STW220CT Data and Information Retrieval

2/24/2022

Coursework

ASSESSMENT CYCLE:

Nov 2021

Submitted by:

Parikshit Balami

Student ID: 160484

CU ID: 11790961

Submitted to:

Kiran Rana

Link:https://github.com/parikshitbalami/Data-and-information-retreivel

Table of Contents

Task-1

Database Design

1. I'm keeping this database as traditional because its simple model, it has high data accuracy and data are easily accessible, it has flexibility and high security and can be easily normalized.

2.

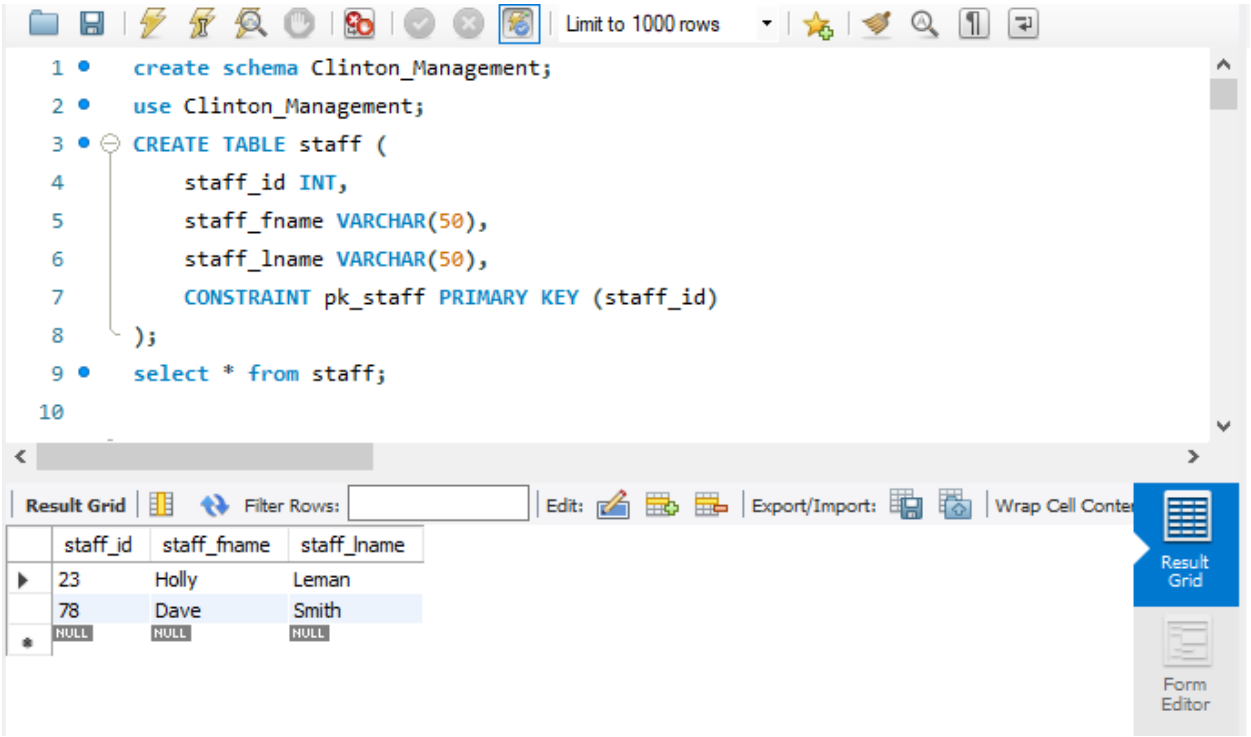


Figure 1: Query

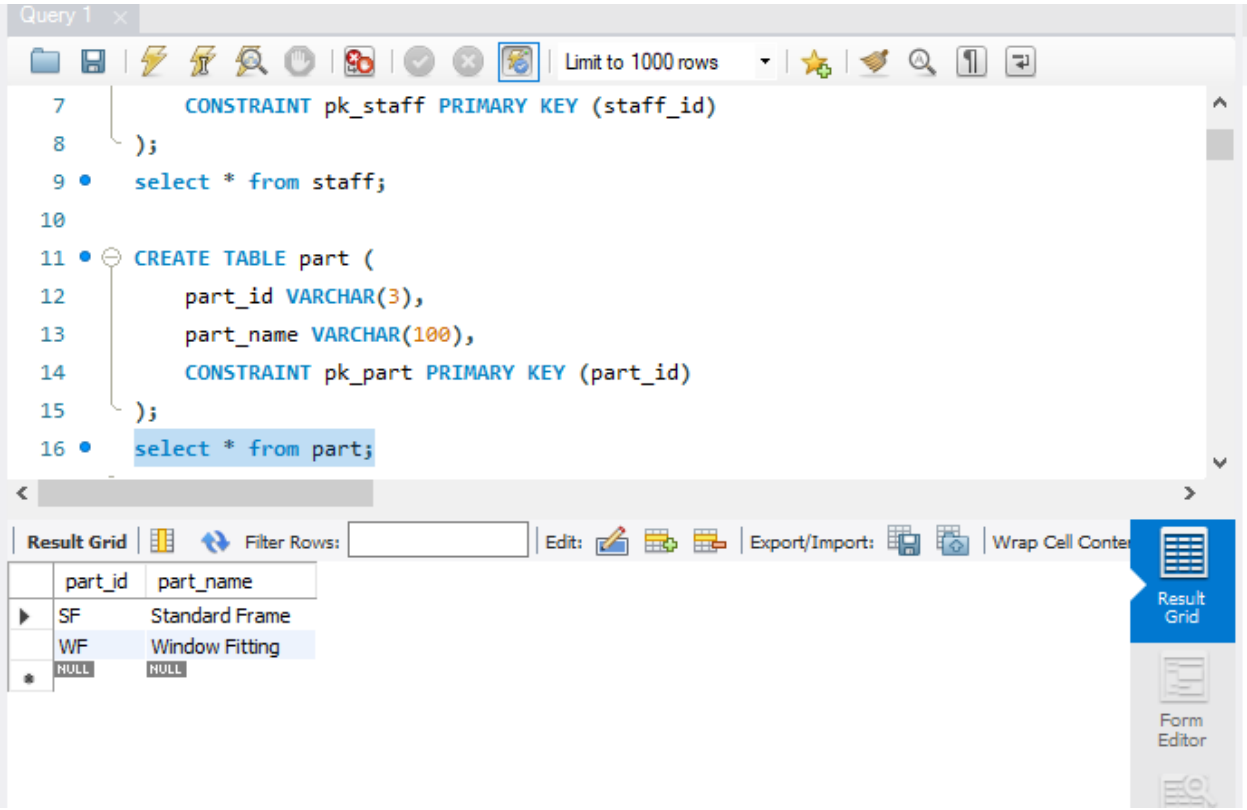


Figure 2: Query (1)

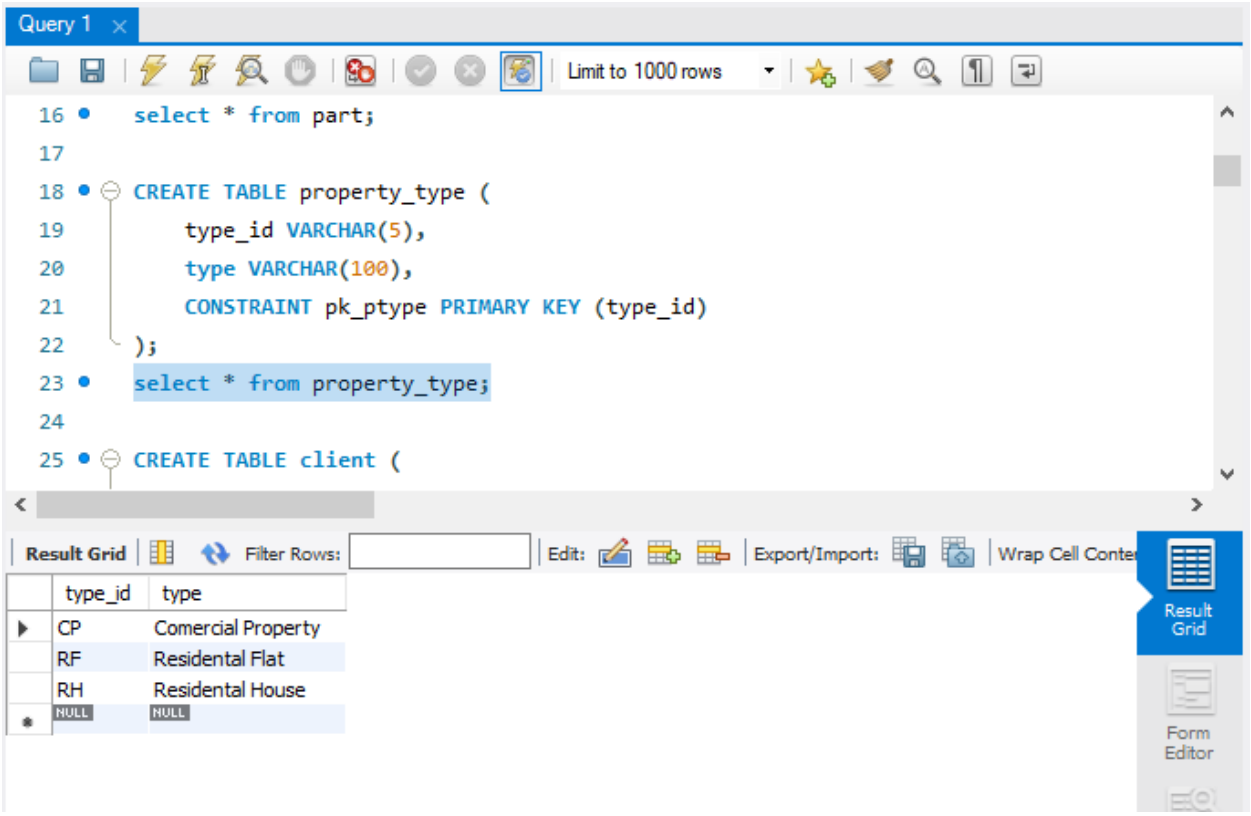


Figure 3: Query (2)

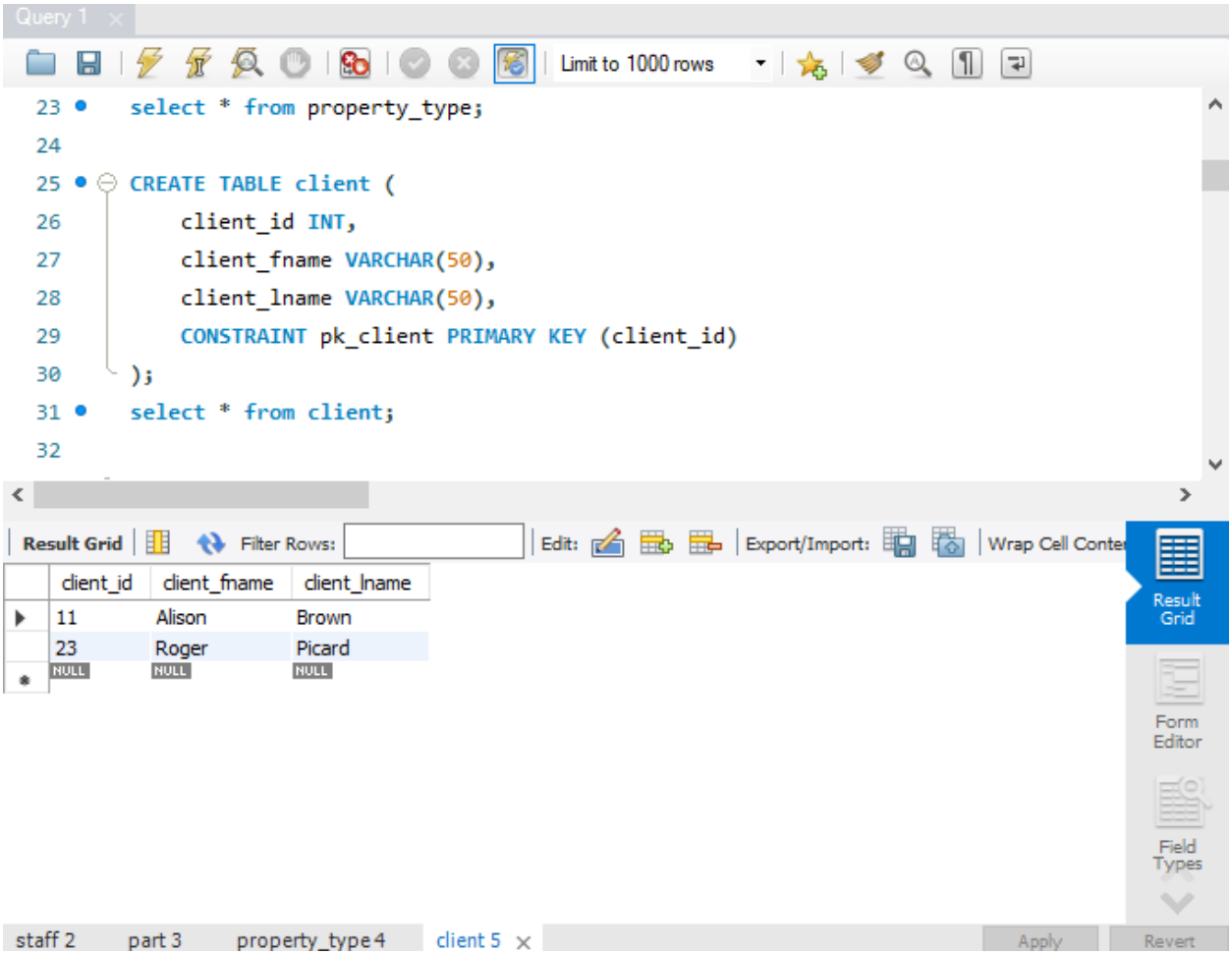


Figure 4: Query (3)

