



Information and communication technologies

Final Project

Project title: Bank Account Management System

Prepared by: Maryam Bayzhigitova
Moldir Kumarbek

Checked by: Ruslan Omirgaliyev

Nur-Sultan, 2020

Table of Contents

Chapter 1

1.1. Introduction.....	2
1.2. The purpose of the Bank Account Management System.....	3
1.3. Crow's Foot notation.....	4
1.4. Database.....	5

Chapter 2

2.1. DDL Statements	7
2.2. DML Statements	8
2.3. Queries.....	21
2.4. Subqueries.....	25

Appendix A.....	29
Appendix B.....	31
Appendix C.....	35

Conclusion.....	37
Reflection.....	38

List of Figures

1. ERD Diagram of Bank Account Management System.....	3
3. Result of ALTER TABLE statements.....	6
4.1. Results of INSERT INTO statement	7
5.3. Results of UPDATE statement.....	12
6.5. Results of DELETE statement.....	17
7.6. Results of query statements.....	21
9.1. Results of subquery statements.....	25

Chapter 1

1.1. Introduction

The “Bank Account Management System” project is a model of Banking Management System. This model shows how Bank Account Management System enables the clients to perform the essential keeping money exchanges and other operations. The system provides the access to the client to form an account, deposit/withdraw money from his account and also to see reports of all accounts present. In this extend, we appear how managing an account framework works with branches. Managing an account facilitates banking transactions by clients. The purpose of our project is to show how “Bank Account Management System” is working.

The essential point of this “Bank Account Management System” is to supply made strides design methodology, which conceives long run development, and alteration, which is fundamental for a center segment like keeping money. This requires the plan to be expandable and modifiable and so a measured approach is utilized in creating the application program. Anyone who is an Account holder in this bank can gotten to be a part of Bank Account Administration Framework. He must fill a shape with his individual subtle elements and Account Number. Bank is the place where clientss feel the sense of security for their property. Within the bank, clients store and withdraw their cash. Exchange of cash too may be a portion where client takes shield of the bank. Presently to keep the conviction and believe of clients, there's the positive require for administration of the bank, which can handle all this with consolation and ease. Smooth and effective administration influences the fulfillment of the clients and staff individuals, by implication.

Bank Account Management System keeps the count record as a total keeping money framework. It can keep the data of Account sort, account opening frame, Store finance, Withdrawal, and Looking the exchange, Exchange reports, Person account opening shape, Gather Account. The existing portion of this venture and it shows exchange reports, Measurable Rundown of Account sort and Intrigued Data.

1.2. The purpose of the Bank Account Management System

- What is the purpose of the database? Why is it required? What ought to it do?

Banks use databases to keep track of client accounts, transactions and deposits. Databases are used anywhere that data needs to be stored and easily retrieved. Database permits the client to form an account, store and pull back cash from his account. Clients can see their account details and perform the exchanges on their account concurring to their requirements.

- Who are the clients and what are their data needs?

A client of a bank is anybody who has connected to the bank to carry out credit, store, settlement, money and other managing an account operation. The bank needs to know the name and surname of the client, phone number, email, gender and date of account registration in order to enter this data into the bank's database. After this, the client can freely open accounts, make transactions, open deposits and etc.

- What are the issues that the system ought to solve?

This system will offer assistance to secure the organization and clients data, and also this system will offer assistance to streamline the capacities of the bank.

- What input information is accessible to the database?

Generally, in this project, accessible input information identified by an id, first name, last name, phone number, email, SSN (Social Security Number), Salary, hire date, designation, description and etc. Since this data is most required within the making of database.

- What kind of data ought to be put away in the database?

Only important information needs to be stored in the database. This can general information and confidential information. The bank requires more information from clients, because it can help the organization in a more streamlined database.

Business rules for Bank Account Management System:

- 1) Bank has branches, while branches connected with departments.
- 2) Connection between bank and branches should be mandatory.
- 3) One (optional) bank has one or many branches, while branches can have many (optional) departments, but department can have many employees.
- 4) Optional many employees can serve one or many products and services, while products and services have many (optional) customer acquests.
- 5) One or many customer acquests related to one or many customers, while customers can have many (optional) accounts.
- 6) One or many customer acquests can have many (optional) transactions.
- 7) Customers can have only one (optional) customer type.
- 8) Accounts can have only one (optional) account type.
- 9) Optional many accounts can have one or many transactions, while transactions can have only one (optional) transaction type.

1.3 Crow's Foot notation

Entities and their Attributes are:

Bank Entity: Attributes of Bank Entity are Bank_id, Bank_name, Bank_country, Bank_city and Bank_address. Bank_id is Primary Key for Bank Entity.

Branches Entity: Attributes of Branches Entity are Branch_id, Branch_name, Branch_city, Branch_country, Branch_address, Bank_id. Branch_id is Primary Key for Branches Entity. Bank_id is Foreign Key for Branches Entity.

Departments Entity: Attributes of Departments Entity are Department_id, Department_name, Description and Branch_id. Department_id is Primary Key for Departments Entity. Branch_id is Foreign Key for Departments Entity.

Products_services Entity: Attributes of Products_services Entity are Services_id, Services_description and Employee_id. Services_id is Primary Key for Products_services Entity. Employee_id is Foreign Key for Products_services Entity.

Customer_acquests Entity: Attributes of Customer_acquests Entity are Acquest_id, Acquest_date, Acquest_number, Customer_id and Services_id. Acquest_id is Primary Key for Customer_acquests Entity. Services_id is Foreign Key for Customer_acquests Entity.

Customers Entity: Attributes of Customers Entity are Customer_id, Customer_first_name, Customer_last_name, Customer_email, Customer_phone_number, registration_date, login, password, Gender, Customer_type_id.

Customer_id is Primary Key for Customers Entity. Customer_type_id is Foreign Key for Customers Entity.

Customer_types Entity: Attributes of Customer_types Entity are Customer_type_id and Customer_type_description. Customer_type_id is Primary Key for Customer_types Entity.

Accounts Entity: Attributes of Account Entity are Account_id, Account_name, Opened_date, Account_type_id, Customer_id. Account_id is Primary Key for Accounts Entity. Account_type_id, Customer_id are Foreign Keys for Accounts Entity.

Account_types Entity: Attributes of Account_types Entity are Account_type_id and Account_type_description.

Transactions Entity: Attributes of Transactions Entity are Transaction_id, Transaction_date, Transaction_amount, Acquest_id, Account_id and Transaction_type_id. Transaction_id is Primary Key for Transactions Entity. Acquest_id, Account_id, Transaction_type_id are Foreign Keys for Transactions Entity.

Transaction_types Entity: Attributes of Transaction_types Entity are Transaction_type_id and Transaction_type_description.

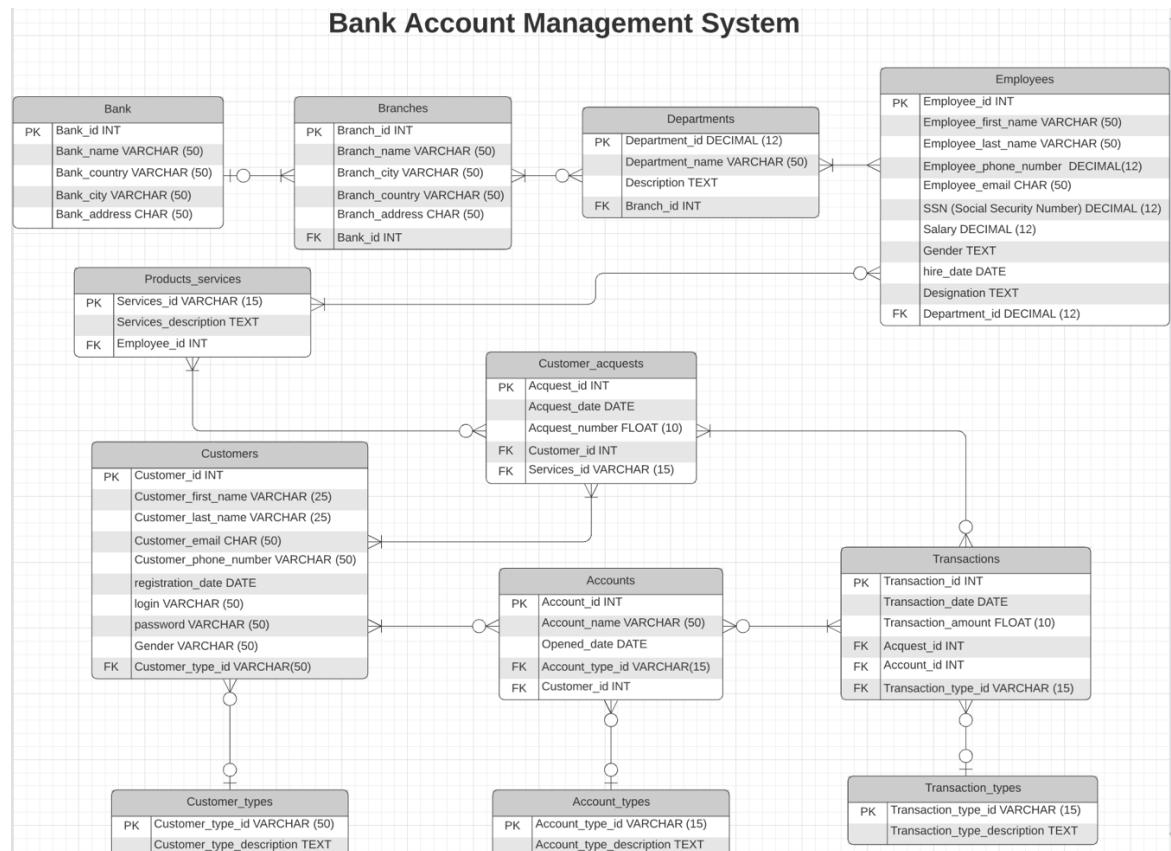


Figure-1: ERD Diagram of Bank Account Management System

This ERD diagram shows key information about Bank, including entities such as branches, departments, employees, products and services, customer acquests, customers, accounts, transactions, customer types, account types and transaction types.

Relationships are:

Bank has Branches: one to many.

One optional Bank can have one or many Branches.

Branches have Departments: many to many.

One or many Branches can have optional many Departments.

Departments have Employees: many to many.

One or many Departments can have many Employees.

Employees held Products_services: many to many

Optional many Employees can hold one or many Products_services.

Products_services relate to Customer_acquest: many to many.

One or many Products_services relate to optional many Customer_acquests.

Customers can have Customer_acquests: many to many.

One or many Customers can have one or many Customer_acquests.

Customer_acquests can have Transactions: many to many.

One or many Customer_acquests can have one or many Transactions.

Customers can have Customer_types: many to one.

Optional many Customers can have only one optional Customer_types that will describe customer's type.

Customers can have Accounts: many to many.

One or many Customers can have optional many accounts. It can be one account, or zero, or many accounts.

Accounts can have Account_types: many to one.

One or many accounts can have only one optional Account_types that will describe account's type.

Transactions can have Transaction_types: many to one.

One or many Transactions can have only one optional Transaction_types that will describe transaction's type.

1.4. Database

In the creating a database for our group project, we used online SQL server compiler. The syntax of language that used in SQL server is the same as in SQL Shell which we used during the ICT course. To create the database we wrote all entities and attributes from our ERD diagram to the code. Also, we linked our tables using constraints (PK, FK and etc.). **Figure-2 shows** result of SQL compiler after creating tables.

Chapter 2

2.1. DDL Statements

```
ALTER TABLE Transactions ADD Transaction_password char;
ALTER TABLE Transactions DROP column Transaction_password;
ALTER TABLE Bank ADD Bank_advertising TEXT;
ALTER TABLE Bank ALTER column Bank_advertising VARCHAR;
ALTER TABLE Bank DROP column Bank_advertising;
ALTER TABLE Accounts ADD FOREIGN KEY (Customer_id)
REFERENCES Customers (Customer_id);
```

Figure-3: Result of ALTER TABLE statements

The next step of creating our database is using ALTER TABLE statements (add, drop and constraints). From the Figure 3, you can see that we add new column Bank_advertising to the table Bank. And after this, we drop our Bank_advertising column. We do the same action with the column Transaction_password. First, we add column Transaction_password to the Transactions table, then we drop it. And at the end we use constraint command, to create a new constraint in the Accounts table (foreign key) to the column Customer_id from the Customers table. Figure-3 shows result of ALTER TABLE statements.

2.2. DML Statements

	Bank_id	Bank_name	Bank_country	Bank_city	Bank_address
1	1	Kaspi Bank	Kazakhstan	Almaty	Nauryzbai batyr street,154
2	2	Halyk Bank	Kazakhstan	Nur-Sultan	Kabanbai batyr street,17
3	3	Qazqom Bank	Kazakhstan	Almaty	Gagarin street,135/8
4	4	Jysan Bank	Kazakhstan	Nur-Sultan	Syganak street,24
5	5	Eurasian Bank	Kazakhstan	Nur-Sultan	Kabanbai Batyr street,30
6	6	Sberbank	Russia	Moscow	Vavilova street,19
7	7	ForteBank	Kazakhstan	Nur-Sultan	Dostyk street,8/1
8	8	RBK Bank	Kazakhstan	Nur-Sultan	Saraishyk street,11
9	9	ATF Bank	Kazakhstan	Nur-Sultan	Saraishyk street,34
10	10	VTB Bank	Russia	Moscow	Myasnitskaya street,35

Figure-4.1: Result of INSERT INTO statement for the Bank table

The next step of our project was using INSERT INTO statement. We use this command to supplement our database. As you can see from the Figure-4.1, we have 10 different Banks with 10 rows with the different Bank id's, Bank names, Bank countries, Bank cities, Bank addresses. Figure-4.1 shows the results of INSERT INTO statement for the Bank table.

