

Final Date is 28 FEB for Submission

Note: Every team member has to fill the Project Submission link

Team leader should upload all the files in Github and same github link should be used by all team members while filling the submission form

Project Submission Link:

https://docs.google.com/forms/d/e/1FAIpQLSeE_41KJTO6cl_iMHAamAdpwbYoopbKYa61GkiUu-aKcSUrlw/viewform

Telegram- <https://t.me/+4lbrgppsNbYxZDc1>

Email Support- teamsip@thesmartbridge.com

Project Submission Starting from 15 Feb Onwards

Mandatory Check Points

1. FSP Registration along with DAF(Stem) Course Enrolment
2. Service Now Registration with Service Now Zurich Enrolment
3. Complete and Submit the IBM Self Paced Course Badge/Certificate
4. Attempted Grand Assessment
5. Submit the Project Github repository link

<https://github.com/Indrac28/Heart-Disease-Analysis>

Data Analytics Self Paced Submission Form: <https://forms.gle/S9KvVktVuQ1nRkPR7>

FSP Registration:

https://drive.google.com/file/d/1czeV1w60_zy0THCgBNaMv3YOunCIIG4X/view?usp=sharing

Service Now Course

<https://skillwalletapp.s3.ap-south-1.amazonaws.com/ServiceNow+Nowlearning+Account+Creation+%26+Enrollment.pdf>

Updated Template Link:

<https://docs.google.com/spreadsheets/d/1BMCcONU1FE7LDL43-CNtFVQf2p8w5EI0SgxHnustt8Y/edit?usp=sharing>

Queries Sheet:

<https://docs.google.com/spreadsheets/d/16e1HIJqcMyveMVmAeHCcHBscQvDDRFnyvqytz2t2HOY/edit?usp=sharing>

Empathy Map:

<https://app.mural.co/template/02299a19-f773-4eef-bb2c-9f4f1a5d8895/06cfc00c-75a9-4acd-92d5-ced90108a96f>

Assignment 1

https://drive.google.com/file/d/1O6OfjByKqR3_N1VapR99ZJqIPNRGrURS/view?usp=sharing

Assignment 2:

<https://drive.google.com/file/d/1LY7y9K311f8dPZE44dyyGf9Onp8RV1IS/view?usp=sharing>

Url : <https://apsche.smartinternz.com/>

My SQL Workbench

Link: <https://dev.mysql.com/downloads/workbench/>

Link: <https://www.youtube.com/watch?v=Rxp3T5GKIR4>

Dataset: https://drive.google.com/file/d/1i1lghiLNgW2qF_vSxQASB9DuJQDyE-q3/view

Tableau Prep:

<https://www.tableau.com/products/prep/download>

Tableau Prep Dataset:

https://help.tableau.com/current/prep/en-us/prep_get_started.htm

APSCHE Long Term Project – Submission Phase
Guidelines

Step 1: FSP NASSCOM Registration

Website: <https://www.futureskillsprime.in/>

Video Guide: <https://www.youtube.com/watch?v=hskRDGntFTw>

Step 2: IBM Self-Paced Learning

[https://skills.yourlearning.ibm.com/activity/PLAN-1C903152880C?ngo-id=0302&utm_campaign=](https://skills.yourlearning.ibm.com/activity/PLAN-1C903152880C?ngo-id=0302&utm_campaign=aca-smartbridge-APSCHELT-event#1)

[aca-smartbridge-APSCHELT-event#1](https://skills.yourlearning.ibm.com/activity/PLAN-1C903152880C?ngo-id=0302&utm_campaign=aca-smartbridge-APSCHELT-event#1)

Download the course completion certificate after finishing.

Step 3: Task Completion

Complete all tasks as per the given guidelines.

Enrollment Guide:

[https://skillwalletapp.s3.ap-south-1.amazonaws.com/ServiceNow+Nowlearning+Account+Cr](https://skillwalletapp.s3.ap-south-1.amazonaws.com/ServiceNow+Nowlearning+Account+Creation+%26+Enrollment.pdf)

[eat](https://skillwalletapp.s3.ap-south-1.amazonaws.com/ServiceNow+Nowlearning+Account+Creation+%26+Enrollment.pdf)

Step 4: Grand Assessment

<https://forms.gle/AQpT2VsYc4pxQsiE9>

Step 5: Project Submission

♦ PROJECT SUBMISSION INSTRUCTIONS

Before submitting your project, ensure the following:

1. Only students who have successfully completed the project should fill out the form.
2. The form allows only one response. Enter all details correctly.
3. Make sure all certificates and required documents are ready before opening the form.
4. Kindly follow the project template format while uploading the project.
5. Follow the step-wise submission process carefully.

📌 Final Step – Project Submission

Submit your project files according to the required format provided below:

👉 Submission Format Folder:

https://drive.google.com/drive/folders/1m_vXdKkujfkVq1x57h6bZ3Kj_07hdh4T?usp=sharing

Make sure you strictly follow the given format while uploading your project files.

GitHub Video Guide:

<https://youtu.be/6l0UR7Co7YA>

Use Case

1. Cosmetic Insights _ Navigating Cosmetics Trends and Consumer Insights with Tableau
<https://www.kaggle.com/datasets/kingabzpro/cosmetics-datasets>
2. Visualization Tool for Electric Vehicle Charge and Range Analysis-Updated
<https://drive.google.com/drive/folders/1Rkzdk6Us1Uq2SRB4nxMAb83jN5bpHl>
3. ToyCraft Tales _ Tableau's Vision into Toy Manufacturer Data
<https://www.kaggle.com/datasets/thedevastator/toy-manufacturers-in-us-states?select=Week+39+-+US+Toy+Manufacturers+-+2005+to+2016.hyper>
4. Measuring the pulse of prosperity: An Index of economic freedom analysis
https://drive.google.com/file/d/1EBla1LtM3Ni2Uh3nekLB6wt3263Q3NeX/view?usp=share_link
5. Heritage Treasures: An In-Depth Analysis of UNESCO World Heritage Sites in Tableau
<https://www.kaggle.com/datasets/uiwalkandi/unesco-world-heritage-sites/data?select=whc-sites-2019.csv>
6. Visualizing Housing Market Trends An Analysis of Sale Prices and Features using Tableau
<https://www.kaggle.com/datasets/rituparnaghosh18/transformed-housing-data-2>
7. Comprehensive Analysis and Dietary Strategies with Tableau_ A College Food Choices Case Study
https://www.kaggle.com/datasets/borapajo/food-choices?select=food_coded.csv
8. Plugging into the Future_ An Exploration of Electricity Consumption Patterns
https://drive.google.com/file/d/1xIkHNwXxjFztKq7ad0_KtkukCqTckNy/view?usp=sharing
9. Strategic Product Placement Analysis
<https://drive.google.com/file/d/1vHDNGw130kbYUPj-wl4640x-cz5349GM/view?usp=sharing>
10. iRevolution_ A Data-driven Exploration of Apple's iPhone Impact in India
<https://docs.google.com/spreadsheets/d/1p1ZWaYcEuFl5UNFcmNvpkXi3JnoHamut/edit?gid=1877446487#gid=1877446487>

11. Heart Disease Analysis

https://drive.google.com/file/d/190Qmq27LeZZ_nWricP3Obl7ys_5otEsp/view?usp=sharing

12. Empowering India: Analysing the Evolution of Union Budget Allocations for Sustainable Growth

<https://www.kaggle.com/datasets/prasenjitsharma/indian-union-budget-fy-21-22-till-23-24>

1950, Problem Communication

1980- Telephones

Can We Carry, ?? No.....

Mobile Phone

2020-2021– Covid...

No Medicine???

Solution-Lockdowns, Existing Medicines

Created New Medicines

Problem???

Solution- New Technology

FMCG Supermarket- ABC Supermarket in Your area

2018— Only Store

2025- Change-Increase/Decrease

5 more superstores in Your Area- Competitors

Problem??

Solutions-

- Adapt New techniques

- Innovative Strategies

- Reduced Prices

- unique ideas

- Understanding about the Competitor

Based on What?

Competition- Analysis

Customers- Understanding

????

Technology- Data Analytics– Data

AI/ML/DA

What Customer Want from Us:

1. Quality Product and Services
2. Excellent Customer Service
3. Value For Money
4. Transparency and Honesty
5. Positive Customer Experience

Customer Retention:

1. Loyalty Programs
2. Regular Communication
3. Customer Feedbacks

Gaining Customer Loyalty

1. Consistency and Reliability
2. Personalization and Customization
3. Strong Brand Identity

Increase User Experience:

1. Customer Journey Mapping
2. Continuous Improvement
3. Innovative Solutions

Understanding Dynamic Customer Behaviour

1. Adaptability
2. Monitor Socio -Economic And Cultural Factors
3. Data Analytics-

The Decision Making Process

1. Identify the Problem or Opportunity
2. Gather Information
3. Generate Options
4. Evaluate Options
5. Select The Best option
6. Implement the Decision

Data Analytics:

It is Process of using data, statistical methods and technology to find useful information from the data and make better decisions

DA Applications:

- Marketing
- Healthcare
- Retail
- Finance
- Transportation and Logistics
- Education

Data Analytics Process

1. Define Problem- Set clear objectives and Goals.
2. Data Collection- Gathering relevant data from appropriate sources
3. Data Cleaning- Prepare and ensure data quality
4. Data Processing-Transform and Organise data for Analysis
5. Data Analysis- Apply techniques to derive the Insights.
6. Interpretation-Make sense of the results and draw conclusions.
7. Communication-Present the Insights in Clear and actionable manner.

Types of Analytics:

1. Descriptive Analytics:What Happened?
2. Diagnostics Analytics:Why Did it happen?
3. Predictive Analytics:What could happen?
4. Prescriptive Analytics:What Should We Do?

1 Supermarket

1hr-20 transaction- 20 records getting generated

10hr- 200

30 days-6000—Small Data

Amazon, flipkart.....Large Data

As the Data increases difficulty to analyse the data also increases

Business Intelligence Tools

Difference Between BI Tools and Excel

1. Scalability
2. Data Integration

3. Advanced Analytics
4. Visualizations and Dashboards

Business Intelligence Tools: These are software applications designed to help organisations collect, analyse and present business data to support decision making and improve business performance.

Data Analytics- Analyzing Data in General

Business Analytics- Analyzing data with focus on improving the business outcomes

Analysis-Statistical and Visualization

Average

Lowest

Highest

Visual Analysis- 3 kinds

1. Univariate Analysis
2. Bivariate Analysis
3. Multivariate Analysis

Database

Types of Databases

1. Relational Database
2. Operational Database
3. Distributed Database
4. Cloud Database
5. Enduser Database

MySQL

Open Source Rdbms by Oracle Corporation

It Supports SQL for managing and Querying the database

MySQL Editions

1. MySQL Community Edition
2. MySQL Standard Edition

3. MySQL Enterprise Edition

SQL?? Structured Query Language

MySQL Workbench is an Integrated Development Environment(IDE) for MySQL Database development and administration.

