DBMS Lab 7

V Venkataraman 106118106

Q1

CREATE DATABASE LAB7;

USE LAB7;

DROP TABLE IF EXISTS PRODUCT;

DROP TABLE IF EXISTS SALE;

DROP TABLE IF EXISTS SALEITEM;

CREATE TABLE PRODUCT (

BARCODE INT NOT NULL,

PNAME VARCHAR(40),

PRICE INT,

QUANTITYINSTOCK INT,

PRIMARY KEY (BARCODE)

);

CREATE TABLE SALE (

SALEID INT NOT NULL,

DELIVERYADDRESS VARCHAR(40),

CREDITCARD INT,

PRIMARY KEY(SALEID)

);

CREATE TABLE SALEITEM (

SALEID INT NOT NULL,

BARCODE INT NOT NULL,

QUANTITY INT,

PRIMARY KEY (BARCODE, SALEID),

FOREIGN KEY (BARCODE) REFERENCES PRODUCT(BARCODE)

);

CREATE TRIGGER UPDATEAVAILABLEQUANTITY

AFTER INSERT

ON SALEITEM

FOR EACH ROW

UPDATE PRODUCT SET PRODUCT.QUANTITYINSTOCK = PRODUCT.QUANTITYINSTOCK - NEW.QUANTITY WHERE NEW.BARCODE = PRODUCT.BARCODE;

