2NF: (i) C_address, C_number, <u>C_id</u>(PK), C_name, C_email.
    (ii) P_name, <u>P_id</u>(PK), P_price, P_brand, <u>C_id</u>(FK).
3NF: Same as 2NF.

UNF (Manufactures): P_name, <u>P_id</u>, P_price, P_brand, <u>M_license</u>, M_number, M_country, M_email, M_name.
1NF: P_name, <u>P_id</u>, P_price, P_brand, <u>M_license</u>, M_number, M_country, M_email, M_name.
2NF: (i) P_name, <u>P_id</u>(PK), P_price, P_brand.
    (ii) <u>M_license</u>(PK), M_number, M_country, M_email, M_name.
    (iii) <u>P_id</u>(PK), <u>M_license</u>(FK).
3NF: Same as 2NF.

UNF (Manages): E_name, <u>E_id</u>, E_email, E_number, E_salary, E_address, C_address, C_number, <u>C_id</u>, C_name, C_email.
1NF: E_name, <u>E_id</u>, E_email, E_number, E_salary, E_address, C_address, C_number, <u>C_id</u>, C_name, C_email.
2NF: (i) E_name, <u>E_id</u>(PK), E_email, E_number, E_salary, E_address.
    (ii) C_address, C_number, <u>C_id</u>(PK), C_name, C_email, <u>E_id</u>(FK).
3NF: Same as 2NF.

UNF (Belongs): <u>D_number</u>, D_name, E_name, <u>E_id</u>, E_email, E_number, E_salary, E_address.
1NF: <u>D_number</u>, D_name, E_name, <u>E_id</u>, E_email, E_number, E_salary, E_address.
2NF: (i) <u>D_number</u>(PK), D_name.
    (ii) E_name, <u>E_id</u>(PK), E_email, E_number, E_salary, E_address, <u>D_number</u>(FK).
3NF: Same as 2NF.

UNF (Pre-books): C_address, C_number, <u>C_id</u>, C_name, C_email, P_name, <u>P_id</u>, P_price, P_brand, date.
1NF: C_address, C_number, <u>C_id</u>, C_name, C_email, P_name, <u>P_id</u>, P_price, P_brand, date.
2NF: (i) C_address, C_number, <u>C_id</u>(PK), C_name, C_email.
    (ii) P_name, <u>P_id</u>(PK), P_price, P_brand.
    (iii) <u>P_id</u>(PK), <u>C_id</u>(FK), date.
3NF: Same as 2NF.

Finalization:

1. C_address, C_number, C_id(PK), C_name, C_email.
2. P_name, P_id(PK), P_price, P_brand, C_id(FK).
3. P_name, P_id(PK), P_price, P_brand.
4. M_license(PK), M_number, M_country, M_email, M_name.
5. P_id(PK), M_license(FK).
6. E_name, E_id(PK), E_email, E_number, E_salary, E_address.
7. C_address, C_number, C_id(PK), C_name, C_email, E_id(FK).
8. D_number(PK), D_name.
9. E_name, E_id(PK), E_email, E_number, E_salary, E_address, D_number(FK).
10. ~~C_address, C_number, C_id(PK), C_name, C_email.~~
11. ~~P_name, P_id(PK), P_price, P_brand.~~
12. P_id(PK), C_id(FK), date.

Final Tables:

SL. No.	Tables	Table Names
1	C_address, C_number, C_id(PK), C_name, C_email	Customer
2	P_name, P_id(PK), P_price, P_brand, C_id(FK)	Purchase
3	P_name, P_id(PK), P_price, P_brand	Product
4	M_license(PK), M_number, M_country, M_email, M_name	Manufacturing Company
5	P_id(PK), M_license(FK)	Manufacture
6	E_name, E_id(PK), E_email, E_number, E_salary, E_address	Employee
7	C_address, C_number, C_id(PK), C_name, C_email, E_id(FK)	Manage
8	D_number(PK), D_name	Department
9	E_name, E_id(PK), E_email, E_number, E_salary, E_address, D_number(FK)	Belong
10	P_id(PK), C_id(FK), date	Pre-book

System creation and granting privilege:

```
create user sun identified by moon

grant connect, resource to sun
```

Results   Explain   Describe   Saved SQL   History

Statement processed.

