Final Project

**Point Value:** 150

# Instructions

## Overview

The Final project for DS700 simulates a real-world scenario. There are two parts to the Final Project-

Part-A: Students will utilize the SQL skills they have acquired in the course to generate the dataset required for the analysis portion in Part-B. The data is present in the form of normalized tables on the SQL server database “”. In the SQL portion, students will write a query that joins multiple tables and creates a dataset

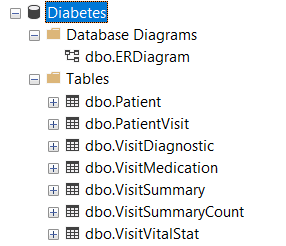
Part-B: The R portion of the project will test your ability to wrangle, graph, analyze, and understand a complicated, real-world data set using R. This portion of the project will use the tidy version of the data you prepared during Part-A.

The data used for constructing the Diabetes database was derived from the 10-years real world clinical care data at 130 US hospitals and integrated delivery networks. Reference: [Kaggle data source](https://www.kaggle.com/brandao/diabetes).

Please note that we have only used a sub-set of the data from the Kaggle source and have made major modifications to it. So please use the SQL server Diabetes database and not the dataset from Kaggle. If you use the dataset from Kaggle, you will get a zero for the project.

## Database

For this project, you will be working with the **Diabetes** database. You can access the database by logging into the Virtual Lab’s SQL server instance.



## Entity Relationship (ER) Diagram

Please refer to the Database Diagram “dbo.ERDiagram” in the database.

Tip: Below is a snapshot of the ER diagram. Please zoom in to see details in the picture below.



## **Instructions: Part-A (SQL portion: 50 points)**

**What to submit:**

For the SQL portion, your submission will include-

* a Word document (LastName\_FirstName.docx) listing all the scenario questions below (1-4), the SQL query for each and a snapshot/screenshot of the SSMS Results window showing first 10 rows for each query execution.
* a SQL file (LastName\_FirstName.sql) file which will include SQL queries for all the questions below (1-4) along with proper comments. I will be executing all the queries inside the .sql file at once, so please make sure it is error free.
* a CSV file (LastName\_FirstName.csv) containing the dataset exported from Results window of question #4 SQL execution.

**SQL Section Details:**

Before we get started with generating the dataset for Part-B, we will try to find out if we are missing any patient data at a high level. At this stage we are not worried about missing values in a particular column but rather entire records. Think of this step as a cursory review of a dataset. And then the last question below (scenario #4) will be used to generate the .csv file that will be uses as the input dataset for the R portion of the project (Part-B).

**Provide the SQL query along with a snapshot of the result window for the following scenarios-**

* 1. Write and execute a SQL query to list all patient numbers and encounter ids which do not have **any** data (missing rows) in VisitSummary table. Order the result by patient number, encounter id. [10pts]

--Question 1 Part 1

--Inner Joined Patient Visit and Visit Summary where encounter id was equal and used that as the subquery to see if anything

--existed that was in visit summary, but not in the patient visit table

SELECT PV2.[patient\_nbr]

,PV2.[encounter\_id]

FROM [Diabetes].[dbo].[VisitSummary] AS VS2

left JOIN [Diabetes].[dbo].[PatientVisit] AS PV2 on PV2.[encounter\_id] = VS2.[encounter\_id]

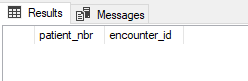
WHERE VS2.[encounter\_id] not in

(SELECT PV.[encounter\_id]

FROM [Diabetes].[dbo].[PatientVisit] AS PV

INNER Join [Diabetes].[dbo].[VisitSummary] AS VS on PV.[encounter\_id] = VS.[encounter\_id])

ORDER BY PV2.[patient\_nbr], PV2.[encounter\_id]



* 1. Write and execute a SQL query to list all patients whose information about race **or** gender has changed during their future visits. A future visit is defined as T1.encounter\_id < T2.encounter\_id. Your output should have patient number, old and new race values, old and new gender values, and the encounter ids. [10pts]

--Question 1 Part 2

--Use the with common table expression to create temporary data set that can be manipulated

--We then use the output of the with clause and self join to create a master table

--that can be filtered down to when we have a race input difference or a gender input difference

With

myTable as

(SELECT PV.[patient\_nbr]

,PV.[encounter\_id]

,VVS.[race]

,VVS.[gender]

