**LAB CYCLE 1**

**QUESTION SET 1**

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

empno                  NUMBER(2)

ename                   VARCHAR2(25)

job                     VARCHAR2(12)

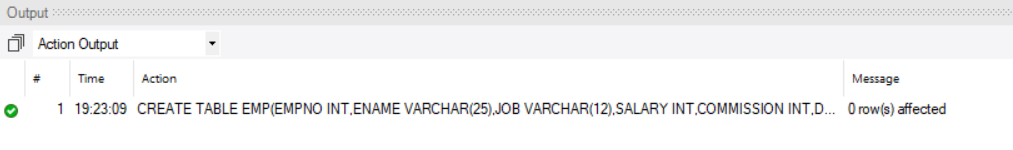
salary                   NUMBER(10,2)

commission              NUMBER(7,2)

deptno                 NUMBER(2)

**Query:** CREATE TABLE EMP(EMPNO INT,ENAME VARCHAR(25),JOB VARCHAR(12),SALARY INT,COMMISSION INT,DEPTNO INT);

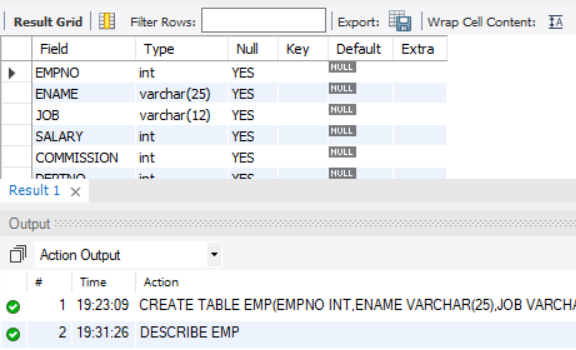
**Output:**



2. Display the structure of ‘EMP’

**Query:** DESCRIBE EMP;

**Output:**



3. Insert the following record into ‘EMP’

**EMPNO ENAME      JOB             SAL       COMM     DEPTNO**

7369 SMITH      CLERK           800                    20

**Query:** INSERT INTO EMP(EMPNO,ENAME,JOB,SALARY,DEPTNO)

VALUES(7369,'SMITH','CLERK',800,20);

**Output:**



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

1. FORD     ANALYST         3000                   20

7934 MILLER     CLERK           1300                   10

**Query:** INSERT INTO EMP(EMPNO,ENAME,JOB,SALARY,COMMISSION,DEPTNO)

VALUES(7499,'ALLEN','SALESMAN',1600,300,30),

(7521,'WARD','SALESMAN',1250,500,30),

(7654,'MARTIN','SALESMAN',1250,1400,30);

INSERT INTO EMP(EMPNO,ENAME,JOB,SALARY,DEPTNO)

VALUES(7566,'JONES','MANAGER',2975,20),

(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:**



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

**Query:** UPDATE EMP SET JOB='CLERK' WHERE JOB='NULL';

**Output:**



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

**Query:** ALTER TABLE EMP ADD DATE\_JOIN VARCHAR(50);

UPDATE EMP SET DATE\_JOIN='17-DEC-80' WHERE EMPNO=7369;

UPDATE EMP SET DATE\_JOIN='20-FEB-81' WHERE EMPNO=7499;

UPDATE EMP SET DATE\_JOIN='22-FEB-81' WHERE EMPNO=7521;

UPDATE EMP SET DATE\_JOIN='02-APR-81' WHERE EMPNO=7566;

UPDATE EMP SET DATE\_JOIN='28-SEP-81' WHERE EMPNO=7654;

UPDATE EMP SET DATE\_JOIN='01-MAY-81' WHERE EMPNO=7698;

UPDATE EMP SET DATE\_JOIN='09-JUN-81' WHERE EMPNO=7782;

UPDATE EMP SET DATE\_JOIN='19-APR-87' WHERE EMPNO=7788;

UPDATE EMP SET DATE\_JOIN='17-NOV-81' WHERE EMPNO=7839;

UPDATE EMP SET DATE\_JOIN='08-SEP-81' WHERE EMPNO=7844;

UPDATE EMP SET DATE\_JOIN='23-MAY-87' WHERE EMPNO=7876;

UPDATE EMP SET DATE\_JOIN='03-DEC-81' WHERE EMPNO=7900;

UPDATE EMP SET DATE\_JOIN='03-DEC-81' WHERE EMPNO=7902;

UPDATE EMP SET DATE\_JOIN='23-JAN-82' WHERE EMPNO=7934;

