-CREATE DB

CREATE DATABASE MS\_DRINK\_DB

USE MS\_DRINK\_DB

--CREATE TABLES

CREATE TABLE TEAM (

TeamID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

TeamName VARCHAR(30) NOT NULL,

TeamDesc VARCHAR(100) DEFAULT NULL

)

CREATE TABLE CAMPUS (

CampusID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

CampusName VARCHAR(30) NOT NULL,

CampusDesc VARCHAR(100) DEFAULT NULL

)

CREATE TABLE FRIDGE\_TYPE (

FridgeTypeID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

FridgeTypeName VARCHAR(30) NOT NULL,

FridgeTypeDesc VARCHAR(100) DEFAULT NULL

)

CREATE TABLE LOCATION (

LocationID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

LocationName VARCHAR(30) NOT NULL,

LocationDesc VARCHAR(100) DEFAULT NULL

)

CREATE TABLE DRINK (

DrinkID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

DrinkName VARCHAR(30) NOT NULL,

DrinkDesc VARCHAR(100) DEFAULT NULL

)

CREATE TABLE EVENT\_TYPE (

EventTypeID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

EventTypeName VARCHAR(30) NOT NULL,

EventTypeDesc VARCHAR(100) DEFAULT NULL

)

CREATE TABLE POSITION (

PositionID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

PositionName VARCHAR(30) NOT NULL,

PositionDesc VARCHAR(100) DEFAULT NULL

)

CREATE TABLE PERSON (

PersonID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

PersonFname VARCHAR(30) NOT NULL,

PersonLname VARCHAR(30) NOT NULL,

PersonDOB DATE NOT NULL

)

CREATE TABLE TEAM\_PERSON (

TeamPersonID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

TeamID INT NOT NULL FOREIGN KEY REFERENCES TEAM(TeamID),

PersonID INT NOT NULL FOREIGN KEY REFERENCES PERSON(PersonID),

PositionID INT NOT NULL FOREIGN KEY REFERENCES POSITION(PositionID),

BeginDate DATE NOT NULL,

EndDate DATE DEFAULT NULL

)

CREATE TABLE BUILDING (

BuildingID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

CampusID INT FOREIGN KEY REFERENCES CAMPUS(CampusID),

BuildingName VARCHAR(30) NOT NULL,

BuildingDesc VARCHAR(100) DEFAULT NULL

)

CREATE TABLE FRIDGE (

FridgeID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

FridgeName VARCHAR(30) NOT NULL,

FridgeTypeID INT NOT NULL FOREIGN KEY REFERENCES FRIDGE\_TYPE(FridgeTypeID),

BuildingID INT NOT NULL FOREIGN KEY REFERENCES BUILDING(BuildingID),

LocationID INT NOT NULL FOREIGN KEY REFERENCES LOCATION(LocationID)

)

CREATE TABLE EVENT (

EventID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

EventTypeID INT NOT NULL FOREIGN KEY REFERENCES EVENT\_TYPE(EventTypeID),

DrinkID INT NOT NULL FOREIGN KEY REFERENCES DRINK(DrinkID),

Quantity INT NOT NULL,

EventTime DATETIME NOT NULL,

FridgeID INT NOT NULL FOREIGN KEY REFERENCES FRIDGE(FridgeID),

TeamPersonID INT NOT NULL FOREIGN KEY REFERENCES TEAM\_PERSON(TeamPersonID)

)

--STORED PROCS

--new team

CREATE PROC uspNewTeam

@TeamName VARCHAR

AS

BEGIN

BEGIN TRANSACTION

INSERT INTO TEAM (TeamName)

VALUES (@CampusName)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new campus

CREATE PROC uspNewCampus

@CampusName VARCHAR

AS

BEGIN

BEGIN TRANSACTION

INSERT INTO CAMPUS (CampusName)

VALUES (@CampusName)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new building

CREATE PROC uspNewBuilding

@BuildingName VARCHAR,

@CampusName VARCHAR

AS

BEGIN

DECLARE @CampusID INT

SET @CampusID = (

SELECT CampusID FROM CAMPUS

WHERE CampusName = @CampusName)

IF @campusID IS NULL

RAISERROR('CampusID Does not exist', 11, 1)

BEGIN TRANSACTION

INSERT INTO BUILDING (CampusID, BuildingName)

VALUES (@CampusID, @BuildingName)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new fridge

CREATE PROC uspNewFridge

@FridgeName VARCHAR,

@FridgeTypeName VARCHAR,

@BuildingName VARCHAR,

@LocationName VARCHAR

AS

BEGIN

DECLARE @FridgeTypeID INT

DECLARE @BuildingID INT

DECLARE @LocationID INT

SET @FridgeTypeID = (

SELECT FridgeTypeID FROM FRIDGE\_TYPE

WHERE FridgeTypeName = @FridgeTypeName)

IF @FridgeTypeID IS NULL

RAISERROR('FridgeTypeID Does not exist', 11, 1)

SET @BuildingID = (

SELECT BuildingID FROM BUILDING

WHERE BuildingName = @BuildingName)

IF @BuildingID IS NULL

RAISERROR('BuildingID does not exist', 11, 1)

SET @LocationID = (

SELECT LocationID FROM LOCATION

WHERE LocationName = @LocationName)

IF @LocationID IS NULL

RAISERROR('locationID does not exist', 11, 1)

BEGIN TRANSACTION

INSERT INTO FRIDGE (FridgeName, FridgeTypeID, BuildingID, LocationID)

VALUES (@FridgeNAme, @FridgeTypeID, @BuildingID, @LocationID)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new fridge type

CREATE PROC uspNewFridgeType

@FridgeTypeName VARCHAR

AS

BEGIN

BEGIN TRANSACTION

INSERT INTO FRIDGE\_TYPE (FridgeTypeName)

