BIRZEIT UNIVERSITY

COMP333-Database Systems

Date: Tuesday 24/11/2015 **Midterm Exam - Fall 2015/2016** **Time: 90 minutes**

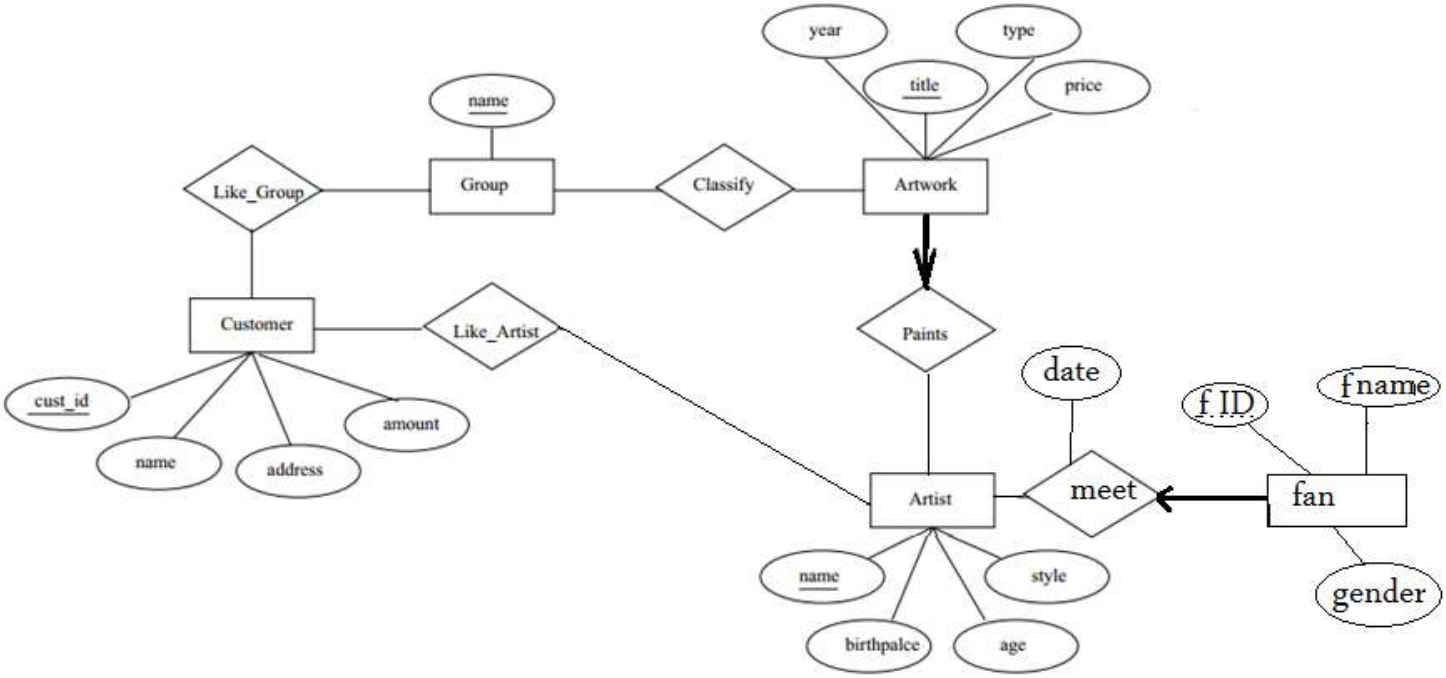
Student Name: **Student ID:**

Instructor: ☐ Mr. Wahbeh Mousa ☐ Dr. Ahmad Abusnaina

Question #1: [25 points]

