



Skill Navora

TRACK 4

Tableau

Tableau Public and Desktop: connecting data, dimensions and measures, every core chart, calculated fields, table calculations and LOD expressions, filters, parameters and sets, data modelling, dashboards, stories and publishing.

6 lessons

Version 2026.10

Updated October 2026

Created by **Varun Kumar**

www.skillnavora.in

Contents

Every entry is a clickable link. The PDF also has bookmarks for each lesson and section.

- [01 Getting Started with Tableau: Interface, Data Connections and Your First Views](#)
- [02 Charts in Tableau: From Bar and Line to Maps, Dual Axis, Histograms and Pareto](#)
- [03 Calculations in Tableau: Calculated Fields, Table Calculations and LOD Expressions](#)
- [04 Filters, Parameters, Sets, Groups and Hierarchies in Tableau](#)
- [05 Data Modelling in Tableau: Relationships, Joins, Unions, Blending, Extracts and Tableau Prep](#)
- [06 Dashboards, Actions, Stories and Publishing in Tableau \(Capstone\)](#)

Getting Started with Tableau: Interface, Data Connections and Your First Views

Understand the Tableau product family, install Tableau Public, connect to the Superstore India data, read the Data Source page, learn dimensions vs measures and discrete vs continuous fields, master the Marks card and shelves, build and sort your first views, and save them to Tableau Public.

★ Tableau Desktop / Tableau Public 2026

Beginner

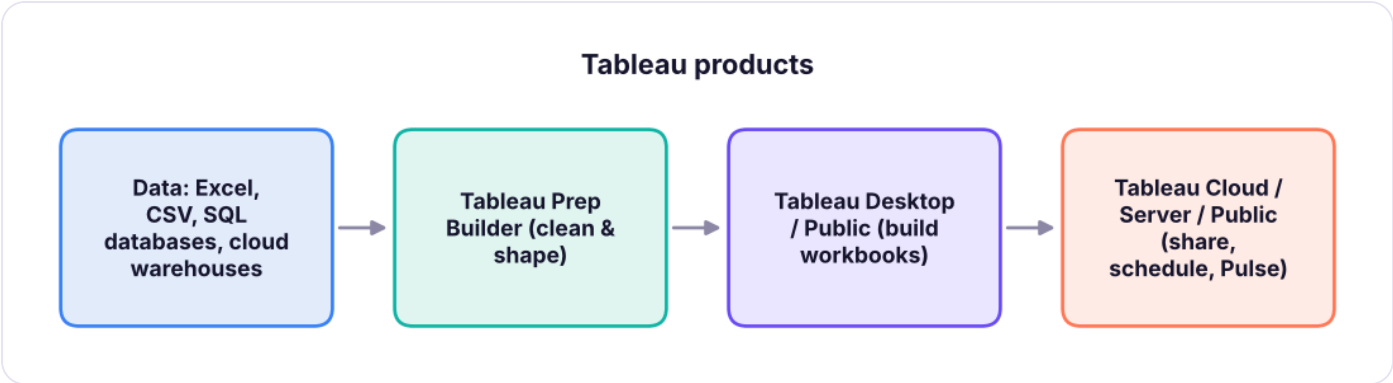
🕒 120 min

By the end of this lesson you can

- Describe Tableau Desktop, Public, Cloud, Server, Prep and Pulse, and when each is used
- Connect to a CSV, check data types and rename or hide fields on the Data Source page
- Explain dimensions vs measures and discrete (blue) vs continuous (green) fields
- Build bar, line and text-table views with the Columns, Rows, Marks, Filters and Pages shelves
- Sort, label, format and save a workbook to Tableau Public

What is Tableau?

Tableau is a visual analytics platform: you drag fields onto shelves and Tableau writes the query, aggregates the data and draws the chart. It is owned by Salesforce and, with Power BI, is one of the two BI tools most often named in Indian analyst job posts. Where Power BI leans on DAX formulas and the Microsoft stack, Tableau is known for fast visual exploration and highly customisable charts.



The Tableau product family. You build in Desktop (or Public), share on Cloud/Server/Public, and prepare data in Prep.

Product	What it is	Cost (2026)
Tableau Public (desktop app + website)	Free version of Desktop; workbooks are saved publicly on public.tableau.com	Free
Tableau Desktop	Full authoring tool; saves locally, connects to databases	Paid (Creator licence)
Tableau Cloud	Tableau's hosted service for sharing, refresh schedules and permissions	Paid, per user
Tableau Server	The same as Cloud, but installed on your company's own servers	Paid
Tableau Prep Builder	Visual tool to clean, join and reshape data before analysis	Included with Creator
Tableau Pulse / Tableau Agent	AI features: metric digests in plain language and an assistant that helps write calculations and build views	Cloud add-ons

TIP

Which one should you install?

Learners should install **Tableau Public** (free). It has almost every authoring feature of Desktop. The two limits: workbooks are saved to your public profile (so never use confidential data) and it connects mainly to files (Excel, CSV, text, JSON, PDF, spatial) and Google Sheets rather than company databases. Your public profile also works as a portfolio for interviews.



Install and connect

STEP BY STEP Install Tableau Public and connect to Superstore

1. Go to public.tableau.com → **Create** → download **Tableau Public** for Windows or Mac and install it; sign up for a free profile.
2. Open Tableau Public. On the **Connect** pane choose **Text file** and pick `superstore_sales.csv` (for an Excel file choose **Microsoft Excel**).
3. The **Data Source page** opens: the table appears in the canvas and a preview grid below it.
4. Check each column's **data type icon** (Abc = string, # = number, calendar = date, globe = geographic). Click an icon to change it.
5. Click **Sheet 1** at the bottom to start building.

The data you connect to is the same Superstore India file used in the Excel and Power BI lessons. Check its size before you build anything:

Python

```
import pandas as pd
df = pd.read_csv("superstore_sales.csv", parse_dates=["order_date", "ship_date"])
print(df.shape)
print(df.dtypes.astype(str).value_counts().to_dict())
print("Orders from", df.order_date.min().date(), "to", df.order_date.max().date())
df.head(3)
```

OUTPUT

```
(6323, 19)
{'object': 12, 'float64': 3, 'int64': 2, 'datetime64[ns]': 2}
Orders from 2023-01-01 to 2025-12-31
```

	row_id	order_id	order_date	ship_date	ship_mode	customer_id	customer_name
0	1	IN-2023-00001	2023-01-01	2023-01-05	Second Class	C0765	Ishaan Gupta
1	2	IN-2023-00001	2023-01-01	2023-01-05	Second Class	C0765	Ishaan Gupta
2	3	IN-2023-00001	2023-01-01	2023-01-05	Second Class	C0765	Ishaan Gupta

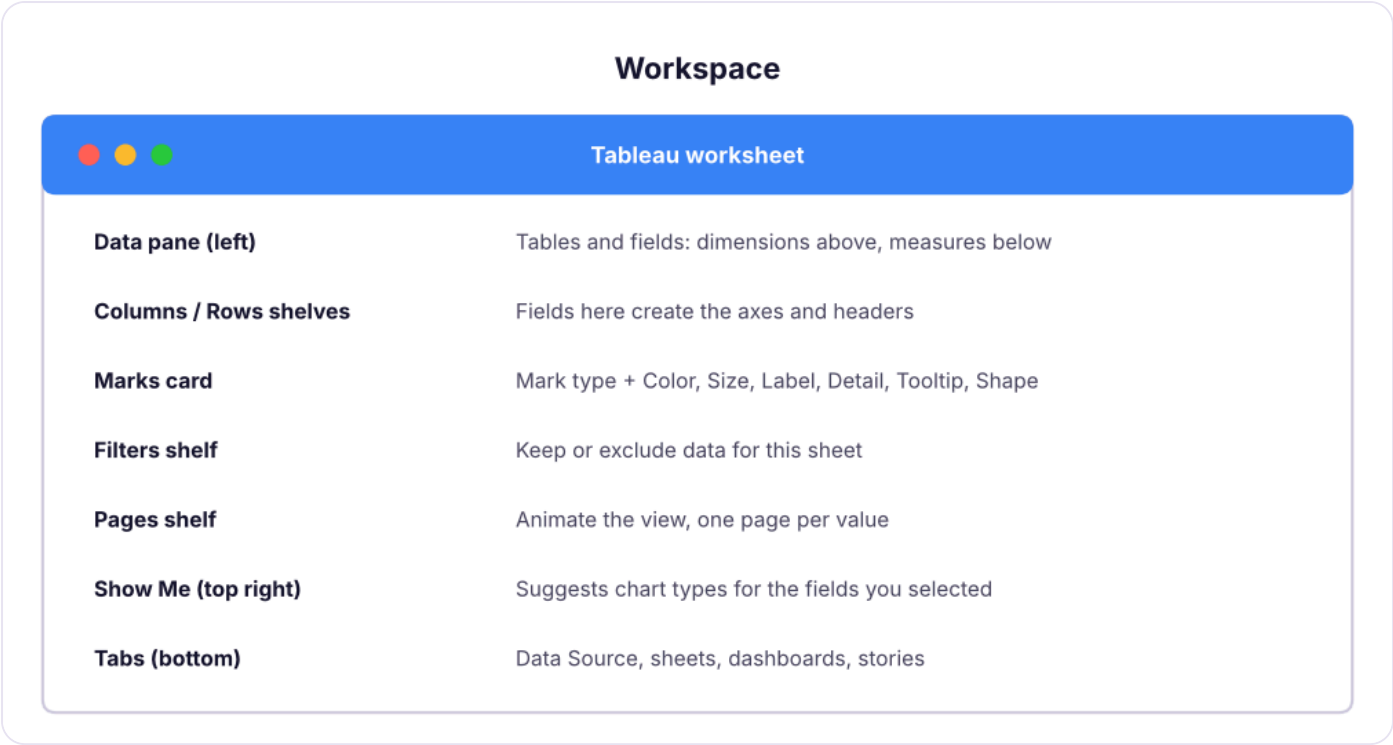
3 rows × 19 columns

6,323 rows (order lines) and 19 columns: Tableau will show these as **fields** in the Data pane.

The Data Source page

Feature	Where	Use
Rename a field	Click the column header → Rename	Friendly names such as "Sub-Category"
Hide a field	Column menu → Hide	Remove IDs you will never use
Change type / geographic role	Click the type icon	Make state a State/Province so it maps
Split	Column menu → Split	customer_name → first and last name
Pivot	Select columns → Pivot	Turn month columns into rows
Data Interpreter	Left pane checkbox	Clean messy Excel sheets with titles and merged cells
Live vs Extract	Top right	Extract (.hyper) is a fast snapshot; Live queries the source every time
Filters (data source)	Top right → Add	Keep only the rows the workbook needs

The workspace



The Tableau worksheet layout. Drag fields from the Data pane onto shelves and the Marks card.

