



TESTING MANUAL

Ben-Louis Opperman 1649



JUNE 14, 2019

DBD621

Contents

Introduction.....	2
Tables.....	3
Users table.....	3
Staff tables	6
Comics tables.....	9
Sales table	12
Total Sales views	16
View to show how many customers there are in each region.....	18
Stored procedures.....	20
Insert stored procedures	20
Testing the stored procedures	22
Update stored procedures.....	25
Testing stored procedures	26
Delete stored procedures and triggers.....	30
History tables	31
Triggers	32
Testing stored procedures and triggers	34
Conclusion	38

Introduction

CTU Comic Solutions is a comic book distribution centre that sells comic books to comics stores all across South Africa. My company has tasked me with handling the client and creating a database for them in accordance to their requirements. After creating the database and implementing its functionality, I proceeded to test the functionality of the database. The following document contains the testing results.

Tables

Users table

```
USE ComicsDB
GO
```

```
CREATE TABLE Bloemfontein.Users(
    UserID          INT IDENTITY PRIMARY KEY,
    UserName        VARCHAR(15),
    EmailAddress     NVARCHAR(50)
                    CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                EmailAddress NOT LIKE '%@%@%' ),
    UserFullName     NVARCHAR(25),
    [Password]       VARCHAR(15),
    DateCreated      DATE DEFAULT GETDATE(),
    ClosestBranch    VARCHAR(20)
);
```

```
CREATE TABLE Boksburg.Users(
    UserID          INT IDENTITY PRIMARY KEY,
    UserName        VARCHAR(15),
    EmailAddress     NVARCHAR(50)
                    CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                EmailAddress NOT LIKE '%@%@%' ),
    UserFullName     NVARCHAR(25),
    [Password]       VARCHAR(15),
    DateCreated      DATE DEFAULT GETDATE(),
    ClosestBranch    VARCHAR(20)
);
```

```
CREATE TABLE Durban.Users(
    UserID          INT IDENTITY PRIMARY KEY,
    UserName        VARCHAR(15),
    EmailAddress     NVARCHAR(50)
                    CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                EmailAddress NOT LIKE '%@%@%' ),
    UserFullName     NVARCHAR(25),
    [Password]       VARCHAR(15),
    DateCreated      DATE DEFAULT GETDATE(),
    ClosestBranch    VARCHAR(20)
);
```

```
CREATE TABLE PortElizabeth.Users(
    UserID          INT IDENTITY PRIMARY KEY,
    UserName        VARCHAR(15),
    EmailAddress     NVARCHAR(50)
                    CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                EmailAddress NOT LIKE '%@%@%' ),
    UserFullName     NVARCHAR(25),
    [Password]       VARCHAR(15),
    DateCreated      DATE DEFAULT GETDATE(),
    ClosestBranch    VARCHAR(20)
);
```

```
CREATE TABLE Polokwane.Users(
    UserID          INT IDENTITY PRIMARY KEY,
    UserName        VARCHAR(15),
    EmailAddress     NVARCHAR(50)
                    CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                EmailAddress NOT LIKE '%@%@%' ),
    UserFullName     NVARCHAR(25),
    [Password]       VARCHAR(15),
```

```

        DateCreated          DATE DEFAULT GETDATE(),
        ClosestBranch        VARCHAR(20)
    );

CREATE TABLE Vereeniging.Users(
    UserID                   INT IDENTITY PRIMARY KEY,
    UserName                 VARCHAR(15),
    EmailAddress             NVARCHAR(50)
                             CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                             EmailAddress NOT LIKE '%@%@%' ),

    UserFullName             NVARCHAR(25),
    [Password]              VARCHAR(15),
    DateCreated              DATE DEFAULT GETDATE(),
    ClosestBranch            VARCHAR(20)
);

CREATE TABLE Potchefstroom.Users(
    UserID                   INT IDENTITY PRIMARY KEY,
    UserName                 VARCHAR(15),
    EmailAddress             NVARCHAR(50)
                             CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                             EmailAddress NOT LIKE '%@%@%' ),

    UserFullName             NVARCHAR(25),
    [Password]              VARCHAR(15),
    DateCreated              DATE DEFAULT GETDATE(),
    ClosestBranch            VARCHAR(20)
);

CREATE TABLE CapeTown.Users(
    UserID                   INT IDENTITY PRIMARY KEY,
    UserName                 VARCHAR(15),
    EmailAddress             NVARCHAR(50)
                             CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                             EmailAddress NOT LIKE '%@%@%' ),

    UserFullName             NVARCHAR(25),
    [Password]              VARCHAR(15),
    DateCreated              DATE DEFAULT GETDATE(),
    ClosestBranch            VARCHAR(20)
);

CREATE TABLE Pretoria.Users(
    UserID                   INT IDENTITY PRIMARY KEY,
    UserName                 VARCHAR(15),
    EmailAddress             NVARCHAR(50)
                             CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                             EmailAddress NOT LIKE '%@%@%' ),

    UserFullName             NVARCHAR(25),
    [Password]              VARCHAR(15),
    DateCreated              DATE DEFAULT GETDATE(),
    ClosestBranch            VARCHAR(20)
);

CREATE TABLE AucklandPark.Users(
    UserID                   INT IDENTITY PRIMARY KEY,
    UserName                 VARCHAR(15),
    EmailAddress             NVARCHAR(50)
                             CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                             EmailAddress NOT LIKE '%@%@%' ),

    UserFullName             NVARCHAR(25),
    [Password]              VARCHAR(15),
    DateCreated              DATE DEFAULT GETDATE(),
    ClosestBranch            VARCHAR(20)
);

```

```

);

CREATE TABLE Roodepoort.Users(
    UserID          INT IDENTITY PRIMARY KEY,
    UserName        VARCHAR(15),
    EmailAddress     NVARCHAR(50)
                    CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                EmailAddress NOT LIKE '%@@@%' ),

    UserFullName     NVARCHAR(25),
    [Password]       VARCHAR(15),
    DateCreated      DATE DEFAULT GETDATE(),
    ClosestBranch    VARCHAR(20)
);

CREATE TABLE Nelspruit.Users(
    UserID          INT IDENTITY PRIMARY KEY,
    UserName        VARCHAR(15),
    EmailAddress     NVARCHAR(50)
                    CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                EmailAddress NOT LIKE '%@@@%' ),

    UserFullName     NVARCHAR(25),
    [Password]       VARCHAR(15),
    DateCreated      DATE DEFAULT GETDATE(),
    ClosestBranch    VARCHAR(20)
);

CREATE TABLE Stellenbosch.Users(
    UserID          INT IDENTITY PRIMARY KEY,
    UserName        VARCHAR(15),
    EmailAddress     NVARCHAR(50)
                    CONSTRAINT chk_email CHECK (EmailAddress LIKE '%_@[^.]% ' AND
                                                EmailAddress NOT LIKE '%@@@%' ),

    UserFullName     NVARCHAR(25),
    [Password]       VARCHAR(15),
    DateCreated      DATE DEFAULT GETDATE(),
    ClosestBranch    VARCHAR(20)
);

