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 blank 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.

A mark of at least 40 out of 101 on this exam is required in order to pass the course.

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

TOTAL: _____/101
Consider the following schema for a program that pairs students with mentors who offer career advice.

**Relations**

- **Mentorship(year, sponsor, budget)**
  A tuple in this relation represents information about a single year of the mentorship program: the year, the company that sponsored the program, and the budget for the program.

- **Mentor(MID, name, email, employer, title, phone)**
  A tuple in this relation represents information about a mentor.

- **MApplication(MID, year, capacity)**
  A tuple in this relation indicates that mentor MID applied to be a mentor in a given year and they were willing to mentor up to capacity students.

- **Student(SID, name, email, phone)**
  A tuple in this relation represents information about a student.

- **SApplication(SID, year, cgpa)**
  A tuple in this relation indicates that a student applied to be mentored in a given year and had the given cgpa at the time.

- **Expertise(MID, year, area)**
  A tuple represents an area, such as “web development” that a mentor declared expertise in for a particular year.

- **Interest(SID, year, area, priority)**
  A tuple represents an area that a student has declared an interest in for a particular year. The attribute priority indicates how strong the student’s interest in that area is, with 1 being the highest priority.

- **Match(MID, SID, year)**
  A tuple indicates that the mentor MID was matched with student SID in the given year.

**Integrity constraints**

The obvious integrity constraints apply:

\[
\begin{align*}
\text{MApplication}[\text{MID}] & \subseteq \text{Mentor}[\text{MID}] & \text{MApplication}[\text{year}] & \subseteq \text{Mentorship}[\text{year}] \\
\text{SApplication}[\text{SID}] & \subseteq \text{Student}[\text{SID}] & \text{SApplication}[\text{year}] & \subseteq \text{Mentorship}[\text{year}] \\
\text{Expertise}[\text{MID}] & \subseteq \text{Mentor}[\text{MID}] & \text{Expertise}[\text{year}] & \subseteq \text{Mentorship}[\text{year}] \\
\text{Interest}[\text{SID}] & \subseteq \text{Student}[\text{SID}] & \text{Interest}[\text{year}] & \subseteq \text{Mentorship}[\text{year}] \\
\text{Match}[\text{MID}] & \subseteq \text{Mentor}[\text{MID}] & \text{Match}[\text{SID}] & \subseteq \text{Student}[\text{SID}] \\
\text{Match}[\text{year}] & \subseteq \text{Mentorship}[\text{year}] & & \\
\end{align*}
\]
Answer Questions 1, 2, and 3 in relational algebra, using only the basic operators that we used on our Assignment 1: \( \Pi, \sigma, \Box, \Diamond, \land, \lor, \land \neg, \rho, \) and the assignment operator :=.

**Question 1. [7 marks]**

Part (a) [2 marks]

Express the following integrity constraint with the notation \( R = \emptyset \), where \( R \) is an expression of relational algebra:

A mentor cannot be recorded in the \( MApplication \) relation with a capacity for less than 1 student.

Part (b) [5 marks]

Consider this integrity constraint:

\[
\begin{align*}
\text{Temp}_1(year, SID) & := \\
& \Pi_{M1\text{.year},M1\text{.SID}} \sigma M1\text{.SID} = M2\text{.SID} \land M1\text{.MID} \neq M2\text{.MID} \land M1\text{.year} = M2\text{.year} (\rho_{M1\text{Match}} \times \rho_{M2\text{Match}}) \\
\text{Temp}_2(year, MID) & := \\
& (\Pi_{year,MID} \text{Mapplication}) - (\Pi_{year,MID} \text{Match}) \\
(\Pi_{year,\text{Temp}_1}) \cap (\Pi_{year,\text{Temp}_2}) & = \emptyset
\end{align*}
\]

1. Create a very small instance of relations \( \text{Match} \) and \( \text{MApplication} \) that violates the constraint but otherwise satisfies the schema.

<table>
<thead>
<tr>
<th>Match:</th>
<th>( \text{MID} )</th>
<th>( \text{SID} )</th>
<th>( \text{year} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MApplication:</th>
<th>( \text{MID} )</th>
<th>( \text{year} )</th>
<th>( \text{capacity} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Express this constraint, precisely, in English.
Question 2. [5 marks]

Let’s say a mentor and a student have an area “in common” in a given year if, that year, the mentor listed it as an area of expertise and the student listed it as an area of interest. Write a query in relational algebra to find all pairs of an MID and an SID for whom the following holds for 2014: (a) they each applied, (b) neither of them is matched to anyone, and (c) they have one more more areas in common that the student rated priority 1. Report simply the MID and SID.

