DMS-Pretest 4

1. Create a database named **dbPretest4** with the following specifications :
   a. Primary file group with the data file **dbPretest4.mdf**. The size, maximum size, and file growth should be 8, unlimited and 20 respectively.

   b. Log file **dbPretest4_log.ldf.** The size, maximum size, and file growth should be 8, 50, and 10% respectively.

2. Create the tables as follows:

- table **tbStudents**

Field Name	Data Type	Constraint
stID	varchar(5)	Primary Key
stName	varchar(50)	Not null
stAge	tinyint	>=14 and <=70
stGender	bit	Default 1

- table **tbProjects**

Field Name	Data Type	Constraint
pID	Varchar(5)	Primary Key
pName	Varchar(50)	Not null, unique
pType	Varchar(5)	'EDU' or 'DEP' or 'GOV'
pStartDate	Date	Not null, Default current-date

- table **tbStudentProject**

Field Name	Data Type	Constraint
studentID	Varchar(5)	Not null, foreign key
projectID	Varchar(5)	Not null, foreign key
joinedDate	date	Not null, Default current-date
rate	tinyint	From 1 to 5
Primary key: studentID + projectID		

3. Insert some records to each table:

a. tbStudents:

Student ID	Student Name	Age	Gender
S01	Tom Hanks	18	1
S02	Phil Collins	18	1
S03	Jennifer Aniston	19	0
S04	Jane Fonda	20	0
S05	Cristiano Ronaldo	24	1

b. tbProjects

Project ID	Project Name	Project Type	Start Date
P20	Social Network	GOV	12/01/2020
P21	React Navtive + NodeJS	EDU	22/08/2020
P22	Google Map API	DEP	15/10/2019
P23	nCovid Vaccine	GOV	16/05/2020

c. tbStudentProject

Student ID	Project ID	Join Date	Rate
S01	P20	12/02/2020	4
S01	P21	12/03/2020	5
S02	P20	16/02/2020	3
S02	P22	01/09/2020	5
S04	P21	12/04/2020	4
S04	P22	01/10/2020	3
S04	P20	16/10/2020	3
S03	P23	04/07/2020	5

4. Create a clustered index '**IX_stname'** for ***stname*** column on **tbStudents** table.
   Create an index '**IX_pID'** for ***projectID*** column on **tbStudentProject** table

5. Create a view '**vwStudentProject**' to display the list of students joined to projects had start-date before '***Jun-01-2020'***, including following information :
   StudentID, Student name, Student Age, Project name, Start date, Join date and Rate.
   Note: this view will need to check for domain integrity and encryption.

6. Create a stored procedure **'upRating'** with an input parameter '**student-name**', and

output parameter '**avg-rate**'

- If '**student-name**' is null, displays all the projects that all students have worked for. Otherwise, displays information about that students and the corresponding projects they have joined.
- Procedure also returns the average rate mark (**avg-mark**) that students joined into projects.

7. Create trigger '**tgDeleteStudent'**, it will remove all projects that student have worked for whenever a **DEL** statement triggered on table '**tbStudents**'.