Admission Management System

- An Admission management system is a software to automate the admission process and reduce the dependency on traditional methods of enrolling students. This eliminates paperwork and improve accuracy in admission procedure.
- An University has number of campuses providing certain courses through admission cell with staff.
- Which then have cmp_name and cmp_code an entity as of Campus Code.
- An Admission in each Campus is done according to different criteria's given for a student like merit-based, Marks cut-off, and reservations for each course.
- Each student is carried out as Student_name, Student_id, Student_course.
- the staff in admission cell have the following attributes such as Add_off.
- Each course is associated with different course_name and course_code.
- The courses may be grouped within different departments. which then entitled as dep_name.
- After getting successfully admitted into the institution, the student gets an unique identification number which uniquely signifies that particular student as nominal_roll.
- There may be more domain to categorise a student wrt. Quota and class that he has been admitted, likely NRI Students, Management Quota, Scholarship Students, Merit_Based Students, etc.

ENTITIES

Person: STUDENTS, ADMIN, FACULTY
Place: CAMPUSES
Object: WEBSITE
Event: REGISTRATION, ENROLLMENT, RESERVATION
Concept: CREDENTIALS, COURSE, MERITS, DEPARTMENTS

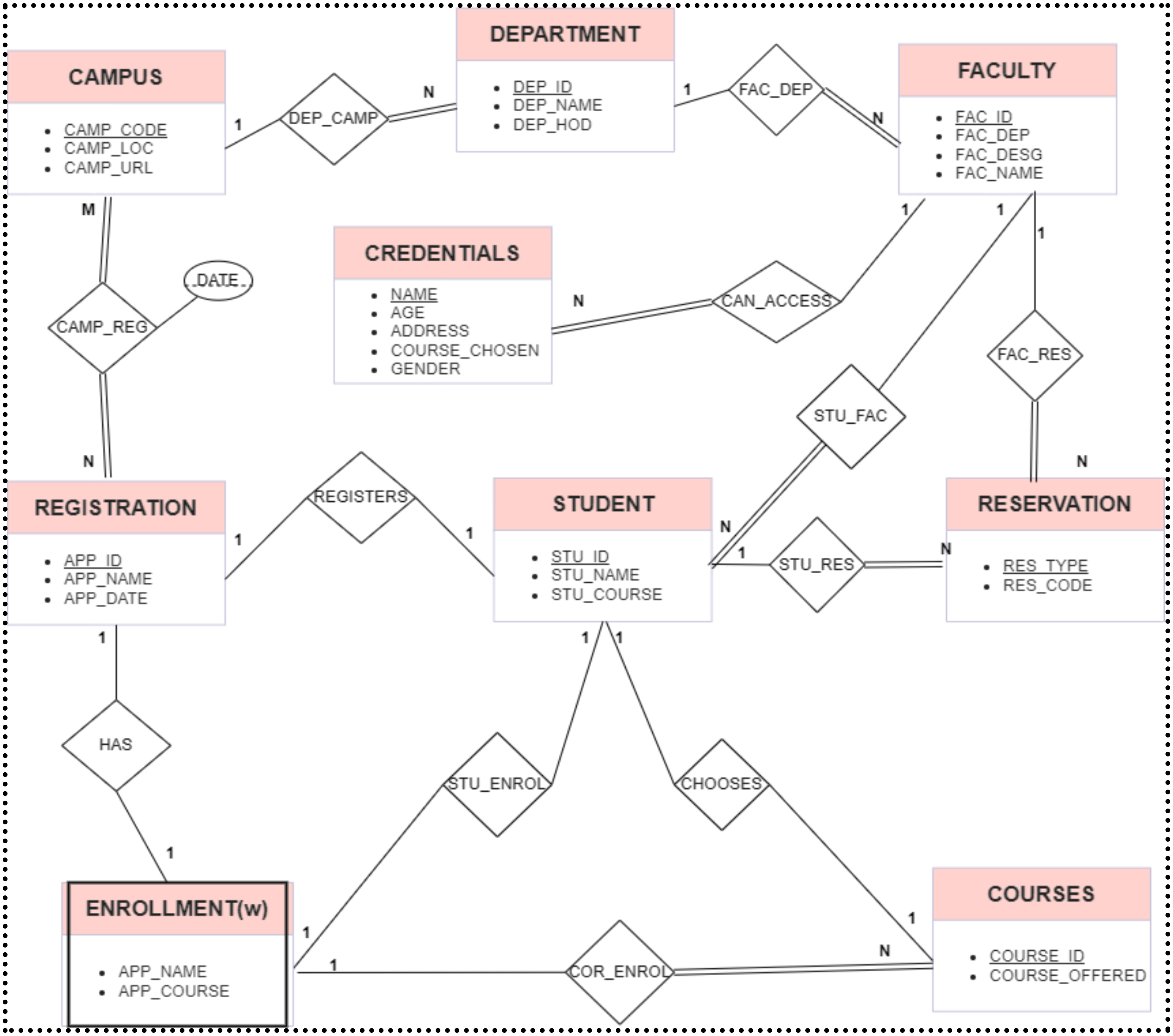
ATTRIBUTES FOR MY PROBLEM STATEMENT:

