Do not turn this page until you have received the signal to start.
In the meantime, please fill in the identification section above and
read the instructions below.

In the SQL questions, you are welcome to use views.
Comments are not required, although they may help us mark your answers.
There are two pages at the end for rough work. If you want any of it marked,
indicate that clearly there, as well as in the question itself.

# 1: _____/ 18
# 2: _____/ 9
# 3: _____/ 7
# 4: _____/ 12
# 5: _____/ 10
# 6: _____/ 11
# 7: _____/ 13
# 8: _____/ 10
# 9: _____/ 12

TOTAL: _____/102
Question 1. [18 marks]
Consider the following schema for tracking customers’ ratings of books for a bookstore website.

- Authors(AID, name, city)
- Customers(CID, name)
- Rates(ISBN, CID, rating). Rating is an integer.

The following inclusion dependencies hold:

- Books(author) ⊆ Authors(AID)
- Rates(CID) ⊆ Customers(CID)

Throughout this question, we will use only the basic Relational Algebra operators Π, σ, ×, ⩾, ∩, −, ρ, :=. Assume the set semantics (not bag semantics) for Relational Algebra.

Part (a) [6 marks]
For each query below, explain in plain English what it computes.

1. \((\rho_{R(CID)}\Pi_{R1.CID}\sigma_{R1.CID=R2.CID\land R1.ISBN\neq R2.ISBN}((\rho_{R1}Rates) \times (\rho_{R2}Rates)))) \cup
   \Pi_{CID\sigma_{\text{length}>500}}(Rates \Join Books)\)

2. \(R0 := \Pi_{AID}\sigma_{\text{city="Iqaluit"}}Author\)
   \(R1(CID, AID) := \Pi_{CID,\text{author}}\sigma_{\text{rating}=10}(Rates \Join Books)\)
   \(R2 := (\Pi_{CIDCustomers}) \times (R0)\)
   \(R3 := R2 - R1\)
   \(Answer := (\Pi_{CIDCustomer}) - (\Pi_{CIDR3})\)
3. $R1(CID, AID) := \Pi_{CID,author}(Rates \bowtie Books)$

$R2(CID, AID) := \Pi_{CID,author}(R1(CID, AID) \bowtie Books)$

$R3 := R1 - R2$

$Answer := (\Pi_{CID}Customer) - (\Pi_{CID}R3)$

---

Part (b)  [12 Marks]

Write the following queries in relational algebra.

1. Find the name and AID of the author of the second shortest book with a rating of 6 or more. If there is a tie, report them all.
Here is that schema again, for easy reference:

- **Books(ISBN, title, author, year, length)**. ISBN is a string used internationally for identifying books.
- **Authors(AID, name, city)**
- **Customers(CID, name)**
- **Rates(ISBN, CID, rating)**. Rating is an integer.

The following inclusion dependencies hold:

- **Books(author) ⊆ Authors(AID)**
- **Rates(CID) ⊆ Customers(CID)**

2. Find the CID and name of customers who have rated at most two books and have never rated one by any author named “Laurence”.
Question 2. [9 marks]

Answer the following questions. Do not guess on the true-false questions; there is one point for each correct answer, -1 for each incorrect answer, and 0 points if you leave the answer blank.

1. If $R$ has one tuple, $R \bowtie R$ has no tuples.
   
   True      False

2. For any relation $R$, $R \bowtie R = R$.
   
   True      False

3. If relation $R$ has arity 1, $R \bowtie R$ has cardinality 0.
   
   True      False

4. Suppose we have a relation R with attributes ABCD and that the following is a legal instance of this relation:

   \[
   \begin{array}{cccc}
   A & B & C & D \\
   1 & 3 & 9 & 6 \\
   2 & 3 & 9 & 6 \\
   1 & 3 & 4 & 6 \\
   5 & 7 & 9 & 6 \\
   \end{array}
   \]

   Consider the possible functional dependencies below. Circle all that definitely do not hold in R.

   \[
   CD \rightarrow B \quad BC \rightarrow D \quad A \rightarrow C \quad C \rightarrow A
   \]

5. Assume the same relation and instance as in the previous question. Can you identify an FD not listed above that does hold? If yes, state it. If not, explain why not.

