Case Study Report

Consulting Services Database

Group-7

Sl. No	Name	NUID
1.	Nishith Burman	001896660
2.	Gaurav Handa	001305563

Guided By

Prof. Xuemin Jin
Associate Teaching Professor
Mechanical and Industrial Engineering

INTRODUCTION

The main reason why we selected this topic for the case study is due to the fact that we wanted to understand the working of the consulting service industry. Apart from this, no work has been previously done before on this topic. So we would have to understand the business problem by ourselves and build a database from scratch and we were able to learn a lot about each and every aspect of designing a database.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project advisor Prof. Xuemin Jin for continuously supporting us throughout the preparation phase of our database. We are grateful to him for his patience and for guiding us in our research through his immense knowledge His guidance helped us in all the time of our research and in writing this case study. We would not have imagined having a better professor and mentor for our case study project.

We would also like to thank our Teaching Assistant Nidhi Saraf for her continuous support throughout the semester. We are grateful to her for helping us with our question related to our coursework and assignment

PROBLEM

- Background:

  Meridian Studies is on the first Educational Consulting Startup that was recently started in the city of Raipur. Initially, it started with just 1 client. But this number gradually increased year after year due to the high performance of the company. Because of their really amazing consulting services, Meridian Studies is now considered as one of the best educational consulting startups of our state

- Problem Statement:

  Recently Meridian Studies saw a tremendous increase in their client base. They were also able to open 3 new offices in three different locations. They also added a lot of new employees in the past year and there are lots of transactions that are taking place each and every week. Initially, when the start-up was founded, the owner and some of the employees used to keep track of everything by recording the events in separate files. But now it has become nearly impossible for the owner and other employees to keep track of each and every activity that is happening on a daily basis in a systematic and concise manner. So the owner has requested us to design a database capable of storing comprehensive information about each and every aspect of their business.

- Objective:

  The objective of this project is to design a fully functional database capable of storing information regarding each and every aspect of the business as requested by the owner of the company. We also plan to learn more about the business as it would help us design the database in a more appropriate manner.

- Business Problem Addressed

  I. Transferring the data between different branches of meridian studies and other supporting organizations was really cumbersome. This issue is solved by having a single industry-standard database.

  II. Initially analyzing the paper-based data was nearly impossible because it would involve manual calculation and the large volume of data. This issue is also solved as the company can use different tools and software's to analyze the electronic data and use the result for effectively driving the business.

  III. Initially, when the company was founded, data used to be quite unorganized and so gaining any insights from the data at that time was really difficult. But in this database everything is pretty organized so gaining insight is fairly easier and thus these insights can be used to provide better services to our customers.

SOLUTION DESIGN

- Understanding the Business Requirement:

  The first and one of the most important steps in designing a database is understanding the business and understanding the requirement of the business. If we have a proper understanding of these two aspects we will be able to incorporate each and every aspect of business required to run it in our design. So we did a lot of research about the educational consulting industry and we also discussed with the owner of the Meridian Studies in detail about the requirement of the business.

- Business Rules:

  I. One student can study in N number of universities before pursuing their higher education. E.g. One can study Bachelors from NIT, Warangal, and Masters from IIT, Bombay.
  II. One service provider can provide its services to only one of the four branches based on the requirement but the branches can have N number of service providers.
  III. 1 Students can study in only one of the centers and a center can have N number of students.
  IV. Different banks are associated with each of the centers to provide financial supports to the student who needs it to pursue their education. Each center can be associated with N number of banks.

- **The requirement of the Database:**

  Based on our research and discussion with the owner we got to know that the database should mainly contain lots of different kinds of information. It should contain information about our Clients, their education and experience details, their scores in different examinations, their family background, their contact details, and their admission details, etc. It should also contain information about employees, branches, other associated organizations, and transaction details.

- Approach:

  After gathering information about the business requirement, the next and most difficult part was to connect the dots between different components (entity types) of the business. Once we figured the relationship between different Entities, we then designed the EER and UML model using Lucid Chart and Toad data Modeller. And after that, we populated our model in MySQL. We also Connected our databases to python and created some of our tables in MongoDB.

