Data Modeling with Power BI

Module 5 (Part 2)

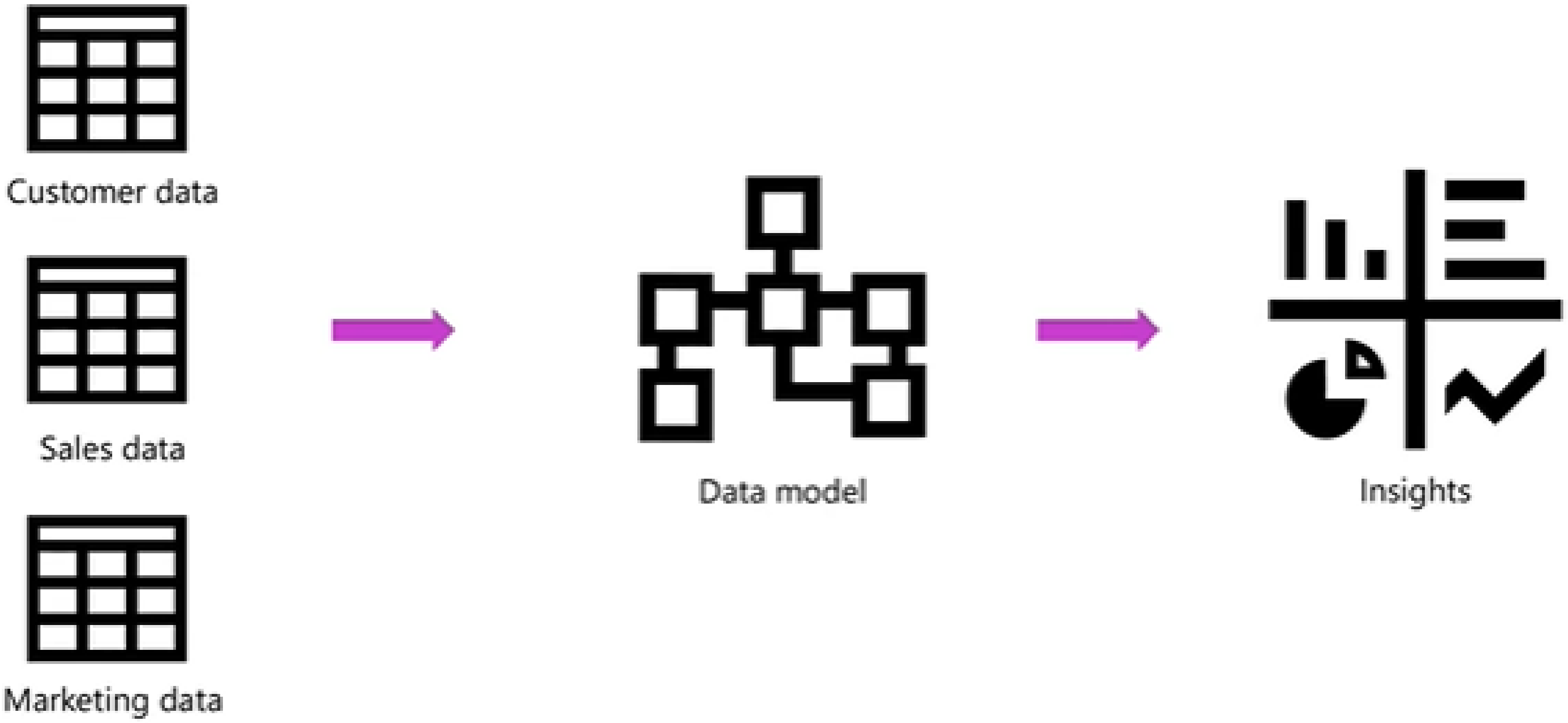
Structure

- Part 1: Extract, Transform and Load (ETL) – identify, explain and configure multiple data sources in Power BI; clean and transform data using Power Query
- Part 2: Data Modeling – identify and create appropriate model relationships; configuring your table and column properties; data analysis expressions (DAX) to configure and optimize your models
- Part 3: Data Analysis and Visualization – add visualizations to reports; format visuals

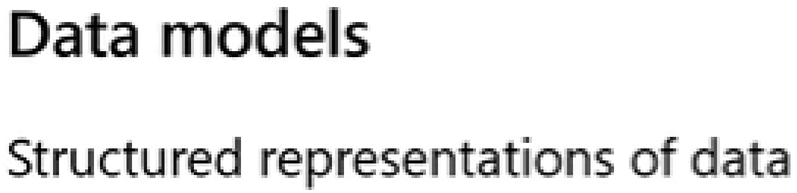
Part 2

Data Modeling

Introduction to Data Models

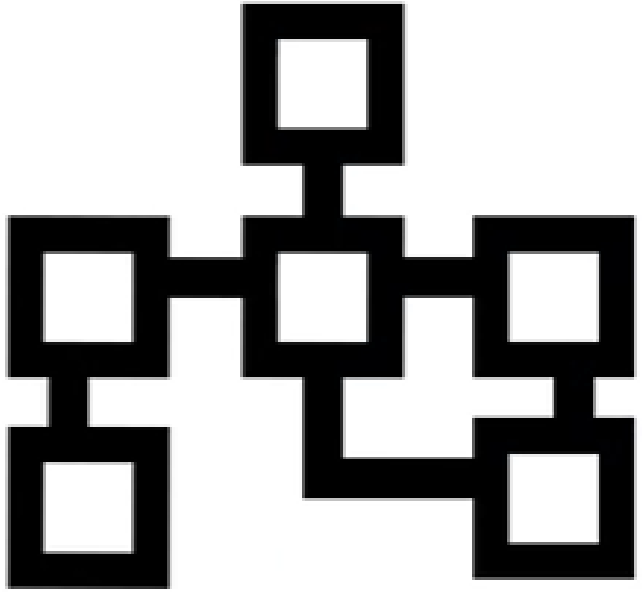


Introduction to Data Models (cont.)

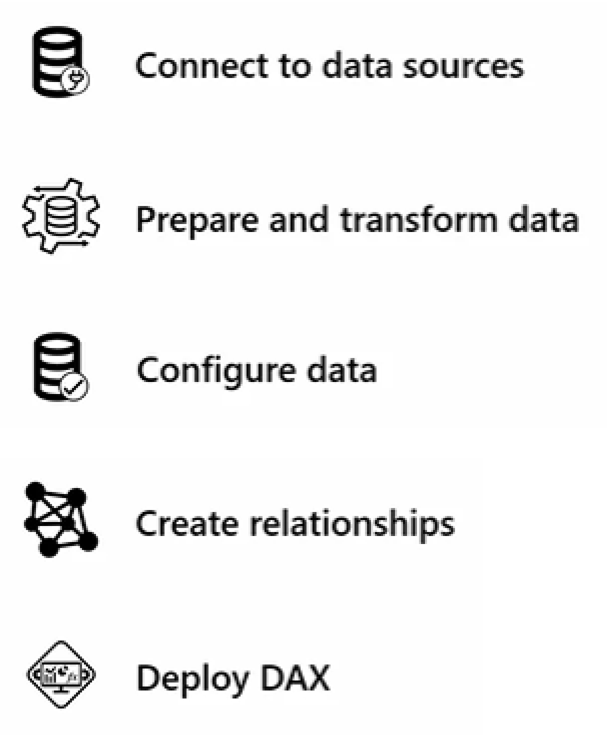


Data models

Structured representations of data

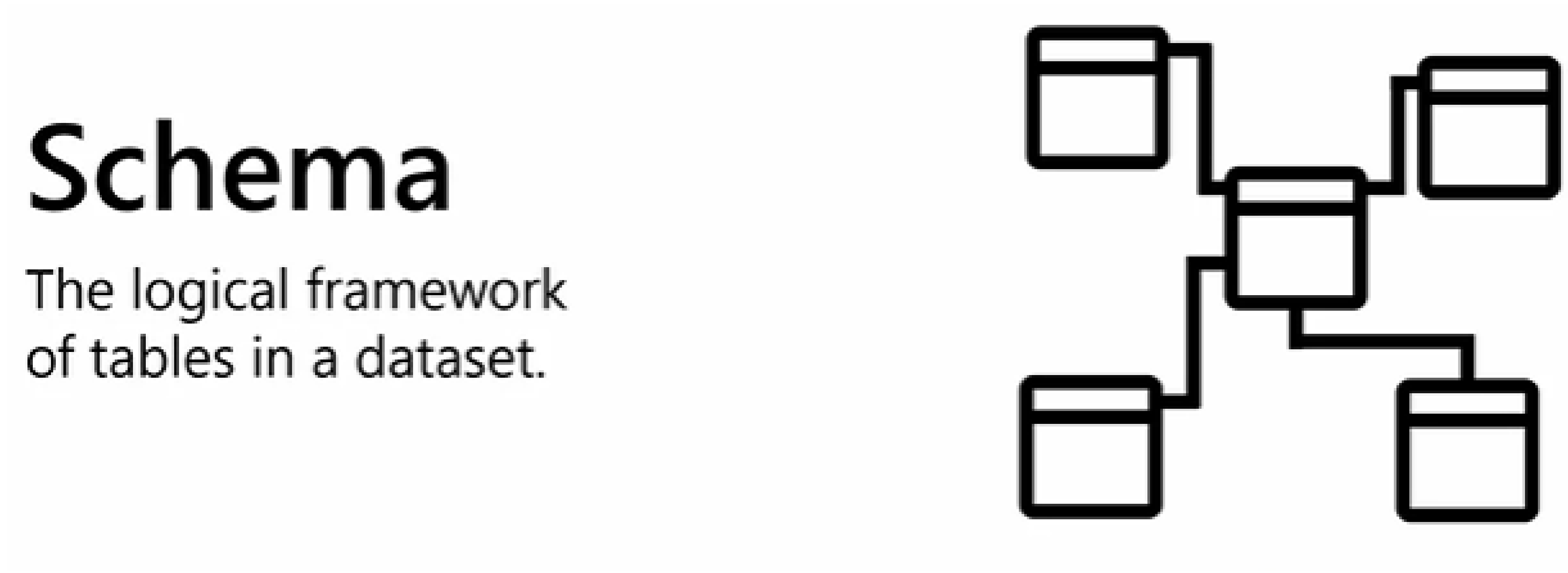


Data Modeling Design Process



- Connect to data sources
- Prepare and transform data
- Configure data
- Create relationships
- Deploy DAX

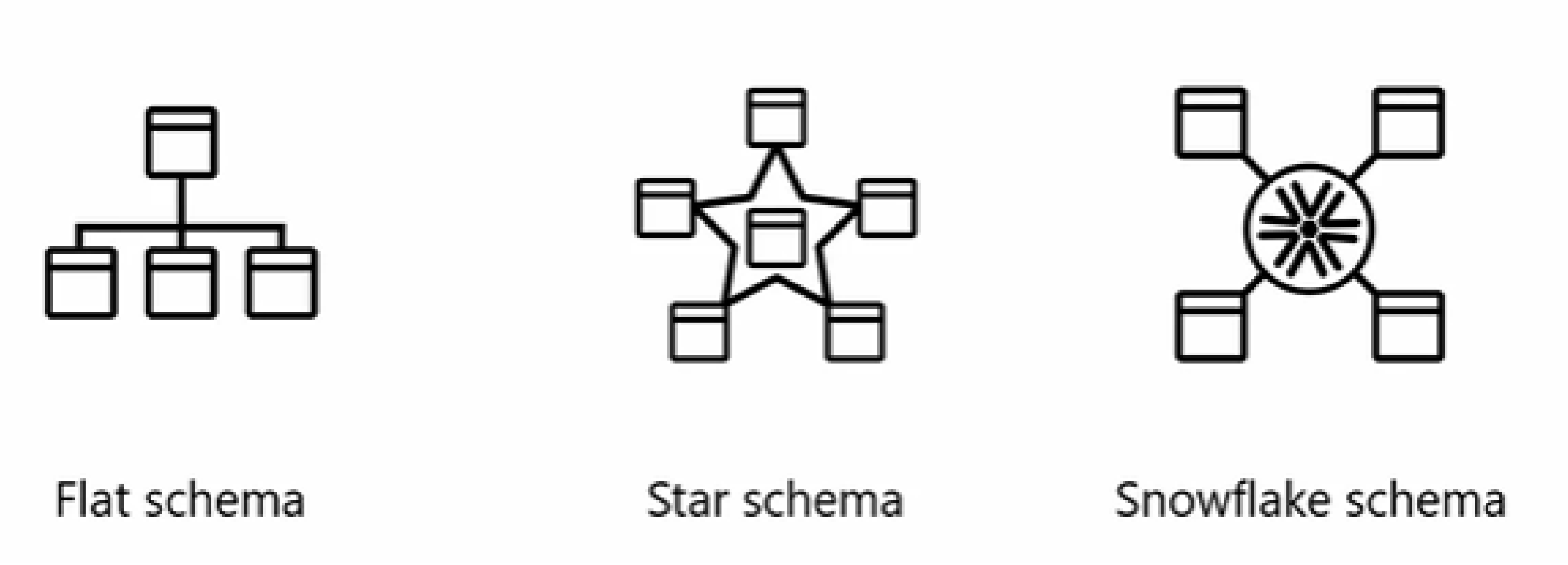
Introduction to Schemas



Schema

The logical framework of tables in a dataset.

Introduction to Schemas – Types

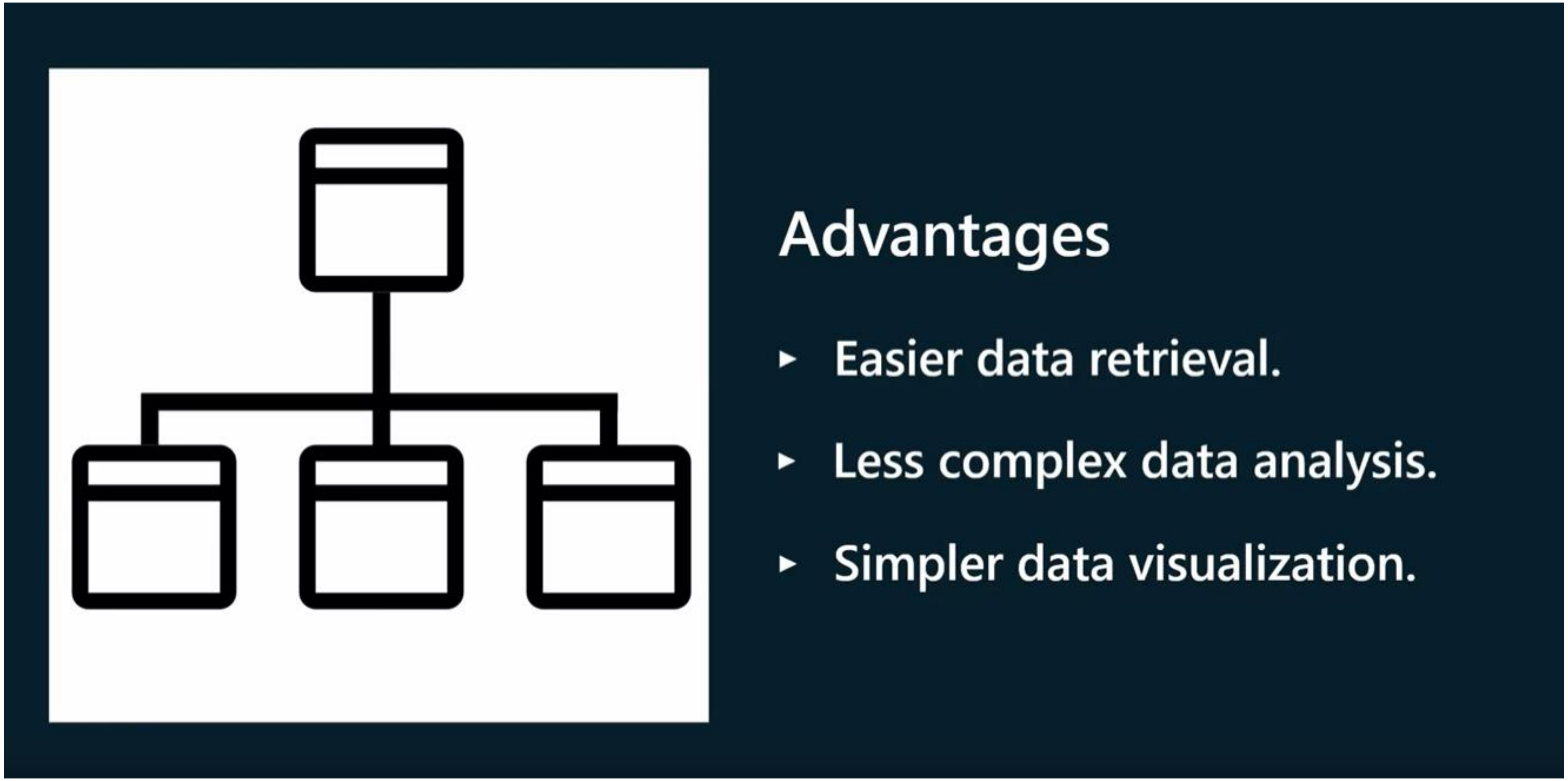


Flat Schema

Flat schema

Stores all data in a single table.

