Project 1 – SQL SERVER

Databases and Business Applications

Technical Document

***Scope of the document:***

* Develop a data model for Bank Corp Ltd.
* Design and build a database to store and retrieve data for a business application.
* Focus on the technical design and provision of a new database server and the development of a new database for Bank Corp Ltd, accompanied with a Technical Design document.
* Create a single database to consolidate all the details into one single source.
* This project delivers the design and development of the database only.
* The data model is designed to meet 3NF (Third Normal Form) requirements.

***Project Requirements****:*

* 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 e.g. a Property Development company may be associated with a Surveying company.
* Associations can be two-way. For example: Company A is the Surveyor of Company B and Company B has Surveyor Company A.
* A customer cannot be associated to another customer more than once under the same Association Type.

***Key Entities List given:***

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

***Data Updates Requirement:***

1. Bank Corp needs the ability to create a new Customer and Association (from the newly created customer to another customer) via a parameterised Stored Procedure.
2. They want the ability to delete a customer and foreign key records associated with this Customer via a parameterised Stored Procedure.

***Reference Data Requirement:***

Accounts, Securities and Associations can be of several types. They do **not** want many tables to store each of these types. They want you to come up with a generic way to store all reference data in the database.

***Management Information Reporting:***

A single view on all their clients and the touch points between them – an SQL View which will return the association details between all clients (both sides of the relationship!

***Deleted Reference Data View***

The company also want visibility of what Reference Data has been deleted. They want to be able to query a SQL View which shows all reference data which has been deleted.

PROJECT SOLUTION

Key Entities and Data Dictionary:

Databases (1)

*  [Project1](#NIB8+1qdG5XrKmfZBwMKpHgdoXU=)

Server Properties

|  |  |
| --- | --- |
| Property | Value |
| Product | Microsoft SQL Server |
| Version | 12.0.2269.0 |
| Language | English (United States) |
| Platform | NT x64 |
| Edition | Express Edition (64-bit) |
| Processors | 4 |
| OS Version | 6.3 (17134) |
| Physical Memory | 6019 |
| Is Clustered | False |
| Root Directory | c:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL |
| Collation | Latin1\_­General\_­CI\_­AS |

|  |
| --- |
| **Project1 Database** |

Database Properties

|  |  |
| --- | --- |
| Property | Value |
| SQL Server Version | SQL Server 2014 |
| Compatibility Level | SQL Server 2014 |
| Last backup time | - |
| Last log backup time | - |
| Creation date | Jun 14 2019 |
| Users | 4 |
| Database Encryption Enabled | False |
| Database Encryption Algorithm | None |
| Database size | 7.00 MB |
| Unallocated space | 2.38 MB |

|  |
| --- |
| **Tables** |

Objects

|  |
| --- |
| Name |
| [dbo.Accounts](#L+lUmbt4+xb81MeRgd9IvbyshoE=) Customers Accounts |
| [dbo.Associations](#4eucx0WCSHNlHKA/cejZrruq6Cs=) Customers Associations |
| [dbo.Clients](#0fXt97gz2YrfhNxSR7bRqrGrZUk=) All Clients |
| [dbo.Reference](#FziJF4JE7Z53Znkyfm41hpwgH/c=) A Reference table for all types like public, corporate, surveyor, cash,... |
| [dbo.Reference­Deleted](#dejimJnYGBlVybH0K55NI9eBAzY=) All deleted records from Reference table |
| [dbo.Securities](#kpo40LK7CwHCKq0TCQfqVY1Lj2k=) Securities table |

|  |
| --- |
| **[dbo].[Accounts]** |

MS\_­Description

Customers Accounts

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Account­Id | int | 4 | False | 1 - 1 |  |
|  | Acc­Ref­Id | int | 4 | False |  |  |
|  | Balance | decimal(18,2) | 9 | True |  |  |
|  | Client­Id | int | 4 | False |  |  |
|  | Is­Active | bit | 1 | False |  | ((1)) |
|  | Date­Created | datetime2(0) | 6 | True |  | (sysdatetime()) |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Accounts | Account­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Accounts\_­Clients | Client­Id->[[dbo].[Clients].[Client­Id]](#0fXt97gz2YrfhNxSR7bRqrGrZUk=) |
| fk\_­Accounts\_­Reference | Acc­Ref­Id->[[dbo].[Reference].[Ref­Id]](#FziJF4JE7Z53Znkyfm41hpwgH/c=) |

