

Online Bank Database

IS443 Group 3

Mohamed Abdi, Folly Adamazan Teko, Nicole Jacobo,

Claude Lee, Paul Lombard, Dego Mohamed

Fall 2020

Table of Contents

Introduction.....	3
Database Structure.....	4
Business Rules.....	4
Definition of Entities.....	5
Entity Relationships.....	6
Database Design.....	8
Bank.....	8
Department.....	9
Employee.....	10
Customer.....	11
Assignment.....	12
Bank Account.....	13
Card.....	14
Attributes Overview & ERD.....	15
Data Entries.....	16
Queries in Database.....	21
Query 1: Checking Account Promotion.....	21
Query 2: Expired Card.....	22
Query 3: Non-Active Accounts.....	23
Summary & Discussions.....	25

Introduction

The goal of our project was to design a basic database for a bank using Oracle Live SQL. While the entities and relationships are rudimentary compared to a complete bank database, we wanted the function of the database to be as close to an actual bank's database as possible. In the initial discussion, we identified the key components of a bank and how they interact. Of course, banks require people to keep them operational, and there needs to be a hierarchy in managing those people. We created the department entity with the intent to create a more efficient way of managing employees, our next entity.

For the customer side of the database, we discussed the flexibility that we wanted a customer to have, and the typical things that a bank allows. In this discussion, we agreed that people often want the ability to have multiple accounts, they want to have a card (credit, debit, or ATM) linked to their account, and they may want to have a joint user for the account. Initially, this prompted just three other entities – Customer, Account, and Card with a M:M relationship between Customer and Account which was later adapted to allow a bridge (Assignment). Using these entities, we developed the ERD along with the entity attributes, to inform and structure the SQL code that we would write for our bank database.

Database Structure

Database structure is based on the business rules. Depending on what rules we have for our online bank, entities their relationships are defined. This information is visually summarized in “Entity Relationship Diagram (ERD).”

- Business Rules

1. A bank has many departments. Each department belongs to one bank.
2. A department has many employees. Each employee belongs to one department.
3. A bank has many customers. Each customer belongs to one bank.
4. A customer can have many account assignments. Each account assignment can belong to one customer.
5. An account can have many account assignments. Each account assignment can belong to one account.
6. An account assignment can have one card. Each card belongs to one account assignment.

Note that rule 4 and 5 are derived from a rule “a customer can have many accounts. An account can belong to many customers.” To avoid any many-to-many relationship, we split the rule into two separate rules by adding account assignment in between.

- Definition of Entities

- Bank

Bank is an entity to store data about the bank itself. The entity includes name, address, and phone number of the bank. Since we only have one bank, there is only one entry for this entity.

- Department

Department is an entity to store data about the different departments in the bank. It includes attributes like department id, name, and bank id which is basically referring to the bank id (primary key in the bank table)

- Employee

Employee is an entity to store records of each individual employee in the bank. It includes attributes the employee id, first name, last name, initial, hire date, job title, and the department id which is a foreign key referring to the department entities primary key.

- Customer

The customer table is built to store data about our customers. The entities are customer id, first name, last name, initial, area code, phone number, and a foreign key called bank id referring to the primary key in the bank table.

- Bank Account

The bank account table summarizes accounts information. It stores data like account number, type, balance, opening day, and status (whether the account is active or not).

- Assignment

Assignment is a bridge entity between Bank Account and Customer. This entity shows which account is assigned to which customer as well as card information if applicable. This allows the database to account for multiple accounts being assigned to one customer as well as

multiple customers being assigned to one account. Additionally, assignment can allow database to assign the card information to the account and the customer (card holder).

- Card

The card table stores card information. The card table attributes are card number, type, issue date, expiring date, pin, and assignment code which is the foreign key pointing to the bridge (assignment table) primary key.

- Entity Relationships

- A bank has many departments. Each department belongs to one bank.

➔ Bank and Department have one-to-many relationship.

Bank entity's primary key can be used to link the two entities.

- A department has many employees. Each employee belongs to one department.

➔ Department and Employee have one-to-many relationship.

Department entity's primary key can be used to link the two entities.

- A bank has many customers. Each customer belongs to one bank.