FROM [Diabetes].[dbo].[PatientVisit] AS PV

Inner Join [Diabetes].[dbo].[VisitVitalStat] AS VVS on PV.[encounter\_id] = VVS.[encounter\_id]

)

select T1.patient\_nbr AS old\_patient\_nbr

,T2.patient\_nbr AS new\_patient\_nbr

,T1.encounter\_id AS old\_encounter\_id

,T2.encounter\_id AS new\_encounter\_id

,T1.race AS old\_race

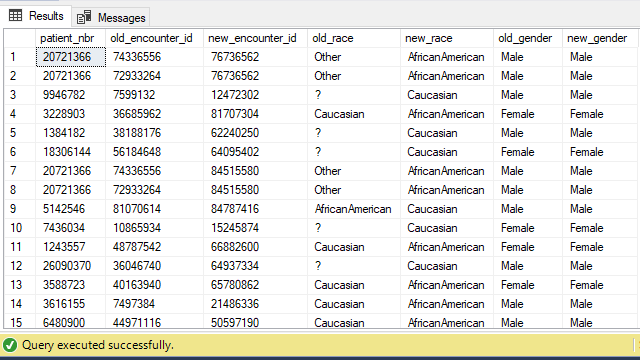
,T2.race AS new\_race

,T1.gender AS old\_gender

,T2.gender AS new\_gender

from mytable t1, mytable t2

where T1.patient\_nbr = T2.patient\_nbr AND T1.encounter\_id < T2.encounter\_id AND (T1.race != T2.race OR T1.gender != T2.gender)



* 1. Write and execute a SQL query to list all the patient data for all encounters. Even if patient data is missing in other tables, you still need to list the patient\_nbr and encounter\_id in your output. Your output should list all the columns (without repeating them) starting with encounter\_id, patient\_nbr… and it should be ordered by patient\_nbr and then encounter\_id. [10pts]

SELECT DISTINCT

PV.[encounter\_id]

,PV.[patient\_nbr]

,VVS.[age]

,VVS.[gender]

,VVS.[race]

,VVS.[weight]

,VS.[admission\_source\_id]

,VS.[admission\_type\_id]

,VS.[discharge\_disposition\_id]

,VS.[medical\_specialty]

,VS.[payer\_code]

,VS.[readmitted]

,VS.time\_in\_hospital

,VD.[diag\_1]

,VD.[diag\_2]

,VD.[diag\_3]

,VD.[number\_diagnoses]

,VSD.[num\_lab\_procedures]

,VSD.[num\_medications]

,VSD.[num\_procedures]

,VSD.[number\_emergency]

,VSD.[number\_inpatient]

,VSD.[number\_outpatient]

,VM.[acarbose]

,VM.[acetohexamide]

,VM.[change]

,VM.[chlorpropamide]

,VM.[citoglipton]

,VM.[diabetesMed]

,VM.[examide]

,VM.[glimepiride]

,VM.[glimepiride\_pioglitazone]

,VM.[glipizide]

,VM.[glipizide\_metformin]

,VM.[glyburide]

,VM.[glyburide\_metformin]

,VM.[insulin]

,VM.[max\_glu\_serum]

,VM.[metformin]

,VM.[metformin\_pioglitazone]

,VM.[metformin\_rosiglitazone]

,VM.[miglitol]

,VM.[nateglinide]

,VM.[pioglitazone]

,VM.[repaglinide]

,VM.[rosiglitazone]

,VM.[tolazamide]

,VM.[tolbutamide]

,VM.[troglitazone]

FROM [Diabetes].[dbo].[PatientVisit] AS PV

LEFT JOIN [Diabetes].[dbo].[VisitVitalStat] AS VVS ON (PV.[encounter\_id] = VVS.[encounter\_id])

