Write an application using HBase and HiveQL for flight information system which will include a. Create Flight Info Hbase Table(with Flight information, schedule, and delay) b. Demonstrate Creating, Dropping, and altering Database tables in Hbase c. Creating an external Hive table to connect to the HBase for Flight Information Table d. Find the total departure delay in Hive e. Find the average departure delay in Hive f. Create index on Flight information Table I am using cloudera vm tfor this expalin detailed steps

**ChatGPT said:**

Here's a detailed step-by-step guide to building a **Flight Information System using HBase and HiveQL** on your **Cloudera VM**. The system will include table creation, connection between HBase and Hive, queries, and indexing.

**⚙️ Prerequisites:**

Ensure these services are running in Cloudera VM:

* HBase
* Hive
* HDFS
* Hadoop MapReduce or YARN
* HCatalog (for Hive-HBase integration)

**Step A: Create Flight Info HBase Table**

**Open HBase Shell in Cloudera VM:**

bash

CopyEdit

hbase shell

**Create HBase Table flight\_info:**

hbase

CopyEdit

create 'flight\_info', 'info'

This creates an HBase table named flight\_info with a column family info.

**Add Sample Data:**

hbase

CopyEdit

put 'flight\_info', 'FL001', 'info:flight\_number', 'AI202'

put 'flight\_info', 'FL001', 'info:origin', 'DEL'

put 'flight\_info', 'FL001', 'info:destination', 'BOM'

put 'flight\_info', 'FL001', 'info:schedule', '2025-05-07 10:00'

put 'flight\_info', 'FL001', 'info:delay', '15'

put 'flight\_info', 'FL002', 'info:flight\_number', 'AI203'

put 'flight\_info', 'FL002', 'info:origin', 'BLR'

put 'flight\_info', 'FL002', 'info:destination', 'DEL'

put 'flight\_info', 'FL002', 'info:schedule', '2025-05-07 12:00'

put 'flight\_info', 'FL002', 'info:delay', '0'

**Step B: HBase Table Operations**

**Drop Table:**

hbase

CopyEdit

disable 'flight\_info'

drop 'flight\_info'

**Alter Table (Add Column Family):**

hbase

CopyEdit

alter 'flight\_info', NAME => 'status'

**Step C: Create External Hive Table for HBase Table**

**Launch Hive Shell:**

bash

CopyEdit

hive

**Create Hive External Table Mapped to HBase:**

sql

CopyEdit

CREATE EXTERNAL TABLE flight\_info\_hive(

rowkey STRING,

flight\_number STRING,

origin STRING,

destination STRING,

schedule STRING,

delay INT

)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES (

"hbase.columns.mapping" = ":key,info:flight\_number,info:origin,info:destination,info:schedule,info:delay"

)

TBLPROPERTIES (

"hbase.table.name" = "flight\_info"

);

✅ Now your Hive table flight\_info\_hive is connected to HBase table flight\_info.

**Step D: Find Total Departure Delay in Hive**

sql

CopyEdit

SELECT SUM(delay) AS total\_delay FROM flight\_info\_hive;

**Step E: Find Average Departure Delay in Hive**

sql

CopyEdit

SELECT AVG(delay) AS average\_delay FROM flight\_info\_hive;

**Step F: Create Index on Flight Number (Hive Table)**

**Note**: Hive indexes are largely deprecated in modern usage, but if you're on older Hive version:

sql

CopyEdit

CREATE INDEX idx\_flight\_number

ON TABLE flight\_info\_hive (flight\_number)

AS 'COMPACT'

WITH DEFERRED REBUILD;

ALTER INDEX idx\_flight\_number ON flight\_info\_hive REBUILD;

📌 **Note:** Hive indexing is often not supported or useful with external tables, especially ones backed by HBase. For performance, prefer **partitioning**, **bucketing**, or **materialized views** instead in production.

**✅ Summary:**

| **Step** | **Task** |
| --- | --- |
| A | Created flight\_info HBase table |
| B | Performed table operations: create, alter, drop |
| C | Linked HBase with Hive using external table |
| D | Queried total delay |
| E | Queried average delay |
| F | Created index (optional / legacy feature) |

Would you like a script file combining all commands to run on your Cloudera VM?

