



राष्ट्रीय प्रौद्योगिकी संस्थान दिल्ली
National Institute of Technology Delhi
(An autonomous Institute under the aegis of Ministry of HRD, Govt. of India)

PROJECT TITLE

COMPLAINT MANAGEMENT SYSTEM

GROUP MEMBERS: -

1. Dikshya Goyal (181210021)
2. Ashish Aggarwal (181210012)
3. Ankit Kr Rouniyar (181210011)
4. Dimpal Kataniya (181210022)

CONTENTS

1. System Description at a Glance
2. Case Study
3. Requirement Analysis
4. Hardware and Software Requirements
5. ER Diagram
6. Mapping from ER Model to Relational Model
7. Data Dictionary
8. SQL Queries
 - Raw Code Of Tables
 - ☐ Relational Algebra Operations (Basic)
 - ☐ Relational Algebra queries
 - ☐ Advanced SQL
9. Front End Functionality
 - Output
10. Bibliography

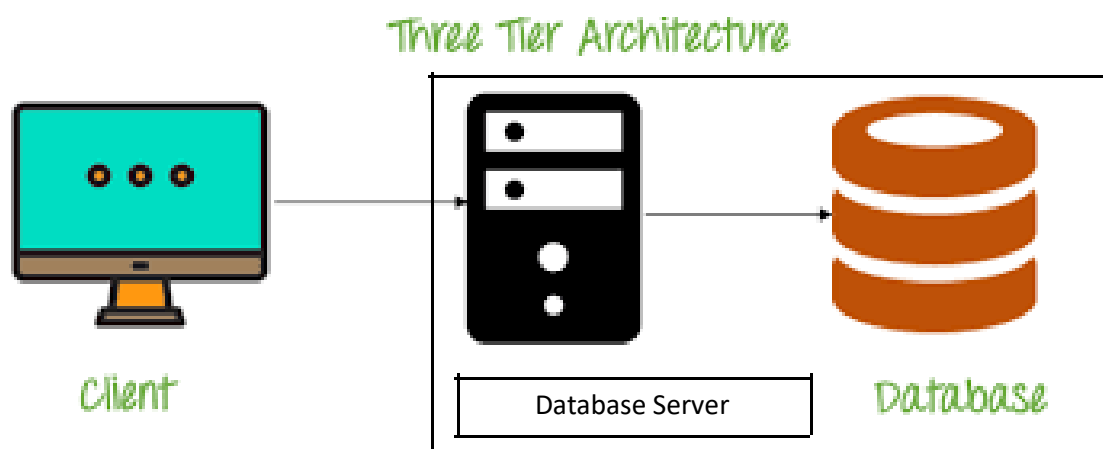
SYSTEM DESCRIPTION AT A GLANCE

The aim of this project is to make a computer generated web-based program with the blend of database management system and PHP. It can manage the details of the students, faculty and administrator members who can register themselves and all of this being supervised by the admin who have all sorts of rights to manipulate data.

This system can help the user to file a complaint and get feedback.

All of this is done with the help of the front-end and back-end with the xampp as a server. We've implemented a three-tier client-server architecture, where the business layer (middle layer) is implemented through PHP.

We've used PHPMyAdmin/MS SQL to store database using MySQL.



CASE STUDY

Scenario: A relational database is to be designed for a complaint management system for a college. The complaint will be registered by the complainant. The complainant can be a faculty or a student or someone from administrative department. The type of complaint will be given as well. The complaint recipient will receive the complaint and the information regarding the complaint being solved or not will be given. Later on the feedback will be taken both by the complainant and the complaint recipient. Other than this, every complaint will be registered with a complaint ID so that if a person complains more than once the complaints can be distinguished

- 1.) Every complainant will be either a student or a faculty or someone from administrative department.
- 2.) Every complaint will have a unique ID different from roll no. and registration no. of students and faculty respectively.
- 3.) Problems will have types which will appear in a checkbox to choose from.
- 4.) All the information regarding students and faculty will be provided.

Objective: The purpose of making this project is to get rid from the manual entry and record system and try to simplify and ease the complaining process by implementation of the Database Management System.

Functionalities: The project is designed to keep the record of the record of the complaint regarding any issues faced by any person in the institution and forward to its respective department where it can be solved.

HARDWARE REQUIREMENTS

Assuming that a typical system offers hundreds of entries. The volume of the information to be handled is thus about millions of characters. Further the whole information has to be processed. All this suggest that the minimum hardware requirements should be: -

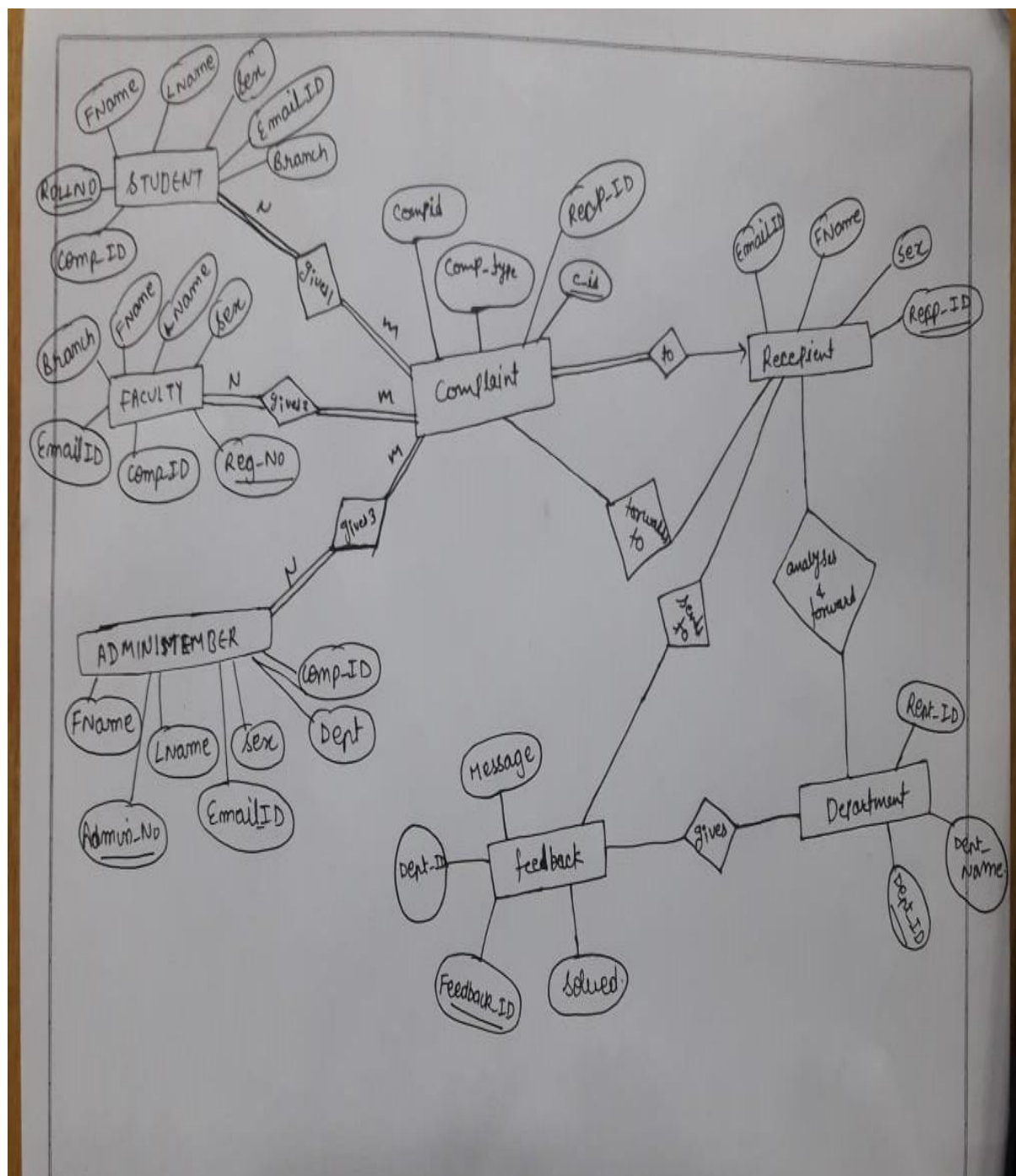
- ❑ Operating **System: Windows** 7/8/8.1/10.
- ❑ Memory (RAM): 1 GB of RAM required.
- ❑ Hard Disk Space: 200 MB of free space required.
- ❑ Processor: Intel Pentium 4 or later.
- ❑ Cache: 512KB