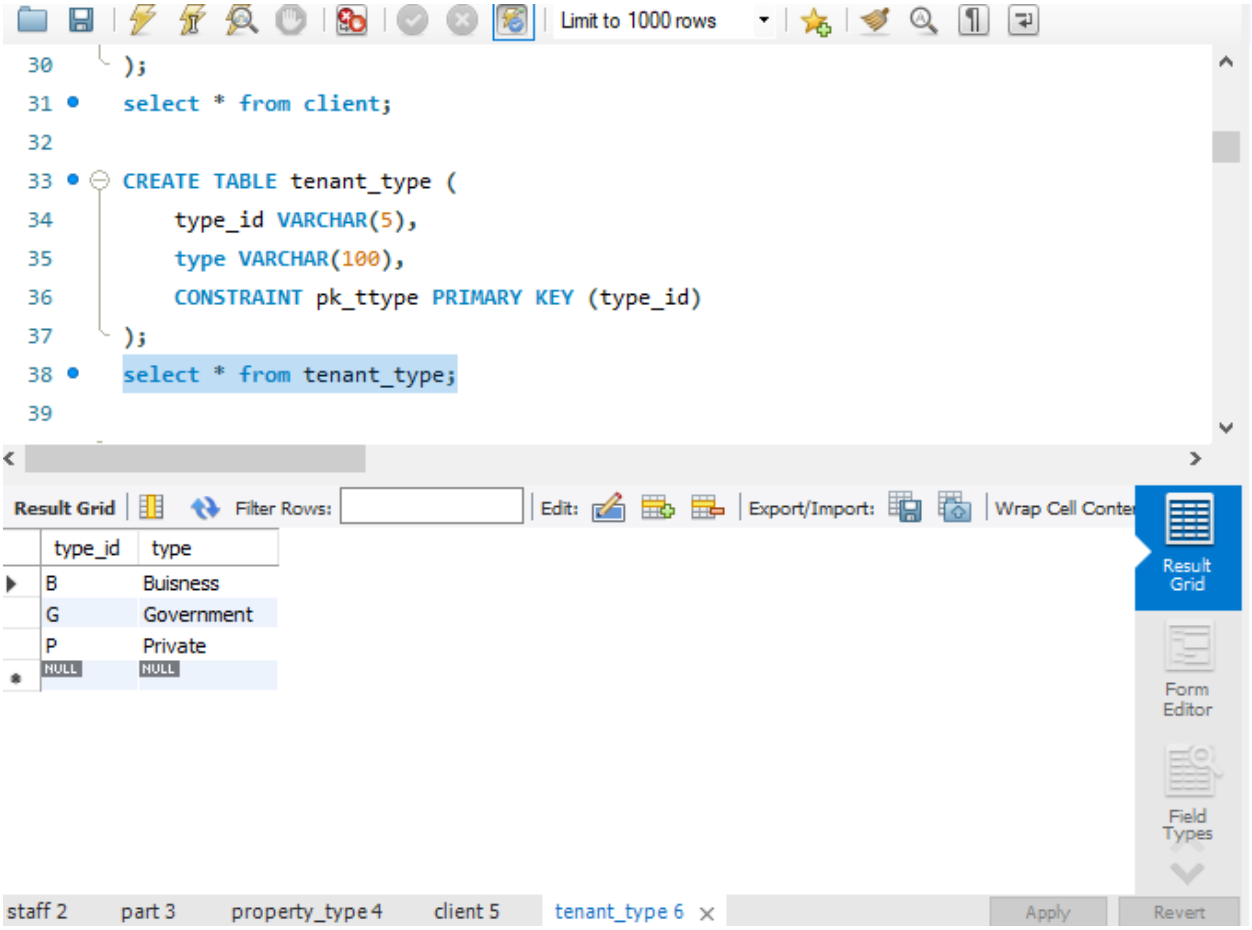


Figure 5: Query (4)

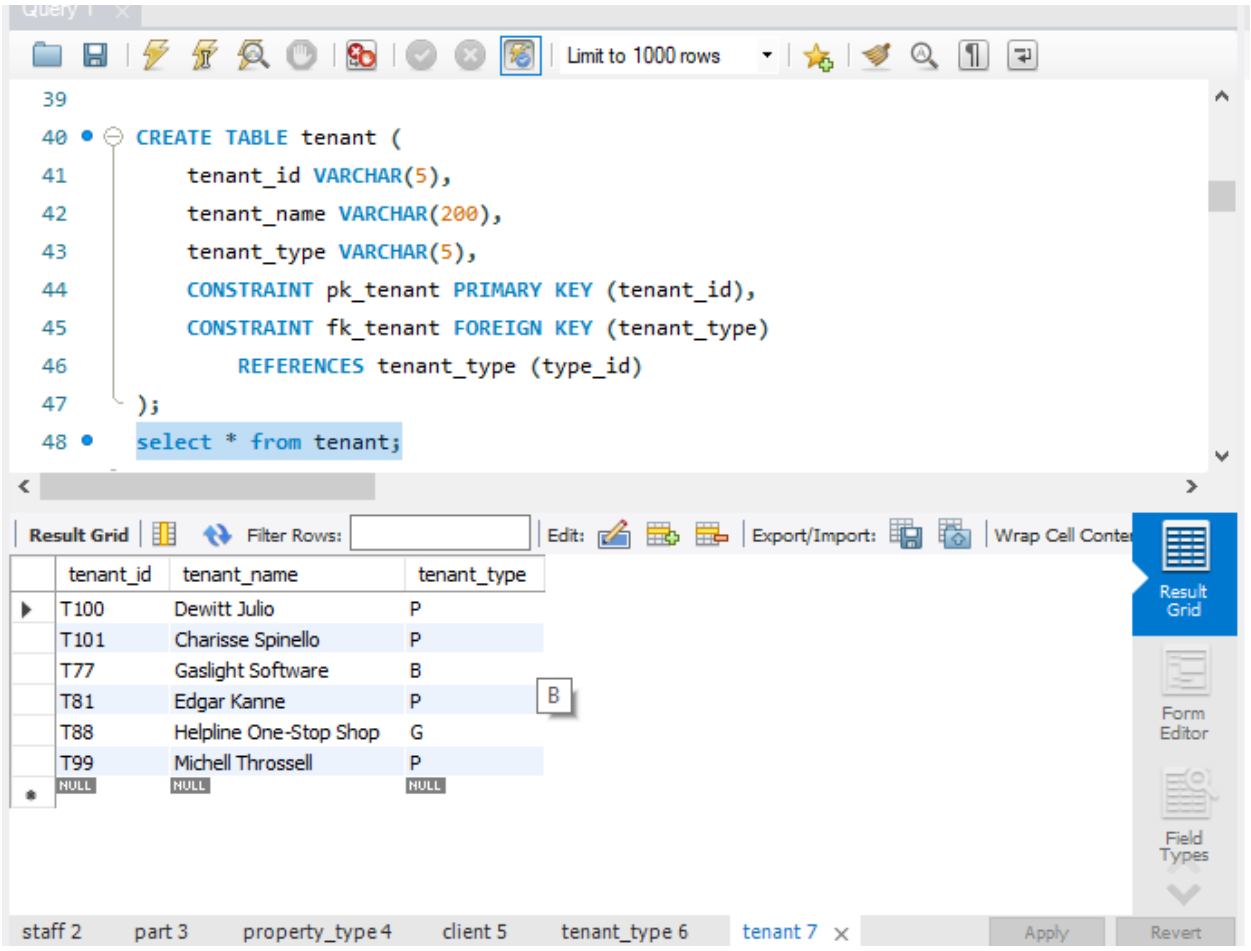


Figure 6:Query (5)

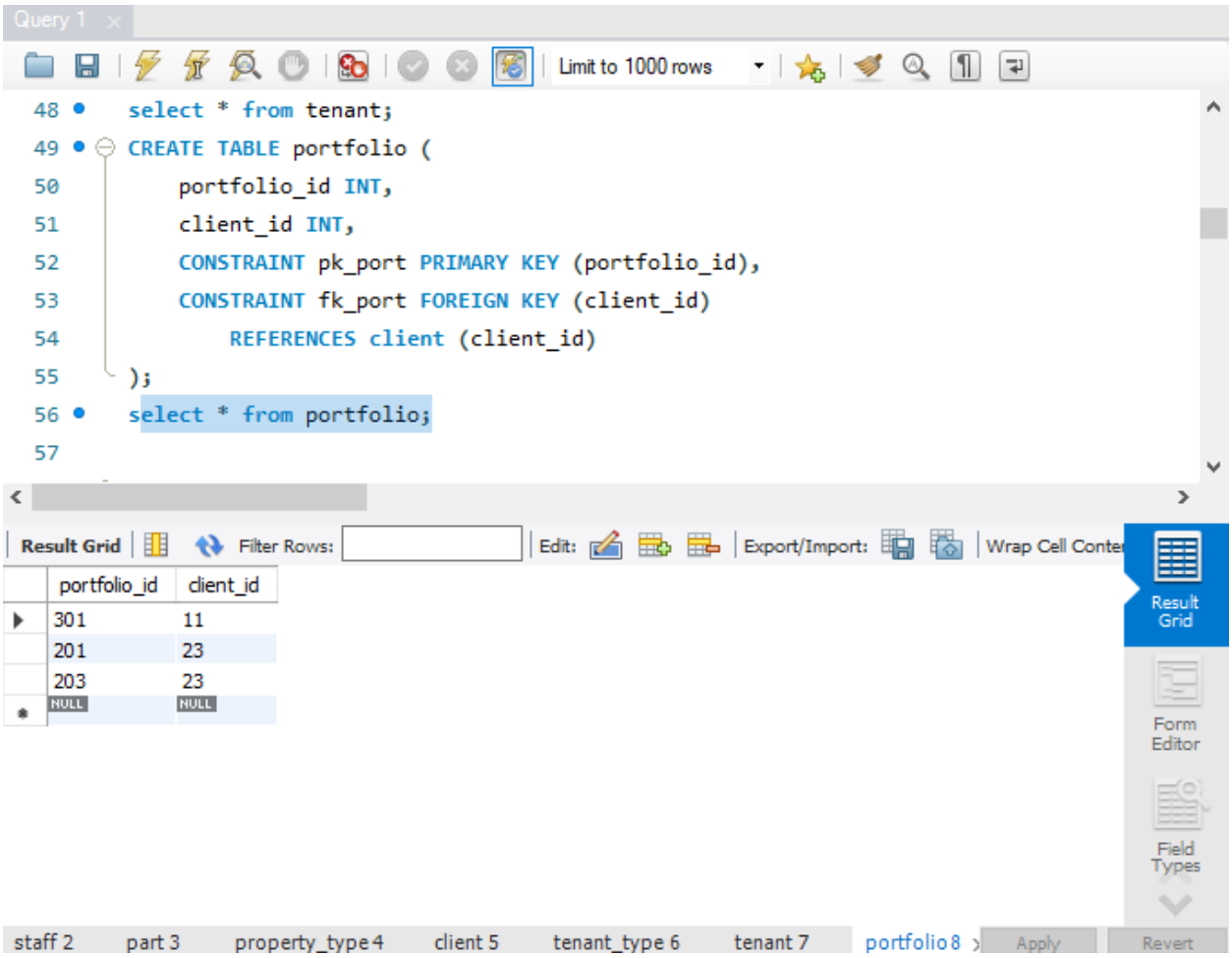


Figure 7: Query (6)

