Table of Contents

[Introduction 1](#_Toc121168892)

[Business analysis: 1](#_Toc121168893)

[Overview: 1](#_Toc121168894)

[Scenario: 1](#_Toc121168895)

[Business Rules: 1](#_Toc121168896)

[Legal and Ethical issues: 2](#_Toc121168897)

[ER (Logical) diagram: 3](#_Toc121168898)

[Relational Table headings 4](#_Toc121168899)

[Set of Tables: 5](#_Toc121168900)

[Display inserted Data: 13](#_Toc121168901)

[SQL Queries 21](#_Toc121168902)

[Testing: 26](#_Toc121168903)

[Data Dictionary: 30](#_Toc121168904)

[Conclusion: 36](#_Toc121168905)

[References: 37](#_Toc121168906)

[Appendix 1 38](#_Toc121168907)

[Appendix 2 52](#_Toc121168908)

[Normalisation: 52](#_Toc121168909)

# Introduction

This report aimed to build a database system that fulfils C of B Bank’s needs which is storing and utilising data digitally. This report also focusses on the business rules, potential issues and risk related to the organisation’s data, as well as the database schema and the technical requirements (including SQL statements) to build the system.

# Business analysis:

## Overview:

Data in a financial institution can be of several forms and relates to different aspects of business operations, such as demographic and personal customer data, employee details, and data regarding the services available, their function, and potential issues as well as feedback from both customers and employees making use of said services. The subcategories of data are all important to different aspects of the business model and ensure the smooth functioning of the company and adequate delivery of services.

## Scenario:

C of B –Cindy’s Bank of Students – A Bank for personal customers who are students, provides several banking services, such as current accounts, loans, paying bills, money transfers, payments, and cards. C of B Bank was established in 2010. After one year, in 2011, the Bank provided account services as well as cards to its customers who are students in the UK. The Bank introduced a variety of loans such as personal and joint loans in the following year. Each branch has several employees who are serving customers as their relationship managers.

Due to expansion in the business industry, and the need to reduce the use of paper, the bank is experiencing a digital transformation. As a result, the bank is developing an information system to collect and store data related to the Bank’s Branches, Employees, Loans, Bank-accounts, Cards, Customer details, Customer-transactions.

## Business Rules:

According to the above scenario, the following business rules for the information system have been established:

* The Bank must have 1 or more branches and each branch must belong to one bank.
* Each branch may have 1 or more employees, and an employee must work for one branch.
* An employee can manage one or more employees but is managed by only one employee. Therefore, the Employee entity has a relationship with itself.
* An employee must work at the bank from 01/01/2010 or afterwards and the employee’s age must be 18 or over. An employee cannot be registered on a future date.
* A branch provides 1 or more accounts, and an account must belong to a branch.
* Accounts must be opened from 01/01/2011 onwards and an account cannot be opened on a future date.
* A branch provides 1 or more loans, and a loan must belong to 1 branch.
* A Loan is issued on 01/01/2012 (2 years after the bank was established).
* A loan is issued for a minimum of 1 year and a maximum of 10 years.
* A loan must be between 2k to 150k.
* An employee may be a relationship manager of 1 or more customers. A customer must have 1 relationship manager.
* Customer age must be between 16 and 100
* A customer must live in 1 address and an address may be associated with 1 or more customers.
* In light of the fact that students have a budget limit and a minimum resource of income, a customer may be a Loan\_owner 1 or 2 times and a Loan\_owner must be 1 customer.
* A loan may be given to 1 or 2 loan\_owner and a loan\_owner must belong to 1 loan.
* A loan must have 1 or more loan payments and a loan payment must belong to 1 loan.
* loan payment is issued from 01/01/2012 (2 years after the bank was established).
* Minimum repayment must be EUR20.00
* A loan must be associated with one type of loan while a loan type must have 1 or more loans.
* A card must have a card type while a card type must be associated with 1 or more cards.
* Card number must consist of 16 characters.
* Debit cards are assigned with bin number “79658100”. (This is the prefix given to Debit cards issued for C of B Bank).
* Credit cards are assigned with bin number “79659100”. (This is the prefix given to Credit cards issued for C of B Bank).
* A card is valid for 5 years.
* Application to issue a card must be submitted from 01/01/2011.
* A Bank account may have 1 or more cards while a card must be associated with 1 account.
* A customer must have only 1 account and an account must be associated with 1 customer.
* A Bank account must belong to 1 account type and an account type must have 1 or more accounts.
* An account type must be either “saving account” or “current account”
* A Bank account may have 1 or more transactions while customer transactions must be associated with 1 Bank account.
* Transaction date must start from 01/01/2011 and must not exceed the processing date.
* A transaction must belong to 1 type of transaction while a transaction type must have 1 or more transactions.
* The bank provides 6 types of transactions.

## Legal and Ethical issues:

In a banking institution, personal data relating to employees and customers, including their financial details, are necessary to offer services tailored to the client’s needs, however, Data Protection Regulations must always be followed. Personal data includes any data relating to an identifiable person (Zahid, 2021). These regulations must be observed in all stages of data collection, storage, sharing and publication both in the present and future. Any application developed using this data must abide by the regulations of the country it will be used in, even if its development is outsourced to a different country. Companies that do not respect these rules may be fined, as in the case of the Bank of Scotland being prosecuted by the Information Commissioner’s Office (ICO0 for repeatedly faxing personal data to erroneous recipients (Palmer, 2013).

In the European Commission, a law was created in 2012 that ensures that subjects are aware of and consent to the collection and use of data about themselves. They have a right to amend this data and request its deletion entirely. Data collectors are responsible for ensuring the accuracy and safety of this data and cannot share it with third parties without the explicit consent of the subject. Failure to abide by this law makes data collectors liable to prosecution (Getta, 2022).

# ER (Logical) diagram:

Diagram

Description automatically generated

# Relational Table headings

1. Bank(bank\_name, license, address)
2. Branch(branch\_id, branch\_name, telephone, email, address, bank\_name\*)
3. Employee(employee\_id, first\_name, last\_name, phone\_number, address, start\_date, date\_of\_birth, branch\_id\*, manager\_id\*)
4. Customer(customer\_id, student\_id, first\_name, last\_name, phone\_number, date\_of\_birth, institution\_name, relationship\_manager\_id\*)
5. Customer\_address(address\_id, building\_house\_details, street, city state\_or\_region postal\_code**)**
6. Account\_type(account\_type\_id, account\_description)
7. Bank\_account(account\_number, balance, opened\_date, branch\_id\*, customer\_id\*, account\_type\_id\*)
8. Card\_type(card\_type\_id, card\_description)
9. Card(card\_no, opened\_date, expiry\_date, account\_number\*, card\_type\_id\*)
10. Transaction\_type(transaction\_type\_id, type\_description)
11. Customer\_transaction(transaction\_id, transaction\_date, amount, transaction\_description, transaction\_completion, processing\_date, account\_number\*, transaction\_type\_id\*)
12. Loan\_type(loan\_type\_id, loan\_description)
13. Loan\_owner(loan\_id, customer\_id)
14. Loan(loan\_id, amount, loan\_request\_date, close\_date, loan\_type\_id\*, branch\_id\*)
15. Loan\_payment(payment\_id, amount, payment\_date, payment\_description, loan\_id\*)

Note: The complete process of normalisation can be found in Appendix 2

# Set of Tables:

Please refer to the below create statements for all the 15 tables:

Note: The complete SQL statements (my environment) can be found in Appendix 1.

CREATE TABLE Bank (

bank\_name VARCHAR(30) NOT NULL,

license CHAR(5) NOT NULL,

address VARCHAR(100) NOT NULL,

CONSTRAINT bank\_pk PRIMARY KEY(bank\_name)

);

CREATE TABLE Branch (

branch\_id char(3) NOT NULL,

branch\_name VARCHAR(30) NOT NULL,

telephone VARCHAR(20) NOT NULL,

email VARCHAR(50) NOT NULL,

address VARCHAR(100) NOT NULL,

bank\_name VARCHAR(30) NOT NULL,

CONSTRAINT branch\_pk PRIMARY KEY(branch\_id),

CONSTRAINT bank\_name\_fk FOREIGN KEY(bank\_name) REFERENCES Bank(bank\_name),

CONSTRAINT unique\_branch UNIQUE (branch\_name, telephone, email, address)

);

CREATE TABLE Employee (

employee\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

first\_name VARCHAR(20) NOT NULL,

last\_name VARCHAR(20) NOT NULL,

phone\_number VARCHAR(20) NOT NULL UNIQUE,

address VARCHAR(100) NOT NULL,

start\_date DATE NOT NULL,

date\_of\_birth DATE NOT NULL,

branch\_id CHAR(3) NOT NULL,

manager\_id NUMBER,

CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id),

CONSTRAINT employee\_branch\_id\_fk FOREIGN KEY(branch\_id) REFERENCES Branch(branch\_id),

CONSTRAINT employee\_manager\_fk FOREIGN KEY(manager\_id) REFERENCES Employee(employee\_id)

);

CREATE TABLE Customer\_address(

address\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

building\_house\_details VARCHAR(100) NOT NULL,

street VARCHAR(100) NOT NULL,

city VARCHAR(30) NOT NULL,

state\_or\_region VARCHAR(100) NOT NULL,

postal\_code CHAR(8) NOT NULL,

CONSTRAINT address\_id\_pk PRIMARY KEY(address\_id)

);

CREATE TABLE Customer(

customer\_id VARCHAR(10),

student\_id NUMBER(7) NOT NULL CHECK (student\_id > 0),

first\_name VARCHAR(20) NOT NULL,

last\_name VARCHAR(20) NOT NULL,

address\_id NUMBER NOT NULL,

phone\_number VARCHAR(20) NOT NULL,

date\_of\_birth DATE NOT NULL,

institution\_name VARCHAR(20) NOT NULL,

relationship\_manager\_id NUMBER NOT NULL,

CONSTRAINT customer\_id\_pk PRIMARY KEY(customer\_id),

CONSTRAINT employee\_id\_fk FOREIGN KEY(relationship\_manager\_id) REFERENCES Employee(employee\_id),

CONSTRAINT unique\_customer UNIQUE (student\_id, phone\_number),

CONSTRAINT customer\_address\_id\_fk FOREIGN KEY(address\_id) REFERENCES Customer\_address(address\_id)

);

CREATE TABLE Account\_type(

account\_type\_id CHAR(1),

account\_description VARCHAR(30) NOT NULL UNIQUE,

CONSTRAINT account\_type\_id\_pk PRIMARY KEY(account\_type\_id),

CONSTRAINT check\_account\_type CHECK (account\_type\_id BETWEEN 1 AND 2)

);

CREATE TABLE Bank\_account(

account\_number NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 10001,

balance NUMBER NOT NULL,

opened\_date DATE NOT NULL,

branch\_id CHAR(3) NOT NULL,

customer\_id VARCHAR(10) NOT NULL UNIQUE,

account\_type\_id CHAR(1) NOT NULL,

CONSTRAINT account\_number\_pk PRIMARY KEY(account\_number),

CONSTRAINT branch\_id\_fk FOREIGN KEY(branch\_id) REFERENCES Branch(branch\_id),

CONSTRAINT ac\_customer\_id\_fk FOREIGN KEY(customer\_id) REFERENCES Customer(customer\_id),

CONSTRAINT account\_type\_id\_fk FOREIGN KEY(account\_type\_id) REFERENCES Account\_type(account\_type\_id)

);

CREATE TABLE Card\_type(

card\_type\_id CHAR(1),

card\_description VARCHAR(30) NOT NULL UNIQUE,

CONSTRAINT card\_type\_id\_pf PRIMARY KEY(card\_type\_id),

CONSTRAINT check\_card\_type CHECK (card\_type\_id BETWEEN 1 AND 2)

);

CREATE TABLE Card(

card\_no CHAR(16),

opened\_date DATE DEFAULT sysdate,

expiry\_date DATE NOT NULL,

card\_type\_id CHAR(1) NOT NULL,

account\_number NUMBER NOT NULL,

CONSTRAINT card\_no\_pk PRIMARY KEY(card\_no),

CONSTRAINT card\_type\_id\_fk FOREIGN KEY(card\_type\_id) REFERENCES Card\_type(card\_type\_id),

CONSTRAINT check\_debit\_credit\_cards

CHECK ((card\_no LIKE '79658100%' AND card\_type\_id = '1') OR (card\_no LIKE '79659100%' AND card\_type\_id = '2')),

CONSTRAINT card\_account\_number\_fk FOREIGN KEY(account\_number) REFERENCES Bank\_account(account\_number)

);

CREATE TABLE Transaction\_type(

transaction\_type\_id CHAR(1),

type\_description VARCHAR(20) NOT NULL UNIQUE,

CONSTRAINT transaction\_type\_id\_pk PRIMARY KEY(transaction\_type\_id),

CONSTRAINT check\_transaction\_type CHECK (transaction\_type\_id BETWEEN 1 AND 6)

);