SOFTWARE REQUIRMENTS

In the development of a project the selection of an appropriate DBMS Software and a platform is of primary importance. With many software options available a developer has to consider the various features and functionalities and ease of handling the software, keeping an account of such things we decided to use Bootstrap Studio for designing the front-end, the front-end has been developed by the use of HTML, CSS and JS. MySQL has been used as a back-end query language. PHP has been chosen as a scripting language. The server chosen is the localhost which would be hosting the website on the machine itself. The following software are required: -

- ❑ Web Browser (Chrome/Edge/Firefox,etc.)
- ❑ XAMPP/WAMPP or any other web server which can host
- ❑ MySQL

ER DIAGRAM



Mapping ER Model to R Model

RELATIONAL MODEL

STUDENT

FName	LName	Sex	EmailID	Comp_Id	<u>RollNo</u>	Branch
-------	-------	-----	---------	---------	---------------	--------

FACULTY

FName	LName	Sex	EmailID	Comp ID	<u>Reg_No</u>	Branch
-------	-------	-----	---------	---------	---------------	--------

ADMINISTRATIVE MEMBER

FName	LName	Sex	EmailID	Comp ID	<u>Admin No</u>	Dept
-------	-------	-----	---------	---------	-----------------	------

GIVES1

Comp_ID	c_id	RollNo
---------	------	--------

GIVES2

Comp_ID	<u>RegNo</u>	c_id
---------	--------------	------

GIVES3

Comp_ID	<u>AdminNo</u>	c_id
---------	----------------	------

COMPLAINT

<u>c_id</u>	Comp ID	Comp Type	Recp ID
-------------	---------	-----------	---------

RECEPIENT

Email_ID	FName	Sex	<u>Recp ID</u>
----------	-------	-----	----------------

DEPARTMENT

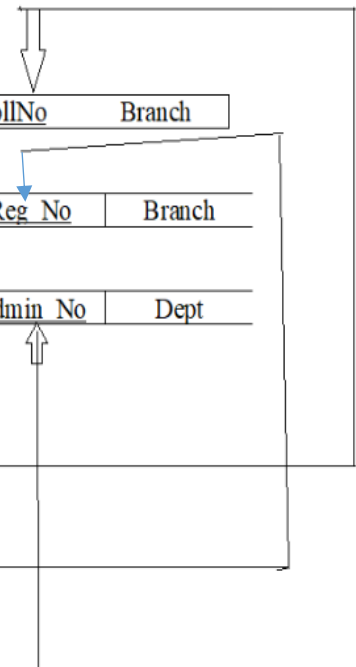
Recp ID	Dept_Name	<u>Dept_ID</u>
---------	-----------	----------------

FEEDBACK

Message	Solved	Feedback_ID	Dept_ID
---------	--------	-------------	---------

LOGIN

<u>Comp_Id</u>	Password
----------------	----------



Data Dictionary

Faculty:

min/tbl_structure.php?db=cms&table=faculty

Server: 127.0.0.1 • Database: cms • Table: faculty

Browse Structure SQL Search Insert Export Import Privileges Operation

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	FNAME	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
2	LNAME	varchar(30)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	EMAILID	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	SEX	varchar(2)	utf8mb4_general_ci		No	None			Change Drop More
5	BRANCH	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
6	REGNO	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
7	COMPID	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More

☐ Check all ☐ With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

Add 1 column(s) after COMPID [Go](#)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	REGNO 7	A		No	

Department :

min/tbl_structure.php?db=cms&table=department

Server: 127.0.0.1 • Database: cms • Table: department

Browse Structure SQL Search Insert Export Import Privileges Operation

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	DEPT_ID	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
2	RECP_ID	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	DEPT_NAME	char(30)	utf8mb4_general_ci		No	None			Change Drop More

☐ Check all ☐ With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

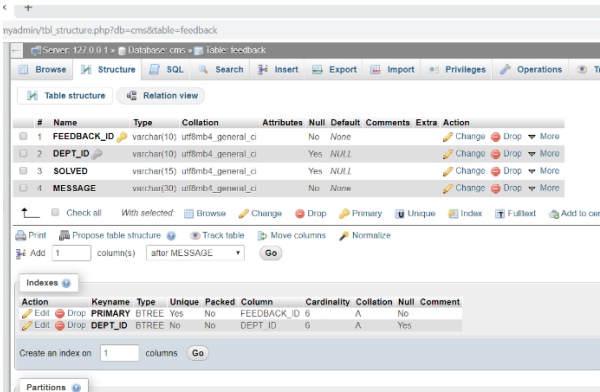
Add 1 column(s) after DEPT_NAME [Go](#)

Indexes

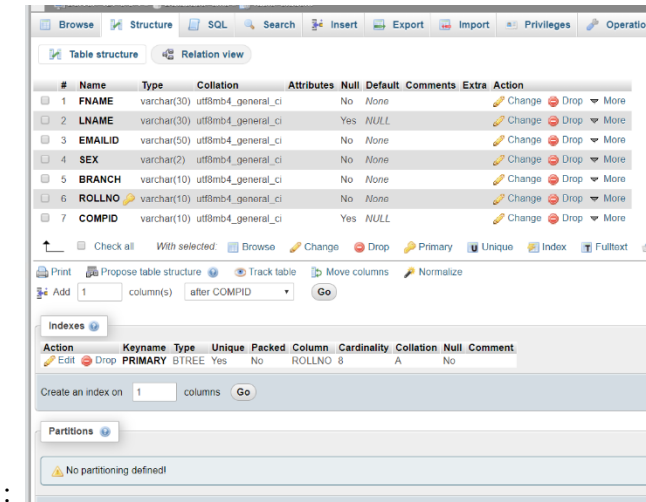
Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	DEPT_ID 6	A		No	
Edit Drop	RECP_ID	BTREE	No	No	RECP_ID 2	A		Yes	

Create an index on 1 columns [Go](#)

Feedback:



Login



Gives 3:

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	compid	varchar(10)	utf8mb4_general_ci		Yes	NULL			
2	ADMINNO	varchar(10)	utf8mb4_general_ci		No	None			
3	C_ID	varchar(10)	utf8mb4_general_ci		No	None			

☐ Check all

With selected:

Print

Propose table structure

Track table

Move columns

Normalize

Add

1

column(s)

after C_ID

Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
	PRIMARY	BTREE	Yes	No	ADMINNO	6	A	No	
	C_ID	BTREE	No	No	C_ID	6	A	No	
	compid	BTREE	No	No	compid	6	A	Yes	

Create an index on 1 columns Go

Partitions

No partitioning defined!

Gives 2:

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Opera

Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	compid	varchar(10)	utf8mb4_general_ci		Yes	NULL			
2	REGNO	varchar(10)	utf8mb4_general_ci		No	None			
3	C_ID	varchar(10)	utf8mb4_general_ci		No	None			

☐ Check all

With selected:

Print

Propose table structure

Track table

Move columns

Normalize

Add

1

column(s)

after C_ID

Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
	PRIMARY	BTREE	Yes	No	REGNO	7	A	No	
	C_ID	BTREE	No	No	C_ID	7	A	No	
	compid	BTREE	No	No	compid	7	A	Yes	

Create an index on 1 columns Go

Gives 1:

Browse Structure SQL Search Insert Export Import Privileges Operations											
Table structure Relation view											
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	ROLLNO	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More		
2	C_ID	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More		
3	COMPID	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More		
<input type="checkbox"/> Check all With selected: Browse Change Drop Primary Unique Index Fulltext											
Print Propose table structure Track table Move columns Normalize											
Add 1 column(s) after COMPID Go											
Indexes											
action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment		
Edit Drop	PRIMARY	BTREE	Yes	No	ROLLNO	8	A	No			
Edit Drop	C_ID	BTREE	No	No	C_ID	8	A	No			
Edit Drop	COMPID	BTREE	No	No	COMPID	8	A	Yes			

Recipient :

Browse Structure SQL Search Insert Export Import Privileges Operations											
Table structure Relation view											
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	EMAILID	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More		
2	SEX	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More		
3	FNAME	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More		
4	RECP_ID	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More		
<input type="checkbox"/> Check all With selected: Browse Change Drop Primary Unique Index Fulltext											
Print Propose table structure Track table Move columns Normalize											
Add 1 column(s) after RECP_ID Go											
Indexes											
action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment		
Edit Drop	PRIMARY	BTREE	Yes	No	RECP_ID	4	A	No			
Create an index on 1 columns Go											

