

Patrick Murray

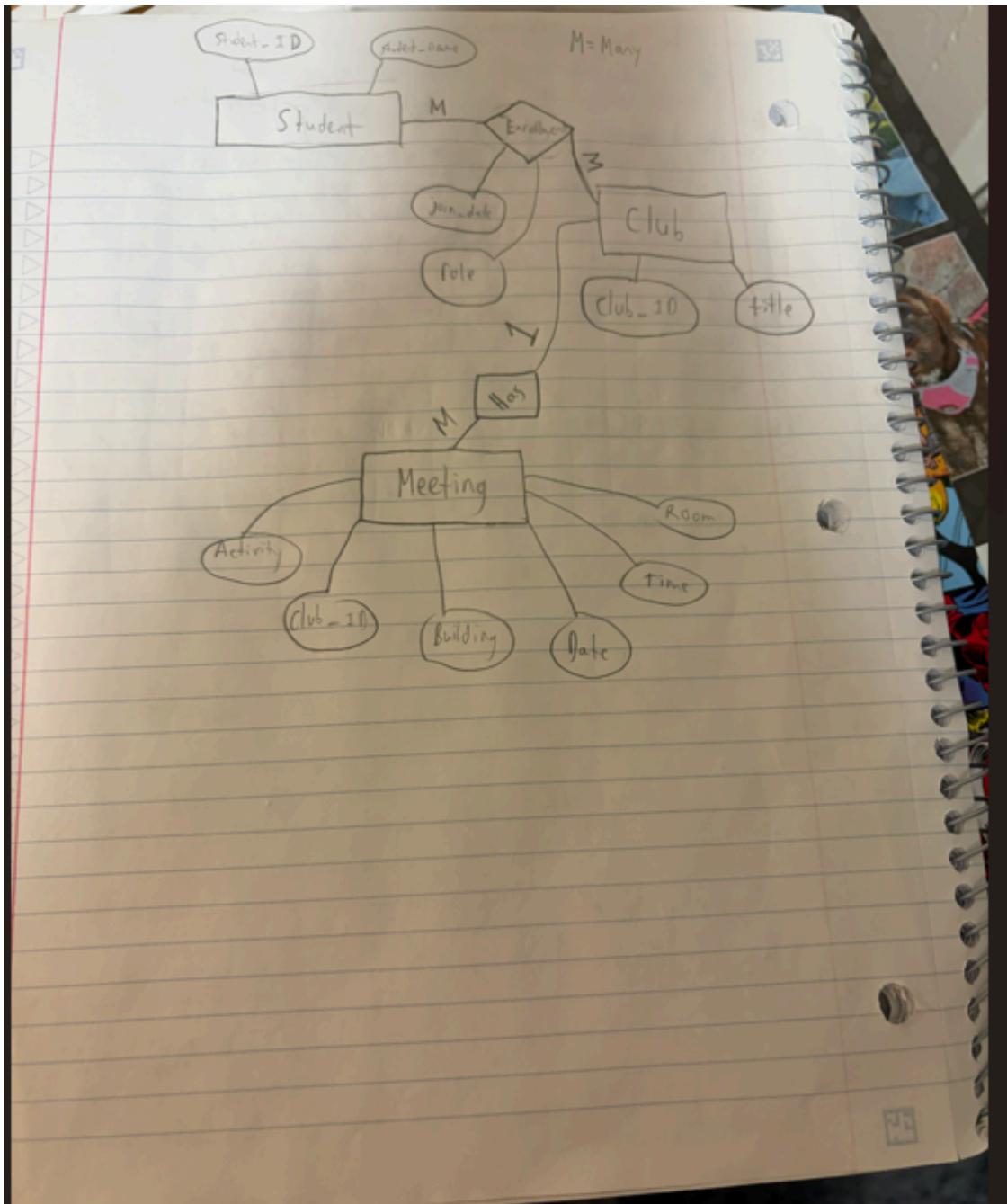
Professor Rivas

Database Management

19 February 2026

Github link: [Not-Patrick1/DatabaseLabs: First lab for DB management.](#)

Part A:



Part B:

Table 1: Student

- Primary Key: student_ID (integer)
- Key: student_Name (text)

Table 2: Club

- Primary Key: club_ID (integer)
- Key: club_Name (text)

Table 3: Enrollment (Join table between Student and Club)

- Primary Key, Foreign Key: student_ID (integer). References the student ID from the Student table.
- Primary Key, Foreign Key: club_ID (integer). References the club ID from the Club table.
- Primary Key: join_date (DATE). Stored at the join table to record when the student joined along with their information and the club they joined.
- Key: club_Role (text). Stored at the join table to record more information about the student when they join.

