

**[PBIP-5] [Data Professional Survey Breakdown](#)** Created: 09/Jan/24 Updated: 09/Jan/24

<b>Status:</b>	In Progress
<b>Project:</b>	<a href="#">Power BI Projects</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Unresolved	<b>Votes:</b>	0
<b>Labels:</b>	BI_Project		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i00053:
<b>Sprint:</b>	PBIP Sprint 1

<b>Comments</b>
Comment by <a href="#">Manjesh Verma</a> [ 09/Jan/24 ]
<a href="#">Manjesh-Verma-13/Manjesh Data_Professional_BI.pbix at BI_Projects · volcanoacademy/Manjesh-Verma-13 (github.com)</a>

<b>[PBIP-4] <a href="#">Job Posting Analysis</a></b> Created: 09/Jan/24 Updated: 09/Jan/24 Resolved: 09/Jan/24	
<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Power BI Projects</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	BI_Project		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0004v:
<b>Sprint:</b>	PBIP Sprint 1

<b>Description</b>
<p><b>Project : Job Posting Data Analysis</b></p> <p><b>Step 1: Prepare</b></p> <p>In this step, I will create the project in Power BI, present test cases and client information. This process is designed for global implementation.</p> <p><b>Step 2: Process</b></p> <p>2.1 Import the Dataset</p> <p>2.3 Import in Power Bi</p> <p>Import the CSV file and analyze the dataset.</p> <p>2.3 Cleaning Steps</p> <p>Here, I commence the cleaning process for job posting data. The initial steps include:</p>

Using the first row as the header.

Renaming the default column index as 'Unnamed.'

Converting the data types of each column; numerical values default to decimals, and dates are adjusted to their local zones, such as US and UK.

Since there is a Job Title column in this dataset of job skills, I combined the column.

Following these steps, I close and apply the transformed dataset, where I proceed to conduct insightful analysis.

I Provide the GitHub link containing the Power BI report, project report, and presentation of the Zomato.

**Step:** I created an app for it and have included a link and barcode as well.

## Comments

Comment by [Manjesh Verma](#) [ 09/Jan/24 ]

[Manjesh-Verma-13/Manjesh Job Posting Analysis.pbix at BI Projects · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#) I made this project, and this is my GitHub link

<b>[PBIP-3] <a href="#">Zomato Data Analysis</a></b> Created: 09/Jan/24 Updated: 09/Jan/24 Resolved: 09/Jan/24	
<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Power BI Projects</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	BI_Project		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0004n:
<b>Sprint:</b>	PBIP Sprint 1

<b>Description</b>
<p><b>Project : Zomato Data Analysis</b></p> <p><b>Step 1: Prepare</b></p> <p>In this step, I will create the project in Power BI, present test cases and client information. This process is designed for global implementation.</p> <p><b>Step 2: Process</b></p> <p>2.1 Import the Dataset</p> <p>2.3 Import in Power Bi</p> <p>Import the CSV file and analyze the dataset.</p> <p>2.3 Cleaning Steps</p> <p>Here, I commence the cleaning process for Zomato data. The initial steps include:</p>

Using the first row as the header.

Renaming the default column index as 'Unnamed.'

Match the data types of both the two table of country columns behalf of this it maintains relationship between the tables.

Converting the data types of each column; numerical values default to decimals, and dates are adjusted to their local zones, such as US and UK.

For this Zomato dataset, Since there are mixed cuisines in the dataset (North and South Indian), I divided the Cuisines column more than eight times.

Following these steps, I close and apply the transformed dataset, where I proceed to conduct insightful analysis.

### **Step 3: Analyze**

For the analysis part, we will extract the most important components of our data to answer our business objectives.

Let's load our data into Power BI and analyze the following:

-Homepage Navigation:

Home

Insights

Top Restaurants

Cost & Rating

Restaurant Selection

-Insights Page:

### **Utilizing a slicer for country search, this page showcases:**

Total restaurants globally

Total cuisines worldwide

Total votes received

Total cities covered

Global locations

Restaurants sorted by rating

- Top Restaurants:

Top restaurants categorized by cuisines

Leading cuisines in specific locations

Distribution of cuisines

Most popular cuisines  
Restaurant counts by locality

- Cost & Rating:

Distribution of costs based on votes  
Rating distribution related to table bookings  
Cuisines categorized by price range  
Rating distributions influenced by online orders.

- Restaurant Selection:

#Options for country and city selection, with details on:

Availability of online delivery  
Table booking options  
Total votes received  
Price range  
Comprehensive restaurant details, including name, locality, aggregate rating, rating text, and average cost for two.

I Provide the GitHub link containing the Power BI report, project report, and presentation of the Zomato.

**Step 6:** I created an app for it and have included a link and barcode as well.

#### Comments

Comment by [Manjesh Verma](#) [ 09/Jan/24 ]

[Manjesh-Verma-13/Manjesh\\_Zomato.pbix at BI Projects · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#) I made this project, and this is my GitHub link.

**[PBIP-2] [H1B Visa Data Analysis Project - Power BI](#)** Created: 06/Dec/23 Updated: 07/Dec/23 Resolved: 07/Dec/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Power BI Projects</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	BI_Project		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0004f:
<b>Sprint:</b>	PBIP Sprint 1

#### Description

### Project : H1B Visa Data Analysis

#### Step 1: Prepare

In this step, I will create the project in Power BI or SQL and present test cases and client information. This process is designed for global implementation.

#### Step 2: Process

##### 2.1 Import the Dataset

##### 2.3 Import in SQL

Import the CSV file into SQL and analyze the dataset.

##### 2.3 Cleaning Steps

Here, I commence the cleaning process for H1B visa data. The initial steps include:

1. Using the first row as the header.

2. Renaming the default column index as 'Unnamed.'
3. Changing 'Employer Name' to 'Company Name.'
4. Removing duplicate and blank rows.
5. Checking for null and blank values and deleting them if they exist.
6. Converting the data types of each column; numerical values default to decimals, and dates are adjusted to their local zones, such as US and UK.
7. For this H1B dataset, I created a duplicate column for 'Worksite' and subsequently split the states and cities.

Following these steps, I close and apply the transformed dataset, where I proceed to conduct insightful analysis.

### Step 3: Analyze

For the analysis part, we will extract the most important components of our data to answer our business objectives.

