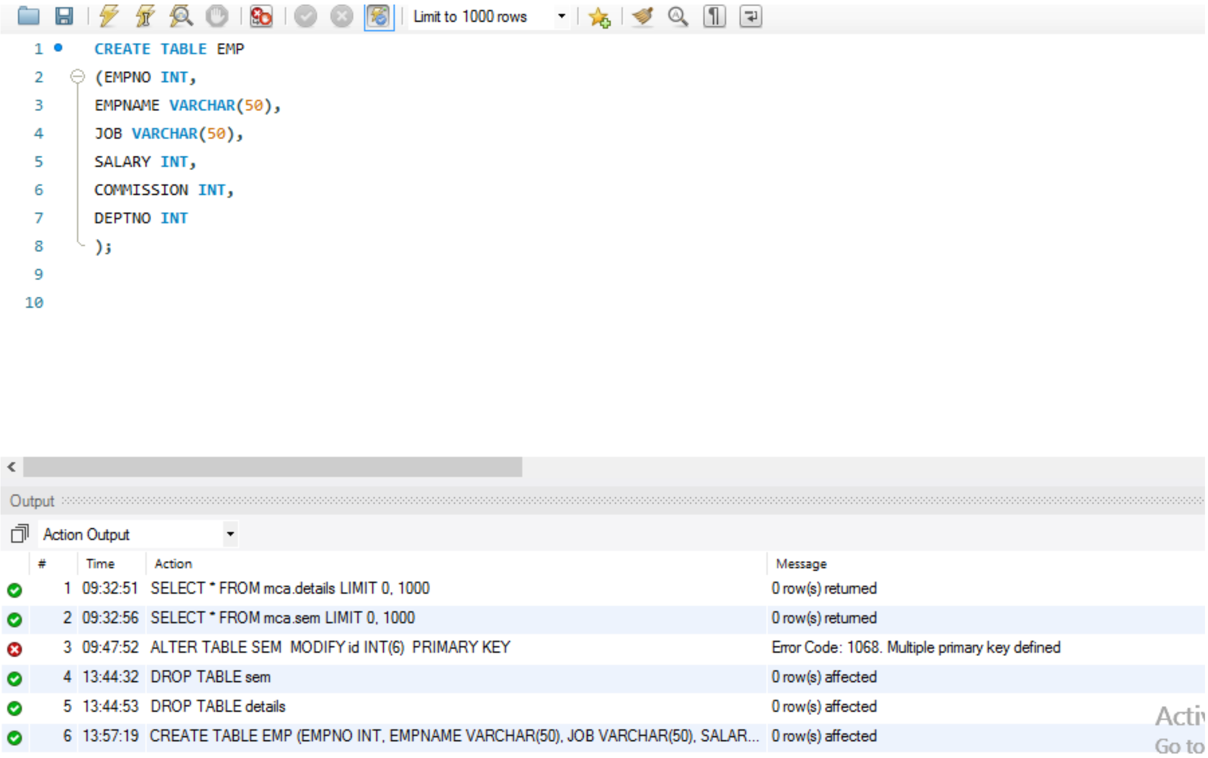
LAB CYCLE 1

QUESTION SET 1

1. Create an employee table ‘EMP’ with following fields :

empno	NUMBER(2)
ename	VARCHAR2(25)
job	VARCHAR2(12)
salary	NUMBER(10,2)
commission	NUMBER(7,2)
deptno	NUMBER(2)

OUTPUT

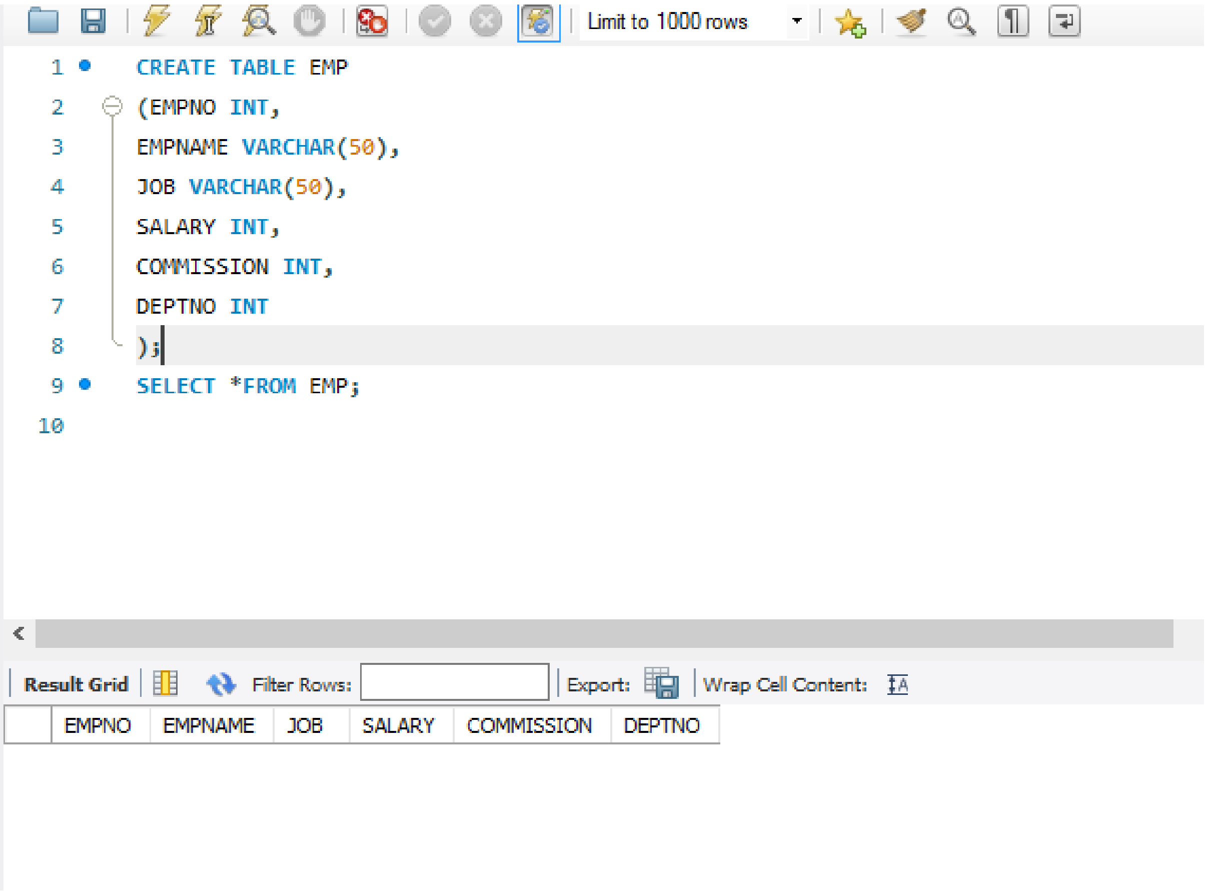


```
CREATE TABLE EMP
(EMPNO INT,
EMPNAME VARCHAR(50),
JOB VARCHAR(50),
SALARY INT,
COMMISSION INT,
DEPTNO INT
);
```

	#	Time	Action	Message
✓	1	09:32:51	SELECT * FROM mca.details LIMIT 0, 1000	0 row(s) returned
✓	2	09:32:56	SELECT * FROM mca.sem LIMIT 0, 1000	0 row(s) returned
☒	3	09:47:52	ALTER TABLE SEM MODIFY id INT(6) PRIMARY KEY	Error Code: 1068. Multiple primary key defined
✓	4	13:44:32	DROP TABLE sem	0 row(s) affected
✓	5	13:44:53	DROP TABLE details	0 row(s) affected
✓	6	13:57:19	CREATE TABLE EMP (EMPNO INT, EMPNAME VARCHAR(50), JOB VARCHAR(50), SALAR...	0 row(s) affected

2. Display the structure of ‘EMP’

OUTPUT



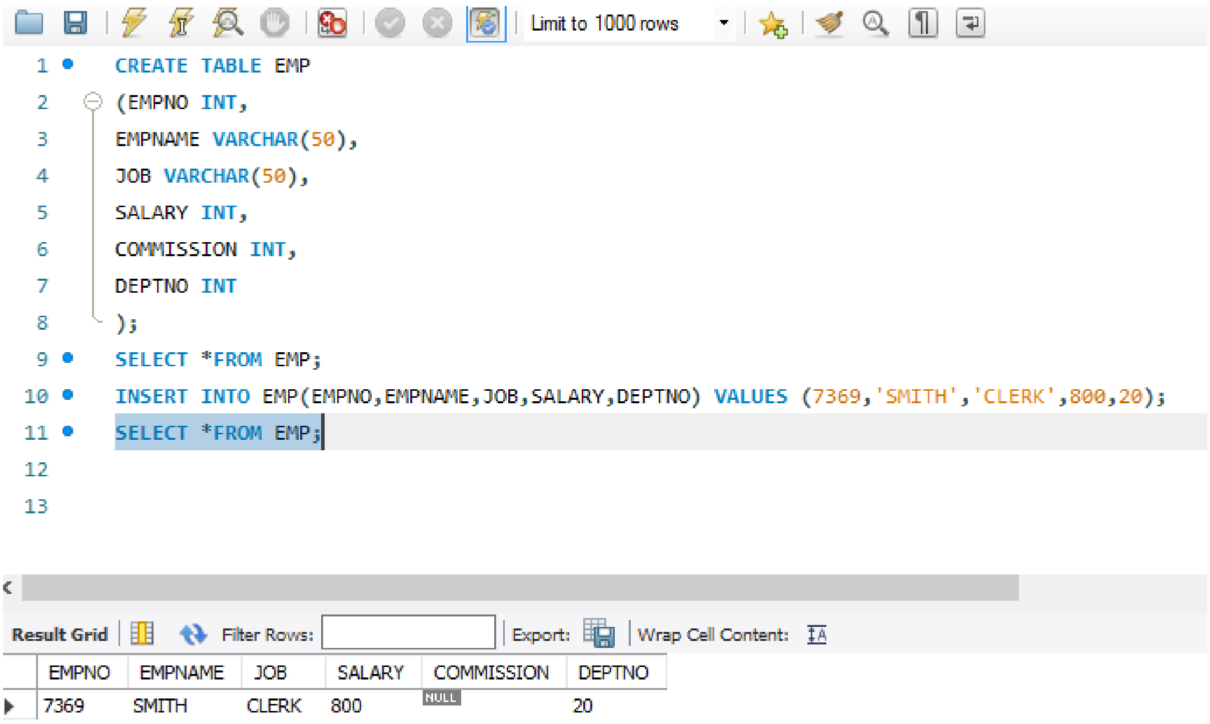
```sql
CREATE TABLE EMP
(EMPNO INT,
EMPNAME VARCHAR(50),
JOB VARCHAR(50),
SALARY INT,
COMMISSION INT,
DEPTNO INT
);
SELECT *FROM EMP;
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO

3. Insert the following record into ‘EMP’

EMPNO	ENAME	JOB	SAL	COMM	DEPTNO
7369	SMITH	CLERK	800		20

OUTPUT



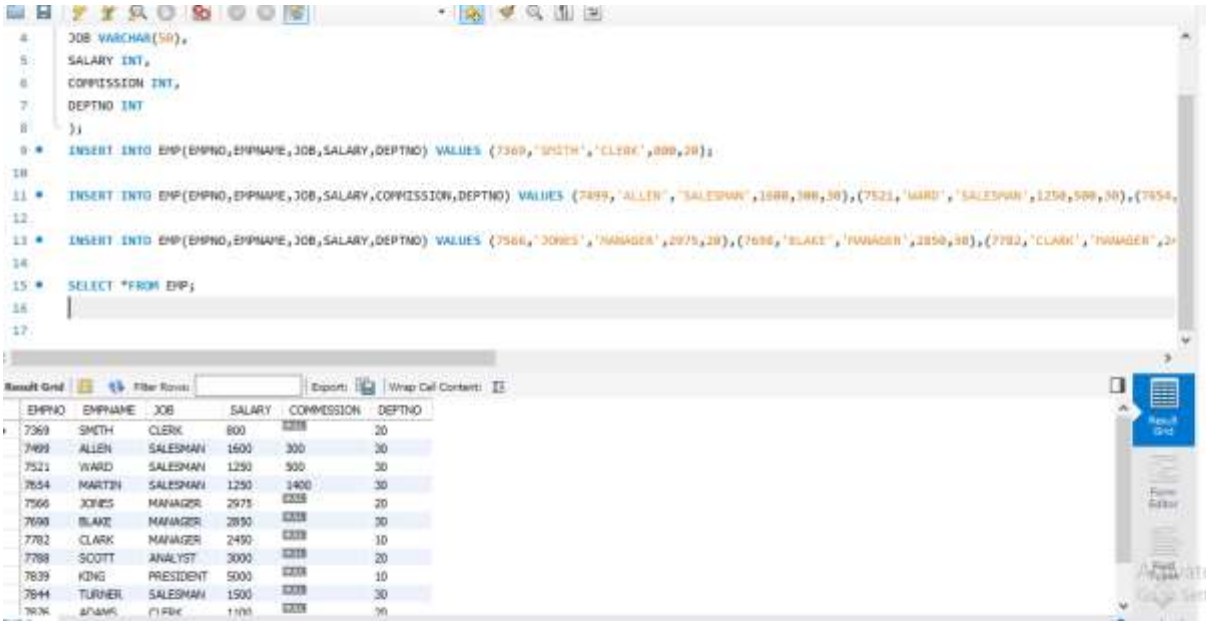
```sql
CREATE TABLE EMP
(EMPNO INT,
EMPNAME VARCHAR(50),
JOB VARCHAR(50),
SALARY INT,
COMMISSION INT,
DEPTNO INT
);
SELECT *FROM EMP;
INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,DEPTNO) VALUES (7369,'SMITH','CLERK',800,20);
SELECT *FROM EMP;
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO
7369	SMITH	CLERK	800	NULL	20

4. Insert the rest of records using substitution variable.

EMPNO	ENAME	JOB	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	1600	300	30
7521	WARD	SALESMAN	1250	500	30
7566	JONES	MANAGER	2975		20
7654	MARTIN	SALESMAN	1250	1400	30
7698	BLAKE	MANAGER	2850		30
7782	CLARK	MANAGER	2450		10
7788	SCOTT	ANALYST	3000		20
7839	KING	PRESIDENT	5000		10
7844	TURNER	SALESMAN	1500		30
7876	ADAMS	CLERK	1100		20
7900	JAMES	NULL	950		30
7902	FORD	ANALYST	3000		20
7934	MILLER	CLERK	1300		10

OUTPUT

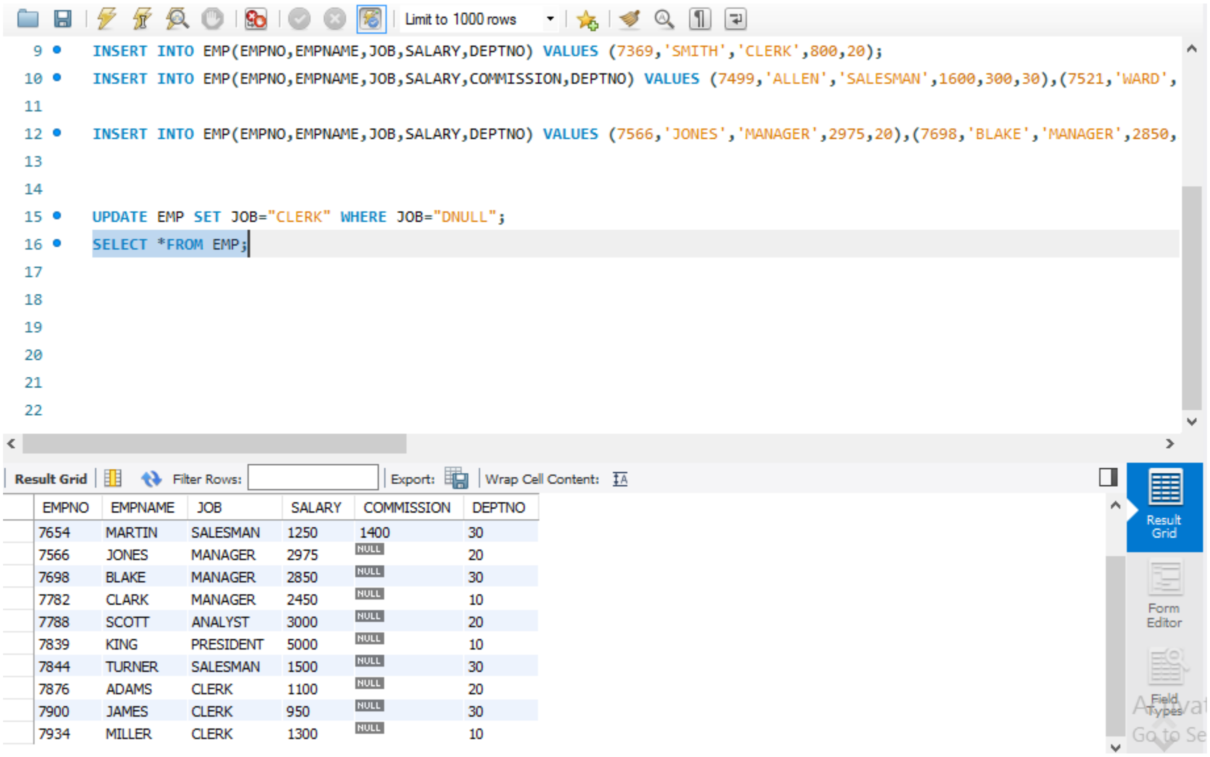


```
JOB VARCHAR(50),
SALARY INT,
COMMISSION INT,
DEPTNO INT
);
INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,DEPTNO) VALUES (7369,'SMITH','CLERK',800,20);

INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,COMMISSION,DEPTNO) VALUES (7499,'ALLEN','SALESMAN',1600,300,30),(7521,'WARD','SALESMAN',1250,500,30),(7654,

INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,DEPTNO) VALUES (7566,'JONES','MANAGER',2975,20),(7698,'BLAKE','MANAGER',2850,30),(7782,'CLARK','MANAGER',2

SELECT *FROM EMP;
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO
7369	SMITH	CLERK	800	NULL	20
7499	ALLEN	SALESMAN	1600	300	30
7521	WARD	SALESMAN	1250	500	30
7654	MARTIN	SALESMAN	1250	1400	30
7566	JONES	MANAGER	2975	NULL	20
7698	BLAKE	MANAGER	2850	NULL	30
7782	CLARK	MANAGER	2450	NULL	10
7788	SCOTT	ANALYST	3000	NULL	20
7839	KING	PRESIDENT	5000	NULL	10
7844	TURNER	SALESMAN	1500	NULL	30
7876	ADAMS	CLERK	1100	NULL	20

5. Insert job as ‘CLERK’ for all ‘NULL’ job types.

OUTPUT



```
INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,DEPTNO) VALUES (7369,'SMITH','CLERK',800,20);
INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,COMMISSION,DEPTNO) VALUES (7499,'ALLEN','SALESMAN',1600,300,30),(7521,'WARD',

INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,DEPTNO) VALUES (7566,'JONES','MANAGER',2975,20),(7698,'BLAKE','MANAGER',2850,


UPDATE EMP SET JOB="CLERK" WHERE JOB="DNULL";
SELECT *FROM EMP;
```

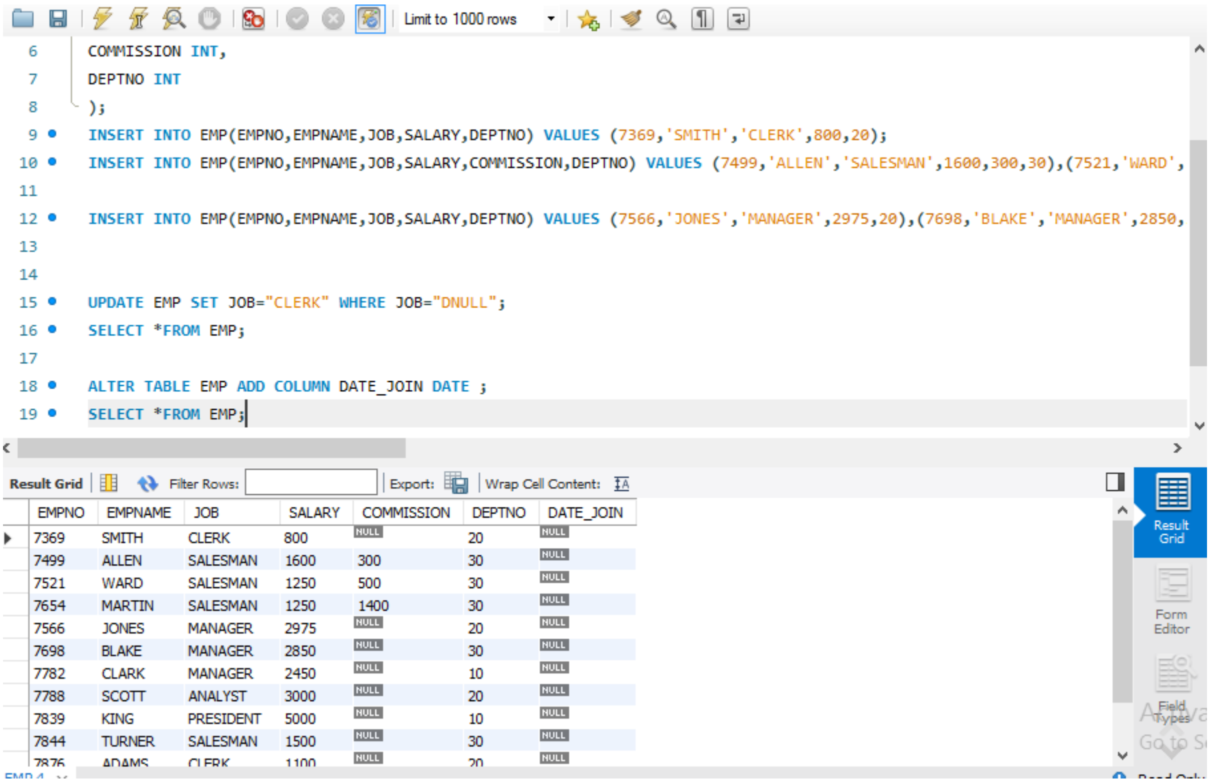
EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO
7654	MARTIN	SALESMAN	1250	1400	30
7566	JONES	MANAGER	2975	NULL	20
7698	BLAKE	MANAGER	2850	NULL	30
7782	CLARK	MANAGER	2450	NULL	10
7788	SCOTT	ANALYST	3000	NULL	20
7839	KING	PRESIDENT	5000	NULL	10
7844	TURNER	SALESMAN	1500	NULL	30
7876	ADAMS	CLERK	1100	NULL	20
7900	JAMES	CLERK	950	NULL	30
7934	MILLER	CLERK	1300	NULL	10

6. Add a new field ‘date_join’ with following values

```
date_join
17-DEC-80
20-FEB-81
22-FEB-81
02-APR-81
28-SEP-81
01-MAY-81
09-JUN-81
```

```
19-APR-87
17-NOV-81
08-SEP-81
23-MAY-87
03-DEC-81
03-DEC-81
23-JAN-82
```

OUTPUT

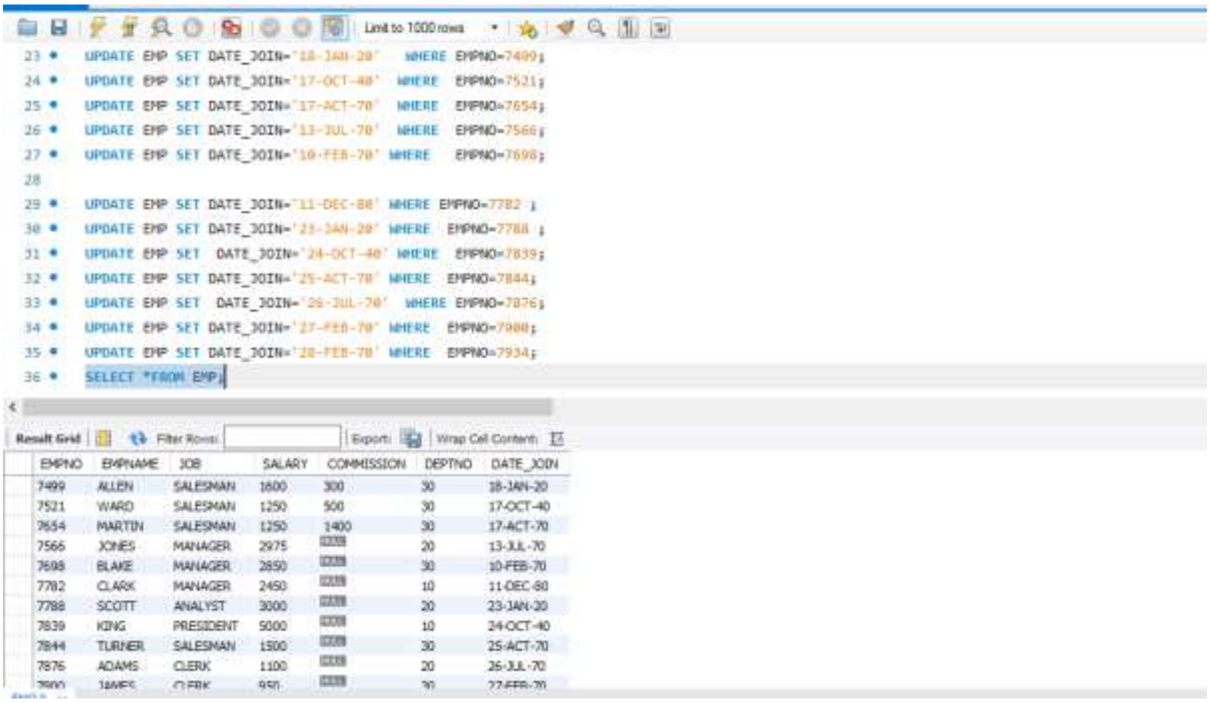


```
COMMISSION INT,
DEPTNO INT
);
INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,DEPTNO) VALUES (7369,'SMITH','CLERK',800,20);
INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,COMMISSION,DEPTNO) VALUES (7499,'ALLEN','SALESMAN',1600,300,30),(7521,'WARD',

INSERT INTO EMP(EMPNO,EMPNAME,JOB,SALARY,DEPTNO) VALUES (7566,'JONES','MANAGER',2975,20),(7698,'BLAKE','MANAGER',2850,


UPDATE EMP SET JOB="CLERK" WHERE JOB="DNULL";
SELECT *FROM EMP;

ALTER TABLE EMP ADD COLUMN DATE_JOIN DATE ;
SELECT *FROM EMP;
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO	DATE_JOIN
7369	SMITH	CLERK	800	NULL	20	NULL
7499	ALLEN	SALESMAN	1600	300	30	NULL
7521	WARD	SALESMAN	1250	500	30	NULL
7654	MARTIN	SALESMAN	1250	1400	30	NULL
7566	JONES	MANAGER	2975	NULL	20	NULL
7698	BLAKE	MANAGER	2850	NULL	30	NULL
7782	CLARK	MANAGER	2450	NULL	10	NULL
7788	SCOTT	ANALYST	3000	NULL	20	NULL
7839	KING	PRESIDENT	5000	NULL	10	NULL
7844	TURNER	SALESMAN	1500	NULL	30	NULL
7876	ADAMS	CLERK	1100	NULL	20	NULL

7. Display details of all employees.

OUTPUT

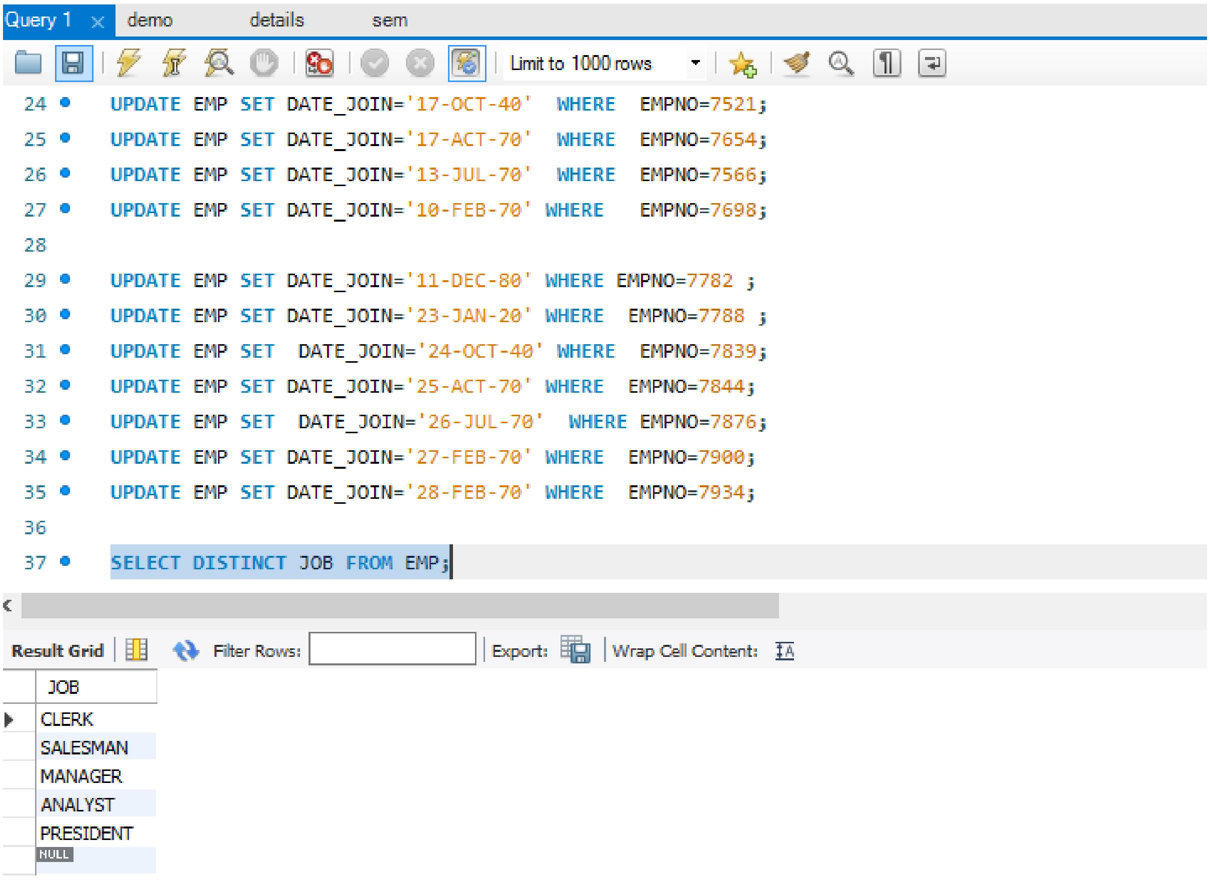


```
UPDATE EMP SET DATE_JOIN='18-JAN-20'   WHERE EMPNO=7499;
UPDATE EMP SET DATE_JOIN='17-OCT-40'  WHERE  EMPNO=7521;
UPDATE EMP SET DATE_JOIN='17-ACT-70'  WHERE  EMPNO=7654;
UPDATE EMP SET DATE_JOIN='13-JUL-70'  WHERE  EMPNO=7566;
UPDATE EMP SET DATE_JOIN='10-FEB-70' WHERE   EMPNO=7698;

UPDATE EMP SET DATE_JOIN='11-DEC-80' WHERE EMPNO=7782 ;
UPDATE EMP SET DATE_JOIN='23-JAN-20' WHERE  EMPNO=7788 ;
UPDATE EMP SET  DATE_JOIN='24-OCT-40' WHERE  EMPNO=7839;
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;
SELECT *FROM EMP;
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO	DATE_JOIN
7499	ALLEN	SALESMAN	1600	300	30	18-JAN-20
7521	WARD	SALESMAN	1250	500	30	17-OCT-40
7654	MARTIN	SALESMAN	1250	1400	30	17-ACT-70
7566	JONES	MANAGER	2975	NULL	20	13-JUL-70
7698	BLAKE	MANAGER	2850	NULL	30	10-FEB-70
7782	CLARK	MANAGER	2450	NULL	10	11-DEC-80
7788	SCOTT	ANALYST	3000	NULL	20	23-JAN-20
7839	KING	PRESIDENT	5000	NULL	10	24-OCT-40
7844	TURNER	SALESMAN	1500	NULL	30	25-ACT-70
7876	ADAMS	CLERK	1100	NULL	20	26-JUL-70
7900	JAMES	CLERK	950	NULL	30	27-FEB-70

8. Display all the distinct job types in ‘EMP’.

OUTPUT

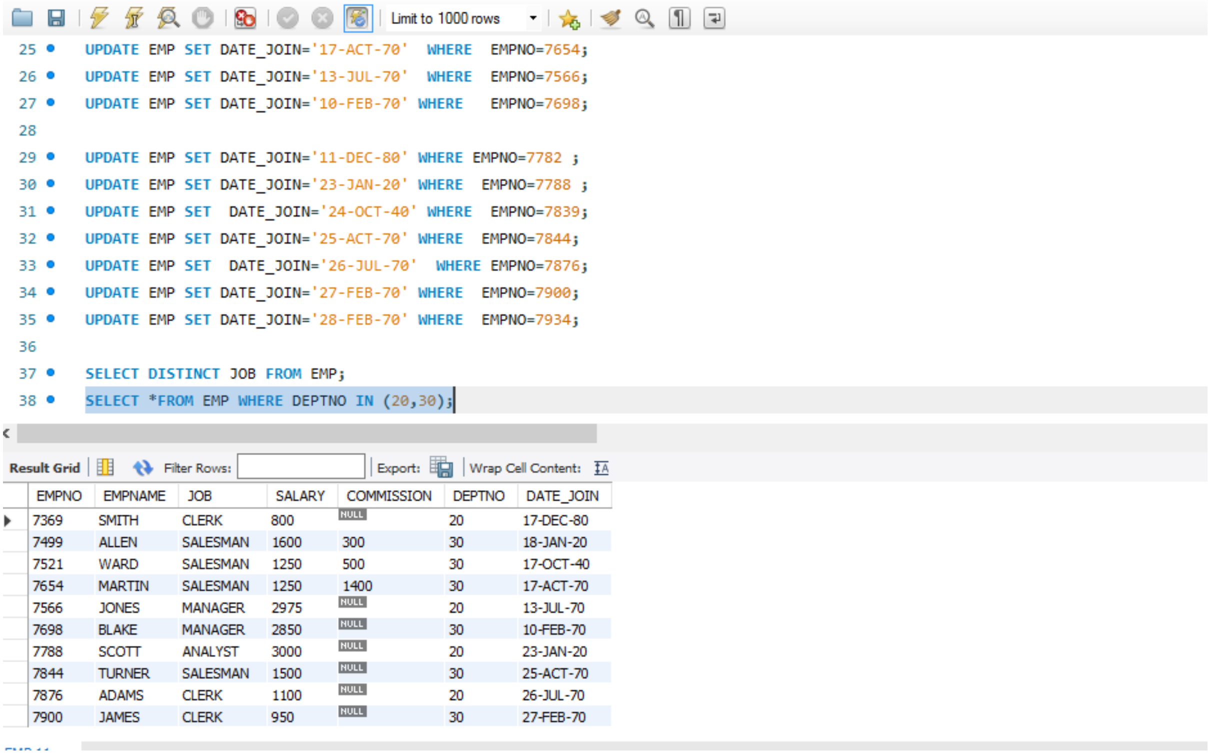


```
UPDATE EMP SET DATE_JOIN='17-OCT-40'  WHERE  EMPNO=7521;
UPDATE EMP SET DATE_JOIN='17-ACT-70'  WHERE  EMPNO=7654;
UPDATE EMP SET DATE_JOIN='13-JUL-70'  WHERE  EMPNO=7566;
UPDATE EMP SET DATE_JOIN='10-FEB-70' WHERE   EMPNO=7698;

UPDATE EMP SET DATE_JOIN='11-DEC-80' WHERE EMPNO=7782 ;
UPDATE EMP SET DATE_JOIN='23-JAN-20' WHERE  EMPNO=7788 ;
UPDATE EMP SET  DATE_JOIN='24-OCT-40' WHERE  EMPNO=7839;
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
```

JOB
CLERK
SALESMAN
MANAGER
ANALYST
PRESIDENT
NULL

9. Display names of all employees in dept 20 and 30

OUTPUT

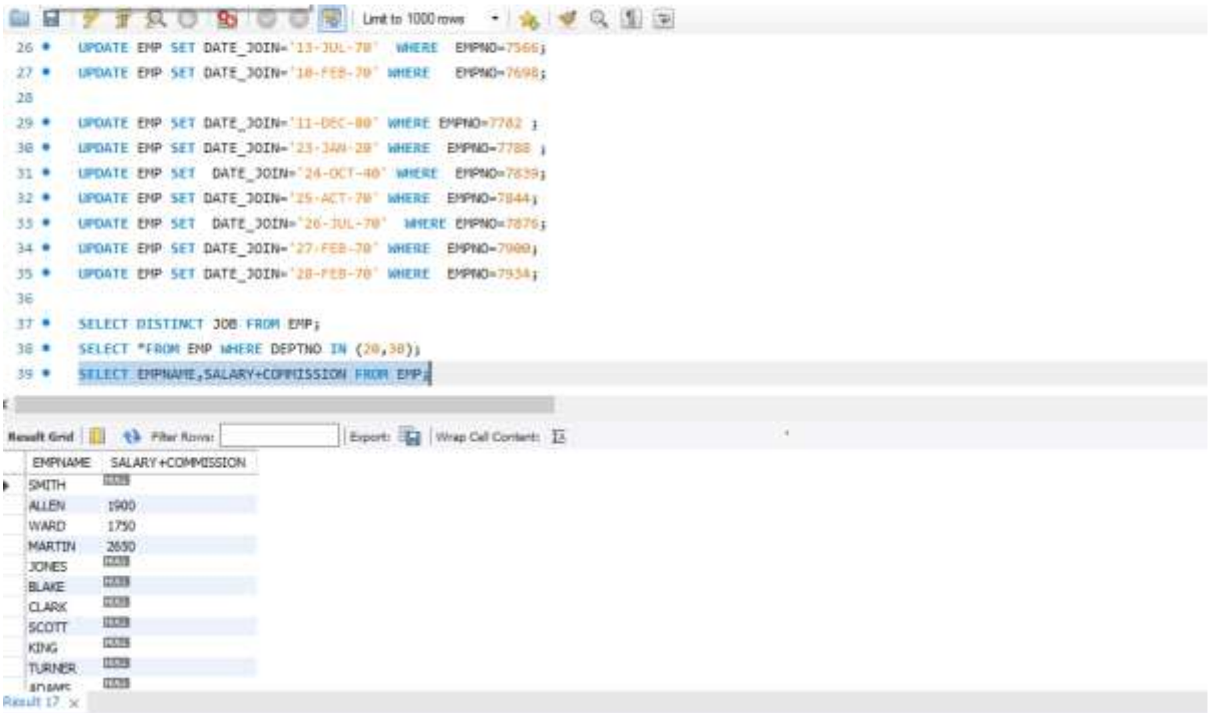


```
UPDATE EMP SET DATE_JOIN='17-ACT-70'  WHERE  EMPNO=7654;
UPDATE EMP SET DATE_JOIN='13-JUL-70'  WHERE  EMPNO=7566;
UPDATE EMP SET DATE_JOIN='10-FEB-70' WHERE   EMPNO=7698;

UPDATE EMP SET DATE_JOIN='11-DEC-80' WHERE EMPNO=7782 ;
UPDATE EMP SET DATE_JOIN='23-JAN-20' WHERE  EMPNO=7788 ;
UPDATE EMP SET  DATE_JOIN='24-OCT-40' WHERE  EMPNO=7839;
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO	DATE_JOIN
7369	SMITH	CLERK	800	NULL	20	17-DEC-80
7499	ALLEN	SALESMAN	1600	300	30	18-JAN-20
7521	WARD	SALESMAN	1250	500	30	17-OCT-40
7654	MARTIN	SALESMAN	1250	1400	30	17-ACT-70
7566	JONES	MANAGER	2975	NULL	20	13-JUL-70
7698	BLAKE	MANAGER	2850	NULL	30	10-FEB-70
7788	SCOTT	ANALYST	3000	NULL	20	23-JAN-20
7844	TURNER	SALESMAN	1500	NULL	30	25-ACT-70
7876	ADAMS	CLERK	1100	NULL	20	26-JUL-70
7900	JAMES	CLERK	950	NULL	30	27-FEB-70

10.List name and Total of salary i.e sal+commission

OUTPUT

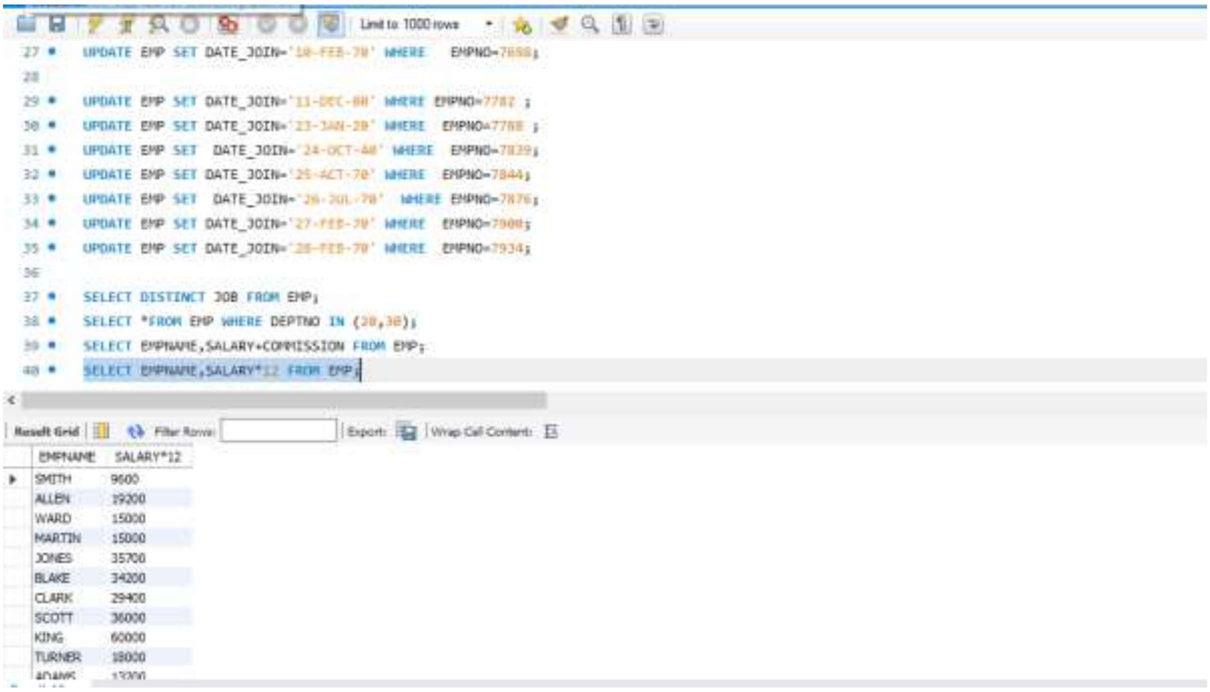


```
UPDATE EMP SET DATE_JOIN='13-JUL-70'  WHERE  EMPNO=7566;
UPDATE EMP SET DATE_JOIN='10-FEB-70' WHERE   EMPNO=7698;

UPDATE EMP SET DATE_JOIN='11-DEC-80' WHERE EMPNO=7782 ;
UPDATE EMP SET DATE_JOIN='23-JAN-20' WHERE  EMPNO=7788 ;
UPDATE EMP SET  DATE_JOIN='24-OCT-40' WHERE  EMPNO=7839;
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY+COMMISSION FROM EMP;
```

EMPNAME	SALARY+COMMISSION
SMITH	NULL
ALLEN	1900
WARD	1750
MARTIN	2650
JONES	NULL
BLAKE	NULL
CLARK	NULL
SCOTT	NULL
KING	NULL
TURNER	NULL
ADAMS	NULL

11.List name and Annual Salary i.e sal*12

OUTPUT

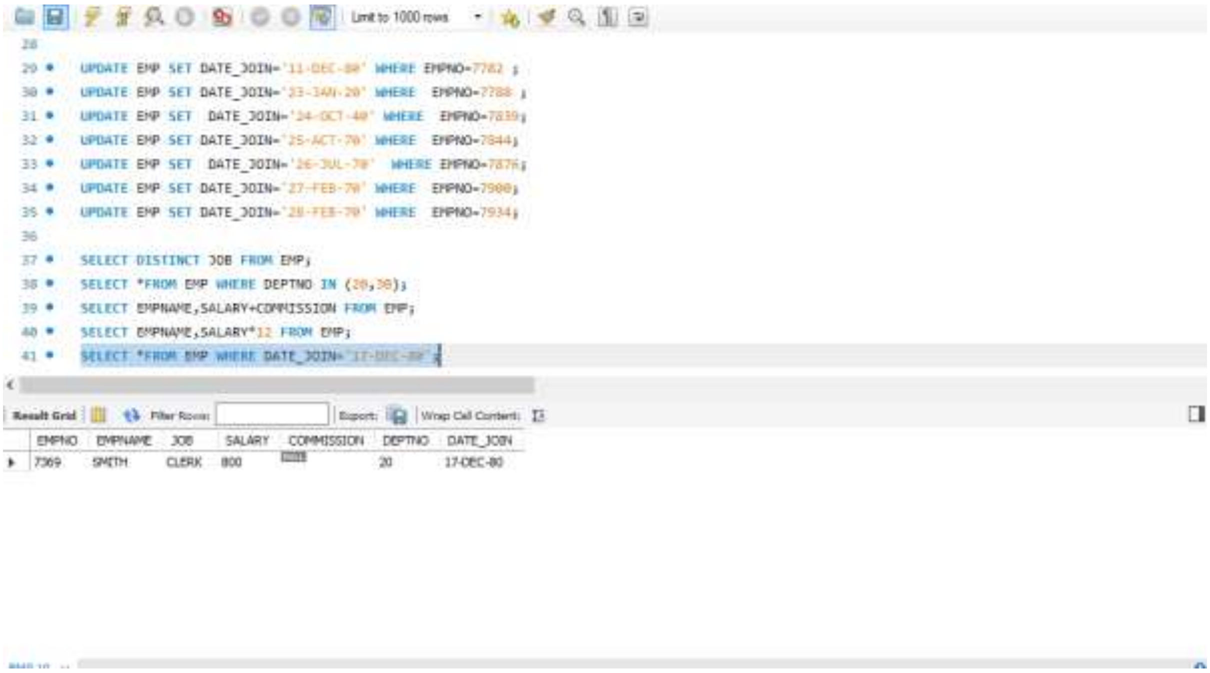


```
UPDATE EMP SET DATE_JOIN='10-FEB-70' WHERE  EMPNO=7698;

UPDATE EMP SET DATE_JOIN='11-DEC-80' WHERE EMPNO=7782 ;
UPDATE EMP SET DATE_JOIN='23-JAN-20' WHERE  EMPNO=7788 ;
UPDATE EMP SET  DATE_JOIN='24-OCT-40' WHERE  EMPNO=7839;
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY+COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;
```

```
EMPNAME   SALARY*12
SMITH     9600
ALLEN     19200
WARD      15000
MARTIN    15000
JONES     35700
BLAKE     34200
CLARK     29400
SCOTT     36000
KING      60000
TURNER    18000
ADAMS     13200
```

12.List the employee who joined in the date ‘03-DEC-81’

OUTPUT

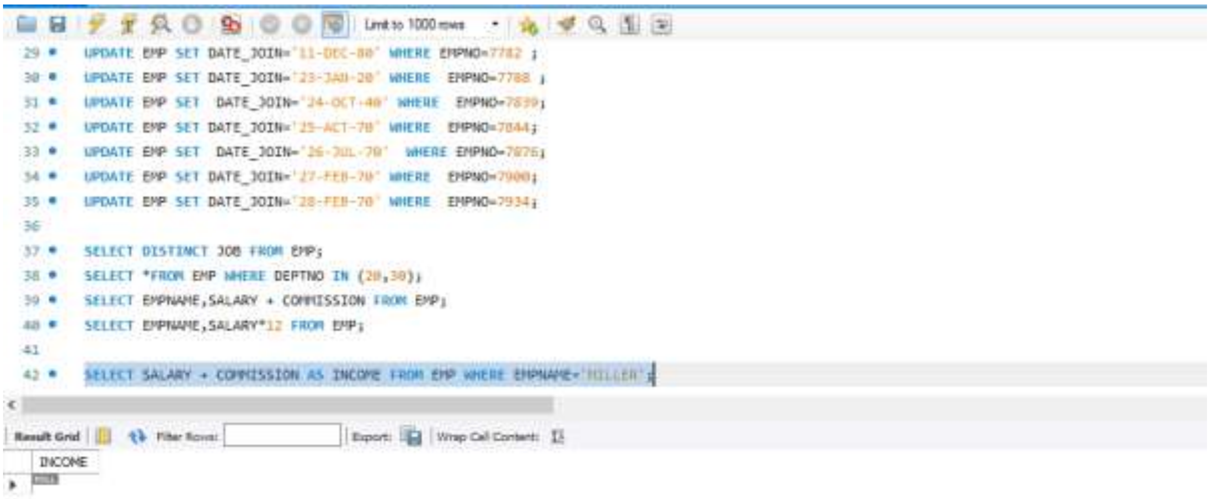


```

UPDATE EMP SET DATE_JOIN='11-DEC-80' WHERE EMPNO=7782 ;
UPDATE EMP SET DATE_JOIN='23-JAN-20' WHERE  EMPNO=7788 ;
UPDATE EMP SET  DATE_JOIN='24-OCT-40' WHERE  EMPNO=7839;
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY+COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;
SELECT *FROM EMP WHERE DATE_JOIN='17-DEC-80';
```

```
EMPNO  EMPNAME  JOB    SALARY  COMMISSION  DEPTNO  DATE_JOIN
7369   SMITH    CLERK  800     NULL        20      17-DEC-80
```

13.Display the total salary of ‘Miller’

OUTPUT

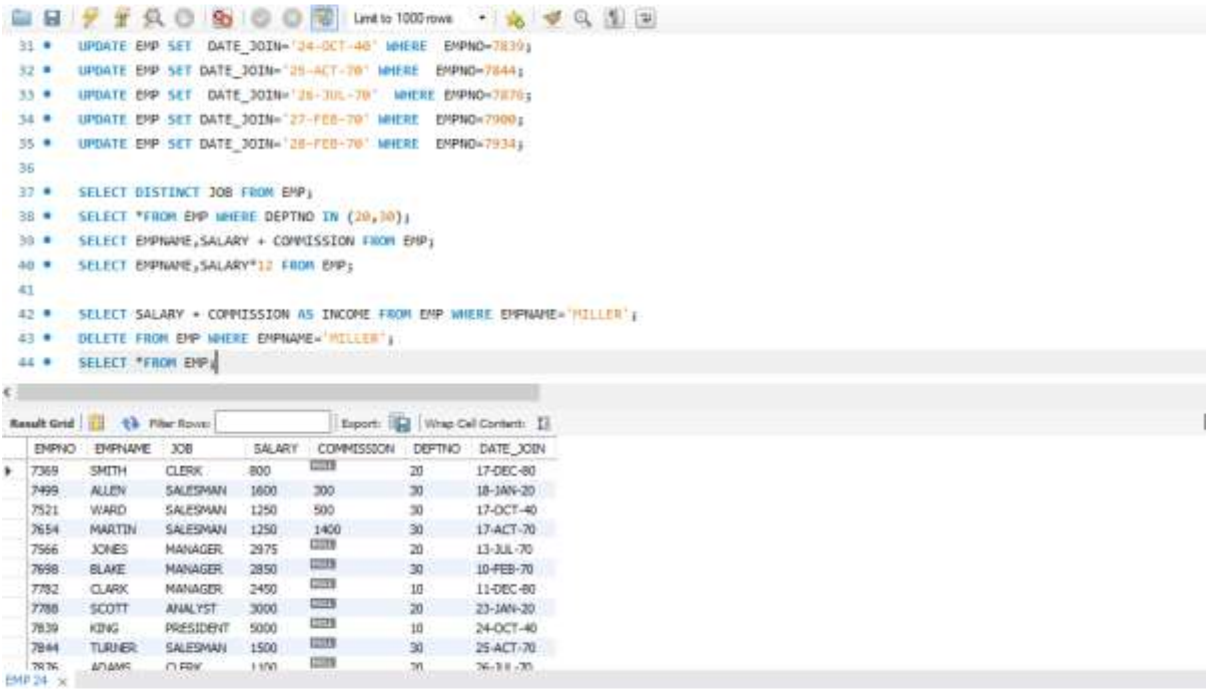


```
UPDATE EMP SET DATE_JOIN='11-DEC-80' WHERE EMPNO=7782 ;
UPDATE EMP SET DATE_JOIN='23-JAN-20' WHERE  EMPNO=7788 ;
UPDATE EMP SET  DATE_JOIN='24-OCT-40' WHERE  EMPNO=7839;
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
```

INCOME
NULL

14.Delete the employee ‘Miller’ from’EMP’

