|  |  |
| --- | --- |
|  | **2015** |
|  | UNICORNS OF LOVE LLC.  Jorge Llerna  Paul Molina  Pablo Valencia  Warren Walters |

|  |
| --- |
| **[Hero World Database Proposal]** |
| Created for the Sole Use of Disney for the Administration of their Upcoming Theme Park |

TABLE OF CONTENTS

Database Diagram and Description....................................................................................... 3

Tables.................................................................................................................................... 5

Views..................................................................................................................................... 8

Triggers.................................................................................................................................. 10

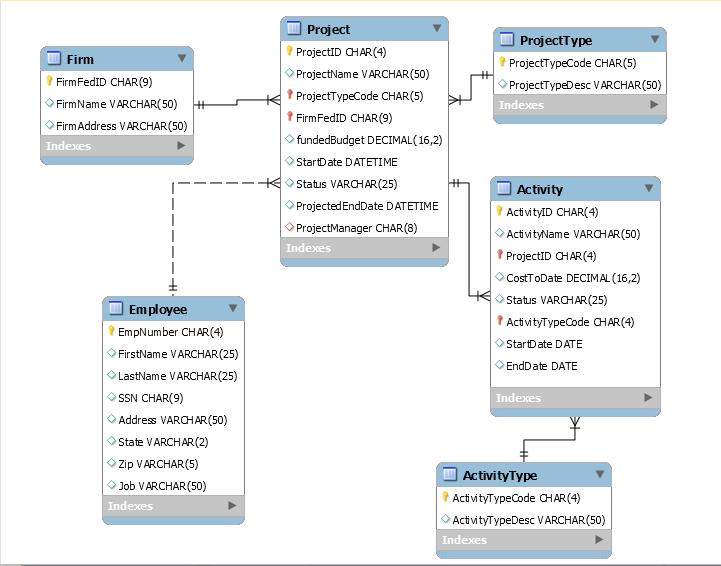
Stored Procedures................................................................................................................. 13

Stored Procedure Test Data.................................................................................................. 18

Data Inserts........................................................................................................................... 20

Extra Credit........................................................................................................................... 24

HERO WORLD DATABASE DIAGRAM



HERO WORLD DATABASE DIAGRAM

**Project** (projectID , projectName ,projectTypeCode ,firmFedID ,fundedBudget ,startDate, status, projectedEndDate ,projectManager)

**FK**  firmFedID → Firm

**FK**  projectTypeCode → Project Type

**SK**  projectName

**Project Type** ( ProjectTypeCode, ProjectTypeDesc)

**AK** ProjectTypeDesc

**Firm** ( firmFedID, firmName, FirmAddress)

**SK** firmName

**SK** FirmAddress

**Activity** (ActivityId, ActivityName, projectID, CostToDate, Status, ActivityTypeCode, StartDate, EndDate)

**FK**  projectID → Project

**FK** ActivityTypeCode → ActivityType

**SK** ActivityName

**ActivityType** ( ActivityTypeCode, ActivityTypeDesc)

**AK** ActivityTypeDesc

**Employee** (empNumber, firstName, lastName, ssn, address, state, zip, job)

**SK** firstName

**AK** ssn

Project Tables:

--Creates Firm Table

CREATE TABLE Firm

(

firmFedID CHAR(9) PRIMARY KEY,

firmName VARCHAR(50),

firmAddress VARCHAR(50),

)

--Creates Employee Table

CREATE TABLE Employee

(

empNumber CHAR(8) PRIMARY KEY,

firstName VARCHAR(25),

lastName VARCHAR(25),

ssn CHAR(9),

address VARCHAR(50),

state VARCHAR(2),

zip VARCHAR(5),

job VARCHAR(50),

CONSTRAINT chk\_EmployeeJob CHECK(job IN('Cast Member', 'Engineer', 'Inspector', 'Project Manager'))

)

--Creates ProjectType Table

CREATE TABLE ProjectType

(

projectTypeCode CHAR(5) PRIMARY KEY,

projectTypeDesc VARCHAR(50)

)

--Creates ActivityType Table

CREATE TABLE ActivityType

(

activityTypeCode CHAR(4) PRIMARY KEY,

activityTypeDesc VARCHAR(50),

CONSTRAINT chk\_ActivityTypeCode CHECK

(activityTypeCode IN ('DE', 'CO', 'WA')),

CONSTRAINT chk\_ActivityTypeDesc CHECK

(activityTypeDesc IN ('DESIGN', 'CONSTRUCTION', 'WARRANTY'))