Basic SQL Components

1. DDL- Data Definition Language- CREATE, ALTER and DROP
2. DML- Data Manipulation language- INSERT, UPDATE and DELETE
3. DQL- Data Query Language-SELECT
4. DCL- Data Control Language- GRANT and REVOKE
5. TCL- Transaction Control Language

Basic SQL Commands

1. Select- Used to retrieve the data from one or more tables in database
2. Insert- Used to add new records to a table
3. Update- Used to modify existing record in a table
4. Create- Used to create a new database objects like tables etc
5. Alter- Used to modify the structure of existing database object
6. Drop- Used to delete database object
7. Grant- Used to grant specific privileges to database users
8. Revoke- Used to revoke previously granted privileges.

```
create database users;  
use users;
```

Comments:

```
-- Single line comment  
/* multi line  
comment */
```

```
-- Abs()  
select abs(-5) as Absolute_Value;
```

```
-- Round()  
select round(3.14159) as Rounded_Value;
```

```
select round(3.14159,2) as Rounded_Value;
```

```
-- Ceil()
```

```
select Ceil(4.25) as ceil_value;
```

```
-- Floor()
```

```
Select floor(4.75) as floor_value;
```

```
-- Power()
```

```
Select Power(4,2);
```

```
Select Power(10,3) as Cubes;
```

```
-- SQRT()
```

```
Select sqrt(144);
```

```
-- Exp()
```

```
select exp(1) as exponential_value;
```

```
-- Rand() Random Floating point number between 0 and 1
```

```
select rand() as random_number;
```

```
-- Mod()
```

```
select mod(15,3) as remainder;
```

```
-- Greatest()
```

```
select greatest(2,5,18,6,12);
```

```
-- Least()
```

```
select least(2,5,18,6,12);
```

```
select truncate(22.879656,2);
```

```
select upper('New York') as Upper_case;
```

```
select lower('NEW YORK') as lower_case;
```

```
select character_length('India') as total_len;
```

```
-- Concat
```

```
select concat ('India' 'is' 'in' 'Asia') as merged;
```

```
select length('New York') as length_of_string;
```

```
select substring('DataAnalytics',1,4);
```

```
select left('Data Analysis',4);
```

```
select Trim('   MySQL   ');
```

```
Select Replace('Hello World','World','Universe') as replaced_string;  
Select Reverse('MYSQL');
```

-- Date Function

-- Curdate()

```
select curdate();
```

-- Curtime()

```
select curtime();
```

-- Now()

```
select now();
```

```
select year('2025-12-19');
```

```
select month('2025-12-19');
```

```
select monthname('2025-12-19');
```

```
select day('2025-12-19');
```

```
select dayname('2025-12-19');
```

```
select date_add('2025-12-19', INTERVAL 10 DAY);
```

```
select date_sub('2025-12-19', INTERVAL 1 Month);
```

```
select Datediff('2025-12-31','2025-12-19');
```

```
select str_to_date('19-12-2025','%d-%m-%Y');
```

```
select date_format('2025-12-19','%d-%M-%Y') as formatted_date;
```

CRUD Operations

C- Create

R- Read

U- Update

D-Delete

Primary Key

Foreign Key

Table- Student

Student_Id, Int Primary Key Auto_Increment,
Name varchar(50)
Age int
Grade varchar(5)

CREATE

```
create table student(  
student_Id Int Primary Key Auto_Increment,  
name varchar(50) not null,  
Age int,  
Grade varchar(5));
```

Insert Into student(name, Age, Grade) values

```
('John Doe',20,'A'),  
('Jane Smith',21,'B'),  
('Johny',23,'A'),  
('Sam',22,'B'),  
('Bob Johnson',19,'C');
```

READ

```
select * from student;  
select name, Grade from student;
```

UPDATE

```
update student set Age=21 where name='John Doe';
```

DELETE

```
Delete from student where name= 'Bob Johnson';  
select * from student where Grade='A';
```

JOINS

Inner Join- Returns record that have matching values in both tables

Left Join- Returns all record from the left table and the matched records from right table.

Right Join- Returns all records from Right table and the matched records from left table

Cross Join- returns the cartesian product

Create another database- Joins

1. Cricket_Students: Student_id, Name
2. Football_Students: Student_id, Name

```
create database joins;  
use joins;
```

```
create table cricket_students(  
student_id int Primary Key,  
name varchar(50));
```

```
create table football_students(  
student_id int Primary Key,  
name varchar(50));
```

```
insert into cricket_students(student_id, name) values  
(1,'Raju'),  
(2,'Mohan'),  
(3,'Suraj'),  
(4,'Karan'),  
(5,'Virat');
```

```
insert into football_students(student_id, name) values  
(2,'Mohan'),  
(3,'Suraj'),  
(5,'Virat'),  
(6,'Alex'),  
(7,'Taylor');
```

-- Inner Join

```
select cricket_students.student_id, cricket_students.name  
from cricket_students  
inner join football_students  
On
```

```
cricket_students.student_id=football_students.student_id;
```

```
select c.student_id, c.name  
from cricket_students as c  
inner join football_students as f  
On  
c.student_id=f.student_id;
```

```
-- Left Join  
select *  
from cricket_students c  
left join football_students f  
on  
c.student_id=f.student_id;
```

```
-- Right Join  
select *  
from cricket_students c  
right join football_students f  
on  
c.student_id=f.student_id;
```

```
-- Cross Join  
select *  
from cricket_students  
cross join football_students;
```

```
-- Full Outer join  
select c.student_id, c.name  
from cricket_students c  
left join football_students f  
on
```

```
c.student_id=f.student_id
```

```
union
```

```
select f.student_id,f.name  
from cricket_students c  
right join football_students f  
on  
c.student_id=f.student_id;
```

