Case Study Report

Pizza Ordering System

Teacher: Ms Rupali Sawant

Students: Leena Badgujar (2018110006)

Navya Fadia (2018110013)

Harshada Patil (2018110035)

Subject: OE-8 Database Management Systems

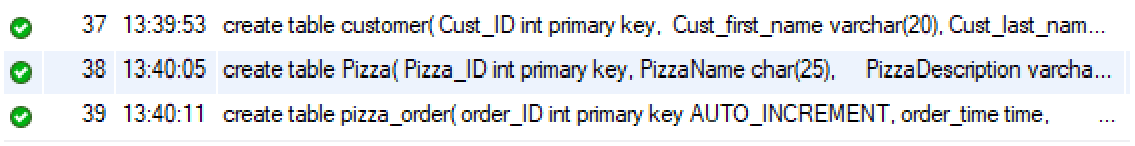
Index

1. Data Definition Language

A) Create

- Customer
```
create table customer(
          Cust_ID int primary key,
          Cust_first_name varchar(20),
          Cust_last_name varchar(20),
          Cust_address varchar(50),
           Cust_email varchar(20),
           Cust_Phone bigint
            );
```
- Pizza
```
create table Pizza(
          Pizza_ID int primary key,
          PizzaName char(25),
          PizzaDescription varchar(100),
          Cost int
          );
```
- Pizza Order
```
create table pizza_order(
          order_ID int primary key AUTO_INCREMENT,
          order_time time,
          PizzaID int,
           PizzaName char(25),
           Cost int
           );
```

Creation of table



```
37  13:39:53  create table customer( Cust_ID int primary key,  Cust_first_name varchar(20), Cust_last_nam...
38  13:40:05  create table Pizza( Pizza_ID int primary key, PizzaName char(25),     PizzaDescription varcha...
39  13:40:11  create table pizza_order( order_ID int primary key AUTO_INCREMENT, order_time time,       ...
```

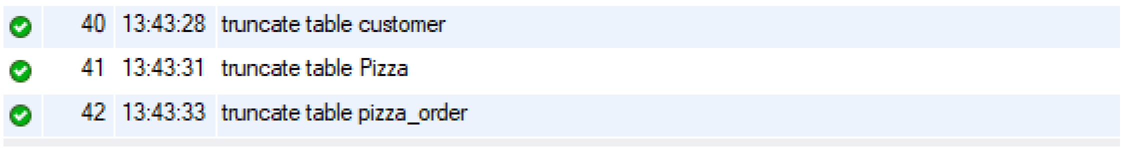
B) Truncate

- Customer
  truncate table customer;

- Pizza
  truncate table Pizza;

- Pizza order
  truncate table pizza_order;

Truncation of tables



```
✓  40  13:43:28  truncate table customer
✓  41  13:43:31  truncate table Pizza
✓  42  13:43:33  truncate table pizza_order
```

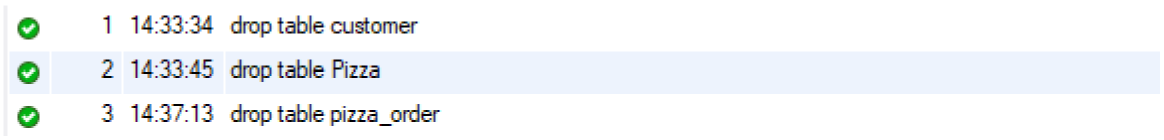
C) Drop

- Customer
  drop table customer;

- Pizza
  drop table Pizza;

- Pizza order
  drop table pizza_order;

Dropping of tables



```
✓  1  14:33:34  drop table customer
✓  2  14:33:45  drop table Pizza
✓  3  14:37:13  drop table pizza_order
```

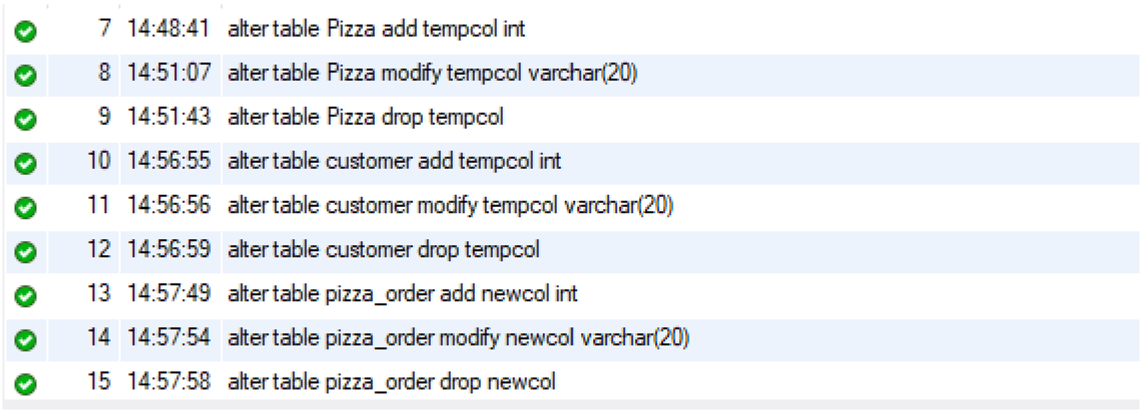
D) Alter

- Customer
  alter table customer add tempcol int;
  alter table customer modify tempcol varchar(20);
  alter table customer drop tempcol;

- Pizza
  alter table Pizza add tempcol int;
  alter table Pizza modify tempcol varchar(20);
  alter table Pizza drop tempcol;

- Pizza order
  alter table pizza_order add newcol int;
  alter table pizza_order modify newcol varchar(20);
  alter table pizza_order drop newcol;

Altering tables



```
✓  7  14:48:41  alter table Pizza add tempcol int
✓  8  14:51:07  alter table Pizza modify tempcol varchar(20)
✓  9  14:51:43  alter table Pizza drop tempcol
✓ 10  14:56:55  alter table customer add tempcol int
✓ 11  14:56:56  alter table customer modify tempcol varchar(20)
✓ 12  14:56:59  alter table customer drop tempcol
✓ 13  14:57:49  alter table pizza_order add newcol int
✓ 14  14:57:54  alter table pizza_order modify newcol varchar(20)
✓ 15  14:57:58  alter table pizza_order drop newcol
```

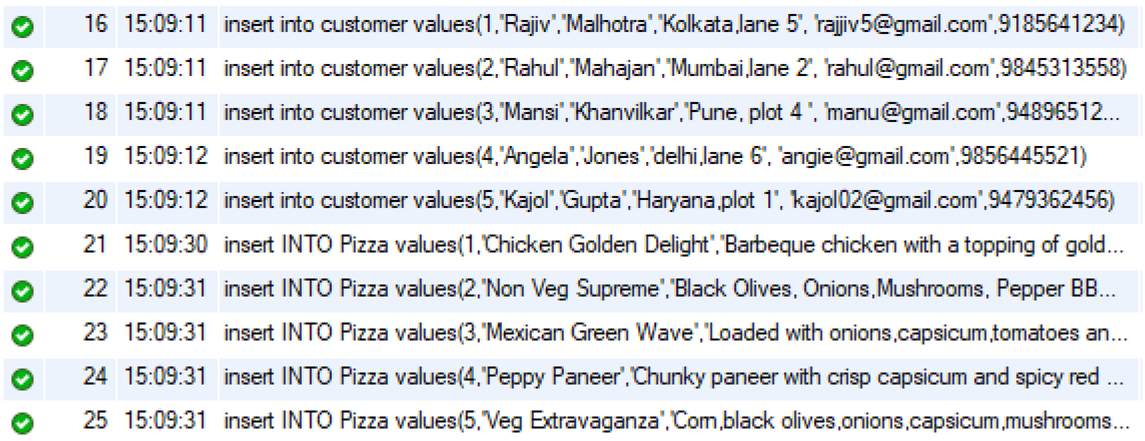
2. Data Manipulation Language

A) Insert

- Customer
  insert into customer values(1,'Rajiv','Malhotra','Kolkata,lane 5', 'rajjiv5@gmail.com',9185641234);
  insert into customer values(2,'Rahul','Mahajan','Mumbai,lane 2', 'rahul@gmail.com',9845313558);
  insert into customer values(3,'Mansi','Khanvilkar','Pune, plot 4 ', 'manu@gmail.com',9489651232);
  insert into customer values(4,'Angela','Jones','delhi,lane 6', 'angie@gmail.com',9856445521);
  insert into customer values(5,'Kajol','Gupta','Haryana,plot 1', 'kajol02@gmail.com',9479362456);