CREATE TABLE Customer\_transaction(

transaction\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

transaction\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

amount NUMBER NOT NULL,

transaction\_description VARCHAR(100) NOT NULL,

transaction\_completion CHAR(1) NOT NULL,

transaction\_type\_id CHAR(1),

account\_number NUMBER NOT NULL,

processing\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

CONSTRAINT transaction\_id\_pk PRIMARY KEY(transaction\_id),

CONSTRAINT transaction\_type\_id\_fk FOREIGN KEY(transaction\_type\_id) REFERENCES Transaction\_type(transaction\_type\_id),

CONSTRAINT tr\_account\_number\_fk FOREIGN KEY(account\_number) REFERENCES Bank\_account(account\_number),

CONSTRAINT check\_transaction\_id CHECK (transaction\_id > 0),

CONSTRAINT check\_transaction\_date\_start CHECK (transaction\_date >= '01-JAN-2011 01:00:00'),

CONSTRAINT check\_transaction\_date\_end CHECK (transaction\_date <= processing\_date),

CONSTRAINT check\_transaction\_status CHECK (transaction\_completion = 'N' OR transaction\_completion = 'Y')

);

CREATE TABLE Loan\_type(

loan\_type\_id CHAR(1),

loan\_description VARCHAR(50) NOT NULL UNIQUE,

CONSTRAINT loan\_type\_id\_pk PRIMARY KEY(loan\_type\_id),

CONSTRAINT check\_loan\_type CHECK (loan\_type\_id BETWEEN 1 AND 5)

);

CREATE TABLE Loan(

loan\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

amount DECIMAL NOT NULL,

loan\_type\_id CHAR(1) NOT NULL,

branch\_id char(3) NOT NULL,

loan\_request\_date DATE NOT NULL,

close\_date DATE DEFAULT sysdate+365,

CONSTRAINT loan\_id\_pk PRIMARY KEY(loan\_id),

CONSTRAINT check\_loan\_date --loan issued 2 years after bank established

CHECK (loan\_request\_date >= TO\_DATE('2012-01-01','YYYY-MM-DD')),

CONSTRAINT check\_loan\_close\_date CHECK (close\_date BETWEEN loan\_request\_date+365 AND loan\_request\_date+365\*10),

CONSTRAINT loan\_type\_id\_fk FOREIGN KEY(loan\_type\_id) REFERENCES Loan\_type(loan\_type\_id),

CONSTRAINT loan\_branch\_id\_fk FOREIGN KEY(branch\_id) REFERENCES Branch(branch\_id),

CONSTRAINT check\_loan\_amount CHECK (amount BETWEEN 2000 AND 150000)

);

CREATE TABLE Loan\_owner (

loan\_id NUMBER,

customer\_id VARCHAR(10),

CONSTRAINT loan\_customer\_ids\_pk PRIMARY KEY(loan\_id, customer\_id),

CONSTRAINT loan\_owner\_id\_fk FOREIGN KEY(loan\_id) REFERENCES Loan(loan\_id),

CONSTRAINT customer\_loan\_id\_fk FOREIGN KEY(customer\_id) REFERENCES Customer(customer\_id)

);

CREATE TABLE Loan\_payment(

payment\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,

amount DECIMAL NOT NULL,

payment\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

payment\_description VARCHAR(50) NOT NULL,

loan\_id NUMBER NOT NULL,

CONSTRAINT payment\_id\_pk PRIMARY KEY(payment\_id),

CONSTRAINT loan\_id\_fk FOREIGN KEY(loan\_id) REFERENCES Loan(loan\_id),

/\*\* loan payment issued 2 years after bank established

minimum repayment must be EUR20.00

\*\*/

CONSTRAINT check\_payment\_date CHECK (payment\_date >= TO\_DATE('2012-01-01','YYYY-MM-DD') AND amount > 20)

);

# Display inserted Data:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SELECT \* FROM Customer\_transaction;**   |  | | --- | | TRANSACTION\_ID TRANSACTION\_DATE                 AMOUNT TRANSACTION\_DESCRIPTION     TRANSACTION\_COMPLETION TRANSACTION\_TYPE\_ID  ACCOUNT\_NUMBER PROCESSING\_DATE | | -------------- ---------------------------- ---------- --------------------------- ---------------------- -------------------- -------------- ---------------------------- | | 1 26-JAN-20 14.03.50.000000000        300 Deposit-360ATM              Y                      2                             10001 03-DEC-22 19.10.37.380387000 | | 2 05-MAY-20 22.01.46.000000000    -348.94 Ophth counsel and instruct  Y                      3                             10002 03-DEC-22 19.10.37.380387000 | | 3 01-JUN-20 13.08.00.000000000    -411.03 Revision of lead            Y                      5                             10003 03-DEC-22 19.10.37.380387000 | | 4 20-JUL-20 22.03.03.000000000       -170 Withdrawal-360ATM           N                      1                             10004 03-DEC-22 19.10.37.380387000 | | 5 09-JUN-20 15.08.22.000000000     -38.95 Cholangiogram NEC           N                      4                             10005 03-DEC-22 19.10.37.380387000 | | 6 21-APR-20 16.02.32.000000000    -835.45 Bronch/lung dx proc NEC     Y                      5                             10006 03-DEC-22 19.10.37.380387000 | | 7 22-MAY-20 19.06.23.000000000       -400 Withdrawal-365ATM           N                      1                             10007 03-DEC-22 19.10.37.380387000 | | 8 18-JUL-20 21.07.18.000000000        580 Deposit-362ATM              Y                      2                             10008 03-DEC-22 19.10.37.380387000 | | 9 14-APR-20 17.09.54.000000000       -340 Withdrawal-365ATM           Y                      1                             10009 03-DEC-22 19.10.37.380387000 | | 10 20-AUG-20 21.07.18.000000000        700 Deposit-360ATM              Y                      2                             10008 03-DEC-22 19.10.37.380387000 | | 11 21-AUG-20 21.09.00.000000000       -100 TK Maxx                     Y                      6                             10008 03-DEC-22 19.10.37.380387000 | | 12 21-APR-20 16.02.32.000000000    1500.45 April Salary-ABC ltd        Y                      4                             10006 03-DEC-22 19.10.37.380387000 | | 13 10-JUN-20 13.08.00.000000000     300.03 Revision - Refund           Y                      4                             10003 03-DEC-22 19.10.37.380387000 | | 14 17-DEC-20 14.06.38.000000000     -88.63 Cardiotomy                  N                      3                             10010 03-DEC-22 19.10.37.380387000 | | **SELECT \* FROM Bank\_account;** | | ACCOUNT\_NUMBER BALANCE OPENED\_DATE BRANCH\_ID CUSTOMER\_ID ACCOUNT\_TYPE\_ID | | -------------- ------- ----------- --------- ----------- --------------- | | 10001 8440.13 17-NOV-2016 018       12001M      1 | | 10002 4035.14 22-SEP-2019 030       42195M      1 | | 10003 4744.44 05-JUN-2021 052       21802D      1 | | 10004 2821.74 14-JUN-2021 052       65076N      1 | | 10005 6692.65 28-SEP-2018 074       76945M      1 | | 10006 1388.89 09-FEB-2012 074       41142G      1 | | 10007  650.86 13-JUN-2018 041       47966M      1 | | 10008 6030.72 11-OCT-2021 041       74020M      1 | | 10009 9304.48 05-FEB-2022 018       42943M      2 | | 10010 8454.91 30-NOV-2021 074       28735M      2 | | **SELECT \* FROM Loan\_payment;** | | PAYMENT\_ID     AMOUNT PAYMENT\_DATE                 PAYMENT\_DESCRIPTION       LOAN\_ID | | ---------- ---------- ---------------------------- ------------------------- ------- | | 1        200 13-SEP-20 13.00.00.000000000 Septemberpayment\_1              1 | | 2        800 16-FEB-20 15.00.00.000000000 February payment\_2              1 | | 3        500 12-MAR-20 16.00.00.000000000 March payment\_1                 2 | | 4        360 14-JUL-20 21.00.00.000000000 April payment\_1                 3 | | 5        150 05-APR-20 14.00.00.000000000 April payment\_1                 4 | | 6        120 19-APR-20 20.00.00.000000000 April payment\_1                 6 | | 7        140 23-JUL-20 21.00.00.000000000 July payment\_2                  6 | | 8        320 08-JUN-20 15.00.00.000000000 payment\_1                       7 | | 9        220 22-SEP-20 18.00.00.000000000  September payment\_1            4 | | 10        130 13-SEP-20 20.00.00.000000000 September payment\_2             2 | | **SELECT \* FROM Loan;** | | LOAN\_ID     AMOUNT LOAN\_TYPE\_ID BRANCH\_ID LOAN\_REQUEST\_DATE CLOSE\_DATE | | ------- ---------- ------------ --------- ----------------- ----------- | | 1       4608 1            018       13-AUG-2013       26-APR-2015 | | 2       8712 2            030       16-JAN-2015       01-OCT-2019 | | 3       4340 3            030       12-FEB-2016       14-FEB-2020 | | 4      11424 4            052       14-JUN-2021       17-AUG-2023 | | 5      12763 2            074       19-MAR-2020       30-SEP-2024 | | 6      14988 4            074       23-JUN-2021       29-JUN-2025 | | 7       3278 3            041       08-MAY-2015       14-MAY-2018 | | 8       2212 4            041       22-AUG-2018       25-NOV-2020 | | **SELECT \* FROM Loan\_owner;** | | LOAN\_ID CUSTOMER\_ID | | ------- ----------- | | 1 12001M | | 1 42943M | | 2 42195M | | 3 42195M | | 4 21802D | | 5 65076N | | 6 41142G | | 6 76945M | | 7 74020M | | 8 47966M | | **SELECT \* FROM Transaction\_type;** | | TRANSACTION\_TYPE\_ID  TYPE\_DESCRIPTION | | -------------------- ------------------- | | 1                    Withdrawal | | 2                    Deposit | | 3                    Bill Payments | | 4                    SEPA Transfer | | 5                    NON-SEPA Transfer | | 6                    Purchase | | **SELECT \* FROM Account\_type;** | | ACCOUNT\_TYPE\_ID ACCOUNT\_DESCRIPTION | | --------------- ------------------- | | 1               Current Account | | 2               Saving Account | |  | | **SELECT \* FROM Card;**  CARD\_NO          OPENED\_DATE EXPIRY\_DATE CARD\_TYPE\_ID ACCOUNT\_NUMBER | | ---------------- ----------- ----------- ------------ -------------- | | 7965810014246618 01-JAN-2011 01-JAN-2016 1                     10001 | | 7965810014246686 01-MAY-2012 01-MAY-2017 1                     10002 | | 7965810014246378 21-SEP-2013 21-SEP-2018 1                     10003 | | 7965810014246619 14-JAN-2016 14-JAN-2021 1                     10004 | | 7965910014246770 01-APR-2017 01-APR-2022 2                     10004 | | 7965910014246771 01-JAN-2019 01-JAN-2024 2                     10005 | | 7965910014246772 16-MAY-2013 16-MAY-2018 2                     10007 | | 7965910014246773 26-OCT-2018 26-OCT-2023 2                     10001 | | 7965910014246774 11-NOV-2018 11-NOV-2023 2                     10009 | | 7965910014246775 14-FEB-2014 14-FEB-2019 2                     10010 | | **SELECT \* FROM Card\_type;** | | CARD\_TYPE\_ID CARD\_DESCRIPTION | | ------------ ---------------- | | 1            Debit Card | | 2            Credit Card | | **SELECT \* FROM Loan\_type;** | | LOAN\_TYPE\_ID LOAN\_DESCRIPTION | | ------------ ---------------- | | 1            Personal Loan | | 2            Joint Loan | | 3            Home Loan | | 4            Vehicle Loan | | **SELECT \* FROM Customer\_address;** | | ADDRESS\_ID BUILDING\_HOUSE\_DETAILS      STREET          CITY             STATE\_OR\_REGION           POSTAL\_COD | | ---------- --------------------------- --------------- ---------------- ------------------------- ---------- | | 1 18 Acrantophis              Henley Rd       Reading          West Berkshire            OA8 1IJ | | 2 152 Neophron percnopterus   High St         Kenilworth       Warwickshire              WN3 8SL | | 3 11 Phalacrocorax niger      Radcliffe Rd    West Bridgford   Nottinghamshire           WA8 6DJ | | 4 38 Phacochoerus aethiopus   Baxtergate      Harlow           North Yorkshire           YO21 1BN | | 5 68 Phascogale calura        Broadley Rd     Harlow           Essex                     CM19 5RD | | 6 501 Cercatetus concinnus    Jonathans       Milton Keynes    Buckinghamshire County    MK6 5DF | | 7 311 Bradypus tridactylus    Stoke Rd        Guildford        Surrey                    GU1 1EZ | | 8 141 Eolophus roseicapillus  Cheltenham Rd   Guildford        Avon                      BS6 5RW | | 9 28 Connochaetus taurinus    30 Easby Rd     Bradford         West Yorkshire            BD7 1QX | | 10 50 Manouria emys            Christchurch Rd Milton Keynes    Bucking                   SO23 9SU | | **SELECT \* FROM Customer;** | | CUSTOMER\_ID STUDENT\_ID FIRST\_NAME  LAST\_NAME   ADDRESS\_ID PHONE\_NUMBER         DATE\_OF\_BIR INSTITUTION\_NAME RELATIONSHIP\_MANAGER\_ID | | ----------- ---------- ----------- ----------- ---------- -------------------- ----------- ---------------- ----------------------- | | 42943M           21795 Hermione    Harwin               1 +62 (964) 166-7364   10-MAR-1983 Devcast                                4 | | 12001M           21587 Carolina    Marin                2 +63 (967) 105-0384   10-FEB-1999 Oyonder                                5 | | 42195M           70849 Merrel      Gillian              3 +1 (209) 220-4172    29-APR-1994 Skibox                                 7 | | 21802D           46906 Ruthanne    Nimmo                4 +62 (454) 265-7679   13-SEP-1972 Tazzy                                  8 | | 65076N           58179 Delano      Stormes              5 +63 (532) 351-6819   12-AUG-1982 Wikibox                                9 | | 76945M           41363 Edwin       Cuthbert             6 +84 (350) 292-4089   26-NOV-1997 Brainlounge                           10 | | 41142G           47107 Keelby      Dunkirk              7 +54 (876) 720-5474   27-AUG-1979 Devpoint                              11 | | 47966M           20966 Oran        Floyed               8 +1 (314) 579-1317    07-JAN-1975 Devpoint                              12 | | 74020M           99256 Aleksandr   Martell              9 +86 (110) 613-7576   07-FEB-1993 Kayveo                                13 | | 28735M           16703 Thurstan    Schubbert           10 +63 (245) 355-0857   14-AUG-1990 Browsebug                             10 | | **SELECT \* FROM Employee;** | | |  | | --- | | EMPLOYEE\_ID FIRST\_NAME  LAST\_NAME   PHONE\_NUMBER         ADDRESS                                              START\_DATE  DATE\_OF\_BIR BRA MANAGER\_ID | | ----------- ----------- ----------- -------------------- ---------------------------------------------------- ----------- ----------- --- ---------- | | 1 Lionel      Messi       +86 (707) 560-9523   45 Southlands, Blaina,NP13 3JN, UK                   12-MAR-2014 02-MAY-1977 502 | | 2 Ethelyn     Kauffman    +351 (142) 478-4871  12 Shaw Court, Malmesbury Road, Morden,SM4 6HH, UK   25-JAN-2012 20-SEP-1972 502          1 | | 3 Jessie      Lintott     +55 (442) 339-5212   2 Brook Cottages, Great Wolford,CV36 5NP, UK         16-NOV-2013 10-JUN-1998 502          2 | | 4 Delinda     Geerling    +86 (707) 560-9529   45 Southlands, Blaina,NP13 3JN, UK                   12-MAR-2014 02-MAY-1977 502          3 | | 5 Jessie      Lintott     +351 (142) 478-4875  12 Shaw Court, Malmesbury Road, Morden,SM4 6HH, UK   25-JAN-2012 20-SEP-1972 018          2 | | 6 Jaye        Abrams      +269 (910) 803-4376  11 Squires Mount, London,NW3 1ED, UK                 15-DEC-2020 06-JUN-1980 018          5 | | 7 Garrick     Parkes      +62 (716) 795-4927   88 Exley Road, Keighley,BD21 1LT, UK                 27-NOV-2014 29-SEP-1982 030          2 | | 8 Jaye        Abrams      +81 (495) 229-2559   36 The Kings Gap, Hoylake,CH47 1HF, UK               24-JUN-2022 03-AUG-1982 030          7 | | 9 Konstantine Anselm      +389 (884) 109-0248  67469 Packers Avenue                                 10-MAR-2010 30-MAY-1972 052          2 | | 10 Kain        Cuppleditch +1 (802) 831-5499    Fairways, Carlisle Road, Longtown,CA6 5SQ, UK        27-SEP-2020 05-MAR-1987 052          9 | | 11 Cale        Larkworthy  +420 (927) 525-4003  165 - 167 Garstang Road, Fulwood,PR2 3BH, UK         17-JUN-2013 22-OCT-1982 074          2 | | 12 Steven      Whatmough   +1 (823) 836-0450    Southview, Bowlers Green, Magdalen Laver,CM5 0ET, UK 04-NOV-2015 30-NOV-1974 074         11 | | 13 Sharona     Cordell     +358 (694) 526-2457  6 Lime Tree Place, St. Albans,AL1 3BD, UK            09-AUG-2017 12-JUN-1978 041          2 | | 14 Steven      Whatmough   +33 (701) 233-4336   8 Mill Hill, Cleator Moor,CA25 5SH, UK               05-JAN-2017 08-DEC-1997 041         13 | | 15 Juline      Quiddinton  +351 (972) 278-4623  10 Rose Garden Lane, Wynyard,TS22 5WB, UK            27-AUG-2019 03-SEP-1988 096          2 | | 16 Sula        Gibbie      +86 (635) 525-8442   Flat 6, Coram House, Wood Street, London,W4 2JW, UK  31-MAY-2019 16-NOV-1974 096         15 | | 17 Noah        Odger       +55 (775) 681-9220   Flat 3, The Coach House, Red Wharf Bay,LL75 8RJ, UK  13-JAN-2022 02-FEB-1974 802          2 | | 18 Tulley      Coie        +63 (328) 430-4413   12 Fitzpiers, Saffron Walden,CB10 2BD, UK            22-AUG-2019 12-JUL-1988 802         17 |   **SELECT \* FROM Branch;**  BRANCH\_ID BRANCH\_NAME            TELEPHONE            EMAIL                   ADDRESS                                  BANK\_NAME | | --------- ---------------------- -------------------- ----------------------- ---------------------------------------- --------- | | 502       Head Office            +4429101000          h.o@cofb.bank           45, Bakery Street, London, UK            C of B | | 018       chelsea anteater       +86 (898) 758-2189   wholdall0@cofb.bank     544 Twin Pines Court, Liverpool, UK      C of B | | 074       Westham  eagle         +258 (604) 650-6700  rwynett2@cofb.bank      66 Lunder Park, Manchister, UK           C of B | | 096       Manchister crocodile   +33 (593) 323-5644   pagney3@cofb.bank       84 Comanche Crossing, Wolverhapton, UK   C of B | | 052       Bermingham lizard      +53 (276) 814-0053   arumsby4@cofb.bank      51664 Blaine Road, Bermingham, UK        C of B | | 802       Wolverhampton          +1 (826) 180-3016    gsmartman5@cofb.bank    8 Autumn Leaf Lane, Westham, UK          C of B | | 041       Leeds, ring-tailed     +86 (155) 974-4047   dcullinane6@cofb.bank   94 Summit Park, Fullham, UK              C of B | | 030       Blackburn rhinoceros   +380 (353) 967-4134  gdewhirst9@cofb.bank    6 Lien Center, Belfast, UK               C of B | | **SELECT \* FROM Bank;** | | BANK\_NAME LICENSE ADDRESS | | --------- ------- ---------------------------------------- | | C of B    BL865   45, Bakery Street, London, UK | |