0.46 seconds

Table creation:

```
create table customer (c_name varchar2(100), c_id number(5) primary key, c_number number(7), c_address varchar2(200), c_email varchar2(50))

describe customer
```

Results   Explain   Describe   Saved SQL   History

Object Type **TABLE** Object **CUSTOMER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	C_NAME	Varchar2	100	-	-	-	✓	-	-
	C_ID	Number	-	5	0	1	-	-	-
	C_NUMBER	Number	-	7	0	-	✓	-	-
	C_ADDRESS	Varchar2	200	-	-	-	✓	-	-
	C_EMAIL	Varchar2	50	-	-	-	✓	-	-
									1 - 5

```
create table purchase (p_name varchar2(100), p_id number(5) primary key, p_brand varchar2(100), p_price number(6), c_id number(5), constraint ci
foreign key(c_id) references customer(c_id))

describe purchase
```

Results   Explain   Describe   Saved SQL   History

Object Type **TABLE** Object **PURCHASE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PURCHASE	P_NAME	Varchar2	100	-	-	-	✓	-	-
	P_ID	Number	-	5	0	1	-	-	-
	P_BRAND	Varchar2	100	-	-	-	✓	-	-
	P_PRICE	Number	-	6	0	-	✓	-	-
	C_ID	Number	-	5	0	-	✓	-	-
									1 - 5

```
create table product (p_name varchar2(100), p_id number(5) primary key, p_brand varchar2(100), p_price number(6))

describe product
```

Object Type **TABLE** Object **PRODUCT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT	P_NAME	Varchar2	100	-	-	-	✔	-	-
	P_ID	Number	-	5	0	1	-	-	-
	P_BRAND	Varchar2	100	-	-	-	✔	-	-
	P_PRICE	Number	-	6	0	-	✔	-	-
									1 - 4

```
create table "MANUFACTURING COMPANY" (m_name varchar2(100), m_license number(5) primary key, m_number number(7), m_country varchar2(50), m_email
varchar2(50))

describe "MANUFACTURING COMPANY"
```

Object Type **TABLE** Object **MANUFACTURING COMPANY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANUFACTURING COMPANY	M_NAME	Varchar2	100	-	-	-	✔	-	-
	M_LICENSE	Number	-	5	0	1	-	-	-
	M_NUMBER	Number	-	7	0	-	✔	-	-
	M_COUNTRY	Varchar2	50	-	-	-	✔	-	-
	M_EMAIL	Varchar2	50	-	-	-	✔	-	-
									1 - 5

```
create table manufacture (p_id number(5) primary key, m_license number(5), constraint ml foreign key(m_license) references "MANUFACTURING COMPANY"
(m_license))

describe manufacture
```

Object Type **TABLE** Object **MANUFACTURE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANUFACTURE	P_ID	Number	-	5	0	1	-	-	-
	M_LICENSE	Number	-	5	0	-	✔	-	-
									1 - 2

```
create table employee (e_name varchar2(100), e_id number(5) primary key, e_number number(7), e_address varchar2(200), e_email varchar2(50),
e_salary number(5))
describe employee
```

Object Type **TABLE** Object **EMPLOYEE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	E_NAME	Varchar2	100	-	-	-	✔	-	-
	E_ID	Number	-	5	0	1	-	-	-
	E_NUMBER	Number	-	7	0	-	✔	-	-
	E_ADDRESS	Varchar2	200	-	-	-	✔	-	-
	E_EMAIL	Varchar2	50	-	-	-	✔	-	-
	E_SALARY	Number	-	5	0	-	✔	-	-
									1 - 6

```
create table manage (c_name varchar2(100), c_id number(5) primary key, c_number number(7), c_address varchar2(200), c_email varchar2(50), e_id
number(5), constraint ei foreign key(e_id) references employee(e_id))

describe manage
```

Object Type **TABLE** Object **MANAGE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGE	C_NAME	Varchar2	100	-	-	-	✓	-	-
	C_ID	Number	-	5	0	1	-	-	-
	C_NUMBER	Number	-	7	0	-	✓	-	-
	C_ADDRESS	Varchar2	200	-	-	-	✓	-	-
	C_EMAIL	Varchar2	50	-	-	-	✓	-	-
	E_ID	Number	-	5	0	-	✓	-	-
									1 - 6

```
create table department (d_name varchar2(100), d_number number(2) primary key)

describe department
```