|  |
| --- |
| ***[dbo].[Associations]*** |

MS\_­Description

Customers Associations

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity |
|  | Assoc­Id | int | 4 | False | 1 - 1 |
|  | Client1 | int | 4 | False |  |
|  | Assoc­Ref­Id | int | 4 | False |  |
|  | Client2 | int | 4 | False |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Associations | Assoc­Id | True |
|  | Unique\_­Association | Client1, Assoc­Ref­Id, Client2 | True |

Check Constraints

|  |  |
| --- | --- |
| Name | Constraint |
| Check\_­Association | ([Client1]<>[Client2]) |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Associations\_­Clients1 | Client1->[[dbo].[Clients].[Client­Id]](#0fXt97gz2YrfhNxSR7bRqrGrZUk=) |
| FK\_­Associations\_­Clients2 | Client2->[[dbo].[Clients].[Client­Id]](#0fXt97gz2YrfhNxSR7bRqrGrZUk=) |
| fk\_­Associations\_­Reference | Assoc­Ref­Id->[[dbo].[Reference].[Ref­Id]](#FziJF4JE7Z53Znkyfm41hpwgH/c=) |

|  |
| --- |
| [dbo].[Clients] |

MS\_­Description

All Clients

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity |
|  | Client­Id | int | 4 | False | 1 - 1 |
|  | Client­Ref­Id | int | 4 | False |  |
|  | FName | varchar(20) | 20 | False |  |
|  | LName | varchar(20) | 20 | False |  |
|  | Date­Of­Birth | date | 3 | True |  |
|  | Company­Name | varchar(50) | 50 | True |  |
|  | City | varchar(20) | 20 | True |  |
|  | Country | varchar(20) | 20 | True |  |
|  | Phone­No | varchar(10) | 10 | True |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Clients | Client­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| fk\_­Clients\_­Reference | Client­Ref­Id->[[dbo].[Reference].[Ref­Id]](#FziJF4JE7Z53Znkyfm41hpwgH/c=) |

|  |
| --- |
| [dbo].[Reference] |

MS\_­Description

A Reference table for all types like public, corporate, surveyor, cash,...

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Ref­Id | int | 4 | False | 1 - 1 |  |
|  | Ref­Description | varchar(50) | 50 | True |  |  |
|  | Is­Deleted | bit | 1 | False |  | ('0') |
|  | Date­Deleted | date | 3 | True |  |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Reference | Ref­Id | True |

Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ANSI Nulls On | Quoted Identifier On | On |
| Softdelete | True | True | Instead Of Delete |

|  |
| --- |
| [dbo].[Reference­Deleted] |

MS\_­Description

All deleted records from Reference table

Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity | Default |
|  | Ref­Del­Id | int | 4 | False | 1 - 1 |  |
|  | Ref­Id | int | 4 | False |  |  |
|  | Ref­Description | varchar(50) | 50 | False |  |  |
|  | Is­Deleted | bit | 1 | True |  | ('1') |
|  | Date­Deleted | date | 3 | True |  |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Reference­Deleted | Ref­Del­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Reference­Deleted\_­Reference | Ref­Id->[[dbo].[Reference].[Ref­Id]](#FziJF4JE7Z53Znkyfm41hpwgH/c=) |

|  |
| --- |
| [dbo].[Securities] |

MS\_­Description

Securities table

Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Data Type | Max Length (Bytes) | Allow Nulls | Identity |
|  | Security­Id | int | 4 | False | 1 - 1 |
|  | Security­Ref­Id | int | 4 | False |  |
|  | Account­Id | int | 4 | False |  |
|  | Value | decimal(10,2) | 9 | True |  |

Indexes

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Name | Key Columns | Unique |
|  | PK\_­Securities | Security­Id | True |

