DBMS Mini Project (College Analysis)

Submission Date - 29th June 2021

Submitted to

Dr. Debanjan Sadhya

Submitted By:-

Amit Kumar (2020-BCS-009)
Gaurav K Sukhramani (2020-BCS-033)
Gurpreet Singh (2020-BCS-036)
Kailash Kejriwal (2020-BCS-044)
Pratham Mishra (2020-BCS-060)
Shrey Pokhriyal (2020-BCS-069)

➢ CONTENTS:-

➢ Overview

IIITs have been a center of attraction for many JEE aspirants. However, during the post-exam counselling, several students get confused and different questions arise in their minds,like

- “Will I get a college of my choice at my rank?”
- “What would be the best college at my rank?”
- “Will I be able to afford to study in a college?”
- “Which colleges offer good placements?”
- ”How is the faculty in a college?”

and the list goes on.

Also, everyone has a distinct preference and taste. So,the criteria for choosing a proper college will vary from one student to another.

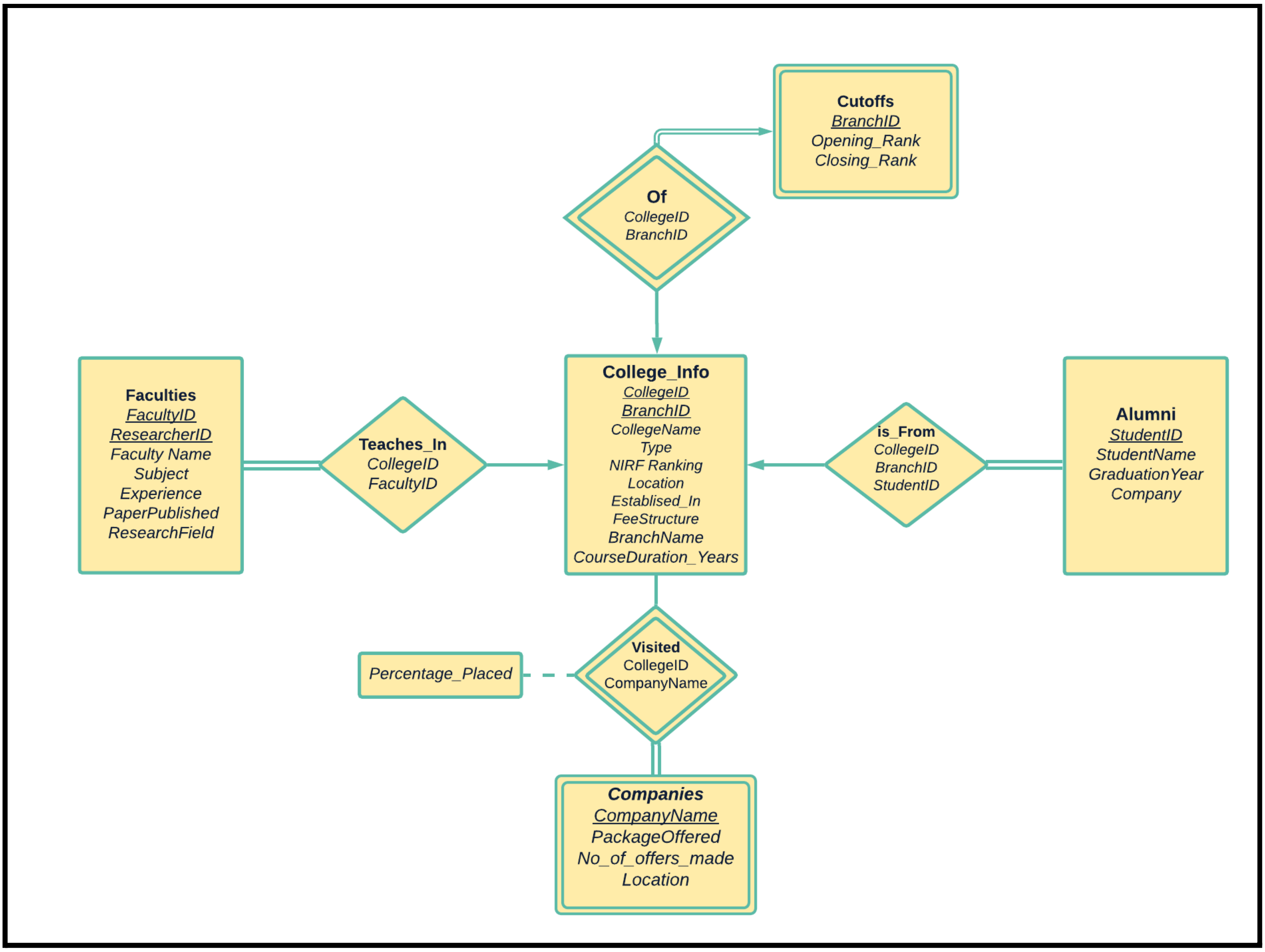
And, our database solves this problem. We have brought our database up to date, which contains almost the entire data required to make the necessary decisions, and help a student find the perfect college for him.

Usefulness

- The database would be tremendously helpful in choosing a college based on preferences
- It can also show which college offers better placements compared to the other colleges.
- It would also help the student to determine if he will be able to afford the college he chooses to study.
- It will also show how is the research culture of a particular college

This data will be sufficient enough for a student to determine a college of his choice. Now, let us have an analysis of the database and the data in it.

➢ ER DIAGRAM


Cutoffs
BranchID
Opening_Rank
Closing_Rank
Of
CollegeID
BranchID
Faculties
FacultyID
ResearcherID
Faculty Name
Subject
Experience
PaperPublished
ResearchField
Teaches_In
CollegeID
FacultyID
College_Info
CollegeID
BranchID
CollegeName
Type
NIRF Ranking
Location
Establised_In
FeeStructure
BranchName
CourseDuration_Years
is_From
CollegeID
BranchID
StudentID
Alumni
StudentID
StudentName
GraduationYear
Company
Percentage_Placed
Visited
CollegeID
CompanyName
Companies
CompanyName
PackageOffered
No_of_offers_made
Location

➢ Entity Sets:

- **College_Info :**

➔ (CollegeID , BranchID , CollegeName , Type , NIRF , Ranking , Location , Establised_In , FeeStructure , BranchName , CourseDuration_Years)

➔ Here CollegeID and BranchID together form the candidate key.Remaining attributes give us detailed information about the college.

- **Cutoffs: (Weak entity set)**

➔ (BranchID , Closing_Rank , Opening_Rank)

➔ Where BranchID is the discriminator. CR is the closing rank and OR is the opening rank for a particular branch of a college.

- **Alumni:**

➔ (StudentID , StudentName , GraduationYear , Company)

➔ StudentID is the candidate key, hence it can never be 'NULL' and is unique. Other attributes give us information about students.

- **Companies: (Weak entity set)**

➔ (CompanyName , PackageOffered , No_of_offers_made , Location)

➔ Where CompanyName is the discriminator. Remaining attributes give us general information like location of company, offers made by the company to students etc.

- **Faculties:**

➔ (FacultyID , ResearcherID , FacultyName , Subject , Experience ,PaperPublished , ResearchField)

➔ FacultyID and ResearcherID combined is the candidate key. Remaining attributes describe the faculty's experience and qualification in their respective fields.

➢ Relationship sets and their Cardinality:

1)is_from:

(CollegeID , BranchID , StudentID)

➔ **'is_from'** is a **One to Many relationship** from 'College_Info' to 'Alumni' as an alumnus belongs to a single college and one college can have several alumni. Here 'Alumni' shows **total participation** as every alumnus belongs to a unique college whereas 'College_Info' shows **partial participation**, if the college has recently started, thus it would not have any alumni.

2)Visited:

(CollegeID , CompanyName , Percentage_Placed)

➔ **'Visited'** is a **Many to Many relationship** between 'College_Info' and 'Companies' as many companies approach various colleges with job opportunities. Here the entity set 'companies' shows **total participation** as it consists of data about companies which have approached at least one college. Whereas 'College_Info' shows **partial participation** as a company does not visit all the colleges present in the database. 'Percentage_Placed' is an additional attribute which stores the percentage of students placed in past academic year.

3)Teaches_In:

(CollegeID , FacultyID)

➔ **'Teaches_In'** is a **Many to One relationship** from 'Faculties' to 'College_Info' respectively as one college can have multiple faculties but a faculty cannot be from multiple colleges. Moreover, 'Faculties' shows **total participation** as every faculty belongs to a college but 'College_Info' shows **partial participation** as a faculty of a specific college from 'College_info' may not be listed in the faculty table.

4)Of:

(CollegeID , BranchID)

➔ **‘Of’** is a **One to One relationship** from ‘Cutoffs’ to ‘College_Info’ as a college and its branch will have a specific cutoff rank (opening and closing rank). ‘Cutoffs’ shows **total participation** as every pair of opening and closing rank is for a specific branch of a college. But ‘College_Info’ shows **partial participation** as there can exist a tuple of CollegeID and BranchID whose cutoff rank isn’t mentioned in ‘Cutoffs’.

➢ Converting Entity Sets to Tables:

1)As Alumni is on many side and College_Info is on one side of many to one relationship and total participation is on many side, so in the relationship set (‘is_From’) between ‘Alumni’ and ‘College_Info, the attributes ‘CollegeID & BranchID’ will be included in Alumni as a foreign key . In short,leaf entity set and relationship set made a combined to make a single table and the Alumni table will be as follows:

- ➔ **Alumni** :
  (CollegeID , BranchID , StudentID , StudentName , GraduationYear , Company)

****

2)As Companies is a **weak entity set**, so the attribute ‘CollegeID’ is included in Companies as a foreign key with CompanyName as ‘discriminator’, such that (CollegeID, CompanyName) collectively makes a primary key for the Companies entity set. Thus, the Companies table will be as follows :

- ➔ **Companies** :
  (CollegeID , CompanyName , PackageOffered , No_of_offers_made , Location , Percentage_Placed)

****

3)As Cutoffs is a **weak entity set**, so the attribute ‘CollegeID’ is included in Companies as a foreign key and BranchID as ‘discriminator’, such that the Companies table will be as follows :

- ➔ **Cutoffs** :
  (CollegeID , BranchID , Closing_Rank , Opening_Rank)

****

4)As Faculties is on many side and College_Info is on one side of many to one relationship and total participation is on many side, so in the relationship set ('Teaches_In') between 'Faculties' and 'College_Info', the attributes 'CollegeID' will be included in Faculties as a foreign key and the Faculties table will be re-named as "Teaches_In". In short, leaf entity set and relationship set made a combined to make a single table and the table is defined as follows:

➔ **Teaches_In** :
(CollegeID , FacultyID , ResearcherID , FacultyName , Subject , Experience , PaperPublished , ResearchField)

****

5)As all the leaf entity sets have made a table joining with their respective relationship sets and the remaining entity set is College_Info which is defined as :

➔ **College_Info** :
(CollegeID , BranchID , CollegeName , Type , NIRF_Ranking , Location , Establised_In , FeeStructure , BranchName , CourseDuration_Years)

****

➢ FUNCTIONAL DEPENDENCIES :-

Alumni:

➔ (CollegeID , BranchID , StudentID , StudentName , GraduationYear , Company)

CollegeID , BranchID , StudentID -> StudentName , GraduationYear , Company
StudentID(P)->StudentName,GraduationYear,Company

****

Companies:

➔ (CollegeID , CompanyName , PackageOffered , No_of_offers_made , Location , Percentage_Placed)

CollegeID, CompanyName-> (PackageOffered , No_of_offers_made , Location , Percentage_Placed)
CompanyName->(Location)

****

CutOffs:

➔ (CollegeID , BranchID , Closing_Rank , Opening_Rank)

CollegeID,BranchID->(Closing_Rank,Opening_Rank)

****

Teaches_In:

➔ (CollegeID , FacultyID , ResearcherID , FacultyName , Subject , Experience , PaperPublished , ResearchField)

FacultyID,ResearcherID->Faculty Name , Subject , Experience , Paper_Published ResearchField
ResearcherID->Paper_Published , ResearchField

****

College_info:

- ➔ (CollegeID , BranchID , CollegeName , NIRF_Ranking , Location , Establised_In , FeeStructure , BranchName , CourseDuration_Year, Type)

CollegeID,BranchID->CollegeName , NIRF Ranking , Location , BranchName , Established_In , Fee_Structure , CourseDuration_Year , Type
CollegeID->CollegeName , NIRF Ranking , Established_In , Fee_Structure , Type , Location
BranchID -> BranchName , CourseDuration_Year

****

➢ DECOMPOSITION (using NORMALISATION):

Alumni :

(CollegeID,BranchID,StudentID) --- Candidate key as closure is closed

----

2NF:

Partial Dependencies : StudentID(P)->StudentName,GraduationYear,Company

Closure of attributes which gives PDs :

[StudentID(P)] += (StudentID(P),StudentName,GraduationYear,Company)

R1 (StudentID(P),StudentName,GraduationYear,Company)

FDs :

StudentID(P)->StudentName,GraduationYear,Company -- (Is is in 2NF as prime attribute is single attribute)

R2 (CollegeID(P),BranchID(P),StudentID(P))

FDs :

[empty] --- BCNF (Done)

3NF:

LHS has Superkey (StudentID) -- 3NF (Done)

BCNF:

LHS has Candidate key (StudentID) -- BCNF (Done)

Decomposed tables of Alumni:

R1 (StudentID(P),StudentName,GraduationYear,Company)

R2 (CollegeID(P),BranchID(P),StudentID(P))

****

Companies:

(CollegeID , CompanyName) -- is Candidate key as (Foreign key (CollegeID) which is primary key of strong entity set + Discriminator(CompanyName))

2NF:

Partial Dependencies : CompanyName->(Location)

Closure of attributes which gives PDs :

[CompanyName] += (CompanyName,Location)

R3 (CompanyName,Location)

FDs:

CompanyName->(Location) -- BCNF (Done) --Because LHS has Candidate key

R4(CollegeID,CompanyName,PackageOffered,No_of_offers_made,Percentage_Placed)

FDs:

CollegeID,CompanyName->PackageOffered,No_of_offers_made,Percentage_Placed

----

3NF:

In FD of R4 LHS has a super key, so it is in 3NF

BCNF:

In FD of R4 LHS has a candidate key, so it is in BCNF

Decomposed table of Companies:

R3 (CompanyName(P),Location)

R4(CollegeID , CompanyName ,PackageOffered,No_of_offers_made, Percentage_Placed)

Cutoffs:

(CollegeID,BranchID) -- Candidate key as closure is closed

2NF:

No Partial Dependencies , it is 2NF

3NF:

LHS has Super key, so it is in 3NF

BCNF:

LHS has Candidate key, so it is in BCNF

Decomposed tables of Cutoffs :

R5: (CollegeID,BranchID,Opening_Rank,Opening_Rank)

****

College_info:

(CollegeID,BranchID) -- Candidate key (As it is closed and minimum super key)

2NF:

Partial Dependencies :

CollegeID->CollegeName, Type, NIRF Ranking, Location, Established_In, Fee_Structure

BranchID -> BranchName, CourseDuration_Year

Closure of attributes which gives PDs :

[CollegeID]+ = (CollegeID,CollegeName, Type, NIRF Ranking, Location, Established_In, Fee_Structure)

[BranchID] + = (BranchID, BranchName, CourseDuration)

R81 (CollegeID,CollegeName, Type, NIRF Ranking, Location, Established_In, Fee_Structure)

FDs:

CollegeID->(CollegeID,CollegeName, Type, NIRF Ranking, Location, Established_In, Fee_Structure)

R9 (BranchID, BranchName, CourseDuration_Year)

FDs:

BranchID->(BranchID, BranchName, CourseDuration_Year)

R82 (BranchID, CollegeID)

FDs: {Empty} -- BCNF

3NF:

R81 and R9 has super keys on LHS of their FDs

So, they are in 3NF

BCNF:

R81 and R9 has candidate keys on LHS of their FDs

So, they are in BCNF

Total tables:

R81 (CollegeID,CollegeName, Type, NIRF Ranking, Location, Established_In, Fee_Structure)

R82 (BranchID, CollegeID)

R9 (BranchID, BranchName, CourseDuration_Year)

****

Teaches_In :

CollegeID,FacultyID,ResearcherID -- Candidate key (As it is closed and minimum super key)

2NF:

Partial Dependencies :

FacultyID,ResearcherID->Faculty Name, Subject, Experience ,Paper_Published, ResearchField

