ADVANCED DATABASE SYSTEMS PROGRAMMING PROJECT

NAME : DINESH REDDY KASU
UID : U00747253

TABLE OF CONTENTS

DESCRIPTION OF THE APPLICATION

Now a days, each and every individual from entry-level to senior citizen who has interested to work for Information technology industries, is trying to improve his/her technical skills. It is being done by taking online classes or visiting the training centers ,where the technical skills training program will be conducted by mentors. Thus, it helps the individual to stay updated on Technical Skills like Java,Mysql,Oracle and Php. So, here my application is *technical skills training program*, will have different levels of users such as Director,Mentor,&Student.The Director setsup the programs in the centers and assign the mentors to these. Then, the mentors will teach programs to students.The students attends these programs, are assigned to mentors and then ask the questions to mentors if they have any in the program i.e technical skill. So,The director can keep track of which student attends and the mentors that taught to conduct the program skills session. For questions that students have, the system will have an option to ask query and quicker way for the mentors to reply the answer. My web based database application goal is to create a Information database system to satisfy the needs of Director, Mentor and Student roles and also it helps to Director and Mentors for analyzing the organization growth in terms of profit and students. This project is aimed at the following tasks.

Task1(Transactions):

- To capture few transactions in *Technical Skills Training Program* database application.In this application, the Director, Mentor, and Student will have their own login credentials to enter this website and do the following functionalaties.

Task2(OLAP Style Analysis):

- To capture few transactions in involving the *Technical Skills Training Program* database application. Here, we used the Dimensional Model Datamart for this application.Here, the Director will have an option to analyse the Organization growth in the states interms of profit and students(male/female) including their ages. Also, the programming skills that are popular in the particular county,city, and states.In addition to these,they can also check the all parameters year,month,day wise.

Task3(Triggers Usage):

- To capture the Transaction involving the Trigger concept while inserting or updating the table record.

Task4(Temporal Database):

- To capture a transaction in my application to maintain seperate history table for a particular table to maintain its updates.This is implemented using bi temporal db concept.

Task5(Query Processing):

- To capture few transactions involving optimized queries to improve the system performance.

DATA REQUIREMENTS

Student

Each Student is uniquely identified by his/her student id, and further described by his/her student name, cell number, email ,address and age.

Mentor

A mentor is uniquely identified by his/her mentor id, and further described by his/her name, cell number, email , and salary.

Director

Each director is uniquely identified by his/her director id and is further described by his/her cell number ,email ,and gender.

Question

Questions are uniquely identified by question id, and further identified by question and answer.

Program

Each Programming skill is uniquely identified by program id, and further described by program name, city, state, fees and its starting date.

INSTANCES

Student “Assigned" to Mentor

Each student is assigned to one mentor.

Each mentor may have more than one student.

Each student must have a mentor.

It is not necessary that Mentor must have a student.

Mentor “Reports to” Director

Each mentor must report to a director.

A director does not have to have mentors assigned to them.

A mentor can only report to one director.

A director can have many mentors assigned to them.

Students “ask” Questions

A student can ask a query.

A query must be asked by a student.

A student can ask many queries

A query can't be asked by a multiple number of students.

Mentor “answers” Queries

Mentors are required to answer queries.

A query must be answered by a mentor.

A mentor can reply to many queries

Each query can be replied by one mentor only.

Students “attend” program

A program has to have students attend to it.

A program can have many students.

A student can attend only one program.

Director “sets up” program

A director could set up programs.

A program must be set up by one director only.

A director can set up multiple programs.

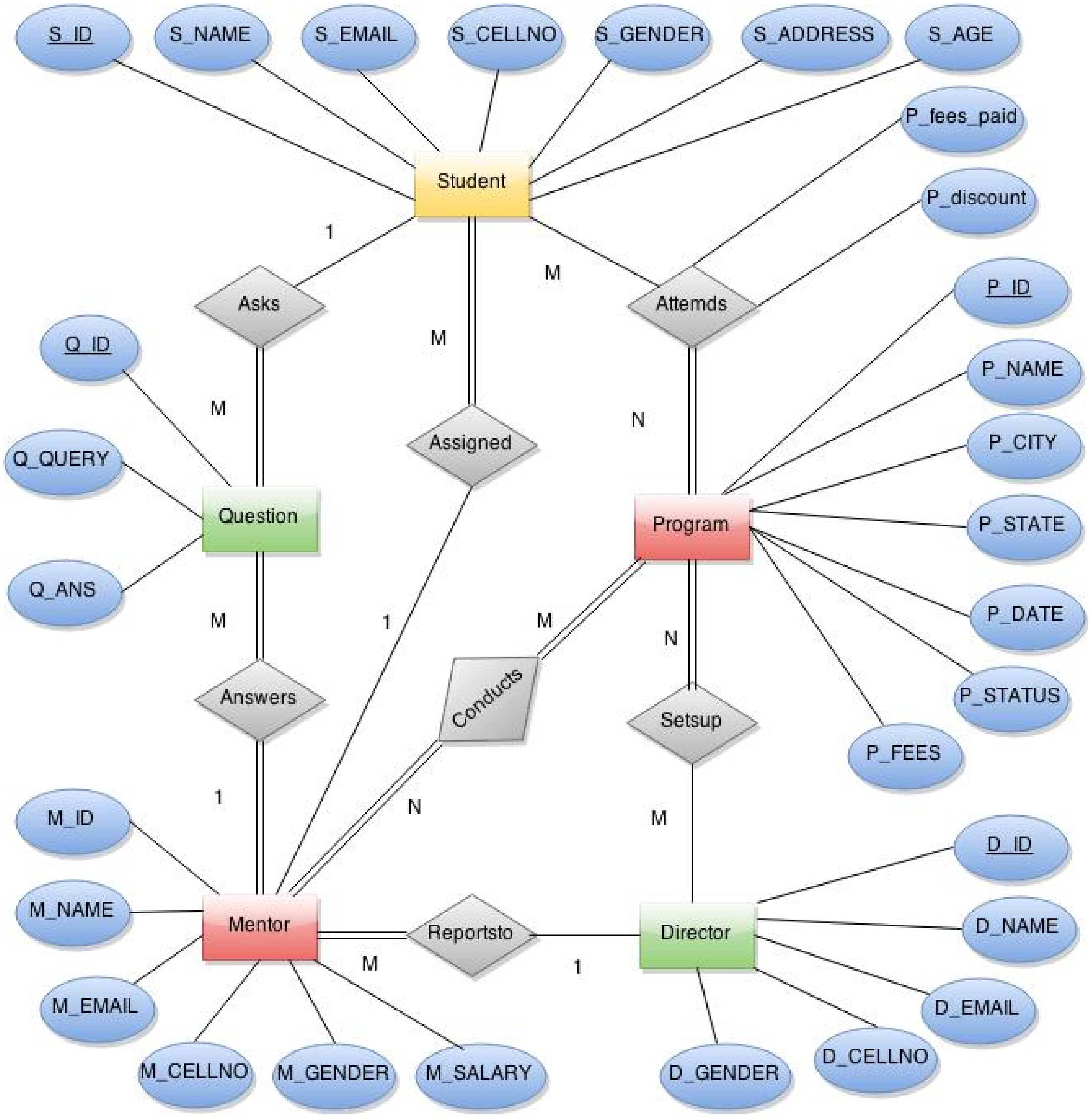
Mentors “conducts” Programming Sessions

A mentor is not required to conduct a programming session.

A program has to be ran by a mentor only.

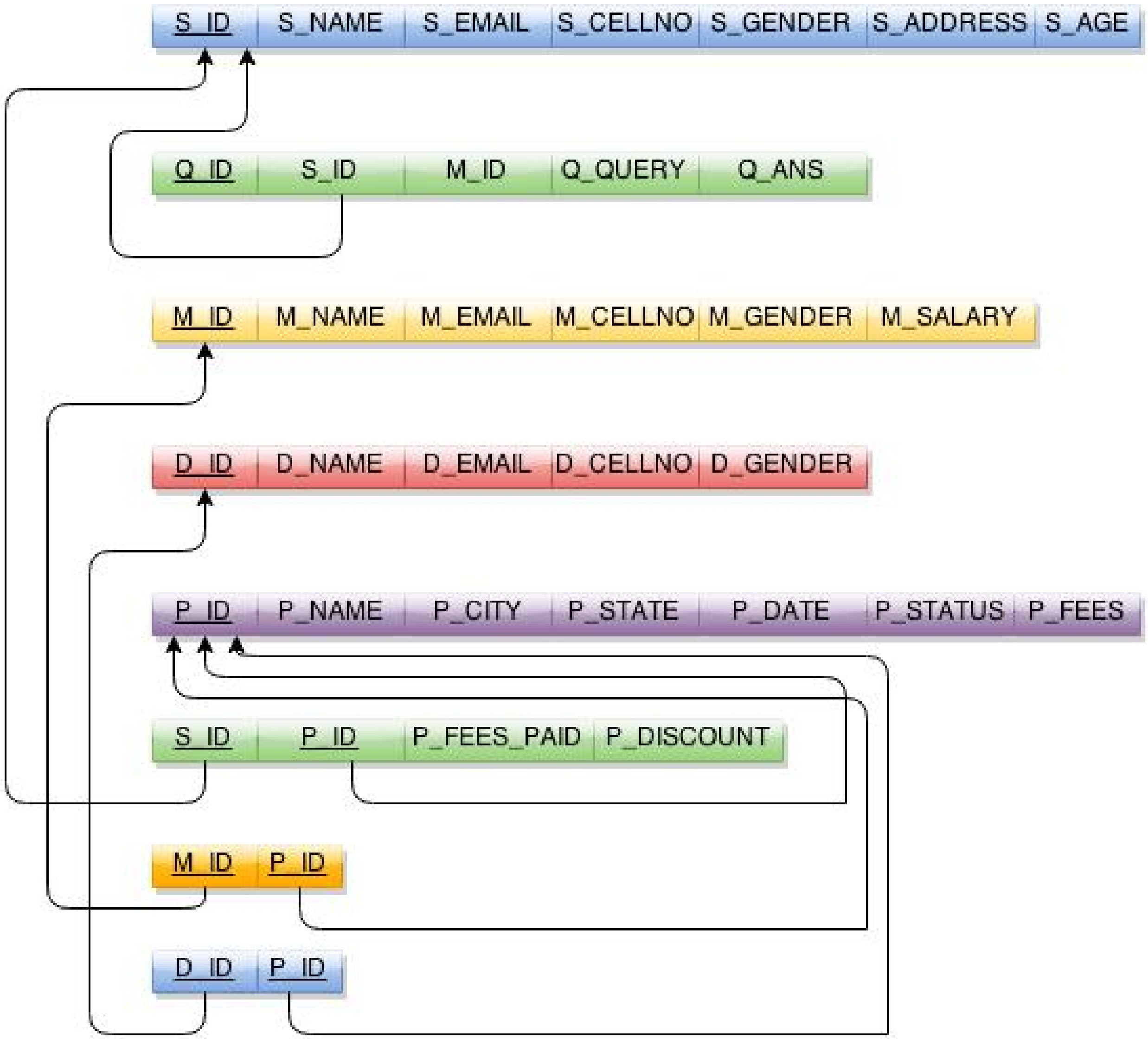
A mentor can conducts multiple programming sessions.

ENTITY RELATIONSHIP DIAGRAM



Figure#1: ER Diagram for Project

RELATIONAL SCHEMA


S_ID S_NAME S_EMAIL S_CELLNO S_GENDER S_ADDRESS S_AGE
Q_ID S_ID M_ID Q_QUERY Q_ANS
M_ID M_NAME M_EMAIL M_CELLNO M_GENDER M_SALARY
D_ID D_NAME D_EMAIL D_CELLNO D_GENDER
P_ID P_NAME P_CITY P_STATE P_DATE P_STATUS P_FEES
S_ID P_ID P_FEES_PAID P_DISCOUNT
M_ID P_ID
D_ID P_ID

Figure#2: Relation Schema for Project

TABLE DEFINITIONS

```
CREATE DATABASE `tstp`;


CREATE TABLE IF NOT EXISTS `tstp`.`student` (
  `s_id` char(10) NOT NULL,
  `s_name` char(30) DEFAULT NULL,
  `s_email` char(30) DEFAULT NULL,
  `s_cellno` char(10) DEFAULT NULL,
  `s_gender` char(10) DEFAULT NULL,
  `s_address` char(50) DEFAULT NULL,
  `s_age` int(11) DEFAULT NULL,
   PRIMARY KEY(`s_id`)
)
CREATE TABLE IF NOT EXISTS `tstp`.`program` (
  `p_id` char(10) NOT NULL,
  `p_name` char(30) DEFAULT NULL,
  `p_city` char(30) DEFAULT NULL,
  `p_state` char(2) DEFAULT NULL,
  `p_date` date DEFAULT NULL,
  `p_status` char(10) DEFAULT NULL,
  `p_fees` decimal(10,2) DEFAULT NULL,
   PRIMARY KEY(`p_id`)
)
CREATE TABLE IF NOT EXISTS `tstp`.`director` (
  `d_id` char(10) NOT NULL,
  `d_name` char(30) DEFAULT NULL,
  `d_email` char(30) DEFAULT NULL,
  `d_cellno` char(10) DEFAULT NULL,
  `d_gender` char(10) DEFAULT NULL,
PRIMARY KET(`d_id`))
```

```
CREATE TABLE IF NOT EXISTS `tstp`.`mentor` (
  `m_id` char(10) NOT NULL,
  `m_name` char(30) DEFAULT NULL,
  `m_email` char(30) DEFAULT NULL,
  `m_cellno` char(10) DEFAULT NULL,
  `m_gender` char(10) DEFAULT NULL,
  `m_salary` decimal(10,2) DEFAULT NULL,
PRIMARY KET(`m_id`)
)

CREATE TABLE IF NOT EXISTS `tstp`.`attends` (
  `s_id` char(10) NOT NULL,
  `p_id` char(10) NOT NULL,
  `P_fees_paid` decimal(10,2) DEFAULT NULL,
  `p_discount` int(11) DEFAULT NULL,
PRIMARY KEY (`s_id`,`p_id`),
FOREIGN KEY(`s_id`) REFERENCES `tstp`.`student`(`s_id`),
FOREIGN KEY(`p_id`) REFERENCES `tstp`.`program`(`p_id`)
)

CREATE TABLE IF NOT EXISTS `tstp`.`conducts` (
  `m_id` char(10) NOT NULL,
  `p_id` char(10) NOT NULL,
PRIMARY KEY (`m_id`,`p_id`),
FOREIGN KEY(`m_id`) REFERENCES `tstp`.`mentor`(`m_id`),
FOREIGN KEY(`p_id`) REFERENCES `tstp`.`program`(`p_id`)
)
```

```
CREATE TABLE IF NOT EXISTS `tstp`.`question` (
  `q_id` char(10) NOT NULL,
  `p_id` char(10) DEFAULT NULL,
  `s_id` char(10) DEFAULT NULL,
  `m_id` char(10) DEFAULT NULL,
  `q_query` char(30) DEFAULT NULL,
  `q_ans` char(100) DEFAULT NULL,
PRIMARY KEY (`q_id`),
FOREIGN KEY(`s_id`) REFERENCES `tstp`.`student`(`s_id`),
FOREIGN KEY(`m_id`) REFERENCES `tstp`.`mentor`(`m_id`)
)
CREATE TABLE IF NOT EXISTS `tstp`.`setsup` (
  `d_id` char(10) NOT NULL,
  `p_id` char(10) NOT NULL,
PRIMARY KEY (`d_id`,`p_id`),
FOREIGN KEY(`d_id`) REFERENCES `tstp`.`director`(`d_id`),
FOREIGN KEY(`p_id`) REFERENCES `tstp`.`program`(`p_id`)
)
```

