DBMS Project Abstract

The Event Management System

Group Members:
Mithil Teja   -   AM.EN.U4CSE20159
Shoaib Akhtar   -   AM.EN.U4CSE20163
Rushikesh   -   AM.EN.U4CSE20131

ABSTRACT:

The "Event Management System" has been developed to override the problems prevailing in the practising manual system.

This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

Objective of Project on Event Management System:

The main objective of the project on event management system is to manage the details of event. Activity, Customer. It manages all the information about Event. Payment, Category-level, Food, Venue, sub-event etc. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the event.

Functionalities Provided by Event Management System are as follows:

- We will create a website which will contains the following:☐   It will ask the client who want to organise a event.

- They will be ask with to choose from the pre-defined list.

  ✓ In which category they fall

  ✓ How many sub events they required

- ✓ which kind of foods they are opting from the list

- ✓ What will be the venue of the event

- We will create a dedicated table for each of the options which we are providing

- For instance, In the food table, there will list of foods which we can provide along with the price of each food.

- And at last we will return the *Net Payable amount* to the client.

Scope of the project Event Management System

It may help collecting perfect management in details. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Event Management System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

- ➜ Our project aims at Business process automation, i.e. we have tried to computerize various processes of Event Management System.

- ➜ To assist the staff in capturing the effort spent on their respective working areas

- ➜ To utilize resources in an efficient manner by increasing their productivity through automation.
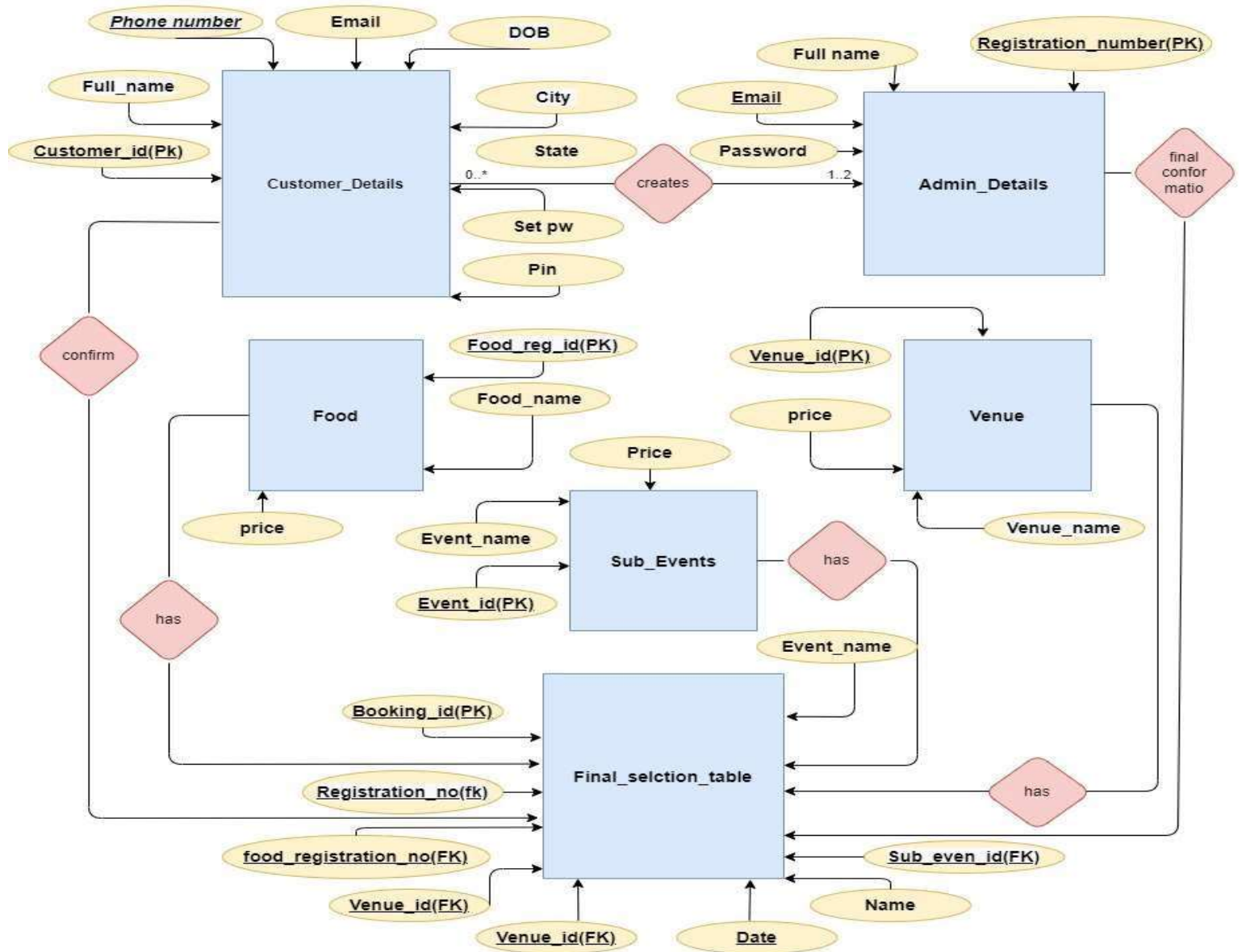
The system generates many types of information that can be used for various purposes.

- ➢ We can analyse the data very efficiently with this system.

- ➢ We can use these analysed data to understand the behaviour of the client, like in which season which type of events are organised frequently.

- ➢ What are the popular demands for food.