# SQL Queries

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Query 1:**  Query to list transactionID, amount and description from customerTrassaction table + accountNumber and balance from Bank\_account table + customerID, firstName and lastName from customer table for those transactions which are between -300 to 300  **SELECT**  **cust\_tr.transaction\_id AS tr\_id,**  **cust\_tr.amount AS tr\_amount,**  **cust\_tr.transaction\_description AS tr\_description,**  **ac.account\_number AS ac\_no,**  **ac.balance AS ac\_balance,**  **cust.customer\_id AS cust\_id,**  **cust.first\_name AS first\_name,**  **cust.last\_name AS last\_name**  **FROM Customer\_transaction cust\_tr**  **LEFT JOIN Bank\_account ac**  **ON cust\_tr.account\_number = ac.account\_number**  **LEFT JOIN Customer cust**  **ON ac.customer\_id = cust.customer\_id**  **WHERE cust\_tr.amount BETWEEN -300 AND 300**  **ORDER BY amount;**  **Results:**   |  |  | | --- | --- | | TR\_ID  TR\_AMOUNT TR\_DESCRIPTION         AC\_NO AC\_BALANCE CUST\_ID    FIRST\_NAME LAST\_NAME |  | | ----- ---------- -------------------- ------- ---------- ---------- ---------- ---------- |  | | 4       -170 Withdrawal-360ATM      10004    2821.74 65076N     Delano     Stormes |  | | 11       -100 TK Maxx                10008    6030.72 74020M     Aleksandr  Martell |  | | 14     -88.63 Cardiotomy             10010    8454.91 28735M     Thurstan   Schubbert |  | | 5     -38.95 Cholangiogram NEC      10005    6692.65 76945M     Edwin      Cuthbert |  | | 1        300 Deposit-360ATM         10001    8440.13 12001M     Carolina   Marin |  | | **Query 2:**  Query to count the number of different cities of the bank's customers  **SELECT**  **COUNT(DISTINCT city) AS "Number of cities"**  **FROM customer\_address;**  **Results:**  Number of cities |  | | ---------------- |  | | 7 |  | | **Query 3:**  Query to list loanID, amount and request date from loan table + customerID from loan owner table + customer date of birth  from customer table  Note: TO\_CHAR function was used on the date columns to convert the date format to DD/MM/YYYY  **SELECT**  **lnn.loan\_id AS loanID,**  **lnn.amount AS amount,**  **TO\_CHAR(lnn.loan\_request\_date, 'DD/MM/YYYY') AS request\_Date,**  **ln\_ow.customer\_id AS customerID,**  **TO\_CHAR(cust.date\_of\_birth, 'DD/MM/YYYY') AS customer\_DOB**  **FROM loan lnn**  **LEFT JOIN loan\_owner ln\_ow**  **ON lnn.loan\_id = ln\_ow.loan\_id**  **LEFT JOIN customer cust**  **ON ln\_ow.customer\_id = cust.customer\_id;**  **Results:**  LOANID   AMOUNT REQUEST\_DATE  CUSTOMERID CUSTOMER\_DOB |  | | ------ -------- ------------- ---------- ------------ |  | | 1     4608 13/08/2013    42943M     10/03/1983 |  | | 1     4608 13/08/2013    12001M     10/02/1999 |  | | 2     8712 16/01/2015    42195M     29/04/1994 |  | | 3     4340 12/02/2016    42195M     29/04/1994 |  | | 4    11424 14/06/2021    21802D     13/09/1972 |  | | 5    12763 19/03/2020    65076N     12/08/1982 |  | | 6    14988 23/06/2021    76945M     26/11/1997 |  | | 6    14988 23/06/2021    41142G     27/08/1979 |  | | 8     2212 22/08/2018    47966M     07/01/1975 |  | | 7     3278 08/05/2015    74020M     07/02/1993 |  | | **Question 4:**  Query to list the total of credit transactions on every customer that is grouped by account number and customer ID and ordered by total of credits in descending order.  The list displaying columns: amount from Customer\_transaction table + account\_number from Bank\_account table + customer\_id from customer table  **SELECT**  **SUM(ct.amount) AS totalCredits,**  **ac.account\_number AS accountNO,**  **c.customer\_id AS customerID**  **FROM Customer\_transaction ct**  **LEFT JOIN Bank\_account ac**  **ON ct.account\_number = ac.account\_number**  **LEFT JOIN Customer c**  **ON ac.customer\_id = c.customer\_id**  **WHERE ct.amount > 0**  **GROUP BY**  **ac.account\_number,**  **c.customer\_id**  **ORDER BY totalCredits DESC;** |  | | **Results:**  TOTALCREDITS  ACCOUNTNO CUSTOMERID |  | | ------------ ---------- ---------- |  | | 1500.45      10006 41142G |  | | 1280      10008 74020M |  | | 300.03      10003 21802D |  | | 300      10001 12001M |  | | **Question 5:**  Query to count and group transaction types. Basically, a list to display the number of transactions per transaction type  **SELECT**  **trtp.type\_description AS transactionTypes,**  **COUNT(ctr.transaction\_type\_id) AS counts**  **FROM Transaction\_type trtp**  **LEFT JOIN Customer\_transaction ctr**  **ON trtp.transaction\_type\_id = ctr.transaction\_type\_id**  **GROUP BY**  **trtp.transaction\_type\_id,**  **trtp.type\_description**;  **Results:**  TRANSACTIONTYPES     COUNTS |  | | -------------------- ------ |  | | Deposit                   3 |  | | NON-SEPA Transfer         2 |  | | Purchase                  1 |  | | Bill Payments             2 |  | | Withdrawal                3 |  | | SEPA Transfer             3 |  | | **Question 6:**  Query to list branches of those employees whose manager ID is 2.  **SELECT**  **br.branch\_id AS branchID,**  **br.branch\_name AS branchName**  **FROM Branch br**  **WHERE branch\_id IN**  **(SELECT branch\_id FROM Employee WHERE manager\_id = 2);**  **Results:** |  | | BRANCHID BRANCHNAME |  | | -------- ----------------------- |  | | 502      Head Office |  | | 018      chelsea anteater |  | | 030      Blackburn rhinoceros |  | | 052      Bermingham lizard |  | | 074      Westham  eagle |  | | 041      Leeds, ring-tailed |  | | 096      Manchister crocodile |  | | 802      Wolverhampton |  | | |

# Testing:

**TEST 1**

DELETE FROM LOAN

WHERE loan\_id = 1

**Result:**

Graphical user interface, text, application

Description automatically generated

The above query test was used to delete a tuple from the parent table which is “loan” table.