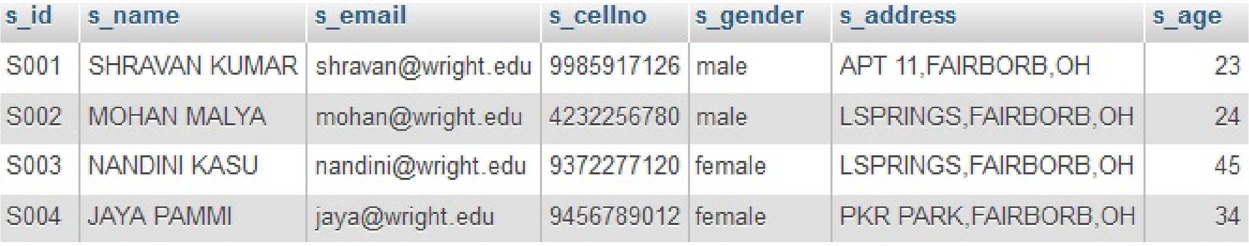
NORMALIZATION

My DB application's relation schema is in 3NF state as it satisfies the following conditions

- It does not have multi valued attributes. i.e The intersection of each row and column have unique value.
- It does not have partial dependencies. i.e There are no redundancies to identify the each tuple with unique attributes.
- It does not have transitive dependencies. i.e There are no non-prime attributes those depend on the primary key.

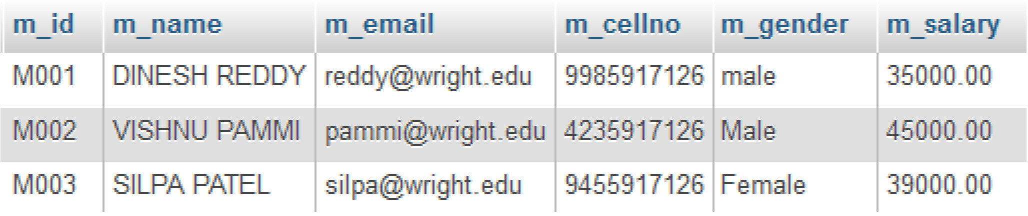
PRINTOUT OF THE CONTENT OF DATA TABLES

STUDENT



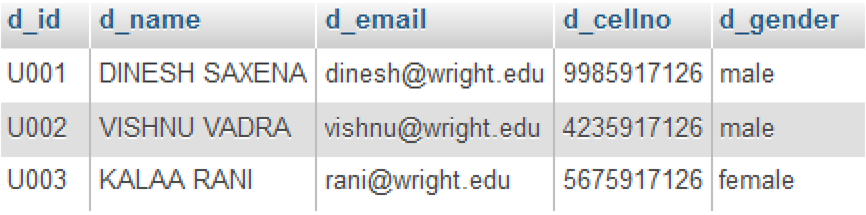
s_id	s_name	s_email	s_cellno	s_gender	s_address	s_age
S001	SHRAVAN KUMAR	shravan@wright.edu	9985917126	male	APT 11,FAIRBORB,OH	23
S002	MOHAN MALYA	mohan@wright.edu	4232256780	male	LSPRINGS,FAIRBORB,OH	24
S003	NANDINI KASU	nandini@wright.edu	9372277120	female	LSPRINGS,FAIRBORB,OH	45
S004	JAYA PAMMI	jaya@wright.edu	9456789012	female	PKR PARK,FAIRBORB,OH	34

MENTOR



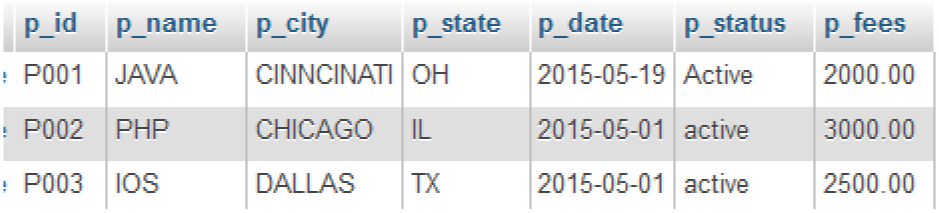
m_id	m_name	m_email	m_cellno	m_gender	m_salary
M001	DINESH REDDY	reddy@wright.edu	9985917126	male	35000.00
M002	VISHNU PAMMI	pammi@wright.edu	4235917126	Male	45000.00
M003	SILPA PATEL	silpa@wright.edu	9455917126	Female	39000.00

DIRECTOR



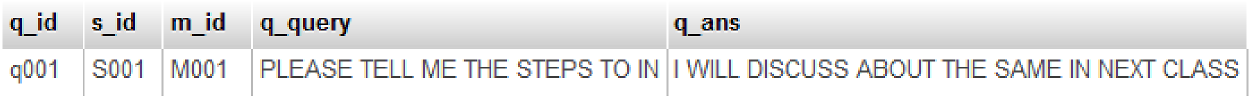
d_id	d_name	d_email	d_cellno	d_gender
U001	DINESH SAXENA	dinesh@wright.edu	9985917126	male
U002	VISHNU VADRA	vishnu@wright.edu	4235917126	male
U003	KALAA RANI	rani@wright.edu	5675917126	female

PROGRAM



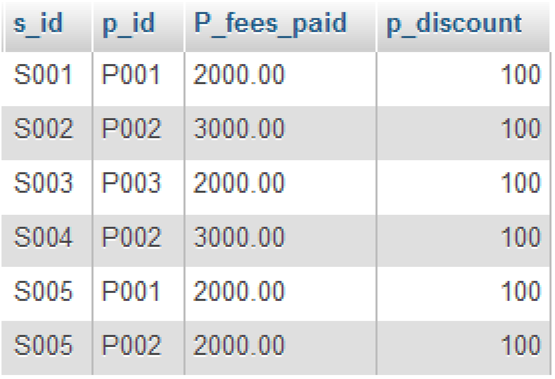
p_id	p_name	p_city	p_state	p_date	p_status	p_fees
P001	JAVA	CINNCINATI	OH	2015-05-19	Active	2000.00
P002	PHP	CHICAGO	IL	2015-05-01	active	3000.00
P003	IOS	DALLAS	TX	2015-05-01	active	2500.00

QUESTION



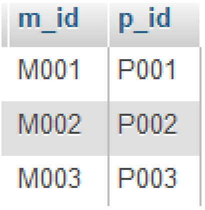
q_id	s_id	m_id	q_query	q_ans
q001	S001	M001	PLEASE TELL ME THE STEPS TO IN	I WILL DISCUSS ABOUT THE SAME IN NEXT CLASS

ATTENDS



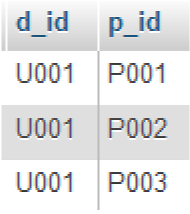
s_id	p_id	P_fees_paid	p_discount
S001	P001	2000.00	100
S002	P002	3000.00	100
S003	P003	2000.00	100
S004	P002	3000.00	100
S005	P001	2000.00	100
S005	P002	2000.00	100