	Branch_id	Branch_name	Branch_city	Branch_country	Branch_address	Bank_id
1	11	Kaspi Bank Branch	Nur-Sultan	Kazakhstan	Kunayev street,14/2	1
2	12	Halyk Bank Branch	Almaty	Kazakhstan	Altynsarin street,23	2
3	13	Qazqom Bank Branch	Shymkent	Kazakhstan	Ilyayeva street,33	3
4	14	Jysan Bank Branch	Almaty	Kazakhstan	Rozybakiyeva street,273	4
5	15	Eurasian Bank Branch	Shymkent	Kazakhstan	Tashenov street,4	5
6	16	Sberbank Branch	Nur-Sultan	Kazakhstan	Dostyk street,9	6
7	17	ForteBank Branch	Shymkent	Kazakhstan	Zheltoksan street,15	7
8	18	RBK Bank Branch	Almaty	Kazakhstan	Sharipova street,84	8
9	19	ATF Bank Branch	Almaty	Kazakhstan	Tole bi street,83	9
10	20	VTB Bank Branch	Nur-Sultan	Kazakhstan	Dostyk street,12	10

Figure-4.2: Result of INSERT INTO statement for the Branches table

Next, we use the same step of using INSERT INTO statement for the Branches table. As you can see from the Figure-4.2, we have 10 rows with the different Branch id's, Branch names, Branch countries, Branch cities, Branch addresses, and Bank id's from the previous table Bank. Figure-4.2 shows the results of INSERT INTO statement for the Branch table.

	Department_id	Department_name	Description	Branch_id
1	101	Risk Management Department	Manage the risk management process on day to day basis	11
2	102	Audit Department	Responsible for evaluating operational procedures,control functions	12
3	103	R&D Department	Responsible for finding knowledge to create new products	13
4	104	Branches Affairs	Responsible for working with branches of bank	14
5	105	International Affairs	Responsible for working with international organizations, governments, etc.	15
6	106	Credit Department	Analyze a credit request and suggest how to improve loan structure	16
7	107	IT Department	Introduce IT solutions for the bank departments	17
8	108	Financial Department	Work with customer finances	18
9	109	HR Department	Analyze future requirements	19
10	111	Financial Department	Work with customer finances	11
11	112	IT Department	Introduce IT solutions for the bank departments and responsible for IT security	12
12	113	Credit Department	Analyze a credit request and suggest how to improve loan structure	13
13	114	International Affairs	Responsible for working with international organizations, governments, etc.	14
14	115	Branches Affairs	Responsible for working with branches of bank	15
15	116	Audit Department	Responsible for evaluating operational procedures,control functions	16
16	117	Risk Management Department	Manage the risk management process on day to day basis	17
17	118	R&D Department	Responsible for finding knowledge to create new products	18
18	119	Corporate Banking Department	Refers to the aspect of banking that deals with corporate customers	19
19	200	Corporate Banking Department	Refers to the aspect of banking that deals with corporate customers	20
20	210	HR Department	Analyze future requirements	20

Figure-4.3: Result of INSERT INTO statement for the Departments table

Next, we use the same step of using INSERT INTO statement for the Departments table. As you can see from the Figure-4.3, we have 20 rows with the different Department id's, Department names, Department descriptions and Branch id's from the previous Branches table. Figure-4.3 shows the results of INSERT INTO statement for the Departments table.

	Employee_id	Employee_first_name	Employee_last_name	Employee_phone_number	Employee_email	SSN	Salary	Gender	Hire_date	Designation	Department_id
1	1111	Assem	Moldash	87010010101	assem@mail.ru	101	290000	Female	29.04.2015 00:00:00	Financier	111
2	1212	Karlyga	Akilova	87019755700	karla@12mail.ru	112	170000	Female	09.09.2014 00:00:00	Corporate Manager	119
3	1313	Asiya	Beimbetova	87010150607	asiya@13mail.ru	113	105000	Female	22.07.2018 00:00:00	Personnel Officer	210
4	2121	Alen	Zhunisbekov	87028765432	alen@mail.ru	114	230000	Male	29.06.2019 00:00:00	Risk Manager	101
5	2212	Lyazat	Saparova	87010000048	lyazat@22mail.ru	13	300000	Female	07.05.2015 00:00:00	Corporate Director	200
6	2222	Alem	Beisenbai	87070020202	alem@2mail.ru	202	90000	Male	20.04.2016 00:00:00	IT Manager	112
7	3131	Daniyar	Orabayev	87054532345	daniyar@1mail.ru	115	470000	Male	19.03.2020 00:00:00	Main Auditor	102
8	3333	Madiyar	Kakharman	87050030303	madiyar@3mail.ru	303	500000	Male	29.09.2015 00:00:00	Engineer	107
9	4141	Zhuldyz	Kozhabayeva	87086300485	zhuldyz@31mail.ru	116	107000	Female	19.04.2019 00:00:00	Specialists assistant	103
10	4444	Zhanar	Alchickeyeva	87020040404	zhanar@4mail.ru	404	120000	Female	04.04.2014 00:00:00	Assistant	113
11	5151	Kamaliya	Sultankzy	87756334323	ksultan@51mail.ru	117	90000	Female	08.12.2018 00:00:00	Manager	105
12	5555	Kamilla	Ospanova	87010050505	kamilla@5mail.ru	505	230000	Female	15.10.2016 00:00:00	General Manager	114
13	6161	Dias	Amangeldin	87029007557	dias@61mail.ru	118	240000	Male	09.03.2020 00:00:00	Valuer	106
14	6666	Dauren	Kambar	87050060606	dauren@6mail.ru	606	250000	Male	19.08.2013 00:00:00	Administrator	115
15	7171	Alima	Nurlybekova	87056551209	alima@71mail.ru	119	200000	Female	19.08.2019 00:00:00	Placeman	104
16	7777	Alina	Timurkyzy	87070070707	alina@7mail.ru	707	310000	Female	21.01.2015 00:00:00	Auditor	116
17	8181	Mansur	Bisembay	87019008558	mansur@81mail.ru	11	105000	Male	03.04.2013 00:00:00	Personnel Officer	109
18	8888	Maksim	Melnikov	87080080808	maksim@8mail.ru	808	230000	Male	01.12.2013 00:00:00	Risk Manager	117
19	9191	Karim	Ermakhanov	87000991551	karim@91mail.ru	12	90000	Male	10.09.2017 00:00:00	Cashier	108
20	9999	Matvey	Stepanenko	87010090909	matvey@9mail.ru	909	600000	Male	17.02.2017 00:00:00	IT Specialist	118

Figure-4.4: Result of INSERT INTO statement for the Employees table

Next, we use the same step of using INSERT INTO statement for the Employees table. As you can see from the Figure-4.4, we have 20 rows with the different Employee id's, Employee first names, Employee last names, Employee phone numbers, Employee emails, SSN's, Salary, Genders, hire_dates, Designations, Department_id's from the previous Departments table. Figure-4.4 shows the results of INSERT INTO statement for the Employees table.

	Services_id	Services_description	Employee_id
1	221	Personal Account Service	1111
2	222	Personal Account Service	2212
3	223	Cyber Security Service	2222
4	224	Credit Service	6161
5	225	Credit Service	4444
6	226	Deposit Service	1111
7	227	Corporate Service	2212
8	228	Credit Service	2121
9	229	Currency Exchange Service	9191
10	230	Cash Collection Service	1212

Figure-4.5: Result of INSERT INTO statement for the Products_services table

Next, we use the same step of using INSERT INTO statement for the Products_services table. As you can see from the Figure-4.5, we have 20 rows with the different Services id's, Services descriptions and Employee id's from the previous Employees table. Figure-4.5 shows the results of INSERT INTO statement for the Products_services table.

	Customer_type_id	Customer_type_description
1	321	Legal Entity: Individual entrepreneur
2	322	Individual Entity: Citizens, foreign citizens and stateless persons
3	323	Corporate Client: Citizens that works in companies, organisations and etc.
4	324	Legal Entity: Individual entrepreneur
5	325	Individual Entity: Citizens, foreign citizens and stateless persons
6	326	Corporate Client: Citizens that works in companies, organisations and etc.
7	327	Legal Entity: Individual entrepreneur
8	328	Individual Entity: Citizens, foreign citizens and stateless persons
9	329	Corporate Client: Citizens that works in companies, organisations and etc.
10	330	Legal Entity: Individual entrepreneur

Figure-4.6: Result of INSERT INTO statement for the Customer_types table

Next, we use the same step of using INSERT INTO statement for the Customer_types table. As you can see from the Figure-4.6, we have 10 rows with the different Customer_type_id and Customer_type_description. Figure-4.6 shows the results of INSERT INTO statement for the Customer_types table.

	Customer_id	Customer_first_name	Customer_last_name	Customer_email	Customer_phone_number	registration_date	login	password	Gender	Customer_type_id
1	551	Assel	Butovand	assel@mail.ru	87059010101	29.04.2015 00:00:00	assel0	29070	Female	321
2	552	Allen	Beikay	allen@2mail.ru	87059020202	20.04.2016 00:00:00	allen9	96070	Male	322
3	553	Aldiyar	Makhan	aldiyar@3mail.ru	87059030303	29.09.2015 00:00:00	aldiyar8	56070	Male	323
4	554	Zhanat	Altynkyzy	zhanat@4mail.ru	87029040404	15.10.2016 00:00:00	zhanat7	12070	Female	324
5	555	Kamshat	Aspanova	kamshat@5mail.ru	87019050505	04.04.2014 00:00:00	kamshat6	23070	Female	325
6	556	Daulet	Kopbar	daulet@6mail.ru	87059060606	21.01.2015 00:00:00	daulet5	25070	Male	326
7	557	Altyn	Timurova	altyn@7mail.ru	87079070707	17.02.2017 00:00:00	altyn4	31070	Female	327
8	558	Maksat	Mashat	maksat@8mail.ru	87089080808	19.08.2013 00:00:00	maksat3	23070	Male	328
9	559	Miras	Sergazyuly	miras@9mail.ru	87019090909	09.09.2014 00:00:00	miras2	66070	Male	329
10	560	Karlygash	Kanatbekova	karlygash@12mail.ru	87018755700	01.12.2013 00:00:00	karlygash1	17070	Female	330

Figure-4.7: Result of INSERT INTO statement for the Customers table

Next, we use the same step of using INSERT INTO statement for the Customers table. As you can see from the Figure-4.7, we have 10 rows with the different Customer id's, Customer first names, Customer last names, Customer phone numbers, Customer emails, registration dates, logins, passwords, Genders and Customer type id's from the previous Customer_types table. Figure-4.7 shows the results of INSERT INTO statement for the Customers table.

	Acquest_id	Acquest_date	Acquest_number	Customer_id	Services_id
1	851	19.05.2020 00:00:00	1	551	221
2	852	10.05.2018 00:00:00	2	552	222
3	853	19.10.2016 00:00:00	3	553	223
4	854	25.11.2017 00:00:00	4	554	224
5	855	24.05.2015 00:00:00	5	555	225
6	856	11.02.2016 00:00:00	6	556	226
7	857	27.03.2018 00:00:00	7	557	227
8	858	29.09.2014 00:00:00	8	558	228
9	859	29.10.2015 00:00:00	9	559	229
10	860	11.01.2014 00:00:00	10	560	230

Figure-4.8: Result of INSERT INTO statement for the Customer_acquests table

Next, we use the same step of using INSERT INTO statement for the Customer_acquests table. As you can see from the Figure-4.8, we have 10 rows with the different Acquest_id, Acquest_date, Acquest_number, and Customer_id, Services_id from the previous Customers and Products_services tables. Figure-4.8 shows the results of INSERT INTO statement for the Customer_acquests table.

	Account_type_id	Account_type_description
1	1000	Legal Entity Account
2	991	Legal Entity Account
3	992	Individual Entity Account
4	993	Corporate Client Account
5	994	Legal Entity Account
6	995	Individual Entity Account
7	996	Corporate Client Account
8	997	Legal Entity Account
9	998	Individual Entity Account
10	999	Corporate Client Account

Figure-4.9: Result of INSERT INTO statement for the Account_types table

Next, we use the same step of using INSERT INTO statement for the Account_types table. As you can see from the Figure-4.9, we have 10 rows with the different Account_type_id, Account_type_description. Figure-4.9 shows the results of INSERT INTO statement for the Account_types table.

	Account_id	Account_name	Opened_date	Account_type_id	Customer_id
1	671	Assel Butovand	29.04.2015 00:00:00	991	551
2	672	Allen Beikay	20.04.2016 00:00:00	992	552
3	673	Aldiyar Makhan	29.09.2015 00:00:00	993	553
4	674	Zhanat Altynkyzy	15.10.2016 00:00:00	994	554
5	675	Kamshat Aspanova	04.04.2014 00:00:00	995	555
6	676	Daulet Kopbar	21.01.2015 00:00:00	996	556
7	677	Altyn Timurova	17.02.2017 00:00:00	997	557
8	678	Maksat Mashat	19.08.2013 00:00:00	998	558
9	679	Miras Sergazyuly	09.09.2014 00:00:00	999	559
10	680	Karlygash Kanatbekova	01.12.2013 00:00:00	1000	560

Figure-5: Result of INSERT INTO statement for the Accounts table

Next, we use the same step of using INSERT INTO statement for the Accounts table. As you can see from the Figure-5, we have 10 rows with the different Account_id, Account_name, Opened_date, and Account_type_id, Customer_id from the previous Customers and Account_types tables. Figure-5 shows the results of INSERT INTO statement for the Accounts table.