The query failed since the Primary key was referenced by a foreign key in a child table which is the” Loan\_owner” table. The error indicates that the foreign key constraint name “loan\_owner\_id\_fk” is being violated in the “Loan\_owner” table.

**Test 2**

INSERT INTO Loan\_owner (loan\_id, customer\_id)

VALUES (9, ‘47966M’);

**Result:**

Graphical user interface, text

Description automatically generated

The above query test was used to insert a record into "Loan\_owner” table.

The query failed to execute successfully and the above error is stating that the chosen foreign key value (9) has no matching primary key value in the parent table “Loan”.

**Test 3**

DELETE FROM loan\_owner

WHERE loan\_id = 8;

ROLLBACK; **/\*\* In order to refrain from any changes, Rollback was used to undo the above row deletion\*\*/**

**Result:**

Graphical user interface, text, application

Description automatically generated

The above query test was used to delete a record from "loan\_owner” table.

The result was successful since the affected field was foreign key in loan\_owner which is the child table.

**Test 4**

UPDATE Bank\_account

SET customer\_id = '123457KL'

WHERE customer\_id = '47966M';

**Result:**

Graphical user interface, text, application

Description automatically generated

The above query test was used to update a record from " Bank\_account” table.

The query failed to execute successfully, and the above error is stating that the chosen foreign key value ‘123457KL’ has no matching primary key value in the parent table “Customer”.

**Test 5**

UPDATE Bank\_account

SET balance = 9000

WHERE account\_number = 10010;

ROLLBACK; **/\*\* In order to refrain from any changes, Rollback was used to undo the above row update\*\*/**

**Result:**

Graphical user interface, text, application

Description automatically generated

The above query test was used to update a record from " bank\_account” table.

The result was successful since the affected field “account\_number” was a valid primary key in the parent table " bank\_account”.

**Test 6**

INSERT INTO Customer

(customer\_id, student\_id, first\_name, last\_name, address\_id, phone\_number, date\_of\_birth, institution\_name, relationship\_manager\_id)

VALUES('42943M', 21700, 'Jack', 'Sparrow',1, '+62 (964) 166-7000', '10-Mar-1993', 'Devcast', 4);

Result:

Graphical user interface, text, application

Description automatically generated

The above query test was used to insert a new customer record into the “Customer” table.

The query failed since the value used on the primary key, the customer\_id: ‘'42943M'’ already exists in the table.

**Test 7:**

**INSERT INTO Customer**

**(**customer\_id, student\_id, first\_name, last\_name, address\_id, phone\_number, date\_of\_birth, institution\_name, relationship\_manager\_id)

VALUES('12345M', 21700, 'Jack', 'Sparrow', 1, '+62 (964) 166-7000', '10-Mar-1993', 'Devcast', 4);

ROLLBACK; **/\*\* In order to refrain from any changes, Rollback was used to undo the above row insert\*\*/**

**Result:**

**Graphical user interface, application

Description automatically generated**

The above query test was used to insert a new customer record into the “Customer” table.

The query was successful since the customer\_id attribute which is the Primary key was a unique value.

# Data Dictionary:

**Bank:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| BANK\_NAME | VARCHAR2(30 BYTE) | No | null | 1 |
| LICENSE | CHAR(5 BYTE) | No | null | 2 |
| ADDRESS | VARCHAR2(100 BYTE) | No | null | 3 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| BANK\_PK | Primary\_Key | null |
| SYS\_C00266810 | Check | "BANK\_NAME" IS NOT NULL |
| SYS\_C00266811 | Check | "LICENSE" IS NOT NULL |
| SYS\_C00266812 | Check | "ADDRESS" IS NOT NULL |

**Account type:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| ACCOUNT\_TYPE\_ID | CHAR(1 BYTE) | No | null | 1 |
| ACCOUNT\_DESCRIPTION | VARCHAR2(30 BYTE) | No | null | 2 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| ACCOUNT\_TYPE\_ID\_PK | Primary\_Key | null |
| CHECK\_ACCOUNT\_TYPE | Check | account\_type\_id BETWEEN 1 AND 2 |
| SYS\_C00266855 | Check | "ACCOUNT\_DESCRIPTION" IS NOT NULL |
| SYS\_C00266858 | Unique | null |

**Bank\_account:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| ACCOUNT\_NUMBER | NUMBER | No | "OPS$2123147"."ISEQ$$\_413860".nextval | 1 |
| BALANCE | NUMBER | No | null | 2 |
| OPENED\_DATE | DATE | No | null | 3 |
| BRANCH\_ID | CHAR(3 BYTE) | No | null | 4 |
| CUSTOMER\_ID | VARCHAR2(10 BYTE) | No | null | 5 |
| ACCOUNT\_TYPE\_ID | CHAR(1 BYTE) | No | null | 6 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| ACCOUNT\_NUMBER\_PK | Primary\_Key | null |
| ACCOUNT\_TYPE\_ID\_FK | Foreign\_Key | null |
| AC\_CUSTOMER\_ID\_FK | Foreign\_Key | null |
| BRANCH\_ID\_FK | Foreign\_Key | null |
| SYS\_C00266859 | Check | "ACCOUNT\_NUMBER" IS NOT NULL |
| SYS\_C00266860 | Check | "BALANCE" IS NOT NULL |
| SYS\_C00266861 | Check | "OPENED\_DATE" IS NOT NULL |
| SYS\_C00266862 | Check | "BRANCH\_ID" IS NOT NULL |

**Branch:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| BRANCH\_ID | CHAR(3 BYTE) | No | null | 1 |
| BRANCH\_NAME | VARCHAR2(30 BYTE) | No | null | 2 |
| TELEPHONE | VARCHAR2(20 BYTE) | No | null | 3 |
| EMAIL | VARCHAR2(50 BYTE) | No | null | 4 |
| ADDRESS | VARCHAR2(100 BYTE) | No | null | 5 |
| BANK\_NAME | VARCHAR2(30 BYTE) | No | null | 6 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| BANK\_NAME\_FK | Foreign\_Key | null |
| BRANCH\_PK | Primary\_Key | null |
| SYS\_C00266814 | Check | "BRANCH\_ID" IS NOT NULL |
| SYS\_C00266815 | Check | "BRANCH\_NAME" IS NOT NULL |
| SYS\_C00266816 | Check | "TELEPHONE" IS NOT NULL |
| SYS\_C00266817 | Check | "EMAIL" IS NOT NULL |
| SYS\_C00266818 | Check | "ADDRESS" IS NOT NULL |
| SYS\_C00266819 | Check | "BANK\_NAME" IS NOT NULL |
| UNIQUE\_BRANCH | Unique | null |

**Card:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| CARD\_NO | CHAR(16 BYTE) | No | null | 1 |
| OPENED\_DATE | DATE | Yes | sysdate | 2 |
| EXPIRY\_DATE | DATE | No | null | 3 |
| CARD\_TYPE\_ID | CHAR(1 BYTE) | No | null | 4 |
| ACCOUNT\_NUMBER | NUMBER | No | null | 5 |

**Constraints:**

|  |  |  |  |
| --- | --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | | SEARCH\_CONDITION |
| CARD\_ACCOUNT\_NUMBER\_FK | Foreign\_Key | null | |
| CARD\_NO\_PK | Primary\_Key | null | |
| CARD\_TYPE\_ID\_FK | Foreign\_Key | null | |
| CHECK\_DEBIT\_CREDIT\_CARDS | Check | (card\_no LIKE '79658100%' AND card\_type\_id = '1') OR   (card\_no LIKE '79659100%' AND card\_type\_id = '2') | |
| SYS\_C00266874 | Check | "EXPIRY\_DATE" IS NOT NULL | |
| SYS\_C00266875 | Check | "CARD\_TYPE\_ID" IS NOT NULL | |
| SYS\_C00266876 | Check | "ACCOUNT\_NUMBER" IS NOT NULL | |

**Card\_type:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| CARD\_TYPE\_ID | CHAR(1 BYTE) | No | null | 1 |
| CARD\_DESCRIPTION | VARCHAR2(30 BYTE) | No | null | 2 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| ONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| CARD\_TYPE\_ID\_PF | Primary\_Key | null |
| CHECK\_CARD\_TYPE | Check | card\_type\_id BETWEEN 1 AND 2 |
| SYS\_C00266870 | Check | "CARD\_DESCRIPTION" IS NOT NULL |
| SYS\_C00266873 | Unique | null |

**Customer:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| CUSTOMER\_ID | VARCHAR2(10 BYTE) | No | null | 1 |
| STUDENT\_ID | NUMBER(7,0) | No | null | 2 |
| FIRST\_NAME | VARCHAR2(20 BYTE) | No | null | 3 |
| LAST\_NAME | VARCHAR2(20 BYTE) | No | null | 4 |
| ADDRESS\_ID | NUMBER | No | null | 5 |
| PHONE\_NUMBER | VARCHAR2(20 BYTE) | No | null | 6 |
| DATE\_OF\_BIRTH | DATE | No | null | 7 |
| INSTITUTION\_NAME | VARCHAR2(20 BYTE) | No | null | 8 |
| RELATIONSHIP\_MANAGER\_ID | NUMBER | No | null | 9 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| CUSTOMER\_ADDRESS\_ID\_FK | Foreign\_Key | null |
| CUSTOMER\_ID\_PK | Primary\_Key | null |
| EMPLOYEE\_ID\_FK | Foreign\_Key | null |
| SYS\_C00266842 | Check | "STUDENT\_ID" IS NOT NULL |
| SYS\_C00266843 | Check | "FIRST\_NAME" IS NOT NULL |
| SYS\_C00266844 | Check | "LAST\_NAME" IS NOT NULL |
| SYS\_C00266845 | Check | "ADDRESS\_ID" IS NOT NULL |
| SYS\_C00266846 | Check | "PHONE\_NUMBER" IS NOT NULL |
| SYS\_C00266847 | Check | "DATE\_OF\_BIRTH" IS NOT NULL |
| SYS\_C00266848 | Check | "INSTITUTION\_NAME" IS NOT NULL |
| SYS\_C00266849 | Check | "RELATIONSHIP\_MANAGER\_ID" IS NOT NULL |
| SYS\_C00266850 | Check | student\_id > 0 |
| UNIQUE\_CUSTOMER | Unique | null |

**Customer\_address**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| ADDRESS\_ID | NUMBER | No | "OPS$2123147"."ISEQ$$\_413851".nextval | 1 |
| BUILDING\_HOUSE\_DETAILS | VARCHAR2(100 BYTE) | No | null | 2 |
| STREET | VARCHAR2(100 BYTE) | No | null | 3 |
| CITY | VARCHAR2(30 BYTE) | No | null | 4 |
| STATE\_OR\_REGION | VARCHAR2(100 BYTE) | No | null | 5 |
| POSTAL\_CODE | CHAR(8 BYTE) | No | null | 6 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| ADDRESS\_ID\_PK | Primary\_Key | null |
| SYS\_C00266835 | Check | "ADDRESS\_ID" IS NOT NULL |
| SYS\_C00266836 | Check | "BUILDING\_HOUSE\_DETAILS" IS NOT NULL |
| SYS\_C00266837 | Check | "STREET" IS NOT NULL |
| SYS\_C00266838 | Check | "CITY" IS NOT NULL |
| SYS\_C00266839 | Check | "STATE\_OR\_REGION" IS NOT NULL |
| SYS\_C00266840 | Check | "POSTAL\_CODE" IS NOT NULL |

**Customer\_transaction:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| TRANSACTION\_ID | NUMBER | No | "OPS$2123147"."ISEQ$$\_413872".nextval | 1 |
| TRANSACTION\_DATE | TIMESTAMP(6) | Yes | CURRENT\_TIMESTAMP | 2 |
| AMOUNT | NUMBER | No | null | 3 |
| TRANSACTION\_DESCRIPTION | VARCHAR2(100 BYTE) | No | null | 4 |
| TRANSACTION\_COMPLETION | CHAR(1 BYTE) | No | null | 5 |
| TRANSACTION\_TYPE\_ID | CHAR(1 BYTE) | Yes | null | 6 |
| ACCOUNT\_NUMBER | NUMBER | No | null | 7 |
| PROCESSING\_DATE | TIMESTAMP(6) | Yes | CURRENT\_TIMESTAMP | 8 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| CHECK\_TRANSACTION\_DATE\_END | Check | transaction\_date <= processing\_date |
| CHECK\_TRANSACTION\_DATE\_START | Check | transaction\_date >= '01-JAN-2011' |
| CHECK\_TRANSACTION\_ID | Check | transaction\_id > 0 |
| CHECK\_TRANSACTION\_STATUS | Check | transaction\_completion = 'N' OR transaction\_completion = 'Y' |
| SYS\_C00266885 | Check | "TRANSACTION\_ID" IS NOT NULL |
| SYS\_C00266886 | Check | "AMOUNT" IS NOT NULL |
| SYS\_C00266887 | Check | "TRANSACTION\_DESCRIPTION" IS NOT NULL |
| SYS\_C00266888 | Check | "TRANSACTION\_COMPLETION" IS NOT NULL |
| SYS\_C00266889 | Check | "ACCOUNT\_NUMBER" IS NOT NULL |
| TRANSACTION\_ID\_PK | Primary\_Key | null |
| TRANSACTION\_TYPE\_ID\_FK | Foreign\_Key | null |
| TR\_ACCOUNT\_NUMBER\_FK | Foreign\_Key | null |