Dimensions vs measures, discrete vs continuous

Tableau sorts every field in two ways, and the colour of a pill tells you which:

	Meaning	Example	What it creates
Dimension	Categories you slice by	Region, Category, Customer Name	Groups (one mark per value)
Measure	Numbers you aggregate	Sales, Profit, Quantity	SUM, AVG, COUNT...
Discrete (blue pill)	Separate values	Region, YEAR(Order Date) as "2023, 2024"	Headers
Continuous (green pill)	An unbroken range	SUM(Sales), exact date	Axes

Pill colours

Discrete (blue)

- Draws headers / labels
- Region, Segment, Ship Mode
- YEAR(Order Date) as separate years
- Colour legend: distinct colours

Continuous (green)

- Draws an axis
- SUM(Sales), AVG(Discount)
- MONTH(Order Date) as a timeline
- Colour legend: a gradient

Blue (discrete) pills create headers; green (continuous) pills create axes.

A measure dropped into the view is **aggregated** (default SUM). Right-click the pill → **Measure** to switch to Average, Median, Count, Count (Distinct), Minimum, Maximum, Standard deviation and so on.

Your first views

Sales by region (bar chart)

Drag **Region** to Rows and **Sales** to Columns. Tableau writes the equivalent of `SELECT region, SUM(sales) FROM superstore GROUP BY region` and draws one bar per region:

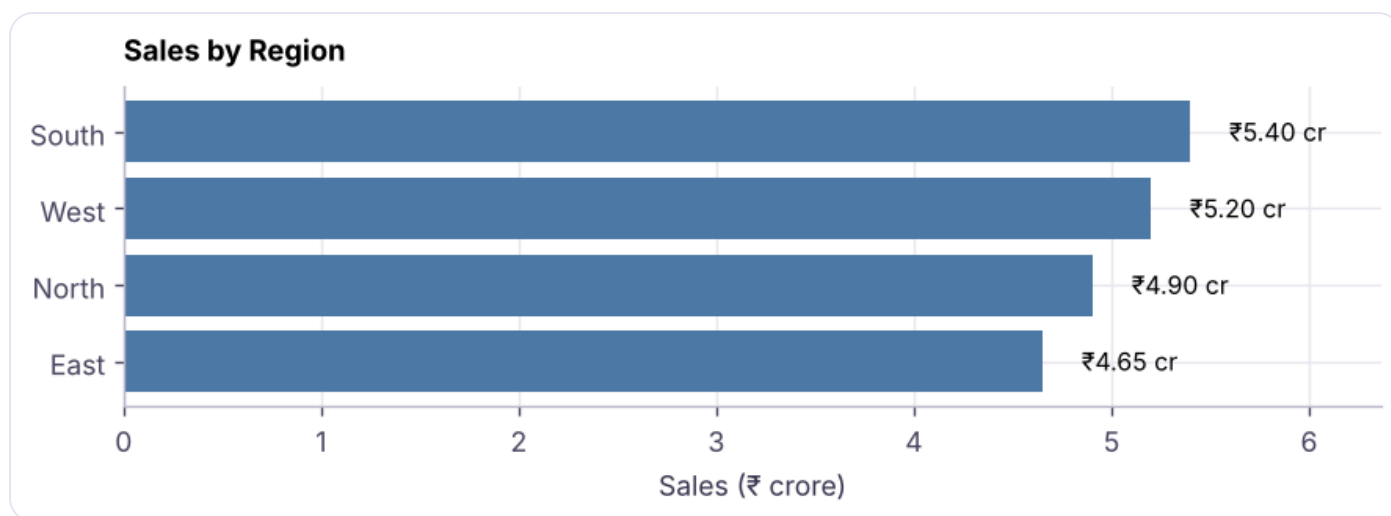
Python

```
by_region = (df.groupby("region")[["sales", "profit"]].sum() /
1e7).round(2).sort_values("sales", ascending=False)
by_region.columns = ["Sales (₹ crore)", "Profit (₹ crore)"]
by_region
```

OUTPUT

	Sales (₹ crore)	Profit (₹ crore)
region		
South	5.40	0.47
West	5.20	0.47
North	4.90	0.46
East	4.65	0.44

4 rows × 2 columns



The view you get: SUM(Sales) by Region, sorted descending, with labels.

The default Tableau colours (blue #4E79A7, orange #F28E2B, red #E15759...) are the **Tableau 10** palette used in the figures in this track.

STEP BY STEP Sort, label and colour it

1. Hover the Sales axis and click the **sort** icon, or click the toolbar sort descending button.
2. Click **Label** on the Marks card → tick *Show mark labels*.
3. Drag **Profit** onto **Color**: bars are shaded by profit, so sales and margin show together.
4. Right-click the Sales axis → **Format** → Numbers → Currency (Custom) → Display units: *Crores* is not built in, so use a calculated field `[Sales]/1e7` named "Sales (₹ cr)" (Lesson 3), or show *Millions*.
5. Double-click the sheet tab and rename it "Sales by Region".

Sales over time (line chart)

Drag **Order Date** to Columns: Tableau shows **YEAR(Order Date)** as a discrete header. Click the **+** on the pill to drill down to Quarter and Month. Right-click the pill and choose the **green** "Month" (continuous) to draw a single timeline, or the **blue** "Month" (discrete) to compare the same month across years.

Python

```
m = df.set_index("order_date").sales.resample("ME").sum() / 1e5
print("Months:", len(m), "| best month:", m.idxmax().strftime("%b %Y"), f"₹{m.max():.1f} lakh", "| weakest:", m.idxmin().strftime("%b %Y"), f"₹{m.min():.1f} lakh")
yearly = (df.groupby(df.order_date.dt.year).sales.sum() / 1e7).round(2); yearly
```

OUTPUT

Months: 36 | best month: Nov 2025 ₹102.8 lakh | weakest: Jan 2023 ₹24.5 lakh

sales	
order_date	
2023	5.76
2024	6.54
2025	7.84

3 rows × 1 columns

Text table (crosstab)

Put **Category** on Rows, **YEAR(Order Date)** on Columns and **Sales** on **Text**. Add **Analysis → Totals → Show Row Grand Totals**. This is Tableau's pivot table:

Python

```
ct = pd.crosstab(df.category, df.order_date.dt.year, values=df.sales, aggfunc="sum",
margins=True, margins_name="Grand Total") / 1e5
ct.round(1)
```

OUTPUT

order_date	2023	2024	2025	Grand Total
category				
Furniture	99.5	121.3	127.6	348.4
Office Supplies	26.1	35.5	39.7	101.3
Technology	450.8	497.2	616.5	1,564.4
Grand Total	576.4	654.0	783.8	2,014.1

4 rows × 4 columns

Show Me

Show Me (top right) highlights the chart types that fit the fields you have selected and states what it needs, e.g. *"1 date, 1 or more dimensions, 1 or more measures"* for a line chart. It is a fast start, but learn to build each chart by hand: interviewers often ask *how* you would build a view.

Saving and sharing

Format	What it contains
.twb workbook	Sheets, dashboards and the connection, no data
.twbx packaged workbook	Workbook plus a copy of the data/extract: share this one
.hyper extract	Tableau's compressed, fast data snapshot
Tableau Public profile	File → Save to Tableau Public ; anyone can view and download it

COMMON MISTAKES Beginner mistakes

- Saving confidential company data to Tableau Public: everything there is public.
- Sending a .twb file: the receiver cannot open it without your data. Send .twbx.
- Leaving a date as a string ("Abc"): you lose drill-down and date functions. Change the type on the Data Source page.
- Using a measure that should be a dimension (e.g. Postal Code as SUM): right-click → Convert to Dimension.

- **Tableau Agent** (formerly Einstein Copilot for Tableau) suggests calculations and builds views from a typed question in Tableau Cloud.
- **Tableau Pulse** sends plain-language metric summaries ("Sales are up 8% this week, driven by the West region") to email and Slack.
- **Tableau Next** is Salesforce's agent-based analytics layer on top of Data Cloud. Desktop and Public remain the tools you learn first.

Summary

KEY TAKEAWAYS

- Tableau turns drag-and-drop into queries: shelves define the structure, the Marks card defines the look.
- Dimensions slice, measures aggregate; blue pills are discrete (headers), green pills are continuous (axes).
- Start with Tableau Public, connect to a file, fix data types on the Data Source page, then build.
- Share packaged workbooks (.twbx) or publish to Tableau Public/Cloud.

Quick quiz

6 questions · tap an answer to check it

1. What does a green pill on Columns create?

A A continuous axis

B A set of headers

C A filter

D A calculated field

Answer: A. Green = continuous = axis; blue = discrete = headers.

2. You drag Sales into the view. What does Tableau show by default?

A SUM(Sales) for each combination of the dimensions in the view

B Every row's sales value

C The average sales

D The count of rows

Answer: A. Measures are aggregated, and SUM is the default.

3. Which file should you email a colleague so they can open your work with the data?

A A packaged workbook (.twbx)

B A workbook (.twb)

C A .hyper file only

D A screenshot

Answer: A. .twbx bundles the workbook with its data.

4. What is the main limitation of Tableau Public?

A Workbooks are saved publicly and it connects mainly to files

B It cannot draw maps

C It has no calculated fields

D It only works on Mac

Answer: A. Never put confidential data on Tableau Public.

5. State is shown as "Abc" and will not draw a map. What do you do?

A Assign the geographic role State/Province

B Convert it to a measure

C Delete the field

D Use a dual axis

Answer: A. Geographic roles give a field latitude and longitude.

6. Which feature suggests chart types for the selected fields?

A Show Me

B Pages shelf

C Data Interpreter

D Extract

Answer: A. Show Me lists the chart types and what each needs.

EXERCISE Practice

Connect to `superstore_sales.csv` in Tableau Public and build three sheets: (1) Sales by Category sorted descending with labels; (2) a continuous monthly Sales line with Profit on Color; (3) a text table of Segment × Year with grand totals. Save as a packaged workbook and publish to your Tableau Public profile.

Show solution (worked answer)

Sheet 1: Category on Rows, SUM(Sales) on Columns, sort descending, Label on. Sheet 2: MONTH(Order Date) continuous (green) on Columns, SUM(Sales) on Rows, SUM(Profit) on Color. Sheet 3: Segment on Rows, YEAR(Order Date) on Columns, SUM(Sales) on Text, Analysis → Totals → Show Row and Column Grand Totals. File → Save to Tableau Public.

INTERVIEW CORNER**Questions you may be asked**

Q: What is the difference between a dimension and a measure? A dimension is a categorical field used to group the data (Region, Category); a measure is a numeric field that Tableau aggregates (SUM of Sales). Each mark represents one combination of the dimensions in the view.

Q: Discrete vs continuous? Discrete fields (blue) create headers and distinct colours; continuous fields (green) create axes and colour gradients. Dates can be either: discrete YEAR gives "2023, 2024" headers, continuous MONTH gives a timeline.

Q: .twb vs .twbx vs .hyper? .twb is the workbook definition without data; .twbx packages the workbook with its data or extract; .hyper is the extract file itself.

Q: Live connection or extract? Live always shows current data but depends on the source's speed; an extract is a fast, compressed snapshot that you refresh on a schedule. Use extracts for speed and offline work, live for real-time needs or very large governed warehouses.

Charts in Tableau: From Bar and Line to Maps, Dual Axis, Histograms and Pareto

Build every core chart in Tableau and know when to use each: bar, stacked and side-by-side bars, line and area, pie and donut, treemap, highlight table, scatter plot, histogram with bins, box plot, filled and symbol maps, dual-axis combo charts, bullet charts and a Pareto chart, all on the Superstore India data.

