



Centralized Counselling Portal

Sharvil Vilas Turbadkar
School of Information Studies
Project Implementation Report
IST659

Project Summary

Summary:

There are millions of students in India that pursue education abroad. With Spring 2018/Fall 2019 having as much as 10-15 million students as a tentative figure the scale of augmenting student experience is huge. As of now India does not have any system in place that helps students with a centralized experience.

Many students pursuing masters must go through an arduous process of paying hefty fees to commercialized counselors that too at the cost of getting stalled and not getting an individualized experience tailor-made for the student. The student must go through many hassles and mundane loads like getting recommendation from multiple teachers without having any prior knowledge of which teachers' recommendation would boost their profile the most.

Counselling companies have multiple students under their supervision and are all governed by a manager. Thus, catering to every student's doubt and giving every student his respective university list becomes a hassle due to time constraint as there are many factors to be taken into consideration. Students must book appointments 2-3 weeks prior till which their process is stalled thus making it cumbersome for student to carry on with their formalities

Keeping legacy data to help future batches is also not in place as forms and log reports generated are backed by manual entries and making an accurate foolproof prediction for the student becomes very tough

Thus, I plan on making a database that enhances the student experience.

The solution I propose plans to make student experience hassle free by centralizing the experience gathering counsellors and students online on a single portal.

Many Counsellors and students alike who have pursued their education abroad have prior knowledge of how the formalities and system fall into place. Their knowledge if they wish on sharing can be utilized utmost by signing them up on the portal.

By centralizing many key drawbacks can be curbed and the entire process can be enhanced and expedited at the same time as students get personalized attention, counselors get their slots booked automatically given availabilities along with getting paid depending upon the total number of hours dedicated to a student.

A student can book appointments and depending upon his profile will get reviewed by the counsellor and will be assigned universities automatically depending upon his profile.

Also, the teachers will be recommended by the counselors making the student and teacher experience more comfortable and seamless

Process

- A Counsellor signs in by filling in his necessary details.
- A student can sign in along with entering his essential details like his Name, Contact info, Academic Profile, Semester of choice, Major
- Upon evaluating his profile, the counsellor will suggest the student teachers to take recommendations from.
- After a teacher gets recommended, she can approve and sign the documents electronically by logging in and writing a letter of recommendation thus saving resources and time
- A counsellor then writes a Sop and reviews the student's profile

- After Reviewing the entire profile, the counsellor suggests universities to apply to along with prescribing the major
- A counsellor tends to the student's needs by answering his questions and helping him in his other formalities like university registration, VISA and OFC, Academic Doubts
- A Counsellor will book a slot depending upon available slots and the slot time update automatically without manual intervention
- A Counsellor will get paid depending upon the total hours he has dedicated to a student
- After student gets placed successfully, he gives a review for his respective counsellor
- This data is aggregated to review the counsellor and help further batch of students

Tables and Attributes

<u>DATAOBJECT:</u> Student Counselling	This database contains all the tables and relations that together build the Centralized Counselling Database.
Student	Stores Student Essential Details
Student ID Student Name Student Mobile Number Student Address	PRIMARY KEY: Each Student will have a unique ID to identify him from the rest such that it is easier for counsellors to keep track of students to
Counsellor	Stores counsellor details
Counsellor ID	PRIMARY KEY: Each Counsellor will have a unique ID to identify him uniquely Stores Counsellor Details
Counsellor Name	Name of the Counsellor
Counsellor Type	Type of Counsellor: Take two values 1) Alumni: Has been alumni of a previous graduate university 2) Professional: Has had professional expertise

Counsellor Mobile	Mobile number of counsellors
Counsellor Office	Office address of counsellor
Feedback	<p>Every Counsellor is given a feedback by a student</p> <p>These feedbacks are aggregated for future uses which will make selecting a counsellor easier for the upcoming batches</p> <p>Primary key: Uniquely identifies each review given by a student to a counsellor by a Review ID</p>
Review ID	Review ID is a surrogate key that breaks many to many relationships between Counsellor and student tables
Student ID	Foreign Key: Uniquely identifies the student
Counsellor ID	Foreign Key: Uniquely identifies the Counsellor
Counsellor Rating	Rating given by student to counsellor
Teacher	Stores details of teacher who would be giving the LOR
Teacher ID	Primary Key: Every teacher has an ID that uniquely identifies it

Teacher Name	Name of the teacher
Teacher Email	Email ID of the teacher
Subjects Taught	Subjects taught by the teacher for counsellor to get estimate which teacher to recommend to the student for getting LOR
Letter of Recommendation	<p>Every teacher gives a letter of recommendation to a student</p> <p>Foreign Key: Associated with primary key of Student and teacher table</p> <p>The cardinality is zero o more as it is up to the teacher to give his/her recommendation</p>
Teacher ID Counsellor ID	Teacher ID and Counsellor ID act as surrogate keys
Length	Length of LOR is the associative entity that breaks many to many relationships between Teacher and counsellor
University	Stores information about the university recommended by the counsellor
University ID	PRIMARY KEY: Each University has a unique ID that Identifies it

