

**CERTIFICATE
OF ACHIEVEMENT**



Data Analyst



THIS IS TO CERTIFY THAT

Bhavin Shah

Has successfully graduated from the program having completed all mandated course requirements and industry projects with distinction

Date: June 26, 2024

Certificate ID: 104846223

<https://success.simplilearn.com/945ca369-89a6-48c2-8650-de6d2b57fe58>

Krishna Kumar, CEO



Contact

Phone

9820-177-312

Email

bhavinshah139@gmail.com

Address

D-29, Ramesh mansion, Gaonthan,
Chembur, Mumbai, 400071

Education

2017-2018

Bachelor in Mass Media
S.k. somaiya college.

Expertise & Skills

- Python
- Tableau
- Excel
- Power bi
- My SQL

Profile Link

LinkedIn

<https://www.linkedin.com/in/bhavin-shah-9b3838306/>

Bhavin Shah

Have gained good exposure to understand various aspects of Data Analytics through the Post Graduate Program which has developed a great sense of confidence at a very personal level. Hence, I seek a challenging position in the area of Data Science, BI, Data Analysis & related fields where I can share my skills and expand my capabilities further in the pursuit of progressive career advancement.

Certification

Masters Program - Data Analyst : Simplilearn Certified in collaboration with IBM - 2024

Projects :

1. Power Bi:

- Making an interactive Dashboard with Zomato data.
- Making an interactive Dashboard with Blinkit data.
- Making an interactive Dashboard for Credit card company and its customers.

2. SQL:

- Analyzing 1000+ rows for Air Cargo Analysis.
- Analyzing 1000+ rows of employee data for Science Qtech company.
- Analyzing data for a music store.

3. Excel:

- Designing a dashboard for a E-commerce business.
- Analyzing data and then Predicting restaurant tips using predictive analytics.

4. Python:

- Using python numpy, pandas and matplotlib to analysis insurance Data and represent it with charts created in python itself.

5. Tableau:

- Comparing Region vise sale for a superstore.

Experience

○ 11th Dec 2023 - 16th April 2024

Accenture CS and Backend team
Level 12 Associate

○ 2nd Sep 2022 - 5th Dec 2023

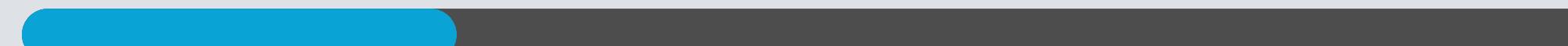
Gebbs medical billing AR/Denials.
Facility AR Associate

○ 30th May 2021- 31 Aug 2022

Wipro medical billing AR/Denials.
AR Associate.



Power BI Project 1.0



Beginner

Intermediate

Expert

WHAT I DID ?

1. Data Import:

- Imported data from all available Excel files.

2. Data Transformation:

- Cleaning the data

3. DAX Usage:

- Added Measures like the number of items, Avg sales, total sales, Avg rating.

4. Creating Metrics

- added the Measures as metrics.

5. Dashboard Report:

- Created a Blinkit dashboard.

Blinkit project

File Home Insert Modeling View Optimize Help

Get data New visual Visual gallery New calculation Publish Copilot

blinkit
India's Last Minute App

FILTER PANEL

- OUTLET LOCATION: All
- OUTLET SIZE: All
- ITEM TYPE: All

TOTAL SALES \$1.20M

Avg Sales \$141

OF ITEMS 8523

Avg Rating 3.9

	Total Sales	Avg Sales	Avg Rating	# of Items
FAT CONTENT	\$425K	\$776K		
Low Fat	\$0.18M	\$0.18M		
Regular	\$0.14M	\$0.12M		
	\$0.10M	\$0.09M		
	\$0.08M	\$0.07M		
	\$0.06M	\$0.06M		
	\$0.04M	\$0.03M		
	\$0.02M	\$0.02M		
	\$0.02M	\$0.01M		

FAT BY OUTLET

	Total Sales
Tier 3	\$0.31M
Tier 2	\$0.25M
Tier 1	\$0.22M

OUTLET ESTABLISHMENT

Year	Sales
2012	\$78K
2014	\$130K
2016	\$132K
2018	\$205K
2020	\$129K
2022	\$131K

OUTLET SIZE

Size	Count
Medium	\$445K
Small	\$508K
High	\$249K

OUTLET LOCATION

Tier	Total Sales
Tier 1	336.40K
Tier 2	393.15K
Tier 3	472.13K

OUTLET TYPE

Outlet Type	Total Sales	# of Items	Avg Rating	Avg Sales	Item Visibility
Supermarket Type1	\$788K	5577	3.9	\$141	0.06
Grocery Store	\$152K	1083	3.9	\$140	0.10
Supermarket Type2	\$131K	928	3.9	\$142	0.06
Supermarket Type3	\$131K	935	3.9	\$140	0.06

Blinkit +



Power BI Project 2.0



Beginner

Intermediate

Expert

WHAT I DID ?

1. Data Import:

- Imported data from all available Excel files.

2. Data Transformation:

- Corrected city names:
 - "Sí£o Paulo" to "São Paulo"
 - "Cedar Rapids/Iowa City" to "Cedar Rapids"
 - "ÜÁstanbul" to "Istanbul"

3. Data Structuring:

- Created separate columns for Restaurant Name and Address.
- Made a dimension table for unique, non-blank cuisines.

4. DAX Usage:

- Added a Rating Color column:
 - Above 4.5: Dark Green
 - 4 to 4.4: Green

5. Measures Created:

- Restaurant Count: `DISTINCTCOUNT(KPIs[Restaurant ID])`
- Average Cost: `AVERAGE(KPIs[Average Cost for two])`
- Average Rating: `AVERAGE(KPIs[Aggregate rating])`
- Cuisine Count: `DISTINCTCOUNT('cuisines and rest id'[Value])`

6. Country Code Table:

- Added a "Continent" column using country-continent mapping (e.g., "Africa – South Africa").

7. Dashboard Report:

- Created a Zomato dashboard.
- Enabled action navigation with arrow images for page toggling.



ZOMATO ANALYSIS DASHBOARD

CONTINENT

Africa

Asia

Europe

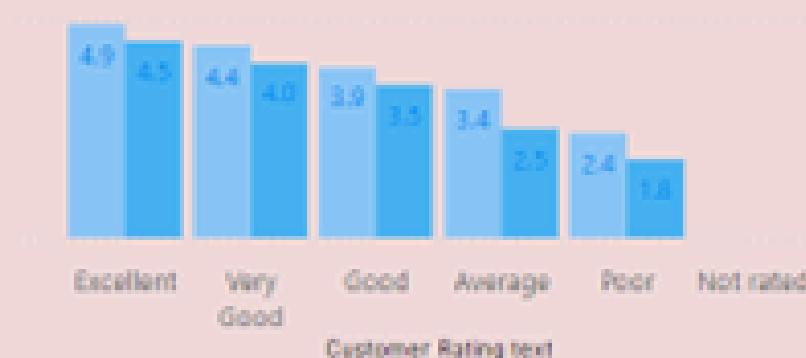
North America

Oceania

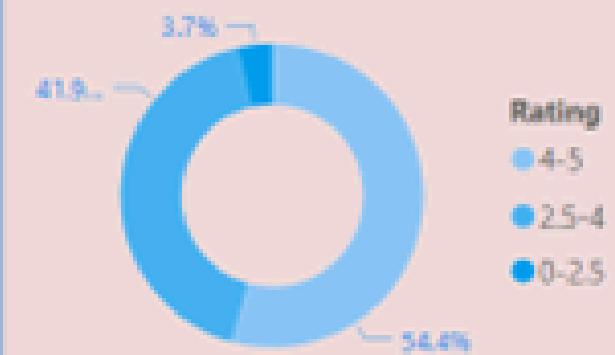
South America

Average Customer Rating - High & Low

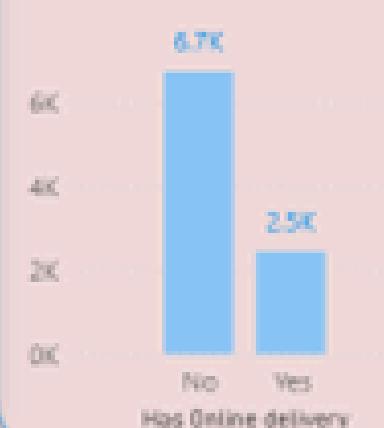
Max of Aggregate rating Min of Aggregate rating



Average Rating



Restaurants having Online Delivery

**9551**

Restaurant Count

226

Cuisine Count

2.67

Average Rating



Page 1

Page 2

+

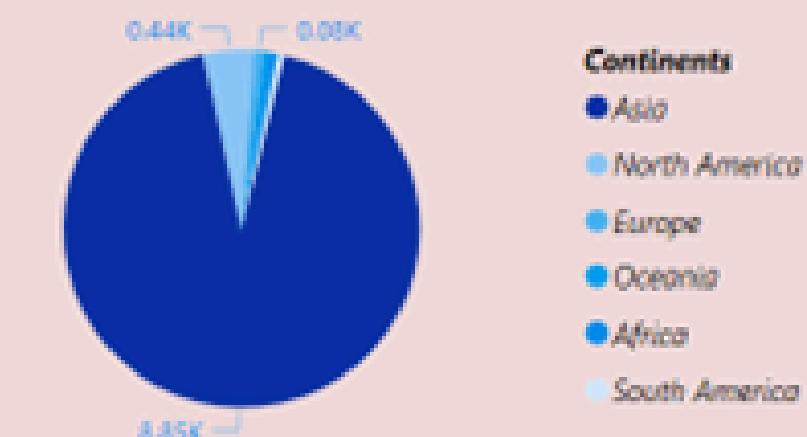


ZOMATO ANALYSIS DASHBOARD

CONTINENT

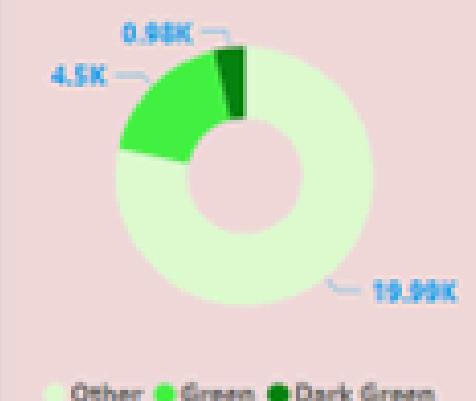
Africa Asia Europe North America Oceania South America

Count of Restaurants by Continents



Customer Rating

Dark Green = above 4.5 & Green = 4 to 4.4



Numbers of Restaurants

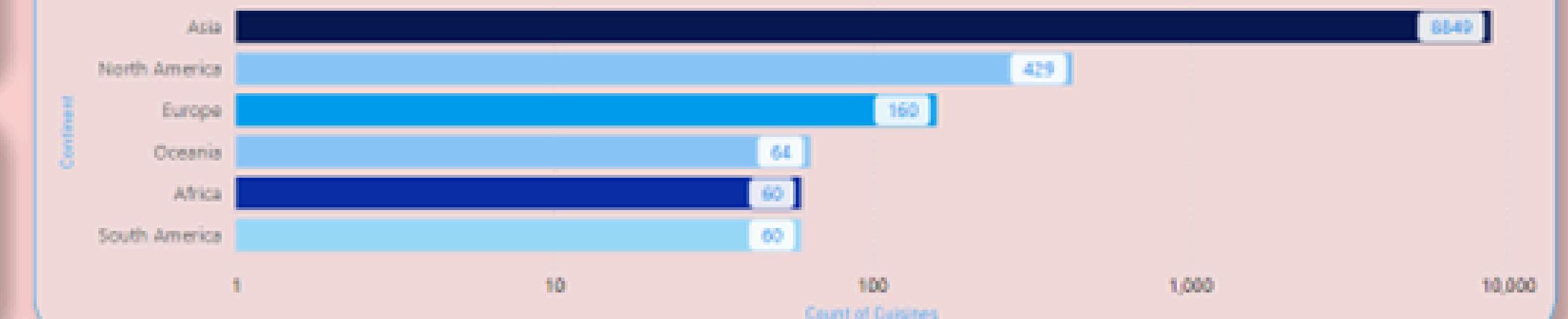
Continent	Count of Restaurant
Asia	8549
North America	438
Europe	160
Oceania	64
Africa	60
South America	60
Total	9631

6
Total Continents

15
Total Countries

140
Total Cities

Count of Cuisine by Continents



Page 1

Page 2





Power BI Project 3.0



Beginner

Intermediate

Expert

WHAT I DID ?

1. Data Import:

- Imported data from all available Excel files.

2. Data Transformation:

- Basic data cleaning

3. Data Structuring:

- Columns like age group and income group were created to create age and income bins.
- Created revenue column and Week number column.

4. DAX Usage:

- age group, income group, revenue, week_number, current week revenue, previous week revenue, WOW revenue.

5. Dashboard Report:

- Created a customer and company dashboard.

Credit Card Transaction Report

TOTAL REVENUE

\$55.3M

TOTAL INTEREST

\$7.8M

TOTAL TRANSACTION

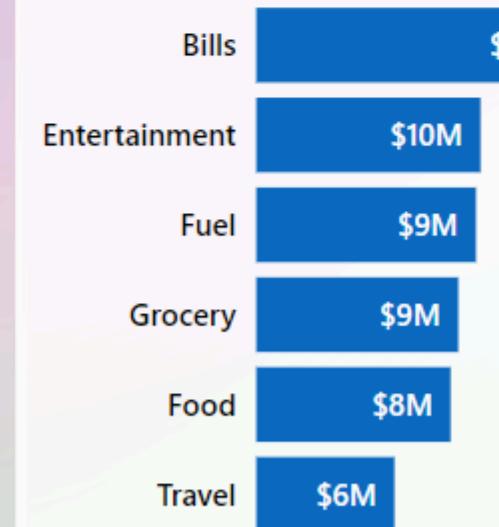
\$44.5M

Sum of Total_Trans_Vol

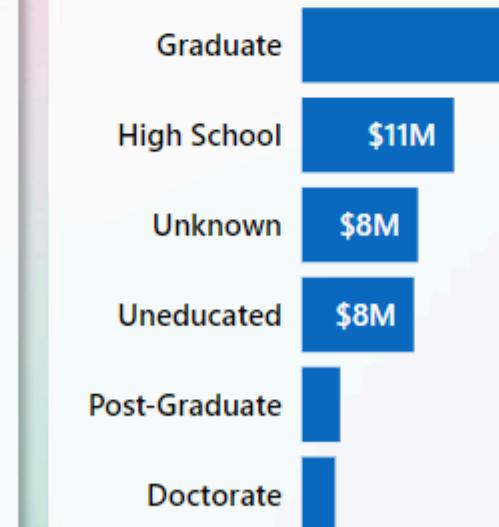
\$656K

CARDS	TOTAL REVENUE	TOTAL INTEREST ERANED	TOTAL ANNUAL FEES
Blue	\$4,61,39,397.7	\$64,95,887.7	2685635
Gold	\$24,54,072.2	\$3,73,784.2	56210
Platinum	\$11,35,608.1	\$1,61,629.1	20665
Silver	\$55,86,332.3	\$8,12,081.3	187505