6. Consider the relation the relation S with attributes MNOP only one key: MO. Give a set of functional dependencies under which S is in 3NF but not BCNF.
Question 3. [7 marks]
Consider the following table definitions.

```sql
CREATE TABLE S ( P INT PRIMARY KEY, Q INT );
CREATE TABLE R ( A INT PRIMARY KEY, B INT, C INT, FOREIGN KEY (B) REFERENCES S(P) ON UPDATE SET NULL ON DELETE CASCADE );
CREATE TABLE T ( M INT, N INT, O INT, PRIMARY KEY (M, N), FOREIGN KEY (M) REFERENCES S(P) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (N) REFERENCES R(A) ON DELETE CASCADE ON UPDATE CASCADE );
```

Suppose the tables have been populated as follows:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>11</td>
<td>2</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>8</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>8</td>
<td>75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>2</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>140</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>N</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>2</td>
<td>11</td>
<td>1112</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>11</td>
<td>1219</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>18</td>
<td>1219</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>15</td>
<td>1317</td>
</tr>
</tbody>
</table>

1. Write a query to delete the tuple (2,11,1112) from T.

2. Show the contents of the three relations after your query is executed.
3. Suppose, instead, we were to start with the original tables shown above and make two changes to R: delete the tuple (18,8,75) and change the tuple (15,8,71) to (15,6,71). Show the contents of the three relations after these changes.

4. Suppose, instead, we were to start with the original tables shown above and delete the tuple (8,142) from S. Show the contents of the 3 relations after these changes.
Question 4. [12 marks]
Consider the following schema for a grocery store. Note that it does not represent that quantity of an item included in a purchase. Assume that no one every buys more than one of anything in a single purchase. (This will make it easy to total up the cost of a purchase.)

-- Category can have values such as 'food', 'drugs', or 'toiletries'.
CREATE TABLE `Product`
    (id INT PRIMARY KEY,
     name TEXT,
     category TEXT
    );

CREATE TABLE `Price`
    (id INT PRIMARY KEY,
     price FLOAT,
     FOREIGN KEY (id) REFERENCES `Product`(id)
    );

-- A purchase on this day by the customer at this phone number.
CREATE TABLE `Purchase`
    (id INT PRIMARY KEY,
     phone TEXT,
     day DATE
    );

-- This item was included in this purchase.
CREATE TABLE `included`
    (item INT,
     purchase INT,
     PRIMARY KEY (item, purchase),
     FOREIGN KEY (item) REFERENCES `Product`(id),
     FOREIGN KEY (purchase) REFERENCES `Purchase`(id)
    );
Write queries in SQL to get the following information.

1. The phone number, day and total cost of the most expensive purchase not counting the cost of any drugs \(i.e.,\) any product whose category is ‘drugs’.
Here is the same schema again, for easy reference:

-- Category can have values such as 'food', 'drugs', or 'toiletries'.
CREATE TABLE Product (  
id INT PRIMARY KEY,
  name TEXT,
  category TEXT
);

CREATE TABLE Price (  
id INT PRIMARY KEY,
  price FLOAT,
  FOREIGN KEY (id) REFERENCES Product(id)
);

-- A purchase on this day by the customer at this phone number.
CREATE TABLE Purchase (  
id INT PRIMARY KEY,
  phone TEXT,
  day DATE
);

-- This item was included in this purchase.
CREATE TABLE included (  
  item INT,
  purchase INT,
  PRIMARY KEY (item, purchase),
  FOREIGN KEY (item) REFERENCES Product(id),
  FOREIGN KEY (purchase) REFERENCES Purchase(id)
);
2. For each product, the number of distinct customers (identified by their phone numbers) who have purchased it. Include products even if no one ever purchased them.
Question 5. [10 marks]
Assume that the following valid DTD is defined in file party.dtd:

```xml
<!ELEMENT GuestList (Invitee*)>
<!ELEMENT Invitee (Who, Response)>
<!ATTLIST Invitee nickname ID #REQUIRED>
<!ELEMENT Who (First, Last)>
<!ATTLIST Who age CDATA #IMPLIED>
<!ELEMENT First (#PCDATA)>
<!ELEMENT Last (#PCDATA)>
<!ELEMENT Response (#PCDATA)>
```

Part (a) [6 marks]
For each XML document below, the first two lines are valid. Circle the correct answer to indicate whether the entire document is valid or not. If it is not valid, circle each error and explain why it is an error.

