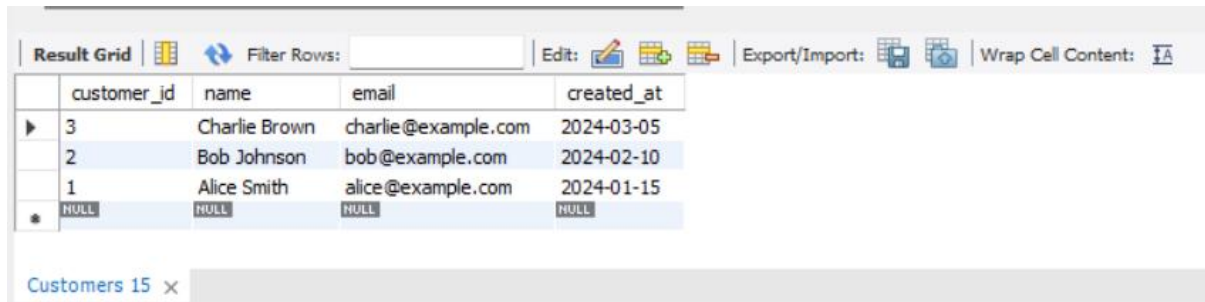


QUERIES OUTPUT

A. USE SELECT ,WHERE, ORDER BY

```
SELECT customer_id, name, email, created_at
FROM Customers
WHERE YEAR(created_at) = 2024
ORDER BY created_at DESC;
```



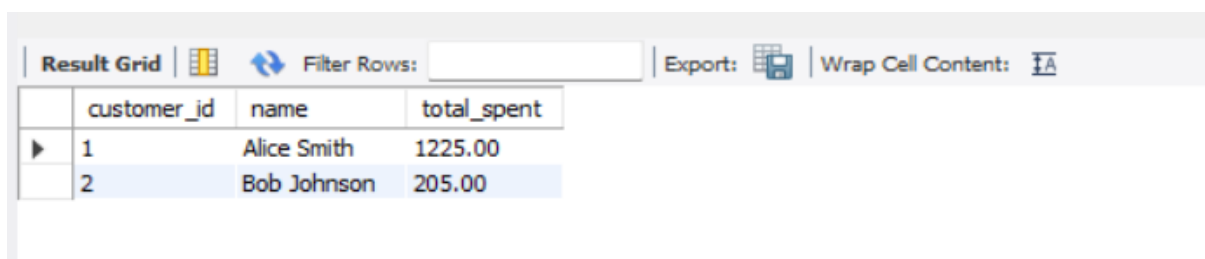
The screenshot shows a database interface with a toolbar at the top containing options like 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. Below the toolbar is a table with the following data:

	customer_id	name	email	created_at
▶	3	Charlie Brown	charlie@example.com	2024-03-05
	2	Bob Johnson	bob@example.com	2024-02-10
	1	Alice Smith	alice@example.com	2024-01-15
*	NULL	NULL	NULL	NULL

At the bottom, there is a tab labeled 'Customers 15' with a close button.

B. USE JOINS

```
SELECT c.customer_id, c.name, SUM(o.total_amount) AS total_spent
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name
ORDER BY total_spent DESC;
```

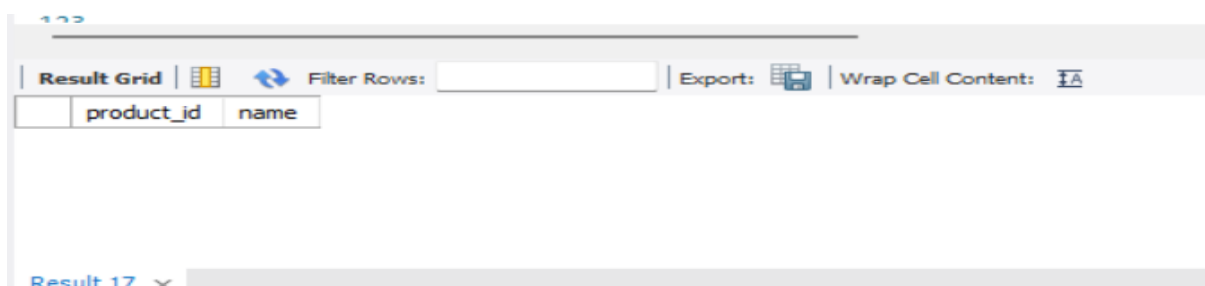


The screenshot shows a database interface with a toolbar at the top containing options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with the following data:

	customer_id	name	total_spent
▶	1	Alice Smith	1225.00
	2	Bob Johnson	205.00

C. USE LEFT JOIN

```
SELECT p.product_id, p.name
FROM Products p
LEFT JOIN OrderItems oi ON p.product_id = oi.product_id
WHERE oi.order_item_id IS NULL;
```



