SECE VAC - Mastering Visual Intelligence

Day - 1 - Tableau — Architecture & Interface

What Tableau Is ? (High Level)

Tableau is a visual analytics platform that enables users to:

- Connect to various data sources
- Prepare and transform data
- Build interactive visualizations
- Share insights (via Tableau Public, Server, or Online)

Core Components: Tableau Desktop/Public, Tableau Server/Online, Data Engine (Hyper), and specialized file types that define and package work.

Core Components

1) Tableau Desktop / Tableau Public

Theory:

- **Desktop:** Authoring environment for building workbooks and dashboards.
- **Public:** Free version with connection limits and public publishing.

Real Example:
A retail analyst uses Desktop to create a dashboard showing monthly sales, top products, and store performance. If using Public, the workbook becomes publicly viewable.

---

2) Tableau Server / Tableau Online

Theory:

- Platforms for sharing, scheduling data refreshes, controlling user access, and hosting dashboards.
- **Server:** Self-hosted.

- **Online:** Tableau-hosted SaaS.

Real Example:
A marketing team publishes a campaign dashboard on Tableau Server so executives can view daily campaign KPIs; scheduled extracts refresh each morning.

---

3) Tableau Data Engine (Hyper)

Theory:
Hyper is Tableau's in-memory columnar analytic engine for fast querying—used for extracts (`.hyper`). It optimizes aggregations and large query performance.

Real Example:
When analyzing 100M transaction rows, creating a Hyper extract dramatically speeds up interactive filtering and visualization compared to live queries on a slow transactional database.

---

4) Key File Types

Theory:

- `.twb` : Workbook (XML) storing visualization metadata (excludes data).
- `.twbx` : Packaged workbook (`.twb` + local data, zipped). Ideal for sharing when recipients lack data source access.
- `.hyper` / `.tde` : Extract files (Hyper is current; TDE is legacy).
- `.tds` / `.tdsx` : Data source files (`.tds` stores connection + metadata; `.tdsx` is packaged with data).

Real Example:
Sending a colleague a `.twbx` file containing the workbook and sample data for offline viewing.

Tableau Interface

Main Areas and Functions

- **Data Pane:** Fields from connected data sources (dimensions & measures).
- **Shelves (Rows/Columns):** Drop fields here to build views.
- **Marks Card:** Controls color, size, label, tooltip, and detail encoding.
- **Filters/Pages/Tooltip Controls:** Add interactivity and animation.
- **Show Me:** Suggests chart types based on selected fields.

- **Analytics Pane:** Quick analytics objects (reference lines, trend lines, forecasts).

Example Workflow:
Drag `Order Date` to Columns, `Sales` to Rows, and `Region` to Color → Generates a line chart showing sales by region.

Connecting & Managing Data Sources

Metadata Editing

Theory:
Editing field names, data types, default aggregations, calculated fields, aliases, hierarchies, and descriptions in Tableau's Data Source tab or data pane. Metadata changes are stored in the data source (`.tds` / `.tdsx`) or workbook.

Real Example:
A Sales CSV contains `ord_dt` —renamed to `Order Date` , type changed to Date, default format set, and a hierarchy created (Year → Quarter → Month).

Extracts (Hyper)

Theory:
A snapshot of data stored locally in Tableau's Hyper format. Extracts speed up analysis, enable offline access, and reduce load on source systems. They can be full or incremental.

Real Example:
Creating an extract of a cloud database to run interactive dashboards daily without impacting the production DB; scheduling nightly refreshes on Server.

Joins

Theory:
Combines two or more tables from the same connection on matching keys (inner, left, right, full). Joins occur at the data source layer in Tableau's logical/physical model (Tableau 2020.2+ supports logical layer relationships before physical joins).

Real Example:
Joining `Orders` and `Customers` on `CustomerID` to incorporate customer attributes into the sales table.

Blends

Theory:
Combines data from different data sources at the worksheet level. One data source is primary; others are secondary. Blending happens after aggregation and is used when sources cannot be joined directly (e.g., different systems).

Real Example:
Blending Sales (in Redshift) with Quota (in Excel) to compare actuals vs. quota. Tableau matches on shared fields (e.g., `Region`) and aggregates each source independently before blending.

Custom SQL

Theory:
When connecting to a database, users can write a custom SQL query to shape data (joins, calculations, filtering) before Tableau ingests it. Tableau treats the SQL as a virtual table.

Real Example:
Using a custom SQL query to pre-aggregate monthly revenue per store in the database, reducing data transfer to Tableau and improving performance.

