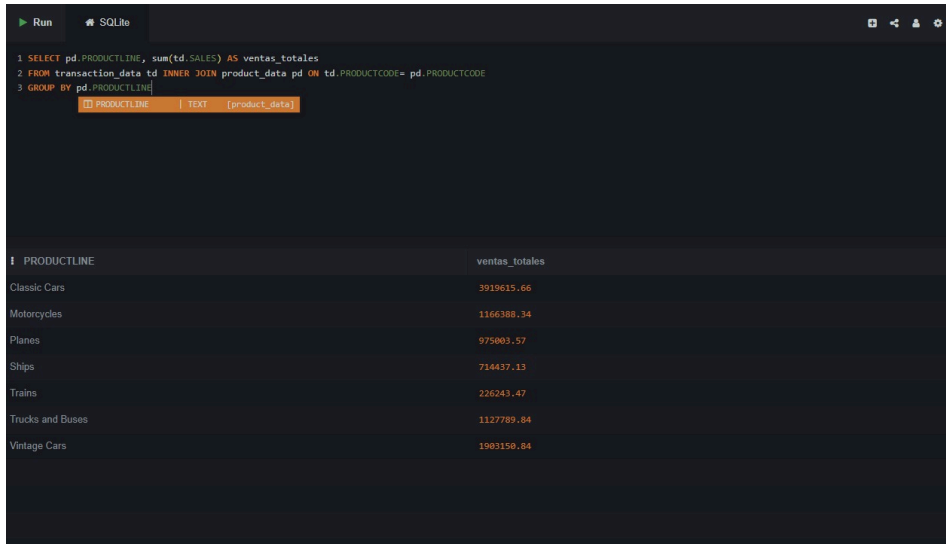


EJERCICIOS BIG DATA-----smart data

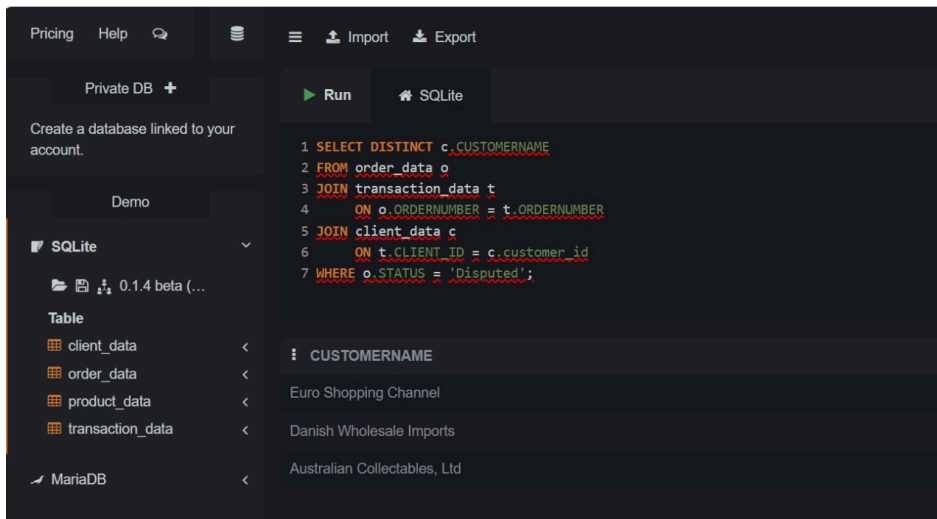
1. SELECT pd.PRODUCTLINE, sum(td.SALES) as ventas_totales
from transaction_data td inner JOIN product_data pd ON td.PRODUCTCODE=
pd.PRODUCTCODE
GROUP by pd.PRODUCTLINE



```
1 SELECT pd.PRODUCTLINE, sum(td.SALES) AS ventas_totales
2 FROM transaction_data td INNER JOIN product_data pd ON td.PRODUCTCODE = pd.PRODUCTCODE
3 GROUP BY pd.PRODUCTLINE
```

