In the SQL questions, you are welcome to use views. Comments are not required, although they may help us mark your answers.

The last page of this exam is a reference page, containing schemas and a dataset. You may tear it off.

There is a blank page at the end for rough work. If you want any of it marked, indicate that clearly there, as well as in the question itself.

A mark of at least 40 out of 100 on this exam is required in order to pass the course.
Question 1. [16 marks]
We begin with SQL. This question uses the schema from Assignment 2. Relevant portions are provided on the reference page at the end of the test. You may tear it off.

Important: You may use the view defined in any subquestion (even if you didn't solve it) when solving other subquestions. It may help to define additional views.

Part (a) [2 marks]
In SQL, define a view called Distance that gives the crows-flies distance, from the source to the destination, for each request. You must use the `<@>` operator to compute this distance. Your result must have the form below (note the column names).

<table>
<thead>
<tr>
<th>rid</th>
<th>miles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Part (b) [2 marks]
Suppose we have just started up psql and we want to be able to refer to our Uber tables and views as simply Request, Place, etc., and to use the `<@>` operator. What command must we give?

If we don't give this command, we can still refer to our tables. Give the SQL query we would have to use to print the contents of table Request.
Part (c)  [4 marks]
Define a view called Unbilled that finds every drop-off for which the client has not been billed. Report the request ID and the client ID. Your result must have the form below (note the column names).

<table>
<thead>
<tr>
<th>rid</th>
<th>cid</th>
</tr>
</thead>
</table>

Part (d)  [4 marks]
Define a view called Debtors that reports the client ID and the number of unbilled drop-offs for each client who has more than one unbilled drop-off. Report the result in order from the client with the most to the client with the fewest unbilled drop-offs. Where clients are tied, the order doesn’t matter. Your result must have the form below (note the column names).

<table>
<thead>
<tr>
<th>cid</th>
<th>numunbilled</th>
</tr>
</thead>
</table>

|-----+-------------|

|-----|-------------|
Part (e) [4 marks]

Write SQL code that, for each of the unbilled drop-offs, bills the client by adding the appropriate row to the Billed table.

There is more space here than you need.
Question 2. [8 marks]
A complete ride is one that goes from request through to drop-off. The length of a ride is the time between pick-up and drop-off. A morning ride is one in which the hour of the pick-up was between 6 and 11 inclusive; an afternoon ride is one in which the hour of the pick-up was between 12 and 17 inclusive.

Write a query to report the average length of all complete morning rides and the average length of all complete afternoon rides. Your result must have the form below and two rows: one with category ‘morn’ and one with category ‘aft’. You may assume that there is at least one complete ride in each category.

<table>
<thead>
<tr>
<th>category</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hint: To get the hour from attribute datetime, use extract(hour from datetime).
Continue your answer here if you need more space.
Question 3. [8 marks]

We will continue with the same schema, but switch to Relational Algebra.

Use Relational Algebra to find all clients who have been picked up by every driver, but have never requested a ride from the location “Bahen Centre”. Report simply their client ID. Use only the basic operators we used in class: \( \Pi, \sigma, \bowtie, \times, \cap, \cup, -, \rho \), and assignment.
Question 4. [6 marks]
Suppose I have a file called nonsense.ddl containing this:

```
DROP SCHEMA IF EXISTS rp CASCADE;
CREATE SCHEMA rp;
SET SEARCH_PATH TO rp;

CREATE TABLE Things (  
    A INT PRIMARY KEY,
    B INT,
    C INT UNIQUE
);

CREATE TABLE Stuff (  
    D INT PRIMARY KEY,
    E INT,
    F INT,
    FOREIGN KEY (E) REFERENCES Things(C) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Junk (  
    G INT PRIMARY KEY,
    H INT,
    I INT,
    FOREIGN KEY (H) REFERENCES Stuff(D) ON UPDATE CASCADE ON DELETE SET NULL,
    FOREIGN KEY (I) REFERENCES Things(A) ON UPDATE RESTRICT ON DELETE CASCADE
);```

```
Part (a) [3 MARKS]
Suppose the tables have been populated as shown below. Modify the data to show the contents of the three tables after this command is executed:

UPDATE Things SET c = c-1 WHERE c < 4;

<table>
<thead>
<tr>
<th>Things</th>
<th>Stuff</th>
<th>Junk</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

Part (b) [3 MARKS]
Suppose we began with the same original tables, shown below, but ran a different command. Modify the data to show the contents of the three tables after this command is executed:

DELETE FROM Things WHERE b = 6;

