100 Tableau Interview Questions and Answers

APOORVA IYER

BASIC TABLEAU CONCEPTS

1. What is tableau and why is it preferred for data visualisation?

Tableau is a self-service Business Intelligence (BI) and analytics platform that lets users connect to almost any data source, transform that data visually, and share interactive insights without extensive coding. It is preferred because:

- **Speed to insight:** Drag-and-drop interface turns queries into visual answers in seconds, letting analysts iterate quickly during exploratory analysis.
- **Rich visual grammar:** Tableau's VizQL engine translates every user interaction directly into SQL/RMD queries, then renders results with automatic best-fit chart types—reducing manual chart-building effort.
- **Interactivity at scale:** End-users can filter, drill, and highlight live on the dashboard, enabling data-driven conversations rather than static reports.
- **Broad ecosystem:** Desktop for authoring, Cloud/Server for governance, Prep for data cleaning, Public/Extensions for community and custom apps—all integrated behind a single sign-in.
- **Performance optimisation:** Extract engine, in-memory columnar storage (.hyper), and query optimisation techniques keep large datasets responsive even on mid-range laptops.
- **Minimal coding barrier:** Analysts familiar with Excel pick up Tableau's calculations and LOD expressions quickly, reducing dependency on IT teams.

---

2. Explain "dimension" vs "measure."

- **Dimensions** are qualitative fields—text, dates, IDs, or categorical numbers—which slice the data into discrete buckets (e.g., *Region*, *Product Category*, *Invoice Date*). Placing a dimension on Rows, Columns or Marks creates headers, labels, or colour groups; it defines the **level of detail**.
- **Measures** are quantitative fields—usually numeric—on which aggregations such as SUM, AVG, MIN or MAX are applied (e.g., *Sales*, *Quantity*, *Profit Margin*).
  Why the distinction matters:

1. **Default aggregation:** Measures are aggregated automatically; dimensions are not.
2. **Visual encoding:** Dimensions often drive colour or shape, while measures drive size or length.

3. **Filter behaviour:** Filtering a dimension removes entire rows; filtering an aggregated measure can retain granular rows but hide marks post-aggregation (measure filters come later in the query pipeline).

---

3. Define discrete vs continuous fields and their impact on the view.

- **Discrete (blue pills):** Produce distinct, separate values—shown as headers or labels on axes. Example: *Order ID*, *Year(Order Date)*. They segment the canvas into finite partitions.
- **Continuous (green pills):** Produce an unbroken range—rendered as a continuous axis. Example: *SUM(Sales)*, *MONTH(Order Date)* treated as continuous.
  Impact: Switching a field's role changes the visual: *Order Date* as discrete *YEAR* creates column headers, whereas as continuous *MONTH* creates a time-series line with a date axis. This choice affects aggregation level, axis format, sorting, and tooltip context.

---

4. List tableau's core products (desktop, cloud/server, public, prep) and one key use-case for each.

- **Tableau Desktop:** Full-featured authoring tool installed locally. *Use-case:* Build a sales performance dashboard pulling from SQL Server and publish to Tableau Cloud.
- **Tableau Cloud (formerly Online) / Tableau Server:** Web-based governed repository for workbooks, data sources and user access controls. *Use-case:* Schedule daily extract refreshes and provide executives row-level-secure dashboards.
- **Tableau Public:** Free cloud edition with limited connectors and mandatory public gallery publishing. *Use-case:* Showcase a portfolio project (e.g., Indian election results visualisation) to recruiters.
- **Tableau Prep Builder / Conductor:** Visual data-prep tool for joins, pivots, cleaning; Conductor automates flows on Server/Cloud. *Use-case:* Merge CSV-based campaign spends with CRM leads, produce a cleaned .hyper file for Desktop analysis.

---

5. What file types do .twb and .twbx represent?

- **.twb (Tableau Workbook):** XML text file that stores metadata—data-source paths, sheet layouts, calculated fields—*but no actual data*. Small in size; requires access to original data for full rendering.
- **.twbx (Tableau Packaged Workbook):** Zipped bundle containing the .twb plus any extracted data (.hyper), local CSVs, shape files, images, and custom geocoding. It is self-contained, ideal for email transfer or uploading to Tableau Public but larger in size. Converting .twb to .twbx is as simple as *File ▸ Export Packaged Workbook*.

6. What are worksheets, dashboards and stories?

- **Worksheet:** Single view (chart or table) built from one data source. Analysts iterate here first.
- **Dashboard:** Canvas that arranges multiple worksheets, filters, legends and objects (images, web pages) into a cohesive analytical interface. Dashboards allow interactive actions across sheets.
- **Story:** Sequence of annotated "story points," each referencing a dashboard or worksheet, used to craft a narrative (e.g., quarterly performance walk-through). Stories anchor the audience, enforcing a guided order.

---

7. How do you create a hierarchy and why is it useful?

Creation: Drag a dimension (e.g., *State*) onto another dimension (e.g., *Country*) in the Data Pane ▸ give the hierarchy a name (e.g., *Geography*).
Usefulness:

- Enables one-click drill-down (Country → State → City) in visuals.
- Preserves parent-child relationship for automatic aggregations and filters.
- Improves workbook organisation—related fields collapse under a single expandable node, reducing clutter.

---

8. Differentiate groups, hierarchies and sets.

Feature	Purpose	How it works	Common use-case
Group	Combine similar members into a single category.	Manual or automated (e.g., "bin" unknown states into *Other*).	Consolidating sub-brands under a master brand for cleaner bar charts.
Hierarchy	Define parent-child drill path.	Drag fields to nest them.	Drill from Continent → Country → State in geo maps.
Set	Subset of dimension members defined by rules or manual pick.	Create Set → use IN/OUT logic.	Highlight Top 10 customers dynamically or create comparisons (Set vs Others).

Feature	Purpose	How it works	Common use-case
	Can be dynamic & used in calculations.		

---

9. Compare a live connection with an extract (.hyper) in terms of speed, scalability and data-freshness.

- **Live connection:** Queries the source database on every interaction. *Pros*: Real-time data, light workbook. *Cons*: Performance depends on network & DB horsepower; heavy load at peak times.
- **Extract:** Snapshot stored in columnar .hyper; Tableau queries locally. *Pros*: Faster rendering, offline analysis, supports incremental refresh. *Cons*: Data freshness limited to refresh schedule, larger file size.
  Choosing: Use live for operational dashboards where "latest five-minute sales" matters; use extracts for historical analytics, weekend executive decks, or when source system has query-limits.

---

10. What is a context filter in tableau and why would you create one?

A **context filter** is an initial filter that Tableau applies *before* any other dimension or measure filters in the query pipeline. Creating one: right-click a filter ▸ "Add to Context."
Reasons to use:

1. **Performance boost:** Reduces data volume early, so subsequent filters/processes (top-N, table calcs) work on a smaller subset.
2. **Dependent top-N:** When you need a Top 10 filter *within* a specific region, make Region the context so the Top N applies post-context.
3. **Conditional sets:** Ensures sets based on aggregated conditions are calculated after the context is applied, keeping results consistent.

DATA CONNECTIONS & MODELLING

11. Name three file-based and three server-based data sources tableau supports.

Tableau offers rich connectivity to both local files and remote servers, enabling analysts to work with almost any data environment.

File-based data sources:

- **Excel files (.xlsx, .xls):** Common in business reporting; drag-and-drop friendly.
- **Text/CSV files (.csv, .txt):** Used for flat exports or system logs.
- **JSON files:** Useful for semi-structured data or hierarchical datasets from web APIs.

Server-based or database sources:

- **Microsoft SQL Server:** Widely used enterprise database; supports custom SQL and stored procedures.
- **Oracle Database:** Known for large-scale transactional systems; robust for handling complex queries.
- **Google BigQuery or Amazon Redshift:** Cloud-native columnar databases optimised for analytical workloads.

These connectors appear in Tableau's native connection list and can be integrated using either live or extract-based connections.