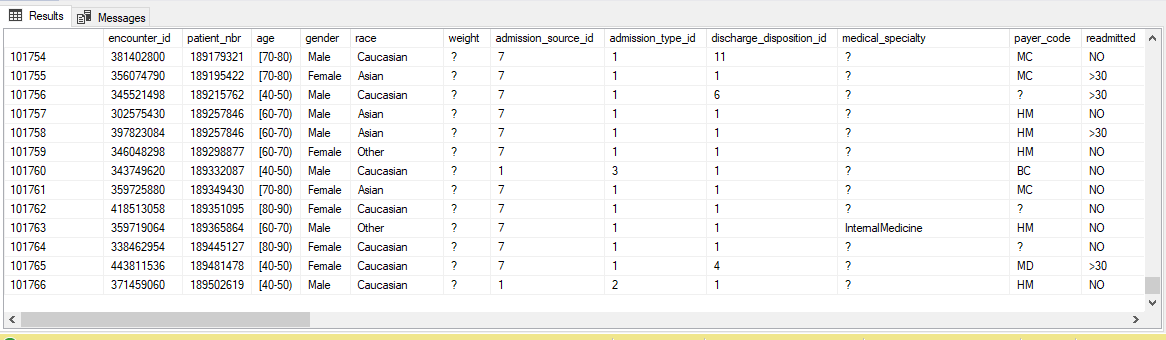
LEFT JOIN [Diabetes].[dbo].[VisitSummary] AS VS ON (PV.[encounter\_id] = VS.[encounter\_id])

LEFT JOIN [Diabetes].[dbo].[VisitDiagnostic] AS VD ON (PV.[encounter\_id] = VD.[encounter\_id])

LEFT JOIN [Diabetes].[dbo].[VisitSummaryCount] AS VSD ON (PV.[encounter\_id] = VSD.[encounter\_id])

LEFT JOIN [Diabetes].[dbo].[VisitMedication] AS VM ON (PV.[encounter\_id] = VM.[encounter\_id])

ORDER BY PV.[patient\_nbr], PV.[encounter\_id]



* 1. Write and execute a SQL query to list all the patient data for all encounters. If the patient data is missing in other tables like Visit summary, then DO NOT include this encounter in your output. Your output should list all the columns (without repeating them) starting with encounter\_id, patient\_nbr… and it should be ordered by patient\_nbr and then encounter\_id. [10pts]

--Used inner joins across all tables and did not include the patient table in this query

SELECT DISTINCT

PV.[encounter\_id]

,PV.[patient\_nbr]

,VVS.[age]

,VVS.[gender]

,VVS.[race]

,VVS.[weight]

,VS.[admission\_source\_id]

,VS.[admission\_type\_id]

,VS.[discharge\_disposition\_id]

,VS.[medical\_specialty]

,VS.[payer\_code]

,VS.[readmitted]

,VS.time\_in\_hospital

,VD.[diag\_1]

,VD.[diag\_2]

,VD.[diag\_3]

,VD.[number\_diagnoses]

,VSD.[num\_lab\_procedures]

,VSD.[num\_medications]

,VSD.[num\_procedures]

,VSD.[number\_emergency]

,VSD.[number\_inpatient]

,VSD.[number\_outpatient]

,VM.[acarbose]

,VM.[acetohexamide]

,VM.[change]

,VM.[chlorpropamide]

,VM.[citoglipton]

,VM.[diabetesMed]

,VM.[examide]

,VM.[glimepiride]

,VM.[glimepiride\_pioglitazone]

,VM.[glipizide]

,VM.[glipizide\_metformin]

,VM.[glyburide]

,VM.[glyburide\_metformin]

,VM.[insulin]

,VM.[max\_glu\_serum]

,VM.[metformin]

,VM.[metformin\_pioglitazone]

,VM.[metformin\_rosiglitazone]

,VM.[miglitol]

,VM.[nateglinide]

,VM.[pioglitazone]

,VM.[repaglinide]

,VM.[rosiglitazone]

,VM.[tolazamide]

,VM.[tolbutamide]

,VM.[troglitazone]

FROM [Diabetes].[dbo].[PatientVisit] AS PV

INNER JOIN [Diabetes].[dbo].[VisitVitalStat] AS VVS ON (PV.[encounter\_id] = VVS.[encounter\_id])

