Polytechnic Campus

Graduate Portfolio

By

Shreya Naresh

30th March, 2023

Evaluator: Dr Tatiana Walsh

TABLE OF CONTENTS

SECTION 1: RESUME

SHREYA NARESH

480.278.5228 • snolas81@asu.edu • www.linkedin.com/in/shreya-naresh-47a8a0162

SUMMARY

Graduate Student pursuing masters in information technology having strong communication and presentation skills with Internship experience as an IT engineer and data analyst. Project experiences include working with Structured Query language (SQL), CouchDB, Tableau, PowerBI, and Python, Developing RestAPIs, and working with Amazon Web Services. Seeking a full-time position in Information technology, analysis, technical role, or any related field. Comfortable with Unix, Windows, and Mac Operating Systems.

EDUCATION

M.S in Information technology Graduating May 2023
Arizona State University, Tempe, AZ GPA 4.0/4.0
Ira Fulton School of Engineering
Relevant Coursework: Advanced database management systems, Information system management, Advanced Data Analytics.

Bachelor of Information Technology Graduated April 2020
Mumbai University, Mumbai, India GPA 3.3/4.0
SIES Graduate School of Technology
Relevant Coursework: Database management systems, Software development, big data Analytics.

TECHNICAL SKILLS

- **Data Analysis and Statistics:** PowerBI, Tableau, Excel.
- **Programming Languages:** Python, C++, Apex, SQL, NoSQL (CouchDB), Javascript, HTML, CSS, React.JS
- **Data processing & ETL skills:** Data warehousing, ETL Design/Development.
- **Software development Model:** Agile methodology, REST API, Amazon Web Services (AWS), Python Django.

PROFESSIONAL EXPERIENCE

Interim Engineering Intern May 2022-August 2022
Qualcomm Inc, San Diego, CA

- Performed data analysis by creating visualizations using tableau to increase the Job analysis rate from 12% to 18%.
- Studied the process of data pre-processing and segregation to organize and process it for information management.
- Improved the knowledge base articles to improvise the database management system.
- Developed restful interface to ServiceNow to provide better JIRA for ServiceNow collaboration and software automation was done for secure information to handle passwords using flask web framework.

Data Analyst Intern July 2021 – Aug 2021
GlobalShala, India

- Data visualizations and analysis was done using PowerBI to identify the 8 unwanted fields by data preprocessing while enforcing and delivering presentations and led the communication flow among 6 teams.

IT Trainee July 2020 – Jan 2021
Fresa Technologies, India

- Designed a JIRA gadget with an HTML front-end and a Javascript and python backend that tracks open tickets.
- Implemented 2 Client projects with SQL and python data analysis.
- Conceptualized staff sessions for customer service support system to reduce complaints by 15%.

ACCOMPLISHMENTS

Medicare Fraud Detection using Big Data Oct 2022-Dec 2022

- Selected 5 organizational benefits of the Medicare industry, addressed solutions the issues using feature engineering and followed data-mining principles to execute exploratory data analysis to perform predictive model classification techniques for best results.
- Developed innumerable charts, deployed the final model and scripted the historical data in python.

Restaurant Database management System Feb 2022-April 2022

- A database was made for restaurants to manage daily activities that include Employees, Customers, Orders, menus, and Payment.
- 8 Structured Complex queries in SQL, Relational tables, and NoSQL (Couchbase) were tested.

Data visualization to eliminate Inefficient Ad campaign Oct 2021-Dec 2021

- Assisted in identifying the 15% low-cost Ad campaign to 18% on the given dataset using data visualization techniques using PowerBI and the data model using visual basic.
- Various graphs, charts, and statistical data of up to 12% were created using Structured query language.

WORK EXPERIENCE

Arizona State University, Tempe, AZ: Teaching Assistant August 2022- Present

Graduate teaching assistant for Cloud architecture using Amazon Web Services to build a defined process and deploy workloads to the Cloud, restructuring of course responsibilities, grade assignments and proctor exams for 120 students.

EXTRACURRICULAR EXPERIENCE

Information technology Gurus Association, Tempe, AZ Jan 2022- Present

- Attended networking sessions among different majors to know each other on all aspects of Information technology field.

SECTION 2: REFLECTION

Looking back at my experience throughout the whole degree, it would be apt to say that the most important thing I have learnt from this experience is how to keep learning. It has allowed me to see the world more widely, how much there is to learn and explore in the information technology and computer science in general. I feel I am more equipped now than I ever was to face the world and start my career. More confident, more skilled than I ever was. I met amazing professors with vast industry experience and learned lot from their stories and took a leaf out of their experience. I had an opportunity to meet likeminded students and discussing vast sea of ideas and hear completely amazing and novel perspectives on how they see the world. I definitely learned a lot of technical concepts, algorithms, ideas, technologies but more importantly my experience here inculcated in me a sense of hunger for keep learning more and more and never stop at that. It has been a wonderful journey at Arizona State University, and I am grateful that I made this decision to pursue this degree.

I have carefully chosen the projects I have chosen because now when I look back, I think those were the most impactful and important project of all. The first project is about designing a database and writing optimized query to fetch the information from the database. I believe this knowledge of lower-level database and how it functions is absolutely necessary in order to optimally work with the data, The second project focuses on using the existing dataset and developing an efficient predictive model which predicts using the data that is existing. Data visualization seems like a simple task at first, but it has lot of nuances that we need to take care of. Each and every figure tells a story. Based on how a visualization is designed, it gives lot of clues subconsciously and that is extremely important to learn. So, once we know about the creating the database and using that data to develop a predictive model, the last step would be to visualize the results we obtained from our model. This is what the third project accomplishes.

Hence, it comes a full circle and as mentioned before, these 3 projects combined encapsulates the job of a data scientist.

I also had an opportunity to apply all this knowledge learned through these projects in industry during my internship on a real-world problem. I worked on visualizing the data in the frontend and querying the database efficiently in the backend. I was successfully able to complete my internship and make a significant contribution to my team.

I am extremely glad that I took this important decision to pursue masters here at Arizona State University and it has been nothing less than a mind-blowing experience. I am grateful to all the professors who have contributed towards my progress and guided me through. It would not have been near the experience I have had, if it was not for the people I met in the process and who uplifted my enthusiasm to another level. I am very excited and confident to head out into the world and start my career as a professional.

SECTION 3: OVERVIEW

This Graduate portfolio demonstrates the skills I have learnt throughout my master's degree in Information technology by demonstrating my top 3 academic projects. I have attained many technical skills from my accomplishments. Here, I would like to describe my top 3 projects that I have received appreciation too for the same with excellent grades. As you can see from the projects I've mentioned here, data science and working closely with the data are my areas of interest. Data is quickly overtaking money as the most valuable currency in the world. I had the chance to study many ideas, kinds of datasets, and cutting-edge technology during this degree, which really motivated my passion to become a data scientist. Out of all the projects I've worked on, these 3 projects that capture my interests and my total degree learnings. From the first semester project to the last semester project, I have covered the projects in chronological order.

The first Project covers the very basic part of data science, the database and its technologies. I did a project that was a part of Advanced database management systems course. In that this project dealt with preparing the database itself, optimizing the searches, creating stored procedures, triggers, and writing very complex SQL queries. In this project, both a SQL and a NOSQL database were utilized. The topic of this project was to create a "Restaurant database management system". We proposed a database system design which would allow automate a restaurant's day-to-day operations. The major purpose of this system is to assist restaurant administrators in managing the restaurant business as well as to assist customers in online ordering. We can also keep track of payments made by clients and track their payment method if it is a cash or a card payment. This will help a restaurant's reputation and market worth, as well as its customer service.

The second Project covers the next essential part of data handling, called the Advanced analytics of big data. The topic of this project was to detect a Medicare fraud using the big data tools and technologies. We used Exploratory data analysis process (EDA), feature engineering and some machine learning classifiers to predict the best working model with the highest accuracy so we can predict better outcomes of fraud. The goal of this project is to develop big data solutions that have significant applicability at the intersection of the health care and protection sectors. Reducing fraud is one way to assist cover costs and lower overall payments. The predictive analysis helped us identify the future trends based on our historical data of patients.

Furthermore, my final project that demonstrates data analysis is my data visualization project about the cricket dataset and its analysis. We performed the entire steps of data analysis, from defining the question all the way down to Embracing the failure and finally getting to the dashboard summary. Here, we collected the data and segregated it as per our objective. We created a strategy for collecting and aggregating the appropriate data. We divided quantitative data and qualitative data for processing. Data cleaning like removal of null values or outliers, filling major gaps in the data was an important task before we proceeded further for analysis. Later, a tableau dashboard was created and shared within the class to gain more ideas of improvement and to get to others thoughts as well.

SECTION 4: ACCOMPLISHMENTS

4.1. Accomplishment 1

Title: Restaurant Database management System.

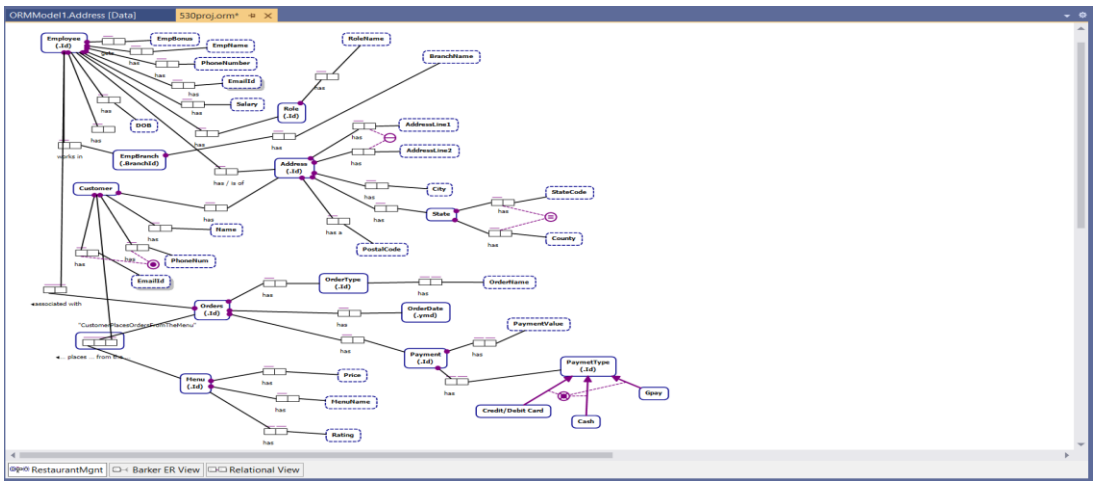
Subject: Advanced Database management Systems

Professor: Robert Rucker

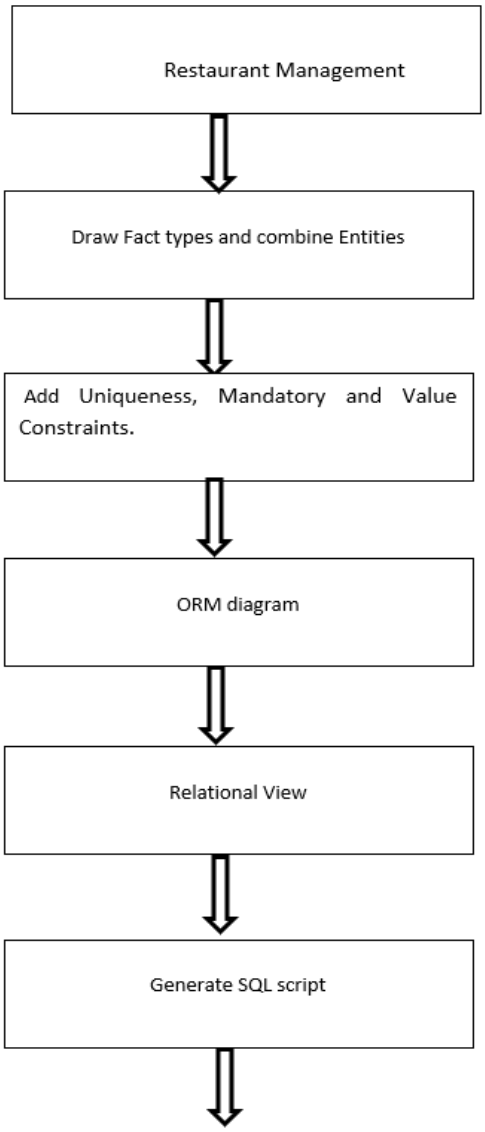
1. Introduction

A database management system is a sophisticated and well-structured computer software that seeks to arrange a database in a logical and user-friendly manner. Database management systems are intended to make the process of storing, processing, and retrieving data easier and more comfortable for all users. Database management systems are used by all organizations, particularly those in industry, education, and health care. The major purpose of this system is to assist restaurant administrators in managing the restaurant business as well as to assist customers in online ordering. We can also keep track of payments made by clients, such as payment value and payment alternatives (cash, card, online transfer, etc..,). This will help a restaurant's reputation and market worth, as well as its customer service.