---

12. Describe the steps to create and refresh an extract.

An extract in Tableau is a snapshot of the data stored in a compressed, columnar .hyper file, used to optimise speed and enable offline use.

Steps to create an extract:

1. **Connect to your data source** (e.g., Excel, SQL Server).
2. In the Data pane, click the drop-down next to the data source name and choose **"Extract."**
3. Optionally, **apply filters or aggregation** before extract creation to reduce size.
4. Click **"Sheet"** to trigger the extract creation dialogue.
5. Save the .hyper file to a chosen location.

Steps to refresh an extract:

- In **Tableau Desktop**: Right-click the data source and select **"Extract > Refresh."**
- In **Tableau Cloud/Server**: Set up a **refresh schedule** via the Data Management interface. You can choose full or incremental refresh, depending on what fields are available (e.g., a date or unique ID).

---

13. Inner vs left vs right vs full outer join—give practical examples.

Joins in Tableau let you combine tables based on shared fields (keys). You choose the join type visually in the Data Source tab.

- **Inner Join:** Returns only matching rows from both tables.
  Use-case: Match only customers who placed an order (Customer INNER JOIN Orders).
- **Left Join:** Returns all rows from the left table and matching rows from the right; unmatched right side returns NULL.
  Use-case: List all customers, even if they have no orders yet.
- **Right Join:** Opposite of left join; all from the right, matches from left.
  Use-case: Show all campaigns and matched user responses, even if some users weren't in CRM.
- **Full Outer Join:** All rows from both tables; unmatched records get NULLs.
  Use-case: Combine two marketing campaign lists to find overlap and gaps.

Joins are row-level operations, and selecting the right type avoids null values or data duplication.

---

14. What is a union and when is it preferable to a join?

A **union** stacks datasets vertically (i.e., adds rows), unlike a join, which adds columns.

Use-case for union:
When you have **multiple files or tables with the same schema** (e.g., monthly sales files from Jan to Dec), you use union to create one consolidated table.

Steps in Tableau:

- Drag one table onto the canvas.
- Drag another similar table **below it** until you see the **"Union"** prompt.
- Tableau will combine them and add a **Table Name** column automatically to distinguish sources.

Prefer union when:

- You have repeatable log data or form submissions.
- You're building a time-series across multiple time-period files.
- You need to compare KPIs month-over-month across the same structure.

---

15. Explain data blending and a scenario that requires it.

Data blending is Tableau's method for combining data from **two different sources** (unlike joins, which are within the same source). It links data on common fields using a **primary (blue tick)** and **secondary (orange tick)** data source.

Scenario:
You have sales data in **Excel** and customer demographic data in **Google Sheets**. Since they are from different systems, you cannot join them directly. Instead:

- Connect to both sources.
- Designate one as the primary (the one used in the view first).
- Drag a common field (e.g., Customer ID) into the view—Tableau blends using that key.

Blending is used when:

- The sources reside on **different platforms**.
- Data granularity differs (e.g., daily transactions vs. monthly targets).
- You want a **loose coupling**, not strict row-by-row joining.

---

16. How do relationships (logical layer) differ from physical joins?

Introduced in Tableau 2020.2, **relationships** are logical connections between tables that **preserve table-level granularity** and use **context-aware joins at query time**.

- **Relationships:** Defined in the **logical layer** of the Data Model. They defer the actual join until you build a view, allowing Tableau to generate **optimised queries** based on what fields you use.
- **Physical Joins:** Done in the **physical layer**, resulting in a fixed flattened table immediately, potentially causing row duplication or filtering issues.

Example:
If you relate *Orders* and *Returns* via Order ID, Tableau only queries *Returns* when return-related fields are used—this avoids unnecessary joins and ensures clean aggregation.

Advantages of relationships:

- More flexible for complex dashboards.
- Prevents data duplication.
- Simplifies LOD expressions.

---

17. When would you use custom sql instead of tableau's join canvas?

Custom SQL is used when:

- You need to **pre-aggregate or filter data** before Tableau reads it.
- You want to use **database-specific logic** like window functions, subqueries, or temp tables.

- You need to work with **complex business logic** not possible in Tableau's join UI.
- Data is too large and needs query optimisation at the DB level (e.g., only load last 6 months).

How to use it:
Click **"New Custom SQL"** in the Data Source tab and paste your SQL query. Tableau treats the result like a virtual table.

Caution: Custom SQL can bypass Tableau's optimisations and may reduce extract performance if not written efficiently.

---

18. Explain primary vs secondary data sources in a blend and how to switch them.

- **Primary data source**: The main source used to create the view first. It shows a **blue check mark** next to its name in the Data pane.
- **Secondary data source**: Any additional source used for blending (shows **orange link icon** when linked on a common field).

Switching roles:

- Start a new sheet and drag a field from the other data source first.
- That data source becomes the new primary.

Why it matters:

- Only **primary source's filters, groups, and parameters** apply fully.
- Secondary source fields may return NULLs if blending key values don't match exactly.
- LOD expressions can behave differently depending on the blending direction.

---

19. What is the data interpreter and when would you turn it on?

Data Interpreter is a tool used when connecting to Excel or CSV files that contain non-tabular elements like headers, merged cells, or footnotes.

What it does:

- Automatically **cleans** the file—detects the real header row, removes blank columns, and trims footnotes.
- Creates a **clean preview** table for analysis.
- Generates an **audit file** in Excel showing the changes it made.

When to use:

- You're importing a messy Excel report, not a database export.
- Column headers are offset by rows, or titles are merged.
- The file contains subtotals, signatures, or footer text.

You can enable it via the checkbox when previewing the data source.

---

20. Why is star-schema modelling recommended for tableau performance?

A **star schema** consists of one central **fact table** (e.g., transactions) connected to multiple **dimension tables** (e.g., customers, products, dates) via foreign keys. This design is recommended because:

- **Simplifies joins:** One-to-many joins from the fact table outward reduce risk of data duplication.
- **Supports modular queries:** Tableau can target only the tables required for a specific visual.
- **Optimised for extracts:** Extract engine can compress dimensions more efficiently.
- **Enhances LOD logic and filters:** Cleaner separation between metrics and context fields.
- **Reduces circular joins or ambiguous paths**, which are common when using snowflake schemas.

A well-modeled star schema helps Tableau generate faster, cleaner SQL, which is critical in enterprise-scale dashboards.

CALCULATIONS & TABLE CALCULATIONS

21. Describe the workflow to build a calculated field.

A **calculated field** in Tableau allows users to create new data columns or derive logic by writing formulas using existing fields. This is essential for creating metrics like profit ratio, sales categories, growth rates, or conditionally formatted segments.

Workflow to create a calculated field:

- **Open a worksheet** and ensure the data source is active.
- In the **Data pane**, right-click anywhere and select **"Create Calculated Field"** or click the drop-down arrow near the search bar.

- **Name your calculation** appropriately (e.g., Profit Ratio).
- In the formula editor, **write your expression** using Tableau functions, field names, and logical/arithmetical operators. Example:
- SUM([Profit]) / SUM([Sales])
- **Click OK**. Tableau will validate the formula syntax. If valid, the field will appear in the Data pane—generally under **Measures** if it's numeric.
- **Drag and drop** this new field into Rows, Columns, or the Marks card to use it like any other field.

Tips:

- Calculated fields can be reused across multiple sheets.
- Use the **Function list** on the right side of the editor to browse available functions.
- Start with **simple row-level** calculations before working with aggregate or table-based logic.

---

22. Row-level vs aggregate calculations—why can mixing them raise errors?

Row-level calculations operate on individual records **before aggregation**, while **aggregate calculations** operate on grouped data **after aggregation**.

Example:

- Row-level: [Sales] * [Discount] — this applies to each row.
- Aggregate: SUM([Sales]) / COUNT([Orders]) — this summarizes over multiple rows.