Let's load our data into MySQL or Power BI and analyze the following:

1. **Top 10 Hiring Companies:**
2. **Distribution of Prevailing Wages:**
3. **Average Prevailing Wage by Employer:**
4. **Top 10 Companies Offering Highest Salaries:**
5. **Geographical Analysis:**
6. **Top 10 Worksite Locations:**
7. **Case Status Distribution:**
8. **Petitions by Year:**
9. **Denied Petitions Over the Years:**
10. **Occupation Analysis:**
11. **Visa Application Trends Over the Years:**
12. **Top 10 States with Highest Application Counts:**
13. **Distribution of Applications by Job Titles:**
14. **Top 10 States with Highest Denied Petitions:**
15. **Analysis of Full-Time vs. Part-Time Positions:**
16. **Top 10 Job Titles with Highest Prevailing Wages**

I Provide the GitHub link containing the Power BI report, SQL file, project report, and presentation of the Superstore project.

### Step 6: Act

The project taught us many innovative things to work on. In the H1B visa analysis project involved an in-depth exploration of various facets within the dataset. The investigation focused on identifying the top 10 companies facilitating visa applications, understanding the distribution of approved petitions versus denied ones, analysing job titles associated with these applications,



and pinpointing the prime worksite locations where employees engage in both part-time and full-time roles across America.

Additionally, scrutiny was applied to discern the certification status of visas, tracking the number of applications over the years and quarters. This comprehensive analysis unearthed meaningful insights that not only contribute to understanding the dynamics of H1B visa applications but also provide valuable information for companies aiming to enhance their application processes.

The findings serve as a guide for improving visa application forms and expanding opportunities for employees to realize their career aspirations in dream companies. The project underscores the importance of continuous analysis and improvement in facilitating the visa application process, thereby fostering a more seamless experience for both employers and applicant.

#### Comments

Comment by [Manjesh Verma](#) [ 07/Dec/23 ]

[volcanusacademy/Manjesh-Verma-13 at BI Projects \(github.com\)](#)

I have completed H1B visa projects, and here is the GitHub link for you to review

**[PBIP-1] [Project 1: Superstore Sales Analysis](#)** Created: 06/Dec/23 Updated: 07/Dec/23 Resolved: 07/Dec/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Power BI Projects</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	BI_Project		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i00047:
<b>Sprint:</b>	PBIP Sprint 1

### Description

## Project 1: Superstore Sales Analysis

### Step 1: Prepare

In this step, I will create the project in Power BI or SQL and present test cases and client information. This process is designed for global implementation.

### Step 2: Process

#### 2.1 Import the Dataset

Initially, we identify the type of data we are working with, such as Excel data, SQL Server data, web data, or Power BI datasets. Once determined, we proceed to load and transform the data, initiating the cleaning process as the first step.

#### 2.2 Cleaning Steps

Here, I commence the cleaning process for H1B visa data. The initial steps include:

Removing duplicate and blank rows.

Checking for null and blank values and deleting them if they exist.

Converting the data types of each column; numerical values to decimals & fixed decimals, and dates are adjusted to their local zones, such as US and UK.

Following these steps, I close and apply the transformed dataset, where I proceed to conduct insightful analysis.

### **2.3 Import in SQL**

Import the CSV file into SQL and analyze the dataset.

#### **Step 3: Analyze**

For the analysis part, we will extract the most important components of our data to answer our business objectives.

Let's load our data into MySQL or Power BI and analyze the following:

**Find total sales and total profits of each year.**

**Find total profits and total sales per quarter.**

**Identify the region that generates the highest sales and profits.**

**Determine which state and city bring in the highest sales and profits.**

**Explore the relationship between discount and sales and calculate the total \*discount per category.**

**Identify the category that generates the highest sales and profits in each region and state.**

**List the names of the products that are the most and least profitable.**

**Determine which segment makes the most profits and sales.**

**Calculate the total number of unique customer IDs and their distribution per region and state.**

**Calculate the Customer Rewards Program.**

#### **Step 5: Share**

I Provide the GitHub link containing the Power BI report, SQL file, project report, and presentation of the Superstore project.

## **Step 6: Act**

### **Trends in Total Sales and Profits:**

Identify trends in total sales and profits over the years to understand overall business performance. Finally, observe the results to identify the sum of total profits and the sum of total profit by year.

### **Quarterly Performance:**

Analyze total profits and sales per quarter to identify any seasonality or specific periods of high/low performance. Finally, observe the results to identify the sum of total profits and the sum of total profit per quarter.

### **Regional and State Analysis:**

Identify the sum of sales and the sum of profits by region. This analysis provides insights into regional and state-wise performance, helping in strategic decision-making and resource allocation.

### **Discount and Sales Relationship:**

Identify the relationship between discount and sales and the total discount per category. This analysis contributes to optimizing discount strategies for different product categories.

### **Category Performance:**

Analyze the names of the most and least profitable products. This insight aids in optimizing the product portfolio by understanding which products contribute significantly to profits and which ones may need further evaluation.

### **Product Profitability:**

Finally, observe the results to identify the segment that contributes most to our sales and profit. This analysis provides a focus on the product segments driving the majority of the business's sales and profits.

### **Customer Metrics:**

Seek to identify the unique customer IDs where the majority of unique customer IDs are per states and regions. This customer-centric analysis assists in tailoring marketing efforts and enhancing customer relationship management strategies.

## Comments

Comment by [Manjesh Verma](#) [ 07/Dec/23 ]

[volcanusacademy/Manjesh-Verma-13 at BI\\_Projects \(github.com\)](#)

I have completed Superstore sales projects, and here is the GitHub link for you to review

**[DAP-15] [SQL queries practice on customers, orders, and shipping tables](#)** Created:

20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i00033:
<b>Sprint:</b>	DAP Sprint 1

#### Description

## SQL queries practice on customers, orders, and shipping tables :

Please find the attched document and complete the assignment.

Desc-- Here I have completed the 60 queries of the customers, orders and shippings tables

#### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[Manjesh-Verma-13/Qu\\_customer\\_Order\\_Shippings.sql at Mysql\\_DB · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#)

I have completed the 60 queries of this practice assignment.

**[DAP-14] Create a database named 'Entertainment' and design three tables with the provided table structures.** Created: 20/Nov/23 Updated: 06/Dec/23 Resolved: 06/Dec/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0003z:
<b>Sprint:</b>	DAP Sprint 1

**Description**

**Create a database named 'Entertainment' and design three tables with the provided table structures. :**

Create a database named 'Entertainment' and design three tables with the provided table structures. Subsequently, write SQL queries to address the following questions using the 'Entertainment' database.