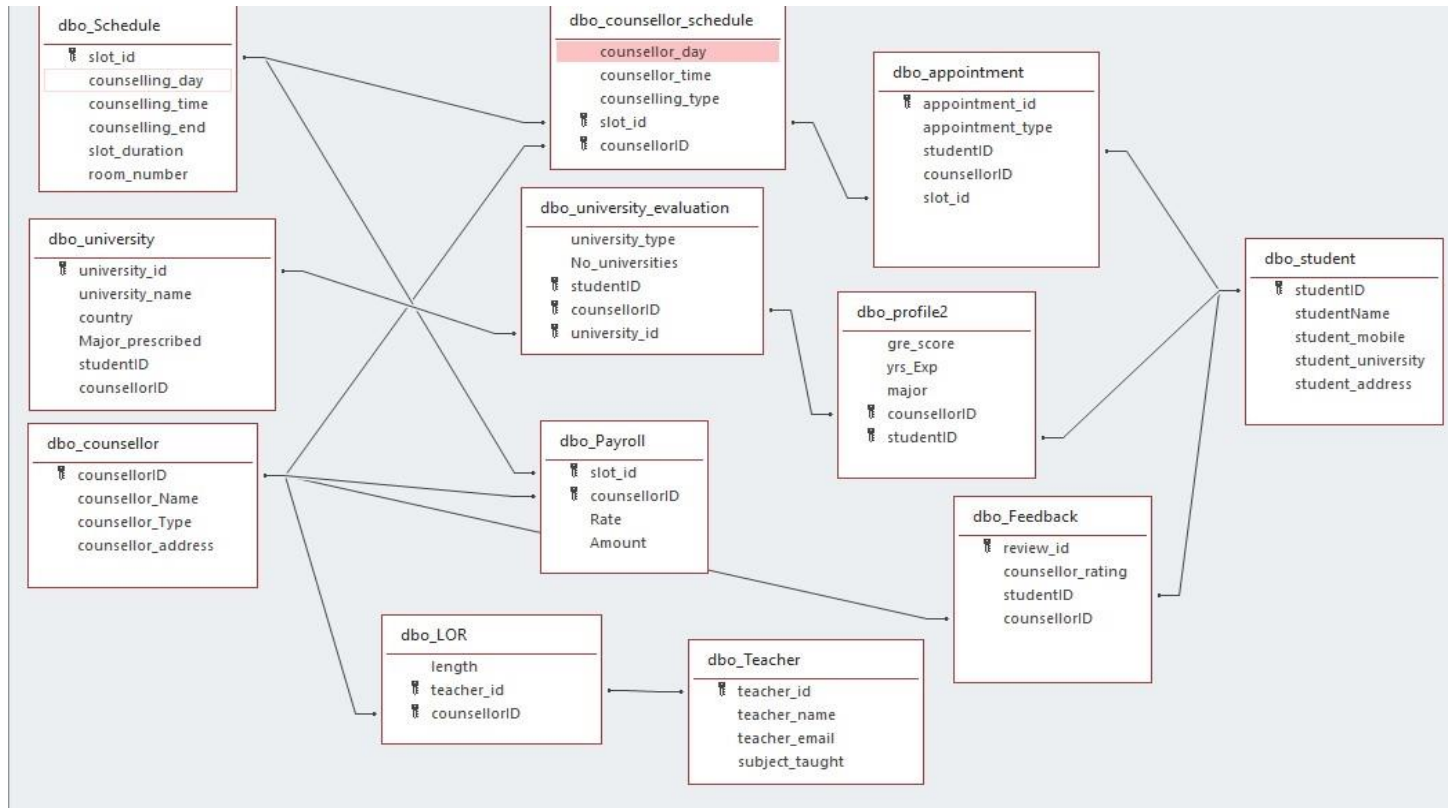
Student ID Counsellor ID	Foreign Key: Student ID and Counsellor ID act as foreign keys as each counsellor reviews a student's academic profile and suggests universities
Country	The country the university is in
University Name	The name of the university
Major prescribed	The Majors prescribed by the counselor in that university
Academic Profile	Contains Academic information of the Student
Student ID Counsellor ID	Foreign Key: Student ID and Counsellor ID are used as foreign keys to uniquely identify an identifying relationship as student and counsellor ID can never be NULL Contains Academic profile of the student
GRE Score	GRE Score of the student
Years of exp	Number of years the student has worked prior coming to the university
Major Pursing	Major the student wants to pursue

University Evaluation	In this entity after the counsellor evaluates the academic profile after which he selects universities for the student
Student ID Counsellor ID University ID	Student ID, Counsellor ID, University ID act as composite keys by breaking many to many relationships between academic profile of student and the university entity
University Type	University Type takes three values Ambitious Moderate and safe and these are assigned automatically depending upon the student's profile Number of universities to be assigned can be decided by the student after discussing with the counsellor
Number of universities	Number of universities the student applies to
Schedule	The Portal has a pre-existing schedule in place where slots are assigned for different counselling sessions during which the counselor cannot tend to other students and a student cannot get counselled by two counselors at once
Slot id	Primary Key: Uniquely identifies the slot assigned by the counsellor to the student Acts as surrogate key
Room Number	Room Number pertains to the room where the appointment will take place