Object Type **TABLE** Object **DEPARTMENT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENT	D_NAME	Varchar2	100	-	-	-	✓	-	-
	D_NUMBER	Number	-	2	0	1	-	-	-
									1 - 2

```
create table belong (e_name varchar2(100), e_id number(5) primary key, e_number number(7), e_address varchar2(200), e_email varchar2(50), e_salary
number(5), d_number number(2), constraint dn foreign key(d_number) references department(d_number))

describe belong
```

Object Type **TABLE** Object **BELONG**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BELONG	E_NAME	Varchar2	100	-	-	-	✓	-	-
	E_ID	Number	-	5	0	1	-	-	-
	E_NUMBER	Number	-	7	0	-	✓	-	-
	E_ADDRESS	Varchar2	200	-	-	-	✓	-	-
	E_EMAIL	Varchar2	50	-	-	-	✓	-	-
	E_SALARY	Number	-	5	0	-	✓	-	-
	D_NUMBER	Number	-	2	0	-	✓	-	-
									1 - 7

```
create table "PRE-BOOK" (p_id number(5) primary key, b_date date, c_id number(5), constraint cid foreign key(c_id) references customer(c_id))

describe "PRE-BOOK"
```

Object Type **TABLE** Object **PRE-BOOK**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRE-BOOK	P_ID	Number	-	5	0	1	-	-	-
	B_DATE	Date	7	-	-	-	✓	-	-
	C_ID	Number	-	5	0	-	✓	-	-
									1 - 3

Tables of values:

Customer:

C_NAME	C_ID	C_NUMBER	C_ADDRESS	C_EMAIL
Habib	68731	0155367	Mirpur	habib31@gmail.com
Siam	68732	0181630	Dhanmondi	siam32@gmail.com
Ripon	68733	0176893	Khilkhet	ripon33@gmail.com
Mamun	68734	0159024	Uttara	mamun34@gmail.com

Purchase:

P_NAME	P_ID	P_BRAND	P_PRICE	C_ID
Monitor	96720	ASUS	10000	68731
Mouse	51824	RAZER	700	68732
Keyboard	27153	HP	1200	68733
RAM	70182	SAMSUNG	3000	68734

Product:

P_NAME	P_ID	P_BRAND	P_PRICE
Monitor	96720	ASUS	10000
Mouse	51824	RAZER	700
Keyboard	27153	HP	1200
RAM	70182	SAMSUNG	3000

Manufacturing Company:

M_NAME	M_LICENSE	M_NUMBER	M_COUNTRY	M_EMAIL
ASUS	00913	8860976	Taiwan	asus@yahoo.com
RAZER	00651	6523980	Singapore	razer@yahoo.com
HP	00571	1629035	USA	hp@gmail.com
SAMSUNG	00465	8216234	South Korea	samsung@gmail.com

Manufacture:

P_ID	M_LICENSE
96720	00913
51824	00651
27153	00571
70182	00465

Employee:

E_NAME	E_ID	E_NUMBER	E_ADDRESS	E_EMAIL	E_SALARY
Rahim	99912	0175254	Mirpur	rahim12@gmail.com	10000
Karim	99922	0158058	Banani	karim22@gmail.com	12000
Rasel	99933	0171854	Mirpur	rasel33@gmail.com	11000

Manage:

C_NAME	C_ID	C_NUMBER	C_ADDRESS	C_EMAIL	E_ID
Habib	68731	0155367	Mirpur	habib31@gmail.com	99912
Siam	68732	0181630	Dhanmondi	siam32@gmail.com	99922
Ripon	68733	0176893	Khilkhet	ripon33@gmail.com	99933
Mamun	68734	0159024	Uttara	mamun34@gmail.com	99912

Department:

D_NAME	D_NUMBER
Sales	10
Accounts	20
Managing	30

Belong:

