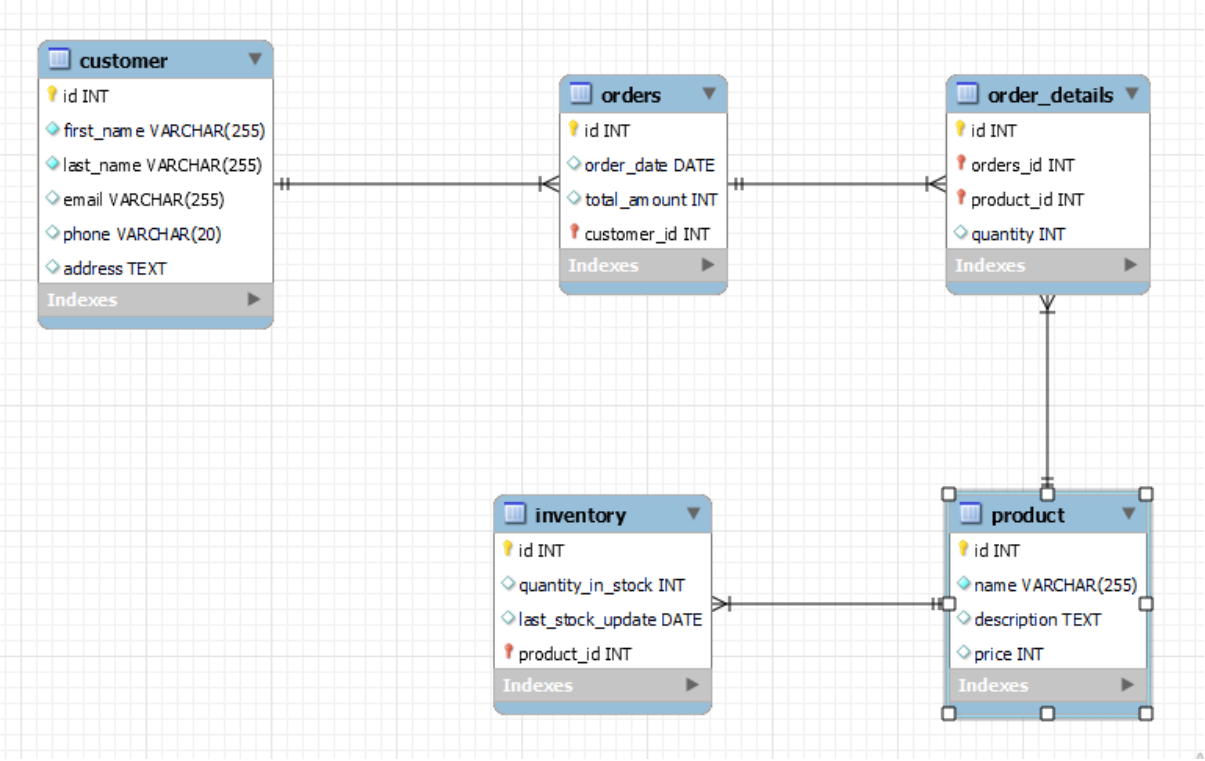
ASSIGNMENT NO : 1

TechShop, an electronic gadgets shop

ER DIAGRAM:



Task:1. Database Design:

```
-- MySQL Workbench Forward Engineering

-- -----------------------------------------------------

-- Schema TechShop

-- -----------------------------------------------------


-- -----------------------------------------------------

-- Schema TechShop

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `TechShop` DEFAULT CHARACTER SET utf8 ;

USE `TechShop` ;


-- -----------------------------------------------------

-- Table `TechShop`.`customer`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `TechShop`.`customer` (
```

```
  `id` INT NOT NULL AUTO_INCREMENT,
  `first_name` VARCHAR(255) NOT NULL,
  `last_name` VARCHAR(255) NOT NULL,
  `email` VARCHAR(255) NULL,
  `phone` VARCHAR(20) NULL,
  `address` TEXT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `TechShop`.`product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TechShop`.`product` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `description` TEXT NULL,
  `price` INT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `TechShop`.`orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TechShop`.`orders` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `order_date` DATE NULL,
  `total_amount` INT NULL,
  `customer_id` INT NOT NULL,
  PRIMARY KEY (`id`, `customer_id`),
  INDEX `fk_orders_customer_idx` (`customer_id` ASC) ,
  CONSTRAINT `fk_orders_customer`
```

```
    FOREIGN KEY (`customer_id`)
    REFERENCES `TechShop`.`customer` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `TechShop`.`inventory`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TechShop`.`inventory` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `quantity_in_stock` INT NULL,
  `last_stock_update` DATE NULL,
  `product_id` INT NOT NULL,
  PRIMARY KEY (`id`, `product_id`),
  INDEX `fk_inventory_product1_idx` (`product_id` ASC) ,
  CONSTRAINT `fk_inventory_product1`
    FOREIGN KEY (`product_id`)
    REFERENCES `TechShop`.`product` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `TechShop`.`order_details`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TechShop`.`order_details` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `orders_id` INT NOT NULL,
  `product_id` INT NOT NULL,
  `quantity` INT NULL,
  PRIMARY KEY (`id`, `orders_id`, `product_id`),
```

```
  INDEX `fk_product_has_orders_orders1_idx` (`orders_id` ASC) ,
  INDEX `fk_product_has_orders_product1_idx` (`product_id` ASC) ,
  CONSTRAINT `fk_product_has_orders_product1`
    FOREIGN KEY (`product_id`)
    REFERENCES `TechShop`.`product` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_product_has_orders_orders1`
    FOREIGN KEY (`orders_id`)
    REFERENCES `TechShop`.`orders` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

INSERTION:

```
-- customer insertion
insert into customer (first_name,last_name,email,phone,address)
values
('dhana','lakshmi','dhana@gmail.com','9360805403','ranipet'),
('dhivya','lakshmi','dhivya@gmail.com','9786205333','mudhaliarpet'),
('dhivya','bharathy','bharathy@gmail.com','9787333355','muthialpet'),
('bhavana','ranganathan','bhavana@gmail.com','8220681348','john paul'),
('pradheesha','sivakumar','pradheesha@gmail.com','8072607205','annanagar'),
('kaviya','lakshmanan','kaviya@gmail.com','9443500160','T nagar'),
('dhivya','prabha','prabha@gmail.com','8610248302','rainbow nagar'),
('dheepika','chellamuthu','dheepika@gmail.com','9787333394','anna nagar'),
('roshini','mani','roshini@gmail.com','9787333394','ranipet'),
('hema','kannan','hema@gmail.com','9787333364','muthialpet');
```

```
mysql> select * from customer;
+----+------------+------------+----------------------+------------+--------------+
| id | first_name | last_name  | email                | phone      | address      |
+----+------------+------------+----------------------+------------+--------------+
|  1 | dhana      | lakshmi    | dhana@gmail.com      | 9360805403 | ranipet      |
|  2 | dhivya     | lakshmi    | dhivya@gmail.com     | 9786205333 | mudhaliarpet |
|  3 | dhivya     | bharathy   | bharathy@gmail.com   | 9787333355 | muthialpet   |
|  4 | bhavana    | ranganathan| bhavana@gmail.com    | 8220681348 | john paul    |
|  5 | pradheesha | sivakumar  | pradheesha@gmail.com | 8072607205 | anna nagar   |
|  6 | kaviya     | lakshmanan | kaviya@gmail.com     | 9443500160 | T nagar      |
|  7 | dhivya     | prabha     | prabha@gmail.com     | 8610248302 | rainbow nagar|
|  8 | dheepika   | chellamuthu| dheepika@gmail.com   | 9787333394 | anna nagar   |
|  9 | roshini    | mani       | roshini@gmail.com    | 9787333394 | ranipet      |
| 10 | hema       | kannan     | hema@gmail.com       | 9787333364 | muthialpet   |
+----+------------+------------+----------------------+------------+--------------+
```

```
-- product insertion
INSERT INTO product (name, description, price) VALUES
('laptop', 'High-performance laptop with Intel Core i7', 36000),
('smartphone', 'Latest smartphone with 6.5-inch display', 27000),
('headphones', 'Wireless noise-canceling headphones', 1500),
('smartwatch', 'Fitness tracker and smartwatch combo', 7000),
('camera', 'Mirrorless camera with 24MP sensor', 10000),
('printer', 'Color inkjet printer with wireless capability', 3000),
('tablet', '10-inch tablet with high-resolution display', 5000),
('desktop', 'Powerful desktop computer for gaming', 1500),
('mouse', 'High-speed wireless router for home network', 80),
('monitor', '2TB external hard drive for backup', 1000);
```