<table>
<thead>
<tr>
<th>Things</th>
<th>Stuff</th>
<th>Junk</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>
Question 5. [15 marks]
This is a DTD representing questions on a quiz, and student responses for that quiz.

```xml
<!ELEMENT quiz (questions, class-responses)>  
<!ELEMENT questions ((mc-question | tf-question)+)>  
<!ELEMENT mc-question (question, option+)>  
<!ATTLIST mc-question qid ID #REQUIRED>  
<!ATTLIST mc-question solution CDATA #REQUIRED>  
<!ELEMENT question (#PCDATA)>  
<!ELEMENT option (#PCDATA)>  
<!ELEMENT tf-question (question)>  
<!ATTLIST tf-question qid ID #REQUIRED>  
<!ATTLIST tf-question solution (True|False) #REQUIRED>  
<!ELEMENT class-responses (student+)>  
<!ATTLIST student sid ID #REQUIRED>  
<!ELEMENT response EMPTY>  
<!ATTLIST response qid IDREF #REQUIRED>  
<!ATTLIST response answer CDATA #REQUIRED>
```

The tear-off reference page at the end of the exam gives an XML file called quiz.xml that is valid with respect to this DTD. The following questions refer to that file.

1. The code below runs without errors. What output does it produce?

   ```xml
   count(doc("quiz.xml")//student/response)
   ```

2. The code below runs without errors. What output does it produce?

   ```xml
   doc("quiz.xml")//student/count(response)
   ```

3. Complete the code at the top of the next page so that it will find the sids of students who answered question Q888. The output must be a list of sid elements, each containing a student number. For example, for quiz.xml, the output should be

   ```xml
   <sid>s998801234</sid>,
   <sid>s001078452</sid>,
   <sid>s997739991</sid>
   ```

   Fill in only the return clause; do not change anything else. You must not use a FLWOR expression in the return clause.
let $d := doc("quiz.xml")
for $s in $d//student
return

4. Complete the following code so that it will produce the text of the mc-question(s) with the most answer options. Fill in only the where clause; do not change anything else. You must not use an FLWOR expression in the where clause.

let $d := doc("quiz.xml")
for $x in $d//mc-question
where

return $x/question

5. Use an every expression to complete the following code so that it will produce the sids of those students whose answers are all correct (for the questions that they did answer). You may assume every student answered at least one question. Fill in only the where clause; do not change anything else, and be sure to use an every expression.

let $d := doc("quiz.xml")
for $s in $d//student
where

return $s/@sid

6. The following code is intended to produce the question element for every question that some student answered. The code doesn’t work. Fix it, making the minimal changes necessary.

<answered>
let $d := doc("quiz.xml")
for $x in $d/mc-question
where $x/@qid = $d//response/@qid
return $x/question
</answered>
Question 6. [6 marks]

Consider this well-formed XML data:

```
<ClassResponses>
  <Student SID = "g0cookie">
    <ResponseSet QID="prefs">
      <Response>recommender systems</Response>
    </ResponseSet>
    <SingleResponse QID="year"></SingleResponse>
    <ResponseSet QID="courses">
      <Response>csc309</Response>
      <Response>csc318</Response>
    </ResponseSet>
    <TimetableResponse QID="times">
      <Slot answer="yes">Mondays, 3pm</Slot>
      <Slot answer="yes">Fridays, 10am</Slot>
    </TimetableResponse>
  </Student>
  <Student SID = "g0bigbird">
  </Student>
  <Student SID = "g0ernie">
    <SingleResponse QID="college">New</SingleResponse>
    <TimetableResponse QID="times">
      <Slot answer="yes">Mondays, 3pm</Slot>
      <Slot answer="maybe">Wednesday evenings</Slot>
      <Slot answer="yes">Fridays, 11am</Slot>
    </TimetableResponse>
    <SingleResponse QID="year">3</SingleResponse>
    <ResponseSet QID="prefs">
      <Response>human genome</Response>
      <Response>location-aware apps</Response>
    </ResponseSet>
  </Student>
</ClassResponses>
```
Part (a) [2 marks]
For each of the following rules, circle Yes or No to indicate whether it could be part of a DTD that accepts the XML provided on the previous page.

<!ATTLIST Slot answer (yes | no | maybe) #REQUIRED>
  Yes  No

<!ATTLIST SingleResponse QID ID #REQUIRED>
  Yes  No

<!ELEMENT SingleResponse (EMPTY)>
  Yes  No

<!ELEMENT ResponseSet (Response, Response*)>
  Yes  No

Part (b) [2 marks]
Write a rule for the element Student (just the element, not its attributes). Ensure that your rule accepts the XML provided.

