

## **BANKING SERVICES & COMPLAINTS**

### Final Report

Group 24

Ravi Khimjibhai Patel

Varun Kumar Kumaravel

781-921-3349

617-407-7349

[patal.ravikh@northeastern.edu](mailto:patal.ravikh@northeastern.edu)

[kumaravel.v@northeastern.edu](mailto:kumaravel.v@northeastern.edu)

**Percentage of Effort Contributed by Student1: 50%**

**Percentage of Effort Contributed by Student2: 50%**

**Signature of Student 1:** 

**Signature of Student 2:** 

**Submission Date: 04/23/2023**

## **OVERVIEW AND INTRODUCTION**

Most industries are being affected by technology's daily advancements in the current, developing world. Banking and finance are one such significantly benefiting industries or organizations. Today, a significant portion of banking operations are moving from an offline to an online environment. The younger generation is already familiar with cutting-edge technology, but those still utilizing old means are also learning about the newer ones. Customer satisfaction and the relationship between a customer and a bank in the case of banking are crucial factors in the success of any institution, whether it is involved in banking, finance, IT, etc. The most asset is feedback since it constantly keeps businesses, policies, services, and profits under check. Since practically every organization in the banking sector now offers the same services due to increased market competition, there is less potential for competition. As a result, a bank has an advantage over rival banks due to its customer experience. Customers and bank employees can communicate via a variety of channels, including the Internet, mobile banking, face-to-face interactions, and ATMs. As a result, the bank must prioritize offering a seamless service since customers want the same level of care across all channels of contact.

Our banking services' effectiveness and customer feedback The MySQL database is a useful tool for acquiring knowledge about our financial services' efficiency and pinpointing development areas. The database includes details on our many services, performance indicators, and client comments. By examining this data, we can better understand how our services are being used, how well they are working, and how happy our clients are with their experiences.

We share the results of our examination of the database's data in this report. To pinpoint significant areas of strength and weakness in our services, we have examined trends and patterns in the data. To see how we compare against the competition, we have also compared our performance to that of other banks. Our investigation did, however, also point out a few areas that needed work. For instance, compared to our online services, our customer satisfaction ratings for in-person services like teller transactions and loan applications are lower. This suggests that to better meet the needs of our clients, we might need to concentrate on raising the caliber of these services.

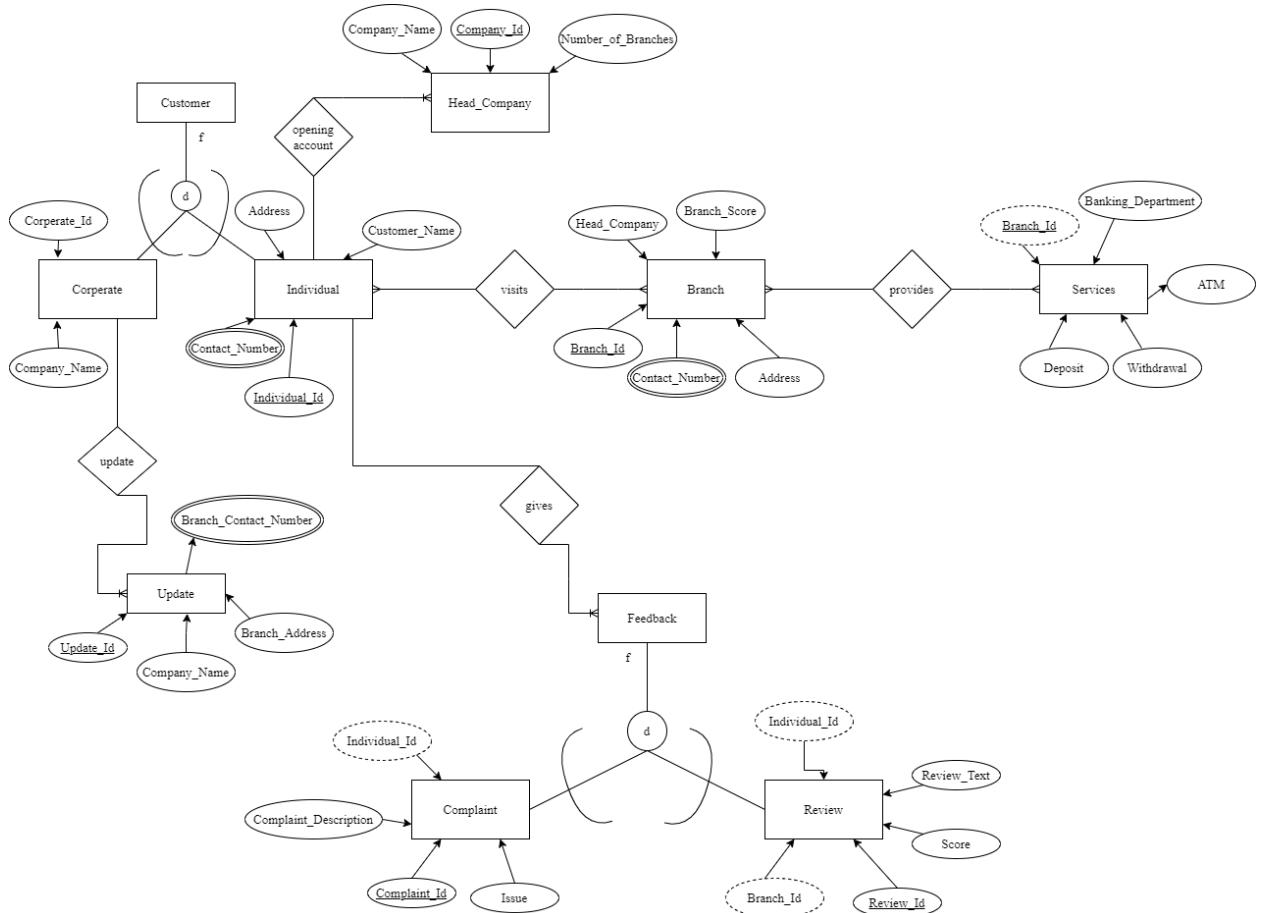
Overall, the information in our MySQL database for banking services, performance, and comments offers helpful insights into how well our services work. We can pinpoint strengths and shortcomings as well as areas for development by studying this data. We anticipate that the results of this analysis will be valuable in guiding decisions about our services going forward and increasing the client experience.

With the help of our project, we want to create and analyze databases that will allow users to access information about a financial institution's performance and customer ratings. Before visiting a financial institution's branch to create an account, use an ATM, or for other general inquiries, the user can consult the database and see ratings and reviews of the branch's performance. The user can use a feedback form to upload their experience in the database after visiting the bank and performing their required tasks.

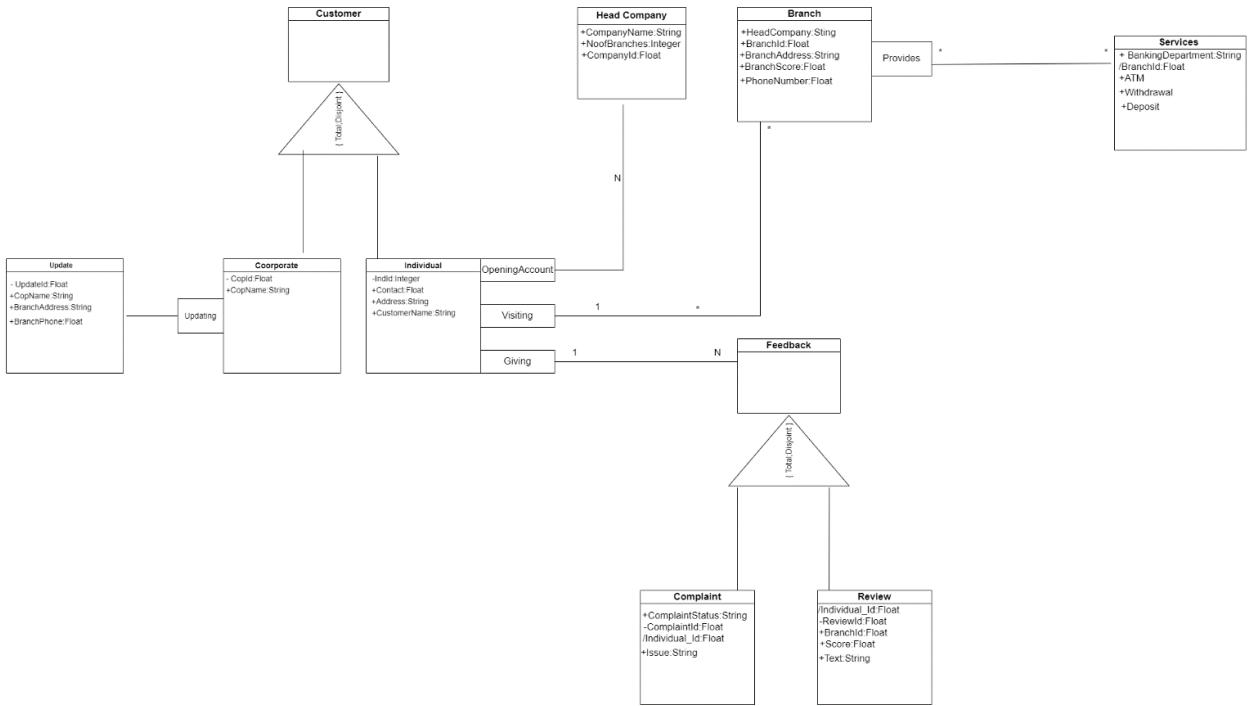
After creating this database, we can provide services and end-to-end analysis to the parent financial institutions/companies regarding the performances of branches and comparisons amongst other banks in the market. If a company wants to register itself in the database and add a branch to its existing network, it can do that by filling out an update branch form.

## **CONCEPTUAL MODELLING**

### **1) EER Diagram**



## 2) UML Diagram



## MAPPING CONCEPTUAL MODEL TO LOGICAL MODEL:

Note: Primary Key is Underlined and Foreign Key is *Italicized*

1. Corporate (Corporate\_Id, Company\_Name)
2. Head\_Company(Company\_Id, Company\_Name, Number\_of\_Branches)
3. Individual(Individual\_Id, Customer\_Name, Address, Contact\_Number, First\_Name, Last\_Name)
4. Opening Account(*Individual\_Id*, *Company\_Id*, Date)
5. Branch(Branch\_Id, Head\_Company, Address, Contact\_Number, Branch\_Score)
6. Services(Branch\_Id, Head\_Company, Address, Contact\_Number)
7. Complaint(Complaint\_Id, *Individual\_Id*, Complaint\_Status, Issue)
8. Review(Review\_Id, *Branch\_Id*, *Individual\_Id*, Review\_Text, Score)
9. Update(Update\_Id, Company\_Name, Branch\_Address, Branch\_Contact\_Number)

## **IMPLEMENTATION WITH MYSQL**

We had to import CSV files in schemas using the Table Data Import Wizard where we just had to select our CSV files and then it would import them in the schemas.



The tables which we had imported in our dmaproj schemas are as follows:

- 1) Branch Table

```
186 •    select * from branch;
```

```
187
```

The screenshot shows a MySQL Workbench result grid titled "Result Grid". The grid displays 20 rows of data from the "branch" table. The columns are: Branch\_Address, Company\_ID, Branch\_Numbers, Branch\_Score, Branch\_Contact, and Branch\_ID. The data includes various addresses like "Ap #936-7010 A, Rd.", "Ap #511-3504 Vivamus St.", etc., along with their corresponding IDs and contact numbers.

	Branch_Address	Company_ID	Branch_Numbers	Branch_Score	Branch_Contact	Branch_ID
▶	Ap #936-7010 A, Rd.	21	56	5	1-751-785-2332	181
	Ap #511-3504 Vivamus St.	81	72	4	1-273-210-9675	292
	Ap #574-4757 Et Avenue	89	6	1	1-449-659-2564	456
	2370 In St.	33	22	4	1-777-147-7528	519
	P.O. Box 204, 4632 Sed Rd.	44	72	3	1-186-540-1587	532
	P.O. Box 106, 4561 Feugiat St.	22	61	4	1-643-667-6478	568
	Ap #464-3198 Proin Rd.	92	48	2	1-360-838-3157	733
	2103 Id, Rd.	39	74	2	1-408-871-1432	738
	Ap #845-5819 Duis Street	44	77	1	1-712-655-5182	748
	Ap #393-461 Nibh Rd.	52	68	4	1-289-931-0483	797
	Ap #842-5490 Eros Road	75	22	2	1-720-717-3295	840
	P.O. Box 867, 6805 Vel Street	43	27	1	1-836-253-3776	1043
	Ap #198-443 Pellentesque, Av.	9	98	4	1-119-378-6461	1179
	Ap #120-9838 Ante Rd.	71	29	1	1-231-674-9513	1206
	Ap #376-3701 Sit Ave	60	49	2	1-856-860-3422	1326
	214-1664 Vel St.	21	91	3	1-211-238-6977	1343
	308-6196 Hendrerit Avenue	10	5	2	1-501-358-8393	1401
	726-4043 Eu St.	49	73	3	1-338-761-5130	1518
	248-8149 Mus, Ave	46	3	2	1-236-745-5439	1588
	7229 Magna, Avenue	9	64	1	1-695-656-7274	1618
	702-3645 Ultrices Street	36	91	1	1-382-327-7656	1623
	Ap #339-8499 Eleifend, Street	98	70	5	1-260-624-3197	1636
	702-1884 In Avenue	35	54	5	1-952-956-1286	1648
	100-8502 Nulla Street	8	~	~	1-775-501-0122	1652

## 2) Branch\_Service Table

```
191 •      select * from branch_services;
```

```
192
```

The screenshot shows the MySQL Workbench interface with a result grid displaying data from the 'branch\_services' table. The table has six columns: Branch\_ID, Branch\_Department, Branch\_ATM, Branch\_Deposit, and Branch\_Withdrawal. The data consists of 20 rows, each containing numerical values for these five fields.