PRODUCTLINE	ventas_totales
Classic Cars	3919615.66
Motorcycles	1166388.34
Planes	975803.57
Ships	714437.13
Trains	226243.47
Trucks and Buses	1127789.84
Vintage Cars	1983150.84

2. SELECT DISTINCT c.CUSTOMERNAME
FROM order_data o
JOIN transaction_data t
ON o.ORDERNUMBER = t.ORDERNUMBER
JOIN client_data c
ON t.CLIENT_ID = c.customer_id
WHERE o.STATUS = 'Disputed';



```
1 SELECT DISTINCT c.CUSTOMERNAME
2 FROM order_data o
3 JOIN transaction_data t
4 ON o.ORDERNUMBER = t.ORDERNUMBER
5 JOIN client_data c
6 ON t.CLIENT_ID = c.customer_id
7 WHERE o.STATUS = 'Disputed';
```

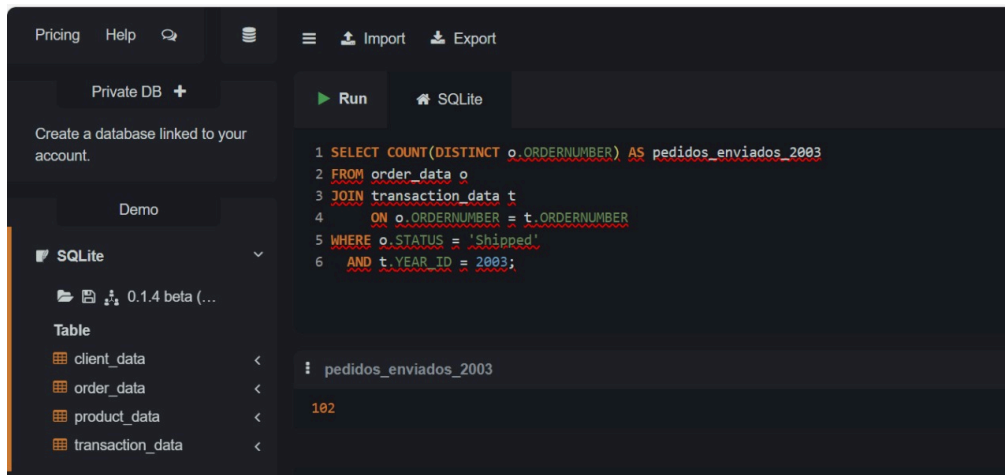
CUSTOMERNAME
Euro Shopping Channel
Danish Wholesale Imports
Australian Collectables, Ltd

3. SELECT COUNT(DISTINCT o.ORDERNUMBER) AS pedidos_enviados_2003
FROM order_data o

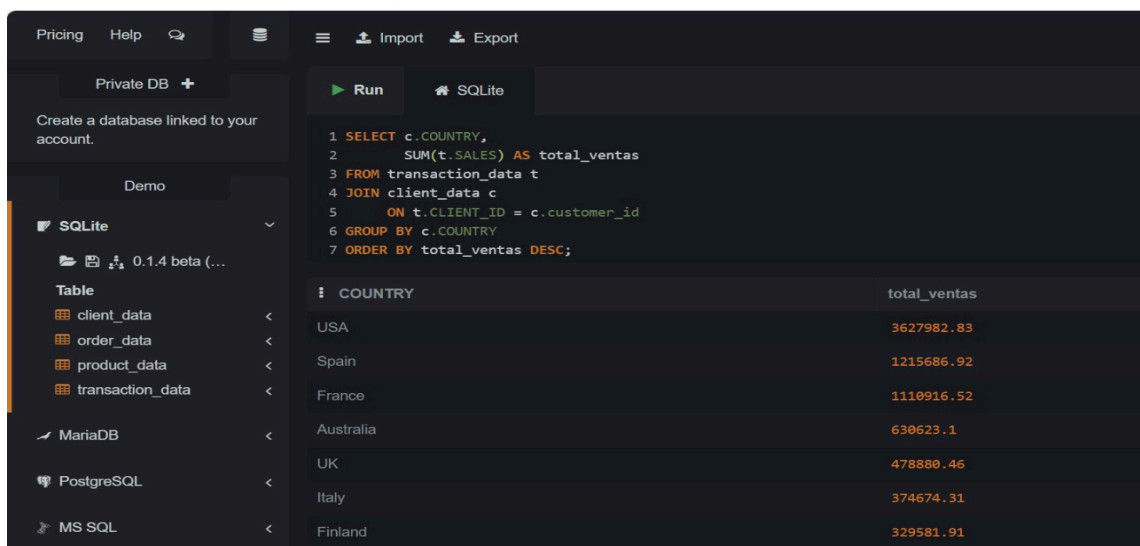
```

JOIN transaction_data t
  ON o.ORDERNUMBER = t.ORDERNUMBER
WHERE o.STATUS = 'Shipped'
AND t.YEAR_ID = 2003;

