QUESTION 1 :

Consider the database for a banking enterprise. Write the queries for the below questions.

(i) Create the following tables

Table	Attributes
Customer	cid,cname,loc,sex,dob
Bank_brn	bcode,bloc,bsate
Deposit	Dacno,dtype,ddate,damt
Loan	Lacno,ltype,ldate,lamt
Accounts_in	Bcode,cid
Depositor	cid,dacno
Borrower	cid,lacno

(ii) Include necessary constraints.
(iii) Tables are created under the database 'bank'
(iv) Display all the tables in bank database
(v) Describe the structure of all tables
(vi) Delete tables

SCRIPTS

(i, ii, iii) Query :

```
CREATE DATABASE bank;

CREATE TABLE customer(
        cid int primary key,
        cname varchar(20),
        loc varchar(20),
        sex varchar(20),
        dob varchar(20));

CREATE TABLE bank_branch(
        bcode varchar(10) primary key,
        bloc varchar(20),
        bstate varchar(20));

CREATE TABLE deposit(
        dacno int primary key,
        dtype varchar(20),
        ddate varchar(20),
        damt varchar(20));

CREATE TABLE loan(
        lacno int primary key,
        ltype varchar(20),
        ldate varchar(20),
        lamt float(20));
```

```
CREATE TABLE accounts_in(
bcode varchar(20) ,
cid int,
FOREIGN KEY (bcode) REFERENCES bank_branch(bcode) ON DELETE CASCADE,
FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE);

CREATE TABLE Depositor(
cid int ,
dacno int,
FOREIGN KEY (dacno) REFERENCES deposit(dacno) ON DELETE CASCADE,
FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE);

CREATE TABLE Borrower(
cid int ,
lacno int,
FOREIGN KEY (lacno) REFERENCES loan(lacno) ON DELETE CASCADE,
FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE);
```

(iv) Query :

```
USE bank;
SHOW TABLES;
```

(v) Query :

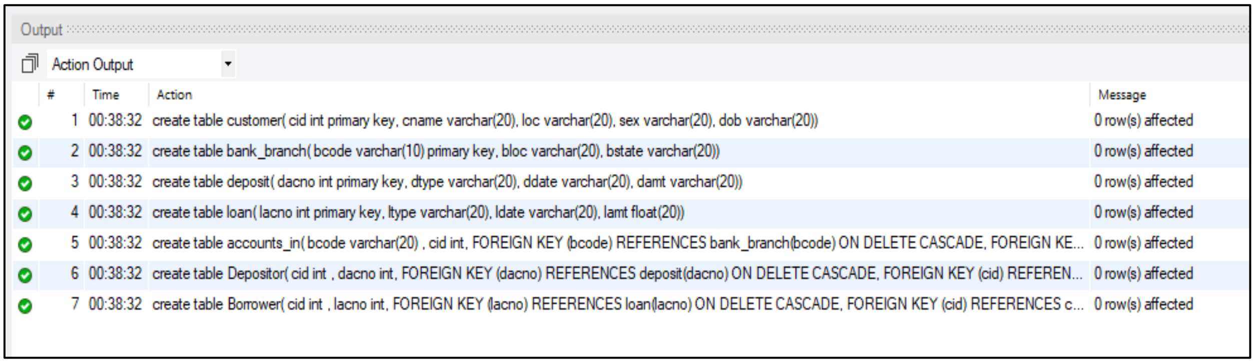
```
DESCRIBE customer;
DESCRIBE bank_branch;
DESCRIBE deposit;
DESCRIBE loan;
DESCRIBE accounts_in;
DESCRIBE Depositor;
DESCRIBE Borrower;
```

(vi) Query :

```
DROP TABLE customer;
DROP TABLE bank_branch;
DROP TABLE deposit;
DROP TABLE loan;
DROP TABLE accounts_in;
DROP TABLE Depositor;
DROP TABLE Borrower;
```

