SQL Joins

Question 1

```
--Questions 1
--INNER JOIN: Orders with Customer and Product Names Question:
--List all orders along with the customer name and product name.
--Expected Output Columns:
--OrderID, OrderDate, CustomerName, ProductName, Quantity

select a.OrderID,···
from orders_large as A
inner join customers_large as B
on A.CustomerID = B.CustomerID
inner join products_large as C
on A.ProductID = C.ProductID
limit 5;
```

See performance (1) Optimize

Table +

	OrderID	OrderDate	Quantity	CustomerName	ProductName
1	1	2023-06-10	10	Customer_1251	Product_2014
2	2	2023-12-07	5	Customer_1236	Product_2004
3	3	2024-10-26	9	Customer_1170	Product_2171
4	4	2023-02-17	2	Customer_1344	Product_2007
5	5	2024-11-06	2	Customer_1319	Product_2061

Question 2

```sql
--2. INNER JOIN: Customers Who Placed Orders Question:
--Which customers have placed at least one order?
--Expected Output Columns:

--CustomerID, CustomerName, Country, OrderID, OrderDate
select A.CustomerID,
        A.CustomerName,
        A.Country,
        B.OrderID,
        B.OrderDate,
        count(a.CustomerID) as total_orders
        from customers_large as A
inner join orders_large as B
on A.CustomerID = B.CustomerID
group by A.CustomerID,
        A.CustomerName,
        A.Country,
        B.OrderID,
        B.OrderDate
        having count(a.CustomerID) > 0
limit 5;
```

See performance (1)

Table +

	CustomerID	CustomerName	Country	OrderID	OrderDate	total_orders
1	1266	Customer_1266	Australia	296	2023-01-23	1
2	1187	Customer_1187	UK	431	2023-01-01	1
3	1048	Customer_1048	India	846	2024-06-15	1
4	1266	Customer_1266	Australia	1176	2023-01-02	1
5	1310	Customer_1310	Germany	3459	2024-08-03	1

Question 3

```sql
/*3. LEFT JOIN: All Customers and Their Orders Question:
List all customers and any orders they might have placed. Include customers who have not placed any orders.
Expected Output Columns:
•
CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity*/

select a.*,
        b.OrderID,
        b.OrderDate,
        b.ProductID,
        b.Quantity
from customers_large as a
left join orders_large as b
on a.CustomerID = b.CustomerID
limit 5;
```

See performance (1)

Table +

	CustomerID	CustomerName	Country	OrderID	OrderDate	ProductID	Quantity
1	1001	Customer_1001	India	3408	2024-04-15	2100	5
2	1002	Customer_1002	Germany	3269	2024-06-07	2058	9
3	1003	Customer_1003	USA	3318	2024-10-17	2144	10
4	1004	Customer_1004	Germany	3910	2024-04-20	2145	8
5	1005	Customer_1005	USA	3986	2024-03-27	2111	2

5 rows | 1.34s runtime

Question 4

```sql
/*4. LEFT JOIN: Product Order Count Question:
List all products and how many times each was ordered (if any).
Expected Output Columns:
•
ProductID, ProductName, TotalOrders
(TotalOrders is the count of how many times the product appears in orders)*/

select a.ProductID,
       a.ProductName,
       count(b.ProductID) as TotalOrders
from products_large as a
left join orders_large as b
on a.ProductID = b.ProductID
group by a.ProductID,
       a.ProductName
limit 5;
```

See performance (1)

Table +

	ProductID	ProductName	TotalOrders
1	2161	Product_2161	20
2	2006	Product_2006	13
3	2061	Product_2061	18
4	2130	Product_2130	25
5	2199	Product_2199	17

5 rows | 1.68s runtime

Question 5

```sql
/*5. RIGHT JOIN: Orders with Product Info (Include Products Not Ordered) Question:
Find all orders along with product details, including any products that might not have been ordered.
Expected Output Columns:
•
OrderID, OrderDate, ProductID, ProductName, Price, Quantity*/
select a.OrderID,
       a.OrderDate,
       a.ProductID,
       b.ProductName,
       b.Price,
       a.Quantity
from orders_large as a
right join products_large as b
on a.ProductID = b.ProductID
limit 5;
```

See performance (1)

Table

	OrderID	OrderDate	ProductID	ProductName	Price	Quantity
1	3788	2023-11-15	2001	Product_2001	833	7
2	3767	2023-02-11	2002	Product_2002	1558	3
3	3925	2023-02-28	2003	Product_2003	1398	4
4	3974	2025-01-30	2004	Product_2004	1996	3
5	3786	2023-06-18	2005	Product_2005	1146	3

Question 6

```sql
/*6. RIGHT JOIN: Customer Info with Orders (Include All Customers) Question:
Which customers have made orders, and include customers even if they have never placed an order.
Expected Output Columns:
•
CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity*/

select a.CustomerID,
       a.CustomerName,
       a.Country,
       b.OrderID,
       b.orderdate,
       b.productid,
       b.quantity
from customers_large as a
right join orders_large as b
on a.CustomerID = b.CustomerID
limit 5;
```

See performance (1)

Table

	CustomerID	CustomerName	Country	OrderID	orderdate	productid	quantity
1	1251	Customer_1251	Germany	1	2023-06-10	2014	10
2	1236	Customer_1236	Australia	2	2023-12-07	2004	5
3	1170	Customer_1170	Germany	3	2024-10-26	2171	9
4	1344	Customer_1344	Canada	4	2023-02-17	2007	2
5	1319	Customer_1319	USA	5	2024-11-06	2061	2

Question 7

```sql
/*7. FULL OUTER JOIN: All Customers and All Orders
Question:
List all customers and orders, showing NULLs where customers have not ordered or where orders have no customer info.
Expected Output Columns:
•
CustomerID, CustomerName, Country, OrderID, OrderDate, ProductID, Quantity*/

select a.customerid,
       a.customername,
       a.country,
       b.orderid,
       b.orderdate,
       b.productid,
       b.quantity
from customers_large as a
full outer join orders_large as b
on a.customerid = b.customerid
limit 5;
```

Table +

	customerid	customername	country	orderid	orderdate	productid	quantity
1	1191	Customer_1191	Germany	18	2023-05-12	2101	7
2	1191	Customer_1191	Germany	499	2023-01-18	2008	3
3	1191	Customer_1191	Germany	636	2023-06-10	2057	5
4	1191	Customer_1191	Germany	1016	2023-10-05	2004	6
5	1191	Customer_1191	Germany	1825	2024-10-29	2068	3

Question 8

```
/*8. FULL OUTER JOIN: All Products and Orders Question:
List all products and orders, showing NULLs where products were never ordered or orders are missing product info.
Expected Output Columns:
•
ProductID, ProductName, Price, OrderID, OrderDate, CustomerID, Quantity*/

select a.productid,
       a.productname,
       a.price,
       b.orderid,
       b.orderdate,
       b.customerid,
       b.quantity
from products_large as a
full outer join orders_large as b
on a.productid = b.productid
limit 5;
```

See performance (1)

Table

	productid	productname	price	orderid	orderdate	customerid	quantity
1	2091	Product_2091	563	785	2024-09-25	1025	1
2	2091	Product_2091	563	924	2023-12-20	1167	5
3	2091	Product_2091	563	999	2023-06-03	1264	4
4	2091	Product_2091	563	1264	2024-05-26	1393	6
5	2091	Product_2091	563	1372	2024-12-13	1059	7