OUTPUT

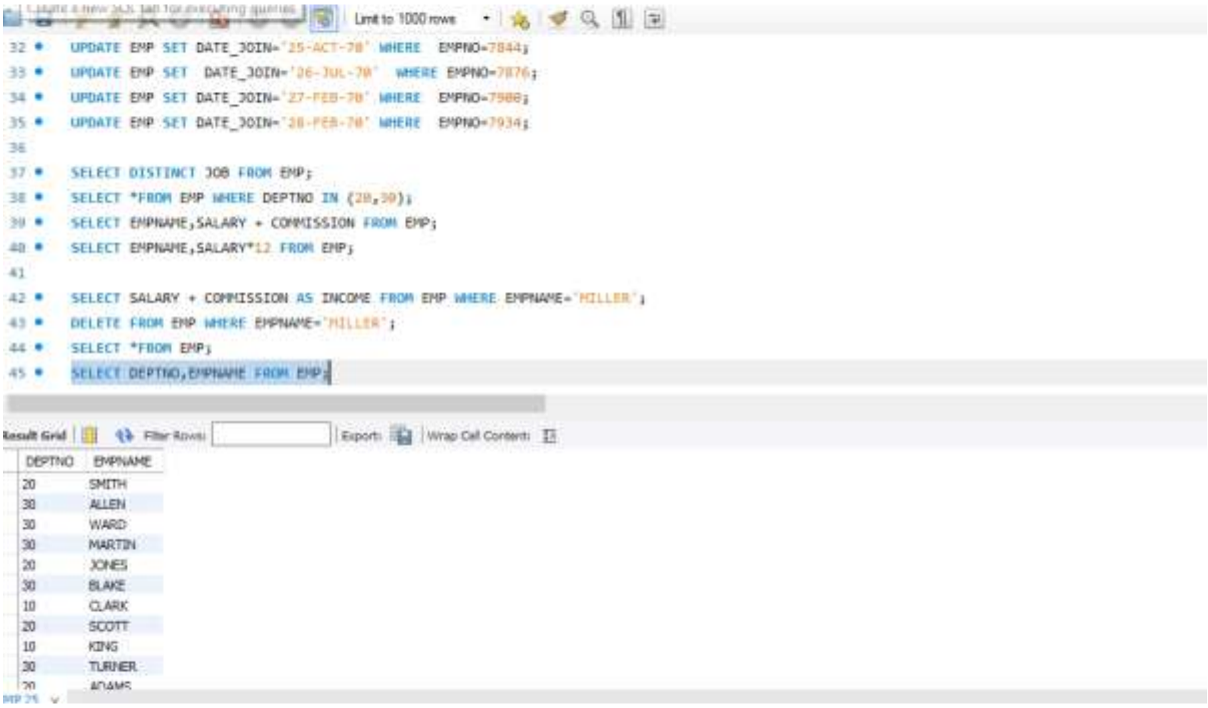


```
UPDATE EMP SET  DATE_JOIN='24-OCT-40' WHERE  EMPNO=7839;
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO	DATE_JOIN
7369	SMITH	CLERK	800	NULL	20	17-DEC-80
7499	ALLEN	SALESMAN	1600	300	30	18-JAN-20
7521	WARD	SALESMAN	1250	500	30	17-OCT-40
7654	MARTIN	SALESMAN	1250	1400	30	17-ACT-70
7566	JONES	MANAGER	2975	NULL	20	13-JUL-70
7698	BLAKE	MANAGER	2850	NULL	30	10-FEB-70
7782	CLARK	MANAGER	2450	NULL	10	11-DEC-80
7788	SCOTT	ANALYST	3000	NULL	20	23-JAN-20
7839	KING	PRESIDENT	5000	NULL	10	24-OCT-40
7844	TURNER	SALESMAN	1500	NULL	30	25-ACT-70
7876	ADAMS	CLERK	1100	NULL	20	26-JUL-70

15.Display name and deptno of all employees.

OUTPUT

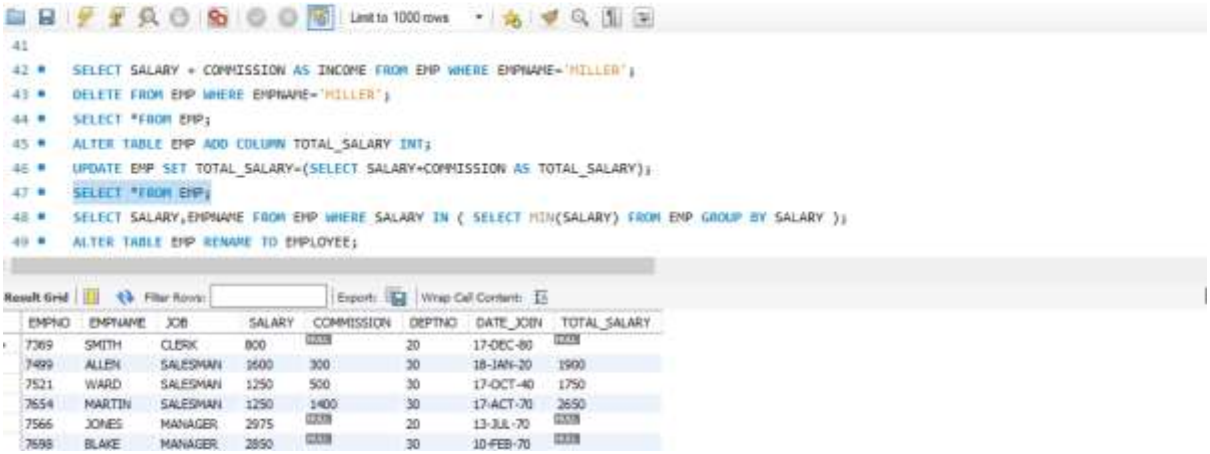


```
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
SELECT DEPTNO,EMPNAME FROM EMP;
```

DEPTNO	EMPNAME
20	SMITH
30	ALLEN
30	WARD
30	MARTIN
20	JONES
30	BLAKE
10	CLARK
20	SCOTT
10	KING
30	TURNER
20	ADAMS

16. Remove the field ‘commission’ fom’EMP’ after updating salary with total salary, i.e sal+commission

OUTPUT

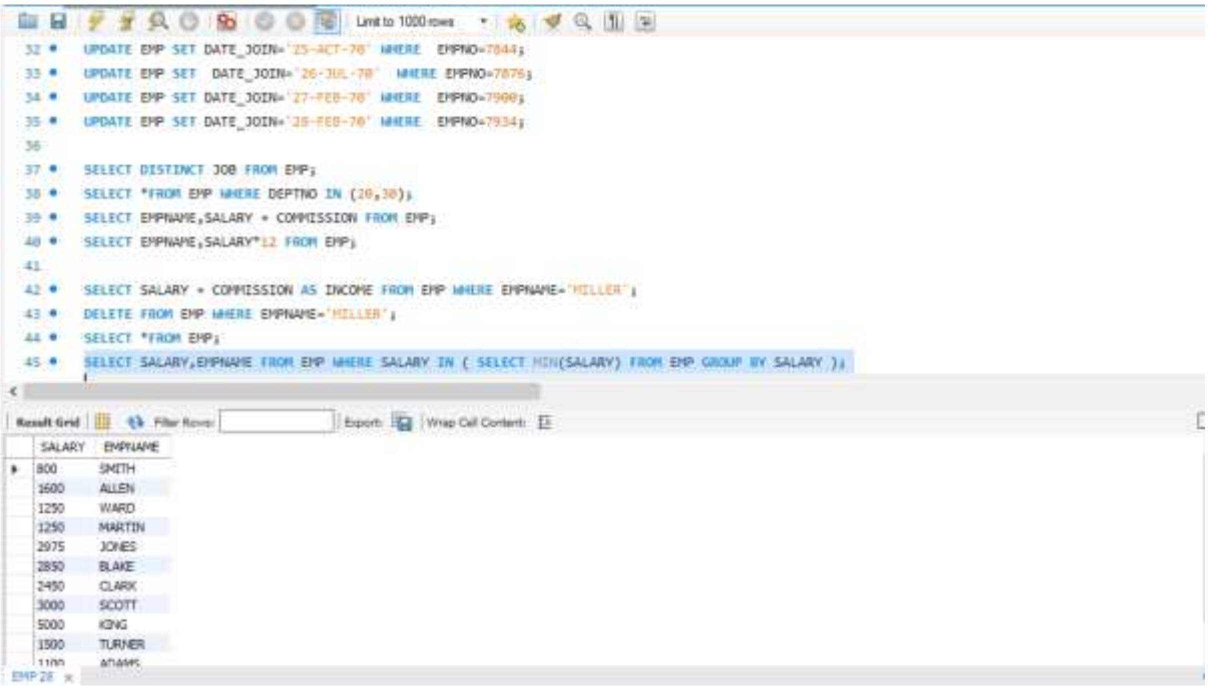


```
SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
ALTER TABLE EMP ADD COLUMN TOTAL_SALARY INT;
UPDATE EMP SET TOTAL_SALARY=(SELECT SALARY+COMMISSION AS TOTAL_SALARY);
SELECT *FROM EMP;
SELECT SALARY,EMPNAME FROM EMP WHERE SALARY IN ( SELECT MIN(SALARY) FROM EMP GROUP BY SALARY );
ALTER TABLE EMP RENAME TO EMPLOYEE;
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO	DATE_JOIN	TOTAL_SALARY
7369	SMITH	CLERK	800	NULL	20	17-DEC-80	NULL
7499	ALLEN	SALESMAN	1600	300	30	18-JAN-20	1900
7521	WARD	SALESMAN	1250	500	30	17-OCT-40	1750
7654	MARTIN	SALESMAN	1250	1400	30	17-ACT-70	2650
7566	JONES	MANAGER	2975	NULL	20	13-JUL-70	NULL
7698	BLAKE	MANAGER	2850	NULL	30	10-FEB-70	NULL

17. Display the name of employees having the same amount of salary **(don’t use subqueries**)

OUT PUT

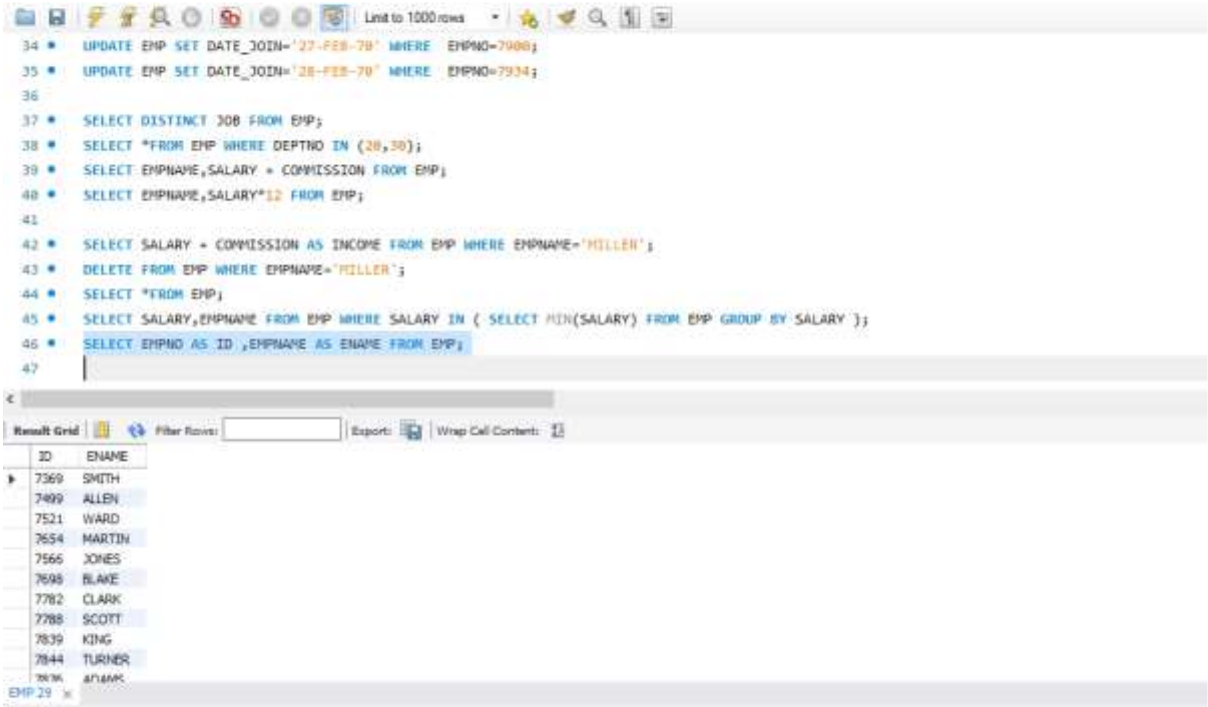


```
UPDATE EMP SET DATE_JOIN='25-ACT-70' WHERE  EMPNO=7844;
UPDATE EMP SET  DATE_JOIN='26-JUL-70'  WHERE EMPNO=7876;
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
SELECT SALARY,EMPNAME FROM EMP WHERE SALARY IN ( SELECT MIN(SALARY) FROM EMP GROUP BY SALARY );
```

SALARY	EMPNAME
800	SMITH
1600	ALLEN
1250	WARD
1250	MARTIN
2975	JONES
2850	BLAKE
2450	CLARK
3000	SCOTT
5000	KING
1500	TURNER
1100	ADAMS

18. Display the name and employee no as ‘name’ and ‘emp_id’

OUTPUT

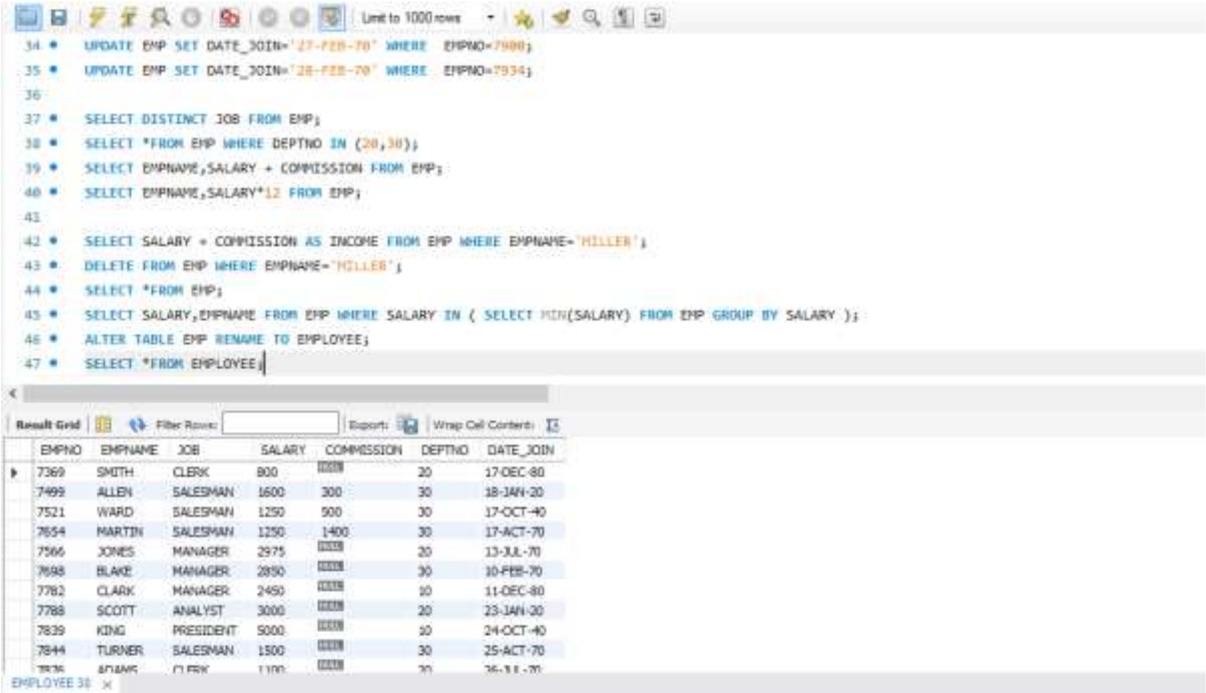


```
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
SELECT SALARY,EMPNAME FROM EMP WHERE SALARY IN ( SELECT MIN(SALARY) FROM EMP GROUP BY SALARY );
SELECT EMPNO AS ID ,EMPNAME AS ENAME FROM EMP;
```

ID	ENAME
7369	SMITH
7499	ALLEN
7521	WARD
7654	MARTIN
7566	JONES
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS

19.Rename table ‘EMP’ to ‘EMPLOYEE’

OUTPUT

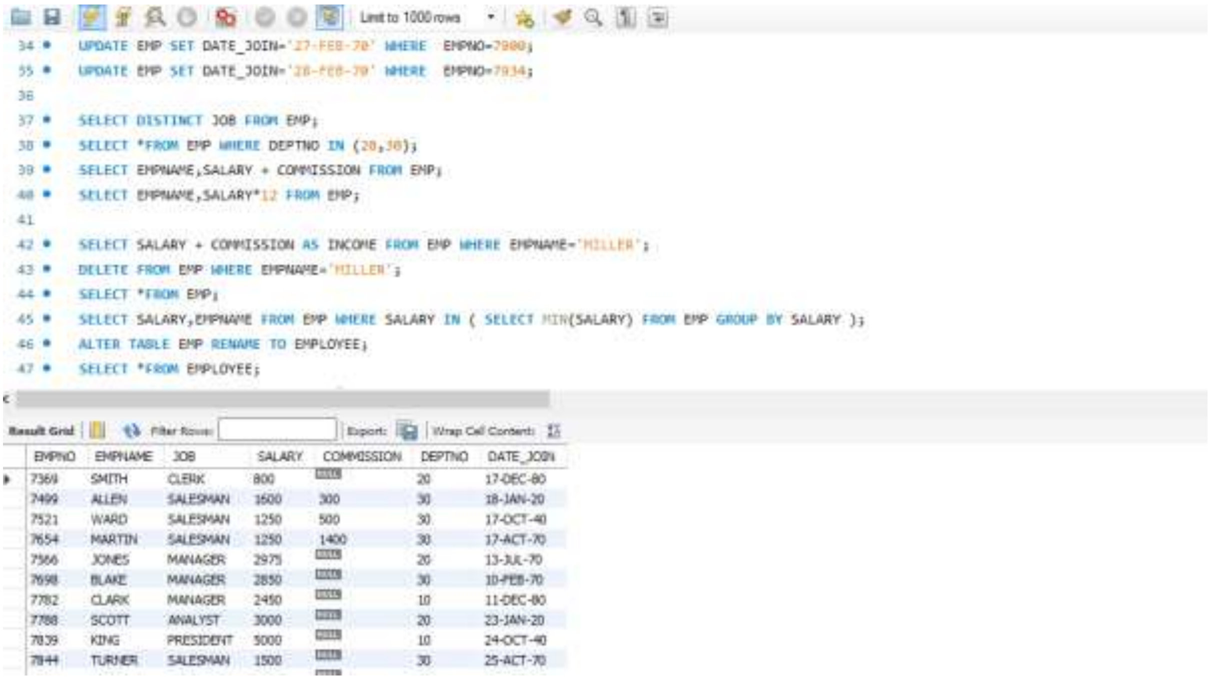


```sql
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
SELECT SALARY,EMPNAME FROM EMP WHERE SALARY IN ( SELECT MIN(SALARY) FROM EMP GROUP BY SALARY );
ALTER TABLE EMP RENAME TO EMPLOYEE;
SELECT *FROM EMPLOYEE;
```

```
EMPNO  EMPNAME  JOB        SALARY  COMMISSION  DEPTNO  DATE_JOIN
7369   SMITH    CLERK      800     NULL        20      17-DEC-80
7499   ALLEN    SALESMAN   1600    300         30      18-JAN-20
7521   WARD     SALESMAN   1250    500         30      17-OCT-40
7654   MARTIN   SALESMAN   1250    1400        30      17-ACT-70
7566   JONES    MANAGER    2975    NULL        20      13-JUL-70
7698   BLAKE    MANAGER    2850    NULL        30      10-FEB-70
7782   CLARK    MANAGER    2450    NULL        10      11-DEC-80
7788   SCOTT    ANALYST    3000    NULL        20      23-JAN-20
7839   KING     PRESIDENT  5000    NULL        10      24-OCT-40
7844   TURNER   SALESMAN   1500    NULL        30      25-ACT-70
7876   ADAMS    CLERK      1100    NULL        20      26-JUL-70
```

20.Create a new table ‘EMP_TAB’ from table ‘EMPLOYEE’

OUTPUT

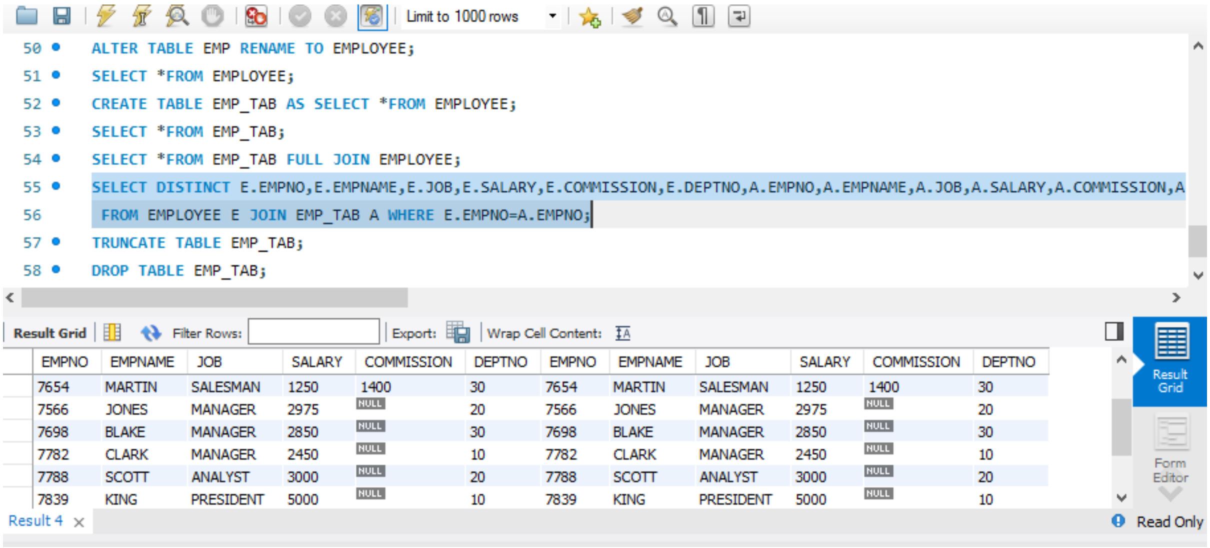


```sql
UPDATE EMP SET DATE_JOIN='27-FEB-70' WHERE  EMPNO=7900;
UPDATE EMP SET DATE_JOIN='28-FEB-70' WHERE  EMPNO=7934;

SELECT DISTINCT JOB FROM EMP;
SELECT *FROM EMP WHERE DEPTNO IN (20,30);
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
SELECT SALARY,EMPNAME FROM EMP WHERE SALARY IN ( SELECT MIN(SALARY) FROM EMP GROUP BY SALARY );
ALTER TABLE EMP RENAME TO EMPLOYEE;
SELECT *FROM EMPLOYEE;
```

```
EMPNO  EMPNAME  JOB        SALARY  COMMISSION  DEPTNO  DATE_JOIN
7369   SMITH    CLERK      800     NULL        20      17-DEC-80
7499   ALLEN    SALESMAN   1600    300         30      18-JAN-20
7521   WARD     SALESMAN   1250    500         30      17-OCT-40
7654   MARTIN   SALESMAN   1250    1400        30      17-ACT-70
7566   JONES    MANAGER    2975    NULL        20      13-JUL-70
7698   BLAKE    MANAGER    2850    NULL        30      10-FEB-70
7782   CLARK    MANAGER    2450    NULL        10      11-DEC-80
7788   SCOTT    ANALYST    3000    NULL        20      23-JAN-20
7839   KING     PRESIDENT  5000    NULL        10      24-OCT-40
7844   TURNER   SALESMAN   1500    NULL        30      25-ACT-70
```

21.List all the details of ‘EMPLOYEE’ and ‘EMP_TAB’

OUTPUT

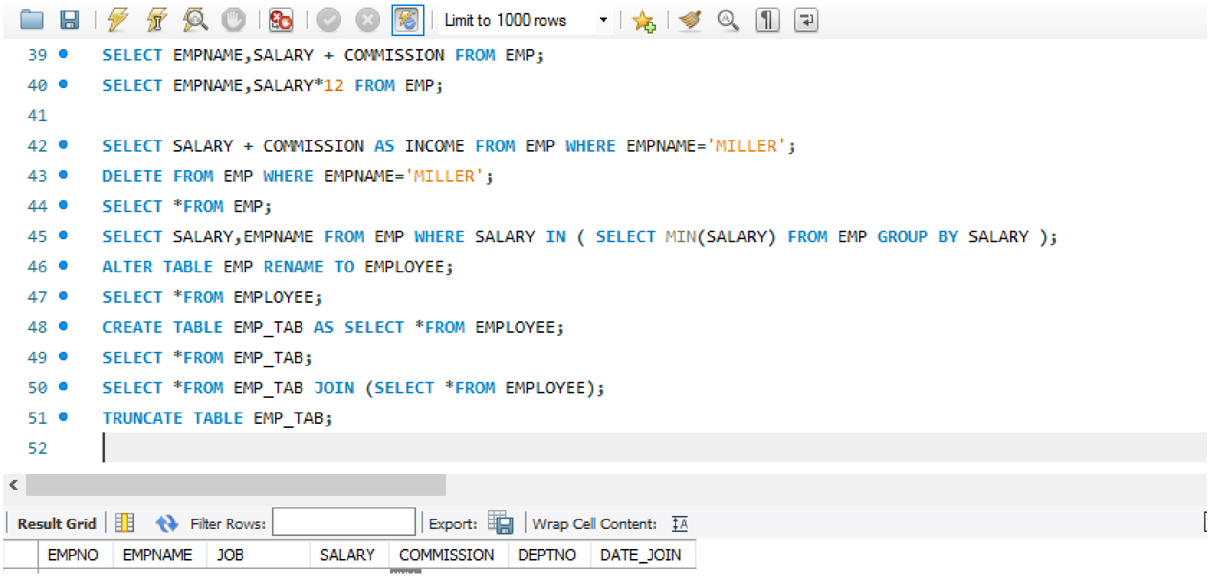


```
ALTER TABLE EMP RENAME TO EMPLOYEE;
SELECT *FROM EMPLOYEE;
CREATE TABLE EMP_TAB AS SELECT *FROM EMPLOYEE;
SELECT *FROM EMP_TAB;
SELECT *FROM EMP_TAB FULL JOIN EMPLOYEE;
SELECT DISTINCT E.EMPNO,E.EMPNAME,E.JOB,E.SALARY,E.COMMISSION,E.DEPTNO,A.EMPNO,A.EMPNAME,A.JOB,A.SALARY,A.COMMISSION,A
 FROM EMPLOYEE E JOIN EMP_TAB A WHERE E.EMPNO=A.EMPNO;
TRUNCATE TABLE EMP_TAB;
DROP TABLE EMP_TAB;
```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO	EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO
7654	MARTIN	SALESMAN	1250	1400	30	7654	MARTIN	SALESMAN	1250	1400	30
7566	JONES	MANAGER	2975	NULL	20	7566	JONES	MANAGER	2975	NULL	20
7698	BLAKE	MANAGER	2850	NULL	30	7698	BLAKE	MANAGER	2850	NULL	30
7782	CLARK	MANAGER	2450	NULL	10	7782	CLARK	MANAGER	2450	NULL	10
7788	SCOTT	ANALYST	3000	NULL	20	7788	SCOTT	ANALYST	3000	NULL	20
7839	KING	PRESIDENT	5000	NULL	10	7839	KING	PRESIDENT	5000	NULL	10

22.Delete all records from ‘EMP’

OUTPUT

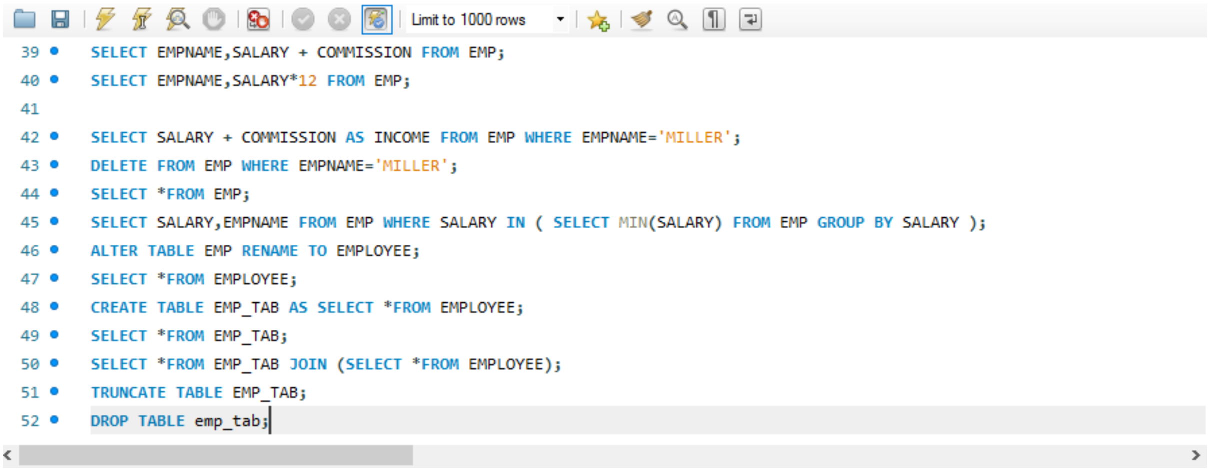


```
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
SELECT SALARY,EMPNAME FROM EMP WHERE SALARY IN ( SELECT MIN(SALARY) FROM EMP GROUP BY SALARY );
ALTER TABLE EMP RENAME TO EMPLOYEE;
SELECT *FROM EMPLOYEE;
CREATE TABLE EMP_TAB AS SELECT *FROM EMPLOYEE;
SELECT *FROM EMP_TAB;
SELECT *FROM EMP_TAB JOIN (SELECT *FROM EMPLOYEE);
TRUNCATE TABLE EMP_TAB;

```

EMPNO	EMPNAME	JOB	SALARY	COMMISSION	DEPTNO	DATE_JOIN

23.Delete the table ‘EMP’

OUTPUT



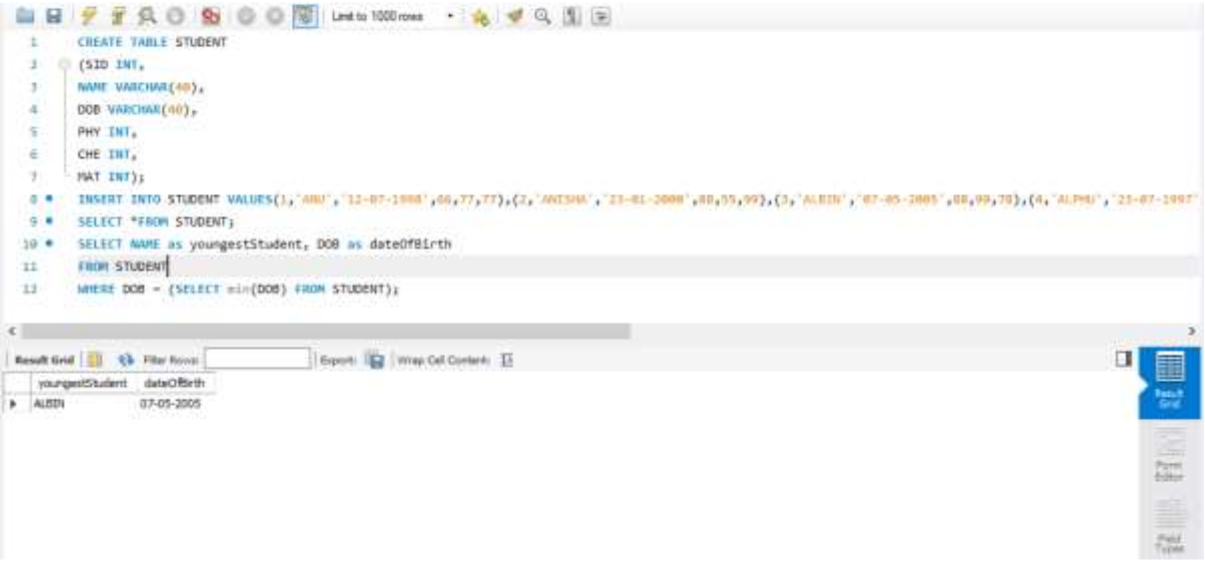
```
SELECT EMPNAME,SALARY + COMMISSION FROM EMP;
SELECT EMPNAME,SALARY*12 FROM EMP;

SELECT SALARY + COMMISSION AS INCOME FROM EMP WHERE EMPNAME='MILLER';
DELETE FROM EMP WHERE EMPNAME='MILLER';
SELECT *FROM EMP;
SELECT SALARY,EMPNAME FROM EMP WHERE SALARY IN ( SELECT MIN(SALARY) FROM EMP GROUP BY SALARY );
ALTER TABLE EMP RENAME TO EMPLOYEE;
SELECT *FROM EMPLOYEE;
CREATE TABLE EMP_TAB AS SELECT *FROM EMPLOYEE;
SELECT *FROM EMP_TAB;
SELECT *FROM EMP_TAB JOIN (SELECT *FROM EMPLOYEE);
TRUNCATE TABLE EMP_TAB;
DROP TABLE emp_tab;
```

QUESTION SET 2

Create a table STUDENT with fields sid, name, dob (date of birth) and marks of 3 subjects (physics, chemistry and maths). Add the details of 5 students. Perform the following queries:

1. Display the id and name of youngest student.

OUTPUT

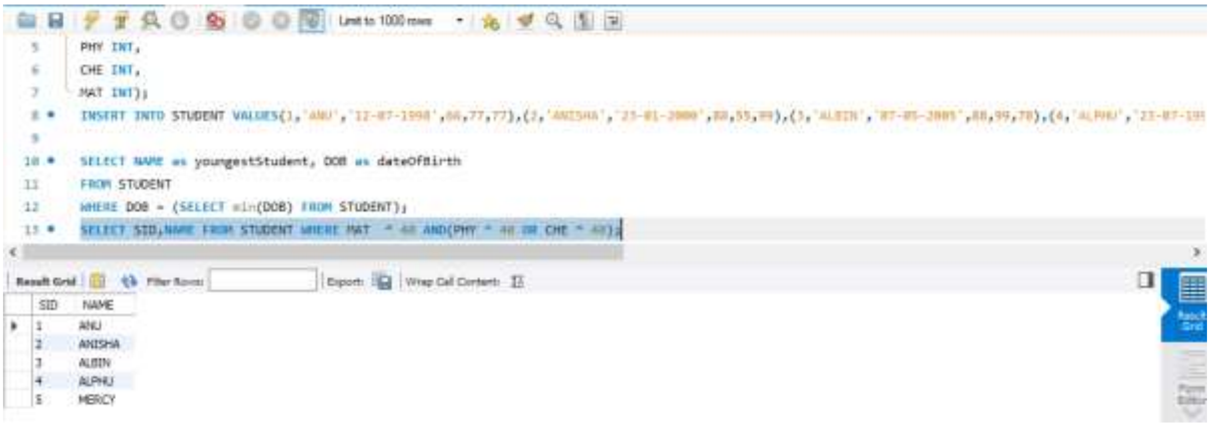


```
CREATE TABLE STUDENT
(SID INT,
NAME VARCHAR(40),
DOB VARCHAR(40),
PHY INT,
CHE INT,
MAT INT);
INSERT INTO STUDENT VALUES(1,'ANU','12-07-1998',66,77,77),(2,'ANISHA','23-01-2000',88,55,99),(3,'ALBIN','07-05-2005',88,99,78),(4,'ALPHU','23-07-1997'
SELECT *FROM STUDENT;
SELECT NAME as youngestStudent, DOB as dateOfBirth
FROM STUDENT
WHERE DOB = (SELECT min(DOB) FROM STUDENT);
```

youngestStudent	dateOfBirth
ALBIN	07-05-2005

2. Display the details of students who have passed in maths and either in physics or chemistry.(pass mark = 40 marks and above)

OUTPUT

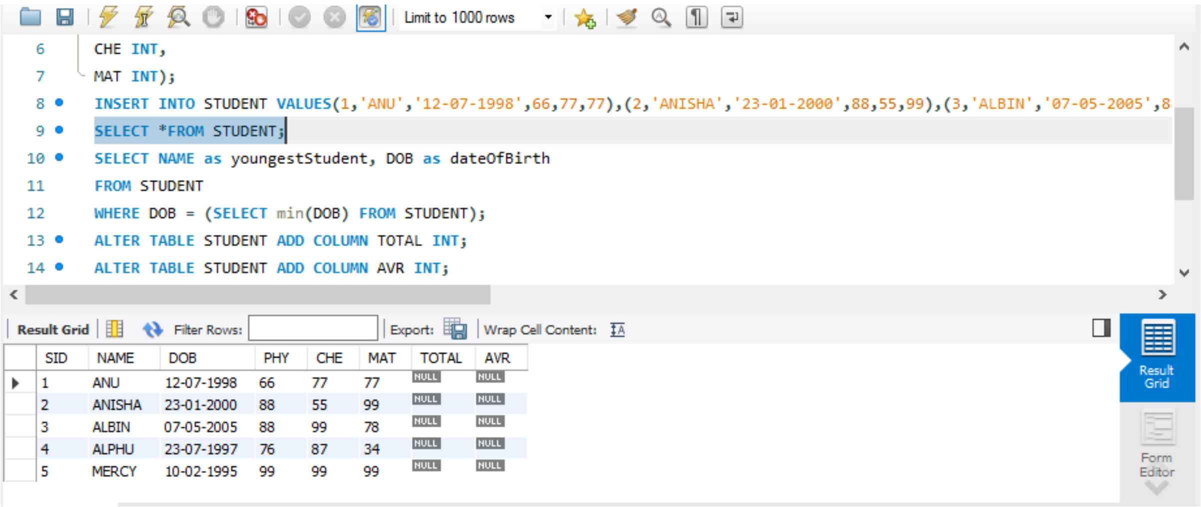


```
PHY INT,
CHE INT,
MAT INT);
INSERT INTO STUDENT VALUES(1,'ANU','12-07-1998',66,77,77),(2,'ANISHA','23-01-2000',88,55,99),(3,'ALBIN','07-05-2005',88,99,78),(4,'ALPHU','23-07-199

SELECT NAME as youngestStudent, DOB as dateOfBirth
FROM STUDENT
WHERE DOB = (SELECT min(DOB) FROM STUDENT);
SELECT SID,NAME FROM STUDENT WHERE MAT > 40 AND(PHY > 40 OR CHE > 40);
```

SID	NAME
1	ANU
2	ANISHA
3	ALBIN
4	ALPHU
5	MERCY

3. Add two more columns total and average.

OUTPUT

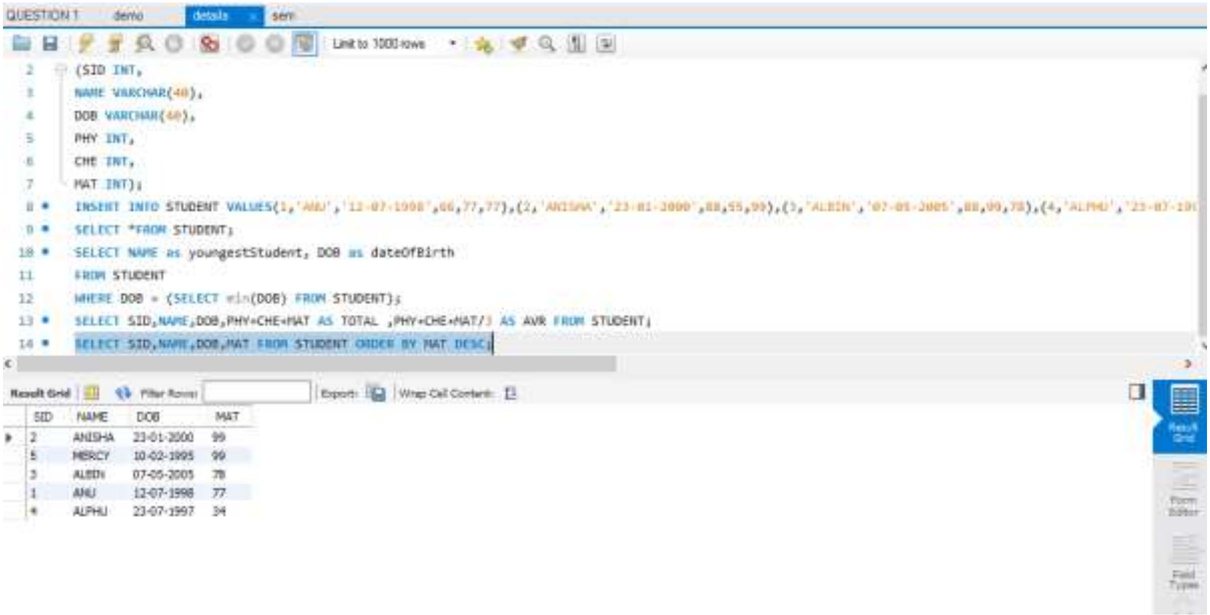


```
CHE INT,
MAT INT);
INSERT INTO STUDENT VALUES(1,'ANU','12-07-1998',66,77,77),(2,'ANISHA','23-01-2000',88,55,99),(3,'ALBIN','07-05-2005',8
SELECT *FROM STUDENT;
SELECT NAME as youngestStudent, DOB as dateOfBirth
FROM STUDENT
WHERE DOB = (SELECT min(DOB) FROM STUDENT);
ALTER TABLE STUDENT ADD COLUMN TOTAL INT;
ALTER TABLE STUDENT ADD COLUMN AVR INT;
```

SID	NAME	DOB	PHY	CHE	MAT	TOTAL	AVR
1	ANU	12-07-1998	66	77	77	NULL	NULL
2	ANISHA	23-01-2000	88	55	99	NULL	NULL
3	ALBIN	07-05-2005	88	99	78	NULL	NULL
4	ALPHU	23-07-1997	76	87	34	NULL	NULL
5	MERCY	10-02-1995	99	99	99	NULL	NULL

4. Display the name of student who scored highest marks in maths.

OUTPUT

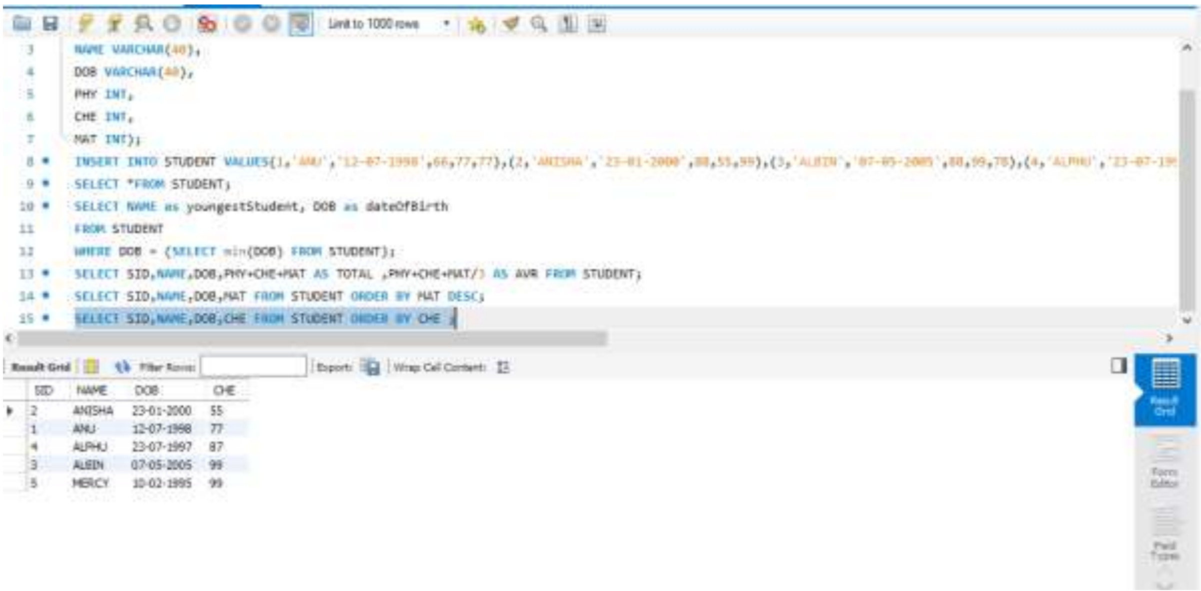


```
(SID INT,
NAME VARCHAR(40),
DOB VARCHAR(40),
PHY INT,
CHE INT,
MAT INT);
INSERT INTO STUDENT VALUES(1,'ANU','12-07-1998',66,77,77),(2,'ANISHA','23-01-2000',88,55,99),(3,'ALBIN','07-05-2005',88,99,78),(4,'ALPHU','23-07-19
SELECT *FROM STUDENT;
SELECT NAME as youngestStudent, DOB as dateOfBirth
FROM STUDENT
WHERE DOB = (SELECT min(DOB) FROM STUDENT);
SELECT SID,NAME,DOB,PHY+CHE+MAT AS TOTAL ,PHY+CHE+MAT/3 AS AVR FROM STUDENT;
SELECT SID,NAME,DOB,MAT FROM STUDENT ORDER BY MAT DESC;
```

SID	NAME	DOB	MAT
2	ANISHA	23-01-2000	99
5	MERCY	10-02-1995	99
3	ALBIN	07-05-2005	78
1	ANU	12-07-1998	77
4	ALPHU	23-07-1997	34

5. Display the name of student who scored least marks in chemistry.

OUTPUT

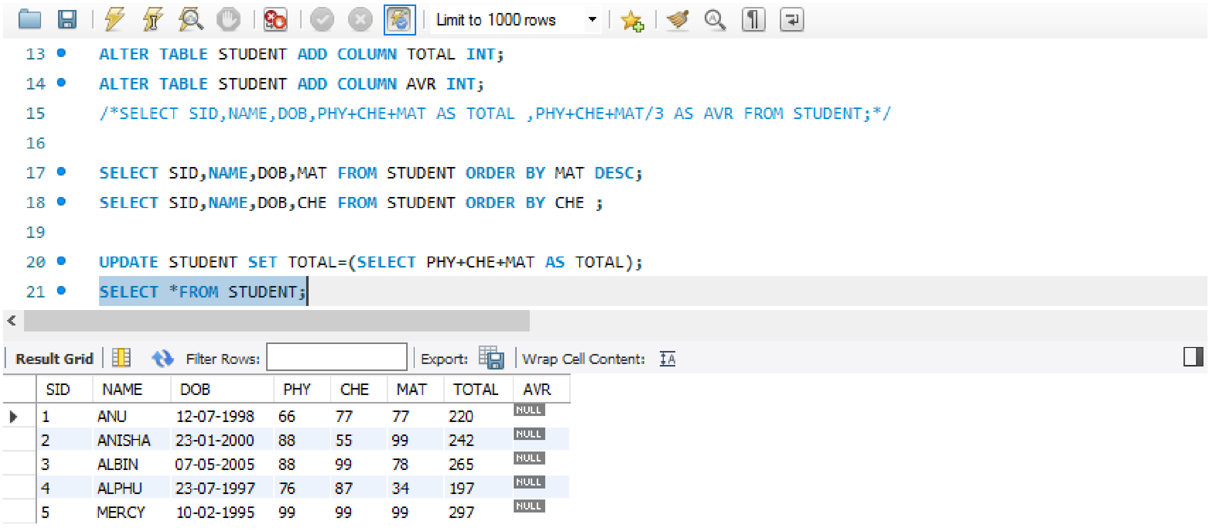


```
NAME VARCHAR(40),
DOB VARCHAR(40),
PHY INT,
CHE INT,
MAT INT);
INSERT INTO STUDENT VALUES(1,'ANU','12-07-1998',66,77,77),(2,'ANISHA','23-01-2000',88,55,99),(3,'ALBIN','07-05-2005',88,99,78),(4,'ALPHU','23-07-19
SELECT *FROM STUDENT;
SELECT NAME as youngestStudent, DOB as dateOfBirth
FROM STUDENT
WHERE DOB = (SELECT min(DOB) FROM STUDENT);
SELECT SID,NAME,DOB,PHY+CHE+MAT AS TOTAL ,PHY+CHE+MAT/3 AS AVR FROM STUDENT;
SELECT SID,NAME,DOB,MAT FROM STUDENT ORDER BY MAT DESC;
SELECT SID,NAME,DOB,CHE FROM STUDENT ORDER BY CHE ;
```

SID	NAME	DOB	CHE
2	ANISHA	23-01-2000	55
1	ANU	12-07-1998	77
4	ALPHU	23-07-1997	87
3	ALBIN	07-05-2005	99
5	MERCY	10-02-1995	99

6. Update column total with total marks.

OUTPUT

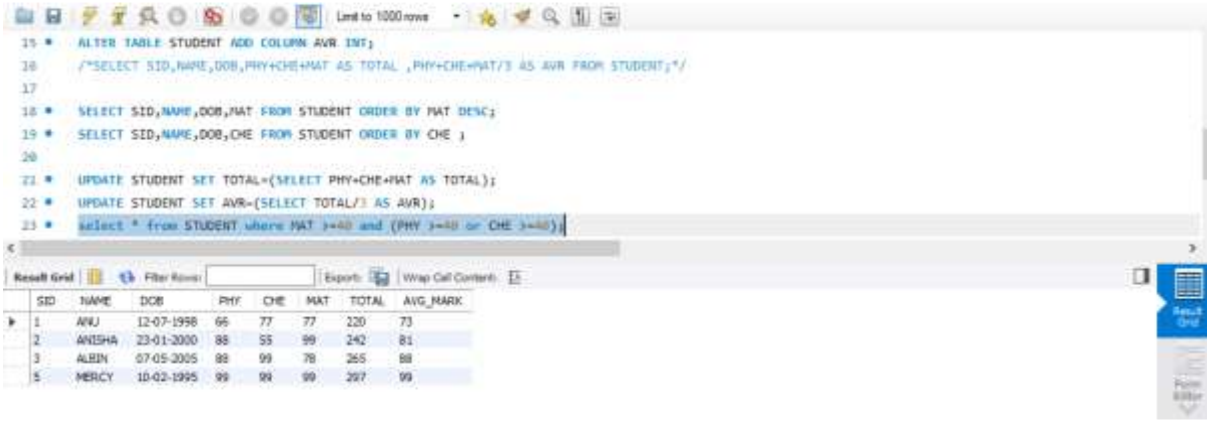


```sql
ALTER TABLE STUDENT ADD COLUMN TOTAL INT;
ALTER TABLE STUDENT ADD COLUMN AVR INT;
/*SELECT SID,NAME,DOB,PHY+CHE+MAT AS TOTAL ,PHY+CHE+MAT/3 AS AVR FROM STUDENT;*/

SELECT SID,NAME,DOB,MAT FROM STUDENT ORDER BY MAT DESC;
SELECT SID,NAME,DOB,CHE FROM STUDENT ORDER BY CHE ;

UPDATE STUDENT SET TOTAL=(SELECT PHY+CHE+MAT AS TOTAL);
SELECT *FROM STUDENT;
```

SID	NAME	DOB	PHY	CHE	MAT	TOTAL	AVR
1	ANU	12-07-1998	66	77	77	220	NULL
2	ANISHA	23-01-2000	88	55	99	242	NULL
3	ALBIN	07-05-2005	88	99	78	265	NULL
4	ALPHU	23-07-1997	76	87	34	197	NULL
5	MERCY	10-02-1995	99	99	99	297	NULL

7. Display details of students in order of total merit.

OUTPUT

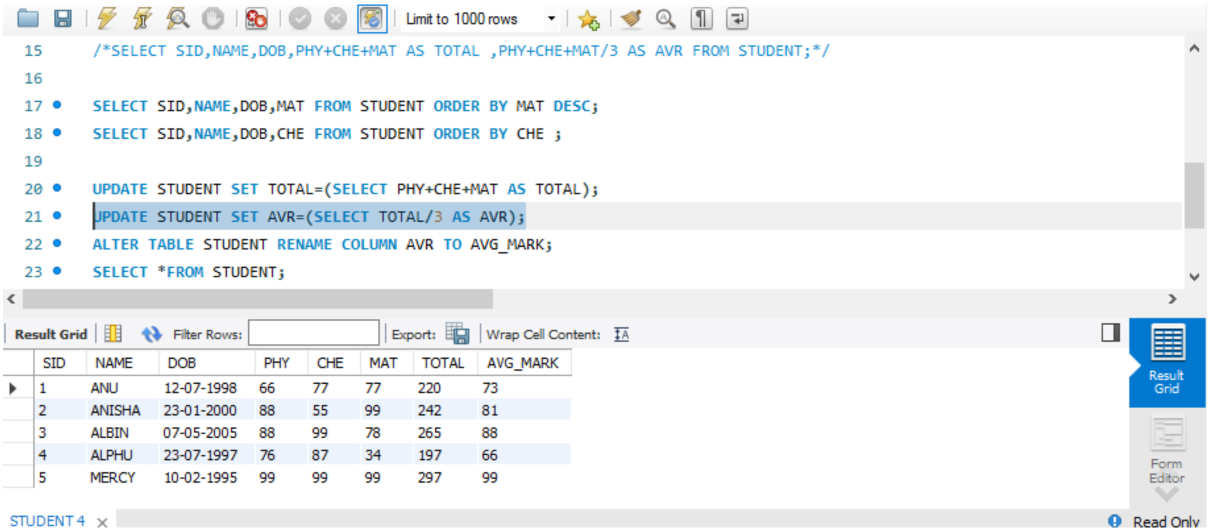


```sql
ALTER TABLE STUDENT ADD COLUMN AVR INT;
/*SELECT SID,NAME,DOB,PHY+CHE+MAT AS TOTAL ,PHY+CHE+MAT/3 AS AVR FROM STUDENT;*/

SELECT SID,NAME,DOB,MAT FROM STUDENT ORDER BY MAT DESC;
SELECT SID,NAME,DOB,CHE FROM STUDENT ORDER BY CHE ;

UPDATE STUDENT SET TOTAL=(SELECT PHY+CHE+MAT AS TOTAL);
UPDATE STUDENT SET AVR=(SELECT TOTAL/3 AS AVR);
select * from STUDENT where MAT >=40 and (PHY >=40 or CHE >=40);
```

SID	NAME	DOB	PHY	CHE	MAT	TOTAL	AVG_MARK
1	ANU	12-07-1998	66	77	77	220	73
2	ANISHA	23-01-2000	88	55	99	242	81
3	ALBIN	07-05-2005	88	99	78	265	88
5	MERCY	10-02-1995	99	99	99	297	99

8. Rename the column average with avg_mark

OUTPUT

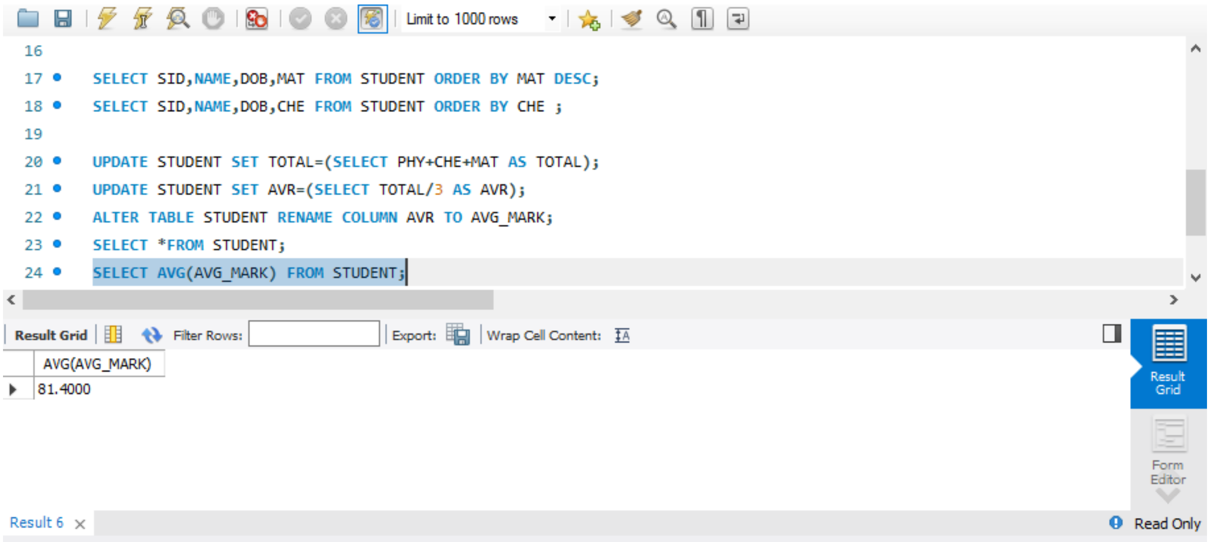


```
/*SELECT SID,NAME,DOB,PHY+CHE+MAT AS TOTAL ,PHY+CHE+MAT/3 AS AVR FROM STUDENT;*/

SELECT SID,NAME,DOB,MAT FROM STUDENT ORDER BY MAT DESC;
SELECT SID,NAME,DOB,CHE FROM STUDENT ORDER BY CHE ;

UPDATE STUDENT SET TOTAL=(SELECT PHY+CHE+MAT AS TOTAL);
UPDATE STUDENT SET AVR=(SELECT TOTAL/3 AS AVR);
ALTER TABLE STUDENT RENAME COLUMN AVR TO AVG_MARK;
SELECT *FROM STUDENT;
```

SID	NAME	DOB	PHY	CHE	MAT	TOTAL	AVG_MARK
1	ANU	12-07-1998	66	77	77	220	73
2	ANISHA	23-01-2000	88	55	99	242	81
3	ALBIN	07-05-2005	88	99	78	265	88
4	ALPHU	23-07-1997	76	87	34	197	66
5	MERCY	10-02-1995	99	99	99	297	99

9. Find out the overall average of class.

OUTPUT

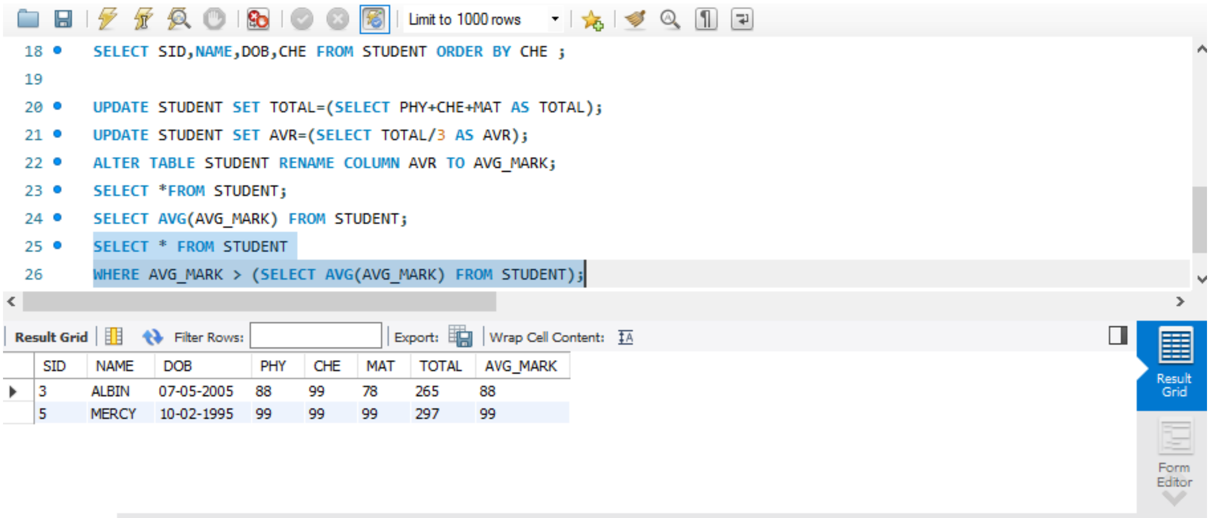


```
SELECT SID,NAME,DOB,MAT FROM STUDENT ORDER BY MAT DESC;
SELECT SID,NAME,DOB,CHE FROM STUDENT ORDER BY CHE ;

UPDATE STUDENT SET TOTAL=(SELECT PHY+CHE+MAT AS TOTAL);
UPDATE STUDENT SET AVR=(SELECT TOTAL/3 AS AVR);
ALTER TABLE STUDENT RENAME COLUMN AVR TO AVG_MARK;
SELECT *FROM STUDENT;
SELECT AVG(AVG_MARK) FROM STUDENT;
```

AVG(AVG_MARK)
81.4000

10.Display details of students whose average is greater than overall average.

