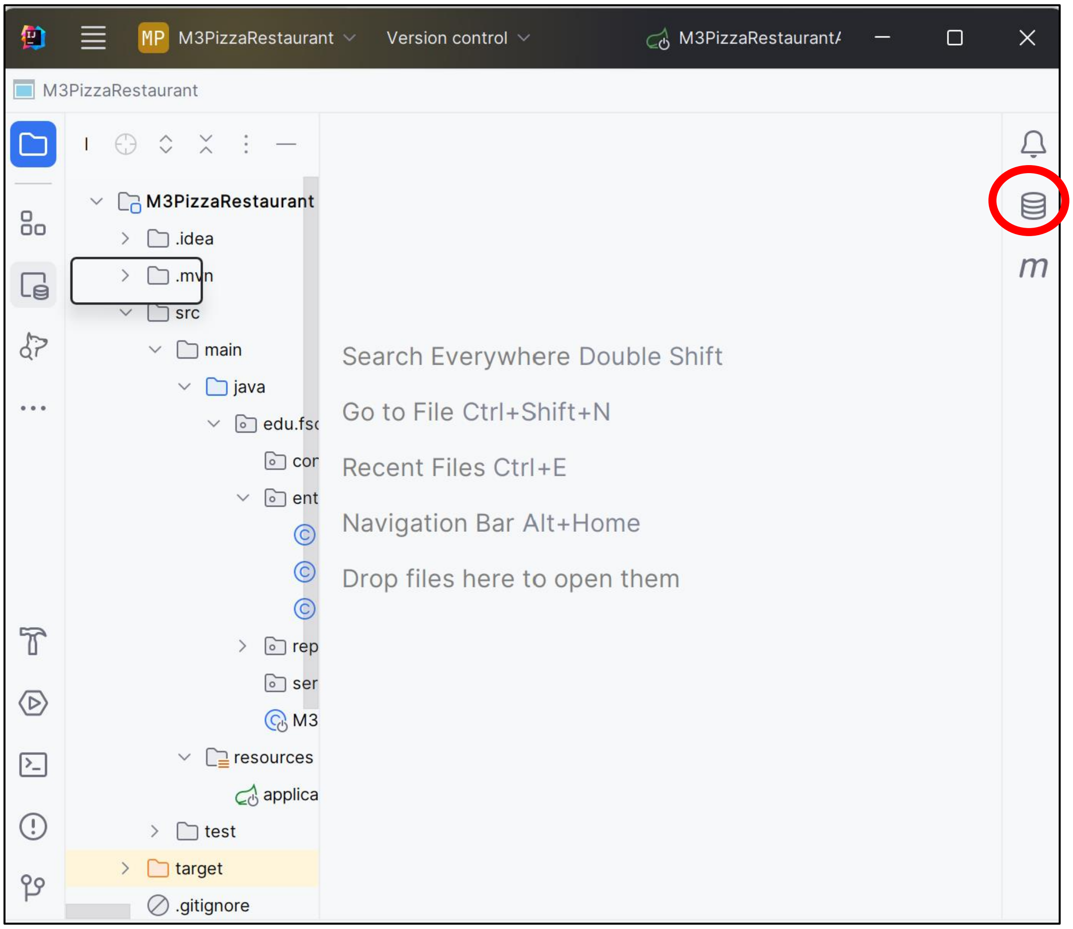
CEN3024C Module 3 Project

For this project we will experiment with using Spring Data JPA and the Database tool in IntelliJ. This tool (highlighted in red, below) allows you to manipulate a database using a configured connection without having to use other tools. Tables, columns, and data can be added, modified, and deleted.



