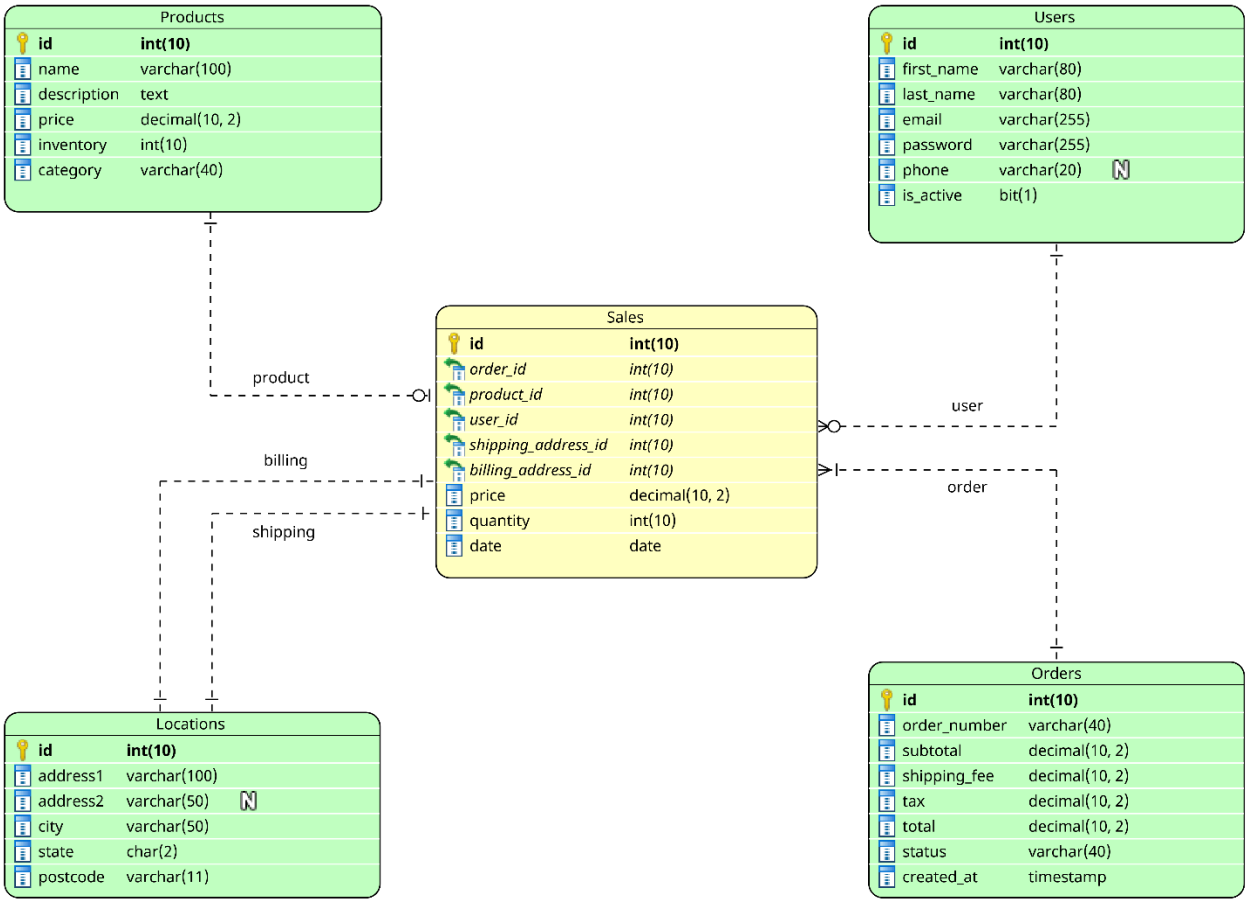
Monitor data quality with SQL

As you've learned, it is important to monitor data quality. By monitoring your data, you become aware of any problems that may occur within the ETL pipeline and data warehouse design. This can help you address problems as early as possible and avoid future problems.