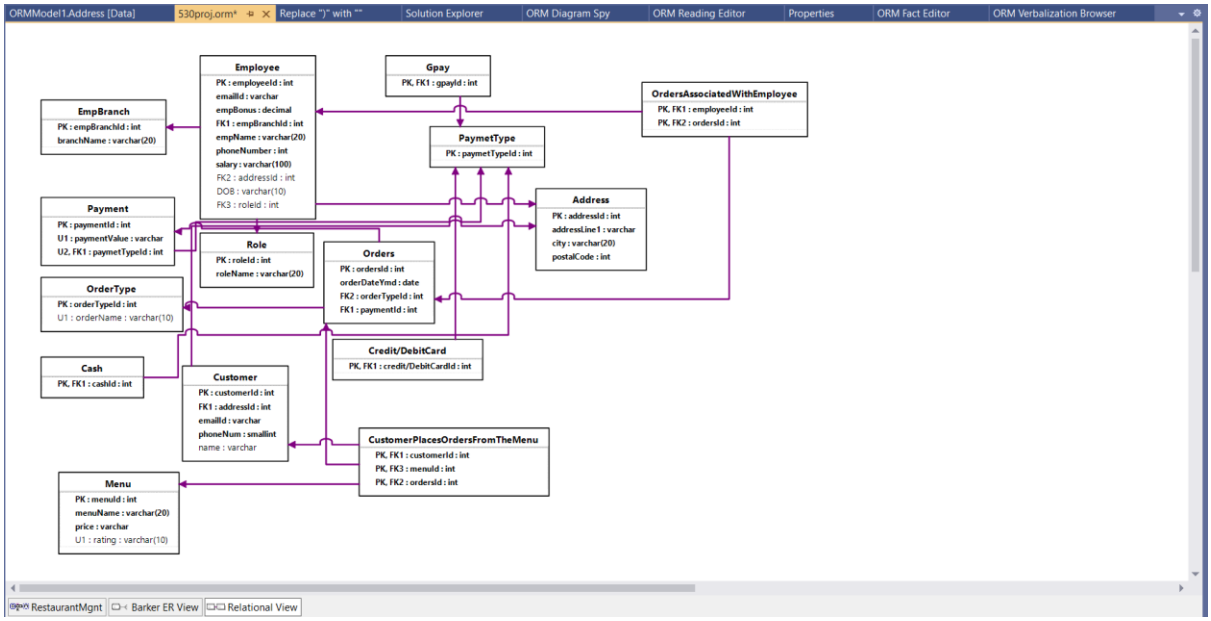
2. ORM Diagram



3. Job Flow

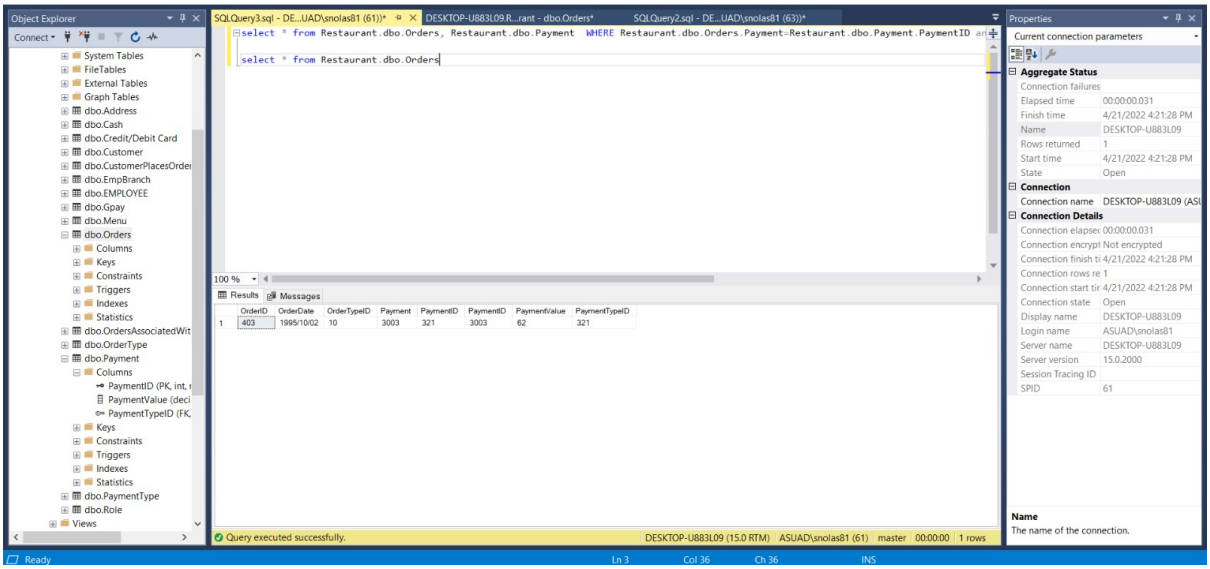


4. Relational Schema

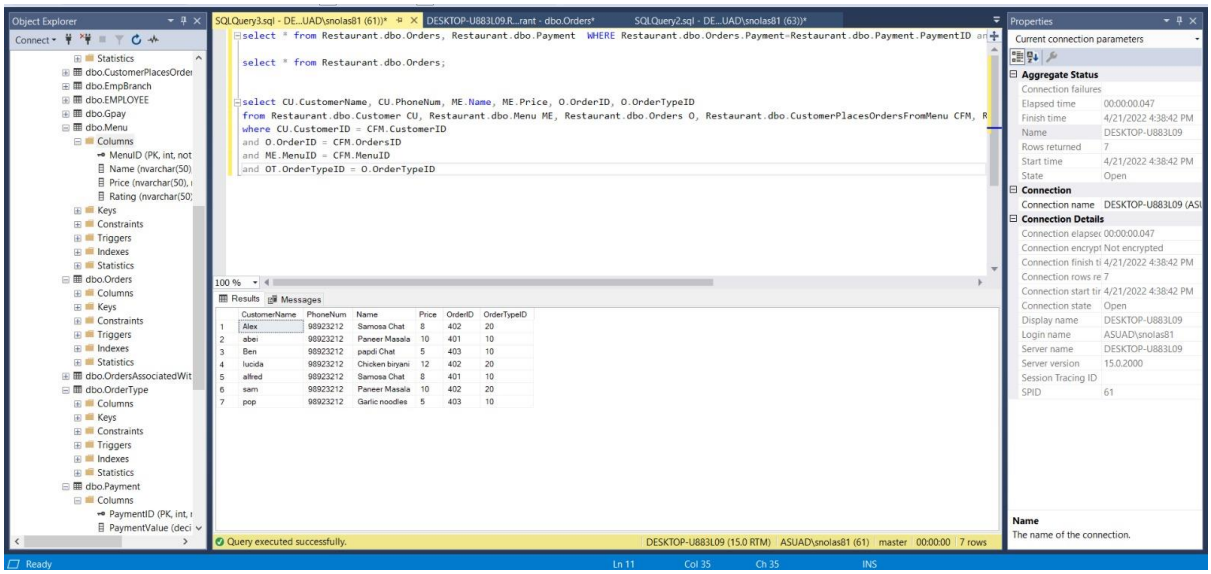


5. Basic Queries

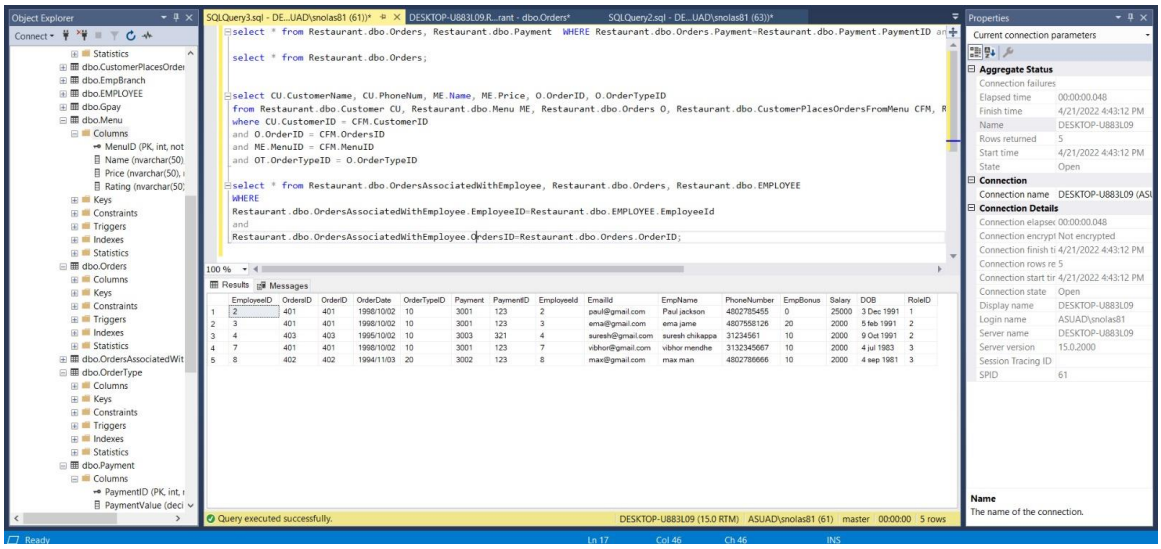
How to retrieve Order details of orders placed on some particular day and display order details along with payment details?



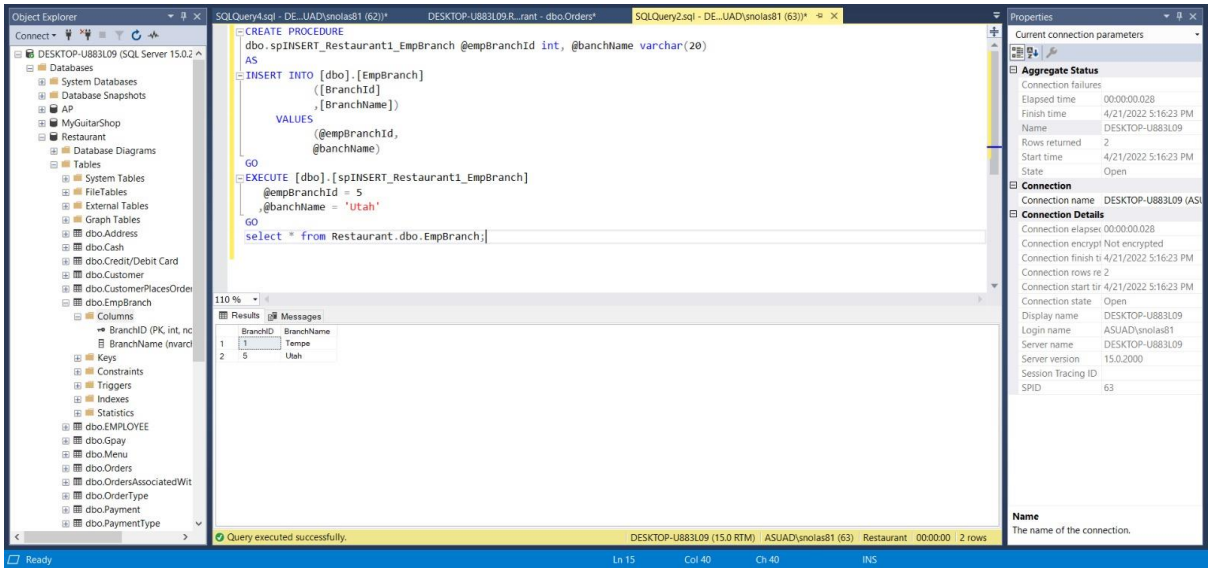
How to extract customers' order details, ordered by the customer?

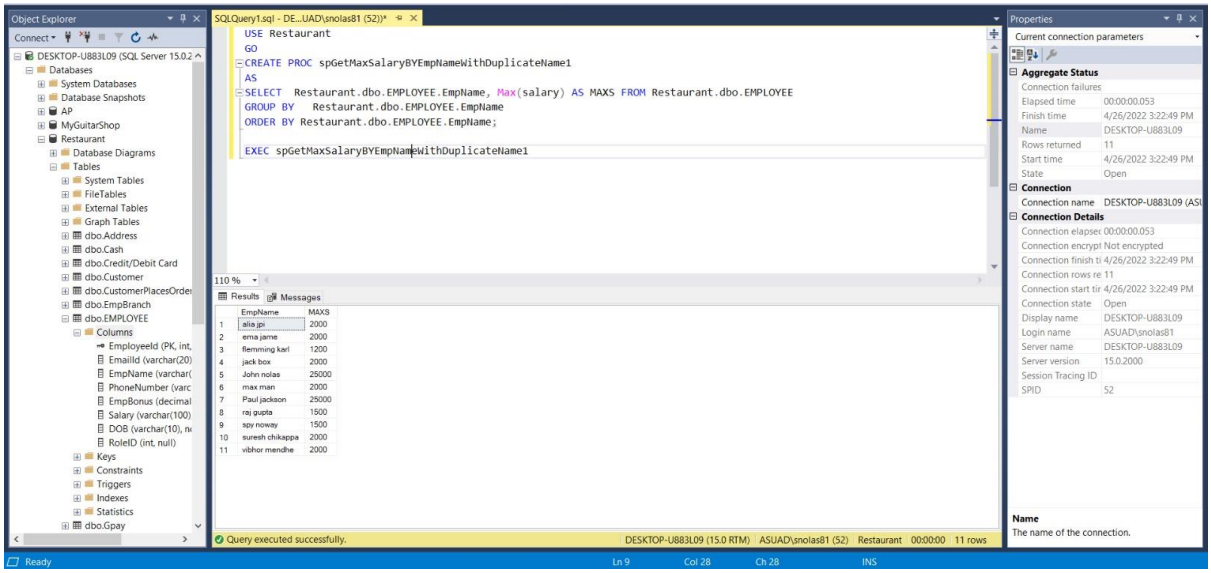


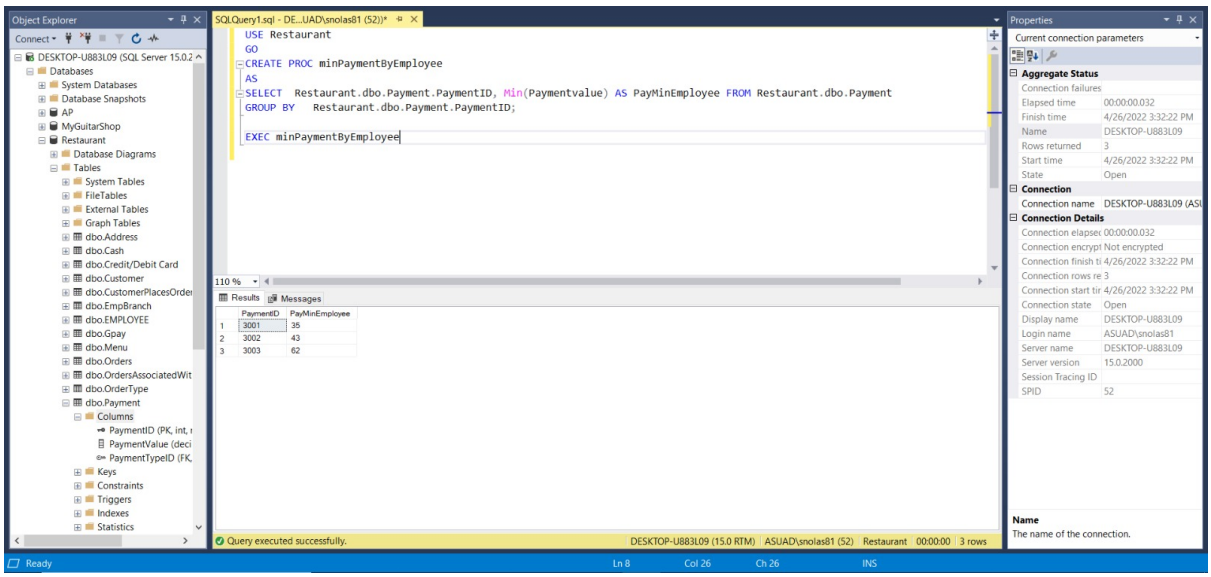
How to retrieve employee details along with order details which the employee has served?



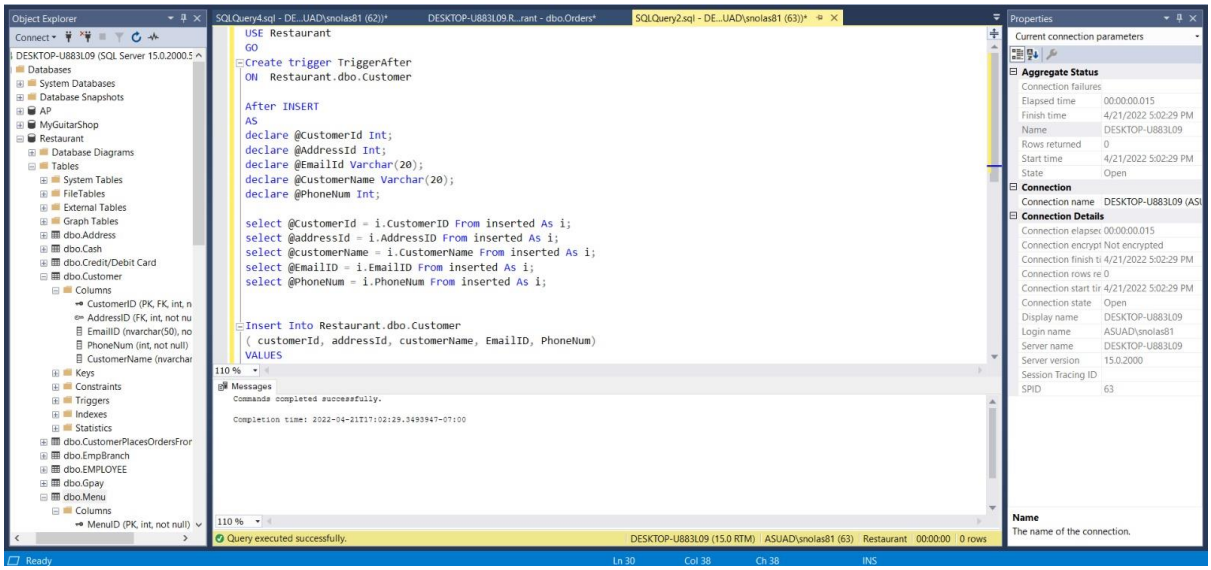
6. Stored Procedures

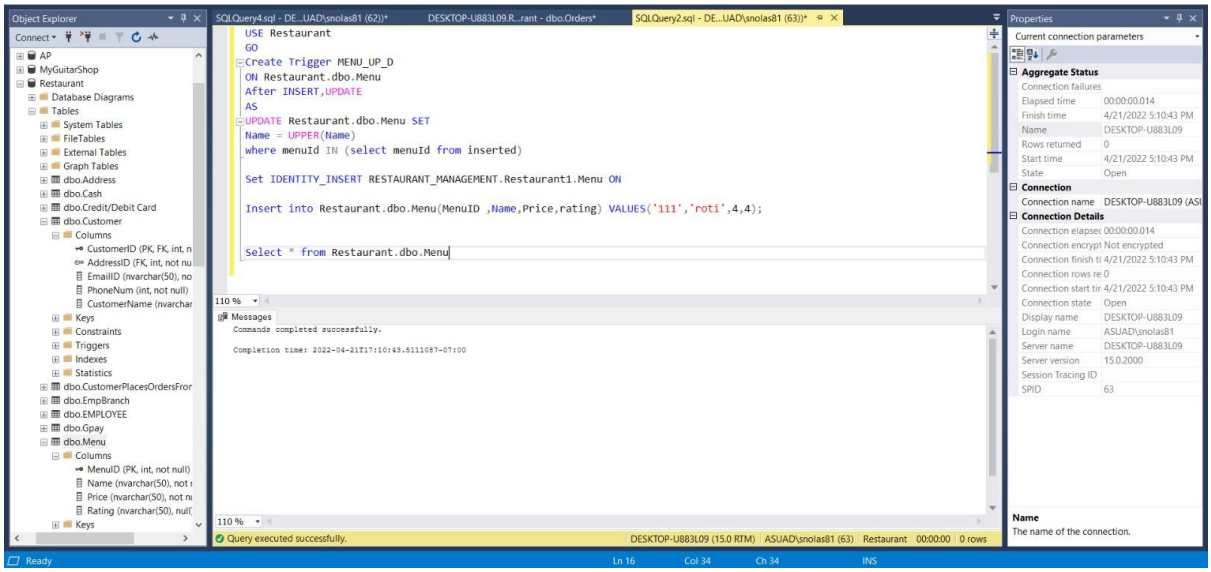

CREATE PROCEDURE
dbo.spINSERT_Restaurant1_EmpBranch @empBranchId int, @banchName varchar(20)
AS
INSERT INTO [dbo].[EmpBranch]
([BranchId]
,[BranchName])
VALUES
(@empBranchId,
@banchName)
GO
EXECUTE [dbo].[spINSERT_Restaurant1_EmpBranch]
@empBranchId = 5
,@banchName = 'Utah'
GO
select * from Restaurant.dbo.EmpBranch;
Query executed successfully.


USE Restaurant
GO
CREATE PROC spGetMaxSalaryBYEmpNameWithDuplicateName1
AS
SELECT Restaurant.dbo.EMPLOYEE.EmpName, Max(salary) AS MAXS FROM Restaurant.dbo.EMPLOYEE
GROUP BY Restaurant.dbo.EMPLOYEE.EmpName
ORDER BY Restaurant.dbo.EMPLOYEE.EmpName;
EXEC spGetMaxSalaryBYEmpNameWithDuplicateName1
Query executed successfully.


USE Restaurant
GO
CREATE PROC minPaymentByEmployee
AS
SELECT Restaurant.dbo.Payment.PaymentID, Min(Paymentvalue) AS PayMinEmployee FROM Restaurant.dbo.Payment
GROUP BY Restaurant.dbo.Payment.PaymentID;
EXEC minPaymentByEmployee
Query executed successfully.