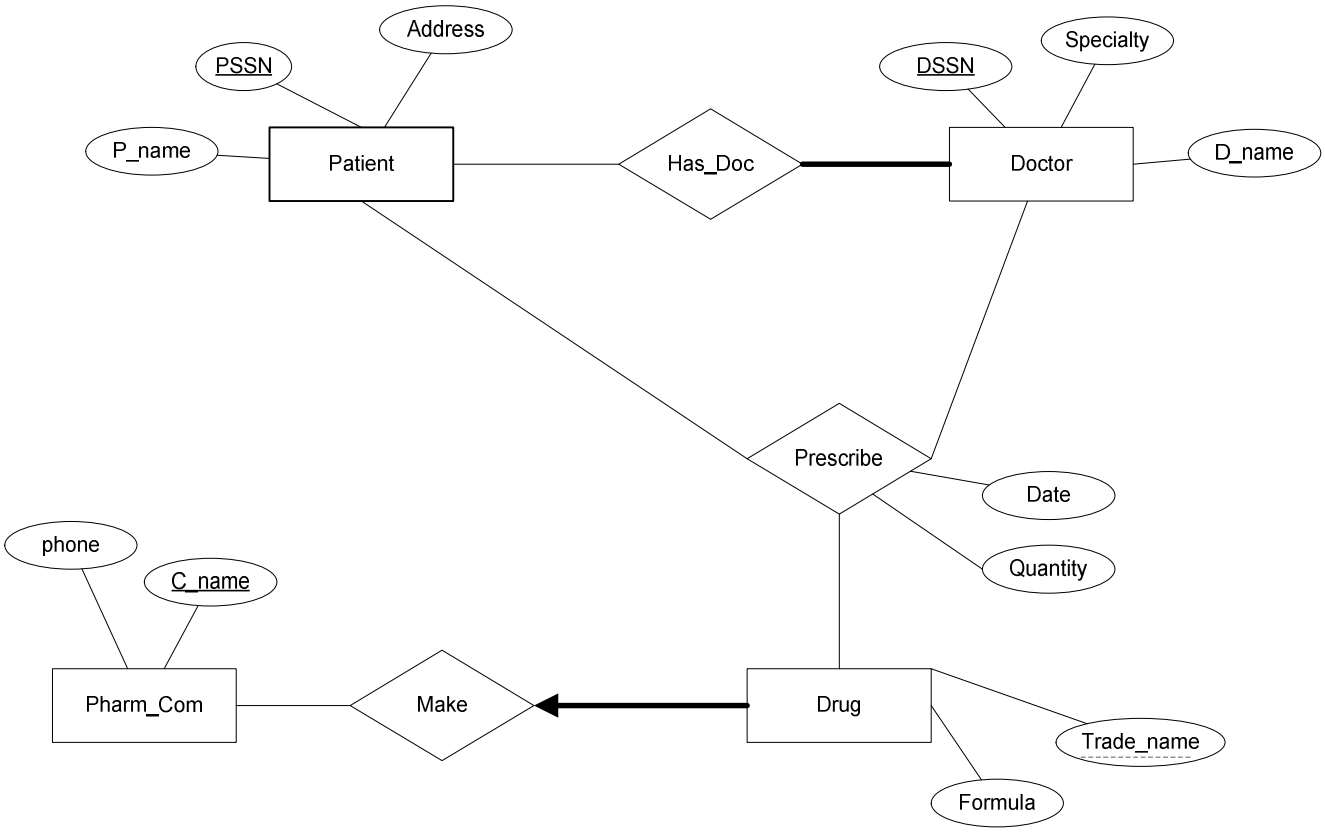
The Association of Arts asked you to build a computerized system to keep track about the artists, their works, fans and customers. You decided to do this business using the concepts of database, you named it *ArtBase*. So, the core of *ArtBase* is a database with a schema that captures all the information the association needs to maintain. **Draw the ER diagram for the *ArtBase*.** Here the information you gather:

- Every artist is identified by his/her ID, name, birthplace, and style of art.
- For each piece of artwork the artist performed, the year it was performed, its unique title, its type of art (e.g. painting, lithograph, sculpture, photograph), and its price must be stored. We know that every artwork is performed by only one artist.
- Pieces of artwork are also classified into groups of various kinds (e.g. Portraits, works by Picasso, or works of the 19th century); a given piece of artwork may belong to more than one group.
- Each group is identified by a name that describes the group.
- The artist has fans. So, the fans can meet with their artist. The date of meeting is to be recorded. Before meeting, each fan should be registered in *ArtBase* by his/her unique ID, name, and gender. Assume that once the artist is retired (or deleted), you do not need to keep track any of his/her fans any longer.
- Finally, *ArtBase* keeps information about the customers. The customer has ID, name, address, total amount of dollars spent. The customer can like many artists, and can like a single group of artwork.


year
type
title
price
name
Like_Group
Group
Classify
Artwork
Customer
Like_Artist
Paints
date
f_ID
fname
cust_id
amount
name
address
Artist
meet
fan
name
style
gender
birthpalce
age

Question #2: [35 points]

