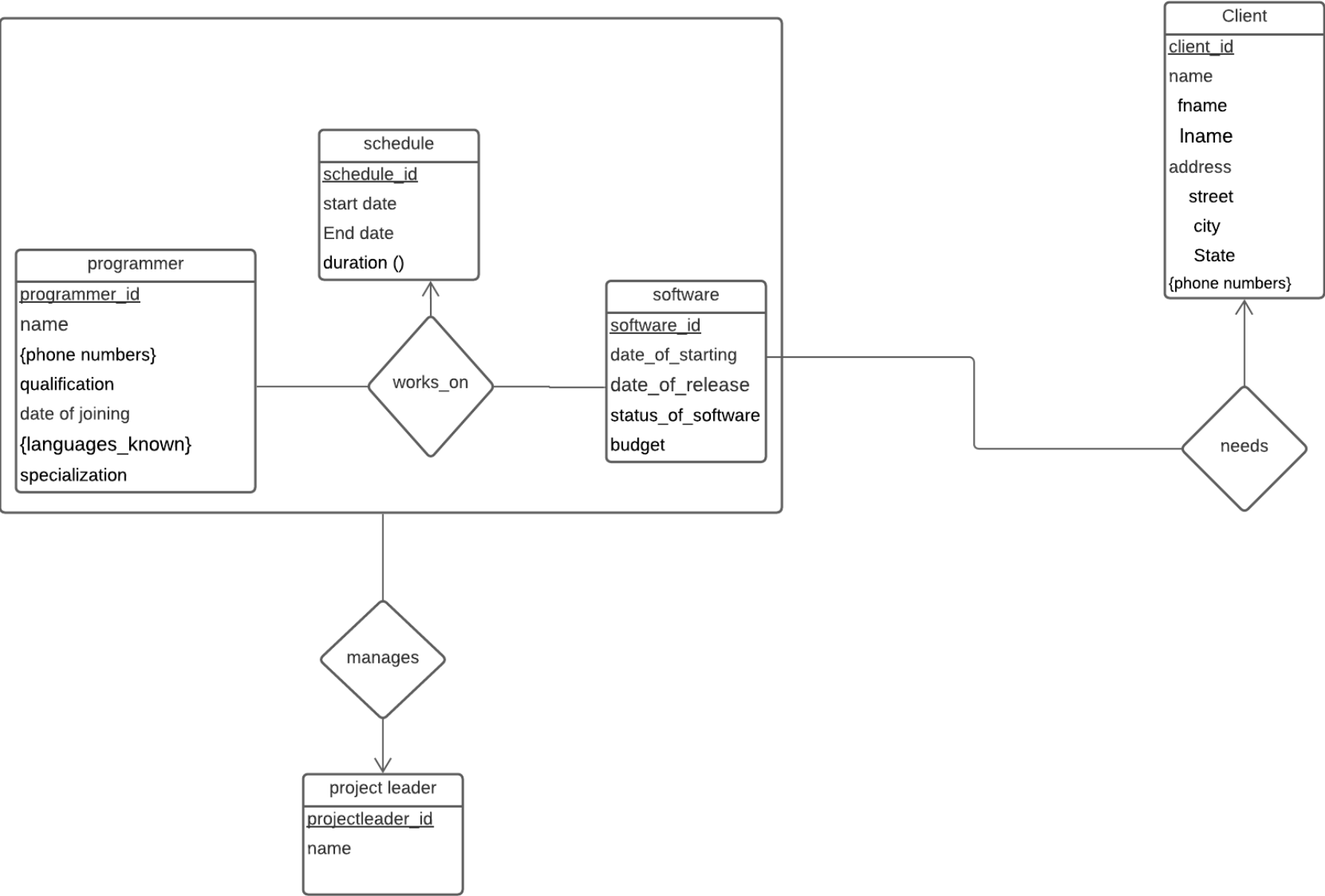


# E-HUB

## Abstract:

E-hub is a software company that provides various types of software solutions to clients across India. It has recruited various programmers for software development. Each programmer is identified by the id, date of joining, experience, qualification, specialization, programming\_languages\_known. Specialization signifies the programming language for which he is most specialized in. The company may develop more than one software for one client. Each client is identified by his id, name, address, phone numbers. The details of software developed for the client such as date\_of\_commencement, date\_of\_release, status\_of\_software, etc. are also maintained. Each software may be developed by more than one programmer with one project leader. The software's working by a programmer will have schedule\_id, start\_date, end\_date.