Slot Duration	Slot Duration is the total time duration the appointment was slotted for
Slot Day	The day the slot has been assigned
Slot time	The time the slot starts
Appointment	Every student will book an appointment depending upon the available slot. A booking ID is generated to log the appointment
Appointment ID	Primary Key: Uniquely identifies when the appointment scheduled Acts as a surrogate key
Student ID Counsellor ID Slot ID	Student ID, Counsellor ID, Slot ID act as foreign keys by breaking many to many relationships between counsellor schedule entity and the student entity
Appointment type	The type of appointment scheduled
Counsellor Schedule	An entity that contains the Time schedule of every counsellor
Counsellor ID Slot ID	Composite keys that break many to many relations between counselor entity and Schedule entity
Counselling Day	The day the counsellor has Appointment with the student for that respective slot

Counselling time	The time when the Appointment begins for that respective slot assisted by the counselor
Counselling End	The time when the Appointment ends
Counselling Type	Type of counselling takes certain Values Academic, VISA, SOP, OFC, LOR, Academic, Miscellaneous
Payroll	Depending upon the total hours allotted for counselling the payroll is assigned
Counsellor ID Slot ID	Composite keys that break many to many relations between counselor entity and Schedule entity
Slot Duration	Number of hours the slot was assigned to the student by the counsellor
Payroll	The total amount the counsellor gets paid for his services

Access Entity Relationship Diagram



Business rules:

1. Every Counsellor must attend an appointment he schedules and abide by the slot duration mentioned
2. Every Student present in the system must belong to a valid University and must have valid credentials and marks uploaded to back up his academic profile
3. Similarly, a counsellor must have his/her qualifications uploaded for him/her to be enrolled as a counsellor
4. A University Type can take only three values: A[Ambitious],M[Moderate],S[Safe]
5. A Counsellor cannot view other counsellor's rating but can view her own rating
6. Counsellor Type can be a professional counsellor or an alumnus of a grad school
7. Counselling type can take only values VISA, SOP, OFC and Misc.
8. Student will get to finally approve the university from the list given
9. Amount is calculated by multiplying the slot duration by hourly pay of counsellor

SQL script for creating tables and inserting sample data.

Creating Student table

```
CREATE TABLE student(
  studentID NUMERIC(5,0) PRIMARY KEY,
  studentName varchar(50)not null,
  student_mobile numeric(11,0) not null,
  student_university VARCHAR(30) NOT NULL,
  student_address VARCHAR(30) ,
);
```

Inserting Values in student table

```
INSERT INTO student VALUES (10,'Sharvil',3158898675,'VJTI','Thane');
INSERT INTO student VALUES (11,'Hitesh',31588985689,'Bhopal University','Ramesh Street');
INSERT INTO student VALUES (12,'Dhiraj',31588985689,'Kolkata University','Madal Avenue');
INSERT INTO student VALUES (13,'Aseem',31588985689,'Delhi University','Red Fort');
INSERT INTO student VALUES (14,'Rohit', 31588947927,'Mumbai University','Seawoods');
INSERT INTO student VALUES (15,'Kunal', 31588978468,'Mumbai University','Chembur');
```

select * from student;

studentID	studentName	student_mobile	student_university	student_address
10	Sharvil	3158898675	VJTI	Thane
11	Hitesh	31588985689	Bhopal University	Ramesh Street
12	Dhiraj	31588985689	Kolkata University	Madal Avenue
13	Aseem	31588985689	Delhi University	Red Fort
14	Rohit	31588947927	Mumbai University	Seawoods
15	Kunal	31588978468	Mumbai University	Chembur

Creating Counsellor table

```
CREATE TABLE counsellor(
  counsellorID NUMERIC(5,0) PRIMARY KEY,
  counsellor_Name VARCHAR(200) NOT NULL,
  counsellor_Type char(20) check(counsellor_Type in ('Professional','Alumni')),
  counsellor_address VARCHAR(30) not null
);
```

Inserting Values in student table

```
INSERT INTO counsellor VALUES (10, 'Vera', 'Professional', '118 Concord Place');
INSERT INTO counsellor VALUES (11, 'Gibbs', 'Professional', '112 Harvard Place');
INSERT INTO counsellor VALUES (12, 'Angela', 'Professional', '8 Marshall Street');
INSERT INTO counsellor VALUES (13, 'Marsha', 'Professional', 'South Campus');
INSERT INTO counsellor VALUES (14, 'Hershell', 'Alumni', '128 Armory Sqaure');
```

```
select * from counsellor;
```

Results		Messages	
counsellorID	counsellor_Name	counsellor_Type	counsellor_address
10	Vera	Professional	118 Concord Place
11	Gibbs	Professional	112 Harvard Place
12	Angela	Professional	8 Marshall Street
13	Marsha	Professional	South Campus
14	Hershell	Alumni	128 Armory Sqaure

Creating Profile table

```
CREATE TABLE profile(
  gre_score int not null,
  yrs_Exp int ,
  major varchar(20) not null,
  counsellorID NUMERIC(5,0) FOREIGN KEY REFERENCES counsellor(counsellorID),
  studentID NUMERIC(5,0) FOREIGN KEY REFERENCES student(studentID),
  CONSTRAINT profile_pk PRIMARY KEY (studentID, counsellorID)
);
```