Mixing them directly causes errors like "Cannot mix aggregate and non-aggregate arguments" because Tableau cannot simultaneously calculate pre- and post-aggregation logic in the same expression.

To fix:

- Either aggregate the row-level field: SUM([Sales]) / SUM([Target])
- Or de-aggregate both sides using LODs or table calcs, depending on intent.
- Use **IF THEN logic outside aggregation** and **wrap aggregates only where needed**.

Understanding aggregation context is critical to avoid silent logic errors or failed calculations.

---

23. Give an example of a simple table calculation (running total).

Table calculations operate on **the result set in the view**, after aggregations are applied.

Example: Running Total of Sales

1. Drag Order Date to Columns (set to MONTH), and Sales to Rows.
2. Click on the SUM(Sales) pill → Quick Table Calculation → **Running Total**.
3. Tableau will calculate a cumulative sum across months.

Customising it:

- Edit table calculation to change **direction** (across, down, table, pane).
- Use **Secondary Calculation Type** to combine, e.g., Running Total with % of Total.

Use-cases of table calculations:

- Moving averages
- Percent of total
- Rank
- Difference from previous
- Year-over-year growth
  They are flexible but must be configured carefully depending on partitioning and addressing settings.

---

24. Define fixed, include and exclude lod expressions with use-cases.

Level of Detail (LOD) expressions control the **granularity at which a calculation happens**, independent of the view.

Types of LODs:

- **FIXED:** Performs calculation at a specified dimension level, regardless of the view.
  Example:
- { FIXED [Customer ID] : SUM([Sales]) }

Use-case: Calculate customer-level lifetime sales even if the view shows regions.

- **INCLUDE:** Adds dimension(s) to the view-level before performing aggregation.
  Example:
- { INCLUDE [Product] : AVG([Profit]) }

Use-case: Compute average product-level profit even if product isn't in the view.

- **EXCLUDE:** Removes dimension(s) from the view-level before aggregation.
  Example:
- { EXCLUDE [Customer ID] : SUM([Sales]) }

Use-case: Show overall sales per region without breaking down by customer.

Note: LODs are often preferred over table calcs for consistency across filters and better performance with extracts.

---

25. LOD VS TABLE CALCULATION—HOW DO YOU CHOOSE?

Criteria	Level of Detail (LOD)	Table Calculation
Timing	Happens in **data source query**	Happens **after** data is returned to Tableau
Filter impact	Can ignore view filters (e.g., FIXED)	Affected by view filters
Flexibility	Works across multiple sheets	Specific to the sheet/view it's created in
Use-case	Aggregate at dimension level regardless of view	Rank, running totals, moving average, YoY, etc.

Example decision:

- Use **LOD** if you want a consistent customer-level sales across views.
- Use **Table Calculation** for dynamic behaviors like YoY or % change based on the current layout.

---

26. Difference between if-else and case statements.

Both are conditional logic statements, but their structure and use-case differ.

IF-ELSE Syntax:

```
IF [Profit] < 0 THEN "Loss"
ELSE "Profit"
END
```

- Supports complex, nested, and multiple logical conditions.
- Allows use of operators like AND, OR, >, <, etc.

CASE Syntax:

```
CASE [Region]
  WHEN "East" THEN "Group A"
  WHEN "West" THEN "Group B"
```

```
  ELSE "Other"
END
```

- Simpler and cleaner when comparing a single field to multiple values.
- Doesn't support logical comparisons like > or !=.

When to use:

- Use **IF-ELSE** for complex multi-variable logic.
- Use **CASE** for cleaner categorical mappings.

---

27. What does attr() return and why is it handy?

ATTR() stands for **attribute** and is used when a dimension has **multiple values** in the current view context but Tableau expects **only one**.

What it returns:

- The value if **all rows have the same value** for that dimension.
- An asterisk (*) if there are **multiple values**.

Why it's useful:

- Prevents Tableau from throwing an error when dropping a dimension into a tooltip or label.
- Allows limited dimensions in aggregate contexts without collapsing the view.

Example:
When placing Customer Name into a tooltip on a bar chart summarised by Region, Tableau uses ATTR([Customer Name]) to avoid errors when multiple customers exist per region.

---

28. How do you create a date scaffold to fill missing dates?

When data is **missing certain dates** (e.g., no sales on weekends), a **date scaffold** ensures the time-series line doesn't break.

Steps to scaffold:

1. Create a new table with all possible dates (e.g., via Excel or in Tableau).
2. Join or relate it with your fact table on the date field.
3. Use the scaffold date field in the view (Columns/Rows).
4. Replace NULL measures with ZN() or custom IFNULL() logic.

This technique is useful for time-series analysis, seasonal comparisons, and smoothing gaps.

29. Explain zn() and its practical importance.

The ZN() function converts NULLs to zero (0).

Syntax:

ZN([Sales])

Why it matters:

- Avoids breaking calculations like percent of total or running sums.
- Helps create accurate totals and averages when missing values would otherwise cause misalignment.
- Essential in line charts—replaces NULL gaps with 0 to maintain continuity.

Often paired with IFNULL() and LOOKUP() to handle missing data gracefully.

---

30. How can window functions help in percent-of-total analysis?

WINDOW_SUM(), WINDOW_AVG(), etc., calculate aggregates over a specified range (partition) of the view.

Percent of total example:

SUM([Sales]) / WINDOW_SUM(SUM([Sales]))

Use-case steps:

1. Drag Region and Category to Rows.
2. Use above formula to show what percent each Category contributes within a Region.
3. Format as percentage.
4. Adjust **Compute Using** to correct scope (e.g., across Category for each Region).

Benefits:

- Dynamic and responsive to filters or changes in the view.
- Avoids pre-aggregating or manual calculations outside Tableau.
- Clean way to build KPIs or stacked visuals with relative context.

VISUALISATION TECHNIQUES

31. Best chart types for comparison, trend, distribution and relationship analyses.

Choosing the right chart type is critical to delivering clear and impactful insights. Tableau offers a range of visualisations tailored for specific analytical goals:

- **Comparison:**
  Bar charts, *bullet graphs*, and *highlight tables* are best when comparing values across categories.
  Use-case: Compare total sales across product categories.
- **Trend:**
  Line charts and *area charts* are ideal for showing changes over time.
  Use-case: Show monthly revenue growth over a fiscal year.
- **Distribution:**
  Histograms, *box plots*, and *density plots* reveal spread, clusters, and outliers.
  Use-case: Display distribution of order quantities or profit margins.
- **Relationship:**
  Scatter plots are perfect for finding correlations between two measures.
  Use-case: Explore relationship between discount percentage and profit.

Note: Use Tableau's "Show Me" panel as a guide, but always align visual choice to the story you're trying to tell.

---

32. Steps to build a dual-axis combo chart.

A **dual-axis chart** combines two measures with different scales into one visual, typically sharing the same dimension.

Example: Sales and Profit over time.

Steps:

1. Drag Order Date to Columns.
2. Drag Sales to Rows.
3. Drag Profit to the same Rows shelf **next to Sales** (not below). Tableau creates a second axis.
4. Right-click the second axis → choose **"Dual Axis."**
5. Right-click either axis → **Synchronize Axis** (if necessary).
6. Use the **Marks card** for each axis to set different chart types (e.g., bar for Sales, line for Profit).
7. Adjust colours and tooltips for clarity.

Tip: Dual-axis charts work best when comparing related metrics like volume and profitability, but avoid overloading with too many axes to preserve readability.

---

33. Create a heat-map to show correlation intensity.

A heat map uses colour shading to represent value intensity, allowing users to spot patterns across combinations of dimensions.

Steps:

1. Drag a **dimension** to Rows (e.g., Region).
2. Drag another **dimension** to Columns (e.g., Category).
3. Drag a **measure** (e.g., Sales) to **Colour** on the Marks card.
4. Change the Mark type to **Square**.
5. Adjust the colour gradient scale (e.g., light to dark) to reflect magnitude.

