

Project Report

Course: BAN 610

Group 7:

Karan Malik
Vinod Kumar Reddy
Subramanian Tirunelveli Padmanabhan

ZOMATO

Introduction

Zomato is an e-food platform which provides information on restaurants, bars, clubs and dine-outs. It offers online food delivery and serves customers across many countries.

Business Goals

Zomato has built a sustainable eco system with its vast delivery network and logistical resources. Below are its business goals:

- Enabling easy customer access to restaurants at affordable prices from various locations
- Efficient technology setup for easy access by customers and other partners
- Provides clear content on social media platforms to promote the brand.
- Assisting restaurants promote their banners on the site to increase visibility to customers.

Products/Services

- Collaborating with payment applications to bring in discounts and many payment options.
- Use of analytical tools to sense customer interest and allowing restaurants to flash offers.
- Partners with restaurants and organizes food events.
- Hassle free food delivery services with subscriptions for all partners
- Publishes data about ratings to evaluate restaurants and build customer loyalty.

User Requirements:

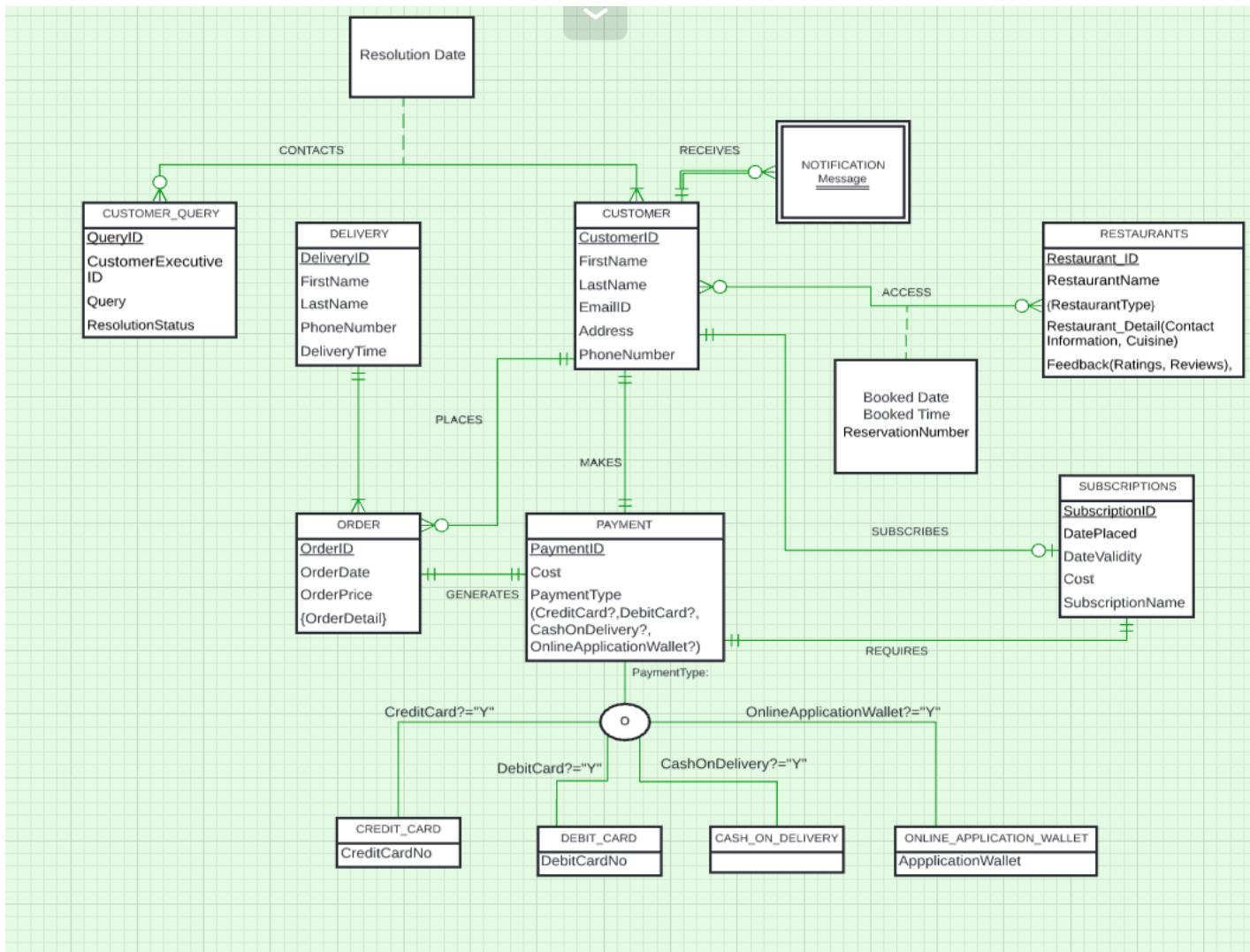
Zomato has stakeholders playing different roles to offer services. Below are the requirements:

1. Customer is identified with his personal information and his orders.
 - Customer has Customer ID as the identifier and has first name, last name, phone number, email ID and address as other attributes.
 - Order has Order ID as identifier and has Order Date, Order Price and Order Detail as attributes. Order Detail can have more than one food.
 - Customer can place any number of orders and order must belong to one customer.
2. Customers can order food or reserve tables in restaurants which have:
 - Restaurant ID as unique identifier and Restaurant Type(s), name, Restaurant Detail (Contact Information, Menu), and Feedback (Ratings, Reviews) as attributes.
 - Reservation number, booked data and time help track the table reserved by customer.
 - Customer can order from none, one or many restaurants and restaurants can serve none, one or many customers.
3. Order placed by customer is handled through secure payment gateway. Payment has
 - Payment ID as identifier and price and mode of payment as attributes.
 - Credit card, Debit card and cash on delivery are some types of payment. There can be other modes of payment. Payment can be done through multiple modes.
 - Payment can be done by only one customer.
4. Each order will be delivered by one agent to customer.

Project Report - BAN 610

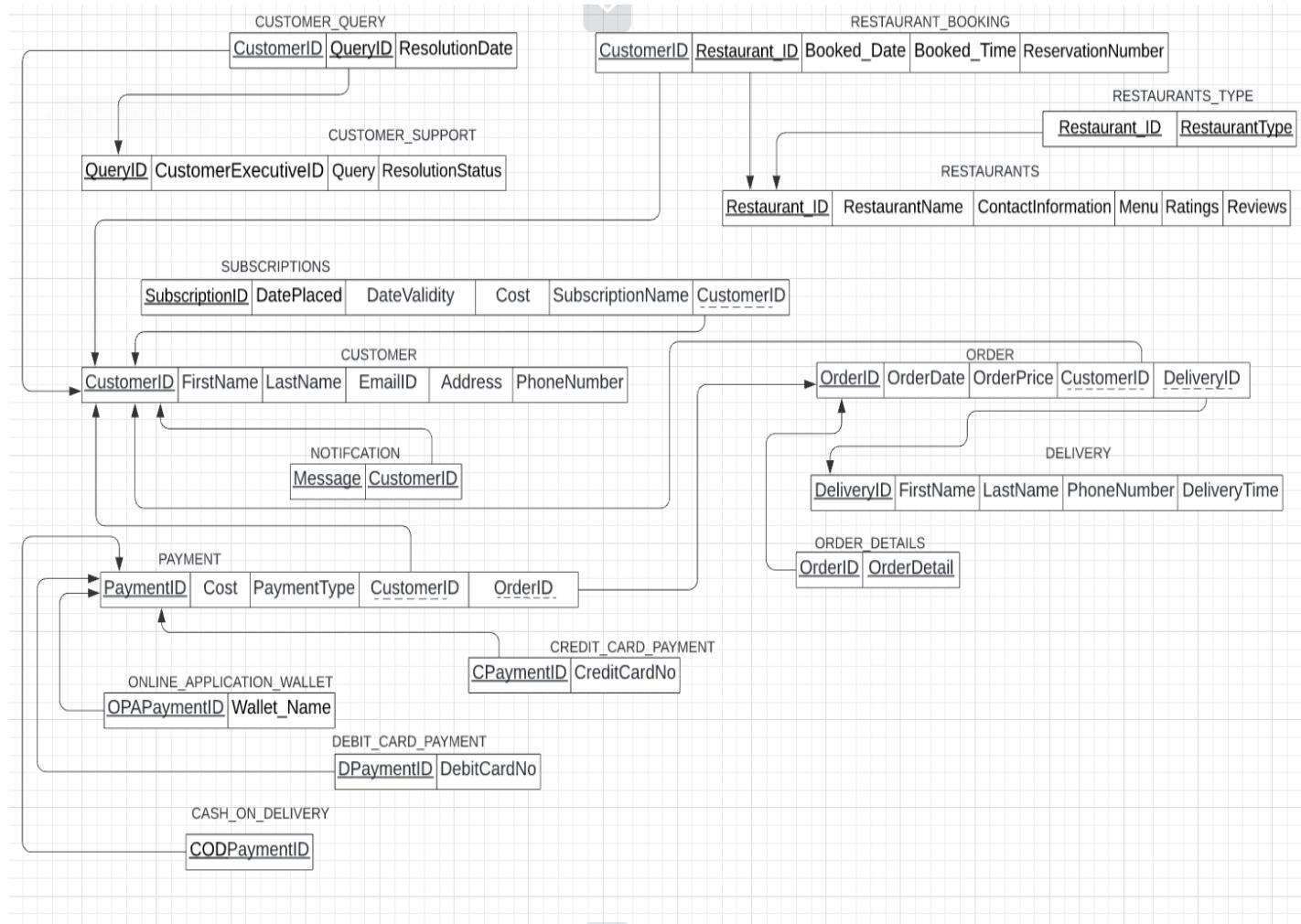
- Delivery will have Delivery ID of agent as identifier and First name, last name, delivery time and his phone number as other attributes.
 - Delivery of order is tracked by Delivery time and agent can deliver many orders.
5. Queries related to payment orders are answered through customer support.
- Query ID, which is auto generated sequence number, is the identifier. Query, Customer executive ID and Resolution Status are other attributes.
 - Customer can have zero or many queries. Resolution date tracks the query completion date. An Executive can respond to multiple customers/queries.
6. Customer can have none, one or many subscriptions.
- Subscription ID is the identifier of the service offered and has Date Placed, Validity, Cost and Subscription Name
 - Subscription is unique for a user and user can have one subscription at a time.
7. Customer can have many notifications.
- Notification is dependent on the customer interest and behavior and Notification Message as an attribute.