CONDUCTS



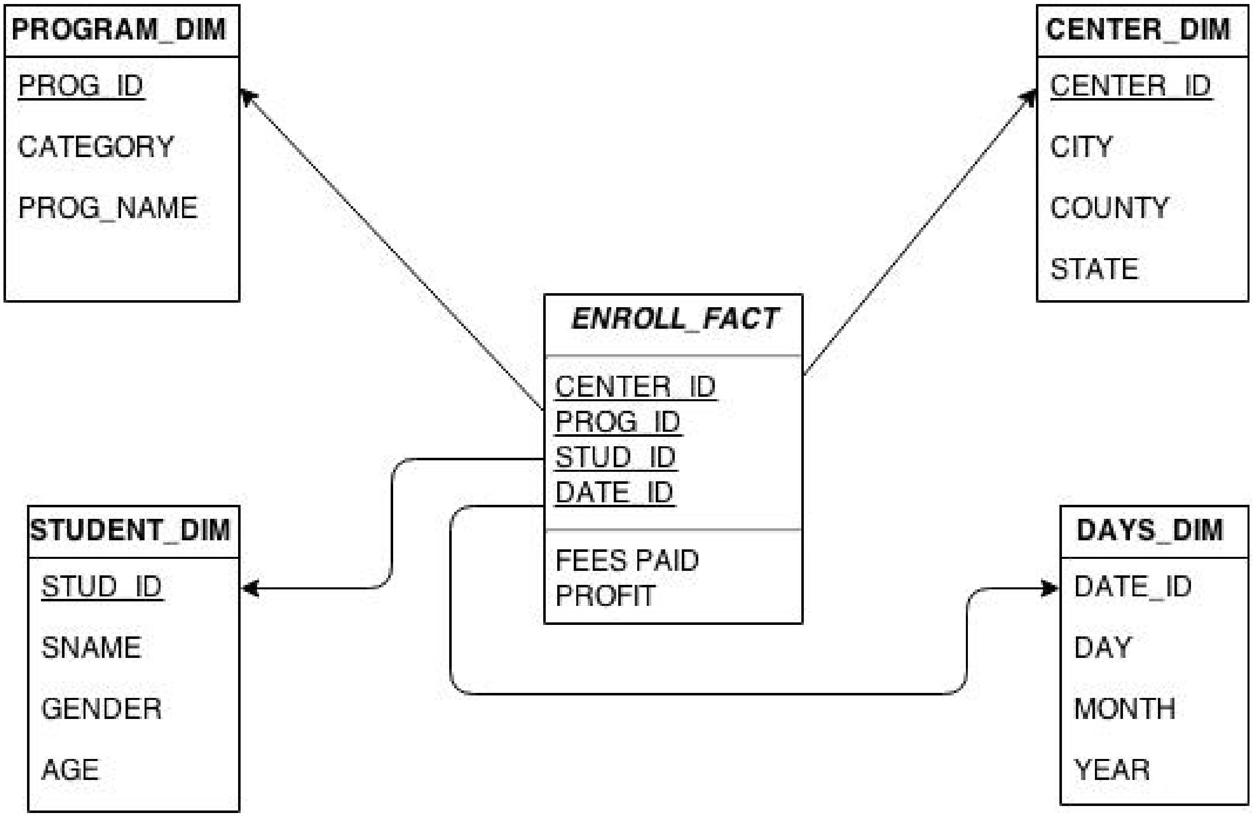
m_id	p_id
M001	P001
M002	P002
M003	P003

SETSUP



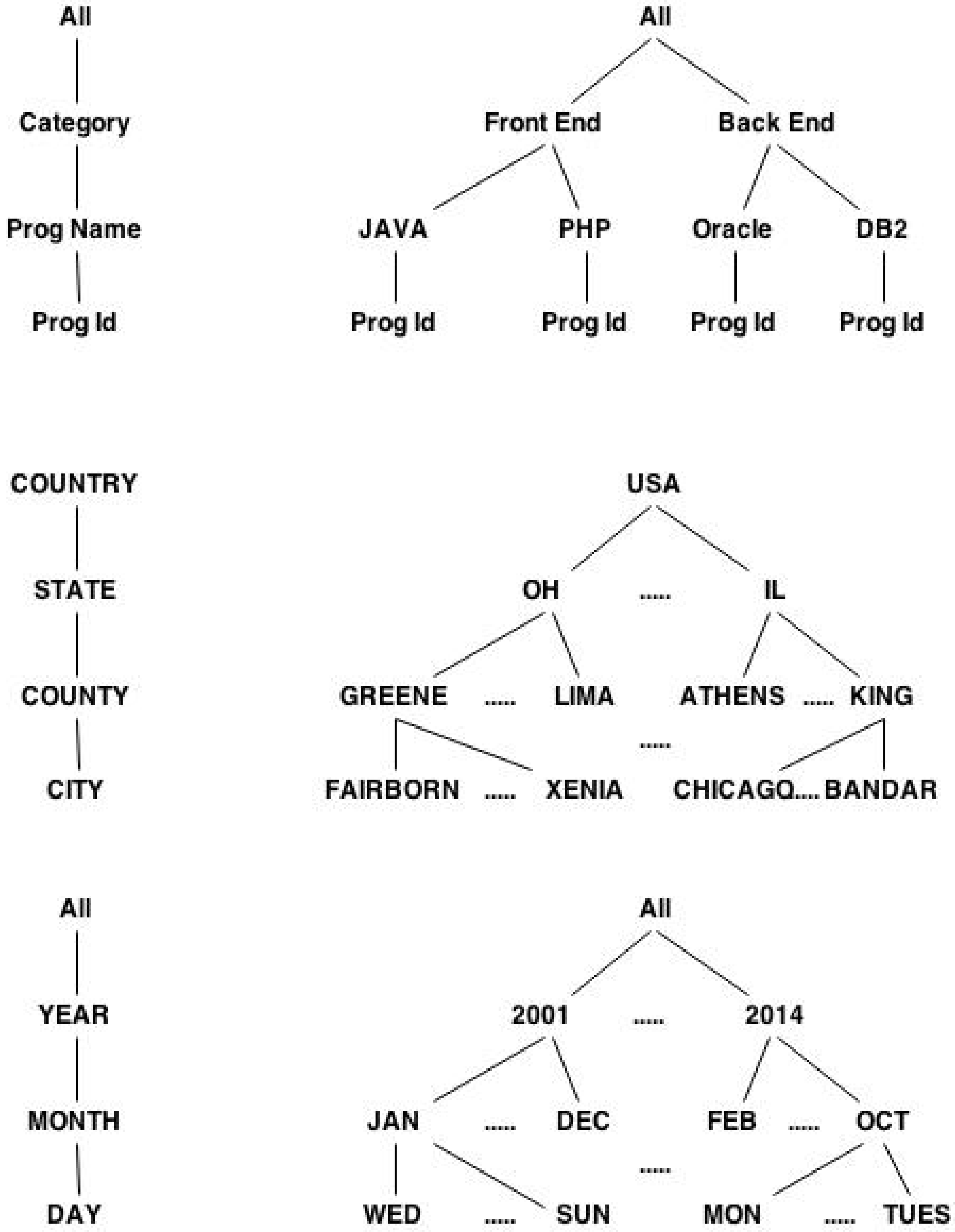
d_id	p_id
U001	P001
U001	P002
U001	P003

STAR SCHEMA FOR APPLICATION DATA MART

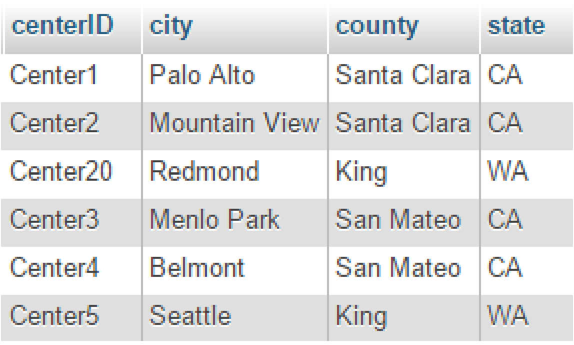


Figure#3 : Star Schema for application data mart

DIMENSION HIERARCHIES

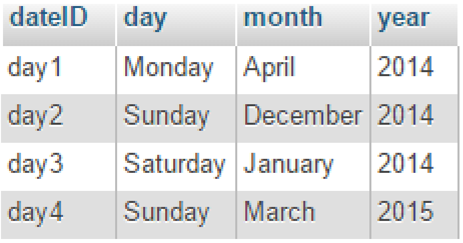


CENTER_DIM



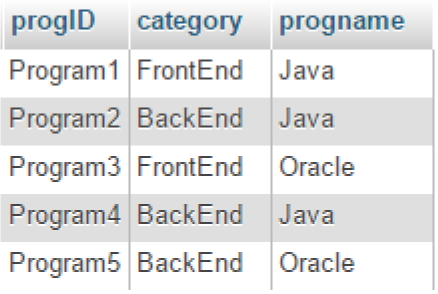
centerID	city	county	state
Center1	Palo Alto	Santa Clara	CA
Center2	Mountain View	Santa Clara	CA
Center20	Redmond	King	WA
Center3	Menlo Park	San Mateo	CA
Center4	Belmont	San Mateo	CA
Center5	Seattle	King	WA

DAYS



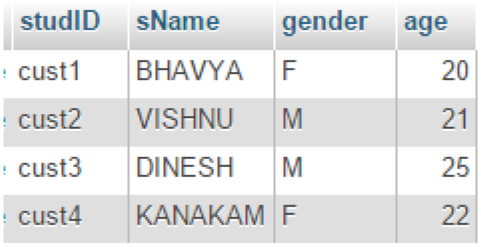
dateID	day	month	year
day1	Monday	April	2014
day2	Sunday	December	2014
day3	Saturday	January	2014
day4	Sunday	March	2015

PROGRAM_DIM



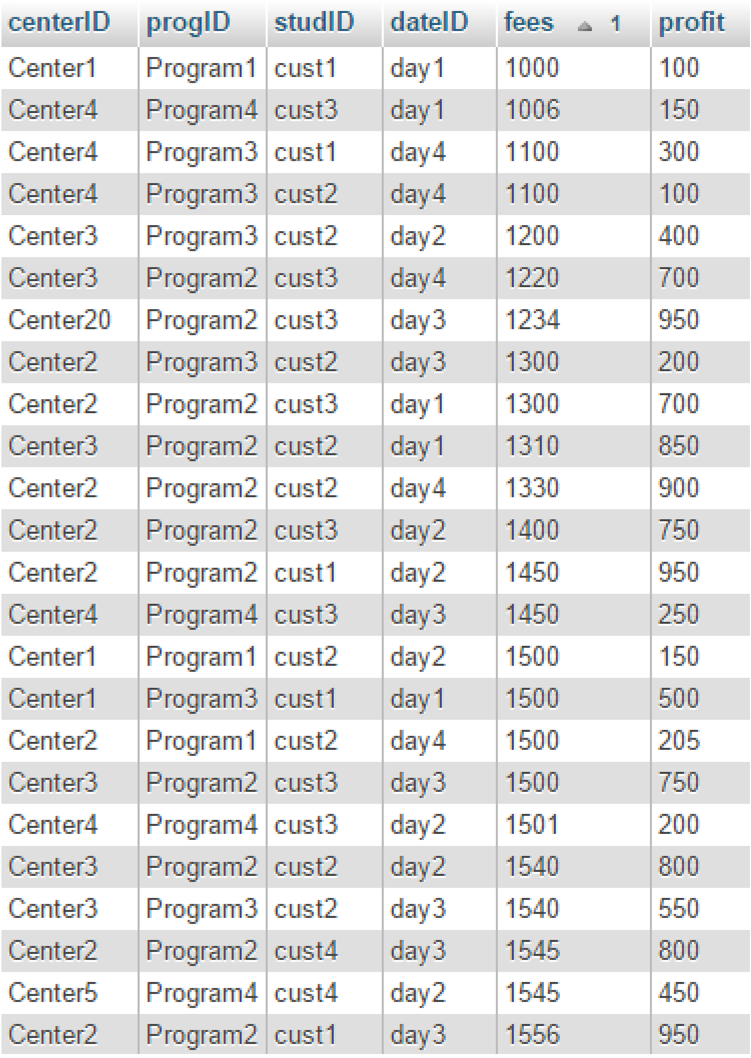
progID	category	progname
Program1	FrontEnd	Java
Program2	BackEnd	Java
Program3	FrontEnd	Oracle
Program4	BackEnd	Java
Program5	BackEnd	Oracle

STUDENT_DIM

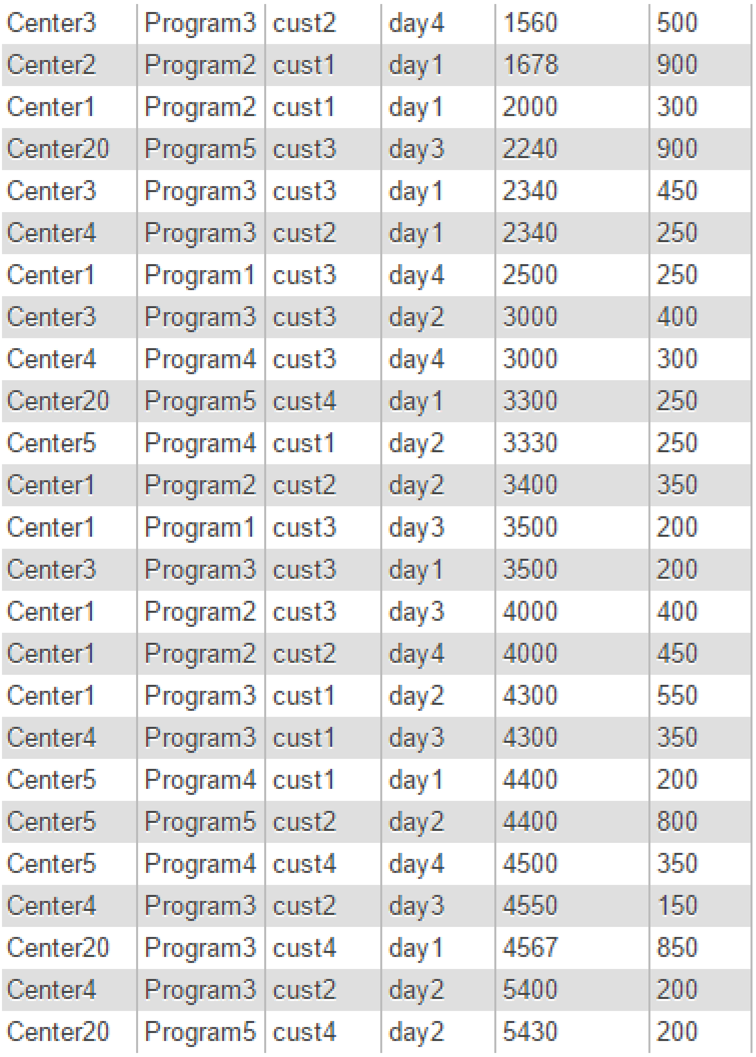


studID	sName	gender	age
cust1	BHAVYA	F	20
cust2	VISHNU	M	21
cust3	DINESH	M	25
cust4	KANAKAM	F	22

ENROLL_FACT



centerID	progID	studID	dateID	fees ▲ 1	profit
Center1	Program1	cust1	day1	1000	100
Center4	Program4	cust3	day1	1006	150
Center4	Program3	cust1	day4	1100	300
Center4	Program3	cust2	day4	1100	100
Center3	Program3	cust2	day2	1200	400
Center3	Program2	cust3	day4	1220	700
Center20	Program2	cust3	day3	1234	950
Center2	Program3	cust2	day3	1300	200
Center2	Program2	cust3	day1	1300	700
Center3	Program2	cust2	day1	1310	850
Center2	Program2	cust2	day4	1330	900
Center2	Program2	cust3	day2	1400	750
Center2	Program2	cust1	day2	1450	950
Center4	Program4	cust3	day3	1450	250
Center1	Program1	cust2	day2	1500	150
Center1	Program3	cust1	day1	1500	500
Center2	Program1	cust2	day4	1500	205
Center3	Program2	cust3	day3	1500	750
Center4	Program4	cust3	day2	1501	200
Center3	Program2	cust2	day2	1540	800
Center3	Program3	cust2	day3	1540	550
Center2	Program2	cust4	day3	1545	800
Center5	Program4	cust4	day2	1545	450
Center2	Program2	cust1	day3	1556	950



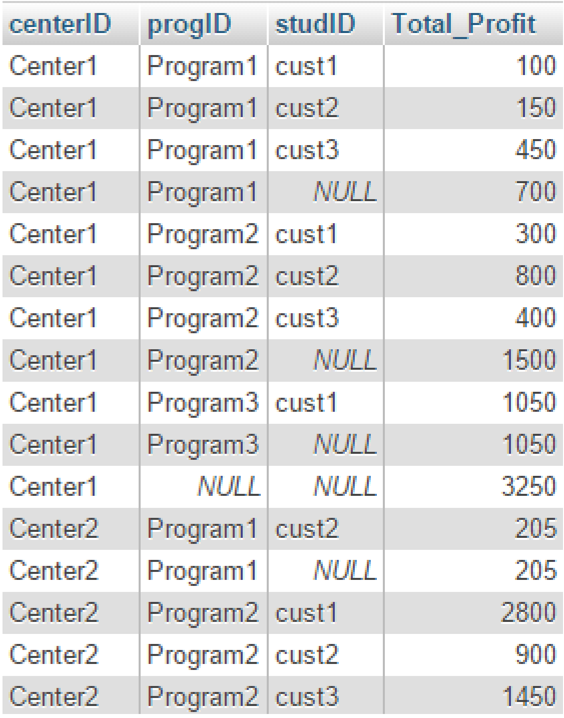
Center3	Program3	cust2	day4	1560	500
Center2	Program2	cust1	day1	1678	900
Center1	Program2	cust1	day1	2000	300
Center20	Program5	cust3	day3	2240	900
Center3	Program3	cust3	day1	2340	450
Center4	Program3	cust2	day1	2340	250
Center1	Program1	cust3	day4	2500	250
Center3	Program3	cust3	day2	3000	400
Center4	Program4	cust3	day4	3000	300
Center20	Program5	cust4	day1	3300	250
Center5	Program4	cust1	day2	3330	250
Center1	Program2	cust2	day2	3400	350
Center1	Program1	cust3	day3	3500	200
Center3	Program3	cust3	day1	3500	200
Center1	Program2	cust3	day3	4000	400
Center1	Program2	cust2	day4	4000	450
Center1	Program3	cust1	day2	4300	550
Center4	Program3	cust1	day3	4300	350
Center5	Program4	cust1	day1	4400	200
Center5	Program5	cust2	day2	4400	800
Center5	Program4	cust4	day4	4500	350
Center4	Program3	cust2	day3	4550	150
Center20	Program3	cust4	day1	4567	850
Center4	Program3	cust2	day2	5400	200
Center20	Program5	cust4	day2	5430	200

CUBE CREATION

Here, I created a Materialized View with the name called `Cube`. and its syntax as follows

```
create table Cube as

select centerID, progID, studID, sum(profit) as Total_Profit

from Enroll_Fact

group by centerID, progID, studID with rollup

union

select centerID, progID, studID, sum(profit)) as Total_Profit

from Sales

group by centerID, progID, studID with rollup

union

select centerID, progID, studID, sum(price) as Total_Profit

from Sales

group by centerID, progID, studID with rollup;
```

*Some portion of the this cube data is attached in below figure.



centerID	progID	studID	Total_Profit
Center1	Program1	cust1	100
Center1	Program1	cust2	150
Center1	Program1	cust3	450
Center1	Program1	NULL	700
Center1	Program2	cust1	300
Center1	Program2	cust2	800
Center1	Program2	cust3	400
Center1	Program2	NULL	1500
Center1	Program3	cust1	1050
Center1	Program3	NULL	1050
Center1	NULL	NULL	3250
Center2	Program1	cust2	205
Center2	Program1	NULL	205
Center2	Program2	cust1	2800
Center2	Program2	cust2	900
Center2	Program2	cust3	1450

TRANSACTIONS involving OLTP

1.Mentor Login

- User is allowed to login only if he/she has a record in MENTOR Table.

2. Create Student Profile

- A record will be created in STUDENT and ATTENDS table.

3.Director Login

- User is allowed to login only if he/she has a record in DIRECTOR Table.

4.Update Mentor Details

- Update the record in MENTOR table.

5.Student Login

- User is allowed to login only if he/she has a record in STUDENT Table.

6.View Program Details

- Select the attributes from PROGRAM table and other tables like CONDUCTS also will be populated in backend.