Table 4: Meeting

- Primary Key, Foreign Key: club_ID (integer). References club_ID from the club table to show what club the meeting is for.
- Primary Key: building (text)
- Primary Key: room (integer)

- Primary Key: date (DATE)
- Primary Key: time (TIME)
- Key: activity (text)

Screenshots:

The screenshot shows a MySQL query editor interface. At the top, there are tabs for "Query" and "Query History". The main area displays the following SQL code:

```
1 CREATE TABLE student(
2     student_ID int PRIMARY KEY,
3     student_Name text
4 )
```

Below the code, there are three tabs: "Data Output", "Messages", and "Notifications". The "Messages" tab is selected, showing the output of the query:

CREATE TABLE

Query returned successfully in 134 msec.

Create table student

Query Query History

```
1 CREATE TABLE Club (
2     club_ID int PRIMARY KEY,
3     club_Name text
4 )
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 41 msec.

Create table club

Query Query History

```
1 CREATE TABLE meeting (
2     club_ID int,
3     FOREIGN KEY (club_ID) REFERENCES club (club_ID),
4     building text,
5     room_Number int,
6     meet_Date DATE,
7     meet_Time TIME,
8     activity text,
9     PRIMARY KEY (club_ID, building, room_Number, meet_Date, meet_Time),
10    constraint check_Room CHECK (room_Number<350),
11    constraint check_Time CHECK (meet_Time<'22:00:00')
12 )|
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 38 msec.

Create table meeting

The screenshot shows the MySQL Workbench interface. The top window is titled "Query History" and contains the SQL code for creating the "Enrollment" table. The code includes foreign key constraints referencing the "student" and "club" tables, and a primary key defined by the combination of student_ID, club_ID, and join_Date.

```
1 CREATE TABLE Enrollment(
2     FOREIGN KEY (student_ID) REFERENCES student(student_ID),
3     FOREIGN KEY (club_ID) REFERENCES club (club_ID),
4     student_ID int,
5     club_ID int,
6     join_Date DATE,
7     club_Role text,
8     PRIMARY KEY (student_ID, club_ID, join_Date)
9 )
10 )
11
```

The bottom window is titled "Messages" and displays the execution results:

```
CREATE TABLE
Query returned successfully in 92 msec.
```

Create table enrollment

Next page is Database

▼	Tables (4)
▼	club
▼	Columns (2)
	club_id integer
	club_name text
>	► Constraints
>	Indexes
>	RLS Policies
>	Rules
>	Triggers
▼	enrollment
>	Columns (4)
>	► Constraints
>	Indexes
>	RLS Policies
>	Rules
>	Triggers
▼	meeting
>	Columns (6)
>	► Constraints
>	Indexes
>	RLS Policies
>	Rules
>	Triggers
▼	student
>	Columns
>	► Constraints
>	Indexes
>	RLS Policies
>	Rules
>	Triggers

Two table join

The screenshot shows a SQL query editor interface with a dark theme. At the top, there are tabs for "Query" and "Query History". Below the tabs is a code editor containing the following SQL query:

```
1 SELECT c.club_Name, m.activity
2 FROM club c
3 JOIN meeting m on c.club_ID = m.club_ID;
```

Below the code editor is a "Data Output" section. It includes a toolbar with icons for file operations like new, save, and print, as well as buttons for "SQL" and "Data". To the right of the toolbar, it says "Showing rows: 1 to 8" and "Page No: 1 of 1". There are also navigation arrows for the results.

	club_name	activity
1	Game Society	Super Smash Tournam...
2	Lifting Club	Group Lift
3	Rock Climbi...	Grip Training
4	Lacrosse	Practice
5	ESports Club	Practice
6	Baseball	Practice
7	Outing Club	Waiver Signing
8	Lifting Club	Group Cardio

Three table join

The screenshot shows the pgAdmin 4 interface with a query editor and a data output viewer.

