```
-- SHAMITHA PICHIKA
-- HW5

CREATE TABLE STUDENTS(
     STU_ID CHAR(11) NOT NULL UNIQUE,
    F_NAME VARCHAR(20) NOT NULL,
    L_NAME VARCHAR(20) NOT NULL,
    STU_GENDER CHAR(1) NOT NULL,
    DOB DATE NOT NULL,
    PRIMARY KEY (STU_ID)
);

CREATE TABLE COURSES(
     COURSE_CODE VARCHAR(6) NOT NULL UNIQUE,
     COURSE_NAME VARCHAR(70) NOT NULL,
     COURSE_LEVEL CHAR(2) NOT NULL,
     COURSE_CREDITS INT NOT NULL,
     PRIMARY KEY (COURSE_CODE)
);

CREATE TABLE REGISTRATION(
     STU_ID CHAR(11) NOT NULL,
    COURSE_CODE VARCHAR(6) NOT NULL,
    GRADE DECIMAL(2, 1) NOT NULL,
    PRIMARY KEY (STU_ID, COURSE_CODE),
    FOREIGN KEY(STU_ID) REFERENCES STUDENTS(STU_ID) ON UPDATE CASCADE,
    FOREIGN KEY(COURSE_CODE) REFERENCES COURSES(COURSE_CODE) ON UPDATE CASCADE
);

INSERT INTO STUDENTS VALUES
('861103-2438', 'ADAM', 'JOHNSON', 'M', '1990-10-01'),
('911212-1746', 'EVA', 'SMITH', 'F', '1991-08-20'),
('950829-1848', 'ANNA', 'WASHINGTON', 'F', '1993-09-06'),
('123456-0980', 'ERIC', 'ALONZO', 'F', '1990-05-26'),
('908023-2456', 'BO', 'EK', 'M', '1992-03-15'),
('098735-3456', 'DANNY', 'GOSS', 'M', '1992-02-01'),
('124345-3430', 'MIKE', 'WHITE', 'M', '1995-06-10'),
('124568-1345', 'EMILY', 'YOUNG', 'F', '1995-04-28'),
('908409-0010', 'CATHY', 'LEE', 'F', '1993-10-06'),
('124587-9088', 'BEN', 'WOO', 'M', '1992-11-30'),
('120953-0909', 'ANNA', 'WASHINGTON', 'F', '1990-10-09'),
('120449-1008', 'JOHN', 'GOSS', 'M', '1995-10-06');

INSERT INTO COURSES VALUES
('CS056', 'DATABASE SYSTEMS', 'G1', 5),
('CS010', 'C++', 'U1', 5),
('ENG111', 'ENGLISH', 'U1', 3),
('FIN052', 'FINANCE', 'G1', 5),
('PHY210', 'PHYSICS', 'U2', 5),
('CHE350', 'CHEMISTRY', 'U3', 3),
('BIO001', 'BIOLOGY', 'U1', 3),
('CS052', 'OPERATING SYSTEMS', 'G1', 5);

INSERT INTO REGISTRATION VALUES
('861103-2438', 'CS056', 4),
('861103-2438', 'CS010', 4),
('861103-2438', 'PHY210', 3.5),
('911212-1746', 'ENG111', 2),
('950829-1848', 'CHE350', 3),
```

```
('950829-1848', 'BIO001', 2.5),
('123456-0980', 'CS052', 3.5),
('123456-0980', 'CS056', 4),
('908023-2456', 'PHY210', 3.0),
('908023-2456', 'CS056', 1.0),
('908023-2456', 'CS010', 2.0),
('124345-3430', 'FIN052', 2.5),
('124345-3430', 'CHE350', 4),
('908409-0010', 'CS052', 2),
('124587-9088', 'BIO001', 4),
('124587-9088', 'CS052', 3.5);


-- Part 1:
-- Question 1:
SELECT DISTINCT COURSES.COURSE_NAME, COURSES.COURSE_CREDITS
FROM STUDENTS
JOIN REGISTRATION ON STUDENTS.STU_ID - REGISTRATION.STU_ID
JOIN COURSES ON REGISTRATION.COURSE_CODE = COURSES.COURSE_CODE
WHERE REGISTRATION.STU_ID = '861103-2438';


-- Question 2:
SELECT STUDENTS.STU_ID, STUDENTS.F_NAME, STUDENTS.L_NAME,
SUM(COURSES.COURSE_CREDITS)
FROM STUDENTS
JOIN REGISTRATION ON STUDENTS.STU_ID = REGISTRATION.STU_ID
JOIN COURSES ON REGISTRATION.COURSE_CODE = COURSES.COURSE_CODE
GROUP BY REGISTRATION.STU_ID, STUDENTS.F_NAME, STUDENTS.L_NAME;


-- Question 3:
CREATE VIEW STUDENTS_AVG_GRADE AS
SELECT STUDENTS.STU_ID, STUDENTS.F_NAME, STUDENTS.L_NAME,
AVG(REGISTRATION.GRADE) AS AVG_GRADE
FROM STUDENTS