Use-case: Visualise sales intensity by region and product category—darker squares signal higher values.

Tip: Use diverging colour palettes for positive vs. negative metrics like profit/loss.

---

34. Build a funnel chart for conversion analysis.

Funnel charts are used to visualise **drop-offs** across a series of stages—commonly in sales pipelines or marketing funnels.

Steps (Quick method):

1. Create a table with stages (e.g., Lead, Contacted, Demo, Closed) and counts.
2. Drag **Stage** to Rows.
3. Drag **Count** to Columns.
4. Sort Stage manually in descending order.
5. Change Mark type to **Bar**.
6. Right-click axis → **Reversed Axis** (optional).
7. Center-align bars using the **Size** shelf or a dual-axis trick.

Tip: Tableau doesn't offer a native funnel chart, but by formatting bar widths or layering dual axes, you can simulate the funnel effect effectively.

---

35. When do you choose a gantt chart?

Gantt charts visualise task schedules and durations—excellent for project timelines or resource tracking.

Use-case: Show product shipment durations or project task planning.

Steps:

1. Drag Task Name to Rows.
2. Drag Start Date to Columns.
3. Change Mark type to **Gantt Bar**.
4. Drag Duration to Size shelf (calculated as End Date - Start Date).
5. Optional: Add colour by project or priority.

Tip: Gantt charts can also be used to visualise **overlaps**, identify schedule delays, and ensure time-based transparency.

---

36. How does tableau recognise geographic roles and plot maps?

Tableau automatically assigns **geographic roles** to fields like Country, State, City, Zip Code, etc.

Steps to plot a map:

1. Drag a geographic field (e.g., State) to **Rows**.
2. Tableau recognises it as geographic and converts it to a **Map view** automatically.
3. Drag a measure (e.g., Sales) to **Colour** or **Size**.
4. Use **Show Me** to convert to filled map or symbol map.

Geographic roles include:

- Country/Region
- State/Province
- City
- Zip Code
- Latitude / Longitude

Tips:

- You can assign a geographic role manually: Right-click the field > **Geographic Role** > Choose type.
- Use **custom geocoding** for proprietary locations (e.g., warehouse zones).

37. Use the pages shelf to animate year-by-year change.

The **Pages shelf** allows for temporal or categorical animations, helping explore trends step-by-step.

Steps:

1. Drag Year(Order Date) to the **Pages shelf**.
2. Tableau adds play/pause controls.
3. Use this control to scroll through years, see visual transition.
4. Optional: Add Region or Segment to view category-wise changes over time.

Use-case: Animate how sales evolved across regions year-by-year.

Tip: Pages shelf is local to Tableau Desktop and doesn't publish to Tableau Server or Cloud (it gets converted to filters).

---

38. Create a histogram for sales distribution.

Histograms show **frequency distribution** of a single numeric variable.

Steps:

1. Drag a continuous field (e.g., Sales) to Columns.
2. Right-click and choose **"Create > Bins…"**
    - Choose bin size or let Tableau decide.
3. Drag the new bin field to Columns.
4. Drag Number of Records to Rows.
5. Adjust bar spacing using Size shelf.

Use-case: See how order amounts are distributed—whether most orders are low-value or skewed to the right.

Tip: Use tooltips and reference lines to annotate average or outliers.

---

39. Explain "measure names" and "measure values."

- **Measure Names**: A **built-in field** that holds the **names** of all measures in your dataset.
- **Measure Values**: Holds the **actual values** of the measures in a stacked structure.

Use-case:
When you want to display **multiple measures** (e.g., Sales, Profit, Quantity) in a single chart or table.

Steps to use:

1. Drag Measure Names to Columns or Filters.
2. Drag Measure Values to Rows.
3. Adjust Marks card to show desired visual (e.g., bar chart, text table).

Tip: Useful for creating side-by-side KPIs, comparative bar charts, or control panels.

---

40. Treemap vs bar chart—when is each superior?

- **Treemap:**
  - Visualises **part-to-whole** relationships with nested rectangles.
  - Encodes values using **area**, not axis.
  - Best when comparing large numbers of categories at once.
  - **Use-case:** Show sales by product sub-category within a region.
- **Bar Chart:**
  - Easier to compare **precise values** using a shared baseline.
  - Preferred when **exact comparison or ranking** is needed.
  - **Use-case:** Show top 10 customers by revenue.

Guidance:
Use **bar chart** when accuracy matters and **treemap** when space or categorical spread is the priority.

VISUALISATION TECHNIQUES

31. Best chart types for comparison, trend, distribution and relationship analyses.

Choosing the right chart type is critical to delivering clear and impactful insights. Tableau offers a range of visualisations tailored for specific analytical goals:

- **Comparison:**
  Bar charts, *bullet graphs*, and *highlight tables* are best when comparing values across categories.
  Use-case: Compare total sales across product categories.
- **Trend:**
  Line charts and *area charts* are ideal for showing changes over time.
  Use-case: Show monthly revenue growth over a fiscal year.
- **Distribution:**
  Histograms, *box plots*, and *density plots* reveal spread, clusters, and outliers.
  Use-case: Display distribution of order quantities or profit margins.
- **Relationship:**
  Scatter plots are perfect for finding correlations between two measures.
  Use-case: Explore relationship between discount percentage and profit.

Note: Use Tableau's "Show Me" panel as a guide, but always align visual choice to the story you're trying to tell.

---

32. Steps to build a dual-axis combo chart.

A **dual-axis chart** combines two measures with different scales into one visual, typically sharing the same dimension.

Example: Sales and Profit over time.

Steps:

8. Drag Order Date to Columns.
9. Drag Sales to Rows.
10. Drag Profit to the same Rows shelf **next to Sales** (not below). Tableau creates a second axis.
11. Right-click the second axis → choose **"Dual Axis."**
12. Right-click either axis → **Synchronize Axis** (if necessary).
13. Use the **Marks card** for each axis to set different chart types (e.g., bar for Sales, line for Profit).
14. Adjust colours and tooltips for clarity.

Tip: Dual-axis charts work best when comparing related metrics like volume and profitability, but avoid overloading with too many axes to preserve readability.

---

33. Create a heat-map to show correlation intensity.

A heat map uses colour shading to represent value intensity, allowing users to spot patterns across combinations of dimensions.

Steps:

6. Drag a **dimension** to Rows (e.g., Region).
7. Drag another **dimension** to Columns (e.g., Category).
8. Drag a **measure** (e.g., Sales) to **Colour** on the Marks card.
9. Change the Mark type to **Square**.
10. Adjust the colour gradient scale (e.g., light to dark) to reflect magnitude.

Use-case: Visualise sales intensity by region and product category—darker squares signal higher values.

Tip: Use diverging colour palettes for positive vs. negative metrics like profit/loss.

---

34. Build a funnel chart for conversion analysis.

Funnel charts are used to visualise **drop-offs** across a series of stages—commonly in sales pipelines or marketing funnels.

Steps (Quick method):

8. Create a table with stages (e.g., Lead, Contacted, Demo, Closed) and counts.
9. Drag **Stage** to Rows.
10. Drag **Count** to Columns.
11. Sort Stage manually in descending order.
12. Change Mark type to **Bar**.
13. Right-click axis → **Reversed Axis** (optional).
14. Center-align bars using the **Size** shelf or a dual-axis trick.

Tip: Tableau doesn't offer a native funnel chart, but by formatting bar widths or layering dual axes, you can simulate the funnel effect effectively.

---

35. When do you choose a gantt chart?

Gantt charts visualise task schedules and durations—excellent for project timelines or resource tracking.

Use-case: Show product shipment durations or project task planning.

Steps:

6. Drag Task Name to Rows.
7. Drag Start Date to Columns.
8. Change Mark type to **Gantt Bar**.
9. Drag Duration to Size shelf (calculated as End Date - Start Date).
10. Optional: Add colour by project or priority.

Tip: Gantt charts can also be used to visualise **overlaps**, identify schedule delays, and ensure time-based transparency.

