

NTWARI CEDRICK	ID:28228
NTAMBARA SHEMA CHRISPIN	ID:28280
IREMBERE OLIVIER	ID:28392

17/09/2025

PL\SQL GROUP ASSIGNMENT

WE CREATED A DATABASE CALLED STUDENT_MANAGEMNT_DB
WITH THREE TABLES , **STUDENTS,COURSES,ENROLLMENTS**

Explanation of Tables

Students Table

This table stores information about each student, including their unique ID, first name, last name, and email address. The email field is unique to prevent duplicate records.

Courses Table

This table contains details of courses offered, including a unique course ID, the course name, and the number of credits. A constraint ensures that the number of credits must be greater than zero.

Enrollments Table

This table records which students are enrolled in which courses, linking the students and courses tables through foreign keys. It also includes the date of enrollment. The **ON DELETE CASCADE** ensures that if a student or course is deleted, all related enrollment records are automatically removed to maintain data consistency.

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:

psql (17.5)
WARNING: Console code page (857) differs from Windows code page (1254)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

postgres=# create database student_management_db;
CREATE DATABASE
postgres=# /c student_management_db;
ERROR:  syntax error at or near "/"
LINE 1: /c student_management_db;
        ^

postgres=# \c student_management_db;
You are now connected to database "student_management_db" as user "postgres".
student_management_db=# -- Create a Students table
student_management_db=# -- we are going to create students table
student_management_db=# CREATE TABLE students (student_id SERIAL PRIMARY KEY,first_name VARCHAR(50) NOT NULL,last_name VARCHAR(50) NOT NULL,email VA
RCHAR(100) UNIQUE NOT NULL);
CREATE TABLE
student_management_db=# -- we are going to create a Courses table
student_management_db=#
student_management_db=# CREATE TABLE courses (course_id SERIAL PRIMARY KEY,course_name VARCHAR(100) NOT NULL,credits INT CHECK (credits > 0));
CREATE TABLE
student_management_db=# -- we are going to create an Enrollments table
student_management_db=#
student_management_db=# CREATE TABLE enrollments (enrollment_id SERIAL PRIMARY KEY,student_id INT REFERENCES students(student_id) ON DELETE CASCADE,
course_id INT REFERENCES courses(course_id) ON DELETE CASCADE,enrollment_date DATE NOT NULL);
CREATE TABLE
student_management_db=# -- we are now going to instert data into our tables
student_management_db=#
student_management_db=# INSERT INTO students (first_name, last_name, email) VALUES ('Alice', 'Mukamana', 'alice.mukamana@example.com'), ('Jean', 'Ni
yonzima', 'jean.niyonzima@example.com'), ('Claudine', 'Uwase', 'claudine.uwase@example.com'), ('Eric', 'Habimana', 'eric.habimana@example.com'), ('S
andrine', 'Iradingunda', 'sandrine.iradingunda@example.com'), ('Patrick', 'Mugisha', 'patrick.mugisha@example.com'), ('Diane', 'Nshimiyimana', 'diane.
nshimiyimana@example.com'), ('Kevin', 'Twagirayezu', 'kevin.twagirayezu@example.com'), ('Josiane', 'Mutoni', 'josiane.mutoni@example.com'), ('Emmanu
el', 'Rugamba', 'emmanuel.rugamba@example.com');
INSERT 0 10
```

```
student_management_db=# select * from students;
```

student_id	first_name	last_name	email
1	Alice	Mukamana	alice.mukamana@example.com
2	Jean	Niyonzima	jean.niyonzima@example.com
3	Claudine	Uwase	claudine.uwase@example.com
4	Eric	Habimana	eric.habimana@example.com
5	Sandrine	Iradukunda	sandrine.iradukunda@example.com
6	Patrick	Mugisha	patrick.mugisha@example.com
7	Diane	Nshimiyimana	diane.nshimiyimana@example.com
8	Kevin	Twagirayezu	kevin.twagirayezu@example.com
9	Josiane	Mutoni	josiane.mutoni@example.com
10	Emmanuel	Rugamba	emmanuel.rugamba@example.com

```
(10 rows)
```

```
student_management_db=# INSERT INTO courses (course_name, credits) VALUES ('Database Systems', 4), ('Web Development', 3), ('Computer Networks', 3), ('Operating Systems', 4), ('Software Engineering', 3), ('Cybersecurity Fundamentals', 2), ('Mobile App Development', 3), ('Cloud Computing', 4), ('Data Structures', 3), ('Artificial Intelligence', 4);
```