Foreign Keys

|  |  |
| --- | --- |
| Name | Columns |
| FK\_­Securities\_­Accounts | Account­Id->[[dbo].[Accounts].[Account­Id]](#L+lUmbt4+xb81MeRgd9IvbyshoE=) |
| fk\_­Securities\_­Reference | Security­Ref­Id->[[dbo].[Reference].[Ref­Id]](#FziJF4JE7Z53Znkyfm41hpwgH/c=) |

TECHNOLOGY USED

* SQL Server Express With Tools 2014, ExpressAndTools 64BIT\SQLEXPRWT\_x64\_ENU.exe (For 64 bit OS)
* Git Hub – used to create a ‘remote’ database repository on the GitHub server.
* RedGate Software – SQL Source Control 7 used to **Link database to source control** GitHub.

REFERENCES

* <https://www.red-gate.com/hub/product-learning/sql-source-control/github-and-sql-source-control>

SCRIPTING

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* create dbo.Clients table \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE TABLE dbo.Clients(

ClientId INT IDENTITY(1,1) NOT NULL,

ClientRefId INT NOT NULL,

FName VARCHAR(20) NOT NULL,

LName VARCHAR(20) NOT NULL,

DateOfBirth DATE NULL,

CompanyName VARCHAR(50) NULL,

City VARCHAR(20) NULL,

Country VARCHAR(20) NULL,

PhoneNo VARCHAR(10) NULL

CONSTRAINT PK\_Clients PRIMARY KEY(ClientId)

)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* create dbo.Reference table \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE TABLE dbo.Reference(

RefId INT IDENTITY(1,1) NOT NULL,

RefDescription VARCHAR(50) NOT NULL,

IsDeleted BIT DEFAULT '0' NOT NULL,

DateDeleted DATE NULL

CONSTRAINT PK\_Reference PRIMARY KEY(RefId)

)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* now we can create an FK for Clients table to refer to the PK in Reference table \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ALTER TABLE Clients

ADD CONSTRAINT fk\_Clients\_Reference FOREIGN KEY(ClientRefId) REFERENCES Reference(RefId)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* create dbo.Accounts table \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE TABLE dbo.Accounts(

AccountId INT IDENTITY(1,1) NOT NULL,

AccRefId INT NOT NULL,

Balance DECIMAL(18,2) NULL,

ClientId INT NOT NULL,

IsActive BIT DEFAULT(1) NOT NULL,

DateCreated DATETIME2(0) DEFAULT (SYSDATETIME())

CONSTRAINT PK\_Accounts PRIMARY KEY(AccountId)

)

ALTER TABLE Accounts

ADD CONSTRAINT FK\_Accounts\_Clients FOREIGN KEY(ClientId) REFERENCES Clients(ClientId)

ALTER TABLE Accounts

ADD CONSTRAINT fk\_Accounts\_Reference FOREIGN KEY(AccRefId) REFERENCES Reference(RefId)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* create dbo.Associations table \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE TABLE dbo.Associations(

AssocId INT IDENTITY(1,1) NOT NULL,

Client1 INT NOT NULL,

AssocRefId INT NOT NULL,

Client2 INT NOT NULL,

CONSTRAINT PK\_Associations PRIMARY KEY(AssocId)

)

ALTER TABLE dbo.Associations

ADD CONSTRAINT fk\_Associations\_Reference FOREIGN KEY(AssocRefId) REFERENCES Reference(RefId)

ALTER TABLE dbo.Associations

ADD CONSTRAINT FK\_Associations\_Clients1 FOREIGN KEY(Client1) REFERENCES Clients(ClientId)

ALTER TABLE dbo.Associations

ADD CONSTRAINT FK\_Associations\_Clients2 FOREIGN KEY(Client2) REFERENCES Clients(ClientId)

ALTER TABLE dbo.Associations

ADD CONSTRAINT Unique\_Association UNIQUE (Client1,AssocRefId,Client2); --A customer cannot be associated more than once under the same association type

ALTER TABLE dbo.Associations

ADD CONSTRAINT Check\_Association CHECK (Client1 <> Client2);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* create dbo.Securities table \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE TABLE dbo.Securities(

SecurityId INT IDENTITY(1,1) NOT NULL,

SecurityRefId INT NOT NULL,

AccountId INT NOT NULL,

Value decimal(10,2) null

CONSTRAINT PK\_Securities PRIMARY KEY(SecurityId)