7. Triggers

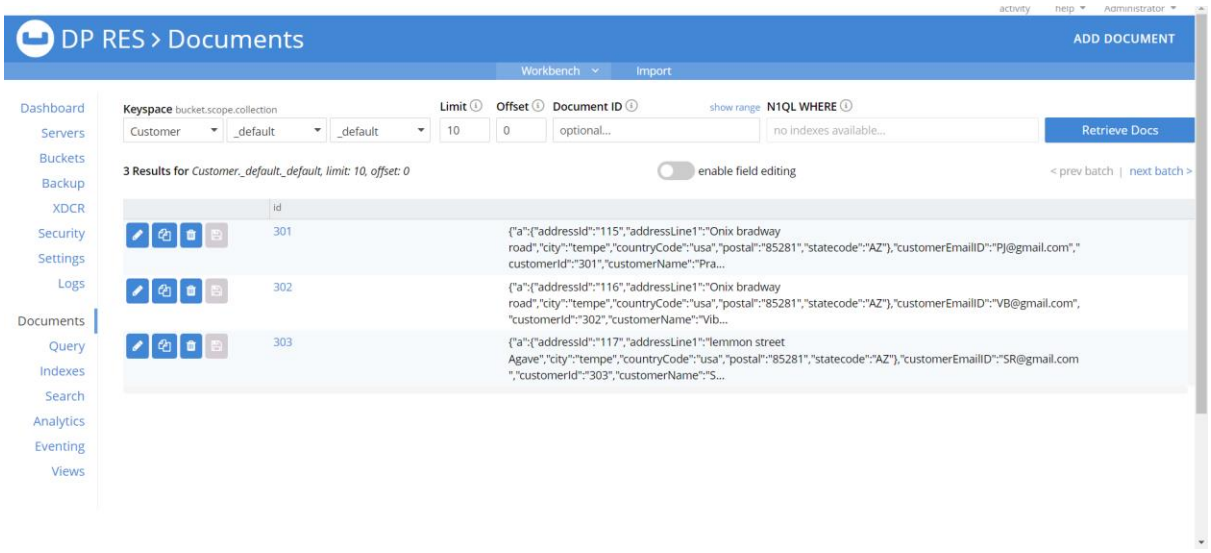




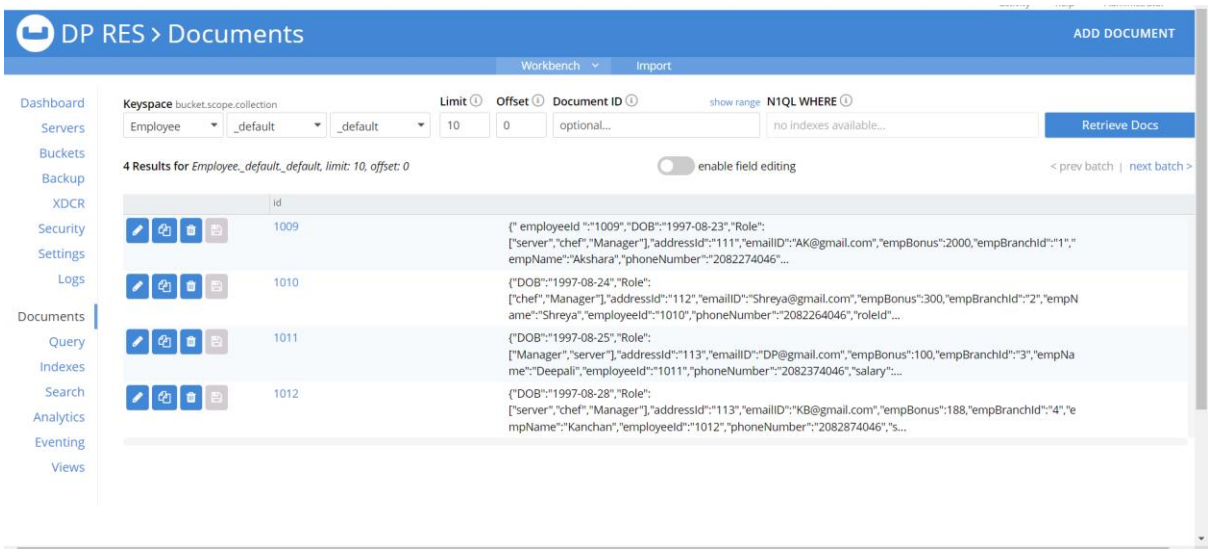
8. Couchbase

8.1. JSON Document

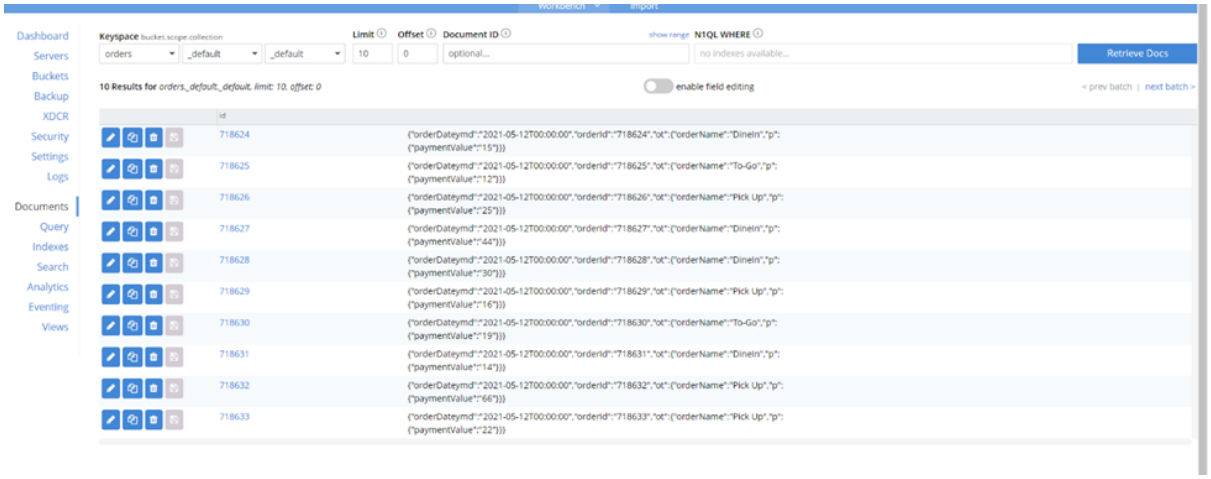
Customer



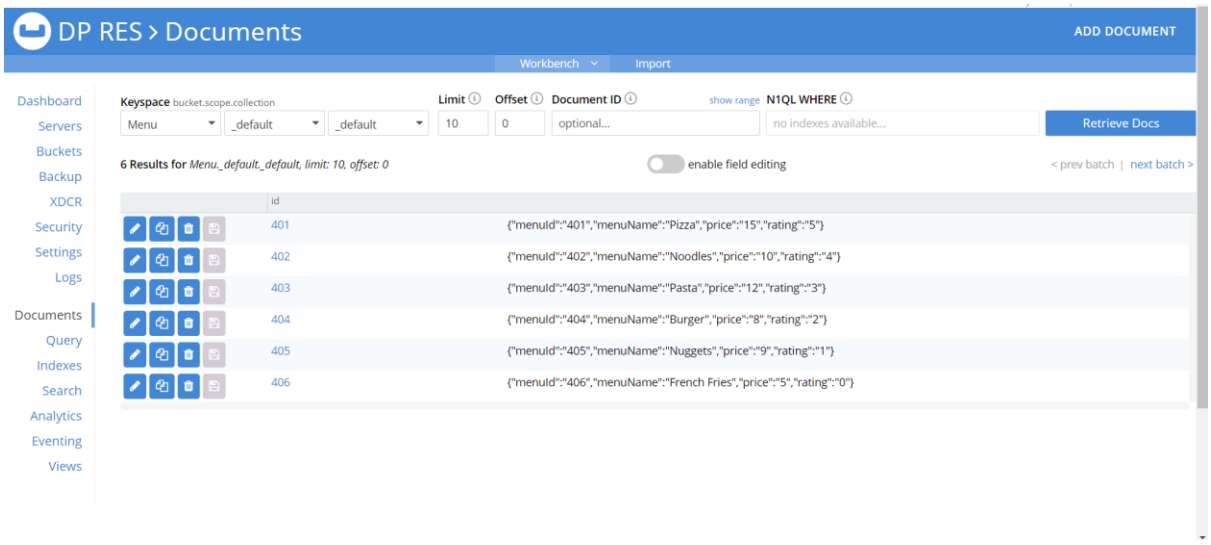
Employee



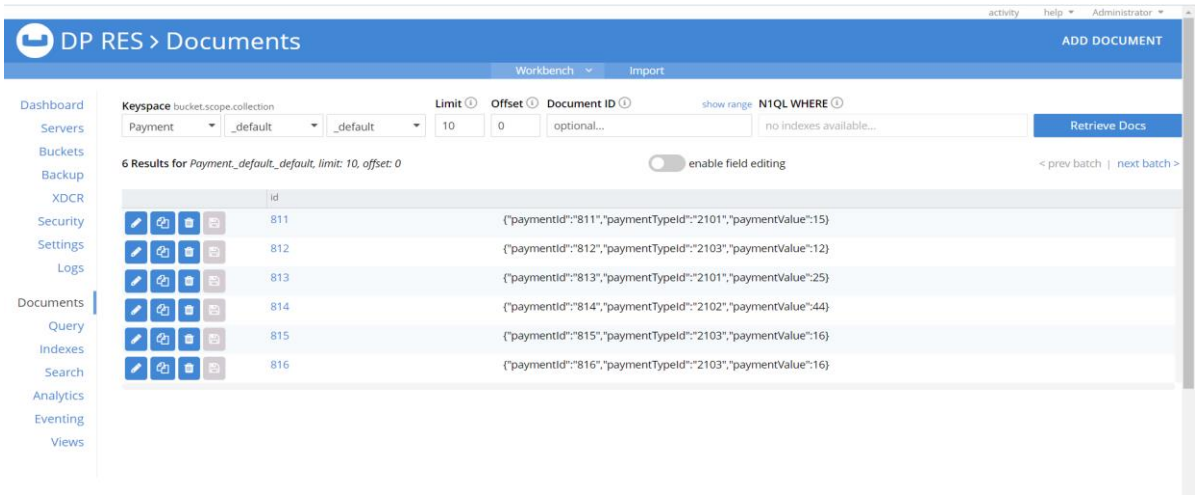
Orders



Menu

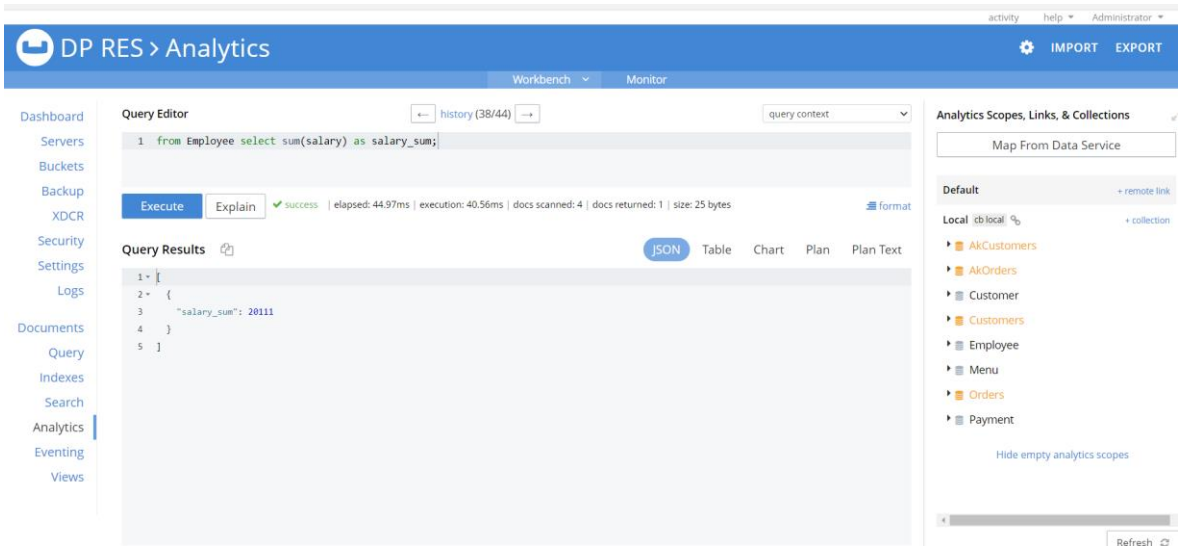


Payment

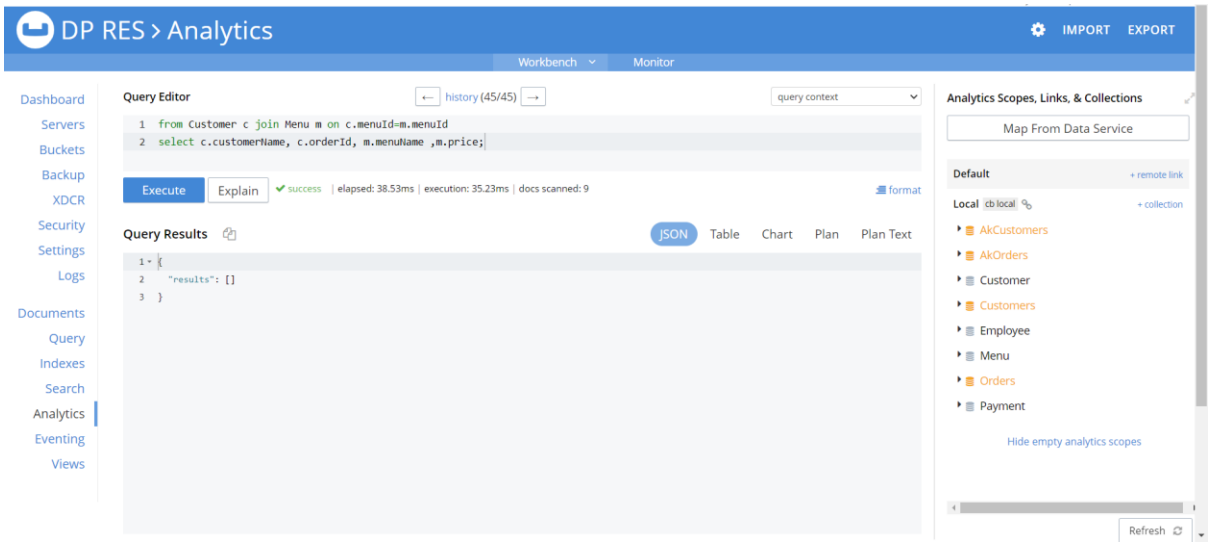


8.2. SQL++ Queries

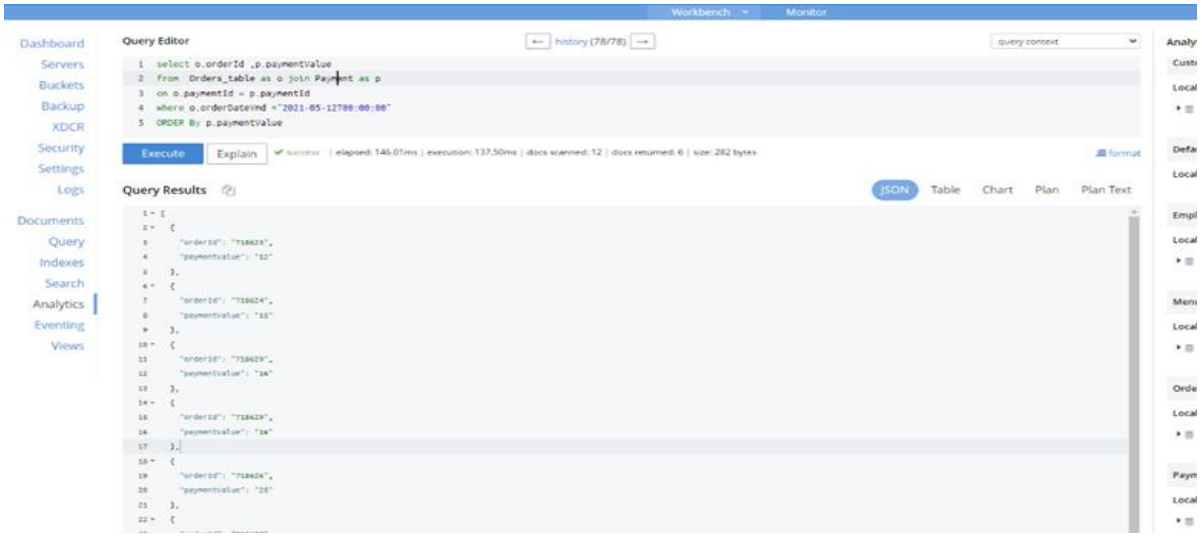
To get the Sum of salary of all Employees



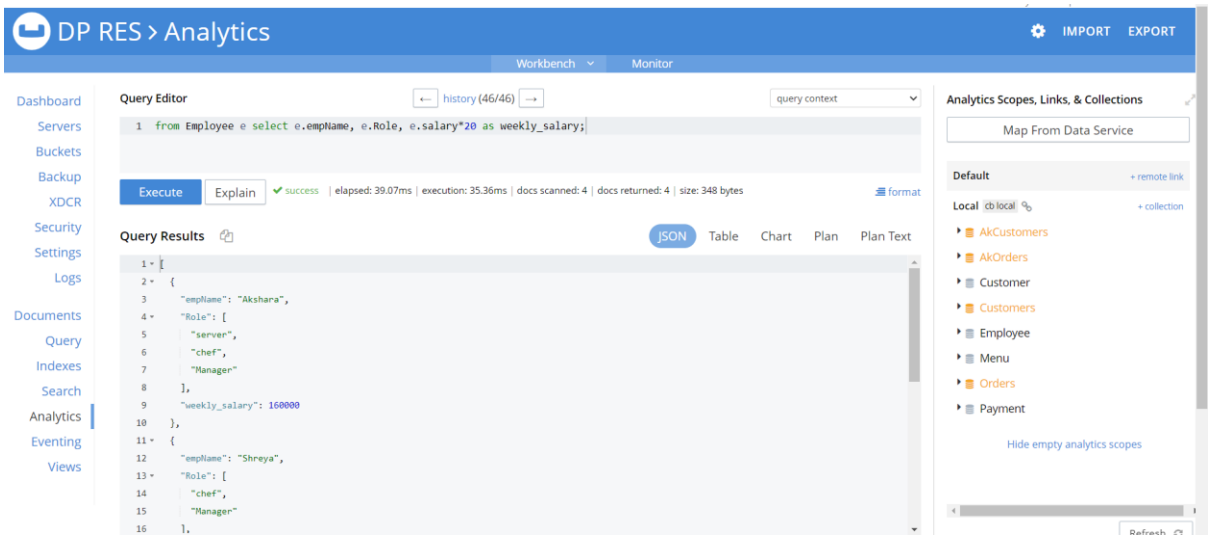
To Get Information about which customer Ordered what menu



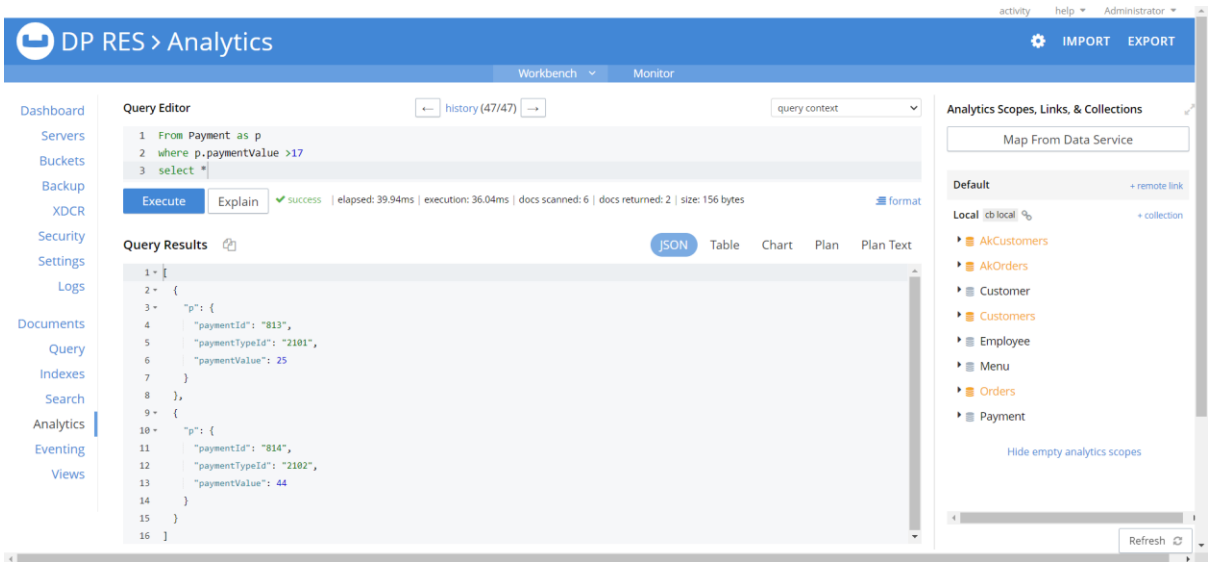
To get Information about order and Payment



To Get Information on Employee their role and weekly salary



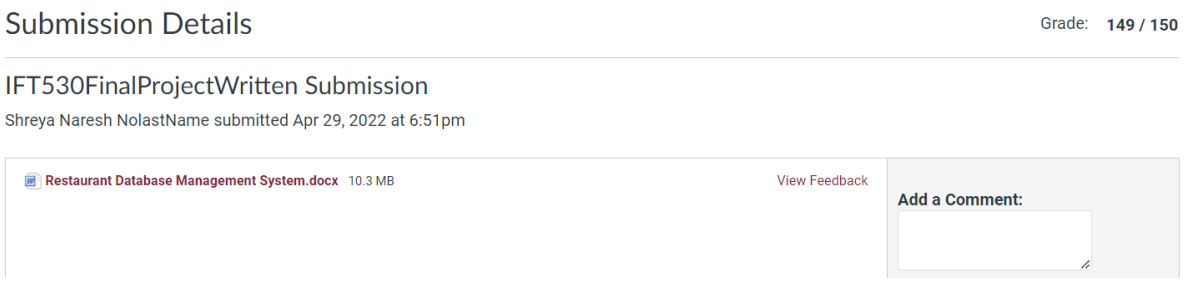
To Get Information about payment which has payment value greater than 17



9. Reflection about the Project

Database management systems are used by every organization, especially those in the domains of business, education, and health care. The major purpose of creating a Restaurant Management database is to keep track of personnel working in a restaurant, orders, client information, and customer ratings, all of which will help us grow our business. We can also keep track of consumer payments, such as the amount paid and the payment alternatives available.