Table1: SciFiWebSeries

```
CREATE TABLE SciFiWebSeries (  
  series_id INT PRIMARY KEY,  
  title VARCHAR(255),  
  release_year INT,  
  genre VARCHAR(255),  
  average_rating DECIMAL(3, 1),  
  seasons INT,  
  episodes INT,  
  main_actors TEXT,  
  language VARCHAR(50),  
  platform VARCHAR(50),
```

```

viewership VARCHAR(50),
production_company VARCHAR(255),
director VARCHAR(255),
awards_won INT,
awards_nominated INT,
creator VARCHAR(255),
streaming_status VARCHAR(50),
description TEXT
);
INSERT INTO SciFiWebSeries
VALUES
(1, "Black Mirror", 2011, "Drama, Sci-Fi", 8.8, 6, 22, "Bryce Dallas Howard, Jon Hamm",
"English", "Netflix", "25 million", "Zeppotron, House of Tomorrow", "Various", 14, 58,
"Charlie Brooker", "Available", "A series of stand-alone dramas - sharp, suspenseful, satirical
tales that explore techno-paranoia."),
(2, "Stranger Things", 2016, "Drama, Horror, Sci-Fi", 8.7, 4, 34, "Millie Bobby Brown, Winona
Ryder, David Harbour", "English", "Netflix", "64 million", "21 Laps Entertainment, 360, Duffer
Brothers Productions", "Duffer Brothers", 7, 43, "Matt Duffer, Ross Duffer", "Available", "In a
small town where everyone knows everyone, a group of kids discovers a strange girl and a
secret experiment."),
(3, "Westworld", 2016, "Drama, Sci-Fi, Western", 8.6, 3, 28, "Evan Rachel Wood, Thandie
Newton, Jeffrey Wright", "English", "HBO", "13 million", "Kilter Films, Bad Robot
Productions, Warner Bros. Television", "Various", 9, 46, "Jonathan Nolan, Lisa Joy",
"Available", "In a futuristic theme park, guests experience the stories of android hosts with
artificial consciousness."),
(4, "The Expanse", 2015, "Drama, Mystery, Sci-Fi", 8.5, 6, 60, "Thomas Jane, Steven Strait,
Shohreh Aghdashloo", "English", "Amazon Prime", "5 million", "Alcon Television Group,
Sean Daniel Company", "Various", 2, 20, "Mark Fergus, Hawk Ostby", "Available", "Two
detectives and a ship's officer must work together to expose a conspiracy that threatens the
Solar System's fragile state."),
(5, "The OA", 2016, "Drama, Fantasy, Mystery, Sci-Fi", 7.9, 2, 16, "Brit Marling, Jason Isaacs,
Emory Cohen", "English", "Netflix", "10 million", "Plan B Entertainment, Anonymous Content,
Netflix", "Zal Batmanglij", 1, 3, "Brit Marling, Zal Batmanglij", "Available", "A young blind
woman returns home, her sight restored, with strange abilities and a mysterious mission."),
(6, "The Mandalorian", 2019, "Action, Adventure, Fantasy, Sci-Fi", 8.7, 2, 16, "Pedro Pascal,
Gina Carano, Carl Weathers", "English", "Disney+", "30 million", "Lucasfilm, Golem
Creations", "Various", 11, 39, "Jon Favreau", "Available", "A lone bounty hunter in the outer
reaches of the galaxy navigates his way through the New Republic's turmoil."),
(7, "Dark", 2017, "Crime, Drama, Mystery, Sci-Fi", 8.8, 3, 26, "Louis Hofmann, Lisa Vicari,
Oliver Masucci", "German", "Netflix", "4 million", "Wiedemann & Berg Television", "Baran
bo Odar, Jantje Friese", 4, 13, "Baran bo Odar, Jantje Friese", "Available", "A missing child
leads four estranged families on a frantic hunt for answers as they unearth a mind-bending time
travel conspiracy."),
(8, "Altered Carbon", 2018, "Action, Drama, Sci-Fi", 8.0, 2, 18, "Joel Kinnaman, James
Purefoy, Martha Higareda", "English", "Netflix", "20 million", "Mythology Entertainment,
Skydance Television, Netflix", "Various", 1, 8, "Laeta Kalogridis", "Available", "In a future

```



where consciousness can be transferred to different bodies, a soldier seeks the truth about his new existence."),  
 (9, "The Witcher", 2019, "Action, Adventure, Drama, Fantasy, Mystery, Sci-Fi", 8.2, 2, 16, "Henry Cavill, Anya Chalotra, Freya Allan", "English", "Netflix", "76 million", "Platige Image, Sean Daniel Company, Netflix", "Various", 5, 30, "Lauren Schmidt Hissrich", "Available", "Geralt of Rivia, a solitary monster hunter, navigates a world filled with magic and dangerous creatures."),  
 (10, "Counterpart", 2017, "Drama, Sci-Fi, Thriller", 8.1, 2, 20, "J.K. Simmons, Harry Lloyd, Nazanin Boniadi", "English", "Starz", "2 million", "Gilbert Films, Anonymous Content, Media Rights Capital", "Various", 2, 8, "Justin Marks", "Available", "A UN employee discovers the agency he works for is hiding a gateway to a parallel dimension.");

Table2: ActorsInSciFiWebSeries

```
CREATE TABLE ActorsInSciFiWebSeries (
  actor_id INT PRIMARY KEY,
  first_name VARCHAR(50),
  last_name VARCHAR(50),
  birthdate DATE,
  gender VARCHAR(10),
  nationality VARCHAR(50),
  total_appearances INT,
  series_id INT,
  FOREIGN KEY (series_id) REFERENCES SciFiWebSeries(series_id)
);
INSERT INTO ActorsInSciFiWebSeries (actor_id, first_name, last_name, birthdate, gender,
nationality, total_appearances, series_id)
VALUES
(1, 'Bryce', 'Dallas Howard', '1981-03-02', 'Female', 'American', 1, 1),
(2, 'Millie', 'Bobby Brown', '2004-02-19', 'Female', 'British', 1, 2),
(3, 'Evan', 'Rachel Wood', '1987-09-07', 'Female', 'American', 2, 3),
(4, 'Thomas', 'Jane', '1949-02-22', 'Male', 'American', 2, 4),
(5, 'Brit', 'Marling', '1982-08-07', 'Female', 'American', 2, 5),
(6, 'Pedro', 'Pascal', '1975-04-02', 'Male', 'Chilean-American', 1, 6),
(7, 'Louis', 'Hofmann', '1997-06-03', 'Male', 'German', 1, 7),
(8, 'Joel', 'Kinnaman', '1979-11-25', 'Male', 'Swedish-American', 1, 8),
(9, 'Henry', 'Cavill', '1983-05-05', 'Male', 'British', 1, 9),
(10, 'J.K.', 'Simmons', '1955-01-09', 'Male', 'American', 1, 10);
```

Table3: RolesInSciFiWebSeries

```
CREATE TABLE RolesInSciFiWebSeries (
  series_id INT,
  actor_id INT,
  role VARCHAR(50),
  FOREIGN KEY (series_id) REFERENCES SciFiWebSeries(series_id),
```

```
FOREIGN KEY (actor_id) REFERENCES ActorsInSciFiWebSeries(actor_id)
);
```

```
INSERT INTO RolesInSciFiWebSeries (series_id, actor_id, role)
VALUES
(1, 1, 'Lacie Pound'),
(2, 2, 'Eleven'),
(3, 3, 'Dolores Abernathy'),
(4, 4, 'Detective Joe Miller'),
(5, 5, 'Prairie Johnson'),
(6, 6, 'The Mandalorian'),
(7, 7, 'Jonas Kahnwald'),
(8, 8, 'Takeshi Kovacs'),
(9, 9, 'Geralt of Rivia'),
(10, 10, 'Howard Silk');
```

#### Questions on Table 1: Sci-Fi Web Series

What are the columns in the "Sci-Fi Web Series" table?