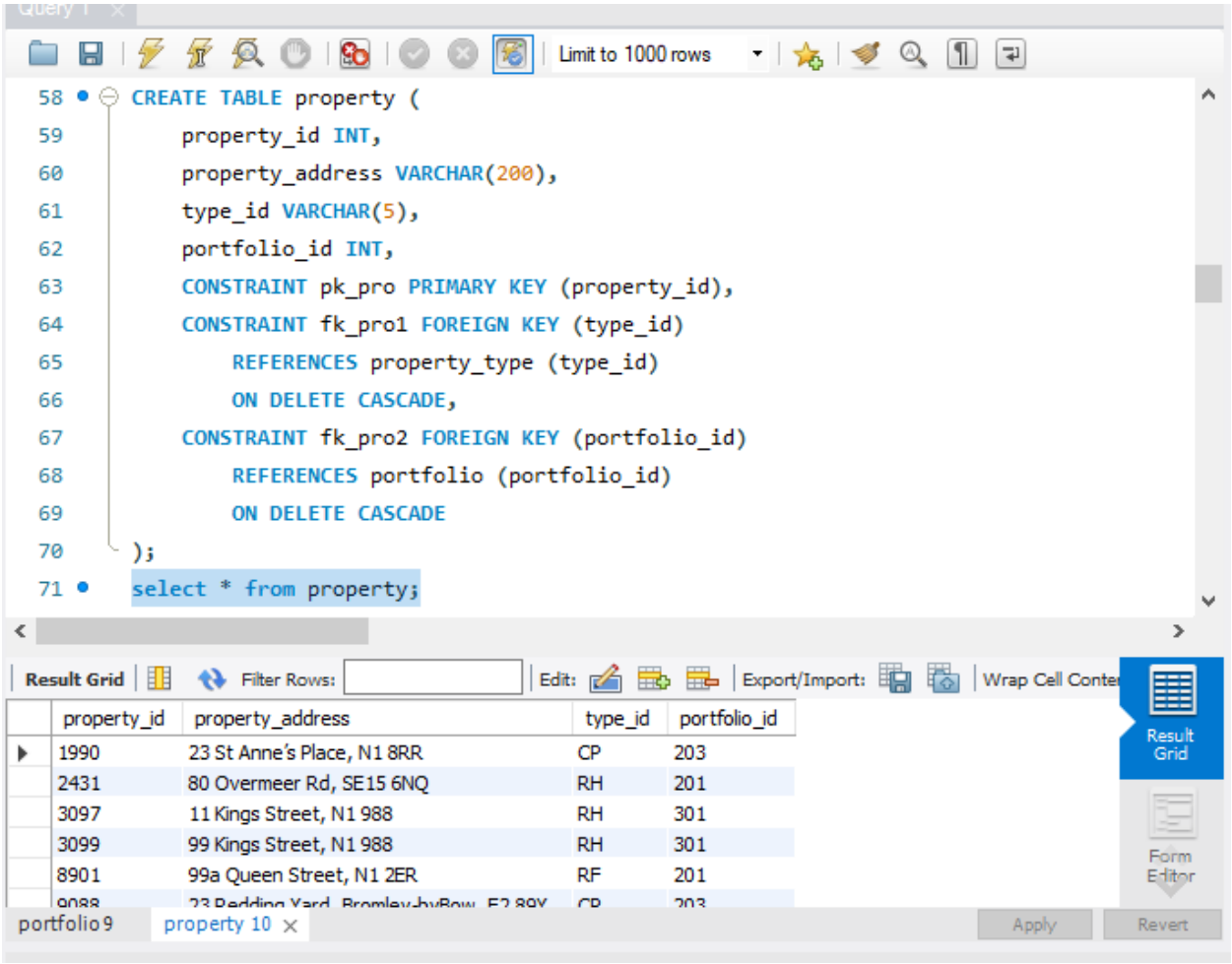


Figure 8: Query (7)

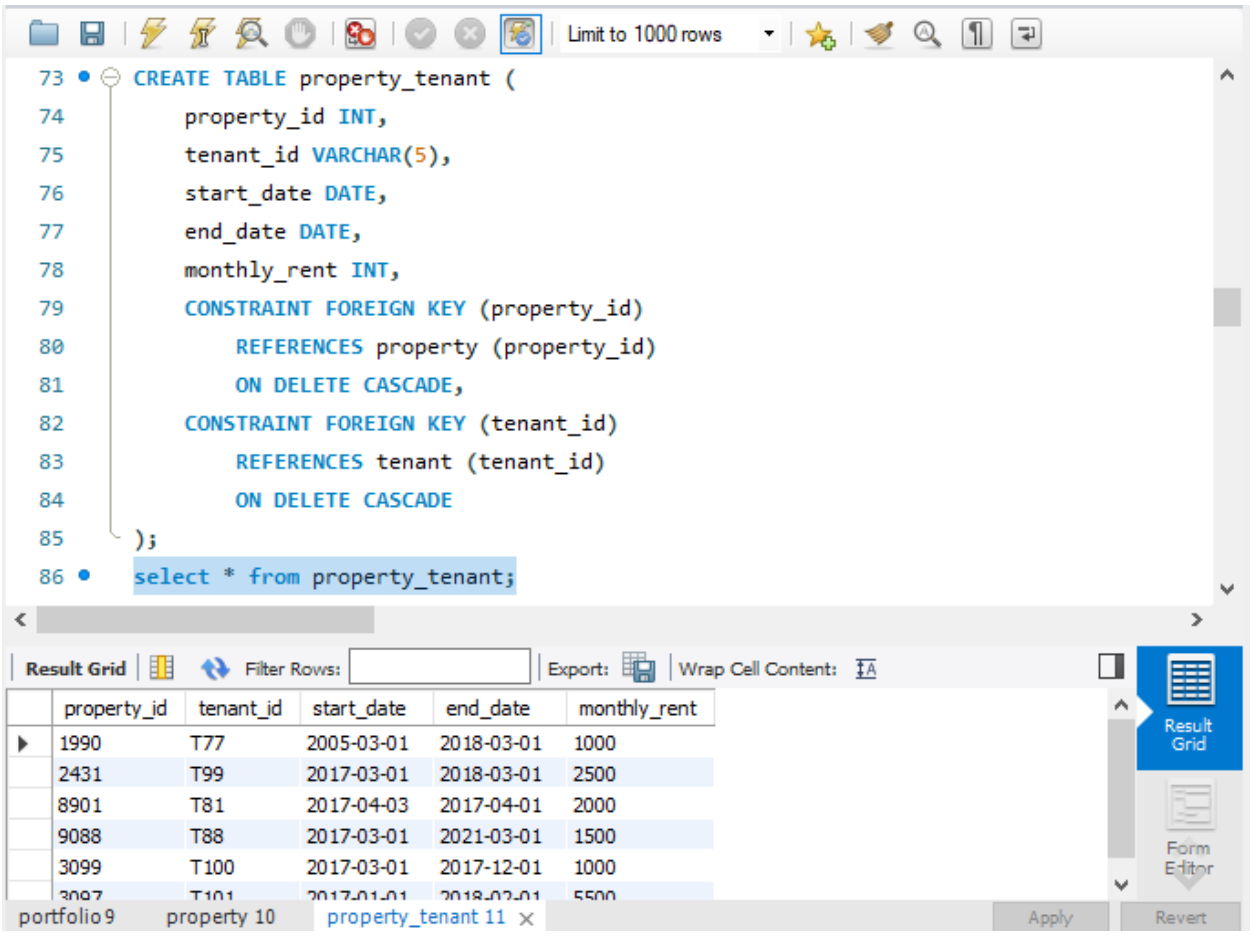


Figure 9: Query (8)

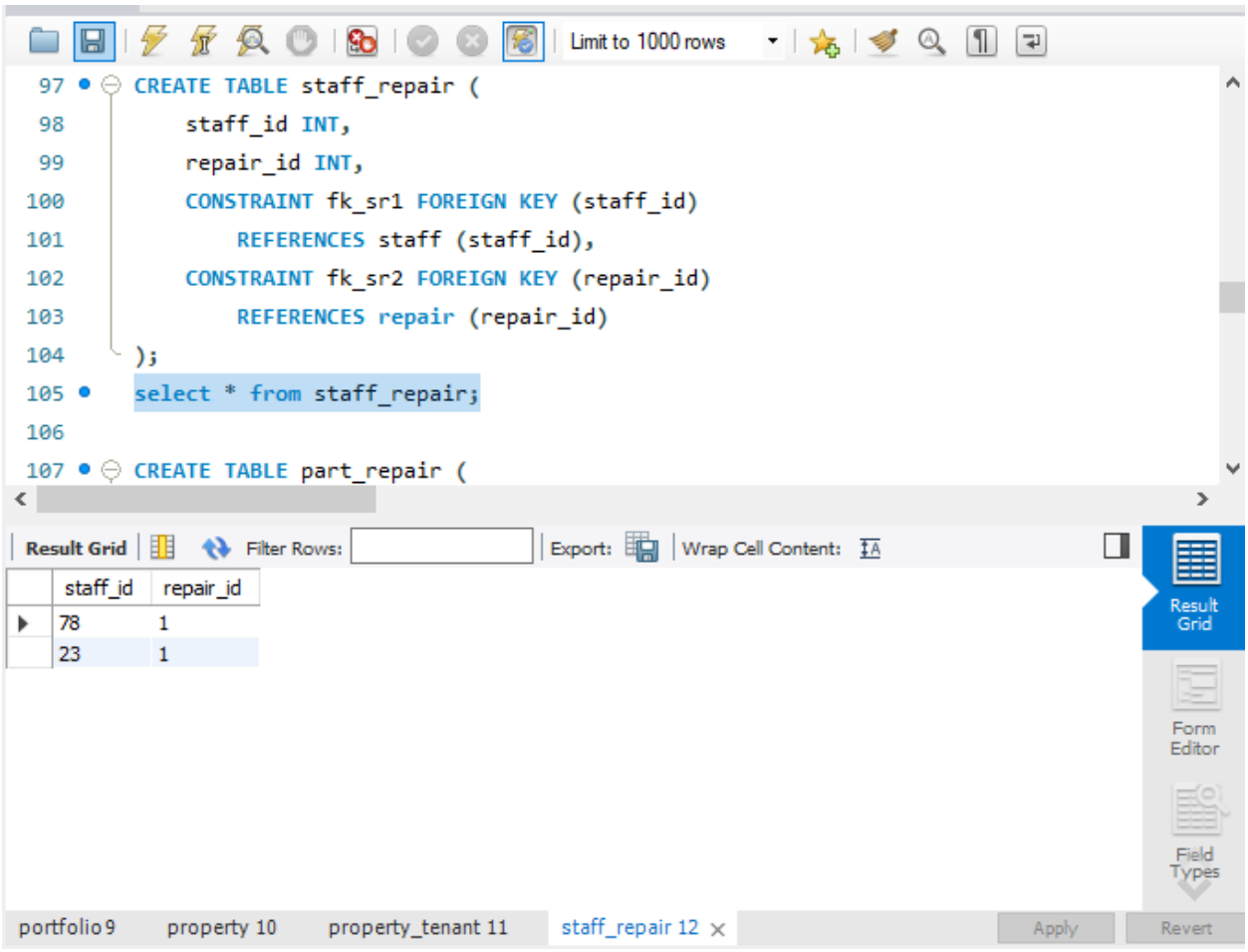


Figure 10: Query (9)