	Transaction_type_id	Transaction_type_description
1	421	Online: online a web-based store or online banking service
2	422	Transfer: Move money from one account to another
3	423	Charge: Record a purchase on a debit card
4	424	Online: online a web-based store or online banking service
5	425	Transfer: Move money from one account to another
6	426	Charge: Record a purchase on a debit card
7	427	Online: online a web-based store or online banking service
8	428	Transfer: Move money from one account to another
9	429	Charge: Record a purchase on a debit card
10	430	Online: online a web-based store or online banking service
11	431	Online: online a web-based store or online banking service
12	432	Transfer: Move money from one account to another
13	433	Charge: Record a purchase on a debit card
14	434	Online: online a web-based store or online banking service
15	435	Transfer: Move money from one account to another
16	436	Charge: Record a purchase on a debit card
17	437	Online: online a web-based store or online banking service
18	438	Transfer: Move money from one account to another
19	439	Charge: Record a purchase on a debit card
20	440	Online: online a web-based store or online banking service

Figure-5.1: Result of INSERT INTO statement for the Transaction_types table

Next, we use the same step of using INSERT INTO statement for the Account_types table. As you can see from the Figure-5.1, we have 20 rows with the different Transaction_type_id, Transaction_type_description. Figure-5.1 shows the results of INSERT INTO statement for the Transaction_types table.

	Transaction_id	Transaction_date	Transaction_amount	Acquest_id	Account_id	Transaction_type_id
1	961	01.04.2020 00:00:00	20000	851	671	421
2	962	09.09.2020 00:00:00	15000	852	672	422
3	963	19.08.2020 00:00:00	5000	853	673	423
4	964	17.10.2019 00:00:00	700000	854	674	424
5	965	21.01.2018 00:00:00	220000	855	675	425
6	966	04.04.2017 00:00:00	400000	856	676	426
7	967	15.10.2020 00:00:00	100000	857	677	427
8	968	29.09.2020 00:00:00	30000	858	678	428
9	969	20.04.2020 00:00:00	2990	859	679	429
10	970	29.04.2019 00:00:00	100000	860	680	430
11	971	19.04.2019 00:00:00	200000	851	671	431
12	972	10.04.2020 00:00:00	1500	852	672	432
13	973	09.09.2020 00:00:00	2000	853	673	433
14	974	05.10.2020 00:00:00	70000	854	674	434
15	975	14.04.2016 00:00:00	170000	855	675	435
16	976	11.01.2017 00:00:00	200000	856	676	436
17	977	17.10.2020 00:00:00	200000	857	677	437
18	978	29.08.2020 00:00:00	20000	858	678	438
19	979	20.05.2020 00:00:00	990	859	679	439
20	980	29.03.2019 00:00:00	150000	860	680	440

Figure-5.2: Result of INSERT INTO statement for the Transactions table

Next, we use the same step of using INSERT INTO statement for the Transactions table. As you can see from the Figure 5.2, we have 20 rows with the different Transaction_id, Transaction_date, Transaction_amount and Account_id, Acquest_id, Transaction_type_id from the previous Accounts, Customer_acquests and Transaction_types tables. Figure-5.2 shows the results of INSERT INTO statement for the Transactions table.

	Bank_id	Bank_name	Bank_country	Bank_city	Bank_address
1	1	Kaspi Bank	Kazakhstan	Almaty	Nauryzbai batyr street,154
2	2	Halyk Bank	Kazakhstan	Nur-Sultan	Kabanbai batyr street,17
3	3	Qazqom Bank	Kazakhstan	Almaty	Gagarin street,135/8
4	4	Jysan Bank	Kazakhstan	Nur-Sultan	Syganak street,24
5	5	Eurasian Bank	Kazakhstan	Nur-Sultan	Kabanbai Batyr street,30
6	6	Kassa Nova Bank	Kazakhstan	Nur-Sultan	Bauyrzhan Momyshuly street,22/14
7	7	ForteBank	Kazakhstan	Nur-Sultan	Dostyk street,8/1
8	8	RBK Bank	Kazakhstan	Nur-Sultan	Saraishyk street,11
9	9	ATF Bank	Kazakhstan	Nur-Sultan	Saraishyk street,34
10	10	VTB Bank	Russia	Moscow	Myasnitskaya street,35

Figure-5.3: Result of UPDATE statement for Bank table

The UPDATE statement is used to update the data of an existing table in database. UPDATE was used for changing information about Bank (Bank name, Bank country, Bank city and Bank address) where Bank_id is 06. Figure-5.3 shows result of UPDATE statement for Bank table.

	Branch_id	Branch_name	Branch_city	Branch_country	Branch_address	Bank_id
1	11	Kaspi Bank Branch	Nur-Sultan	Kazakhstan	Kunayev street,14/2	1
2	12	Halyk Bank Branch	Almaty	Kazakhstan	Altynsarın street,23	2
3	13	Qazqom Bank Branch	Shymkent	Kazakhstan	Ilyayeva street,33	3
4	14	Jysan Bank Branch	Almaty	Kazakhstan	Rozybakiyeva street,273	4
5	15	Eurasian Bank Branch	Shymkent	Kazakhstan	Tashenov street,4	5
6	16	Kassa Nova Branch	Almaty	Kazakhstan	Altynsarın street, 27	6
7	17	ForteBank Branch	Shymkent	Kazakhstan	Zheltoksan street,15	7
8	18	RBK Bank Branch	Almaty	Kazakhstan	Sharipova street,84	8
9	19	ATF Bank Branch	Almaty	Kazakhstan	Tole bi street,83	9
10	20	VTB Bank Branch	Nur-Sultan	Kazakhstan	Dostyk street,12	10

Figure-5.4: Result of UPDATE statement for Branches table

For the next step, UPDATE was used for changing information about Branches (Branch name, Branch city and Bank address) where Branch_id is 16. Figure-5.4 shows result of UPDATE statement for Branches table.

	Department_id	Department_name	Description	Branch_id
1	101	Risk Management Department	Manage the risk management process on day to day basis	11
2	102	Audit Department	Responsible for evaluating operational procedures, control functions	12
3	103	R&D Department	Responsible for finding knowledge to create new products	13
4	104	Branches Affairs	Responsible for working with branches of bank	14
5	105	International Affairs	Responsible for working with international organizations, governments, etc.	15
6	106	Credit Department	Analyze a credit request and suggest how to improve loan structure	16
7	107	IT Department	Introduce IT solutions for the bank departments	17
8	108	Financial Department	Work with customer finances	18
9	109	HR Department	Analyze future requirements	19
10	111	Financial Department	Work with customer finances	11
11	112	IT Department	Introduce IT solutions for the bank departments and responsible for IT security	12
12	113	Credit Department	Analyze a credit request and suggest how to improve loan structure	13
13	114	Marketing Department	Provides the necessary research to identify your target customers and other audiences.	14
14	115	Branches Affairs	Responsible for working with branches of bank	15
15	116	Audit Department	Responsible for evaluating operational procedures, control functions	16
16	117	Risk Management Department	Manage the risk management process on day to day basis	17
17	118	R&D Department	Responsible for finding knowledge to create new products	18
18	119	Corporate Banking Department	Refers to the aspect of banking that deals with corporate customers	19
19	200	Corporate Banking Department	Refers to the aspect of banking that deals with corporate customers	20
20	210	HR Department	Analyze future requirements	20

Figure-5.5: Result of UPDATE statement for Departments table

For the next step, UPDATE was used for changing information about Departments (Department_name and Description) where Department_id is 114. Figure-5.5 shows result of UPDATE statement for Departments table.

	Employee_id	Employee_first_name	Employee_last_name	Employee_phone_number	Employee_email	SSN	Salary	Gender	hire_date	Designation	Department_id
1	1111	Azat	Janmolda	87050150505	azatl@mail.ru	101	300000	Male	20.08.2019 00:00:00	Personnel Officer	109
2	1212	Karlyga	Akilova	87019755700	karla@12mail.ru	112	170000	Female	09.09.2014 00:00:00	Corporate Manager	119
3	1313	Asiya	Beimbetova	87010150607	asiya@13mail.ru	113	105000	Female	22.07.2018 00:00:00	Personnel Officer	210
4	2121	Alen	Zhunisbekov	87028765432	alen@mail.ru	114	230000	Male	29.06.2019 00:00:00	Risk Manager	101
5	2212	Lyzazat	Saparova	87010000048	lyazat@22mail.ru	115	300000	Female	07.05.2015 00:00:00	Corporate Director	200
6	2222	Alem	Beisenbai	87070020202	alem@2mail.ru	202	90000	Male	20.04.2016 00:00:00	IT Manager	112
7	3131	Daniyar	Orazbayev	87054532345	daniyar@21mail.ru	116	470000	Male	19.03.2020 00:00:00	Main Auditor	102
8	3333	Madiyar	Kakharman	87050030303	madiyar@3mail.ru	303	500000	Male	29.09.2015 00:00:00	Engineer	107
9	4141	Zhuldyz	Kozhabayeva	87086300485	zhuldyz@31mail.ru	117	107000	Female	19.04.2019 00:00:00	Specialists assistant	103
10	4444	Zhanar	Alchickeyeva	87020040404	zhanar@4mail.ru	404	120000	Female	04.04.2014 00:00:00	Assistant	113
11	5151	Kamaliya	Sultankzy	87756334323	ksultan@51mail.ru	118	90000	Female	08.12.2018 00:00:00	Manager	105
12	5555	Kamilla	Ospanova	87010050505	kamilla@5mail.ru	505	230000	Female	15.10.2016 00:00:00	General Manager	114
13	6161	Dias	Amangeldin	87029007557	dias@61mail.ru	119	240000	Male	09.03.2020 00:00:00	Valuer	106
14	6666	Dauren	Kambar	87050060606	dauren@6mail.ru	606	250000	Male	19.08.2013 00:00:00	Administrator	115
15	7171	Alima	Nurlybekova	87056551209	alima@71mail.ru	119	200000	Female	19.08.2019 00:00:00	Placeman	104
16	7777	Alina	Timurkyzy	87070070707	alina@7mail.ru	707	310000	Female	21.01.2015 00:00:00	Auditor	116
17	8181	Mansur	Bisembay	87019008558	mansur@81mail.ru	11	105000	Male	03.04.2013 00:00:00	Personnel Officer	109
18	8888	Maksim	Melnikov	87080080808	maksim@8mail.ru	808	230000	Male	01.12.2013 00:00:00	Risk Manager	117
19	9191	Karim	Ermakhanov	87000991551	karim@91mail.ru	12	90000	Male	10.09.2017 00:00:00	Cashier	108
20	9999	Matvey	Stepanenko	87010090909	matvey@9mail.ru	909	600000	Male	17.02.2017 00:00:00	IT Specialist	118

Figure-5.6: Result of UPDATE statement for Employees table

For the next step, UPDATE was used for changing information about Employees (Employee_first_name, Employee_last_name, Employee_phone_number, Employee_email, Salary, Gender, hire_date, Designation, Department_id) where Employee_id is 1111. Figure-5.6 shows result of UPDATE statement for Employees table.

	Services_id	Services_description	Employee_id
1	221	Personal Account Service	1111
2	222	Personal Account Service	2212
3	223	Credit Service	8888
4	224	Credit Service	6161
5	225	Credit Service	4444
6	226	Deposit Service	1111
7	227	Corporate Service	2212
8	228	Credit Service	2121
9	229	Currency Exchange Service	9191
10	230	Cash Collection Service	1212

Figure-5.7: Result of UPDATE statement for Products_services table

For the next step, UPDATE was used for changing information about Departments (Services_description) where Services_id is 223. Figure-5.7 shows result of UPDATE statement for Products_services table.

	Customer_type_id	Customer_type_description
1	321	Individual Entity: Citizens, foreign citizens and stateless persons
2	322	Individual Entity: Citizens, foreign citizens and stateless persons
3	323	Corporate Client: Citizens that works in companies, organisations and etc.
4	324	Legal Entity: Individual entrepreneur
5	325	Individual Entity: Citizens, foreign citizens and stateless persons
6	326	Corporate Client: Citizens that works in companies, organisations and etc.
7	327	Legal Entity: Individual entrepreneur
8	328	Individual Entity: Citizens, foreign citizens and stateless persons
9	329	Corporate Client: Citizens that works in companies, organisations and etc.
10	330	Legal Entity: Individual entrepreneur

Figure-5.8: Result of UPDATE statement for table Customer_types

For the next step, UPDATE was used for changing information about Customer_types (Customer_type_description) where Customer_type_id is 321. Figure-5.8 shows result of UPDATE statement for Customer_types table.