Complaint

om x +

/phpmyadmin/tbl_structure.php?db=cms&table=complaint

Server: 127.0.0.1 » Database: cms » Table: complaint

Browse Structure SQL Search Insert Export Import Privileges Operations

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	COMP_TYPE	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
2	C_ID	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
3	compid	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	RECP_ID	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after RECP_ID Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	C_ID	21	A	No	
Edit Drop	RECP_ID	BTREE	No	No	RECP_ID	2	A	Yes	

Create an index on 1 columns Go

Admin member:-

admin/tbl_structure.php?db=cms&table=admin_member

Server: 127.0.0.1 » Database: cms » Table: admin_member

Browse Structure SQL Search Insert Export Import Privileges Operations

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	FNAME	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
2	LNAME	varchar(30)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	EMAILID	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	SEX	varchar(2)	utf8mb4_general_ci		No	None			Change Drop More
5	DEPT	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
6	ADMINNO	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
7	COMPID	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Populated database: -

Student:

	FNAME	LNAME	EMAILID	SEX	BRANCH	ROLL...	COMPID	city
1	Ram	Kumar	ramkumar@gmail.com	M	CSE	1001	6001	NULL
2	ravi	Kumar	ravikumar@gmail.com	M	CSE	1002	6002	NULL
3	ankit	Kumar	ankitkumar@gmail.com	M	CSE	1003	6003	NULL
4	hari	prasad	rrrhari@gmail.com	M	CSE	1004	6004	NULL
5	preet	Kumar	preetkumar@gmail.com	M	ECE	1006	6005	NULL
6	aashish	agarwal	agarwal121@gmail.com	M	CSE	1007	6006	NULL
7	dikshya	goyal	goyalddd@gmail.com	M	CSE	1008	6007	NULL
8	preeti	Kumari	preetikumari@gmail.com	f	ece	1009	6008	NULL

Faculty:

	FNAME	LNAME	EMAILID	SEX	BRANCH	REG...	COMPID
1	jeran	deep	jideep@gmail.com	M	ece	9980	6009
2	money	monga	mjmon@gmail.com	M	civil	9981	6010
3	abhishek	patel	patelabhishek@gmail.com	M	ece	9982	6011
4	hيرانanjans	singh	hiroksjht@gmail.com	M	cse	9983	6012
5	kiran	verma	kiran@gmail.com	M	EEE	9984	6013
6	rakul	preet	dsrama@gmail.com	f	EEE	9985	6014
7	Dr manoj	yadav	myadaav@gmail.com	M	CSE	9986	6015

Query executed successfully.

Admin member:

	FNAME	LNAME	EMAILID	SEX	DEPT	ADMINNO	COMPID
1	Dr manoj	yadav	myadaav@gmail.com	M	CSE	3320	6016
2	Dr manav	yadav	mnydaav@gmail.com	M	CSE	3321	6017
3	Radhikakumari	yadav	rkumariyadav@gmail.com	f	ECE	3322	6018
4	anuska	gambhir	kkrgmbhir@gmail.com	f	CSE	3323	6019
5	anuska	gambhir	kkrgmbhir@gmail.com	f	CSE	3324	6020
6	ranjan	meena	rammm@gmail.com	M	EEE	3325	6021
7	aman	malhotra	manmalhotr@gmail.com	M	EEE	3326	6022

Gives 1:

	ROLL...	C_ID	COMPID
1	1001	111	6001
2	1002	112	6002
3	1003	113	6003
4	1004	114	6004
5	1006	116	6006
6	1007	117	6007
7	1008	118	6008

Query executed successfully.

Gives 2:

	compid	REG...	C_ID
1	6009	9980	119
2	6010	9981	120
3	6011	9982	121
4	6012	9983	122
5	6013	9984	123
6	6014	9985	124
7	6015	9986	125

Query executed successfully.

Gives 3:

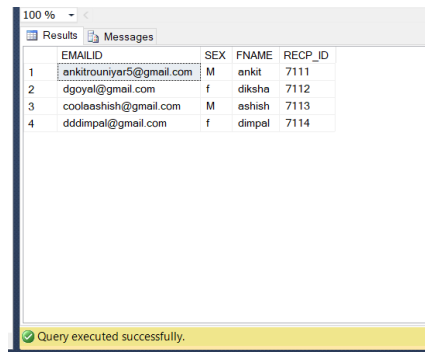
	compid	ADMINNO	C_ID
1	6017	3321	127
2	6018	3322	128
3	6019	3323	129
4	6020	3324	130
5	6021	3325	131
6	6022	3326	132

Complaint:

	COMP_TYPE	C_ID	com...	RECP_ID
7	related to labs	117	6007	7111
8	related to labs	118	6008	7111
9	related to infra	119	6009	7111
10	related to massbunks	120	6010	7111
11	related to fest	121	6011	7111
12	related to labs	122	6012	7111
13	related to fest	123	6013	7111
14	related to fest	124	6014	7111
15	related to schedule	125	6015	7111
16	related to schedule	127	6017	7111
17	related to fee	128	6018	7111
18	related to forms	129	6019	7111
19	related to late fee	130	6020	7111
20	related to forms	131	6021	7111
21	related to late fee	132	6022	7111

Query executed successfully.


Recipient: -



	EMAILID	SEX	FNAME	RECP_ID
1	ankitrouniyar5@gmail.com	M	ankit	7111
2	dgoyal@gmail.com	f	diksha	7112
3	coolaashish@gmail.com	M	ashish	7113
4	dddimpal@gmail.com	f	dimpal	7114

Query executed successfully.

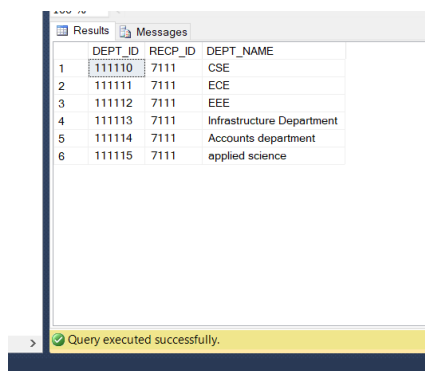
Feedback: -

Results  Messages

	FEEDBACK_ID	DEPT...	SOLV...	MESSAGE
f101	111110	yes	This has been resolved	
f102	111111	yes	This has been resolved	
f103	111112	yes	This has been resolved	
f104	111113	no	This ill take some time	
f105	111114	yes	This has been resolved	
f106	111115	yes	This has been resolved	

Query executed successfully.

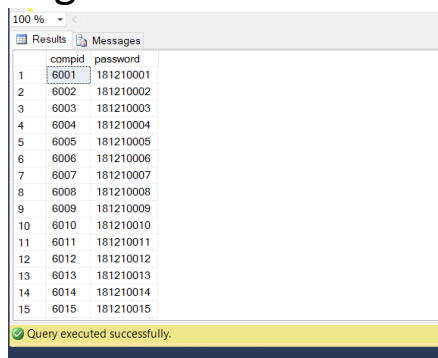
Department:-



	DEPT_ID	RECP_ID	DEPT_NAME
1	111110	7111	CSE
2	111111	7111	ECE
3	111112	7111	EEE
4	111113	7111	Infrastructure Department
5	111114	7111	Accounts department
6	111115	7111	applied science

Query executed successfully.

Login:-



	compid	password
1	6001	181210001
2	6002	181210002
3	6003	181210003
4	6004	181210004
5	6005	181210005
6	6006	181210006
7	6007	181210007
8	6008	181210008
9	6009	181210009
10	6010	181210010
11	6011	181210011
12	6012	181210012
13	6013	181210013
14	6014	181210014
15	6015	181210015

Query executed successfully.