Conceptual Data Modeling



Project Report - BAN 610

Logical Database Design



METADATA

The physical implementation required 16 tables to be created.

1	Delivery					
2	Attribute	Data Type	Size	AUTO	Null	Constraint
3	Delivery_id	int	11	yes	no	primary
4	First name	varchar	50	no	no	
5	last name	varchar	50	no	no	
6	Phone number	varchar	30	no	no	
7	Delivery_time	varchar	50	no	no	
8						
9	Order					
10	Attribute	Data Type	Size	AUTO	Null	Constraint
11	order_id	int	11	yes	no	primary
12	order_date	date		no	no	
13	order_price	double		no	no	
14	customer_id	int	11	no	no	Foreign
15	Delivery_id	int	11	no	no	Foreign

Project Report - BAN 610

10	Order details					
17	Attribute	Data Type	Size	AUTO	Null	Constraint
18	order_id	int	11		no	Part of Primary,Foreign
19	order_details	varchar	200	no	no	Primary
21						
22	Credit card payment					
23	Attribute	Data Type	Size	AUTO	Null	Constraint
24	cpayment_id	int	11	yes	no	Part of Primary,Foreign
25	credit_card_number	int	16	no	no	
26						
27	cod payment					
28	Attribute	Data Type	Size	AUTO	Null	Constraint
29	codpayment_id	int	11	yes	no	Part of Primary,Foreign
30						
31	CUSTOMER					
32	Attribute	Data Type	Size	AUTO	Null	Constraint
33	Customer_ID	int	11	yes	no	Primary
34	First_Name	varchar	50	no	no	
35	Last_Name	varchar	50	no	yes	
36	EmailID	varchar	40	no	no	
37	Address	varchar	200	no	no	
38	PhoneNumber	varchar	30	no	no	
39						
40	SUBSCRIPTIONS					
41	Attribute	Data Type	Size	AUTO	Null	Constraint
42	Subscription_ID	int	11	yes	no	Primary
43	Date_Placed	Date		no	no	
44	Date_Validity	Date		no	no	
45	Cost	double		No	no	
46	Subscription_Name	varchar	100	no	no	
47	Customer_ID	int	11			Foreign
48						
49	CUSTOMER SUPPORT					
50	Attribute	Data Type	Size	AUTO	Null	Constraint
51	Query_ID	int	11	yes	no	Primary
52	Customer_Executive_ID	Int	11	no	no	
53	query_description	varchar	1000	no	no	
54	Resolution_Status	varchar	40	no	no	
56						
57	CUSTOMER QUERY					
58	Attribute	Data Type	Size	AUTO	Null	Constraint
59	CustomerID	int	11			Part of Primary,Foreign
60	QueryID	int	11			Part of Primary,Foreign
61	ResolutionDate	Date		no	no	
62						
63	payment					
64	Attribute	Data Type	Size	AUTO	Null	Constraint
65	payment_id	int	11	yes	no	Primary
66	cost	double		no	no	
67	payment_type	varchar	100	no	no	
68	customer_id	int	11			Foreign
69	order_id	int	11			Foreign
70						
71	notification					
72	Attribute	Data Type	Size	AUTO	Null	Constraint
73	message	varchar	100	no	no	Primary
74	customer_id	int	11			Part of Primary,Foreign
75						
76	restaurant					
77	Attribute	Data Type	Size	AUTO	Null	Constraint
78	Restaurant_id	Int	10	Yes	No	Primary Key
79	contact infromation	varchar	50	No	No	
80	cusine	Varchar	25	No	No	
81	Ratings	double	10	NO	yes	
82	Reviews	Varchar	150	No	yes	
83	restaurant name	varchar	500	NO	no	
84						

Project Report - BAN 610

84					
85	restaurant_type				
86	Attribute	Data	Size	AUTO	Null
87	Restaurant id	Int	10		Constraint Part of primary key, Foreign
88	Restaurant type	varchar	10	No	primary key
89					
90					
91	restaurant_booking				
92	Attribute	Data	Size	AUTO	Null
93	restaurant_id	Int	10		Part of Primary Key, Foreign
94	Customer id	int	10		Part of Primary Key, Foreign
95	Booked date	date		No	No
96	reservation_num	int	15	no	yes
97	Booked time	time	date	NO	No
98					
99	debit_card_payment				
100	Attribute	Data	Size	AUTO	Null
101	dpayment_id	Int	11		Part of Primary Key, Foreign
102	debit card number	varchar	30	no	no
103					
104					
105	ONLINE APPLICATION WALLET				
106	Attribute	Data	Size	AUTO	Null
107	oaw_payment_id	Int	20		Part of Primary Key, Foreign
108	wallet_name	varchar	20		not null
109					
110					
111					

Implementation in MySQL

CREATE TABLE

1.CREATE Table Query: customer_t

```
8      /*Create the customer_t table*/
9 •  CREATE TABLE customer_t (
10    customer_id int(11) not null AUTO_INCREMENT,
11    first_name varchar(50) not null,
12    last_name varchar(50),
13    email_id varchar(40) not null,
14    address varchar(200) not null,
15    phone_number varchar(30) not null,
16    CONSTRAINT customer_pk PRIMARY KEY (customer_id))
17    ENGINE = INNODB;
18
```

2.CREATE Table Query: subscriptions_T

```
/*Create the subscriptions_T table*/
CREATE TABLE subscriptions_T (
    subscription_id int(11) not null AUTO_INCREMENT,
    date_placed date not null,
    date_validity date not null,
    cost double not null,
    subscription_name varchar(100) not null,
    customer_id int (11),
    CONSTRAINT subscriptions_pk PRIMARY KEY (subscription_id),
    CONSTRAINT subscriptions_fk FOREIGN KEY (customer_id) REFERENCES customer_t (customer_id)
    ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

3. CREATE Table Query: customer_support_t

```
/*Create the customer_support_t table*/
CREATE TABLE customer_support_t (
    query_id int(11) not null AUTO_INCREMENT,
    customer_executive_id int(11) not null,
    query_description varchar(1000) not null,
    resolution_status varchar(40) not null,
    CONSTRAINT customer_support_pk PRIMARY KEY (query_id))
ENGINE = INNODB;
```

4. CREATE Table Query: customer_query_t

```
/*Create the customer_query_t table*/
CREATE TABLE customer_query_t (
    query_id int(11),
    customer_id int (11),
    resolution_Date date not null,
    CONSTRAINT customer_query_pk PRIMARY KEY (query_id, customer_id),
    CONSTRAINT customer_query_fk1 FOREIGN KEY (query_id) REFERENCES customer_support_t (query_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT customer_query_fk2 FOREIGN KEY (customer_id) REFERENCES customer_t (customer_id)
    ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

5. CREATE Table Query: delivery_t

```
/*Create the delivery table*/
CREATE TABLE delivery_t (
    delivery_id int(11) not null AUTO_INCREMENT,
    first_name varchar(50) not null,
    last_name varchar(50) ,
    phone_number varchar(30) not null,
    delivery_time datetime not null,
    CONSTRAINT delivery_pk PRIMARY KEY (delivery_id)
)
ENGINE = INNODB;
```

6. CREATE Table Query: order_t

```
/*Create the order table*/
CREATE TABLE order_t (
    order_id int(11) not null AUTO_INCREMENT,
    order_date date NOT NULL,
    order_price double NOT NULL,
    customer_id int(11),
    delivery_id int(11),
    CONSTRAINT order_pk PRIMARY KEY (order_id),
    CONSTRAINT order_fk1 FOREIGN KEY (delivery_id) REFERENCES delivery_t (delivery_id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT order_fk2 FOREIGN KEY (customer_id) REFERENCES customer_t (customer_id)
        ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

7. CREATE Table Query: order_details_t

```
/*Create the order details table*/
CREATE TABLE order_details_t (
    order_id int(11) NOT NULL,
    order_details varchar(200) NOT NULL,
    CONSTRAINT order_detail_Pk PRIMARY KEY (order_id, order_details),
    CONSTRAINT order_detail_fk FOREIGN KEY (order_id) REFERENCES order_t (order_id)
        ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

8.CREATE Table Query: payment_t

```
/*Create the payment_t table*/
CREATE TABLE payment_t (
    payment_id int(11) not null AUTO_INCREMENT,
    cost double not null,
    payment_type varchar(100) not null,
    customer_id int(11),
    order_id int(11),
    CONSTRAINT payment_pk PRIMARY KEY (payment_id),
    CONSTRAINT payment_fk1 FOREIGN KEY (order_id) REFERENCES order_t (order_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT payment_fk2 FOREIGN KEY (customer_id) REFERENCES customer_t (customer_id)
    ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

9. CREATE Table Query: notification_t

```
/*Create the notification_t table*/
CREATE TABLE notification_t (
    message varchar(100) not null,
    customer_id int(11),
    CONSTRAINT notification_pk PRIMARY KEY (message,customer_id),
    CONSTRAINT notification_fk FOREIGN KEY (customer_id) REFERENCES customer_t (customer_id)
    ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

