

# Assignment 1

- Due Feb 11 by 11:59p.m.
- Points 0

## SmartCity Initiative: Assignment 1

**Due Date: February 11**

**Points: 95**

**Last change: Feb 7 (highlighted in yellow)**

### Context



In this assignment, you are a Data Analyst Intern for the SmartCity Initiative Team. The team is working on projects for Urban Connect 2.0, a next-generation smart city infrastructure system. Your supervisor has outlined some schemas and queries that need to be completed using relational algebra (RA) and SQL.

### Instructions & Important Notes

#### Group Work

- **Group Size:** You must work in a group of **two or three**.
- **Group Declaration:** Ensure all group members are declared in MarkUs before submission.

#### Relational Algebra:

- **Type** (do not handwrite) your RA queries in **A1.txt** file.
- Use the **relational algebra notation** taught in class. These notations should work on [ReLax](https://dbis-uibk.github.io/relax/calc/local/uibk/local/0)  [\\_\(https://dbis-uibk.github.io/relax/calc/local/uibk/local/0\)\\_](https://dbis-uibk.github.io/relax/calc/local/uibk/local/0) where your solutions will be run. If your solutions do not run on [ReLax](https://dbis-uibk.github.io/relax/calc/local/uibk/local/0)  [\\_\(https://dbis-uibk.github.io/relax/calc/local/uibk/local/0\)\\_](https://dbis-uibk.github.io/relax/calc/local/uibk/local/0), you will receive a 0.
- Each query should **include a comment that states the question**.
- If a query cannot be expressed in RA, write **"Cannot be done in RA"** and explain why.
- Break down complex queries into smaller queries and assign those to a variable. The names of the variables should be appropriate. Provide **explanations** for complex RA queries to help the TAs understand your solution.
- Example: Given relations Student, Courses, Enrollments consider the following task:

**Task:** Find the names of all students enrolled in courses with more than 3 credits.

### Step-by-Step Relational Algebra Query:

Extract courses with more than 3 credits:

$\text{HighCreditCourses} = \sigma_{\text{credits} > 3}(\text{Courses})$

Join students with courses they are enrolled in:

$\text{StudentCourseJoin} = \text{Students} \bowtie \text{Enrollments}$

Join the result with high-credit courses:

$\text{HighCreditEnrollments} = \text{StudentCourseJoin} \bowtie \text{HighCreditCourses}$

Project only student names:

$\pi_{\text{name}}(\text{HighCreditEnrollments})$

### SQL:

- Save your SQL queries in a file named **A1.sql**.
- Create tables that match the **defined schemas**.
- Ensure your **SQL queries compile and execute correctly on SQLite**. Queries that fail to run will receive a mark of 0.
- If you define any views for a question (you are encouraged to), you must drop them
- Each query should **include a comment that states the question**.
- SQL queries **must return results matching the provided test data** to earn full marks.

### Remaining Questions:

- Save your answers to questions **1, 4, 5, and 6** in a PDF file named **A1.pdf**.

### Acknowledgment Statement:

Each group must include a paragraph in the **A1.pdf** file that:

1. Clearly states who worked on which parts of the assignment.
2. Confirms that all work was completed by the group members without external help.

Example:

*"We, [Student 1 Name], [Student 2 Name] and [Student 3 Name], confirm that all parts of this assignment were completed collaboratively and without external assistance. [Student 1 Name] was primarily responsible for questions X and Y, while [Student 2 Name] focused on questions Z and W and [Student 3 Name] focused on questions P and Q. We shared responsibilities for testing and proofreading."*

### Submission Guidelines:

- **Testing Requirements:**
  - Test all Relational Algebra solutions on **ReLax**.
  - Test all SQL solutions on **SQLite**.
- **No Schema Submission:**
  - Do not submit the schema. The provided schema will be used to test your solutions.
- **Handwritten Solutions:** Will not be marked.
- **File Submission:** Submit the following three files:
  - **A1.pdf:** Contains Q1, Q4, Q5, Q6, and acknowledgement statement .
  - **A1.txt:** Contains all RA queries (Q2).
  - **A1.sql:** Contains all SQL queries (Q3).
- **Late Submissions:**
  - Late submissions will incur a penalty of **10% per day**, up to **3 days** past the deadline.