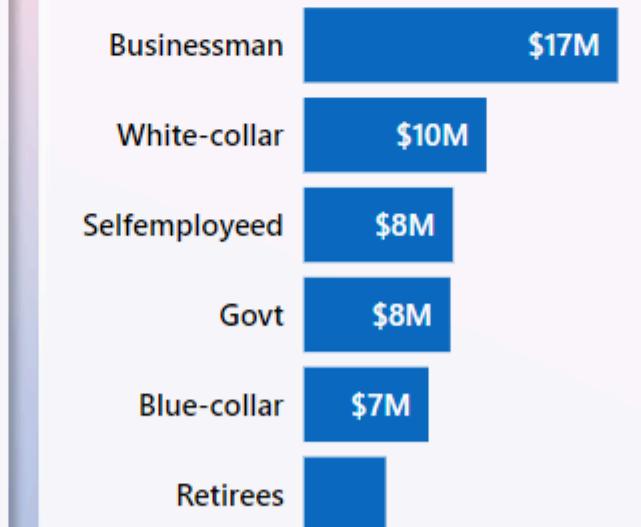
REVENUE BY EXPENDITURE TYPE



REVENUE BY EDUCATION



REVENUE BY CUSTOMER JOB



WEEK SELECTION

All

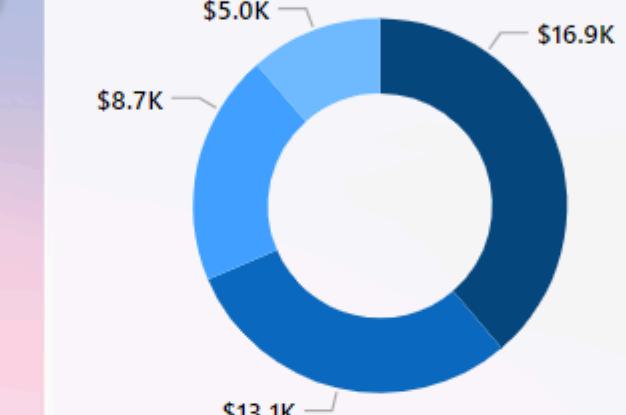
F M

MED LOW HIGH

Silver Blue Gold Platinum

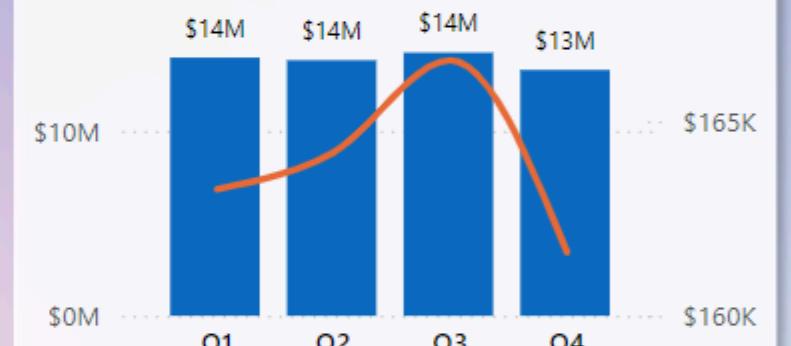
Customer Acq Cost

Cards ● Platinum ● Gold ● Silver ● Blue

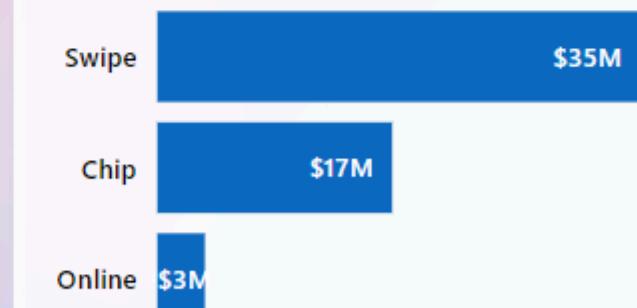


QTR REVENUE & TRANSCACTIONS COUNT

● REVENUE ● Sum of Total_Trans_Vol



REVENUE BY USE CHIPS



Credit Card Customer Report

TOTAL REVENUE

\$55.3M

TOTAL INTEREST

\$7.8M

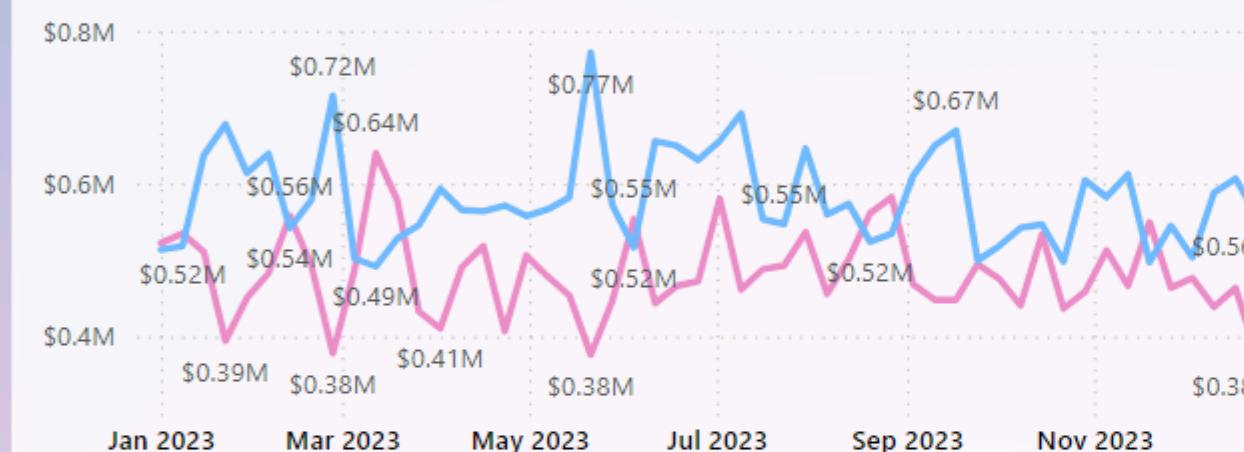
TOTAL INCOME

576M

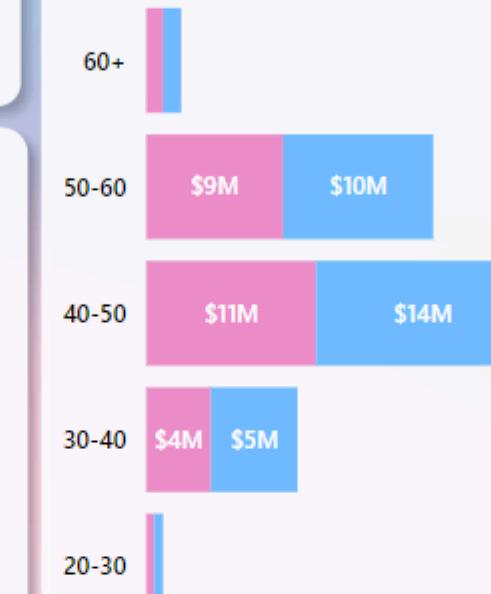
CSS

3.19

REVENUE VS GENDER

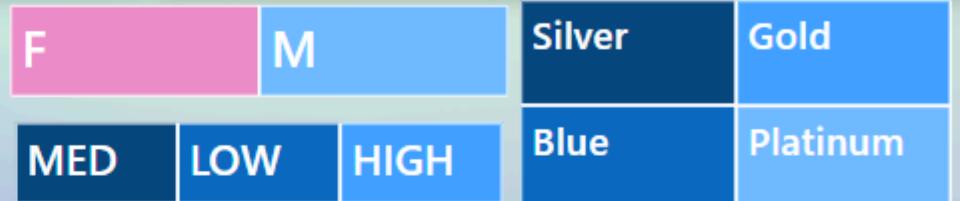


AGE GROUP



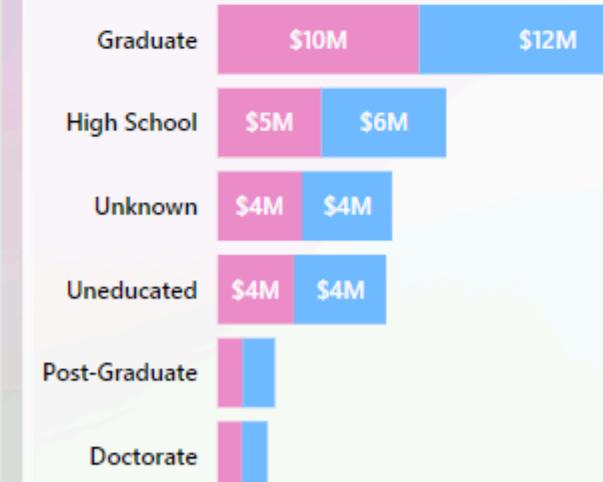
WEEK SELECTION

All

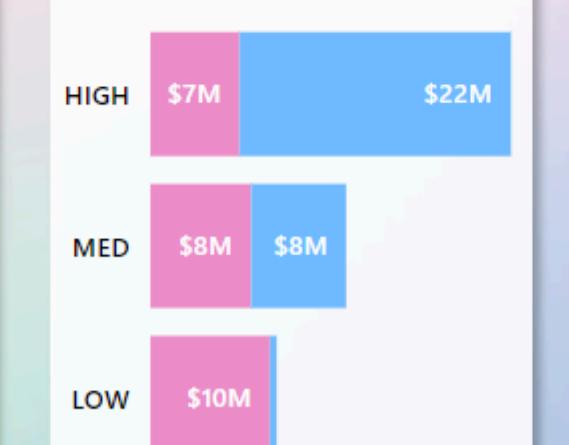


Customer_Job	REVENUE	TRANSACTION AMT	INCOME
Businessman	\$1,73,87,832.3	\$1,42,85,412.0	186959919
White-collar	\$1,01,14,655.9	\$82,21,742.0	103930055
Selfemployed	\$82,61,757.8	\$63,95,026.0	75313288
Govt	\$81,11,701.4	\$65,07,875.0	88773989
Blue-collar	\$69,04,278.6	\$54,88,838.0	72262158
Retirees	\$45,35,184.2	\$36,23,120.0	48675030
Total	\$5,53,15,410.2	\$4,45,22,013.0	575914439

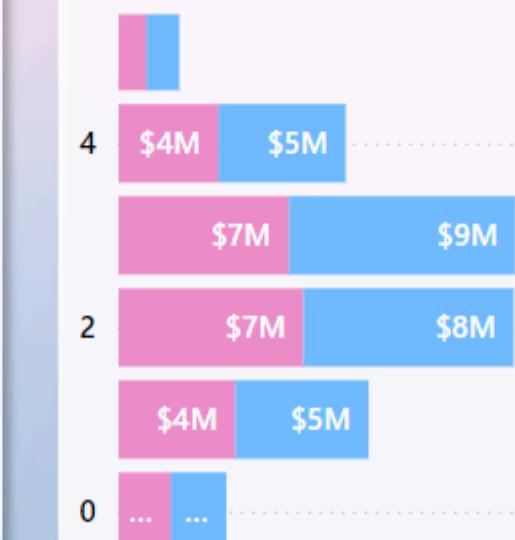
EDUCATION LEVEL



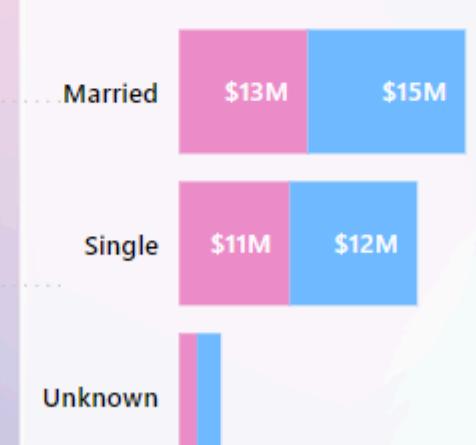
INCOME GROUP



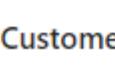
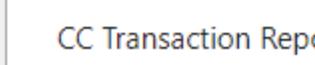
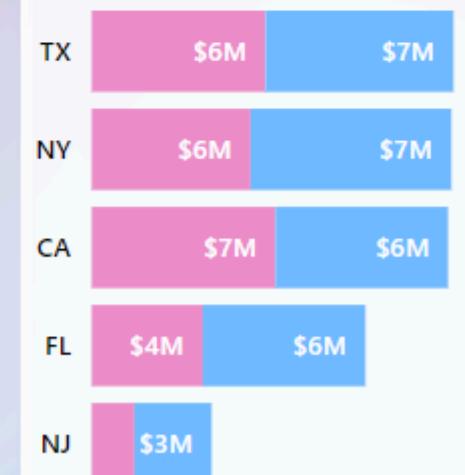
DEPENDENT COUNT



MARITAL STATUS



TOP 5 STATE





Power BI Project 4.0



Beginner

Intermediate

Expert

WHAT I DID ?

1. Data Collection:

- Gathered sales data from multiple sources such as CSV files, databases, and APIs.

2. Data Cleaning:

- Identified and handled missing values.
- Removed duplicates and irrelevant data.
- Standardized data formats (e.g., date formats, numerical values).

3. Data Transformation:

- Used Power Query to transform and shape the data.
- Created calculated columns and measures.
- Merged and appended datasets as required.

4. Data Modeling:

- Defined relationships between tables.
- Created a star schema to optimize data models.
- Established hierarchies for drill-down analysis.

5. Visualizations:

- Developed various visualizations to represent data insights effectively.
- Used charts, graphs, and maps to display sales performance.
- Created interactive dashboards with filters and slicers for user interactivity.

6. DAX Calculations:

- Implemented DAX (Data Analysis Expressions) to perform complex calculations.
- Created measures for Key Performance Indicators (KPIs) such as total sales, average sales, and growth rates.

7. Report Creation:

- Designed the report layout to ensure it is user-friendly and visually appealing.

Coffee Shop Sales dashboard

Sign in

File Home Insert Modeling View Optimize Help

Get data New visual Visual gallery New calculation Publish Copilot

COFFEE SHOP SALES Date Range : 01-01-2023 to 30-06-2023

Sales Report

FILTERS MONTH|YEAR May 2023

May 2023

Mon Tue Wed Thu Fri Sat Sun

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Hover on this visual to see the details

Sales Trend Over the Period

Avg Sales: \$5,056

Total sales \$157K ▲+31.8% | +37.8K vs LM

Total Orders 33527 ▲+32.3% | +8.2K vs LM

Total Quantity sold 48233 ▲+32.3% | +11.8K vs LM

Sales by Weekday/Weekend

Weekend \$40K (25.59%)

Weekday \$117K (74.41%)

\$157K REVENUE

Sales by Product Category

Coffee \$60.36k	▲+31.3%
Tea \$44.54k	▲+33.5%
Bakery \$18.57k	▲+32.4%
Drinking Chocolate \$16.32k	▲+33.0%
Coffee beans \$8.77k	▲+...
Branded \$2.89k	
Loose Tea \$2.40k	
Flavours \$1.91k	
Packaged Chocolate \$0.98k	

Sales by Product (TOP 10)

Barista Espresso \$20.42k	▲+31.3%
Brewed Chai tea \$17.43k	▲+31.1%
Hot chocolate \$16.32k	▲+33.0%
Gourmet brewed coffee \$15.56k	▲+31.6%
Brewed herbal tea \$10.93k	▲+35.8%
Brewed Black tea \$10.78k	▲+34.3%
Premium brewed coffee \$8.74k	▲+33.1%
Organic brewed coffee \$8.35k	▲+26.3%
Scone \$8.31k	▲+29.2%
Drip coffee \$7.29k	▲+34.6%

Sales by Hours|Days

Hour	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Total
6								\$5K
7								\$14K
8								\$19K
9								\$19K
10								\$20K
11								\$10K
12								\$9K
13								\$9K
14								\$9K
15								\$10K
16								\$9K
17								\$9K
18								\$8K
19								\$6K
20								\$1K

Dashboard Tooltip- calendar Tooltip- hours|days +

Data

Search

Date Table

- colour for bars
- Daily AVG sales
- Date
- Day_name
- Day#
- Label for Prod...
- Label for Prod...
- Label for store...
- Month
- Month NO
- Month Year
- Placeholder
- tt for hours
- Week#
- Weekday/We...
- weekday#

Transactions

- CM orders
- CM qty sold
- CM sales
- Hour
- MOM Grouth ...
- MOM Grouth ...
- MOM Grouth ...

Page 1 of 3

76%

Athletics Final result

Search

13:08 06-08-2024 ENG IN

COFFEE SHOP SALES Date Range : 01-01-2023 to 30-06-2023

Sales Report

FILTERS MONTH | YEAR May 2023

May 2023

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Hover on this visual to see the details

Sales by Weekday/Weekend

Weekend \$40K (25.59%)

Weekday \$117K (74.41%)

\$157K REVENUE

Sales by Product Category

- Coffee | \$60.36k
- Tea | \$44.54k
- Bakery | \$18.57k
- Drinking Chocolate | \$16.32k
- Coffee beans | \$8.77k
- Branded | \$2.89k
- Loose Tea | \$2.40k
- Flavours | \$1.91k
- Packaged Chocolate | \$0.98k

Sales by Store Location

- Hell's Kitchen | \$52.60k
- Astoria | \$52.43k
- Lower Manhattan | \$51.70k

▲+30.5% | +12.3K vs LM

▲+32.8% | +13.1K vs LM

▲+32.0% | +12.5K vs LM

Total sales

\$157K

▲+31.8% | +37.8K vs LM

Total Orders

33527

▲+32.3% | +8.2K vs LM

Total Quantity sold

48233

▲+32.3% | +11.8K vs LM

Sales Trend Over the Period

AVG SALES: \$5,056

\$0K \$5K

07 May 28 May

Total sales **\$3K**

Total Orders **612**

Total Quantity sold **874**

Tue Hour# : 8

Orders Qty Sales

Sales by Hours|Days

Tue	Wed	Thu	Fri	Sat	Sun	Total
\$25K	\$25K	\$20K	\$20K	\$21K	\$19K	\$140K

Brewed Black tea | \$10.78k

Premium brewed coffee | \$8.74k

Organic brewed coffee | \$8.35k

Scone | \$8.31k

Drip coffee | \$7.29k

10
11
12
13
14
15
16
17
18
19
20

Data

Search

✓ Date Table

-  colour for bars
 -  Daily AVG sales
 -  Date
 -  Day_name
 -  Day#
 -  Label for Prod...
 -  Label for Prod...
 -  Label for store...
 -  Month
 -  Month NO
 -  Month Year
 -  Placeholder
 -  tt for hours
 -  Week#
 -  Weekday/We...
 -  weekday#

✓ Transactions

-  CM orders
 -  CM qty sold
 -  CM sales
 -  Hour
 -  MOM Grouth ...
 -  MOM Grouth ...
 -  MOM Grouth ...



MySQL Project 1.0



Beginner

Intermediate

Expert

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File*

```
1 • Create database if not exists music_store_sales;
2 • use music_store_sales;
3
4 -- QUESTION SET 1
5
6 -- Q1: Who is the senior most employee based on job title?
7
8 • select * from employee;
9
10 • select * from employee
11 order by levels desc
12 limit 1 ;
13
14 • insert into employee (employee_id, last_name, first_name, title, levels ,address, city,
15 state, country, postal_code,email)
16 value (9,"Madan","Mohan","Senior General Manager","L7", "1008 Vrinda Ave MT",
17 "Edmonton","AB","Canada","T5K 2N1","madan.mohan@chinookcorp.com");
18
19 • select * from employee
20 order by levels desc
21 limit 1 ;
22
23 -- Q2: Which countries have the most Invoices?
24 • select * from invoice;
25
26 • select billing_country , count(invoice_id) as Total_invoice
from invoice
group by 1
order by 2 desc
27
28
29
```

No object selected

Output:

27°C Mostly cloudy

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File*

```
31
32 • select billing_country, count(billing_country) as Invoice_Count
from invoice
33 group by billing_country
34 order by Invoice_Count desc;
35
36
37 -- Q3: Which countries have the 2nd most Invoices?
38 • select billing_country, count(billing_country) as Invoice_Count
from invoice
39 group by billing_country
40 order by Invoice_Count desc
41 limit 1 offset 1;
42
43
44 -- Q4: What are top 3 values of total invoice?
45 • select * from invoice;
46
47 • select total , invoice_id
from invoice
48 order by total desc
49 limit 3;
50
51
52 • select total from invoice
group by total
53 order by total desc
54 limit 3;
55
56
57
58 /* Q5: Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money.
59 Write a query that returns one city that has the highest sum of invoice totals.
60 Return both the city name & sum of all invoice totals */
61
62 • select billing_city , sum(total) as total_sales
from invoice
63 group by 1
64 order by 2 desc
65 limit 1;
66
67
68 • select * from invoice;
69
70 • select billing_city, sum(total) as total_sales
from invoice
71 group by billing_city
72 order by total_sales desc
73 limit 1;
74
75
76 /* Q6: Who is the best customer? The customer who has spent the most money will be declared the best customer.
77 Write a query that returns the person who has spent the most money.*/
78
79 • select * from customer ;
80
81 • select c.customer_id , c.first_name , c.last_name , sum(i.total) as total_spent
from customer c
join invoice i on i.customer_id = c.customer_id
82 group by 1,2,3
83 order by 4 desc
84 limit 1;
85
86
87
88 • select c.customer_id , c.first_name , c.last_name , sum(invoice.total) as total_sales
from customer c
join invoice on c.customer_id = invoice.customer_id
89 group by c.customer_id , c.first_name , c.last_name
90 order by total_sales desc
91 limit 1;
92
93
94 -- Question set 2
95
96
97 -- Q1. Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
98 -- Return your list ordered alphabetically by email starting with A.
99 • select * from customer ;
100 • select c.email, c.first_name , c.last_name , g.name from customer c
join invoice i on i.customer_id = c.customer_id
101
```

No object selected

Output:

27°C Mostly cloudy

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File*

```
52 • select total from invoice
53 group by total
54 order by total desc
55 limit 3;
56
57
58 /* Q5: Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money.
59 Write a query that returns one city that has the highest sum of invoice totals.
60 Return both the city name & sum of all invoice totals */
61
62 • select billing_city , sum(total) as total_sales
from invoice
63 group by 1
64 order by 2 desc
65 limit 1;
66
67
68 • select * from invoice;
69
70 • select billing_city, sum(total) as total_sales
from invoice
71 group by billing_city
72 order by total_sales desc
73 limit 1;
74
75
76 /* Q6: Who is the best customer? The customer who has spent the most money will be declared the best customer.
77 Write a query that returns the person who has spent the most money.*/
78
79 • select * from customer ;
80
81 • select c.customer_id , c.first_name , c.last_name , sum(i.total) as total_spent
from customer c
join invoice i on i.customer_id = c.customer_id
82 group by 1,2,3
83 order by 4 desc
84 limit 1;
85
86
87
88 • select c.customer_id , c.first_name , c.last_name , sum(invoice.total) as total_sales
from customer c
join invoice on c.customer_id = invoice.customer_id
89 group by c.customer_id , c.first_name , c.last_name
90 order by total_sales desc
91 limit 1;
92
93
94 -- Question set 2
95
96
97 -- Q1. Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
98 -- Return your list ordered alphabetically by email starting with A.
99 • select * from customer ;
100 • select c.email, c.first_name , c.last_name , g.name from customer c
join invoice i on i.customer_id = c.customer_id
101
```

No object selected

Output:

27°C Mostly cloudy

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File*

```
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
```

No object selected

Output:

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File* x

```
94 -- Question set 2
95
96 -- Q1. Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
97 -- Return your list ordered alphabetically by email starting with A.
98
99 • select * from customer ;
100 • select c.email, c.first_name , c.last_name , g.name from customer c
101 join invoice i on i.customer_id = c.customer_id
102 join invoice_line inv on inv.invoice_id = i.invoice_id
103 join track t on t.track_id = inv.track_id
104 join genre g on g.genre_id = t.genre_id
105 where g.name = "Rock"
106 order by 1;
107
108 • select distinct c.email, c.first_name , c.last_name, g.name from customer c
109 join invoice i on i.customer_id = c.customer_id
110 join invoice_line inv on inv.invoice_id = i.invoice_id
111 join track t on t.track_id = inv.track_id
112 join genre g on g.genre_id = t.genre_id
113 where g.name like "Rock"
114 order by c.email;
115
116 • select distinct c.email as email, c.first_name , c.last_name from customer c
117 join invoice i on i.customer_id = c.customer_id
118 join invoice_line inv on inv.invoice_id = i.invoice_id
119 where track_id in (
120     select t.track_id from track
121     join genre g on g.genre_id = t.genre_id
122     where g.name like "Rock")
```

No object selected

Output:

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File* x

```
121
122     join genre g on g.genre_id = t.genre_id
123     where g.name like "Rock"
124
125     order by c.email desc;
126
127 -- Q2. Let's invite the artists who have written the most rock music in our dataset.
128 -- Write a query that returns the Artist name and total track count of the top 10 rock bands.
129
130 • select * from track;
131
132 • select a.artist_id, a.name , count(t.track_id) as total_tracks  from artist a
133 join album2 al on al.artist_id = a.artist_id
134 join track t on t.album_id = al.album_id
135 join genre g on g.genre_id = t.genre_id
136 where g.name like "Rock"
137
138 group by a.artist_id, a.name
139 order by total_tracks desc
140 limit 10;
```

No object selected

Output:

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File* x

```
141
142
143 JOIN genre ON genre.genre_id = track.genre_id
144 WHERE genre.name LIKE 'Rock'
145 GROUP BY artist.artist_id , artist.name
146 ORDER BY number_of_songs DESC
147
148
149
```

No object selected

Output:

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File* x

```
148
149
150 -- Q3: Return all the track names that have a song length longer than the average song length.
151 -- Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.
152
153 • select * from track ;
154 • select name , milliseconds from track
155 where milliseconds > ( select avg(milliseconds) as avg_milli from track)
156 order by milliseconds desc;
157
158 -- Question set 3
159
160 -- Q1. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent
161 -- first find top selling artist and then use this artist to find top spending customer for this artist.
162
163 • with best_selling_artist as
164 (select a.artist_id , a.name , sum(inv.unit_price * inv.quantity) as total_sales from artist a
165 join album2 al on a.artist_id = al.artist_id
166 join track t on t.album_id = al.album_id
167 join invoice_line inv on inv.track_id = t.track_id
168 group by 1,2
169 order by total_sales desc
170 limit 1)
171 select c.first_name , c.last_name , bsa.name , sum(inv.unit_price * inv.quantity) as Amount_spent from customer c
172 join invoice i on i.customer_id =c.customer_id
173 join invoice_line inv on inv.invoice_id = i.invoice_id
174 join track t on t.track_id = inv.track_id
175 join album2 al on al.album_id = t.album_id
176 join best_selling_artist bsa on bsa.artist_id = al.artist_id
```

No object selected

Output:

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator SQL File* x

```
177
178     order by amount_spent desc
179     limit 5;
180
181 -- Q2: We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre
182 -- with the highest amount of purchases. Write a query that returns each country along with the top genre. For countries where
183 -- the maximum number of purchases is shared return all Genres.
184
185 • with popular_genre as (
186 select c.country , g.name , count(inv.quantity) as Quantuity_sold ,
187 row_number() over (partition by c.country order by count(inv.quantity) desc) as row_no
188 from invoice i
189 join customer c on c.customer_id = i.customer_id
190 join invoice_line inv on inv.invoice_id = i.invoice_id
191 join track t on t.track_id = inv.track_id
192 join genre g on g.genre_id = t.genre_id
193 group by 1,2
194 order by 1 asc , 3 desc)
195 select * from popular_genre
196 where row_no <=1;
197
198
199 • WITH popular_genre AS
200 (
201     SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
202     ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
203     FROM invoice_line
204     JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
205     JOIN customer ON customer.customer_id = invoice.customer_id
206     JOIN track ON track.track_id = invoice_line.track_id
```

No object selected

Output:

27°C Mostly cloudy



MySQL Project 2.0



Beginner

Intermediate

Expert

WHAT I DID ?

