TRIGGER EXERCISES:

1) We have a table student_marks with 10 columns and 4 rows. There are data only in STUDENT_ID and NAME columns.

mysql> SELECT * FROM STUDENT_MARKS;

```
mysql> SELECT * FROM STUDENT_MARKS;
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
| STUDENT_ID | NAME            | SUB1 | SUB2 | SUB3 | SUB4 | SUB5 | TOTAL | PER_MARKS | GRADE |
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
|          1 | Steven King     |    0 |    0 |    0 |    0 |    0 |     0 |      0.00 |       |
|          2 | Neena  Kochhar  |    0 |    0 |    0 |    0 |    0 |     0 |      0.00 |       |
|          3 | Lex  De Haan    |    0 |    0 |    0 |    0 |    0 |     0 |      0.00 |       |
|          4 | Alexander Hunold|    0 |    0 |    0 |    0 |    0 |     0 |      0.00 |       |
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
```

Now the exam is over and we have received all subject marks, now we will update the table, total marks of all subject, the percentage of total marks and grade will be automatically calculated. For this sample calculation, the following conditions are assumed :

Total Marks (will be stored in TOTAL column) : TOTAL = SUB1 + SUB2 + SUB3 + SUB4 + SUB5

Percentage of Marks (will be stored in PER_MARKS column) : PER_MARKS = (TOTAL)/5

Grade (will be stored GRADE column) :
- If PER_MARKS>=90 -> 'EXCELLENT'
- If PER_MARKS>=75 AND PER_MARKS<90 -> 'VERY GOOD'
- If PER_MARKS>=60 AND PER_MARKS<75 -> 'GOOD'
- If PER_MARKS>=40 AND PER_MARKS<60 -> 'AVERAGE'
- If PER_MARKS<40-> 'NOT PROMOTED'

2) Create a small example database for a blogging application. Two tables are required:

- **`blog`:** stores a unique post ID, the title, content, and a deleted flag.

- **`audit`:** stores a basic set of historical changes with the blog post ID, the change type (NEW, EDIT or DELETE) and the date/time of that change.

We require two triggers:

- When a record is INSERTed into the blog table, we want to add a new entry into the audit table containing the blog ID and a type of ‘NEW’ (or ‘DELETE’ if it was deleted immediately).

- When a record is UPDATEd in the blog table, we want to add a new entry into the audit table containing the blog ID and a type of ‘EDIT’ or ‘DELETE’ if the deleted flag is set.

Creating tables:

CREATE TABLE blog (

id int,

title varchar(20),

content varchar(20),

deleted int,

PRIMARY KEY (id)

);

CREATE TABLE audit(blog_id int,

changetype enum('NEW','EDIT','DELETE') NOT NULL,

changetime timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

foreign key(blog_id) references blog(id)

) ;

3) Create hospital database requires 4 tables:

- **PATIENT**
- **ROOM**
- **MEDICINE**
- **BILL**

We require trigger:
When a record is DELETED from PATIENT table, It should be deleted from MEDICINE, BILL tables and set the room status as ‘EMPTY’ and patientID as null in ROOM table.