10. Evidence of Accomplishment



4.2. Accomplishment 2

Title: ODI Batting cricket Analysis

Subject: Data visualization & Reporting in IT

Professor: Asmaa Elbadrawy

1. Introduction

Any aspirant data scientist will consider all information in light of the data. Even when relaxing with friends, enjoying live cricket, and supporting the preferred team. The content in our dataset includes the ODI cricket data with the batting data. We concluded that the ODI cricket batting dataset needs a proper dashboard so the ICC teams, R&D researchers, the international cricket board and all the players can analyse their batting performance over a span of years of the ODI seasons specifically for batting performance analyses. **The dashboard consists of 10 different visualizations as discussed below:**

Analysis of performance trends

Analysis of Innings

Average score of all players

Analysis of Active players with Durations

Analysis of Ducks to the 100s score

Analysis of Innings to not out Players

Highest score above 100

100s scored vs Not Outs (Innings)

Average score above 30

Analysis of ODI scores more than 10,000.

2. Data Pre-processing

1. The data collected from any source would be raw and needs to be preprocessed to remove any unnecessary columns. Our dataset consists of 17 columns and 2485 records. We need to pre-process the data before it can be processed for visualization.
2. We have used Python Pandas library for pre-processing the raw data in Jupyter Notebook. isnull() method is used to find any null columns and drop() to drop those columns.

	Unnamed: 0	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	Unnamed: 13
0	0	SR Tendulkar (INDIA)	1989-2012	463	452	41	18426	200*	44.83	21367	86.23	49	96	20	NaN
1	1	KC Sangakkara (Asia/ICC/SL)	2000-2015	404	380	41	14234	169	41.98	18048	78.86	25	93	15	NaN
2	2	RT Ponting (AUS/ICC)	1995-2012	375	365	39	13704	164	42.03	17046	80.39	30	82	20	NaN
3	3	ST Jayasuriya (Asia/SL)	1989-2011	445	433	18	13430	189	32.36	14725	91.2	28	68	34	NaN
4	4	DPMD Jayawardene (Asia/SL)	1998-2015	448	418	39	12650	144	33.37	16020	78.96	19	77	28	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2495	45	ZS Ansari (ENG)	2015-2015	1	-	-	-	-	-	-	-	-	-	-	NaN
2496	46	Ariful Haque (BDESH)	2018-2018	1	-	-	-	-	-	-	-	-	-	-	NaN
2497	47	Ashfaq Ahmed (PAK)	1994-1994	3	-	-	-	-	-	-	-	-	-	-	NaN
2498	48	MD Bailey (NZ)	1998-1998	1	-	-	-	-	-	-	-	-	-	-	NaN
2499	49	GR Beard (AUS)	1981-1981	2	-	-	-	-	-	-	-	-	-	-	NaN

2500 rows × 15 columns

```
In [41]: [col for col in ODI_data.columns if ODI_data[col].isnull().sum() > 0]
```

```
Out[41]: ['Unnamed: 13']
```

```
In [42]: ODI_data.drop('Unnamed: 13',axis=1,inplace=True)
         ODI_data.drop('Unnamed: 0',axis=1,inplace=True)
         ODI_data
```

Out[42]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0
0	SR Tendulkar (INDIA)	1989-2012	463	452	41	18426	200*	44.83	21367	86.23	49	96	20
1	KC Sangakkara (Asia/ICC/SL)	2000-2015	404	380	41	14234	169	41.98	18048	78.86	25	93	15
2	RT Ponting (AUS/ICC)	1995-2012	375	365	39	13704	164	42.03	17046	80.39	30	82	20
3	ST Jayasuriya (Asia/SL)	1989-2011	445	433	18	13430	189	32.36	14725	91.2	28	68	34

3. We have used str.split() function to split the Player column into Regions and span column into start year and end year as shown in the screenshot below:

```
In [62]: ODI_data[['Player', 'Region']] = ODI_data['Player'].str.split('(', n=1, expand=True)
```

```
In [63]: ODI_data['Region'] = ODI_data['Region'].map(lambda x: x.rstrip(')'))
         ODI_data[['Start Year', 'Last Year']] = ODI_data['Span'].str.split('-', n=1, expand=True)
         ODI_data[['Region1', 'Region']] = ODI_data['Region'].str.split('/', n=1, expand=True)
         ODI_data
```

Out[63]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	Region	Start Year	Last Year	Region1
0	SR Tendulkar	1989-2012	463	452	41	18426	200*	44.83	21367	86.23	49	96	20	None	1989	2012	INDIA
1	KC Sangakkara	2000-2015	404	380	41	14234	169	41.98	18048	78.86	25	93	15	ICC/SL	2000	2015	Asia
2	RT Ponting	1995-2012	375	365	39	13704	164	42.03	17046	80.39	30	82	20	ICC	1995	2012	AUS
3	ST Jayasuriya	1989-2011	445	433	18	13430	189	32.36	14725	91.2	28	68	34	SL	1989	2011	Asia
4	DPMD Jayawardene	1998-2015	448	418	39	12650	144	33.37	16020	78.96	19	77	28	SL	1998	2015	Asia
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2495	ZS Ansari	2015-2015	1	-	-	-	-	-	-	-	-	-	-	None	2015	2015	ENG
2496	Ariful Haque	2018-2018	1	-	-	-	-	-	-	-	-	-	-	None	2018	2018	BDESH
2497	Ashfaq Ahmed	1994-1994	3	-	-	-	-	-	-	-	-	-	-	None	1994	1994	PAK
2498	MD Bailey	1998-1998	1	-	-	-	-	-	-	-	-	-	-	None	1998	1998	NZ
2499	GR Beard	1981-1981	2	-	-	-	-	-	-	-	-	-	-	None	1981	1981	AUS

2500 rows × 17 columns

```
In [81]: O_bat_data.head(10)
```

4. From 2 regions we are filtering the `Final region`, by using the python function as given below.

```
In [71]: ODI_data['Final Region'] = ODI_data.apply(new_co, axis=1)
```

```
In [72]: ODI_data.head(60)
```

Out[72]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	Region	Start Year	Last Year	Region1	Region2	Final Region
0	SR Tendulkar	1989-2012	463	452	41	18426	200*	44.83	21367	86.23	49	96	20	None	1989	2012	INDIA	None	INDIA
1	KC Sangakkara	2000-2015	404	380	41	14234	169	41.98	18048	78.86	25	93	15	SL	2000	2015	Asia	ICC	SL
2	RT Ponting	1995-2012	375	365	39	13704	164	42.03	17046	80.39	30	82	20	None	1995	2012	AUS	ICC	AUS
3	ST Jayasuriya	1989-2011	445	433	18	13430	189	32.36	14725	91.2	28	68	34	None	1989	2011	Asia	SL	SL
4	DPMD Jayawardene	1998-2015	448	418	39	12650	144	33.37	16020	78.96	19	77	28	None	1998	2015	Asia	SL	SL
5	Inzamam-ul-Haq	1991-2007	378	350	53	11739	137*	39.52	15812	74.24	10	83	20	None	1991	2007	Asia	PAK	PAK
6	V Kohli	2008-2019	242	233	39	11609	183	59.84	12445	93.28	43	55	13	None	2008	2019	INDIA	None	INDIA

```
In [73]: print(ODI_data['Final Region'].unique())
```

```
['INDIA' 'SL' 'AUS' 'PAK' 'SA' 'WI' 'NZ' 'ENG' 'BDESH' 'ZIM' 'IRE' 'KENYA'
 'AFG' 'SCOT' 'CAN' 'NL' 'UAE' 'HKG' 'BMUDA' 'PNG' 'USA' 'NAM' 'NEPAL'
 'OMAN' '3) (PAK' None '1) (UAE']
```

```
In [74]: ODI_data[ODI_data['Final Region'].isna()]
```

Out[74]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	Region	Start Year	Last Year	Region1	Region2	Final Region
1497	Frasat Ali	1975-1975	3	3	0	57	45	19.00	176	32.38	0	0	1	None	1975	1975	EAf	None	None
1524	RK Sethi	1975-1975	3	3	0	54	30	18.00	185	29.18	0	0	0	None	1975	1975	EAf	None	None
1591	Jawahir Shah	1975-1975	3	3	0	46	37	15.33	114	40.35	0	0	0	None	1975	1975	EAf	None	None

5. Further we can drop the tables created during preprocessing to get the Final Region and also get unique values in the Final region and drop the rows which contains invalid country names.

```
In [76]: ODI_data=ODI_data[ODI_data['Final Region'].isnull()==False]

In [21]: ODI_data=ODI_data.drop('Region',axis=1)
         ODI_data=ODI_data.drop('Region1',axis=1)
         ODI_data=ODI_data.drop('Region2',axis=1)

In [22]: print(ODI_data['Final Region'].unique())

['INDIA' 'SL' 'AUS' 'PAK' 'SA' 'WI' 'NZ' 'ENG' 'BDESH' 'ZIM' 'IRE' 'KENYA'
 'AFG' 'SCOT' 'CAN' 'NL' 'UAE' 'HKG' 'BMUDA' 'PNG' 'USA' 'NAM' 'NEPAL'
 'OMAN' '3) (PAK' '1) (UAE']

In [77]: ODI_data[ODI_data['Final Region'].str.contains('1')]
```

Out[77]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	Region	Start Year	Last Year	Region1	Region2	Final Region
2148	Asif Iqbal	2015-2015	1	1	0	8	8	8.00	8	100.0	0	0	0	None	2015	2015	1) (UAE	None	1) (UAE

```
In [78]: ODI_data=ODI_data.drop(2148)

In [79]: ODI_data[ODI_data['Final Region'].str.contains('3')]
```

Out[79]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	Region	Start Year	Last Year	Region1	Region2	Final Region
949	Mohammad Nawaz	2016-2019	15	12	3	199	53	22.11	213	93.42	0	1	0	None	2016	2019	3) (PAK	None	3) (PAK

```
In [80]: ODI_data=ODI_data.drop(949)
         print(ODI_data['Final Region'].unique())

['INDIA' 'SL' 'AUS' 'PAK' 'SA' 'WI' 'NZ' 'ENG' 'BDESH' 'ZIM' 'IRE' 'KENYA'
 'AFG' 'SCOT' 'CAN' 'NL' 'UAE' 'HKG' 'BMUDA' 'PNG' 'USA' 'NAM' 'NEPAL'
 'OMAN']
```

6. Finally the columns in which '-' is present is replaced by '0' using str.replace() and relevant columns such as 'Average', 'Strike Rate' etc. are converted to float type.

```
In [81]: ODI_data['Runs']=ODI_data['Runs'].str.replace('-','0')
         ODI_data['HS']=ODI_data['HS'].str.replace('*','')
         ODI_data['Ave']=ODI_data['Ave'].str.replace('-','0')
         ODI_data['SR']=ODI_data['SR'].str.replace('-','0')
         ODI_data['Inns']=ODI_data['Inns'].str.replace('-','0')
         ODI_data['NO']=ODI_data['NO'].str.replace('-','0')
         ODI_data['HS']=ODI_data['HS'].str.replace('-','0')
         ODI_data['BF']=ODI_data['BF'].str.replace('-','0')
         ODI_data['100']=ODI_data['100'].str.replace('-','0')
         ODI_data['50']=ODI_data['50'].str.replace('-','0')
         ODI_data['0']=ODI_data['0'].str.replace('-','0')

In [87]: ODI_data
```

Out[87]:

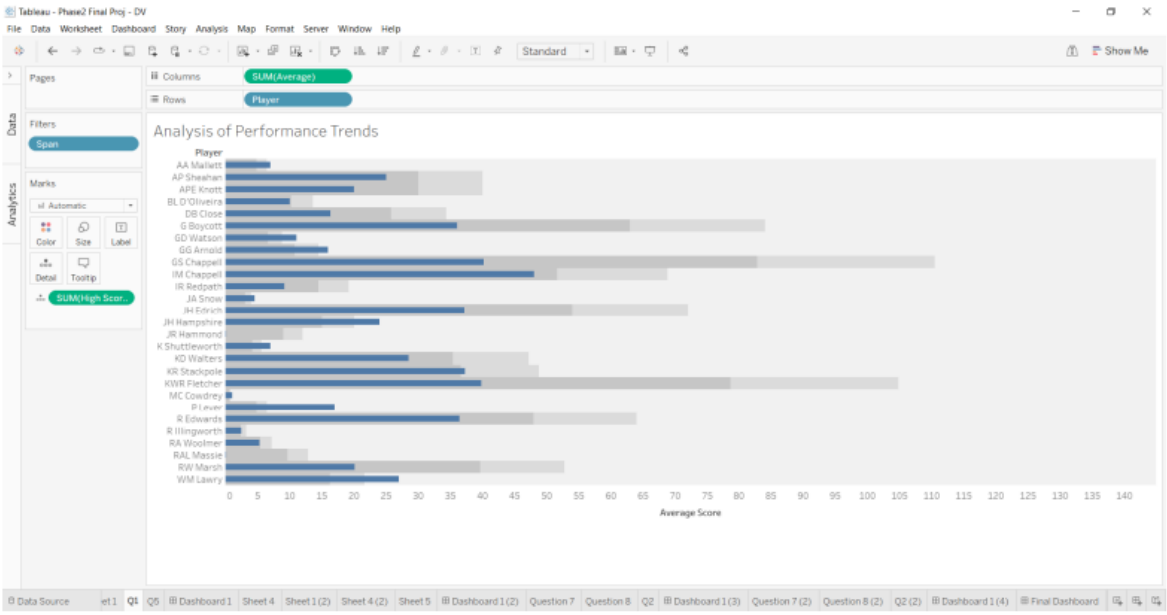
	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	Start Year	Last Year	Final Region	Duration
0	SR Tendulkar	1989-2012	463	452	41	18426	200	44.83	21367	86.23	49	96	20	1989	2012	INDIA	23
1	KC Sangakkara	2000-2015	404	380	41	14234	169	41.98	18048	78.86	25	93	15	2000	2015	SL	15
2	RT Ponting	1995-2012	375	365	39	13704	164	42.03	17046	80.39	30	82	20	1995	2012	AUS	17
3	ST Jayasuriya	1989-2011	445	433	18	13430	189	32.36	14725	91.2	28	68	34	1989	2011	SL	22
4	DPMD Jayawardene	1998-2015	448	418	39	12650	144	33.37	16020	78.96	19	77	28	1998	2015	SL	17
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2495	ZS Ansari	2015-2015	1	0	0	0	0	0	0	0	0	0	0	2015	2015	ENG	0
2496	Ariful Haque	2018-2018	1	0	0	0	0	0	0	0	0	0	0	2018	2018	BDESH	0
2497	Ashfaq Ahmed	1994-1994	3	0	0	0	0	0	0	0	0	0	0	1994	1994	PAK	0
2498	MD Bailey	1998-1998	1	0	0	0	0	0	0	0	0	0	0	1998	1998	NZ	0
2499	GR Beard	1981-1981	2	0	0	0	0	0	0	0	0	0	0	1981	1981	AUS	0

2485 rows × 17 columns

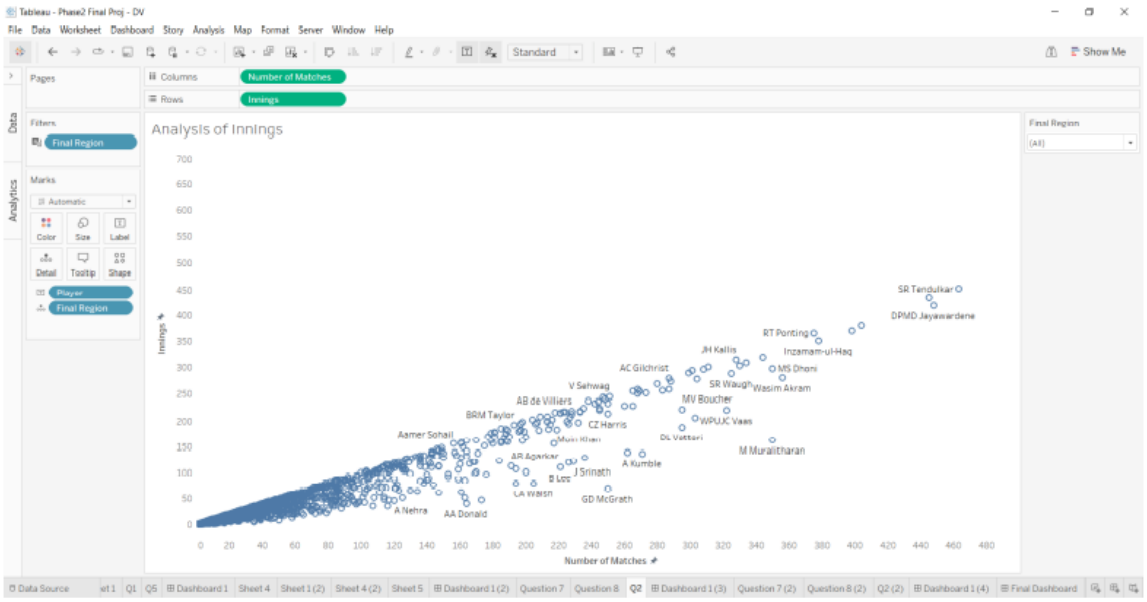
```
In [33]: O_bat_data['Ave']=O_bat_data['Ave'].astype('float')
         O_bat_data['HS']=O_bat_data['HS'].astype('float')
```

3. Dashboard Plots & Interactivity

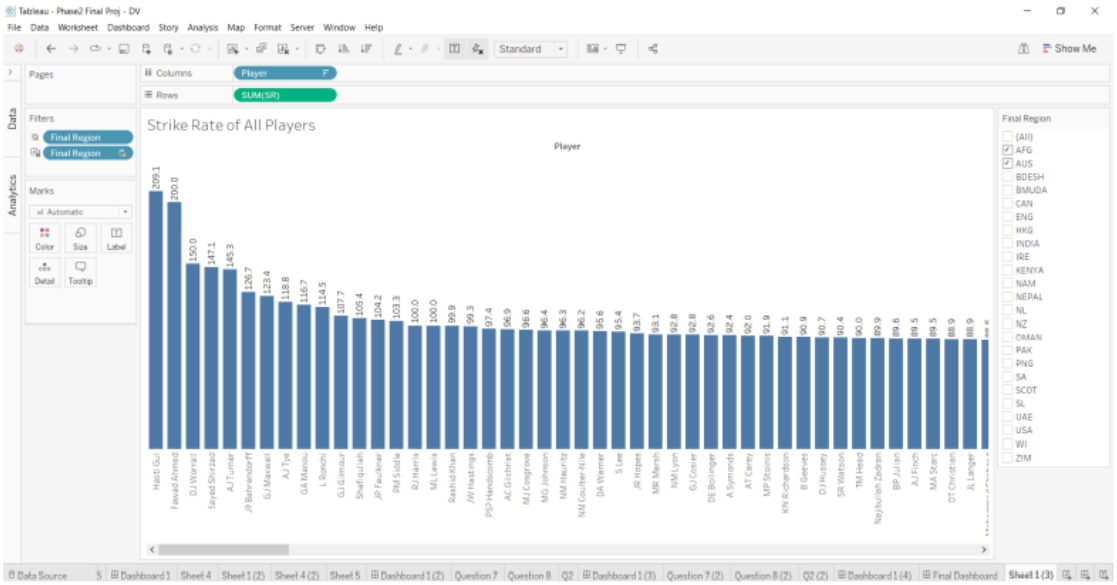
1. A Horizontal bar chart is used to display the average sum of scores to all the players over a period of year. We further filter it by span to allow the user to check the filter which is needed at the particular time as per any player. It also marks the sum of highest scores and gives a median analysis of the data to view the median of the sum of scores scored by the players over the span.