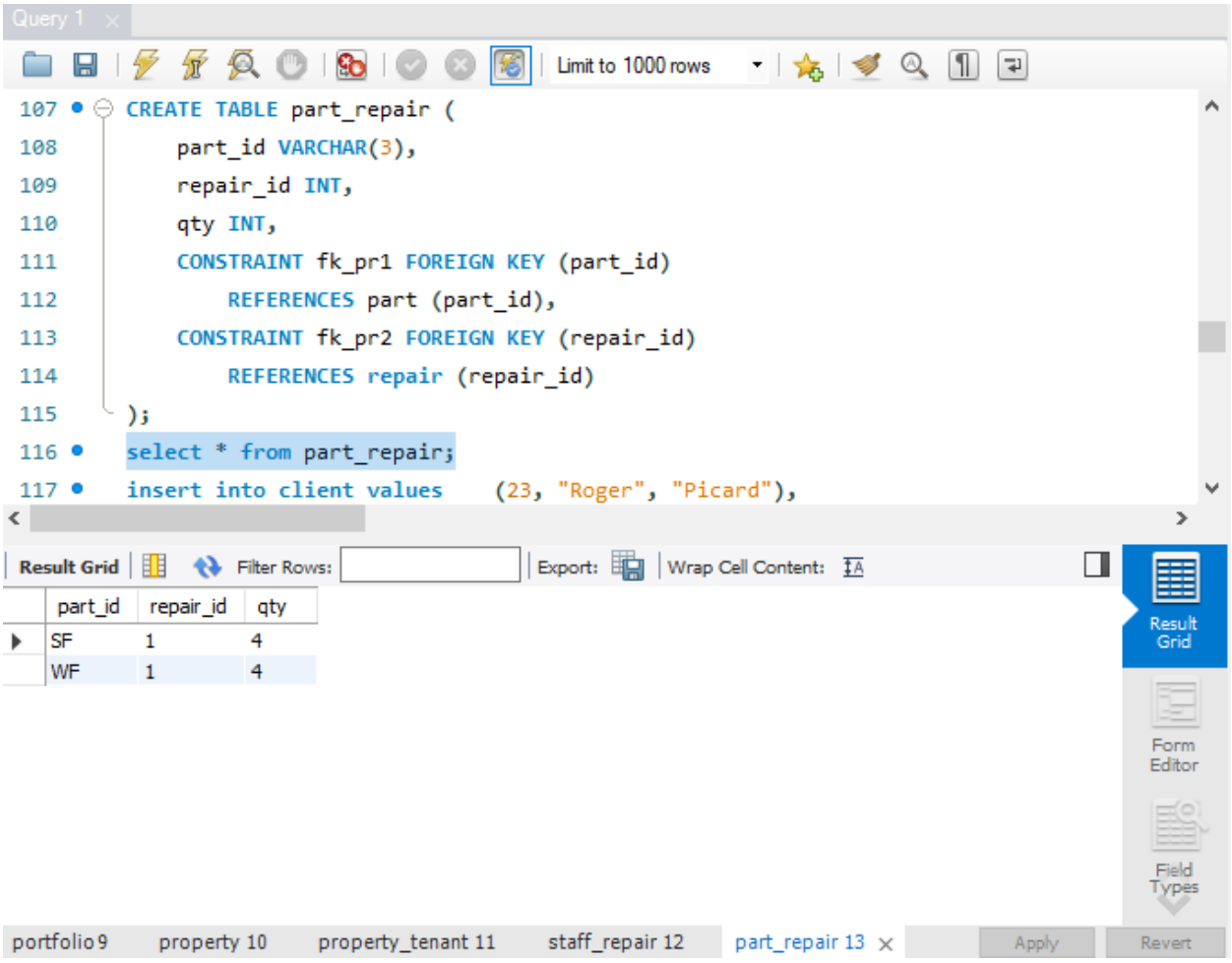


Figure 11: Query (10)

2

a.

```
131 •  insert into part_repair values ("SF",1,4), ("WF",1,4);
132 •  insert into staff_repair values (78,1), (23,1);
133
134    -- a.   Write a query that selects all the portfolios and properties for a particular owner.
135 •  SELECT    *
136    FROM    client AS c,
137        portfolio AS p,
138        property AS pr
139    WHERE    c.client_id = p.client_id
140            AND p.portfolio_id = pr.portfolio_id;
141
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

client_id	client_fname	client_lname	portfolio_id	client_id	property_id	property_address	type_id	portfolio_id
11	Alison	Brown	301	11	3097	11 Kings Street, N1 988	RH	301
11	Alison	Brown	301	11	3099	99 Kings Street, N1 988	RH	301
23	Roger	Picard	201	23	2431	80 Overmeer Rd, SE15 6NQ	RH	201
23	Roger	Picard	201	23	8901	99a Queen Street, N1 2ER	RF	201
23	Roger	Picard	203	23	1990	23 St Anne's Place, N1 8RR	CP	203
23	Roger	Picard	203	23	9088	23 Redding Yard, Bromley-byBow, E2 89Y	CP	203

portfolio 9 | property 10 | property_tenant 11 | staff_repair 12 | part_repair 13 | Result 14 | Result 15 | Apply

Figure 12: Question a solution

b.

```
Limit to 1000 rows
140             AND p.portfolio_id = pr.portfolio_id;
141
142     -- b.   Write a query that selects the tenants and their tenancy dates.
143     SELECT    t.tenant_name, tp.start_date, tp.end_date
144     FROM    tenant AS t,
145         property_tenant AS tp
146     WHERE    t.tenant_id = tp.tenant_id;
147
148
149     -- c.   Write a query that shows all parts involved in the repair of a particular property.
150     SELECT    *
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

tenant_name	start_date	end_date
Dewitt Julio	2017-03-01	2017-12-01
Charisse Spinello	2017-01-01	2018-02-01
Gaslight Software	2005-03-01	2018-03-01
Edgar Kanne	2017-04-03	2017-04-01
Helpline One-Stop Shop	2017-03-01	2021-03-01
Michell Throssell	2017-03-01	2018-03-01

portfolio 9 | property 10 | property_tenant 11 | staff_repair 12 | part_repair 13 | Result 14 | Result 15 | Result 16 | Apply | Revert