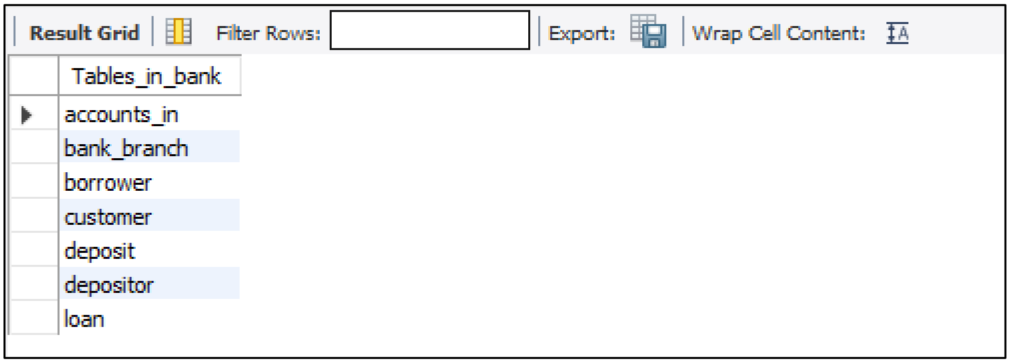
OUTPUT

Output of (i ,ii ,iii)



	#	Time	Action	Message
✓	1	00:38:32	create table customer(cid int primary key, cname varchar(20), loc varchar(20), sex varchar(20), dob varchar(20))	0 row(s) affected
✓	2	00:38:32	create table bank_branch(bcode varchar(10) primary key, bloc varchar(20), bstate varchar(20))	0 row(s) affected
✓	3	00:38:32	create table deposit(dacno int primary key, dtype varchar(20), ddate varchar(20), damt varchar(20))	0 row(s) affected
✓	4	00:38:32	create table loan(lacno int primary key, ltype varchar(20), ldate varchar(20), lamt float(20))	0 row(s) affected
✓	5	00:38:32	create table accounts_in(bcode varchar(20) , cid int, FOREIGN KEY (bcode) REFERENCES bank_branch(bcode) ON DELETE CASCADE, FOREIGN KE...	0 row(s) affected
✓	6	00:38:32	create table Depositor(cid int , dacno int, FOREIGN KEY (dacno) REFERENCES deposit(dacno) ON DELETE CASCADE, FOREIGN KEY (cid) REFEREN...	0 row(s) affected
✓	7	00:38:32	create table Borrower(cid int , lacno int, FOREIGN KEY (lacno) REFERENCES loan(lacno) ON DELETE CASCADE, FOREIGN KEY (cid) REFERENCES c...	0 row(s) affected

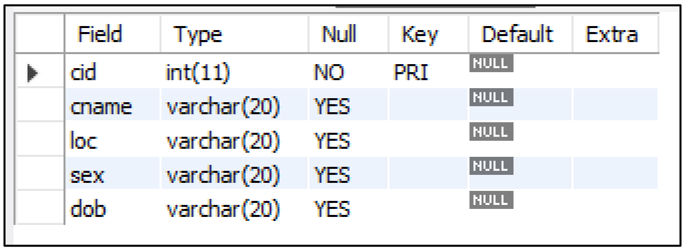
Output of (iv)



Tables_in_bank
accounts_in
bank_branch
borrower
customer
deposit
depositor
loan

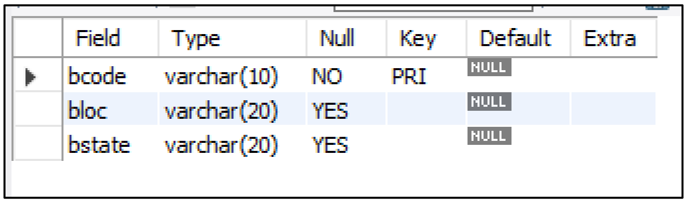
Output of (v)

customer



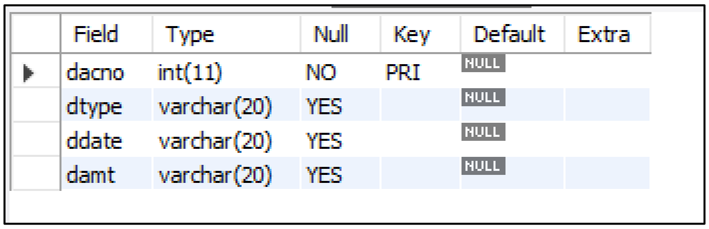
Field	Type	Null	Key	Default	Extra
cid	int(11)	NO	PRI	NULL	
cname	varchar(20)	YES		NULL	
loc	varchar(20)	YES		NULL	
sex	varchar(20)	YES		NULL	
dob	varchar(20)	YES		NULL	