	Customer_id	Customer_first_name	Customer_last_name	Customer_email	Customer_phone_number	registration_date	login	password	Gender	Customer_type_id
1	551	Damir	Shokhan	damirl@mail.ru	87015514334	22.07.2019 00:00:00	damirl	04070	Male	321
2	552	Allen	Beikay	allen@2mail.ru	87059020202	20.04.2016 00:00:00	allen9	96070	Male	322
3	553	Aldiyar	Makhan	aldiyar@3mail.ru	87059030303	29.09.2015 00:00:00	aldiyar8	56070	Male	323
4	554	Zhanat	Altynkyzy	zhanat@4mail.ru	87029040404	15.10.2016 00:00:00	zhanat7	12070	Female	324
5	555	Kamshat	Aspanova	kamshat@5mail.ru	87019050505	04.04.2014 00:00:00	kamshat6	23070	Female	325
6	556	Daulet	Kopbar	daulet@6mail.ru	87059060606	21.01.2015 00:00:00	daulet5	25070	Male	326
7	557	Altyn	Timurova	altyn@7mail.ru	87079070707	17.02.2017 00:00:00	altyn4	31070	Female	327
8	558	Maksat	Mashat	maksat@8mail.ru	87089080808	19.08.2013 00:00:00	maksat3	23070	Male	328
9	559	Miras	Sergazyuly	miras@9mail.ru	87019090909	09.09.2014 00:00:00	miras2	66070	Male	329
10	560	Karlygash	Kanatbekova	karlygash@12mail.ru	87018755700	01.12.2013 00:00:00	karlygash1	17070	Female	330

Figure-5.9: Result of UPDATE statement for table Customers

For the next step, UPDATE was used for changing information about Customers (Customer_first_name, Customer_last_name, Customer_phone_number, Customer_email, registration_date, login, password, Gender) where Customer_id is 551. Figure-5.9 shows result of UPDATE statement for Customers table.

	Acquest_id	Acquest_date	Acquest_number	Customer_id	Services_id
1	851	20.05.2019 00:00:00	20	551	222
2	852	10.05.2018 00:00:00	2	552	222
3	853	19.10.2016 00:00:00	3	553	223
4	854	25.11.2017 00:00:00	4	554	224
5	855	24.05.2015 00:00:00	5	555	225
6	856	11.02.2016 00:00:00	6	556	226
7	857	27.03.2018 00:00:00	7	557	227
8	858	29.09.2014 00:00:00	8	558	228
9	859	29.10.2015 00:00:00	9	559	229
10	860	11.01.2014 00:00:00	10	560	230

Figure-6: Result of UPDATE statement for table Customer_acquests

For the next step, UPDATE was used for changing information about Customer_acquests (Acquest_date, Acquest_number, Services_id) where Acquest_id is 851. Figure-6 shows result of UPDATE statement for Customer_acquests table.

	Account_type_id	Account_type_description
1	1000	Legal Entity Account
2	991	Individual Entity Account
3	992	Individual Entity Account
4	993	Corporate Client Account
5	994	Legal Entity Account
6	995	Individual Entity Account
7	996	Corporate Client Account
8	997	Legal Entity Account
9	998	Individual Entity Account
10	999	Corporate Client Account

Figure-6.1: Result of UPDATE statement for table Account_types

For the next step, UPDATE was used for changing information about Account_types (Account_type_description) where Account_type_id is 991. Figure-6.1 shows result of UPDATE statement for Account_types table.

	Account_id	Account_name	Opened_date	Account_type_id	Customer_id
1	671	Damir Shokhan	22.07.2019 00:00:00	992	551
2	672	Allen Beikay	20.04.2016 00:00:00	992	552
3	673	Aldiyar Makhan	29.09.2015 00:00:00	993	553
4	674	Zhanat Altynkyzy	15.10.2016 00:00:00	994	554
5	675	Kamshat Aspanova	04.04.2014 00:00:00	995	555
6	676	Daulet Kopbar	21.01.2015 00:00:00	996	556
7	677	Altyn Timurova	17.02.2017 00:00:00	997	557
8	678	Maksat Mashat	19.08.2013 00:00:00	998	558
9	679	Miras Sergazyuly	09.09.2014 00:00:00	999	559
10	680	Karlygash Kanatbekova	01.12.2013 00:00:00	1000	560

Figure-6.2: Result of UPDATE statement for table Accounts

For the next step, UPDATE was used for changing information about Accounts (Account_name, Opened_date, Account_type_id) where Account_id is. Figure-6.2 shows result of UPDATE statement for Accounts table.

	Transaction_type_id	Transaction_type_description
1	421	Transfer: Move money from one account to another
2	422	Transfer: Move money from one account to another
3	423	Charge: Record a purchase on a debit card
4	424	Online: online a web-based store or online banking service
5	425	Transfer: Move money from one account to another
6	426	Charge: Record a purchase on a debit card
7	427	Online: online a web-based store or online banking service
8	428	Transfer: Move money from one account to another
9	429	Charge: Record a purchase on a debit card
10	430	Online: online a web-based store or online banking service
11	431	Online: online a web-based store or online banking service
12	432	Transfer: Move money from one account to another
13	433	Charge: Record a purchase on a debit card
14	434	Online: online a web-based store or online banking service
15	435	Transfer: Move money from one account to another
16	436	Charge: Record a purchase on a debit card
17	437	Online: online a web-based store or online banking service
18	438	Transfer: Move money from one account to another
19	439	Charge: Record a purchase on a debit card
20	440	Online: online a web-based store or online banking service

Figure-6.3: Result of UPDATE statement for table Transaction_types

For the next step, UPDATE was used for changing information about Transaction_types (Transaction_type_description) where Transaction_type_id is 421. Figure-6.3 shows result of UPDATE statement for Transaction_types table.

	Transaction_id	Transaction_date	Transaction_amount	Acquest_id	Account_id	Transaction_type_id
1	961	09.08.2020 00:00:00	25000	852	672	421
2	962	09.09.2020 00:00:00	15000	852	672	422
3	963	19.08.2020 00:00:00	5000	853	673	423
4	964	17.10.2019 00:00:00	700000	854	674	424
5	965	21.01.2018 00:00:00	220000	855	675	425
6	966	04.04.2017 00:00:00	400000	856	676	426
7	967	15.10.2020 00:00:00	100000	857	677	427
8	968	29.09.2020 00:00:00	30000	858	678	428
9	969	20.04.2020 00:00:00	2990	859	679	429
10	970	29.04.2019 00:00:00	100000	860	680	430
11	971	19.04.2019 00:00:00	200000	851	671	431
12	972	10.04.2020 00:00:00	1500	852	672	432
13	973	09.09.2020 00:00:00	2000	853	673	433
14	974	05.10.2020 00:00:00	70000	854	674	434
15	975	14.04.2016 00:00:00	170000	855	675	435
16	976	11.01.2017 00:00:00	200000	856	676	436
17	977	17.10.2020 00:00:00	200000	857	677	437
18	978	29.08.2020 00:00:00	20000	858	678	438
19	979	20.05.2020 00:00:00	990	859	679	439
20	980	29.03.2019 00:00:00	150000	860	680	440

Figure-6.4: Result of UPDATE statement for table Transactions

For the next step, UPDATE was used for changing information about Transactions (Transaction_date, Transaction_amount, Account_id, Acquest_id, Transaction_type_id) where Transaction_id is 961. Figure-6.4 shows result of UPDATE statement for Transactions table.

	Transaction_id	Transaction_date	Transaction_amount	Acquest_id	Account_id	Transaction_type_id
1	962	09.09.2020 00:00:00	15000	852	672	422
2	963	19.08.2020 00:00:00	5000	853	673	423
3	964	17.10.2019 00:00:00	700000	854	674	424
4	965	21.01.2018 00:00:00	220000	855	675	425
5	966	04.04.2017 00:00:00	400000	856	676	426
6	967	15.10.2020 00:00:00	100000	857	677	427
7	968	29.09.2020 00:00:00	30000	858	678	428
8	969	20.04.2020 00:00:00	2990	859	679	429
9	970	29.04.2019 00:00:00	100000	860	680	430
10	971	19.04.2019 00:00:00	200000	851	671	431
11	972	10.04.2020 00:00:00	1500	852	672	432
12	973	09.09.2020 00:00:00	2000	853	673	433
13	974	05.10.2020 00:00:00	70000	854	674	434
14	975	14.04.2016 00:00:00	170000	855	675	435
15	976	11.01.2017 00:00:00	200000	856	676	436
16	977	17.10.2020 00:00:00	200000	857	677	437
17	978	29.08.2020 00:00:00	20000	858	678	438
18	979	20.05.2020 00:00:00	990	859	679	439
19	980	29.03.2019 00:00:00	150000	860	680	440

Figure-6.5: Result of DELETE statement for table Transactions

For the next step, DELETE was used to delete all rows from Transactions (Transaction_date, Transaction_amount, Account_id, Acquest_id, Transaction_type_id) where Transaction_id is 961. Figure-6.5 shows result of DELETE statement for Transactions table.

	Transaction_type_id	Transaction_type_description
1	422	Transfer: Move money from one account to another
2	423	Charge: Record a purchase on a debit card
3	424	Online: online a web-based store or online banking service
4	425	Transfer: Move money from one account to another
5	426	Charge: Record a purchase on a debit card
6	427	Online: online a web-based store or online banking service
7	428	Transfer: Move money from one account to another
8	429	Charge: Record a purchase on a debit card
9	430	Online: online a web-based store or online banking service
10	431	Online: online a web-based store or online banking service
11	432	Transfer: Move money from one account to another
12	433	Charge: Record a purchase on a debit card
13	434	Online: online a web-based store or online banking service
14	435	Transfer: Move money from one account to another
15	436	Charge: Record a purchase on a debit card
16	437	Online: online a web-based store or online banking service
17	438	Transfer: Move money from one account to another
18	439	Charge: Record a purchase on a debit card
19	440	Online: online a web-based store or online banking service

Figure-6.6: Result of DELETE statement for table Transaction_types

For the next step, DELETE was used to delete all rows from Transaction_types (Transaction_type_description) where Transaction_type_id is 421. Figure-6.6 shows result of DELETE statement for Transaction_types table.

	Account_id	Account_name	Opened_date	Account_type_id	Customer_id
1	672	Allen Beikay	20.04.2016 00:00:00	992	552
2	673	Aldiyar Makhan	29.09.2015 00:00:00	993	553
3	674	Zhanat Altynkyzy	15.10.2016 00:00:00	994	554
4	675	Kamshat Aspanova	04.04.2014 00:00:00	995	555
5	676	Daulet Kopbar	21.01.2015 00:00:00	996	556
6	677	Altyn Timurova	17.02.2017 00:00:00	997	557
7	678	Maksat Mashat	19.08.2013 00:00:00	998	558
8	679	Miras Sergazyuly	09.09.2014 00:00:00	999	559
9	680	Karlyqash Kanatbekova	01.12.2013 00:00:00	1000	560

Figure-6.7: Result of DELETE statement for table Accounts

For the next step, UPDATE was used to delete all rows from Accounts. So, in this step, Account_id is a foreign key in Transactions table, that's why we cannot delete it like in the previous example. We use UPDATE statement to give Account_id NULL value, and then we can use DELETE statement. Here, DELETE was used to delete all rows from Accounts table, where Account_id is 67. Figure-6.7 shows result of DELETE statement for Accounts table.

	Acquest_id	Acquest_date	Acquest_number	Customer_id	Services_id
1	852	10.05.2018 00:00:00	2	552	222
2	853	19.10.2016 00:00:00	3	553	223
3	854	25.11.2017 00:00:00	4	554	224
4	855	24.05.2015 00:00:00	5	555	225
5	856	11.02.2016 00:00:00	6	556	226
6	857	27.03.2018 00:00:00	7	557	227
7	858	29.09.2014 00:00:00	8	558	228
8	859	29.10.2015 00:00:00	9	559	229
9	860	11.01.2014 00:00:00	10	560	230

Figure-6.8: Result of DELETE statement for table Customer_acquests

For the next step, UPDATE was used to delete all rows from Customer_acquests. So, in this step, Acquest_id is a foreign key in Transactions table, that's why we cannot delete it like in the previous example. We use UPDATE statement to give Acquest_id NULL value, and then we can use DELETE statement. Here, DELETE was used to delete all rows from Customer_acquests table, where Acquest_id is 851. Figure-6.8 shows result of DELETE statement for Customer_acquests table.

	Customer_id	Customer_first_name	Customer_last_name	Customer_email	Customer_phone_number	registration_date	login	password	Gender	Customer_type_id
1	552	Allen	Beikay	allen@2mail.ru	87059020202	20.04.2016 00:00:00	allen9	96070	Male	322
2	553	Aldiyar	Makhan	aldiyar@3mail.ru	87059030303	29.09.2015 00:00:00	aldiyar8	56070	Male	323
3	554	Zhanat	Altynkyzy	zhanat@4mail.ru	87029040404	15.10.2016 00:00:00	zhanat7	12070	Female	324
4	555	Kamshat	Aspanova	kamshat@5mail.ru	87019050505	04.04.2014 00:00:00	kamshat6	23070	Female	325
5	556	Daulet	Kopbar	daulet@6mail.ru	87059060606	21.01.2015 00:00:00	daulet5	25070	Male	326
6	557	Altyn	Timurova	altyn@7mail.ru	87079070707	17.02.2017 00:00:00	altyn4	31070	Female	327
7	558	Maksat	Mashat	maksat@8mail.ru	87089080808	19.08.2013 00:00:00	maksat3	23070	Male	328
8	559	Miras	Sergazyuly	miras@9mail.ru	87019090909	09.09.2014 00:00:00	miras2	66070	Male	329
9	560	Karlyqash	Kanatbekova	karlyqash@12mail.ru	87018755700	01.12.2013 00:00:00	karlyqash1	17070	Female	330

Figure-6.9: Result of DELETE statement for table Customers

For the next step, DELETE was used to delete all rows from Customers where Customer_id is 551. Figure-6.9 shows result of DELETE statement for Customers table.

