W4111 Introduction to Databases
Fall 2016
Midterm Exam Solutions

Closed Book, 1 page notes: 8.5x11" letter paper, both sides
614 Schermerhorn if first letter of your UNI is between a-n (inclusive)
1024 Mudd otherwise
Duration: 75 minutes

Instructor: Eugene Wu

Your Name: Alice H. Acker
Your UNI: aa0000

Problem	Points	Score
0	5	
1	15	
2	20	
3	28	
4	17	
TOTAL	85	

Good Luck!

For regrade requests If we added your score incorrectly, we will correct it in office hours.
If there is an error in the solutions, please let us know with a **private piazza message**.
If you want a regrade of a question, we will *regrade the entire exam* carefully.

(5 points) Values

1. **(2 points)** Pick the top 2 values are most important to you and write "1" or "2" next to them.

   _____ Being good at art
   _____ Creativity
   _____ Relationships with family and friends
   _____ Government or politics
   _____ Independence
   _____ Learning and gaining knowledge
   _____ Athletic ability
   _____ Belonging to a social group (such as your community, racial group, or school club)
   _____ Music
   _____ Career
   _____ Spiritual or religious values
   _____ Sense of humor

   _____ Other: _____

2. **(3 points)** Write 1-3 sentences about why are the values that you selected are important to you.

We accepted pretty much anything. However, thank you for the thoughtful responses.

1 (15 points) Fact Checking with Relational Algebra

It is currently election season and many news organizations fact check candidate statements. Each statement is given a score between 1 and 5 where 1 means "Pants on Fire Lying" and 5 means "Completely True". Your job will be to write **relational algebra statements** using the following SQL schema to express the queries below.

```
CREATE TABLE Cands(
  cid int PRIMARY KEY,
  name text,
  party text,
  CHECK (party IN ('democrat', 'republican', 'other'))
);

CREATE TABLE Stmts(
  sid int PRIMARY KEY,
  cid int NOT NULL REFERENCES Cands,
  nid int NOT NULL REFERENCES NewsOrgs,
  statement text NOT NULL,
  score int NOT NULL,
  CHECK (score > 0 AND score <= 5)
);

CREATE TABLE NewsOrgs(
  nid int PRIMARY KEY,
  name text
);
```

1. **(2 points)** Write a relational algebra expression to compute ids of statements with a score of 4 or higher.

$\pi_{sid}\sigma_{score>=4}(Stmts)$

2. **(3 points)** Write a relational algebra expression to compute the names of democrat candidates that have made completely true (score is 5) statements.

$\pi_{name}((\sigma_{party='democrat'}(Cands)) \bowtie_{cid} (\sigma_{score=5}Stmts))$

3. **(5 points)** Write a relational algebra expression to compute the names of candidates that been fact checked by every news organization.

$\pi_{name}((Cands) \bowtie_{cid} (\pi_{cid,nid}(Stmts)/\pi_{nid}(NewsOrgs)))$

4. **(5 points)** Given the following values for the Cands relation:

   And the following result relation:

   Write a relational algebra statement that will generate the above result.

$\rho(A, Cands)$
$\rho(B, Cands)$
$(A \bowtie_{A.party<>B.partyANDA.cid<B.cidANDA.cid=1} B)$

cid	name	party
1	Hillary	democrat
2	Trump	republican
3	Lincoln	republican
4	Jefferson	republican
5	ScoobyDoo	other