Note: The schema is repeated on the last page of the exam. You may tear it off for reference.
Question 3. [6 marks]
Suppose we want to ensure we find a mentor for students who have applied for the first time in 2014, or who applied in at least one previous year but have never been matched. Write a query in relational algebra to find these students. Report their SID and the area that is their highest priority for 2014. (Remember that a lower number indicates a higher priority.) If a student has several interests tied for their highest priority, include a tuple for each.
Now answer Questions 4, 5, and 6 on the same schema, but in SQL.

**Question 4. [8 marks]**

**Important:** You may use the results of any subquestion when solving other subquestions.

**Part (a) [2 marks]**
Create a view called `matched`, with the form below, that finds all mentors who were matched in 2013 and the number of mentees they were matched with.

```
mid | mentees
-----+---------
```

**Part (b) [2 marks]**
Create a view called `unMatched`, with the same form as in part (a), that finds all mentors who applied in 2013 but were not matched with any mentee in 2013. The `mentees` column must contain only zeros.

**Hint:** You can use a constant, such as 0, in place of a column name in a `SELECT` statement.
For example, this is legal: `SELECT 0 as mentees ...`
Part (c)  [2 marks]
Create a view called `matchCount`, again with the same form as in part (a), that finds all mentors who applied in 2013 and the number of mentees they were matched with. Include mentor applicants who were not matched with anyone.

Part (d)  [2 marks]
Write a query to find every mentor who applied in 2013 and who had capacity for more students than they were matched with. Report the MID, their capacity, and the number of mentees they were matched with, in a table with the following form:

<table>
<thead>
<tr>
<th>mid</th>
<th>capacity</th>
<th>mentees</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----</td>
<td>----------</td>
<td>---------</td>
</tr>
</tbody>
</table>
Question 5. [6 marks]
Part (a) [4 marks]
Create a view called `noSpecifiedInterests` that finds every student who applied to the mentorship program in 2014 but specified no interests. Your view should have a single attribute: `SID`.

Part (b) [2 marks]
Insert the students in `noSpecifiedInterests` into the `Interest` table with their area of interest as ‘CS’ and priority 1. You may assume that `noSpecifiedInterests` was created properly, even if you have not solved part (a).

Remember: You can use a constant, such as ‘CS’ or 1, in place of a column name in a `SELECT` statement.
Question 6. [7 marks]

Find all the matches from 2014 where the mentor and student have at least three areas in common that year (as defined in Question 2). Report the MID, SID, number of areas in common, and number of years in which the mentor has applied to be a mentor in the mentorship program. You may use views. Produce a table with the following form:

<table>
<thead>
<tr>
<th>mid</th>
<th>sid</th>
<th>common</th>
<th>years</th>
</tr>
</thead>
</table>

---

...
Question 7. [6 marks]
Consider a relation R on attributes ABCD. For each question below, circle one answer to indicate whether the two sets of functional dependencies are equivalent. If equivalent, briefly explain why. If not equivalent, 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.

1. \( S_1 = \{ A \rightarrow C, \ B \rightarrow C \} \) and \( S_2 = \{ AB \rightarrow C \} \).
   Equivalent \quad \text{Not equivalent}
   If equivalent, give your explanation; if not, give your counterexample:

2. \( S_1 = \{ A \rightarrow B, \ A \rightarrow C, \ B \rightarrow D, \ C \rightarrow D \} \) and \( S_2 = \{ A \rightarrow BC, \ BC \rightarrow D \} \).
   Equivalent \quad \text{Not equivalent}
   If equivalent, give your explanation; if not, give your counterexample:

3. \( S_1 = \{ A \rightarrow B, \ B \rightarrow C, \ C \rightarrow B \} \) and \( S_2 = \{ A \rightarrow C, \ C \rightarrow B \} \).
   Equivalent \quad \text{Not equivalent}
   If equivalent, give your explanation; if not, give your counterexample:
Question 8. [6 marks]

Suppose we have a database about a hockey league. It includes a relation
Players(number, surname, team, coach, goals) and these functional dependencies:

- team, number → surname, goals
- team → coach

Part (a) [1 mark]

This relation violates BCNF. Give a very small instance of the relation that has the characteristic that BCNF decomposition is designed to prevent.

