

Lesson 5

Topic: Data Modeling Basics

Prerequisites:

Download customer.csv, product.csv, sales.csv

Lesson 5 – Data Modeling Basics

1. What is a primary key in a table?

A primary key is a unique identifier for each row in a table. For example:

- CustomerID is the primary key in the **Customer** table.
- ProductID is the primary key in the **Products** table.
- OrderID is the primary key in the **Sales** table.

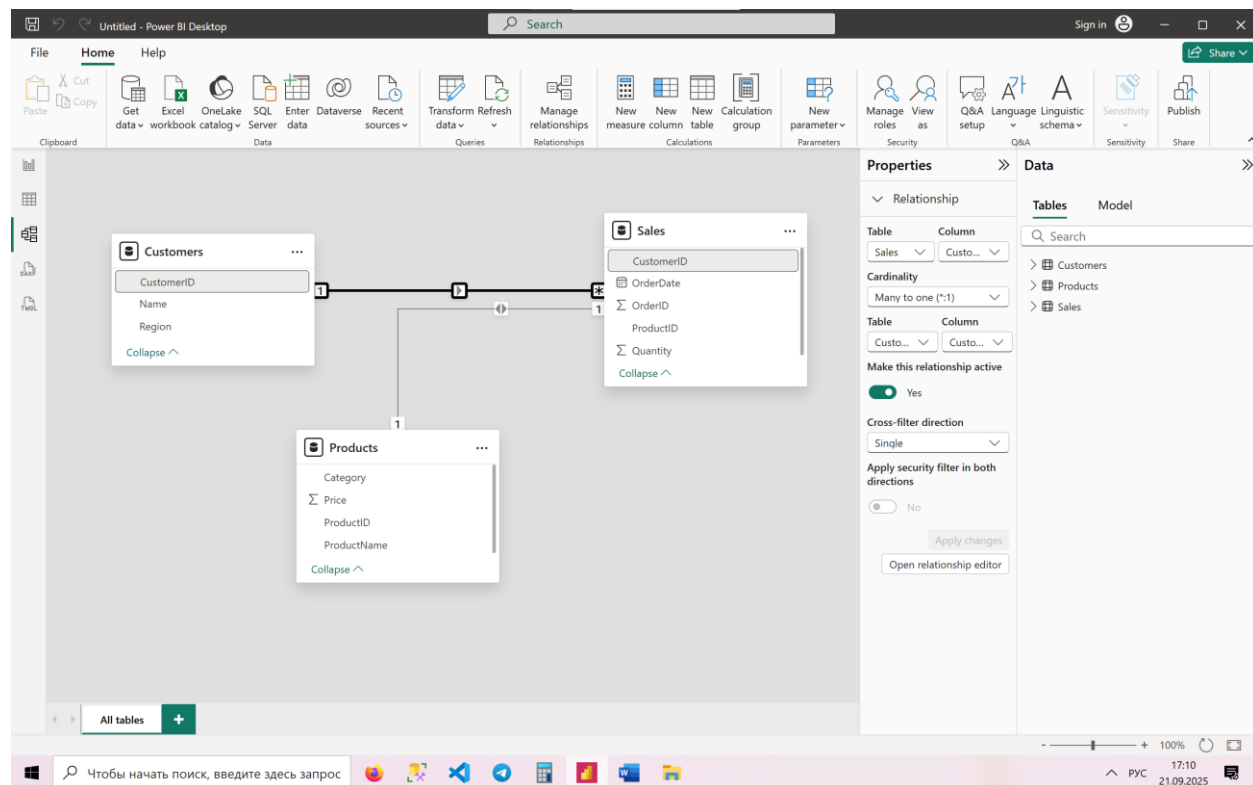
2. Name the two types of table relationships in Power BI.

The two main relationship types are:

- **One-to-Many (1:*)**
- **Many-to-Many (:)**

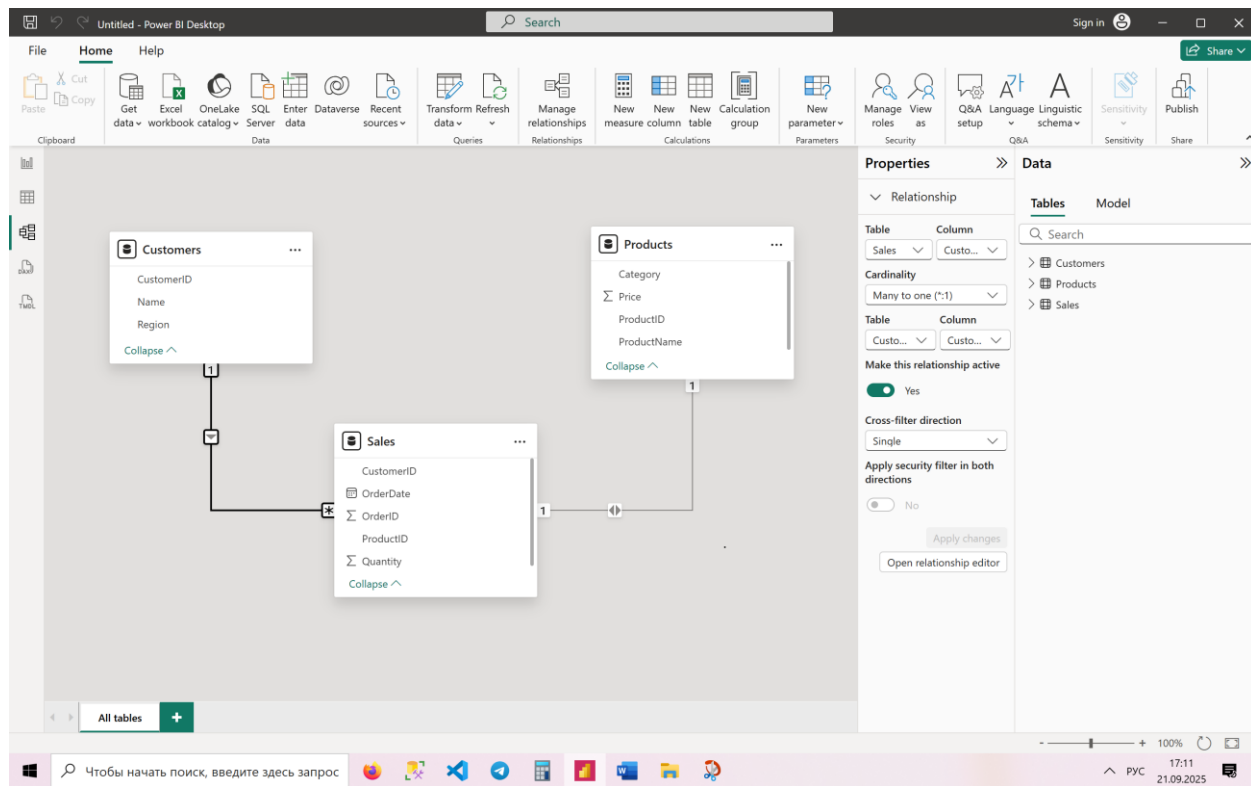
3. How do you create a relationship between two tables in Power BI?

Go to **Model View** in Power BI. Drag a field (for example CustomerID in Customer) to the same field in another table (CustomerID in Sales). Power BI will automatically detect and create the relationship.



4. What is a "star schema"?

A star schema is a data model where a central **fact table** (such as Sales) connects to multiple surrounding **dimension tables** (like Customers, Products, Dates). This forms a star-like structure.