)

--Create Project Table

CREATE TABLE Project

(

projectID CHAR(4) PRIMARY KEY,

projectName VARCHAR(50),

projectTypeCode CHAR(5),

firmFedID CHAR(9) NOT NULL,

fundedBudget DECIMAL(16,2),

startDate DATE,

status VARCHAR(25),

projectedEndDate DATE,

projectManager CHAR(8),

FOREIGN KEY (projectTypeCode) REFERENCES ProjectType(projectTypeCode),

FOREIGN KEY (firmFedID) REFERENCES Firm(firmFedID),

FOREIGN KEY (projectManager) REFERENCES Employee(empNumber),

CONSTRAINT chk\_ProjectStatus CHECK(status IN('Active', 'Inactive', 'Cancelled', 'On-Hold'))

)

--Creates Activity Table

CREATE TABLE Activity

(

activityID CHAR(4) PRIMARY KEY,

activityName VARCHAR(50),

projectID CHAR(4),

costToDate DECIMAL(16,2),

status VARCHAR(25),

activityTypeCode CHAR(4),

startDate DATE,

endDate DATE,

FOREIGN KEY(projectID) REFERENCES Project(projectID),

FOREIGN KEY(activityTypeCode) REFERENCES ActivityType(activityTypeCode),

CONSTRAINT chk\_ActivityStatus CHECK(status IN('Active', 'Inactive', 'Cancelled', 'On-Hold'))

)

--Creates Project Audit Table

CREATE TABLE ProjectAudit

(

projectID CHAR(4) NOT NULL,

projectName VARCHAR(50),

projectTypeCode CHAR(5),

firmFedID CHAR(9),

startDate DATE,

status VARCHAR(25),

projectedEndDate DATE,

projectManager CHAR(8),

fundedBudget DECIMAL(16,2),

operation VARCHAR(50),

dateTimeCreated smalldatetime,

userName varchar(50)

)

--Creates Activity Audit Table

CREATE TABLE ActivityAudit

(

activityID CHAR(4) NOT NULL,

activityName VARCHAR(50),

projectID CHAR(4),

costToDate DECIMAL(16,2),

status VARCHAR(25),

activityTypeCode CHAR(4),

startDate DATE,

endDate DATE,

operation VARCHAR(50),

dateTimeCreated smalldatetime,

userName varchar(50)

)

--Creates Employee Audit

CREATE TABLE EmployeeAudit

(

empNumber CHAR(8) NOT NULL,

firstName VARCHAR(25),

lastName VARCHAR(25),

ssn CHAR(9),

address VARCHAR(50),

state VARCHAR(2),

zip VARCHAR(5),

job VARCHAR(50),

operation VARCHAR(50),

dateTimeCreated smalldatetime,

userName varchar(50)

)

PROJECT VIEWS:

--View has been made to show absolutely all projects available

--A project might not be fully explained yet, therefore it might not have

---a totally-defined description (though it may have a type).

--This view allows the view of all projects, as long as they have a project ID assigned to them. FINALIZED

CREATE VIEW vw\_Project AS

SELECT projectId, projectName, Project.firmFedID, firmName, firmAddress, fundedbudget, startDate, status, ProjectType.projectTypeCode, projectTypeDesc, projectedEndDate, projectManager

FROM Project

INNER JOIN Firm ON

Project.firmFedID = Firm.firmFedID

LEFT JOIN ProjectType ON

Project.ProjectTypeCode = ProjectType.ProjectTypeCode

--Similar principle as the project view, all activities MAY be defined, but a description may not have been defined for them

--Therefore it is possible to see all current activities, though they may not have yet been described. FINALIZED

CREATE VIEW vw\_Activity AS

SELECT activityId, activityName, projectId, costToDate, status, Activity.activityTypeCode, activityTypeDesc, startDate, endDate FROM Activity

LEFT JOIN ActivityType

ON Activity.ActivityTypeCode = ActivityType.ActivityTypeCode

--Simple view. Extracts all relevant information from the employee table. FINALIZED

CREATE VIEW vw\_Employee AS

SELECT \* FROM Employee

--Shows which projects have activities will go beyond the project's projected end date.