## Er Diagram:



Schema Diagram:

Composite attributes – name,address(client entity)

Multivalued attributes -languages\_known(programmer entity)

Derived attributes – duration(schedule entity)

The strong entity set reduces to schema with the same attributes.

The composite attributes get simplified by creating separate attributes.

Programmer(programmer\_id, name , phone\_numbers, qualification,specialization date\_of\_joining)

Languages\_known(programmer\_id,language)

Programmer\_phoneno(programmer\_id,phone\_no)

Client(client\_id,fname,lname,street,city,state)

Client\_phonenumbers(client\_id,phonenumber)

Software(software\_id,date\_of\_starting,date\_of\_release,status\_of\_software,client\_id)

Project\_leader(projectleader\_id,name)

Manages(projectleader\_id,software\_id,schedule\_id,programmer\_id)

Schedule(schedule\_id,start\_date,end\_date)

## DBMS PROJECT REVIEW-02

### 1NF:

- A relation is said to be in 1NF if and only if all the attributes are having atomic domains(cannot be decomposed into smaller pieces).
  - Address is a composite attribute which contains street, city, state. So, we will split address into street,city,state.
  - Programming\_languages\_known is multivalued attribute. So, we will create two separate columns as programming\_languages\_known1 and programming\_languages\_known2
  - Client name is composite attribute contains fname, lname. So, we will split client name into fname and lname.

After 1nf:

software_id	software_name	date_of_release	status_of_software	client_id	fname	lname	street	city	state	phone_number	programmer_id	name	qualification	specialization	Date_of_joining	Programming_languages_known1	Programming_languages_known2	schedule_id	start_date	end_date	project_leader_id	leader_name
S001	software_1	10-11-2019	released	C1	Karthik	Sharma	street-4	mumbai	Maharashtra	9897678955	P001	Rohit	B.Tech	Java	10-05-2019	Java	Python	SC1	07-09-2019	27-09-2019	P001	Rahul
S001	Software_1	10-11-2019	released	C1	Karthik	Sharma	street-4	mumbai	Maharashtra	9897678955	P002	Raju	M.Sc	C++	11-06-2019	C++	Java	SC6	27-09-2019	01-11-2019	P001	Rahul
S002	software_2	09-03-2020	released	C3	Ravi	Teja	street-11	vizag	Andhra	9090877909	P003	Prasanth	M.Sc	Java	12-05-2019	Python	Java	SC2	03-01-2020	09-02-2020	P002	Siddharth
S002	software_2	09-03-2020	released	C3	Ravi	Teja	street-11	vizag	Andhra	9090877909	P004	Aravind	B.tech	Python	13-07-2019	C++	Python	SC7	#####	01-03-2020	P002	Siddharth
S003	software_3	15-06-2020	released	C1	Karthik	Sharma	street-4	mumbai	Maharashtra	9897678955	P005	Ramesh	M.sc	C++	14-05-2019	C++	Python	SC3	10-04-2020	14-04-2020	P003	Krishna
S003	software_3	15-06-2020	released	C1	Karthik	Sharma	street-4	mumbai	Maharashtra	9897678955	P006	Sundar	B.Tech	C	15-08-2019	Python	C	SC8	11-04-2020	29-04-2020	P003	Krishna
S003	software_3	15-06-2020	released	C1	Karthik	Sharma	street-4	mumbai	Maharashtra	9897678955	P007	Kunal	M.Sc	Kotlin	16-07-2019	Python	Kotlin	SC9	#####	27-05-2020	P003	Krishna
S003	software_3	15-06-2020	released	C1	Karthik	Sharma	street-4	mumbai	Maharashtra	9897678955	P001	Rohit	B.Tech	Java	10-05-2019	Java	Python	SC10	#####	14-06-2020	P003	Krishna
S004	software_4	20-08-2020	released	C2	Ruhani	Singh	street-3	kolkata	westbengal	9645463791	P002	Raju	M.Sc	C++	11-06-2019	C++	Java	SC4	#####	20-07-2020	P002	Siddharth
S004	software_4	20-08-2020	released	C2	Ruhani	Singh	street-3	kolkata	westbengal	9645463791	P005	Ramesh	M.sc	C++	14-05-2019	C++	Python	SC11	11-07-2020	30-07-2020	P002	Siddharth
S004	software_4	20-08-2020	released	C2	Ruhani	Singh	street-3	kolkata	westbengal	9645463791	P006	Sundar	B.Tech	C	15-08-2019	Python	C	SC12	#####	14-08-2020	P002	Siddharth
S005	software_5	19-12-2020	in progress	C4	Koushik	Jami	Jagadama	vizag	Andhra	9090877708	P007	Kunal	M.Sc	Kotlin	16-07-2019	Python	Kotlin	SC5	#####	01-12-2020	P003	Krishna
S005	software_5	19-12-2020	in progress	C4	Koushik	Jami	Jagadama	Vizag	Andhra	9090877708	P003	Prasanth	M.Sc	Java	10-05-2019	Python	Java	SC13	01-10-2020	13-12-2020	P003	Krishna

