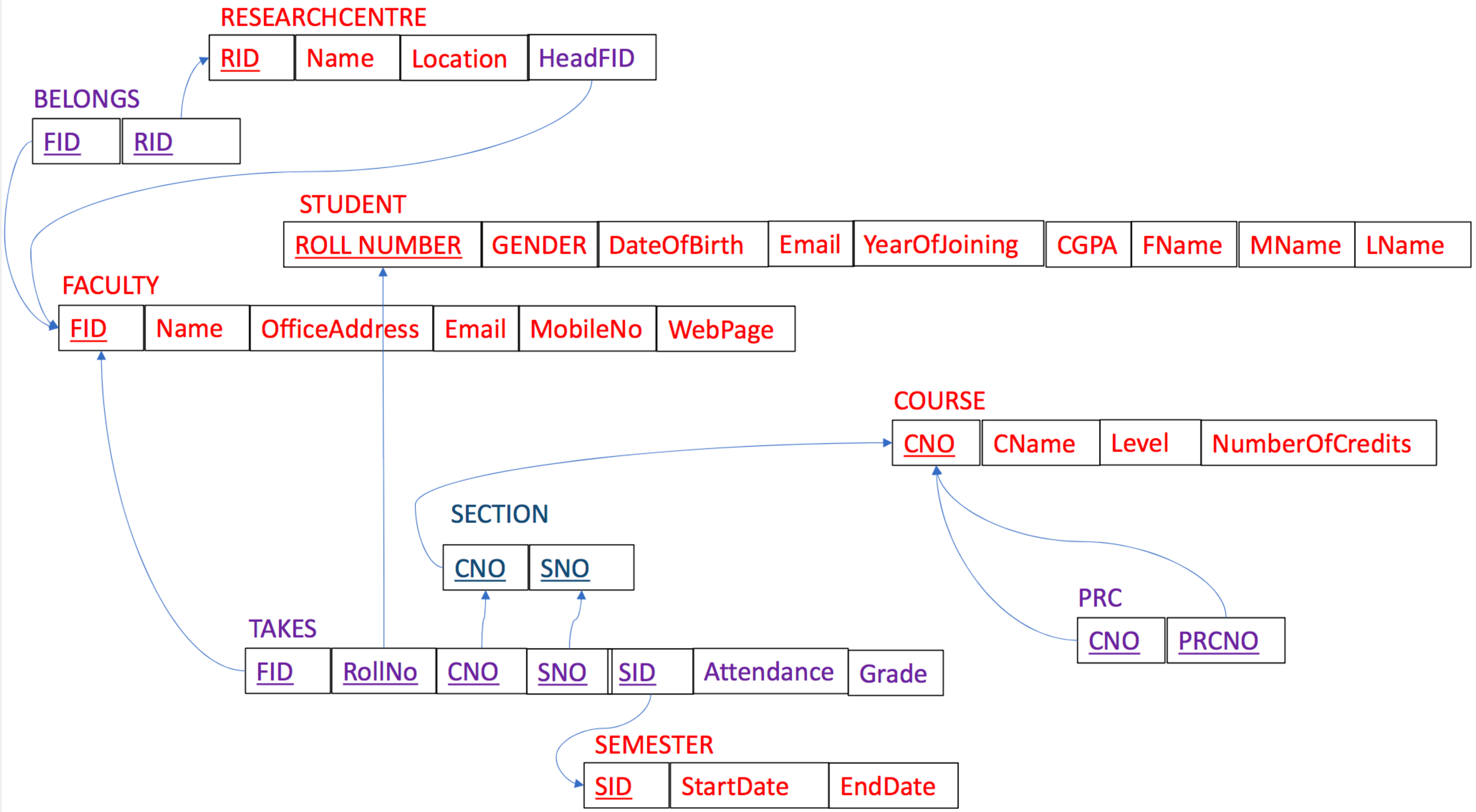
Intro to Databases

MySQL Class

Database as collection of tables

If you already haven't imported the database:

```
bash> mysql -u root -p < student_database.sql
```