✦ Tableau Desktop / Tableau Public 2026

Beginner

🕒 150 min

By the end of this lesson you can

- Choose the right chart for comparison, trend, part-to-whole, distribution, relationship and geography questions
- Build bar, line, area, pie, donut, treemap and highlight-table views
- Create bins and histograms, box plots and scatter plots with trend lines
- Map states with geographic roles, and combine two measures with dual and blended axes
- Build a bullet chart and a Pareto (80/20) chart

Choose the chart from the question

Question → chart

Compare

Bar, side-by-side bar, bullet. 'Which region sells most?'

Trend

Line, area, dual-axis. 'How did sales move by month?'

Part-to-whole

Stacked bar, treemap, pie/donut (few slices). 'What share is Technology?'

Distribution

Histogram, box plot. 'How big is a typical order?'

Relationship

Scatter with trend line. 'Do discounts hurt profit?'

Geography

Filled or symbol map. 'Which states lead?'

Match the business question to the chart family.

All examples below use the Superstore India data. The tables show the numbers your Tableau view should display, computed here with pandas so you can check your work.

Python

```
import pandas as pd, numpy as np
df = pd.read_csv("superstore_sales.csv", parse_dates=["order_date"])
df["year"] = df.order_date.dt.year
print(len(df), "order lines,", df.order_id.nunique(), "orders,", df.customer_id.nunique(), "customers")
```

OUTPUT

6323 order lines, 3200 orders, 785 customers



Bar charts

Variant	Build	Use
Bar	Dimension on Rows, measure on Columns	Ranking categories
Stacked bar	Add a second dimension to Color	Total and composition
Side-by-side bar	Second dimension on Columns <i>before</i> the measure (or Show Me → side-by-side)	Comparing subgroups directly
100% stacked	Stacked bar + Quick Table Calculation → Percent of Total (compute using the colour dimension)	Mix comparisons

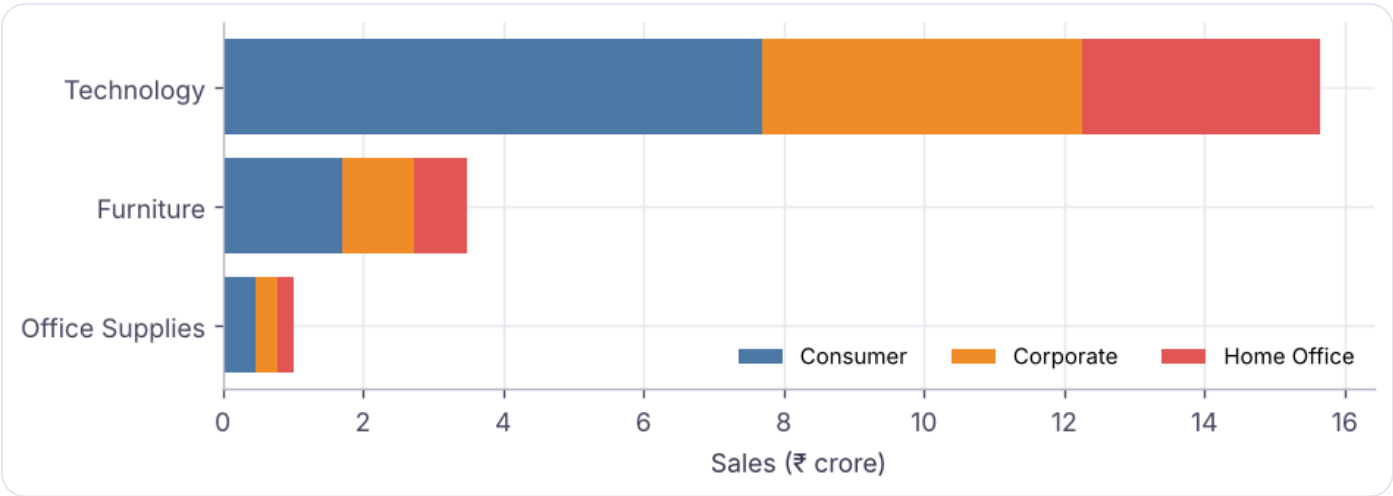
Python

```
seg = pd.crosstab(df.category, df.segment, values=df.sales, aggfunc="sum") / 1e7
seg.round(2)
```

OUTPUT

segment	Consumer	Corporate	Home Office
category			
Furniture	1.70	1.03	0.75
Office Supplies	0.47	0.32	0.23
Technology	7.69	4.56	3.39

3 rows × 3 columns



Stacked bar: Sales by Category, coloured by Segment (₹ crore).

Line and area charts

Put a **continuous** date (green MONTH(Order Date)) on Columns and Sales on Rows for one timeline.
Put **YEAR** on Color and a **discrete** MONTH on Columns to compare the same months across years (seasonality).

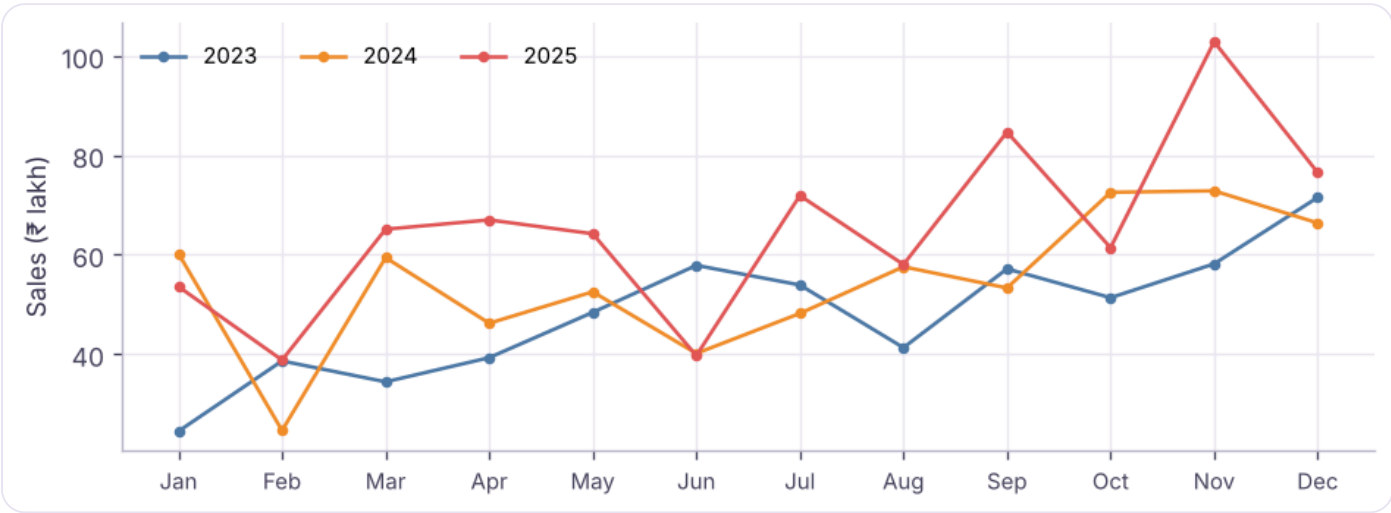
Python

```
pv = df.pivot_table(index=df.order_date.dt.month, columns="year", values="sales",  
aggfunc="sum") / 1e5  
pv.index.name = "month"  
pv.round(1).head(6)
```

OUTPUT

year	2023	2024	2025
month			
1	24.5	60.2	53.6
2	38.6	24.6	38.8
3	34.4	59.4	65.1
4	39.3	46.2	67.0
5	48.4	52.5	64.2
6	57.9	40.1	39.7

6 rows × 3 columns



Discrete month on Columns, Year on Color: each year is a line, so seasonality is easy to compare.

An **area chart** is the same view with Mark type = Area; stack a dimension on Color to show how the total is built up.

Part-to-whole: pie, donut, treemap

- **Pie**: Mark type Pie, dimension on Color, measure on **Angle**. Use for 2–5 slices only.
- **Donut**: two pies on a dual axis. Put `AVG(0)` (type `avg(0)` on Rows) twice, make the second smaller and white, turn on dual axis, and put the total on the inner label.
- **Treemap**: dimension on **Detail/Label**, measure on **Size** and **Color**. Good for many categories (e.g. the 13 sub-categories).

Python

```
sub = (df.groupby("sub_category").agg(sales=("sales", "sum"), profit=("profit", "sum")) /
1e5).sort_values("sales", ascending=False)
sub["margin %"] = (sub.profit / sub.sales * 100).round(1)
sub.round(1).head(8)
```

OUTPUT

	sales	profit	margin %
sub_category			
Copiers	787.5	84.4	10.7
Machines	425.9	51.9	12.2
Phones	319.2	36.6	11.5
Tables	140.5	-8.2	-5.8
Bookcases	112.7	-4.2	-3.7
Chairs	80.9	3.4	4.2
Appliances	63.3	9.4	14.9
Accessories	31.8	3.7	11.6

8 rows × 3 columns

Highlight table (heat map)

Category on Rows, YEAR on Columns, Sales on **Text** and **Color**, Mark type **Square**. Tableau shades each cell, so high and low cells stand out without reading every number.

Distribution: histogram and box plot

STEP BY STEP Histogram of order value

1. Right-click **Sales** → Create → **Bins**; set the bin size (e.g. ₹10,000). A new dimension "Sales (bin)" appears.
2. Drag **Sales (bin)** to Columns and **CNT(Sales)** (or Number of Records) to Rows.
3. Or select Sales and pick the histogram in Show Me: it creates the bins for you.

Python

```
bins = pd.cut(df.sales, bins=[0, 5000, 10000, 20000, 50000, 100000, df.sales.max()],
right=False)
print(bins.value_counts().sort_index())
print("median line value: ₹", round(df.sales.median()), "| mean: ₹",
round(df.sales.mean()))
```

OUTPUT

```
sales
[0.0, 5000.0)          2530
[5000.0, 10000.0)      829
[10000.0, 20000.0)     896
[20000.0, 50000.0)     942
[50000.0, 100000.0)    546
[100000.0, 619290.0)   579
Name: count, dtype: int64
median line value: ₹ 8430 | mean: ₹ 31854
```

The mean is well above the median: a few large orders pull it up (right skew). A **box plot** (Show Me → box-and-whisker) shows the median, quartiles and outliers per category at a glance.

Relationship: scatter plot

Put SUM(Sales) on Columns, SUM(Profit) on Rows and **Sub-Category** on Detail: one dot per sub-category. Add **Analytics pane** → **Trend Line** (linear) and hover it to read R^2 and the p-value.

Python

```
s = df.groupby("sub_category")[["sales", "profit", "discount"]].agg({"sales": "sum",
"profit": "sum", "discount": "mean"})
print("correlation sales vs profit across sub-categories:", round(s.sales.corr(s.profit),
2))
print("correlation avg discount vs profit margin:", round(s.discount.corr(s.profit /
s.sales), 2))
```

OUTPUT

correlation sales vs profit across sub-categories: 0.95
correlation avg discount vs profit margin: -0.88

Maps

1. Make sure **State** has the geographic role *State/Province* (and the country is India: Map → Edit Locations).
2. Double-click State: Tableau adds Latitude/Longitude (generated) and draws a **symbol map**. Change Mark type to **Map** for a **filled (choropleth) map**.
3. Put Sales on Color for a filled map, or on Size for a symbol map. Use Profit on Color with a diverging palette (orange-blue) to show loss-making states.