- ➢ What percentage of our client are of high-class, middle-class, lower-class, so that we can modify our services according to that
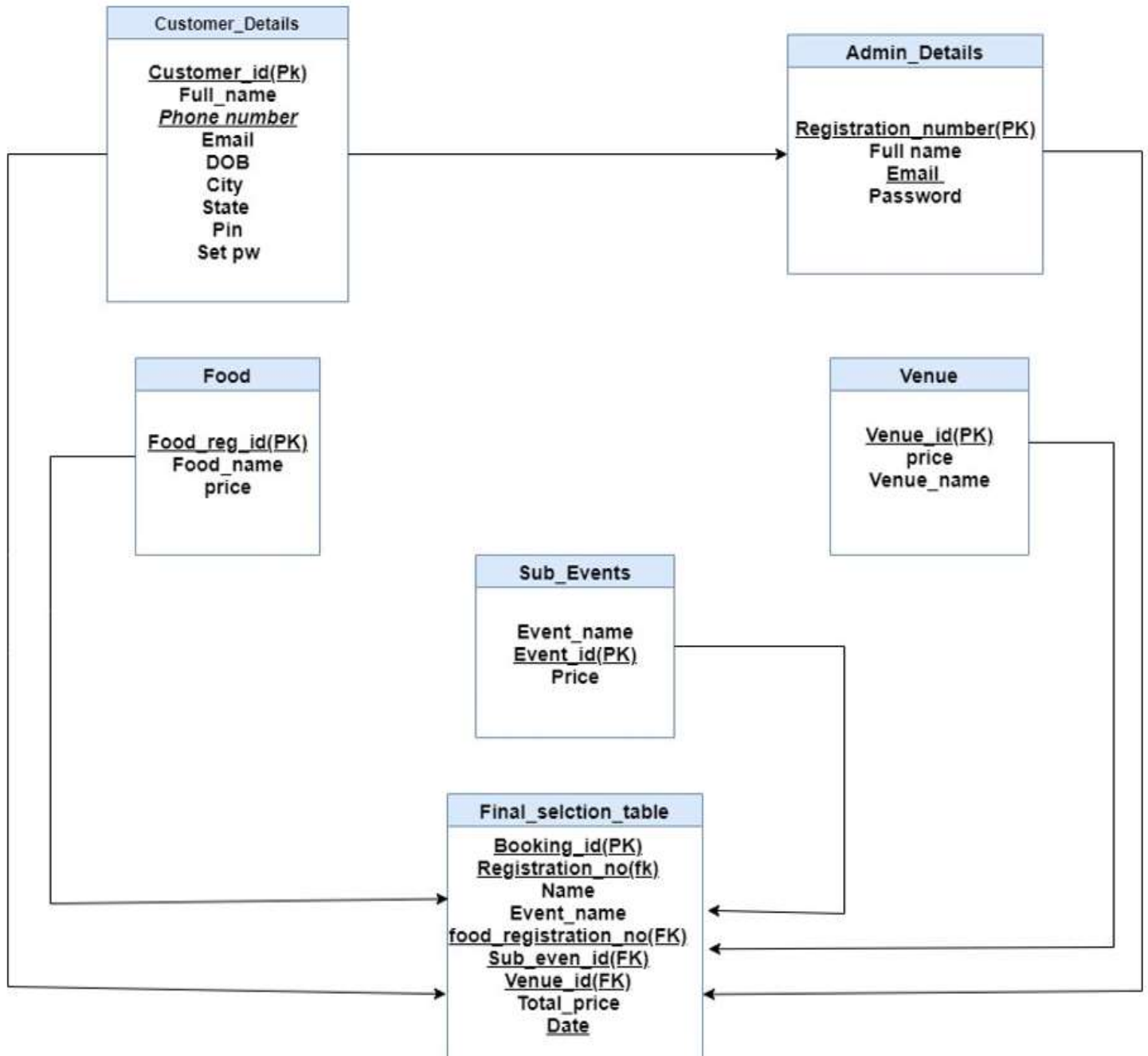
- Have a good user interface.

- We can modify our business strategy to earn maximum profits

ER DIAGRAM:-

SCHEMA DIAGRAM:-

Customer_Details

Customer_id(Pk)
Full_name
Phone number
Email
DOB
City
State
Pin
Set pw

Admin_Details

Registration_number(PK)
Full name
Email
Password

Food

Food_reg_id(PK)
Food_name
price

Venue

Venue_id(PK)
price
Venue_name

Sub_Events

Event_name
Event_id(PK)
Price

Final_selction_table

Booking_id(PK)
Registration_no(fk)
Name
Event_name
food_registration_no(FK)
Sub_even_id(FK)
Venue_id(FK)
Total_price
Date

RELATIONAL SCHEMA:-

- customer_details (<u>registration_no [auto assign]</u>, full_name, <u>phone_no</u>, <u>email</u>, DOB, city, state, PIN, set_password)

- admin_details (<u>registration_no,</u> full_name<u>, email</u>, password)

- food (<u>food_reg_no,</u> food_name, price)

- venue (<u>venue_id</u>, venue_name, price)

- sub_events (<u>sub_event_id,</u> sub_event_name, price)

- final_selection_table (<u>booking_no [auto assign],</u> registration_no, full_name, <u>event_type, venue_id,</u> <u>date,</u> <u>food_reg_no,</u> <u>sub_event_ID,</u> total_amount)

FINAL NORMALIZATION TABLE:-

Admin user ID	Admin pw	Customer User id (pk)	Customer password	Customer Full name	Customer Phone no:	Customer g-mail	Main event	Main event price	venue	Venue cost	food	Food cost	Sub event	Sub Event price	people	total
Rushi	Rushi	Nihaas	Nihaas 123	Nihaas reddy	9834938980	Nihaasreddy@gmail.com	marriage	50000	patna	5000	EIT	60	_	0	100	61000
Rushi	Rushi	Ravi	Ravi 345	Ravi chandu	9389483459	ravichandu@gmail.com	birthday	20000	delhi	15000	SIT	70	sangeet	10000	200	59000
Rushi	Rushi	Koushik	Koushik 678	koushik	9283475849	koushik@gmail.com	Engagement	30000	hyd	10000	WIT	80	mehendi	5000	300	69000
Rushi	Rushi	Karthik	Karthik 901	Karthik reddy	9382923784	karthikreddy@gmail.com	Reunion party	8000	chennai	12000	NIT	100	Oil bath	4000	50	29000
Rushi	Rushi	Pratyush	Pratyush 234	Pratyush ranjan	9012834859	Pratyushrocks@gmail.com	Family party	10000	mumbai	18000	EIT	60	_	0	400	52000
Rushi	Rushi	Raviteja	Raviteja 567	raviteja	9184278934	raviteja@gmail.com	marriage	50000	delhi	15000	WIT	80	Haldi ritual	3000	150	80000
Rushi	Rushi	Aquib	Aquib 890	Shaik aquib	9837482734	shaikaquibsharky@gmail.com	birthday	20000	Hyd	10000	NIT	100	mahendi	5000	250	60000
Rushi	Rushi	Nikil	Nikil 123	Nikil koushik	9387283948	Nikilkoushik47@gmail.com	Reunion party	8000	mumbai	18000	NIT	100	sangeet	10000	600	87000
Rushi	Rushi	Satvik	Satvik 456	Satvik	9382734973	Satvik2003@gmail.com	marriage	50000	delhi	15000	SIT	70	Oil bath	4000	800	121400
Rushi	Rushi	Abhi	Abhi 789	abhishek	9037483928	Abhishek1947@gmail.com	Family party	10000	delhi	15000	EIT	60	_	0	200	37000