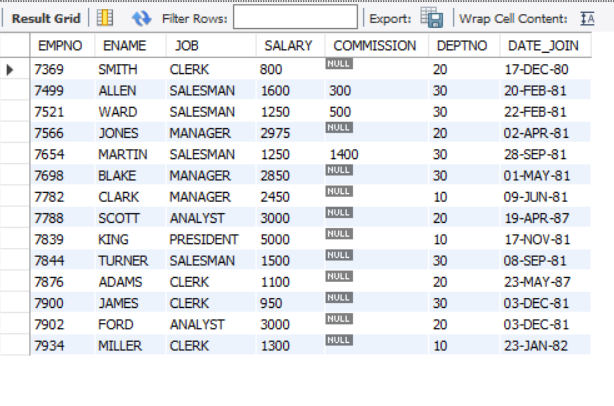
**Output:**



7. Display details of all employees.

**Query:** SELECT \* FROM EMP ORDER BY EMPNO;

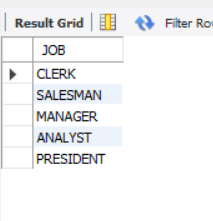
**Output:**



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

**Query:** SELECT DISTINCT JOB FROM EMP;

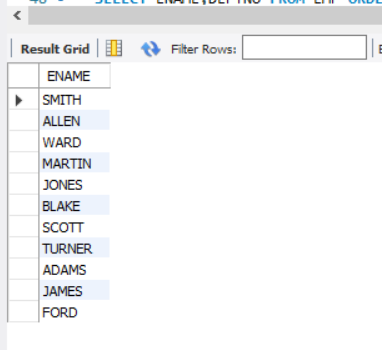
**Output:**



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

**Query:** SELECT ENAME FROM EMP WHERE DEPTNO IN(20,30);

**Output:**

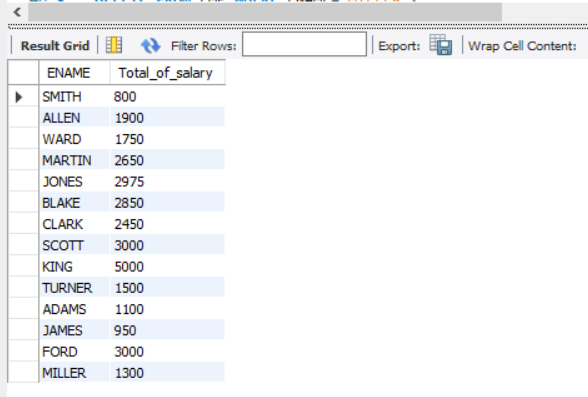


10. List name and Total of salary i.e sal+commission

**Query:** SELECT ENAME,SALARY+IFNULL(COMMISSION,0)

AS Total\_of\_salary FROM EMP;

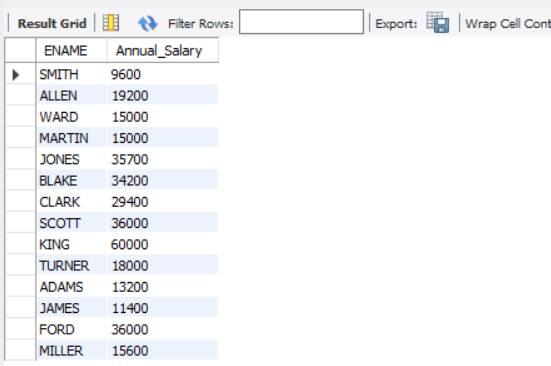
**Output:**



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

**Query:** SELECT ENAME,SALARY\*12 AS Annual\_Salary  FROM EMP;

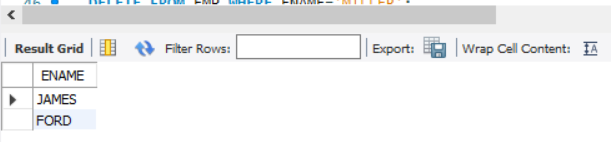
**Output:**



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

**Query:** SELECT ENAME FROM EMP WHERE DATE\_JOIN='03-DEC-81';

**Output:**



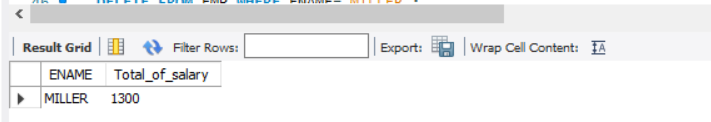
13. Display the total salary of ‘Miller’

**Query:** SELECT ENAME,SALARY+IFNULL(COMMISSION,0)

AS Total\_of\_salary FROM EMP

WHERE ENAME='MILLER';