## Schema Definitions

1. **Buildings (building\_id: integer, name: string, address: string, city: string, height: real)**  
A tuple in this relation represents a building in the city.
2. **Rooms (room\_id: integer, building\_id: integer, floor: integer, capacity: integer)**  
A tuple in this relation represents a room in a specific building.
3. **Sensors (sensor\_id: integer, room\_id: integer, type: string, status: string)**  
A tuple in this relation represents a sensor installed in a room.
4. **Readings (reading\_id: integer, sensor\_id: integer, timestamp: text, value: real, reading\_month: number)**  
A tuple in this relation represents a reading from a specific sensor.
5. **Maintenance (maintenance\_id: integer, building\_id: integer, date: text, description: string, maintenance\_month: number)**  
A tuple in this relation represents a maintenance event for a building.
6. **Suppliers (supplier\_id: integer, name: string, city: string)**  
A tuple in this relation represents a supplier.
7. **Supplies (supply\_id: integer, supplier\_id: integer, building\_id: integer, cost: real)**  
A tuple in this relation represents resources supplied to a building by a supplier.
8. **Tasks (task\_id: integer, building\_id: integer, task\_type: string, priority: string)**

A tuple in this relation represents a task assigned to a building.

9. **Personnel** (**person\_id**: integer, **name**: string, **role**: string, **contact**: string)

*A tuple in this relation represents personnel responsible for maintenance tasks.*

10. **PersonnelAssignment** (**person\_id**: integer, **maintenance\_id**: integer)

*A tuple in this relation assigns personnel to specific maintenance events.*

## Questions

### Q1. Convert to English (10 points)

Explain the following relational algebra queries in plain English. Be precise in your wording and include terms like "all," "every," or "at least one" where appropriate. If a query cannot be expressed, clearly state the reason.:

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

## Q2. Relational Algebra Queries (36 points)

Provide **relational algebra queries** for the following tasks:

If a task cannot be expressed in relational algebra, write "Cannot be done in RA" and justify why.

1. Retrieve the names of buildings that have at least one active sensor.
2. Retrieve the IDs of buildings with rooms on every floor between 1 and 10 inclusive.
3. Retrieve the IDs of buildings with tasks of all priority levels (Assume there are three priority levels: High, Medium, Low).
4. Assume there are three buildings with IDs 1, 2, and 3 among other buildings. Find the names of buildings out of the three where every room has at least one sensor installed.
5. Retrieve the IDs of suppliers providing resources to every building in Toronto.
6. Retrieve the names of suppliers who supply resources to all buildings that have high-priority tasks assigned.

7. Retrieve all rooms that have at least two sensors of different types installed.
8. Find all buildings that have both inactive sensors and a scheduled maintenance event in the same month.
9. Retrieve the IDs of rooms in Toronto where the total capacity of two rooms exceeds 500. Each pair should have unique rooms.
10. Retrieve the ID of the tallest building such that sensors installed in rooms on every floor. Assume there is only one tallest building.
11. Find the names of personnel responsible for maintenance tasks in buildings where resources were supplied by suppliers located in the same city, and the building has at least one high-priority task assigned.
12. Retrieve the IDs of buildings where at least one sensor has recorded a reading above 75 after 23rd October 2024, the buildings have had maintenance tasks in the same period, and they received supplies from suppliers located in a different city.

### Q3. SQL Queries (36 points)

For each part in Q2, write the corresponding **SQL query**. Ensure that:

- The schema matches the one provided above.
- Your queries execute correctly in SQLite.

Save all your RA queries in a file named **A1.txt** and all SQL queries in a file named **A1.sql**.

### Q4. Primary Key and Foreign Key Relationships (5 points)

Identify the **primary key** and **foreign key** relationships for each schema defined above.

### Q5. Revise the Schema (2 points)

In your opinion, this database schema is bad. Show off your skills to the manager and rewrite the schema. There are no right or wrong answers. Marks are based on the quality of answers, and any theories, guidelines or practices you've incorporated. Feel free to take some time to research the topic but remember to cite your sources.

Provide:

1. The **revised schema**.
2. **Reasons** for your decisions.

**Q6. Bonus (1 point)**

Create a **funny meme or short story** about your interaction with the smart city manager. If it makes the TA grin, you'll get the bonus mark!

**Styling marks (5 points)**

**Styling of your submission will be evaluated and carries 4 points.**

**Please follow submission instructions mentioned at the beginning.**

**Submit your assignment (A1.txt, A1.sql, A1.pdf) on MarkUs.**