- Tools Used:

  1.) Data Modelling: Lucid Chart, Visio, Toad Data Modeler
  2.) Relational Database: MySQL
  3.) NoSQL Database : MongoDB
  4.) Other Languages: Python (Jupyter Notebook)

- **Design Requirement**

Entity Type	Why it is required
1.) Branch Details	This entity type contains all the information about the different branches of meridian studies like the Name of a particular branch, address and contact details of that branch.
2.) Clients details	This entity type is central to our database design and one of our primary purposes to design this database is to keep track of all the necessary information of our client(which are mainly Student). This entity type has each and every information regarding our clients like their name, date of birth, gender, etc.
3.) Employees	This entity type contains all the necessary information of the different employees that are working in the Meridian Studies like their name, contact details, position, salary, etc. This entity type is extremely important as it could be used to retrieve information About our employees
4.) Associated Comp.	To run a business successfully a company needs support from many other companies. That is the reason why this entity type is included in this database. It contains information about different companies E.g. their name, the kind of service they provide, their contact details and their address, etc.

5.) Bank Details	Many students who pursue their master's take education loans to fulfill their tuition and other requirements. That is the reason why this entity type has been included in the database. This entity type contains all the information about different banks that provide education loans to the student. It contains Information like Bank name, address, contact detail, manager name, etc.
6.) Test Scores, College, Company Details	These are some of the entity types that are also important and information of these entity types can be used for the analytical purpose. For instance, it can be used to find the average GRE scores to get admission into any particular university
7.) Application Details	It is included in order to keep track of all the students who get admit or reject decision from a particular university, for a particular program

CONCEPTUAL DATA MODELING (RELATIONSHIP B/W ENTITIES)

Entity-1	Relationship Type	Entity-2
Branch	1:N 1:N 1:N 1:N 1:N	Banks Partner Companies Transaction 1 Employees Client
GRE TOEFL GMAT IELTS	1:N 1:N 1:N 1:N	Client Client Client Client
Banks Client Contact Details	1:N 1:1	Client Client
Clients	M:N M:N M:N	College Companies Worked Grad School

MAPPING CONCEPTUAL MODEL TO RELATIONAL MODEL

Table: banks

Columns:

Bank_Id	int(11) PK
Bank_Name	char(20)
Bank_Address	char(20)
Phone_No	int(11)
Loan_Amount	int(11)
Loan_Type	char(20)
Center_id	int(11) PK

Table: client

Columns:

Student_Id	int(11) PK
Student_Name	char(20
Gender	char(20
DOB	date
Bank_Id	int(11)
Center_id	int(11)
Examiner_Id	int(11)
Examiner_Id1	int(11)
Examiner_Id3	int(11)
Examiner_Id2	int(11)

Table: college

Columns:

Clg_Id	int(11) PK
Clg_Name	char(20)
Level	char(20)
Major	char(20)
GPA	float
Start_Year	date
End_Year	date
Clg_Address	char(20)

Table: companies_worked

Columns:

Id	int(11) PK
Name	char(20)
Address	char(20(
Contact	double
Position	char(20(
Supervisor	char(20)
Total_Duration	float

Table: employees

Columns:

Employee_id	int(11) PK
Employee_Name	char(2(
Employee_Email	char(2(
Employee_Contact	double
Position	char(2(
Salary	int(11)
Employee_Address	char(2(
Center_id	int(11) PK

Table: gre_scores

Columns:

Examiner_Id	int(11) PK
Location	char(20)
Exam_Date	date
Verbal_Score	int(11)
Quant_Score	int(11)
AWA_Score	float
Total_Score	int(11)

Table: gmat_scores

Columns:

Examiner_Id3	int(11) PK
Location	char(2(
Exam_Date	date
Verbal_Score	int(11)
Quant_Score	int(11)
AWA_Score	int(11)
Reasoning_Score	int(11)
Total_Score	int(11)

Table: grad_school

Columns:

School_Id	int(11) PK
School_Name	char(2(
School_Address	char(2(
Application_Status	char(2(
Major	char(2(
Tution	int(11)
Session_StartTime	char(2(

Table: partner_companies

Columns:

Com_id	int(11) PK
Com_Name	char(20
Com_address	char(20
Com_phoneno	double
Service_offered	char(20
Center_id	int(11) PK

Table: partner_companies

Columns:

Com_id	int(11) PK
Com_Name	char(20
Com_address	char(20
Com_phoneno	double
Service_offered	char(20
Center_id	int(11) PK

Table: toefl_scores

Columns:

Examiner_Id1	int(11) PK
Location	char(2(
Exam_Date	date
Reading_Score	int(11)
Speaking_Score	int(11)
Listening_Score	int(11)
Writing_Score	int(11)
Total_Score	int(11)

Table: transaction1

Columns:

Transaction_Id	int(11) PK
Sender	char(2(
Receipient	char(2(
Date	date
Amount	int(11)
Transaction_type	char(2(
Center_id	int(11) PK

Table: branch

Columns:

Center_id	int(11) PK
Center_name	char(20)
Contact	int(11)
Address	char(20)

Table: client_college_info

Columns:

Student_Id	int(11) PK
Clg_Id	int(11) PK

Table: client_company_info

Columns:

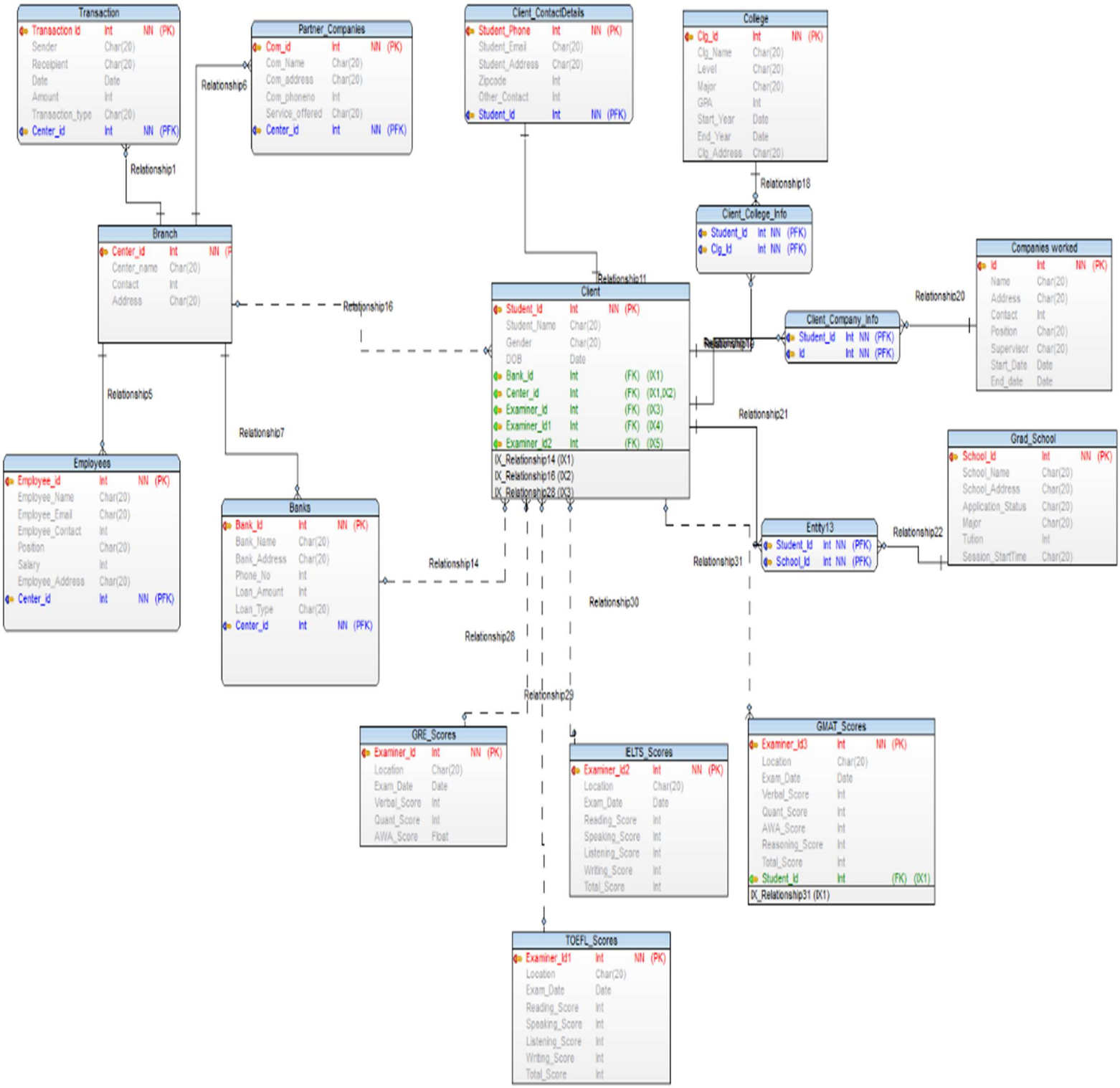
Student_Id	int(11) PK
Id	int(11) PK

Table: grad_clientdetails

Columns:

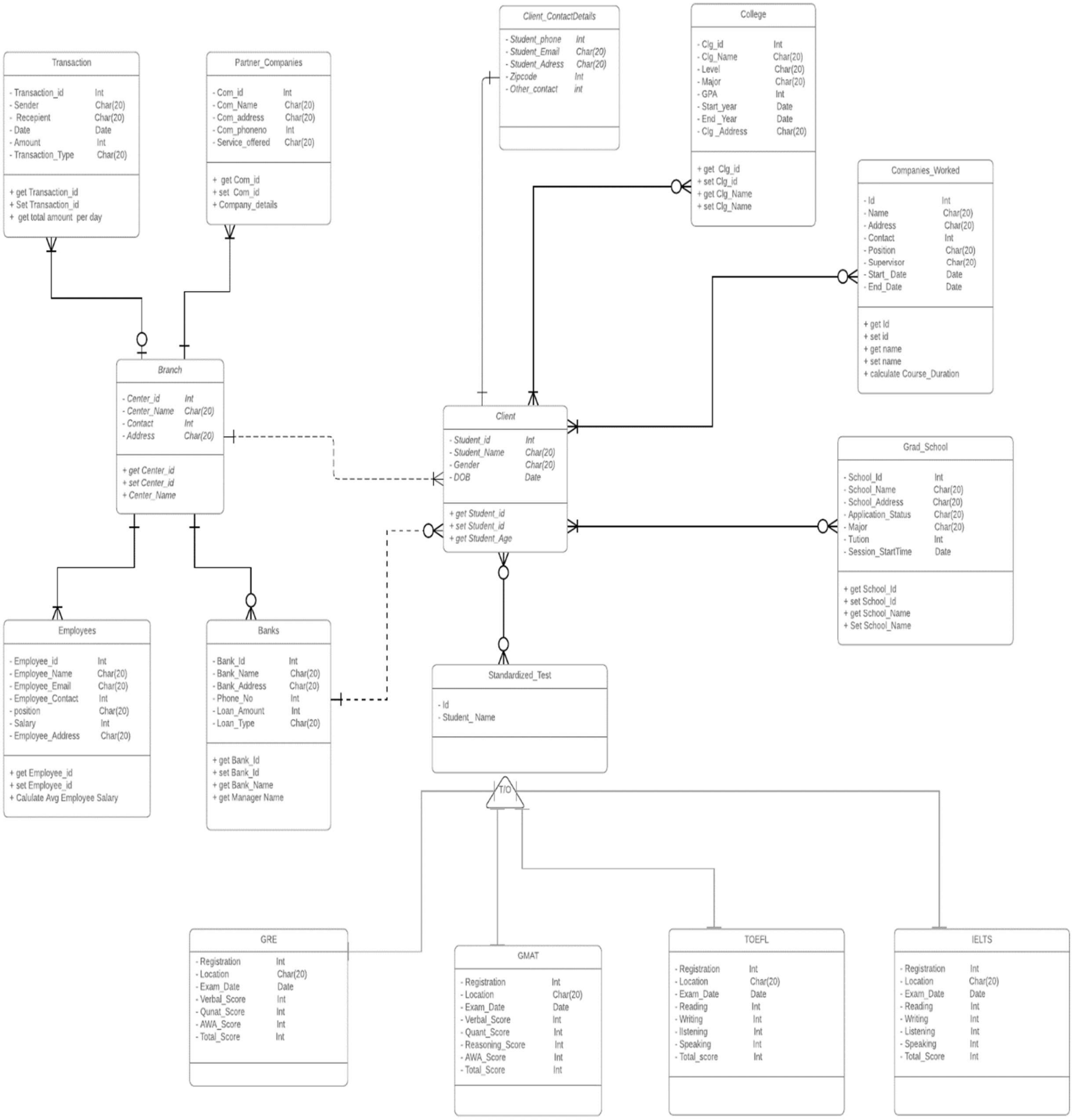
Student_Id	int(11) PK
School_Id	int(11) PK