### Functional Dependencies:

Software\_id -> software name, date\_of\_release, status\_of\_software, projectleader\_id, name, client\_id, fname, lname, street, city, state

ClientID -> fname, lname,street,city,state

Programmer\_id -> name, specialization, qualification, date\_of\_joining, programming\_languages\_known1, programming\_languages\_known2

Software\_id,programmer\_id -> schedule\_id, start\_date, end\_date

Schedule\_id -> start\_date, end\_date

Projectleader\_id -> name

Candidate key is software\_idprogrammer\_id

### Second Normal Form(2NF):

- A relation is said to be in 2NF if and only if it is in 1NF.
- There should not be any partial dependency present in the relation.

*P.D = proper subset of CK -----> non-prime attributes*

In 1nf table the partial dependencies are software\_id -> software\_name, date\_of\_release, status\_of\_software, projectleader\_id, name, client\_id, fname, lname, street, city

Programmer\_id -> name, specialization, qualification, date\_of\_joining, programming\_languages\_known1, programming\_languages\_known2

In 1nf table the full dependency is software\_id, programmer\_id -> schedule\_id, start\_date, end\_date

After Decomposing tables in 2nf the tables are:

SoftwareClient(2NF):

<u>Software_id</u>	software name	Date_of_release	Status_of_software	Projectleader_id	name	Client_id	fname	lname	street	city	State
--------------------	---------------	-----------------	--------------------	------------------	------	-----------	-------	-------	--------	------	-------

In this table transitive dependencies are present. They are projectleader\_id ->name

Client\_id->fname, lname, street, city, state

software_id	software_name	date_of_releas	status_of_software	client_id	fname	lname	street	city	state	phonenumber	projectleader_id	name
S001	software_1	10-11-2019	released	C1	Karthik	Sharma	street -4	mumbai	Maharastra	9897678955	P001	Rahul
S002	software_2	09-03-2020	released	C3	Ravi	Teja	street-11	vizag	Andhra	9090877909	P002	Siddharth
S003	software_3	15-06-2020	released	C1	Karthik	Sharma	street-4	mumbai	Maharastra	9897678955	P003	Krishna
S004	software_4	20-08-2020	released	C2	Ruhani	Singh	street-3	Kolkata	Westbengal	9645463791	P002	Siddharth
S005	software_5	19-12-2020	in progress	C4	Koushik	jami	jagadama	vizag	Andhra	9090877708	P003	Krishna

Programmer(3NF):

<u>Programmer_id</u>	name	specialization	qualification	Date_of_joining	Programming_languages_known1	Programming_languages_known2
----------------------	------	----------------	---------------	-----------------	------------------------------	------------------------------

In this table there are no transitive dependencies so, this table is already in 3nf.