Inserting Values in profile table

```
INSERT INTO profile VALUES (312,4,'Computer',10,10);
INSERT INTO profile VALUES (300,1,'Electrical',10,11);
insert into profile values(324,0,'Computer',11,12);
insert into profile values(290,0,'Environment',12,13);
insert into profile values(304,0,'Arts',10,14);
insert into profile values(303,1,'Chemistry',12,15);
```

```
select * from profile;
```

Results		Messages		
gre_score	yrs_Exp	major	counsellorID	studentID
312	4	Computer	10	10
300	1	Electrical	10	11
324	0	Computer	11	12
290	0	Environment	12	13
304	0	Arts	10	14
303	1	Chemistry	12	15

Creating University table

```
create table university(
  university_id numeric(4,0) primary key,
  university_name varchar(50) not null,
  country varchar(20) not null,
  Major_prescribed varchar(20) not null,

  studentID NUMERIC(5,0) FOREIGN KEY references student(studentID),
  counsellorID NUMERIC(5,0) FOREIGN KEY REFERENCES counsellor(counsellorID),
);
```


Inserting Values in university table

```
INSERT INTO university VALUES (101,'NYU','USA','ADS',10,10);
INSERT INTO university VALUES (102,'Rochester University','USA','ADS',10,10);
INSERT INTO university VALUES (103,'AAachen University','Germany','Automobile',14,10);
INSERT INTO university VALUES (104,'Syracuse','USA','IM',11,10);
INSERT INTO university VALUES (105,'Berlin University','Germany','Automobile',12,14);
INSERT INTO university VALUES (106,'Monash University','Australia','Computer Engineering',14,10);
```

```
select * from university;
```

Results

Messages

university_id	university_name	country	Major_prescribed	studentID	counsellorID
101	NYU	USA	ADS	10	10
102	Rochester University	USA	ADS	10	10
103	AAachen University	Germany	Automobile	14	10
104	Syracuse	USA	IM	11	10
105	Berlin University	Germany	Automobile	12	14
106	Monash University	Australia	Computer Engineering	14	10

Creating university evaluation table

```
create table university_evaluation(
university_type char(50) check(university_type in('Ambitious','Moderate','Safe')),
No_universities numeric(5,0),
studentID NUMERIC(5,0) foreign key references student(studentID),
counsellorID numeric(5,0) foreign key references counsellor(counsellorID),
university_id numeric(4,0) foreign key references university(university_id),
constraint university_evaluation_pk primary key(studentID,counsellorID,university_id)
);
```

Inserting Values in university evaluation table

```
insert into university_evaluation values('Ambitious',null,10,10,101);
insert into university_evaluation values('Safe' ,null,14,10,106);
insert into university_evaluation values('Safe' ,null,14,10,102);
insert into university_evaluation values('Moderate' ,null,11,11,103);
insert into university_evaluation values('Moderate' ,null,11,11,104);
```

```
select * from university_evaluation ;
```

university_type	No_universities	studentID	counsellorID	university_id
Ambitious	NULL	10	10	101
Moderate	NULL	11	11	103
Moderate	NULL	11	11	104
Safe	NULL	14	10	102
Safe	NULL	14	10	106

Creating Feedback table

```
create table Feedback(
  review_id numeric(4,0) primary key,
  counsellor_rating int not null,

  studentID NUMERIC(5,0) foreign key references student(studentID),
  counsellorID numeric(5,0) foreign key references counsellor(counsellorID)
)
```

Inserting values in Feedback Table

```
insert into Feedback values(1000,9,10,10);
insert into Feedback values(1001,7,14,10);
insert into Feedback values(1002,4,10,11);
insert into Feedback values(1003,2,11,11);
insert into Feedback values(1004,2,12,12);
insert into Feedback values(1005,7,15,11);
insert into Feedback values(1006,10,14,13);
insert into Feedback values(1007,8,12,11);
```

```
select * from Feedback ;
```

review_id	counsellor_rating	studentID	counsellorID
1000	9	10	10
1001	7	14	10
1002	4	10	11
1003	2	11	11
1004	2	12	12
1005	7	15	11
1006	10	14	13
1007	8	12	11

Creating Teacher table

```
create table Teacher(
  teacher_id numeric(5,0) primary key,
  teacher_name varchar(50) not null,
  teacher_email varchar(50) not null,
  subject_taught varchar(50) not null
)
```

Inserting values in Teacher Table

```
insert into Teacher values(1, 'Madhuri', 'madhuri@gmail.com', 'Digital Signals');
insert into Teacher values(2, 'Nidhi', 'nidhi01@gmail.com', 'ERP');
insert into Teacher values(3, 'Ingle', 'ingle@gmail.com', 'Computer vision');
insert into Teacher values(4, 'Rahul', 'rpatil@gmail.com', 'Intro to C++');
insert into Teacher values(5, 'Prerna', 'pren@gmail.com', 'SPCC');
```

```
select * from teacher;
```

Results Messages

teacher_id	teacher_name	teacher_email	subject_taught
1	Madhuri	madhuri@gmail.com	Digital Signals
2	Nidhi	nidhi01@gmail.com	ERP
3	Ingle	ingle@gmail.com	Computer vision
4	Rahul	rpatil@gmail.com	Intro to C++
5	Prema	pren@gmail.com	SPCC

Creating LOR table

```
create table LOR(
  length int,
  teacher_id numeric(5,0) foreign key references Teacher(teacher_id),
  counsellorID numeric(5,0) foreign key references counsellor(counsellorID)
  constraint LOR_pk primary key(teacher_id,counsellorID)
)
```