Tableau Web Authoring: When and How to Use It

When to Use Web Authoring

Use Case	Why Web Authoring Works
Learning Tableau	Free, no installation needed
Building public dashboards	One-click publish & share
Connecting to Google Sheets	Real-time updates from Sheets
Simple joins / calculated fields	Fully supported in web version
Personal or portfolio projects	Ideal, since Public = open gallery

---

How to Start Using Web Authoring (Step-by-Step)

Step 1 – Go to Tableau Public

Visit https://public.tableau.com/app/discover

Step 2 – Sign In / Create Account

- Click **Sign In** at the top right (use Google or email)
- After login, you'll be taken to your Tableau Public profile

Step 3 – Start a New Project

Two options:

- **Create → Web Authoring**
- **Upload a file** (e.g., Sales.csv) – automatically opens Web Authoring

Step 4 – Connect to Data

Options include:

- **Upload from computer** → choose CSV or Excel file
- **Connect to Google Sheets** → sign in and pick spreadsheet

Tableau shows data preview and automatically detects data types

Step 5 – Build a Visualization

Web authoring canvas components:

- **Data Pane (left):** Fields (dimensions & measures)
- **Shelves (top):** Columns, Rows, Filters
- **Marks card (middle):** Color, Size, Label, Tooltip
- **Analytics tab:** Trend lines, reference lines, etc.

Example Workflow:

1. Drag `OrderDate` → Columns
2. Drag `Sales` → Rows
3. Drag `Region` → Color
4. Click "Show Me" → Choose "Line Chart"

You've built your first visualization!

Step 6 – Create a Dashboard

- Click **New Dashboard** at bottom tab bar
- Drag existing sheets (charts) onto dashboard canvas
- **Add filters:** Click sheet → dropdown → "Show Filter"
- Adjust layout using **Layout panel** (right side)
- Add text boxes/titles using toolbar

Step 7 – Publish Your Work

- Click **File → Save to Tableau Public**
- Name your visualization (e.g., "Regional Sales Overview")
- Saves to your Tableau Public profile:
  `https://public.tableau.com/app/profile/<yourusername>`
- **Share your dashboard link with anyone!**

Part 1 – Tableau Architecture & Interface (Theory + Demo Examples)

Core Components Comparison

Component	What It Does	Online Equivalent	Example
Tableau Desktop	Main authoring tool for building dashboards	Web Authoring in Tableau Public	Create charts & dashboards online
Tableau Server / Online	Sharing platform with permissions management	Tableau Public (free, public only)	Publish dashboard online
Data Engine (Hyper)	Fast in-memory engine for extracts	Not in Public Web (Desktop/Server only)	Live Google Sheet used instead
File Types	How Tableau saves workbooks and data	.twb, .twbx, .tds, .hyper, .tdsx	Hidden in web – autosaves online

How Components Work Together

1. **In Desktop** → Connect to data → Build dashboards → Save as .twb or .twbx
2. **Publish** to Server or Tableau Public
3. **Hyper Data Engine** processes data quickly

4. **File types** share and store data and design

Tableau Public Web Workflow

- Connect to Google Sheets/Excel in browser
- Build visualizations
- Click "Save to Tableau Public" → Publishes to cloud (like Server)

Summary Table

Concept	You Learned	Online Equivalent
Tableau Desktop	Local authoring	Web Authoring
Tableau Server	Sharing platform	Tableau Public
Data Engine (Hyper)	Fast extract	Live Google Sheet
File Types	.twb, .twbx, etc.	Saved in Public Cloud
Metadata Editing	Rename, calc fields	Works in Web
Extracts	Offline snapshot	Live data from Sheets
Joins	Combine tables	Works online
Blends	Combine data sources	Simulate via dashboards
Custom SQL	Query databases	Simulate via Google Sheet exports

Tableau Public Web: Relationships Model (Hands on Example)

What Changed: "Joins" → "Relationships"

Old Tableau (Desktop Classic): You had to physically join tables (like in SQL).

New Tableau (Web & Desktop Modern): You relate tables instead. Tableau decides how to join them dynamically when you use fields from both.

Think of Relationships as "smart joins" — Tableau only joins the data when you use it in a chart.

Data Model We'll Build

You have one Google Sheet with 3 tabs: Sales, Customers, Quotas

We'll build relationships like this:

- Sales –(CustomerID)– Customers
- Sales –(Region)– Quotas

Then, you'll use all 3 together in dashboards.

Step-by-Step in Tableau Public (Web)

Step 1 – Open Tableau Web & Connect to Google Sheets

1. Go to https://public.tableau.com/app/discover
2. Click Sign in (top right). If you don't have an account → click Create one (free)
3. Once signed in, click Create → Web Authoring
4. Under Connect to Data, click Google Sheets
5. Log in with your Google account
6. Select your file: Tableau_Online_Practice

Tableau opens the Data Source Page – this is where we define relationships.