programmer_id	name	qualification	specialization	Date_of_joining	Programming_languages_known_1	Programming_languages_known_2
P001	Rohit	B.Tech	Java	10-05-2019	Java	Python
P002	Raju	M.Sc	C++	11-06-2019	C++	Java
P003	Prasanth	M.Sc	Java	12-05-2019	Python	Java
P004	Aravind	B.tech	Python	13-07-2019	C++	Python
P005	Ramesh	M.sc	C++	14-05-2019	C++	Python
P006	Sundar	B.Tech	C	15-08-2019	Python	C
P007	Kunal	M.Sc	Kotlin	16-07-2019	Python	Kotlin

SoftwareProjectSchedule(2NF):

Software_id	Programmer_id	Schedule_id	Start_date	End_date
-------------	---------------	-------------	------------	----------

In this table transitive dependency is schedule\_id -> start\_date, End\_Date

software_id	programmer_id	schedule_id	start_date	end_date
S001	P001	SC1	7/9/2019	27-09-2019
S001	P002	SC6	27-09-2019	1/11/2019
S002	P003	SC2	3/1/2020	9/2/2020
S002	P004	SC7	9/2/2020	1/3/2020
S003	P005	SC3	10/4/2020	14-04-2020
S003	P006	SC8	11/4/2020	29-04-2020
S003	P007	SC9	29-04-2020	27-05-2020
S003	P001	SC10	27-05-2020	14-06-2020
S004	P002	SC4	6/4/2020	20-07-2020
S004	P005	SC11	11/7/2020	30-07-2020
S004	P006	SC12	9/8/2020	1/12/2020
S005	P007	SC5	9/8/2020	1/12/2020
S005	P003	SC13	1/10/2020	13-12-2020

### Third Normal Form(3NF):

- A relation is said to be in 3NF if and only if it is 2NF.
- There should not be any transitive dependency for non prime attributes in the relation.

T.D = non-prime attribute ---> non-prime attribute

After decomposing the tables in 3nf the tables are:

### Software(3NF):

<u>Software id</u>	Software name	Date_of_relea se	Status_of_soft ware	Projectleader _id	Client_id
--------------------	---------------	---------------------	------------------------	----------------------	-----------



software_id	software_name	date_of_releas	status_of_software	project_leader_id	client_id
S001	software_1	10-11-2019	released	PI001	C1
S002	software_2	09-03-2020	released	PI002	C3
S003	software_2	09-03-2020	released	PI003	C1
S004	software_4	20-08-2020	released	PI002	C2
S005	software_5	19-12-2020	in progress	PI003	C4

Projectleader(3NF):

<u>Projectleader</u> <u>id</u>	Leader_name
-----------------------------------	-------------

project_leader_id	leader_name
PI001	Rahul
PI002	Siddharth
PI003	Krishna

Client(3NF):

<u>Client id</u>	fname	lname	street	city	state
------------------	-------	-------	--------	------	-------

client_id	fname	lname	street	city	State	phonenumber
C1	Karthik	Sharma	street -4	mumbai	Maharastra	9897678955
C2	Ruhani	Singh	street -3	kolkata	west bengal	9645463791
C3	Ravi	Teja	street-11	vizag	Andhra	9090877908
C4	Koushik	Jami	jagadama	vizag	Andhra	9090877708

#### Programmer(3NF):

<u>Programmer_id</u>	Name	specialization	qualification	Date_of_joining	Programming_languages_known
----------------------	------	----------------	---------------	-----------------	-----------------------------

programmer_id	name	qualification	specialization	Date_of_joining	Programming_languages_known_1	Programming_languages_known_2
P001	Rohit	B.Tech	Java	10-05-2019	Java	Python
P002	Raju	M.Sc	C++	11-06-2019	C++	Java
P003	Prasanth	M.Sc	Java	12-05-2019	Python	Java
P004	Aravind	B.tech	Python	13-07-2019	C++	Python
P005	Ramesh	M.sc	C++	14-05-2019	C++	Python
P006	Sundar	B.Tech	C	15-08-2019	Python	C
P007	Kunal	M.Sc	Kotlin	16-07-2019	Python	Kotlin

#### SoftwareSchedule(3NF):

<u>Software_id</u>	<u>Programmer_id</u>	<u>Schedule_id</u>
--------------------	----------------------	--------------------

software_id	programmer_id	schedule_id
S1	P001	SC1
	P002	SC6
S2	P003	SC2
	P004	SC7
S3	P005	SC3
	P006	SC8
	P007	SC9
	P001	SC10
S4	P002	SC4
	P005	SC11
	P006	SC12
S5	P007	SC5
	P003	SC13