Part (c) [2 marks]
Write a rule for the SID attribute of the element Student. Ensure that your rule accepts the XML provided and enforces this rule: no two students may have the same SID.
**Question 7.** [10 marks]

Consider the relation $R$ on attributes $LMNOPQ$, with the following functional dependencies:

\[ N \rightarrow OP, \quad M \rightarrow L, \quad LN \rightarrow M, \quad O \rightarrow NQ, \quad P \rightarrow M. \]

This question is about performing BCNF decomposition on $R$.

1. For each functional dependency, indicate whether it violates BCNF for this relation.

<table>
<thead>
<tr>
<th>Functional Dependency</th>
<th>Violates BCNF?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N \rightarrow OP$</td>
<td>Yes</td>
</tr>
<tr>
<td>$M \rightarrow L$</td>
<td>Yes</td>
</tr>
<tr>
<td>$LN \rightarrow M$</td>
<td>Yes</td>
</tr>
<tr>
<td>$O \rightarrow NQ$</td>
<td>Yes</td>
</tr>
<tr>
<td>$P \rightarrow M$</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2. Suppose we decompose relation $R$ into two new relations: relation $R_1$ on attributes $PQNO$ and relation $R_2$ on attributes $LMP$. Project the functional dependencies of $R$ onto $R_1$.

3. Project the functional dependencies of $R$ onto $R_2$. 
4. For each new relation, indicate whether it is in BCNF.

Is $R_1$ in BCNF? Yes No

Is $R_2$ in BCNF? Yes No

5. Is the algorithm complete? Yes No

If yes, explain why. If no, state what is left to do.

Below is space for additional rough work. This will not be marked unless you indicate clearly that it should be marked, and with which part of this question it belongs.
Question 8. [6 marks]
Consider relation $R(A, B, C, D, E, F)$ with functional dependencies $S$.

$$S = \{ AC \rightarrow B, \quad B \rightarrow ACE, \quad BC \rightarrow E, \quad BD \rightarrow AF \}$$

Compute a minimal basis for $S$. Show your rough work, and put your final answer where shown on the next page. There will be no marks for a correct answer without the rough work.
Question 9. [4 marks]

Suppose we are employing the 3NF synthesis algorithm on a relation $R(A, B, C, D, E, F)$, and we already have the following minimal basis:

$$ S = \{ A \rightarrow C, \quad BF \rightarrow D, \quad DF \rightarrow C, \quad EF \rightarrow D \} $$

Suppose I declare that the following schema is the result of the 3NF synthesis algorithm:

$$ R_1(A, C) \quad R_2(B, F, D) \quad R_3(D, F, C) \quad R_4(E, F, D) $$

Why is this an incorrect application of the 3NF synthesis algorithm? Justify all points in your answer.

Produce a correct schema, according to the 3NF synthesis algorithm. Explain all steps in your answer.
Question 10. [11 Marks]

Part (a) [4 Marks]
Suppose we have the tables Cats(A, B, C) with m rows, and Dogs(B, C, D) with n rows. Fill in the table to show the minimum and the maximum possible number of rows in the result of each kind of SQL join.

<table>
<thead>
<tr>
<th>SQL join</th>
<th>minimum number of rows</th>
<th>maximum number of rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cats natural join Dogs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cats natural left outer join Dogs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cats natural right outer join Dogs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cats natural full outer join Dogs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Part (b) [1 Mark]
Give one important difference between a trigger and an assertion in SQL.

Part (c) [2 Marks]
Consider a relation R on attributes ABCDE. Are the following sets of functional dependencies equivalent? $S_1 = \{ A \rightarrow B, \ B \rightarrow C, \ C \rightarrow D \}$ and $S_2 = \{ A \rightarrow D \}$.

Circle one: Yes No

If yes, explain why. If no, give a counterexample: an instance of R that satisfies one set of functional dependencies but not the other. Indicate which set of FDs your instance satisfies.
Part (d)  [2 marks]
Suppose schema $R(A, B, C, D, E, F)$ violates BCNF, and I observe that if I were to decompose it into $R_1(A, B, C)$ and $R_2(B, D, E, F)$, both $R_1$ and $R_2$ would satisfy the BCNF property. Is the new schema guaranteed to have a lossless join?

Circle one: Yes  No

Explain your answer in the space below.

Part (e)  [2 marks]
Suppose I have a schema that I generated using the 3NF synthesis algorithm. Will this ensure that there will be no anomalies?

Circle one: Yes  No

Explain your answer in the space below. Below is space for additional rough work. This will not be marked unless you indicate clearly that it should be marked, and with which part of this question it belongs.
Question 11. [4 marks]

Below is an Entity-Relationship diagram about a studio that offers private and group lessons, such as music studio or yoga studio. It may or may not represent the domain well.