OUTPUT

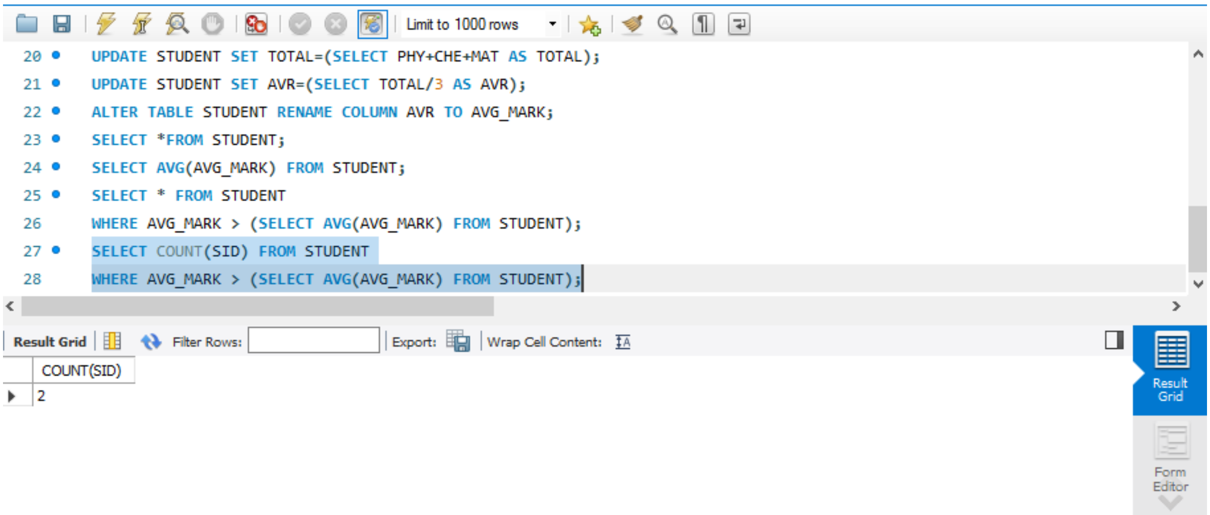


```
SELECT SID,NAME,DOB,CHE FROM STUDENT ORDER BY CHE ;

UPDATE STUDENT SET TOTAL=(SELECT PHY+CHE+MAT AS TOTAL);
UPDATE STUDENT SET AVR=(SELECT TOTAL/3 AS AVR);
ALTER TABLE STUDENT RENAME COLUMN AVR TO AVG_MARK;
SELECT *FROM STUDENT;
SELECT AVG(AVG_MARK) FROM STUDENT;
SELECT * FROM STUDENT
WHERE AVG_MARK > (SELECT AVG(AVG_MARK) FROM STUDENT);
```

SID	NAME	DOB	PHY	CHE	MAT	TOTAL	AVG_MARK
3	ALBIN	07-05-2005	88	99	78	265	88
5	MERCY	10-02-1995	99	99	99	297	99

11.Find the total no: of students whose average is greater than overall average.

OUTPUT



```
UPDATE STUDENT SET TOTAL=(SELECT PHY+CHE+MAT AS TOTAL);
UPDATE STUDENT SET AVR=(SELECT TOTAL/3 AS AVR);
ALTER TABLE STUDENT RENAME COLUMN AVR TO AVG_MARK;
SELECT *FROM STUDENT;
SELECT AVG(AVG_MARK) FROM STUDENT;
SELECT * FROM STUDENT
WHERE AVG_MARK > (SELECT AVG(AVG_MARK) FROM STUDENT);
SELECT COUNT(SID) FROM STUDENT
WHERE AVG_MARK > (SELECT AVG(AVG_MARK) FROM STUDENT);
```

COUNT(SID)
2

QUESTION SET 3 *DATE 1-6-2021*

Create the Table LOAN_ACCOUNTS with the structure given below

Field Name	Data Type	Length
Accno	CHAR	4
Cust_name	VARCHAR2	15
Loan_Amount	NUMBER	7 digits and 2 decimal places
Installments	NUMBER	
int_rate	NUMBER	2 digits and 2 decimal places
Start_date	DATE	

Interest	NUMBER	7 digits and 2 decimal places

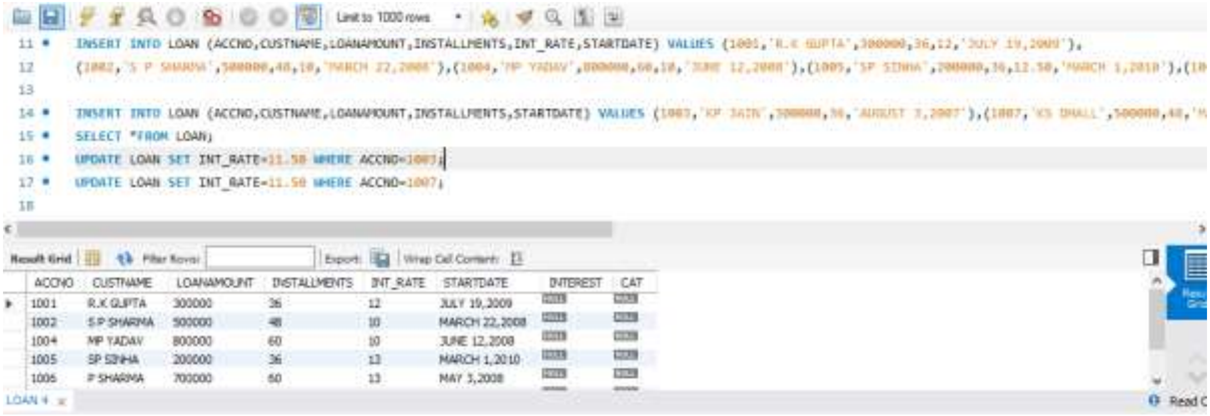
Add another column ‘category’ of type varchar2(1) in the Loan Table.

Insert the following details into the table

Accno	Cust_name	Loan_Amount	Installments	int_rate	Start_date	Interest
1001	R.K Gupta	300,000.00	36	12.00	July 19, 2009	
1002	S.P Sharma	500,000.00	48	10.00	March 22, 2008	
1003	K.P Jain	300,000.00	36	NULL	August 3, 2007	
1004	M.P Yadav	800,000.00	60	10.00	June 12, 2008	
1005	S.P Sinha	200,000.00	36	12.50	March 1, 2010	
1006	P. Sharma	700,000.00	60	12.50	May 6, 2008	
1007	K.S Dhall	500,000.00	48	NULL	May 3, 2008	

1. Put the interest rate 11.50% for all the loans for which the interest rate is NULL.

OUTPUT

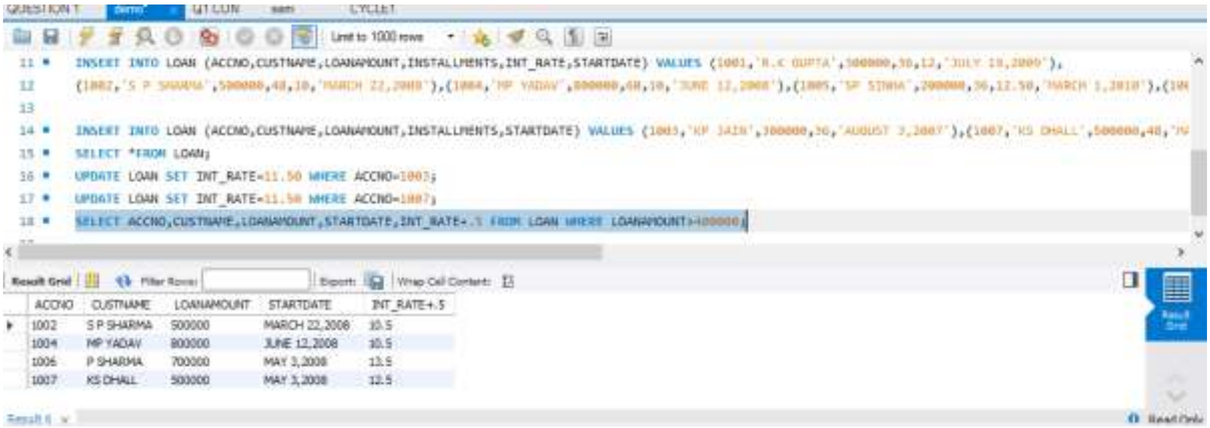


```
INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,INT_RATE,STARTDATE) VALUES (1001,'R.K GUPTA',300000,36,12,'JULY 19,2009'),
(1002,'S P SHARMA',500000,48,10,'MARCH 22,2008'),(1004,'MP YADAV',800000,60,10,'JUNE 12,2008'),(1005,'SP SINHA',200000,36,12.50,'MARCH 1,2010'),(10

INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,STARTDATE) VALUES (1003,'KP JAIN',300000,36,'AUGUST 3,2007'),(1007,'KS DHALL',500000,48,'M
SELECT *FROM LOAN;
UPDATE LOAN SET INT_RATE=11.50 WHERE ACCNO=1003;
UPDATE LOAN SET INT_RATE=11.50 WHERE ACCNO=1007;
```

```
ACCNO  CUSTNAME    LOANAMOUNT  INSTALLMENTS  INT_RATE  STARTDATE       INTEREST  CAT
1001   R.K GUPTA   300000      36            12        JULY 19,2009    NULL      NULL
1002   S.P SHARMA  500000      48            10        MARCH 22,2008   NULL      NULL
1004   MP YADAV    800000      60            10        JUNE 12,2008    NULL      NULL
1005   SP SINHA    200000      36            13        MARCH 1,2010    NULL      NULL
1006   P SHARMA    700000      60            13        MAY 3,2008      NULL      NULL
```

2. Increase the interest rate by 0.5% for all the Loans for which the Loan amount is more than 400000.

OUTPUT

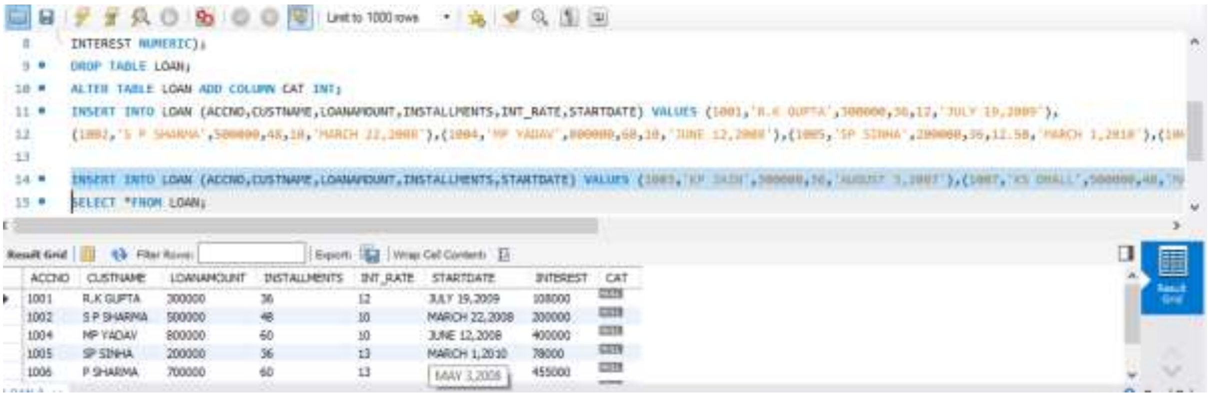


```
INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,INT_RATE,STARTDATE) VALUES (1001,'R.K GUPTA',300000,36,12,'JULY 19,2009'),
(1002,'S P SHARMA',500000,48,10,'MARCH 22,2008'),(1004,'MP YADAV',800000,60,10,'JUNE 12,2008'),(1005,'SP SINHA',200000,36,12.50,'MARCH 1,2010'),(10

INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,STARTDATE) VALUES (1003,'KP JAIN',300000,36,'AUGUST 3,2007'),(1007,'KS DHALL',500000,48,'M
SELECT *FROM LOAN;
UPDATE LOAN SET INT_RATE=11.50 WHERE ACCNO=1003;
UPDATE LOAN SET INT_RATE=11.50 WHERE ACCNO=1007;
SELECT ACCNO,CUSTNAME,LOANAMOUNT,STARTDATE,INT_RATE+.5 FROM LOAN WHERE LOANAMOUNT>400000;
```

ACCNO	CUSTNAME	LOANAMOUNT	STARTDATE	INT_RATE+.5
1002	S P SHARMA	500000	MARCH 22,2008	10.5
1004	MP YADAV	800000	JUNE 12,2008	10.5
1006	P SHARMA	700000	MAY 3,2008	13.5
1007	KS DHALL	500000	MAY 3,2008	12.5

3. For each Loan replace Interest with (Loan_amount * Int_rate* installments)/(12*100).

OUTPUT

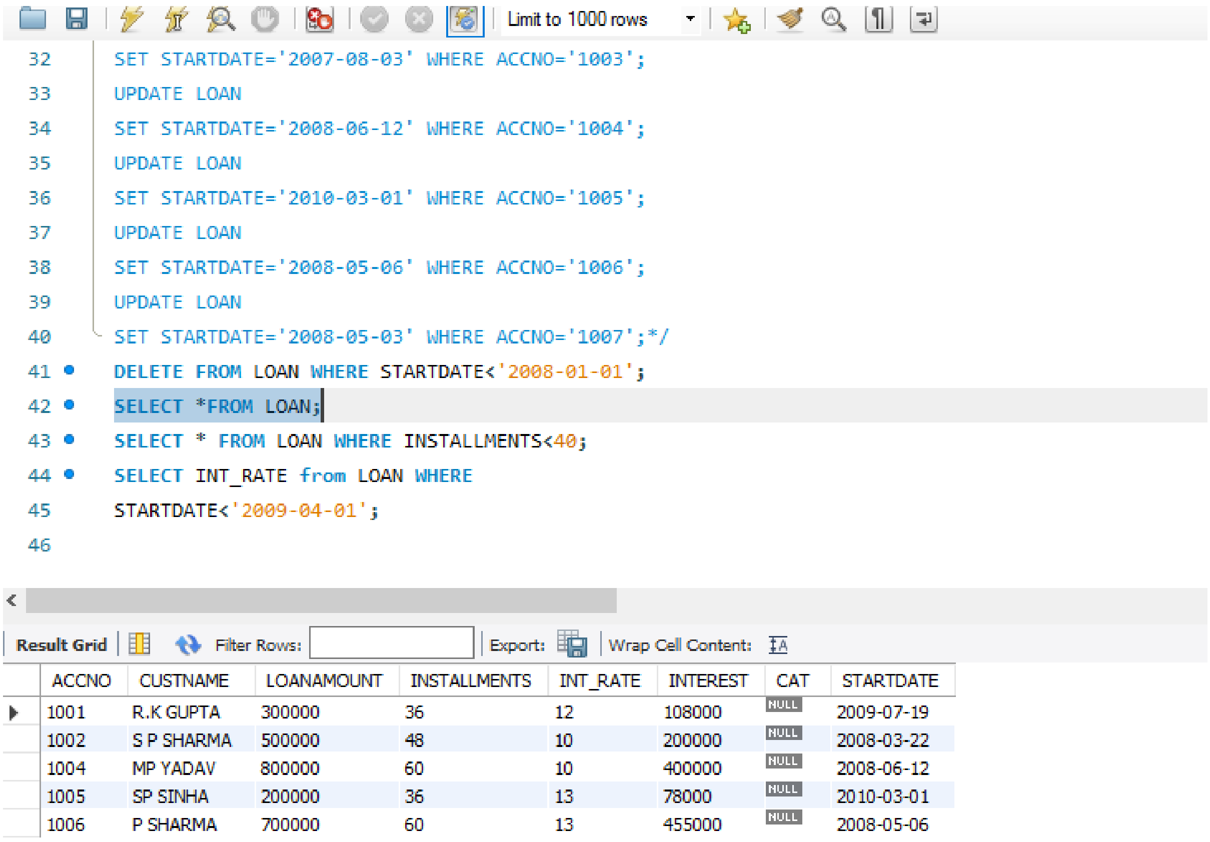


```
INTEREST NUMERIC);
DROP TABLE LOAN;
ALTER TABLE LOAN ADD COLUMN CAT INT;
INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,INT_RATE,STARTDATE) VALUES (1001,'R.K GUPTA',300000,36,12,'JULY 19,2009'),
(1002,'S P SHARMA',500000,48,10,'MARCH 22,2008'),(1004,'MP YADAV',800000,60,10,'JUNE 12,2008'),(1005,'SP SINHA',200000,36,12.50,'MARCH 1,2010'),(10

INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,STARTDATE) VALUES (1003,'KP JAIN',300000,36,'AUGUST 3,2007'),(1007,'KS DHALL',500000,48,'M
SELECT *FROM LOAN;
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	STARTDATE	INTEREST	CAT
1001	R.K GUPTA	300000	36	12	JULY 19,2009	108000	NULL
1002	S P SHARMA	500000	48	10	MARCH 22,2008	200000	NULL
1004	MP YADAV	800000	60	10	JUNE 12,2008	400000	NULL
1005	SP SINHA	200000	36	13	MARCH 1,2010	78000	NULL
1006	P SHARMA	700000	60	13	MAY 3,2008	455000	NULL

4. Delete the records of all the Loans whose start date is before 2008.

OUTPUT

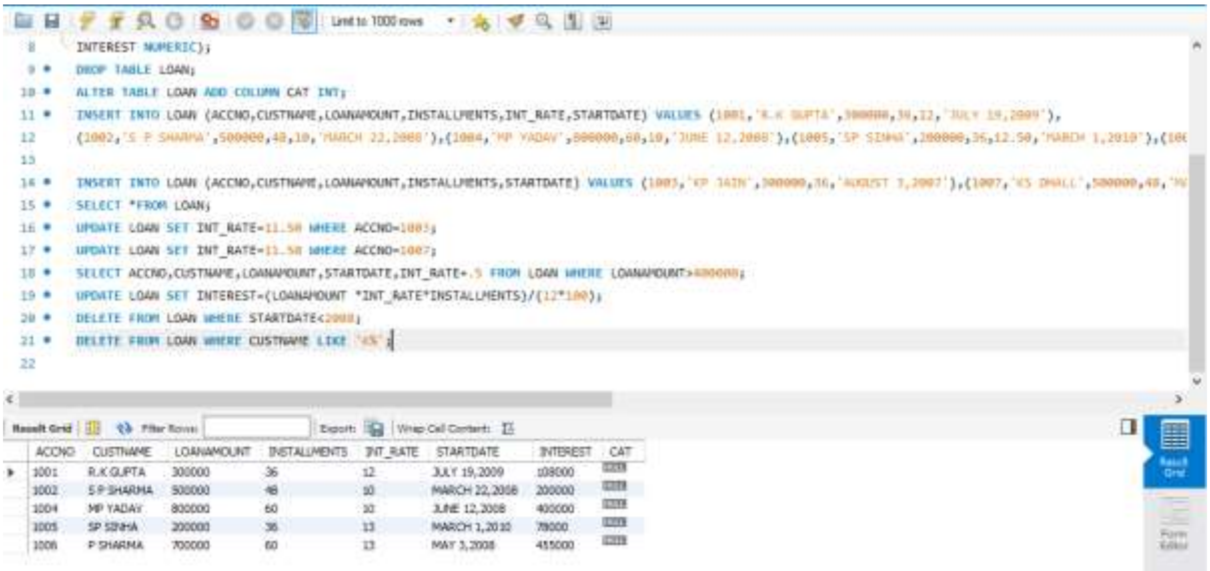


```
SET STARTDATE='2007-08-03' WHERE ACCNO='1003';
UPDATE LOAN
SET STARTDATE='2008-06-12' WHERE ACCNO='1004';
UPDATE LOAN
SET STARTDATE='2010-03-01' WHERE ACCNO='1005';
UPDATE LOAN
SET STARTDATE='2008-05-06' WHERE ACCNO='1006';
UPDATE LOAN
SET STARTDATE='2008-05-03' WHERE ACCNO='1007';*/
DELETE FROM LOAN WHERE STARTDATE<'2008-01-01';
SELECT *FROM LOAN;
SELECT * FROM LOAN WHERE INSTALLMENTS<40;
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE
1001	R.K GUPTA	300000	36	12	108000	NULL	2009-07-19
1002	S P SHARMA	500000	48	10	200000	NULL	2008-03-22
1004	MP YADAV	800000	60	10	400000	NULL	2008-06-12
1005	SP SINHA	200000	36	13	78000	NULL	2010-03-01
1006	P SHARMA	700000	60	13	455000	NULL	2008-05-06

5. Delete the records of all the Loans whose name starts with ‘K’

OUTPUT

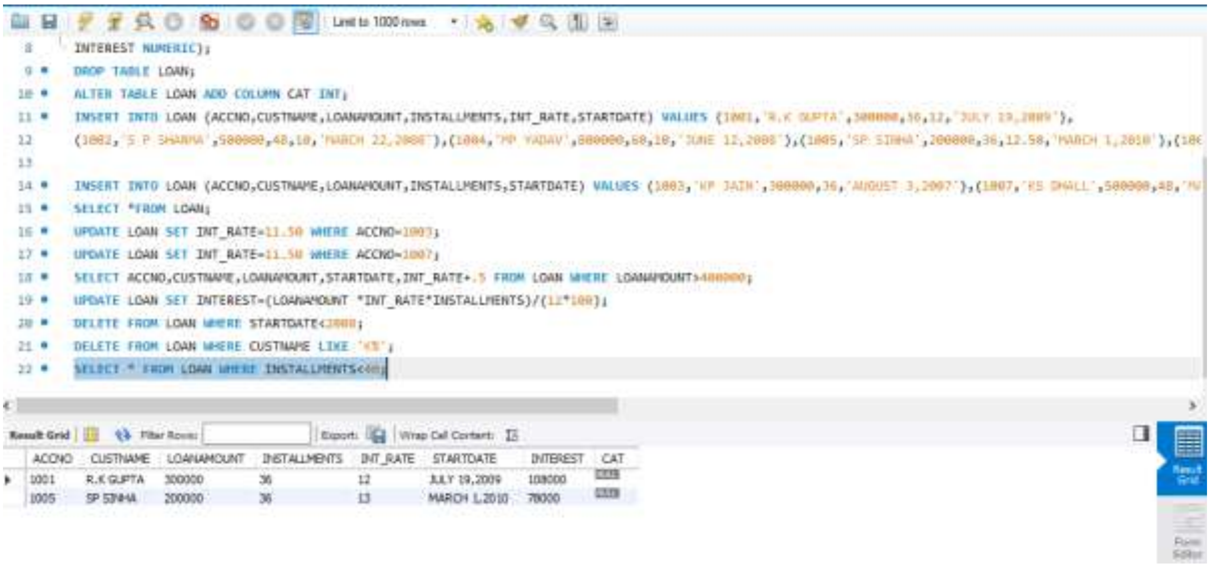


```
INTEREST NUMERIC);
DROP TABLE LOAN;
ALTER TABLE LOAN ADD COLUMN CAT INT;
INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,INT_RATE,STARTDATE) VALUES (1001,'R.K GUPTA',300000,36,12,'JULY 19,2009'),
(1002,'S P SHARMA',500000,48,10,'MARCH 22,2008'),(1004,'MP YADAV',800000,60,10,'JUNE 12,2008'),(1005,'SP SINHA',200000,36,12.50,'MARCH 1,2010'),(10

INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,STARTDATE) VALUES (1003,'KP JAIN',300000,36,'AUGUST 3,2007'),(1007,'KS DHALL',500000,48,'M
SELECT *FROM LOAN;
UPDATE LOAN SET INT_RATE=11.50 WHERE ACCNO=1003;
UPDATE LOAN SET INT_RATE=11.50 WHERE ACCNO=1007;
SELECT ACCNO,CUSTNAME,LOANAMOUNT,STARTDATE,INT_RATE+.5 FROM LOAN WHERE LOANAMOUNT>400000;
UPDATE LOAN SET INTEREST=(LOANAMOUNT *INT_RATE*INSTALLMENTS)/(12*100);
DELETE FROM LOAN WHERE STARTDATE<2008;
DELETE FROM LOAN WHERE CUSTNAME LIKE 'K%';
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	STARTDATE	INTEREST	CAT
1001	R.K GUPTA	300000	36	12	JULY 19,2009	108000	NULL
1002	S P SHARMA	500000	48	10	MARCH 22,2008	200000	NULL
1004	MP YADAV	800000	60	10	JUNE 12,2008	400000	NULL
1005	SP SINHA	200000	36	13	MARCH 1,2010	78000	NULL
1006	P SHARMA	700000	60	13	MAY 3,2008	455000	NULL

6. Display the details of all the Loans with less than 40 installments.

OUTPUT

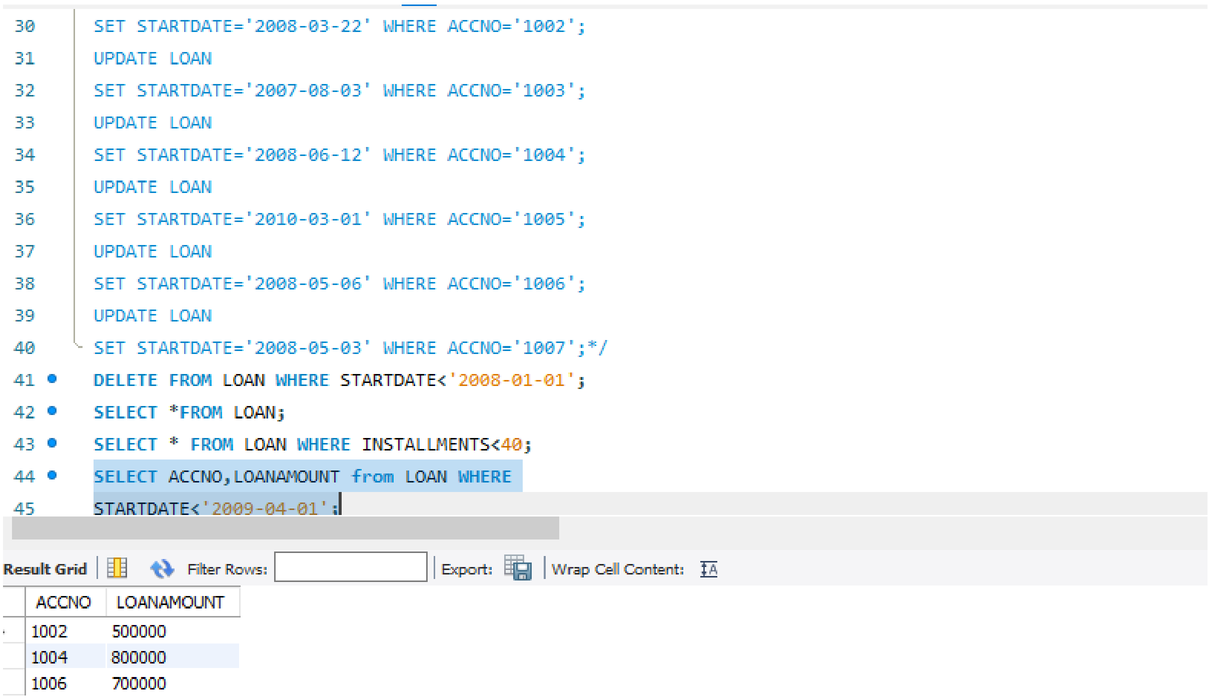


```
INTEREST NUMERIC);
DROP TABLE LOAN;
ALTER TABLE LOAN ADD COLUMN CAT INT;
INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,INT_RATE,STARTDATE) VALUES (1001,'R.K GUPTA',300000,36,12,'JULY 19,2009'),
(1002,'S P SHARMA',500000,48,10,'MARCH 22,2008'),(1004,'MP YADAV',800000,60,10,'JUNE 12,2008'),(1005,'SP SINHA',200000,36,12.50,'MARCH 1,2010'),(10

INSERT INTO LOAN (ACCNO,CUSTNAME,LOANAMOUNT,INSTALLMENTS,STARTDATE) VALUES (1003,'KP JAIN',300000,36,'AUGUST 3,2007'),(1007,'KS DHALL',500000,48,'M
SELECT *FROM LOAN;
UPDATE LOAN SET INT_RATE=11.50 WHERE ACCNO=1003;
UPDATE LOAN SET INT_RATE=11.50 WHERE ACCNO=1007;
SELECT ACCNO,CUSTNAME,LOANAMOUNT,STARTDATE,INT_RATE+.5 FROM LOAN WHERE LOANAMOUNT>400000;
UPDATE LOAN SET INTEREST=(LOANAMOUNT *INT_RATE*INSTALLMENTS)/(12*100);
DELETE FROM LOAN WHERE STARTDATE<2008;
DELETE FROM LOAN WHERE CUSTNAME LIKE 'K%';
SELECT * FROM LOAN WHERE INSTALLMENTS<40;
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	STARTDATE	INTEREST	CAT
1001	R.K GUPTA	300000	36	12	JULY 19,2009	108000	NULL
1005	SP SINHA	200000	36	13	MARCH 1,2010	78000	NULL

7. Display the Accno and Loan_amount of all the loans started before 01-04-2009.

OUTPUT

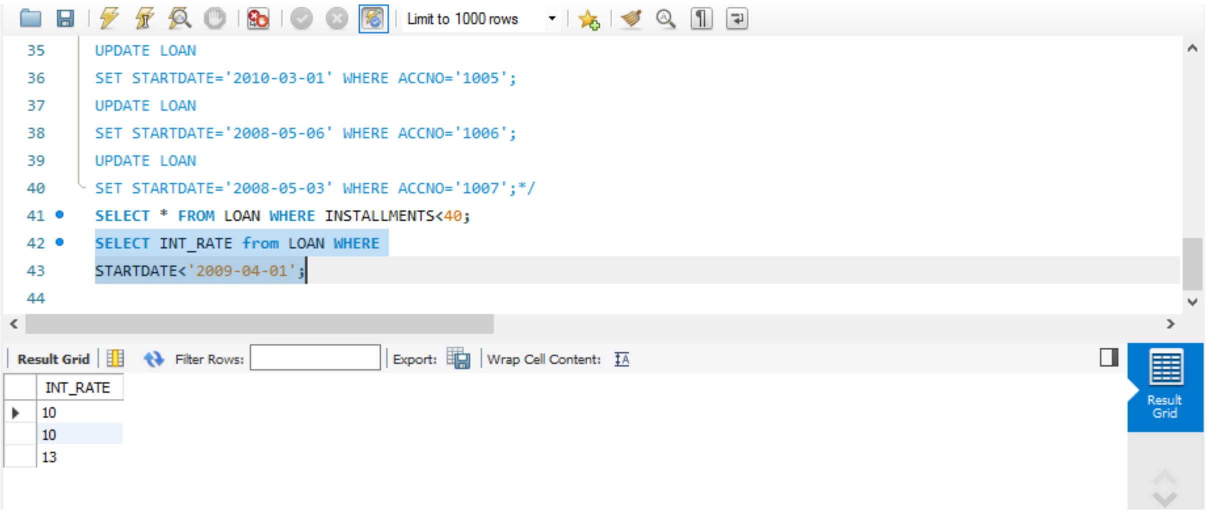


```
SET STARTDATE='2008-03-22' WHERE ACCNO='1002';
UPDATE LOAN
SET STARTDATE='2007-08-03' WHERE ACCNO='1003';
UPDATE LOAN
SET STARTDATE='2008-06-12' WHERE ACCNO='1004';
UPDATE LOAN
SET STARTDATE='2010-03-01' WHERE ACCNO='1005';
UPDATE LOAN
SET STARTDATE='2008-05-06' WHERE ACCNO='1006';
UPDATE LOAN
SET STARTDATE='2008-05-03' WHERE ACCNO='1007';*/
DELETE FROM LOAN WHERE STARTDATE<'2008-01-01';
SELECT *FROM LOAN;
SELECT * FROM LOAN WHERE INSTALLMENTS<40;
SELECT ACCNO,LOANAMOUNT from LOAN WHERE
STARTDATE<'2009-04-01';
```

ACCNO	LOANAMOUNT
1002	500000
1004	800000
1006	700000

8. Display the int_rate of all Loans started after 01-04-2009.

OUTPUT

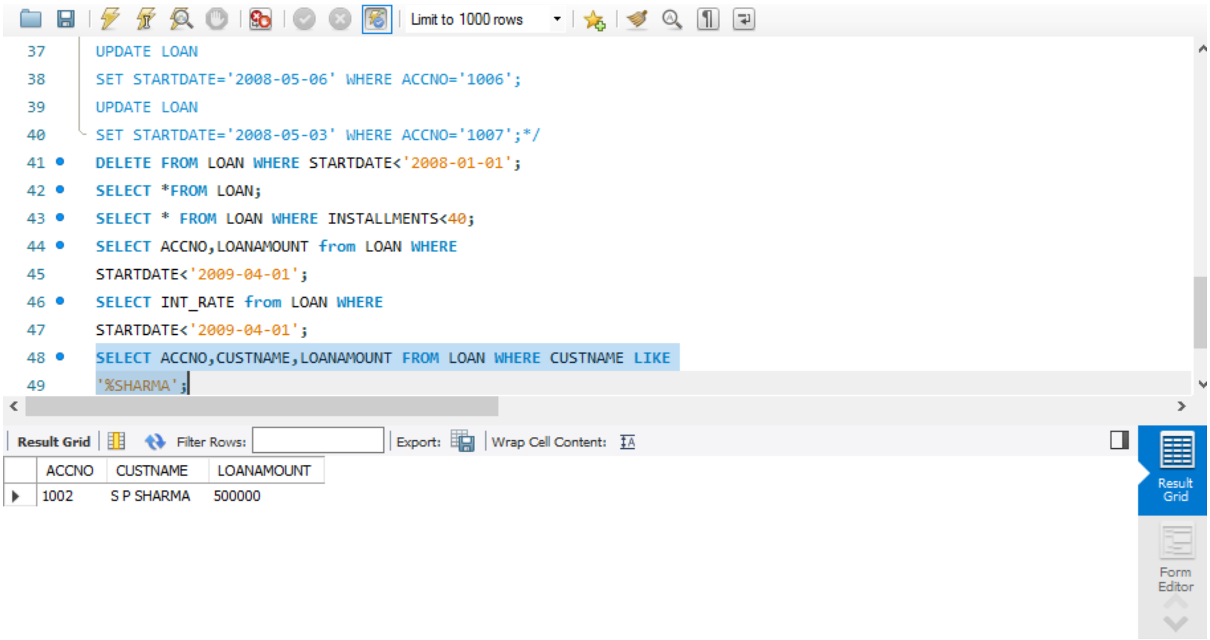


```
UPDATE LOAN
SET STARTDATE='2010-03-01' WHERE ACCNO='1005';
UPDATE LOAN
SET STARTDATE='2008-05-06' WHERE ACCNO='1006';
UPDATE LOAN
SET STARTDATE='2008-05-03' WHERE ACCNO='1007';*/
SELECT * FROM LOAN WHERE INSTALLMENTS<40;
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
```

INT_RATE
10
10
13

9. Display the Accno, cust_name and Loan amount for all the Loans for which the cust_name ends with‘Sharma’.

OUTPUT

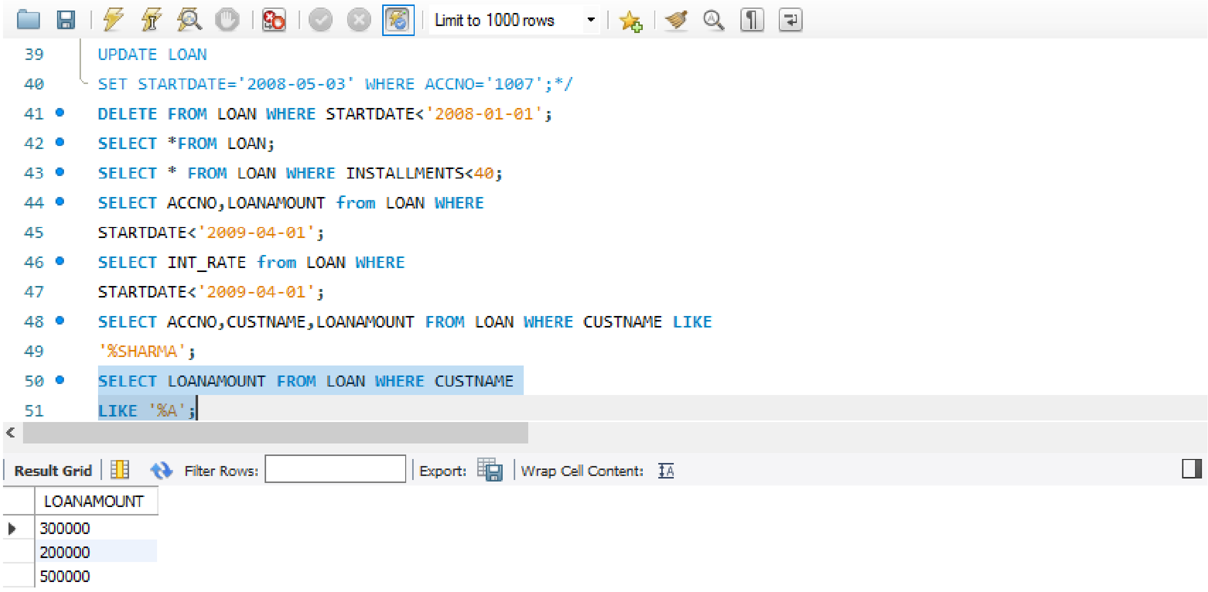


```
UPDATE LOAN
SET STARTDATE='2008-05-06' WHERE ACCNO='1006';
UPDATE LOAN
SET STARTDATE='2008-05-03' WHERE ACCNO='1007';*/
DELETE FROM LOAN WHERE STARTDATE<'2008-01-01';
SELECT *FROM LOAN;
SELECT * FROM LOAN WHERE INSTALLMENTS<40;
SELECT ACCNO,LOANAMOUNT from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
```

ACCNO	CUSTNAME	LOANAMOUNT
1002	S P SHARMA	500000

10.Loan_Amount of all the Loans for which the Cust_name ends with ‘a’.

OUTPUT

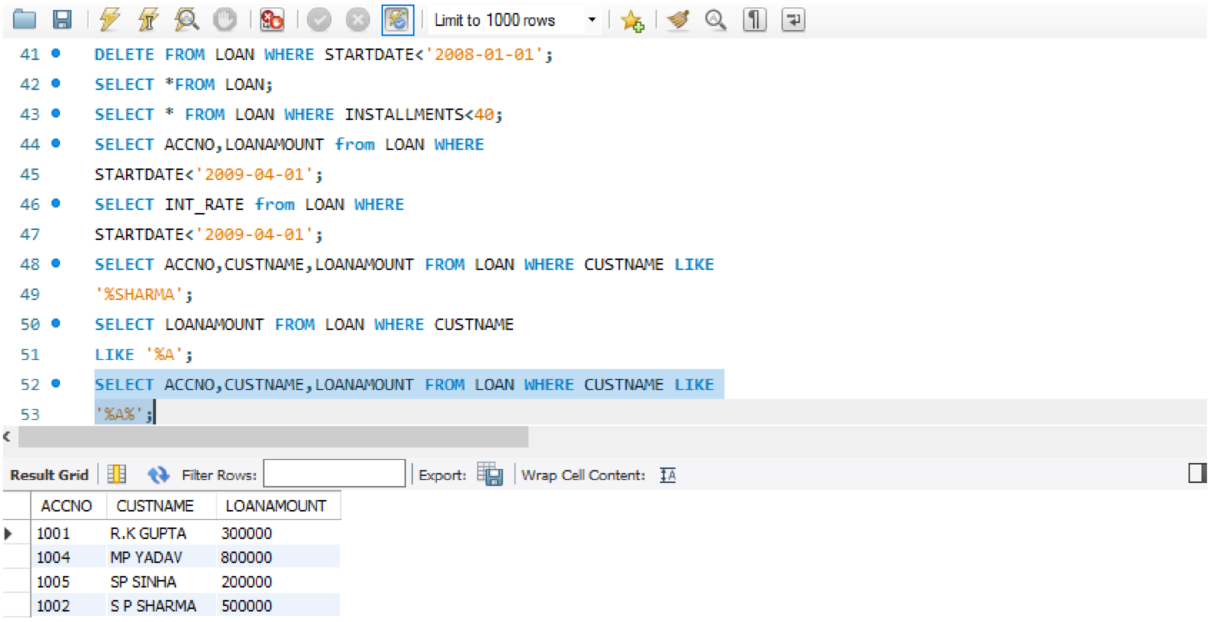


```
UPDATE LOAN
SET STARTDATE='2008-05-03' WHERE ACCNO='1007';*/
DELETE FROM LOAN WHERE STARTDATE<'2008-01-01';
SELECT *FROM LOAN;
SELECT * FROM LOAN WHERE INSTALLMENTS<40;
SELECT ACCNO,LOANAMOUNT from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
```

```
LOANAMOUNT
300000
200000
500000
```

11.Display the Accno, Cust_name and Loan_Amount for the Loans for which the Cust_name contains ‘a’.

OUTPUT

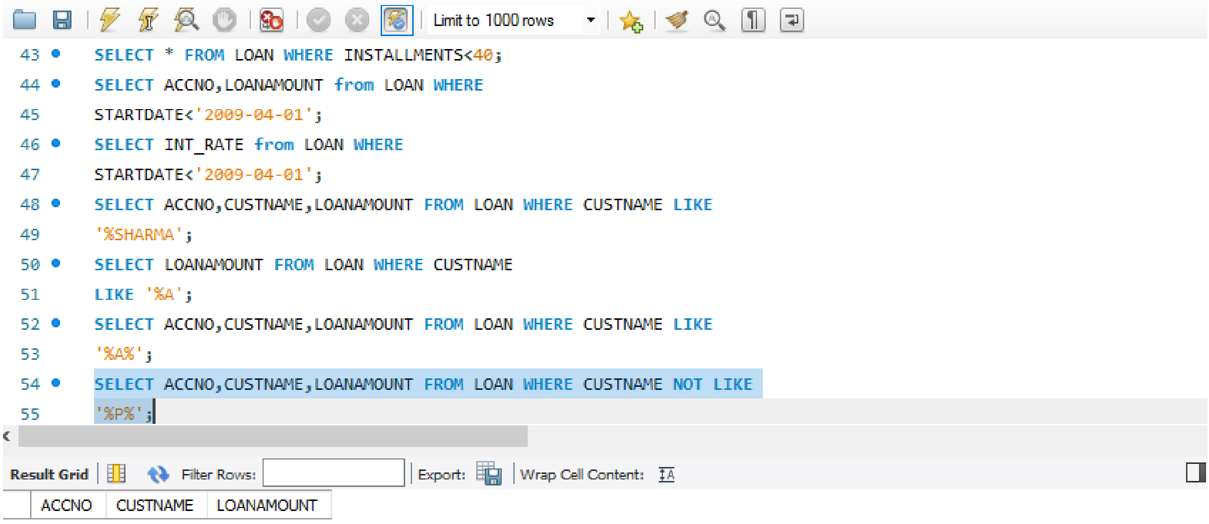


```
DELETE FROM LOAN WHERE STARTDATE<'2008-01-01';
SELECT *FROM LOAN;
SELECT * FROM LOAN WHERE INSTALLMENTS<40;
SELECT ACCNO,LOANAMOUNT from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
```

```
ACCNO   CUSTNAME     LOANAMOUNT
1001    R.K GUPTA    300000
1004    MP YADAV     800000
1005    SP SINHA     200000
1002    S P SHARMA   500000
```

12.Dsiplay the Accno, Cust_name and Loan_Amount for all the Loans for which the Cust_name does not contain ‘P’.