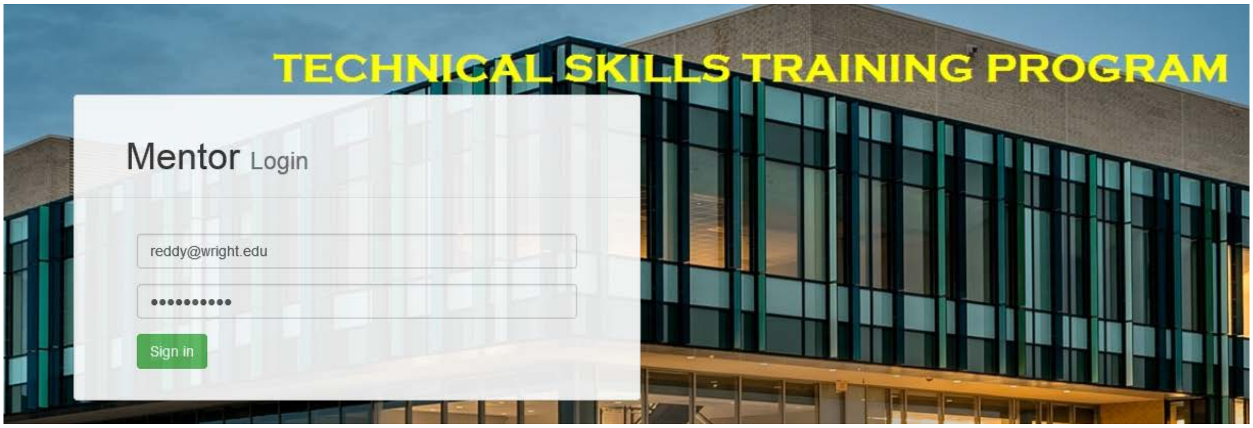
7.Drop Student Profile

- Delete the record from ATTENDS, QUESTION, and STUDENT Tables.

1.Mentor Login

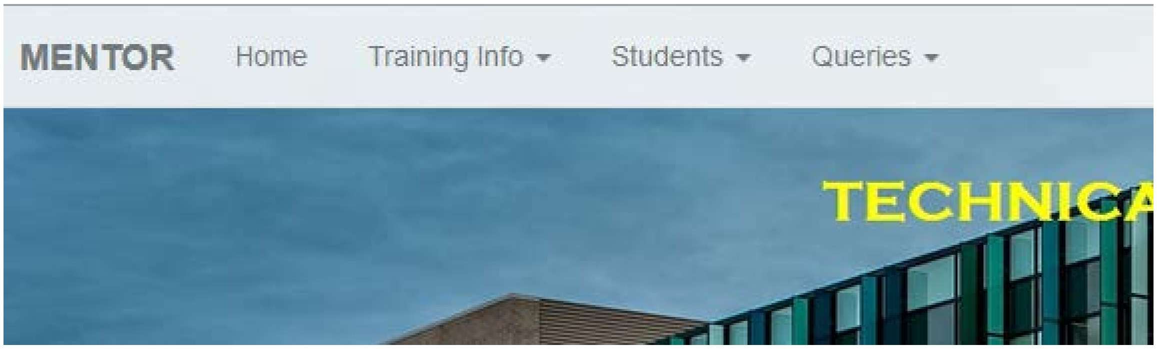
login_mentor.php

- **Input Data :** MailId as UserName and CellNo as password
- **Functionality:** Data Entry Form that takes the UserName and password of Mentor and passes it to login form.
- **SQL Query:** "select * from mentor where m_email='$UserName' and m_cellno='$Password'";



admin_mentor.php

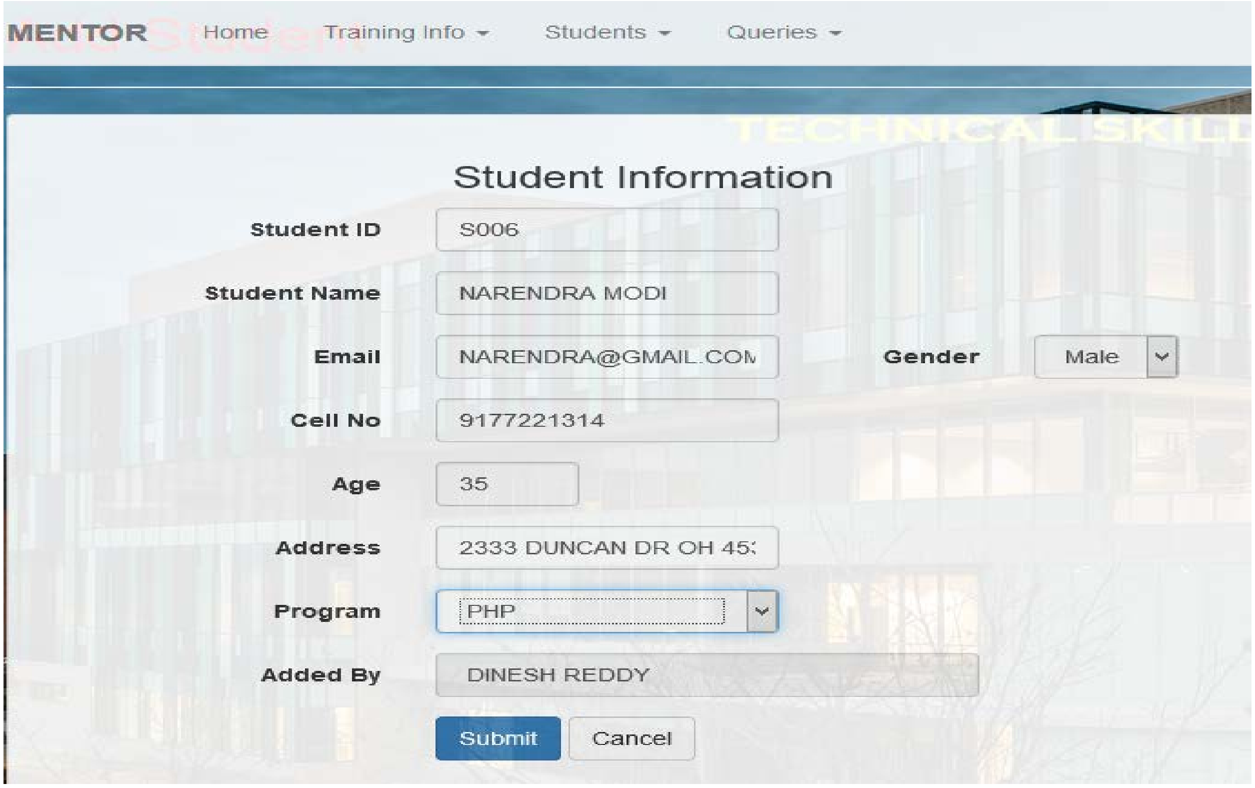
- **Output Data:** If the above login is successful, then user comes to welcome page of mentor .It has the below functionalities from its menu.



2. Create Student Profile

admin_mentor.php

- **Input Data :** Student Details such as Name,Email,Address,Age,Cellno,Program,Gender, and Id.
- **Functionality:** Data Entry Form that takes the student details and passes it to add_student.php form.
- **SQL Query:**
 "INSERT INTO STUDENT (s_id,s_name,s_email,s_cellno,s_gender,s_address,s_age)
VALUES ('$a','$b','$c','$d','$e','$f','$g')";

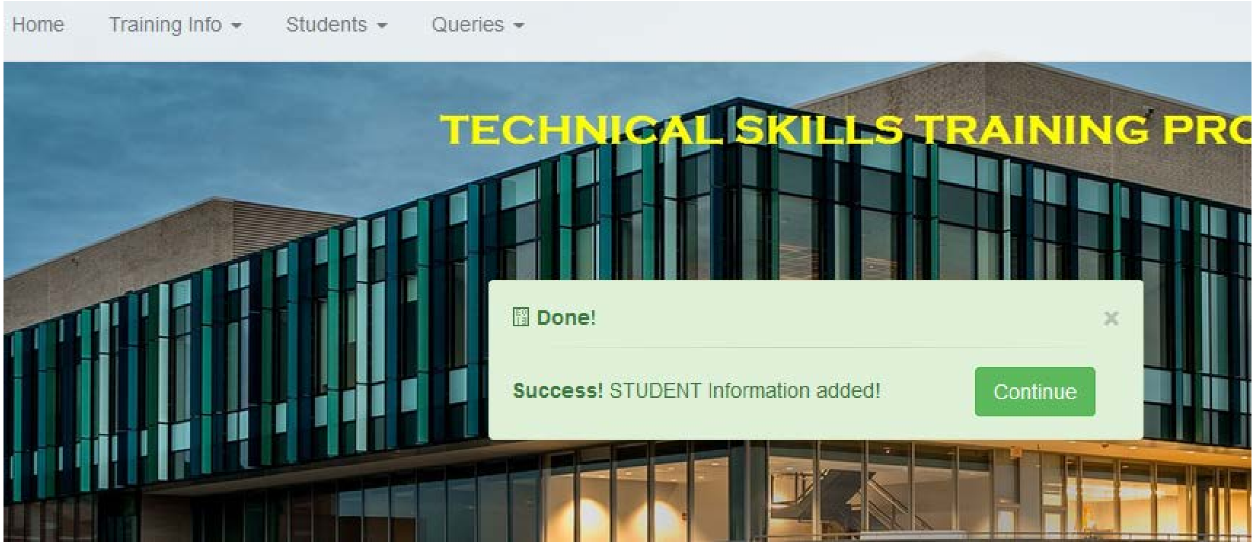


Student Information

Student ID	S006
Student Name	NARENDRA MODI
Email	NARENDRA@GMAIL.COM
Gender	Male
Cell No	9177221314
Age	35
Address	2333 DUNCAN DR OH 45:
Program	PHP
Added By	DINESH REDDY

add_student.php

- **Output Data:** This form stores the student details into STUDENT table

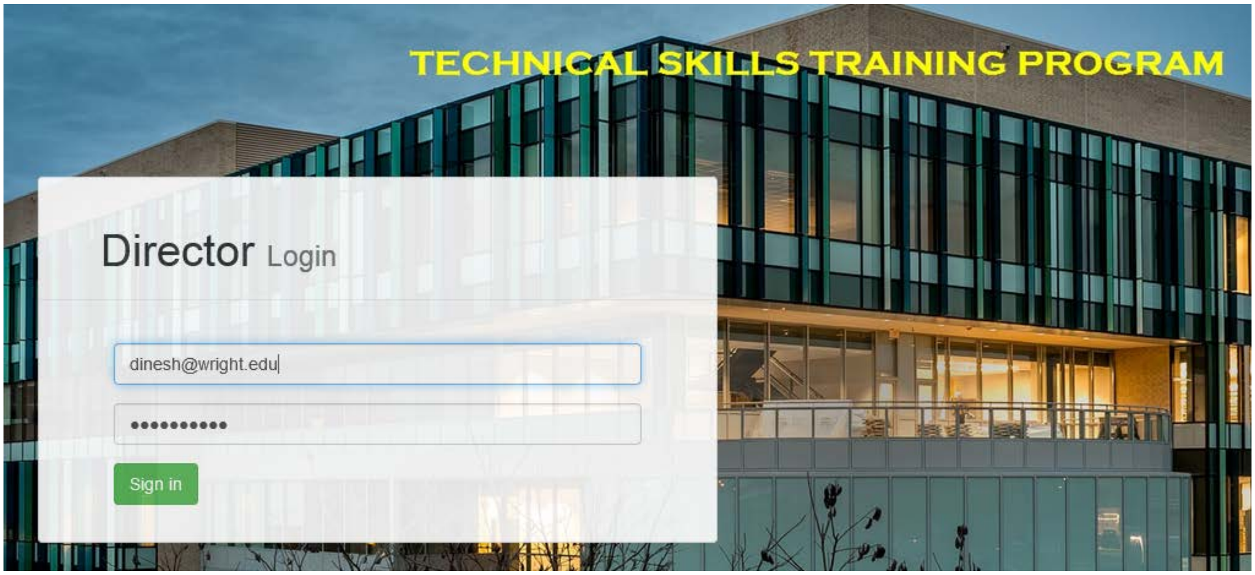


Success! STUDENT Information added!

3.Director Login

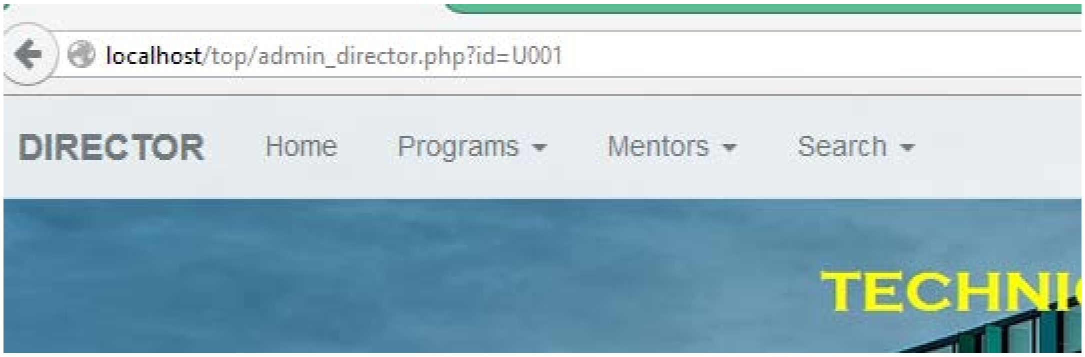
Index.php

- **Input Data :** MailId as UserName and CellNo as password
- **Functionality:** Data Entry Form that takes the UserName and password of Director and passes it to login form.
- **SQL Query:** "select * from director where d_email='$UserName' and d_cellno='$Password'";



admin_director.php

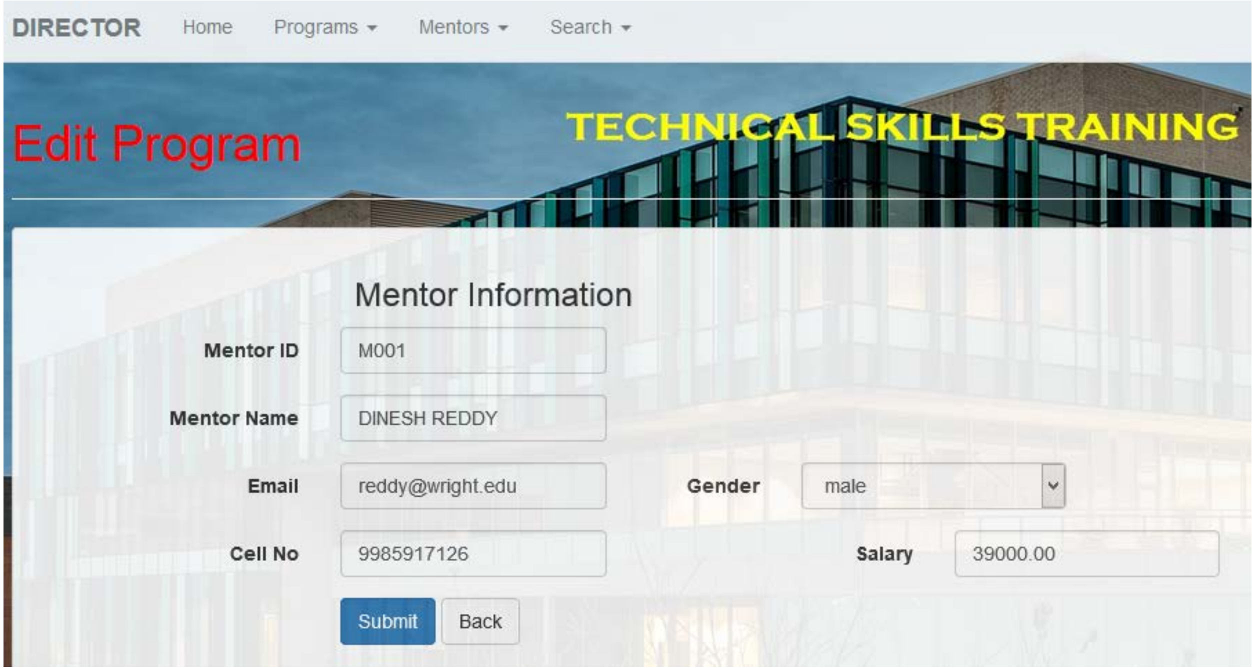
- **Output Data:** If the above login is successful, then user comes to welcome page of director .It has the below functionalities from its menu.