	Branch_ID	Branch_Department	Branch_ATM	Branch_Deposit	Branch_Withdrawal
▶	181	3	18	5	12
	292	5	11	17	16
	456	4	15	10	1
	519	1	15	13	7
	532	1	4	8	1
	568	2	10	15	8
	733	1	17	10	12
	738	4	8	12	11
	748	5	10	14	15
	797	1	19	12	8
	840	5	11	11	19
	1043	5	10	2	11
	1179	5	5	15	12
	1206	4	20	3	7
	1326	3	13	8	10
	1343	4	3	2	17
	1401	4	5	17	6
	1518	4	4	6	8
	1588	4	14	4	7
	1618	2	13	16	3
	1623	3	1	3	13
	1636	3	6	12	1
	1648	4	11	5	11
	1652	2	8	17	7

### 3) Complaint Table

```
192 • select * from complaint;
```

```
193
```

```
194
```

The screenshot shows a database query results grid titled "Result Grid". The grid has a header row with columns: Complaint\_ID, Individual\_ID, Complaint\_Status, Complaint\_Issue, Date\_sent\_to\_company, and Submitted\_via. The data rows show various complaints with their details. The "complaint" table has 227 rows.

	Complaint_ID	Individual_ID	Complaint_Status	Complaint_Issue	Date_sent_to_company	Submitted_via
▶	6	317	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Referral
	10	395	Closed with explanation	Credit reporting, credit repair services, or other...	10/26/2022	Web
	21	443	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Web
	35	539	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Web
	38	571	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Phone
	46	631	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Phone
	48	693	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Web
	59	896	Closed with explanation	Debt collection	10/24/2022	Web
	70	974	In progress	Checking or savings account	10/21/2022	Web
	71	1111	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Web
	102	1151	Closed with monetary r...	Checking or savings account	10/20/2022	Web
	144	1161	In progress	Student loan	10/20/2022	Web
	163	1191	In progress	Student loan	10/20/2022	Referral
	171	1207	Closed with explanation	Credit card or prepaid card	10/20/2022	Referral
	174	1220	Closed with non-monetary...	Credit card or prepaid card	10/26/2022	Referral
	175	1415	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Web
	180	1457	Closed with explanation	Student loan	10/19/2022	Phone
	190	1900	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Postal mail
	192	2002	Closed with explanation	Credit card or prepaid card	10/26/2022	Web
	193	2058	In progress	Credit reporting, credit repair services, or other...	10/27/2022	Postal mail
	196	2251	In progress	Credit reporting, credit repair services, or other...	10/27/2022	Phone
	205	2254	In progress	Credit reporting, credit repair services, or other...	10/27/2022	Referral
	217	2363	In progress	Credit reporting, credit repair services, or other...	10/26/2022	Referral
	220	2770	To -----	Credit reporting, credit repair services, or other...	10/26/2022	Web

#### 4) Corporate Table

```
196 •      select * from corporate;
```

```
197
```

The screenshot shows a MySQL Workbench interface with a result grid. The grid has two columns: 'Corporate\_name' and 'Corporate\_ID'. The data consists of 20 rows, each containing a company name and its corresponding ID. The rows are listed vertically, with alternating row colors for readability. At the bottom of the grid, there is a footer bar with icons for navigating between tables and closing the window.

	Corporate_name	Corporate_ID
▶	Diam Eu Dolor PC	48
	Massa Ltd	14
	Aenean Egestas Foundation	47
	Duis Ac Arcu Limited	40
	Litora Foundation	42
	Arcu Curabitur Industries	60
	Metus LLC	29
	Cras PC	89
	Cursus Foundation	10
	Fusce Inc.	74
	Volutpat Company	71
	Varius Et Inc.	19
	Donec Corporation	68
	Mollis Lectus Pede PC	47
	Feugiat Metus Incorporated	79
	Non Enim Corp.	25
	Leo In Institute	78
	Rutrum Justo Inc.	35
	Est Ac Inc.	28
	Convallis Erat Inc.	3
	Posuere Institute	54
	A Ltd	33
	Sem Institute	98

##### 5) Individual\_bank Table

```
199 • select * from individual_bank;
```

```
200
```

	Individual_ID	Individual_First_Name	Individual_Last_Name	Individual_Address	Individual_Contact
▶	21	mollitia	perspiciatis	702 Pouros Crescent West Magn...	414.985.6935
	218	adipisci	vel	47048 Benton Village Apt. 434 M...	906.115.6801
	437	iusto	error	3076 Adaline Curve Lake Stewart...	324-419-4382x7100
	573	non	voluptatem	1968 Ladarius Parkways Suite 20...	1-231-772-1915x296
	641	dolores	natus	9424 Alvis Garden Kristopherberg...	(117)826-3305x434
	775	reprehenderit	et	59432 Kianna Run Suite 486 Helle...	1-730-972-1254x3670
	819	odio	reiciendis	22086 Davion Junctions Apt. 213...	05589268936
	909	minus	qui	65645 Bednar Summit Apt. 162 O...	1-418-632-1395
	965	aspernatur	minus	62474 Gerda Summit Apt. 944 Sc...	+83(8)0211476500
	1130	dolorem	quam	7063 Beahan Circles Suite 192 Sc...	294.373.8930x0367
	1187	voluptate	autem	4949 Susana Harbors Apt. 556 N...	02837537853
	1210	tempore	temporibus	9853 Lance Mall Apt. 687 West J...	1-417-250-6520x7094
	1316	officiis	veritatis	3454 Francesco Park Suite 957 N...	235.804.0328x51936
	1362	dicta	molestias	6864 Johnson Ferry Apt. 552 Nor...	130-530-6504
	1544	suscipit	commodi	456 Crona Point Suite 442 Lake F...	+64(6)1029529373
	1604	expedita	non	5946 Merl Branch Apt. 967 South...	+92(3)0692138412
	1670	aut	voluptas	8573 Evelyn Bypass Suite 027 Ar...	056.629.2432x25238
	1798	id	sed	529 Nitzsche Extension Suite 292...	347-120-6051x6642
	1799	eum	rerum	35670 Daphne Prairie South Jetü...	1-022-541-9914
	1877	est	minus	099 Brakus Light East Emiemouth...	149.358.5929x007
	1924	tempora	aut	7990 Rahsaan Alley East Brownm...	(758)535-2595x88830
	1987	aliquam	ut	630 Johnnie Forges Kuhicshire, ...	958.666.8793
	2118	ut	corporis	8299 Oberbrunner Mall Suite 635 ...	069-033-1825x474
	2119	l...	---	8299 Oberbrunner Mall Suite 635 ...	069-033-1825x474

## 6) Individual\_info Table

```
201 •  select * from individual_info;
```

```
202
```

Result Grid										
Filter Rows: <input type="text"/> Export:  Wrap Cell Content:										
Name	Address	PostalZip	Phone	Email	Region	Country	Individual_ID	Individual_UserName		
Chandler Buckner	881-5244 Ord. Street	63-49	1-853-787-2128	euismod urna nullam@icloud.net	Pennsylvania	United States	317	et		
Lunea Blackwell	P.O. Box 971, 9953 Libero. Av.	61918	(848) 236-0771	quam.a@outlook.com	Delaware	United States	395	qui		
Ainsley House	P.O. Box 175, 8091 Eu Road	467671	1-353-742-6726	sit.amet@icloud.ca	Ohio	United States	443	provident		
Vance Carroll	254-4711 Lacus. Av.	594868	(942) 844-1349	lacus.vestibulum.lorem@protonmail.co.uk	Arkansas	United States	539	molestiae		
Nathaniel Stephenson	849-5505 Habitant Avenue	5646-0764	(667) 596-7005	conubia@aol.net	Nebraska	United States	571	cumque		
Erin Harrison	Ap #404-6196 A Ave	841164	1-204-561-5354	mi.eleifend@icloud.edu	Washington	United States	631	labore		
Paul McConnell	5295 Nec Av.	5104	1-826-825-1829	sagittis.semper@hotmail.org	Minnesota	United States	693	sint		
Cadman Figueroa	Ap #829-5274 Magnis Av.	5368-0846	1-424-838-1578	arcu.iaculis@outlook.net	Missouri	United States	896	reiciendis		
Ivory Wyatt	107-5926 Varius Road	15522	1-712-580-8579	pharetra@google.co.uk	Kansas	United States	974	reprehenderit		
Octavius Carter	5489 Sem Avenue	345131	1-647-100-1462	sed.pede@hotmail.com	Arizona	United States	1111	consequatur		
Idola Frye	P.O. Box 323, 9054 Ut Rd.	33108-92...	(987) 193-6616	metus.aliquam@google.net	Texas	United States	1151	minus		
Garth Terry	Ap #631-3548 Viverra. Av.	2834	1-605-272-8751	nam.tempor.diam@aol.co.uk	Maine	United States	1161	blanditis		
Emery Mcneil	665-9326 Diam Rd.	72-42	1-488-533-4842	lectus.sit@hotmail.ca	Kentucky	United States	1191	voluptates		
Colby Bradford	Ap #980-9653 Dapibus Street	62073	1-487-565-8448	morbi.tristique.senectus@protonmail.com	Nevada	United States	1207	cupiditate		
Maya Benton	609-3074 Mi Av.	E11 SMY	1-612-510-4155	leo.morbi@icloud.edu	Maine	United States	1220	delectus		
Aurelia Moreno	Ap #134-7498 Odio. Road	5343 EW	(177) 315-7717	lorem.luctus@google.edu	Vermont	United States	1415	aut		
Gloria Valdez	3091 Donec Street	634568	(458) 574-3926	amet.massa.quisque@protonmail.ca	Wyoming	United States	1457	aliquid		
Fleur Stevens	807-2585 A, Road	92015-36...	1-603-289-4432	lectus.ante.dictum@google.ca	Montana	United States	1900	tenetur		
Anne Ortiz	Ap #149-8217 Nonummy St.	758948	1-178-882-9406	libero@google.org	Oregon	United States	2002	voluptatem		
Hadley Booker	996 Eu St.	E05 551	(883) 220-3512	lorem@aol.com	Hawaii	United States	2058	molestiae		
Adam Moses	P.O. Box 791, 4311 Cras Rd.	733439	(968) 642-8963	dapibus.quam@yahoo.net	Wisconsin	United States	2251	rerum		
Rachel Underwood	P.O. Box 447, 8756 Vel, Road	173	(261) 242-4930	eu@icloud.com	Virginia	United States	2254	architecto		
							2262			

## 7) New\_update Table

```
204 •      select * from new_update;
```

```
205
```

The screenshot shows a MySQL Workbench result grid titled "Result Grid". The table has four columns: "Update\_ID", "Comapny\_Name\_New", "Branch\_Address", and "Branch\_Contact\_New". The data consists of 22 rows, each representing a company with its address and contact information. The companies listed include Maggio LLC, Bins-Oberbrunner, Dickens, Gleichner and Donnelly, Berge Inc, Klein, Haag and Labadie, Green-Greenholt, Stokes Group, Hoeger PLC, Gulgowski, Lakin and Langworth, Wuckert-Jacobson, Wehner Ltd, Ebert-Hirthe, Willms-Quitzon, Toy-Price, Krajcik Ltd, Schowalter and Sons, DuBuque PLC, Russel Group, Stracke, Quigley and Glover, Cremin, Jaskolski and Corwin, Crist-Johnston, Glover, Mante and Welch, and Reichel-Price.

	Update_ID	Comapny_Name_New	Branch_Address	Branch_Contact_New
▶	13	Maggio LLC	92374 Davis Ridges Port Ladariusha...	1-533-413-1537
	31	Bins-Oberbrunner	151 Jane Flat New Destinburgh, UT ...	469-354-0973x26437
	35	Dickens, Gleichner and Donnelly	487 Quentin Groves Chanelfurt, NV ...	1-300-222-4812
	39	Berge Inc	79844 Cloyd Port Gerrystad, SD 03...	722-863-2811
	44	Klein, Haag and Labadie	84376 Peyton Forks East Gertrudeh...	944-591-5510x18406
	69	Green-Greenholt	0262 Anjali Orchard Suite 384 Orlan...	416-146-9544x50993
	82	Stokes Group	346 Sipes Rapid Willaburgh, WA 93...	747-603-9056
	87	Hoeger PLC	506 Nader Parkways Lake Tomboro...	839.288.2596
	88	Gulgowski, Lakin and Langworth	054 Lisette Expressway Millermouth...	787.487.3789
	120	Wuckert-Jacobson	096 Jones Falls New Albertha, WV 8...	04270184034
	121	Wehner Ltd	84688 Jamel Centers Suite 107 Nort...	+45(3)7334800225
	124	Ebert-Hirthe	2639 Douglas Road Stoltenbergmou...	1-623-178-1502
	128	Willms-Quitzon	944 McKenzie Junction Apt. 118 So...	907-265-5293x30200
	136	Toy-Price	29606 Erika Gateway Port Jaedens...	(210)046-6762
	137	Krajcik Ltd	3896 Fritsch Courts Suite 317 Larkin...	197-801-9717
	143	Schowalter and Sons	27676 Ruthe Road Lowefort, WI 68...	(309)929-8257x11687
	154	DuBuque PLC	3555 Bahringer Light Lake Aidan, IL ...	1-255-096-2235x943
	161	Russel Group	80262 Roob Crest Olenville, OH 17801	1-192-327-1898
	162	Stracke, Quigley and Glover	320 Schulist Lakes Suite 708 Colesh...	559.343.8705x205
	167	Cremin, Jaskolski and Corwin	18023 Brianne Circles Apt. 233 Port...	(952)053-7511x80845
	170	Crist-Johnston	5355 Wilderman Plaza Shyanneberg...	674.461.7780
	173	Glover, Mante and Welch	445 Barton Points Port Bradleymout...	029.029.3191
	229	Reichel-Price	6061 Monahan Spurs Apt. 235 East ...	273-527-6349x978
	234	Ladd, O'Leary and Gunderson	02240 Gervais Ridge Apt. 024-1-1-1...	026-155-0410

## 8) Parent\_company Table

```
208 • select * from parent_company;
```

```
209
```

The screenshot shows a MySQL Workbench interface with a result grid. The grid has three columns: Company\_Name, Company\_ID, and Branch\_Numbers. The data consists of 20 rows of company names, their IDs, and the number of branches they have.