ENTITY TYPE	ATTRIBUTE	TYPE
● STUDENT	● STUDENT_NAME	● SIMPLE
	● STUDENT_ID	● SIMPLE
	● STUDENT_COURSE	● COMPOSITE
● DEPARTMENT	● DEP_ID	● SINGLE
	● DEP_NAME	● SIMPLE
	● DEP_HOD	● COMPOSITE
● FACULTY	● FAC_NAME	● COMPOSITE
	● FAC_DEP	● SIMPLE
	● FAC_DSGN	● SINGLE
● CAMPUS	● CAMP_CODE	● SINGLE
	● CAMP_ID	● SINGLE
	● CAMP_ADD	● COMPOSITE
	● CAMP_URL	● COMPOSITE
● REGISTRATION	● APP_ID	● SINGLE
	● APP_NAME	● COMPOSITE
	● APP_DATE	● SIMPLE
● ENROLLMENT	● APP_NAME	● COMPOSITE
	● APP_COURSE	● SIMPLE
● RESERVATION	● RES_TYPE	● SIMPLE
	● RES_CODE	● SINGLE
	●	●
● CREDENTIALS(RAW)	● NAME	● COMPOSITE
	● AGE	● SIMPLE
	● ADDRESS	● SIMPLE
	● COURSE_CHOSEN	● SIMPLE
	● GENDER	● SINGLE
	● S_NO	● SINGLE
	●	●

Relation for my Entities:

Person: STUDENTS, ADMIN, FACULTY
Place: CAMPUSES
Object: WEBSITE
Event: REGISTRATION, ENROLLMENT, RESERVATION
Concept: CREDENTIALS, COURSE, MERITS, DEPARTMENTS

ER DIAGRAM:



Entities/Tables:

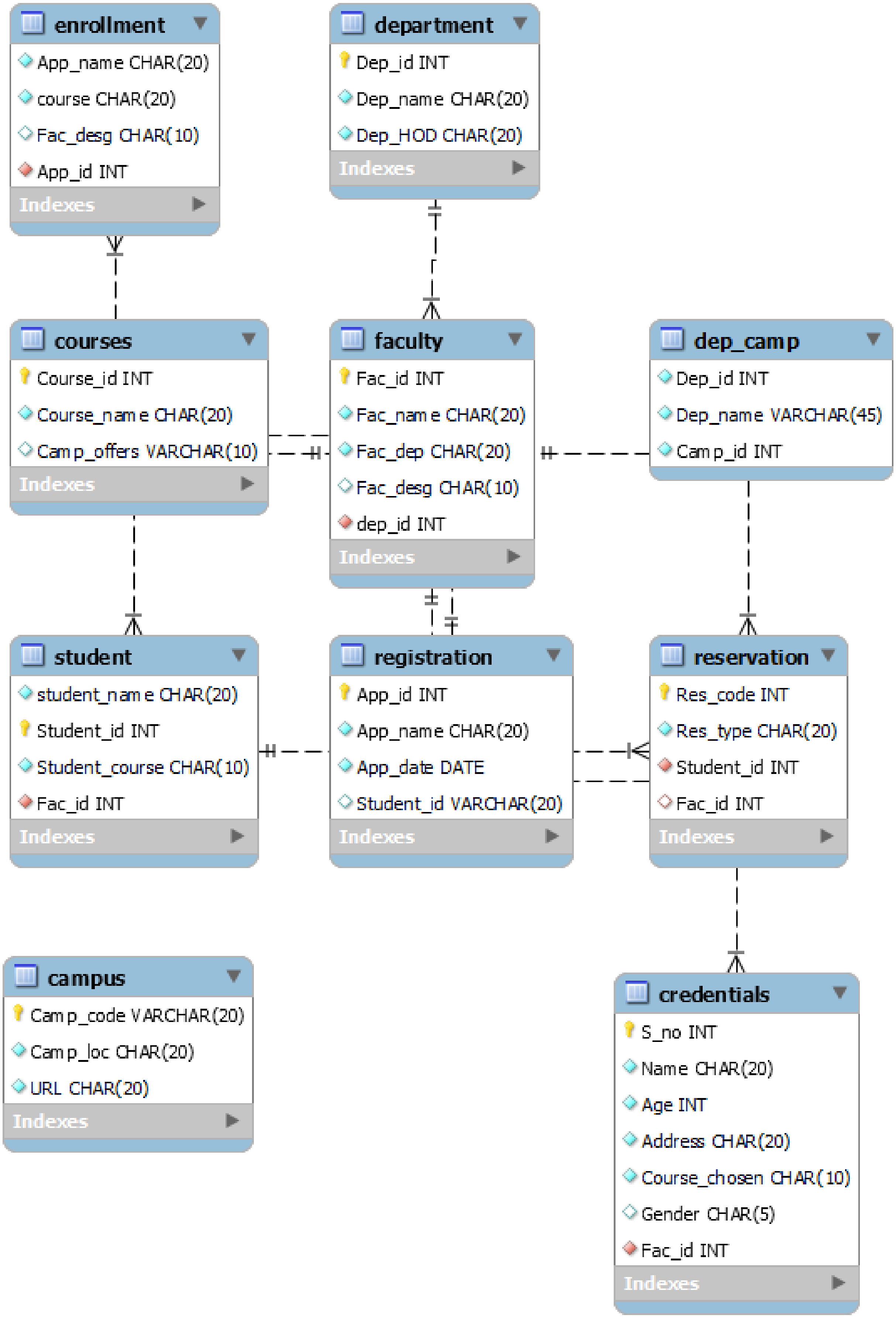
- Campus: with attributes(Code, Location, URL)
    - Camp-Reg: (Camp_code, App_id, date){**descriptive attribute**}
- Registration: with attributes(ID, Name)
- Department: with attributes (Id, Name, HOD)
- Faculty: with attributes (ID,Name, Department, Designation)
- Credentials: with attributes(Name, Age, Address, Course_chosen, Gender)
- Student: with attributes (Name, ID, course)
- Reservation: with attributes (Type, Code)
- Enrollment: with attributes(Name, Course){WEAK}
- Course: with attributes (ID, Offered Course)

The relationship sets design listed above:

- DEP_CAMP : Relating DEPARTMENT to CAMPUS with N:1 relation . Where one campus has N departments and access through N departments.
- CAMP_REG: Descriptive attribute relating CAMPUS to REGISTRATION with M:N relation, whose attributes are the primary keys of CAMPUS to REGISTRATION and an additional for distinctness as DATE . Where both the entities have complete participation.
- FAC_DEP: Relating FACULTY to DEPARTMENT with N:1 . Where one department can have utmost N faculty but a faculty must be in at least one department.
- CAN_ACCESS: Relating FACULTY to CREDENTIALS with 1:N relation. FACULTY can access through N credentials, with partial participation.
- STU_FAC: Relating STUDENT to FACULTYwith 1:N relation. Where FACULTY can go through STUDENT details, with partial participation.
- FAC_RES: Relating FACULTY to RESERVATION with 1:N relation. Where FACULTY can regulate the reservations, with partial participation.
- STU_RES: Relating STUDENT to RESERVATION with 1:N relation. Where STUDENTS can go through the reservation type and include that in application.
- REGISTERS: STUDENT registers into College whose details will be stored in the REGISTRATION table.

- STU_ENROL: Enrolling STUDENT into the different Courses, Where one student can enroll into only one course.
- CON_ENROL:
- CHOOSES: A Student chooses a course.