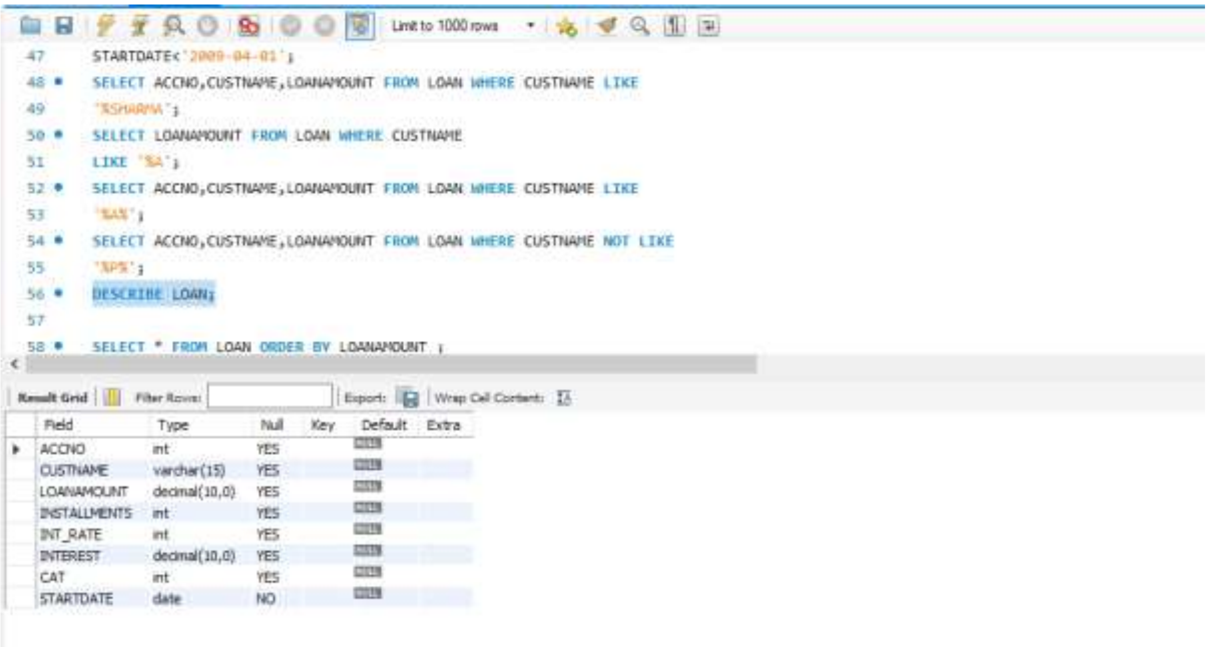
OUTPUT



```
SELECT * FROM LOAN WHERE INSTALLMENTS<40;
SELECT ACCNO,LOANAMOUNT from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME NOT LIKE
'%P%';
```

ACCNO	CUSTNAME	LOANAMOUNT

13.Display the structure of table LOAN_ACCOUNTS so that you can verify that the table is created as required.
OUTPUT

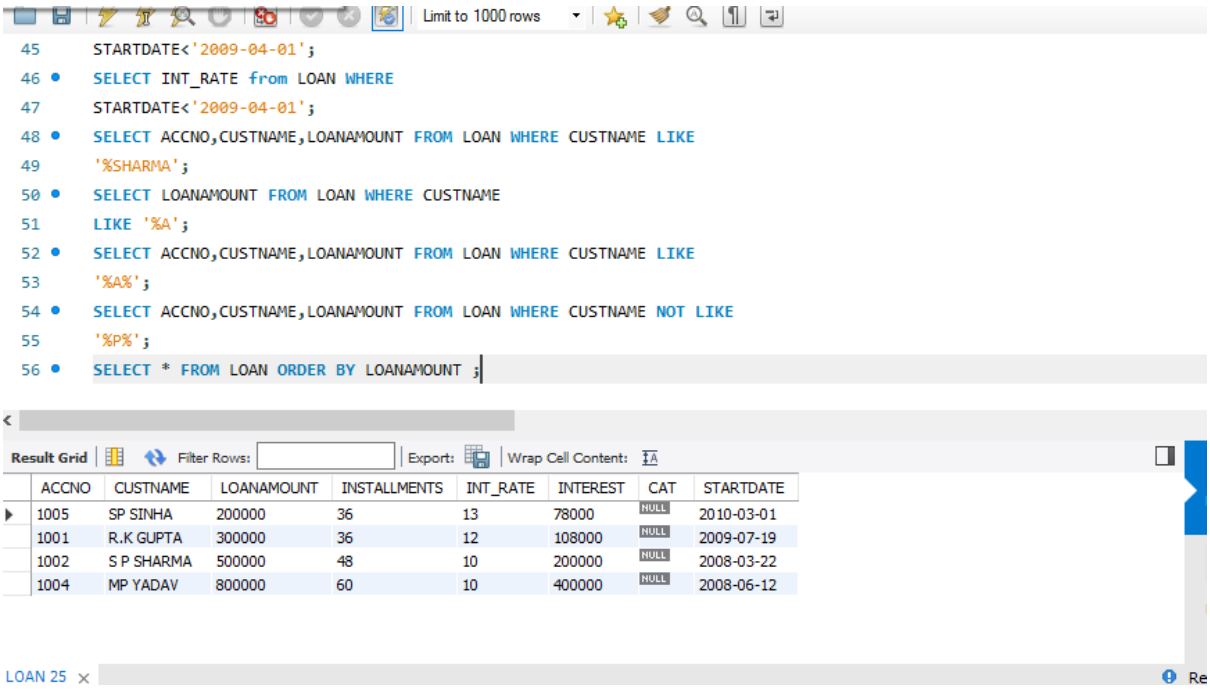


```
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME NOT LIKE
'%P%';
DESCRIBE LOAN;

SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
```

Field	Type	Null	Key	Default	Extra
ACCNO	int	YES		NULL	
CUSTNAME	varchar(15)	YES		NULL	
LOANAMOUNT	decimal(10,0)	YES		NULL	
INSTALLMENTS	int	YES		NULL	
INT_RATE	int	YES		NULL	
INTEREST	decimal(10,0)	YES		NULL	
CAT	int	YES		NULL	
STARTDATE	date	NO		NULL	

14.Display the details of all the loans in the ascending order of their Loan_Amount.

OUTPUT

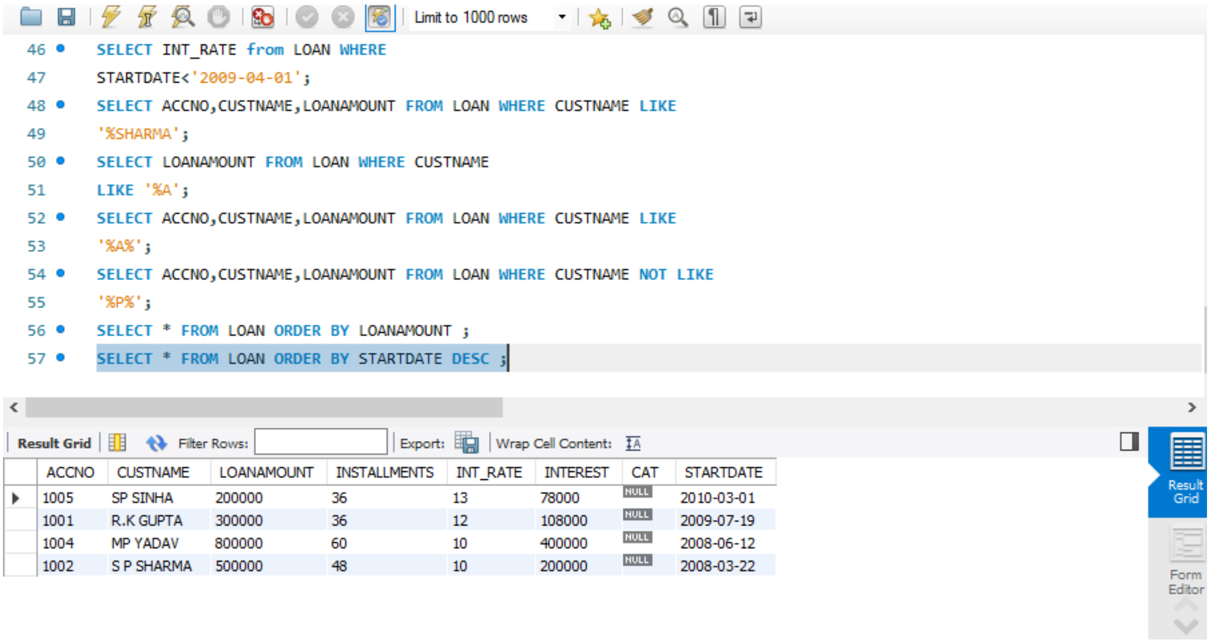


```
STARTDATE<'2009-04-01';
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME NOT LIKE
'%P%';
SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE
1005	SP SINHA	200000	36	13	78000	NULL	2010-03-01
1001	R.K GUPTA	300000	36	12	108000	NULL	2009-07-19
1002	S P SHARMA	500000	48	10	200000	NULL	2008-03-22
1004	MP YADAV	800000	60	10	400000	NULL	2008-06-12

15.Display the details of all the Loans in the descending order of their Start_date.

OUTPUT

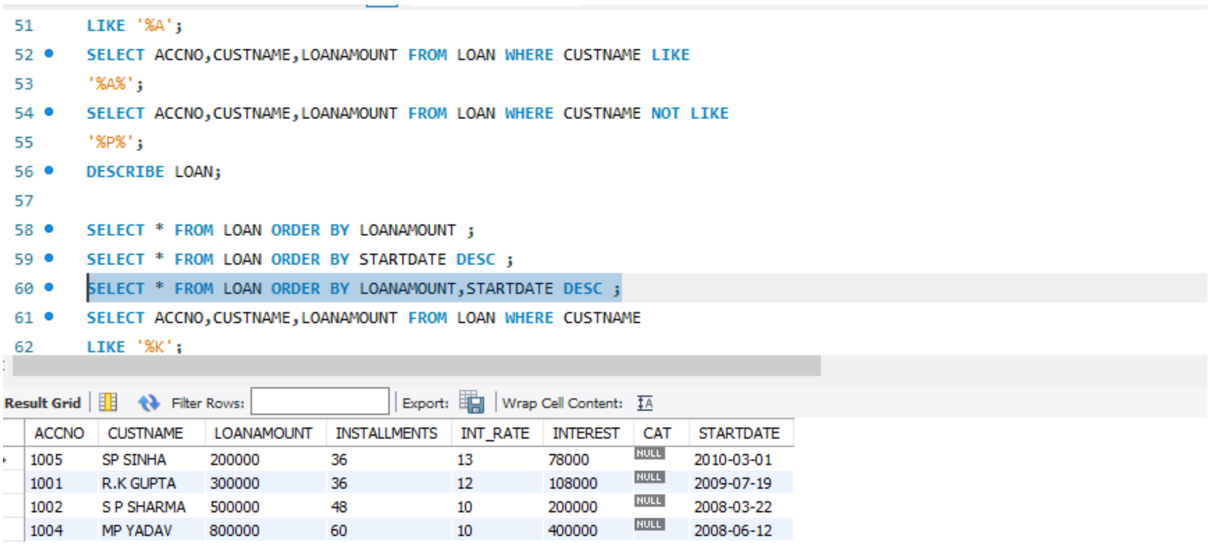


```
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME NOT LIKE
'%P%';
SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE
1005	SP SINHA	200000	36	13	78000	NULL	2010-03-01
1001	R.K GUPTA	300000	36	12	108000	NULL	2009-07-19
1004	MP YADAV	800000	60	10	400000	NULL	2008-06-12
1002	S P SHARMA	500000	48	10	200000	NULL	2008-03-22

16.Display the details of all the Loans in the ascending order of their Loan_amount and within Loan_amount in the descending order of their Start_date.

OUTPUT

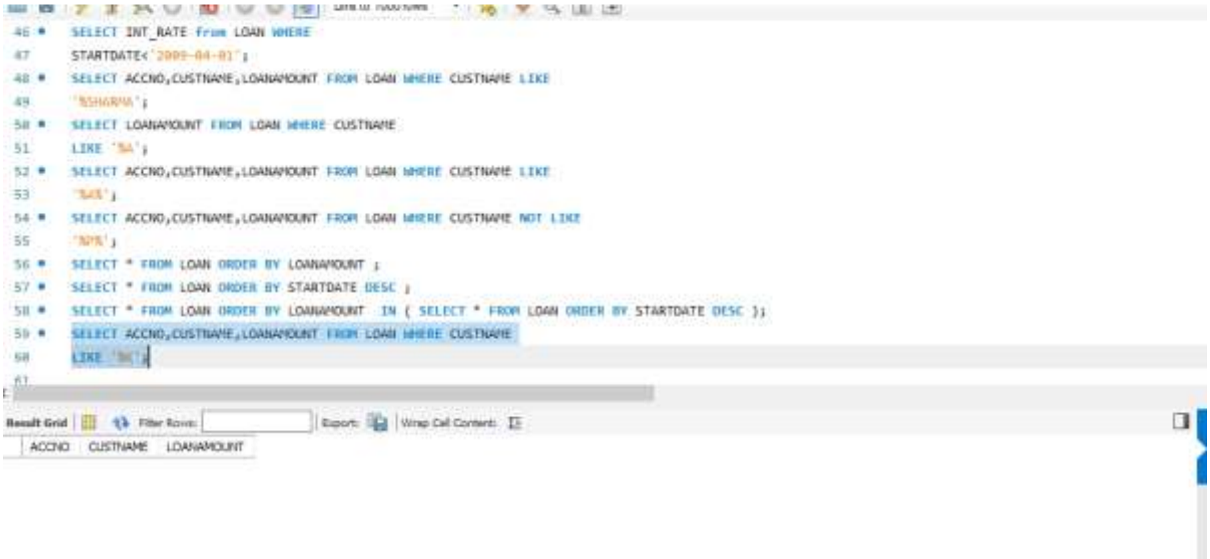


```
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME NOT LIKE
'%P%';
DESCRIBE LOAN;

SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT,STARTDATE DESC ;
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE
1005	SP SINHA	200000	36	13	78000	NULL	2010-03-01
1001	R.K GUPTA	300000	36	12	108000	NULL	2009-07-19
1002	S P SHARMA	500000	48	10	200000	NULL	2008-03-22
1004	MP YADAV	800000	60	10	400000	NULL	2008-06-12

17. Display the Accno, Cust_name and Loan_Amount of all the Loans for which the Cust_name starts with 'K'.

OUTPUT

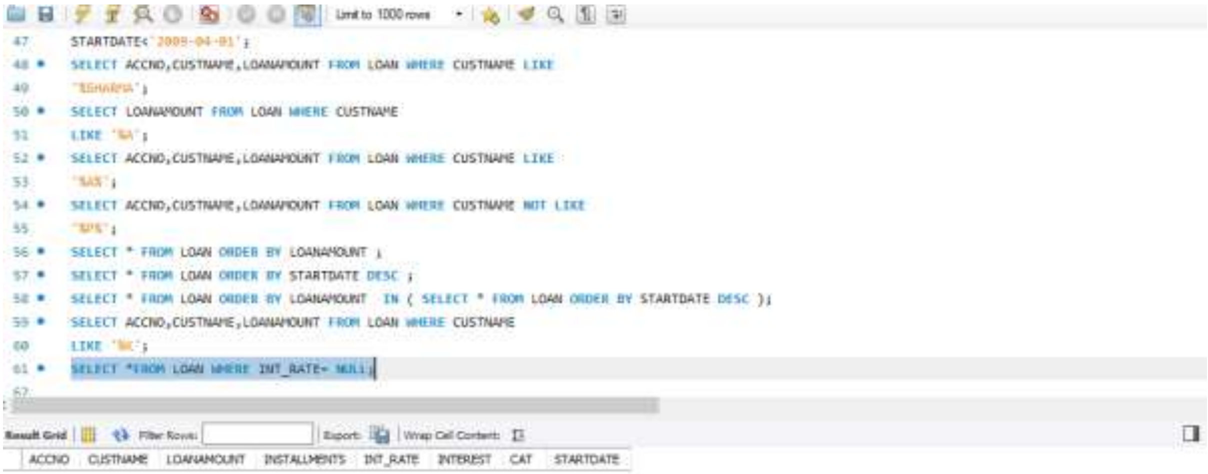


```
SELECT INT_RATE from LOAN WHERE
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME NOT LIKE
'%P%';
SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT  IN ( SELECT * FROM LOAN ORDER BY STARTDATE DESC );
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
```

ACCNO	CUSTNAME	LOANAMOUNT

18. Display the details of all the Loans whose rate of interest in NULL.

OUTPUT

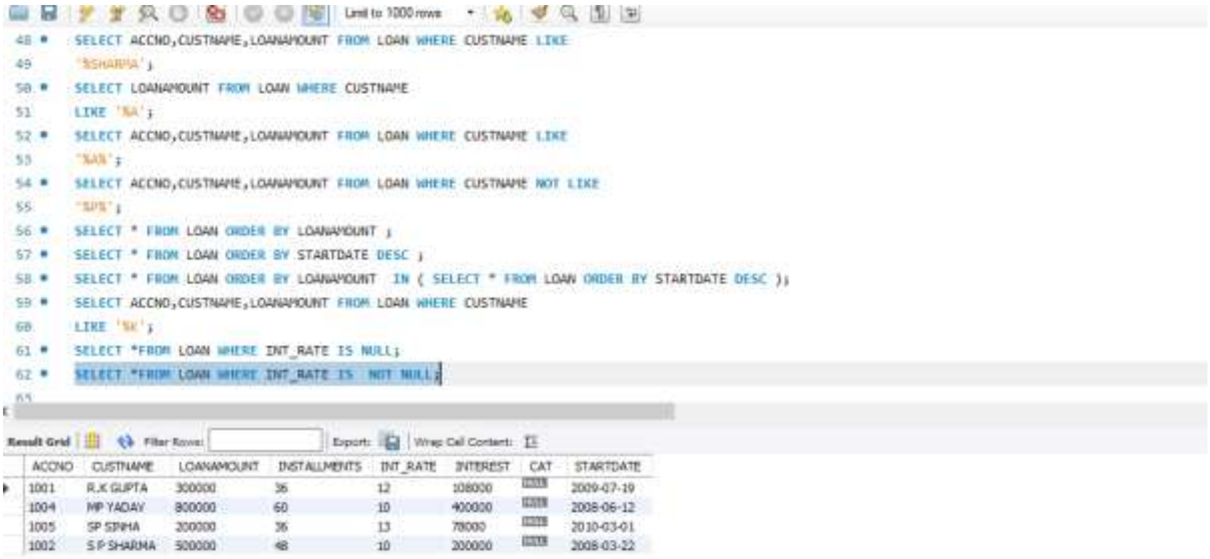


```
STARTDATE<'2009-04-01';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME NOT LIKE
'%P%';
SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT  IN ( SELECT * FROM LOAN ORDER BY STARTDATE DESC );
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE= NULL;
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE

19.Display the details of all the loans whose rate of interest is not NULL.

OUTPUT

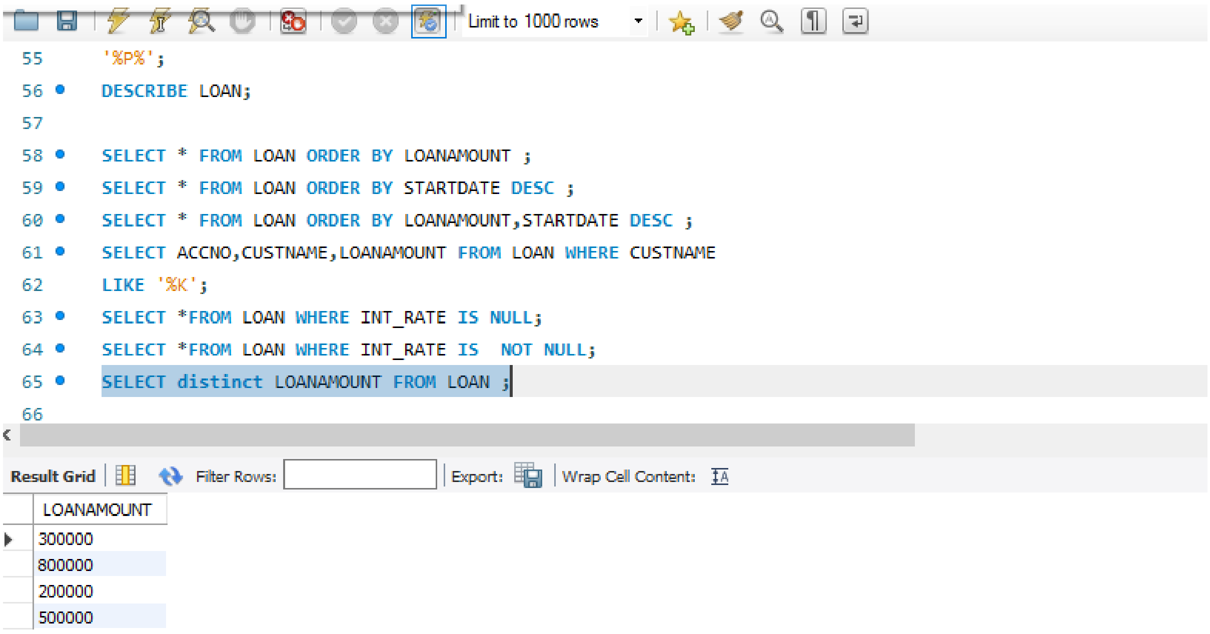


```
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%SHARMA';
SELECT LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%A';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME LIKE
'%A%';
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME NOT LIKE
'%P%';
SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT  IN ( SELECT * FROM LOAN ORDER BY STARTDATE DESC );
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE IS NULL;
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE
1001	R.K GUPTA	300000	36	12	108000	NULL	2009-07-19
1004	MP YADAV	800000	60	10	400000	NULL	2008-06-12
1005	SP SINHA	200000	36	13	78000	NULL	2010-03-01
1002	S.P SHARMA	500000	48	10	200000	NULL	2008-03-22

20.Display the amounts of various loans from the table Loan_Accounts. A Loan_Amount should appear only once.

OUTPUT

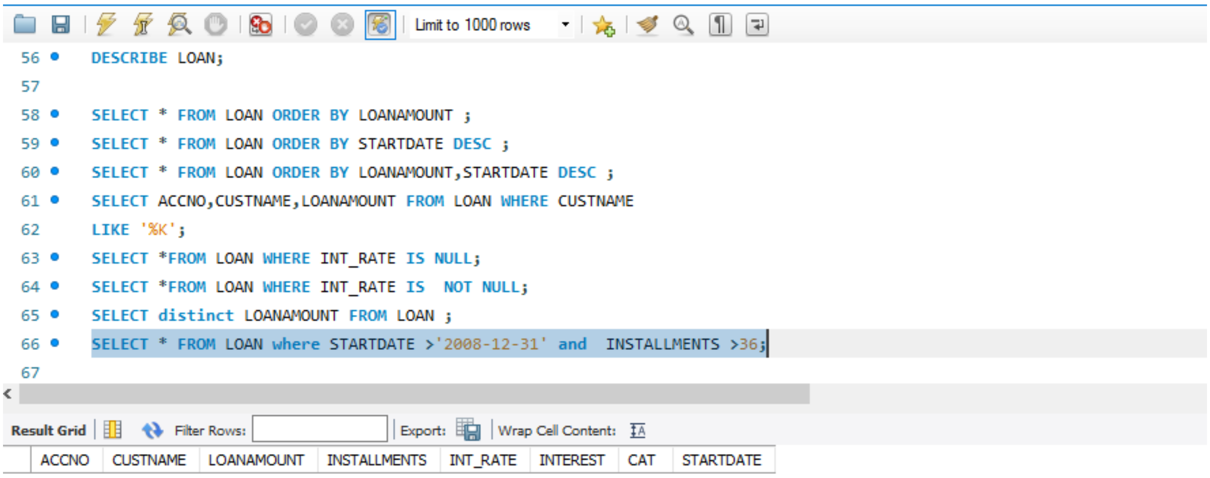


```
'%P%';
DESCRIBE LOAN;

SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT,STARTDATE DESC ;
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE IS NULL;
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
SELECT distinct LOANAMOUNT FROM LOAN ;

```

LOANAMOUNT
300000
800000
200000
500000

21.Display the details of all the loans started after 31-12-2008 for which the number of installments are more than 36.

OUTPUT

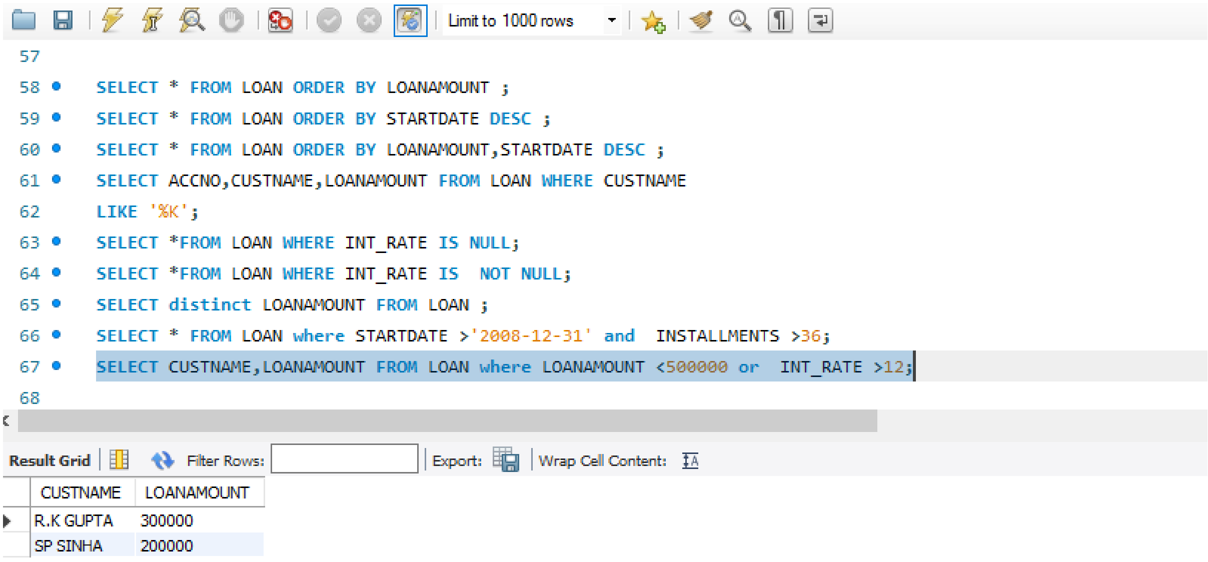


```
DESCRIBE LOAN;

SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT,STARTDATE DESC ;
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE IS NULL;
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
SELECT distinct LOANAMOUNT FROM LOAN ;
SELECT * FROM LOAN where STARTDATE >'2008-12-31' and  INSTALLMENTS >36;

```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE

22.Display the Customer_name and Loan_amount for all the Loans for which the Loan amount is less than 500000 or int_rate is more than 12.

OUTPUT

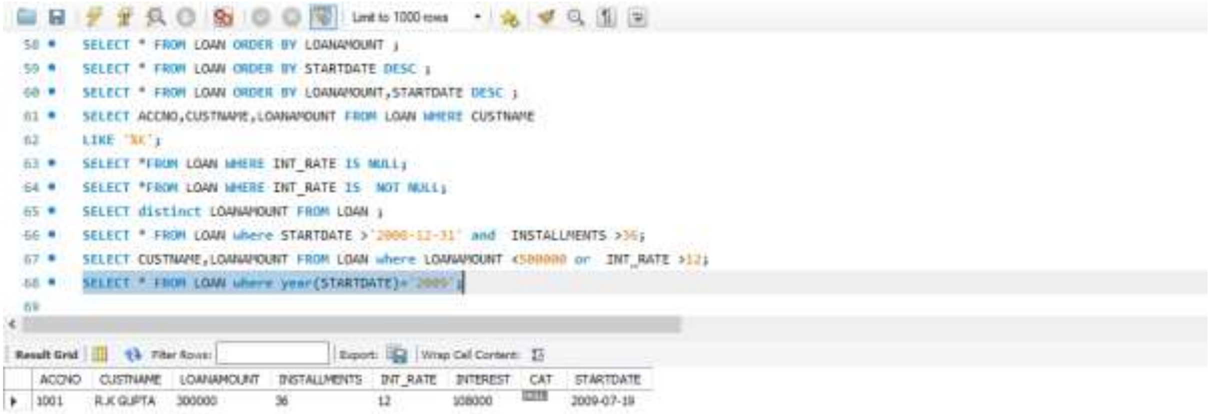


```
SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT,STARTDATE DESC ;
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE IS NULL;
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
SELECT distinct LOANAMOUNT FROM LOAN ;
SELECT * FROM LOAN where STARTDATE >'2008-12-31' and  INSTALLMENTS >36;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where LOANAMOUNT <500000 or  INT_RATE >12;
```

CUSTNAME	LOANAMOUNT
R.K GUPTA	300000
SP SINHA	200000

23.Display the details of all Loans which started in the year 2009.

OUTPUT

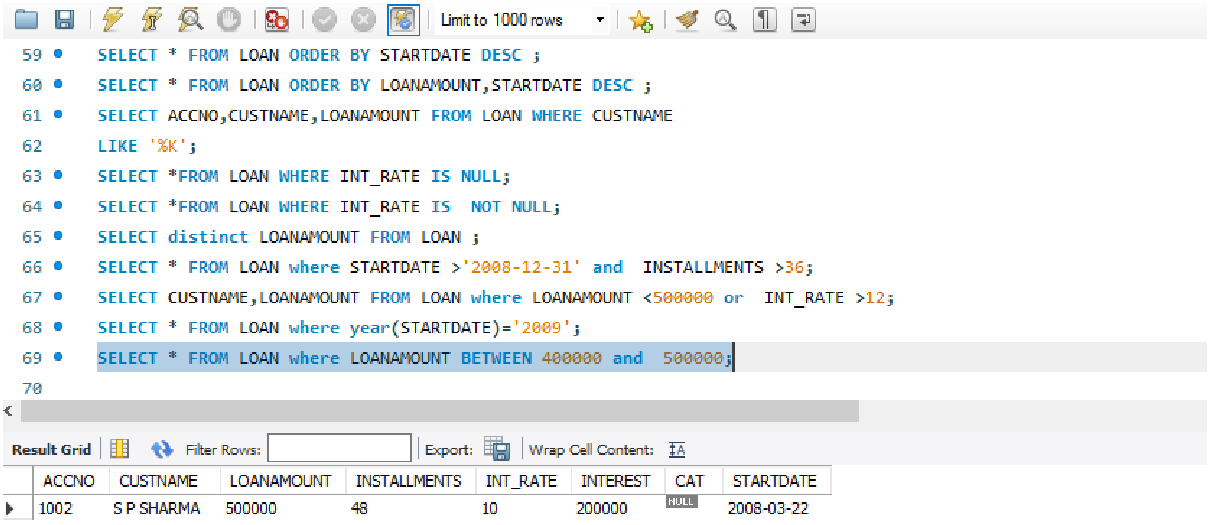


```
SELECT * FROM LOAN ORDER BY LOANAMOUNT ;
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT,STARTDATE DESC ;
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE IS NULL;
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
SELECT distinct LOANAMOUNT FROM LOAN ;
SELECT * FROM LOAN where STARTDATE >'2008-12-31' and  INSTALLMENTS >36;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where LOANAMOUNT <500000 or  INT_RATE >12;
SELECT * FROM LOAN where year(STARTDATE)='2009';
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE
1001	R.K GUPTA	300000	36	12	108000	BLOB	2009-07-19

24.Display the details of all the Loans whose Loan amount is in the Range 400000 to 500000.

OUTPUT

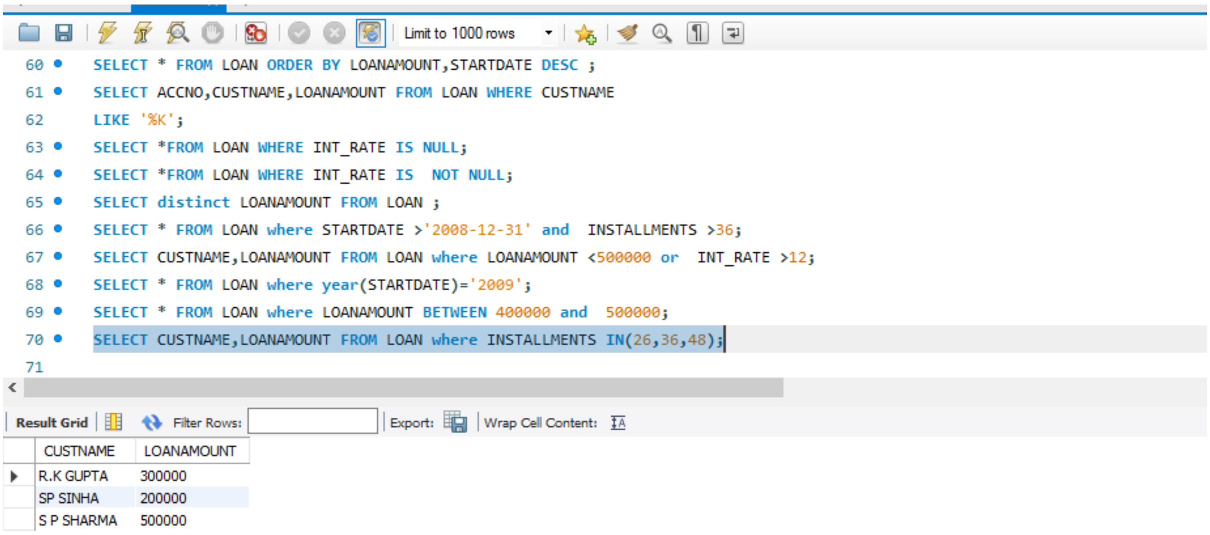


```
SELECT * FROM LOAN ORDER BY STARTDATE DESC ;
SELECT * FROM LOAN ORDER BY LOANAMOUNT,STARTDATE DESC ;
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE IS NULL;
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
SELECT distinct LOANAMOUNT FROM LOAN ;
SELECT * FROM LOAN where STARTDATE >'2008-12-31' and  INSTALLMENTS >36;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where LOANAMOUNT <500000 or  INT_RATE >12;
SELECT * FROM LOAN where year(STARTDATE)='2009';
SELECT * FROM LOAN where LOANAMOUNT BETWEEN 400000 and  500000;
```

ACCNO	CUSTNAME	LOANAMOUNT	INSTALLMENTS	INT_RATE	INTEREST	CAT	STARTDATE
1002	S P SHARMA	500000	48	10	200000	NULL	2008-03-22

25.Display the Customer_name and Loan_amount of all the Loans for which the number of installments are 26, 36 and 48.

OUTPUT

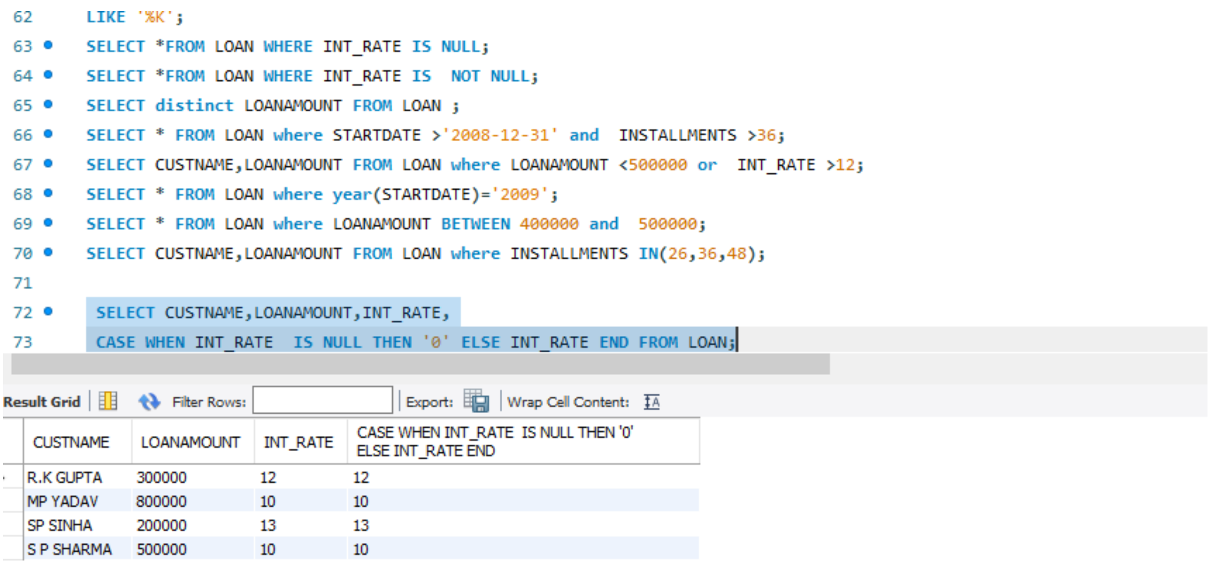


```
SELECT * FROM LOAN ORDER BY LOANAMOUNT,STARTDATE DESC ;
SELECT ACCNO,CUSTNAME,LOANAMOUNT FROM LOAN WHERE CUSTNAME
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE IS NULL;
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
SELECT distinct LOANAMOUNT FROM LOAN ;
SELECT * FROM LOAN where STARTDATE >'2008-12-31' and  INSTALLMENTS >36;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where LOANAMOUNT <500000 or  INT_RATE >12;
SELECT * FROM LOAN where year(STARTDATE)='2009';
SELECT * FROM LOAN where LOANAMOUNT BETWEEN 400000 and  500000;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where INSTALLMENTS IN(26,36,48);
```

CUSTNAME	LOANAMOUNT
R.K GUPTA	300000
SP SINHA	200000
S P SHARMA	500000

26.Display the customer name, loan_amount and interest rate. If interest rate is NULL, display it as 0.

OUTPUT

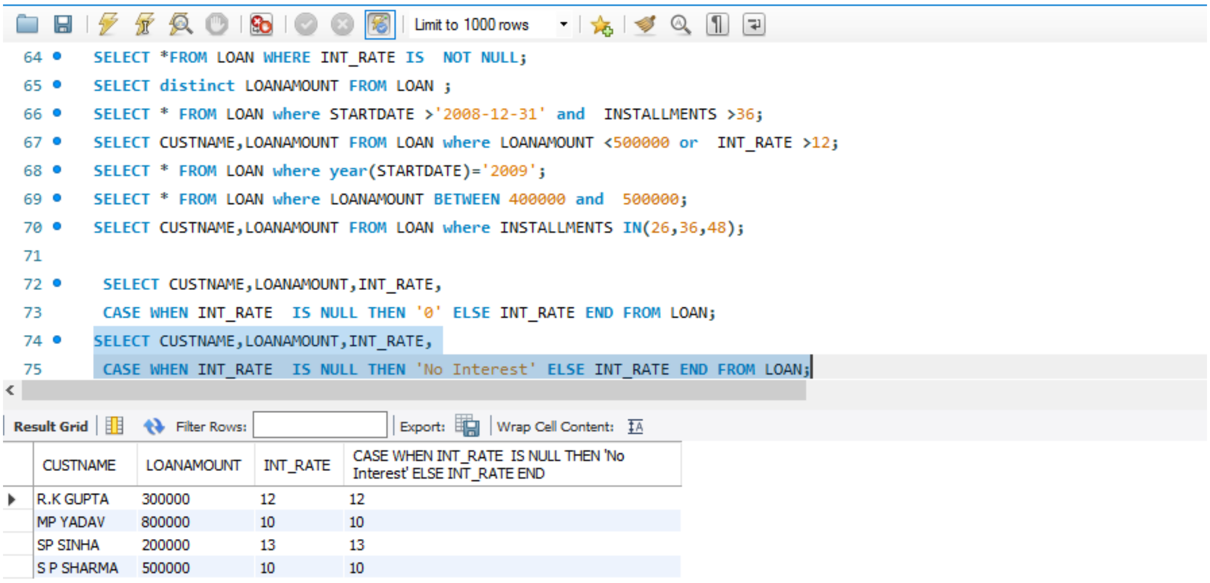


```
LIKE '%K';
SELECT *FROM LOAN WHERE INT_RATE IS NULL;
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
SELECT distinct LOANAMOUNT FROM LOAN ;
SELECT * FROM LOAN where STARTDATE >'2008-12-31' and  INSTALLMENTS >36;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where LOANAMOUNT <500000 or  INT_RATE >12;
SELECT * FROM LOAN where year(STARTDATE)='2009';
SELECT * FROM LOAN where LOANAMOUNT BETWEEN 400000 and  500000;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where INSTALLMENTS IN(26,36,48);

 SELECT CUSTNAME,LOANAMOUNT,INT_RATE,
 CASE WHEN INT_RATE  IS NULL THEN '0' ELSE INT_RATE END FROM LOAN;
```

CUSTNAME	LOANAMOUNT	INT_RATE	CASE WHEN INT_RATE IS NULL THEN '0' ELSE INT_RATE END
R.K GUPTA	300000	12	12
MP YADAV	800000	10	10
SP SINHA	200000	13	13
S P SHARMA	500000	10	10

27.Display the customer name, loan_amount and interest rate. If interest rate is NULL, display it as “No Interest”.

OUTPUT



```
SELECT *FROM LOAN WHERE INT_RATE IS  NOT NULL;
SELECT distinct LOANAMOUNT FROM LOAN ;
SELECT * FROM LOAN where STARTDATE >'2008-12-31' and  INSTALLMENTS >36;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where LOANAMOUNT <500000 or  INT_RATE >12;
SELECT * FROM LOAN where year(STARTDATE)='2009';
SELECT * FROM LOAN where LOANAMOUNT BETWEEN 400000 and  500000;
SELECT CUSTNAME,LOANAMOUNT FROM LOAN where INSTALLMENTS IN(26,36,48);

 SELECT CUSTNAME,LOANAMOUNT,INT_RATE,
 CASE WHEN INT_RATE  IS NULL THEN '0' ELSE INT_RATE END FROM LOAN;
SELECT CUSTNAME,LOANAMOUNT,INT_RATE,
 CASE WHEN INT_RATE  IS NULL THEN 'No Interest' ELSE INT_RATE END FROM LOAN;
```

CUSTNAME	LOANAMOUNT	INT_RATE	CASE WHEN INT_RATE IS NULL THEN 'No Interest' ELSE INT_RATE END
R.K GUPTA	300000	12	12
MP YADAV	800000	10	10
SP SINHA	200000	13	13
S P SHARMA	500000	10	10

LAB CYCLE II

Q.SET 1 ***date :8-6-2021***

Create the following tables and execute the queries given below

SAILORS

sid	**sname**	**rating**	**age**
22	Dustin	7	45
29	Brutas	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	26
95	Bob	3	64

BOATS

Bid	**bname**	**color**
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	Red

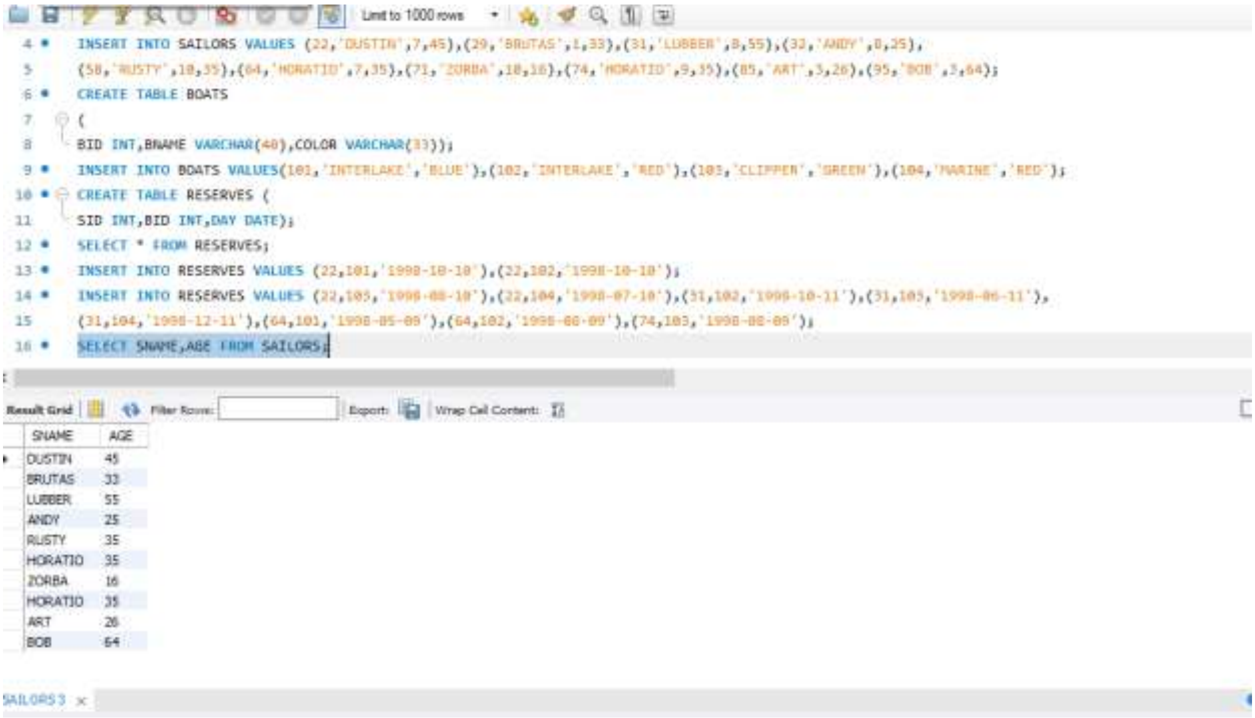
RESERVES

sid	**bid**	**day**
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98

31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

1. Find the names and ages of all sailors

OUTPUT

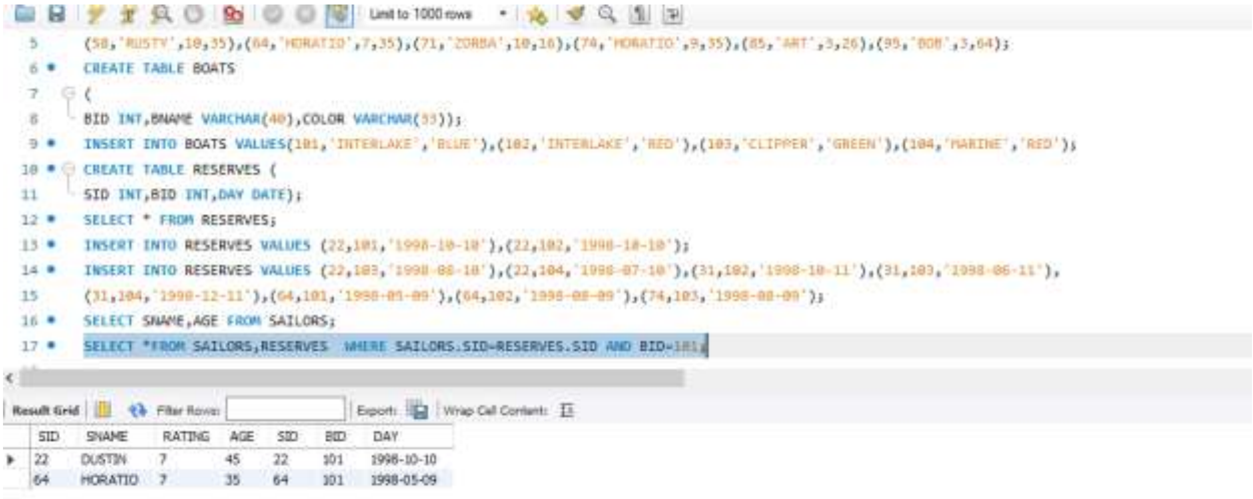


```
INSERT INTO SAILORS VALUES (22,'DUSTIN',7,45),(29,'BRUTAS',1,33),(31,'LUBBER',8,55),(32,'ANDY',8,25),
(58,'RUSTY',10,35),(64,'HORATIO',7,35),(71,'ZORBA',10,16),(74,'HORATIO',9,35),(85,'ART',3,26),(95,'BOB',3,64);
CREATE TABLE BOATS
(
BID INT,BNAME VARCHAR(40),COLOR VARCHAR(33));
INSERT INTO BOATS VALUES(101,'INTERLAKE','BLUE'),(102,'INTERLAKE','RED'),(103,'CLIPPER','GREEN'),(104,'MARINE','RED');
CREATE TABLE RESERVES (
SID INT,BID INT,DAY DATE);
SELECT * FROM RESERVES;
INSERT INTO RESERVES VALUES (22,101,'1998-10-10'),(22,102,'1998-10-10');
INSERT INTO RESERVES VALUES (22,103,'1998-08-10'),(22,104,'1998-07-10'),(31,102,'1998-10-11'),(31,103,'1998-06-11'),
(31,104,'1998-12-11'),(64,101,'1998-05-09'),(64,102,'1998-08-09'),(74,103,'1998-08-09');
SELECT SNAME,AGE FROM SAILORS;
```

SNAME	AGE
DUSTIN	45
BRUTAS	33
LUBBER	55
ANDY	25
RUSTY	35
HORATIO	35
ZORBA	16
HORATIO	35
ART	26
BOB	64

2. Find all information of sailors who have reserved boat number 101.

OUTPUT

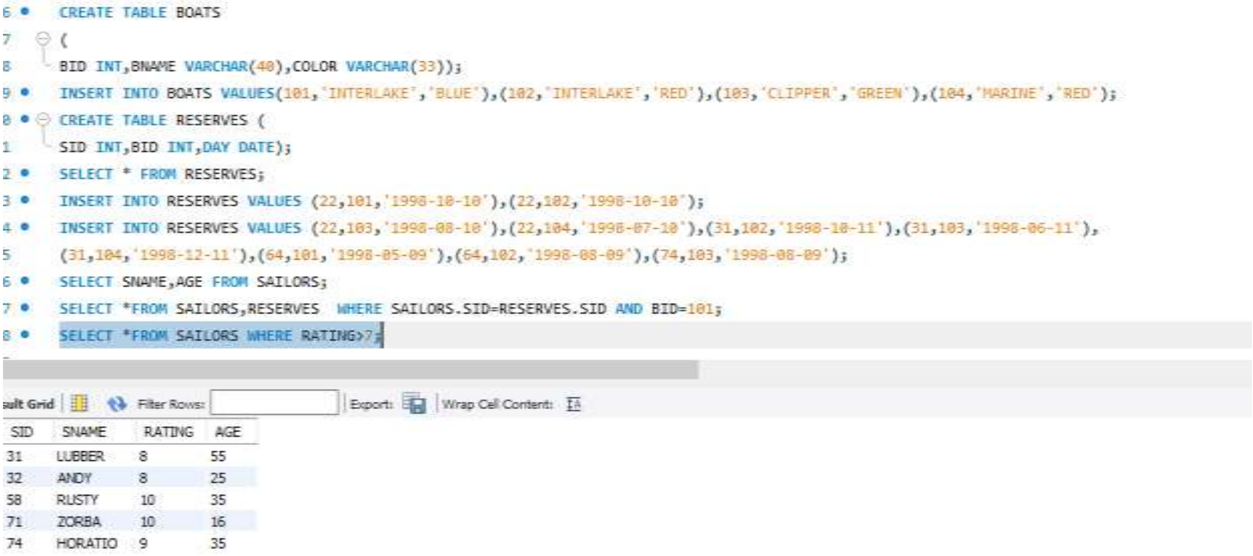


```sql
(58,'RUSTY',10,35),(64,'HORATIO',7,35),(71,'ZORBA',10,16),(74,'HORATIO',9,35),(85,'ART',3,26),(95,'BOB',3,64);
CREATE TABLE BOATS
(
BID INT,BNAME VARCHAR(40),COLOR VARCHAR(33));
INSERT INTO BOATS VALUES(101,'INTERLAKE','BLUE'),(102,'INTERLAKE','RED'),(103,'CLIPPER','GREEN'),(104,'MARINE','RED');
CREATE TABLE RESERVES (
SID INT,BID INT,DAY DATE);
SELECT * FROM RESERVES;
INSERT INTO RESERVES VALUES (22,101,'1998-10-10'),(22,102,'1998-10-10');
INSERT INTO RESERVES VALUES (22,103,'1998-08-10'),(22,104,'1998-07-10'),(31,102,'1998-10-11'),(31,103,'1998-06-11'),
(31,104,'1998-12-11'),(64,101,'1998-05-09'),(64,102,'1998-08-09'),(74,103,'1998-08-09');
SELECT SNAME,AGE FROM SAILORS;
SELECT *FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=101;
```

SID	SNAME	RATING	AGE	SID	BID	DAY
22	DUSTIN	7	45	22	101	1998-10-10
64	HORATIO	7	35	64	101	1998-05-09

3. Find all sailors with rating above 7

OUTPUT

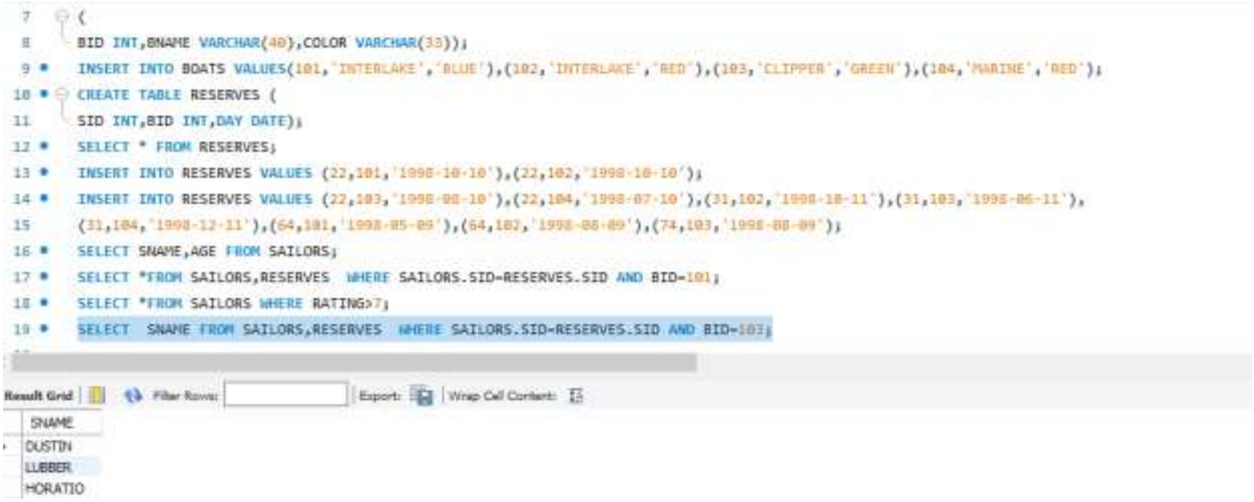


```sql
CREATE TABLE BOATS
(
BID INT,BNAME VARCHAR(40),COLOR VARCHAR(33));
INSERT INTO BOATS VALUES(101,'INTERLAKE','BLUE'),(102,'INTERLAKE','RED'),(103,'CLIPPER','GREEN'),(104,'MARINE','RED');
CREATE TABLE RESERVES (
SID INT,BID INT,DAY DATE);
SELECT * FROM RESERVES;
INSERT INTO RESERVES VALUES (22,101,'1998-10-10'),(22,102,'1998-10-10');
INSERT INTO RESERVES VALUES (22,103,'1998-08-10'),(22,104,'1998-07-10'),(31,102,'1998-10-11'),(31,103,'1998-06-11'),
(31,104,'1998-12-11'),(64,101,'1998-05-09'),(64,102,'1998-08-09'),(74,103,'1998-08-09');
SELECT SNAME,AGE FROM SAILORS;
SELECT *FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=101;
SELECT *FROM SAILORS WHERE RATING>7;
```

