```
# 1

SELECT COURSENAME, CREDITS FROM REGISTRATION, COURSES
WHERE COURSES.COURSECODE = REGISTRATION.COURSECODE
AND STUDENTID = '861103-2438';

# 2

SELECT REGISTRATION.STUDENTID, FIRSTNAME, LASTNAME, SUM(CREDITS) FROM REGISTRATION,
STUDENTS, COURSES
WHERE STUDENTS.STUDENTID = REGISTRATION.STUDENTID
AND REGISTRATION.COURSECODE = COURSES.COURSECODE
GROUP BY STUDENTID;

# 3

CREATE VIEW STUDENT_GPA AS SELECT REGISTRATION.STUDENTID, LASTNAME, FIRSTNAME,
AVG(GRADE) AS GPA FROM REGISTRATION, STUDENTS
WHERE REGISTRATION.STUDENTID = STUDENTS.STUDENTID
GROUP BY STUDENTID;

SELECT * FROM STUDENT_GPA
WHERE GPA = (SELECT MAX(GPA) FROM STUDENT_GPA);

# 4

SELECT FIRSTNAME, LASTNAME FROM STUDENTS, REGISTRATION, COURSES
WHERE COURSENAME = 'Database Systems'
AND STUDENTS.STUDENTID = REGISTRATION.STUDENTID
AND COURSES.COURSECODE = REGISTRATION.COURSECODE;

# 5

SELECT FIRSTNAME, LASTNAME FROM REGISTRATION, STUDENTS, COURSES
WHERE COURSENAME = 'C++'
AND STUDENTS.STUDENTID IN (SELECT REGISTRATION.STUDENTID FROM REGISTRATION,
STUDENTS, COURSES
WHERE COURSENAME = 'Database Systems')
AND STUDENTS.STUDENTID = REGISTRATION.STUDENTID
AND COURSES.COURSECODE = REGISTRATION.COURSECODE;


# 6

SELECT STUDENTID, FIRSTNAME, LASTNAME, COURSENAME, GRADE FROM STUDENTS
LEFT JOIN REGISTRATION USING (STUDENTID)
LEFT JOIN COURSES USING (COURSECODE);

# 7

SELECT FIRSTNAME, LASTNAME, COURSENAME FROM STUDENTS, REGISTRATION, COURSES
WHERE REGISTRATION.COURSECODE LIKE 'CS%%%'
AND STUDENTS.STUDENTID = REGISTRATION.STUDENTID
AND COURSES.COURSECODE = REGISTRATION.COURSECODE
ORDER BY FIRSTNAME;

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)
);


# 1

DELIMITER |
CREATE TRIGGER insert_inventory AFTER INSERT ON Inventory
FOR EACH ROW BEGIN
INSERT INTO Inventory_history
SET action ='add an item',
itemid = NEW.itemid,
time = NOW(),
oldprice = null;
END |
DELIMITER ;

# 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 ;


# 3

DELIMITER $$
CREATE TRIGGER change_price BEFORE UPDATE ON Inventory
FOR EACH ROW BEGIN
INSERT INTO Inventory_history
SET action ='price change',
itemid = OLD.itemid,
time = NOW(),
```

```
oldprice = OLD.price;
END $$
DELIMITER ;
```