➔ Bank and Customer have one-to-many relationship.

Bank entity's primary key can be used to link the two entities.

- A customer can have many account assignments. Each account assignment can belong to one customer.

➔ Customer and Assignment have one-to-many relationship.

Assignment entity's primary key can be used to link the two entities.

- An account can have many account assignments. Each account assignment can belong to one account.

➔ Bank Account and Assignment have one-to-many relationship.

Assignment entity's primary key can be used to link the two entities.

- An account assignment can have one card. Each card belongs to one account assignment.

➔ Card and Assignment have one-to-one relationship.

Assignment entity's primary key can be used to link the two entities.

Note that having Assignment entity allowed the database to avoid having many-to-many relationship between Customer and Bank Account and ambiguous relationship of Card in between them.

Database Design

In database design, we defined entity attributes including primary keys and foreign keys, their entries, and the SQL code used to create the database. Note that PK represents primary key and FK represents foreign key.

- Bank

Attribute Name	Format	Description
BANK_ID (PK)	Number	Bank ID Number
BANK_NAME	Up to 15 Chars	Bank Name
BANK_ADDR	Up to 35 Chars	Bank Street Address
BANK_ZIP	5 Numbers	Bank Zip Code
BANK_AREACODE	3 Chars	Bank Area Code
BANK_PHONE	8 Chars	Bank Phone Number

- SQL Code:

```
CREATE TABLE BANK (
  BANK_ID          NUMBER          PRIMARY KEY NOT NULL,
  BANK_NAME        VARCHAR(15) NOT NULL,
  BANK_ADDR        VARCHAR(35) NOT NULL,
  BANK_ZIP         NUMBER(5)      NOT NULL,
  BANK_AREACODE    CHAR(3)        NOT NULL,
  BANK_PHONE       CHAR(8)        NOT NULL);
```


- Department

Attribute Name	Format	Description
DEPART_ID (PK)	Number	Department ID Number
BANK_ID (FK)	Number	Bank ID Number
DEPART_NAME	Up to 15 Chars	Name of Department

Foreign key BANK_ID is used to show one-to-many relationship between Bank and Department.

- SQL Code:

```
CREATE TABLE DEPARTMENT (
DEPART_ID          NUMBER          PRIMARY KEY NOT NULL,
DEPART_NAME        VARCHAR(15)     NOT NULL,
BANK_ID            NUMBER          NOT NULL,
FOREIGN KEY        (BANK_ID)       REFERENCES BANK);
```

- Employee

Attribute Name	Format	Description
EMP_ID (PK)	Number	Employee ID Number
DEPART_ID (FK)	Number	Department ID Number
EMP_FNAME	Up to 15 Chars	Employee's First Name
EMP_LNAME	Up to 15 Chars	Employee's Last Name
EMP_INITIAL	1 Char	Employee's Middle Initial
EMP_HIREDATE	Date	Employee's Hiring Date
JOB_TITLE	Up to 25 Chars	Employee's Job Title

Foreign key DEPART_ID is used to show one-to-many relationship between Department and Employee.

- SQL Code:

```
CREATE TABLE EMPLOYEE (
EMP_ID          NUMBER          PRIMARY KEY NOT NULL,
EMP_FNAME       VARCHAR(15) NOT NULL,
EMP_LNAME       VARCHAR(15) NOT NULL,
EMP_INITIAL     CHAR(1) ,
EMP_HIREDATE    DATE,
JOB_TITLE       VARCHAR(25) ,
DEPART_ID       NUMBER,
FOREIGN KEY     (DEPART_ID) REFERENCES DEPARTMENT) ;
```

- Customer

Attribute Name	Format	Description
CUS_ID (PK)	Number	Customer ID Number
BANK_ID (FK)	Number	Bank ID Number
CUS_LNAME	Up to 15 Chars	Customer's Last Name
CUS_FNAME	Up to 15 Chars	Customer's First Name
CUS_INITIAL	1 Char	Customer's Middle Initial
CUS_AREACODE	3 Chars	Customer's Area Code
CUS_PHONE	8 Chars	Customer's Phone Number

Foreign key BANK_ID is used to show one-to-many relationship between Bank and Customer.