Schedule(3NF):

<u>Schedule_id</u>	Start_date	End_date
--------------------	------------	----------

schedule_id ▾	start_date ▾	end_date ▾
SC1	07-09-2019	27-09-2019
SC2	03-01-2020	09-02-2020
SC3	10-04-2020	14-04-2020
SC4	06-04-2020	20-07-2020
SC5	09-08-2020	01-12-2020
SC6	27-09-2019	01-11-2019
SC7	09-02-2020	01-03-2020
SC8	11-04-2020	29-04-2020
SC9	29-04-2020	27-05-2020
SC10	27-05-2020	14-06-2020
SC11	11-07-2020	30-07-2020
SC12	09-08-2020	01-12-2020
SC13	01-10-2020	13-12-2020

#### Differences between the relations obtained from the ER Model and after Normalization :

- Before applying normalization, the tables were defined and realised based on the ER diagram. As a result, there was a possibility for data redundancy and inconsistency. In order to avoid that, we applied normalization on the complete table and hence, the number of tables in the final model is considerably reduced.

- After normalization, we reduced to 6 tables as opposed from the tables obtained via the ER Model - eliminating redundancy by dropping the Recognized relation

**DDL statements (WITH CONSTRAINTS SET WHEREVER NECESSARY) based on this set of normalized relations.**

**Programmer:**

```
create table Programmer
( programmer_id varchar(5),
  name varchar(20) not null,
  qualification varchar(20),
  specialization varchar(20),
  date_of_joining date,
  programming_languages_known1 varchar(10),
  programming_languages_known2 varchar(10),
  primary key (programmer_id)
);
```

**Client:**

```
create table client
( client_id varchar(5),
  fname varchar(15) not null,
  lname varchar(15),
  street varchar(15),
  city varchar(15),
```

```
state varchar(15),  
phonenum numeric(10),  
primary key(client_id)  
);
```

### **Projectleader:**

```
create table projectleader  
( projectleader_id varchar(5),  
name varchar(20),  
primary key(projectleader_id)  
);
```

### **Schedule:**

```
create table schedule (  
schedule_id varchar(5),  
start_date date,  
end_date date,  
primary key(schedule_id));
```

### **Software:**

```
create table software
(software_id varchar(5),
software_name varchar(15),
Date_of_release date,
status_of_software varchar(10),
projectleader_id varchar(5) ,
client_id varchar(5),
primary key(software_id),
foreign key(projectleader_id) references projectleader(projectleader_id) on delete set null,
foreign key(client_id) references client(client_id) on delete set null
);
```

#### **Softwareschedule :**

```
create table softwareschedule
(software_id varchar(5),
programmer_id varchar(5),
schedule_id varchar(5),
primary key(software_id,programmer_id),
foreign key(software_id) references software(software_id)on delete cascade,
foreign key(programmer_id) references programmer(programmer_id) on delete cascade,
foreign key(schedule_id) references schedule(schedule_id) on delete set null
);
```

## **DML STATEMENTS:**

insert into programmer values

```
('P001','Rohit','B.Tech' ,'Java','10/5/2019','Java','Python'),  
( 'P002','Raju','M.Sc','C++','11/6/2019','C++','Java'),  
( 'P003','Prasanth','M.Sc','Java','12/5/2019','Python','Java'),  
( 'P004','Aravind','B.tech','Python','13/7/2019','C++','Python'),  
( 'P005','Ramesh','M.sc','C++','14/5/2019','C++','Python'),  
( 'P006','Sundar','B.Tech','C','15/8/2019','Python','C'),  
( 'P007','Kunal','M.Sc','Kotlin','16/7/2019','Python','Kotlin');
```

insert into client

values

```
('C1','Karthik','Sharma','street -4','mumbai','Maharastra',9897678955),  
( 'C2','Ruhani','Singh','street -3','kolkata','west bengal',9645463791),  
( 'C3','Ravi','Teja','street-11','vizag','Andhra',9090877908),  
( 'C4','Koushik','Jami','jagadama','vizag','Andhra',9090877708);
```

insert into projectleader values

```
('PI001','Rahul'),  
( 'PI002','Siddharth'),
```



