```sql
-- Duplicate detection
SELECT Invoice_ID, COUNT(*) AS DuplicateCount
FROM Transactions GROUP BY Invoice_ID HAVING COUNT(*) > 1;

SELECT Email, COUNT(*) AS DupCount
FROM Customers GROUP BY Email HAVING COUNT(*) > 1;

SELECT Product_ID, COUNT(*) AS DupCount
FROM Products GROUP BY Product_ID HAVING COUNT(*) > 1;

-- Missing values
SELECT COUNT(*) AS TotalRows,
      SUM(CASE WHEN Name IS NULL THEN 1 ELSE 0 END) AS Missing_Name,
      SUM(CASE WHEN Email IS NULL THEN 1 ELSE 0 END) AS Missing_Email,
      SUM(CASE WHEN Country IS NULL THEN 1 ELSE 0 END) AS Missing_Country
FROM Customers;

SELECT COUNT(*) AS TotalRows,
      SUM(CASE WHEN Product_ID IS NULL THEN 1 ELSE 0 END) AS Missing_ProductID,
      SUM(CASE WHEN Store_ID IS NULL THEN 1 ELSE 0 END) AS Missing_StoreID,
      SUM(CASE WHEN Employee_ID IS NULL THEN 1 ELSE 0 END) AS Missing_EmployeeID,
      SUM(CASE WHEN Quantity IS NULL THEN 1 ELSE 0 END) AS Missing_Quantity,
      SUM(CASE WHEN Unit_Price IS NULL THEN 1 ELSE 0 END) AS Missing_UnitPrice
FROM Transactions;

-- Referential integrity
SELECT COUNT(*) AS Invalid_Customers
FROM Transactions t LEFT JOIN Customers c ON t.Customer_ID = c.Customer_ID
WHERE c.Customer_ID IS NULL;

SELECT COUNT(*) AS Invalid_Products
FROM Transactions t LEFT JOIN Products p ON t.Product_ID = p.Product_ID
WHERE p.Product_ID IS NULL;

SELECT COUNT(*) AS Invalid_Stores
FROM Transactions t LEFT JOIN Stores s ON t.Store_ID = s.Store_ID
WHERE s.Store_ID IS NULL;

SELECT COUNT(*) AS Invalid_Employees
FROM Transactions t LEFT JOIN Employees e ON t.Employee_ID = e.Employee_ID
WHERE e.Employee_ID IS NULL;

-- Numeric ranges
SELECT COUNT(*) AS Invalid_UnitPrice FROM Transactions WHERE Unit_Price <= 0;
SELECT COUNT(*) AS Invalid_Quantity FROM Transactions WHERE Quantity <= 0;
SELECT COUNT(*) AS Invalid_Totals FROM Transactions WHERE Line_Total <= 0 OR Invoice_Total <= 0;

-- Customer DOB validation
SELECT COUNT(*) AS Invalid_DOB
FROM Customers WHERE Date_Of_Birth < '1900-01-01' OR Date_Of_Birth > GETDATE();
```