Databases and Business Applications - CA

# Database Technical Design Document

Version 1.0

11th June 2018

Student Name: Barry Sheppard

Student Number: 10387786

Lecturer: Clive Gargan

Table of Contents

[1. Project Scope 1](#_Toc516566178)

[1.1 Scope of this document 1](#_Toc516566179)

[1.2 Project requirements 1](#_Toc516566180)

[1.3 Design Decisions 1](#_Toc516566181)

[1.4 Technology Used 2](#_Toc516566182)

[2. Technical Design 3](#_Toc516566183)

[2.1 ER Diagram 3](#_Toc516566184)

[2.2 Physical Model in Third Normal Form (3NF) 3](#_Toc516566185)

[2.3 Data Dictionary 4](#_Toc516566186)

[3. Testing 12](#_Toc516566187)

[4. Reflections on Learnings 13](#_Toc516566188)

[Appendix A: References 14](#_Toc516566189)

## Project Scope

### Scope of this document

This document provides a technical overview of the newly developed client database for BankCorp Ltd. This project will supersede the existing Bank Corp Ltd databases and will not have backwards support. This document relates to the design and development of the database only. Data migration will be within the scope of a later project.

### Project requirements

There are 4 key entities:

1. Customers
2. Accounts
3. Securities
4. Associations

A customer can have many accounts and an account can be secured by many securities. A customer can be associated to another customer in the bank. Accounts, Securities, and Associations can be of several types.

The project also requires the following:

* A procedure to create new customer and association.
* A procedure to delete a customer and associated records.
* A report showing the association details between all clients.
* A procedure to delete a reference data entry.
* A report showing what reference data has been deleted.

### Design Decisions

* As per the project requirements, four key entity tables were designed.
* The DataTypes table was created to allow a generic type system. Each type in this table has a category to allow grouping of types. For example, the ‘Corporate’ and ‘Personal’ types would be part of the ‘Customer’ category while the ‘Current’ and ‘Savings’ types would be part of the ‘Accounts’ category.
* Addresses were segregated from the customer table as address details are related to each other, this helps the database adhere to 3NF. This also allows multiple addresses for a single customer.
* The association table was designed to be directional, with the first customer as the client and the second customer as a service provider. For example, “Customer A” has an “Accountant” who is “Customer B”. This can be expressed in reverse to say, “Customer B” is the “Accountant” for “Customer A”. This is shown in both directions in the association view report.
* The customer table was designed to be as simple as possible. The design can be expanded to include additional information if needed in a form similar to the addresses table.

### Technology Used

This database is constructed using the relational database management system Microsoft SQL Server. MS SQL is an industry standard with reliable data integrity and security.

### SQL Code

SQL script to create tables



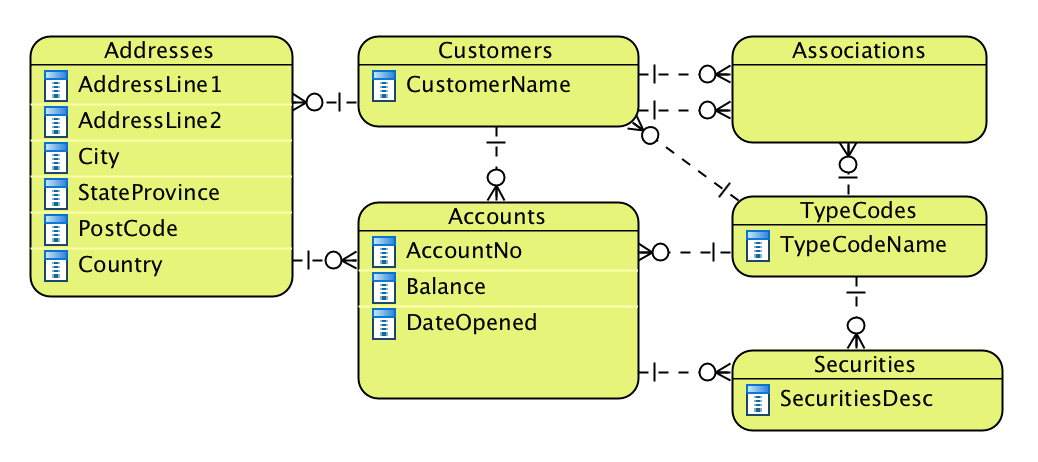
SQL script to create procedures and views