**Output:**



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

**Query:** DELETE FROM EMP WHERE ENAME='MILLER';

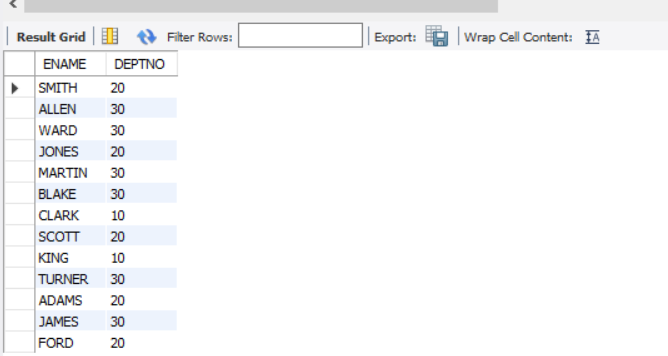
**Output:**



15. Display name and deptno of all employees.

**Query:** SELECT ENAME,DEPTNO FROM EMP ORDER BY EMPNO;

**Output:**



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

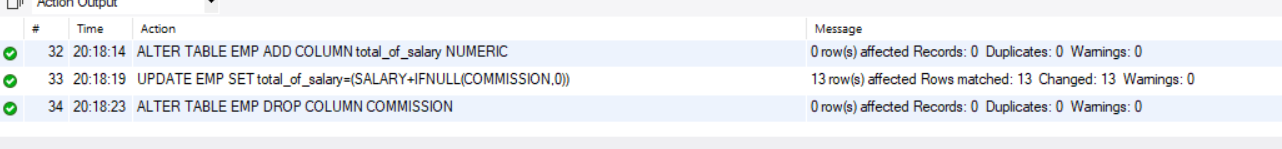
i.e sal+commission

**Query:** ALTER TABLE EMP ADD COLUMN total\_of\_salary NUMERIC;

UPDATE EMP SET total\_of\_salary=(SALARY+IFNULL(COMMISSION,0));

ALTER TABLE EMP DROP COLUMN COMMISSION;

**Output:**



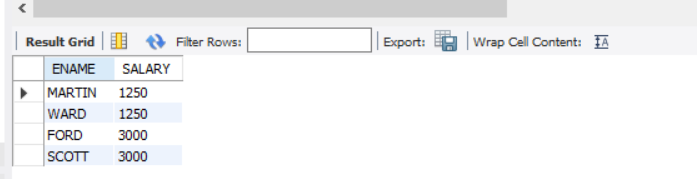
17. Display the name of employees having the same amount of salary **( don’t use subqueries**)

**Query:** SELECT E.ENAME,E.SALARY

FROM EMP E INNER JOIN EMP B ON E.SALARY=B.SALARY

AND E.ENAME <> B.ENAME;

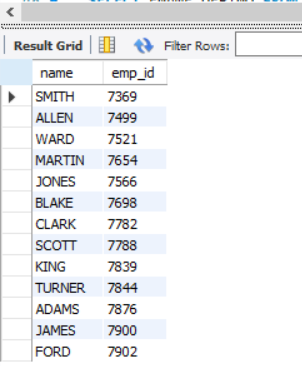
**Output:**



18. Display the name and employee no as ‘name’ and ‘emp\_id’

**Query:** SELECT ENAME AS 'name',EMPNO AS 'emp\_id' FROM EMP;

**Output:**



19. Rename table ‘EMP’ to ‘EMPLOYEE’

**Query:** RENAME TABLE EMP TO EMPLOYEE;

**Output:**

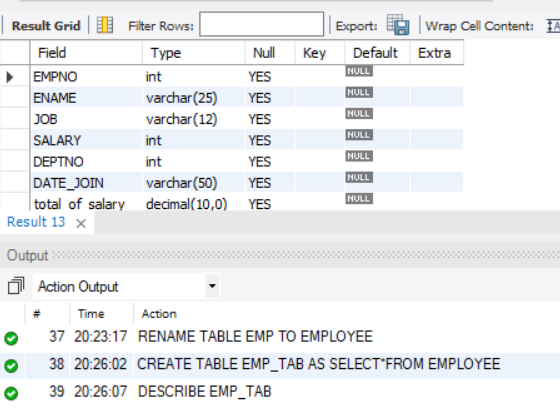


20. Create a new table ‘EMP\_TAB’ from table ‘EMPLOYEE’

**Query:** CREATE TABLE EMP\_TAB AS SELECT\*FROM EMPLOYEE;

DESCRIBE EMP\_TAB;

**Output:**



21. List all the details of ‘EMPLOYEE’ and ‘EMP\_TAB’

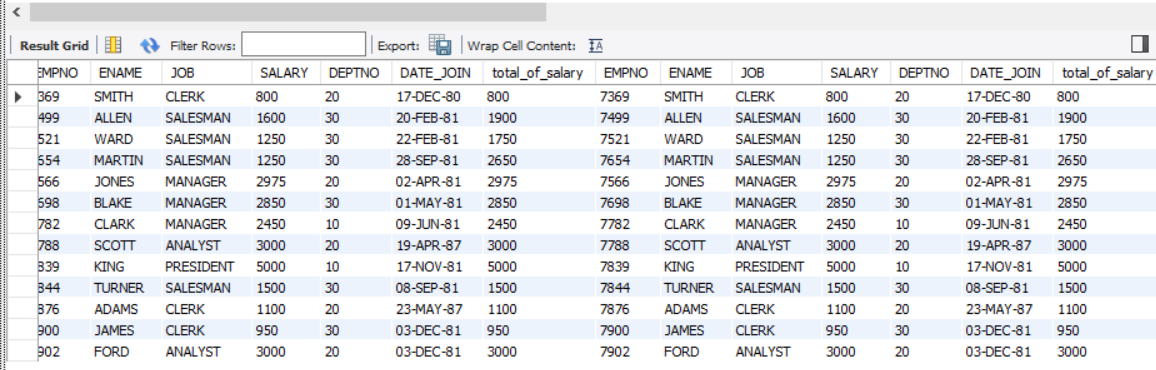
**Query:** SELECT DISTINCT E.EMPNO,E.ENAME,E.JOB,E.SALARY,E.DEPTNO,E.DATE\_JOIN,E.total\_of\_salary,