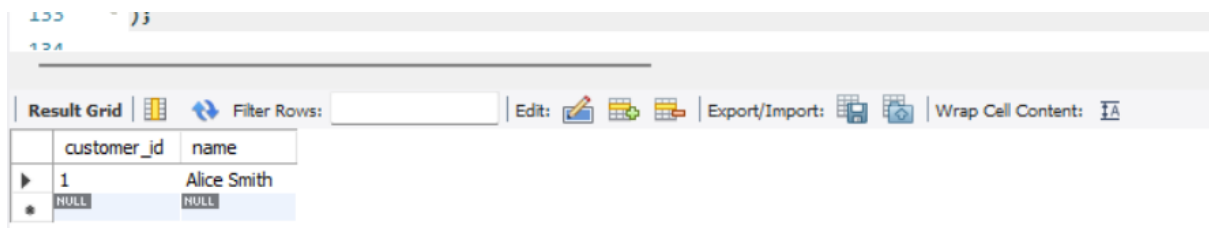
The screenshot shows a database interface with a toolbar at the top containing options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with the following data:

	product_id	name
--	------------	------

At the bottom, there is a tab labeled 'Result 17' with a close button.

D. USE SUM, AVG

```
SELECT customer_id, name
FROM Customers
WHERE customer_id IN (
    SELECT o.customer_id
    FROM Orders o
    GROUP BY o.customer_id
    HAVING SUM(o.total_amount) > (
        SELECT AVG(total_amount) FROM Orders
    )
);
```

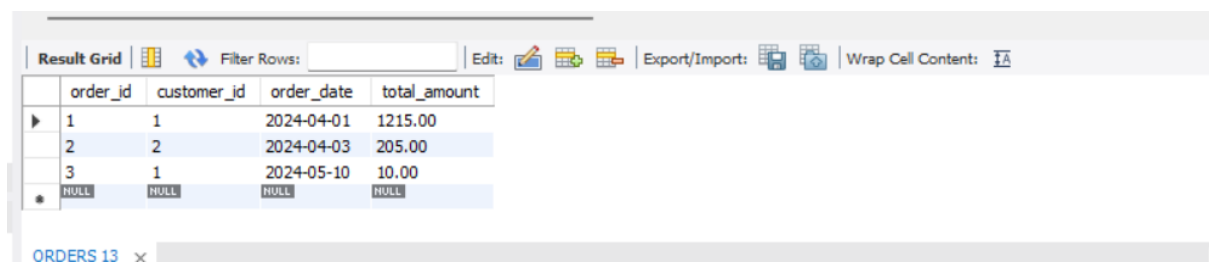


The screenshot shows a database query result grid. The grid has two columns: 'customer_id' and 'name'. The first row shows '1' for customer_id and 'Alice Smith' for name. There is a 'Filter Rows' field and a 'Wrap Cell Content' button at the top right of the grid.

customer_id	name
1	Alice Smith

E. View for Monthly Sales Summary

```
CREATE OR REPLACE VIEW Monthly_Sales_Summary AS
SELECT DATE_FORMAT(order_date, '%Y-%m') AS month,
       SUM(total_amount) AS revenue,
       COUNT(*) AS order_count
FROM Orders
GROUP BY month;
SELECT * FROM ORDERS;
```



The screenshot shows a database query result grid. The grid has four columns: 'order_id', 'customer_id', 'order_date', and 'total_amount'. The first three rows show order data: order_id 1 for customer_id 1 on 2024-04-01 with total_amount 1215.00; order_id 2 for customer_id 2 on 2024-04-03 with total_amount 205.00; and order_id 3 for customer_id 1 on 2024-05-10 with total_amount 10.00. The fourth row shows NULL values for all columns. There is a 'Filter Rows' field and a 'Wrap Cell Content' button at the top right of the grid.

order_id	customer_id	order_date	total_amount
1	1	2024-04-01	1215.00
2	2	2024-04-03	205.00
3	1	2024-05-10	10.00
NULL	NULL	NULL	NULL

F. Index for Optimization

```
CREATE INDEX idx_orders_customer_id ON Orders(customer_id);
SELECT * FROM ORDERS;
```