```

Staff tables

USE ComicsDB

GO

```
CREATE TABLE Bloemfontein.Staff(  
    EmpNo          INT IDENTITY PRIMARY KEY,  
    FirstName      NVARCHAR(15),  
    LastName       NVARCHAR(20),  
    Branch         VARCHAR(20),  
    PhoneNo        CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),  
    HireDate       DATE DEFAULT GETDATE(),  
    JobDescription  VARCHAR(50),  
    Gender         CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),  
    Birthdate      DATE,  
    Salary         MONEY  
);
```

```
CREATE TABLE Boksburg.Staff(  
    EmpNo          INT IDENTITY PRIMARY KEY,  
    FirstName      NVARCHAR(15),  
    LastName       NVARCHAR(20),  
    Branch         VARCHAR(20),  
    PhoneNo        CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),  
    HireDate       DATE DEFAULT GETDATE(),  
    JobDescription  VARCHAR(50),  
    Gender         CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),  
    Birthdate      DATE,  
    Salary         MONEY  
);
```

```
CREATE TABLE Durban.Staff(  
    EmpNo          INT IDENTITY PRIMARY KEY,  
    FirstName      NVARCHAR(15),  
    LastName       NVARCHAR(20),  
    Branch         VARCHAR(20),  
    PhoneNo        CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),  
    HireDate       DATE DEFAULT GETDATE(),  
    JobDescription  VARCHAR(50),  
    Gender         CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),  
    Birthdate      DATE,  
    Salary         MONEY  
);
```

```
CREATE TABLE PortElizabeth.Staff(  
    EmpNo          INT IDENTITY PRIMARY KEY,  
    FirstName      NVARCHAR(15),  
    LastName       NVARCHAR(20),  
    Branch         VARCHAR(20),  
    PhoneNo        CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),  
    HireDate       DATE DEFAULT GETDATE(),  
    JobDescription  VARCHAR(50),  
    Gender         CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),  
    Birthdate      DATE,  
    Salary         MONEY  
);
```

```
CREATE TABLE Polokwane.Staff(  
    EmpNo          INT IDENTITY PRIMARY KEY,  
    FirstName      NVARCHAR(15),  
    LastName       NVARCHAR(20),  
    Branch         VARCHAR(20),  
    PhoneNo        CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),
```

```

        HireDate    DATE DEFAULT GETDATE(),
        JobDescription VARCHAR(50),
        Gender       CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
        Birthdate    DATE,
        Salary       MONEY
    );

CREATE TABLE Vereeniging.Staff(
    EmpNo           INT IDENTITY PRIMARY KEY,
    FirstName       NVARCHAR(15),
    LastName        NVARCHAR(20),
    Branch          VARCHAR(20),
    PhoneNo         CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%),
    HireDate        DATE DEFAULT GETDATE(),
    JobDescription   VARCHAR(50),
    Gender          CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
    Birthdate       DATE,
    Salary          MONEY
);

CREATE TABLE Potchefstroom.Staff(
    EmpNo           INT IDENTITY PRIMARY KEY,
    FirstName       NVARCHAR(15),
    LastName        NVARCHAR(20),
    Branch          VARCHAR(20),
    PhoneNo         CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%),
    HireDate        DATE DEFAULT GETDATE(),
    JobDescription   VARCHAR(50),
    Gender          CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
    Birthdate       DATE,
    Salary          MONEY
);

CREATE TABLE CapeTown.Staff(
    EmpNo           INT IDENTITY PRIMARY KEY,
    FirstName       NVARCHAR(15),
    LastName        NVARCHAR(20),
    Branch          VARCHAR(20),
    PhoneNo         CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%),
    HireDate        DATE DEFAULT GETDATE(),
    JobDescription   VARCHAR(50),
    Gender          CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
    Birthdate       DATE,
    Salary          MONEY
);

CREATE TABLE AucklandPark.Staff(
    EmpNo           INT IDENTITY PRIMARY KEY,
    FirstName       NVARCHAR(15),
    LastName        NVARCHAR(20),
    Branch          VARCHAR(20),
    PhoneNo         CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%),
    HireDate        DATE DEFAULT GETDATE(),
    JobDescription   VARCHAR(50),
    Gender          CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
    Birthdate       DATE,
    Salary          MONEY
);

CREATE TABLE Roodepoort.Staff(
    EmpNo           INT IDENTITY PRIMARY KEY,
    FirstName       NVARCHAR(15),

```



```

        LastName    NVARCHAR(20),
        Branch      VARCHAR(20),
        PhoneNo     CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),
        HireDate    DATE DEFAULT GETDATE(),
        JobDescription VARCHAR(50),
        Gender      CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
        Birthdate   DATE,
        Salary      MONEY
    );

CREATE TABLE Nelspruit.Staff(
    EmpNo          INT IDENTITY PRIMARY KEY,
    FirstName      NVARCHAR(15),
    LastName       NVARCHAR(20),
    Branch         VARCHAR(20),
    PhoneNo       CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),
    HireDate      DATE DEFAULT GETDATE(),
    JobDescription VARCHAR(50),
    Gender        CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
    Birthdate     DATE,
    Salary        MONEY
);

CREATE TABLE Stellenbosch.Staff(
    EmpNo          INT IDENTITY PRIMARY KEY,
    FirstName      NVARCHAR(15),
    LastName       NVARCHAR(20),
    Branch         VARCHAR(20),
    PhoneNo       CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),
    HireDate      DATE DEFAULT GETDATE(),
    JobDescription VARCHAR(50),
    Gender        CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
    Birthdate     DATE,
    Salary        MONEY
);

CREATE TABLE Pretoria.Staff(
    EmpNo          INT IDENTITY PRIMARY KEY,
    FirstName      NVARCHAR(15),
    LastName       NVARCHAR(20),
    Branch         VARCHAR(20),
    PhoneNo       CHAR(10) CHECK (PhoneNo LIKE '%[0-9]%'),
    HireDate      DATE DEFAULT GETDATE(),
    JobDescription VARCHAR(50),
    Gender        CHAR(1) CHECK (Gender LIKE '%[M]%' OR Gender LIKE '%[F]%),
    Birthdate     DATE,
    Salary        MONEY
);

```

Comics tables

```
USE ComicsDB
GO
```

```
CREATE TABLE Bloemfontein.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);
```

```
CREATE TABLE Boksburg.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);
```

```
CREATE TABLE CapeTown.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);
```

```
CREATE TABLE Durban.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);
```

```
CREATE TABLE Nelspruit.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);
```

```
CREATE TABLE Polokwane.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);
```

```

);

CREATE TABLE PortElizabeth.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);

CREATE TABLE Pretoria.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);

CREATE TABLE Roodepoort.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);

CREATE TABLE Stellenbosch.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);

CREATE TABLE AucklandPark.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);

CREATE TABLE Vereeniging.Comics(
    ComicId          INT IDENTITY PRIMARY KEY,
    Title            NVARCHAR(50),
    Author           NVARCHAR(50),
    [Description]    TEXT,
    Issue            INT,
    Publisher        VARCHAR(20),
    CoverPage        VARBINARY(MAX)
);

```

```
CREATE TABLE Potchefstroom.Comics(  
    ComicId          INT IDENTITY PRIMARY KEY,  
    Title            NVARCHAR(50),  
    Author           NVARCHAR(50),  
    [Description]    TEXT,  
    Issue            INT,  
    Publisher        VARCHAR(20),  
    CoverPage        VARBINARY(MAX)  
);
```

Sales table

```
USE ComicsDB
GO
```

```
CREATE TABLE Bloemfontein.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);
```

```
CREATE TABLE Boksburg.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);
```

```
CREATE TABLE Durban.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);
```

```
CREATE TABLE PortElizabeth.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);
```

```
CREATE TABLE Polokwane.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);
```

```
CREATE TABLE Vereeniging.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);
```

```
CREATE TABLE Potchefstroom.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
```

```

        BeforeTax    MONEY,
        AfterTax     AS BeforeTax + (BeforeTax * 0.15),
        Quantity     TINYINT
    );

CREATE TABLE CapeTown.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);

CREATE TABLE Pretoria.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);

















































CREATE TABLE AucklandPark.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);

































CREATE TABLE Roodepoort.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);

CREATE TABLE Nelspruit.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);

CREATE TABLE Stellenbosch.Sales(
    SaleId          INT IDENTITY PRIMARY KEY,
    ComicId         INT,
    SalesDate       DATE,
    BeforeTax       MONEY,
    AfterTax        AS BeforeTax + (BeforeTax * 0.15),
    Quantity        TINYINT
);