1. ```xml
   <?xml version="1.0" standalone="no" ?>
   <!DOCTYPE GuestList SYSTEM "party.dtd">
   <GuestList>
     <Invitee nickname="sweetPea">
       <Who age="almost 4">
         <First>Catherine</First>
         <Last>Fairgrieve</Last>
       </Who>
       <Response>It's my party!!!</Response>
     </Invitee>
     <Invitee nickname="sunshine">
       <Who age="3" First="Avery" Last="McLaughlin"/>
       <Response>Love to come!</Response>
     </Invitee>
     <Invitee nickname="sunshine">
       <Who>
         <First>Sam</First>
         <Last>Kingston</Last>
       </Who>
       <Response>Can’t wait!</Response>
     </Invitee>
   </GuestList>
   <GuestList>
     <Invitee nickname="dub">
       <Who age="12">
         <First>William</First>
         <Last>Fairgrieve</Last>
       </Who>
       <Response>Count me in</Response>
     </Invitee>
   </GuestList>

   Valid     Not Valid
```
2. ```xml
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE GuestList SYSTEM "party.dtd">
<GuestList>
</GuestList>
```

Valid   Not Valid

3. ```xml
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE GuestList SYSTEM "party.dtd">
<GuestList>
   <Invitee nickname="dub">
      <Who age="12">
         <First>William</First>
         <Last>Fairgrieve</Last>
      </Who>
      <Response>Count me in</Response>
   </Invitee>
   <Invitee nickname="sunshine">
      <Response>Can’t wait!</Response>
      <Who>
         <First>Sam</First>
         <Last>Kingston</Last>
      </Who>
   </Invitee>
</GuestList>
```

Valid   Not Valid

**Part (b)  [4 MARKS]**

Suppose we want to keep the same structure in our XML files, but to enforce additional rules. Which of these can be expressed in our DTD. **Do not guess.** Each question is worth 1 mark if correct, -1 if incorrect, and 0 if you don’t respond. Your minimum mark is 0.

1. An invitee’s age must be between 0 and 100.
   
   Can be   Cannot be

2. An invitee’s last name is optional.
   
   Can be   Cannot be

3. A guest list must have at least 3 guests.
   
   Can be   Cannot be

4. A response must be either “yes” or “no”.
   
   Can be   Cannot be
Question 6.  [11 marks]
Consider the following well-formed XML document, stored in a file called “minutes.xml”.

```
<proceedings>
  <members>
    <member rid="96124" reps="Trinity">
      <name>Barbara Reid</name>
      <votes>
        <vote mid="784" choice="for"/>
        <vote mid="899" choice="against"/>
        <vote mid="955" choice="for"/>
      </votes>
    </member>
    <member rid="11234" reps="Vic">
      <name>Quentin Blake</name>
      <votes>
        <vote mid="784" choice="for"/>
        <vote mid="899" choice="for"/>
        <vote mid="955" choice="against"/>
      </votes>
    </member>
    <member rid="67890" reps="Woodsworth">
      <name>Kenneth Oppel</name>
      <votes>
        <vote mid="784" choice="for"/>
        <vote mid="899" choice="for"/>
        <vote mid="955" choice="against"/>
      </votes>
    </member>
  </members>
  <motions>
    <motion mid="784" passed="true" madeby="96124">
      The committee should set aside $100 for budget overruns.
    </motion>
    <motion mid="899" passed="true" madeby="67890">
      The chair should discuss space issues with the Dean.
    </motion>
    <motion mid="955" passed="false" madeby="96124">
      The party planned for May 12th should be cancelled.
    </motion>
  </motions>
</proceedings>
```
**Part (a) [3 MARKS]**

What does the following XQuery code return? Whitespace in your answer does not matter.

```xml
let $d := doc("minutes.xml")
for $x in $d/proceedings/members/member//vote[@choice="for"]
return $x
```

**Part (b) [3 MARKS]**

What does the following XQuery code return? Whitespace in your answer does not matter.