--This view is only able to see projects that are currently active (read: have started and

--have yet to finish). NOTE: This view is designed to work only on projects that have activities

--associated with them. FINALIZED

CREATE VIEW vw\_LateProjects AS

SELECT \* FROM Project

WHERE projectID IN

(SELECT DISTINCT Project.projectID FROM Project

INNER JOIN Activity ON Project.projectID = Activity.projectID

WHERE Activity.endDate > Project.projectedEndDate AND Project.status='Active')

--Shows whichever project has gone over their Firm's funded budget with their activity budget FINALIZED

CREATE VIEW vw\_OverBudget AS

SELECT ProjectName, SUM(costToDate) AS SummarizedActivityCost, FundedBudget FROM

(SELECT Project.projectName, Activity.costToDate, Project.fundedBudget FROM Project

INNER JOIN Activity ON Project.ProjectID = Activity.ProjectID

INNER JOIN Firm ON Firm.FirmFedID = Project.FirmFedID

WHERE Project.status='Active')ActivityAndProjectBudget

GROUP BY ProjectName, fundedBudget

HAVING SUM(costToDate) > fundedBudget

--Shows First Name, Last Name and SSN of Employees who have more than 5 projects FINALIZED

CREATE VIEW vw\_OverAllocatedPM AS

SELECT firstName, lastName, ssn FROM Employee

WHERE empNumber IN

(

SELECT projectManager FROM Project

GROUP BY projectManager

HAVING COUNT(projectManager) > 5

)

--Shows the projects whose activities have proven costly. Each costly activity is shown alongside the project it belongs to. FINALIZED

CREATE VIEW vw\_CostlyActivity AS

SELECT Project.projectID, Project.projectName, Activity.activityName, Activity.costToDate AS ActivityCost FROM Activity

INNER JOIN Project ON Activity.projectID = Project.projectID

WHERE (Activity.status='Cancelled' OR Activity.status='On-Hold') AND Project.status='Active'

--Shows projects which have been completed FINALIZED

CREATE VIEW vw\_CompletedProjects AS

SELECT DISTINCT Project.projectID, Project.projectName, Project.projectTypeCode FROM Project

INNER JOIN Activity ON Project.projectID = Activity.projectID

WHERE Activity.activityTypeCode='WA' AND Activity.endDate IS NOT NULL

PROJECT TRIGGERS:

CREATE TABLE ProjectAudit

(

projectID CHAR(4) PRIMARY KEY,

projectTypeCode CHAR(5),

firmFedID CHAR(9) NOT NULL,

projectName VARCHAR(50),

startDate DATE,

status VARCHAR(25),

projectedEndDate DATE,

projectManager CHAR(8),

fundedBudget DECIMAL(16,2),

operation VARCHAR(50),

dateTimeCreated SMALLDATETIME,

userName VARCHAR(50)

)

GO

CREATE TABLE ActivityAudit

(

activityID CHAR(4) NOT NULL,

activityName VARCHAR(50),

projectID CHAR(4),

costToDate DECIMAL(16,2),

status VARCHAR(25),

activityTypeCode CHAR(4),

startDate DATE,

endDate DATE,

operation VARCHAR(50),

dateTimeCreated SMALLDATETIME,

userName VARCHAR(50)

)

GO

CREATE TABLE EmployeeAudit

(

empNumber CHAR(8) NOT NULL,

firstName VARCHAR(25),

lastName VARCHAR(25),

ssn CHAR(9),

address VARCHAR(50),

state VARCHAR(2),

zip VARCHAR(5),

job VARCHAR(50),

operation VARCHAR(50),

dateTimeCreated SMALLDATETIME,

userName VARCHAR(50)

)

GO

CREATE TRIGGER Trg\_ProjectAudit

ON Project

AFTER INSERT, DELETE

AS

BEGIN

IF EXISTS(SELECT 1 FROM inserted)

BEGIN

INSERT INTO ProjectAudit

(projectID, projectTypeCode,firmFedID,projectName ,startDate ,

status,projectedEndDate,projectManager,fundedBudget,operation,datetimecreated,username )