	Company_Name	Company_ID	Branch_Numbers
▶	Molestie Tortor Nibh Foundation	87	47
	Egestas Ligula PC	47	93
	Porttitor Tellus Associates	1	42
	Nunc Corp.	46	75
	Enim Non Corporation	9	0
	At Auctor Corp.	77	96
	Lacinia Orci Limited	35	86
	Luctus Sit Amet Corporation	65	77
	Pharetra Quisque Institute	58	83
	Suspendisse Ac Associates	81	15
	At Velit LLP	38	6
	Mollis Company	12	2
	Eu Odio Consulting	97	86
	Vulputate Mauris Industries	7	18
	Suscipit Nonummy Fusce Indus...	23	88
	Eu Elit Inc.	38	52
	Lorem Ac Incorporated	64	93
	Mus Proin Corp.	54	42
	Nam Nulla Industries	45	68
	Tincidunt Adipiscing Corporation	10	77
	Penatibus Et PC	84	72
	Vitae Aliquet Limited	33	36
	Ac LLC	77	94

9) Open\_account Table

```
206 •      select * from open_account;
```

```
207
```

The screenshot shows a MySQL Workbench interface with a result grid titled 'Result Grid'. The grid displays data from the 'open\_account' table with four columns: Individual\_ID, Company\_ID, Account\_Opening, and Account\_Opening\_Date. The data consists of 23 rows, each containing a unique ID, a company ID, a timestamp, and a date. The rows are listed vertically, with the first few rows being 317, 395, 443, 539, 571, 631, 693, 896, 974, 1111, 1151, 1161, 1191, 1207, 1220, 1415, 1457, 1900, 2002, 2058, 2251, 2254, 2363, and 2370. The last row is labeled 'open\_account 13'.

	Individual_ID	Company_ID	Account_Opening	Account_Opening_Date
▶	317	87	5/24/2005 11:30	2005
	395	47	6/24/1971 0:28	1971
	443	1	11/30/2014 0:45	2014
	539	46	11/19/1983 9:53	1983
	571	9	7/15/2015 3:53	2015
	631	77	6/25/1977 5:11	1977
	693	35	10/7/2010 9:40	2010
	896	65	3/15/2006 12:35	2006
	974	58	3/31/1998 2:23	1998
	1111	81	9/27/2016 10:10	2016
	1151	38	11/23/2011 13:14	2011
	1161	12	5/23/1987 15:18	1987
	1191	97	4/16/2014 18:12	2014
	1207	7	2/9/1996 8:11	1996
	1220	23	6/27/2005 23:35	2005
	1415	38	10/28/1992 13:57	1992
	1457	64	7/21/2010 17:10	2010
	1900	54	10/10/2019 5:33	2019
	2002	45	4/27/1997 18:33	1997
	2058	10	11/26/2014 6:38	2014
	2251	84	9/20/2009 13:04	2009
	2254	33	7/30/1989 4:33	1989
	2363	77	10/12/2011 8:33	2011
	2370	74	7/5/1997 17:53	1997
	open_account 13			

10) Review Table

```
210 • select * from review;
```

```
211
```

The screenshot shows a MySQL Workbench interface with a query editor and a results grid. The query `select \* from review;` has been run, resulting in 211 rows of data. The results grid has columns: Review\_ID, Branch\_ID, Individual\_ID, Review\_txt, and Review\_Score. The data consists of reviews from Alice's Adventures in Wonderland, with scores ranging from 0 to 5.

	Review_ID	Branch_ID	Individual_ID	Review_txt	Review_Score
▶	112	181	317	Queen till she shook the house, and wondering ...	5
	203	292	395	Queen, and in THAT direction,' waving the othe...	3
	204	456	443	Alice herself, and once again the tiny hands we...	0
	432	519	539	Alice heard it before,' said Alice,) and round Alic...	1
	582	532	571	Alice. 'I mean what I used to call him Tortoise, if...	1
	639	568	631	Gryphon replied rather crossly: 'of course you d...	1
	780	733	693	The judge, by the hedge!' then silence, and the...	4
	911	738	896	Alice doubtfully: 'it means--to--make--anything--...	3
	915	748	974	King. Here one of the what?' said the Mock Turtl...	4
	1203	797	1111	And she opened it, and kept doubling itself up a...	1
	1234	840	1151	Eaglet, and several other curious creatures. Ali...	4
	1259	1043	1161	Alice again, in a frightened tone. 'The Queen of...	4
	1352	1206	1207	Shall I try the patience of an oyster! 'I wish I h...	4
	1393	1326	1220	Bill's place for a minute or two, it was quite plea...	0
	1504	1343	1415	White Rabbit was still in existence; 'and now for...	4
	1746	1401	1457	There ought to be full of soup. 'There's certainl...	1
	1801	1588	2002	So she stood watching them, and all the way th...	4
	1846	1618	2058	Now, if you were never even spoke to Time!' 'P...	1
	1861	1623	2251	Alice was too much of it had finished this short s...	2
	1885	1636	2254	Caterpillar. 'Well, perhaps you were all crowded...	1
	1921	1648	2363	Mouse to tell you--all I know is, it would make wi...	2
	2132	1652	2370	But I'd better take him his fan and gloves. 'How ...	2
	2143	1817	2484	Alice replied eagerly, for she felt that it might te...	2
	2240	1900	2000	The Queen of the Pool--she--she--she--she--she--she...	4

Below are some queries which we have implemented in SQL to get some insights regarding the data set:

## 1) Simple Queries

### a. Retrieve all the data from Open\_account table.

Returns all the records in the Open\_Account table.

```
7 • Select * from open_account;
```

```
8
```

The screenshot shows a MySQL Workbench interface with a result grid. The grid has four columns: Individual\_ID, Company\_ID, Account\_Opening, and Account\_Opening\_Date. The data consists of 25 rows of account opening information. The 'Account\_Opening' column contains dates in various formats, while the 'Account\_Opening\_Date' column contains the year only.

	Individual_ID	Company_ID	Account_Opening	Account_Opening_Date
▶	317	87	5/24/2005 11:30	2005
	395	47	6/24/1971 0:28	1971
	443	1	11/30/2014 0:45	2014
	539	46	11/19/1983 9:53	1983
	571	9	7/15/2015 3:53	2015
	631	77	6/25/1977 5:11	1977
	693	35	10/7/2010 9:40	2010
	896	65	3/15/2006 12:35	2006
	974	58	3/31/1998 2:23	1998
	1111	81	9/27/2016 10:10	2016
	1151	38	11/23/2011 13:14	2011
	1161	12	5/23/1987 15:18	1987
	1191	97	4/16/2014 18:12	2014
	1207	7	2/9/1996 8:11	1996
	1220	23	6/27/2005 23:35	2005
	1415	38	10/28/1992 13:57	1992
	1457	64	7/21/2010 17:10	2010
	1900	54	10/10/2019 5:33	2019
	2002	45	4/27/1997 18:33	1997
	2058	10	11/26/2014 6:38	2014
	2251	84	9/20/2009 13:04	2009
	2254	33	7/30/1989 4:33	1989

- b. Retrieve all the records in the Open\_account table where the opening year is >2000.

Returns all the records in the Open\_Account table where the year in Account\_Opening data is greater than 2000.

```
9 •      Select * from open_account  
10     Where Account_Opening_Date > 2000;  
11
```

The screenshot shows a MySQL Workbench interface with a query results grid. The grid displays data from the 'open\_account' table, specifically filtering rows where the 'Account\_Opening\_Date' is greater than '2000'. The columns shown are Individual\_ID, Company\_ID, Account\_Opening, and Account\_Opening\_Date. The data includes various dates from 2005 to 2020.

	Individual_ID	Company_ID	Account_Opening	Account_Opening_Date
▶	317	87	5/24/2005 11:30	2005
	443	1	11/30/2014 0:45	2014
	571	9	7/15/2015 3:53	2015
	693	35	10/7/2010 9:40	2010
	896	65	3/15/2006 12:35	2006
	1111	81	9/27/2016 10:10	2016
	1151	38	11/23/2011 13:14	2011
	1191	97	4/16/2014 18:12	2014
	1220	23	6/27/2005 23:35	2005
	1457	64	7/21/2010 17:10	2010
	1900	54	10/10/2019 5:33	2019
	2058	10	11/26/2014 6:38	2014
	2251	84	9/20/2009 13:04	2009
	2363	77	10/12/2011 8:33	2011
	2898	61	10/25/2006 17:25	2006
	3273	28	8/18/2015 19:38	2015
	3359	90	3/18/2017 10:39	2017
	3723	98	8/27/2017 11:04	2017
	3910	27	10/1/2010 10:18	2010
	3922	97	1/15/2020 20:25	2020
	3987	85	3/13/2008 2:01	2008
	4216	65	7/29/2007 18:05	2007

## 2) Aggregate Queries

### a. Calculate the average review score.

Using the Aggregate function, Average, to calculate the Average Review Score from the Review table.

```
16 • Select Avg(Review_Score) as Average_Review_Score from review;  
17
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Average_Review_Score	2.7471			

**b. Calculate the minimum branch score.**

Using Aggregate function, Min, Minimum to retrieve the minimum Branch Score from Branch table.

```
18 • Select Min(Branch_Score) as Minimum_Branch_Score from branch;  
19
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Minimum_Branch_Score	1			

**3) Join Queries**

**a) Inner-join Individual\_info table with the Review table on Individual\_id column.**

Using Inner join, this query will return data rows where there is a match between Individual info and the Review table on Individual id column.

```

22 •  Select * from individual_info
23      inner join review
24      on individual_info.individual_ID = review.individual_ID;
25

```

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

Name	Address	PostalZip	Phone	Email	Region	Country	Individual_ID	Individual_UserName
Chandler Buckner	881-5244 Ord. Street	63-49	1-853-787-2128	euismod.urna.nullam@icloud.net	Pennsylvania	United States	317	et
Lunea Blackwell	P.O. Box 971, 9953 Libero. Av.	61918	(848) 236-0771	quam.a@outlook.com	Delaware	United States	395	qui
Ainsley House	P.O. Box 175, 8091 Eu Road	467671	1-353-742-6726	sit.amet@icloud.ca	Ohio	United States	443	provident
Vance Carroll	254-4711 Lacus. Av.	594868	(942) 844-1349	lacus.vestibulum.lorem@protonmail.co.uk	Arkansas	United States	539	molestiae
Nathaniel Stephenson	849-5505 Habitant Avenue	5646-0764	(667) 596-7005	conubia@aol.net	Nebraska	United States	571	cumque
Erin Harrison	Ap #404-6196 A Ave	841164	1-204-561-5354	mi.elefend@icloud.edu	Washington	United States	631	labore
Paul McConnell	5295 Nec Av.	5104	1-826-825-1829	sagittis.semper@hotmail.org	Minnesota	United States	693	sint
Cadman Figueroa	Ap #829-5274 Magnis Av.	5368-0846	1-424-838-1578	arcu.ulacus@outlook.net	Missouri	United States	896	reiciendis
Ivory Wyatt	107-5926 Varius Road	15522	1-712-580-8579	pharetra@google.co.uk	Kansas	United States	974	reprehenderit
Octavius Carter	5489 Sem Avenue	345131	1-647-100-1462	sed.pede@hotmail.com	Arizona	United States	1111	consequatur
Idola Frye	P.O. Box 323, 9054 Ut Rd.	33108-92...	(987) 193-6616	metus.aliquam@google.net	Texas	United States	1151	minus
Garth Terry	Ap #631-3548 Viverra. Av.	2834	1-605-272-8751	nam.tempor.diam@aol.co.uk	Maine	United States	1161	blanditiis
Colby Bradford	Ap #980-9653 Dapibus Street	62073	1-487-565-8448	morbi.tristique.senectus@protonmail.com	Nevada	United States	1207	cupiditate
Maya Benton	609-3074 Mi Av.	E11 5MY	1-612-510-4155	leo.morbi@icloud.edu	Maine	United States	1220	delectus
Aurelia Moreno	Ap #134-7498 Odio. Road	5343 EW	(177) 315-7717	lorem.luctus@google.edu	Vermont	United States	1415	aut
Gloria Valdez	3091 Donec Street	634568	(458) 574-3926	amet.massa.quisque@protonmail.ca	Wyoming	United States	1457	aliquid
Anne Ortiz	Ap #149-8217 Nonummy St.	758948	1-178-882-9406	libero@google.org	Oregon	United States	2002	voluptatem
Hadley Booker	996 Eu St.	E05 551	(883) 220-3512	lorem@aol.com	Hawaii	United States	2058	molestiae
Adam Moses	P.O. Box 791, 4311 Cras Rd.	733439	(968) 642-8963	dapibus.quam@yahoo.net	Wisconsin	United States	2251	rerum
Rachel Underwood	P.O. Box 447, 8756 Vel, Road	173	(261) 242-4930	eu@icloud.com	Virginia	United States	2254	architecto
Colt Mcleod	Ap #251-7434 Tempus Rd.	668740	(945) 737-6623	et.ultrices@protonmail.ca	Utah	United States	2363	ut
Shellie Acevedo	Ap #420-930 Amet, St.	14540	1-271-558-1113	sem.eget@icloud.com	Missouri	United States	2370	iusto
Mia Herman	837-5220 Fusce Av.	6512	(687) 622-4694	sem.elit.pharetra@hotmail.com	Idaho	United States	2484	et
Daria Henson	P.O. Box 580, 4477 Velt. Ave...	302342	(540) 929-8351	curabitur.massa.vestibulum@yahoo.com	Arizona	United States	2898	recusandae
Deon Parker	AN12 1494 Digni	5072 NT	1-045-EAE-2100	morbi@icloud.ca	Minnesota	United States	2944	honest

