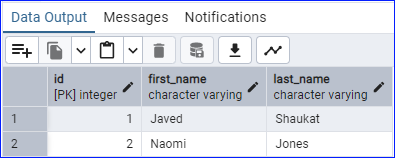
***Javed Shaukat***

**Lab 12A – Postgresql | Tuesday, April 29, 2025 – (Introduction)**

Class Example:



**Lab Exercise 1 | Tuesday, April 29, 2025 – (Exercise)**

Manually create a database, table, and columns.

* Database name: demoDB
* Tables: courses, tuition

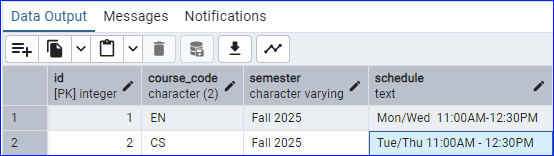
Courses table will have four columns: id (primary key), course\_code (2 characters), semester (character varying), schedule (text).

* Create two courses’ data entries.

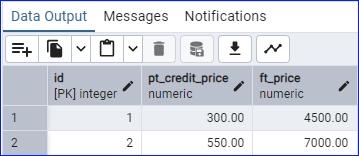
Tuition table will have three columns: id (primary key), pt\_credit\_price (numerical), ft\_price(numerical).

* Create one data entry for local students.
* Create one data entry for international students.

Courses Table and data entry:

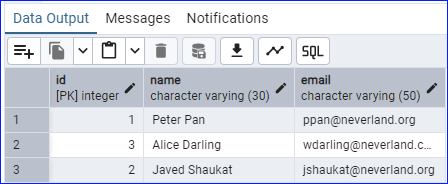


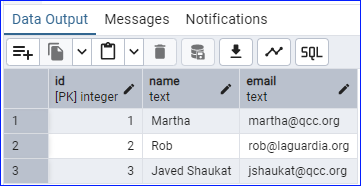
Tuition Table and data entry:



**Thursday, May 01, 2025 (CRUD)**

Class Example:

****



**12B Lab Exercise:**

-- 1. CREATE THE TABLE:

CREATE TABLE books (

id SERIAL PRIMARY KEY,

title VARCHAR(100) NOT NULL,

author VARCHAR(100) NOT NULL,

year\_published INT

);



-- 2. INSERT RECORDS:

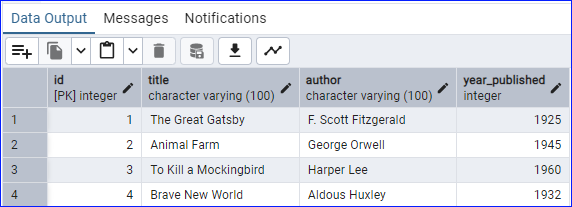
INSERT INTO books (title, author, year\_published) VALUES

('The Great Gatsby', 'F. Scott Fitzgerald', 1925),

('Animal Farm', 'George Orwell', 1945),

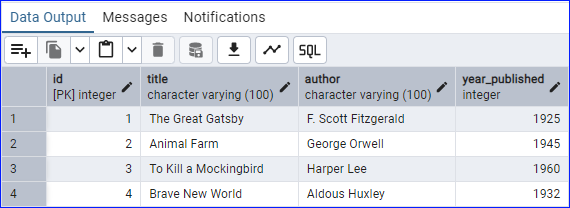
('To Kill a Mockingbird', 'Harper Lee', 1960),

('Brave New World', 'Aldous Huxley', 1932);



-- 3. READ DATA

SELECT \* FROM books;



-- 3B SELECT BOOKS BY GEORGE ORWELL

SELECT \* FROM books WHERE author = 'George Orwell';

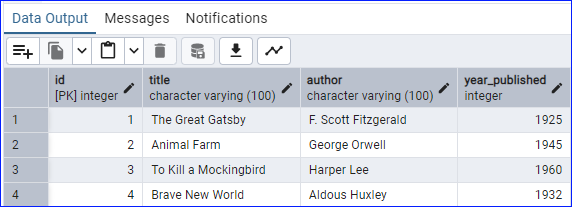


-- 4. UPDATE DATA, UPDATE THE PUBLICATION YEAR OF BRAVE NEW WORLD:

UPDATE books

SET year\_published = 1931

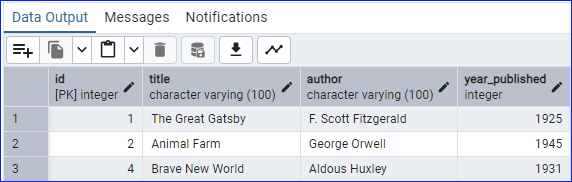
WHERE title = 'Brave New World';



-- 5. DELETE DATA, DELETE TO KILL A MOCKING BIRD:

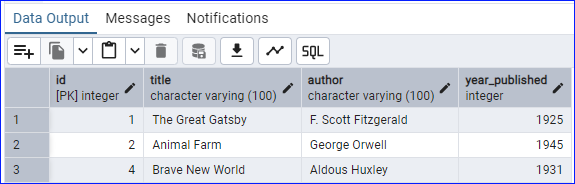
DELETE FROM books

WHERE title = 'To Kill a Mockingbird';



-- 6. VIEW FINAL TABLE:

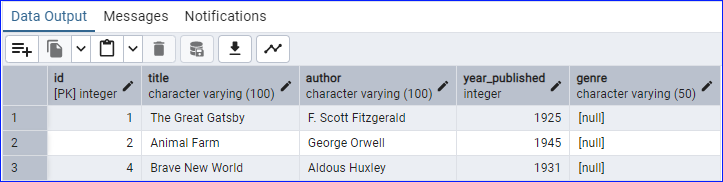
SELECT \* FROM books;



-- BONUS CHALLENG, ADD A NEW COLUMN 'GENRE' TO THE BOOKS TABLE:

ALTER TABLE books

ADD COLUMN genre VARCHAR(50);

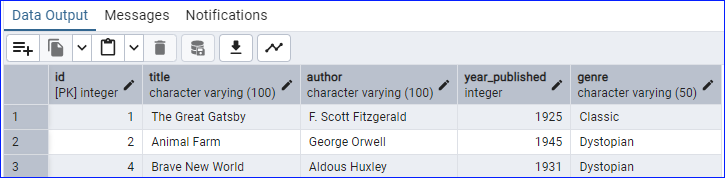


-- UPDATE THE GENRE FOR EACH BOOK:

UPDATE books SET genre = 'Classic' WHERE title = 'The Great Gatsby';

UPDATE books SET genre = 'Dystopian' WHERE title = 'Animal Farm';

UPDATE books SET genre = 'Dystopian' WHERE title = 'Brave New World';

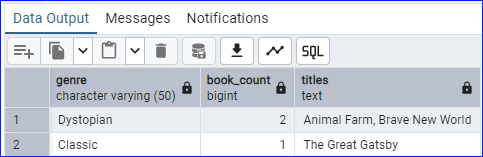


-- SELECT ALL BOOKS GROUPED BY GENERE:

SELECT genre, COUNT(\*) as book\_count, STRING\_AGG(title, ', ') as titles

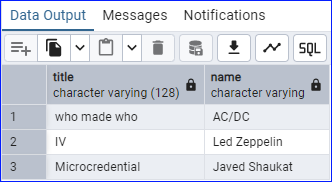
FROM books

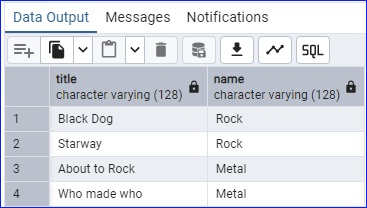
GROUP BY genre;



**Friday, May 02, 2025**

Class Example:





**LAB 12C – LAB EXERCISE**

Relational database

Activity description: Create a database for a university to manage students, courses, and

enrollments, and practice performing some SQL queries.

Assignment:

• Create three tables:

o ‘students’ table with columns:

▪ id, serial and primary key

▪ name, variant character maximum of 100 characters

▪ major, variant character maximum of 100 characters

o ‘courses’ table with columns:

▪ id, serial and primary key

▪ title, variant character maximum of 100 characters.