SELECT projectID, projectTypeCode,firmFedID,projectName,startDate,status,

projectedEndDate,projectManager,fundedBudget,'INSERT', getdate(),System\_user

FROM Inserted

END

if exists(select 1 from deleted)

begin

INSERT INTO ProjectAudit

(projectID, projectTypeCode,firmFedID,projectName ,startDate ,

status,projectedEndDate,projectManager,operation,datetimecreated,username )

SELECT projectID, projectTypeCode,firmFedID,projectName,startDate,status,

projectedEndDate,projectManager,'DELETE', getdate(),System\_user

FROM deleted

END

END

GO

CREATE TRIGGER Trg\_ActivityAudit

ON Activity

AFTER INSERT, DELETE

AS

BEGIN

IF EXISTS(select 1 from inserted)

BEGIN

INSERT INTO ActivityAudit

(activityID,activityName,projectID,costToDate,status,

activityTypeCode,startDate,endDate,operation,datetimecreated,username )

SELECT activityID,activityName,projectID,costToDate,status,activityTypeCode,

startDate,endDate,'INSERT', getdate(),System\_user

FROM Inserted

END

IF EXISTS(select 1 from deleted)

BEGIN

INSERT INTO ActivityAudit

(activityID,activityName,projectID,costToDate,status,

activityTypeCode,startDate,endDate,operation,datetimecreated,username )

SELECT activityID,activityName,projectID,costToDate,status,activityTypeCode,

startDate,endDate,'DELETE', getdate(),System\_user

FROM deleted

END

END

GO

CREATE TRIGGER Trg\_EmployeeAudit

ON Employee

AFTER INSERT, DELETE

AS

BEGIN

IF EXISTS(select 1 from inserted)

BEGIN

INSERT INTO EmployeeAudit

(empNumber,firstName,lastName,ssn,address,

state,zip,job,operation,datetimecreated,username )

SELECT empNumber,firstName,lastName,ssn,address,

state,zip,job,'INSERT', getdate(),System\_user

FROM Inserted

END

IF EXISTS(select 1 from deleted)

BEGIN

INSERT INTO EmployeeAudit

(empNumber,firstName,lastName,ssn,address,

state,zip,job,operation,datetimecreated,username )

SELECT empNumber,firstName,lastName,ssn,address,

state,zip,job,'INSERT', getdate(),System\_user

FROM deleted

END

END

GO

PROJECT STORED PROCEDURES:

-- Project

CREATE PROCEDURE U\_DIS\_AddProject

(

   @projectId CHAR(4),

   @projectName VARCHAR(50),

   @firmFedID CHAR(9),

   @firmName VARCHAR (50),

   @firmAddress VARCHAR (50),

   @fundedBudget DECIMAL (16,2),

   @StartDate DATE,

   @Status VARCHAR(25),

   @projectTypeCode CHAR(5),

   @projectTypeDesc VARCHAR(50),

   @projectedEndDate DATE,

   @ProjectManager CHAR(4)

)

AS BEGIN

   IF EXISTS (SELECT 1 FROM Firm WHERE firmFedID = @firmFedID)

   BEGIN

   PRINT 'Firm already exist. Updating Firm'

   UPDATE Firm

   SET

   firmName = @firmName,

   firmAddress = @firmAddress

   WHERE firmFedId = @firmFedID

   END

   IF EXISTS (SELECT 1 FROM ProjectType WHERE ProjectTypeCode = @projectTypeCode)

   BEGIN

   PRINT 'ProjectType already exist. Updating ProjectType'

   UPDATE ProjectType

   SET

   projectTypeDesc = @projectTypeCode

   WHERE projectTypeCode = @projectTypeDesc

   END

   IF NOT EXISTS (SELECT 1 FROM Firm WHERE firmFedID = @firmFedID)

   BEGIN

   PRINT 'Adding a Firm'

   INSERT INTO Firm

   (FirmFedId, FirmName, FirmAddress)

   VALUES

   (@firmFedID, @firmName, @firmAddress)

   END

   IF NOT EXISTS (SELECT 1 FROM ProjectType WHERE ProjectTypeCode = @projectTypeCode)

   BEGIN

   PRINT 'Adding a projectType'

   INSERT INTO ProjectType

   (projectTypeCode, ProjectTypeDesc)

   VALUES

   (@projectTypeCode, @projectTypeDesc)

   END

PRINT 'Adding a project'

INSERT INTO Project

(ProjectID, ProjectName, ProjectTypeCode, FirmFedID, fundedBudget, StartDate, Status, ProjectedEndDate, ProjectManager)