)

ALTER TABLE dbo.Securities

ADD CONSTRAINT FK\_Securities\_Accounts FOREIGN KEY(AccountId) REFERENCES Accounts(AccountId)

ALTER TABLE dbo.Securities

ADD CONSTRAINT fk\_Securities\_Reference FOREIGN KEY(SecurityRefId) REFERENCES Reference(RefId)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* inserting data into tables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

INSERT INTO dbo.Reference (RefDescription)

VALUES ('Personal Client'),

('Corporate Client'),

('Accountant'),

('Surveyor'),

('Consultant'),

('Cleaner'),

('Savings Account'),

('Loan Account'),

('Mortgage'),

('Cash'),

('Shares'),

('Property'),

('Solicitor')

SELECT \* FROM dbo.Reference

INSERT INTO dbo.Clients (ClientRefId, FName, LName, DateOfBirth, CompanyName, City, Country, PhoneNo) VALUES

('2','Mark','Regan','06/09/1970','Regan Accountants','Dublin','Ireland','0865594447'),

('1','Dana','Coleman','01/03/1977','','Cork','Ireland','0835523332'),

('2','Diana','Mandaji','08/03/1977','Oregon Surveyors','Wicklow','Ireland','0867979856'),

('2','Paul','Paullis','01/03/1977','Paullis Developers','Dublin','Ireland','0862231118'),

('2','Orla','Collins','09/05/1969','Collins Solicitors','Dublin','Ireland','0851294456'),

('1','Roxana','Butenco','04/07/1982','','Galati','Romania','0835523897'),

('1','Mihail','Valenko','06/03/1986','','Cahul','Moldova','0868579132'),

('2','Ally','Rene','02/05/1960','Rene Developers','Dublin','Ireland','0862235558')

SELECT \* FROM dbo.Clients

DELETE FROM dbo.Clients WHERE ClientId BETWEEN 16 AND 25

DBCC CHECKIDENT ('dbo.Clients', RESEED, 0)

DBCC CHECKIDENT ('dbo.Clients', NORESEED)

INSERT INTO dbo.Accounts (AccRefId, Balance, ClientId, IsActive)

VALUES ('7', 56000,'3','1'),

('7', 56000,'1','1'),

('8', 46000,'2','1'),

('9', 200000,'6','1'),

('7', 654000,'4','1'),

('7', 56000,'5','1'),

('8', 46000,'7','1'),

('9', 350000,'1','1'),

('11', 100000,'5','1'),

('7', 15,'1','0'),

('12', 46000,'4','1'),

('10', 35,'8','0')

SELECT \* FROM dbo.Accounts

/\*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 e.g. a Property Development

company may be associated with a Surveying company. \*/

INSERT INTO dbo.Associations (Client1, AssocRefId, Client2)

VALUES

('1','3','4'),

('6','5','4'),

('3','4','4'),

('5','13','1'),

('5','13','4'),

('2','6','3'),

('7','5','4')

SELECT \* FROM dbo.Associations

/\*\*\*\*\*\*\*\* Associations can be two-way.

For example: Company A is the Surveyor of Company B and Company B has Surveyor Company A.

A customer cannot be associated to another customer more than once under the same

Association Type.

Client1 and Client2 is refering to a different ClientId \*\*\*\*\*\*\*\*\*\*/

INSERT INTO dbo.Associations (Client1, AssocRefId, Client2) VALUES ('4','3','1')

/\*A customer cannot be associated to another customer more than once

under the same Association Type. The following line will give this error:

"Violation of UNIQUE KEY constraint 'Unique\_Association'. Cannot insert duplicate key..." \*/

/\* INSERT INTO dbo.Associations (Client1, AssocRefId, Client2) VALUES ('4','3','1') \*/

INSERT INTO dbo.Securities (SecurityRefId, AccountId, Value)

VALUES ('10','4', 5600),

('11','8', 230000),

('12','8', 150000),

('10','5', 500000),

('10','6', 3650),

('11','5', 260000),

('12','7', 150000)