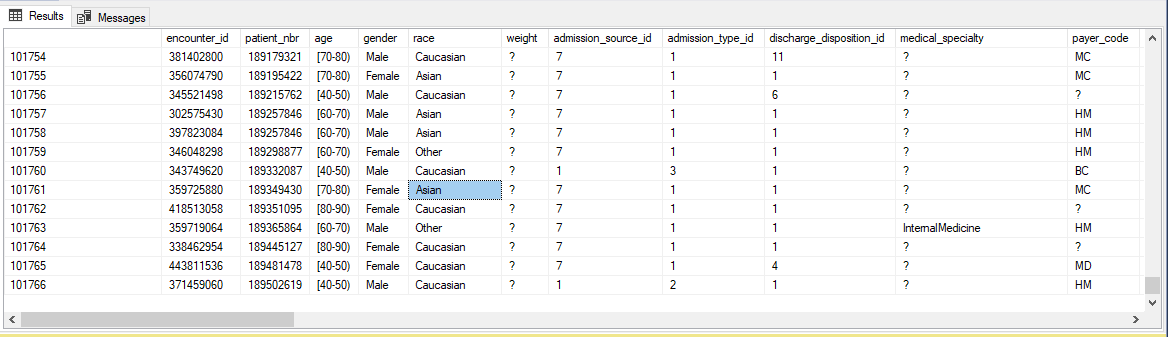
INNER JOIN [Diabetes].[dbo].[VisitSummary] AS VS ON (PV.[encounter\_id] = VS.[encounter\_id])

INNER JOIN [Diabetes].[dbo].[VisitDiagnostic] AS VD ON (PV.[encounter\_id] = VD.[encounter\_id])

INNER JOIN [Diabetes].[dbo].[VisitSummaryCount] AS VSD ON (PV.[encounter\_id] = VSD.[encounter\_id])

INNER JOIN [Diabetes].[dbo].[VisitMedication] AS VM ON (PV.[encounter\_id] = VM.[encounter\_id])

ORDER BY PV.[patient\_nbr], PV.[encounter\_id]



Lastly, you will export the Result dataset from scenario #4 above into a .cvs file (LastName\_FirstName.csv). You will use this exported file in Part-B of the project and will also be submitting it with your project submission. [10pts]

Export steps

* *Before you begin:* Please save all your work in SSMS before you attempt the export.
* Follow any of the below methods to export the query results to csv file-
  + <https://learnsql.com/blog/how-to-export-csv-from-sql-query/>   (Methods 2/3/4)
  + <https://stackoverflow.com/questions/3169220/export-query-result-to-csv-file-in-sql-server-2008>
  + <https://stackoverflow.com/questions/45849662/ssms-export-query-results-to-excel-or-csv>
* Save your .CSV file with the name – “LastName\_FirstName.csv”

## **INSTRUCTIONS: PART-B (R portion: 100 points)**

**What to submit:**

For this section of the project, you will be submitting

* a .Rmd file (LastName\_FirstName.Rmd) that will contain all the code and written answers for the steps mentioned below.
* a knitted file (LastName\_FirstName.pdf ) from the .Rmd file.

**Details:**

The R portion of the project will test your ability to wrangle, graph, analyze, and understand a complicated, real-world data set using R. This portion of the project utilizes the tidy version of the data (LastName\_FirstName.csv) you generated during the SQL portion of the project.

**In a .Rmd file, write the code and include written answers to the following steps:**

Qus 1 – [50 pts]

1a. Read the data. Tell R to treat admission\_type\_id, discharge\_disposition\_id, and admission\_source\_id as categorical.

1b. Clean the variables `diag\_1`, `diag\_2`, and `diag\_3` using the code you wrote in the Lesson 11 Homework. You should end up with 3 new variables, `diagnosis1`, `diagnosis2`, and `diagnosis3`, containing human-readable interpretations of the ICD-9 billing codes.

1c. Remove the old variables `diag\_1`, `diag\_2`, `diag\_3`. If you created any intermediate variables (such as `injury\_cause1`), remove those also.

1d. Use code to determine which columns have no variability (i.e., all rows contain the same value). Print the names of the columns, and remove those columns.

Examide and citoglipton are the columns that were removed due to having no variability.googl

Qus 2 – [50 pts]

2a. Count the number of missing values that are denoted by NA. Count the number of missing values that are denoted by "?". (In a factor/character variable, R will not automatically ignore rows containing "?".) Which column has the most missing values?

Weight has the most missing values of any column with 98569 observations of ?.

2b. Replace the "?" in the `race` column with NAs. Then use mode imputation to impute the NAs in the data. (Leave the "?" in other columns alone.)

Extra credit questions 3 & 4 [OPTIONAL]

Please note the following for questions 3 & 4 below-

* Questions 3 & 4 are OPTIONAL and students will get total 50 extra points for completing them.
* Both these questions have been purposefully kept vague and specific guidance will not be provided by the instructor. We want to see if you are able to synthesize the knowledge gained in R portion of the course.
* Students are allowed to discuss a solution to Q3 & Q4 without revealing specific code on Piazza. You are however allowed to share very small chunks of code to help with code errors.

