S No.	Mini-project description
1	Student Management: This is a small case study to manage student enrollment and allocation of department for a recently opened university.

Student Table		
Coloumn Name	Type	Constraint
Student_Id	Number	Primary Key
Student_Name	Varchar	Not null
Age	number	default 17
Address	Varchar	null
Parent_Contact	Number	null ; should be of 10 digits
P_G_Name	Varchar	Not null
email	Varchar	null
Department	number	FK related to Department Table
Year of Joining	number	4 digit year; default current year

Department Table		
Coloumn Name	Type	Constraint
Department_Id	number	Primary Key
Department_Name	varchar	
Total_Students	Number	Max 30 students

1. Create the above mentioned tables as per the given specification and constraints
2. Populate Department table with sample data of 5 rows. Fix minimum of 5 departments and add them.
    a. Keep 0 for Total_Students.
3. Populate Student table with sample data of 15 rows.
    a. Initial few rows should be fully filled with all fields values
    b. Insert few rows with null data. 2 rows with address as null and 2 contact no as null, one row with both as null.
    c. Make few rows to be inserted with default value for age and year of joining.
4. Update Department table's Total_Student column with total number of records found in student table for that department. [Use getStudentsCount(deptId) function described below]
5. Create procedure to insert data into StudentTable.
    a. Provide 9 IN parameters for table columns and 1 OUT parameter to give the status.
    b. Before Insert please check whether whether Total_students is < 30.

	i. if total is < 30 then 1. Insert record into student table 2. Update Total_Students + 1. 3. Assign “inserted” to status variable ii. Else 1. Assign “Department Full” to status 6. Create a function called getContact which take one *IN parameter* of Student_Id and returns the contact number of that Student. 7. Create a function call getStudentsCount which takes one in parameter of department_Id and returns the total students available in that department. 8. Create a table called Student_BackUp with all the records from the Student table. 9. Create a table called Test_Report with columns Department_Id, Department_Name, Tot_No_Students, Avg_Age_Students. 10. Create a script called Table_recover.sql in which you write a query that takes a table name as from user during time and recovers that table if it is dropped accidently. 11. If all documentation is over, delete the Test_Report table, take care no one can retrieve it back