1. Database and Data Import:

- Created an `employee` database.
- Imported `data_science_team.csv`, `proj_table.csv`, and `emp_record_table.csv` into the database.

2. ER Diagram:

- Created an ER diagram for the employee database.

3. Basic Queries:

- Retrieved EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table.
- Retrieved EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING with conditions on EMP_RATING.
- Concatenated FIRST_NAME and LAST_NAME of employees in the Finance department as NAME.
- Listed employees with subordinates and counted reporters.
- Listed employees from healthcare and finance departments using UNION.

4. Advanced Queries:

- Grouped employees by department with EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING, including max rating.
- Calculated minimum and maximum salary by role.
- Assigned ranks to employees based on experience.
- Created a view for employees with salary > 6000 in various countries.
- Used nested query to find employees with > 10 years of experience.
- Created a stored procedure for employees with > 3 years of experience.
- Used a stored function to validate job profiles of data science team.
- Created an index to optimize search for FIRST_NAME 'Eric'.

5. Bonus and Salary Calculations:

- Calculated employee bonuses using the formula: 5% of salary * employee rating.
- Calculated average salary distribution by continent and country.

MySQL Workbench

Local instance MySQL80 × MySQL Model (PROJECT 1.mwb) × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- employee
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - GetJobProfile
- sys

CEP-1 ×

```

1 -- Creating a database.
2 * create database if not exists Employee;
3 * use Employee;
4 * select database();
5
6
7 -- Importing the necessary dataset.
8 * show tables;
9 * desc data_science_team;
10 * select * from data_science_team;
11 * desc emp_record_table;
12 * select * from emp_record_table;
13 * desc proj_table;
14 * select * from proj_table;
15
16
17 -- changing data type, correcting some values and assigning primary and foreign keys.
18 * alter table emp_record_table
19   modify emp_id char(4) primary key,
20     modify proj_id char(4);

```

Administration Schemas

No object selected

Output

#	Time	Action	Message	Duration / Fetch
49	14:13:13	SELECT	continent, country, AVG(salary) AS average_salary FROM emp_record_table GROUP BY...	0.000 sec / 0.000 sec
50	14:15:07	SELECT	CONTINENT, COUNTRY, AVG(SALARY) AS average_salary FROM emp_record_table ...	7 row(s) returned 0.000 sec / 0.000 sec

Action Output

#	Time	Action	Message	Duration / Fetch
49	14:13:13	SELECT	continent, country, AVG(salary) AS average_salary FROM emp_record_table GROUP BY...	0.000 sec / 0.000 sec
50	14:15:07	SELECT	CONTINENT, COUNTRY, AVG(SALARY) AS average_salary FROM emp_record_table ...	7 row(s) returned 0.000 sec / 0.000 sec

Object Info Session

SQL script saved to 'C:\Users\Priya\OneDrive\Desktop\Bhavin Data course\SQL\Project\CEP-1.sql'

90°F Partly sunny

MySQL Workbench

Local instance MySQL80 × MySQL Model (PROJECT 1.mwb) × EER Diagram ×

File Edit View Arrange Model Database Tools Scripting Help

Navigator

SCHEMAS

- employee
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - GetJobProfile
- sys

CEP-1 ×

```

21
22 * alter table data_science_team
23   modify emp_id char(4) primary key;
24
25 * alter table proj_table
26   modify column PROJECT_ID char(4) primary key;
27
28 * alter table emp_record_table
29   modify column proj_id char(4);
30
31 * update emp_record_table set
32   proj_id = null
33 where emp_id = 'E260';
34
35 * alter table emp_record_table
36   add foreign key fk_proj_id (proj_id) references proj_table(project_id);
37
38 * alter table data_science_team
39   add foreign key fk_emp_id(emp_id) references emp_record_table(emp_id);
40

```

Administration Schemas

No object selected

Output

#	Time	Action	Message	Duration / Fetch
49	14:13:13	SELECT	continent, country, AVG(salary) AS average_salary FROM emp_record_table GROUP BY...	0.000 sec / 0.000 sec
50	14:15:07	SELECT	CONTINENT, COUNTRY, AVG(SALARY) AS average_salary FROM emp_record_table ...	7 row(s) returned 0.000 sec / 0.000 sec

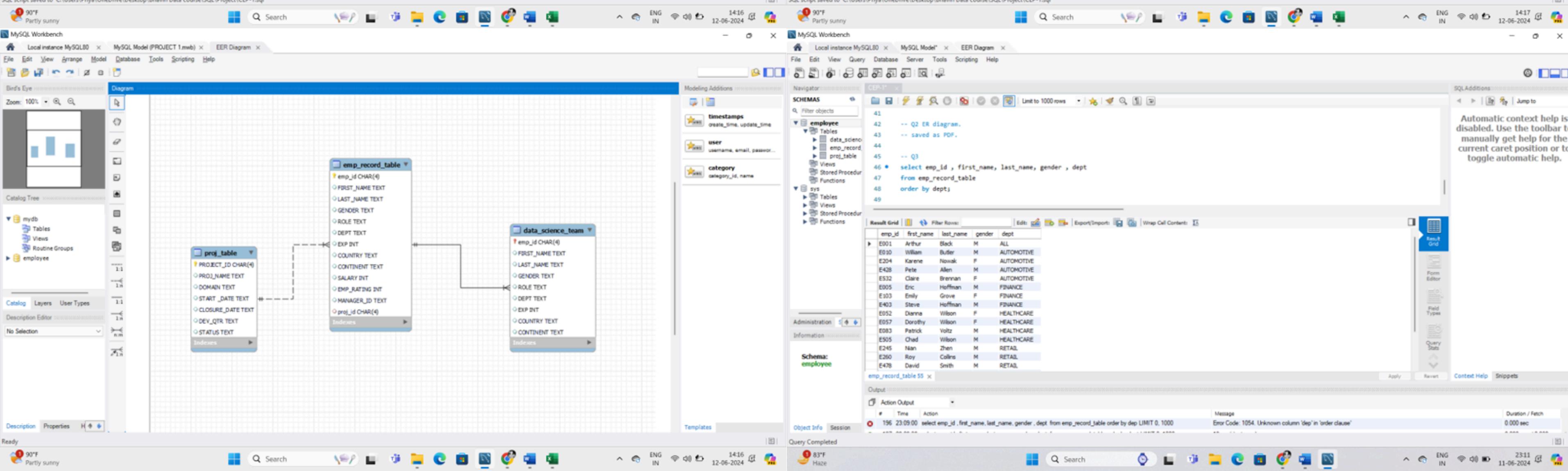
Action Output

#	Time	Action	Message	Duration / Fetch
49	14:13:13	SELECT	continent, country, AVG(salary) AS average_salary FROM emp_record_table GROUP BY...	0.000 sec / 0.000 sec
50	14:15:07	SELECT	CONTINENT, COUNTRY, AVG(SALARY) AS average_salary FROM emp_record_table ...	7 row(s) returned 0.000 sec / 0.000 sec

Object Info Session

SQL script saved to 'C:\Users\Priya\OneDrive\Desktop\Bhavin Data course\SQL\Project\CEP-1.sql'

90°F Partly sunny



MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

employee

Tables

Views

Stored Procedure

Functions

sys

Tables

Views

Stored Procedure

Functions

CEP-1* ×

SQLAdditions

Navigator

SCHEMAS

employee

Tables

Views

Stored Procedure

Functions

sys

Tables

Views

Stored Procedure

Functions

CEP-1* ×

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows | Export/Import | Wrap Cell Content

```
46 • select emp_id , first_name , last_name , gender , dept  
47 from emp_record_table  
48 order by dept;  
49  
50 -- Q4  
51 • select emp_id , first_name , last_name , gender , dept , emp_rating  
52 from emp_record_table  
53 where emp_rating <2 ;  
54  
55 • select emp_id , first_name , last_name , gender , dept , emp_rating  
56 from emp_record_table  
57 where emp_rating >4 ;  
58
```

Result Grid | Filter Rows | Export/Import | Wrap Cell Content

emp_id	first_name	last_name	gender	dept	emp_rating
E057	Dorothy	Wilson	F	HEALTHCARE	1
E532	Clare	Brennan	F	AUTOMOTIVE	1
E620	Katrina	Allen	F	RETAIL	1
E003	Miss	Miss	Miss	Miss	Miss

Administration Information

Schema: employee

emp_record_table \$8 ×

Action Output

#	Time	Action	Message	Duration / Fetch
199	23:16:50	select emp_id , first_name , last_name , gender , dept , emp_rating from emp_record_table where emp_rating >4 LIMIT 0,1000	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

83°F Haze

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

employee

Tables

Views

Stored Procedure

Functions

sys

Tables

Views

Stored Procedure

Functions

CEP-1* ×

SQLAdditions

Navigator

SCHEMAS

employee

Tables

Views

Stored Procedure

Functions

sys

Tables

Views

Stored Procedure

Functions

CEP-1* ×

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows | Export/Import | Wrap Cell Content

```
50 -- Q4  
51 • select emp_id , first_name , last_name , gender , dept , emp_rating  
52 from emp_record_table  
53 where emp_rating <2 ;  
54  
55 • select emp_id , first_name , last_name , gender , dept , emp_rating  
56 from emp_record_table  
57 where emp_rating >4 ;  
58
```

Result Grid | Filter Rows | Export/Import | Wrap Cell Content

emp_id	first_name	last_name	gender	dept	emp_rating
E001	Arthur	Black	M	ALL	5
E052	Donna	Wilson	F	HEALTHCARE	5
E083	Patrick	Voltz	M	HEALTHCARE	5
E204	Karen	Nowak	F	AUTOMOTIVE	5
E003	Miss	Miss	Miss	Miss	Miss

Administration Information

Schema: employee

emp_record_table \$9 ×

Action Output

#	Time	Action	Message	Duration / Fetch
200	23:17:23	select emp_id , first_name , last_name , gender , dept , emp_rating from emp_record_table where emp_rating <2 LIMIT 0,1000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

83°F Haze

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

employee

Tables

Views

Stored Procedure

Functions

sys

Tables

Views

Stored Procedure

Functions

CEP-1* ×

SQLAdditions

Navigator

SCHEMAS

employee

Tables

Views

Stored Procedure

Functions

sys

Tables

Views

Stored Procedure

Functions

CEP-1* ×

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows | Export/Import | Wrap Cell Content

```
54  
55 • select emp_id , first_name , last_name , gender , dept , emp_rating  
56 from emp_record_table  
57 where emp_rating >4 ;  
58  
59 • select emp_id , first_name , last_name , gender , dept , emp_rating  
60 from emp_record_table  
61 where emp_rating between 2 and 4 ;  
62
```

Result Grid | Filter Rows | Export/Import | Wrap Cell Content

emp_id	first_name	last_name	gender	dept	emp_rating
E005	Eric	Hoffman	M	FINANCE	3
E010	Willan	Butler	M	AUTOMOTIVE	2
E103	Emily	Grove	F	FINANCE	4
E245	Nan	Zhen	M	RETAIL	2
E260	Roy	Collins	M	RETAIL	3
E403	Steve	Hoffman	M	FINANCE	3
E428	Pete	Allen	M	AUTOMOTIVE	4
E478	David	Smith	M	RETAIL	4
E505	Chad	Wilson	M	HEALTHCARE	2
E583	Janet	Hale	F	RETAIL	2
E612	Tracy	Norris	F	RETAIL	4
E640	Jenifer	Jones	F	RETAIL	4
E003	Miss	Miss	Miss	Miss	Miss

Administration Information

Schema: employee

emp_record_table \$60 ×

Action Output

#	Time	Action	Message	Duration / Fetch
201	23:17:36	select emp_id , first_name , last_name , gender , dept , emp_rating from emp_record_table where emp_rating >4 LIMIT 0,1000	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

83°F Haze

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

employee

Tables

Views

Stored Procedure

Functions

sys

Tables

Views

Stored Procedure

Functions

CEP-1* ×

SQLAdditions

Navigator

SCHEMAS

employee

Tables

Views

Stored Procedure

Functions

sys

Tables

Views

Stored Procedure

Functions

CEP-1* ×

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows | Export/Import | Wrap Cell Content

```
60 from emp_record_table  
61 where emp_rating between 2 and 4 ;  
62  
63  
64 -- Q5  
65 • select concat(first_name,' ',last_name) as NAME  
66 from emp_record_table  
67 where dept = 'FINANCE';  
68
```

Result Grid | Filter Rows | Export/Import | Wrap Cell Content

NAME
Eric Hoffman
Emily Grove
Steve Hoffman

Administration Information

Schema: employee

Result 64 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
205	23:21:18	select distinct dept from emp_record_table LIMIT 0,1000	5 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

Query Completed

83°F Haze

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1* ×

Schemas

employee
Tables
Views
Stored Procedure
Functions
sys
Tables
Views
Stored Procedure
Functions

```

70 -- Q6
71 * SELECT
72   e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE,
73   COUNT(r.EMP_ID) AS number_of_reporters
74   FROM
75     emp_record_table e
76   LEFT JOIN
77     emp_record_table r ON e.EMP_ID = r.MANAGER_ID
78   GROUP BY
79     e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE
80   HAVING
81     number_of_reporters > 0;
82

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	number_of_reporters
E001	Arthur	Black	PRESIDENT	5
E083	Patrick	Voltz	MANAGER	3
E103	Emily	Grove	MANAGER	2
E408	Pete	Allen	MANAGER	3
E503	Janet	Hole	MANAGER	3
E612	Tracy	Norris	MANAGER	2

Administration

Information

Schema: employee

Result 65 ×

Action Output

#	Time	Action	Message	Duration / Fetch
206	23:21:27	select concat(first_name,'last_name) as NAME from emp_record_table where dept = 'FINANCE' LIMIT 0,1000	3 row(s) returned	0.000 sec / 0.000 sec

Query Completed

83°F Haze

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1* ×

Schemas

employee
Tables
Views
Stored Procedure
Functions
sys
Tables
Views
Stored Procedure
Functions

```

83
84 -- Q7
85 * SELECT
86   EMP_ID, FIRST_NAME, LAST_NAME, GENDER,
87   ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP_RATING, MANAGER_ID, PROJ_ID
88   FROM emp_record_table
89   WHERE DEPT = 'Healthcare'
90
91 UNION
92
93 * SELECT
94   EMP_ID, FIRST_NAME, LAST_NAME, GENDER,
95   ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP_RATING, MANAGER_ID, PROJ_ID
96   FROM emp_record_table
97   WHERE DEPT = 'Finance'

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5	E083	P103
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	
E050	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E083	P103
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E103	P105
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4	E001	
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E103	P105

Administration

Information

Schema: employee

Result 66 ×

Action Output

#	Time	Action	Message	Duration / Fetch
207	23:27:28	SELECT e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE, COUNT(e.EMP_ID) AS number_of_r...	6 row(s) returned	0.000 sec / 0.000 sec

Query Completed

83°F Haze

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1* ×

Schemas

employee
Tables
Views
Stored Procedure
Functions
sys
Tables
Views
Stored Procedure
Functions

```

91
92 * SELECT
93   EMP_ID, FIRST_NAME, LAST_NAME, GENDER,
94   ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP_RATING, MANAGER_ID, PROJ_ID
95   FROM emp_record_table
96   WHERE DEPT = 'Finance'
97
98 -- Q8
99 * select emp_id , first_name , last_name , role , dept , emp_rating ,
100 max(emp_rating) over (partition by dept) as dept_max_rating
101   from emp_record_table;
102

```

Result Grid

emp_id	first_name	last_name	role	dept	emp_rating	dept_max_rating
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	1	5
E083	Patrick	Voltz	MANAGER	HEALTHCARE	5	5
E050	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
E245	Nan	Zhen	SENIOR DATA SCIENTIST	RETAIL	2	4
E260	Roy	Collins	SENIOR DATA SCIENTIST	RETAIL	3	4
E478	David	Smith	ASSOCIATE DATA SCIENTIST	RETAIL	4	4
E503	Janet	Hale	MANAGER	RETAIL	2	4
E612	Tracy	Norris	MANAGER	RETAIL	4	4
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	1	4

Administration

Information

Schema: employee

Result 69 ×

Action Output

#	Time	Action	Message	Duration / Fetch
208	23:31:57	SELECT EMP_ID,FIRST_NAME,LAST_NAME,GENDER, ROLE,DEPT,EXP,COUNTRY,CONTINENT,S...	7 row(s) returned	0.000 sec / 0.000 sec
209	23:33:48	SELECT EMP_ID,FIRST_NAME,LAST_NAME,GENDER, ROLE,DEPT,EXP,COUNTRY,CONTINENT,S...	7 row(s) returned	0.000 sec / 0.000 sec
210	23:33:48	SELECT EMP_ID,FIRST_NAME,LAST_NAME,GENDER, ROLE,DEPT,EXP,COUNTRY,CONTINENT,S...	7 row(s) returned	0.000 sec / 0.000 sec

Query Completed

83°F Haze

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1* ×

Schemas

employee
Tables
Views
Stored Procedure
Functions
sys
Tables
Views
Stored Procedure
Functions

```

102
103 -- Q9
104 * SELECT
105   ROLE,
106   MIN(SALARY) AS min_salary,
107   MAX(SALARY) AS max_salary
108   FROM
109   emp_record_table
110   GROUP BY
111   ROLE
112
113

```

Result Grid

ROLE	min_salary	max_salary
PRESIDENT	16500	16500
LEAD DATA SCIENTIST	8500	9000
SENIOR DATA SCIENTIST	5500	7700
MANAGER	8500	11000
ASSOCIATE DATA SCIENTIST	4000	5000
JUNIOR DATA SCIENTIST	2800	3000

Administration

Information

Schema: employee

Result 70 ×

Action Output

#	Time	Action	Message	Duration / Fetch
212	23:37:51	select emp_id , first_name , last_name , role , dept , emp_rating , max(emp_rating) over (partition by dept) as dept_max...	19 row(s) returned	0.000 sec / 0.000 sec
213	23:45:19	SELECT ROLE, MIN(SALARY) AS min_salary, MAX(SALARY) AS max_salary FROM emp_record_table G...	6 row(s) returned	0.015 sec / 0.000 sec

Query Completed

83°F Haze

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

employee

- Tables
 - data_sceno
 - emp_record
 - proj_table
- Views
- Stored Procedure
- Functions

sys

- Tables
- Views
- Stored Procedure
- Functions

CEP-1* ×

```

114 -- Q10
115 • SELECT
116   EMP_ID,
117   FIRST_NAME,
118   LAST_NAME,
119   ROLE,
120   DEPT,
121   EXP,
122   RANK() OVER (ORDER BY EXP DESC) AS experience_rank,
123   dense_rank() over (order by exp) as exp_cont_rank
124   FROM
125     emp_record_table;
  
```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	experience_rank	exp_cont_rank
E640	Jenifer	Jhones	JUNIOR DATA SCIENTIST	RETAIL	1	19	1
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	2	18	2
E478	David	Smith	ASSOCIATE DATA SCIENTIST	RETAIL	3	16	3
E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	3	16	3
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	4	15	4
E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	14	5
E052	Diana	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	6	12	6
E245	Nan	Zhen	SENIOR DATA SCIENTIST	RETAIL	6	12	6
E260	Roy	Collins	SENIOR DATA SCIENTIST	RETAIL	7	11	7

Result 72 ×

Administration

Information

Schema: employee

Object Info Session

Output

Action Output

#	Time	Action	Message	Duration / Fetch
215	23:47:57	SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EXP, RANK() OVER (ORDER BY ...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server v...	0.000 sec
216	23:48:16	SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EXP, RANK() OVER (ORDER BY ...	19 rows) returned	0.000 sec / 0.000 sec

