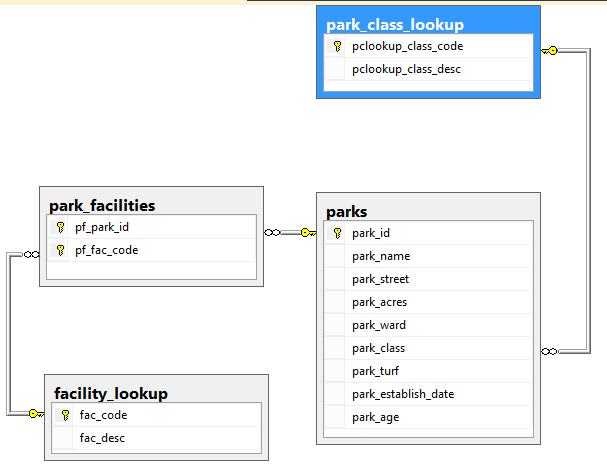
IST359 - Project 1 (30 pts.) Due: Sunday, Oct 1st. Early Bird Deadline Friday, sept 29th is worth 2 bonus pts.

## Overview

In this project you will build a relational database model in SQL to support the functions of a Chicago Park system. It focuses on a very small portion of the system - the part dealing with parks and facilities. The example is based on the poorly designed table we looked at during week one. The model you build will include an internal model (schema), some sample data, and an important verification step.

The following is a **poorly constructed** database diagram of the model you will implement. It provides solutions to many of the problems you identified in weeks 1 and 2 but we will make more changes after we look more closely at logical modeling and normalization later this semester.

**Final Project Grade** = Pts Earned (90 max) / 3 + Bonus Pts – Penalty Pts (Wrong Format and other issues that prevent me from grading your assignment as submitted.



# Assignment Q&A

## The big picture

To complete this project you will write and hand in 1 SQL script (template provided) and 1 database diagram. The script will drop and create the data model and populate it with initial sample data. This script should re-create the entire schema each time it is executed and it should run successfully top-to-bottom, over and over again. Some of the instructions you’ll be given are specific, others are vague, it is **your responsibility** to translate what the assignment is asking for into working SQL code.

In general, the set of tables helps keep track of the parks and facility info needed for a fictitious park management company. It is a scaled down set of data that includes some basic information about different parks and facilities. Later in the semester we’ll get some additional event and usages information in there.

## Can I work with a partner on this assignment?

You can work alone or with 1 partner. If you work with a partner you and your partner will receive the same grade. If you plan on working with a partner you must submit one assignment file with both names included as a comment at the top of your script. Do not turn in the assignment separately. **Sharing answers across groups is strictly prohibited.** Violations of this rule will insure that both parties (the giving and taking team) receive a 0 on the assignment, forfeit the right to work in groups on future assignments, and will have an Academic Integrity violation filed with the appropriate offices on campus.

A note of caution: don’t let one person do the entire assignment. It will only hurt the person who doesn’t actively participate when it comes time for the mid-term exam.

## How much time is this going to take me?

Estimates have this project taking between 4 and 5 hours, depending on your skill and comfort level with the class work so far. If you wait until the day it’s due you’ll jeopardize your ability to get help so please don’t procrastinate. I will **NOT** be holding office hours for this assignment AFTER Thursday Sept 28th so waiting until that time to start the assignment is not a great plan if you want the support of others. Late assignments will not be accepted for any reason. Deadlines serve a valuable purpose and missing them have consequences. 12 days is more than enough time to accomplish this task.

## What if I need some help?

I am more than willing to give you the help you need. I will not, however, do the project for you. If you need clarification of what’s expected to complete part of the assignment then please ask me. If you’ve written some SQL but can’t decide if it is correct or cannot find your mistake I will nudge you in the right direction, but I will not answer the questions for you. For example, if you don’t know how to create a table, I will not show you how to create one of the tables in the assignment. I will show you how to create some other table or refer you to the work we did in the labs, in class examples, or in your books and notes. If you are getting syntax errors I can help you identify the source of the problem but won’t correct the code for you. This is your opportunity to shine!

## What will I be handing in for this Project?

You will upload one **ZIP** file. It will contain your **SQL submission script file and one database diagram**. The script will create the SQL schema and data (all the SQL you write for parts 1-3 and 5 – 6). The zip will also contain one WORD or PDF file from Part 4 that will contain the Database Diagram. Included with this assignment is an empty template you can use to build out your SQL scripts. Using this template is a huge time saver. You must use it. Be sure to include your name (and the name of your partner if working in a group) at the top of your script and on the database diagram. This expedites grading.

## How will this be graded?

Check the project details section for details related to how much each item is worth. You should be able to estimate your score on this assignment; I will return a completed grading form to you after the project due date.