In this reading, you'll follow a fictional scenario where a BI engineer performs quality testing on their pipeline and suggests SQL queries that one could use for each step of testing.

The scenario

At Francisco's Electronics, an electronics manufacturing company, a BI engineer named Sage designed a data warehouse for analytics and reporting. After the ETL process design, Sage created a diagram of the schema.



The diagram of the schema of the **sales_warehouse** database contains different symbols and connectors that represent two important pieces of information: the major tables within the system and the relationships among these tables.

The **sales_warehouse** database schema contains five tables:

- Sales
- Products
- Users
- Locations
- Orders

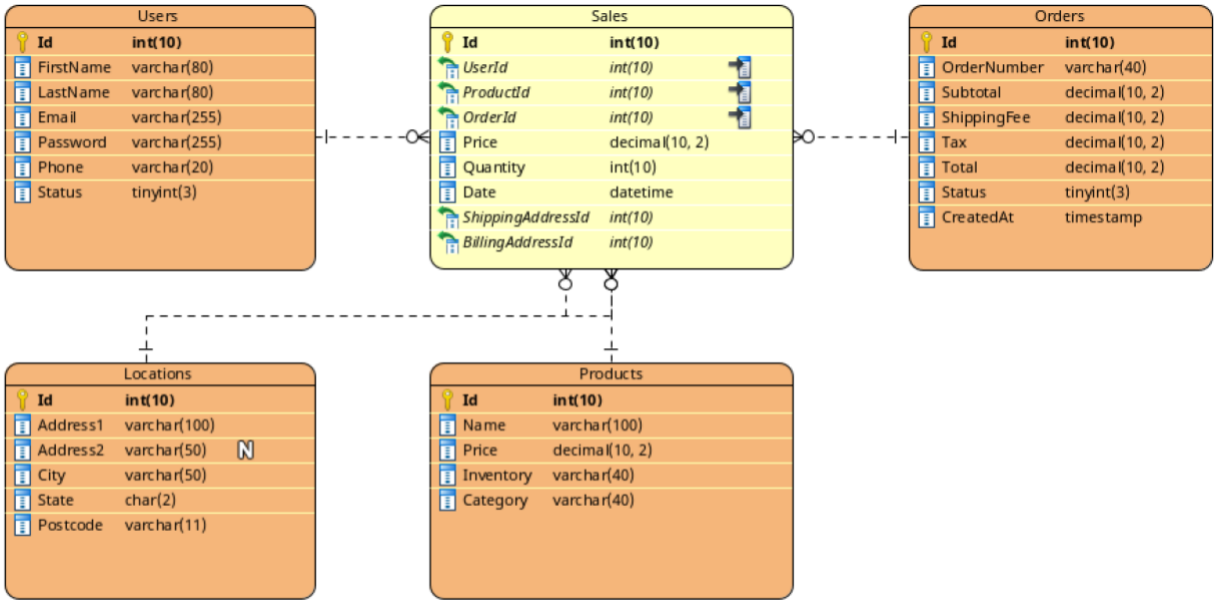
These tables are connected via keys. The tables contain five to eight columns (or attributes) ranging in data type. The data types include varchar or char (or character), integer, decimal, date, text (or string), timestamp, and bit.

The foreign keys in the Sales table link to each of the other tables:

- The “product_id” foreign key links to the Products table
- The “user_id” foreign key links to the Users table
- The “order_id” foreign key links to the Orders table
- The “shipping_address_id” and “billing_address_id” foreign keys link to the Locations table

After Sage made the **sales_warehouse** database, the development team made changes to the sales site. As a result, the original OLTP database changed. Now, Sage needs to ensure the ETL pipeline works properly and that the warehouse data matches the original OLTP database.

Sage used the original OLTP schema from the **store** database to design the warehouse.