▪ department, variant character maximum of 100 characters.

o ‘enrollments’ table with columns:

▪ id, serial and primary key

▪ grade, 2 characters

▪ Foreign key named ‘student\_id’ to student’s id

▪ foreign key named ‘courses\_id' to courses’ id

Create a database:

Right-click on "Databases" and select "Create" > "Database"

Name it "university" and click "Save"

-- Students table

CREATE TABLE students (

id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

major VARCHAR(100) NOT NULL

);

-- Courses table

CREATE TABLE courses (

id SERIAL PRIMARY KEY,

title VARCHAR(100) NOT NULL,

department VARCHAR(100) NOT NULL

);

-- Enrollments table

CREATE TABLE enrollments (

id SERIAL PRIMARY KEY,

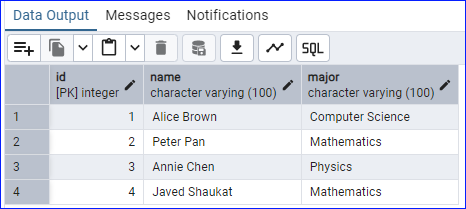
grade CHAR(2),

student\_id INT REFERENCES students(id),

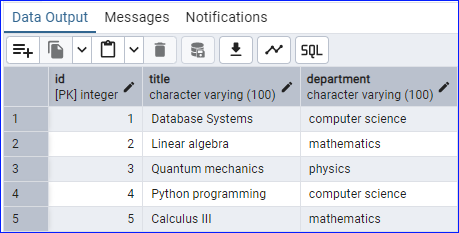
course\_id INT REFERENCES courses(id)

);

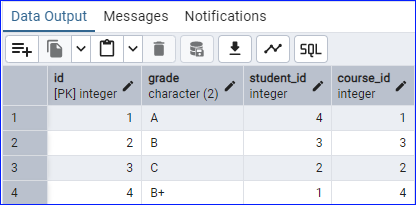
SELECT \* FROM students;



SELECT \* FROM courses;



SELECT \* FROM enrollments;



-- Insert students

INSERT INTO students (name, major) VALUES

('Alice Brown', 'Computer Science'),

('Peter Pan', 'Mathematics'),

('Annie Chen', 'Physics'),

('Javed Shaukat', 'Mathematics');

-- Insert courses

INSERT INTO courses (title, department) VALUES

('Database Systems', 'computer science'),

('Linear algebra', 'mathematics'),

('Quantum mechanics', 'physics'),

('Python programming', 'computer science'),

('Calculus III', 'mathematics');

-- Insert enrollments

-- Note: You'll need to use the actual IDs from your tables

-- This assumes the IDs were created in order as inserted

INSERT INTO enrollments (grade, student\_id, course\_id) VALUES

('A', 4, 1), -- Javed in Database Systems

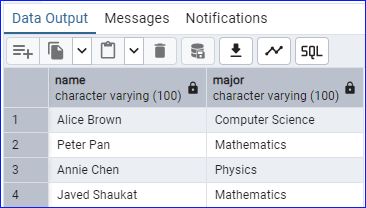
('B', 3, 3), -- Annie in Quantum mechanics

('C', 2, 2), -- Peter in Linear algebra

('B+', 1, 4); -- Alice in Python programming

-- Read all students and their majors

SELECT name, major FROM students;



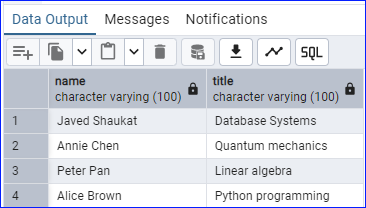
-- Read the title of courses each student is enrolled in

SELECT s.name, c.title

FROM students s

JOIN enrollments e ON s.id = e.student\_id

JOIN courses c ON c.id = e.course\_id;



-- Read all students with grades and courses

SELECT s.name, c.title, e.grade

FROM students s

JOIN enrollments e ON s.id = e.student\_id

JOIN courses c ON c.id = e.course\_id;