Example of star schema

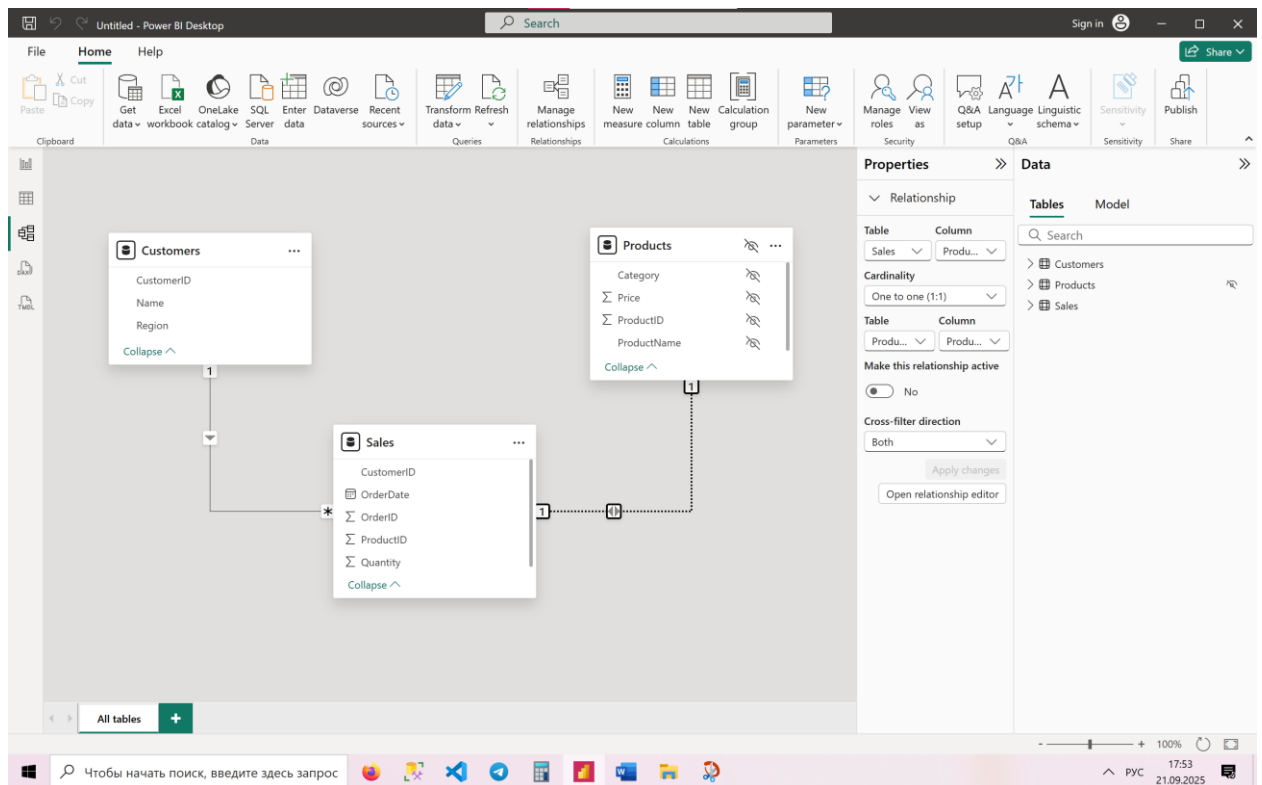
5. Which table is typically the fact table in a sales dataset?

In a sales dataset, the **Sales table** is the fact table because it contains transaction-level data (who bought, what product, when, and how much).

6. Link Sales.csv to Customers.csv using CustomerID (one-to-many).

Create a relationship between:

- Customer[CustomerID] (one side) → Sales[CustomerID] (many side).



7. Why is ProductID in Sales.csv a foreign key?

Because it refers to the unique ProductID in the **Products** table. This way each sales record knows which product was sold.

8. Fix a relationship error where ProductID has mismatched data types. Open **Transform Data** in Power BI. Check both Sales[ProductID] and Products[ProductID]. Change their data type to the same type, usually **Whole Number**.

Untitled - Power BI Desktop

Search

Sign in

File Home Help Table tools Column tools

Name ProductID Format Whole number

Data type Whole number

Summarization Don't summarize

Data category Uncategorized

Sort by column

Data groups

Manage relationships

New column

Structure

Formatting

Properties

Sort

Groups

Relationships

Calculations

OrderID	CustomerID	ProductID	Quantity	OrderDate
1001	101	1	2	5 января 2023 г.
1002	102	2	5	10 января 2023 г.
1003	101	3	1	15 января 2023 г.

Table: Sales (3 rows) Column: ProductID (3 distinct values)

Search

Customers

Products

Sales

CustomerID

OrderDate

OrderID

ProductID

Quantity

Пуск

Чтобы начать поиск, введите здесь запрос

Пуск

17:56 21.09.2025

Untitled - Power BI Desktop

Search

Sign in

File Home Help Table tools Column tools

Name ProductID Format Whole number

Data type Whole number

Summarization Don't summarize

Data category Uncategorized

Sort by column

Data groups

Manage relationships

New column

Structure

Formatting

Properties

Sort

Groups

Relationships

Calculations

ProductID	ProductName	Category	Price
1	Laptop	Electronics	1200
2	Mouse	Accessories	25
3	Keyboard	Accessories	80

Table: Products (3 rows) Column: ProductID (3 distinct values)

