During this quarter, we learned a variety of components of a database and how to develop SQL statements/queries that Inserted, Updated, Deleted, and Selected records.  When Selecting records, we learned to develop criteria within the WHERE clause to find specific records and sort them.  We also learned to connect tables through Inner and Outer Joins.  We also learned how to create Views to allow us to pull specific queries from existing query results.  Lastly, we learned to re-run popular/necessary queries numerous times based on parameters used in the Stored Procedures.

**Please perform all parts (Parts 1 thru 6) and use screen shots to verify the SQL and resulting data from each part and task.  Please post the Part and sub-request numbers above each screen shot along with a brief description of the task.**

**PLEASE READ ENTIRE DOCUMENT BEFORE STARTING.  PART 6 uses components of other parts to create Views and Stored Procedures.  Knowing this ahead of time can save re-typing SQL code.**

**1) Create the following tables**:

**Students**  
Student\_ID (Big Int, Primary Key, Identity Spec =  Yes)  
FIrstName (NVarChar, 25)  
LastName (NVarChar, 25)  
Email (NVarChar, 75)  
Phone (NVarChar, 10)

**Courses**  
Course\_ID (Big Int, Primary Key, Identity Spec =  Yes)  
CourseName (NVarChar, 50)  
CourseDescription (Text - Text allows us to write as much as we want.)  
Credits (int)  
  
**Grades** (notice that this table connects the other two)  
Grade\_ID (Big Int, Primary Key, Identity Spec =  Yes)  
Course\_ID (Big Int)  
Student\_ID (Big Int)  
LetterGrade (NVarChar, 2) Potential: A, A-, B+, B, B-, C+, C, C-, D+, D, F  
DateCompleted (DateTime)

**Part 2: Create records in your new tables**  
Create at least 10 students  
Create at least 10 different courses (Bogus, have fun, do NOT have to be NEIT classes)  
Create grades records for some students using a variety of information including students who took numerous classes (Multiple records, same students, different classes.)   
Leave some courses that were never taken and students that never earned grades (no grade records for a course or a student.)  
Have some students take a class a couple of times (failed it first time, then passed it the second time.)  
**Make sure that Grades are connected to actual students and courses.**

**Part 3: Give me an example of an insert, update, and delete in SQL.**  
Make sure that the SQL statements include a WHERE statement specifying a record by its primary field, so only one field gets updated or deleted.  For the Deletion, make sure it is for a record that will not adversely effect the other tables. I recommend deleting a record from the Grades table.

**Part 4: Creating basic SQL Queries   
\*\* make sure there are more than 1 record matching.  You may need to add records to make this happen, if you did not enter enough sample data. \*\***  
**(SELECT \* FROM tablename WHERE criteria ORDER BY field)**  
1) Create a query to pull up all student info and sort by Last Name  
2) Create a query to pull up all student info and sort by Last Name and First Name  
3) Create a query to pull up all of a students Grades based on their Student ID.  
4) Create a query to pull up all the grades for a specific course.

**Part 5: Getting data from multiple tables**  
1) Pull up a specific course's list of students (Name & email) and their grades (inner join)  
2) Pull up a  specific student's courses and their grades (inner join)  
3) Pull up a list of courses and the grades gotten in them (left join - should have some nulls)

**Part 6: Creating Views and Stored Procedures**  
1) Create a View with a query to pull up all student info and sort by Last Name and First Name (Part 4 #2)  
  
2) Create a Stored Procedure to Pull up a specific student's courses and their grades (inner join) based on the student ID parameter (Part 5 #2)  
  
3) Create a Stored Procedure to Pull up a specific course's list of students (Name & email) and their grades (inner join) based on the Course ID parameter (Part 5 #1)