SQL Exercises

Business Intelligence Bootcamp by Ai DataYard

Introduction

These slides covers a set of SQL practice questions given by the trainer in the bootcamp of **BI by Ai Datayard** to strengthen my understanding of **JOINs**, **aggregation**, and **filtering** in relational databases.

These queries focus on combining data from multiple tables such as Customers, Orders and Products using real-world business scenarios.

The goal is to help me:

- Learn how to retrieve meaningful data using different types of **JOINs**
- Perform **aggregations** like totals and counts
- Understand how SQL helps in solving real-world data problems

Dataset Overview

There are 3 tables:

1. **Customers** (all the info of the customers)
2. **Orders** (all the info about the order that customer place)
3. **Products** (all the info of the products that were in the orders)

Dataset Overview

1. Customers Table

CustomerID	CustomerName	Age	City	Country	Email	PhoneNumber
1	Nathan Espinoza PhD	61	Lake Justinmouth	Holy See (Vatican City State)	zgarcia@yahoo.com	+1-140-026-5817x02274
2	Colleen Dougherty	21	New Dustin	Guadeloupe	elizabethmorgan@wilson.com	910-063-1781x80070
3	Samuel Kelly	38	Lake Richard	Greenland	gcampbell@gmail.com	7602143447
4	Gabrielle James	66	Jonesshire	Bangladesh	rferguson@warner.com	842.023.8783x35129
5	Sandra Castillo	52	South Sabrina	Latvia	ccruz@yahoo.com	522.620.4489
6	Jasmine Lynch	58	Gomezborough	Vanuatu	miguel52@black.com	001-924-599-4950x497
7	Jennifer Gardner	33	New Paigeview	Qatar	samantha26@gmail.com	612.203.8168x88754
8	Ronnie Mcbride	32	North Jimmy	Antarctica (the territory South of 60 deg S)	uwatson@hotmail.com	050-715-1335x54616
9	Jonathan Hobbs	65	Frankmouth	Libyan Arab Jamahiriya	hillmario@harrison-wilson.biz	(169)582-8506x505
10	James Christian	23	Andersonton	Zambia	samanthasmith@gmail.com	126.100.1771
11	Chelsea Anderson	51	Jessicaville	Malaysia	carl19@turner.info	205.191.4476
12	Michelle Clark	29	Lake Austin	Myanmar	abrown@yahoo.com	(670)306-4794x4370
13	Patricia Jones	62	North Sarahhaven	Mexico	jason93@sutton.com	342-232-4043
14	Linda Wu	23	West Erica	Solomon Islands	daniel69@freeman-riley.net	(177)601-1234x2760
15	Stephanie Mitchell	30	Shawnton	Myanmar	jrice@gmail.com	971-562-0952x688
16	Tracie Morris	40	Ianstad	Turkmenistan	moralestammy@yahoo.com	(619)586-8544
17	Kimberly Davis	36	North Erika	Dominica	qburton@sharp.biz	4512139680
18	Eric Torres	56	Lisaland	French Guiana	andrea85@butler.com	001-463-694-4564x4332

Dataset Overview

2. Orders Table

OrderID	CustomerID	ProductID	Quantity	TotalAmount	City	OrderDate
1	47	3	5	2083.7	Lauratown	2025-01-04
2	7	12	1	1863.07	Port Sethmouth	2025-03-05
3	33	14	2	4675.7	Charlesfurt	2025-02-08
4	22	4	3	3190.27	East Timothytown	2025-01-12
5	8	20	2	2078.35	Freemanside	2025-01-16
6	23	19	1	4178.44	Mariahfurt	2025-02-19
7	46	18	4	958.55	Jenniferhaven	2025-02-21
8	6	12	3	1395.99	Haystown	2025-01-09
9	31	4	2	2198.2	Port Brentland	2025-01-19
10	50	4	5	1829.49	Dawnstad	2025-02-22
11	6	17	3	210.68	Port Jamesshire	2025-03-14
12	19	8	4	4012.17	East Emmaberg	2025-01-16
13	11	1	1	1106.21	New Steve	2025-03-09
14	32	5	2	3987.95	Johnsonbury	2025-01-04
15	13	13	4	4424.5	New Brandon	2025-02-02
16	8	8	2	3464.05	North Tiffany	2025-02-09
17	24	18	3	1207.53	Stricklandstad	2025-01-16
18	42	5	1	2599.17	West Sharon	2025-02-24

