Question 1. [7 marks]

Recall this schema, which we have used many times in class. Here we are adding one more relation called Club. It records the university clubs that students have joined (if any). For example, if a student joins the ‘Chess’ club, a tuple would appear in table Club indicating that this student is part of the Chess club.

Relations

Student(sID, surName, firstName, campus, email, cgpa)
Course(dept, cNum, name, breadth)
Offering(oID, dept, cNum, term, instructor)
Took(sID, oID, grade)
Club(sID, clubName)

Integrity constraints

Offering[dept, cNum] ⊆ Course[dept, cNum]
Took[sID] ⊆ Student[sID]
Took[oID] ⊆ Offering[oID]
Club[sID] ⊆ Student[sID]

Part (a) [4 marks]

Write a query to find the sIDs of students at campus ‘UTM’ who are part of both the Chess club and the Drama club. Use only the basic operators Π, σ, ⊸, ×, ∩, ∪, −, ρ, and assignment.

Solution:

\[ DramaUTM(sID) := \Pi_{SID} \sigma_{\text{clubName} = 'Drama' \land \text{campus} = 'UTM'}(\text{Club} \bowtie \text{Student}) \]

\[ ChessUTM(sID) := \Pi_{SID} \sigma_{\text{clubName} = 'Chess' \land \text{campus} = 'UTM'}(\text{Club} \bowtie \text{Student}) \]

\[ Answer(sID) := DramaUTM \cap ChessUTM \]
Part (b) [3 marks]

Consider the following query:

\[
\text{Apple}(sID, \text{clubName}) := \Pi_{sID} \text{Student} \times \Pi_{\text{clubName}} \text{Club}
\]

\[
\text{Orange}(sID) := \Pi_{sID}(\text{Apple} - \text{Club})
\]

\[
\text{Answer}(sID) := (\Pi_{sID} \text{Student} - \text{Orange}) \bowtie \Pi_{sID}(\sigma_\text{campus='UTM'} \text{Student})
\]

You are given below, instances of the relations that are relevant to this query. Add or remove the minimum number of rows to these relations so that student 1111 does appear in the result and students 2222 and 3333 do not:

<table>
<thead>
<tr>
<th>sID</th>
<th>clubName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>dance</td>
</tr>
<tr>
<td>2222</td>
<td>chess</td>
</tr>
<tr>
<td>2222</td>
<td>reading</td>
</tr>
<tr>
<td>3333</td>
<td>dance</td>
</tr>
<tr>
<td>3333</td>
<td>chess</td>
</tr>
<tr>
<td>3333</td>
<td>reading</td>
</tr>
<tr>
<td>2222</td>
<td>dance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sID</th>
<th>surName</th>
<th>firstName</th>
<th>campus</th>
<th>email</th>
<th>cgpa</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>Smith</td>
<td>Allen</td>
<td>UTM</td>
<td>NULL</td>
<td>90</td>
</tr>
<tr>
<td>2222</td>
<td>Babaei</td>
<td>Ali</td>
<td>UStG</td>
<td><a href="mailto:b@uoft.com">b@uoft.com</a></td>
<td>85</td>
</tr>
<tr>
<td>3333</td>
<td>Campbell</td>
<td>eric</td>
<td>UTM</td>
<td><a href="mailto:c@uoft.com">c@uoft.com</a></td>
<td>70</td>
</tr>
</tbody>
</table>

**Solution:** 2222 is not in the result; there is nothing to do. For student 1111 add two rows with clubs chess and reading to Club relation. Remove any row of 3333 from relation Club.

**Question 2.** [4 marks]

**Part (a) [2 marks]**

In the schema of question 1, relation \( \text{Club}(sID, \text{clubName}) \) records the university clubs that students have joined (if any). Assuming that instead of making a separate \( \text{Club} \) relation, if we add a column for \( \text{clubName} \) in the \( \text{Student} \) relation, which statement is correct? Circle one answer:

- A student is the member of exactly one club.
- A student can join utmost one club.
- As student can join as many clubs as he/she is interested in.