- Pizza
  insert INTO Pizza values(1,'Chicken Golden Delight','Barbeque chicken with a topping of golden corn loaded with extra cheese.',250);
  insert INTO Pizza values(2,'Non Veg Supreme', 'Black Olives, Onions,Mushrooms, Pepper BBQ Chicken, Peri-Peri Chicken',400);
  insert INTO Pizza values(3,'Mexican Green Wave','Loaded with onions,capsicum,tomatoes and jalapeno with Mexican herbs.',200);
  insert INTO Pizza values(4,'Peppy Paneer','Chunky paneer with crisp capsicum and spicy red pepper - quite a mouthful!',300);
  insert INTO Pizza values(5,'Veg Extravaganza','Corn,black olives,onions,capsicum,mushrooms,tomatoes,jalapeno with cheese.',400);

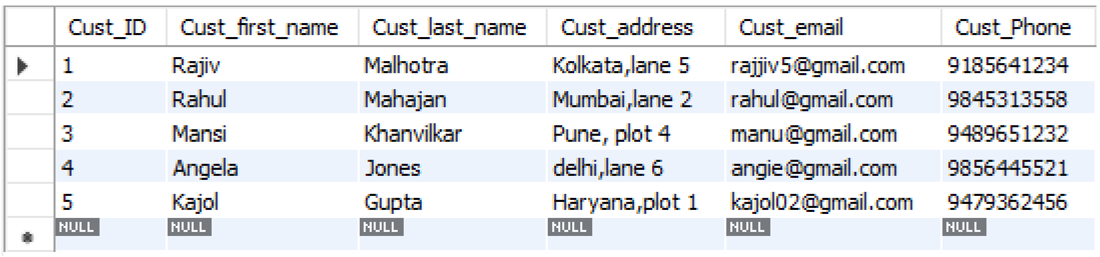
Inserting into table



```
16  15:09:11  insert into customer values(1,'Rajiv','Malhotra','Kolkata,lane 5', 'rajjiv5@gmail.com',9185641234)
17  15:09:11  insert into customer values(2,'Rahul','Mahajan','Mumbai,lane 2', 'rahul@gmail.com',9845313558)
18  15:09:11  insert into customer values(3,'Mansi','Khanvilkar','Pune, plot 4 ', 'manu@gmail.com',94896512...
19  15:09:12  insert into customer values(4,'Angela','Jones','delhi,lane 6', 'angie@gmail.com',9856445521)
20  15:09:12  insert into customer values(5,'Kajol','Gupta','Haryana,plot 1', 'kajol02@gmail.com',9479362456)
21  15:09:30  insert INTO Pizza values(1,'Chicken Golden Delight','Barbeque chicken with a topping of gold...
22  15:09:31  insert INTO Pizza values(2,'Non Veg Supreme','Black Olives, Onions,Mushrooms, Pepper BB...
23  15:09:31  insert INTO Pizza values(3,'Mexican Green Wave','Loaded with onions,capsicum,tomatoes an...
24  15:09:31  insert INTO Pizza values(4,'Peppy Paneer','Chunky paneer with crisp capsicum and spicy red ...
25  15:09:31  insert INTO Pizza values(5,'Veg Extravaganza','Corn,black olives,onions,capsicum,mushrooms...
```

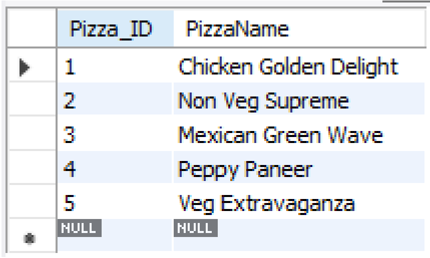
B) Select

- Customer
  select* from customer;



	Cust_ID	Cust_first_name	Cust_last_name	Cust_address	Cust_email	Cust_Phone
▶	1	Rajiv	Malhotra	Kolkata,lane 5	rajjiv5@gmail.com	9185641234
	2	Rahul	Mahajan	Mumbai,lane 2	rahul@gmail.com	9845313558
	3	Mansi	Khanvilkar	Pune, plot 4	manu@gmail.com	9489651232
	4	Angela	Jones	delhi,lane 6	angie@gmail.com	9856445521
	5	Kajol	Gupta	Haryana,plot 1	kajol02@gmail.com	9479362456
*	NULL	NULL	NULL	NULL	NULL	NULL

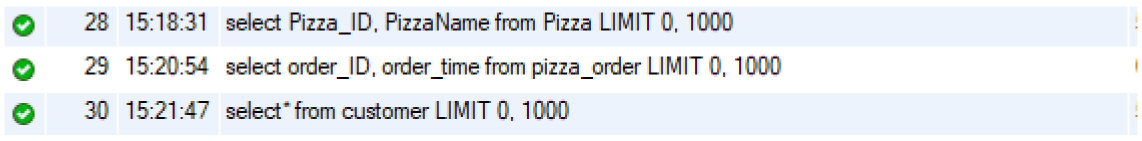
- Pizza
  select Pizza_ID, PizzaName from Pizza;



	Pizza_ID	PizzaName
▶	1	Chicken Golden Delight
	2	Non Veg Supreme
	3	Mexican Green Wave
	4	Peppy Paneer
	5	Veg Extravaganza
*	NULL	NULL

- Pizza order
  select order_ID, order_time from pizza_order;

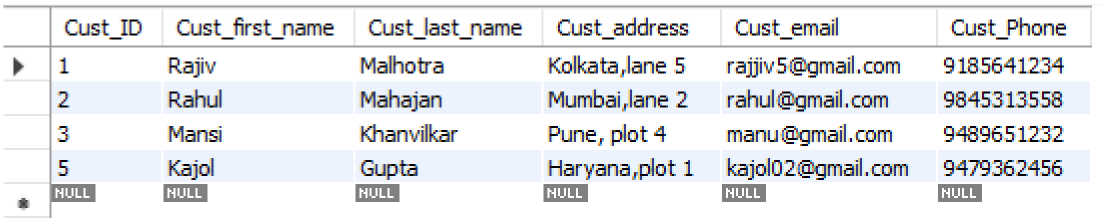
Selection of tables



✓	28	15:18:31	select Pizza_ID, PizzaName from Pizza LIMIT 0, 1000
✓	29	15:20:54	select order_ID, order_time from pizza_order LIMIT 0, 1000
✓	30	15:21:47	select* from customer LIMIT 0, 1000

C) Delete

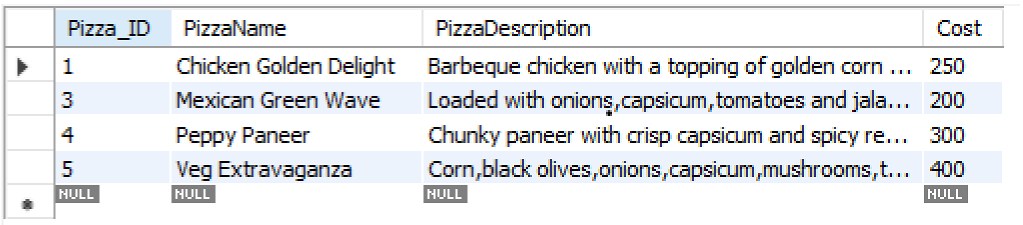
- Customer
  delete from customer where Cust_ID=4;



Cust_ID	Cust_first_name	Cust_last_name	Cust_address	Cust_email	Cust_Phone
1	Rajiv	Malhotra	Kolkata,lane 5	rajjiv5@gmail.com	9185641234
2	Rahul	Mahajan	Mumbai,lane 2	rahul@gmail.com	9845313558
3	Mansi	Khanvilkar	Pune, plot 4	manu@gmail.com	9489651232
5	Kajol	Gupta	Haryana,plot 1	kajol02@gmail.com	9479362456
NULL	NULL	NULL	NULL	NULL	NULL

  (Row with Cust_ID=4 is now deleted.)