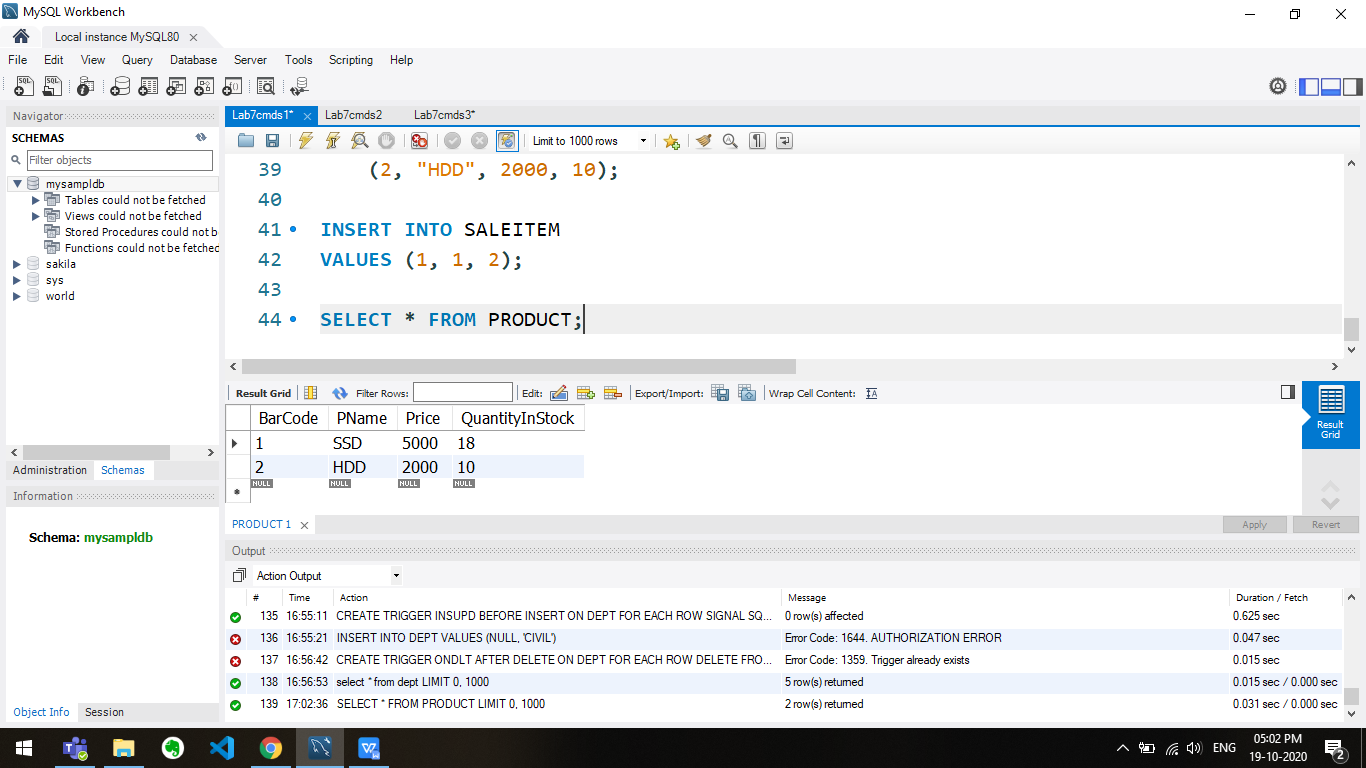
INSERT INTO PRODUCT

VALUES (1, "SSD", 5000, 20),

(2, "HDD", 2000, 10);

INSERT INTO SALEITEM

VALUES (1, 1, 2);



Q2

USE LAB7;

DROP TABLE IF EXISTS CONCERTS;

DROP TABLE IF EXISTS PERFORMERS;

DROP TABLE IF EXISTS ARENAS;

DROP TABLE IF EXISTS ACTIVITIES;

CREATE TABLE ACTIVITIES (

ACTIVITYID INT NOT NULL,

ACTIVITYNAME VARCHAR(40),

PRIMARY KEY (ACTIVITYID)

);

CREATE TABLE PERFORMERS (

PERFORMERID INT NOT NULL,

PERFORMERNAME VARCHAR(40),

STREET VARCHAR(40),

CITY VARCHAR(40),

STATE VARCHAR(40),

ZIP INT(6),

ACTIVITYID INT,

PRIMARY KEY (PERFORMERID),

FOREIGN KEY (ACTIVITYID) REFERENCES ACTIVITiES(ACTIVITYID)

);

CREATE TABLE ARENAS (

ARENAID INT NOT NULL,

ARENANAME VARCHAR(40),

CITY VARCHAR(40),

ARENACAPACITY VARCHAR(40),

PRIMARY KEY (ARENAID)

);

CREATE TABLE CONCERTS (

PERFORMERID INT,

ARENAID INT,

CONCERTDATE DATETIME,

TICKETPRICE INT,

PRIMARY KEY (PERFORMERID, ARENAID),

FOREIGN KEY (PERFORMERID) REFERENCES PERFORMERS(PERFORMERID),

FOREIGN KEY (ARENAID) REFERENCES ARENAS(ARENAID)

);

INSERT INTO ACTIVITiES

VALUES (1, "DANCING"),

(2, "SWIMMING"),

(3, "ATHLETICS");

INSERT INTO PERFORMERS

VALUES (1, "ASHUTOSH", "THUVAKUDI", "TRICHY", "TAMIL NADU", 620015, 1),

(2, "ABHSISHEK", "THUVAKUDI", "TRICHY", "TAMIL NADU", 620015, 1),

(3, "DHRUV", "THUVAKUDI", "TRICHY", "TAMIL NADU", 620015, 2);

DELIMITER //

CREATE TRIGGER DELETEPERFORMER

BEFORE DELETE

ON PERFORMERS

FOR EACH ROW

BEGIN

IF (1 = (SELECT COUNT(PERFORMERID)

FROM PERFORMERS

WHERE PERFORMERS.ACTIVITYID = OLD.ACTIVITYID

)