B.EMPNO,B.ENAME,B.JOB,B.SALARY,B.DEPTNO,B.DATE\_JOIN,B.total\_of\_salary

FROM EMP\_TAB E JOIN EMPLOYEE B

WHERE E.EMPNO=B.EMPNO;

**Output:**



22. Delete all records from ‘EMP’

**Query:** TRUNCATE TABLE EMPLOYEE;

**Output:**



23. Delete the table ‘EMP’

**Query:** DROP TABLE EMPLOYEE;

**Output:**



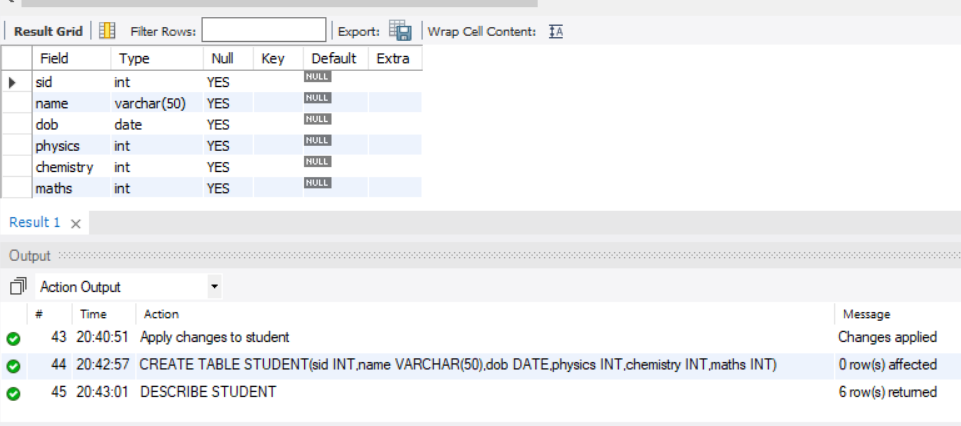
**QUESTION SET 2**

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:

**Query:** CREATE TABLE STUDENT(sid INT,name VARCHAR(50),dob DATE,

physics INT,chemistry INT,maths INT);

**Output:**



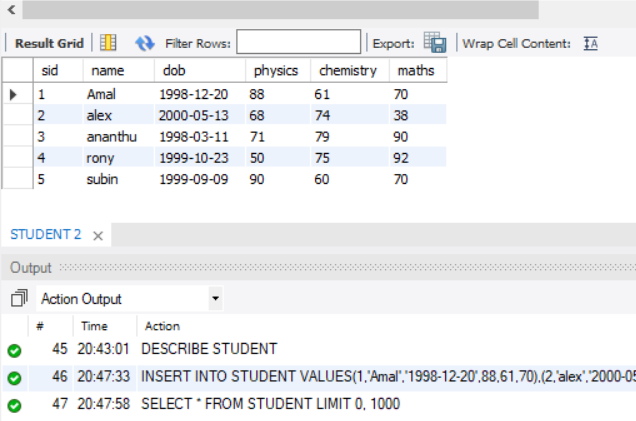
**Query:** INSERT INTO STUDENT

VALUES(1,'Amal','1998-12-20',88,61,70),(2,'alex','2000-05-13',68,74,38),

(3,'ananthu','1998-03-11',71,79,90),(4,'rony','1999-10-23',50,75,92),(5,'subin','1999-09-09',90,60,70);

SELECT \* FROM STUDENT;

**Output:**



1. Display the id and name of youngest student.

**Query:** SELECT sid,name FROM STUDENT WHERE dob=(SELECT MAX(dob)FROM STUDENT);

**Output:**

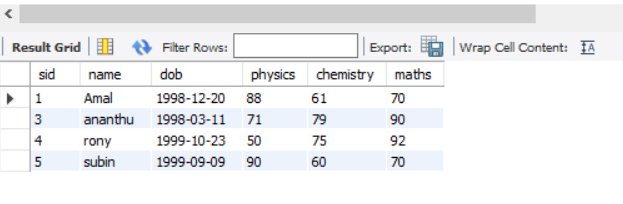


2. Display the details of students who have passed in maths and either in physics or

chemistry.(pass mark = 40 marks and above)

**Query:** SELECT \* FROM STUDENT WHERE maths >=40 and (physics >=40 or chemistry >=40);

**Output:**



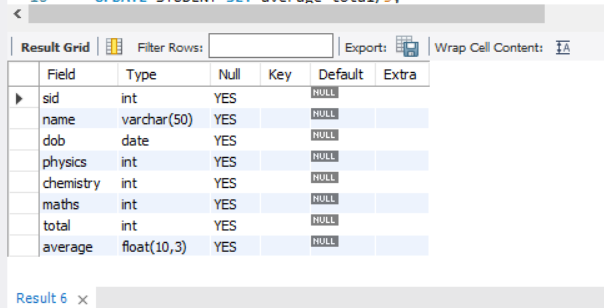
3. Add two more columns total and average.