ER-Diagram: (Work-bench designed)


enrollment
App_name CHAR(20)
course CHAR(20)
Fac_desg CHAR(10)
App_id INT
Indexes
department
Dep_id INT
Dep_name CHAR(20)
Dep_HOD CHAR(20)
Indexes
courses
Course_id INT
Course_name CHAR(20)
Camp_offers VARCHAR(10)
Indexes
faculty
Fac_id INT
Fac_name CHAR(20)
Fac_dep CHAR(20)
Fac_desg CHAR(10)
dep_id INT
Indexes
dep_camp
Dep_id INT
Dep_name VARCHAR(45)
Camp_id INT
student
student_name CHAR(20)
Student_id INT
Student_course CHAR(10)
Fac_id INT
Indexes
registration
App_id INT
App_name CHAR(20)
App_date DATE
Student_id VARCHAR(20)
Indexes
reservation
Res_code INT
Res_type CHAR(20)
Student_id INT
Fac_id INT
Indexes
campus
Camp_code VARCHAR(20)
Camp_loc CHAR(20)
URL CHAR(20)
Indexes
credentials
S_no INT
Name CHAR(20)
Age INT
Address CHAR(20)
Course_chosen CHAR(10)
Gender CHAR(5)
Fac_id INT
Indexes

Queries:-

Query_no	Query	mysql_Query	Relational Algebraic Query
1	SELECT BDATE, ADDRESS FROM EMPLOYEE WHERE FNAME='John' AND MINIT='B' AND LNAME='Smith'	SELECT App_date, App_id FROM registration WHERE App_name = 'raj';	
2	SELECT FNAME, LNAME, ADDRESS FROM EMPLOYEE, DEPARTMENT WHERE DNAME='Research' AND DNUMBER=DNO	SELECT Name, Address FROM credentials, students WHERE Course_chosen = 'B.com' AND name = student_name;	
3	SELECT PNUMBER, DNUM, LNAME, BDATE, ADDRESS FROM PROJECT, DEPARTMENT, EMPLOYEE WHERE DNUM=DNUMBER AND MGRSSN=SSN AND PLOCATION='Stafford'	SELECT Student_id, Camp_loc, Age, Address FROM student, campus, credentials WHERE name = student_name AND Address='palampur';	
4	SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM EMPLOYEE E S WHERE E.SUPERSSN=S.SSN	SELECT Name ,Fac_name , Course_chosen, Fac_id FROM credentials, faculty WHERE credentials.Fac_id = faculty.Fac_id ;	
5	SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM EMPLOYEE AS E, EMPLOYEE AS S WHERE E.SUPERSSN=S.SSN	SELECT student_id, Student_name, Student_course, Fac_name FROM student as s, faculty as f	

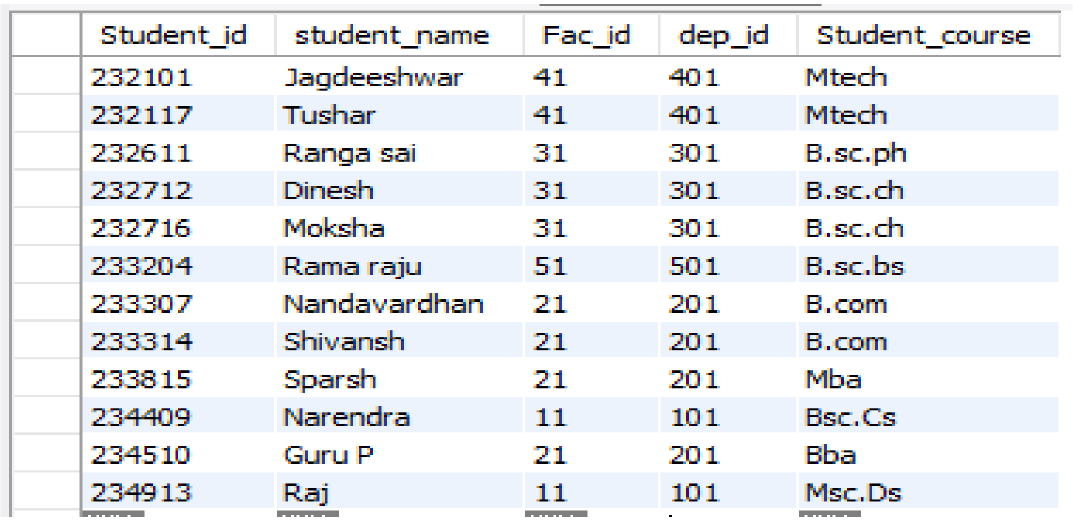
		WHERE s.Fac_id = f.Fac_id	
6	SELECT SSN FROM EMPLOYEE	SELECT Dep_HOD FROM department;	
7	SELECT SSN, DNAME FROM EMPLOYEE, DEPARTMENT	SELECT Fac_name , Dep_HOD FROM faculty , DEPARTMENT WHERE faculty.dep_id = departme	
8	SELECT * FROM EMPLOYEE WHERE DNO=5	SELECT * FROM faculty WHERE Fac_desg = 'hod'	
9	SELECT SALARY FROM EMPLOYEE	SELECT Fac_desg , Fac_dep FROM faculty;	
10	SELECT * FROM EMPLOYEE, DEPARTMENT WHERE DNAME='Research' AND DNO=DNUMBER	SELECT * FROM student , reservation WHERE res_code = '1001' AND reservation.Student_id = student.Student_id;	
11	SELECT DISTINCT SALARY FROM EMPLOYEE	SELECT DISTINCT Address FROM credentials;	
12	(SELECT PNAME FROM PROJECT, DEPARTMENT, EMPLOYEE WHERE DNUM=DNUMBER AND MGRSSN=SSN AND LNAME='Smith') UNION (SELECT PNAME FROM PROJECT, WORKS_ON, EMPLOYEE WHERE PNUMBER=PNO AND ESSN=SSN AND LNAME='Smith')	(SELECT name FROM student, credentials WHERE Student_course = Course_chosen and Student_name = Name) UNION (SELECT PNAME FROM Student, Credentials, Faculty WHERE Student.Fac_id = Faculty.Fac_id and Fac_desg = "hod");	