- SQL Code:

```
CREATE TABLE CUSTOMER (
CUS_ID          NUMBER          PRIMARY KEY NOT NULL,
CUS_LNAME       VARCHAR(15) NOT NULL,
CUS_FNAME       VARCHAR(15) NOT NULL,
CUS_INITIAL     CHAR(1) ,
CUS_AREACODE    CHAR(3) ,
CUS_PHONE       CHAR(8) ,
BANK_ID         NUMBER,
FOREIGN KEY     (BANK_ID) REFERENCES BANK);
```

- Bank Account

Attribute Name	Format	Description
ACCT_NUM (PK)	Number	Customer ID Number
ACCT_TYPE	Up to 10 Chars	Type of Account (ex. Checking, Savings, IRA)
ACCT_BAL	Up to 99999999.99	Account Balance
OPEN_DATE	Date	Date of Account Opening
ACTIV_STATUS	Up to 10 Chars	Account Activity Status (ex. Active, Inactive, Closed)

- SQL Code:

```
CREATE TABLE BANK_ACCOUNT (
ACCT_NUM      NUMBER      PRIMARY KEY NOT NULL,
ACCT_TYPE     VARCHAR(10)  NOT NULL,
ACCT_BAL      NUMBER(10,2) NOT NULL,
OPEN_DATE     DATE,
ACTIV_STATUS  VARCHAR(10)  NOT NULL);
```

- Assignment

Attribute Name	Format	Description
ASSIGN_CODE (PK)	Number	Account Assignment Code
CUS_ID (FK)	Number	Customer ID Number
ACCT_NUM (FK)	Number	Account ID Number

Since Assignment is a bridge entity between Bank Account and Customer, foreign keys ACCT_NUM from Bank Account and CUS_ID from Customer are used to show one-to-many relationships between Bank Account and Assignment and Customer and Assignment.

- SQL Code:

```
CREATE TABLE ASSIGNMENT (
  ASSIGN_CODE    NUMBER      PRIMARY KEY NOT NULL,
  CUS_ID         NUMBER,
  ACCT_NUM       NUMBER,
  FOREIGN KEY    (CUS_ID)    REFERENCES CUSTOMER,
  FOREIGN KEY    (ACCT_NUM)  REFERENCES BANK_ACCOUNT);
```

- Card

Attribute Name	Format	Description
CARD_NUM (PK)	16 Digit Number	Card Number
ASSIGN_CODE (FK)	Number	Links Card to Assignment
CARD_TYPE	Up to 10 Chars	Type of Card (ex. Debit Card, Credit Card, ATM Card)
ISSUE_DATE	Date	Date of Issue
EXPIR_DATE	Date	Date of Expiration
CARD_PIN	4 Digit Number	Card PIN

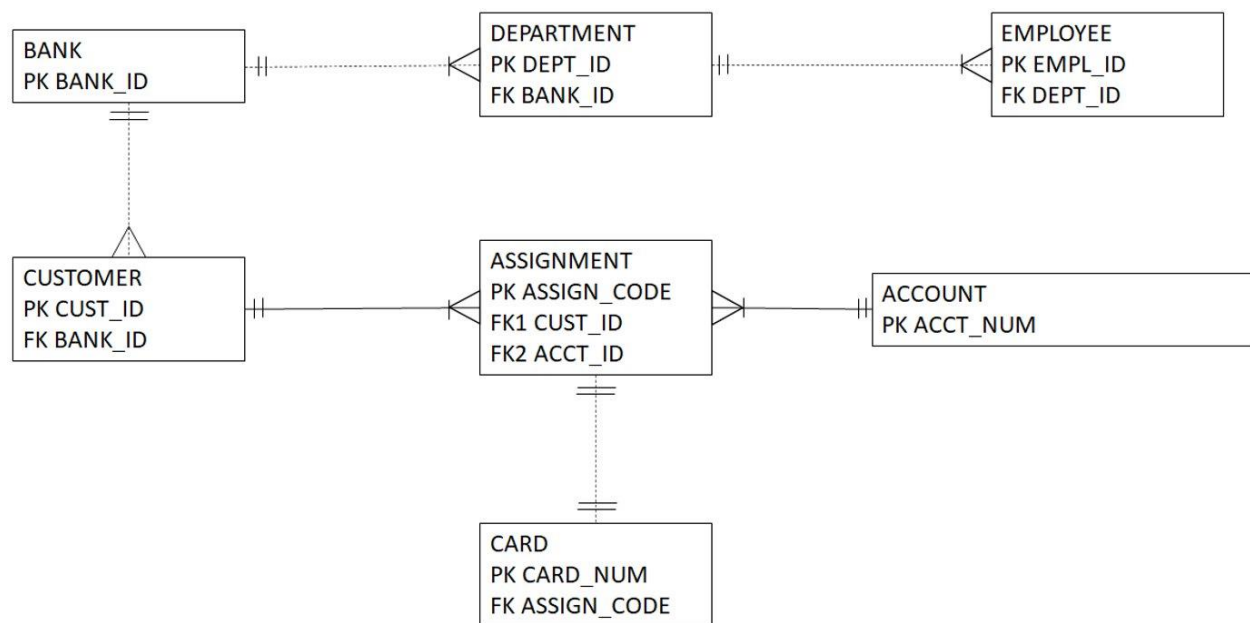
Foreign key ASSIGN_CODE is used to show one-to-many relationship between Assignment and Card.