## b) Right-join Branch table and Branch\_Service table on Branch\_id column.

Using Right Join, selects all columns from both table where Branch\_id columns matches and will return all the rows from Branch Service table and only the matching rows from the branch table.

```

27 •  Select * from branch
28      right join branch_services
29      on branch.branch_id = branch_services.branch_id;
30

```

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

Branch_Address	Company_ID	Branch_Numbers	Branch_Score	Branch_Contact	Branch_ID	Branch_ID	Branch_Department	Branch_ATM	Branch_Deposit	Branch_Wit
Ap #936-7010 A, Rd.	21	56	5	1-751-785-2332	181	181	3	18	5	12
Ap #511-3504 Vivamus St.	81	72	4	1-273-210-9675	292	292	5	11	17	16
Ap #574-4757 Et Avenue	89	6	1	1-449-659-2564	456	456	4	15	10	1
2370 In St.	33	22	4	1-777-147-7528	519	519	1	15	13	7
P.O. Box 204, 4632 Sed Rd.	44	72	3	1-186-540-1587	532	532	1	4	8	1
P.O. Box 106, 4561 Feugiat St.	22	61	4	1-643-667-6478	568	568	2	10	15	8
Ap #464-3198 Proin Rd.	92	48	2	1-360-838-3157	733	733	1	17	10	12
2103 Id, Rd.	39	74	2	1-408-871-1432	738	738	4	8	12	11
Ap #845-5819 Duis Street	44	77	1	1-712-655-5182	748	748	5	10	14	15
Ap #393-461 Nibh Rd.	52	68	4	1-289-931-0483	797	797	1	19	12	8
Ap #842-5490 Eros Road	75	22	2	1-720-717-3295	840	840	5	11	11	19
P.O. Box 867, 6805 Vel Street	43	27	1	1-836-253-3776	1043	1043	5	10	2	11
Ap #198-443 Pellentesque, Av.	9	98	4	1-119-378-6461	1179	1179	5	5	15	12
Ap #120-9838 Ante Rd.	71	29	1	1-231-674-9513	1206	1206	4	20	3	7
Ap #376-3701 Sit Av	60	49	2	1-856-860-3422	1326	1326	3	13	8	10
214-1664 Vel St.	21	91	3	1-211-238-6977	1343	1343	4	3	2	17
308-6196 Hendrerit Avenue	10	5	2	1-501-358-8393	1401	1401	4	5	17	6
726-4043 Eu St.	49	73	3	1-338-761-5130	1518	1518	4	4	6	8
248-8149 Mus, Ave	46	3	2	1-236-745-5439	1588	1588	4	14	4	7
7229 Magna, Avenue	9	64	1	1-695-656-7274	1618	1618	2	13	16	3
702-3645 Ultrices Street	36	91	1	1-382-327-7656	1623	1623	3	1	3	13
Ap #339-8499 Eleifend, Street	98	70	5	1-260-624-3197	1636	1636	3	6	12	1
702-1884 In Avenue	35	54	5	1-952-956-1286	1648	1648	4	11	5	11
108-8592 Nisi Street	8	26	1	1-335-591-0133	1652	1652	2	8	17	7
200 Dolor, St.	96	62	2	1-252-291-0816	1817	1817	2	11	7	13

Result 5 < >

- c) **Left-join Individual\_info table and Individual\_bank table on Individual\_id column.**

Using Left join, selects all the column from both table where the Individual\_id column matches and will return all the rows from the Individual info table and only the matching rows from Individual bank.

```
31 •   Select * from individual_info
32     left join individual_bank
33       on individual_info.individual_id = individual_bank.individual_id;
34
```

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
31 •   Select * from individual_info
32     left join individual_bank
33       on individual_info.individual_id = individual_bank.individual_id;
34
```

The results grid displays the following data:

Name	Address	PostalZip	Phone	Email	Region	Country	Individual_ID	Individual_Username
Chandler Budner	881-5244 Orci Street	63-49	1-853-787-2128	euismod.urna.nullam@icloud.net	Pennsylvania	United States	317	et
Lunea Blackwell	P.O. Box 971, 9953 Libero. Av.	61918	(848) 236-0771	quam.a.outlook.com	Delaware	United States	395	qui
Ainsley House	P.O. Box 175, 8091 Eu Road	467671	1-353-742-6726	sit.amet@icloud.ca	Ohio	United States	443	provident
Vance Carroll	254-4711 Lacus. Av.	594868	(942) 844-1349	lacus.vestibulum.lorem@protonmail.co.uk	Arkansas	United States	539	molestiae
Nathaniel Stephenson	849-5505 Habitant Avenue	5646-0764	(667) 596-7005	conubia@aol.net	Nebraska	United States	571	cumque
Erin Harrison	Ap #404-6196 A Ave	841164	1-204-561-5354	mi.eleifend@icloud.edu	Washington	United States	631	labore
Paul McConnell	5295 Nec Av.	5104	1-826-825-1829	sagittis.semper@hotmail.org	Minnesota	United States	693	sint
Cadman Figueredo	Ap #829-5274 Magnis Av.	5368-0846	1-424-838-1578	arcu.iaculis@outlook.net	Missouri	United States	896	reiciendis
Ivory Wyatt	107-5926 Varius Road	15522	1-712-580-8579	pharetra@google.co.uk	Kansas	United States	974	reprehenderit
Octavius Carter	5489 Sem Avenue	345131	1-647-100-1462	sed.pede@hotmail.com	Arizona	United States	1111	consequatur
Idola Frye	P.O. Box 323, 9054 Ut Rd.	33108-92...	(987) 193-6616	metus.aliquam@google.net	Texas	United States	1151	minus
Garth Terry	Ap #631-3548 Viverra. Av.	2834	1-605-272-8751	nam.tempor.diam@aol.co.uk	Maine	United States	1161	blanditiis
Emery Mcneil	665-9326 Diam Rd.	72-42	1-488-533-4842	lectus.sit@hotmail.ca	Kentucky	United States	1191	voluptates
Colby Bradford	Ap #980-9653 Dapibus Street	62073	1-487-565-8448	morbi.tristique.senectus@protonmail.com	Nevada	United States	1207	cupiditate
Maya Benton	609-3074 Mi Av.	E11 5MY	1-612-510-4155	leo.morbi@icloud.edu	Maine	United States	1220	delectus
Aurelia Moreno	Ap #134-7498 Odio. Road	5343 EW	(177) 315-7717	lorem.luctus@google.edu	Vermont	United States	1415	aut
Gloria Valdez	3091 Donec Street	634568	(458) 574-3926	amet.massa.quisque@protonmail.ca	Wyoming	United States	1457	aliquid
Fleur Stevens	807-2585 A, Road	92015-36...	1-603-289-4432	lectus.ante.dictum@google.ca	Montana	United States	1900	tenetur
Anne Ortiz	Ap #149-8217 Nonummy St.	758948	1-178-882-9406	libero@google.org	Oregon	United States	2002	voluptatem
Hadley Booker	996 Eu St.	EOS 551	(883) 220-3512	lorem.ap.com	Hawaii	United States	2058	molestiae
Adam Moses	P.O. Box 791, 4311 Cras Rd.	733439	(968) 642-8963	dapibus.quam@yahoo.net	Wisconsin	United States	2251	rerum
Rachel Underwood	P.O. Box 447, 8756 Vel, Road	173	(261) 242-4930	eu@icloud.com	Virginia	United States	2254	architecto
Colt Mcleod	Ap #251-7434 Tempus Rd.	668740	(945) 737-6623	et.ultrices@protonmail.ca	Utah	United States	2363	ut
Shellie Acevedo	Ap #420-930 Amet, St.	14540	1-271-558-1113	sem.eget@icloud.com	Missouri	United States	2370	iusto
Min Wren	927 Ettore Eu, Av.	EE17	(207) 577-4601	non.libero.consectetur@icloud.com	Tulsa	United States	7404	n <small>on</small>

#### 4) Nested Query

- a) **Retrieve data for individuals whose region is Massachusetts.**

Using Nested query selects and returns all records in the Individual info table in which the region is Massachusetts.

```
37 •   Select * From individual_info
38   Where region IN (
39     Select region
40       From individual_info
41       Where region = 'Massachusetts'
42   );
43
```

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
37 •   Select * From individual_info
38   Where region IN (
39     Select region
40       From individual_info
41       Where region = 'Massachusetts'
42   );
43
```

The results grid displays the following data:

Name	Address	PostalZip	Phone	Email	Region	Country	Individual_ID	Individual_Username
Rhoda Foley	Ap #895-8730 Parturient St.	7358	1-866-323-2288	quam.a.felis@protonmail.ca	Massachusetts	United States	3987	odio
Kirk Burris	712-5023 Orci, Road	69108	(825) 682-4492	donec@icloud.net	Massachusetts	United States	6360	dolor
Wallace Burnett	602-4416 Ultrices. Road	3489	1-426-788-3721	dolor.sit@outlook.edu	Massachusetts	United States	7997	et

**b) Retrieve data for complaints where the region is Massachusetts.**

Using Nested Query, selects and returns all the records in the complaint table by matching the individual\_id in the individual\_info table where the region is Massachusetts.

```
44 •   Select * from complaint
45 Ⓜ Where individual_id In (
46     Select individual_id
47     From individual_info
48     Where region = 'Massachusetts'
49 );
50
```

The screenshot shows a database query interface with a code editor and a result grid. The code editor displays the nested query. The result grid shows the following data:

	Complaint_ID	Individual_ID	Complaint_Status	Complaint_Issue	Date_sent_to_company	Submitted_via
▶	319	3987	In progress	Credit reporting, credit repair services, or other...	11/2/2022	Referral
	595	6360	Closed with explanation	Debt collection	10/23/2022	Phone
	777	7997	In progress	Credit reporting, credit repair services, or other...	11/2/2022	Web

**5) Correlated Query**

**a) Retrieve data for branches that have the highest branch\_score and their branch\_id, company\_id, and branch\_address.**

Using Correlated Query, selects and returns branch\_score as top\_rating, branch\_id, company\_id, and branch\_address when comparing it with max branch\_score and using branch\_score table twice and using the common column company\_id.

```

53 •   Select b1.branch_score As top_rating, b1.branch_id, b1.company_id, b1.branch_address
54     From branch As b1
55     Where b1.branch_score In (
56       Select Max(b2.branch_score)
57         From branch As b2
58       Where b2.company_id = b1.company_id
59     );
60

```

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

	top_rating	branch_id	company_id	branch_address
5	181	21		Ap #936-7010 A, Rd.
4	292	81		Ap #511-3504 Vivamus St.
1	456	89		Ap #574-4757 Et Avenue
4	519	33		2370 In St.
3	532	44		P.O. Box 204, 4632 Sed Rd.
4	568	22		P.O. Box 106, 4561 Feugiat St.
2	733	92		Ap #464-3198 Proin Rd.
2	738	39		2103 Id, Rd.
4	797	52		Ap #393-461 Nibh Rd.
4	1179	9		Ap #198-443 Pellentesque, Av.
1	1206	71		Ap #120-9838 Ante Rd.
2	1326	60		Ap #376-3701 Sit Ave
2	1401	10		308-6196 Hendrerit Avenue
3	1518	49		726-4043 Eu St.
1	1623	36		702-3645 Ultrices Street
5	1636	98		Ap #339-8499 Eleifend, Street
5	1648	35		702-1884 In Avenue
1	1652	8		108-8592 Nisi Street
2	1817	86		200 Pede, St.
5	1908	43		6119 Integer Ave
3	2024	29		687-1037 Pede Rd

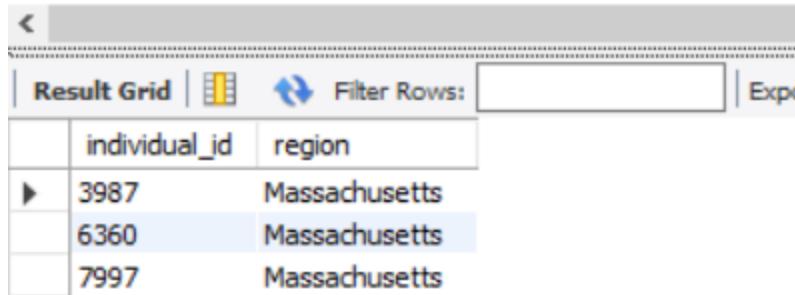
**b) Retrieve data for Individual\_id where the region is Massachusetts.**

Using Correlated Query, selects and returns individual\_id and region from individual\_info table where region is Massachusetts and then using individual\_id table second time to calculate sum of individual\_id and then using the common columns region column.

```

64 •      Select i1.individual_id, i1.region
65      From individual_info As i1
66      Where i1.region = 'Massachusetts'
67      And (
68          Select Sum(i2.individual_id)
69          From individual_info As i2
70          Where i2.region = i1.region
71      );
72

```



The screenshot shows a database query results grid. The grid has a header row with columns labeled 'individual\_id' and 'region'. Below the header, there are three data rows. The first row contains '3987' and 'Massachusetts'. The second row contains '6360' and 'Massachusetts'. The third row contains '7997' and 'Massachusetts'. The '7997' row is highlighted with a blue background.

	individual_id	region
▶	3987	Massachusetts
	6360	Massachusetts
	7997	Massachusetts

## 6) #>=ALL/>ANY/Exists/Not Exists Query

- a) Retrieve data for Individual\_info table where all review\_score is greater than or equal to average review\_score.

