Final Project Outline

Your job is to create the SQL infrastructure needed for a startup company to track its metrics. This can take several different shapes, maybe employee data, customer data, or whatever other types of data they need. You need to make sure that they have a way to ingest data (ETL), a way to easily display their data (View), they must also have procedures for backing up or purging data or some other task (Stored Procedure). You must also create a function to do some task that needs to be repeated on the data (Function). You must create a safe way for a user to manipulate the data (Select, Insert, Update, Delete) without compromising the original data (view).

You can use data from a real source like stock values, or you can use mock data.

Mock Data Websites:

- https://www.mockaroo.com/
- https://cobbl.io/
- https://generatedata.com/

I recommend before starting you define the goal you want to achieve and the structural outline of your solution before diving into code.

Requirements

- ETL Package
- Database
- Tables
- Stored Procedure
- Function
- SQL Queries (Select, Insert, Update, Delete)
- Cleanup Script (Removes everything that you created for easy clean up)

Part 1 Due 1/4/2024:

Database Diagram:

Keeping in mind the end goals of your final project you will need to create a diagram for your solution. This is the same idea as the diagram from the first project. This should include all tables, foreign keys, and primary keys. You may wish to include any other constraints on columns, and datatypes for each column. You can use whatever tool you wish to create this. This will be handed in first and a separate assignment on teams.

Here are some tools you may wish to use:

- Word
- https://drawsql.app/
- https://dbdiagram.io/home
- https://app.diagrams.net/
- https://www.quickdatabasediagrams.com/
- https://miro.com/index/

Or any other tool that you feel comfortable using.

Part 2 Due 1/12/2024:

Write Up:

For your project, you will write a brief 1-page description of your final project. This should include a description of the structure of your database and tables, as well as descriptions of the functionality of your code such as the functions, stored procedures, and ETL package. This should be easily understood by someone with little knowledge of SQL. The purpose of this is to provide people who do not know SQL the ability to roughly understand what your project does.

Submission:

To submit your final project, you must create a complete package that someone can run to add to a SQL Server everything that is needed for your project. So this should consist of SQL files to create the database, tables, stored procedures, functions, and Queries. I would also recommend providing a Database backup file for easier setup, though not required. You should also provide your ETL package and data to load the data into the database. You can either submit the project in a Google Drive folder with public access to download the files or a public Github Repository with all the files (recommended, as long as your files are not too large 25MB or greater). For the write up if using the Github submission method, you can have a Word file in your repository or a README file in your repository (recommended). For those using the drive method just place a word document in the folder with the write-up in it.

To Submit the project to CanCodeComunities you will place a link to your drive folder or Github repository in the teams' assignment and submit it.

Rubric (This rubric will be used for reviewing your project):

Criterion (Score 0 if element is absent)	Below Expectations (1)	Meets Expectations (2)	Exceeds Expectations (3)	Score
STORED PROCEDURE	Stored Procedure does not compile; DML items within store procedure do not execute properly to perform their intended function	Stored Procedure compiles properly and all DML within stored procedure execute to their intended function	Stored Procedure compiles properly, all DML within stored procedure execute to their intended function, and stored procedure calls or creates additional DDL	
FUNCTION	Function does not compile; DML items within function do not execute properly to perform their intended function	Function compiles properly and all DML within function execute to their intended function	Function compiles properly, all DML within function execute to their intended function, and stored procedure calls or creates additional DDL	
TABLE	Table creation does not compile, or table does not have all the needed columns	Table compiles and has all needed columns	Table compiles and has all needed columns; indexes and referential integrity constraints are also created	
VIEW	View does not compile; DML items within view do not execute properly to perform their intended function	View compiles properly and all DML within view execute to their intended function	View compiles properly, all DML within view execute to their intended function, and stored procedure calls or creates additional DDL	
SELECT	SELECT does not execute properly to perform their intended function	SELECT executes to its/their intended function(s)	SELECT executes to its/their intended function(s), and involves multiple joins, grouping, or union-ing that could not be handled in one query	
INSERT	INSERT does not execute properly to perform their intended function	INSERT executes to its/their intended function(s)	INSERT executes to its/their intended function(s), and demonstrates all three INSERT techniques (INSERT INTO VALUES, INSERT INTO SELECT, and SELECT INTO)	

UPDATE	UPDATE does not execute properly to perform their intended function	UPDATE executes to its/their intended function(s)	UPDATE executes to its/their intended function(s), and involves multiple joins	
DELETE	DELETE does not execute properly to perform their intended function	DELETE executes to its/their intended function(s)	DELETE executes to its/their intended function(s), and involves multiple joins	
Talend Job(s)	Talend Job(s) does not compile, contain tasks within the Talend job, or error upon execution	Talend Job(s) compile successfully, all tasks execute successfully (inbound files load, outbound files produce, etc).	Talend Job(s) compile successfully, all tasks execute successfully (inbound files load, outbound files produce, etc) and have multiple transformations to the data.	
File Deliverables	No files delivered or not all files delivered	All files delivered, and file layout matches requirements	N/A	
			TOTAL	