CREATE VIEW CustomerOrderSummary AS

SELECT c.customer\_name,

COUNT(o.order\_id) AS total\_orders,

SUM(o.total\_amount) AS total\_spent

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

WHERE o.order\_date >= '2024-01-01'

GROUP BY c.customer\_name

ORDER BY total\_spent DESC;

**Views for Abstraction and Security**

Views are powerful tools for both **abstraction** and **security**.

* **Abstraction**: Views simplify complex database schemas for users. By creating a view, you can hide the underlying complexity of multiple tables, joins, and calculations. Users can interact with a simplified, intuitive view that presents only the information they need, without having to understand the intricate relationships between the underlying tables. This improves usability and reduces the risk of errors from incorrect queries.
* **Security**: Views provide a robust layer of security by restricting user access to a subset of data. Instead of granting users direct access to sensitive tables, you can give them access only to a view. For example, you can create a view that shows an employee's name and department but hides their salary or social security number. The view can be configured to only show specific columns or rows, effectively controlling which data a user can see. This is a common practice for implementing role-based access control.