RESEARCHCENTRE: RID | Name | Location | HeadFID

BELONGS: FID | RID

STUDENT: ROLL NUMBER | GENDER | DateOfBirth | Email | YearOfJoining | CGPA | FName | MName | LName

FACULTY: FID | Name | OfficeAddress | Email | MobileNo | WebPage

COURSE: CNO | CName | Level | NumberOfCredits

SECTION: CNO | SNO

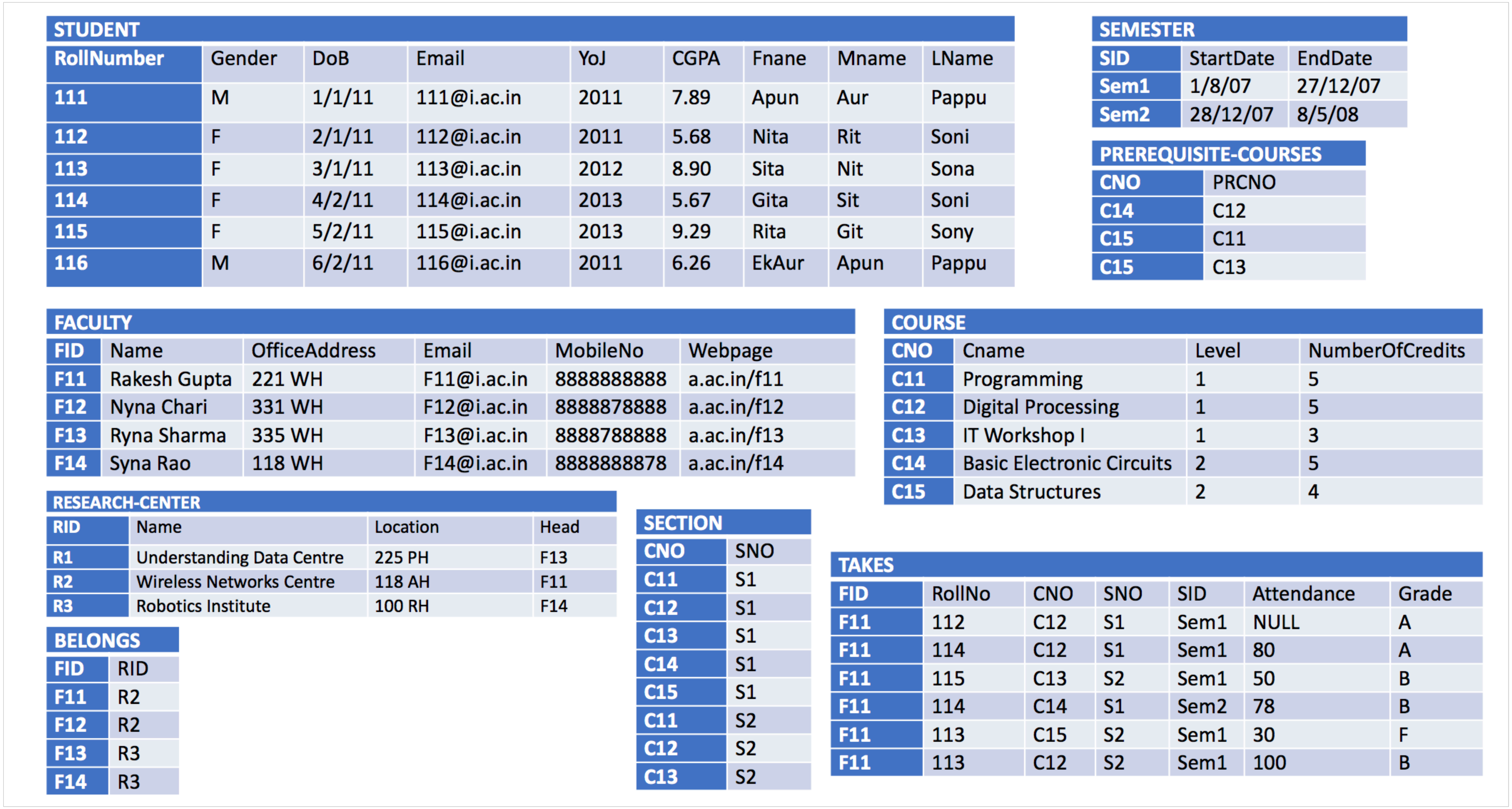
PRC: CNO | PRCNO

TAKES: FID | RollNo | CNO | SNO | SID | Attendance | Grade

SEMESTER: SID | StartDate | EndDate

Relational Schema of the Database

Sample Database with Tables



STUDENT

RollNumber	Gender	DoB	Email	YoJ	CGPA	Fnane	Mname	LName
111	M	1/1/11	111@i.ac.in	2011	7.89	Apun	Aur	Pappu
112	F	2/1/11	112@i.ac.in	2011	5.68	Nita	Rit	Soni
113	F	3/1/11	113@i.ac.in	2012	8.90	Sita	Nit	Sona
114	F	4/2/11	114@i.ac.in	2013	5.67	Gita	Sit	Soni
115	F	5/2/11	115@i.ac.in	2013	9.29	Rita	Git	Sony
116	M	6/2/11	116@i.ac.in	2011	6.26	EkAur	Apun	Pappu

SEMESTER

SID	StartDate	EndDate
Sem1	1/8/07	27/12/07
Sem2	28/12/07	8/5/08

PREREQUISITE-COURSES

CNO	PRCNO
C14	C12
C15	C11
C15	C13

FACULTY

FID	Name	OfficeAddress	Email	MobileNo	Webpage
F11	Rakesh Gupta	221 WH	F11@i.ac.in	8888888888	a.ac.in/f11
F12	Nyna Chari	331 WH	F12@i.ac.in	8888878888	a.ac.in/f12
F13	Ryna Sharma	335 WH	F13@i.ac.in	8888788888	a.ac.in/f13
F14	Syna Rao	118 WH	F14@i.ac.in	8888888878	a.ac.in/f14

COURSE

CNO	Cname	Level	NumberOfCredits
C11	Programming	1	5
C12	Digital Processing	1	5
C13	IT Workshop I	1	3
C14	Basic Electronic Circuits	2	5
C15	Data Structures	2	4

RESEARCH-CENTER

RID	Name	Location	Head
