SQL Exercise: Union and Union ALL - Exercise 7

1 SELECT customer_name
FROM online_sales
UNION
SELECT customer_name
FROM store_sales;

Expected output table

customer_name
Alice
Brian
Carol
Daniel
Emma
Fiona
George
Henry

2 SELECT customer_name
FROM online_sales
UNION All
SELECT customer_name
FROM store_sales;

Expected output table

customer_name
Alice
Brian
Carol
Daniel

Emma

Fiona

Brian

George

Alice

Henry

displays all customer names, including duplicates like 'Alice' & 'Brian' who appear in both tables

3. Unique Sales Dates

Syntax

```
SELECT      sale_date
FROM        online_sales
UNION
SELECT      sale_date
FROM        store_sales
Order BY    sale_date ASC;
```

Expected Output table

sale_date

2025-01-12

2025-01-20

2025-02-05

2025-02-08

2025-03-10

2023-03-25

2025-04-15

2025-04-18

2025-05-02

2025-05-05

UNION combines multiple lists into one, removing any duplicates, & ORDER BY puts that final list in chronological order.

4. All sale Dates (including Duplicates)

SQL Code

```
SELECT      sale_date
FROM        online_sales
UNION ALL
SELECT      sale_date
FROM        store_sales
ORDER BY    sale_date ASC;
```

Expected Output table

sale_date
2025-01-12
2025-01-20
2025-02-05
2025-02-08
2025-03-10
2025-03-25
2025-04-15
2025-04-18
2025-05-02
2025-05-05

* All dates are unique and not duplicated

5. High - Value Customes

SQL Code

```
SELECT      customer_name, amount
FROM        online_sales
WHERE       amount > 250
UNION
SELECT      customer_name, amount
FROM        store_sales
WHERE       amount > 250
```

Expected Output:

Customer_name	amount
Carol	300
George	310
Henry	270

6. Combined Sales Data

```
SELECT      customer_name, amount, sale_date
FROM        online_sales
UNION ALL
SELECT      customer_name, amount, sale_date
FROM        store_sales;
```

customer_name	amount	sale_date
ALICE	150	2025-01-12
BRIAN	250	2025-02-05
CAROL	300	2025-03-10
Daniel	220	2025-04-15
Emma	180	2025-05-02
Fiona	200	2025-01-20
Brian	250	2025-02-08
George	310	2025-03-25
Alice	150	2025-04-18
Henry	270	2025-05-05

- UNION ALL combines every single row from both tables into one list

7. Add Sales Source Label

```
SELECT      customer_name, amount, sale_date, 'online'
            AS source
FROM        online_sales
UNION ALL
SELECT      customer_name, amount, sale_date, 'store'
            AS source
FROM        store_sales;
```

customer_name	amount	sale_date	source
Alice	150	2025-01-12	Online
Brian	250	2025-02-05	Online
Carol	300	2025-03-10	Online
Daniel	220	2025-04-15	Online
Emma	180	2025-05-02	Online
Fiona	200	2025-01-20	Store
Brian	250	2025-02-08	Store
George	310	2025-03-25	Store
Alice	150	2025-04-18	Store
Henry	210	2025-05-05	Store

The source column explains where each sale came from.

8. Customers Appearing in Both Tables

~~SELECT customer_name, COUNT(*)~~

~~AS occurences~~

~~FROM~~

~~SELECT customer_name, online_sales~~

~~UNION ALL~~

~~SELECT customer_name FROM store_sales~~

```
SELECT      customer_name, COUNT(*)          } This is my final result / How many total rows
            AS occurences                      for that customer
FROM (                                       - merging one big pile before analyzing
SELECT      customer_name FROM online_sales  - The 1st half
UNION ALL
SELECT      customer_name FROM store_sales   - The 2nd half
                                               Makes one table with all
            ) AS combined_sales              - customer names from both sources
GROUP BY    customer_name
HAVING      count(*) > 1;                    - only keeping customers that
                                               appeared in both tables
```

Output columns

customer_name	Occurences
Alice	2
Brian	2

9. Total combined sales

```
SELECT          SUM(amount)
                AS total_amount
FROMC
SELECT          amount FROM online_sales
UNION ALL
SELECT          amount FROM store_sales
                )AS combined_amounts;
```

Output Column

total_amount
2280

Bonus Challenge

```
SELECT          customer_name,
                SUM(amount) AS total_spent
FROMC
SELECT          customer_name, amount
                FROM online_sales
UNION ALL
SELECT          customer_name, amount
                FROM store_sales
                )AS combined
GROUP BY        customer_name
ORDER BY        total_spent DESC;
```

- adds all customer amounts together.
- Everything inside the brackets becomes one combined list.
- Takes the name & amount from the online sales list
- If a customer bought in both places we want both rows counted.
- names the combined list, so we can use it like a table
- ensures the biggest spenders appear first — highest totals on top.

Output column

customer_name	total_spent
George	310
Carol	300
Henry	270
Brian	500
Daniel	220
Fiona	200
Alice	300
Emma	180