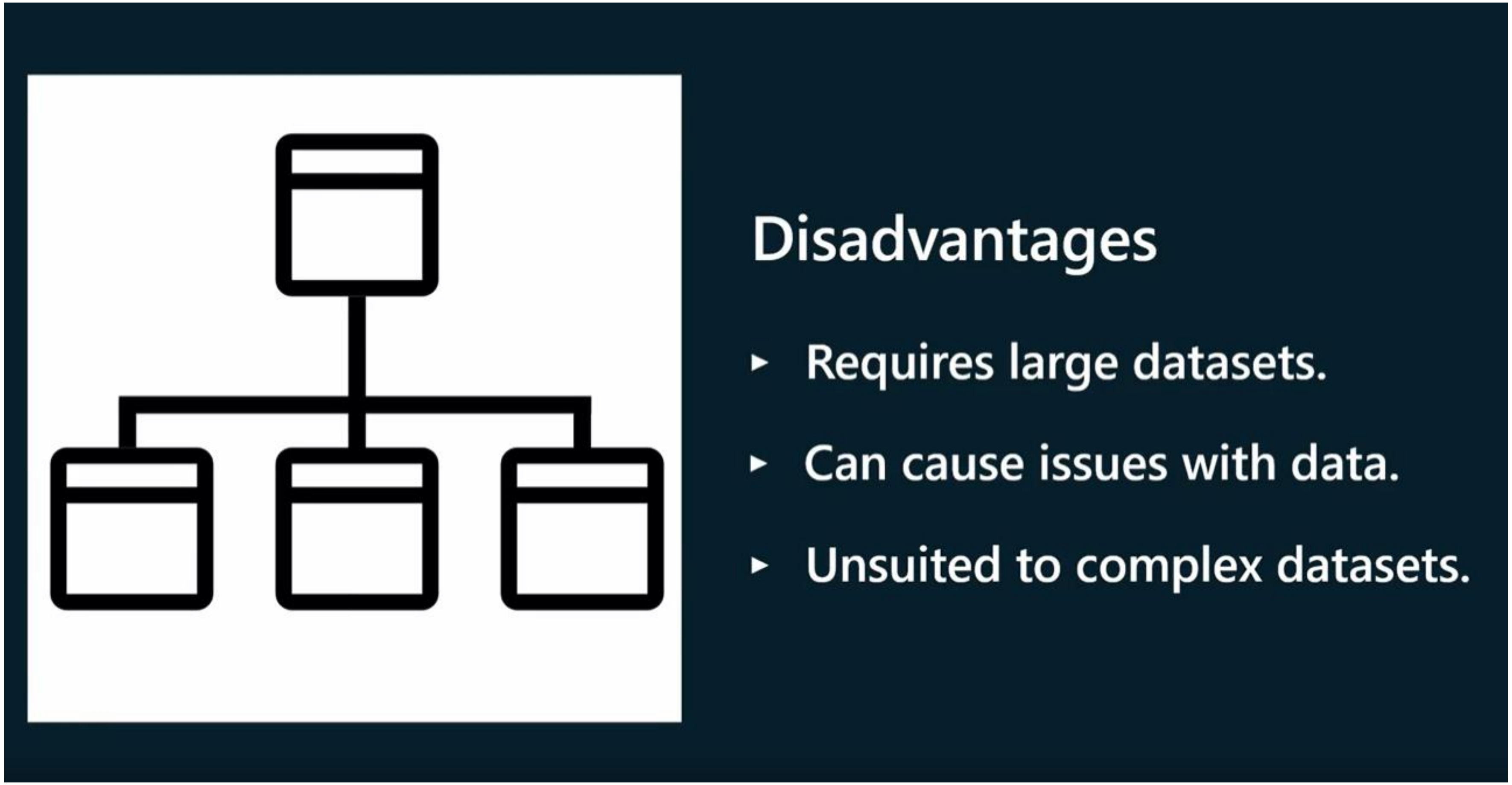
Flat Schema (cont.)



Advantages

- Easier data retrieval.
- Less complex data analysis.
- Simpler data visualization.

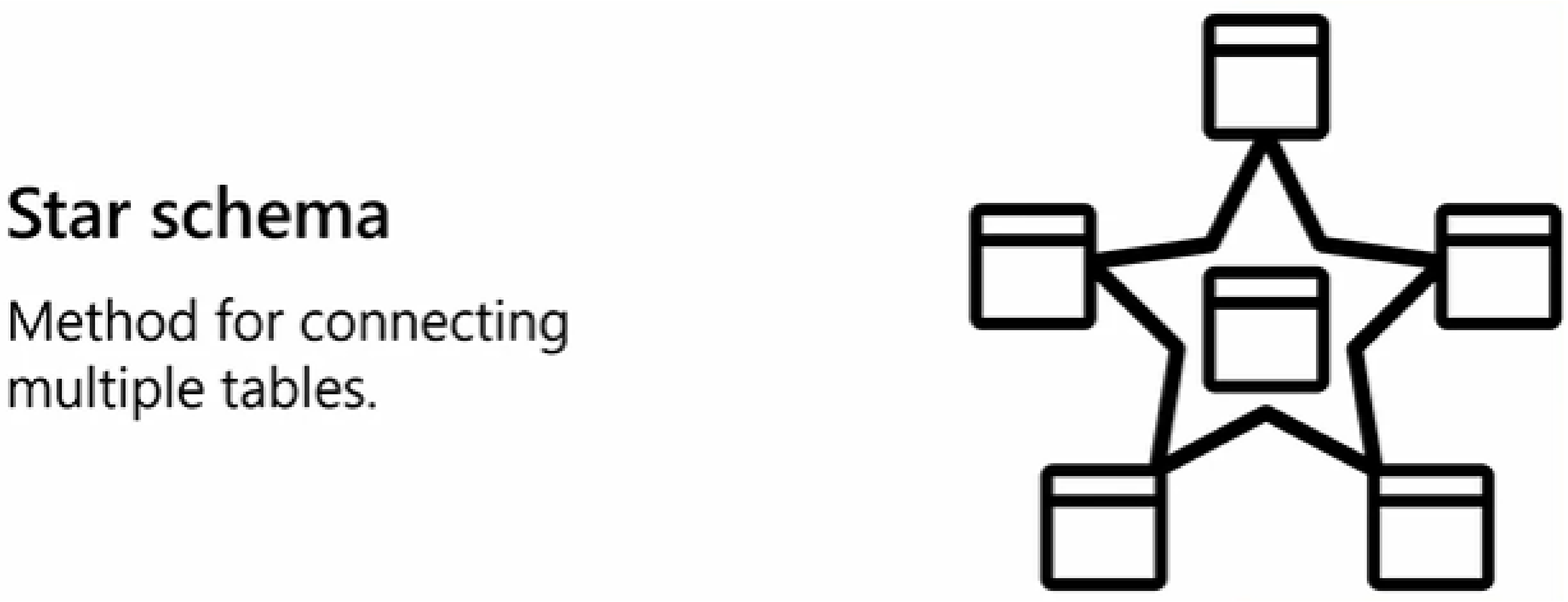
Flat Schema (cont.)



Disadvantages

- Requires large datasets.
- Can cause issues with data.
- Unsuited to complex datasets.

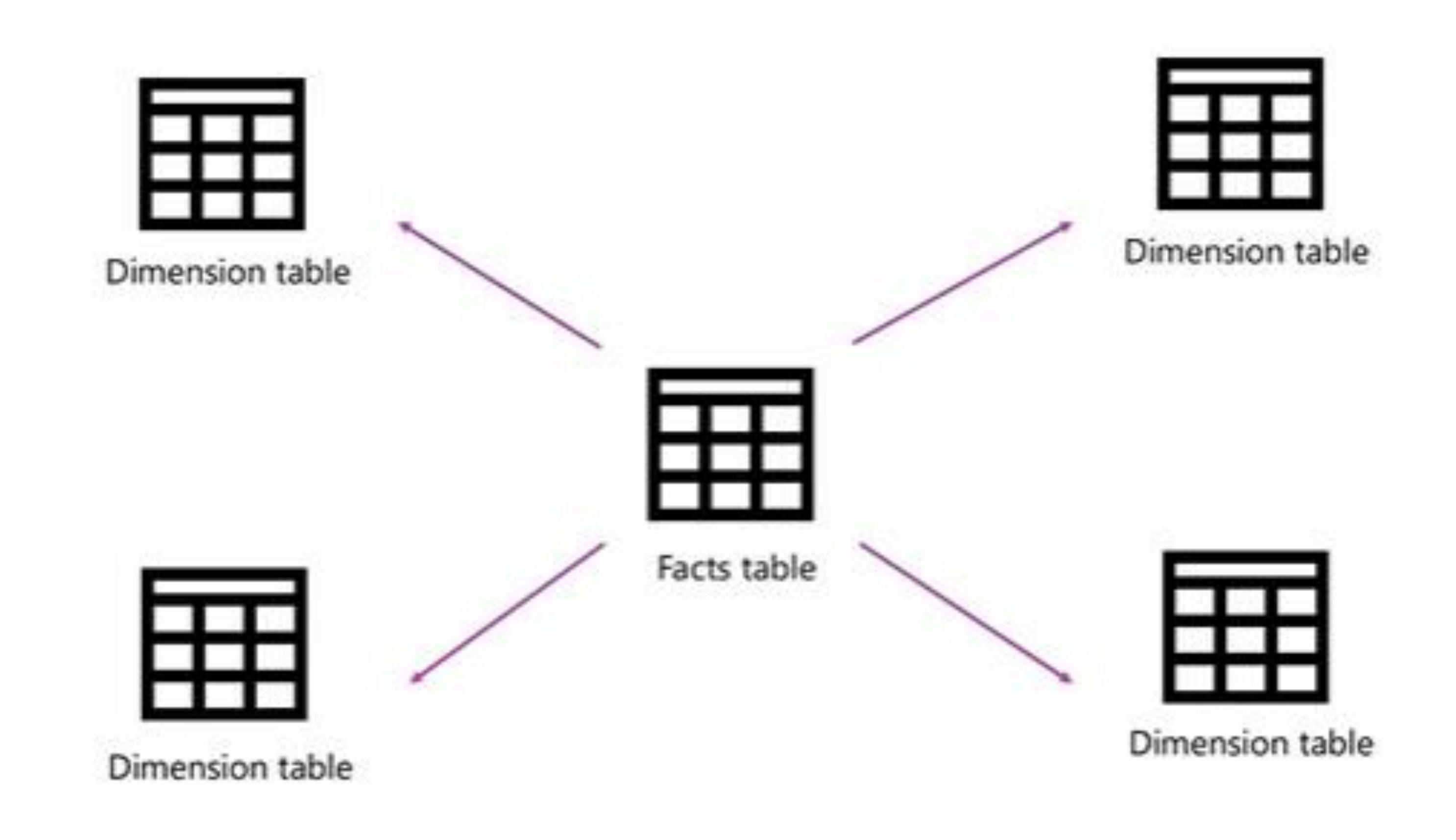
Star Schema



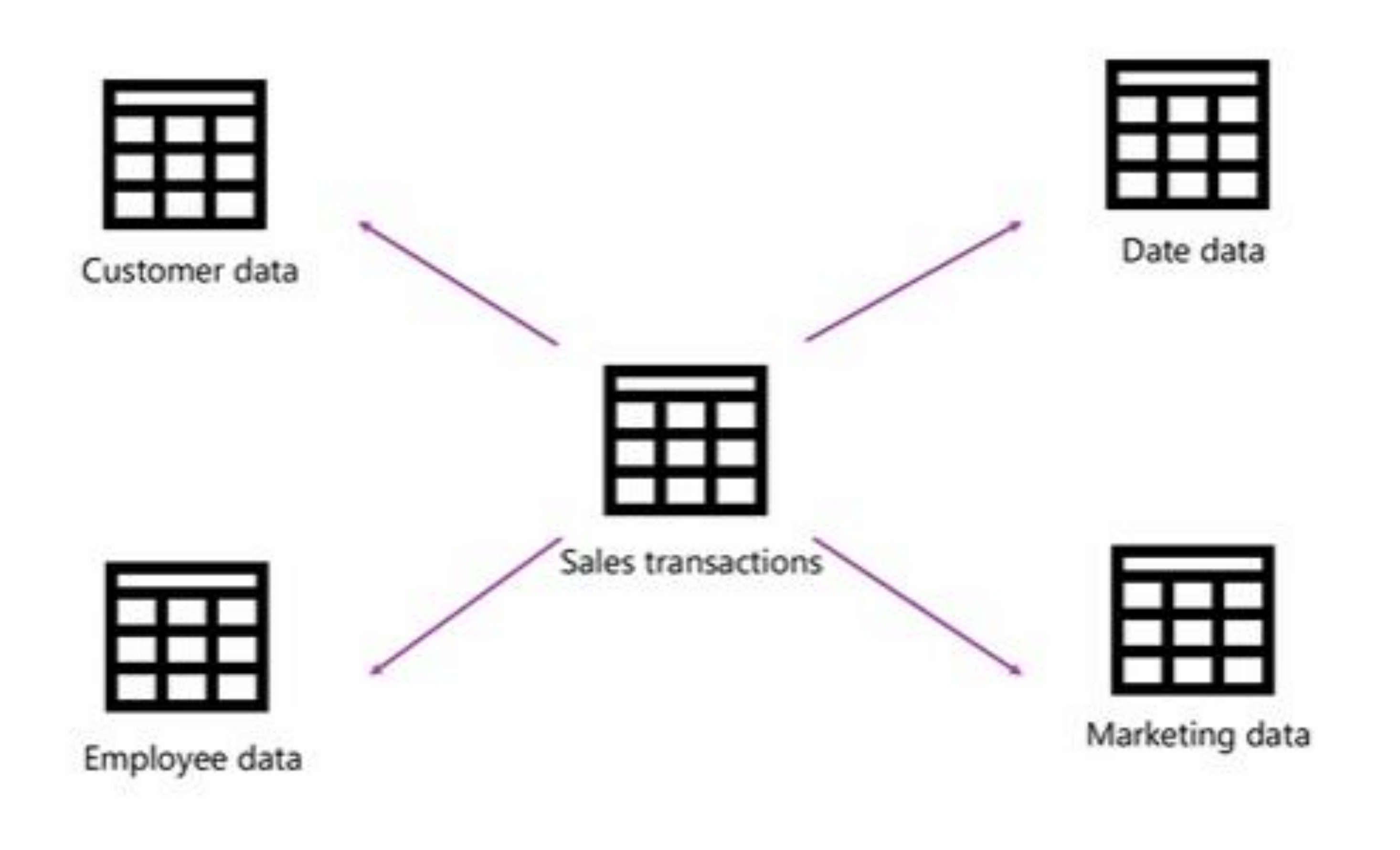
Star schema

Method for connecting multiple tables.

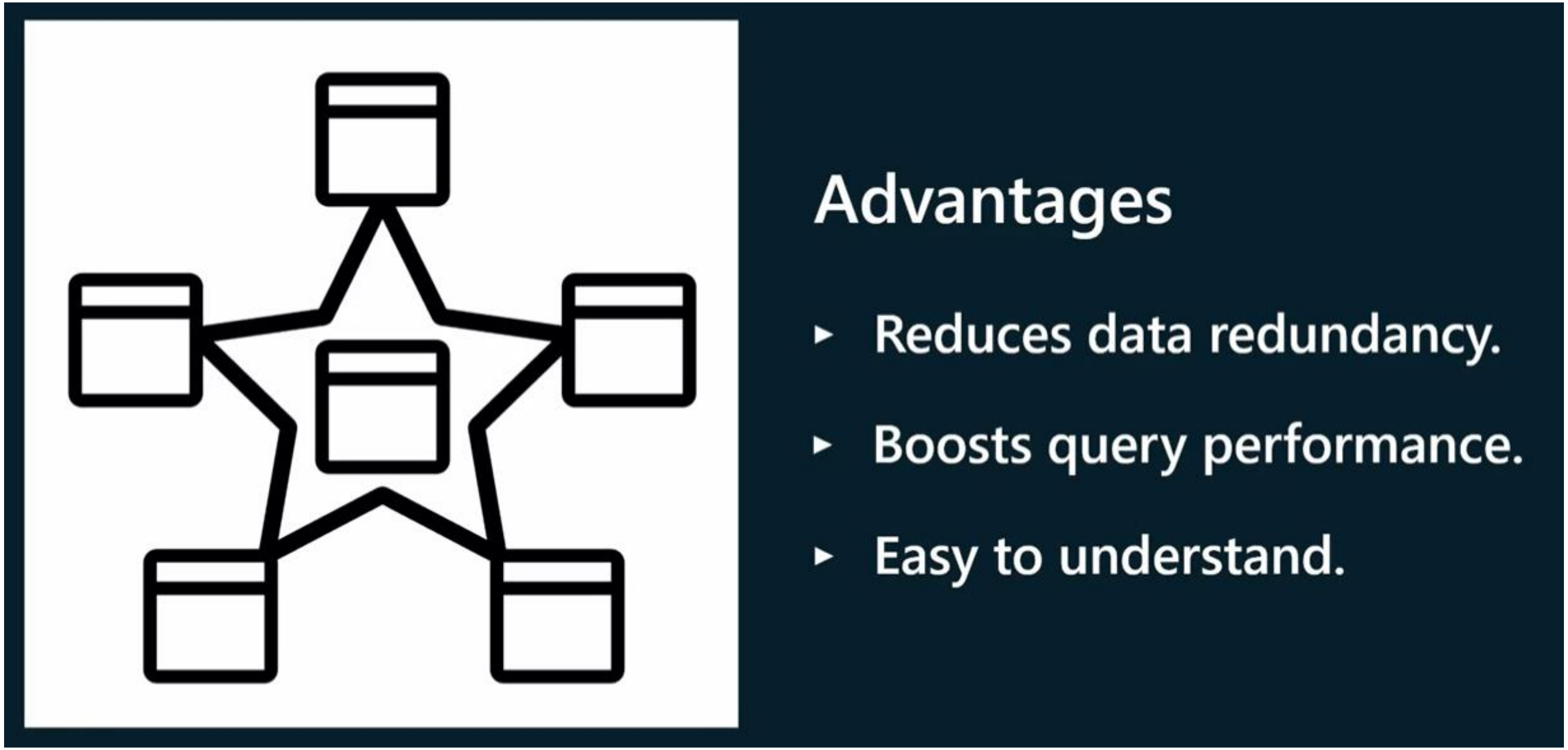
Star Schema (cont.)



Star Schema (cont.)