**Query:** ALTER TABLE STUDENT add total int;

ALTER TABLE STUDENT add average float(10,3);

DESCRIBE STUDENT;

**Output:**

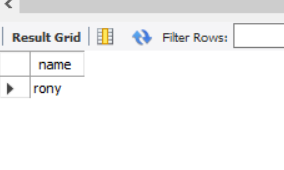


4. Display the name of student who scored highest marks in maths.

**Query:** SELECT name from STUDENT

WHERE maths=(select max(maths) from STUDENT);

**Output:**

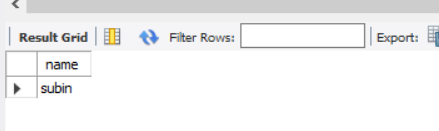


5. Display the name of student who scored least marks in chemistry.

**Query:** SELECT name from STUDENT

WHERE chemistry=(select min(chemistry) from STUDENT);

**Output:**



6. Update column total with total marks.

**Query:** UPDATE STUDENT set total=physics+chemistry+maths;

**Output:**

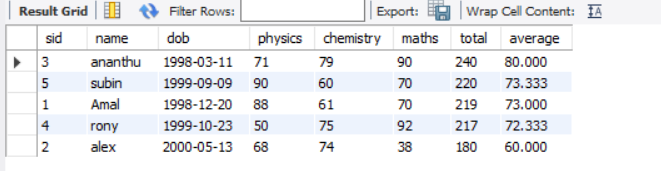


7. Display details of students in order of total merit.

**Query:** UPDATE STUDENT SET average=total/3;

SELECT \* FROM STUDENT ORDER BY total DESC;

**Output:**



8. Rename the column average with avg\_mark

**Query:** ALTER TABLE STUDENT RENAME COLUMN average TO avg\_mark;

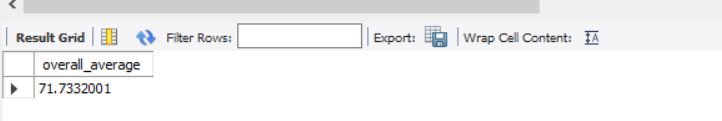
**Output:**



9. Find out the overall average of class.

**Query:** SELECT AVG(avg\_mark) AS overall\_average FROM STUDENT;

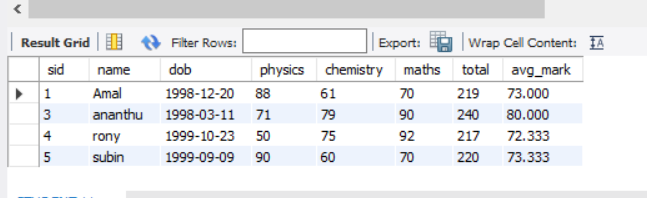
**Output:**



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

**Query:** SELECT\*FROM STUDENT WHERE avg\_mark >(SELECT AVG(avg\_mark)AS overall\_average FROM STUDENT);

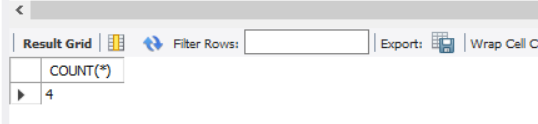
**Output:**



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

**Query:** SELECT COUNT(\*)FROM STUDENT WHERE avg\_mark > (SELECT AVG(avg\_mark)AS overall\_average FROM STUDENT);

**Output:**



**QUESTION SET 3**

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 |  |

**Query:** CREATE TABLE LOAN\_ACCOUNTS(ACCNO CHAR(4),

CUST\_NAME VARCHAR(15),

LOAN\_AMOUNT decimal(10,2),

INSTALLMENT int,

INT\_RATE decimal(10,2),

START\_DATE date,

INTEREST decimal(10,2));

DESCRIBE LOAN\_ACCOUNTS;

ALTER TABLE LOAN\_ACCOUNTS

ADD COLUMN CATEGORY VARCHAR(1);

INSERT INTO LOAN\_ACCOUNTS(ACCNO,CUST\_NAME,LOAN\_AMOUNT,INSTALLMENT,INT\_RATE,START\_DATE)