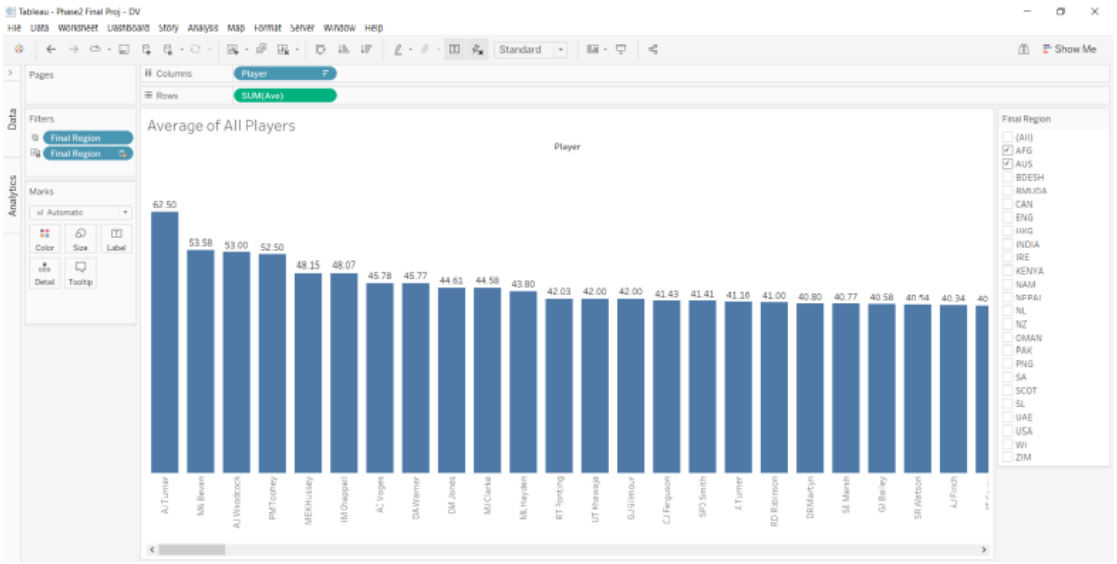


2. The below visualization filters the players played in a particular country with their number of innings marked as labels. Further, to compare the innings with the number of matches played we have used the Sum function with the number of matches played.

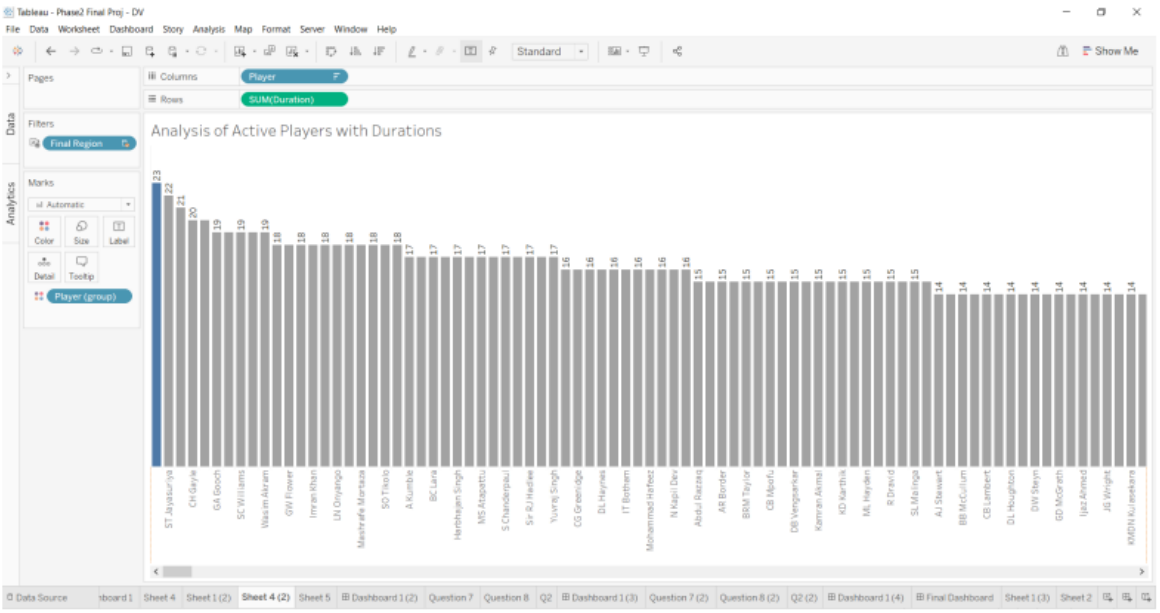


3. A dashboard has been created to show the strike rate and the average of all the players in the ODI. A filter has been created to select the specific country that needs to be shown. 3.1 sheet shows the players with higher strike rate shown in descending order. 3.2 sheet shows the players with higher average for the selected country.

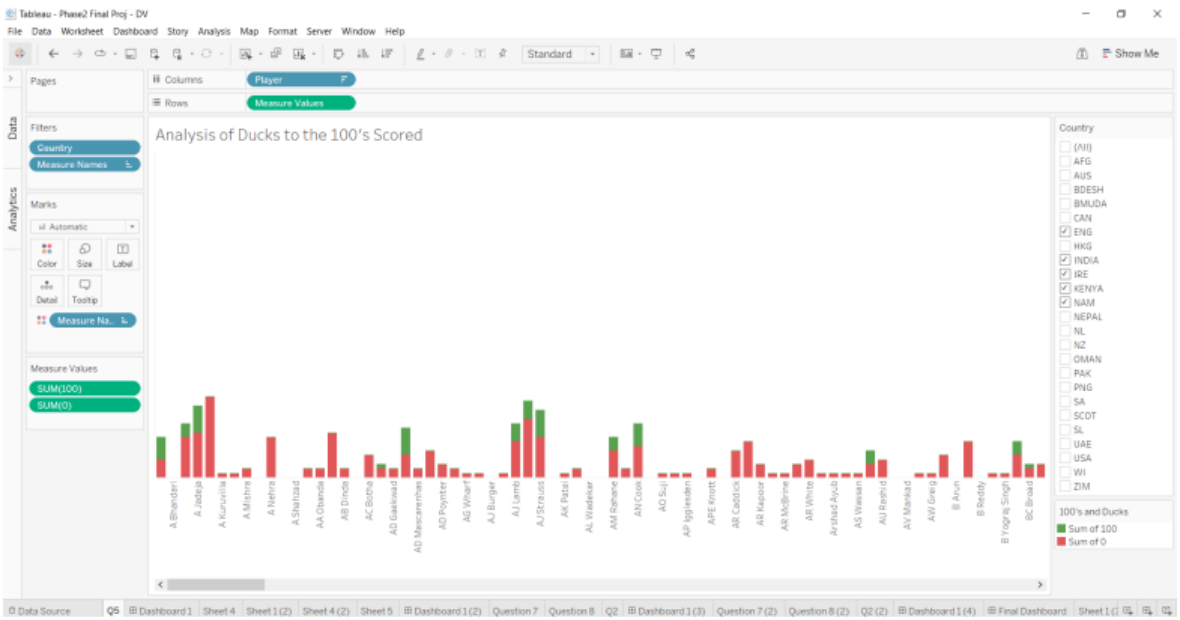




4. A vertical bar chart is used to display the duration of which each player has played in the ODI. Duration is found for each player as explained above in the preprocessing steps. The player with the greatest number of durations has been highlighted to get the viewers' attention.

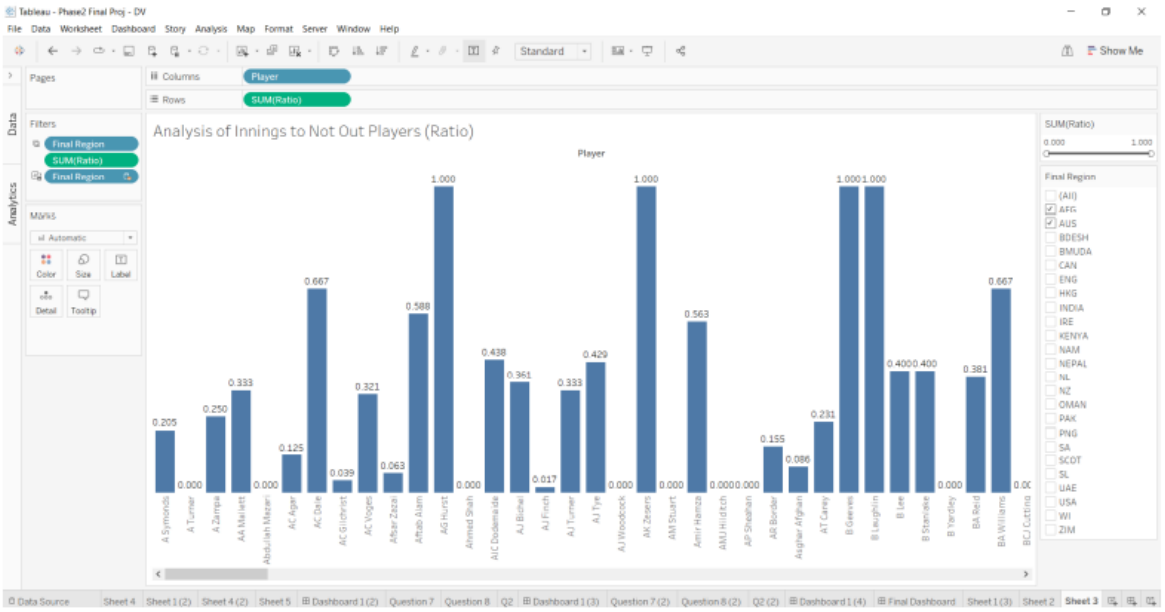


5. A stacked bar chart has been created to show the number of ducks and the 100s scored by a player. The players in a specific country can be selected by using the filter. The number of docks has been stacked over the number of 100s in each bar. The legend in the chart shows the color mentioned in the bar.

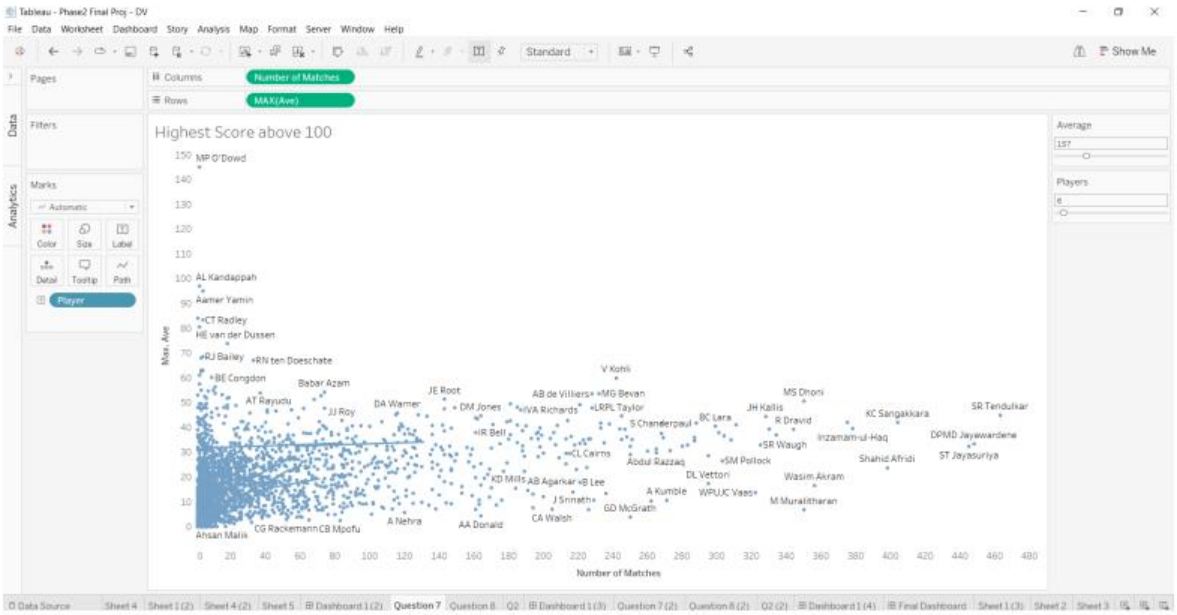


6. The vertical bar chart is created to show the ratio of the number of innings played by each player to the number of innings the player has stayed not out. A calculated field is

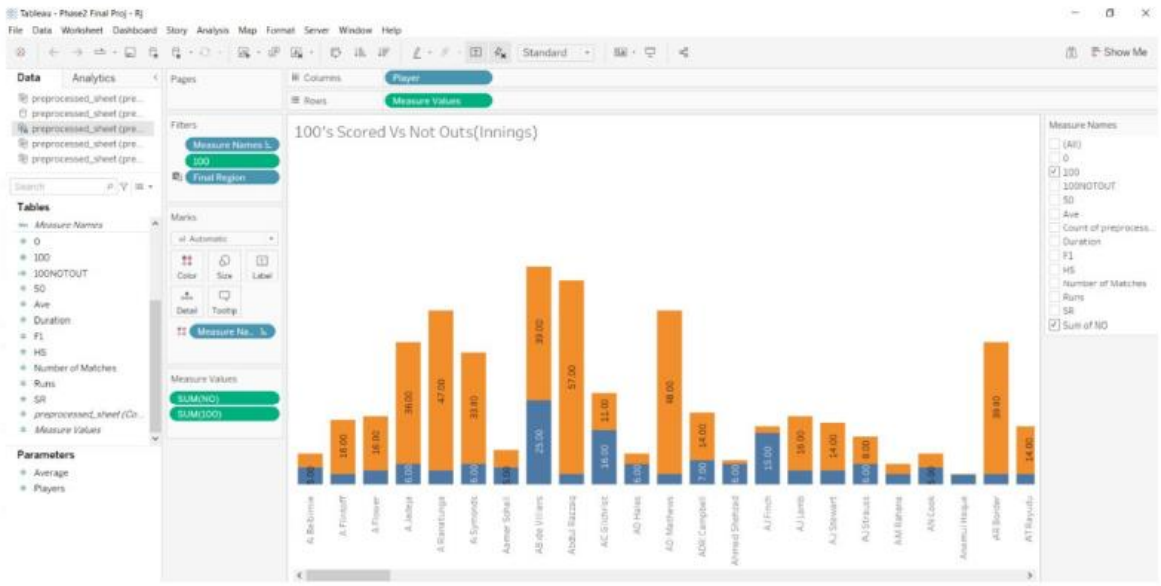
created to calculate the ratio in all the players. A filter is also created to select the country of the players.



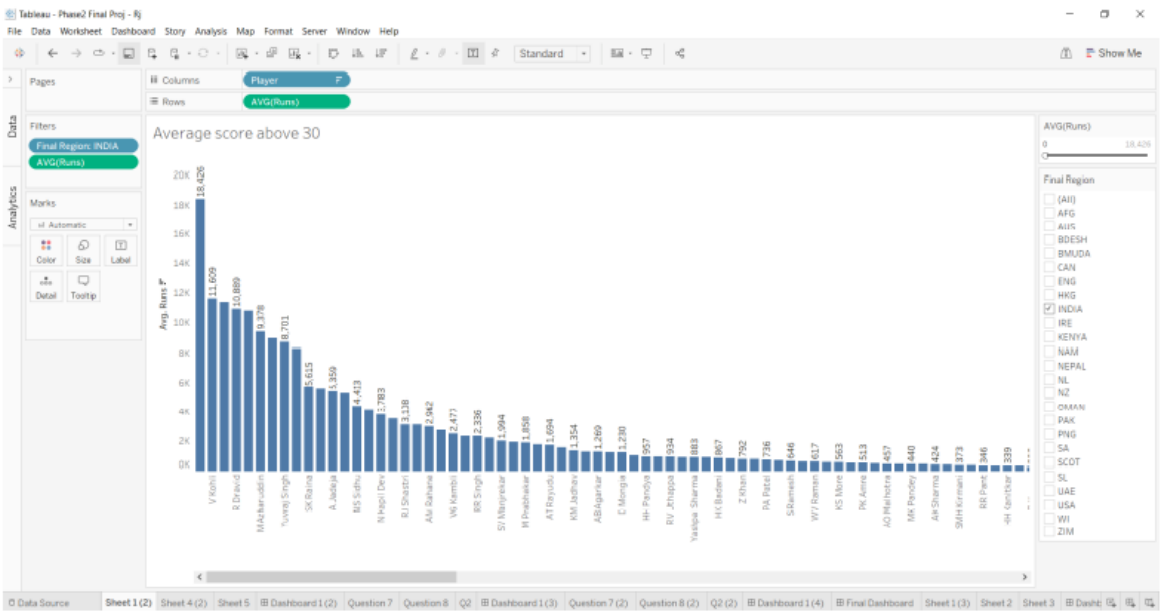
7. The vertical bar chart displays the players with the highest score more than 100. A filter has been created to show the players scored above 100 and the visualizations show how much runs the players have scored above 100 for the selected country.