2/2

4o

**You said:**

Write an application using HBase and HiveQL for Customer information system which will include a. Creation of –Cutomer\_info(Cust-ID,Cust-Name,orderID), order\_info(OrderID,ItemID,Quantity), item\_info(Item-ID,Item-Name,ItemPrice) tables in Hive b. Load table with data from local storage in Hive. c. Perform Join tables with Hive d. Create Index on Customer information system in Hive. e. Find the total, average sales in Hive f. Find Order details with maximum cost. g. Creating an external Hive table to connect to the HBase for Customer Information System. h. Display records of Customer Information Table in Hbase.

**ChatGPT said:**

Here is a **complete Hive and HBase-based application** for a **Customer Information System** on **Cloudera VM**, covering table creation, data loading, queries, joins, indexing, and integration with HBase.

**🔧 Prerequisites**

Make sure the following services are **running** on your Cloudera VM:

* Hive
* HBase
* HDFS
* YARN
* HCatalog (for Hive-HBase integration)

**🅰️ a. Create Tables in Hive**

**Launch Hive:**

bash

CopyEdit

hive

**Create customer\_info Table:**

sql

CopyEdit

CREATE TABLE customer\_info (

cust\_id INT,

cust\_name STRING,

order\_id INT

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

**Create order\_info Table:**

sql

CopyEdit

CREATE TABLE order\_info (

order\_id INT,

item\_id INT,

quantity INT

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

**Create item\_info Table:**

sql

CopyEdit

CREATE TABLE item\_info (

item\_id INT,

item\_name STRING,

item\_price FLOAT

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

**🅱️ b. Load Table with Data from Local Storage**

**Example data files:**

Place these files under /home/cloudera/customer\_data/ in Cloudera VM:

* customer\_info.txt

CopyEdit

1,John,1001

2,Alice,1002

3,Bob,1003

* order\_info.txt

yaml

CopyEdit

1001,2001,2

1002,2002,1

1003,2001,3

* item\_info.txt

yaml

CopyEdit

2001,Laptop,50000

2002,Phone,20000

**Load Data into Tables:**

sql

CopyEdit

LOAD DATA LOCAL INPATH '/home/cloudera/customer\_data/customer\_info.txt' INTO TABLE customer\_info;

LOAD DATA LOCAL INPATH '/home/cloudera/customer\_data/order\_info.txt' INTO TABLE order\_info;

LOAD DATA LOCAL INPATH '/home/cloudera/customer\_data/item\_info.txt' INTO TABLE item\_info;

**🆑 c. Perform Join Tables in Hive**

**Join to Get Full Order Details with Cost:**

sql

CopyEdit

SELECT

c.cust\_id,

c.cust\_name,

i.item\_name,

o.quantity,

i.item\_price,

(o.quantity \* i.item\_price) AS total\_cost

FROM

customer\_info c

JOIN

order\_info o ON c.order\_id = o.order\_id

JOIN

item\_info i ON o.item\_id = i.item\_id;

**🅳 d. Create Index on Customer Table**

Hive indexing is rarely used now, but here's the legacy syntax:

sql

CopyEdit

CREATE INDEX idx\_customer\_name

ON TABLE customer\_info (cust\_name)

AS 'COMPACT'

WITH DEFERRED REBUILD;

ALTER INDEX idx\_customer\_name ON customer\_info REBUILD;

**🅴 e. Find Total and Average Sales**

sql

CopyEdit

SELECT

SUM(o.quantity \* i.item\_price) AS total\_sales,

AVG(o.quantity \* i.item\_price) AS average\_sales

FROM order\_info o

JOIN item\_info i ON o.item\_id = i.item\_id;

**🅵 f. Find Order Details with Maximum Cost**

sql

CopyEdit

SELECT

o.order\_id,

i.item\_name,

o.quantity,

i.item\_price,

(o.quantity \* i.item\_price) AS total\_cost

FROM order\_info o

JOIN item\_info i ON o.item\_id = i.item\_id

ORDER BY total\_cost DESC

LIMIT 1;