Python

```
st = (df.groupby("state")[["sales", "profit"]].sum() / 1e5).round(1).sort_values("sales",
ascending=False)
st.head(6)
```

OUTPUT

	sales	profit
state		
Maharashtra	256.7	24.4
Delhi	156.6	17.3
Kerala	147.1	12.6
Madhya Pradesh	144.1	12.7
West Bengal	138.9	14.6
Karnataka	136.3	11.8

TIP Indian maps

Tableau's built-in geocoding knows Indian states and many districts and cities. If a name does not match (old spellings, "Orissa" vs "Odisha"), use **Map → Edit Locations** to match it, or join a custom latitude/longitude file.

Dual-axis and combo charts

Show two measures with different scales, e.g. Sales as bars and Profit margin as a line:

1. Continuous MONTH(Order Date) on Columns; SUM(Sales) and the margin calculation on Rows.
2. Right-click the second pill → **Dual Axis**. Set the Marks card for Sales to **Bar** and for margin to **Line**.
3. Right-click the right axis → **Synchronize Axis** only when both measures share units (Sales and Profit); never synchronise ₹ with %.

A **blended axis** (drag a second measure onto the *existing* axis until the double-ruler icon shows) puts two measures with the same unit on one axis, using Measure Names on Color.

Bullet chart (actual vs target)

Select a measure and a target (e.g. Sales and a target field) → Show Me → **Bullet graph**. The bar is the actual, the reference line is the target, and the shaded bands are % of target (60%, 80%). If the bars and lines come out the wrong way round, right-click the axis → **Swap Reference Line Fields**.

Pareto chart (80/20)

STEP BY STEP Build a Pareto of products

1. Product Name on Columns sorted by SUM(Sales) descending; SUM(Sales) on Rows as bars.
2. Drag SUM(Sales) to Rows again → Quick Table Calculation → **Running Total**, then Edit Table Calculation → *Add secondary calculation* → **Percent of Total**.
3. Make it dual axis; set the second mark to Line; add a reference line at 80%.
4. Optional: replace Product Name on Columns with a table calc INDEX()/SIZE() so the x-axis reads "% of products".

Python

```
p = df.groupby("product_name").sales.sum().sort_values(ascending=False)
cum = p.cumsum() / p.sum()
n80 = int((cum < .8).sum()) + 1
print(f"{n80} of {len(p)} products ({n80 / len(p):.0%}) make 80% of sales")
```

OUTPUT

21 of 78 products (27%) make 80% of sales

Sales are concentrated: about a quarter of the products bring in 80% of sales, close to the classic 80/20 rule. Those top products deserve the best stock availability and marketing. A Pareto chart shows how concentrated a business really is, so you can judge whether a "focus on the top products" strategy makes sense.

Summary

KEY TAKEAWAYS

- Pick the chart from the question: compare, trend, part-to-whole, distribution, relationship, geography.
- Continuous dates draw timelines; discrete dates compare periods.
- Bins create histograms; the Analytics pane adds trend lines, reference lines and box plots.
- Assign geographic roles before mapping; use filled maps for rates and symbol maps for totals.
- Dual axis combines two marks (bar + line); synchronise only when units match.

Quick quiz

6 questions · tap an answer to check it

1. Which chart best shows how a single order-value variable is distributed?

A Histogram

B Pie chart

C Treemap

D Line chart

Answer: A. Histograms show distributions; create bins first.

2. How do you get Profit as a line on top of Sales bars with its own scale?

A Put both on Rows, use Dual Axis, set one mark to Bar and the other to Line

B Put Profit on Color

C Use a blended axis

D Synchronise the axes

Answer: A. Dual axis gives two independent axes and separate mark types.

3. Which is the right way to show 13 sub-categories as a part-to-whole?

A Treemap or sorted bar chart

B Pie chart with 13 slices

C Scatter plot

D Line chart

Answer: A. Pies stop being readable beyond about five slices.

4. You want each year as its own line to compare seasonality. What goes on Columns?

- A** Discrete MONTH(Order Date)
- B** Continuous MONTH(Order Date)
- C** YEAR(Order Date) only
- D** Sales

Answer: A. Discrete month gives Jan-Dec headers; Year goes on Color.

5. What does the Pareto line show?

- A** The running total of sales as a percent of the grand total
- B** Profit margin
- C** Average sales
- D** The number of orders

Answer: A. Running total + percent of total secondary calculation.

6. When should you synchronise a dual axis?

- A** When both measures share the same unit and scale makes sense
- B** Always
- C** When one is in ₹ and the other in %
- D** Never

Answer: A. Synchronising ₹ with % distorts the chart.

EXERCISE Practice

Build: (1) a side-by-side bar of Sales by Region and Segment; (2) a filled map of Profit by State with a diverging palette; (3) a histogram of Sales with ₹10,000 bins; (4) a dual-axis chart of monthly Sales (bars) and Profit (line); (5) a Pareto chart of sub-categories with an 80% reference line.

Show solution (worked answer)

Use the steps in each section. For the map, set Profit on Color → Edit Colors → Orange-Blue Diverging, centre 0. For the Pareto use Running Total + Percent of Total, compute using Sub-Category, then Analytics → Reference Line → Constant 0.8 on the line axis.

INTERVIEW CORNER Questions you may be asked

Q: Dual axis vs blended axis? A dual axis gives two independent axes (left and right) and allows different mark types; a blended axis puts several measures on one shared axis (same unit) using Measure Names/Values.

Q: How do you build a Pareto chart in Tableau? Sort the dimension by the measure, show bars, add the measure again with a Running Total table calculation and a secondary Percent of Total, make it dual axis as a line, and add an 80% reference line.

Q: Filled map or symbol map? Filled maps suit rates and ratios (profit margin by state) because area does not scale with value; symbol maps with size suit totals (sales by city).

Calculations in Tableau: Calculated Fields, Table Calculations and LOD Expressions

Write calculated fields (row-level and aggregate), use string, date, number and logical functions, handle nulls, then master table calculations (running total, percent of total, difference and growth, rank, moving average, and Compute Using) and Level of Detail expressions (FIXED, INCLUDE, EXCLUDE), with every result checked against pandas.

✦ Tableau Desktop / Tableau Public 2026

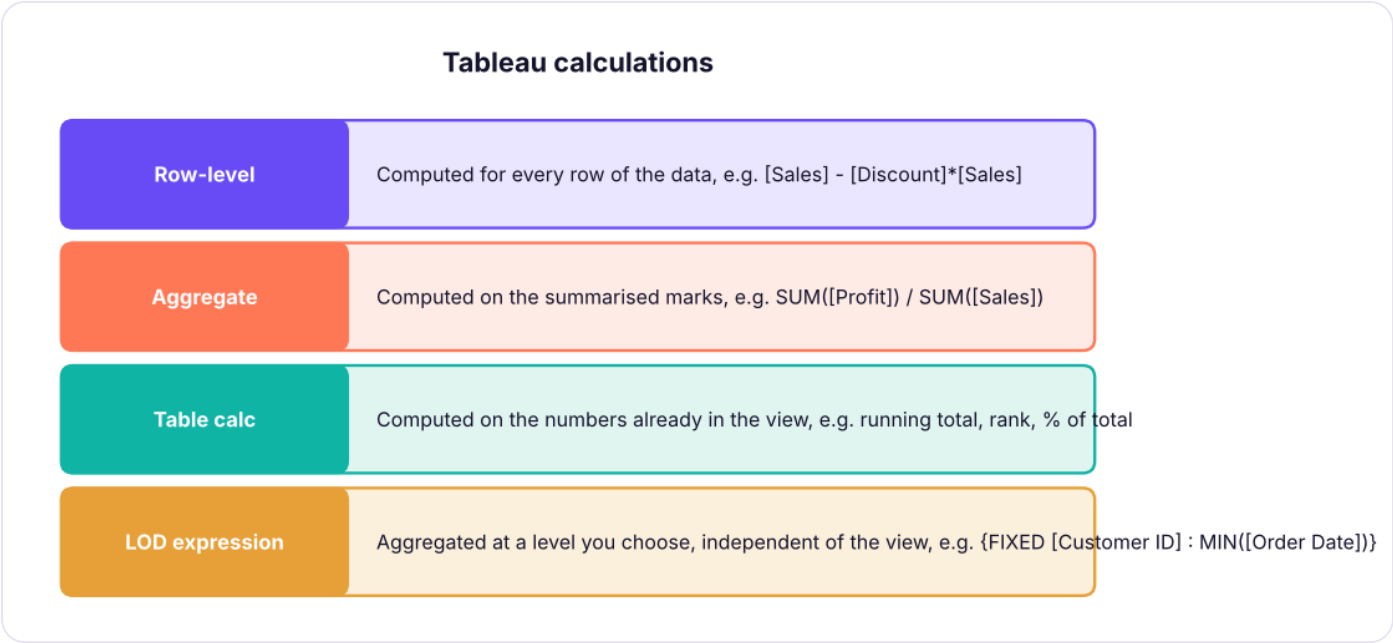
Intermediate

🕒 150 min

By the end of this lesson you can

- Write row-level and aggregate calculated fields and know which one a question needs
- Use IF/ELSEIF, CASE, IIF, ZN, IFNULL, DATEDIFF, DATETRUNC and string functions
- Apply quick table calculations and control them with Compute Using
- Write FIXED, INCLUDE and EXCLUDE LOD expressions and explain where they sit in the order of operations
- Solve classic interview problems: profit margin, customer first-purchase cohorts and percent of total

Three kinds of calculation



Where each kind of calculation is computed.

Create one with **Analysis → Create Calculated Field** (or right-click a field → Create → Calculated Field). Field names go in square brackets; comments start with `//`.

Python

```
import pandas as pd, numpy as np
df = pd.read_csv("superstore_sales.csv", parse_dates=["order_date", "ship_date"])
print(df[["sales", "profit", "discount"]].describe().round(1).loc[["mean", "min", "max"]])
```

OUTPUT

	sales	profit	discount
mean	31854.1	2902.4	0.1
min	42.0	-75873.8	0.0
max	619290.0	149424.9	0.4

Row-level vs aggregate: the profit-margin trap

The most common beginner bug in Tableau (and Power BI) is averaging ratios.

Calculated field	Formula	What it means
Margin (row)	<code>[Profit] / [Sales]</code>	Margin of each order line; in the view it becomes SUM or AVG of those ratios: wrong
Margin (aggregate)	<code>SUM([Profit]) / SUM([Sales])</code>	Total profit ÷ total sales for whatever the mark represents: right

Python

```
by_cat = df.groupby("category").apply(lambda g: pd.Series({
    "aggregate margin %": g.profit.sum() / g.sales.sum() * 100,
    "average of row margins %": (g.profit / g.sales).mean() * 100}), include_groups=False)
by_cat.round(2)
```

OUTPUT

	aggregate margin %	average of row margins %
category		
Furniture	-2.35	-2.38
Office Supplies	14.95	13.76
Technology	11.29	9.95

3 rows × 2 columns

The two columns differ because the average of row ratios gives a ₹500 order the same weight as a ₹5 lakh order. Format the aggregate field as a percentage (right-click → Default Properties → Number Format → Percentage).