Dataset Overview

3. Products Table

ProductID	ProductName	Category	Price
1	Fall Pm	Electronics	852.7
2	Adult Run	Home Appliances	890.97
3	Age Plan	Beauty	1131.38
4	With Little	Clothing	1944.13
5	Without Value	Books	876.98
6	Teach Growth	Home Appliances	938.33
7	War Fire	Clothing	329.95
8	Hour Idea	Beauty	1440.86
9	College Despite	Electronics	122.5
10	Throughout Suddenly	Home Appliances	1604.09
11	Although Suggest	Beauty	491.87
12	Turn Him	Electronics	934.47
13	Front Become	Clothing	630.57
14	Trip Everybody	Clothing	1408.24
15	Allow People	Beauty	346.77
16	Believe Page	Beauty	983.62
17	Face Run	Clothing	822.93
18	Trade Son	Beauty	292.79

SQL Joins

1. Retrieve all customers and their corresponding orders. If a customer has no orders, still include them.

```
SELECT
    *
FROM
    customers c
        LEFT JOIN
    orders o ON c.CustomerID = o.CustomerID
```

SQL Joins

1. Retrieve all customers and their corresponding orders. If a customer has no orders, still include them.

CustomerID	CustomerName	Age	City	Country	Email
1	Nathan Espinoza PhD	61	Lake Justinmouth	Holy See (Vatican City State)	zgarcia@yahoo.com
2	Colleen Dougherty	21	New Dustin	Guadeloupe	elizabethmorgan@wilson.com
3	Samuel Kelly	38	Lake Richard	Greenland	gcampbell@gmail.com
4	Gabrielle James	66	Jonesshire	Bangladesh	rferguson@warner.com
5	Sandra Castillo	52	South Sabrina	Latvia	ccruz@yahoo.com
6	Jasmine Lynch	58	Gomezborough	Vanuatu	miguel52@black.com
6	Jasmine Lynch	58	Gomezborough	Vanuatu	miguel52@black.com
7	Jennifer Gardner	33	New Paigeview	Qatar	
7	Jennifer Gardner	33	New Paigeview	Qatar	
7	Jennifer Gardner	33	New Paigeview	Qatar	
8	Ronnie Mcbride	32	North Jimmy	Antarctica (the territory	
8	Ronnie Mcbride	32	North Jimmy	Antarctica (the territory	
9	Jonathan Hobbs	65	Frankmouth	Libyan Arab Jamahiriya	

PhoneNumber	OrderID	CustomerID	ProductID	Quantity	TotalAmount	City	OrderDate
+1-140-026-5817x02274	NULL	NULL	NULL	NULL	NULL	NULL	NULL
910-063-1781x80070	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7602143447	32	3	1	3	1130.64	East Katherine	2025-01-11
842.023.8783x35129	34	4	8	1	1704.01	East Austinmouth	2025-02-26
522.620.4489	NULL	NULL	NULL	NULL	NULL	NULL	NULL
001-924-599-4950x497	11	6	17	3	210.68	Port Jamesshire	2025-03-14
001-924-599-4950x497	8	6	12	3	1395.99	Haystown	2025-01-09
612.203.8168x88754	38	7	13	4	2567.04	Nicholaschester	2025-01-14
612.203.8168x88754	27	7	9	3	2608.17	Lake Kevin	2025-02-07
612.203.8168x88754	2	7	12	1	1863.07	Port Sethmouth	2025-03-05
050-715-1335x54616	16	8	8	2	3464.05	North Tiffany	2025-02-09
050-715-1335x54616	5	8	20	2	2078.35	Freemanside	2025-01-16
(169)582-8506x505	28	9	18	3	637.12	Port Adam	2025-02-22

SQL Joins

2. Get a list of all orders along with customer details.

```
SELECT
    c.CustomerID, c.CustomerName, o.OrderID, o.TotalAmount
FROM
    orders o
        LEFT JOIN
    customers c ON o.CustomerID = c.CustomerID
```

CustomerID	CustomerName	OrderID	TotalAmount
47	Catherine Zhang	1	2083.7
7	Jennifer Gardner	2	1863.07
33	Melissa Henderson	3	4675.7
22	Michelle Craig	4	3190.27
8	Ronnie Mcbride	5	2078.35
23	Heather Fry	6	4178.44
46	Jeffery Swanson	7	958.55
6	Jasmine Lynch	8	1395.99
31	Jennifer Reyes	9	2198.2
50	Joy Rogers	10	1829.49
6	Jasmine Lynch	11	210.68
19	Thomas Mclaughlin	12	4012.17
11	Chelsea Anderson	13	1106.21
32	Sara Vaughn	14	3987.95