	Customer_type_id	Customer_type_description
1	322	Individual Entity: Citizens, foreign citizens and stateless persons
2	323	Corporate Client: Citizens that works in companies, organisations and etc.
3	324	Legal Entity: Individual entrepreneur
4	325	Individual Entity: Citizens, foreign citizens and stateless persons
5	326	Corporate Client: Citizens that works in companies, organisations and etc.
6	327	Legal Entity: Individual entrepreneur
7	328	Individual Entity: Citizens, foreign citizens and stateless persons
8	329	Corporate Client: Citizens that works in companies, organisations and etc.
9	330	Legal Entity: Individual entrepreneur

Figure-7: Result of DELETE statement for table Customer_types

For the next step, DELETE was used to delete all rows from Customer_types where Customer_type_id is 321. Figure-7 shows result of DELETE statement for Customer_types table.

	Services_id	Services_description	Employee_id
1	222	Personal Account Service	2212
2	223	Cyber Security Service	2222
3	224	Credit Service	6161
4	225	Credit Service	4444
5	226	Deposit Service	1111
6	227	Corporate Service	2212
7	228	Credit Service	2121
8	229	Currency Exchange Service	9191
9	230	Cash Collection Service	1212

Figure-7.1: Result of DELETE statement for table Products_services

For the next step, DELETE was used to delete all rows from Products_services where Services_id is 221. Figure-7.1 shows result of DELETE statement for Products_services table.

	Employee_id	Employee_first_name	Employee_last_name	Employee_phone_number	Employee_email	SSN	Salary	Gender	hire_date	Designation	Department_id
1	1111	Assem	Moldash	87010010101	assem@mail.ru	101	290000	Female	29.04.2015 00:00:00	Financier	111
2	1212	Karlyga	Akilova	87019755700	karla@12mail.ru	112	170000	Female	09.09.2014 00:00:00	Corporate Manager	119
3	1313	Asiya	Beimbetova	87010150607	asiya@13mail.ru	113	105000	Female	22.07.2018 00:00:00	Personnel Officer	210
4	2121	Alen	Zhunisbekov	87028765432	alen@mail.ru	114	230000	Male	29.06.2019 00:00:00	Risk Manager	101
5	2212	Lyazat	Saparova	87010000048	lyazat@22mail.ru	13	300000	Female	07.05.2015 00:00:00	Corporate Director	200
6	3131	Daniyar	Orazbayev	87054532345	daniyar@21mail.ru	115	470000	Male	19.03.2020 00:00:00	Main Auditor	102
7	3333	Madiyar	Kakharman	87050030303	madiyar@3mail.ru	303	500000	Male	29.09.2015 00:00:00	Engineer	107
8	4141	Zhuldyz	Kozhabayeva	87086300485	zhuldyz@31mail.ru	116	107000	Female	19.04.2019 00:00:00	Specialists assistant	103
9	4444	Zhanar	Alchickeyeva	87020040404	zhanar@4mail.ru	404	120000	Female	04.04.2014 00:00:00	Assistant	113
10	5151	Kamaliya	Sultankzy	87756334323	ksultante@5mail.ru	117	90000	Female	08.12.2018 00:00:00	Manager	105
11	5555	Kamilla	Ospanova	87010050505	kamilla@5mail.ru	505	230000	Female	15.10.2016 00:00:00	General Manager	114
12	6161	Dias	Amangeldin	87029007557	dias@61mail.ru	118	240000	Male	09.03.2020 00:00:00	Valuer	106
13	6666	Dauren	Kambar	87050060606	dauren@6mail.ru	606	250000	Male	19.08.2013 00:00:00	Administrator	115
14	7171	Alima	Nurlybekova	87056551209	alima@71mail.ru	119	200000	Female	19.08.2019 00:00:00	Placeman	104
15	7777	Alina	Timurkyzy	87070070707	alina@7mail.ru	707	310000	Female	21.01.2015 00:00:00	Auditor	116
16	8181	Mansur	Bisembay	87019008558	mansur@81mail.ru	11	105000	Male	03.04.2013 00:00:00	Personnel Officer	109
17	8888	Maksim	Melnikov	87080080808	maksim@9mail.ru	808	230000	Male	01.12.2013 00:00:00	Risk Manager	117
18	9191	Karim	Ermakhanov	87000991551	karim@91mail.ru	12	90000	Male	10.09.2017 00:00:00	Cashier	108
19	9999	Matvey	Stepanenko	87010090909	matvey@9mail.ru	909	600000	Male	17.02.2017 00:00:00	IT Specialist	118

Figure-7.2: Result of DELETE statement for table Employees

For the next step, UPDATE was used to delete all rows from Employees. So, in this step, Employee_id is a foreign key in Products_services table, that's why we cannot delete it like in the previous example. We use UPDATE statement to give Employee_id NULL value, and then we can use DELETE statement. Here, DELETE was used to delete all rows from Employees table, where Employee_id is 2222. Figure-7.2 shows result of DELETE statement for Employees table.

	Department_id	Department_name	Description	Branch_id
1	101	Risk Management Department	Manage the risk management process on day to day basis	11
2	102	Audit Department	Responsible for evaluating operational procedures,control functions	12
3	103	R&D Department	Responsible for finding knowledge to create new products	13
4	104	Branches Affairs	Responsible for working with branches of bank	14
5	105	International Affairs	Responsible for working with international organizations, governments, etc.	15
6	106	Credit Department	Analyze a credit request and suggest how to improve loan structure	16
7	107	IT Department	Introduce IT solutions for the bank departments	17
8	108	Financial Department	Work with customer finances	18
9	109	HR Department	Analyze future requirements	19
10	111	Financial Department	Work with customer finances	11
11	113	Credit Department	Analyze a credit request and suggest how to improve loan structure	13
12	114	International Affairs	Responsible for working with international organizations, governments, etc.	14
13	115	Branches Affairs	Responsible for working with branches of bank	15
14	116	Audit Department	Responsible for evaluating operational procedures,control functions	16
15	117	Risk Management Department	Manage the risk management process on day to day basis	17
16	118	R&D Department	Responsible for finding knowledge to create new products	18
17	119	Corporate Banking Department	Refers to the aspect of banking that deals with corporate customers	19
18	200	Corporate Banking Department	Refers to the aspect of banking that deals with corporate customers	20
19	210	HR Department	Analyze future requirements	20

Figure-7.3: Result of DELETE statement for table Departments

For the next step, DELETE was used to delete all rows from Departments where Department_id is 112. Figure-7.3 shows result of DELETE statement for Departments table.

	Branch_id	Branch_name	Branch_city	Branch_country	Branch_address	Bank_id
1	11	Kaspi Bank Branch	Nur-Sultan	Kazakhstan	Kunayev street,14/2	1
2	12	Halyk Bank Branch	Almaty	Kazakhstan	Altynsarın street,23	2
3	13	Qazqom Bank Branch	Shymkent	Kazakhstan	Ilyayeva street,33	3
4	14	Jysan Bank Branch	Almaty	Kazakhstan	Rozybakiyeva street,273	4
5	16	Sberbank Branch	Nur-Sultan	Kazakhstan	Dostyk street,9	6
6	17	ForteBank Branch	Shymkent	Kazakhstan	Zheltoksan street,15	7
7	18	RBK Bank Branch	Almaty	Kazakhstan	Sharipova street,84	8
8	19	ATF Bank Branch	Almaty	Kazakhstan	Tole bi street,83	9
9	20	VTB Bank Branch	Nur-Sultan	Kazakhstan	Dostyk street,12	10

Figure-7.4: Result of DELETE statement for table Branches

For the next step, UPDATE was used to delete all rows from Branches. So, in this step, Branch_id is a foreign key in Departments table, that's why we cannot delete it like in the previous example. We use UPDATE statement to give Branch_id NULL value, and then we can use DELETE statement. Here, DELETE was used to delete all rows from Branches table, where Branch_id is 15. Figure-7.4 shows result of DELETE statement for Branches table.

	Bank_id	Bank_name	Bank_country	Bank_city	Bank_address
1	2	Halyk Bank	Kazakhstan	Nur-Sultan	Kabanbai batyr street,17
2	3	Qazqom Bank	Kazakhstan	Almaty	Gagarin street,135/8
3	4	Jysan Bank	Kazakhstan	Nur-Sultan	Syganak street,24
4	5	Eurasian Bank	Kazakhstan	Nur-Sultan	Kabanbai Batyr street,30
5	6	Sberbank	Russia	Moscow	Vavilova street,19
6	7	ForteBank	Kazakhstan	Nur-Sultan	Dostyk street,8/1
7	8	RBK Bank	Kazakhstan	Nur-Sultan	Saraishyk street,11
8	9	ATF Bank	Kazakhstan	Nur-Sultan	Saraishyk street,34
9	10	VTB Bank	Russia	Moscow	Myasnitskaya street,35

Figure-7.5: Result of DELETE statement for table Bank

For the next step, UPDATE was used to delete all rows from Bank. So, in this step, Bank_id is a foreign key in Branches table, that's why we cannot delete it like in the previous example. We use UPDATE statement to give Bank_id NULL value, and then we can use DELETE statement. Here, DELETE was used to delete all rows from Bank table, where Bank_id is 01. Figure-7.5 shows result of DELETE statement for Bank table.

2.3. Queries

	Employee_id	Department_name
1	1111	Financial Department
2	1212	Corporate Banking Department
3	1313	HR Department
4	2121	Risk Management Department
5	2212	Corporate Banking Department
6	2222	IT Department
7	3131	Audit Department
8	3333	IT Department
9	4141	R&D Department
10	4444	Credit Department
11	5151	International Affairs
12	5555	International Affairs
13	6161	Credit Department
14	6666	Branches Affairs
15	7171	Branches Affairs
16	7777	Audit Department
17	8181	HR Department
18	8888	Risk Management Department
19	9191	Financial Department
20	9999	R&D Department

Figure-7.6: Result of INNER JOIN query

We used INNER JOIN keyword, selected from Employees and Departments tables, Employee_id from Employees table and Department_name from Departments table, we used ON operator to select table together. Show result of INNER JOIN keyword from Employees and Departments tables. Figure-7.6 shows result of INNER JOIN query.

	Customer_first_name	Customer_last_name	Customer_type_id
1	Aldiyar	Makhan	323
2	Allen	Beikay	322
3	Altyn	Timurova	327
4	Assel	Butovand	321
5	Daulet	Kopbar	326
6	Kamshat	Aspanova	325
7	Karlygash	Kanatbekova	330
8	Maksat	Mashat	328
9	Miras	Sergazyuly	329
10	Zhanat	Altynkyzy	324

Figure-7.7: Result of LEFT JOIN query

The LEFT JOIN keyword returns all records from the Customers, and the matched records from the Customer_types. We used ON keyword to create a relation between two tables. We used ORDER BY to select order of showing tables. Figure-7.7 shows result of LEFT JOIN query.

	Transaction_id	Transaction_date	Transaction_amount	Account_id
1	961	01.04.2020 00:00:00	20000	671
2	962	09.09.2020 00:00:00	15000	672
3	963	19.08.2020 00:00:00	5000	673
4	964	17.10.2019 00:00:00	700000	674
5	965	21.01.2018 00:00:00	220000	675
6	966	04.04.2017 00:00:00	400000	676
7	967	15.10.2020 00:00:00	100000	677
8	968	29.09.2020 00:00:00	30000	678
9	969	20.04.2020 00:00:00	2990	679
10	970	29.04.2019 00:00:00	100000	680
11	971	19.04.2019 00:00:00	200000	671
12	972	10.04.2020 00:00:00	1500	672
13	973	09.09.2020 00:00:00	2000	673
14	974	05.10.2020 00:00:00	70000	674
15	975	14.04.2016 00:00:00	170000	675
16	976	11.01.2017 00:00:00	200000	676
17	977	17.10.2020 00:00:00	200000	677
18	978	29.08.2020 00:00:00	20000	678
19	979	20.05.2020 00:00:00	990	679
20	980	29.03.2019 00:00:00	150000	680

Figure-7.8: Result of RIGHT JOIN query

The RIGHT JOIN keyword returns all records from the right table Accounts, and the matched records from the left table Transactions. We used ON keyword to create a relation between two tables. We used ORDER BY to select order of showing tables. Figure-7.8 shows result of RIGHT JOIN query.

	Branch_name	Branch_city	Branch_country	Bank_id
1	ATF Bank Branch	Almaty	Kazakhstan	9
2	Eurasian Bank Branch	Shymkent	Kazakhstan	5
3	ForteBank Branch	Shymkent	Kazakhstan	7
4	Halyk Bank Branch	Almaty	Kazakhstan	2
5	Jysan Bank Branch	Almaty	Kazakhstan	4
6	Kaspi Bank Branch	Nur-Sultan	Kazakhstan	1
7	Qazqom Bank Branch	Shymkent	Kazakhstan	3
8	RBK Bank Branch	Almaty	Kazakhstan	8
9	Sberbank Branch	Nur-Sultan	Kazakhstan	6
10	VTB Bank Branch	Nur-Sultan	Kazakhstan	10

Figure-7.9: Result FULL OUTER JOIN of query

The FULL OUTER JOIN keyword returns all records when there is a match in left Branches or right Bank table records. We used FULL OUTER JOIN can potentially return very large result-sets! We select tables from Branches, name, city and country. Also, from Bank table choose Bank_id. We used ORDER BY to select order of showing tables. Figure-7.9 shows result of FULL OUTER JOIN query.