Wrap Call Content:

Result 72 ×

Context Help Snippets

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
219	23:52:49	high_salary_employees	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server v...	0.000 sec
220	23:53:41	SELECT * FROM high_salary_employees LIMIT 0,1000	12 row(s) returned	0.000 sec / 0.000 sec

Wrap Call Content:

Result 73 ×

Context Help Snippets

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
219	23:52:49	high_salary_employees	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server v...	0.000 sec
220	23:53:41	SELECT * FROM high_salary_employees LIMIT 0,1000	12 row(s) returned	0.000 sec / 0.000 sec

Wrap Call Content:

Result 73 ×

Context Help Snippets

Read Only

MySQL Workbench

Local instance MySQL80 × MySQL Model × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

employee

- Tables
 - data_sceno
 - emp_record
 - proj_table
- Views
 - high_salary
 - Stored Procedure
 - Functions

sys

- Tables
- Views
- Stored Procedure
- Functions

CEP-1* ×

```

140
141 -- Q12
142 • SELECT
143   EMP_ID, FIRST_NAME, LAST_NAME, GENDER, EXP
144   FROM emp_record_table
145   WHERE EMP_ID IN (
146     SELECT EMP_ID
147     FROM emp_record_table
148     WHERE EXP > 10);
  
```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	EXP
E001	Arthur	Black	M	20
E005	Eric	Hoffman	M	11
E010	William	Butler	M	12
E083	Patrick	Voltz	M	15
E103	Emily	Grove	F	14
E428	Pete	Allen	M	14
E583	Janet	Hale	F	14
E612	Tracy	Norris	F	13

emp_record_table 74 ×

Administration

Information

View: high_salary_emp

Columns:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	COUNTRY	SALARY
--------	------------	-----------	------	------	---------	--------

Object Info Session

Output

Action Output

#	Time	Action	Message	Duration / Fetch
220	23:53:41	SELECT * FROM high_salary_employees LIMIT 0,1000	12 row(s) returned	0.000 sec / 0.000 sec
221	23:58:27	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, EXP FROM emp_record_table WHERE EMP_ID IN (...	8 row(s) returned	0.000 sec / 0.000 sec

Wrap Call Content:

Result 74 ×

Context Help Snippets

Read Only

SQLAdditions

Schemas

employee

- Tables
 - data_sceno
 - emp_record
 - proj_table
- Views
 - high_salary
 - Stored Procedure
 - Functions

sys

- Tables
- Views
- Stored Procedure
- Functions

CEP-1* ×

```

149
150
151 -- Q13
152 DELIMITER //
153 • CREATE PROCEDURE GetExperiencedEmployees()
154 BEGIN
155   SELECT
156     EMP_ID, FIRST_NAME, LAST_NAME, GENDER,
157     ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY
158   FROM emp_record_table
159   WHERE EXP > 3;
160 END //
161 DELIMITER ;
  
```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY
--------	------------	-----------	------	------	-----	---------	-----------	--------

high_salary_emp 75 ×

Administration

Information

View: high_salary_emp

Columns:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY
--------	------------	-----------	------	------	-----	---------	-----------	--------

Object Info Session

Output

Action Output

#	Time	Action	Message	Duration / Fetch
223	00:03:43	CREATE PROCEDURE GetExperiencedEmployees() BEGIN SELECT EMP_ID, FIRST_NAME, LAST_NAM...	Error Code: 1304. PROCEDURE GetExperiencedEmployees already exists	0.000 sec
224	00:03:45	CALL GetExperiencedEmployees()	15 row(s) returned	0.016 sec / 0.000 sec

Wrap Call Content:

Result 75 ×

Context Help Snippets

Read Only

SQLAdditions

Schemas

employee

- Tables
 - data_sceno
 - emp_record
 - proj_table
- Views
 - high_salary
 - Stored Procedure
 - Functions

sys

- Tables
- Views
- Stored Procedure
- Functions

CEP-1* ×

```

171
  
```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY
--------	------------	-----------	------	------	-----	---------	-----------	--------

high_salary_emp 76 ×

Administration

Information

View: high_salary_emp

Columns:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY
--------	------------	-----------	------	------	-----	---------	-----------	--------

Object Info Session

Output

Action Output

#	Time	Action	Message	Duration / Fetch
224	00:03:45	CALL GetExperiencedEmployees()	15 row(s) returned	0.016 sec / 0.000 sec

Wrap Call Content:

Result 76 ×

Context Help Snippets

Read Only

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80 X MySQL Model X EER Diagram X

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1*

Schemas

```

153 • CREATE PROCEDURE GetExperiencedEmployees()
154 BEGIN
155   SELECT
156     EMP_ID, FIRST_NAME, LAST_NAME, GENDER,
157     ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY
158   FROM eep_record_table
159   WHERE EXP > 3;
160 END //
161 DELIMITER ;
163 • -- USING THE STORED PROCEDURE
164 CALL GetExperiencedEmployees();

```

Result Grid | Filter Rows! Exports Wrap Cell Content:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY
E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500
E204	Karen	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500
E245	Nan	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500

Administration Schemas Information

View: **high_salary_emp**

Columns:

- EMP_ID
- FIRST_NAME
- LAST_NAME
- ROLE
- DEPT

Action Output

#	Time	Action	Message	Duration / Fetch
224	00:03:45	CALL GetExperiencedEmployees()	15 row(s) returned	0.016 sec / 0.000 sec
225	00:04:15	CALL GetExperiencedEmployees()	15 row(s) returned	0.000 sec / 0.000 sec

Object Info Session Query Completed

84°F Mostly cloudy 90°F Partly sunny 13:51 12-06-2024

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1*

Schemas

```

165
166
167 -- Q14
168 DELIMITER //
170 • CREATE FUNCTION GetJobProfile(exp tinyint) RETURNS VARCHAR(30)
171 BEGIN
172   DECLARE job_profile VARCHAR(30);
174   IF exp <= 2 THEN
175     SET job_profile = 'JUNIOR DATA SCIENTIST';
176   ELSEIF exp > 2 AND exp <= 5 THEN
177     SET job_profile = 'ASSOCIATE DATA SCIENTIST';
178   ELSEIF exp > 5 AND exp <= 10 THEN
179     SET job_profile = 'SENIOR DATA SCIENTIST';
180   ELSEIF exp > 10 AND exp <= 12 THEN
181     SET job_profile = 'LEAD DATA SCIENTIST';
182   ELSE
183     SET job_profile = 'MANAGER';
184   END IF;
185
186   RETURN job_profile;
187 END //
188 DELIMITER ;

```

Result Grid | Filter Rows! Exports Wrap Cell Content:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	
E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500
E204	Karen	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500
E245	Nan	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500

Administration Schemas Information

No object selected

Output

#	Time	Action	Message	Duration / Fetch
16	13:49:51	SET GLOBAL log_bin_trust_function_creators = 1	0 row(s) affected, 1 warning(s): 1287 @log_bin_trust_function_creators' is deprecated and will be removed.	0.000 sec
17	13:50:07	CALL GetExperiencedEmployees()	15 row(s) returned	0.000 sec / 0.000 sec
18	13:50:11	CREATE FUNCTION GetJobProfile(exp tinyint) RETURNS VARCHAR(30) BEGIN DECLARE job_profile VARCHAR(30); END //	0 row(s) affected	0.000 sec

Context Help Snippets

84°F Mostly cloudy 90°F Partly sunny 13:51 12-06-2024

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1*

Schemas

```

190 • -- CALLING THE FUNCTION
191   SELECT
192     EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EXP,
193     GetJobProfile(EXP) AS Standard_Profile,
194     CASE
195       WHEN GetJobProfile(EXP) = ROLE THEN 'MATCHING'
196       ELSE 'NOT MATCHING'
197     END AS Matching
198   FROM
199     data_science_team;

```

Result Grid | Filter Rows! Exports Wrap Cell Content:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	Standard_Profile	Matching
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	11	LEAD DATA SCIENTIST	MATCHING
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	12	LEAD DATA SCIENTIST	MATCHING
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	6	SENIOR DATA SCIENTIST	MATCHING
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	9	SENIOR DATA SCIENTIST	MATCHING
E204	Karen	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	SENIOR DATA SCIENTIST	MATCHING
E245	Nan	Zhen	SENIOR DATA SCIENTIST	RETAIL	6	SENIOR DATA SCIENTIST	MATCHING
E260	Roy	Collins	SENIOR DATA SCIENTIST	RETAIL	7	SENIOR DATA SCIENTIST	MATCHING
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	4	ASSOCIATE DATA SCIENTIST	MATCHING
E478	David	Smith	ASSOCIATE DATA SCIENTIST	RETAIL	3	ASSOCIATE DATA SCIENTIST	MATCHING
E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	ASSOCIATE DATA SCIENTIST	MATCHING
E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	3	ASSOCIATE DATA SCIENTIST	MATCHING
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	2	JUNIOR DATA SCIENTIST	MATCHING
E744	Tanner	Yann	JUNIOR DATA SCIENTIST	RETAIL	1	JUNIOR DATA SCIENTIST	MATCHING

Administration Schemas Information

No object selected

Output

#	Time	Action	Message	Duration / Fetch
20	13:50:29	CREATE FUNCTION GetJobProfile(exp tinyint) RETURNS VARCHAR(30) BEGIN DECLARE job_profile VARCHAR(30); END //	Error Code: 1304. FUNCTION GetJobProfile already exists	0.000 sec
21	13:51:36	SELECT * FROM emp_record_table WHERE first_name = 'Eric' LIMIT 0, 1000	13 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

90°F Partly sunny 13:52 12-06-2024

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1*

Schemas

```

197 END AS Matching
198 FROM
199   data_science_team;
200
201 • SET GLOBAL log_bin_trust_function_creators = 1;
202
203 -- Q15
204 -- Checking execution plan before creating the index
205
206 • SELECT * FROM emp_record_table WHERE first_name = 'Eric';
207

```

Visual Explain | Display Info Read + Eval cost Overview 

emp_record_table 13 X

Output

#	Time	Action	Message	Duration / Fetch
24	13:58:29	SELECT * FROM emp_record_table WHERE first_name = 'Eric' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
25	13:58:30	EXPLAIN SELECT * FROM emp_record_table WHERE first_name = 'Eric'	OK	0.000 sec

Object Info Session

90°F Partly sunny 13:58 12-06-2024

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1*

Schemas

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

SQLAdditions

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator

CEP-1*

Schemas

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas CEP-1* X

```

206 • SELECT * FROM emp_record_table WHERE first_name = 'Eric';
207
208 • alter table emp_record_table
209     modify column First_name varchar(30);
210
211 -- Created the index
212 • CREATE INDEX idx_emp_name ON emp_record_table(first_name);
213
214 -- Checking execution plan after creating the index
215 • SELECT * FROM emp_record_table WHERE first_name = 'Eric';
216

```

Visual Explain [Display Info] Read + Eval cost [Overview] [View Source]

Non-Unique Key Lookup
emp_record_table
idx_emp_name

Field Types Query Stats Execution Plan

Administration Schemas Information No object selected

emp_record_table 16 X

Action Output

#	Time	Action	Message	Duration / Fetch
36	14:02:16	SELECT * FROM emp_record_table WHERE first_name = 'Eric' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
37	14:02:22	EXPLAIN SELECT * FROM emp_record_table WHERE first_name = 'Eric'	OK	0.000 sec

Object Info Session

90°F Partly sunny 14:02 12-06-2024 ENG IN

MySQL Workbench Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas CEP-1* X

```

211 -- Created the index
212 • CREATE INDEX idx_emp_name ON emp_record_table(first_name);
213
214 -- Checking execution plan after creating the index
215 • SELECT * FROM emp_record_table WHERE first_name = 'Eric';
216
217 -- Q16
218 • SELECT emp_id, first_name, last_name, salary, emp_rating, (0.05 * salary * emp_rating) AS bonus
219     FROM emp_record_table;
220
221

```

Result Grid [Filter Rows] Export! Wrap Cell Content:

emp_id	first_name	last_name	salary	emp_rating	bonus
E001	Arthur	Blad	16500	5	4125.00
E005	Eric	Hoffman	8500	3	1275.00
E010	William	Buder	9000	2	900.00
E052	Dionna	Wilson	5500	5	1375.00
E057	Dorothy	Wilson	7700	1	385.00
E083	Patrick	Voltz	9500	5	2375.00
E103	Emily	Grove	10500	4	2100.00
E204	Karen	Nowak	7500	5	1875.00
E245	Nan	Zhen	6500	2	650.00
E260	Roy	Collins	7000	3	1050.00
E403	Steve	Hoffman	5000	3	750.00
E428	Pete	Allen	11000	4	2200.00
E439	David	Gwin	8000	4	1600.00

Result 22 X

Action Output

#	Time	Action	Message	Duration / Fetch
46	14:09:00	SELECT emp_id, first_name, last_name, salary, emp_rating, (0.05 * salary * emp_rating) AS bonus FROM emp_record_table;	10 row(s) returned	0.000 sec / 0.000 sec
47	14:09:04	SELECT emp_id, first_name, last_name, salary, emp_rating, (0.05 * salary * emp_rating) AS bonus FROM emp_record_table;	19 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

90°F Partly sunny 14:09 12-06-2024 ENG IN

MySQL Workbench Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas CEP-1* X

```

222
223 -- Q17
224 • SELECT
225     CONTINENT,
226     COUNTRY,
227     AVG(SALARY) AS average_salary
228     FROM
229         emp_record_table
230     GROUP BY
231     CONTINENT, COUNTRY;
232

```

Result Grid [Filter Rows] Export! Wrap Cell Content:

CONTINENT	COUNTRY	average_salary
NORTH AMERICA	USA	9440.0000
EUROPE	FRANCE	9000.0000
NORTH AMERICA	CANADA	7000.0000
EUROPE	GERMANY	7600.0000
ASIA	CHINA	6500.0000
ASIA	INDIA	6166.6667
SOUTH AMERICA	COLOMBIA	5600.0000

Administration Schemas Information No object selected

Result 24 X

Action Output

#	Time	Action	Message	Duration / Fetch
49	14:13:13	SELECT continent, country, AVG(salary) AS average_salary FROM emp_record_table GROUP BY...	7 row(s) returned	0.000 sec / 0.000 sec
50	14:15:07	SELECT CONTINENT, COUNTRY, AVG(SALARY) AS average_salary FROM emp_record_table ...	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

90°F Partly sunny 14:15 12-06-2024 ENG IN

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.



MySQL Project 3.0



Beginner

Intermediate

Expert

WHAT I DID ?

1. Designed an ER Diagram:

- Created an Entity-Relationship diagram for the airline database, identifying and illustrating the relationships between different entities such as passengers, flights, routes, and tickets.

2. Created and Managed Database Tables:

- Implemented a `route_details` table with appropriate data types and constraints including a unique constraint on `route_id` and a check constraint to ensure valid `flight_num` and positive `distance_miles`.

3. Executed Complex Queries:

- Wrote queries to retrieve specific data, such as displaying passengers who traveled on routes 01 to 25, and extracting customer details by combining information from multiple tables (`passengers_on_flights`, `customer`, `ticket_details`).

4. Performed Data Analysis and Aggregation:

- Used aggregate functions and window functions to analyze data, including calculating the number of passengers, total

revenue in business class, maximum ticket prices for each class, and grouping data based on travel distance.

5. Enhanced Database Performance:

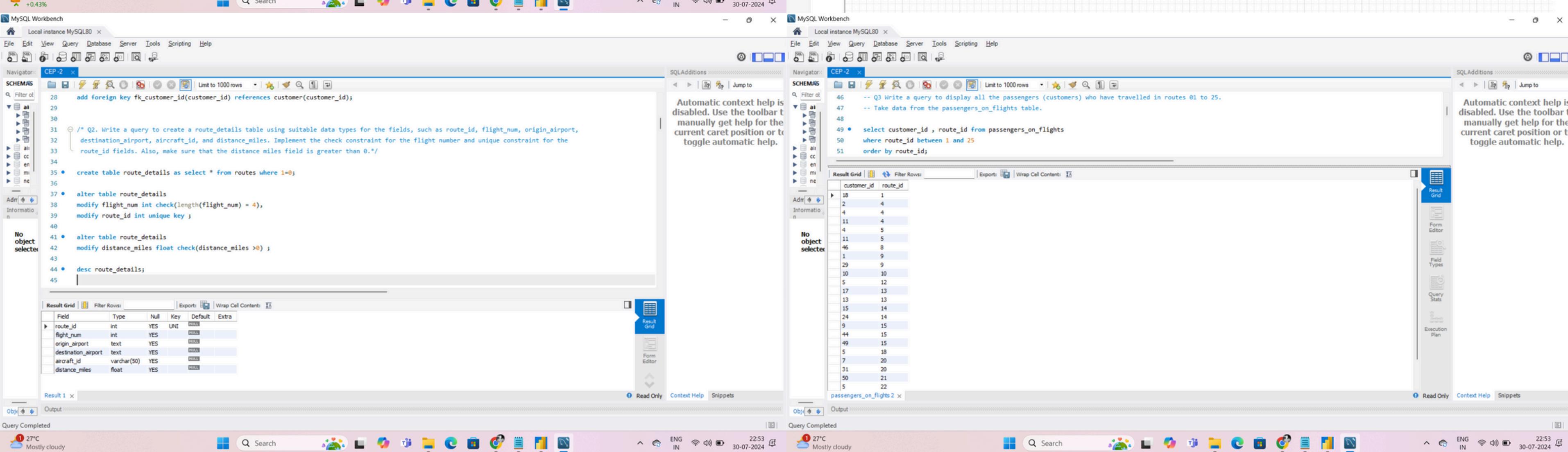
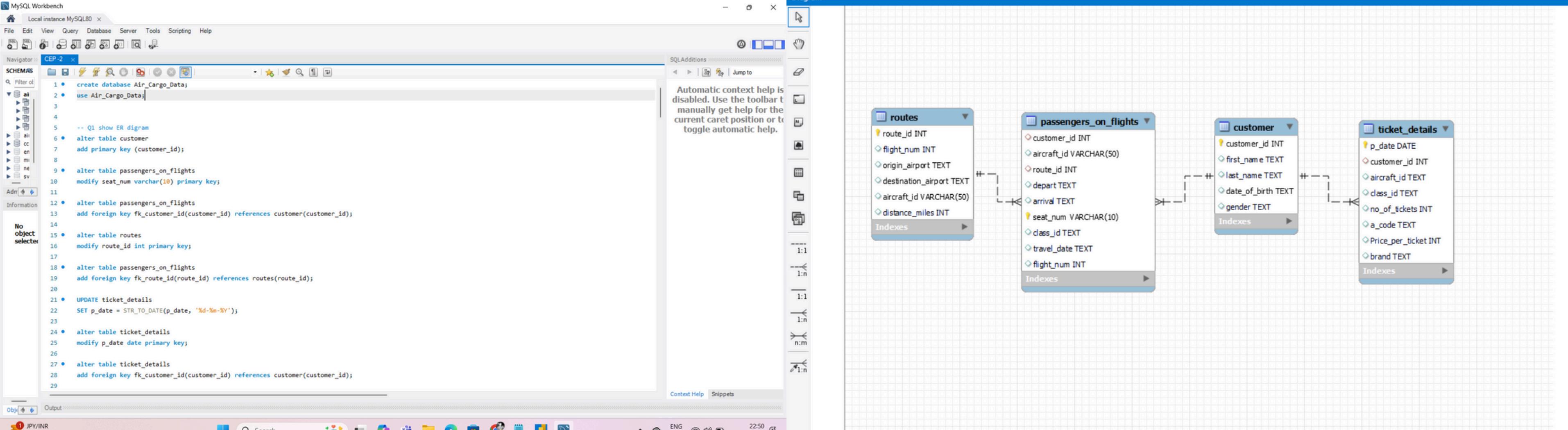
- Improved query performance for retrieving passengers on a specific route (route ID 4) by optimizing the `passengers_on_flights` table and viewing the execution plan to identify and address performance bottlenecks.

6. Implemented Stored Procedures and Functions:

- Created stored procedures to handle dynamic queries, such as retrieving passengers flying between a range of routes and categorizing travel distances. Also, implemented a stored function to determine if complimentary services are provided based on class.

7. Managed Database Security:

- Created and granted access to new users, ensuring secure and controlled database operations by managing user permissions effectively.



MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

Navigator CEP-2

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
53 -- Q4 Write a query to identify the number of passengers and total sales for each class.
54 Set limit for number of rows returned by queries.
55 • select count(*) as No_of_passengers , class_id , sum(no_of_tickets * Price_per_ticket) as total_sales
56 from ticket_details
57 where class_id = "Business"
58 group by 2;
```

Result Grid

No_of_passengers	class_id	total_sales
13	Business	6034

Result 3 x Read Only Context Help Snippets

No object selected

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

Navigator CEP-2

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
59 -- Q5 Write a query to display the full name of the customer by extracting the first name and last name from the customer table.
60
61 • select concat(first_name, " ", last_name) as full_name from customer
62
63 order by full_name;
```

Result Grid

full_name
Aaron Kim
Adam Paul
Alexander Scot
Alexis Scott
Anderson Stewart
Billy Brian
Bryan Colin
Calvin Willis
Carol Vernon
Catherina Emily
Catherine Shad
Cherly Vernon
Christine Wills
Christoper Sean
Christy Josh
Donack Dukins
Doris Walter
Du plessis Chris
Erwin Tosh
Floyd Ted
Gloria Richie
Jacqueline Keith
James Robert

Result 4 x Read Only Context Help Snippets

No object selected

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

Navigator CEP-2

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
63 order by full_name;
64
65 -- Q6. Write a query to extract the customers who have registered and booked a ticket.
66 -- Use data from the customer and ticket_details tables.
67
68 • select c.customer_id , c.first_name , c.last_name , t.no_of_tickets from customer c
69 join ticket_details t on t.customer_id = c.customer_id
70 where t.brand is not null
71 group by 1,2,3,4
72 order by 1;
```

Result Grid

customer_id	first_name	last_name	no_of_tickets
1	Julie	Sam	1
2	Steve	Ryan	1
4	Catherina	Emily	1
5	Aaron	Kim	1
7	Anderson	Stewart	1
8	Floyd	Ted	1
9	Leo	Travis	1
10	Melvin	Tracy	1
11	Roger	Walson	1
13	Solomon	Walter	1
14	Carol	Vernon	1
15	Linda	William	1
16	Christine	Willis	1
17	Catherine	Shad	1

Result 5 x Read Only Context Help Snippets

No object selected

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

Navigator CEP-2

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
72 order by 1;
73
74 -- Q7 Write a query to identify the customer's first name and
75 -- last name based on their customer ID and brand (Emirates) from the ticket_details table.
76
77 • select t.customer_id , c.first_name , c.last_name , t.brand from customer c
78 join ticket_details t on t.customer_id = c.customer_id
79 where t.brand = "Emirates"
80 order by 1;
```

Result Grid

customer_id	first_name	last_name	brand
2	Steve	Ryan	Emirates
4	Catherina	Emily	Emirates
4	Catherina	Emily	Emirates
5	Aaron	Kim	Emirates
7	Anderson	Stewart	Emirates
9	Leo	Travis	Emirates
11	Roger	Walson	Emirates
11	Roger	Walson	Emirates
14	Carol	Vernon	Emirates
18	Gloria	Richie	Emirates
18	Gloria	Richie	Emirates
19	Joyce	Paul	Emirates
25	Moss	Morris	Emirates
25	Moss	Morris	Emirates

Result 6 x Read Only Context Help Snippets

No object selected

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: CEP-2 x

SCHEMAS

No object selected

```
78 join ticket_details t on t.customer_id = c.customer_id
79 where t.brand = "Emirates"
80 order by 1;
81
82 -- Q8 Write a query to identify the customers who have travelled
83 -- by Economy Plus class using Group By and Having clause on the passengers_on_flights table.
84
85 • select customer_id , seat_num , class_id from passengers_on_flights
86 group by 1,2,3
87 having class_id = "Economy Plus";
88
89 -- Q9 Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.
90
91 • select sum(no_of_tickets*Price_per_ticket) as total_revenue ,if(sum(no_of_tickets*Price_per_ticket) > 10000 , "revenue has crossed 10000" , "revenue has not crossed 10000")
92 from ticket_details;
93
94 -- Q10 Write a query to create and grant access to a new user to perform operations on a database.
95
96 -- Create a new user and set a password
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

customer_id	seat_num	class_id
1	01EP	Economy Plus
8	02EP	Economy Plus
11	03EP	Economy Plus
17	04EP	Economy Plus
19	05EP	Economy Plus
19	06EP	Economy Plus
22	07EP	Economy Plus
32	08EP	Economy Plus
47	09EP	Economy Plus
50	10EP	Economy Plus
*	HULL	HULL

passengers_on_flights 7 x

Obj Output

Query Completed

Current temp Near record

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: CEP-2 x

SCHEMAS

No object selected

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
84
85 • select customer_id , seat_num , class_id from passengers_on_flights
86 group by 1,2,3
87 having class_id = "Economy Plus";
88
89 -- Q9 Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.
90
91 • select sum(no_of_tickets*Price_per_ticket) as total_revenue ,if(sum(no_of_tickets*Price_per_ticket) > 10000 , "revenue has crossed 10000" , "revenue has not crossed 10000")
92 from ticket_details;
93
94 -- Q10 Write a query to create and grant access to a new user to perform operations on a database.
95
96 -- Create a new user and set a password
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

total_revenue	Revenue_Status
15369	revenue has crossed 10000

Result 8 x

Obj Output

Query Completed

Current temp Near record

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: CEP-2 x

SCHEMAS

No object selected

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
92 from ticket_details;
93
94 -- Q10 Write a query to create and grant access to a new user to perform operations on a database.
95
96 -- Create a new user and set a password
97 • CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'YourStrongPassword';
98
99 -- Grant all privileges on a specific database
100 • GRANT ALL PRIVILEGES ON YourDatabase.* TO 'new_user'@'localhost';
101
102 -- Make sure the changes take effect
103 • FLUSH PRIVILEGES;
104
105 -- Q11 Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.
106
107 • with Max_price_ticket as
108   (select class_id , price_per_ticket , row_number() over (partition by class_id order by price_per_ticket desc) as row_no
109   from ticket_details)
110   select * from Max_price_ticket
111   where row_no = 1;
112
113 -- method 2
114 • select class_id , price_per_ticket , max(Price_per_ticket) over (partition by class_id order by price_per_ticket desc) as row_no
115   from ticket_details;
116
117
118 -- Q12. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of
119 -- the passengers_on_flights table.
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Context Help Snippets

Output

Query Completed

Current temp Near record

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: CEP-2 x

SCHEMAS

No object selected

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
92 from ticket_details;
93
94 -- Q10 Write a query to create and grant access to a new user to perform operations on a database.
95
96 -- Create a new user and set a password
97 • CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'YourStrongPassword';
98
99 -- Grant all privileges on a specific database
100 • GRANT ALL PRIVILEGES ON YourDatabase.* TO 'new_user'@'localhost';
101
102 -- Make sure the changes take effect
103 • FLUSH PRIVILEGES;
104
105 -- Q11 Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.
106
107 • with Max_price_ticket as
108   (select class_id , price_per_ticket , row_number() over (partition by class_id order by price_per_ticket desc) as row_no
109   from ticket_details)
110   select * from Max_price_ticket
111   where row_no = 1;
112
113 -- method 2
114 • select class_id , price_per_ticket , max(Price_per_ticket) over (partition by class_id order by price_per_ticket desc) as row_no
115   from ticket_details;
116
117
118 -- Q12. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of
119 -- the passengers_on_flights table.
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Context Help Snippets

Output

Query Completed

Current temp Near record

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: CEP-2 x

SCHEMAS

No object selected

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
92 from ticket_details;
93
94 -- Q10 Write a query to create and grant access to a new user to perform operations on a database.
95
96 -- Create a new user and set a password
97 • CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'YourStrongPassword';
98
99 -- Grant all privileges on a specific database
100 • GRANT ALL PRIVILEGES ON YourDatabase.* TO 'new_user'@'localhost';
101
102 -- Make sure the changes take effect
103 • FLUSH PRIVILEGES;
104
105 -- Q11 Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.
106
107 • with Max_price_ticket as
108   (select class_id , price_per_ticket , row_number() over (partition by class_id order by price_per_ticket desc) as row_no
109   from ticket_details)
110   select * from Max_price_ticket
111   where row_no = 1;
112
113 -- method 2
114 • select class_id , price_per_ticket , max(Price_per_ticket) over (partition by class_id order by price_per_ticket desc) as row_no
115   from ticket_details;
116
117
118 -- Q12. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of
119 -- the passengers_on_flights table.
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Context Help Snippets

Output

Query Completed

Current temp Near record

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: CEP-2 x

SCHEMAS

No object selected

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
84
85 • select customer_id , seat_num , class_id from passengers_on_flights
86 group by 1,2,3
87 having class_id = "Economy Plus";
88
89 -- Q9 Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.
90
91 • select sum(no_of_tickets*Price_per_ticket) as total_revenue ,if(sum(no_of_tickets*Price_per_ticket) > 10000 , "revenue has crossed 10000" , "revenue has not crossed 10000")
92 from ticket_details;
93
94 -- Q10 Write a query to create and grant access to a new user to perform operations on a database.
95
96 -- Create a new user and set a password
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Context Help Snippets

Output

Query Completed

Current temp Near record

The screenshot shows the MySQL Workbench interface with the following details:

- Title Bar:** MySQL Workbench - Local instance MySQL80
- Menu Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Includes icons for Home, New, Open, Save, Run, Stop, Refresh, Undo, Redo, Find, Replace, Copy, Paste, and Print.
- Navigator:** CEP-2 (selected), SCHEMAS (selected), showing databases: ai, all, cc, en, mi, ne.
- SQL Editor:** Contains a large block of SQL code with numbered comments (Q10, Q11, Q12) and a note about improving performance.
- Right Panel:** SQLAdditions, Automate, disabled, manual, current, toggle.
- Status Bar:** Context Help

```
from ticket_details;
-- Q10 Write a query to create and grant access to a new user to perform operations on a database.
-- Create a new user and set a password
CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'YourStrongPassword';
-- Grant all privileges on a specific database
GRANT ALL PRIVILEGES ON YourDatabase.* TO 'new_user'@'localhost';
-- Make sure the changes take effect
FLUSH PRIVILEGES;

-- Q11 Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.
with Max_price_ticket as
(select class_id , price_per_ticket, row_number() over (partition by class_id order by price_per_ticket desc) as row_no
from ticket_details)
select * from Max_price_ticket
where row_no = 1j

-- method 2
select class_id , price_per_ticket, max(Price_per_ticket) over (partition by class_id order by price_per_ticket desc) as row_no
from ticket_details;

-- Q12. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of
-- the passengers_on_flights table.
```

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for database management. The main area is titled "CEP - 2" and contains a code editor with the following SQL queries:

```
122
123 •  create index per_index_route_id on passengers_on_flights(route_id);
124
125 •  select * from passengers_on_flights where route_id = 4;
126
127 -- Q13. For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.
128
129 •  EXPLAIN
130   SELECT *
131   FROM passengers_on_flights
132   WHERE route_id = 4;
133
134 -- Q14. Write a query to calculate the total price of all tickets booked by a customer across
```

Below the code editor is a "Result Grid" table showing the execution plan for the EXPLAIN query:

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	passengers_on_flights	HULL	ref	per_index_route_id	per_index_route_id	5	const	3	100.00	HULL

The bottom right corner of the interface features a vertical toolbar with four tabs: Result Grid (selected), Form Editor, Field Types, and Query Stats.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

The screenshot shows the MySQL Workbench interface. The top bar displays the Windows taskbar with icons for search, file explorer, and various applications. The MySQL Workbench title bar is visible, along with the 'Local instance MySQL80' connection tab.

The main area contains a query editor window titled 'CEP - 2'. It displays the following SQL code:

```
122
123 •  create index per_index_route_id on passengers_on_flights(route_id);
124
125 •  select * from passengers_on_flights where route_id = 4;
126
127 -- Q13. For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.
128
129 •  EXPLAIN
130
131   SELECT *
132     FROM passengers_on_flights
133       WHERE route_id = 4;
134
135 -- Q14. Write a query to calculate the total price of all tickets booked by a customer across
```

Below the code, the 'Result Grid' tab is active, showing the execution plan for the EXPLAIN query:

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	passengers_on_flights	NULL	ref	per_index_route_id	per_index_route_id	5	const	3	100.00	NULL

The right side of the interface features a vertical toolbar with icons for Result Grid, Form Editor, Field Types, and Query Stats. The status bar at the bottom indicates 'Query Completed'.

The screenshot shows the MySQL Workbench interface with a query editor titled "CEP-2". The code in the editor is as follows:

```
150 •    select * from business_class_customers;
151
152
153 -- Q16. Write a query to create a stored
154 -- procedure that extracts all the details from the routes table where the travelled distance is more than 2000 miles.
155
156 delimiter //
157 •  create procedure routes_with_dis_more_then_2000 ()
158 begin
159     select * from routes
160     where distance_miles > 2000;
161     End //
162     delimiter ;
163
164 •  call routes_with_dis_more_then_2000();
165
166
167 -- Q17. Write a query to create a stored procedure that groups the distance travelled by each flight into three categories.
168 -- The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500,
169 -- and long-distance travel (LDT) for >6500.
170
171 delimiter //
172
173 •  create procedure travelled_distance()
174 begin
175     select route_id,
176     distance_miles,
177     case
```

The screenshot shows the SSMS interface. At the top, there's a toolbar with icons for file operations like New, Open, Save, and Print. Below the toolbar is a menu bar with 'File', 'Edit', 'View', 'Tools', 'Help', and 'SQL Server Object Explorer'. The main window has a title bar 'SQLAdditions' and a status bar at the bottom. In the center, there's a large text area containing the message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.' At the bottom of the screen, there's a tab bar with 'Context Help' and 'Snippets' tabs.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

```
150 • select * from business_class_customers;
151
152 -- Q16. Write a query to create a stored
153 -- procedure that extracts all the details from the routes table where the travelled distance is more than 2000 miles.
154
155
156 delimiter //
157 • create procedure routes_with_dis_more_then_2000 ()
158 begin
159 select * from routes
160 where distance_miles > 2000;
161 End //
162 delimiter ;
163
164 • call routes_with_dis_more_then_2000();
165
166
167 -- Q17. Write a query to create a stored procedure that groups the distance travelled by each flight into three categories.
168 -- The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500,
169 -- and long-distance travel (LDT) for >6500.
170
171 delimiter //
172
173 • create procedure travelled_distance()
174 begin
175 select route_id,
176 distance_miles,
177 case
178 when distance_miles >=0 AND distance_miles <= 2000 then "short distance travel (SDT)"
179 when distance_miles >2000 AND distance_miles <= 6500 then "intermediate distance travel (IDT)"
180 when distance_miles > 6500 then "long-distance travel (LDT)"
181 end as x
182 from routes;
183 end //
```

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

```
163
164 • call routes_with_dis_more_then_2000();
165
166
167 -- Q17. Write a query to create a stored procedure that groups the distance travelled by each flight into three categories.
168 -- The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500,
169 -- and long-distance travel (LDT) for >6500.
170
171 delimiter //
172
173 • create procedure travelled_distance()
174 begin
175 select route_id,
176 distance_miles,
177 case
178 when distance_miles >=0 AND distance_miles <= 2000 then "short distance travel (SDT)"
179 when distance_miles >2000 AND distance_miles <= 6500 then "intermediate distance travel (IDT)"
180 when distance_miles > 6500 then "long-distance travel (LDT)"
181 end as x
182 from routes;
183 end //
```

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

```
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
```

```
163
164 • call routes_with_dis_more_then_2000();
165
166
167 -- Q17. Write a query to create a stored procedure that groups the distance travelled by each flight into three categories.
168 -- The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500,
169 -- and long-distance travel (LDT) for >6500.
170
171 delimiter //
172
173 • create procedure travelled_distance()
174 begin
175 select route_id,
176 distance_miles,
177 case
178 when distance_miles >=0 AND distance_miles <= 2000 then "short distance travel (SDT)"
179 when distance_miles >2000 AND distance_miles <= 6500 then "intermediate distance travel (IDT)"
180 when distance_miles > 6500 then "long-distance travel (LDT)"
181 end as x
182 from routes;
183 end //
```

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

```
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
```

```
163
164 • call routes_with_dis_more_then_2000();
165
166
167 -- Q17. Write a query to create a stored procedure that groups the distance travelled by each flight into three categories.
168 -- The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500,
169 -- and long-distance travel (LDT) for >6500.
170
171 delimiter //
172
173 • create procedure travelled_distance()
174 begin
175 select route_id,
176 distance_miles,
177 case
178 when distance_miles >=0 AND distance_miles <= 2000 then "short distance travel (SDT)"
179 when distance_miles >2000 AND distance_miles <= 6500 then "intermediate distance travel (IDT)"
180 when distance_miles > 6500 then "long-distance travel (LDT)"
181 end as x
182 from routes;
183 end //
```

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

```
163
164 • call routes_with_dis_more_then_2000();
165
166
167 -- Q17. Write a query to create a stored procedure that groups the distance travelled by each flight into three categories.
168 -- The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500,
169 -- and long-distance travel (LDT) for >6500.
170
171 delimiter //
172
173 • create procedure travelled_distance()
174 begin
175 select route_id,
176 distance_miles,
177 case
178 when distance_miles >=0 AND distance_miles <= 2000 then "short distance travel (SDT)"
179 when distance_miles >2000 AND distance_miles <= 6500 then "intermediate distance travel (IDT)"
180 when distance_miles > 6500 then "long-distance travel (LDT)"
181 end as x
182 from routes;
183 end //
```

Output

Query Completed

27°C Mostly cloudy

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Schemas

No object selected

