

Marissa Ratschki

CMPT\_308

Lab 02

GITHUB REPO: [mratschki/CMPT\\_308N\\_112: Repo for Database Design](https://github.com/mratschki/CMPT_308N_112)

## PART A

1. A relation is a table within a relational database, that where rows represent records and columns represent a rows attribute
2. A tuple is a singular row within a relation (think individual tables in ER diagram).
3. Attributes are represented in columns of the relation and describe a property of the data in the table.
4. Domains represent a set of allowed values to be used for an attribute (for ex. negative integers).
5. A schema is the stated structure of a database (including tables, columns, datatypes). An instance is an instance of data stored in the database at a given moment.
6. Keys are attributes or combinations of attributes that uniquely ID a tuple within a relation. Primary keys enforce data integrity by ensuring that no duplicate rows exist.

## PART B

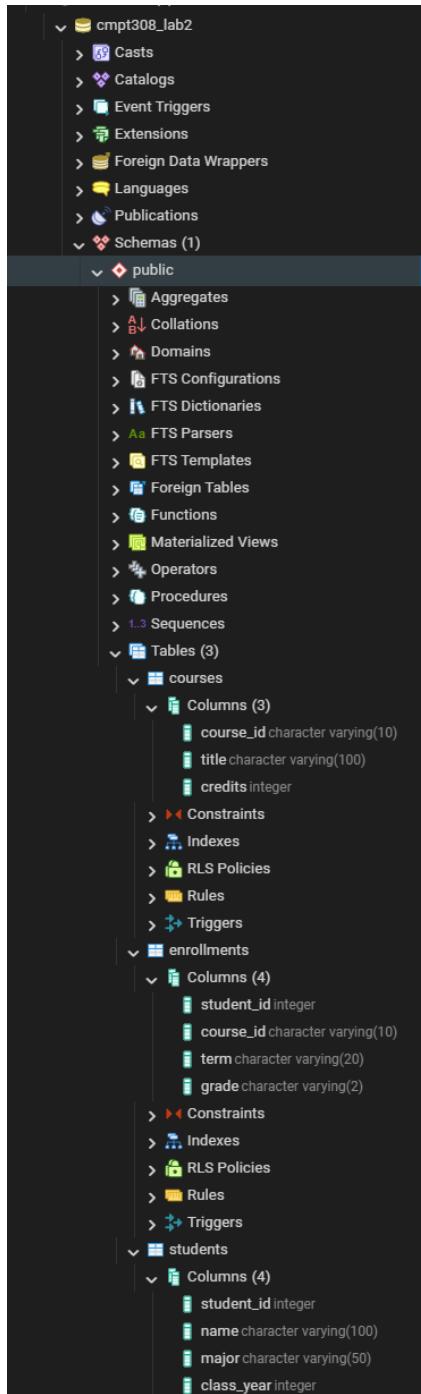


Figure 1: pgAdmin showing database and the three tables

```

4   CREATE TABLE Students (
5     student_id INT PRIMARY KEY,
6     name VARCHAR(100) NOT NULL,
7     major VARCHAR(50) NOT NULL,
8     class_year INT NOT NULL
9   );
10
11  CREATE TABLE Courses (
12    course_id VARCHAR(10) PRIMARY KEY,
13    title VARCHAR(100) NOT NULL,
14    credits INT NOT NULL CHECK (credits > 0)
15  );
16
17  CREATE TABLE Enrollments (
18    student_id INT,
19    course_id VARCHAR(10),
20    term VARCHAR(20) NOT NULL,
21    grade VARCHAR(2),
22
23    PRIMARY KEY (student_id, course_id, term),
24
25    FOREIGN KEY (student_id)
26      REFERENCES Students(student_id),
27
28    FOREIGN KEY (course_id)
29      REFERENCES Courses(course_id)
30  );
31

```

Figure 2: CREATE TABLE statements

## PART C

The screenshot shows a SQL query editor interface. The top section is labeled "Query History" and contains the following SQL code:

```

1 SELECT student_id, name
2 FROM Students;
3

```

The bottom section is labeled "Data Output" and displays the results of the query. The results are presented in a table with two columns: "student\_id" and "name". The data is as follows:

	student_id	name
1	29	Mila
2	57	John
3	48	Julia
4	25	Artyom
5	34	Lucy
6	55	Tyler

Figure 3: Projection query

Query    Query History

```
1 SELECT *
2 FROM Students
3 WHERE major = 'CMPT-308';
4
```

Data Output    Messages    Notifications

Showing rows: 1 to 2     | Page No: 1 of 1 |

	student_id [PK] integer	name character varying (100)	major character varying (50)	class_year integer
1	57	John	CMPT-308	2027
2	25	Artyom	CMPT-308	2026

Figure 4: Selection Query

Query    Query History

```
1 SELECT *
2 FROM Courses
3 WHERE credits >= 3;
4
```

Data Output    Messages    Notifications

Showing rows: 1 to 4     | Page No: 1 of 1 |

	course_id [PK] character varying (10)	title character varying (100)	credits integer
1	CMPT-423	Hacking and Pentesting	3
2	CMPT-308	Database Management	4
3	CMPT-466	Special Topics	4
4	CMPT-307	Internetworking	4

Figure 5: Multiple conditions query

A screenshot of a SQL query interface. The query window shows the following code:

```
1 SELECT *
2 FROM Students
3 WHERE name LIKE 'M%';
```

The results table has four columns: student\_id, name, major, and class\_year. One row is displayed:

	student_id [PK] integer	name character varying (100)	major character varying (50)	class_year integer
1	29	Mila	CMPT-423	2026

Figure 6: Like query

A screenshot of a SQL query interface. The query window shows the following code:

```
1 SELECT *
2 FROM Enrollments
3 WHERE grade IS NULL;
```

The results table has five columns: student\_id, course\_id, term, and grade. Two rows are displayed:

	student_id [PK] integer	course_id [PK] character varying (10)	term [PK] character varying (20)	grade character varying (2)
1	57	CMPT-466	Spring 2025	[null]
2	48	CMPT-308	Spring 2065	[null]

Figure 7: NULL query

*Figure 8: Order by query*

## PART D

The screenshot shows a database interface with a query editor and a results table. The query is:

```
1 SELECT student_id, name
2 FROM Students;
3
```

The results table has two columns: student\_id and name. The data is:

	student_id [PK] integer	name character varying (100)
1	29	Mila
2	57	John
3	48	Julia
4	25	Artyom
5	34	Lucy
6	55	Tyler

Relational algebra:  $\pi$  student\_id, name (Students)

The screenshot shows a database interface with a query editor and a results table. The query is:

```
1 SELECT *
2 FROM Students
3 WHERE major = 'CMPT-308';
4
```

The results table has four columns: student\_id, name, major, and class\_year. The data is:

	student_id [PK] integer	name character varying (100)	major character varying (50)	class_year integer
1	57	John	CMPT-308	2027
2	25	Artyom	CMPT-308	2026

Relational algebra:  $\sigma$  major = 'CMPT-308' (Students)