---

36. How does tableau recognise geographic roles and plot maps?

Tableau automatically assigns **geographic roles** to fields like Country, State, City, Zip Code, etc.

Steps to plot a map:

5. Drag a geographic field (e.g., State) to **Rows**.
6. Tableau recognises it as geographic and converts it to a **Map view** automatically.
7. Drag a measure (e.g., Sales) to **Colour** or **Size**.
8. Use **Show Me** to convert to filled map or symbol map.

Geographic roles include:

- Country/Region
- State/Province
- City
- Zip Code
- Latitude / Longitude

Tips:

- You can assign a geographic role manually: Right-click the field > **Geographic Role** > Choose type.
- Use **custom geocoding** for proprietary locations (e.g., warehouse zones).

---

37. Use the pages shelf to animate year-by-year change.

The **Pages shelf** allows for temporal or categorical animations, helping explore trends step-by-step.

Steps:

5. Drag Year(Order Date) to the **Pages shelf**.
6. Tableau adds play/pause controls.
7. Use this control to scroll through years, see visual transition.
8. Optional: Add Region or Segment to view category-wise changes over time.

Use-case: Animate how sales evolved across regions year-by-year.

Tip: Pages shelf is local to Tableau Desktop and doesn't publish to Tableau Server or Cloud (it gets converted to filters).

---

38. Create a histogram for sales distribution.

Histograms show **frequency distribution** of a single numeric variable.

Steps:

6. Drag a continuous field (e.g., Sales) to Columns.
7. Right-click and choose **"Create > Bins…"**
    - Choose bin size or let Tableau decide.
8. Drag the new bin field to Columns.
9. Drag Number of Records to Rows.
10. Adjust bar spacing using Size shelf.

Use-case: See how order amounts are distributed—whether most orders are low-value or skewed to the right.

Tip: Use tooltips and reference lines to annotate average or outliers.

---

39. Explain "measure names" and "measure values."

- **Measure Names**: A **built-in field** that holds the **names** of all measures in your dataset.
- **Measure Values**: Holds the **actual values** of the measures in a stacked structure.

Use-case:
When you want to display **multiple measures** (e.g., Sales, Profit, Quantity) in a single chart or table.

Steps to use:

4. Drag Measure Names to Columns or Filters.

5. Drag Measure Values to Rows.
6. Adjust Marks card to show desired visual (e.g., bar chart, text table).

Tip: Useful for creating side-by-side KPIs, comparative bar charts, or control panels.

---

40. Treemap vs bar chart—when is each superior?

- **Treemap:**
  - Visualises **part-to-whole** relationships with nested rectangles.
  - Encodes values using **area**, not axis.
  - Best when comparing large numbers of categories at once.
  - **Use-case:** Show sales by product sub-category within a region.
- **Bar Chart:**
  - Easier to compare **precise values** using a shared baseline.
  - Preferred when **exact comparison or ranking** is needed.
  - **Use-case:** Show top 10 customers by revenue.

Guidance:
Use **bar chart** when accuracy matters and **treemap** when space or categorical spread is the priority.

FILTERS, PARAMETERS & SETS

41. Normal filter vs context filter—key differences and performance impact.

In Tableau, filters are applied in a specific order known as the **order of operations**. A **normal filter** applies after data is queried and aggregated. A **context filter**, however, creates a subset of the data first, and then other filters are applied on top of this subset.

Normal Filter:

- Most filters (dimension or measure) are considered normal.
- They apply after extract, data source, and context filters.
- Default filtering behavior when you drag a field into Filters shelf.

Context Filter:

- Acts as a **pre-filter**—defines the data subset for other filters.
- Set by right-clicking a filter and choosing **"Add to Context."**
- Tableau creates a temporary table based on the context filter.

Performance benefit:
When a dashboard is slow due to large volumes, adding high-cardinality filters (like Region or Segment) as **context filters** reduces processing for other filters and calculations.

Use-case:
If you want to filter Top 10 customers *within* a specific region, make **Region** the context filter, so **Top N** applies only on that subset.

---

42. Create a parameter and show a dynamic reference-line.
Parameters are dynamic input controls that let users pass values into calculations or filters.

Steps to create a parameter:

1. Right-click in the Data pane → **Create Parameter**.
2. Name it (e.g., Sales Threshold).
3. Set the data type (Integer, Float, String, Date).
4. Provide a range or list of allowed values.
5. Right-click the parameter → **Show Parameter Control**.

Use in reference line:

1. Drag a measure (e.g., Sales) to Rows.
2. Right-click Y-axis → **Add Reference Line**.
3. Set Line Value to the created parameter.
4. Style the line and label (e.g., "Threshold Line").

Use-case:
Give users the ability to highlight dynamic benchmarks (e.g., sales goal of 50,000) which adjusts based on their input.

---

43. Parameter vs quick filter—choosing the right control.

Feature	Parameter	Quick Filter (Standard Filter)
Data source dependency	Independent of data source	Directly tied to data field
Use-case	Drive logic in calculated fields	Filter records directly
User input	Manual (slider, dropdown, input box)	Based on actual field values
Interactivity	Can change multiple sheets/calculations	Mostly filters a single sheet or context
Performance	Faster if well-used (e.g., no live DB hits)	Can be slower on large datasets

Example:
Use a parameter to allow users to switch between Sales and Profit as a KPI; use a quick filter to allow Region selection directly from the data.

---

44. Static set vs dynamic set vs combined set.

Sets are custom fields that define **IN/OUT** membership for a dimension.

- **Static Set:**
  - Created by selecting specific values manually.
  - Doesn't update when data changes.
  - *Example:* Manually select Top 5 Products by name.
- **Dynamic Set:**
  - Created using a condition (e.g., Top 10 by Sales).
  - Automatically updates as data changes.
  - *Example:* Show customers with sales > ₹1L.
- **Combined Set:**
  - Combines two sets using **AND / OR / EXCEPT** logic.
  - *Example:* Customers who bought both A and B.

Usage:
Sets enable custom groupings, set-based filters, and powerful comparisons using IN/OUT logic on the Marks card or calculations.

---

45. Explain filter, highlight and url dashboard actions.

Dashboard actions let users **interact with multiple views**.

- **Filter Action:**
  - Filters the target worksheet based on selection in source.
  - *Use-case:* Clicking a region in map filters sales bar chart.
- **Highlight Action:**
  - Emphasises marks in other views matching selection.
  - Doesn't remove data—just dims non-matching values.
  - *Use-case:* Hover over a product to highlight customer details.
- **URL Action:**
  - Opens a web page based on field value.
  - *Use-case:* Clicking an Order ID opens order details on company portal.

Configuration:
Go to Dashboard → **Actions** → Add Action → Choose type and fields to pass.

---

46. Configure a filter action to drive a detail sheet.

Filter actions are widely used to enable interactivity between views.

Steps:

1. Create two worksheets:
   - Summary (e.g., Sales by Region).
   - Detail (e.g., Sales by Customer).
2. Combine them in a dashboard.
3. Go to Dashboard → **Actions** → Add **Filter Action**.
4. Choose source sheet, target sheet, and fields to pass.
5. Set behavior (hover, select, menu) and what happens on deselect.

Use-case:
Clicking on "South" region filters the customer-level detail table to show only customers in South.

---

47. Implement row-level security with user functions.

Row-level security (RLS) restricts what data users can see, based on their identity or group.

Method 1: USERNAME() or ISMEMBEROF() in a calculation

IF USERNAME() = 'ram@abc.com' THEN [Region] = 'South'

ELSEIF ISMEMBEROF('NorthGroup') THEN [Region] = 'North'

Method 2: Create a mapping table

1. Create a table: User ID | Region
2. Join or blend this with your main dataset.
3. Filter views based on this mapping.

Publish with permissions and use Viewer Role in Tableau Cloud/Server.