4.Update Mentor Details

update_mentor.php

- **Input Data :** Mentor Details such as Name,Email,Address,Salary, Cellno,Gender, and Id.
- **Functionality:** Data Entry Form that takes the Mentor details and passes it to edit_mentor.php form.
- **SQL Query:** "UPDATE `MENTOR` SET `M_NAME` = '$b',`M_EMAIL` = '$c',`M_CELLNO` = '$d',`M_GENDER` = '$e',`M_SALARY` = '$f' WHERE M_ID = '$a'";

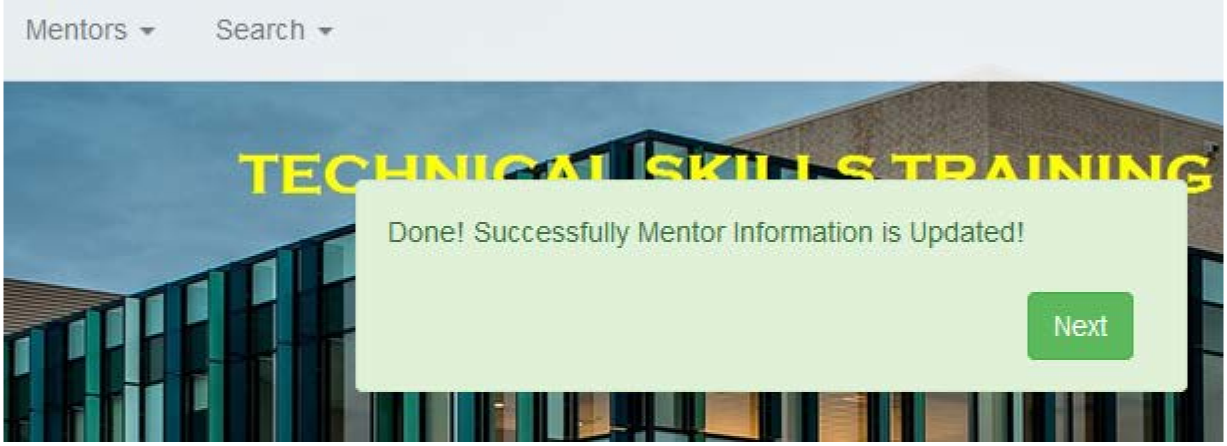


Mentor Information

Mentor ID	M001		
Mentor Name	DINESH REDDY		
Email	reddy@wright.edu	Gender	male
Cell No	9985917126	Salary	39000.00

edit_mentor.php

- **Output Data:** This form updates the mentor details into MENTOR table.

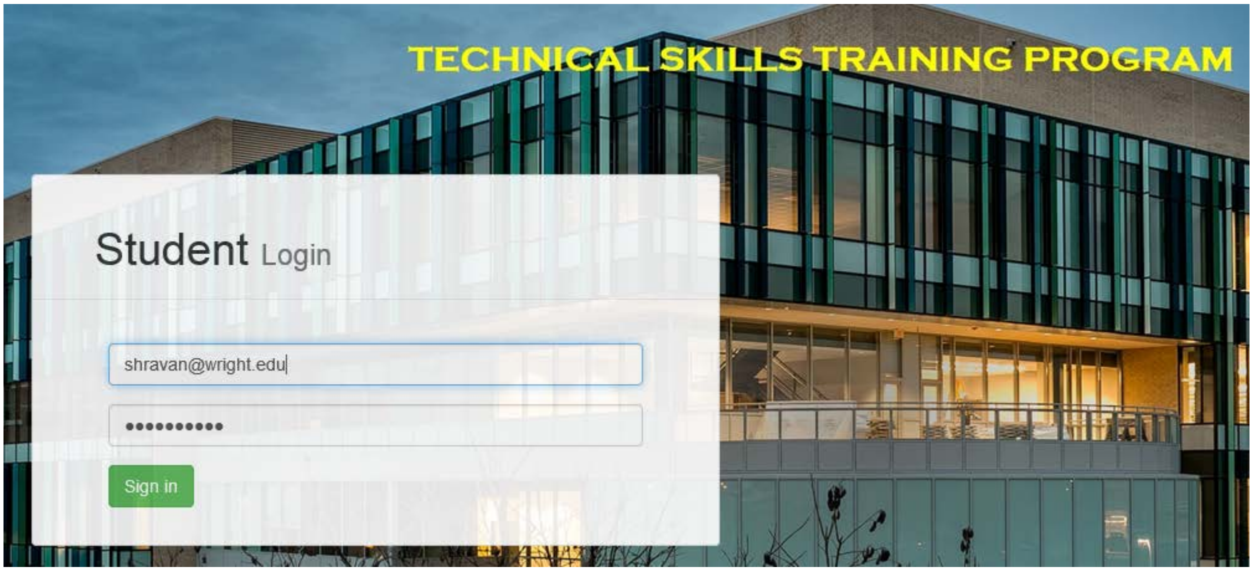


Done! Successfully Mentor Information is Updated!

5.Student Login

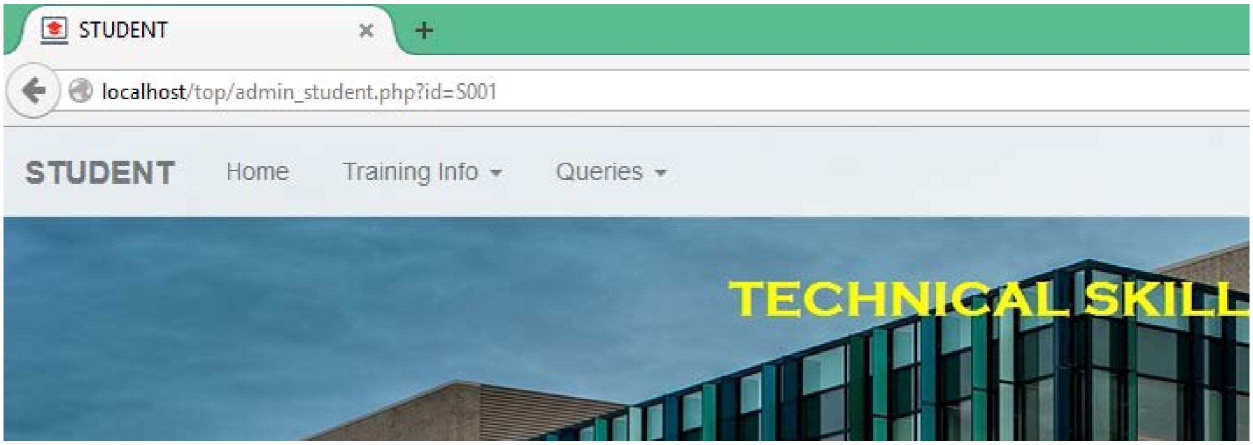
login_student.php

- **Input Data :** MailId as UserName and CellNo as password
- **Functionality:** Data Entry Form that takes the UserName and password of Student and passes it to login form.
- **SQL Query:** "select * from student wheres_email='$UserName' and s_cellno='$Password'";



admin_student.php

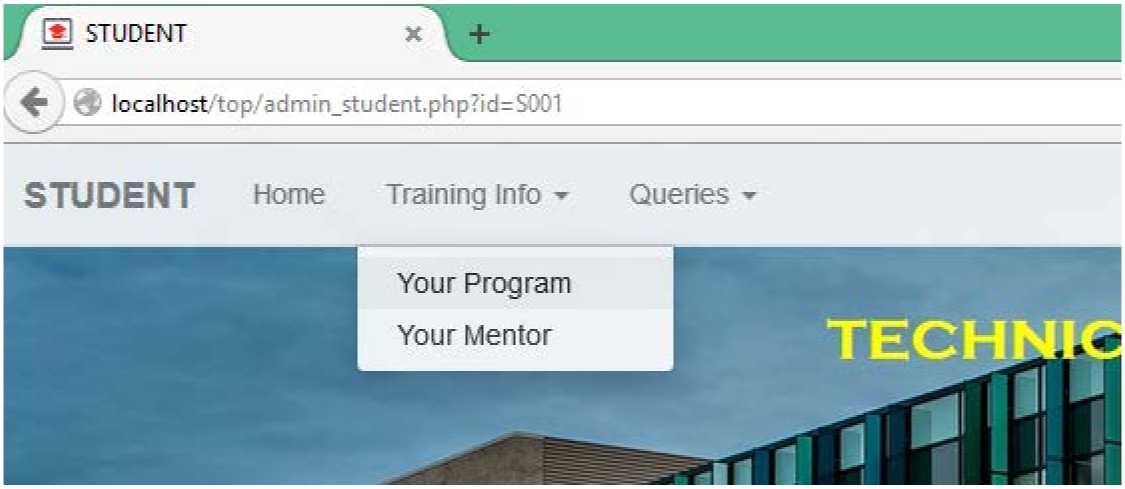
- **Output Data:** If the above login is successful, then user comes to welcome page of student .It has the below functionalities from its menu.



6.View Program Details

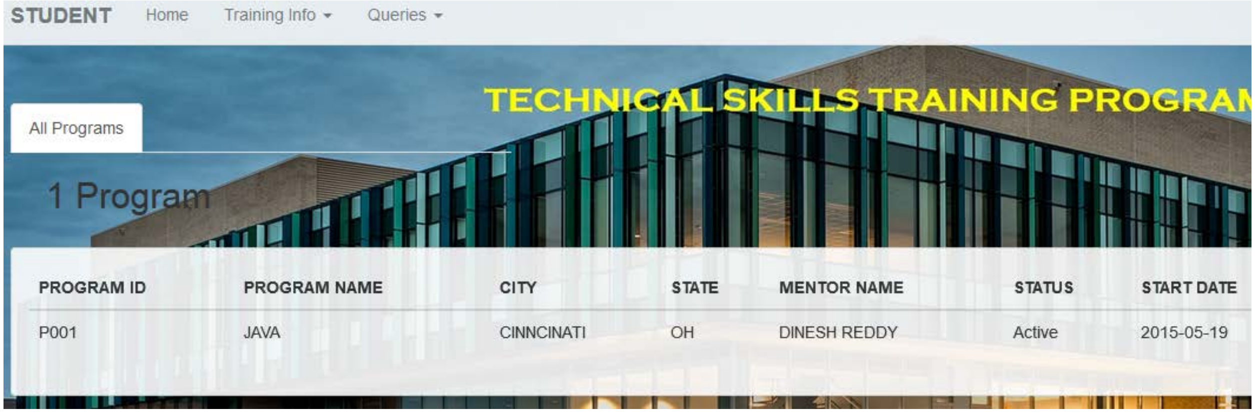
admin_Student.php

- **Input Data :** Student id will be passed in this panel
- **Functionality:** This page fetches the student program details from student_program.php
- **SQL Query:** " SELECT * FROM PROGRAM WHERE P_ID IN(SELECT P_ID FROM ATTENDS WHERE S_ID = '$id'";



student_program.php

- **Output Data:** Student Program details will be displayed from PROGRAM table as below.



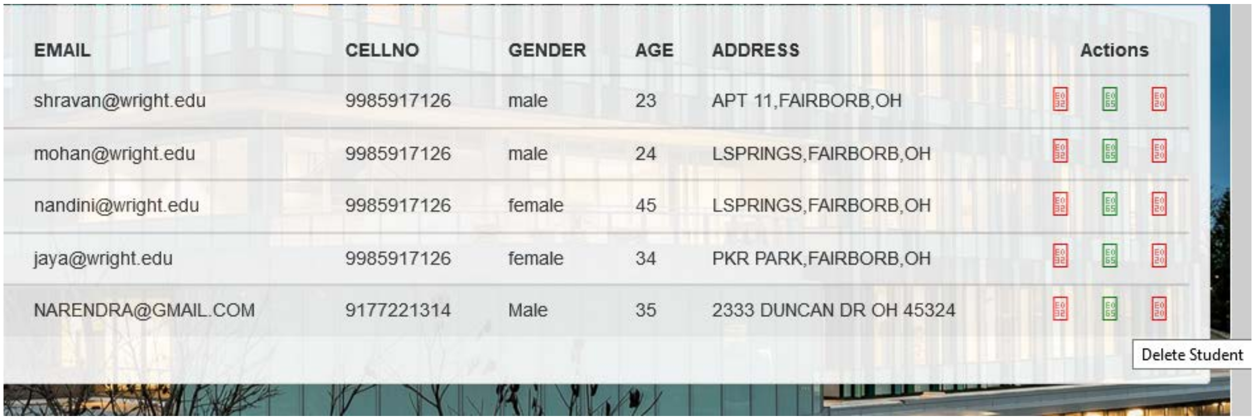
1 Program

PROGRAM ID	PROGRAM NAME	CITY	STATE	MENTOR NAME	STATUS	START DATE
P001	JAVA	CINNCINATI	OH	DINESH REDDY	Active	2015-05-19

7.Drop Student Profile

Student_list.php

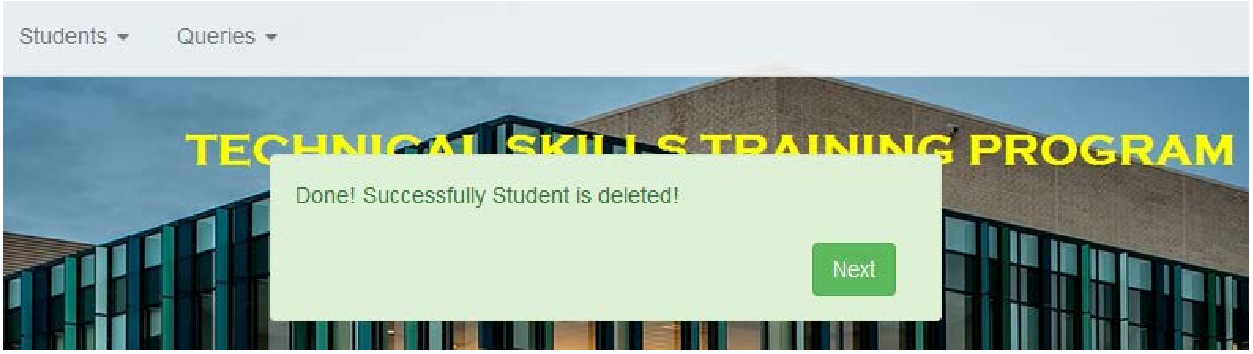
- **Input Data :** Mentor id will be passed for security acess
- **Functionality:** This page fetches the all student details and show the options such as view, update ,delete to the Mentor. Then User clicks on delete student option to delete the student record.
- **SQL Query:**
  "DELETE FROM `ATTENDS` WHERE `S_ID` = '$_GET[id]'";
  "DELETE FROM `QUESTION` WHERE `S_ID` = '$_GET[id]'";
  "DELETE FROM `STUDENT` WHERE `S_ID` = '$_GET[id]'";



EMAIL	CELLNO	GENDER	AGE	ADDRESS	Actions
shravan@wright.edu	9985917126	male	23	APT 11,FAIRBORB,OH	
mohan@wright.edu	9985917126	male	24	LSPRINGS,FAIRBORB,OH	
nandini@wright.edu	9985917126	female	45	LSPRINGS,FAIRBORB,OH	
jaya@wright.edu	9985917126	female	34	PKR PARK,FAIRBORB,OH	
NARENDRA@GMAIL.COM	9177221314	Male	35	2333 DUNCAN DR OH 45324	

student_stat.php

- **Output Data:** All the referencing tables data will be deleted before deleting the student details from STUDENT table



Done! Successfully Student is deleted!