VALUES(1001,'R.K GUPTA',300000.00,36,12.00,'2009-07-19'),

(1002,'S.P SHARMA',500000.00,48,10.00,'2008-03-22'),

(1003,'K.P JAIN',300000.00,36,NULL,'2007-08-03'),

(1004,'M.P YADAV',800000.00,60,10.00,'2008-06-12'),

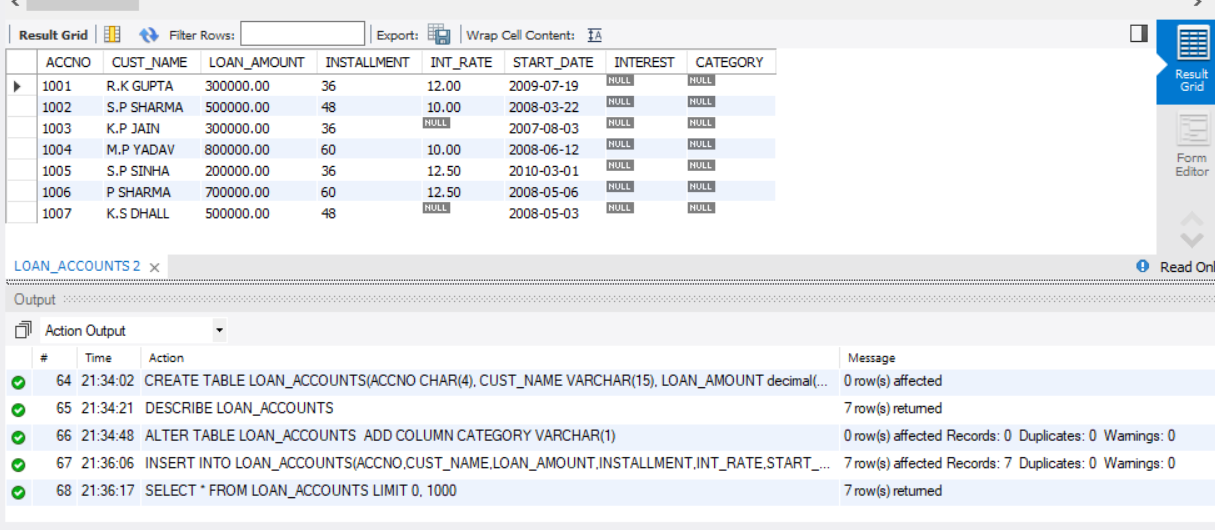
(1005,'S.P SINHA',200000.00,36,12.50,'2010-03-01'),

(1006,'P SHARMA',700000.00,60,12.50,'2008-05-06'),

(1007,'K.S DHALL',500000.00,48,NULL,'2008-05-03');

SELECT \* FROM LOAN\_ACCOUNTS;

**Output:**



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

**Query:** update LOAN\_ACCOUNTS

SET INT\_RATE=11.50

WHERE INT\_RATE is null;

**Output:**



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

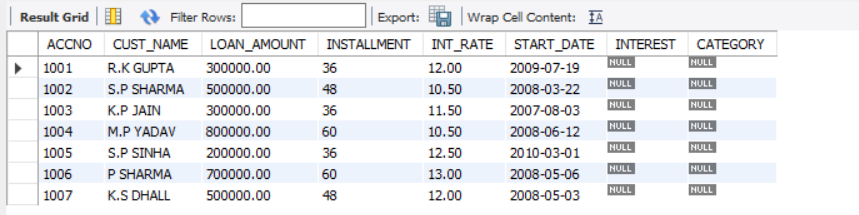
**Query:** Update Loan\_Accounts

SET INT\_RATE=INT\_RATE + 0.5

WHERE Loan\_Amount > 400000;

SELECT \* FROM LOAN\_ACCOUNTS;

**Output:**



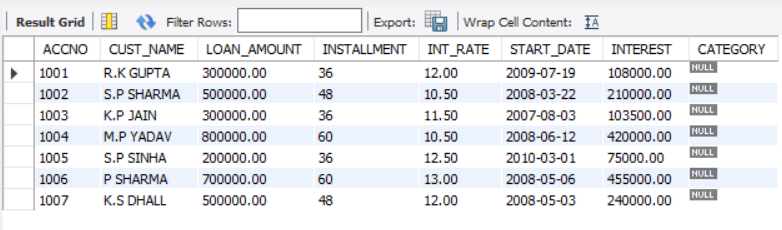
3. For each Loan replace Interest with (Loan\_amount \* Int\_rate\* installments)/(12\*100).

**Query:** Update Loan\_Accounts

SET INTEREST=(Loan\_amount \* Int\_rate\* installment)/(12\*100) ;

SELECT \* FROM LOAN\_ACCOUNTS;

**Output:**



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

**Query:** delete from LOAN\_ACCOUNTS

where START\_DATE < '2008-01-01';

**Output:**



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

**Query:** DELETE FROM LOAN\_ACCOUNTS

WHERE CUST\_NAME like 'K%';

**Output:**

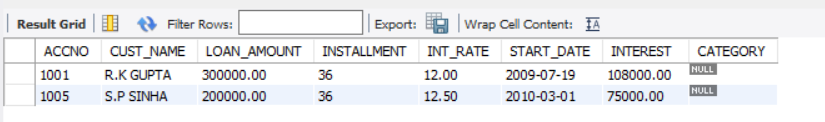


6. Display the details of all the Loans with less than 40 installments.

**Query:** SELECT \* FROM LOAN\_ACCOUNTS

where INSTALLMENT < 40;

**Output:**



7. Display the Accno and Loan\_amount of all the loans started before 01-04-2009.

**Query:** SELECT ACCNO, Loan\_amount FROM LOAN\_ACCOUNTS

where START\_DATE < '2009-04-01';