---

48. Add show/hide "apply" buttons for multiple filters.

When a dashboard has multiple filters, users may prefer to **apply all at once** rather than auto-updating on every selection.

Steps:

1. Add all filters to the dashboard.
2. Right-click on each filter → **Customize** → **Show Apply Button**.
3. User can now make multiple selections, then click **Apply** once to trigger refresh.

Use-case:
Reduces lag and improves experience when working with large data sources or slow dashboards.

---

49. What is a parameter action and one interactive use-case?

Parameter actions allow users to **update parameter values** based on interaction with views (e.g., click, hover, select).

Use-case Example:
Let user **click on a metric name** to update KPI card.

Steps:

1. Create a String Parameter: Metric Choice with values: "Sales", "Profit".
2. Create calculated field:
3. CASE [Metric Choice]
4. WHEN "Sales" THEN [Sales]
5. WHEN "Profit" THEN [Profit]
6. END

7. Add metric buttons in a sheet using Metric Choice.
8. Go to Dashboard → Actions → **Add Parameter Action**.
9. Set source sheet, target parameter, and field.

Now, clicking "Profit" dynamically updates charts using the selected metric.

---

50. Describe tableau's order of operations (query pipeline).

Tableau processes filters and calculations in a defined **order**, which affects results, especially with LODs and context filters.

Key stages (simplified):

1. **Extract filters** (if data source is extract)
2. **Data source filters**
3. **Context filters**
4. **Top N / Conditional filters**
5. **Dimension filters**
6. **Measure filters**
7. **Table calculations**
8. **Rendering filters (like on Pages or animation)**

Why it matters:

- A **measure filter** can only apply **after** aggregation.
- To make a Top 10 filter work inside a region, **make Region a context filter** first.
- LOD expressions are affected by **which filters come before/after** them.

Understanding this sequence avoids logical errors in calculated fields and dashboard interactivity.

DASHBOARDS & INTERACTIVITY

51. Steps to assemble a dashboard from existing worksheets.

A Tableau dashboard is a canvas where you can combine multiple visualisations, text, images, filters, and interactive elements to create a cohesive analytical story.

Steps to build a dashboard:

1. Click the **"New Dashboard"** tab at the bottom of the workbook.
2. In the left panel, you'll see all available **Worksheets**. Drag the desired sheets onto the canvas.
3. Use the **"Tiled" or "Floating"** layout option to position elements precisely.
4. Adjust sizes by dragging borders, or use **container objects** (horizontal/vertical) for structured layouts.
5. Add **legends, filters, or parameters** by dragging them from the left pane or choosing "Use as Filter" from a worksheet.
6. Use the **"Dashboard"** menu > **"Actions"** to create interactivity (e.g., Filter, Highlight, URL actions).
7. Use **Device Preview** to test how your dashboard looks on different screen sizes.
8. Add titles, images, or instructions using **Text** objects.

Tip: Keep the design clean and logical. Place summary views on top and detailed breakdowns below or in adjacent sheets.

---

52. What is a story and how can it complement a dashboard?

A **Story** in Tableau is a **sequence of worksheets or dashboards** arranged in a linear narrative format to guide the audience through insights step-by-step.

Structure of a Story:

- Made up of **story points**, each referencing a worksheet or dashboard.
- Each point can have **a title, annotation, and description**.
- Navigation buttons let users move forward/backward like a slideshow.

When to use:

- To present **before-after analyses** (e.g., pre-campaign vs post-campaign).
- To tell a **data-driven business case** step-by-step.
- To create a **progressive reveal** for management or clients.

Tip: Use a story when you want a fixed sequence; use a dashboard when you want interactive exploration.

---

53. Three visual-design best practices you follow.

1. **Keep it clean and minimalistic:**
    - Avoid clutter. Use whitespace to separate visual elements.
    - Limit colours to 2–3 consistent tones. Reserve bold colours for emphasis.
2. **Ensure readability and hierarchy:**
    - Use clear labels, consistent font sizes, and proper alignment.
    - Order KPIs and charts based on user priority (top-left = highest attention).
3. **Use interactivity intentionally:**
    - Filter actions, highlights, and tooltips should enhance—not overwhelm—the user.
    - Add "Reset" buttons or clear indicators when filters are applied.

Pro Tip: Always test with a few end-users before publishing. What's obvious to the creator may not be obvious to a stakeholder.

---

54. Process to publish to tableau cloud/server with version comments.

Publishing your dashboard to Tableau Cloud or Server allows others to access it securely via browser.

Steps to publish:

1. Click **File > Save to Tableau Cloud/Server**.
2. Sign in to your account.
3. Select the **project folder** (e.g., "Marketing Dashboards").
4. Name the workbook appropriately.
5. Add **"Description" or version comments** for clarity.
6. Choose whether to publish **data source separately** or embed it.
7. Set **permissions** for viewers/editors.
8. Click **Publish**.

Tips:

- Use **incremental versioning** (e.g., v1.0, v1.1) in workbook names or comments.
- Schedule extract refresh if required.

---

55. Schedule an extract refresh on tableau cloud.

Scheduling ensures your dashboard stays up-to-date without manual effort.

Steps:

1. Go to Tableau Cloud > **Explore > Data Sources**.
2. Click on the relevant published data source.
3. Choose **"Schedules"** tab > **Create Schedule**.
4. Select **Frequency** (daily, hourly, etc.), **Time Zone**, and **Target Time**.
5. Click **Apply** and ensure the data source has the correct credentials for refresh.

Tip: Use incremental refresh for large datasets to save time and reduce load.

---

56. Customise tooltips for richer insight.

Tooltips appear when a user hovers over a data point and can offer contextual details.

Steps to customise:

1. Click the **Tooltip** shelf in the Marks card.
2. Edit content with text, fields, or calculations. Use formatting tools (bold, colour).
3. Add KPIs or derived metrics like Rank, Percent of Total, or Change from Previous.
4. Use **conditional formatting**:
5. IF SUM([Sales]) > 50000 THEN "High Performer" ELSE "Below Target"
6. Preview by hovering over marks.

Tip: Keep tooltips short and actionable—avoid dumping raw data.

---

57. Build a viz-in-tooltip mini-chart.

Viz in Tooltip lets you embed a secondary visual inside a tooltip that updates dynamically based on the hovered mark.

Steps:

1. Create a supporting worksheet (e.g., sales trend for each customer).
2. In your main worksheet, open **Tooltip editor**.
3. Click **Insert > Sheets > [Supporting Sheet Name]**.
4. Set sheet filters to **"All Using This Field"** to match the hover selection.
5. Adjust width, height, and formatting as needed.

Use-case:
Hovering over a customer name shows their last 6-month trend without leaving the main dashboard.

---

58. Embed external web content in a dashboard.

You can embed live web content (maps, videos, documents) using a **Web Page object**.

Steps:

1. In Dashboard view, drag **Web Page** from the Objects pane.
2. Enter a fixed URL (e.g., Google Maps, company portal).
3. Or, use a **URL action** to pass a dynamic URL (e.g., link to an Order ID).

Use-case:
Embed a YouTube explainer, a live Google Doc, or a web-based report for contextual references.

Tip: Ensure the content is HTTPS-enabled, or Tableau Cloud may block it.

---

59. Design for mobile layouts using device designer.

Tableau allows you to optimise dashboards for different screens: desktop, tablet, and phone.

Steps:

1. In Dashboard view, open **Device Preview** (top-left toolbar).
2. Choose **Add Device Layout** > Select screen type (Phone, Tablet).
3. Rearrange or resize charts for touch-friendly and scrollable layout.
4. Use vertical containers and larger fonts/buttons for mobile usability.

Best Practices:

- Stack visuals vertically on mobile.
- Hide less critical charts using the **Layout > Show/Hide** option.
- Test scrolling, filter size, and responsiveness on actual mobile screens.

---

60. Toggle between table and chart views with a sheet-selector.

Sometimes users want to switch between a tabular view and a graphical summary.