**Transactions\_type:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| TRANSACTION\_TYPE\_ID | CHAR(1 BYTE) | No | null | 1 |
| TYPE\_DESCRIPTION | VARCHAR2(20 BYTE) | No | null | 2 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| CHECK\_TRANSACTION\_TYPE | Check | transaction\_type\_id BETWEEN 1 AND 6 |
| SYS\_C00266881 | Check | "TYPE\_DESCRIPTION" IS NOT NULL |
| SYS\_C00266884 | Unique | null |
| TRANSACTION\_TYPE\_ID\_PK | Primary\_Key | null |

**Employee:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| EMPLOYEE\_ID | NUMBER | No | "OPS$2123147"."ISEQ$$\_413847".nextval | 1 |
| FIRST\_NAME | VARCHAR2(20 BYTE) | No | null | 2 |
| LAST\_NAME | VARCHAR2(20 BYTE) | No | null | 3 |
| PHONE\_NUMBER | VARCHAR2(20 BYTE) | No | null | 4 |
| ADDRESS | VARCHAR2(100 BYTE) | No | null | 5 |
| START\_DATE | DATE | No | null | 6 |
| DATE\_OF\_BIRTH | DATE | No | null | 7 |
| BRANCH\_ID | CHAR(3 BYTE) | No | null | 8 |
| MANAGER\_ID | NUMBER | Yes | null | 9 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| EMPLOYEE\_BRANCH\_ID\_FK | Foreign\_Key | null |
| EMPLOYEE\_ID\_PK | Primary\_Key | null |
| EMPLOYEE\_MANAGER\_FK | Foreign\_Key | null |
| SYS\_C00266823 | Check | "EMPLOYEE\_ID" IS NOT NULL |
| SYS\_C00266824 | Check | "FIRST\_NAME" IS NOT NULL |
| SYS\_C00266825 | Check | "LAST\_NAME" IS NOT NULL |
| SYS\_C00266826 | Check | "PHONE\_NUMBER" IS NOT NULL |
| SYS\_C00266827 | Check | "ADDRESS" IS NOT NULL |
| SYS\_C00266828 | Check | "START\_DATE" IS NOT NULL |
| SYS\_C00266829 | Check | "DATE\_OF\_BIRTH" IS NOT NULL |
| SYS\_C00266830 | Check | "BRANCH\_ID" IS NOT NULL |
| SYS\_C00266832 | Unique | null |

**Loan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| LOAN\_ID | NUMBER | No | "OPS$2123147"."ISEQ$$\_413878".nextval | 1 |
| AMOUNT | NUMBER(38,0) | No | null | 2 |
| LOAN\_TYPE\_ID | CHAR(1 BYTE) | No | null | 3 |
| BRANCH\_ID | CHAR(3 BYTE) | No | null | 4 |
| LOAN\_REQUEST\_DATE | DATE | No | null | 5 |
| PROCESSING\_DATE | DATE | Yes | sysdate | 6 |
| CLOSE\_DATE | DATE | Yes | sysdate+365 | 7 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| ONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| CHECK\_LOAN\_AMOUNT | Check | amount BETWEEN 2000 AND 150000 |
| CHECK\_LOAN\_CLOSE\_DATE | Check | close\_date BETWEEN loan\_request\_date+365 AND loan\_request\_date+365\*10 |
| CHECK\_LOAN\_DATE | Check | loan\_request\_date >= TO\_DATE('2012-01-01','YYYY-MM-DD') AND loan\_request\_date+365 < processing\_date |
| LOAN\_BRANCH\_ID\_FK | Foreign\_Key | null |
| LOAN\_ID\_PK | Primary\_Key | null |
| LOAN\_TYPE\_ID\_FK | Foreign\_Key | null |
| SYS\_C00266901 | Check | "LOAN\_ID" IS NOT NULL |
| SYS\_C00266902 | Check | "AMOUNT" IS NOT NULL |
| SYS\_C00266903 | Check | "LOAN\_TYPE\_ID" IS NOT NULL |
| SYS\_C00266904 | Check | "BRANCH\_ID" IS NOT NULL |
| SYS\_C00266905 | Check | "LOAN\_REQUEST\_DATE" IS NOT NULL |

**Loan\_owner:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| LOAN\_ID | NUMBER | No | null | 1 |
| CUSTOMER\_ID | VARCHAR2(10 BYTE) | No | null | 2 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| ONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| CUSTOMER\_LOAN\_ID\_FK | Foreign\_Key | null |
| LOAN\_CUSTOMER\_IDS\_PK | Primary\_Key | null |
| LOAN\_OWNER\_ID\_FK | Foreign\_Key | null |

**Loan\_payment:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| PAYMENT\_ID | NUMBER | No | "OPS$2123147"."ISEQ$$\_413883".nextval | 1 |
| AMOUNT | NUMBER(38,0) | No | null | 2 |
| PAYMENT\_DATE | TIMESTAMP(6) | Yes | CURRENT\_TIMESTAMP | 3 |
| PAYMENT\_DESCRIPTION | VARCHAR2(50 BYTE) | No | null | 4 |
| LOAN\_ID | NUMBER | No | null | 5 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| CHECK\_PAYMENT\_DATE | Check | payment\_date >= TO\_DATE('2012-01-01','YYYY-MM-DD') AND amount > 20 |
| LOAN\_ID\_FK | Foreign\_Key | null |
| PAYMENT\_ID\_PK | Primary\_Key | null |
| SYS\_C00266915 | Check | "PAYMENT\_ID" IS NOT NULL |
| SYS\_C00266916 | Check | "AMOUNT" IS NOT NULL |
| SYS\_C00266917 | Check | "PAYMENT\_DESCRIPTION" IS NOT NULL |
| SYS\_C00266918 | Check | "LOAN\_ID" IS NOT NULL |

**Loan\_type:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLUMN\_NAME | DATA\_TYPE | NULLABLE | DATA\_DEFAULT | COLUMN\_ID |
| LOAN\_TYPE\_ID | CHAR(1 BYTE) | No | null | 1 |
| LOAN\_DESCRIPTION | VARCHAR2(50 BYTE) | No | null | 2 |

**Constraints:**

|  |  |  |
| --- | --- | --- |
| ONSTRAINT\_NAME | CONSTRAINT\_TYPE | SEARCH\_CONDITION |
| CHECK\_LOAN\_TYPE | Check | loan\_type\_id BETWEEN 1 AND 5 |
| LOAN\_TYPE\_ID\_PK | Primary\_Key | null |
| SYS\_C00266897 | Check | "LOAN\_DESCRIPTION" IS NOT NULL |
| SYS\_C00266900 | Unique | null |

# Conclusion:

In this report, several aspects of the database management system were covered which help the bank to transform to digitalisation. Such plans that meet today’s technology will also attract more students to the bank and provide a competitive advantage over other banks.

During the database system implementation stage, several challenges were encountered, such as replicating my environment (the full SQL statements) on oracle live SQL. The issue was related to a check constraint on a timestamp data type I had created on “Customer\_transaction” table.

According to the below error, I was violating a constraint I had created while the same code could run successfully on Oracle SQL developer:

ORA-02290: check constraint (SQL\_OEKUHGHUOKQYVTGNEYBYNZITL.CHECK\_TRANSACTION\_DATE\_START) violated ORA-06512: at "SYS.DBMS\_SQL", line 1721

The solution to the above issue was to also specify the time as well in the check constraint.

As part of future work, digital transformation can take a step further within the Bank’s database management system. Customers and employees can get access to their assets and financial status with the bank directly through their phone or any front-end applications (HTTP/S client). With the help of Oracle REST Data Services (ORDS) which is used to translate REST requests to Oracle database and transform the database response to JSON then send it back to the user.

# References:

1. Danny Palmer (2013) *ICO fines Bank of Scotland for ‘unforgivable’ breach of Data Protection Act*. Available at: https://www.computing.co.uk/news/2287087/ico-fines-bank-of-scotland-for-unforgivable-breach-of-data-protection-act (Accessed: 29 November 2022).
2. Getta, J.R. (2022) *CSIT115 Data Management and Security Legal and Ethical Issues in Data Management Legal and Ethical Issues in Data Management*.
3. Zahid, H. (2021) *Data Management Basics 2: Ethical and legal issues in data sharing*.

# Appendix 1

IMPORTANT NOTE: The below SQL statements were built and run successfully on Oracle SQL Developer, Version 22.2.1.234 while connected to the Wolverhampton database server.