DATABASE NORMALIZATION:-

Database normalization can reduce data redundancy caused by insertion,deletion and updation anomalies and makes the data more meaningful.

First Normal Form Table:

The rules for the table to be in first normal form are:

1.   Single Values Atributes : Each column of the table should be single valued which means they should not contain multiple values.

2.   Attribute domain should not change : In each column,the values stored must be of same type or kind.

3.   Unique name for columns : To avoid confusion,each column must have a unique name.

4.   Order doesn't matter:-The rule says the order in which you store data in your table doesn't matter.

Admin user ID	Admin pw	Customer User id (pk)	Customer password	Customer Full name	Customer Phone no:	Customer g-mail	Main event	Main event price	venue	Venue cost	food	Food cost	Sub event	Sub Event price	people	total
Rushi	Rushi	Nihaas	Nihaas 123	Nihaas reddy	9.835E+09	Nihaasreddy@g mail.co	marriage	50000	patna	5000	EIT	60	_	0	100	61000
Rushi	Rushi	Nihaas	Nihaas 123	Nihaas reddy	9.835E+09	Nihaasreddy@g mail.co m	marriage	50000	Patna	5000	EIT	60	_	0	100	61000
Rushi	Rushi	Ravi	Ravi 234	Ravi chandu	9.383E+09	ravichandu@ga mil.com	Birthday	20000	Delhi	15000	SIT	70	Sangeet	10000	200	59000
Rushi	Rushi	Koushik	Koushik 678	koushik	9.283E+09	koushik@gmail. com	Engage ment	30000	Hyd	10000	WIT	80	Mehend i	5000	300	69000
Rushi	Rushi	Karthik	Karthik 901	Karthik reddy	9.383E+09	karthikr eddy@g mail.co	Reunion party	8000	Chennai	12000	NIT	100	Oil bath	4000	50	29000
Rushi	Rushi	Karthik	Karthik 901	Karthik reddy	9.383E+09	karthikr eddy@g mail.co	Reunion party	8000	Chennai	12000	NIT	100	Oil bath	4000	50	29000
Rushi	Ruhsi	Pratyush	Pratyush 234	Pratyush Ranjan	9.838E+09	Pratyush rocks@g amil.co	Family Party	10000	Mumbai	18000	EIT	60	_	0	400	52000
Rushi	Rushi	Raviteja	Raviteja 567	raviteja	9.184E+09	raviteja @gmail. com	marriage	50000	Delhi	15000	WIT	80	Haldi ritual	3000	150	80000
Rushi	Rushi	Aquib	Aquib 890	Shaik aquib	9.837E+09	shaikaq uibshar ky@gma il.com	birthday	20000	Hyd	10000	NIT	100	mahend i	5000	250	60000
Rushi	Rushi	Nikil	Nikil 123	Nikil koushik	9.387E+09	Nikilkou shik47@ gmail.co	Reunion party	8000	mumbai	18000	NIT	100	sangeet	10000	600	87000
Rushi	Rushi	Satvik	Satvik 456	Satvik	9.383E+09	Satvik20 03@gma il.com	marriage	50000	delhi	15000	SIT	70	Oil bath	4000	800	121400
Rushi	Rushi	Abhi	Abhi 789	abhishe k	9.037E+09	Abhishe k1947@ gmail.co	Family party	10000	delhi	15000	EIT	60	_	0	200	37000

Second normal form table:-

The rules for table to be in second normal form are:-

1.   The table should be in the First Normal Form.
2.   There should be no partial dependency i.e any attribute of the table shouldn't depend on a part of the primary key but depend on the whole primary key instead.

Customer User id (pk)	Customer password password	Customer Full name	Customer Phone no:	Customer g-mail
Nihaas	Nihaas 123	Nihaas reddy	9834938980	Nihaasreddy@gmail.com
Nihaas	Nihaas 123	Nihaas reddy	9834938980	Nihaasreddy@gmail.com
Ravi	Ravi234	Ravi chandu	9382938293	ravichandu@gamil.com
Koushik	Koushik 678	koushik	9283475849	koushik@gmail.com
Karthik	Karthik 901	Karthik reddy	9382923784	karthikreddy@gmail.com
Karthik	Karthik 901	Karthik reddy	9382923784	karthikreddy@gmail.com
Pratyush	Pratyush 234	Pratyush Ranjan	9837849390	Pratyushrocks@gamil.com
Raviteja	Raviteja 567	raviteja	9184278934	raviteja@gmail.com
Aquib	Aquib 890	Shaik aquib	9837482734	shaikaquibsharky@gmail.com
Nikil	Nikil 123	Nikil koushik	9387283948	Nikilkoushik47@gmail.com
Satvik	Satvik 456	Satvik	9382734973	Satvik2003@gmail.com
Abhi	Abhi 789	abhishek	9037483928	Abhishek1947@gmail.com