SID	SNAME	RATING	AGE
31	LUBBER	8	55
32	ANDY	8	25
58	RUSTY	10	35
71	ZORBA	10	16
74	HORATIO	9	35

4. Find the names of sailors who have reserved boat no 103

OUTPUT

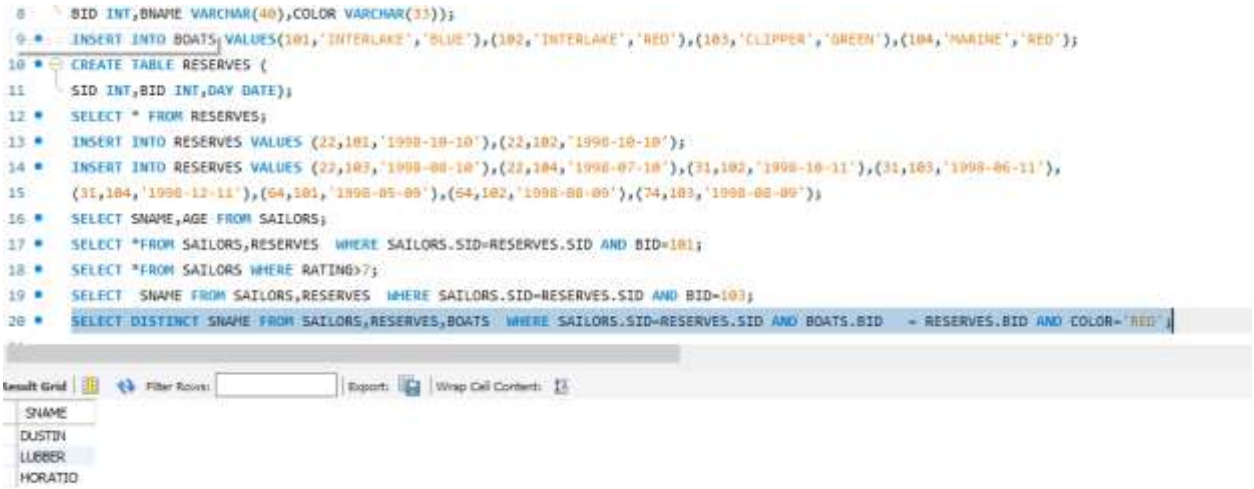


```sql
(
BID INT,BNAME VARCHAR(40),COLOR VARCHAR(33));
INSERT INTO BOATS VALUES(101,'INTERLAKE','BLUE'),(102,'INTERLAKE','RED'),(103,'CLIPPER','GREEN'),(104,'MARINE','RED');
CREATE TABLE RESERVES (
SID INT,BID INT,DAY DATE);
SELECT * FROM RESERVES;
INSERT INTO RESERVES VALUES (22,101,'1998-10-10'),(22,102,'1998-10-10');
INSERT INTO RESERVES VALUES (22,103,'1998-08-10'),(22,104,'1998-07-10'),(31,102,'1998-10-11'),(31,103,'1998-06-11'),
(31,104,'1998-12-11'),(64,101,'1998-05-09'),(64,102,'1998-08-09'),(74,103,'1998-08-09');
SELECT SNAME,AGE FROM SAILORS;
SELECT *FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=101;
SELECT *FROM SAILORS WHERE RATING>7;
SELECT  SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=103;
```

```
SNAME
DUSTIN
LUBBER
HORATIO
```

5. Find the names of sailors who have reserved a red boat, and list in the order of age.

OUTPUT

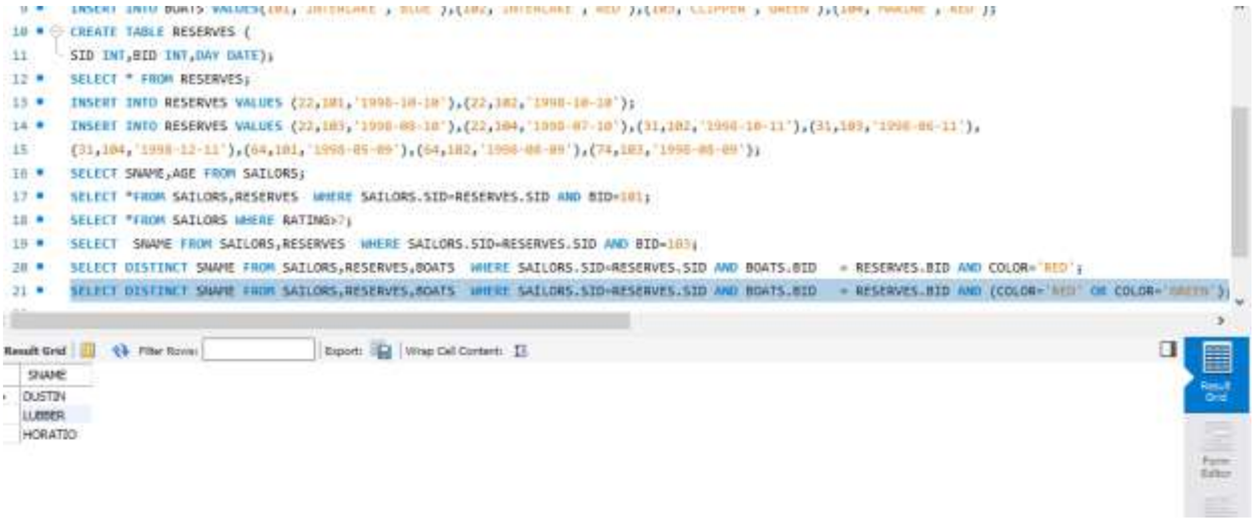


```sql
BID INT,BNAME VARCHAR(40),COLOR VARCHAR(33));
INSERT INTO BOATS VALUES(101,'INTERLAKE','BLUE'),(102,'INTERLAKE','RED'),(103,'CLIPPER','GREEN'),(104,'MARINE','RED');
CREATE TABLE RESERVES (
SID INT,BID INT,DAY DATE);
SELECT * FROM RESERVES;
INSERT INTO RESERVES VALUES (22,101,'1998-10-10'),(22,102,'1998-10-10');
INSERT INTO RESERVES VALUES (22,103,'1998-08-10'),(22,104,'1998-07-10'),(31,102,'1998-10-11'),(31,103,'1998-06-11'),
(31,104,'1998-12-11'),(64,101,'1998-05-09'),(64,102,'1998-08-09'),(74,103,'1998-08-09');
SELECT SNAME,AGE FROM SAILORS;
SELECT *FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=101;
SELECT *FROM SAILORS WHERE RATING>7;
SELECT  SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=103;
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND COLOR='RED';
```

```
SNAME
DUSTIN
LUBBER
HORATIO
```

6. Find the names of sailors who have reserved either a red or green boat

OUTPUT

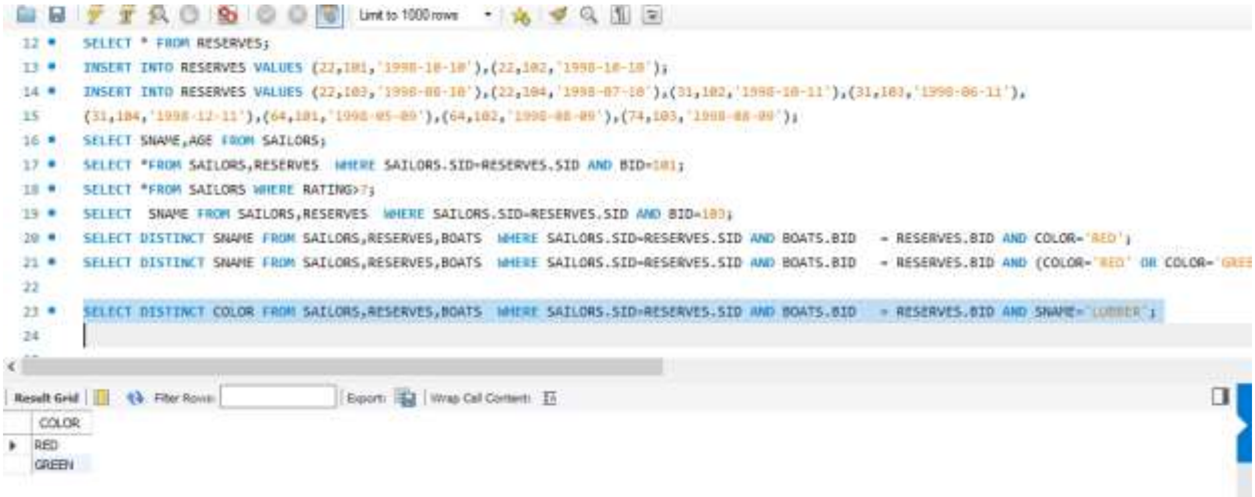


```
INSERT INTO BOATS VALUES(101,'INTERLAKE','BLUE'),(102,'INTERLAKE','RED'),(103,'CLIPPER','GREEN'),(104,'MARINE','RED');
CREATE TABLE RESERVES (
SID INT,BID INT,DAY DATE);
SELECT * FROM RESERVES;
INSERT INTO RESERVES VALUES (22,101,'1998-10-10'),(22,102,'1998-10-10');
INSERT INTO RESERVES VALUES (22,103,'1998-08-10'),(22,104,'1998-07-10'),(31,102,'1998-10-11'),(31,103,'1998-06-11'),
(31,104,'1998-12-11'),(64,101,'1998-05-09'),(64,102,'1998-08-09'),(74,103,'1998-08-09');
SELECT SNAME,AGE FROM SAILORS;
SELECT *FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=101;
SELECT *FROM SAILORS WHERE RATING>7;
SELECT  SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=103;
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND COLOR='RED';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND (COLOR='RED' OR COLOR='GREEN');
```

```
SNAME
DUSTIN
LUBBER
HORATIO
```

7. Find the colors of boats reserved by “Lubber”.

OUTPUT

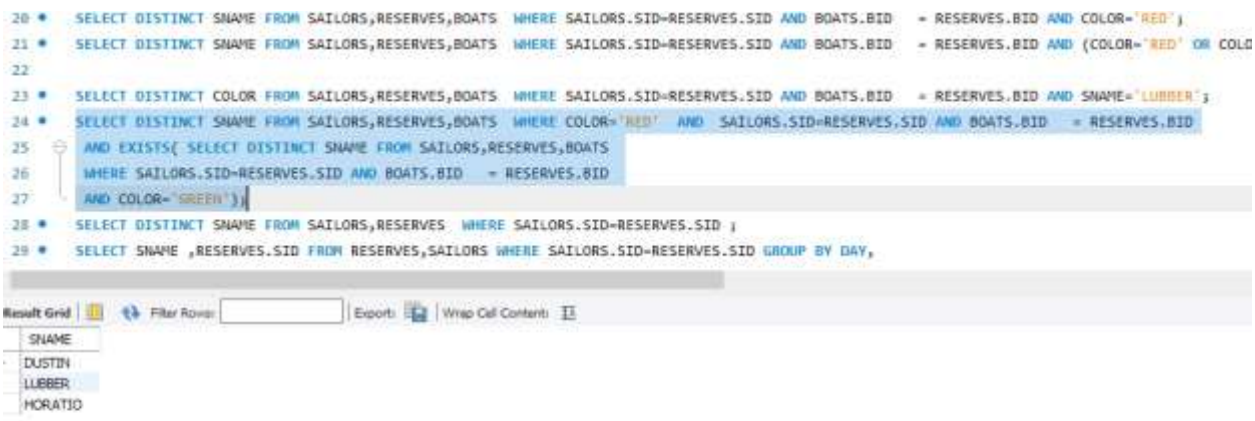


```
SELECT * FROM RESERVES;
INSERT INTO RESERVES VALUES (22,101,'1998-10-10'),(22,102,'1998-10-10');
INSERT INTO RESERVES VALUES (22,103,'1998-08-10'),(22,104,'1998-07-10'),(31,102,'1998-10-11'),(31,103,'1998-06-11'),
(31,104,'1998-12-11'),(64,101,'1998-05-09'),(64,102,'1998-08-09'),(74,103,'1998-08-09');
SELECT SNAME,AGE FROM SAILORS;
SELECT *FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=101;
SELECT *FROM SAILORS WHERE RATING>7;
SELECT  SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=103;
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND COLOR='RED';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND (COLOR='RED' OR COLOR='GREE

SELECT DISTINCT COLOR FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND SNAME='LUBBER';
```

```
COLOR
RED
GREEN
```

8. Find the names of sailors who have reserved both red and green boats

OUTPUT

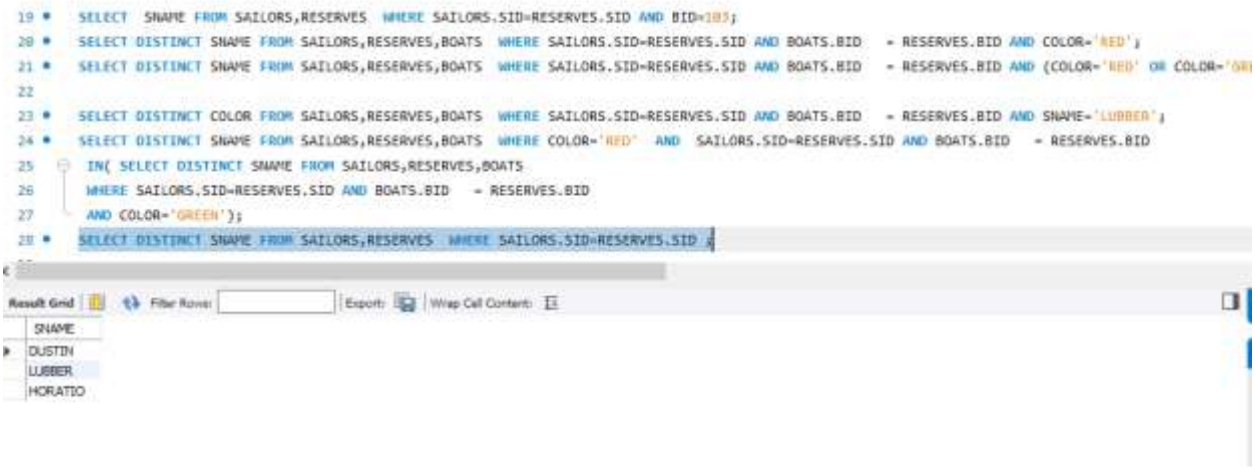


```sql
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND COLOR='RED';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND (COLOR='RED' OR COLO

SELECT DISTINCT COLOR FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND SNAME='LUBBER';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE COLOR='RED'  AND  SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 AND EXISTS( SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS
 WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 AND COLOR='GREEN');
SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
SELECT SNAME ,RESERVES.SID FROM RESERVES,SAILORS WHERE SAILORS.SID=RESERVES.SID GROUP BY DAY,
```

```
SNAME
DUSTIN
LUBBER
HORATIO
```

9. Find the names of sailors who have reserved at least one boat

OUTPUT

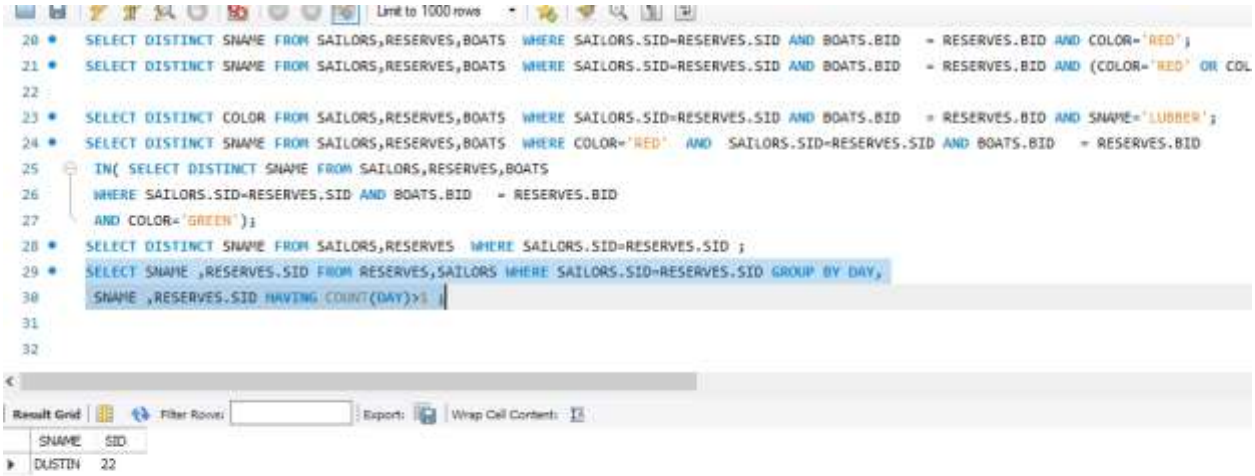


```sql
SELECT  SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=103;
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND COLOR='RED';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND (COLOR='RED' OR COLOR='GR

SELECT DISTINCT COLOR FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND SNAME='LUBBER';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE COLOR='RED'  AND  SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 IN( SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS
 WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 AND COLOR='GREEN');
SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
```

```
SNAME
DUSTIN
LUBBER
HORATIO
```

10. Find the ids and names of sailors who have reserved two different boats on the same day.

OUTPUT

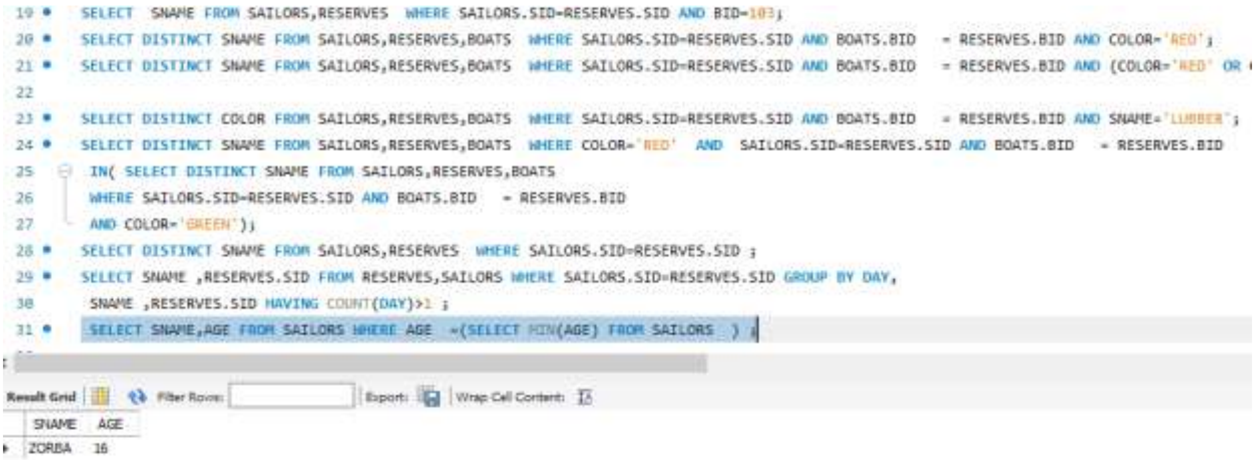


```
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND COLOR='RED';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND (COLOR='RED' OR COL

SELECT DISTINCT COLOR FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND SNAME='LUBBER';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE COLOR='RED'  AND  SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 IN( SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS
 WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 AND COLOR='GREEN');
SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
SELECT SNAME ,RESERVES.SID FROM RESERVES,SAILORS WHERE SAILORS.SID=RESERVES.SID GROUP BY DAY,
 SNAME ,RESERVES.SID HAVING COUNT(DAY)>1 ;
```

```
  SNAME   SID
▸ DUSTIN  22
```

11.Find the name and the age of the youngest sailor.

OUTPUT

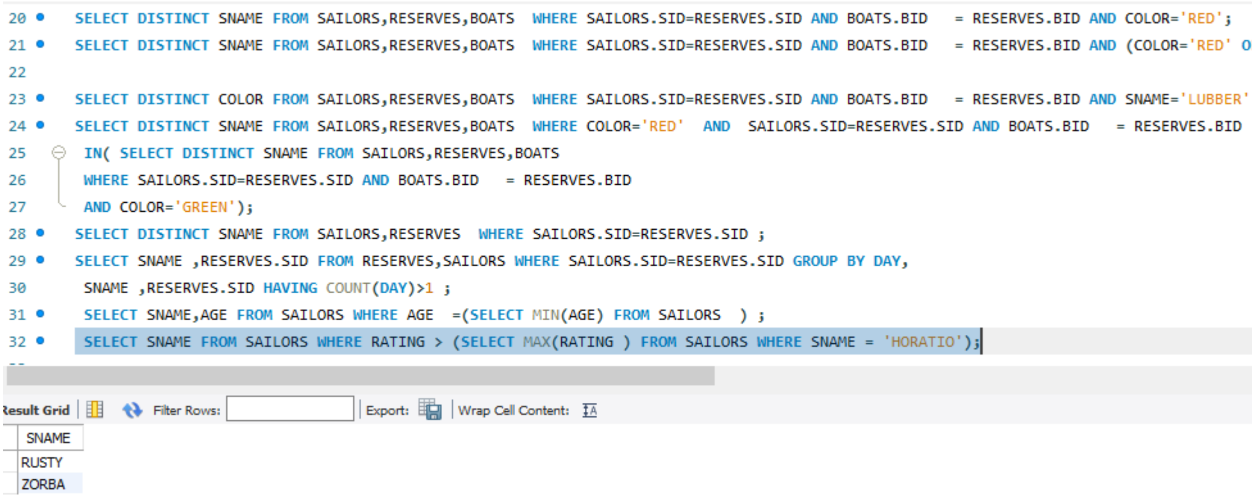


```
SELECT  SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID AND BID=103;
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND COLOR='RED';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND (COLOR='RED' OR (

SELECT DISTINCT COLOR FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND SNAME='LUBBER';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE COLOR='RED'  AND  SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 IN( SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS
 WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 AND COLOR='GREEN');
SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
SELECT SNAME ,RESERVES.SID FROM RESERVES,SAILORS WHERE SAILORS.SID=RESERVES.SID GROUP BY DAY,
 SNAME ,RESERVES.SID HAVING COUNT(DAY)>1 ;
 SELECT SNAME,AGE FROM SAILORS WHERE AGE  =(SELECT MIN(AGE) FROM SAILORS  ) ;
```

```
  SNAME   AGE
▸ ZORBA   16
```

12.Find the names and ratings of sailor whose rating is better than some sailor called Horatio.

OUTPUT

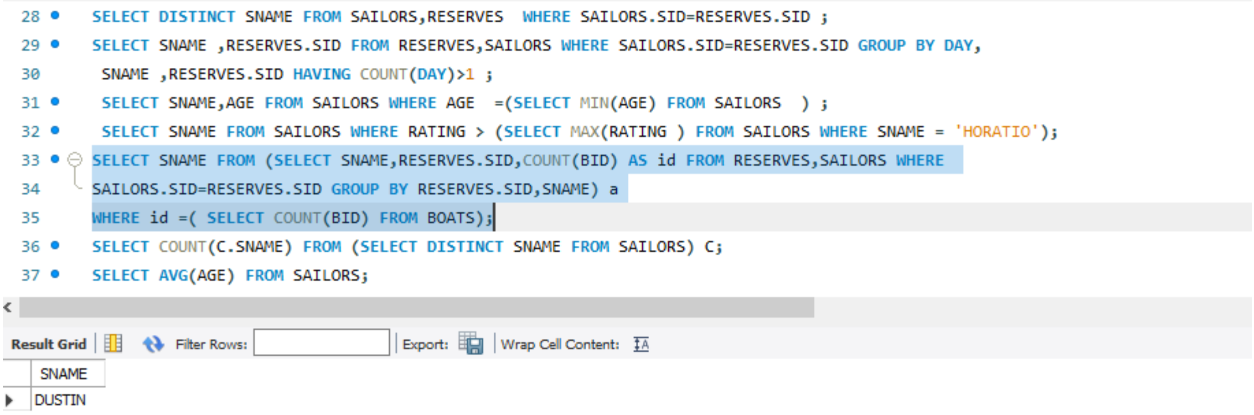


```
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND COLOR='RED';
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND (COLOR='RED' O

SELECT DISTINCT COLOR FROM SAILORS,RESERVES,BOATS  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID AND SNAME='LUBBER'
SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS  WHERE COLOR='RED'  AND  SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 IN( SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS
 WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 AND COLOR='GREEN');
SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
SELECT SNAME ,RESERVES.SID FROM RESERVES,SAILORS WHERE SAILORS.SID=RESERVES.SID GROUP BY DAY,
 SNAME ,RESERVES.SID HAVING COUNT(DAY)>1 ;
 SELECT SNAME,AGE FROM SAILORS WHERE AGE  =(SELECT MIN(AGE) FROM SAILORS  ) ;
 SELECT SNAME FROM SAILORS WHERE RATING > (SELECT MAX(RATING ) FROM SAILORS WHERE SNAME = 'HORATIO');
```

SNAME
RUSTY
ZORBA

13.Find the names of sailors who have reserved all boats.

OUTPUT

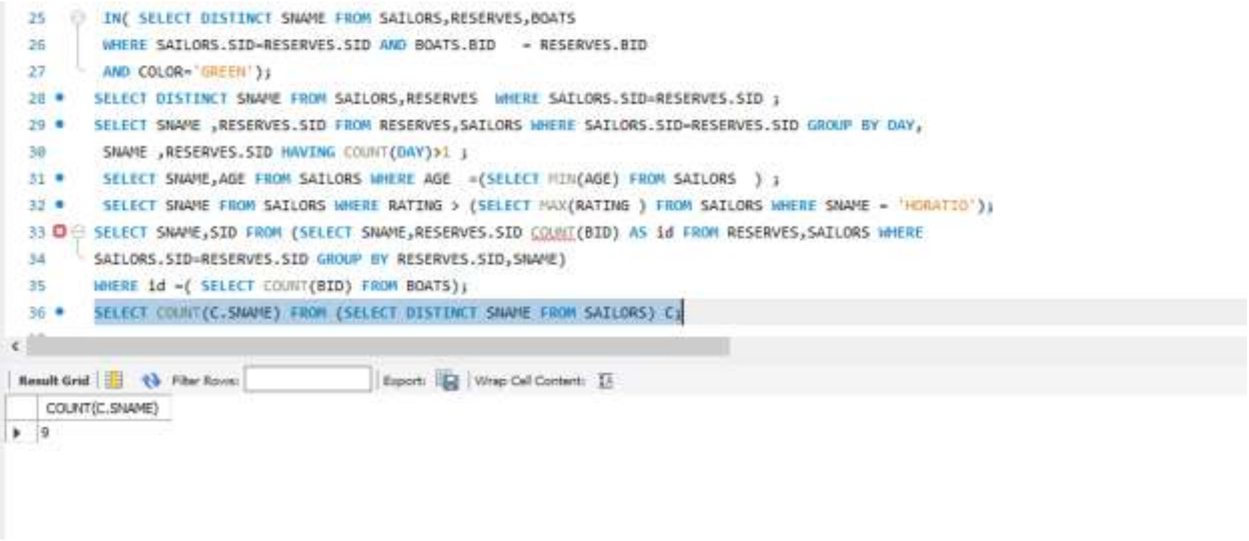


```
SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
SELECT SNAME ,RESERVES.SID FROM RESERVES,SAILORS WHERE SAILORS.SID=RESERVES.SID GROUP BY DAY,
 SNAME ,RESERVES.SID HAVING COUNT(DAY)>1 ;
 SELECT SNAME,AGE FROM SAILORS WHERE AGE  =(SELECT MIN(AGE) FROM SAILORS  ) ;
 SELECT SNAME FROM SAILORS WHERE RATING > (SELECT MAX(RATING ) FROM SAILORS WHERE SNAME = 'HORATIO');
SELECT SNAME FROM (SELECT SNAME,RESERVES.SID,COUNT(BID) AS id FROM RESERVES,SAILORS WHERE
SAILORS.SID=RESERVES.SID GROUP BY RESERVES.SID,SNAME) a
WHERE id =( SELECT COUNT(BID) FROM BOATS);
SELECT COUNT(C.SNAME) FROM (SELECT DISTINCT SNAME FROM SAILORS) C;
SELECT AVG(AGE) FROM SAILORS;
```

SNAME
DUSTIN

14.Count the number of different sailor names.

OUTPUT

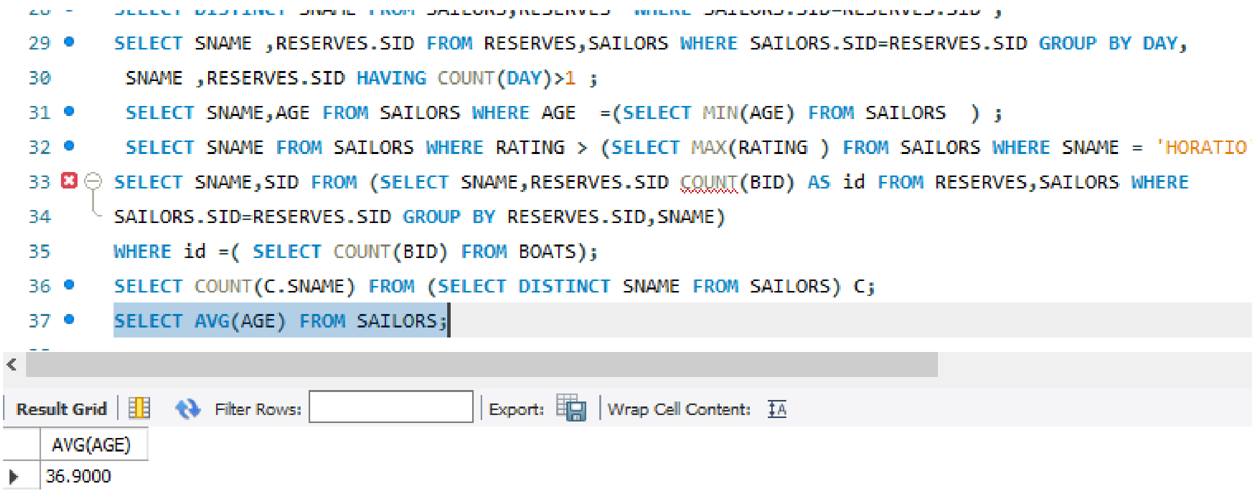


```sql
 IN( SELECT DISTINCT SNAME FROM SAILORS,RESERVES,BOATS
 WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
 AND COLOR='GREEN');
SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
SELECT SNAME ,RESERVES.SID FROM RESERVES,SAILORS WHERE SAILORS.SID=RESERVES.SID GROUP BY DAY,
 SNAME ,RESERVES.SID HAVING COUNT(DAY)>1 ;
 SELECT SNAME,AGE FROM SAILORS WHERE AGE  =(SELECT MIN(AGE) FROM SAILORS  ) ;
 SELECT SNAME FROM SAILORS WHERE RATING > (SELECT MAX(RATING ) FROM SAILORS WHERE SNAME = 'HORATIO'));
SELECT SNAME,SID FROM (SELECT SNAME,RESERVES.SID COUNT(BID) AS id FROM RESERVES,SAILORS WHERE
SAILORS.SID=RESERVES.SID GROUP BY RESERVES.SID,SNAME)
WHERE id =( SELECT COUNT(BID) FROM BOATS);
SELECT COUNT(C.SNAME) FROM (SELECT DISTINCT SNAME FROM SAILORS) C;
```

```
COUNT(C.SNAME)
9
```

15.Calculate the average age of all sailors.

OUTPUT

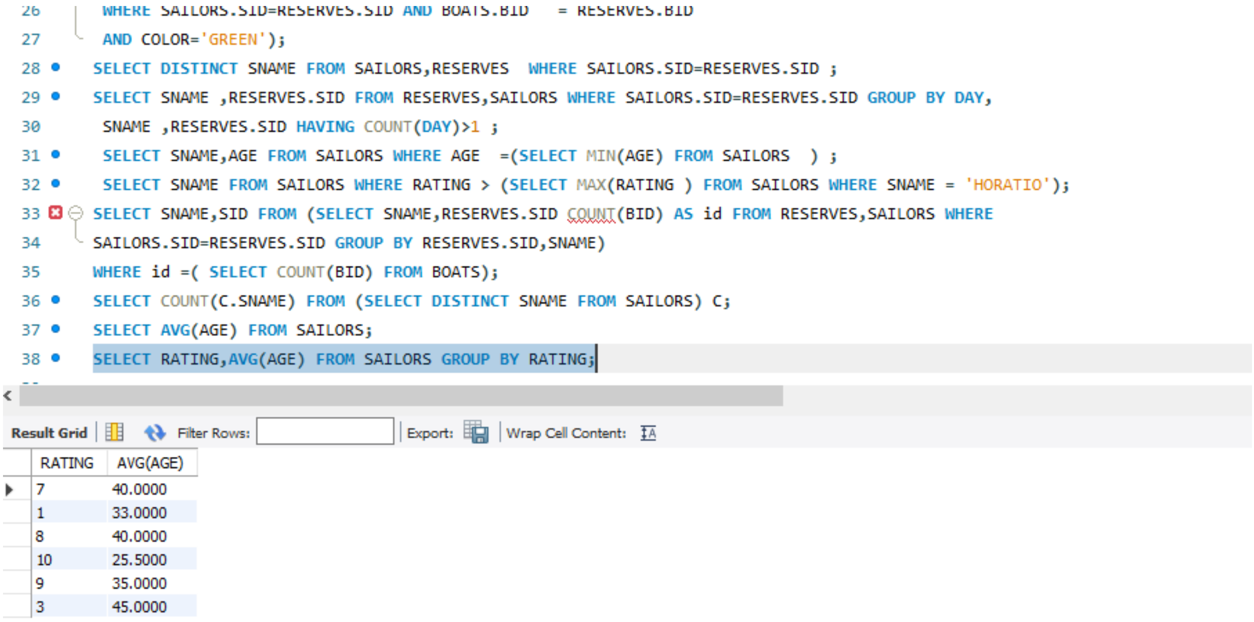


```sql
SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
SELECT SNAME ,RESERVES.SID FROM RESERVES,SAILORS WHERE SAILORS.SID=RESERVES.SID GROUP BY DAY,
 SNAME ,RESERVES.SID HAVING COUNT(DAY)>1 ;
 SELECT SNAME,AGE FROM SAILORS WHERE AGE  =(SELECT MIN(AGE) FROM SAILORS  ) ;
 SELECT SNAME FROM SAILORS WHERE RATING > (SELECT MAX(RATING ) FROM SAILORS WHERE SNAME = 'HORATIO
SELECT SNAME,SID FROM (SELECT SNAME,RESERVES.SID COUNT(BID) AS id FROM RESERVES,SAILORS WHERE
SAILORS.SID=RESERVES.SID GROUP BY RESERVES.SID,SNAME)
WHERE id =( SELECT COUNT(BID) FROM BOATS);
SELECT COUNT(C.SNAME) FROM (SELECT DISTINCT SNAME FROM SAILORS) C;
SELECT AVG(AGE) FROM SAILORS;
```

```
AVG(AGE)
36.9000
```

16.Find the average age of sailors for each rating level.

OUTPUT

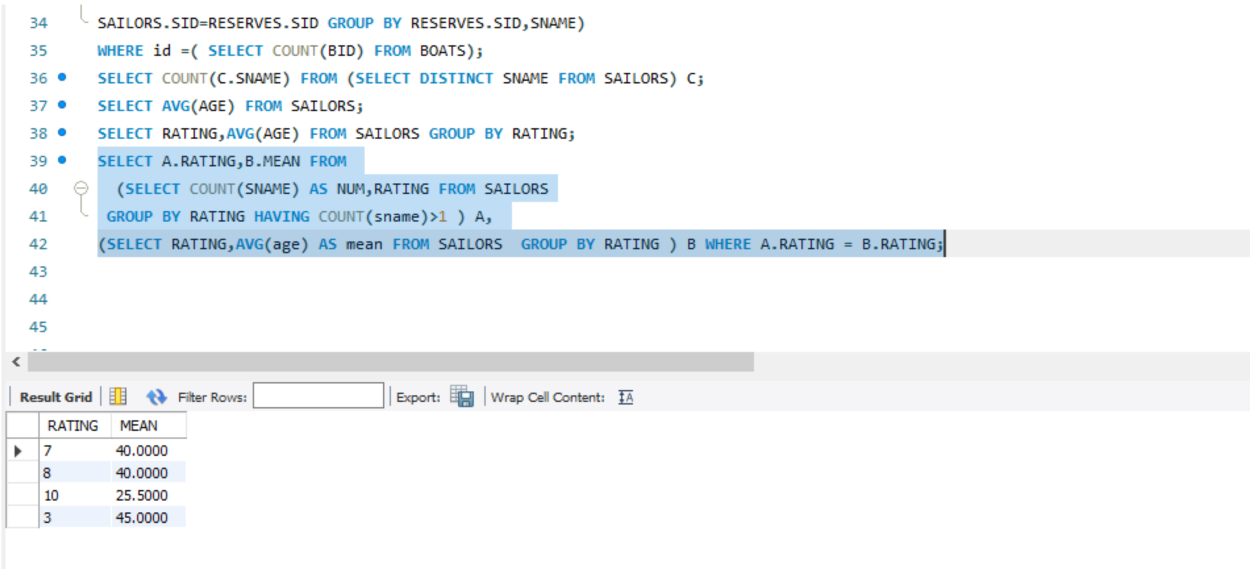


```
  WHERE SAILORS.SID=RESERVES.SID AND BOATS.BID   = RESERVES.BID
  AND COLOR='GREEN');
 SELECT DISTINCT SNAME FROM SAILORS,RESERVES  WHERE SAILORS.SID=RESERVES.SID ;
 SELECT SNAME ,RESERVES.SID FROM RESERVES,SAILORS WHERE SAILORS.SID=RESERVES.SID GROUP BY DAY,
  SNAME ,RESERVES.SID HAVING COUNT(DAY)>1 ;
  SELECT SNAME,AGE FROM SAILORS WHERE AGE  =(SELECT MIN(AGE) FROM SAILORS  ) ;
  SELECT SNAME FROM SAILORS WHERE RATING > (SELECT MAX(RATING ) FROM SAILORS WHERE SNAME = 'HORATIO');
 SELECT SNAME,SID FROM (SELECT SNAME,RESERVES.SID COUNT(BID) AS id FROM RESERVES,SAILORS WHERE
 SAILORS.SID=RESERVES.SID GROUP BY RESERVES.SID,SNAME)
 WHERE id =( SELECT COUNT(BID) FROM BOATS);
 SELECT COUNT(C.SNAME) FROM (SELECT DISTINCT SNAME FROM SAILORS) C;
 SELECT AVG(AGE) FROM SAILORS;
 SELECT RATING,AVG(AGE) FROM SAILORS GROUP BY RATING;
```

RATING	AVG(AGE)
7	40.0000
1	33.0000
8	40.0000
10	25.5000
9	35.0000
3	45.0000

17.Find the average age of sailors for each rating level that has at least two sailors.

OUTPUT



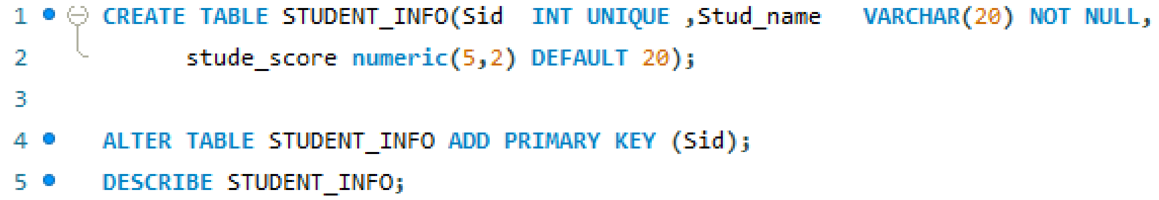
```
 SAILORS.SID=RESERVES.SID GROUP BY RESERVES.SID,SNAME)
 WHERE id =( SELECT COUNT(BID) FROM BOATS);
 SELECT COUNT(C.SNAME) FROM (SELECT DISTINCT SNAME FROM SAILORS) C;
 SELECT AVG(AGE) FROM SAILORS;
 SELECT RATING,AVG(AGE) FROM SAILORS GROUP BY RATING;
 SELECT A.RATING,B.MEAN FROM
   (SELECT COUNT(SNAME) AS NUM,RATING FROM SAILORS
  GROUP BY RATING HAVING COUNT(sname)>1 ) A,
 (SELECT RATING,AVG(age) AS mean FROM SAILORS  GROUP BY RATING ) B WHERE A.RATING = B.RATING;
```

RATING	MEAN
7	40.0000
8	40.0000
10	25.5000
3	45.0000

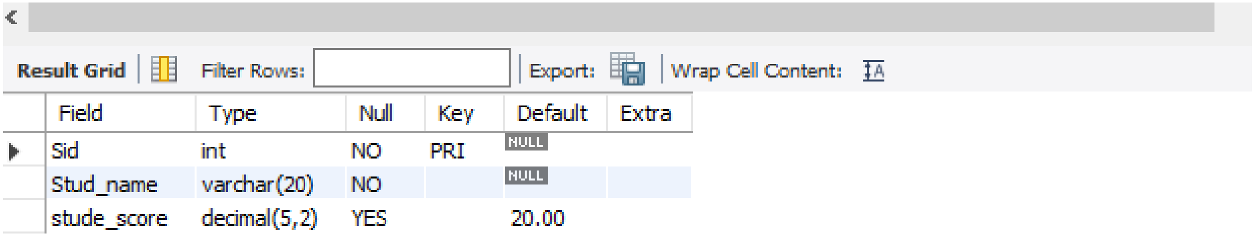
1. Create the table STUDENT_INFO with Columns: Sid, Stud_name & stude_score.

    - Insert values into STUDENT_INFO with the following constraints:Sid should be unique, Stud name NOT NULL and stude_score DEFAULT value of 20.

OUTPUT



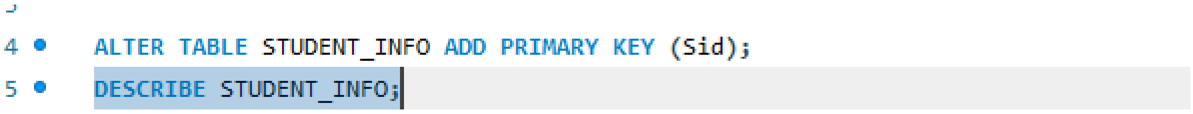
```
CREATE TABLE STUDENT_INFO(Sid  INT UNIQUE ,Stud_name   VARCHAR(20) NOT NULL,
        stude_score numeric(5,2) DEFAULT 20);

ALTER TABLE STUDENT_INFO ADD PRIMARY KEY (Sid);
DESCRIBE STUDENT_INFO;
```



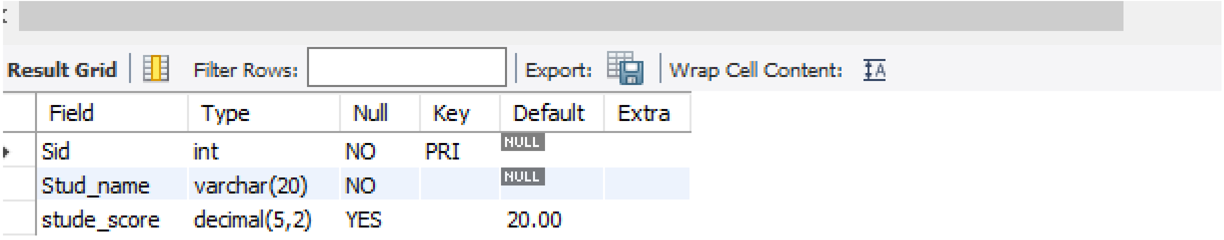
Field	Type	Null	Key	Default	Extra
Sid	int	NO	PRI	NULL	
Stud_name	varchar(20)	NO		NULL	
stude_score	decimal(5,2)	YES		20.00	

- Set Sid as primary key.

OUTPUT



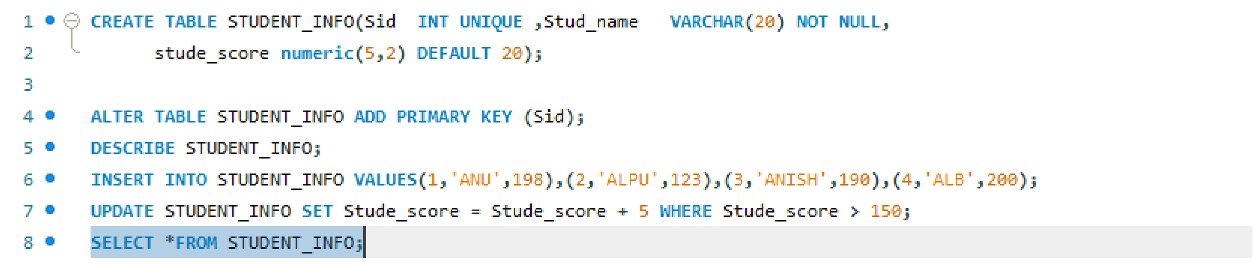
```
ALTER TABLE STUDENT_INFO ADD PRIMARY KEY (Sid);
DESCRIBE STUDENT_INFO;
```



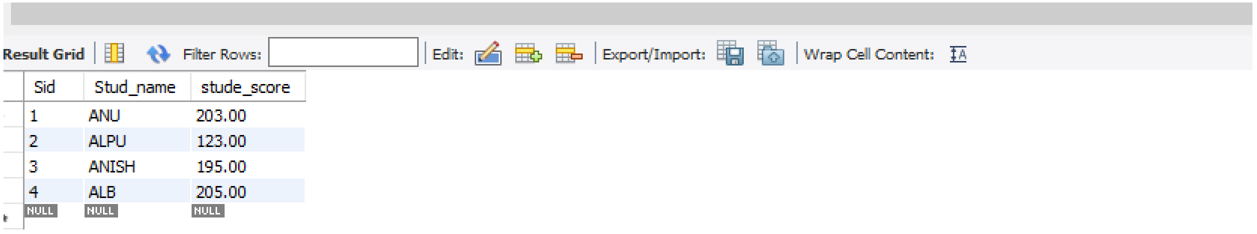
Field	Type	Null	Key	Default	Extra
Sid	int	NO	PRI	NULL	
Stud_name	varchar(20)	NO		NULL	
stude_score	decimal(5,2)	YES		20.00	

- Update stude_score by adding a value of 5 to stude_score in the table STUDENT_INFO for the rows satisfying the condition of stude_score >150 (Using CASE)

OUTPUT



```
CREATE TABLE STUDENT_INFO(Sid  INT UNIQUE ,Stud_name   VARCHAR(20) NOT NULL,
      stude_score numeric(5,2) DEFAULT 20);

ALTER TABLE STUDENT_INFO ADD PRIMARY KEY (Sid);
DESCRIBE STUDENT_INFO;
INSERT INTO STUDENT_INFO VALUES(1,'ANU',198),(2,'ALPU',123),(3,'ANISH',190),(4,'ALB',200);
UPDATE STUDENT_INFO SET Stude_score = Stude_score + 5 WHERE Stude_score > 150;
SELECT *FROM STUDENT_INFO;
```

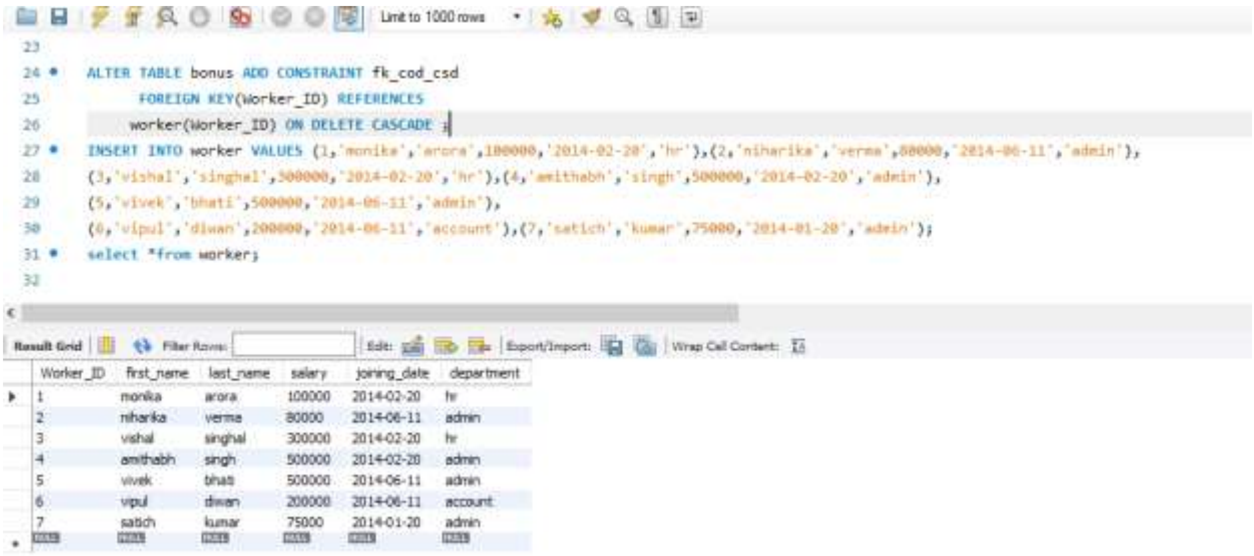


Sid	Stud_name	stude_score
1	ANU	203.00
2	ALPU	123.00
3	ANISH	195.00
4	ALB	205.00
NULL	NULL	NULL

2. Create the tables **worker** and **bonus** with the following fields. The primary key of Worker table is Worker_ID. Set Worker_id as foreign key of bonus on update and delete cascade constraints. Each constraint should be given a name

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	2014-02-20	HR
2	Niharika	Verma	80000	2014-06-11	Admin
3	Vishal	Singhal	300000	2014-02-20	HR
4	Amitabh	Singh	500000	2014-02-20	Admin
5	Vivek	Bhati	500000	2014-06-11	Admin
6	Vipul	Diwan	200000	2014-06-11	Account
7	Satish	Kumar	75000	2014-01-20	Account
8	Geetika	Chauhan	90000	2014-04-11	Admin

Output



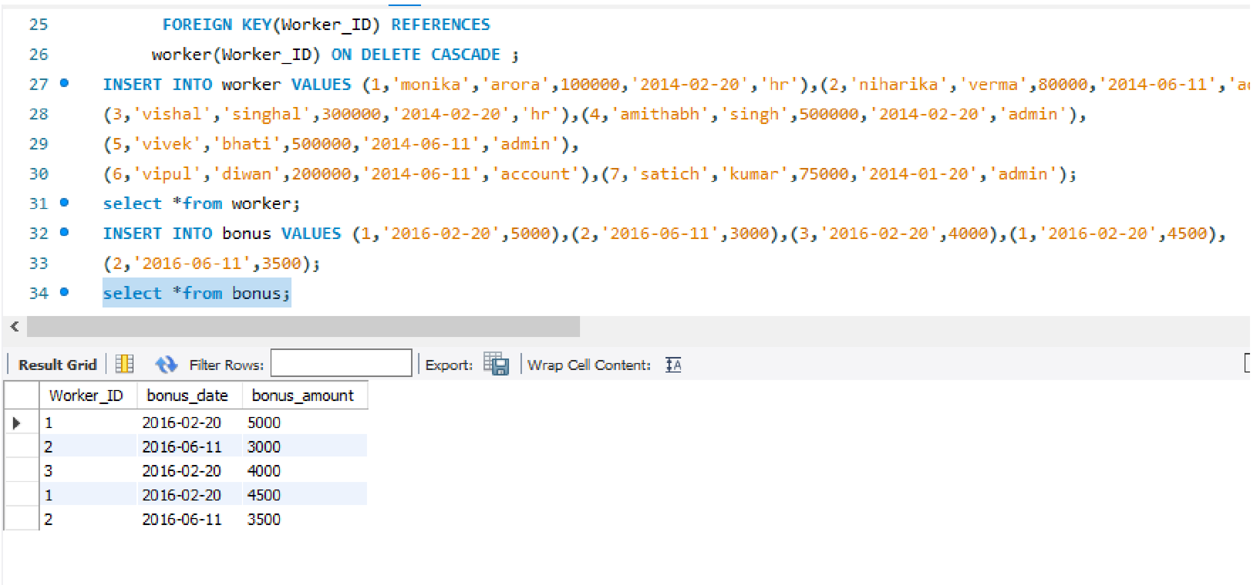
```
ALTER TABLE bonus ADD CONSTRAINT fk_cod_csd
      FOREIGN KEY(Worker_ID) REFERENCES
     worker(Worker_ID) ON DELETE CASCADE ;
INSERT INTO worker VALUES (1,'monika','arora',100000,'2014-02-20','hr'),(2,'niharika','verma',80000,'2014-06-11','admin'),
(3,'vishal','singhal',300000,'2014-02-20','hr'),(4,'amithabh','singh',500000,'2014-02-20','admin'),
(5,'vivek','bhati',500000,'2014-06-11','admin'),
(6,'vipul','diwan',200000,'2014-06-11','account'),(7,'satich','kumar',75000,'2014-01-20','admin');
select *from worker;
```

Worker_ID	first_name	last_name	salary	joining_date	department
1	monika	arora	100000	2014-02-20	hr
2	niharika	verma	80000	2014-06-11	admin
3	vishal	singhal	300000	2014-02-20	hr
4	amithabh	singh	500000	2014-02-20	admin
5	vivek	bhati	500000	2014-06-11	admin
6	vipul	diwan	200000	2014-06-11	account
7	satich	kumar	75000	2014-01-20	admin
NULL	NULL	NULL	NULL	NULL	NULL

3. Sample Table – Bonus

WORKER_ID	BONUS_DATE	BONUS_AMOUNT
1	2016-02-20	5000
2	2016-06-11	3000
3	2016-02-20	4000
1	2016-02-20	4500
2	2016-06-11	3500