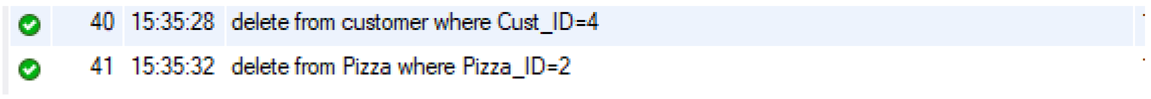
- Pizza
  delete from Pizza where Pizza_ID=2;



Pizza_ID	PizzaName	PizzaDescription	Cost
1	Chicken Golden Delight	Barbeque chicken with a topping of golden corn ...	250
3	Mexican Green Wave	Loaded with onions,capsicum,tomatoes and jala...	200
4	Peppy Paneer	Chunky paneer with crisp capsicum and spicy re...	300
5	Veg Extravaganza	Corn,black olives,onions,capsicum,mushrooms,t...	400
NULL	NULL	NULL	NULL

  (Row with Pizza_ID=2 is now deleted.)

Deletion of tables

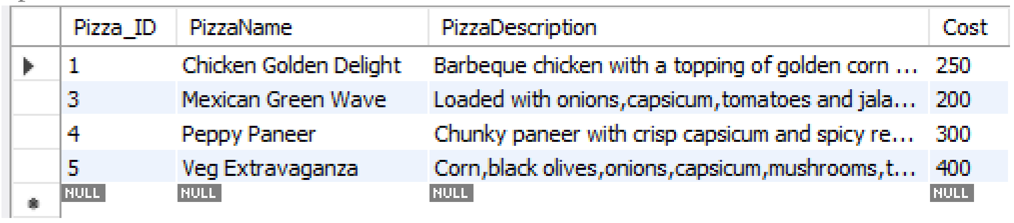


```
40  15:35:28  delete from customer where Cust_ID=4
41  15:35:32  delete from Pizza where Pizza_ID=2
```

D) Update

- Pizza

  update Pizza set Cost=300 where Pizza_ID=4;

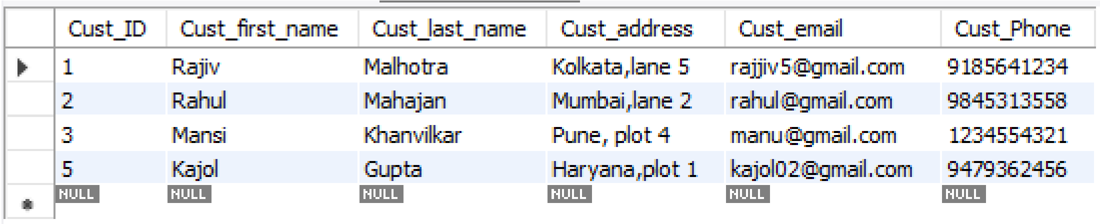


Pizza_ID	PizzaName	PizzaDescription	Cost
1	Chicken Golden Delight	Barbeque chicken with a topping of golden corn ...	250
3	Mexican Green Wave	Loaded with onions,capsicum,tomatoes and jala...	200
4	Peppy Paneer	Chunky paneer with crisp capsicum and spicy re...	300
5	Veg Extravaganza	Corn,black olives,onions,capsicum,mushrooms,t...	400
NULL	NULL	NULL	NULL

  (Row with Pizza_ID=4 has Cost updated to 300)

- Customer

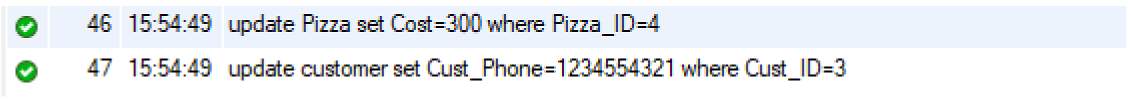
  update customer set Cust_Phone=1234554321 where Cust_ID=3;



Cust_ID	Cust_first_name	Cust_last_name	Cust_address	Cust_email	Cust_Phone
1	Rajiv	Malhotra	Kolkata,lane 5	rajjiv5@gmail.com	9185641234
2	Rahul	Mahajan	Mumbai,lane 2	rahul@gmail.com	9845313558
3	Mansi	Khanvilkar	Pune, plot 4	manu@gmail.com	1234554321
5	Kajol	Gupta	Haryana,plot 1	kajol02@gmail.com	9479362456
NULL	NULL	NULL	NULL	NULL	NULL

  (Row with Cust_ID=4 has Cust_Phone updated to 1234554321)

Updating the tables

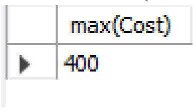


```
46  15:54:49  update Pizza set Cost=300 where Pizza_ID=4
47  15:54:49  update customer set Cust_Phone=1234554321 where Cust_ID=3
```

3. Aggregate Functions

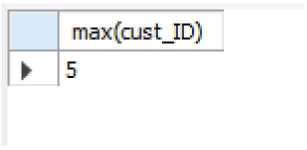
A) Max

- Pizza
  select max(Cost) from Pizza;



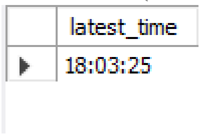
	max(Cost)
▶	400

- Order Delivery
  select max(cust_ID) from order_delivery;



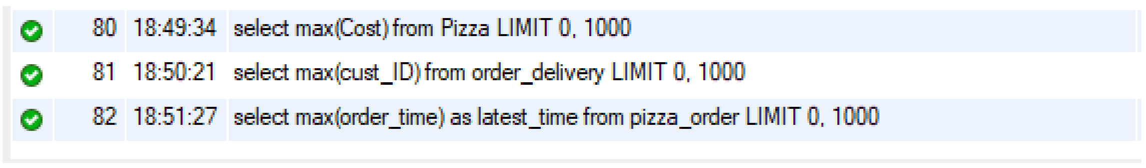
	max(cust_ID)
▶	5

- Pizza order
  select max(order_time) as latest_time from pizza_order;



	latest_time
▶	18:03:25

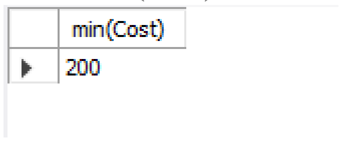
Getting Max value from tables



```
✓  80  18:49:34  select max(Cost) from Pizza LIMIT 0, 1000
✓  81  18:50:21  select max(cust_ID) from order_delivery LIMIT 0, 1000
✓  82  18:51:27  select max(order_time) as latest_time from pizza_order LIMIT 0, 1000
```

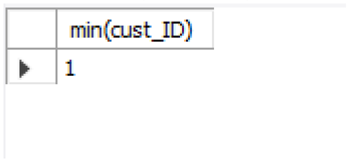
B) Min

- Pizza
  select min(Cost) from Pizza;



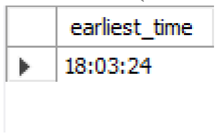
	min(Cost)
▶	200

- Order Delivery
  select min(cust_ID) from order_delivery;