Search

Customers

Products

Category

Price

ProductID

ProductName

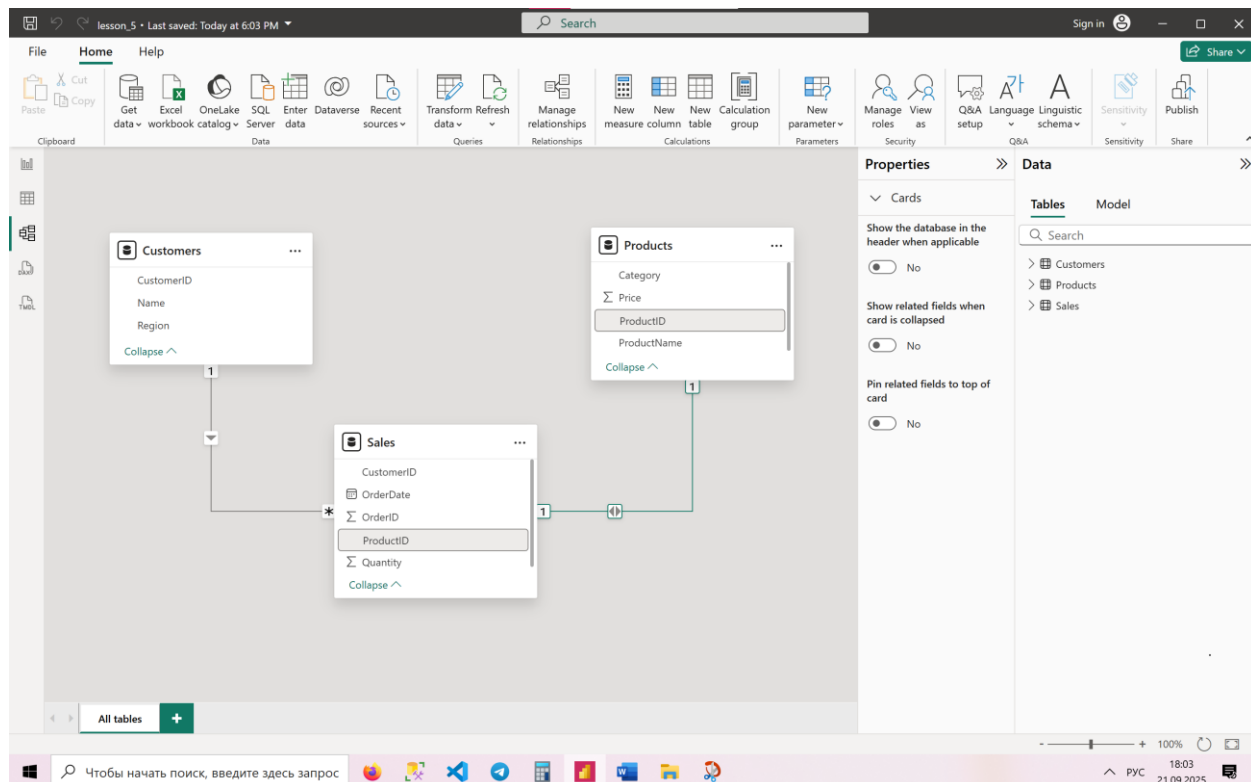
Sales

Пуск

Чтобы начать поиск, введите здесь запрос

Пуск

17:59 21.09.2025



9. Explain why a star schema improves performance.