ResearcherID->Paper_Published, ResearchField

Closure of attributes which gives PDs :

[FacultyID,ResearcherID]+= FacultyID,ResearcherID,Faculty Name, Subject, Experience ,Paper_Published, ResearchField
[ResearcherID]+ = (ResearcherID,Paper_Published, ResearchField)

R7 (ResearcherID,Paper_Published, ResearchField)
FDs:
-ResearcherID->Paper_Published, ResearchField

R6 (FacultyID,ResearcherID,Faculty Name, Subject, Experience)
FDs:
-FacultyID,ResearcherID->,Faculty Name, Subject, Experience

R10 (CollegeID,FacultyID,ResearcherID)
FDs:{empty} --- BCNF

3NF:
R7, R6 and R10 both have super key on LHS of their FDs
So, they are in 3NF

BCNF:
R7, R6 and R10 both have candidate key on LHS of their FDs
So, they are in BCNF

Total tables:
R7 (ResearcherID,Paper_Published, ResearchField)
R6 (FacultyID,ResearcherID,Faculty Name, Subject, Experience)
R10 (CollegeID,FacultyID,ResearcherID)
****

➔ NAMING THE TABLES

R1 -- Alumni_info
R2 -- Studied_from
R3 -- Companies
R4 -- Placements
R5 -- Cutoffs
R6 -- Faculty_info
R7 -- Researcher_info
R81 -- College_info
R82 -- College_Branch
R9 -- Branches
R10 -- Teaches_In

➢ FINAL TABLES:

1) Alumni_info
(StudentID , StudentName , GraduationYear , Company)

2)Studied_from
(CollegeID , BranchID , StudentID)

3) Companies
(CompanyName ,Location)

4) Placements
(CollegeID , CompanyName , Percentage_Placed , No_of_offers_made , PackageOffered)

5)Cutoffs
(CollegeID , BranchID , Closing_Rank , Opening_Rank)

6)Faculty_info
(FacultyID , ResearcherID , FacultyName , Subject , Experience)

7)Researcher_info
(ResearcherID , Paper_Published , ResearchField)

8)College_info
(CollegeID ,CollegeName , Type , NIRF Ranking , Location , Established_In , Fee_Structure)

9) College_Branch
(CollegeID , BranchID)

10)Branches

(BranchID , BranchName , CourseDuration_Year)

11)Teaches_In

(CollegeID , FacultyID , ResearcherID)

****

➢ TABLES AND DATA

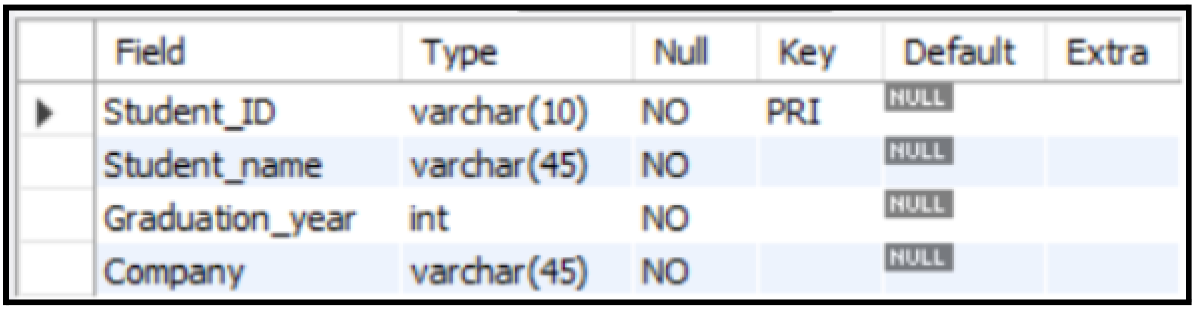
➔ Alumni_Info

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`alumni_info` (
  `Student_ID` VARCHAR(10) NOT NULL,
  `Student_name` VARCHAR(45) NOT NULL,
  `Graduation_year` INT NOT NULL,
  `Company` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`Student_ID`));
```

OUTPUT



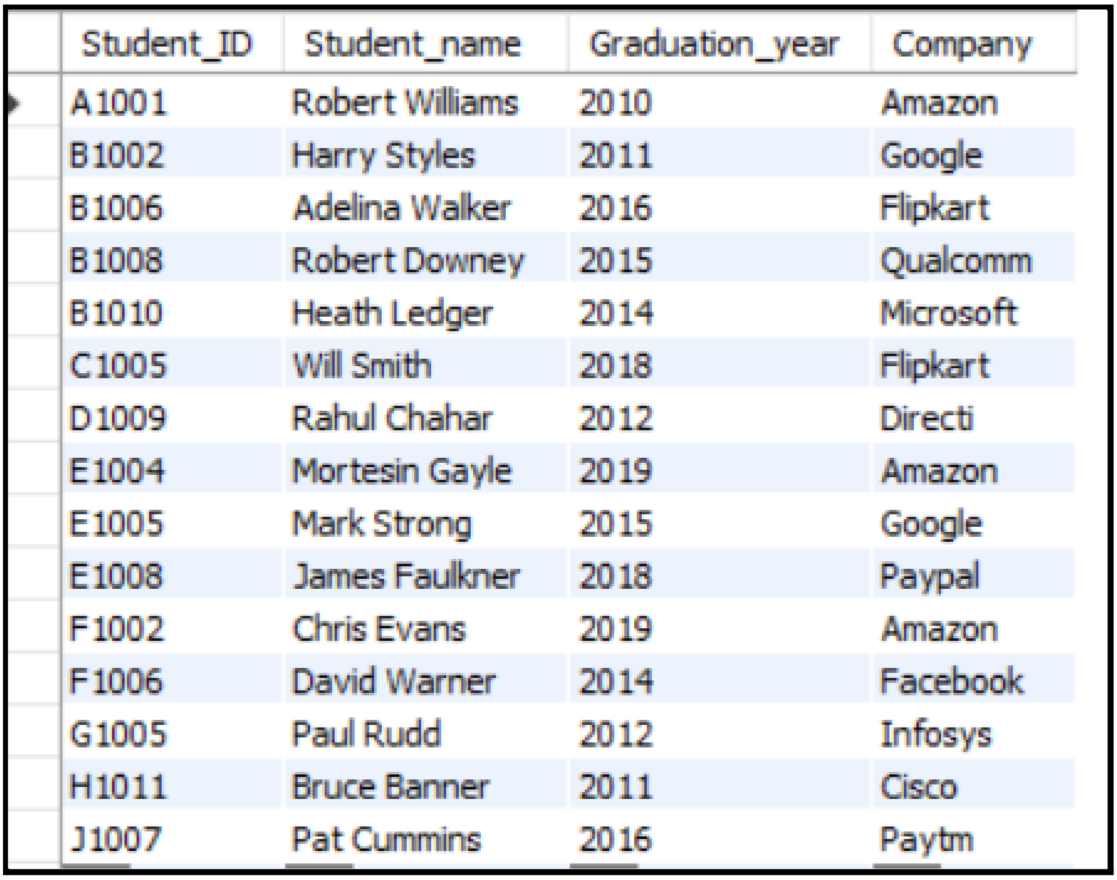
	Field	Type	Null	Key	Default	Extra
▶	Student_ID	varchar(10)	NO	PRI	NULL	
	Student_name	varchar(45)	NO		NULL	
	Graduation_year	int	NO		NULL	
	Company	varchar(45)	NO		NULL	

INSERTING DATA:

QUERY

```
insert into alumni_info values ( 'A1001', 'Robert Williams', 2010, 'Amazon');
insert into alumni_info values ( 'B1002', 'Harry Styles', 2011,'Google');
insert into alumni_info values ( 'B1010', 'Heath Ledger', 2014, 'Microsoft');
insert into alumni_info values ( 'B1006', 'Adelina Walker', 2016, 'Flipkart');
insert into alumni_info values ( 'B1008', 'Robert Downey', 2015, 'Qualcomm');
insert into alumni_info values ( 'C1005', 'Will Smith', 2018, 'Flipkart');
insert into alumni_info values ( 'D1009', 'Rahul Chahar', 2012, 'Directi');
insert into alumni_info values ( 'E1004', 'Mortesin Gayle', 2019, 'Amazon');
insert into alumni_info values ( 'E1005', 'Mark Strong', 2015, 'Google');
insert into alumni_info values ( 'E1008', 'James Faulkner', 2018, 'Paypal');
insert into alumni_info values ( 'F1006', 'David Warner', 2014, 'Facebook');
insert into alumni_info values  ( 'F1002', 'Chris Evans', 2019, 'Amazon');
insert into alumni_info values ( 'G1005', 'Paul Rudd', 2012, 'Infosys');
insert into alumni_info values ( 'H1011', 'Bruce Banner', 2011 ,'Cisco');
insert into alumni_info values ( 'J1007', 'Pat Cummins', 2016, 'Paytm');
select *from alumni_info;
```

OUTPUT



Student_ID	Student_name	Graduation_year	Company
A1001	Robert Williams	2010	Amazon
B1002	Harry Styles	2011	Google
B1006	Adelina Walker	2016	Flipkart
B1008	Robert Downey	2015	Qualcomm
B1010	Heath Ledger	2014	Microsoft
C1005	Will Smith	2018	Flipkart
D1009	Rahul Chahar	2012	Directi
E1004	Mortesin Gayle	2019	Amazon
E1005	Mark Strong	2015	Google
E1008	James Faulkner	2018	Paypal
F1002	Chris Evans	2019	Amazon
F1006	David Warner	2014	Facebook
G1005	Paul Rudd	2012	Infosys
H1011	Bruce Banner	2011	Cisco
J1007	Pat Cummins	2016	Paytm

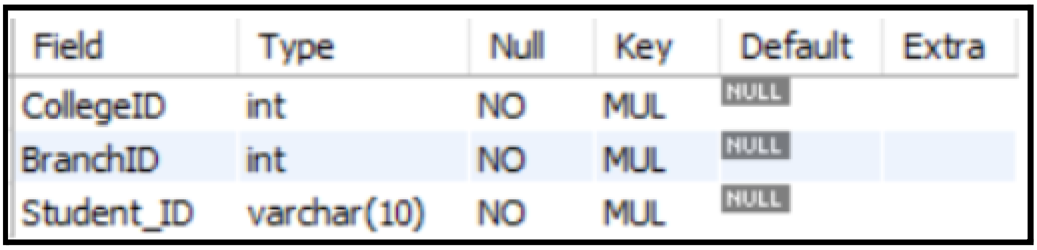
➔ Studied_From

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`studied_from` (
  `CollegeID` INT NOT NULL,
  `BranchID` INT NOT NULL,
  `Student_ID` VARCHAR(10) NOT NULL,
FOREIGN KEY (Student_ID) REFERENCES alumni_info(Student_ID),
FOREIGN KEY (BranchID) REFERENCES branches(BranchID),
FOREIGN KEY (CollegeID) REFERENCES college_info(CollegeID)
        );
```

OUTPUT



Field	Type	Null	Key	Default	Extra
CollegeID	int	NO	MUL	NULL	
BranchID	int	NO	MUL	NULL	
Student_ID	varchar(10)	NO	MUL	NULL	

INSERTING DATA:

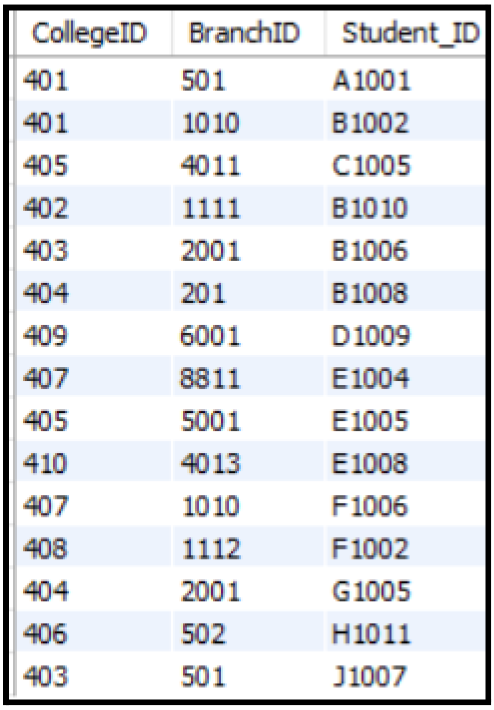
QUERY

```
insert into studied_from values ( 401, 501 , 'A1001');
insert into studied_from values ( 401, 1010, 'B1002');
insert into studied_from values ( 405, 4011, 'C1005');
insert into studied_from values (402, 1111, 'B1010');
insert into studied_from values (403, 2001, 'B1006');
insert into studied_from values (404, 201, 'B1008');
insert into studied_from values (409, 6001, 'D1009');
insert into studied_from values (407, 8811, 'E1004');
insert into studied_from values (405, 5001, 'E1005');
insert into studied_from values (410, 4013, 'E1008');
insert into studied_from values (407, 1010, 'F1006');
insert into studied_from values (408, 1112, 'F1002');
insert into studied_from values (404, 2001, 'G1005');
insert into studied_from values (406, 502, 'H1011');
```

insert into studied_from values (403, 501, 'J1007');

select *from studied_from;

OUTPUT



CollegeID	BranchID	Student_ID
401	501	A1001
401	1010	B1002
405	4011	C1005
402	1111	B1010
403	2001	B1006
404	201	B1008
409	6001	D1009
407	8811	E1004
405	5001	E1005
410	4013	E1008
407	1010	F1006
408	1112	F1002
404	2001	G1005
406	502	H1011
403	501	J1007

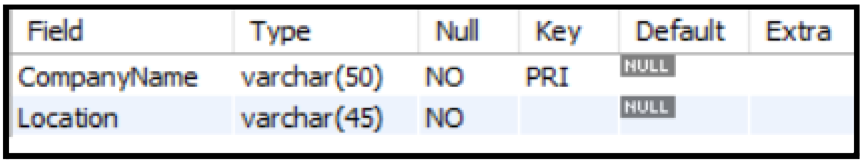
→ Companies

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`companies` (
  `CompanyName` VARCHAR(50) NOT NULL,
  `Location` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`CompanyName`));
```

OUTPUT



Field	Type	Null	Key	Default	Extra
CompanyName	varchar(50)	NO	PRI	NULL	
Location	varchar(45)	NO		NULL	

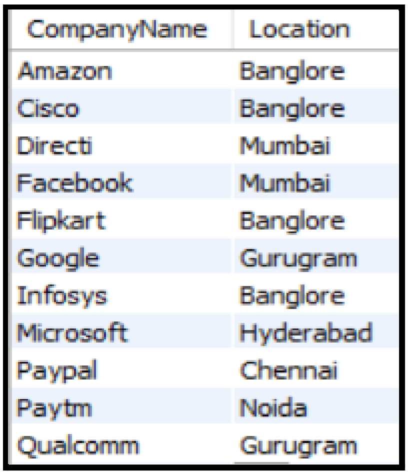
INSERTING DATA:

QUERY

```
insert into companies values ( 'Amazon', 'Banglore');
insert into companies values ( 'Cisco', 'Banglore');
insert into companies values ( 'Directi', 'Mumbai');
insert into companies values ( 'Facebook', 'Mumbai');
insert into companies values ( 'Flipkart', 'Banglore');
insert into companies values ( 'Google', 'Gurugram');
insert into companies values ( 'Infosys', 'Banglore');
insert into companies values ( 'Microsoft', 'Hyderabad');
insert into companies values ( 'Paytm', 'Noida');
insert into companies values ( 'Paypal', 'Chennai');
insert into companies values ( 'Qualcomm', 'Gurugram');
```

select *from companies;

OUTPUT



CompanyName	Location
Amazon	Banglore
Cisco	Banglore
Directi	Mumbai
Facebook	Mumbai
Flipkart	Banglore
Google	Gurugram
Infosys	Banglore
Microsoft	Hyderabad
Paypal	Chennai
Paytm	Noida
Qualcomm	Gurugram

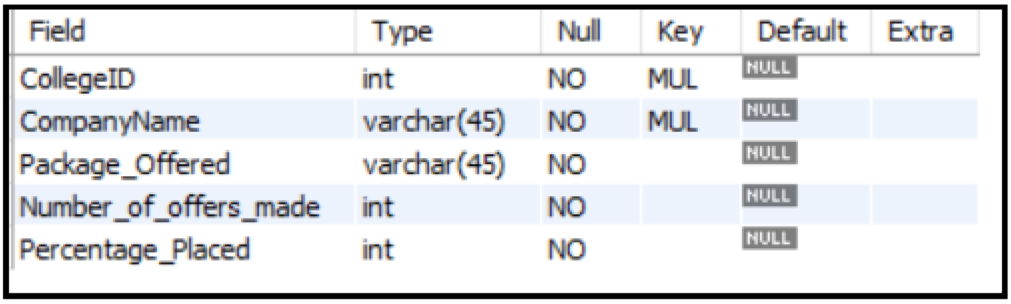
→ Placements

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`placements` (
  `CollegeID` INT NOT NULL,
  `CompanyName` VARCHAR(45) NOT NULL,
  `Package_Offered` VARCHAR(45) NOT NULL,
  `Number_of_offers_made` INT NOT NULL,
  `Percentage_Placed` INT NOT NULL,
FOREIGN KEY (CollegeID) REFERENCES college_info(CollegeID));
```