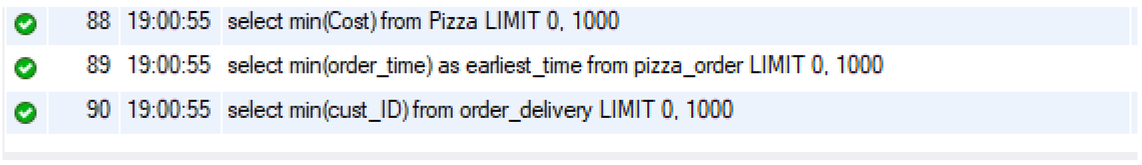
	min(cust_ID)
▶	1

- Pizza order
  select min(order_time) as latest_time from pizza_order;



	earliest_time
▶	18:03:24

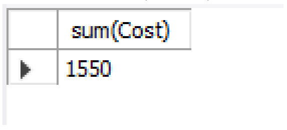
Getting Min value from tables



✓	88	19:00:55	select min(Cost) from Pizza LIMIT 0, 1000
✓	89	19:00:55	select min(order_time) as earliest_time from pizza_order LIMIT 0, 1000
✓	90	19:00:55	select min(cust_ID) from order_delivery LIMIT 0, 1000

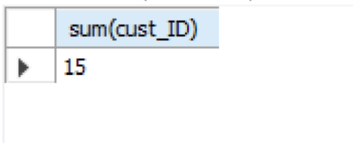
C) Sum

- Pizza
  select sum(Cost) from Pizza;



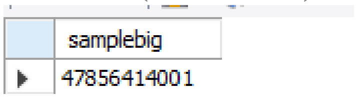
	sum(Cost)
▶	1550

- Order delivery
  select sum(cust_ID) from order_delivery;



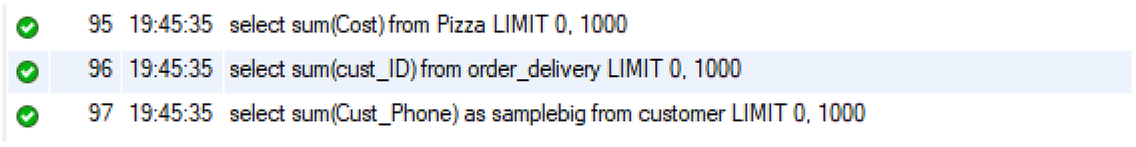
	sum(cust_ID)
▶	15

- Customer
  select sum(Cust_Phone) as samplebig from customer;



	samplebig
▶	47856414001

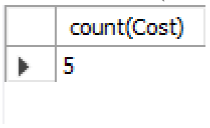
Getting Sum in tables



```
✓  95  19:45:35  select sum(Cost) from Pizza LIMIT 0, 1000
✓  96  19:45:35  select sum(cust_ID) from order_delivery LIMIT 0, 1000
✓  97  19:45:35  select sum(Cust_Phone) as samplebig from customer LIMIT 0, 1000
```

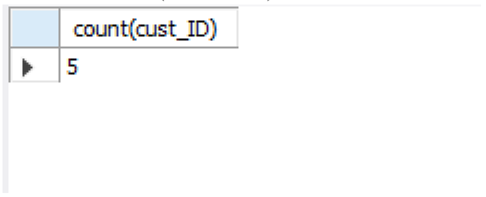
D) Count

- Pizza
  select count(Cost) from Pizza;



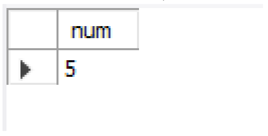
	count(Cost)
▶	5

- Order Delivery
  select count(cust_ID) from order_delivery;



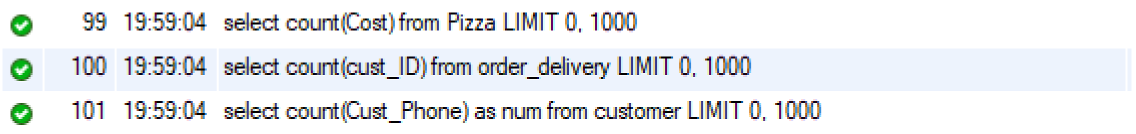
	count(cust_ID)
▶	5

- Customer
  select count(Cust_Phone) as num from customer;



	num
▶	5

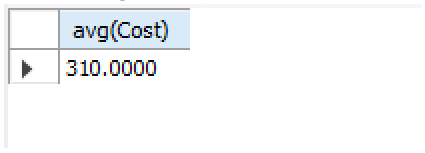
Getting Count from tables



```
✓  99   19:59:04  select count(Cost) from Pizza LIMIT 0, 1000
✓  100  19:59:04  select count(cust_ID) from order_delivery LIMIT 0, 1000
✓  101  19:59:04  select count(Cust_Phone) as num from customer LIMIT 0, 1000
```

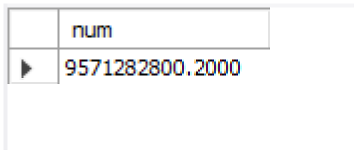
E) Average

- Pizza
  select avg(Cost) from Pizza;



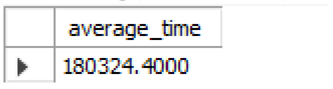
	avg(Cost)
▶	310.0000

- Customer
  select avg(Cust_Phone) as num from customer;



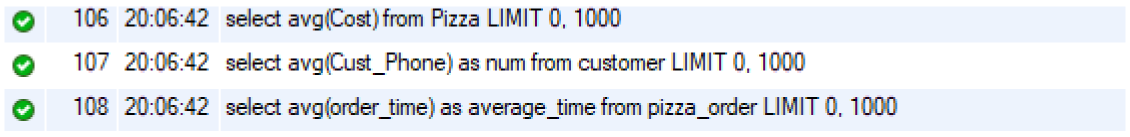
	num
▶	9571282800.2000

- Pizza Order
  select avg(order_time) as average_time from pizza_order;



	average_time
▶	180324.4000

Getting Average from tables



✓	106	20:06:42	select avg(Cost) from Pizza LIMIT 0, 1000
✓	107	20:06:42	select avg(Cust_Phone) as num from customer LIMIT 0, 1000
✓	108	20:06:42	select avg(order_time) as average_time from pizza_order LIMIT 0, 1000

4. Join:

A. INNER JOIN

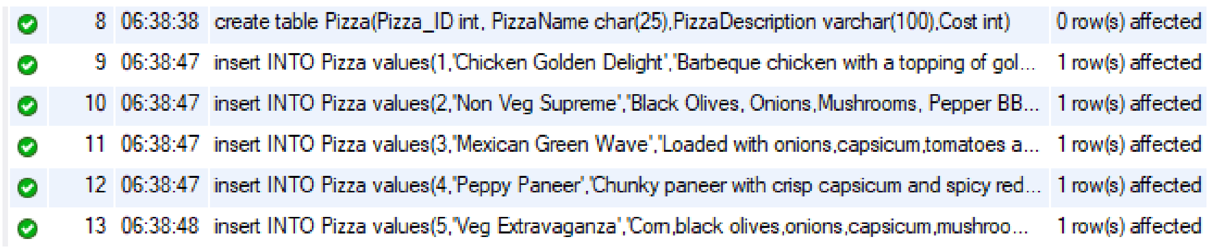
SELECT Pizza.PizzaName, Pizza.PizzaDescription, Toppings.ToppingName

FROM Pizza

INNER JOIN Toppings

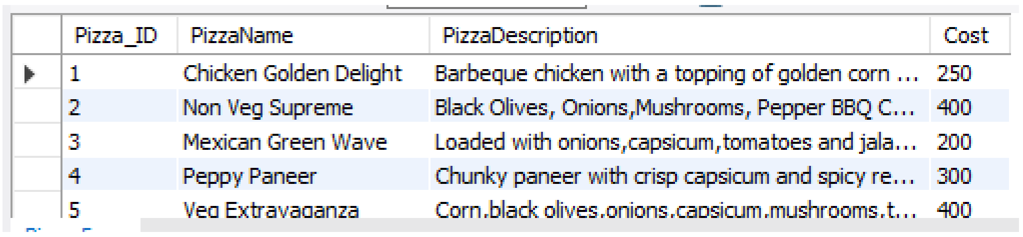
WHERE Pizza.Pizza_ID = Toppings.ToppingID;