**Output:**

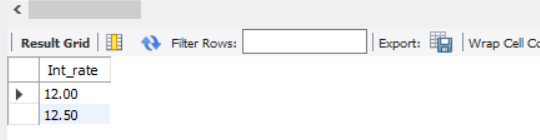


8. Display the int\_rate of all Loans started after 01-04-2009.

**Query:** SELECT Int\_rate FROM LOAN\_ACCOUNTS

WHERE START\_DATE > '2009-04-01';

**Output:**



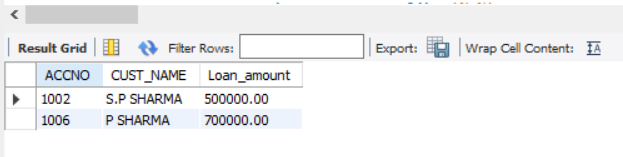
9. Display the Accno, cust\_name and Loan amount for all the Loans for which the cust\_name ends with‘Sharma’.

**Query:** SELECT ACCNO, CUST\_NAME, Loan\_amount

FROM LOAN\_ACCOUNTS

WHERE CUST\_NAME like '%SHARMA';

**Output:**

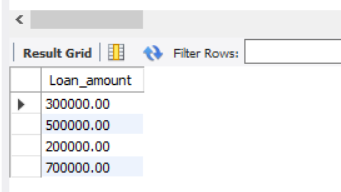


10. Loan\_Amount of all the Loans for which the Cust\_name ends with ‘a’.

**Query:** SELECT Loan\_amount FROM LOAN\_ACCOUNTS

WHERE CUST\_NAME like '%a';

**Output:**

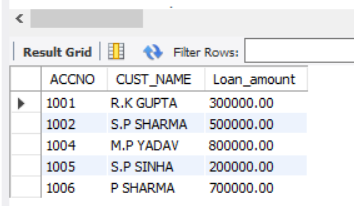


11. Display the Accno, Cust\_name and Loan\_Amount for the Loans for which the Cust\_name contains ‘a’.

**Query:** SELECT ACCNO, CUST\_NAME, Loan\_amount

FROM LOAN\_ACCOUNTS where CUST\_NAME like '%a%';

**Output:**



12. Display the Accno, Cust\_name and Loan\_Amount for all the Loans for which the Cust\_name does not contain ‘P’.

**Query:** SELECT ACCNO, CUST\_NAME, Loan\_amount

FROM LOAN\_ACCOUNTS where NOT(CUST\_NAME like '%P%');

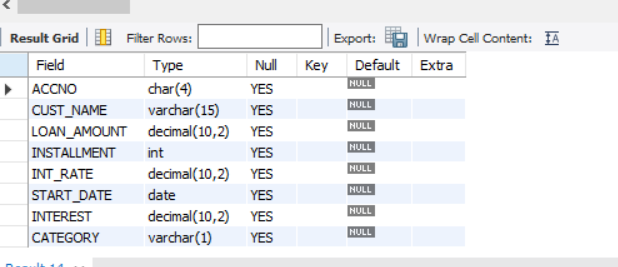
**Output:**



13. Display the structure of table LOAN\_ACCOUNTS so that you can verify that the table is created as required.

**Query:** DESCRIBE LOAN\_ACCOUNTS;

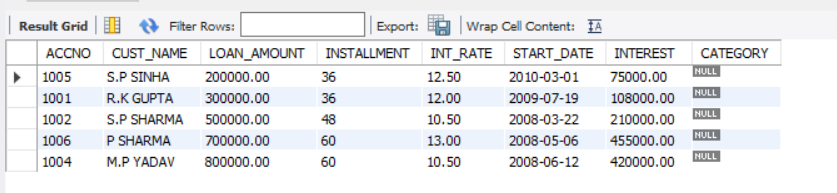
**Output:**



14. Display the details of all the loans in the ascending order of their Loan\_Amount.

**Query:** SELECT \* FROM LOAN\_ACCOUNTS order by Loan\_amount asc;

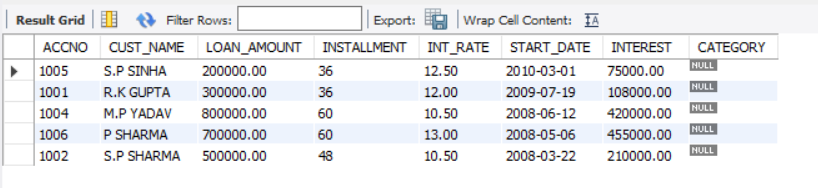
**Output:**



15. Display the details of all the Loans in the descending order of their Start\_date.

**Query:** SELECT \* FROM LOAN\_ACCOUNTS order by START\_DATE desc;

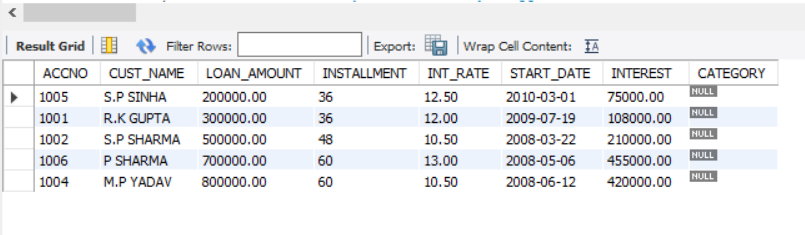
**Output:**



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.