10. CREATE Table Query: restaurant_T

```
/*Restaurant_T*/
CREATE TABLE restaurant_T(
    restaurant_id int(10) AUTO_INCREMENT NOT NULL,
    restaurant_name varchar(500) not null,
    contact_information varchar(15) not null,
    cusine varchar(15) not null,
    ratings DOUBLE ,
    reviews varchar(150),
    CONSTRAINT restaurant_pk PRIMARY KEY(Restaurant_id));
```

11. CREATE Table Query: restaurant_type_T

```
/*RestaurantType_T*/
CREATE TABLE restaurant_type_T(
    restaurant_id int(10) ,
    restaurant_type varchar(15) not null,
    CONSTRAINT restaurant_type_pk PRIMARY KEY(restaurant_id,restaurant_type),
    CONSTRAINT restaurant_type_fk FOREIGN KEY(restaurant_id) REFERENCES restaurant_T(restaurant_id)
    ON UPDATE CASCADE ON DELETE CASCADE);
```

12. CREATE Table Query: restaurant_booking_T

```
/*create table restaurant_booking_T*/
CREATE TABLE restaurant_booking_T(
    restaurant_id int(10) ,
    customer_id int(11) ,
    booked_date DATE not null,
    booked_time TIME(6) not null,
    reservation_num INT(15) not null,
    CONSTRAINT restaurant_booking_pk PRIMARY KEY(restaurant_id,customer_id),
    CONSTRAINT restaurant_booking_fk1 FOREIGN KEY(restaurant_id) REFERENCES restaurant_T(Restaurant_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT restaurant_booking_fk2 FOREIGN KEY(customer_id) REFERENCES Customer_T(Customer_id)
    ON UPDATE CASCADE ON DELETE CASCADE);
```

13. CREATE Table Query: debit_card_payment_T

```
/*CREATING DEBIT CARD PAYMENT TABLE*/
CREATE TABLE debit_card_payment_T(
    dpayment_id int(11),
    debit_card_number varchar(30) not null,
    CONSTRAINT debit_card_payment_pk PRIMARY KEY (dpayment_id),
    CONSTRAINT debit_card_payment_fk FOREIGN KEY (dpayment_id) REFERENCES payment_t(payment_id)
    ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

14. CREATE Table Query: online_application_wallet_T

```
/*CREATING WALLET TABLE*/
Create Table online_application_wallet_T(
oaw_payment_id int(20),
wallet_name VARCHAR(20) not null,
CONSTRAINT online_application_wallet_pk PRIMARY KEY (oaw_payment_id),
CONSTRAINT online_application_wallet_fk FOREIGN KEY (oaw_payment_id) REFERENCES payment_t (payment_id)
ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

15. CREATE Table Query: credit_card_payment_t

```
/*Create the credit card payment_t table*/
CREATE TABLE credit_card_payment_t (
cpayment_id int(11),
credit_card_number VARCHAR(30) not null,
CONSTRAINT credit_card_payment_pk PRIMARY KEY (cpayment_id),
CONSTRAINT credit_card_payment_fk FOREIGN KEY (cpayment_id) REFERENCES payment_t (payment_id)
ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

16. CREATE Table Query: cash_on_delivery_payment_t

```
/*Create the cod payment_t table*/
CREATE TABLE cash_on_delivery_payment_t (
codpayment_id int(11) ,
CONSTRAINT cash_on_delivery_payment_pk PRIMARY KEY (codpayment_id),
CONSTRAINT cash_on_delivery_payment_fk FOREIGN KEY (codpayment_id) REFERENCES payment_t (payment_id)
ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;
```

Project Report - BAN 610

INSERTING VALUES INTO TABLES

1) Inserting values into Customer table:

```
/*insert into customer_t*/
INSERT INTO customer_t (customer_id, first_name, last_name, email_id, address, phone_number) VALUES
(115, 'John', 'Doe', 'john@gmail.com', '200 Maple Street, Indiranagar, Bangalore 560008, India', '455-908-6781'),
(101, 'Bill', 'Doll', 'bill@gmail.com', '300 West, Indiranagar, Bangalore 560008, India', '455-920-6782'),
(102, 'Karthik', 'Turburn', 'karthik@gmail.com', '250 North, Indiranagar, Bangalore 560008, India', '455-921-6739'),
(103, 'Mary', 'Lee', 'mary@gmail.com', '200 South, Indiranagar, Bangalore 560008, India', '455-908-6811'),
(104, 'Jane', 'Smith', 'jane@gmail.com', '120 University, MG Road, Mumbai 762112, India', '455-908-6900'),
(105, 'Vincent', 'Smith', 'vincent@gmail.com', '200 Maple, MG Road, Mumbai 762112, India', '455-908-6814'),
(106, 'David', 'Brown', 'david@gmail.com', '300 West, MG Road, Mumbai 762112, India', '455-908-6888'),
(107, 'Carl', 'Smith', 'carl@gmail.com', '250 North, MG Road, Mumbai 762112, India', '455-908-6670'),
(108, 'John', 'Doll', 'doll@gmail.com', '200 South, MG Road, Mumbai 762112, India', '455-908-6123'),
(109, 'Jim', 'Chris', 'chris@gmail.com', '120 University, Castle Road, Hyderabad 87102, India', '455-908-6321'),
(110, 'Joyce', 'French', 'joyce@gmail.com', '200 Main, Castle Road, Hyderabad 87102, India', '455-908-6456'),
(111, 'Manav', 'English', 'manav@gmail.com', '250 State, Castle Road, Hyderabad 87102, India', '455-908-6653'),
(112, 'Tom', 'Borg', 'jtom@gmail.com', '3000 Sunset, Castle Road, Hyderabad 87102, India', '455-908-6654'),
(113, 'Donald', 'Thomas', 'donald@gmail.com', '1200 Hollywood, Majestic Road, Delhi 87123, India', '455-908-6546'),
(114, 'Vishwas', 'Borg', 'vishwas@gmail.com', '100 College, Majestic Road, Delhi 87123, India', '455-908-6711'),
(116, 'Brandon', 'French', 'bran@gmail.com', '200 Main, Windtunnel Road, Chennai 76112, India', '455-908-6712'),
(117, 'Victor', 'English', 'vctor@gmail.com', '250 State, Windtunnel Road, Chennai 76112, India', '455-908-6713'),
(118, 'Alice', 'Borg', 'alice@gmail.com', '3000 Sunset, Windtunnel Road, Chennai 76112, India', '455-908-6715'),
(119, 'Margaret', 'Thomas', 'amrg@gmail.com', '1200 Hollywood, Windtunnel Road, Chennai 76112, India', '455-908-6714'),
(120, 'Aston', 'Borg', 'aston@gmail.com', '100 College, Windtunnel Road, Chennai 76112, India', '455-908-6717'),
(121, 'Joel', 'French', 'joel@gmail.com', '200 Main, Brigade Road, Kolkata 88111, India', '455-908-6811'),
(122, 'Robert', 'English', 'robert@gmail.com', '250 State, Brigade Road, Kolkata 88111, India', '455-908-6812'),
(123, 'Leonardo', 'Borg', 'leo@gmail.com', '3000 Sunset, Brigade Road, Kolkata 88111, India', '455-908-6813'),
(124, 'Bruce', 'Thomas', 'bruce@gmail.com', '1200 Hollywood, Brigade Road, Kolkata 88111, India', '455-908-6814'),
(125, 'Shawn', 'Borg', 'shawn@gmail.com', '100 College, Brigade Road, Kolkata 88111, India', '455-908-6815');
```

2) Inserting values into Subscription table:

```
/*insert into subscriptions_t*/
• INSERT INTO subscriptions_t (subscription_id, date_placed, date_validity, cost, subscription_name, customer_id) VALUES
(501, '2020-10-24', '2021-01-23', '300', 'Zomato Pro', '101'),
(502, '2020-09-23', '2020-12-22', '300', 'Zomato Pro', '102'),
(503, '2020-08-22', '2020-11-21', '300', 'Zomato Pro', '103'),
(504, '2020-07-21', '2020-10-20', '300', 'Zomato Pro', '104'),
(505, '2020-06-20', '2020-09-19', '300', 'Zomato Pro', '105'),
(515, '2020-05-19', '2020-08-18', '300', 'Zomato Pro', '106'),
(506, '2020-04-18', '2020-07-17', '300', 'Zomato Pro', '107'),
(507, '2020-03-17', '2020-06-16', '300', 'Zomato Pro', '108'),
(508, '2020-02-16', '2020-05-15', '300', 'Zomato Pro', '109'),
(509, '2020-01-15', '2020-04-14', '300', 'Zomato Pro', '110'),
(510, '2020-11-14', '2021-02-13', '300', 'Zomato Pro', '111'),
(511, '2020-12-13', '2021-03-12', '300', 'Zomato Pro', '112'),
(512, '2021-01-12', '2021-04-11', '300', 'Zomato Pro', '113'),
(513, '2021-02-11', '2021-05-10', '300', 'Zomato Pro', '114'),
(514, '2021-03-10', '2021-06-09', '300', 'Zomato Pro', '115'),
(516, '2019-11-14', '2020-02-13', '500', 'Zomato Pro', '116'),
(517, '2019-12-13', '2020-03-12', '500', 'Zomato Pro', '117'),
(518, '2019-01-12', '2019-04-11', '500', 'Zomato Pro', '118'),
(519, '2019-02-11', '2019-05-10', '500', 'Zomato Pro', '119'),
(520, '2019-03-10', '2019-06-09', '500', 'Zomato Pro', '120'),
(521, '2018-11-14', '2019-02-13', '400', 'Zomato Pro', '121'),
(522, '2018-12-13', '2019-03-12', '400', 'Zomato Pro', '122'),
(523, '2018-01-12', '2018-04-11', '400', 'Zomato Pro', '123'),
(524, '2018-02-11', '2018-05-10', '400', 'Zomato Pro', '124'),
(525, '2018-03-10', '2018-06-09', '400', 'Zomato Pro', '125');
```

Project Report - BAN 610

3) Inserting values into Customer Support table:

```
INSERT INTO customer_support_t (query_id, customer_executive_id, query_description, resolution_status) VALUES  
(601, '901','Unable to do payment ','In Progress'),  
(602, '902','Unable to add items to cart ','In Progress'),  
(603, '903','Unable to do add subscription ','In Progress'),  
(604, '904','Need more info about payment options ','In Progress'),  
(605, '905','Zomato offers on various food items in restaurants','In Progress'),  
(615, '906','Restaurant offers in specific location','Closed'),  
(606, '907','Unable to order food','Closed'),  
(607, '908','Zomato subscription page not loading','Closed'),  
(608, '909','Technical error appearing on adding to cart','Closed'),  
(609, '910','Need more information on offers during festival','Closed'),  
(610, '911','Did not get the order despite ordering 2 hours ago','Closed'),  
(611, '912','Not happy with the food delivered. Need refund','Escalated'),  
(612, '913','Unable to reach the delivery person','Escalated'),  
(613, '914','Restaurant is not allowing to reserve the table','In Progress'),  
(614, '915','Order contains fewer items than what was ordered','Escalated'),  
(623, '916','Unable to get subscription activated','Escalated'),  
(616, '917','Offer for online delivery of pizza','Closed'),  
(617, '918','Any more subscriptions in the future','Closed'),  
(618, '919','How to enroll as a vendor for zomato','Closed'),  
(619, '920','Not happy with food quality','In Progress'),  
(620, '921','How much is the maximum discount using subscription','Closed'),  
(621, '922','When is the validity ending for subscription','Closed'),  
(622, '923','Can zomato update the validity of subscription','Closed'),  
(624, '924','Does zomato deliver during late night hours','Closed'),  
(625, '925','What is the last order time for meals','Closed');
```

4) Inserting values into Customer Query table:

```
/*insert into customer_query_t*/  
INSERT INTO customer_query_t (query_id, customer_id, resolution_date) VALUES  
(601, '101','2021-03-10'),  
(602, '101','2021-02-10'),  
(603, '101','2021-01-11'),  
(604, '101','2021-04-13'),  
(605, '101','2021-01-01'),  
(606, '101','2021-04-10'),  
  
(607, '102','2020-03-10'),  
(608, '102','2020-02-10'),  
(609, '103','2020-01-11'),  
(610, '104','2020-04-13'),  
(611, '105','2020-01-01'),  
(612, '106','2020-04-10'),  
  
(613, '107','2019-03-10'),  
(614, '108','2019-02-10'),  
(615, '109','2019-01-11'),  
(616, '110','2019-04-13'),  
(617, '111','2019-08-01'),  
(618, '112','2019-12-10'),  
  
(619, '113','2018-03-10'),  
(620, '114','2018-02-10'),  
(621, '115','2018-01-11'),  
(622, '116','2018-10-13'),  
(623, '117','2018-09-01'),
```

5) Inserting values into Delivery table:

```

/*insert values into the Delivery table*/
INSERT INTO delivery_T(delivery_id, first_name, last_name, phone_number, delivery_time) VALUES
(701,'Karan','jindal','123-908-6789',NOW()),
(702,'Rahul','Malik','123-908-6789',NOW()),
(703,'Stephen','chau','123-908-6789',NOW()),
(704,'Karthik','Malik','187-888-87',NOW()),
(705,'Karan','chau','333-908-6789',NOW()),
(706,'andrew','patel','763-908-6789',NOW()),
(707,'nice','guy','555-908-6789',NOW()),
(708,'apoorna','roy','553-908-6789',NOW()),
(709,'raj','patel','123-508-6789',NOW()),
(710,'sakshi','jain','153-908-6789',NOW()),
(711,'prachi','Malik','456-908-6789',NOW()),
(712,'parvi','goyal','123-999-6789',NOW()),
(713,'ruchi','bajaj','123-000-6789',NOW()),
(714,'anita','garg','123-000-6789',NOW()),
(715,'ankit','garg','189-888-6789',NOW()),
(716,'ankur','mehta','176-908-6789',NOW()),
(717,'raghav','Munja','563-908-8888',NOW()),
(718,'sonam','saini Munjal','651-788-6789',NOW()),
(719,'Karan','sachdeva','563-908-6789',NOW()),
(720,'nitish','sethi','981-108-6789',NOW()),
(721,'bharat','kumar','989-908-6789',NOW()),
(722,'sakshi','kumar','123-908-5432',NOW()),
(723,'vineet','pushkar','123-908-8976',NOW()),
(724,'vineet','pushkar','415-808-6789',NOW()),
(725,'vinod','bajaj','213-908-6789',NOW()),
(726,'Karan','jindal','123-908-6789',NOW()),

```

6) Inserting values into Order table:

```

/*insert values into the order table*/
INSERT INTO order_T(order_id, order_date, customer_id, order_price, delivery_id) VALUES
(201, '2008-10-24', 101, 100, 701),
(202, '2008-10-21', 102, 200, 702),
(203, '2008-10-22', 103, 300, 703),
(204, '2008-10-22', 104, 100, 704),
(205, '2008-10-24', 105, 1400, 705),
(206, '2008-10-24', 106, 1010, 706),
(207, '2008-10-27', 107, 1200, 707),
(208, '2008-10-30', 108, 100, 708),
(209, '2008-11-01', 109, 200, 709),
(210, '2008-11-05', 110, 300, 710),
(211, '2008-10-24', 111, 1400, 711),
(212, '2008-10-21', 112, 500, 711),
(213, '2008-10-22', 113, 1300, 713),
(214, '2008-10-22', 114, 1080, 714),
(215, '2008-10-24', 115, 1800, 715),
(216, '2008-10-24', 116, 1020, 716),
(217, '2008-10-27', 117, 1010, 717),
(218, '2008-10-30', 118, 1300, 718),
(219, '2008-11-01', 119, 1200, 719),
(220, '2008-11-05', 120, 1100, 720),
(221, '2008-10-24', 121, 1100, 721),
(222, '2008-10-21', 122, 1010, 722),
(223, '2008-10-22', 123, 1005, 723),
(224, '2008-10-22', 124, 200, 724),
(225, '2008-10-24', 125, 300, 725),
(226, '2008-10-24', 101, 100, 726),
(227, '2008-10-21', 102, 200, 727),

```

7) Inserting values into Order Details table:

```

/*insert values into the order detail table*/
INSERT INTO order_details_T(order_id, order_details) VALUES
(201, 'paneer'),
(202, 'Pizza'),
(203, 'Fried rice'),
(204, 'brinjal'),
(205, 'Dal'),
(206, 'potato'),
(207, 'pizza'),
(208, 'Fish'),
(209, 'Chicken'),
(210, 'chicken'),
(211, 'fish'),
(212, 'burrito'),
(213, 'Paneer'),
(214, 'Lamb'),
(215, 'pizza'),
(216, 'steakl'),
(217, 'chicken'),
(218, 'pasta'),
(219, 'beef'),
(220, 'steak'),
(221, 'lamb'),
(222, 'paneer'),
(223, 'pasta'),
(224, 'cheese'),
(225, 'pizza'),
(226, 'paneer'),
(227, 'pasta'),

```

8) Inserting values into Payment table:

```

/*insert into payment_t*/
INSERT INTO payment_t (payment_id, cost,payment_type, customer_id,order_id) VALUES
(301, '200','Credit Card','101','201'),
(302, '300','Credit Card','102','202'),
(303, '200','Credit Card','103','203'),
(304, '400','Credit Card','104','204'),
(305, '500','Credit Card','105','205'),

(306, '200','Credit Card','106','206'),
(307, '300','Credit Card','107','207'),
(308, '200','Credit Card','108','208'),
(309, '400','Credit Card','109','209'),
(310, '500','Credit Card','110','210),

(311, '200','Credit Card','111','211'),
(312, '300','Credit Card','112','212'),
(313, '200','Credit Card','113','213'),
(314, '400','Credit Card','114','214'),
(315, '500','Credit Card','115','215'),

(316, '200','Debit Card','116','216'),
(317, '300','Debit Card','117','217'),
(318, '200','Debit Card','118','218'),
(319, '400','Debit Card','119','219'),
(320, '500','Debit Card','120','220'),

(321, '200','Debit Card','121','221'),

```

Project Report - BAN 610

9) Inserting values into Notification table:

```
/*insert into notification_t*/
INSERT INTO notification_t (message, customer_id) VALUES
('Half Price Offer starts tonight','101'),
('Weekend Offer on Pizzas!!!!','102'),
('Zomato Pro to cost 10% less!!!!','103'),
('Book a Restaurant table today and get 25% discount','104'),
('Mandarin is now offering a Full course meal at 50% price today','105'),
('Zomato pro validity is not extended to 4 months','106'),
('Burgers at Burger king to cost less from tomorrow','107'),
('Biryani zone is now at your nearest location!!!!','108'),
('Half Price Offer starts tonight','109'),
('Weekend Offer on Pizzas!!!!','110'),
('Zomato Pro to cost 10% less!!!!','111'),
('Book a Restaurant table today and get 25% discount','112'),
('Mandarin is now offering a Full course meal at 50% price today','113'),
('Zomato pro validity is not extended to 4 months','114'),
('Burgers at Burger king to cost less from tomorrow','115'),
('Biryani zone is now at your nearest location!!!!','116'),
('Half Price Offer starts tonight','117'),
('Weekend Offer on Pizzas!!!!','118'),
('Zomato Pro to cost 10% less!!!!','119'),
('Book a Restaurant table today and get 25% discount','120');