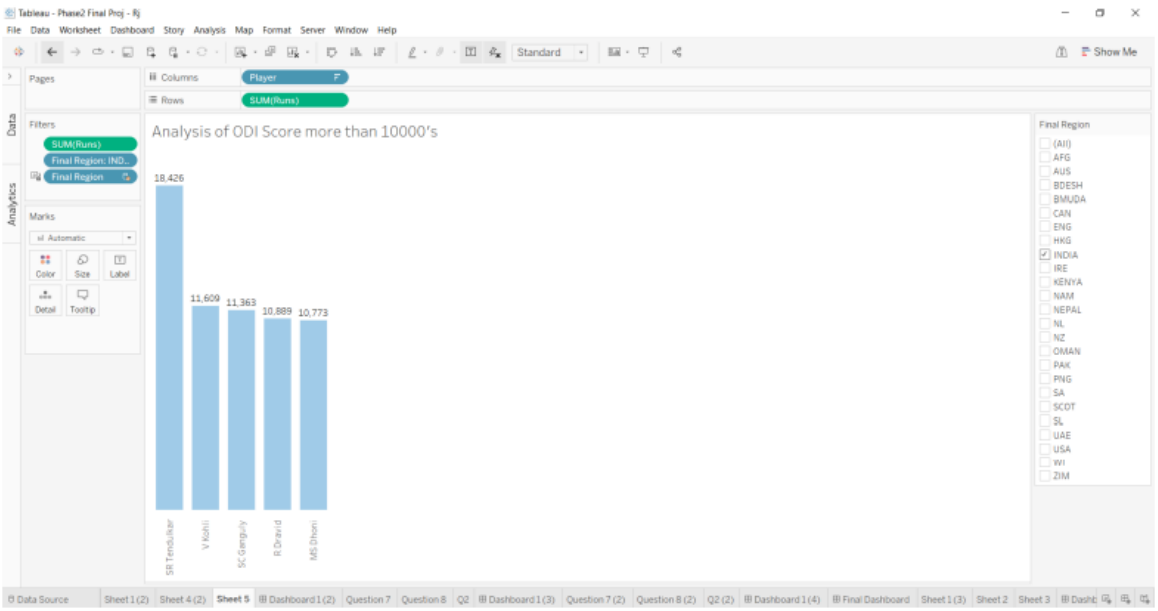
8. The stacked bar chart shows the number of hundreds and the number of innings the players have stayed not out. A filter has been created and selected the hundreds and the not out to show them on a stacked bar chart. The number of not out have been stacked over the number of hundreds.



9. In this bar chart the players with the average score more than 30 are displayed. The filter has been made in the average section and filled 30 in the minimum field so that it will show the players above 30 in the average.



10. The bar chart displays the players who have scored above 10,000 runs in the selected team. Two filters have been used. One is to select the country and the other is to show the minimum of 10,000 runs. 10,000 has been entered in the minimum field to show above the 10,000 runs.



Interactivity

Dashboards are important for viewing performance metrics, identifying profitability, and other visualizations.

● We try to maximize the data-ink ratio and eliminate as much Chart junk as possible before adding interactivity.

● We created Parameter fields and calculated fields so we can add some conditional statements to the questions we have addressed for our dataset.

● Dashboards are important for viewing performance metrics, identifying profitability, and so on.

The following filters have been used in our visualizations:

1. Filter by Country: We try to filter the dataset by country example, India or any other country. We have used this filter as it requires a set of players and their ratios divided as per the country.

2. Filter by Player: We have added questions as per all the players but we have filtered the data per every player so the visualization becomes more interactive as with the help of a calculated field we can change the range of players from 1 to 100 or any value we require through the slider that the parameter shows.

3. Filter by Measure Values: Measure names are measures of the data that are in the numerical order where the data is correlated or is to be compared. All of the measurements in your data are gathered into a single variable called Measure Values, which has continuous values. To remove a measure field from view, drag it out of the Measure Values card. We calculate the total 100s and total 0s scored by every player by filtering them as per Measure values.

4. Filter by Measure Names: The names of all the measures in your data are gathered into a single variable called Measure Names, which has discrete values. We use this filterto plot a stacked bar chart to represent the total innings with the no out players.

5. Filter by Average: Filter by average is the average runs scored by the players. We

have used this as it requires the average score of every player to be filtered as per their country. We use this filter to calculate the maximum score of the players. In question 9 we find the sum of all the average scores with all the players.

6. Filter by ratio: Question 6 uses this filter to calculate the sum of the ratios per every innings played as per every match and compares it with the players. It uses a condition that if the attribute is not equal to the innings, then it is divided as per the no out to the innings else gives one value as per the region classification.

7. Filter by Span: In the 1st question we split the columns as per the start year and the end year to show the sum of high scores comparing every player year wise.

4. Final Dashboard

After all the data pre-processing, cleaning and migrating data and joining them as per our visualization we created the below dashboard that we published on Mural and Tableau Public as well.

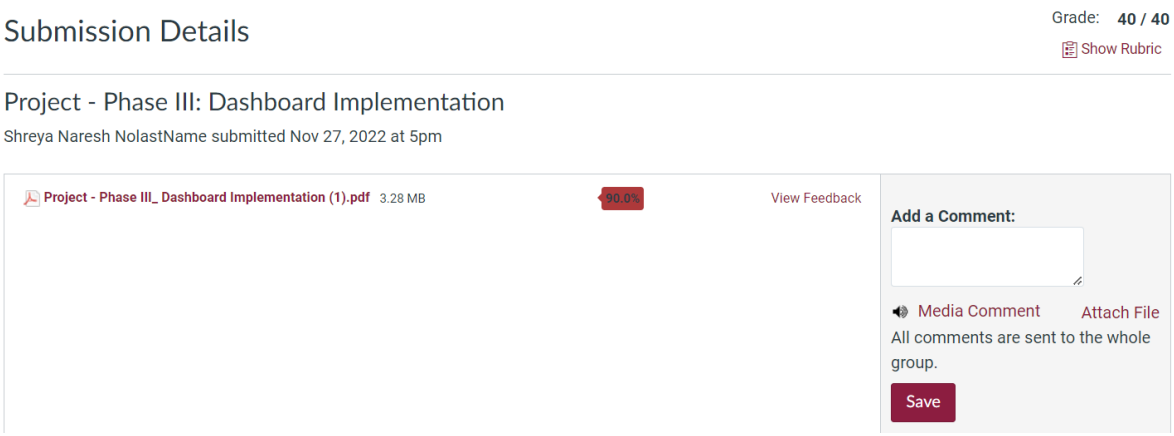
Published Dashboard Link:

https://public.tableau.com/app/profile/rajkumar5102/viz/IFT598DVProject/FinalDashboard?publish=yes

5. Reflection about the Project

The ODI batting dataset gave us an insight about the batting patterns with respect to each player, their order and various other required analysis. This project in all was a good way to learn about data analysis for me.

6. Evidence of Accomplishment



4.3. Accomplishment 3

Title: Medicare Analysis for Fraud Detection

Subject: Advanced analytics of big data

Professor: Robert Rucker

1. Introduction

In the United States, there is a sizable healthcare business with both private and public organizations. The cost of medical services keeps rising, in part as a result of the aging population growth. Between 2012 and 2014, U.S. health insurance spending increased by 6.7% to reach $3 trillion, while Medicare spending, at nearly $600 billion, accounts for 20% of all human services spending in the country. Reducing fraud is one way to assist cover costs and lower overall payments. The goal of this project is to develop big data solutions that have significant applicability at the intersection of the health care and protection sectors.

2. Questions that have been answered.

1. Identify if a particular case is a potential fraud or not?
2. Identifying the region with most frauds in the Medicare industry?
3. Which is the best model used that gives higher accuracy value based on the comparison of different models tested on a given dataset?
4. Which age group has committed the greatest number of fraudulent cases?
5. Identify the top providers with fraudulent cases.

3. Dataset & Technologies

3.1. Dataset

We have selected the database from three different resources that sums up our problem. This dataset was created using data from administrative claims submitted by Medicare

beneficiaries who are Part D program participants and are available from the CMS Chronic Condition Data Warehouse.

They are taken from:

1) CMS Prescriber Data 2017

2) Payment Data 2017

3) Excluded (LEIE) dataset

3.2. Technologies

Python: We have used pandas to read and check the column number using a shape function, data frame that drops the column that has the most frequently repeating or common values.

Exploratory Data Analysis: EDA is used in our 1st question where we check the beneficiary data and deduce that most of the beneficiaries are from the YEAR - 1919 to 1943.

Feature engineering: An example of using feature engineering is where we see that there is no difference in the distribution of Claim Duration for Potentially Fraud and Non-Fraud Providers. Here, we reach the conclusion that Claim Duration alone might not be useful in segregating the Fraud cases.

4. Data Mining Steps (CRISP-DM)

The CRISP-DM cycle involves six steps:

Business Understanding

The first step is to visualize the data in various forms. These include Data exploration, cleansing and preparation. Then we are building a simple data model to join all datasets. Here, we plan to use feature engineering to choose the effective feature sets for the different fraud patterns. Later we are building a machine learning model to detect the different fraud patterns.

Data Understanding

This step involves understanding the raw data that will be used to solve the problem. This means understanding the data attributes and problems as well as the relationship between data attributes and the problem. That is the one main reason we do data pre-processing to eliminate unwanted fields and null values from the dataset. The dataset looks more imbalanced and so in terms of fraud detection context the dataset is very skewed with 99% no fraudulent cases and less than 1% fraudulent cases.

Data Preparation

After understanding the business and data, the next step is data preparation. In this step, some classification codes are run on the data set using machine learning or data mining library. For data mining activities, each library provides functions that take data from files and require the data set to be saved in a specific format (CSV). In order to use machine learning or a data mining library, we have identified the data for our project and will store it in CSV format.

Modelling

This step involves applying data mining methods, running actual code, building models to solve the data mining task. In this step, we have performed logistic regression for machine learning modelling and Gaussian naive bayes classifier. Standard scaler is used for the main task when performing feature engineering. Random forest classifier and Gradient boosting classifier are two main concepts we use for analysing the data after feature engineering is performed.

Evaluation

After running the code and performing the required tasks, the results are obtained based on the modelling we performed with the data. We check if all types of fraud patterns are detected and we figure out the best resulting model considering all the factors used for ML and analysis.

Deployment

This is the concluding stage for the data analysis and the machine learning process we have performed to be brought into execution. To automate the decision-making process, more data

is gathered and we use the data mining techniques to make better visualizations and perform better analysis. At the end, we see that a random forest classifier gives an accuracy rate of 72%.

5. Model Evaluation & Process

1. Identify if a particular case is a potential fraud or not?

CODE & RESULT:

```
In [75]: tmp = pd.DataFrame(train_iobp_df.groupby(['Provider','PotentialFraud'])['BeneID'].count()).reset_index()
         tmp.columns = ['Provider', 'Fraud?', 'Num_of_cases']
         print(tmp.head())
         tot_fraud_cases = tmp[tmp['Fraud?'] == 'Yes']['Num_of_cases'].sum()
         tot_non_fraud_cases = tmp[tmp['Fraud?'] == 'No']['Num_of_cases'].sum()
         tmp['Cases'] = tmp['Fraud?'].apply(lambda val: tot_non_fraud_cases if val == "No" else tot_fraud_cases)
         tmp['Percentage'] = round(((tmp['Num_of_cases'] / tmp['Cases']) * 100),2)
         tmp.head()
```

```
   Provider Fraud?  Num_of_cases
0  PRV51001     No            25
1  PRV51003    Yes           132
2  PRV51004     No           149
3  PRV51005    Yes          1165
4  PRV51007     No            72
```

Out[75]:

	Provider	Fraud?	Num_of_cases	Cases	Percentage
0	PRV51001	No	25	345415	0.01
1	PRV51003	Yes	132	212796	0.06
2	PRV51004	No	149	345415	0.04
3	PRV51005	Yes	1165	212796	0.55
4	PRV51007	No	72	345415	0.02

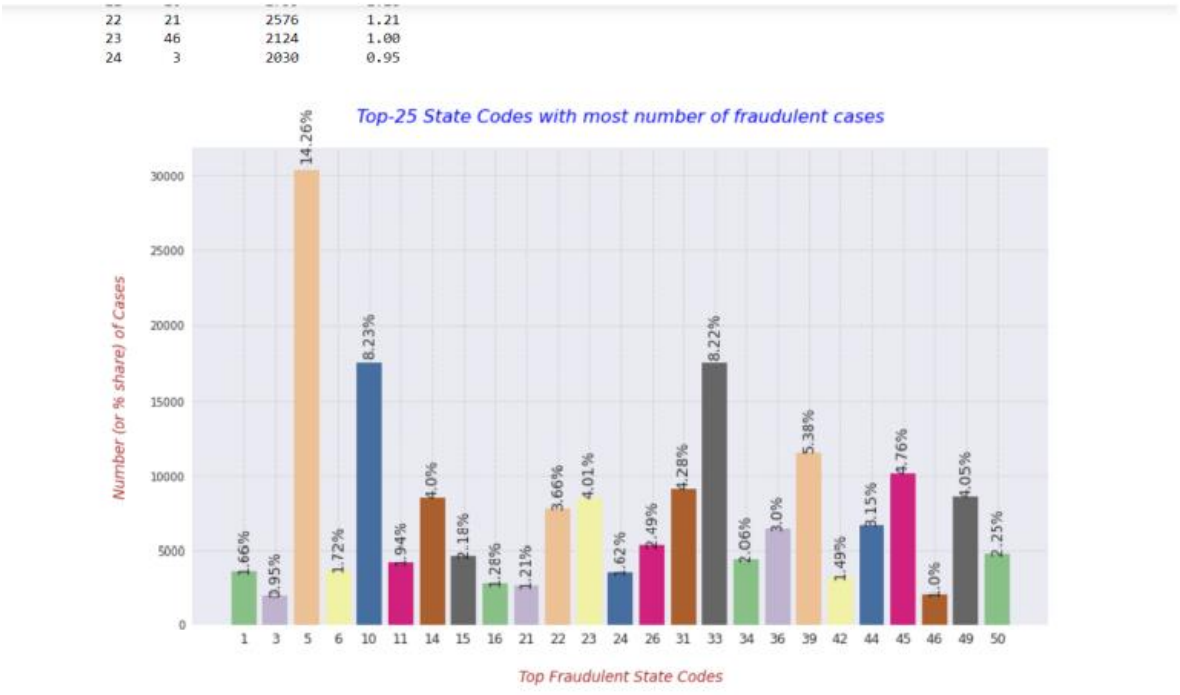
2. Identifying the region with most frauds in the medicare industry?

CODE:

```
# Providing the labels and title to the graph
plt.xlabel("\nTop Fraudulent State Codes", fontdict=label_font_dict)
plt.xticks(rotation=0, fontsize=12)
plt.ylabel("Number (or % share) of Cases\n", fontdict=label_font_dict)
plt.minorticks_on()
plt.grid(which='major', linestyle="--", color='lightgrey')
plt.title("Top-25 State Codes with most number of fraudulent cases\n", fontdict=title_font_dict)
plt.plot();
```

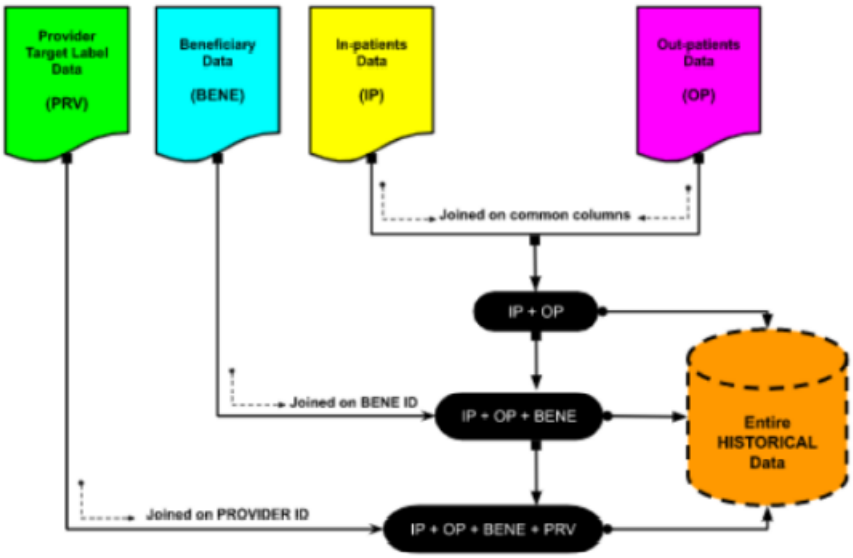
	State	Num_of_cases	Percentage
0	5	30335	14.26
1	10	17512	8.23
2	33	17492	8.22
3	39	11448	5.38
4	45	10135	4.76
5	31	9112	4.28
6	49	8613	4.05
7	23	8538	4.01
8	14	8509	4.00
9	22	7798	3.66
10	44	6709	3.15
11	36	6381	3.00
12	26	5301	2.49
13	50	4782	2.25
14	15	4635	2.18
15	34	4385	2.06
16	11	4123	1.94
17	6	3666	1.72
18	1	3525	1.66
19	24	3453	1.62
20	42	3180	1.49
21	16	2733	1.28
22	21	2576	1.21
23	46	2124	1.00
24	3	2030	0.95

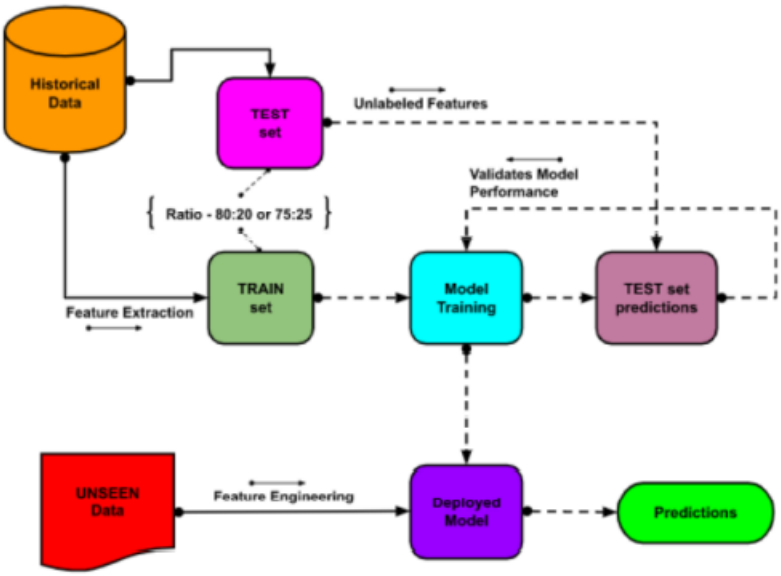
RESULT:



3. Which is the best model used that gives higher accuracy value based on the comparison of different models tested on a given dataset?