```
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
```

```
163
164 • call routes_with_dis_more_then_2000();
165
166
167 -- Q17. Write a query to create a stored procedure that groups the distance travelled by each flight into three categories.
168 -- The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500,
169 -- and long-distance travel (LDT) for >6500.
170
171 delimiter //
172
173 • create procedure travelled_distance()
174 begin
175 select route_id,
176 distance_miles,
177 case
178 when distance_miles >=0 AND distance_miles <= 2000 then "short distance travel (SDT)"
179 when distance_miles >2000 AND distance_miles <= 6500 then "intermediate distance travel (IDT)"
180 when distance_miles > 6500 then "long-distance travel (LDT)"
181 end as x
182 from routes;
183 end //
```

Output

Query Completed

27°C Mostly cloudy



Excel Project Work 1.0



Beginner

Intermediate

Expert

WHAT I DID ?

- Utilized Sample E-Commerce Database:
 - Imported and prepared the sample e-commerce database for analysis.
- Sales and Profit Table (Month-wise):
 - Created a detailed table displaying monthly sales and profit data in a working sheet.
- Sales Table (Region-wise):
 - Prepared a table showing sales data categorized by different regions.
- User Control Combo Box for Product Category:
 - Implemented a user control combo box to allow users to select different product categories.
- Column Chart Creation:
 - Developed a column chart visualizing month-wise sales and profit data.
 - Created another column chart to illustrate region-wise sales data.
- *Table and Combo Box Linking:
 - Linked the sales and profit tables with the combo box to dynamically update data based on the selected product category.
- Dashboard Creation:
 - Designed an interactive sales dashboard that integrates all the elements (tables, charts, combo box).
 - Enabled users to analyze sales trends month-wise and product-wise based on selected categories using the dashboard.
- Tools Utilized:
 - Employed Microsoft Excel and Data Analysis Add-in to perform all tasks and analyses.

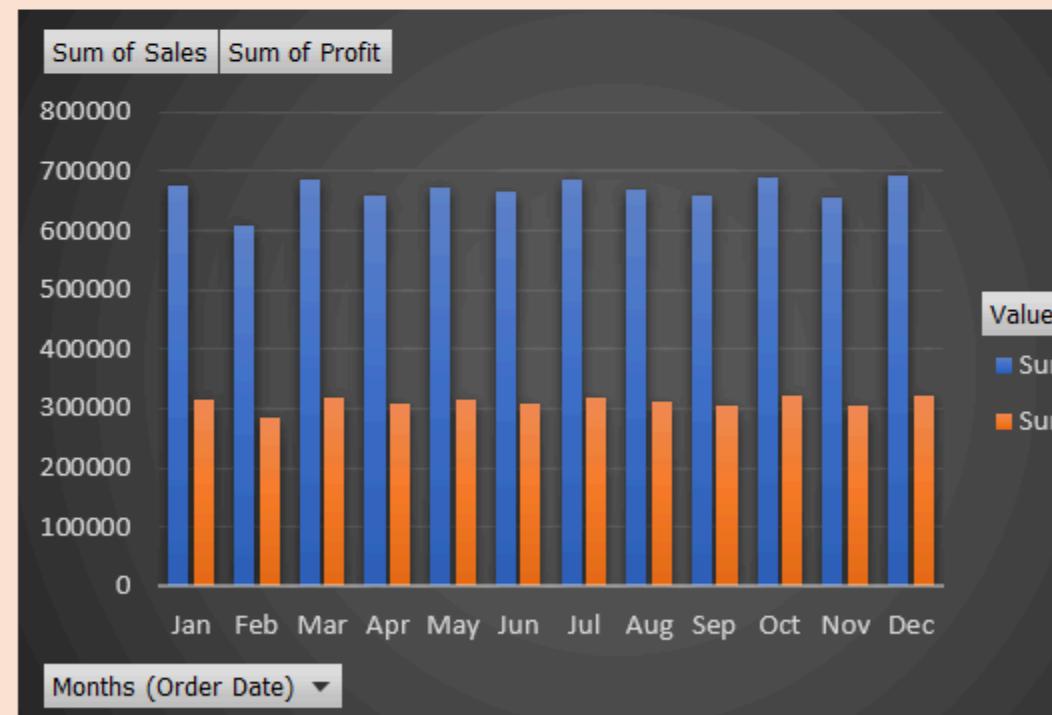
01

DASHBOARD



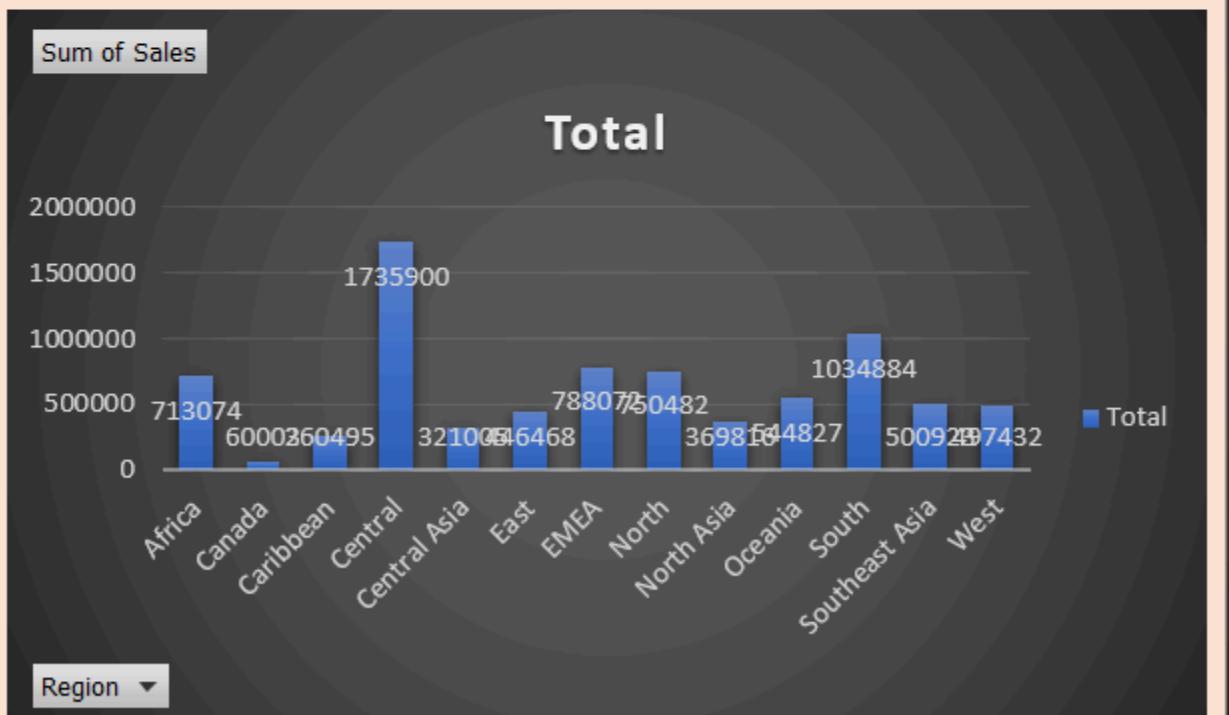
SALES

8023381



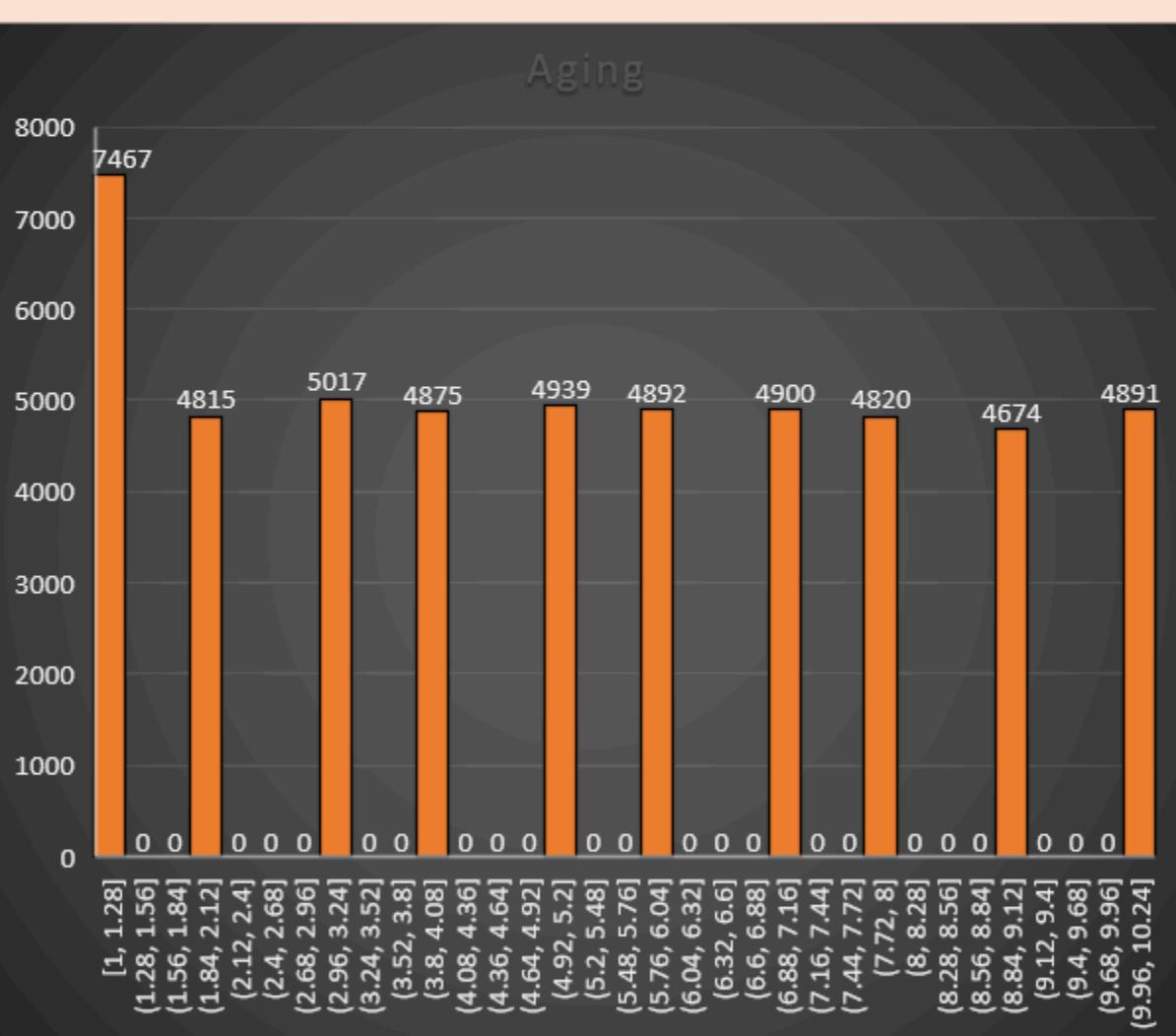
PROFIT

3729902.95



QUANTITY

153732



AutoSave Off H Designing a Sales dashboard... • Saved to this PC Search Priyanka Patel PP

File Home Insert Page Layout Formulas Data Review View Developer Help

Cut Copy Format Painter Paste Clipboard Calibri 11 A A Wrap Text General Conditional Formatting AutoSum

Font Alignment Merge & Center Number Format as Table Fill

Cell Styles Insert Delete Format Sort & Filter Clear

Format Painter

Clipboard G22 : fx

A B C D E F G H I J K L M N O P Q R S

1

2

3 Months Sum of Sales Sum of Profit Region Sum of Sales Row Labels Sum of Quantity Row Labels Sum of Aging AGE

4 Jan 676313 313566.3467 Africa 713074 Jan 12985 AU-2015-1 8

5 Feb 610240 286102.6233 Canada 60003 Feb 11501 AU-2015-10 10

6 Mar 686681 317186.0067 Caribbean 260495 Mar 13349 AU-2015-100 2

7 Apr 659404 308364.5133 Central 1735900 Apr 12314 AU-2015-1000 8

8 May 672547 313751.2467 Central Asia 321005 May 12854 AU-2015-1001 3

9 Jun 664560 307585.0233 East 446468 Jun 12851 AU-2015-1002 7

10 Jul 685152 318703.2 EMEA 788072 Jul 13277 AU-2015-1003 8

11 Aug 670788 310442.8433 North 750482 Aug 13122 AU-2015-1004 7

12 Sep 658844 305334.4567 North Asia 369816 Sep 12568 AU-2015-1005 8

13 Oct 689116 320748.67 Oceania 544827 Oct 13087 AU-2015-1006 2

14 Nov 656663 304716.1 South 1034884 Nov 12683 AU-2015-1007 5

15 Dec 693073 323401.92 Southeast Asia 500923 Dec 13141 AU-2015-1008 8

16 Grand Total 8023381 3729902.95 West 497432 Grand Total 153732 AU-2015-1009 7

17 8023381 3729902.95 Grand Total 8023381 153732 AU-2015-101 6

18

19

20

21

22

23

24

25

26

27

28

DASHBOARD TABLES RAW DATA +

Ready Accessibility: Investigate 32°C Cloudy ENG IN 13:34 02-07-2024 PRE



Excel Project Work 2.0



Beginner

Intermediate

Expert

WHAT I DID ?

1. Data Preparation:

- Used the restaurant tips dataset for analysis.
- Checked for missing values and cleaned the data accordingly.

2. Feature Identification:

- Identified independent features (predictors) and dependent features (target).
- Independent features: e.g., total bill, size of the party, etc.
- Dependent feature: tips.

3. Predictive Problem Identification:

- Recognized the problem as a regression problem to predict a continuous variable (tips).

4. Data Encoding:

- Encoded categorical variables into numeric values using IF conditions.
- Ensured all variables were in a format suitable for analysis.

5. Model Building:

- Built a regression model using the dataset.
- Derived the mathematical equation for predicting tips.

6. Prediction and Evaluation:

- Calculated predicted tips using the regression model.
- Compared predicted tips to actual tips values.
- Computed the Root Mean Square Error (RMSE) to evaluate model performance.

7. Tools Used:

- Utilized Microsoft Excel and Data Analysis Add-in for the entire process.

8. Deliverables:

- Developed a model to predict restaurant tips.
- Provided the mathematical equation for tip prediction.

AutoSave Off H Predicting restaurant tips using predictive anal... • Saved to this PC Search Priyanka Patel PP

File Home Insert Page Layout Formulas Data Review View Developer Help

Cut Copy Format Painter Paste Clipboard Font Alignment Number Styles Cells Editing Add-ins

B3 : fx =IF(TIPS!B2="Yes",1,IF(TIPS!B2="No",0,""))

Categorical variables to numeric values

sex	smoker	Sun	Fri	Sat	Thur	time	size	total_bill	tip
0	0	1	0	0	0	1	2	16.99	1.01
1	0	1	0	0	0	1	3	10.34	1.66
1	0	1	0	0	0	1	3	21.01	3.5
1	0	1	0	0	0	1	2	23.68	3.31
0	0	1	0	0	0	1	4	24.59	3.61
1	0	1	0	0	0	1	4	25.29	4.71
1	0	1	0	0	0	1	2	8.77	2
1	0	1	0	0	0	1	4	26.88	3.12
1	0	1	0	0	0	1	2	15.04	1.96
1	0	1	0	0	0	1	2	14.78	3.23
1	0	1	0	0	0	1	2	10.27	1.71
0	0	1	0	0	0	1	4	35.26	5
1	0	1	0	0	0	1	2	15.42	1.57
1	0	1	0	0	0	1	4	18.43	3
0	0	1	0	0	0	1	2	14.83	3.02
1	0	1	0	0	0	1	2	21.58	3.92
0	0	1	0	0	0	1	3	10.33	1.67
1	0	1	0	0	0	1	3	16.29	3.71
0	0	1	0	0	0	1	3	16.97	3.5
1	0	0	0	1	0	1	3	20.65	3.35
1	0	0	0	1	0	1	2	17.92	4.08
0	0	0	0	1	0	1	2	20.29	2.75
0	0	0	0	1	0	1	2	15.77	2.23
1	0	0	0	1	0	1	4	39.42	7.58

CORRELATION

	sex	smoker	Sun	Fri	Sat	Thur	time	size	total_bill	tip
sex	1									
smoker	0.00282	1								
Sun	0.16811	-0.18162	1							
Fri	-0.07106	0.24432	-0.19545	1						
Sat	0.05396	0.15574	-0.50068	-0.21632	1					
Thur	-0.19444	-0.12853	-0.39257	-0.16961	-0.43448	1				
time	0.20523	0.05492	0.41807	-0.05816	0.46271	-0.918	1			
size	0.08619	-0.13318	0.19305	-0.14218	-0.04112	-0.0726	0.10341	1		
total_bill	0.14488	0.08572	0.12295	-0.08617	0.05492	-0.13817	0.18312	0.59832	1	
tip	0.08886	0.00593	0.12511	-0.05546	-0.00279	-0.09588	0.12163	0.4893	0.67573	1

THE ABOVE DATA SUGGEST US THAT ONLY "SIZE & TOTAL BILL" HAS STRONG RELATIONSHIP WITH TIPS, HENCE WE WILL USE THEM IN THE PREDICTION MODE.

TIPS CORRELATION Sheet1 PREDICTION SHEET +

Ready Accessibility: Investigate

Nifty midcap -1.17% Search

ENG IN 13:36 02-07-2024 PRE

AutoSave Off H Predicting restaurant tips using predictive anal... • Saved to this PC Search Priyanka Patel PP

File Home Insert Page Layout Formulas Data Review View Developer Help Comments Share

Cut Calibri 11 A A Wrap Text General AutoSum
Copy Merge & Center Conditional Formatting Table Format as Cell Insert AutoSum
Format Painter B I U Alignment Number Delete Format Fill Sort & Find & Add-ins
Paste Clipboard Font Styles Add-ins
Font Alignment Number Styles Cells Editing Add-ins

G17 : X ✓ fx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	size	total_bill	tip	PREDICTED TIPS	SQUARE ERROR													
2	2	16.99	1.01	2.62934	2.622261784													
3	3	10.34	1.66	2.20539	0.297454645													
4	3	21.01	3.5	3.19465	0.093241474													
5	2	23.68	3.31	3.24959	0.003649109													
6	4	24.59	3.61	3.71916	0.011915223													
7	4	25.29	4.71	3.78406	0.857371908													
8	2	8.77	2	1.86724	0.017626202													
9	4	26.88	3.12	3.93147	0.65848423													
10	2	15.04	1.96	2.44855	0.238680043													
11	2	14.78	3.23	2.42444	0.648921359													
12	2	10.27	1.71	2.00631	0.087797423													
13	4	35.26	5	4.70841	0.085025792													
14	2	15.42	1.57	2.48378	0.834993858													
15	4	18.43	3	3.14804	0.021916646													
16	2	14.83	3.02	2.42908	0.349187493													
17	2	21.58	3.92	3.05489	0.748408152													
18	3	10.33	1.67	2.20447	0.28565486													
19	3	16.29	3.71	2.75704	0.908135847													
20	3	16.97	3.5	2.82008	0.462286515													
21	3	20.65	3.35	3.16127	0.035619568													
22	2	17.92	4.08	2.71556	1.861687438													
23	2	20.29	2.75	2.93529	0.034333842													
24	2	15.77	2.23	2.51623	0.081927413													
25	4	39.42	7.58	5.09410	6.179720406													
26	2	19.82	3.18	2.89172	0.083106128													
27	4	17.81	2.34	3.09056	0.563340985													

MULTIPLE LINEAR REGRESSION ANALYSIS

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.684009729
R Square	0.467869309
Adjusted R Square	0.463453286
Standard Error	1.013505967
Observations	244

ANOVA

	df	SS	MS	F	Significance F
Regression	2	217.6586401	108.82932	105.9481301	9.66509E-34
Residual	241	247.553837	1.027194344		
Total	243	465.212477			

Coefficients Standard Error t Stat P-value Lower 95% Upper 95% Lower 95.0% Upper 95.

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.
Intercept	0.668944741	0.193609331	3.455126549	0.000649806	0.287562197	1.050327285	0.287562197	1.050327
size	0.192597794	0.085314557	2.25750214	0.024872446	0.024540385	0.360655204	0.024540385	0.360655
total_bill	0.092713337	0.009114682	10.17186688	1.88092E-20	0.074758723	0.110667951	0.074758723	0.110667

RMSE 1.007

TIPS CORRELATION Sheet1 PREDICTION SHEET +

Ready Accessibility: Investigate

1 Nifty midcap -1.17% 1 Search 21 13:37 ENG IN 02-07-2024 PRE



Python Project Work



Beginner

Intermediate

Expert

WHAT I DID ?

1. Libraries Imported:

- Used Python libraries: numpy, pandas, seaborn, and matplotlib.

2. Data Inspection:

- Checked dataset shape, type, and information using `shape`, `info`, and `dtype`.

3. Missing Values:

- Identified null values using `isna` and `sum` functions.

4. Numerical Variable Analysis:

- Analyzed age, BMI, and children vs. charges, considering smoking status.
 - Age: Higher age correlates with higher charges.
 - BMI: No direct effect on charges.
 - Children: Charges decrease as the number of children increases.
 - Smoking: Smokers pay higher charges compared to non-smokers.

5. Categorical Variable Analysis:

- Analyzed sex, region, and smokers.
 - Fewer female smokers compared to males.

- Southwest has the least smokers.

- More non-smokers than smokers in the dataset.

6. Charges vs. Categorical Variables:

- Compare sex, region, and smoking status against charges.
 - Males and females pay similar charges.
 - No significant difference in charges across regions.
 - Smokers pay higher charges than non-smokers.

7. Age vs. Charges:

- Compared age to charges considering smoking status.
 - Charges increase with age.
 - Smokers of all ages pay more than non-smokers.

Conclusion:

- Recommend insurance companies to encourage early-age insurance adoption.
- Raise awareness among smokers for affordable health insurance.
- Offer effective health insurance for families with more children based on locality surveys.

The screenshot shows a Jupyter Notebook interface with the following details:

- File Menu:** File, Edit, View, Run, Kernel, Tabs, Settings, Help.
- Toolbar:** +, - (New, Close), Refresh, Code, Python 3 [3.10].
- File Explorer:** Shows a list of files in the current directory (work/). The "Python Ins..." file is selected.
- Code Cell 1:** Displays code for importing libraries (Pandas, matplotlib, NumPy, seaborn) and loading the insurance dataset from a CSV file.
- Text Cell:** A descriptive note explaining the purpose of the first task.
- Code Cell 2:** Displays code for checking the variables of the dataset and data structure, followed by the output showing the first five rows of the dataset.
- Data Preview:** A table showing the first five rows of the "insurance" dataset with columns: age, sex, bmi, children, smoker, region, and charges.

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

The screenshot shows a Jupyter Notebook interface with the following details:

- File Menu:** File, Edit, View, Run, Kernel, Tabs, Settings, Help.
- Toolbar:** Includes icons for New, Open, Save, Run, Stop, Cell, Kernel, Help, and Code.
- Python Version:** Python 3 [3.10]
- File Explorer:** Shows files in the current directory: Feb 10 sess..., Feb 11 sess..., Feb 17 sess..., Feb 18 sess..., Feb 24 sess..., Feb 25 sess..., insurance.csv, iris.csv, March 02 sess..., March 03 sess..., March 09 sess..., March 10 sess..., March 11 sess..., Pandas-Sal..., and Python Ins... (selected).
- Code Editor:** Displays Python code for analyzing the insurance dataset.

Code Content:

```
[3]: #2.*Check the shape of the data along with the data types of the column:  
insurance.shape  
  
## dataset has 1338 rows and 7 columns.  
[3]: (1338, 7)  
  
for 2nd task used shape , info and dtype to check shape of data and type of data in the dataset.  
  