SQL QUERIES

Queries to Create the relations and populating the database:

CREATE TABLE COMMAND

```
CREATE TABLE STUDENT (  
FNAME VARCHAR (30) NOT NULL,  
LNAME VARCHAR (30),  
EMAILID VARCHAR (50) NOT NULL,  
SEX VARCHAR (2) NOT NULL,  
BRANCH VARCHAR (10) NOT NULL,  
ROLLNO VARCHAR (10) NOT NULL,  
COMPID VARCHAR (10) ,  
PRIMARY KEY (rollno));
```

INSERTION COMMANDS

```
INSERT INTO STUDENT VALUES ('Ram','Kumar','ramkumar@gmail.com','M','CSE','1001','6001');  
INSERT INTO STUDENT VALUES ('ravi','Kumar','ravikumar@gmail.com','M','CSE','1002','6002');  
INSERT INTO STUDENT VALUES ('ankit','Kumar','ankitkumar@gmail.com','M','CSE','1003','6003');  
INSERT INTO STUDENT VALUES ('hari','prasad','rrrrhari@gmail.com','M','CSE','1004','6004');  
INSERT INTO STUDENT VALUES ('preet','Kumar','preetkumar@gmail.com','M','ECE','1006','6005');  
INSERT INTO STUDENT VALUES ('aashish','agarawl','agarwala121@gmail.com','M','CSE','1007','6006');  
INSERT INTO STUDENT VALUES ('dikshya','goyal','goyalddd8@gmail.com','M','CSE','1008','6007');  
INSERT INTO STUDENT VALUES ('preeti','Kumari','preetikumari@gmail.com','f','ece','1009','6008');  
select * from student;
```

CREATE TABLE COMMAND

```
CREATE TABLE FACULTY (  
FNAME VARCHAR (30) NOT NULL,  
LNAME VARCHAR (30),  
EMAILID VARCHAR (50) NOT NULL,  
SEX VARCHAR (2) NOT NULL,  
BRANCH VARCHAR (10) NOT NULL,  
REGNO VARCHAR (10) NOT NULL,  
COMPID VARCHAR (10) ,  
PRIMARY KEY (regno));
```

INSERTION COMMANDS

```
INSERT INTO FACULTY VALUES ('jeren','deep','jjdeep@gmail.com','M','ece','9980','6009');  
INSERT INTO FACULTY VALUES ('money','monga','mjmon@gmail.com','M','civil','9981','6010');  
INSERT INTO FACULTY VALUES ('abhishek','patel','patelabhishek@gmail.com','M','ece','9982','6011');  
INSERT INTO FACULTY VALUES ('hيرانjans','singh','hhiroksjih@gmail.com','M','cse','9983','6012');  
INSERT INTO FACULTY VALUES ('kiran','verma','kiran@gmail.com','M','EEE','9984','6013');  
INSERT INTO FACULTY VALUES ('rakul','preet','dsrama@gmail.com','f','EEE','9985','6014');  
INSERT INTO FACULTY VALUES ('Dr manoj','yadav','myadaav@gmail.com','M','CSE','9986','6015');  
select * from faculty;
```

CREATE TABLE COMMAND

```
CREATE TABLE ADMIN_MEMBER (  
FNAME VARCHAR (30) NOT NULL,  
LNAME VARCHAR (30),  
EMAILID VARCHAR (50) NOT NULL,
```

SEX VARCHAR (2) NOT NULL,
DEPT VARCHAR (10) NOT NULL,
ADMINNO VARCHAR (10) NOT NULL,
COMPID VARCHAR (10) ,
PRIMARY KEY (adminno));

INSERTION COMMANDS

```
INSERT INTO ADMIN_MEMBER VALUES ('Dr  
manoj','yadav','myadaav@gmail.com','M','CSE','3320','6016');  
INSERT INTO ADMIN_MEMBER VALUES ('Dr  
manav','yadav','mnydaav@gmail.com','M','CSE','3321','6017');  
INSERT INTO ADMIN_MEMBER VALUES  
('Radhikakumari','yadav','rkumariyadav@gmail.com','f','ECE','3322','6018');  
INSERT INTO ADMIN_MEMBER VALUES  
('anuska','gambhir','kkrgrmbhir@gmail.com','f','CSE','3323','6019');  
INSERT INTO ADMIN_MEMBER VALUES  
('anuska','gambhir','kkrgrmbhir@gmail.com','f','CSE','3324','6020');  
INSERT INTO ADMIN_MEMBER VALUES  
('ranjan','meena','rammr@gmail.com','M','EEE','3325','6021');  
INSERT INTO ADMIN_MEMBER VALUES  
('aman','malhotra','manmalhotr@gmail.com','M','EEE','3326','6022');  
select * from ADMIN_MEMBER;
```

CREATE TABLE COMMAND

```
CREATE TABLE gives1(  
ROLLNO VARCHAR (10) NOT NULL,  
C_ID VARCHAR (10),  
COMPID VARCHAR(10),  
PRIMARY KEY (ROLLNO,C_ID),  
foreign key (rollno) references student (rollno),  
foreign key (C_ID) references COMPLAINT (C_ID),  
foreign key (compid) references login_info (compid));
```

INSERTION COMMANDS

```
insert into gives1 values ('1001','111','6001');  
insert into gives1 values ('1002','112','6002');  
insert into gives1 values ('1003','113','6003');  
insert into gives1 values ('1004','114','6004');  
insert into gives1 values ('1005','115','6005');  
insert into gives1 values ('1006','116','6006');  
insert into gives1 values ('1007','117','6007');  
insert into gives1 values ('1008','118','6008');  
select * from gives1;
```

CREATE TABLE COMMAND

```
CREATE TABLE gives2(  
compid varchar (10),  
REGNO VARCHAR (10) NOT NULL,  
C_ID VARCHAR (10),  
PRIMARY KEY (REGNO,C_ID),  
foreign key (regno) references faculty (regno),  
foreign key (C_ID) references COMPLAINT (C_ID),  
foreign key (compid) references login_info (compid));
```

INSERTION COMMANDS

```
insert into gives2 values ('6009','9980','119');  
insert into gives2 values ('6010','9981','120');  
insert into gives2 values ('6011','9982','121');  
insert into gives2 values ('6012','9983','122');
```

```
insert into gives2 values ('6013','9984','123');
insert into gives2 values ('6014','9985','124');
insert into gives2 values ('6015','9986','125');
select * from gives2;
```

CREATE TABLE COMMAND

```
CREATE TABLE gives3(
compid varchar (10),
ADMINNO VARCHAR (10) NOT NULL,
C_ID VARCHAR (10),
PRIMARY KEY (ADMINNO,C_ID),
foreign key (adminno) references admin_member (adminno),
foreign key (C_ID) references COMPLAINT (C_ID),
foreign key (compid) references login_info (compid));
```

INSERTION COMMANDS

```
insert into gives3 values ('6017','3321','127');
insert into gives3 values ('6018','3322','128');
insert into gives3 values ('6019','3323','129');
insert into gives3 values ('6020','3324','130');
insert into gives3 values ('6021','3325','131');
insert into gives3 values ('6022','3326','132');
select * from gives3;
```

CREATE TABLE COMMAND

```
CREATE TABLE recipient (
EMAILID VARCHAR (50) NOT NULL,
SEX VARCHAR (10),
FNAME VARCHAR (30) NOT NULL,
RECP_ID VARCHAR (10),
PRIMARY KEY (RECP_ID),);
```

INSERTION COMMANDS

```
insert into recipient values ('ankitrouniyar5@gmail.com','M','ankit','7111');
insert into recipient values ('dgoyal@gmail.com','f','diksha','7112');
insert into recipient values ('coolaashish@gmail.com','M','ashish','7113');
insert into recipient values ('dddimpal@gmail.com','f','dimpal','7114');
select * from recipient;
```

CREATE TABLE COMMAND

```
CREATE TABLE COMPLAINT (
COMP_TYPE VARCHAR(20) NOT NULL,
C_ID VARCHAR (10),
compid VARCHAR (10),
RECP_ID VARCHAR (10),
PRIMARY KEY (C_ID),
FOREIGN KEY (RECP_ID) REFERENCES recipient(recp_id)
);
```

INSERTION COMMANDS

```
insert into complaint values ('related to labs','111','6001','7111');
insert into complaint values ('related to classroom','112','6002','7111');
insert into complaint values ('related to library','113','6003','7111');
insert into complaint values ('related to labs','114','6004','7111');
insert into complaint values ('related to buses','115','6005','7111');
insert into complaint values ('related to classroom','116','6006','7111');
insert into complaint values ('related to labs','117','6007','7111');
insert into complaint values ('related to labs','118','6008','7111');
insert into complaint values ('related to infra','119','6009','7111');
```

```

insert into complaint values ('related to massbunks','120','6010','7111');
insert into complaint values ('related to fest','121','6011','7111');
insert into complaint values ('related to labs','122','6012','7111');
insert into complaint values ('related to fest','123','6013','7111');
insert into complaint values ('related to fest','124','6014','7111');
insert into complaint values ('related to schedule','125','6015','7111');
insert into complaint values ('related to schedule','127','6017','7111');
insert into complaint values ('related to fee','128','6018','7111');
insert into complaint values ('related to forms','129','6019','7111');
insert into complaint values ('related to late fee','130','6020','7111');
insert into complaint values ('related to forms','131','6021','7111');
insert into complaint values ('related to late fee','132','6022','7111');
select * from complaint;