) THEN

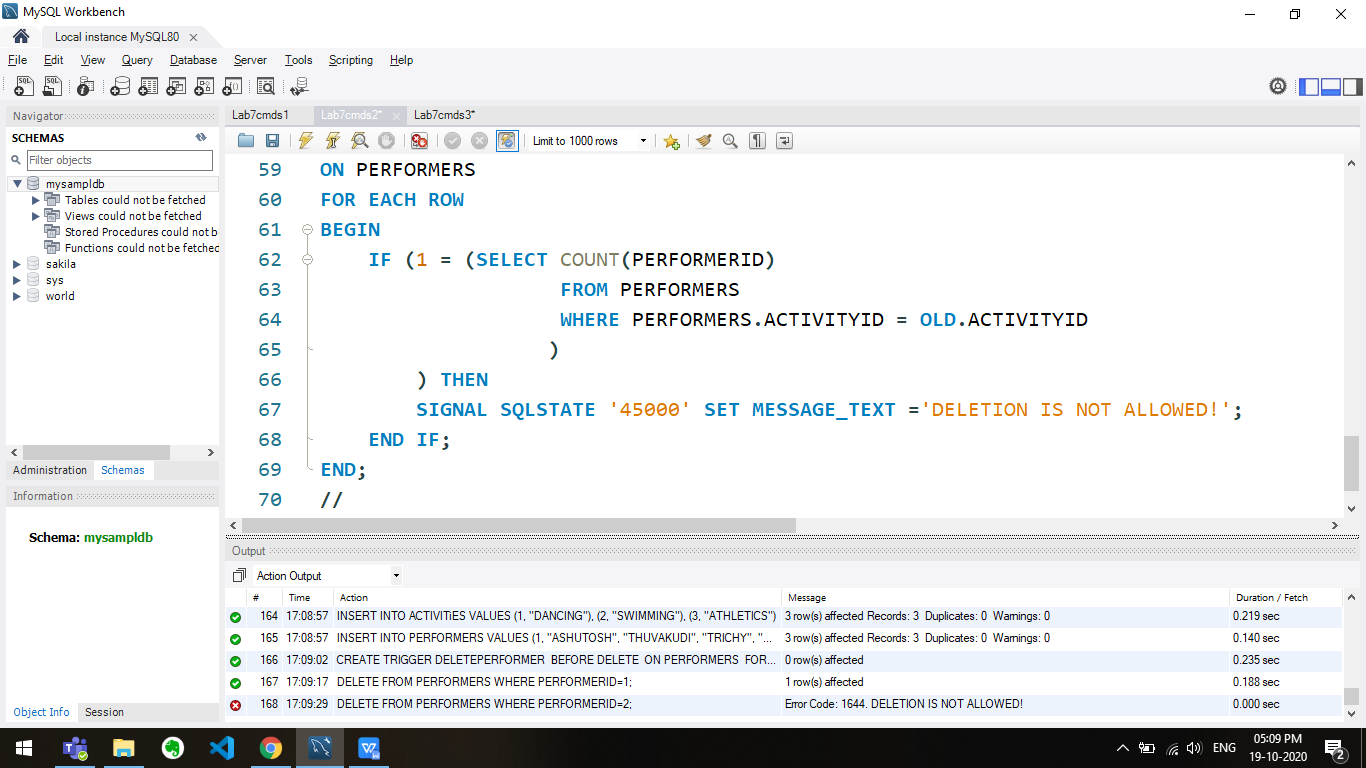
SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT ='DELETION IS NOT ALLOWED!';

END IF;

END;

//

DELETE FROM PERFORMERS WHERE PERFORMERID=2;



Q3

USE LAB7;

DROP TABLE IF EXISTS DEPT;

DROP TABLE IF EXISTS EMPLOYEE;

DELETE FROM DEPT;

CREATE TABLE DEPT (

DEPT\_NO INT,

DEPT\_NAME VARCHAR(20)

);

CREATE TABLE EMPLOYEE (

EMP\_NO INT,

DEPT\_NO INT,

JOB VARCHAR(20),

SALARY INT

);

-- 3.1

DELIMITER //

CREATE TRIGGER CHKNULL

BEFORE INSERT

ON DEPT

FOR EACH ROW

BEGIN

IF(new.DEPT\_NO IS NULL)