In Dataset

Segment- Consumer, Home Office, Corporate

Region- South, East, West and Central

Category- Furniture, Office Supplies and Technology

Sub- Categories- 17 Unique

Product Name

Sale, Quantity , Discount and Profit

```
create database store;
```

```
use store;
```

```
select * from superstore;
```

```
select * from superstore Limit 5;
```

```
select `Order Date`, Segment, `Sub-Category`, Sales from superstore;
```

```
-- Count()
```

```
select count(*) as total_records from superstore;
```

```
-- Sum()
```

```
select round(sum(sales)) as total_sales from superstore;
```

```
-- Average()
```

```
select round(avg(sales),2) as Average_Sales from Superstore;
```

```
-- Min()
```

```
select min(sales) as Min_sales from superstore;
```

```
-- Max Sales
```

```
select max(sales) as Max_sales from superstore;
```

```
-- Rename Column
```

```
Alter table superstore
```

```
change column `Customer Name` Customer_name varchar(255);
```

```
select * from superstore;
```

```
-- Where
```

```
Select * from superstore Where Region='Central';
```

```
Select * from superstore Where Category='Technology';
```

```
select * from superstore where Profit>2000;
```

```
select * from superstore where Category='Furniture' and Region='South';
```

```
select * from superstore where Segment = 'Corporate' and State='California';
```

```
select * from superstore where State= 'New York' or State='Texas';
```

```
Select * from superstore where Sales between 1000 and 5000;
```

```
select * from superstore where not country='United States';
```

```
--Group By
```

```
Select Column_name, Aggregated_fun(Column_name)
```

```
From table_name
```

```
Group By Column_name
```

```
Select Category, round(Sum(Sales)) as Total_Sales
```

```
from superstore
```

```
group by Category;
```

```
-- Group By
```

```
Select count(`Customer ID`), Country from superstore
```

```
group by Country;
```

```
select state , Count('Row ID') as total_Customers from superstore
```

```
group by State;
```

```
select Region, round(Sum(Profit)) as Total_Profit
```

```
from superstore
```

```
group by Region;
```



```
Select Category, `Sub-Category`, round(Sum(Sales)) as total_sales  
from Superstore  
group by Category, `Sub-Category`;
```

```
Select Region, Count(Distinct `Customer ID`) as total_customer  
from superstore  
group by Region;
```

```
Select Category, round(Sum(Sales)) as total_Sales  
from superstore  
where Region='West'  
Group By Category;
```

- 1.Total Profit for Each Sub Category
- 2.Total Orders for Each Ship Mode

```
select `Sub-Category`, sum(Profit) as total_profit from superstore  
group by `Sub-Category`;
```

```
select `Ship Mode`, Count(`Order ID`)as total_orders  
from superstore group by `Ship Mode`;
```

– Having

```
Select State, Count('Row ID') as Total_Customers  
from superstore  
group by State  
having Count('Row ID')>500;
```

```
Select State, round(Sum(Sales)) as Total_Sales  
from superstore group by State  
having Total_Sales>100000;
```

```
Select `Sub-Category`, Avg(Profit) as avg_profit  
from superstore  
group by `Sub-Category`  
having Avg(Profit)>50;
```

– Loss Making Sub-Categories

```
select `Sub-Category`, sum(Profit) AS total_profit  
from superstore
```

```
group by `Sub-Category`  
having SUM(profit) < 0;
```

```
Regions Avg(Sales) is less than 500  
select region, avg(sales) AS avg_sales  
from superstore  
group by region  
having avg(sales) < 500;
```

```
Show Sub-Categories where total_sales> 20000 and Profit> 10000  
Select `Sub-Category`,  
round(Sum(Sales))as total_Sales,  
round(sum(Profit))as total_Profit  
from superstore  
group by `Sub-Category`  
having total_Sales>20000 and total_Profit >10000;
```

```
–Order By  
Select `Order ID`, Customer_name, Sales from superstore  
Order by Sales;
```

```
Select `Order ID`, Customer_name, Sales from superstore  
Order by Sales Desc;
```

```
select `Order ID` , Region, Sales  
from superstore  
order by Region Asc, Sales Desc;
```

```
-- Order from Technology category sorted by Profit  
Select `Order ID` , Category, Profit  
from superstore  
Where Category='Technology'  
Order by Profit Desc;
```

```
-- Top 5 Highest Sales Orders  
Select `Order ID`, Customer_name, Sales  
from Superstore  
order by Sales Desc  
limit 5;
```

```
-- Region Wise total Sales(Highest First)
Select Region, round(Sum(Sales)) as total_Sales
from superstore
Group By Region
Order By total_Sales Desc;
```

```
-- Category wise Average Profit
Select Category, round(Avg(Profit),2) as Avg_Profit
from superstore
Group By Category
Order By Avg_Profit Desc;
```

```
-- Adding New Column
alter table superstore add revenue int;
update superstore set revenue=((Sales* Quantity)-Discount);
```

```
update superstore
set `Order Date`= str_to_date(`Order Date`, '%d-%m-%Y');
```

```
alter table superstore
Modify `Order Date` Date;
```

```
-- Top 5 States with Highest Sales
select state, Sales from superstore
order by sales desc limit 5
```

```
-- Year wise total Sales
Select Year(`Order Date`) as year, round(Sum(Sales),2) as total_sales
from superstore
group by Year(`Order Date`)
Order by year;
```

```
-- Total Sales and Profit in Each Region in Year 2016
select Region, sum(sales) as totalsales,
sum(profit) as totalprofit
FROM Superstore
WHERE YEAR(`Order Date`) =2016
```

GROUP BY Region;

– Which State is making overall loss??

```
select State , round(sum(Profit),2) AS total_profit
from superstore
group by State
having SUM(profit) < 0;
```

– State with highest orders?

```
Select State, Count(`Order Id`) as total_order
from superstore
group by state
order by total_order desc limit 5;
```

– Subcategories with Profit margin below 0
profit/sales

```
Select `Sub-Category`,
sum(profit)/Sum(sales) as profit_margin
from superstore
group by `Sub-Category`
having profit_margin<0;
```

– Year Wise Max and Min Sales

```
SELECT Year(`Order Date`) as year,
      MIN(Sales) AS Lowest_Sales,
      MAX(Sales) AS Highest_Sales
FROM superstore
GROUP BY Year(`Order Date`)
ORDER BY year;
```

– Show no of order placed in each year

```
SELECT YEAR(`Order Date`) AS year,
COUNT(*) AS total_orders
FROM superstore
GROUP BY YEAR(`Order Date`)
ORDER BY year;
```

– Which State has highest Profit(overall)?

```
select State ,sum(Profit)
from superstore group by State
```

order by sum(Profit) Desc limit 1;

– Top 3 Customer by total Sales in West Region

```
select Customer_name ,sum(Sales)
from superstore where Region='west'
group by Customer_name
order by sum(Sales) desc limit 3;
```

– Month wise sales for year 2016

```
Select monthname(`Order Date`) as month, round(Sum(Sales),2) as total_sales
from Superstore
where Year(`Order Date`)=2016
group by monthname(`Order Date`)
order by total_sales Desc;
```