- SQL Code:

```
CREATE TABLE CARD (
  CARD_NUM      NUMBER(16)      PRIMARY KEY NOT NULL,
  CARD_TYPE     VARCHAR(10)     NOT NULL,
  ISSUE_DATE    DATE,
  EXPIR_DATE    DATE,
  CARD_PIN      NUMBER(4)       NOT NULL,
  ASSIGN_CODE   NUMBER,
  FOREIGN KEY   (ASSIGN_CODE) REFERENCES ASSIGNMENT);
```

- Attributes Overview & ERD

Entity	Primary Key	Foreign Key	Other Attributes
BANK	BANK_ID	NONE	BANK_NAME, BANK_ADDR, BANK_ZIP, BANK_AREACODE, BANK_PHONE
DEPARTMENT	DEPART_ID	BANK_ID	DEPART_NAME
EMPLOYEE	EMP_ID	DEPART_ID	EMP_FNAME, EMP_LNAME, EMP_INITIAL, EMP_HIREDATE, JOB_TITLE
CUSTOMER	CUS_ID	BANK_ID	CUS_LNAME, CUS_FNAME, CUS_INITIAL, CUS_AREACODE, CUS_PHONE
BANK_ACCOUNT	ACCT_NUM	NONE	ACCT_TYPE, ACCT_BAL, OPEN_DATE, ACTIV_STATUS
ASSIGNMENT	ASSIGN_CODE	CUS_ID, ACCT_NUM	
CARD	CARD_NUM	ASSIGN_CODE	CARD_TYPE, ISSUE_DATE, EXPIR_DATE, CARD_PIN



- Data Entries

- Bank

BANK_ID	BANK_NAME	BANK_ADDR	BANK_ZIP	BANK_AREACODE	BANK_PHONE
123	GROUP 3 BANK	720 4TH AVE S	56301	678	999-8212

[Download CSV](#)

- Department

DEPART_ID	DEPART_NAME	BANK_ID
1	Retail	123
2	Lending	123
3	Loan Operations	123
4	Private Bank	123
5	Corporate Bank	123
6	Treasury	123
7	Credit Risk	123
8	Finance	123
9	HR	123
10	Administration	123

[Download CSV](#)

10 rows selected.

- Employee

EMP_ID	EMP_FNAME	EMP_LNAME	EMP_INITIAL	EMP_HIREDATE	JOB_TITLE	DEPART_ID
12345	Adam	Thielen	J	20-JAN-16	Teller	1
23456	Kirk	Cousins	A	20-FEB-18	Loan Officer	2
34567	Joe	Mauer	J	20-JAN-01	Loan Supervisor	3
45678	Justin	Jefferson	D	20-APR-20	Financial Advisor	4
56789	Tom	Brady	P	20-JAN-00	Business Lending	5
67891	Michael	Jordan	J	20-JAN-84	Accounting	6

[Download CSV](#)

6 rows selected.