```
mysql> select * from product;
+----+------------+-----------------------------------------------+-------+
| id | name       | description                                   | price |
+----+------------+-----------------------------------------------+-------+
|  1 | laptop     | High-performance laptop with Intel Core i7    | 36000 |
|  2 | smartphone | Latest smartphone with 6.5-inch display       | 27000 |
|  3 | headphones | Wireless noise-canceling headphones           |  1500 |
|  4 | smartwatch | Fitness tracker and smartwatch combo          |  7000 |
|  5 | camera     | Mirrorless camera with 24MP sensor            | 10000 |
|  6 | printer    | Color inkjet printer with wireless capability |  3000 |
|  7 | tablet     | 10-inch tablet with high-resolution display   |  5000 |
|  8 | desktop    | Powerful desktop computer for gaming          |  1500 |
|  9 | mouse      | High-speed wireless router for home network   |    80 |
| 10 | monitor    | 2TB external hard drive for backup            |  1000 |
+----+------------+-----------------------------------------------+-------+
```

```
-- order insertion
INSERT INTO orders (order_date,total_amount,customer_id) values
```

```
('2020/03/30',54000,1),
('2020/01/31',36000,2),
('2024/03/30',20000,3),
('2021/12/30',9000,4),
('2022/11/25',10000,5),
('2024/01/14',7500,6),
('2020/01/14',1000,7),
('2024/03/04',800,8),
('2023/02/28',54000,9),
('2021/03/30',36000,1),
('2023/03/30',14000,1),
('2023/10/15',81000,2),
('2020/08/25',3000,2),
('2021/08/25',7000,2),
('2022/08/25',10000,2),
('2023/08/25',10000,8),
('2020/03/30',15000,9),
('2020/03/30',400,9),
('2022/05/11',10000,3),
('2022/01/14',50000,3);
```

```
mysql> select * from orders;
+----+------------+--------------+-------------+
| id | order_date | total_amount | customer_id |
+----+------------+--------------+-------------+
|  1 | 2020-03-30 |        54000 |           1 |
|  2 | 2020-01-31 |        36000 |           2 |
|  3 | 2024-03-30 |        20000 |           3 |
|  4 | 2021-12-30 |         9000 |           4 |
|  5 | 2022-11-25 |        10000 |           5 |
|  6 | 2024-01-14 |         7500 |           6 |
|  7 | 2020-01-14 |         1000 |           7 |
|  8 | 2024-03-04 |          800 |           8 |
|  9 | 2023-02-28 |        54000 |           9 |
| 10 | 2021-03-30 |        36000 |           1 |
| 11 | 2023-03-30 |        14000 |           1 |
| 12 | 2023-10-15 |        81000 |           2 |
| 13 | 2020-08-25 |         3000 |           2 |
| 14 | 2021-08-25 |         7000 |           2 |
| 15 | 2022-08-25 |        10000 |           2 |
| 16 | 2023-08-25 |        10000 |           8 |
| 17 | 2020-03-30 |        15000 |           9 |
| 18 | 2020-03-30 |          400 |           9 |
| 19 | 2022-05-11 |        10000 |           3 |
| 20 | 2022-01-14 |        50000 |           3 |
+----+------------+--------------+-------------+
```

```
-- order detail insertion
insert into order_details(orders_id,product_id,quantity) values
(1,2,2),
(2,1,1),
(3,5,2),
(4,6,3),
(5,7,2),
(6,3,5),
(7,10,1),
(8,4,10),
(9,9,10),
(10,1,1),
(11,7,1),
(12,2,3),
(13,3,2),
(14,4,1),
(15,5,1),
(16,10,10),
(17,8,10),
(18,9,5),
(19,5,1),
(20,7,10);
```

```
mysql> select * from order_details;
+----+-----------+------------+----------+
| id | orders_id | product_id | quantity |
+----+-----------+------------+----------+
|  1 |         1 |          2 |        2 |
|  2 |         2 |          1 |        1 |
|  3 |         3 |          5 |        2 |
|  4 |         4 |          6 |        3 |
|  5 |         5 |          7 |        2 |
|  6 |         6 |          3 |        5 |
|  7 |         7 |         10 |        1 |
|  8 |         8 |          4 |       10 |
|  9 |         9 |          9 |       10 |
| 10 |        10 |          1 |        1 |
| 11 |        11 |          7 |        1 |
| 12 |        12 |          2 |        3 |
| 13 |        13 |          3 |        2 |
| 14 |        14 |          4 |        1 |
| 15 |        15 |          5 |        1 |
| 16 |        16 |         10 |       10 |
| 17 |        17 |          8 |       10 |
| 18 |        18 |          9 |        5 |
| 19 |        19 |          5 |        1 |
| 20 |        20 |          7 |       10 |
+----+-----------+------------+----------+
```