Figure 13: Question b solution

```
Query 1
Limit to 1000 rows
149     -- c.   Write a query that shows all parts involved in the repair of a particular property.
150     SELECT    *
151     FROM    repair AS r,
152         part_repair AS pr,
153         part AS p
154     WHERE    r.repair_id = pr.repair_id
155             AND pr.part_id = p.part_id;
156
157     -- d.   Write a query that shows all the tenants for a particularowner.
158     SELECT    c.client_fname, t.tenant_name
159
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

repair_id	repair_detail	property_id	part_id	repair_id	qty	part_id	part_name
1	Replacement Front windows	2431	SF	1	4	SF	Standard Frame
1	Replacement Front windows	2431	WF	1	4	WF	Window Fitting

portfolio 9 | property 10 | property_tenant 11 | staff_repair 12 | part_repair 13 | Result 14 | Result 15 | Result 16 | Apply

Figure 14: Question c solution

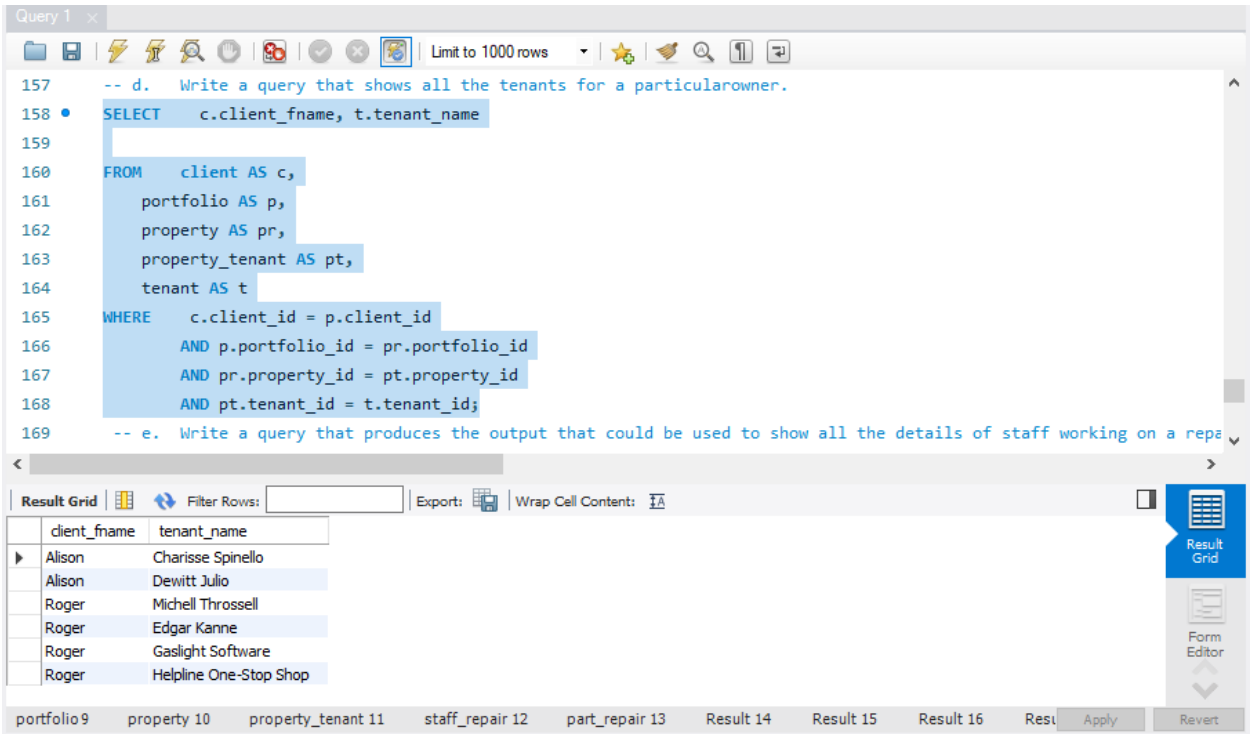


Figure 15: Question d solution

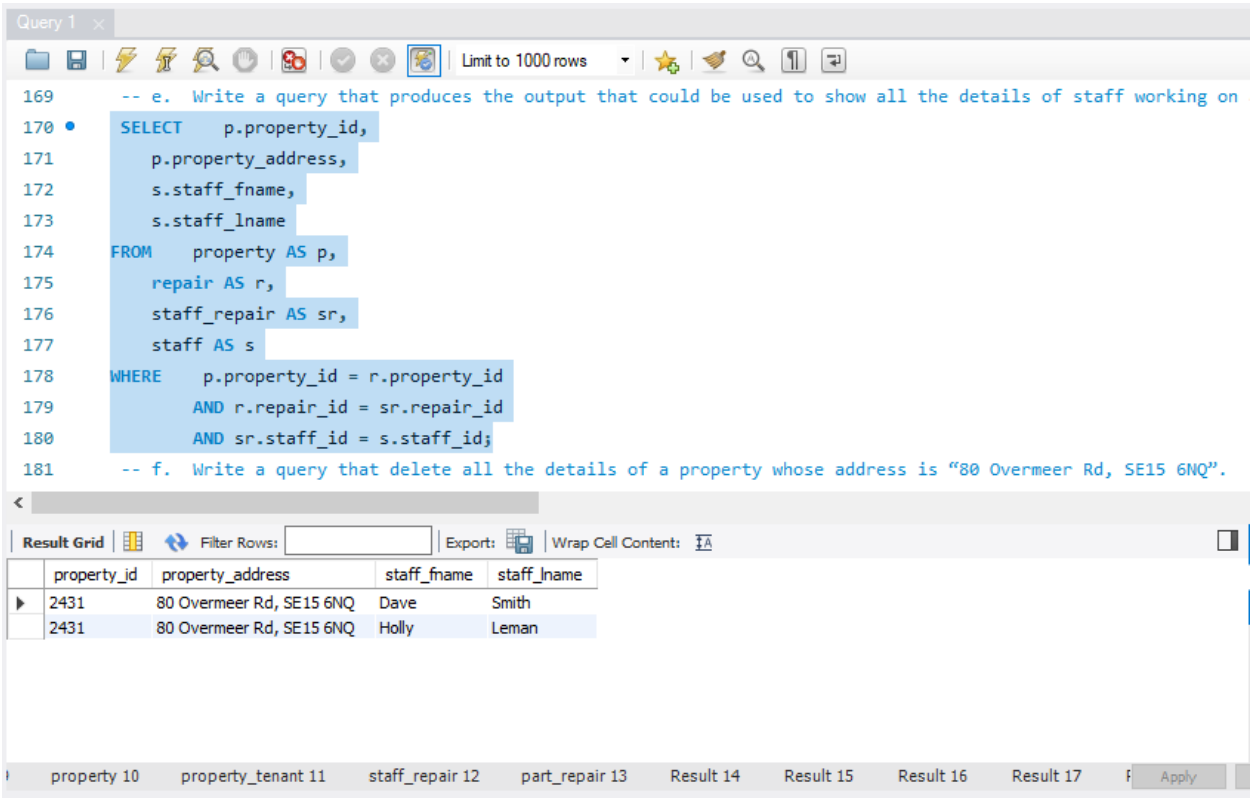


Figure 16: Question e solution

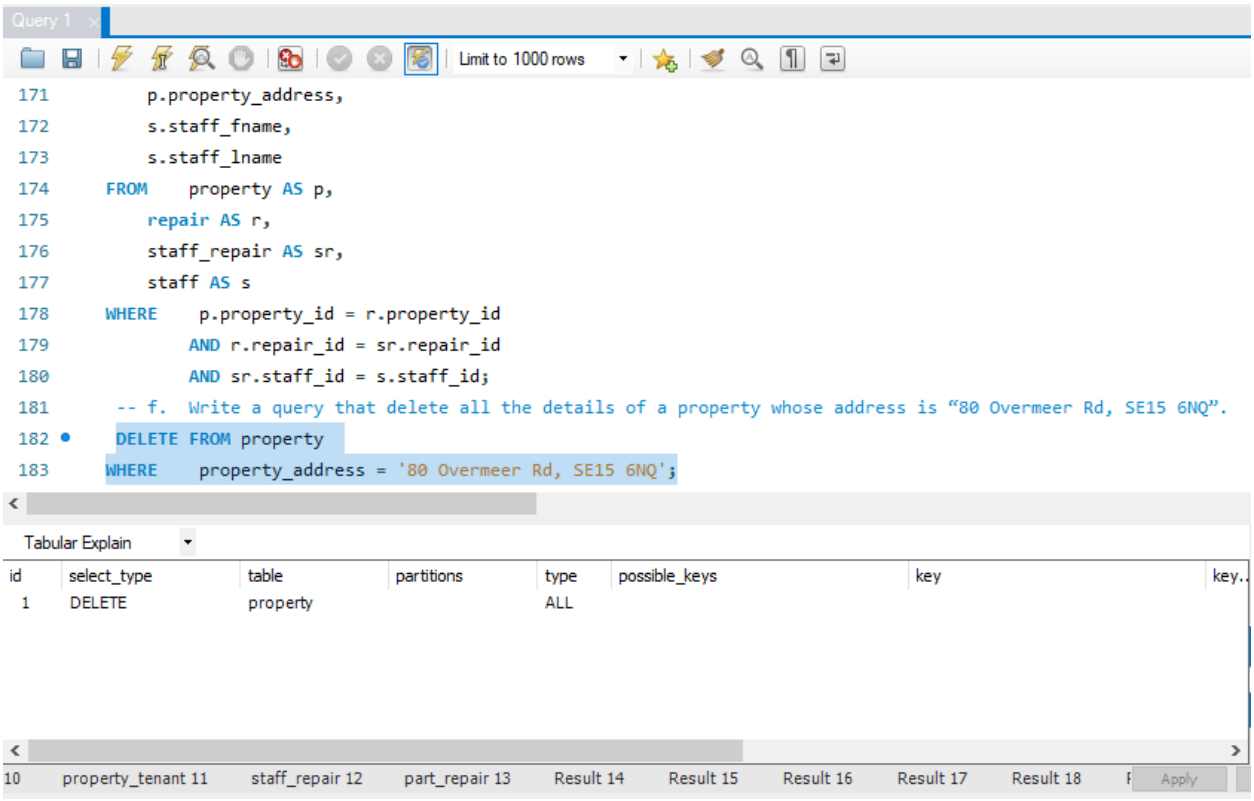


Figure 17: Question f solution

g.

A distributed database is a database that is not restricted to a single system and is spread across numerous places, such as multiple computers or a network of computers. A distributed database system is made up of several locations with no physical components in common. This may be necessary if a database has to be viewed by a large number of people all over the world. This database provides security, consistency and integrity, so this maybe the reason why companies are going for it, to secure the data.

Task -2 Big Data

Aims

The main aim of my assessment was to explore about the information including billionaires list which was published by The Forbes in 2021 which will give me insight about average age about different industries and countries.

Objectives

After analyzing the data from dataset, I will able to understand about information about billionaires and about their age group. This evaluation might be useful for future too, if any sort of comparing is done in future work. I will be able to full get the grasp about which country has the most billionaires and what kind of industry they come from.

To complete my analysis, following are my objectives:

- Collecting data about billionaires from 2021
- Putting the data into analytic tools such as Tableau
- Analyzing and visualizing the data
- Determining the results
- Looking for future work

Collecting the data

To get the data for this assessment, I went to Kaggle website and searched about datasets about billionaires. Upon comparing different datasets, I have chosen this data published by The Forbes by doing the thorough research. The link for the chosen datasets is: https://www.kaggle.com/roysouravcu/forbes-billionaires-of-2021 . There are total of 2756 records in the datasets. The data are shown in Microsoft excel sheet. Following are the screenshot of data in Microsoft Excel.

	A	B	C	D	E	F	G	H
1	Name	NetWorth	Country	Source	Rank	Age	Industry	
2	Jeff Bezos	$177 B	United States	Amazon	1	57	Technology	
3	Elon Musk	$151 B	United States	Tesla, SpaceX	2	49	Automotive	
4	Bernard Arnault & family	$150 B	France	LVMH	3	72	Fashion & Retail	
5	Bill Gates	$124 B	United States	Microsoft	4	65	Technology	
6	Mark Zuckerberg	$97 B	United States	Facebook	5	36	Technology	
7	Warren Buffett	$96 B	United States	Berkshire Hathaway	6	90	Finance & Investments	
8	Larry Ellison	$93 B	United States	software	7	76	Technology	
9	Larry Page	$91.5 B	United States	Google	8	48	Technology	
10	Sergey Brin	$89 B	United States	Google	9	47	Technology	
11	Mukesh Ambani	$84.5 B	India	diversified	10	63	Diversified	
12	Amancio Ortega	$77 B	Spain	Zara	11	85	Fashion & Retail	
13	Francoise Bettencourt M	$73.6 B	France	L'OrÃ©al	12	67	Fashion & Retail	
14	Zhong Shanshan	$68.9 B	China	beverages, pharmaceutic	13	66	Food & Beverage	
15	Steve Ballmer	$68.7 B	United States	Microsoft	14	65	Technology	
16	Ma Huateng	$65.8 B	China	internet media	15	49	Technology	
17	Carlos Slim Helu & family	$62.8 B	Mexico	telecom	16	81	Telecom	
18	Alice Walton	$61.8 B	United States	Walmart	17	71	Fashion & Retail	
19	Jim Walton	$60.2 B	United States	Walmart	18	72	Fashion & Retail	
20	Rob Walton	$59.5 B	United States	Walmart	19	76	Fashion & Retail	
21	Michael Bloomberg	$59 B	United States	Bloomberg LP	20	79	Media & Entertainment	
22	Colin Zheng Huang	$55.3 B	China	e-commerce	21	41	Technology	
23	MacKenzie Scott	$53 B	United States	Amazon	22	50	Technology	
24	Daniel Gilbert	$51.9 B	United States	Quicken Loans	23	59	Finance & Investments	
25	Gautam Adani & family	$50.5 B	India	infrastructure, commodit	24	58	Diversified	
26	Phil Knight & family	$49.9 B	United States	Nike	25	83	Fashion & Retail	
27	Jack Ma	$48.4 B	China	e-commerce	26	56	Technology	
28	Charles Koch	$46.4 B	United States	Koch Industries	27	85	Diversified	
29	Julia Koch & family	$46.4 B	United States	Koch Industries	27	58	Diversified	
30	Masayoshi Son	$45.4 B	Japan	internet, telecom	29	63	Telecom	
31	Michael Dell	$45.1 B	United States	Dell computers	30	56	Technology	

Figure 18: Data in Microsoft Excel

2727	Shin Dong-joo	$1 B	South Korea	retail	2674	67	Fashion & Retail
2728	Scott Smith	$1 B	United States	cloud computing	2674	71	Technology
2729	Zakhar Smushkin	$1 B	Russia	pulp and paper, diversifie	2674	59	Diversified
2730	Marco Squinzi	$1 B	Italy	chemical products	2674	49	Construction & Engineering
2731	Veronica Squinzi	$1 B	Italy	chemical products	2674	49	Construction & Engineering
2732	Axel Stawski	$1 B	United States	real estate	2674	70	Real Estate
2733	Manny Stul	$1 B	Australia	toys	2674	71	Manufacturing
2734	Vlad Tenev	$1 B	United States	stock trading	2674	34	Finance & Investments
2735	Rit Thirakomen	$1 B	Thailand	restaurants	2674	69	Food & Beverage
2736	Tong Judy Wenhong	$1 B	China	e-commerce	2674	51	Technology
2737	Tsai Chi-jui	$1 B	Taiwan	shoes	2674	81	Fashion & Retail
2738	Surin Upatkoon	$1 B	Thailand	telecom, lotteries, insura	2674	71	Diversified
2739	Ruben Vardanyan	$1 B	Russia	investment banking	2674	52	Finance & Investments
2740	Murat Vargi	$1 B	Turkey	telecom	2674	73	Telecom
2741	Vlad Vendrow & family	$1 B	United States	software	2674	53	Technology
2742	Wang Qiangxiang	$1 B	China	artificial turf	2674	49	Service
2743	Wang Wenjian	$1 B	China	optical devices	2674	74	Technology
2744	J. Wayne Weaver	$1 B	United States	Shoes	2674	85	Diversified
2745	Sandy Weill	$1 B	United States	Citigroup	2674	88	Finance & Investments
2746	Xia Zhisheng & family	$1 B	China	home appliances	2674	79	Technology
2747	Xu Jin	$1 B	China	wine	2674	56	Food & Beverage
2748	Vadim Yakunin	$1 B	Russia	pharmacy	2674	58	Healthcare
2749	Mark Haoyong Yang	$1 B	China	e-commerce	2674	46	Technology
2750	Yao Hsiao Tung	$1 B	Singapore	Manufacturing	2674	81	Manufacturing
2751	Yu De-Chao	$1 B	United States	pharmaceuticals	2674	57	Healthcare
2752	Daniel Yong Zhang	$1 B	China	e-commerce	2674	49	Technology
2753	Zhang Yuqiang	$1 B	China	Fiberglass	2674	65	Manufacturing
2754	Zhao Meiguang	$1 B	China	gold mining	2674	58	Metals & Mining
2755	Zhong Naixiong	$1 B	China	conglomerate	2674	58	Diversified
2756	Zhou Wei family	$1 B	China	Software	2674	54	Technology
2757							

Figure 19: Data in Microsoft Excel (2)

Analyzing data in tableau

Manually putting the dataset file in analytic tool ‘tableau’. Tableau is a data visualization and analytics application that is widely utilized in the business today. For data-science-related employment, many firms consider it crucial. The fact that Tableau features a drag-and-drop interface contributes to its simplicity of usage. This functionality makes it simple and quick to accomplish operations like sorting, comparing, and analyzing. Tableau is also interoperable with a variety of data sources, including Excel, SQL Server, and cloud-based data repositories, making it a great option for Data Scientists.

The dataset was imported and analyzed thoroughly to achieve the aim set for it. If we manually look the data, we cannot understand quite nicely or it will be harder for us to understand. To make us understand quite easily, visualization helps us to do that.

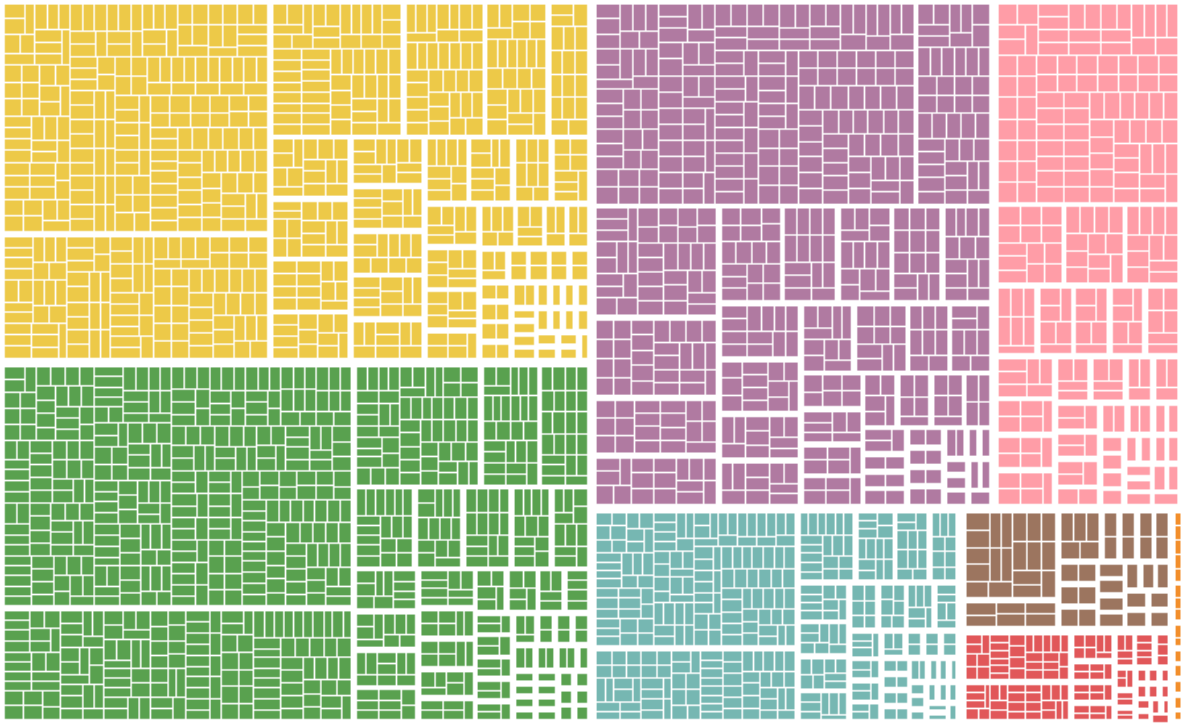


Figure 20: Data shown with respective of their age group

In the above picture, we can see distinctive colors where same colors are grouped together. All the data are shown here with respective of their age group. Data is grouped according to their respective age. But we cannot verify it, because we don’t know the exact number. By supposedly, we can say that.

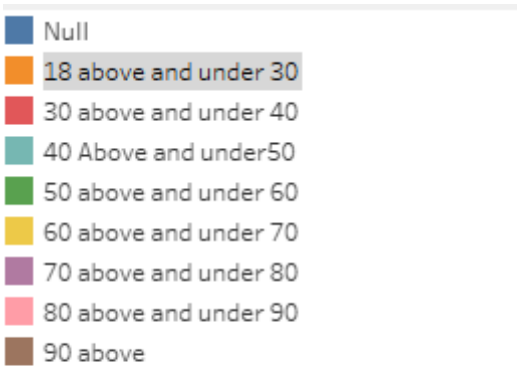


Figure 21: Different color representing different age group

As we can see, these are the colors representing the data in figure 3. In figure 3, we can see there are most billionaires in 60 above and under 70 age group compared to other.

	Age (group)									
Country	18 above..	30 above..	40 Abov..	50 above..	60 above..	70 above..	80 above..	90 above	Other	Grand To..
Algeria						1				1
Argentina						1	2	1	1	5
Australia	1	2	6	6	8	14	6		1	44
Austria			2	1	2	5	1	1		12
Belgium			1	1		1				3
Brazil	1	5	9	11	12	17	6		4	65
Canada		4	9	14	15	10	3	5	4	64
Chile				1	2	5	1			9
China	1	25	122	311	104	42	5		16	626
Colombia		1			1		3			5
Cyprus			1	2	1	1				5
Czechia			2	5	2					9
Denmark		3	3	1		2			1	10
Egypt				2	1	2		1		6
Eswatini (Swaziland)							1			1
Finland		1		2	2	1	1			7
France		1	3	9	9	12	6	1	1	42
Georgia			1		1					2
Germany	2	7	13	26	26	27	12	2	21	136
Greece				1	2		1			4
Guernsey					1					1

Figure 22: Data shown in age group and country along with grand total

In this figure 5, data are again shown in respective of their age group along with the country but here we can see the numbers that will make this study easy. If we give numbers, we can easily understand how many billionaires are there in the country in certain age group. Look at the data of China, there are total of 626 billionaires and the most billionaire are in age of 50 above and 60 under, numbering 311. If we show the numbers, we can easily compare the data and get good understanding of the data.

	Age (group)									
Country	18 above..	30 above..	40 Abov..	50 above..	60 above..	70 above..	80 above..	90 above	Other	Grand To..
Russia		5	18	56	34	5				118
Singapore		2	2	3	10	6	1	2	1	27
Slovakia				2						2
South Africa				1	2	2				5
South Korea			11	14	11	6	1			43
Spain			2	6	5	10	5		2	30
St. Kitts and Nevis							1			1
Sweden		5	8	10	7	10	1			41
Switzerland		1	6	9	7	12	4	1		40
Taiwan				3	17	17	6	2	2	47
Tanzania			1							1
Thailand			2	3	9	8	5	1	3	31
Turkey		1	1	5	8	8	3	1		27
Ukraine			1	6						7
United Arab Emirates					1				3	4