13	SELECT FNAME, LNAME, ADDRESS FROM EMPLOYEE WHERE DNO IN (SELECT DNUMBER FROM DEPARTMENT WHERE DNAME='Research')	SELECT NAME, ADDRESS FROM credentials WHERE Fac_id IN (SELECT Fac_id FROM Faculty WHERE Fac_desg = "Mhod" and dep_id = 301);	
14	SELECT E.FNAME, E.LNAME FROM EMPLOYEE AS E WHERE E.SSN IN (SELECT ESSN FROM DEPENDENT WHERE ESSN=E.SSN AND E.FNAME=DEPENDENT_NAME)	SELECT course_id, camp_offers FROM courses AS C WHERE C.camp IN (SELECT Camp_offers FROM campus WHERE Camp_code = "400MDH"	-
15	SELECT E.FNAME, E.LNAME FROM EMPLOYEE E, DEPENDENT D WHERE E.SSN=D.ESSN AND E.FNAME=D.DEPENDENT_NAME	SELECT Student_id, Fac_id, Camp_offers FROM student, faculty, courses WHERE Course.Course_name = Student.Student_course;	
17	SELECT FNAME, LNAME FROM EMPLOYEE WHERE EXISTS (SELECT * FROM DEPENDENT WHERE SSN=ESSN AND FNAME=DEPENDENT_NAME)	SELECT student_name FROM student WHERE EXISTS (SELECT * FROM credentials WHERE Student_name = name)	
18	SELECT FNAME, LNAME FROM EMPLOYEE WHERE NOT EXISTS (SELECT * FROM DEPENDENT WHERE SSN=ESSN)	SELECT Student_name FROM student WHERE NOT EXISTS (SELECT * FROM enrollmenr WHERE student_name =	

		App_name)	
19	SELECT DISTINCT ESSN FROM WORKS_ON WHERE PNO IN (1, 2, 3)	SELECT DISTINCT age FROM credentials ;	
20	SELECT FNAME, LNAME FROM EMPLOYEE WHERE SUPERSSN IS NULL	SELECT App_id , App_name FROM registration WHERE Student_id IS NULL;	
21	SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM EMPLOYEE E S WHERE E.SUPERSSN=S.SSN	SELECT age, gender FROM credentials ;	
22	SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM (EMPLOYEE E LEFT OUTER JOIN EMPLOYEES ON E.SUPERSSN=S.SSN)	-	
23	SELECT FNAME, LNAME, ADDRESS FROM EMPLOYEE, DEPARTMENT WHERE DNAME='Research' AND DNUMBER=DNO	SELECT Student_id, Res_code, Res_type FROM student, reservation WHERE Stuent.student_id = reservation.Student_id;	
24	SELECT FNAME, LNAME, ADDRESS FROM (EMPLOYEE JOIN DEPARTMENT ON DNUMBER=DNO) WHERE DNAME='Research’		
25	SELECT COUNT (*) FROM EMPLOYEE	SELECT COUNT (*) FROM faculty	
26	SELECT FNAME, LNAME FROM EMPLOYEE WHERE ADDRESS LIKE '%Houston,TX%’	SELECT student_name FROM student WHERE Student_course LIKE "B%";	
27	SELECT FNAME, LNAME FROM EMPLOYEE WHERE BDATE LIKE '_______5_’	SELECT App_name, App_id FROM	