VALUES

(@projectId, @projectName, @projectTypeCode, @firmFedID, @fundedBudget,  @StartDate, @Status, @projectedEndDate, @ProjectManager)

END

GO

CREATE PROCEDURE U\_DIS\_DeleteProject

(

@projectID CHAR (4)

)

AS BEGIN

PRINT 'Delecting activity that is referencing project'

DELETE FROM Activity WHERE projectID = @projectID

PRINT 'Delectig project'

DELETE FROM Project WHERE projectId = @projectID

END

GO

CREATE PROCEDURE U\_DIS\_UpdateProject

(

   @projectId CHAR(4),

   @projectName VARCHAR(50),

   @firmFedID CHAR(9),

   @firmName VARCHAR (50),

   @firmAddress VARCHAR (50),

   @fundedBudget DECIMAL (16,2),

   @StartDate DATE,

   @Status VARCHAR(25),

   @projectTypeCode CHAR(5),

   @projectTypeDesc VARCHAR(50),

   @projectedEndDate DATE,

   @ProjectManager CHAR(4)

)

AS BEGIN

PRINT 'Updating project'

UPDATE Project

SET

projectTypeCode = @projectTypeCode,

firmFedID = @firmFedID,

projectName = @projectName,

startDate = @StartDate,

status = @Status,

projectedEndDate = @projectedEndDate,

projectManager = @ProjectManager,

fundedBudget = @fundedBudget

WHERE projectID = @projectId

PRINT 'Updating firm'

UPDATE Firm

SET

firmName = @firmName,

firmAddress = @firmAddress

WHERE firmFedId = @firmFedID

PRINT 'Updating project type'

UPDATE ProjectType

SET

projectTypeDesc = @projectTypeCode

WHERE projectTypeCode = @projectTypeCode

END

GO

-- ACTIVITY

CREATE PROCEDURE U\_DIS\_AddActivity

(

@activityID CHAR (4),

@activityName VARCHAR (50),

@projectID CHAR (4),

@costToDate DECIMAL (10,2),

@status VARCHAR(25),

@activityTypeCode CHAR (4),

@activityTypeDesc VARCHAR (50),

@startDate DATE,

@endDate DATE

)

AS BEGIN

   IF NOT EXISTS (SELECT 1 FROM Project WHERE projectID = @projectID)

   BEGIN

   PRINT 'PROJECT DOES NOT EXIST PLEASE ENTER VALID PROJECT'

   RETURN

   END

   IF EXISTS (SELECT 1 FROM ActivityType WHERE activityTypeCode = @activityTypeCode)

   BEGIN

   PRINT 'Updating activity type'

   UPDATE ActivityType

   SET

   activityTypeDesc = @activityTypeDes

   WHERE activityTypeCode = @activityTypeCode

   END

   IF NOT EXISTS (SELECT 1 FROM ActivityType WHERE activityTypeCode = @activityTypeCode)

   BEGIN

   PRINT 'Adding activity type'

   INSERT INTO ActivityType

   (activityTypeCode, activityTypeDesc)

   VALUES

   (@activityTypeCode, @activityTypeDesc)

   END

PRINT 'Adding activity'

INSERT INTO Activity

(activityID, activityName, projectID,  activityTypeCode,  costToDate, status, startDate, endDate)

VALUES

(@activityID, @activityName, @projectID,  @activityTypeCode,  @costToDate, @status, @startDate, @endDate)

END

GO

CREATE PROCEDURE U\_DIS\_DeleteActivity

(

@activityID CHAR (4)

)

AS BEGIN

PRINT 'Delecting activity'

DELETE FROM Activity WHERE activityID = @activityID

END

GO

CREATE PROCEDURE U\_DIS\_UpdateActivity

(

@activityID CHAR (4),

@activityName VARCHAR (50),

@projectID CHAR (4),

@costToDate DECIMAL (10,2),

@status VARCHAR(25),

@activityTypeCode CHAR (4),

@activityTypeDesc VARCHAR (50),

@startDate DATE,

@endDate DATE

)

AS BEGIN

PRINT 'Updating activity'

UPDATE Activity

SET

projectId = @projectID,

activityTypeCode = @activityTypeCode,

activityName = @activityName,

costToDate = @costToDate,

status = @status,

startDate = @startDate,

endDate = @endDate

WHERE activityID = @activityID

PRINT 'Updating activity type'