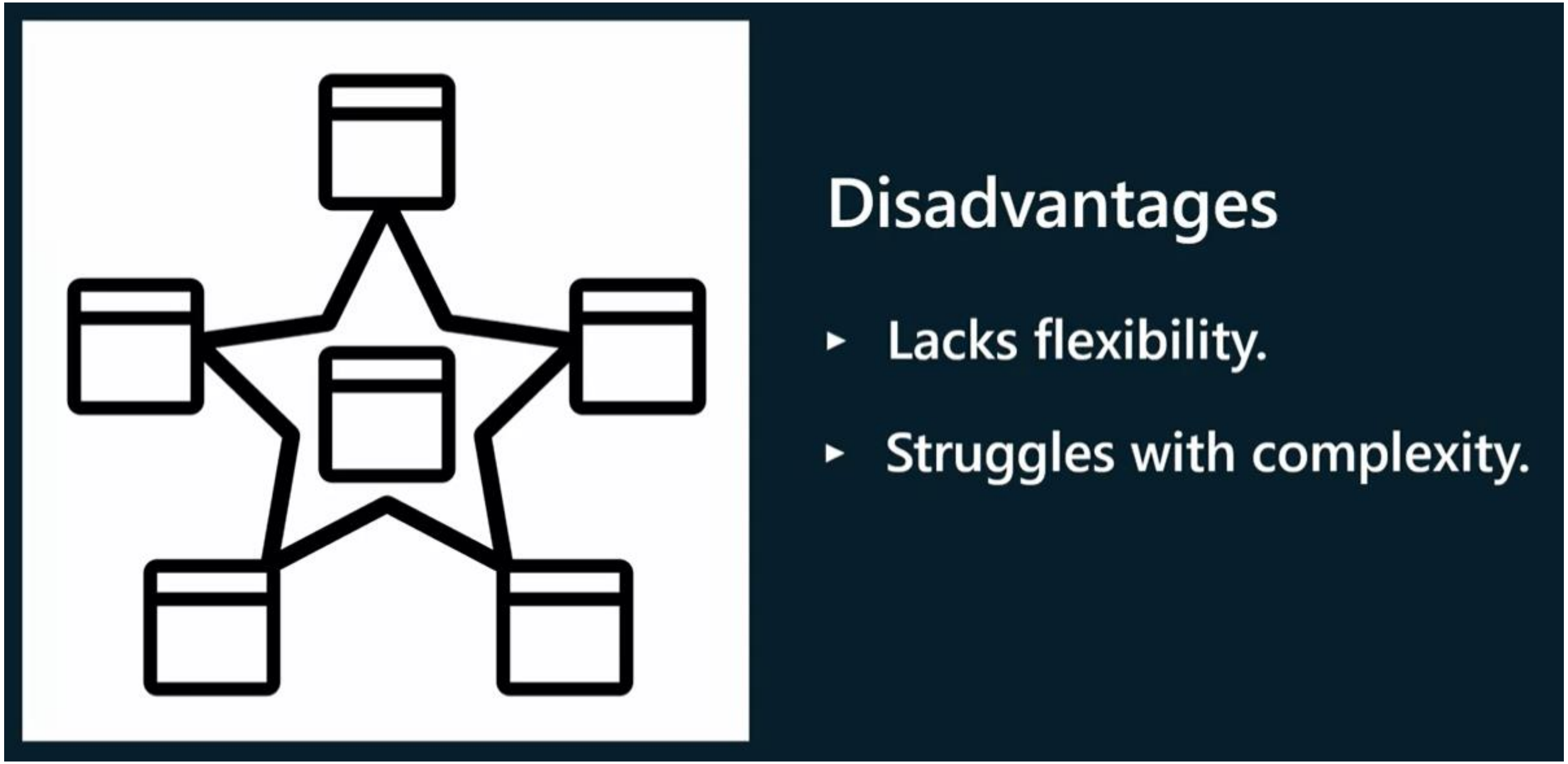
Star Schema (cont.)



Advantages

- Reduces data redundancy.
- Boosts query performance.
- Easy to understand.

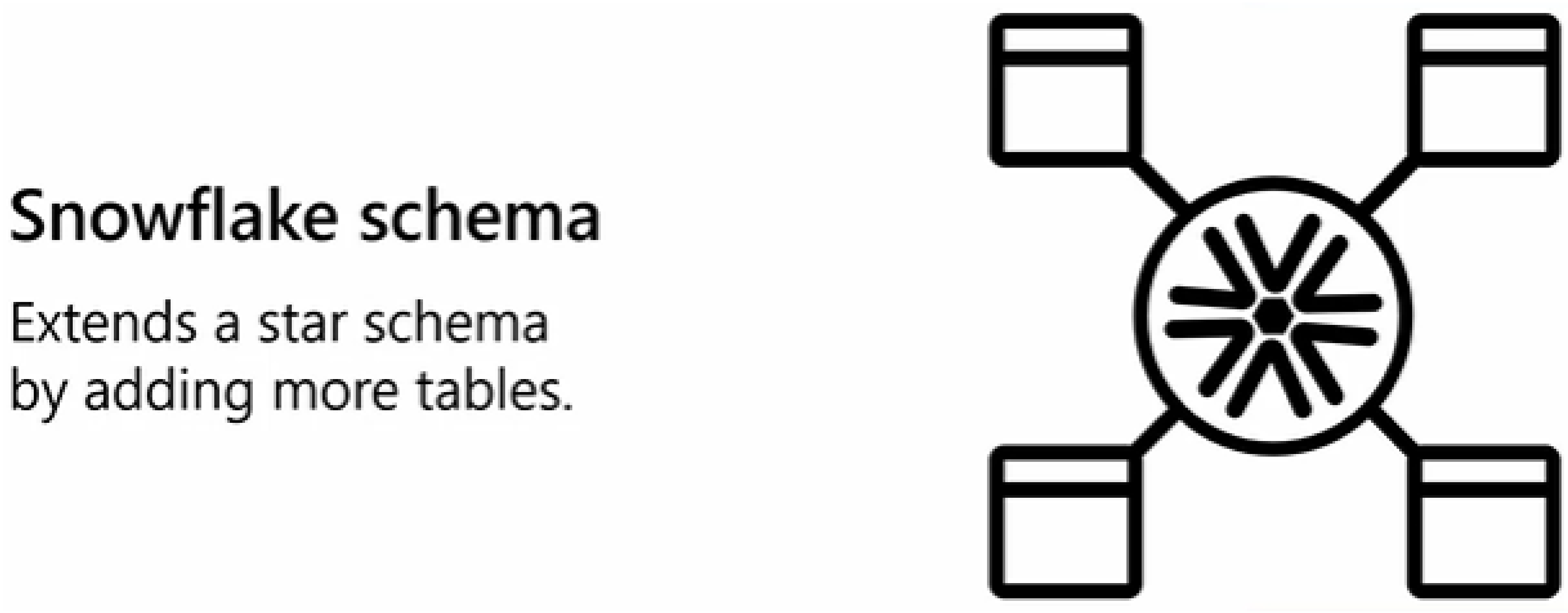
Star Schema (cont.)



Disadvantages

- Lacks flexibility.
- Struggles with complexity.

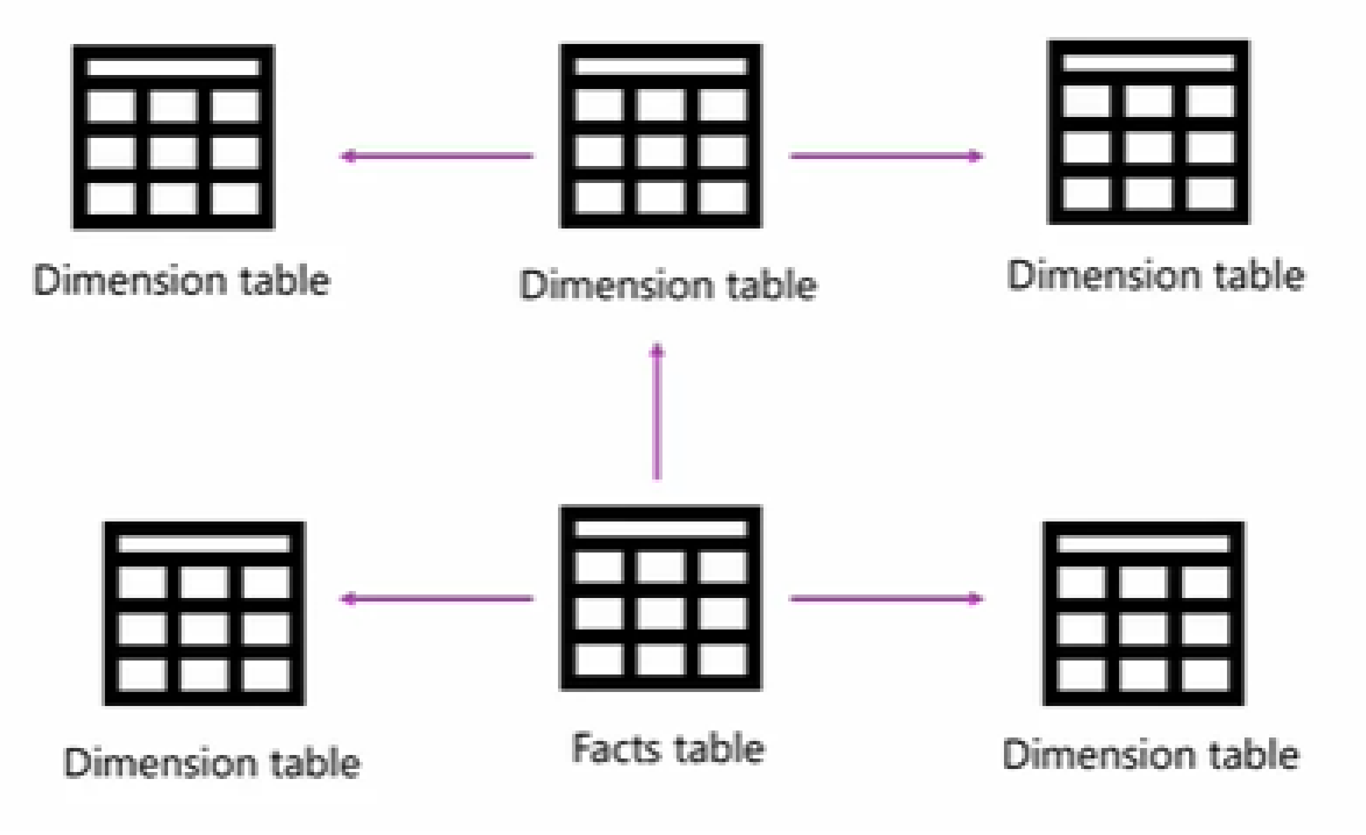
Snowflake Schema



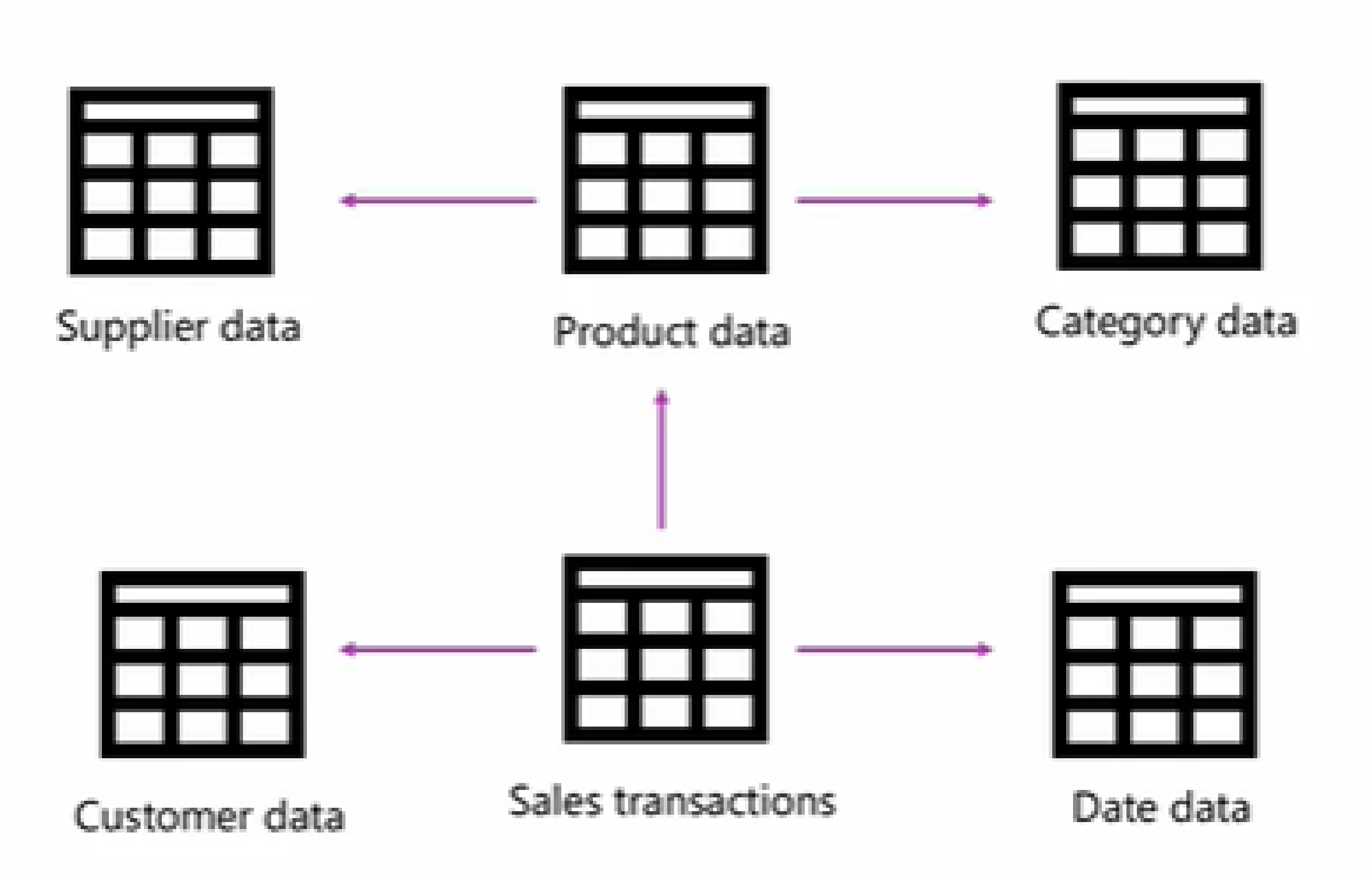
Snowflake schema

Extends a star schema by adding more tables.

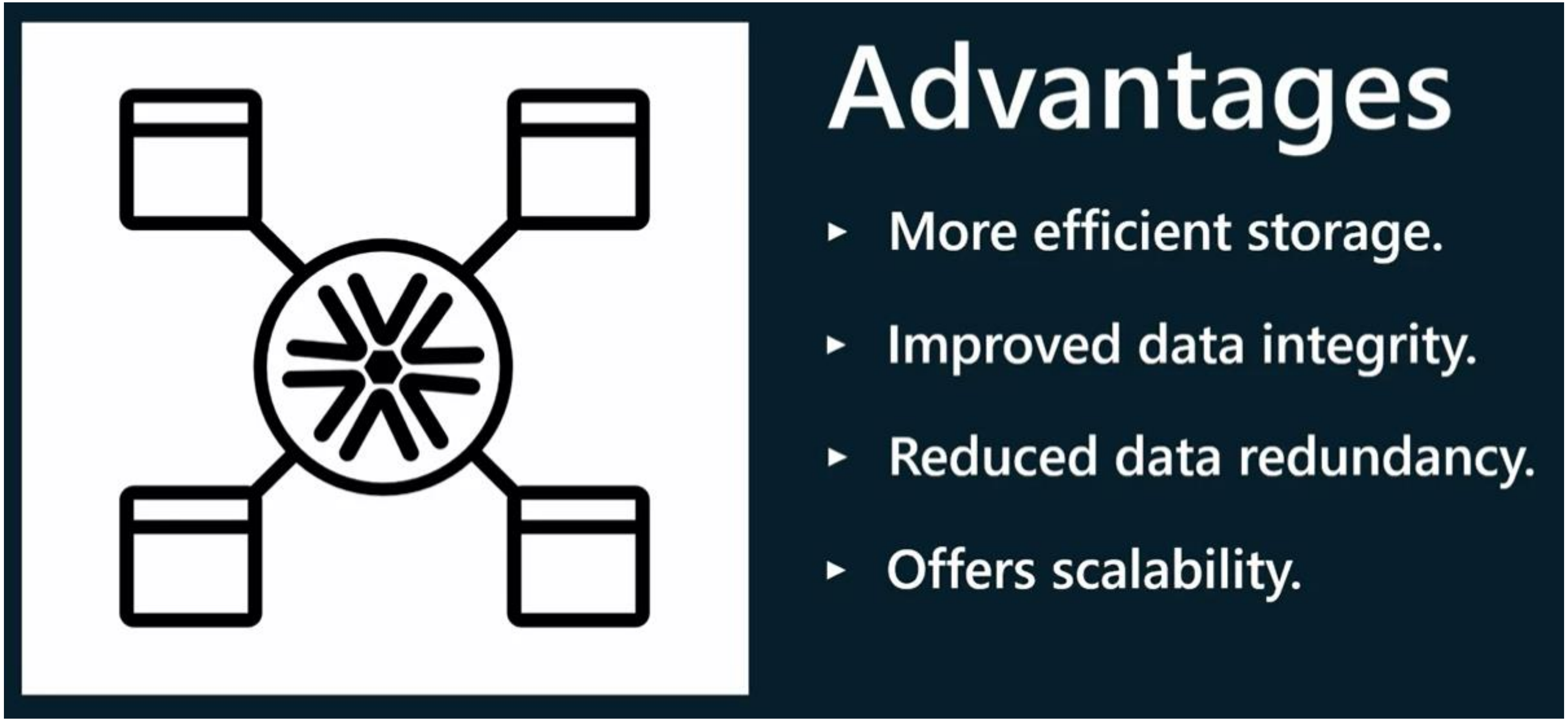
Snowflake Schema (cont.)



Snowflake Schema (cont.)