**🅶 g. Create External Hive Table Connected to HBase**

**Step 1: Create HBase Table**

bash

CopyEdit

hbase shell

hbase

CopyEdit

create 'customer\_info\_hbase', 'info'

put '1', 'info:cust\_name', 'John'

put '1', 'info:order\_id', '1001'

put '2', 'info:cust\_name', 'Alice'

put '2', 'info:order\_id', '1002'

**Step 2: Create Hive Table Linked to HBase Table**

sql

CopyEdit

CREATE EXTERNAL TABLE customer\_info\_ext (

key STRING,

cust\_name STRING,

order\_id STRING

)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES (

"hbase.columns.mapping" = ":key,info:cust\_name,info:order\_id"

)

TBLPROPERTIES (

"hbase.table.name" = "customer\_info\_hbase"

);

**🅷 h. Display Records of Customer Information from HBase**

sql

CopyEdit

SELECT \* FROM customer\_info\_ext;

**✅ Summary**

| **Task** | **Operation** |
| --- | --- |
| a | Created Hive tables for customer, order, and item info |
| b | Loaded data from local files |
| c | Performed joins to fetch complete order info |
| d | Created index on customer table |
| e | Calculated total and average sales |
| f | Found order with maximum cost |
| g | Created external Hive table connected to HBase |
| h | Displayed HBase table data in Hive |

Would you like me to generate a script file or all commands in .hql and .sh format for easier execution?

4o

**You said:**

Write an application using HBase and HiveQL for OnlineRetail Dataset which will include i. Create and Load table with Online Retail data in Hive. j. Create Index on Online Retail Table in Hive. k. Find the total, average sales in Hive l. Find Order details with maximum cost. m. Find Customer details with maximum order total. n. Find the Country with maximum and minimum sale. o. Creating an external Hive table to connect to the HBase for OnlineRetail. p. Display records of OnlineRetail Table in Hbase.

**ChatGPT said:**

Here is a **step-by-step Hive + HBase application** using the **OnlineRetail dataset**. This includes Hive table creation, loading data, running sales analytics, connecting Hive with HBase, and querying HBase through Hive.

**🗂️ Dataset: *Online Retail (UCI Repository)***

Expected columns (typical format):

* **InvoiceNo**
* **StockCode**
* **Description**
* **Quantity**
* **InvoiceDate**
* **UnitPrice**
* **CustomerID**
* **Country**

**✅ i. Create and Load Hive Table for Online Retail Data**

**Step 1: Launch Hive shell**

bash

CopyEdit

hive

**Step 2: Create Hive Table**

sql

CopyEdit

CREATE TABLE online\_retail (

invoice\_no STRING,

stock\_code STRING,

description STRING,

quantity INT,

invoice\_date STRING,

unit\_price FLOAT,

customer\_id STRING,

country STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

**Step 3: Load Data from Local File**

Place file in VM:

bash

CopyEdit

# Move file to local folder

mv OnlineRetail.csv /home/cloudera/data/

Load it into Hive:

sql

CopyEdit

LOAD DATA LOCAL INPATH '/home/cloudera/data/OnlineRetail.csv' INTO TABLE online\_retail;

**✅ j. Create Index on Online Retail Table**

sql

CopyEdit

CREATE INDEX idx\_customer\_id

ON TABLE online\_retail (customer\_id)

AS 'COMPACT'

WITH DEFERRED REBUILD;

ALTER INDEX idx\_customer\_id ON online\_retail REBUILD;

🛈 Hive indexing is deprecated in modern versions; better alternatives: **partitioning** or **materialized views**.