```

- +  AdventureWorksDW2016
 -  ComicsDB
 - +  Database Diagrams
 -  Tables
 - +  System Tables
 - +  FileTables
 - +  External Tables
 - +  Graph Tables
 - +  AucklandPark.Comics
 - +  AucklandPark.HistoryComics
 - +  AucklandPark.HistorySales
 - +  AucklandPark.HistoryStaff
 - +  AucklandPark.HistoryUsers
 - +  AucklandPark.Sales
 - +  AucklandPark.Staff
 - +  AucklandPark.Users
 - +  Bloemfontein.Comics
 - +  Bloemfontein.Sales
 - +  Bloemfontein.Staff
 - +  Bloemfontein.Users
 - +  Boksburg.Comics
 - +  Boksburg.Sales
 - +  Boksburg.Staff
 - +  Boksburg.Users
 - +  CapeTown.Comics
 - +  CapeTown.Sales
 - +  CapeTown.Staff
 - +  CapeTown.Users
 - +  Durban.Comics
 - +  Durban.Sales
 - +  Durban.Staff
 - +  Durban.Users
 - +  Nelspruit.Comics
 - +  Nelspruit.Sales
 - +  Nelspruit.Staff
 - +  Nelspruit.Users
 - +  Polokwane.Comics
 - +  Polokwane.Sales
 - +  Polokwane.Staff
 - +  Polokwane.Users
 - +  PortElizabeth.Comics
 - +  PortElizabeth.Sales
 - +  PortElizabeth.Staff
 - +  PortElizabeth.Users
 - +  Potchefstroom.Comics
 - +  Potchefstroom.Sales
 - +  Potchefstroom.Staff
 - +  Potchefstroom.Users
-

-   Pretoria.Comics
-   Pretoria.Sales
-   Pretoria.Staff
-   Pretoria.Users
-   Roodepoort.Comics
-   Roodepoort.Sales
-   Roodepoort.Staff
-   Roodepoort.Users
-   Stellenbosch.Comics
-   Stellenbosch.Sales
-   Stellenbosch.Staff
-   Stellenbosch.Users
-   Vereeniging.Comics
-   Vereeniging.Sales
-   Vereeniging.Staff
-   Vereeniging.Users

Total Sales views

```
USE ComicsDB
GO
```

```
CREATE VIEW TotalRegionSales
AS
SELECT 'AucklandPark' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM AucklandPark.Sales
UNION
SELECT 'Bloemfontein' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Bloemfontein.Sales
UNION
SELECT 'Boksburg' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Boksburg.Sales
UNION
SELECT 'CapeTown' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM CapeTown.Sales
UNION
SELECT 'Durban' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Durban.Sales
UNION
SELECT 'Nelspruit' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Nelspruit.Sales
UNION
SELECT 'Polokwane' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Polokwane.Sales
UNION
SELECT 'PortElizabeth' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM PortElizabeth.Sales
UNION
SELECT 'Potchefstroom' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Potchefstroom.Sales
UNION
SELECT 'Pretoria' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Pretoria.Sales
UNION
SELECT 'Roodepoort' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Roodepoort.Sales
UNION
SELECT 'Stellenbosch' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Stellenbosch.Sales
UNION
SELECT 'Vereeniging' AS Region, SUM(Quantity) AS TotalQuantity, SUM(AfterTax) AS
TotalAfterTax
FROM Vereeniging.Sales
GO

SELECT *
FROM TotalRegionSales;
```

```

45
46 SELECT *
47 FROM TotalRegionSales;

```

100 %

Results Messages

	Region	TotalQuantity	TotalAfterTax
1	AucklandPark	10	632.385000
2	Bloemfontein	10	632.385000
3	Boksburg	10	632.385000
4	CapeTown	10	632.385000
5	Durban	10	632.385000
6	Nelspruit	10	632.385000
7	Polokwane	10	632.385000
8	PortElizabeth	10	632.385000
9	Potchefstroom	10	632.385000
10	Pretoria	10	632.385000
11	Roodepoot	10	632.385000
12	Stellenbosch	10	632.385000
13	Vereeniging	10	632.385000

View to show how many customers there are in each region

```
USE ComicsDB
GO
```

```
CREATE VIEW TotalCustomersPerRegion
AS
SELECT 'AucklandPark' AS Region, COUNT(UserID) AS TotalCustomers
FROM AucklandPark.Users
UNION
SELECT 'Bloemfontein' AS Region, COUNT(UserID) AS TotalCustomers
FROM Bloemfontein.Users
UNION
SELECT 'Boksburg' AS Region, COUNT(UserID) AS TotalCustomers
FROM Boksburg.Users
UNION
SELECT 'CapeTown' AS Region, COUNT(UserID) AS TotalCustomers
FROM CapeTown.Users
UNION
SELECT 'Durban' AS Region, COUNT(UserID) AS TotalCustomers
FROM Durban.Users
UNION
SELECT 'Nelspruit' AS Region, COUNT(UserID) AS TotalCustomers
FROM Nelspruit.Users
UNION
SELECT 'Polokwane' AS Region, COUNT(UserID) AS TotalCustomers
FROM Polokwane.Users
UNION
SELECT 'PortElizabeth' AS Region, COUNT(UserID) AS TotalCustomers
FROM PortElizabeth.Users
UNION
SELECT 'Potchefstroom' AS Region, COUNT(UserID) AS TotalCustomers
FROM Potchefstroom.Users
UNION
SELECT 'Pretoria' AS Region, COUNT(UserID) AS TotalCustomers
FROM Pretoria.Users
UNION
SELECT 'Roodepoort' AS Region, COUNT(UserID) AS TotalCustomers
FROM Roodepoort.Users
UNION
SELECT 'Stellenbosch' AS Region, COUNT(UserID) AS TotalCustomers
FROM Stellenbosch.Users
UNION
SELECT 'Vereeniging' AS Region, COUNT(UserID) AS TotalCustomers
FROM Vereeniging.Users
GO

SELECT *
FROM TotalCustomersPerRegion;
```

```
45  
46 SELECT *  
47 FROM TotalCustomersPerRegion;
```

100 %

Results Messages

	Region	TotalCustomers
1	AucklandPark	10
2	Bloemfontein	10
3	Boksburg	10
4	CapeTown	10
5	Durban	10
6	Nelspruit	10
7	Polokwane	10
8	PortElizabeth	10
9	Potchefstroom	10
10	Pretoria	10
11	Roodepoort	10
12	Stellenbosch	10
13	Vereeniging	10