Steps:

1. Create a **parameter** called View Type with values: “Table”, “Chart”.
2. Create a **calculated field**:

3. [View Type] = "Table"
4. Create two sheets—one with a table and one with a chart.
5. On the dashboard, drag both sheets into a **Vertical container**.
6. Use the **Layout > Control Visibility** option (or parameter filter) to **show/hide** based on parameter selection.

Alternative:
Use **navigation buttons** or **custom shapes** as toggles that update parameter actions.

Benefit:
Users can choose how they consume insights—visually or in detailed format—without overloading the dashboard.

PERFORMANCE OPTIMISATION

61. Top techniques to optimise slow tableau dashboards.

A slow dashboard can hurt user experience and reduce adoption. Tableau performance tuning involves multiple strategies:

- **Use Extracts over Live Connections:** Extracts (.hyper files) reduce database load and improve response time.
- **Reduce Mark Count:** Large mark counts (>10,000) slow rendering. Aggregate data before visualising.
- **Use Context Filters Strategically:** Place high-cardinality filters as context filters to reduce computation for dependent filters.
- **Limit Quick Filters:** Avoid adding too many filters. Replace multi-select filters with parameters if possible.
- **Hide Unused Fields:** Remove unused fields from extracts to reduce workbook size and load time.
- **Avoid Nested Calculations in Views:** Simplify calculated fields and pre-compute values when possible.
- **Minimise Use of LODs and Table Calcs:** Use sparingly on large datasets or try pre-aggregated fields instead.

- **Optimise Joins:** Avoid unnecessary joins and prefer relationships for complex data models.
- **Reduce Sheets in a Dashboard:** Every additional sheet increases query load. Combine where possible.
- **Enable Workbook Performance Recorder** to track bottlenecks (discussed next).

---

62. Handling large datasets while maintaining responsiveness.

Tableau is built to handle large data, but it requires thoughtful design:

- **Use Aggregation:** Summarise data to the lowest useful grain before building visuals.
- **Incremental Extracts:** Refresh only new rows using a unique date/timestamp column.
- **Partitioned Extracts:** Split large extracts by Region, Year, etc., for faster loading.
- **Custom SQL Limits:** Apply WHERE clauses in Custom SQL to pre-filter data.
- **Avoid Cartesian Joins:** Ensure joins are properly keyed to avoid data explosion.
- **Paginate Reports:** Use filters and parameters to show data for 1 region, month, or category at a time.
- **Leverage Filters on Context:** Filter early to reduce processing.
- **Minimise Marks & Sheets:** Simplify visuals—use summary KPIs instead of massive data tables.

Tip: Always test extract file size and open time before publishing.

---

63. COMMON REASONS FOR SLOW DASHBOARDS.

Several issues can cause dashboards to lag:

- **Excessive Data Volumes:** Large mark count or wide tables slow rendering.
- **Complex Calculations:** Nested IFs, LODs, or multiple LOOKUP() calls.
- **Too Many Quick Filters:** Especially on high-cardinality fields (e.g., Customer Name).
- **Unnecessary Joins:** Particularly many-to-many or full outer joins.
- **Unoptimised Extracts:** Extracts with unnecessary columns or rows.
- **Rendering Problems:** Too many sheets in one dashboard or floating objects causing layout recalculations.
- **Heavy Tooltips or Viz-in-Tooltips:** If used on every mark, these can degrade performance.

- **Auto-updating Filters:** Filters set to auto-update with every interaction can delay dashboard readiness.

Tip: Prioritise filter order and simplify visual layout to improve speed.

---

64. EXTRACT VS LIVE CONNECTION—PERFORMANCE DIFFERENCE.

Feature	Extract	Live
Speed	Faster (queries run on local .hyper file)	Depends on network and source DB
Offline Access	Yes	No
Refresh	Manual or scheduled	Always real-time
Size	Increases workbook size	Lightweight
Data Freshness	Stale until next refresh	Always current

When to use:

- Use **extracts** when performance is critical, and real-time data isn't mandatory.
- Use **live** when dashboards are monitoring operational KPIs and must reflect the latest data.

---

65. Using the performance recorder to identify bottlenecks.

The **Performance Recorder** shows how Tableau is processing each action—loading sheets, running queries, rendering, etc.

Steps to enable:

1. From the top menu: **Help > Settings and Performance > Start Performance Recording**.
2. Interact with your workbook (navigate, apply filters, load dashboards).
3. Stop recording: **Help > Settings and Performance > Stop Performance Recording**.
4. Tableau opens a new workbook showing:
    - Time spent on each action
    - Query load duration
    - Computation and rendering times
    - Dashboard object load order

Use-cases:

- Identify which sheet/query is slowing things down.
- Spot unresponsive filters or over-complicated joins.
- Decide which calculations or fields to optimise.

---

66. Real example: dashboard optimised from 14 to 4 seconds.

Scenario: A customer sales dashboard for India's five zones was taking 14 seconds to load.

Problems identified:

- Data was 3 years of transactional details (~2M rows).
- 10 quick filters including Customer Name and Product ID.
- Multiple nested calculations (e.g., tiered logic for rating customers).
- Dashboard had 6 sheets, many using table calcs.

Fixes applied:

- Replaced **Customer Name filter** with a **parameter selector**.
- Aggregated sales data to monthly level in **Tableau Prep** before loading.
- Converted live query to **incremental extract**.
- Removed unused fields and reduced sheet count from 6 to 3.

Result: Load time dropped to **4 seconds** without loss of business value.

---

67. How context filters can improve performance.

A **context filter** creates a temporary dataset (in-memory sub-cube), reducing the amount of data evaluated by subsequent filters.

When to use:

- When using a **Top N filter** that depends on another dimension.
- When applying **dependent filters** on a large data set.
- When combining **expensive calculations** that depend on a smaller subset.

Tip: Avoid overusing context filters. They increase memory usage and require recomputation every time the context changes.

---

68. Blending multiple sources—performance considerations.

Blending requires Tableau to run **separate queries** on each source and then **merge results client-side**, which can slow dashboards.

Recommendations:

- Use **relationships or joins** when possible (within same data source).
- Ensure **blending keys** (e.g., Customer ID) are consistent in format and granularity.
- Limit fields from the **secondary source** to reduce processing.
- Use **aggregated blending** instead of row-level blending when possible.
- Avoid filters on the secondary source unless needed.

Use-case: Blend Google Analytics data with CRM data to track campaign performance across marketing and sales.

---

69. Effect of complex calculations or high marks on speed.

Complex calculations like nested IFs, WINDOW_SUM(), and LODs require more processing time—especially when used across large datasets.

High number of marks (>10k):

- Slows rendering in browser or Tableau Desktop.
- Causes tooltips, hover actions, and filters to lag.

Recommendations:

- **Pre-compute values** using Tableau Prep or in the database.
- Use **summary-level dashboards** for high-volume data.
- Avoid row-level visualisations unless they're truly necessary.
- Replace **SHOW ALL TABLES** with **aggregated summaries** plus drill-through options.

---

70. Managing filter count for efficient dashboarding.

Having too many filters (especially on high-cardinality fields) affects dashboard performance and user experience.

Tips:

- Replace filters with **parameters** for static control.
- Use **cascading filters**—only load options based on previous filter (e.g., Segment → Customer Name).
- **Hide filter controls** not used frequently.
- Use **global filters** instead of per-sheet filters if applicable.
- Group or consolidate categories where possible (e.g., combine states into regions).

Pro Tip: Keep filter controls limited to 3–5 most used fields; move advanced filters to a collapsible panel.

ADVANCED TECHNICAL CONCEPTS IN TABLEAU

81. What is Tableau Prep and when should you use it?
Tableau Prep is a self-service data preparation tool designed to clean, reshape, and combine raw datasets before loading them into Tableau Desktop.