United Kingdom		1	5	12	15	13	7	1	2	56
United States	5	30	66	143	192	175	90	19	4	724
Venezuela					1					1
Vietnam			2	3	1					6
Zimbabwe				1						1
Grand Total	15	106	338	757	646	536	223	55	79	2,755

Figure 23: Data shown in age group and country along with grand total

In figure 6, now we can see in which age group there is most billionaires because there is number and at the last there is grand total which helps us to know about the required data. In 50 years and 60 under, there is most billionaires resulting to 757. Now we can be 100% sure, and we can say that because of numbers included. In figure 3, there also we can say about it but its like beating in the bush kind of a situation.

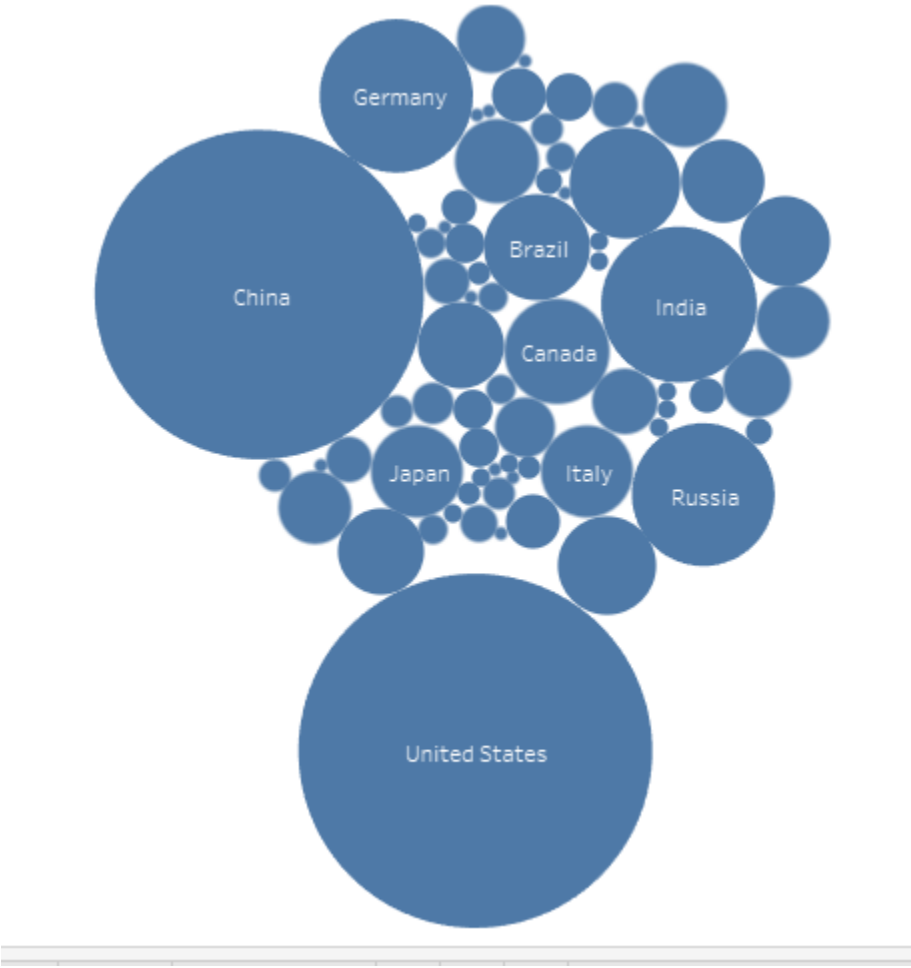


Figure 24: Data shown in bubble along with net worth

In figure 7, data are being shown in bubble with country and net worth. Net worth is being totaled and that totaled is referred to the size so, more net worth means bigger in size.

Figure 25: Data are shown in colorful variation with country and their respective age average

Here in this figure, I tired to take the average age of the country and put it according in ascending orders and colorful variation is used.

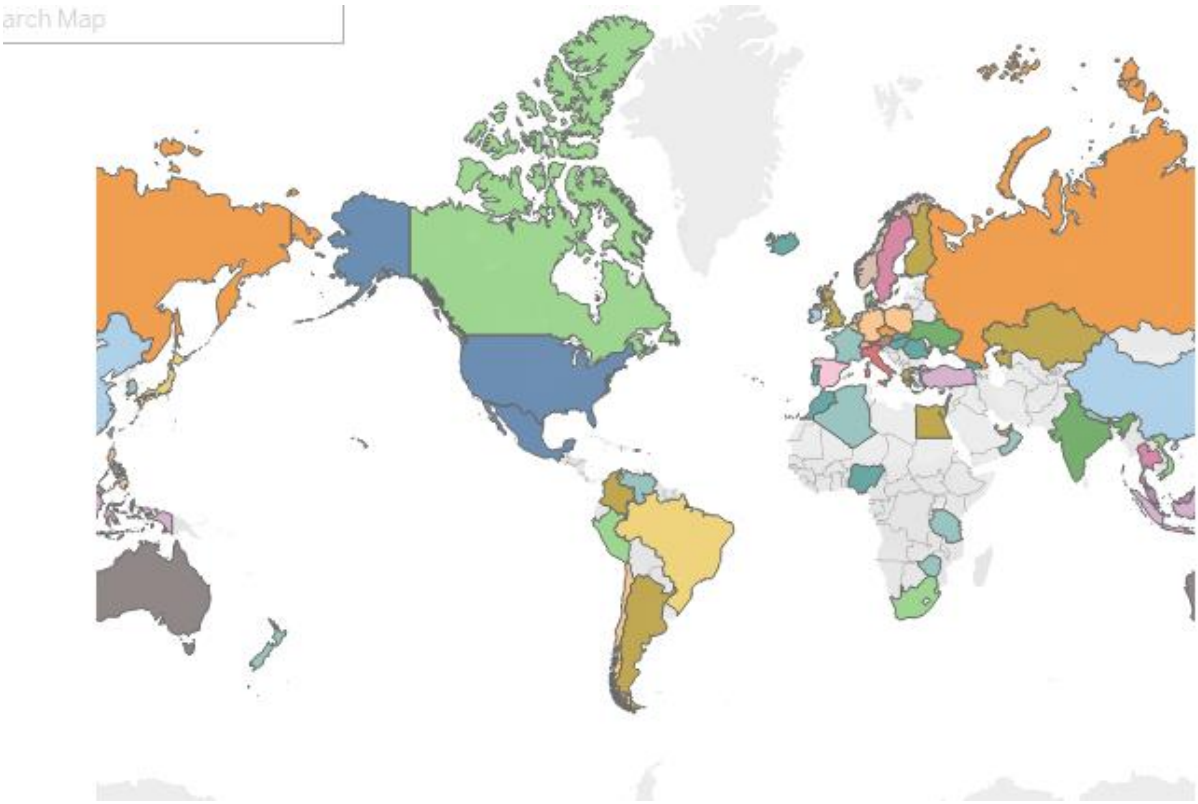


Figure 26: Ranked among the countries

In this figure 9, I have tried to rank the countries by number of their billionaires present there, so by counting the billionaires I tried to rank them, United States come first and China second.

`

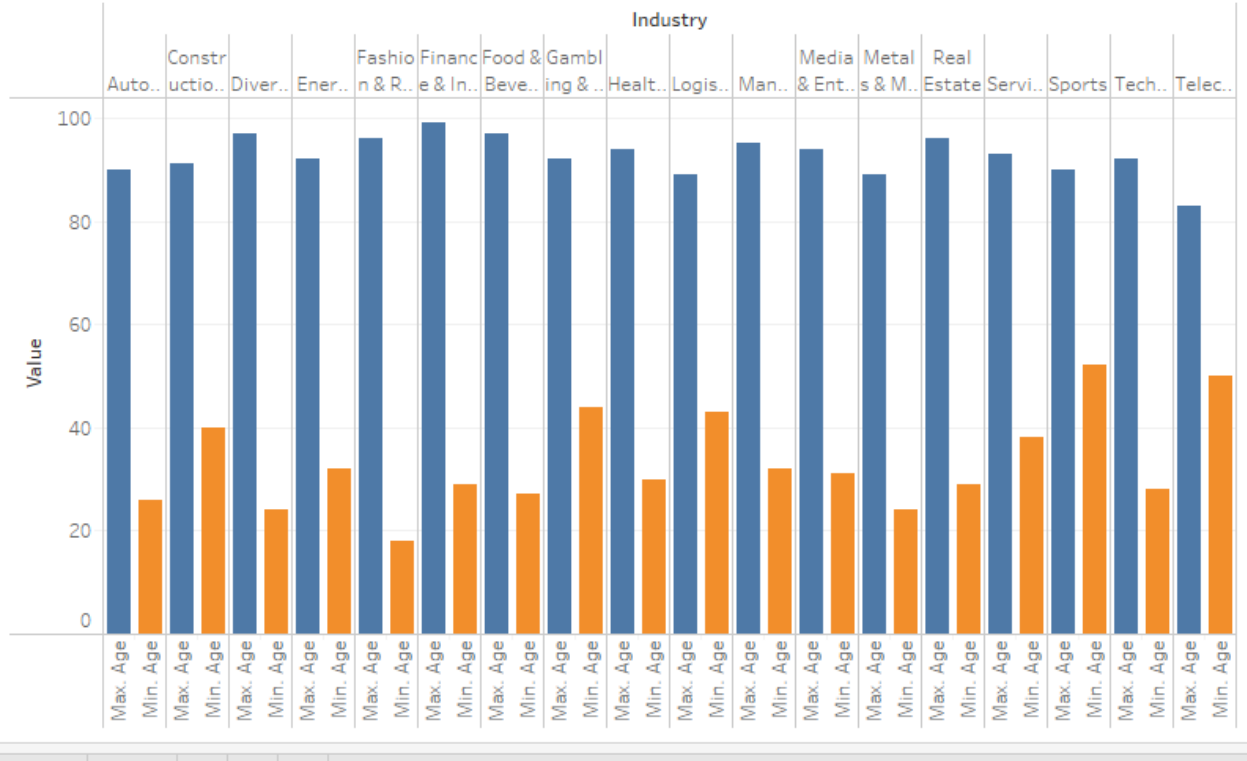
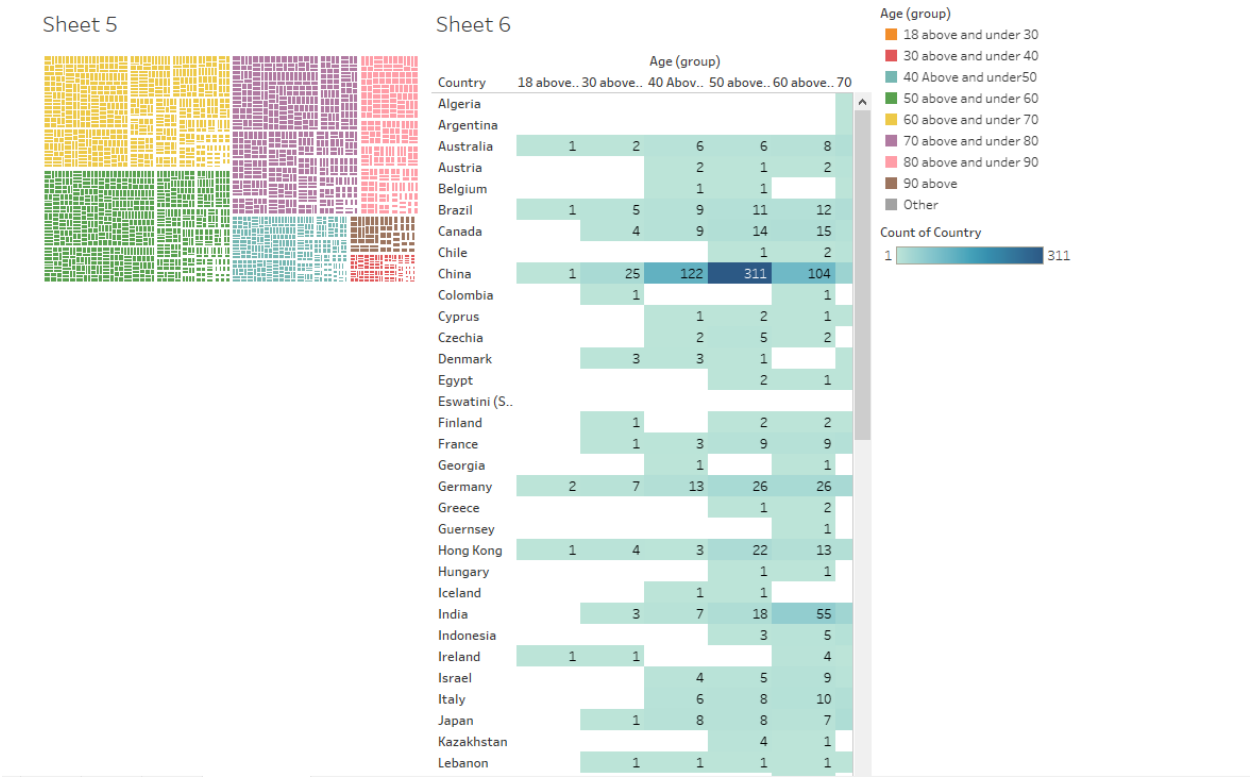


Figure 27: Minimum and Maximum age for respective industry

In this figure 10, I have tried to show the data of minimum and maximum of each industry.



Country	18 above..	30 above..	40 Abov..	50 above..	60 above..	70
Algeria						
Argentina						
Australia	1	2	6	6	8	
Austria			2	1	2	
Belgium			1	1		
Brazil	1	5	9	11	12	
Canada		4	9	14	15	
Chile				1	2	
China	1	25	122	311	104	
Colombia		1			1	
Cyprus			1	2	1	
Czechia			2	5	2	
Denmark		3	3	1		
Egypt				2	1	
Eswatini (S..						
Finland		1		2	2	
France		1	3	9	9	
Georgia			1		1	
Germany	2	7	13	26	26	
Greece				1	2	
Guernsey					1	
Hong Kong	1	4	3	22	13	
Hungary				1	1	
Iceland			1	1		
India		3	7	18	55	
Indonesia				3	5	
Ireland	1	1			4	
Israel			4	5	9	
Italy			6	8	10	
Japan		1	8	8	7	
Kazakhstan				4	1	
Lebanon		1	1	1	1	

Figure 28: Comparing the data of two different sheets

Conclusion

In conclusion, I have found that there are currently 2755 billionaires (till this date), among them the most billionaires are from United States and 2nd most of them are from China. The more billionaires, the more they amass the net worth, so total net worth from a single country is also highest in United States. In China, in the age group of 50 above and 60 under, there is 311 billionaires which is highest in the age group.

Future Work

In future, maybe in 2022, we can compare the data of 2021 and 2022, do some extensive research about it. As the datasets, I choose somehow smaller datasets which didn't have detailed things but in future I can take detailed datasets of these billionaires and can-do detailed analysis about them.

Task -3 (Recommendation System)

Introduction

The computer programs that provide recommendations to users based on a variety of criteria are recommendation systems. These algorithms forecast the most likely product that consumers will buy and that they will be interested in. This type of recommendation systems is used in big companies like amazon, Netflix, YouTube, etc. to assist their customer in finding the right product or movie for them.

For this project, recommendation system was asked to build for video streaming platforms. So, I have chosen Amazon Prime Video to build the recommendation system around them.

Current Status

To make this recommendation system for Amazon Prime Video, as I was searching for the files related to this, I couldn't find anything and it was hard for me to find some files too. I looked in YouTube how to make this type of dataset myself, with the help of YouTube video I made a dataset file for this myself.

	A	B	C	D	E	F	G	H
1		action 1	action 2	action 3	romantic 1	romantic 2	romantic 3	
2	user 1	1	4	5		2	1	
3	user 2	2	5	3	3	2		
4	user 3	3	1		4	5	4	
5	user 4		2	1	4		3	
6	user 5	1		2	3	3	4	
7								
8								
9								
10								
11								

Figure 29: CSV File

	movieId	title	genres
2	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy
3	2	Jumanji (1	Adventure|Children|Fantasy
4	3	Grumpier	Comedy|Romance
5	4	Waiting to	Comedy|Drama|Romance
6	5	Father of	Comedy
7	6	Heat (199	Action|Crime|Thriller
8	7	Sabrina (1	Comedy|Romance
9	8	Tom and H	Adventure|Children
10	9	Sudden De	Action
11	10	GoldenEye	Action|Adventure|Thriller
12	11	American	Comedy|Drama|Romance
13	12	Dracula: D	Comedy|Horror
14	13	Balto (199	Adventure|Animation|Children
15	14	Nixon (19	Drama
16	15	Cutthroat	Action|Adventure|Romance
17	16	Casino (19	Crime|Drama
18	17	Sense and	Drama|Romance
19	18	Four Roon	Comedy
20	19	Ace Ventu	Comedy
21	20	Money Tra	Action|Comedy|Crime|Drama|Thriller
22	21	Get Shorty	Comedy|Crime|Thriller
23	22	Copycat (1	Crime|Drama|Horror|Mystery|Thriller
24	23	Assassins	Action|Crime|Thriller
25	24	Powder (1	Drama|Sci-Fi
26	25	Leaving La	Drama|Romance

Figure 30: CSV file (1)

1	userId	movieId	rating	timestamp
2	1	1	4	9.65E+08
3	1	3	4	9.65E+08
4	1	6	4	9.65E+08
5	1	47	5	9.65E+08
6	1	50	5	9.65E+08
7	1	70	3	9.65E+08
8	1	101	5	9.65E+08
9	1	110	4	9.65E+08
10	1	151	5	9.65E+08
11	1	157	5	9.65E+08
12	1	163	5	9.65E+08
13	1	216	5	9.65E+08
14	1	223	3	9.65E+08
15	1	231	5	9.65E+08
16	1	235	4	9.65E+08
17	1	260	5	9.65E+08
18	1	296	3	9.65E+08
19	1	316	3	9.65E+08
20	1	333	5	9.65E+08
21	1	349	4	9.65E+08
22	1	356	4	9.65E+08
23	1	362	5	9.65E+08
24	1	367	4	9.65E+08
25	1	423	3	9.65E+08
26	1	441	4	9.65E+08

Figure 31: CSV file (2)

Amazon Prime Video currently uses 4k as max streaming video quality. So, 4k video quality takes more bandwidth too. Amazon Prime Video uses the Amazon Web Service (AWS) cloud as the underlying technology for all its services. This video services used Amazon DynamoDB which helps to optimize scalability and performance. Amazon DynamoDB is fast NOSQL key-valued database.

Aim for the collection of the data

Above mentioned csv file is made by myself because I couldn't find the files related to this. The main aim of making this recommendation system is to recommend the user by collaborative filtering. Collaborative filtering filters data based on other users' interactions and data. It's based on the premise that individuals who agreed on a given item's rating are more likely to agree again in the future.

The infrastructure required for this system to work properly is to keep the servers locally rather than keeping them on cloud. As cloud system relies online so much but we are currently focusing locally. Servers should be made with high storage with hard drives and 8 GB ram should be used for the servers. The bandwidth is high because we want to stream it on 4k so storage should be more. The databases should be kept locally as the servers are kept locally for this current setup.

Quality of Service

Any system that regulates data flow to decrease packet loss, latency, and jitter on a network is known as quality of service (QoS). QoS regulates and controls network resources by allocating resources to specified categories of data. The QoS baseline recommendation for this system is DSCP CS4. The bandwidth required is around 25 Mbps because the video will be in streaming in 4k. And the jitter should be around 10ms-50ms. The loss should be no more than 5 percent and latency should be no more than 4 or 5 seconds.

Mainly the QoS is focused on data right now where the data is prioritized first. So, data can smoothly be shown in user end. Traffic prioritization and route selection should be done to achieve the QoS. There are different types of QoS like application QoS where the bandwidth used by application is being controlled, IP QoS where the bandwidth used by IP is being controlled and lastly role QoS where the bandwidth of selected roles is being controlled.

Types of Databases

Traditional data is structured data that is maintained by a wide range of enterprises, from small firms to large corporations. A centralized database design was used in traditional database systems to store and preserve data in a set structure or fields in a file.

A cloud database is a database service that is created, installed, and provided via could platform. It’s basically a cloud Platform as a Service (PaaS) delivery paradigm that allows businesses, end users, and their applications to store, manage and retrieve from the cloud.

The hybrid database uses both traditional database and cloud-based database according its use. And we are using hybrid database so that when we want flexibility. Because we want to access database locally and from online too according to the needs.

Data Analysis

After taking the data to make the recommendation system, the data was analyzed further more. Because there are different types of recommendation system where different techniques can be used to filter it and make the recommendation. There are three main types of recommendation system that are collaborative filtering, content-based filtering and hybrid of the two. Collaborative filtering is where the data are based on other uses’ interactions and data. It is based when the users might agree on the same stuff in the future. Content-based filtering is a machine learning approach that makes choices based on feature similarity. This strategy is frequently employed in recommendation systems, which are programs that promote or recommend items to people based on information gathered about them. Hybrid filtering is simply means combining both of them, to recommend users on both content-based and collaborative based.

Our filtration of the data is based on collaborative-based filtering where we suggest the user videos through this. The concept behind collaborative filtering is that consumers generally get the best recommendation from others who have similar preferences to them. Techniques for pairing users with common interests and creating suggestions based on this include collaborative filtering.

Currently we are not taking QoS into consideration because we want to make sure that the recommendation system works finely. After achieving that we can focus on QoS in future to make the quality of service.

Recommendation System

By using jupyter notebook, all the algorithm was conducted and below are the screenshots.

In [1]:
```
print('Juoyter Notebook')
```