Stored procedures

Insert stored procedures

```
USE ComicsDB
GO
```

```
--Inserting data
```

```
CREATE PROCEDURE AucklandPark.NewUser
    @UserName varchar(15),
    @EmailAddress nvarchar(50),
    @UserFullName nvarchar(25),
    @Password varchar(15),
    @ClosestBranch varchar(20)
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
    BEGIN TRY
        INSERT INTO AucklandPark.Users(UserName, EmailAddress, UserFullName,
[Password], ClosestBranch)
            VALUES(@UserName, @EmailAddress, @UserFullName, @Password,
@ClosestBranch);
    END TRY
    BEGIN CATCH
        SELECT @ERR_MSG = ERROR_MESSAGE(),
            @ERR_STA = ERROR_STATE()

        SET @ERR_MSG = 'Error occured while inserting data into the table: ' +
@ERR_MSG;

        THROW 50001, @ERR_MSG, @ERR_STA;
    END CATCH
GO
```

```
CREATE PROCEDURE AucklandPark.NewStaff
    @FirstName nvarchar(15),
    @LastName nvarchar(20),
    @Branch varchar(20),
    @PhoneNo char(10),
    @JobDescription varchar(50),
    @Gender char(1),
    @Birthdate date,
    @Salary money
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
    BEGIN TRY
        INSERT INTO AucklandPark.Staff(FirstName, LastName, Branch, PhoneNo,
JobDescription, Gender, Birthdate, Salary)
            VALUES(@FirstName, @LastName, @Branch, @PhoneNo, @JobDescription,
@Gender, @Birthdate, @Salary);
    END TRY
    BEGIN CATCH
        SELECT @ERR_MSG = ERROR_MESSAGE(),
            @ERR_STA = ERROR_STATE()

        SET @ERR_MSG = 'Error occured while inserting data into the table: ' +
@ERR_MSG;

        THROW 50001, @ERR_MSG, @ERR_STA;
    END CATCH
GO
```

```
CREATE PROCEDURE AucklandPark.NewComic
    @Title nvarchar(50),
```

```

@Author nvarchar(50),
@Description varchar(max),
@Issue int,
@Publisher varchar(20),
@CoverPage varbinary(max)
AS
DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
BEGIN TRY
    INSERT INTO AucklandPark.Comics
    VALUES(@Title, @Author, @Description, @Issue, @Publisher, @CoverPage);
END TRY
BEGIN CATCH
    SELECT @ERR_MSG = ERROR_MESSAGE(),
    @ERR_STA = ERROR_STATE()

    SET @ERR_MSG = 'Error occurred while inserting data into the table: ' +
@ERR_MSG;

    THROW 50001, @ERR_MSG, @ERR_STA;
END CATCH
GO

CREATE PROCEDURE AucklandPark.NewSale
@ComicId int,
@SalesDate date,
@BeforeTax money,
@Quantity tinyint
AS
DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
BEGIN TRY
    INSERT INTO AucklandPark.Sales(ComicId, SalesDate, BeforeTax, Quantity)
    VALUES(@ComicId, @SalesDate, @BeforeTax, @Quantity);
END TRY
BEGIN CATCH
    SELECT @ERR_MSG = ERROR_MESSAGE(),
    @ERR_STA = ERROR_STATE()

    SET @ERR_MSG = 'Error occurred while inserting data into the table: ' +
@ERR_MSG;

    THROW 50001, @ERR_MSG, @ERR_STA;
END CATCH
GO

```

Testing the stored procedures

NewUser

```
EXEC AucklandPark.NewUser 'itscoolguy', 'coolguy@hotmail.com', 'Ben Shapiro',
'bnk6789', 'Auckland Park';
GO
SELECT * FROM AucklandPark.Users;
GO
```

```
18 --Testing stored procedures
19 EXEC AucklandPark.NewUser 'itscoolguy', 'coolguy@hotmail.com', 'Ben Shapiro', 'bnk6789', 'Auckland Park';
20 GO
21 SELECT * FROM AucklandPark.Users;
22 GO
23
24 DECLARE @page varbinary(max)
25 SET @page = (SELECT * FROM OPENROWSET(BULK N'A:\Documents\CTU Training\MCSE\Semester 1\Summatives\DBD621\Cover
26 EXEC AucklandPark.NewComic 'Sonic the Hedgehog',
27 'Ian Flynn',
28 'The Fate of Dr. Eggman, Part 1. After his last battle with Sonic, Dr. Eggman's gc
29 5,
30 'IDW Publishing',
31 @page;
32
```

100 %

Results Messages

(1 row affected)

(11 rows affected)

	UserID	UserName	EmailAddress	UserFullName	Password	DateCreated	ClosestBranch
1	1	john123	johsmith@yahoo.com	John Smith	ekscool123	2019-05-31	AucklandPark
2	2	samjack	samuelj@gmail.com	Samuel Jackson	sybiotic3d	2019-05-31	AucklandPark
3	3	FireZ	bill34@hotmail.com	Bill Clinton	hithere45	2019-05-31	AucklandPark
4	4	grimdown	willion@yahoo.com	William Barton	jkl23asd	2019-05-31	AucklandPark
5	5	bug	ivanf@gmail.com	Ivan Ivanhof	sdaf68dsd	2019-05-31	AucklandPark
6	6	cardiff	benl@hotmail.com	Ben Lovette	gvhyg88	2019-05-31	AucklandPark
7	7	waggler	narekaxuf-9517@yopmail.com	Nadia Dean	bhjf789dd	2019-05-31	AucklandPark
8	8	beese	jsmith@icloud.com	Jeniane Summers	advsw2134	2019-05-31	AucklandPark
9	9	grobble	rupak@me.com	Celine Dion	sdfj789	2019-05-31	AucklandPark
10	10	chissing	storem@att.net	Matt Damon	jnvgsdkf78999	2019-05-31	AucklandPark
11	11	itscoolguy	coolguy@hotmail.com	Ben Shapiro	bnk6789	2019-06-08	Auckland Park

NewStaff

```
EXEC AucklandPark.NewStaff 'Max', 'Steel', 'Auckland Park', '0834560012', 'Cashier',
'M', '1990-07-17', 7000;
GO
SELECT * FROM AucklandPark.Staff;
```

```
24 --Testing Staff
25 EXEC AucklandPark.NewStaff 'Max', 'Steel', 'Auckland Park', '0834560012', 'Cashier', 'M', '1990-07-17', 7000;
26 GO
27 SELECT * FROM AucklandPark.Staff;
28
29 DECLARE @page varbinary(max)
30 SET @page = (SELECT * FROM OPENROWSET(BULK N'A:\Documents\CTU Training\MCSE\Semester 1\Summatives\DBD621\Coverpag
31 EXEC AucklandPark.NewComic 'Sonic the Hedgehog',
32 'Ian Flynn',
```

100 %

Messages

(1 row affected)

	EmpNo	FirstName	LastName	Branch	PhoneNo	HireDate	JobDescription	Gender	Birthdate	Salary
1	1	John	Smith	Bloemfontein	0768550187	2019-05-31	Cashier	M	1996-06-05	5000.00
2	2	Sally	Wickham	Bloemfontein	0728560109	2019-05-31	Accountant	F	1994-04-08	6500.00
3	3	Angelina	Jolie	Bloemfontein	0837530142	2019-05-31	Sales	F	1995-11-28	6000.00
4	4	Brad	Pritt	Bloemfontein	0821450199	2019-05-31	Stocks	M	1992-02-04	7000.00
5	5	Tally	Ho	Bloemfontein	0844560287	2019-05-31	Cashier	M	1990-07-15	5000.00
6	6	John	Miller	Bloemfontein	0749500151	2019-05-31	Accountant	M	1995-04-12	7000.00
7	7	Ryder	Orozco	Bloemfontein	0768550187	2019-05-31	Sales	M	1991-09-23	5000.00
8	8	Nakita	Lee	Bloemfontein	0818544512	2019-05-31	Stocks	F	1992-10-09	6000.00
9	9	Jess	Beattie	Bloemfontein	0826620212	2019-05-31	Cashier	F	1993-03-21	5000.00
10	10	Christina	Conroy	Bloemfontein	0836520855	2019-05-31	Accountant	F	1996-08-11	70000.00
11	11	Max	Steel	Auckland Park	0834560012	2019-06-08	Cashier	M	1990-07-17	7000.00

NewComic

```

DECLARE @page varbinary(max)
SET @page = (SELECT * FROM OPENROWSET(BULK N'A:\Documents\CTU Training\MCSE\Semester
1\Summatives\DBD621\Coverpages\dune.jpg',SINGLE_BLOB) AS img)
EXEC AucklandPark.NewComic 'Sonic the Hedgehog',
                             'Ian Flynn',
                             'The Fate of Dr. Eggman, Part 1. After his last battle
with Sonic, Dr. Eggman's gone quiet. His Badnik forces are still causing trouble, but
without the bad doctor's usual fanfare. What's Eggman been doing all this time? And
what will happen to Sonic when he finds out?',
                             5,
                             'IDW Publishing',
                             @page;
SELECT * FROM AucklandPark.Comics;
GO