Customer User id (pw)	Main event	Main event price	venue	Venue cost	food	Food cost	Sub event	Sub Event price
Nihaas	marriage	50000	patna	5000	EIT	60	_	0
Nihaas	marriage	50000	Patna	5000	EIT	60	_	0
Ravi	Birthday	20000	Delhi	15000	SIT	70	Sangeet	10000
Koushik	Engagement	30000	Hyd	10000	WIT	80	Mehendi	5000
Karthik	Reunion party	8000	Chennai	12000	NIT	100	Oil bath	4000
Karthik	Reunion party	8000	Chennai	12000	NIT	100	Oil bath	4000
Pratyush	Family Party	10000	Mumbai	18000	EIT	60	_	0
Raviteja	marriage	50000	Delhi	15000	WIT	80	Haldi ritual	3000
Aquib	birthday	20000	Hyd	10000	NIT	100	mahendi	5000
Nikil	Reunion party	8000	mumbai	18000	NIT	100	sangeet	10000
Satvik	marriage	50000	delhi	15000	SIT	70	Oil bath	4000
Abhi	Family party	10000	delhi	15000	EIT	60	_	0

Admin user ID	Admin pw	Customer User id (pw)
Rushi	Rushi	Nihaas
Rushi	Rushi	Nihaas
Rushi	Rushi	Ravi
Rushi	Rushi	Koushik
Rushi	Rushi	Karthik
Rushi	Rushi	Karthik
Rushi	Ruhsi	Pratyush
Rushi	Rushi	Raviteja
Rushi	Rushi	Aquib
Rushi	Rushi	Nikil
Rushi	Rushi	Satvik
Rushi	Rushi	Abhi

Customer User id (pw)	total
Nihaas	61000
Nihaas	61000
Ravi	59000
Koushik	69000
Karthik	29000
Karthik	29000
Pratyush	52000
Raviteja	80000
Aquib	60000
Nikil	87000
Satvik	121400
Abhi	37000

Third Normal Form

The rules for the table to be in Third Normal Form are:-

1.   It should be in Second Normal Form.
2.   It should have no transitive dependency i.e. any non-prime attribute shouldn't depend on any other non-prime attribute but only a prime attribute or primary key.

Customer User id (pk)	Customer password password	Customer Full name	Customer Phone no:	Customer g-mail
Nihaas	Nihaas 123	Nihaas reddy	9834938980	Nihaasreddy@gmail.com
Nihaas	Nihaas 123	Nihaas reddy	9834938980	Nihaasreddy@gmail.com
Ravi	Ravi234	Ravi chandu	9382938293	ravichandu@gamil.com
Koushik	Koushik 678	koushik	9283475849	koushik@gmail.com
Karthik	Karthik 901	Karthik reddy	9382923784	karthikreddy@gmail.com
Karthik	Karthik 901	Karthik reddy	9382923784	karthikreddy@gmail.com
Pratyush	Pratyush 234	Pratyush Ranjan	9837849390	Pratyushrocks@gamil.com
Raviteja	Raviteja 567	raviteja	9184278934	raviteja@gmail.com
Aquib	Aquib 890	Shaik aquib	9837482734	shaikaquibsharky@gmail.com
Nikil	Nikil 123	Nikil koushik	9387283948	Nikilkoushik47@gmail.com
Satvik	Satvik 456	Satvik	9382734973	Satvik2003@gmail.com
Abhi	Abhi 789	abhishek	9037483928	Abhishek1947@gmail.com

Admin user ID	Admin pw
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi
Rushi	Rushi

Admin user ID	Customer User id (pw)
Rushi	Nihaas
Rushi	Nihaas
Rushi	Ravi
Rushi	Koushik
Rushi	Karthik
Rushi	Karthik
Rushi	Pratyush
Rushi	Raviteja
Rushi	Aquib
Rushi	Nikil
Rushi	Satvik
Rushi	Abhi

Customer User id (pw)	Main event	Main event price
Nihaas	marriage	50000
Nihaas	marriage	50000
Ravi	Birthday party	20000
Koushik	Engagement	30000
Karthik	Reunion party	8000
Karthik	Reunion party	8000
Pratyush	Family party	10000
Raviteja	marriage	50000
Aquib	birthday	20000
Nikil	Reunion party	8000
Satvik	marriage	50000
Abhi	Family party	10000

Customer User id (pw)	venue	Venue cost
Nihaas	patna	5000
Nihaas	Patna	5000
Ravi	Delhi	15000
Koushik	Hyd	10000
Karthik	Chennai	12000
Karthik	Chennai	12000
Pratyush	mumbai	18000
Raviteja	Delhi	15000
Aquib	Hyd	10000
Nikil	mumbai	18000
Satvik	delhi	15000
Abhi	delhi	15000

Customer User id (pw)	food	Food cost
Nihaas	EIT	60
Nihaas	EIT	60
Ravi	SIT	70
Koushik	WIT	80
Karthik	NIT	100
Karthik	NIT	100
Pratyush	EIT	60
Raviteja	WIT	80
Aquib	NIT	100
Nikil	NIT	100
Satvik	SIT	70
Abhi	EIT	60

Customer User id (pw)	Sub event	Sub Event price
Nihaas	_	0
Nihaas	_	0
Ravi	Sangeet	10000
Koushik	Mehendi	5000
Karthik	Oil bath	4000
Karthik	Oil bath	4000
Pratyush	-	0
Raviteja	Haldi ritual	3000
Aquib	mahendi	5000
Nikil	sangeet	10000
Satvik	Oil bath	4000

Customer User id (pw)	total
Nihaas	61000
Nihaas	61000
Ravi	59000
Koushik	69000
Karthik	29000
Karthik	29000
Pratyush	52000
Raviteja	80000
Aquib	60000
Nikil	87000
Satvik	121400
Abhi	37000

COMMAND:
create table table_1_3NF
(
serial_no numeric(10),
admin_username varchar(30),
admin_password varchar(30),

primary key(serial_no)
)
INSERTION:

insert into table_1_3nf(admin_username, admin_password)
values
('rushi1', 'rushi'),
('rushi2', 'rushi'),
('rushi3', 'rushi'),
('rushi4', 'rushi'),
('rushi5', 'rushi'),
('rushi6', 'rushi'),
('rushi7', 'rushi'),
('rushi8', 'rushi'),
('rushi9', 'rushi'),
('rushi10', 'rushi'),
('rushi11', 'rushi'),
('rushi12', 'rushi');