```
INSERT 0 10
```

```
student_management_db=# select * from courses;
```

course_id	course_name	credits
1	Database Systems	4
2	Web Development	3
3	Computer Networks	3
4	Operating Systems	4
5	Software Engineering	3
6	Cybersecurity Fundamentals	2
7	Mobile App Development	3
8	Cloud Computing	4
9	Data Structures	3
10	Artificial Intelligence	4

```
(10 rows)
```

```
student_management_db=# INSERT INTO enrollments (student_id, course_id, enrollment_date) VALUES (1, 1, '2025-09-01'), (2, 2, '2025-09-01'), (3, 3, '2025-09-02'), (4, 4, '2025-09-02'), (5, 5, '2025-09-03'), (6, 6, '2025-09-03'), (7, 7, '2025-09-04'), (8, 8, '2025-09-04'), (9, 9, '2025-09-05'), (10, 10, '2025-09-05');
```

```
INSERT 0 10
```

```
student_management_db=# INSERT INTO enrollments (student_id, course_id, enrollment_date) VALUES (1, 1, '2025-09-01'), (2, 2, '2025-09-01'), (3, 3, '2025-09-02'), (4, 4, '2025-09-02'), (5, 5, '2025-09-03'), (6, 6, '2025-09-03'), (7, 7, '2025-09-04'), (8, 8, '2025-09-04'), (9, 9, '2025-09-05'), (10, 10, '2025-09-05');
INSERT 0 10
```

```
student_management_db=# select * from enrollments;
```

enrollment_id	student_id	course_id	enrollment_date
1	1	1	2025-09-01
2	2	2	2025-09-01
3	3	3	2025-09-02
4	4	4	2025-09-02
5	5	5	2025-09-03
6	6	6	2025-09-03
7	7	7	2025-09-04
8	8	8	2025-09-04
9	9	9	2025-09-05
10	10	10	2025-09-05

```
(10 rows)
```

```
student_management_db=# --NOW WE ARE GOING TO PERFORM DIFFERENT TYPES OF JOINS
```

```
student_management_db=#
```

```
student_management_db=# --LET US USE INNER JOIN TO SEE STUDENTS ENROLLED IN COURSE
```

```
student_management_db=# SELECT s.first_name, s.last_name, c.course_name FROM students s INNER JOIN enrollments e ON s.student_id = e.student_id INNER JOIN courses c ON e.course_id = c.course_id;
```

first_name	last_name	course_name
Alice	Mukamana	Database Systems
Jean	Niyonzima	Web Development
Claudine	Uwase	Computer Networks
Eric	Habimana	Operating Systems
Sandrine	Iradukunda	Software Engineering
Patrick	Mugisha	Cybersecurity Fundamentals
Diane	Nshimiyimana	Mobile App Development
Kevin	Twagirayezu	Cloud Computing
Josiane	Mutoni	Data Structures
Emmanuel	Rugamba	Artificial Intelligence

```
(10 rows)
```

```
SQL Shell (psql)

student_management_db=# -- NOW LET'S USE LEFT JOIN TO SEE ALL STUDENTS AND THEIR COURSES EVEN IF NOT ENROLLED
student_management_db=#
student_management_db=# SELECT s.first_name, s.last_name, c.course_name FROM students s LEFT JOIN enrollments e ON s.student_id = e.student_id LEFT JOIN courses c ON
e.course_id = c.course_id;
 first_name | last_name |      course_name
-----+-----+-----
 Alice      | Mukamana | Database Systems
 Jean       | Niyonzima | Web Development
 Claudine   | Uwase    | Computer Networks
 Eric       | Habimana | Operating Systems
 Sandrine   | Iradukunda | Software Engineering
 Patrick    | Mugisha  | Cybersecurity Fundamentals
 Diane     | Nshimiyimana | Mobile App Development
 Kevin      | Twagirayezu | Cloud Computing
 Josiane    | Mutoni   | Data Structures
 Emmanuel   | Rugamba  | Artificial Intelligence
(10 rows)

student_management_db=# -- NOW LET'S USE RIGHT JOIN TO SEE ALL STUDENTS AND THEIR COURSES EVEN IF NOT ENROLLED
student_management_db=#
student_management_db=# SELECT s.first_name, s.last_name, c.course_name FROM students s RIGHT JOIN enrollments e ON s.student_id = e.student_id RIGHT JOIN courses c ON
N e.course_id = c.course_id;
 first_name | last_name |      course_name
-----+-----+-----
 Alice      | Mukamana | Database Systems
 Jean       | Niyonzima | Web Development
 Claudine   | Uwase    | Computer Networks
 Eric       | Habimana | Operating Systems
 Sandrine   | Iradukunda | Software Engineering
 Patrick    | Mugisha  | Cybersecurity Fundamentals
 Diane     | Nshimiyimana | Mobile App Development
 Kevin      | Twagirayezu | Cloud Computing
 Josiane    | Mutoni   | Data Structures
 Emmanuel   | Rugamba  | Artificial Intelligence
(10 rows)
```