**✅ k. Find Total, Average Sales in Hive**

Sales = quantity \* unit\_price

sql

CopyEdit

SELECT

SUM(quantity \* unit\_price) AS total\_sales,

AVG(quantity \* unit\_price) AS average\_sales

FROM online\_retail;

**✅ l. Find Order Details with Maximum Cost**

Assuming invoice\_no is the order ID:

sql

CopyEdit

SELECT

invoice\_no,

SUM(quantity \* unit\_price) AS order\_total

FROM online\_retail

GROUP BY invoice\_no

ORDER BY order\_total DESC

LIMIT 1;

**✅ m. Find Customer with Maximum Order Total**

sql

CopyEdit

SELECT

customer\_id,

SUM(quantity \* unit\_price) AS total\_spent

FROM online\_retail

GROUP BY customer\_id

ORDER BY total\_spent DESC

LIMIT 1;

**✅ n. Find Country with Maximum and Minimum Sale**

sql

CopyEdit

-- Maximum Sale

SELECT

country,

SUM(quantity \* unit\_price) AS total\_sales

FROM online\_retail

GROUP BY country

ORDER BY total\_sales DESC

LIMIT 1;

-- Minimum Sale

SELECT

country,

SUM(quantity \* unit\_price) AS total\_sales

FROM online\_retail

GROUP BY country

ORDER BY total\_sales ASC

LIMIT 1;

**✅ o. Create External Hive Table Connected to HBase**

**Step 1: Create HBase Table**

bash

CopyEdit

hbase shell

hbase

CopyEdit

create 'online\_retail\_hbase', 'info'

put '1', 'info:invoice\_no', '536365'

put '1', 'info:description', 'WHITE HANGING HEART T-LIGHT HOLDER'

put '1', 'info:quantity', '6'

put '1', 'info:unit\_price', '2.55'

put '1', 'info:customer\_id', '17850'

put '1', 'info:country', 'United Kingdom'

**Step 2: Create External Hive Table for HBase Table**

CREATE EXTERNAL TABLE online\_retail\_hbase (

rowkey STRING,

invoice\_no STRING,

description STRING,

quantity INT,

unit\_price FLOAT,

customer\_id STRING,

country STRING

)

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES (

"hbase.columns.mapping" = ":key,info:invoice\_no,info:description,info:quantity,info:unit\_price,info:customer\_id,info:country"

)

TBLPROPERTIES (

"hbase.table.name" = "online\_retail\_hbase"

);

**✅ p. Display Records from HBase in Hive**

sql

CopyEdit

SELECT \* FROM online\_retail\_hbase;

**📘 Summary**

| **Task** | **Action** |
| --- | --- |
| i | Created and loaded Hive table for OnlineRetail |
| j | Created index on customer\_id |
| k | Calculated total and average sales |
| l | Found order with maximum cost |
| m | Found customer with highest spending |
| n | Identified countries with max/min sales |
| o | Created Hive-HBase external table |
| p | Displayed HBase records via Hive |

Would you like me to generate a .hql script or bash automation for this entire setup?

4o

Command for shared foler

su

mount -t vboxsf data data

Tableau

Perform the data visualization operations using Tableau to get answers to various business questions on Retail dataset. a. Find and Plot top 10 products based on total sale b. Find and Plot product contribution to total sale c. Find and Plot the month wise sales in year 2010 in descending order d. Find and Plot most loyal customers based on purchase order e. Find and Plot yearly sales comparison f. Find and Plot country wise total sales price and show on Geospatial graph tel me detailed steps to perform each operation InvoiceNo StockCode Description Quantity InvoiceDate UnitPrice CustomerID Country this are column names tell me detailed steps from starting tableau its fetures where are they

**ChatGPT said:**

To perform data visualization using Tableau on your Retail dataset with columns like:

InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country

Here are the complete, detailed steps from launching Tableau to building the required dashboards and visualizations.

🟢 STEP 1: Starting Tableau and Importing the Dataset

1. Launch Tableau Desktop (or Tableau Public).
2. Click on “File” → “Open” or directly choose “Connect” → “Microsoft Excel” (if your file is .xlsx) or “Text File” (if CSV).
3. Locate and load your Retail dataset file.
4. Once loaded, go to the “Sheet 1” tab at the bottom to begin visualization.