R1	Understanding Data Centre	225 PH	F13
R2	Wireless Networks Centre	118 AH	F11
R3	Robotics Institute	100 RH	F14

BELONGS

FID	RID
F11	R2
F12	R2
F13	R3
F14	R3

SECTION

CNO	SNO
C11	S1
C12	S1
C13	S1
C14	S1
C15	S1
C11	S2
C12	S2
C13	S2

TAKES

FID	RollNo	CNO	SNO	SID	Attendance	Grade
F11	112	C12	S1	Sem1	NULL	A
F11	114	C12	S1	Sem1	80	A
F11	115	C13	S2	Sem1	50	B
F11	114	C14	S1	Sem2	78	B
F11	113	C15	S2	Sem1	30	F
F11	113	C12	S2	Sem1	100	B

Starting with Databases

```
mysql> show databases;

mysql> create database <database name>;

mysql> use <database name>;

mysql> drop database <database name>;
```

Intro to Tables

```
mysql> show tables;
mysql> create table <table name> (
            id  INT AUTO_INCREMENT,
            Name varchar(50) NOT NULL,
            DOB DATE,
            email varchar(100),
            PRIMARY KEY(id)
            )
```

Optional Arguments for attributes:

- NOT NULL
- DEFAULT
- UNIQUE
- AUTO_INCREMENT
- PRIMARY KEY

```
mysql> describe <table name>;
```

Handling Data

mysql> INSERT INTO <table name> (field1, field2, ..) values
(value1, value2 , …);

Selecting all columns and records from table

- **mysql>** select * from <table name>;

Selecting a particular column:

- **mysql>** select column1, column2 from <tablename>;

LIMIT to top 10 of the Records:

mysql> select * from STUDENT LIMIT 10;

Handling Data

Using the *WHERE* Keyword:

```
mysql> select * from <table name> WHERE field1 = value1;
```

Ex:

Selecting all columns and records from table where name of student is ‘leo’ (not case-sensitive).