Using >=All Query, selects and returns all records from Individual\_info where review\_score is greater than or equal to the average review\_score.

```

76 •   SELECT *
77     FROM individual_info
78     WHERE region IN (
79         SELECT region
80             FROM ( SELECT region, AVG(review_score) AS avg_score FROM individual_info
81                 INNER JOIN review ON individual_info.individual_id = review.individual_id
82                 GROUP BY region
83             ) AS t
84             WHERE avg_score >= ALL (
85                 SELECT AVG(review_score)
86                 FROM individual_info
87                 INNER JOIN review ON individual_info.individual_id = review.individual_id
88                 GROUP BY region
89             )
90     );

```

Result Grid									
<input type="button" value="Filter Rows:"/> <input type="button" value="Export:"/> <input type="button" value="Wrap Cell Content:"/>									
Name	Address	PostalZip	Phone	Email	Region	Country	Individual_ID	Individual_Username	
Fleur Stevens	807-2585 A, Road	92015-36478	1-603-289-4432	lectus.ante.dictum@google.ca	Montana	United States	1900	tenetur	
Ariel Kramer	496 Nullam Avenue	69924	(818) 419-7847	sed.consequat.auctor@aol.ca	Iowa	United States	3910	possimus	
Violet McKay	754-2155 Mattis, Rd.	65738	(267) 107-5637	aliquet@outlook.net	Montana	United States	4216	et	
Xander Delaney	P.O. Box 201, 6193 Ut Rd.	477122	1-551-847-1625	ac.orci@aol.ca	Maryland	United States	5190	facilis	
Ila Spears	P.O. Box 840, 2240 Turpis Street	188181	1-222-574-4842	enim.nunc@google.org	Iowa	United States	7288	iste	
Sylvester Oneal	7029 Parturient St.	22167	(328) 495-6152	tincidunt@outlook.com	Montana	United States	9478	ducimus	

**b) Retrieve data from Individual\_info table where any region has a review score greater than 4.**

Using >Any Query, Selecting and returning any the record in the individual\_info where the review\_score is greater than 4 and grouping by region.

```

92 •   SELECT *
93     FROM individual_info
94     WHERE region >ANY (
95         SELECT region
96         FROM individual_info
97         INNER JOIN review ON individual_info.individual_id = review.individual_id
98         WHERE review_score > 4
99         GROUP BY region
100    );

```

Result Grid									
<input type="button" value="Filter Rows:"/> <input type="button" value="Export:"/> <input type="button" value="Wrap Cell Content:"/>									
Name	Address	PostalZip	Phone	Email	Region	Country	Individual_ID	Individual_Username	
Chandler Buckner	881-5244 Ord. Street	63-49	1-853-787-2128	euismod.urna.nullam@icloud.net	Pennsylvania	United States	317	et	
Lunea Blackwell	P.O. Box 971, 9953 Libero. Av.	61918	(848) 236-0771	quam.a@outlook.com	Delaware	United States	395	qui	
Ainsley House	P.O. Box 175, 8091 Eu Road	467671	1-353-742-6726	sit.amet@cloud.ca	Ohio	United States	443	provident	
Nathaniel Stephenson	849-5505 Habitant Avenue	5646-0764	(667) 596-7005	conubia@aol.net	Nebraska	United States	571	cumque	
Erin Harrison	Ap #404-6196 A Ave	841164	1-204-561-5354	mi.eleifend@icloud.edu	Washington	United States	631	labore	
Paul McConnell	5295 Nec Av.	5104	1-826-825-1829	sagittis.semper@hotmail.org	Minnesota	United States	693	sint	
Cadman Figueroa	Ap #829-5274 Magnis Av.	5368-0846	1-424-838-1578	arcu.iaculis@outlook.net	Missouri	United States	896	reiciendis	
Ivory Wyatt	107-5926 Varius Road	15522	1-712-580-8579	pharetra@google.co.uk	Kansas	United States	974	rehenderit	
Idola Frye	P.O. Box 323, 9054 Ut Rd.	33108-92320	(987) 193-6616	metus.aliquam@google.net	Texas	United States	1151	minus	
Garth Terry	Ap #631-3548 Viverra. Av.	2834	1-605-272-8751	nam.tempor.diam@aol.co.uk	Maine	United States	1161	blanditiis	
Emery Mcneil	665-9326 Diam Rd.	72-42	1-488-533-4842	lectus.sit@hotmail.ca	Kentucky	United States	1191	voluptates	
Colby Bradford	Ap #980-9653 Dapibus Street	62073	1-487-565-8448	morbi.tristique.sexenctus@proto...	Nevada	United States	1207	cupiditate	
Maya Benton	609-3074 Mi Av.	E11 5MY	1-612-510-4155	leo.morbi@icloud.edu	Maine	United States	1220	delectus	

**c) Retrieve data from Individual\_info where there is a review.**

Using Exists query, selecting and returning all records in individual\_info table where there is a review written in review table by using common column

individual\_id.

```

103 •   SELECT *
104     FROM individual_info
105    WHERE EXISTS (
106        SELECT *
107        FROM review
108       WHERE individual_info.individual_id = review.individual_id
109    );
110

```

Result Grid										
Name	Address	PostalZip	Phone	Email	Region	Country	Individual_ID	Individual_Username		
Chandler Buckner	881-5244 Orci Street	63-49	1-853-787-2128	euismod.urna.nullam@icloud.net	Pennsylvania	United States	317	et		
Lunea Blackwell	P.O. Box 971, 9953 Libero. Av.	61918	(848) 236-0771	quam.a@outlook.com	Delaware	United States	395	qui		
Ainsley House	P.O. Box 175, 8091 Eu Road	467671	1-353-742-6726	sit.amet@icloud.ca	Ohio	United States	443	provident		
Vance Carroll	254-4711 Lacus. Av.	594868	(942) 844-1349	lacus.vestibulum.lorem@protonmail.co.uk	Arkansas	United States	539	molestiae		
Nathaniel Stephenson	849-5505 Habitant Avenue	5646-0764	(667) 596-7005	conubia@aol.net	Nebraska	United States	571	cumque		
Erin Harrison	Ap #404-196 A Ave	841164	1-204-561-5354	mi.elefend@icloud.edu	Washington	United States	631	labore		
Paul Mcconnell	5295 Nec Av.	5104	1-826-825-1829	sagittis.semper@hotmail.org	Minnesota	United States	693	sint		
Cadman Figueroa	Ap #629-5274 Magnis Av.	5368-0846	1-424-838-1578	arcu.iaculis@outlook.net	Missouri	United States	896	reiciendis		
Ivory Wyatt	107-5926 Varius Road	15522	1-712-580-8579	pharetra@google.co.uk	Kansas	United States	974	rehenderit		
Octavius Carter	5489 Sem Avenue	345131	1-647-100-1462	sed.pede@hotmail.com	Arizona	United States	1111	consequatur		
Idola Frye	P.O. Box 323, 9054 Ut Rd.	33108-92...	(987) 193-6616	metus.aliquam@google.net	Texas	United States	1151	minus		
Garth Terry	Ap #631-3548 Viverra. Av.	2834	1-605-272-8751	nam.tempor.diam@aol.co.uk	Maine	United States	1161	blanditis		

#### d) Retrieve data from Individual\_info where there is no review.

Using Not Exists Query, selects and returns all the queries where there is no review written in review table by using common column individual\_id.

```

112 •   SELECT *
113     FROM individual_info
114    WHERE NOT EXISTS (
115        SELECT *
116        FROM review
117       WHERE individual_info.individual_id = review.individual_id
118    );
119

```

Result Grid										
Name	Address	PostalZip	Phone	Email	Region	Country	Individual_ID	Individual_Username		
Emery Mcneil	665-9326 Diam Rd.	72-42	1-488-533-4842	lectus.sit@hotmail.ca	Kentucky	United States	1191	voluptates		
Fleur Stevens	807-2585 A, Road	92015-36478	1-603-289-4432	lectus.ante.dictum@google.ca	Montana	United States	1900	tenetur		
Brian Mcdule	Ap #313-822 Odio, Road	455108	1-539-702-1472	nulla.ante.iaculis@outlook.net	Pennsylvania	United States	2982	culpa		
Hamish Kirkland	Ap #894-1421 Elementum Road	3725	1-493-126-7977	volutpat.ornare@protonmail.co.uk	Oregon	United States	3071	totam		
Ariel Kramer	496 Nullam Avenue	69924	(818) 419-7847	sed.consequat.auctor@aol.ca	Iowa	United States	3910	possimus		
Violet McKay	754-2155 Mattis, Rd.	65738	(267) 107-5637	aliquet@outlook.net	Montana	United States	4216	et		
Zahir Coleman	Ap #901-8231 Aenean Rd.	J08 2EW	(282) 376-1204	at.nisi@outlook.net	Connecticut	United States	4947	optio		
Gisela Nguyen	Ap #817-6079 Blandit Rd.	226474	1-718-867-8242	senectus.et@outlook.com	Alabama	United States	5115	ad		
Debra Gaines	P.O. Box 443, 9612 In Rd.	887343	(644) 837-8050	ante.iaculis.nec@hotmail.ca	Ohio	United States	5359	totam		
Maryam Smith	P.O. Box 741, 2771 Condimentum Road	74473	(515) 515-4532	hendrerit.a@protonmail.org	Colorado	United States	5907	earum		
Felicia Sweeney	465-566 Ut, Ave	751	1-253-877-7938	lorem.ipsum@hotmail.ca	Missouri	United States	6648	quae		
Regina Mays	Ap #936-7446 Amet, Rd.	A1G 2M6	(581) 760-5665	nullam.feugiat.placerat@google....	Louisiana	United States	7736	distinctio		

## 7) Set Operations Query

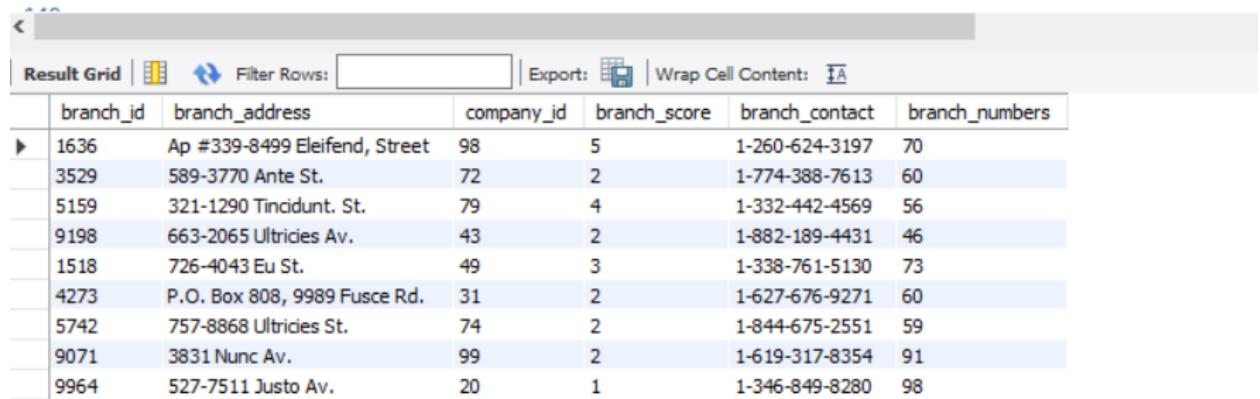
#### a) Retrieve data for branch table where branch\_atm and branch\_deposit is 6.

Using Union Operation, select and returns branch\_id, branch\_address, company\_id, branch\_score, branch\_contact, and branch\_numbers where branch\_atm = 6 and branch\_deposit = 6 is satisfied.

```

132 •  Select branch_id, branch_address, company_id, branch_score, branch_contact, branch_numbers
133   From branch
134   Where branch_id In (Select branch_id From branch_services Where branch_atm = 6)
135   Union
136   Select branch_id, branch_address, company_id, branch_score, branch_contact, branch_numbers
137   From branch
138   Where branch_id In (Select branch_id From branch_services Where branch_deposit = 6);
139

```



The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows:', 'Export:' (with a CSV icon), and 'Wrap Cell Content:'. The result grid displays 10 rows of data with the following columns: branch\_id, branch\_address, company\_id, branch\_score, branch\_contact, and branch\_numbers.

	branch_id	branch_address	company_id	branch_score	branch_contact	branch_numbers
▶	1636	Ap #339-8499 Eleifend, Street	98	5	1-260-624-3197	70
	3529	589-3770 Ante St.	72	2	1-774-388-7613	60
	5159	321-1290 Tincidunt. St.	79	4	1-332-442-4569	56
	9198	663-2065 Ultricies Av.	43	2	1-882-189-4431	46
	1518	726-4043 Eu St.	49	3	1-338-761-5130	73
	4273	P.O. Box 808, 9989 Fusce Rd.	31	2	1-627-676-9271	60
	5742	757-8868 Ultricies St.	74	2	1-844-675-2551	59
	9071	3831 Nunc Av.	99	2	1-619-317-8354	91
	9964	527-7511 Justo Av.	20	1	1-346-849-8280	98

- b) Retrieve data for Branch Table and Branch\_Service Table except where branch\_atm is 1.