bank_branch



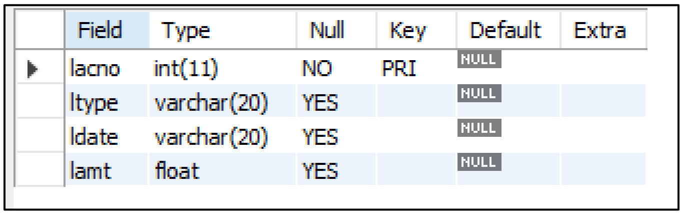
Field	Type	Null	Key	Default	Extra
bcode	varchar(10)	NO	PRI	NULL	
bloc	varchar(20)	YES		NULL	
bstate	varchar(20)	YES		NULL	

deposit



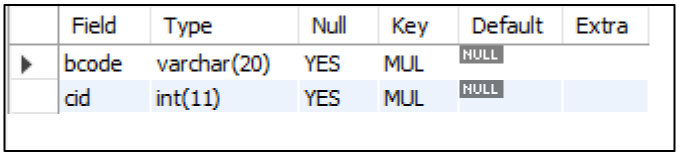
	Field	Type	Null	Key	Default	Extra
▶	dacno	int(11)	NO	PRI	NULL	
	dtype	varchar(20)	YES		NULL	
	ddate	varchar(20)	YES		NULL	
	damt	varchar(20)	YES		NULL	

Loan



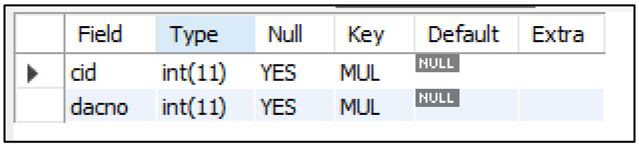
	Field	Type	Null	Key	Default	Extra
▶	lacno	int(11)	NO	PRI	NULL	
	ltype	varchar(20)	YES		NULL	
	ldate	varchar(20)	YES		NULL	
	lamt	float	YES		NULL	

accounts_in



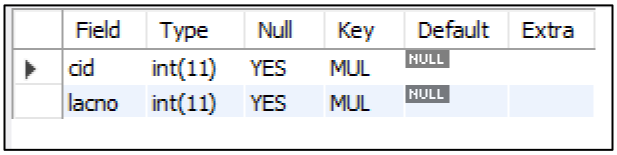
	Field	Type	Null	Key	Default	Extra
▶	bcode	varchar(20)	YES	MUL	NULL	
	cid	int(11)	YES	MUL	NULL	

Depositor



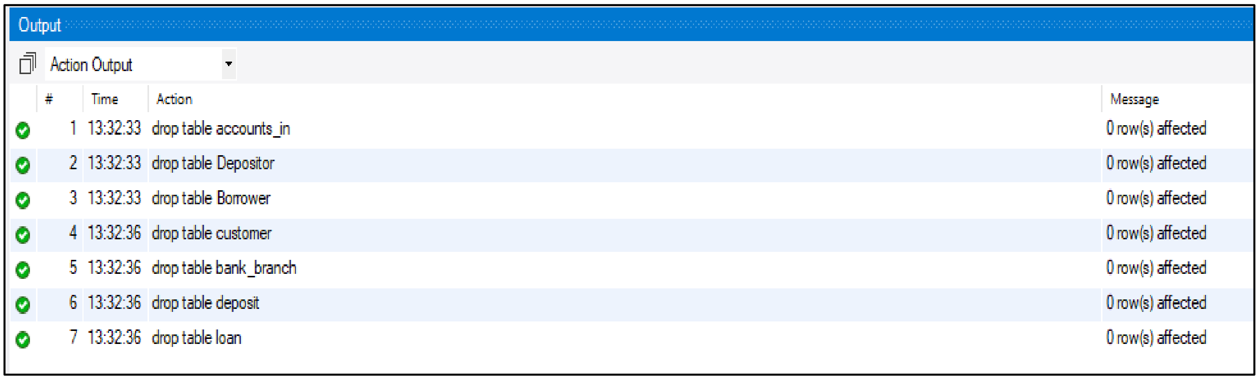
	Field	Type	Null	Key	Default	Extra
▶	cid	int(11)	YES	MUL	NULL	
	dacno	int(11)	YES	MUL	NULL	