Inserting values in LOR Table

```
insert into LOR values(1000,1,10);
insert into LOR values(1001,2,11);
insert into LOR values(1002,3,10);
insert into LOR values(1003,4,11);
insert into LOR values(1004,1,12);
insert into LOR values(1005,2,13);
```

```
select * from LOR;
```

length	teacher_id	counsellorID
1000	1	10
1004	1	12
1001	2	11
1005	2	13
1002	3	10
1003	4	11

Creating Schedule table

```
create table Schedule(
slot_id numeric(5,0) primary key,
counselling_day date default getdate() not null ,
counselling_time datetime not null,
counselling_end datetime ,
slot_duration int not null,
room_number int
)
```

Inserting values in Schedule Table

```
insert into Schedule values(900,'12/12/2017','12/12/2017 10:00:00',null,2,201);
insert into Schedule values(901,'12/13/2017','12/13/2017 13:00:00',null,4,201);
insert into Schedule values(902,'11/6/2017','11/6/2017 13:00:00',null,2,201);
insert into Schedule values(903,'12/1/2018','12/1/2018 13:00:00',null,1,201);
insert into Schedule values(904,'11/6/2019','11/6/2019 15:00:00',null,5,201);
insert into Schedule values(905,'2/15/2018','2/15/2018 3:00:00',null,6,301);
insert into Schedule values (906,'6/11/2017','6/11/2017 7:00:00',null,13,301)
insert into Schedule values (907,'1/11/2018','1/11/2018 17:00:00',null,13,301)
```

`select * from Schedule;`

slot_id	counselling_day	counselling_time	counselling_end	slot_duration	room_number
900	2017-12-12	2017-12-12 10:00:00.000	NULL	2	201
901	2017-12-13	2017-12-13 13:00:00.000	NULL	4	201
902	2017-11-06	2017-11-06 13:00:00.000	NULL	2	201
903	2018-12-01	2018-12-01 13:00:00.000	NULL	1	201
904	2019-11-06	2019-11-06 15:00:00.000	NULL	5	201
905	2018-02-15	2018-02-15 03:00:00.000	NULL	6	301
906	2017-06-11	2017-06-11 07:00:00.000	NULL	13	301
907	2018-01-11	2018-01-11 17:00:00.000	NULL	13	301

Creating Payroll table

```

create table Payroll(
  slot_id numeric(5,0) foreign key references Schedule(slot_id),
  counsellorID numeric(5,0) foreign key references counsellor(counsellorID)
  constraint Payroll_pk primary key(slot_id,counsellorID),

  Rate numeric(5,2) not null,
  Amount numeric(5,0) not null,
)

create table Schedule(
  slot_id numeric(5,0) primary key,
  counselling_day date default getdate() not null ,
  counselling_time datetime not null,
  counselling_end datetime ,
  slot_duration int not null,
  room_number int
)

```

Inserting values in Payroll Table

```

insert into Payroll values(900,10,11,100);
insert into Payroll values(901,10,11,20);
insert into Payroll values(902,12,11,200);
insert into Payroll values(903,12,11,190);
insert into Payroll values(904,13,11,120);
insert into Payroll values(905,14,11,110);
insert into Payroll values(906,13,11,10);

```

Random Values have ben inserted in the amount section for the purpose of displaying Counsellor Payroll Report. This discrepancy will be addressed to in the trigger section

```
select * from Payroll;
```

slot_id	counsellorID	Rate	Amount
900	10	11.00	0
901	10	11.00	0
902	12	11.00	0
903	12	11.00	0
904	13	11.00	0
905	14	11.00	0
906	13	11.00	0

Creating Counsellor Schedule table

```

create table counsellor_schedule
(
counsellor_day varchar(50) not null,
counsellor_time datetime not null,
counselling_type varchar(10) check(counselling_type in ('VISA','SOP','OFC','LOR','Academic','Misc')),

slot_id numeric(5,0) foreign key references Schedule(slot_id),
counsellorID numeric(5,0) foreign key references counsellor(counsellorID),
constraint counsellor_schedule_pk primary key(slot_id,counsellorID)
)

```

Inserting values in Counsellor Schedule Table


```

insert into counsellor_schedule values('Monday','12/12/2017 10:00:00','VISA',900,10);
insert into counsellor_schedule values('Tuesday','2/21/2018 10:00:00','Academic',901,11);
insert into counsellor_schedule values('Monday','12/12/2017 10:00:00','Academic',902,12);
insert into counsellor_schedule values('Friday','2/14/2019 10:00:00','SOP',903,10);
insert into counsellor_schedule values('Friday','12/3/2017 10:00:00','VISA',904,11);
insert into counsellor_schedule values('Tuesday','12/27/2018 10:00:00','SOP',905,12);
insert into counsellor_schedule values('Thursday','5/2/2017 10:00:00','Misc',906,13);
insert into counsellor_schedule values('Thursday','6/21/2017 10:00:00','Academic',907,14);