OUTPUT



Field	Type	Null	Key	Default	Extra
CollegeID	int	NO	MUL	NULL	
CompanyName	varchar(45)	NO	MUL	NULL	
Package_Offered	varchar(45)	NO		NULL	
Number_of_offers_made	int	NO		NULL	
Percentage_Placed	int	NO		NULL	

INSERTING DATA:

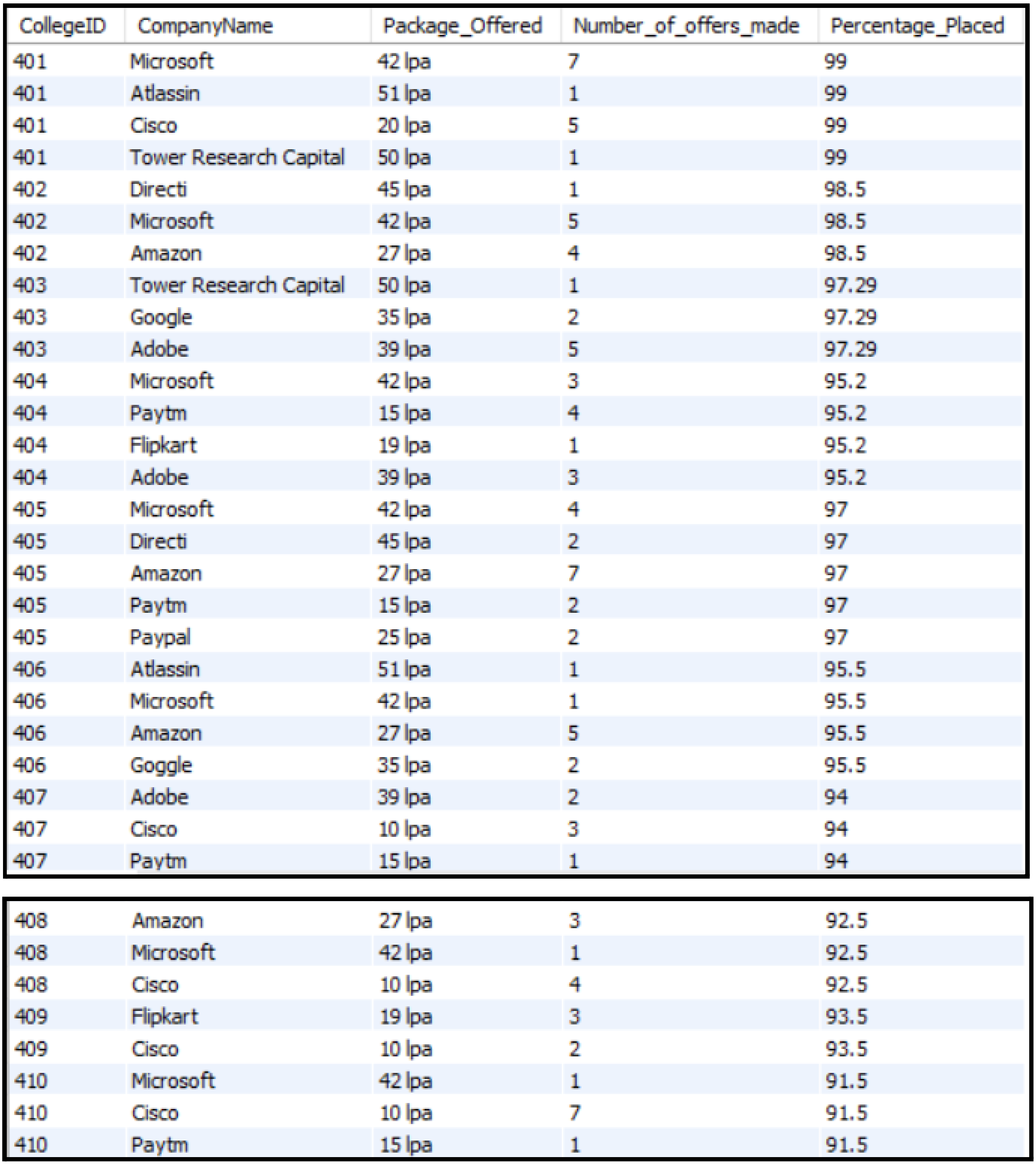
QUERY

```
        insert into placements values ( 401, 'Microsoft', '42 lpa', 7, 99);
        insert into placements values ( 401, 'Atlassin', '51 lpa', 1, 99);
        insert into placements values (401, 'Cisco', '20 lpa', 5, 99);
        insert into placements values (401, 'Tower Research Capital', '50 lpa', 1,99);
        insert into placements values (402,'Directi', '45 lpa', 1, 98.5);
        insert into placements values (402, 'Microsoft', '42 lpa', 5, 98.5);
        insert into placements values (402, 'Amazon', '27 lpa', 4, 98.5);
        insert into placements values (403, 'Tower Research Capital', '50 lpa', 1, 97.29);
        insert into placements values (403, 'Google', '35 lpa', 2, 97.29);
        insert into placements values (403, 'Adobe', '39 lpa', 5, 97.29);
        insert into placements values (404, 'Microsoft', '42 lpa', 3, 95);
        insert into placements values (404, 'Paytm', '15 lpa', 4, 95);
        insert into placements values (404, 'Flipkart', '19 lpa', 1, 95);
        insert into placements values (404, 'Adobe', '39 lpa', 3, 95);
        insert into placements values (405, 'Microsoft', '42 lpa', 4, 97);
        insert into placements values (405, 'Directi', '45 lpa', 2, 97);
        insert into placements values (405, 'Amazon', '27 lpa', 7, 97);
```

```
insert into placements values (405, 'Paytm', '15 lpa', 2, 97);
insert into placements values (405, 'Paypal', '25 lpa', 2, 97);
insert into placements values (406, 'Atlassin', '51 lpa', 1 , 95.5);
insert into placements values (406, 'Microsoft', '42 lpa', 1, 95.5);
insert into placements values (406, 'Amazon', '27 lpa', 5, 95.5);
insert into placements values (406, 'Goggle', '35 lpa', 2, 95.5);
insert into placements values (407, 'Adobe', '39 lpa', 2, 94);
insert into placements values (407, 'Cisco', '10 lpa', 3, 94);
insert into placements values (407, 'Paytm', '15 lpa', 1, 94);
insert into placements values (408, 'Amazon', '27 lpa', 3, 92.5);
insert into placements values (408, 'Microsoft', '42 lpa', 1, 92.5);
insert into placements values (408, 'Cisco', '10 lpa', 4, 92.5);
insert into placements values (409, 'Flipkart', '19 lpa', 3, 93.5);
insert into placements values (409, 'Cisco', '10 lpa', 2, 93.5);
insert into placements values (410, 'Microsoft', '42 lpa', 1, 91.5);
insert into placements values (410, 'Cisco', '10 lpa', 7, 91.5);
insert into placements values (410, 'Paytm', '15 lpa', 1, 91.5);
```

select *from placements;

OUTPUT



CollegeID	CompanyName	Package_Offered	Number_of_offers_made	Percentage_Placed
401	Microsoft	42 lpa	7	99
401	Atlassin	51 lpa	1	99
401	Cisco	20 lpa	5	99
401	Tower Research Capital	50 lpa	1	99
402	Directi	45 lpa	1	98.5
402	Microsoft	42 lpa	5	98.5
402	Amazon	27 lpa	4	98.5
403	Tower Research Capital	50 lpa	1	97.29
403	Google	35 lpa	2	97.29
403	Adobe	39 lpa	5	97.29
404	Microsoft	42 lpa	3	95.2
404	Paytm	15 lpa	4	95.2
404	Flipkart	19 lpa	1	95.2
404	Adobe	39 lpa	3	95.2
405	Microsoft	42 lpa	4	97
405	Directi	45 lpa	2	97
405	Amazon	27 lpa	7	97
405	Paytm	15 lpa	2	97
405	Paypal	25 lpa	2	97
406	Atlassin	51 lpa	1	95.5
406	Microsoft	42 lpa	1	95.5
406	Amazon	27 lpa	5	95.5
406	Goggle	35 lpa	2	95.5
407	Adobe	39 lpa	2	94
407	Cisco	10 lpa	3	94
407	Paytm	15 lpa	1	94
408	Amazon	27 lpa	3	92.5
408	Microsoft	42 lpa	1	92.5
408	Cisco	10 lpa	4	92.5
409	Flipkart	19 lpa	3	93.5
409	Cisco	10 lpa	2	93.5
410	Microsoft	42 lpa	1	91.5
410	Cisco	10 lpa	7	91.5
410	Paytm	15 lpa	1	91.5

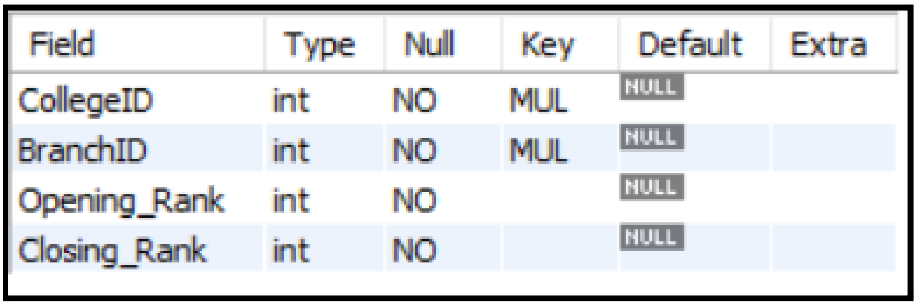
→ Cutoffs

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`cutoff` (
  `CollegeID` INT NOT NULL,
  `BranchID` INT NOT NULL,
  `Opening_Rank` INT NOT NULL,
  `Closing_Rank` INT NOT NULL,
FOREIGN KEY (CollegeID) REFERENCES college_info(CollegeID),
FOREIGN KEY (BranchID) REFERENCES branches(BranchID));
```

OUTPUT



Field	Type	Null	Key	Default	Extra
CollegeID	int	NO	MUL	NULL	
BranchID	int	NO	MUL	NULL	
Opening_Rank	int	NO		NULL	
Closing_Rank	int	NO		NULL	

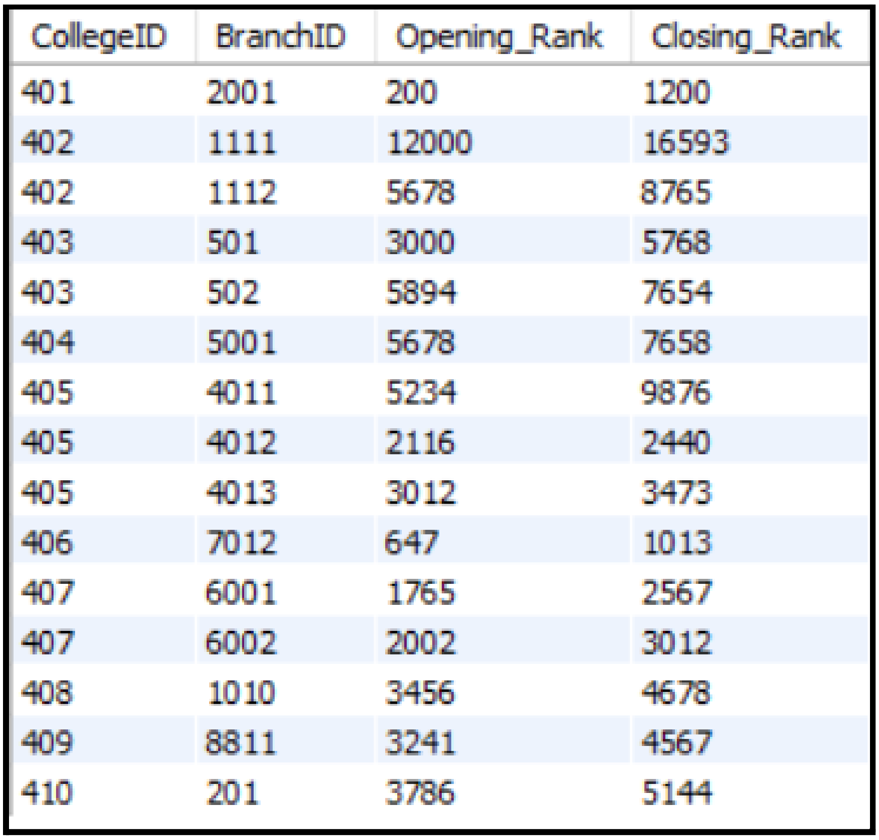
INSERTING DATA:

QUERY

```
insert into cutoff values (401, 2001, 200, 1200);
insert into cutoff values (402, 1111, 12000, 16593);
insert into cutoff values (402, 1112, 5678, 8765);
insert into cutoff values (403, 501, 3000, 5768);
insert into cutoff values (403, 502, 5894, 7654);
insert into cutoff values (404, 5001, 5678, 7658);
insert into cutoff values (405, 4011, 5234, 9876);
insert into cutoff values (405, 4012, 2116, 2440);
insert into cutoff values (405,4013, 3012, 3473);
insert into cutoff values (406, 7012, 647, 1013);
insert into cutoff values (407, 6001, 1765, 2567);
insert into cutoff values (407, 6002, 2002, 3012);
insert into cutoff values (408, 1010, 3456, 4678);
insert into cutoff values (409, 8811, 3241, 4567);
insert into cutoff values (410, 201, 3786, 5144);
```

select *from cutoff;

OUTPUT



CollegeID	BranchID	Opening_Rank	Closing_Rank
401	2001	200	1200
402	1111	12000	16593
402	1112	5678	8765
403	501	3000	5768
403	502	5894	7654
404	5001	5678	7658
405	4011	5234	9876
405	4012	2116	2440
405	4013	3012	3473
406	7012	647	1013
407	6001	1765	2567
407	6002	2002	3012
408	1010	3456	4678
409	8811	3241	4567
410	201	3786	5144

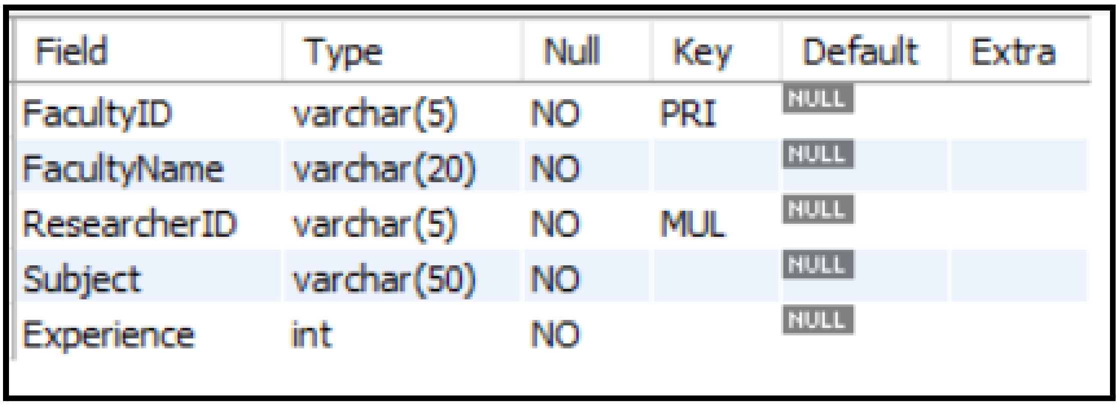
→ Faculty_Info

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`faculty_info` (
  `FacultyID` VARCHAR(5) NOT NULL,
  `FacultyName` VARCHAR(20) NOT NULL,
  `ResearcherID` VARCHAR(5) NOT NULL,
  `Subject` VARCHAR(50) NOT NULL,
  `Experience` INT NOT NULL,
FOREIGN KEY (`ResearcherID`) REFERENCES researcher_info(`ResearcherID`),
PRIMARY KEY (`FacultyID`));
```