Below is the process that we followed:





RESULTS:

jupyter AdvProjFinal Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help — Notebook saved — Trusted — Python 3 (ipykernel)

Run Code

In [126]:
```
from sklearn.model_selection import train_test_split as tts
X_train, X_test, y_train, y_test = tts(X, y, test_size=0.20, stratify=y, random_state=39)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Out[126]:
```
((4328, 30), (1082, 30), (4328,), (1082,))
```

In [127]:
```
from sklearn.preprocessing import RobustScaler
# Standardize the data (train and test)
robust_scaler = RobustScaler()
robust_scaler.fit(X_train)
X_train_std = robust_scaler.transform(X_train)
X_test_std = robust_scaler.transform(X_test)
```

In [128]:
```
from sklearn.linear_model import LogisticRegressionCV
from sklearn import metrics
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, confusion_matrix, roc_curve,
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.calibration import CalibratedClassifierCV
```

In [129]:
```
# Training the model with all features and hyper-parameterized values
log_reg_1 = LogisticRegression(C=0.0316228, penalty='l1',
                               fit_intercept=True, solver='liblinear', tol=0.0001, max_iter=500,
                               class_weight='balanced',
                               verbose=0,
                               intercept_scaling=1.0,
                               multi_class='auto',
                               random_state=49)

log_reg_1.fit(X_train_std, y_train)
```

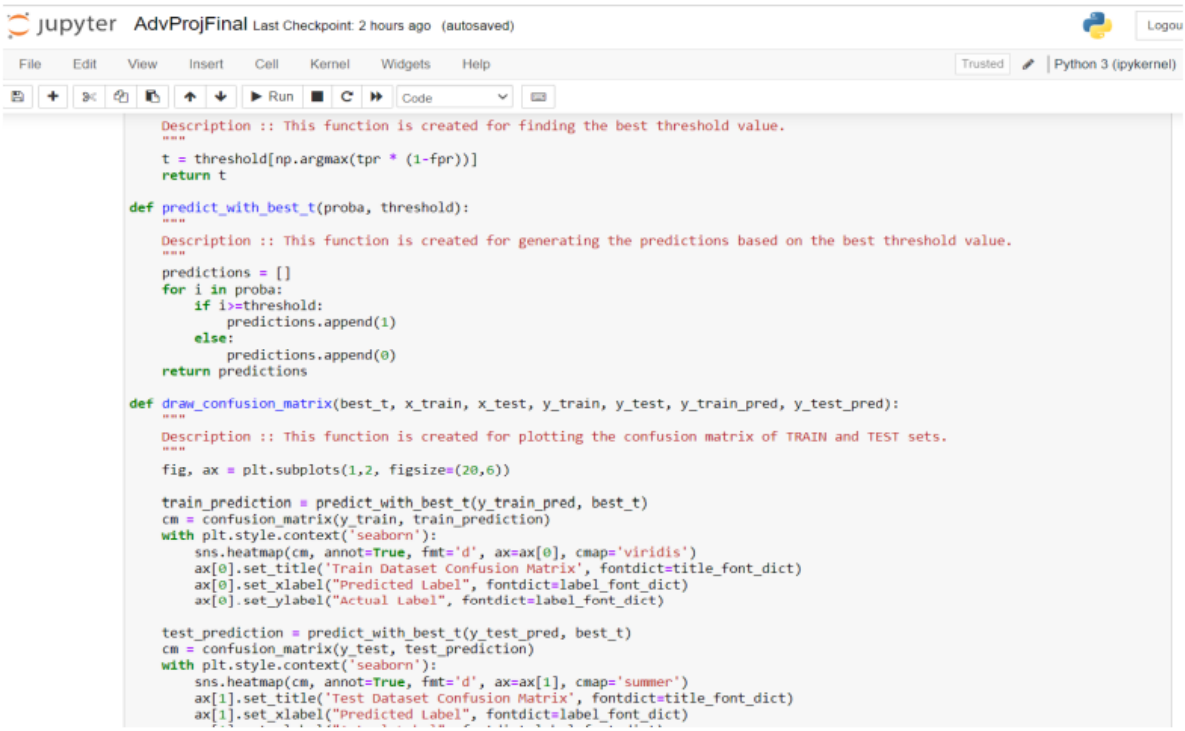
Out[129]:
```
LogisticRegression(C=0.0316228, class_weight='balanced', intercept_scaling=1.0,
                   max_iter=500, penalty='l1', random_state=49,
                   solver='liblinear')
```

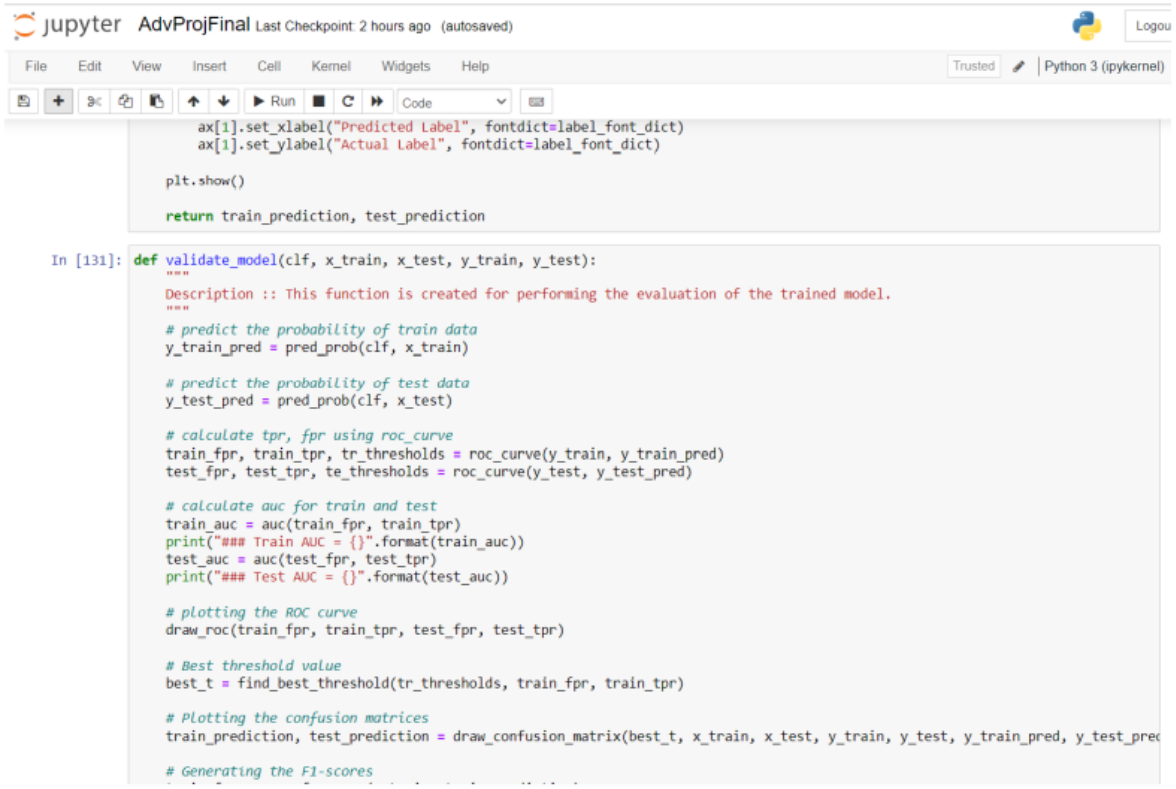
jupyter AdvProjFinal Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help — Trusted — Python 3 (ipykernel)

Run Code

In [130]:
```
def pred_prob(clf, data):
    """
    Description :: This function is created for storing the predicted probabability using the trained model.

    Input :: It accepts below input parameters :
      - clf : Trained model classifier
      - data : Dataset for which we want to generate the predictions
    """
    y_pred = clf.predict_proba(data)[:,1]
    return y_pred

def draw_roc(train_fpr, train_tpr, test_fpr, test_tpr):
    """
    Description :: This function is created for calculating the AUC score on train and test data. And, plotting the ROC curve.

    Input :: It accepts below input parameters :
      - train_fpr : Train False +ve rate
      - train_tpr : Train True +ve rate
      - test_fpr : Test False +ve rate
      - test_tpr : Test True +ve rate
    """
    # calculate auc for train and test
    train_auc = auc(train_fpr, train_tpr)
    test_auc = auc(test_fpr, test_tpr)
    with plt.style.context('seaborn-poster'):
      plt.plot(train_fpr, train_tpr, label="Train AUC ="+"{:.4f}".format(train_auc), color='blue')
      plt.plot(test_fpr, test_tpr, label="Test AUC ="+"{:.4f}".format(test_auc), color='red')
      plt.legend()
      plt.xlabel("False Positive Rate(FPR)", fontdict=label_font_dict)
      plt.ylabel("True Positive Rate(TPR)", fontdict=label_font_dict)
      plt.title("Area Under Curve", fontdict=title_font_dict)
      plt.grid(b=True, which='major', color='lightgrey', linestyle='--')
      plt.minorticks_on()
      plt.show()

def find_best_threshold(threshold, fpr, tpr):
```


jupyter AdvProjFinal Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3 (ipykernel)
Run Code
Description :: This function is created for finding the best threshold value.
t = threshold[np.argmax(tpr * (1-fpr))]
return t
def predict_with_best_t(proba, threshold):
Description :: This function is created for generating the predictions based on the best threshold value.
predictions = []
for i in proba:
if i>=threshold:
predictions.append(1)
else:
predictions.append(0)
return predictions
def draw_confusion_matrix(best_t, x_train, x_test, y_train, y_test, y_train_pred, y_test_pred):
Description :: This function is created for plotting the confusion matrix of TRAIN and TEST sets.
fig, ax = plt.subplots(1,2, figsize=(20,6))
train_prediction = predict_with_best_t(y_train_pred, best_t)
cm = confusion_matrix(y_train, train_prediction)
with plt.style.context('seaborn'):
sns.heatmap(cm, annot=True, fmt='d', ax=ax[0], cmap='viridis')
ax[0].set_title('Train Dataset Confusion Matrix', fontdict=title_font_dict)
ax[0].set_xlabel("Predicted Label", fontdict=label_font_dict)
ax[0].set_ylabel("Actual Label", fontdict=label_font_dict)
test_prediction = predict_with_best_t(y_test_pred, best_t)
cm = confusion_matrix(y_test, test_prediction)
with plt.style.context('seaborn'):
sns.heatmap(cm, annot=True, fmt='d', ax=ax[1], cmap='summer')
ax[1].set_title('Test Dataset Confusion Matrix', fontdict=title_font_dict)
ax[1].set_xlabel("Predicted Label", fontdict=label_font_dict)


jupyter AdvProjFinal Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3 (ipykernel)
Run Code
ax[1].set_xlabel("Predicted Label", fontdict=label_font_dict)
ax[1].set_ylabel("Actual Label", fontdict=label_font_dict)
plt.show()
return train_prediction, test_prediction
In [131]:
def validate_model(clf, x_train, x_test, y_train, y_test):
Description :: This function is created for performing the evaluation of the trained model.
# predict the probability of train data
y_train_pred = pred_prob(clf, x_train)
# predict the probability of test data
y_test_pred = pred_prob(clf, x_test)
# calculate tpr, fpr using roc_curve
train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
# calculate auc for train and test
train_auc = auc(train_fpr, train_tpr)
print("### Train AUC = {}".format(train_auc))
test_auc = auc(test_fpr, test_tpr)
print("### Test AUC = {}".format(test_auc))
# plotting the ROC curve
draw_roc(train_fpr, train_tpr, test_fpr, test_tpr)
# Best threshold value
best_t = find_best_threshold(tr_thresholds, train_fpr, train_tpr)
# Plotting the confusion matrices
train_prediction, test_prediction = draw_confusion_matrix(best_t, x_train, x_test, y_train, y_test, y_train_pred, y_test_pred
Generating the F1-scores

jupyter AdvProjFinal Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

```
    train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
    test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)

    # calculate auc for train and test
    train_auc = auc(train_fpr, train_tpr)
    print("### Train AUC = {}".format(train_auc))
    test_auc = auc(test_fpr, test_tpr)
    print("### Test AUC = {}".format(test_auc))

    # plotting the ROC curve
    draw_roc(train_fpr, train_tpr, test_fpr, test_tpr)

    # Best threshold value
    best_t = find_best_threshold(tr_thresholds, train_fpr, train_tpr)

    # Plotting the confusion matrices
    train_prediction, test_prediction = draw_confusion_matrix(best_t, x_train, x_test, y_train, y_test, y_train_pred, y_test_pred

    # Generating the F1-scores
    train_f1_score = f1_score(y_train, train_prediction)
    test_f1_score = f1_score(y_test, test_prediction)

    return test_auc, train_f1_score, test_f1_score, best_t
```

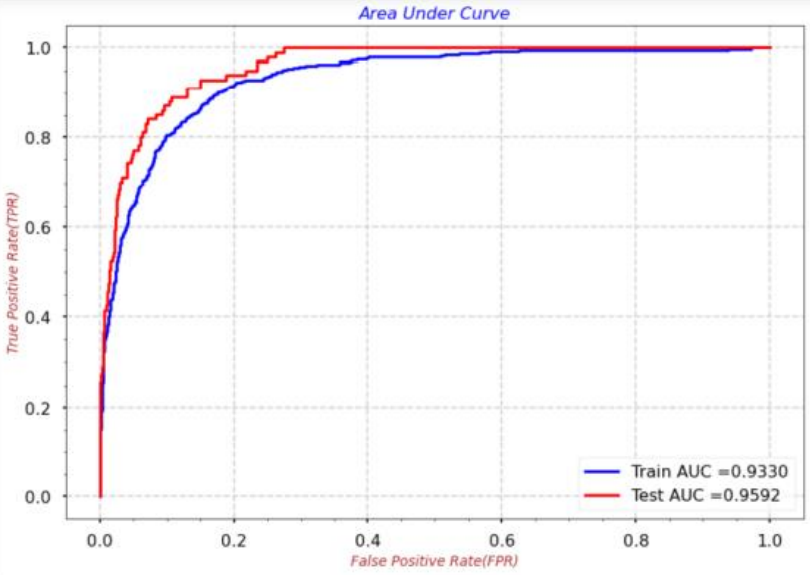
In [138]:

```
# Validate Logistic Regression model
test_auc, train_f1_score, test_f1_score, best_t = validate_model(log_reg_1, X_train_std, X_test_std, y_train, y_test)

print("\n")
print("### Best Threshold = {:.4f}".format(best_t))
print("### Model AUC is : {:.4f}".format(test_auc))
print("### Model Train F1 Score is : {:.4f}".format(train_f1_score))
print("### Model Test F1 Score is : {:.4f}".format(test_f1_score))
```

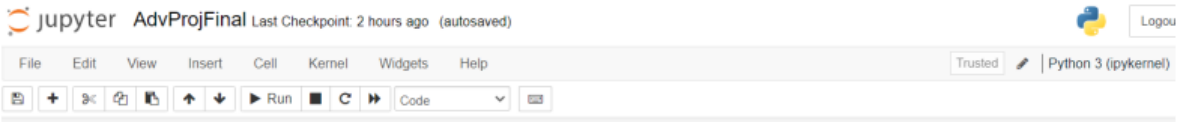
```
### Train AUC = 0.9330161157844056
### Test AUC = 0.9591950020690143
```

jupyter AdvProjFinal Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help




```
### Best Threshold = 0.3468
### Model AUC is : 0.9592
### Model Train F1 Score is : 0.5121
### Model Test F1 Score is : 0.5481
```



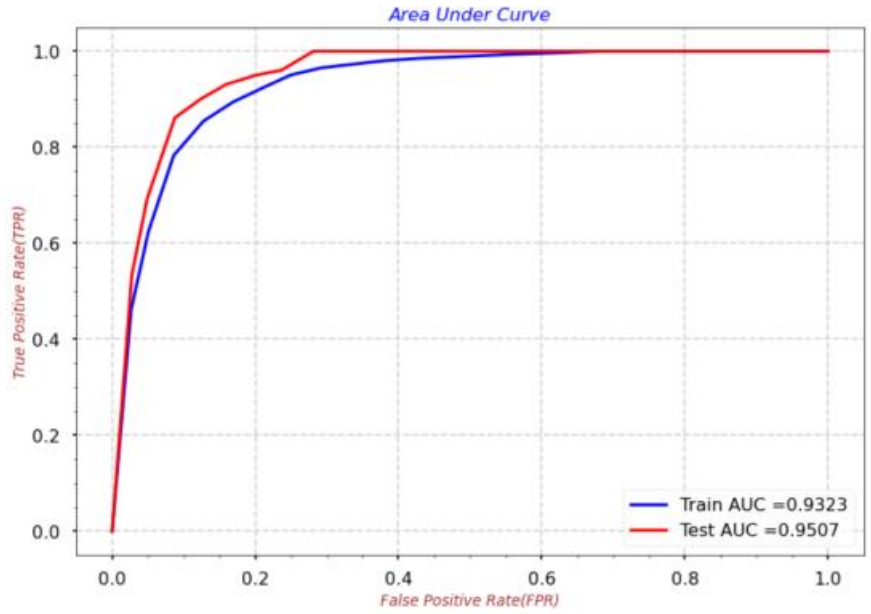
In [144]:
```
from sklearn.tree import DecisionTreeClassifier
```

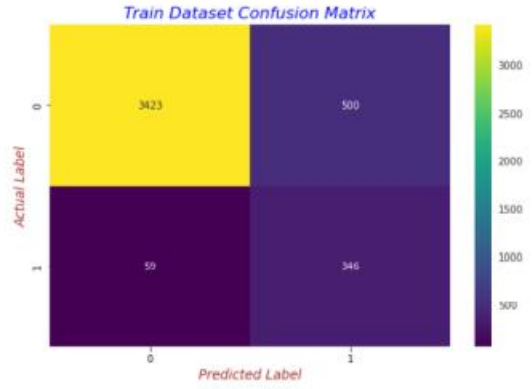
In [145]:
```
dec_tree_2 = DecisionTreeClassifier(criterion='gini',
                                    max_depth= 6,
                                    max_features='log2',
                                    min_samples_leaf=150,
                                    min_samples_split=150,
                                    class_weight='balanced',
                                    random_state=49,
                                    splitter='best',
                                    min_weight_fraction_leaf=0.0,
                                    max_leaf_nodes=None,
                                    min_impurity_decrease=0.0,
                                    ccp_alpha=0.0,)

dec_tree_2.fit(X_train_std, y_train)
```

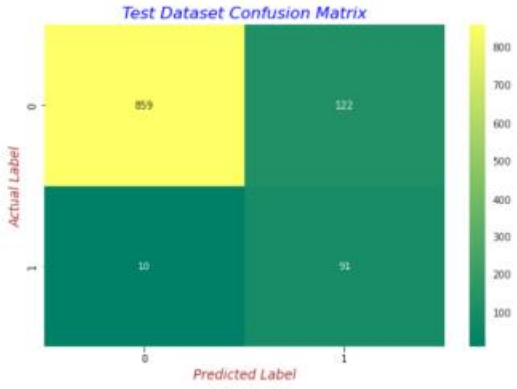
Out[145]:
```
DecisionTreeClassifier(class_weight='balanced', max_depth=6,
                       max_features='log2', min_samples_leaf=150,
                       min_samples_split=150, random_state=49)
```

In [146]:
```
test_auc, train_f1_score, test_f1_score, best_t = validate_model(dec_tree_2, X_train_std, X_test_std, y_train, y_test)

print("\n")
print("### Best Threshold = {:.4f}".format(best_t))
print("### Model AUC is : {:.4f}".format(test_auc))
print("### Model Train F1 Score is : {:.4f}".format(train_f1_score))
print("### Model Test F1 Score is : {:.4f}".format(test_f1_score))
```