🟢 STEP 2: Understand Tableau Interface

Top Features You’ll Use:

* Dimensions (blue): Categorical fields (like Description, Country, InvoiceNo)
* Measures (green): Numerical fields (like Quantity, UnitPrice)
* Rows & Columns: For building charts
* Filters: To limit what data is shown
* Marks Card: To control chart type, color, label, tooltip, size, etc.
* Show Me: Suggests chart types based on selected fields

Now let’s go through each business question:

───────────────────────────────  
A. 📊 Top 10 Products by Total Sales  
───────────────────────────────

1. Go to Sheet 1 and rename it “Top 10 Products”.
2. Create a Calculated Field:
   * Click on Analysis → Create Calculated Field
   * Name: Total Sales
   * Formula: [Quantity] \* [UnitPrice]
   * Click OK.
3. Drag “Description” to Rows.
4. Drag “Total Sales” to Columns.
5. Click on the drop-down on “Total Sales” axis → Sort Descending.
6. Click on “Top N” filter:
   * Drag “Description” to Filters → Top → By Field → Top 10 by Total Sales.
7. You can choose Bar Chart from “Show Me” or Marks → Bar.
8. Optional: Drag “Total Sales” to Label on the Marks card.

───────────────────────────────  
B. 🧮 Product Contribution to Total Sales  
───────────────────────────────

1. Create a new sheet → rename to “Product Contribution”.
2. Drag “Description” to Rows.
3. Drag “Total Sales” to Columns.
4. Click on the drop-down of “Total Sales” pill → Quick Table Calculation → Percent of Total.
5. Choose Pie Chart from “Show Me”.
6. Drag “Description” to Label.
7. Drag “Percent of Total” to Label and Tooltip.
8. Drag “Total Sales” to Color to visualize magnitude.

──────────────────────────────────────────  
C. 📆 Month-wise Sales in 2010 (Descending)  
──────────────────────────────────────────

1. Create a new sheet → rename to “Monthly Sales 2010”.
2. Create a new Calculated Field: Year = YEAR([InvoiceDate])
3. Drag it to Filters → select 2010.
4. Drag InvoiceDate to Columns → right-click → Select “Month” (not continuous).
5. Drag Total Sales to Rows.
6. Sort descending: Click axis → Sort descending icon.
7. Optional: Change to Bar or Line Chart from “Show Me”.

────────────────────────────────────────────  
D. 🧑‍🤝‍🧑 Most Loyal Customers (By Order Count)  
────────────────────────────────────────────

1. Create a new sheet → rename to “Loyal Customers”.
2. Drag CustomerID to Rows.
3. Drag InvoiceNo to Columns → change Measure from SUM to Count (right-click InvoiceNo → Measure → Count).
4. Sort in Descending Order.
5. Apply Top 10 filter on CustomerID.
6. Optional: Add bar chart and label the count.

────────────────────────────  
E. 📅 Yearly Sales Comparison  
────────────────────────────

1. Create a new sheet → rename to “Yearly Comparison”.
2. Drag InvoiceDate to Columns → right-click → Choose Year.
3. Drag Total Sales to Rows.
4. Optional: Convert to Bar or Line Chart.
5. Drag Total Sales to Label and Tooltip.

──────────────────────────────────────────────  
F. 🗺️ Country-wise Sales on Geospatial Map  
──────────────────────────────────────────────

1. Create a new sheet → rename to “Country Sales Map”.
2. Drag Country to Rows.
3. Drag Total Sales to Columns.
4. Drag Country to Detail on the Marks card.
5. Choose Map from “Show Me”.
6. Drag Total Sales to Color and Size for visual weight.
7. Optional: Add Total Sales to Tooltip and Label.

📌 Optional Enhancements

* Use Dashboards to combine all sheets into a single interactive view.
* Add filters for Year, Country, Product etc. for user interactivity.