Pizza Table:



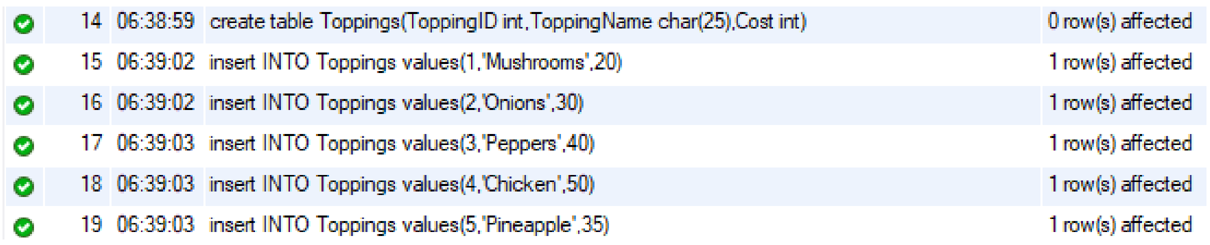
```
8   06:38:38  create table Pizza(Pizza_ID int, PizzaName char(25),PizzaDescription varchar(100),Cost int)   0 row(s) affected
9   06:38:47  insert INTO Pizza values(1,'Chicken Golden Delight','Barbeque chicken with a topping of gol...   1 row(s) affected
10  06:38:47  insert INTO Pizza values(2,'Non Veg Supreme','Black Olives, Onions,Mushrooms, Pepper BB...     1 row(s) affected
11  06:38:47  insert INTO Pizza values(3,'Mexican Green Wave','Loaded with onions,capsicum,tomatoes a...    1 row(s) affected
12  06:38:47  insert INTO Pizza values(4,'Peppy Paneer','Chunky paneer with crisp capsicum and spicy red...  1 row(s) affected
13  06:38:48  insert INTO Pizza values(5,'Veg Extravaganza','Corn,black olives,onions,capsicum,mushroo...   1 row(s) affected
```

Pizza table created:



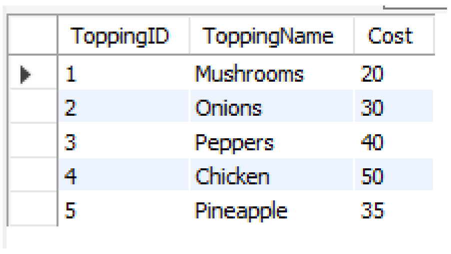
Pizza_ID	PizzaName	PizzaDescription	Cost
1	Chicken Golden Delight	Barbeque chicken with a topping of golden corn ...	250
2	Non Veg Supreme	Black Olives, Onions,Mushrooms, Pepper BBQ C...	400
3	Mexican Green Wave	Loaded with onions,capsicum,tomatoes and jala...	200
4	Peppy Paneer	Chunky paneer with crisp capsicum and spicy re...	300
5	Veg Extravaganza	Corn,black olives,onions,capsicum,mushrooms,t...	400

Toppings Table:



```
14  06:38:59  create table Toppings(ToppingID int,ToppingName char(25),Cost int)   0 row(s) affected
15  06:39:02  insert INTO Toppings values(1,'Mushrooms',20)   1 row(s) affected
16  06:39:02  insert INTO Toppings values(2,'Onions',30)      1 row(s) affected
17  06:39:03  insert INTO Toppings values(3,'Peppers',40)     1 row(s) affected
18  06:39:03  insert INTO Toppings values(4,'Chicken',50)     1 row(s) affected
19  06:39:03  insert INTO Toppings values(5,'Pineapple',35)   1 row(s) affected
```

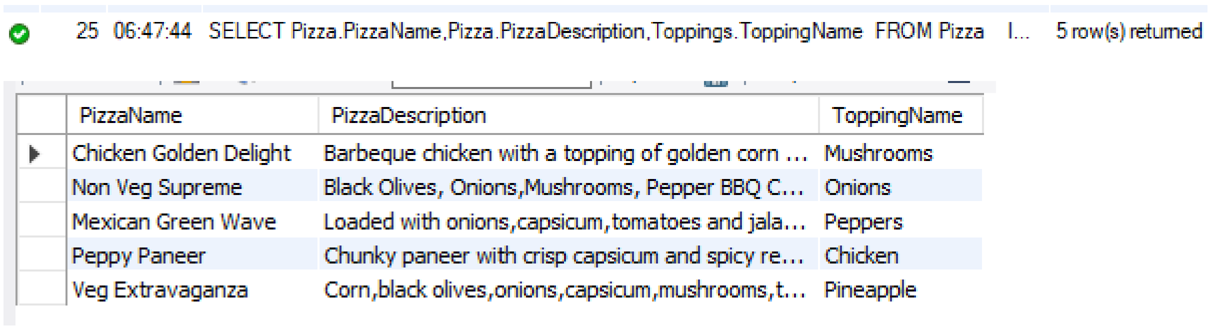
Toppings Table Created:



	ToppingID	ToppingName	Cost
▶	1	Mushrooms	20
	2	Onions	30
	3	Peppers	40
	4	Chicken	50
	5	Pineapple	35

After joining Pizza and Toppings table by using where clause which uses condition Pizza_ID = Toppings.ToppingID

This type of join is used by the Pizza Ordering system to display the recommended combinations of Pizza and toppings for the customers



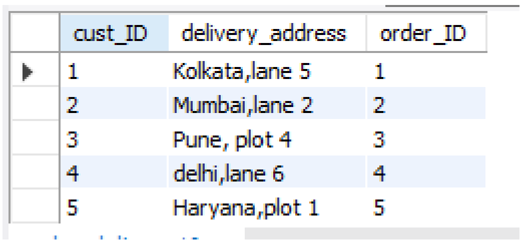
```
25  06:47:44  SELECT Pizza.PizzaName,Pizza.PizzaDescription,Toppings.ToppingName  FROM Pizza   I...   5 row(s) returned
```

	PizzaName	PizzaDescription	ToppingName
▶	Chicken Golden Delight	Barbeque chicken with a topping of golden corn ...	Mushrooms
	Non Veg Supreme	Black Olives, Onions,Mushrooms, Pepper BBQ C...	Onions
	Mexican Green Wave	Loaded with onions,capsicum,tomatoes and jala...	Peppers
	Peppy Paneer	Chunky paneer with crisp capsicum and spicy re...	Chicken
	Veg Extravaganza	Corn,black olives,onions,capsicum,mushrooms,t...	Pineapple

B. Left Join:

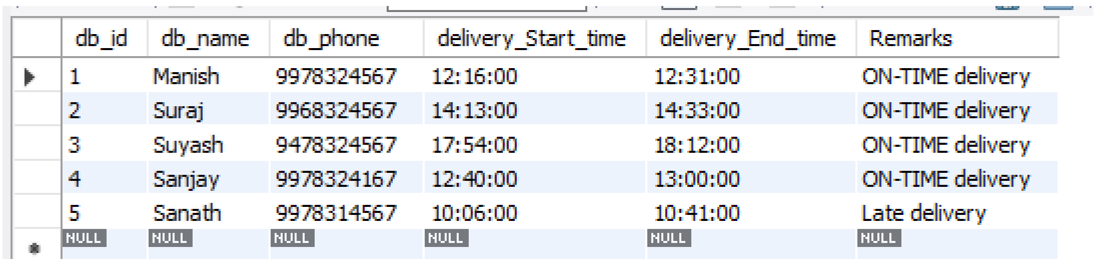
```
select
order_delivery.order_ID,order_delivery.delivery_address,delivery_boy.db_name,delivery_boy.db_phone

from order_delivery

left join delivery_boy

on Cust_ID=db_id;
```

Created a table order delivery:



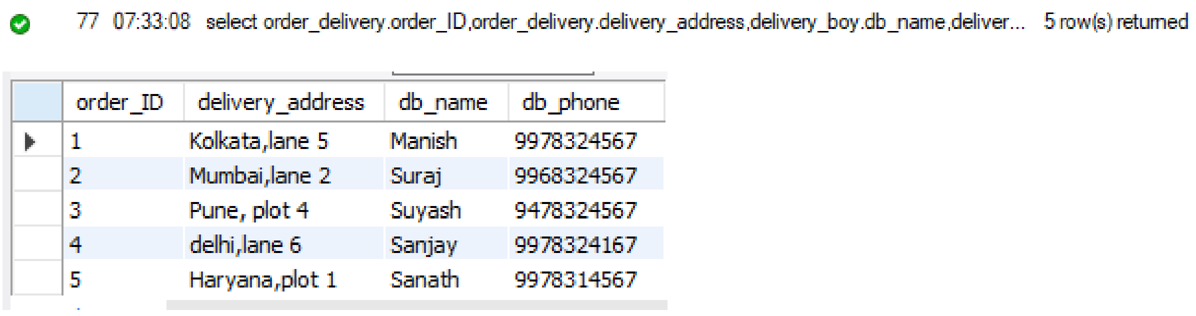
cust_ID	delivery_address	order_ID
1	Kolkata,lane 5	1
2	Mumbai,lane 2	2
3	Pune, plot 4	3
4	delhi,lane 6	4
5	Haryana,plot 1	5

Created a table delivery_boy:



db_id	db_name	db_phone	delivery_Start_time	delivery_End_time	Remarks
1	Manish	9978324567	12:16:00	12:31:00	ON-TIME delivery
2	Suraj	9968324567	14:13:00	14:33:00	ON-TIME delivery
3	Suyash	9478324567	17:54:00	18:12:00	ON-TIME delivery
4	Sanjay	9978324167	12:40:00	13:00:00	ON-TIME delivery
5	Sanath	9978314567	10:06:00	10:41:00	Late delivery
NULL	NULL	NULL	NULL	NULL	NULL

Joined columns cust_ID ,delivery_address from order_delivery with db_name and db_phone from delivery_boy table by using left join to show which delivery boy delivered order to which customer



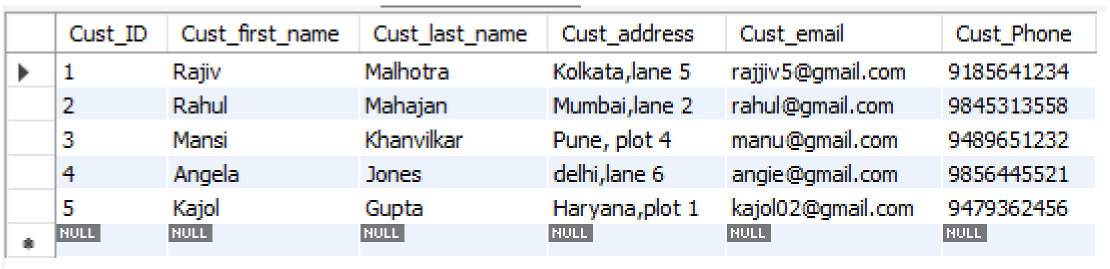
```
77  07:33:08  select order_delivery.order_ID,order_delivery.delivery_address,delivery_boy.db_name,deliver...  5 row(s) returned
```

order_ID	delivery_address	db_name	db_phone
1	Kolkata,lane 5	Manish	9978324567
2	Mumbai,lane 2	Suraj	9968324567
3	Pune, plot 4	Suyash	9478324567
4	delhi,lane 6	Sanjay	9978324167
5	Haryana,plot 1	Sanath	9978314567

C. Right Join:

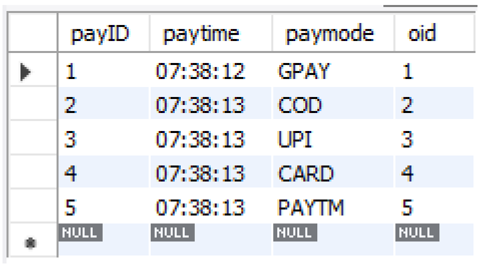
```
select customer.Cust_ID,customer.Cust_first_name,Payment.Paymode
from customer
right join Payment
on Cust_ID=PayID;
```

Created a table of customers who ordered pizza:



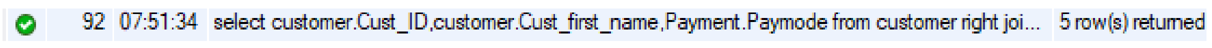
Cust_ID	Cust_first_name	Cust_last_name	Cust_address	Cust_email	Cust_Phone
1	Rajiv	Malhotra	Kolkata,lane 5	rajjiv5@gmail.com	9185641234
2	Rahul	Mahajan	Mumbai,lane 2	rahul@gmail.com	9845313558
3	Mansi	Khanvilkar	Pune, plot 4	manu@gmail.com	9489651232
4	Angela	Jones	delhi,lane 6	angie@gmail.com	9856445521
5	Kajol	Gupta	Haryana,plot 1	kajol02@gmail.com	9479362456
NULL	NULL	NULL	NULL	NULL	NULL

Created a table Payment which consists of payment details

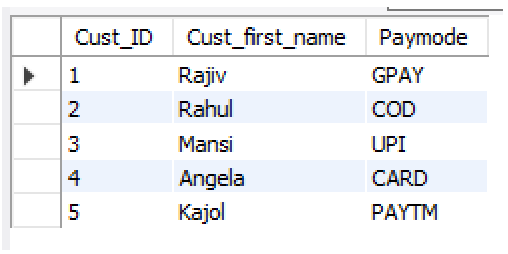


payID	paytime	paymode	oid
1	07:38:12	GPAY	1
2	07:38:13	COD	2
3	07:38:13	UPI	3
4	07:38:13	CARD	4
5	07:38:13	PAYTM	5
NULL	NULL	NULL	NULL

Now to show a in general payment history of the customer right join is used to join columns Cust_ID, Cust_first_name, from customer table and paymode from Payment table where the Cust_ID= payID



```
92  07:51:34  select customer.Cust_ID,customer.Cust_first_name,Payment.Paymode from customer right joi...  5 row(s) returned
```



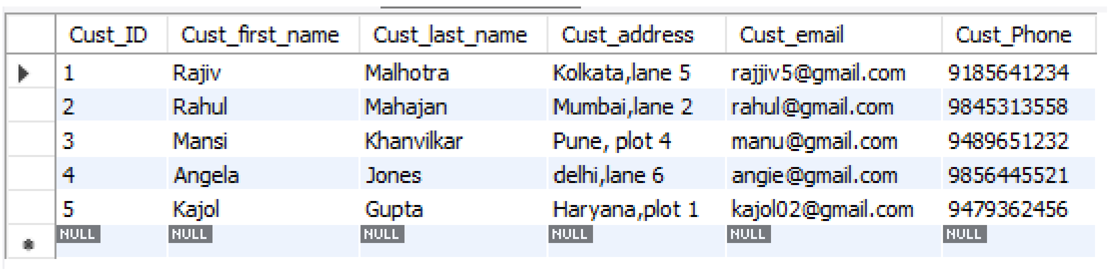
Cust_ID	Cust_first_name	Paymode
1	Rajiv	GPAY
2	Rahul	COD
3	Mansi	UPI
4	Angela	CARD
5	Kajol	PAYTM

5. View:

A .create view Customer_details as (select Cust_ID,concat(Cust_first_name,Cust_last_name), Cust_address,Cust_Phone from customer)

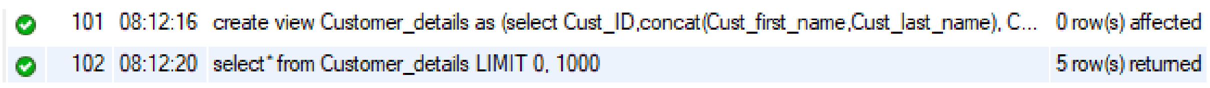
select* from Customer_details

Created a table of customers:

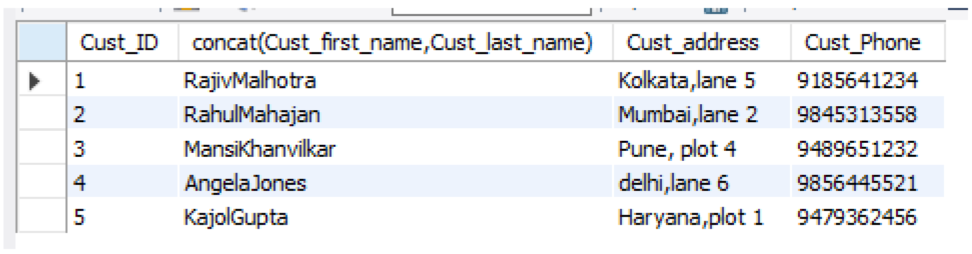


	Cust_ID	Cust_first_name	Cust_last_name	Cust_address	Cust_email	Cust_Phone
▶	1	Rajiv	Malhotra	Kolkata,lane 5	rajjiv5@gmail.com	9185641234
	2	Rahul	Mahajan	Mumbai,lane 2	rahul@gmail.com	9845313558
	3	Mansi	Khanvilkar	Pune, plot 4	manu@gmail.com	9489651232
	4	Angela	Jones	delhi,lane 6	angie@gmail.com	9856445521
	5	Kajol	Gupta	Haryana,plot 1	kajol02@gmail.com	9479362456
*	NULL	NULL	NULL	NULL	NULL	NULL

Now to display only specific details of customer to the delivery boy, a view of the table is created to give only Cust_ID,name, address and phone number of the customer.



✓	101	08:12:16	create view Customer_details as (select Cust_ID,concat(Cust_first_name,Cust_last_name), C...	0 row(s) affected
✓	102	08:12:20	select* from Customer_details LIMIT 0, 1000	5 row(s) returned

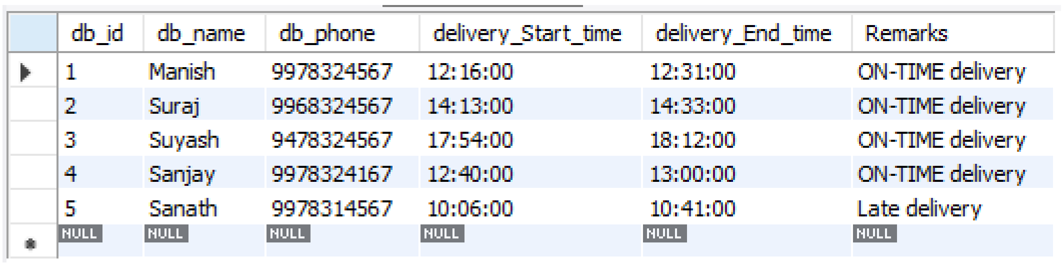


	Cust_ID	concat(Cust_first_name,Cust_last_name)	Cust_address	Cust_Phone
▶	1	RajivMalhotra	Kolkata,lane 5	9185641234
	2	RahulMahajan	Mumbai,lane 2	9845313558
	3	MansiKhanvilkar	Pune, plot 4	9489651232
	4	AngelaJones	delhi,lane 6	9856445521
	5	KajolGupta	Haryana,plot 1	9479362456

B. create view delivery_boy_details as (select db_name,db_phone from delivery_boy)

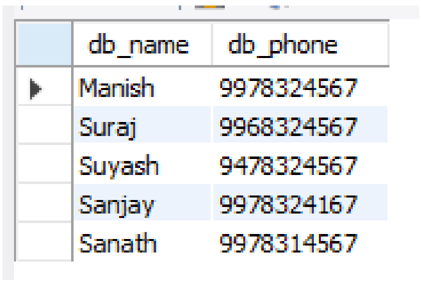
select * from delivery_boy_details

Created a table by name delivery_boy:



db_id	db_name	db_phone	delivery_Start_time	delivery_End_time	Remarks
1	Manish	9978324567	12:16:00	12:31:00	ON-TIME delivery
2	Suraj	9968324567	14:13:00	14:33:00	ON-TIME delivery
3	Suyash	9478324567	17:54:00	18:12:00	ON-TIME delivery
4	Sanjay	9978324167	12:40:00	13:00:00	ON-TIME delivery
5	Sanath	9978314567	10:06:00	10:41:00	Late delivery
NULL	NULL	NULL	NULL	NULL	NULL

Now to display only some specific details like delivery boy name, phone number of delivery boy to the customer a view of the table delivery boy is created



db_name	db_phone
Manish	9978324567
Suraj	9968324567
Suyash	9478324567
Sanjay	9978324167
Sanath	9978314567

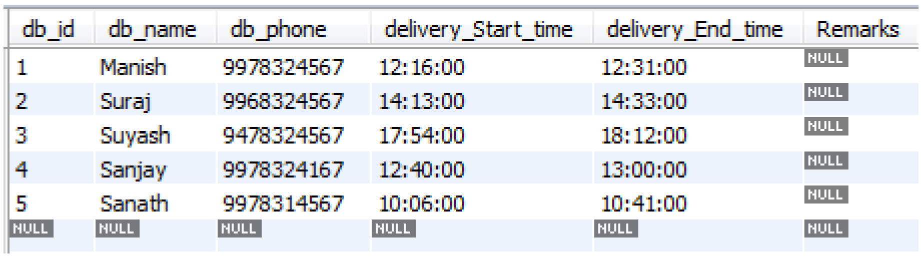
6. Triggers

A. Before Insert

```
delimiter $$
CREATE trigger  delivery BEFORE INSERT ON delivery_boy
       FOR EACH ROW
       BEGIN
           IF timediff(new.delivery_End_time,NEW.delivery_Start_time)<'00:30:00'
THEN SET NEW.Remarks = 'ON-TIME delivery';
           ELSE SET NEW.Remarks='Late delivery';
           END IF;
       END$$;
 delimiter ;
```

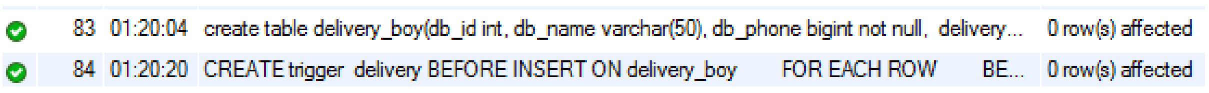
Before Insert is used on delivery boy table for checking if the order is delivered on time or late.

Delivery Boy table before applying trigger.



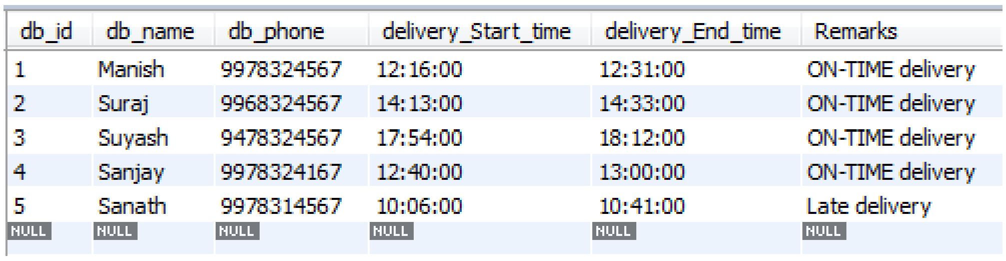
db_id	db_name	db_phone	delivery_Start_time	delivery_End_time	Remarks
1	Manish	9978324567	12:16:00	12:31:00	NULL
2	Suraj	9968324567	14:13:00	14:33:00	NULL
3	Suyash	9478324567	17:54:00	18:12:00	NULL
4	Sanjay	9978324167	12:40:00	13:00:00	NULL
5	Sanath	9978314567	10:06:00	10:41:00	NULL
NULL	NULL	NULL	NULL	NULL	NULL

