

Lab 2: relational model + SQL basics

Part A:

1. A relation is rows & columns
2. A tuple is a row
3. Each column is an attribute.
An attribute is usually made up by the standard English alphabet
4. A domain are the allowed names for an attribute (type + rule)
5. A schema is the design of tables.
Table names, column names → relations
- A instance is the actual data

6. A key is a set of attributes that uniquely identifies a tuple.

} ex) for the database created cmp308_lab2
• the schema holds the tuples being created with students, courses & enrollments.
• the instance would be a segment of the data → "102,Seema,"database..."

Part B:)

```
CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    name VARCHAR(100),
    major VARCHAR(50),
    class_year INT
);

CREATE TABLE Courses (
    course_id INT PRIMARY KEY,
    title VARCHAR(100),
    credits INT CHECK (credits > 0)
);

CREATE TABLE Enrollments (
    student_id INT,
    course_id INT,
    term VARCHAR(20),
    grade VARCHAR(2),
    PRIMARY KEY (student_id, course_id, term),
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);
```

```
SELECT * FROM public.students
ORDER BY student_id ASC
```

student_id	name	major	class_year
1001	Seema	Computer Science	2027
1002	Sienna	Psychology	2027
1003	Matt	Education	2026
1004	Marina	Fashion	2025
1005	Jeffrey	Pre Law	2028
1006	Jose	Mathematics	2029

pgAdmin 4 Object Tools Edit View Window Help pgAdmin 4

Servers (1) PostgreSQL 18 Databases (3) NinjaSchool cmp308_lab2 Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1) public Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Functions Materialized Views Operators Procedures Sequences Tables (3) courses enrollments students Trigger Functions Types Views Subscriptions postgres Login/Group Roles Tablespaces

Query History

```
1 SELECT * FROM public.courses
2 ORDER BY course_id ASC
```

Data Output Messages Notifications

course_id	title	credits
1	English	3
2	Database Management	4
3	Discrete Mathematics	4
4	Psychology	3

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

Total rows: 4 Query complete 00:00:00.409 LF Ln 1, Col 1

pgAdmin 4 Object Tools Edit View Window Help pgAdmin 4

Servers (1) PostgreSQL 18 Databases (3) NinjaSchool cmp308_lab2 Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas (1) public Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Functions Materialized Views Operators Procedures Sequences Tables (3) courses enrollments students Trigger Functions Types Views Subscriptions postgres Login/Group Roles Tablespaces

Query History

```
1 SELECT * FROM public.enrollments
2 ORDER BY student_id ASC, course_id ASC, term ASC
```

Data Output Messages Notifications

student_id	course_id	term	grade
1001	222	Spring 2026	[null]
1001	333	Spring 2026	B
1002	133	Fall 2025	A
1002	444	Spring 2026	[null]
1003	133	Fall 2025	A
1004	133	Spring 2026	B
1005	333	Fall 2025	C
1006	333	Spring 2026	A

Showing rows: 1 to 8 Page No: 1 of 1

Total rows: 8 Query complete 00:00:00.200 LF Ln 1, Col 1

PART C:

1. projections:

The screenshot shows the pgAdmin 4 interface with the title bar "pgAdmin 4" and the status bar "Wed Feb 4 10:11AM". The left sidebar displays the database structure for "NinaSchool" and "cmp308_lab2". The main query editor window contains the following SQL query and its results:

```
1 SELECT student_id FROM Students
```

student_id
1001
1002
1003
1004
1005
1006

Total rows: 6 Query complete 00:00:00.383 LF Ln 1, Col 33

2. selection:

The screenshot shows the pgAdmin 4 interface with the title bar "pgAdmin 4" and the status bar "Wed Feb 4 10:11AM". The left sidebar displays the database structure for "NinaSchool" and "cmp308_lab2". The main query editor window contains the following SQL query and its results:

```
1 SELECT * FROM Students  
2 Where major = 'Computer Science'
```

student_id	name	major	class_year
1001	Seema	Computer Science	2027

Total rows: 1 Query complete 00:00:00.143 LF Ln 2, Col 33

3. multiple conditions :

The screenshot shows the pgAdmin 4 interface with a database tree on the left and a query editor on the right. The query is:

```
SELECT * FROM Courses  
Where credits >= 3
```

The results table shows four rows of course data:

course_id	title	credits
333	Discrete Mathematics	4
222	Database Management	4
133	English	3
444	Psychology	3

Total rows: 4 Query complete 00:00:00.122

4. pattern matching (like)

The screenshot shows the pgAdmin 4 interface with a database tree on the left and a query editor on the right. The query is:

```
SELECT name FROM Students  
Where name LIKE 'Si%';
```

The results table shows two rows of student names:

name
Sema
Sienna

Total rows: 2 Query complete 00:00:00.260

5. null check

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure under "NinjaSchool" and "cmp308_lab2".
- Query Editor:** Contains the SQL query:

```
1 SELECT * FROM Enrollments
2 WHERE grade IS NULL;
```
- Data Output:** Displays the results of the query:

	student_id	course_id	term	grade
1	1001	222	Spring 2026	[null]
2	1002	444	Spring 2026	[null]
- Information:** Total rows: 2, Query complete 00:00:00.125.

6. order by class_year, name

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure under "NinjaSchool" and "cmp308_lab2".
- Query Editor:** Contains the SQL query:

```
1 SELECT * FROM Students
2 ORDER BY class_year, name;
```
- Data Output:** Displays the results of the query:

	student_id	name	major	class_year
1	1004	Marina	Fashion	2025
2	1003	Matt	Education	2026
3	1001	Seema	Computer Science	2027
4	1002	Sienna	Psychology	2027
5	1005	Jeffrey	Pre Law	2028
6	1006	Jose	Mathematics	2029
- Information:** Total rows: 6, Query complete 00:00:00.320.

part 0: relational algebra

- sigma σ — "where"
- projection π — "select"
- join = \bowtie — "multiple tables"

ex) select name from students

WHERE major = 'cs'

vt students (major='cs'), name)

#1 $\pi \leftarrow \text{FROM public.students} \rightarrow \pi(\text{students})$
order by student_id ASC

#2 $\eta \leftarrow \text{FROM Enrollments} \rightarrow \eta (\sigma \text{ grade is NULL} (\text{Enrollments}))$
 $\sigma \text{ grade is NULL;}$