Step 2 – Add Tables (Tabs) as Related Tables

You should now see an empty canvas with a section titled "Drag tables here"

1. From the left panel, drag Sales into the canvas – this becomes your primary table
2. Now drag Customers into the same canvas (drop it near "Sales")
3. Tableau will automatically open a "Create Relationship" dialog

Step 3 – Define Relationship Between Sales and Customers

In the Create Relationship dialog:

- Under Sales → choose CustomerID
- Under Customers → choose CustomerID
- Click Close

You'll see a relationship line connecting them – that's the relationship line (instead of a join).

This means Tableau will use CustomerID to connect those two tables whenever both are used in a chart.

Step 4 – Add Relationship to Quotas

1. Drag Quotas into the canvas (drop it near Sales)

2. Tableau opens another Create Relationship box
3. Match these: Sales.Region → Quotas.Region
4. Click Close

Now you'll see three tables connected by two relationship lines.

You've just built your data model: Customers ←(CustomerID)→ Sales ←(Region)→ Quotas

Step 5 — Go to the Worksheet to Build Visuals

Click Sheet 1 (bottom-left corner of the screen)

Tableau switches from Data Source view → Worksheet view

On the left, you'll see three table names (Sales, Customers, Quotas) in the data pane — each can be expanded to show their fields.

Step 6 — Metadata Editing (in Web)

Let's fix field names and data types:

1. Expand Sales
2. Right-click (or click the three dots ...) next to Sales → Rename → type Revenue
3. Make sure OrderDate has a calendar icon (if not, click → change Data Type → Date)
4. Expand Customers
5. Rename fields if needed (e.g., CustomerName → Customer Name)
6. Expand Quotas
7. Rename MonthlyQuota → Quota

That's metadata editing — renaming, data type fixing, and organizing your fields.

Step 7 — Create Calculated Field (Profit)

1. In the Data Pane, click the small down arrow beside "Sales" or right-click inside the pane
2. Choose Create Calculated Field
3. Name: Profit
4. Formula: `[Revenue] * 0.3`
5. Click Save

You've now added a new measure to your Sales table.

Step 8 — Create Your First Chart (Revenue by Region & Segment)

Make sure you're in Sheet 1:

1. From Sales, drag Region → Columns
2. Drag Revenue → Rows
3. Drag Segment (from Customers) → Color (under "Marks" card)

Tableau automatically connects Sales + Customers using your relationship on CustomerID.

You'll now see a bar chart showing Revenue per Region, colored by Segment.

Tableau pulled data from both Sales (Revenue) and Customers (Segment) — automatically joined on the relationship!

Step 9 — Create a Second Sheet (Quota vs Revenue)

1. Click the + icon at the bottom → choose New Worksheet
2. Drag Region → Columns
3. Drag Quota (from Quotas table) → Rows
4. Drag Revenue (from Sales) → Rows
5. You'll get two measures (one blue, one orange)
6. Click the Marks dropdown → choose Bar
7. Drag Measure Names → Color (so each bar gets its own color)

Now you're comparing Quota vs Revenue per Region.

Step 10 — Combine Charts in a Dashboard

1. Click New Dashboard (bottom tab)
2. On the left, under "Sheets", drag your two sheets into the dashboard
3. Put the first chart (Revenue by Region) on top
4. Put Quota vs Revenue chart below
5. Add a title: On the left, under "Objects", drag Text to the top → type: "Sales vs Quota Dashboard"
6. Click Done

To add a filter (like Region):

- Click on a chart → use the dropdown arrow (top-right of that chart) → choose Show Filter
- The Region filter appears on the right

Now your dashboard is interactive.

Step 11 — Save & Publish

1. Click File → Save to Tableau Public

2. Enter a name (e.g., "Sales_Quota_Relationship_Dashboard")
3. Click Save

It publishes your work — Tableau takes you to your profile page. You can copy the Share link and view it live online.

Summary — What You Just Learned

Concept	What You Did	Example
Relationships	Linked tables using common fields	Sales.CustomerID = Customers.CustomerID
Metadata Editing	Renamed, changed types	Sales → Revenue
Calculated Field	Created new metric	[Revenue] * 0.3
Joins vs Relationships	Replaced joins with smart dynamic linking	Relationship lines
Blending	Used multiple sources (Sales, Customers, Quotas)	Combined visually
Dashboard	Combined multiple sheets	Sales vs Quota view
Publishing	Saved to Tableau Public	Shareable link

Optional Next Hands-on

Once this works:

1. Add more data to your Google Sheet (e.g., new regions, customers)
2. Refresh your Tableau workbook online — it updates automatically
3. Try creating a calculated field: `[Performance %] = ([Revenue] / [Quota]) * 100`
4. Then visualize it as a bar chart with color gradient