Creation of Trigger



```
83  01:20:04  create table delivery_boy(db_id int, db_name varchar(50), db_phone bigint not null,  delivery...   0 row(s) affected
84  01:20:20  CREATE trigger  delivery BEFORE INSERT ON delivery_boy       FOR EACH ROW       BE...        0 row(s) affected
```

Delivery Boy table after applying the trigger



db_id	db_name	db_phone	delivery_Start_time	delivery_End_time	Remarks
1	Manish	9978324567	12:16:00	12:31:00	ON-TIME delivery
2	Suraj	9968324567	14:13:00	14:33:00	ON-TIME delivery
3	Suyash	9478324567	17:54:00	18:12:00	ON-TIME delivery
4	Sanjay	9978324167	12:40:00	13:00:00	ON-TIME delivery
5	Sanath	9978314567	10:06:00	10:41:00	Late delivery
NULL	NULL	NULL	NULL	NULL	NULL

B. After Insert

```
DELIMITER $$

CREATE TRIGGER trigger_n AFTER INSERT

ON customer FOR EACH ROW

BEGIN

        insert into order_delivery values(new.Cust_ID,
new.Cust_address,new.Cust_ID);

END$$

DELIMITER ;
```

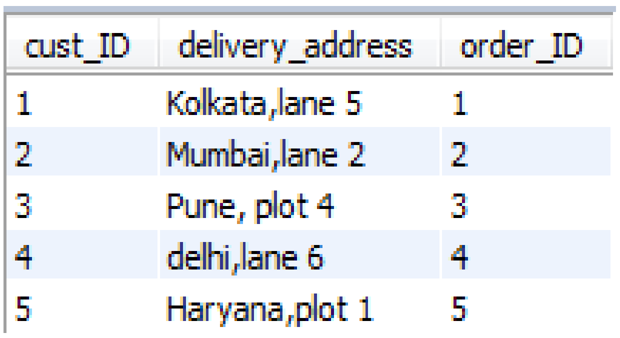
After insert is used on order delivery table to fetch the customer ID , delivery address and order ID which is going to be same as customer ID which can be joined with order table which has ordered pizza details.

<u>Order Delivery table</u>

```
create table order_delivery(cust_ID int, delivery_address varchar(50),order_ID int);
```

After order delivery table is created with trigger so as any data is entered in customer table so the trigger will help order delivery table to get updated automatically.

After trigger on order delivery table



cust_ID	delivery_address	order_ID
1	Kolkata,lane 5	1
2	Mumbai,lane 2	2
3	Pune, plot 4	3
4	delhi,lane 6	4
5	Haryana,plot 1	5

C. After Insert

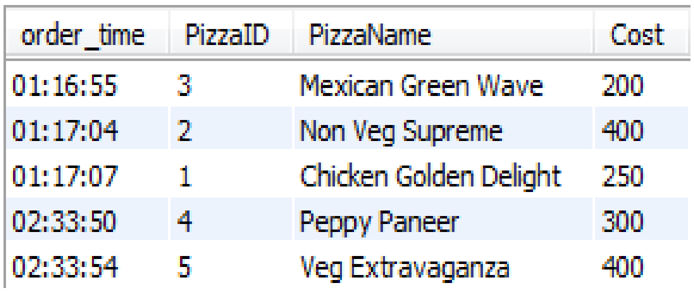
```
 DELIMITER $$
CREATE TRIGGER trigger_new AFTER INSERT
ON Pizza FOR EACH ROW
BEGIN
        insert into pizza_order (order_time, PizzaID, PizzaName, Cost)
        values(CURTIME(), new.Pizza_ID , new.PizzaName,new.Cost);
END$$
DELIMITER ;
```

After insert trigger is created on pizza order table so that whenever and pizza is inserted in pizza table it will update the pizza order table and when we join pizza order table and order delivery table created before we get to see the complete order table.

<u>Pizza Order table</u>

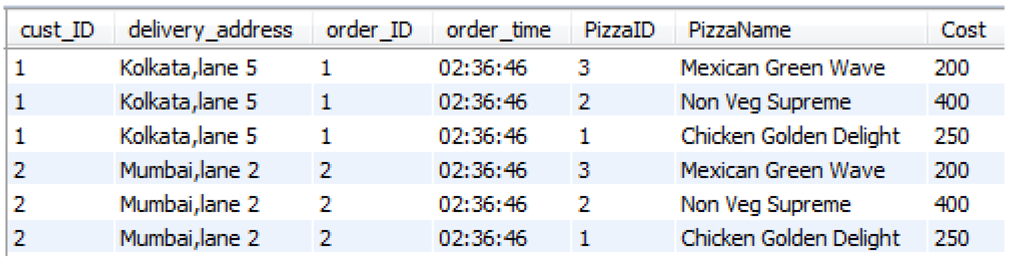
```
create table pizza_order(order_time time, PizzaID int, PizzaName char(25) ,Cost int);
```

After trigger on pizza order table



order_time	PizzaID	PizzaName	Cost
01:16:55	3	Mexican Green Wave	200
01:17:04	2	Non Veg Supreme	400
01:17:07	1	Chicken Golden Delight	250
02:33:50	4	Peppy Paneer	300
02:33:54	5	Veg Extravaganza	400

Final order table after joining order delivery table and pizza order table



cust_ID	delivery_address	order_ID	order_time	PizzaID	PizzaName	Cost
1	Kolkata,lane 5	1	02:36:46	3	Mexican Green Wave	200
1	Kolkata,lane 5	1	02:36:46	2	Non Veg Supreme	400
1	Kolkata,lane 5	1	02:36:46	1	Chicken Golden Delight	250
2	Mumbai,lane 2	2	02:36:46	3	Mexican Green Wave	200
2	Mumbai,lane 2	2	02:36:46	2	Non Veg Supreme	400
2	Mumbai,lane 2	2	02:36:46	1	Chicken Golden Delight	250

7. Normalization:

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

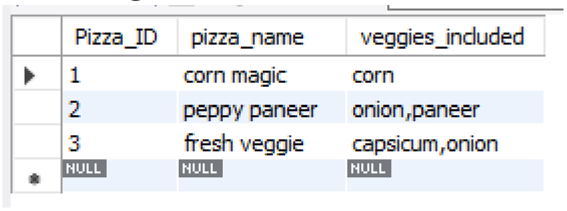
There are various types of normal froms

1.1NF
2.2NF
3.3NF

1. 1NF: (First Normal Form)

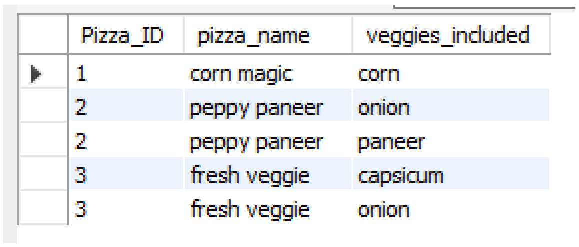
- Values must be single valued and atomic
- Values must be simple
- Each row must be unique

Following is a menu_card table



Pizza_ID	pizza_name	veggies_included
1	corn magic	corn
2	peppy paneer	onion,paneer
3	fresh veggie	capsicum,onion
NULL	NULL	NULL

Conversion of above table in 1NF is as follows



Pizza_ID	pizza_name	veggies_included
1	corn magic	corn
2	peppy paneer	onion
2	peppy paneer	paneer
3	fresh veggie	capsicum
3	fresh veggie	onion

2. 2NF: (Second Normal Form)

- Table must be in 1NF form
- Non-key attributes must be dependent on the primary-key values

3. 3NF: (Third Normal Form)

- Table must be in 2NF
- Transition dependency must not exist

Conclusion:

Thus we have successfully implemented all the concepts and applied on the project. In report all the all aspects are given clear explanation, hence the conclusion would just be a simple summary of our previous reports. In our first chapter we saw the DDL and DML commands and applied them and similarly further we saw other functions like aggregate, set, join, view, trigger and normalization.