```
Juoyter Notebook
```

In [2]:
```
import pandas as pd
from scipy import sparse
from sklearn.metrics.pairwise import cosine_similarity
```

In [4]:
```
ratings= pd.read_csv("dataset.csv", index_col=0)
ratings= ratings.fillna(0)
ratings
```

Out[4]:

	action 1	action 2	action 3	romantic 1	romantic 2	romantic 3
user 1	1.0	4.0	5.0	0.0	2.0	1.0
user 2	2.0	5.0	3.0	3.0	2.0	0.0
user 3	3.0	1.0	0.0	4.0	5.0	4.0
user 4	0.0	2.0	1.0	4.0	0.0	3.0
user 5	1.0	0.0	2.0	3.0	3.0	4.0

In [5]:
```
def standardize(row):
    new_row = (row - row.mean()) / (row.max() - row.min())
    return new_row

ratings_std= ratings.apply(standardize)
ratings_std
```

Figure 32: Recommendation System Algorithm

Out[5]:

	action 1	action 2	action 3	romantic 1	romantic 2	romantic 3
user 1	-0.133333	0.32	0.56	-0.70	-0.08	-0.35
user 2	0.200000	0.52	0.16	0.05	-0.08	-0.60
user 3	0.533333	-0.28	-0.44	0.30	0.52	0.40
user 4	-0.466667	-0.08	-0.24	0.30	-0.48	0.15
user 5	-0.133333	-0.48	-0.04	0.05	0.12	0.40

In [6]:
```
def standardize(row):
    new_row = (row - row.mean()) / (row.max() - row.min())
    return new_row

