**📄 Report Structure**

**Title Page**

* **Project Title**: Advanced Analysis of Sales / E‑commerce Dataset
* **Intern Name**, **Internship Duration**
* **CODTECH Logo** (add at top-center)

**1. Introduction**

* Purpose: uncover trends & patterns using SQL.
* Dataset: e‑commerce/sales sample (orders, customers, products).
* Tools: PostgreSQL (or MySQL), pgAdmin / DBeaver.
* Objectives: demonstrate Window Functions, CTEs, Subqueries.

**2. Dataset Overview**

Describe data tables:

* **orders**: order\_id, customer\_id, order\_date, total\_amount
* **order\_items**: order\_id, product\_id, quantity, price
* **customers**: customer\_id, name, region

**🔍 Analysis Sections**

**A. Window Functions (3+ examples)**

1. **Running Total of Daily Sales**

sql

CopyEdit

SELECT

order\_date,

SUM(total\_amount) OVER (ORDER BY order\_date) AS running\_total

FROM orders

ORDER BY order\_date;

* Uses SUM(...) OVER (ORDER BY …) for cumulative total [en.wikipedia.org](https://en.wikipedia.org/wiki/Window_function_%28SQL%29?utm_source=chatgpt.com)[github.com](https://github.com/aniass/SQL-Sales-Analysis?utm_source=chatgpt.com)[en.wikipedia.org](https://en.wikipedia.org/wiki/Hierarchical_and_recursive_queries_in_SQL?utm_source=chatgpt.com)[reddit.com](https://www.reddit.com/r/SQL/comments/1hkj47b/windows_subqueries_and_ctes/?utm_source=chatgpt.com)[medium.com+4crunchydata.com+4geeksforgeeks.org+4](https://www.crunchydata.com/developers/playground/postgres-window-functions-for-data-analysis?utm_source=chatgpt.com)[datacamp.com](https://www.datacamp.com/cheat-sheet/sql-window-functions-cheat-sheet?utm_source=chatgpt.com)[en.wikipedia.org](https://en.wikipedia.org/wiki/Correlated_subquery?utm_source=chatgpt.com)
* *Interpretation*: shows how revenue has grown day by day.

1. **Per-Customer Ranking by Lifetime Spend**

sql

CopyEdit

SELECT

customer\_id,

SUM(total\_amount) AS total\_spent,

RANK() OVER (ORDER BY SUM(total\_amount) DESC) AS spend\_rank

FROM orders

GROUP BY customer\_id;

* Uses RANK() OVER (ORDER BY SUM(...)) to rank customers by spend
* *Insight*: identify top spenders.

1. **Month-over-Month Sales Growth**

sql

CopyEdit

WITH Monthly AS (

SELECT

date\_trunc('month', order\_date) AS month,

SUM(total\_amount) AS total

FROM orders

GROUP BY month

)

SELECT

month,

total,

LAG(total,1) OVER (ORDER BY month) AS prev\_total,

ROUND((total - LAG(total,1) OVER (ORDER BY month)) \* 100.0 / LAG(total,1) OVER (ORDER BY month),2) AS pct\_change

FROM Monthly;

* Combines CTE with LAG() for MoM change [crunchydata.com+4crunchydata.com+4ai2sql.io+4](https://www.crunchydata.com/developers/playground/ctes-and-window-functions?utm_source=chatgpt.com)
* *Outcome*: calculates monthly growth %.