Query Editor:

```
1 SELECT s.student_Name, c.club_Name, e.club_Role
2 FROM student s
3 JOIN enrollment e ON s.student_ID = e.student_ID
4 JOIN club c ON e.club_ID = c.club_ID;
```

Data Output:

	student_name	club_name	club_role
1	Rock Jones	Rock Climbi...	President
2	Greg Wall	Lifting Club	President
3	Amy Morano	Track	[null]
4	Peter Jacobsen	Baseball	[null]
5	Peter Jacobsen	ESports Club	[null]
6	Peter Jacobsen	Lacrosse	Treasurer
7	Don Cheeter	Rock Climb...	[null]
8	Don Cheeter	Game Society	Treasurer

JOIN + filter

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** cmpt308_lab3/postgres@PostgreSQL 18
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query Tab:** Labeled "Query". The current query is:

```
1 SELECT s.student_Name, c.club_Name, e.club_Role
2 FROM student s
3 JOIN enrollment e ON s.student_ID = e.student_ID
4 JOIN club c ON e.club_ID = c.club_ID
5 WHERE e.club_Role = 'President';
6
7
8
9
10 |
```
- Data Output Tab:** Labeled "Data Output". It displays the results of the query:

	student_name	club_name	club_role
1	Rock Jones	Rock Climbing Club	President
2	Greg Wall	Lifting Club	President
- Bottom Bar:** Includes buttons for file operations, a "SQL" button, and navigation controls for rows and pages.

JOIN + projection + SORT

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** cmpt308_lab3/postgres@PostgreSQL 18
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Query Editor:** Shows the following SQL query:

```
1 SELECT s.student_Name, c.club_Name, e.club_Role
2 FROM student s
3 JOIN enrollment e ON s.student_ID = e.student_ID
4 JOIN club c ON e.club_ID = c.club_ID
5 WHERE c.club_Name like '%r%'
6 ORDER BY c.club_Name DESC;
```
- Data Output:** Displays the results of the query in a table format:

	student_name	club_name	club_role
1	Amy Morano	Track	[null]
2	Peter Jacobsen	Lacrosse	Treasurer
3	Peter Jacobsen	ESports Club	[null]
- Messages and Notifications:** No messages or notifications are present.
- Page Navigation:** Shows "Showing rows: 1 to 3" and "Page No: 1 of 1" with navigation icons.

Null Query

The screenshot shows a PostgreSQL client interface with a dark theme. The top bar displays the connection information: 'cmpt308_lab3/postgres@PostgreSQL 18'. Below the bar are various toolbar icons. The main area has two tabs: 'Query' (selected) and 'Query History'. The 'Query' tab contains the following SQL code:

```
1 SELECT s.student_Name, c.club_Name, e.club_Role
2 FROM student s
3 JOIN enrollment e ON s.student_ID = e.student_ID
4 JOIN club c ON e.club_ID = c.club_ID
5 WHERE e.club_Role IS NULL;
```

The 'Data Output' tab is selected at the bottom, showing a table with four rows of data. The columns are labeled 'student_name', 'club_name', and 'club_role'. The data is as follows:

	student_name	club_name	club_role
1	Don Cheetah	Rock Climbing	[null]
2	Peter Jacobsen	Baseball	[null]
3	Peter Jacobsen	ESports Club	[null]
4	Amy Morano	Track	[null]

Check Violation

The screenshot shows a PostgreSQL client interface with a dark theme. The top bar displays the connection information: `cmpt308_lab3/postgres@PostgreSQL 18*`. The main area is a query editor with the following SQL code:

```
1 INSERT INTO meeting (club_ID, building, room_number, meet_date, meet_time, activity)
2 Values(401, 'McCann Center', '166', '03/19/2026', '23:00', 'Group Lift')
3
```

Below the query editor, there are tabs for `Data Output`, `Messages`, and `Notifications`. The `Messages` tab is selected, showing the following error message:

```
ERROR: new row for relation "meeting" violates check constraint "check_time"
Failing row contains (401, McCann Center, 166, 2026-03-19, 23:00:00, Group Lift).

SQL state: 23514
Detail: Failing row contains (401, McCann Center, 166, 2026-03-19, 23:00:00, Group Lift).
```

cmpt308_lab3/postgres@PostgreSQL 18

Query History

```
1
2 INSERT INTO enrollment (student_ID, club_ID, join_Date, club_Role)
3 Values(222222, 224, '09-04-2022', 'President')
```

Data Output Messages Notifications

ERROR: insert or update on table "enrollment" violates foreign key constraint "enrollment_student_id_fkey"
Key (student_id)=(222222) is not present in table "student".

SQL state: 23503
Detail: Key (student_id)=(222222) is not present in table "student".