```
SQL Shell (psql)

student_management_db=# -- NOW LET'S USE FULL JOIN TO SEE ALL STUDENTS AND THEIR COURSES EVEN IF NOT ENROLLED
student_management_db=#
student_management_db=# SELECT s.first_name, s.last_name, c.course_name FROM students s FULL JOIN enrollments e ON s.student_id = e.student_id FULL JOIN courses c ON e.course_id = c.course_id;
 first_name | last_name | course_name
-----
 Alice      | Mukamana | Database Systems
 Jean       | Niyonzima | Web Development
 Claudine   | Uwase    | Computer Networks
 Eric       | Habimana | Operating Systems
 Sandrine   | Iradukunda | Software Engineering
 Patrick    | Mugisha  | Cybersecurity Fundamentals
 Diane      | Nshimiyimana | Mobile App Development
 Kevin      | Twagirayezu | Cloud Computing
 Josiane    | Mutoni   | Data Structures
 Emmanuel   | Rugamba  | Artificial Intelligence
(10 rows)

student_management_db=# -- NOW WE ARE GOING TO CREATE INDEX TO OPTIMIZE QUERY PERFORMANCE
student_management_db=#
student_management_db=#
student_management_db=# CREATE INDEX idx_student_id ON enrollments(student_id);
CREATE INDEX
student_management_db=# CREATE INDEX idx_course_id ON enrollments(course_id);
CREATE INDEX
student_management_db=# --NOW WE ARE GOING TO CREATE VIEW TO SIMPLIFY DATA ACCESS
student_management_db=#
student_management_db=# CREATE VIEW student_courses AS SELECT s.first_name, s.last_name, c.course_name, e.enrollment_date FROM students s JOIN enrollments e ON s.student_id = e.student_id JOIN courses c ON e.course_id = c.course_id;
CREATE VIEW
student_management_db=# -- GENERATING A SIMPLE REPORT LET'S REFER TO THE VIEW WE CREATED
student_management_db=#
student_management_db=# SELECT * FROM student_courses ORDER BY enrollment_date;
 first_name | last_name | course_name | enrollment_date
-----
 Alice      | Mukamana | Database Systems | 2025-09-01
 Jean       | Niyonzima | Web Development | 2025-09-01
 Claudine   | Uwase    | Computer Networks | 2025-09-02
 Eric       | Habimana | Operating Systems | 2025-09-02
 Sandrine   | Iradukunda | Software Engineering | 2025-09-03
 Patrick    | Mugisha  | Cybersecurity Fundamentals | 2025-09-03
 Diane      | Nshimiyimana | Mobile App Development | 2025-09-04
 Kevin      | Twagirayezu | Cloud Computing | 2025-09-04
 Josiane    | Mutoni   | Data Structures | 2025-09-05
 Emmanuel   | Rugamba  | Artificial Intelligence | 2025-09-05
(10 rows)
```

```

student_management_db=# \dt
List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | courses   | table | postgres
public | enrollments | table | postgres
public | students  | table | postgres
(3 rows)

student_management_db=# --WE CAN ALSO CREATE A SIMPLE VIEW BY COUNTING ENROLLMENTS PER COURSE
student_management_db=#
student_management_db=# SELECT c.course_name, COUNT(e.student_id) AS total_students FROM courses c LEFT JOIN enrollments e ON c.course_id = e.course_id GROUP BY c.course_name;
   course_name   | total_students
-----+-----
Database Systems |             1
Cloud Computing  |             1
Mobile App Development |             1
Computer Networks |             1
Artificial Intelligence |             1
Data Structures  |             1
Web Development  |             1
Software Engineering |             1
Operating Systems |             1
Cybersecurity Fundamentals |             1
(10 rows)

student_management_db=# |

```

WE EVEN ADDED ER DIAGRAM TO SHOW RELATIONSHIPS AND SIMPLE VIEW WE CREATED WHICH WE CALLED STUDENT_COURSES