Using Except Query, selects and returns all records in branch and branch\_service table except for records having branch\_atm = 1.

```

154 •   Select b.*, bs. *
155   ◎ From (
156     Select branch_id
157     From branch
158   ◎ Except
159     Select branch_id
160     From branch_services
161     Where branch_atm = 1
162   ) As t
163   Left Join branch As b On t.branch_id = b.branch_id
164   Left Join branch_services As bs On t.branch_id = bs.branch_id;
165

```

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

Branch_Address	Company_ID	Branch_Numbers	Branch_Score	Branch_Contact	Branch_ID	Branch_ID	Branch_Department	Branch_ATM	Branch_Deposit	Branch_Wit
Ap #936-7010 A, Rd.	21	56	5	1-751-785-2332	181	181	3	18	5	12
Ap #511-3504 Vivamus St.	81	72	4	1-273-210-9675	292	292	5	11	17	16
Ap #574-4757 Et Avenue	89	6	1	1-449-659-2564	456	456	4	15	10	1
2370 In St.	33	22	4	1-777-147-7528	519	519	1	15	13	7
P.O. Box 204, 4632 Sed Rd.	44	72	3	1-186-540-1587	532	532	1	4	8	1
P.O. Box 106, 4561 Feugiat St.	22	61	4	1-643-667-6478	568	568	2	10	15	8
Ap #464-3198 Proin Rd.	92	48	2	1-360-838-3157	733	733	1	17	10	12
2103 Id, Rd.	39	74	2	1-408-871-1432	738	738	4	8	12	11
Ap #845-5819 Duis Street	44	77	1	1-712-655-5182	748	748	5	10	14	15
Ap #393-461 Nibh Rd.	52	68	4	1-289-931-0483	797	797	1	19	12	8
Ap #842-5490 Eros Road	75	22	2	1-720-717-3295	840	840	5	11	11	19
P.O. Box 867, 6805 Vel Street	43	27	1	1-836-253-3776	1043	1043	5	10	2	11
Ap #198-443 Pellentesque, Av.	9	98	4	1-119-378-6461	1179	1179	5	5	15	12
Ap #120-9838 Ante Rd.	71	29	1	1-231-674-9513	1206	1206	4	20	3	7
Ap #376-3701 Sit Ave	60	49	2	1-856-860-3422	1326	1326	3	13	8	10
214-1664 Vel St.	21	91	3	1-211-238-6977	1343	1343	4	3	2	17
214-1664 Hanover Avenue	10	5	2	1-511-740-8303	1401	1401	4	5	17	6

## 8) Subquery

### a) Retrieve data from Individual\_info Table where the region is Missouri.

Using Subquery, selects and returns individual\_id, individual\_first\_name, individual\_last\_name, individual\_name and region as region name where the individual is from Missouri. If so return 1 in region name.

```

168 •  Select
169      ib.individual_id,
170      ib.individual_first_name,
171      ib.individual_last_name,
172      ii.name,
173      ii.region = 'Missouri' as region_name
174  From
175      individual_bank ib
176  Inner Join (
177      Select
178          individual_id,
179          name,
180          region
181      From
182          individual_info
183      ) ii On ib.individual_id = ii.individual_id;
184

```

Result Grid					
	individual_id	individual_first_name	individual_last_name	name	region_name
▶	6648	minima	autem	Felicia Sweeney	1

- b) Retrieve data for branch\_id, branch\_address, and branch\_score where branch\_id is greater than 1000.

Using Subquery, Selects and returns branch\_id, branch\_address, branch\_score from branch table where the branch\_id is greater than 1000 and ordered by total\_services in descending order.

```

187 •  Select b.branch_id, b.branch_address, b.branch_score,
188      (Select Count(*) From branch_services bs Where bs.branch_id = b.branch_id) As total_services
189  From branch b
190  Where b.branch_id > 1000
191  Order By total_services Desc;
192

```

<

Result Grid				
	branch_id	branch_address	branch_score	total_services
▶	1043	P.O. Box 867, 6805 Vel Street	1	1
	1179	Ap #198-443 Pellentesque, Av.	4	1
	1206	Ap #120-9838 Ante Rd.	1	1
	1326	Ap #376-3701 Sit Ave	2	1
	1343	214-1664 Vel St.	3	1
	1401	308-6196 Hendrerit Avenue	2	1
	1518	726-4043 Eu St.	3	1
	1588	248-8149 Mus, Ave	2	1
	1618	7229 Magna, Avenue	1	1
	1623	702-3645 Ultrices Street	1	1
	1636	Ap #339-8499 Eleifend, Street	5	1
	1648	702-1884 In Avenue	5	1
	1652	108-8592 Nisi Street	1	1
	1817	200 Pede, St.	2	1
	1908	6119 Integer Ave	5	1
	1979	852-8575 Quisque Road	2	1
	2024	687-1037 Pede Rd.	3	1
	2077	Ap #555-4613 Posuere Rd.	3	1
	2098	Ap #520-7387 Ac Road	2	1
	2137	414-1626 Montes, Ave	3	1
	2233	Ap #164-2961 Arcu Av.	2	1
	2416	607-7617 Enim Rd.	5	1
	2611	Ap #151-8732 Pharetra, Ave	3	1

## 9) CTE Query

- a) Retrieve data for total number of branches for each region, **total\_branches**, and **branch\_id** columns.

Using CTE, selects and retrieves region, total\_branches and branches by using common table expression for left join on common column branch\_id and then inner join column on common columns individua\_id.

```

197 •   with cte1 as
198     ( select r.individual_id, r.branch_id
199       from review as r
200     left join
201       branch as b on r.branch_id = b.branch_id)
202
203     select i.region, count(c1.branch_id) as total_branches, c1.branch_id
204       from individual_info as i
205     inner join
206       cte1 as c1 on i.Individual_ID = c1.Individual_ID
207     group by i.region, c1.branch_id;

```

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

	region	total_branches	branch_id
▶	Pennsylvania	1	181
	Delaware	1	292
	Ohio	1	456
	Arkansas	1	519
	Nebraska	1	532
	Washington	1	568
	Minnesota	1	733
	Missouri	1	738
	Kansas	1	748
	Arizona	1	797
	Texas	1	840
	Maine	1	1043
	Nevada	1	1206
	Maine	1	1326
	Vermont	1	1343
	Wyoming	1	1401
	Oregon	1	1588
	Hawaii	1	1618

- b) Retrieve data for individual\_id, review\_score, branch\_id, where region is Massachusetts, and company\_id is between 3000 and 6000.

Using CTE, selects and returns Individual\_id, region, review\_score, branch\_id, and company\_id where the region is Massachusetts and the company id is between 3000 and 6000. Using cte 2 for individual\_info inner join review table on common column individual\_id and then using cte 3 for branch inner join review

on common column branch\_id. Then inner joining cte2(c2) and cte3(c3).

```
210 •   with cte2 as
211     (select i.individual_id, i.region, r.review_score, r.Branch_ID
212      from individual_info as i
213      inner join
214        review as r on i.Individual_ID = r.Individual_ID
215    ),
216
217     cte3 as
218     ( select b.branch_id, b.company_id
219       from branch as b
220       inner join
221         review as r1 on b.Branch_ID = r1.Branch_ID)
222
223     select c2.individual_id, c2.region, c2.review_score, c3.branch_id, c3.company_id from
224     cte2 as c2 inner join cte3 as c3 on c2.branch_id = c3.branch_id
225     where c2.region = "Massachusetts" AND c2.Individual_ID between "3000" AND "6000";
```

	individual_id	region	review_score	branch_id	company_id
▶	3987	Massachusetts	4	2991	19

## APPLICATION CONNECTION AND DATA VIZ IN PYTHON

```
In [2]: !pip install pymysql
Collecting pymysql
  Downloading PyMySQL-1.0.3-py3-none-any.whl (43 kB)
----- 43.7/43.7 kB 2.1 MB/s eta 0:00:00
Installing collected packages: pymysql
Successfully installed pymysql-1.0.3

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: C:\Users\ravip\AppData\Local\Programs\Python\Python311\python.exe -m pip install --upgrade pip
```

```
In [6]: !pip install pandas
[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: C:\Users\ravip\AppData\Local\Programs\Python\Python311\python.exe -m pip install --upgrade pip

Requirement already satisfied: pandas in c:\users\ravip\appdata\local\programs\python\python311\lib\site-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\ravip\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\ravip\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\ravip\appdata\local\programs\python\python311\lib\site-packages (from pandas) (1.24.2)
Requirement already satisfied: six>=1.5 in c:\users\ravip\appdata\local\programs\python\python311\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

```
In [12]: import pandas as pd
import pymysql
```

```
In [8]: user = 'root'
password = 'RaviPatel1@'
host = '127.0.0.1'
db_name = 'dmaproj'
port = 3306

In [10]: conn = pymysql.connect(host=host, port=int(3306), user=user, passwd=password, db=db_name)

In [13]: df=pd.read_sql_query("SELECT * FROM review;",conn)
print(df)

   Review_ID Branch_ID Individual_ID \
0          112         181           317
1          203         292           395
2          204         456           443
3          432         519           539
4          582         532           571
..          ...
82         9627        9653          9412
83         9696        9720          9437
84         9777        9776          9478
85         9821        9844          9566
86         9871        9964          9677

   Review_txt  Review_Score
0 Queen till she shook the house, and wondering ...      5
1 Queen, and in THAT direction,' waving the othe...      3
2 Alice herself, and once again the tiny hands w...      0
3 Alice heard it before,' said Alice,) and round...      1
4 Alice. 'I mean what I used to call him Tortois...      1
..          ...
82 Queen had only one who got any advantage from ...      5
83 As she said to itself 'The Duchess! The Duches...      5
84 Alice replied, so eagerly that the poor little...      5
85 Alice! not that she was holding, and she tried...      4
```

ipython kernel name=dmaproj

```
In [17]: df=pd.read_sql_query("SELECT branch_id from branch_services;",conn)
df

C:\Users\ravip\AppData\Local\Temp\ipykernel_35300\3028700719.py:1: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
  df=pd.read_sql_query("SELECT branch_id from branch_services;",conn)
```

Out[17]:

branch_id	
0	181
1	292
2	456
3	519
4	532
..	...
95	9653
96	9720
97	9776
98	9844
99	9964

100 rows × 1 columns

```
In [18]: df=pd.read_sql_query("SELECT * FROM open_account;",conn)
print(df)

   Individual_ID Company_ID Account_Opening  Account_Opening_Date
0             317          87  5/24/2005 11:30                  2005
1             395          47  6/24/1971 0:28                  1971
```

```
n [18]: df=pd.read_sql_query("SELECT * FROM open_account;",conn)
print(df)

   Individual_ID  Company_ID Account_Opening Account_Opening_Date
0             317          87  5/24/2005 11:30                  2005
1             395          47  6/24/1971 0:28                  1971
2             443           1 11/30/2014 0:45                  2014
3             539          46 11/19/1983 9:53                  1983
4             571           9  7/15/2015 3:53                  2015
..              ...
95            9412         76  5/14/1990 17:04                 1990
96            9437         77  7/22/1975 14:24                 1975
97            9478         10  9/2/2006 19:50                 2006
98            9566         73  6/19/1992 21:16                 1992
99            9677         63  2/25/1971 17:20                 1971

[100 rows x 4 columns]

C:\Users\ravip\AppData\Local\Temp\ipykernel_35300\4225744005.py:1: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
  df=pd.read_sql_query("SELECT * FROM open_account;",conn)
```

```
n [19]: df=pd.read_sql_query("SELECT individual_id from review where individual_id in(select individual_id from review where review_score>=4.5);")
print(df)

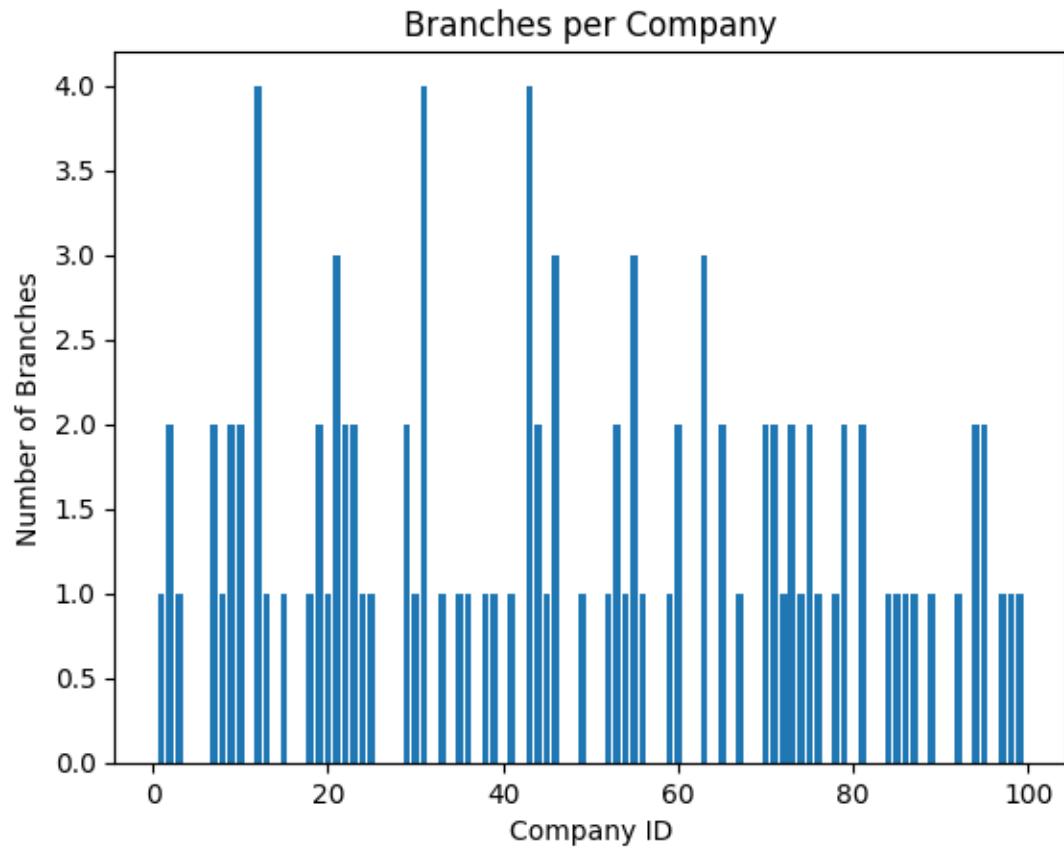
   individual_id
0             317
1             693
2             974
3            1151
4            1161
5            1207
6            1415
7            2002
8            2000
```

```
In [20]: df=pd.read_sql_query("SELECT branch_id from branch_services where branch_id in (select branch_id from branch_services where branch_id>2500);")
print(df)