[4]: insurance.info()  
  
## dataset has data types of the column: integer, object , float.  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1338 entries, 0 to 1337  
Data columns (total 7 columns):  
 #   Column   Non-Null Count   Dtype     
---  --    
 0   age      1338 non-null    int64    
 1   sex      1338 non-null    object    
 2   bmi      1338 non-null    float64   
 3   children 1338 non-null    int64    
 4   smoker    1338 non-null    object    
 5   region   1338 non-null    object    
 6   charges   1338 non-null    float64   
dtypes: float64(2), int64(2), object(3)  
memory usage: 73.3+ KB  
  
[5]: None more method to check data type of column is:  
insurance.dtypes  
  
[5]: age      int64  
sex     object  
bmi    float64  
children  int64  
smoker  object
```

File Edit View Run Kernel Tabs Settings Help

Python Insurance Project.ipynb

Filter files by name

/ work /

Name	Last Modified
Feb 10 sess...	4 days ago
Feb 11 sess...	4 days ago
Feb 17 sess...	4 days ago
Feb 18 sess...	4 days ago
Feb 24 sess...	3 days ago
Feb 25 sess...	3 days ago
insurance.csv	16 hours ago
iris.csv	3 days ago
March 02 s...	3 days ago
March 03 s...	3 days ago
March 09se...	2 days ago
March 10 s...	a day ago
March 11 s...	16 hours ago
Pandas-Sal...	3 days ago
Python Ins...	6 minutes ago

[5]:

```
age      int64
sex      object
bmi     float64
children  int64
smoker    object
region    object
charges   float64
dtype: object
```

[6]: #3. Check missing values in the dataset and find the appropriate measures to fill in the missing values:

```
insurance.isna().sum()
```

there are no missing value in the data set.

[6]:

```
age      0
sex      0
bmi     0
children  0
smoker    0
region    0
charges   0
dtype: int64
```

for 3rd task used isna function and sum function to get null values in the data set

[7]: #4. Explore the relationship between the feature and target column using a count plot of categorical columns and a scatter plot of numerical columns

```
#scatter plot of numerical columns age, bmi and children:
```

```
fig,axes=plt.subplots(1,3,figsize=(21,7))

