Midterm Exam #1 DRAFT SOLUTION

General Questions (4 points)

1. Select the type of schema that would be used for each of the following: (3 questions, 1 point each)
   1.1. The hard drives in the data center that stores EECS 484 student information
      a. Physical
      b. Conceptual
      c. External
      d. None of the above

   1.2. The schema students see when registering for EECS 484 on wolverine access.
      a. Physical
      b. Conceptual
      c. External
      d. None of the above

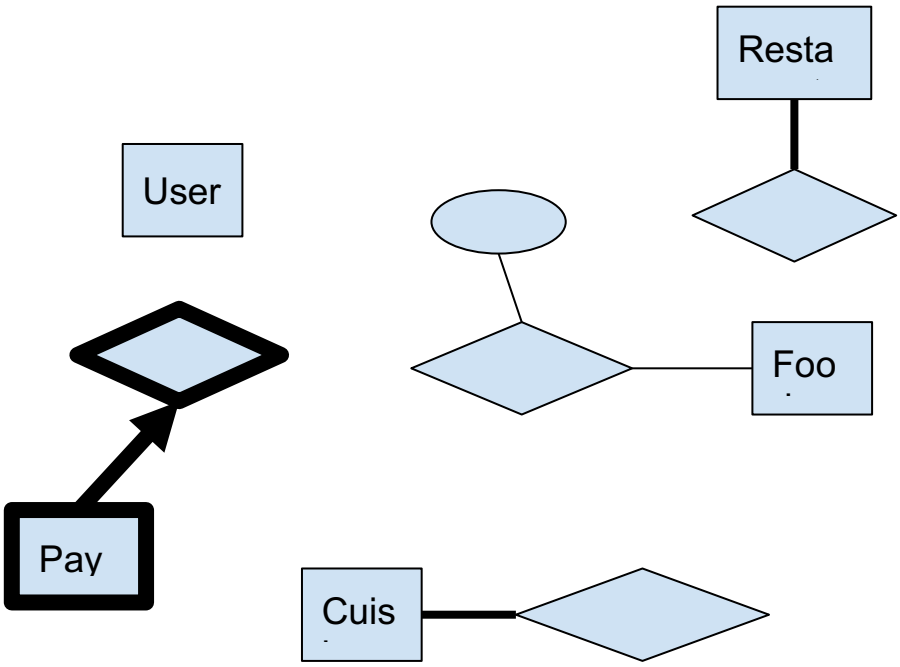
   1.3. The schema the database administrator sees when making new public datasets for EECS 484 Project 1.
      a. Physical
      b. Conceptual
      c. External
      d. None of the above

2. The UofM database recently underwent changes that further removed redundancy in their tables by enforcing that all tables should be in BCNF. Despite this change, you as a user do not notice any difference when using wolverine access. What kind of independence did this change to the UofM database achieve? (1 point)
   a. Physical independence
   b. Logical independence
   c. None of the above

ER Diagrams (32 points)

3. Suppose you are designing an ER diagram for a database that will be used by a food delivery app. The following are constraints that need to be represented in the diagram.
   - Each user must own at least one payment method.
   - A user can order as much food as they want.
   - A food can be contained in at most one cuisine.
   - A food must be made by a restaurant, and no two restaurants can make the same food.
   - A user is uniquely identified by their UID.
   - Payment methods have two attributes: a CardNumber and a type. CardNumber is the partial key.

   Looking at the **incomplete** ER diagram below, answer the following questions. (Tip: Complete the ER diagram using the description above before answering the questions)



3.1. What should the line between *User* and *Owns* be? (2 points)

a. Thin line
b. Thin arrow
c. Thick line
d. Thick arrow

3.2. What should the line between *User* and *Orders* be? (2 points)
   a. Thin line
   b. Thin arrow
   c. Thick line
   d. Thick arrow