Which of the following is true, according to the Entity-Relationship diagram:

1. This model contains a weak entity set.
   True False

2. An instructor can teach at most two classes.
   True False

3. There can be two clients with the same email as long as they are registered in different classes.
   True False

4. An instructor must have at least one qualification.
   True False

5. For a given class, every registered client pays the same price.
   True False

6. There can be at most twice as many instructors as there are classes.
   True False

7. The registered relationship is a one-to-many relationship.
   True False

8. There cannot be two classes with the same name.
   True False
Question 12. [6 MARKS]
Translate this Entity-Relationship diagram into a relational schema. For each relation, provide its name, attributes and keys. To indicate a key, underline all attributes that are part of the key using a single line. Also include all referential integrity constraints, using relational notation (not SQL notation).
[Use the space below for rough work. This page will not be marked, unless you clearly indicate the part of your work that you want us to mark.]
DDL for SQL questions

Important notes:
* As on Assignment 2, all tables are defined within the psql schema called "uber"
* Only the tables you need for the exam questions are included here.

-- The name, such as 'Pearson Airport', associated with a location.
CREATE TABLE Place (  
    name varchar(30) PRIMARY KEY,  
    location point NOT NULL ) ;

-- A request for a ride. source is where the client wants to be picked up from, and destination is where they want to be driven to.
CREATE TABLE Request (  
    request_id integer PRIMARY KEY,  
    client_id integer NOT NULL REFERENCES Client,  
    datetime timestamp NOT NULL,  
    source varchar(30) NOT NULL References Place(name),  
    destination varchar(30) NOT NULL References Place(name) ) ;

-- A driver was dispatched to pick up a client, in response to their request. car_location is where his or her car was at the time when the driver was dispatched.
CREATE TABLE Dispatch (  
    request_id integer PRIMARY KEY REFERENCES Request,  
    driver_id integer NOT NULL REFERENCES Driver,  
    car_location point,  
    datetime timestamp ) ;

-- The client who made this request was picked up at this time.
CREATE TABLE Pickup (  
    request_id integer PRIMARY KEY NOT NULL REFERENCES Dispatch,  
    datetime timestamp NOT NULL ) ;

-- The client who made this request was dropped off at this time.
CREATE TABLE Dropoff (  
    request_id integer PRIMARY KEY NOT NULL REFERENCES Pickup,  
    datetime timestamp NOT NULL ) ;

-- This table must have a single row indicating the current rates. base is the cost for being picked up, and per_mile is the additional cost for every mile travelled.
CREATE TABLE Rates (  
    base real NOT NULL,  
    per_mile real NOT NULL ) ;

-- This client associated with this request was billed this amount for the ride.
CREATE TABLE Billed (  
    request_id integer PRIMARY KEY REFERENCES Dropoff,  
    amount real NOT NULL ) ;
File quiz.xml

<?xml version="1.0" standalone="no" ?>
<!DOCTYPE quiz SYSTEM "quiz.dtd">
<quiz>
  <questions>
    <mc-question qid="Q516" solution="1">
      <question>
        What do you promise when you take the oath of citizenship?
      </question>
      <option>To pledge your loyalty to Queen Elizabeth II</option>
      <option>To fulfill the duties of a Canadian</option>
      <option>To pledge your allegiance to the flag</option>
      <option>To pledge your loyalty to Canada from sea to sea</option>
    </mc-question>
    <tf-question qid="Q637" solution="False">
      <question>
        The Prime Minister is Canada's Head of State.
      </question>
    </tf-question>
    <mc-question qid="Q888" solution="4">
      <question>What is the Underground Railroad?</question>
      <option>The first railway to cross Canada</option>
      <option>The CPR's secret railway line</option>
      <option>The TTC subway system</option>
      <option>A network used by slaves who escaped the United States into Canada</option>
    </mc-question>
  </questions>
  <class-responses>
    <student sid="s998801234">
      <response qid="Q516" answer="1" />
      <response qid="Q637" answer="False" />
      <response qid="Q888" answer="4" />
    </student>
    <student sid="s001078452">
      <response qid="Q516" answer="4" />
      <response qid="Q637" answer="False" />
      <response qid="Q888" answer="4" />
    </student>
    <student sid="s997733991">
      <response qid="Q516" answer="1" />
      <response qid="Q637" answer="True" />
      <response qid="Q888" answer="2" />
    </student>
    <student sid="s555555555">
      <response qid="Q637" answer="False" />
    </student>
  </class-responses>
</quiz>