- Customer

CUS_ID	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	BANK_ID
1	Lombard	Paul	L	320	767-9575	123
2	Lee	Claude	-	320	435-5978	123
3	Mohamed	Dego	-	507	395-4893	123
4	Mohamed	abdi	M	789	456-9803	123
5	Adamazan teko	folly	-	507	375-9845	123
6	Jacobo	Nicole	N	320	487-8946	123
7	John	Doe	-	750	574-7909	123
8	Jane	Doe	-	869	679-8943	123
9	Ashey	Cole	C	507	689-7890	123
10	Abigail	Ayoko	-	700	345-8983	123

[Download CSV](#)

10 rows selected.

- Bank Account

ACCT_NUM	ACCT_TYPE	ACCT_BAL	OPEN_DATE	ACTIV_STATUS
922404	TRADIRA	86336.41	24-JAN-96	ACTIVE
599364	SAVINGS	165.54	14-OCT-18	ACTIVE
260080	SAVINGS	6393.48	04-JAN-06	ACTIVE
169609	CHECKING	1303.82	30-MAR-95	ACTIVE
742833	CHECKING	-359.65	18-FEB-14	OVERDRAWN
333390	SAVINGS	49.32	11-MAR-02	INACTIVE
755869	HSA	741.05	29-NOV-14	ACTIVE
551836	SAVINGS	121620.68	21-AUG-06	ACTIVE
795894	CHECKING	-12.92	19-DEC-18	OVERDRAWN
181260	CHECKING	439.81	25-FEB-19	ACTIVE

[Download CSV](#)

10 rows selected.

- Assignment

ASSIGN_CODE	CUS_ID	ACCT_NUM
1	1	922404
2	1	599364
3	2	260080
4	3	169609
5	4	169609
6	5	169609
7	6	742833
8	7	742833
9	8	333390
10	8	181260
11	9	181260
12	9	755869
13	10	551836
14	10	795894

Download CSV

14 rows selected.

- Card

CARD_NUM	CARD_TYPE	ISSUE_DATE	EXPIR_DATE	CARD_PIN	ASSIGN_CODE
4243507034864858	Debit	22-OCT-18	23-OCT-22	1234	1
4243507034863858	Debit	22-OCT-10	23-OCT-14	1534	2
4243507034862858	Credit	22-OCT-14	23-OCT-18	1274	3
4243507034865858	ATM	22-OCT-12	22-OCT-16	2234	4
4243407034864858	Credit	22-OCT-16	23-OCT-20	6244	5
4243507134865858	ATM	22-OCT-06	23-OCT-10	4334	6
4142507034864858	Debit	22-JAN-18	23-JAN-22	8834	7
4142507134865858	Credit	02-FEB-14	03-FEB-18	6666	8
4243570034864858	Debit	06-JUN-18	07-JUN-22	1294	9
4243507034864059	Credit	22-OCT-20	23-OCT-24	1010	10

Download CSV

10 rows selected.

