

ID	Feature	Description	SQL Query	Expected Result	Actual Result	Status	Severity	Priority
TC_SQL_01	Customers	Verify that every order belongs to an existing customer.	SELECT o.OrderID, o.CustomerID FROM Orders o LEFT JOIN Customers c ON o.CustomerID = c.CustomerID WHERE c.CustomerID IS NULL;	Query returns 0 rows (no orphaned orders).	Query returned 0 rows - no orphaned orders found.	Pass	High	High
TC_SQL_02	Orders	Verify that every order was handled by an existing employee.	SELECT o.OrderID, o.EmployeeID FROM Orders o LEFT JOIN Employees e ON o.EmployeeID = e.EmployeeID WHERE e.EmployeeID IS NULL;	0 invalid employees.	Query returned = invalid employees (all Employee IDs valid)	Pass	High	High
TC_SQL_03	Products	Verify that discontinued products are not used in new orders.	SELECT od.OrderID, od.ProductID FROM [Order Details] od JOIN Products p ON od.ProductID = p.ProductID WHERE p.Discontinued = 1;	Query returns 0 rows.	Query returned 0 rows - no discontinued product appears in Order Details.	Pass	Medium	Medium
TC_SQL_04	Products	Verify that every product belongs to an existing category.	SELECT p.ProductID, p.CategoryID FROM Products p LEFT JOIN Categories c ON p.CategoryID = c.CategoryID WHERE c.CategoryID IS NULL;	No missing category references.	Query returned 0 missing category references.	Pass	High	High
TC_SQL_05	Order Details	Verify that every order contains at least one order detail line.	SELECT o.OrderID FROM Orders o LEFT JOIN [Order Details] od ON o.OrderID = od.OrderID WHERE od.OrderID IS NULL;	0 empty orders.	Query returned 0 empty orders (all orders have details).	Pass	High	High
TC_SQL_06	Employees	Verify that employee ReportsTo links point to valid managers.	SELECT e.EmployeeID, e.ReportsTo FROM Employees e LEFT JOIN Employees m ON e.ReportsTo = m.EmployeeID WHERE e.ReportsTo IS NOT NULL AND m.EmployeeID IS NULL;	No invalid manager references.	Query returned 0 invalid manager references - all managers exist.	Pass	Medium	Medium
TC_SQL_07	Product Pricing	Verify that all product unit prices are positive values.	SELECT ProductID, UnitPrice FROM Products WHERE UnitPrice <= 0;	0 rows with invalid prices.	Query returned 0 rows - no products with invalid (<=0) prices.	Pass	High	High
TC_SQL_08	Customer Data	Verify that Customer CompanyName field is never NULL	SQL Query: SELECT CustomerID FROM Customers WHERE CompanyName IS NULL;	No null CompanyName values.	Query returned 0 null CompanyName values.	Pass	High	High

TC_SQL_09	Orders	Verify that no order has a ShipDate earlier than OrderDate.	<pre>SELECT OrderID, OrderDate, ShippedDate FROM Orders WHERE ShippedDate < OrderDate;</pre>	0 invalid timeline records.	Query returned 0 invalid timeline records.	Pass	Medium	Medium
TC_SQL_10	Categories	Validate that all categories have at least one product.	<pre>SELECT c.CategoryID, c.CategoryName FROM Categories c LEFT JOIN Products p ON c.CategoryID = p.CategoryID WHERE p.CategoryID IS NULL;</pre>	Query returns 0 empty categories.	Query returned 0 empty categories - all categories have products.	Pass	Low	Low
TC_SQL_11	Orders	Verify Freight field contains >= 0 values.	<pre>SELECT OrderID, Freight FROM Orders WHERE Freight < 0;</pre>	No negative freight values.	Query returned 0 rows - no negative Freight values.	Pass	Medium	Medium
TC_SQL_12	Products	Validate ProductName is not empty or null.	<pre>SELECT ProductID FROM Products WHERE ProductName IS NULL OR ProductName = '';</pre>	0 invalid product names.	Query returned 0 invalid product names.	Pass	Medium	Medium
TC_SQL_13	Order Details	Verify Discount values are between 0 and 1.	<pre>SELECT OrderID, ProductID, Discount FROM [Order Details] WHERE Discount < 0 OR Discount > 1;</pre>	All discounts within valid range.	Query returned 0 out-of-range discount values.	Pass	Low	Low
TC_SQL_14	Employees	Verify BirthDate is not null and before HireDate.	<pre>SELECT EmployeeID, BirthDate, HireDate FROM Employees WHERE BirthDate IS NULL OR BirthDate >= HireDate;</pre>	All employee dates follow logical order.	Query returned 0 invalid date combinations.	Pass	Low	Medium
TC_SQL_15	Order Details	Verify each order subtotal (quantity × unit price - discount) is positive.	<pre>SELECT OrderID, ProductID, (UnitPrice * Quantity * (1 - Discount)) AS Subtotal FROM [Order Details] HAVING Subtotal <= 0;</pre>	All subtotals must be > 0.	Query returned 0 invalid subtotals - all > 0	Pass	High	Medium