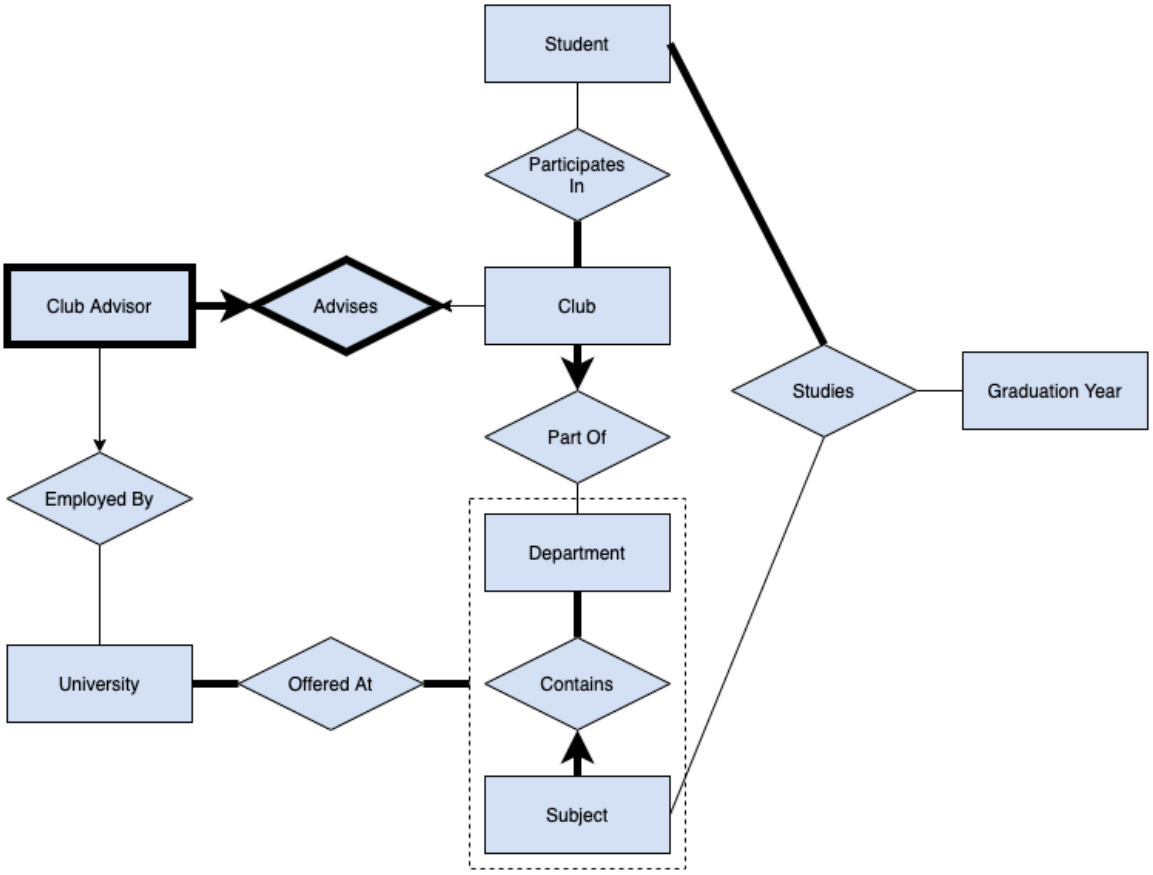
3.3. What should the line between *Food* and *Contains* be? (2 points)
   a. Thin line
   b. Thin arrow
   c. Thick line
   d. Thick arrow

3.4. What should the line between *Food* and *Makes* be? (2 points)
   a. Thin line
   b. Thin arrow
   c. Thick line
   d. Thick arrow

3.5. Having *Payment Method* as a weak entity does **not** indicate which of the following? (3 points)
   a. The primary key of *Payment Method* is (UID, CardNumber)
   b. On the ER diagram, the CardNumber attribute should be underlined with a dotted line
   c. Given a CardNumber, the database administrator can uniquely identify which user owns this *Payment Method*.
   d. If a user deactivates the food app account, all of that user's *Payment Method* information will be deleted as well

3.6. Which of the following is true about the diagram above? (3 points)
   a. Two users cannot own the same CardNumber as their payment method.
   b. Given a record from the Orders table, the app cannot identify which restaurant should be making this food.
   c. A user cannot order the same food twice on different dates.
   d. None of the above

4. Examine the following ER diagram that represents the relationships between various entities involved in clubs at a certain university. Answer the following questions based on the **completed** ER diagram below.



4.1. True or false: A Student may graduate in several different Graduation Years. (2 points)

a. True
b. False

4.2. True or false: It is possible that, in a given Graduation Year, no Students graduate. (2 points)

a. True
b. False

4.3. True or false: The primary key of the Club Advisor entity is a composite of its own partial key along with the Club entity's primary key. (2 points)

a. True
b. False

4.4. True or false: It is possible for a Club to have no Club Advisor. (2 points)

a. True
b. False

4.5. True or false: A University must employ at least one Club Advisor. (2 points)
   a. True
   b. False

4.6. True or false: Every department contains at least one Subject that is being studied by at least one Student. (2 points)
   a. True
   b. False

4.7. True or false: It is possible for a student to study a Subject that is not offered at any University. (2 points)
   a. True
   b. False

4.8. Which of the following is true? (4 points)
   a. A Student can only participate in a Club that is part of the Department that contains the subject that they study.
   b. A Student can study multiple Subjects only if the Subjects are contained within the same Department.
   c. A Student can participate in multiple Clubs only if the Clubs are part of the same Department.
   d. A Student can study multiple Subjects only if the Graduation Year for these Subjects are all the same.
   e. None of the above.

