# SQL Stored Procedures

## Stored Procedures:

(1) Login

(2) Get students

(3) Create Account

(4) Get Account Details for Main Display

(5) Create Student from new account

(6) Create Account

(7) Edit Account

(8) Deposit Monies

(9) Withdraw Monies

(10) Create transaction

(11) Get Recipient Name

(12) Create Transfer

(13) Get All Transactions Details

(14) Get All Transfer Details per account

(15) Get All Account Details per specified account

### (1) Login

CREATE PROCEDURE uspCheckLogin

@UserName NVARCHAR(40),

@UserPassword NVARCHAR(40),

@UserID INTEGER OUTPUT

AS

SET NOCOUNT ON;

SELECT @UserId = u.UserID

FROM Users u

WHERE

u.UserName = @UserName

AND

u.UserPassword = @UserPassword

GO

### (2) Get students

CREATE PROCEDURE spGetStudents

AS

SELECT \*

FROM Students

### (3) Create Account

CREATE PROCEDURE spCreateAccount

@AccountType NVARCHAR(20),

@SortCode NVARCHAR(10),

@InitialBalance INTEGER,

@CurrentBalance INTEGER,

@OverdraftLimit INTEGER,

@StudentId INTEGER,

@AccountNum INTEGER OUTPUT

AS

DECLARE @AcNoInserted TABLE

(

AccIdValue INTEGER

);

BEGIN

SET NOCOUNT ON

INSERT INTO Accounts

(AccountType, SortCode, InitialBalance, CurrentBalance, OverdraftLimit, StudentID)

OUTPUT inserted.AccountNumber INTO @AcNoInserted(AccIdValue)

VALUES

(@AccountType, @SortCode, @InitialBalance, @CurrentBalance, @OverdraftLimit, @StudentId)

SELECT

@AccountNum = AccIdValue FROM @AcNoInserted

END

### (4) Get Account Details for Main Display

CREATE PROCEDURE spGetAccountDetails

AS

SELECT \*

FROM Accounts

Test:

EXEC spGetAccountDetails

### (5) Create Student from new account

CREATE PROCEDURE spCreateStudent

@FirstName NVARCHAR(50),

@LastName NVARCHAR(50),

@Email NVARCHAR(80),

@PhoneNumber NVARCHAR(30),

@Address1 NVARCHAR(50),

@Address2 NVARCHAR(50),

@City NVARCHAR(50),

@County NVARCHAR(50),

@StudentId INTEGER OUTPUT

AS

DECLARE @StudIdInserted TABLE

(

StudIdValue INTEGER

);

BEGIN

SET NOCOUNT ON

INSERT INTO Students

(FirstName, LastName, Email, PhoneNumber, Address1, Address2, City, County)

OUTPUT inserted.StudentID INTO @StudIdInserted(StudIdValue)

VALUES

(@FirstName, @LastName, @Email, @PhoneNumber, @Address1, @Address2, @City, @County)

SELECT

@StudentId = StudIdValue FROM @StudIdInserted

END

### (6) Create Account

CREATE PROCEDURE spCreateAccount

@AccountType NVARCHAR(20),

@SortCode NVARCHAR(10),

@InitialBalance INTEGER,

@OverdraftLimit INTEGER,

@StudentId INTEGER,

@AccountNum INTEGER OUTPUT

AS

DECLARE @AcNoInserted TABLE

(

AccIdValue INTEGER

);

BEGIN

SET NOCOUNT ON

INSERT INTO Accounts

(AccountType, SortCode, InitialBalance, OverdraftLimit, StudentID)

OUTPUT inserted.AccountNumber INTO @AcNoInserted(AccIdValue)

VALUES

(@AccountType, @SortCode, @InitialBalance, @OverdraftLimit, @StudentId)

SELECT

@AccountNum = AccIdValue FROM @AcNoInserted

END

### (7) Edit Account

CREATE PROCEDURE [dbo].[spEditStudentDetails]

