

DATABASE MANAGEMENT SYSTEM

PROJECT REPORT

TOPIC: RESTAURANT MANAGEMENT SYSTEM

TEAM COMPOSITION:

NAME	SRN
VAIDHOORYA	PES1UG20EC274
VAISHNAVI KULKARNI	PES1UG20EC275
AMULYA A SHETTY	PES1UG20EC284

INTRODUCTION:

People nowadays hardly want to move from their places for shopping and not even for eating this is because of their hectic schedule. In such circumstances, we have decided to create a system for restaurants. This system is well known as Restaurant Management Services. It is a multi-functioned software that helps a food business or restaurant owner to operate the business effectively.

Restaurant Management System is about ordering online food from favourite restaurants. Anybody can create the account and order the food from the website , if he is already a customer he can login to his account using the login credentials which is stored in database. The Admin can add and remove the food items, restaurants, etc.

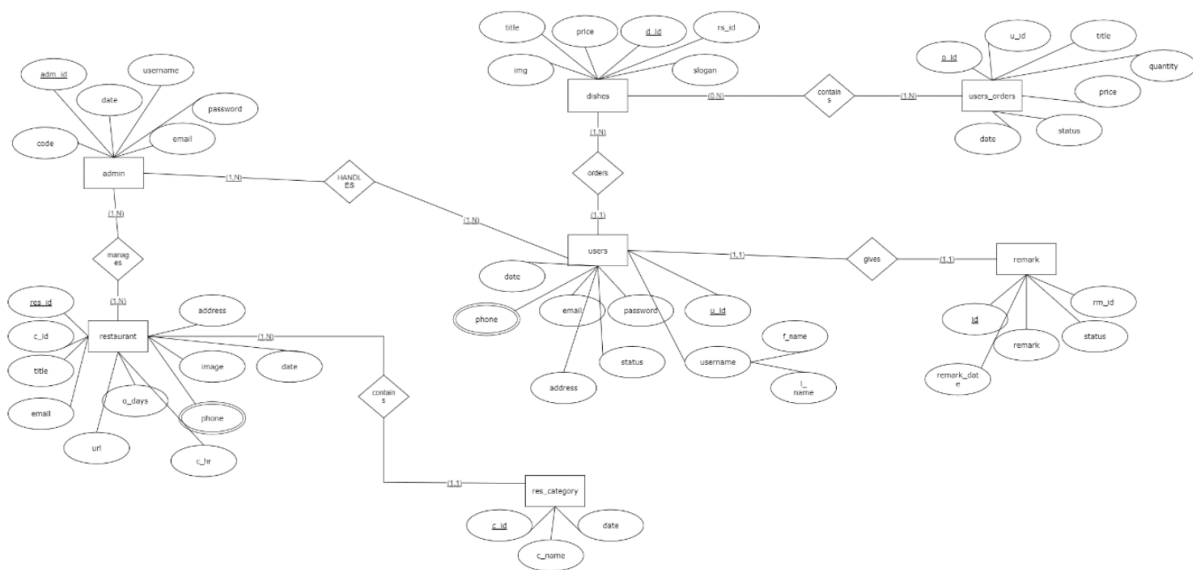
SCOPE:

A restaurant management system is a multi-functioned software that helps a food business or restaurant owner to operate the business effectively.

This project is helpful for both user as well as restaurant manager , it is user friendly as it allows the user to order their favourite dishes from their favourite restaurants and order as many as dishes the user wants and the payment process is very simple to make .the user can create their account or if they already have their account they can login using their login credentials and after ordering the user gets their order summary(bill).

In case of admin, he can login using his username and password , admin can add the dishes or delete the dishes or he can change the dishes as well as he can manage the users and different restaurants , billing system, etc.

ER DIAGRAM:



RELATIONAL SCHEMA:



DDL STATEMENTS TO BUILD THE DATABASE:

TO CREATE TABLE ADMIN:

CREATE TABLE admin

```
(    adm_id int(222) NOT NULL,  
    username varchar(222) NOT NULL,  
    password varchar(222) NOT NULL,  
    email varchar(222) NOT NULL,  
    code varchar(222) NOT NULL,  
    date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()  
)ENGINE=InnoDB DEFAULT CHARSET=latin1
```

```
1 CREATE TABLE admin  
2 (    adm_id int(222) NOT NULL,  
3    username varchar(222) NOT NULL,  
4    password varchar(222) NOT NULL,  
5    email varchar(222) NOT NULL,  
6    code varchar(222) NOT NULL,  
7    date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()  
8 )ENGINE=InnoDB DEFAULT CHARSET=latin1  
9  
10 |
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE admin ( adm_id int(222) NOT NULL, username varchar(222) NOT NULL, password varchar(222) NOT NULL, email varchar(222) NOT NULL, code varchar(222) NOT NULL,  
date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() )ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

TO CREATE TABLE DISHES:

CREATE TABLE dishes

```
(  
d_id int(222) NOT NULL,  
rs_id int(222) NOT NULL,  
title varchar(222) NOT NULL,  
slogan varchar(222) NOT NULL,  
price decimal(10,2) NOT NULL,  
img varchar(222) NOT NULL  
)ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```

1 CREATE TABLE dishes
2 (
3   d_id int(222) NOT NULL,
4   rs_id int(222) NOT NULL,
5   title varchar(222) NOT NULL,
6   slogan varchar(222) NOT NULL,
7   price decimal(10,2) NOT NULL,
8   img varchar(222) NOT NULL
9 )ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```

CREATE TABLE dishes ( d_id int(222) NOT NULL, rs_id int(222) NOT NULL, title varchar(222) NOT NULL, slogan varchar(222) NOT NULL, price decimal(10,2) NOT NULL, img
varchar(222) NOT NULL )ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

TO CREATE TABLE REMARK:

```

CREATE TABLE remark (
  id int(11) NOT NULL,
  frm_id int(11) NOT NULL,
  status varchar(255) NOT NULL,
  remark mediumtext NOT NULL,
  remarkDate timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

1 CREATE TABLE remark (
2   id int(11) NOT NULL,
3   frm_id int(11) NOT NULL,
4   status varchar(255) NOT NULL,
5   remark mediumtext NOT NULL,
6   remarkDate timestamp NOT NULL DEFAULT current_timestamp()
7 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```

CREATE TABLE remark ( id int(11) NOT NULL, frm_id int(11) NOT NULL, status varchar(255) NOT NULL, remark mediumtext NOT NULL, remarkDate timestamp NOT NULL DEFAULT
current_timestamp() ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

TO CREATE TABLE RESTAURANT:

```

CREATE TABLE restaurant (
  rs_id int(222) NOT NULL,
  c_id int(222) NOT NULL,
  title varchar(222) NOT NULL,
  email varchar(222) NOT NULL,
  phone varchar(222) NOT NULL,
  url varchar(222) NOT NULL,
  o_hr varchar(222) NOT NULL,
  c_hr varchar(222) NOT NULL,
  o_days varchar(222) NOT NULL,
  address text NOT NULL,
  image text NOT NULL,
  date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

1 CREATE TABLE restaurant (
2   rs_id int(222) NOT NULL,
3   c_id int(222) NOT NULL,
4   title varchar(222) NOT NULL,
5   email varchar(222) NOT NULL,
6   phone varchar(222) NOT NULL,
7   url varchar(222) NOT NULL,
8   o_hr varchar(222) NOT NULL,
9   c_hr varchar(222) NOT NULL,
10  o_days varchar(222) NOT NULL,
11  address text NOT NULL,
12  image text NOT NULL,
13  date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
14 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
15 |

