

Power Query, Star Schema & Data Modelling (Adventure Cycles)

Prepare and model data the professional way. Clean data in Power Query, load a star schema (fact and dimension tables), create one-to-many relationships, understand filter direction, build and mark a date table, create hierarchies, and avoid the most common modelling mistakes, using the Adventure Cycles case study.

● Power BI class 3

✦ Power BI Desktop 2026

Intermediate

🕒 140 min

By the end of this lesson you can

- Apply common Power Query steps and read the Applied Steps and M code
- Explain fact tables, dimension tables and the star schema, and why they beat one wide table
- Create relationships with the right cardinality and filter direction, and explain how filters flow
- Build a date table, mark it as a date table and add hierarchies
- Diagnose common modelling problems (many-to-many, ambiguous paths, missing dates)

Datasets used

adventure_cycles/fact_sales.csv

adventure_cycles/dim_product.csv

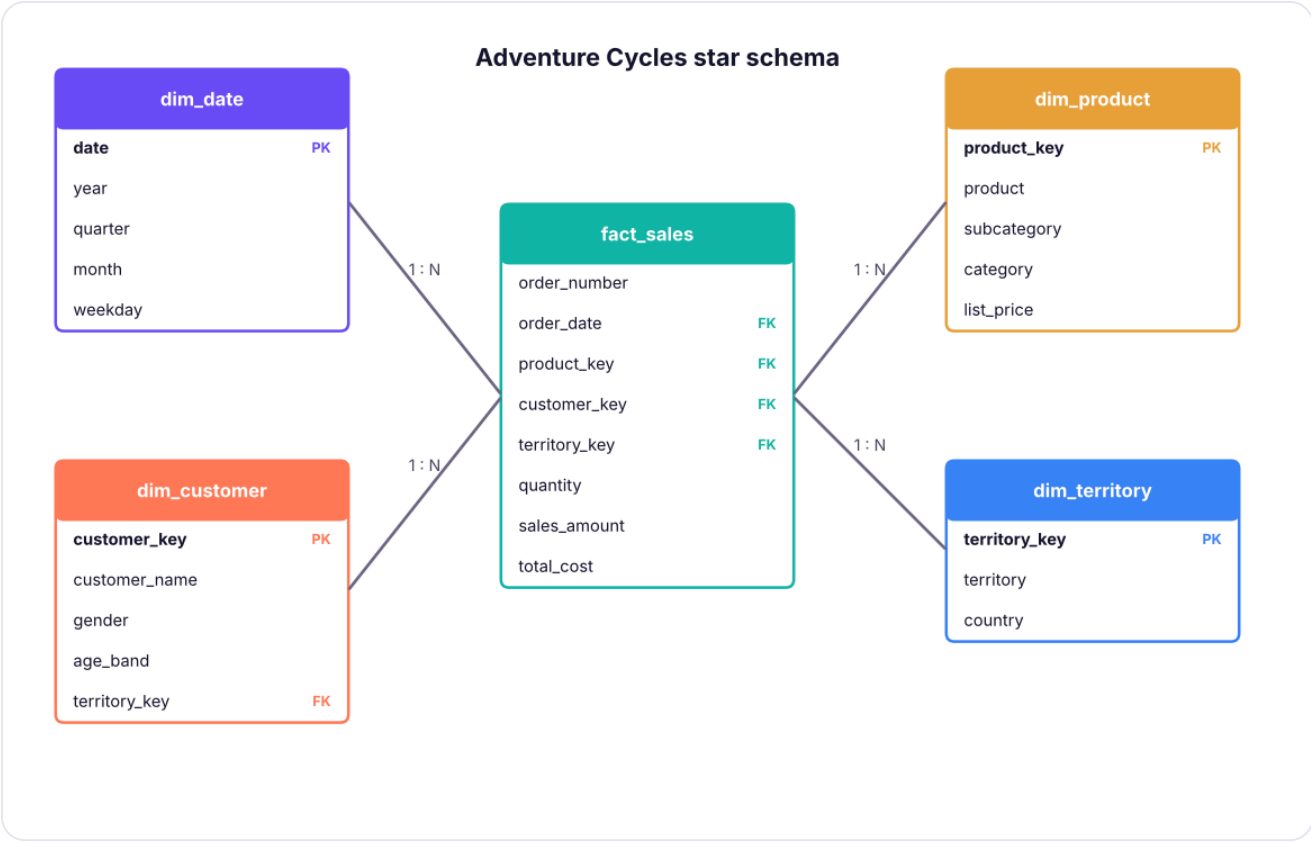
adventure_cycles/dim_customer.csv

adventure_cycles/dim_territory.csv

adventure_cycles/dim_date.csv

Why modelling matters more than visuals

Beautiful visuals on a bad model give wrong or slow answers. In Power BI the **data model** decides what numbers mean: how tables relate, which table filters which, and how fast DAX runs. The golden rule: **model as a star schema**.



A star schema: one central fact table of events/numbers surrounded by descriptive dimension tables. Filters flow from dimensions (one side) to the fact (many side).

	Fact table	Dimension table
Describes	Events or transactions (a sale, a click, a shipment)	The "who, what, where, when" of those events
Rows	Many (millions)	Fewer (thousands)
Columns	Keys + numeric measures (quantity, amount, cost)	Keys + descriptive attributes (name, category, city)
Changes	Grows constantly	Changes rarely
Example	fact_sales	dim_product , dim_customer , dim_date

Why a star beats one big flat table: smaller file (text stored once), faster visuals, one place to fix a name, correct filtering across several facts, and simpler DAX. A **snowflake** (dimensions normalised into sub-dimensions) works but adds joins; prefer a star in Power BI.

The Adventure Cycles dataset

A fictional bicycle company sells bikes, components, clothing and accessories in India, the UAE and Singapore. The model has one fact table and four dimensions (plus the date table):

Python

```
import pandas as pd
D = "adventure_cycles/"
fact = pd.read_csv(D + "fact_sales.csv", parse_dates=["order_date"]); prod = pd.read_csv(D
+ "dim_product.csv")
cust = pd.read_csv(D + "dim_customer.csv"); terr = pd.read_csv(D + "dim_territory.csv");
