

## 2. Write a stored procedure to calculate the total marks of each student

```
mysql> select * from student_marks;
+-----+-----+-----+-----+-----+
| Roll_NO | Name | Science | English | Maths | Economics |
+-----+-----+-----+-----+-----+
|      1 | divya |      52 |       45 |     62 |       36 |
|      2 | suraj |      23 |       62 |     52 |       41 |
|      3 | gayu  |      52 |       63 |     85 |       62 |
|      4 | veena |      52 |       62 |     39 |       41 |
+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)
```

```
mysql> SELECT *, SUM(Science + English + Maths + Economics)as Total
-> from student_marks
-> Group by Roll_NO;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| Roll_NO | Name | Science | English | Maths | Economics | Total |
+-----+-----+-----+-----+-----+-----+-----+
|      1 | divya |      52 |       45 |     62 |       36 |     195 |
|      2 | suraj |      23 |       62 |     52 |       41 |     178 |
|      3 | gayu  |      52 |       63 |     85 |       62 |     262 |
|      4 | veena |      52 |       62 |     39 |       41 |     194 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.17 sec)
```

## STORED PROCEDURE

```
mysql> CREATE PROCEDURE TotalMarks(IN ROLL INT )
-> SELECT *,SUM(Science+English+Maths+Economics) as Total
-> from student_marks
-> where Roll_NO = ROLL;
```

```
mysql>CALL TotalMarks(1);
```

```
+-----+-----+-----+-----+-----+-----+-----+
| Roll_NO | Name | Science | English | Maths | Economics | Total |
+-----+-----+-----+-----+-----+-----+-----+
|      1 | divya |      52 |       45 |     62 |       36 |     195 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.03 sec)
```

```
mysql>CALL TotalMarks(2);
```

```
+-----+-----+-----+-----+-----+-----+-----+
| Roll_NO | Name | Science | English | Maths | Economics | Total |
+-----+-----+-----+-----+-----+-----+-----+
|      2 | suraj |      23 |       62 |     52 |       41 |     178 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.17 sec)
```

**3. Write a query which displays the list of students in ascending order based on their name along with the highest to lowest marks scored in different subject by each student.**

```
mysql> select *, SUM(Science + English + Maths + Economics) as Total  
-> FROM student_marks  
-> GROUP BY Roll_NO  
-> ORDER BY Name ASC, Total DESC;
```

```
mysql> select *, SUM(Science + English + Maths + Economics) as Total  
-> FROM student_marks  
-> GROUP BY Name  
-> ORDER BY Name ASC, Total DESC;  
+-----+-----+-----+-----+-----+-----+  
| Roll_NO | Name   | Science | English | Maths  | Economics | Total  |  
+-----+-----+-----+-----+-----+-----+  
|      1 | divya  |      52 |      45 |      62 |       36 |    195 |  
|      3 | gayu   |      52 |      63 |      85 |       62 |    262 |  
|      2 | suraj   |      23 |      62 |      52 |       41 |    178 |  
|      4 | veena   |      52 |      62 |      39 |       41 |    194 |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.13 sec)
```

**4. Add constraints in such a way that if a primary key related record is deleted then the corresponding record containing the foreign key should be deleted**

**DELETE CASCADE:** When we create a foreign key using this option, it deletes the referencing rows in the child table when the referenced row is deleted in the parent table which has a primary key.

**Table 1 : Countries**

```
mysql> CREATE TABLE countries (CountryID int primary key, CountryName  
Varchar(50), CountryCode varchar(3));
```

```
mysql> select * from countries;  
+-----+-----+-----+  
| CountryID | CountryName | CountryCode |  
+-----+-----+-----+  
|      101 | INDIA      | IN         |  
|      102 | United States | US         |  
|      103 | Australia   | AUS        |  
|      104 | United Kingdom | UK         |  
+-----+-----+-----+  
4 rows in set (0.18 sec)
```

**Table 2 : States**

```
mysql> CREATE TABLE states (StateID int Primary key, StateName
```

```
varchar(50),StateCode varchar(3),CountryID int);
```

```
mysql> select * from states;
+-----+-----+-----+-----+
| StateID | StateName | StateCode | CountryID |
+-----+-----+-----+-----+
|      1 | Maharashtra | MU        |      101 |
|      2 | Taxas       | TX        |      102 |
|      3 | Arizona     | AR        |      102 |
|      4 | England     | ENG       |      103 |
|      5 | Gujarat     | GUJ       |      101 |
|      6 | London      | LND       |      103 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> ALTER TABLE states ADD CONSTRAINT FOREIGN KEY(CountryID)
REFERENCES Countries (CountryID) ON DELETE CASCADE;
```

```
mysql> DELETE FROM Countries WHERE CountryID = 102;
```

```
mysql> Delete from Countries where CountryID = 102;
Query OK, 1 row affected (2.13 sec)
```

```
mysql> select * from countries;
```

```
+-----+-----+-----+
| CountryID | CountryName | CountryCode |
+-----+-----+-----+
|      101 | INDIA      | IN          |
|      103 | Australia   | AUS         |
|      104 | United Kingdom | UK          |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from states;
```

```
+-----+-----+-----+-----+
| StateID | StateName | StateCode | CountryID |
+-----+-----+-----+-----+
|      1 | Maharashtra | MU        |      101 |
|      4 | England     | ENG       |      103 |
|      5 | Gujarat     | GUJ       |      101 |
|      6 | London      | LND       |      103 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

**5. Add constraints in such a way that if a primary key related record is deleted then the corresponding record containing the foreign key should be updated**

**UPDATE CASCADE:** When we create a foreign key using UPDATE CASCADE the referencing rows are updated in the child table when the referenced row is updated in the parent table which has a primary key.

```
mysql>ALTER TABLE states ADD CONSTRAINT FOREIGN KEY(CountryID)
REFERENCES Countries (CountryID) ON UPDATE CASCADE;
```

```
mysql> UPDATE states SET CountryID = 102 where CountryID = 103;
```