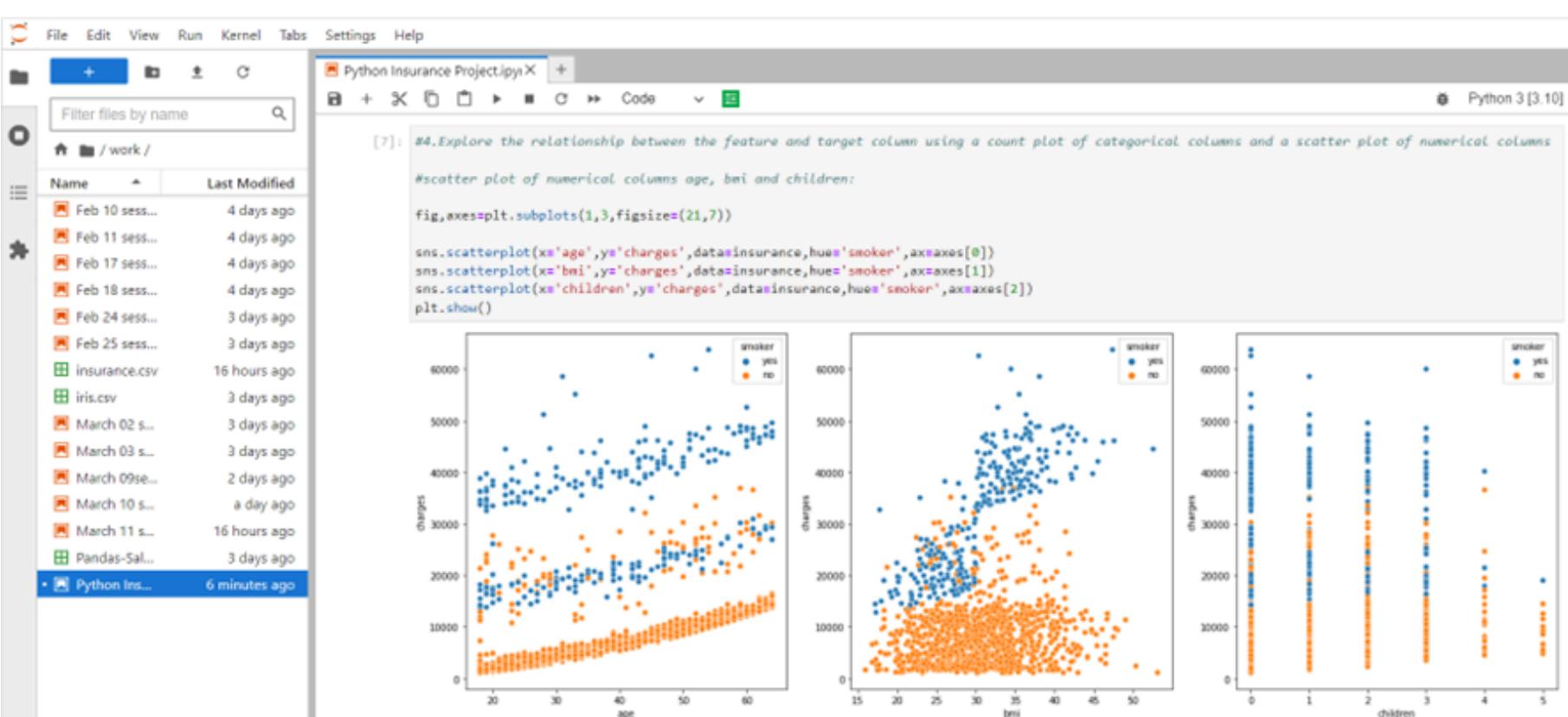
sns.scatterplot(x='age',y='charges',data=insurance,hue='smoker',ax=axes[0])
sns.scatterplot(x='bmi',y='charges',data=insurance,hue='smoker',ax=axes[1])
sns.scatterplot(x='children',y='charges',data=insurance,hue='smoker',ax=axes[2])
plt.show()
```

smoker
• yes

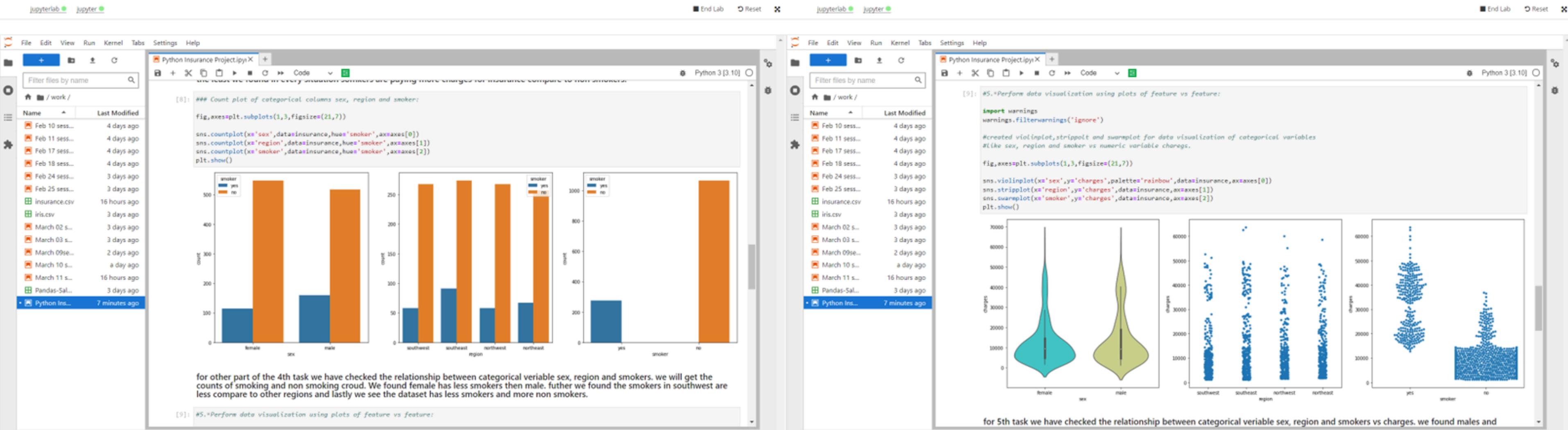
smoker
• yes

smoker
• yes

smoker
• yes

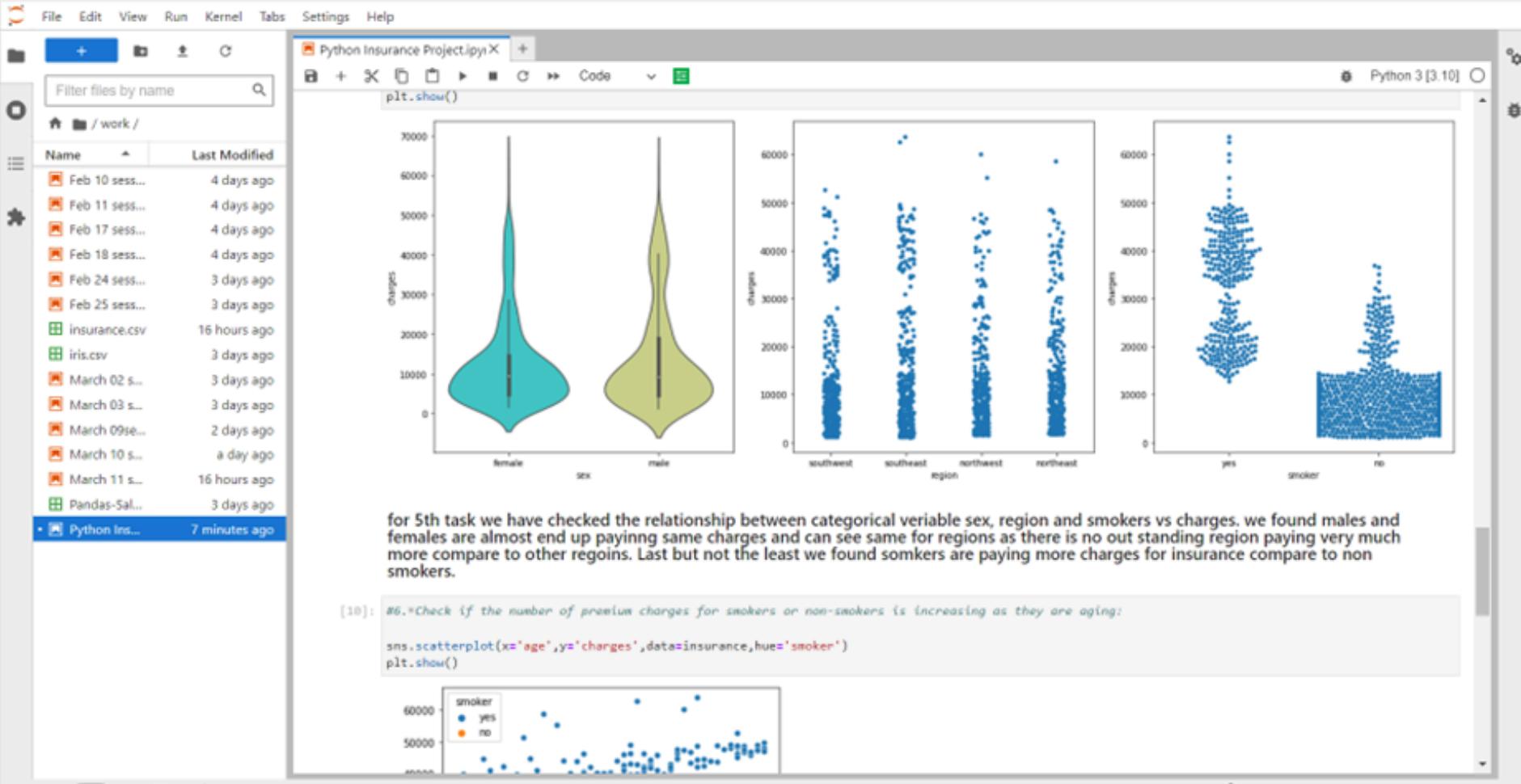


For 4th task we have checked the relationship between numerical variable age , bmi and children vs charges considering the effects of smoking and non smoking crowd. We found as age increases the charges for the insurance is also increased. where in the case of bmi it shows there is no direct effect of bmi to charges and for children the charges decrease as the number of childrens increase. Last but not least we found in every situation somkers are paying more charges for insurance compare to non smokers.

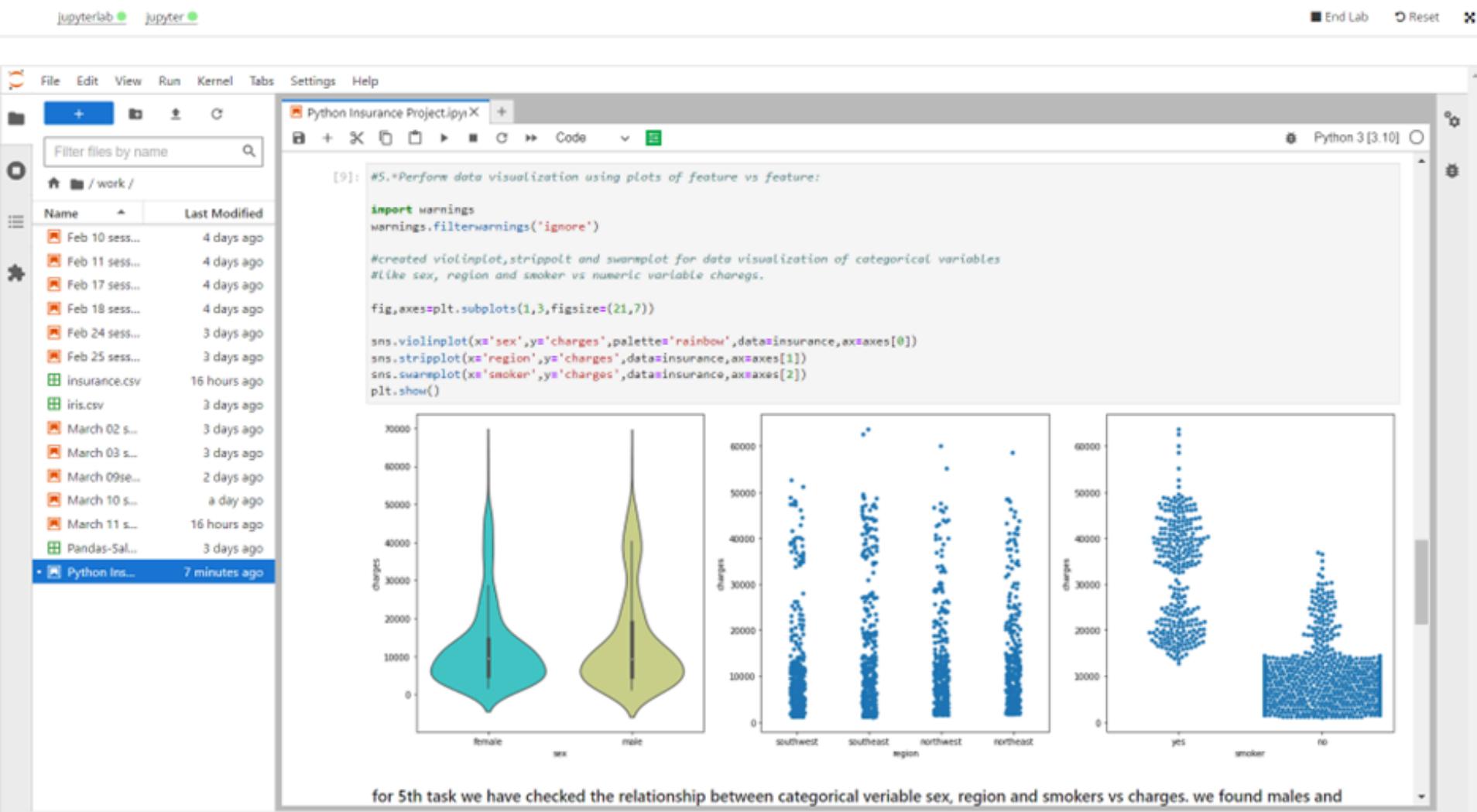


Simple 0 1 Python 3 [3.10] | Idle Mode: Command L 1, Col 1 Python Insurance Project.ipynb

jupyterlab jupyter

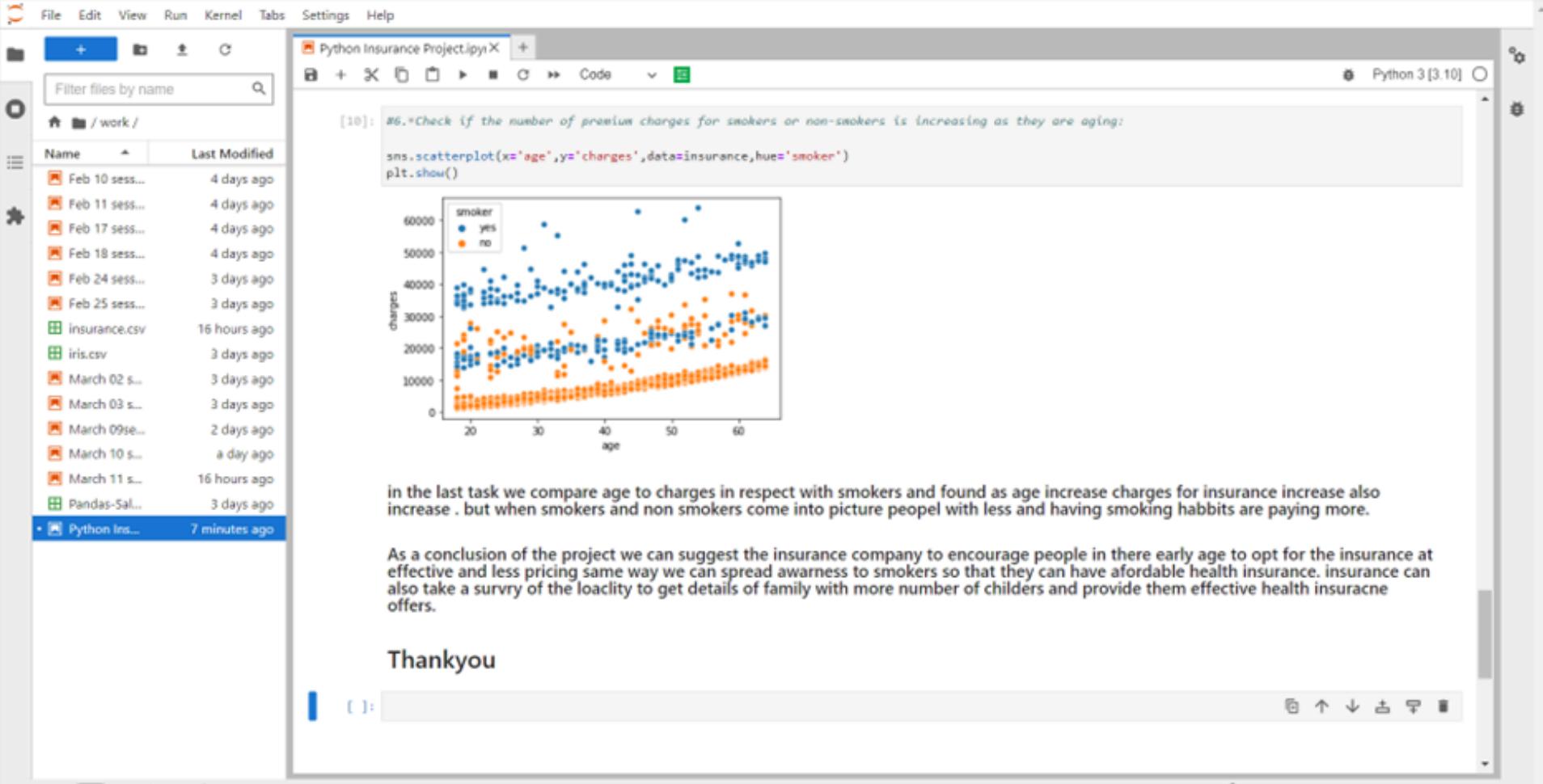


Simple 0 1 Python 3 [3.10] | Idle Mode: Command L 1, Col 1 Python Insurance Project.ipynb



Simple 0 1 Python 3 [3.10] | Idle Mode: Command L 1, Col 1 Python Insurance Project.ipynb

jupyterlab jupyter



Simple 0 1 Python 3 [3.10] | Idle Mode: Command L 1, Col 1 Python Insurance Project.ipynb

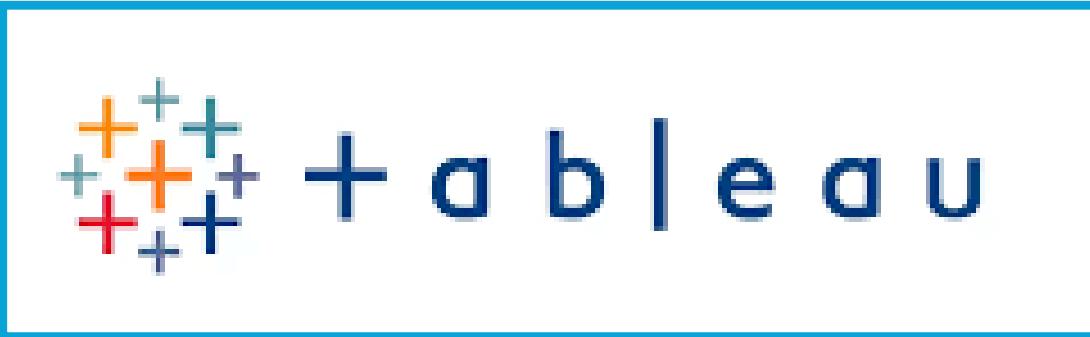


Tableau Project Work



Beginner

Intermediate

Expert

WHAT I DID ?

1. Dataset Selection

- Utilized the Sample Superstore dataset for the analysis.

2. Data Preparation

- Grouped data by Customer Name and Order ID to organize the dataset thoroughly.

3. Hierarchy Creation

- Created a hierarchy named "Location" for the variable Country to structure geographic data.

4. Parameter Creation

- Developed two parameters:
 - **Primary Region**: Included all regions for comparison.
 - **Secondary Region**: Included all regions for comparison.

5. Calculated Fields

- Created calculated fields for:
 - **Primary Region**: To filter and compare data.
 - **Secondary Region**: To filter and compare data.

- 1 - **First Order Date**: Determined the first order date for analysis.

6. Dashboard Creation

- Designed a dashboard aligning all relevant sheets.
- Partitioned the dashboard to display detailed comparisons between Primary and Secondary Regions including:
 - First Order Date
 - Total Sales
 - Average Sales per Order
 - Number of Customers
 - Number of Orders
 - Number of Products in Sale



Dashboard

Layout

Default

Phone

Device Preview

Size

Automatic

Sheets

Pri Measures

Sec Measures

Pri Map

Sec Map

Pri Sub...

Sec Sub...

Pri Line Chart

Sec Line Chart

Pri Bar Chart

Objects

Horizontal Container

Vertical Container

Text

Extension

Data Story

Image

Blank

Tiled

Floating

 Show dashboard title

PRIMARY AND SECONDARY REGION PARAMETER IN USE

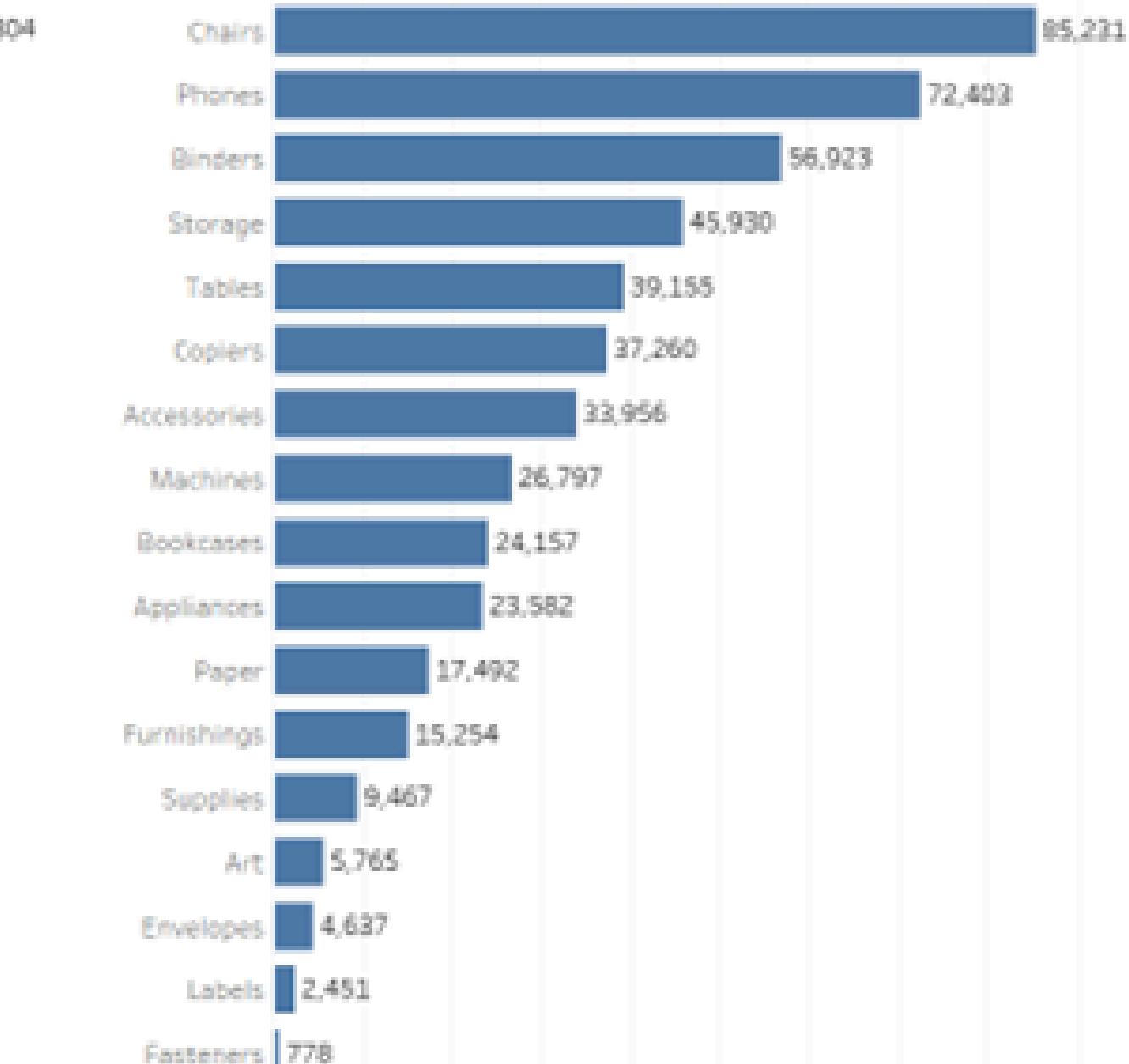
PRIMARY REGION

Sub-Cat.. F



SECONDARY REGION

Sub-Cat.. F



PRIMARY REGION

South

SECONDARY REGION

Central

Data Source

Sec Map

Pri Sub-Category Sales

Sec Sub-Category Sales

Pri Line Chart

Sec Line Chart

Pri Bar Chart

Sec Bar Chart

PRIMARY REGION

SECONDARY REGION

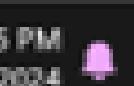
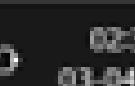
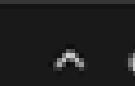
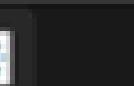
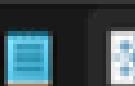
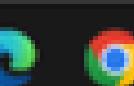
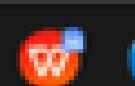
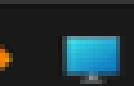
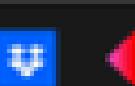
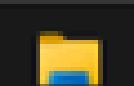
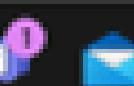
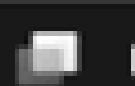
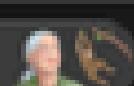
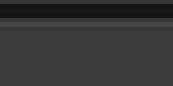
PRI/SEC COMP With PARA...

SALES COMPAIRSON

G1 G2 G3

82:35 PM
03-04-2024

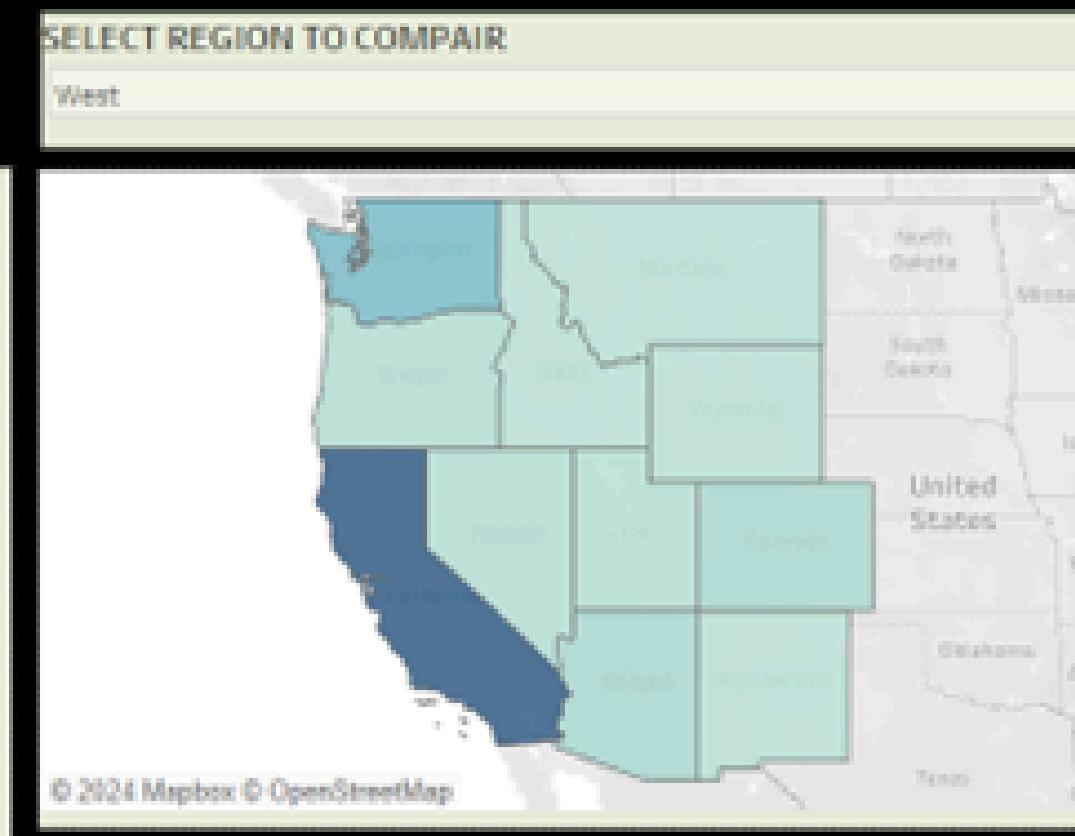
Search



SALES COMPARISONS BY REGION

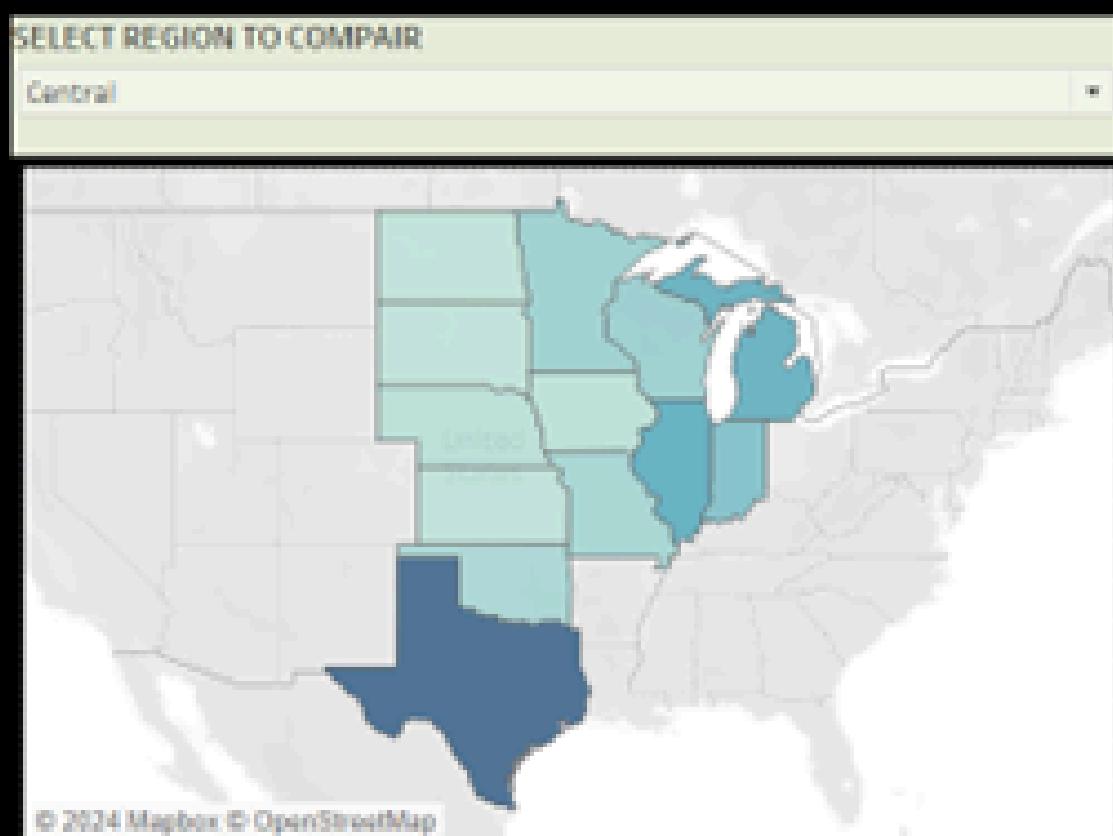
PRIMARY REGION

No. of Customers : 686
 First Order Date : 06-01-2014
 Total Sales : 725,458
 Average Sale Per Order : 218.7
 No. of Orders : 1,611
 No. of Product in Sale : 1,509

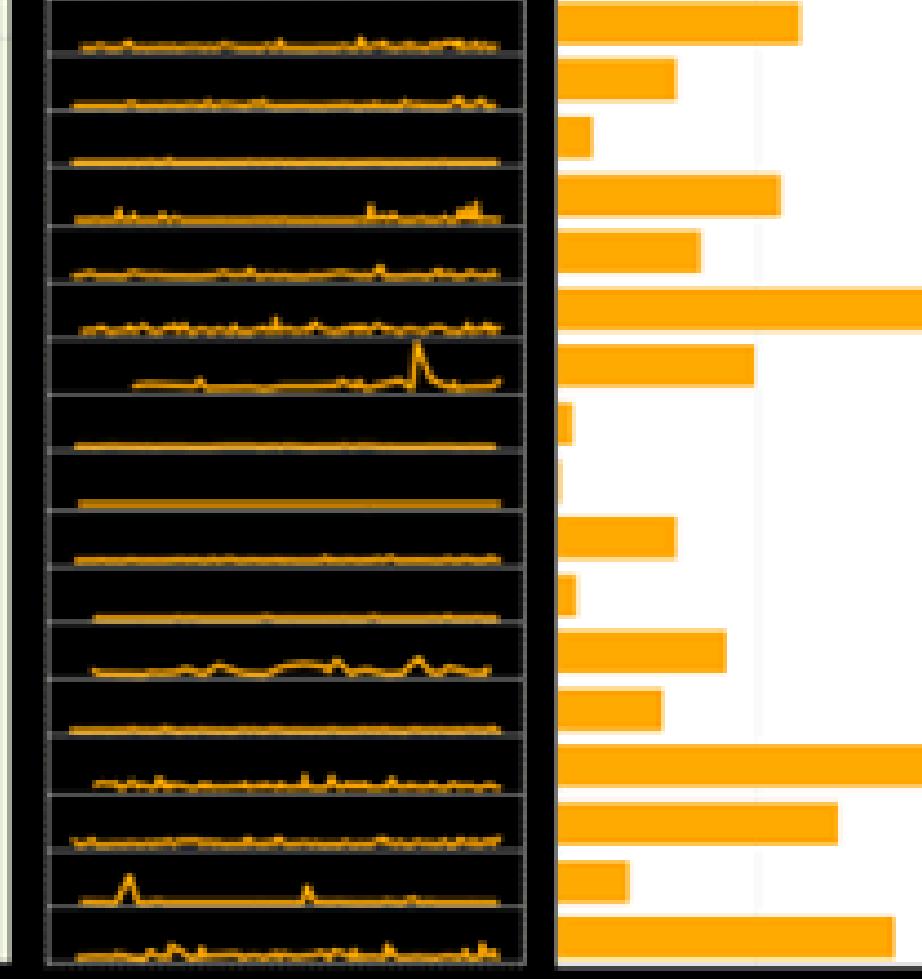


SECONDARY REGION

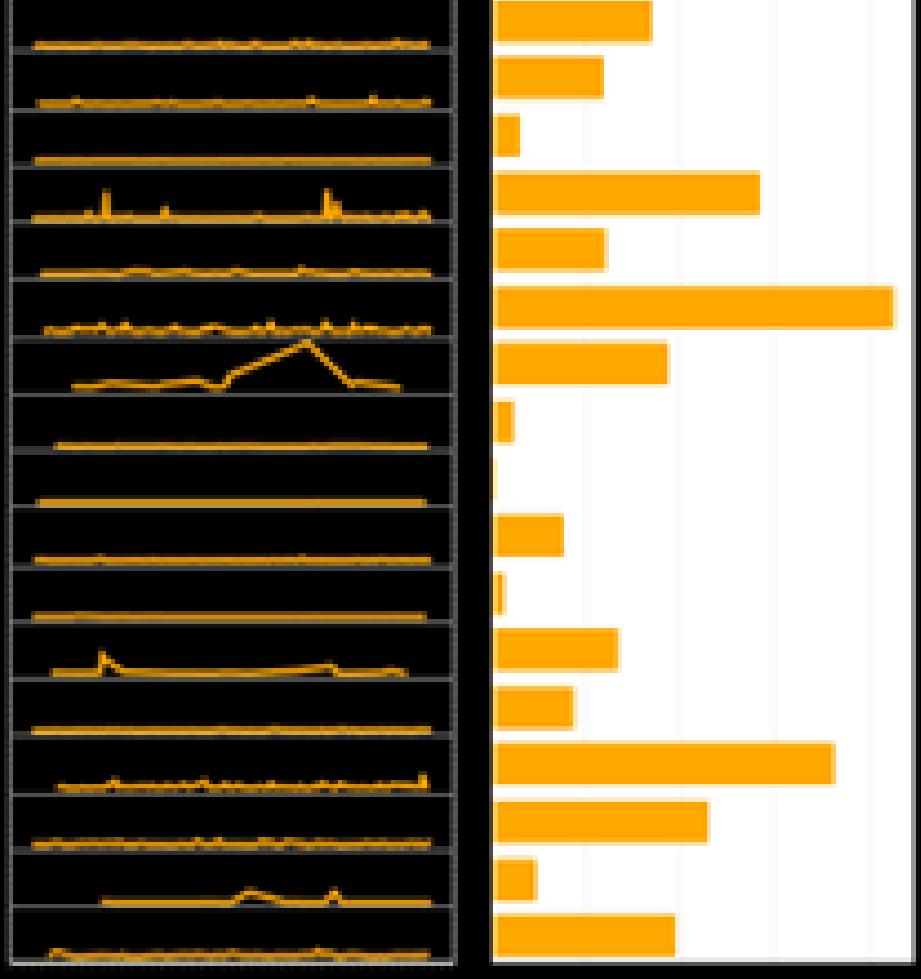
No. of Customers : 629
 First Order Date : 03-01-2014
 Total Sales : 501,240
 Average Sale Per Order : 212.2
 No. of Orders : 1,175
 No. of Product in Sale : 1,310



Sub-Cat	Total Sales	Min. Sales	Max. Sales
Chairs	\$ 101,791	\$ 41	\$ 2,804
Phones	\$ 98,684	\$ 6	\$ 4,159
Tables	\$ 84,755	\$ 24	\$ 3,611
Storage	\$ 70,533	\$ 6	\$ 1,856
Accessories	\$ 61,114	\$ 1	\$ 3,347
Binders	\$ 55,961	\$ 1	\$ 5,064
Copiers	\$ 49,749	\$ 300	\$ 14,000
Machines	\$ 42,444	\$ 12	\$ 4,900
Bookcases	\$ 36,004	\$ 49	\$ 3,407
Appliances	\$ 30,236	\$ 3	\$ 2,518
Furniture	\$ 30,073	\$ 3	\$ 1,049
Paper	\$ 26,664	\$ 4	\$ 734
Supplies	\$ 18,127	\$ 6	\$ 8,188
Art	\$ 9,212	\$ 1	\$ 1,113
Labels	\$ 5,079	\$ 3	\$ 786
Envelopes	\$ 4,118	\$ 4	\$ 420
Fasteners	\$ 923	\$ 2	\$ 93



Sub-Cat	Total Sales	Min. Sales	Max. Sales
Chairs	\$ 65,231	\$ 27	\$ 3,505
Phones	\$ 72,403	\$ 6	\$ 2,726
Binders	\$ 56,923	\$ 1	\$ 9,893
Storage	\$ 45,930	\$ 9	\$ 1,555
Tables	\$ 39,155	\$ 67	\$ 2,679
Copiers	\$ 37,260	\$ 220	\$ 17,500
Accessories	\$ 33,956	\$ 2	\$ 1,929
Machines	\$ 26,797	\$ 84	\$ 8,160
Bookcases	\$ 24,157	\$ 68	\$ 2,396
Appliances	\$ 23,582	\$ 0	\$ 2,405
Paper	\$ 17,462	\$ 4	\$ 629
Furniture	\$ 15,254	\$ 2	\$ 1,326
Supplies	\$ 9,467	\$ 2	\$ 4,164
Art	\$ 5,765	\$ 1	\$ 210
Envelopes	\$ 4,637	\$ 2	\$ 605
Labels	\$ 2,451	\$ 2	\$ 492
Fasteners	\$ 776	\$ 2	\$ 58



Project By :

BHAVIN SHAH

bhavinshah139@gmail.com

contact : 9820177312.

THANK YOU