@email NVARCHAR(80),

@phone NVARCHAR(30),

@add1 NVARCHAR(50),

@add2 NVARCHAR(50),

@city NVARCHAR(50),

@county NVARCHAR(50),

@studID INTEGER

AS

BEGIN

UPDATE Students

SET

Email = @email,

PhoneNumber = @phone,

Address1 = @add1,

Address2 = @add2,

City = @city,

County = @county

WHERE

StudentID = @studID

END

### (8) Deposit Monies

CREATE PROCEDURE [dbo].[spDeposit]

@newbalance INTEGER,

@accNum INTEGER,

@accType NVARCHAR(20)

AS

BEGIN

UPDATE Accounts

SET

CurrentBalance = @newbalance

WHERE

AccountNumber = @accNum

AND

AccountType = @accType

END

### (9) Withdraw Monies

CREATE PROCEDURE [dbo].[spWithdraw]

@newbalance INTEGER,

@accNum INTEGER,

@accType NVARCHAR(20)

AS

BEGIN

UPDATE Accounts

SET

CurrentBalance = @newbalance

WHERE

AccountNumber = @accNum

AND

AccountType = @accType

END

### (10) Create transaction

CREATE PROCEDURE spCreateTransaction

@TransType NVARCHAR (50),

@AccNo INTEGER,

@Date DATETIME,

@Ammount INTEGER,

@Balance INTEGER,

@TransactionID INTEGER OUTPUT

AS

DECLARE @TransactinoCreated TABLE

(

TransactionIdValue INTEGER

);

BEGIN

SET NOCOUNT ON

INSERT INTO Transactions

(TransactionType, AccountNumber, DateTimeOfTransaction, TransactionAmmount, Balance)

OUTPUT inserted.TransactionNumber INTO @TransactinoCreated(TransactionIdValue)

VALUES

(@TransType, @AccNo, @Date, @Ammount, @Balance)

SELECT

@TransactionID = TransactionIdValue FROM @TransactinoCreated

END

### (11) Get Recipient Name

CREATE PROCEDURE spGetStudentName

@StudentID INTEGER

AS

BEGIN

SET NOCOUNT ON

SELECT s.FirstName, s.LastName

FROM Students s

WHERE s.StudentID = @StudentID

END

### (12) Create Transfer

CREATE PROCEDURE spCreateTransferRecord

@RecName VARCHAR(100),

@DebitedAcc INTEGER,

@TransactionNum INTEGER,

@DesctinationSCode VARCHAR(10),

@DestinationAccNo INTEGER,

@TransferAmmount INTEGER,

@Date DATETIME,

@Description NVARCHAR(200)

AS

BEGIN

SET NOCOUNT ON

INSERT INTO Transfers

(RecepientName, DebitedAccount, TransactionNumber, DestinationSortCode, DestinationAccountNumber, TransferAmmount, DateTimeOfTransfer, TransferDescription)

VALUES (@RecName, @DebitedAcc, @TransactionNum, @DesctinationSCode, @DestinationAccNo, @TransferAmmount, @Date, @Description)

END

### (13) Get All Transactions Details

CREATE PROCEDURE spGetTransactionsDetails

AS

SELECT \*

FROM Transactions

Test:

EXEC spGetTransactionsDetails

### (14) Get All Transfer Details per account

CREATE PROCEDURE spGetTransfers

AS

BEGIN

SELECT t1.AccountNumber, t1.TransactionNumber, t1.TransactionType, t1.DateTimeOfTransaction, t1.TransactionAmmount, t2.DestinationSortCode, t2.DestinationAccountNumber, t2.TransferDescription

FROM Transactions AS t1

INNER JOIN Transfers AS t2

ON t1.TransactionNumber = t2.TransactionNumber

END

### (15) Get All Account Details per specified account

CREATE PROCEDURE spGetAccount

@accNo INTEGER

AS

BEGIN

SET NOCOUNT ON

SELECT \*

FROM Accounts

WHERE Accounts.AccountNumber = @accNo

END