Can you provide the titles of all the sci-fi web series released before 2018?

Which sci-fi web series has the highest average rating?

How many seasons does "Stranger Things" have?

What is the viewership of "The Witcher"?

List the streaming platforms for all the series in the table.

Find the series with the most awards won.

#### Questions on Table 2: Actors in Sci-Fi Web Series

What information is stored in the "Actors in Sci-Fi Web Series" dataset?

Identify the actors born in the 1990s.

Who is the youngest actor in the dataset?

List actors who have appeared in more than one sci-fi web series.

Find the nationality of the actor with the most total appearances.

Retrieve actors who have worked in series with an average rating above 8.

#### Questions on Table 3: Roles in Sci-Fi Web Series

Explain the purpose of the "Roles in Sci-Fi Web Series" dataset.

Find all roles played by an actor with the first name "Millie."

List the actors who portrayed the character "Dolores Abernathy."

Identify the series where "Henry Cavill" played a role.

List roles played in the series with an average rating above 8.

Find the series with the most roles listed in the dataset.

#### Comments

Comment by [Manjesh Verma](#) [ 06/Dec/23 ]

[Manjesh-Verma-13/Qy by Entertainment table.sql at Mysql DB · volcanusacademy/Manjesh-Verma-1](#)



**[DAP-13] [SELECT, GROUP BY and HAVING, SUB Query assignment](#)** Created:

20/Nov/23 Updated: 21/Nov/23

<b>Status:</b>	In Progress
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Unresolved	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0003r:
<b>Sprint:</b>	DAP Sprint 1

**Description****SELECT, GROUP BY and HAVING, SUB Query assignment :****SELECT**

1. Retrieve the first and last names of all students.
2. Get a list of students born after the year 2000.
3. List all female students.
4. Find the students who live in 'CA' (California).
5. Retrieve the email addresses and phone numbers of students with the last name 'Smith'.
6. Get a list of students who live in 'NY' (New York) and were born before 2000.
7. List the first names of male students who live in 'TX' (Texas).
8. Retrieve the first and last names of students who live in 'CA' and were born in or after 1998.
9. Find the students who have 'Smith' in their last name or 'John' in their first name.
10. List the students in alphabetical order of their last names.

**GROUP BY and HAVING**

1. Find the count of students in each state.
2. Calculate the average age of students by gender.
3. List states with more than 5 students.
4. Find the count of students who have the same last name and display only those with a count greater than 1.
5. Get the average length of email addresses by state, but only for states with more than 2 students.
6. Find the minimum and maximum birth year for each state with at least 3 students.
7. Calculate the average phone number length for male and female students, but only for genders with more than 5 students.
8. List states where the youngest student is less than 20 years old.
9. Get the average length of addresses for students in states that have 'ville' in the city name.
10. Find the count of students by the first letter of their first names for students in states with more than 2 students.

### **SUB Query**

1. Find the student with the earliest birthdate.
2. List all students with the same last name as the youngest student.
3. Retrieve the states with more students than the state with the fewest students.
4. Find the students with a birthdate later than the average birthdate of female students.
5. List students who share the same birthdate with at least one other student.
6. Get the count of students in states with at least one male student.
7. Retrieve the names of students who do not have a matching email address in another student's record.
8. List states with students who are older than the average age of students in 'CA'.
9. Find the students with the same phone number as another student.
10. Retrieve students who were born after the most recent birthdate in the table.

**[DAP-12] [School Database assignment](#)** Created: 20/Nov/23 Updated: 21/Nov/23

<b>Status:</b>	In Progress
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Unresolved	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0003f:
<b>Sprint:</b>	DAP Sprint 1

**Description****School Database assignment :****Q.1 Create a Database**

1. Create a new database named SchoolDatabase.
2. Inside the SchoolDatabase, create a table named Students with columns student\_id

(auto-increment primary key), first\_name, last\_name, and birthdate.

1. In the SchoolDatabase, create another table named Courses with columns course\_id

(auto-increment primary key), course\_name, and instructor.

1. Insert three sample student records into the Students table in the SchoolDatabase.
2. Insert five sample course records into the Courses table in the SchoolDatabase.

**Q.2** Write a SQL statement to create a table named jobs including columns job\_id, job\_title, min\_salary, max\_salary and check whether the max\_salary amount exceeds the upper limit 25000.

**Q.3** Write a SQL statement to create a table job\_history including columns

employee\_id, start\_date, end\_date, job\_id and department\_id and make sure that, the employee\_id column does not contain any duplicate value at the time of insertion and the foreign key column job\_id contain only those values which are exists in the jobs table.

Here is the structure of the table jobs;

---

Field Type Null Key Default Extra

---

JOB_ID	varchar(10)	NO	PRI	
JOB_TITLE	varchar(35)	NO		NULL
MIN_SALARY	decimal(6,0)	YES		NULL
MAX_SALARY	decimal(6,0)	YES		NULL

---

Q. 4. Write a SQL statement to create a table employees including columns employee\_id, first\_name, last\_name, email, phone\_number hire\_date, job\_id, salary, commission, manager\_id and department\_id and make sure that, the employee\_id column does not contain any duplicate value at the time of insertion and the foreign key columns combined by department\_id and manager\_id columns contain only those unique combination values, which combinations are exists in the departments table.

Assume the structure of departments table below.

---

Field Type Null Key Default Extra

---

DEPARTMENT_ID	decimal(4,0)	NO	PRI 0	
DEPARTMENT_NAME	varchar(30)	NO		NULL
MANAGER_ID	decimal(6,0)	NO	PRI 0	
LOCATION_ID	decimal(4,0)	YES		NULL

---

Q. 5. Write a SQL statement to create a table employees including columns employee\_id, first\_name, last\_name, job\_id, salary and make sure that, the employee\_id column does not contain any duplicate value at the time of insertion, and the foreign key column job\_id, referenced by the column job\_id of jobs table, can contain only those values which are exists in the jobs table. The InnoDB Engine have been used to create the tables. The specialty of the

statement is that, The ON UPDATE CASCADE action allows you to perform cross-table update and ON DELETE RESTRICT action reject the deletion. The default action is ON DELETE RESTRICT.