OUTPUT



Field	Type	Null	Key	Default	Extra
FacultyID	varchar(5)	NO	PRI	NULL	
FacultyName	varchar(20)	NO		NULL	
ResearcherID	varchar(5)	NO	MUL	NULL	
Subject	varchar(50)	NO		NULL	
Experience	int	NO		NULL	

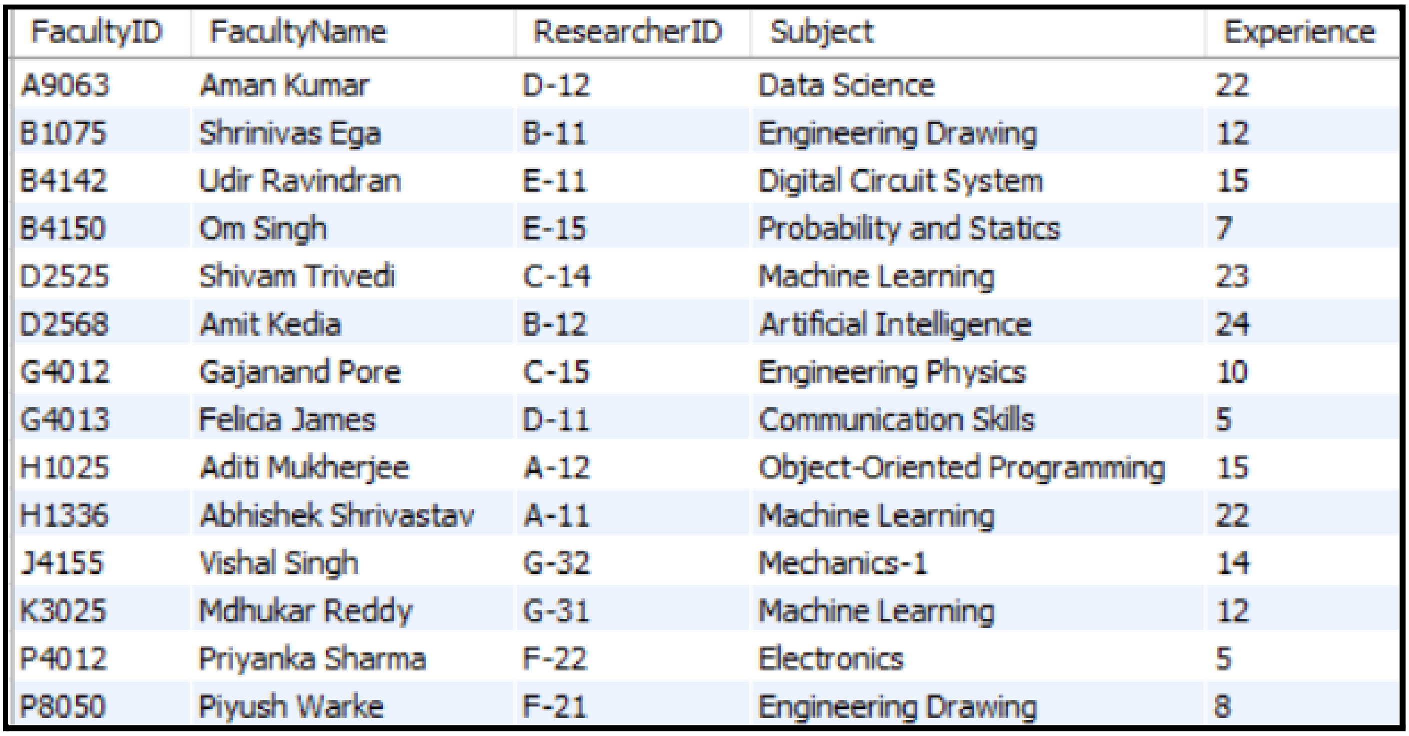
INSERTING DATA:

QUERY

```
insert into faculty_info values
('H1336’, 'Abhishek Shrivastav', 'A-11', 'Machine Learning', 22);
insert into faculty_info values
(‘H1025’, 'Aditi Mukherjee', 'A-12', 'Object-Oriented Programming', 15);
insert into faculty_info values
(‘B1075’, 'Shrinivas Ega', 'B-11', 'Engineering Drawing', 12);
insert into faculty_info values
(‘D2568’, 'Amit Kedia', 'B-12', 'Artificial Intelligence', 24);
insert into faculty_info values
(‘D2525’, 'Shivam Trivedi', 'C-14', 'Machine Learning', 23);
insert into faculty_info values
```

```
(‘G4012’,'Gajanand Pore', 'C-15', 'Engineering Physics', 10);
insert into faculty_info values
(‘G4013’, 'Felicia James', 'D-11', 'Communication Skills', 5);
insert into faculty_info values
(‘A9063’, 'Aman Kumar', 'D-12', 'Data Science', 22);
insert into faculty_info values
(‘B4142’, 'Udir Ravindran', 'E-11', 'Digital Circuit System', 15);
insert into faculty_info values
(‘B4150’, 'Om Singh', 'E-15', 'Probability and Statics', 7);
insert into faculty_info values
(‘P8050’, 'Piyush Warke', 'F-21', 'Engineering Drawing', 8);
insert into faculty_info values
(‘P4012’, 'Priyanka Sharma', 'F-22', 'Electronics', 5);
insert into faculty_info values
(‘K3025’, 'Mdhukar Reddy', 'G-31', 'Machine Learning',12);
insert into faculty_info values
(‘J4155’, 'Vishal Singh', 'G-32', 'Mechanics-1', 14);
```

OUTPUT



FacultyID	FacultyName	ResearcherID	Subject	Experience
A9063	Aman Kumar	D-12	Data Science	22
B1075	Shrinivas Ega	B-11	Engineering Drawing	12
B4142	Udir Ravindran	E-11	Digital Circuit System	15
B4150	Om Singh	E-15	Probability and Statics	7
D2525	Shivam Trivedi	C-14	Machine Learning	23
D2568	Amit Kedia	B-12	Artificial Intelligence	24
G4012	Gajanand Pore	C-15	Engineering Physics	10
G4013	Felicia James	D-11	Communication Skills	5
H1025	Aditi Mukherjee	A-12	Object-Oriented Programming	15
H1336	Abhishek Shrivastav	A-11	Machine Learning	22
J4155	Vishal Singh	G-32	Mechanics-1	14
K3025	Mdhukar Reddy	G-31	Machine Learning	12
P4012	Priyanka Sharma	F-22	Electronics	5
P8050	Piyush Warke	F-21	Engineering Drawing	8

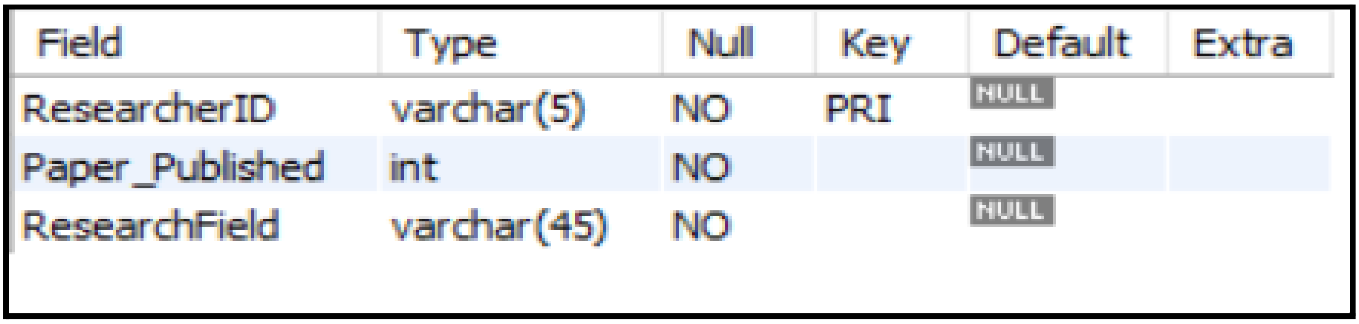
➔ Researcher_Info

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`researcher_info` (
  `ResearcherID` VARCHAR(5) NOT NULL,
  `Paper_Published` INT NOT NULL,
  `ResearchField` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`ResearcherID`));
```

OUTPUT



Field	Type	Null	Key	Default	Extra
ResearcherID	varchar(5)	NO	PRI	NULL	
Paper_Published	int	NO		NULL	
ResearchField	varchar(45)	NO		NULL	

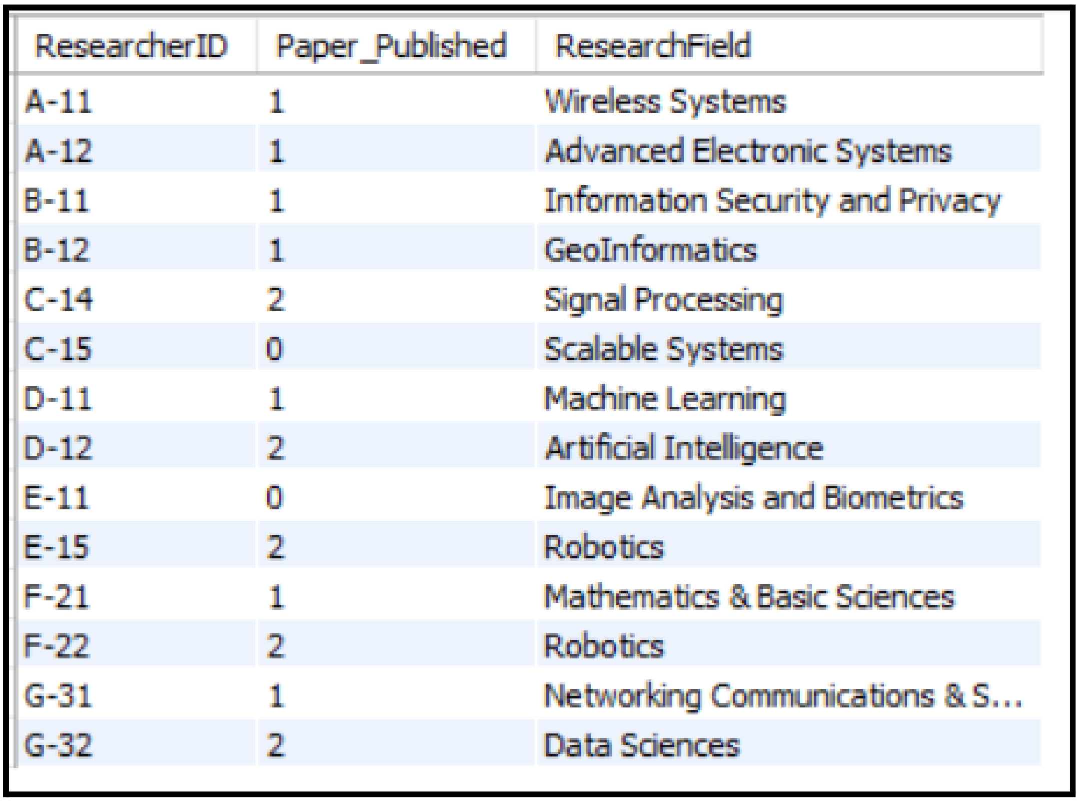
INSERTING DATA:

QUERY

```
insert into researcher_info values
('A-11', 1, 'Wireless Systems');
insert into researcher_info values
('A-12', 1, 'Advanced Electronic Systems');
insert into researcher_info values
('B-11', 1, 'Information Security and Privacy');
insert into researcher_info values
('B-12', 1, 'GeoInformatics');
insert into researcher_info values
('C-14', 2, 'Signal Processing');
insert into researcher_info values
('C-15', 0, 'Scalable Systems');
insert into researcher_info values
('D-11', 1, 'Machine Learning');
```

```
insert into researcher_info values
('D-12', 2, 'Artificial Intelligence');
insert into researcher_info values
('E-11', 0, 'Image Analysis and Biometrics');
insert into researcher_info values
('E-15', 2, 'Robotics');
insert into researcher_info values
('F-21', 1, 'Mathematics & Basic Sciences');
insert into researcher_info values
('F-22', 2, 'Robotics');
insert into researcher_info values
('G-31', 1, 'Networking Communications & Signal Processing');
insert into researcher_info values
('G-32', 2, 'Data Sciences');
```

OUTPUT



ResearcherID	Paper_Published	ResearchField
A-11	1	Wireless Systems
A-12	1	Advanced Electronic Systems
B-11	1	Information Security and Privacy
B-12	1	GeoInformatics
C-14	2	Signal Processing
C-15	0	Scalable Systems
D-11	1	Machine Learning
D-12	2	Artificial Intelligence
E-11	0	Image Analysis and Biometrics
E-15	2	Robotics
F-21	1	Mathematics & Basic Sciences
F-22	2	Robotics
G-31	1	Networking Communications & S...
G-32	2	Data Sciences

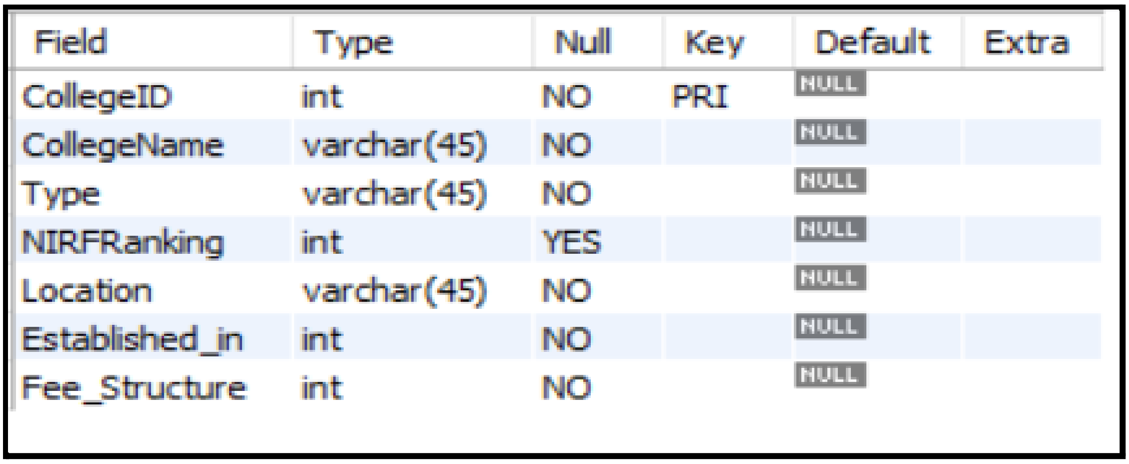
→ College_Info

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`college_info` (
  `CollegeID` INT NOT NULL,
  `CollegeName` VARCHAR(45) NOT NULL,
  `Type` VARCHAR(45) NOT NULL,
  `NIRFRanking` INT NULL,
  `Location` VARCHAR(45) NOT NULL,
  `Established_in` INT NOT NULL,
  `Fee_Structure` INT NOT NULL,
PRIMARY KEY (`CollegeID`));
```

OUTPUT



Field	Type	Null	Key	Default	Extra
CollegeID	int	NO	PRI	NULL	
CollegeName	varchar(45)	NO		NULL	
Type	varchar(45)	NO		NULL	
NIRFRanking	int	YES		NULL	
Location	varchar(45)	NO		NULL	
Established_in	int	NO		NULL	
Fee_Structure	int	NO		NULL	

INSERTING DATA:

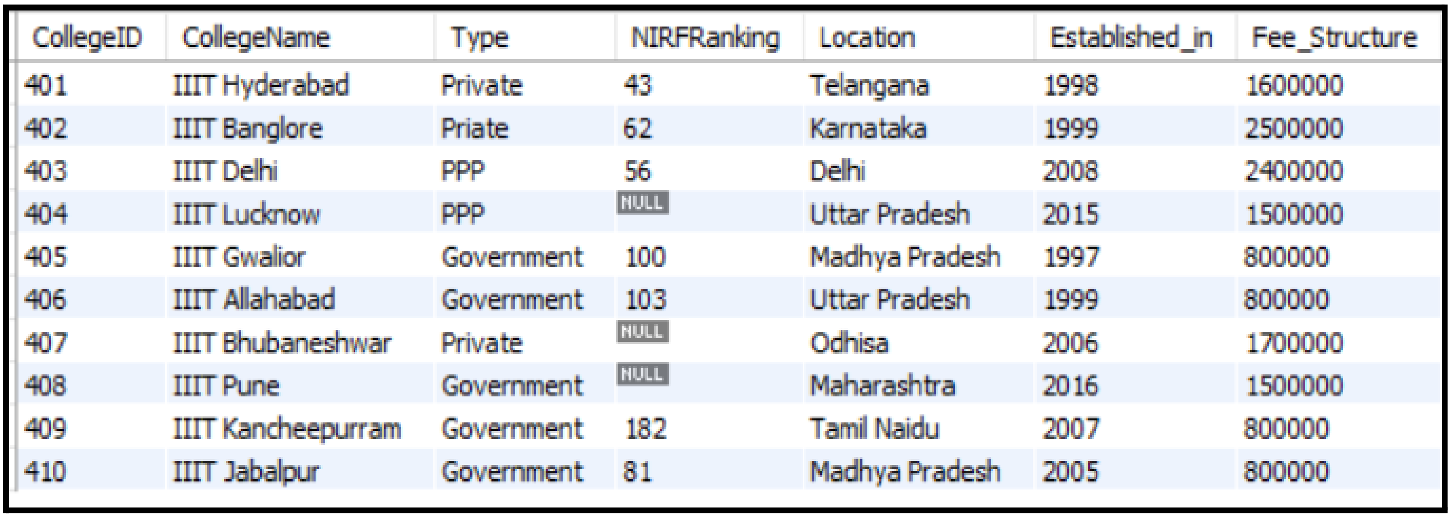
QUERY

```
insert into college_info values
(401, 'IIIT Hyderabad', 'Private', 43, 'Telangana', 1998, 1600000);
insert into college_info values
(402, 'IIIT Banglore',  'Priate', 62, 'Karnataka', 1999, 2500000);
insert into college_info values
(403, 'IIIT Delhi', 'PPP', 56, 'Delhi', 2008, 2400000);
insert into college_info values
(404, 'IIIT Lucknow', 'PPP', null, 'Uttar Pradesh', 2015, 1500000);
insert into college_info values
(405, 'IIIT Gwalior', 'Government', 100, 'Madhya Pradesh', 1997, 800000);
```

```
insert into college_info values
(406, 'IIIT Allahabad', 'Government', 103, 'Uttar Pradesh', 1999, 800000);
insert into college_info values
(407, 'IIIT Bhubaneshwar', 'Private', null, 'Odhisa', 2006, 1700000);
insert into college_info values
(408, 'IIIT Pune', 'Government', null, 'Maharashtra', 2016, 1500000);
insert into college_info values
(409, 'IIIT Kancheepurram', 'Government', 182, 'Tamil Naidu', 2007, 800000);
insert into college_info values
(410, 'IIIT Jabalpur', 'Government', 81, 'Madhya Pradesh', 2005, 800000);
```

select *from college_info;

OUTPUT



CollegeID	CollegeName	Type	NIRFRanking	Location	Established_in	Fee_Structure
401	IIIT Hyderabad	Private	43	Telangana	1998	1600000
402	IIIT Banglore	Priate	62	Karnataka	1999	2500000
403	IIIT Delhi	PPP	56	Delhi	2008	2400000
404	IIIT Lucknow	PPP	NULL	Uttar Pradesh	2015	1500000
405	IIIT Gwalior	Government	100	Madhya Pradesh	1997	800000
406	IIIT Allahabad	Government	103	Uttar Pradesh	1999	800000
407	IIIT Bhubaneshwar	Private	NULL	Odhisa	2006	1700000
408	IIIT Pune	Government	NULL	Maharashtra	2016	1500000
409	IIIT Kancheepurram	Government	182	Tamil Naidu	2007	800000
410	IIIT Jabalpur	Government	81	Madhya Pradesh	2005	800000

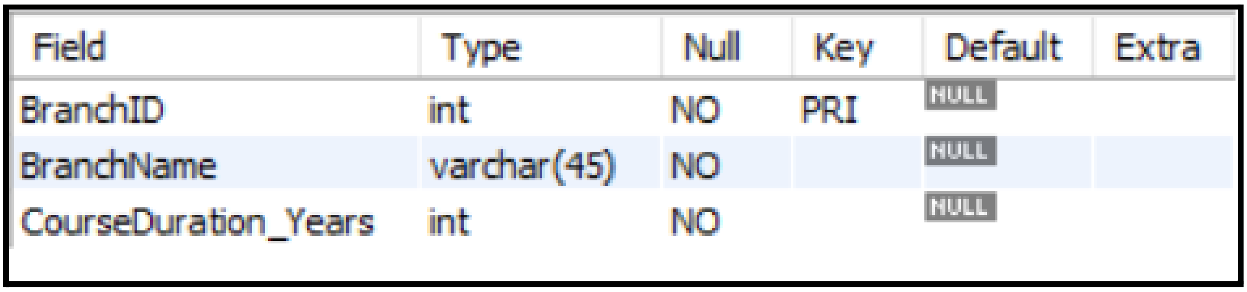
→ Branches

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`branches` (
  `BranchID` INT NOT NULL,
  `BranchName` VARCHAR(45) NOT NULL,
  `CourseDuration_Years` INT NULL,
PRIMARY KEY (`BranchID`));
```

OUTPUT



Field	Type	Null	Key	Default	Extra
BranchID	int	NO	PRI	NULL	
BranchName	varchar(45)	NO		NULL	
CourseDuration_Years	int	NO		NULL	

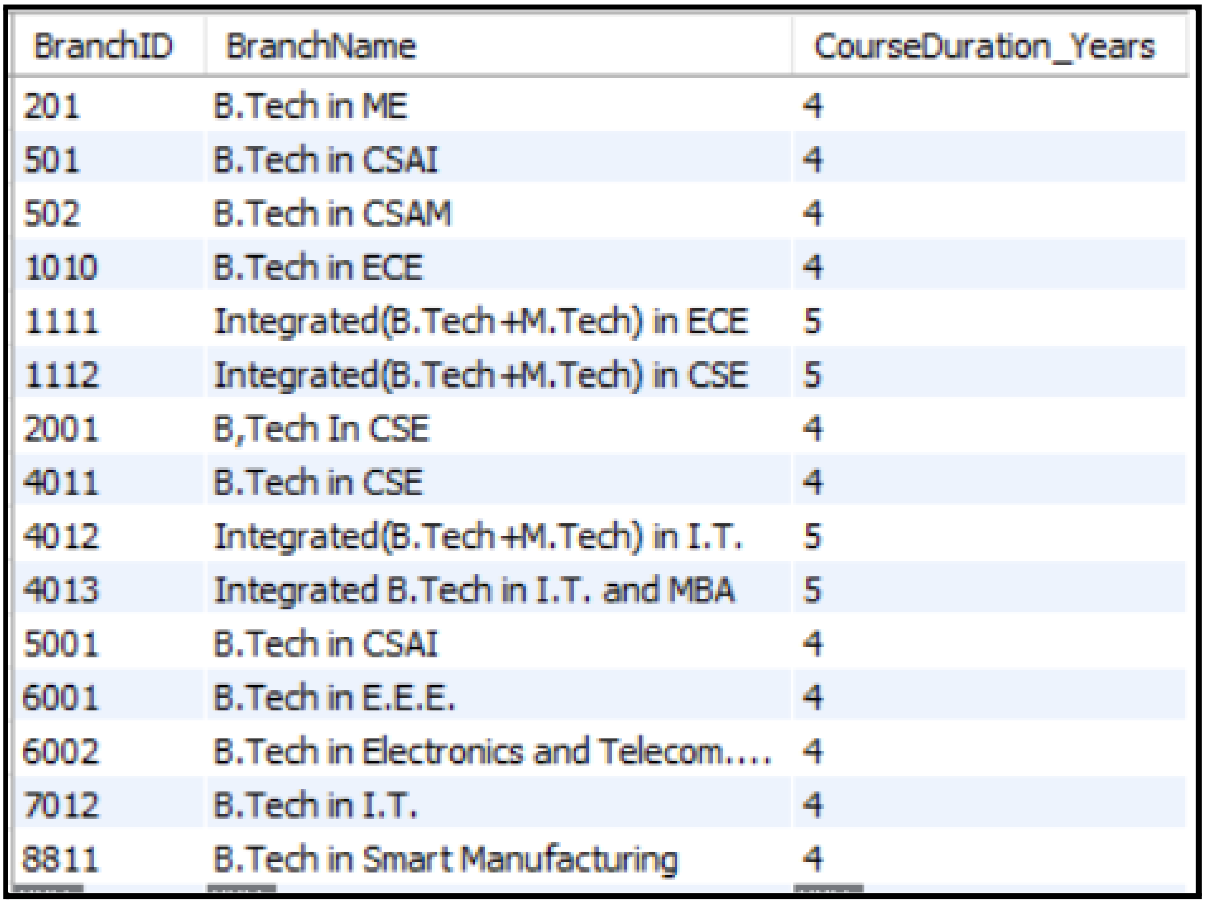
INSERTING DATA:

QUERY

```
insert into branches values (2001, 'B,Tech In CSE', 4);
insert into branches values (1111, 'Integrated(B.Tech+M.Tech) in ECE', 5);
insert into branches values (1112, 'Integrated(B.Tech+M.Tech) in CSE', 5);
insert into branches values (501, 'B.Tech in CSAI', 4);
insert into branches values (502,'B.Tech in CSAM', 4);
insert into branches values (5001, 'B.Tech in CSAI', 4);
insert into branches values (4011, 'B.Tech in CSE', 4);
insert into branches values (4012, 'Integrated(B.Tech+M.Tech) in I.T.', 5);
insert into branches values (4013, 'Integrated B.Tech in I.T. and MBA', 5);
insert into branches values (7012, 'B.Tech in I.T.', 4);
insert into branches values (6001, 'B.Tech in E.E.E.', 4);
insert into branches values (6002, 'B.Tech in Electronics and Telecom. Eng.', 4);
insert into branches values (1010, 'B.Tech in ECE', 4);
insert into branches values (8811, 'B.Tech in Smart Manufacturing', 4);
insert into branches values (201, 'B.Tech in ME', 4);
```

select * from branches

OUTPUT



BranchID	BranchName	CourseDuration_Years
201	B.Tech in ME	4
501	B.Tech in CSAI	4
502	B.Tech in CSAM	4
1010	B.Tech in ECE	4
1111	Integrated(B.Tech+M.Tech) in ECE	5
1112	Integrated(B.Tech+M.Tech) in CSE	5
2001	B,Tech In CSE	4
4011	B.Tech in CSE	4
4012	Integrated(B.Tech+M.Tech) in I.T.	5
4013	Integrated B.Tech in I.T. and MBA	5
5001	B.Tech in CSAI	4
6001	B.Tech in E.E.E.	4
6002	B.Tech in Electronics and Telecom....	4
7012	B.Tech in I.T.	4
8811	B.Tech in Smart Manufacturing	4

→ Teaches_In

CREATING TABLE:

QUERY

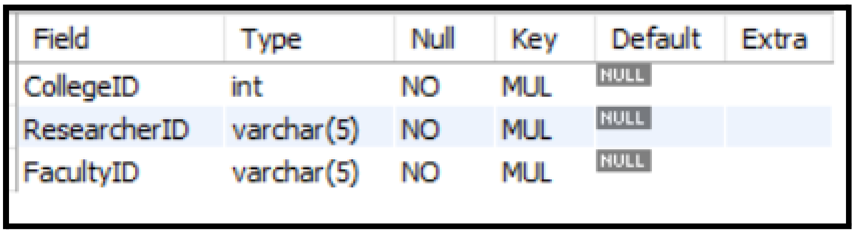
```
CREATE TABLE `dbms`.`teaches_in` (
  `CollegeID` INT NOT NULL,
  `ResearcherID` VARCHAR(5) NOT NULL,
  `FacultyID` VARCHAR(5) NOT NULL);

ALTER TABLE teaches_in
ADD FOREIGN KEY (CollegeID)
REFERENCES
college_info(CollegeID);

ALTER TABLE teaches_in
ADD FOREIGN KEY (ResearcherID)
REFERENCES
researcher_info(ResearcherID);

ALTER TABLE teaches_in
ADD FOREIGN KEY (FacultyID)
REFERENCES
faculty_info(FacultyID);
```

OUTPUT



Field	Type	Null	Key	Default	Extra
CollegeID	int	NO	MUL	NULL	
ResearcherID	varchar(5)	NO	MUL	NULL	
FacultyID	varchar(5)	NO	MUL	NULL	

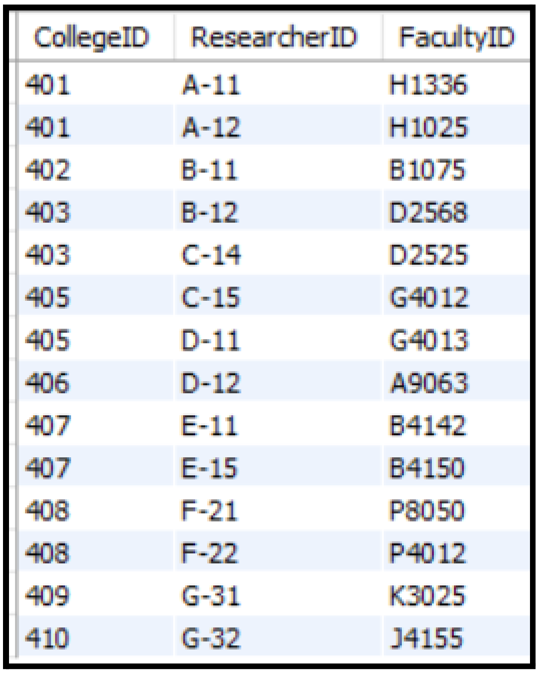
INSERTING DATA:

QUERY

```
insert into teaches_in values (401, 'A-11' ,'H1336');
insert into teaches_in values (401, 'A-12', 'H1025');
insert into teaches_in values (402, 'B-11', 'B1075');
insert into teaches_in values (403, 'B-12', 'D2568');
insert into teaches_in values (403, 'C-14', 'D2525');
insert into teaches_in values (405, 'C-15', 'G4012');
insert into teaches_in values (405, 'D-11', 'G4013');
insert into teaches_in values (406, 'D-12', 'A9063');
insert into teaches_in values (407, 'E-11', 'B4142');
insert into teaches_in values (407, 'E-15', 'B4150');
insert into teaches_in values (408, 'F-21', 'P8050');
insert into teaches_in values (408, 'F-22', 'P4012');
insert into teaches_in values (409, 'G-31', 'K3025');
insert into teaches_in values (410, 'G-32', 'J4155');
```

select *from teaches_in;

OUTPUT



CollegeID	ResearcherID	FacultyID
401	A-11	H1336
401	A-12	H1025
402	B-11	B1075
403	B-12	D2568
403	C-14	D2525
405	C-15	G4012
405	D-11	G4013
406	D-12	A9063
407	E-11	B4142
407	E-15	B4150
408	F-21	P8050
408	F-22	P4012
409	G-31	K3025
410	G-32	J4155

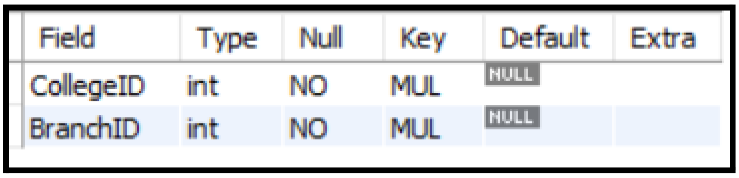
→ Colleges_branch

CREATING TABLE:

QUERY

```
CREATE TABLE `dbms`.`college_branch` (
  `CollegeID` INT NOT NULL,
  `BranchID` INT NOT NULL,
FOREIGN KEY (CollegeID) REFERENCES college_info(CollegeID),
FOREIGN KEY (BranchID) REFERENCES branches(BranchID));
```

OUTPUT



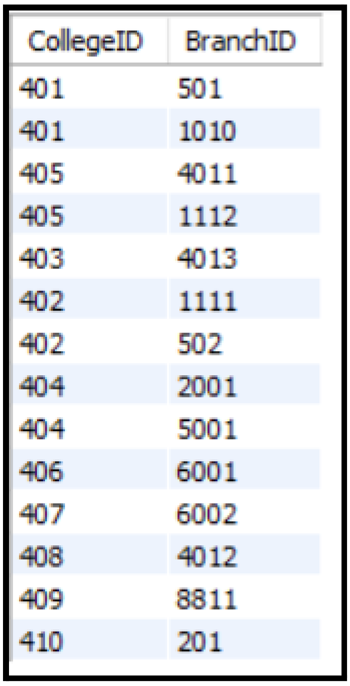
Field	Type	Null	Key	Default	Extra
CollegeID	int	NO	MUL	NULL	
BranchID	int	NO	MUL	NULL	

INSERTING DATA:

QUERY