Use Tableau Prep when you need to:

- Clean messy or unstructured data
- Merge multiple data sources
- Reshape or pivot data (wide ↔ long format)
- Remove duplicates, nulls, or errors
- Create row-level calculations before analysis

Key benefits:

- Visual, drag-and-drop workflow
- Automatic profile pane shows data quality
- Reusable, refreshable flows
- Easy export to Tableau Desktop or .hyper extract

Example: If you receive 12 Excel files from regional teams every month with inconsistent formats, you can use Tableau Prep to union them, clean field names, and combine into one clean dataset.

---

82. How is Tableau Prep different from Desktop?

Feature	Tableau Prep	Tableau Desktop
Primary purpose	Data preparation (cleaning, shaping, combining)	Data visualisation and dashboarding
Output	Clean dataset or .hyper extract	Visual reports, dashboards

Feature	Tableau Prep	Tableau Desktop
Interface	Flow-based, node-style visual interface	Worksheet and dashboard views
Advanced analytics	No charting or advanced visualisation	Full visualisation and interactivity support

In short: Use Prep before analysis to fix the data, then use Desktop to visualise it.

---

83. Common tasks in Tableau Prep.
Some of the most common data preparation tasks performed in Tableau Prep include:

- Joining and unioning multiple tables from different sources
- Pivoting columns to rows (or vice versa) for reshaping
- Cleaning fields (remove nulls, rename headers, standardise formatting)
- Filtering rows before visualisation
- Creating calculated fields (e.g., category labels, percent growth)
- Aggregating values before export to reduce size
- Grouping similar values using fuzzy matching (e.g., Mumbai vs. Mmbai)

---

84. Share a scenario where Tableau Prep improved your workflow.
Scenario:
In a retail project, the client shared daily sales data split across 12 monthly Excel sheets. Each file had different column names and extra rows.

Problem:

- Manual merging took hours
- Column mismatches caused join failures
- Frequent errors in aggregation

Solution using Tableau Prep:

- Used wildcard union to merge all monthly files
- Standardised headers using a Clean step
- Created a calculated field for Quarter
- Output the cleaned dataset as .hyper for Tableau Desktop

Result:
Reduced manual effort from 6 hours to 30 minutes per month, and the dataset was now refreshable via Tableau Server.

---

85. How do you output and use Prep data in Tableau Desktop?
Once your data is cleaned in Tableau Prep, you can export and load it in Tableau Desktop for analysis.

Steps:

1. Add Output step in Prep → Choose .hyper, .tde, or .csv as format
2. Save the output locally or publish directly to Tableau Server/Cloud
3. In Tableau Desktop:
    - Click "Connect to Data"
    - Select "Hyper" file
    - Load it like any other dataset

Tip: If published to Server/Cloud, you can also schedule flow refreshes to keep the data up to date.

---

86. What are level of detail (LOD) expressions and why are they important?
Level of Detail (LOD) expressions are advanced calculations that allow you to compute values at the data granularity you define, independent of the view's dimensions.

There are three types of LOD expressions:

- FIXED: Aggregates data at a fixed level regardless of view filters
- INCLUDE: Adds a dimension to the aggregation context
- EXCLUDE: Removes a dimension from the aggregation context

They are useful for calculating things like customer lifetime value, averages by sub-category, or sales targets unaffected by user filters.

---

87. How does Tableau handle NULL values in visualisations?
Tableau displays NULL values as a special category or omits them depending on context.

- For dimensions: NULL appears as a separate category
- For measures: NULL is usually ignored unless handled

Techniques to manage NULLs:

- Use IFNULL() or ZN() to replace with defaults
- Filter them out explicitly

- Use calculated fields to treat them conditionally

---

88. What is the difference between data blending and relationships?
Data blending links data from multiple sources on a common field but runs separate queries and combines results client-side. It is used when joining across different databases.

Relationships, introduced in Tableau 2020.2, connect tables logically, deferring joins until query time and avoiding data duplication. They offer better performance and flexibility for modern data models.

---

89. How can you use parameters to create dynamic dashboards?
Parameters are user-controlled inputs that can be used in calculations, filters, reference lines, and actions.

Common uses include:

- Switching between metrics (e.g., Sales vs. Profit)
- Toggling between views (e.g., Table vs. Chart)
- Controlling what-if analysis inputs (e.g., discount rate)

You pair a parameter with a calculated field to influence dashboard elements dynamically.

---

90. What are table calculations and how are they different from regular calculations?
Table calculations are computed after data is aggregated and loaded into the view. They are dependent on the layout and can be used for:

- Running total
- Rank
- Moving average
- Percent of total

Unlike row-level or aggregate calculations, table calculations are scoped by the worksheet structure and require defining partitioning and addressing.

---

91. Explain the purpose of the INDEX() and RANK() functions.

- INDEX(): Returns the position of a row in the partition (e.g., 1st, 2nd, etc.)
- RANK(): Assigns a rank to values based on the specified measure. It can be configured as RANK_DENSE, RANK_UNIQUE, etc.

Use-cases include:

- Sorting visuals dynamically
- Creating top-N filters
- Displaying row numbers in tables

92. What are the best practices for joining large datasets in Tableau?

- Use relationships instead of physical joins when possible
- Avoid many-to-many joins to prevent data duplication
- Join on indexed keys
- Limit the number of joined tables
- Use filters or custom SQL to reduce row volume
- Pre-aggregate data in the source if possible

---

93. How can you implement row-level security (RLS) in Tableau?
Row-level security restricts data access based on user identity.

Methods:

1. Use USERNAME() or ISMEMBEROF() logic in calculated filters
2. Create a user-to-region mapping table and join with main data
3. Use Tableau Server groups and apply data source filters accordingly

Always validate using Tableau Server's 'View As' feature.

---

94. How does Tableau optimise queries for performance?

- Uses query culling to avoid fetching unused fields
- Pushes filters to the database (predicate pushdown)
- Uses extract engine (.hyper) for faster query resolution
- Utilises join culling when relationships are defined correctly
- Caches frequently used queries in memory

Use the Performance Recorder to monitor how queries execute and spot bottlenecks.

---

95. What is the use of the ATTR() function?
ATTR() returns a single value if all rows have the same value for a field; otherwise, it returns an asterisk (*).

Used when:

- You want to show a dimension in a tooltip or title safely
- Avoiding aggregation errors in mixed-level visualisations
- Preserving context without grouping explicitly

---

96. How can you enable drill-down in Tableau charts?
Drill-down allows users to click on a higher-level category (like Region) and see lower levels (like State or City).

Steps:

- Create a hierarchy by dragging dimensions onto each other
- Use this hierarchy in a bar or map visual
- Click "+" on the chart to drill into the next level

This interactivity improves user exploration and insight discovery.

---

97. How does Tableau manage data extracts and refreshes?
Data extracts are snapshots of data stored in .hyper format.

Management options:

- Refresh extracts manually or schedule them
- Use incremental refreshes to update only new rows
- Publish extracts to Tableau Server and define refresh schedules

Incremental refresh requires a column like date or unique ID to track new records.

---

98. What is a scaffold dataset and when would you use it?
A scaffold dataset contains a complete set of values (like a date calendar) to ensure visuals don't skip missing data points.

Use-cases include:

- Showing zero sales days in a time series
- Filling in missing combinations (e.g., every product in every region)
- Avoiding gaps in bar or line charts

Scaffolding improves analytical accuracy.

---

99. How do you show top N values and group the rest into 'Other'?
Steps:

1. Create a set for Top N items based on a measure
2. Use a calculated field:
   IF [Top N Set] THEN [Category] ELSE 'Other'
3. Use this new field in your view

This improves readability by consolidating long tails into an 'Other' category.

---

100. How can you use custom SQL in Tableau and what are the risks?
Custom SQL can be written during data connection to pre-define joins, filters, or derived tables.

Pros:

- Pre-process data before Tableau reads it
- Handle complex logic not possible in join UI

Cons:

- Increases load time
- May bypass Tableau's query optimisations
- Harder to maintain for non-SQL users

Use custom SQL only when necessary and always document the query.