- `mysql> select * from students where name=‘leo’;`

Updating Records:

```
mysql> update table SET field1 = value1, … WHERE field1 = value1, …;
```

Deleting Records:

```
mysql> delete table WHERE field1 = value1, …;
```

```
mysql> drop table <table name>;
```

Handling Data

Using *AND, OR, NOT* keywords:

```
mysql> select * from <table name> WHERE field1 = value1 AND
field2 = value2 AND field3 = value3;
```

The comparison also works with regular relational operators like >, <, >=, <=, =, != etc.

Ex:

```
mysql> select * from students WHERE (name = ‘leo’ AND age
>= 18) OR (name = ‘kate’ AND age >= 12);
```

The **NOT** Keyword **negates** the condition. All complementary records are returned.

Some Keywords

- ***DISTINCT***

```
mysql> select DISTINCT age from STUDENTS;
```

- ***LIKE (for strings)***

```
mysql> select * from STUDENT where Fname like ‘A____';
```

```
mysql> select * from STUDENT where email like ‘%gmail%’;
```

- ***ORDER BY (for sorting records)***

```
mysql> select * from STUDENT ORDER BY RollNo;
mysql> select * from STUDENT ORDER BY RollNo,Age DESC;
```

Some Keywords

- ***AS***

```
mysql> select Name, Age AS FavNum from STUDENTS;
```

- ***IS NOT NULL (detecting NULL records/entries)***

```
mysql> select * from STUDENT where name IS NULL;
```

- ***CURDATE***

```
mysql> select CURDATE();
```

```
mysql> select *,TIMESTAMPDIFF(Year,DoB,CURDATE()) as Age
from STUDENT;
```

- ***MONTH, DAY, YEAR***

```
mysql> select * from STUDENT where Month(Dob) = 12;
mysql> select * from STUDENT where Day(Dob) = 24;
mysql> select * from STUDENT where Year(Dob) = 1998;
```

Aggregate Functions

- ***COUNT***

```
mysql> select COUNT(*) from STUDENT;
```

- ***GROUP BY (for grouping records)***

```
mysql> select Gender, count(*) from STUDENT GROUP BY Gender;
```

- ***MAX(colname), MIN(colname), AVG(colname), SUM(colname)***

```
mysql> select MAX(Age), MIN(Age) from STUDENT;
mysql> select AVG(Age), SUM(Age), count(*) from STUDENT;
mysql> select Gender, MAX(*) from STUDENT GROUP BY Gender;
mysql> select Gender, Age, MAX(*) from STUDENT GROUP BY Gender, Age;
```

- ***HAVING (checks on aggregates)***

```
mysql> select YoJ, count(*) from STUDENT group by YoJ HAVING count(*) > 4;
```

What does a join do?

ID	Name	Age
1	Adam	21
2	Alex	24
3	Chris	22

ID	City
1	London
2	Delhi
3	Tokyo

ID	Name	Age	City
1	Adam	21	London
2	Alex	24	Delhi
3	Chris	22	Tokyo

INNER JOIN (Default)

ID	Name	Age
1	Adam	21
2	Alex	24
3	Chris	22
4	Eva	25

ID	City
1	London
2	Delhi
3	Tokyo
6	NULL

ID	Name	Age	City
1	Adam	21	London
2	Alex	24	Delhi
3	Chris	22	Tokyo

LEFT JOIN

ID	Name	Age
1	Adam	21
2	Alex	24
3	Chris	22
4	Eva	25

ID	City
1	London
2	Delhi
3	Tokyo
6	NULL

ID	ID	Name	Age	City
1	1	Adam	21	London
2	2	Alex	24	Delhi
3	3	Chris	22	Tokyo
4	NULL	Eva	25	NULL