Borrower



	Field	Type	Null	Key	Default	Extra
▶	cid	int(11)	YES	MUL	NULL	
	lacno	int(11)	YES	MUL	NULL	

Output of (vi)



	#	Time	Action	Message
✓	1	13:32:33	drop table accounts_in	0 row(s) affected
✓	2	13:32:33	drop table Depositor	0 row(s) affected
✓	3	13:32:33	drop table Borrower	0 row(s) affected
✓	4	13:32:36	drop table customer	0 row(s) affected
✓	5	13:32:36	drop table bank_branch	0 row(s) affected
✓	6	13:32:36	drop table deposit	0 row(s) affected
✓	7	13:32:36	drop table loan	0 row(s) affected

QUESTION 2 :

Consider the database for a college. Write the query for the following.

- **(i)** Insert at least 5 tuples into each table.
- **(ii)** List the details of students in the ascending order of date of birth
- **(iii)** Display the details of students from computer department
- **(iv)** List the faculties in the descending order of salary
- **(v)** Display the total number of students in each department
- **(vi)** Display the total number of faculties in each department with salary greater than 25000

Answers

Creating required tables

```
CREATE TABLE dept(
id int primary key,
name varchar(20));

CREATE TABLE student(
id int primary key,
name varchar(20),
dept_id int ,
dob date,
FOREIGN KEY (dept_id) REFERENCES dept(id) ON DELETE CASCADE);

CREATE TABLE faculty(
name varchar(20),
id int primary key,
salary double,
dept_id int,
FOREIGN KEY (dept_id) REFERENCES dept(id) ON DELETE CASCADE);
```

(i) Query :

```
INSERT INTO dept VALUES(1,'CS');
INSERT INTO dept VALUES(2,'ECE');
INSERT INTO dept VALUES(3,'Physics');
INSERT INTO dept VALUES(4,'Chemistry');
INSERT INTO dept VALUES(5,'Maths');

INSERT INTO student VALUES(10,'Marvin',1,'2001-01-01');
INSERT INTO student VALUES(11,'Shibili',1,'2000-01-02');
INSERT INTO student VALUES(12,'Soni',3,'2002-01-03');
INSERT INTO student VALUES(13,'Joyal',4,'1999-01-04');
INSERT INTO student VALUES(14,'Jeslin',2,'1998-01-05');
INSERT INTO faculty VALUES('Mike',101,12000,1);
INSERT INTO faculty VALUES('Sam',102,23000,1);
INSERT INTO faculty VALUES('Ethan',103,45000,3);
INSERT INTO faculty VALUES('Ross',104,50000,2);
INSERT INTO faculty VALUES('Rachel',105,120000,4);
```

(ii) Query :

```
SELECT *
FROM student
ORDER BY dob;
```

(iii) Query :

```
SELECT *
FROM student s JOIN dept d
        ON s.dept_id=d.id
WHERE d.name="CS";
```

(iv) Query :

```
SELECT *
FROM faculty
ORDER BY salary DESC;
```

(v) Query :

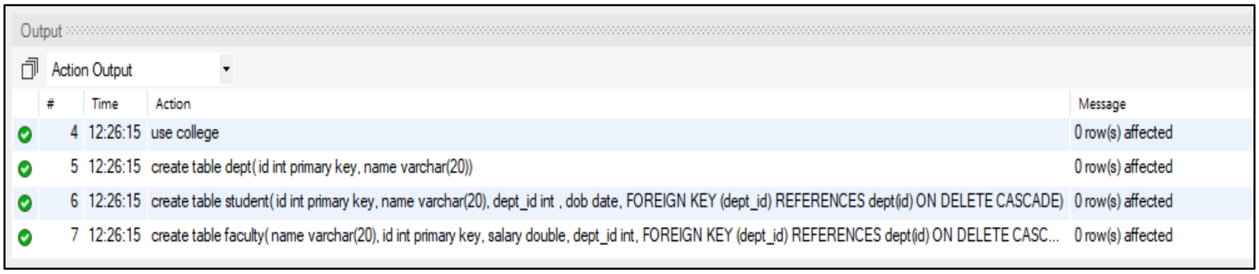
```
SELECT d.name, count(s.id) 'student'
FROM student s JOIN dept d
        ON s.dept_id=d.id
GROUP BY d.id;
```