TRANSACTIONS involving OLAP Analysis

Director will log into this application and does the analysis of cube w.r.t dimension tables.

Using Drill Down Approach

- User can see the analysis results of total profit by State, county, category and Gender.
- User can also see the analysis results of Total Profit by center, program, and student.

Using Slicing Approach

- User can see the analysis results of total profit by center, program, and student for a particular program.

Using Dicing Approach

- User can see the analysis results of total profit by center, program, and student for a particular program and in particular state.

Using Roll-up Approach

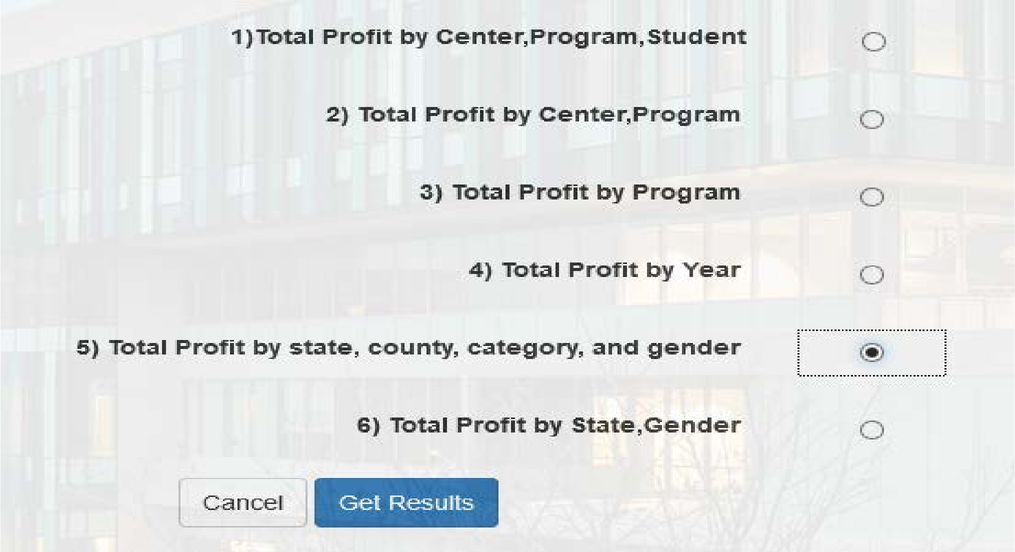
- User can see the analysis results of total profit by State, county, category and Gender.
- User can also see the analysis results of Total Profit by center, program, and student.

1. Drill-Down Approach Total Profit by State, county, category and Gender

Search_results.php

- **Input Data :** Radio button option 5 will be passed to other page.
- **Functionality:** This page fetches the details of total profit by State, county, category and Gender.
- **SQL Query:**

```
$sql = "select state, county, category, gender, sum(profit) as Total_Profit
        from Enroll_Fact F, Center_Dim S, Program_Dim I, Student_Dim C
        where F.CenterID = S.CenterID and F.ProgID = I.ProgID and F.studID = C.studID
        group by state, county, category, gender with rollup";
```



1)Total Profit by Center,Program,Student ○

2) Total Profit by Center,Program ○

3) Total Profit by Program ○

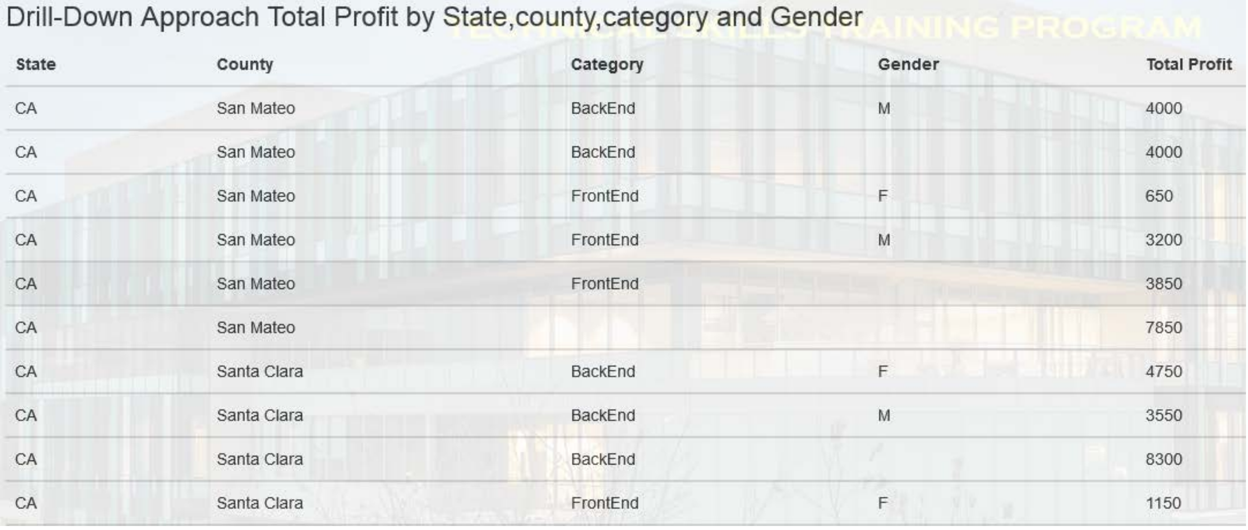
4) Total Profit by Year ○

5) Total Profit by state, county, category, and gender ●

6) Total Profit by State,Gender ○

results.php

- **Output Data:**



Drill-Down Approach Total Profit by State,county,category and Gender

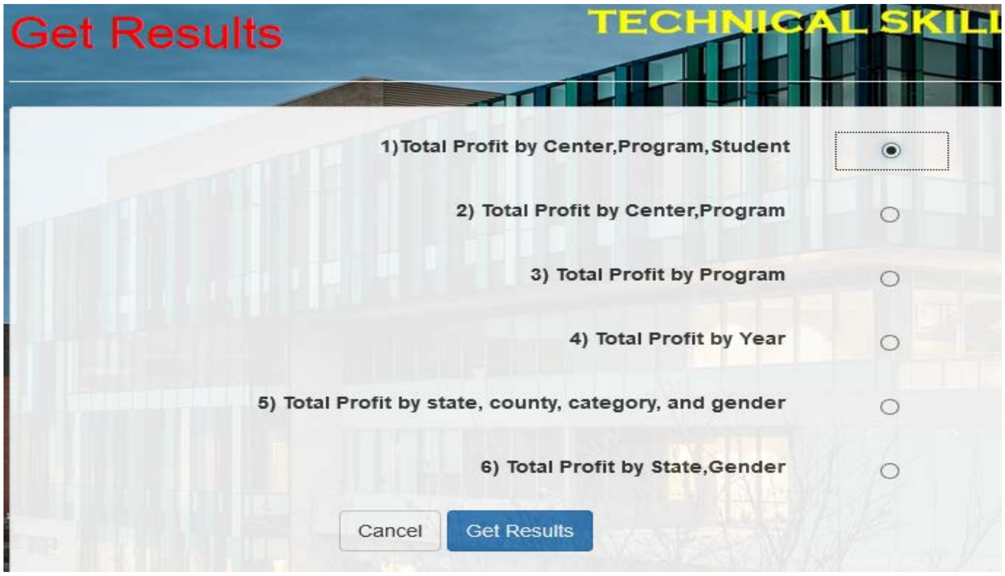
State	County	Category	Gender	Total Profit
CA	San Mateo	BackEnd	M	4000
CA	San Mateo	BackEnd		4000
CA	San Mateo	FrontEnd	F	650
CA	San Mateo	FrontEnd	M	3200
CA	San Mateo	FrontEnd		3850
CA	San Mateo			7850
CA	Santa Clara	BackEnd	F	4750
CA	Santa Clara	BackEnd	M	3550
CA	Santa Clara	BackEnd		8300
CA	Santa Clara	FrontEnd	F	1150

2. Drill-Down Approach Total Profit by center, program, and student

Search_results.php

- **Input Data :** Radio button option 1 will be passed to other page.
- **Functionality:** This page fetches the details of total profit by center, program, and student
- **SQL Query:**

```
$sql = " select centerID, progID, studID, sum(Total_Profit)
          as Total_Profit from cube
          group by centerID, progID, studID ";
```



1)Total Profit by Center,Program,Student ●

2) Total Profit by Center,Program ○

3) Total Profit by Program ○

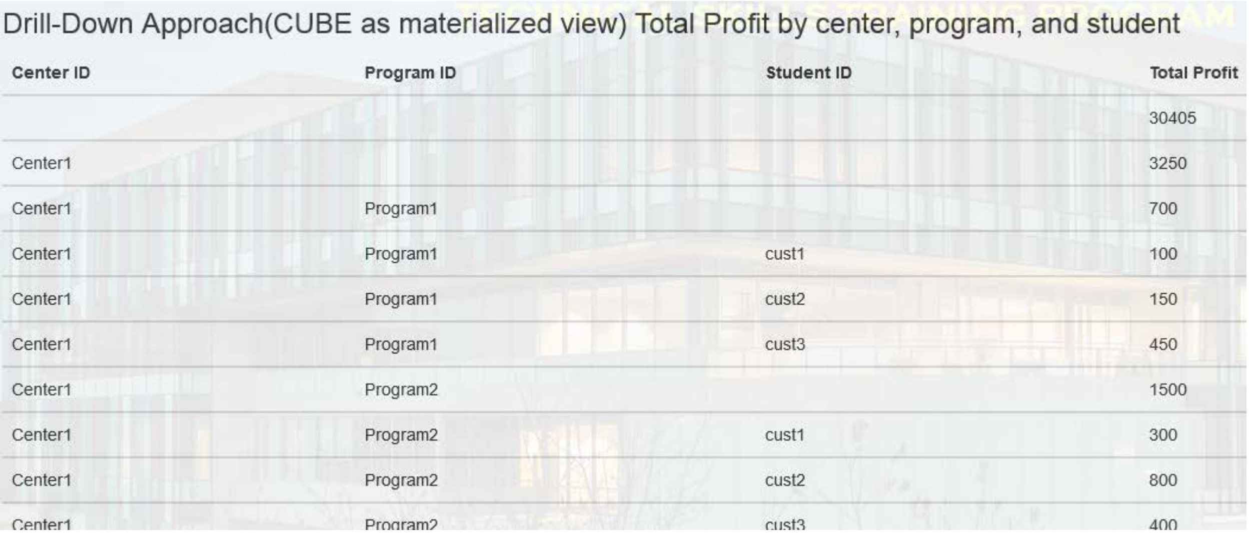
4) Total Profit by Year ○

5) Total Profit by state, county, category, and gender ○

6) Total Profit by State,Gender ○

results.php

- **Output Data:**



Drill-Down Approach(CUBE as materialized view) Total Profit by center, program, and student

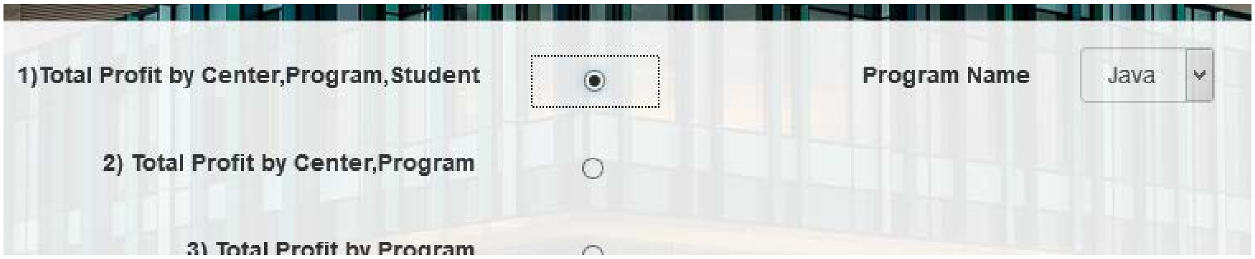
Center ID	Program ID	Student ID	Total Profit
			30405
Center1			3250
Center1	Program1		700
Center1	Program1	cust1	100
Center1	Program1	cust2	150
Center1	Program1	cust3	450
Center1	Program2		1500
Center1	Program2	cust1	300
Center1	Program2	cust2	800
Center1	Program2	cust3	400

3. Slicing- Approach Total Profit by center, program, and student for a particular program

Search_results.php

- **Input Data :** Radio button option 8 will be passed to other page.
- **Functionality:** This page fetches the details of total profit by center, program, and student for java program only.
- **SQL Query:**

```
$sql = " select F.centerID, I.progID, studID, sum(Total_Profit) as Total_Profit
          from cube F, Program_Dim I
          where F.progID = I.progID and progname = '$b'
         group by F.centerID, I.progID, studID ";
```