Assume that the structure of the table jobs and InnoDB Engine have been used to create the table jobs:

```
CREATE TABLE IF NOT EXISTS jobs(  
JOB_ID integer NOT NULL UNIQUE PRIMARY KEY,  
JOB_TITLE varchar(35) NOT NULL DEFAULT '',  
MIN_SALARY decimal(6,0) DEFAULT 8000,  
MAX_SALARY decimal(6,0) DEFAULT NULL  
)ENGINE=InnoDB;
```

Field Type Null Key Default Extra

JOB_ID	int(11)	NO	PRI	NULL
JOB_TITLE	varchar(35)	NO		
MIN_SALARY	decimal(6,0)	YES		8000
MAX_SALARY	decimal(6,0)	YES		NULL



**[DAP-11] [GroupBy and Having assignment](#)** Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i00027:
<b>Sprint:</b>	DAP Sprint 1

#### Description

## GroupBy and Having assignment :

To excute the provided queries involving GROUP BY and HAVING clauses, you should have two main tables in your database: `Employees` and `Departments`.

### GROUP BY and HAVING queries

1. Find Departments with More Than 5 Employees:
2. List the departments with more than 5 employees.
3. Calculate Average Salary by Department for Departments with More Than 3 Employees:
4. Identify Locations with High Average Salaries:
5. Find Job Titles with at Least 3 Employees:
6. Calculate Total Salary Expenditure for Departments with Average Salaries Below \$55,000:
7. Count the Number of Employees Hired in Each Year:
8. Identify Departments with No Employees Earning Over \$70,000:
9. Calculate the Average Years of Service by Department for Departments with at Least 2 Employees:
10. Find Locations with Only One Department:

11. Identify Departments with the Highest Average Salary:

#### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[Manjesh-Verma-13/Groupby\\_Having\\_Queries.sql at Mysql\\_DB · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#)

I have complete the 11 queries of group by having assignment.

**[DAP-10] [FacultyMembers and Courses tables assignment](#)** Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0001z:
<b>Sprint:</b>	DAP Sprint 1

#### Description

## FacultyMembers and Courses tables assignment :

In this assignment, students are required to work with the `FacultyMembers` and `Courses` tables. The task involves writing SQL queries to extract specific information from these tables.

1. List the details of all faculty members.
2. List the details of all courses.
3. List the courses taught by a specific faculty member (e.g., 'Amit Kumar').
4. Find the faculty member who teaches a specific course (e.g., 'Introduction to Computer Science').
5. List the faculty members in the 'Computer Science' department.
6. List the courses in the 'Physics' department.
7. Find the faculty members who have 'Machine Learning' in their research interests.
8. List the courses with a start date after '2023-02-01'.
9. Find the course with the highest number of credits.
10. List the faculty members who are not currently active.
11. List the courses with descriptions containing the word 'principles'.
12. Find the faculty members who teach courses with more than 3 credits.
13. List the courses taught by faculty members with a 'Ph.D.' academic degree.

14. Find the faculty members who are tenured.
15. List the faculty members with more than 5 years of experience.
16. Update the phone number of 'Amit Kumar' to '987-654-3210'.
17. Add a new course with the following details: 'Database Management', '1', 'Computer Science', 'DB101', '2023-03-01', '2023-06-01', 'C302', '3.5', 'Introduction to database management.'.
18. Change the department of 'Priya Sharma' to 'Computer Science'.
19. Delete the course with course code 'CHEM201'.
20. Mark 'Rahul Yadav' as not currently active.
21. Add 2 years of experience to all faculty members.
22. Remove the academic degree of 'Rajesh Patel'.
23. Add a new faculty member named 'Sandeep Gupta' with relevant details.
24. Update the course room number for 'Introduction to Computer Science' to 'C302'.
25. Change the description of 'Physics 101' to 'Fundamental principles of physics and electromagnetism.'.
26. Calculate the average years of experience for all faculty members.
27. Find the faculty member with the highest years of experience.
28. Count the number of courses in each department.
29. Calculate the total number of credits offered by all courses.
30. Determine the number of faculty members in each department.
31. Calculate the average number of credits for all courses.
32. Find the faculty member with the most research interests.
33. Count the number of full-time and part-time faculty members.
34. Calculate the total number of credits offered by the 'Computer Science' department.
35. Determine the department with the highest average years of experience.
36. List the names of faculty members and the courses they teach.
37. Find the faculty members who teach more than one course.
38. List the courses and their faculty members' last names.
39. Find the faculty members who teach courses with more than 4 credits.
40. List the courses, their faculty members, and the department in which they teach.
41. Find the faculty members who teach courses with start dates in 2023.
42. List the courses and their faculty members who are tenured.
43. Find the faculty members who teach in the same department as 'Amit Kumar'.
44. List the courses taught by faculty members with 'Physics' in their research interests.
45. Find the faculty members who have the same first name as any course name.
46. Find the faculty members who teach courses with the highest number of credits.
47. List the courses and their faculty members who are not currently active.
48. Find the courses with start dates later than the latest hire date of faculty members.
49. List the courses with descriptions longer than the average description length.
50. Find the faculty members who teach courses with the earliest start date.
51. List the courses and their faculty members who teach courses in the 'Computer Science' department.
52. Find the faculty members who have the same department as 'Physics 101'.
53. List the courses and their faculty members who teach in departments with more than 3 faculty members.

