1. Create a student table with the student id, name, and marks as attributes where the student id is the primary key.

CREATE DATABASE STUDENTS;

USE STUDENTS;

CREATE TABLE STUDENT1(

ID INT(3),NAME CHAR(50),MARKS FLOAT(10,2),

CONSTRAINT PK PRIMARY KEY(ID)

);

Answer:

|  |  |  |
| --- | --- | --- |
| ID | NAME | MARKS |

2. Insert the details of a new student in the above table

INSERT INTO STUDENT1 VALUES(1,"AKASH",95);

INSERT INTO STUDENT1 VALUES(2,"BHARATH",92.5);

INSERT INTO STUDENT1 VALUES(3,"DEV",97);

Answer:

|  |  |  |
| --- | --- | --- |
| ID | NAME | MARKS |
| 1 | AKASH | 95.00 |
| 2 | BHARATH | 92.50 |
| 3 | DEV | 97.00 |

3. Delete the detail of a student in the above table.

TRUNCATE TABLE STUDENT1;

Answer:

|  |  |  |
| --- | --- | --- |
| ID | NAME | MARKS |

4. Insert more records to the above student table more than 7.

INSERT INTO STUDENT1 VALUES(1,"DEV",97);

INSERT INTO STUDENT1 VALUES(2,"ARVIND",90.5);

INSERT INTO STUDENT1 VALUES(3,"DINESH",85);

INSERT INTO STUDENT1 VALUES(4,"GOKUL",75);

INSERT INTO STUDENT1 VALUES(5,"VIGNESH",80);

INSERT INTO STUDENT1 VALUES(6,"VIJAY",60.5);

INSERT INTO STUDENT1 VALUES(7,"GOPI",40);

INSERT INTO STUDENT1 VALUES(8,"KARTHICK",84);

INSERT INTO STUDENT1 VALUES(9,"MANI",45);

INSERT INTO STUDENT1 VALUES(10,"NAVEEN",55);

INSERT INTO STUDENT1 VALUES(11,"PRAVEEN",50);

Answer:

|  |  |  |
| --- | --- | --- |
| ID | NAME | MARKS |
| 1 | DEV | 97.00 |
| 2 | ARVIND | 90.50 |
| 3 | DINESH | 85.00 |
| 4 | GOKUL | 75.00 |
| 5 | VIGNESH | 80.00 |
| 6 | VIJAY | 60.50 |
| 7 | GOPI | 40.00 |
| 8 | KARTHICK | 84.00 |
| 9 | MANI | 45.00 |
| 10 | NAVEEN | 55.00 |
| 11 | PRAVEEN | 50.00 |

5. Use the select command to get the details of the students with marks more than 80.

SELECT \* FROM STUDENT1

WHERE MARKS > 80;

Answer:

|  |  |  |
| --- | --- | --- |
| ID | NAME | MARKS |
| 1 | DEV | 97.00 |
| 2 | ARVIND | 90.50 |
| 3 | DINESH | 85.00 |
| 8 | KARTHICK | 84.00 |

6. Find the min, max, sum, and average of the marks in a student marks table.

SELECT \* FROM STUDENT1;

SELECT MAX(MARKS) AS MAX\_MARKS FROM STUDENT1;

SELECT MIN(MARKS) AS MIN\_MARKS FROM STUDENT1;

SELECT SUM(MARKS) AS SUM\_MARKS FROM STUDENT1;

SELECT AVG(MARKS) AS AVG\_MARKS FROM STUDENT1;

Answer:

|  |
| --- |
| MAX\_MARKS |
| 97.00 |

|  |
| --- |
| MIN\_MARKS |
| 40.00 |

|  |
| --- |
| SUM\_MARKS |
| 762.00 |

|  |
| --- |
| AVG\_MARKS |
| 69.272727 |

7. Find the student names whose marks are greater than 60 and less than 40

SELECT NAME,MARKS FROM STUDENT1

WHERE MARKS > 40 AND MARKS < 60;

Answer:

|  |  |
| --- | --- |
| NAME | MARKS |
| MANI | 45.00 |
| NAVEEN | 55.00 |
| PRAVEEN | 50.00 |

8. Write a SQL query to order the (student ID, marks) table in descending order of the mark

SELECT ID,MARKS FROM STUDENT1

ORDER BY MARKS DESC;

Answer:

|  |  |
| --- | --- |
| ID | MARKS |
| 1 | 97.00 |
| 2 | 90.50 |
| 3 | 85.00 |
| 8 | 84.00 |
| 5 | 80.00 |
| 4 | 75.00 |
| 6 | 60.00 |
| 10 | 55.00 |
| 11 | 50.00 |
| 9 | 45.00 |
| 7 | 40.00 |

9. Write a SQL query to display the marks without decimal places, display the remainder after dividing marks by 3 and display the square of marks

SELECT FLOOR(MARKS) AS MRKS FROM STUDENT1;