– year wise total sales and Profit

–Top 5 Customer y Order Count

–Sub-Category with low average Sales

–Category wise Order volume

Data Preprocessing: It involves converting raw data into clean and Structed format

1. Improves the Data Quality
2. Better Insights and Analysis
3. Saves Time in the long run
4. Ensures Consistency
5. Standardizes the format— Male/male/M

Tableau Prep Builder

Tableau was founded in 2003

Tableau prep was introduced in 2018

Tableau Prep Builder in 2019

Operations in Tableau Prep

1. Take data from multiple sources
2. Cleans the Data
 - a. Data Type Conversion
 - b. Normalization
 - c. Data Renaming:
 - d. Remove the unwanted columns
3. Transform Data
 - a. Splitting Columns
 - b. Joining Data
 - c. Unioning Data
 - d. Aggregation
 - e. Pivot
 - f. Calculating New Fields
 - g. Sorting and Filtering
4. Handle Null Values

In Order Central:

Region is Missing, Discount has None, Separate Columns for Order and Ship for Date, Month and Year,

In West

State with abbreviated values

In East

Sales with Prefix of usd

Datatype of Sales Field.

Central

Calculation Field 1

Name: Region

Expression- "Central"

Calculation Field 2

Name: Order Date

Expression- MAKEDATE([Order Year],[Order Month],[Order Day])

Calculation Field 3

Name- Ship Date

Expression- MAKEDATE([Ship Year],[Ship Month],[Ship Day])

Remove Order Year, Order Month and Order Day

Remove Ship Year, Ship Month and Ship Day

Replace None in Discounts by 0

And Changed data type to number decimal

In West Region

Rename all states

AZ- Arizona

CA- California

CO- Colorado

ID-Idaho

MT-Montana

NM- New Mexico

NV-Nevada

OR-Oregon

Ut-Utah

WA-Washington

WY-Wyoming

East Region

Click on Sales> Clean> Remove Letters

Change Datatype to Decimal

Join 4 Regions using union operation and merge the mismatched fields.

In Returns Table

Remove-Row ID, Order Date, Sub-Category,Manufacturer, Product Name

Notes> Clean> Trim Spaces

Notes>Split Values>Automatic Split

Remove Notes

Rename Notes Split 1 Field- Approver Notes

Rename Notes Split 2 Field -Approver Name

Approver Name> Group Values> Common Characters

Approver Name> Clean> Trim Spaces

Since we want all order and only return data list hence select left join

Click on Clean Step

Remove Order ID 1

Remove Product ID 1

Create Calculation Field

Returned

IF ISNULL([Return Reason]) then 'No'

ELSE 'Yes' END

Days to Ship

DATEDIFF('day',[Order Date],[Ship Date])

Tableau:

Its an American Company Started in 2003 and was acquired by Salesforce in 2019

Tableau Desktop- Two Version

Professional Edition- 14 days free trail

Public Edition- free to use

<https://www.tableau.com/products/public/download>

Dataset:

https://docs.google.com/spreadsheets/d/1WKEDLhV4zeD73BfoYugrdT8r9GuQQVz9/edit?usp=drive_link&ouid=113247709954189786236&rtpof=true&sd=true

When we load a dataset to tableau, it automatically assigns a role and type to each field

Type- datatype to each field

Roles- Dimension and Measure

Qualitative Data- Qualities, Characteristics or Categories

- Non Numeric (Text, Label, Date)
- Used for grouping and Categorization
- Does not involve Calculations

- Focus on Description

Quantitative Data- Numbers

- Numeric Values
- Can be Calculated and Aggregated
- Used for Comparison and trends
- Focuses on Measurement

Dimension - Qualitative Fields that describes the data

- Displayed as Blue pills

Measure- Quantitative Fields

- Displayed as Green Pills

Business Question:

Sales by Categories

Sales by Sub Categories

Profit by Region

Staked Bar chart-Sales across segments by categories

C- Sales

R- Segment

Col- Category

Filter- Year, Country

Vertical Bar Chart-Avg Sales Across Different Sub-Categories by Region

C- Sub-Categories

R- Average Sales

Col- Categories

Filter- Region and Year

Side by Side bar chart- Orders by Categories by Shipmode

C- Ship Mode, Category

R- Count Order Id

Col- Category

Line Chart- Monthly Sales Trend over Years

C- Month

R- Sales
Col- Sale
Filter- Year

Area Chart- Month Wise Profit across Segments
C- Month
R- Profit
Col- Segments
Filter- Year and Region

Histogram- Quantity by Segments
https://help.tableau.com/current/pro/desktop/en-us/buildexamples_histogram.htm

Box Plot- Segment wise Discount across region
https://help.tableau.com/current/pro/desktop/en-us/buildexamples_boxplot.htm

Packed Bubble - Revenue By Sub Category

Create New Field: Revenue
([Sales]*[Quantity])-[Discount]

C- Sub Category
R- Revenue
Show me- Select packed Bubble
Col- Category

Pie Chart- Region wise Profit
C- Region
R- Profit

Bullet Chart
Total Sales= Sales * Quantity
Sales Target= Sum(Total Sales)* 1.2
Target is 20% higher than actual sales

C-Revenue
Detail Card- Sales Target
R- Sub-Categories
Col-Categories

Scatter Plot- Sales and Profit by Sub -Category

C- Total Sales

R- Profit

Col- Category

Detail- Sub-Category

Tree Map

C- State

R- Revenue

Word Cloud

Text- Sub Categories

Size- Profit

Drop Down Select Text

Col- Category

Heat Map

C- Segment

R-Sub-Category

Size- Revenue

Col- Profit

Text Tables:(Crosstab)

Simple Tabular visualization where numbers and text values are displayed in rows and columns

C- Year

R- Category, Sub-Category

Text- Revenue

Highlight Table:

Combines Text table with color encoding

C- Quarter

R-Sub-Category

Text-Profit

Symbol Map
Sales
State

Filled Map
Profit
State

https://help.tableau.com/current/pro/desktop/en-gb/datasource_datamodel.htm

Different Kind of layers

1. Data Source Layer: The stage where you connect to data
2. Logical Layer: How tables are related logically, not physically merged.
 - Introduced in 2020.2
 - Uses Relationship
 - Tables remain Separate
 - Tableau decides how to combine data at query time.
3. Physical Layer: how tables are physically combined
 - Uses Join
 - Data is merged row by row
 - Similar to SQL Joins
4. Visualization Layer: Build Charts and Dashboards

JOINS (Traditional Method)

https://help.tableau.com/current/pro/desktop/en-us/joining_tables.htm

- It is way to physically combine two or more tables into one table based on a common field before analysis starts
- Created in Physical Layer
- When Connecting two tables at Physical layer through joins You need to choose type of join:
 - Inner
 - Left
 - Right
 - Full

Problems:

- Joins happens before aggregation
- If there is any record which matches multiple rows , data gets multiplied
- Joins repeat measure when one row matched multiples rows in another table this is called row multiplication

Relationship: It is logical connection between tables that tells tableau how tables are related without physically merging them

Cardinality- It defines relationships between tables like how many records in one table match records in another table.

One to One

One to Many

Many to Many

Aggregate the data before combining

Advantages

- Keeps table separate
- Aggregate measures correctly
- Combines only the needed result

Hide and Unhide Fields

Rename

Hierarchy

Principal/Director>Dean>Hods>Professors>Assistant Professors>Clerks>Peon

Year>Quarter>Months>Days

Category> Sub-Category>Product Name

Country> State> City> Postal Code

Folders

Groups

Splitting data

Unions

Data Blending

It is used to combine data from multiple data-sources at **the worksheet level**

- Data is not physically joined
- Tableau blends aggregated results

Primary Data Source denoted with Blue color

Secondary Data Source denoted with Orange Color

Cross Database Join

It allows to join the tables from different databases or filetypes at the data source level

Calculation Fields: It is Custom formula that you can create to generate new data from the existing data fields

1. Profit Ratio
[Profit/Sales]
2. Sales with Tax(assume 10% Tax)
Sales *1.10
3. Discount Category
IF [Discount]=0 THEN "No Discount"
ELSEIF [Discount]<=0.2 THEN "Low Discount"
ELSE "High Discount"
END
4. Sales Category
High Revenue and Low Revenue
Revenue>5000
if [Revenue] > 5000 Then "High Revenue"
else " Low Revenue"
End
5. Order Size
Small order <2
Medium Order <5
Large Order
If [Quantity]<=2 then "Low Order"
ELSEIF [Quantity]<=5 then "Medium Order"

- ELSE "Large Order"
- End
- 6. Profit margin
Profit/Sales *100
- 7. Days to Ship
DATEDIFF('day',[Order Date],[Ship Date])
- 8. Shipping Delay Category
Less than 2 day " On Time", "Delayed"
IF [Days_to_Ship]<=2 THEN "ON-TIME"
ELSE "DELAYED"
END
- 9. Order Month Name
DATENAME('month',[Order Date])
- 10. Category- Sub Category Combination
[Category] + "-" + [Sub-Category]
- 11. Rank
Rank(Sum{Profit})

Null Values - Missing values

1. Filter out the Null Values
2. To Check Null Values
IsNull([Sales])
3. Replace the Null Values with Default Values
IFNULL([Sales],0)
IFNULL([Category],"Unknown")
4. Handling Null Values using ZN()
ZN(Profit)

Sets

They are used to create subset of data based on certain conditions or Criteria

It can be either IN Set or OUT Set

1. Static: Create manually by selecting members
2. Dynamic : Members are selected using conditions
3. Combined Set

Filters:

1. Extract Filter (Tableau Desktop Professional Edition)
2. Data Source Filter
3. Context Filter
4. Dimension Filter
5. Measure Filter

Dataset...Billions of record

Extract subset of Data from the original Data (.hyper)

Order Id, C Name, State, Region, Sales, Profit

Region- North, South, East , and West

Extract of Data with only East Region

Parameters:

1. Top N and Bottom N
2. Date Field Parameter
3. Dynamic Dimension
4. Dynamic Measure
5. Ref Line Parameter
6. Dynamic Dimension and Dynamic Measure

1.Parameter - DateParameter

Calculation field

Year, Quarter, Month, Week, Day

Create Parameter



Name

DateParameter

Properties

Data type

String

Display format

Year

Current value

Year

Value when workbook opens

Current value

Allowable values

☐ All ☒ List ☐ Range

Value	Display As	
Year	Year	
Quarter	Quarter	
Month	Month	
Week	Week	
Day	Day	

☒ Fixed

☐ When workbook opens

Add values from ▼

Remove Selected

```

CASE [DateParameter]
WHEN 'Year' THEN STR(YEAR([Order Date]))
WHEN 'Quarter' THEN 'Q' + STR(DATEPART('quarter', [Order Date]))
WHEN 'Month' THEN DATENAME('month', [Order Date])
WHEN 'Week' THEN STR(DATEPART('week', [Order Date]))
WHEN 'Day' THEN STR(DATEPART('day', [Order Date]))
END
    
```

Dynamic Dimension

Edit Parameter [Dimension Parameter]

Name

Dimension Parameter

Properties

Data type

String

Display format

Category

Current value

Category

Value when workbook opens

Current value

Allowable values

All

List

Range

Value	Display As	
Category	Category	
Sub-Category	Sub-Category	
Segment	Segment	
Click to add		

Fixed

When workbook opens

Add values from

Remove Selected

Cancel

OK

CASE [Dimension Parameter]
when "Category" then [Category]

```
when "Sub-Category" Then [Sub-Category]
when "Segment" then [Segment]
End
```

Dynamic Measure

Edit Parameter [MeasureParameter] ×

Name

MeasureParameter

Properties

Data type

String

▼

Display format

Sales

▼

Current value

Sales

▼

Value when workbook opens

Current value

▼

Allowable values

☐ All

☒ List

☐ Range

Value	Display As	
Sales	Sales	
Profit	Profit	
Discount	Discount	
Quantity	Quantity	
Click to add		