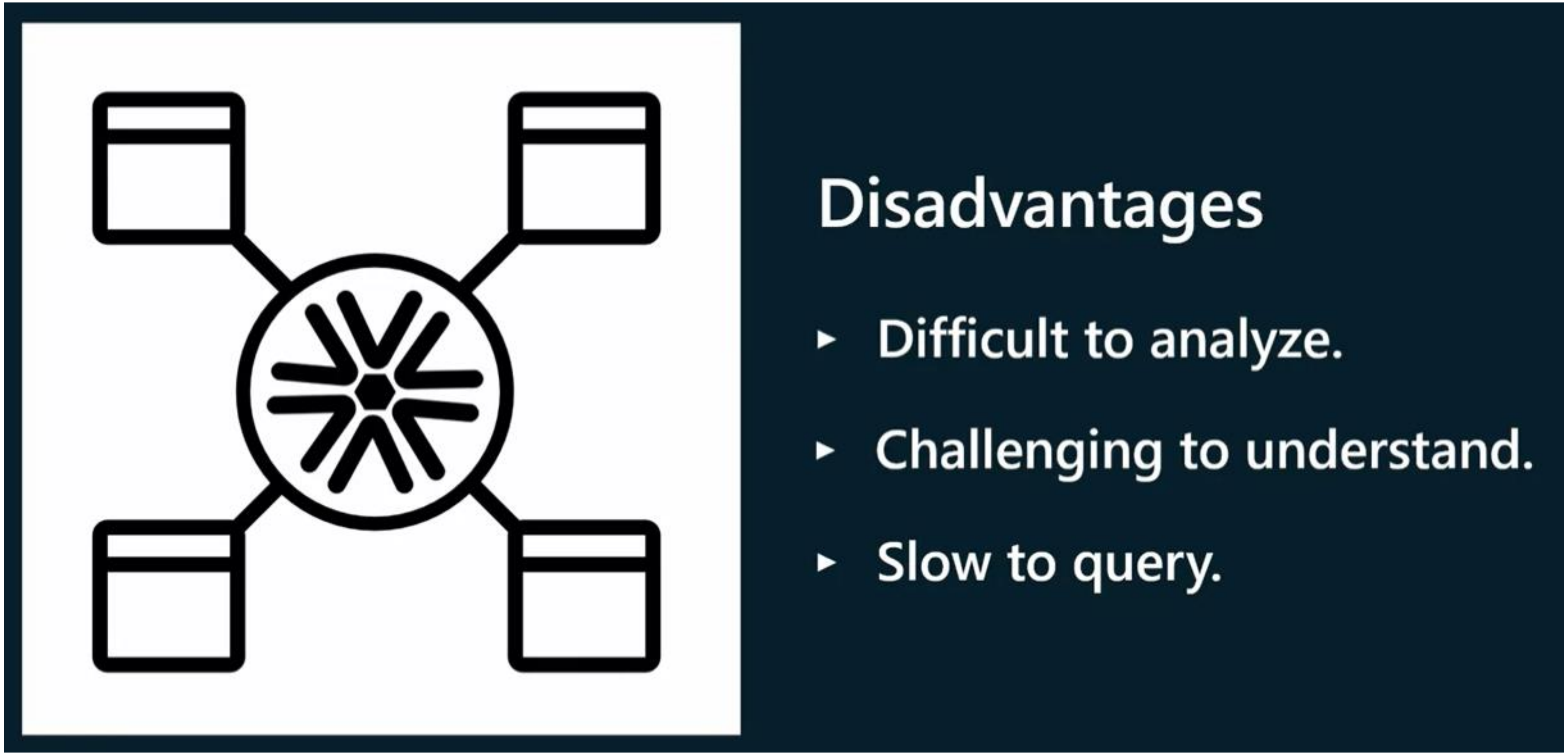
Snowflake Schema (cont.)



Advantages

- More efficient storage.
- Improved data integrity.
- Reduced data redundancy.
- Offers scalability.

Snowflake Schema (cont.)



Disadvantages

- Difficult to analyze.
- Challenging to understand.
- Slow to query.

Exercise: Configuring a Flat schema

Download the Excel Files

Download and open the Microsoft Excel workbook ***AdventureWorksDataset.xlsx***. The workbook contains only one worksheet called ***AdventureWorksData***.

Connect to the Excel workbook and load the data into Power BI

1. Connect to the Excel spreadsheet in Power BI.
2. Select the table from the dataset and load it into the Power BI data model.
3. Open a preview of the table in the **Preview** pane.

Exercise: Configuring a Flat schema (cont.)

Configure the table properties

Configure the table properties by renaming the table to **Product** and adding a brief description of the table.

Configure the column properties

Order ID should be a unique value for each sales order. It’s essential to eliminate all duplicate values in the column to generate accurate analytical results. Identify and remove all duplicate values in the **OrderID** column in the worksheet. Review the number of rows in the **Query Editor** to ensure all duplicate rows have been deleted.

Exercise: Configuring a Flat schema (cont.)

Save the Power BI Project

1. Navigate to the **Model view** of the Power BI desktop and ensure that all required tables are present in the model.
2. Save your flat schema Power BI project to your local machine.

Activity: Configure a Flat schema with multiple sources

Download the Excel Files

Download and open the Microsoft Excel workbook ***AdventureWorksDataset.xlsx***. The workbook contains two worksheets named **Sales** and **Salespersons**.

Get data from the Excel workbook

1. Import the data from the Excel sheet into Power BI.
2. Open a preview of the table in the **Preview pane**.

Activity: Configure a Flat schema with multiple sources (cont.)

Load the tables from your data source and merge the tables into one dataset

1. Identify and remove all duplicate values in the **SalesOrderNumber** column in the **Sales** dataset.
2. Identify common columns with matching or similar values in both tables. These values can be used to merge both tables.
3. In the **Query Editor**, select the **Sales** table and click on **Merge queries**. This action opens the **Merge** dialog box in which you can configure the merge options. You must select the second table to merge with, ensuring you match the column and **Join Kind**.
4. The merged table is displayed as a column at the end of the table. Select the **column** to expand the merged table. Select **Employee ID**, **Salesperson**, and **Title** columns to expand and select **OK**.
5. After merging the tables, you can delete the **Salesperson** table from the model, as this table has been integrated with the **Sales** table as a Flat schema.

Activity: Configure a Flat schema with multiple sources (cont.)

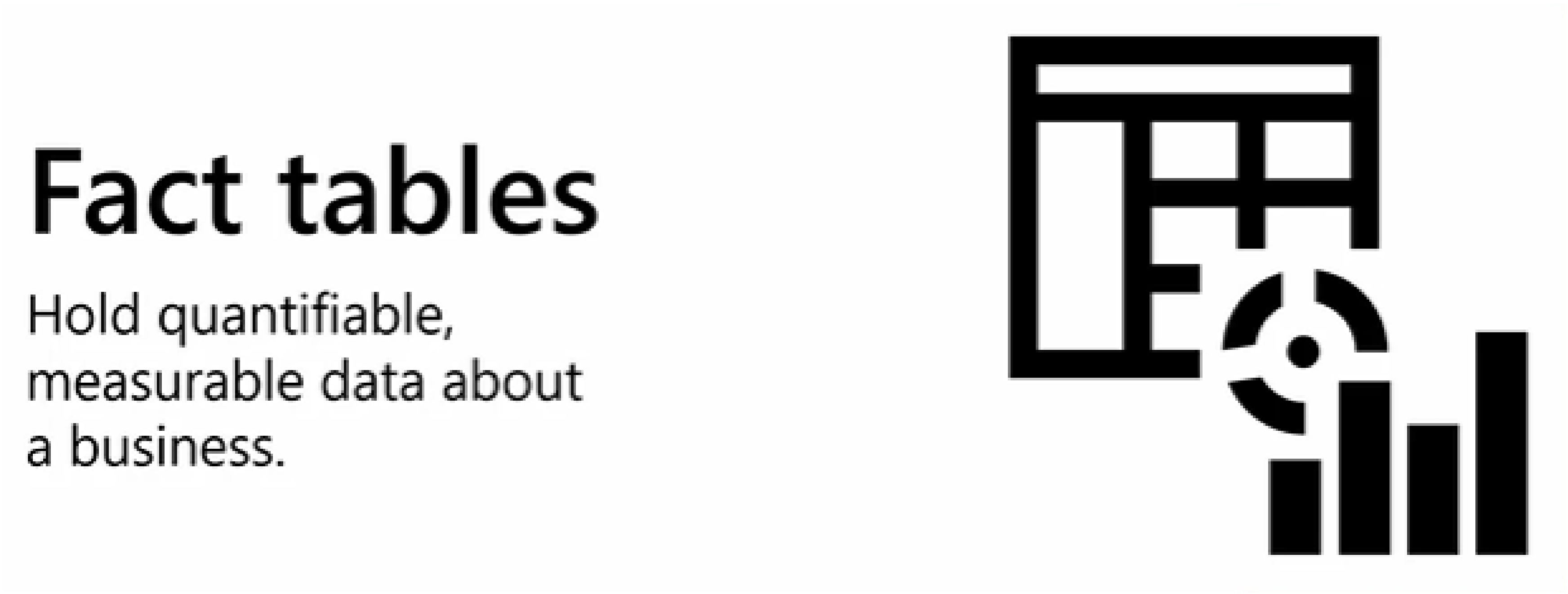
Configure table and column properties

1. Configure the table properties by renaming the table **Adventure Works Sales Data** and adding a brief description of the table in Power BI desktop.
2. Configure column properties as follows:
   - Merge queries autogenerate column names. Rename the columns as follows:
     - **Saleperson.EmployeeID** to **Employee ID**
     - **Salesperson.Salesperson** to **Salesperson**
     - **Salesperson.Title** to **Title**
   - Change and format the data types of the columns.

Quiz

- In Power BI, relationships are established between the tables based on _____________ that match between the tables.
- What are the limitations of using a Flat schema in Power BI? Select all that apply.
  - a) A Flat schema cannot be used to perform aggregations.
  - b) A Flat schema offers limited capacity for storing large volumes of data.
  - c) A Flat schema offers a lack of flexibility for organizing data from multiple sources.
- In Power BI, a schema is automatically created when you import data from various sources and establish relationships between tables. True or False?
- Which property cannot be adjusted for a table or column in Power BI?
  - a) Sort order
  - b) Data type
  - c) Table relationship

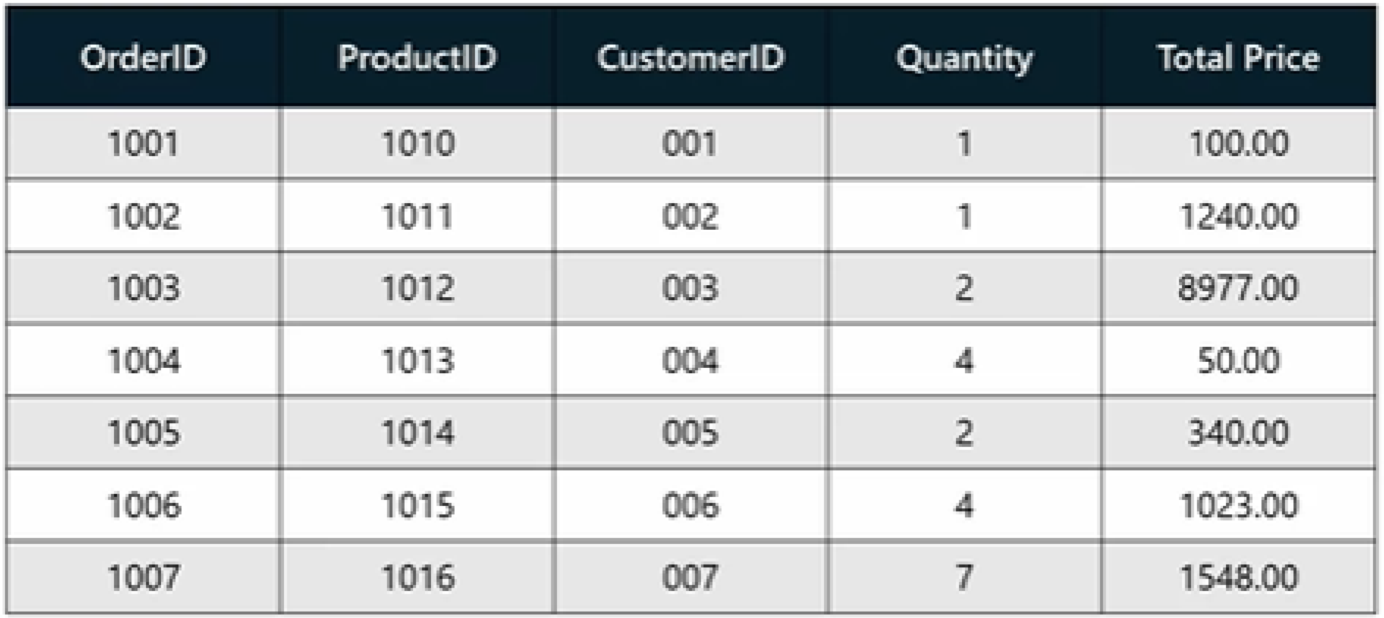
Fact and Dimension Tables



Fact tables

Hold quantifiable, measurable data about a business.

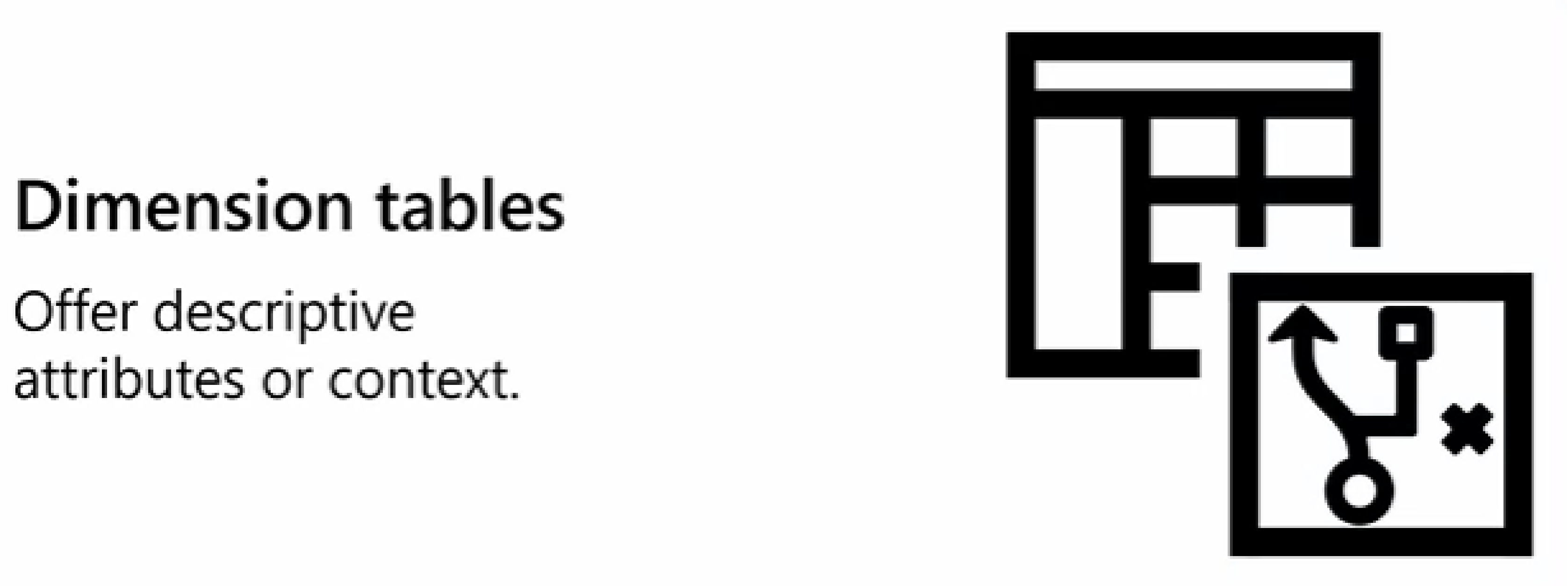
Fact and Dimension Tables (cont.)



OrderID	ProductID	CustomerID	Quantity	Total Price
1001	1010	001	1	100.00
1002	1011	002	1	1240.00
1003	1012	003	2	8977.00
1004	1013	004	4	50.00
1005	1014	005	2	340.00
1006	1015	006	4	1023.00
1007	1016	007	7	1548.00

The Transactions table contains core facts about each sale.

