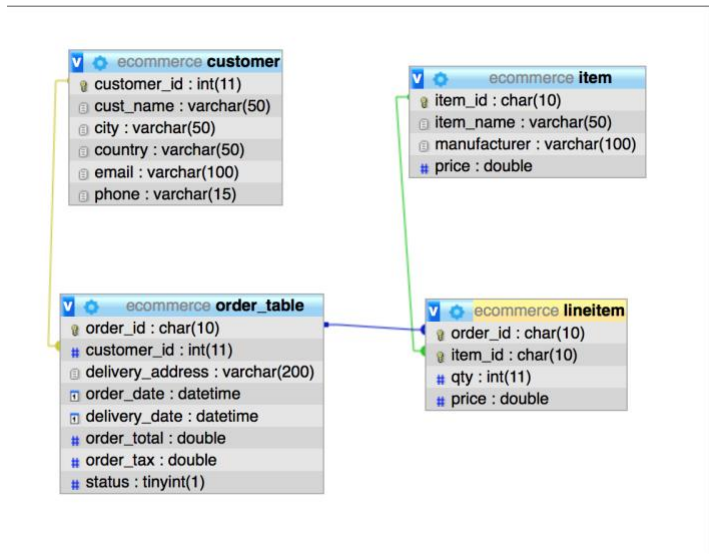


- 1) Create database ecommerce
 - a. Create database ecommerce;
- 2) Create tables with following structure :
 - a. Customer table
 - i. customer_id primary key
 - b. Item table
 - i. item_id primary key
 - c. Order_table
 - i. order_id primary key
 - ii. customer_id foreign key
 - iii. order_date should have default value of current timestamp
 - iv. order_table should be default 0
 - d. Create lineitem table
 - i. Order_id and item_id should be composite key



```

CREATE TABLE `customer` (
  `customer_id` int(11) NOT NULL PRIMARY KEY,
  `cust_name` varchar(50) DEFAULT NULL,
  `city` varchar(50) DEFAULT NULL,
  `country` varchar(50) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `phone` varchar(15) DEFAULT NULL
);

```

```

CREATE TABLE `item` (
  `item_id` char(10) NOT NULL PRIMARY KEY,
  `item_name` varchar(50) DEFAULT NULL,
  `manufacturer` varchar(100) DEFAULT NULL,
  `price` double DEFAULT NULL
);

```

```

CREATE TABLE `order_table` (
  `order_id` char(10) NOT NULL PRIMARY KEY,
  `customer_id` int(11) NOT NULL,
  `delivery_address` varchar(200) NOT NULL,
  `order_date` datetime DEFAULT CURRENT_TIMESTAMP,
  `delivery_date` datetime DEFAULT NULL,
  `order_total` double NOT NULL DEFAULT '0',

```

```

`order_tax` double DEFAULT NULL,
`status` tinyint(1) NOT NULL,
FOREIGN KEY (customer_id) REFERENCES customer(customer_id),

);

```

```

CREATE TABLE `lineitem` (
  `order_id` char(10) NOT NULL,
  `item_id` char(10) NOT NULL,
  `qty` int(11) NOT NULL,
  `price` double NOT NULL,
  PRIMARY KEY (order_id, item_id),
  FOREIGN KEY (order_id) REFERENCES order_table(order_id),
  FOREIGN KEY (item_id) REFERENCES item(item_id)
);

```

- 3) Alter table and add a column to store gst no of the customer
ALTER TABLE customer ADD COLUMN gst varchar(20)
- 4) Alter table customer and modify customer_id to be auto_increment
Alter table customer modify customer_id int AUTO_INCREMENT
- 5) Alter order_table to add status as a field with false as default value
ALTER TABLE order_table MODIFY STATUS tinyint DEFAULT 0
- 6) drop column order_total in order_table
ALTER table order_table drop order_total
ALTER table order_table add column order_total double AFTER order_date
- 7) Insert data in all the tables
INSERT INTO `customer` (`customer_id`, `cust_name`, `city`, `country`, `email`, `phone`) VALUES
 (1, 'Roy', 'Delhi', 'India', 'roya@test.com', '3636363636'),
 (2, 'Ronit', 'Delhi', 'India', 'ronit@techgatha.com', '1212121212'),
 (3, 'Sujoy', 'Pune', 'India', 'sujoy@test.com', '8787878787'),
 (4, 'Sam', 'Chicago', 'USA', 'sam@tehcl.com', '776765646465'),
 (5, 'Jack Ma', 'Singapore', 'Singapore', 'jack@test.com', '9898989898'),
 (6, 'Neil', 'San Francisco', 'USA', 'neil@techgatha.com', '9292929292'),
 (7, 'Prakash', 'London', 'UK', 'prakash@test.com', '9273645352'),
 (8, 'Ray', 'Mumbai', 'India', 'ray@gmail.com', '8585858585'),
 (9, 'Alexa', 'Mumbai', 'India', 'alexa@gmail.com', '9292929292'),
 (10, 'Chinki', 'Mumbai', 'China', 'chinki@gmail.com', '5454545454');

```

INSERT INTO `item` (`item_id`, `item_name`, `manufacturer`, `price`) VALUES
('I001', 'Personal Computer', 'Apple', 35000),
('I002', 'Personal Computer', 'Windows', 45000),
('I003', 'Bluetooth', 'Bose', 15000),
('I004', 'laptop', 'Apple', 55000),
('I005', 'laptop', 'HP', 25000),
('I006', 'Headphone', 'Sony', 20000),
('I007', 'Wireless Mouse', 'Sony', 0);