SELECT \* FROM dbo.Securities

/\*create a new Customer and Association (from the newly created customer to another customer)

via a parameterised Stored Procedure. \*/

/\*User defined stored procedure\*/

/\*\*\*\*\*\*\*\* CREATE NEW CUSTOMER AND ASSOCIATION STORED PROCEDURE \*\*\*\*\*\*\*\*\*\*\*/

create procedure uspCreateNewCustomer

(

@FName VARCHAR(20),

@LName VARCHAR(20),

@DateOfBirth DATE,

@CompanyName VARCHAR(50),

@City VARCHAR(20),

@Country VARCHAR(20),

@PhoneNo VARCHAR(10),

@Client2 int ,

@AssocRefId int,

@ClientRefId int

)

as

begin

insert into Clients

(ClientRefId, FName, LName, DateOfBirth, CompanyName, City, Country, PhoneNo)

values

(@ClientRefId, @FName, @LName, @DateOfBirth, @CompanyName, @City, @Country, @PhoneNo)

Declare @ClientId int = @@identity

insert into Associations (Client1, AssocRefId, Client2)

VALUES (@ClientId, @AssocRefId, @Client2)

end

go

/\* now we can use the above created stored procedure to add new data\*/

exec uspCreateNewCustomer

@FName = 'Allan',

@LName = 'Bloom',

@DateOfBirth = '06-06-1979',

@CompanyName = '',

@City = 'Limerick',

@Country = 'Ireland',

@PhoneNo = '0864472221',

@Client2 = '4',

@AssocRefId = '4',

@ClientRefId = '1'

select \* from Clients

select \* from Associations

/\* delete a customer and foreign key records associated with this customer via a parameterised Stored Procedure. \*/

/\* create a stored procedure to delete client and FKs associated with it\*/

CREATE PROCEDURE uspDeleteClientAndFKs

@ClientId INT

AS

DELETE FROM Accounts WHERE ClientId = @ClientId

DELETE FROM Associations WHERE Client1 = @ClientId

DELETE FROM Clients WHERE ClientId = @ClientId

go

/\*we now need to trace back all FKs dependent on PK ClientId \*/

sp\_help dbo.Clients --will give all information about table, including all foreign keys

sp\_fkeys @fktable\_name='Accounts' --give all FKs from that given table

/\* delete client and its FKs using the above stored procedure\*/

exec uspDeleteClientAndFKs '9'

select \* from Accounts

select \* from Associations

select \* from Clients

/\*\*\*\*\*\*\* IMPLEMENTING SOFT DELETE FOR DATA INTEGRITY \*\*\*\*\*\*\*\*/

/\*\*\*\*\*\* create a table for deletes \*\*\*\*\*\*/

CREATE TABLE dbo.ReferenceDeleted(

RefDelId INT IDENTITY(1,1) NOT NULL,

RefId INT NOT NULL,

RefDescription VARCHAR(50) NOT NULL,

IsDeleted BIT DEFAULT '1',

DateDeleted DATE NULL

CONSTRAINT PK\_ReferenceDeleted PRIMARY KEY(RefDelId)

)

ALTER TABLE dbo.ReferenceDeleted

ADD CONSTRAINT FK\_ReferenceDeleted\_Reference FOREIGN KEY(RefId) REFERENCES Reference(RefId)

/\*ALTER TABLE dbo.ReferenceDeleted

drop CONSTRAINT FK\_ReferenceDeleted\_Reference

drop table ReferenceDeleted\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SOFTDELETE TRIGGER \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE TRIGGER Softdelete

ON dbo.Reference

INSTEAD OF DELETE

AS

DECLARE @RefId int

DECLARE @RefDescription VARCHAR(50)

DECLARE @IsDeleted bit

DECLARE @DateDeleted date

SELECT RefId = @RefId

FROM sys.dm\_exec\_sessions

WHERE session\_id = @@SPID

SET @IsDeleted = '1'

-- insert the deleted row in ReferenceDeleted table

IF EXISTS ( SELECT 0 FROM Deleted )

BEGIN

IF EXISTS ( SELECT 0 FROM Inserted )

BEGIN

INSERT INTO dbo.ReferenceDeleted

( RefId,RefDescription, IsDeleted, DateDeleted )