Perform the data visualization operations using Tableau to get answers to various business questions on Retail dataset. a. Find and Plot country wise popular product b. Find and Plot bottom 10 products based on total sale c. Find and Plot top 5 purchase order d. Find and Plot most popular products based on sales e. Find and Plot half yearly sales for the year 2011 f. Find and Plot country wise total sales quantity and show on Geospatial graph also this in detailed

**ChatGPT said:**

Here is a detailed guide to perform these new Tableau visualizations using your Retail dataset with columns:

InvoiceNo | StockCode | Description | Quantity | InvoiceDate | UnitPrice | CustomerID | Country

Let’s go through each business question step-by-step in Tableau:

─────────────────────────────────────────────  
A. 🌍 Country-wise Popular Product  
─────────────────────────────────────────────

Objective: For each Country, show the most sold product (by Quantity)

Step-by-step:

1. Drag Country to Rows
2. Drag Description to Columns
3. Drag Quantity to Size or Tooltip
4. Now, apply a Top 1 filter per Country:
   * Drag Description to Filters → go to "Top" tab
   * Select “By Field” → Top 1 by SUM(Quantity)
   * Click “Add Context” if prompted
5. To ensure it’s grouped per country:
   * Drag Country to Filters first → then Description
6. Or, build a Set or use LOD expression (if advanced)
7. Use a Bar chart, and show Label as Description and Quantity

📝 Optional: Create a Tooltip showing total units sold for each product-country combo.

─────────────────────────────────────────────  
B. ⬇ Bottom 10 Products by Total Sales  
─────────────────────────────────────────────

Objective: Show 10 products with the lowest total sales.

Steps:

1. Create a calculated field:  
   Total Sales = [Quantity] \* [UnitPrice]
2. Drag Description to Rows
3. Drag Total Sales to Columns
4. Sort → Ascending (Low to High)
5. Drag Description to Filters → Top → select “By Field” → Bottom 10 by SUM(Total Sales)
6. Choose Bar chart, add labels

📌 Tip: You can also filter out negative or zero-value products if needed.

─────────────────────────────────────────────  
C. 🧾 Top 5 Purchase Orders (by value)  
─────────────────────────────────────────────

Objective: Show top 5 invoices with the highest total sale.

Steps:

1. Drag InvoiceNo to Rows
2. Drag Total Sales to Columns (use the same calculated field)
3. Sort in Descending order
4. Drag InvoiceNo to Filters → Top tab → Top 5 by Total Sales
5. Choose Horizontal Bar chart
6. Add Label to show total sale

📝 Tooltip can include CustomerID and Country.

─────────────────────────────────────────────  
D. ⭐ Most Popular Products by Sales  
─────────────────────────────────────────────

Objective: Show top products based on total sales (not quantity).

Steps:

1. Use your Total Sales field
2. Drag Description to Rows
3. Drag Total Sales to Columns
4. Sort descending
5. (Optional) Filter to Top N:
   * Drag Description to Filters → Top tab → Top 10 or Top 20
6. Choose Bar chart, show Total Sales as Label

🎯 This tells you which products earn the most revenue.

─────────────────────────────────────────────  
E. 📆 Half-Yearly Sales for 2011  
─────────────────────────────────────────────

Objective: Compare Jan–Jun and Jul–Dec sales in 2011

Steps:

1. Drag InvoiceDate to Filters → Choose Years → Select 2011
2. Create a calculated field called “HalfYear”:  
   IF MONTH([InvoiceDate]) <= 6 THEN "H1" ELSE "H2" END
3. Drag “HalfYear” to Columns
4. Drag Total Sales to Rows
5. Choose Bar Chart
6. Show labels for clarity

✅ This clearly compares sales performance across H1 and H2 of 2011.

──────────────────────────────────────────────────  
F. 🌎 Country-wise Total Sales Quantity on Map  
──────────────────────────────────────────────────

Objective: Show countries on a map sized by Quantity sold.

Steps:

1. Drag Country to Rows
2. Drag Quantity to Columns
3. Click “Show Me” → Choose “Map”
4. Tableau will geocode Country automatically
5. Drag Quantity to Size on Marks card
6. Drag Quantity to Color to show gradient
7. Optional:
   * Add Quantity to Tooltip
   * Add Labels for quantity

🎯 You now have a geospatial map showing where most items were sold.