**Solution:** A student is the member of exactly one club.

**Part (b) [2 marks]**

Consider this schema:

\[
\text{One}(a, b, c)
\]

\[
\text{Two}(b, e)
\]

Assuming relation \( \text{One} \) has 100 tuples and relation \( \text{Two} \) has 80 tuples, how many tuples could relation \( \text{One} \bowtie \text{Two} \) have? Circle all possible answers.

| 0 | 100 | 1 | 80 |

**Solution:** 1, 100, 80
Question 3. [6 marks]

Recall this schema, which we have used many times in class. Here we are adding a few more relations:

- **Residence:** all the student residences from all 3 campuses.
- **LivesInRes:** which students live in these residences.
- **AppliedToRes:** the current new applications to live in residence, either from students who do not yet have a spot in a residence, or from those who wish to switch residences. For example, if a student applies for the ‘New College’ residence, then a new application entry will be recorded in ‘AppliedInRes’, mapping the student’s sID to the rID for the residence with rname = ‘New College’.

**Relations**

- Student(sID, surName, firstName, campus, email, cgpa)
- Course(dept, cNum, name, breadth)
- Offering(oID, dept, cNum, term, instructor)
- Took(sID, oID, grade)
- Residence(rID, rname, address, campus)
- LivesInRes(sID, rID)
- AppliedToRes(aID, rID, sID)

**Integrity constraints**

- Offering[dept, cNum] ⊆ Course[dept, cNum]
- Took[sID] ⊆ Student[sID]
- Took[oID] ⊆ Offering[oID]
- LivesInRes[sID] ⊆ Student[sID]
- LivesInRes[rID] ⊆ Residence[rID]
- AppliedToRes[sID] ⊆ Student[sID]
- AppliedToRes[rID] ⊆ Residence[rID]

Please note that:
- It is possible that some students do not live in any residence at all, in any campus (e.g., if they rent their own place off-campus, or live with parents).
- In the Student relation, the ‘campus’ field represents the campus where the student is registered. A student can be registered in one campus, but live in a residence located on a different campus (see the ‘campus’ field from the Residence relation).

Write a query to find the sIDs of students who have applied to residences on at least 2 different campuses.

**Solution:**

SELECT sID FROM AppliedToRes A1, AppliedToRes A2, Residence R1, Residence R2
WHERE A1.sID = A2.sID AND A1.rID <> A2.rID AND A1.rID = R1.rID AND A2.rID = R2.rID AND R1.campus <> R2.campus;

(Note that many other possible solutions exist for these SQL questions.)
Question 4. [15 marks]

For this question, we will use the same schema from Question 3.

Part (a) [3 marks]

Indicate below if there are any items in the Having clause that are invalid. Circle only the illegal ones, if any on the query below.

```
SELECT count(*), rID
FROM Student S, AppliedToRes A
WHERE S.sID = A.sID
GROUP BY rID
HAVING avg(cgpa) > 50 AND aID > 0 AND surName LIKE 'S%'
ORDER BY count(*);
```

Solution: Only avg is valid.

Part (b) [4 marks]

For this question, please assume that all students live in a residence on campus.

Create a view called LocalStudent(sID, campus), which contains all students who are registered in the same campus as the one they live in (the one where their home residence is located). The fields for this View represent the student’s sID, and their campus.

Solution: CREATE VIEW LocalStudent(sID, campus) AS
SELECT S.sID, S.campus
FROM Student S, Residence R, LiveInResidence L
WHERE S.sID = L.sID AND R.rID = L.rID
AND S.campus = R.campus;
Part (c)  [4 marks]

Create a view called nonLocalStudent(sID), containing those students who do not live on the same campus where they’re registered in. You may reuse the view from Part (b) if you wish.

Solution:

CREATE VIEW nonLocalStudent(sID) AS
(SELECT sID FROM Student) EXCEPT (SELECT sID FROM LocalStudent);

Part (d)  [4 marks]

Out of the students from the view in (c), find the sIDs for those students who applied to switch residences, but only applied to residences on the same campus where they’re registered in. You must use the view you created in (c) for this query.

Solution: SELECT N.sID FROM nonLocalStudent N, AppliedToRes A, Residence R
WHERE N.sID = A.sID and R.rID = A.rID
AND R.campus IN
(SELECT campus FROM Student S
WHERE S.sID = N.sID);