INSERT INTO Restaurant_T (restaurant_id, restaurant_name, contact_information, cusine, ratings, reviews) VALUES
(401, 'Biryani Zone', '479-200-2755', 'Indian', 3.5, 'Nice restaurant good food & environment and service.'),
```

10) Inserting values into Restaurant table:

```
INSERT INTO Restaurant_T (restaurant_id, restaurant_name, contact_information, cusine, ratings, reviews) VALUES
(401, 'Biryani Zone', '479-200-2755', 'Indian', 3.5, 'Nice restaurant good food & environment and service.'),
(402, 'Barbeque Nation', '479-201-3613', 'Multi cusine', 3.3, 'For me, it was little spicy.'),
(403, 'Punjabi Rasoi', '479-202-0130', 'Noth Indian', 4.5, 'I ordered Fish Fillet and House Special Fried rice. very good'),
(404, 'Toit', '479-203-6427', 'Mexican', 4.1, 'This location did not disappoint!'),
(405, 'Vapour', '479-204-1014', 'Italian', 4.1, 'We enjoyed our food our tater tots and mozzarella sticks'),
(406, 'Nandoos', '479-205-7475', 'South African', 2.9, 'Disappointing flavor on almost everything I got'),
(407, 'Lazy Suzy', '479-206-6425', 'Mexican', 3.5, 'fast food did not sound appealing.'),
(408, 'Shangrilla', '479-207-1184', 'Chinese', 3.5, 'Nice restaurant good food .'),
(409, 'Mainland China', '479-208-9302', 'Chinese', 3, ' Everything here cost $60 total. Not nearly enough.'),
(410, 'Copper Chimney', '479-209-0151', 'German', 3.8, 'everything was good!'),
(411, 'Absolute Barbeque', '479-212-3920', 'Multi Cusine', 4, ' Already a tight parking lot!'),
(412, 'The Taj', '479-213-3474', 'Indian', 2, 'Disappointingly small portions'),
(413, 'California Burritos', '479-214-4992', 'American', 4.1, 'Perfect classic burger'),
(414, 'Dominos Pizza', '479-215-4770', 'Italian', 2.5, 'Hot dogs are okay'),
(415, 'KFC', '479-216-1063', 'American', 3.5, 'Good for spice lovers'),
(416, 'McDonalds', '479-217-1471', 'American', 3.3, 'Fries are good and fairly crunchy.'),
(417, 'Manchester United Sports Bar', '479-218-8739', 'Multi Cuisine', 3, 'Not going back here, again.'),
(418, 'Tacobell', '479-219-9903', 'American', 4.7, 'Nice restaurant good food & environment and service.'),
(419, 'Hyderabad Biryani', '479-220-8232', 'Indian', 4, ' Kheer was very sweet.'),
(420, 'Green Ginger', '479-221-7072', 'Indian', 3.7, 'Cheeseburgers are fantastic and hefty.'),
(421, 'Subway', '479-223-4255', 'American', 3.5, 'Food ready faster than originally scheduled ready time.'),
(422, 'Leela Palace', '423-219-4358', 'Indian', 4.2, 'Ordered on grubhub for pick up'),
(423, 'Rajdhani', '423-220-9616', 'Indian', 2.5, 'Good crispy noodles'),
(424, 'Khandhani', '423-221-9540', 'Indian', 3.5, 'House Special Fried rice was nice'),
(425, 'Little Italy', '423-222-6135', 'Italian', 4.2, ' Fast service');
```

11) Inserting values into Restaurant Type table:

```
INSERT INTO restaurant_type_T (restaurant_id,restaurant_type)
VALUES (401,'3 star'),
(402,'5 star'),
(403,'3 star'),
(404,'2 star'),
(405,'5 star'),
(406,'4 star'),
(407,'4 star'),
(408,'2 star'),
(409,'3 star'),
(410,'5 star'),
(411,'2 star'),
(412,'3 star'),
(413,'2 star'),
(414,'4 star'),
(415,'5 star'),
(416,'3 star'),
(417,'4 star'),
(418,'2 star'),
(419,'4 star'),
(420,'5 star'),
(421,'3 star'),
(422,'4 star'),
(423,'3 star'),
(424,'3 star'),
(425,'4 star');
```

12) Inserting values into Restaurant Booking table:

```
INSERT INTO restaurant_booking_T (restaurant_id, customer_id, booked_date, booked_time, reservation_num)
VALUES
(401,101,'2018-04-06','08:49:36',1),
(402,102,'2019-02-13','02:35:48',2),
(403,103,'2018-05-29','19:08:33',3),
(404,104,'2018-05-23','12:49:41',4),
(405,105,'2018-03-26','04:57:07',5),
(406,106,'2019-03-15','09:46:27',6),
(407,107,'2019-04-25','18:55:08',7),
(408,108,'2018-04-24','11:04:28',8),
(409,109,'2018-09-21','08:59:18',9),
(410,110,'2018-06-14','02:02:21',10),
(411,111,'2018-05-14','15:49:25',11),
(412,112,'2019-04-12','17:22:37',12),
(413,113,'2018-08-08','15:57:42',13),
(414,114,'2019-01-03','04:02:02',14),
(415,115,'2018-08-23','03:45:02',15),
(416,116,'2018-10-08','17:58:05',16),
(417,117,'2019-01-03','04:39:10',17),
(418,118,'2018-08-23','22:30:01',18),
(419,119,'2018-10-08','19:47:47',19),
(420,120,'2019-01-04','16:58:15',20),
(421,121,'2018-04-12','21:52:52',21),
(422,122,'2019-01-09','06:42:08',22),
(423,123,'2018-03-21','09:46:21',23),
(424,124,'2018-05-10','11:54:47',24),
(425,125,'2019-04-02','07:27:17',25);
```

13)Inserting values into Debit Card Payment table:

```
INSERT INTO debit_card_payment_t (dpayment_id,debit_card_number) VALUES  
(316,1234234534564560),  
(317,6011178546895840),  
(318,6011821377230930),  
(319,601102117602000),  
(320,6011373700477490),  
(321,6011249521324830),  
(322,4910928338978980),  
(323,4727263726464340),  
(324,4294296376501290),  
(325,4564463448507280),  
(326,4418861812239650),  
(327,5244964209912990),  
(328,5156146401210970),  
(329,5389116636935880),  
(330,5147709584291520),  
(346,5577035756102180),  
(347,3555335862848410),  
(348,3562118978160960),  
(349,3533912539224210),  
(350,3549401568397160);
```

14)Inserting values into Online Application Wallet table:

```
INSERT INTO Online_Application_Wallet_T (oaw_payment_id,wallet_name)  
VALUES(346,'Zomato Wallet'),  
(347,'Zomato Wallet'),  
(348,'Zomato Wallet'),  
(349,'Zomato Wallet'),  
(350,'Zomato Wallet'),  
(336,'Zomato Wallet'),  
(337,'Zomato Wallet'),  
(338,'Zomato Wallet'),  
(339,'Zomato Wallet'),  
(340,'Zomato Wallet'),  
(341,'Zomato Wallet'),  
(342,'Zomato Wallet'),  
(343,'Zomato Wallet'),  
(344,'Zomato Wallet'),  
(345,'Zomato Wallet'),  
(331,'Google Pay'),  
(332,'Google Pay'),  
(333,'Google Pay'),  
(334,'Google Pay'),  
(335,'Google Pay');
```

15)Inserting values into Credit Card Payment table:

```
INSERT INTO credit_card_payment_t(cpayment_id, credit_card_number) VALUES  
(301, 1234567891234567),  
(302, 1234567891234568),  
(303, 1234567891234569),  
(304, 1234567891234561),  
(305, 1234567891234562),  
(306, 1234567891234563),  
(307, 1234567891234564),  
(308, 1234567891234566),  
(309, 1234567891234561),  
(310, 1234567891234562),  
(311, 1234567891234563),  
(312, 1234567891234564),  
(313, 1234567891234565),  
(314, 1234567891234566),  
(315, 1234567891234568),  
(341, 1234567891234566),  
(342, 1234567891234568),  
(343, 1234567891234562),  
(344, 1234567891234561),  
(345, 1234567891234563);
```

16)Inserting values into Cash On Delivery table:

```
INSERT INTO cash_on_delivery_payment_t(codpayment_id) VALUES  
(336),  
(337),  
(338),  
(339),  
(340),  
(351),  
(352),  
(353),  
(354),  
(355),  
(356),  
(357),  
(358),  
(359),  
(360),  
(361),  
(362),  
(363),  
(364),  
(365);
```

Project Report - BAN 610

TABLE RECORDS

1. Customer_t

1 • Select * from customer_t;

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
customer_id	first_name	last_name	email_id	address	phone_number
101	Bill	Doll	bill@gmail.com	300 West,Indiranagar, Bangalore 560008,India	455-920-6782
102	Karthik	Turburn	karthik@gmail.com	250 North, Indiranagar, Bangalore 560008,India	455-921-6739
103	Mary	Lee	mary@gmail.com	200 South, Indiranagar, Bangalore 560008,India	455-908-6811
104	Jane	Smith	jane@gmail.com	120 University, MG Road, Mumbai 762112,India	455-908-6900
105	Vincent	Smith	vincent@gmail.com	200 Maple, MG Road, Mumbai 762112,India	455-908-6814
106	David	Brown	david@gmail.com	300 West, MG Road, Mumbai 762112,India	455-908-6888
107	Carl	Smith	carl@gmail.com	250 North, MG Road, Mumbai 762112,India	455-908-6670
108	John	Doll	doll@gmail.com	200 South, MG Road, Mumbai 762112,India	455-908-6123
109	Jim	Chris	chris@gmail.com	120 University, Castle Road, Hyderabad 87102,India	455-908-6321
110	Joyce	French	joyce@gmail.com	200 Main, Castle Road, Hyderabad 87102,India	455-908-6456
111	Manav	English	manav@gmail.com	250 State, Castle Road, Hyderabad 87102,India	455-908-6653
112	Tom	Borg	jtom@gmail.com	3000 Sunset, Castle Road, Hyderabad 87102,India	455-908-6654
113	Donald	Thomas	donald@gmail.com	1200 Hollywood,Majestic Road, Delhi 87123,India	455-908-6546
114	Vishwas	Borg	vishwas@gmail.com	100 College,Majestic Road, Delhi 87123,India	455-908-6711
115	John	Doe	john@gmail.com	200 Maple Street, Indiranagar, Bangalore 560008,India	455-908-6781
116	Brandon	French	bran@gmail.com	200 Main, Windtunnel Road, Chennai 76112,India	455-908-6712
117	Victor	English	victor@gmail.com	250 State, Windtunnel Road, Chennai 76112,India	455-908-67713
118	Alice	Borg	alice@gmail.com	3000 Sunset, Windtunnel Road, Chennai 76112,India	455-908-6715
119	Margaret	Thomas	amrg@gmail.com	1200 Hollywood, Windtunnel Road, Chennai 76112,India	455-908-6714
120	Aston	Borg	aston@gmail.com	100 College, Windtunnel Road, Chennai 76112,India	455-908-6717
121	Joel	French	joel@gmail.com	200 Main, Brigade Road, Kolkata 88111,India	455-908-6811
122	Robert	English	robert@gmail.com	250 State, Brigade Road, Kolkata 88111,India	455-908-6812
123	Leonardo	Borg	leo@gmail.com	3000 Sunset, Brigade Road, Kolkata 88111,India	455-908-6813
124	Bruce	Thomas	bruce@gmail.com	1200 Hollywood, Brigade Road, Kolkata 88111,India	455-908-6814
125	Shawn	Borg	shawn@gmail.com	100 College, Brigade Road, Kolkata 88111,India	455-908-6815

2. subscriptions_T

4 • Select * from subscriptions_T;

Result Grid		Filter Rows:	Edit:	Export/Import:	
subscription_id	date_placed	date_validity	cost	subscription_name	customer_id
501	2020-10-24	2021-01-23	300	Zomato Pro	101
502	2020-09-23	2020-12-22	300	Zomato Pro	102
503	2020-08-22	2020-11-21	300	Zomato Pro	103
504	2020-07-21	2020-10-20	300	Zomato Pro	104
505	2020-06-20	2020-09-19	300	Zomato Pro	105
506	2020-04-18	2020-07-17	300	Zomato Pro	107
507	2020-03-17	2020-06-16	300	Zomato Pro	108
508	2020-02-16	2020-05-15	300	Zomato Pro	109
509	2020-01-15	2020-04-14	300	Zomato Pro	110
510	2020-11-14	2021-02-13	300	Zomato Pro	111
511	2020-12-13	2021-03-12	300	Zomato Pro	112
512	2021-01-12	2021-04-11	300	Zomato Pro	113
513	2021-02-11	2021-05-10	300	Zomato Pro	114
514	2021-03-10	2021-06-09	300	Zomato Pro	115
515	2020-05-19	2020-08-18	300	Zomato Pro	106
516	2019-11-14	2020-02-13	500	Zomato Pro	116
517	2019-12-13	2020-03-12	500	Zomato Pro	117
518	2019-01-12	2019-04-11	500	Zomato Pro	118
519	2019-02-11	2019-05-10	500	Zomato Pro	119
520	2019-03-10	2019-06-09	500	Zomato Pro	120
521	2018-11-14	2019-02-13	400	Zomato Pro	121
522	2018-12-13	2019-03-12	400	Zomato Pro	122
523	2018-01-12	2018-04-11	400	Zomato Pro	123
524	2018-02-11	2018-05-10	400	Zomato Pro	124
525	2018-03-10	2018-06-09	400	Zomato Pro	125

Project Report - BAN 610

3. customer_support_t

4 • `Select * from customer_support_t;`

	query_id	customer_executive_id	query_description	resolution_status
▶	601	901	Unable to do payment	In Progress
	602	902	Unable to add items to cart	In Progress
	603	903	Unable to do add subscription	In Progress
	604	904	Need more info about payment options	In Progress
	605	905	Zomato offers on various food items in restaura...	In Progress
	606	907	Unable to order food	Closed
	607	908	Zomato subscription page not loading	Closed
	608	909	Technical error appearing on adding to cart	Closed
	609	910	Need more information on offers during festival	Closed
	610	911	Did not get the order despite ordering 2 hours ago	Closed
	611	912	Not happy with the food delivered. Need refund	Escalated
	612	913	Unable to reach the delivery person	Escalated
	613	914	Restaurant is not allowing to reserve the table	In Progress
	614	915	Order contains fewer items than what was ordered	Escalated
	615	906	Restaurant offers in specific location	Closed
	616	917	Offer for online delivery of pizza	Closed
	617	918	Any more subscriptions in the future	Closed
	618	919	How to enroll as a vendor for zomato	Closed
	619	920	Not happy with food quality	In Progress
	620	921	How much is the maximum discount using subscription	Closed
	621	922	When is the validity ending for subscription	Closed
	622	923	Can zomato update the validity of subscription	Closed
	623	916	Unable to get subscription activated	Escalated
	624	924	Does zomato deliver during late night hours	Closed
	625	925	What is the last order time for meals	Closed

4. customer_query_t

4 • `Select * from customer_query_t;`

	query_id	customer_id	resolution_date
▶	601	101	2021-03-10
	602	101	2021-02-10
	603	101	2021-01-11
	604	101	2021-04-13
	605	101	2021-01-01
	606	103	2021-04-10
	607	102	2020-03-10
	608	102	2020-02-10
	609	103	2020-01-11
	610	104	2020-04-13
	611	105	2020-01-01
	612	103	2020-04-10
	613	107	2019-03-10
	614	108	2019-02-10
	615	109	2019-01-11
	616	110	2019-04-13
	617	107	2019-08-01
	618	107	2019-12-10
	619	113	2018-03-10
	620	114	2018-02-10
	621	115	2018-01-11
	622	115	2018-10-13
	623	116	2018-09-01
	624	118	2018-07-10
	625	125	2018-06-10

Project Report - BAN 610

5. delivery_t

```
4 • Select * from delivery_t;
```

	delivery_id	first_name	last_name	phone_number	delivery_time
▶	701	Karans	jindal	123-908-6789	2008-10-25 00:00:00
	702	Rahu	Malik	123-908-6789	2008-10-26 00:00:00
	703	Stephens	chau	123-908-6789	2008-11-24 00:00:00
	704	Karthik	Malika	187-888-87	2008-12-24 00:00:00
	705	Karan	chau	333-908-6789	2008-01-24 00:00:00
	706	andrew	patelll	763-908-6789	2008-02-24 00:00:00
	707	nice	guy	555-908-6789	2008-03-24 00:00:00
	708	apoovraaaa	roy	553-908-6789	2008-10-14 00:00:00
	709	raj	patel	123-508-6789	2009-10-24 00:00:00
	710	sakshiii	jain	153-908-6789	2010-10-24 00:00:00
	711	prach	gera	456-908-6789	2011-10-24 00:00:00
	712	parviii	goyal	123-999-6789	2007-10-24 00:00:00
	713	ruchiii	bajaj	123-000-6789	2008-11-04 00:00:00
	714	anitaaa	garg	123-000-6789	2001-11-24 00:00:00
	715	ankit	gargs	189-888-6789	2008-06-18 00:00:00
	716	ankura	mehta	176-908-6789	2000-11-24 00:00:00
	717	ragsi	Munja	563-908-8888	2002-10-29 00:00:00
	718	sonam	saini Mun...	651-788-6789	2008-12-22 00:00:00
	719	Karan	sachdevaaa	563-908-6789	2008-01-26 00:00:00
	720	nitish	sethi	981-108-6789	2008-10-07 00:00:00
	721	bharat	kumar	989-908-6789	2018-10-24 00:00:00
	722	sakshi	kumar	123-908-5432	2017-10-24 00:00:00
	723	vineet	pushkar	123-908-8976	2016-10-24 00:00:00
	724	Rahul	pushkar	415-808-6789	1998-10-24 00:00:00
	725	vinod	bajaj	213-908-6789	2007-11-24 00:00:00
	726	Karan	jindal	123-908-6789	2006-10-24 00:00:00

6. order_t

```
4 • Select * from order_t;
```

	order_id	order_date	order_price	customer_id	delivery_id
▶	201	2008-10-24	100	101	701
	202	2008-10-21	200	102	702
	203	2008-10-22	300	103	703
	204	2008-10-22	100	104	704
	205	2008-10-24	1400	105	705
	206	2008-10-24	1010	106	706
	207	2008-10-27	1200	107	707
	208	2008-10-30	100	108	708
	209	2008-11-01	200	109	709
	210	2008-11-05	300	110	710
	211	2008-10-24	1400	111	711
	212	2008-10-21	500	112	711
	213	2008-10-22	1300	113	713
	214	2008-10-22	1080	114	714
	215	2008-10-24	1800	115	715
	216	2008-10-24	1020	116	716
	217	2008-10-27	1010	117	717
	218	2008-10-30	1300	118	718
	219	2008-11-01	1200	119	719
	220	2008-11-05	1100	120	720
	221	2008-10-24	1100	121	721
	222	2008-10-21	1010	122	722
	223	2008-10-22	1005	123	723
	224	2008-10-22	200	124	724
	225	2008-10-24	300	125	725
	226	2008-10-24	100	101	726

7. order_details_t