```

```
select * from counsellor_schedule;
```

counsellor_day	counsellor_time	counselling_type	slot_id	counsellorID
Monday	2017-12-12 10:00:00.000	VISA	900	10
Tuesday	2018-02-21 10:00:00.000	Academic	901	11
Monday	2017-12-12 10:00:00.000	Academic	902	12
Friday	2019-02-14 10:00:00.000	SOP	903	10
Friday	2017-12-03 10:00:00.000	VISA	904	11
Tuesday	2018-12-27 10:00:00.000	SOP	905	12
Thursday	2017-05-02 10:00:00.000	Misc	906	13
Thursday	2017-06-21 10:00:00.000	Academic	907	14

Creating Appointment table

```

create table appointment
(
  appointment_id numeric(5,0) primary key ,
  appointment_type varchar(50),

  studentID NUMERIC(5,0) foreign key references student(studentID),
  counsellorID numeric(5,0) foreign key references counsellor(counsellorID),
  slot_id numeric(5,0) foreign key references Schedule(slot_id)
)

```

Inserting values in Appointment Table

```

insert into appointment values(1000,'NULL',10,10,900);
insert into appointment values(1001,'NULL',11,11,901);
insert into appointment values(1002,'NULL',12,10,902);
insert into appointment values(1003,'NULL',10,12,903);
insert into appointment values(1004,'NULL',13,13,904);

```



```
select * from appointment;
```

appointment_id	appointment_type	studentID	counsellorID	slot_id
1000	NULL	10	10	900
1001	NULL	11	11	901
1002	NULL	12	10	902
1003	NULL	10	12	903
1004	NULL	13	13	904

Major data questions

1. Which Counsellors earn the most?
2. How does every counsellor fare in their services (Feedback)?
3. Students seek which counselling help more often
4. Which country is the most preferred for pursuing further education?
5. Which countries are hotbeds for which subjects?
6. Which counsellors are the busiest?
7. How many universities do students apply to?
8. Which teachers get recommended more for LOR's
9. How many Students have enrolled?

Reports

Report 1: Which Counsellors earn the most?

This report gives counsellors an estimate of how much they have earned throughout their course of counselling students

Can only be viewed by counsellors and admin department for paycheck to validate

Counsellor Payroll



Counsellor ID	Counsellor Name	Amount
10	Vera	120
12	Angela	390
13	Marsha	130
14	Hershell	110

Friday, December 13, 2019

Page 1 of 1

```
SELECT Sum(Payroll.Amount) AS SumOfAmount, counsellor.counsellor_Name, counsellor.counsellorID
FROM Payroll INNER JOIN counsellor ON Payroll.counsellorID = counsellor.counsellorID
GROUP BY counsellor.counsellor_Name, counsellor.counsellorID;
```

results Messages

SumOfAmount	counsellor_Name	counsellorID
120	Vera	10
390	Angela	12
130	Marsha	13
110	Hershell	14

Report 2: How does every counsellor fare in their services (Feedback)

This can be viewed only by students who want to pursue masters and join our service for next semester. This report will give the students fair idea on which counsellor to pick

Counsellor Rating



Name	Type	Address	Rating
Angela	Professional	8 Marshall Street	2
Gibbs	Professional	112 Harvard Place	5.25
Marsha	Professional	South Campus	10
Vera	Professional	118 Concord Place	8

Friday, December 13, 2019

Page 1 of 1

```
SELECT round(Avg(cast(Feedback.counsellor_rating as float)),2) AS AvgOfcounsellor_rating, counsellor.counsellor_Name,
counsellor.counsellor_Type, counsellor.counsellor_address
FROM counsellor INNER JOIN Feedback ON counsellor.counsellorID = Feedback.counsellorID
GROUP BY counsellor.counsellor_Name, counsellor.counsellor_Type, counsellor.counsellor_address;
```

AvgOfcounsellor_rating	counsellor_Name	counsellor_Type	counsellor_address
2	Angela	Professional	8 Marshall Street
5.25	Gibbs	Professional	112 Harvard Place
10	Marsha	Professional	South Campus
8	Vera	Professional	118 Concord Place

Cast is used to convert a columns data type here I casted average rating data type to float and rounded of by 2 digits

Report 3: Students seek which counselling help more often

Can only be viewed by counsellors and admin department for getting rough estimate on where to focus upon in the future batches and where students need more help with

Counselling Time Allotment



Counselling Type	Slot Duration
Academic	19
Misc	13
SOP	7
VISA	7

Friday, December 13, 2019

Page 1 of 1

```
SELECT counsellor_schedule.counselling_type, Sum(Schedule.slot_duration) AS SumOfslot_duration
FROM Schedule INNER JOIN counsellor_schedule ON Schedule.slot_id = counsellor_schedule.slot_id
GROUP BY counsellor_schedule.counselling_type;
```

Results Messages

counselling_type	SumOfslot_duration
Academic	19
Misc	13
SOP	7
VISA	7

Report 4 Which countries are hotbeds for which subjects

Can be viewed by all parties (students, counsellors, teachers and admin dept) for everyone to get a brief idea of how many students are pursuing subjects in which country

As shown Germany is a hotbed for the automobile sector. Thus, students who want to automobile can target German universities. If a student is unaware a counsellor could guide him through

Subjects Pursued by Country



Country	Major Pursuing	Count
Australia	Computer Engineering	1
Germany	Automobile	2
USA	ADS	2
USA	IM	1