THEN SET new.DEPT\_NO=0;

END IF;

END;

//

INSERT INTO DEPT

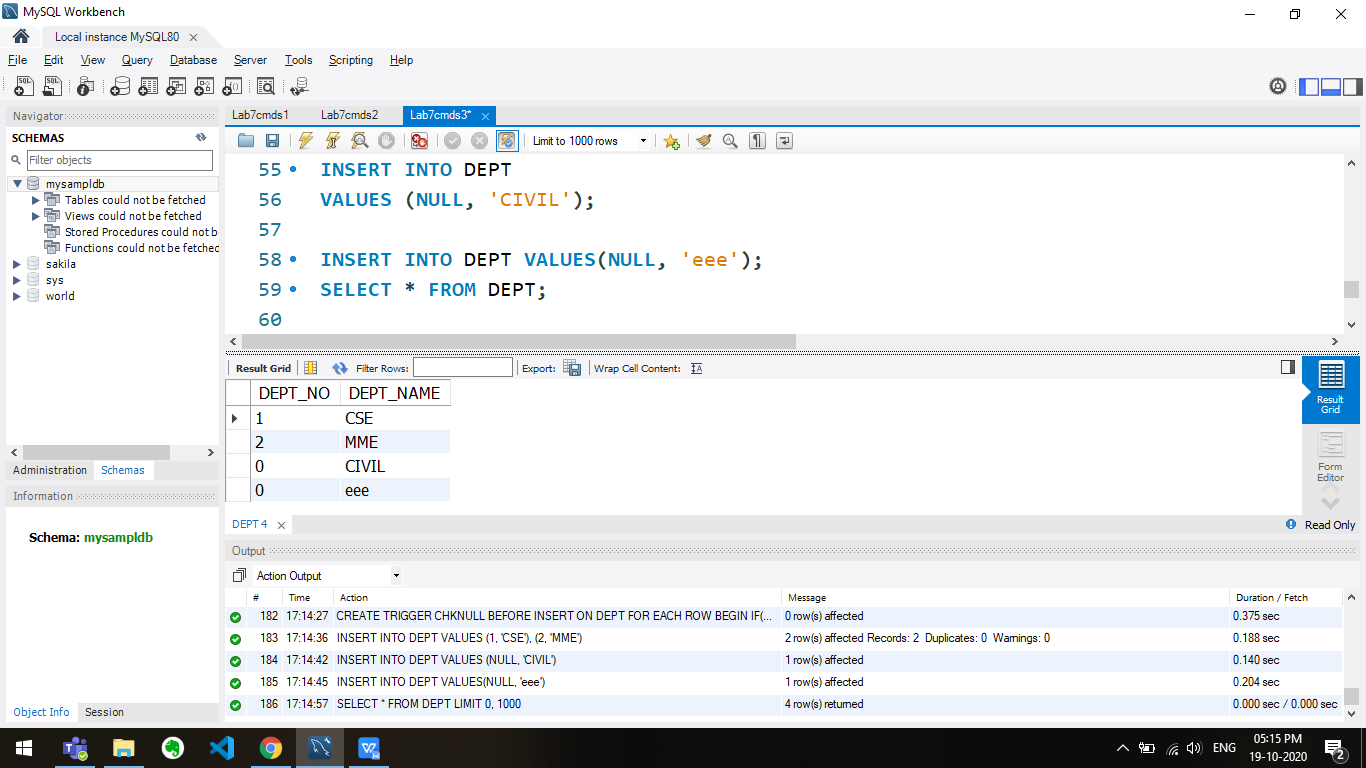
VALUES (1, 'CSE'),

(2, 'MME');

INSERT INTO DEPT

VALUES (NULL, 'CIVIL');

INSERT INTO DEPT VALUES(NULL, 'eee');



-- 3.2

CREATE TRIGGER ONDLT

AFTER DELETE

ON DEPT

FOR EACH ROW

DELETE FROM EMPLOYEE WHERE EMPLOYEE.DEPT\_NO = OLD.DEPT\_NO;