```

```

INSERT INTO `order_table` (`order_id`, `customer_id`, `delivery_address`, `order_date`, `delivery_date`, `order_tax`,
`status`) VALUES
('O001', 1, 'street name charcoal, plot no 60, Nagpur-898989', '2021-03-07 21:42:38', '2021-03-09 00:00:00', NULL, 0),
('O002', 1, 'street no 1234, Saunflower Apts, Delhi', '2021-03-07 21:53:50', '2021-03-14 19:30:00', NULL, 0),
('O003', 4, 'Islander Apartments, Chicago-898989', '2021-03-09 01:42:38', '2021-03-10 12:45:00', NULL, 0),
('O004', 4, 'Tousdale apts, NewYork, USA', '2021-04-09 21:42:38', '2021-05-10 09:00:00', NULL, 0),
('O005', 5, 'Indian Street, Singapore', '2021-03-12 05:10:10', '2021-03-15 05:00:00', NULL, 0),
('O006', 2, '93 neelkamal apartments, Mumbai-34333', '2021-03-08 12:12:38', '2021-03-10 09:00:00', NULL, 0);

```

```

INSERT INTO `lineitem` (`order_id`, `item_id`, `qty`, `price`) VALUES
('O001', 'I001', 1, 35000),
('O001', 'I006', 1, 20000),

```

```

('O002', 'I003', 1, 15000),
('O002', 'I005', 1, 25000),
('O002', 'I006', 1, 20000),
('O003', 'I004', 1, 55000),
('O004', 'I002', 1, 45000),
('O004', 'I003', 2, 15000),
('O005', 'I003', 3, 15000),
('O005', 'I005', 2, 25000),
('O006', 'I001', 1, 35000),
('O006', 'I004', 1, 55000);

```

- 8) Create duplicate line item table. Insert all records from line item table to duplicate line item table.
Create table dup_lineitem as select * from lineitem;
- 9) Delete one line item belonging to a particular order id from dup line item
- 10) Update price for line item I003 to 10% of the price where qty is more than 1
- 11) Truncate duplicate line item.
- 12) Drop duplicate line item
- 13) Get customer from country India and USA using IN operator
Select * from customer where country = 'India' and email like '%gmail%'
- 14) Write a query to get sum of price for each order in line item
 - a. select sum(qty*price), order_id from lineitem group by order_id
- 15) Get order_id and total of all the orders whose total is between 50000 and 70000

<u>total</u>	<u>order_id</u>	
55000	<u>O001</u>	
60000	<u>O002</u>	
55000	<u>O003</u>	

```

SELECT sum(price*qty) as total , order_id from lineitem GROUP by order_id HAVING sum(price*qty) BETWEEN 50000
and 70000

```

- 16) Get all orders placed in the month of April
select * from order_table where month(order_date) = 04
- 17) Write a query to get sum of price for each customer and display total orders placed, customer id and customer name

← T →				<u>Sum</u>	<u>customer_id</u>	<u>cust_name</u>
Edit	Copy	Delete		115000	1	Roy
Edit	Copy	Delete		90000	2	Ronit
Edit	Copy	Delete		130000	4	Sam
Edit	Copy	Delete		95000	5	Jack Ma

- a. select sum(qty*price) as Sum, customer.customer_id , customer.customer_name from lineitem,
order_table, customer
where order_table.order_id = lineitem.order_id
and order_table.customer_id=customer.customer_id
group by customer.customer_id

- 18) Write a query to get sum of price for customers having more than 100000 in orders.

sum(qty*price)	customer_id	cust_name
115000	1	Roy
130000	4	Sam

- b. `select sum(qty*price), customer.customer_id, customer.cust_name from lineitem, order_table, customer
where order_table.order_id = lineitem.order_id
and order_table.customer_id=customer.customer_id
group by customer.customer_id
HAVING sum(qty*price) >100000`

- 19) To get customer id, name and email of the customers who haven't placed order yet

cust_name	email	customer_id
Sujoy	sujoy@test.com	3
Neil	neil@techgatha.com	6
Prakash	prakash@test.com	7
Ray	ray@gmail.com	8
Alexa	alexa@gmail.com	9
Chinki	chinki@gmail.com	10

[USING SUB QuERIES]

`SELECT cust_name, email, customer_id from customer WHERE
customer_id NOT IN (SELECT customer_id from order_table)`

OR [using join]

`SELECT cust_name, email, c.customer_id from customer c left join order_table ot on c.customer_id = ot.customer_id
where ot.customer_id is null`