SELECT RefId , D.RefDescription , @IsDeleted , GETDATE() 'U'

FROM Deleted D

END

ELSE

BEGIN

INSERT INTO dbo.ReferenceDeleted

( RefId, RefDescription, IsDeleted, DateDeleted )

SELECT RefId , D.RefDescription , @IsDeleted , GETDATE() 'D'

FROM Deleted D

END

END

ELSE

BEGIN

INSERT INTO dbo.ReferenceDeleted

( RefId, RefDescription, IsDeleted, DateDeleted )

SELECT RefId , I.RefDescription , @IsDeleted , GETDATE() 'I'

FROM Inserted I

END

--update the deleted row in Reference table

UPDATE u SET u.RefDescription = NULL, u.IsDeleted = '1', u.DateDeleted = CONVERT(CHAR(20), GETDATE() )

FROM dbo.Reference AS u

INNER JOIN deleted as d

ON u.RefId = d.RefId

GO

/\* END OF TRIGGER \*/

delete from Reference where RefId = '5'

select \* from Reference

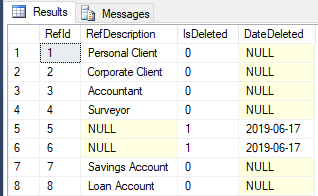
select \* from ReferenceDeleted

--to allow update on RefDescription field

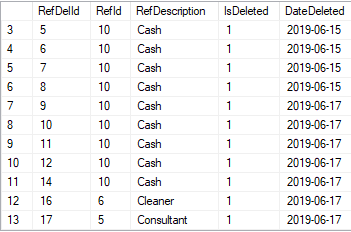
alter table dbo.Reference

alter column RefDescription varchar(50) NULL

Reference table showing updated rows instead of delete:



ReferenceDeleted table showing inserts due to Softdelete trigger:



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF SOFTDELETE TRIGGER \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*For Management Information Reporting:

We need to create a single view on all the clients and the touch points between them – an SQL View which will return the association details between all clients (both sides of the relationship)! \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Management Information Extract View \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CREATE VIEW vwMIExtract

--WITH SCHEMABINDING

AS

SELECT temp.\*,C2.Fname AS a,

CONCAT (temp.FName,' ',temp.LName,' ',temp.CompanyName,' ', + ' is ' + temp.RefDescription + ' for '+ C2.FName,' ', C2.LName,' ', C2.CompanyName) as FirstGradeAssociation,

CONCAT (C2.FName,' ',C2.LName,' ',C2.CompanyName,' ', + ' has ' + temp.RefDescription + ' ', ' ', temp.FName,' ', temp.LName,' ', temp.CompanyName) as SecondGradeAssociation

FROM ( SELECT

A.AssocId,

A.Client1,

C.Fname,

C.LName,

C.CompanyName,

A.AssocRefId,

R.RefDescription,

A.Client2

FROM dbo.Associations A

INNER JOIN dbo.Clients C

ON

A.Client1 = C.ClientId

INNER JOIN dbo.Reference R

ON

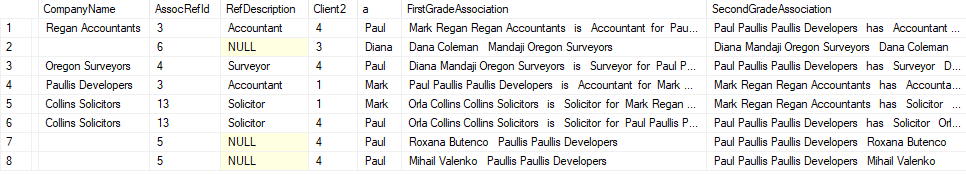
A.AssocRefId = R.RefId

)

AS temp

INNER JOIN Clients C2 ON C2.ClientId=temp.Client2

WITH CHECK OPTION



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Deleted Reference Data View \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*The company also want visibility of what Reference Data has been deleted.

They want to be able to query a SQL View which shows all reference data which has been deleted.

\*/

CREATE VIEW vwReferenceDeleted

AS

SELECT \* FROM ReferenceDeleted

SELECT \* FROM vwReferenceDeleted

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end of Deleted Reference Data View \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/