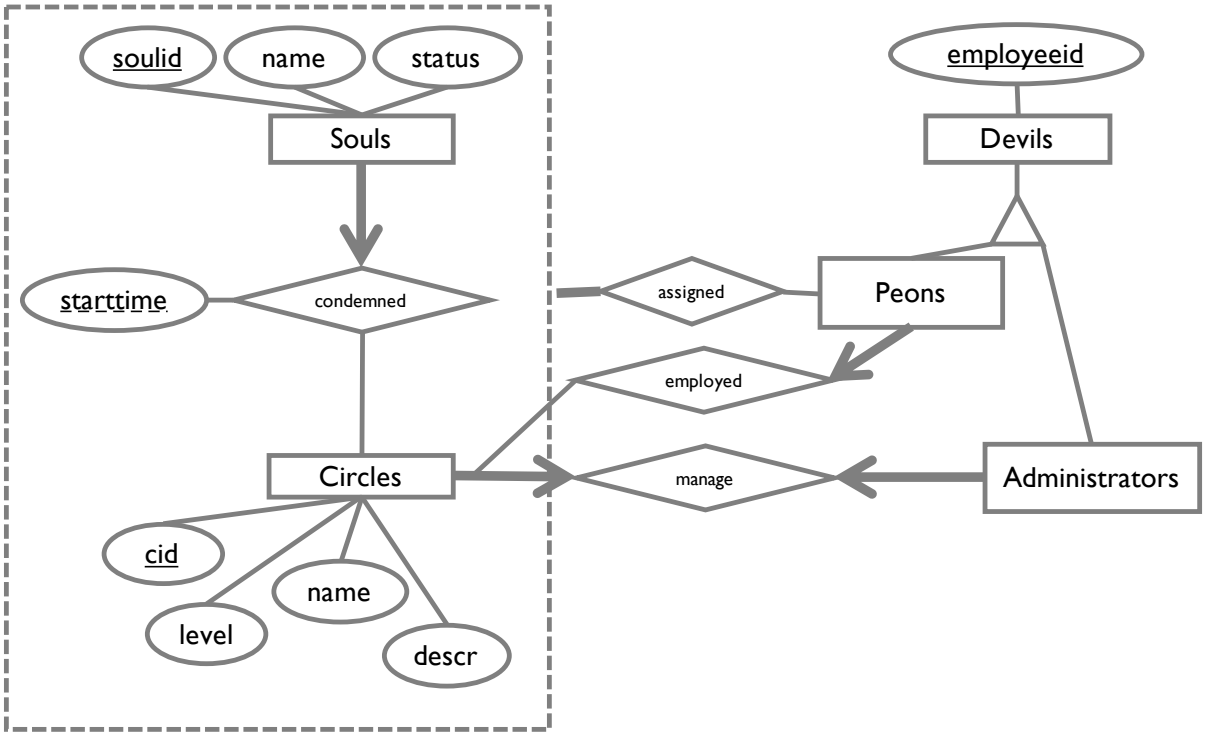
cid	name	party	cid	name	party
1	Hillary	democrat	2	Trump	republican
1	Hillary	democrat	3	Lincoln	republican
1	Hillary	democrat	4	Jefferson	republican
1	Hillary	democrat	5	ScoobyDoo	other

2 (20 points) Entity-Relationship Models

Dante's inferno is a complicated book about a complicated place. A certain professor that has not read the book, however he surmises that the complication stems from the lack of a proper entity-relationship model to adequately capture Hell's administration. In this problem, you will help this professor design a schema to keep track of souls, demons, circles and other pertinent information. Specifically, you will need to track:

1. The name, and status for each soul. A soul can have the status of "condemned" or "saved". In addition, track the timestamp for when each soul arrived in hell.
2. Each Circle of hell including its name, a short description, and a wickedness level (from 1 to 9).
3. At any given time, each soul is condemned to exactly one circle of hell, and each circle may hold multiple condemned souls. However, a soul may be moved to a different circle of hell, so it is important to track the time that a soul was condemned to a particular circle.
4. Devils are employees in hell and are distinguished by a hell-wide employee id. All devils are either a peons or administrators, and cannot be both.
5. Each administrator exclusively manages a circle, and each circle must have exactly one administrator.
6. Peons are employed by exactly one circle. When a soul is condemned in a particular circle, the condemnation must be assigned to at least one peon working in that circle.

Part 1: (10 points) Draw an ER diagram representing your database. Include 1-3 sentences of justification for why you drew it the way you did.



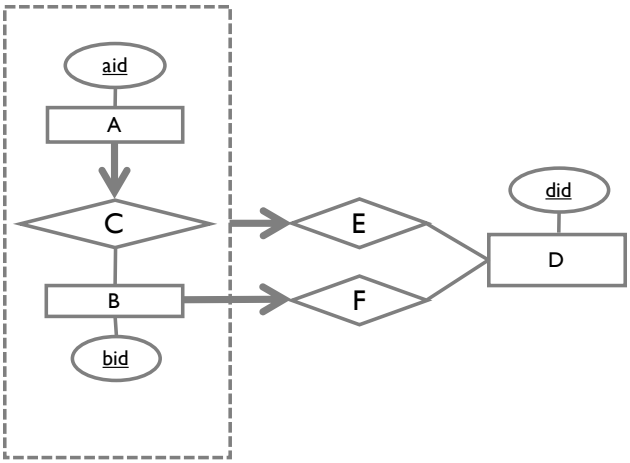
Alternatives and notes:

- We accept basically any reasonable unique id for the entities, but it was an error to forget to specify ids. The question should have specified that it is okay to add ids if that makes things easier.
- Souls should have an at least one relationship with Circle. It should not be an "at most one" relationship because souls can be move between circles.
- The Condemned relationship set should include the start time as part of its key.
- Is-A relationship between devil and peon/administrator
- A Peon has an exactly one relationship with Circle but not vice versa.
- An administrator has an exactly one relationship with Circle and vice versa.
- There is an aggregation around Condemned so that Peons can be assigned to a condemnation. There is an "at-least-one" relationship from Condemned to Peon.
- Equipment needs either an exactly one or at most one relationship with Center. Otherwise you can have a machine at multiple testing centers, which isn't physically possible. The fact that there is a "status" attribute means that each Equipment entity represents one physical machine. The note about multiple machines of the same kind is a distraction and can effectively be ignored. Adding a quantity field isn't quite right, because then multiple machines share the same status.

Close but incorrect alternatives:

- Directly connect Peons and Souls.

Part 2: (10 points) Write the SQL schema to express the following ER diagram. Include < 3 sentences of justification for why you chose the tables you did.



We evaluated if you correctly translated the ER diagram into SQL. A common mistake was to keep the relationship set tables and not merge them when needed.

```
CREATE TABLE B_F(
  bid int PRIMARY KEY,
  did int NOT NULL REFERENCES D
);
CREATE TABLE A_C_E(
  aid int PRIMARY KEY,
  bid int NOT NULL REFERENCES B
  did int NOT NULL REFERENCES D
);
CREATE TABLE D(
  did int PRIMARY KEY
);
```

3 (28 points) SQLSQLSQLSQLSQL

Consider a chat message application and the following simplified schema. Users have names and phone numbers, and can send messages to other users. A message tracks the sender, the recipient, the time of the message and the message content.

```
CREATE TABLE Users(
  uid int PRIMARY KEY,
  name text,
  phone text  --- in the format xxx-yyy-zzzz
              --- xxx is the three digit area code
);

CREATE TABLE Messages(
  recipient int REFERENCES Users(uid),
  sender int REFERENCES Users(uid),
  time timestamp NOT NULL,
  message text NOT NULL,
  PRIMARY KEY (recipient, sender, time)
);
```

1. **(6 points)** Circle true or false for the following statements:

   (a) **True / False** I can send a message to myself.

> **TRUE**: There is no unique constraint on `recipient, sender`.

   (b) **True / False** If I delete my user from the service, my messages will be erased.

> **FALSE**: There is no `ON DELETE CASCADE` on the foreign key relationships.

   (c) **True / False** A user must send at least one message to another user.

> **FALSE**: To express that constraint, `Users` and `Messages` would need to be combined.

   (d) **True / False** If I have sent a message to you, both you and I cannot delete our users.

> **TRUE**: The foreign key constraints on `recipient` and `sender` ensure that you and I cannot delete our users before deleting the messages first.

   (e) **True / False** If I have never sent a message, then I can delete my user.

> **FALSE**: I could have recieved a message.

   (f) **True / False** I cannot send two messages at the exact same time.

> **FALSE**: I can if it is to two different recipients.

Write SQL queries to answer the following questions:

2. **(2 points)** Let `strlen(<text>)` be a user defined function that takes as input a string and returns the length of the string. When were the 5 longest messages sent?

Took points off if you did not return `time`.

```
SELECT time
FROM Messages
ORDER BY strlen(message) DESC
LIMIT 5;
```

3. **(2 points)** How many messages have users from the bay area (area code 510) sent?
   We checked for `COUNT(*)` (no distinct), `LIKE '510-%'`, and `sender = uid`.

```
SELECT COUNT(*)
FROM Users AS u, Messages AS m
WHERE u.phone LIKE '510-%' AND m.sender = u.uid
```

4. **(4 points)** List the names of users that have sent more messages than they have recieved.