UPDATE ActivityType

SET

activityTypeDesc = @activityTypeDesc

WHERE activityTypeCode = @activityTypeCode

END

GO

-- Employee

CREATE PROCEDURE U\_DIS\_AddEmployee

(

@empNumber CHAR (8),

@firstName VARCHAR (25),

@lastName VARCHAR (25),

@ssn CHAR (9),

@address VARCHAR (25),

@state CHAR (2),

@zip CHAR (5),

@job VARCHAR (50)

)

AS BEGIN

PRINT 'Adding Employee'

INSERT INTO Employee

(

empNumber, firstName, lastName, ssn, address, state, zip, job

)

VALUES

(

@empNumber, @firstName, @lastName, @ssn, @address, @state, @zip, @job

)

END

GO

CREATE PROCEDURE U\_DIS\_DeleteEmployee

(

@empNumber CHAR (8)

)

AS BEGIN

PRINT 'Deleting employee'

UPDATE Project

SET

projectManager = null

WHERE projectManager = @empNumber

DELETE FROM Employee WHERE empNumber = @empNumber

END

GO

CREATE PROCEDURE U\_DIS\_UpdateEmployee

(

@empNumber CHAR (8),

@firstName VARCHAR (25),

@lastName VARCHAR (25),

@ssn CHAR (9),

@address VARCHAR (25),

@state CHAR (2),

@zip CHAR (5),

@job VARCHAR (50)

)

AS BEGIN

PRINT 'Updating employee'

UPDATE Employee

SET

empNumber = @empNumber,

firstName = @firstName,

lastName = @lastName,

ssn = @ssn,

address = @address,

state = @state,

zip = @zip,

job = @job

WHERE empNumber = @empNumber

END

PROJECT STORED PROCEDURES:

-- SQL SP TEST

--Adding a project

exec U\_DIS\_AddProject @projectId = 4444,

@projectName = 'Happy Land',

@firmFedID = '4444',

@firmName = 'Money Inc',

@firmAddress = '33 NW 123 PL Country Land FL 33313',

@fundedBudget = 12354.2,

@StartDate = '12-12-1932',

@Status = 'Active',

@projectTypeCode = 'FAC',

@projectTypeDesc = 'Facility',

@projectedEndDate = '12-12-1933',

@ProjectManager = null

SELECT \* FROM Project

--Adding an activity if projectid doesnt exist

exec U\_DIS\_AddActivity

@activityID = 1,

@activityName = 'test',

@projectID = 305 ,

@costToDate = 1231.23,

@status = 'Active',

@activityTypeCode = 'DE',

@activityTypeDesc = 'DESIGN',

@startDate = '12-12-1232',

@endDate = '12-12-1233'

--Adding an activity if project does exist

exec U\_DIS\_AddActivity

@activityID = 349,

@activityName = 'test',

@projectID = 4444 ,

@costToDate = 123123,

@status = 'Active',

@activityTypeCode = 'DE',

@activityTypeDesc = 'DESIGN',

@startDate = '12-12-1232',

@endDate = '12-12-1233'

SELECT \* FROM Activity

-- Creating employee

exec U\_DIS\_AddEmployee

@empNumber = 1337,

@firstName = 'Paul',

@lastName = 'Not-In-This-Class',

@ssn = '123-321-123',

@address = 'Under the sea',

@state = 'fl',

@zip = '33323',

@job = 'Engineer'

SELECT \* FROM Employee

--Updating a project with manager that doesnt exist

exec U\_DIS\_UpdateProject @projectId = 4444,

@projectName = 'Looking good',

@firmFedID = '4444',

@firmName = 'Money Bags',

@firmAddress = 'Somewhere over the rainbow',

@fundedBudget = 12354.2,

@StartDate = '12-12-1666',

@Status = 'Active',

@projectTypeCode = 'FOOD',

@projectTypeDesc = 'Good Burger',

@projectedEndDate = '12-12-2012',

@ProjectManager = 5445

--Updating a project with manager that does exist

exec U\_DIS\_UpdateProject @projectId = 4444,

@projectName = 'Looking good',

@firmFedID = '4444',

@firmName = 'Money Bags',

@firmAddress = 'Somewhere over the rainbow',

@fundedBudget = 12354.2,

@StartDate = '12-12-1666',

@Status = 'Active',

@projectTypeCode = 'FOOD',

@projectTypeDesc = 'Good Burger',

@projectedEndDate = '12-12-2012',

@ProjectManager = 1337

SELECT \* FROM Project

--Updating activity

exec U\_DIS\_UpdateActivity

@activityID = 349,

@activityName = 'Merp City',

@projectID = 4444 ,

@costToDate = 321321,

@status = 'Active',

@activityTypeCode = 'DE',

@activityTypeDesc = 'CONSTRUCTION',

@startDate = '4-9-2222',

@endDate = '12-12-2431'