CODE:

```
Query Editor    Query History
  1    select * from table_1_3nf;
```

OUTPUT:

	admin_username [PK] character varying (30)	admin_password character varying (30)
1	rushi1	rushi
2	rushi2	rushi
3	rushi3	rushi
4	rushi4	rushi
5	rushi5	rushi
6	rushi6	rushi
7	rushi7	rushi
8	rushi8	rushi
9	rushi9	rushi
10	rushi10	rushi
11	rushi11	rushi
12	rushi12	rushi

COMMAND:

create table table_2_3NF
(
admin_username varchar(30),
customer_username varchar(30),

primary key(customer_username),
~~**foreign key(admin_username) references table_1_3NF(admin_username)**~~
foreign key(admin_username) references table_1_3NF(serial_no)
);

INSERTION:

insert into table_2_3nf(admin_username, customer_username)
values
('rushi1', 'nihaas'),
('rushi2', 'nihaasreddy'),
('rushi3', 'ravi'),
('rushi4', 'kaushik'),

('rushi5', 'kartheek'),
('rushi6', 'karthik'),
('rushi7', 'pratyush'),
('rushi8', 'raviteja'),
('rushi9', 'aqib'),
('rushi10', 'nikhil'),
('rushi11', 'sarthak'),
('rushi12', 'abhi');

CODE:

Query Editor   Query History

```
1    select * from table_2_3nf;
```

OUTPUT:

Data Output   Messages   Explain   Notifications

	admin_username character varying (30)	customer_username [PK] character varying (30)
1	rushi1	nihaas
2	rushi2	nihaasreddy
3	rushi3	ravi
4	rushi4	kaushik
5	rushi5	kartheek
6	rushi6	karthik
7	rushi7	pratyush
8	rushi8	raviteja
9	rushi9	aqib
10	rushi10	nikhil
11	rushi11	sarthak
12	rushi12	abhi

COMMAND:

create table table_3_3NF
(
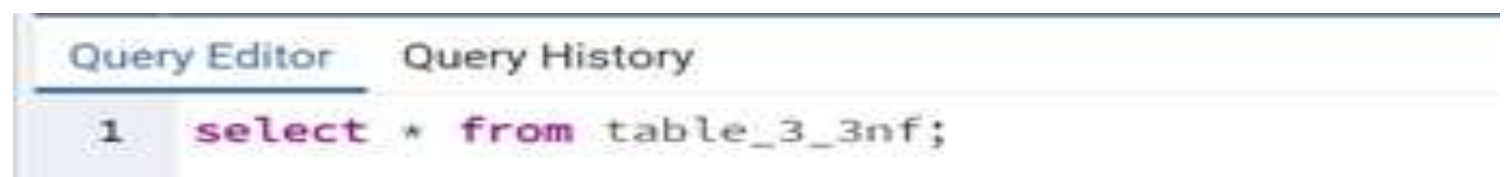
customer_username varchar(30),
customer_password varchar(30),
customer_fullname varchar(30),
customer_phone_no numeric(10),
customer_email varchar(30),

foreign key(customer_username) references table_2_3NF(customer_username)
);

INSERTION:

insert into table_3_3nf(customer_username, customer_password, customer_fullname, customer_phone_no, customer_email)
values
('nihaas', 'nihaas123', 'nihas reddy', 9998979695, 'nihas@abc.com'),
('nihaasreddy', 'nihaasreddy123', 'nihaas teja', 9998979695, 'nihaas@abc.com'),
('ravi', 'ravi123', 'bollepali ravi ', 9998979695, 'ravi@abc.com'),
('kaushik', 'kaushik123', 'kaushik kumar', 9998979695, 'kaushik@abc.com'),
('kartheek', 'kartheek123', 'kartheek vishal', 9998979695, 'kartheek@abc.com'),
('karthik', 'karthik123', 'karthik reddy', 9998979695, 'karthik@abc.com'),
('pratyush', 'pratyush123', 'pratyush ranjan', 9998979695, 'pratyush@abc.com'),
('raviteja', 'raviteja123', 'ravi teja', 9998979695, 'ravitaja@abc.com'),
('aqib', 'aqib123', 'shaik aquibuddin', 9998979695, 'shaik@abc.com'),
('nikhil', 'nikhil123', 'nikhil kumar', 9998979695, 'nikhil@abc.com'),
('sarthak', 'sarthak123', 'sarthak goswami', 9998979695, 'sarthak@abc.com'),
('abhi', 'abhi123', 'abhi reddy', 9998979695, 'abhi@abc.com');

CODE:

```
Query Editor    Query History

  1    select * from table_3_3nf;
```

OUTPUT:

	customer_username character varying (30)	customer_password character varying (30)	customer_fullname character varying (30)	customer_phone_no numeric (10)	customer_email character varying (30)
1	nihaas	nihaas123	nihas reddy	9998979695	nihas@abc.com
2	nihaasreddy	nihaasreddy123	nihaas teja	9998979695	nihaas@abc.com
3	ravi	ravi123	bollepali ravi	9998979695	ravi@abc.com
4	kaushik	kaushik123	kaushik kumar	9998979695	kaushik@abc.com
5	kartheek	kartheek123	kartheek vishal	9998979695	kartheek@abc.com
6	karthik	karthik123	karthik reddy	9998979695	karthik@abc.com
7	pratyush	pratyush123	pratyush ranjan	9998979695	pratyush@abc.com
8	raviteja	raviteja123	ravi teja	9998979695	ravitaja@abc.com
9	aqib	aqib123	shaik aquibuddin	9998979695	shaik@abc.com
10	nikhil	nikhil123	nikhil kumar	9998979695	nikhil@abc.com
11	sarthak	sarthak123	sarthak goswami	9998979695	sarthak@abc.com
12	abhi	abhi123	abhi reddy	9998979695	abhi@abc.com

COMMAND:

create table table_4_3NF
(

customer_username varchar(30),
event varchar(30),
event_price numeric(10),

foreign key(customer_username) references table_2_3NF(customer_username)
);