('PI003', 'Krishna');

insert into software values

('S001','software\_1','10/11/2019','released','PI001','C1'),

('S002','software\_2','9/3/2020','released','PI002','C3'),

('S003','software\_3','15/6/2020','released','PI003','C1'),

('S004','software\_4','20/8/2020','released','PI002','C2'),

('S005','software\_5','19/12/2020','inprogress','PI003','C4');

insert into Schedule values

('SC1','7/9/2019','27/9/2019'),

('SC2','3/1/2020','9/2/2020'),

('SC3','10/4/2020','14/4/2020'),

('SC4','6/4/2020','20/7/2020'),

('SC5','9/8/2020','1/12/2020'),

('SC6','27/09/2019','01/11/2019'),

('SC7','09/02/2020','01/03/2020'),

('SC8','11/04/2020','29/04/2020'),

('SC9','29/04/2020','27/05/2020'),

('SC10','27/05/2020','14/06/2020'),

('SC11','11/07/2020','30/07/2020'),

('SC12','30/07/2020','14/08/2020'),


('SC13','01/10/2020','13/12/2020');

insert into softwareschedule values

('S001','P001','SC1'),  
('S001','P002','SC6'),  
('S002','P003','SC2'),  
('S002','P004','SC7'),  
('S003','P005','SC3'),  
('S003','P006','SC8'),  
('S003','P007','SC9'),  
('S003','P001','SC10'),  
('S004','P002','SC4'),  
('S004','P005','SC11'),  
('S004','P006','SC12'),  
('S005','P007','SC5'),  
('S005','P003','SC13');

Queries:

1) count the programmers who are specialized in Java.

 project/postgres@PostgreSQL 12

[Query Editor](#) [Query History](#) [Scratch Pad](#)

1

`select count(*) from programmer`

2

`where specialization='Java'`

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

	count bigint	
1	2	

2) list the clients in alphabetically order according to fname.

project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

Scratch Pad

1 select \* from client

2 order by fname

Data Output

Explain

Messages

Notifications

	client_id [PK] character varying (5)	fname character varying (15)	lname character varying (15)	street character varying (15)	city character varying (15)	state character varying (15)	phonenumber numeric (10)
1	C1	Karthik	Sharma	street -4	mumbai	Maharastra	9897678955
2	C4	Koushik	Jami	jagadama	vizag	Andhra	9090877708
3	C3	Ravi	Teja	street-11	vizag	Andhra	9090877908
4	C2	Ruhani	Singh	street -3	kolkata	west bengal	9645463791

3) count number of clients of each state

project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

1

select state,count(\*) from client

2

group by state

Data Output

Explain

Messages

Notifications

	state character varying (15)	count bigint	
1	Maharastra	1	
2	Andhra	2	
3	west bengal	1	

4) list the states having more than one client.

 project/postgres@PostgreSQL 12

Query Editor   Query History   Scratch Pad

1   `select state,count(*) from client`

2   `group by state`

3   `having count(*)>1`

Data Output   Explain   Messages   Notifications

	<div>state</div> <div>character varying (15)</div>	<div>count</div> <div>bigint</div>	
1	Andhra	2	

5) list all software's with their project leaders.

project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

1

select

software\_name

,

name

from

2

software

join

projectleader

on

software.projectleader\_id=projectleader.projectleader\_id

Data Output


Explain

Messages

Notifications

	<div>software_name</div> <div>character varying (15)</div>	<div>name</div> <div>character varying (20)</div>	
1	software_1	Rahul	
2	software_2	Siddharth	
3	software_3	Krishna	
4	software_4	Siddharth	
5	software_5	Krishna	

6) list client whose fname starts with K and lname starts with J

 project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

Scratch Pad

1

select \* from client

2

where fname like 'K%' and lname like 'J%'

Data Output

Explain


Messages

Notifications

	client_id [PK] character varying (5)	fname character varying (15)	lname character varying (15)	street character varying (15)	city character varying (15)	state character varying (15)	phonenumber numeric (10)
1	C4	Koushik	Jami	jagadama	vizag	Andhra	9090877708