```
insert into college_branch values (401,501);
insert into college_branch values (401,1010);
insert into college_branch values (405,4011);
insert into college_branch values (405,1112);
insert into college_branch values (403,4013);
insert into college_branch values (402,1111);
insert into college_branch values (402,502);
insert into college_branch values (404,2001);
insert into college_branch values (404,5001);
insert into college_branch values (406,6001);
insert into college_branch values (407,6002);
insert into college_branch values (408,4012);
insert into college_branch values (409,8811);
insert into college_branch values (410,201);
```

select *from college_branch;



CollegeID	BranchID
401	501
401	1010
405	4011
405	1112
403	4013
402	1111
402	502
404	2001
404	5001
406	6001
407	6002
408	4012
409	8811
410	201

➢QUERIES

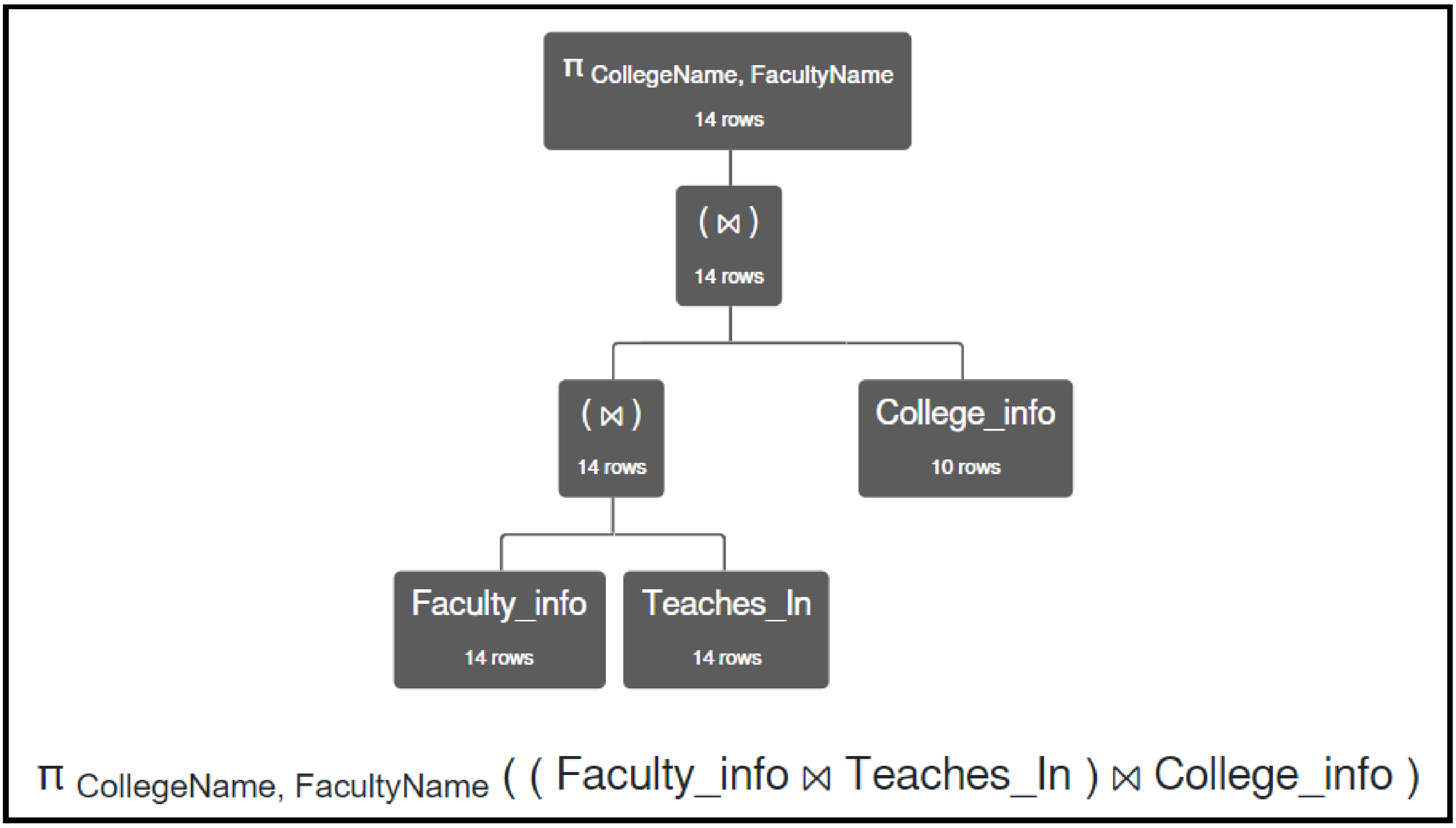
→ RELATIONAL ALGEBRA

1)Display the names of faculties and the college where they teach.

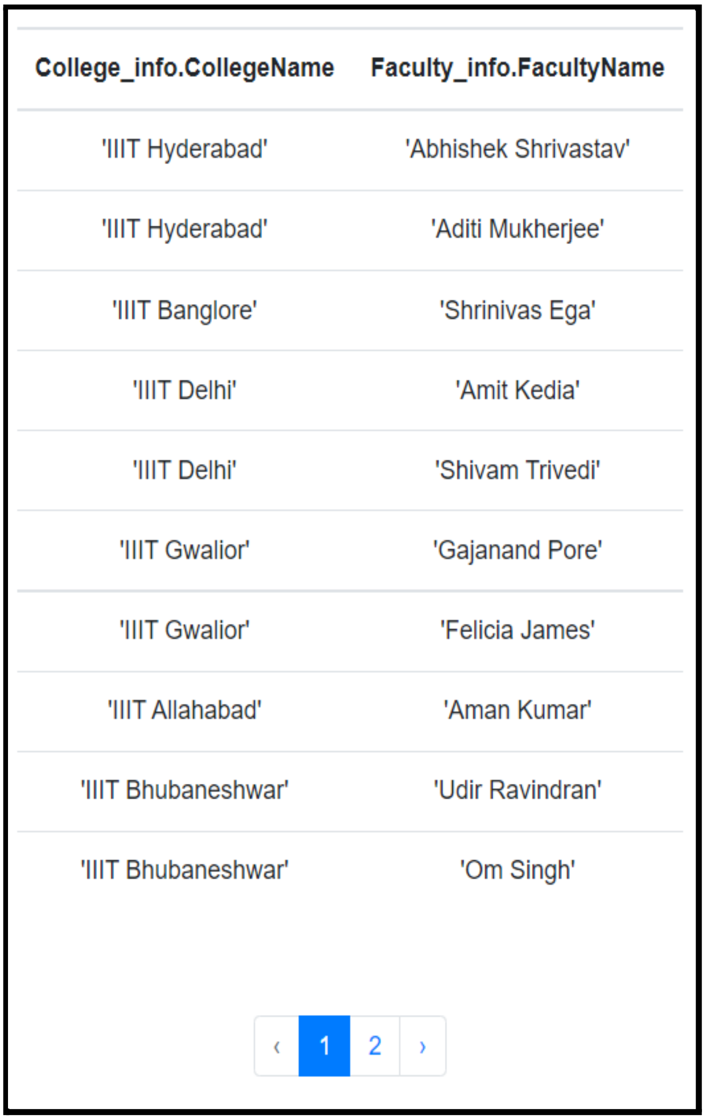
INPUT

π CollegeName , FacultyName ((Faculty_info ⋈ Teaches_in) ⋈ College_info)

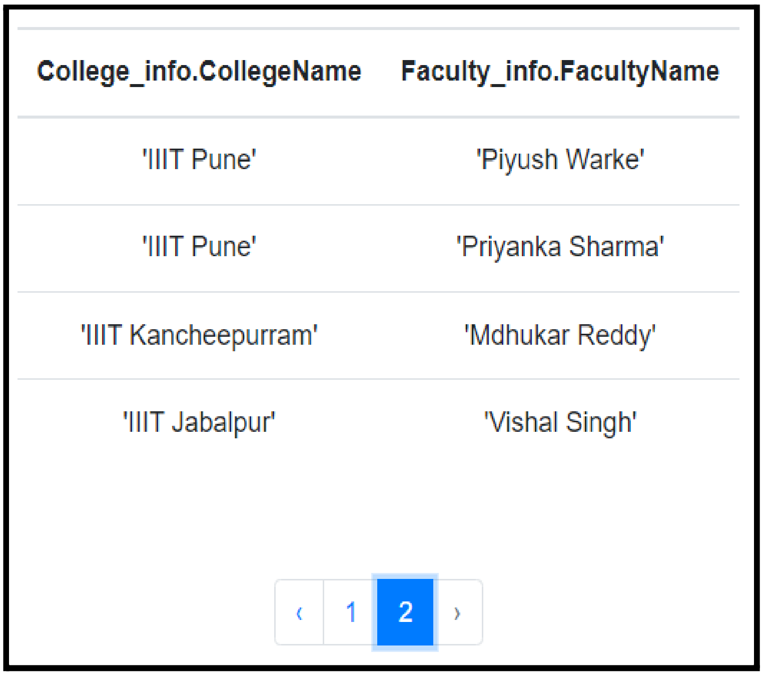
OUTPUT



π CollegeName, FacultyName ((Faculty_info ⋈ Teaches_In) ⋈ College_info)



College_info.CollegeName	Faculty_info.FacultyName
'IIIT Hyderabad'	'Abhishek Shrivastav'
'IIIT Hyderabad'	'Aditi Mukherjee'
'IIIT Banglore'	'Shrinivas Ega'
'IIIT Delhi'	'Amit Kedia'
'IIIT Delhi'	'Shivam Trivedi'
'IIIT Gwalior'	'Gajanand Pore'
'IIIT Gwalior'	'Felicia James'
'IIIT Allahabad'	'Aman Kumar'
'IIIT Bhubaneshwar'	'Udir Ravindran'
'IIIT Bhubaneshwar'	'Om Singh'



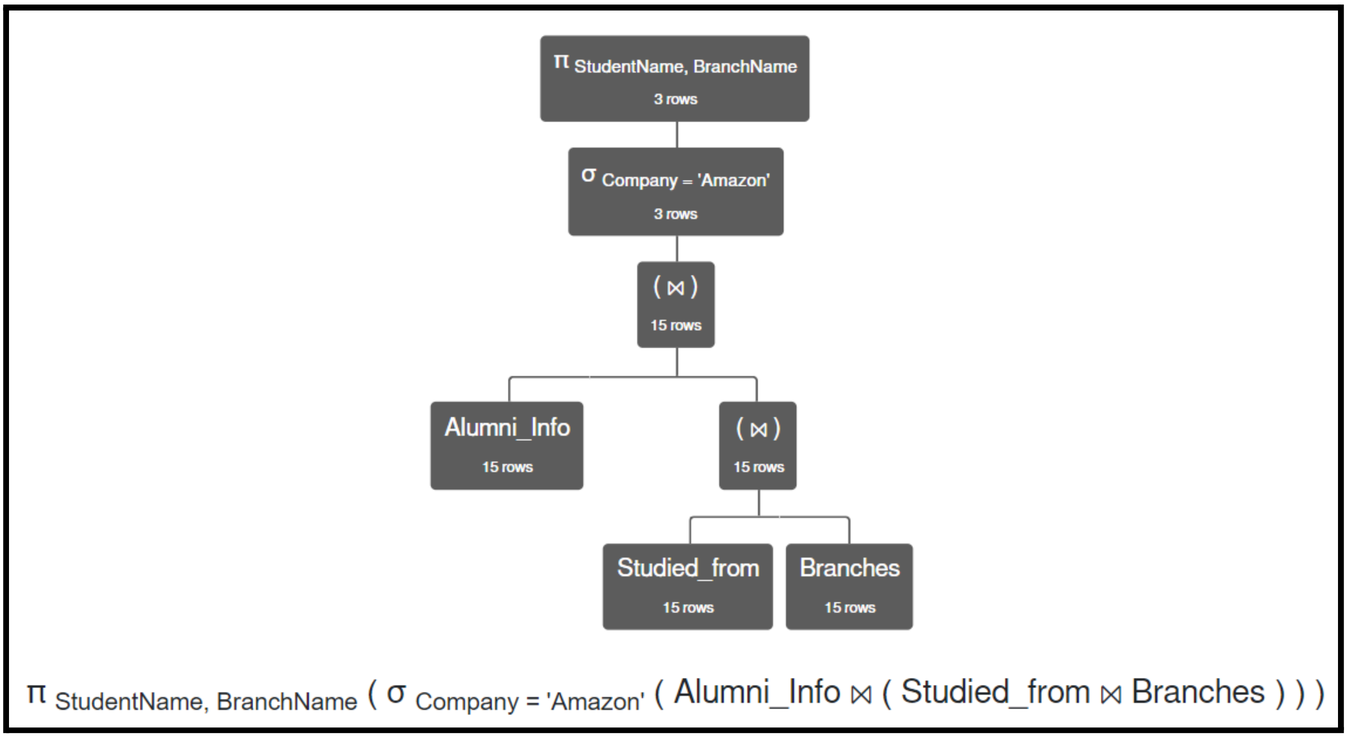
College_info.CollegeName	Faculty_info.FacultyName
'IIIT Pune'	'Piyush Warke'
'IIIT Pune'	'Priyanka Sharma'
'IIIT Kancheepurram'	'Mdhukar Reddy'
'IIIT Jabalpur'	'Vishal Singh'

2)Display the name of the alumni and their branch name,who are placed in ‘Amazon’.

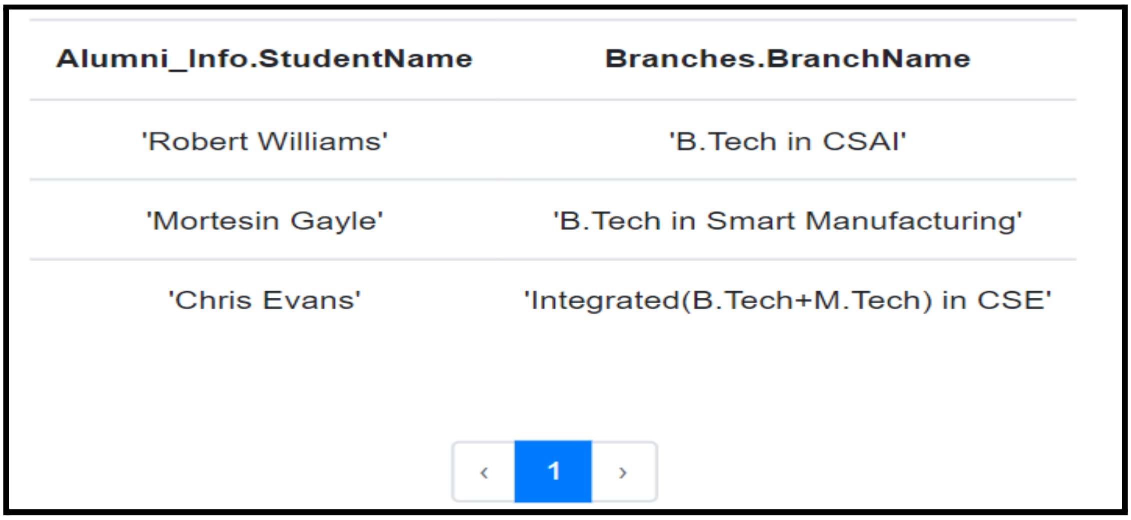
INPUT

π StudentName,BranchName (σ Company= 'Amazon' (Alumni_Info ⋈ (Studied_from ⋈ Branches)))

OUTPUT



$\pi_{\text{StudentName, BranchName}}$ ($\sigma_{\text{Company = 'Amazon'}}$ (Alumni_Info ⋈ (Studied_from ⋈ Branches)))



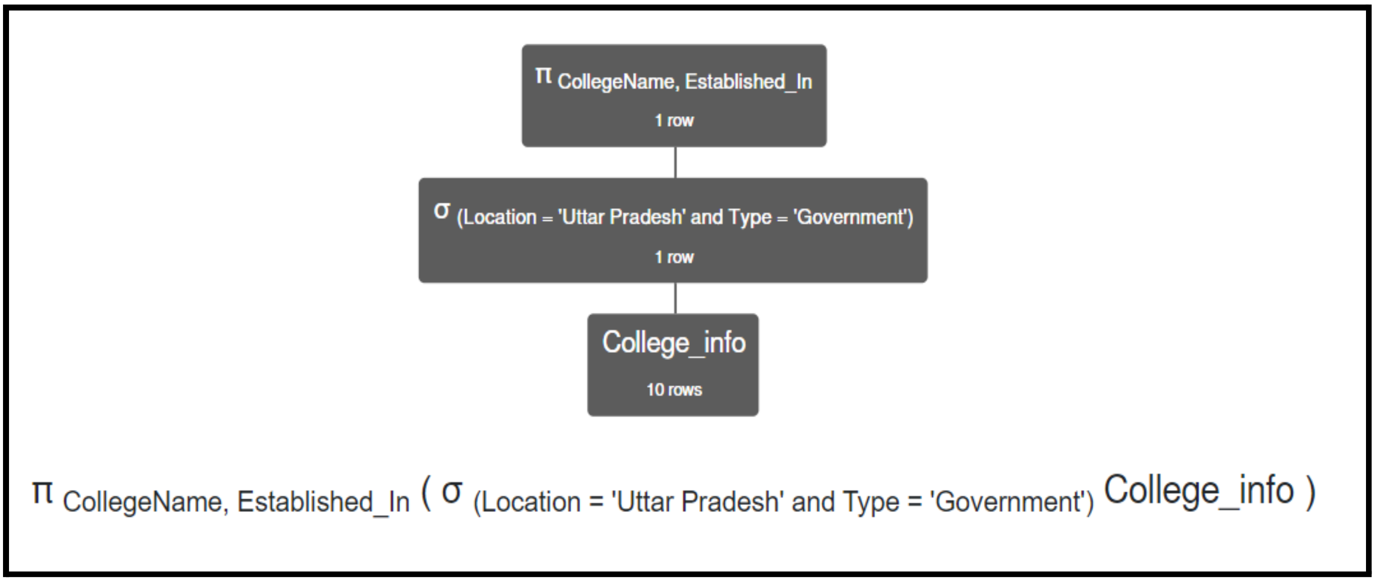
Alumni_Info.StudentName	Branches.BranchName
'Robert Williams'	'B.Tech in CSAI'
'Mortesin Gayle'	'B.Tech in Smart Manufacturing'
'Chris Evans'	'Integrated(B.Tech+M.Tech) in CSE'

3)Display all the Government colleges in Uttar Pradesh, and their foundation year.

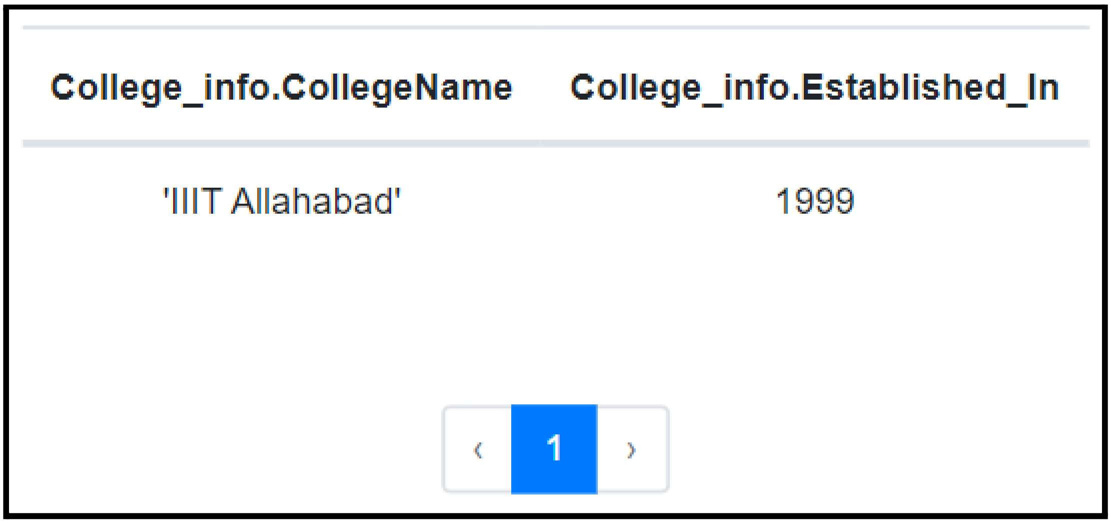
INPUT

π CollegeName,Established_In (σ (Location = 'Uttar Pradesh' ∧ Type = 'Government') College_info)

OUTPUT


π CollegeName, Established_In
1 row
σ (Location = 'Uttar Pradesh' and Type = 'Government')
1 row
College_info
10 rows

π CollegeName, Established_In (σ (Location = 'Uttar Pradesh' and Type = 'Government') College_info)



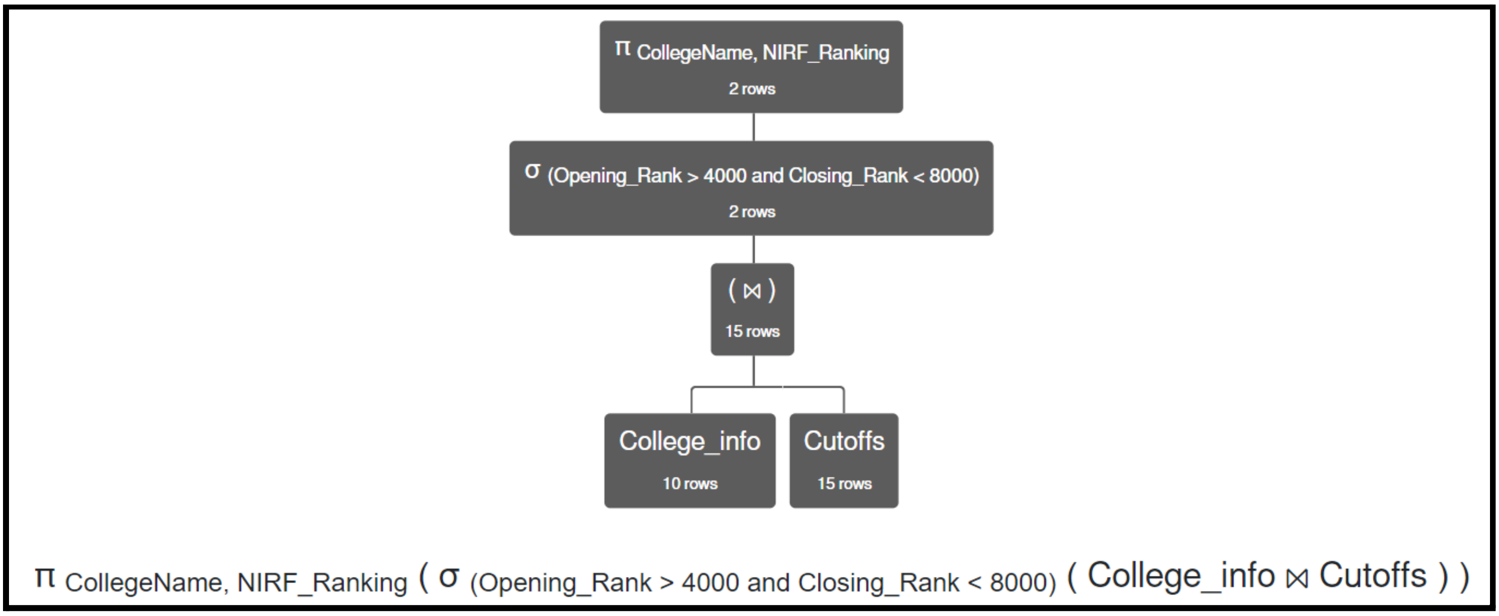
College_info.CollegeName	College_info.Established_In
'IIIT Allahabad'	1999

4)Display the name and NIRF Ranking of the college, available to a student, having rank in the range 4000-8000. (College Predictor Query)

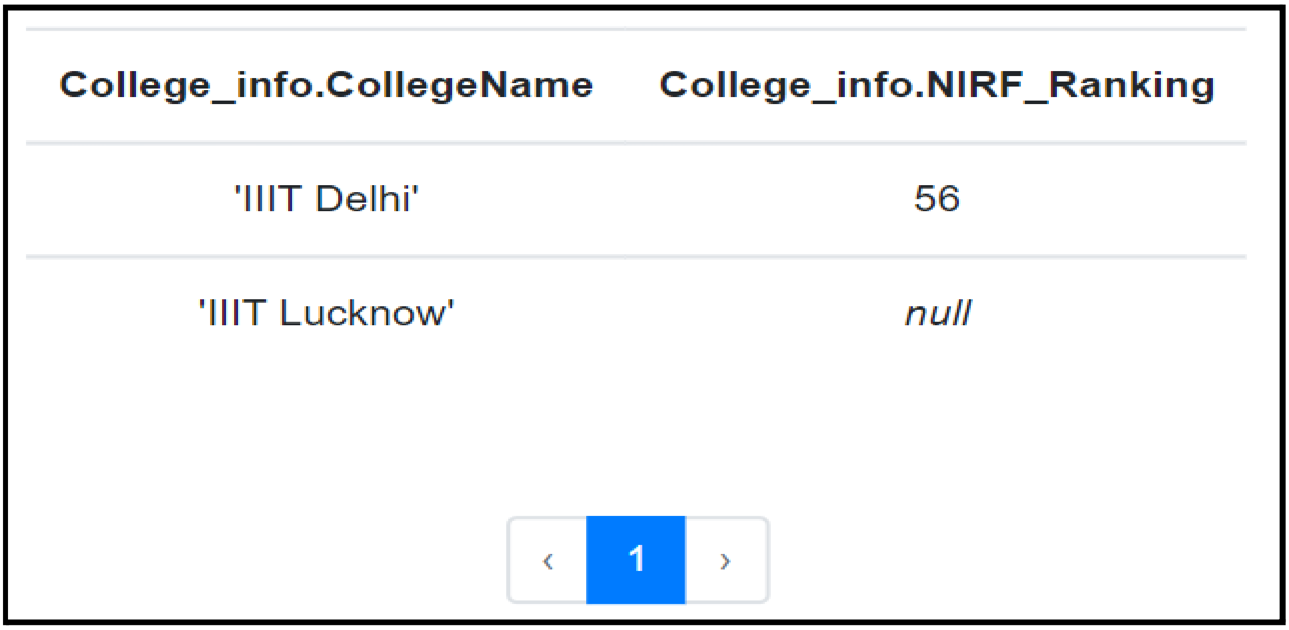
INPUT

π CollegeName, NIRF_Ranking (σ (Opening_Rank>4000 ∧ Closing_Rank <8000) (College_info ⋈ Cutoffs))

OUTPUT


π CollegeName, NIRF_Ranking
2 rows
σ (Opening_Rank > 4000 and Closing_Rank < 8000)
2 rows
(⋈)
15 rows
College_info
10 rows
Cutoffs
15 rows

π CollegeName, NIRF_Ranking (σ (Opening_Rank > 4000 and Closing_Rank < 8000) (College_info ⋈ Cutoffs))



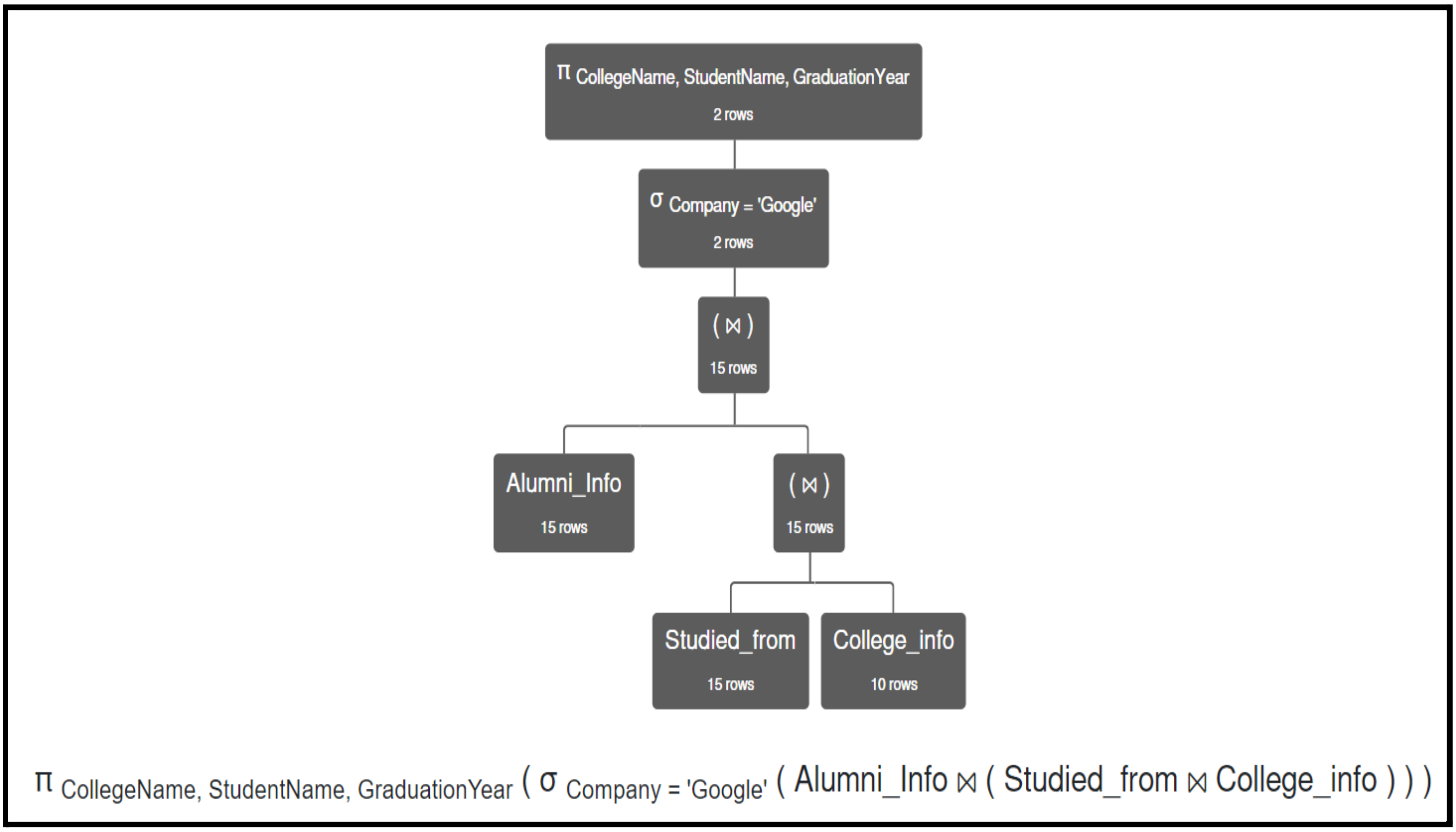
College_info.CollegeName	College_info.NIRF_Ranking
'IIIT Delhi'	56
'IIIT Lucknow'	*null*

5)Which colleges have alumni in ‘Google’ ? Also display the name and graduation year of each alumnus.

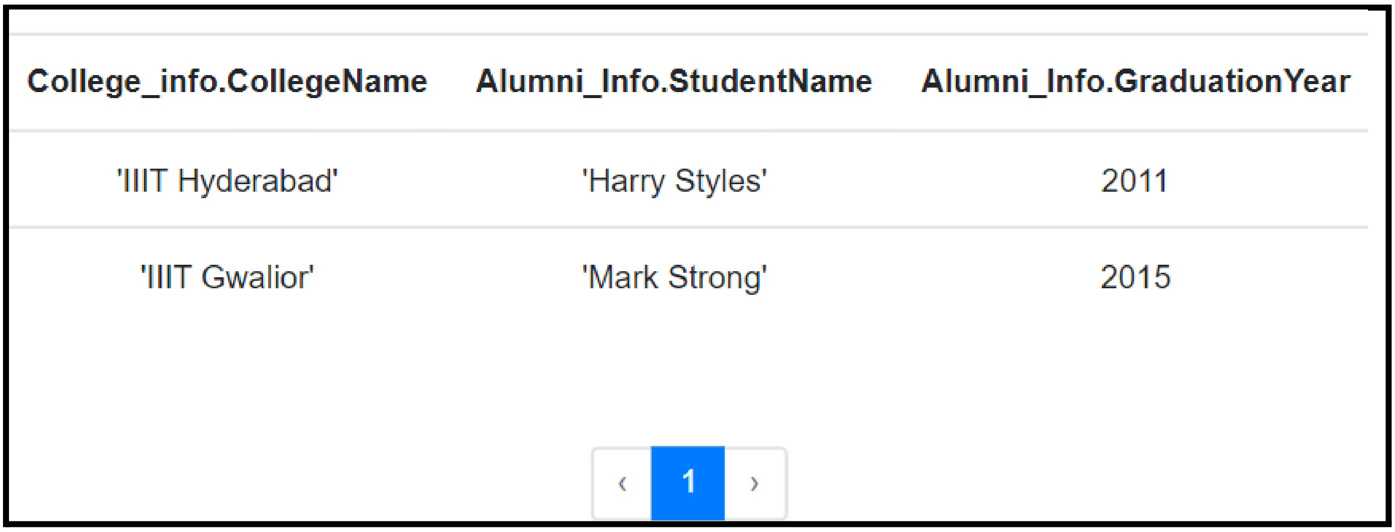
INPUT

π CollegeName,StudentName,GraduationYear (σ Company='Google' ((Alumni_Info ⋈ (Studied_from ⋈ College_info))))

OUTPUT



π CollegeName, StudentName, GraduationYear (σ Company = 'Google' (Alumni_Info ⋈ (Studied_from ⋈ College_info)))



College_info.CollegeName	Alumni_Info.StudentName	Alumni_Info.GraduationYear
'IIIT Hyderabad'	'Harry Styles'	2011
'IIIT Gwalior'	'Mark Strong'	2015

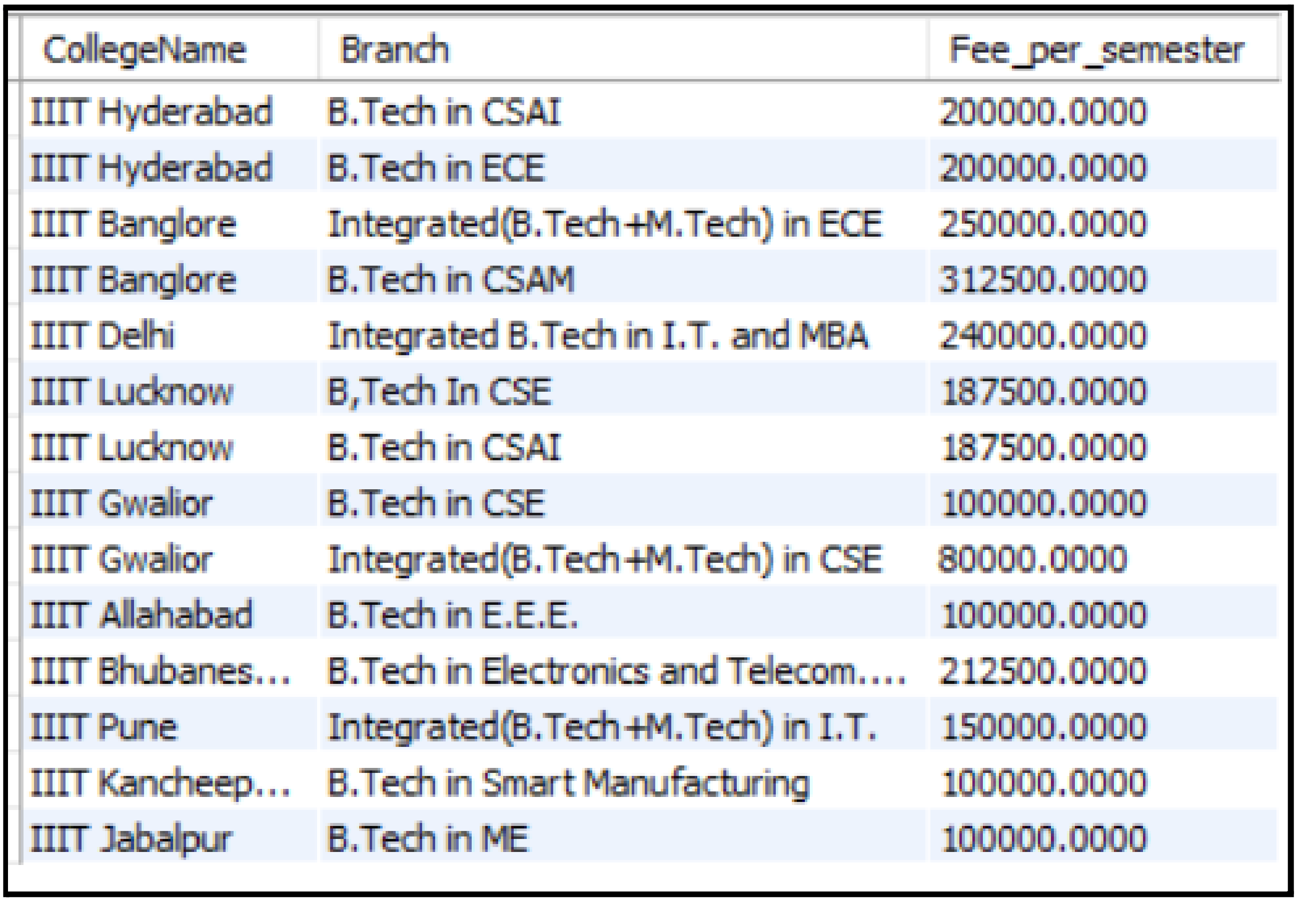
→ STRUCTURED QUERY LANGUAGE(SQL)

1)Display the fees per semester of every college.

SQL QUERY :