Output

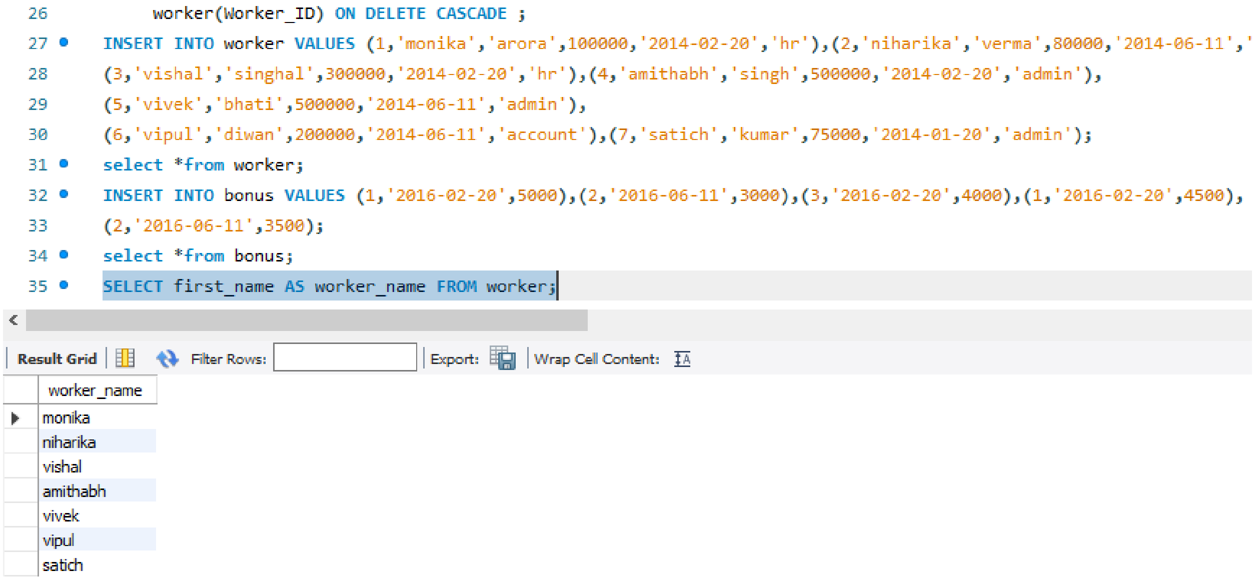


```
         FOREIGN KEY(Worker_ID) REFERENCES
        worker(Worker_ID) ON DELETE CASCADE ;
INSERT INTO worker VALUES (1,'monika','arora',100000,'2014-02-20','hr'),(2,'niharika','verma',80000,'2014-06-11','a
(3,'vishal','singhal',300000,'2014-02-20','hr'),(4,'amithabh','singh',500000,'2014-02-20','admin'),
(5,'vivek','bhati',500000,'2014-06-11','admin'),
(6,'vipul','diwan',200000,'2014-06-11','account'),(7,'satich','kumar',75000,'2014-01-20','admin');
select *from worker;
INSERT INTO bonus VALUES (1,'2016-02-20',5000),(2,'2016-06-11',3000),(3,'2016-02-20',4000),(1,'2016-02-20',4500),
(2,'2016-06-11',3500);
select *from bonus;
```

Worker_ID	bonus_date	bonus_amount
1	2016-02-20	5000
2	2016-06-11	3000
3	2016-02-20	4000
1	2016-02-20	4500
2	2016-06-11	3500

4. Write An SQL Query To Fetch "FIRST_NAME" From Worker Table Using The Alias Name As <WORKER_NAME>.

Output

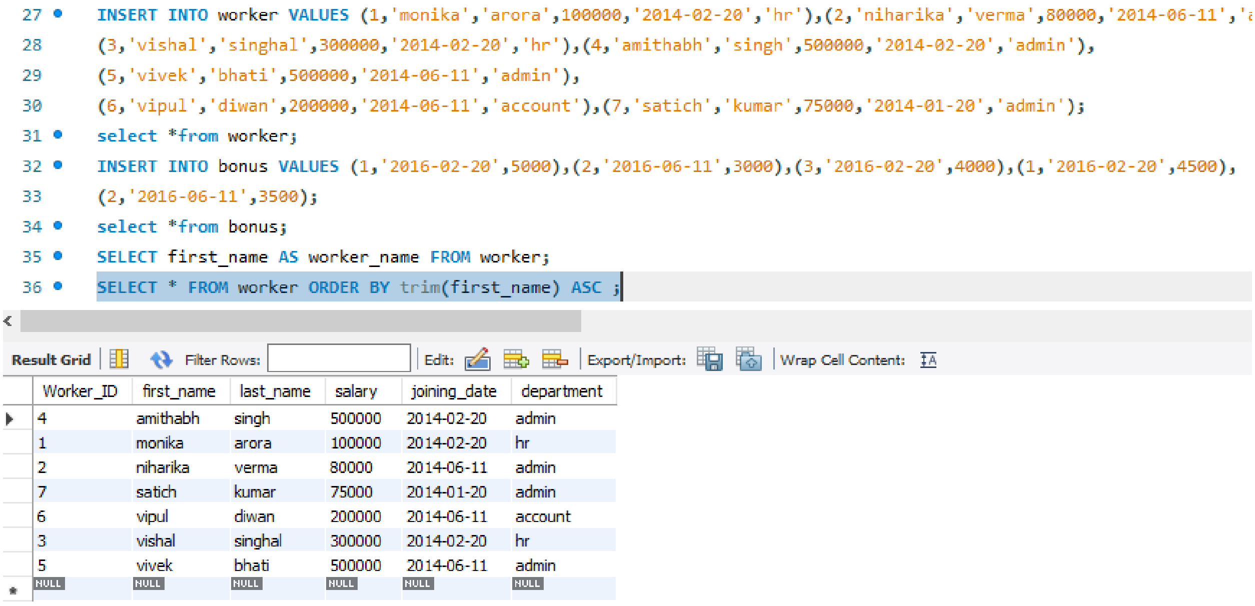


```
        worker(Worker_ID) ON DELETE CASCADE ;
INSERT INTO worker VALUES (1,'monika','arora',100000,'2014-02-20','hr'),(2,'niharika','verma',80000,'2014-06-11','
(3,'vishal','singhal',300000,'2014-02-20','hr'),(4,'amithabh','singh',500000,'2014-02-20','admin'),
(5,'vivek','bhati',500000,'2014-06-11','admin'),
(6,'vipul','diwan',200000,'2014-06-11','account'),(7,'satich','kumar',75000,'2014-01-20','admin');
select *from worker;
INSERT INTO bonus VALUES (1,'2016-02-20',5000),(2,'2016-06-11',3000),(3,'2016-02-20',4000),(1,'2016-02-20',4500),
(2,'2016-06-11',3500);
select *from bonus;
SELECT first_name AS worker_name FROM worker;
```

worker_name
monika
niharika
vishal
amithabh
vivek
vipul
satich

5. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending

Output

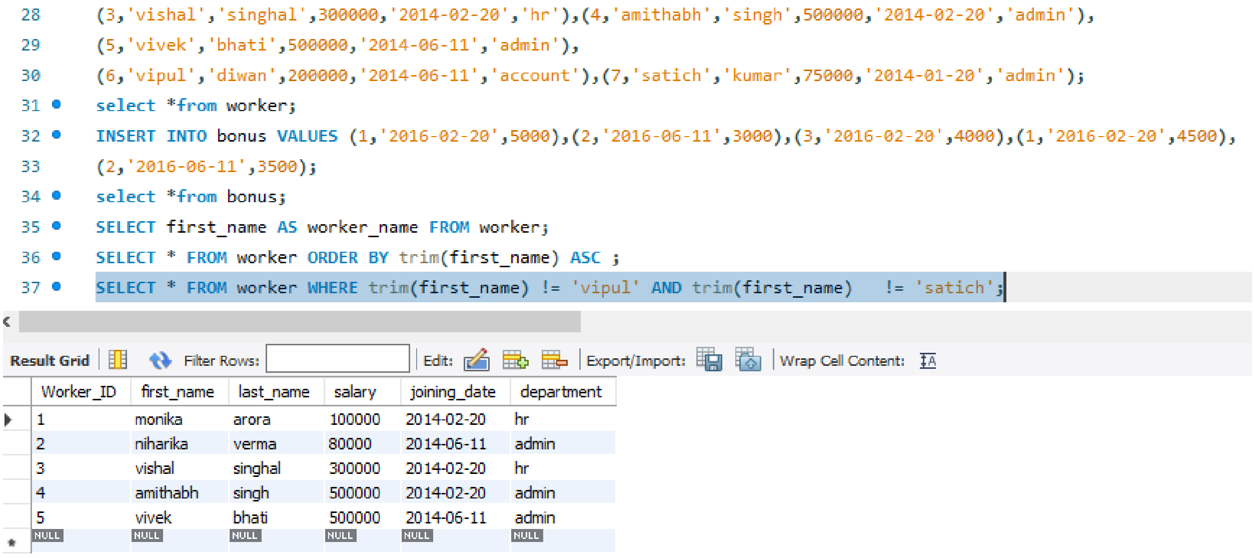


```sql
INSERT INTO worker VALUES (1,'monika','arora',100000,'2014-02-20','hr'),(2,'niharika','verma',80000,'2014-06-11','a
(3,'vishal','singhal',300000,'2014-02-20','hr'),(4,'amithabh','singh',500000,'2014-02-20','admin'),
(5,'vivek','bhati',500000,'2014-06-11','admin'),
(6,'vipul','diwan',200000,'2014-06-11','account'),(7,'satich','kumar',75000,'2014-01-20','admin');
select *from worker;
INSERT INTO bonus VALUES (1,'2016-02-20',5000),(2,'2016-06-11',3000),(3,'2016-02-20',4000),(1,'2016-02-20',4500),
(2,'2016-06-11',3500);
select *from bonus;
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
```

Worker_ID	first_name	last_name	salary	joining_date	department
4	amithabh	singh	500000	2014-02-20	admin
1	monika	arora	100000	2014-02-20	hr
2	niharika	verma	80000	2014-06-11	admin
7	satich	kumar	75000	2014-01-20	admin
6	vipul	diwan	200000	2014-06-11	account
3	vishal	singhal	300000	2014-02-20	hr
5	vivek	bhati	500000	2014-06-11	admin
NULL	NULL	NULL	NULL	NULL	NULL

6. Write An SQL Query To Print Details Of Workers Excluding First Names, “Vipul” And “Satish” From Worker Table.

Output

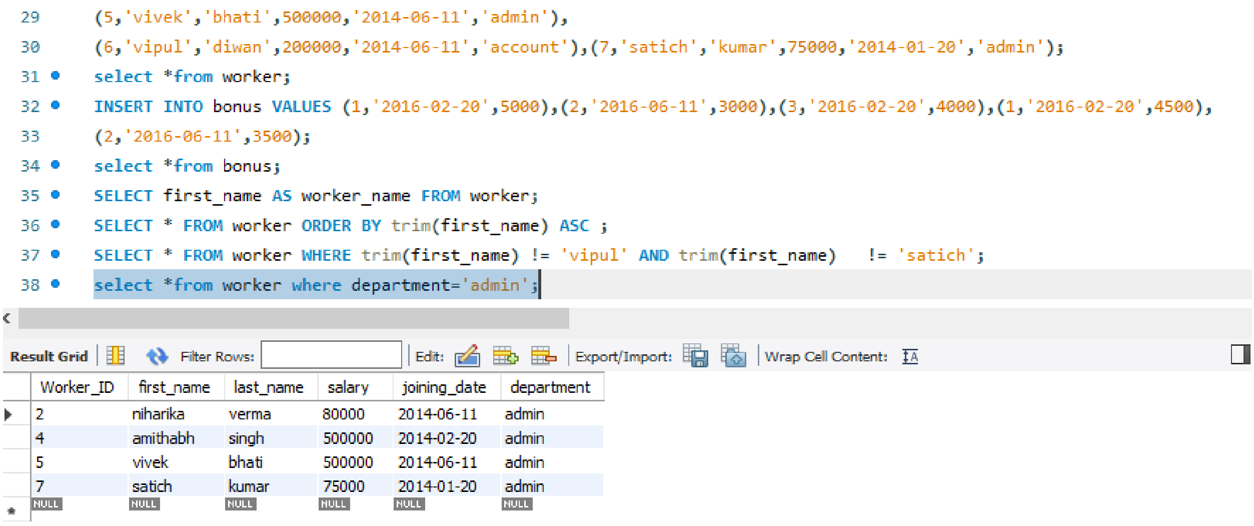


```sql
(3,'vishal','singhal',300000,'2014-02-20','hr'),(4,'amithabh','singh',500000,'2014-02-20','admin'),
(5,'vivek','bhati',500000,'2014-06-11','admin'),
(6,'vipul','diwan',200000,'2014-06-11','account'),(7,'satich','kumar',75000,'2014-01-20','admin');
select *from worker;
INSERT INTO bonus VALUES (1,'2016-02-20',5000),(2,'2016-06-11',3000),(3,'2016-02-20',4000),(1,'2016-02-20',4500),
(2,'2016-06-11',3500);
select *from bonus;
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
```

Worker_ID	first_name	last_name	salary	joining_date	department
1	monika	arora	100000	2014-02-20	hr
2	niharika	verma	80000	2014-06-11	admin
3	vishal	singhal	300000	2014-02-20	hr
4	amithabh	singh	500000	2014-02-20	admin
5	vivek	bhati	500000	2014-06-11	admin
NULL	NULL	NULL	NULL	NULL	NULL

7. Write An SQL Query To Print Details Of Workers With DEPARTMENT Name As “Admin”.

Output

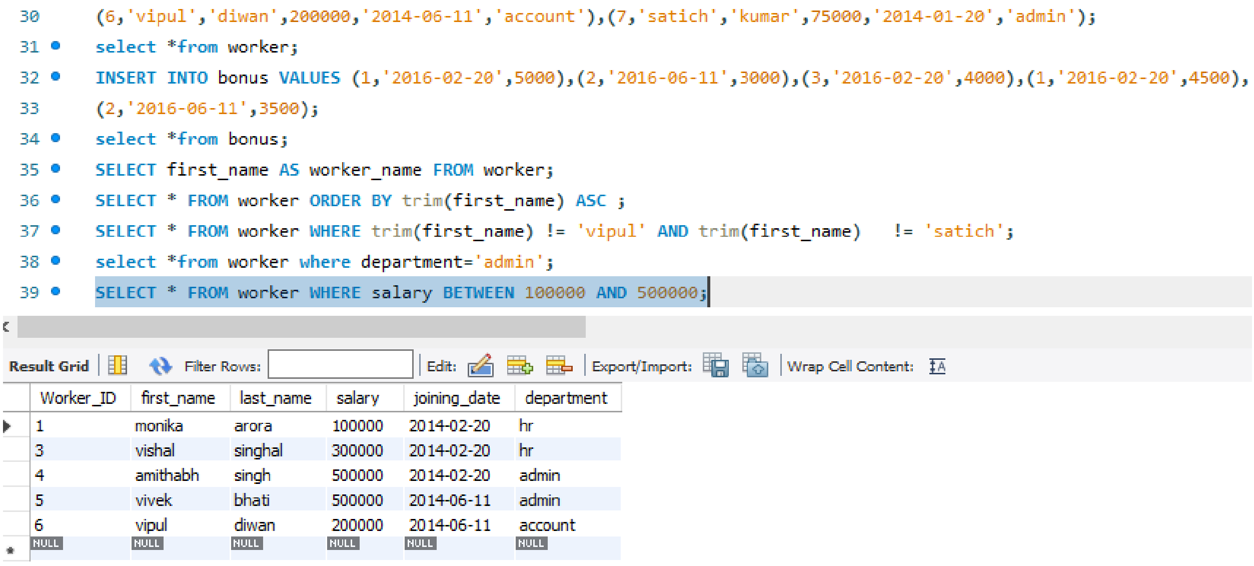


```sql
(5,'vivek','bhati',500000,'2014-06-11','admin'),
(6,'vipul','diwan',200000,'2014-06-11','account'),(7,'satich','kumar',75000,'2014-01-20','admin');
select *from worker;
INSERT INTO bonus VALUES (1,'2016-02-20',5000),(2,'2016-06-11',3000),(3,'2016-02-20',4000),(1,'2016-02-20',4500),
(2,'2016-06-11',3500);
select *from bonus;
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
```

Worker_ID	first_name	last_name	salary	joining_date	department
2	niharika	verma	80000	2014-06-11	admin
4	amithabh	singh	500000	2014-02-20	admin
5	vivek	bhati	500000	2014-06-11	admin
7	satich	kumar	75000	2014-01-20	admin
NULL	NULL	NULL	NULL	NULL	NULL

8. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000

Output

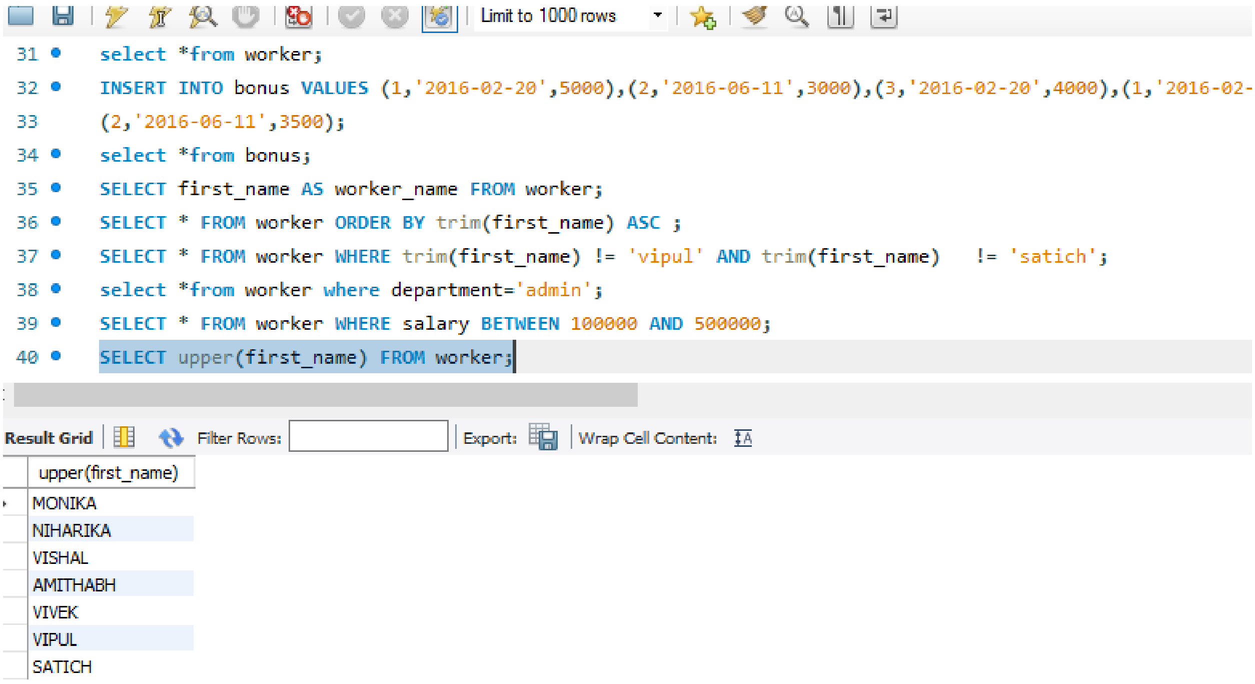


```sql
(6,'vipul','diwan',200000,'2014-06-11','account'),(7,'satich','kumar',75000,'2014-01-20','admin');
select *from worker;
INSERT INTO bonus VALUES (1,'2016-02-20',5000),(2,'2016-06-11',3000),(3,'2016-02-20',4000),(1,'2016-02-20',4500),
(2,'2016-06-11',3500);
select *from bonus;
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
```

Worker_ID	first_name	last_name	salary	joining_date	department
1	monika	arora	100000	2014-02-20	hr
3	vishal	singhal	300000	2014-02-20	hr
4	amithabh	singh	500000	2014-02-20	admin
5	vivek	bhati	500000	2014-06-11	admin
6	vipul	diwan	200000	2014-06-11	account
NULL	NULL	NULL	NULL	NULL	NULL

9. Write An SQL Query To Fetch “FIRST_NAME” From Worker Table In Upper Case. (upper())

Output

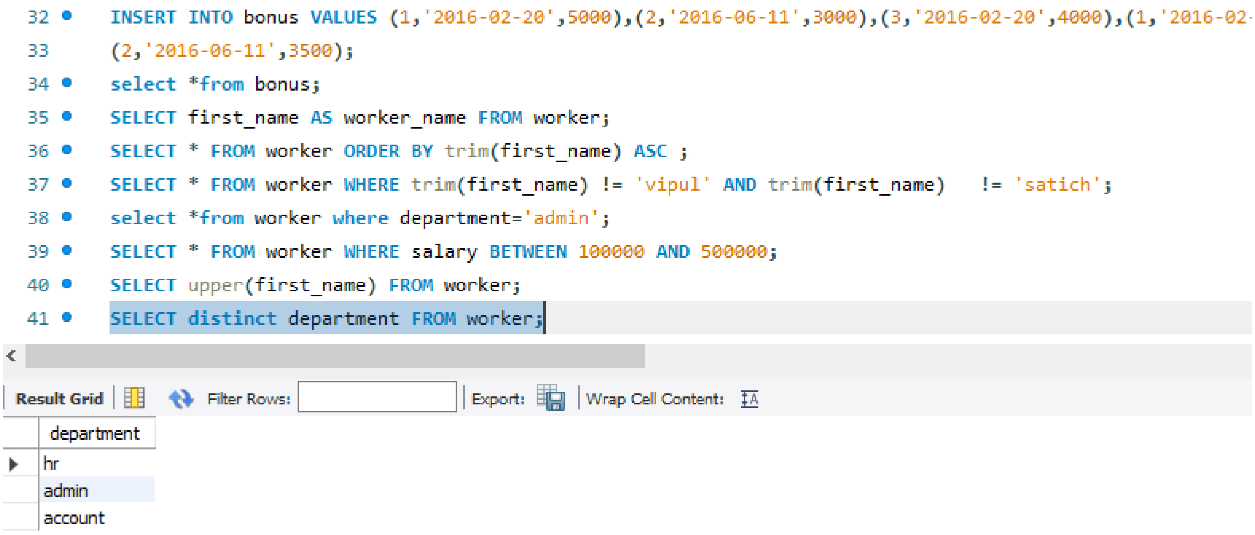


```
select *from worker;
INSERT INTO bonus VALUES (1,'2016-02-20',5000),(2,'2016-06-11',3000),(3,'2016-02-20',4000),(1,'2016-02-
(2,'2016-06-11',3500);
select *from bonus;
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
SELECT upper(first_name) FROM worker;
```

upper(first_name)
MONIKA
NIHARIKA
VISHAL
AMITHABH
VIVEK
VIPUL
SATICH

10. Write An SQL Query To Fetch Unique Values Of DEPARTMENT From Worker Table.

Output

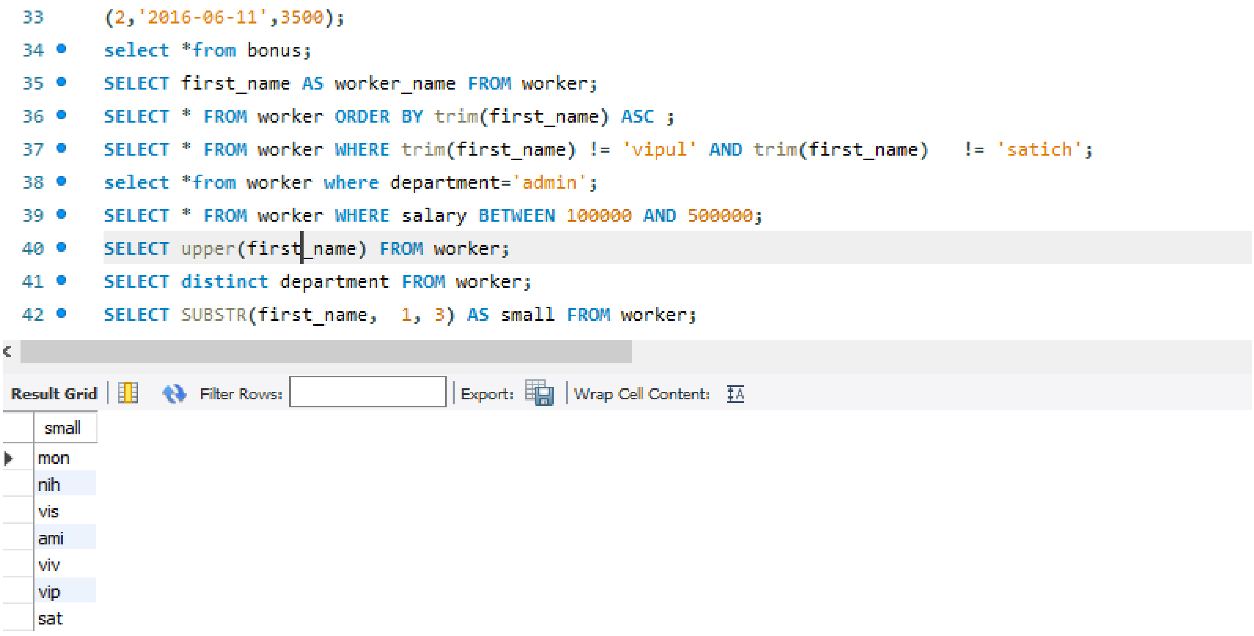


```
INSERT INTO bonus VALUES (1,'2016-02-20',5000),(2,'2016-06-11',3000),(3,'2016-02-20',4000),(1,'2016-02-
(2,'2016-06-11',3500);
select *from bonus;
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
SELECT upper(first_name) FROM worker;
SELECT distinct department FROM worker;
```

department
hr
admin
account

11. Write An SQL Query To Print First Three Characters Of FIRST_NAME From Worker Table.(substring())

Output

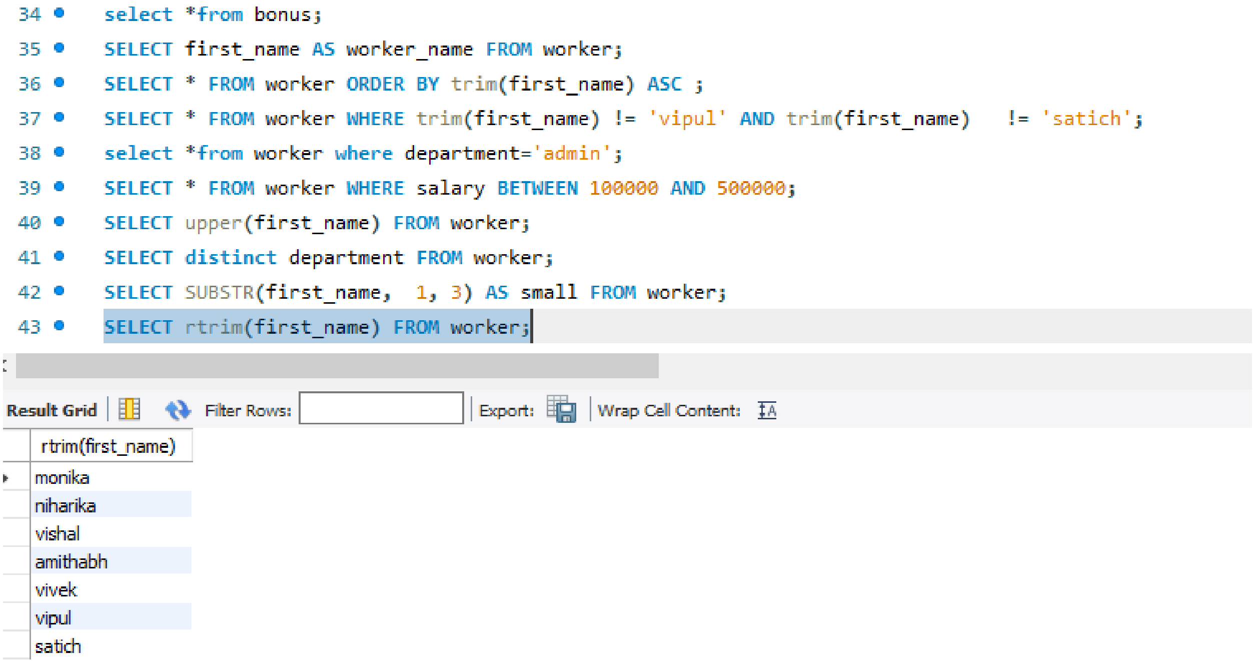


```
(2,'2016-06-11',3500);
select *from bonus;
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
SELECT upper(first_name) FROM worker;
SELECT distinct department FROM worker;
SELECT SUBSTR(first_name,  1, 3) AS small FROM worker;
```

small
mon
nih
vis
ami
viv
vip
sat

12. Write An SQL Query To Print The FIRST_NAME From Worker Table After Removing White Spaces From The Right Side(RTRIM ( ))

Output

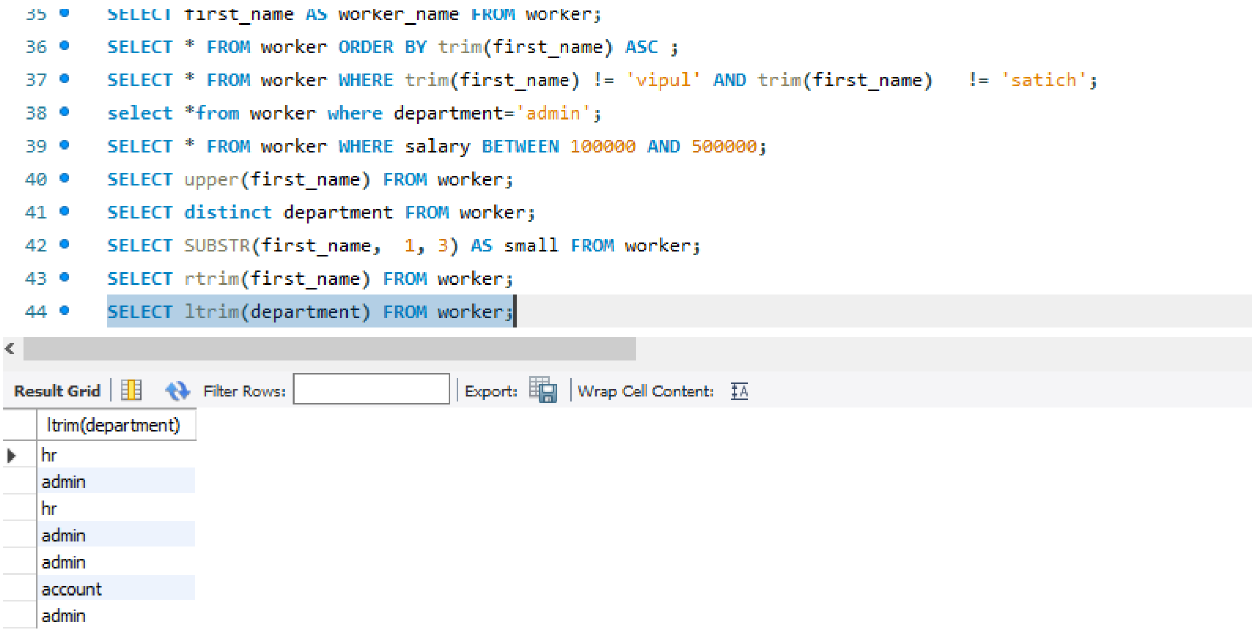


```
select *from bonus;
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
SELECT upper(first_name) FROM worker;
SELECT distinct department FROM worker;
SELECT SUBSTR(first_name,  1, 3) AS small FROM worker;
SELECT rtrim(first_name) FROM worker;
```

rtrim(first_name)
monika
niharika
vishal
amithabh
vivek
vipul
satich

13. Write An SQL Query To Print The DEPARTMENT From Worker Table After Removing White Spaces From The Left Side. (LTRIM ( ))

Output

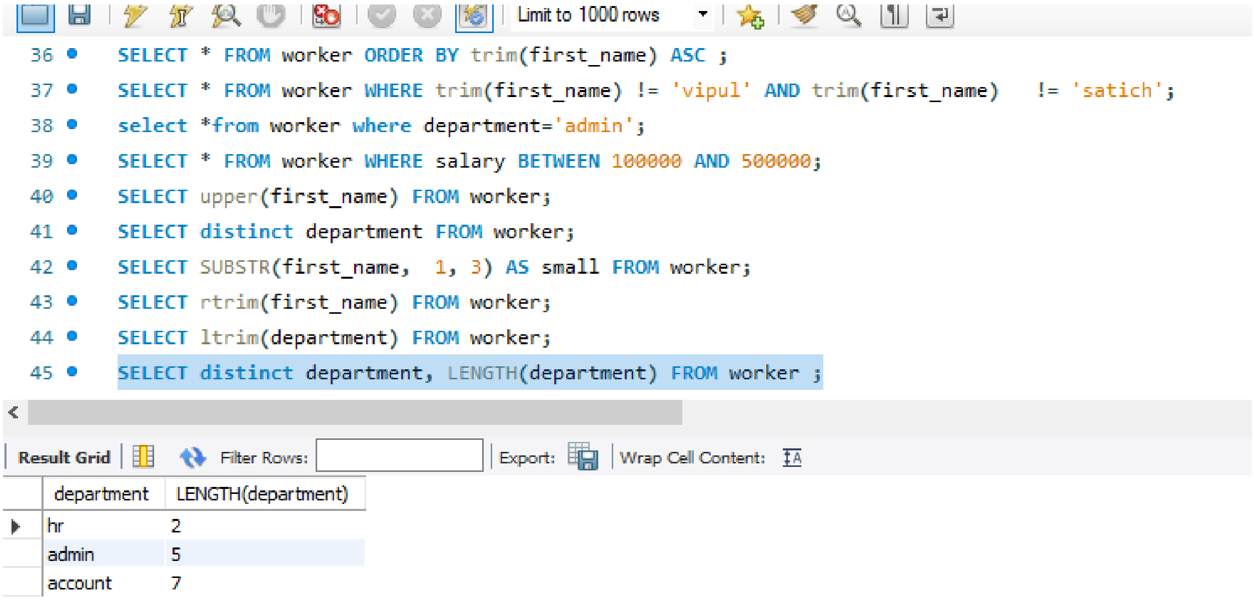


```
SELECT first_name AS worker_name FROM worker;
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
SELECT upper(first_name) FROM worker;
SELECT distinct department FROM worker;
SELECT SUBSTR(first_name,  1, 3) AS small FROM worker;
SELECT rtrim(first_name) FROM worker;
SELECT ltrim(department) FROM worker;
```

ltrim(department)
hr
admin
hr
admin
admin
account
admin

14. Write An SQL Query That Fetches The Unique Values Of DEPARTMENT From Worker Table And Prints Its Length.(length())

Output

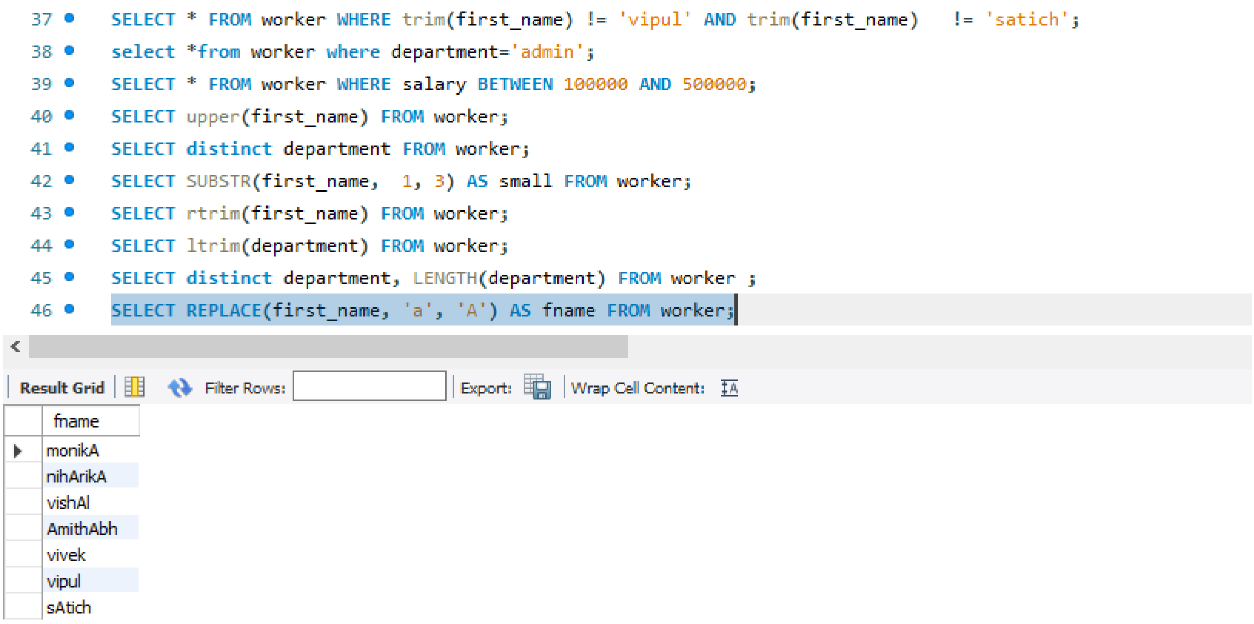


```
SELECT * FROM worker ORDER BY trim(first_name) ASC ;
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
SELECT upper(first_name) FROM worker;
SELECT distinct department FROM worker;
SELECT SUBSTR(first_name,  1, 3) AS small FROM worker;
SELECT rtrim(first_name) FROM worker;
SELECT ltrim(department) FROM worker;
SELECT distinct department, LENGTH(department) FROM worker ;
```

department	LENGTH(department)
hr	2
admin	5
account	7

15. Write An SQL Query To Print The FIRST_NAME From Worker Table After Replacing ‘a’ With ‘A’.(REPLACE( ))

Output

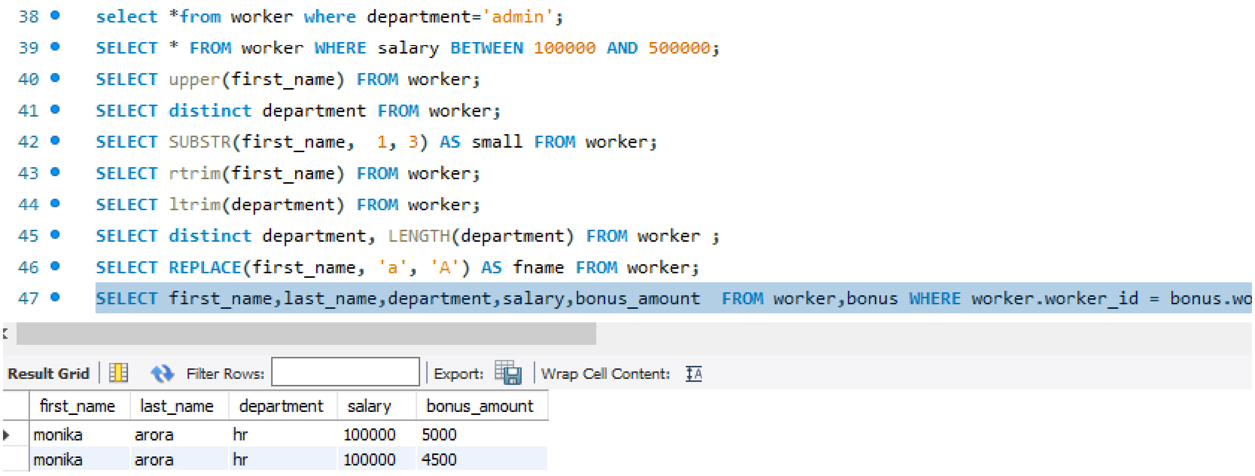


```
SELECT * FROM worker WHERE trim(first_name) != 'vipul' AND trim(first_name)   != 'satich';
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
SELECT upper(first_name) FROM worker;
SELECT distinct department FROM worker;
SELECT SUBSTR(first_name,  1, 3) AS small FROM worker;
SELECT rtrim(first_name) FROM worker;
SELECT ltrim(department) FROM worker;
SELECT distinct department, LENGTH(department) FROM worker ;
SELECT REPLACE(first_name, 'a', 'A') AS fname FROM worker;
```

fname
monikA
nihArikA
vishAl
AmithAbh
vivek
vipul
sAtich

16. Find the First name , last name ,Department, Salary and Bonus of employees whose bonus amount is greater than 4000

Output

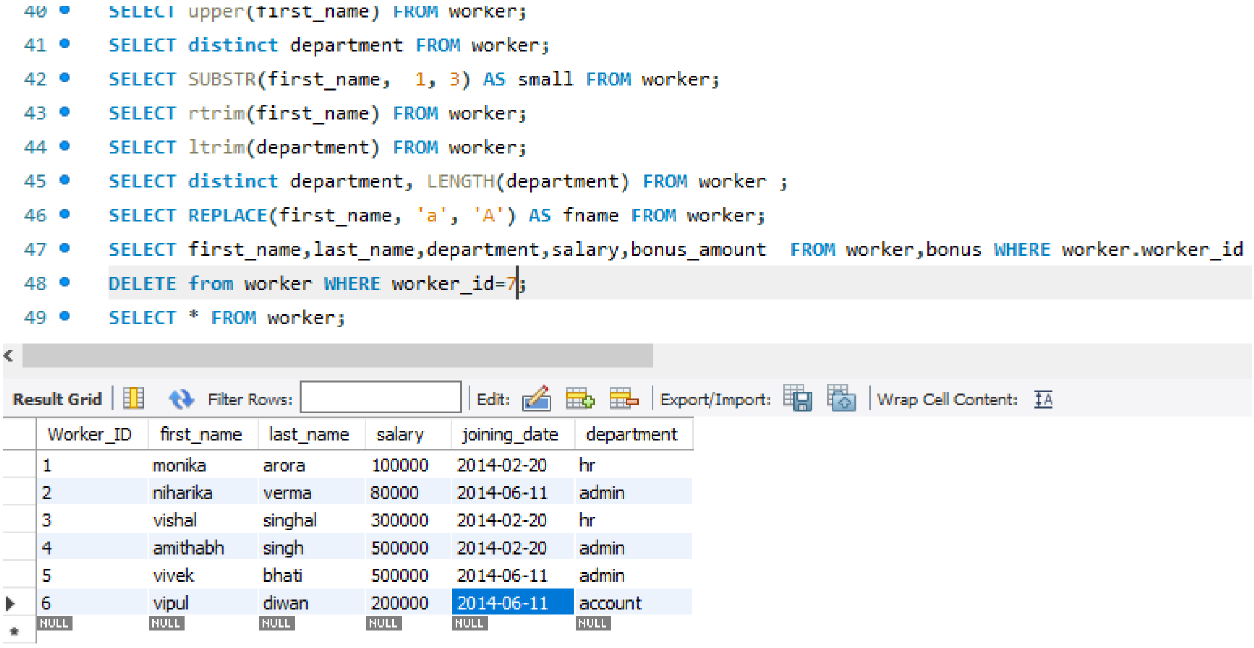


```
select *from worker where department='admin';
SELECT * FROM worker WHERE salary BETWEEN 100000 AND 500000;
SELECT upper(first_name) FROM worker;
SELECT distinct department FROM worker;
SELECT SUBSTR(first_name,  1, 3) AS small FROM worker;
SELECT rtrim(first_name) FROM worker;
SELECT ltrim(department) FROM worker;
SELECT distinct department, LENGTH(department) FROM worker ;
SELECT REPLACE(first_name, 'a', 'A') AS fname FROM worker;
SELECT first_name,last_name,department,salary,bonus_amount  FROM worker,bonus WHERE worker.worker_id = bonus.wo
```

first_name	last_name	department	salary	bonus_amount
monika	arora	hr	100000	5000
monika	arora	hr	100000	4500

17. Delete the employee with worker_id=7 from worker and display the details of both tables.

Output

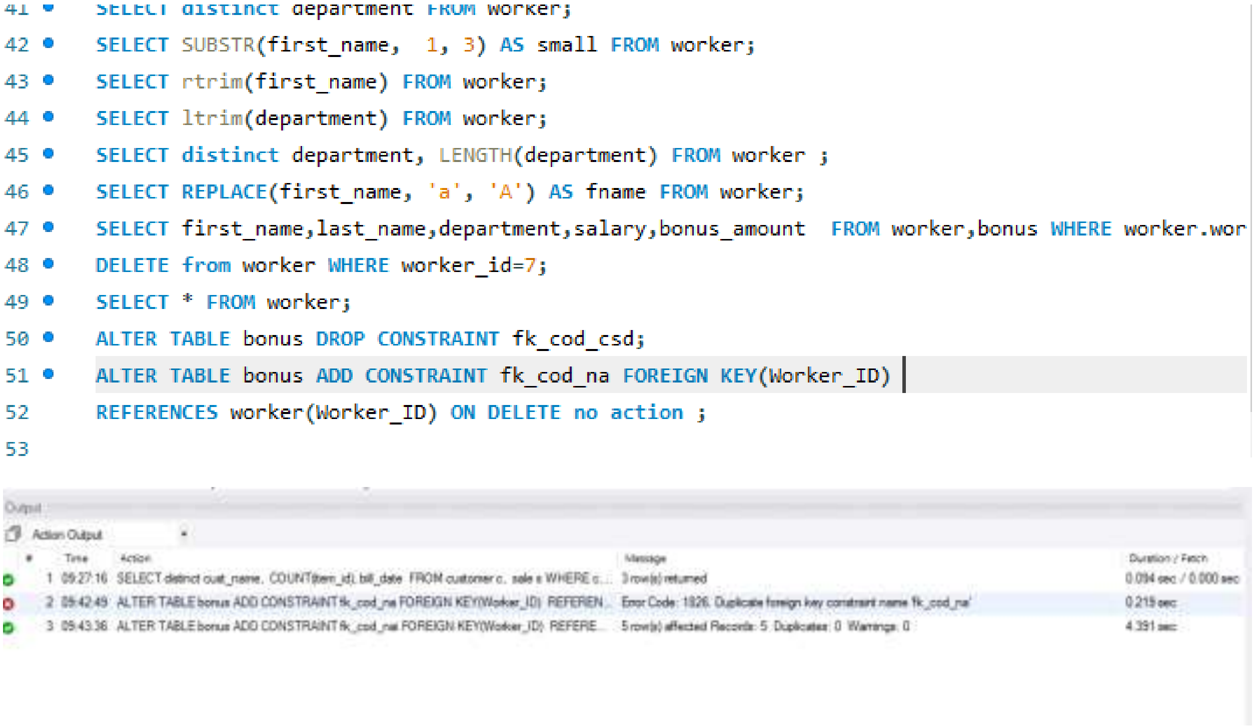


```
SELECT upper(first_name) FROM worker;
SELECT distinct department FROM worker;
SELECT SUBSTR(first_name,  1, 3) AS small FROM worker;
SELECT rtrim(first_name) FROM worker;
SELECT ltrim(department) FROM worker;
SELECT distinct department, LENGTH(department) FROM worker ;
SELECT REPLACE(first_name, 'a', 'A') AS fname FROM worker;
SELECT first_name,last_name,department,salary,bonus_amount  FROM worker,bonus WHERE worker.worker_id
DELETE from worker WHERE worker_id=7;
SELECT * FROM worker;
```

Worker_ID	first_name	last_name	salary	joining_date	department
1	monika	arora	100000	2014-02-20	hr
2	niharika	verma	80000	2014-06-11	admin
3	vishal	singhal	300000	2014-02-20	hr
4	amithabh	singh	500000	2014-02-20	admin
5	vivek	bhati	500000	2014-06-11	admin
6	vipul	diwan	200000	2014-06-11	account
NULL	NULL	NULL	NULL	NULL	NULL

18. Drop the foreign key constraint and add a new referential integrity constraint with ‘on update or delete with no action’

Output

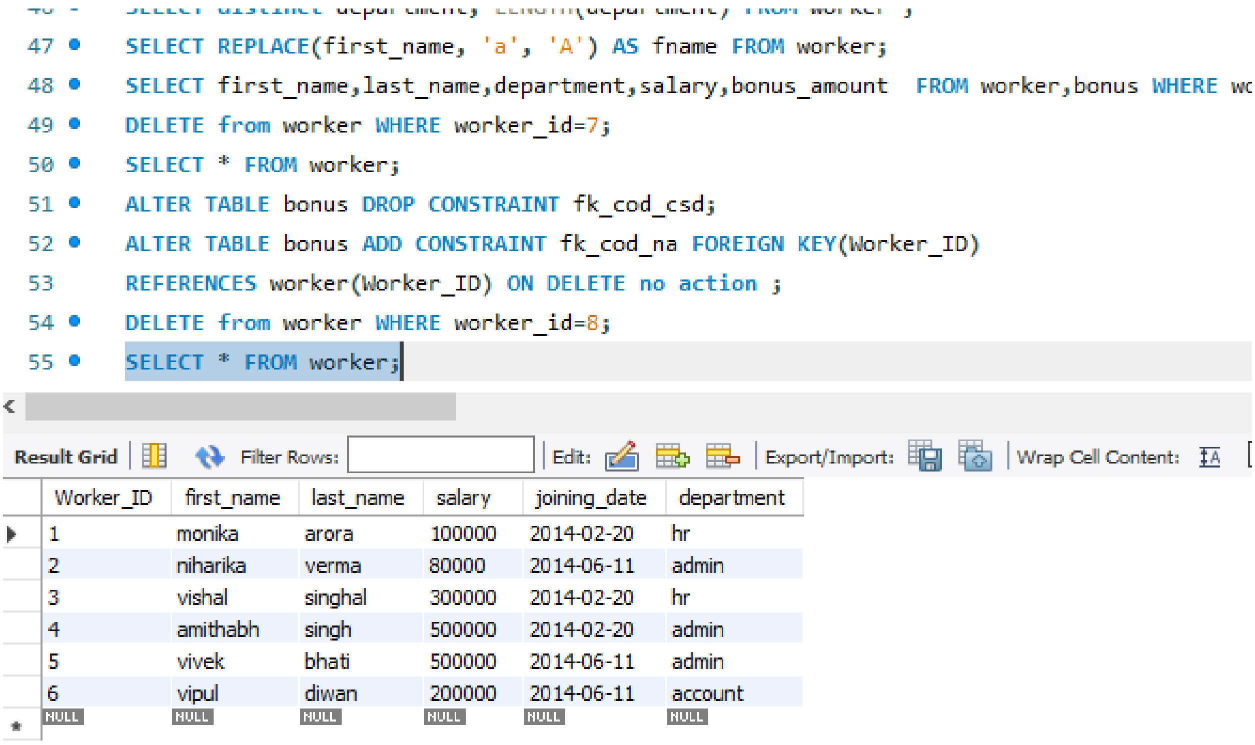


```
SELECT distinct department FROM worker;
SELECT SUBSTR(first_name,  1, 3) AS small FROM worker;
SELECT rtrim(first_name) FROM worker;
SELECT ltrim(department) FROM worker;
SELECT distinct department, LENGTH(department) FROM worker ;
SELECT REPLACE(first_name, 'a', 'A') AS fname FROM worker;
SELECT first_name,last_name,department,salary,bonus_amount  FROM worker,bonus WHERE worker.wor
DELETE from worker WHERE worker_id=7;
SELECT * FROM worker;
ALTER TABLE bonus DROP CONSTRAINT fk_cod_csd;
ALTER TABLE bonus ADD CONSTRAINT fk_cod_na FOREIGN KEY(Worker_ID)
REFERENCES worker(Worker_ID) ON DELETE no action ;
```

```
#  Time      Action                                                                     Message                                                          Duration / Fetch
1  09:27:16  SELECT distinct cust_name, COUNT(item_id), bill_date  FROM customer c, sale s WHERE c...  3 row(s) returned                                   0.094 sec / 0.000 sec
2  09:42:49  ALTER TABLE bonus ADD CONSTRAINT fk_cod_na FOREIGN KEY(Worker_ID)  REFEREN...  Error Code: 1826. Duplicate foreign key constraint name 'fk_cod_na'  0.219 sec
3  09:43:36  ALTER TABLE bonus ADD CONSTRAINT fk_cod_nai FOREIGN KEY(Worker_ID)  REFERE...  5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0   4.391 sec
```

19. Delete the employee with worker_id = 8 from worker.

Output



```
SELECT distinct department, LENGTH(department) FROM worker ;
SELECT REPLACE(first_name, 'a', 'A') AS fname FROM worker;
SELECT first_name,last_name,department,salary,bonus_amount  FROM worker,bonus WHERE wc
DELETE from worker WHERE worker_id=7;
SELECT * FROM worker;
ALTER TABLE bonus DROP CONSTRAINT fk_cod_csd;
ALTER TABLE bonus ADD CONSTRAINT fk_cod_na FOREIGN KEY(Worker_ID)
REFERENCES worker(Worker_ID) ON DELETE no action ;
DELETE from worker WHERE worker_id=8;
SELECT * FROM worker;
```

Worker_ID	first_name	last_name	salary	joining_date	department
1	monika	arora	100000	2014-02-20	hr
2	niharika	verma	80000	2014-06-11	admin
3	vishal	singhal	300000	2014-02-20	hr
4	amithabh	singh	500000	2014-02-20	admin
5	vivek	bhati	500000	2014-06-11	admin
6	vipul	diwan	200000	2014-06-11	account
NULL	NULL	NULL	NULL	NULL	NULL

Q.SET 3 **date :11-06-2021**

Create the tables given below and execute the queries:

Customer(Cust id : integer, cust_name: string)

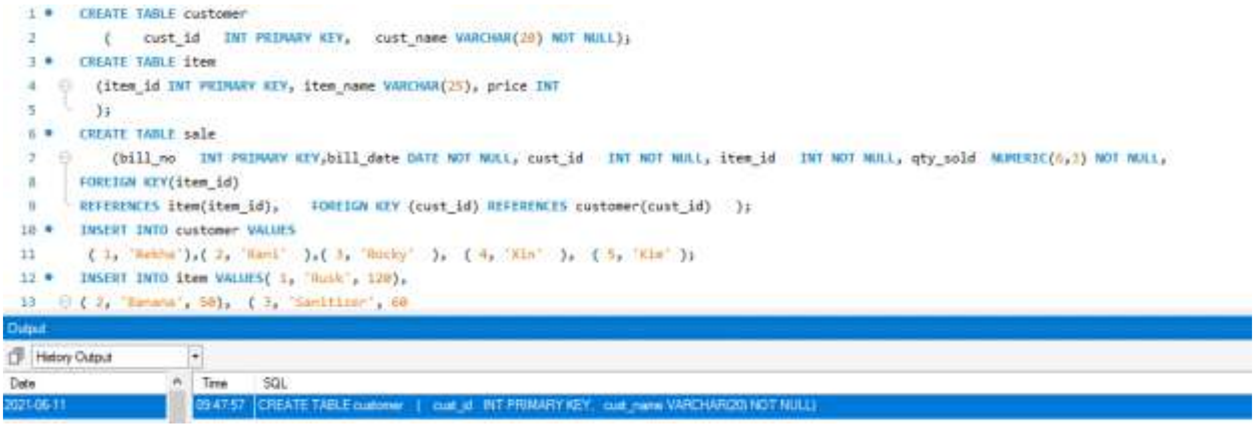
Item(item_id: integer, item_name: string, price: integer)

Sale(bill_no: integer, bill_date: date, cust_id: integer, item_id: integer, qty_sold: integer)

For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

Output

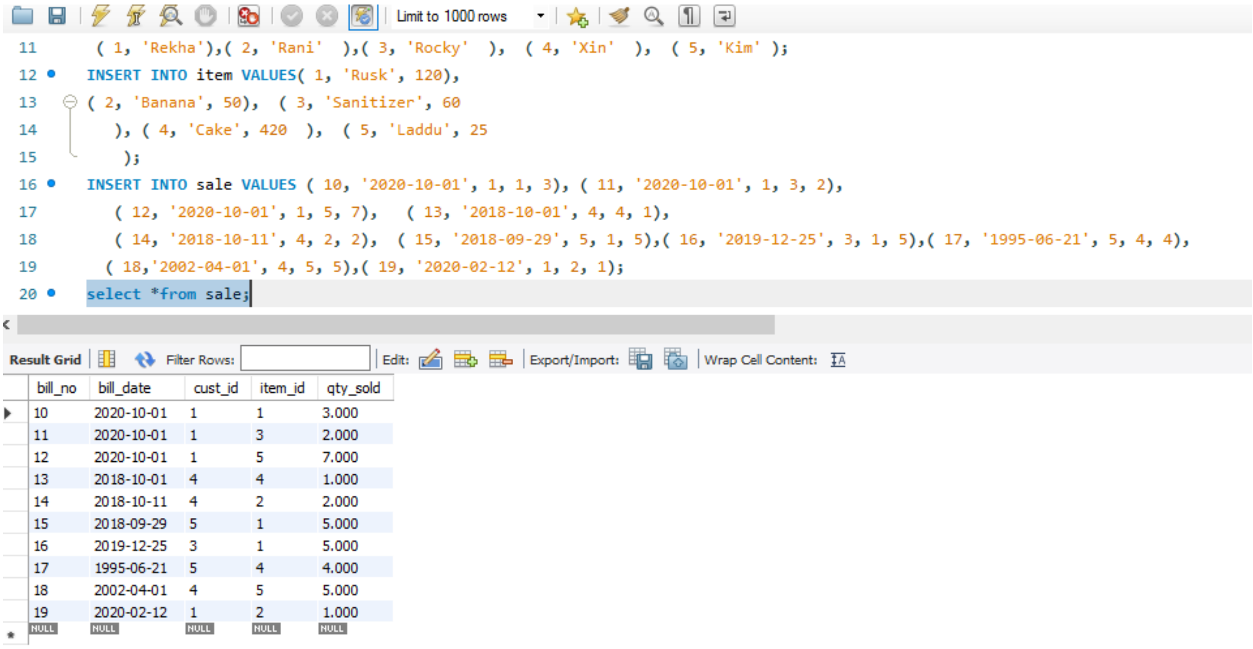


```
CREATE TABLE customer
  (    cust_id   INT PRIMARY KEY,   cust_name VARCHAR(20) NOT NULL);
CREATE TABLE item
 (item_id INT PRIMARY KEY, item_name VARCHAR(25), price INT
 );
CREATE TABLE sale
   (bill_no   INT PRIMARY KEY,bill_date DATE NOT NULL, cust_id   INT NOT NULL, item_id   INT NOT NULL, qty_sold  NUMERIC(6,2) NOT NULL,
FOREIGN KEY(item_id)
REFERENCES item(item_id),   FOREIGN KEY (cust_id) REFERENCES customer(cust_id)   );
INSERT INTO customer VALUES
 ( 1, 'Rekha' ),( 2, 'Rani'  ),( 3, 'Rocky'  ),  ( 4, 'Xin'  ),  ( 5, 'Kim' );
INSERT INTO item VALUES( 1, 'Rusk', 120),
( 2, 'Banana', 50),  ( 3, 'Sanitizer', 60
```

```
Date          Time      SQL
2021-06-11    09:47:57  CREATE TABLE customer  |  cust_id  INT PRIMARY KEY,  cust_name VARCHAR(20) NOT NULL)
```

b) Insert details of 5 customers, 5 items and 10 sales details. There should be one customer ‘rekha’ who had purchased 3 different products on the same date. And there should be atleast one customer who had purchased 2 different products on the same date in the year ‘2018’.