VALUES (@FridgeTypeName)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new location

CREATE PROC uspNewLocation

@LocationName VARCHAR

AS

BEGIN

BEGIN TRANSACTION

INSERT INTO LOCATION (LocationName)

VALUES (@LocationName)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new event type

CREATE PROC uspNewEventType

@EventTypeName VARCHAR

AS

BEGIN

BEGIN TRANSACTION

INSERT INTO EVENT\_TYPE(EventTypeName)

VALUES (@EventTypeName)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new drink

CREATE PROC uspNewDrink

@DrinkName VARCHAR

AS

BEGIN

BEGIN TRANSACTION

INSERT INTO DRINK (DrinkName)

VALUES (@DrinkName)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new position

CREATE PROC uspNewPosition

@PositionName VARCHAR

AS

BEGIN

BEGIN TRANSACTION

INSERT INTO POSITION (PositionName)

VALUES (@PositionName)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new person

CREATE PROC uspNewPerson

@FName VARCHAR,

@Lname VARCHAR,

@DOB DATE

AS

BEGIN

BEGIN TRANSACTION

INSERT INTO PERSON (PersonFname, PersonLname, PersonDOB)

VALUES (@FName, @Lname, @DOB)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--new TeamPerson

CREATE PROC uspNewTeamPerson

@BeginDate DATE,

@FName VARCHAR,

@Lname VARCHAR,

@DOB DATE,

@PositionName VARCHAR,

@TeamName VARCHAR

AS

BEGIN

DECLARE @PersonID INT

DECLARE @PositionID INT

DECLARE @TeamID INT

SET @PersonID = (

SELECT PersonID FROM PERSON

WHERE PersonFname = @Fname

AND PersonLname = @Lname

AND PersonDOB = @DOB)

IF @PersonID IS NULL

RAISERROR('personID does not exist', 11, 1)

SET @PositionID = (

SELECT PositionID FROM POSITION

WHERE PositionName = @PositionName)

IF @PositionID IS NULL

RAISERROR('positionID does not exist', 11, 1)

SET @TeamID = (

SELECT TeamID FROM TEAM

WHERE TeamName = @TeamName)

IF @TeamID IS NULL

RAISERROR('teamID does not exist', 11, 1)

BEGIN TRANSACTION

INSERT INTO TEAM\_PERSON (TeamID, PersonID, PositionID, BeginDate)

VALUES (@TeamID, @PersonID, @PositionID, @BeginDate)

IF @@ERROR <> 0

ROLLBACK TRAN

ELSE

COMMIT TRAN

END

--populate the Person table with the given number of people

CREATE PROC uspPopPerson

@Run INT

AS

DECLARE @FnameID INT

DECLARE @LnameID INT

DECLARE @Fname VARCHAR(30)

DECLARE @Lname VARCHAR(30)

DECLARE @DOB DATE

DECLARE @Dig1 varChar (20)

DECLARE @Rand Numeric (16,16)

WHILE @Run > 0

BEGIN TRY

SELECT @Rand = RAND()

SET @Dig1 = @Rand

SET @FnameID = (SELECT SUBSTRING(@Dig1, 3, 4)) % 20000

SET @LnameID = (SELECT SUBSTRING(@Dig1, 6, 5)) % 177597

SET @Fname = (

SELECT FirstName FROM CUSTOMER\_BUILD.dbo.tblFIRST\_NAME

WHERE FirstNameID = @FnameID)

SET @Lname = (

SELECT LastName FROM CUSTOMER\_BUILD.dbo.tblLAST\_NAME

WHERE LastNameID = @LnameID)

SET @DOB = (SELECT CAST(DateAdd(d, ROUND(DateDiff(d, '1950-01-01', '1999-12-31') \* RAND(), 0), '1950-01-01') as date))

EXEC uspNewPerson @Fname, @Lname, @DOB

SET @Run = @Run - 1

END TRY

BEGIN CATCH

END CATCH

--populate the team table with the given number of items

ALTER PROC uspPopTeam

@Run int

AS

DECLARE @TeamNameID INT

DECLARE @TeamName VARCHAR(30)

DECLARE @Dig1 varChar (20)

DECLARE @Rand Numeric (16,16)

WHILE @Run > 0

BEGIN TRY

SELECT @Rand = RAND()

SET @Dig1 = @Rand

SET @TeamNameID = (SELECT SUBSTRING(@Dig1, 3, 4)) % 20000

SET @TeamName = (

SELECT Zip FROM CUSTOMER\_BUILD.dbo.tblCITY\_STATE\_ZIP

WHERE CityStateZipID = @TeamNameID)

EXEC uspNewTeam @TeamName

SET @Run = @Run - 1

END TRY

BEGIN CATCH

END CATCH

--populate the campus table with the given number of rows

CREATE PROC uspPopCampus

@Run int

AS

DECLARE @CampusNameID INT

DECLARE @CampusName VARCHAR(30)

DECLARE @Dig1 varChar (20)

DECLARE @Rand Numeric (16,16)

WHILE @Run > 0

BEGIN TRY

SELECT @Rand = RAND()

SET @Dig1 = @Rand

SET @CampusNameID = (SELECT SUBSTRING(@Dig1, 3, 4)) % 20000

SET @CampusName = (

SELECT CityName FROM CUSTOMER\_BUILD.dbo.tblCITY\_STATE\_ZIP

WHERE CityStateZipID = @CampusNameID)

EXEC uspNewCampus @CampusName

SET @Run = @Run - 1

END TRY

BEGIN CATCH

END CATCH

--populate the TEAM\_PERSON table with the given number of entries

--Grabs team and position, creates a new PERSON, then creates a new TEAM\_PERSON