Useful functions

Purpose	Tableau	Example
Logic	IF ... THEN ... ELSEIF ... ELSE ... END	IF SUM([Profit]) < 0 THEN "Loss" ELSE "Profit" END
Logic	CASE [Region] WHEN "South" THEN ... END, IIF(test, then, else)	Short mappings
Nulls	ZN([Profit]), IFNULL([x], 0), ISNULL([x])	Replace nulls with 0
Dates	DATEDIFF('day', [Order Date], [Ship Date]), DATETRUNC('month', [Order Date]), DATEPART('weekday', ...), DATEADD, TODAY()	Shipping days, month start
Strings	LEFT, RIGHT, MID, SPLIT([Customer Name], " ", 1), CONTAINS, UPPER, TRIM, REPLACE	First name
Numbers	ROUND, ABS, CEILING, FLOOR, INT	Rounding
Type	STR(), INT(), FLOAT(), DATE()	Conversions
Aggregates	SUM, AVG, COUNT, COUNTD, MIN, MAX, MEDIAN, ATTR	COUNTD([Order ID]) = orders

Python

```
df["ship_days"] = (df.ship_date - df.order_date).dt.days # DATEDIFF('day', [Order
Date], [Ship Date])
df["first_name"] = df.customer_name.str.split(" ").str[0] # SPLIT([Customer Name],
" ", 1)
df["order_size"] = np.select([df.sales ≥ 50000, df.sales ≥ 10000], ["Large", "Medium",
"Small"]) # IF/ELSEIF
print(df.groupby("ship_mode").ship_days.mean().round(1).sort_values())
print(df.order_size.value_counts().to_dict())
```

OUTPUT

```
ship_mode
Same Day      0.0
First Class   1.5
Second Class  3.0
Standard Class 5.6
Name: ship_days, dtype: float64
{'Small': 3359, 'Medium': 1838, 'Large': 1126}
```

Table calculations

A table calculation works on the **numbers already in the view**, so its result depends on the layout. Right-click a measure → **Quick Table Calculation**:

Quick table calc	Use
Running Total	Cumulative sales through the year
Difference / Percent Difference	Month-on-month or year-on-year change
Percent of Total	Share of each category
Rank	Top products
Percentile	Position in a distribution
Moving Average	Smooth noisy monthly data (Edit → previous 2 values + current)
YTD Total / YTD Growth / YoY Growth	Financial reporting

Compute Using (addressing and partitioning)

Compute Using tells Tableau the direction of the calculation. Fields it computes *along* are **addressing** fields; the remaining fields **partition** the view into separate calculations. Example: Category on Rows, MONTH on Columns, Running Total of Sales:

- Compute Using **Table (across)** → running total along months, restarting for each category (partitioned by Category).
- Compute Using **Table (down)** → running total down the categories within each month.

Python

```
m = df[df.order_date.dt.year == 2025].assign(month=lambda x:
x.order_date.dt.to_period("M"))
t = m.pivot_table(index="category", columns="month", values="sales", aggfunc="sum") / 1e5
running = t.cumsum(axis=1) # Table (across): restarts per category
print(running.iloc[:, [0, 1, 2, -1]].round(1))
share = t.sum(axis=1) / t.values.sum() * 100 # Percent of Total, Table (down)
print((share).round(1).to_dict())
```

OUTPUT

month	2025-01	2025-02	2025-03	2025-12
category				
Furniture	12.9	16.8	28.7	127.6
Office Supplies	2.6	4.8	9.4	39.7
Technology	38.0	70.8	119.4	616.5
{ 'Furniture': 16.3, 'Office Supplies': 5.1, 'Technology': 78.7 }				

Common table-calculation functions you write yourself: RUNNING_SUM(SUM([Sales])), WINDOW_AVG(SUM([Sales]), -2, 0) (3-month moving average), LOOKUP(SUM([Sales]), -1) (previous value), RANK(SUM([Sales])), INDEX(), SIZE(), TOTAL(SUM([Sales])).

YoY growth as a calculation: (SUM([Sales]) - LOOKUP(SUM([Sales]), -1)) / ABS(LOOKUP(SUM([Sales]), -1)) with YEAR(Order Date) in the view.

Python

```
y = df.groupby(df.order_date.dt.year).sales.sum()
print(((y / y.shift(1) - 1) * 100).round(1).dropna().to_dict(), "% YoY")
```

OUTPUT

```
{2024: 13.5, 2025: 19.8} % YoY
```

Level of Detail (LOD) expressions

LODs compute an aggregate at a level **you** choose, regardless of the dimensions in the view.

Syntax	Meaning	Example
{FIXED [Dim] : AGG([Measure])}	Exactly at [Dim], ignoring the view (but after context filters)	{FIXED [Customer ID] : MIN([Order Date])} first purchase
{INCLUDE [Dim] : AGG(...)}	View's dimensions plus [Dim]	Average sales per customer by region
{EXCLUDE [Dim] : AGG(...)}	View's dimensions minus [Dim]	Region total next to each category, for % of region
{ : SUM([Sales])}	Table-scoped (grand total)	% of grand total without table calcs

FIXED: customer cohorts by first-purchase year

[First Purchase] = {FIXED [Customer ID] : MIN([Order Date])}, then YEAR([First Purchase]) on Rows and COUNTD([Customer ID]) gives new customers per year: an acquisition cohort.

Python

```
first = df.groupby("customer_id").order_date.min().rename("first_purchase")
d2 = df.merge(first, on="customer_id")
cohort = d2.groupby([d2.first_purchase.dt.year.rename("cohort"),
d2.order_date.dt.year.rename("order year")]).customer_id.nunique().unstack(fill_value=0)
cohort
```

OUTPUT

order year	2023	2024	2025
cohort			
2023	540	397	415
2024	0	193	156
2025	0	0	52

3 rows × 3 columns

Read it row by row: customers acquired in 2023 (first row) and how many of them bought again in 2024 and 2025. This retention table is a favourite interview task.

INCLUDE: average sales per customer by region

AVG({INCLUDE [Customer ID] : SUM([Sales])}) with Region in the view first sums sales per customer *within* each region, then averages those customer totals.

Python

```
per_cust = df.groupby(["region", "customer_id"]).sales.sum().groupby("region").mean() / 1e3
print((per_cust).round(1).rename("avg sales per customer (₹ thousand)"))
```

OUTPUT

region

East 238.4

North 263.5

South 261.9

West 262.4

Name: avg sales per customer (₹ thousand), dtype: float64

EXCLUDE: share of the region

With Region and Category in the view, SUM([Sales]) / SUM({EXCLUDE [Category] : SUM([Sales])}) gives each category's share of its region.

Python

```
rc = df.groupby(["region", "category"]).sales.sum()
(rc / rc.groupby(level=0).transform("sum") * 100).round(1).unstack()
```

OUTPUT

category	Furniture	Office Supplies	Technology
region			
East	17.1	5.9	77.1
North	16.3	4.5	79.2
South	17.8	4.6	77.6
West	18.0	5.2	76.8

4 rows × 3 columns

Order of operations

Filters and calculations run in a fixed order. This explains most "why did my number not change?" questions:



Tableau's order of operations (simplified). FIXED LODs ignore normal dimension filters unless the filter is added to context.

TIP Make a FIXED LOD respect a filter

A FIXED LOD is computed before dimension filters, so filtering Region does not change {FIXED [Customer ID] : MIN([Order Date])}. To make it respect the filter, right-click the filter → **Add to Context**.

Summary

KEY TAKEAWAYS

- Row-level calculations run per row; aggregate calculations run on marks: compute margins as SUM/SUM.
- Table calculations work on the view's numbers; Compute Using sets the direction (addressing) and the restart points (partitioning).
- FIXED sets an exact level; INCLUDE adds a level; EXCLUDE removes one.
- FIXED LODs run before dimension filters; use context filters when they must respect a filter.

Quick quiz

6 questions · tap an answer to check it

1. Which formula gives the correct profit margin for a category?

A $\text{SUM}([\text{Profit}]) / \text{SUM}([\text{Sales}])$

B $\text{AVG}([\text{Profit}] / [\text{Sales}])$

C [Profit] / [Sales] summed

D $\text{MAX}([\text{Profit}]) / \text{MAX}([\text{Sales}])$

Answer: A. Divide the totals; do not average row ratios.

2. You want each customer's first order date regardless of the view. Which expression?

- A** {FIXED [Customer ID] : MIN([Order Date])}
- B {INCLUDE [Customer ID] : MIN([Order Date])}
- C RUNNING_MIN([Order Date])
- D MIN([Order Date])

Answer: A. FIXED computes at exactly the customer level.

3. A running total restarts for every category. Why?

- A** Category is partitioning the calculation (Compute Using is across months)
- B The data is wrong
- C It is a FIXED LOD
- D The filter is in context

Answer: A. Fields not used for addressing partition the calculation.

4. Your FIXED LOD ignores the Region filter. How do you fix it?

- A** Add the Region filter to context
- B Use a measure filter
- C Change to a table calculation
- D Hide the field

Answer: A. Context filters run before FIXED LODs.

5. Which table-calculation function returns the previous value in the partition?

A LOOKUP(SUM([Sales]), -1)

B INDEX()

C SIZE()

D TOTAL(SUM([Sales]))

Answer: A. LOOKUP with offset -1 looks one mark back.

6. Which expression gives each category's share of its region with Region and Category in the view?

A SUM([Sales]) / SUM({EXCLUDE [Category] : SUM([Sales])})

B SUM([Sales]) / SUM([Sales])

C {FIXED [Category] : SUM([Sales])}

D RANK(SUM([Sales]))

Answer: A. EXCLUDE removes Category, leaving the region total.

EXERCISE Practice

Create: (1) a "Profit status" field that labels each sub-category Loss or Profit and colour a bar chart with it; (2) a 3-month moving average of monthly sales; (3) a cohort table of customers by first-purchase year and order year; (4) a percent-of-region field with EXCLUDE; (5) a Top-N rank of products by profit.

Show solution (worked answer)

(1) IF SUM([Profit]) < 0 THEN "Loss" ELSE "Profit" END on Color. (2) WINDOW_AVG(SUM([Sales]), -2, 0) with continuous MONTH on Columns. (3) {FIXED [Customer ID] : MIN([Order Date])}, YEAR of it on Rows, YEAR(Order Date) on Columns, COUNTD([Customer ID]) on Text. (4) As in the lesson. (5) RANK(SUM([Profit])) computed along Product Name, then filter rank ≤ 10 (a table-calculation filter).

INTERVIEW CORNER Questions you may be asked

Q: Explain FIXED, INCLUDE and EXCLUDE. FIXED aggregates at exactly the listed dimensions, ignoring the view; INCLUDE adds dimensions to the view's level (finer), then the result is re-aggregated; EXCLUDE removes dimensions from the view's level (coarser). FIXED runs before dimension filters; INCLUDE and EXCLUDE run after them.

Q: Table calculation vs LOD expression? A table calculation runs last, on the aggregated numbers in the view, and depends on layout (Compute Using). An LOD runs in the data source query at a level you define and can be used as a dimension, e.g. for cohorts.