```
-- inventory insertion

insert into inventory(quantity_in_stock,last_stock_update,product_id)
values

(5,'2020/01/01',1),

(10,'2020/02/01',2),

(10,'2020/05/11',3),

(20,'2020/12/31',4),

(5,'2022/01/01',5),

(5,'2021/08/25',6),

(25,'2021/12/31',7),

(15,'2020/02/28',8),

(30,'2020/01/01',9),

(20,'2020/01/01',10);
```

```
mysql> select * from inventory;
+----+-------------------+-------------------+------------+
| id | quantity_in_stock | last_stock_update | product_id |
+----+-------------------+-------------------+------------+
|  1 |                 5 | 2020-01-01        |          1 |
|  2 |                10 | 2020-02-01        |          2 |
|  3 |                10 | 2020-05-11        |          3 |
|  4 |                20 | 2020-12-31        |          4 |
|  5 |                 5 | 2022-01-01        |          5 |
|  6 |                 5 | 2021-08-25        |          6 |
|  7 |                25 | 2021-12-31        |          7 |
|  8 |                15 | 2020-02-28        |          8 |
|  9 |                30 | 2020-01-01        |          9 |
| 10 |                20 | 2020-01-01        |         10 |
+----+-------------------+-------------------+------------+
```

-- Tasks 2: Select, Where, Between, AND, LIKE:

```
-- 1. Write an SQL query to retrieve the names and emails of all customers.

select concat(first_name," ",last_name) as name,email from customer;

-- 2. Write an SQL query to list all orders with their order dates and
corresponding customer names.
```

```
select o.id,concat(c.first_name," ",c.last_name) as customer_name,
o.order_date

from customer c,orders o

where c.id=o.customer_id;


/* 3. Write an SQL query to insert a new customer record into the
"Customers" table. Include customer information such as name, email, and
address.*/


insert into customer (first_name,last_name,email,phone,address)

values

('vishnu','priya','vishnu@gmail.com','9360805402','adayar');


/* 4. Write an SQL query to update the prices of all electronic gadgets in
the "Products" table by increasing them by 10%.*/


update product

set price=price+(0.1*price);


/* 5. Write an SQL query to delete a specific order and its associated
order details from the "Orders" and "OrderDetails" tables. Allow users to
input the order ID as a parameter.*/


delete o,s

from order_details o,orders s

where s.id=o.orders_id and s.id=13;


/* 6. Write an SQL query to insert a new order into the "Orders" table.
Include the customer ID, order date, and any other necessary information.*/


INSERT INTO orders (order_date,total_amount,customer_id) values

('2024/02/29',7000,10);


/* 7. Write an SQL query to update the contact information (e.g., email and
address) of a specific customer in the "Customers" table. Allow users to
input the customer ID and new contact information. */


update customer
```

```
set email='lakshmi@gmail.com',address='anna nagar' where id=1;


/*8. Write an SQL query to recalculate and update the total cost of each
order in the "Orders" table based on the prices and quantities in the
"OrderDetails" table.*/


update orders o

set total_amount=(select p.price*od.quantity from

product p join order_details od on p.id=od.product_id

where o.id=od.orders_id);


/* 9. Write an SQL query to delete all orders and their associated order
details for a specific customer from the "Orders" and "OrderDetails"
tables. Allow users to input the customer ID as a parameter. */


alter table order_details

add constraint fk_deletion

foreign key (orders_id)

references orders(id)

on delete cascade;


delete from orders where id=14;


/* 10. Write an SQL query to insert a new electronic gadget product into
the "Products" table, including product name, category, price, and any
other relevant details.*/


INSERT INTO product (name, description, price) VALUES

('mac book', 'High-performance laptop with mac os', 150000);


alter table product

add category varchar(255);


update product

set category=case

      when id=1 then 'gadget'

    when id=2 then 'gadget'
```

```
    when id=3 then 'i/o device'
    when id=4 then 'gadget'
    when id=5 then 'gadget'
    when id=6 then 'i/o device'
    when id=7 then 'gadget'
    when id=8 then 'i/o device'
    when id=9 then 'i/o device'
    when id=10 then 'i/o device'
    else 'gadget' end;
```

```
/* 11. Write an SQL query to update the status of a specific order in the
"Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input
the order ID and the new status.*/
```

```
alter table orders
add status varchar(255);
```

```
update orders
set status= case when id%2=0 then 'shipped' else 'pending' end;
```

```
update orders
set status='shipped' where id=1;
```

```
/* 12. Write an SQL query to calculate and update the number of orders
placed by each customer in the "Customers" table based on the data in the
"Orders" table.*/
```

```
alter table customer
add number_of_orders int;
```

```
update customer c
set number_of_orders=(select count(*)
                      from orders o
                      where c.id=o.customer_id);
```

-- Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

