**Chapter 1 problems - Database systems**

-- Q1

-- Count of records in file

1. Select count(1) from PROB\_1\_01;

-- Number of fields in file is 5

1. select \* from PROB\_1\_01;

-- Q2

The problem is field city is part of the manager\_address. For E.g. for the record with project code '21-5Z',the address is '3334 Lee Rd., Gainesville, FL 37123', which is present in the city 'Gainesville'.

Dividing the field 'Manager\_address' into 'Address\_line\_No\_1','Address\_line\_No\_2', 'City', 'State' and 'ZipCode' would be the best option.

Another alternative is getting the ZIP to City mapping and joining it with right(trim(Manager\_address,5)) would give us the city of each address, because zip code is unique to each city

--Q3

I would make the following changes to the file

'PROJECT\_MANAGER' - 'FIRST\_NAME','MIDDLE\_NAME','LAST\_NAME' -- Also would ensure the table is more than 100 characters to ensure no loss of data in name

'MANAGER\_PHONE' - 'AREA\_CODE','PREFIX\_NUMBER','LINE\_NUMBER' -- The columns have to be INT variables

'MANAGER\_ADDRESS' - 'Address\_line\_No\_1','Address\_line\_No\_2','City','State','ZipCode'

--Q4

We see that Holly B. Parker and George F. Dorts have 3 and 2 separate projects respectively. So their manager information is repeated in the dataset. The solutions for this is to have two tables: One with the Manager info and one with the project info (which includes Manager ID or Manager Name).

Another would be Manager Name could be entered wrongly which might result in duplicates.

--Q5

The table has `PROJ\_NAME`, `EMP\_NAME`, `JOB\_CH0\_HOUR` and `EMP\_PHONE` have repeated multiple times in the given table as there could have been other foreign keys from different tables such as ‘PROJ\_NUM’,’ EMP\_NUM’ and ‘JOB\_CODE’.

--Q6

EMP\_NAME: Could be divided into `FIRST\_NAME`, `MIDDLE\_NAME` and `LAST\_NAME`.

EMP\_PHONE: Could be divided into `AREA\_CODE`, `EXCHANGE\_NUMBER` and `LINE\_NUMBER`.

-- Q9

`TEACHER\_FNAME`, `TEACHER\_LNAME` AND `TEACHER\_INITIAL` repeats every time with repeating teachers time slot. Adding a column called `TEACHER\_EMP\_CODE` to simplify the table and add another table with Teacher info with the mapping information.

**Chapter 2 problems - Database models**

-- Q1

`AGENT\_CODE` is a foreign key in the CUSTOMER table which gives us the agent information for each transaction on insurance details. It is a 1:M relationship between AGENT and CUSTOMER.

--Q2

--Q3

--Q4

One Store can only have one region but there could be multiple stores in one region. Therefore, the relationship between REGION and STORE is 1: M.

One Store can have multiple employees but usually one employee can only be in one store. Therefore, the relationship between STORE and EMPLOYEE is 1: M.

JOB\_CODE refers to Job info and can be specific to a specific role in the store assigned to an employee. For example, there can be more than one janitor in a store. Therefore, the relationship between JOB and EMPLOYEE is 1: M. (Note: Assuming each employee has a only one role to attend to at the store)

-- Q5

PROBLEM 1:

1. -- Faculty table
2. SELECT \* FROM Faculty;
3. -- Course table
4. SELECT \* FROM Course;
5. -- Offering table
6. SELECT \* FROM Offering;
7. -- Student table
8. SELECT \* FROM Student;
9. -- Enrollment table
10. SELECT \* FROM Enrollment;

PROBLEM 2:

1. -- PART 1
2. SELECT StdNo,StdFirstName, StdLastName
3. FROM Student
4. ORDER BY StdLastName ASC , StdFirstName ASC;
5. -- PART 2
6. SELECT StdNo,StdFirstName, StdLastName
7. FROM Student
8. ORDER BY StdLastName , StdFirstName;

PROBLEM 3:

1. – PART 1
2. SELECT StdLastName,StdFirstName,StdGPA
3. FROM Student
4. ORDER BY StdGPA DESC, StdLastName , StdFirstName;
5. -- PART 2
6. SELECT StdLastName,StdFirstName,StdGPA,
7. rank() over (order by StdGPA desc) as student\_rank
8. FROM Student
9. ORDER BY student\_rank, StdLastName , StdFirstName;

PROBLEM 4:

1. --PART 1
2. SELECT StdCity,StdState
3. FROM Student;
4. --PART 2
5. SELECT DISTINCT StdCity,StdState
6. FROM Student;

PROBLEM 5:

1. SELECT StdLastName,StdFirstName, StdGPA
2. FROM Student
3. WHERE StdGPA > 3.2;

PROBLEM 6:

1. SELECT StdLastName,StdFirstName, StdGPA
2. FROM Student
3. WHERE (StdGPA > 2.2 AND StdGPA < 2.7) OR (StdGPA > 3.2 AND StdGPA < 3.8);

PROBLEM 7:

1. SELECT StdLastName,StdFirstName, StdGPA
2. FROM Student
3. WHERE StdGPA BETWEEN 2.7 AND 3.2;

PROBLEM 8:

1. SELECT OfferNo, CourseNo, OffYear, FacNo
2. FROM Offering
3. WHERE FacNo IS NULL;