```

```

17
18 --Testing stored procedures
19 DECLARE @page varbinary(max)
20 SET @page = (SELECT * FROM OPENROWSET(BULK N'A:\Documents\CTU Training\MCSE\Semester 1\Summatives\DBD621\Coverpages\dune.jpg
21 EXEC AucklandPark.NewComic 'Sonic the Hedgehog',
22                             'Ian Flynn',
23                             'The Fate of Dr. Eggman, Part 1. After his last battle with Sonic, Dr. Eggman's gone quiet. His l
24                             5,
25                             'IDW Publishing',
26                             @page;
27
28 SELECT * FROM AucklandPark.Comics;
29 GO

```

	ComicId	Title	Author	Description	Issue	Publisher	CoverPage
1	2	Ultimate Spider-Man	Mark Bagley	Death of Spider-Man	156	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
2	3	Ultimate X-Men	Aron Coleite	Ultimatum	99	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
3	4	Dune	Ralph Macchio	Comic adpatation of the 1984 film	2	Marvel	0xFFD8FFE000104A46494600010101004800480000FFFE001...
4	5	Akham Knight	Peter Tomasi	Prequel to the Akham Knight video game	2	DC Comics	0xFFD8FFE000104A46494600010101012C012C0000FFE222...
5	6	Elektra (2017)	Matthew Owens	Running With The Devil	4	Marvel	0xFFD8FFE123A345786966000049492A0008000000000000...
6	7	Exiles	Saladin Ahmed	THE WRATH OF KHAN! The Exiles' beloved Khan-who ...	11	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
7	8	Falcon	Rodney Barnes	As Brooklyn becomes vampire central, Deacon Frost conti...	7	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
8	9	Fantastic Five	Tom DeFalco	Though his metal implants have been destroyed, a severel...	4	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
9	10	Foolkiller	Max Bemis	Foolkiller versus a bunch of skinheads!! Alone with an ars...	2	Marvel	0xFFD8FFE1260245786966000049492A0008000000000000...
10	11	Generation X	Christina Strain	It's JUBILEE vs. M-PLATE...vampire vs. vampire! Will the ...	87	Marvel	0xFFD8FFE1291945786966000049492A0008000000000000...
11	14	Sonic the Hedgehog	Ian Flynn	The Fate of Dr. Eggman, Part 1. After his last battle with S...	5	IDW Publishing	0xFFD8FFE000104A46494600010101004800480000FFFE001...

NewSale

```

EXEC AucklandPark.NewSale 14, '2019-05-31', 54.99, 2;
GO
SELECT * FROM AucklandPark.Sales;
GO

```



```

41 --Testing Sales
42 EXEC AucklandPark.NewSale 14, '2019-05-31', 54.99, 2;
43 GO

```

100 %

Messages

(1 row affected)

Results Messages

	SaleId	ComicId	SalesDate	BeforeTax	AfterTax	Quantity
1	2	2	2019-02-04	54.99	63.238500	1
2	3	2	2019-02-04	54.99	63.238500	1
3	4	3	2019-02-04	54.99	63.238500	1
4	5	5	2019-02-04	54.99	63.238500	1
5	6	4	2019-02-05	54.99	63.238500	1
6	7	6	2019-02-05	54.99	63.238500	1
7	8	9	2019-02-05	54.99	63.238500	1
8	9	10	2019-02-07	54.99	63.238500	1
9	10	8	2019-02-07	54.99	63.238500	1
10	11	7	2019-02-07	54.99	63.238500	1
11	12	14	2019-05-31	54.99	63.238500	2

Update stored procedures

USE ComicsDB

GO

--Updating data

```
CREATE PROCEDURE AucklandPark.EditUser
    @Id int,
    @UserName varchar(15),
    @EmailAddress nvarchar(50),
    @UserFullName nvarchar(25),
    @Password varchar(15),
    @ClosestBranch varchar(20)
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
    BEGIN TRY
        UPDATE AucklandPark.Users SET
            UserName = @UserName, EmailAddress = @EmailAddress, UserFullName =
@UserFullName, [Password] = @Password, ClosestBranch = @ClosestBranch
        WHERE UserID = @Id;
    END TRY
    BEGIN CATCH
        SELECT @ERR_MSG = ERROR_MESSAGE(),
            @ERR_STA = ERROR_STATE()

        SET @ERR_MSG = 'Error occured while updating the record: ' + @ERR_MSG;

        THROW 50001, @ERR_MSG, @ERR_STA;
    END CATCH
GO
```

```
CREATE PROCEDURE AucklandPark.EditStaff
    @Id int,
    @FirstName nvarchar(15),
    @LastName nvarchar(20),
    @Branch varchar(20),
    @PhoneNo char(10),
    @JobDescription varchar(50),
    @Gender char(1),
    @Birthdate date,
    @Salary money
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
    BEGIN TRY
        UPDATE AucklandPark.Staff SET
            FirstName = @FirstName,
            LastName = @LastName,
            Branch = @Branch,
            PhoneNo = @PhoneNo,
            JobDescription = @JobDescription,
            Gender = @Gender,
            Birthdate = @Birthdate,
            Salary = @Salary
        WHERE EmpNo = @Id
    END TRY
    BEGIN CATCH
        SELECT @ERR_MSG = ERROR_MESSAGE(),
            @ERR_STA = ERROR_STATE()

        SET @ERR_MSG = 'Error occured while updating the record: ' + @ERR_MSG;

        THROW 50001, @ERR_MSG, @ERR_STA;
    END CATCH
```

GO

```
CREATE PROCEDURE AucklandPark.EditComic
    @Id int,
    @Title nvarchar(50),
    @Author nvarchar(50),
    @Description varchar(max),
    @Issue int,
    @Publisher varchar(20),
    @CoverPage varbinary(max)
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
    BEGIN TRY
        UPDATE AucklandPark.Comics SET
            Title = @Title,
            Author = @Author,
            [Description] = @Description,
            Issue = @Issue,
            Publisher = @Publisher,
            CoverPage = @CoverPage
        WHERE ComicId = @Id
    END TRY
    BEGIN CATCH
        SELECT @ERR_MSG = ERROR_MESSAGE(),
            @ERR_STA = ERROR_STATE()

        SET @ERR_MSG = 'Error occured while updating the record: ' + @ERR_MSG;

        THROW 50001, @ERR_MSG, @ERR_STA;
    END CATCH
```

GO

Testing stored procedures

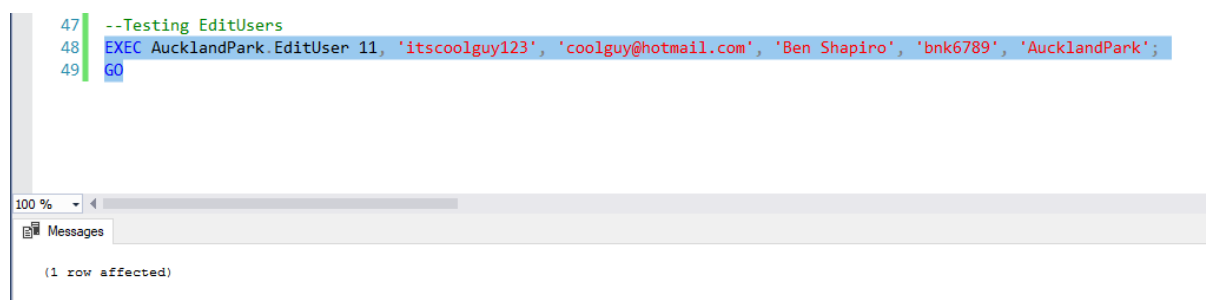
EditUser

```
EXEC AucklandPark.EditUser 11, 'itscoolguy123', 'coolguy@hotmail.com', 'Ben Shapiro',
'bnk6789', 'AucklandPark';
```