Using a nested query was also allowed. A common mistake was to put an aggregation function in the `WHERE` clause.

```
SELECT name
FROM Users
WHERE (SELECT COUNT(*) FROM Messages WHERE Messages.sender = Users.uid) >
      (SELECT COUNT(*) FROM Messages WHERE Messages.recipient= Users.uid);
```

5. **(4 points)** List the messages that have been sent between at least two distinct pairs of users. For instance, if the message "hi" was sent from user 1 to 2 but not by any other pair of users, then it should not be in the result set. If the message "bye" was send from user 1 to 2 as well as from user 100 to 101, then "bye" should be in the result set.

```
SELECT message
FROM (SELECT distinct message, sender, recipient
      FROM Messages) AS m
GROUP BY message
HAVING count(*) > 1
```

or

```
SELECT message
FROM Messages
GROUP BY message
HAVING count(distinct (sender, recipient)) > 1
```

Note that the following has a subtle error—if the same message repeatedly send from user 1 to user 2 and between no other users, then it should not be in the result.

```
SELECT message
FROM Messages
GROUP BY message
HAVING count(*) > 1
```

6. **(4 points)** Consider the 5 users that have sent the most messages. What is the average message length of all messages that these 5 users have sent?

We looked for `AVG(strlen())`, `ORDER BY COUNT(*)` and `sender IN`. Common mistakes forgot to include a `GROUP BY` when using `COUNT()`.

```
SELECT AVG(strlen(message))
FROM Messages
WHERE Messages.sender IN (
  SELECT sender
  FROM Messages
  GROUP BY sender
  ORDER BY COUNT(*) DESC
  LIMIT 5
);
```

We also accepted this interpretation:

```
SELECT sender, AVG(strlen(message))
FROM Messages
GROUP BY sender
ORDER BY COUNT(*) DESC
LIMIT 5;
```

7. **(6 points)** Consider all of the users that "Trump" has sent messages to. How many other users have sent at least one message to all of the users that the user "Trump" has sent messages to?

We gave points for having the double negation structure.

```
SELECT name
FROM Users AS U1
WHERE NOT EXISTS (
  SELECT *
   FROM Messages AS M1, Users AS U2
   WHERE U2.name = ``Trump'' AND M1.sender = U2.uid AND
         NOT EXISTS (
          SELECT
          FROM Messages AS M2
          WHERE M2.recipient = M1.recipient AND
                U1.uid = M2.sender));
```

4 (17 points) Misc. Questions

(2 points each)

1. Provide a simple example (e.g., a diagram and/or at most 2 short sentences) that highlights the difference between the **Hierarchical Model** and the **Relational Model**.

Hierarchical model stored in tree structure. No physical independence. Incurs duplication.
A model in which entities and relationships are expressed as relations.
Relational models everything all tuples and grouped as relations.

2. Let table A contain $|A|$ tuples, let table B contain $|B|$ tuples, and let R be the result of joining A and B. What is the minimum and maximum cardinality of R?

minimum: 0
maximum: $|A| \times |B|$

1 point: for minimum. 1 point: for maximum.

3. In at most 2 short sentences, describe the difference between inner join and **Outer Join**, and the significance of **Outer Joins** in the context of data management systems.

In SQL, the resulting table retains each input row even if no matching row exists.
Cannot always be expressed as the output of a cross-product.
Deals with Nulls

4. In at most 2 short sentences, define **Integrity Constraint** and describe its significance in the context of data management systems

Enforces application level guaranties
Ensures consistency
Applies to every *instance* of the database.

(1 point each) Circle True or False for the following statements.

1. True/False: ER diagrams are precise and unambiguous represention of the application data.

False

2. True/False: ER diagrams can be automatically translated into the relational model.

False

3. True/False: All relational models can be expressed as ER diagrams.

False

(2 points each) Consider the following table schemas:

```
A(a, b, c)
B(a, b, d)
C(c, d, e)
```

1. Write the output schema of the following statement: $T = \sigma_{c>10}(A) \bowtie (B)$

T(a, b, c, d)

2. Let A and C each contain 10 tuples. What is the cardinality of the following statement: $\pi_{a,b}(A) \bowtie (C)$

100. A natural join on tables with no overlapping attributes is equivalent to a cross product.

(2 points) In lecture, we stated how many applications or concepts in the physical world (e.g., not in a computer) can be viewed as a join. For instance, yelp can be viewed as a *join* between a user's location and restaurant establishments. Provide another example of a join in the physical world by filling in the following sentence.

We accepted almost any interpretable response.

A hurricane can be viewed as a join between a butterfly and global weather patterns on the keys of chaos theory