|  |
| --- |
| --drop an existing table in the database.  BEGIN  EXECUTE IMMEDIATE 'DROP TABLE Customer\_transaction';  EXECUTE IMMEDIATE 'DROP TABLE Card';  EXECUTE IMMEDIATE 'DROP TABLE Card\_type';  EXECUTE IMMEDIATE 'DROP TABLE Bank\_account';  EXECUTE IMMEDIATE 'DROP TABLE Loan\_payment';  EXECUTE IMMEDIATE 'DROP TABLE Loan\_owner';  EXECUTE IMMEDIATE 'DROP TABLE Loan';  EXECUTE IMMEDIATE 'DROP TABLE Transaction\_type';  EXECUTE IMMEDIATE 'DROP TABLE Account\_type';  EXECUTE IMMEDIATE 'DROP TABLE Loan\_type';  EXECUTE IMMEDIATE 'DROP TABLE Customer';  EXECUTE IMMEDIATE 'DROP TABLE Customer\_address';  EXECUTE IMMEDIATE 'DROP TABLE Employee';  EXECUTE IMMEDIATE 'DROP TABLE Branch';  EXECUTE IMMEDIATE 'DROP TABLE Bank';  EXCEPTION  WHEN OTHERS THEN  IF SQLCODE != -942 THEN  RAISE;  END IF;  END;  /  CREATE TABLE Bank (  bank\_name VARCHAR(30) NOT NULL,  license CHAR(5) NOT NULL,  address VARCHAR(100) NOT NULL,  CONSTRAINT bank\_pk PRIMARY KEY(bank\_name)  );  -- Bank Data  INSERT INTO Bank VALUES ('C of B', 'BL865', '45, Bakery Street, London, UK');  CREATE TABLE Branch (  branch\_id char(3) NOT NULL,  branch\_name VARCHAR(30) NOT NULL,  telephone VARCHAR(20) NOT NULL,  email VARCHAR(50) NOT NULL,  address VARCHAR(100) NOT NULL,  bank\_name VARCHAR(30) NOT NULL,  CONSTRAINT branch\_pk PRIMARY KEY(branch\_id),  CONSTRAINT bank\_name\_fk FOREIGN KEY(bank\_name) REFERENCES Bank(bank\_name),  CONSTRAINT unique\_branch UNIQUE (branch\_name, telephone, email, address)  );  -- Bank Branches Data  INSERT INTO Branch (branch\_id, branch\_name, telephone, email, address, bank\_name)  WITH br AS (  SELECT '502', 'Head Office', '+4429101000', 'h.o@cofb.bank', '45, Bakery Street, London, UK', 'C of B' FROM dual UNION ALL  SELECT '018', 'chelsea anteater', '+86 (898) 758-2189', 'wholdall0@cofb.bank', '544 Twin Pines Court, Liverpool, UK', 'C of B' FROM dual UNION ALL  SELECT '074', 'Westham eagle', '+258 (604) 650-6700', 'rwynett2@cofb.bank', '66 Lunder Park, Manchister, UK', 'C of B' FROM dual UNION ALL  SELECT '096', 'Manchister crocodile', '+33 (593) 323-5644', 'pagney3@cofb.bank', '84 Comanche Crossing, Wolverhapton, UK', 'C of B' FROM dual UNION ALL  SELECT '052', 'Bermingham lizard', '+53 (276) 814-0053', 'arumsby4@cofb.bank', '51664 Blaine Road, Bermingham, UK', 'C of B' FROM dual UNION ALL  SELECT '802', 'Wolverhampton', '+1 (826) 180-3016', 'gsmartman5@cofb.bank', '8 Autumn Leaf Lane, Westham, UK', 'C of B' FROM dual UNION ALL  SELECT '041', 'Leeds, ring-tailed', '+86 (155) 974-4047', 'dcullinane6@cofb.bank', '94 Summit Park, Fullham, UK', 'C of B' FROM dual UNION ALL  SELECT '030', 'Blackburn rhinoceros', '+380 (353) 967-4134', 'gdewhirst9@cofb.bank', '6 Lien Center, Belfast, UK', 'C of B' FROM dual  )  SELECT \* FROM br;  CREATE TABLE Employee (  employee\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  first\_name VARCHAR(20) NOT NULL,  last\_name VARCHAR(20) NOT NULL,  phone\_number VARCHAR(20) NOT NULL UNIQUE,  address VARCHAR(100) NOT NULL,  start\_date DATE NOT NULL,  date\_of\_birth DATE NOT NULL,  branch\_id CHAR(3) NOT NULL,  manager\_id NUMBER,  CONSTRAINT employee\_id\_pk PRIMARY KEY(employee\_id),  CONSTRAINT employee\_branch\_id\_fk FOREIGN KEY(branch\_id) REFERENCES Branch(branch\_id),  CONSTRAINT employee\_manager\_fk FOREIGN KEY(manager\_id) REFERENCES Employee(employee\_id)  );  -- Restriction: Employment date starts from 01-01-2010  -- Restriction: Employee's age must be 18 or over  CREATE OR REPLACE TRIGGER trg\_dob\_employee  BEFORE INSERT OR UPDATE  on Employee  FOR EACH ROW  DECLARE  emp\_age NUMBER;  BEGIN  -- Finding employee age by date of birth  SELECT MONTHS\_BETWEEN(TO\_DATE(sysdate,'DD-MON-YYYY'), TO\_DATE(:new.date\_of\_birth,'DD-MON-YYYY'))/12  INTO EMP\_AGE FROM dual;  -- Check employee age  IF (EMP\_AGE < 18) THEN  RAISE\_APPLICATION\_ERROR(-20000, EMP\_AGE);  END IF;  -- Allow only dates from when the bank was established  IF(:new.start\_date < '01-JAN-2010') THEN  RAISE\_APPLICATION\_ERROR(-20000,'Employment date must start from 01-01-2010.');  END IF;  IF(:new.start\_date > sysdate) THEN  RAISE\_APPLICATION\_ERROR(-20000,'Employee cannot be registred on a futurte date.');  END IF;  END;  /  --nls\_date\_format is used to specify the default date format to use with the TO\_CHAR at a later stage  ALTER SESSION SET nls\_date\_format = 'DD-MON-YYYY';  -- Employee Data  INSERT INTO Employee (first\_name, last\_name, phone\_number, address, start\_date, date\_of\_birth, branch\_id, manager\_id)  WITH emp AS(  SELECT 'Lionel', 'Messi', '+86 (707) 560-9523', '45 Southlands, Blaina,NP13 3JN, UK', '12-Mar-2014', '02-May-1977', '502', NULL FROM dual UNION ALL  SELECT 'Ethelyn', 'Kauffman', '+351 (142) 478-4871', '12 Shaw Court, Malmesbury Road, Morden,SM4 6HH, UK', '25-Jan-2012', '20-Sep-1972', '502', 1 FROM dual UNION ALL  SELECT 'Jessie', 'Lintott', '+55 (442) 339-5212', '2 Brook Cottages, Great Wolford,CV36 5NP, UK', '16-Nov-2013', '10-Jun-1998', '502', 2 FROM dual UNION ALL  SELECT 'Delinda', 'Geerling', '+86 (707) 560-9529', '45 Southlands, Blaina,NP13 3JN, UK', '12-Mar-2014', '02-May-1977', '502', 3 FROM dual UNION ALL  SELECT 'Jessie', 'Lintott', '+351 (142) 478-4875', '12 Shaw Court, Malmesbury Road, Morden,SM4 6HH, UK', '25-Jan-2012', '20-Sep-1972', '018', 2 FROM dual UNION ALL  SELECT 'Jaye', 'Abrams', '+269 (910) 803-4376', '11 Squires Mount, London,NW3 1ED, UK', '15-Dec-2020', '06-Jun-1980', '018', 5 FROM dual UNION ALL  SELECT 'Garrick', 'Parkes', '+62 (716) 795-4927', '88 Exley Road, Keighley,BD21 1LT, UK', '27-Nov-2014', '29-Sep-1982', '030', 2 FROM dual UNION ALL  SELECT 'Jaye', 'Abrams', '+81 (495) 229-2559', '36 The Kings Gap, Hoylake,CH47 1HF, UK', '24-Jun-2022', '03-Aug-1982', '030', 7 FROM dual UNION ALL  SELECT 'Konstantine', 'Anselm', '+389 (884) 109-0248', '67469 Packers Avenue', '10-Mar-2010', '30-May-1972', '052', 2 FROM dual UNION ALL  SELECT 'Kain', 'Cuppleditch', '+1 (802) 831-5499', 'Fairways, Carlisle Road, Longtown,CA6 5SQ, UK', '27-Sep-2020', '05-Mar-1987', '052', 9 FROM dual UNION ALL  SELECT 'Cale', 'Larkworthy', '+420 (927) 525-4003', '165 - 167 Garstang Road, Fulwood,PR2 3BH, UK', '17-Jun-2013', '22-Oct-1982', '074', 2 FROM dual UNION ALL  SELECT 'Steven', 'Whatmough', '+1 (823) 836-0450', 'Southview, Bowlers Green, Magdalen Laver,CM5 0ET, UK', '04-Nov-2015', '30-Nov-1974', '074', 11 FROM dual UNION ALL  SELECT 'Sharona', 'Cordell', '+358 (694) 526-2457', '6 Lime Tree Place, St. Albans,AL1 3BD, UK', '09-Aug-2017', '12-Jun-1978', '041', 2 FROM dual UNION ALL  SELECT 'Steven', 'Whatmough', '+33 (701) 233-4336', '8 Mill Hill, Cleator Moor,CA25 5SH, UK', '05-Jan-2017', '08-Dec-1997', '041', 13 FROM dual UNION ALL  SELECT 'Juline', 'Quiddinton', '+351 (972) 278-4623', '10 Rose Garden Lane, Wynyard,TS22 5WB, UK', '27-Aug-2019', '03-Sep-1988', '096', 2 FROM dual UNION ALL  SELECT 'Sula', 'Gibbie', '+86 (635) 525-8442', 'Flat 6, Coram House, Wood Street, London,W4 2JW, UK', '31-May-2019', '16-Nov-1974', '096', 15 FROM dual UNION ALL  SELECT 'Noah', 'Odger', '+55 (775) 681-9220', 'Flat 3, The Coach House, Red Wharf Bay,LL75 8RJ, UK', '13-Jan-2022', '02-Feb-1974', '802', 2 FROM dual UNION ALL  SELECT 'Tulley', 'Coie', '+63 (328) 430-4413', '12 Fitzpiers, Saffron Walden,CB10 2BD, UK', '22-Aug-2019', '12-Jul-1988', '802', 17 FROM dual  )  SELECT \* FROM emp;  CREATE TABLE Customer\_address(  address\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  building\_house\_details VARCHAR(100) NOT NULL,  street VARCHAR(100) NOT NULL,  city VARCHAR(30) NOT NULL,  state\_or\_region VARCHAR(100) NOT NULL,  postal\_code CHAR(8) NOT NULL,  CONSTRAINT address\_id\_pk PRIMARY KEY(address\_id)  );  -- customer address data  INSERT INTO Customer\_address (building\_house\_details, street, city, state\_or\_region, postal\_code)  WITH cust\_addr AS(  SELECT '18 Acrantophis', 'Henley Rd', 'Reading', 'West Berkshire', 'OA8 1IJ' FROM dual UNION ALL  SELECT '152 Neophron percnopterus ', 'High St', 'Kenilworth', 'Warwickshire', 'WN3 8SL' FROM dual UNION ALL  SELECT '11 Phalacrocorax niger', 'Radcliffe Rd', 'West Bridgford', 'Nottinghamshire', 'WA8 6DJ' FROM dual UNION ALL  SELECT '38 Phacochoerus aethiopus', 'Baxtergate', 'Harlow', 'North Yorkshire', 'YO21 1BN' FROM dual UNION ALL  SELECT '68 Phascogale calura', 'Broadley Rd', 'Harlow', 'Essex', 'CM19 5RD' FROM dual UNION ALL  SELECT '501 Cercatetus concinnus', 'Jonathans', 'Milton Keynes', 'Buckinghamshire County', 'MK6 5DF' FROM dual UNION ALL  SELECT '311 Bradypus tridactylus', 'Stoke Rd', 'Guildford', 'Surrey', 'GU1 1EZ' FROM dual UNION ALL  SELECT '141 Eolophus roseicapillus', 'Cheltenham Rd', 'Guildford', 'Avon', 'BS6 5RW' FROM dual UNION ALL  SELECT '28 Connochaetus taurinus', '30 Easby Rd', 'Bradford', 'West Yorkshire', 'BD7 1QX' FROM dual UNION ALL  SELECT '50 Manouria emys', 'Christchurch Rd', 'Milton Keynes', 'Bucking', 'SO23 9SU' FROM dual  )  SELECT \* FROM cust\_addr;  CREATE TABLE Customer(  customer\_id VARCHAR(10),  student\_id NUMBER(7) NOT NULL CHECK (student\_id > 0),  first\_name VARCHAR(20) NOT NULL,  last\_name VARCHAR(20) NOT NULL,  address\_id NUMBER NOT NULL,  phone\_number VARCHAR(20) NOT NULL,  date\_of\_birth DATE NOT NULL,  institution\_name VARCHAR(20) NOT NULL,  relationship\_manager\_id NUMBER NOT NULL,  CONSTRAINT customer\_id\_pk PRIMARY KEY(customer\_id),  CONSTRAINT employee\_id\_fk FOREIGN KEY(relationship\_manager\_id) REFERENCES Employee(employee\_id),  CONSTRAINT unique\_customer UNIQUE (student\_id, phone\_number),  CONSTRAINT customer\_address\_id\_fk FOREIGN KEY(address\_id) REFERENCES Customer\_address(address\_id)  );  -- Restriction: Customer age must be between 16 and 100  CREATE OR REPLACE TRIGGER trg\_dob\_customer  BEFORE INSERT OR UPDATE  ON Customer  FOR EACH ROW  DECLARE  cust\_age NUMBER;  BEGIN  -- Finding customer age by date of birth  SELECT MONTHS\_BETWEEN(TO\_DATE(sysdate,'DD-MON-YYYY'), TO\_DATE(:new.date\_of\_birth,'DD-MON-YYYY'))/12  INTO cust\_age FROM dual;  -- Check customer age  IF (cust\_age NOT BETWEEN 16 AND 100) THEN  RAISE\_APPLICATION\_ERROR(-20000,'Customer age must be greater than or equal to 16.');  END IF;  END;  /  -- Customer Data  INSERT INTO Customer (customer\_id, student\_id, first\_name, last\_name, address\_id, phone\_number, date\_of\_birth, institution\_name, relationship\_manager\_id)  WITH cust AS (  SELECT '42943M', 21795, 'Hermione', 'Harwin', 1, '+62 (964) 166-7364', '10-Mar-1983', 'Devcast', 4 FROM dual UNION ALL  SELECT '12001M', 21587, 'Carolina', 'Marin', 2, '+63 (967) 105-0384', '10-Feb-1999', 'Oyonder', 5 FROM dual UNION ALL  SELECT '42195M', 70849, 'Merrel', 'Gillian', 3, '+1 (209) 220-4172', '29-Apr-1994', 'Skibox', 7 FROM dual UNION ALL  SELECT '21802D', 46906, 'Ruthanne', 'Nimmo', 4, '+62 (454) 265-7679', '13-Sep-1972', 'Tazzy', 8 FROM dual UNION ALL  SELECT '65076N', 58179, 'Delano', 'Stormes', 5, '+63 (532) 351-6819', '12-Aug-1982', 'Wikibox', 9 FROM dual UNION ALL  SELECT '76945M', 41363, 'Edwin', 'Cuthbert', 6, '+84 (350) 292-4089', '26-Nov-1997', 'Brainlounge', 10 FROM dual UNION ALL  SELECT '41142G', 47107, 'Keelby', 'Dunkirk', 7, '+54 (876) 720-5474', '27-Aug-1979', 'Devpoint', 11 FROM dual UNION ALL  SELECT '47966M', 20966, 'Oran', 'Floyed', 8, '+1 (314) 579-1317', '07-Jan-1975', 'Devpoint', 12 FROM dual UNION ALL  SELECT '74020M', 99256, 'Aleksandr', 'Martell', 9, '+86 (110) 613-7576', '07-Feb-1993', 'Kayveo', 13 FROM dual UNION ALL  SELECT '28735M', 16703, 'Thurstan', 'Schubbert', 10, '+63 (245) 355-0857', '14-Aug-1990', 'Browsebug', 10 FROM dual  )  SELECT \* FROM cust;  CREATE TABLE Account\_type(  account\_type\_id CHAR(1),  account\_description VARCHAR(30) NOT NULL UNIQUE,  CONSTRAINT account\_type\_id\_pk PRIMARY KEY(account\_type\_id),  CONSTRAINT check\_account\_type CHECK (account\_type\_id BETWEEN 1 AND 2)  );  -- Account type data  INSERT INTO Account\_type (account\_type\_id, account\_description) VALUES (1, 'Current Account');  INSERT INTO Account\_type (account\_type\_id, account\_description) VALUES (2, 'Saving Account');    /\*\* Account number is generated automatically starting from 10001.  This incrementation affect null values as well \*\*/  CREATE TABLE Bank\_account(  account\_number NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY START WITH 10001,  balance NUMBER NOT NULL,  opened\_date DATE NOT NULL,  branch\_id CHAR(3) NOT NULL,  customer\_id VARCHAR(10) NOT NULL UNIQUE,  account\_type\_id CHAR(1) NOT NULL,  CONSTRAINT account\_number\_pk PRIMARY KEY(account\_number),  CONSTRAINT branch\_id\_fk FOREIGN KEY(branch\_id) REFERENCES Branch(branch\_id),  CONSTRAINT ac\_customer\_id\_fk FOREIGN KEY(customer\_id) REFERENCES Customer(customer\_id),  CONSTRAINT account\_type\_id\_fk FOREIGN KEY(account\_type\_id) REFERENCES Account\_type(account\_type\_id)  );  /\*\* Restriction:  Account opened date starts from 01-01-2011  Account cannot be opened in future date \*\*/  CREATE OR REPLACE TRIGGER trg\_accounts\_date  BEFORE INSERT OR UPDATE  on Bank\_account  FOR EACH ROW  BEGIN  -- check account account open date  IF(:new.opened\_date < '01-JAN-2011') THEN  RAISE\_APPLICATION\_ERROR(-20000,'account opening date must start from 01-01-2011.');  END IF;  -- account date must not be on a future date  IF(:new.opened\_date > sysdate) THEN  RAISE\_APPLICATION\_ERROR(-20000,'Account cannot be opened on future date.');  END IF;  END;  /  -- Bank account data  INSERT INTO Bank\_account (balance, opened\_date, branch\_id, customer\_id, account\_type\_id)  WITH bank\_ac AS(  SELECT 8440.13, '17-Nov-2016', '018', '12001M', '1' FROM dual UNION ALL  SELECT 4035.14, '22-Sep-2019', '030', '42195M', '1' FROM dual UNION ALL  SELECT 4744.44, '05-Jun-2021', '052', '21802D', '1' FROM dual UNION ALL  SELECT 2821.74, '14-Jun-2021', '052', '65076N', '1' FROM dual UNION ALL  SELECT 6692.65, '28-Sep-2018', '074', '76945M', '1' FROM dual UNION ALL  SELECT 1388.89, '09-Feb-2012', '074', '41142G', '1' FROM dual UNION ALL  SELECT 650.86, '13-Jun-2018', '041', '47966M', '1' FROM dual UNION ALL  SELECT 6030.72, '11-Oct-2021', '041', '74020M', '1' FROM dual UNION ALL  SELECT 9304.48, '05-Feb-2022', '018', '42943M', '2' FROM dual UNION ALL  SELECT 8454.91, '30-Nov-2021', '074', '28735M', '2' FROM dual  )  SELECT \* FROM bank\_ac;  CREATE TABLE Card\_type(  card\_type\_id CHAR(1),  card\_description VARCHAR(30) NOT NULL UNIQUE,  CONSTRAINT card\_type\_id\_pf PRIMARY KEY(card\_type\_id),  CONSTRAINT check\_card\_type CHECK (card\_type\_id BETWEEN 1 AND 2)  );  -- Card type data  INSERT INTO card\_type (card\_type\_id, card\_description) VALUES ('1', 'Debit Card');  INSERT INTO card\_type (card\_type\_id, card\_description) VALUES ('2', 'Credit Card');  CREATE TABLE Card(  card\_no CHAR(16),  opened\_date DATE DEFAULT