Project 2

50 points

Building upon your project 1, the park will be a Star Wars themed park. You must design additional parts of the database and create the following SQL Script.

Step 1: Design and create the tables

You must create additional tables to hold Project and Activity Data.

A **project** represents the construction of a facility with a limited scope of work and financial funding. A Project can be composed of many activities which indicate the different phases in the construction cycle.

Example Project Name: Bobba Fett's Bounty Chase Ride

An **activity** represents the work that must be done to complete the project.

Example Activity Name:

For Example activity name could be "Phase 1 Design of Bounty Chase ride"
Or name could be "Final construction of Bounty Chase ride"
Etc…

You must normalize the project table to come up with a new set of tables. You will then write the create script for these tables.

> **Project** (projectId, projectName, firmFedID, firmName, firmAddress, fundedbudget, projectStartDate, projectStatus , projectTypeCode, projectTypeDesc, projectedEndDate, projectManager, (activityId, activityName, costToDate, activityStatus, startDate,endDate))

To normalize the tables, you must use the following function dependencies:

> ProjectId,ActivityId -> projectName, firmFedID, firmName, firmAddress, fundedbudget, startDate, projectStatus , projectTypeCode, projectTypeDesc, projectedEndDate, projectManager, activityName, costToDate, activityStatus, startDate, endDate.
>
> projectId -> projectName, firmFedID, fundedbudget, startDate, projectStatus , projectTypeCode, projectedEndDate, projectManager.
>
> projectTypeCode -> projectTypeDesc
>
> firmFedID -> firmName, firmAddress
>
> ProjectId, ActivityId -> activityName, costToDate, activityStatus, startDate, endDate

When creating the tables, **use the following column names and data types (important)** for columns:

- **projectId** (char(4)) : A 4 character unique identifier (numbers and letters).
- **projectName** (varchar(50)) : The name of the construction project.
- **firmFedID** (char(9)) : A 9 character Federal ID (Example: 123456789)
- **firmName** (varchar(50)): The name of the construction firm.
- **firmAddress** (varchar(50)) : The address of the construction firm.

- **fundedbudget** (decimal(16,2)): The money amount allocated to this project
- **projectStartDate** (date): The date the project started.
- **projectstatus** (varchar(25)): The status of the project (either active,inactive,cancelled,completed)
- **projectTypeCode** (char(5)): The project type code are FAC, RIDE, RET, and FOOD.
- **projectTypeDesc** (varchar(50)): The project type descriptions for a project are: Facility, Ride, Retail and Restaurant
- **projectedEndDate** (date) The date the project is scheduled to end.
- **projectManager** (char(8)) The employee number of the employee who is managing this project
- **activityId** (char(4)): A 4 character unique identifier for the activity.
- **activityName** (varchar(50)): The name of the activity.
- **costToDate** (decimal(16,2)): The cost of the activity to date.
- **activityStatus** (varchar(25)) : The status of the activity (either active,inactive,cancelled,completed)
- **startDate** (date): The date the activity began.
- **endDate** (date): The date the activity ended.

You will write the script to create the tables which resulted from your normalization. Each table should have a primary key defined. You should have more than one table after you normalize.

Step 2: Create Stored Procedures to populate the tables

You will create the SQL Scripts to create procedures to insert/ update data. The following definitions specify the parameters that can be passed in. The underlined parameters are required.

Make sure that your procedure inserts records if the required parameters do not exist, but they update records if the required parameters do exist.

For example:

If SP_AddUpdateProject: passes in projectID “AA01” and it DOESN’T exists in the project table(s) , it will insert the values passed in.

If SP_AddUpdateProject: passes in projectID “AA01” and it DOES exists in the project table(s) , it will UPDATE the values passed in for the AA01 record.

Procedures Needed:

- SP_AddUpdateProject: Adds/Updates a project with all the field information.
  - **Parameters:** projectId, projectName, firmFedID, fundedbudget, projectStartDate, projectStatus, projectTypeCode, projectedEndDate and projectManager

- SP_DeleteProject: Deletes a project by the project Id.
  - **Parameters:** projectId

- SP_AddUpdateActity: Adds/Updates activity with all the field information.
  - **Parameters:** activityId, activityName, projectId, costToDate, activityStatus, startDate, endDate

- SP_DeleteActivity: Deletes an activity by the activity Id.
  - **Parameters:** projectId, activityId

Step 3: Create Stored Procedure to Process Project Delays

You will create the SQL Script to create procedures to insert/ update data and process a project delay

- SP_ProcessProjectDelay: Given a project Id, this procedure finds if any max end date of any activity within the project is after the project's projected end date. If this happens, the procedure will calculate how many days it is late (use DATEDIFF) and fines the project $100 for each day late it is late.
  In addition, the project table's "projectedenddate" will be updated with the new end date and the "fundedbudget" will be updated with the original funded budget plus the fines per day late.
  - **Parameters:** projectId.

Example:

The Falcon Coaster has a ProjectId "AA01" has a projected end date of 6/30/2017. It has 2 activities:

ActivityId: AA90 ActivityName: Build Coaster EndDate: 6/01/2017

ActivityId: AA91 ActivityName: Inspect Coster EndDate: 7/30/2017

Since Activity AA91 ends 30 days after the projected end date of the project, the project will have an additional $3,000 (30 X $100) added to the fundedbudget column's original value. Also, the project's new projected end date will be "7/30/17"

How you will turn in your project

You will turn in 1 SQL Script file, which will have the filename format First Four letters of last name + underscore + First Name Initial + PantherId + PROJ2.

So,if your name is John Smith, ID 166723 your filename will be

Smith_J166723_PROJ2.sql

(If your last name is less than 5 characters then just use those characters)

Header and Database:

You will have a header on the file and create all the SQL objects under your own database

```
/*      Name: FirstName Last Name
        Project #
        PantherId: ######
        Semester:
*/

--All your project will create the objects under your own personal database that you will have to
create manually.

--Your personal database should have the following name format:
--First Four letters of last name + underscore + First Name Initial + PantherId
Use Smith_J166723
GO

Then you will need to have an insert statement to an assignment table that I will use for grading by
populating this script with your pantherId, firstname, lastname, databasename (see above) and
assignment (1, 2 or 3)

/***This must be created for every assignment and must be done at the beginning) ****/
insert into master.dbo.assignments
(pantherId, firstname, lastname, databasename, assignment)
values
('7777777', 'John', 'Smith','Smith_J7777777',2)
GO
/*************************************************************************/

--Rest of Project Scripts HERE
```

Grading Criteria:

- Proper Normalization of Project into multiple tables with create scripts. (20 points)
- Completed Project AddUpdate Procedure (5 points)
- Completed Activity AddUpdate Procedure (5 points)
- Completed DeleteProject Procedure (5 points)
- Completed DeleteActivity Procedure (5 points)
- Completed ProcessProjectDelay Procedure (10 points)

I will run all your scripts on a database so please make sure there are no errors and everything runs smoothly on a brand-new database.