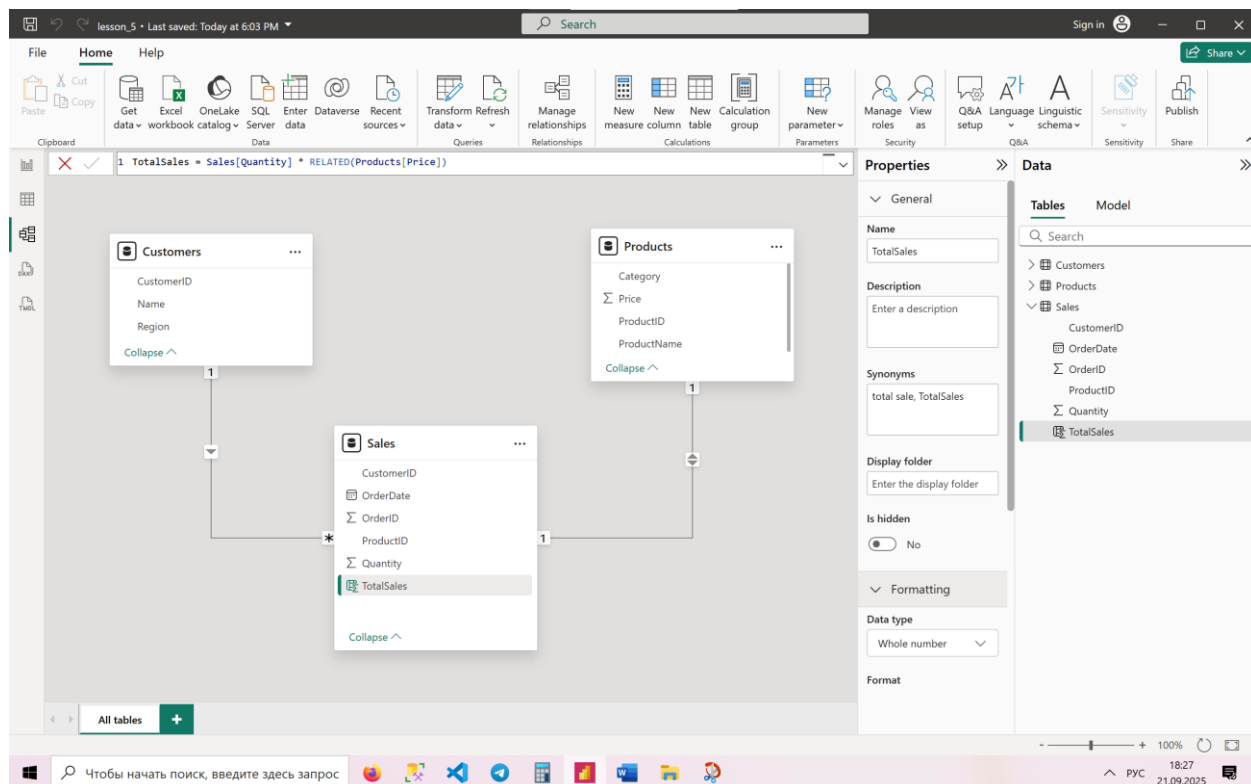
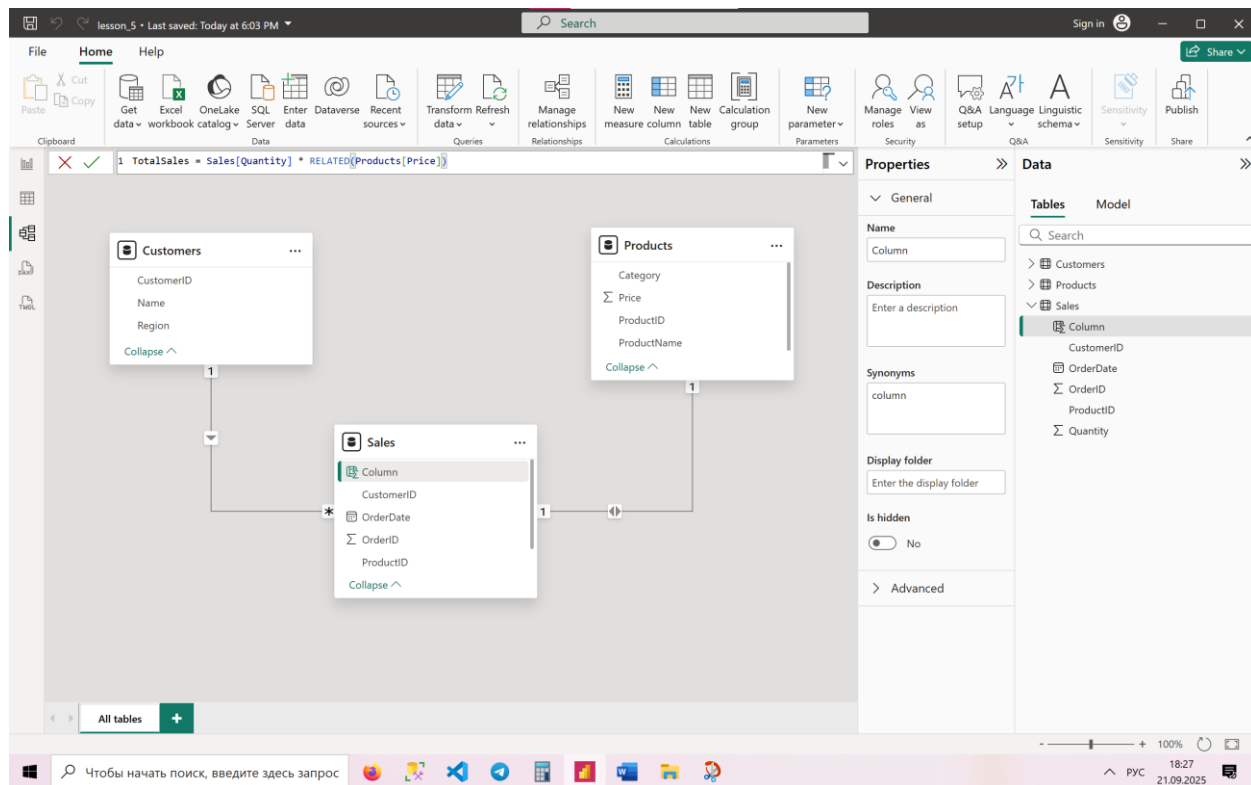
A star schema improves performance because it:

- Simplifies queries.
- Reduces the number of joins.
- Makes aggregations faster.
- Prevents circular or ambiguous relationships.