**B. CTE-Based Queries (2+ examples)**

1. **Sales per Customer & Regional Average**

sql

CopyEdit

WITH CustSales AS (

SELECT customer\_id, SUM(total\_amount) AS total\_spent

FROM orders

GROUP BY customer\_id

)

SELECT

cs.customer\_id,

cs.total\_spent,

AVG(cs.total\_spent) OVER () AS avg\_spent,

cs.total\_spent - AVG(cs.total\_spent) OVER () AS diff\_from\_avg

FROM CustSales cs;

* CTE encapsulates aggregated data; window function calculates average [crunchydata.com+2crunchydata.com+2ai2sql.io+2](https://www.crunchydata.com/developers/playground/ctes-and-window-functions?utm_source=chatgpt.com)[datacamp.com+8en.wikipedia.org+8datalemur.com+8](https://en.wikipedia.org/wiki/Window_function_%28SQL%29?utm_source=chatgpt.com)[towardsai.net+13medium.com+13crunchydata.com+13](https://medium.com/%40jshweta2592/understanding-cte-and-window-functions-in-sql-a-deep-dive-with-examples-and-benefits-5a64438eeacb?utm_source=chatgpt.com)
* *Insight*: compares individual to average.

1. **Top Products by Quantity and Revenue**

sql

CopyEdit

WITH ProdStats AS (

SELECT

oi.product\_id,

SUM(oi.quantity) AS total\_qty,

SUM(oi.quantity \* oi.price) AS total\_revenue

FROM order\_items oi

GROUP BY oi.product\_id

)

SELECT

product\_id,

total\_qty,

total\_revenue,

RANK() OVER (ORDER BY total\_revenue DESC) AS rev\_rank

FROM ProdStats

WHERE total\_qty > 100;

* CTE simplifies raw stats; filter & rank via window [towardsai.net+3learnsql.com+3datacamp.com+3](https://learnsql.com/blog/practical-sql-cte-examples/?utm_source=chatgpt.com)[ai2sql.io+1medium.com+1](https://ai2sql.io/mastering-sql-window-functions-for-analytics?utm_source=chatgpt.com)
* *Findings*: high-volume, high-revenue products.

**C. Subqueries (2+ examples)**

1. **Customers Above-Avg Spend**

sql

CopyEdit

SELECT

customer\_id,

SUM(total\_amount) AS total\_spent

FROM orders

GROUP BY customer\_id

HAVING SUM(total\_amount) > (

SELECT AVG(total\_amount \* cnt) FROM (

SELECT customer\_id, SUM(total\_amount) \* COUNT(\*) AS total\_amount

FROM orders

GROUP BY customer\_id

) AS subs

);

* Subquery computes average to filter above‑average customers [en.wikipedia.org+1reddit.com+1](https://en.wikipedia.org/wiki/Correlated_subquery?utm_source=chatgpt.com)[github.com+11towardsai.net+11datalemur.com+11](https://towardsai.net/p/data-analysis/advanced-sql-for-data-analysis-part-1-subqueries-and-cte?utm_source=chatgpt.com)
* *Result*: list of customers spending above peer average.

1. **Correlated Subquery: Above-Department Avg Sales**

sql

CopyEdit

SELECT

s.salesperson\_id,

s.total\_sales

FROM (

SELECT salesperson\_id, SUM(sale\_amount) AS total\_sales

FROM sales

GROUP BY salesperson\_id

) s

WHERE s.total\_sales > (

SELECT AVG(total\_sales) FROM (

SELECT salesperson\_id, SUM(sale\_amount) AS total\_sales

FROM sales

GROUP BY salesperson\_id

) AS t

);

* Correlated logic inside WHERE clause [bigtechinterviews.com+11en.wikipedia.org+11medium.com+11](https://en.wikipedia.org/wiki/Correlated_subquery?utm_source=chatgpt.com)[medium.com+12chayansraj.medium.com+12en.wikipedia.org+12](https://chayansraj.medium.com/sql-window-functions-and-ctes-13105a3bd710?utm_source=chatgpt.com)
* *Insight*: top-performing sales reps.

**📊 Output Tables & Screenshots**

Include screenshots (or small tables) of query results for each section.

**✅ Key Findings (Bullet Summary)**

* Daily sales growth shown via running totals.
* Top customers & deviations from average identified.
* Certain products outperform in both quantity and revenue.
* Month-over-month revenue trends detected (growth or decline).
* A subset of customers spends significantly above peer average.

Bottom of Form