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

```

CREATE TABLE restaurant ( rs_id int(222) NOT NULL, c_id int(222) NOT NULL, title varchar(222) NOT NULL, email varchar(222) NOT NULL, phone varchar(222) NOT NULL, url
varchar(222) NOT NULL, o_hr varchar(222) NOT NULL, c_hr varchar(222) NOT NULL, o_days varchar(222) NOT NULL, address text NOT NULL, image text NOT NULL, date timestamp
NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

TO CREATE TABLE RES_CATEGORY:

```

\CREATE TABLE res_category (
  c_id int(222) NOT NULL,
  c_name varchar(222) NOT NULL,
  date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

1 CREATE TABLE res_category (
2   c_id int(222) NOT NULL,
3   c_name varchar(222) NOT NULL,
4   date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
5 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```

CREATE TABLE res_category ( c_id int(222) NOT NULL, c_name varchar(222) NOT NULL, date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() )
ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

TO CREATE TABLE USERS:

```

CREATE TABLE users (
  u_id int(222) NOT NULL,
  username varchar(222) NOT NULL,
  f_name varchar(222) NOT NULL,
  l_name varchar(222) NOT NULL,
  email varchar(222) NOT NULL,
  phone varchar(222) NOT NULL,
  password varchar(222) NOT NULL,
  address text NOT NULL,
  status int(222) NOT NULL DEFAULT 1,
  date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

1 CREATE TABLE users (
2   u_id int(222) NOT NULL,
3   username varchar(222) NOT NULL,
4   f_name varchar(222) NOT NULL,
5   l_name varchar(222) NOT NULL,
6   email varchar(222) NOT NULL,
7   phone varchar(222) NOT NULL,
8   password varchar(222) NOT NULL,
9   address text NOT NULL,
10  status int(222) NOT NULL DEFAULT 1,
11  date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
12 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```

CREATE TABLE users ( u_id int(222) NOT NULL, username varchar(222) NOT NULL, f_name varchar(222) NOT NULL, l_name varchar(222) NOT NULL, email varchar(222) NOT NULL,
phone varchar(222) NOT NULL, password varchar(222) NOT NULL, address text NOT NULL, status int(222) NOT NULL DEFAULT 1, date timestamp NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp() ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

TO CREATE TABLE USERS_ORDERS:

```

CREATE TABLE users_orders (

```

```
o_id int(222) NOT NULL,  
u_id int(222) NOT NULL,  
title varchar(222) NOT NULL,  
quantity int(222) NOT NULL,  
price decimal(10,2) NOT NULL,  
status varchar(222) DEFAULT NULL,  
date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
1 CREATE TABLE users_orders (  
2   o_id int(222) NOT NULL,  
3   u_id int(222) NOT NULL,  
4   title varchar(222) NOT NULL,  
5   quantity int(222) NOT NULL,  
6   price decimal(10,2) NOT NULL,  
7   status varchar(222) DEFAULT NULL,  
8   date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()  
9 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE users_orders ( o_id int(222) NOT NULL, u_id int(222) NOT NULL, title varchar(222) NOT NULL, quantity int(222) NOT NULL, price decimal(10,2) NOT NULL,  
status varchar(222) DEFAULT NULL, date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp() ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

ALTER TABLE:

```
ALTER TABLE admin  
ADD PRIMARY KEY (adm_id);
```

```
1 ALTER TABLE admin  
2 ADD PRIMARY KEY (adm_id);  
3 |
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
ALTER TABLE admin ADD PRIMARY KEY (adm_id);
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

```
ALTER TABLE `dishes`  
ADD PRIMARY KEY (`d_id`);
```

```
1 ALTER TABLE `dishes`  
2 ADD PRIMARY KEY (`d_id`);  
3 |
```


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
ALTER TABLE `dishes` ADD PRIMARY KEY (`d_id`);
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

```
ALTER TABLE `remark`  
ADD PRIMARY KEY (`id`);
```

```
1 ALTER TABLE `remark`  
2 ADD PRIMARY KEY (`id`);
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
ALTER TABLE `remark` ADD PRIMARY KEY (`id`);
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

```
ALTER TABLE res_category  
ADD PRIMARY KEY (`c_id`);
```

```
1 ALTER TABLE restaurant  
2 ADD PRIMARY KEY (`rs_id`);  
3
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

```
ALTER TABLE restaurant ADD PRIMARY KEY (`rs_id`);
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

```
ALTER TABLE users  
ADD PRIMARY KEY (`u_id`);
```

```
1 ALTER TABLE users  
2 ADD PRIMARY KEY (`u_id`);  
3 |
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
ALTER TABLE users ADD PRIMARY KEY (`u_id`);
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

```
ALTER TABLE users_orders  
ADD PRIMARY KEY (`o_id`);
```

```
1 ALTER TABLE users_orders  
2 ADD PRIMARY KEY (`o_id`);
```

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

ALTER TABLE users_orders ADD PRIMARY KEY (`o_id`);

[ Edit inline ] [ Edit ] [ Create PHP code ]
```

INSERTING DISHES:

```
INSERT INTO `dishes` (`d_id`, `rs_id`, `title`, `slogan`, `price`, `img`) VALUES
(1, 1, 'Paneer Saag', 'Cottage cheese cooked with fresh ground spinach, onion, garlic and indian herbs.',
'390.00', '606d72f3cb12f.jpg'),
(2, 1, 'Chicken Achari', 'Dahi, achar, tomatoes, mustard oil, fennel seeds', '250.00', '606d73302ece2.jpg'),
(3, 1, 'Murgh Tikka Biryani', 'Chicken Tikka Biryani Recipe is the famous Indian delicacy of Punjabi Cuisine.
Boneless Chicken is marinated in spicy mix of Yogurt and Spices.', '470.00', '606d73771366a.jpg'),
(4, 1, 'Murgh Zafrani Kabab', 'Murgh Zafrani Kabab. A plateful of succulent pieces of meat, straight out of
tandoor, is tough to resist.', '380.00', '606d73d2d37f4.jpg'),
(6, 2, 'Cheesy Mashed Potato', 'Covered with mozzarella cheese.', '250.00', '606d74c416da5.jpg'),
(7, 2, 'Crispy Chicken Strips', 'Fried chicken strips, served with special honey mustard sauce.', '460.00',
'606d74f6ecbbb.jpg'),
(8, 2, 'Lemon Grilled Chicken And Pasta', 'Marinated rosemary grilled chicken breast served with mashed
potatoes and your choice of pasta.', '380.00', '606d752a209c3.jpg'),
(9, 3, 'Vegetable Fried Rice', 'Chinese rice wok with cabbage, beans, carrots, and spring onions.', '350.00',
'606d7575798fb.jpg'),
(10, 3, 'Prawn Crackers', '12 pieces deep-fried prawn crackers', '120.00', '606d75a7e21ec.jpg'),
(11, 3, 'Spring Rolls', 'Lightly seasoned shredded cabbage, onion and carrots, wrapped in house made
spring roll wrappers, deep fried to golden brown.', '470.00', '606d75ce105d0.jpg'),
(12, 3, 'Manchurian Chicken', 'Chicken pieces slow cooked with spring onions in our house made
manchurian style sauce.', '500.00', '606d7600dc54c.jpg'),
(13, 4, 'Buffalo Wings', 'Fried chicken wings tossed in spicy Buffalo sauce served with crisp celery sticks
and Blue cheese dip.', '450.00', '606d765f69a19.jpg'),
(14, 4, 'Mac N Cheese Bites', 'Served with our traditional spicy queso and marinara sauce.', '350.00',
'606d768a1b2a1.jpg'),
(16, 4, 'Meatballs Penne Pasta', 'Garlic-herb beef meatballs tossed in our house-made marinara sauce and
penne pasta topped with fresh parsley.', '470.00', '606d76eedbb99.jpg'),
(17, 2, 'French Fries', 'crispy spicy fried potato ', '130.00', '637f3c0c05ed1.jpg');
```

```
✓ 16 rows inserted. (Query took 0.0006 seconds.)

INSERT INTO `dishes` (`d_id`, `rs_id`, `title`, `slogan`, `price`, `img`) VALUES (1, 1, 'Paneer Saag', 'Cottage cheese cooked with fresh
ground spinach, onion, garlic and indian herbs.', '390.00', '606d72f3cb12f.jpg'), (2, 1, 'Chicken Achari', 'Dahi, achar, tomatoes,
mustard oil, fennel seeds', '250.00', '606d73302ece2.jpg'), (3, 1, 'Murgh Tikka Biryani', 'Chicken Tikka Biryani Recipe is the famous
Indian delicacy of Punjabi Cuisine. Boneless Chicken is marinated in spicy mix of Yogurt and Spices.', '470.00', '606d73771366a.jpg'),
(4, 1, 'Murgh Zafrani Kabab', 'Murgh Zafrani Kabab. A plateful of succulent pieces of meat, straight out of tandoor, is tough to
resist.', '380.00', '606d73d2d37f4.jpg'), (5, 2, 'Pink Spaghetti Gamberoni', 'Spaghetti pasta, grilled shrimps, parmesan cheese, with
[ Edit ]
```