10. Add a new column TotalSales in Sales (Quantity * Price from Products).

In Power BI, go to **Modeling** → **New Column** and enter this DAX formula:

$\text{TotalSales} = \text{Sales}[\text{Quantity}] * \text{RELATED}(\text{Products}[\text{Price}])$

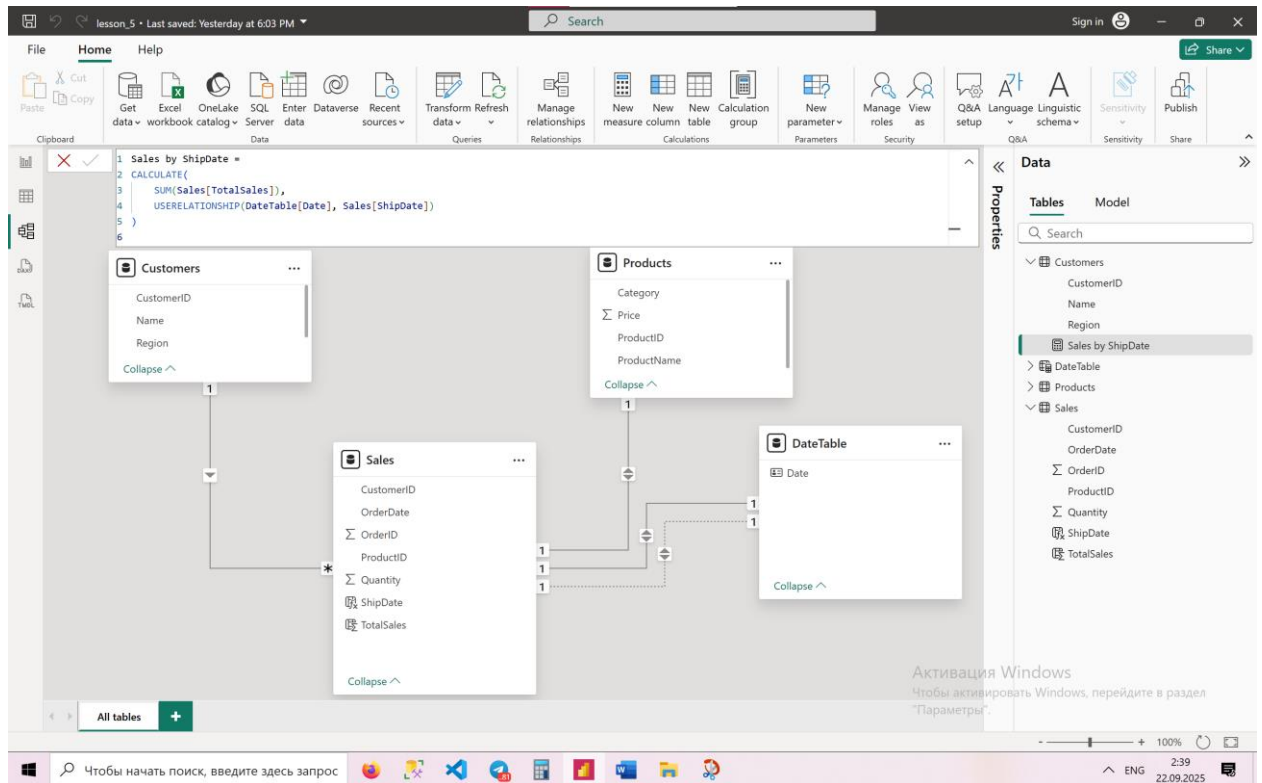


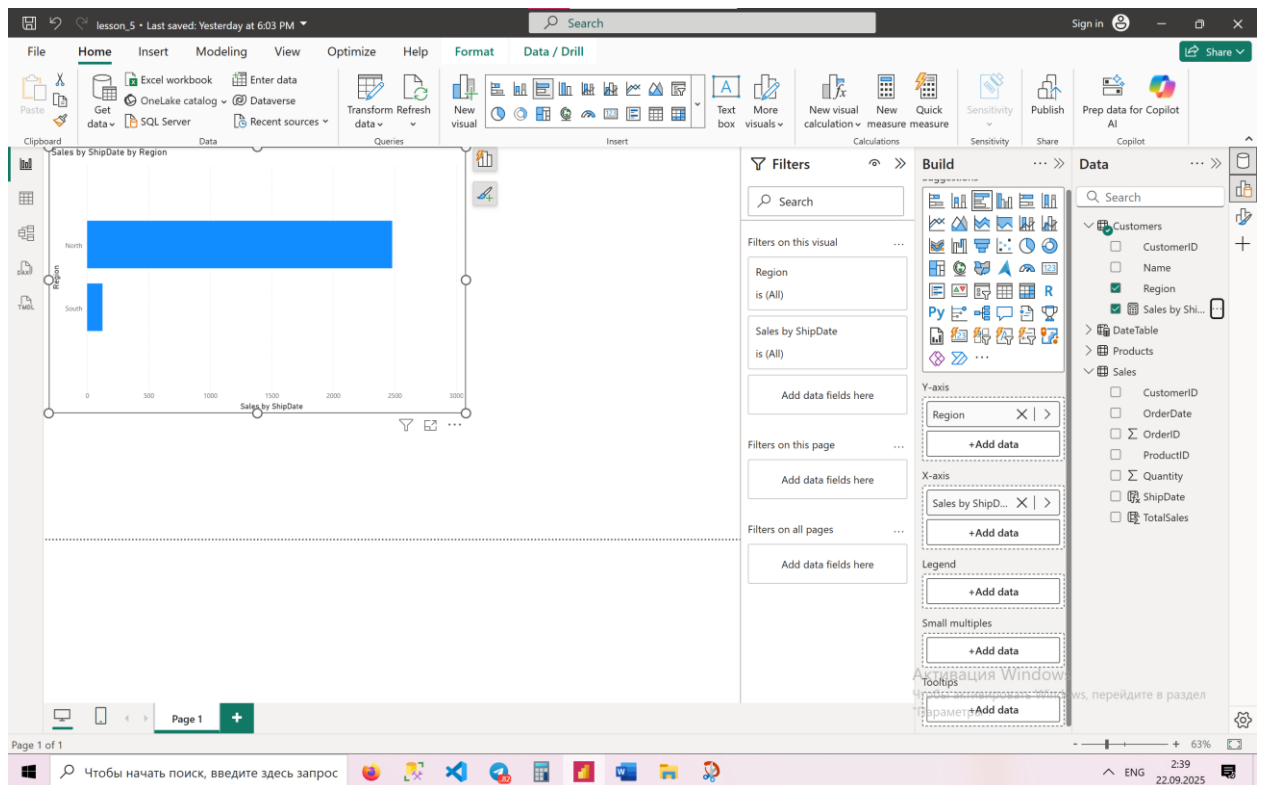
11. Optimize a model with circular relationships—how would you resolve it?

To resolve circular relationships:

- Remove unnecessary relationships.

12. Create a role-playing dimension for OrderDate and ShipDate.
Duplicate the Date table so that you have two separate tables:





13. Handle a many-to-many relationship between Customers and Products.

In this case we use the **Sales table** as a bridge between Customers and Products. This allows to analyze which customers bought which products without creating a direct many-to-many link.

To handle a many-to-many relationship between Customers and Products, the best practice is to use a bridge table (CustomerID–ProductID). This table resolves the relationship into two one-to-many relationships, which keeps the star schema clean and avoids calculation errors. Direct many-to-many relationships in Power BI should be avoided unless absolutely necessary.

lesson_5 • Last saved: Yesterday at 6:03 PM

Search

File Home Help Table tools Column tools

Name ProductID Format Whole number Summarization Sum Data type Whole number Data category Uncategorized

Structure Formatting Properties Sort Data groups Manage relationships New column

CustomerID ProductID

101	1
101	3
102	2
103	1
103	2

Table: CustomerProductBridge (5 rows) Column: ProductID (3 distinct values)

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

ENG 2:48 22.09.2025

lesson_5 • Last saved: Yesterday at 6:03 PM

Search

File Home Help

Clipboard Get data Excel data OneLake data SQL Server data Enter data Datawarehouse Recent sources Transform data Refresh data Queries Manage relationships Relationships New measure column Calculations Calculation group New parameter Parameters Manage roles View as Q&A setup Language Linguistic schema Sensitivity Publish