```
### Train AUC = 0.932310558498
### Test AUC = 0.9506868118004461
```








```
### Best Threshold = 0.6328
### Model AUC is : 0.9507
### Model Train F1 Score is : 0.5532
### Model Test F1 Score is : 0.5796
```

jupyter AdvProjFinal Last Checkpoint: 3 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [147]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [148]:
```
rfc_3 = RandomForestClassifier(n_estimators=30,criterion='gini',
                               max_depth= 4,
                               max_features='auto',
                               min_samples_leaf=50,
                               min_samples_split=50,
                               class_weight='balanced',
                               random_state=49,
                               min_weight_fraction_leaf=0.0,
                               max_leaf_nodes=None,
                               min_impurity_decrease=0.0,
                               ccp_alpha=0.0,)

rfc_3.fit(X_train_std, y_train)
```

Out[148]:
```
RandomForestClassifier(class_weight='balanced', max_depth=4,
                       min_samples_leaf=50, min_samples_split=50,
                       n_estimators=30, random_state=49)
```

In [149]:
```
test_auc, train_f1_score, test_f1_score, best_t = validate_model(rfc_3, X_train_std, X_test_std, y_train, y_test)

print("\n")
print("### Best Threshold = {:.4f}".format(best_t))
print("### Model AUC is : {:.4f}".format(test_auc))
print("### Model Train F1 Score is : {:.4f}".format(train_f1_score))
print("### Model Test F1 Score is : {:.4f}".format(test_f1_score))
```

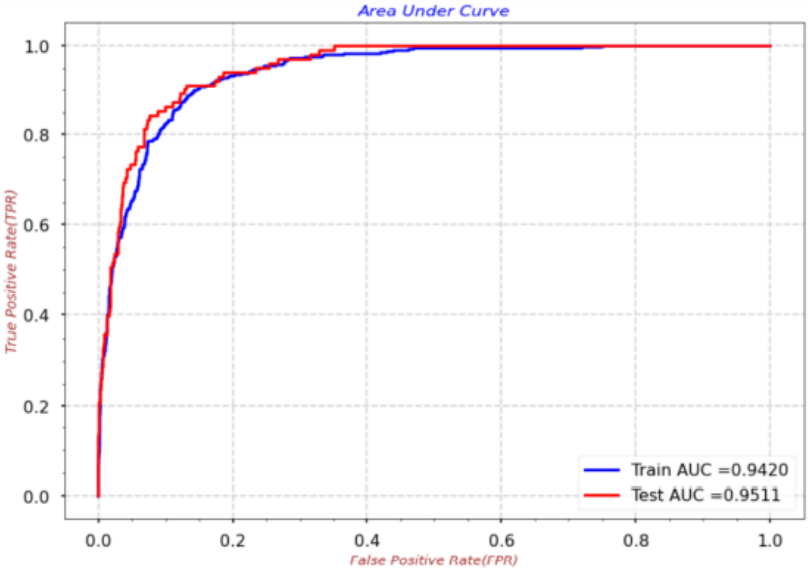
```
### Train AUC = 0.9420366751320952
### Test AUC = 0.9510955682724236
```

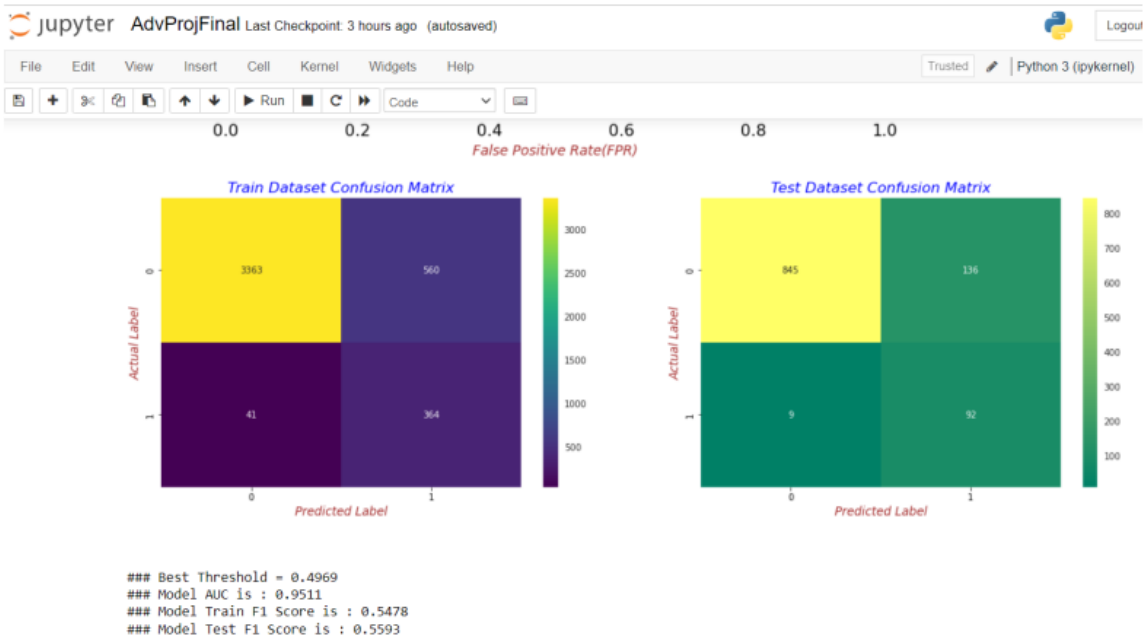
jupyter AdvProjFinal Last Checkpoint: 3 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

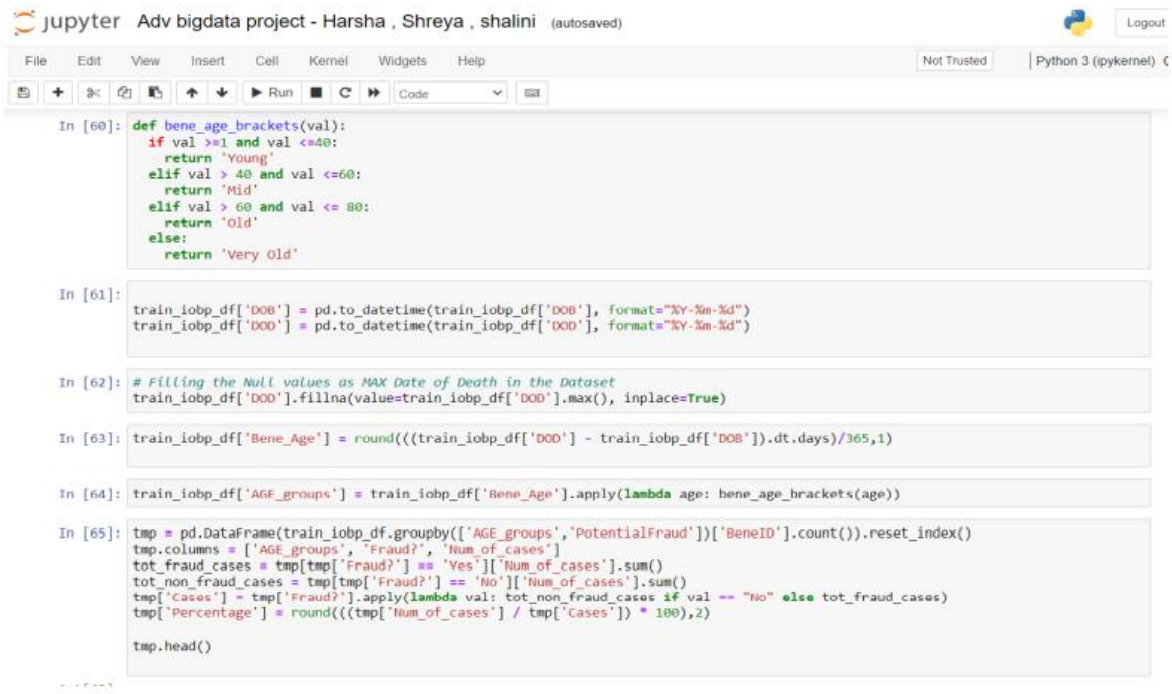
Run Code





4. Which age group has committed the greatest number of fraudulent cases?

RESULTS:



jupyter Adv bigdata project - Harsha , Shreya , shalini (autosaved)

File Edit View Insert Cell Kernel Widgets Help — Not Trusted — Python 3 (ipykernel)

Out[65]:

	AGE_groups	Fraud?	Num_of_cases	Cases	Percentage
0	Mid	No	35524	345415	10.28
1	Mid	Yes	21152	212796	9.94
2	Old	No	190334	345415	55.10
3	Old	Yes	116676	212796	54.83
4	Very Old	No	110885	345415	32.10

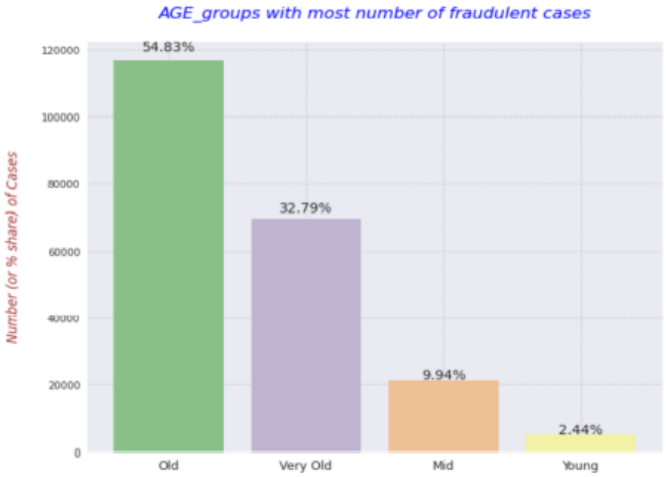
In [66]:
```
tmp_only_frauds = tmp[tmp['Fraud?'] == 'Yes'].sort_values(by=['Percentage'], ascending=False).reset_index(drop=True)
```

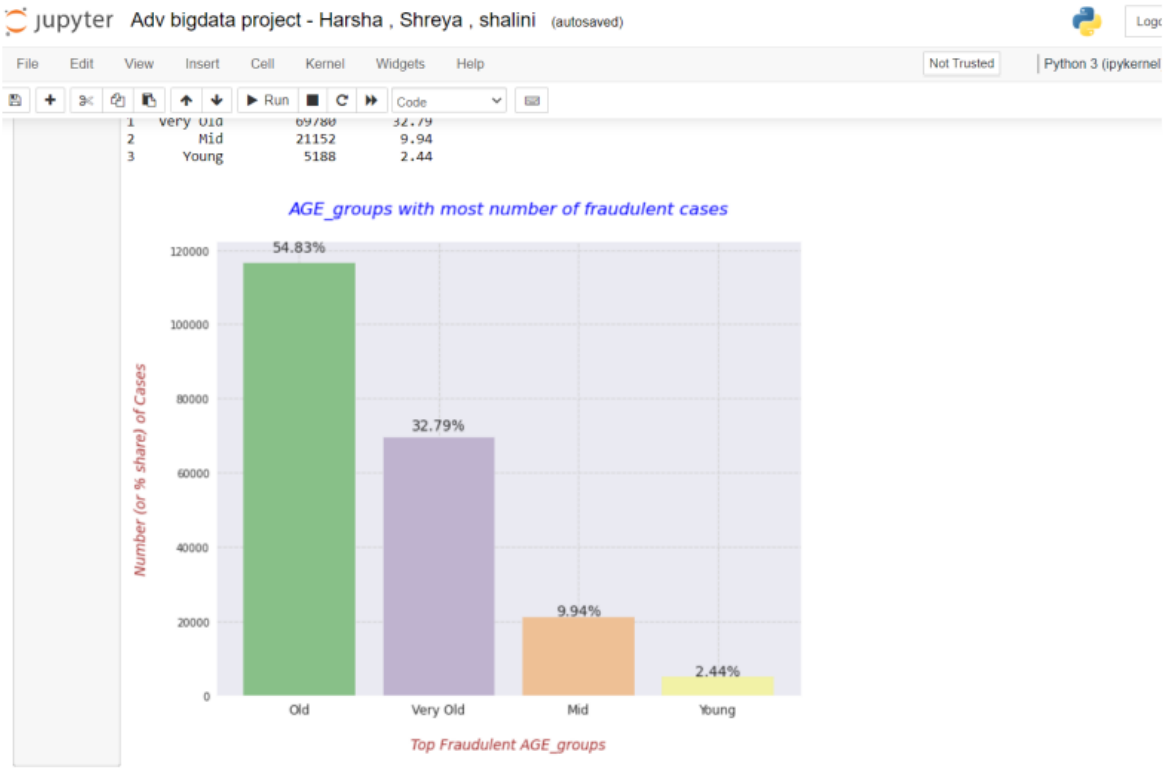
In [67]:
```
print(tmp_only_frauds[['AGE_groups','Num_of_cases','Percentage']].head(25), "\n")

with plt.style.context('seaborn'):
    plt.figure(figsize=(10,8))
    fig = sns.barplot(data=tmp_only_frauds, x="AGE_groups", y="Num_of_cases", palette='Accent')
    # Using the "patches" function we will get the location of the rectangle bars from the graph.
    ## Then by using those location(width & height) values we will add the annotations
    for p in fig.patches:
        width = p.get_width()
        height = p.get_height()
        x, y = p.get_xy()
        fig.annotate(f'{str(round((height*100)/tot_fraud_cases,2))+"%"}', (x + width/2, y + height*1.025), ha='center', fontsize
    # Providing the labels and title to the graph
    plt.xlabel("\nTop Fraudulent AGE_groups", fontdict=label_font_dict)
    plt.xticks(rotation=0, fontsize=12)
    plt.ylabel("Number (or % share) of Cases\n", fontdict=label_font_dict)
    plt.minorticks_on()
    plt.grid(which='major', linestyle="--", color='lightgrey')
    plt.title("AGE_groups with most number of fraudulent cases\n", fontdict=title_font_dict)
    plt.plot();
```

jupyter Adv bigdata project - Harsha , Shreya , shalini (autosaved)

File Edit View Insert Cell Kernel Widgets Help — Not Trusted — Python 3 (ipykernel)

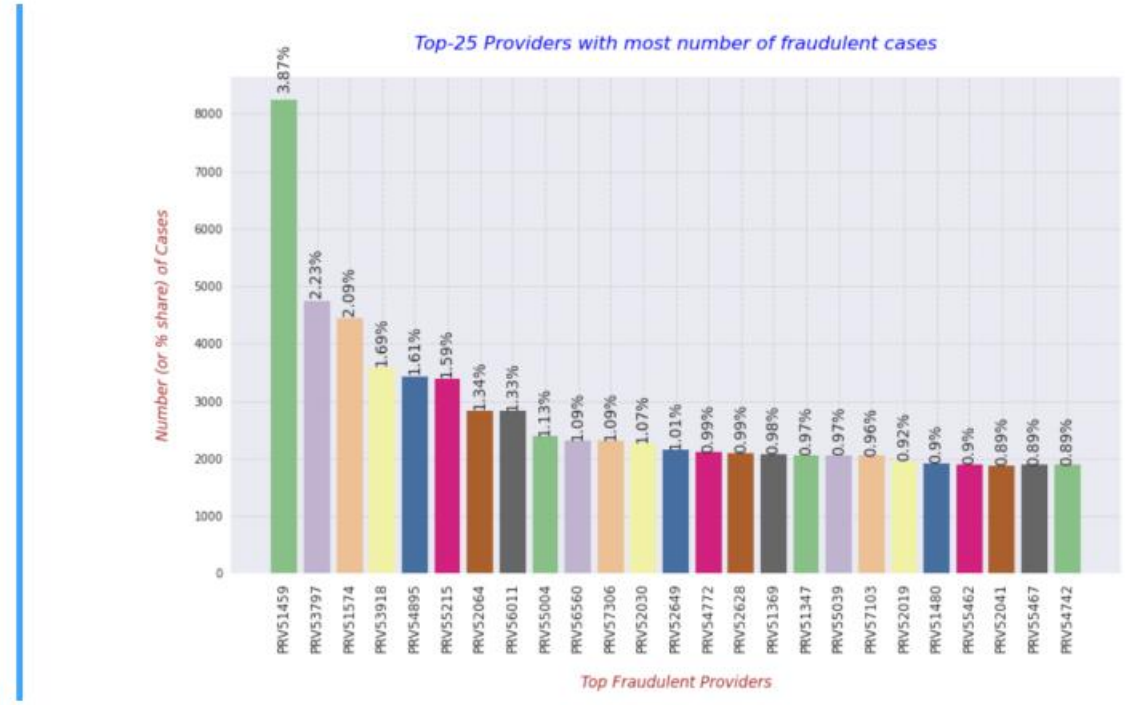
```
  AGE_groups  Num_of_cases  Percentage
0        Old        116676       54.83
1   Very Old         69780       32.79
2        Mid         21152        9.94
3      Young          5188        2.44
```





5. Identify the top providers with fraudulent cases.

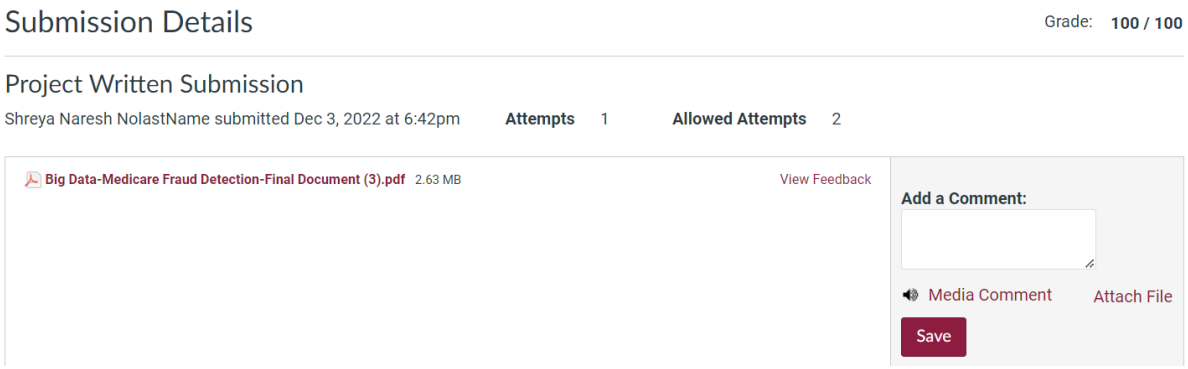
RESULT:



6. Reflection of the Project

Out of the ML Models Performed, the best-resulting model is Random Forest with an AUC of 72 %. I learned the process of using Python for various different data analysis with huge datasets and classify them as per the various machine learning models too. We used random forest classifier, Naive-bayes classifier and K-means algorithm so we could understand the basic classification techniques also. I worked vey closely towards my project as it was a very good project to showcase in my data field as well as a professional.

7. Evidence of Accomplishment



SECTION 5: REFERENCES

Accomplishment 1: IFT 530- Advanced Database management systems

1. https://www.w3schools.com/sql/
2. https://www.w3schools.blog/couchdb-tutorial

Accomplishment 2: IFT 598: Data visualization & Reporting in IT

1. https://www.kaggle.com/datasets/mahendran1/icc-cricket
2. https://app.mural.co/t/asudvworkspace7059/m/asudvworkspace7059/1668114350401/e4738c1807844ca1185a22b884eb302a3d806640?sender=u2dd9184f2a2a660b2ab74201

Accomplishment 3: IFT 512: Advanced analytics of big data

1. Part D Prescriber Data CY 2017. (n.d.). Retrieved June 23, 2020, from https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/PartD2017
2. Scikit Learn - Boosting Methods. (n.d.). Retrieved October 15, 2022, from https://www.tutorialspoint.com/scikit_learn/scikit_learn_boosting_methods.htm
3. Scikit Learn - Boosting Methods. (n.d.). Retrieved October 15, 2022, from https://www.tutorialspoint.com/scikit_learn/scikit_learn_boosting_methods.htm