		registration WHERE App_date LIKE "____7____"	
28	SELECT FNAME, LNAME, 1.1*SALARY FROM EMPLOYEE, WORKS_ON, PROJECT WHERE SSN=ESSN AND PNO=PNUMBER AND PNAME='ProductX’	SELECT MAX(age), MIN(age), AVG(age) FROM credentials WHERE course_chosen like "M%";	
29	SELECT FNAME, LNAME, ADDRESS FROM (EMPLOYEE NATURAL JOIN DEPARTMENT AS DEPT(DNAME, DNO, MSSN, MSDATE) WHERE DNAME='Research’	SELECT campus_name FROM (department NATURAL JOIN campus AS dep_camp values (dep_id, dep_name, camp_id))	
30		select url from campus where Camp_code like "%00%";	

Table for Normalization :



Student_id	student_name	Fac_id	dep_id	Student_course
232101	Jagdeeshwar	41	401	Mtech
232117	Tushar	41	401	Mtech
232611	Ranga sai	31	301	B.sc.ph
232712	Dinesh	31	301	B.sc.ch
232716	Moksha	31	301	B.sc.ch
233204	Rama raju	51	501	B.sc.bs
233307	Nandavardhan	21	201	B.com
233314	Shivansh	21	201	B.com
233815	Sparsh	21	201	Mba
234409	Narendra	11	101	Bsc.Cs
234510	Guru P	21	201	Bba
234913	Raj	11	101	Msc.Ds

This relation is already in 1-NF form because no multivalued attribute(atomic).

The candidate keys for this table are : Student_id, Fac_id, dep_id

In this relations let the attributes named as

> A: Student_id

>B: Student_name

>C: Fac_id

>D: dep_id

>E: Student_course

The functional dependencies are:

{AC} -> {B,D,E}, {A}->{C}, {C} -> {D,E}

Here, A is already a subset of {AC}. It is causing a REDUNDANCY.

F = {AC to BDE} , BDE may have redundant values. So, they don't form the KEY.

Prime Attributes: A,C

Non-Prime Attributes: B,D,E

Checking for partial dependency:

A->B / A->D / A->E

C->B / C->D / C->E

Hence, No non-prime attributes are partially dependents as a part of composite keys

2nf Decomposition:

A->B / A->D / A->E

C->B / C->D / C->E

All the above 6 functional dependencies are creating problem because non-prime attributes are partially dependent on the candidate key.

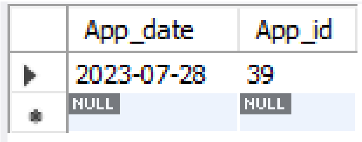
SQL Questions and Queries :

1. Get the applicant Id and the date of application of the applicant whose name is “raj”.

   SELECT App_date, App_id

   FROM registration

   WHERE App_name = 'raj';



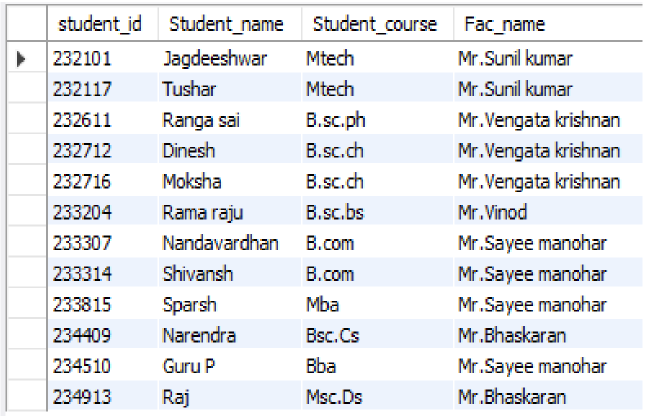
App_date	App_id
2023-07-28	39
NULL	NULL

2. Get student name, Student Id and Student course along with the name of the faculty name who is incharge of that particular department that he/she belongs to.