```
SELECT
   A.CollegeName AS CollegeName,
   C.BranchName AS Branch,
   A.fee_structure/(2*CourseDuration_Years) AS Fee_per_semester
 FROM
   College_info AS A,
   College_Branch AS B,
   branches AS C
WHERE
   A.CollegeID = B.CollegeID AND C.BranchID = B.BranchID;
```

OUTPUT :



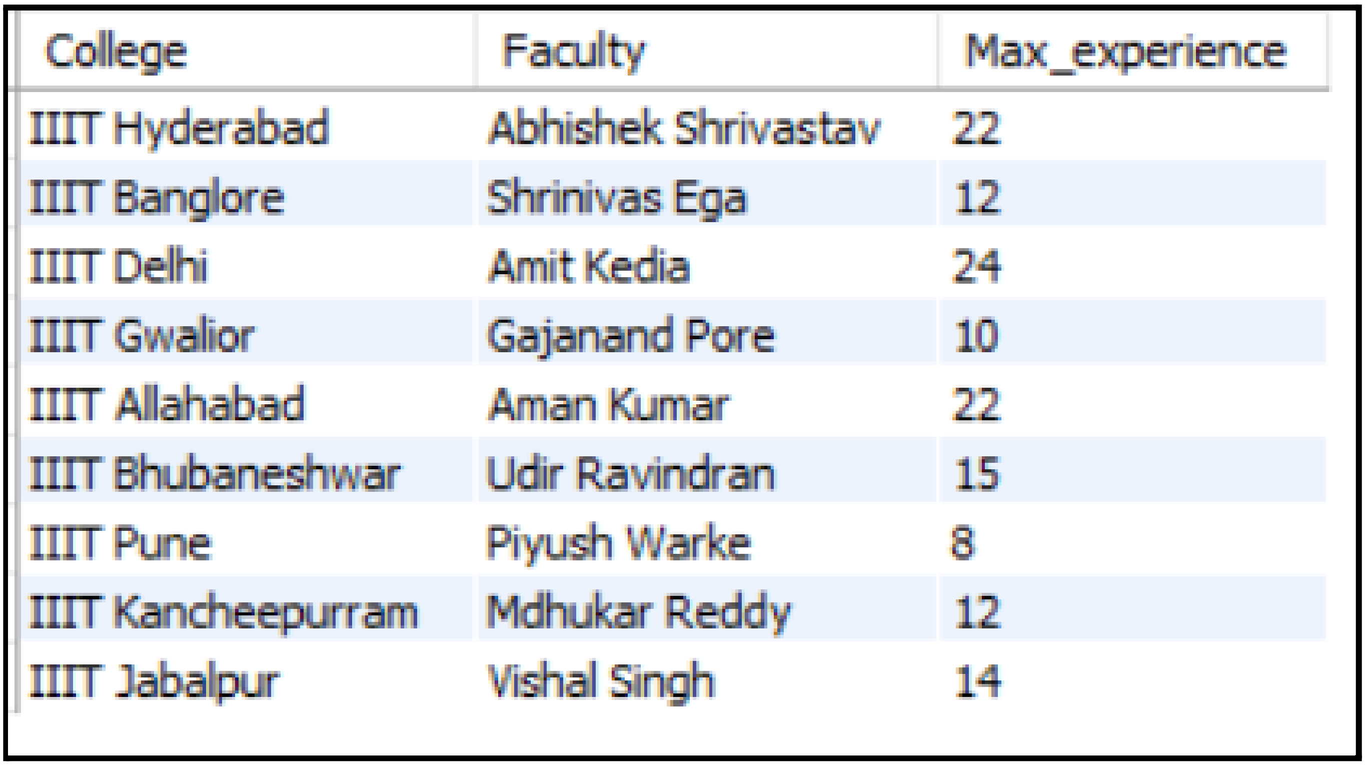
CollegeName	Branch	Fee_per_semester
IIIT Hyderabad	B.Tech in CSAI	200000.0000
IIIT Hyderabad	B.Tech in ECE	200000.0000
IIIT Banglore	Integrated(B.Tech+M.Tech) in ECE	250000.0000
IIIT Banglore	B.Tech in CSAM	312500.0000
IIIT Delhi	Integrated B.Tech in I.T. and MBA	240000.0000
IIIT Lucknow	B,Tech In CSE	187500.0000
IIIT Lucknow	B.Tech in CSAI	187500.0000
IIIT Gwalior	B.Tech in CSE	100000.0000
IIIT Gwalior	Integrated(B.Tech+M.Tech) in CSE	80000.0000
IIIT Allahabad	B.Tech in E.E.E.	100000.0000
IIIT Bhubanes...	B.Tech in Electronics and Telecom....	212500.0000
IIIT Pune	Integrated(B.Tech+M.Tech) in I.T.	150000.0000
IIIT Kancheep...	B.Tech in Smart Manufacturing	100000.0000
IIIT Jabalpur	B.Tech in ME	100000.0000

2)Display the names of the faculty with maximum experience for each college.

SQL QUERY :