```



4. SELECT c.COUNTRY,
 SUM(t.SALES) AS TotalSales
 FROM transaction_data t
 JOIN client_data c
 ON t.CLIENT_ID = c.customer_id
 GROUP BY c.COUNTRY
 ORDER BY TotalSales DESC;



5. SELECT p.PRODUCTLINE,
 AVG(t.PRICEEACH) AS AvgPrice
 FROM transaction_data t
 JOIN product_data p
 ON t.PRODUCTCODE = p.PRODUCTCODE
 GROUP BY p.PRODUCTLINE

ORDER BY AvgPrice DESC;

The screenshot shows a database interface with a sidebar on the left containing a 'Demo' section with a 'SQLite' database. The main area displays a SQL query and its results. The query is:

```
1 SELECT p.PRODUCTLINE,
2       AVG(t.PRICEEACH) AS precio_promedio
3 FROM transaction_data t
4 JOIN product_data p
5     ON t.PRODUCTCODE = p.PRODUCTCODE
6 GROUP BY p.PRODUCTLINE;
```

The results are shown in a table with two columns: 'PRODUCTLINE' and 'precio_promedio'.

PRODUCTLINE	precio_promedio
Classic Cars	87.33578076525336
Motorcycles	82.99755287009063
Planes	81.74091503267974
Ships	83.85547008547009
Trains	75.65467532467532
Trucks and Buses	87.52794019933555
Vintage Cars	78.14820428336078

6. SELECT c.CUSTOMERNAME,
 SUM(t.SALES) AS TotalSales
 FROM transaction_data t
 JOIN client_data c
 ON t.CLIENT_ID = c.customer_id
 GROUP BY c.CUSTOMERNAME
 ORDER BY TotalSales DESC
 LIMIT 5;

The screenshot shows a database interface with a sidebar on the left containing a 'Demo' section with a 'SQLite' database. The main area displays a SQL query and its results. The query is:

```
1 SELECT c.CUSTOMERNAME,
2       SUM(t.SALES) AS total_ventas
3 FROM transaction_data t
4 JOIN client_data c
5     ON t.CLIENT_ID = c.customer_id
6 GROUP BY c.CUSTOMERNAME
7 ORDER BY total_ventas DESC
8 LIMIT 5;
```

The results are shown in a table with two columns: 'CUSTOMERNAME' and 'total_ventas'.

CUSTOMERNAME	total_ventas
Euro Shopping Channel	912294.11
Mini Gifts Distributors Ltd.	654858.06
Australian Collectors, Co.	200995.41
Muscle Machine Inc	197736.94
La Rochelle Gifts	180124.9

7. SELECT t.PRODUCTCODE,
 SUM(t.QUANTITYORDERED) AS TotalUnits

```
FROM transaction_data t
GROUP BY t.PRODUCTCODE
ORDER BY TotalUnits DESC
LIMIT 3;
```

The screenshot shows a database interface with a sidebar on the left containing a 'Demo' section with a 'SQLite' database. The main area displays a SQL query and its results. The query is:

```
1 SELECT t.PRODUCTCODE,
2       SUM(t.QUANTITYORDERED) AS total_unidades
3 FROM transaction_data t
4 GROUP BY t.PRODUCTCODE
5 ORDER BY total_unidades DESC
6 LIMIT 3;
```

The results are shown in a table with two columns: 'PRODUCTCODE' and 'total_unidades'.