SELECT \* FROM Activity

--Update employee

exec U\_DIS\_UpdateEmployee

@empNumber = 1337,

@firstName = 'TEST',

@lastName = 'TEST',

@ssn = '123-321-123',

@address = 'somewhere',

@state = 'fl',

@zip = '33323',

@job = 'Engineer'

SELECT \* FROM Employee

Data Inserts:

VALUES ('4','Jamie','Almazan','261572144','4609 Trails End Road

Boca Raton','FL','33486','Project Manager');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('5','Jenifer','Haskell','261572144','1100 Lighthouse Drive

Branson','MO','65616','Cast Member');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('6','Thomas','Ross','473169635','3499 Jewell Road

Minneapolis','MN','55406','Engineer');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('7','Pedro','King','259362145','359 Despard Street

Atlanta','GA','30318','Inspector');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('8','Gabriel','Morgan','259362145','2015 Forest Drive

Chantilly','VA','22021','Project Manager');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('9','Charline','Miner','081382314','1945 Henry Ford Avenue

New York','NY','10007','Cast Member');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('10','Michael','Stewart','081382314','4435 Dancing Dove Lane

Garden City','NY','11530','Engineer');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('11','Derrick','Lay','772342348','2067 County Line Road

Tampa','FL','33602','Project Manager');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('12','Mary','Clark','120785631','394 Stanley Avenue

Garden City','NY','11530','Project Manager');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('13','Susan','Howard','360168563','9 Cherry Camp Road

Chicago','IL','60643','Engineer');

INSERT INTO Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('14','Ernest','Parker','360168563','2695 Windy Ridge Road

Fort Wayne','IN','46804','Project Manager');

--Data ProjectType :

INSERT INTO ProjectType (projectTypeCode, projectTypeDesc)

VALUES ('FAC','Facility');

INSERT INTO ProjectType (projectTypeCode, projectTypeDesc)

VALUES ('RIDE','Ride');

INSERT INTO ProjectType (projectTypeCode, projectTypeDesc)

VALUES ('FOOD','Restaurant');

--Data Activity Type :

INSERT INTO ActivityType (activityTypeCode, activityTypeDesc)

VALUES ('DE','Design');

INSERT INTO ActivityType (activityTypeCode, activityTypeDesc)

VALUES ('CO','Construction');

INSERT INTO ActivityType (activityTypeCode, activityTypeDesc)

VALUES ('WA','Warranty');

--Data Project:

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('1','Timely Pluto','FAC ','1','45845','20160511','Inactive','20160811','4');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('2','Rich Mars','RIDE','3','65225','20150511','Active','20160311','8');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('3','Lost Kangaroo','FOOD','4','52365','20150706','Cancelled','20160420','4');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('4','Ninth Snake','FOOD','6','105542','20150804','On-Hold','20151207','8');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('5','Cold Zeus','RIDE','7','78524','2015-10-12','Active','2016-04-23','4');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('6','Black Scorpion','RIDE','8','142554','2015-11-12','On-Hold','2016-06-11','14');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('7','Steamy Nitrogen','FOOD','5','235445','2015-09-06','Active','2016-05-05','12');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('8','Rough Haystack','FAC','3','101445','2015-09-23','Active','2016-07-18','12');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('9','Sticky Hideous Lobster','FOOD','9','86004','2015-07-15','On-Hold','2016-03-16','14');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('10','Needless Clown','RIDE','2','452225','2015-09-23','Active','2016-12-21','4');

INSERT INTO Project (projectID, projectName, projectTypeCode, firmFedID , fundedBudget, startDate, status, projectedEndDate,projectManager)

VALUES ('11','Cloudy Eagle','FAC','10','786522','2015-05-23','Inactive','2017-06-13','8');