```

CREATE TABLE COMMAND

```

CREATE TABLE department (
DEPT_ID VARCHAR (10) NOT NULL,
RECP_ID VARCHAR (10) ,
DEPT_NAME CHAR (30) NOT NULL,
PRIMARY KEY (DEPT_ID),
FOREIGN KEY (RECP_ID) REFERENCES recipient (RECP_ID));

```

INSERTION COMMANDS

```

insert into department values ('111110','7111','CSE');
insert into department values ('111111','7111','ECE');
insert into department values ('111112','7111','EEE');
insert into department values ('111113','7111','Infrastructure Department');
insert into department values ('111114','7111','Accounts department');
insert into department values ('111115','7111','applied science');
select * from department;

```

CREATE TABLE COMMAND

```

CREATE TABLE FEEDBACK (
FEEDBACK_ID VARCHAR (10) NOT NULL,
DEPT_ID VARCHAR (10) ,
SOLVED VARCHAR (15) ,
MESSAGE VARCHAR (30) NOT NULL,
PRIMARY KEY (FEEDBACK_ID),
FOREIGN KEY (DEPT_ID) REFERENCES DEPARTMENT(DEPT_ID));

```

INSERTION COMMANDS

```

insert into feedback values ('f101','111110','yes','This has been resolved');
insert into feedback values ('f102','111111','yes','This has been resolved');
insert into feedback values ('f103','111112','yes','This has been resolved');
insert into feedback values ('f104','111113','no','This ill take some time');
insert into feedback values ('f105','111114','yes','This has been resolved');
insert into feedback values ('f106','111115','yes','This has been resolved');
select * from feedback;

```

CREATE TABLE COMMAND

```

create table login_info (
compid varchar (10),
password varchar (20) not null,
primary key(compid));

```

INSERTION COMMANDS

```

insert into login_info values ('6001','181210001');
insert into login_info values ('6002','181210002');
insert into login_info values ('6003','181210003');

```

```
insert into login_info values ('6004','181210004');
insert into login_info values ('6005','181210005');
insert into login_info values ('6006','181210006');
insert into login_info values ('6007','181210007');
insert into login_info values ('6008','181210008');
insert into login_info values ('6009','181210009');
insert into login_info values ('6010','181210010');
insert into login_info values ('6011','181210011');
insert into login_info values ('6012','181210012');
insert into login_info values ('6013','181210013');
insert into login_info values ('6014','181210014');
insert into login_info values ('6015','181210015');
insert into login_info values ('6016','181210016');
insert into login_info values ('6017','181210017');
insert into login_info values ('6018','181210018');
insert into login_info values ('6019','181210019');
insert into login_info values ('6020','181210020');
insert into login_info values ('6021','181210021');
insert into login_info values ('6022','181210022');
select * from login_info;
```

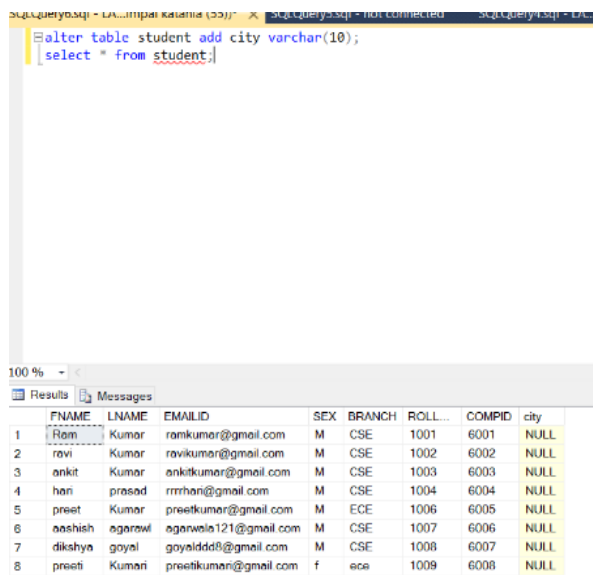
QUERIES

DDL COMMANDS

Q1. Add column “city “to student table and then drop it.

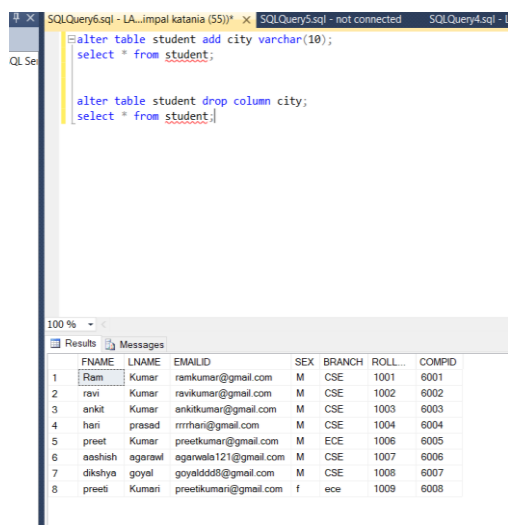
Ans 1. Alter table student add city varchar(10);

Alter table student drop column city;



```
alter table student add city varchar(10);
select * from student;
```

	FNAME	LNAME	EMAILID	SEX	BRANCH	ROLL...	COMPID	city
1	Ram	Kumar	ramkumar@gmail.com	M	CSE	1001	6001	NULL
2	ravi	Kumar	ravikumar@gmail.com	M	CSE	1002	6002	NULL
3	ankit	Kumar	ankitkumar@gmail.com	M	CSE	1003	6003	NULL
4	hari	prasad	rrrhan@gmail.com	M	CSE	1004	6004	NULL
5	preet	Kumar	preetkumar@gmail.com	M	ECE	1006	6005	NULL
6	aashish	agarwal	agarwala121@gmail.com	M	CSE	1007	6006	NULL
7	dikshya	goyal	goyalddd@gmail.com	M	CSE	1008	6007	NULL
8	preeti	Kumari	preetikumar@gmail.com	f	ece	1009	6008	NULL



```
alter table student add city varchar(10);
select * from student;

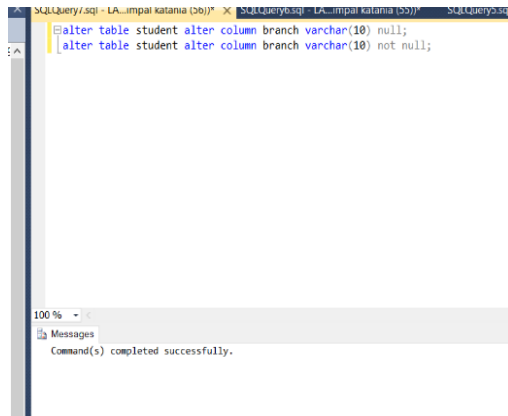
alter table student drop column city;
select * from student;
```

	FNAME	LNAME	EMAILID	SEX	BRANCH	ROLL...	COMPID
1	Ram	Kumar	ramkumar@gmail.com	M	CSE	1001	6001
2	ravi	Kumar	ravikumar@gmail.com	M	CSE	1002	6002
3	ankit	Kumar	ankitkumar@gmail.com	M	CSE	1003	6003
4	hari	prasad	rrrhan@gmail.com	M	CSE	1004	6004
5	preet	Kumar	preetkumar@gmail.com	M	ECE	1006	6005
6	aashish	agarwal	agarwala121@gmail.com	M	CSE	1007	6006
7	dikshya	goyal	goyalddd@gmail.com	M	CSE	1008	6007
8	preeti	Kumari	preetikumar@gmail.com	f	ece	1009	6008

Q2. Remove not null constraint from branch attribute of student table and add it again

Ans 2.a) Alter table student alter column branch varchar(10) null;

b) Alter table student alter column branch varchar(10) not null;



```
SQLQuery.sql - LA...mpai katania (2b) * x SQLQuery.sql - LA...mpai katania (2b) * SQLQuery.sql
alter table student alter column branch varchar(10) null;
alter table student alter column branch varchar(10) not null;
```

100 %
Messages
Command(s) completed successfully.