ratings_std= ratings.apply(standardize)


item_similarity=cosine_similarity(ratings_std.T)
print(item_similarity)
```

```
[[ 1.          0.02114775 -0.27357645  0.18681618  0.86904819  0.02414023]
 [ 0.02114775  1.          0.66437118 -0.484248   -0.38492561 -0.98222397]
 [-0.27357645  0.66437118  1.         -0.9333387  -0.31480009 -0.74407295]
 [ 0.18681618 -0.484248   -0.9333387   1.          0.11725441  0.53602017]
 [ 0.86904819 -0.38492561 -0.31480009  0.11725441  1.          0.39393939]
 [ 0.02414023 -0.98222397 -0.74407295  0.53602017  0.39393939  1.        ]]
```

In [7]:
```
item_similarity_df=pd.DataFrame(item_similarity, index=ratings.columns,columns=ratings.columns)
item_similarity_df
```

Figure 33: Recommendation System Algorithm (1)

	action 1	action 2	action 3	romantic 1	romantic 2	romantic 3
action 1	1.000000	0.021148	-0.273576	0.186816	0.869048	0.024140
action 2	0.021148	1.000000	0.664371	-0.484248	-0.384926	-0.982224
action 3	-0.273576	0.664371	1.000000	-0.933339	-0.314800	-0.744073
romantic 1	0.186816	-0.484248	-0.933339	1.000000	0.117254	0.536020
romantic 2	0.869048	-0.384926	-0.314800	0.117254	1.000000	0.393939
romantic 3	0.024140	-0.982224	-0.744073	0.536020	0.393939	1.000000

In [11]:
```
## Lets make recommendations

def get_similar_movies(movie_name,user_rating):
    similar_score = item_similarity_df[movie_name]*(user_rating-2.5)
    similar_score = similar_score.sort_values(ascending=False)

    return similar_score

print(get_similar_movies("romantic 3",1))
```

```
action 2      1.473336
action 3      1.116109
action 1     -0.036210
romantic 2   -0.590909
romantic 1   -0.804030
romantic 3   -1.500000
Name: romantic 3, dtype: float64
```

Figure 34: Recommendation System Algorithm (2)

```
Name: romantic 3, dtype: float64
```

In [14]:
```python
action_lover = [("action 1",5), ("romantic 2", 1), ("romantic 3", 2)]

similar_movies=pd.DataFrame()

for movie, rating in action_lover:
    similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)

similar_movies.head()
similar_movies.sum().sort_values(ascending=False)
```

```
C:\Users\JIWON\AppData\Local\Temp\ipykernel_10904\2730301076.py:6: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)
C:\Users\JIWON\AppData\Local\Temp\ipykernel_10904\2730301076.py:6: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)
C:\Users\JIWON\AppData\Local\Temp\ipykernel_10904\2730301076.py:6: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)
```

Out[14]:
```
action 1      1.184358
action 2      1.121370
romantic 2    0.475651
action 3      0.160295
romantic 1    0.023149
romantic 3   -1.030559
dtype: float64
```

Figure 35: Recommendation System Algorithm (3)

In [11]:
```python
ratings = pd.read_csv('ratings.csv')
movies = pd.read_csv('movies.csv')
ratings = pd.merge(movies, ratings).drop(['genres', 'timestamp'],axis=1)
ratings.head()
```

Out[11]:

	movieId	title	userId	rating
0	1	Toy Story (1995)	1	4.0
1	1	Toy Story (1995)	5	4.0
2	1	Toy Story (1995)	7	4.5
3	1	Toy Story (1995)	15	2.5
4	1	Toy Story (1995)	17	4.5

In [13]:
```python
user_ratings = ratings.pivot_table(index=['userId'],columns=['title'],values='rating')
user_ratings.head()
```

Out[13]:

title	'71 (2014)	'Hellboy': The Seeds of Creation (2004)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'Tis the Season for Love (2015)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)	...	Zulu (2013)	[REC] (2007)	[REC]² (2009)	[REC]³ 3 Génesis (2012)	anohana: The Flower We Saw That Day - The Movie (2013)	eXistenZ (1999)	x (20
userId																		
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	N
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	N

Figure 36: Recommendation System Algorithm (4)

In [14]:
```
user_ratings = user_ratings.dropna(thresh=10,axis=1).fillna(0)
user_ratings.head()
```

Out[14]:

title	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)	12 Angry Men (1957)	12 Years a Slave (2013)	127 Hours (2010)	...	Zack and Miri Make a Porno (2008)	Zero Dark Thirty (2012)	Zero Effect (1998)	Zodiac (2007)	Zombieland (2009)	Zoolander (2001)
userId																	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 2269 columns

In [15]:
```
## let's build the similairty
item_similarity_df = user_ratings.corr(method='pearson')
item_similarity_df.head(50)
```

Out[15]:

title	'burbs, The	(500) Days of	10 Cloverfield	10 Things I Hate	10,000 BC	101 Dalmatians	101 Dalmatians (One Hundred	12 Angry Men	12 Years a Slave	127 Hours	...	Zack and Miri Make a	Zero Dark	Ze Effe

Figure 37: Recommendation System Algorithm (5)

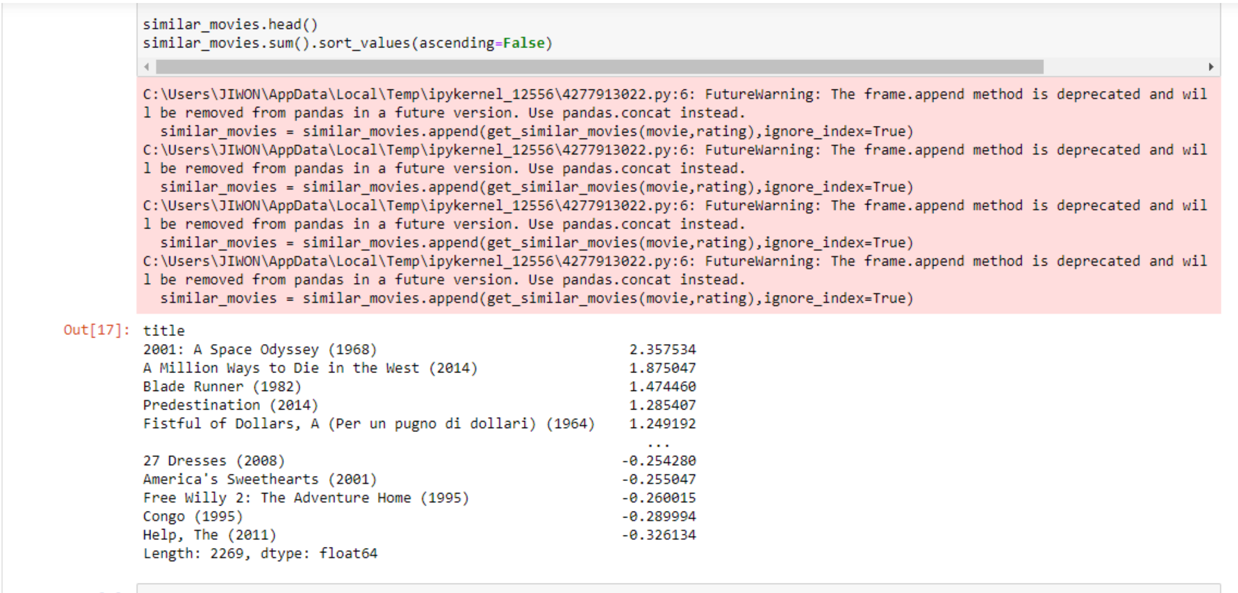
(500) Days of Summer (2009)	0.063117	1.000000	0.142471	0.273989	0.193960	0.148903	0.142141	0.159756	0.135486	0.200135	...	0.374515	0.178655	0.0684
10 Cloverfield Lane (2016)	-0.023768	0.142471	1.000000	-0.005799	0.112396	0.006139	-0.016835	0.031704	-0.024275	0.272943	...	0.242663	0.099059	-0.0234
10 Things I Hate About	0.143482	0.273989	-0.005799	1.000000	0.244670	0.223481	0.211473	0.011784	0.091964	0.043383		0.243118	0.104858	0.1324

In [16]:
```
def get_similar_movies(movie_name,user_rating):
    similar_score = item_similarity_df[movie_name]*(user_rating-2.5)
    similar_score = similar_score.sort_values(ascending=False)

    return similar_score
```

In [17]:
```
action_lover = [("A Million Ways to Die in the West (2014)",5), ("21 Jump Street (2012)", 1), ("48 Hrs. (1982)", 2),("2001: A Spa

similar_movies=pd.DataFrame()

for movie, rating in action_lover:
    similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)

similar_movies.head()
similar_movies.sum().sort_values(ascending=False)
```

```
C:\Users\JIWON\AppData\Local\Temp\ipykernel_12556\4277913022.py:6: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)
C:\Users\JIWON\AppData\Local\Temp\ipykernel_12556\4277913022.py:6: FutureWarning: The frame.append method is deprecated and wil
```

Figure 38: Recommendation System Algorithm (6)



```
similar_movies.head()
similar_movies.sum().sort_values(ascending=False)

C:\Users\JIWON\AppData\Local\Temp\ipykernel_12556\4277913022.py:6: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)
C:\Users\JIWON\AppData\Local\Temp\ipykernel_12556\4277913022.py:6: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)
C:\Users\JIWON\AppData\Local\Temp\ipykernel_12556\4277913022.py:6: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)
C:\Users\JIWON\AppData\Local\Temp\ipykernel_12556\4277913022.py:6: FutureWarning: The frame.append method is deprecated and wil
l be removed from pandas in a future version. Use pandas.concat instead.
  similar_movies = similar_movies.append(get_similar_movies(movie,rating),ignore_index=True)

Out[17]: title
2001: A Space Odyssey (1968)                               2.357534
A Million Ways to Die in the West (2014)                   1.875047
Blade Runner (1982)                                        1.474460
Predestination (2014)                                      1.285407
Fistful of Dollars, A (Per un pugno di dollari) (1964)     1.249192
                                                             ...
27 Dresses (2008)                                         -0.254280
America's Sweethearts (2001)                              -0.255047
Free Willy 2: The Adventure Home (1995)                   -0.260015
Congo (1995)                                              -0.289994
Help, The (2011)                                          -0.326134
Length: 2269, dtype: float64
```

Figure 39:Recommendation System Algorithm (7)

Data collected after implementing

After implementing this recommendation system using collaborative-based filtering, above data was acquired. The data that was acquired shows that the sci-fi movie was given more ratings, as the movie with emotional feelings are given less rating here. So, more ratings mean, it will be recommended to user more likely and less rating means, it won't be recommended to user. Now, the recommendation system will compare with other users and based on how much they are similar, recommends you the video as other user's choice because it is quite similar.

Feedback Loops

The recommendation system really works as we can see above. The user's choice was similar and the movie recommended was similar. It is not fine tuned but still the recommendation system works. To make it work perfectly, more different type of algorithm should be used and content-based filtering should be also done to get the based result. So, this recommendation system which is based on collaborative filtering really works and sometimes only it misses it's target.