Qus 3 – [25 extra PTS]

3a. Use linear regression to model the number of diagnoses a patient has as a function of `age` and `diagnosis1`.

3b. As age increases from [0-10) to [90-100), what happens to the number of diagnoses? Explain your answer based on the linear regression. Illustrate your answer with a graph using `ggformula`.

3c. Compared to people whose diagnosis1 is a blood issue (the default value), which types of diagnosis1 are significantly associated with an increased number of diagnoses? Which types are significantly associated with a decreased number of diagnoses? Explain. Illustrate your answer with a graph using `ggformula`.

Qus 4 – [25 extra PTS]

4a. Under the Hospital Readmission Reduction Program, hospitals where large proportions of patients are readmitted within 30 days may be penalized in their Medicare reimbursements. Create a binary variable that tells whether `readmitted` is `<30` or not.

4b. Make a bar graph and a conditional bar graph of the relationship between `diabetesMed` and whether a patient was readmitted within 30 days. Also make a bar graph and a conditional bar graph of the relationship between `gender` and whether a patient was readmitted within 30 days.

4c. If we want to predict whether a patient will be readmitted within 30 days, which variable is more informative: `diabetesMed` or `gender`? Explain your choice, referring to specific aspects of the graphs you made.

4d. Use logistic regression to model whether a patient will be readmitted within 30 days, based on `number\_inpatient` and \*one\* of the variables `diabetesMed` or `gender` (choose the one that is more informative).

4e. Write 2-3 sentences describing the direction of association between the predictor variables and the probability that the patient will be readmitted within 30 days.

4f. Make \*one\* graph showing the predicted probability of readmission as a function of both predictor variables.

**Now, knit your .Rmd file containing all the steps above into a LastName\_FirstName.pdf file**. You will be submitting the knitted file along with the pdf file for the R portion of the project. Your knitted file should not be more than 15 pages. *If you submit a knitted file with more than 15 pages, you will get a zero for the R portion of the project.*

# What to submit

1. You will be uploading **five** files-
   * A Word document (LastName\_FirstName.docx): listing all the scenario questions (1-4), the SQL query for each question and a snapshot/screenshot of the SSMS Results window showing first 10 rows for each query execution.
   * A SQL file (LastName\_FirstName.sql): containing the SQL queries for all the questions (1-4) along with question numbers and proper comments. I will be executing all the queries inside the sql file at once, so please make sure it is error free.
   * A CSV file (LastName\_FirstName.csv) containing the dataset exported from Results window of question #4 SQL execution.
   * A .Rmd file (LastName\_FirstName.Rmd) containing the code and written answers for the R portion of the project.
   * A knitted file (LastName\_FirstName.pdf ) from the .Rmd file in R portion of the project [up to 15 pages].
2. For each of the questions in the SQL portion of the project, provide the SQL query and along with a snapshot of the output obtained from the query execution in SSMS.

Example:

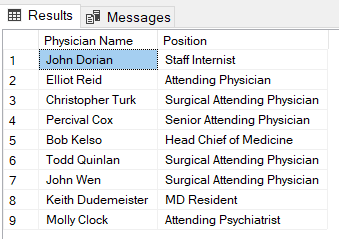
Q. List all the physician names and their positions.

Answer:

SELECT [Name] as "Physician Name"