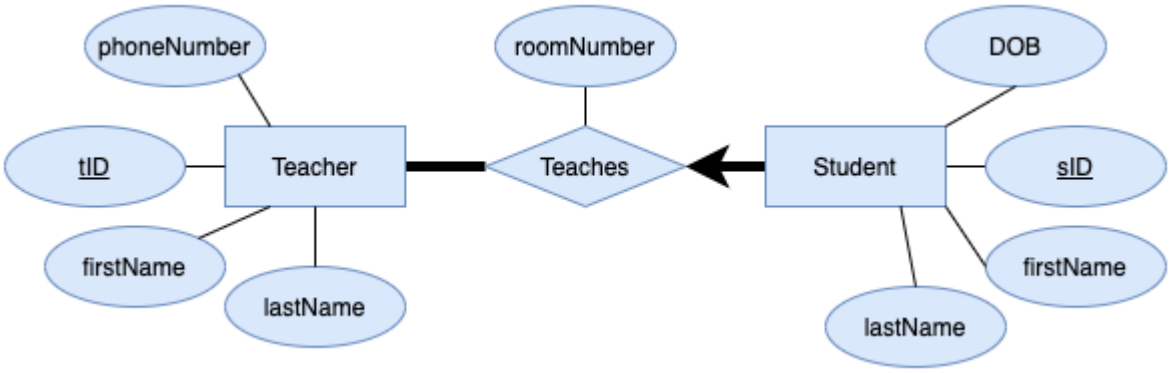
SQL (31 points)

5. What condition does this SQL schema satisfy? (3 points)

```
CREATE TABLE TableA (
    PID NUMBER NOT NULL,
    QID NUMBER,
    PRIMARY KEY (PID, QID)
);
```

   a. PID cannot be NULL
   b. QID cannot be NULL
   c. (PID, QID) must be unique
   d. Both a and c
   e. All a, b, and c
   f. None of the above

6. Consider the following ER diagram and constraints:



- The firstName and lastName of each Teacher are required, and the combination of firstName and lastName for each Teacher must be unique
- The phoneNumber of each Teacher is an optional field
- The DOB, firstName, and lastName of each Student are required fields
- The types of each field can be anything of your choice (e.g. NUMBER, INTEGER, VARCHAR(100), etc).

Using SQL, create the **least redundant** set of``tables needed to capture all information from the ER diagram and as many constraints as possible. (8 points)

```
CREATE TABLE Teacher (
   tID INTEGER,
   firstName VARCHAR2(100) NOT NULL,
   lastName VARCHAR2(100) NOT NULL,
   phoneNumber VARCHAR2(100),
   PRIMARY KEY(tID),
   UNIQUE(firstName, lastName)
);

CREATE TABLE Teaching_Student (
   tID INTEGER NOT NULL,
   sID INTEGER,
   roomNumber INTEGER,
   DOB INTEGER NOT NULL,
   firstName VARCHAR2(100) NOT NULL,
   lastName VARCHAR2(100) NOT NULL,
   PRIMARY KEY(sID),
   FOREIGN KEY(tID) REFERENCES Teacher(tID)
);
```

More space is available on the next page

7. Consider a database consisting of the following tables. Primary keys are underlined and attributes with the same name are foreign keys. Assume that all attributes are NOT NULL.

- Animals(AnimalID, AnimalName, ScientificName, Genus)
- Zoo(ZooID, ZooName, ZooAddress)
- NumberOfAnimals(ZooID, AnimalID, AnimalCount)
- TourGroups(TourID, TourName)
- Visited(TourID, ZooID, Times)

The Animal*Count* attribute in the NumberOfAnimals table refers to the number of animals with AnimalID within the zoo with ZooID. The Animal*Count* can be zero.
The *Times* attribute in the Visited table refers to the number of times the tour group with TourID has visited the zoo with ZooID.

For the following subproblems, write an SQL code following the instructions. You are free to create additional views to help with your query. You do not have to worry about dropping these views.