The **store** database schema also contains five tables—Sales, Products, Users, Locations, and Orders—which are connected via keys. The tables contain four to eight columns ranging in data type. The data types include varchar or char, integer, decimal, date, text, timestamp, bit, tinyint, and datetime.

Every table in the **store** database has an **id** field as a primary key. The database contains the following tables:

- The **Sales** table has price, quantity, and date columns. It references a user who made a sale (**UserId**), purchased a product (**ProductId**), and a related order (**OrderId**). Also, it references the **Locations** table for shipping and billing addresses (**ShippingAddressId** and **BillingAddressId,** respectively).
- The **Users** table has **FirstName**, **LastName**, **Email**, **Password,** and other user-related columns.
- The **Locations** table contains address information (**Address1**, **Address2**, **City**, **State,** and **Postcode**).
- The **Products** table has **Name**, **Price**, **InventoryNumber,** and **Category** of products.
- The **Orders** table has **OrderNumber** and purchase information (**Subtotal**, **ShippingFee**, **Tax**, **Total,** and **Status**).

Using SQL to find problems

Sage compared the **sales_warehouse** database to the original **store** database to check for completeness, consistency, conformity, accuracy, redundancy, integrity, and timeliness. Sage ran SQL queries to examine the data and identify quality problems. Then Sage prepared the following table of lists, which include the types of quality issues found, the quality strategies that were violated, the SQL codes used to find the issues, and specific descriptions of the issues.

Quality testing *sales_warehouse*

Tested quality	Quality strategy	SQL query	Sage's observation
Integrity	Is the data accurate, complete, consistent, and trustworthy?	**SELECT * FROM Orders**	In the **sales_warehouse** database, the order with ID 7 has the incorrect total value.
Completeness	Does the data contain all of the desired components or measures?	**SELECT COUNT(*) FROM Locations**	The **Locations** table of the **sales_warehouse** database has an extra address. In the **store** database there are 60 records, whereas the **sales_warehouse** database table has 61.
Consistency	Is the data compatible and in agreement across all systems?	**SELECT Phone FROM Users**	Several users within the **sales_warehouse** database have phones without the "+" prefix.
Conformity	Does the data fit the required destination format?	**SELECT id, postcode FROM sales_warehouse.Locations**	The location ZIP code for the record with ID 6 in the **sales_warehouse** database is 722434213, which is wrong. The United States postal code contains either five digits or five digits followed by a hyphen (dash) and another four digits (e.g., 12345-1234).

Quality testing *store*

Feature	Quality Strategy	SQL query	Sage's Observation
Integrity	Is the data accurate, complete, consistent, and trustworthy?	**DESCRIBE Users**	**Users.Status** from the **store** database and **Users.is_active** from the **sales_warehouse** database seem to be related fields. However, it is not obvious how the **Status** column is transformed into the **is_active** boolean column. Is it possible that with a new status value, the ETL pipeline will fail?
Consistency	Is the data compatible and in agreement across all systems?	**DESCRIBE Products**	**Products.Inventory** from the **store** database has the **varchar** type instead of the **int(10)** in the **sales_warehouse** database **Products.inventory** field. This can be a problem if there is a value with characters.
Accuracy	Does the data conform to the actual entity being measured or described?	**DESCRIBE Sales**	The data type of **Sales.Date** in the **store** database is different from its data type in **sales_warehouse** (**date** vs **datetime**). It might not be a problem if time is not important for the **sales_warehouse** database fact table.
Redundancy	Is only the necessary data being moved, transformed, and stored for use?	**DESCRIBE Sales**	The table **Sales** from the **sales_warehouse** database has a unique index constraint on **OrderId**, **ProductId**, **UserId** columns. It can be added to the warehouse schema.

Key takeaways

Testing data quality is an essential skill of a BI professional that ensures good analytics and reporting. Just as Sage does in this example, you can use SQL commands to examine BI databases and find potential problems. The sooner you know the problems in your system, the sooner you can fix them and improve your data quality.