SQL Joins

3. Find all products that have never been ordered.

```
SELECT
    *
FROM
    products p
        LEFT JOIN
    orders o ON p.ProductID = o.ProductID
WHERE o.OrderID IS NULL
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	ProductID	ProductName	Category	Price	OrderID	CustomerID	ProductID	Quantity	TotalAmount	City	OrderDate
▶	16	Believe Page	Beauty	983.62	NULL	NULL	NULL	NULL	NULL	NULL	NULL

DataYard

MySQL

SQL Joins

4. Retrieve all customers who have placed at least one order.

```
SELECT
    DISTINCT c.CustomerID,c.CustomerName
FROM
    customers c
        LEFT JOIN
    orders o ON c.CustomerID = o.CustomerID
```

CustomerID	CustomerName
1	Nathan Espinoza PhD
2	Colleen Dougherty
3	Samuel Kelly
4	Gabrielle James
5	Sandra Castillo
6	Jasmine Lynch
7	Jennifer Gardner
8	Ronnie Mcbride
9	Jonathan Hobbs
10	James Christian
11	Chelsea Anderson
12	Michelle Clark
13	Patricia Jones
14	Linda Wu

DataYard

MySQL

SQL Joins

5. List all orders along with the product details.

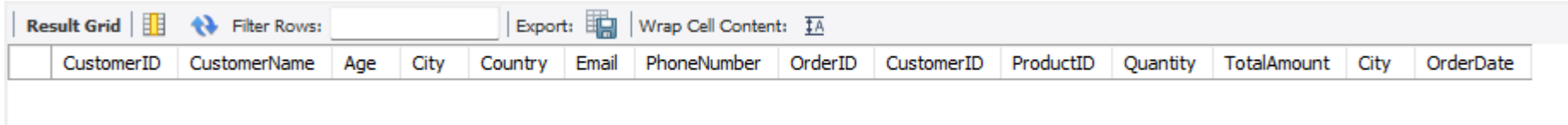
```
SELECT
    o.OrderID,o.OrderDate,p.ProductID,p.ProductName,p.Category,p.Price
FROM
    orders o
        LEFT JOIN
    products p ON p.ProductID = o.ProductID
```

OrderID	OrderDate	ProductID	ProductName	Category	Price
1	2025-01-04	3	Age Plan	Beauty	1131.38
2	2025-03-05	12	Turn Him	Electronics	934.47
3	2025-02-08	14	Trip Everybody	Clothing	1408.24
4	2025-01-12	4	With Little	Clothing	1944.13
5	2025-01-16	20	On Always	Home Appliances	1858.01
6	2025-02-19	19	Candidate Recently	Clothing	1700.37
7	2025-02-21	18	Trade Son	Beauty	292.79
8	2025-01-09	12	Turn Him	Electronics	934.47
9	2025-01-19	4	With Little	Clothing	1944.13
10	2025-02-22	4	With Little	Clothing	1944.13
11	2025-03-14	17	Face Run	Clothing	822.93
12	2025-01-16	8	Hour Idea	Beauty	1440.86
13	2025-03-09	1	Fall Pm	Electronics	852.7
14	2025-01-04	5	Without Value	Books	876.98

SQL Joins

6. Find all orders placed in "New York" along with customer details.

```
SELECT
    *
FROM
    customers c
        RIGHT JOIN
    orders o ON c.CustomerID = o.CustomerID
WHERE c.City = "New York"
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	CustomerName	Age	City	Country	Email	PhoneNumber	OrderID	CustomerID	ProductID	Quantity	TotalAmount	City	OrderDate

There is no customer from New York.