7) concat programmer name with specialization

 project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

1


```
select name || ' specialized in ' || specialization from programmer
```

Data Output

Explain

Messages

Notifications

	<div><div>?column?</div><div>text</div></div>	
1	Rohit specialized in Java	
2	Raju specialized in C++	
3	Prasanth specialized in Java	
4	Aravind specialized in Python	
5	Ramesh specialized in C++	
6	Sundar specialized in C	
7	Kunal specialized in Kotlin	

8) Format the start date of schedule in YYYY/Mon/DD as well as end date in YYYY/Mon/DD.

project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

1

select

schedule\_id,to\_char(start\_date,'YYYY/Mon/DD')

as

startdate,to\_char(end\_date,'YYYY/Mon/DD')

2

as

enddate

from

schedule

Data Output

Explain

Messages

Notifications

	<div>schedule_id</div> <div>[PK] character varying (5)</div>	<div>startdate</div> <div>text</div>	<div>enddate</div> <div>text</div>	
1	SC1	2019/Sep/07	2019/Sep/27	<div>✓ Successful</div>
2	SC2	2020/Jan/03	2020/Feb/09	
3	SC3	2020/Apr/10	2020/Apr/14	
4	SC4	2020/Apr/06	2020/Jul/20	
5	SC5	2020/Aug/09	2020/Dec/01	
6	SC6	2019/Sep/27	2019/Nov/01	
7	SC7	2020/Feb/09	2020/Mar/01	
8	SC8	2020/Apr/11	2020/Apr/29	
9	SC9	2020/Apr/29	2020/May/27	
10	SC10	2020/May/27	2020/Jun/14	

9) list all the software's which are released in year 2020

```
1 select * from software
2 where extract('year' from date_of_release)='2020'
```

	<div>software_id</div> <div>[PK] character varying (5)</div>	<div>software_name</div> <div>character varying (15)</div>	<div>date_of_release</div> <div>date</div>	<div>status_of_software</div> <div>character varying (10)</div>	<div>projectleader_id</div> <div>character varying (5)</div>	<div>client_id</div> <div>character varying (5)</div>
1	S002	software_2	2020-03-09	released	PI002	C3
2	S003	software_3	2020-06-15	released	PI003	C1
3	S004	software_4	2020-08-20	released	PI002	C2
4	S005	software_5	2020-12-19	inprogress	PI003	C4

10) list the software's between 1-1-2019 and 30-06-2020

project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

Scratch Pad

1

select \* from software

2

where date\_of\_release between '1-1-2019' and '30-08-2020'

Data Output

Explain

Messages

Notifications

	software_id [PK] character varying (5)	software_name character varying (15)	date_of_release date	status_of_software character varying (10)	projectleader_id character varying (5)	client_id character varying (5)
1	S001	software_1	2019-11-10	released	PI001	C1
2	S002	software_2	2020-03-09	released	PI002	C3
3	S003	software_3	2020-06-15	released	PI003	C1
4	S004	software_4	2020-08-20	released	PI002	C2

11) find programmer name whose name starts with 'R' or ending with 'h' or both



project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad


```
1 select name from programmer
2 where name like 'R%'
3 union
4 select name from programmer
5 where name like '%h'
```

Data Output

Explain

Messages

Notifications

	<b>name</b> character varying (20) 	
1	Rohit	
2	Ramesh	
3	Prasanth	
4	Raju	

12) list programmer name, specialization, date\_of\_joining whose programmer id is 'P001','P002','P003' .

project/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

1

select name,specialization,date\_of\_joining from programmer

2

where programmer\_id in (select programmer\_id from programmer

3

where programmer\_id in ('P001','P002','P003') );

Data Output

Explain

Messages

Notifications

	name character varying (20)	specialization character varying (20)	date_of_joining date	
1	Rohit	Java	2019-05-10	
2	Raju	C++	2019-06-11	
3	Prasanth	Java	2019-05-12	

- T.Raviteja (18354)
- S.M.Mustaq Ahammad(18350)
- Jami Koushik (18363)
- P.V.Dinesh(18341)
- R.Manoj(18344)