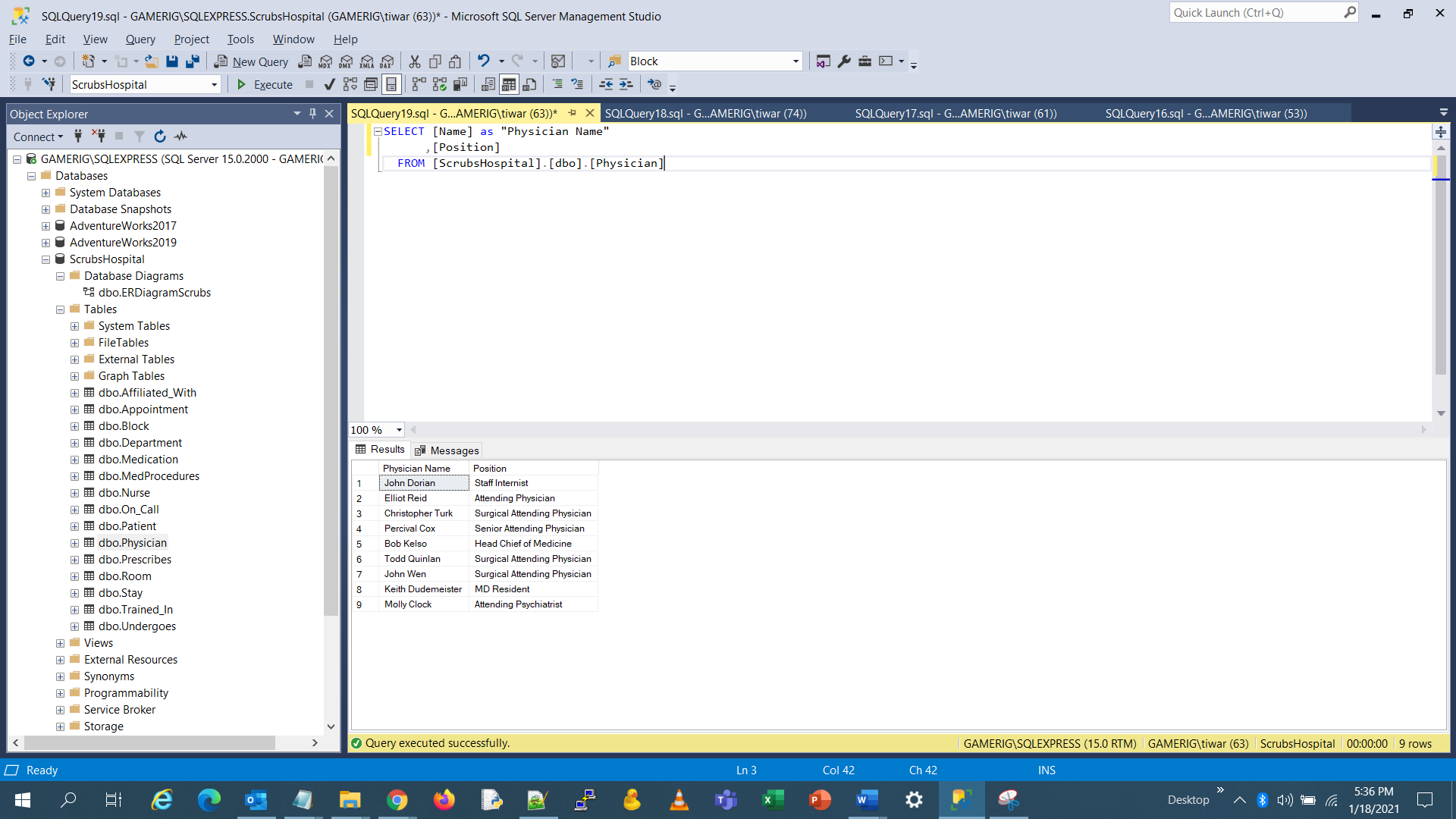
,[Position]

FROM [ScrubsHospital].[dbo].[Physician]



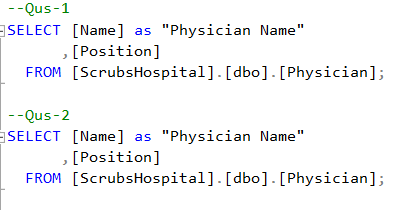
1. DO NOT insert a screenshot of the entire screen. You only need to provide the snapshot of Results window. Points will be deducted if the snapshot is not properly edited to display only the relevant information as shown in point #2 above.

Example: this is NOT acceptable



1. In your SQL file, you will put a comment corresponding to each question number and then the query below it. All queries need to be separated with semicolon and the entire sql file should be error free. I will execute all the queries at the same time from the file.

Example: You SQL file will look something like this-



1. Make sure that all your files are named correctly. Upload all the FIVE files on the assignment page in Canvas for submission. Note: you need to submit all the five files together in your submission.
2. Please note that the system might automatically append numbers to your file names after the submission. This is an acceptable behavior. If I can see your last name and first name, your submission will be valid. If the file name does not have your last name and first name, points will be deducted.
3. You can submit multiple times before the project deadline and your most recent version will be graded. *But during each submission, all five files (.docx, .sql, .csv, .Rmd, .pdf) need to be submitted together*.