(vi) Query :

```
SELECT d.name,count(f.id) 'faculty'
FROM faculty f JOIN dept d
        ON f.dept_id=d.id
where f.salary>25000
```

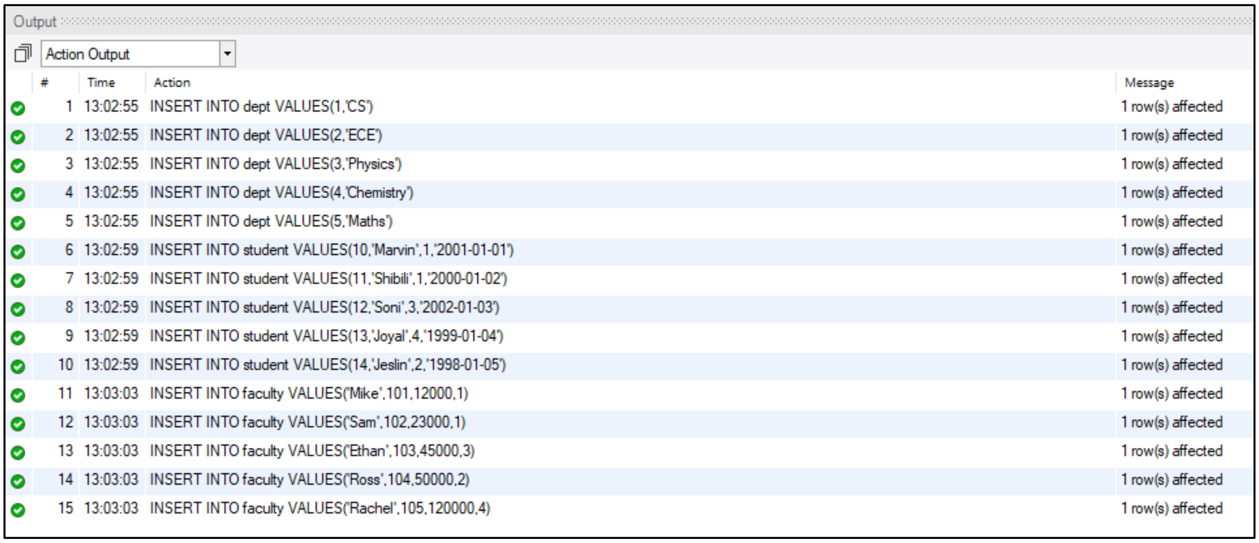
OUTPUT

Output of table creation



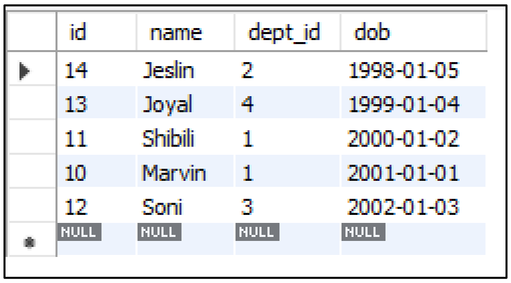
```
   #  Time      Action                                                                                                                              Message
   4  12:26:15  use college                                                                                                                         0 row(s) affected
   5  12:26:15  create table dept( id int primary key, name varchar(20))                                                                            0 row(s) affected
   6  12:26:15  create table student( id int primary key, name varchar(20), dept_id int , dob date, FOREIGN KEY (dept_id) REFERENCES dept(id) ON DELETE CASCADE)  0 row(s) affected
   7  12:26:15  create table faculty( name varchar(20), id int primary key, salary double, dept_id int, FOREIGN KEY (dept_id) REFERENCES dept(id) ON DELETE CASC...  0 row(s) affected
```

Output of (i)



