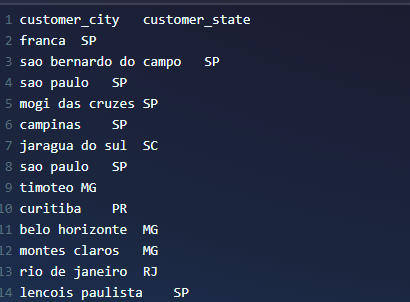
**SQL OUTPUTS SCREENSHORTS**

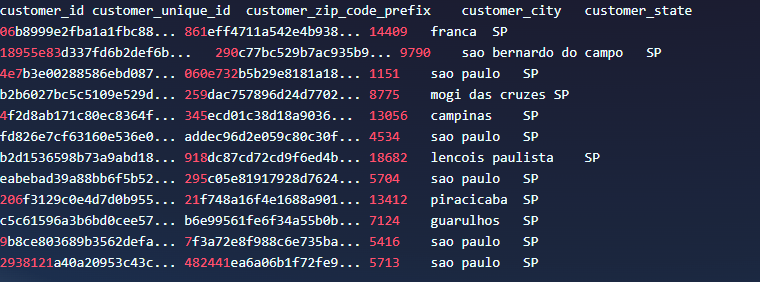
-- Selects the city and state for all customers

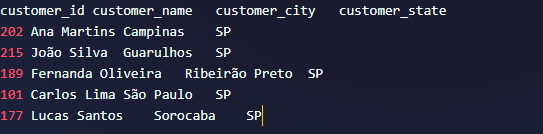
SELECT customer\_city, customer\_state

FROM customers;

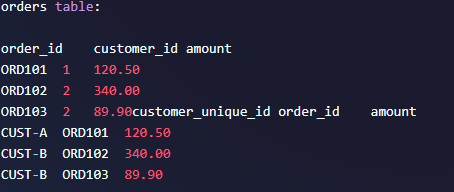
1. .**Use SELECT, WHERE, ORDER BY, GROUP BY**

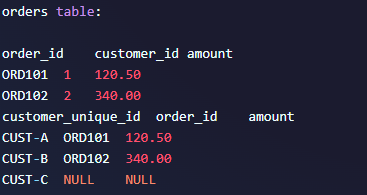


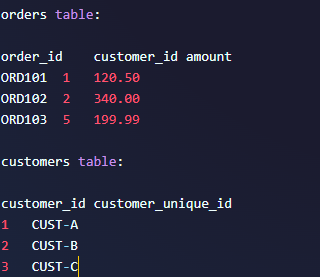


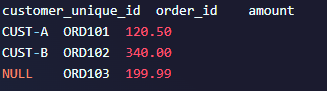


1. .**Use JOINS (INNER, LEFT, RIGHT)**

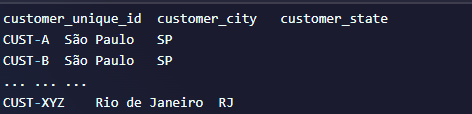


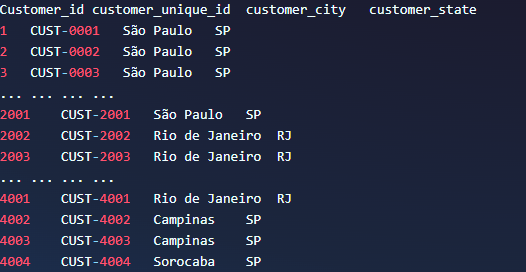




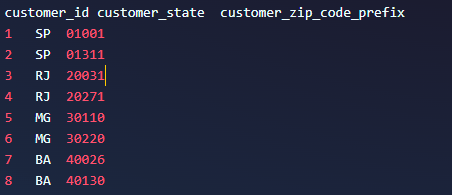


1. **Write subqueries:-**

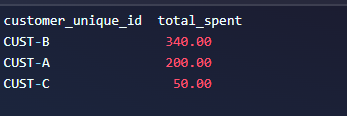




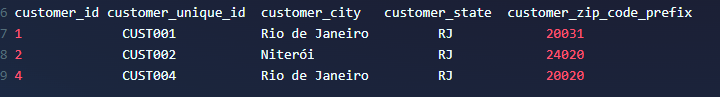
1. **Use aggregate functions (SUM, AVG):-**

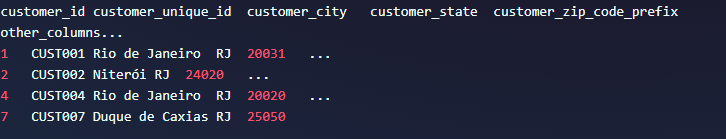
****

****

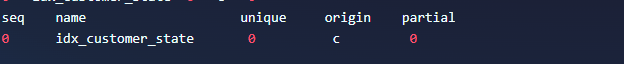
****

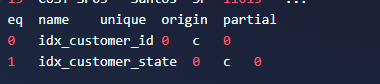
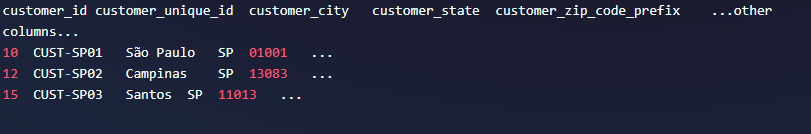
1. **Create views for analysis:-**

****

****

1. **Optimize queries with indexes:-**

****

****

1. **Difference between WHERE and HAVING?**
   * **WHERE** filters rows *before* grouping.
   * **HAVING** filters groups *after* GROUP BY.
2. **Different types of joins?**
   * INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN, CROSS JOIN.
3. **Calculate average revenue per user in SQL?**

sql

Copy-edit

SELECT AVG (total\_revenue) FROM (

SELECT user\_id, SUM (revenue) AS total\_revenue

FROM sales

GROUP BY user\_id

);

1. **What are sub queries?**
   * Queries nested inside another query to filter or calculate intermediate results.
2. **How do you optimize a SQL query?**
   * Use indexes, avoid SELECT \*, limit data, use joins efficiently, analyse query plans.
3. **What is a view in SQL?**
   * A virtual table representing the result of a stored query.
4. **How to handle NULL values in SQL?**
   * Use IS NULL or IS NOT NULL to filter, use functions like COALESCE () or IFNULL () to replace NULLs.