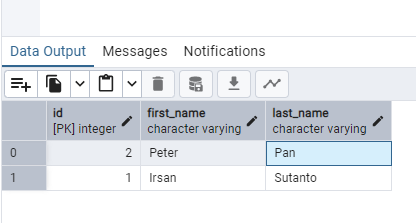
**Lab 12 – Postgresql – pgAdmin 4**

Student’s name : Irsan Sutanto

**Tuesday, April 29, 2025 (introduction)**

Class example:



Lab 12A exercise: Manually create a database, table, and columns

* Database: demoDB
* Tables: courses, tuition
* Courses table will have 3 columns: course\_code (2 characters), semester (character varying), schedule (text)
  + - Create 2 courses data entries.

-- create courses table

CREATE TABLE courses(

id SERIAL PRIMARY KEY,

course\_code CHAR(2),

semester VARCHAR(20),

schedule TEXT

);

-- create tuition table

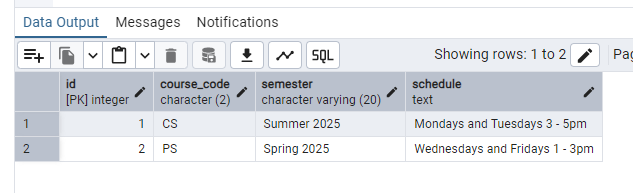
INSERT INTO courses (id, course\_code, semester, schedule)

VALUES

(1, 'CS', 'Summer 2025', 'Mondays and Tuesdays 3 - 5pm'),

(2, 'PS', 'Spring 2025', 'Wednesdays and Fridays 1 - 3pm');

SELECT \* FROM courses



* Tuition table will have 3 columns: id(primary key), pt\_credit\_per(numerical), ft\_price(numerical)
* Create 1 data entry for local students
* Create 1 data entry for international students

-- create tuition table

CREATE TABLE tuition(

id SERIAL PRIMARY KEY,

pt\_credit\_price NUMERIC,

ft\_price NUMERIC,

student\_type VARCHAR(30));

-- insert data for local student

INSERT INTO tuition(id, pt\_credit\_price, ft\_price, student\_type)

VALUES

(1, 500.00, 6500.00, 'local student');

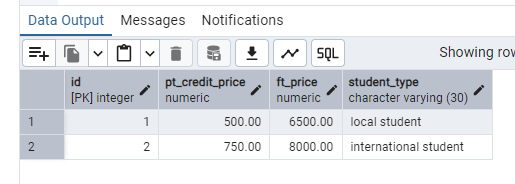
-- insert data for international student

INSERT INTO tuition(id, pt\_credit\_price, ft\_price, student\_type)

VALUES

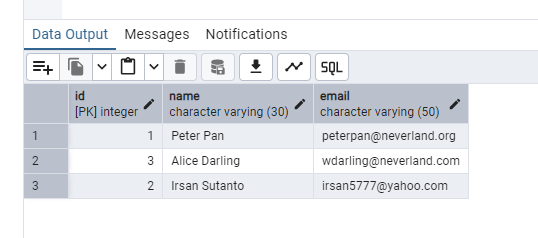
(2, 750.00, 8000.00, 'international student');

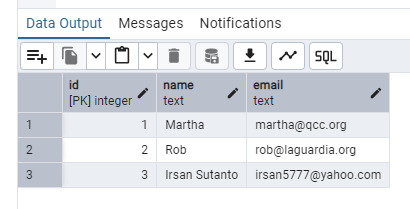
SELECT \* FROM tuition;



**Thursday, May 1, 2025 (CRUD)**

**Class notes**  
**Class example:**





-- drop the 'users' table if exists

DROP TABLE IF EXISTS users;

-- 1. CREATE A TABLE

CREATE TABLE users(

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

email TEXT UNIQUE NOT NULL);

-- 2. INSERT DATA

INSERT INTO users(name, email)

VALUES

('Martha', 'martha@qcc.org'),

('Rob', 'rob@qcc.org'),

('Charlie', 'charlie@qcc.org');

-- READ

SELECT \* FROM users

-- 4. UPDATE, update email for 'Rob'

UPDATE users

SET email = 'rob@laguardia.org'

-- 5. DELETE

DELETE FROM users

WHERE name = 'Charlie';

-- CREATE NEW USER WITH YOUR NAME

INSERT INTO users (name, email)

