# M20Temp18: DATA SYSTEMS

# PROJECT PHASE 0

### Deadline : 11:00 PM, 30 August 2020,Sunday

In the phase 1 you are supposed to get familiar with the codebase and complete two basic tasks.Kindly go through the tutorial 1 recording to get basic familiarity with the boiler plate code.

**Programming Language Allowed: C++**
**Note: Kindly follow the submission instructions before submitting your code as otherwise there will be a penalty.**

## INSTRUCTIONS:

## PLEASE REGISTER ONLY AFTER YOU'VE READ ALL THE INSTRUCTIONS

1. First, make sure you and your teammate are on a call together.
2. Next, ONE of the two of you click on the following like. **NOT BOTH. ONE OF THE TWO.**
   https://classroom.github.com/g/7Vy_6zeb

To the one that clicked on the link first:

1. Ensure your teammate hasn't already created a team for you. If they have then follow the instructions meant for the second teammate.
2. You should see a screen prompting you to either create a new team or join an existing team. If you and your teammate have followed the instructions, a repo for your team shouldn't exist, if it does mail the TAs.

3. Create a new team by entering your team name exactly as is in the moodle group we've created for you. This should take you to your own private team repository where you should already have the boiler plate code in the repository

To the teammate going second:

1. Click on the same link - https://classroom.github.com/g/7Vy_6zeb
2. Grant all the necessary permissions.
3. You should now be at the same page your first teammate was at - a page prompting you to join a team or create a new one.
4. If you and your teammate read the instructions carefully, your team should already exist in the list of teams that show up. Just click on the join button for your team's repo

You may now register for the project. For any issues mail the TAs at datasystems_tas_m20@IIITAPhyd.onmicrosoft.com
**Make sure to register for the project by 23rd August 2020, Sunday, 11:59PM**

# TASK 1:

## Dataset:

A basic version of the employee database with various tables has been provided to you to work with for this phase. Remember your code will be evaluated on a superset of this database. The database has following restrictions:

- All the entries in the database are integers.
- No spaces are allowed in table names as well as column names.
- Keep the table names and assignment variable names unique.

You can find a detailed description of the dataset in this Link

## QUERIES

Given the employee database you are supposed to write queries to find the following. Duplicates rows are allowed.

**Q1)** List all employees who work in departments different from their supervisor (i.e. employee and supervisor work in different departments)
FORMAT: **Q1(Ssn, Dno, Super_ssn, Super_dno)**

**Q2)** List all employees who work on any projects from their departments
FORMAT: **Q2(Ssn, Dno, Pno, P_dno)**

**Q3)** List all employees who work on any projects from outside their department
FORMAT: **Q3(Ssn, Dno, Pno, P_dno)**

**Q4)** Find all supervisors who are younger than their employees
FORMAT: **Q4(Super_ssn, Super_bdate, Essn, E_bdate)**

**Q5)** List all departments that have both male and female employees working in that department
FORMAT: **Q5(Dno)**

Note that the format of the output table has been given i.e. the table you finally export as the solution to the query has to have the same table name and column names as described in the format.
If some resultant table is empty, you do not have to export the table

## TASK 2:

Given a NXN (N can be large) positive integer matrix A. You need to store it in data blocks for stable storage. Compute the number of data blocks needed to store the matrix A. Now implement a program to read matrix A, and store back transpose of matrix A in the **same data blocks**. It would be best if you decided how you will store the matrix in data blocks and access the matrix for the project.

Detail your approach in a report file named **Matrix.md.** State the number of data block accesses taken for the above operation in terms of BLOCK_SIZE and N in the Matrix.md file.

To do this task you will need to make the following changes

- Implement a MATRIX Object like TABLES
- In the case of tables, we implicitly assume a block is at least as big as a table row, but this isn't the case for matrices. A whole row or column of the matrix need not fit into a block.
- The maximum block size you are allowed is 8KB. Note that the BLOCK_SIZE parameter is in KB

**SYNTAX**

To load a matrix from a csv file named A.csv

```
> LOAD MATRIX A
```

Where A.csv is of the form

Int1, int2, … intN

Int, int, …, int

In matrices, there are no column names, so unlike tables, the first row of the .csv file will not contain the column names, it contains the first row of the matrix.

To transpose

```
> TRANSPOSE A
```

## SUBMISSION INSTRUCTIONS:

1) For Task-1 you are supposed to write the queries for all the subtasks in a **single file named queries.txt** and assign the output of each subtask to a table name **Qn** where n represents the subtask number.

   Result File should look like this:

   Logic for Subtask 1

   - 
   - 

   **Q1 <- Result for Subtask 1**

   - 
   - 

   **Q5 <- Result for Subtask 5**

   **EXPORT Q1**

   - 
   - 

   **EXPORT Q5**

   **QUIT**

You will have to export your resultant tables using the export command and the last line of your file has to be the QUIT command

2) Read the **README.MD** file in the project github repo carefully to set up your project work. Make sure to push your project work back to your repository.

**Any case of Plagiarism will be seriously dealt with and will lead to an F in the course.**