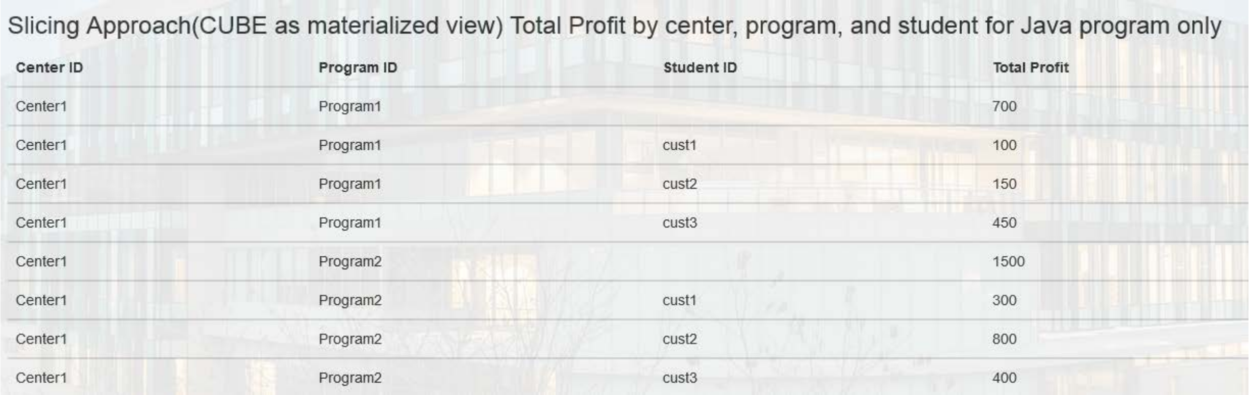
1)Total Profit by Center,Program,Student ● Program Name Java

2) Total Profit by Center,Program ○

3) Total Profit by Program ○

results.php

- **Output Data:**



Slicing Approach(CUBE as materialized view) Total Profit by center, program, and student for Java program only

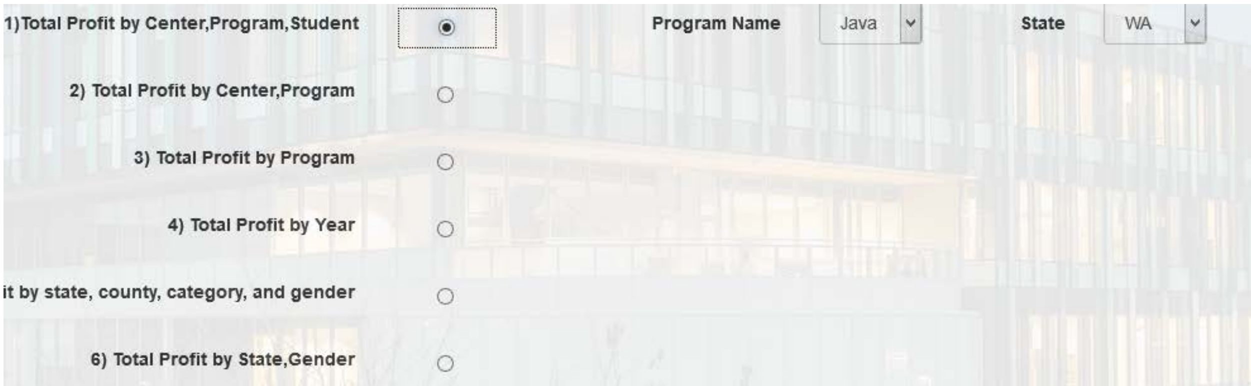
Center ID	Program ID	Student ID	Total Profit
Center1	Program1		700
Center1	Program1	cust1	100
Center1	Program1	cust2	150
Center1	Program1	cust3	450
Center1	Program2		1500
Center1	Program2	cust1	300
Center1	Program2	cust2	800
Center1	Program2	cust3	400

4. **Dicing- Approach Total Profit by center, program, and student for a particular program in particular state.**

Search_results.php

- **Input Data :** Radio button option 9 will be passed to other page.
- **Functionality:** This page fetches the details of total profit by center, program, and student for java program and in WA State.
- **SQL Query:**

```
$sql = " select F.centerID, I.progID, studID, sum(Total_Profit) as Total_Profit
             from cube F, Center_Dim S, Program_Dim I
            where F.centerID = S.centerID and F.progID = I.progID
            and state = '$c' and progname = '$b'
             group by F.centerID, I.progID, studID";
```



1)Total Profit by Center,Program,Student ● Program Name Java State WA

2) Total Profit by Center,Program ○

3) Total Profit by Program ○

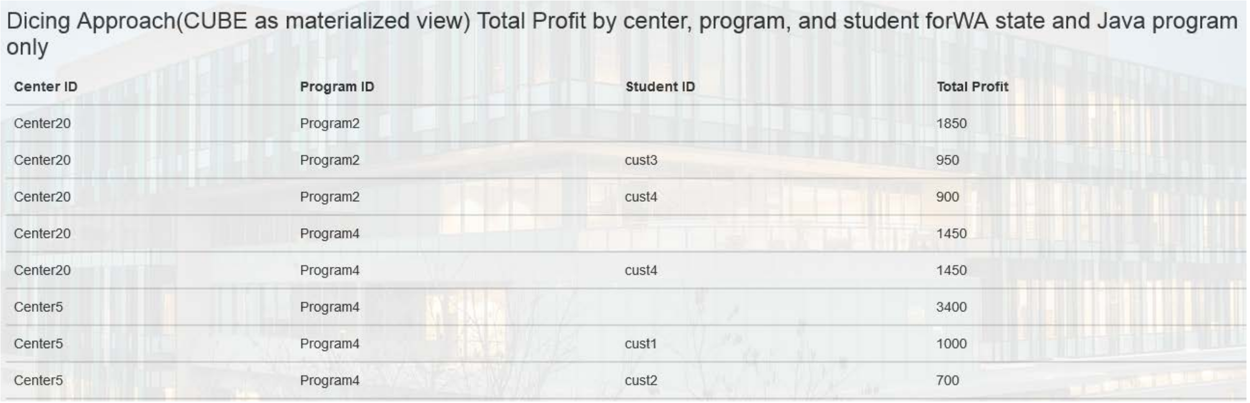
4) Total Profit by Year ○

it by state, county, category, and gender ○

6) Total Profit by State,Gender ○

results.php

- **Output Data:**



Dicing Approach(CUBE as materialized view) Total Profit by center, program, and student forWA state and Java program only

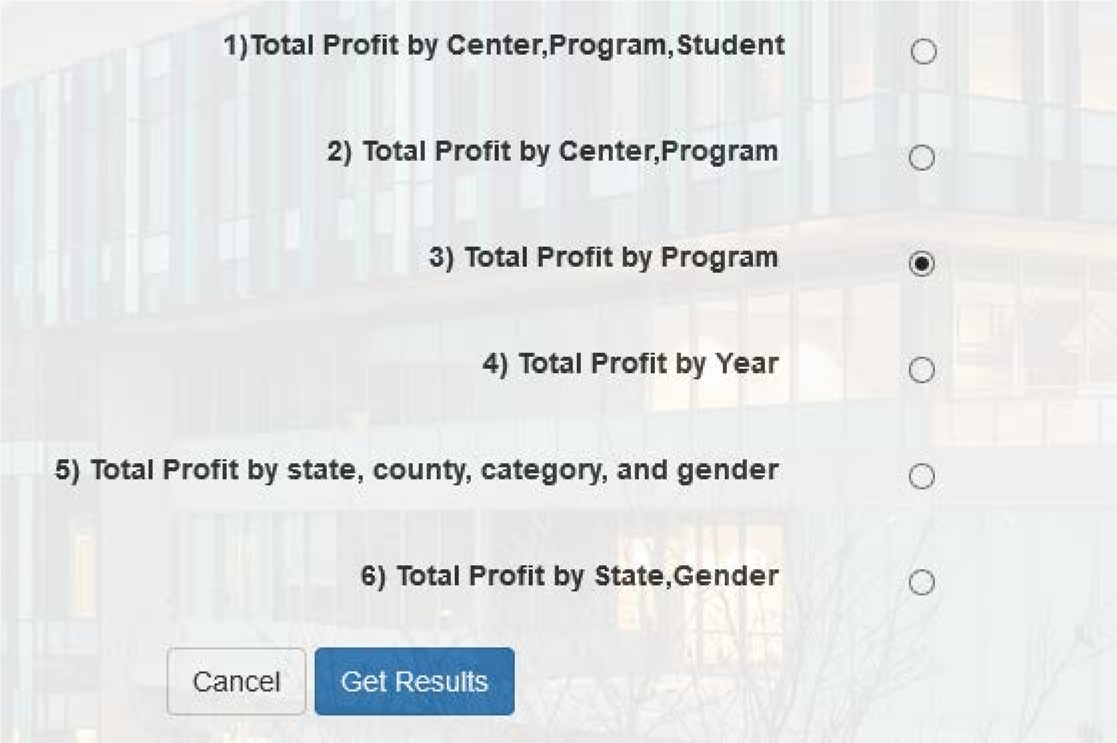
Center ID	Program ID	Student ID	Total Profit
Center20	Program2		1850
Center20	Program2	cust3	950
Center20	Program2	cust4	900
Center20	Program4		1450
Center20	Program4	cust4	1450
Center5	Program4		3400
Center5	Program4	cust1	1000
Center5	Program4	cust2	700

5. **Roll-Up Approach -Total Profit by program**

Search_results.php

- **Input Data :** Radio button option 3 will be passed to other page.
- **Functionality:** This page fetches the details of total profit by center, program, and student
- **SQL Query:**

```
$sql = " select progID, sum(Total_Profit) as Total_Profit from cube group by progID";
```



1)Total Profit by Center,Program,Student ○

2) Total Profit by Center,Program ○

3) Total Profit by Program ●

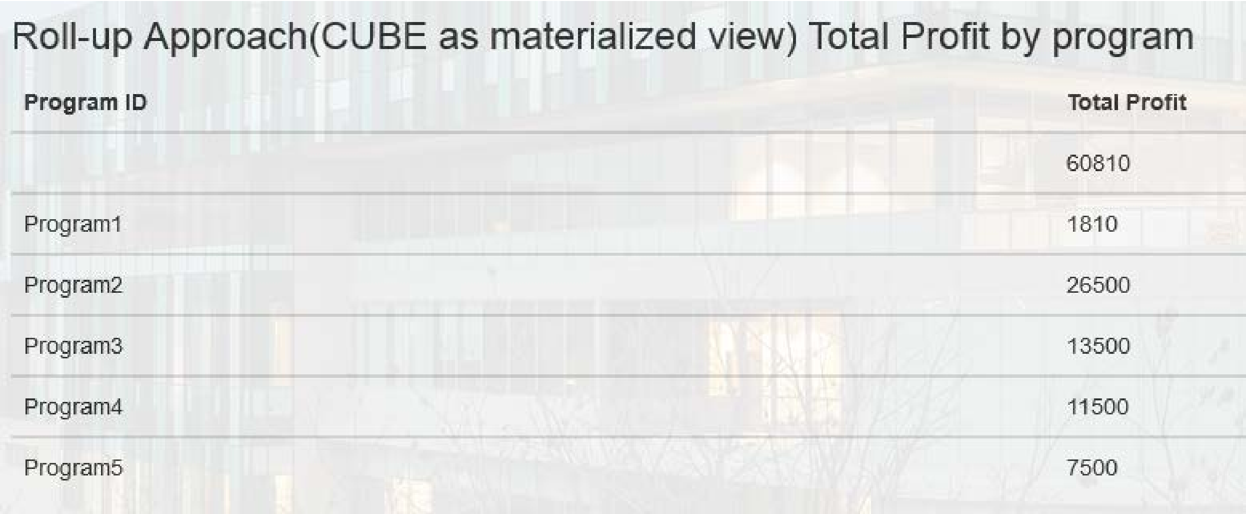
4) Total Profit by Year ○

5) Total Profit by state, county, category, and gender ○

6) Total Profit by State,Gender ○

results.php

- **Output Data:**



Roll-up Approach(CUBE as materialized view) Total Profit by program

Program ID	Total Profit
	60810
Program1	1810
Program2	26500
Program3	13500
Program4	11500
Program5	7500

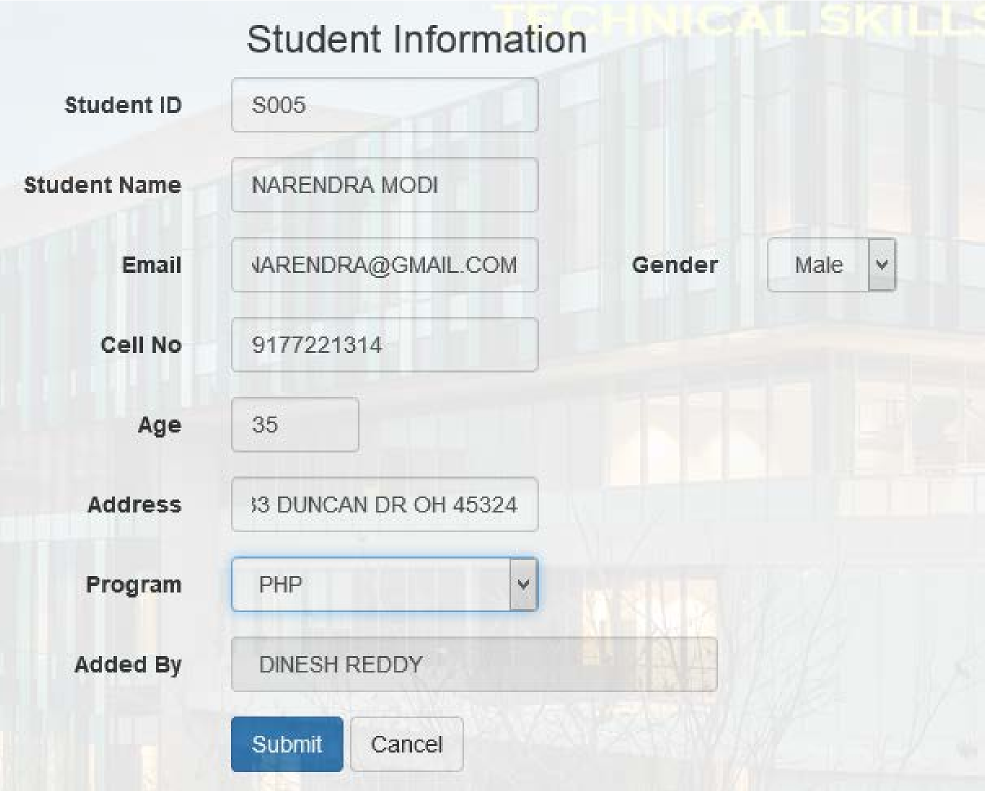
*The remaining transactions can be found in the program source code and also from the applications.

TRIGGER IMPLEMENTATION

I have implemented the trigger called 'BEFORE_INSERT_STUDENT' .It's purpose is to insert the current date in the Added_Date column of Student table. In other words, whenever the mentor tries to add the student profile from the application, then the above mentioned trigger will insert the current date for Added_Date column of STUDENT table. As part of trigger implementation, this column is added to the STUDENT table. It is implemented in create_student.php program.