☒ Fixed

☐ When workbook opens

Add values from ▼

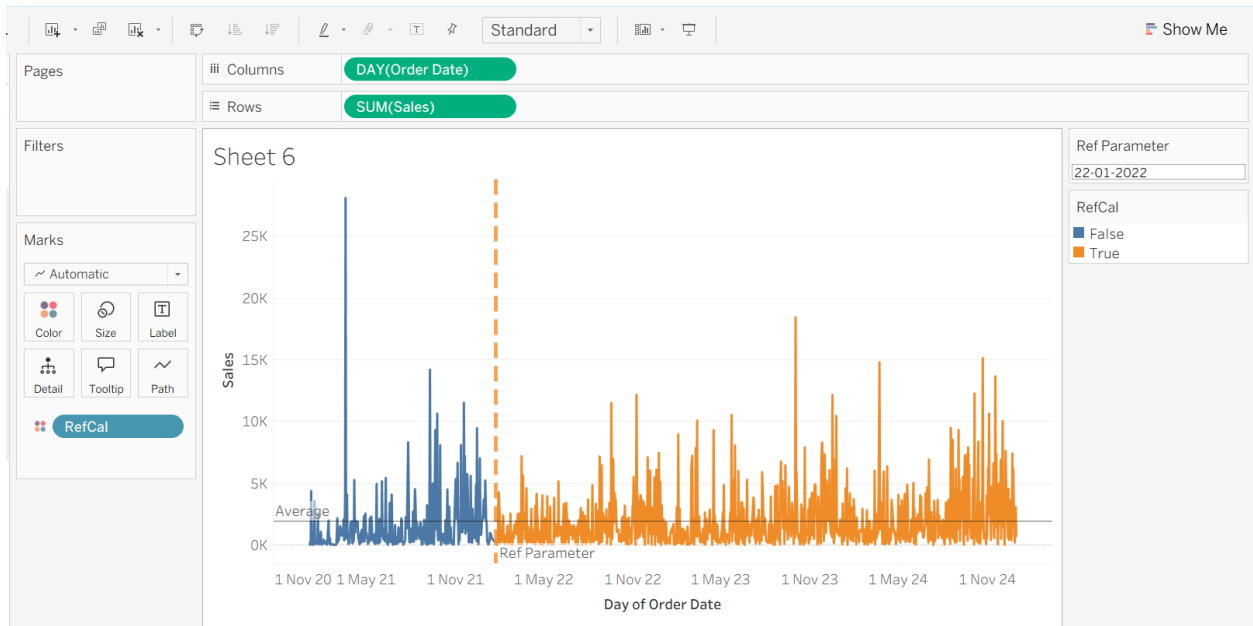
Remove Selected

Cancel

OK

```
Case[MeasureParameter]
WHEN "Sales" then [Sales]
When "Profit" then [Profit]
When "Discount" then [Discount]
When "Quantity" then [Quantity]
End
```

Reference Parameter



Sets with parameter

Edit Set [Sub-Category Set]

Name: Sub-Category Set

General Condition Top

☐ None

☒ By field:

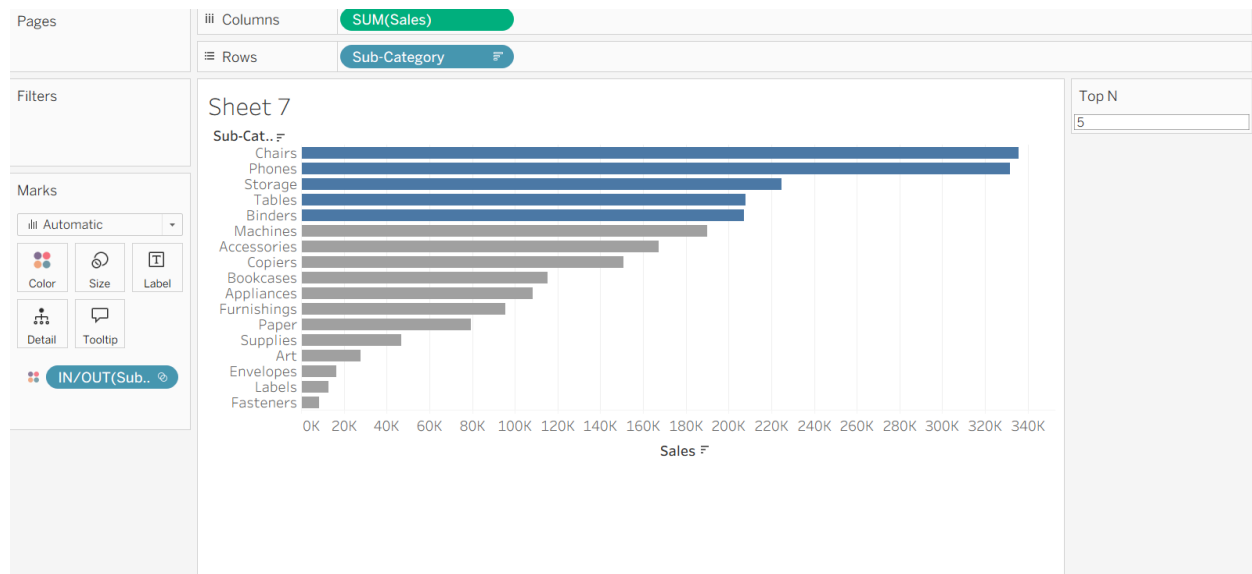
Top Top N by

Sales Sum

☐ By formula:

Top 10 by

Reset OK Cancel Apply



Quick Table Calculation:

1. Running Total: Computes the Cumulative sum of measure over the dimension
2. Percent of Total: Calculates the percentage contribution of each datapoint to the total
3. Percent Difference: Calculates the percentage difference between two consecutive datapoints
4. Difference: Calculates the difference between two consecutive datapoints
5. Moving Average: Calculates the Average of Measure over a moving window of data points
6. Percentile: Calculates the value if specified percentile for a given measure
1,2,3,4,5,6,7,8,9,10,12,14,15
25 percentile=5
7. Rank: Ranks each datapoint within its partition based on the measure values
8. YTD total(Year to Date) - Calculates the cumulative sum of measure from the beginning of the year up to current datapoint
9. YTD Growth- Calculates Year to date growth rate of measure from the beginning of the year up to current datapoint
10. Year Over Year Growth Rate: Calculates percentage change in measure compared to the same period in the previous Year
11. Compound Growth rate: Calculates the annual growth rate of any business over a specified period of time.