SET UPDATE COMMAND

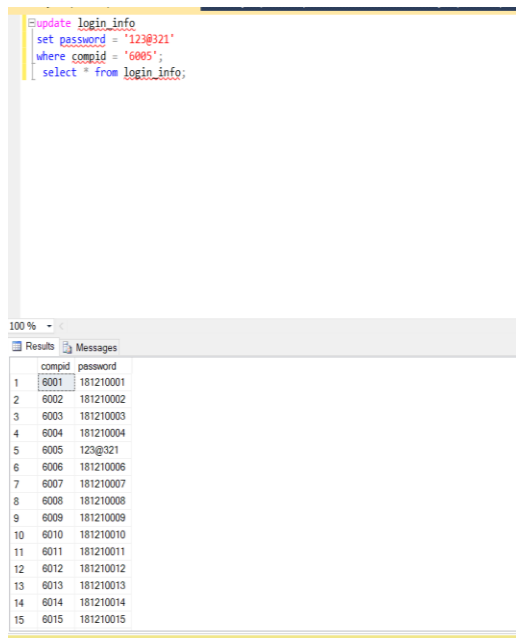
Q1. Change the password to “123@321” of the complainants having comp_id “6005”

Ans :Update login_info

Set password = '123@321'

Where compid == '6005'

Select * from login_info;



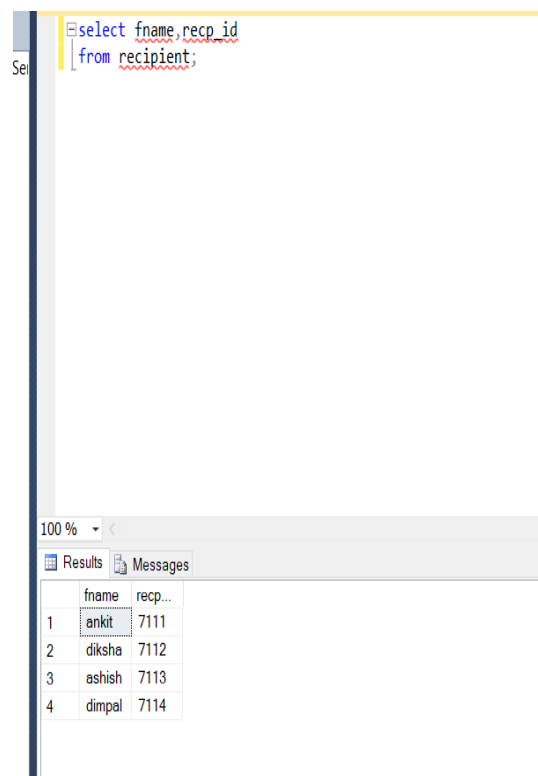
```
update login_info
set password = '123@321'
where compid = '6005';
select * from login_info;
```

	compid	password
1	6001	181210001
2	6002	181210002
3	6003	181210003
4	6004	181210004
5	6005	123@321
6	6006	181210006
7	6007	181210007
8	6008	181210008
9	6009	181210009
10	6010	181210010
11	6011	181210011
12	6012	181210012
13	6013	181210013
14	6014	181210014
15	6015	181210015

BASIC RELATIONAL ALGEBRA QUERIES

Q1. List the name of all the recipients and their recp_id.

Ans 1. Select fname,recp_id
From recipient;



The screenshot shows a database query interface. At the top, a text area contains the SQL query: `select fname, recp_id from recipient;`. Below the text area, there is a tabbed interface with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'fname' and 'recp...'. The table contains four rows of data, numbered 1 through 4.

	fname	recp...
1	ankit	7111
2	diksha	7112
3	ashish	7113
4	dimpal	7114

Q2. List the name of students whose sex is 'm' and they are from cse branch.

Ans 2. Select Fname,Lname

From student

Where sex='m' and branch ='cse';

```
select fname, lname
from student
where sex = 'm' and branch = 'cse';
```

100 %

Results Messages

	fname	lname
1	Ram	Kumar
2	ravi	Kumar
3	ankit	Kumar
4	hari	prasad
5	aashish	agarwal
6	dikshya	goyal

Q3. retrieve the name of faculties who have name contains “a” and they are from eee branch.

Ans :- select fname

From faculty

Where branch = 'eee' and fname like '%a%';

```
select fname
from faculty
where branch = 'eee' and fname like '%a%';
```

100 %

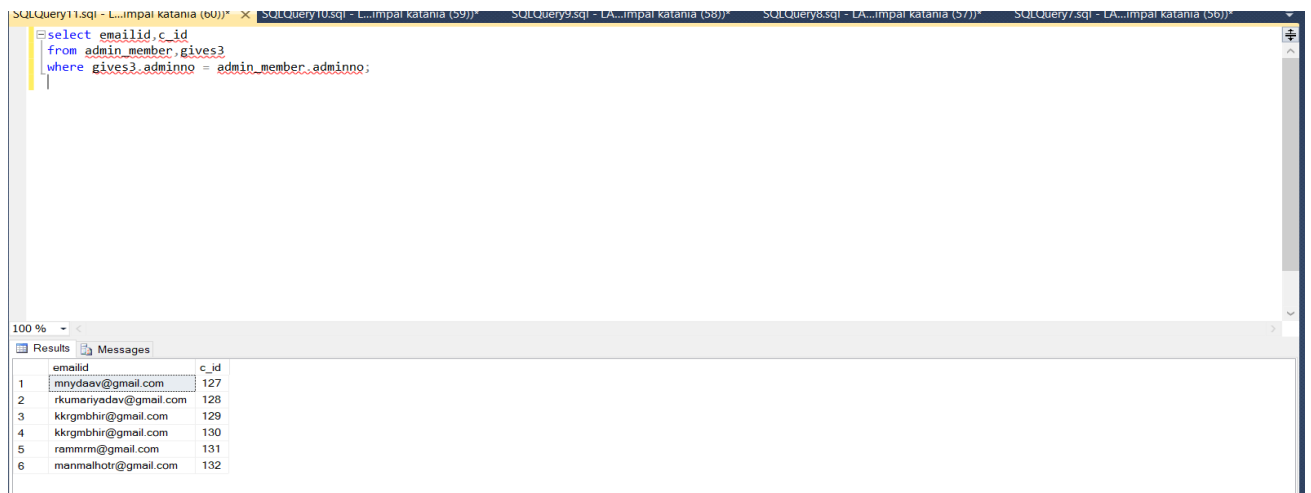
Results Messages

	fname
1	kiran
2	rakul

DERIVED RELATIONAL ALGEBRA QUERIES

Q1. List the email ids and c_ids of the administrative members who have filed the complaints

Ans1. Select emailed,c_id
From admin_member ,gives3
Where gives3.adminno=admin_member.adminno;



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL query:

```
select emailed,c_id
from admin_member,gives3
where gives3.adminno = admin_member.adminno;
```

The results window displays the following data:

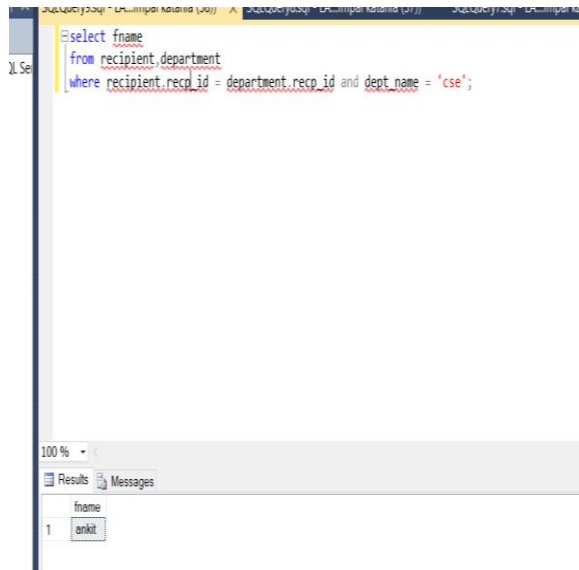
	emailed	c_id
1	mnydaav@gmail.com	127
2	rkumaryadav@gmail.com	128
3	kkrgmbhir@gmail.com	129
4	kkrgmbhir@gmail.com	130
5	rammm@gmail.com	131
6	manmalhotra@gmail.com	132

Q2. Retrieve the name of recipients who are handling the complaints of cse department

Ans2: Select fname

From recipient ,department

Where recipient.recip_id=department.recip_id and
dept_name='cse';



Q 3. List the name of administrative members who have at least one complaint.