INSERTION:

insert into table_4_3nf(customer_username, event, event_price)
values
('nihaas', 'marriage', 50000),
('nihaasreddy', 'marriage', 50000),
('ravi', 'birthday party', 20000),

('kaushik', 'engagement ceremony', 30000),
('kartheek', 'reunion party', 8000),
('karthik', 'reunion party', 8000),
('pratyush', 'family function', 10000),
('raviteja', 'marriage', 50000),
('aqib', 'birthday party', 20000),
('nikhil', 'reunion party', 8000),
('sarthak', 'marriage', 50000),
('abhi', 'family function', 10000);

CODE:

```
Query Editor    Query History
1    select * from table_4_3nf;
```

OUTPUT:

Data Output   Messages   Explain   Notifications

	customer_username character varying (30)	event character varying (30)	event_price numeric (10)
1	nihaas	marriage	50000
2	nihaasreddy	marriage	50000
3	ravi	birthday party	20000
4	kaushik	engagement ceremony	30000
5	kartheek	reunion party	8000
6	karthik	reunion party	8000
7	pratyush	family function	10000
8	raviteja	marriage	50000
9	aqib	birthday party	20000
10	nikhil	reunion party	8000
11	sarthak	marriage	50000
12	abhi	family function	10000

COMMAND:

create table table_5_3NF
(

customer_username varchar(30),
venue varchar(30),
venue_price numeric(10),

foreign key(customer_username) references table_2_3NF(customer_username)
);

insert into table_5_3nf(customer_username, venue, venue_price)
values
('nihaas', 'patna', 5000),
('nihaasreddy', 'patna', 5000),
('ravi', 'delhi', 15000),
('kaushik', 'hyderabad', 10000),
('kartheek', 'chennai', 12000),
('karthik', 'chennai', 12000),
('pratyush', 'mumbai', 18000),
('raviteja', 'delhi', 15000),
('aqib', 'hyderabad', 10000),
('nikhil', 'mumbai', 18000),
('sarthak', 'delhi', 15000),
('abhi', 'delhi', 15000);

CODE:

```
Query Editor    Query History
1    select * from table_5_3nf;
```

OUTPUT:

	customer_username character varying (30)	venue character varying (30)	venue_price numeric (10)
1	nihaas	patna	5000
2	nihaasreddy	patna	5000
3	ravi	delhi	15000
4	kaushik	hyderabad	10000
5	kartheek	chennai	12000
6	karthik	chennai	12000
7	pratyush	mumbai	18000
8	raviteja	delhi	15000
9	aqib	hyderabad	10000
10	nikhil	mumbai	18000
11	sarthak	delhi	15000
12	abhi	delhi	15000

COMMAND:

create table table_6_3NF
(

customer_username varchar(30),
food varchar(30),
food_price numeric(10),

foreign key(customer_username) references table_2_3NF(customer_username)
);

INSERTION:

insert into table_6_3nf(customer_username, food, food_price)
values
('nihaas', 'east indian food', 60),
('nihaasreddy', 'east indian food', 60),
('ravi', 'south indian food', 70),
('kaushik', 'west indian food', 80),

('kartheek', 'north indian food', 100),
('karthik', 'north indian food', 100),
('pratyush', 'east indian food', 60),
('raviteja', 'west indian food', 80),
('aqib', 'north indian food', 100),
('nikhil', 'north indian food', 100),
('sarthak', 'south indian food', 70),
('abhi', 'east indian food', 60);

CODE:

Query Editor   Query History

```
1   select * from table_6_3nf;
```

OUTPUT:

Data Output   Messages   Explain   Notifications

	customer_username character varying (30)	food character varying (30)	food_price numeric (10)
1	nihaas	east indian food	60
2	nihaasreddy	east indian food	60
3	ravi	south indian food	70
4	kaushik	west indian food	80
5	kartheek	north indian food	100
6	karthik	north indian food	100
7	pratyush	east indian food	60
8	raviteja	west indian food	80
9	aqib	north indian food	100
10	nikhil	north indian food	100
11	sarthak	south indian food	70
12	abhi	east indian food	60

COMMAND:

create table table_7_3NF

(

customer_username varchar(30),
total_amount numeric(10),

foreign key(customer_username) references table_2_3NF(customer_username)
);

insert into table_7_3nf(customer_username, total_amount)
values
('nihaas', 60000),
('nihaasreddy', 60000),
('ravi', 59000),
('kaushik', 69000),
('kartheek', 29000),
('karthik', 29000),
('pratyush', 52000),
('raviteja', 80000),
('aqib', 60000),
('nikhil', 87000),
('sarthak', 121400),
('abhi', 37000);

CODE:

```
Query Editor    Query History

1    select * from table_7_3nf;
```

OUTPUT:

	customer_username character varying (30)	total_amount numeric (10)
1	nihaas	60000
2	nihaasreddy	60000
3	ravi	59000
4	kaushik	69000
5	kartheek	29000
6	karthik	29000
7	pratyush	52000
8	raviteja	80000
9	aqib	60000
10	nikhil	87000
11	sarthak	121400
12	abhi	37000

COMMAND:

create table table_8_3NF
(

customer_username varchar(30),
subevent varchar(30),
subevent_amount numeric(10),

foreign key(customer_username) references table_2_3NF(customer_username)
);

INSERTION:

insert into table_8_3nf(customer_username, subevent, subevent_amount)
values
('nihaas', 'no-subevent', 0),
('nihaasreddy', 'no-subevent', 0),
('ravi', 'sangeet', 10000),

('kaushik', 'mehandi', 5000),
('kartheek', 'oil-bath event', 4000),
('karthik', 'north indian food', 100),
('pratyush', 'no-subevent', 0),
('raviteja', 'haldi ritual', 3000),
('aqib', 'mehandi', 5000),
('nikhil', 'sangeet', 10000),
('sarthak', 'oil-bath event', 4000),
('abhi', 'no-subevent', 0);

CODE:

```
Query Editor    Query History

1    select * from table_8_3nf;
```

OUTPUT:

Data Output   Messages   Explain   Notifications

	customer_username character varying (30)	subevent character varying (30)	subevent_amount numeric (10)
1	nihaas	no-subevent	0
2	nihaasreddy	no-subevent	0
3	ravi	sangeet	10000
4	kaushik	mehandi	5000
5	kartheek	oil-bath event	4000
6	karthik	north indian food	100
7	pratyush	no-subevent	0
8	raviteja	haldi ritual	3000
9	aqib	mehandi	5000
10	nikhil	sangeet	10000
11	sarthak	oil-bath event	4000
12	abhi	no-subevent	0

Q) select sum(total_amount) from table_7_3nf;

CODE:

DBMS_Project/postgres@PostgreSQL 13 ∨
Query Editor   Query History
1 select sum(total_amount) from table_7_3nf;
2

OUTPUT:

Data Output   Messages   Explain   Notifications

	sum numeric 🔒
1	743400

Q) select customer_username from table_8_3nf group by customer_username;

CODE:

Query Editor   Query History
1 select customer_username from table_8_3nf group by customer_username;
2

OUTPUT:-

	customer_username character varying (30)
1	pratyush
2	nihaas
3	kaushik
4	abhi
5	nihaasreddy
6	kartheek
7	sarthak
8	raviteja
9	karthik
10	ravi
11	aqib
12	nikhil

Q) select customer_username from table_8_3nf order by customer_username;

CODE:

```
select customer_username from table_8_3nf order by customer_username;
```

OUTPUT:

	customer_username character varying (30)
1	abhi
2	aqib
3	kartheek
4	karthik
5	kaushik
6	nihaas
7	nihaasreddy
8	nikhil
9	pratyush
10	ravi
11	raviteja
12	sarthak

Q) select table_3_3nf.customer_username, table_3_3nf.customer_password, table_3_3nf.customer_fullname, table_7_3nf.total_amount from table_3_3nf join table_7_3nf on table_3_3nf.customer_username = table_7_3nf.customer_username;
CODE:

Query Editor   Query History

```
1
2  select table_3_3nf.customer_username, table_3_3nf.customer_password, table_3_3nf.customer_fullname, table_7_3nf.total_amount
3  from table_3_3nf join table_7_3nf on table_3_3nf.customer_username = table_7_3nf.customer_username;
4  |
5
```

OUTPUT:

Data Output   Messages   Explain   Notifications

	customer_username character varying (30)	customer_password character varying (30)	customer_fullname character varying (30)	total_amount numeric (10)
1	nihaas	nihaas123	nihas reddy	60000
2	nihaasreddy	nihaasreddy123	nihaas teja	60000
3	ravi	ravi123	bollepali ravi	59000
4	kaushik	kaushik123	kaushik kumar	69000
5	kartheek	kartheek123	kartheek vishal	29000
6	karthik	karthik123	karthik reddy	29000
7	pratyush	pratyush123	pratyush ranjan	52000
8	raviteja	raviteja123	ravi teja	80000
9	aqib	aqib123	shaik aquibuddin	60000
10	nikhil	nikhil123	nikhil kumar	87000
11	sarthak	sarthak123	sarthak goswami	121400
12	abhi	abhi123	abhi reddy	37000

Q) select table_3_3nf.customer_username, table_3_3nf.customer_password, table_3_3nf.customer_fullname, table_7_3nf.total_amount, table_3_3nf.customer_email from table_3_3nf full outer join table_7_3nf on table_3_3nf.customer_username = table_7_3nf.customer_username;
CODE:

Query Editor   Query History

```
1  select table_3_3nf.customer_username, table_3_3nf.customer_password, table_3_3nf.customer_fullname,
2  table_7_3nf.total_amount, table_3_3nf.customer_email
3  from table_3_3nf full outer join table_7_3nf on table_3_3nf.customer_username = table_7_3nf.customer_username;
```

OUTPUT:

	customer_username character varying (30)	customer_password character varying (30)	customer_fullname character varying (30)	total_amount numeric (10)	customer_email character varying (30)
1	nihaas	nihaas123	nihas reddy	60000	nihas@abc.com
2	nihaasreddy	nihaasreddy123	nihaas teja	60000	nihaas@abc.com
3	ravi	ravi123	bollepali ravi	59000	ravi@abc.com
4	kaushik	kaushik123	kaushik kumar	69000	kaushik@abc.com
5	kartheek	kartheek123	kartheek vishal	29000	kartheek@abc.com
6	karthik	karthik123	karthik reddy	29000	karthik@abc.com
7	pratyush	pratyush123	pratyush ranjan	52000	pratyush@abc.com
8	raviteja	raviteja123	ravi teja	80000	ravitaja@abc.com
9	aqib	aqib123	shaik aquibuddin	60000	shaik@abc.com
10	nikhil	nikhil123	nikhil kumar	87000	nikhil@abc.com
11	sarthak	sarthak123	sarthak goswami	121400	sarthak@abc.com
12	abhi	abhi123	abhi reddy	37000	abhi@abc.com

ARITHMETIC OPERATOR:

Q) select customer_username, 2*total_amount from table_7_3nf;

CODE:

Query Editor   Query History

```
1
2   select customer_username, 2*total_amount from table_7_3nf;
```

OUTPUT:

Data Output   Messages   Explain   Notifications

	customer_username character varying (30)	?column? numeric
1	nihaas	120000
2	nihaasreddy	120000
3	ravi	118000
4	kaushik	138000
5	kartheek	58000
6	karthik	58000
7	pratyush	104000
8	raviteja	160000
9	aqib	120000
10	nikhil	174000
11	sarthak	242800
12	abhi	74000

Q) select * from table_3_3nf where customer_username like 'a%';

CODE:

Query Editor	Query History

```
1
2   select * from table_3_3nf where customer_username like 'a%';
```

OUTPUT:

Data Output	Messages	Explain	Notifications

	customer_username character varying (30)	customer_password character varying (30)	customer_fullname character varying (30)	customer_phone_no numeric (10)	customer_email character varying (30)
1	aqib	aqib123	shaik aquibuddin	9998979695	shaik@abc.com
2	abhi	abhi123	abhi reddy	9998979695	abhi@abc.com