Q: Why might a running total be wrong? Usually Compute Using: it is computing along the wrong field, or a dimension is partitioning it. Check Edit Table Calculation → Specific Dimensions and the sort order.

Filters, Parameters, Sets, Groups and Hierarchies in Tableau

Control what a view shows and let users interact with it. Learn every filter type and where it sits in the order of operations, context filters and Top N, groups, hierarchies, bins, sets (fixed, conditional, top) and combined sets, and parameters for dynamic Top N, measure swapping and what-if analysis.

✦ Tableau Desktop / Tableau Public 2026

Intermediate

🕒 120 min

By the end of this lesson you can

- Use dimension, measure, date, context, data-source and table-calculation filters correctly
- Build hierarchies and groups, and explain groups vs sets
- Create fixed, conditional and Top N sets and use IN/OUT in calculations
- Create parameters for Top N, measure selection and what-if discount scenarios
- Combine filters, sets and parameters into interactive views

Filters

Filter	How	Notes
Dimension filter	Drag a dimension to Filters → General / Wildcard / Condition / Top tabs	e.g. Region = South
Measure filter	Drag a measure to Filters → choose aggregation → range	e.g. SUM(Profit) < 0 (loss-making items)
Date filter	Relative dates (last 3 months), range of dates, or discrete years/months	Relative dates update automatically
Context filter	Right-click a filter → Add to Context (grey pill)	Runs first; Top N and FIXED LODs respect it
Data source filter	Data Source page → Filters → Add	Applies to every sheet; good for security and speed
Extract filter	When creating an extract	Shrinks the extract itself
Table - calc filter	Filter on a table calc such as <code>LAST()=0</code> or rank	Hides marks without changing other calculations

Show Filter (right-click the pill → Show Filter) puts an interactive control on the sheet: single value list, dropdown, multiple values, slider, wildcard match. **Apply to Worksheets → All Using This Data Source / Selected Worksheets** makes one filter drive several sheets on a dashboard.

Python

```
import pandas as pd, numpy as np
df = pd.read_csv("superstore_sales.csv", parse_dates=["order_date"])
loss = df.groupby("sub_category").profit.sum().sort_values()
print("Loss-making sub-categories (measure filter SUM(Profit) < 0):")
print((loss[loss < 0] / 1e5).round(1))
```

OUTPUT

```
Loss-making sub-categories (measure filter SUM(Profit) < 0):
sub_category
Tables      -8.2
Bookcases   -4.2
Name: profit, dtype: float64
```

Top N and context

Top N is computed **after** context filters but **before** ordinary dimension filters. So "Top 5 products **in the South**" needs the Region filter in context; otherwise Tableau finds the national Top 5 first and

then keeps only those that happen to be in the South.

Python

```
top_all = df.groupby("product_name").sales.sum().nlargest(5)
south = df[df.region == "South"]
top_south = south.groupby("product_name").sales.sum().nlargest(5)
print("National top 5:", list(top_all.index))
print("South top 5 (Region in context):", list(top_south.index))
print("Overlap:", len(set(top_all.index) & set(top_south.index)), "of 5")
```

OUTPUT

National top 5: ['Orbit Copier 746', 'Apex Copier 347', 'Vertex Copier 262', 'Zenith Copier 719', 'Zenith Machine 401']

South top 5 (Region in context): ['Orbit Copier 746', 'Vertex Copier 262', 'Zenith Machine 401', 'Apex Copier 347', 'Pioneer Copier 366']

Overlap: 4 of 5

Hierarchies, groups and bins

- **Hierarchy**: drag Sub-Category onto Category in the Data pane → name it "Product hierarchy". A + on the pill drills down. Date fields have a built-in Year → Quarter → Month → Day hierarchy.
- **Group**: select marks or right-click a field → Create → Group, e.g. group Indian states into "North-East". A group is a new dimension; its members are fixed.
- **Bins**: equal-width buckets of a measure (Lesson 2).

Sets

A **set** is a subset of a dimension that is either **IN** or **OUT**.

Set type	Created by	Example
Fixed (manual)	Select members	Six key accounts
Conditional	Condition tab, e.g. SUM(Profit) > 0	Profitable customers
Top	Top tab, e.g. Top 10 by SUM(Sales)	Top 10 customers
Combined	Two sets → Create Combined Set (union, intersection, difference)	Top customers who are also unprofitable

Use a set on **Color** to show IN vs OUT, on Filters to keep only members, or in calculations: `IF [Top Customers] THEN [Sales] END` . **Set actions** (Lesson 6) let users change a set by clicking marks.

Python

```
cust = df.groupby("customer_id").agg(sales=("sales", "sum"), profit=("profit", "sum"))
top10 = cust.sales.nlargest(10).index
in_set = cust.index.isin(top10)
print(f"Top 10 customers (IN): {cust.sales[in_set].sum() / cust.sales.sum():.1%} of sales")
unprofitable = cust.index[cust.profit < 0]
print("Customers in Top 10 AND unprofitable (combined set, intersection):", len(set(top10)
& set(unprofitable)))
print("All unprofitable customers:", len(unprofitable), "of", len(cust))
```

OUTPUT

Top 10 customers (IN): 5.3% of sales

Customers in Top 10 AND unprofitable (combined set, intersection): 1

All unprofitable customers: 142 of 785

NOTE Groups vs sets

A **group** combines members into new categories (many buckets, fixed). A **set** splits a dimension into IN/OUT, can be dynamic (conditional or Top N) and can be changed by users with set actions.

Parameters

A **parameter** is a single user-controlled value (number, string, date or list) that does nothing until a calculation, filter or reference line uses it. Right-click → **Show Parameter**.

1. Dynamic Top N

1. Create parameter **Top N** (integer, range 5–20, step 5).
2. Product Name filter → Top tab → By field → Top **[Top N]** by SUM(Sales).
3. Show the parameter: users choose 5, 10, 15 or 20.

2. Measure swap

```
// Parameter "Choose measure": list of strings Sales, Profit, Quantity
CASE [Choose measure]
WHEN "Sales" THEN SUM([Sales])
WHEN "Profit" THEN SUM([Profit])
WHEN "Quantity" THEN SUM([Quantity])
END
```

Put this calculated field on the axis; one chart now shows whichever measure the user picks. Use a dynamic title: `<Parameters.Choose measure> by Region` .

3. What-if: extra discount

```
// Parameter "Extra discount" : float 0.00-0.20
[Sales] * (1 - [Extra discount])           // projected sales
[Profit] - [Sales] * [Extra discount]      // projected profit
```

Python

```
for extra in (0, .05, .10):
    proj_profit = (df.profit - df.sales * extra).sum()
    print(f"extra discount {extra:.0%}: projected profit ₹{proj_profit / 1e7:.2f} crore")
```

OUTPUT

```
extra discount 0%: projected profit ₹1.84 crore
extra discount 5%: projected profit ₹0.83 crore
extra discount 10%: projected profit ₹-0.18 crore
```

With margins around 9%, a further 10% discount turns the whole business loss-making. That is exactly the kind of insight a what-if parameter makes obvious to a manager.

2026 UPDATE Dynamic parameters and zone visibility

Parameters can refresh their list and default value from a field when the workbook opens (*When workbook opens* → *value of field*). Combined with **dynamic zone visibility** (show or hide a dashboard container based on a boolean field or parameter), they power menus and drill-in panels without bookmarks.

Summary

KEY TAKEAWAYS

- Pick the right filter level: data source → context → dimension → measure → table calc.
- Top N and FIXED LODs ignore normal dimension filters unless those filters are in context.
- Groups make fixed buckets; sets make IN/OUT subsets that can be dynamic and combined.
- Parameters are user inputs: wire them into filters, calculations, titles and reference lines.

Quick quiz

6 questions · tap an answer to check it

1. You need "Top 10 products within the selected region". What must you do?

A Add the Region filter to context

B Use a measure filter

C Use a group

D Hide the other regions

Answer: A. Top N runs before normal dimension filters.

2. Which object can a user change by clicking marks on a dashboard?

A A set (through a set action)

B A group

C A bin

D A hierarchy

Answer: A. Set actions update set membership.

3. A parameter on its own does nothing. Why?

- A** It must be referenced by a calculation, filter or reference line
- B** It needs an extract
- C** It must be added to context
- D** It only works on Tableau Server

Answer: A. Parameters are inputs that something must use.

4. Which filter applies to every sheet using the data source and is good for row-level restrictions?

- A** Data source filter
- B** Table-calculation filter
- C** Measure filter
- D** Page shelf

Answer: A. Data source filters run near the start for all sheets.

5. Which set type keeps the 10 largest customers by sales, updating as data changes?

- A** Top set
- B** Fixed set
- C** Group
- D** Combined set

Answer: A. The Top tab makes a dynamic set.

6. Which calculation swaps the measure shown in a chart?

A A CASE on a string parameter returning SUM([Sales]), SUM([Profit]) or SUM([Quantity])

B A FIXED LOD

C A group

D A data source filter

Answer: A. The parameter value chooses the branch.

EXERCISE Practice

Build: (1) a dynamic Top N products chart with a parameter and Region in context; (2) a Top 20 customers set shown on Color in a scatter of Sales vs Profit; (3) a measure-swap chart with a dynamic title; (4) a what-if discount parameter showing projected profit by Category.

Show solution (worked answer)

Follow the three parameter recipes. For (2): Customer Name on Detail, SUM(Sales) and SUM(Profit) on the axes, the Top 20 set on Color. For (4): a "Projected profit" calculated field $\text{SUM}([\text{Profit}]) - \text{SUM}([\text{Sales}]) * [\text{Extra discount}]$ by Category with the parameter shown as a slider.

INTERVIEW CORNER Questions you may be asked

Q: Difference between a set and a group? A group merges members into new categories and is static; a set divides members into IN/OUT, can be conditional or Top N, can be combined with other sets and changed through set actions.

Q: What is a context filter and when do you use it? A filter that runs before other dimension filters, Top N filters and FIXED LODs. Use it for dependent Top N, to make FIXED LODs respect a filter, and to speed up large workbooks by reducing the data early.

Q: Parameter vs filter? A filter removes data; a parameter is a single input value that you can use anywhere: in calculations, Top N, reference lines, titles, or to swap measures and dimensions.

Data Modelling in Tableau: Relationships, Joins, Unions, Blending, Extracts and Tableau Prep

Combine and shape data in Tableau. Understand the logical layer (relationships) and the physical layer (joins), when to union or blend, how row duplication inflates totals, live connections vs extracts, data interpreter and pivoting, and how Tableau Prep Builder cleans data in repeatable flows, using the Adventure Cycles star schema.

✦ Tableau Desktop / Tableau Public 2026

Intermediate

🕒 120 min

By the end of this lesson you can

- Build a relationships-based data model and explain why it avoids duplicated totals
- Choose between relationships, joins, unions and blends
- Pivot, split and clean data on the Data Source page and in Tableau Prep
- Decide between live connections and extracts, and plan refreshes
- Explain the logical vs physical layer in an interview

The two layers of a Tableau data source

Logical vs physical

Logical layer (relationships, 'noodles')

- Tables stay separate
- Join happens at query time, at the right level
- No duplicated measures
- Default since Tableau 2020.2

Physical layer (joins, unions)

- Double-click a logical table to enter it
- Tables merged into one flat table first
- Can duplicate rows and inflate SUMs
- Needed for some calculations and older workbooks

Relationships live in the logical layer; joins and unions live inside a logical table (physical layer).