INSERTING USER ORDERS:

```
INSERT INTO users_orders (o_id, u_id, title, quantity, price, status, date) VALUES
```

```
(10, 40, 'Murgh Tikka Biryani', 1, '470.00', NULL, '2023-05-024 08:43:49'),
(11, 40, 'Paneer Saag', 1, '390.00', NULL, '2023-05-024 08:43:49'),
(12, 40, 'Crispy Chicken Strips', 1, '460.00', NULL, '2023-05-024 08:43:49'),
(13, 40, 'Cheesy Mashed Potato', 1, '250.00', NULL, '2023-05-024 08:43:49'),
(14, 41, 'Murgh Tikka Biryani', 1, '470.00', NULL, '2023-05-024 08:43:49'),
(15, 41, 'Paneer Saag', 1, '390.00', NULL, '2023-05-024 08:43:49'),
(16, 42, 'Crispy Chicken Strips', 1, '460.00', NULL, '2023-05-024 08:43:49'),
(17, 42, 'Cheesy Mashed Potato', 1, '250.00', NULL, '2023-05-024 08:43:49'),
(18, 42, 'Pink Spaghetti Gamberoni', 1, '380.00', NULL, '2023-05-024 08:43:49'),
(19, 43, 'Crispy Chicken Strips', 1, '460.00', NULL, '2023-05-024 08:43:49'),
(20, 43, 'Pink Spaghetti Gamberoni', 1, '380.00', NULL, '2023-05-024 08:43:49'),
(21, 43, 'Lemon Grilled Chicken And Pasta', 1, '380.00', NULL, '2023-05-024 08:43:49'),
(45, 42, 'Paneer Saag', 2, '780.00', 'NULL', '0000-00-00 00:00:00'),
(49, 56, 'Murgh Tikka Biryani', 2, '780.00', 'NULL', '0000-00-00 00:00:00'),
(55, 52, 'Paneer Saag', 2, '780.00', 'NULL', '0000-00-00 00:00:00'),
(78, 78, 'Murgh Tikka Biryani', 2, '780.00', 'NULL', '0000-00-00 00:00:00');
```

INSERTING ADMIN:

```
*INSERT INTO admin (adm_id, username, password, email, code, date) VALUES
(1, 'admin', 'admin123', 'admin@gmail.com', '', '2023-05-23 08:40:28');
```

```
*INSERT INTO admin (adm_id, username, password, email, code, date) VALUES
(1, 'admin', 'admin123', 'admin@gmail.com', '', '2023-05-23 08:40:28');
```

✓ 1 row inserted. (Query took 0.0004 seconds.)

```
INSERT INTO `admin` (`adm_id`, `username`, `password`, `email`, `code`, `date`) VALUES (1, 'admin', '0192023a7bbd73250516f069df18b500',
'admin@gmail.com', '', '2021-04-07 08:40:28');
```

[Edit inline](#) [\[Edit \]](#) [\[Create PHP code \]](#)

INSERTING VALUES USING ONLY VALUES:

```
INSERT INTO res_category VALUES (1, 'Indian', '2023-05-23 08:45:20'), (2, 'Italian', '2023-05-23
08:45:20'), (3, 'Chinese', '2023-05-23 08:45:20'), (4, 'American', '2023-05-23 08:45:20');
```

```
1 INSERT INTO res_category VALUES (1, 'Indian', '2023-05-23 08:45:20'), (2, 'Italian', '2023-05-23
08:45:20'), (3, 'Chinese', '2023-05-23 08:45:20'), (4, 'American', '2023-05-23 08:45:20');
```

2

3

✓ 4 rows inserted. (Query took 0.0037 seconds.)

```
INSERT INTO `res_category` VALUES (1, 'Indian', '2021-04-07 08:45:20'), (2, 'Italian', '2021-04-07 08:45:23'), (3, 'Chinese', '2021-04-07
08:45:25'), (4, 'American', '2021-04-07 08:45:28');
```

[Edit inline](#) [\[Edit \]](#) [\[Create PHP code \]](#)

INSERTING USERS :

```
INSERT INTO 'users' (`u_id`, `username`, `f_name`, `l_name`, `email`, `phone`, `password`, `address`,  
`status`, `date`) VALUES  
(1, 'Suj', 'Sujatha', 'kulkarni', 'sujatha@gmail.com', '1324325445', 'sujatha123', 'Mira Road', 1, '2023-05-024  
08:43:49'),  
(2, 'PAR', 'Parth', 'Desai', 'parth@gmail.com', '4325345332', 'parth123', 'Vasai', 1, '2023-05-024 08:43:49'),
```