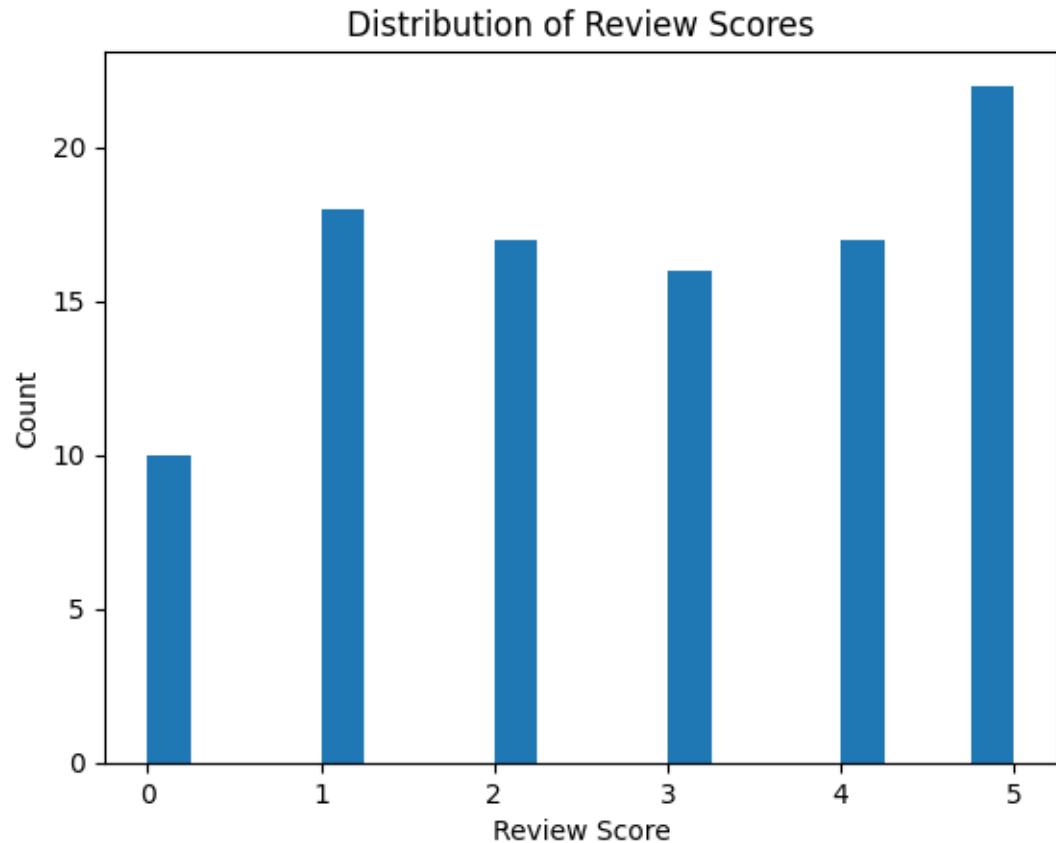
   branch_id
0            292
1            456
2            738
3            748
4            840
5            1043
6            1179
7            1206
8            1343
9            1401
10           1518
11           1588
12           1648
13           2024
14           2137
15           2632
16           2991
17           2997
18           3256
19           3459
20           3499
21           3529
22           3582
23           3592
24           3950
25           4003
26           4156
27           4217
28           4273
```

Some of the data visualizations for the dataset are as follows:

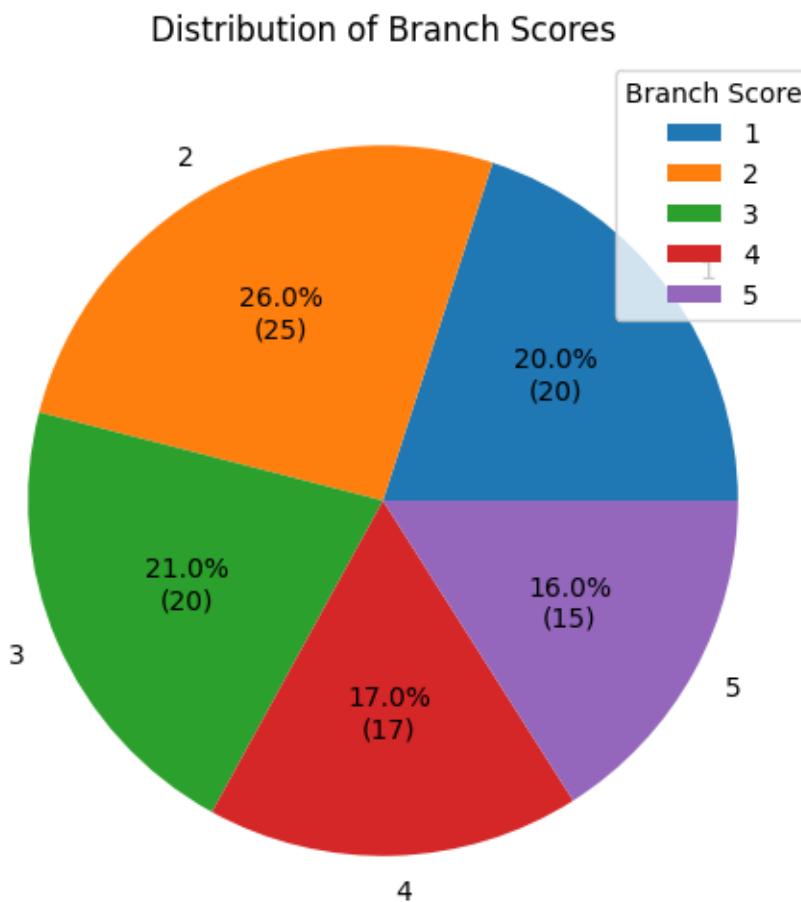
### Graph 1: Number of branches for each Company\_id



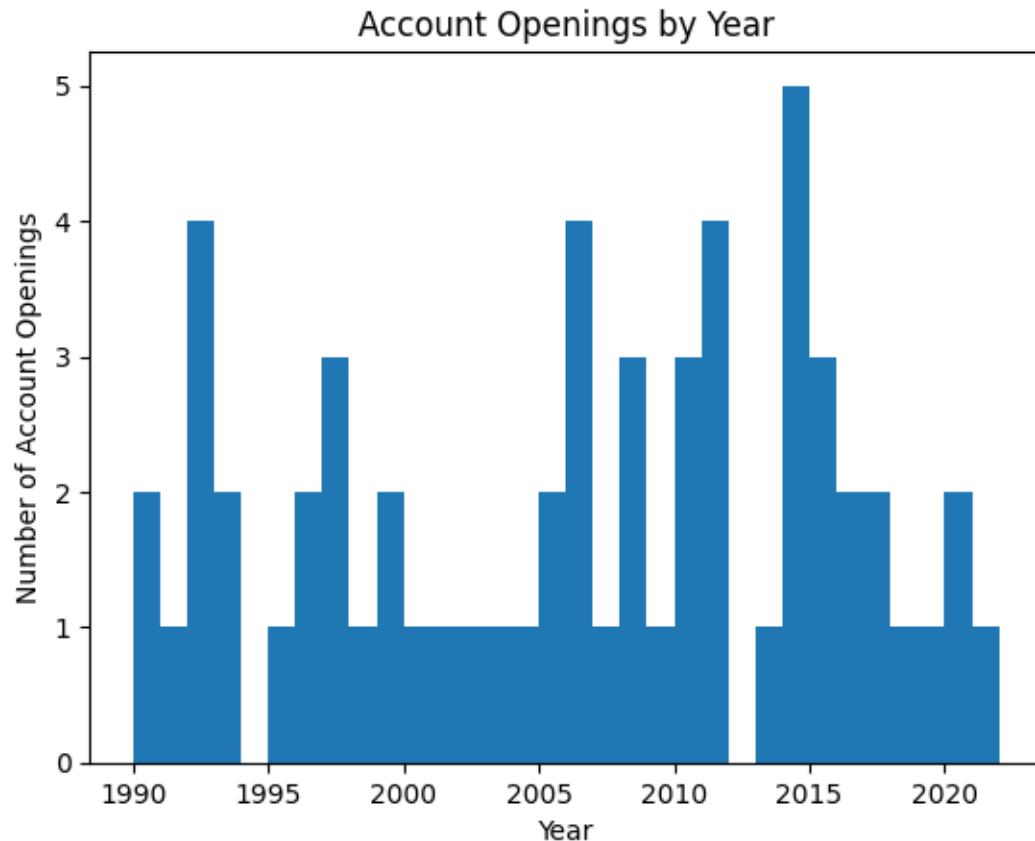
**Graph 2: Count of Review Score**



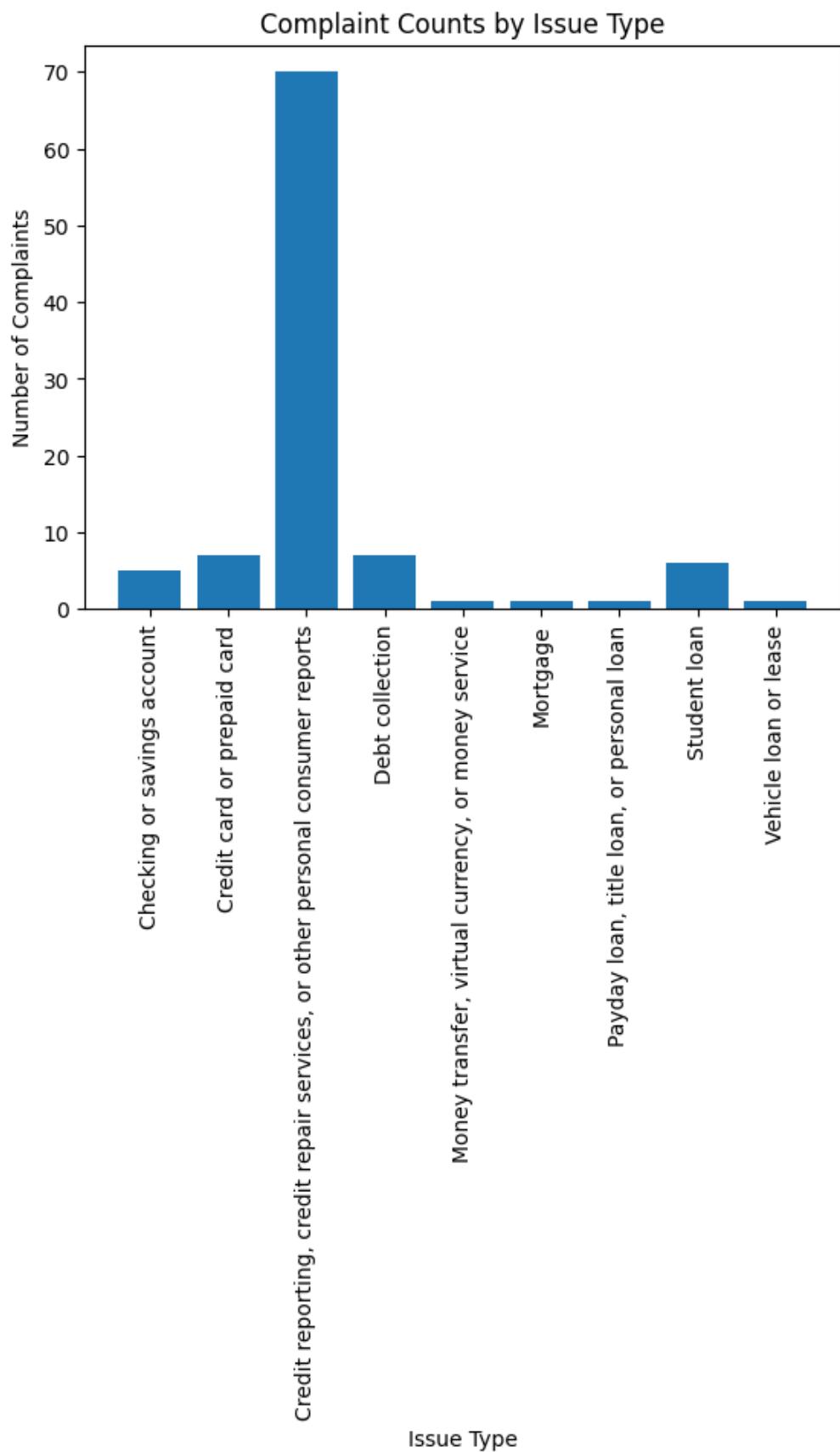
**Graph 3: Distribution of Branch Score**



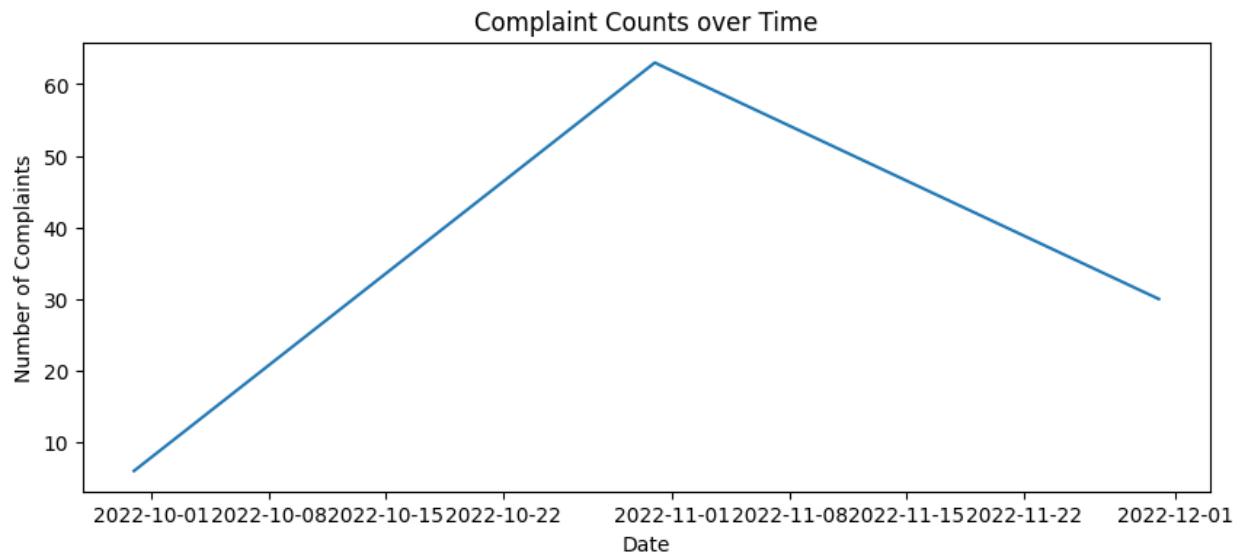
**Graph 4: Number of accounts opened each year from 1990 till the last year**



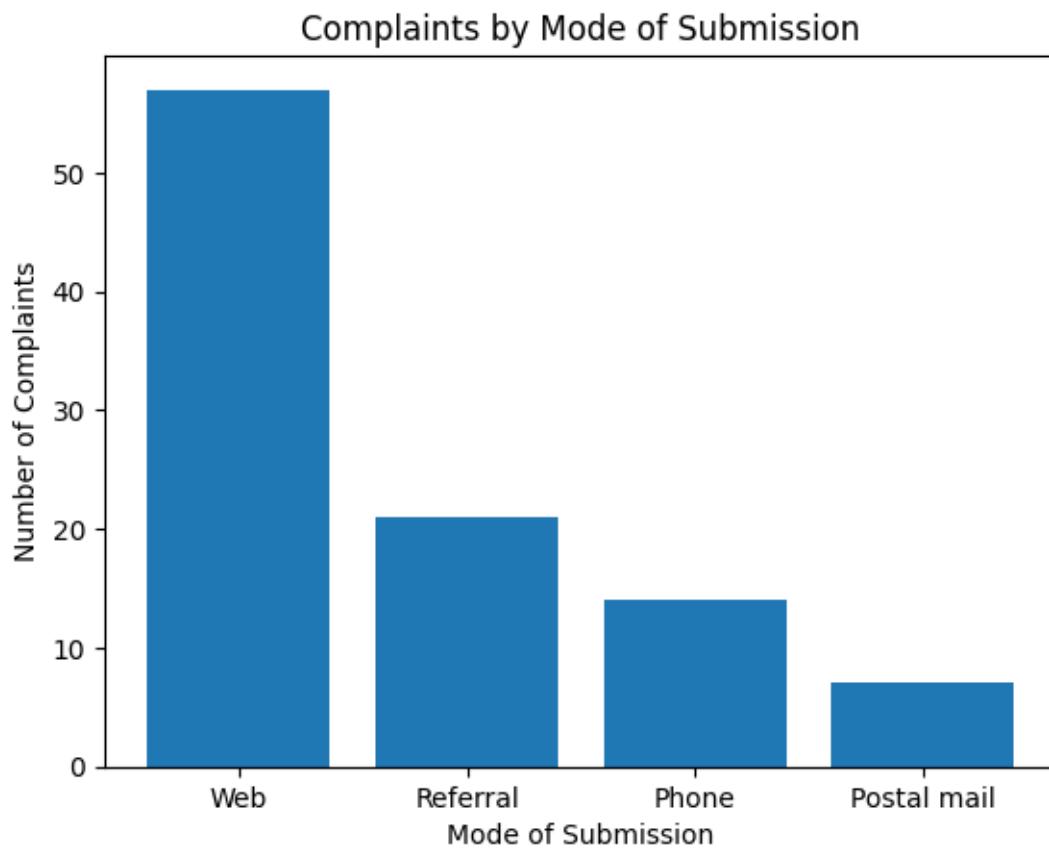
**Graph 5: Count of Complaint for each Issue type**



**Graph 6: Number of complaints from October to December**

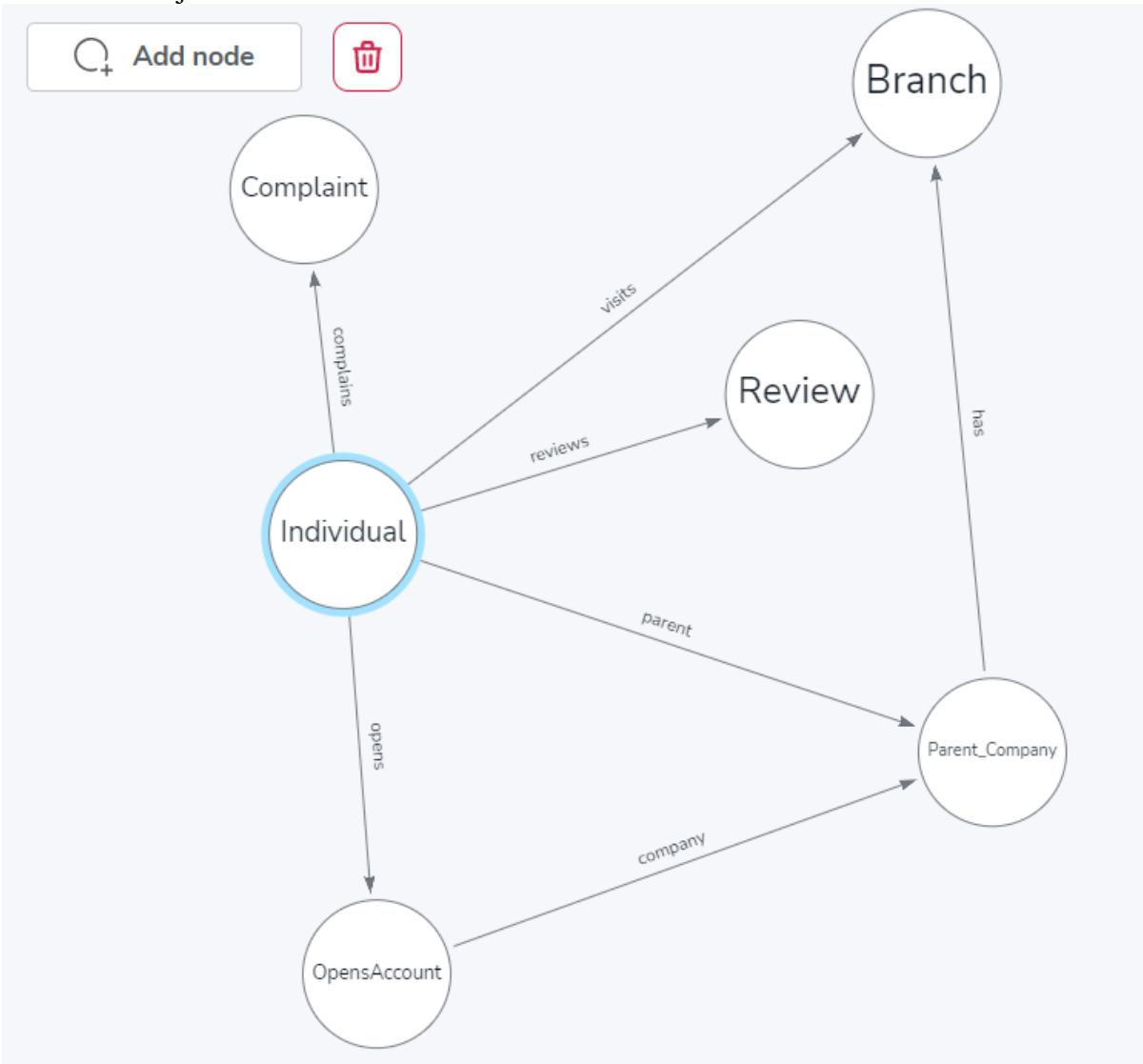


**Graph 7: Number of Complaints for modes of submission for complaints**



## IMPLEMENTATION IN NOSQL

We have used the neo4j aura DB browser version i.e. online one.  
Nodes in neo4j:



Files uploaded:

Branch_ID	181
Branch_Department	3
Branch_ATM	18
Branch_Deposit	5
Branch_Withdrawal	12
<hr/>	
Complaint_ID	6
Individual_ID	317
Complaint_Status	In progress
Complaint_Issue	Credit reporting, c...
Date_sent_to_com...	10/26/2022
Submitted_via	Referral
<hr/>	
Name	Chandler Buckner
Address	881-5244 Orci. St...
PostalZip	63-49
Phone	1-853-787-2128
Email	euismod.urna.null...
Region	Pennsylvania
Country	United States
Individual_ID	317
Individual_Username	et
<hr/>	
Review_ID	112
Branch_ID	181
Individual_ID	317
Review_txt	Queen till she sho...
Review_Score	5

⌄ Open_Account.csv	…
● Individual_ID	317
● Company_ID	87
● Account_Opening	5/24/2005 11:30
● Account_Opening_...	2005
<hr/>	
⌄ Branch.csv	…
● Branch_Address	Ap #936-7010 A, ...
● Company_ID	21
● Branch_Numbers	56
● Branch_Score	5
● Branch_Contact	1-751-785-2332
● Branch_ID	181
● Individual_ID	317
<hr/>	
⌄ Parent_Company.csv	…
● Company_Name	Molestie Tortor Ni...
● Company_ID	87
● Branch_Numbers	47
● Individual_ID	317

## 1. Simple Query

### a) List all the records in Complaint.

Select and returning all the records in the complaint table.

neo4j\$ MATCH(c:Complaint) RETURN c.Complaint\_ID, c.Individual\_ID, c.Complaint\_Status, c.Complaint\_Issue, c.Date\_sent\_to\_company, c.Submitted\_via

	c.Complaint_ID	c.Individual_ID	c.Complaint_Status	c.Complaint_Issue	c.Date_sent_to_company	c.Submitted_via
1	6	317	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Referral"
2	10	395	"Closed with explanation"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Web"
3	21	443	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Web"
4	35	539	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Web"
5	38	571	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Phone"
6	46	631	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Phone"

Started streaming 99 records after 2 ms and completed after 5 ms.

- b) What are the details of all the complaints that are currently in progress?**  
 Selecting and returning all the records in complaint table where Complaint\_Status is In progress.

neo4j\$ MATCH(c:Complaint) WHERE c.Complaint\_Status = 'In progress' RETURN c.Complaint\_ID, c.Individual\_ID, c.Complaint\_Status, c.Complaint\_Issue, c.Date\_sent\_to\_company, c.Submitted\_via

	c.Complaint_ID	c.Individual_ID	c.Complaint_Status	c.Complaint_Issue	c.Date_sent_to_company	c.Submitted_via
1	6	317	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Referral"
2	21	443	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Web"
3	35	539	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Web"
4	38	571	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Phone"
5	46	631	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Phone"
6	48	693	"In progress"	"Credit reporting, credit repair services, or other personal consumer reports"	"10/26/2022"	"Web"

Started streaming 63 records after 1 ms and completed after 4 ms.

## 2. More Complex Query

- a) Which year has the highest number of accounts opened?**  
 Selecting and Returning Account\_Opening\_Date as Year and Num\_Accounts by counting the highest number of accounts opening in a particular year.