When you drag a second table onto the canvas, Tableau draws a **relationship** (a flexible "noodle") and asks for the matching fields. Tableau then generates the right join for each sheet, using only the tables that sheet needs and aggregating each table at its own level.

The Adventure Cycles star schema

The same star schema used in the Power BI lessons: one fact table of 9,000 sales lines and four dimension tables.

Python

```
import pandas as pd
base = "adventure_cycles/"
fact = pd.read_csv(base + "fact_sales.csv")
dims = {n: pd.read_csv(base + f"dim_{n}.csv") for n in ["product", "territory", "customer",
"date"]}
print("fact_sales:", fact.shape, list(fact.columns))
for n, d in dims.items(): print(f"dim_{n}:", d.shape, list(d.columns)[:6])
```

OUTPUT

```
fact_sales: (9000, 9) ['order_number', 'order_date', 'product_key', 'customer_key',
'territory_key', 'quantity', 'unit_price', 'sales_amount', 'total_cost']
dim_product: (45, 6) ['product_key', 'product', 'subcategory', 'category',
'list_price', 'standard_cost']
dim_territory: (6, 3) ['territory_key', 'territory', 'country']
dim_customer: (1200, 5) ['customer_key', 'customer_name', 'gender', 'age_band',
```

```
'territory_key']  
dim_date: (1096, 6) ['date', 'year', 'quarter', 'month_no', 'month', 'weekday']
```

STEP BY STEP Build the model with relationships

1. Connect to the folder of CSVs (Text file → choose `fact_sales.csv`). Drag it to the canvas.
2. Drag `dim_product.csv` next to it; Tableau suggests the relationship on the shared key (ProductKey). Check it in the relationship dialog.
3. Repeat for territory, customer and date.
4. Go to a sheet: the Data pane shows each table as a folder. Build Sales by Category: Tableau relates only fact and product for this view.

Why joins can inflate totals

A join merges tables **before** aggregation. If a table on one side has several matching rows, the other side's rows are repeated, and SUM counts them more than once. Relationships avoid this because each table is aggregated at its own level before combining.

Python

```
# A targets table at territory + year level (one row per territory per year), related to  
# line-level sales  
fact["year"] = pd.to_datetime(fact.order_date).dt.year  
targets = fact[["territory_key", "year"]].drop_duplicates().assign(target=10_00_00_000) #  
₹10 crore each  
joined = fact.merge(targets, on=["territory_key", "year"])  
# physical join at line level  
print("territories x years:", len(targets), "| sales lines:", len(fact))  
print("true total target: ₹", f"{targets.target.sum():,}")  
print("SUM(target) after a row-level join: ₹", f"{joined.target.sum():,}")  
print("inflation factor:", round(joined.target.sum() / targets.target.sum()), "x")
```

OUTPUT

```
territories x years: 18 | sales lines: 9000  
true total target: ₹ 1,800,000,000  
SUM(target) after a row-level join: ₹ 900,000,000,000  
inflation factor: 500 x
```

The join repeats each territory's target once per sales line, so SUM(target) is wildly overstated. With a **relationship** between the sales table and a targets table on territory and year, Tableau aggregates targets at their own level and the total is correct.

Relationships, joins, unions and blends

Method	Combines	When to use
Relationship	Tables at different levels, related on keys	Default for star schemas and multi-level data
Join (inner, left, right, full)	Columns from tables into one flat table	Same grain, or when you need row-level calcs across tables
Union	Rows of tables with the same columns	Monthly files, one sheet per year
Blend	Two separate data sources, linked in a sheet on a common dimension (orange chain icon)	Sources that cannot be joined (e.g. a Google Sheet of targets with a SQL database); aggregated before linking

TIP

Wildcard union

On the Data Source page, drag **New Union** onto a table and choose **Wildcard (automatic)** with a pattern such as `sales_2025_*.csv` . Every new monthly file in the folder is picked up on refresh, and Tableau adds a *Path* or *Sheet* column so you know where each row came from.

Cleaning on the Data Source page

Task	Feature
Messy Excel with titles, blank rows, merged cells	Use Data Interpreter
Months as columns (Jan, Feb, ...)	Select the columns → Pivot → rename "Pivot Field Names" to Month
"City, State" in one column	Column menu → Split / Custom Split
Wrong types	Click the type icon
Unneeded columns	Hide

Python

```
wide = pd.DataFrame({"region": ["East", "North"], "Jan": [120, 140], "Feb": [130, 150], "Mar": [125, 160]})
long = wide.melt(id_vars="region", var_name="month", value_name="sales") # Tableau Pivot
print(wide, "\n"); print(long)
```

OUTPUT

```
region Jan Feb Mar
0 East 120 130 125
1 North 140 150 160
```

```
region month sales
0 East Jan 120
1 North Jan 140
2 East Feb 130
3 North Feb 150
4 East Mar 125
5 North Mar 160
```

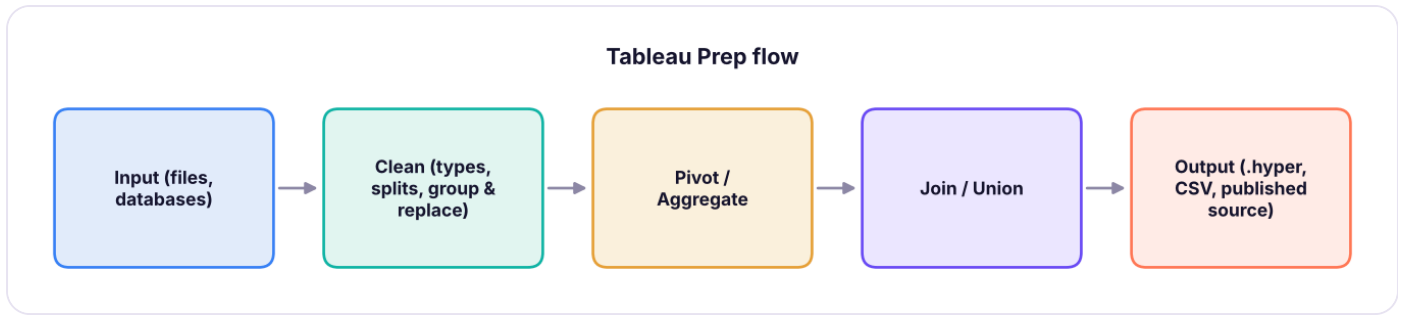
Live connection vs extract

	Live	Extract (.hyper)
Freshness	Always current	As of the last refresh
Speed	Depends on the database	Fast, columnar, compressed
Works offline	No	Yes
Good for	Real-time ops, governed warehouses with fast engines (Snowflake, BigQuery, Databricks)	Most dashboards, files, slow sources, Tableau Public
Refresh	Not needed	Full or incremental, scheduled on Tableau Cloud/Server

Extract options: keep only needed fields (**Hide all unused fields**), add extract filters, aggregate to visible dimensions, and use **incremental refresh** on a date or ID column for large tables.

Tableau Prep Builder

Prep is a separate app for repeatable cleaning flows. A flow is a left-to-right chain of steps:



A typical Tableau Prep flow.

Useful steps: **Clean** (profile pane shows each field's distribution; fix types, trim spaces, *Group and Replace* to merge spelling variants such as "Bengaluru"/"Bangalore"), **Aggregate**, **Pivot** (columns to rows and back), **Join** (with a join-clause summary that shows unmatched rows), **Union**, **New Rows** (fill missing dates) and **Script** (Python/R). Outputs can be scheduled with Tableau Prep Conductor on Cloud/Server.

Summary

KEY TAKEAWAYS

- Relationships (logical layer) are the default; they join at query time at the right level and avoid duplicated totals.
- Joins flatten tables first and can inflate SUMs when grains differ; unions stack rows; blends link separate sources per sheet.
- Use Data Interpreter, Pivot and Split for quick fixes; use Tableau Prep for repeatable flows.
- Extracts are fast snapshots; live connections are always current.

Quick quiz

6 questions · tap an answer to check it

1. Sales lines are joined to a territory-level targets table and SUM(Target) is far too high. Why?

A The join repeated each target once per sales line

B Targets are text

C The extract is stale

D A context filter is missing

Answer: A. Different grains duplicate rows; use a relationship.

2. Twelve monthly CSVs with identical columns should be combined. Which method?

A Union (ideally a wildcard union)

B Join

C Blend

D Relationship

Answer: A. Unions stack rows of same-shaped tables.

3. When is data blending the right choice?

A When two sources cannot be joined in one data source and you link them on a common dimension in a sheet

B For every star schema

C To union files

D To speed up extracts

Answer: A. Blends combine separate sources after aggregation.

4. Your Excel sheet has a title row and merged headers. What helps first?

A Data Interpreter

B A FIXED LOD

C A parameter

D A table calculation

Answer: A. Data Interpreter detects the real table.

5. Which is TRUE about extracts?

A They are fast compressed snapshots that need refreshing

B They always show real-time data

C They cannot be filtered

D They only work on Tableau Server

Answer: A. Schedule full or incremental refreshes.

6. Where do relationships live?

A In the logical layer

B In the physical layer

C In Tableau Prep only

D In the Marks card

Answer: A. Double-click a logical table to see its physical joins.

EXERCISE Practice

Build a relationships model of the Adventure Cycles tables and a sheet of Sales by Category and Country. Then create a small targets table (Country, Year, Target) in Excel, relate it on Country and Year, and show Sales vs Target as a bullet chart. Finally, simulate the join problem: join the targets instead, and compare SUM(Target).

Show solution (worked answer)

Relate fact_sales to each dimension on its key; relate the targets table to the territory dimension on Country (and to the date table on Year). With relationships, SUM(Target) by Country equals the targets you typed. With a physical join to the fact table, each target repeats per sales line and the sum is inflated.

INTERVIEW CORNER Questions you may be asked

Q: Relationships vs joins in Tableau? Relationships keep tables separate in the logical layer and Tableau generates the right joins per query, aggregating each table at its native level, so totals are not duplicated. Joins merge tables in the physical layer into one table before aggregation, which can duplicate rows when grains differ.

Q: When would you use data blending? When the data is in separate sources that cannot be related in one data source, such as a published data source plus a local spreadsheet of targets. Blends link on common dimensions and aggregate the secondary source first; they are slower and less flexible than relationships.

Q: How do you make a slow dashboard faster? Use an extract with only needed fields and rows, data source or context filters, fewer marks and quick filters, avoid high-cardinality filters showing all values, and check the Performance Recorder.

Dashboards, Actions, Stories and Publishing in Tableau (Capstone)

Turn sheets into an interactive dashboard and a data story. Plan the layout with containers, add filter, highlight, URL, parameter and set actions, design for phones, tell a story with Story Points, publish to Tableau Public or Cloud, and complete two capstones, a Superstore India sales dashboard and the video-game regional sales case study from class.