sysdate,  expiry\_date DATE NOT NULL,  card\_type\_id CHAR(1) NOT NULL,  account\_number NUMBER NOT NULL,  CONSTRAINT card\_no\_pk PRIMARY KEY(card\_no),  CONSTRAINT card\_type\_id\_fk FOREIGN KEY(card\_type\_id) REFERENCES Card\_type(card\_type\_id),  CONSTRAINT check\_debit\_credit\_cards  CHECK ((card\_no LIKE '79658100%' AND card\_type\_id = '1') OR (card\_no LIKE '79659100%' AND card\_type\_id = '2')),  CONSTRAINT card\_account\_number\_fk FOREIGN KEY(account\_number) REFERENCES Bank\_account(account\_number)  );  /\*\* Restriction:  Card must be valid for 5 years  Card application must start from 01-01-2011 \*\*/  CREATE OR REPLACE TRIGGER trg\_exp\_card  BEFORE INSERT OR UPDATE  on Card  FOR EACH ROW  DECLARE  exp\_card NUMBER;  BEGIN  -- Finding number of years by expiry date  SELECT MONTHS\_BETWEEN(TO\_DATE(:NEW.expiry\_date,'DD-MON-YYYY'), TO\_DATE(:NEW.opened\_date,'DD-MON-YYYY'))/12  INTO exp\_card FROM dual;  -- Check if card is valid for 5 years  IF (exp\_card != 5) THEN  RAISE\_APPLICATION\_ERROR(-20000,exp\_card);  END IF;  -- Allow cards to be issued from 01/01/2011  IF(:new.expiry\_date < '01-JAN-2011') THEN  RAISE\_APPLICATION\_ERROR(-20001,'Card application must start from 01-01-2011.');  END IF;  END;  /  -- Card data  INSERT INTO Card (card\_no, opened\_date, expiry\_date, card\_type\_id, account\_number)  WITH crd AS(  SELECT '7965810014246618','01-JAN-2011','01-JAN-2016','1', 10001 FROM dual UNION ALL  SELECT '7965810014246686','01-MAY-2012','01-MAY-2017','1', 10002 FROM dual UNION ALL  SELECT '7965810014246378','21-SEP-2013','21-SEP-2018','1', 10003 FROM dual UNION ALL  SELECT '7965810014246619','14-JAN-2016','14-JAN-2021','1', 10004 FROM dual UNION ALL  SELECT '7965910014246770','01-APR-2017','01-APR-2022','2', 10004 FROM dual UNION ALL  SELECT '7965910014246771','01-JAN-2019','01-JAN-2024','2', 10005 FROM dual UNION ALL  SELECT '7965910014246772','16-MAY-2013','16-MAY-2018','2', 10007 FROM dual UNION ALL  SELECT '7965910014246773','26-OCT-2018','26-OCT-2023','2', 10001 FROM dual UNION ALL  SELECT '7965910014246774','11-NOV-2018','11-NOV-2023','2', 10009 FROM dual UNION ALL  SELECT '7965910014246775','14-FEB-2014','14-FEB-2019','2', 10010 FROM dual  )  SELECT \* FROM crd;  CREATE TABLE Transaction\_type(  transaction\_type\_id CHAR(1),  type\_description VARCHAR(20) NOT NULL UNIQUE,  CONSTRAINT transaction\_type\_id\_pk PRIMARY KEY(transaction\_type\_id),  CONSTRAINT check\_transaction\_type CHECK (transaction\_type\_id BETWEEN 1 AND 6)  );  --Transaction type data  INSERT INTO Transaction\_type (transaction\_type\_id, type\_description)  WITH tr\_type AS(  SELECT 1, 'Withdrawal' FROM dual UNION ALL  SELECT 2, 'Deposit' FROM dual UNION ALL  SELECT 3, 'Bill Payments' FROM dual UNION ALL  SELECT 4, 'SEPA Transfer' FROM dual UNION ALL  SELECT 5, 'NON-SEPA Transfer' FROM dual UNION ALL  SELECT 6, 'Purchase' FROM dual  )  SELECT \* FROM tr\_type;  CREATE TABLE Customer\_transaction(  transaction\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  transaction\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  amount NUMBER NOT NULL,  transaction\_description VARCHAR(100) NOT NULL,  transaction\_completion CHAR(1) NOT NULL,  transaction\_type\_id CHAR(1),  account\_number NUMBER NOT NULL,  processing\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  CONSTRAINT transaction\_id\_pk PRIMARY KEY(transaction\_id),  CONSTRAINT transaction\_type\_id\_fk FOREIGN KEY(transaction\_type\_id) REFERENCES Transaction\_type(transaction\_type\_id),  CONSTRAINT tr\_account\_number\_fk FOREIGN KEY(account\_number) REFERENCES Bank\_account(account\_number),  CONSTRAINT check\_transaction\_id CHECK (transaction\_id > 0),  CONSTRAINT check\_transaction\_date\_start CHECK (transaction\_date >= '01-JAN-2011 01:00:00'),  CONSTRAINT check\_transaction\_date\_end CHECK (transaction\_date <= processing\_date),  CONSTRAINT check\_transaction\_status CHECK (transaction\_completion = 'N' OR transaction\_completion = 'Y')  );  -- Customer transaction data  INSERT INTO Customer\_transaction (transaction\_date, amount, transaction\_description, transaction\_completion, transaction\_type\_id, account\_number)  WITH cust\_tr AS(  SELECT TO\_TIMESTAMP('26-Jan-2014 3:50'), 300.00, 'Deposit-360ATM', 'Y', '2', 10001 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('05-May-2022 1:46'), -348.94, 'Ophth counsel and instruct', 'Y', '3', 10002 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('01-Jun-2013 8:00'), -411.03, 'Revision of lead', 'Y', '5', 10003 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('20-Jul-2022 3:03'), -170.00, 'Withdrawal-360ATM', 'N', '1', 10004 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('09-Jun-2015 8:22'), -38.95, 'Cholangiogram NEC', 'N', '4', 10005 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('21-Apr-2016 2:32'), -835.45, 'Bronch/lung dx proc NEC', 'Y', '5', 10006 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('22-May-2019 6:23'), -400.00, 'Withdrawal-365ATM', 'N', '1', 10007 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('18-Jul-2021 7:18'), 580.00, 'Deposit-362ATM', 'Y', '2', 10008 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('14-Apr-2017 9:54'), -340.00, 'Withdrawal-365ATM', 'Y', '1', 10009 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('20-Aug-2021 7:18'), 700.00, 'Deposit-360ATM', 'Y', '2', 10008 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('21-Aug-2021 9:00'), -100.00, 'TK Maxx', 'Y', '6', 10008 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('21-Apr-2016 2:32'), 1500.45, 'April Salary-ABC ltd', 'Y', '4', 10006 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('10-Jun-2013 8:00'), 300.03, 'Revision - Refund', 'Y', '4', 10003 FROM dual UNION ALL  SELECT TO\_TIMESTAMP('17-Dec-2014 6:38'), -88.63, 'Cardiotomy', 'N', '3', 10010 FROM dual  )  SELECT \* FROM cust\_tr;  -- Loan type data  CREATE TABLE Loan\_type(  loan\_type\_id CHAR(1),  loan\_description VARCHAR(50) NOT NULL UNIQUE,  CONSTRAINT loan\_type\_id\_pk PRIMARY KEY(loan\_type\_id),  CONSTRAINT check\_loan\_type CHECK (loan\_type\_id BETWEEN 1 AND 5)  );  INSERT INTO Loan\_type (loan\_type\_id, loan\_description)  WITH ln\_tp AS(  SELECT 1, 'Personal Loan' FROM dual UNION ALL  SELECT 2, 'Joint Loan' FROM dual UNION ALL  SELECT 3, 'Home Loan' FROM dual UNION ALL  SELECT 4, 'Vehicle Loan' FROM dual  )  SELECT \* FROM ln\_tp;  --Loan is granted for minimum one year  CREATE TABLE Loan(  loan\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  amount DECIMAL NOT NULL,  loan\_type\_id CHAR(1) NOT NULL,  branch\_id char(3) NOT NULL,  loan\_request\_date DATE NOT NULL,  close\_date DATE DEFAULT sysdate+365,  CONSTRAINT loan\_id\_pk PRIMARY KEY(loan\_id),  CONSTRAINT check\_loan\_date --loan issued 2 years after bank established  CHECK (loan\_request\_date >= TO\_DATE('2012-01-01','YYYY-MM-DD')),  CONSTRAINT check\_loan\_close\_date CHECK (close\_date BETWEEN loan\_request\_date+365 AND loan\_request\_date+365\*10),  CONSTRAINT loan\_type\_id\_fk FOREIGN KEY(loan\_type\_id) REFERENCES Loan\_type(loan\_type\_id),  CONSTRAINT loan\_branch\_id\_fk FOREIGN KEY(branch\_id) REFERENCES Branch(branch\_id),  CONSTRAINT check\_loan\_amount CHECK (amount BETWEEN 2000 AND 150000)  );    -- Loan data  INSERT INTO Loan (amount, branch\_id, loan\_type\_id, loan\_request\_date, close\_date)  WITH ln\_tp AS(  SELECT 4608, '018', '1', '13-Aug-2013', '26-Apr-2015' FROM dual UNION ALL  SELECT 8712, '030', '2', '16-Jan-2015', '01-Oct-2019' FROM dual UNION ALL  SELECT 4340, '030', '3', '12-Feb-2016', '14-Feb-2020' FROM dual UNION ALL  SELECT 11424, '052', '4', '14-Jun-2021', '17-Aug-2023' FROM dual UNION ALL  SELECT 12763, '074', '2', '19-Mar-2020', '30-Sep-2024' FROM dual UNION ALL  SELECT 14988, '074', '4', '23-Jun-2021', '29-Jun-2025' FROM dual UNION ALL  SELECT 3278, '041', '3', '08-May-2015', '14-May-2018' FROM dual UNION ALL  SELECT 2212, '041', '4', '22-Aug-2018', '25-Nov-2020' FROM dual  )  SELECT \* FROM ln\_tp;  CREATE TABLE Loan\_owner (  loan\_id NUMBER,  customer\_id VARCHAR(10),  CONSTRAINT loan\_customer\_ids\_pk PRIMARY KEY(loan\_id, customer\_id),  CONSTRAINT loan\_owner\_id\_fk FOREIGN KEY(loan\_id) REFERENCES Loan(loan\_id),  CONSTRAINT customer\_loan\_id\_fk FOREIGN KEY(customer\_id) REFERENCES Customer(customer\_id)  );  CREATE OR REPLACE TRIGGER trg\_loan\_ownership  BEFORE INSERT OR UPDATE  on Loan\_owner  FOR EACH ROW  DECLARE  v\_loan\_count NUMBER;  v\_customer\_count NUMBER;  BEGIN  SELECT COUNT(1) INTO v\_loan\_count FROM Loan\_owner  WHERE loan\_id = :new.loan\_id;  IF(v\_loan\_count > 1) THEN  RAISE\_APPLICATION\_ERROR(-20000, 'A loan can not be associated to more than two customers');  END IF;    SELECT COUNT(1) INTO v\_customer\_count FROM Loan\_owner  WHERE customer\_id = :new.customer\_id;  IF(v\_customer\_count > 1) THEN  RAISE\_APPLICATION\_ERROR(-20000, 'A customer cannot take more than one loan');  END IF;  END;  /  -- Loan Owner data  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (1, '42943M');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (1, '12001M');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (2, '42195M');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (3, '42195M');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (4, '21802D');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (5, '65076N');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (6, '41142G');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (6, '76945M');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (7, '74020M');  INSERT INTO Loan\_owner (loan\_id, customer\_id) VALUES (8, '47966M');  CREATE TABLE Loan\_payment(  payment\_id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  amount DECIMAL NOT NULL,  payment\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  payment\_description VARCHAR(50) NOT NULL,  loan\_id NUMBER NOT NULL,  CONSTRAINT payment\_id\_pk PRIMARY KEY(payment\_id),  CONSTRAINT loan\_id\_fk FOREIGN KEY(loan\_id) REFERENCES Loan(loan\_id),  /\*\* loan payment issued 2 years after bank established  minimum repayment must be EUR20.00  \*\*/  CONSTRAINT check\_payment\_date CHECK (payment\_date >= TO\_DATE('2012-01-01','YYYY-MM-DD') AND amount > 20)  );  -- Loan payment data  INSERT INTO Loan\_payment (amount, payment\_date, payment\_description, loan\_id)  WITH ln\_tp AS(  SELECT 200, '13-Sep-2013', 'Septemberpayment\_1', 1 FROM dual UNION ALL  SELECT 800, '16-Feb-2015', 'February payment\_2', 1 FROM dual UNION ALL  SELECT 500, '12-Mar-2016', 'March payment\_1', 2 FROM dual UNION ALL  SELECT 360, '14-Jul-2021', 'April payment\_1', 3 FROM dual UNION ALL  SELECT 150, '05-Apr-2014', 'April payment\_1', 4 FROM dual UNION ALL  SELECT 120, '19-Apr-2020', 'April payment\_1', 6 FROM dual UNION ALL  SELECT 140, '23-Jul-2021', 'July payment\_2', 6 FROM dual UNION ALL  SELECT 320, '08-Jun-2015', 'payment\_1', 7 FROM dual UNION ALL  SELECT 220, '22-Sep-2018', ' September payment\_1', 4 FROM dual UNION ALL  SELECT 130, '13-Sep-2020', 'September payment\_2', 2 FROM dual  )  SELECT \* FROM ln\_tp;  COMMIT;  -- Display all tables  SELECT \* FROM Customer\_transaction;  SELECT \* FROM Bank\_account;  SELECT \* FROM Loan\_payment;  SELECT \* FROM Loan;  SELECT \* FROM Loan\_owner;  SELECT \* FROM Transaction\_type;  SELECT \* FROM Account\_type;  SELECT \* FROM Card;  SELECT \* FROM Card\_type;  SELECT \* FROM Loan\_type;  SELECT \* FROM Customer\_address;  SELECT \* FROM Customer;  SELECT \* FROM Employee;  SELECT \* FROM Branch;  SELECT \* FROM Bank;    /\*\* Query to list transactionID, amount and description from customerTrassaction table +  accountNumber and balance from Bank\_account table +  customerID, firstName and lastName from custoemr table  for those transactions which are between -300 to 300  \*\*/  column last\_name format a10  column tr\_descr format a27  SELECT  cust\_tr.transaction\_id AS tr\_id,  cust\_tr.amount AS tr\_amount,  cust\_tr.transaction\_description AS tr\_descr,  ac.account\_number AS ac\_no,  ac.balance AS ac\_balance,  cust.customer\_id AS cust\_id,  cust.first\_name AS first\_name,  cust.last\_name AS last\_name  FROM Customer\_transaction cust\_tr  LEFT JOIN Bank\_account ac  ON cust\_tr.account\_number = ac.account\_number  LEFT JOIN Customer cust  ON ac.customer\_id = cust.customer\_id  WHERE cust\_tr.amount BETWEEN -300 AND 300  ORDER BY amount;  /\*\*  Query to count the number of different cities of the bank's customers  \*\*/  column cities format a6  SELECT  COUNT(DISTINCT city) AS "Number of cities"  FROM customer\_address;  --/\*\*Query to list loanID, amount and request date from loan table +  -- customerID from loan owner table +  -- customer date of birth from from customer table  -- Note: TO\_CHAR function was used on the date columns  --\*\*/  column amount format a8  column loanID format a6  column customer\_DOB format a12  column request\_Date format a13  column customerID format a10  SELECT  lnn.loan\_id AS loanID,  lnn.amount AS amount,  TO\_CHAR(lnn.loan\_request\_date, 'DD/MM/YYYY') AS request\_Date,  ln\_ow.customer\_id AS customerID,  TO\_CHAR(cust.date\_of\_birth, 'DD/MM/YYYY') AS customer\_DOB  FROM loan lnn  LEFT JOIN loan\_owner ln\_ow  ON lnn.loan\_id = ln\_ow.loan\_id  LEFT JOIN customer cust  ON ln\_ow.customer\_id = cust.customer\_id;  /\*\*  Query to list the total of credit transactions on every customer that are  grouped by account number and customer ID and ordered by total of credits in descending order  The list is displaying columns:  amount from Customer\_transaction table +  account\_number from Bank\_account table +  customer\_id from customer table  \*\*/  column customer\_id format a10  SELECT  SUM(ct.amount) AS totalCredits,  ac.account\_number AS accountNO,  c.customer\_id AS customerID  FROM Customer\_transaction ct  LEFT JOIN Bank\_account ac  ON ct.account\_number = ac.account\_number  LEFT JOIN Customer c  ON ac.customer\_id = c.customer\_id  WHERE ct.amount > 0  GROUP BY  ac.account\_number,  c.customer\_id  ORDER BY totalCredits DESC;  /\*\*  Query to count and group transaction types.  Basically, a list to display the number of transactions per transaction type  \*\*/  column counts format a6  SELECT  trtp.type\_description AS transactionTypes,  COUNT(ctr.transaction\_type\_id) AS counts  FROM Transaction\_type trtp  LEFT JOIN Customer\_transaction ctr  ON trtp.transaction\_type\_id = ctr.transaction\_type\_id  GROUP BY  trtp.transaction\_type\_id,  trtp.type\_description;  /\*\*  Query to list branches of those employees whose manager ID is 2.  \*\*/  column branchID format a10  SELECT  br.branch\_id AS branchID,  br.branch\_name AS branchName  FROM Branch br  WHERE branch\_id IN  (SELECT branch\_id FROM Employee WHERE manager\_id = 2); |

