Fundamental queries on Restaurant Management System

- Tables and Queries

```
create table customer(cust_id int NOT NULL AUTO_INCREMENT PRIMARY KEY ,Name varchar(100) ,Phone_No Dec(10) ,email varchar(100) ,address varchar(200));
Query OK, 0 rows affected (0.15 sec)

alter table customer add column password dec(6);
Query OK, 0 rows affected (0.39 sec)
Records: 0  Duplicates: 0  Warnings: 0

alter table customer add column username varchar(100);
Query OK, 0 rows affected (0.42 sec)
Records: 0  Duplicates: 0  Warnings: 0


mysql> alter table customer AUTO_INCREMENT=100;
Query OK, 0 rows affected (0.32 sec)
Records: 0  Duplicates: 0  Warnings: 0

alter table customer add column status varchar(100) default 'Preparing';
Query OK, 0 rows affected (0.32 sec)
Records: 0  Duplicates: 0  Warnings: 0


create table orders(cust_id int ,items varchar(200), price int , foreign key(cust_id) references customer(cust_id));
Query OK, 0 rows affected (0.47 sec)

alter table orders add column Date date default(CURRENT_DATE);
Query OK, 0 rows affected (0.28 sec)
Records: 0  Duplicates: 0  Warnings: 0

create table menu(menu_id int primary key,items varchar(100),I_price int);
Query OK, 0 rows affected (0.31 sec)


create table cart(item_name varchar(100),quantity int ,amount int);
Query OK, 0 rows affected (0.12 sec)


 alter table cart add column menu_id int ;
Query OK, 0 rows affected (0.41 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table cart add foreign key(menu_id) references menu(menu_id);
```

Query OK, 0 rows affected (0.63 sec)
Records: 0 Duplicates: 0 Warnings: 0

alter table cart add column cart_id int primary key;
Query OK, 0 rows affected (0.53 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create table Admins (id int PRIMARY KEY AUTO_INCREMENT, username VARCHAR(255) UNIQUE, password VARCHAR(255), email VARCHAR(255), phone VARCHAR(20));
Query OK, 0 rows affected (0.02 sec)

mysql> Alter table customer drop column status;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> Alter table orders add column status;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

Basic Queries:

0.Select * from Menu; // select * from table name
1.SELECT * FROM Customer LIMIT 5;
2.Select cust_id ,items,price Date,status from Orders limit 12;
3.select count(*) from menu;
4.SELECT * FROM menu WHERE menu_id= 101;
5.select min(i_price),max(i_price) from menu;
6.SELECT * FROM orders WHERE date > '2015-10-20';
7.update customer set name='bob', address ='london' where id=101;
8.delete from customer where id=101;
9.select * from orders where I_price<150 && I_price>100;
10. select * from Customer order by Name;
11.Select * from orders order by price desc;
12.Select cust_id,items,price ,date_format(Date,'%eth,%M,%Y') as Date from orders;
13.select * from Member where Name like '%B';
14.select * from customer where Name like '_%a';
15.select Customer.cust_id,Name, from BOOK right outer join Issue_Return on Issue_Return.BookId=BOOK.BookId and Fine!=0;
16.CREATE VIEW Title as select item_name, cart_id ,quantity,amount from cart;
17.CREATE VIEW Info as select cust_id,Name,email,address,Phone_No from Customer,orders where Customer.cust_id=Customer.cust_id ;
UPDATE Info set Name="Disha Patil" where Name ="Disha Patni";

18.Write a procedure to update the price of a item taking the Menu id and New Price as input Parameters.
mysql> delimiter //

```
create procedure u_cost(in M_id int, in new_price int)
BEGIN
update Menu set I_price =new_price where Menu_id=M_id;
END
//
Query OK, 0 rows affected (0.00 sec)
mysql> delimiter ;
mysql> call u_cost(101,900);
Query OK, 1 row affected (0.04 sec)
```

19.2. Write a function which accepts the customer id as input and returns the number of orders by him/ her.

```
mysql> create function func(C_id varchar(30))
-> returns int deterministic
-> begin
-> declare a int;
-> select Count(items) into a from orders where Cust_id=C_id ;
-> return a;
-> end //
Query OK, 0 rows affected (0.04 sec)

mysql> select func("101") //
+-------------+
| func("101") |
+-------------+
| 1 |
+-------------+
1 row in set (0.00 sec)
```

20.Write a trigger which will convert the Name to upper-case after inserting a new record in the customer table.

```
mysql> delimiter //
mysql> create trigger upper after insert on Customer
-> for each row
-> begin
-> set NEW.Name=UPPER(NEW.Name)
-> end //
```

Normalization :Table is not in Normalized form

```
+---------+---------------------------------------+---------+
| menu_id | items                                 | I_price |
+---------+---------------------------------------+---------+
|     101 | pani puri ,idli fry, mango juice ,chips |     350 |
|     102 | colddrink,chips                       |     200 |
|     103 | cold cofee                            |      40 |
|     104 | samosa,panipuri                       |     170 |
+---------+---------------------------------------+---------+
4 rows in set (0.00 sec)
```

1NF :First Normal form

- A table is referred to as being in its First Normal Form if atomicity of the table is 1.
- Here, atomicity states that a single cell cannot hold multiple values. It must hold only a single-valued attribute.
- The First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

After Nomallization

menu_id	Items_name	I_price	quanitity
101	pani puri	350	1
101	Idli fry	350	2
101	Mango juice	350	1
101	chips	350	1
102	colddrink	200	3
102	chips	200	2
103	coldcoffee	40	1
104	samosa	170	4
104	panipuri	170	1

2NF:Second Normal Form:Table is not in 2NF:Cart Table

menu_id	Items_name	I_price	Cart_id	quanitity
101	pani puri	350	501	1
101	Idli fry	350	501	2
101	Mango juice	350	501	1

101	chips	350	501	1
102	colddrink	200	502	3
102	chips	200	502	2
103	coldcoffee	40	503	1
104	samosa	170	504	4
104	panipuri	170	504	1

the table is not in **2NF**. We should separate them to a different table to make it 2NF.

It should be already in first normal form (1NF).

All non-key columns in the table are dependent on the entire primary key.

1st table :

menu_id	Cart_id
101	501
101	501
101	501
101	501
102	502
102	502
103	503
104	504
104	504

2nd Table:

Items_name	I_price	Cart_id	quanitity
pani puri	350	501	1
Idli fry	350	501	2
Mango juice	350	501	1
chips	350	501	1

colddrink	200	502	3
chips	200	502	2
coldcoffee	40	503	1
samosa	170	504	4
panipuri	170	504	1

To take this a step further, we should separate them again to a different table to make it 3NF.

3NF:Third Normal Form

It is already in second normal form (2NF).

All non-key columns in the table are dependent only on the primary key.

menu_id	Items_name	I_price
101	pani puri	350
101	Idli fry	350
101	Mango juice	350
101	chips	350
102	colddrink	200
102	chips	200
103	coldcoffee	40
104	samosa	170
104	panipuri	170

Cart_id	quanitity
501	1
501	2
501	1

501	1
502	3
502	2
503	1
504	4
504	1