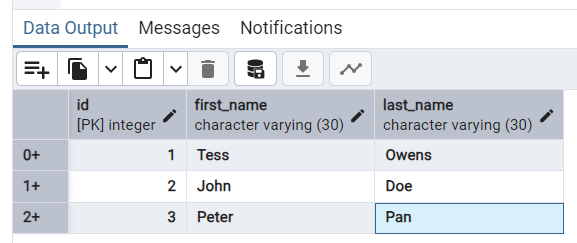
Lab 12 : PostgreSQL

Tess Owens

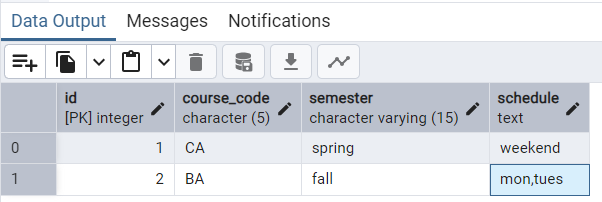
April 29, 2025

Class Example:



Lab Exercise: Manually create a database, table and columns.

Courses Table:



Tuition Table:

A screenshot of a computer

AI-generated content may be incorrect.

Thursday, May 1, 2025(CRUD)

Class Example:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Lab exercise –**CRUD 12B**

**1. Create the Table**

Create a table named books with the following structure:

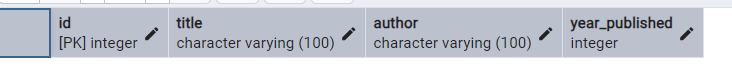
• id – SERIAL PRIMARY KEY

• title – VARCHAR(100) NOT NULL

• author – VARCHAR(100) NOT NULL

• year\_published – INT

|  |
| --- |
| -- 1. Create a table named books with the following structure:  CREATE TABLE books(  id SERIAL PRIMARY KEY,  title VERCHAR(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'); |

A screenshot of a computer

AI-generated content may be incorrect.

3. Read Data

Write SQL queries to:

• Select all books.

|  |
| --- |
| SELECT title FROM books; |

**A screenshot of a computer

AI-generated content may be incorrect.**

• Select books where the author is 'George Orwell'.

|  |
| --- |
| SELECT title  FROM books  WHERE author = 'George Orwell'; |

A screenshot of a computer

AI-generated content may be incorrect.

4. Update Data

Update the year\_published of Brave New World to 1931.

|  |
| --- |
| UPDATE books  SET year\_published = 1931  WHERE title = 'Brave New World' |

5. Delete Data

Delete the record for To Kill a Mockingbird.

|  |
| --- |
| --5. Delete Data  --Delete the record for To Kill a Mockingbird.  DELETE FROM books WHERE title ='To Kill a Mockingbird'; |

6. View Final Table

Select all records from the books table to verify your changes.

|  |
| --- |
| SELECT \*  FROM books; |
|  |

A screenshot of a computer

AI-generated content may be incorrect.

Bonus Challenge (Optional)

• Add a new column genre to the books table.

|  |
| --- |
| ALTER TABLE books ADD COLUMN genre TEXT; |

• Update the genre for each book.

|  |
| --- |
| UPDATE books SET genre = 'Tragedy' WHERE title ='The Great Gasby';  UPDATE books SET genre = 'Fable' WHERE title ='Animal Farm';  UPDATE books SET genre = 'Science Fiction' WHERE title = 'Brave'; |

• Select all books grouped by genre.

|  |
| --- |
| SELECT genre, COUNT(\*) AS total\_books  FROM books  GROUP BY genre; |

A screenshot of a computer

AI-generated content may be incorrect.

**Friday , May 2, 2025**

Class Example:

|  |
| --- |
| -- EXERCISE  SELECT album.title, artist.name  FROM album  JOIN artist  ON album.artist\_id = artist.id; |

A screenshot of a computer

AI-generated content may be incorrect.

|  |
| --- |
| -- 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; |

A screenshot of a computer

AI-generated content may be incorrect.

LAB EXERCISE 12C:

Relational database

Create three tables:

‘students’ table with columns:

|  |
| --- |
| CREATE TABLE lab\_ students(  id SERIAL PRIMARY KEY,  name VARCHAR(100),  major VARCHAR(100)  ); |

‘courses’ table with columns:

|  |
| --- |
| CREATE TABLE course(  id SERIAL PRIMARY KEY,  title VARCHAR(100),  department VARCHAR(100)  ); |

‘enrollments’ table with columns:

|  |
| --- |
| CREATE TABLE enrollments(  id SERIAL PRIMARY KEY,  grade VARCHAR(2),  student\_id INTEGER REFERENCE students(id),  course\_id INTERGER REFERENCE courses(id)  ); |

Data entry for each table:

|  |
| --- |
| INSERT INTO lab\_students (name, major)  VALUES  ('Alice Brown', 'Computer Science'),  ('Peter Pan', 'Mathematics'),  ('Annie Chen', 'Physics'),  ('Tess Owens', 'Mathematics'); |

A screenshot of a computer

AI-generated content may be incorrect.

ADD Five Courses:

|  |
| --- |
| INSERT INTO course (title, department)  VALUES  ('Database Systems', 'Computer Science'),  ('Linear algebra', 'Mathematics'),  ('Quantum mechanics', 'Physics'),  ('Python programming', 'computer science'),  ('Calculus III', 'mathematics'); |

A screenshot of a computer

AI-generated content may be incorrect.

Enrollments:

▪ Grade of ‘A’, for ‘Tess’

▪ Grade of ‘B’ for Annie

▪ Grade of ‘C’ for Peter

▪ Grade of ‘B+’ for Alice

|  |
| --- |
| INTO enrollments (grade, student\_id, courses\_id) VALUES ('A', 4, 5);  INSERT INTO enrollments (grade, student\_id, courses\_id) VALUES ('B', 3, 3);  INSERT INTO enrollments (grade, student\_id, courses\_id) VALUES ('C', 2, 2);  INSERT INTO enrollments (grade, student\_id, courses\_id) VALUES ('B+', 1, 1); |

A screenshot of a computer

AI-generated content may be incorrect.

*read all students and their major*

|  |
| --- |
| SELECT name, major  FROM lab\_students; |

A screenshot of a computer

AI-generated content may be incorrect.

*read the titles of courses each student is enrolled in.*

|  |
| --- |
| SELECT lab\_students.name, course.title  FROM enrollments  JOIN lab\_students ON enrollments.student\_id = lab\_students.id  JOIN course ON enrollments.courses\_id = course.id; |

A screenshot of a computer

AI-generated content may be incorrect.

*read all students with grade and courses.*

|  |
| --- |
| *SELECT lab\_students.name, enrollments.grade, course.title*  *FROM enrollments*  *JOIN lab\_students ON enrollments.student\_id = lab\_students.id*  *JOIN course ON enrollments.courses\_id = course.id;* |

*A screenshot of a computer

AI-generated content may be incorrect.*