dates = pd.read_csv(D + "dim_date.csv", parse_dates=["date"])
for name, df in [("fact_sales", fact), ("dim_product", prod), ("dim_customer", cust),
("dim_territory", terr), ("dim_date", dates)]:
    print(f"{name:14s} {df.shape[0]:>6,} rows × {df.shape[1]} columns")
fact.head(4)
```

OUTPUT

fact_sales 9,000 rows × 9 columns
dim_product 45 rows × 6 columns
dim_customer 1,200 rows × 5 columns
dim_territory 6 rows × 3 columns
dim_date 1,096 rows × 6 columns

	order_number	order_date	product_key	customer_key	territory_key	quantity	ur
0	SO50387	2023-01-01	11	103	1	3	11
1	SO55797	2023-01-01	32	798	1	3	20
2	SO51236	2023-01-01	9	750	5	1	12
3	SO55082	2023-01-01	13	759	1	1	75

4 rows × 9 columns

Power Query: prepare each table

Open **Transform Data**. Typical steps for each table (all recorded in **Applied Steps**):

Table	Steps
fact_sales	Set types (order_date Date; quantity Whole number; amounts Fixed decimal number), remove blank rows, add Profit = [sales_amount] - [total_cost] as a custom column, keep only needed columns
dim_product	Trim/clean text, set list_price and standard_cost as decimals, remove duplicates on product_key
dim_customer	Replace M/F with Male/Female, remove duplicates on customer_key
dim_territory	Check there is one row per territory_key
dim_date	Set date to Date; sort month by month_no later in the model

Query settings (right pane) shows each step; click any step to see the data at that point, and delete a step to undo it. Rename queries (Sales, Product...) because those names become table names in the model.

Common transformations (from class and beyond)

Goal	Where	Notes
Promote headers, set types	Home	Always check types: wrong types break relationships
Remove duplicates / blank rows	Home → Remove rows	Needed for dimension keys
Split column by delimiter	Transform → Split Column	e.g. IN-2025-00042
Replace values, trim, clean, format	Transform	
Unpivot columns	Transform → Unpivot	Turn Jan, Feb, Mar columns into rows (Excel Lesson 12)
Merge queries	Home → Merge	A join; use to bring a category into the fact (but prefer relationships)
Append queries	Home → Append	Stack monthly files
Group by	Transform → Group By	Pre-aggregate large tables
Conditional / Custom column	Add Column	M formulas
Reference vs Duplicate	Right-click query	<i>Reference</i> creates a new query that starts from the output of another (changes flow through); <i>Duplicate</i> copies the steps
Parameters	Home → Manage Parameters	A variable such as a file path or a start date, used in queries (Lesson 4)
Disable load	Right-click query	Keep helper queries out of the model

What is "query folding"?