54. Find the faculty members who teach courses with credits greater than the average credits.
55. List the courses and their faculty members who teach courses with descriptions containing 'electromagnetism'.
56. List the faculty members whose first names contain the letter 'a' and last names contain the letter 's'.
57. Find the courses whose names start with the word 'Introduction.'
58. List the faculty members who have an email address with the domain <http://example.com>.
59. Find the courses with descriptions containing the phrase 'advanced topics.'
60. List the faculty members and their email addresses in all uppercase.
61. Find the courses with course codes that end with '101'.
62. List the faculty members whose last names are exactly four characters long.
63. Find the courses with names that do not contain the word 'Advanced.'
64. List the faculty members and their phone numbers without dashes.
65. Find the courses with descriptions longer than 100 characters
66. Retrieve the faculty members' full names by concatenating their first name and last name with a space in between.
67. Find the courses that have the word 'Science' located within their descriptions.
68. List the first 5 characters of the course names.
69. Get the last 3 characters of the course codes.
70. Display the course names reversed (e.g., 'Computer Science' should appear as 'ecneicS retupmoC').
71. Replace the word 'Advanced' in course names with 'Intermediate.'
72. Compare the faculty members' first names with the string 'John' and retrieve faculty members with names that are greater (alphabetically) than 'John'.
73. Find courses with descriptions containing 'math' and replace it with 'mathematics.'
74. Retrieve the current date and time.
75. List the faculty members and their hire dates.
76. Find the courses that have already started (based on the current date).
77. Calculate the duration (in days) of each course by subtracting the start date from the end date.
78. Display the day of the week for the start date of each course.
79. List the faculty members and their years of experience.
80. Calculate the difference in years between the current date and each faculty member's hire date to determine their years of service.
81. Find the courses that have a start date in the future (after the current date).
82. Calculate the average duration (in days) of all courses.
83. Determine the day of the week for the current date.
84. Retrieve the courses and their start dates sorted in ascending order.
85. List the faculty members and their ages based on their birthdates.
86. Find the courses taught by faculty members with more than 10 years of experience.
87. Calculate the total number of days between the start date of the course 'Physics 101' and the end date of the course 'Chemistry 101.'
88. Retrieve the courses that have their start dates on a Monday.
89. Calculate the average years of experience for faculty members who are tenured.

90. List the faculty members and the number of years remaining until their retirement (assuming retirement at age 65).
91. List all faculty members and their research interests as a comma-separated list.
92. Display the courses taught by each faculty member as a comma-separated list.
93. List all course codes within each department as a comma-separated list.
94. Show all research interests of faculty members in the 'Computer Science' department as a comma-separated list.
95. Display the course descriptions for each faculty member's courses as a comma-separated list.
96. List the start dates of courses for each department as a comma-separated list.
97. Show the full-time or part-time status of faculty members along with their course names as a comma-separated list.
98. List the course names taught by faculty members with 'Ph.D.' degrees as a comma-separated list.
99. Show all faculty members and their respective department's course codes as a comma-separated list.
100. List the departments along with the tenure status of faculty members in each department as a comma-separated list.

#### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[Manjesh-Verma-13/FacultyMembers\\_Courses\\_Queries.sql at Mysql\\_DB · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#)

I have complete the 100 queries of facultymembers and courses

**[DAP-9] [Database Creation](#)** Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0001r:
<b>Sprint:</b>	DAP Sprint 1

#### Description

## Database Creation :

### Assignment Questions:

#### Database Creation:

- Create a database named `Company`. Desc-- I have create the database named Company.

#### Table Creation:

- Inside the `Company` database, create a table called `Employee` with the following fields:
- `EmployeeID` (Integer, Primary Key)
- `FirstName` (VARCHAR, 50 characters)
- `LastName` (VARCHAR, 50 characters)
- `Salary` (Decimal, 10 digits in total with 2 decimal places)
- `Department` (VARCHAR, 50 characters)
- `HireDate` (Date)
- `JobTitle` (VARCHAR, 100 characters)

- `Location` (VARCHAR, 100 characters) Desc-- I have create the table name employee with the of given dataset.

**Data Insertion:**

- Insert 10 records into the `Employee` table with appropriate values for each field. Desc-- I have insert the 10 records of Employee in the table Employee.

**Comments**

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[https://github.com/volcanusacademy/Manjesh-Verma-13/blob/Mysql\\_DB/Database\\_Creation\\_Queries.sql](https://github.com/volcanusacademy/Manjesh-Verma-13/blob/Mysql_DB/Database_Creation_Queries.sql) I have completed the creation of database and also the insertion of data into the table.



**[DAP-8]** [MySQL Case Queries](#) Created: 20/Nov/23 Updated: 21/Nov/23

<b>Status:</b>	In Progress
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Unresolved	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0003j:
<b>Sprint:</b>	DAP Sprint 1

#### Description

## MySQL Case Queries :

MySQL Case Queries refer to using the `CASE` statement in MySQL queries to perform conditional logic and return different values based on specified conditions within the database.

Please find the attached document and complete it.

[DAP-7] [Select Queries](#) Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

Status:	Done
Project:	<a href="#">Data Analytics Project</a>
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Task	Priority:	Medium
Reporter:	<a href="#">Manjesh Verma</a>	Assignee:	<a href="#">Manjesh Verma</a>
Resolution:	Done	Votes:	0
Labels:	Mysql		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0001b:
Sprint:	DAP Sprint 1

#### Description

### Select Queries :

1. Retrieve the first and last names of all students.
2. Get a list of students born after the year 2000.
3. List all female students.
4. Find the students who live in 'CA' (California).
5. Retrieve the email addresses and phone numbers of students with the lastname 'Smith'.
6. Get a list of students who live in 'NY' (New York) and were born before 2000.
7. List the first names of male students who live in 'TX' (Texas).
8. Retrieve the first and last names of students who live in 'CA' and were born in or after 1998.
9. Find the students who have 'Smith' in their last name or 'John' in their firstname.
10. List the students in alphabetical order of their last names.

#### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[Manjesh-Verma-13/Select\\_Queries.sql at Mysql\\_DB · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#) I have completed the 10 select queries of SQL



**[DAP-6] [SQL Joins Practice](#)** Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i00013:
<b>Sprint:</b>	DAP Sprint 1

**Description**

- (Refer to the 'Customers,' 'Orders,' and 'Shippings' tables.)
- 1. Retrieve the first names of customers along with the item names they have ordered.
- 2. List all customers and their orders, including those who haven't placed any orders.
- 3. Display all orders and their corresponding customer names, including orders without associated customers.
- 4. Retrieve a list of all customers and their orders. Include both customers without orders and orders without customers.
- 5. List customers who haven't placed any orders.
- 6. Display orders without associated customers.
- 7. Find pairs of customers from the same country.
- 8. Generate all possible combinations of products (items) and customers.
- 9. Retrieve the item names and shipping statuses for orders with shipping information.
- 10. List all orders and their shipping statuses, including orders without shipping information.
- 11. Display all shipping records and their corresponding order items, including records without associated orders.
- 12. Retrieve a list of all orders and their shipping statuses. Include both orders without shipping information and shipping records without associated orders.
- 13. List orders without shipping information.
- 14. Display shipping records without associated orders.

- 15. Find pairs of orders with the same item and amount.
- 16. Generate all possible combinations of orders and shipping records.

### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[Manjesh-Verma-13/Sql\\_Joins.sql at Mysql\\_DB · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#) I have completed the 16 queries of SQL joins.