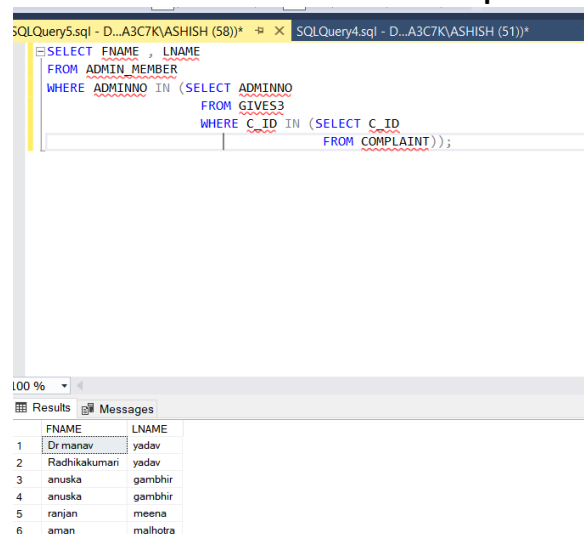
Ans : select fname ,lname

From admin member

Where adminno in (Select adminno

From gives3

Where c_id in (select c_id from
complaint));



The screenshot shows a SQL Developer window with two tabs. The active tab, 'SQLQuery5.sql', contains the following SQL query:

```
SELECT FNAME , LNAME
FROM ADMIN_MEMBER
WHERE ADMINNO IN (SELECT ADMINNO
FROM GIVES3
WHERE C_ID IN (SELECT C_ID
FROM COMPLAINT));
```

Below the query editor, the 'Results' tab is selected, displaying the output of the query in a table with 6 rows and 2 columns: FNAME and LNAME.

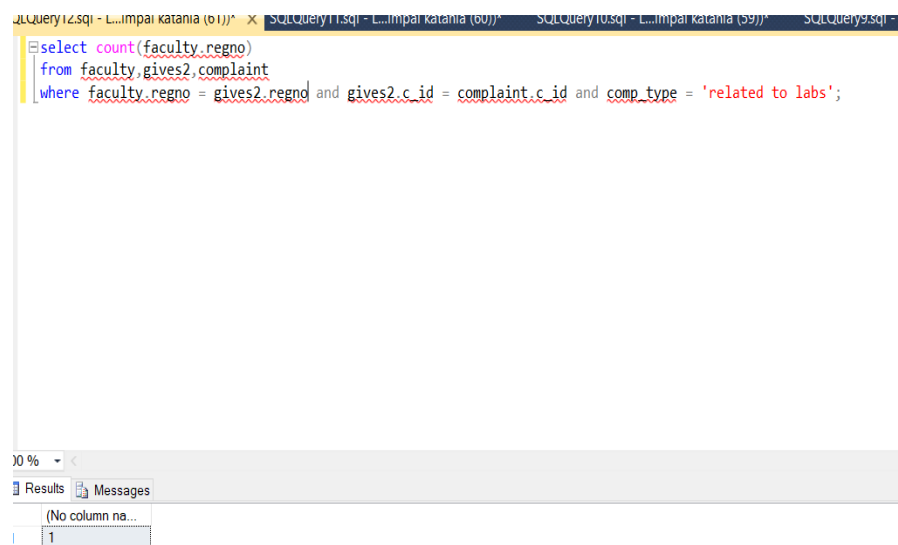
	FNAME	LNAME
1	Dr manav	yadav
2	Radhikakumari	yadav
3	anuska	gambhir
4	anuska	gambhir
5	ranjan	meena
6	aman	malhotra

Q4. Count the number of faculties who have filed complaints related to labs.

Ans :- Select count (faculty.regno)

From faculty,gives2,complaint

Where faculty.regno = gives2.regno and gives2.c_id and comp_type = 'related to labs';



The screenshot shows the SQL Developer interface. The top pane displays the following SQL query:

```
select count(faculty.regno)
from faculty,gives2,complaint
where faculty.regno = gives2.regno and gives2.c_id = complaint.c_id and comp_type = 'related to labs';
```

The bottom pane shows the 'Results' tab with a single row containing the value '1'.

(No column na...)
1

Q5. Write the query to fetch second last roll number from student table.

Ans: Select * from student
Where rollno in (select max(rollno)
From student
Where rollno not in
(Select Max(RollNo)
From Student));

```
SELECT * FROM STUDENT
WHERE ROLLNO IN ( SELECT MAX(ROLLNO)
                  FROM STUDENT
                  WHERE ROLLNO NOT IN
                  (SELECT MAX(ROLLNO)
                   FROM STUDENT)
                );
```

100 %

Results Messages

	FNAME	LNAME	EMAILID	SEX	BRANCH	ROLLNO	COMPID
1	dikshya	goyal	goyalddd@gmail.com	M	CSE	1008	6007

Q 6.Retrieve branch name of student having recipient id greater than equals to 7111.

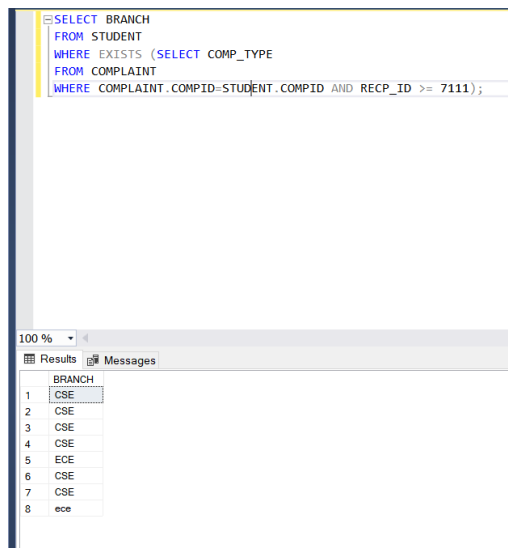
Ans: Select Branch

From Student

Where exists(SelectCOMP_TYPE

From Complaint

Where Complaint.compid=Student.compid and
recp_id>=7111);



```
SELECT BRANCH
FROM STUDENT
WHERE EXISTS (SELECT COMP_TYPE
FROM COMPLAINT
WHERE COMPLAINT.COMPID=STUDENT.COMPID AND RECP_ID >= 7111);
```

	BRANCH
1	CSE
2	CSE
3	CSE
4	CSE
5	ECE
6	CSE
7	CSE
8	ece

ADVANCED SQL QUERIES

Q1. List departments name whose complaints have been solved in ascending order

Ans: Select dept_name

From department, feedback where solved = 'yes'

Order by dept_name;

```
select dept_name  
from department, feedback  
where solved = 'yes'  
order by dept_name;
```

100 %

Results Messages

	dept_name
1	Accounts department
2	Accounts department
3	Accounts department
4	Accounts department
5	Accounts department
6	applied science
7	applied science
8	applied science
9	applied science
10	applied science
11	CSE
12	CSE
13	CSE
14	CSE
15	CSE

Q2. Retrieve recipient id and complain type having count = 1 and group them as per the complaint type.

Ans :Select count (recp_id),comp_type

From complaint

Group by comp_type

Having count (recp_id)=1;

The screenshot shows the SQL Developer interface. The top pane displays the following SQL query:

```
SELECT COUNT(RECP_ID), COMP_TYPE  
FROM COMPLAINT  
GROUP BY COMP_TYPE  
HAVING COUNT(RECP_ID) = 1;
```

The bottom pane shows the results of the query in a table with two columns: (No column name) and COMP_TYPE. The results are as follows:

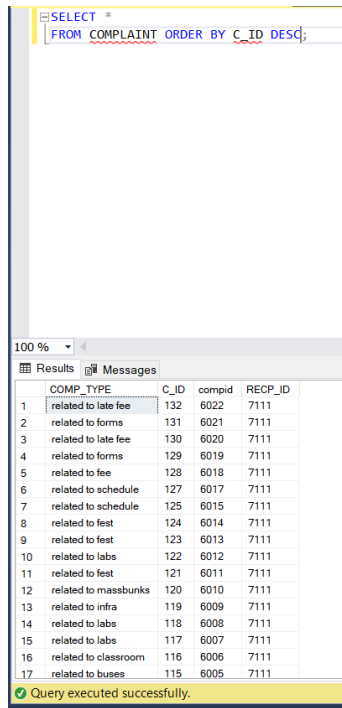
(No column name)	COMP_TYPE
1	related to buses
2	related to fee
3	related to infra
4	related to library
5	related to massbunks

At the bottom of the interface, a status bar indicates: Query executed successfully.

Q3.Retrieve c_ids from complaint and display them in descending order.