Customers CustomerProductBridge Products DateTable Sales

CustomerID Name Region

CustomerID ProductID

Category Price ProductID ProductName

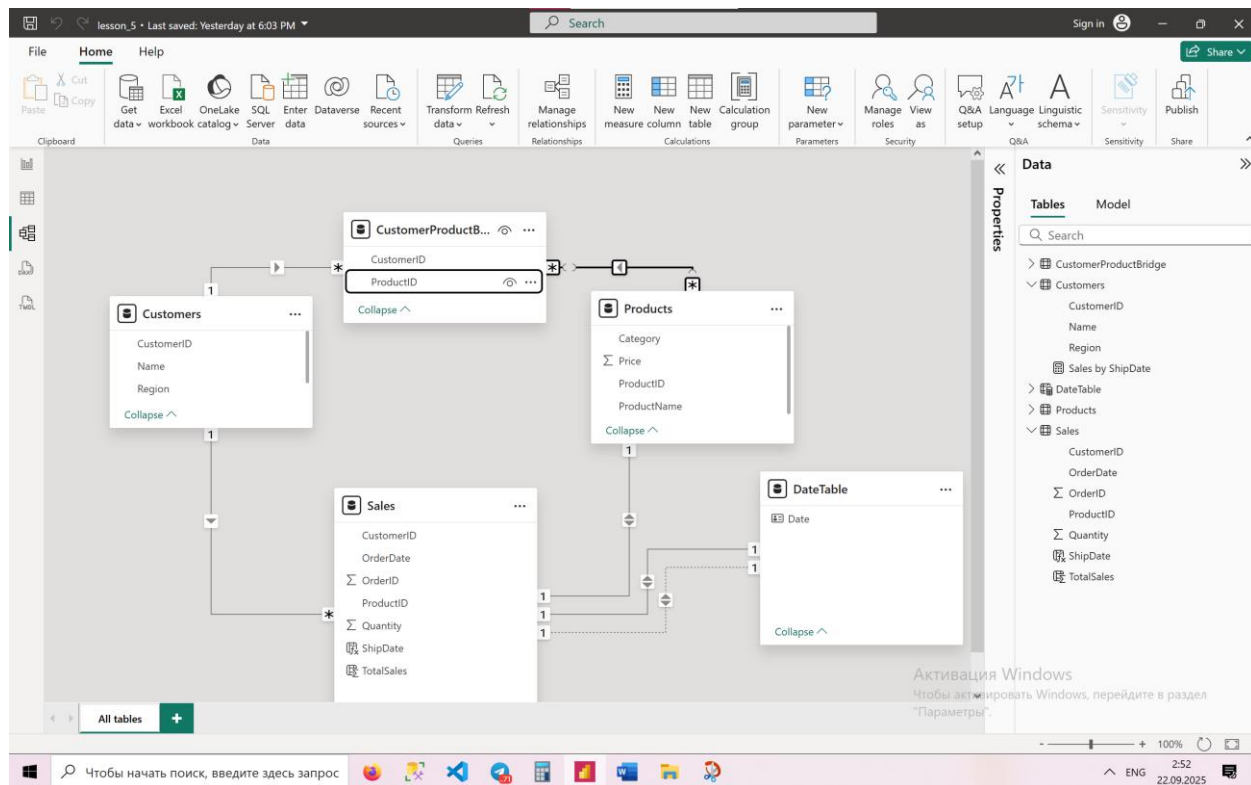
CustomerID OrderDate ProductID Quantity ShipDate TotalSales

Date

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

ENG 2:49 22.09.2025



14. Use **bidirectional filtering** sparingly—when is it appropriate?

Bidirectional filtering should be used only when needed, for example:

- In **row-level security (RLS)** to apply filters across related tables.
- In complex many-to-many scenarios where both sides need to filter each other.

Bidirectional filtering should be used sparingly. It is appropriate when you need to filter in both directions, such as in many-to-many relationships with a bridge table or when building small lookup tables that must influence both sides. However, in most cases, single-direction filtering is preferred to avoid ambiguity, circular dependencies, and performance issues.

15. Write **DAX** to enforce referential integrity if a **CustomerID** is deleted.

You can create a measure to detect orphaned sales records:

```
InvalidOrders =
COUNTROWS(
    FILTER(
        Sales,
        ISBLANK(RELATED(Customer[CustomerID]))
    )
)
```

This measure will return the number of Sales rows that no longer have a valid CustomerID.

To enforce referential integrity if a CustomerID is deleted, you can use DAX with LOOKUPVALUE or RELATED to check if the customer exists. If not, return “Unknown Customer” or exclude the sales row. This ensures reports remain consistent and do not break when IDs are missing.
