Java module 4

Exercises Day 2

1.1 - Create	Create database
Instructions	Using pgAdmin, create a new PostgreSQL database named **exercise_db**

1.2 - Create	Create table
Instructions	Using pgAdmin, in the **exercise_db** database, create a table named **products** with the following columns: • **product_id** as an integer (primary key) • **product_name** as a variable character string • **price** as a numeric type • **available** as a boolean • **launch_date** as a date.

1.3 - Create	Create table
Instructions	Using pgAdmin, create another table called **orders** that includes a foreign key reference to **products**. Each order should include: • an order ID • the reference to the product that was ordered • the date of the order • the quantity of the order

1.4 - Insert	Insert data
Instructions	Using pgAdmin, insert sample data into products and orders tables.

1.5 - Update	Update data
Instructions	Using pgAdmin, increase the **price** of all products by 10%.

1.6 - Rename	Update the table structure
Instructions	Using pgAdmin, rename the **orders** table to **customer_orders**. Rename the column **available** to **is_available** in the **products** table.

1.8	Add a new column `description` to the `products` table.
1.9	Change the data type of the `price` column in the `products` table from `NUMERIC` to `FLOAT`.

1.10	Create a table named `employees` with a unique constraint on the `email` column.
1.11	Create a table named `departments` with a `CHECK` constraint that ensures the `budget` is greater than zero.
1.12	Add a `status` column to the `employees` table that defaults to 'Active'.
1.13	Increase the budget by 10% for departments with a budget less than $5000.
1.14	Delete employees who are marked as 'Inactive'.
1.15	Create a `project_assignments` table where both `project_id` and `employee_id` together form the primary key.
1.16	Modify the `employees` table to ensure that the `name` column cannot contain null values.
1.17	Rename the database `exercise_db` to `new_exercise_db`.

2.1 - Insert	Insert data
Instructions	Write a Java program to insert a new actor into the actor table.
Expected output	After executing the program, verify that the new row was added to the actor table using pgAdmin.

2.2 - Update	Update data
Instructions	Write a Java program to update an actor's name in the actor table.
Expected output	After executing the program, verify that the row was updated in the actor table using pgAdmin.

2.3 - Delete	Delete data
Instructions	Write a Java program to delete an actor in the actor table.
Expected output	After executing the program, verify that the row was deleted from the actor table using pgAdmin.