- SQL Code:

```

/* Bank */
INSERT INTO BANK VALUES (123, 'GROUP 3 BANK','720 4TH AVE S',
56301, '678','999-8212');
/* Department */
INSERT INTO DEPARTMENT VALUES (001,'Retail',123);
INSERT INTO DEPARTMENT VALUES (002,'Lending',123);
INSERT INTO DEPARTMENT VALUES (003,'Loan Operations',123);
INSERT INTO DEPARTMENT VALUES (004,'Private Bank',123);
INSERT INTO DEPARTMENT VALUES (005,'Corporate Bank',123);
INSERT INTO DEPARTMENT VALUES (006,'Treasury',123);
INSERT INTO DEPARTMENT VALUES (007,'Credit Risk',123);
INSERT INTO DEPARTMENT VALUES (008,'Finance',123);
INSERT INTO DEPARTMENT VALUES (009,'HR',123);
INSERT INTO DEPARTMENT VALUES (010,'Adminisration',123);
/* Employee */
INSERT INTO EMPLOYEE VALUES (12345,'Adam','Thielen','J','20-JAN-
2016','Teller',001);
INSERT INTO EMPLOYEE VALUES (23456,'Kirk','Cousins','A','20-FEB-2018','Loan
Officer',002);
INSERT INTO EMPLOYEE VALUES (34567,'Joe','Mauer','J','20-JAN-2001','Loan
Supervisor',003);
INSERT INTO EMPLOYEE VALUES (45678,'Justin','Jefferson','D','20-APR-
2020','Financial Advisor',004);
INSERT INTO EMPLOYEE VALUES (56789,'Tom','Brady','P','20-JAN-2000','Business
Lending',005);
INSERT INTO EMPLOYEE VALUES (67891,'Michael','Jordan','J','20-JAN-
1984','Accounting',006);
/* Customer */
INSERT INTO CUSTOMER VALUES(1, 'Lombard', 'Paul', 'L', '320', '767-9575',
123);
INSERT INTO CUSTOMER VALUES(2, 'Lee', 'Claude', '', '320', '435-5978', 123);
INSERT INTO CUSTOMER VALUES(3, 'Mohamed', 'Dego', '', '507', '395-4893',
123);
INSERT INTO CUSTOMER VALUES(4, 'Mohamed', 'abdi', 'M', '789', '456-9803',
123);
INSERT INTO CUSTOMER VALUES(5, 'Adamazan teko', 'folly', '', '507', '375-
9845', 123);
INSERT INTO CUSTOMER VALUES(6, 'Jacobo', 'Nicole', 'N', '320', '487-8946',
123);
INSERT INTO CUSTOMER VALUES(7, 'John', 'Doe', '', '750', '574-7909', 123);
INSERT INTO CUSTOMER VALUES(8, 'Jane', 'Doe', '', '869', '679-8943', 123);
INSERT INTO CUSTOMER VALUES(9, 'Ashey', 'Cole', 'C', '507', '689-7890',123);
INSERT INTO CUSTOMER VALUES(10, 'Abigail', 'Ayoko', '', '700', '345-8983',
123);
/* Bank Account */
INSERT INTO BANK_ACCOUNT VALUES(922404,'TRADIRA',86336.41,'24-JAN-
1996','ACTIVE');
INSERT INTO BANK_ACCOUNT VALUES(599364,'SAVINGS',165.54,'14-OCT-
2018','ACTIVE');
INSERT INTO BANK_ACCOUNT VALUES(260080,'SAVINGS',6393.48,'4-JAN-
2006','ACTIVE');
INSERT INTO BANK_ACCOUNT VALUES(169609,'CHECKING',1303.82,'30-MAR-
1995','ACTIVE');