	SmallestSalary
1	90000

Figure-8: Result of MIN query

We used MIN function returns the smallest value of the selected column (Salary) and used AS keyword to add new column to show a minimum amount table from Employees. Figure-8 shows result of MIN query.

	LargestAmount
1	700000

Figure-8.1: Result of MAX query

We used MAX function returns the largest value of the selected column Transaction_amount, and used AS keyword to add new column to show a maximum amount table from Transactions. Figure-8.1 shows result of MAX query.

	Num_Salary
1	20

Figure-8.2: Result of COUNT query

	AverageOfTransactionAmount
1	130374

Figure-8.3: Result of AVG query

We used AVG function returns the average of transaction_amount. We used AVG to calculate average transaction amount from Transactions, and used AS keyword to add new column name is Average Of Transaction Amount. Figure-8.3 shows result of AVG query.

	Total_of_salary
1	4457000

Figure-8.4: Result of SUM query

The SUM function returns the total sum of a numeric column. We used SUM function to add full amount biggest than 100000 from Employees table and used AS keyword to add new column named to Total_of_salary . We also used “>” function to determine the quantity change. Figure-8.4 shows result of SUM query.

	Bank_id	Bank_name	Bank_country	Bank_city	Bank_address
1	1	Kaspi Bank	Kazakhstan	Almaty	Nauryzbai batyr street, 154
2	6	Sberbank	Russia	Moscow	Vavilova street, 19

Figure-8.5: Result of OR query

We used the OR command with WHERE to include rows where either condition is true. The following SQL statement selects all fields from "Bank" where Bank_name is 'Kaspi Bank' or 'Sberbank'. Figure-8.5 shows result of OR query.

	Customer_id	Customer_first_name	Customer_last_name	Customer_email	Customer_phone_number	registration_date	login	password	Gender	Customer_type_id
1	557	Altyn	Timurova	altyn@7mail.ru	87079070707	17.02.2017 00:00:00	altyn4	31070	Female	327
2	560	Karlygash	Kanatbekova	karlygash@12mail.ru	87018755700	01.12.2013 00:00:00	karlygash1	17070	Female	330

Figure-8.6: Result of AND query

We selected from Customers table to create new table include value from gender only ‘female’. We also used AND keyword to add other variable Customer_id largest than 555 , that used “>” function. Figure-8.6 shows result of AND query.

2.4. Subqueries

	Branch_name	Branch_id	Branch_city	Branch_country
1	Kaspi Bank Branch	11	Nur-Sultan	Kazakhstan

Figure-9: Result of single-row subquery operation for Branches table

A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved. For the first step, we use single-row subquery for Branches table that shows Branch_name, Branch_id, Branch_city, Branch_country where Branch_id is a Branch_id from the query where Branch_name is 'Kaspi Bank Branch'. Figure 9 shows result of single-row subquery operation for Branches table.

	Transaction_id	Transaction_date	Transaction_amount	Account_id	Acquest_id	Transaction_type_id
1	964	17.10.2019 00:00:00	700000	674	854	424
2	965	21.01.2018 00:00:00	220000	675	855	425
3	966	04.04.2017 00:00:00	400000	676	856	426
4	971	19.04.2019 00:00:00	200000	671	851	431
5	976	11.01.2017 00:00:00	200000	676	856	436
6	977	17.10.2020 00:00:00	200000	677	857	437

Figure-9.1: Result of single-row subquery operation for Transactions table

For the next step, we use single-row subquery for Transactions table that shows Transaction_id, Transaction_date, Transaction_amount, Account_id, Acquest_id, Transaction_type_id where Transaction_amount greater than Transaction_amount from the query where Transaction_date is '14-APR-2016'. Figure-9.1 shows result of single-row subquery operation for Transactions table.

	Transaction_id	Transaction_date	Transaction_amount	Account_id	Acquest_id	Transaction_type_id
1	961	01.04.2020 00:00:00	20000	671	851	421
2	962	09.09.2020 00:00:00	15000	672	852	422
3	963	19.08.2020 00:00:00	5000	673	853	423
4	967	15.10.2020 00:00:00	100000	677	857	427
5	968	29.09.2020 00:00:00	30000	678	858	428
6	969	20.04.2020 00:00:00	2990	679	859	429
7	970	29.04.2019 00:00:00	100000	680	860	430
8	972	10.04.2020 00:00:00	1500	672	852	432
9	973	09.09.2020 00:00:00	2000	673	853	433
10	974	05.10.2020 00:00:00	70000	674	854	434
11	978	29.08.2020 00:00:00	20000	678	858	438
12	979	20.05.2020 00:00:00	990	679	859	439
13	980	29.03.2019 00:00:00	150000	680	860	440

Figure-9.2: Result of single-row subquery operation for Transactions table

For the next step, we use single-row subquery for Transactions table that shows Transaction_id, Transaction_date, Transaction_amount, Account_id, Acquest_id, Transaction_type_id where Transaction_amount less than Transaction_amount from the query where Transaction_date is '14-APR-2016'. Figure-9.2 shows result of single-row subquery operation for Transactions table.

	Employee_id	Employee_first_name	Employee_last_name	Salary
1	1212	Karlyga	Akilova	170000
2	1313	Asiya	Beimbetova	105000
3	2121	Alen	Zhunisbekov	230000
4	2212	Lyazat	Saparova	300000
5	2222	Alem	Beisenbai	90000
6	3131	Daniyar	Orazbayev	470000
7	3333	Madiyar	Kakharman	500000
8	4141	Zhuldyz	Kozhabayeva	107000
9	4444	Zhanar	Alchickeyeva	120000
10	5151	Kamaliya	Sultankzyzy	90000
11	5555	Kamilla	Ospanova	230000
12	6161	Dias	Amangeldin	240000
13	6666	Dauren	Kambar	250000
14	7171	Alima	Nurlybekova	200000
15	7777	Alina	Timurkyzy	310000
16	8181	Mansur	Bisembay	105000
17	8888	Maksim	Melnikov	230000
18	9191	Karim	Ermakhanov	90000
19	9999	Matvey	Stepanenko	600000

Figure-9.3: Result of single-row subquery operation for Employees table

For the next step, we use single-row subquery for Employees table that shows Employee_id, Employee_first_name, Employee_last_name, Salary where Employee_id is not Employee_id from the query where Salary is '290000'. Figure-9.3 shows result of single-row subquery operation for Employees table.

	Bank_name	Bank_id	Bank_city	Bank_country
1	Sberbank	6	Moscow	Russia
2	VTB Bank	10	Moscow	Russia

Figure-9.4: Result of multiple-row subquery operation for Bank table

Multiple-row subqueries returns multiple rows, it must be handled by set comparison operators (IN, ALL, ANY). For this step we use multiple-row subquery for Bank table that shows Bank_name, Bank_id, Bank_city, Bank_country where Bank_id is in the query where Bank_country is 'Russia'. Figure-9.4 shows result of multiple-row subquery operation for Bank table.

	Bank_name	Bank_id	Bank_city	Bank_country
1	Kaspi Bank	1	Almaty	Kazakhstan
2	Halyk Bank	2	Nur-Sultan	Kazakhstan
3	Qazqom Bank	3	Almaty	Kazakhstan
4	Jysan Bank	4	Nur-Sultan	Kazakhstan
5	Eurasian Bank	5	Nur-Sultan	Kazakhstan
6	ForteBank	7	Nur-Sultan	Kazakhstan
7	RBK Bank	8	Nur-Sultan	Kazakhstan
8	ATF Bank	9	Nur-Sultan	Kazakhstan

Figure-9.5: Result of multiple-row subquery operation for Bank table

For the next step we use multiple-row subquery for Bank table that shows Bank_name, Bank_id, Bank_city, Bank_country where Bank_id is not in the query where Bank_country is 'Russia'. Figure-9.5 shows result of multiple-row subquery operation for Bank table.

	Customer_id	Customer_first_name	Customer_last_name	Customer_type_id
1	551	Assel	Butovand	321
2	554	Zhanat	Altynkyzy	324
3	555	Kamshat	Aspanova	325
4	557	Altyn	Timurova	327
5	560	Karlygash	Kanatbekova	330

Figure-9.6: Result of multiple-row subquery operation for Customers table

For the next step we use multiple-row subquery for Customers table that shows Customer_id, Customer_first_name, Customer_last_name, Customer_type_id where Customer_id can be any Customer_id in the query where Gender is 'Female'. Figure-9.6 shows result of multiple-row subquery operation for Customers table.

Department_id	Department_name	Description	Branch_id
1	102	Audit Department Responsible for evaluating operational procedures, control functions	12
2	112	IT Department Introduce IT solutions for the bank departments and responsible for IT security	12

Figure-9.7: Result of multiple-row subquery operation for Departments table

For the next step we use multiple-row subquery for Departments table that shows Department_id, Department_name, Description, Branch_id where Department_id in the query where Branch_id is 12. Figure-9.6 shows result of multiple-row subquery operation for Departments table.

	Employee_id	Employee_first_name	Employee_last_name	Department_id
1	1111	Assem	Moldash	111

Figure-9.8: Result of multiple-column subquery operation for Employees table

For the next step we use multiple-row subquery for Employees table that shows Employee_id, Employee_first_name, Employee_last_name, Department_id in the query where Department_id and Employee_id is in (1111, 111). Figure-9.8 shows result of multiple- column subquery operation for Employees table.

Appendix A

Figure-2: Process of creating database

```
CREATE TABLE Bank(
    Bank_id INT NOT NULL,
    Bank_name VARCHAR(50) NOT NULL,
    Bank_country VARCHAR(50) NOT NULL,
    Bank_city VARCHAR(50) NOT NULL,
    Bank_address CHAR(50) NOT NULL,
    PRIMARY KEY(Bank_id)
);

CREATE TABLE Branches(
    Branch_id INT NOT NULL,
    Branch_name VARCHAR(50) NOT NULL,
    Branch_city VARCHAR(50) NOT NULL,
    Branch_country VARCHAR(50) NOT NULL,
    Branch_address CHAR(50) NOT NULL,
    Bank_id INT,
    FOREIGN KEY (Bank_id) REFERENCES Bank(Bank_id),
    PRIMARY KEY(Branch_id)
);

CREATE TABLE Departments(
    Department_id DECIMAL(12) NOT NULL,
    Department_name VARCHAR(50) NOT NULL,
    Description TEXT NOT NULL,
    Branch_id INT,
    FOREIGN KEY (Branch_id) REFERENCES Branches(Branch_id),
    PRIMARY KEY(Department_id)
);

CREATE TABLE Employees(
    Employee_id INT NOT NULL,
    Employee_first_name VARCHAR(50) NOT NULL,
    Employee_last_name VARCHAR(50) NOT NULL,
    Employee_phone_number DECIMAL(12) NOT NULL,
    Employee_email CHAR(50) NOT NULL,
    SSN DECIMAL(12) NOT NULL,
    Salary DECIMAL(12) NOT NULL,
    Gender TEXT NOT NULL,
    hire_date DATE NOT NULL,
    Designation TEXT NOT NULL,
    Department_id DECIMAL(12) NOT NULL,
    PRIMARY KEY(Employee_id),
    FOREIGN KEY (Department_id) REFERENCES Departments(Department_id)
);

CREATE TABLE Products_services(
    Services_id VARCHAR(15) NOT NULL,
    Services_description TEXT NOT NULL,
    Employee_id INT,
    PRIMARY KEY(Services_id),
```

```

        FOREIGN KEY (Branch_id) REFERENCES Branches(Branch_id),
        PRIMARY KEY(Department_id)
    );
CREATE TABLE Employees(
    Employee_id INT NOT NULL,
    Employee_first_name VARCHAR(50) NOT NULL,
    Employee_last_name VARCHAR(50) NOT NULL,
    Employee_phone_number DECIMAL(12) NOT NULL,
    Employee_email CHAR(50) NOT NULL,
    SSN DECIMAL(12) NOT NULL,
    Salary DECIMAL(12) NOT NULL,
    Gender TEXT NOT NULL,
    hire_date DATE NOT NULL,
    Designation TEXT NOT NULL,
    Department_id DECIMAL(12) NOT NULL,
    PRIMARY KEY(Employee_id),
    FOREIGN KEY (Department_id) REFERENCES Departments(Department_id)
);
CREATE TABLE Products_services(
    Services_id VARCHAR(15) NOT NULL,
    Services_description TEXT NOT NULL,
    Employee_id INT,
    PRIMARY KEY(Services_id),
    FOREIGN KEY (Employee_id) REFERENCES Employees(Employee_id)
);
CREATE TABLE Customer_types (
    Customer_type_id VARCHAR(50) NOT NULL,
    Customer_type_description TEXT NOT NULL,
    PRIMARY KEY(Customer_type_id)
);
CREATE TABLE Customers(
    Customer_id INT NOT NULL,
    Customer_first_name VARCHAR(25) NOT NULL,
    Customer_last_name VARCHAR(25) NOT NULL,
    Customer_email CHAR (50) NOT NULL,
    Customer_phone_number VARCHAR(50) NOT NULL,
    registration_date DATE NOT NULL,
    login VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL,
    Gender VARCHAR(50) NOT NULL,
    Customer_type_id VARCHAR(50),
    PRIMARY KEY(Customer_id),
    FOREIGN KEY (Customer_type_id) REFERENCES Customer_types (Customer_type_id)
);
CREATE TABLE Customer_acquests(