GO

```
SELECT * FROM AucklandPark.Users;
```

GO



```

50 SELECT * FROM AucklandPark.Users;
51 GO

```

100 %

	UserID	UserName	EmailAddress	UserFullName	Password	DateCreated	ClosestBranch
1	1	john123	johsmith@yahoo.com	John Smith	ekscool123	2019-05-31	AucklandPark
2	2	samjack	samuelj@gmail.com	Samuel Jackson	sybiotic3d	2019-05-31	AucklandPark
3	3	FireZ	bill34@hotmail.com	Bill Clinton	hithere45	2019-05-31	AucklandPark
4	4	grimdown	willion@yahoo.com	William Barton	jkl23asd	2019-05-31	AucklandPark
5	5	bug	ivanf@gmail.com	Ivan Ivanhof	sdaf68sdsd	2019-05-31	AucklandPark
6	6	cardiff	benl@hotmail.com	Ben Lovette	gvhyg88	2019-05-31	AucklandPark
7	7	waggler	narekaxuf-9517@yopmail.com	Nadia Dean	bhjf789dd	2019-05-31	AucklandPark
8	8	beese	jsmith@icloud.com	Jeniane Summers	advsw2134	2019-05-31	AucklandPark
9	9	grobble	rupak@me.com	Celine Dion	sdfj789	2019-05-31	AucklandPark
10	10	chissing	storem@att.net	Matt Damon	jnvgsdkf78999	2019-05-31	AucklandPark
11	11	itscoolguy123	coolguy@hotmail.com	Ben Shapiro	bnk6789	2019-06-08	AucklandPark

UserName was changed.

EditStaff

```

EXEC AucklandPark.EditStaff 11, 'Max', 'Steel', 'AucklandPark', '0834261593',
'Cashier', 'M', '1990-07-17', 7000;
GO
SELECT * FROM AucklandPark.Staff;
GO

```

```

53 --Testing EditStaff
54 EXEC AucklandPark.EditStaff 11, 'Max', 'Steel', 'AucklandPark', '0834261593', 'Cashier', 'M', '1990-07-17', 7000;
55 GO

```

100 %

Messages

(1 row affected)

```
56 SELECT * FROM AucklandPark.Staff;
57 GO
```

100 %

Results Messages

	EmpNo	FirstName	LastName	Branch	PhoneNo	HireDate	JobDescription	Gender	Birthdate	Salary
1	1	John	Smith	Bloemfontein	0768550187	2019-05-31	Cashier	M	1996-06-05	5000.00
2	2	Sally	Wickham	Bloemfontein	0728560109	2019-05-31	Accountant	F	1994-04-08	6500.00
3	3	Angelina	Jolie	Bloemfontein	0837530142	2019-05-31	Sales	F	1995-11-28	6000.00
4	4	Brad	Pritt	Bloemfontein	0821450199	2019-05-31	Stocks	M	1992-02-04	7000.00
5	5	Tally	Ho	Bloemfontein	0844560287	2019-05-31	Cashier	M	1990-07-15	5000.00
6	6	John	Miller	Bloemfontein	0749500151	2019-05-31	Accountant	M	1995-04-12	7000.00
7	7	Ryder	Orozco	Bloemfontein	0768550187	2019-05-31	Sales	M	1991-09-23	5000.00
8	8	Nakita	Lee	Bloemfontein	0818544512	2019-05-31	Stocks	F	1992-10-09	6000.00
9	9	Jess	Beattie	Bloemfontein	0826620212	2019-05-31	Cashier	F	1993-03-21	5000.00
10	10	Christina	Conroy	Bloemfontein	0836520855	2019-05-31	Accountant	F	1996-08-11	700000.00
11	11	Max	Steel	AucklandPark	0834261593	2019-06-08	Cashier	M	1990-07-17	7000.00

PhoneNo was changed.

EditComic

```
DECLARE @page varbinary(max)
SET @page = (SELECT * FROM OPENROWSET(BULK N'A:\Documents\CTU Training\MCSE\Semester
1\Summatives\DBD621\Coverpages\sonic.jpg',SINGLE_BLOB) AS img)
EXEC AucklandPark.EditComic 14,
    'Sonic the Hedgehog',
    'Ian Flynn',
    'The Fate of Dr. Eggman, Part 1. After his last battle
with Sonic, Dr. Eggman's gone quiet. His Badnik forces are still causing trouble, but
without the bad doctor's usual fanfare. What's Eggman been doing all this time? And
what will happen to Sonic when he finds out?',
    6,
    'IDW Publishing',
    @page;
GO
SELECT * FROM AucklandPark.Comics;
GO
```

```

59  --Testing EditComics
60  DECLARE @page varbinary(max)
61  SET @page = (SELECT * FROM OPENROWSET(BULK N
62  EXEC AucklandPark.EditComic 14,
63      'Sonic the Hedge
64      'Ian Flynn',
65      'The Fate of Dr.
66      6,
67      'IDW Publishing'
68      @page;
69  GO

```

100 %

Messages

(1 row affected)

```

70  SELECT * FROM AucklandPark.Comics;
71  GO

```

100 %

Results Messages

	ComicId	Title	Author	Description	Issue	Publisher	CoverPage
1	2	Ultimate Spider-Man	Mark Bagley	Death of Spider-Man	156	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
2	3	Ultimate X-Men	Aron Coleite	Ultimatum	99	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
3	4	Dune	Ralph Macchio	Comic adpatation of the 1984 film	2	Marvel	0xFFD8FFE000104A46494600010101004800480000FFFE001...
4	5	Arkham Knight	Peter Tomasi	Prequel to the Arkham Knight video game	2	DC Comics	0xFFD8FFE000104A46494600010101012C012C0000FFFE222...
5	6	Elektra (2017)	Matthew Owens	Running With The Devil	4	Marvel	0xFFD8FFE123A345786966000049492A000800000000E0000...
6	7	Exiles	Saladin Ahmed	THE WRATH OF KHAN! The Exiles' beloved Khan—who ...	11	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
7	8	Falcon	Rodney Barnes	As Brooklyn becomes vampire central, Deacon Frost conti...	7	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
8	9	Fantastic Five	Tom DeFalco	Though his metal implants have been destroyed, a severl...	4	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
9	10	Foolkiller	Max Bemis	Foolkiller versus a bunch of skinheads!! Alone with an ars...	2	Marvel	0xFFD8FFE1260245786966000049492A000800000000E00000...
10	11	Generation X	Christina Strain	It's JUBILEE vs. M-PLATE..vampire vs. vampire! Will the ...	87	Marvel	0xFFD8FFE1291945786966000049492A000800000000E00000...
11	14	Sonic the Hedgehog	Ian Flynn	The Fate of Dr. Eggman, Part 1. After his last battle with S...	6	IDW Publishing	0xFFD8FFE000104A46494600010100004800480000FFFE1004...

Issue was changed.

