**Instructions:**

Please share your answers filled in line in the Word document. Submitcode

separately wherever applicable.

Please ensure you update all the details:

**Name: E PAVAN KUMAR Batch ID:04/08/2023**

**Topic: Introduction to Database**

1. **Write an SQL query to accomplish the following tasks:**

* 1. Create a database named **student\_db**.
  2. Create a table named **students\_details** with columns **id** (integer), **name** (varchar), **age** (integer), and **grade** (float). id should be set as the primary key.
  3. Insert any four records into **students\_details**.
  4. Create a new table named **students\_details\_copy** with the same columns as **students\_details**. **id** should also be set as the primary key.
  5. Create a trigger named **after\_insert\_details** that inserts a new record into **students\_details\_copy** every time a record is inserted into students\_details.
  6. Insert a new record into **students\_details.**
  7. check whether a record is filling in **students\_details\_copy** as you insert value in **students\_details.**

**ANSWERS;** #1a) answer

CREATE DATABASE STUDENT\_DB;

#1b) answer

CREATE TABLE STUDENTS\_DETAILS(id int, name varchar(20), AGE INT, GRADE FLOAT);

SELECT \* FROM STUDENTS\_DETAILS;

ALTER TABLE STUDENTS\_DETAILS

ADD CONSTRAINT ID

PRIMARY KEY(ID);

#1c) answer

INSERT INTO STUDENTS\_DETAILS (id, name, AGE, GRADE) VALUES

(1, 'John', 22, 85.5),

(2, 'Alice', 21, 92.0),

(3, 'Bob', 23, 78.5),

(4, 'Eva', 20, 88.7);

#1d) answer

CREATE TABLE students\_details\_copy(id int PRIMARY KEY, name varchar(20), AGE INT, GRADE FLOAT);

#1e) answer

delimiter //

create trigger after\_insert\_details

after insert on students\_DETAILS

for each row

begin

insert into students\_details\_copy (id, name , age, grade)

values(new.id, new.name, new.age, new.grade);

end;

//

delimiter ;

#1f) answer

insert into students\_details values(5, 'mam', 25,68);

#1g) answer

select \* from students\_details\_copy;

1. **Write an SQL question that accomplishes the following tasks:**

* 1. use **student\_db ,**
  2. Create a trigger named **update\_grade** that automatically updates the **grade** column every time a record in **students\_details** is updated based on the following criteria:
  3. If the updated record has an age value less than 18, multiply the grade by 0.9.
  4. If the updated record has an age value between 18 and 20 (inclusive), multiply the grade by 1.1.

* 1. If the updated record has an age value greater than 20, multiply the grade by 1.05.
  2. Update the age value of one of the records in students\_new to see the trigger in action.

**ANSWERS;**

#2a) ANSWER

use student\_db;

select \* from students\_details;

#2b,2c,2d,2e) ANSWER

delimiter //

create trigger update\_grade

before update on students\_details

for each row

begin

if new.age < 18 then

set new.grade = new.grade \* 0.9;

elseif

new.age>= 18 and new.age <= 20

then set new.grade = new.grade \* 1.1;

else

set new.grade = new.grade \* 1.05;

end if;

end;

// delimiter ;

#2f) ANSWER

select \* from students\_details ;

/\*

1 John 22 85.5

2 Alice 21 92

3 Bob 23 78.5

4 Eva 20 88.7

5 mam 25 68 \*/

set sql\_safe\_updates = 0;

update students\_details set age = 30 where name = 'bob';

select \* from students\_details ;

/\*

1 John 22 85.5

2 Alice 21 92

3 Bob 30 78.5

4 Eva 20 88.7

5 mam 25 68 \*/

1. **Explain the difference between the AFTER and INSTEAD OF trigger operators in SQL.**

**ANSWER**

AFTER Trigger: It fires after the triggering event (e.g., INSERT, UPDATE, DELETE) has occurred. It is used to perform actions after the event, such as auditing or logging changes.

INSTEAD OF Trigger: It fires instead of the triggering event. It is commonly used with views to intercept the event and perform custom actions, including preventing the event from occurring, or transforming the data before the event.

1. **What is the purpose of the INSTEAD OF DELETE trigger operator in SQL?**

**ANSWER**

The purpose of an INSTEAD OF DELETE trigger is to intercept and handle DELETE operations on a view or table. Instead of allowing the standard DELETE operation to proceed, this trigger allows you to define custom logic for handling deletions, which can include deleting related records in other tables, preventing certain deletions, or performing other actions before deleting. INSTEAD OF DELETE triggers are often used with complex views or to enforce referential integrity constraints in databases.