**[DAP-5] [Queries on the 'Shippings' Table:](#)** Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i0000v:
<b>Sprint:</b>	DAP Sprint 1

#### Description

### Queries on the 'Shippings' Table: :

1. Retrieve all shipping information from the 'Shippings' table.
2. Find the number of shipping records in the 'Shippings' table.
3. Retrieve the shipping statuses for orders with customer IDs 3 and 4.
4. List the distinct shipping statuses.
5. Count the number of pending and delivered orders.
6. Retrieve the shipping statuses for orders with shipping IDs 1 and 3.
7. Find the most common shipping status in the 'Shippings' table.
8. Retrieve the shipping statuses for orders with customer IDs 1, 2, and 3.
9. List the shipping records in ascending order of shipping ID.
10. Find the shipping status for order with shipping ID 5.
11. Retrieve the shipping statuses for orders with customer IDs 1, 2, and 4.
12. List the shipping records in descending order of status.
13. Find the total number of pending orders.
14. Retrieve the shipping statuses for orders with customer IDs 2 and 5.
15. Count the number of delivered orders.
16. Retrieve the shipping statuses for orders with shipping IDs 2 and 4.
17. List the shipping records in alphabetical order by status.
18. Find the total number of records with 'Pending' status.

19. Retrieve the shipping statuses for orders with customer IDs 3 and 5.
20. List the shipping records in reverse alphabetical order by status.

### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[Manjesh-Verma-13/Qu\\_by\\_shipping\\_table.sql at Mysql DB · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#)

I have completed the 20 queries of shippings tables.

<b>[DAP-4] <a href="#">Queries on the 'Orders' Table:</a></b> Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23	
<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Issue links:</b>	<b>Blocks</b>		
	blocks	<a href="#">DAP-4</a> Database Creation and Manipulation	Done
<b>Rank:</b>	0 i0000n:		
<b>Sprint:</b>	DAP Sprint 1		

## Description

### Queries on the 'Orders' Table: :

1. Retrieve all order information from the 'Orders' table.
2. Find the number of orders in the 'Orders' table.
3. Retrieve the item names and amounts for orders placed by customers from the 'USA.'
4. List the distinct items that have been ordered.
5. Find the average order amount.
6. Retrieve the item names and customer IDs for orders with amounts greater than \$500.
7. Find the most expensive item in the 'Orders' table.
8. Retrieve the item names and customer IDs for orders with amounts less than or equal to \$200.
9. Count the number of orders placed by each customer.
10. Retrieve the item names and amounts for orders placed by customers from the 'UK.'
11. List the orders in descending order of amount.
12. Find the order with the highest amount in the 'Orders' table.
13. Retrieve the item names and amounts for orders placed by customers from the 'USA' or 'UK.'



14. Count the number of orders placed for each item.
15. Retrieve the item names and amounts for the top 3 most expensive orders.
16. List the orders in alphabetical order by item name.
17. Find the order with the lowest amount.
18. Retrieve the item names and amounts for orders placed by customers from the 'UK' or 'USA.'
19. Find the total number of orders with amounts greater than \$300.
20. Retrieve the item names and amounts for orders placed by customers with IDs 1 and 2.

#### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[Manjesh-Verma-13/Qu\\_by\\_order\\_table.sql at Mysql DB · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#) I have completed the 20 queries of orders table.

**[DAP-3] [Queries on the 'Customers' Table:](#)** Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Issue links:</b>	<b>Blocks</b>
	blocks <a href="#">DAP-1</a> Database Creation and Manipulation Done
<b>Rank:</b>	0 i0000f:
<b>Sprint:</b>	DAP Sprint 1

### Description

## Queries on the 'Customers' Table: :

1. Retrieve all customer information from the 'Customers' table. Desc-- Here we use select all to show full information on the table.
2. Find the number of customers in the 'Customers' table. Desc--Here we find the total numbers of customers.
3. Retrieve the first name and age of customers who are from the 'UK.' Desc-- Here we retrieve the firstname and age of customers.
4. List the distinct countries where customers are located.
5. Find the average age of customers.
6. Retrieve the first name and country of customers who are not from the 'USA.'
7. Find the oldest customer in the 'Customers' table.
8. Retrieve the first name and last name of customers whose last names contain the substring 'Doe.'
9. Count the number of customers in each country.
10. Retrieve the first name and age of customers from the 'USA' who are older than 30.
11. List the customers in descending order of age.

12. Find the youngest customer in the 'Customers' table.
13. Retrieve the first name and country of customers who are not from the 'UK.'
14. Count the number of customers in each age group (e.g., 20-29, 30-39, etc.).
15. Retrieve the first name and age of the top 5 oldest customers.
16. List the customers in alphabetical order by last name.
17. Find the customer with the highest age.
18. Retrieve the first name and age of customers who are either from the 'USA' or 'UK.'
19. Find the total number of customers who are older than 25.
20. Retrieve the first name and age of customers from the 'UK' who are younger than 25.

### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[https://github.com/volcanusacademy/Manjesh-Verma-13/blob/Mysql\\_DB/Qu\\_by\\_customer\\_table.sql](https://github.com/volcanusacademy/Manjesh-Verma-13/blob/Mysql_DB/Qu_by_customer_table.sql) I have completed the 20 queries of customers tables.

**[DAP-2] [Create a Database and Tables in SQL](#)** Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23

<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Rank:</b>	0 i00007:
<b>Sprint:</b>	DAP Sprint 1

#### Description

## Create a Database and Tables in SQL :

**Question 1:** Create a new database named 'Volcanus.'

Desc-- have create the database named volcanus.

**Question 2:** Create a table called 'Customers' with the following columns:

customer\_id (INT, Primary Key)

first\_name (VARCHAR(50))

last\_name (VARCHAR(50))

age (INT)

country (VARCHAR(50))

Desc--I have create the table name Customers with the help given following given dataset.

**Question 3:** Insert the following data into the 'Customers' table:

Customer ID: 1, First Name: Marcus, Last Name: Doe, Age: 31, Country: USA

Customer ID: 2, First Name: Robert, Last Name: Luna, Age: 22, Country: USA

Customer ID: 3, First Name: David, Last Name: Robinson, Age: 22, Country: UK

Customer ID: 4, First Name: Paul, Last Name: Reinhardt, Age: 25, Country: UK

Customer ID: 5, First Name: Alexandra, Last Name: Doe, Age: 28, Country: UAE

Desc--I insert given dataset into the table, here I insert the five Customers.

**Question 4:** Create a table called 'Orders' with the following columns:

order\_id (INT, Primary Key)

item (VARCHAR(50))

amount (DECIMAL(10, 2))

customer\_id (INT)

Desc--I have create the table name orders with the help given following given dataset.