Delete stored procedures and triggers

In this section I will be testing the DELETE triggers as well as the stored procedures.

```
USE ComicsDB
GO
```

```
--Deleting data
```

```
CREATE PROCEDURE AucklandPark.DeleteUser
    @UserId int
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
    BEGIN TRY
        DELETE FROM AucklandPark.Users WHERE UserID = @UserId;
    END TRY
    BEGIN CATCH
        SELECT @ERR_MSG = ERROR_MESSAGE(),
               @ERR_STA = ERROR_STATE()

        SET @ERR_MSG = 'Error occured while deleting the record: ' + @ERR_MSG;

        THROW 50001, @ERR_MSG, @ERR_STA;
    END CATCH
```

```
GO
```

```
CREATE PROCEDURE AucklandPark.DeleteStaff
    @EmpNo int
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
    BEGIN TRY
        DELETE FROM AucklandPark.Staff WHERE EmpNo = @EmpNo;
    END TRY
    BEGIN CATCH
        SELECT @ERR_MSG = ERROR_MESSAGE(),
               @ERR_STA = ERROR_STATE()

        SET @ERR_MSG = 'Error occured while deleting the record: ' + @ERR_MSG;

        THROW 50001, @ERR_MSG, @ERR_STA;
    END CATCH
```

```
GO
```

```
CREATE PROCEDURE AucklandPark.DeleteComic
    @ComicId int
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
    BEGIN TRY
        DELETE FROM AucklandPark.Comics WHERE ComicId = @ComicId;
    END TRY
    BEGIN CATCH
        SELECT @ERR_MSG = ERROR_MESSAGE(),
               @ERR_STA = ERROR_STATE()

        SET @ERR_MSG = 'Error occured while deleting the record: ' + @ERR_MSG;

        THROW 50001, @ERR_MSG, @ERR_STA;
    END CATCH
```

```
GO
```

```
CREATE PROCEDURE AucklandPark.DeleteSale
    @SaleId int
AS
    DECLARE @ERR_MSG AS NVARCHAR(4000), @ERR_STA AS SMALLINT
```

```

BEGIN TRY
    DELETE FROM AucklandPark.Sales WHERE SaleId = @SaleId;
END TRY
BEGIN CATCH
    SELECT @ERR_MSG = ERROR_MESSAGE(),
    @ERR_STA = ERROR_STATE()

    SET @ERR_MSG = 'Error occurred while deleting the record: ' + @ERR_MSG;

    THROW 50001, @ERR_MSG, @ERR_STA;
END CATCH
GO

```

History tables

```

USE ComicsDB
GO

```

```

CREATE TABLE AucklandPark.HistoryComics(
    Id INT IDENTITY PRIMARY KEY,
    ComicId INT,
    Title NVARCHAR(50),
    Author NVARCHAR(50),
    [Description] VARCHAR(MAX),
    Issue INT,
    Publisher VARCHAR(20),
    CoverPage VARBINARY(MAX),
    DateDeleted DATETIME2(7) DEFAULT SYSDATETIME(),
    DeletedBy NVARCHAR(128) DEFAULT CURRENT_USER
);
GO

```

```

CREATE TABLE AucklandPark.HistorySales(
    Id INT IDENTITY PRIMARY KEY,
    SaleId INT,
    ComicId INT,
    SalesDate DATE,
    BeforeTax MONEY,
    AfterTax MONEY,
    Quantity TINYINT,
    DateDeleted DATETIME2(7) DEFAULT SYSDATETIME(),
    DeletedBy NVARCHAR(128) DEFAULT CURRENT_USER
);
GO

```

```

CREATE TABLE AucklandPark.HistoryStaff(
    Id INT IDENTITY PRIMARY KEY,
    EmpNo INT,
    FirstName NVARCHAR(15),
    LastName NVARCHAR(20),
    Branch VARCHAR(20),
    PhoneNo CHAR(10),
    HireDate DATE,
    JobDescription VARCHAR(50),
    Gender CHAR(1),
    Birthdate DATE,
    Salary MONEY,
    DateDeleted DATETIME2(7) DEFAULT SYSDATETIME(),
    DeletedBy NVARCHAR(128) DEFAULT CURRENT_USER
);
GO

```

```

CREATE TABLE AucklandPark.HistoryUsers(

```



```

        Id            INT IDENTITY PRIMARY KEY,
        UserID        INT,
        UserName      VARCHAR(15),
        EmailAddress  NVARCHAR(50),
        UserFullName  NVARCHAR(25),
        [Password]    VARCHAR(15),
        DateCreated   DATE,
        ClosestBranch VARCHAR(20),
        DateDeleted   DATETIME2(7) DEFAULT SYSDATETIME(),
        DeletedBy     NVARCHAR(128) DEFAULT CURRENT_USER
    );
GO

```

Triggers

```

USE ComicsDB
GO

```

```

CREATE TRIGGER AucklandPark.comics_history
ON AucklandPark.Comics
AFTER DELETE
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO
        AucklandPark.HistoryComics(
            ComicId,
            Title,
            Author,
            [Description],
            Issue,
            Publisher,
            CoverPage
        )
    SELECT
        d.ComicId,
        Title,
        Author,
        [Description],
        Issue,
        Publisher,
        CoverPage
    FROM deleted AS d;
END
GO

```

```

CREATE TRIGGER AucklandPark.users_history
ON AucklandPark.Users
AFTER DELETE
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO
        AucklandPark.HistoryUsers(
            UserID,
            UserName,
            EmailAddress,
            UserFullName,
            [Password],
            DateCreated,
            ClosestBranch
        )
    SELECT

```

```

        d.UserID,
        UserName,
        EmailAddress,
        UserFullName,
        [Password],
        DateCreated,
        ClosestBranch
    FROM deleted AS d;
END
GO

CREATE TRIGGER AucklandPark.staff_history
ON AucklandPark.Staff
AFTER DELETE
AS
BEGIN
    INSERT INTO
        AucklandPark.HistoryStaff(
            EmpNo,
            FirstName,
            LastName,
            Branch,
            PhoneNo,
            HireDate,
            JobDescription,
            Gender,
            Birthdate,
            Salary
        )
    SELECT
        d.EmpNo,
        d.FirstName,
        d.LastName,
        d.Branch,
        d.PhoneNo,
        d.HireDate,
        d.JobDescription,
        d.Gender,
        d.Birthdate,
        d.Salary
    FROM deleted AS d;
END
GO

CREATE TRIGGER AucklandPark.sales_history
ON AucklandPark.Sales
AFTER DELETE
AS
BEGIN
    INSERT INTO AucklandPark.HistorySales(
        SaleId,
        ComicId,
        SalesDate,
        BeforeTax,
        AfterTax,
        Quantity
    )
    SELECT
        d.SaleId,
        d.ComicId,
        d.SalesDate,
        d.BeforeTax,

```

```

        AfterTax,
        Quantity
    FROM deleted AS d;
END
GO

Testing stored procedures and triggers

DeleteUser
EXEC AucklandPark.DeleteUser 11;
GO
SELECT * FROM AucklandPark.Users;
SELECT * From AucklandPark.HistoryUsers;

```

```

73  --Testing DeleteUser
74  EXEC AucklandPark.DeleteUser 11;
75  GO

```

100 %

Messages

(1 row affected)

	UserID	UserName	EmailAddress	UserFullName	Password	DateCreated	ClosestBranch
1	1	john123	johsmith@yahoo.com	John Smith	ekscool123	2019-05-31	AucklandPark
2	2	samjack	samuelj@gmail.com	Samuel Jackson	sybmbiotic3d	2019-05-31	AucklandPark
3	3	FireZ	bill34@hotmail.com	Bill Clinton	hithere45	2019-05-31	AucklandPark
4	4	grimdown	willion@yahoo.com	William Barton	jkl23asd	2019-05-31	AucklandPark
5	5	bug	ivanf@gmail.com	Ivan Ivanhof	sdaf68sdsd	2019-05-31	AucklandPark
6	6	cardiff	benl@hotmail.com	Ben Lovette	gvhyg88	2019-05-31	AucklandPark
7	7	waggler	narekaxuf-9517@yopmail.com	Nadia Dean	bhjf789dd	2019-05-31	AucklandPark
8	8	beese	jsmith@icloud.com	Jeniane Summers	advsw2134	2019-05-31	AucklandPark
9	9	grobble	rupak@me.com	Celine Dion	sdfj789	2019-05-31	AucklandPark
10	10	chissing	storem@att.net	Matt Damon	jnvgsdkf78999	2019-05-31	AucklandPark

```
77 | SELECT * From AucklandPark.HistoryUsers;
```