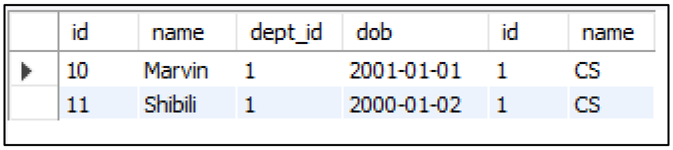
```
   #  Time      Action                                                     Message
   1  13:02:55  INSERT INTO dept VALUES(1,'CS')                            1 row(s) affected
   2  13:02:55  INSERT INTO dept VALUES(2,'ECE')                           1 row(s) affected
   3  13:02:55  INSERT INTO dept VALUES(3,'Physics')                       1 row(s) affected
   4  13:02:55  INSERT INTO dept VALUES(4,'Chemistry')                     1 row(s) affected
   5  13:02:55  INSERT INTO dept VALUES(5,'Maths')                         1 row(s) affected
   6  13:02:59  INSERT INTO student VALUES(10,'Marvin',1,'2001-01-01')     1 row(s) affected
   7  13:02:59  INSERT INTO student VALUES(11,'Shibili',1,'2000-01-02')    1 row(s) affected
   8  13:02:59  INSERT INTO student VALUES(12,'Soni',3,'2002-01-03')       1 row(s) affected
   9  13:02:59  INSERT INTO student VALUES(13,'Joyal',4,'1999-01-04')      1 row(s) affected
  10  13:02:59  INSERT INTO student VALUES(14,'Jeslin',2,'1998-01-05')     1 row(s) affected
  11  13:03:03  INSERT INTO faculty VALUES('Mike',101,12000,1)             1 row(s) affected
  12  13:03:03  INSERT INTO faculty VALUES('Sam',102,23000,1)              1 row(s) affected
  13  13:03:03  INSERT INTO faculty VALUES('Ethan',103,45000,3)            1 row(s) affected
  14  13:03:03  INSERT INTO faculty VALUES('Ross',104,50000,2)             1 row(s) affected
  15  13:03:03  INSERT INTO faculty VALUES('Rachel',105,120000,4)          1 row(s) affected
```

Output of (ii)



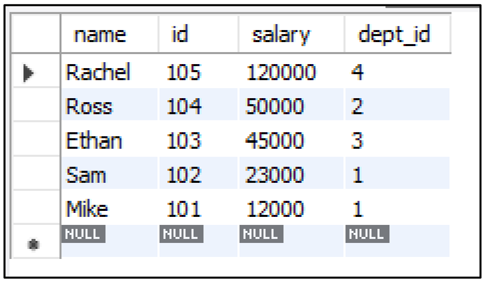
id	name	dept_id	dob
14	Jeslin	2	1998-01-05
13	Joyal	4	1999-01-04
11	Shibili	1	2000-01-02
10	Marvin	1	2001-01-01
12	Soni	3	2002-01-03
NULL	NULL	NULL	NULL

Output of (iii)



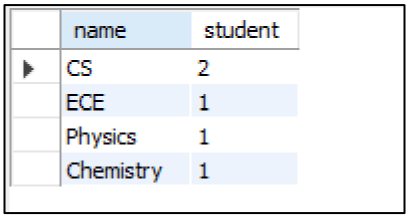
id	name	dept_id	dob	id	name
10	Marvin	1	2001-01-01	1	CS
11	Shibili	1	2000-01-02	1	CS

Output of (iv)



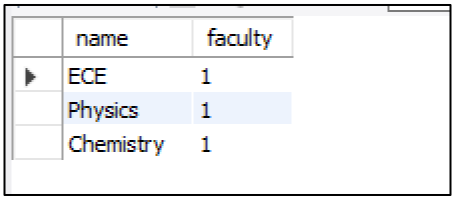
name	id	salary	dept_id
Rachel	105	120000	4
Ross	104	50000	2
Ethan	103	45000	3
Sam	102	23000	1
Mike	101	12000	1
NULL	NULL	NULL	NULL

Output of (v)



name	student
CS	2
ECE	1
Physics	1
Chemistry	1

Output of (vi)



name	faculty
ECE	1
Physics	1
Chemistry	1

QUESTION 3 :

Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by,Order by,Having.

E_ID	E_NAME	AGE	SALARY
101	ANU	22	9000
102	Shane	29	8000
103	Rohan	34	6000
104	Scott	44	10000
105	Tiger	35	8000
106	Alex	27	7000
107	Abhi	29	8000

(i) Create Employee table containing all Records.

(ii) Count number of employee names from employee table.

(iii) Find the Maximum age from employee table

(iv) Find the Minimum age from employee table.

(v) Display the Sum of age employee table.

(vi) Display the Average of age from Employee table

(vii) Create a View for age in employee table

(viii) Display views

(ix) Find grouped salaries of employees.