(10, 'DHE', 'dheekshith', 'naik', 'dheek@gmail.com', '8296805117', 'dheek88', 'mysore', 0, '2023-05-024 08:43:49'),
(11, 'MON', 'monika', 'naik', 'monika64@gmail.com', '9609645366', 'monika468', 'banglore', 0, '2023-05-024 08:43:49'),
(12, 'CHI', 'chirag', 'patel', 'chirag@gmail.com', '9845832567', 'chirag34', 'tumkur', 0, '2023-05-024 08:43:49'),
(13, 'ARU', 'arushi', 'singh', 'arushi56@gmail.com', '8764987493', 'arushi55', 'banglore', 0, '2023-05-024 08:43:49'),
(14, 'ANA', 'ananya', 'pandey', 'ananya99@gmail.com', '9687528024', 'ananya456', 'tumkur', 0, '2023-05-024 08:43:49'),
(15, 'VIH', 'vihar', 'lal', 'vihar567@gmail.com', '9434780884', 'vihar998', 'banglore', 0, '2023-05-024 08:43:49'),
(16, 'SOU', 'sourabh', 'kumar', 'sourabh789@gmail.com', '9643576198', 'sour456', 'mysore', 0, '2023-05-024 08:43:49'),
(18, 'NIK', 'nikhil', 'goel', 'nikhil123@gmail.com', '1234567891', 'nikhil123', 'banglore', 0, '0000-00-00 00:00:00'),
(21, 'username', 'f_name', 'l_name', 'email', 'phone', 'password', 'address', 0, '0000-00-00 00:00:00'),
(23, 'NAY', 'nayana', 'shetty', 'nayana86@gmail.com', '9986853436', 'nayana34', 'tumkur', 0, '0000-00-00 00:00:00'),
(28, 'RAJ', 'rajveer', 'singh', 'rajveer12@gmail.com', '2345675634', 'raj124', 'banglore', 0, '0000-00-00 00:00:00'),
(32, 'JOY', 'joseph', 'paul', 'joseph@gmail.com', '8765467845', 'joseph78', 'mysore', 0, '0000-00-00 00:00:00'),
(33, 'AKA', 'akash', 'mehta', 'akash67@gmail.com', '9876545567', 'akash11', 'tumkur', 0, '0000-00-00 00:00:00'),
(38, 'NEH', 'neha', 'sharma', 'neha34@gmail.com', '8796788678', 'neha898', 'banglore', 0, '0000-00-00 00:00:00'),
(39, 'rishika', 'rishika', 'kulkarni', 'rishika123@gmail.com', '9866875532', 'rishika123', 'banglore', 1, '2023-05-024 08:43:49'),
(40, 'Sinch', 'sinchana', 'd', 'sinchana123@gmail.com', '9866784566', 'sinch123', 'banglore', 1, '2023-05-024 08:43:49'),
(41, 'shrav', 'shravani', 'p', 'shravani@gmail.com', '88967552347', 'shrav123', 'mysore', 1, '2023-05-024 08:43:49'),
(42, 'Sneha', 'sneha', 'priya', 'sneha@gmail.com', '8966443356', 'sneha123', 'Tumkur', 1, '2023-05-024 08:43:49'),
(43, 'LEXI', 'LEXI', 'RIVERA', 'lexi@gmail.com', '6534567898', 'lexi123', 'chitradurga', 1, '2023-05-024 08:43:49'),
(44, 'vaish', 'vaishnavi', 'r', 'vaishnavi@gmail.com', '9987876654', 'vaish123', 'banglore', 1, '2023-05-024 08:43:49'),
(45, 'rac', 'rachitha', 'singh', 'rachitha@gmail.com', '9986854321', 'rachita123', 'banglore', 1, '2023-05-024 08:43:49');

```

1 INSERT INTO users (u_id, username, f_name, l_name, email, phone, password, address, status, date)
  VALUES
2 (1, 'Suj', 'Sujatha', 'kulkarni', 'sujatha@gmail.com', '1324325445', 'sujatha123', 'Mira Road', 1,
  '2023-05-024 08:43:49'),
3 (2, 'PAR', 'Parth', 'Desai', 'parth@gmail.com', '4325345332', 'parth123', 'Vasai', 1, '2023-05-024
  08:43:49'),
4 (10, 'DHE', 'dheekshith', 'naik', 'dheek@gmail.com', '8296805117', 'dheek88', 'mysore', 0, '2023-05-
  024 08:43:49'),
5 (11, 'MON', 'monika', 'naik', 'monika64@gmail.com', '9609645366', 'monika468', 'bangalore', 0, '2023-
  05-024 08:43:49'),
6 (12, 'CHI', 'chirag', 'patel', 'chirag@gmail.com', '9845832567', 'chirag34', 'tumkur', 0, '2023-05-024
  08:43:49'),
7 (13, 'ARU', 'arushi', 'singh', 'arushi56@gmail.com', '8764987493', 'arushi55', 'bangalore', 0, '2023-
  05-024 08:43:49'),
8 (14, 'ANA', 'ananya', 'pandey', 'ananya99@gmail.com', '9687528024', 'ananya456', 'tumkur', 0, '2023-
  05-024 08:43:49'),

```

✓ 22 rows inserted. (Query took 0.0011 seconds.)

```

INSERT INTO users (u_id, username, f_name, l_name, email, phone, password, address, status, date) VALUES (1, 'Suj', 'Sujatha',
'kulkarni', 'sujatha@gmail.com', '1324325445', 'sujatha123', 'Mira Road', 1, '2023-05-024 08:43:49'), (2, 'PAR', 'Parth', 'Desai',
'parth@gmail.com', '4325345332', 'parth123', 'Vasai', 1, '2023-05-024 08:43:49'), (10, 'DHE', 'dheekshith', 'naik', 'dheek@gmail.com',
'8296805117', 'dheek88', 'mysore', 0, '2023-05-024 08:43:49'), (11, 'MON', 'monika', 'naik', 'monika64@gmail.com', '9609645366',
'monika468', 'bangalore', 0, '2023-05-024 08:43:49'), (12, 'CHI', 'chirag', 'patel', 'chirag@gmail.com', '9845832567', 'chirag34',
'tumkur', 0, '2023-05-024 08:43:49'), (13, 'ARU', 'arushi', 'singh', 'arushi56@gmail.com', '8764987493', 'arushi55', 'bangalore', 0,
'tumkur', 0, '2023-05-024 08:43:49'), (14, 'ANA', 'ananya', 'pandey', 'ananya99@gmail.com', '9687528024', 'ananya456', 'tumkur', 0,
'tumkur', 0, '2023-05-024 08:43:49'),

```

JOIN STATEMENTS:

INNER JOIN:

Get the users information who have ordered the dish 'paneer saag'

```

SELECT users.u_id,users.username,users.email,users.address
FROM users inner join users_orders on users.u_id = users_orders.u_id
WHERE users_orders.title = 'paneer saag';

```

```

1 SELECT users.u_id,users.username,users.email,users.address
2 FROM users inner join users_orders on users.u_id = users_orders.u_id
3 WHERE users_orders.title = 'paneer saag';

```

✓ Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```

SELECT users.u_id,users.username,users.email,users.address FROM users inner join users_orders on users.u_id = users_orders.u_id WHERE users_orders.title = 'paneer
saag';

```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

u_id	username	email	address
40	Sinch	sinchana123@gmail.com	bangalore
41	shrav	shravani@gmail.com	mysore
42	Sneha	sneha@gmail.com	Tumkur

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

LEFT OUTER JOIN:

Retrieve category id from res_category and restaurant table

SELECT *

FROM res_category

LEFT OUTER JOIN restaurant ON res_category.c_id = restaurant.c_id

```
1 SELECT *
2 FROM res_category
3 LEFT OUTER JOIN restaurant ON res_category.c_id = restaurant.c_id
```

c_id	c_name	date	rs_id	c_id	title	email	phone	url	o_hr	c_hr	o_days	address	image	dis
1	Indian	2023-05-24 08:43:49	5	1	GJB Cafetaria	pesuniversity@gmail.com	9900885927	gjbc.in	8am	6pm	mon-sat	PES University , RR Campus	646da6fab9918.jpg	2023-05-24 08:43:49
1	Indian	2023-05-24 08:43:49	6	1	Hallimane	hallimane@gmail.com	9900990099	hallimane.com	8am	6pm	Mon-Sat	Pes university	646da6a7c38b8.jpg	2023-05-24 08:43:49
2	Italian	2023-05-24 08:43:49	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	indo-chinese	2023-05-24 08:43:49	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	American-mexican	2023-05-24 08:43:49	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

NATURAL JOIN :

Retrive restaurant category using natural join:

SELECT *

FROM res_category

JOIN restaurant;