**Question 5:** Insert the following data into the 'Orders' table:

Order ID: 1, Item: Keyboard, Amount: \$400.00, Customer ID: 4

Order ID: 2, Item: Mouse, Amount: \$300.00, Customer ID: 4

Order ID: 3, Item: Monitor, Amount: \$12,000.00, Customer ID: 3

Order ID: 4, Item: Keyboard, Amount: \$400.00, Customer ID: 1

Order ID: 5, Item: Mousepad, Amount: \$250.00, Customer ID: 2

Desc--I insert given dataset into the table, here I insert the five Orders.

**Question 6:** Create a table called 'Shippings' with the following columns:

shipping\_id (INT, Primary Key)

status (VARCHAR(50))

customer (INT)

Desc--I have create the table name shippings with the help given following given dataset.

**Question 7:** Insert the following data into the 'Shippings' table:

Shipping ID: 1, Status: Pending, Customer: 2

Shipping ID: 2, Status: Pending, Customer: 4

Shipping ID: 3, Status: Delivered, Customer: 3

Shipping ID: 4, Status: Pending, Customer: 5

Shipping ID: 5, Status: Delivered, Customer: 1

Desc--I insert given dataset into the table, here I insert the five shipping data.

#### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

[https://github.com/volcanusacademy/Manjesh-Verma-13/blob/Mysql\\_DB/SQL\\_Tables.sql](https://github.com/volcanusacademy/Manjesh-Verma-13/blob/Mysql_DB/SQL_Tables.sql) I have completed the creation of database and tables.

<b>[DAP-1] <a href="#">Database Creation and Manipulation</a></b> Created: 20/Nov/23 Updated: 21/Nov/23 Resolved: 21/Nov/23	
<b>Status:</b>	Done
<b>Project:</b>	<a href="#">Data Analytics Project</a>
<b>Components:</b>	None
<b>Affects versions:</b>	None
<b>Fix versions:</b>	None

<b>Type:</b>	Task	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">Manjesh Verma</a>	<b>Assignee:</b>	<a href="#">Manjesh Verma</a>
<b>Resolution:</b>	Done	<b>Votes:</b>	0
<b>Labels:</b>	Mysql		
<b>Remaining Estimate:</b>	Not Specified		
<b>Time Spent:</b>	Not Specified		
<b>Original estimate:</b>	Not Specified		

<b>Issue links:</b>	<b>Blocks</b>		
	is blocked by	<a href="#">DAP-3</a> Queries on the 'Customers' Table:	Done
	is blocked by	<a href="#">DAP-4</a> Queries on the 'Orders' Table:	Done
<b>Rank:</b>	0 hzzzzz:		
<b>Sprint:</b>	DAP Sprint 1		

## Description

# Database Creation and Manipulation :

## Q.1 Create a Database

1. Create a new database named SchoolDatabase. Desc--Here I have created a database named SchoolDatabase.
2. Inside the SchoolDatabase, create a table named Students with columns student\_id (auto-increment primary key), first\_name, last\_name, and birthdate. Desc--I have create a table named SchoolDatabase and help to the given dataset.
3. In the SchoolDatabase, create another table named Courses with columns course\_id (auto-increment primary key), course\_name, and instructor. Desc--I have create a table named Courses and help to the given dataset.
4. Insert three sample student records into the Students table in the SchoolDatabase. Desc-- I insert the five students details in the table students

5. Insert five sample course records into the Courses table in the SchoolDatabase. Desc-- I insert the five Courses details in the table students

Q.2 Write a SQL statement to create a table named jobs including columns job\_id, job\_title, min\_salary, max\_salary and check whether the max\_salary amount exceeds the upper limit 25000.

Q.3 Write a SQL statement to create a table job\_history including columns employee\_id, start\_date, end\_date, job\_id and department\_id and make sure that, the employee\_id column does not contain any duplicate value at the time of insertion and the foreign key column job\_id contain only those values which are exists in the jobs table.

Here is the structure of the table jobs;

Field	Type	Null	Key	Default	Extra
JOB_ID	varchar(10)	NO	PRI		
JOB_TITLE	varchar(35)	NO		NULL	
MIN_SALARY	decimal(6,0)	YES		NULL	
MAX_SALARY	decimal(6,0)	YES		NULL	

Q. 4. Write a SQL statement to create a table employees including columns employee\_id, first\_name, last\_name, email, phone\_number hire\_date, job\_id, salary, commission, manager\_id and department\_id and make sure that, the employee\_id column does not contain any duplicate value at the time of insertion and the foreign key columns combined by department\_id and manager\_id columns contain only those unique combination values, which combinations are exists in the departments table.

Assume the structure of departments table below.

Field	Type	Null	Key	Default	Extra
DEPARTMENT_ID	decimal(4,0)	NO	PRI	0	



```

DEPARTMENT_NAME varchar(30) NO NULL
MANAGER_ID decimal(6,0) NO PRI 0
LOCATION_ID decimal(4,0) YES NULL

```

Q. 5. Write a SQL statement to create a table employees including columns employee\_id, first\_name, last\_name, job\_id, salary and make sure that, the employee\_id column does not contain any duplicate value at the time of insertion, and the foreign key column job\_id, referenced by the column job\_id of jobs table, can contain only those values which are exists in the jobs table. The InnoDB Engine have been used to create the tables. The specialty of the statement is that, The ON UPDATE CASCADE action allows you to perform cross-table update and ON DELETE RESTRICT action reject the deletion. The default action is ON DELETE RESTRICT.

Assume that the structure of the table jobs and InnoDB Engine have been used to create the table jobs.

```

CREATE TABLE IF NOT EXISTS jobs (
JOB_ID integer NOT NULL UNIQUE PRIMARY KEY,
JOB_TITLE varchar(35) NOT NULL DEFAULT '',
MIN_SALARY decimal(6,0) DEFAULT 8000,
MAX_SALARY decimal(6,0) DEFAULT NULL
)ENGINE=InnoDB;

```

Field	Type	Null	Key	Default	Extra
-------	------	------	-----	---------	-------

JOB_ID	int(11)	NO	PRI	NULL	
JOB_TITLE	varchar(35)	NO			
MIN_SALARY	decimal(6,0)	YES		8000	

MAX\_SALARY decimal(6,0) YES NULL

---

### Comments

Comment by [Manjesh Verma](#) [ 21/Nov/23 ]

1. [Manjesh-Verma-13/Database\\_Creation\\_Queries.sql at Mysql\\_DB · volcanusacademy/Manjesh-Verma-13 \(github.com\)](#)

I have complete the query of MYSQL.

Generated at Thu Feb 15 11:49:30 UTC 2024 by Manjesh Verma using Jira 1001.0.0-SNAPSHOT#100247-sha1:972536d58906b31eea2912f9bfe79ae550e66b11.