# Assignment Details

This section outlines each of the SQL objects you need to create and add to your SQL script file for this assignment.

# Part 1 – table Drops and Meta-data creation [20 points]

Create each of the following tables using the SQL CREATE TABLE statement. Be sure to use the submission script template file that I provided. The script has been constructed such that you can use it over and over. It should execute top to bottom as you build each step.

**How This Part Is Graded:** You will receive **the** points indicated for dropping all of the tables and your procedure. Remember your script must be able to execute without error multiple times. Do your DROPs work the very first time you execute your script? Will they execute correctly once each table has data? Each table must have **all** columns defined correctly. A correct column has the column name, data type, default and null values set correctly in your working SQL script.

### Drop All [3 points]

Using the appropriate SQL statement DROP each of the **your tables.**

**Reminder:** The order you drop tables is important so you don’t violate FK constraints. Do they work the first time and will they work on subsequent executions of your script? I run each script multiple times.

### TABLE: facility\_lookup [2 points]

This table contains information about facilities that are found in parks. It is a lookup table to insure that we add valid facilities for each park. You’ll note that I’ve described each column data type or given an example of what it may contain. But my words may not be valid transact-sql data types. That is up to you to select the best physical constraint.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Allow null | Default Value |
| fac\_code | Allow 2 characters | No | Ex. SC |
| fac\_desc | Allow up to 30 characters | No | Ex. Senior Center |

### TABLE: park\_class\_lookup [2 points]

Just another little lookup so we don’t code invalid park classes when we create a park.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Allow null | Default Value |
| pclookup\_class\_code | You decide | No |  |
| pclookup\_class\_desc | Allow up to 50 characters | No | Ex. Community |

### TABLE: parks [5 points]

This table is the heart of our database as it contains are park information. There is a lot of data here and we’ll be adding even more later in the semester.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Allow null | Default Value |
| park\_id | This is the event id. Surrogate key | No | We’d like our parks to be higher than 500 and increment by one. |
| park\_name | This is the name of the park | No |  |
| park\_street | You decide | No |  |
| park\_acres | You decide | No |  |
| park\_ward | You decide | No |  |
| park\_class | You decide |  |  |
| park\_turf | This will be a flag that indicates whether or not the park has any turf fields. It is a yes/no field (true/false). | No | False |
| park\_establish\_date | This is the **year** the park was purchased | No |  |
| park\_age | COMPUTED!!!! | NA | This represents the number of years the park has been in service. Note you will let the database create **the data** for this column much like it does for surrogates. **WARNING:** Independent research required. |

### TABLE: park\_facilities [8 points]

This table is a bit different as it contains the information about a particular facility at a certain park. This table solves the problem we saw in our original data when we had parks that contained multiple facililities. It is the BRIDGE table that resolves a Many-to-Many relationship if you think back to IST352.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Allow null | Default Value |
| pf\_park\_id | You’ll be placing an actual park id here so what should the data type be? | No |  |
| pf\_fac\_code | You’ll be placing an actual facility code here so what should the data type be? | No |  |

# Part 2 – Table Constraints [10 points]

**Using the ALTER TABLE statement**, add these table constraints to the 4 tables you created in the previous part. Once you know your SQL is correct be sure to add it to your submission script!

**How This Part Is Graded:** You will receive points for each correct table constraint you add.

### 2.1 TABLE: park\_class\_lookup [2 point]

|  |  |  |
| --- | --- | --- |
| Constraint Name | Constraint Type | Value or condition used in the constraint |
| pk\_pclookup\_class\_code | Primary key | pclookup\_class\_code |

### 2.2 TABLE: facility\_lookup [2 points]

|  |  |  |
| --- | --- | --- |
| Constraint Name | Constraint Type | Value or condition used in the constraint |
| pk\_fac\_code | Primary key | fac\_code |

### 2.3 TABLE: parks [3 points]

|  |  |  |
| --- | --- | --- |
| Constraint Name | Constraint Type | Value or condition used in the constraint |
| pk\_park\_id | Primary key | park\_id |
| ck\_park\_ward | Check | Investigate WARDS in Chicago and write a check constraint that insures a reasonable number is entered. |
| ck\_park\_acres | Check | Let’s be sure this is always greater than 0. |

### 2.4 TABLE: park\_facilities[3 points]

|  |  |  |
| --- | --- | --- |
| Constraint Name | Constraint Type | Value or condition used in the constraint |
| pk\_park\_fac\_id | Primary key | pf\_park\_id and the pf\_fac\_code (Research Required) |

# Part 3 – Foreign Key Constraints [10 points]

Add these foreign key constraints to each of the following tables using the ALTER TABLE statement. Once you know your SQL is correct be sure to add it to your submission script!