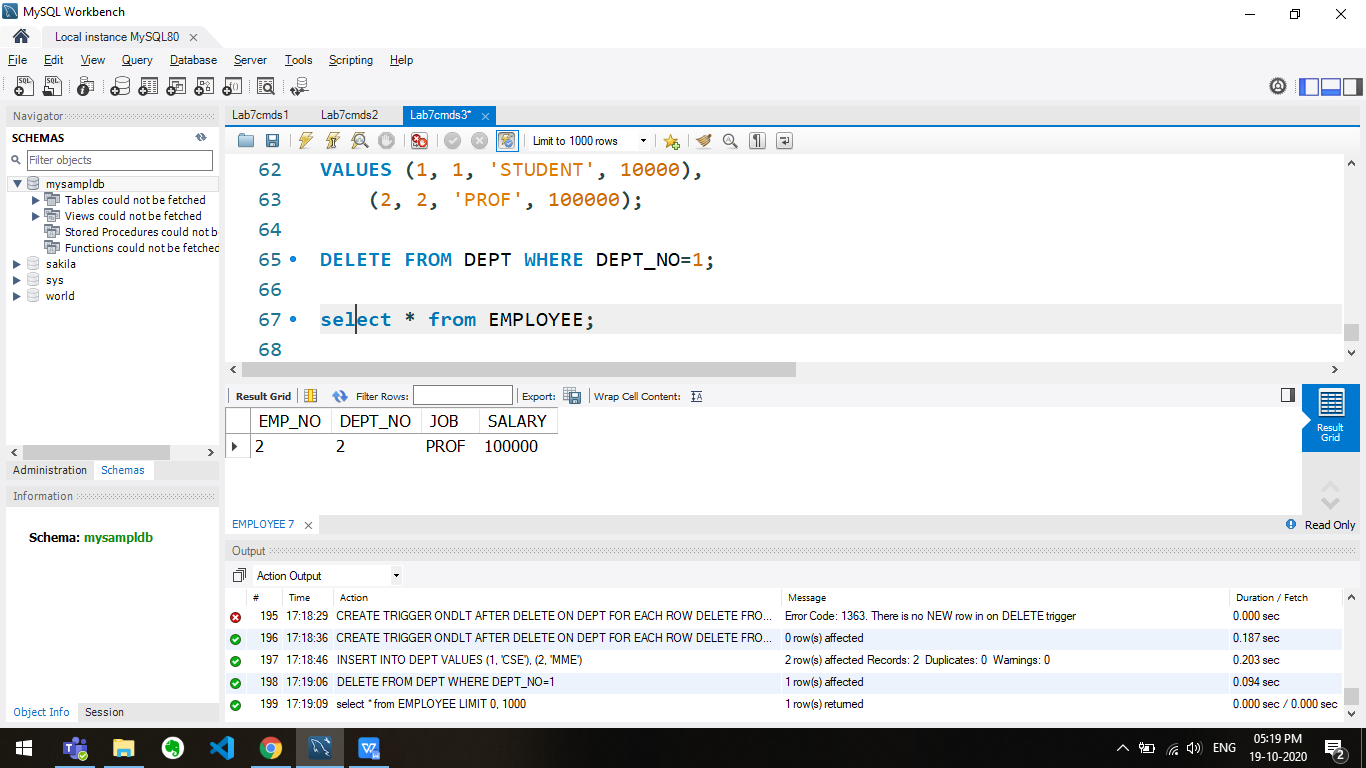
INSERT INTO EMPLOYEE

VALUES (1, 1, 'STUDENT', 10000),

(2, 2, 'PROF', 100000);

DELETE FROM DEPT WHERE DEPT\_NO=1;

select \* from dept;



-- 3.3

CREATE TRIGGER INSUPD

BEFORE INSERT

ON DEPT

FOR EACH ROW

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT ='AUTHORIZATION ERROR';