E_NAME	E_ID	E_NUMBER	E_ADDRESS	E_EMAIL	E_SALARY	D_NUMBER
Rahim	99912	0175254	Mirpur	rahim12@gmail.com	10000	10
Karim	99922	0158058	Banani	karim22@gmail.com	12000	20
Rasel	99933	0171854	Mirpur	rasel33@gmail.com	11000	30

Pre-book:

P_ID	B_DATE	C_ID
96720	12-FEB-23	68731
51824	20-APR-23	68732
27153	26-DEC-22	68733
70182	11-MAR-23	68734

Value insertion:

```
insert into customer values('Habib',68731,0155367,'Mirpur','habib31@gmail.com')
insert into customer values('Siam',68732,0181630,'Dhanmondi','siam32@gmail.com')
insert into customer values('Ripon',68733,0176893,'Khilkhet','ripon33@gmail.com')
insert into customer values('Mamun',68734,0159024,'Uttara','mamun34@gmail.com')

select *
from customer
```

Results Explain Describe Saved SQL History

C_NAME	C_ID	C_NUMBER	C_ADDRESS	C_EMAIL
Habib	68731	155367	Mirpur	habib31@gmail.com
Siam	68732	181630	Dhanmondi	siam32@gmail.com
Ripon	68733	176893	Khilkhet	ripon33@gmail.com
Mamun	68734	159024	Uttara	mamun34@gmail.com

4 rows returned in 0.00 seconds   CSV Export

```
insert into purchase values('Monitor',96720,'ASUS',10000,68731)
insert into purchase values('Mouse',51824,'RAZER',700,68732)
insert into purchase values('Keyboard',27153,'HP',1200,68733)
insert into purchase values('RAM',70182,'SAMSUNG',3000,68734)

select *
from purchase
```

Results Explain Describe Saved SQL History

P_NAME	P_ID	P_BRAND	P_PRICE	C_ID
Monitor	96720	ASUS	10000	68731
Mouse	51824	RAZER	700	68732
Keyboard	27153	HP	1200	68733
RAM	70182	SAMSUNG	3000	68734

4 rows returned in 0.00 seconds   CSV Export

```
insert into product values('Monitor',96720,'ASUS',10000)
insert into product values('Mouse',51824,'RAZER',700)
insert into product values('Keyboard',27153,'HP',1200)
insert into product values('RAM',70182,'SAMSUNG',3000)

select *
from product
```

Results Explain Describe Saved SQL History

P_NAME	P_ID	P_BRAND	P_PRICE
Monitor	96720	ASUS	10000
Mouse	51824	RAZER	700
Keyboard	27153	HP	1200
RAM	70182	SAMSUNG	3000

4 rows returned in 0.02 seconds   CSV Export

```
insert into "MANUFACTURING COMPANY" values('ASUS',00913,8860976,'Taiwan','asus@yahoo.com')
insert into "MANUFACTURING COMPANY" values('RAZER',00651,6523980,'Singapore','razer@yahoo.com')
insert into "MANUFACTURING COMPANY" values('HP',00571,1629035,'USA','hp@gmail.com')
insert into "MANUFACTURING COMPANY" values('SAMSUNG',00465,8216234,'South Korea','samsung@gmail.com')

select *
from "MANUFACTURING COMPANY"
```

Results Explain Describe Saved SQL History

M_NAME	M_LICENSE	M_NUMBER	M_COUNTRV	M_EMAIL
ASUS	913	8860976	Taiwan	asus@yahoo.com
RAZER	651	6523980	Singapore	razer@yahoo.com
HP	571	1629035	USA	hp@gmail.com
SAMSUNG	465	8216234	South Korea	samsung@gmail.com

4 rows returned in 0.00 seconds   CSV Export

```
insert into manufacture values(96720,00913)
insert into manufacture values(51824,00651)
insert into manufacture values(27153,00571)
insert into manufacture values(70182,00465)

select *
from manufacture
```

Results Explain Describe Saved SQL History

P_ID	M_LICENSE
96720	913
51824	651
27153	571
70182	465

4 rows returned in 0.00 seconds   CSV Export

```
insert into employee values('Rahim',99912,0175254,'Mirpur','rahim12@gmail.com',10000)
insert into employee values('Karim',99922,0158058,'Banani','karim22@gmail.com',12000)
insert into employee values('Rasel',99933,0171854,'Mirpur','rasel33@gmail.com',11000)

select *
from employee
```

