SQL assignment

```
mysql> select * from studentbasicinfo;
+---------+----------+--------+----------+
| name    | surname  | rollno | address  |
+---------+----------+--------+----------+
| aman    | patel    |    100 | lko      |
| shivam  | rajput   |    101 | lko      |
| akash   | rai      |    102 | lko      |
| piyush  | saraswat |    103 | delhi    |
| tarun   | bajaj    |    104 | delhi    |
| utkarsh | sonkar   |    105 | kanpur   |
| avinash | kumar    |    106 | banglore |
| rohan   | agarlwal |    107 | jhansi   |
| anadi   | bajpai   |    108 | lko      |
| atul    | sahai    |    109 | noida    |
+---------+----------+--------+----------+
10 rows in set (0.70 sec)
```

```
mysql> select * from paymentdetails;
+--------+------+---------+
| rollno | paid | balance |
+--------+------+---------+
|    100 | 1000 |       0 |
|    101 |  900 |     100 |
|    102 |  800 |     200 |
|    103 |  700 |     300 |
|    104 |  600 |     400 |
|    105 |  700 |     300 |
|    106 |  800 |     200 |
|    107 |  500 |     500 |
|    108 |  600 |     400 |
|    109 | 1000 |       0 |
+--------+------+---------+
10 rows in set (0.03 sec)
```

```
mysql> select * from subjectinfo;
+----------+--------+------------+---------------+---------+
| opted    | rollno | totalmarks | obtainedmarks | percent |
+----------+--------+------------+---------------+---------+
| science  |    100 |        100 |            95 |      95 |
| science  |    101 |        100 |            90 |      90 |
| science  |    102 |        100 |            85 |      85 |
| science  |    103 |        100 |            80 |      80 |
| arts     |    104 |        100 |           100 |     100 |
| arts     |    105 |        100 |            97 |      97 |
| arts     |    106 |        100 |            95 |      95 |
| arts     |    107 |        100 |           100 |     100 |
| commerce |    108 |        100 |            89 |      89 |
| commerce |    109 |        100 |            93 |      93 |
+----------+--------+------------+---------------+---------+
10 rows in set (0.44 sec)
```

```
mysql> select * from schlarshipinfo;
+--------+------------+-------------+---------+----------+
| rollno | name       | description | amount  | category |
+--------+------------+-------------+---------+----------+
|    100 | btech fees | engineering | 1200000 | A        |
|    101 | btech fees | engineering | 1200000 | A        |
|    102 | btech fees | engineering | 1200000 | A        |
|    103 | bsc fees   | economics   | 1200000 | B        |
|    104 | bsc fees   | economics   | 1200000 | B        |
|    105 | bsc fees   | maths       | 1000000 | C        |
|    106 | bsc fees   | maths       | 1000000 | C        |
|    107 | bsc fees   | maths       | 1000000 | C        |
|    108 | bsc fees   | computing   |  900000 | C        |
|    109 | bsc fees   | computing   |  900000 | C        |
+--------+------------+-------------+---------+----------+
10 rows in set (0.00 sec)
```

>Joins and when to use:

INNER JOIN

This type of join returns those records which have matching values in both tables. So, if you perform an INNER join operation between the Employee table and the Projects table, all the tuples which have matching values in both the tables will be given as output.

FULL JOIN

Full Join or the Full Outer Join returns all those records which either have a match in the left(Table1) or the right(Table2) table.

LEFT JOIN

The LEFT JOIN or the LEFT OUTER JOIN returns all the records from the left table and also those records which satisfy a condition from the right table. Also, for the records having no matching values in the right table, the output or the result-set will contain the NULL values.

RIGHT JOIN

The RIGHT JOIN or the RIGHT OUTER JOIN returns all the records from the right table and also those records which satisfy a condition from the left table. Also, for the records having no matching values in the left table, the output or the result-set will contain the NULL values.

A Natural Join is also a Join operation that is used to give you an output based on the columns in both the tables between which, this join operation must be implemented. To understand the situations n which natural join is used, you need to understand the difference between Natural Join and Inner Join.

The main difference the Natural Join and the Inner Join relies on the number of columns returned. Refer below for example.

Table1

Column1	Column2
a	b

Table2

Column1	Column3
a	c

Now, if you apply INNER JOIN on these 2 tables, you will see an output as below:

1.Column1	1.Column2	2.Column1	2.Column3
a	b	a	c

If you apply NATURAL JOIN, on the above two tables, the output will be as below:

Column1	Column2	Column3
a	b	c

From the above example, you can clearly see that the number of columns returned from the Inner Join is more than that of the number of columns returned from Natural Join. So, if you wish to get an output, with less number of columns, then you can use Natural Join

---

->Mention the differences between the delete, drop and truncate commands

Unlike TRUNCATE which only deletes the data of the tables, the DROP command deletes the data of the table as well as removes the entire schema/structure of the table from the database.

->Difference between Stored Procedure, SQL Function, and Trigger

Executable

Store procedure: We can execute the stored procedures when required.

Function: We can call a function whenever required. Function can't be executed because a function is not in pre-compiled form.

Trigger: Trigger can be executed automatically on specified action on a table like, update, delete, or update.

Calling

Stored procedure: Stored Procedures can't be called from a function because functions can be called from a select statement and Stored Procedures can't be called from. But you can call Store Procedure from Trigger.

Function: Function can be called from Store Procedure or Trigger.

Trigger: Trigger can’t be called from Store Procedure or Function.

Parameter

Store procedure: Stored Procedures can accept any type of parameter. Stored Procedures also accept out parameter.

Function: Function can accept any type of parameter. But function can’t accept out parameter.

Trigger: We can’t pass a parameter to trigger.

Return

Store procedure: Stored Procedures may or may not return any values (Single or table) on execution.

Function: Function must return any value.

Trigger: Trigger never return value on execution.