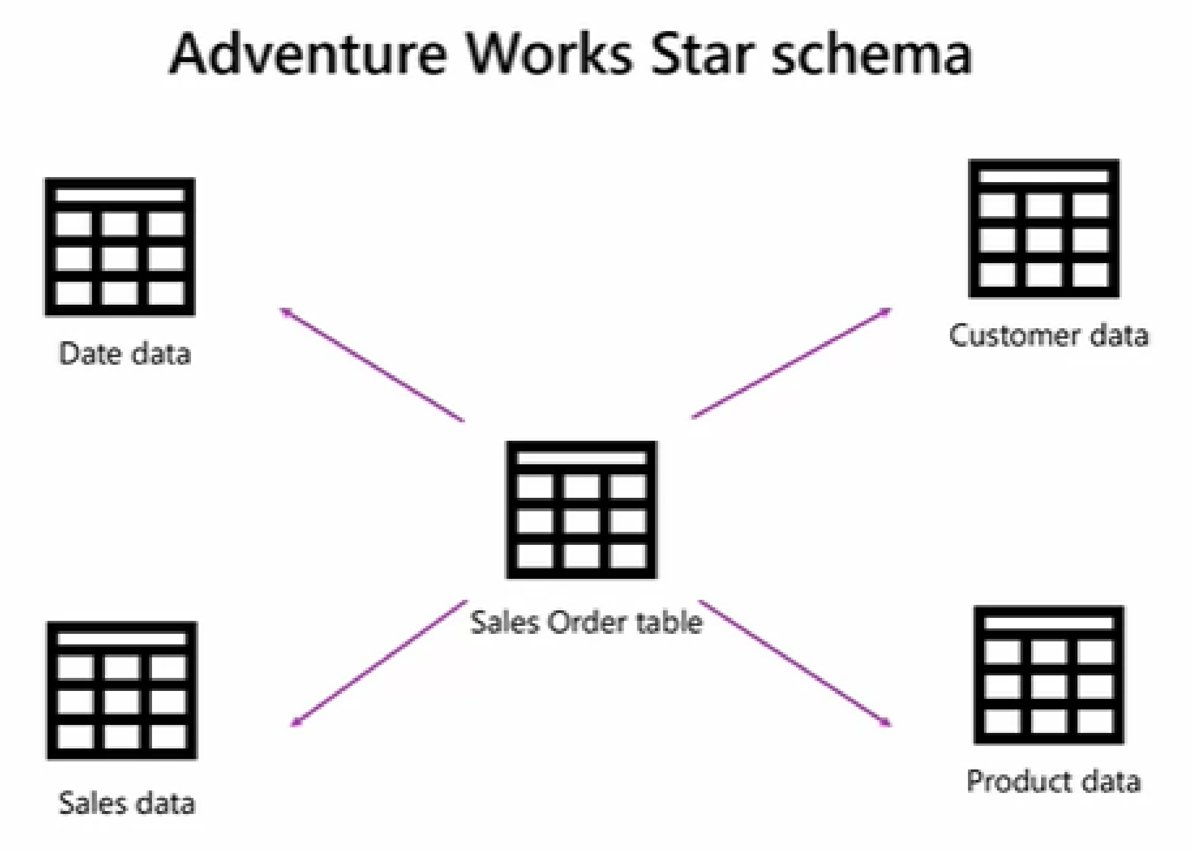
Fact and Dimension Tables (cont.)



Dimension tables

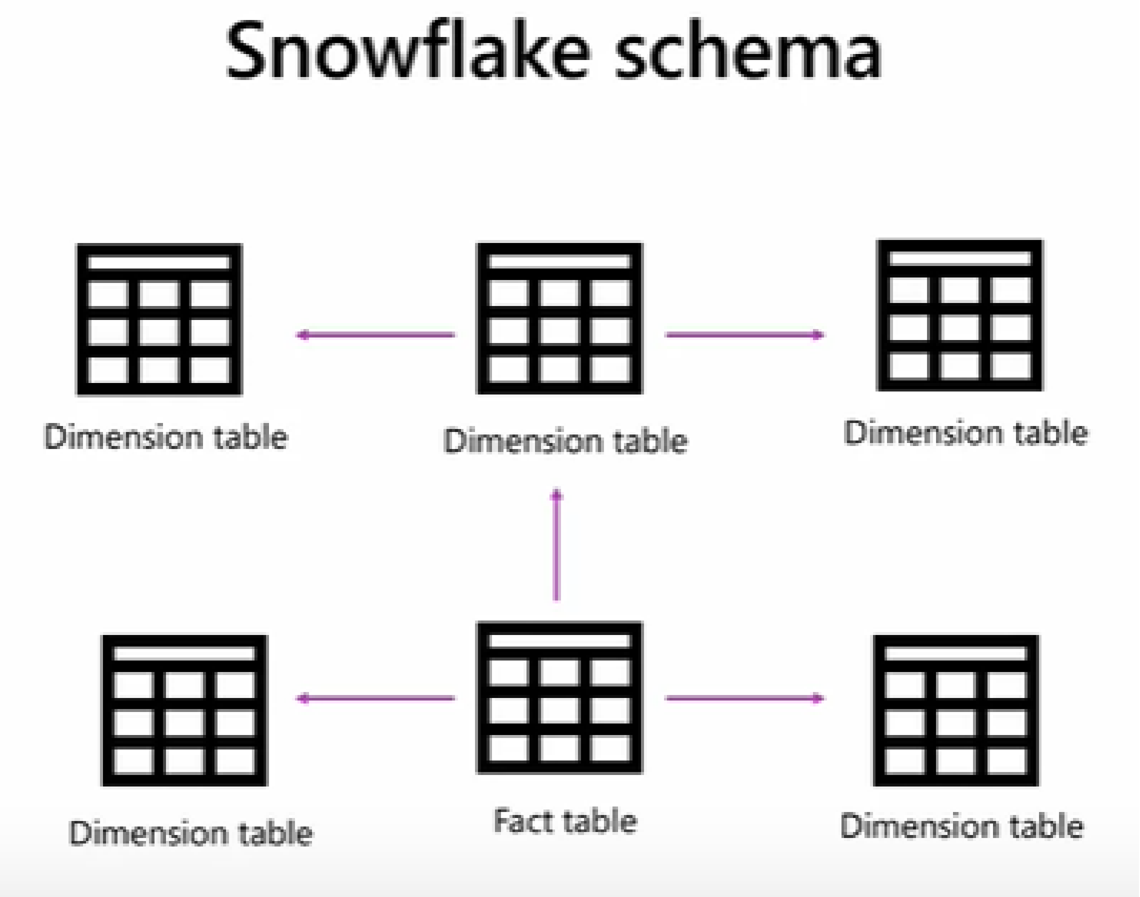
Offer descriptive attributes or context.

Fact and Dimension Tables (cont.)



The dimension tables add detailed descriptions to the fact table.

Fact and Dimension Tables (cont.)



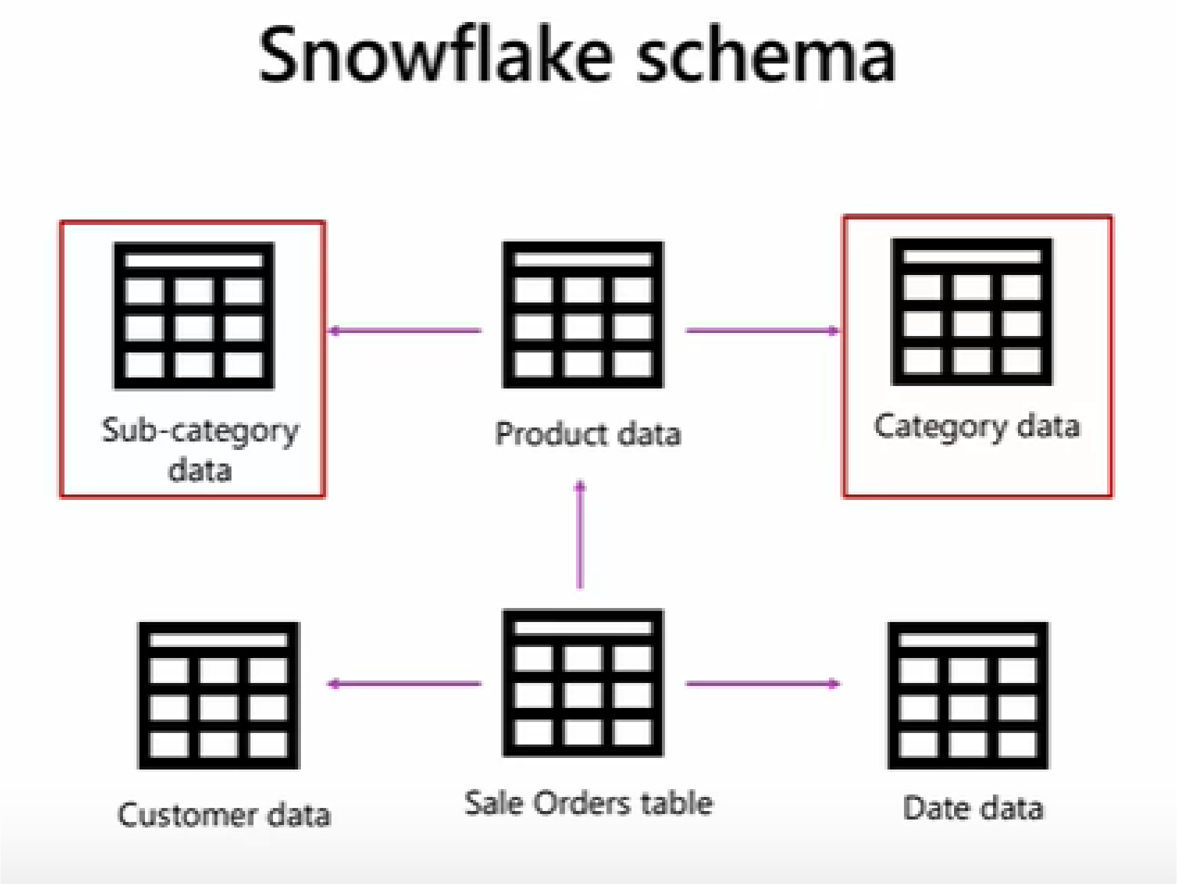
A Snowflake schema makes use of dimension tables by normalizing them.

Fact and Dimension Tables (cont.)

Normalization

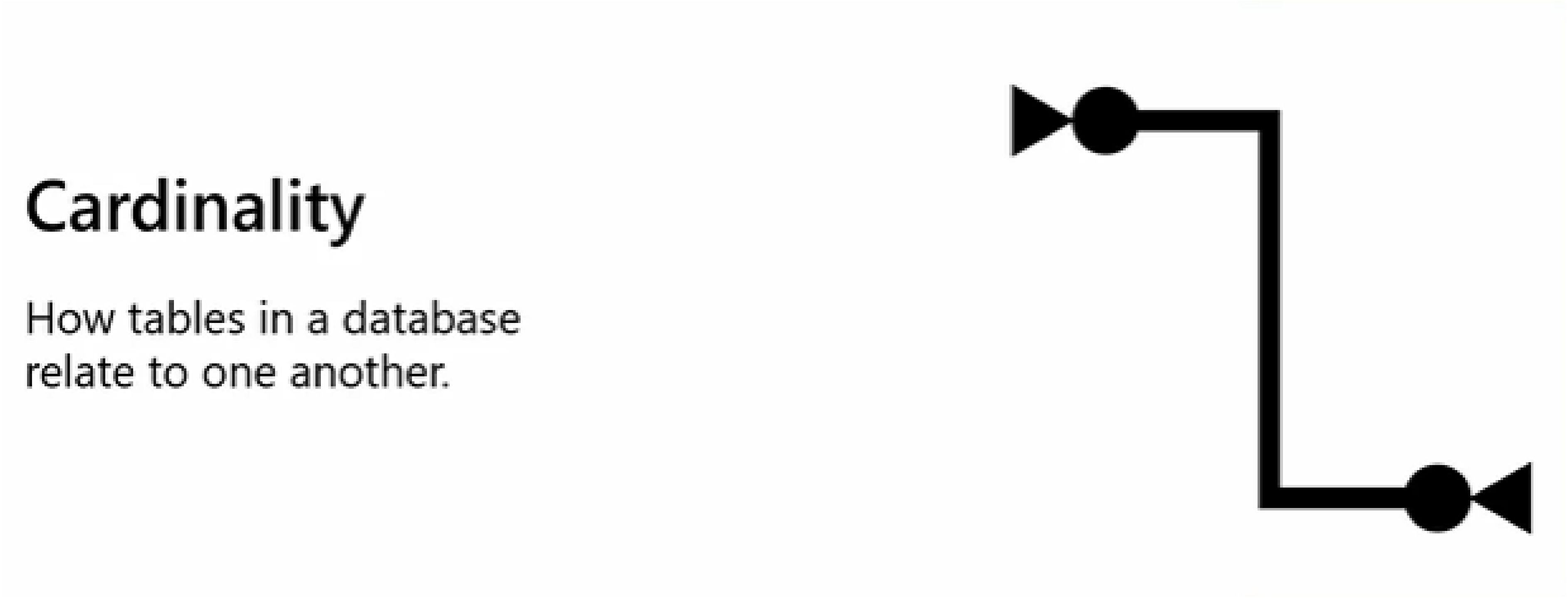
Existing schema tables are divided into additional tables.

Fact and Dimension Tables (cont.)



The Product table is normalized into Category and Sub-category tables.
This schema reduces data redundancy, but adds complexity to queries.

Cardinality

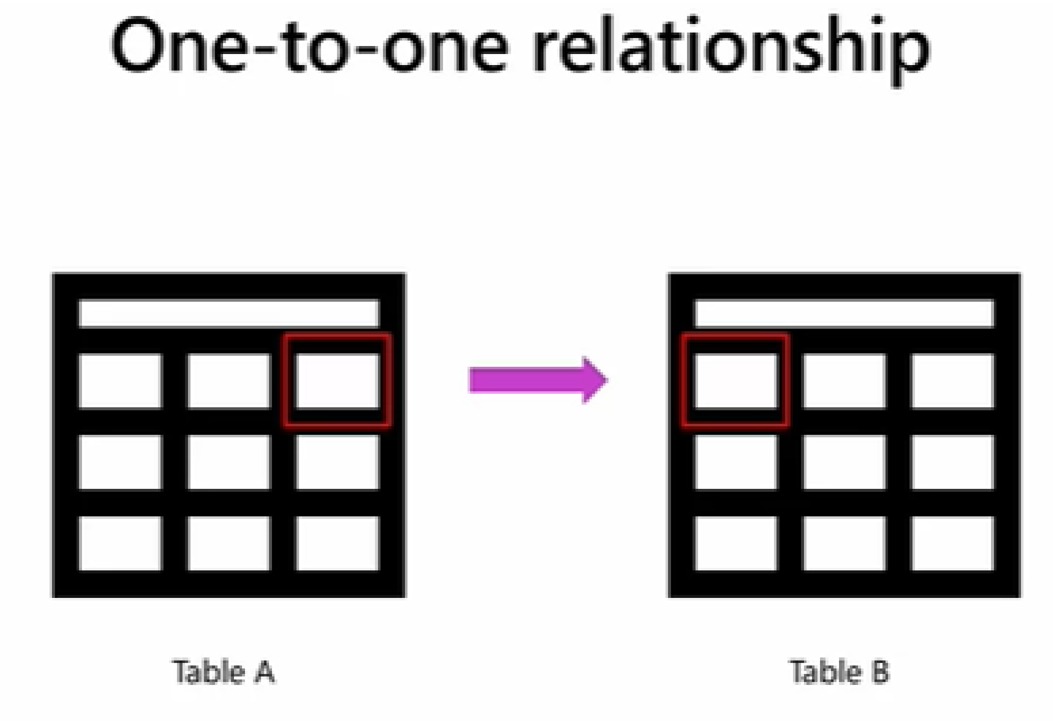


Cardinality

How tables in a database relate to one another.

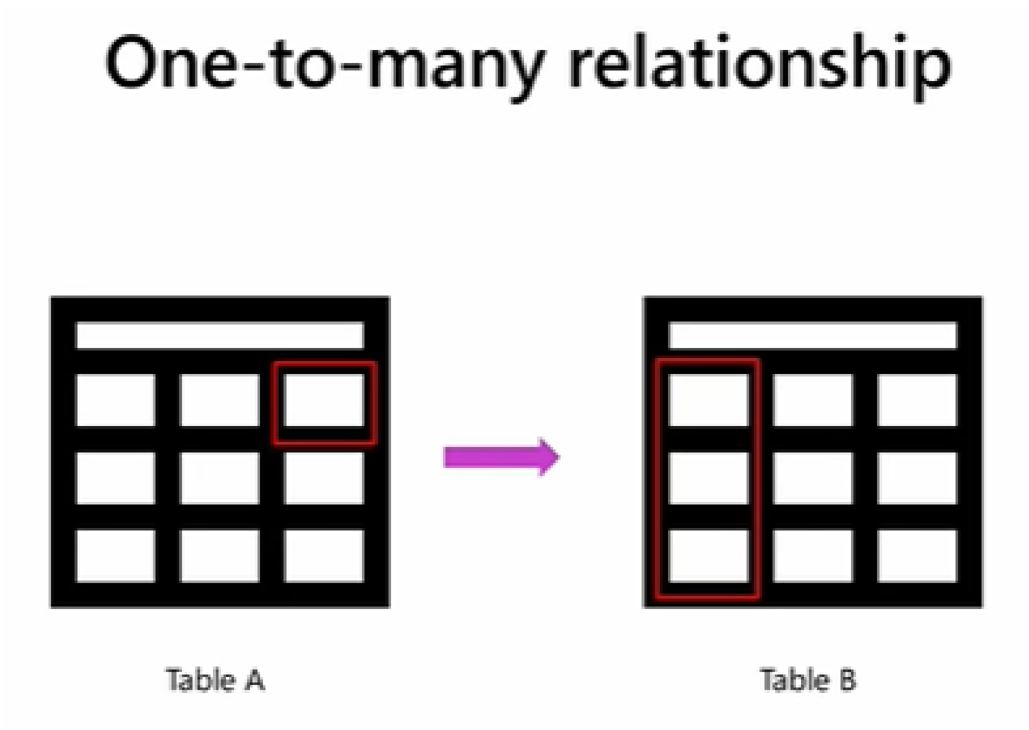
There are three types of cardinality: one-to-one, many-to-one and many-to-many

Cardinality (cont.)