Results Explain Describe Saved SQL History

E_NAME	E_ID	E_NUMBER	E_ADDRESS	E_EMAIL	E_SALARY
Rahim	99912	175254	Mirpur	rahim12@gmail.com	10000
Karim	99922	158058	Banani	karim22@gmail.com	12000
Rasel	99933	171854	Mirpur	rasel33@gmail.com	11000

3 rows returned in 0.00 seconds   CSV Export

```
insert into manage values('Habib',68731,0155367,'Mirpur','habib31@gmail.com',99912)
insert into manage values('Siam',68732,0181630,'Dhanmondi','siam32@gmail.com',99922)
insert into manage values('Ripon',68733,0176893,'Khilkhet','ripon33@gmail.com',99933)
insert into manage values('Mamun',68734,0159024,'Uttara','mamun34@gmail.com',99912)

select *
from manage
```

Results Explain Describe Saved SQL History

C_NAME	C_ID	C_NUMBER	C_ADDRESS	C_EMAIL	E_ID
Habib	68731	155367	Mirpur	habib31@gmail.com	99912
Siam	68732	181630	Dhanmondi	siam32@gmail.com	99922
Ripon	68733	176893	Khilkhet	ripon33@gmail.com	99933
Mamun	68734	159024	Uttara	mamun34@gmail.com	99912

4 rows returned in 0.00 seconds   CSV Export

```
insert into department values('Sales',10)
insert into department values('Accounts',20)
insert into department values('Managing',30)

select *
from department
```

Results Explain Describe Saved SQL History

D_NAME	D_NUMBER
Sales	10
Accounts	20
Managing	30

3 rows returned in 0.00 seconds   CSV Export

```
insert into belong values('Rahim',99912,0175254,'Mirpur','rahim12@gmail.com',10000,10)
insert into belong values('Karim',99922,0158058,'Banani','karim22@gmail.com',12000,20)
insert into belong values('Rasel',99933,0171854,'Mirpur','rasel33@gmail.com',11000,30)

select *
from belong
```

Results Explain Describe Saved SQL History

E_NAME	E_ID	E_NUMBER	E_ADDRESS	E_EMAIL	E_SALARY	D_NUMBER
Rahim	99912	175254	Mirpur	rahim12@gmail.com	10000	10
Karim	99922	158058	Banani	karim22@gmail.com	12000	20
Rasel	99933	171854	Mirpur	rasel33@gmail.com	11000	30

3 rows returned in 0.00 seconds   CSV Export

```
insert into "PRE-BOOK" values(96720,'12-FEB-23',68731)
insert into "PRE-BOOK" values(51824,'20-APR-23',68732)
insert into "PRE-BOOK" values(27153,'26-DEC-22',68733)
insert into "PRE-BOOK" values(70182,'11-MAR-23',68734)

select *
from "PRE-BOOK"
```

Results   Explain   Describe   Saved SQL   History

P_ID	B_DATE	C_ID
96720	12-FEB-23	68731
51824	20-APR-23	68732
27153	26-DEC-22	68733
70182	11-MAR-23	68734

4 rows returned in 0.00 seconds   CSV Export

Query

Single row sub-query:

1. Show product names whose price is less than the price of monitor.

☑ Autocommit Display 10 ▾

```
select p_name
 from product
  where p_price <
                 ( select p_price
                    from product
                     where p_name = 'Monitor')
```

Results Explain Describe Saved SQL History

P_NAME
Mouse
Keyboard
RAM

3 rows returned in 0.01 seconds   CSV Export

2. Show employee name whose address is not same as Karim's address and whose salary is less than Rasel's salary.

☑ Autocommit Display 10 ▾

```
select e_name
 from employee
  where e_address <>
                   ( select e_address
                      from employee
                       where e_name = 'Karim')
   and e_salary <
                ( select e_salary
                   from employee
                    where e_name = 'Rasel')
```

Results Explain Describe Saved SQL History

E_NAME
Rahim

1 rows returned in 0.00 seconds   CSV Export

Multiple row sub-query:

1. Show salary and employee names whose salary is more than all the employees live in Mirpur.

☑ Autocommit   Display 10 ▾

```
select e_name, e_salary
 from employee
  where e_salary > all
                    ( select e_salary
                        from employee
                          where e_address = 'Mirpur')
```

Results Explain Describe Saved SQL History

E_NAME	E_SALARY
Karim	12000

1 rows returned in 0.00 seconds   CSV Export

2.Show salary and employee names whose salary is more than any the employees live in Mirpur.