```xml
let $d := doc("minutes.xml")
for $x in $d/proceedings/*/motion/@madeby
return <made> $x </made>
```
Here is the contents of “minutes.xml” again, for easy reference.

<proceedings>
  <members>
    <member rid="96124" reps="Trinity">
      <name>Barbara Reid</name>
      <votes>
        <vote mid="784" choice="for"/>
        <vote mid="899" choice="against"/>
        <vote mid="955" choice="for"/>
      </votes>
    </member>
    <member rid="11234" reps="Vic">
      <name>Quentin Blake</name>
      <votes>
        <vote mid="784" choice="for"/>
        <vote mid="899" choice="for"/>
        <vote mid="955" choice="against"/>
      </votes>
    </member>
    <member rid="67890" reps="Woodsworth">
      <name>Kenneth Oppel</name>
      <votes>
        <vote mid="784" choice="for"/>
        <vote mid="899" choice="for"/>
        <vote mid="955" choice="against"/>
      </votes>
    </member>
  </members>
  <motions>
    <motion mid="784" passed="true" madeby="96124">
      The committee should set aside $100 for budget overruns.
    </motion>
    <motion mid="899" passed="true" madeby="67890">
      The chair should discuss space issues with the Dean.
    </motion>
    <motion mid="955" passed="false" madeby="96124">
      The party planned for May 12th should be cancelled.
    </motion>
  </motions>
</proceedings>
Part (c) [5 marks]

What does the following XQuery code return? Whitespace in your answer does not matter.

```xml
<passed>
{
let $d := doc("minutes.xml")
for $passed in $d//motions/motion[@passed="true"]
return
(
  <motion> {data($passed)} </motion>,
  <infavour>
    { for $member in $d//member
      for $vote in $member//vote
        where $vote/@mid=$passed/@mid and $vote/@choice="for"
          return <person>{data($member/name)}</person>
    }
  </infavour>
)
}
</passed>
```
**Question 7.** [13 marks]

**Part (a) [7 marks]**
Suppose we have a relation $R(A, B, C, D)$ with functional dependency: $B \rightarrow C$. Suppose that we are going to decompose $R$ into two new relations $S$ and $T$.

1. Give schemas for $S$ and $T$ that will yield a lossy join.

2. Give an example demonstrating that the join is lossy. Explain your answer.

3. Give schemas for $S$ and $T$ that will yield a lossless join.

**Part (b) [3 marks]**
Is the following a valid rule about FDs: If $AB \rightarrow D$ and $BC \rightarrow D$ then $AC \rightarrow D$. Circle one answer.

Valid

Not valid

If it is valid, prove it. If it is not valid, demonstrate this by giving an instance of a relation that satisfies $AB \rightarrow D$ and $BC \rightarrow D$ but not $AC \rightarrow D$. 
Part (c)  [3 marks]
Suppose we have a relation R with attributes ABCDE and functional dependencies $A \rightarrow D$, $B \rightarrow A$, $C \rightarrow A$, $D \rightarrow CE$. Project the functional dependencies of R onto the attribute set ABD.
Question 8. [10 marks]
Suppose we have a relation R with attributes ABCD and functional dependencies $AB \rightarrow C$, $C \rightarrow A$, and $C \rightarrow D$.

1. Circle the FD(s) that violate BCNF: $AB \rightarrow C$, $C \rightarrow A$, $C \rightarrow D$.

2. Perform BCNF decomposition on relation R. Show your rough work here and your final answer at the bottom of the next page.
3. Show your final answer here: List the relations that are in your final decomposition of R and state the FDs of each.
Question 9. [12 marks]
Below is an Entity/Relationship diagram about the music industry. It may or may not represent the domain well.
Part (a)  [7 Marks]
Which of the following statements are true according to this Entity/Relationship diagram? Do not guess. There is one point for each correct answer, -1 for each incorrect answer, and 0 points if you leave the answer blank.

1. The same recording can be on more than one album.
   True False

2. There can be two albums in the same year as long as they have different titles.
   True False

3. For every recording, there must be an artist who recorded it.
   True False

4. For every recording, there must be an album that it is on.
   True False

5. Album titles are unique.
   True False

6. There cannot be two recordings of the same song in the same year.
   True False

7. A recording can be by several artists.
   True False

Part (b)  [2 Marks]
Name a weak entity set in this diagram.

Name its supporting entity set(s):

Part (c)  [3 Marks]
Extend the diagram above to represent the following information: Record companies (such as Sony Music or EMI) distribute albums. Record companies have names, which are unique. Every album is distributed by exactly one record company, but a record company can distribute many albums.
[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.]