NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
NH 66, Srinivas Nagar, Surathkal, Mangaluru, Karnataka 575025, India

DBMS Assignment-8 on Trigger

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. We will update the table, total marks of all subjects, the percentage of total marks and grade will be automatically calculated. For this sample calculation, the following conditions are assumed: (When you entered 5 subjects mark in DML Query, then the trigger will automatically update total, percentage and grade for each student at a time)
   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’

Example output:
mysql> SELECT * FROM STUDENT_MARKS;

STUDENT_ID	NAME	SUB1	SUB1	SUB1	SUB1	SUB1	TOTAL	PER_MARKS	GRADE
1	**Steven King**	**54**	**69**	**89**	**87**	**59**	**358**	**71.60**	**GOOD**
2	Neena Kochhar	0	0	0	0	0	0	0.0	
3	Lex De Haan	0	0	0	0	0	0	0.0	
4	Alexander Hunold	0	0	0	0	0	0	0.0	

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:
a) 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)
b) 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));
```

Example Output:

a)

A new entry appears in the `blog` table as you'd expect:

id	title	content	deleted
1	Article One	Initial text	0

In addition, a new entry appears in our `audit` table:

id	blog_id	changetype	changetime
1	1	NEW	2011-05-20 09:00:00

b)

```
UPDATE blog SET content = 'Edited text' WHERE id = 1;
```

As well as changing the post, a new entry appears in the `audit` table:

id	blog_id	changetype	changetime
1	1	NEW	2011-05-20 09:00:00
2	1	EDIT	2011-05-20 09:01:00

3. Create a **Customer_Bank table** which consists of **Account No, CustomerName, Address, Branch_code, Type_of_Transaction and Balance_Amount.** Create a trigger for updating the balance automatically depends on the type of the transaction which is made by the customer. Include the following constraints:
   - Customers are not allowed to withdraw amount greater than Rs. 50000 in one transaction
   - Customers are not allowed to withdraw if the balance amount will fall below of minimum balance
     (you may use one or more than one trigger is depends on your choice)

4. Create a Cursor for the table created in **first problem** (**STUDENT_MARKS table**)
   Consider all 5 marks for all the students were updated in the **STUDENT_MARKS table** alone
   Write a code using cursor will update the total, percentage and grade for all the students at once

5. Create a Cursor for the table created in **third problem** (**Customer_Bank table**)
   (Alter the table add with CustomerLevel Column in **Customer_Bank**)
   Write a code to know about the customer level based on the balance amount they maintained their account in Bank
   Customer Level will be calculated as:
   - Balance > 1,00,000 THEN CustomerLevel = 'PLATINUM';
   - Balance is between 50,000 to 1,00,000 THEN CustomerLevel = 'GOLD';
   - Balance < 50,000 THEN CustomerLevel = 'SILVER';

Put all your screenshots (query with output) in a single PDF file and upload. The PDF must contain your name and roll no.

(10)