# Appendix 2

## Normalisation:

The benefits of using a database that has a suitable normalised set of relations are that the database will be easier for the user to access and maintain the data. Take up minimal storage space on the computer and it minimises data redundancy.

The required data relating to C of B Bank at this stage is in an unnormalized form (UNF). Through the below steps, the data can be normalised:

**UNF:**

Bank(bank\_name, license, address, branch\_id, branch\_name, telephone, email, address, employee\_id, first\_name, last\_name, phone\_number, address, start\_date, date\_of\_birth, address\_id, building\_house\_details, street, city state\_or\_region postal\_code**,** customer\_id, student\_id, first\_name, last\_name, phone\_number, date\_of\_birth, institution\_name, account\_type\_id, account\_description, account\_number, balance, opened\_date, card\_type\_id, card\_description, card\_no, opened\_date, expiry\_date, transaction\_type\_id, type\_description, transaction\_id, transaction\_date, amount, transaction\_description, transaction\_completion, processing\_date, loan\_type\_id, loan\_description, loan\_id, amount, loan\_request\_date, close\_date, payment\_id, amount, payment\_date, payment\_description)

**1st Normal Form:**

Each attribute of the relation must be atomic

* Each table cell should contain a single value.
* Each record needs to be unique.

Bank(bank\_name, license, address, branch\_id, branch\_name, telephone, email, address, employee\_id, first\_name, last\_name, phone\_number, address, start\_date, date\_of\_birth, address\_id, building\_house\_details, street, city state\_or\_region postal\_code**,** customer\_id, student\_id, first\_name, last\_name, phone\_number, date\_of\_birth, institution\_name, account\_type\_id, account\_description, account\_number, balance, opened\_date, card\_type\_id, card\_description, card\_no, opened\_date, expiry\_date, transaction\_type\_id, type\_description, transaction\_id, transaction\_date, amount, transaction\_description, transaction\_completion, processing\_date, loan\_type\_id, loan\_description, loan\_id, amount, loan\_request\_date, close\_date, payment\_id, amount, payment\_date, payment\_description)

**2nd Normal Form:**

* The table satisfies 1NF (first normal form).
* Every non-primary-key attribute is fully functionally dependent on the primary key.

1. Bank(bank\_name, license, address)
2. Branch(branch\_id, branch\_name, telephone, email, address, bank\_name\*)
3. Employee(employee\_id, first\_name, last\_name, phone\_number, address, start\_date, date\_of\_birth, branch\_id\*, manager\_id\*)
4. Customer(customer\_id, student\_id, first\_name, last\_name, phone\_number, date\_of\_birth, institution\_name, address\_id, building\_house\_details, street, city state\_or\_region postal\_code**,** relationship\_manager\_id\*)
5. Account\_type(account\_type\_id, account\_description)
6. Bank\_account(account\_number, balance, opened\_date, branch\_id\*, customer\_id\*, account\_type\_id\*)
7. Card\_type(card\_type\_id, card\_description)
8. Card(card\_no, opened\_date, expiry\_date, account\_number\*, card\_type\_id\*)
9. Transaction\_type(transaction\_type\_id, type\_description)
10. Customer\_transaction(transaction\_id, transaction\_date, amount, transaction\_description, transaction\_completion, processing\_date, account\_number\*, transaction\_type\_id\*)
11. Loan\_type(loan\_type\_id, loan\_description)
12. Loan\_owner(loan\_id, customer\_id)
13. Loan(loan\_id, amount, loan\_request\_date, close\_date, loan\_type\_id\*, branch\_id\*)
14. Loan\_payment(payment\_id, amount, payment\_date, payment\_description, loan\_id\*)

**3rd Normal Form:**

* The table meets the criteria for 2NF.
* All the non-key elements are fully functionally dependent on the primary key
* No non-key element is functionally dependent on any other non-key element

1. Bank(bank\_name, license, address)
2. Branch(branch\_id, branch\_name, telephone, email, address, bank\_name\*)
3. Employee(employee\_id, first\_name, last\_name, phone\_number, address, start\_date, date\_of\_birth, branch\_id\*, manager\_id\*)
4. Customer(customer\_id, student\_id, first\_name, last\_name, phone\_number, date\_of\_birth, institution\_name, relationship\_manager\_id\*)
5. Customer\_address(address\_id, building\_house\_details, street, city state\_or\_region postal\_code**)**
6. Account\_type(account\_type\_id, account\_description)
7. Bank\_account(account\_number, balance, opened\_date, branch\_id\*, customer\_id\*, account\_type\_id\*)
8. Card\_type(card\_type\_id, card\_description)
9. Card(card\_no, opened\_date, expiry\_date, account\_number\*, card\_type\_id\*)
10. Transaction\_type(transaction\_type\_id, type\_description)
11. Customer\_transaction(transaction\_id, transaction\_date, amount, transaction\_description, transaction\_completion, processing\_date, account\_number\*, transaction\_type\_id\*)
12. Loan\_type(loan\_type\_id, loan\_description)
13. Loan\_owner(loan\_id, customer\_id)
14. Loan(loan\_id, amount, loan\_request\_date, close\_date, loan\_type\_id\*, branch\_id\*)
15. Loan\_payment(payment\_id, amount, payment\_date, payment\_description, loan\_id\*)