```
4 • Select * from order_details_t;
```

Result Grid | Filter Rows: [] | Export

order_id	order_details
201	paneer
202	Pizza
203	Fried rice
204	brinjal
205	Dal
206	potato
207	pizza
208	Fish
209	Chicken
210	chicken
211	fish
212	burrito
213	Paneer
214	Lamb
215	pizza
216	steak
217	chicken
218	pasta
219	beef
220	steak
221	lamb
222	paneer
223	pasta
224	cheese
225	pizza
226	paneer

8. payment_t

```
4 • Select * from payment_t;
```

Result Grid | Filter Rows: [] | Edit: [] | Export/Import: [] | Wrap Cell Content

payment_id	cost	payment_type	customer_id	order_id
301	200	Credit Card	101	201
302	300	Credit Card	102	202
303	200	Credit Card	103	203
304	400	Credit Card	104	204
305	500	Credit Card	105	205
306	200	Credit Card	106	206
307	300	Credit Card	107	207
308	200	Credit Card	108	208
309	400	Credit Card	109	209
310	500	Credit Card	110	210
311	200	Credit Card	111	211
312	300	Credit Card	112	212
313	200	Credit Card	113	213
314	400	Credit Card	114	214
315	500	Credit Card	115	215
316	200	Debit Card	116	216
317	300	Debit Card	117	217
318	200	Debit Card	118	218
319	400	Debit Card	119	219
320	500	Debit Card	120	220
321	200	Debit Card	121	221
322	300	Debit Card	122	222
323	200	Debit Card	123	223
324	400	Debit Card	124	224
325	500	Debit Card	125	225
326	200	Debit Card	101	226

Project Report - BAN 610

9. notification_t

```
4 •   Select * from notification_t;
```

Result Grid | Filter Rows: | Edit: | Export: |

	message	customer_id
▶	Half Price Offer starts tonight	101
	Weekend Offer on Pizzas!!!!!!	102
	Zomato Pro to cost 10% less!!!!	103
	Book a Restaurant table today and get 25% dis...	104
	Mandarin is now offering a Full course meal at 5...	105
	Zomato pro validity is not extended to 4 months	106
	Burgers at Burger king to cost less from tomorrow	107
	Biryani zone is now at your nearest location!!!!	108
	Half Price Offer starts tonight	109
	Weekend Offer on Pizzas!!!!!!	110
	Zomato Pro to cost 10% less!!!!	111
	Book a Restaurant table today and get 25% dis...	112
	Mandarin is now offering a Full course meal at 5...	113
	Zomato pro validity is not extended to 4 months	114
	Burgers at Burger king to cost less from tomorrow	115
	Biryani zone is now at your nearest location!!!!	116
	Half Price Offer starts tonight	117
	Weekend Offer on Pizzas!!!!!!	118
	Zomato Pro to cost 10% less!!!!	119
	Book a Restaurant table today and get 25% dis...	120

10. restaurant_T

```
119 •   select * from restaurant_t;
120
121
122
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	restaurant_id	restaurant_name	contact_information	cusine	ratings	reviews
▶	401	Biryani Zone	479-200-2755	Indian	3.5	Nice restaurant good food & environment and service.
	402	Barbeque Nation	479-201-3613	Multi cusine	3.3	For me, it was little spicy.
	403	Punjabi Rasoi	479-202-0130	Noth Indian	4.5	I ordered Fish Fillet and House Special Fried rice. very good
	404	Toit	479-203-6427	Mexican	4.1	This location did not disappoint!
	405	Vapour	479-204-1014	Italian	4.1	We enjoyed our food our tater tots and mozzarella sticks
	406	Nandoos	479-205-7475	South African	2.9	Disappointing flavor on almost everything I got
	407	Lazy Suzy	479-206-6425	Mexican	3.5	fast food did not sound appealing.
	408	Shangrilla	479-207-1184	Chinese	3.5	Nice resturant good food .
	409	Mainland China	479-208-9302	Chinese	3	Everything here cost \$60 total. Not nearly enough.
	410	Copper Chimney	479-209-0151	German	3.8	everything was good!
	411	Absolute Barbeque	479-212-3920	Multi Cusine	4	Already a tight parking lot!
	412	The Taj	479-213-3474	Indian	2	Disappointingly small portions
	413	California Burritos	479-214-4992	American	4.1	Perfect classic burger
	414	Dominos Pizza	479-215-4770	Italian	2.5	Hot dogs are okay
	415	KFC	479-216-1063	American	3.5	Good for spice lovers
	416	McDonalds	479-217-1471	American	3.3	Fries are good and fairly crunchy.
	417	Manchester Unit...	479-218-8739	Multi Cuisine	3	Not going back here, again.
	418	Tacobell	479-219-9903	American	4.7	Nice resturant good food & environment and service.
	419	Hyderabadi Biryani	479-220-8232	Indian	4	Kheer was very sweet.
	420	Green Ginger	479-221-7072	Indian	3.7	Cheesburgers are fantastic and hefty.
	421	Subway	479-223-4255	American	3.5	Food ready faster than originally scheduled ready time.
	422	Leela Palace	423-219-4358	Indian	4.2	Ordered on grubhub for pick up
	423	Rajdhani	423-220-9616	Indian	2.5	Good crispy noodles
	424	Khandhani	423-221-9540	Indian	3.5	House Special Fried rice was nice
	425	Little Italy	423-222-6135	Italian	4.2	Fast service

11. restaurant_type_T

```
4 • Select * from restaurant_type_T;
```

	restaurant_id	restaurant_type
▶	401	3 star
	402	5 star
	403	3 star
	404	2 star
	405	5 star
	406	4 star
	407	4 star
	408	2 star
	409	3 star
	410	5 star
	411	2 star
	412	3 star
	413	2 star
	414	4 star
	415	5 star
	416	3 star
	417	4 star
	418	2 star
	419	4 star
	420	5 star
	421	3 star
	422	4 star
	423	3 star
	424	3 star
	425	4 star

12. restaurant_booking_T

```
4 • Select * from restaurant_booking_T;
```

	restaurant_id	customer_id	booked_date	booked_time	reservation_num
▶	401	101	2018-04-06	08:49:36.000000	1
	402	102	2019-02-13	02:35:48.000000	2
	403	103	2018-05-29	19:08:33.000000	3
	404	104	2018-05-23	12:49:41.000000	4
	405	105	2018-03-26	04:57:07.000000	5
	406	106	2019-03-15	09:46:27.000000	6
	407	107	2019-04-25	18:55:08.000000	7
	408	108	2018-04-24	11:04:28.000000	8
	409	109	2018-09-21	08:59:18.000000	9
	410	110	2018-06-14	02:02:21.000000	10
	411	111	2018-05-14	15:49:25.000000	11
	412	112	2019-04-12	17:22:37.000000	12
	413	113	2018-08-08	15:57:42.000000	13
	414	114	2019-01-03	04:02:02.000000	14
	415	115	2018-08-23	03:45:02.000000	15
	416	116	2018-10-08	17:58:05.000000	16
	417	117	2019-01-03	04:39:10.000000	17
	418	118	2018-08-23	22:30:01.000000	18
	419	119	2018-10-08	19:47:47.000000	19
	420	120	2019-01-04	16:58:15.000000	20
	421	121	2018-04-12	21:52:52.000000	21
	422	122	2019-01-09	06:42:08.000000	22
	423	123	2018-03-21	09:46:21.000000	23
	424	124	2018-05-10	11:54:47.000000	24
	425	125	2019-04-02	07:27:17.000000	25

13. debit_card_payment_T

```
4 • Select * from debit_card_payment_T;
```

Result Grid | Filter Rows: | Edit: |

	dpayment_id	debit_card_number
▶	316	1234234534564560
	317	6011178546895840
	318	6011821377230930
	319	6011102117602000
	320	6011373700477490
	321	6011249521324830
	322	4910928338978980
	323	4727263726464340
	324	4294296376501290
	325	4564463448507280
	326	4418861812239650
	327	5244964209912990
	328	5156146401210970
	329	5389116636935880
	330	5147709584291520
	346	5577035756102180
	347	3555335862848410
	348	3562118978160960
	349	3533912539224210
	350	3549401568397160

14. online_application_wallet_T

```
4 • Select * from online_application_wallet_T;
```

Result Grid | Filter Rows: | Edit: |

	oaw_payment_id	wallet_name
▶	331	Google Pay
	332	Google Pay
	333	Google Pay
	334	Google Pay
	335	Google Pay
	336	Zomato Wallet
	337	Zomato Wallet
	338	Zomato Wallet
	339	Zomato Wallet
	340	Zomato Wallet
	341	Zomato Wallet
	342	Zomato Wallet
	343	Zomato Wallet
	344	Zomato Wallet
	345	Zomato Wallet
	346	Zomato Wallet
	347	Zomato Wallet
	348	Zomato Wallet
	349	Zomato Wallet
	350	Zomato Wallet

15. credit_card_payment_t

```
4 • Select * from credit_card_payment_t;
```

The screenshot shows a database result grid titled "Result Grid". It has two columns: "cpayment_id" and "credit_card_number". The "cpayment_id" column contains values from 301 to 345. The "credit_card_number" column contains values consisting of 123456789 followed by a sequence of digits (e.g., 1234567891234567, 1234567891234568, ..., 1234567891234563). The rows are numbered on the left.

	cpayment_id	credit_card_number
▶	301	1234567891234567
	302	1234567891234568
	303	1234567891234569
	304	1234567891234561
	305	1234567891234562
	306	1234567891234563
	307	1234567891234564
	308	1234567891234566
	309	1234567891234561
	310	1234567891234562
	311	1234567891234563
	312	1234567891234564
	313	1234567891234565
	314	1234567891234566
	315	1234567891234568
	341	1234567891234566
	342	1234567891234568
	343	1234567891234562
	344	1234567891234561
	345	1234567891234563

16. cash_on_delivery_payment_t

```
4 • Select * from cash_on_delivery_payment_t;
```