Friday, December 13, 2019

Page 1 of 1

```
SELECT Count(university.country) AS CountOfcountry, university.country, university.Major_prescribed
FROM university
GROUP BY university.country, university.Major_prescribed
ORDER BY Count(university.country) DESC;
```

CountOfcountry	country	Major_prescribed
2	USA	ADS
2	Germany	Automobile
1	Australia	Computer Engineering
1	USA	IM

Forms

Form 1: To view Summarized report of Student

After you enter the student ID a summarized report of the student is generated

Student Report

Student Name

View Report



After Entering Student ID Student Report is Generated

studentID	studentName	gre_score	major	CountOfNo_uni
10	Sharvil	312	Computer	0

```
SELECT student.studentID, student.studentName, profile.gre_score, profile.major,
Count(university_evaluation.No_universities) AS CountOfNo_universities
FROM university_evaluation INNER JOIN (profile INNER JOIN student
ON profile.studentID = student.studentID) ON university_evaluation.studentID = profile.studentID
GROUP BY student.studentID, student.studentName, profile.gre_score, profile.major
having student.studentID=10
```

results Messages

studentID	studentName	gre_score	major	CountOfNo_universities
10	Sharvil	312	Computer	0

Form 2: Counsellor books slot and day of her shift

Counsellor must fill in his/her details after which she can book a slot on given day

Counsellor Report

Name

Type

Slot

Day of Shift



ACHIEVE WELLBEING
COUNSELLING

Form 3: Student Registration

Student RegistrationStudent Name University

Rating

MS

A student enters his name and the university he graduated from

After clicking the Rating Button, he can view the counsellor rating report

Counsellor Rating

Name	Type	Address	Rating
Angela	Professional	8 Marshall Street	2
Gibbs	Professional	112 Harvard Place	5.25
Marsha	Professional	South Campus	10
Vera	Professional	118 Concord Place	8

Friday, December 13, 2019

Page 1 of 1

Here I have linked the combo button to the Counsellor rating report

Stored procedure

I have used a stored procedure that automatically calculates the total number of universities a student has applied to and buckets them accordingly to their respective university types.

Stored Procedure

```
create procedure no_univ
as
begin
update university_evaluation
set No_universities=a.number
from
(select studentID,university_type,count(1)as number from university_evaluation
group by studentID,university_type)a
where a.studentID=university_evaluation.studentID
and a.university_type=university_evaluation.university_type
end;
```

Before Stored procedure

	university_type	No_universities	studentID	counsellorID	university_id
1	Moderate	NULL	10	10	101
2	Ambitious	NULL	11	11	103
3	Ambitious	NULL	11	11	104
4	Moderate	NULL	14	10	102
5	Moderate	NULL	14	10	106

A student nor can a counsellor view how many universities a student has applied to after bucketing them into categories which is very imperative while applying to universities abroad as students must prepare documents accordingly

After Stored procedure

```
exec no_univ;
```

Executes the stored procedure

	university_type	No_universities	studentID	counsellorID	university_id
1	Moderate	1	10	10	101
2	Ambitious	2	11	11	103
3	Ambitious	2	11	11	104
4	Moderate	2	14	10	102
5	Moderate	2	14	10	106

Student ID 14 has applied to 2 Moderate universities

Student ID 11 has applied to 2 Ambitious universities

Student ID 10 has applied to 1 Moderate university

Triggers

Trigger 1: I created a trigger that automatically assigns university to the student given his GRE score.

This not only expedites the process but also eases the counsellors work

Trigger

```
create trigger university_type
on profile
for insert,update
as
if @@ROWCOUNT>=1
begin
update university_evaluation
set university_type=tmp.uni
from
(select a.gre_score,a.yrs_Exp,a.studentID,
case when a.gre_score between 270 and 300 then 'Safe'
when a.gre_score between 300 and 315 then 'Moderate'
when a.gre_score between 315 and 340 then 'Ambitious'
else 'Invalid'
end as uni,
b.university_type from
profile a inner join university_evaluation b
on a.studentID=b.studentID)as tmp
where university_evaluation.studentID=tmp.studentID
end;
```

I created this trigger using case when

When the gre score of the student gets updated or inserted the university type changes automatically given the case when condition is followed .

Using the case when I have bucketed the types into three categories

1. Ambitious
2. Moderate
3. Safe

Before trigger

```
insert into university_evaluation values('Safe',null,12,12,104);
```

```
select * from profile;
select * from university_evaluation;

insert into university_evaluation values('Safe',null,12,12,104);
```

Results Messages

gre_score	yrs_Exp	major	counsellorID	studentID
312	4	Computer	10	10
321	1	Electrical	10	11
324	0	Computer	11	12
290	0	Environment	12	13
304	0	Arts	10	14
303	1	Chemistry	12	15

university_type	No_universities	studentID	counsellorID	university_id
Moderate	1	10	10	101
Ambitious	2	11	11	103
Ambitious	2	11	11	104
Safe	NULL	12	12	104
Moderate	2	14	10	102
Moderate	2	14	10	106

As we can see student ID 12 has gre score of 324 as a result he should be assigned an ambitious university but is assigned a safe university

After trigger

```

update profile
set gre_score=324 where studentID=12 ;
exec no_univ;
select * from university_evaluation;

```

university_type	No_universities	studentID	counsellorID	university_id
Moderate	1	10	10	101
Ambitious	2	11	11	103
Ambitious	2	11	11	104
Ambitious	1	12	12	104
Moderate	2	14	10	102
Moderate	2	14	10	106

After creating trigger making update and running the stored procedure we can see student with ID 12 has been assigned 1 ambitious university

```

update profile
set gre_score=321 where studentID=11 ;

```

university_type	No_universities	studentID	counsellorID	university_id
Moderate	NULL	10	10	101
Ambitious	NULL	11	11	103
Ambitious	NULL	11	11	104
Moderate	NULL	14	10	102
Moderate	NULL	14	10	106

Trigger 2:

For a student to book a slot he needs to know which rooms are booked and till what time

Similarly, a counsellor also needs this knowledge to jock down her own schedule.