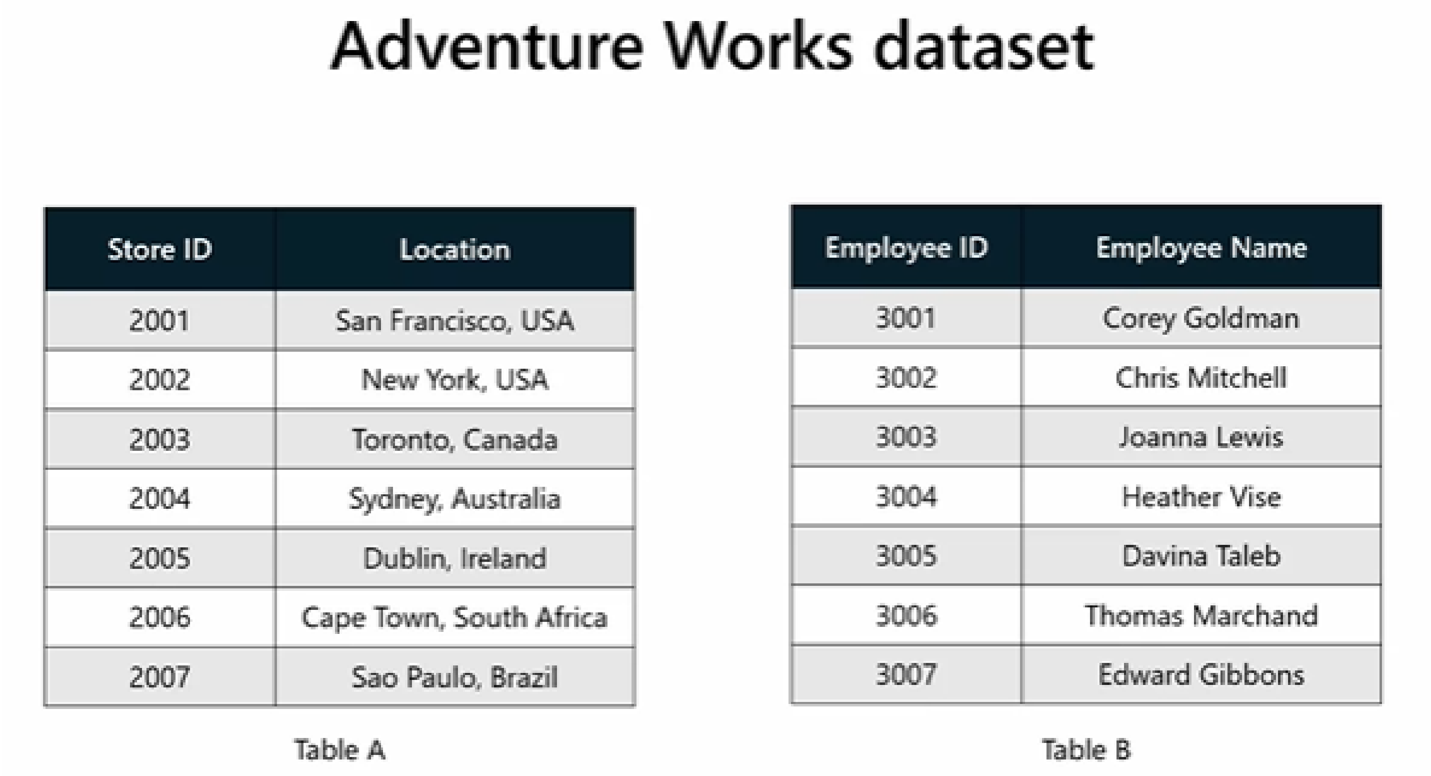
A record in one column of Table A corresponds to a unique record in one column of Table B.

Cardinality (cont.)



Each record in a column of Table A corresponds to multiple records in a column of Table B, but not the other way around.

Cardinality (cont.)



Adventure Works dataset

Store ID	Location
2001	San Francisco, USA
2002	New York, USA
2003	Toronto, Canada
2004	Sydney, Australia
2005	Dublin, Ireland
2006	Cape Town, South Africa
2007	Sao Paulo, Brazil

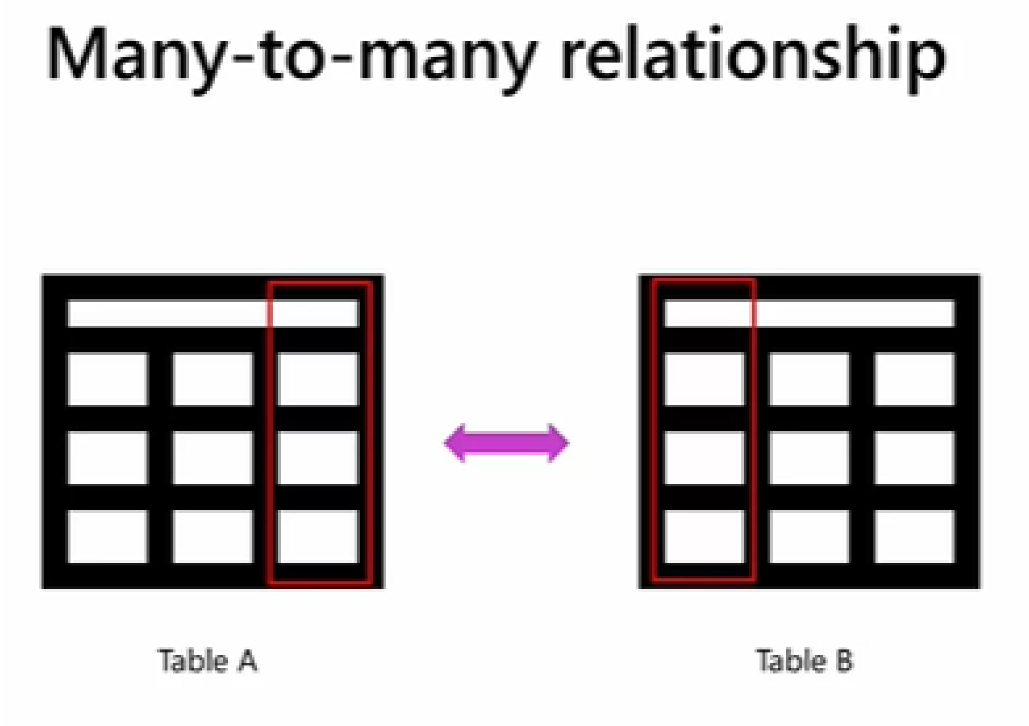
Table A

Employee ID	Employee Name
3001	Corey Goldman
3002	Chris Mitchell
3003	Joanna Lewis
3004	Heather Vise
3005	Davina Taleb
3006	Thomas Marchand
3007	Edward Gibbons

Table B

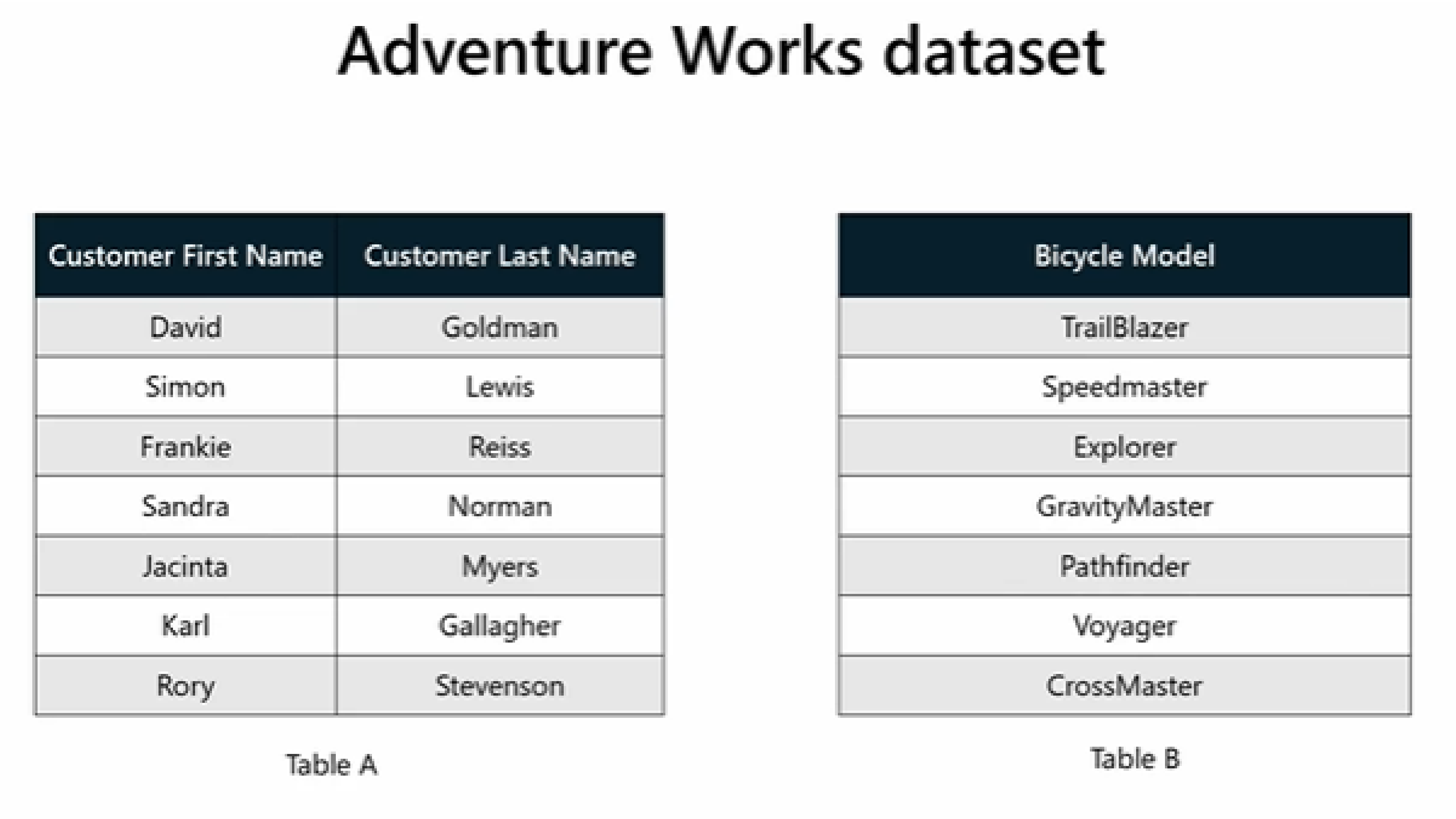
The relationship between the stores and their employees is a one-to-many relationship. This is because each employee works for one store, but each store has many employees.

Cardinality (cont.)



Multiple records in a column of Table A are related to multiple records in a column of Table B in both directions.

Cardinality (cont.)



Adventure Works dataset

Customer First Name	Customer Last Name
David	Goldman
Simon	Lewis
Frankie	Reiss
Sandra	Norman
Jacinta	Myers
Karl	Gallagher
Rory	Stevenson

Table A

Bicycle Model
TrailBlazer
Speedmaster
Explorer
GravityMaster
Pathfinder
Voyager
CrossMaster

Table B

A customer can purchase many different bicycle models logged in Table B.
Each bicycle model can be purchased by multiple customers recorded in Table A.

Quiz

- In the context of Power BI, which of the following descriptions best outlines the main purpose of a Fact table?
  a) Storing detailed, transactional business data.
  b) Storing measured, quantitative data about a business process.
  c) Storing descriptive attributes of business dimensions.
- What is the role of dimension tables in Power BI?
  a) They store the descriptive attributes of a business process.
  b) They store transactional data related to a business process.
  c) They store measured, quantitative data about a business process.

Exercise: Configuring a Star Schema

Download the Excel file and disable auto-detect relationships

1. Download and save the Excel workbook **AdventureWorksData.xlsx**. The workbook contains four tables of data: **Sales**, **Products**, **Region**, and **Salesperson**.
2. Disable Power BI's **auto-detect** function. (File -> Options and Settings -> Options -> Current File -> Data Load -> Uncheck Autodetect Relationships.)

Load the data from the Excel workbook

1. Load the data from the Excel sheet into Power BI. Ensure you load all tables in the workbook.
2. Open a preview of the table in the **Preview** pane.

Exercise: Configuring a Star Schema (cont.)

Configure a Star Schema

1. Once the data is loaded to Power BI, Identify the main fact table in your dataset and determine the unique identifier for the fact table.
2. Identify the related dimension tables within your dataset and determine the unique identifiers for each dimension table.
3. Establish relationships between the fact and dimension tables based on the common fields.
4. Next, configure the relationships. For example, establish cardinality type and cross-filter direction between the Fact and dimension tables.
5. Review the data type and formats of the columns and adjust them if necessary. Apply any necessary transformation to prepare the data for analysis, if needed (Note that this step is optional).

Quiz

- Which of the following is true about a Star Schema fact table?
  a) A Fact table stores an accumulation of business events.
  b) A Fact table stores an accumulation of business entities.
  c) A Fact table must have a unique column.

- How are dimension tables structured in a Snowflake Schema, as opposed to a Star Schema?
  a) They are fully de-normalized, with all attributes in a single table.
  b) They are normalized with a separate table for each attribute.
  c) They are connected in a hierarchical structure with multiple levels.

Quiz (cont.)

- What is the primary benefit of normalizing dimension tables in Power BI?
  a) It simplifies data querying and reporting.
  b) It reduces storage requirements.
  c) It improves data quality and accuracy.

- Which of the following statements is true about relationships in Power BI?
  a) Relationships can only be created between columns that contain the same data type.
  b) Relationships can be created between tables that contain different types of data.
  c) Relationships can only be created between tables with the same number of rows.

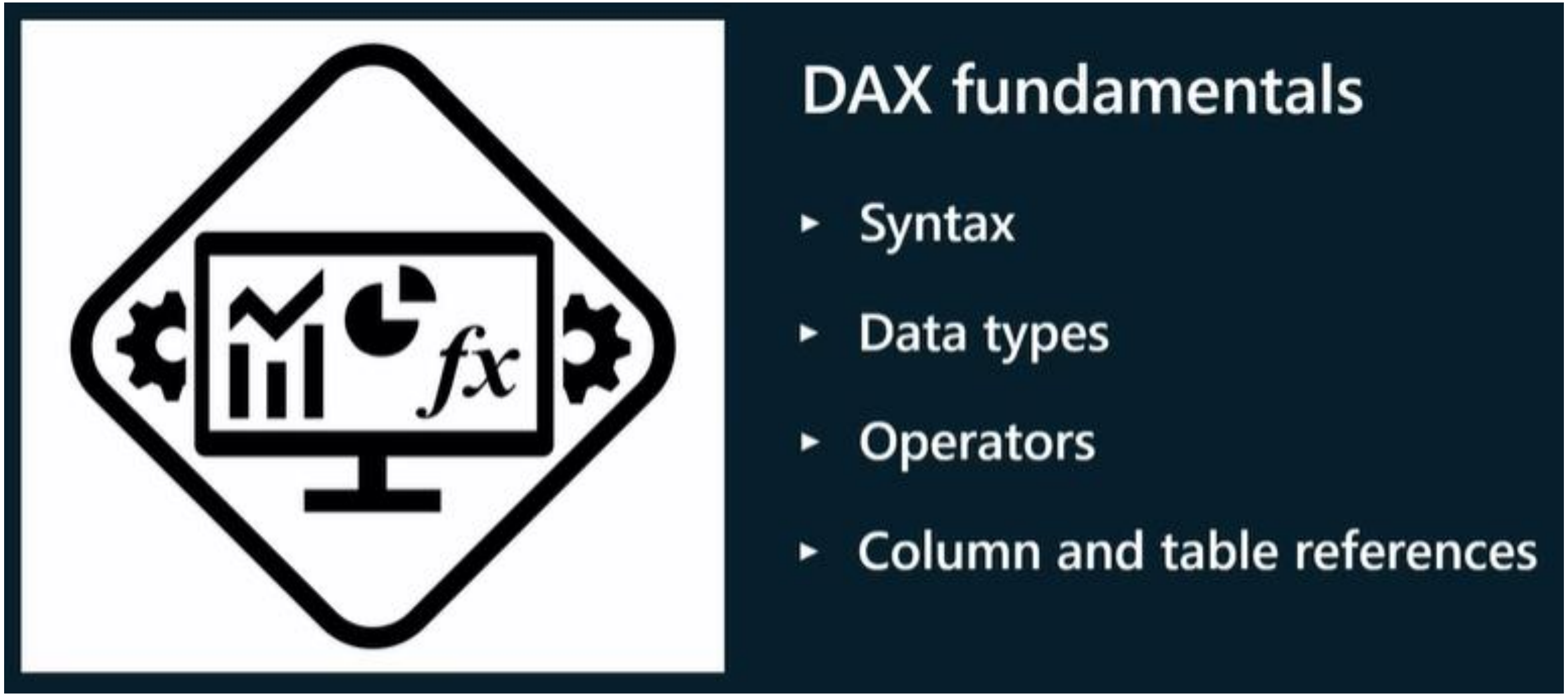
- A Star schema is more suitable for complex hierarchies and relationships. True or False?

Data Analysis Expressions

DAX

Adds new information about existing data.

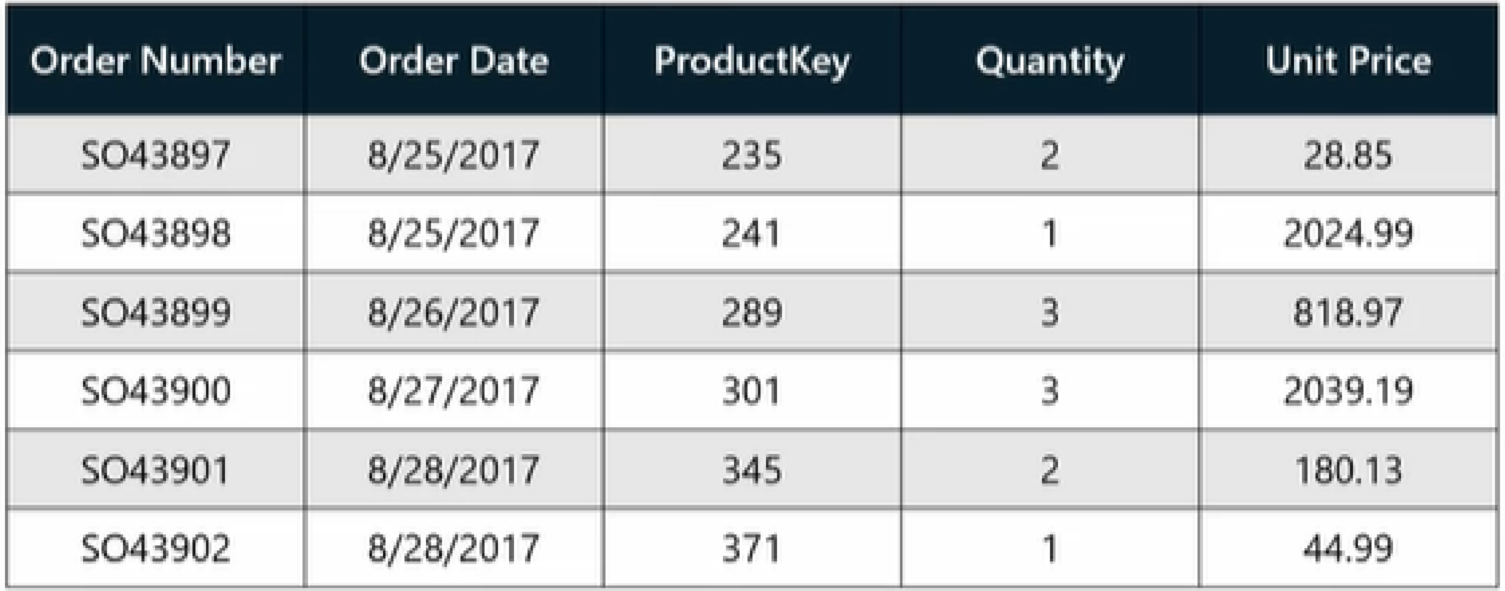
Data Analysis Expressions (cont.)