Advanced Visualization:

Donut Chart

Lollipop Chart

Pareto Chart

Pareto Principle: 80% of the results comes from 20% of the cause

Funnel Chart

Waterfall Chart

Butterfly Chart

Granularity??? Level of detail in your data

High Granularity-More Detail

Low Granularity-Less detail

Country> State>City

Country- Sale—Less Granularity

City—More Detail

Normally in Tableau Calculation based on what is put in the view

LOD- Level of Detail Expression- It controls the level at which tableau performs a calculation independent of the view (rows, columns, filters)

1. What level you need
2. What is Aggregation- Sum, Avg, Count, Max

Three Main types of LOD Expressions

1. Fixed LOD
2. Exclude LOD
3. Include LOD

Syntax:

{ Type [Dim] : Aggregation }

1. Fixed LOD

Calculates the Data at a specified level, ignoring the dimension in the view

{ Fixed [Dim] : Aggregation }

2. Exclude LOD

Calculates the data at specific level by excluding certain dimension from the view calculation

{ Exclude[Dim] : Aggregation }

3. Include LOD

Adds extra dimension to existing view before aggregation

{ Include [Dim] : Aggregation }

Map-

Filled Map by double click on states and selecting the filled map

Country

Region

State

Total Sales { Fixed :Sum(Sales)}

State Sales { Fixed[State] :Sum(Sales)}

Regional Sales { Exclude[State/Province],[Country/Region]:SUM([Sales])}

Average Sales

Flask

Flask is a python Web Framework that helps in:

- Build Web application

- Create Apis

- Show Data, Dashboards, integrate with ML Models

Flask contains by default:

- Routing

- HTTP request and response handling

- Jinja2 Template Engine

- Development Server

Flask is built on 2 powerful libraries

1. Werkzeug (WSGI)- Web Server Gateway Interface)

- a. Handling HTTP request and responses

- b. URL routing

- c. Error handling

- d. Debugging tool

2. Jinja2(Template Engine)

- a. Renders dynamic Web Pages

- b. Embeds Python Variables into HTML

Steps Involved in Building the application

1. Import the required libraries
2. Initialize the flask application
 - a. Ex:
`app=Flask(__name__)`
3. Routing and rendering HTML pages
 - a. Define the routes using flask decoratos, these roiutes decides which HTML pages should be shown to the user
Ex:
`@app.route("/")`
`def home()`
`return "Welcome"`
`render_template("index.html")`
4. Rendering Tableau Dashboard and Stories
5. `__name__ == __main__`
6. Default port in Flask
 - a. Port 5000 by default
 - b. `http://localhost:5000`

```
from flask import Flask
```

```
app= Flask(__name__)
```

```
@app.route('/')
```

```
def home():
```

```
    return "Welcome to Data Analytics with Flask"
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

`Flask(__name__)` — Create Flask app

`@app.route('/')` - URL mapping

`def home():` – Function for Page

Return - What browser displays

`debug=True`- Auto reload + error help


```
from flask import Flask
```

Importing Flask from Flask Library

```
app= Flask(__name__)
```

Create Flask Application Object

```
@app.route('/')
```

Flask Decorator, when the user opens home page(/) and it runs function written below it

```
def home():
```

```
    return "Welcome to Data Analytics with Flask"
```

This function runs when the home page is accessed

It returns a simple message that appears in the browser

When you open <http://localhost:5000> it will display- "Welcome to Data Analytics with Flask"

```
if __name__ == '__main__':
```

It check whether the python file is being run directly

```
app.run(debug=True)
```

It starts the Flask Web Server

Main Folder

- templates

 - Html

- static

 - Css, Js, Img

- [app.py](#)

Visit

<https://bootstrapmade.com/>

Download a free template

Create folder with sub folders templates, static and [app.py](#)

Copy the index.html file from the downloaded template and store in templates folder

Copy all sub folders in assets folder and paste in static folder

In Index.html replace all assets keyword with static

Make all the necessary changes in index.html wrt your project

Add the embed link of Dashboard and Story from tableau public in one of the section in index.html

Project Folder

static - css, js, img,scss, vendor

Templates- index.html

App.py file-

```
from flask import Flask, render_template
app= Flask(__name__)

@app.route('/')
def home():
    return render_template("index.html")
if __name__=='__main__':
    app.run(debug=True)
```

Understand Your Use Case

Collect the dataset

Load the dataset to Tableau Desktop

Perform Data Preprocessing in Tableau Desktop

Create 8-10 business Question and Generate Visualization

Develop Dashboard

Develop Story

Publish Dashboard and Story on tableau Public

Use the Embed code of Dashboard and Story from Tableau Public and Integrate with Web Page using Flask

Complete all the templates

Follow the given folder structure and upload the files on Github for final submission

Templates:

1 Ideation Phase

- Problem Statement
- Empathy Map
- Brain Storming

2. Requirement Analysis

- Customer Journey Map

- Solution Requirement
- Data Flow Diagram
- Technology Stack

3. Project Design Phase

- Problem Solution Fit
- Proposed Solution
- Solution Architecture

4. Project Planning Phase

- Project Planning Template

5. Project Development Phase

- 1st PDF-Preprocessing Steps and Business Questions with visualisation
- 2nd PDF -Dashboard Screenshot, Story Screenshot along and Web Page Screenshot
- Flask Folder - static, templates, app.py
- Dataset

6. Performance Testing

- Performance Testing template

7.Doc and Demo

- Final report
- Demonstration Video