**Query:** SELECT \* FROM LOAN\_ACCOUNTS order by Loan\_amount, START\_DATE desc;

**Output:**



17.Display the Accno, Cust\_name and Loan\_Amount of all the Loans for which the Cust\_name starts with ‘K’.

**Query:** SELECT \* FROM LOAN\_ACCOUNTS where CUST\_NAME like 'K%';

**Output:**



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

**Query:** SELECT \* FROM LOAN\_ACCOUNTS where INT\_RATE is null;

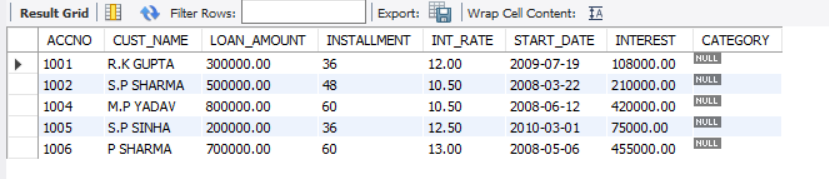
**Output:**



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

**Query:** SELECT \* FROM LOAN\_ACCOUNTS where INT\_RATE is not null;

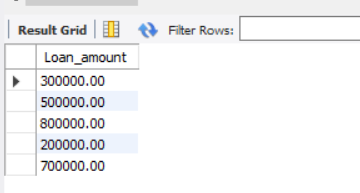
**Output:**



20. Display the amounts of various loans from the table Loan\_Accounts. A Loan\_Amount should appear only once.

**Query:** SELECT distinct Loan\_amount FROM LOAN\_ACCOUNTS ;

**Output:**



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

**Query:** SELECT \* FROM LOAN\_ACCOUNTS where START\_DATE >'2008-12-31' and INSTALLMENT >36;

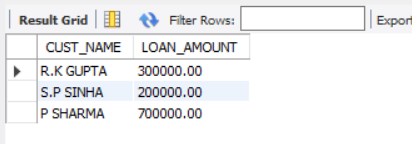
**Output:**



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.

**Query:** SELECT CUST\_NAME,LOAN\_AMOUNT FROM LOAN\_ACCOUNTS WHERE LOAN\_AMOUNT <500000 or INT\_RATE >12;

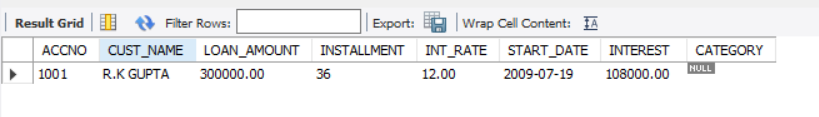
**Output:**



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

**Query:** SELECT \* FROM LOAN\_ACCOUNTS where year(START\_DATE)='2009';

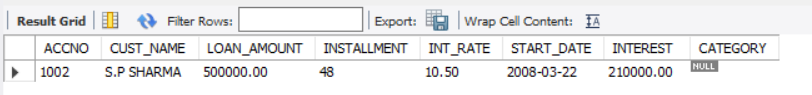
**Output:**



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

**Query:** SELECT \* FROM LOAN\_ACCOUNTS where LOAN\_AMOUNT BETWEEN 400000 and 500000;

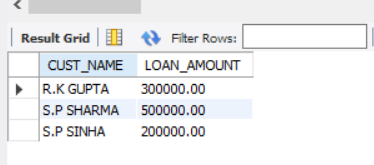
**Output:**



25. Display the Customer\_name and Loan\_amount of all the Loans for which the number of installments are 26, 36 and 48.

**Query:** SELECT CUST\_NAME,LOAN\_AMOUNT FROM LOAN\_ACCOUNTS where INSTALLMENT IN(26,36,48);

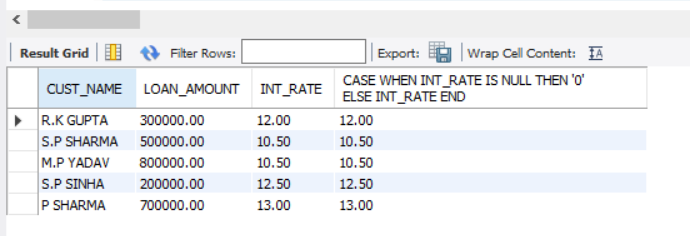
**Output:**



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

**Query:** SELECT CUST\_NAME,LOAN\_AMOUNT,INT\_RATE,CASE WHEN INT\_RATE IS NULL THEN '0' ELSE INT\_RATE END FROM LOAN\_ACCOUNTS;

**Output:**



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

**Query:** SELECT CUST\_NAME,LOAN\_AMOUNT,INT\_RATE,CASE WHEN INT\_RATE IS NULL THEN 'No Interest' ELSE INT\_RATE END FROM LOAN\_ACCOUNTS;

**Output:**

