

1. Unique Customer Names

List all unique customer names from both tables using UNION. — removes duplicates

Output col: customer-name

```
SELECT sale_id,  
       customer_name,  
FROM online_sales  
UNION  
SELECT sale_id,  
       customer_name,  
FROM store_sales;
```

Return:

Customer-name
Alice
Brian
Carol
Daniel
Emma
Fiona
George
Henry

2. All Customers (Including Duplicates)

List all cx names from both tables using UNION ALL.

Output Col: customer-name

```
SELECT customer-name,  
FROM online_sales  
UNION ALL  
SELECT customer-name,  
FROM store_sales;
```

customer-name

Alice
Brian
Carol
Daniel
Emma
Fiona
Brian
George
Alice
Henry

3. Unique sale dates.

Show all unique sale dates from both tables in ascending order.

Output col: sale-date

```
SELECT sale-date,  
FROM online-sales  
UNION  
SELECT sale-date,  
FROM store-date  
ORDER BY sale-date ASC;
```

Return:

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

S D F G H U I O P

4. All Sale Dates (Including Duplicates)
List all sale dates (with duplicates) using UNION ALL.
Output col : sale_date

```
SELECT sale_date,  
FROM online_sales,  
UNION ALL  
SELECT sale_date  
FROM store_date;
```

Return :

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

5. High-Value Customers

Combine both tables and list unique customers who made purchases greater than ~~at~~ 250.
Output col : customer_name, amount

8 9

WHERE statement to filter info

```

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

```

Return:

customer_name	amount
Henry	300
Carol	310
George	270

6. Combined Sales Data

Combine all records from both tables using UNION ALL.

Output col: customer_name, amount, sale_date

```

SELECT customer_name,
       amount,
       sale_date,
FROM online_sale
UNION ALL
SELECT customer_name,
       amount,
       sale_date,
FROM store_sales;

```

Return :

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

7. Add Sales Source Label

Combine both tables but include a new column 'source' that indicates whether the sale was made online or store.

Output col: customer-name, amount, sale-date, source

```
SELECT customer-name,  
       amount,  
       sale-date,  
       'Online' AS source
```

FROM online-sales,

UNION ALL

```
SELECT customer-name,
```

amount,

sale-date

'Online' AS source

FROM store-sales;

Return :

customer-name	amount	sale-date	source
Alice	150	2025-01-12	Online
Brian	250	2025-02-05	Online
Carol	300	2025-04-15	Online
Daniel	220	2025-05-02	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	2025-04-15	Store
Henry	270	2025-05-05	Store

Q5. Customers Appearing in Both Tables

Find all customers who appear in both online-sales and store-sales (Hint : use UNION ALL, GROUP BY and HAVING)

Output col : customer-name, occurrences

~~SELECT customer-name,~~

~~(O~~

~~SELECT COUNT(*) As Occurrences~~

~~customer-name,~~

~~FROM online-sales~~

~~UNION ALL~~

~~SELECT customer-name,~~

~~FROM store-sales~~

~~As all customers~~

~~GROUP BY customer-name~~

~~HAVING COUNT(*) > 1;~~

Q W E R T Y U I O P
A S D F G H J K L

Return :

customer-name	occurrences
Alice	2
Brian	2

* 9. Total Combined Sales

Combine both tables using UNION ALL and calculate the total sales amount across both.

Output col: total amount

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

ADD ALL AMOUNTS

$$(150 + 250 + 300 + 220 + 180) + (200 + 250 + 310 + 150 + 270)$$
$$= 2330$$

Output / Return :

Total amount

2330

* Bonus Challenge (Optional)

Create a single query that lists each customer's total combined amount from both tables (sum of all their purchases).

Output columns: customer-name, total_spent

~~SELECT customer_name~~

~~SELECT SUM(amount) AS total_spent
FROM online_sales~~

~~SELECT customer_name,
amount~~

~~FROM store_sales,~~

~~AS combined_sales~~

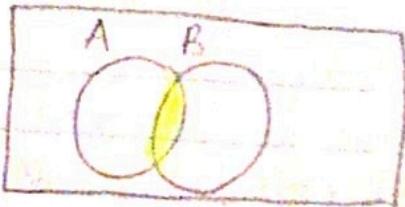
~~GROUP BY customer_name~~

~~ORDER BY total_spent DESC;~~

Return:

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

14.10.2025 Revision Joins Q1 Exercise - Corrections



- You can select any columns between the two tables.
- The type of Join we use affect the rows returned.