JOIN REGISTRATION ON STUDENTS.STU_ID = REGISTRATION.STU_ID
GROUP BY REGISTRATION.STU_ID, STUDENTS.F_NAME, STUDENTS.L_NAME;

SELECT STU_ID, F_NAME, L_NAME, AVG_GRADE
FROM STUDENTS_AVG_GRADE
WHERE AVG_GRADE = (SELECT MAX(AVG_GRADE) FROM STUDENTS_AVG_GRADE);


-- Question 4:
SELECT DISTINCT STUDENTS.F_NAME, STUDENTS.L_NAME
FROM STUDENTS
JOIN REGISTRATION ON STUDENTS.STU_ID = REGISTRATION.STU_ID
JOIN COURSES ON REGISTRATION.COURSE_CODE = COURSES.COURSE_CODE
WHERE COURSES.COURSE_NAME = 'DATABASE SYSTEMS';


-- Question 5:
SELECT DISTINCT STUDENTS.F_NAME, STUDENTS.L_NAME
FROM STUDENTS
JOIN REGISTRATION R1 ON STUDENTS.STU_ID = R1.STU_ID
JOIN COURSES C1 ON R1.COURSE_CODE = C1.COURSE_CODE
JOIN REGISTRATION R2 ON STUDENTS.STU_ID = R2.STU_ID
JOIN COURSES C2 ON R2.COURSE_CODE = C2.COURSE_CODE
WHERE C1.COURSE_NAME = 'DATABASE SYSTEMS'
AND C2.COURSE_NAME = 'C++';


-- Question 6:
SELECT DISTINCT STUDENTS.STU_ID, STUDENTS.F_NAME, STUDENTS.L_NAME,
```

```sql
COURSES.COURSE_NAME, REGISTRATION.GRADE
FROM STUDENTS
LEFT JOIN REGISTRATION ON STUDENTS.STU_ID = REGISTRATION.STU_ID
LEFT JOIN COURSES ON REGISTRATION.COURSE_CODE = COURSES.COURSE_CODE;

-- Question 7:
SELECT DISTINCT STUDENTS.F_NAME, STUDENTS.L_NAME, COURSES.COURSE_CODE
FROM STUDENTS
JOIN REGISTRATION ON STUDENTS.STU_ID = REGISTRATION.STU_ID
JOIN COURSES ON REGISTRATION.COURSE_CODE = COURSES.COURSE_CODE
WHERE COURSES.COURSE_CODE LIKE 'CS%';

-- Part 2:

CREATE TABLE INVENTORY (
ITEMID VARCHAR(20) PRIMARY KEY, NAME VARCHAR(30),
PRICE DECIMAL(6, 2),
QUANTITY INT
);

CREATE TABLE TRANSACTION (
TRANSID INT AUTO_INCREMENT PRIMARY KEY,
ITEMID VARCHAR(20),
QUANTITY INT,
TIME DATETIME,
FOREIGN KEY (ITEMID) REFERENCES INVENTORY(ITEMID)
);

CREATE TABLE INVENTORY_HISTORY (
ID INT AUTO_INCREMENT PRIMARY KEY,
ITEMID VARCHAR(20),
ACTION VARCHAR(20),
OLDPRICE DECIMAL (6,2),
TIME DATETIME,
FOREIGN KEY (ITEMID) REFERENCES INVENTORY(ITEMID)
);

-- Question 1:
DELIMITER |
CREATE TRIGGER INSERT_INVENTORY AFTER INSERT ON INVENTORY
FOR EACH ROW
BEGIN INSERT INTO INVENTORY_HISTORY(ITEMID, ACTION, OLDPRICE, TIME)
VALUES (NEW.ITEMID, 'ADD AN ITEM', NULL, NOW());
END
|
DELIMITER ;

INSERT INTO INVENTORY(ITEMID, NAME, PRICE, QUANTITY)
VALUES('12345', 'SOME ITEM', 7.11, 10);

-- Question 2:

DELIMITER |
CREATE TRIGGER CHANGE_QUANTITY AFTER INSERT ON TRANSACTION
FOR EACH ROW
BEGIN UPDATE INVENTORY
SET QUANTITY = QUANTITY - NEW.QUANTITY
WHERE ITEMID = NEW.ITEMID;
END;
```

```
|
DELIMITER ;

INSERT INTO INVENTORY(ITEMID, NAME, PRICE, QUANTITY)
VALUES('IWATCH', 'APPLE WATCH', 7.11, 10);

INSERT INTO TRANSACTION (ITEMID, QUANTITY, TIME)
VALUES('IWATCH', 3, NOW());

SELECT * FROM INVENTORY WHERE ITEMID = 'IWATCH';

-- Question 3:

DELIMITER |
CREATE TRIGGER CHANGE_PRICE
BEFORE UPDATE ON INVENTORY
FOR EACH ROW
BEGIN IF OLD.PRICE != NEW.PRICE THEN
INSERT INTO INVENTORY_HISTORY(ITEMID, ACTION, OLDPRICE, TIME)
VALUES (NEW.ITEMID, 'PRICE CHANGE', OLD.PRICE, NOW());
END IF;
END;
|
DELIMITER ;

UPDATE INVENTORY SET PRICE  = 9.99 WHERE ITEMID = '12345';

SELECT * FROM INVENTORY_HISTORY;
```