Q1) Create a customer audit table such that it is updated for each update in customer table.

Sol)

CREATE TABLE CUSTOMER\_AUDIT( ID INT AUTO\_INCREMENT PRIMARY KEY,

-> NAME VARCHAR(30) NOT NULL,

-> STREET VARCHAR(25) NOT NULL,

-> ACTION VARCHAR(50) DEFAULT NULL)$$

mysql> DELIMITER $$

CREATE TRIGGER BEFORE\_CUSTOMER\_UPDATE

-> BEFORE UPDATE ON CUSTOMER

-> FOR EACH ROW

-> BEGIN

-> INSERT INTO CUSTOMER\_AUDIT

-> SET ACTION='UPDATE',

-> NAME=OLD.NAME,

-> STREET=OLD.STREET,

-> END $$

To check use - UPDATE CUSTOMER SET STREET='ABC' WHERE NAME='haritha';

SELECT \* FROM CUSTOMER\_AUDIT;

Q2)Create an account audit table such that it concats "update record- acc\_no deleted' for each delete in accounts table.

Sol)

CREATE TABLE ACCOUNT\_AUDIT( ACC\_NO VARCHAR(20) NOT NULL,BR\_NAME VARCHAR(25) NOT NULL,BALANCE DECIMAL(10,0) DEFAULT NULL)$$

CREATE TRIGGER AFTER\_ACCOUNT\_DELETE

AFTER DELETE ON ACCOUNT

FOR EACH ROW

BEGIN

INSERT INTO ACCOUNT\_AUDIT((OLD.ACC\_NO,OLD.BR\_NAME,OLD.BALANCE,CONCAT('UPDATE RECORD',OLD.ACC\_NO,'DELETED'));

END $$

To check use - DELETE FROM ACCOUNT WHERE ACC\_NO = 2000000001 ;

SELECT \* FROM ACCOUNT;

SELECT \* FROM ACCOUNT\_AUDIT;

Q3)Use the following query to create a student\_marks table and to update it with subject marks.For each update output the grade.

CREATE TABLE STUDENT\_MARKS(ID INT AUTO INCREMENT,NAME VARCHAR(20),SUB1 INT,SUB2 INT,SUB3 INT,SUB4 INT,SUB5 INT, TOTAL INT,PER\_MARKS INT,GRADE CHAR(2));

UPDATE STUDENT\_MARKS SET SUB1 = 54, SUB2 = 69, SUB3 = 89, SUB4 = 87, SUB5 = 59 WHERE STUDENT\_ID = 1;

Total Marks : TOTAL = SUB1 + SUB2 + SUB3 + SUB4 + SUB5

Percentage of Marks : 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'

Sol)

DELIMITER

$$

CREATE TRIGGER `student\_marks`

BEFORE UPDATE

ON STUDENT\_MARKS FOR EACH ROW

BEGIN

SET NEW.TOTAL = NEW.SUB1 + NEW.SUB2 + NEW.SUB3 + NEW.SUB4 + NEW.SUB5;

SET NEW.PER\_MARKS = NEW.TOTAL/5;

IF NEW.PER\_MARKS >=90 THEN

SET NEW.GRADE = 'EXCELLENT';

ELSEIF NEW.PER\_MARKS>=75 AND NEW.PER\_MARKS<90 THEN

SET NEW.GRADE = 'VERY GOOD';

ELSEIF NEW.PER\_MARKS>=60 AND NEW.PER\_MARKS<75 THEN

SET NEW.GRADE = 'GOOD';

ELSEIF NEW.PER\_MARKS>=40 AND NEW.PER\_MARKS<60 THEN

SET NEW.GRADE = 'AVERAGE';

ELSESET NEW.GRADE = 'NOT PROMOTED';

END IF;

END$$

To check use - SELECT \* FROM STUDENT\_MARKS;

Q4) Write a trigger such that for each update in table1

- the values are copied in table2

- those values are deleted from table3

- frequency of those values are stored in second column of table4

CREATE TABLE table1(a1 INT);

CREATE TABLE table2(a2 INT);

CREATE TABLE table3(a3 INT NOT NULL AUTO\_INCREMENT PRIMARY KEY);

CREATE TABLE table4(

a4 INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

b4 INT DEFAULT 0

);

Sol)CREATE TRIGGER change\_table

BEFORE INSERT ON table1

FOR EACH ROW

BEGIN

INSERT INTO table2 SET a2 = NEW.a1;

DELETE FROM table3 WHERE a3 = NEW.a1;

UPDATE table4 SET b4 = b4 + 1 WHERE a4 = NEW.a1;

END$$

To check use -

INSERT INTO table3 (a3) VALUES

(NULL), (NULL), (NULL), (NULL), (NULL),

(NULL), (NULL), (NULL), (NULL), (NULL);

INSERT INTO table4 (a4) VALUES

(0), (0), (0), (0), (0), (0), (0), (0), (0), (0);

INSERT INTO test1 VALUES

(1), (3), (1), (7), (1), (8), (4), (4);

Output - mysql> SELECT \* FROM table1;

+------+

| a1 |

+------+

| 1 |

| 3 |

| 1 |

| 7 |

| 1 |

| 8 |

| 4 |

| 4 |

+------+

8 rows in set (0.00 sec)

mysql> SELECT \* FROM table2;

+------+

| a2 |

+------+

| 1 |

| 3 |

| 1 |

| 7 |

| 1 |

| 8 |

| 4 |

| 4 |

+------+

8 rows in set (0.00 sec)

mysql> SELECT \* FROM table3;

+----+

| a3 |

+----+

| 2 |

| 5 |

| 6 |

| 9 |

| 10 |

+----+

5 rows in set (0.00 sec)

mysql> SELECT \* FROM table4;

+----+------+

| a4 | b4 |

+----+------+

| 1 | 3 |

| 2 | 0 |

| 3 | 1 |

| 4 | 2 |

| 5 | 0 |

| 6 | 0 |

| 7 | 1 |

| 8 | 1 |

| 9 | 0 |

| 10 | 0 |

+----+------+

10 rows in set (0.00 sec)

Q5)Create the following two tables :

ext\_words

-------------

| id | word |

-------------

| 1 | this |

-------------

| 2 | that |

-------------

| 3 | this |

-------------

ext\_words\_count

---------------------

| id | word | count |

---------------------

| 1 | this | 2 |

---------------------

| 2 | that | 1 |

---------------------

Create a trigger which updates count when word is updated in ext\_words and if ext\_words.word does not exist in ext\_words\_count when ext\_words is updated,insert the word into ext\_words\_count and set count as 1.

Sol)

DELIMITER $$

CREATE TRIGGER update\_count

AFTER UPDATE ON ext\_words

FOR EACH ROW

BEGIN

IF NOT EXISTS (SELECT 1 FROM ext\_words\_count WHERE word = NEW.word) THEN

INSERT INTO ext\_words\_count (word) VALUES (NEW.word);

ELSE

UPDATE ext\_words\_count SET word\_count = word\_count + 1 WHERE word = NEW.word;

END IF;

END $$

DELIMITER;

OR

DELIMITER $$

CREATE TRIGGER update\_count

AFTER UPDATE ON ext\_words

FOR EACH ROW

BEGIN

SELECT count INTO @x FROM ext\_words\_count LIMIT 1;

UPDATE ext\_words\_count

SET count = @x + 1

WHERE word = NEW.word;

END$$

To check use - insert into ext\_words values ('there');