```

```

INSERT INTO BANK_ACCOUNT VALUES (742833, 'CHECKING', -359.65, '18-FEB-
2014', 'OVERDRAWN');
INSERT INTO BANK_ACCOUNT VALUES (333390, 'SAVINGS', 49.32, '11-MAR-
2002', 'INACTIVE');
INSERT INTO BANK_ACCOUNT VALUES (755869, 'HSA', 741.05, '29-NOV-2014', 'ACTIVE');
INSERT INTO BANK_ACCOUNT VALUES (551836, 'SAVINGS', 121620.68, '21-AUG-
2006', 'ACTIVE');
INSERT INTO BANK_ACCOUNT VALUES (795894, 'CHECKING', -12.92, '19-DEC-
2018', 'OVERDRAWN');
INSERT INTO BANK_ACCOUNT VALUES (181260, 'CHECKING', 439.81, '25-FEB-
2019', 'ACTIVE');
/* Assignment */
INSERT INTO ASSIGNMENT VALUES (1, 1, 922404);
INSERT INTO ASSIGNMENT VALUES (2, 1, 599364);
INSERT INTO ASSIGNMENT VALUES (3, 2, 260080);
INSERT INTO ASSIGNMENT VALUES (4, 3, 169609);
INSERT INTO ASSIGNMENT VALUES (5, 4, 169609);
INSERT INTO ASSIGNMENT VALUES (6, 5, 169609);
INSERT INTO ASSIGNMENT VALUES (7, 6, 742833);
INSERT INTO ASSIGNMENT VALUES (8, 7, 742833);
INSERT INTO ASSIGNMENT VALUES (9, 8, 333390);
INSERT INTO ASSIGNMENT VALUES (10, 8, 181260);
INSERT INTO ASSIGNMENT VALUES (11, 9, 181260);
INSERT INTO ASSIGNMENT VALUES (12, 9, 755869);
INSERT INTO ASSIGNMENT VALUES (13, 10, 551836);
INSERT INTO ASSIGNMENT VALUES (14, 10, 795894);
/* Card */
INSERT INTO CARD VALUES (4243507034864858, 'Debit', '22-OCT-2018', '23-OCT-
2022', 1234, 1);
INSERT INTO CARD VALUES (4243507034863858, 'Debit', '22-OCT-2010', '23-OCT-
2014', 1534, 2);
INSERT INTO CARD VALUES (4243507034862858, 'Credit', '22-OCT-2014', '23-OCT-
2018', 1274, 3);
INSERT INTO CARD VALUES (4243507034865858, 'ATM', '22-OCT-2012', '22-OCT-
2016', 2234, 4);
INSERT INTO CARD VALUES (4243407034864858, 'Credit', '22-OCT-2016', '23-OCT-
2020', 6244, 5);
INSERT INTO CARD VALUES (4243507134865858, 'ATM', '22-OCT-2006', '23-OCT-
2010', 4334, 6);
INSERT INTO CARD VALUES (4142507034864858, 'Debit', '22-JAN-2018', '23-JAN-
2022', 8834, 7);
INSERT INTO CARD VALUES (4142507134865858, 'Credit', '2-FEB-2014', '3-FEB-
2018', 6666, 8);
INSERT INTO CARD VALUES (4243570034864858, 'Debit', '06-JUN-2018', '07-JUN-
2022', 1294, 9);
INSERT INTO CARD VALUES (4243507034864059, 'Credit', '22-OCT-2020', '23-OCT-
2024', 1010, 10);

```

Queries in Database

Now that we have developed database for the online bank, let's solve some of the queries that can arise in real-world online banking environment.

- Query 1: Checking Account Promotion

Non-checking accounts tend to influence bank more significant than checking accounts.

We would like to see types and customer of those accounts.

- Process

We can obtain list of customers and their accounts by filtering out ones whose account type is not checking.

- Data Needed

Customer's name (CUSTOMER.CUS_LNAME, CUSTOMER.CUS_FNAME)

Account type & number (BANK_ACCOUNT.ACCT_TYPE, BANK_ACCOUNT.ACCT_NUM)

- Entities & Needed for Query

ACCOUNT, CUSTOMER, ASSIGNMENT

Account type (BANK_ACCOUNT.ACCT_TYPE)

➔ Used to filter out ones that are not checking

Customer ID and Account number (CUS_ID, ACCT_NUM)

➔ Used to match reference

- SQL Code:

```
SELECT CUS_LNAME, CUS_FNAME, ACCT_TYPE, BANK_ACCOUNT.ACCT_NUM
FROM CUSTOMER, ASSIGNMENT, BANK_ACCOUNT
WHERE CUSTOMER.CUS_ID = ASSIGNMENT.CUS_ID
AND BANK_ACCOUNT.ACCT_NUM = ASSIGNMENT.ACCT_NUM
AND BANK_ACCOUNT.ACCT_TYPE != 'CHECKING';
```

- Output

CUS_LNAME	CUS_FNAME	ACCT_TYPE	ACCT_NUM
Lombard	Paul	TRADIRA	922404
Lombard	Paul	SAVINGS	599364
Lee	Claude	SAVINGS	260080
Jane	Doe	SAVINGS	333390
Ashey	Cole	HSA	755869
Abigail	Ayoko	SAVINGS	551836

- Query 2: Expired Card

We would like to inform customers whose cards have been expired. To be able to obtain the list of those customers, we need to find all cards that have been expired and pull the assigned customer information.

- Process

We can obtain list of customers and their card numbers by filtering out ones whose card expiration date is before today (as of 12/9/20).