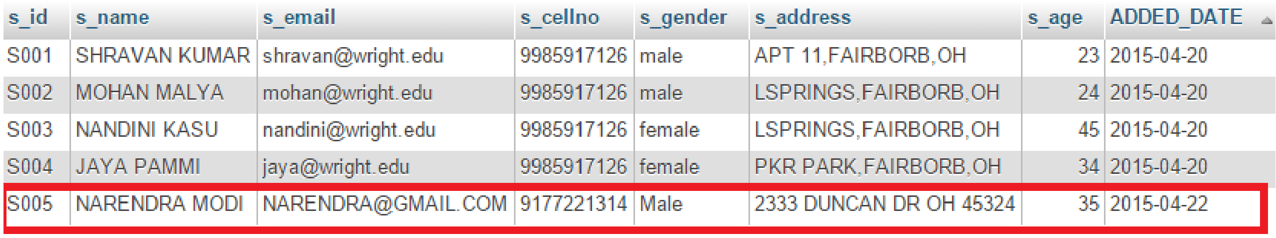
Trigger Syntax:

```
CREATE TRIGGER `tstp`.`BEFORE_INSERT_STUDENT`

BEFORE INSERT ON `tstp`.`student` FOR EACH ROW

 SET NEW.ADDED_DATE = NOW();
```



Student Information

Student ID	S005		
Student Name	NARENDRA MODI		
Email	NARENDRA@GMAIL.COM	Gender	Male
Cell No	9177221314		
Age	35		
Address	33 DUNCAN DR OH 45324		
Program	PHP		
Added By	DINESH REDDY		



s_id	s_name	s_email	s_cellno	s_gender	s_address	s_age	ADDED_DATE
S001	SHRAVAN KUMAR	shravan@wright.edu	9985917126	male	APT 11,FAIRBORB,OH	23	2015-04-20
S002	MOHAN MALYA	mohan@wright.edu	9985917126	male	LSPRINGS,FAIRBORB,OH	24	2015-04-20
S003	NANDINI KASU	nandini@wright.edu	9985917126	female	LSPRINGS,FAIRBORB,OH	45	2015-04-20
S004	JAYA PAMMI	jaya@wright.edu	9985917126	female	PKR PARK,FAIRBORB,OH	34	2015-04-20
S005	NARENDRA MODI	NARENDRA@GMAIL.COM	9177221314	Male	2333 DUNCAN DR OH 45324	35	2015-04-22

QUERY PROCESSING

We can observe the same results using different queries in database. By considering system performance, we have to use the best queries among them. It is nothing but tuning the queries based on the requirements.

I have used this in some programs while retrieving the attributes in the selection queries. Instead of using * in the select query, I used attributes for some of the queries and the same can be verified in my source code.

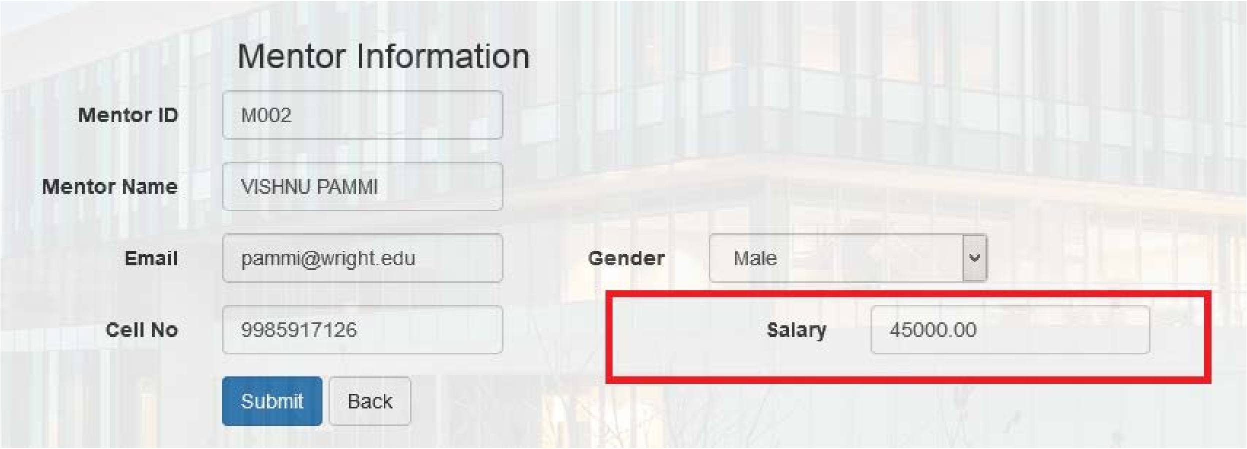
TEMPORAL DATA MANAGEMENT

A first step in the temporal database is to time stamp the data. Such that different database states can be distinguished. In my project I used this concept to store the history of Mentor. The mentor profile will be updated by Director when ever he/she wants to increase the Mentor Salary or other personal information. If we use the MENTOR_HISTORY table to record all these different states, then it will be helpful to review the changes that are implemented for Mentor. So, I included the 'valid start time' and 'valid end time' for the time constraints of the data. It is implemented while adding the Mentor profile and also updating the same profile.

In the below figures, we can observe that mentor id'M002' salary is updated from 31000$ to 45000$ in the application and then these changes were recorded in Mentor_History table.

Transaction Details: It is implemented in **edit_mentor.php** and **add_mentor.php** programs

Input: Login using Director Profile and then click on update Mentors tab



Mentor Information

Mentor ID: M002

Mentor Name: VISHNU PAMMI

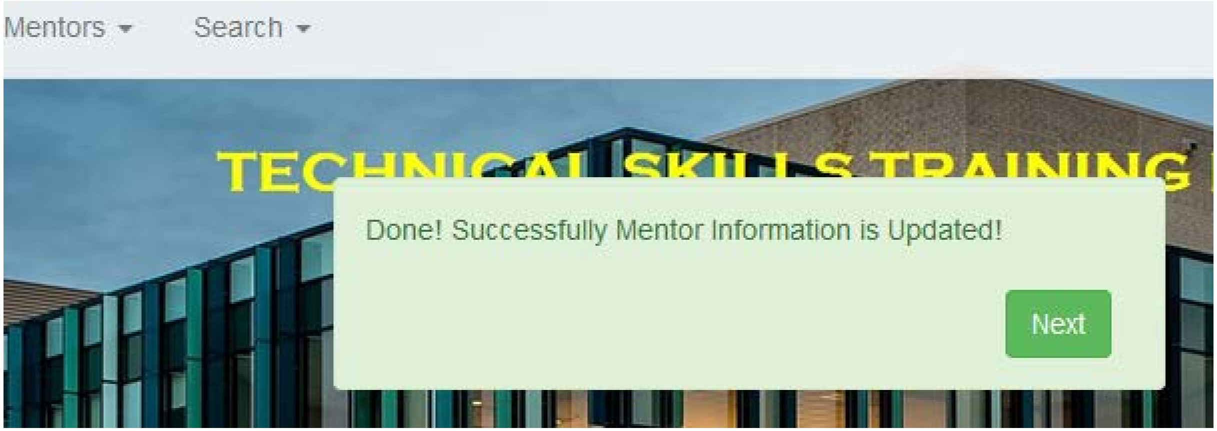
Email: pammi@wright.edu

Gender: Male

Cell No: 9985917126

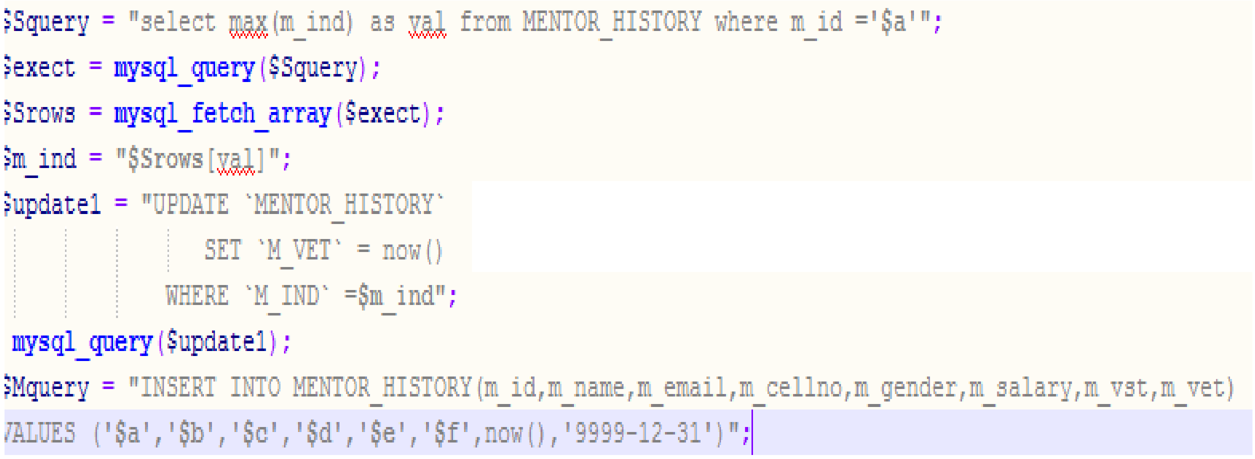
Salary: 45000.00

Output:



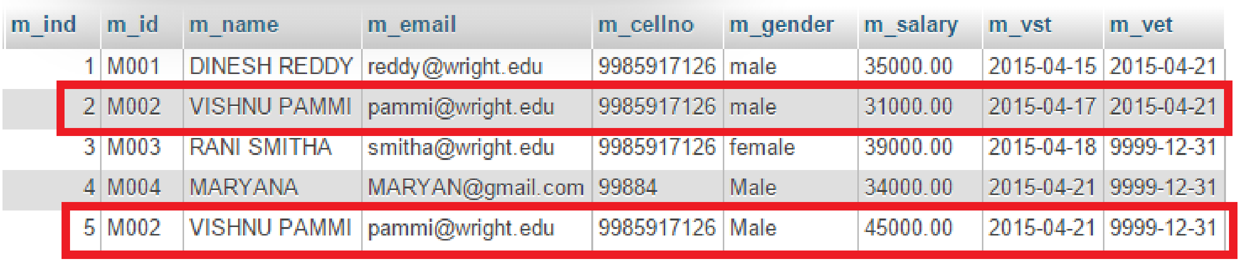
Done! Successfully Mentor Information is Updated!

The same can be observed in the below tables using bitemporal database management concept.



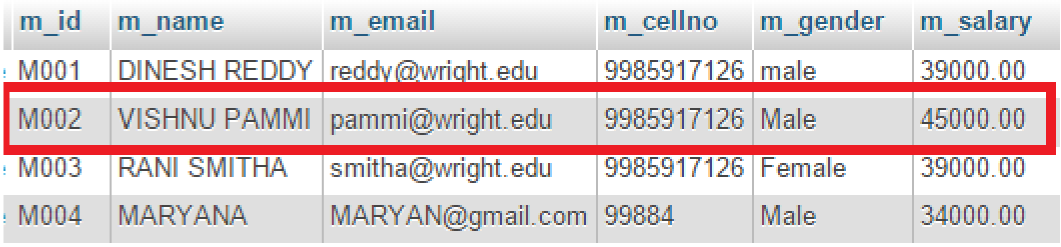
```
$Squery = "select max(m_ind) as val from MENTOR_HISTORY where m_id ='$a'";
$exect = mysql_query($Squery);
$Srows = mysql_fetch_array($exect);
$m_ind = "$Srows[val]";
$update1 = "UPDATE `MENTOR_HISTORY`
                SET `M_VET` = now()
             WHERE `M_IND` =$m_ind";
 mysql_query($update1);
$Mquery = "INSERT INTO MENTOR_HISTORY(m_id,m_name,m_email,m_cellno,m_gender,m_salary,m_vst,m_vet)
VALUES ('$a','$b','$c','$d','$e','$f',now(),'9999-12-31')";
```

MENTOR_HISTORY



m_ind	m_id	m_name	m_email	m_cellno	m_gender	m_salary	m_vst	m_vet
1	M001	DINESH REDDY	reddy@wright.edu	9985917126	male	35000.00	2015-04-15	2015-04-21
2	M002	VISHNU PAMMI	pammi@wright.edu	9985917126	male	31000.00	2015-04-17	2015-04-21
3	M003	RANI SMITHA	smitha@wright.edu	9985917126	female	39000.00	2015-04-18	9999-12-31
4	M004	MARYANA	MARYAN@gmail.com	99884	Male	34000.00	2015-04-21	9999-12-31
5	M002	VISHNU PAMMI	pammi@wright.edu	9985917126	Male	45000.00	2015-04-21	9999-12-31

MENTOR



m_id	m_name	m_email	m_cellno	m_gender	m_salary
M001	DINESH REDDY	reddy@wright.edu	9985917126	male	39000.00
M002	VISHNU PAMMI	pammi@wright.edu	9985917126	Male	45000.00
M003	RANI SMITHA	smitha@wright.edu	9985917126	Female	39000.00
M004	MARYANA	MARYAN@gmail.com	99884	Male	34000.00

SYSTEM AND LIMITATIONS

This project is a web-based application and its primary purpose is to provide an environment for the Director, Mentor and Student to create programs, profiles, attends for technical skills program and also to discuss about the programs in the Question forums. Also, this application will help the director and Mentor to visualize the analysis of organization growth and its profit w.r.t certain parameters.

This application has computed successfully and also tested by taking the test-cases. It has a required options, which can be utilized by the different levels of users to perform the desired operations.

The Goals that are achieved by this application are:

- Instant access and user friendly
- Record are managed efficiently.
- Operations covered lot of the areas w.r.t Training centers of programming.
- Portable and flexible for further enhancement.

This application is implemented using PHP as Frontend and MYSQL as back end in windows environment. has computed successfully and also tested by taking the test-cases.

Limitations on the Design and Implementation:

- It would have been better if I take the Birth date instead of age. But currently I am taking age. If I take Birthrate as attribute(input field), then using trigger I can calculate the individual age and insert into age attribute.
- I thought of adding the rating section between Student and Mentor objects. So that Students can give their feedback to the Mentor regarding the program skills session or other admin related issues. But, I have not included this relation also in my design and did not implemented it.

Limitations on Performance and Functionality of DBMS:

- In ADS class, I studied and learnt about Query optimization. But, I didn't try to verify whether the joined Queries that I used in my project are optimized or not. If it is verified, then system performance can be improved when we have more data in DB system. Instead I have performed optimization on few select queries.

Future Topics:

- I can make this *Technical Skills Training Program Database Application* available online. Such that each student can create the account on their own .Thus everyone will get access to the data which is relevant to them. And also, the ads can be embedded into my web application using yellow pages.
- We can use this application for further OLAP results like how many students are coming to particular program in a globally and which technology is in high demand.