Output

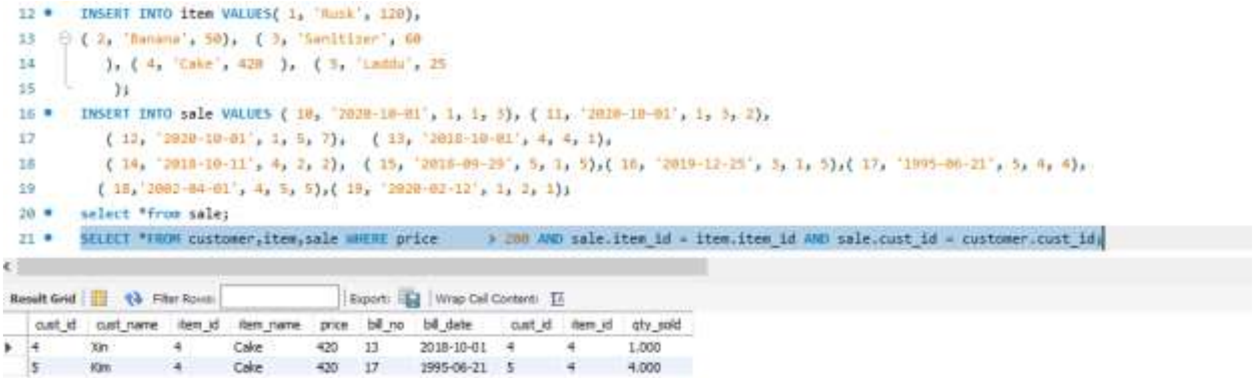


```
     ( 1, 'Rekha'),( 2, 'Rani'  ),( 3, 'Rocky'  ),  ( 4, 'Xin'  ),  ( 5, 'Kim' );
INSERT INTO item VALUES( 1, 'Rusk', 120),
( 2, 'Banana', 50),  ( 3, 'Sanitizer', 60
   ), ( 4, 'Cake', 420  ),  ( 5, 'Laddu', 25
    );
INSERT INTO sale VALUES ( 10, '2020-10-01', 1, 1, 3), ( 11, '2020-10-01', 1, 3, 2),
   ( 12, '2020-10-01', 1, 5, 7),   ( 13, '2018-10-01', 4, 4, 1),
   ( 14, '2018-10-11', 4, 2, 2),  ( 15, '2018-09-29', 5, 1, 5),( 16, '2019-12-25', 3, 1, 5),( 17, '1995-06-21', 5, 4, 4),
  ( 18,'2002-04-01', 4, 5, 5),( 19, '2020-02-12', 1, 2, 1);
select *from sale;
```

bill_no	bill_date	cust_id	item_id	qty_sold
10	2020-10-01	1	1	3.000
11	2020-10-01	1	3	2.000
12	2020-10-01	1	5	7.000
13	2018-10-01	4	4	1.000
14	2018-10-11	4	2	2.000
15	2018-09-29	5	1	5.000
16	2019-12-25	3	1	5.000
17	1995-06-21	5	4	4.000
18	2002-04-01	4	5	5.000
19	2020-02-12	1	2	1.000
NULL	NULL	NULL	NULL	NULL

c) List the details of the customer who have bought a product which has a price>200 .

Output

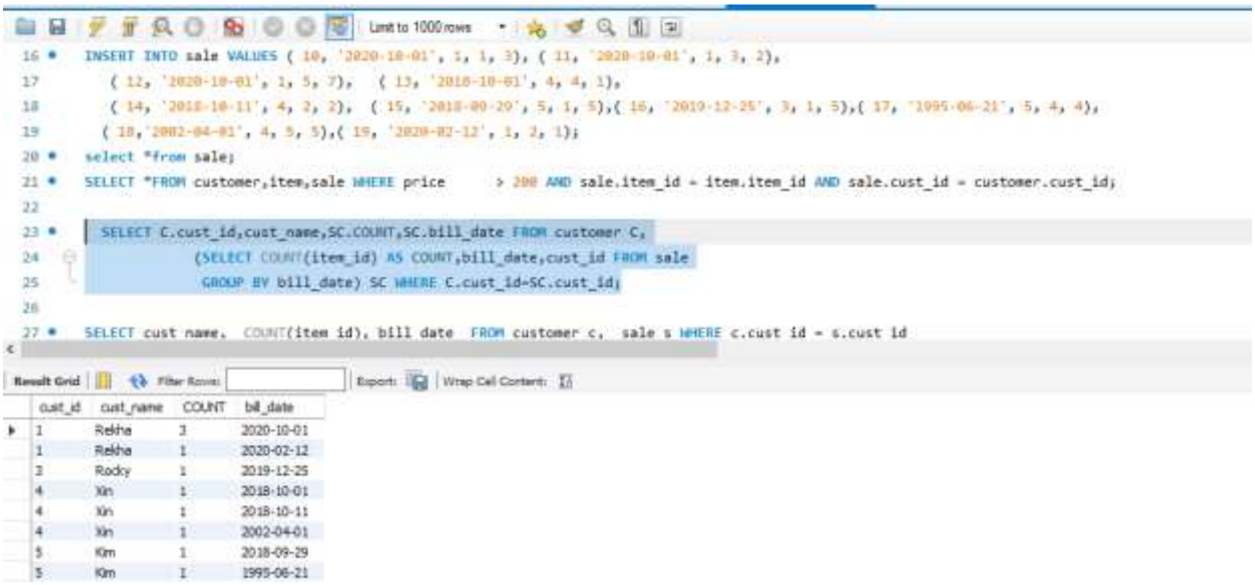


```
INSERT INTO item VALUES( 1, 'Rusk', 120),
( 2, 'Banana', 50),  ( 3, 'Sanitizer', 60
   ), ( 4, 'Cake', 420  ),  ( 5, 'Laddu', 25
    );
INSERT INTO sale VALUES ( 10, '2020-10-01', 1, 1, 3), ( 11, '2020-10-01', 1, 3, 2),
   ( 12, '2020-10-01', 1, 5, 7),   ( 13, '2018-10-01', 4, 4, 1),
   ( 14, '2018-10-11', 4, 2, 2),  ( 15, '2018-09-29', 5, 1, 5),( 16, '2019-12-25', 3, 1, 5),( 17, '1995-06-21', 5, 4, 4),
  ( 18,'2002-04-01', 4, 5, 5),( 19, '2020-02-12', 1, 2, 1);
select *from sale;
SELECT *FROM customer,item,sale WHERE price       > 200 AND sale.item_id = item.item_id AND sale.cust_id = customer.cust_id;
```

cust_id	cust_name	item_id	item_name	price	bill_no	bill_date	cust_id	item_id	qty_sold
4	Xin	4	Cake	420	13	2018-10-01	4	4	1.000
5	Kim	4	Cake	420	17	1995-06-21	5	4	4.000

d) Give a count of how many products have been bought by each customer group by bill date.

Output

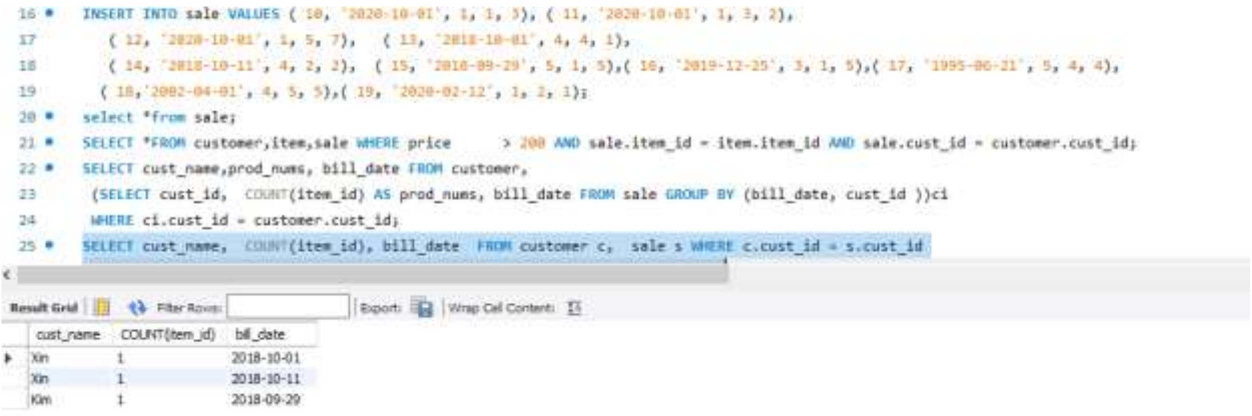


```
INSERT INTO sale VALUES ( 10, '2020-10-01', 1, 1, 3), ( 11, '2020-10-01', 1, 3, 2),
   ( 12, '2020-10-01', 1, 5, 7),  ( 13, '2018-10-01', 4, 4, 1),
   ( 14, '2018-10-11', 4, 2, 2),  ( 15, '2018-09-29', 5, 1, 5),( 16, '2019-12-25', 3, 1, 5),( 17, '1995-06-21', 5, 4, 4),
  ( 18,'2002-04-01', 4, 5, 5),( 19, '2020-02-12', 1, 2, 1);
select *from sale;
SELECT *FROM customer,item,sale WHERE price     > 200 AND sale.item_id = item.item_id AND sale.cust_id = customer.cust_id;

 SELECT C.cust_id,cust_name,SC.COUNT,SC.bill_date FROM customer C,
             (SELECT COUNT(item_id) AS COUNT,bill_date,cust_id FROM sale
              GROUP BY bill_date) SC WHERE C.cust_id=SC.cust_id;

SELECT cust name,  COUNT(item id), bill date  FROM customer c,  sale s WHERE c.cust id = s.cust id
```

cust_id	cust_name	COUNT	bill_date
1	Rekha	3	2020-10-01
1	Rekha	1	2020-02-12
3	Rocky	1	2019-12-25
4	Xin	1	2018-10-01
4	Xin	1	2018-10-11
4	Xin	1	2002-04-01
5	Kim	1	2018-09-29
5	Kim	1	1995-06-21

e) Give a count of how many products have been bought by each customer group by bill date only for the year 2018.

Output

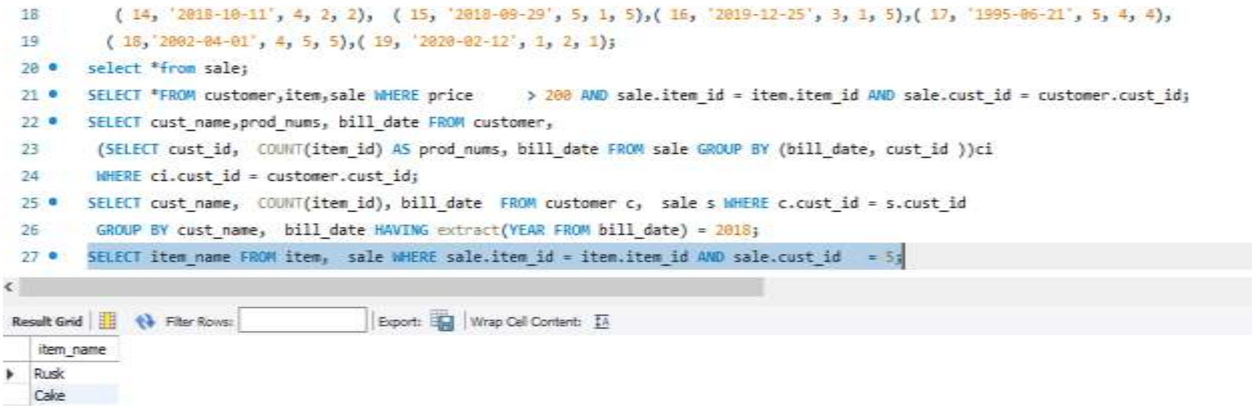


```
INSERT INTO sale VALUES ( 10, '2020-10-01', 1, 1, 3), ( 11, '2020-10-01', 1, 3, 2),
   ( 12, '2020-10-01', 1, 5, 7),   ( 13, '2018-10-01', 4, 4, 1),
   ( 14, '2018-10-11', 4, 2, 2),  ( 15, '2018-09-29', 5, 1, 5),( 16, '2019-12-25', 3, 1, 5),( 17, '1995-06-21', 5, 4, 4),
  ( 18,'2002-04-01', 4, 5, 5),( 19, '2020-02-12', 1, 2, 1);
select *from sale;
SELECT *FROM customer,item,sale WHERE price     > 200 AND sale.item_id = item.item_id AND sale.cust_id = customer.cust_id;
SELECT cust_name,prod_nums, bill_date FROM customer,
 (SELECT cust_id,  COUNT(item_id) AS prod_nums, bill_date FROM sale GROUP BY (bill_date, cust_id ))ci
 WHERE ci.cust_id = customer.cust_id;
SELECT cust_name,  COUNT(item_id), bill_date  FROM customer c,  sale s WHERE c.cust_id = s.cust_id
```

cust_name	COUNT(item_id)	bill_date
Xin	1	2018-10-01
Xin	1	2018-10-11
Kim	1	2018-09-29

f) Give a list of products bought by a customer having cust_id as 5

Output

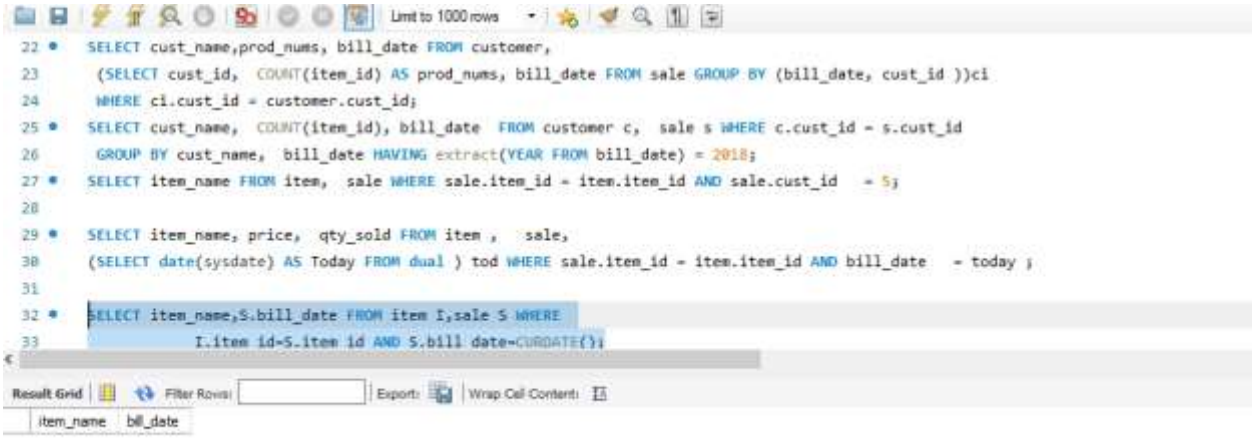


```
   ( 14, '2018-10-11', 4, 2, 2),  ( 15, '2018-09-29', 5, 1, 5),( 16, '2019-12-25', 3, 1, 5),( 17, '1995-06-21', 5, 4, 4),
  ( 18,'2002-04-01', 4, 5, 5),( 19, '2020-02-12', 1, 2, 1);
select *from sale;
SELECT *FROM customer,item,sale WHERE price     > 200 AND sale.item_id = item.item_id AND sale.cust_id = customer.cust_id;
SELECT cust_name,prod_nums, bill_date FROM customer,
 (SELECT cust_id,  COUNT(item_id) AS prod_nums, bill_date FROM sale GROUP BY (bill_date, cust_id ))ci
 WHERE ci.cust_id = customer.cust_id;
SELECT cust_name,  COUNT(item_id), bill_date  FROM customer c,  sale s WHERE c.cust_id = s.cust_id
 GROUP BY cust_name,  bill_date HAVING extract(YEAR FROM bill_date) = 2018;
SELECT item_name FROM item,  sale WHERE sale.item_id = item.item_id AND sale.cust_id   = 5;
```

item_name
Rusk
Cake

g) List the item details which are sold as of today

Output

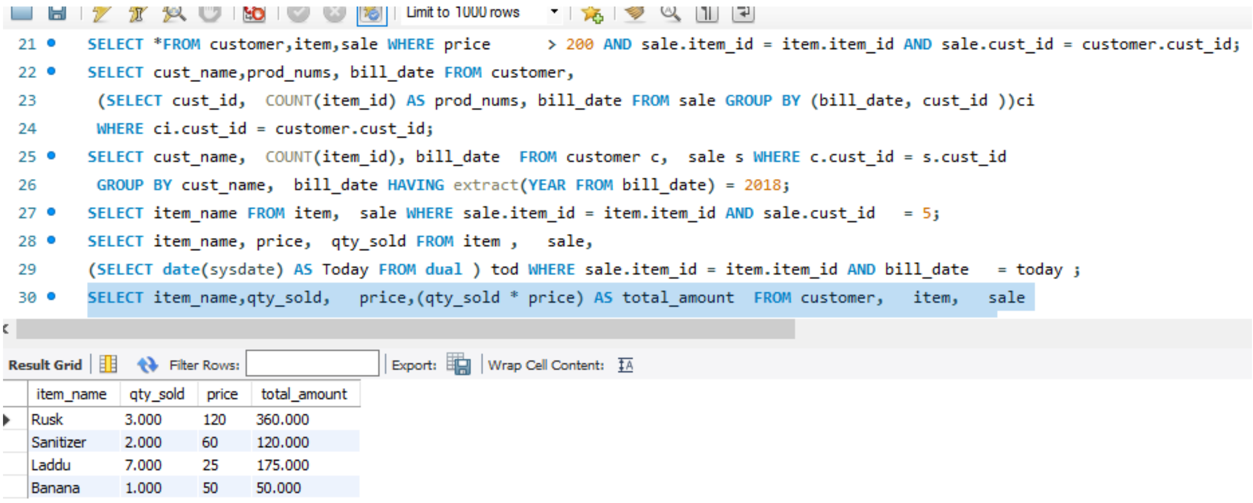


```
SELECT cust_name,prod_nums, bill_date FROM customer,
 (SELECT cust_id,  COUNT(item_id) AS prod_nums, bill_date FROM sale GROUP BY (bill_date, cust_id ))ci
 WHERE ci.cust_id = customer.cust_id;
SELECT cust_name,  COUNT(item_id), bill_date  FROM customer c,  sale s WHERE c.cust_id = s.cust_id
 GROUP BY cust_name,  bill_date HAVING extract(YEAR FROM bill_date) = 2018;
SELECT item_name FROM item,  sale WHERE sale.item_id = item.item_id AND sale.cust_id   = 5;

SELECT item_name, price,  qty_sold FROM item ,   sale,
(SELECT date(sysdate) AS Today FROM dual ) tod WHERE sale.item_id = item.item_id AND bill_date   = today ;

SELECT item_name,S.bill_date FROM item I,sale S WHERE
            I.item_id=S.item_id AND S.bill_date=CURDATE();
```

item_name	bill_date

h) Print the bill in a neat format with the quantity sold, price of the item and the final amount of customer ‘rekha’

Output



```
SELECT *FROM customer,item,sale WHERE price      > 200 AND sale.item_id = item.item_id AND sale.cust_id = customer.cust_id;
SELECT cust_name,prod_nums, bill_date FROM customer,
 (SELECT cust_id,  COUNT(item_id) AS prod_nums, bill_date FROM sale GROUP BY (bill_date, cust_id ))ci
 WHERE ci.cust_id = customer.cust_id;
SELECT cust_name,  COUNT(item_id), bill_date  FROM customer c,  sale s WHERE c.cust_id = s.cust_id
 GROUP BY cust_name,  bill_date HAVING extract(YEAR FROM bill_date) = 2018;
SELECT item_name FROM item,  sale WHERE sale.item_id = item.item_id AND sale.cust_id   = 5;
SELECT item_name, price,  qty_sold FROM item ,   sale,
(SELECT date(sysdate) AS Today FROM dual ) tod WHERE sale.item_id = item.item_id AND bill_date   = today ;
SELECT item_name,qty_sold,   price,(qty_sold * price) AS total_amount  FROM customer,   item,   sale
```

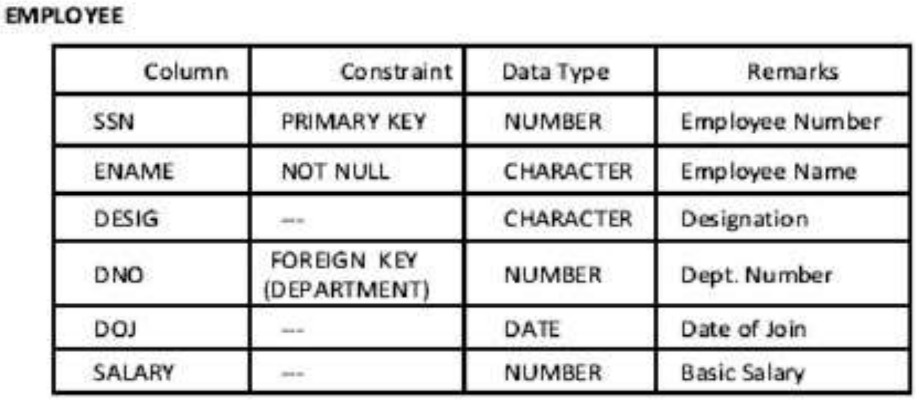
item_name	qty_sold	price	total_amount
Rusk	3.000	120	360.000
Sanitizer	2.000	60	120.000
Laddu	7.000	25	175.000
Banana	1.000	50	50.000

LAB CYCLE 3

QUESTION SET 3 date:03-08-2021

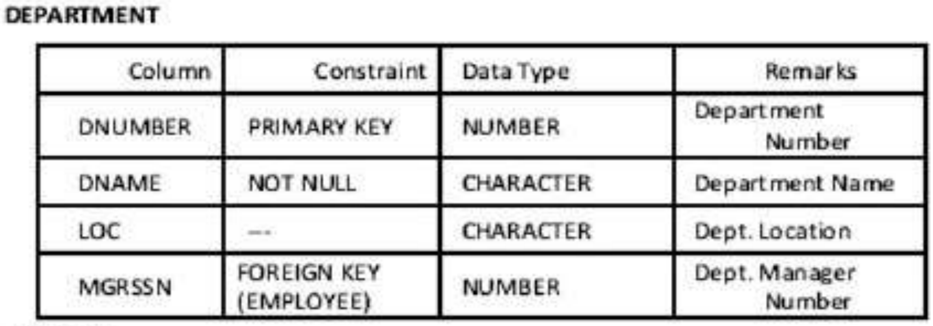
Create the following tables.

- Primary key, SSN of EMPLOYEE should be created as a sequence starting at 1.
- There should be at least 8 employees and 5 departments
- Check salary range of employees is between 30,000 and 75,000 using check predicate.



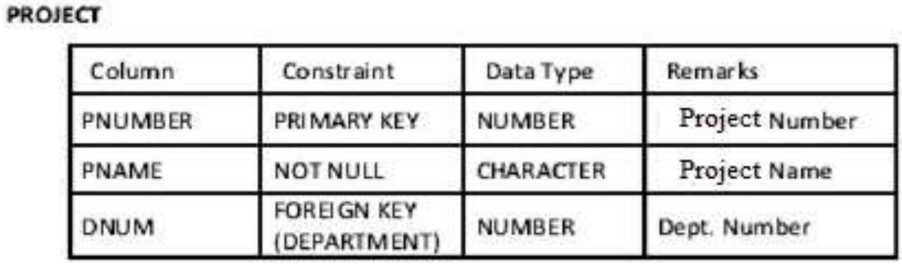
EMPLOYEE

Column	Constraint	Data Type	Remarks
SSN	PRIMARY KEY	NUMBER	Employee Number
ENAME	NOT NULL	CHARACTER	Employee Name
DESIG	---	CHARACTER	Designation
DNO	FOREIGN KEY (DEPARTMENT)	NUMBER	Dept. Number
DOJ	---	DATE	Date of Join
SALARY	---	NUMBER	Basic Salary



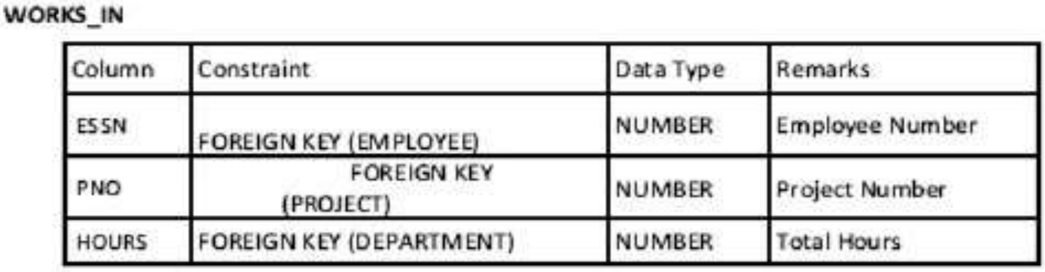
DEPARTMENT

Column	Constraint	Data Type	Remarks
DNUMBER	PRIMARY KEY	NUMBER	Department Number
DNAME	NOT NULL	CHARACTER	Department Name
LOC	---	CHARACTER	Dept. Location
MGRSSN	FOREIGN KEY (EMPLOYEE)	NUMBER	Dept. Manager Number



PROJECT

Column	Constraint	Data Type	Remarks
PNUMBER	PRIMARY KEY	NUMBER	Project Number
PNAME	NOT NULL	CHARACTER	Project Name
DNUM	FOREIGN KEY (DEPARTMENT)	NUMBER	Dept. Number

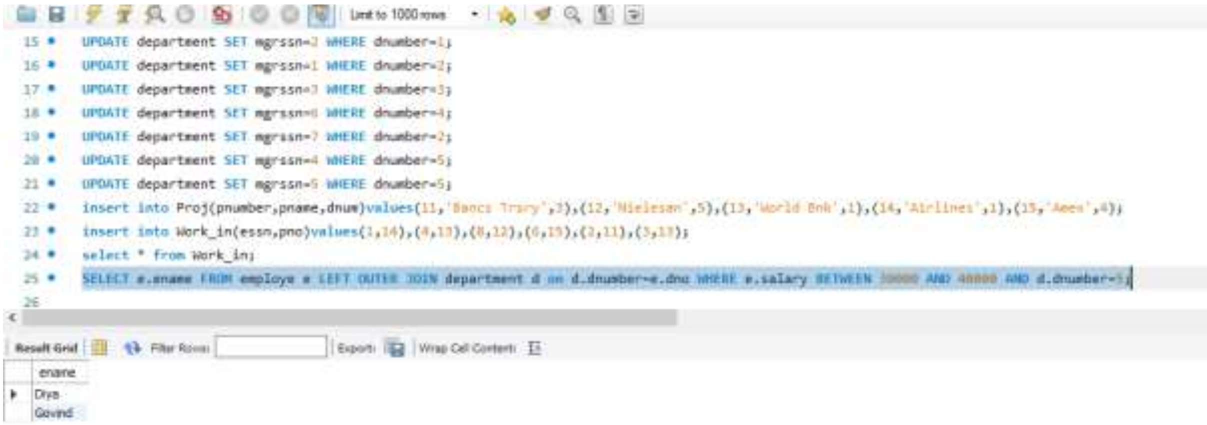


WORKS_IN

Column	Constraint	Data Type	Remarks
ESSN	FOREIGN KEY (EMPLOYEE)	NUMBER	Employee Number
PNO	FOREIGN KEY (PROJECT)	NUMBER	Project Number
HOURS	FOREIGN KEY (DEPARTMENT)	NUMBER	Total Hours

1. Retrieve all employees in department 5 whose salary is between Rs 30,000 and Rs 40,000.

Output



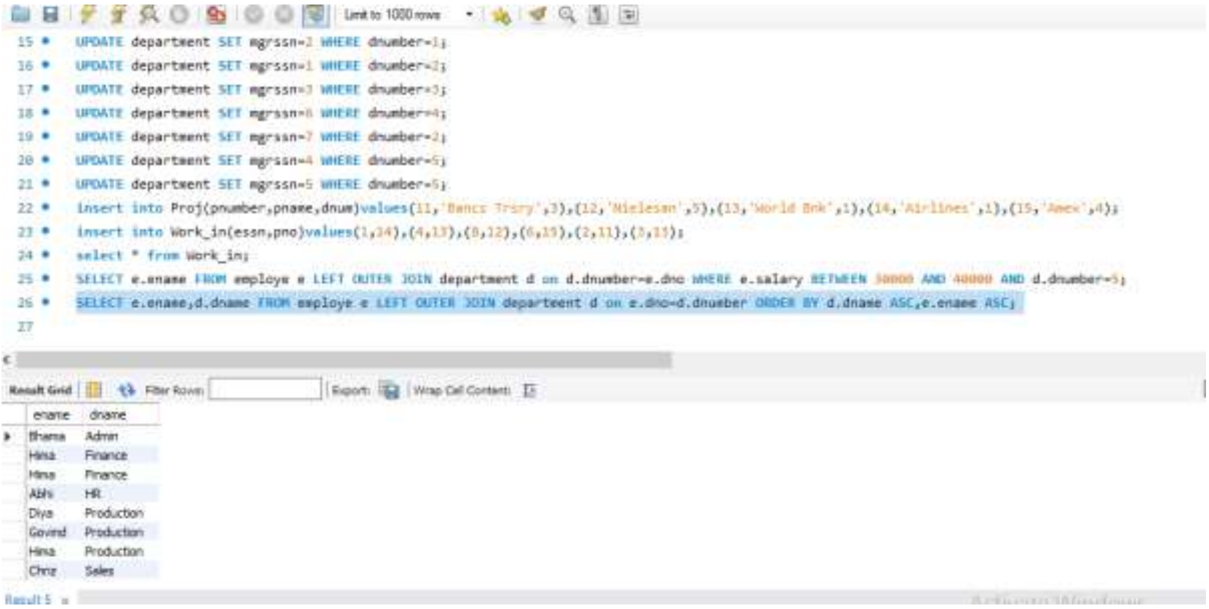
Limit to 1000 rows

```
UPDATE department SET mgrssn=2 WHERE dnumber=1;
UPDATE department SET mgrssn=1 WHERE dnumber=2;
UPDATE department SET mgrssn=3 WHERE dnumber=3;
UPDATE department SET mgrssn=6 WHERE dnumber=4;
UPDATE department SET mgrssn=7 WHERE dnumber=2;
UPDATE department SET mgrssn=4 WHERE dnumber=5;
UPDATE department SET mgrssn=5 WHERE dnumber=5;
insert into Proj(pnumber,pname,dnum)values(11,'Bancs Trsry',3),(12,'Nielesan',5),(13,'World Bnk',1),(14,'Airlines',1),(15,'Amex',4);
insert into Work_in(essn,pno)values(1,14),(4,13),(8,12),(6,15),(2,11),(3,13);
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
```

ename
Diya
Govind

2. Retrieve a list of employees and the projects they are working on, where the departments and the employees within the department are alphabetically by name.

Output



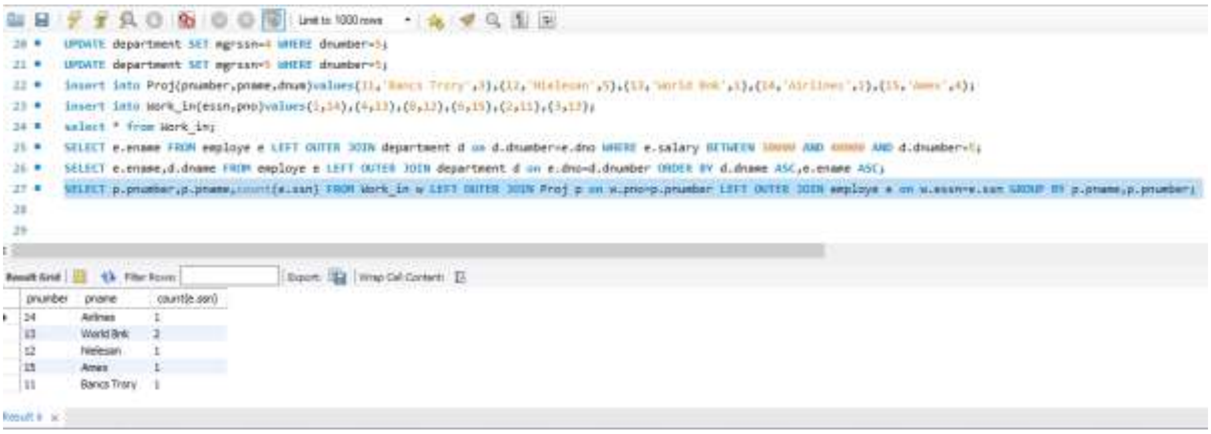
Limit to 1000 rows

```
UPDATE department SET mgrssn=2 WHERE dnumber=1;
UPDATE department SET mgrssn=1 WHERE dnumber=2;
UPDATE department SET mgrssn=3 WHERE dnumber=3;
UPDATE department SET mgrssn=6 WHERE dnumber=4;
UPDATE department SET mgrssn=7 WHERE dnumber=2;
UPDATE department SET mgrssn=4 WHERE dnumber=5;
UPDATE department SET mgrssn=5 WHERE dnumber=5;
insert into Proj(pnumber,pname,dnum)values(11,'Bancs Trsry',3),(12,'Nielesan',5),(13,'World Bnk',1),(14,'Airlines',1),(15,'Amex',4);
insert into Work_in(essn,pno)values(1,14),(4,13),(8,12),(6,15),(2,11),(3,13);
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
```

ename	dname
Bhama	Admin
Hina	Finance
Hina	Finance
Abhi	HR
Diya	Production
Govind	Production
Hina	Production
Chriz	Sales

3. Retrieve the project number, the project name, and the number of employees who work in each project.

Output

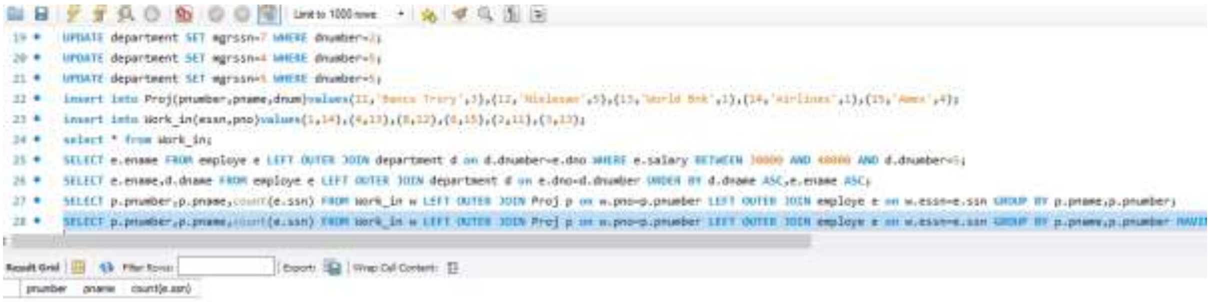


```
UPDATE department SET mgrssn=4 WHERE dnumber=5;
UPDATE department SET mgrssn=5 WHERE dnumber=5;
insert into Proj(pnumber,pname,dnum)values(11,'Bancs Trory',3),(12,'Nielesan',5),(13,'World Bnk',1),(14,'Airlines',1),(15,'Amex',4);
insert into Work_in(essn,pno)values(1,14),(4,13),(8,12),(6,15),(2,11),(3,13);
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pnumber;
```

```
pnumber  pname        count(e.ssn)
14       Airlines     1
13       World Bnk    2
12       Nielesan     1
15       Amex         1
11       Bancs Trory  1
```

4. For the project on which more than two employees work, retrieve the project number, the project name, and the number of employees who work on the project.

Output

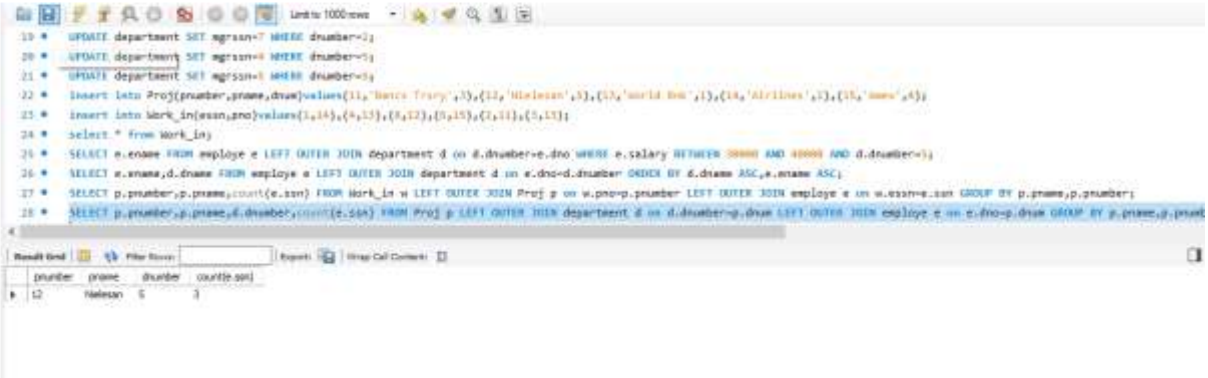


```
UPDATE department SET mgrssn=7 WHERE dnumber=2;
UPDATE department SET mgrssn=4 WHERE dnumber=5;
UPDATE department SET mgrssn=5 WHERE dnumber=5;
insert into Proj(pnumber,pname,dnum)values(11,'Bancs Trory',3),(12,'Nielesan',5),(13,'World Bnk',1),(14,'Airlines',1),(15,'Amex',4);
insert into Work_in(essn,pno)values(1,14),(4,13),(8,12),(6,15),(2,11),(3,13);
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pnumber;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pnumber HAVI
```

```
pnumber  pname  count(e.ssn)
```

5. For each project, retrieve the project number, the project name, and the number of employees from department 5 who work on the project.

Output

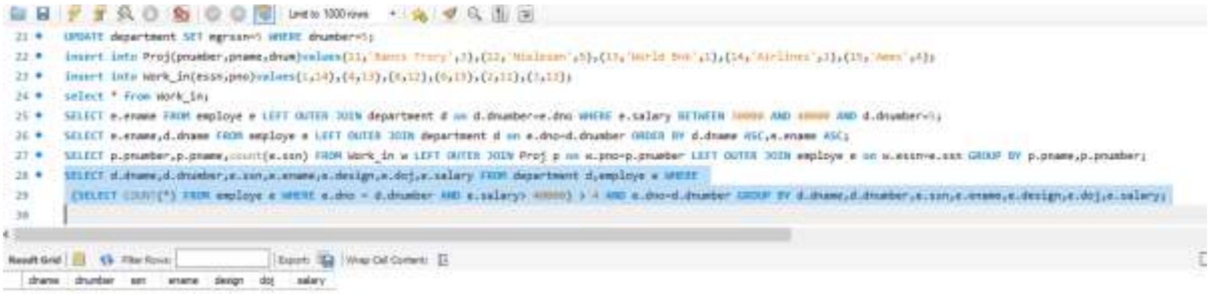


```
UPDATE department SET mgrssn=7 WHERE dnumber=2;
UPDATE department SET mgrssn=4 WHERE dnumber=5;
UPDATE department SET mgrssn=5 WHERE dnumber=5;
insert into Proj(pnumber,pname,dnum)values(11,'Bancs Trory',3),(12,'Nielesan',5),(13,'World Bnk',1),(14,'Airlines',1),(15,'Amex',4);
insert into Work_in(essn,pno)values(1,14),(4,13),(8,12),(5,15),(2,11),(3,13);
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pnumber;
SELECT p.pnumber,p.pname,d.dnumber,count(e.ssn) FROM Proj p LEFT OUTER JOIN department d on d.dnumber=p.dnum LEFT OUTER JOIN employe e on e.dno=p.dnum GROUP BY p.pname,p.pnumb
```

```
pnumber  pname     dnumber  count(e.ssn)
12       Nielesan  5        3
```

6. For the departments having more than five employees, display the department id and the number and details of employees earning more than Rs 40,000 per month.

Output



```
UPDATE department SET mgrssn=5 WHERE dnumber=5;
insert into Proj(pnumber,pname,dnum)values(11,'Bancs Trary',3),(12,'Nielsen',5),(13,'World Bnk',1),(14,'Airlines',1),(15,'Amex',4);
insert into Work_in(essn,pno)values(1,14),(4,13),(6,12),(6,15),(2,11),(3,13);
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pnumber;
SELECT d.dname,d.dnumber,e.ssn,e.ename,e.design,e.doj,e.salary FROM department d,employe e WHERE
 (SELECT COUNT(*) FROM employe e WHERE e.dno = d.dnumber AND e.salary> 40000) > 4 AND e.dno=d.dnumber GROUP BY d.dname,d.dnumber,e.ssn,e.ename,e.design,e.doj,e.salary;
```

dname	dnumber	ssn	ename	design	doj	salary

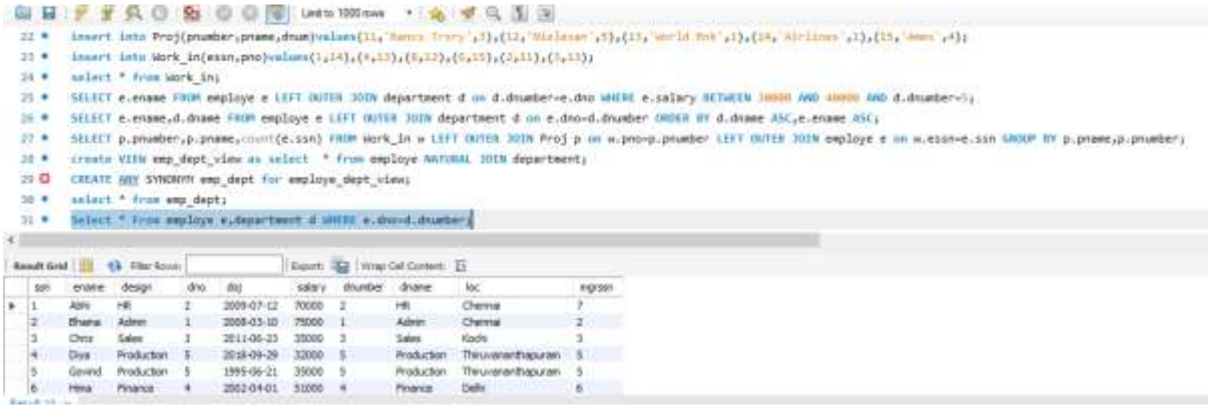
7. Create a synonym for the VIEW created on natural join of emp and dept tables.

Output

8. Use the tables Employee, and Department. Perform the operations as mentioned below:

(a) Display the employee details, departments that the departments are same in both the emp and dept. (Equi-join)

Output

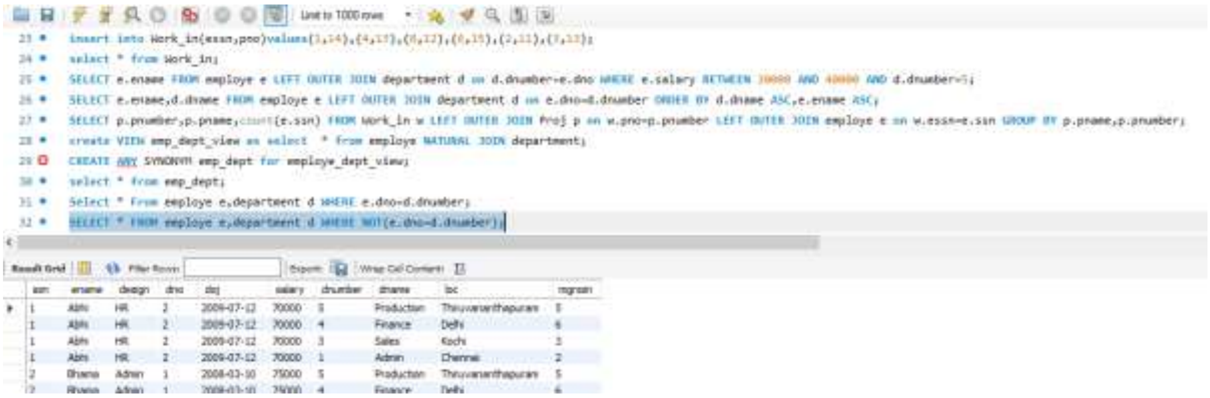


```
insert into Proj(pnumber,pname,dnum)values(11,'Bancs Trary',3),(12,'Nielsen',5),(13,'World Bnk',1),(14,'Airlines',1),(15,'Amex',4);
insert into Work_in(essn,pno)values(1,14),(4,13),(6,12),(6,15),(2,11),(3,13);
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pnumber;
create VIEW emp_dept_view as select  * from employe NATURAL JOIN department;
CREATE ANY SYNONYM emp_dept for employe_dept_view;
select * from emp_dept;
Select * from employe e,department d WHERE e.dno=d.dnumber;
```

ssn	ename	design	dno	doj	salary	dnumber	dname	loc	mgrssn
1	Abhi	HR	2	2009-07-12	70000	2	HR	Chennai	7
2	Bhama	Admin	1	2008-03-10	75000	1	Admin	Chennai	2
3	Chris	Sales	3	2011-06-23	35000	3	Sales	Kochi	3
4	Diya	Production	5	2018-09-29	32000	5	Production	Thiruvananthapuram	5
5	Govind	Production	5	1995-06-21	35000	5	Production	Thiruvananthapuram	5
6	Hima	Finance	4	2002-04-01	31000	4	Finance	Delhi	6

(b) Display the employee details, departments that the departments are not same in both the emp and dept. (Non Equi-join)

Output

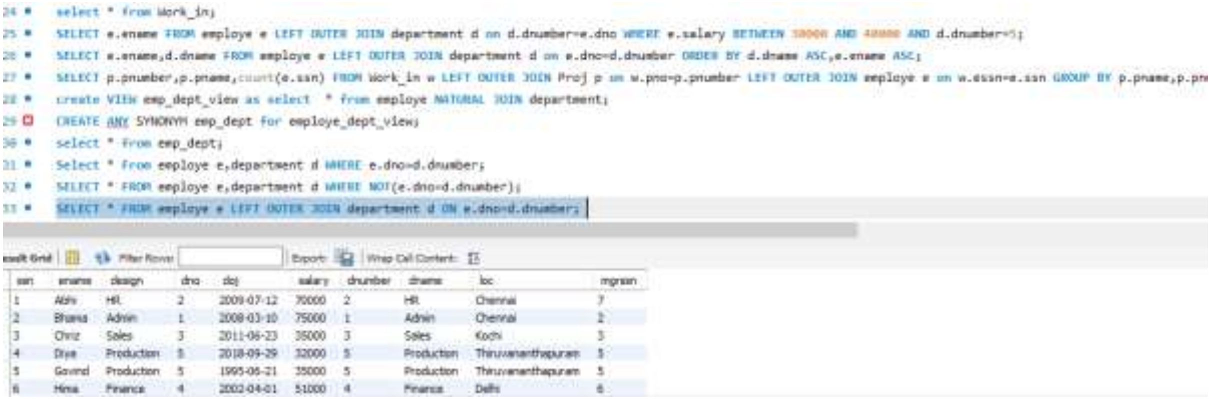


```sql
insert into Work_in(essn,pno)values(1,14),(4,13),(6,12),(6,15),(2,11),(7,13);
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pnumber;
create VIEW emp_dept_view as select  * from employe NATURAL JOIN department;
CREATE ANY SYNONYM emp_dept for employe_dept_view;
select * from emp_dept;
Select * From employe e,department d WHERE e.dno=d.dnumber;
SELECT * FROM employe e,department d WHERE NOT(e.dno=d.dnumber);
```

ssn	ename	design	dno	doj	salary	dnumber	dname	loc	mgrssn
1	Abhi	HR	2	2009-07-12	70000	5	Production	Thiruvananthapuram	5
1	Abhi	HR	2	2009-07-12	70000	4	Finance	Delhi	6
1	Abhi	HR	2	2009-07-12	70000	3	Sales	Kochi	3
1	Abhi	HR	2	2009-07-12	70000	1	Admin	Chennai	2
2	Bhama	Admin	1	2008-03-10	75000	5	Production	Thiruvananthapuram	5
2	Bhama	Admin	1	2008-03-10	75000	4	Finance	Delhi	6

(c) Perform Left outer join on the emp and dept tables.