Part (b) [1 mark]

Explain why this instance is a problem.

Part (c) [2 marks]

Show the two relations that would result if the first step of the BCNF algorithm were applied to decompose table Players.

Part (d) [2 marks]

Show how the information in your example from part (a) would be represented within this new schema.
Question 9.  [6 marks]
Consider relation $R(A, B, C, D, E, F)$ with functional dependencies $S$.

$$S = \{ABD \rightarrow EF, \ ABE \rightarrow D, \ ABEF \rightarrow C, \ B \rightarrow AD, \ BCDE \rightarrow A, \ F \rightarrow AC\}$$

Compute a minimal basis for $S$.  

Question 10. [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:

$$ S = \{ A \rightarrow EF, \ CDF \rightarrow E, \ E \rightarrow BCD \} $$

Part (a) [2 marks]
Compute all keys for $R$. Show your rough work and explain why your list of keys is complete.

Part (b) [2 marks]
Show the final schema produced by the 3NF algorithm. Explain your answer in terms of the steps of the algorithm. Do not project the functional dependencies onto the relations, just show the relations.
Question 11. [10 marks]
Assume the following XML document amazon.xml:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<br书记store>
  <book isbn="1111" category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book isbn="2222" category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book isbn="3333" category="WEB">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanatham Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book isbn="4444" category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>69.95</price>
  </book>
</bookstore>
```
For each of the following XPath expressions, write the output of the expression.


2. \texttt{fn:doc("amazon.xml")/descendant::book[@isbn="4444"]/preceding-sibling::book/year}

3. \texttt{count(fn:doc("amazon.xml")//price/parent::book[@isbn="3333"]/author)}

4. \texttt{avg(fn:doc("amazon.xml")//book[1]/(price | year))}

5. \texttt{fn:doc("amazon.xml")//book/price[ancestor::book[@category="WEB"]]}

...
**Question 12.**  [10 marks]

Assume an additional XML document `bn.xml` about the Barnes and Noble bookstore:

```xml
<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

<book isbn="3333" category="WEB">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>59.99</price>
</book>

<book isbn="4444" category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>49.95</price>
</book>

<book isbn="5555" category="WEB">
    <title lang="en">Data on the Web</title>
    <author>Serge Abiteboul</author>
    <author>Peter Buneman</author>
    <author>Suciu Dan</author>
    <year>2000</year>
    <price>59.95</price>
</book>