When the source is a database (SQL Server, Snowflake), Power Query **translates your steps into SQL** and lets the database do the work, instead of downloading everything first. A step that cannot be translated "breaks folding" and everything after it runs locally. Check by right-clicking a step → *View Native Query*. Keep foldable steps (filter, select columns, merge, group) **before** non-foldable ones.

Power Query M

```
let
    Source = Csv.Document(File.Contents("C:\Data\fact_sales.csv"), [Delimiter="," ,
Encoding=65001]),
    Promoted = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    Typed = Table.TransformColumnTypes(Promoted, {"order_date", type date},
{"quantity", Int64.Type},
{"sales_amount", Currency.Type}, {"total_cost", Currency.Type})),
    Profit = Table.AddColumn(Typed, "profit", each [sales_amount] - [total_cost],
Currency.Type)
in
    Profit
```

Relationships

A **relationship** links a column in one table to a column in another so that filters can flow between them. In **Model view**, drag dim_product[product_key] onto fact_sales[product_key]. Power BI usually detects relationships on same-named columns (Home → Manage relationships → Autodetect).

Cardinality

Type	Meaning	Use
One-to-many (1:*)	A key is unique in one table, repeated in the other	The standard: dimension → fact
One-to-one (1:1)	Unique on both sides	Splitting a wide table
Many-to-many (*:*)	Duplicates on both sides	Avoid; use a bridge table

Check uniqueness before creating a one-to-many: if the "one" side has duplicates Power BI will complain.

Python

```
checks = {"dim_product[product_key]": prod.product_key, "dim_customer[customer_key]":
cust.customer_key, "dim_territory[territory_key]": terr.territory_key, "dim_date[date]":
dates.date}
for name, col in checks.items(): print(f"{name:30s} unique: {col.is_unique}")
orphans = {"product_key": ~fact.product_key.isin(prod.product_key), "customer_key":
~fact.customer_key.isin(cust.customer_key), "territory_key":
~fact.territory_key.isin(terr.territory_key)}
for k, m in orphans.items(): print(f"fact rows with no matching {k}: {m.sum()}")
```

OUTPUT

```

dim_product[product_key]      unique: True
dim_customer[customer_key]    unique: True
dim_territory[territory_key]  unique: True
dim_date[date]                unique: True
fact rows with no matching product_key: 0
fact rows with no matching customer_key: 0
fact rows with no matching territory_key: 0

```

Unique keys on the "one" side and no orphans on the "many" side: the model is safe.

Filter direction



Filters flow along relationships from the one side to the many side (single direction). Selecting Bikes in dim_product filters fact_sales, which then filters the other dimensions only if bi-directional filtering is on.

Setting	Behaviour	Advice
Single (dimension → fact)	A selection in a dimension filters the fact	Default and best
Both (bi-directional)	Filters also flow back from fact to dimension	Use only when needed (many-to-many, "show only customers who bought"); can cause ambiguity and slowness
Active vs inactive relationship	Only one active path between two tables; an inactive one can be switched on in DAX with USERELATIONSHIP	e.g. Order Date and Ship Date both link to the date table

Seeing the model work

Total sales by product category and by country, using the relationships (a join in pandas):

Python

```
m = fact.merge(prod, on="product_key").merge(terr, on="territory_key")
m["profit"] = m.sales_amount - m.total_cost
out = m.pivot_table(index="category", columns="country", values="sales_amount",
aggfunc="sum", margins=True, margins_name="Total") / 1e6
out.round(1).rename_axis("Sales (₹ million)")
```

OUTPUT

country	India	Singapore	UAE	Total
Sales (₹ million)				
Accessories	6.7	1.7	1.3	9.8
Bikes	320.6	74.8	71.3	466.8
Clothing	6.0	1.3	1.1	8.5
Components	73.5	18.8	17.2	109.6
Total	406.9	96.7	91.0	594.6

5 rows × 4 columns

In Power BI you get this with a matrix: Rows category, Columns country, Values SUM(sales_amount). No join code needed: the relationships do it.