DAX fundamentals

- Syntax
- Data types
- Operators
- Column and table references

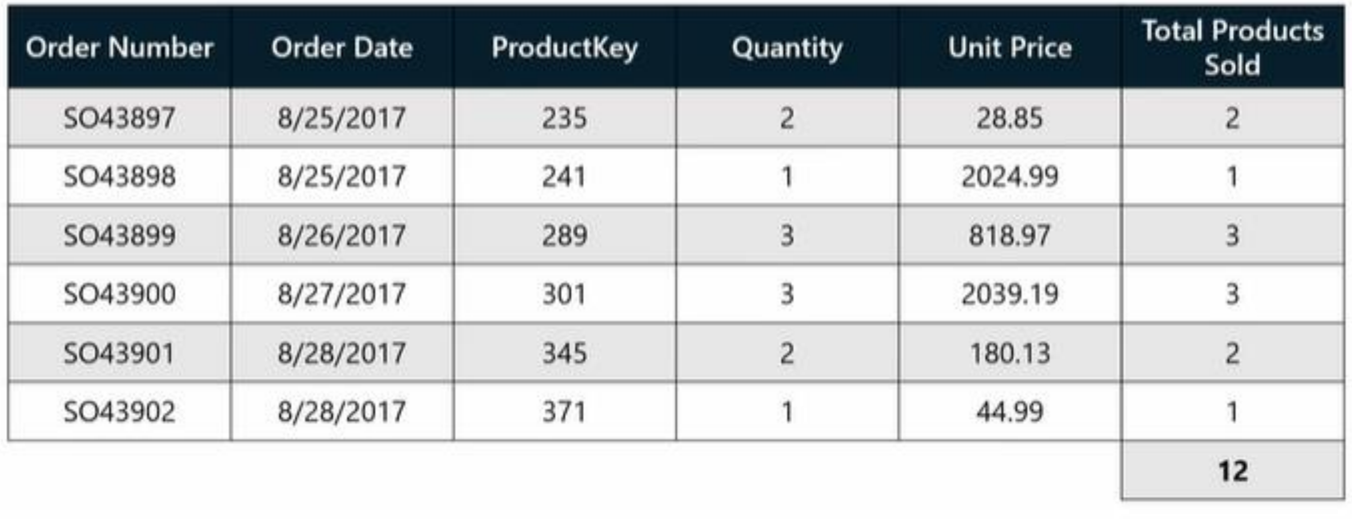
Data Analysis Expressions (cont.)



Order Number	Order Date	ProductKey	Quantity	Unit Price
SO43897	8/25/2017	235	2	28.85
SO43898	8/25/2017	241	1	2024.99
SO43899	8/26/2017	289	3	818.97
SO43900	8/27/2017	301	3	2039.19
SO43901	8/28/2017	345	2	180.13
SO43902	8/28/2017	371	1	44.99

The Adventure Works sales table doesn't include any data that denotes the total number of products sold. The company could generate this data using DAX.

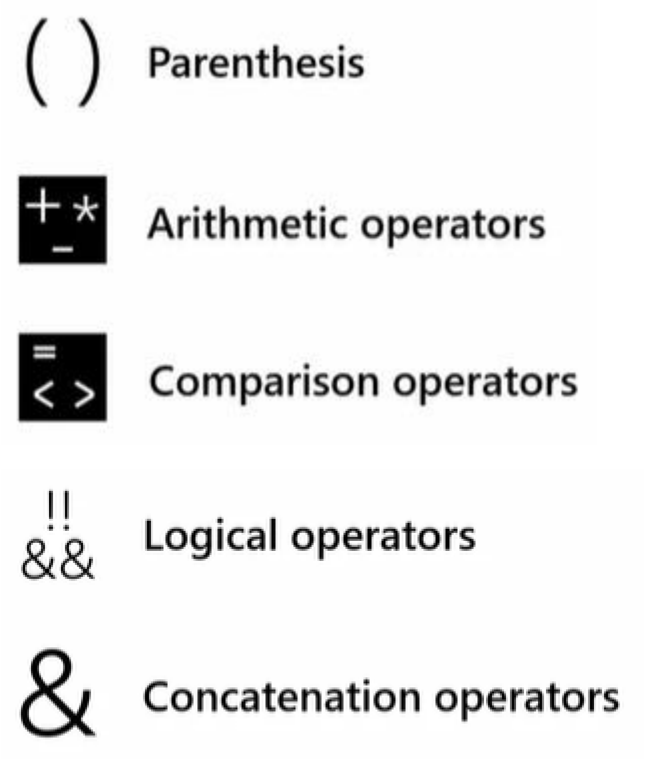
Data Analysis Expressions (cont.)



Order Number	Order Date	ProductKey	Quantity	Unit Price	Total Products Sold
SO43897	8/25/2017	235	2	28.85	2
SO43898	8/25/2017	241	1	2024.99	1
SO43899	8/26/2017	289	3	818.97	3
SO43900	8/27/2017	301	3	2039.19	3
SO43901	8/28/2017	345	2	180.13	2
SO43902	8/28/2017	371	1	44.99	1
					12

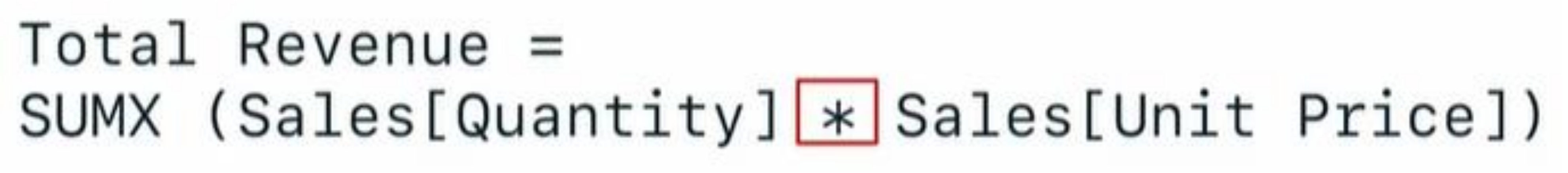
```
Total Products Sold = SUM ('Sales'[Quantity])
```

DAX Operators



( )	Parenthesis
+ * −	Arithmetic operators
= < >	Comparison operators
!! &&	Logical operators
&	Concatenation operators

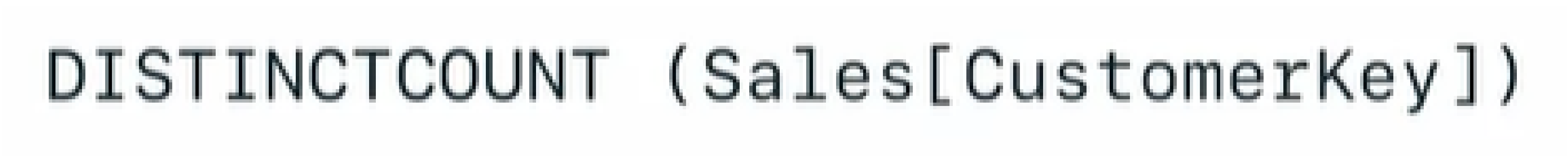
DAX Operators (cont.)



```
Total Revenue =
SUMX (Sales[Quantity] * Sales[Unit Price])
```

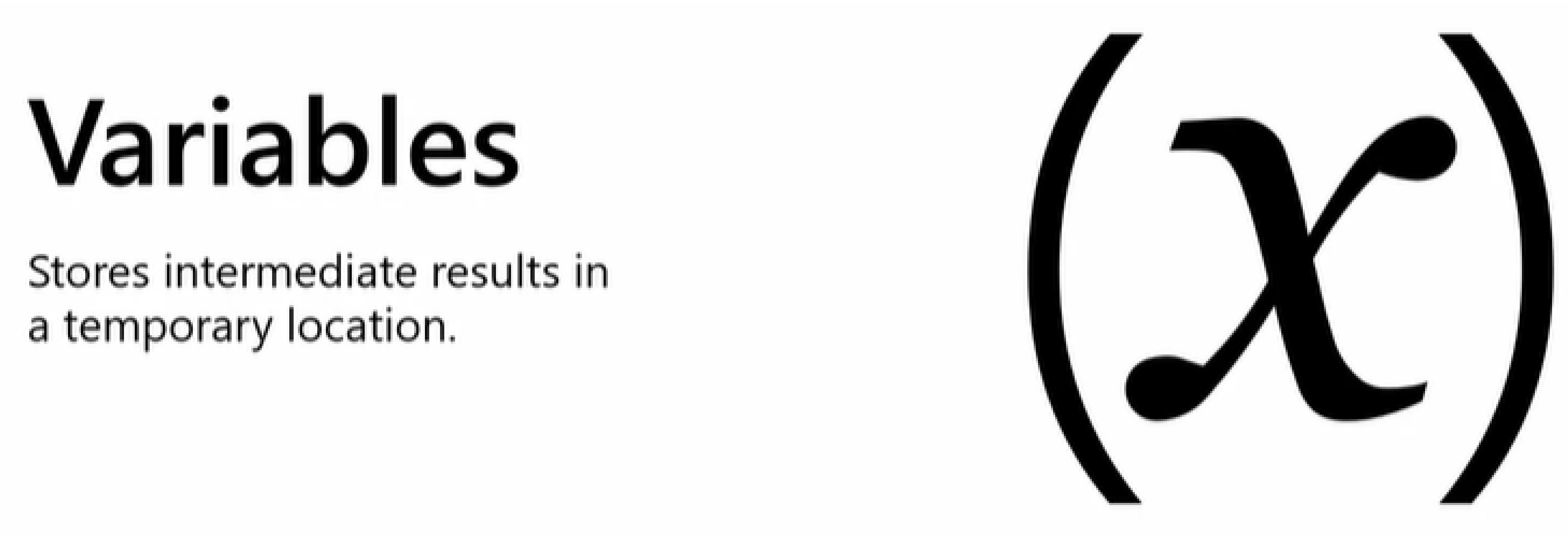
Formulas and Functions in DAX

Formulas and Functions in DAX (cont.)



```
DISTINCTCOUNT (Sales[CustomerKey])
```

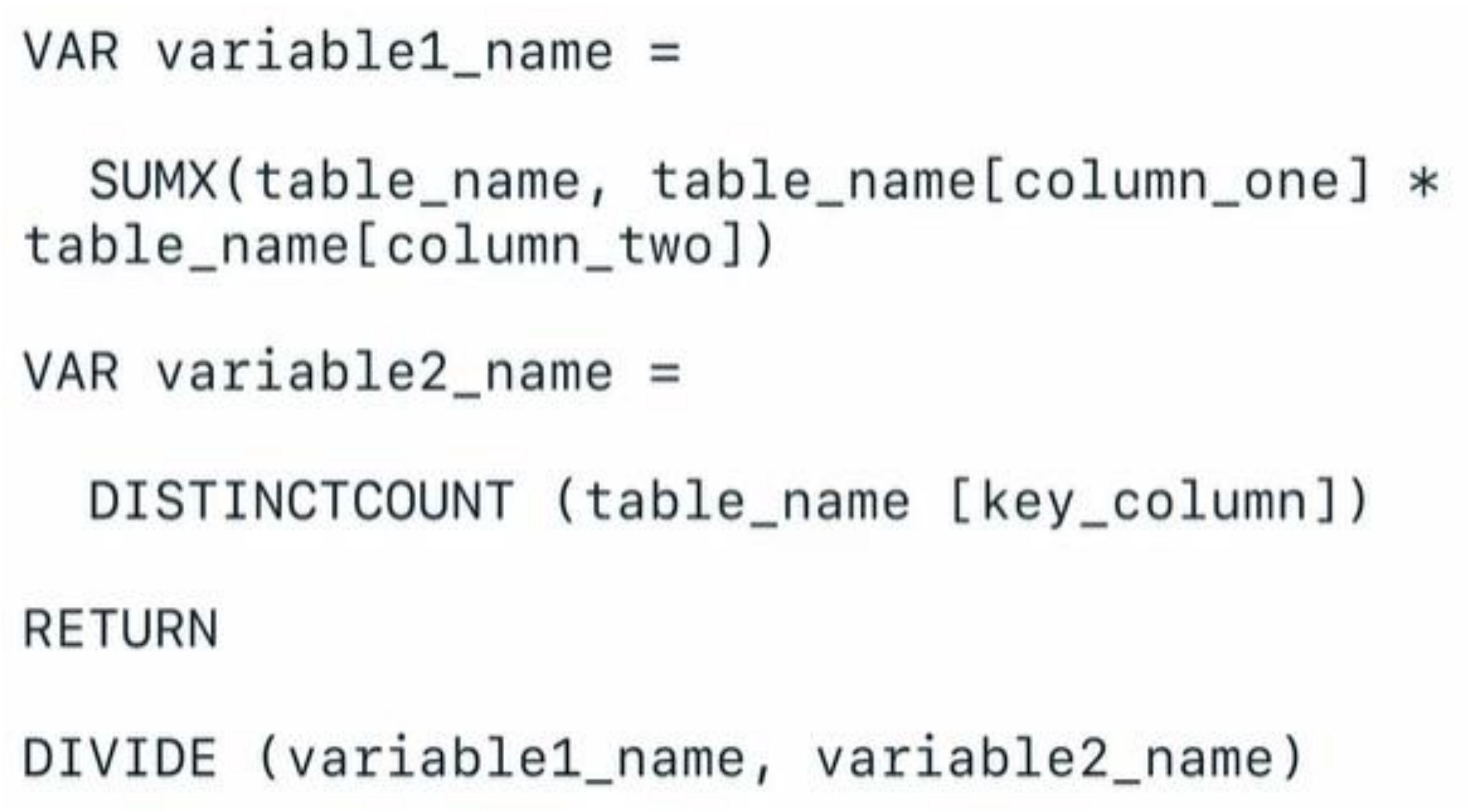
Formulas and Functions in DAX (cont.)



Variables

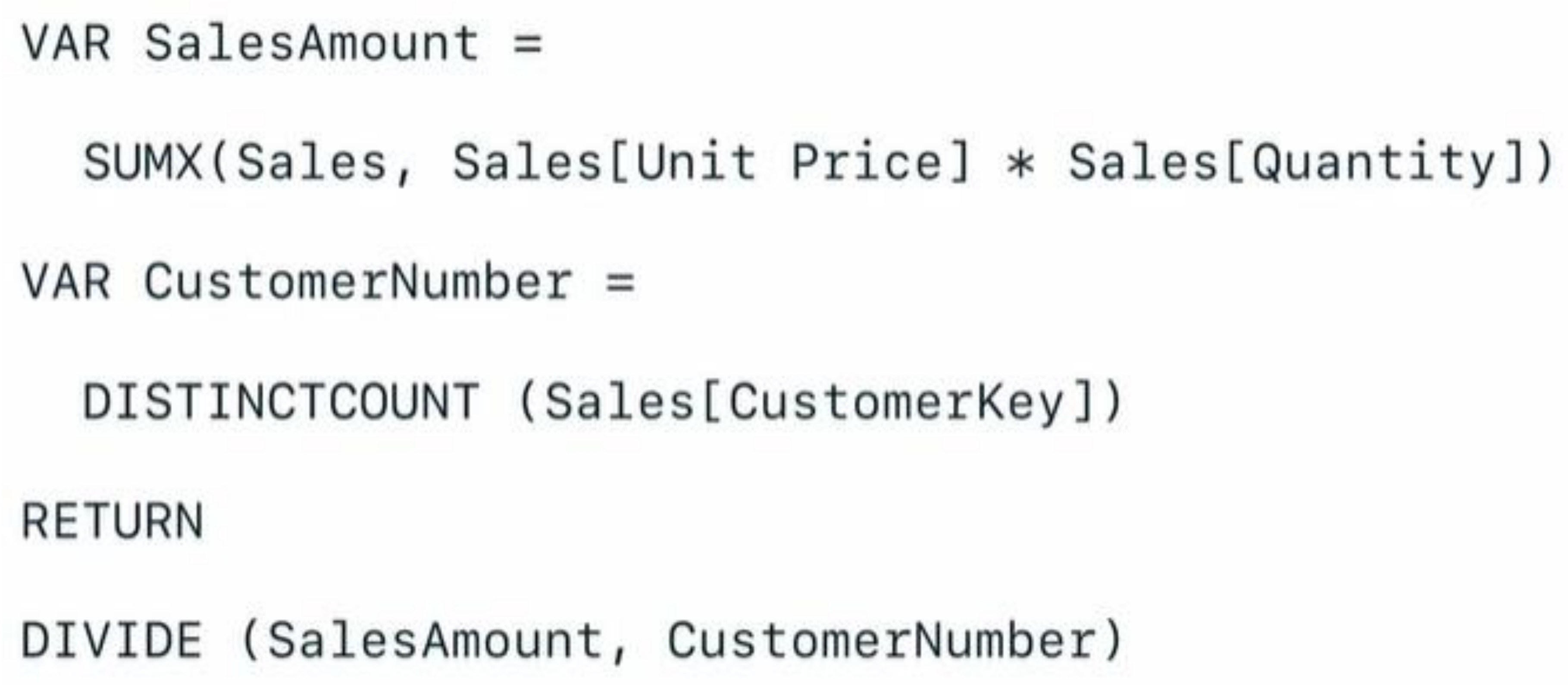
Stores intermediate results in a temporary location.