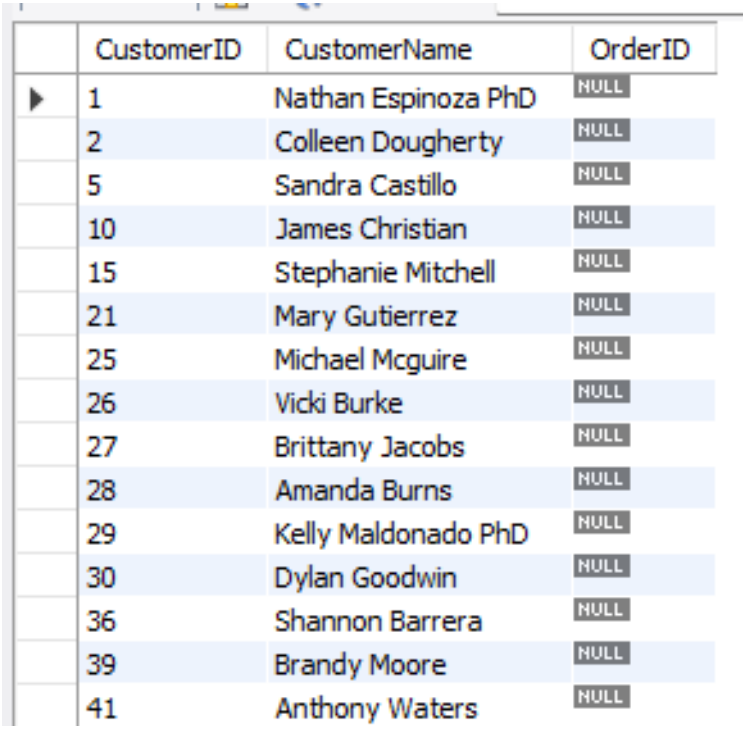
DataYard

MySQL

SQL Joins

7. Get a list of customers who have never placed an order.

```
SELECT
    c.CustomerID,c.CustomerName, o.OrderID
FROM
    customers c
       Left JOIN
    orders o ON c.CustomerID = o.CustomerID
WHERE o.OrderID IS NULL
```



CustomerID	CustomerName	OrderID
1	Nathan Espinoza PhD	NULL
2	Colleen Dougherty	NULL
5	Sandra Castillo	NULL
10	James Christian	NULL
15	Stephanie Mitchell	NULL
21	Mary Gutierrez	NULL
25	Michael Mcguire	NULL
26	Vicki Burke	NULL
27	Brittany Jacobs	NULL
28	Amanda Burns	NULL
29	Kelly Maldonado PhD	NULL
30	Dylan Goodwin	NULL
36	Shannon Barrera	NULL
39	Brandy Moore	NULL
41	Anthony Waters	NULL

SQL Joins

8. Retrieve all product categories along with the products that belong to them.

```
SELECT
    ProductName, Category
FROM
    products
```

ProductName	Category
Fall Pm	Electronics
Adult Run	Home Appliances
Age Plan	Beauty
With Little	Clothing
Without Value	Books
Teach Growth	Home Appliances
War Fire	Clothing
Hour Idea	Beauty
College Despite	Electronics
Throughout Suddenly	Home Appliances
Although Suggest	Beauty
Turn Him	Electronics
Front Become	Clothing
Trip Everybody	Clothing
Allow People	Beauty
Believe Page	Beauty
Face Run	Clothing
Trade Son	Beauty

SQL Joins

9. Find the total amount spent by each customer (include customers who haven't spent anything).

```
SELECT
    c.CustomerID,c.CustomerName , sum(o.TotalAmount) as TotalSpent
FROM
    customers c
       LEFT JOIN
    orders o ON c.CustomerID = o.CustomerID
group by c.CustomerID,c.CustomerName
```

CustomerID	CustomerName	TotalSpent
1	Nathan Espinoza PhD	NULL
2	Colleen Dougherty	NULL
3	Samuel Kelly	1130.64
4	Gabrielle James	1704.01
5	Sandra Castillo	NULL
6	Jasmine Lynch	1606.67
7	Jennifer Gardner	7038.28
8	Ronnie Mcbride	5542.4
9	Jonathan Hobbs	637.12
10	James Christian	NULL
11	Chelsea Anderson	1725.610...
12	Michelle Clark	3026.04
13	Patricia Jones	9416.18
14	Linda Wu	972.88
15	Stephanie Mitchell	NULL
16	Tracie Morris	2794.24
17	Kimberly Davis	6105.36
18	Eric Torres	1890.56

SQL Joins

10. Retrieve all orders along with the respective customer's city.

```
SELECT
    o.OrderID,c.City
FROM
    customers c
        INNER JOIN
    orders o ON c.CustomerID = o.CustomerID
```

OrderID	City
1	South Robert
2	New Paigeview
3	West Janetborough
4	South Megan
5	North Jimmy
6	Christopherview
7	New Matthew
8	Gomezborough
9	South Steven
10	West Josephfurt
11	Gomezborough
12	Coopershire
13	Jessicaville
14	West Robert
15	North Sarahhaven
16	North Jimmy