- Data Needed

Customer's information (CUSTOMER.CUS_LNAME, CUSTOMER.CUS_FNAME,
CUSTOMER.CUS_INITIAL, CUSTOMER.CUS_AREACODE, CUSTOMER.CUS_PHONE)

Card number (CARD.CARD_NUM)

- Entities & Needed for Query

CARD, CUSTOMER, ASSIGNMENT

Card's expiration date (CARD.EXPIRE_DATE)

➔ Used to filter out ones whose expiration date is passed.

Customer ID and Assignment Code (CUS_ID, ASSIGN.CODE)

➔ Used to match reference

- SQL Code:

```
SELECT CUS_LNAME, CUS_FNAME, CUS_INITIAL, CUS_AREACODE,
CUS_PHONE, CARD_NUM
FROM CUSTOMER, ASSIGNMENT, CARD
WHERE CUSTOMER.CUS_ID = ASSIGNMENT.CUS_ID
AND ASSIGNMENT.ASSIGN_CODE = CARD.ASSIGN_CODE
AND CARD.EXPIR_DATE >= '09-DEC-2020';
```

- Output

CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CARD_NUM
Lombard	Paul	L	320	767-9575	4243507034864858
Jacobo	Nicole	N	320	487-8946	4142507034864858
Jane	Doe	-	869	679-8943	4243507034864858
Jane	Doe	-	869	679-8943	4243507034864059

[Download CSV](#)

4 rows selected.

- Query 3: Non-Active Accounts

We would like to see accounts that are not active, and the customers associated with those accounts.

- Process

We can obtain list of customers and their accounts by filtering out ones whose account's active status is not active.

- Data Needed

Customer's name (CUSTOMER.CUS_LNAME, CUSTOMER.CUS_FNAME, CUSTOMER.CUS_INITIAL)

Account number & active status (BANK_ACCOUNT.ACCT_NUM, BANK_ACCOUNT.ACTIVE_STATUS)

- Entities & Needed for Query

ACCOUNT, CUSTOMER, ASSIGNMENT

Account active status (BANK_ACCOUNT.ACTIVE_STATUS)

➔ Used to filter out ones that are not active

Customer ID and Account number (CUS_ID, ACCT_NUM)

➔ Used to match reference

- SQL Code:

```
SELECT CUS_LNAME, CUS_FNAME, CUS_PHONE, BANK_ACCOUNT.ACCT_NUM,
ACTIV_STATUS
FROM CUSTOMER, BANK_ACCOUNT, ASSIGNMENT
WHERE CUSTOMER.CUS_ID = ASSIGNMENT.CUS_ID
AND BANK_ACCOUNT.ACCT_NUM = ASSIGNMENT.ACCT_NUM
AND BANK_ACCOUNT.ACTIV_STATUS != 'ACTIVE';
```

- Output

CUS_LNAME	CUS_FNAME	CUS_INITIAL	ACCT_NUM	ACTIV_STATUS
Jacobo	Nicole	N	742833	OVERDRAWN
John	Doe	-	742833	OVERDRAWN
Jane	Doe	-	333390	INACTIVE
Abigail	Ayoko	-	795894	OVERDRAWN

[Download CSV](#)

4 rows selected.

Summary & Discussions

There were challenges throughout the project regarding the ERD relationships and the SQL application. The main obstacle was to understand how our bridge would connect the tables to ensure that database maintenance is simple, and to ensure that customers have greater control and accessibility of their financial needs. Our second challenge was to produce results from a query and to make sure that the results were accurate. We navigated this by approaching our queries from a business mindset, identifying information that would be most useful to a business owner, and the information needed to complete those queries.

Lastly, our online database is a simplified version of an actual online banking system with limited assumptions. Some ideas to expand on this model to make it more applicable include:

- Accounting for transactions
 - o The complexity of tracking these transactions expand beyond the scope of our project and would require a more robust database
- Organizational structure
 - o The database does not explore more complex relationships within the business side of the bank. For example, the database could be built to allow physical branches within the hierarchy, assign department leaders, or reference job code tables. Banking entails numerous business operations that are not identified within this database.

Given the ability to expand the scope of this project, these ideas could be easily implemented because we accounted for both business needs, and customer needs at every step in the development process. By recognizing the need for simplicity in database management and

the importance of customer satisfaction to a business, we hope that this elementary database structure will allow for expansion of not just the database, but of a potential business.