The date table

Time intelligence (year-to-date, same period last year, Lesson 4) requires a **proper date table**: one row per day with **no gaps**, covering all dates in the data, and **marked as a date table**.

DAX

```
Date =  
VAR MinDate = DATE ( YEAR ( MIN ( fact_sales[order_date] ) ), 1, 1 )  
VAR MaxDate = DATE ( YEAR ( MAX ( fact_sales[order_date] ) ), 12, 31 )  
RETURN  
ADDCOLUMNS (   
    CALENDAR ( MinDate, MaxDate ),  
    "Year",      YEAR ( [Date] ),  
    "Quarter",   "Q" & QUARTER ( [Date] ),  
    "Month No",  MONTH ( [Date] ),  
    "Month",     FORMAT ( [Date], "mmm" ),  
    "Year-Month", FORMAT ( [Date], "yyyy-mm" ),  
    "Weekday",   FORMAT ( [Date], "ddd" )  
)
```

STEP BY STEP Set it up

1. Model → **New table** → paste the DAX above (or load dim_date.csv).
2. Select the table → **Table tools** → **Mark as date table** → choose the Date column. Power BI validates that dates are unique and continuous.
3. Create the relationship Date[Date] → fact_sales[order_date] (one-to-many).
4. **Sort Month by Month No** (select Month → Column tools → Sort by column) so April does not come before January.
5. Hide the raw order_date in the fact table so report authors use the date table.

Python

```
print("Fact order dates :", fact.order_date.min().date(), "to",  
      fact.order_date.max().date())  
print("Date table      :", dates.date.min().date(), "to", dates.date.max().date(), "|  
rows:", len(dates))  
full = pd.date_range(dates.date.min(), dates.date.max())  
print("Continuous (no gaps):", len(full) == len(dates) and (dates.date.sort_values().values  
= full.values).all())  
dates.head(4)
```

OUTPUT

```
Fact order dates : 2023-01-01 to 2025-12-30  
Date table      : 2023-01-01 to 2025-12-31 | rows: 1096  
Continuous (no gaps): True
```

	date	year	quarter	month_no	month	weekday
0	2023-01-01	2023	Q1	1	Jan	Sun
1	2023-01-02	2023	Q1	1	Jan	Mon
2	2023-01-03	2023	Q1	1	Jan	Tue
3	2023-01-04	2023	Q1	1	Jan	Wed

4 rows × 6 columns

WATCH OUT Do not rely on auto date/time

Power BI can create hidden date tables automatically ("Auto date/time"). They bloat the model and cannot be customised or shared. Turn it off (File → Options → Data Load → Time intelligence) and always use your own marked date table.

Hierarchies

A **hierarchy** lets users drill down through levels with one click. In the Data pane, right-click category in `dim_product` → **Create hierarchy**, then drag subcategory and product onto it. Visuals with a hierarchy get drill-down buttons (arrows at the top-right of the visual): from category → sub-category → product. The date table's Year → Quarter → Month → Date is the classic one.

Common modelling problems

Symptom	Likely cause	Fix
Totals repeat the same number for every category	No relationship between the table you filter and the fact	Create the relationship
"A circular dependency was detected" / ambiguous paths	Two paths between two tables	Make one inactive, or remove the bi-directional filter
Many-to-many warning	Duplicate keys on both sides	Build a clean dimension with unique keys, or a bridge table
Blank row ("(Blank)") appears in visuals	Fact rows whose key is missing in the dimension (orphans)	Add the missing rows to the dimension, or fix keys
YTD / last-year measures give wrong or blank results	Date table not marked, has gaps, or not related	Fix the date table
Slow model, big file	Wide tables, high-cardinality text columns, auto date/time	Star schema, remove unused columns, reduce precision, disable auto date/time
Months sort alphabetically	Missing sort-by-column	Sort Month by Month No

TIP Best-practice checklist

1. Star schema with surrogate integer keys.
2. One-to-many, single-direction relationships.
3. One marked date table.
4. Hide keys and technical columns from report view.
5. Disable auto date/time.
6. Remove columns you do not use.
7. Name tables and columns for business users.
8. Put measures in a separate _Measures table (Lesson 3).