```
1 SELECT *
2 FROM res_category
3 JOIN restaurant;
```

Server: 127.0.0.1 » Database: dbms_project » Table: res_category														
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking More														
c_id	c_name	date	rs_id	c_id	title	email	phone	url	o_hr	c_hr	o_days	address	image	
1	Indian	2023-05-24 08:43:49	5	1	GJB Cafetaria	pesuniversity@gmail.com	9900885927	gjbc.in	8am	6pm	mon-sat	University, RR Campus	646da6fab9918.jpg	2
1	Indian	2023-05-24 08:43:49	6	1	Hallimane	hallimane@gmail.com	9900990099	hallimane.com	8am	6pm	Mon-Sat	Pes university	646da6a7c38b8.jpg	2
2	Italian	2023-05-24 08:43:49	5	1	GJB Cafetaria	pesuniversity@gmail.com	9900885927	gjbc.in	8am	6pm	mon-sat	PES University, RR Campus	646da6fab9918.jpg	2
2	Italian	2023-05-24 08:43:49	6	1	Hallimane	hallimane@gmail.com	9900990099	hallimane.com	8am	6pm	Mon-Sat	Pes university	646da6a7c38b8.jpg	2
3	indo-chinese	2023-05-24 08:43:49	5	1	GJB Cafetaria	pesuniversity@gmail.com	9900885927	gjbc.in	8am	6pm	mon-sat	PES University, RR Campus	646da6fab9918.jpg	2
3	indo-chinese	2023-05-24 08:43:49	6	1	Hallimane	hallimane@gmail.com	9900990099	hallimane.com	8am	6pm	Mon-Sat	Pes university	646da6a7c38b8.jpg	2

INNER JOIN:

Get the users information who have ordered murgh tikka biryani not in between '2022-10-28' and '2022-11-18' but between the date '2022-10-28' and '2022-11-25'

```
SELECT users.u_id,users.username,users.email,users.address,users.f_name,users.l_name
FROM users inner JOIN users_orders
ON users.u_id = users_orders.u_id
WHERE users_orders.title = 'Murgh Tikka Biryani' and users_orders.date not between '2022-10-28' and '2022-11-18'and users_orders.date between '2023-02-28' and '2023-11-25';
```

```
1 SELECT users.u_id,users.username,users.email,users.address,users.f_name,users.l_name
2 FROM users inner JOIN users_orders
3 ON users.u_id = users_orders.u_id
4 WHERE users_orders.title = 'Murgh Tikka Biryani' and users_orders.date not between '2022-10-28' and
5 '2022-11-18'and users_orders.date between '2023-02-28' and '2023-11-25';
```


✓ Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT users.u_id,users.username,users.email,users.address,users.f_name,users.l_name FROM users inner JOIN users_orders ON users.u_id = users_orders.u_id WHERE users_orders.title = 'Murgh Tikka Biryani' and users_orders.date not between '2022-10-28' and '2022-11-18'and users_orders.date between '2023-02-28' and '2023-11-25';
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: | Filter rows: | Sort by key:

Extra options

u_id	username	email	address	f_name	l_name
40	Sinch	sinchana123@gmail.com	banglore	sinchana	d
41	shrav	shravani@gmail.com	mysore	shravani	p
47	Amulya	amu@gmail.com	My home	Amulya	kulkarni

AGGREGATE FUNCTIONS:

COUNT

Display the number of price corresponding to the user_id(s) 1,2,40

SELECT U_ID, COUNT(price) FROM users_orders

WHERE U_ID IN (1,2,40)

GROUP BY U_ID;

```
1 SELECT U_ID, COUNT(price) FROM users_orders
2 WHERE U_ID IN (1,2,40)
3 GROUP BY U_ID;
4
```

Your SQL query has been executed successfully.

```
SELECT U_ID, COUNT(price) FROM users_orders WHERE U_ID IN (1,2,40) GROUP BY U_ID;
```

U_ID	COUNT(price)
40	4

AVERAGE:

Find the average price every dishes display the d_id and corresponding average price as 'AVG_PRICE' in decreasing order of AVG_PRICE:




SELECT d_id , AVG(price) AS 'AVG_PRICE' FROM dishes
GROUP BY d_id

ORDER BY AVG(price) DESC;

```
1 SELECT d_id , AVG(price) AS 'AVG_PRICE' FROM dishes
2 GROUP BY d_id
3 ORDER BY AVG(price) DESC;
```

✓ Showing rows 0 - 14 (15 total, Query took 0.0665 seconds.)

```
SELECT d_id , AVG(price) AS 'AVG_PRICE' FROM dishes GROUP BY d_id ORDER BY AVG(price) DESC;
```

				d_id	AVG_PRICE
<input type="checkbox"/>	 Edit	 Copy	 Delete	12	500.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	470.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	470.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	16	470.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	7	460.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	13	450.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	390.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	380.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	8	380.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	9	350.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	14	350.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	250.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	250.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	17	130.000000
<input type="checkbox"/>	 Edit	 Copy	 Delete	10	120.000000

MAX:

Find the dish which is having highest price.

```
SELECT *  
FROM DISHES  
WHERE PRICE = (SELECT MAX(PRICE) AS MAX_PRICE FROM DISHES);
```

```
1 SELECT *  
2 FROM DISHES  
3 WHERE PRICE = (SELECT MAX(PRICE) AS MAX_PRICE FROM DISHES);  
4
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0015 seconds.)

```
SELECT * FROM DISHES WHERE PRICE = (SELECT MAX(PRICE) AS MAX_PRICE FROM DISHES);
```

	d_id	rs_id	title	slogan	price	img
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	3	Manchurian Chicken	Chicken pieces slow cooked with spring onions in o...	500.00	606d7600dc54c.jpg

MIN:

Find the dish which is having lowest price.

```
SELECT *  
FROM DISHES  
WHERE PRICE = (SELECT MIN(PRICE) AS MIN_PRICE FROM DISHES);
```

```
1 SELECT *  
2 FROM DISHES  
3 WHERE PRICE = (SELECT MIN(PRICE) AS MIN_PRICE FROM DISHES);  
4
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0028 seconds.)

```
SELECT * FROM DISHES WHERE PRICE = (SELECT MIN(PRICE) AS MIN_PRICE FROM DISHES);
```

	d_id	rs_id	title	slogan	price	img
<input type="checkbox"/>			10	3 Prawn Crackers	12 pieces deep-fried prawn crackers	120.00 606d75a7e21ec.jpg

COUNT:

Count the number of users whose name consists of 'MA'

SELECT COUNT(*) FROM users WHERE USERNAME LIKE '%MA%';

```
1 SELECT COUNT(*) FROM users WHERE USERNAME LIKE '%MA%';
2
```

Your SQL query has been executed successfully.