──────────────────────  
📌 Optional Dashboard  
──────────────────────

* Click “New Dashboard” at the bottom
* Drag and drop all sheets to combine them into a single interactive view
* Add filters for Year, Country, Product
* Use highlight actions and drop-down selectors

Perform the data visualization operations using Tableau to get answers to various questions on the census bureau databset(Adult data sets). a. Find and Plot Income class of People whose education is master’s and doctorate. b. Find and Plot Income class of people who have private jobs. c. Find and Plot yearly sales comparison d. Find and Plot country wise statistics on Geospatial graph e. Plot agewise- education vs salary statistics. f. Plot Countrywise male female ratio. g. Plot Income class based on workclass(Government and other) explain each step in detailed how to do it

**ChatGPT said:**

Here's a step-by-step guide to performing **data visualization in Tableau** using the **Census Bureau Adult dataset (UCI Adult Dataset)** for the listed tasks. This explanation assumes you have already cleaned and prepared the dataset (e.g., missing values handled, appropriate column names assigned).

**✅ Initial Setup: Import Data into Tableau**

1. **Open Tableau Desktop or Tableau Public**.
2. Click on **"Connect" → "Text File"** and select the CSV file of the **Adult dataset**.
3. Tableau loads the dataset into **Data Source tab** – preview it to ensure columns are properly detected (e.g., age, workclass, education, sex, salary, native-country).

Ensure columns like education, salary, workclass, native-country, etc., are treated as **dimensions** and age, hours-per-week, etc., as **measures**.

**📊 a. Find and Plot Income class of People whose education is Master's and Doctorate**

**Steps:**

1. Drag **education** to the **Filters** shelf.
2. In the filter box, select only **"Masters"** and **"Doctorate"**.
3. Drag **salary** (income class: <=50K, >50K) to **Columns**.
4. Drag **Number of Records** to **Rows**.
5. Add **education** to **Color** for differentiation.

📘 **Chart Type**: Use a **Bar Chart** (automatically selected).

✅ You’ll see a bar plot comparing income classes for individuals with Master's and Doctorate degrees.

**📊 b. Find and Plot Income class of people who have private jobs**

**Steps:**

1. Drag **workclass** to Filters and choose **"Private"**.
2. Drag **salary** to Columns.
3. Drag **Number of Records** to Rows.
4. Add **salary** to Color to distinguish income class.

📘 **Chart Type**: Bar Chart

✅ This chart shows income distribution for people working in the private sector.

**📊 c. Find and Plot Yearly Sales Comparison**

**Note**: The original dataset doesn’t have "yearly sales". If you assume "capital-gain" + "capital-loss" as a proxy for sales:

**Steps:**

1. Create a **calculated field**:
   * Name: Yearly Sales
   * Formula: [capital-gain] + [capital-loss]
2. Drag Yearly Sales to Rows.
3. Drag salary or education to Columns (for category-wise comparison).
4. Use **SUM(Yearly Sales)** as aggregation.

📘 **Chart Type**: Use **Bar Chart** or **Line Chart** (if time-based data exists).

✅ Displays total "yearly sales" (as proxy) per group.

**📊 d. Find and Plot Country-wise Statistics on Geospatial Graph**

**Steps:**

1. Drag **native-country** to Rows.
2. Go to **Show Me** and select **Map (Filled Map)**.
3. Drag **Number of Records** or **salary** to **Color**.
4. Tableau automatically geocodes countries and plots them.

📘 **Chart Type**: **Filled Map**

✅ Shows population distribution or salary distribution per country.

**📊 e. Plot Age-wise Education vs Salary Statistics**

**Steps:**

1. Drag **age** to Columns.
2. Drag **Number of Records** to Rows.
3. Drag **education** to **Color**.
4. Drag **salary** to **Filter** or **Detail** to analyze income class.

📘 **Chart Type**: **Line Chart** or **Stacked Bar Chart**

✅ This will show how education level and income relate to age.

**📊 f. Plot Country-wise Male Female Ratio**

**Steps:**

1. Drag **native-country** to Rows.
2. Drag **sex** to Columns.
3. Drag **Number of Records** to Size or Rows.
4. Drag **sex** to Color for gender differentiation.

