Indian Institute of Information Technology Chittoor, Sri City

Course: **Database Management System**

Duration: **120 Minutes**

Roll No.: **Yash Gupta**

Name: S202000010234

INSTRUCTIONS:

1. Read all questions carefully and answer them in the space provided (**Strictly within the box / space provided**).
2. Convert this document to **pdf (portable document format)** after filling the boxes.
3. The name of the pdf must be your **Roll no.** Upload single pdf file.
4. Crop only required part in the screenshot. Don't paste the entire screenshot unnecessarily.

---

1. Consider the LibGen website where the administrators promote their books by displaying the front page. Create a table (name:LibGen_your Roll Number) with the following constraints

   Id → NOT NULL AND UNIQUE

   type → varchar

   cost → int and cost >0

   Promoted → True (fixed)

   Create another new table called "Summary" with three columns:

   type→ category name

   priceMin→ the lowest price for all promoted books and

   priceMax→ highest price for all promoted books

   The "promoted" attribute on a promoted book is set to True.

   Create a trigger so that, when a tuple from the "Summary" table is deleted, all Books from the corresponding category have their "promoted" attribute set to False. For example, if the entry in "Summary" for category "Novel" is deleted, all entries in "Books" with category "Novel" have their "promoted" attribute set to False.

Insert table data here

id	type	COST	promoted
10001	Fiction	800	true
10002	Novel	2000	true
10003	Fables	1500	true
10004	Novel	800	true
10005	Autobiography	1000	true
10006	Fiction	3000	true
10007	Novel	1200	true
10008	Fables	1800	true
10009	Fiction	5500	true
10010	Novel	500	true

10 rows in set (0.01 sec)

type	priceMin	priceMax
Fiction	800	5500
Novel	500	2000
Fables	1500	1800
Autobiography	1000	1000
Fantasy	0	0
Romance	0	0
Thriller	0	0
Adventure	0	0
Finance	0	0
Sci-fi	0	0

10 rows in set (0.00 sec)

Program:

```
mysql> delimiter //
mysql> create trigger trig
    -> after delete
    -> on Summary for each row
    -> begin
    -> declare del_type varchar(40);
    -> set del_type = old.type;
    -> update libgen_S20200010234 set promoted = "False" where type = del_type;
    -> end //
Query OK, 0 rows affected (0.02 sec)

mysql> delimiter ;
```

Results:

```
mysql> select * from libgen_S20200010234;
+-------+---------------+------+----------+
| id    | type          | COST | promoted |
+-------+---------------+------+----------+
| 10001 | Fiction       |  800 | true     |
| 10002 | Novel         | 2000 | False    |
| 10003 | Fables        | 1500 | true     |
| 10004 | Novel         |  800 | False    |
| 10005 | Autobiography | 1000 | true     |
| 10006 | Fiction       | 3000 | true     |
| 10007 | Novel         | 1200 | False    |
| 10008 | Fables        | 1800 | true     |
| 10009 | Fiction       | 5500 | true     |
| 10010 | Novel         |  500 | False    |
+-------+---------------+------+----------+
10 rows in set (0.00 sec)
```

Any Observations:

When I deleted "Novel" category from the database, all entries in "Books" with category "Novel" have their "promoted" attribute set to False.

2. Assume the below schema for customer data belonging to BSNL mobile company.
   CUSTOMER (ID, Name, age, MobileNum, Plan)
   BASEPLAN (Code, ConnectionFee, PricePerSecond)
   PHONECALL (ID, Date, Time, CalledNum, Seconds)
   BILL (ID, Month, Year, amount)

   Create a trigger that updates the customer's bill after every phone call.

Insert screenshots of table data in the space provided.

```
mysql> select * from customer
    -> ;
+----+--------+------+------------+---------+
| id | name   | age  | Mobilenum  | plan    |
+----+--------+------+------------+---------+
|  1 | Priya  |   30 | 9567543765 | 1_month |
|  2 | Riya   |   10 | 9037621897 | 6_month |
|  3 | Shreya |   15 | 9389023674 | 3_month |
|  4 | Prachi |   19 | 9673876475 | 6_month |
|  5 | Manju  |   90 | 8637263746 | 1_month |
+----+--------+------+------------+---------+
5 rows in set (0.00 sec)
```

```
mysql> select * from baseplan;
+---------+----------------+----------------+
| code    | Connection_fee | pricepersecond |
+---------+----------------+----------------+
| 1_month |           1.12 |           0.11 |
| 3_month |           1.88 |           0.16 |
| 4_month |           3.64 |           0.90 |
| 6_month |           2.55 |           0.16 |
| 1_year  |           5.42 |           0.01 |
+---------+----------------+----------------+
5 rows in set (0.00 sec)
```

```
mysql> select * from phonecall;
+------+------------+-----------+------------+---------+
| id   | date_call  | time_call | callednum  | seconds |
+------+------------+-----------+------------+---------+
|    1 | 2021-11-15 | 02:52     | 9037621897 |     210 |
|    2 | 2021-11-04 | 12:42     | 9567543765 |      10 |
|    3 | 2021-11-05 | 10:02     | 9999988888 |       5 |
|    4 | 2021-11-26 | 6:02      | 9876543211 |      20 |
|    5 | 2021-11-10 | 7:20      | 9272625242 |      80 |
+------+------------+-----------+------------+---------+
5 rows in set (0.00 sec)
```

```
mysql> select * from bill;
+------+----------+------+---------+
| id   | month    | year | amount  |
+------+----------+------+---------+
|    1 | november | 2021 | 1000.00 |
|    2 | november | 2021 | 2000.00 |
|    3 | november | 2021 | 1200.00 |
|    4 | november | 2021 | 3400.00 |
|    5 | november | 2021 | 1340.00 |
+------+----------+------+---------+
5 rows in set (0.00 sec)
```

Program:

```
mysql> delimiter //
mysql> create trigger CallCharges
    -> after insert on PHONECALL
    -> for each row
    -> begin
    ->  update BILL B
    ->  set Amount = Amount + ( select BP.Connection_Fee +
    ->  BP.PricePerSecond * new.Seconds
    ->  from BASEPLAN BP join CUSTOMER C
    ->  on C.Plan = BP.Code
    ->  where new.ID = C.ID )
    ->  where B.ID = new.ID;
    -> end //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
```

Result:

```
mysql> select * from bill where id = 1;
+------+----------+------+---------+
| id   | month    | year | amount  |
+------+----------+------+---------+
|    1 | november | 2021 | 1005.52 |
+------+----------+------+---------+
```

Any Observations/Justification:

Updates the bill after every call and decide the amount accordingly.