Relational Model Schematics



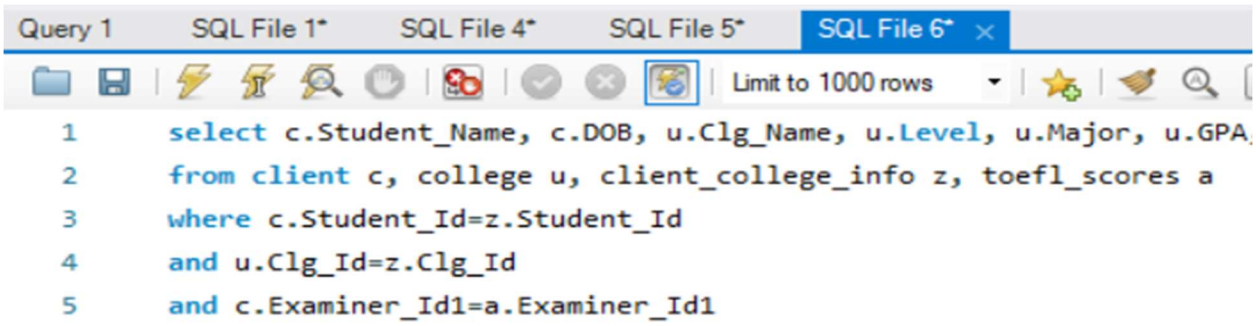
UML Class Diagram

Nishith Burman | November 1, 2019



IMPLEMENTING THE MODEL IN MYSQL

After designing the data model in Toad Data Modeler, We generated the DDL Query in the Toad Data Modeler and then we run that generated query in MYSQL workbench. This populated our databases and all the table, views and stored procedure associated with the database in MYSQL Then we manually filled records in our tables. Fig. below is a snippet of a query run on MYSQL.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Student_Name	DOB	Clg_Name	Level	Major	GPA	Toefl_score
Naruto_UzuMaki	1995-01-21	Academia	Bachelors	Electrical	3.8	107
Hinata	2000-01-19	Academia	Bachelors	ComputerSci	4	108
Neji	1998-06-22	Yosen	Masters	Economics	4	113
Ichigo	1994-09-18	Caligiri	Bachelors	Robotics	4	115
Itachi	1995-12-19	Yosen	Bachelors	Electrical	4	119

IMPLEMENTING THE MODEL IN NoSQL

We used MongoDB for NoSQL implementation. We did not populate our entire database on the NoSQL rather we just populated some of the collection in a pre-existing database(Restaurant database in our case). Fig. below is the NoSQL implementation of our database.

```
db.transaction.aggregate([
    {
      $lookup:
        {
          from: "center",
          localField: "center_id",
          foreignField: "transaction_id",
          as: "branch_transaction"

        }
    },
    { $match : { center_id : 3 } }
]).pretty();
```

ACCESSING THE DATABASE USING PYTHON

In order to query data from MYSQL in python, we are using a cursor that has been created using the connection's cursor() method. We first open a connection to the MySQL server and store the connection object in the variable. We then create a new cursor, by default a MySQLCursor object, using the connection's cursor() method.

we store the SELECT statement in the variable sql_select_Query. We then execute the operation stored in the query variable using the execute() method. After executing the query, the MySQL server is ready to send the data. Finally, we are using Fetchall to get all the records returned by the above SQL query. The figure below shows a snippet of querying data from MYSQL using the Jupyter notebook.

```
#
import mysql.connector
#
from mysql.connector import Error
#
try:
    connection = mysql.connector.connect(host='127.0.0.1',
                                         database='meridian_studies_db',
                                         user='root',
                                         password='NA$13@719212b',
                                         auth_plugin = 'mysql_native_password')
    sql_select_Query = "select c.Student_Name, c.DOB, u.Clg_Name, u.Level, u.Ma
    cursor = connection.cursor()
    cursor.execute(sql_select_Query)
    records = cursor.fetchall()
    print("Total number of rows in Laptop is: ", cursor.rowcount)
```

CONCLUSION

From a Data Scientist's perspective having a solid understanding of database design and database management is really important. This project has helped us develop a vast array of database skills from Data Modeling to Data mapping to implementing the model in MYSQL and NoSQL database to accessing the data from MYSQL using R and Python. In this process, we learned to work with different tools like Toad Data Modeler, Lucid Chart, MYSQL, MongoDB and python.