INSTRUCTIONS:

<u>Due Date</u>: **Wednesday, April 7th, 2020, before 11:59 PM**

Deliverables for part 1:

1. CREATE TABLE statements (including all key constraints).
2. Proof (screenshots) that you executed the CREATE TABLE commands successfully.

Deliverables for part 2:

1. All queries for the given problems.
2. Proof of the results (screenshots) of the execution for each query.

Submission Instructions:

1. If you are working in a pair of two, only one student should submit the project.
2. Clearly specify your name, your team member's name and student ids in the documentation of this project.
3. Submit a single word/pdf document containing part 1 and part 2 queries and proofs/results. **Do not submit zip files.**
4. File naming convention: <NetId1>_<NetId2>_phase2.doc or <NetId1>_<NetId2>_phase2.pdf
5. Make sure you use clear image quality for the screenshots. The screenshots should be readable and should show all result records.
6. Missing honor code shall cost - 20 points. No bonus for the honor code.
7. Submit your assignment any time before midnight of the due date. Late policy: -5% out of 100 each day.

Students are required to not share any of the project related documents and solutions with others in any way or form even after the completion of the project. Students may, however, show their projects to interviewers.

HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

DOCTORAL database:

INSTRUCTOR(<u>InstructorId</u>, FName, LName, StartDate, Degree, Rank, Type)
TT(<u>InstructorId,</u> NoOfPhDStudents)
PHDSTUDENT(<u>StudentId</u>, FName, LName, StSem, StYear, Supervisor)
GTA(SectionId, MonthlyPay, <u>StudentId</u>)
GRA(<u>StudentId</u>, Funding, MonthlyPay)
SELFSUPPORT(<u>StudentId</u>)
SCHOLARSHIPSUPPORT(<u>StudentId,</u> Type, Source)
COURSE(<u>CourseID</u>, CName)
SECTION(<u>SectionId</u>, CourseId)
COURSESTAUGHT(<u>CourseID, InstructorId</u>)
MILESTONE(<u>MId,</u> MName)
MILESTONESPASSED(<u>StudentId, MId,</u> PassDate)
GRANTS(<u>AccountNo,</u> Type, GrantTitle, Source, StDate, EndDate, StAmount, CurrBalance)
GRANTSASSOCIATED(<u>AccountNo, InstructorID,</u> PIType)
PHDCOMMITTEE(<u>StudentId, InstructorId</u>)

 Part 1: Write CREATE TABLE statements to create the 15 tables for the DOCTORAL database. Each table should follow one of the 15 input files. Your CREATE statements should also mention all primary key and foreign key constraints. Then load the data from each input file to the corresponding table.

Execute the following simple SQL query/statement on each of the 15 database tables to get the result: SELECT * FROM <table name>

Provide clear screenshots of the outputs you get on executing above SELECT query as proof of successful CREATE TABLE statements.

Part 2: Write the following queries and views on the database.

1. CREATE VIEW Statement-
    Create a view COURSESTAUGHT_INFO(INSTRUCTOR.InstructorId, INSTRUCTOR.FName, INSTRUCTOR.LName, COURSE.CourseId, COURSE.CName) that retrieves the info about each INSTRUCTOR and the COURSE they teach. The view should have the attributes renamed as: (IId, IFName, ILName, CId, CoName).
2. Write the following queries on the VIEW you created:
    A. Retrieve for each instructor the IId, IFName, ILName, NoOfCoursesTaught
    B. Retrieve the course name(s) taught by the instructor Ahmed Sarhan
    C. Retrieve the names of the instructors (FName and LName) who teach - CSE4351, CSE4354, and CSE5324. List the course id against the instructor's name.

3. Write the following additional queries on the original database tables:

i) Retrieve the names of all GRAs and list the grant supporting them.

ii) List the instructor names, and the number of students they are currently supervising.

iii) Retrieve course Id, course name and the number of corresponding sections for all courses that have more than two sections.

iv) Retrieve funding account number, grant type, grant title for all grants that had a starting amount greater than 400,000. Additionally, list the student id of the students they are supporting

v) Retrieve the name, rank, and type of the instructors whose names start with 'C' and the ids of the courses they teach.

vi) Retrieve the total, average, minimum, and maximum monthly payment for the GRAs.

vii) Retrieve the course name, course id, section id for all courses having the word 'programming' and order the results in descending order.

viii) Execute a command to delete a record that violates a referential integrity constraint.

ix) Execute 3 insert commands that attempt to insert 3 records, such that the records violate the integrity constraints. Make each of the 3 records violate a different type of integrity constraint.

x) Retrieve the student id, name, and the milestone passed by the students along with the dates when they did so. You do not need to include the students who have not passed any milestone.

xi) Retrieve the names of the instructors who started teaching after December 25, 1989, and the total number of the students they have supervised.

xii) Retrieve the names, starting year, and milestone name for all students who started in the year of 2017 or after and they passed their milestone after May 1st, 2018.

xiii)Retrieve the name, start date and the degree of all instructors who started between May 1st, 1999 and May 1st, 2004.

xiv) Retrieve the student id, name, and milestones passed (milestone id) for the students who are supported by Maverick scholarship.

xv) Retrieve the first name of all PhD students and any sections they might be assigned as GTAs. List your results in the ascending order by the first name of the GTA.

xvi) Retrieve the first name of all GTAs and the sections they are assigned to. List your results in the ascending order by the first names of the GTA.