UNIVERSITY ASSIGNMENT BRIEF (SQL Server + Olist Dataset)

Course: Modern Data Engineering Practices

Module: Analytics Engineering with dbt

Tooling: dbt Core, SQL Server, Git

Assignment ID: AE301-SQL-OLIST

Assessment Type: Practical Project

Estimated Time Required: 12–15 hours

# Assignment Title

Building an End-to-End Analytics Warehouse Using dbt & SQL Server (Olist E-commerce Data)

# Objective

Simulate the responsibilities of an Analytics Engineer by using dbt Core with SQL Server to transform the Olist Brazilian E-Commerce dataset into reliable, documented, and tested analytics-ready datasets. You will practice data modeling, testing and documentation, Jinja/macros for reusable logic, and version control with Git.

# Background

You’ve been hired as a Junior Analytics Engineer at dbtMart, a fictional e-commerce company modeled on the Olist marketplace. The leadership team wants a centralized analytics warehouse to answer questions across Marketing, Finance, Product, Growth, and Executive Leadership. Your job is to transform raw Olist data into clean, analytics-ready marts that deliver insights.

# Stakeholders & Business Questions

## Marketing Manager

- Who are our most valuable customers?

- How many are repeat vs one-time buyers?

- What is the average time between first and repeat purchases?

- Which customers are “At Risk” of churn?

- Expected Models: customer\_lifetime\_value, customer\_orders\_summary, customer\_segmentation

## Finance Team

- What are daily, weekly, and monthly sales?

- What is the average order value?

- Which seasons or months show spikes in revenue?

- How do cancellations and refunds impact net sales?

- Expected Models: daily\_sales, sales\_summary

## Product Manager

- What are the top-selling products and categories?

- Which products have high return or cancellation rates?

- Are there items with strong visibility but weak conversion?

- Which categories drive the highest contribution to revenue?

- Expected Models: product\_performance, product\_conversion (optional)

## Growth Team (Optional Advanced)

- How many customers move from product view → cart → purchase?

- Where are the biggest funnel drop-offs?

- How do first-time vs repeat customers behave differently?

- Expected Models: user\_engagement\_funnel

## Executive Leadership

- What share of revenue comes from the top 10% of customers?

- Are we growing faster in new vs returning customers?

- What’s our retention vs acquisition balance?

- How do product categories contribute to total revenue mix?

- Expected Models: executive\_dashboard\_metrics

# Tasks & Requirements

- Project Setup: Initialize dbt with SQL Server and configure profiles.yml

- Source Layer: Declare raw Olist tables as dbt sources

- Staging Models: One per raw table (stg\_)

- Intermediate Models: At least 2 (int\_) handling joins or aggregations

- Marts Models: At least 3 answering stakeholder needs

- Testing: Apply generic tests (unique, not null) and one custom test

- Documentation: Add descriptions and generate dbt docs with lineage

- Macros & Jinja: Create at least one reusable macro

- Seeds (Optional): Load a CSV and use it as a seed table

- Version Control: Use GitHub for commits and branches

- Deployment Simulation: Separate dev and prod schemas

# Deliverables

- GitHub repo or zipped dbt project folder

- README.md explaining models and dataset usage

- Optional Loom walkthrough video

# Assessment Checklist & Grading Rubric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Area | Task | Points | Completed? | Notes / Links |
| Project Setup | dbt project initialized with SQL Server | 5 pts | ☐ |  |
| Project Setup | profiles.yml properly configured | 5 pts | ☐ |  |
| Source Layer | All raw Olist tables declared as sources | 5 pts | ☐ |  |
| Source Layer | Source freshness config added (optional) | 2 pts | ☐ |  |
| Staging Models | One staging model per raw table (stg\_) | 10 pts | ☐ |  |
| Staging Models | Correct naming conventions used | 2 pts | ☐ |  |
| Intermediate Models | At least 2 int\_ models created | 5 pts | ☐ |  |
| Intermediate Models | Joins/aggregations applied correctly | 5 pts | ☐ |  |
| Marts Models | At least 3 final models answering stakeholder needs | 10 pts | ☐ |  |
| Marts Models | Business logic aligns to stakeholder questions | 5 pts | ☐ |  |
| Testing | Generic tests (unique, not null) applied | 5 pts | ☐ |  |
| Testing | One custom/singular test written | 5 pts | ☐ |  |
| Documentation | All models documented with descriptions | 5 pts | ☐ |  |
| Documentation | dbt docs generate + lineage graph works | 2 pts | ☐ |  |
| Jinja & Macros | At least one macro created and used | 5 pts | ☐ |  |
| Jinja & Macros | Jinja used in model logic | 3 pts | ☐ |  |
| Seeds (Optional) | At least one seed CSV imported & used | 3 pts | ☐ |  |
| Version Control | Project under GitHub/Git | 5 pts | ☐ |  |
| Version Control | Commit messages are meaningful | 2 pts | ☐ |  |
| Deployment Simulation | Use separate dev & prod schemas in SQL Server | 3 pts | ☐ |  |
| Deployment Simulation | Simulate PR via branch workflow | 2 pts | ☐ |  |
| Bonus | Snapshots used | 3 pts | ☐ |  |
| Bonus | Exposures defined (e.g., dashboards) | 2 pts | ☐ |  |
| Bonus | Loom walkthrough provided | 3 pts | ☐ |  |

# Grading Criteria

A (Distinction): 85 – 100 pts

B (Merit): 70 – 84 pts

C (Pass): 50 – 69 pts

D (Fail): < 50 pts