- 20) get all data from items and lineitem for the items for which the orders has been placed. The item id should not be repeated

item_id	item_name
I001	Personal Computer
I002	Personal Computer
I003	Bluetooth
I004	laptop
I005	laptop
I006	Headphone

`select DISTINCT i.item_id,i.item_name from item i inner join lineitem l on i.item_id=l.item_id`

- 21) Get the item id, item name and total quantity of each item sold.

item_name	item_id	sum(l.qty)
Personal Computer	I001	2
Personal Computer	I002	1
Bluetooth	I003	6
laptop	I004	2
laptop	I005	3
Headphone	I006	2

```
select DISTINCT i.item_id,i.item_name , sum(l.qty) from item i inner join lineitem l on i.item_id=l.item_id
group by i.item_id,i.item_name
```

- 22) Get customer name, item name, manufacturer, quantity of each item purchased by each customer for items placed between 25 mar and 30 apr [both dates inclusive] and order_date

<u>cust_name</u>	<u>item_name</u>	<u>manufacturer</u>	<u>qty</u>	<u>order_date</u>	
Sam	Personal Computer	Windows	1	2021-04-09 21:42:38	
Sam	Bluetooth	Bose	2	2021-04-09 21:42:38	

```
select c.cust_name, i.item_name, i.manufacturer , l.qty
from order_table ot, customer c, item i, lineitem l
where ot.order_date
between '2021-03-25' and '2021-04-30' and ot.order_id= l.order_id and ot.customer_id=c.customer_id and i.item_id
= l.item_id
```

- 23) Get item name, manufacturer and total items sold of each type

<u>item_name</u>	<u>manufacturer</u>	<u>Total Items Sold</u>	
Personal Computer	Apple	2	
Personal Computer	Windows	1	
Bluetooth	Bose	6	
laptop	Apple	2	
laptop	HP	3	
Headphone	Sony	2	

```
SELECT item_name, manufacturer, sum(qty) as 'Total Items Sold'
FROM item , lineitem
WHERE item.item_id = lineitem.item_id GROUP by lineitem.item_id
```

- 24) Display customer name, total orders placed by the customer and type of member on the basis of following conditions
 If Sale for customer is more than 100000 – Gold
 If Sale is between 90000 and 100000 – Silver
 Default is regular

<u>cust_name</u>	<u>Total Orders</u>	<u>typeofmember</u>	
Roy	115000	GOLD	
Ronit	90000	Regular	
Sam	130000	GOLD	
Jack Ma	95000	SILVER	

```

select cust_name,
sum(qty*price)as 'Total Orders',
CASE
    WHEN sum(qty*price)>100000 then 'GOLD'
    WHEN sum(qty*price)<100000 and sum(qty*price)>90000 then 'SILVER'
    Else 'Regular'
END typeofmember

from customer c, order_table o, lineitem l
WHERE c.customer_id = o.customer_id and o.order_id = l.order_id
group by o.customer_id

```

OR

```

SELECT cust_name, total ,
CASE
    WHEN total > 100000 THEN 'GOLD'
    WHEN total > 90000 THEN 'SILVER'
    ELSE 'REGULAR'
END Member_type
from customer c,
(select customer_id, sum(order_total) as total
from order_table o GROUP BY o.customer_id ) as temptable
WHERE temptable.customer_id = c.customer_id;

```

OR

```

SELECT cust_name, total ,
CASE
    WHEN total > 100000 THEN 'GOLD'
    WHEN total > 90000 THEN 'SILVER'
    ELSE 'REGULAR'
END Member_type
from customer c
join
(select customer_id, sum(order_total) as total
from order_table o GROUP BY o.customer_id ) as temptable
WHERE temptable.customer_id = c.customer_id;

```

```

SELECT cust_name, total ,
from customer c
join
(select customer_id, sum(order_total) as total
from order_table o GROUP BY o.customer_id ) as temptable
WHERE temptable.customer_id = c.customer_id;

```

OR

```

select cust_name,
sum(o.order_total) as 'Total Orders',
CASE
    WHEN 'Total Orders'>100000 then 'GOLD'
    WHEN 'Total Orders'<100000 and 'Total Orders'>90000 then 'SILVER'

```

```
        Else 'Regular'
        END typeofmember
```

```
from customer c, order_table o
WHERE c.customer_id = o.customer_id
group by o.customer_id
```

OR

```
select cust_name,
sum(o.order_total)as 'Total Orders',
CASE
    WHEN 'Total Orders'>100000 then 'GOLD'
    WHEN 'Total Orders'<100000 and 'Total Orders'>90000 then 'SILVER'
    Else 'Regular'
    END typeofmember
```

```
from customer c
join
order_table o
on c.customer_id = o.customer_id
group by o.customer_id
```

25) Get email of the customer whose order id = O002

```
SELECT email
FROM `customer`
WHERE customer_id = (
select customer_id from order_table where order_id='O002'
)
```

OR

```
select email from customer c, order_table o
where o.order_id='O002'
and o.customer_id = c.customer_id
```