```
SELECT COUNT(*) FROM users WHERE USERNAME LIKE '%MA%';
```

COUNT(*)

0

UNION:

(SELECT USERS.u_id , USERS.username , USERS.f_name,USERS.l_name,USERS.email
FROM users_orders

LEFT OUTER JOIN USERS ON users_orders.u_id = users.u_id

WHERE users_orders.TITLE = 'Paneer Saag')

UNION

(SELECT USERS.u_id , USERS.username , USERS.f_name,USERS.l_name,USERS.email
FROM users_orders

LEFT OUTER JOIN USERS ON users_orders.u_id = users.u_id

WHERE users_orders.TITLE = 'Chicken Achari')

```

1 (SELECT USERS.u_id , USERS.username , USERS.f_name,USERS.l_name,USERS.email
2 FROM users_orders
3 LEFT OUTER JOIN USERS ON users_orders.u_id = users.u_id
4 WHERE users_orders.TITLE = 'Paneer Saag')
5 UNION
6 (SELECT USERS.u_id , USERS.username , USERS.f_name,USERS.l_name,USERS.email
7 FROM users_orders
8 LEFT OUTER JOIN USERS ON users_orders.u_id = users.u_id
9 WHERE users_orders.TITLE = 'Chicken Achari')
10

```

u_id	username	f_name	l_name	email
40	Sinch	sinchana	d	sinchana123@gmail.com
41	shrav	shravani	p	shravani@gmail.com
42	Sneha	sneha	priya	sneha@gmail.com

INTERSECTION:

Find the list of users who have ordered cheesy mashed potato

```

select u.u_id, u.f_name, u.l_name
from users_orders as o, users as u
where o.u_id = u.u_id AND
o_id = 13 AND
EXISTS(SELECT title
FROM users_orders
WHERE o.price=price and title='Cheesy Mashed Potato')
AND
EXISTS(SELECT title
FROM users_orders
WHERE o.price=price and title='Cheesy Mashed Potato'

```

```

1 select u.u_id, u.f_name, u.l_name
2 from users_orders as o, users as u
3 where o.u_id = u.u_id AND
4 o_id = 13 AND
5 EXISTS(SELECT title
6 FROM users_orders
7 WHERE o.price=price and title='Cheesy Mashed Potato')
8 AND
9 EXISTS(SELECT title
10 FROM users_orders
11 WHERE o.price=price and title='Cheesy Mashed Potato')|

```

✓ Showing rows 0 - 0 (1 total, Query took 0.0068 seconds.)

`select u.u_id, u.f_name, u.l_name from users_orders as o, users as u where o.u_id = u.u_id AND o_id = 13 AND EXISTS(SELECT title FROM users_orders WHERE o.price=price and title='Cheesy Mashed Potato') AND EXISTS(SELECT title FROM users_orders WHERE o.price=price and title='Cheesy Mashed Potato');`

				u_id	f_name	l_name
				40	sinchana	d

EXIST

Find the list of users (u_id , First name and last name) who have ordered cheesy mashed potato

```

select u.u_id, u.f_name, u.l_name, o.o_id, o.title
from users_orders as o, users as u
where o.u_id = u.u_id AND
u.u_id = 40 AND
EXISTS(SELECT title
FROM users_orders
WHERE o.price=price and title='cheesy mashed potato')

```

```

1 select u.u_id, u.f_name, u.l_name, o.o_id, o.title
2 from users_orders as o, users as u
3 where o.u_id = u.u_id AND
4 u.u_id = 40 AND
5 EXISTS(SELECT title
6 FROM users_orders
7 WHERE o.price=price and title='cheesy mashed potato')

```

✓ Showing rows 0 - 0 (1 total, Query took 0.0018 seconds.)

```

select u.u_id, u.f_name, u.l_name, o.o_id, o.title from users_orders as o, users as u where o.u_id = u.u_id AND u.u_id = 40 AND EXISTS(SELECT title FROM users_orders WHERE
o.price=price and title='cheesy mashed potato');

```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

u_id	f_name	l_name	o_id	title
40	sinchana	d	13	Cheesy Mashed Potato

NOT EXIST

Find the list of users (u_id , First name and last name) who have not ordered cheesy mashed potato

```

select u.u_id, u.f_name, u.l_name, o.o_id, o.title
from users_orders as o, users as u
where o.u_id = u.u_id AND
u.u_id = 40 AND
not EXISTS(SELECT title
FROM users_orders
WHERE o.price=price and title='cheesy mashed potato')

```

```

1 select u.u_id, u.f_name, u.l_name, o.o_id, o.title
2 from users_orders as o, users as u
3 where o.u_id = u.u_id AND
4 u.u_id = 40 AND
5 not EXISTS(SELECT title
6 FROM users_orders
7 WHERE o.price=price and title='cheesy mashed potato')

```

✓ Showing rows 0 - 2 (3 total, Query took 0.0015 seconds.)

```

select u.u_id, u.f_name, u.l_name, o.o_id, o.title from users_orders as o, users as u where o.u_id = u.u_id AND u.u_id = 40 AND not EXISTS(SELECT title FROM users_orders
WHERE o.price=price and title='cheesy mashed potato');

```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

u_id	f_name	l_name	o_id	title
40	sinchana	d	10	Murgh Tikka Biryani
40	sinchana	d	11	Paneer Saag
40	sinchana	d	12	Crispy Chicken Strips

FUNCTIONS:

DELIMITER \$\$

CREATE FUNCTION orders(o_id varchar(255))

RETURNS varchar(100)

DETERMINISTIC

BEGIN

DECLARE cnt int;

DECLARE msg varchar(100);

set cnt=(select count(*) from users_orders as o where o.u_id=o_id);

if (cnt>2) then

set msg:= "cannot order current limit is over";

else

set msg:= "can order the dishes";

end if;

RETURN msg;

END; \$\$

DELIMITER ;


```

1 DELIMITER $$
2 CREATE FUNCTION orders(o_id varchar(255))
3 RETURNS varchar(100)
4 DETERMINISTIC
5 BEGIN
6 DECLARE cnt int;
7 DECLARE msg varchar(100);
8 set cnt=(select count(*) from users_orders as o where o.u_id=o_id);
9 if (cnt>2) then
10 set msg:= "cannot order current limit is over";
11 else
12 set msg:= "can order the dishes";
13 end if;
14 RETURN msg;
15 END; $$
16 DELIMITER ;
17
18 select u_id,price,orders(u_id) as STATUS from users_orders

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0066 seconds.)

```

CREATE FUNCTION orders(o_id varchar(255)) RETURNS varchar(100) DETERMINISTIC BEGIN DECLARE cnt int; DECLARE msg varchar(100); set cnt=
(select count(*) from users_orders as o where o.u_id=o_id); if (cnt>2) then set msg:= "cannot order current limit is over"; else set msg:=
"can order the dishes"; end if; RETURN msg; END;;

```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

select u_id,price,orders(u_id) as STATUS from users_orders

Server: 127.0.0.1 » Database: dbms_project » Table: users_orders						
<div> Browse Structure SQL Search Insert Export Import </div>						
			u_id	price	STATUS	
<input type="checkbox"/>		Edit		Copy		Delete
			40	470.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			40	390.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			40	460.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			40	250.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			41	470.00	can order the dishes	
<input type="checkbox"/>		Edit		Copy		Delete
			41	390.00	can order the dishes	
<input type="checkbox"/>		Edit		Copy		Delete
			42	460.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			42	250.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			42	380.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			43	460.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			43	380.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			43	380.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			42	780.00	cannot order current limit is over	
<input type="checkbox"/>		Edit		Copy		Delete
			56	780.00	can order the dishes	
<input type="checkbox"/>		Edit		Copy		Delete
			52	780.00	can order the dishes	

PROCEDURE:

```





1 DELIMITER $$
2 create PROCEDURE temp()
3 BEGIN
4 DECLARE _rollback BOOL default 0;
5 DECLARE CONTINUE HANDLER FOR 1051 SET _rollback=1;
6 DECLARE CONTINUE HANDLER FOR 1048 SET _rollback=1;
7 DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET _rollback=1;
8 START TRANSACTION;
9 INSERT into users_orders(o_id,u_id,title,quantity,price,status,date) values('','','','','','','','');
10 UPDATE users SET price = '' WHERE o_id = '' AND title = '' AND price = '';
11 INSERT into remark(u_id,id) values ('','');
12 if _rollback = 1 then
13 rollback;
14 else
15 commit;
16 end if;
17 END $$