The screenshot shows a database result grid titled "Result Grid". It has one column: "codpayment_id". The values range from 336 to 365. The rows are numbered on the left.

	codpayment_id
▶	336
	337
	338
	339
	340
	351
	352
	353
	354
	355
	356
	357
	358
	359
	360
	361
	362
	363
	364
	365

QUERIES

QUERY – 1
(ORDER HISTORY OF A PARTICULAR USER)
- USING NATURAL JOIN

```
use zomato_order;  
SELECT ot.order_id,ot.order_date,ot.order_price,odt.order_details  
FROM order_t ot NATURAL JOIN order_details_t odt  
where ot.order_id=odt.order_id  
and customer_id=101  
order by order_id;
```

	order_id	order_date	order_price	order_details
▶	201	2008-10-24	100	paneer
	226	2008-10-24	100	paneer
	251	2008-10-24	1020	Brinjal

QUERY-2
(RESTAURANT LIST ACCORDING TO CUISINE)
- USING ORDER BY & DESC

```
Select rt.restaurant_name,rt.contact_information,
rt.ratings,
rt.reviews,rt.cuisine
from restaurant_t rt
where cuisine='Indian'
order by ratings desc;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

restaurant_name	contact_information	ratings	reviews	cuisine
Leela Palace	423-219-4358	4.2	Ordered on grubhub for pick up	Indian
Hyderabadi Biryani	479-220-8232	4	Kheer was very sweet.	Indian
Green Ginger	479-221-7072	3.7	Cheeseburgers are fantastic and hefty.	Indian
Biryani Zone	479-200-2755	3.5	Nice restaurant good food & environment and se...	Indian
Khandhani	423-221-9540	3.5	House Special Fried rice was nice	Indian
Rajdhani	423-220-9616	2.5	Good crispy noodles	Indian
The Taj	479-213-3474	2	Disappointingly small portions	Indian

QUERY - 3
(COUNT OF MONTHLY ORDERS PLACED IN AN YEAR)
- USING STORED PROCEDURE

```

DELIMITER //
CREATE PROCEDURE storedproc_monthlyordernumber(IN yearid INT(10))
begin
SELECT MONTHNAME(order_date),count(*) FROM zomato_order.order_t
where year(order_date)=yearid
group by month(order_date);
END //
DELIMITER ;

```

```

call storedproc_monthlyordernumber(2008);

```

	MONTHNAME(order_date)	count(*)
▶	October	53
	November	12

QUERY – 4

**(PEOPLE WHO HAVE RECEIVED A PARTICULAR MESSAGE VIA NOTIFICATION) -
USING EXISTS**

```
SELECT customer_t.first_name ,customer_id FROM customer_T
where EXISTS
    (SELECT notification_T.customer_id FROM notification_T
     WHERE message='half price offer starts tonight'
      and customer_t.customer_id=notification_T.customer_id);
```

	first_name	customer_id
▶	Bill	101
	Jim	109
	Victor	117
*	NULL	NULL

QUERY – 5**(SUBSCRIPTIONS taken AFTER PARTICULAR DATE in a city) – USING equi join**

```

SELECT customer_t.*
FROM customer_t,subscriptions_t
WHERE
customer_t.customer_id=subscriptions_t.customer_id
AND
customer_t.address LIKE '%Bangalore%' AND
subscriptions_t.date_placed>'2020-01-18';

```

	customer_id	first_name	last_name	email_id	address	phone_number
▶	101	Bill	Doll	bill@gmail.com	300 West,Indiranagar, Bangalore 560008,India	455-920-6782
	102	Karthik	Turburn	karthik@gmail.com	250 North, Indiranagar, Bangalore 560008,India	455-921-6739
	103	Mary	Lee	mary@gmail.com	200 South, Indiranagar, Bangalore 560008,India	455-908-6811
	115	John	Doe	john@gmail.com	200 Maple Street, Indiranagar, Bangalore 560008,India	455-908-6781

QUERY – 6
(COUNT OF customer queries>1)

- USING NATURAL JOIN

```
SELECT customer_t.customer_id,
customer_t.first_name, customer_t.last_name,
COUNT(*) as Count_of_Questions
FROM customer_t NATURAL JOIN
customer_support_t NATURAL JOIN
customer_query_t
GROUP BY customer_t.customer_id
HAVING count(customer_query_t.customer_id)>1;
```

	customer_id	first_name	last_name	Count_of_Questions
▶	101	Bill	Doll	5
	103	Mary	Lee	3
	102	Karthik	Turburn	2
	107	Carl	Smith	3
	115	John	Doe	2

QUERY – 7

(FIND AVERAGE cost FOR ORDERS IN A city avg payment) – USING VIEW

```
CREATE VIEW v_city_customers AS
SELECT "Mumbai" as City,AVG(payment_T.cost)
FROM payment_T, customer_t
WHERE customer_t.customer_id=payment_T.customer_id AND
customer_t.address LIKE '%Mumbai%' ;
SELECT * FROM v_city_customers;
```

City	avg(payment_T.cost)
Mumbai	365.2

QUERY – 8**details of delivery agent who Delivered orders above a price– STORED PROCEDURE**

```

DELIMITER //
CREATE PROCEDURE sp_delivery_price(IN price_x varchar(11))
BEGIN
    select delivery_t.delivery_id,
    delivery_t.first_name,
    delivery_t.last_name,order_t.order_price
    from delivery_t INNER JOIN order_t ON
    delivery_t.delivery_id = order_t.delivery_id
    AND order_t.order_price>price_x;
END//
DELIMITER ;

```

```

CALL sp_delivery_price(1300);

```

	delivery_id	first_name	last_name	order_price
▶	705	Karan	chau	1400
	711	prach	gera	1400
	715	ankit	gargs	1800
	730	Karan	chau	1400
	736	prachi	Malik	1400
	740	ank	garg	1800

QUERY – 9
(LISTING ALL CUSTOMERS WHO BOOKED TABLES) – USING INNER JOIN

```

SELECT c.CUSTOMER_ID,first_name,
restaurant_name,booked_date,BOOKED_TIME
FROM restaurant_T r INNER JOIN customer_t c
INNER JOIN restaurant_booking_T rb
ON c.CUSTOMER_ID=rb.customer_id AND
rb.restaurant_id=r.restaurant_id
WHERE booked_date
BETWEEN '2018-03-01' AND '2018-10-01'
AND BOOKED_TIME
BETWEEN '10:00:00' AND '23:00:00'
AND first_name
LIKE 'J%';

```

	CUSTOMER_ID	first_name	restaurant_name	booked_date	BOOKED_TIME
▶	104	Jane	Toit	2018-05-23	12:49:41.000000
	108	John	Shangrilla	2018-04-24	11:04:28.000000
	121	Joel	Subway	2018-04-12	21:52:52.000000

QUERY – 10

(listing customers who visited restaurant with rating >2.5 in a city) – USING NATURAL JOIN

```
SELECT c.first_name,restaurant_name,
ratings,reviews,address
FROM restaurant_booking_t
NATURAL JOIN restaurant_t
NATURAL JOIN customer_T c
where ratings>2.5 AND
address LIKE '%HYDERABAD%' ;
```

first_name	restaurant_name	ratings	reviews	address
Jim	Mainland China	3	Everything here cost \$60 total. Not nearly eno...	120 University, Castle Road, Hyderabad 87102,India
Joyce	Copper Chimney	3.8	everything was good!	200 Main, Castle Road, Hyderabad 87102,India
Manav	Absolute Barbeque	4	Already a tight parking lot!	250 State, Castle Road, Hyderabad 87102,India

QUERY – 11**(DETAILS OF CUSTOMERS WHO PLACED ORDERS USING DIFFERENT PAYMENT TYPES) – USING VIEW**

```

CREATE VIEW filter_by_payment AS
SELECT PAYMENT_TYPE,first_name,
order_details,order_date,
order_price FROM payment_T
NATURAL JOIN
customer_t NATURAL JOIN
ORDER_T NATURAL JOIN
order_details_T ;

SELECT * FROM filter_by_payment WHERE PAYMENT_TYPE='google pay';

```

	PAYMENT_TYPE	first_name	order_details	order_date	order_price
▶	Google Pay	David	potato	2008-10-24	1010
	Google Pay	Carl	dal	2008-10-27	1200
	Google Pay	John	Fish	2008-10-30	100
	Google Pay	Jim	Chicken	2008-11-01	200
	Google Pay	Joyce	Veg	2008-11-05	300

DISPLAYING SCHEMA:

SCHEMAS

Filter objects

▼ **zomato_order**

- ▼ **Tables**
 - ▶ cash_on_delivery_payment_t
 - ▶ credit_card_payment_t
 - ▶ customer_query_t
 - ▶ customer_support_t
 - ▶ customer_t
 - ▶ debit_card_payment_t
 - ▶ delivery_t
 - ▶ notification_t
 - ▶ online_application_wallet_t
 - ▶ order_details_t
 - ▶ order_t
 - ▶ payment_t
 - ▶ restaurant_booking_t
 - ▶ restaurant_t
 - ▶ restaurant_type_t
 - ▶ subscriptions_t
- ▼ **Views**
 - ▶ filter_by_payment
 - ▶ v_city_customers
- ▼ **Stored Procedures**
 - ▶ sp_delivery_price
 - ▶ storedproc_monthlyordernumber

Administration Schemas

The screenshot shows a database schema browser interface. At the top, there's a header bar with the title 'SCHEMAS' and a search bar labeled 'Filter objects'. Below this is a tree view of the 'zomato_order' schema. The 'Tables' node is expanded, showing 14 table entries. The 'Views' and 'Stored Procedures' nodes are also expanded, showing their respective contents. At the bottom of the interface, there are two tabs: 'Administration' and 'Schemas', with 'Schemas' being the active tab.