</bookstore>
```
For each of the following XQuery expressions, write the output of the expression.

1. for $x$ in fn:doc("amazon.xml")//book
   where $x$/price < 50
   order by $x$/@category descending
   return
       if ($x$/@category = "CHILDREN")
           then <child>{$x$/title/text()}</child>
           else <adult>{$x$/title/text()}</adult>

2. for $a$ in fn:doc("amazon.xml")//book
   for $b$ in fn:doc("bn.xml")//book
   where $a$/@isbn = $b$/@isbn
   order by $a$/title descending
   return
       <title>{ $a$/title/text() }</title>

3. for $a$ in fn:doc("amazon.xml")//book
   for $b$ in fn:doc("bn.xml")//book
   where $a$/@isbn = $b$/@isbn and $a$/price < $b$/price
   return
       <book>
           <title>{ $a$/title/text() }</title>
           <price-amazon>{ $a$/price/text() }</price-amazon>
           <price-bn>{ $b$/price/text() }</price-bn>
       </book>
4. for $c$ in distinct-values(fn:doc("amazon.xml")//book/@category)
   let $amazonc := count(fn:doc("amazon.xml")//book[@category=$c])
   let $bnc := count(fn:doc("bn.xml")//book[@category=$c])
   order by $bnc descending
   return
       <category>{$c}
           <amazon>{$amazonc}</amazon>
           <bn>{$bnc}</bn>
       </category>

5. <ul>
   { for $b$ in fn:doc("bn.xml")//book
     order by $b/year ascending, $b/price descending
     return
         <li>{$b/title/text()}</li>
   }
</ul>
Question 13. [10 marks]
Recall the two XML documents about bookstores presented in the two previous Questions (i.e., amazon.xml and bn.xml). Write XQuery expressions to answer the following questions.

XQuery 1: Find the title(s) of the book(s) in Amazon that belong to the "WEB" category. Sort the results in ascending order according to book title.

XQuery 2: Find the title(s) of the book(s) in Amazon whose price is larger than the average price of all books in the Barnes and Noble bookstore. Sort the results in descending order according to book title.
Question 14. [5 marks]
Your company is asked to design a simple Entity-Relationship Model (ER Diagram) about tourists visiting cities of the world. The entities (rectangles) of the E-R diagram, their relationships (diamonds) and attributes (ovals) have already been provided. Your task is to determine the multiplicity of each of the relationships below (StaysIn, CitizenOf, Visits, belongsTo, LocatedIn). To help you construct the right answer, you are asked to first indicate the cardinalities with which an entity participates in a relationship based on the following assumptions about the domain:

- A city belongs in exactly one country, but a country can have any number of cities.
- A hotel is located in exactly one city, but a city can have any number of hotels.
- A tourist is a citizen of exactly one country, can visit a city any number of times and can stay in the same hotel any number of times.

Rough Work
As rough work, fill in the gaps inside the empty parentheses below. Recall that each parenthesis defines the minimum and maximum cardinality (min, max) with which an entity participates in a relationship.
Your Answer
To complete your answer, fill in the table below by defining the multiplicity of each relation in your E-R diagram. Recall that the multiplicity of a relationship can be one of: 1-1, 1-N or N-N.

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>StaysIn</td>
<td>–</td>
</tr>
<tr>
<td>CitizenOf</td>
<td>–</td>
</tr>
<tr>
<td>Visits</td>
<td>–</td>
</tr>
<tr>
<td>belongsTo</td>
<td>–</td>
</tr>
<tr>
<td>LocatedIn</td>
<td>–</td>
</tr>
</tbody>
</table>
Question 15. [5 marks]
Translate the Entity-Relationship Model (ER Diagram) given below into a logical model (database schema). For each relation in your schema, indicate its name, attributes, keys (underlined) and foreign keys (bold or circled).

![Diagram of Entity-Relationship Model](image)

Figure 1: Entity-Relationship Diagram
[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.]
[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.]
Below is a copy of the schema from page 2. Nothing on this page will be graded. You may tear this page off for reference.

Relations

- Mentorship(year, sponsor, budget)
  A tuple in this relation represents information about a single year of the mentorship program: the year, the company that sponsored the program, and the budget for the program.

- Mentor(MID, name, email, employer, title, phone)
  A tuple in this relation represents information about a mentor.

- MApplication(MID, year, capacity)
  A tuple in this relation indicates that mentor MID applied to be a mentor in a given year and they were willing to mentor up to capacity students.

- Student(SID, name, email, phone)
  A tuple in this relation represents information about a student.

- SApplication(SID, year, cgpa)
  A tuple in this relation indicates that a student applied to be mentored in a given year and had the given cgpa at the time.

- Expertise(MID, year, area)
  A tuple represents an area, such as “web development” that a mentor declared expertise in for a particular year.

- Interest(SID, year, area, priority)
  A tuple represents an area that a student has declared an interest in for a particular year. The attribute priority indicates how strong the student’s interest in that area is, with 1 being the highest priority.

- Match(MID, SID, year)
  A tuple indicates that the mentor MID was matched with student SID in the given year.

Integrity constraints

The obvious integrity constraints apply:

\[
\begin{align*}
MApplication[MID] & \subseteq Mentor[MID] & MApplication[year] & \subseteq Mentorship[year] \\
SApplication[SID] & \subseteq Student[SID] & SApplication[year] & \subseteq Mentorship[year] \\
Expertise[MID] & \subseteq Mentor[MID] & Expertise[year] & \subseteq Mentorship[year] \\
Interest[SID] & \subseteq Student[SID] & Interest[year] & \subseteq Mentorship[year] \\
Match[MID] & \subseteq Mentor[MID] & Match[SID] & \subseteq Student[SID] \\
Match[year] & \subseteq Mentorship[year] \\
\end{align*}
\]

Total Marks = 101