   SELECT student_id, Student_name, Student_course, Fac_name

   FROM student as s, faculty as f

   WHERE s.Fac_id = f.Fac_id;



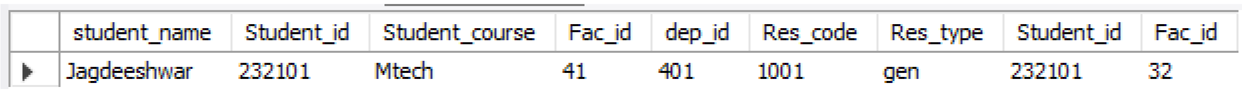
student_id	Student_name	Student_course	Fac_name
232101	Jagdeeshwar	Mtech	Mr.Sunil kumar
232117	Tushar	Mtech	Mr.Sunil kumar
232611	Ranga sai	B.sc.ph	Mr.Vengata krishnan
232712	Dinesh	B.sc.ch	Mr.Vengata krishnan
232716	Moksha	B.sc.ch	Mr.Vengata krishnan
233204	Rama raju	B.sc.bs	Mr.Vinod
233307	Nandavardhan	B.com	Mr.Sayee manohar
233314	Shivansh	B.com	Mr.Sayee manohar
233815	Sparsh	Mba	Mr.Sayee manohar
234409	Narendra	Bsc.Cs	Mr.Bhaskaran
234510	Guru P	Bba	Mr.Sayee manohar
234913	Raj	Msc.Ds	Mr.Bhaskaran

3. Get all the details of the students/students who belongs to the reservation code “1001”.

   SELECT *

   FROM student , reservation

   WHERE res_code = '1001' AND reservation.Student_id = student.Student_id;



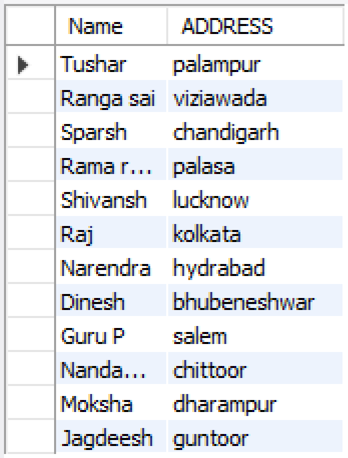
student_name	Student_id	Student_course	Fac_id	dep_id	Res_code	Res_type	Student_id	Fac_id
Jagdeeshwar	232101	Mtech	41	401	1001	gen	232101	32

4. Get name and address of the students whose incharge faculty’s designation is Management Head of the Department and has the department Id ‘301’.

```
 SELECT Name, ADDRESS

FROM credentials

WHERE Fac_id IN (SELECT Fac_id FROM Faculty
WHERE Fac_desg = "Mhod" and dep_id = 301 );
```

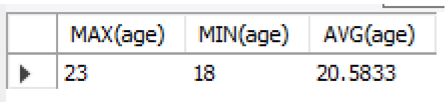


	Name	ADDRESS
▶	Tushar	palampur
	Ranga sai	viziawada
	Sparsh	chandigarh
	Rama r...	palasa
	Shivansh	lucknow
	Raj	kolkata
	Narendra	hydrabad
	Dinesh	bhubeneshwar
	Guru P	salem
	Nanda...	chittoor
	Moksha	dharampur
	Jagdeesh	guntoor

5. Find the minimum, maximum and average age of the students who has applied for admission to this university.

```
SELECT MAX(age), MIN(age), AVG(age)

 FROM credentials
```



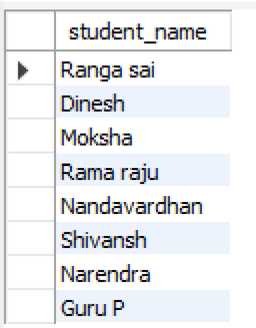
	MAX(age)	MIN(age)	AVG(age)
▶	23	18	20.5833

6. Get the names of all the students who has enrolled for UG program using LIKE operator.

```
SELECT student_name

FROM student

WHERE Student_course LIKE "B%";
```



	student_name
▶	Ranga sai
	Dinesh
	Moksha
	Rama raju
	Nandavardhan
	Shivansh
	Narendra
	Guru P