Formulas and Functions in DAX (cont.)



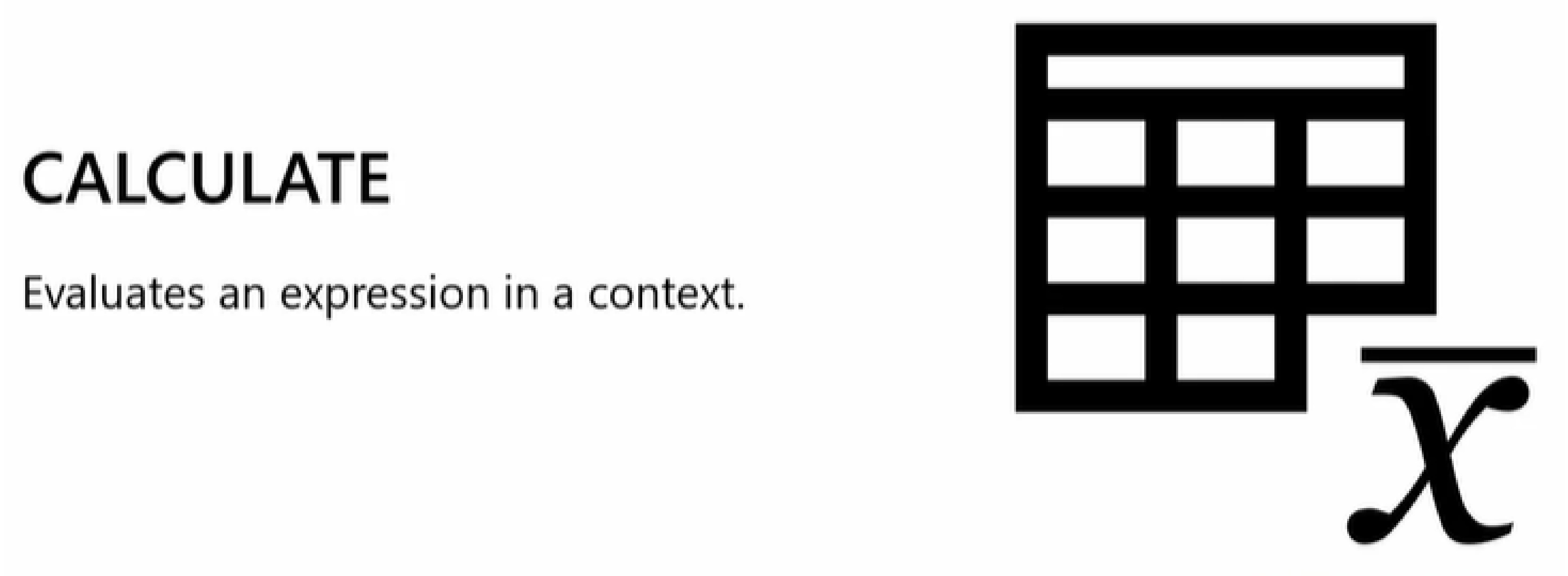
```
VAR variable1_name =

  SUMX(table_name, table_name[column_one] *
table_name[column_two])

VAR variable2_name =

  DISTINCTCOUNT (table_name [key_column])

RETURN

DIVIDE (variable1_name, variable2_name)
```

Formulas and Functions in DAX (cont.)



```
VAR SalesAmount =

  SUMX(Sales, Sales[Unit Price] * Sales[Quantity])

VAR CustomerNumber =

  DISTINCTCOUNT (Sales[CustomerKey])

RETURN

DIVIDE (SalesAmount, CustomerNumber)
```

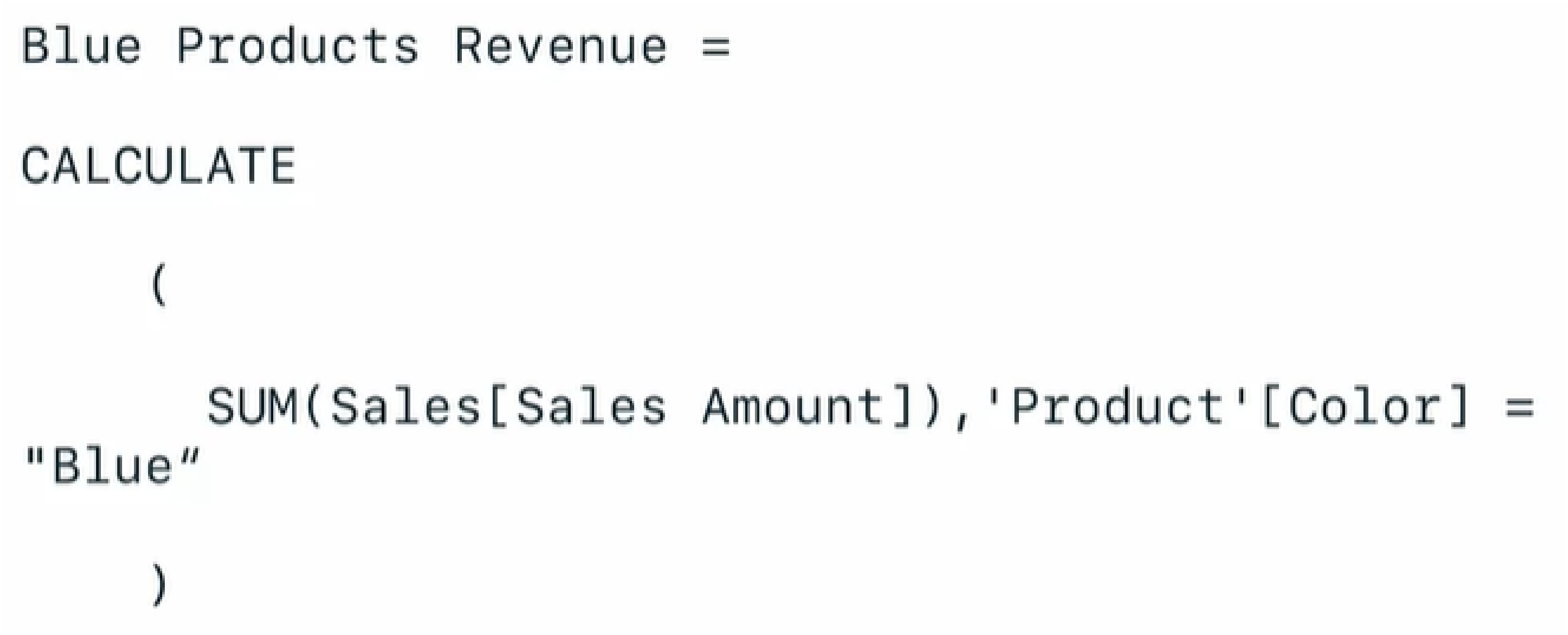
Formulas and Functions in DAX (cont.)



CALCULATE

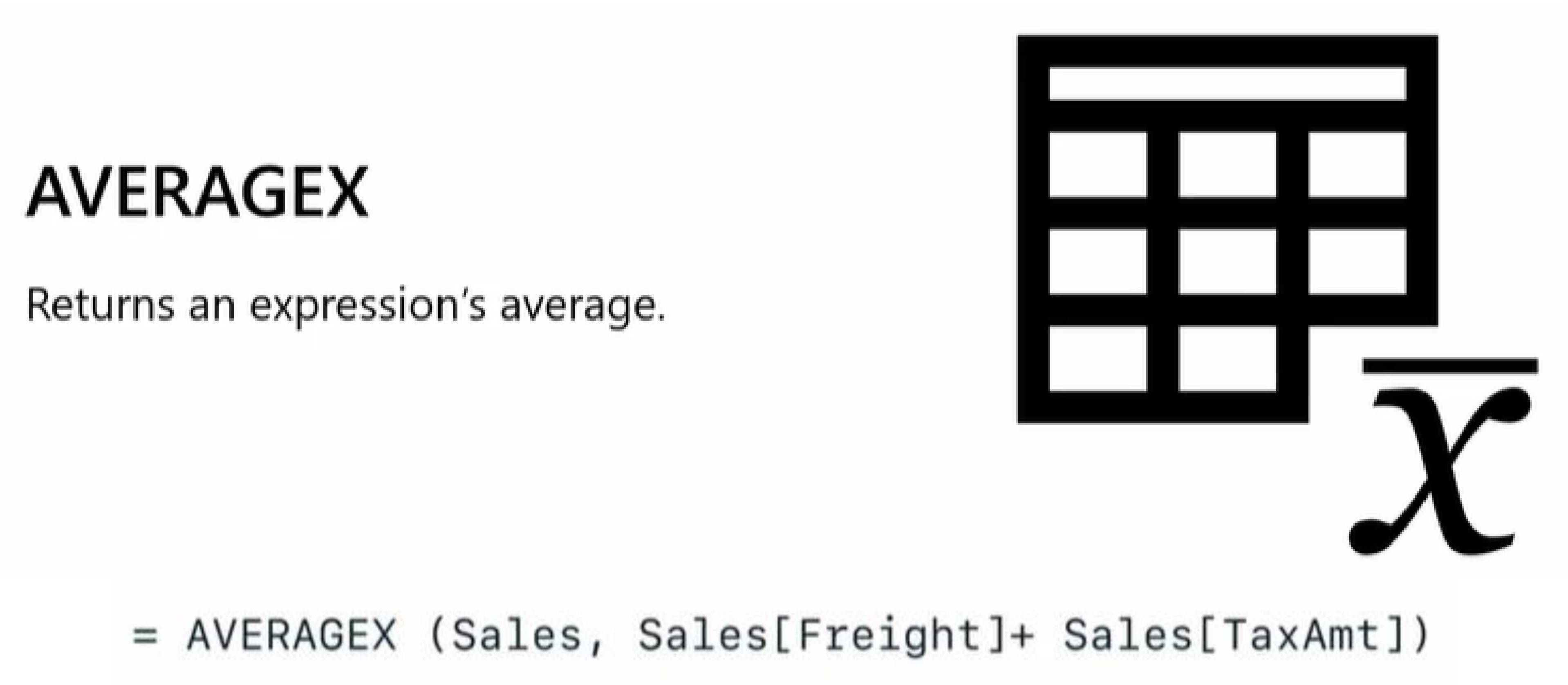
Evaluates an expression in a context.

Formulas and Functions in DAX (cont.)



```
Blue Products Revenue =

CALCULATE

    (

      SUM(Sales[Sales Amount]),'Product'[Color] =
"Blue”

    )
```

Formulas and Functions in DAX (cont.)

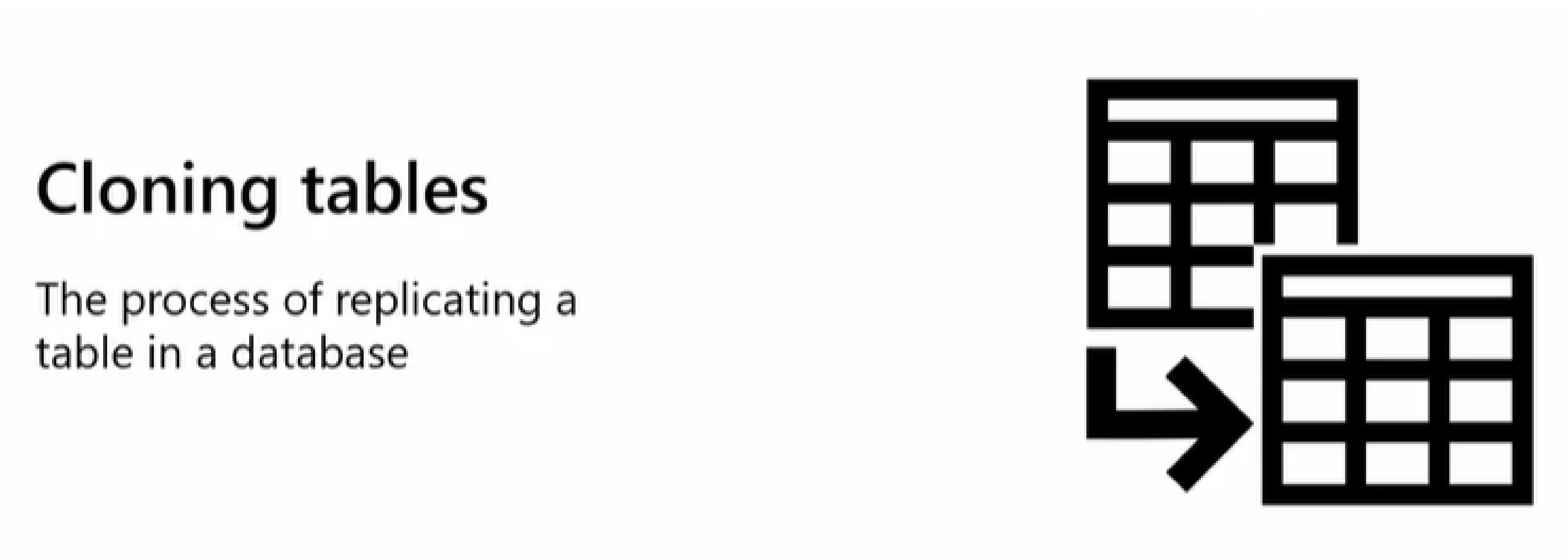


AVERAGEX

Returns an expression's average.

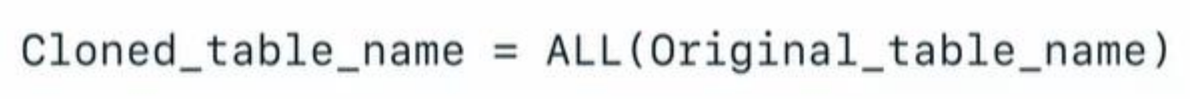
```
= AVERAGEX (Sales, Sales[Freight]+ Sales[TaxAmt])
```

Calculated Tables



Cloning tables

The process of replicating a table in a database



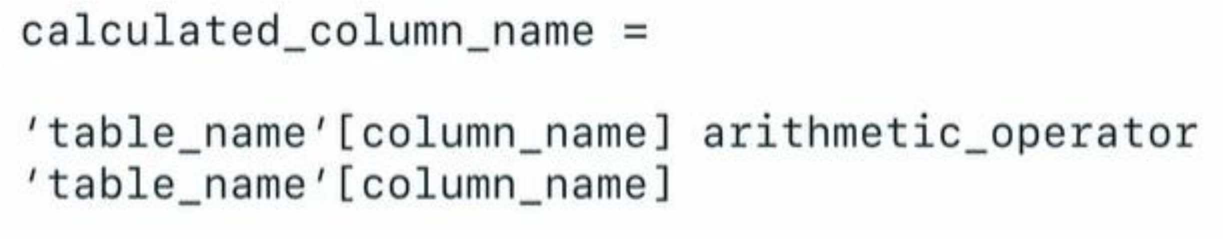
```
Cloned_table_name = ALL(Original_table_name)
```

Calculated Columns

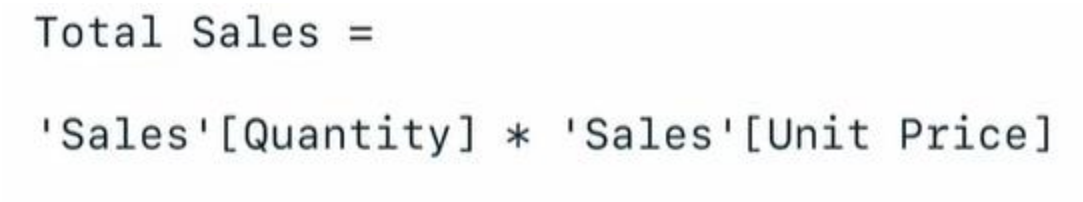
Calculated columns

New column in an existing database table.

Calculated Columns – syntax



```
calculated_column_name =

’table_name’[column_name] arithmetic_operator
’table_name’[column_name]
```



```
Total Sales =

'Sales'[Quantity] * 'Sales'[Unit Price]
```

Exercise: Adding a calculated table and column

Download and connect to the Adventure Works Dataset

1. Download and save the Excel workbook **AdventureWorksData.xlsx**. The workbook contains five data tables: **Sales**, **Date**, **Products**, **Region**, and **Salesperson**.
2. Load the data from the Excel sheet into Power BI. Ensure you load all tables in the workbook.
3. Open a preview of the table in the **Preview** pane.

Remove all duplicate values and set the relationships between the tables

1. Remove all duplicate values from the **SalesOrderNumber** column of the **Sales** fact table.
2. Ensure that one-to-many relationships exist between the fact table and all dimension tables. Once you load the data, Power BI will establish the table relationships automatically. If any relationship is missing, create it manually.

Exercise: Adding a calculated table and column (cont.)

Create a calculated table

Create a new calculated table called **Yearly Sales by Color** that contains the following data:

- All data from the **Sales** table,
- All **Product Color** data from the **Product** table,
- And all Yearly values from the **Date** table.

Tip: Create the calculated table using the **ADDCOLUMNS** and **RELATED** DAX functions

Exercise: Adding a calculated table and column (cont.)

Create calculated columns

1. Create a calculated column in the **Date** table called **Qtr** and populate it with data for each quarter of the year. **Hint:** Look up the function **QUARTER** in the cheat-sheet.
2. Create a second calculated column in the **Date** table called **Month** and populate it with the name of each month (Display each month's name as just the first three letters of each month's name). **Hint:** Look up the function **LEFT** in the cheat-sheet.
3. Create a calculated column in the **Sales** table for **Product Color**.

Tip: You can create columns using the **New column** feature from the table tools tab of Power BI's desktop interface, along with the **RELATED**, **MONTH**, and **QUARTER** DAX functions. You can use the **MONTH DAX** function to display each month's name.