Clone the GitHub Classroom assignment repo and use the following commands to create the postgresql database:

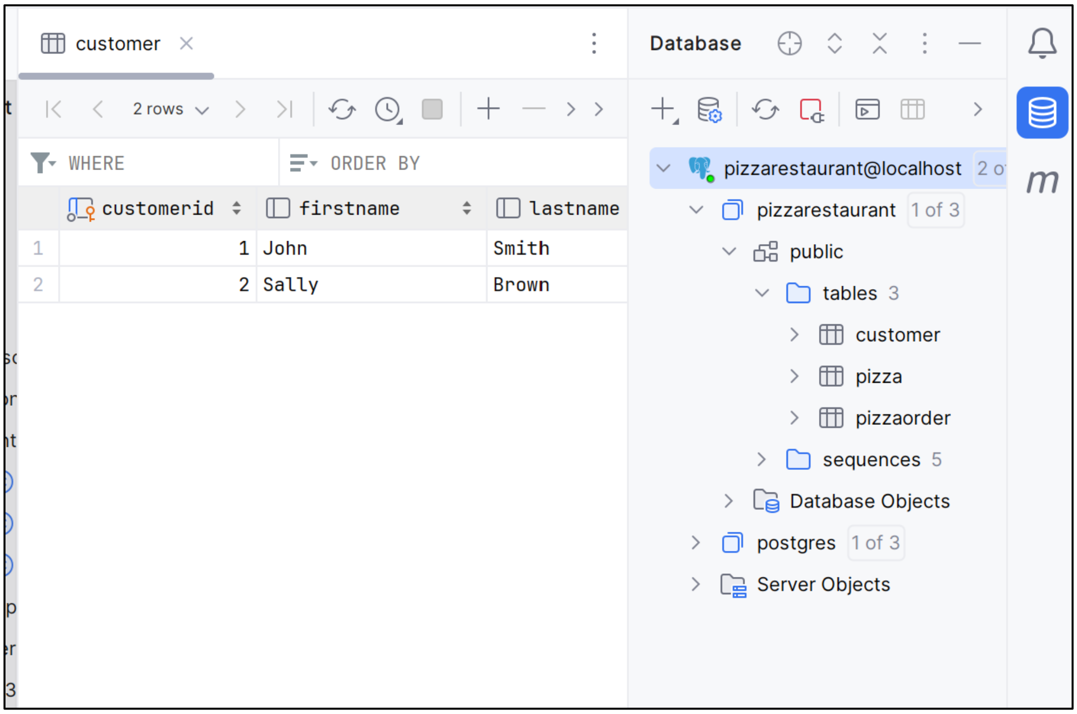
```
C:\> c:\"program files"\postgresql\16\bin\psql -U postgres
...
postgres=# create database pizzarestaurant;
CREATE DATABASE
postgres=#\q

C:\> c:\"program files"\postgresql\16\bin\psql -U postgres
pizzarestaurant <db\m3pizzarestaurant.sql
```

The last command will restore the database (including the data) from the pizzadb.psql file.

Note that this file was created using the pg_dump utility, which is a quick way to do a full backup of your database:

```
C:\> c:\"program files"\postgresql\16\bin\pg_dump -U postgres
pizzarestaurant >db\m3pizzarestaurant.sql
```

Start by creating the connection to the data source using the "+" button in the Database tool. Once you have done that you should be able to expand the database to view the schema and contents. Here's an example of using the tool to display the contents of the customer table (double-clicking on "customer" after expanding pizzarestaurant/public/tables:



	customerid	firstname	lastname
1	1	John	Smith
2	2	Sally	Brown

Examine the source code for the project and note the use of the @OneToMany annotations in the Customer and Pizza entity classes. This establishes a many-to-many relationship between the two classes which is implemented as a join table via the PizzaOrder class. Using the join table is necessary if additional columns are needed; otherwise, a simple @ManyToMany annotation could be used without the join table.

There are corresponding JpaRepository-based interfaces in the repository package.

The program executes queries via the "testDatabaseRepositories" method in the main application class via a CommandLineRunner, an interface in Spring Boot used to execute code after the Spring Boot application has started.

For this assignment you will need to add the following columns to the database using the IntelliJ Database tool:

- Add an orderdate column to the pizzaorder table.
- Add address, phone, city, state, postalcode columns to the customer table.

- Add a crusttype column to the pizza table.
- Add the associated fields to the entity classes with the necessary column mapping values.
- Add the necessary data values to the new columns for the existing records, then add several more records to the tables with associated data (some possible crusttype values: thick, thin, stuffed).

Submit your project to the GitHub Classroom repo and add 3-4 screenshots showing your use of the Database tool in IntelliJ to perform the tasks listed above.