--Data Activity

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('1','Steep Tuba','3','25000','Active','CO','2015-08-04','2015-12-30');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('2','Gloomy Roadrunner','2','14500','Active','DE','2015-10-15','2016-08-21');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('3','Wild Dog','7','27000','Inactive','CO','2015-11-05','2016-05-03');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('4','Tasty Uranium','3','12000','Cancelled','WA','2015-07-25','2016-02-13');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('5','Dead Parachute','8','18004','Cancelled','WA','2015-07-25','2016-02-13');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('6','Tidy Compass','6','154000','Active','CO','2015-05-27','2016-08-12');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('7','Disappointed Spider','4','220000','Active','CO','2015-09-14','2016-07-11');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('8','White Iron','8','25300','On-Hold','DE','2015-12-04','2016-11-03');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('9','Random Snake','9','55000','Active','CO','2016-02-04','2016-12-31');

INSERT INTO Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate)

VALUES ('10','Sad Steel','5','34564','Active','WA','2015-09-25','2016-07-21');

EXTRA CREDIT:

--Trigger to handle inserts into Project view

CREATE TRIGGER Trg\_INS\_vw\_Project ON vw\_Project

INSTEAD OF INSERT

AS

BEGIN

    --Check if the firm exists if it does not then insert it

    IF (NOT EXISTS (SELECT Firm.FirmFedId

                 FROM Firm, inserted

                 WHERE Firm.FirmFedId = inserted.FirmFedId))

                    BEGIN

                    INSERT INTO Firm

                    select FirmFedId, FirmName, FirmAddress

                   from inserted

                    END

    --Check if the project type code exists if it does not then insert it

    IF (NOT EXISTS (SELECT ProjectType.projectTypeCode

                 FROM ProjectType, inserted

                 WHERE ProjectType.projectTypeCode = inserted.projectTypeCode))

                    BEGIN

                    INSERT INTO ProjectType

                    select projectTypeCode, ProjectTypeDesc

                    from inserted

                    END

    --Finally insert the rest of the project data, validation is handle by key constraints

    INSERT INTO Project

    select projectId,ProjectName, ProjectTypeCode,FirmFedID,fundedBudget,StartDate, Status, ProjectedEndDate, ProjectManager

    from inserted

END

--Test Data for project entity view

insert into vw\_Project

(projectID, projectName, firmFedID, firmName, firmAddress,fundedBudget,startDate,status,projectTypeCode,projectTypeDesc,projectedEndDate,projectManager)

VALUES

('14','Super Fast Teacups','3','Paleoxo','2506 Ralph Drive New Lyme ,OH 44066',52365,'20150511','Active','RIDE','Ride','20160311','8')

--Trigger to handle inserts into Activity view

CREATE TRIGGER Trg\_INS\_vw\_Activity ON vw\_Activity

INSTEAD OF INSERT

AS

BEGIN

-- Insert data into activity

               INSERT INTO Activity

               select activityID, activityName, projectID, costToDate,status,activityTypeCode,startDate,endDate

               from inserted

-- Check if the activity type code exists and if it doesn't. This part only exists if somewhow activity type gets more than just 3 different values

IF (NOT EXISTS (SELECT ActivityType.activityTypeCode

  FROM ActivityType, inserted

  WHERE ActivityType.activityTypeCode = inserted.activityTypeCode))

               INSERT INTO ActivityType

               select activityTypeCode, activityTypeDesc

               from inserted

END

--Test for activity entity

INSERT INTO vw\_Activity (activityID, activityName, projectID, costToDate,status,activityTypeCode,activityTypeDesc,startDate,endDate)

VALUES ('14','Sad Steel','5','34564','Active','WA','Warranty','2015-09-25','2016-07-21');

--Trigger to handle inserts into the Employee view

CREATE TRIGGER Trg\_INS\_vw\_Employee ON vw\_Employee

INSTEAD OF INSERT

AS

BEGIN

--Dump the inserted data into the employee view

               INSERT INTO Employee

               select \*

               from inserted

END

--Test for Employee View

INSERT INTO vw\_Employee (empNumber, firstName, lastName, ssn,address,state,zip,job)

VALUES ('16','Pablo','Valencia','463845623','4981 Mulberry Street

Nacogdoches','TX  ','75961','Engineer');