📘 **Chart Type**: **Side-by-side Bar Chart**

✅ You’ll see a comparison of males and females in each country.

**📊 g. Plot Income Class Based on Workclass (Government and Other)**

**Steps:**

1. Create a **calculated field**:
   * Name: Work Type
   * Formula:

sql

CopyEdit

IF CONTAINS([workclass], "gov") THEN "Government"

ELSE "Other"

END

1. Drag **Work Type** to Columns.
2. Drag **salary** to Color.
3. Drag **Number of Records** to Rows.

📘 **Chart Type**: **Bar Chart**

✅ Compares income distribution across government vs other job sectors.

**💡 Tips:**

* Use **Tooltips** to enhance interactivity.
* Use **Filters** to narrow down specific age groups, countries, or education levels.
* Use **Dashboards** to combine multiple visualizations into one interactive view.
* Format charts (labels, colors, axes) for clarity.

Would you like a sample Tableau workbook or need help building a specific chart in Tableau?

4o

**You said:**

age workclass fnlwgt education educational-num marital-status occupation relationship race gender capital-gain capital-loss hours-per-week native-country income this are my column names give accordingly

**ChatGPT said:**

Thanks for sharing your column names. Based on your dataset, here's the mapping for your Tableau tasks using the actual column names:

**✅ COLUMN NAME MAPPING (based on your image):**

| **Task Concept** | **Your Column Name in Dataset** |
| --- | --- |
| Age | age |
| Workclass | workclass |
| Education | education |
| Education-Num | educational-num |
| Marital Status | marital-status |
| Occupation | occupation |
| Relationship | relationship |
| Race | race |
| Gender | gender |
| Capital Gain | capital-gain |
| Capital Loss | capital-loss |
| Hours Per Week | hours-per-week |
| Country (Native) | native-country |
| Income Class | income (<=50K or >50K) |

Now let’s walk through the steps **again, tailored with these exact column names**:

**a. Income class of people with Master's and Doctorate**

1. Drag education to **Filters** → select **Master's** and **Doctorate**.
2. Drag income to **Columns**.
3. Drag Number of Records to **Rows**.
4. Drag education to **Color** or **Detail**.

✅ Bar chart comparing income class for Master’s and Doctorate holders.

**b. Income class of people with private jobs**

1. Drag workclass to **Filters** → select **Private**.
2. Drag income to **Columns**.
3. Drag Number of Records to **Rows**.

✅ Bar chart showing income class distribution for private sector employees.

**c. Yearly Sales Comparison**

Assuming capital-gain + capital-loss as a proxy for yearly sales:

1. Create a **Calculated Field** called Yearly Sales:

tableau

CopyEdit

[capital-gain] + [capital-loss]

1. Drag income or education to **Columns**.
2. Drag Yearly Sales to **Rows** → use **SUM aggregation**.
3. (Optional) Drag education or income to **Color**.

✅ Displays total proxy "sales" for different groups.

**d. Country-wise Statistics on Geospatial Map**

1. Drag native-country to **Rows**.
2. Click **Show Me** → select **Map (Filled)**.
3. Drag Number of Records or income to **Color** or **Size**.

✅ Displays world map showing country-wise population/income.

**e. Age-wise Education vs Salary Statistics**

1. Drag age to **Columns**.
2. Drag Number of Records to **Rows**.
3. Drag education to **Color**.
4. Drag income to **Detail** or Filter.

✅ Line chart or bar chart showing how education and salary vary with age.

**f. Country-wise Male Female Ratio**

1. Drag native-country to **Rows**.
2. Drag gender to **Columns**.
3. Drag Number of Records to **Rows**.
4. Drag gender to **Color**.

✅ Side-by-side bar chart of male vs female in each country.

**g. Income Class Based on Workclass (Government and Other)**

1. Create a **Calculated Field** called Work Type:

tableau

CopyEdit

IF CONTAINS(LOWER([workclass]), "gov") THEN "Government"

ELSE "Other"

END

1. Drag Work Type to **Columns**.
2. Drag income to **Color**.
3. Drag Number of Records to **Rows**.

✅ Bar chart comparing income class for government and other sectors.