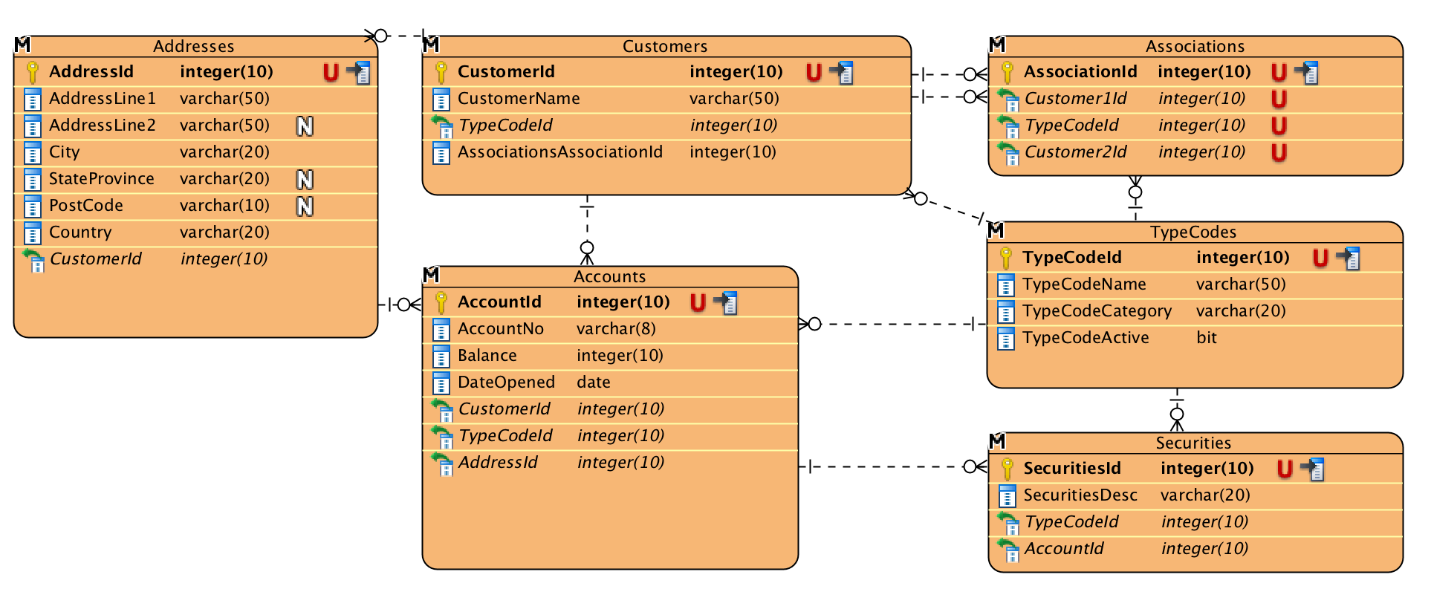
For test data insert scripts, see section 3. Testing

## Technical Design

### ER Diagram



### Physical Model in Third Normal Form (3NF)



### Data Dictionary

**Tables**

**Table: dbo.Customers**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Descriptive Name** | **Valid Values** | **Index Column** | **Allow Nulls** | **Description** |
| CustomerId | int | Customer Id number | Values are system generated number | Y | N | This unique reference is the primary key for this table. |
| CustomerName | Varchar(50) | Customer name | Text characters up to 50. | N | N | This text entry is the full name of the customer |
| TypeCodeId | int | Customer type code | The id reference of an entry from the TypeCode table. This should be of the ‘Customer’ category only. | N | N | This reference will indicate whether the customer is Corporate or Personal customer. Additional types can be added by updating the TypeCodes table. |

**Linked From**

|  |  |  |
| --- | --- | --- |
| **Table** | **Join** | **Description** |
| dbo.Accounts | **dbo.Customers.**CustomerId = **dbo.Accounts.**CustomerId | Each account is owned by a single customer. A customer can have multiple accounts. |
| dbo.Addresses | **dbo.Customers.**CustomerId = **dbo.Addresses.**CustomerId | Each address is owned by a single customer. A customer can have multiple addresses. |
| dbo.Associations | **dbo.Customers.**CustomerId = **dbo.Associations.**Customer1Id | Each association has a client customer reference. |
| dbo.Associations | **dbo.Customers.**CustomerId = **dbo.Associations.**Customer2Id | Each associations has a service provided customer reference. |