PRODUCTCODE	total_unidades
S18_3232	1774
S24_3856	1052
S18_4600	1031

```
8. SELECT t.QTR_ID,
       SUM(t.SALES) AS TotalSales
FROM transaction_data t
WHERE t.YEAR_ID = 2003
GROUP BY t.QTR_ID
ORDER BY t.QTR_ID;
```

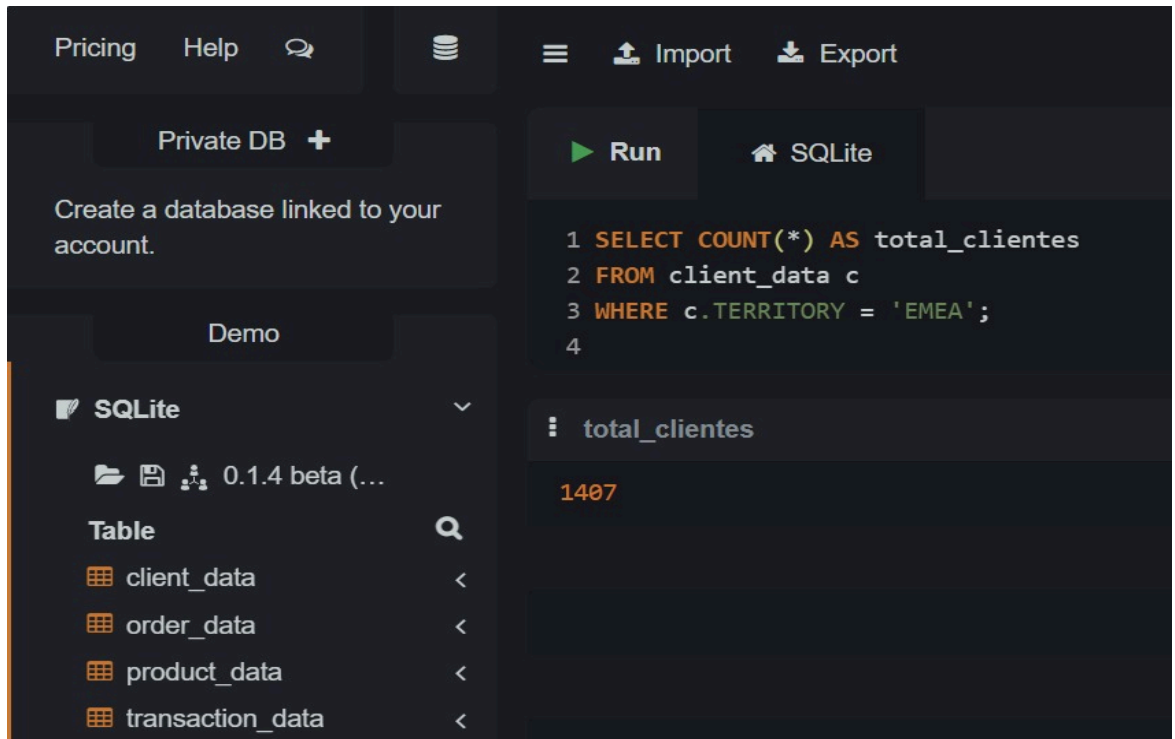
The screenshot shows the same database interface as the first image, but with a different SQL query and results. The query is:

```
1 SELECT t.QTR_ID,
2       SUM(t.SALES) AS total_ventas
3 FROM transaction_data t
4 WHERE t.YEAR_ID = 2003
5 GROUP BY t.QTR_ID
6 ORDER BY t.QTR_ID;
```

The results are shown in a table with two columns: 'QTR_ID' and 'total_ventas'.