7.1. Write a SQL query that returns the zoo IDs and the names of all zoos that have more than 5 animals with AnimalName “Tiger”. Results should be in descending order according to the zoo ID with no duplicates. Hint: You should not need to use GROUP BY or COUNT(*) for this problem. (5 points)

```
SELECT DISTINCT Z.ZooID, Z.ZooName
FROM Zoo Z
JOIN NumberOfAnimals N ON Z.ZooID = N.ZooID
JOIN Animals A ON A.AnimalID = N.AnimalID
WHERE N.AnimalCount > 5 AND A.AnimalName = 'Tiger'
ORDER BY Z.ZooID DESC;
```

The above answer is what the person setting the problem was thinking of. But on deeper thought, the answer is not quite right. It is possible that there are two animals, both named 'Tiger', since AnimalName not a unique attribute. Consider a zoo with 3 animals with one ID and 4 animals with another ID and both kinds of animals are named 'Tiger'. Then zoo has 7 animals named Tiger, but the above answer would exclude such a zoo.

Below seems to be a more accurate answer that handles the above situation. It does require GROUP BY, contrary to the instructions.

```
SELECT Z.ZooID, Z.ZooName FROM
Zoo Z, Animals A, NumberofAnimals N WHERE
Z.ZooID = N.ZooID AND  A.AnimalID = N.AnimalID AND
A.AnimalName = 'Tiger'
GROUP BY Z.ZooID, Z.ZooName
HAVING SUM(N.NumberofAnimals) > 5;
```

7.2. Write a SQL query that returns the AnimalID and AnimalName of animals, along with the number of zoos that have at least one of that particular animal. Results should be in descending order of the number of zoos that have this particular animal. Note: The query does not need to return an animal if the number of zoos that have this particular animal is zero. (5 points)

```
SELECT A.AnimalID, A.AnimalName, COUNT(*)
FROM NumberOfAnimals N, Animals A
WHERE  N.AnimalID = A.AnimalID AND N.AnimalCount > 0
GROUP BY N.AnimalID, A.AnimalName
ORDER BY COUNT(*) DESC;
```

Another solution:

```
SELECT A.AnimalID, A.AnimalName, COUNT(*)
FROM NumberOfAnimals N
INNER JOIN Animals A ON N.AnimalID = A.AnimalID AND
WHERE N.AnimalCount > 0
GROUP BY N.AnimalID, A.AnimalName
ORDER BY COUNT(*) DESC;
```

Identical information is provided here again for convenience

Primary keys are underlined and attributes with the same name are foreign keys. Assume that all attributes are NOT NULL.

- Animals(AnimalID, AnimalName, ScientificName, Genus)
- Zoo(ZooID, ZooName, ZooAddress)
- NumberOfAnimals(ZooID, AnimalID, AnimalCount)
- TourGroups(TourID, TourName)
- Visited(TourID, ZooID, Times)

The Animal*Count* attribute in the NumberOfAnimals table refers to the number of animals with AnimalID within the zoo with ZooID. The Animal*Count* can be zero.
The *Times* attribute in the Visited table refers to the number of times the tour group with TourID has visited the zoo with ZooID.

7.3. Write a SQL query that creates a view BestZoos. In the BestZoos views, it should have the IDs, names, addresses, and the total number of animals they have. (5 points)

```
CREATE VIEW BestZoos AS
SELECT Z.ZooID, Z.ZooName, Z.ZooAddress, Sum(N.AnimalCount)
FROM Zoo Z
INNER JOIN NumberOfAnimals N ON Z.ZooID = N.ZooID
GROUP BY Z.ZooID, Z.ZooName, Z.ZooAddress;



Another solution:

CREATE VIEW BestZoos AS
SELECT Z.ZooID, Z.ZooName, Z.ZooAddress, Temp.total
FROM Zoo Z
JOIN (
    SELECT ZooID, SUM(AnimalCount) AS total
    FROM NumberOfAnimals N
    GROUP BY ZooID
) Temp ON Z.ZooID = Temp.ZooID;
```

7.4. We are trying to merge tour groups together. Find the pairs of tourIDs that have visited the same zoo the same number of times. Do not return duplicate results. Have the first TourID renamed as T1 and the second TourID renamed as T2. To remove duplicate results, make T1 be smaller than T2. (5 points).

```
SELECT DISTINCT V1.TourID AS T1, V2.TourID AS T2
FROM Visited V1
INNER JOIN Visited V2 ON V1.ZooID = V2.ZooID
WHERE V1.TourID < V2.TourID AND V1.Times = V2.Times;
```

Another answer:

```
SELECT V1.TourID AS T1, V2.TourID AS T2
FROM Visited V1,  Visited V2 WHERE V1.ZooID = V2.ZooID
AND  V1.TourID < V2.TourID AND V1.Times = V2.Times;
```

The DISTINCT keyword is not necessary.