**Table: dbo.Accounts**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Descriptive Name** | **Valid Values** | **Index Column** | **Allow Nulls** | **Description** |
| AccountId | int | Account ID number | Values are system generated number | Y | N | This unique reference is the primary key for this table. |
| AccountNo | varchar(8) | Account number | 8 digit account number | N | N | This 8 digit account number is a unique account reference readable for customers or other banks. |
| Balance | money | Balance | Currency amount | N | N | This is the current account balance. |
| OpenDate | datetime | Account opening date | Date | N | N | This is the date the account was opened. |
| CustomerId | int | Customer ID number | ID reference from the Customers table | N | N | This foreign key reference links to the Customers database and shows the customer who owns this account. |
| TypeCodeId | int | Account type code | ID reference from the TypeCode table | N | N | This foreign key reference links to the TypeCode database and shows the type for this account. |
| AddressId | int | Statement address ID | ID reference from the Address table | N | N | This foreign key reference links to the Addresses database and shows the statement address for this account. |
| **Constraint** – This table has a constraint which prevents duplicate account numbers. Each account number must be unique. | | | | | | |

**Linked From**

|  |  |  |
| --- | --- | --- |
| **Table** | **Join** | **Description** |
| dbo.Securities | **dbo. Accounts.**AccountId = **dbo. Securities.**AccountId | Each account is owned by a single customer. A customer can have multiple accounts. |

**Table: dbo.Securities**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Descriptive Name** | **Valid Values** | **Index Column** | **Allow Nulls** | **Description** |
| SecurityId | int | Security ID number | Values are system generated number | Y | N | This unique reference is the primary key for this table. |
| SecuritiesDesc | varchar(50) | Security description | Alphanumeric up to 50 characters. | N | N | This text field describes the security held |
| TypeCodeId | int | Security typed code | ID reference from the TypeCodes table. | N | N | This foreign key reference links to the TypeCodes table and shows what type of Security the entry is. |
| AccountId | int | Connected account ID | ID reference from the Accounts table. | N | N | This foreign key reference links to the Accounts table and shows the account the security is held for. |

**Table: dbo.Associations**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Descriptive Name** | **Valid Values** | **Index Column** | **Allow Nulls** | **Description** |
| AssociationId | int | Association ID number | Values are system generated number | Y | N | This unique reference is the primary key for this table. |
| Customer1Id | int | Client Customer ID | ID reference from the Customers table. | N | N | This foreign key reference links to the Customers table and shows the client in the relationship. |
| TypeCodeId | int | Association Type ID | ID reference from the TypeCodes table. | N | N | This foreign key reference links to the TypeCodes table and shows what the kind of association between customers. |
| Customer2Id | int | Provided Customer ID | ID reference from the Customers table. | N | N | This foreign key reference links to the Customers table and shows the service provided in the relationship. |
| **Constraint** – This table has a constraint which prevents duplicate entries with the same Customer1Id, TypeCodeId, and Customer2Id. | | | | | | |

**Table: dbo.TypeCodes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Descriptive Name** | **Valid Values** | **Index Column** | **Allow Nulls** | **Description** |
| TypeCodeId | int | Type code ID number | Values are system generated number | Y | N | This unique reference is the primary key for this table. |
| TypeCodeName | varchar(50) | Descriptive name of type code | Alphanumeric up to 50 characters. Should be descriptive of the TypeCode | N | N | This is a descriptive name for the code |
| TypeCodeCategory | varchar(20) | Category | Alphanumeric up to 20 characters. Should be descriptive. | N | N | This indicates what the code relates to. Examples include ‘Customer’, ‘Account’, ‘Association’. |
| TypeCodeActive | bit | Is this code active? | 1 is active, 0 is not active | N | N | If this is 1, then the Type Code is active and can be used. If this is 0, then the TypeCode has been deleted. |

**Linked From**

|  |  |  |
| --- | --- | --- |
| **Table** | **Join** | **Description** |
| dbo.Customers | **dbo. Customers.**TypeCodeId = **dbo. TypeCodes.**TypeCodeId | Each customer has a customer type, for example Personal or Corporate |
| dbo.Accounts | **dbo. Accounts.**TypeCodeId = **dbo. TypeCodes.**TypeCodeId | Each account has an account type, for example Current or Savings |
| dbo.Associations | **dbo. Associations.**TypeCodeId = **dbo.TypeCodes.**TypeCodeId | Each association has a type which describes the service provider, for example Lawyer or Accountant. |
| dbo.Securities | **dbo. Securities.**TypeCodeId = **dbo.TypeCodes.**TypeCodeId | Each security has a type which describes the service provider, for example Property or Shares. |

