 📚 **Lesson** | Exploring Other Types of SQL Joins

— Types of JOINs

PROMPT: Let’s practice using the different types of joins in SQL. For this practice, we’ll be using the **initech** schema in our queries.

1. Write a query that returns the customer name from the **initech\_customers** table and the order number from the **initech\_orders** table. Ensure that your output keeps any customer who has not placed an order.

|  |
| --- |
| select  customername,  ordernumber  from initech\_customers ic  left join initech\_orders io  on io.customernumber = ic.customernumber |

1. Modify your query so that for each customer, your query counts how many orders they have placed. How many customers in our database have NOT placed any orders? Note: There are several ways to do this, but the simplest way would be to use the HAVING keyword to only return those who have zero orders and look at the output metadata to see how many rows were returned.

|  |
| --- |
| with temp\_table as (  select  customername,  count(ordernumber) as num\_orders  from initech\_customers ic  left join initech\_orders io  on io.customernumber = ic.customernumber  group by 1  having count(ordernumber) = 0  )  SELECT count(\*)  from temp\_table |

|  |
| --- |
| 24 |

1. It is common for databases to store the order number and the order details in separate data tables. This helps reduce redundancy in the database. Write a query that returns the order number and the total number of items placed (**quantityordered**) for each order. How many items were sold in the order with the most items?

|  |
| --- |
| SELECT  io.ordernumber,  sum(quantityordered) as total\_items  from initech\_orders io  join initech\_orderdetails id  on io.ordernumber = id.ordernumber  group by 1  order by sum(quantityordered) desc  limit 1 |

|  |
| --- |
| 717 |