Relational Algebra (13 points)

8. For the following questions, consider the following schema:

Player(<u>pid</u>, name)
Relationship(<u>pid, pid2</u>, time, type)

Players may be part of a relationship with other players but cannot be in a relationship with themselves. The relationship type is either 'friendly' or 'enemy'.

Time is of type timestamp, and you may assume that arithmetic and comparison operators will work on it.

You may also assume that if (p1, p2) exists in the table, then (p2, p1) also exists in the table, i.e. this relationship is symmetric. We will use P to denote the Player table and R to denote the Relationship table, and you may use P and R to denote these tables as well.

8.1. Write a relational algebra expression that finds the name of all players that are not an enemy with any other player (5 points)

π_{name} (P ⋈
(π_{pid}(P) – π_{pid}($\sigma_{type='Enemy'}$ R))
)

Another answer:

π_{name} (P ⋈
(π_{pid}(P) – π_{pid2}($\sigma_{type='Enemy'}$ R))
)

Another (more complicated) answer:

$$\rho(P1, Player);\ \rho(P2, Player);\ \rho(R, Relationship);$$

$$\pi_{name}(\pi_{pid,name}(P2)\ -\ \pi_{P1.pid,P1.name}(\sigma_{P1.pid\ =\ R.pid\ \wedge\ R.type\ =\ 'Enemy'}(P1\ X\ R)))$$

More space is available on the next page

8.2. There are many players named 'Josh'. We want to find the pid of players that are enemies of **all** of the Joshs' enemies. Select the correct relational algebra expression below. (5 points)

a. $\pi_{pid2}((\sigma_{type='Enemy'} R) \bowtie (\sigma_{name='Josh'} P))$

b. $\pi_{pid2}(\sigma_{R.type='Enemy' \wedge P.name='Josh'} (R \bowtie P))$

c. $(\pi_{pid, pid2}(\sigma_{type='Enemy'} R)) \ / \ (\sigma_{type='Enemy'} (\sigma_{name='Josh'} (P \bowtie R)))$

d. $(\pi_{pid, pid2} (\sigma_{type='Enemy'} R)) \ / \ (\pi_{pid2} ((\sigma_{name='Josh'} P) \bowtie (\sigma_{type='Enemy'} R)))$

e. None of the above

8.3. Are the two following expressions equivalent? (3 points)

Expression 1: $\sigma_{type='Friendly'}(R) - (\sigma_{type='Friendly'}(R) - \sigma_{time<'2018-01-01'}(R))$

Expression 2: $\sigma_{time<'2018-01-01' \wedge \ type='Friendly'}(R)$

a. True
b. False

Normalization (20 points)

9. Mark the following statements are true or false.

9.1. BCNF implies 3NF, and 3NF implies BCNF (1 point)

a. True
b. False

9.2. Lossless-join, dependency-preserving decomposition of a relationship R into a collection of 3NF relations is always possible (1 point)

a. True
b. False

10. Which of the following is **not** implied by Armstrong's axioms? (2 points)

a. If A→B and B→C, then AB→BC

b. If AC→BC, then A→B

c. If A→B, then AC→BC

d. AB→A

e. None of the above

11. Suppose you have the following relation R with six columns (A, B, C, D, E, F). The set of functional dependencies for this relation is F = {E → AD, CD → F, C → B, BE → C}. What is/are the (minimal) key(s) for table R? **Select all that apply.** (5 points, no partial credit)
    a. AD
    b. BC
    c. BE
    d. ABD
    e. CDE
    f. CE

12. Relation R has five columns (A, B, C, D, E) and its (minimal) keys are CD and AB. answer the following questions.

    12.1. If BE → C is one of relation R's functional dependencies, which of the following does this functional dependency satisfy? (2 points)
    a. BCNF only
    b. 3NF only
    c. Both BCNF and 3NF
    d. Neither BCNF or 3NF

    12.2. If CD → E is one of relation R's functional dependencies, which of the following does this functional dependency satisfy? (2 points)
    a. BCNF only
    b. 3NF only
    c. Both BCNF and 3NF
    d. Neither BCNF or 3NF

    12.3. Is decomposition of R into ABDE and ABC a lossless join decomposition? (2 points)
    a. Yes
    b. No

    12.4. Assume F={CD→ABCDE, AB→ABCDE} are the functional dependencies for the relation R. Which of the following functional dependencies would be lost by the decomposition of R into ABE and ABCD? (5 points)
    a. AB → CE

b. CD → BE

c. AB → BD

d. CD → AB

e. None of the above