```
SELECT
   C.CollegeName AS College,
   A.FacultyName AS Faculty,
   MAX(Experience) as Max_experience
FROM
   Faculty_info AS A
   INNER JOIN Teaches_In AS B
   ON A.FacultyID = B.FacultyID
   INNER JOIN College_info AS C
   ON C.CollegeID = B.CollegeID
GROUP BY
   C.CollegeID
```

OUTPUT :



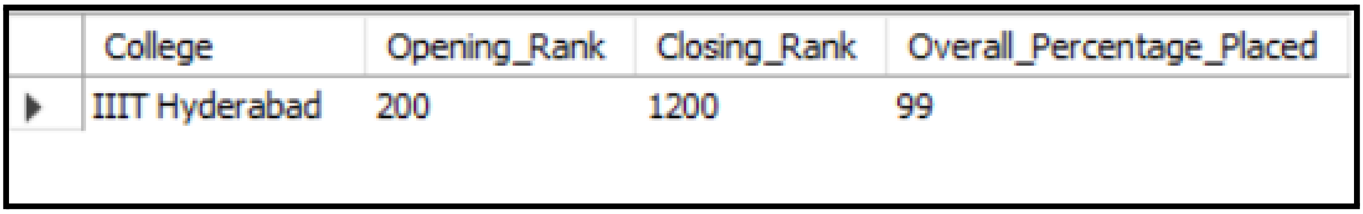
College	Faculty	Max_experience
IIIT Hyderabad	Abhishek Shrivastav	22
IIIT Banglore	Shrinivas Ega	12
IIIT Delhi	Amit Kedia	24
IIIT Gwalior	Gajanand Pore	10
IIIT Allahabad	Aman Kumar	22
IIIT Bhubaneshwar	Udir Ravindran	15
IIIT Pune	Piyush Warke	8
IIIT Kancheepurram	Mdhukar Reddy	12
IIIT Jabalpur	Vishal Singh	14

3)What are the opening and closing ranks for the college having the maximum students placed ? Also display the overall percentage of students placed in that college.

SQL QUERY :

```
SELECT
   college_info .CollegeName AS College,
   cutoff.Opening_Rank AS Opening_Rank,
   cutoff.Closing_Rank AS Closing_Rank,
   MAX(placements.Percentage_Placed) AS Overall_Percentage_Placed
FROM
   placements
   NATURAL JOIN cutoff
   NATURAL JOIN college_info
```

OUTPUT :



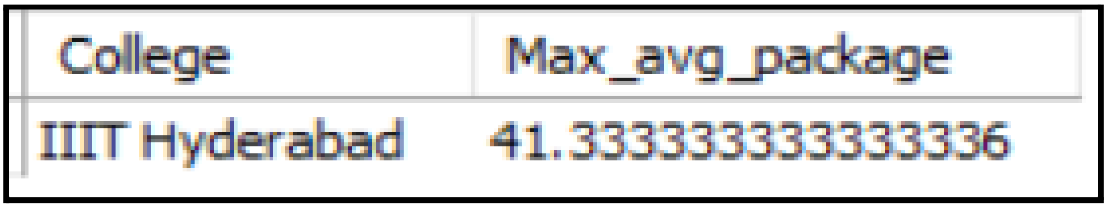
College	Opening_Rank	Closing_Rank	Overall_Percentage_Placed
IIIT Hyderabad	200	1200	99

4)Display the name of the colleges having the highest average package.

SQL QUERY :

```
SELECT
   C.CollegeName AS College,
   MAX(C.Avg_package) AS Max_avg_package
FROM
   (SELECT
      B.CollegeName AS CollegeName,
      avg(A.package_offered) AS Avg_package
   FROM
      Placements AS A
      INNER JOIN college_info AS B
      ON A.CollegeID = B.CollegeID
   GROUP BY
      A.CollegeID)
      AS C
```

OUTPUT :



College	Max_avg_package
IIIT Hyderabad	41.333333333333336

5)Display the list of faculties who teach 'Machine Learning', and their college, in descending order of their experience.

SQL QUERY :

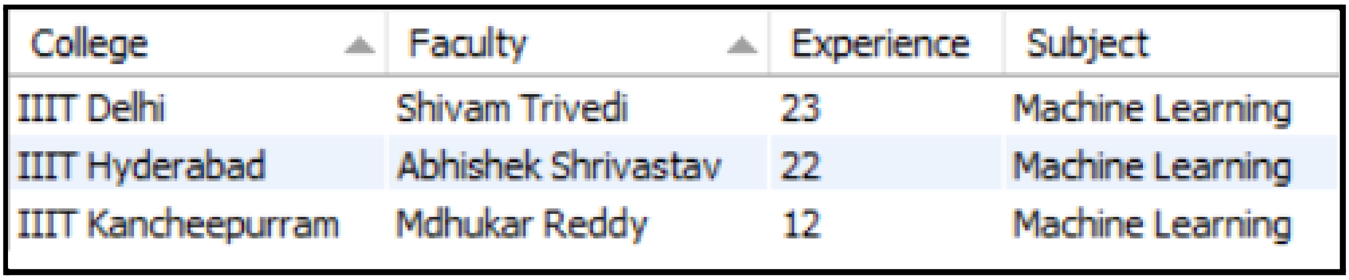
```
SELECT
   C.CollegeName AS College,
   A.FacultyName AS Faculty,
   A.Experience ,
   A.Subject

FROM
  ( faculty_info AS A
   INNER JOIN teaches_in AS B
   ON A.FacultyID = B.FacultyID )
   INNER JOIN college_info AS C
   ON C.CollegeID = B.CollegeID

 WHERE
       A.Subject= "Machine Learning"

ORDER BY A.Experience DESC
```

OUTPUT :



College	Faculty	Experience	Subject
IIIT Delhi	Shivam Trivedi	23	Machine Learning
IIIT Hyderabad	Abhishek Shrivastav	22	Machine Learning
IIIT Kancheepurram	Mdhukar Reddy	12	Machine Learning

CONCLUSION:

To summarise this was our analysis on different IIIT's across the country, their branches , location , faculties , ranking , placements , companies, cut-offs , alumni , research culture etc. We hope this will help the JEE aspirants to find the best college of their choice without any difficulties . We plan on extending our database including more factors such as reviews from actual students , number of seats , courses taught in every college , electives available , foreign exchange programme etc.

Thank You