**How This Part Is Graded:** You will receive **one** point for each correct foreign key constraint you add.

### 3.1: TABLE: parks [5 points]

|  |  |  |
| --- | --- | --- |
| Constraint | Column | PK table |
| fk\_park\_class | park\_class | park\_class\_lookup |

### 3.2: TABLE: park\_facilities [5 points]

|  |  |  |
| --- | --- | --- |
| Constraint | Column | PK table |
| fk\_pf\_park\_id | pf\_park\_id | parks |
| fk\_pf\_fac\_code | pf\_fac | facility\_lookup |

# Part 4 – Database Diagram (10 Points)

In this part, you will create a database diagram that includes all of the tables you just created. When you’re done creating the tables, table constraints, and foreign keys, you should be able to create a database diagram, called **ParkDIAGRAM** from your SQL tables. NOTE: similar to the diagrams we’ve created in the labs, your diagram **must include the column name, data type, and allow null column** for each table. Be sure to take the time to **line up the relationship that is drawn so the ends of the lines actually point to the corresponding pk and fk columns**. Then place this diagram into a word document or PDF or jpg. If your lines do not connect the appropriate column names or you omit physical constraints I cannot give credit. Please be sure to include your name (and the name of your partner) on this diagram.

# Part 5 – Insert Base Data (Data 20)

In this part, you will write **SQL INSERT** statements to add data into the tables as you determine necessary based on the data on the following pages. You’ll see it is a subset of what we talked about in week 1.

**How This Part Is Graded:** You will receive points for each part when you insert the data correctly using the SQL INSERT statements into each of our 4 tables. Remember, the order you place data in your tables is critical to insure referential integrity.

**IMPORTANT:**  Please insert all of the data you see here except the city, state, zip and GEO LOCATION INFO. After each **INSERT**, please execute a **SELECT** all rows, all columns on the table to make sure the data order matches what you see here.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Location | Acres | Ward | Class | Turf? | Facilities | Date |
| ABBOTT (ROBERT) | 49 E 95th St Chicago, IL 60628 Chicago, IL 60628 (41.72153854, -87.62214661) | 22.74 | 6 | COMMUNITY | No | Senior Center  Spray Feature | 1947 |
| ADAMS (GEORGE & ADELE) | 1919 N Seminary Ave Chicago, IL 60614 Chicago, IL 60614 (41.91665268, -87.65535736) | 0.66 | 43 | NEIGHBORHOOD | False | Playground  Water Playground | July 7, 1959 |
| ARCHER (WILLIAM BEATTY) | 4901 S. Kilbourn Ave. Chicago IL 60632 Chicago, IL 60632 (41.80400848, -87.73569489) | 13.22 | 23 | MINI-PARK | No | Fitness Center  Gymnasium  Spray Feature  Boce Court | 1934 |
| ARCHER (WILLIAM BEATTY) | 3309 S Shields Ave Chicago, IL 60616 Chicago, IL 60616 (41.83438492, -87.63520813) | 8.22 | 11 | COMMUNITY PARK | 1 | Football/Soccer  Pool(outdoor)  Pool (Indoor)  Go-Cart Track | 12/1/1941 |
| ARMSTRONG (LILLIAN HARDIN) | 4433 S St Lawrence Ave Chicago, IL 60653 Chicago, IL 60653 (41.81406021, -87.61099243) | 8.6 | 6 | COMMUNITY PARK | NA | Tennis  Hiking Trail | 1941 |

# Part 6 – Data Verification via Queries (20 Points -10 each)

In this part, you will write **SQL Query** statements to insure your data matches the input criteria above.

**Query 1:** Please show all of the columns in the parks table along with the facilities description contained within.

**Query 2:** Please provide the following for each ward containing a park. Your output will look like this

**Ward Num Parks Total Acres**

6 2 31.34

11 1 8.22

23 1 13.22

43 1 0.66

# Submission Requirements

If you have an error in your script file, it’s best to comment out the SQL code causing the error so that you can get the full points for other sections. I want to avoid the lengthy process of hand-editing your SQL in order to get it to work! So I will only grade scripts that execute from top to bottom twice!

## Handing it in

Take both your SQL script file and your database diagram file (pdf, jpg, or word) and zip them up in a “Zip file Archive” then upload the zip file to Blackboard. Note that the **file extension must be ZIP**. I cannot open RAR or any other format and no credit for the assignment will be given. The zipped file should be named “last name\_names\_A1.zip”. For example, if you were in a group of 2 it would be smith\_jones\_A1. If you did the assignment alone it would be just smith\_a1.

## Grading Note

How is this graded? I add up your points earned. After that I add in bonus points if earned and subtract penalty points (for not following submission instructions). This is a great opportunity to earn 32 points this week.  
  
**Best of Luck to you on Assignment 1!!!**