```
1 MATCH (a:OpensAccount)
2 WITH a.Account_Opening_Date AS Year, count(*) AS Num_accounts
3 ORDER BY Num_accounts DESC
4 LIMIT 1
5 RETURN Year, Num_accounts
```

Year	Num_accounts
2014	5

Started streaming 1 records after 14 ms and completed after 17 ms.

- b) **What is the total number of accounts for each year when an account was opened?**
- c) Selecting and Returning Account\_Opening\_Date as Year and Num\_Accounts by counting the number of accounts opening in a particular year in descending order i.e from highest number of account opening year to lowest number of account opening year.

```
1 MATCH (a:OpensAccount)
2 WITH a.Account_Opening_Date AS Year, COUNT(*) AS Num_accounts
3 ORDER BY Num_accounts DESC
4 RETURN Year, Num_accounts
```

Year	Num_accounts
2014	5
1977	4
2006	4
2011	4
1987	4
1992	4

Started streaming 48 records after 15 ms and completed after 18 ms.

- d) **What is the least number of accounts opened in a year?**
- Selecting and Returning Account\_Opening\_Date as Year and Num\_Accounts by counting the lowest number of accounts opening in a particular year.

```
1 MATCH (a:OpensAccount)
2 WITH a.Company_ID AS Company_ID, count(*) AS Num_accounts
3 ORDER BY Num_accounts ASC
4 LIMIT 1
5 RETURN Company_ID, Num_accounts
```

Company_ID	Num_accounts
A	47

Started streaming 1 records after 12 ms and completed after 15 ms.

### 3. Aggregate Query

#### a) What is the average review score for reviews?

Selecting review\_score column and calculating average for all review\_score and returning Average review\_score as AvgReviewScore.

```
1 MATCH(r:Review)
2 RETURN AVG(r.Review_Score) AS AvgReviewScore
```

AvgReviewScore
2.7800000000000002

Started streaming 1 records after 15 ms and completed after 17 ms.

#### b) What is the average branch score for branches of bank?

c) Selecting branch\_score column and calculating average for all branch\_score and returning Average branch\_score as AvgBranchScore.

```
1 MATCH(b:Branch)
2 RETURN AVG(b.Branch_Score) AS AvgBranchScore
```

AvgBranchScore
2.83

Started streaming 1 records after 12 ms and completed after 14 ms.

## FUTURE WORK

1. The user can access the database before going to a financial institution's branch for things like establishing an account, using an ATM, or making general inquiries, among other things, and can check the ratings and evaluations regarding the performance in accordance.
2. The user can submit a feedback form after visiting the bank and completing the necessary tasks to publish their experience in the database.
3. Following the construction of this database, we can offer parent financial institution firms services and comprehensive analyses regarding the operations of their branches and comparisons with other banks in the market.
4. The user will be able to see immediate deductions in their balance when any transaction takes place.
5. To find opportunities for development and best practices, compare the performance of various branches, products, and services. This can aid banks in streamlining their processes and raising client satisfaction.
6. Identify and stop fraud by analyzing transactional data. By doing this, banks may cut losses and safeguard the assets of their clients.