Output

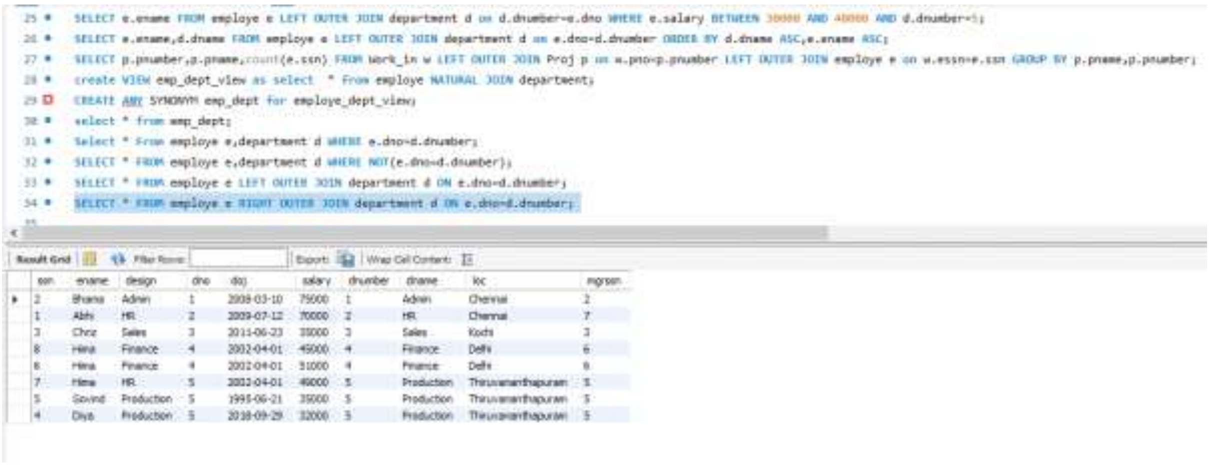


```sql
select * from Work_in;
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pn
create VIEW emp_dept_view as select  * from employe NATURAL JOIN department;
CREATE ANY SYNONYM emp_dept for employe_dept_view;
select * from emp_dept;
Select * From employe e,department d WHERE e.dno=d.dnumber;
SELECT * FROM employe e,department d WHERE NOT(e.dno=d.dnumber);
SELECT * FROM employe e LEFT OUTER JOIN department d ON e.dno=d.dnumber;
```

ssn	ename	design	dno	doj	salary	dnumber	dname	loc	mgrssn
1	Abhi	HR	2	2009-07-12	70000	2	HR	Chennai	7
2	Bhama	Admin	1	2008-03-10	75000	1	Admin	Chennai	2
3	Chriz	Sales	3	2011-06-23	35000	3	Sales	Kochi	3
4	Diya	Production	5	2018-09-29	32000	5	Production	Thiruvananthapuram	5
5	Govind	Production	5	1995-06-21	35000	5	Production	Thiruvananthapuram	5
6	Hima	Finance	4	2002-04-01	51000	4	Finance	Delhi	6

(d) Perform Right outer join on the emp and dept tables.

Output

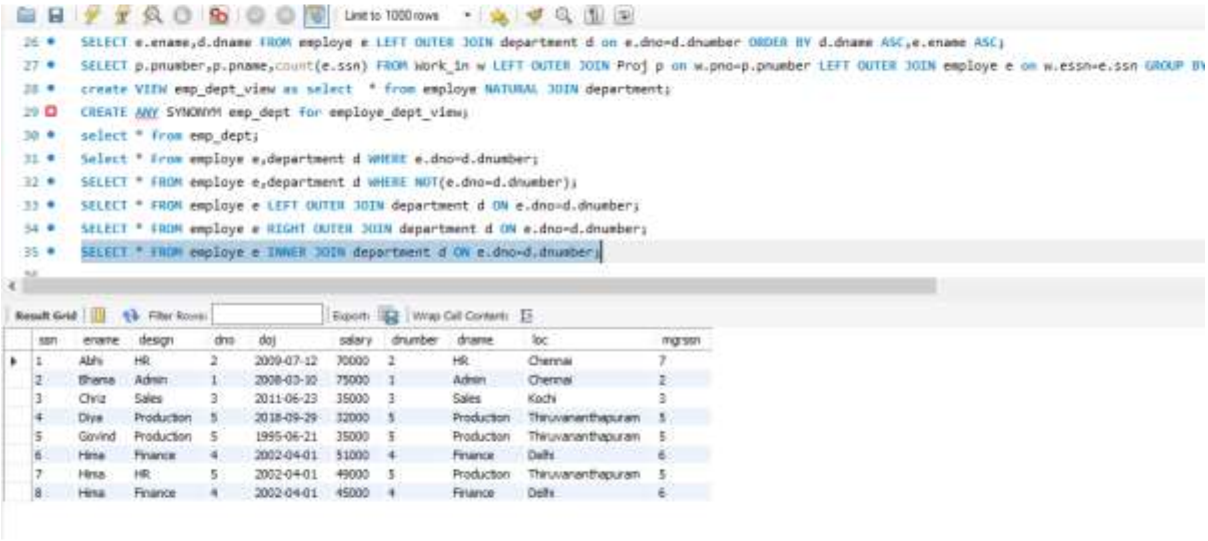


```sql
SELECT e.ename FROM employe e LEFT OUTER JOIN department d on d.dnumber=e.dno WHERE e.salary BETWEEN 30000 AND 40000 AND d.dnumber=5;
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY p.pname,p.pnumber;
create VIEW emp_dept_view as select  * from employe NATURAL JOIN department;
CREATE ANY SYNONYM emp_dept for employe_dept_view;
select * from emp_dept;
Select * From employe e,department d WHERE e.dno=d.dnumber;
SELECT * FROM employe e,department d WHERE NOT(e.dno=d.dnumber);
SELECT * FROM employe e LEFT OUTER JOIN department d ON e.dno=d.dnumber;
SELECT * FROM employe e RIGHT OUTER JOIN department d ON e.dno=d.dnumber;
```

ssn	ename	design	dno	doj	salary	dnumber	dname	loc	mgrssn
2	Bhama	Admin	1	2008-03-10	75000	1	Admin	Chennai	2
1	Abhi	HR	2	2009-07-12	70000	2	HR	Chennai	7
3	Chriz	Sales	3	2011-06-23	35000	3	Sales	Kochi	3
8	Hima	Finance	4	2002-04-01	45000	4	Finance	Delhi	6
6	Hima	Finance	4	2002-04-01	51000	4	Finance	Delhi	6
7	Hima	HR	5	2002-04-01	49000	5	Production	Thiruvananthapuram	5
5	Govind	Production	5	1995-06-21	35000	5	Production	Thiruvananthapuram	5
4	Diya	Production	5	2018-09-29	32000	5	Production	Thiruvananthapuram	5

(e) Perform inner join on the emp and dept tables.

Output



```
SELECT e.ename,d.dname FROM employe e LEFT OUTER JOIN department d on e.dno=d.dnumber ORDER BY d.dname ASC,e.ename ASC;
SELECT p.pnumber,p.pname,count(e.ssn) FROM Work_in w LEFT OUTER JOIN Proj p on w.pno=p.pnumber LEFT OUTER JOIN employe e on w.essn=e.ssn GROUP BY
create VIEW emp_dept_view as select  * from employe NATURAL JOIN department;
CREATE ANY SYNONYM emp_dept for employe_dept_view;
select * from emp_dept;
Select * from employe e,department d WHERE e.dno=d.dnumber;
SELECT * FROM employe e,department d WHERE NOT(e.dno=d.dnumber);
SELECT * FROM employe e LEFT OUTER JOIN department d ON e.dno=d.dnumber;
SELECT * FROM employe e RIGHT OUTER JOIN department d ON e.dno=d.dnumber;
SELECT * FROM employe e INNER JOIN department d ON e.dno=d.dnumber;
```

```
ssn  ename   design      dno  doj         salary  dnumber  dname       loc                  mgrssn
1    Abhi    HR          2    2009-07-12  70000   2        HR          Chennai              7
2    Bhama   Admin       1    2008-03-10  75000   1        Admin       Chennai              2
3    Chriz   Sales       3    2011-06-23  35000   3        Sales       Kochi                3
4    Diya    Production  5    2018-09-29  32000   5        Production  Thiruvananthapuram   5
5    Govind  Production  5    1995-06-21  35000   5        Production  Thiruvananthapuram   5
6    Hima    Finance     4    2002-04-01  51000   4        Finance     Delhi                6
7    Hima    HR          5    2002-04-01  49000   5        Production  Thiruvananthapuram   5
8    Hima    Finance     4    2002-04-01  45000   4        Finance     Delhi                6
```

LAB CYCLE 4

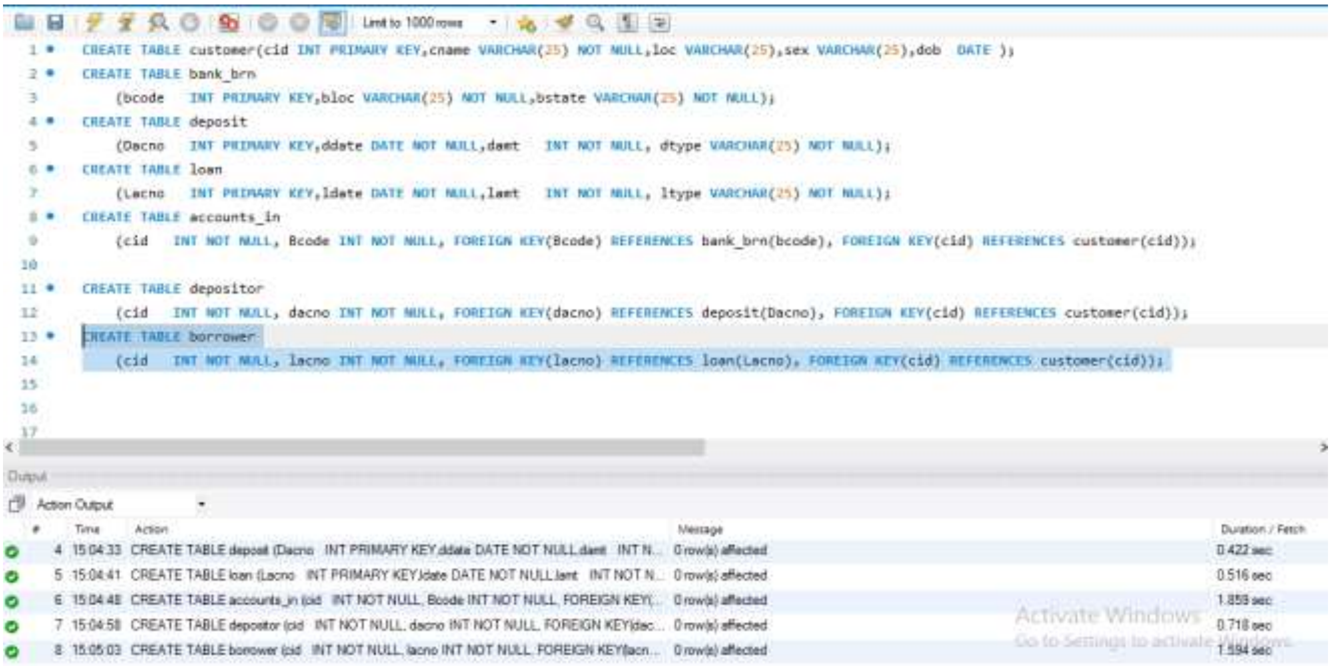
QUESTION SET 6 ***date:08-08-2021***

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

Output



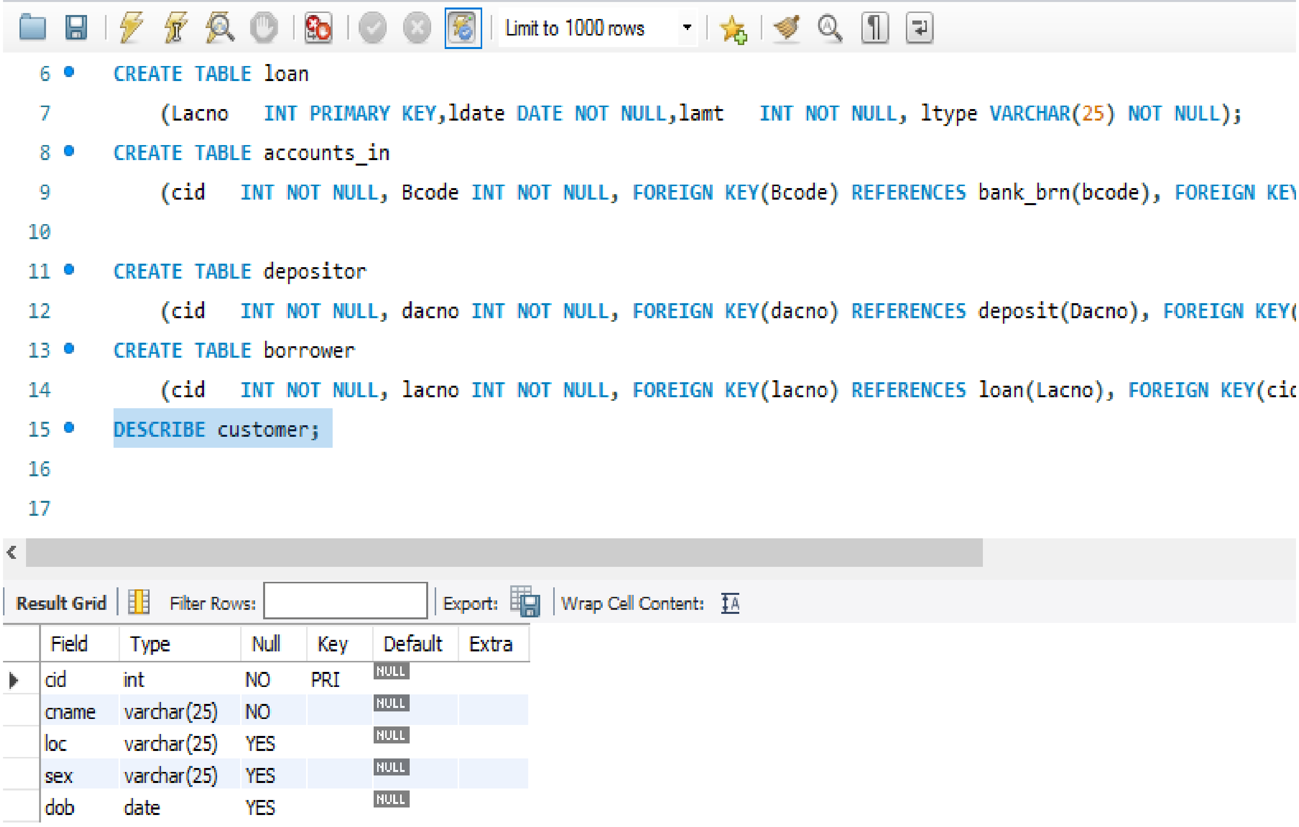
```
CREATE TABLE customer(cid INT PRIMARY KEY,cname VARCHAR(25) NOT NULL,loc VARCHAR(25),sex VARCHAR(25),dob  DATE );
CREATE TABLE bank_brn
    (bcode   INT PRIMARY KEY,bloc VARCHAR(25) NOT NULL,bstate VARCHAR(25) NOT NULL);
CREATE TABLE deposit
    (Dacno   INT PRIMARY KEY,ddate DATE NOT NULL,damt   INT NOT NULL, dtype VARCHAR(25) NOT NULL);
CREATE TABLE loan
    (Lacno   INT PRIMARY KEY,ldate DATE NOT NULL,lamt   INT NOT NULL, ltype VARCHAR(25) NOT NULL);
CREATE TABLE accounts_in
    (cid   INT NOT NULL, Bcode INT NOT NULL, FOREIGN KEY(Bcode) REFERENCES bank_brn(bcode), FOREIGN KEY(cid) REFERENCES customer(cid));

CREATE TABLE depositor
    (cid   INT NOT NULL, dacno INT NOT NULL, FOREIGN KEY(dacno) REFERENCES deposit(Dacno), FOREIGN KEY(cid) REFERENCES customer(cid));
CREATE TABLE borrower
    (cid   INT NOT NULL, lacno INT NOT NULL, FOREIGN KEY(lacno) REFERENCES loan(Lacno), FOREIGN KEY(cid) REFERENCES customer(cid));
```

```
   #  Time      Action                                                                   Message            Duration / Fetch
   4  15:04:33  CREATE TABLE deposit (Dacno   INT PRIMARY KEY,ddate DATE NOT NULL,damt   INT N...   0 row(s) affected   0.422 sec
   5  15:04:41  CREATE TABLE loan (Lacno   INT PRIMARY KEY,ldate DATE NOT NULL,lamt   INT NOT N...   0 row(s) affected   0.516 sec
   6  15:04:48  CREATE TABLE accounts_in (cid   INT NOT NULL, Bcode INT NOT NULL, FOREIGN KEY(...   0 row(s) affected   1.859 sec
   7  15:04:58  CREATE TABLE depositor (cid   INT NOT NULL, dacno INT NOT NULL, FOREIGN KEY(dac...   0 row(s) affected   0.718 sec
   8  15:05:03  CREATE TABLE borrower (cid   INT NOT NULL, lacno INT NOT NULL, FOREIGN KEY(lacn...   0 row(s) affected   1.594 sec
```

Activate Windows
Go to Settings to activate Windows.

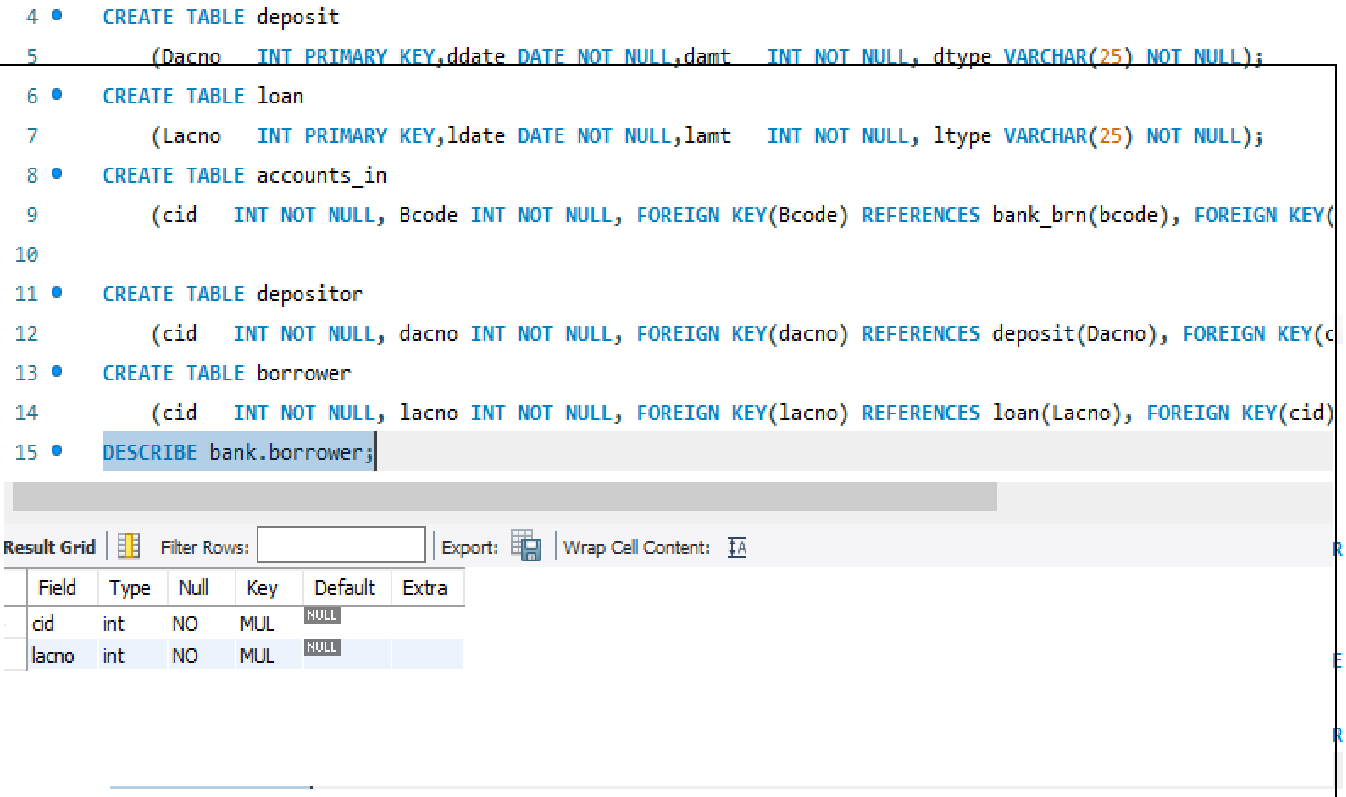
(ii) Include necessary constraints.

Output



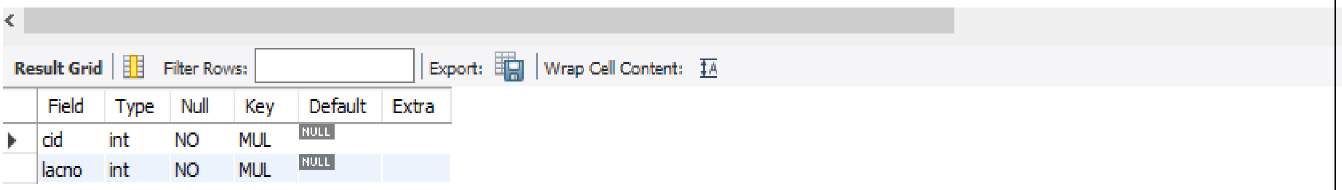
```
CREATE TABLE loan
    (Lacno   INT PRIMARY KEY,ldate DATE NOT NULL,lamt   INT NOT NULL, ltype VARCHAR(25) NOT NULL);
CREATE TABLE accounts_in
    (cid   INT NOT NULL, Bcode INT NOT NULL, FOREIGN KEY(Bcode) REFERENCES bank_brn(bcode), FOREIGN KEY

CREATE TABLE depositor
    (cid   INT NOT NULL, dacno INT NOT NULL, FOREIGN KEY(dacno) REFERENCES deposit(Dacno), FOREIGN KEY(
CREATE TABLE borrower
    (cid   INT NOT NULL, lacno INT NOT NULL, FOREIGN KEY(lacno) REFERENCES loan(Lacno), FOREIGN KEY(cic
DESCRIBE customer;
```

Field	Type	Null	Key	Default	Extra
cid	int	NO	PRI	NULL	
cname	varchar(25)	NO		NULL	
loc	varchar(25)	YES		NULL	
sex	varchar(25)	YES		NULL	
dob	date	YES		NULL	



```
CREATE TABLE deposit
    (Dacno   INT PRIMARY KEY,ddate DATE NOT NULL,damt   INT NOT NULL, dtype VARCHAR(25) NOT NULL);
CREATE TABLE loan
    (Lacno   INT PRIMARY KEY,ldate DATE NOT NULL,lamt   INT NOT NULL, ltype VARCHAR(25) NOT NULL);
CREATE TABLE accounts_in
    (cid   INT NOT NULL, Bcode INT NOT NULL, FOREIGN KEY(Bcode) REFERENCES bank_brn(bcode), FOREIGN KEY(

CREATE TABLE depositor
    (cid   INT NOT NULL, dacno INT NOT NULL, FOREIGN KEY(dacno) REFERENCES deposit(Dacno), FOREIGN KEY(c
CREATE TABLE borrower
    (cid   INT NOT NULL, lacno INT NOT NULL, FOREIGN KEY(lacno) REFERENCES loan(Lacno), FOREIGN KEY(cid)
DESCRIBE bank.borrower;
```

```
Field   Type   Null   Key   Default   Extra
cid     int    NO     MUL   NULL
lacno   int    NO     MUL   NULL
```



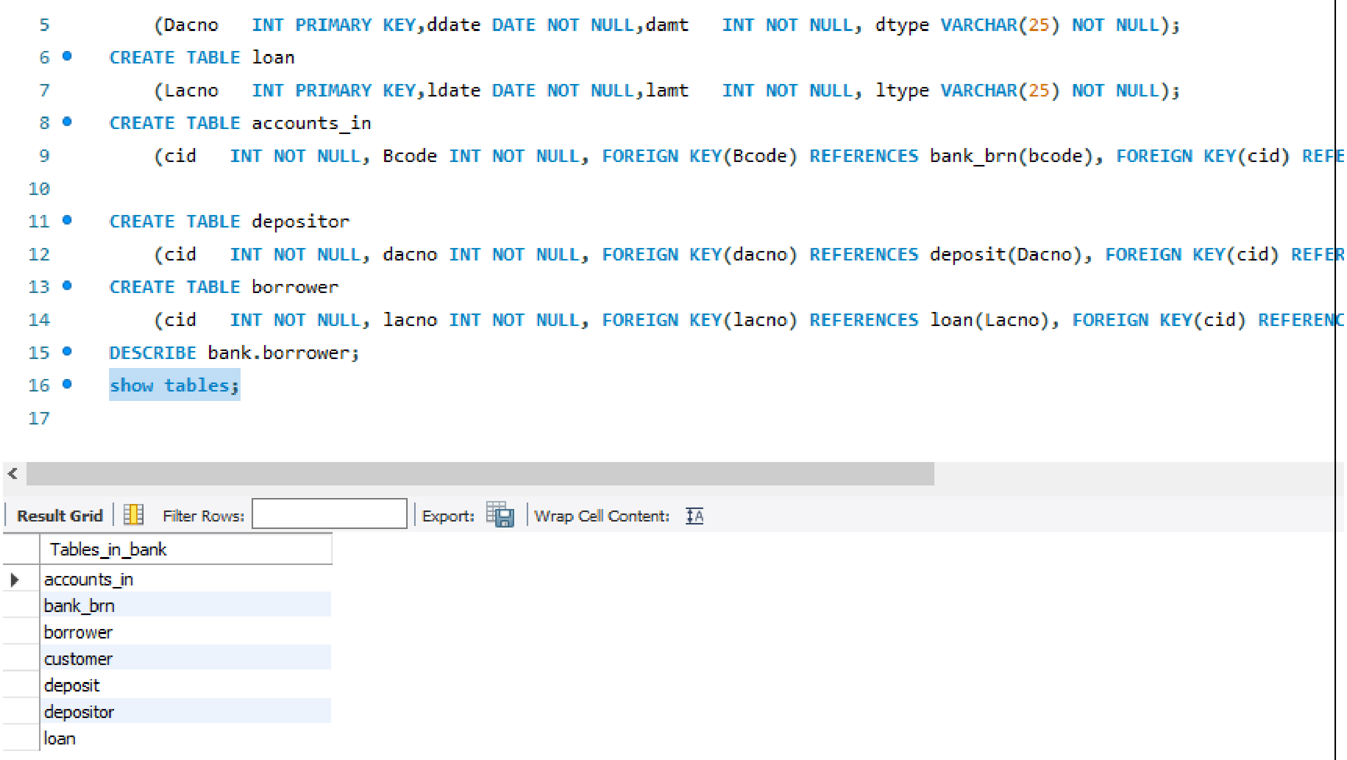
```
Field   Type   Null   Key   Default   Extra
cid     int    NO     MUL   NULL
lacno   int    NO     MUL   NULL
```

(iii)Tables are created under the database 'bank'

output

(iv) Display all the tables in bank database

Output

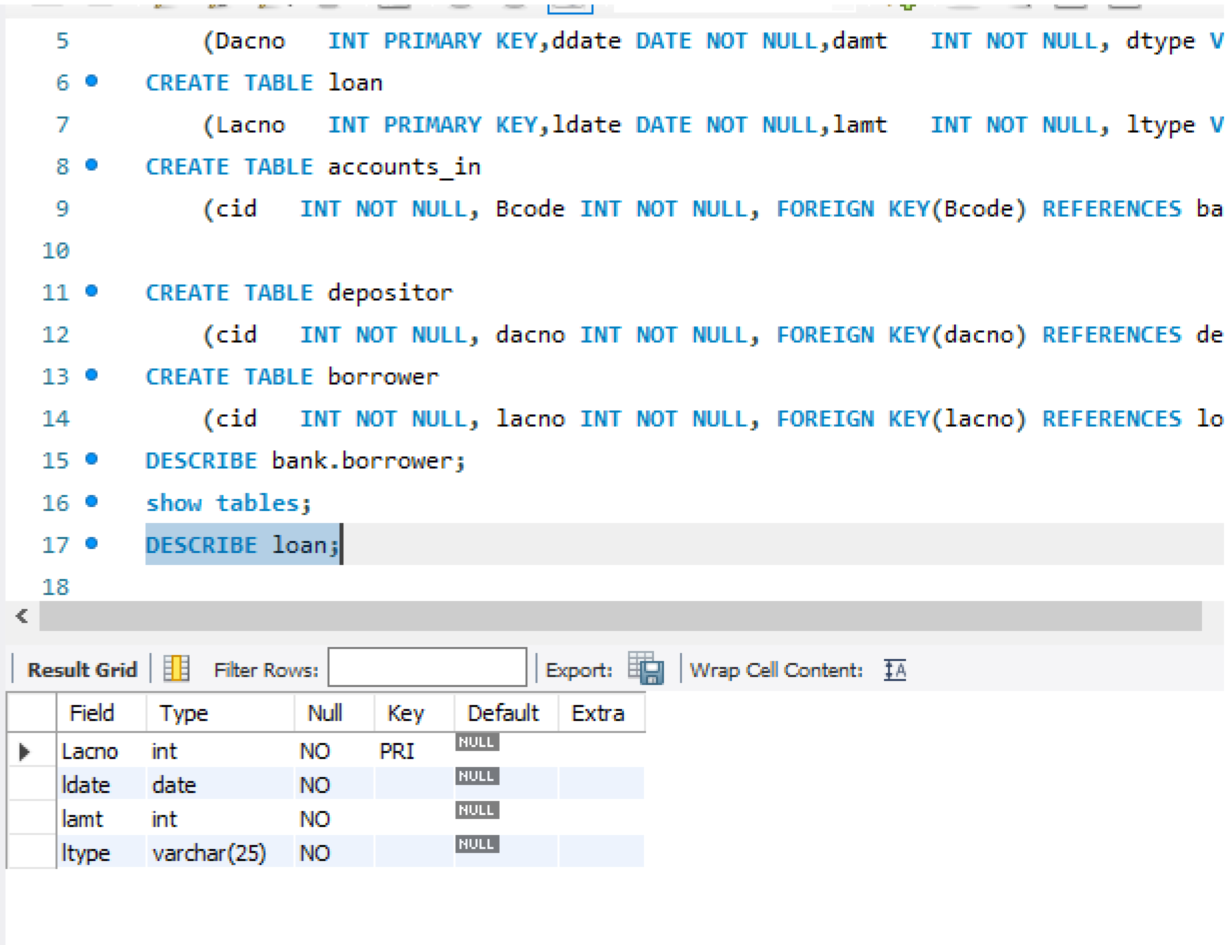


```
    (Dacno   INT PRIMARY KEY,ddate DATE NOT NULL,damt   INT NOT NULL, dtype VARCHAR(25) NOT NULL);
CREATE TABLE loan
    (Lacno   INT PRIMARY KEY,ldate DATE NOT NULL,lamt   INT NOT NULL, ltype VARCHAR(25) NOT NULL);
CREATE TABLE accounts_in
    (cid   INT NOT NULL, Bcode INT NOT NULL, FOREIGN KEY(Bcode) REFERENCES bank_brn(bcode), FOREIGN KEY(cid) REFE

CREATE TABLE depositor
    (cid   INT NOT NULL, dacno INT NOT NULL, FOREIGN KEY(dacno) REFERENCES deposit(Dacno), FOREIGN KEY(cid) REFER
CREATE TABLE borrower
    (cid   INT NOT NULL, lacno INT NOT NULL, FOREIGN KEY(lacno) REFERENCES loan(Lacno), FOREIGN KEY(cid) REFERENC
DESCRIBE bank.borrower;
show tables;
```

```
Tables_in_bank
accounts_in
bank_brn
borrower
customer
deposit
depositor
loan
```

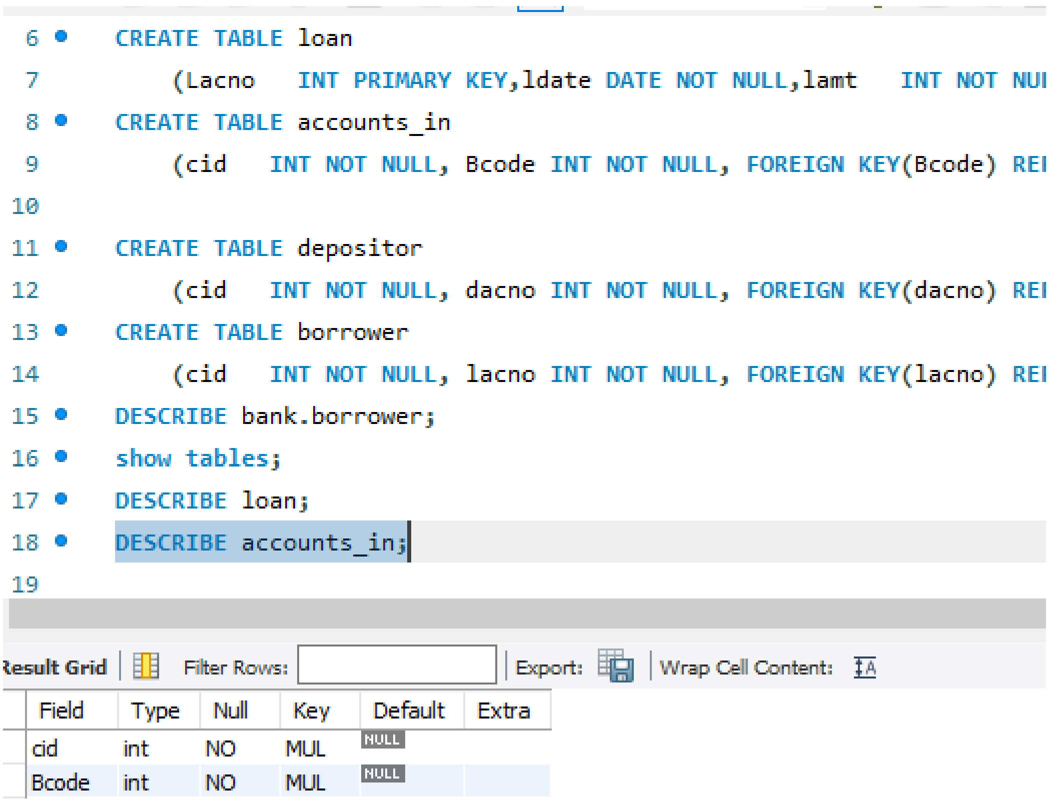
(v) Describe the structure of all tables

Output



```
    (Dacno   INT PRIMARY KEY,ddate DATE NOT NULL,damt   INT NOT NULL, dtype V
CREATE TABLE loan
    (Lacno   INT PRIMARY KEY,ldate DATE NOT NULL,lamt   INT NOT NULL, ltype V
CREATE TABLE accounts_in
    (cid   INT NOT NULL, Bcode INT NOT NULL, FOREIGN KEY(Bcode) REFERENCES ba

CREATE TABLE depositor
    (cid   INT NOT NULL, dacno INT NOT NULL, FOREIGN KEY(dacno) REFERENCES de
CREATE TABLE borrower
    (cid   INT NOT NULL, lacno INT NOT NULL, FOREIGN KEY(lacno) REFERENCES lo
DESCRIBE bank.borrower;
show tables;
DESCRIBE loan;

```

Field	Type	Null	Key	Default	Extra
Lacno	int	NO	PRI	NULL	
ldate	date	NO		NULL	
lamt	int	NO		NULL	
ltype	varchar(25)	NO		NULL	

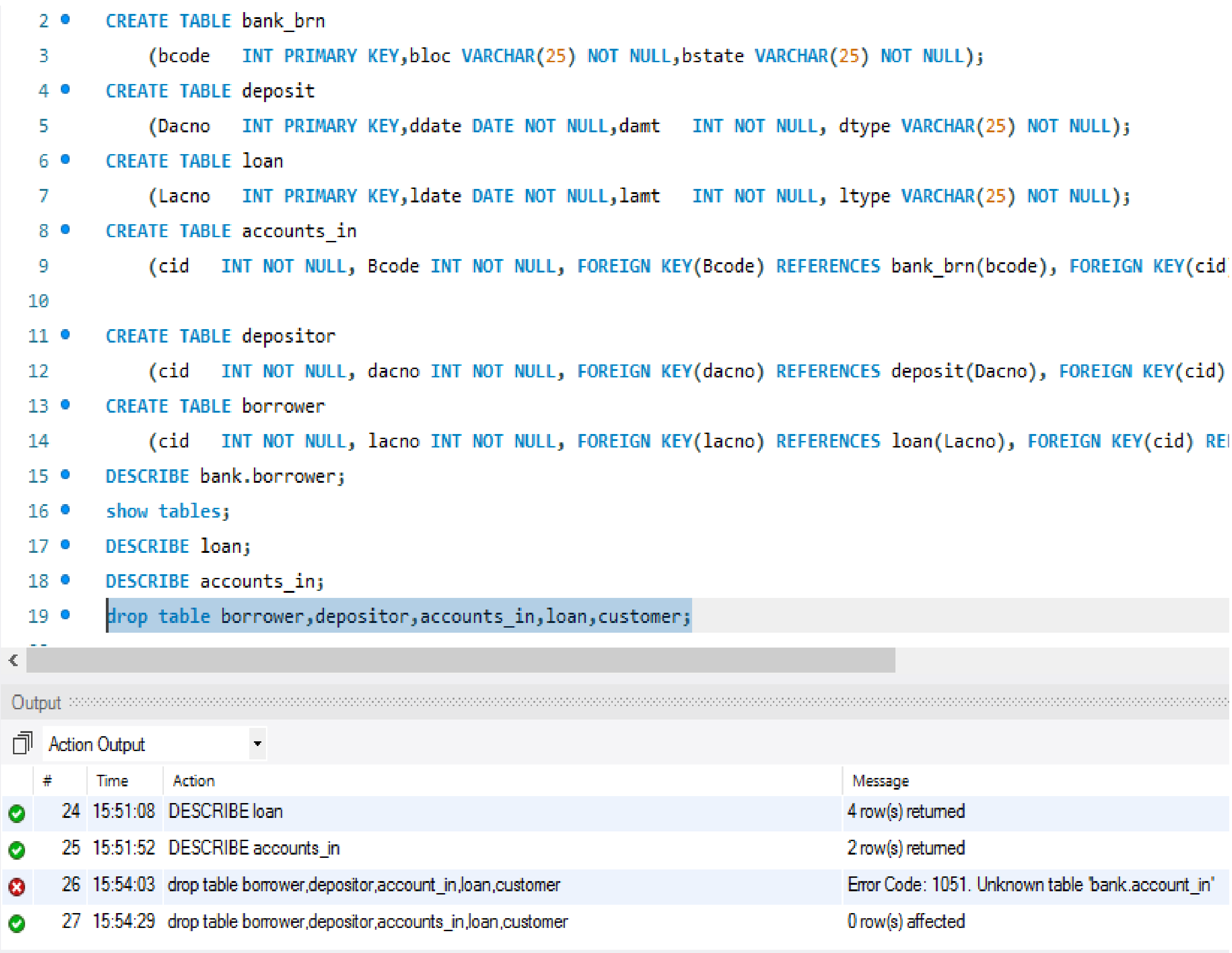


```
CREATE TABLE loan
    (Lacno   INT PRIMARY KEY,ldate DATE NOT NULL,lamt   INT NOT NUI
CREATE TABLE accounts_in
    (cid   INT NOT NULL, Bcode INT NOT NULL, FOREIGN KEY(Bcode) REI

CREATE TABLE depositor
    (cid   INT NOT NULL, dacno INT NOT NULL, FOREIGN KEY(dacno) REI
CREATE TABLE borrower
    (cid   INT NOT NULL, lacno INT NOT NULL, FOREIGN KEY(lacno) REI
DESCRIBE bank.borrower;
show tables;
DESCRIBE loan;
DESCRIBE accounts_in;

```

Field	Type	Null	Key	Default	Extra
cid	int	NO	MUL	NULL	
Bcode	int	NO	MUL	NULL	

(vi) Delete tables.

Output



```
CREATE TABLE bank_brn
    (bcode   INT PRIMARY KEY,bloc VARCHAR(25) NOT NULL,bstate VARCHAR(25) NOT NULL);
CREATE TABLE deposit
    (Dacno   INT PRIMARY KEY,ddate DATE NOT NULL,damt   INT NOT NULL, dtype VARCHAR(25) NOT NULL);
CREATE TABLE loan
    (Lacno   INT PRIMARY KEY,ldate DATE NOT NULL,lamt   INT NOT NULL, ltype VARCHAR(25) NOT NULL);
CREATE TABLE accounts_in
    (cid   INT NOT NULL, Bcode INT NOT NULL, FOREIGN KEY(Bcode) REFERENCES bank_brn(bcode), FOREIGN KEY(cid

CREATE TABLE depositor
    (cid   INT NOT NULL, dacno INT NOT NULL, FOREIGN KEY(dacno) REFERENCES deposit(Dacno), FOREIGN KEY(cid)
CREATE TABLE borrower
    (cid   INT NOT NULL, lacno INT NOT NULL, FOREIGN KEY(lacno) REFERENCES loan(Lacno), FOREIGN KEY(cid) REI
DESCRIBE bank.borrower;
show tables;
DESCRIBE loan;
DESCRIBE accounts_in;
drop table borrower,depositor,accounts_in,loan,customer;
```

```
#   Time      Action                                                    Message
24  15:51:08  DESCRIBE loan                                             4 row(s) returned
25  15:51:52  DESCRIBE accounts_in                                      2 row(s) returned
26  15:54:03  drop table borrower,depositor,account_in,loan,customer   Error Code: 1051. Unknown table 'bank.account_in'
27  15:54:29  drop table borrower,depositor,accounts_in,loan,customer  0 row(s) affected
```

Q.SET 8

Consider the following database for a banking enterprise.

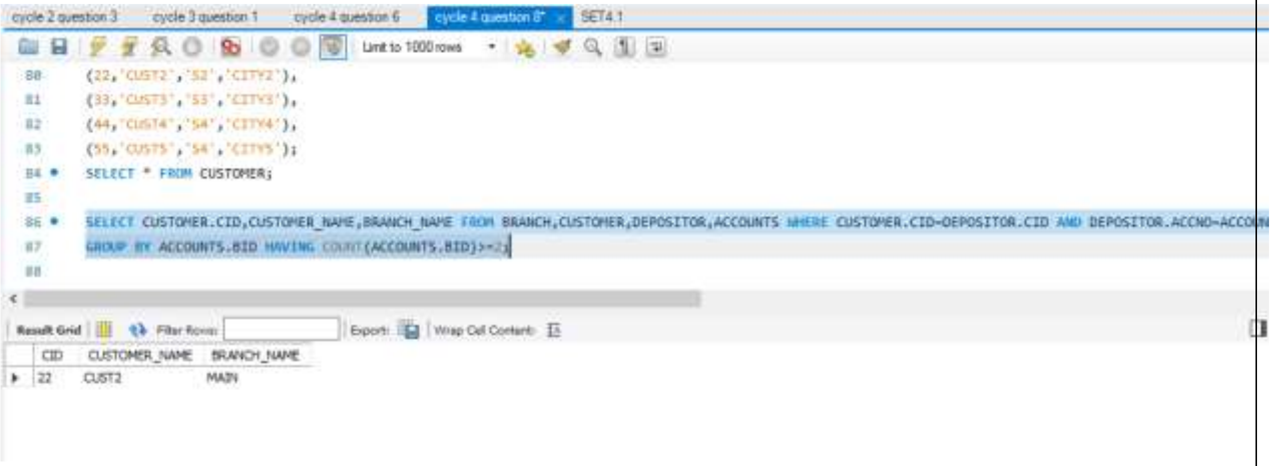
- BRANCH (bid:int, branch-name: String, branch-city: String, assets: int)
- ACCOUNTS (accno: int, bid:int, balance: int)
- DEPOSITOR (cid:int, accno: int)
- CUSTOMER(cid:int, customer-name:String,customer-street:String,customer-city: String)

Set primary key and foreign keys and insert valid records based on questions.

Write SQL queries to

1. Find all the customers who have at least two accounts at the Mainbranch.

Output

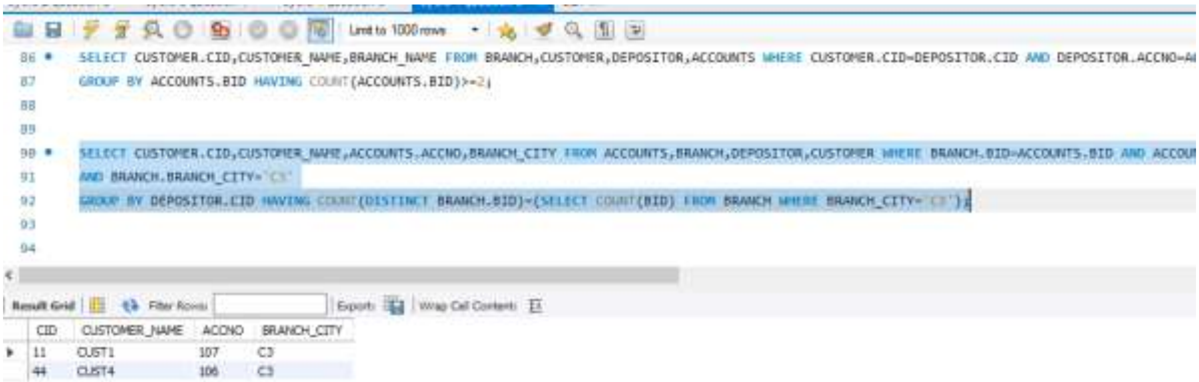


```
(22,'CUST2','S2','CITY2'),
(33,'CUST3','S3','CITY3'),
(44,'CUST4','S4','CITY4'),
(55,'CUST5','S4','CITY5');
SELECT * FROM CUSTOMER;

SELECT CUSTOMER.CID,CUSTOMER_NAME,BRANCH_NAME FROM BRANCH,CUSTOMER,DEPOSITOR,ACCOUNTS WHERE CUSTOMER.CID=DEPOSITOR.CID AND DEPOSITOR.ACCNO=ACCOUN
GROUP BY ACCOUNTS.BID HAVING COUNT(ACCOUNTS.BID)>=2;
```

```
CID  CUSTOMER_NAME  BRANCH_NAME
22   CUST2          MAIN
```

2. Find all the customers who have an account at all the branches located in a specific city.

Output

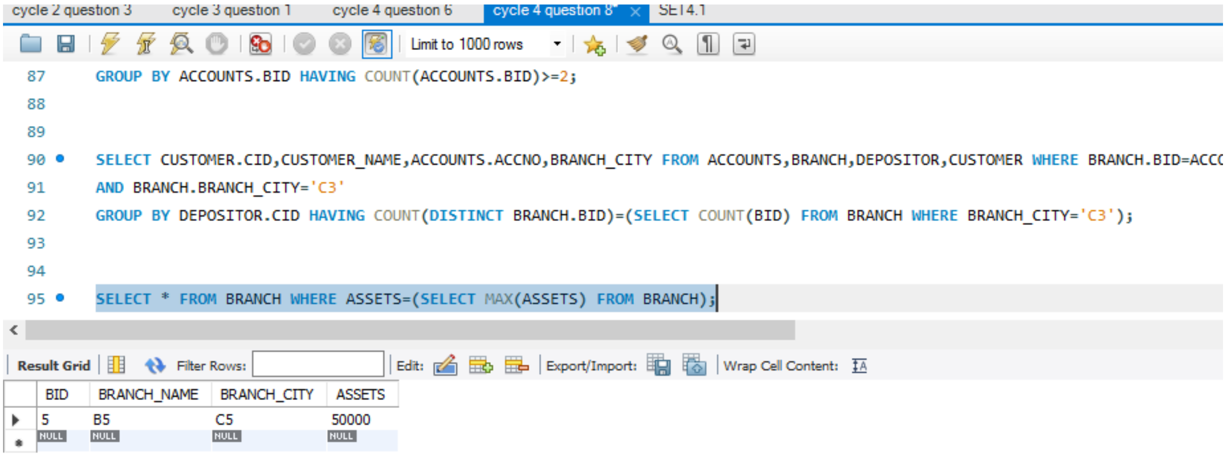


```
SELECT CUSTOMER.CID,CUSTOMER_NAME,BRANCH_NAME FROM BRANCH,CUSTOMER,DEPOSITOR,ACCOUNTS WHERE CUSTOMER.CID=DEPOSITOR.CID AND DEPOSITOR.ACCNO=A
GROUP BY ACCOUNTS.BID HAVING COUNT(ACCOUNTS.BID)>=2;


SELECT CUSTOMER.CID,CUSTOMER_NAME,ACCOUNTS.ACCNO,BRANCH_CITY FROM ACCOUNTS,BRANCH,DEPOSITOR,CUSTOMER WHERE BRANCH.BID=ACCOUNTS.BID AND ACCOU
AND BRANCH.BRANCH_CITY='C3'
GROUP BY DEPOSITOR.CID HAVING COUNT(DISTINCT BRANCH.BID)=(SELECT COUNT(BID) FROM BRANCH WHERE BRANCH_CITY='C3');
```

```
CID  CUSTOMER_NAME  ACCNO  BRANCH_CITY
11   CUST1          107    C3
44   CUST4          106    C3
```

3. Find the branch with greatest asset.

Output

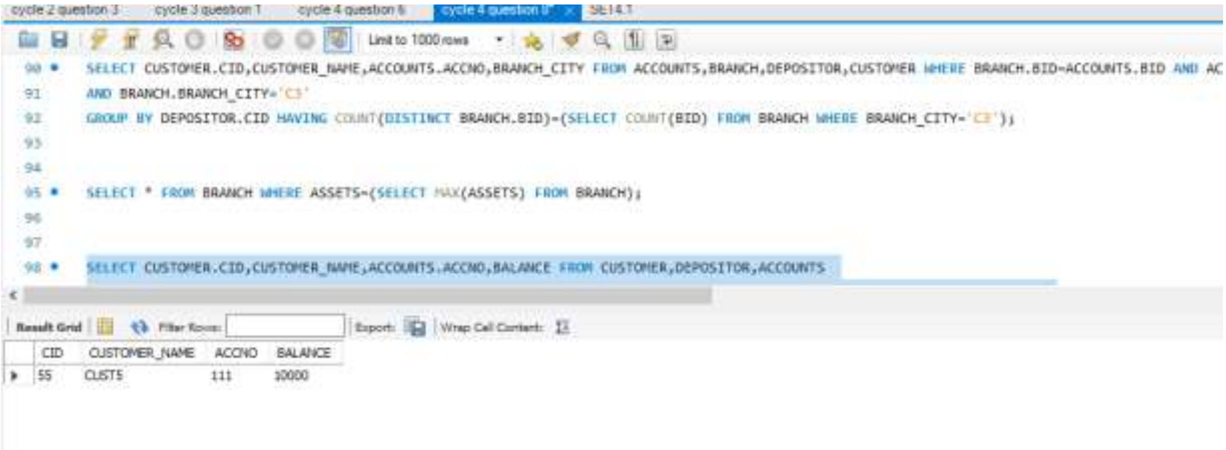


```
GROUP BY ACCOUNTS.BID HAVING COUNT(ACCOUNTS.BID)>=2;


SELECT CUSTOMER.CID,CUSTOMER_NAME,ACCOUNTS.ACCNO,BRANCH_CITY FROM ACCOUNTS,BRANCH,DEPOSITOR,CUSTOMER WHERE BRANCH.BID=ACCC
AND BRANCH.BRANCH_CITY='C3'
GROUP BY DEPOSITOR.CID HAVING COUNT(DISTINCT BRANCH.BID)=(SELECT COUNT(BID) FROM BRANCH WHERE BRANCH_CITY='C3');


SELECT * FROM BRANCH WHERE ASSETS=(SELECT MAX(ASSETS) FROM BRANCH);
```

```
BID   BRANCH_NAME  BRANCH_CITY  ASSETS
5     B5           C5           50000
NULL  NULL         NULL         NULL
```

4. Find the customer with highest balance.

Output



```
SELECT CUSTOMER.CID,CUSTOMER_NAME,ACCOUNTS.ACCNO,BRANCH_CITY FROM ACCOUNTS,BRANCH,DEPOSITOR,CUSTOMER WHERE BRANCH.BID=ACCOUNTS.BID AND AC
AND BRANCH.BRANCH_CITY='C3'
GROUP BY DEPOSITOR.CID HAVING COUNT(DISTINCT BRANCH.BID)=(SELECT COUNT(BID) FROM BRANCH WHERE BRANCH_CITY='C3');


SELECT * FROM BRANCH WHERE ASSETS=(SELECT MAX(ASSETS) FROM BRANCH);


SELECT CUSTOMER.CID,CUSTOMER_NAME,ACCOUNTS.ACCNO,BALANCE FROM CUSTOMER,DEPOSITOR,ACCOUNTS
```

CID	CUSTOMER_NAME	ACCNO	BALANCE
55	CUST5	111	10000