Results										
	Id	UserID	UserName	EmailAddress	UserFullName	Password	DateCreated	ClosestBranch	DateDeleted	DeletedBy
1	1	11	itscoolguy123	coolguy@hotmail.com	Ben Shapiro	bnk6789	2019-06-08	AucklandPark	2019-06-08 13:44:07.1320322	dbo

DeleteStaff

```
EXEC AucklandPark.DeleteStaff 13;
```

```
GO
```

```
SELECT * FROM AucklandPark.HistoryStaff;
```

```
SELECT * FROM AucklandPark.Staff;
```

```
79 | --Testing DeleteStaff
```

```
80 | EXEC AucklandPark.DeleteStaff 13;
```

```
81 | GO
```

```
82 | SELECT * FROM AucklandPark.HistoryStaff;
```

```
83 | SELECT * FROM AucklandPark.Staff;
```

Messages										
(1 row affected)										
(1 row affected)										

Results										
	EmpNo	FirstName	LastName	Branch	PhoneNo	HireDate	JobDescription	Gender	Birthdate	Salary
1	1	John	Smith	Bloemfontein	0768550187	2019-05-31	Cashier	M	1996-06-05	5000.00
2	2	Sally	Wickham	Bloemfontein	0728560109	2019-05-31	Accountant	F	1994-04-08	6500.00
3	3	Angelina	Jolie	Bloemfontein	0837530142	2019-05-31	Sales	F	1995-11-28	6000.00
4	4	Brad	Pritt	Bloemfontein	0821450199	2019-05-31	Stocks	M	1992-02-04	7000.00
5	5	Tally	Ho	Bloemfontein	0844560287	2019-05-31	Cashier	M	1990-07-15	5000.00
6	6	John	Miller	Bloemfontein	0749500151	2019-05-31	Accountant	M	1995-04-12	7000.00
7	7	Ryder	Orozco	Bloemfontein	0768550187	2019-05-31	Sales	M	1991-09-23	5000.00
8	8	Nakita	Lee	Bloemfontein	0818544512	2019-05-31	Stocks	F	1992-10-09	6000.00
9	9	Jess	Beattie	Bloemfontein	0826620212	2019-05-31	Cashier	F	1993-03-21	5000.00
10	10	Christina	Conroy	Bloemfontein	0836520855	2019-05-31	Accountant	F	1996-08-11	700000.00

History table

	Id	EmpNo	FirstName	LastName	Branch	PhoneNo	HireDate	JobDescription	Gender	Birthdate	Salary	DateDeleted	DeletedBy
1	1	12	Max	Steel	Auckland Park	0834560012	2019-06-08	Cashier	M	1990-07-17	7000.00	2019-06-08 13:48:16.3675406	dbo
2	2	13	Max	Steel	Auckland Park	0834560012	2019-06-08	Cashier	M	1990-07-17	7000.00	2019-06-08 16:33:32.8287009	dbo

DeleteComic

```
EXEC AucklandPark.DeleteComic 14;
GO
SELECT * FROM AucklandPark.Comics;
SELECT * FROM AucklandPark.HistoryComics;
```

00 %

```

86 EXEC AucklandPark.DeleteComic 14;
87 GO
88 SELECT * FROM AucklandPark.Comics;
89 SELECT * FROM AucklandPark.HistoryComics;
```

(0 rows affected)

Msg 50001, Level 16, State 0, Procedure AucklandPark.DeleteComic, Line 15 [Batch Start Line 85]
Error occurred while deleting the record: The DELETE statement conflicted with the REFERENCE constraint "FK_Sales_ComicId_0662F0A3". The conflict occurred in database 'AucklandPark'.

An error occurred here, which shows that the TRY CATCH statement works. The problem was solved by removing the foreign key constraint.

100 %

```

86 EXEC AucklandPark.DeleteComic 14;
87 GO
88 SELECT * FROM AucklandPark.Comics;
89 SELECT * FROM AucklandPark.HistoryComics;
```

(1 row affected)

	ComicId	Title	Author	Description	Issue	Publisher	CoverPage
1	2	Ultimate Spider-Man	Mark Bagley	Death of Spider-Man	156	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
2	3	Ultimate X-Men	Aron Coleite	Ultimatum	99	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
3	4	Dune	Ralph Macchio	Comic adpatation of the 1984 film	2	Marvel	0xFFD8FFE000104A46494600010101004800480000FFFE001...
4	5	Akham Knight	Peter Tomasi	Prequel to the Akham Knight video game	2	DC Comics	0xFFD8FFE000104A46494600010101012C012C0000FFFE222...
5	6	Elektra (2017)	Matthew Owens	Running With The Devil	4	Marvel	0xFFD8FFE123A345786966000049492A000800000000E0000...
6	7	Exiles	Saladin Ahmed	THE WRATH OF KHAN! The Exiles' beloved Khan—who ...	11	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
7	8	Falcon	Rodney Barnes	As Brooklyn becomes vampire central, Deacon Frost conti...	7	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
8	9	Fantastic Five	Tom DeFalco	Though his metal implants have been destroyed, a severe...	4	Marvel	0xFFD8FFE1001845786966000049492A0008000000000000...
9	10	Foolkiller	Max Bemis	Foolkiller versus a bunch of skinheads!! Alone with an ars...	2	Marvel	0xFFD8FFE1260245786966000049492A000800000000E0000...
10	11	Generation X	Christina Strain	It's JUBILEE vs. M-PLATE...vampire vs. vampire! Will the ...	87	Marvel	0xFFD8FFE1291945786966000049492A000800000000E0000...

History table

	Id	ComicId	Title	Author	Description	Issue	Publisher	CoverPage	DateDeleted	Deleted
1	1	13	Sonic the Hedgehog	Ian Flynn	The Fate of Dr. Eggman, Part 1. After his last b...	5	IDW Publishing	0xFFD8FFE000104A46494600010100004800480000FFE100...	2019-06-06 16:46:43.6492001	dbo
2	2	14	Sonic the Hedgehog	Ian Flynn	The Fate of Dr. Eggman, Part 1. After his last b...	6	IDW Publishing	0xFFD8FFE000104A46494600010100004800480000FFE100...	2019-06-08 13:54:31.9450510	dbo

DeleteSale

EXEC AucklandPark.DeleteSale 13;

GO

SELECT * FROM AucklandPark.Sales;

SELECT * FROM AucklandPark.HistorySales;

```

94 EXEC AucklandPark.DeleteSale 13;
95 GO
96 SELECT * FROM AucklandPark.Sales;
97 SELECT * FROM AucklandPark.HistorySales;
98

```

100 %

	SaleId	ComicId	SalesDate	BeforeTax	AfterTax	Quantity
1	2	2	2019-02-04	54.99	63.238500	1
2	3	2	2019-02-04	54.99	63.238500	1
3	4	3	2019-02-04	54.99	63.238500	1
4	5	5	2019-02-04	54.99	63.238500	1
5	6	4	2019-02-05	54.99	63.238500	1
6	7	6	2019-02-05	54.99	63.238500	1
7	8	9	2019-02-05	54.99	63.238500	1
8	9	10	2019-02-07	54.99	63.238500	1
9	10	8	2019-02-07	54.99	63.238500	1
10	11	7	2019-02-07	54.99	63.238500	1

History table

	Id	SaleId	ComicId	SalesDate	BeforeTax	AfterTax	Quantity	DateDeleted	DeletedBy
1	1	13	14	2019-05-31	54.99	63.2385	2	2019-06-08 15:26:11.0793031	dbo

Conclusion

This concludes the testing manual. After creating the ComicsDB database and implementing its functionality, I proceeded to test the functionality of the database.