✦ Tableau Desktop / Tableau Public 2026

Intermediate

🕒 180 min

By the end of this lesson you can

- Lay out a dashboard with tiled and floating objects, containers and device layouts
- Add filter, highlight, URL, go-to-sheet, parameter and set actions
- Build a Story with Story Points that explains an insight step by step
- Publish to Tableau Public or Cloud and set up refresh, permissions and subscriptions
- Deliver an executive dashboard and a written insight summary

Plan before you build

Dashboard workflow



The order that keeps dashboards clear: questions first, layout last.

People read a screen in a **Z-pattern**: top-left first. Put the KPI cards across the top, the main trend or map in the middle, and the detail (tables, Top N) at the bottom. Keep to about 5–7 views and one colour meaning per dashboard (for example orange = loss everywhere).

Building the dashboard

Element	Use
Size	Fixed (e.g. 1200 × 800) for predictable layouts, Automatic or Range for flexible screens
Tiled vs floating	Tiled objects snap into a grid; floating objects sit on top (legends, logos). Prefer tiled.
Containers	Horizontal and vertical containers group objects; <i>Distribute Contents Evenly</i> keeps KPI cards equal
Objects	Text (titles), Image (logo), Web Page (not on Tableau Public), Blank (spacing), Navigation and Download buttons, Extension
Show/Hide button	Collapsible filter panel on a floating container
Device layouts	Device Preview → Phone: stack KPIs and one chart vertically

KPI cards

Each card is a sheet: the measure on **Text**, Mark type Text, a formatted label such as `₹<SUM(Sales)> cr`, and a small YoY calculation in the second line with ▲/▼ and colour.

Python

```
import pandas as pd, numpy as np
df = pd.read_csv("superstore_sales.csv", parse_dates=["order_date"])
y = df.groupby(df.order_date.dt.year).agg(sales=("sales", "sum"), profit=("profit", "sum"),
orders=("order_id", "nunique"), customers=("customer_id", "nunique"))
cur, prev = y.loc[2025], y.loc[2024]
for k in y.columns:
    unit = 1e7 if k in ("sales", "profit") else 1
    val = f"₹{cur[k] / unit:,.2f} cr" if unit > 1 else f"{cur[k]:.0f}"
    print(f"{k:10s} 2025: {val:>12s} YoY {cur[k] / prev[k] - 1:+.1%}")
print(f"margin      2025: {cur.profit / cur.sales:.1%} (2024: {prev.profit / prev.sales:.1%})")
```

OUTPUT

sales	2025:	₹7.84 cr	YoY +19.8%
profit	2025:	₹0.75 cr	YoY +22.1%
orders	2025:	1,232	YoY +13.5%
customers	2025:	623	YoY +5.6%
margin	2025:	9.6%	(2024: 9.4%)



The KPI row for the 2025 Superstore India dashboard (values from the data above).

Dashboard actions

Dashboard → Actions → Add Action:

Action	What happens	Example
Filter	Clicking a mark filters other sheets	Click a state on the map → the bar charts show only that state
Highlight	Matching marks stay bright, others fade	Hover a category to highlight it everywhere
Go to Sheet	Navigates to another dashboard	Click a KPI to open a detail page
URL	Opens a web page, with field values in the link	Open a product page or a Google search for the customer
Change Parameter	Clicking sets a parameter value	Click a region to set the "Selected region" parameter used in titles
Change Set Values	Clicking adds/removes marks from a set	Drill-in: show sub-categories only for the clicked category (set on a calculation)

Choose **Run action on: Select / Hover / Menu** and what happens **when clearing the selection** (show all values is the usual choice). Use *Use as Filter* (the funnel icon on a sheet) for the quick version.

TIP

The "drill-in" pattern with a set action

Create a set on Category; a calculation `IF [Category Set] THEN [Sub-Category] ELSE [Category] END` on Rows; a Change Set Values action targeting the set. Clicking a category expands it to its sub-categories in the same chart, and clicking again collapses it.

Stories

A **Story** is a sequence of **Story Points**, each a captioned snapshot of a sheet or dashboard with its filters. Use it for a presentation: "1. Sales grew 20% in 2025 → 2. Growth came from Technology in the South → 3. Tables and Bookcases lose money → 4. Recommendation".

Publishing and sharing

Destination	How	Notes
Tableau Public	File → Save to Tableau Public	Free; public; you can allow viewers to download the workbook; embed code for your portfolio site
Tableau Cloud / Server	Server → Publish Workbook	Choose project, permissions, embedded or prompted credentials, refresh schedule
Subscriptions and alerts	On the published view	Email snapshots on a schedule; alerts when a measure crosses a threshold
Export	Worksheet/Dashboard → Export Image, PowerPoint, PDF, Crosstab	For offline reports
Embedding	Share → copy embed code	Place a dashboard inside a website or portal

Before publishing: give every sheet a clear title, hide unused sheets (right-click the dashboard tab → Hide All Sheets), add tooltips in plain language, set the default filters, and test on the phone layout.

Capstone 1: Superstore India executive dashboard

Build a one-page dashboard for the Head of Sales:

- 1. **KPI row**: Sales, Profit, Margin, Orders for the selected year with YoY change.
- 2. **Trend**: monthly Sales (bars) and Margin (line) on a dual axis.
- 3. **Map**: filled map of profit margin by state (diverging palette).
- 4. **Category → sub-category** bars with the drill-in set action; loss-makers in orange.
- 5. **Top 10 customers** table with a Top N parameter.
- 6. Filters: Year (single value), Region, Segment, applied to all sheets.

Python

```
d25 = df[df.order_date.dt.year == 2025]
sub = d25.groupby("sub_category").agg(sales=("sales", "sum"), profit=("profit", "sum"))
sub["margin %"] = (sub.profit / sub.sales * 100).round(1)
print("2025 loss-making sub-categories:"); print(sub[sub.profit < 0].round(0))
st = d25.groupby("state").agg(sales=("sales", "sum"), profit=("profit", "sum")); st["margin %"] = st.profit / st.sales * 100
print("\nHighest and lowest margin states:", st["margin %"].idxmax(), f"{st['margin %'].max():.1f}%", "|", st["margin %"].idxmin(), f"{st['margin %'].min():.1f}%")
```

OUTPUT

2025 loss-making sub-categories:

	sales	profit	margin %
sub_category			
Bookcases	3918268.0	-169018.0	-4.0
Tables	5297064.0	-286188.0	-5.0

Highest and lowest margin states: West Bengal 13.7% | Telangana 7.5%

Write three insight sentences under the dashboard, for example: what grew, what lost money, and what you recommend (discount caps on loss-making sub-categories, focus on the highest-margin states).

Capstone 2: Regional video-game sales (the class case study)

The same business case you solved in Excel in class: *how do video-game sales vary across North America, Europe, Japan and other regions, and where are the opportunities?*

Python

```
vg = pd.read_csv("video_game_sales.csv")
print(vg.shape, "| years", vg.year.min(), "to", vg.year.max())
regions = ["na_sales", "eu_sales", "jp_sales", "other_sales"]
best_genre = vg.groupby("genre")[regions].sum().idxmax()
print("Best genre by region:", best_genre.to_dict())
vg["decade"] = (vg.year // 10 * 10).astype(str) + "s"
print((vg.groupby("decade")[regions].sum()).round(1))
```

OUTPUT

(2400, 11) | years 2010 to 2025

Best genre by region: {'na_sales': 'Sports', 'eu_sales': 'Sports', 'jp_sales': 'Role-Playing', 'other_sales': 'Sports'}

	na_sales	eu_sales	jp_sales	other_sales
decade				

2010s	312.5	249.4	136.8	108.9
2020s	188.1	148.9	83.8	62.6

Tableau tasks: 1. **Pivot** the four regional sales columns into Region / Sales on the Data Source page (select the columns → Pivot), so Region becomes one dimension. 2. Top 5 games per region: Region on Rows, Name on Rows, SUM(Sales) on Columns, a **Top 5 by Sales computed per region** using `RANK(SUM([Sales]))` with Compute Using = Name, filtered ≤ 5. 3. Regional sales by decade: a calculated field `STR(INT(YEAR([Year])/10)*10) + "s"` (or `([Year] // 10) * 10` if Year is a number) and a stacked bar coloured by Region. 4. Best genre in each region: highlight table of Genre × Region with the maximum per region marked using `WINDOW_MAX`. 5. A stacked bar of regional distribution over time with a clear title, then assemble all four into a dashboard with a Region filter action.

Summary

KEY TAKEAWAYS

- Plan the questions and the layout before building; use containers and a Z-pattern.
- Actions make dashboards interactive: filter, highlight, go to sheet, URL, parameter and set actions.
- Stories present insights step by step.
- Publish to Tableau Public for portfolios and to Cloud/Server for work, with refresh schedules and permissions.

Quick quiz

6 questions · tap an answer to check it

1. Which action makes clicking a state on a map filter every other chart?

A A filter action (or Use as Filter)

B A highlight action

C A URL action

D A story point

Answer: A. Filter actions pass the selected values to target sheets.

2. Which feature expands a category into its sub-categories in place when clicked?

A A set action with a calculation that switches between Category and Sub-Category

B A Story

C A blend

D A data source filter

Answer: A. Change Set Values drives the drill-in pattern.

3. Where should the KPI cards usually go?

A Across the top-left of the dashboard

B Bottom right

C In a hidden sheet

D In a tooltip only

Answer: A. Readers start top-left (Z-pattern).

4. What is a Story in Tableau?

A A sequence of captioned story points showing sheets or dashboards

B A type of chart

C An extract

D A calculated field

Answer: A. Stories present an analysis step by step.

5. Which is NOT available on Tableau Public?

A Keeping your workbook private on the server

B Saving to your public profile

C Dashboards with actions

D Calculated fields

Answer: A. Everything saved to Tableau Public is public.

6. How do you make one Region filter apply to all sheets on a dashboard?

A Apply to Worksheets → All Using This Data Source (or Selected Worksheets)

B Duplicate the filter on every sheet

C Use a parameter only

D Add it to context

Answer: A. One filter can drive many sheets.

EXERCISE Capstone deliverables

Publish both dashboards to Tableau Public. For each, write: the audience, three questions it answers, three insights with numbers, and two recommendations. Add the links to your resume and LinkedIn Featured section.

Show solution (worked answer)

Superstore: KPI row, dual-axis trend, margin map, drill-in category chart, Top N customers, three filters applied to all sheets, filter actions from the map and the category chart. Video games: pivoted regions, Top 5 per region (RANK table calc), decade stacked bar, genre × region highlight table, filter action on Region. Insights should quote numbers from your own views.

INTERVIEW CORNER**Questions you may be asked**

Q: How do you design a dashboard for executives? Start from their decisions: 3–5 KPIs with comparisons at the top, one trend, one breakdown, one exception list; consistent colours; filters applied to all sheets; tooltips that explain; a phone layout; and a short written insight.

Q: Filter action vs highlight action? A filter action removes non-selected data from target sheets; a highlight action keeps all data but fades the non-matching marks, preserving context.

Q: How do you share a Tableau dashboard securely? Publish to Tableau Cloud or Server with project permissions, use row-level security (user filters or entitlement tables with USERNAME()), schedule extract refreshes, and share via links, subscriptions or embedding with authentication. Never use Tableau Public for confidential data.