☑ Autocommit   Display 10 ▾

```
select e_name, e_salary
 from employee
  where e_salary > any
                    ( select e_salary
                        from employee
                          where e_address = 'Mirpur')
```

Results Explain Describe Saved SQL History

E_NAME	E_SALARY
Karim	12000
Rasel	11000

2 rows returned in 0.00 seconds   CSV Export

Equijoin :

1. Show c_name, c_id, p_name and p_price from customer and purchase table using a joining condition.

☑ Autocommit **Display** 10 ⌄

```
select c.c_name, c.c_id, p.p_name, p.p_price
 from customer c, purchase p
  where c.c_id = p.c_id ;
```

Results Explain Describe Saved SQL History

C_NAME	C_ID	P_NAME	P_PRICE
Habib	68731	Monitor	10000
Siam	68732	Mouse	700
Ripon	68733	Keyboard	1200
Mamun	68734	RAM	3000

4 rows returned in 0.00 seconds   CSV Export

Self-join :

1. Display the message like the salary of e_name is e_salary using self joining.

☑ Autocommit **Display** 10 ⌄

```
select 'The salary of '|| a.e_name || ' is ' || b.e_salary Salary
 from employee a, employee b
  where a.e_id = b.e_id;
```

Results Explain Describe Saved SQL History

SALARY
The salary of Rahim is 10000
The salary of Karim is 12000
The salary of Rasel is 11000

3 rows returned in 0.00 seconds   CSV Export

Aggregate function :

1. Show the lowest salary among the employees who live in Mirpur.

☑ Autocommit Display 10 ▼

```
select min(e_salary)
 from employee
  where e_address = 'Mirpur'|
```

Results Explain Describe Saved SQL History

MIN(E_SALARY)
10000

1 rows returned in 0.00 seconds   CSV Export

Simple view :

1. Create a view showing the names and salaries less than 12000?

☑ Autocommit Display 10 ▼

```
create view e_salaryvu12000
as select e_name, e_salary
from employee
where e_salary<12000
```

Results Explain Describe Saved SQL History

View created.

0.00 seconds

☑ Autocommit Display [10 ⌄]

```
DESCRIBE e_salaryvu12000
```

Results **Explain** **Describe** **Saved SQL** **History**

Object Type **VIEW** Object **E_SALARYVU12000**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
E_SALARYVU12000	E_NAME	Varchar2	100	-	-	-	✓	-	-
	E_SALARY	Number	-	5	0	-	✓	-	-
									1 - 2

☑ Autocommit Display [10 ⌄]

```
select *
from e_salaryvu12000
```

Results **Explain** **Describe** **Saved SQL** **History**

E_NAME	E_SALARY
Rahim	10000
Rasel	11000

2 rows returned in 0.00 seconds   CSV Export

Complex view :

1. Create a view showing customer names and product names from different table?

✔ Autocommit   Display [10   ▾]

```
create view purchasevu as
select c.c_name,p.p_name
from customer c, purchase p
where c.c_id = p.c_id
```

Results   Explain   Describe   Saved SQL   History

View created.

0.00 seconds

✔ Autocommit   Display [10   ▾]

```
describe purchasevu
```

Results   Explain   **Describe**   Saved SQL   History

Object Type **VIEW** Object **PURCHASEVU**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PURCHASEVU	C_NAME	Varchar2	100	-	-	-	✓	-	-
	P_NAME	Varchar2	100	-	-	-	✓	-	-
									1 - 2

☑ Autocommit **Display** 10 ▾

```
select *
from purchasevu
```

Results **Explain** **Describe** **Saved SQL** **History**

C_NAME	P_NAME
Habib	Monitor
Siam	Mouse
Ripon	Keyboard
Mamun	RAM

4 rows returned in 0.00 seconds   CSV Export