Capital One Slingshot

Cortex Code CLI testing prompt bank

28 prompts designed for SNOWFLAKE_SAMPLE_DATA.TPCDS_SF10TCL

Catalog discovery

1. What databases do I have access to? For each one, show me the schemas and a rough count of tables/views.
2. List all tables in TPCDS_SF10TCL with their row counts and column counts, sorted by row count descending.
3. Describe the star schema structure of TPCDS_SF10TCL. Identify the fact tables, their FK columns, and the dimension tables they reference.
4. Compare the column structure of STORE_SALES, CATALOG_SALES, and WEB_SALES. Which columns are shared across all three? Which are unique to each?
5. Identify which dimension tables are slowly changing (have multiple versions of the same entity). Show an example.
6. Write a query showing total net paid and total quantity by store for calendar year 2001.

SQL generation & optimization

1. Find the top 10 item categories by total net paid across STORE_SALES for Q4 of any year. Include the item class and average selling price.
2. Write a query showing each customer's total spend across all three sales channels in 2002. Rank by total cross-channel spend and return the top 25.
3. Calculate the return rate by item category for STORE_SALES joined to STORE_RETURNS. Flag categories with a return rate above 10%.
4. Write a year-over-year revenue comparison for STORE_SALES using LAG(), comparing 2001 and 2002 by month.
5. Compare average transaction value with and without promotions for the top 10 item categories.
6. The store revenue query is running slowly. Analyze it and suggest clustering keys, pruning improvements, or rewrite strategies.

Capital One Slingshot

Cortex Code testing prompt bank

28 prompts designed for SNOWFLAKE_SAMPLE_DATA.TPCDS_SF10TCL

Streamlit app builder

1. Generate a Streamlit app showing monthly revenue trend as a line chart. Add a dropdown to filter by store name.
2. Build a Streamlit app with a year selector and three KPI cards showing total net paid for store, catalog, and web channels. Add a stacked bar chart by quarter.
3. Create a Streamlit app where a user selects an item category and sees a bar chart of the top 10 items by total SS_NET_PAID.
4. Build a Streamlit scatter plot of return rate vs avg selling price by category. Include a slider for minimum items sold.
5. Generate a customer lookup app: text input for last name, results table with state and lifetime spend, bar chart of monthly spend history.

FinOps & cost analysis

1. What are top 5 virtual warehouses by total credit consumption over the past 30 days? Show average credits query.
2. Compare the cost of running a full STORE_SALES scan on X-Small vs Medium. Estimate credits and recommend the right warehouse size.
3. Find the 10 most expensive queries run against TPCDS_SF10TCL in the past 7 days.
4. Identify queries against STORE_SALES that performed full table scans in the last 14 days.
5. Build a cost attribution report: for each user, show total credits consumed, average query duration, and number of full-table scans.

dbt model generation

1. Generate stg_store_sales.sql that selects from STORE_SALES, renames columns to snake_case, and casts SS_SOLD_DATE_SK to a proper date using DATE_DIM.
2. Create mart_store_revenue.sql that joins stg_store_sales to ITEM and STORE dimensions, exposing monthly revenue by category and store.
3. Rewrite mart_store_revenue for incremental materialisation with a watermark on sold date.
4. Generate mart_return_rate.sql and add dbt tests for not_null and accepted_values on the channel column.
5. Write mart_customer_ltv.sql calculating LTV across all three channels, using ref() to reference staging models.
6. Generate a complete dbt project: staging models for each fact table, intermediate models for common joins, two mart models. Include schema.yml with column descriptions.