(x) Find salaries of employee in Ascending Order

(xi) Find salaries of employee in Descending Order

Script:

(i) Query :

```
CREATE DATABASE EMPLOYEE;
USE EMPLOYEE;
CREATE TABLE Employee(
E_ID INT(25),
E_NAME VARCHAR(25),
AGE INT(10),
SALARY FLOAT(10),
PRIMARY KEY(E_ID));
```

Inserting rows :

```
INSERT INTO Employee VALUES (101,"ANU",22,9000);
INSERT INTO Employee VALUES (102,"SHANE",29,8000);
INSERT INTO Employee VALUES (103,"ROHAN",34,6000);
INSERT INTO Employee VALUES (104,"SCOTT",44,10000);
INSERT INTO Employee VALUES (105,"TIGER",35,8000);
INSERT INTO Employee VALUES (106,"ALEX",27,7000);
INSERT INTO Employee VALUES (107,"ABHI",29,8000);
```

(ii) Query :

```
SELECT COUNT(E_NAME) FROM Employee;
```

(iii) Query :

```
SELECT MAX(AGE) FROM Employee;
```

(iv) Query :

```
SELECT MIN(AGE) FROM Employee;
```

(v) Query :

```
SELECT SUM(AGE) FROM Employee;
```

(vi) Query :

```
SELECT AVG(AGE) FROM Employee;
```

(vii) Query :

```
CREATE VIEW E_AGE AS
SELECT E_NAME,AGE FROM Employee;
```

(viii) Query :

```
SELECT * FROM E_AGE;
```

(ix) Query :

```
SELECT E_NAME,SALARY FROM Employee GROUP BY E_NAME;
```

(x) Query :

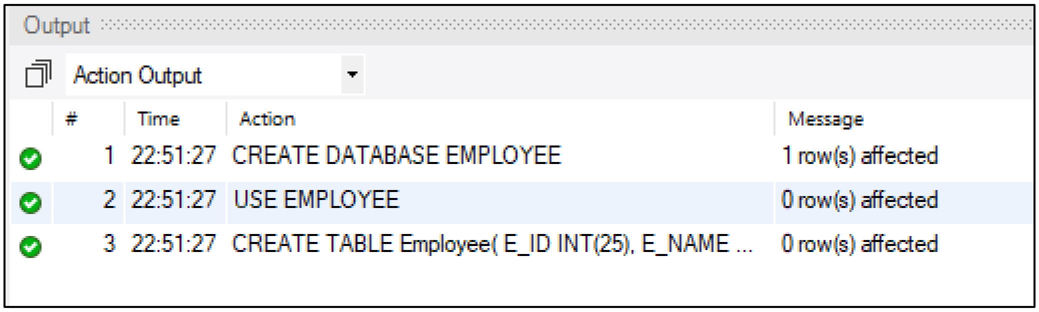
```
SELECT SALARY FROM Employee ORDER BY SALARY ASC;
```

(xi) Query :

```
SELECT SALARY FROM Employee ORDER BY SALARY DESC;
```

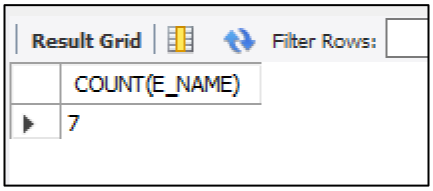
OUTPUT

Output of (i)



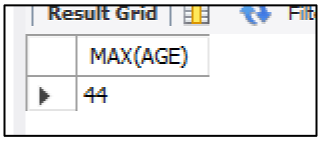
```
    #  Time      Action                                             Message
    1  22:51:27  CREATE DATABASE EMPLOYEE                           1 row(s) affected
    2  22:51:27  USE EMPLOYEE                                       0 row(s) affected
    3  22:51:27  CREATE TABLE Employee( E_ID INT(25), E_NAME ...    0 row(s) affected
```

Output of (ii)



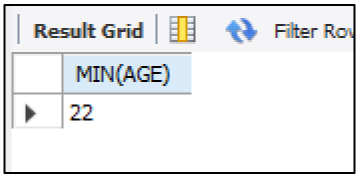
```
COUNT(E_NAME)
7
```

Output of (iii)



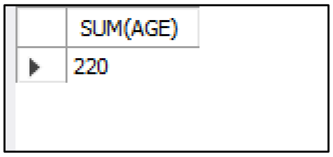
```
MAX(AGE)
44
```

Output of (iv)