TOCHAR:-

Q) select TO_CHAR(total_amount, '₹9,99,999')total_amount from table_7_3nf;

CODE:

```
1
2
3   select TO_CHAR(total_amount, '₹9,99,999')total_amount from table_7_3nf;
4
```

OUTPUT:

	total_amount text
1	₹ 60,000
2	₹ 60,000
3	₹ 59,000
4	₹ 69,000
5	₹ 29,000
6	₹ 29,000
7	₹ 52,000
8	₹ 80,000
9	₹ 60,000
10	₹ 87,000
11	₹ 1,21,400
12	₹ 37,000

EXTRACT:

cannot be used because date column nhi not available --

SYNTAX - SELECT EXTRACT(WEEK FROM "2017-06-15");

BETWEEN:-

Q) select customer_username, total_amount from table_7_3nf where total_amount BETWEEN 80000 and 200000 ;

CODE:

```
Query Editor    Query History
1
2   select customer_username, total_amount from table_7_3nf where total_amount BETWEEN 80000 and 200000 ;
3
```

OUTPUT:

	customer_username character varying (30)	total_amount numeric (10)
1	raviteja	80000
2	nikhil	87000
3	sarthak	121400

NOT BETWEEN:

Q) select customer_username, total_amount from table_7_3nf where total_amount not BETWEEN 80000 and 200000 ;

CODE:

```
Query Editor   Query History
1
2  select customer_username, total_amount from table_7_3nf where total_amount not BETWEEN 80000 and 200000 ;
```

OUTPUT:

Data Output	Messages	Explain	Notifications

	customer_username character varying (30)	total_amount numeric (10)
1	nihaas	60000
2	nihaasreddy	60000
3	ravi	59000
4	kaushik	69000
5	kartheek	29000
6	karthik	29000
7	pratyush	52000
8	aqib	60000
9	abhi	37000

IN:

Q) select * from table_5_3nf where venue in ('patna', 'delhi');

CODE:

```
Query Editor   Query History
1
2  select * from table_5_3nf where venue in ('patna', 'delhi');
```

OUTPUT:

Data Output | Messages | Explain | Notifications

	customer_username character varying (30)	venue character varying (30)	venue_price numeric (10)
1	nihaas	patna	5000
2	nihaasreddy	patna	5000
3	ravi	delhi	15000
4	raviteja	delhi	15000
5	sarthak	delhi	15000
6	abhi	delhi	15000

NOT IN:

Q) select * from table_5_3nf where venue not in ('patna', 'delhi');

CODE:

Query Editor | Query History

```
1
2   select * from table_5_3nf where venue not in ('patna', 'delhi');
```

OUTPUT:

Data Output | Messages | Explain | Notifications

	customer_username character varying (30)	venue character varying (30)	venue_price numeric (10)
1	kaushik	hyderabad	10000
2	kartheek	chennai	12000
3	karthik	chennai	12000
4	pratyush	mumbai	18000
5	aqib	hyderabad	10000
6	nikhil	mumbai	18000

UNION:

Q) select * from table_5_3nf

union
select * from table_6_3nf;
CODE:

```
1
2    select * from table_5_3nf
3    union
4    select * from table_6_3nf;
```

OUTPUT:

Data Output   Messages   Explain   Notifications

	customer_username character varying (30)	venue character varying (30)	venue_price numeric (10)
1	nihaas	patna	5000
2	ravi	delhi	15000
3	karthik	chennai	12000
4	kaushik	hyderabad	10000
5	ravi	south indian food	70
6	nihaasreddy	east indian food	60
7	aqib	north indian food	100
8	aqib	hyderabad	10000
9	nikhil	north indian food	100
10	pratyush	east indian food	60
11	abhi	delhi	15000
12	kartheek	north indian food	100

<mark>**EXISTS:**</mark>

Q) select * from table_5_3nf
where exists(select venue from table_5_3nf where venue = 'patna');

CODE:

```
3
4  select * from table_5_3nf
5  where  exists(select venue from table_5_3nf where venue = 'patna');
6
```

OUTPUT:

	customer_username character varying (30)	venue character varying (30)	venue_price numeric (10)
1	nihaas	patna	5000
2	nihaasreddy	patna	5000
3	ravi	delhi	15000
4	kaushik	hyderabad	10000
5	kartheek	chennai	12000
6	karthik	chennai	12000
7	pratyush	mumbai	18000
8	raviteja	delhi	15000
9	aqib	hyderabad	10000
10	nikhil	mumbai	18000
11	sarthak	delhi	15000
12	abhi	delhi	15000

NOT EXISTS:

Q) select * from table_5_3nf
where not exists(select venue from table_5_3nf where venue = 'patna');

CODE:

Query Editor   Query History

```
1
2
3
4  select * from table_5_3nf
5  where not exists(select venue from table_5_3nf where venue = 'patna');
6
```

OUTPUT:

customer_username character varying (30)	venue character varying (30)	venue_price numeric (10)

Q) select * from table_5_3nf
where venue = all(select venue from table_5_3nf where venue = 'patna');

CODE:

```
Query Editor    Query History
1  |
2  select * from table_5_3nf
3  where venue = all( select venue from table_5_3nf where venue = 'patna');
4
```

OUTPUT:

Data Output   Messages   Explain   Notifications

	customer_username character varying (30)	venue character varying (30)	venue_price numeric (10)
1	nihaas	patna	5000
2	nihaasreddy	patna	5000

Q) select * from table_5_3nf
where venue = any(select venue from table_5_3nf where venue = 'chennai');

CODE:

```
Query Editor    Query History
1
2  select * from table_5_3nf
3  where venue = any( select venue from table_5_3nf where venue = 'chennai');
4
```

OUTPUT:

	customer_username character varying (30)	venue character varying (30)	venue_price numeric (10)
1	kartheek	chennai	12000
2	karthik	chennai	12000

CONCLUSION

Thus we have successfully implemented online event registration database system which helps users to register for a particular event and he can keep track of his bookings. We have successfully implemented various functionalities of MySQL and created the fully functional database management System for Event Registration.

REFERENCES

MySQL references :

- https://www.w3schools.in/mysql/