VALUES

('Irsan Sutanto', 'irsan5777@yahoo.com');

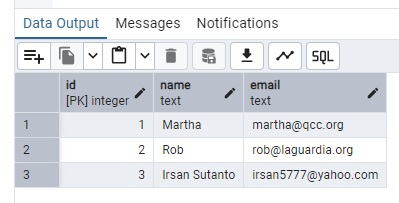
UPDATE users

SET id=3

WHERE id=4;

-- READ

SELECT \* FROM users



**NOTES :**

**CLASS EXAMPLES:**

**Friday, May 2, 2025**

-- create artist

CREATE TABLE artist(

id SERIAL PRIMARY KEY,

name VARCHAR(128)

);

-- genre table

CREATE TABLE genre(

id SERIAL PRIMARY KEY,

name VARCHAR(128)

);

-- create album table

CREATE TABLE album(

id SERIAL PRIMARY KEY,

title VARCHAR(128),

artist\_id INTEGER REFERENCES artist(id) ON DELETE CASCADE

);

-- create track table

CREATE TABLE track(

id SERIAL PRIMARY KEY,

title VARCHAR(128),

len INTEGER,

rating INTEGER,

count\_track INTEGER,

album\_id INTEGER REFERENCES album(id) ON DELETE CASCADE,

genre\_id INTEGER REFERENCES genre(id) ON DELETE CASCADE

);

-- insert data

-- GENRE

INSERT INTO genre(name)

VALUES

('Rock'),

('Metal');

-- artist table

INSERT INTO artist(id, name)

VALUES

(1,'Led Zeppelin'),

(2,'AC/DC');

-- album title

INSERT INTO album(id, title, artist\_id)

VALUES

(1, 'Who made who',2),

(2, 'IV',1);

-- track table

INSERT INTO track(title, rating, len, count\_track, album\_id, genre\_id)

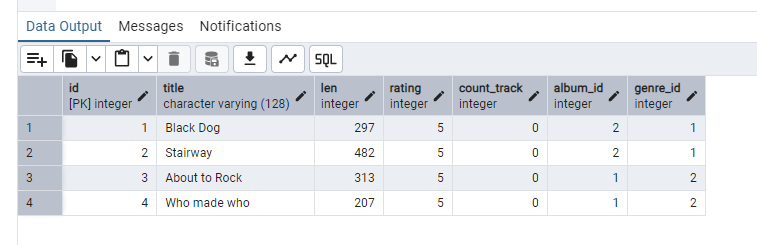
VALUES

('Black Dog', 5, 297, 0, 2, 1),

('Stairway', 5, 482, 0, 2, 1),

('About to Rock', 5, 313, 0, 1, 2),

('Who made who', 5, 207, 0, 1,2);



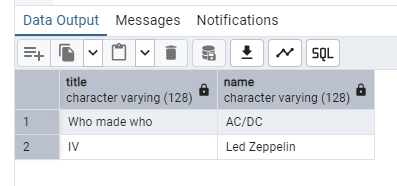
-- display the name of the album with the artist name

SELECT album.title, artist.name

FROM album

JOIN artist

ON album.artist\_id = artist.id;



-- exercise

INSERT INTO artist(id, name)

VALUES

(3, 'Irsan Sutanto');

INSERT INTO album(id, title, artist\_id)

VALUES

(3, 'Microcredentials', 3);

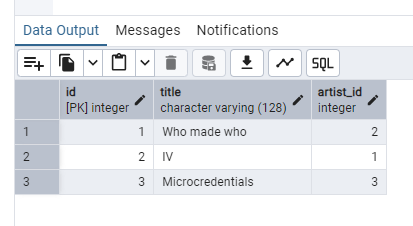
-- display the name of the album with the artist name

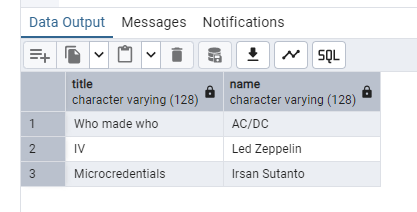
SELECT album.title, artist.name

FROM album

JOIN artist

ON album.artist\_id = artist.id;





-- display the name of the album with the artist name

SELECT album.title, artist.name

FROM album

JOIN artist

ON album.artist\_id = artist.id;

-- display the title of the track with the genre's name

SELECT track.title, genre.name

FROM track

JOIN genre

ON track.genre\_id = genre.id;