RIGHT JOIN

ID	Name	Age
1	Adam	21
2	Alex	24
3	Chris	22
4	Eva	25

ID	City
1	London
2	Delhi
3	Tokyo
6	Paris

ID	ID	Name	Age	City
1	1	Adam	21	London
2	2	Alex	24	Delhi
3	3	Chris	22	Tokyo
NULL	6	NULL	NULL	Paris

FULL JOIN

ID	Name	Age
1	Adam	21
2	Alex	24
3	Chris	22
4	Eva	25

ID	City
1	London
2	Delhi
3	Tokyo
6	NULL

ID	ID	Name	Age	City
1	1	Adam	21	London
2	2	Alex	24	Delhi
3	3	Chris	22	Tokyo
4	NULL	Eva	25	NULL
NULL	6	NULL	NULL	Paris

Joining Tables

Regular:

```
SELECT column-name-list
from table-name1, table-name2
WHERE
table-name1.column-name = table-name2.column-name;
```

Ex:

```
mysql> select S.FName,T.Grade, T.CNO
 from student as S, takes as T
 where S.RollNumber = T.RollNo AND FName like ‘a%’;
```

How to make it case-sensitive ?

Joining Tables

Joins:

Must use keyword ‘ON’

```
mysql> select * from table1 as T1 INNER JOIN table2 as T2
ON T1.id = T2.id;
```

```
mysql> select * from table1 as T1 RIGHT JOIN table2 as T2
ON T1.id = T2.id;
```

```
mysql> select * from table1 as T1 LEFT JOIN table2 as T2
ON T1.id = T2.id;
```

Joining Tables

UNION :

Ex:

```
mysql> (select * from STUDENT where Fname = "Nita")
UNION (select * from STUDENT where Fname = “Akhil");
```

```
mysql> (select id from STUDENT) UNION (select id from
FACULTY);
```

```
mysql> (select RollNumber from STUDENT) UNION (select
Name from FACULTY);
```

Nested Queries

IN:

```
mysql> select * from STUDENT where Fname IN (“Nita”,
“Akhil”);
```

Ex:

Select Names of Students who took a course under Rakesh Gupta.

```
mysql> select Name from Student
where RollNumber IN (select RollNo from Faculty as F, Takes
as T where F.name = “Rakesh Gupta” and F.FID = T.FID);
```

More Keywords

```
mysql> select * from STUDENT where EXISTS ( select *
from FACULTY as F where F.Name = ‘Alex’);
```

```
mysql> select * from STUDENT where NOT EXISTS ( select
* from FACULTY as F where F.Name = ‘Alex’);
```

Examples

Retrieve the number of students who took course Information Technology and their average CGPA.

```
mysql> select AVG(CGPA), count(*)
 from student as S, takes as T, course as C
 where S.RollNumber = T.RollNo AND C.CNO = T.CNO AND
C.CName = ‘information technology';
```

Examples

Retrieve the names of the faculty who is not head of a research centre.

```
mysql> select * from faculty as F LEFT JOIN
research_center as R on F.FID = R.Head where RID is NULL;
```

```
mysql> select * from faculty where FID NOT IN (select FID
from faculty as F, research_center as R where F.FID =
R.Head);
```

Examples

Retrieve the course name that have at least one student got an 'A' grade.

```
mysql> SELECT CNO,Grade, count(*) FROM TAKES
GROUP BY CNO, Grade having Grade='A' and count(*)>1;
```

Finally

References:

Official Documentation from :

https://dev.mysql.com/doc/refman/5.7/en/

Do check out other useful commands:

- REGEXP - Regex can be used instead of LIKE keyword. Makes searching for patterns much easier.
- ALTER TABLE - To add/remove columns from a Table
- MAKE VIEW - To make views for the queries