**ASSIGNMENT 3 (JOIN)**

**Assignment: Join Operations**

Here's an assignment that covers join operations in PostgreSQL, including the necessary CREATE TABLE and INSERT commands, along with the solutions:

Assume you're managing data for a company that sells products. You have two tables, **products** and **orders**, with the following structures:

1. **products**

|  |  |  |  |
| --- | --- | --- | --- |
| Col Name | product\_id | product\_name | category\_id |
| Data Type | INTEGER | VARCHAR(100) | INTEGER |
| Constraint | P.K. | Not Null |  |
| Record 1 | 10 | Product A | 1 |
| Record 2 | 20 | Product B | 2 |
| Record 3 | 30 | Product C | 3 |

1. **orders**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Col Name | order\_id | product\_id | quantity | order\_date |
| Data Type | INTEGER | INTEGER | INTEGER | DATE |
| Constraint | P.K. | REFERENCES products(product\_id) |  |  |
| Record 1 | 101 | 10 | 10 | '2023-01-15' |
| Record 2 | 102 | 20 | 5 | '2023-01-20' |
| Record 3 | 103 | 10 | 8 | '2023-01-20' |

Tasks:

1. Create the products and orders tables with the given structures.
2. Insert sample data into both tables.
3. Write a simple English statement describing an INNER JOIN operation between the products and orders tables.
4. Write an SQL query using INNER JOIN to retrieve the names of products and the corresponding order quantities.
5. Write an SQL query using LEFT JOIN to retrieve all product names and their corresponding order quantities, if any.
6. Write an SQL query using RIGHT JOIN to retrieve all order quantities along with their corresponding product names, if available.
7. Write an SQL query using FULL OUTER JOIN to retrieve all product names and order quantities, including cases where there are no matching orders or products.

Solutions:

Creating the tables:

CREATE TABLE products (

product\_id INTEGER PRIMARY KEY,

product\_name VARCHAR(100) NOT NULL,

category\_id INTEGER

);

CREATE TABLE orders (

order\_id SERIAL PRIMARY KEY,

product\_id INTEGER REFERENCES products(product\_id),

quantity INTEGER,

order\_date DATE

);

**Inserting sample data (you can adjust the data as needed):**

-- Insert sample data into the products table

INSERT INTO products VALUES

(10, 'Product A', 1),

(20, 'Product B', 2),

(30, 'Product C', 1);

-- Insert sample data into the orders table

INSERT INTO orders VALUES

(101, 10, 10, '2023-01-15'),

(102, 20, 5, '2023-01-20'),

(103, 10, 8, '2023-01-25');

Simple English statement describing an INNER JOIN:

An INNER JOIN combines rows from both the products and orders tables based on a common column (product\_id) to retrieve product names and their corresponding order quantities when there is a match.

**SQL query using INNER JOIN:**

SELECT p.product\_name, o.quantity

FROM products p

INNER JOIN orders o ON p.product\_id = o.product\_id;

**SQL query using LEFT JOIN:**

SELECT p.product\_name, o.quantity

FROM products p

LEFT JOIN orders o ON p.product\_id = o.product\_id;

SQL query using RIGHT JOIN:

SELECT p.product\_name, o.quantity

FROM products p

RIGHT JOIN orders o ON p.product\_id = o.product\_id;

SQL query using FULL OUTER JOIN:

SELECT p.product\_name, o.quantity

FROM products p

FULL OUTER JOIN orders o ON p.product\_id = o.product\_id;

You can execute these SQL queries in a PostgreSQL environment to understand how each type of join works and to verify the results. Adjust the data or queries as needed for your specific requirements.