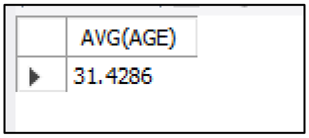
```
MIN(AGE)
22
```

Output of (v)



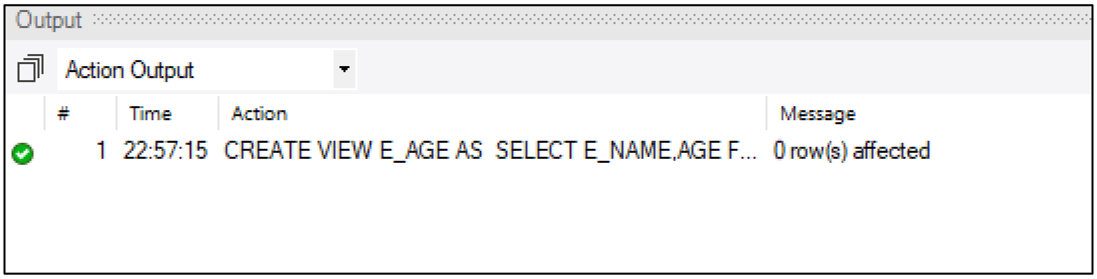
```
SUM(AGE)
220
```

Output of (vi)



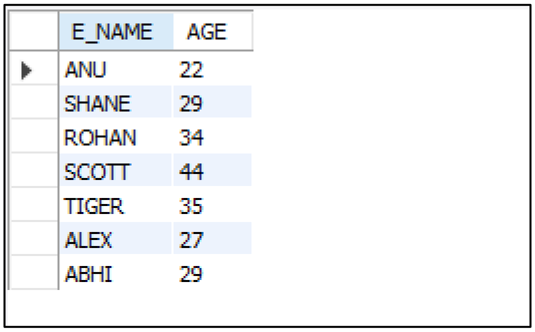
```
AVG(AGE)
31.4286
```

Output of (vii)



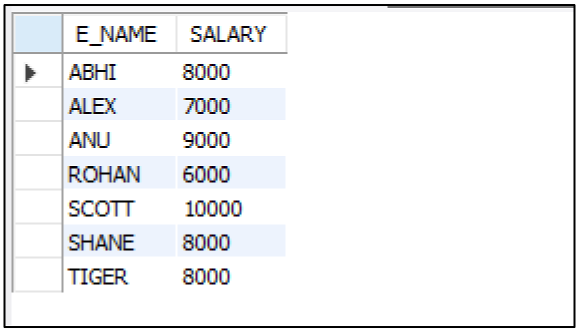
#	Time	Action	Message
1	22:57:15	CREATE VIEW E_AGE AS SELECT E_NAME,AGE F...	0 row(s) affected

Output of (viii)



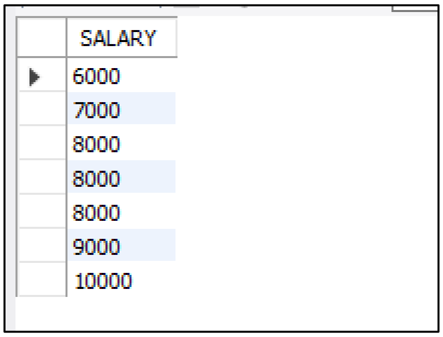
E_NAME	AGE
ANU	22
SHANE	29
ROHAN	34
SCOTT	44
TIGER	35
ALEX	27
ABHI	29

Output of (ix)



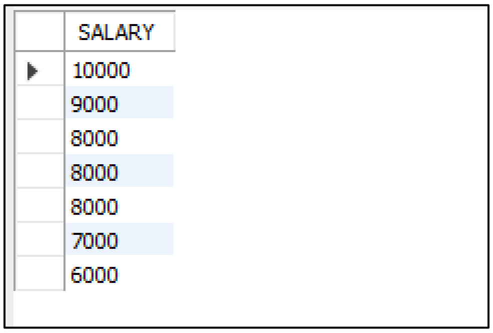
E_NAME	SALARY
ABHI	8000
ALEX	7000
ANU	9000
ROHAN	6000
SCOTT	10000
SHANE	8000
TIGER	8000

Output of (x)



SALARY
6000
7000
8000
8000
8000
9000
10000

Output of (xi)



SALARY
10000
9000
8000
8000
8000
7000
6000