SELECT FLOOR(MARKS) AS MRKS,MOD(MARKS,3),SQRT(MARKS) FROM STUDENT1;

Answer:

|  |  |  |
| --- | --- | --- |
| MRKS | MOD(MARKS,3) | SQRT(MARKS) |
| 97 | 1.00 |  |
| 90 | 0.50 |  |
| 85 | 1.00 |  |
| 75 | 0.00 |  |
| 80 | 2.00 |  |
| 60 | 0.00 |  |
| 40 | 1.00 |  |
| 84 | 0.00 |  |
| 45 | 0.00 |  |
| 55 | 1.00 |  |
| 50 | 2.00 |  |

10. Write a SQL query to display names into capital letters, small letters, display first 3 letters of name, display last 3 letters of name, display the position the letter A in name

SELECT LOWER(NAME) , UPPER(NAME) , LEFT(NAME,3) , RIGHT(NAME,3) FROM STUDENT1;

SELECT NAME,POSITION("A" IN NAME) FROM STUDENT1;

Answer:

|  |  |  |  |
| --- | --- | --- | --- |
| LOWER(NAME) | UPPER(NAME) | LEFT(NAME,3) | RIGHT(NAME,3) |
| dev | DEV | DEV | DEV |
| arvind | ARVIND | ARV | IND |
| dinesh | DINESH | DIN | ESH |
| gokul | GOKUL | GOK | KUL |
| vignesh | VIGNESH | VIG | ESH |
| vijay | VIJAY | VIJ | JAY |
| gopi | GOPI | GOP | OPI |
| karthick | KARTHICK | KAR | ICK |
| mani | MANI | MAN | ANI |
| naveen | NAVEEN | NAV | EEN |
| praveen | PRAVEEN | PRA | EEN |

|  |  |
| --- | --- |
| NAME | POSITION("A" IN NAME) |
| DEV | 0 |
| ARVIND | 1 |
| DINESH | 0 |
| GOKUL | 0 |
| VIGNESH | 0 |
| VIJAY | 4 |
| GOPI | 0 |
| KARTHICK | 2 |
| MANI | 2 |
| NAVEEN | 2 |
| PRAVEEN | 3 |

Additional Questions :

11. Remove extra spaces from left, right and both sides from the text – ” Innomatics Research Lab “.

SELECT TRIM(" Innomatics Research Lab ") as trim;

Answer:

|  |
| --- |
| trim |
| “Innomatics Research Lab ” |

SELECT LTRIM(" Innomatics Research Lab ") as ltrim; #remove leading space

Answer:

|  |
| --- |
| ltrim |
| “Innomatics Research Lab “ |

SELECT RTRIM(" Innomatics Research Lab ") as rtrim; # remove trailing space

Answer:

|  |
| --- |
| rtrim |
| “ Innomatics Research Lab” |

12. Display today’s date in “Date/Month/Year” format

SELECT DATE\_FORMAT(CURDATE(),'%d-%m-%Y') AS FORMATTED\_DATE;

Answer:

|  |
| --- |
| FORMATTED\_DATE |
| 11-10-2022 |

13. Display dayname, monthname, day, dayname, day of month, day of year for today’s date.

SELECT DATE\_FORMAT(CURDATE(), "%Y");

Answer:

|  |
| --- |
| DATE\_FORMAT(CURDATE(), "%Y") |
| 2022 |

SELECT DATE\_FORMAT(CURDATE(), "%d");

Answer:

|  |
| --- |
| DATE\_FORMAT(CURDATE(), "%d") |
| 11 |

SELECT DATE\_FORMAT(CURDATE(), "%m");

Answer:

|  |
| --- |
| DATE\_FORMAT(CURDATE(), "%m") |
| 10 |

SELECT DATE\_FORMAT(CURDATE(), "%a");

Answer:

|  |
| --- |
| DATE\_FORMAT(CURDATE(), "%a") |
| TUE |

SELECT DATE\_FORMAT(CURDATE(), "%b");

Answer:

|  |
| --- |
| DATE\_FORMAT(CURDATE(), "%b") |
| OCT |

14. Explain the difference between primary key and foreign key.

| S.NO. | PRIMARY KEY | FOREIGN KEY |
| --- | --- | --- |
| 1 | A primary key is used to ensure data in the specific column is unique. | A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. |
| 2 | It uniquely identifies a record in the relational database table. | It refers to the field in a table which is the primary key of another table. |
| 3 | Only one primary key is allowed in a table. | Whereas more than one foreign key are allowed in a table. |
| 4 | It is a combination of UNIQUE and Not Null constraints. | It can contain duplicate values and a table in a relational database. |
| 5 | It does not allow NULL values. | It can also contain NULL values. |
| 6 | Its value cannot be deleted from the parent table. | Its value can be deleted from the child table. |
| 7 | It constraint can be implicitly defined on the temporary tables. | It constraint cannot be defined on the local or global temporary tables. |