QTR_ID	total_ventas
1	445094.69
2	562365.22
3	649514.54
4	1860005.09

9. `SELECT COUNT(*) AS TotalClientesEMEA
FROM client_data c
WHERE c.TERRITORY = 'EMEA';`



10. `SELECT o.ORDERNUMBER,
c.CUSTOMERNAME
FROM order_data o
JOIN transaction_data t
ON o.ORDERNUMBER = t.ORDERNUMBER
JOIN client_data c
ON t.CLIENT_ID = c.customer_id
WHERE o.STATUS = 'Cancelled';`

Pricing Help Import Export

Private DB +

Create a database linked to your account.

Demo

SQLite

0.1.4 beta (...)

Table

- client_data
- order_data
- product_data
- transaction_data

MariaDB

PostgreSQL

MS SQL

Run SQLite

```

5 ON o.ORDERNUMBER = t.ORDERNUMBER
6 JOIN client_data c
7 ON t.CLIENT_ID = c.customer_id
8 WHERE o.STATUS = 'Cancelled';

```

ORDERNUMBER	CUSTOMERNAME
10167	Scandinavian Gift Ideas
10167	Scandinavian Gift Ideas
10167	Scandinavian Gift Ideas
10167	Scandinavian Gift Ideas
10167	Scandinavian Gift Ideas
10167	Scandinavian Gift Ideas
10167	Scandinavian Gift Ideas
10167	Scandinavian Gift Ideas
10167	Scandinavian Gift Ideas

COMPLEJOS

- SELECT t.YEAR_ID,
 SUM(t.SALES) AS total_ventas
 FROM transaction_data t
 JOIN client_data c
 ON t.CLIENT_ID = c.customer_id
 WHERE c.COUNTRY = 'USA'
 AND t.QTR_ID = 1
 GROUP BY t.YEAR_ID
 ORDER BY t.YEAR_ID;

Pricing Help Import Export

Private DB +

Create a database linked to your account.

Demo

SQLite

0.1.4 beta (...)

Table

- client_data
- order_data
- product_data
- transaction_data

Run SQLite

```

6 WHERE c.COUNTRY = 'USA'
7 AND t.QTR_ID = 1
8 GROUP BY t.YEAR_ID
9 ORDER BY t.YEAR_ID;

```

YEAR_ID	total_ventas
2003	103274.02
2004	244774.7
2005	402041.42

```

2. SELECT DISTINCT c.CUSTOMERNAME
FROM transaction_data t
JOIN product_data p
    ON t.PRODUCTCODE = p.PRODUCTCODE
JOIN client_data c
    ON t.CLIENT_ID = c.customer_id
WHERE p.PRODUCTLINE = 'Classic Cars';

```

The screenshot shows a database client interface with a dark theme. The top bar includes 'Pricing', 'Help', and icons for 'Import' and 'Export'. Below this, there's a 'Private DB +' button and a 'Demo' button. The left sidebar lists database types: 'SQLite' (selected), 'MariaDB', 'PostgreSQL', and 'MS SQL'. Under 'SQLite', there's a 'Table' section with a search icon and a list of tables: 'client_data', 'order_data', 'product_data', and 'transaction_data'. The main area displays a SQL query with line numbers 5 through 8. The query is:


```

5 JOIN client_data c
6   ON t.CLIENT_ID = c.customer_id
7 WHERE p.PRODUCTLINE = 'Classic Cars';
8

```

 Below the query, the results are shown in a table with a single column header 'CUSTOMERNAME'. The results list several company names: 'Corrida Auto Replicas, Ltd', 'Dragon Souvenirs, Ltd.', 'Classic Legends Inc.', 'Australian Gift Network, Co', 'Classic Gift Ideas, Inc', 'Saveley & Henriot, Co.', 'Canadian Gift Exchange Network', 'West Coast Collectables Co.', and 'C...'.