**Table: dbo.Addresses**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Descriptive Name** | **Valid Values** | **Index Column** | **Allow Nulls** | **Description** |
| AddressId | int | Address ID number | Values are system generated number | Y | N | This unique reference is the primary key for this table. |
| AddressLine1 | varchar(50) | Address Line 1 | Alphanumeric characters u p to 50 characters. | N | N | This is the first line on the address. |
| AddressLine2 | varchar(50) | Address Line 1 | Alphanumeric characters u p to 50 characters. | N | Y | This is the second line on the address. This can be left blank. |
| City | varchar(20) | City name | City name | N | N | This is the City on the address |
| StateProvince | varchar(20) | State or Province name | State or province name | N | Y | This is the state or province on the address. This can be left blank. |
| PostCode | varchar(10) | Post Code | Post code | N | Y | This is the postcode of the address. This can be left blank. |
| Country | varchar(20) | Country Name | Country name | N | N | This is the country of the address. |
| CustomerId | int | Customer ID | ID reference from the Customers table. | N | N | This ID links the address to a single customer. |

**Linked From**

|  |  |  |
| --- | --- | --- |
| **Table** | **Join** | **Description** |
| dbo.Accounts | **dbo. Addresses.**AddressId = **dbo. Accounts.**AddressId | Each account has a unique address for statements. Multiple customers can use the same address. |

**Views**

**View: vwDeletedTypes**

This view returns all entries in the TypeCodes table where TypeCodeActive is 0 indicating the type code has been deleted.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Descriptive Name** | **Description** |
| TypeCodeId | int | Type code ID number | This unique reference is the primary key for this table. |
| TypeCodeName | varchar(50) | Descriptive name of type code | This is a descriptive name for the code |
| TypeCodeCategory | varchar(20) | Category | This indicates what the code relates to. Examples include ‘Customer’, ‘Account’, ‘Association’. |
| TypeCodeActive | bit | Is this code active? | In this view, this field will always be 0. |

**View: vwCustomerAssocations**

This view returns a full list of associations between all customers expressed in both directions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Descriptive Name** | **Description** |
| Associations | varchar | List of Associations | This field is a text entry of the format “[Customer 1] has [Association type] who is [Customer 2]” and “[Customer 2] is the [Association type] for [Customer 1]”. Each relationship is expressed both ways. |

**Procedures**

|  |  |  |
| --- | --- | --- |
| **Procedure Name** | **Parameters** | **Description** |
| CreateNewCustomerAndAssociation | @NewCustomerName, @NewCustomerTypeCodeId, @ExistingCustomerId, @AssociationCodeId | This procedure creates a new customer and links it as an association to the existing customer. This is of the format [ExistingCustomer] has a [Association] who is [NewCustomer] |
| DeleteCustomer | @CustomerId | This procedure deletes the customer of the specified customer id and all associated details. This includes entries on the accounts and addresses tables with the same customerId, and entries in the securities database that are linked to accounts which in turn are linked to the customer id. |
| DeleteTypeCode | @TypeCodeId | This procedure is a soft delete which updates the type code entry to change the TypeCodeActive marker to 0 for not active. |

## Testing

The database was tested using sample data, see attached.

Data with errors was included to confirm that constraints were properly working.

Procedures were executed using sample data with the results then confirmed by viewing the updated tables.

Testing SQL scripts

Test data insert



Test of Procedures and views



## Reflections on Learnings

While working through this assignment I found there was a certain element of scope creep. For example - I knew customers could be corporate or personal, so it made sense to have two separate tables to contain the distinct information. This and other ideas quickly increased the potential workload.

Rather than going too far into these extra options, I scaled back the project to focus on the actual requests. For future real world situations, I’ll aim to include or discount these ideas early in the process, setting a firm scope as soon as possible before estimating any timeframe.

Appendix A: References

| Document Name | Document Location and/or URL | Purpose |
| --- | --- | --- |
| DatabaseDesignDocument.docx | <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/DatabaseDesignDocument.docx> | This document was used as the original template for this report. |
| Dataedo repository.pdf | <https://dataedo.com/download/Dataedo%20repository.pdf> | Example Data Dictionary with detailed tables. |
| W3C Schools | <https://www.w3schools.com/sql> | SQL code reference guide. |
| SQL Style Guide | <http://www.sqlstyle.guide/> | Style guide used when creating SQL code |