```

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0092 seconds.)

create PROCEDURE temp() BEGIN DECLARE _rollback BOOL default 0; DECLARE CONTINUE HANDLER FOR 1051 SET _rollback=1; DECLARE CONTINUE
HANDLER FOR 1048 SET _rollback=1; DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET _rollback=1; START TRANSACTION; INSERT into
users_orders(o_id,u_id,title,quantity,price,status,date) values('','','','','','',''); UPDATE users SET price = '' WHERE o_id = ''
AND title = '' AND price = ''; INSERT into remark(u_id,id) values ('',''); -- throws error if pyamentInfo is incorrect if _rollback = 1
then rollback; else commit; end if; END;
```

[Edit inline] [Edit] [Create PHP code]

	Name	Type	Returns	
<input type="checkbox"/>	temp	PROCEDURE		 Edit  Execute  Export  Drop

TRIGGERS:

Write a trigger on insert to the 'users_orders' table if user tries to order order paneer saag show the msg "this is dish is currently unavailable"

BEFORE INSERT:

```
DELIMITER $$
CREATE OR REPLACE TRIGGER insert_user_order
BEFORE INSERT ON users_orders
FOR EACH ROW
BEGIN
IF NEW.title="Paneer Saag"
THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'this is dish is currently unavailable';
END IF;
END; $$
DELIMITER ;
```

```
1 DELIMITER $$
2 CREATE OR REPLACE TRIGGER insert_user_order
3 BEFORE INSERT ON users_orders
4 FOR EACH ROW
5 BEGIN
6 IF NEW.title="Paneer Saag" THEN
7 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'this is dish is currently unavailable';
8 END IF;
9 END; $$
10 DELIMITER ;
11
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0151 seconds.)

```
CREATE OR REPLACE TRIGGER insert_user_order BEFORE INSERT ON users_orders FOR EACH ROW BEGIN IF NEW.title="Paneer Saag" THEN SIGNAL  
SQLSTATE '45000' SET MESSAGE_TEXT = 'this is dish is currently unavailable'; END IF; END;;
```


[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

```
1 INSERT into users_orders values(49,56,'Paneer Saag',2,780,'NULL',2023-05-24);
```

Error

SQL query: [Copy](#)

```
INSERT into users_orders values(49,56,'Paneer Saag',2,780,'NULL',2023-05-24);
```

MySQL said: 

#1644 - this is dish is currently unavailable

```
CREATE TABLE backup_users (o_id int(10),u_id char(10),price decimal(6,2),date timestamp,  
title char(60), quantity int(80), username char(20),f_name char(20),  
l_name char(20), email varchar(20),phone int(20),password varchar(80),address char(20)  
);
```

CURSORS:

DELIMITER \$\$

```
create PROCEDURE delete_users_orders(IN u_id_val int(10))
```

```
BEGIN
```

```
declare o_id int(10);
```

```
declare price decimal(6,2);
```

```
declare date timestamp;
```

```
declare title char(60);
```

```
declare quantity int(80);
```

```
declare u_id char(10);
```

```
declare username char(20);
```

```
declare f_name char(20);
```

```
declare l_name char(20);
```

```
declare email varchar(20);
```

```
declare phone int(20);
```

```
declare password varchar(80);
```

```
declare address char(20);
```

```
declare done int default 0;
```

```
declare val int;
```

```
declare cur cursor for select O.o_id,O.u_id,O.price,O.date,O.title,O.quantity,  
B.username,B.f_name,B.l_name,B.email,B.phone,B.password,B.address from users_orders O,users B
```

```

WHERE O.u_id=B.u_id and O.u_id=u_id_val;
declare continue handler for not found set done=1;
open cur;
c1:LOOP
fetch cur into o_id,u_id,price,date,title,quantity,username,f_name,l_name,email,phone,password,address;
if done=1 then leave c1;
end if;
insert into backup_users values(8,39,470,2022-11-24,'Spring
Rolls',1,'rishika','rishika','kulkarni','rishika123@gmail.com','9866875532','rishika123','banglore');
end loop;
close cur;

set val=(SELECT count(*) from backup_users where backup_users.u_id=u_id_val);
if val>0 THEN
DELETE from users where users.u_id=u_id_val;
end if;
end $$
DELIMITER ;
DELIMITER $$
create trigger before_delete_orders
before DELETE
on users_orders for each ROW
begin
call delete_users_orders(old.u_id);
end $$
DELIMITER $$

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0135 seconds.)

```

create PROCEDURE delete_users_orders(IN u_id_val int(10)) BEGIN declare o_id int(10); declare price decimal(6,2); declare date
timestamp; declare title char(60); declare quantity int(80); declare u_id char(10); declare username char(20); declare f_name char(20);
declare l_name char(20); declare email varchar(20); declare phone int(20); declare password varchar(80); declare address char(20);
declare done int default 0; declare val int; declare cur cursor for select O.o_id,O.u_id,O.price,O.date,O.title,O.quantity,
B.username,B.f_name,B.l_name,B.email,B.phone,B.password,B.address from users_orders O,users B WHERE O.u_id=B.u_id and O.u_id=u_id_val;
declare continue handler for not found set done=1; open cur; c1:LOOP fetch cur into

```

[Edit]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0073 seconds.)

```

create trigger before_delete_orders before DELETE on users_orders for each ROW begin call delete_users_orders(old.u_id); end;

```

[Edit inline] [Edit] [Create PHP code]

delete from users_orders where u_id=39;













✓ 1 row affected. (Query took 0.0005 seconds.)

```

delete from users_orders where u_id=39;

```

[Edit inline] [Edit] [Create PHP code]

Browse		Structure		SQL		Search		Insert		Export		Import		Privileges		Operations		Tracking		More	
		u_id		username		f_name		l_name		email		phone		password				address			
<input type="checkbox"/>	 Edit	 Copy	 Delete	33	AKA	akash	mehta	akash67@gmail.com		9876545567		akash11				tumkur					
<input type="checkbox"/>	 Edit	 Copy	 Delete	38	NEH	neha	sharma	neha34@gmail.com		8796788678		neha898				banglor					
<input type="checkbox"/>	 Edit	 Copy	 Delete	40	Sinch	sinchana	d	sinchana123@gmail.com		9866784566		sinch123				banglor					
<input type="checkbox"/>	 Edit	 Copy	 Delete	41	shrav	shravani	p	shravani@gmail.com		88967552347		shrav123				mysore					

```
INSERT INTO users VALUES
```

(39,'rishika','rishika','kulkarni','rishika123@gmail.com',9866875532,'rishika123','bangalore','1','2023-05-024 08:43:49')

```
INSERT INTO users_orders(o_id,u_id,title,quantity,price,date) VALUES (8,39,'Spring Rolls',1,470,'2023-05-024 08:43:49');
```

DBMS lab 10

Vaidhoorya

PES1UG20EC274