This trigger serves a double purpose as it not only notifies other counselors when the other counsellors slots ends but also stores this time to calculates the payroll of the counsellor

After a slot has been booked the time is recorded along with when the slot ends after which the counsellor can view his/her total Payroll doing so the time is recorded

Here two tables get updated

1. Payroll that stores the counsellors Payroll
2. Schedule that stores the counselling end time

Used Dateadd function for adding datetime to a numeric value

```

create trigger Schedule_Payroll_Update
on Schedule
for update
as
if @@ROWCOUNT>=1
begin
update Payroll
set Amount=amt.Amount *Payroll.Rate
from
(select a.counsellorID,a.slot_id ,(sum(b.slot_duration))as Amount
from Payroll a inner join Schedule b
on a.slot_id=b.slot_id
group by a.counsellorID,a.slot_id)as amt
where amt.slot_id=Payroll.slot_id

update Schedule
set counselling_end=ending.end1
from
(select DATEADD(HOUR,slot_duration,counselling_time)as end1,slot_duration
from Schedule group by slot_duration,counselling_time)ending
where ending.slot_duration=Schedule.slot_duration;
end;

```


Before trigger

```
select * from Payroll;
select * from Schedule;
```

Results		Messages	
slot_id	counsellorID	Rate	Amount
900	10	11.00	100
901	10	11.00	20
902	12	11.00	200
903	12	11.00	190
904	13	11.00	120
905	14	11.00	110
906	13	11.00	10

slot_id	counselling_day	counselling_time	counselling_end	slot_duration	room_number
900	2017-12-12	2017-12-12 10:00:00.000	NULL	2	201
901	2017-12-13	2017-12-13 13:00:00.000	NULL	4	201
902	2017-11-06	2017-11-06 13:00:00.000	NULL	2	201
903	2018-12-01	2018-12-01 13:00:00.000	NULL	1	201
904	2019-11-06	2019-11-06 15:00:00.000	NULL	5	201
905	2018-02-15	2018-02-15 03:00:00.000	NULL	6	201

As I mentioned earlier random values were added for the sake of the report

Thus counsellor 10 has wrong Payroll assigned to her and counselling end time is null for all slot ID's

After trigger

```

update Schedule
set slot_duration=7 where slot_id=900;

select * from Payroll;
select * from Schedule;

```

results		Messages	
slot_id	counsellorID	Rate	Amount
900	10	11.00	77
901	10	11.00	44
902	12	11.00	22
903	12	11.00	11
904	13	11.00	55
905	14	11.00	66
906	13	11.00	143

slot_id	counselling_day	counselling_time	counselling_end	slot_duration	room_number
900	2017-12-12	2017-12-12 10:00:00.000	2017-12-12 17:00:00.000	7	201
901	2017-12-13	2017-12-13 13:00:00.000	2017-12-13 17:00:00.000	4	201
902	2017-11-06	2017-11-06 13:00:00.000	2017-11-06 15:00:00.000	2	201
903	2018-12-01	2018-12-01 13:00:00.000	2018-12-01 14:00:00.000	1	201

After counsellor 10 books slot 900 for 7 hours she gets paid for those 7 hours at 11dollar /hour rate thus total amount becomes 77dollars

The counselling slot time now starts at 10 am and ends at 5 pm

```

update Schedule
set slot_duration=13 where slot_id=902;

select * from Payroll;
select * from Schedule;

```

Results Messages

slot_id	counsellorID	Rate	Amount
900	10	11.00	77
901	10	11.00	44
902	12	11.00	143
903	12	11.00	11
904	13	11.00	55
905	14	11.00	66
906	13	11.00	143

slot_id	counselling_day	counselling_time	counselling_end	slot_duration	room_number
900	2017-12-12	2017-12-12 10:00:00.000	2017-12-12 17:00:00.000	7	201
901	2017-12-13	2017-12-13 13:00:00.000	2017-12-13 17:00:00.000	4	201
902	2017-11-06	2017-11-06 13:00:00.000	2017-11-07 02:00:00.000	13	201
903	2018-12-01	2018-12-01 13:00:00.000	2018-12-01 14:00:00.000	1	201
904	2019-11-06	2019-11-06 15:00:00.000	2019-11-06 20:00:00.000	5	201

After counsellor 12 books slot 902 for 13 hours she gets paid for those 13 hours at 11dollar /hour rate thus total amount becomes 143 dollars

The counselling slot time now starts at 1 pm on 6th November and ends at 2 am on 7th November (Hypothetical example to demonstrate the strength of this trigger)