DO IT WITH AI Do it with AI

Ask Copilot or any assistant: *"Review this list of tables and columns and propose a star schema with a fact table and dimensions, the keys, and the relationships"* and paste your column lists. Compare its answer with the checklist above, and verify key uniqueness yourself in Power Query (Column profile → *Distinct equals Count*).

Summary

KEY TAKEAWAYS

- Model as a **star schema**: facts (events and numbers) surrounded by dimensions (descriptions), joined by **one-to-many, single-direction** relationships.
- Power Query prepares each table: types, cleaning, unpivot, merge/append, group, parameters; **query folding** pushes work to the source.
- Filters flow from the **one side** (dimension) to the **many side** (fact); use bi-directional filters sparingly; one active relationship per table pair, others inactive (USERRELATIONSHIP).
- A continuous, **marked date table** is required for time intelligence; sort month names by month number; turn off auto date/time.
- Hierarchies enable drill-down; most "wrong totals" come from missing relationships, duplicate keys or orphans.

Quick quiz

6 questions · tap an answer to check it

1. In a star schema, which table typically contains the numeric measures such as quantity and sales amount?

A Dimension table

B Fact table

C Date table

D Parameter table

Answer: B. Facts hold events and measures; dimensions hold descriptions.

2. A one-to-many relationship between dim_product and fact_sales means:

- A** A product appears once in sales
- B** Each product_key appears once in dim_product and many times in fact_sales
- C** Both tables have unique keys
- D** The relationship is bidirectional

Answer: B. The "one" side must have unique keys.

3. Why is a bi-directional relationship risky?

- A** It is not allowed
- B** It can create ambiguous filter paths and slow performance
- C** It deletes data
- D** It only works with dates

Answer: B. Use it only when you must (for example many-to-many scenarios).

4. What does "Mark as date table" require?

- A** A column of text
- B** A unique, continuous Date column with no gaps
- C** A relationship to every table
- D** A measure

Answer: B. Time-intelligence functions need a valid contiguous date table.

5. Months show as Apr, Aug, Dec, Feb... in alphabetical order. Fix?

- A** Rename the months
- B** Use a pie chart
- C** Sort the Month column by the Month No column
- D** Delete the date table

Answer: C. Sort by column controls the display order.

6. What does Power Query's "query folding" do?

- A** Compresses the PBIX file
- B** Translates steps into the source system's query (e.g. SQL) so the database does the work
- C** Hides the query
- D** Merges two queries

Answer: B. Folding improves refresh performance for database sources.

EXERCISE Model it yourself

Load the five Adventure Cycles CSV files, set types, create the four relationships, mark `dim_date` as the date table, hide key columns, create a Category → Subcategory → Product hierarchy, and build a matrix of sales by category (rows) and year (columns).

Show solution (worked answer)

In Power Query set `order_date` and `date` to Date. In Model view drag `product_key`, `customer_key`, `territory_key` from each dimension to the fact, and `date` to `order_date`. Table tools → Mark as date table → `date`. Right-click the key columns → Hide in report view. Create the hierarchy by right-clicking `category`. In Report view add a Matrix with Rows `dim_product[category]`, Columns `dim_date[year]`, Values Sum of `sales_amount`. Compare with the pandas pivot: Bikes should be about ₹46.7 crore of the total ₹59.5 crore.

INTERVIEW CORNER Questions you may be asked

Q: What is a star schema and why use it in Power BI? One fact table joined to several dimension tables with one-to-many relationships. It compresses well, filters predictably, and makes DAX simple and fast.

Q: How do filters propagate in a Power BI model? From the one side to the many side along active relationships, in the filter direction. Bi-directional relationships also filter from many back to one.

Q: Why do you need a date table? Built-in time-intelligence DAX (YTD, SAMEPERIODLASTYEAR) needs a continuous, marked date table; it also gives you year/quarter/month columns, fiscal calendars and holidays.

Q: How do you handle two dates in the fact table (order date and ship date)? Role-playing dimension: one date table with one active relationship (order date) and one inactive (ship date) activated with `USERELATIONSHIP` in specific measures, or two copies of the date table.

Practise more online

Free, trusted places to test yourself on this topic. Links open the original site in a new tab.