```
front_end.py
C:\Users> nanuv > Desktop > front_end.py > ...
1 import mysql.connector
2 import streamlit as st
3 import pandas as pd
4
5 db = mysql.connector.connect(
6     host="localhost",
7     port="3306",
8     user="root",
9     password="",
10    database="final"
11)
12
13 cursor = db.cursor()
14
15 ...
16 ...
17
18 st.title("Train Details")
19
20 cursor.execute("SELECT * FROM train")
21 st.dataframe(pd.DataFrame(cursor.fetchall(), columns=("Train No", "Train Name", "Train Type", "Source", "Destination", "Availability")))
22
23 st.subheader("Add a new train")
24 train_no = st.number_input("Train no",key="train_no_create")
25 train_name = st.text_input("Train name",key="train_name_create")
26 train_type = st.text_input("Train Type",key="train_type_create")
27 source = st.text_input("Source",key="source_create")
28 destination = st.text_input("Destination",key="destination_create")
29 availability = st.text_input("Availability",key="availability_create")
30 if st.button("Add train"):
31     cursor.execute("INSERT INTO train VALUES (%s,%s,%s,%s,%s,%s)", (train_no, train_name, train_type, source, destination, availability))
32     db.commit()
33     st.success("Train added successfully")
34
35 st.subheader("Update a train")
36 train_no = st.number_input("Train no",key="train_no_update")
37 train_name = st.text_input("Train name",key="train_name_update")
38 train_type = st.text_input("Train Type",key="train_type_update")
```

Welcome to XAMPP

localhost / 127.0.0.1 / final | phpMyAdmin

localhost/phpmyadmin/index.php?route=/database/structure&db=final

Server: 127.0.0.1 > Database: final

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers More

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> compartment	Browse Structure Search Insert Empty Drop	20	InnoDB	utf8mb4_general_ci	32.0 KIB	-
<input type="checkbox"/> fare	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> passenger	Browse Structure Search Insert Empty Drop	33	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> payment_backup	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> payment_info	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> route_info	Browse Structure Search Insert Empty Drop	44	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> ticket	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> ticket_passenger	Browse Structure Search Insert Empty Drop	21	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> train	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> user_train	Browse Structure Search Insert Empty Drop	103	InnoDB	utf8mb4_general_ci	16.0 KIB	-
10 tables	Sum	268	InnoDB	utf8mb4_general_ci	176.0 KIB	0 B

☐ Check all With selected:

Print Data dictionary

Console

Bookmarks Options History Clear

AMULYA A SHETTY
PES1UG20EC284

Experiment 9:

```
DELIMITER $$
CREATE TRIGGER check_compartment_full
BEFORE INSERT ON compartment
for each ROW
BEGIN
if (SELECT COUNT(*) FROM compartment WHERE train_number= NEW.train_number ) >=4
THEN
    SIGNAL SQLSTATE '56001' SET MESSAGE_TEXT ='Train cannot have more than 4
compartments ';
END IF ;
END; $$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0107 seconds.)

```
CREATE TRIGGER check_compartment_full BEFORE INSERT ON compartment for each ROW BEGIN if (SELECT COUNT(*) FROM compartment WHERE train_number= NEW.train_number ) >=4 THEN
SIGNAL SQLSTATE '56001' SET MESSAGE_TEXT ='Train cannot have more than 4 compartments '; END IF ; END;;
```

INSERT INTO compartment VALUES("F01","Class IV",18,0,58450);

✓ 1 row inserted. (Query took 0.0050 seconds.)

```
INSERT INTO compartment VALUES ('F01', 'Class IV', 18, 0, 58450);
```

Experiment 10:

```
import mysql.connector
```

```
import streamlit as st
```

```
import pandas as pd
```

```
db = mysql.connector.connect(
    host="localhost",
    port="3306",
    user="root",
    password="Amulya@8606",
    database="new_railway"
)
```

```
cursor = db.cursor()
```

```
'''
```

```
# Railway Reservation System
```

```
PES1UG21EC284 - Amulya A Shetty
```

```
'''
```

```
st.title("Train Details")
```

```
cursor.execute("SELECT * FROM train")
```

```
st.dataframe(pd.DataFrame(cursor.fetchall(), columns=("Train No", "Train Name", "Train Type", "Source",
"Destination", "Availability")))
```

```
st.subheader("Add a new train")
```

```
train_no = st.number_input("Train_no",key="train_no_create")
```

```
train_name = st.text_input("Train_name",key="train_name_create")
```

```
train_type = st.text_input("Train_Type",key="train_type_create")
```

```
source = st.text_input("Source",key="source_create")
```

```
destination = st.text_input("Destination",key="destination_create")
```

```
availability = st.text_input("Availability",key="availability_create")
```

```
if st.button("Add Train"):
```

```
    cursor.execute("INSERT INTO train VALUES (%s, %s, %s, %s, %s, %s)", (train_no, train_name,
train_type, source, destination, availability))
```

```
    db.commit()
```

```
    st.success("Train added successfully")
```

```
st.subheader("Update a train")
```

```
train_no = st.number_input("Train_no",key="train_no_update")
```

```
train_name = st.text_input("Train_name",key="train_name_update")
```

```
train_type = st.text_input("Train_Type",key="train_type_update")
```

```
source = st.text_input("Source",key="source_update")
```

```

destination = st.text_input("Destination",key="destination_update")
availability = st.text_input("Availability",key="availability_update")
if st.button("Update Train"):
    cursor.execute("UPDATE train SET Train_name = %s, Train_Type = %s, source = %s, destination = %s,
availability = %s WHERE Train_no = %s", (train_name, train_type, source, destination, availability,
train_no))
    db.commit()
    st.success("Train updated successfully")
st.subheader("Delete a train")
train_no = st.number_input("Train_no",key="train_no_delete")
if st.button("Delete Train"):
    cursor.execute("DELETE FROM train WHERE Train_no = %s", (train_no,))
    db.commit()
    st.success("Train deleted successfully")

cursor.close()
db.close()

```

Railway Reservation System

PES1UG21EC284 - Amulya A Shetty

Train Details

	Train No	Train Name	Train Type	Source	Destination	Availability
0	25,260	Chennai Mail	Mail	Mangaluru	Chennai	Yes
1	25,261	Managaluru Mail	Mail	Chennai	Mangaluru	Yes
2	58,450	MAN-BEN Express	Fast	Mangaluru	Bengaluru	yes
3	58,451	BEN-MAN Express	Fast	Bengaluru	Mangaluru	yes
4	62,620	CHE-BEN Shatabdi	Superfast	Chennai	Bengaluru	No
5	62,621	BEN-CHE Shatabdi	Superfast	Bengaluru	Chennai	yes
6	1	Chennai express	AC	JP Nagar	Magestic	5
7	1	Chennai express	AC	JP Nagar	Magestic	yes

Add a new train

Train_no

62624.00

- +

Update a train

Train_no

0.00

- +

Train_name

Chennai express

Train_Type

Passenger

Source

Bangalore

Destination

Mumbai

Availability

No

Update Train

Delete a train

Train_no

58451.00

- +

Delete Train

Train deleted successfully

Railway Reservation System

PES1UG21EC284 - Amulya A Shetty

Train Details

	Train No	Train Name	Train Type	Source	Destination	Availability
0	25,260	Chennai Mail	Mail	Mangaluru	Chennai	Yes
1	25,261	Managaluru Mail	Mail	Chennai	Mangaluru	Yes
2	58,450	MAN-BEN Express	Fast	Mangaluru	Bengaluru	yes
3	58,451	BEN-MAN Express	Fast	Bengaluru	Mangaluru	yes
4	62,620	CHE-BEN Shatabdi	Superfast	Chennai	Bengaluru	No
5	62,621	BEN-CHE Shatabdi	Superfast	Bengaluru	Chennai	yes
6	1	Chennai express	AC	JP Nagar	Magestic	5
7	1	Chennai express	AC	JP Nagar	Magestic	yes