```

```

Acquest_id INT NOT NULL,
Acquest_date DATE NOT NULL,
Acquest_number FLOAT(10) NOT NULL,
Customer_id INT,
Services_id VARCHAR(15),
PRIMARY KEY(Acquest_id),
FOREIGN KEY (Customer_id) REFERENCES Customers (Customer_id),
FOREIGN KEY (Services_id) REFERENCES Products_services (Services_id)
);
CREATE TABLE Account_types (
Account_type_id VARCHAR(15) NOT NULL,
Account_type_description TEXT NOT NULL,
PRIMARY KEY(Account_type_id)
);
CREATE TABLE Accounts(
Account_id INT NOT NULL,
Account_name VARCHAR(50) NOT NULL,
Opened_date DATE NOT NULL,
Account_type_id VARCHAR(15) NOT NULL,
Customer_id INT NOT NULL,
PRIMARY KEY(Account_id),
FOREIGN KEY (Account_type_id) REFERENCES Account_types (Account_type_id),
FOREIGN KEY (Customer_id) REFERENCES Customers (Customer_id)
);
CREATE TABLE Transaction_types (
Transaction_type_id VARCHAR(15) NOT NULL,
Transaction_type_description TEXT NOT NULL,
PRIMARY KEY(Transaction_type_id)
);
CREATE TABLE Transactions(
Transaction_id INT NOT NULL,
Transaction_date DATE NOT NULL,
Transaction_amount FLOAT(10) NOT NULL,
Acquest_id INT,
Account_id INT,
Transaction_type_id VARCHAR(15),
PRIMARY KEY(Transaction_id),
FOREIGN KEY (Acquest_id) REFERENCES Customer_acquests(Acquest_id),
FOREIGN KEY (Account_id) REFERENCES Accounts (Account_id),
FOREIGN KEY (Transaction_type_id)
REFERENCES Transaction_types (Transaction_type_id)
);

```

Appendix B

INSERT INTO Statement:

```

INSERT INTO Bank(Bank_id, Bank_name, Bank_country, Bank_city, Bank_address)
VALUES
(01,'Kaspi Bank', 'Kazakhstan', 'Almaty', 'Nauryzbai batyr street,154'),
(02,'Halyk Bank', 'Kazakhstan', 'Nur-Sultan', 'Kabanbai batyr street,17'),
(03,'Qazqom Bank', 'Kazakhstan', 'Almaty', 'Gagarin street,135/8'),
(04,'Jysan Bank', 'Kazakhstan', 'Nur-Sultan', 'Syганак street,24'),
(05,'Eurasian Bank', 'Kazakhstan', 'Nur-Sultan', 'Kabanbai Batyr street,30'),
(06,'Sberbank', 'Russia', 'Moscow', 'Vavilova street,19'),
(07,'ForteBank', 'Kazakhstan', 'Nur-Sultan', 'Dostyk street,8/1'),
(08,'RBK Bank', 'Kazakhstan', 'Nur-Sultan', 'Saraishyk street,11'),
(09,'ATF Bank', 'Kazakhstan', 'Nur-Sultan', 'Saraishyk street,34'),
(10,'VTB Bank', 'Russia', 'Moscow', 'Myasnitskaya street,35');

INSERT INTO Branches(Branch_id, Branch_name, Branch_country, Branch_city, Branch_address, Bank_id)
VALUES
(11,'Kaspi Bank Branch', 'Kazakhstan', 'Nur-Sultan', 'Kunayev street,14/2',01),
(12,'Halyk Bank Branch', 'Kazakhstan', 'Almaty', 'Altynsarın street,23',02),
(13,'Qazqom Bank Branch', 'Kazakhstan', 'Shymkent', 'Ilyayeva street,33',03),
(14,'Jysan Bank Branch', 'Kazakhstan', 'Almaty', 'Rozybakiyeva street,273',04),
(15,'Eurasian Bank Branch', 'Kazakhstan', 'Shymkent', 'Tashenov street,4',05),
(16,'Sberbank Branch', 'Kazakhstan', 'Nur-Sultan', 'Dostyk street,9',06),
(17,'ForteBank Branch', 'Kazakhstan', 'Shymkent', 'Zheltoksan street,15',07),
(18,'RBK Bank Branch', 'Kazakhstan', 'Almaty', 'Sharipova street,84',08),
(19,'ATF Bank Branch', 'Kazakhstan', 'Almaty', 'Tole bi street,83',09),
(20,'VTB Bank Branch', 'Kazakhstan', 'Nur-Sultan', 'Dostyk street,12',10);

INSERT INTO Departments(Department_id, Department_name, Description, Branch_id)
VALUES
(111,'Financial Department','Work with customer finances',11),
(112,'IT Department','Introduce IT solutions for the bank departments and responsible for IT security',12),
(113,'Credit Department','Analyze a credit request and suggest how to improve loan structure',13),
(114,'International Affairs','Responsible for working with international organizations, governments, etc.',14),
(115,'Branches Affairs','Responsible for working with branches of bank',15),
(116,'Audit Department','Responsible for evaluating operational procedures,control functions',16),
(117,'Risk Management Department','Manage the risk management process on day to day basis',17),
(118,'R&D Department','Responsible for finding knowledge to create new products',18),
(119,'Corporate Banking Department','Refers to the aspect of banking that deals with corporate customers',19),
(210,'HR Department','Analyze future requirements',20),
(101,'Risk Management Department','Manage the risk management process on day to day basis',11),
(102,'Audit Department','Responsible for evaluating operational procedures,control functions',12),
(103,'R&D Department','Responsible for finding knowledge to create new products',13),
(104,'Branches Affairs','Responsible for working with branches of bank',14),
(105,'International Affairs','Responsible for working with international organizations, governments, etc.',15),
(106,'Credit Department','Analyze a credit request and suggest how to improve loan structure',16),
(107,'IT Department','Introduce IT solutions for the bank departments',17),
(108,'Financial Department','Work with customer finances',18),
(109,'HR Department','Analyze future requirements',19),
(200,'Corporate Banking Department','Refers to the aspect of banking that deals with corporate customers',20);

INSERT INTO Employees(Employee_id, Employee_first_name, Employee_last_name, Employee_phone_number,
Employee_email,SSN,Salary, Gender,hire_date,Designation,Department_id)
VALUES
(1111,'Assem','Moldash','87010010101','assem@mail.ru','00101','290000','Female','29-APR-2015','Financier',111),
(2222,'Alem','Beisenbai','87070020202','alem@mail.ru','00202','90000','Male','20-APR-2016','IT Manager',112),
(3333,'Madiyar','Kakharman','87050030303','madiyar@mail.ru','00303','500000','Male','29-SEP-2015','Engineer',107),
(4444,'Zhanar','Alchickeyeva','87020040404','zhanar@mail.ru','00404','120000','Female','04-APR-2014','Assistant',113),
(5555,'Kamilla','Ospanova','87010050505','kamilla@mail.ru','00505','230000','Female','15-OCT-2016','General Manager',114),
(6666,'Dauren','Kambar','87050060606','dauren@mail.ru','00606','250000','Male','19-AUG-2013','Administrator',115),
(7777,'Alina','Timurkyzy','87070070707','alina@mail.ru','00707','310000','Female','21-JAN-2015','Auditor',116),
(8888,'Maksim','Melnikov','87080080808','maksim@mail.ru','00808','230000','Male','01-DEC-2013','Risk Manager',117),
(9999,'Matvey','Stepanenko','87010090909','matvey@mail.ru','00909','600000','Male','17-FEB-2017','IT Specialist',118),
(1212,'Karlyga','Akilova','87019755700','karla@mail.ru','00112','170000','Female','09-SEP-2014','Corporate Manager',119),
(1313,'Asiya','Beimbetova','87010150607','asiya@mail.ru','00113','105000','Female','22-JUL-2018','Personnel Officer',210),
(2121,'Alen','Zhunisbekov','87028765432','alen@mail.ru','00114','230000','Male','29-JUN-2019','Risk Manager',101),
(3131,'Daniyar','Orazbayev','87054532345','daniyar@mail.ru','00115','470000','Male','19-MAR-2020','Main Auditor',102),
(4141,'Zhuldyz','Kozhabayeva','87086300485','zhuldyz@mail.ru','00116','107000','Female','19-APR-2019','Specialists assistant',103),
(5151,'Kamaliya','Sultankazy','87756334323','ksultant@mail.ru','00117','90000','Female','08-DEC-2018','Manager',105),
(6161,'Dias','Amangeldin','87029007557','dias@mail.ru','00118','240000','Male','09-MAR-2020','Valuer',106),
(7171,'Alima','Nurlybekova','87056551209','alima@mail.ru','00119','200000','Female','19-AUG-2019','Placeman',104),
(8181,'Mansur','Bisembay','87019008558','mansur@mail.ru','00011','105000','Male','03-APR-2013','Personnel Officer',109),
(9191,'Karim','Ermakhanov','87000991551','karim@mail.ru','00012','90000','Male','10-SEP-2017','Cashier',108),

```

```

(2212,'Lyazat','Saparova','87010000048','lyazat@22mail.ru','00013','300000','Female','07-MAY-2015','Corporate Director',200);
INSERT INTO Products_services(Services_id, Services_description, Employee_id)
VALUES
(221,'Personal Account Service',1111),
(222,'Personal Account Service',2212),
(223,'Cyber Security Service',2222),
(224,'Credit Service',6161),
(225,'Credit Service',4444),
(226,'Deposit Service',1111),
(227,'Corporate Service',2212),
(228,'Credit Service',2121),
(229,'Currency Exchange Service',9191),
(230,'Cash Collection Service',1212);
INSERT INTO Customer_types(Customer_type_id,Customer_type_description)
VALUES
(321,'Legal Entity: Individual entrepreneur'),
(322,'Individual Entity: Citizens, foreign citizens and stateless persons'),
(323,'Corporate Client: Citizens that works in companies, organisations and etc.'),
(324,'Legal Entity: Individual entrepreneur'),
(325,'Individual Entity: Citizens, foreign citizens and stateless persons'),
(326,'Corporate Client: Citizens that works in companies, organisations and etc.'),
(327,'Legal Entity: Individual entrepreneur'),
(328,'Individual Entity: Citizens, foreign citizens and stateless persons'),
(329,'Corporate Client: Citizens that works in companies, organisations and etc.'),
(330,'Legal Entity: Individual entrepreneur');
INSERT INTO Customers (Customer_id, Customer_first_name, Customer_last_name, Customer_phone_number, Customer_email,
registration_date, login,password, Gender, Customer_type_id)
VALUES
(551,'Assel','Butovand','87059010101','asse1@mail.ru','29-APR-2015','asse10','Female',321),
(552,'Allen','Beikay','87059020202','allen2@mail.ru','20-APR-2016','allen9','Male',322),
(553,'Aldiyar','Makhan','87059030303','aldiyar3@mail.ru','29-SEP-2015','aldiyar8','56070','Male',323),
(554,'Zhanat','Altynkyzy','87029040404','zhanat8@mail.ru','15-OCT-2016','zhanat7','12070','Female',324),
(555,'Kamshat','Aspanova','87019050505','kamshat5@mail.ru','04-APR-2014','kamshat6','23070','Female',325),
(556,'Daulet','Kopbar','87059060606','daulet6@mail.ru','21-JAN-2015','daulet5','25070','Male',326),
(557,'Altyn','Timurova','87079070707','altyn7@mail.ru','17-FEB-2017','altyn4','31070','Female',327),
(558,'Maksat','Mashat','87089080808','maksat8@mail.ru','19-AUG-2013','maksat3','23070','Male',328),
(559,'Miras','Sergazyuly','87019090909','miras9@mail.ru','09-SEP-2014','miras2','66070','Male',329),
(560,'Karlygash','Kanatbekova','87018755700','karlygash12@mail.ru','01-DEC-2013','karlygash1','17070','Female',330);
INSERT INTO Customer_acquests(Acquest_id, Acquest_date, Acquest_number, Customer_id, Services_id)
VALUES
(851,'19-MAY-2020','0001',551,221),
(852,'10-MAY-2018','0002',552,222),
(853,'19-OCT-2016','0003',553,223),
(854,'25-NOV-2017','0004',554,224),
(855,'24-MAY-2015','0005',555,225),
(856,'11-FEB-2016','0006',556,226),
(857,'27-MAR-2018','0007',557,227),
(858,'29-SEP-2014','0008',558,228),
(859,'29-OCT-2015','0009',559,229),
(860,'11-JAN-2014','0010',560,230);
INSERT INTO Account_types(Account_type_id, Account_type_description)
VALUES
(991,'Legal Entity Account'),
(992,'Individual Entity Account'),
(993,'Corporate Client Account'),
(994,'Legal Entity Account'),
(995,'Individual Entity Account'),
(996,'Corporate Client Account'),
(997,'Legal Entity Account'),
(998,'Individual Entity Account'),
(999,'Corporate Client Account'),
(1000,'Legal Entity Account');
INSERT INTO Accounts (Account_id, Account_name, Opened_date, Account_type_id, Customer_id)
VALUES
(671,'Assel Butovand','29-APR-2015',991,551),
(672,'Allen Beikay','20-APR-2016',992,552),
(673,'Aldiyar Makhan','29-SEP-2015',993,553),
(674,'Zhanat Altynkyzy','15-OCT-2016',994,554),
(675,'Kamshat Aspanova','04-APR-2014',995,555),

