Project 2

Project Info

Due Date

Week 5: Friday Oct. 4th, 2024

LATE SUBMISSIONS WILL NOT BE ACCEPTED

Submission: Submit to the Project 2 submission drop box.

See "Submission Requirements" below for details on how to generate the required file.

Keep working with the same assignment group. Significant overlap between submissions from two or more students may be flagged for plagiarism.

Grading

Item	Description	Weight
FK Indexing	Indexes across all foreign keys, including junction tables, are correct.	35%
Query Indexing	Indexes for identified queries are correctly created.	15%
Unique Constraints	Proper Alternate key constraints/unique indexes exist for all required columns.	25%
Check Constraints	All required check constraints exist and are appropriate.	15%
Defaults	All required defaults exist and are appropriate.	10%
Redundant Indexes	Deduction for overlapping indexes	Up to -20%
Deduction:		
Error on Execution	Deduction for failure to run without modification	Up to -20%
Submission Requirements	Deduction for failure to adhere to the submission requirements	Up to -20%
Total		**100%**

Business Requirements

You have been provided with a partially completed FanshaweDroneShare database for a system that tracks a drone/accessory sharing program. After some testing, it has been determined that the structure alone is insufficient to properly constrain the database, and testing also revealed performance concerns, especially when doing Select queries to join 2 tables together. You have been asked to add constraints and indexes to satisfy these business and technical requirements.

The customer has told you that whenever drone/accessories are checked in or out by pilots, a scanner reads the serial number which is used to uniquely identify that drone/accessory in the database. Searches on SerialNumber will need to be properly optimized and constrained. To help identify & communicate with the Pilots who are primarily responsible for the Account, they will often do lookups to determine this information, so those searches must be optimized as well.

In testing, the customer found that a station's maximum capacity would sometimes incorrectly be set to a negative number. Since this is invalid, they would like to prevent negative numbers from being added to the MaxCapacity column. As well weight of Drone equipment must always be non-negative. Testing also found that station records were frequently searched by StationName, which can be used to uniquely identify records. They would like these lookups optimized.

The customer has identified dates in the system (DroneEquipment.ManufacturedDate and Account.AccountOpenDate) that should never be a future date, so they must be equal to or less than the current datetime. When testing these tables, the customer discovered that Accounts will often be uniquely identified by AccountNumber, and Pilots may be uniquely searched for by TransportCanadaCertNumber. These lookups will need to be optimized.

To improve consistency, the customer has asked you to provide some appropriate defaults when complete account details are not available. When not provided, CurrentBalance should be set to 0 and AccountOpenDate should be set to the current date. In addition, since most pilots will be from Fanshawe and the surrounding area, suitable defaults should be included for all appropriate address fields.

A review of queries on the Address table identified three queries that need to be optimized. The first query creates a sorted list of provinces and cities. This list is sorted by province first, then by city. The second query looks up records by province only. The last query looks up records by city only.

They would like lookups by the parent and child columns optimized for all foreign keys. They would also like you to optimize lookups in both directions across the junction tables - the customer has found that lookups will frequently be done in both directions across all foreign keys. Covering indexes should be provided to quickly look up Pilots by Addresses, Addresses by Pilots, Pilots by Accounts, or Accounts by Pilots.

Technical Requirements

In addition to satisfying the business requirements, you have been asked to follow these technical standards.

1) You must work with the Project 2 Initial Database schema provided in the Project 2 folder. You may not use your own.
2) You should not be altering the existing schema, including the database name. The correct solution will only add new constraints and indexes.
3) There should be no duplication of indexes. If an index is entirely covered by another index it should be removed. Remember, some indexes are automatically generated by SQL Server (Unique constraints for example .. hint hint..).
4) Indexes may not include any redundant columns.
5) The function GETDATE() should be used to get the current datetime.
6) You will submit the entire database using the instructions below. This includes the database that was provided to you along with the modifications you made to satisfy these requirements.

Submission Requirements

You will submit a complete database script generated by or run in SQL Server. You could/should submit the code you used to create your database (including these Project 2 constraint and Index requirements), and you are encouraged to do so for maintainability purposes, to ensure you understand & can re-use this SQL code (eg. when you must use this raw SQL code to create a test version of this database for use by the rest of your development team).

1st Submission .txt File: Continue using the Project 2 Initial Database based on Project 1 SQL script and:

1. Add your own scripts for the new required tables into the same file. Use comments to separate the answers for each requirement
2. Once you finish, rename the SQL file name, then submit it.

Note: Ideally you will create the database directly using the SQL code required to create the tables/relationships/constraints (vs using the GUI design tool/wizards – as mentioned in class using these GUI design tools/wizards is not the best way to learn SQL/design since these tools will always be different from one DBMS to another, but the SQL will be much more standard/portable). If you have the complete SQL code, including all the CREATE TABLE, Relationships/ Constraints, etc. as per the requirements above then submit all that SQL code in 1 .sql file (and in a separate version you save with a .txt subscript).

2nd Submission .sql file: Script out the database (**Make sure you are satisfied with the database you have created before scripting it out/submitting it to the submission folder**):

1) Right click on your database and select "Tasks->Generate Scripts…"
2) If you see an introduction page, click "Next >"
3) On the "Choose Objects" page, select **"Script entire database and all database objects".**
4) Click "Next >"
5) On the "Set Scripting Options" page, ensure that "Save scripts to a specific location", "Save to file", and "Single file" are all selected.
6) Change the filename so that it is your first and last name with a .sql extension (e.g. "BruceWayne.sql")
7) Click "Next >"
8) Review your selections and click "Next >"

This will create a .sql file. Submit this file to the dropbox.

You must test your script before submitting. (ie. Re-run it in SQL Server Management Studio to ensure it executes).