I-Translate the following ER diagram into a relational schema using the create SQL statements. [25 points]




```
CREATE  TABLE  Patient ( pssn CHAR(11),
                         p_name CHAR(20),
                         address CHAR(20),
                         PRIMARY KEY (pssn))

CREATE  TABLE  Doctor_Has ( dssn CHAR(11),
                            d_name CHAR(20),
                            specialty CHAR(20),
                            pssn CHAR(20) NOT NULL,
                            PRIMARY KEY (dssn)
                            FOREIGN KEY (pssn) REFERENCES Patient)

CREATE  TABLE  Make_Drug (trade name CHAR(20),
                          formula CHAR(30),
                          c_name CHAR(11),
                          PRIMARY KEY (trade name, c_name),
                          FOREIGN KEY (c_name) REFERENCES
                          Pharm_Com ON DELETE CASCADE)
```

```
CREATE  TABLE  Prescribe ( pssn CHAR(11),
                           dssn CHAR(11),
                           date CHAR(11),
                           quantity INTEGER,
                           trade name CHAR(20),
                           c_name CHAR(11),
                          PRIMARY KEY (pssn, dssn),
                          FOREIGN KEY (pssn) REFERENCES Patient,
                          FOREIGN KEY (dssn) REFERENCES Doctor_Has,
                          FOREIGN KEY (trade name, c_name)
                          REFERENCES Make_Drug)


CREATE  TABLE  Pharm_Com (c_name CHAR(20),
                          phone CHAR(20),
                          PRIMARY KEY (c_name))
```

II- Use appropriate SQL commands to perform the following actions on the database. [10 points]

- Add the following person to table patient: pssn=222333, name= ABC, address=Palestine.

```
INSERT
INTO Patient (pssn, name, address)
VALUES (222333, 'ABC', 'Palestine')
```

- Delete the doctor whose DSSN is 555.

```
DELETE
FROM Doctor_Has  D
WHERE D.dssn = 555
```

- Change the phone number of ‘DarElshefa’ company from 333222 to 888999.

```
UPDATE Pharm_Com  C
SET C.phone = 888999
WHERE   C. c_name =  ‘DarElshefa’
```

Question #3: [25 points]

Write the following queries in **SQL** by considering the following relational schema. An employee can work in more than one department. The *pct_time* field of the Works relation shows the percentage of time that a given employee works in a given department. The *managerid* field in Dept is foreign key refers to *eid* from Emp.

Emp(*eid:* integer, *ename:* string, *age:* integer, *salary:* real)
Works(*eid:* integer, *did:* integer, *pct_time:* integer)
Dept(*did:* integer, *dname:* string, *budget:* real, *managerid:* integer)

1. Find the names and ages of each employee who works in both the 'Hardware' department and the 'Software' department.

```
SELECT  E.ename, E.age
FROM    Emp E, Works W1, Works W2, Dept D1, Dept D2
WHERE  E.eid = W1.eid AND W1.did = D1.did AND D1.dname = 'Hardware' AND
          E.eid = W2.eid AND W2.did = D2.did AND D2.dname = 'Software'
```

2. Find the names of employee whose salary exceeds the budget of all of the departments that he or she works in.

```
SELECT E.ename
FROM Emp E
WHERE E.salary > ALL (SELECT D.budget
                      FROM Dept D, Works W
                      WHERE E.eid = W.eid AND D.did = W.did)
```

3. Find the *enames* of managers who manage the departments with the largest budgets.

```
SELECT E.ename
FROM    Emp E
WHERE E.eid IN (SELECT D.managerid
                FROM Dept D
                WHERE D.budget = (SELECT MAX (D2.budget)
                                  FROM Dept D2))
```

4. Find the names of employees who are older than the oldest employee with a salary greater than 200.

```
SELECT E.ename
FROM Emp E
WHERE E.age > (SELECT MAX(E2.age)
               FROM Emp E2
               WHERE E2.salary =200)
```

5. Find the average age of employees who are at least 25 years for each department name.

```
SELECT   D.name, AVG(E.age) AS avgage
FROM Emp E, Dept D, Works W
WHERE E.age>=25 AND E.eid = W.eid AND D.did = W.did
GROUP BY D.dname
```

Question #4: [15 points]

By considering the following schema, write the following queries in Relational Algebra

Sailors(sid, sname, age, rating)
Boats(bid, bname, color, size)
Reserves(sid, bid, day)

a) Find the sailor names whose age is 25.

$$\pi_{sname}\left(\sigma_{age=25}(sailors)\right)$$

b) Find the colors of boats reserved by 'Osama'.

$$\rho\left(T, Reserves \bowtie \sigma_{sname='Osama'}(sailors)\right)$$
$$\pi_{color}(T \bowtie Boats)$$

c) Find the names boats that were not reserved by any sailor.

$$\pi_{Boats.bname}\left(\pi_{bid,bname}(Boats) - \pi_{Reserves.bid,Boats.bname}(Reserves \bowtie Boats)\right)$$