Ans:Select *

From Complaint Order By c_id Desc;



The screenshot shows a SQL query editor with the following query:

```
SELECT *  
FROM COMPLAINT ORDER BY C_ID DESC;
```

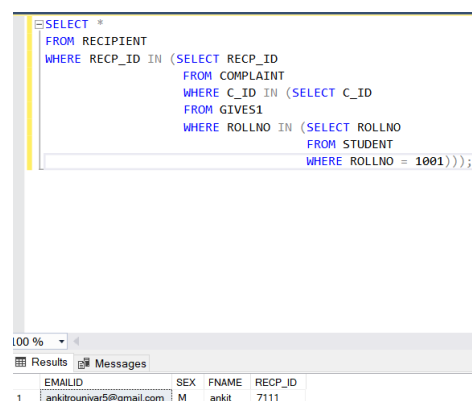
Below the query editor, the results are displayed in a table. The table has four columns: COMP_TYPE, C_ID, compid, and RECP_ID. The results are ordered by C_ID in descending order.

	COMP_TYPE	C_ID	compid	RECP_ID
1	related to late fee	132	6022	7111
2	related to forms	131	6021	7111
3	related to late fee	130	6020	7111
4	related to forms	129	6019	7111
5	related to fee	128	6018	7111
6	related to schedule	127	6017	7111
7	related to schedule	125	6015	7111
8	related to fest	124	6014	7111
9	related to fest	123	6013	7111
10	related to labs	122	6012	7111
11	related to fest	121	6011	7111
12	related to massbunks	120	6010	7111
13	related to infra	119	6009	7111
14	related to labs	118	6008	7111
15	related to labs	117	6007	7111
16	related to classroom	116	6006	7111
17	related to buses	115	6005	7111

At the bottom of the screenshot, a green status bar indicates: "Query executed successfully."

Q4. list all details about the recipient who is handling the complaint of students having roll greater than equals to 1001.

Ans: Select *
From Receptient
Where Recp_id IN (Select recp_id
From Complaint
Where c_id IN (Select c_id
From gives1
Where RollNo IN (Select RollNo
From Student
Where RollNo=1001)));



```
SELECT *  
FROM RECIPIENT  
WHERE RECP_ID IN (SELECT RECP_ID  
FROM COMPLAINT  
WHERE C_ID IN (SELECT C_ID  
FROM GIVES1  
WHERE ROLLNO IN (SELECT ROLLNO  
FROM STUDENT  
WHERE ROLLNO = 1001))));
```

	EMAILID	SEX	FNAME	RECP_ID
1	ankitrouniyar5@gmail.com	M	ankit	7111

Q5.Retrieve name of students and faculty who have filed complaints about lab.

Ans:Select fname as name

From student

Where rollno in(select rollno

From gives1

Where c_id in(Select c_id

From complaint

Where comp_type = 'related to labs'))

Union

Select fname

From faculty

Where regno in (Select regno

From gives2

Where c_id in (Select c_id

From complaint

Where comp_type='related to labs ');

```
select fname as name
from student
where rollno in(select rollno
                from gives1
                where c_id in (select c_id
                              from complaint
                              where comp_type = 'related to labs'))

union

select fname
from faculty
where regno in(select regno
               from gives2
               where c_id in (select c_id
                              from complaint
                              where comp_type = 'related to labs'));
```

100 %

Results Messages

	name
1	aashish
2	dikshya
3	hari
4	hiraranjans
5	Ram

Q6.Retrieve department ids and dept name having total department id equals to 1 .

Ans: Select count(dept_id),dept_name

From department

Group By dept_name

Having count(dept_id)=1;

```
Select count(DEPT_ID),dept_name
from department
group by dept_name
having count(dept_id) =1
```

100 %

Results Messages

	(No column name)	dept_name
1	1	Accounts department
2	1	applied science
3	1	CSE
4	1	ECE
5	1	EEE
6	1	Infrastructure Department

Q7. list the information about administrative members hose complaints is analyzed by cse department.

Ans: Select *

From admin_member

Where Exists (Select adminno

From gives3

Where admin_member.adminno=gives3.adminno and exists (Select c_id

From Complaint

Where complaint.c_id=Gives.c_id and exists (Select recp_id

From Department Where dept_name='cse')));

```
select *
from admin_member
where EXISTS ( SELECT ADMINNO
                FROM GIVES3
                WHERE ADMIN_MEMBER.ADMINNO = GIVES3.ADMINNO AND EXISTS (SELECT C_ID
                                                                            FROM COMPLAINT
                                                                            WHERE COMPLAINT.C_ID = GIVES3.C_ID AND EXISTS
                                                                            (SELECT RECP_ID
                                                                              FROM DEPARTMENT
                                                                              WHERE COMPLAINT.RECP_ID = DEPARTMENT.RECP_ID AND DEPT_NAME = 'CSE'))));
```

	FNAME	LNAME	EMAILID	SEX	DEPT	ADMINNO	COMPID
1	Dr manav	yadav	mnydaav@gmail.com	M	CSE	3321	6017
2	Radhikakumari	yadav	rkumariyadav@gmail.com	f	ECE	3322	6018
3	anusha	gambhir	kkrgmbhir@gmail.com	f	CSE	3323	6019
4	anusha	gambhir	kkrgmbhir@gmail.com	f	CSE	3324	6020
5	ranjan	meena	ramnm@gmail.com	M	EEE	3325	6021
6	aman	malhotra	manmalhot@gmail.com	M	EEE	3326	6022

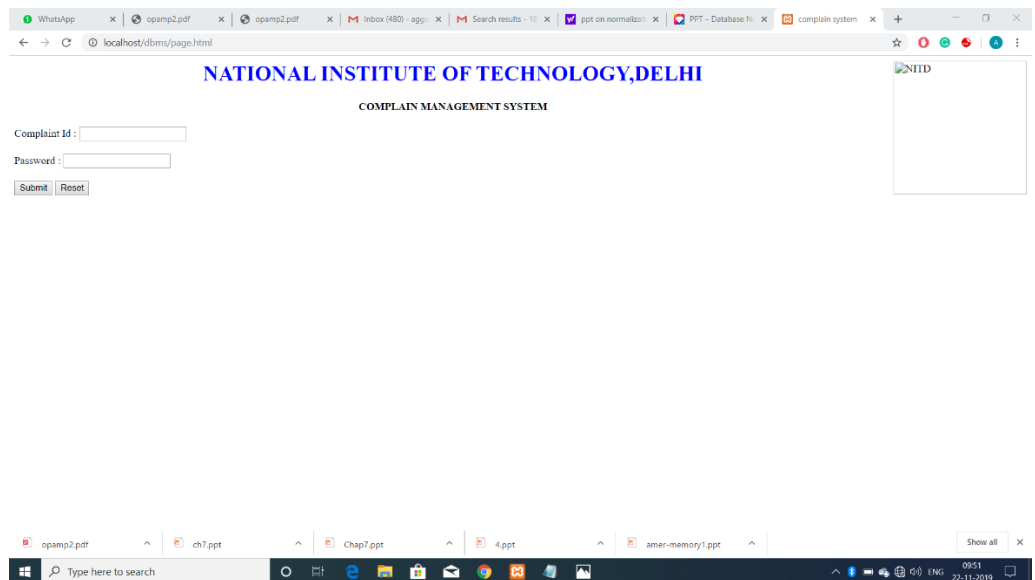
Front End Functionality

The Front End has the following functionalities: -

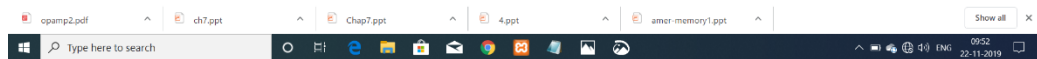
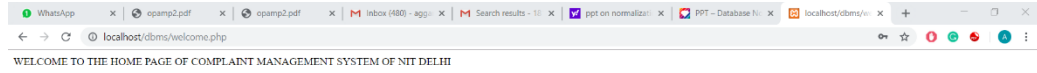
- ☐ A home page which contains login page where the user can login to file complain.
- ☐ Only the admins have the access to modify the data of the users.
- ☐ The user can see the status of the complain they have registered.
- ☐ The admin has features to filter out the searches by selecting options from a drop-down menu.

Some of the samples of the Front End: -

1. The LOGIN page



2. When the password is correct



Bibliography

□ Fundamentals of DATABASE SYSTEMS

By -Elmasri, Navathe

□ Database Management Systems

-Ramakrishnan, Gehrke

- www.stackoverflow.com
- www.youtube.com
- www.w3schools.com