7/28/2025

Unit Testing

Online Retail Sales Analysis

Team 5

Jyoti Maske

Hiral Parekh

Neel Patel

Yash Khamar

| **ID** | **Type** | **Name** | **Description / Purpose** | **Instructions** | **Expected Results** | **Actual Results** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC01 | Functional | Data File Ingestion | Verify Kaggle CSV is correctly read and essential columns are present | Run Python ETL script. Check if data.csv is read without errors and columns like CustomerID, InvoiceNo, etc. are present. | File is read with correct schema and rows > 0 | File read successfully | Success |
| TC02 | Validation | Drop Null and Invalid Rows | Ensure rows with missing CustomerID or Quantity ≤ 0 are removed | Inspect clean\_orders.csv, clean\_customers.csv, etc. Ensure rows with missing critical fields are gone. | Files contain no null in required fields, only valid values | No null or invalid values found | Success |
| TC03 | Functional | Product Enrichment via API | Verify products are enriched with brand, category, rating, image | Check clean\_products.csv for columns: Category, Brand, Rating, ImageURL. Validate source from FakeStoreAPI. | All products enriched with new metadata columns | Metadata added correctly | Success |
| TC04 | Validation | Brand Table Generation | Verify unique brands are inserted into Brands table and referenced in Products | Open SQL Server, run SELECT \* FROM Brands. Ensure all brand names are unique and referenced in Products table via BrandID. | Unique brands with IDs exist; FK constraint working | BrandIDs mapped properly | Success |
| TC05 | Integrity | Foreign Key Checks | Ensure foreign keys (CustomerID, ProductID, BrandID) are valid across tables | Try inserting invalid ProductID in OrderDetails. Should fail. Validate SQL constraints. | Invalid foreign key insertion fails; existing keys match | Constraints working as expected | Success |
| TC06 | Functional | Orders and OrderDetails Sync | Validate OrderDetails rows reference only valid OrderIDs from Orders | Use SQL join: SELECT \* FROM OrderDetails od LEFT JOIN Orders o ON od.OrderID = o.OrderID WHERE o.OrderID IS NULL | No unmatched OrderID in OrderDetails | No unmatched found | Success |
| TC07 | Visualization | Top Countries by Revenue | Create Tableau chart to show revenue by Country | Use Tableau → connect to SQL Server → build bar chart (Country vs Revenue). Check if countries rank as expected. | Top countries like UK, Germany should show based on total sales | Chart reflects expected results | Success |
| TC08 | Visualization | Brand Performance Analysis | Visualize total quantity sold per Brand to identify leading brands | Use Tableau or Power BI → join OrderDetails with Products and Brands. Aggregate Quantity per Brand. | Accurate sales distribution by Brand | Brands ranked by quantity sold | Success |
| TC09 | Performance | Data Load Time | Measure time to run full load\_csv\_to\_sqlserver.py pipeline | Use stopwatch or script logging to measure end-to-end execution duration. Target under 60 seconds for full load on dev machine. | Load time under 60 seconds | 48 seconds | Success |
| TC10 | Accuracy | Duplicate Prevention | Ensure Customers, Orders, and Products tables don't have duplicate keys | Run: SELECT CustomerID, COUNT(\*) FROM Customers GROUP BY CustomerID HAVING COUNT(\*) > 1 (and similar for others) | All primary keys are unique; duplicate inserts fail | No duplicates detected | Success |
| TC11 | Validation | Rating Value Range Check | Ensure all product ratings fall between 0 and 5 | SQL: SELECT \* FROM Products WHERE Rating < 0 OR Rating > 5 | No product rating outside 0–5 | All ratings valid | Success |
| TC12 | Functional | Dashboard Tile Accuracy | Verify KPIs like Total Sales, Total Orders, Total Customers show correct values | Compare Tableau dashboard tiles with SQL aggregates. For example: SELECT COUNT(DISTINCT OrderID) for Total Orders. | KPI tiles match backend data | KPIs accurate | Success |
| TC13 | Validation | Referential Integrity - Product | Ensure every ProductID in OrderDetails exists in Products table | SQL: SELECT \* FROM OrderDetails WHERE ProductID NOT IN (SELECT ProductID FROM Products) | All ProductIDs in OrderDetails should exist in Products | No mismatch found | Success |
| TC14 | Automation | Schedule Job Check | Validate scheduled ETL batch job runs every day and loads new data | Check Task Scheduler or .bat script logs. Ensure new data from API is appended or updated daily. | Daily job runs, logs created | Logs found in scheduled folder | Success |