SQL Joins

11. List all customers who have purchased products from the "Electronics" category.

```
SELECT
   c.CustomerID,  c.CustomerName, p.ProductName, p.Category
FROM
    customers c
    JOIN
    orders o ON c.CustomerID = o.CustomerID
    JOIN
    products p ON p.productID = o.productID
WHERE p.Category = "Electronics"
```

CustomerID	CustomerName	ProductName	Category
3	Samuel Kelly	Fall Pm	Electronics
6	Jasmine Lynch	Turn Him	Electronics
7	Jennifer Gardner	College Despite	Electronics
7	Jennifer Gardner	Turn Him	Electronics
11	Chelsea Anderson	College Despite	Electronics
11	Chelsea Anderson	Fall Pm	Electronics
18	Eric Torres	Turn Him	Electronics
35	Andrea Gill	Turn Him	Electronics
46	Jeffery Swanson	Turn Him	Electronics

SQL Joins

12.Find customers who have made purchases of more than $500 in a single order.

```
SELECT
    c.CustomerID,c.CustomerName, o.TotalAmount
FROM
    customers c
        LEFT JOIN
    orders o ON c.CustomerID = o.CustomerID
WHERE o.TotalAmount > 500
```

CustomerID	CustomerName	TotalAmount
3	Samuel Kelly	1130.64
4	Gabrielle James	1704.01
6	Jasmine Lynch	1395.99
7	Jennifer Gardner	2567.04
7	Jennifer Gardner	2608.17
7	Jennifer Gardner	1863.07
8	Ronnie Mcbride	3464.05
8	Ronnie Mcbride	2078.35
9	Jonathan Hobbs	637.12
11	Chelsea Anderson	619.4
11	Chelsea Anderson	1106.21
12	Michelle Clark	3026.04
13	Patricia Jones	4991.68
13	Patricia Jones	4424.5
14	Linda Wu	972.88
16	Tracie Morris	2794.24
17	Kimberly Davis	1637.92

DataYard

MySQL

SQL Joins

13. Retrieve all customers and the total quantity of products they have purchased.

```
SELECT
    c.CustomerID,c.CustomerName , sum(o.Quantity) as TotalQuantity
FROM
    customers c
       LEFT JOIN
    orders o ON c.CustomerID = o.CustomerID
group by c.CustomerID,c.CustomerName
```

CustomerID	CustomerName	TotalQuantity
1	Nathan Espinoza PhD	NULL
2	Colleen Dougherty	NULL
3	Samuel Kelly	3
4	Gabrielle James	1
5	Sandra Castillo	NULL
6	Jasmine Lynch	6
7	Jennifer Gardner	8
8	Ronnie Mcbride	4
9	Jonathan Hobbs	3
10	James Christian	NULL
11	Chelsea Anderson	6
12	Michelle Clark	5
13	Patricia Jones	6
14	Linda Wu	5
15	Stephanie Mitchell	NULL

SQL Joins

14. Get a list of all products and the number of times each has been ordered.

```
SELECT p.ProductID, p.ProductName,
       COUNT(o.OrderID) AS TimesOrdered
FROM Products p
LEFT JOIN orders o ON p.ProductID = o.ProductID
GROUP BY p.ProductID, p.ProductName;
```

ProductID	ProductName	TimesOrdered
1	Fall Pm	2
2	Adult Run	2
3	Age Plan	2
4	With Little	3
5	Without Value	4
6	Teach Growth	1
7	War Fire	1
8	Hour Idea	5
9	College Despite	2
10	Throughout Suddenly	2
11	Although Suggest	3
12	Turn Him	5
13	Front Become	3
14	Trip Everybody	4
15	Allow People	1
16	Believe Page	0

SQL Joins

15. Retrieve the most expensive order placed along with customer details.

```
SELECT
   o.OrderId, c.CustomerID,c.CustomerName , sum(o.TotalAmount) as TotalSpent
FROM
    customers c
       LEFT JOIN
    orders o ON c.CustomerID = o.CustomerID
group by o.OrderId, c.CustomerID,c.CustomerName
Order by TotalSpent DESC
LIMIT 1
```

Result Grid | Filter Rows: | Export:

OrderId	CustomerID	CustomerName	TotalSpent
47	13	Patricia Jones	4991.68