```

```

(676,'Daulet Kopbar','21-JAN-2015',996,556),
(677,'Altyn Timurova','17-FEB-2017',997,557),
(678,'Maksat Mashat','19-AUG-2013',998,558),
(679,'Miras Sergazyuly','09-SEP-2014',999,559),
(680,'Karlygash Kanatbekova','01-DEC-2013',1000,560);
INSERT INTO Transaction_types(Transaction_type_id, Transaction_type_description)
VALUES
(421,'Online: online a web-based store or online banking service'),
(422,'Transfer: Move money from one account to another'),
(423,'Charge: Record a purchase on a debit card'),
(424,'Online: online a web-based store or online banking service'),
(425,'Transfer: Move money from one account to another'),
(426,'Charge: Record a purchase on a debit card'),
(427,'Online: online a web-based store or online banking service'),
(428,'Transfer: Move money from one account to another'),
(429,'Charge: Record a purchase on a debit card'),
(430,'Online: online a web-based store or online banking service'),
(431,'Online: online a web-based store or online banking service'),
(432,'Transfer: Move money from one account to another'),
(433,'Charge: Record a purchase on a debit card'),
(434,'Online: online a web-based store or online banking service'),
(435,'Transfer: Move money from one account to another'),
(436,'Charge: Record a purchase on a debit card'),
(437,'Online: online a web-based store or online banking service'),
(438,'Transfer: Move money from one account to another'),
(439,'Charge: Record a purchase on a debit card'),
(440,'Online: online a web-based store or online banking service');
INSERT INTO Transactions(Transaction_id, Transaction_date, Transaction_amount, Account_id, Transaction_type_id)
VALUES
(961,'01-APR-2020','20000',671,851,421),
(962,'09-SEP-2020','15000',672,852,422),
(963,'19-AUG-2020','5000',673,853,423),
(964,'17-OCT-2019','700000',674,854,424),
(965,'21-JAN-2018','220000',675,855,425),
(966,'04-APR-2017','400000',676,856,426),
(967,'15-OCT-2020','1000000',677,857,427),
(968,'29-SEP-2020','30000',678,858,428),
(969,'20-APR-2020','2990',679,859,429),
(970,'29-APR-2019','100000',680,860,430),
(971,'19-APR-2019','200000',671,851,431),
(972,'10-APR-2020','1500',672,852,432),
(973,'09-SEP-2020','2000',673,853,433),
(974,'05-OCT-2020','70000',674,854,434),
(975,'14-APR-2016','170000',675,855,435),
(976,'11-JAN-2017','200000',676,856,436),
(977,'17-OCT-2020','200000',677,857,437),
(978,'29-AUG-2020','20000',678,858,438),
(979,'20-MAY-2020','990',679,859,439),
(980,'29-MAR-2019','150000',680,860,440);

```

UPDATE Statement:

```

UPDATE Bank
SET Bank_name='Kassa Nova Bank',Bank_country='Kazakhstan',
Bank_city='Nur-Sultan', Bank_address='Bauyrzhan Momyshuly street,22/14'
WHERE Bank_id=06;
SELECT *FROM Bank;

UPDATE Branches
SET Branch_name='Kassa Nova Branch',Branch_city='Almaty',
Branch_address='Altynsarın street, 27'
WHERE Branch_id=16;
SELECT *FROM Branches;

UPDATE Departments
SET Department_name='Marketing Department',
Description='Provides the necessary research to identify your target customers and other audiences.'
WHERE Department_id=114;
SELECT *FROM Departments;

UPDATE Employees
SET Employee_first_name='Azat', Employee_last_name='Janmolda',
Employee_phone_number='87050150505',Employee_email='azatl@mail.ru',
Salary='300000', Gender='Male',hire_date='20-AUG-2019',
Designation='Personnel Officer',Department_id=109

```

```

WHERE Employee_id=1111;
SELECT *FROM Employees;

UPDATE Products_services
SET Services_description='Credit Service', Employee_id=8888
WHERE Services_id=223;
SELECT *FROM Products_services;

UPDATE Customer_types
SET Customer_type_description='Individual Entity: Citizens, foreign citizens and stateless persons'
WHERE Customer_type_id=321;
SELECT *FROM Customer_types;

UPDATE Customers
SET Customer_first_name='Damir', Customer_last_name='Shokhan',
Customer_phone_number='87015514334', Customer_email='damirl@mail.ru',
registration_date='22-JUL-2019', login='damirl', password='04070', Gender='Male'
WHERE Customer_id=551;
SELECT *FROM Customers;

UPDATE Customer_acquests
SET Acquest_date='20-MAY-2019', Acquest_number='0020', Services_id=222
WHERE Acquest_id=851;
SELECT *FROM Customer_acquests;

UPDATE Account_types
SET Account_type_description='Individual Entity Account'
WHERE Account_type_id=991;
SELECT *FROM Account_types;

UPDATE Accounts
SET Account_name='Damir Shokhan', Opened_date='22-JUL-2019', Account_type_id=992
WHERE Account_id=671;
SELECT *FROM Accounts;

UPDATE Transaction_types
SET Transaction_type_description='Transfer: Move money from one account to another'
WHERE Transaction_type_id=421;
SELECT *FROM Transaction_types;

UPDATE Transactions
SET Transaction_date='09-AUG-2020', Transaction_amount='25000',
Account_id=672, Acquest_id=852, Transaction_type_id=421
WHERE Transaction_id=961;
SELECT *FROM Transactions;

```

DELETE Statement:

```
DELETE FROM Transactions
WHERE Transaction_id=961;
SELECT *FROM Transactions;

DELETE FROM Transaction_types
WHERE Transaction_type_id=421;
SELECT *FROM Transaction_types;

UPDATE Transactions SET Account_id=NULL;
UPDATE Transactions SET Acquest_id=NULL;

DELETE FROM Accounts
WHERE Account_id=671;
SELECT *FROM Accounts;

DELETE FROM Customer_acquests
WHERE Acquest_id=851;
SELECT *FROM Customer_acquests;

DELETE FROM Customers
WHERE Customer_id=551;
SELECT *FROM Customers;

DELETE FROM Customer_types
WHERE Customer_type_id=321;
SELECT *FROM Customer_types;

DELETE FROM Products_services
WHERE Services_id=221;
SELECT *FROM Products_services;

UPDATE Products_services SET
Employee_id=NULL;

DELETE FROM Employees
WHERE Employee_id=2222;
SELECT *FROM Employees;

DELETE FROM Departments
WHERE Department_id=112;
SELECT *FROM Departments;

UPDATE Departments SET Branch_id=NULL;

DELETE FROM Branches
WHERE Branch_id=15;
SELECT *FROM Branches;

UPDATE Branches SET Bank_id=NULL;

DELETE FROM Bank
WHERE Bank_id=01;
SELECT *FROM Bank;
```

Appendix C

Query Statement:

```
SELECT Employees.Employee_id,Departments.Department_name
FROM Employees
INNER JOIN Departments ON Employees.Department_id = Departments.Department_id;

SELECT Customers.Customer_first_name,Customers.Customer_last_name,Customer_types.Customer_type_id
FROM Customers
LEFT JOIN Customer_types
ON Customer_types.Customer_type_id = Customers.Customer_type_id
ORDER BY Customers.Customer_first_name,Customers.Customer_last_name;

SELECT Transactions.Transaction_id,Transactions.Transaction_date,
Transactions.Transaction_amount,Accounts.Account_id
FROM Transactions
RIGHT JOIN Accounts
ON Transactions.Account_id = Accounts.Account_id
ORDER BY Transactions.Transaction_id,Transactions.Transaction_amount,Transactions.Transaction_date;

SELECT Branches.Branch_name,Banches.Branch_city,Banches.Branch_country,Bank.Bank_id
FROM Branches
FULL OUTER JOIN Bank ON Branches.Bank_id = Bank.Bank_id
ORDER BY Branches.Branch_name,Banches.Branch_city,Banches.Branch_country;
SELECT MIN(Salary) AS SmallestSalary
FROM Employees;

SELECT MAX(Transaction_amount ) AS LargestAmount
FROM Transactions;

SELECT COUNT (Salary) AS Num_Salary
FROM Employees;

SELECT AVG (Transaction_amount) AS AverageOfTransactionAmount
FROM Transactions;

SELECT SUM (Salary) AS Total_of_salary
FROM Employees
WHERE Salary > 100000;

SELECT*FROM Customers
WHERE Gender='Female' AND Customer_id > 555;

SELECT*FROM Bank
WHERE Bank_name='Kaspi Bank' OR Bank_name = 'Sberbank';

SELECT*FROM Departments
WHERE NOT Department_name = 'Risk Management Department';
```

Subquery Statement:

Single-row subqueries:

```
SELECT Branch_name, Branch_id, Branch_city, Branch_country
FROM Branches
WHERE Branch_id=(SELECT Branch_id FROM Branches WHERE Branch_name='Kaspi Bank Branch');

SELECT Transaction_id, Transaction_date, Transaction_amount, Account_id, Acquest_id, Transaction_type_id
FROM Transactions
WHERE Transaction_amount>(SELECT Transaction_amount
FROM Transactions WHERE Transaction_date='14-APR-2016');

SELECT Transaction_id, Transaction_date, Transaction_amount, Account_id, Acquest_id, Transaction_type_id
FROM Transactions
WHERE Transaction_amount<(SELECT Transaction_amount
FROM Transactions WHERE Transaction_date='14-APR-2016');

SELECT Employee_id, Employee_first_name, Employee_last_name, Salary
FROM Employees
WHERE Employee_id <>(SELECT Employee_id
FROM Employees WHERE Salary='290000');
```

Multiple-row subqueries:

```
SELECT Bank_name, Bank_id, Bank_city, Bank_country
FROM Bank
WHERE Bank_id IN(SELECT Bank_id FROM Bank WHERE Bank_country='Russia');

SELECT Bank_name, Bank_id, Bank_city, Bank_country
FROM Bank
WHERE Bank_id NOT IN(SELECT Bank_id FROM Bank WHERE Bank_country='Russia');

SELECT Customer_id, Customer_first_name, Customer_last_name, Customer_type_id
FROM Customers
WHERE Customer_id =ANY(SELECT Customer_id FROM Customers WHERE Gender='Female');

SELECT Department_id, Department_name, Description, Branch_id
FROM Departments
WHERE Department_id IN(SELECT DISTINCT Department_id
FROM Departments WHERE Branch_id=12);
```

Multiple-column subquery:

```
SELECT Employee_id, Employee_first_name, Employee_last_name, Department_id
FROM Employees
WHERE Department_id IN(SELECT Department_id
FROM Employees WHERE Employee_id IN(1111,111));
```

Conclusion

Bank management system is a virtualization of transactions in banking system. The banking system are used manual working but when we used online banking system it is totally virtualization process which avoid manual process and converts it in automatic process. This project is developed to nurture the needs of a user in a banking sector by embedding all the tasks of transactions taking place in a bank. Future version of this project will still be much enhanced than the current version. Access to the balance in your checking account can also be limited by businesses that place holds on your funds. Banks are providing internet banking services also so that the customers can be attracted. By asking the bank employs we came to know that maximum numbers of internet bank account holders are youth and businessman. Bank Management System is an innovative tool that is fast becoming a necessity. It is a successful strategic weapon for banks to remain profitable in a volatile and competitive marketplace of today. If proper training should be given to customer by the bank employs to open an account will be beneficial secondly the website should be made friendlier from where the first time customers can directly make and access their accounts. Thus the Bank Management System it is developed and executed successfully.

The traditional way of storing user data in a bank is to enter data and record it. Every time a user needs to perform some kind of transaction, he must go to the bank and perform the necessary actions, which can always be done. This can be a simple task for both users and bankers. The project provides a real understanding of the banking system and the activities performed by the various roles in the supply chain. Here we provide banking system automation. Banking system project captures the actions performed by different roles in real-world banking, which provides improved methods for keeping the information you need up to date, leading to increased efficiency. The project provides a real understanding of the banking system and the activities performed by the various roles in the supply chain.

Reflection

Moldir Kumarbek

My main role in the project is to take advantage (just kidding). In the production of the project, I showed a function that I didn't know before, which is to find errors quickly and quickly. When writing code, there will be many errors. I can observe carefully and accurately Find its errors and improve.

I mainly write database, ALTER TABLE also has queries. In the process of writing, we encountered a lot of problems. The wrong format or lack of the code caused a lot of troubles, but we have overcome them. Taking a role in the team to help the team is the ultimate task. Finally, I wrote a presentation to our small team. Maryam is the multifunctional one in our small team, and she has given a lot.

Maryam Bayzhigitova

My role in this project consisted of creating a diagram, and then parsing the tasks associated with subqueries and DML statements. Then, after solving all the problems, Moldir and I checked everything for correctness, and then analyzed our problems together. For us, the most difficult thing was to define our theme, since we wanted to choose something global. I believe that we made our project without any drawbacks, since we did it together.

I think we did a good job because we did it together. Sometimes, we have a call with Moldir on the video chat and discussed our tasks. First, we divided our project equally, after having united, we analyzed all the tasks. For us it was not only a project, but also a lesson in group work.