```
/* 1. Write an SQL query to retrieve a list of all orders along with
customer information (e.g.,customer name) for each order.*/


select concat(c.first_name," ",c.last_name) as
name,c.phone,c.email,o.order_date,o.total_amount,o.status

from customer c join orders o on

c.id=o.customer_id;


/* 2. Write an SQL query to find the total revenue generated by each
electronic gadget product. Include the product name and the total
revenue.*/


select p.name, sum(o.total_amount) as revenue

from product p join order_details od on p.id=od.product_id

join orders o on o.id=od.orders_id

group by p.id;


/* 3. Write an SQL query to list all customers who have made at least one
purchase. Include their names and contact information.*/


select concat(c.first_name," ",c.last_name),c.phone,c.email

from customer c join orders o on c.id=o.customer_id

group by c.id

having count(c.id)>=1;


/* 4. Write an SQL query to find the most popular electronic gadget, which
is the one with the highest total quantity ordered. Include the product
name and the total quantity ordered.*/


select p.name,sum(od.quantity) as popular_gadget

from product p join order_details od

on p.id=od.product_id

group by p.id

order by popular_gadget desc limit 0,1;


/* 5. Write an SQL query to retrieve a list of electronic gadgets along
with their corresponding categories.*/
```

```
select category,group_concat(name) as devices from product
group by category;
```

```
/* 6. Write an SQL query to calculate the average order value for each
customer. Include the customer's name and their average order value.*/
```

```
select c.first_name,avg(o.total_amount)
from customer c join orders o on c.id=o.customer_id
group by c.id;
```

```
/* 7. Write an SQL query to find the order with the highest total revenue.
Include the order ID, customer information, and the total revenue.*/
```

```
select o.id,c.*,o.total_amount
from customer c join orders o on c.id=o.customer_id
having max(o.total_amount);
```

```
/* 8. Write an SQL query to list electronic gadgets and the number of times
each product has been ordered.*/
```

```
select p.name,count(p.id) as number_of_times_ordered
from product p join order_details od  on p.id=od.product_id
group by p.id;
```

```
/* 9. Write an SQL query to find customers who have purchased a specific
electronic gadget product. Allow users to input the product name as a
parameter.*/
```

```
select p.name,group_concat(concat(c.first_name," ",c.last_name)) as
customers
from customer c join orders o on c.id=o.customer_id
join order_details od on o.id=od.orders_id
join product p on p.id=od.product_id
group by p.id;
```

```
/* 10. Write an SQL query to calculate the total revenue generated by all
orders placed within a specific time period. Allow users to input the start
and end dates as parameters.*/
```

```
select sum(total_amount) as total_revenue from orders
where order_date between '2024-01-01' and '2024-12-31';
```

-- Task 4. Subquery and its type:

```
-- 1. Write an SQL query to find out which customers have not placed any
orders.

select concat(first_name," ",last_name) as customer
from customer where id not in(select customer_id from orders);

-- 2. Write an SQL query to find the total number of products available for
sale.

select i.product_id,(i.quantity_in_stock- (select sum(od.quantity)
from order_details od
where od.product_id=i.product_id)) as number_of_products_available_for_sale
from inventory i ;

-- 3. Write an SQL query to calculate the total revenue generated by
TechShop.

select sum(total_amount) as toatal_revenue
from (select total_amount from orders) as revenue_by_techshop;

-- 4. Write an SQL query to calculate the average quantity ordered for
products in a specific category.

select p.category,(select avg(quantity) from order_details od
                              where od.id in (select id from product
                                    where category=p.category)
as average_quantity_category
from product p group by p.category;

/* 5. Write an SQL query to calculate the total revenue generated by a
specific customer. Allow users to input the customer ID as a parameter.*/
```

```
select concat(c.first_name," ",c.last_name) as name ,
(select sum(o.total_amount) from orders o
where o.customer_id=c.id
group by o.customer_id)as total_revenue
 from customer c;
```

/* 6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.*/

```
select concat(c.first_name," ",c.last_name) as name ,
(select count(o.customer_id) from orders o
where o.customer_id=c.id
group by o.customer_id)as order_count
 from customer c
 order by order_count desc;
```

/* 7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.*/

```
select p.name , (select sum(od.quantity)
                        from order_details od
                where p.id=od.product_id
                group by od.product_id) as popular_product ,p.category
from product p
order by popular_product desc limit 0,1;
```

/* 8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.*/

```
select concat(c.first_name," ",c.last_name) as most_money_spender,
(select sum(o.total_amount) from orders o
where c.id=o.customer_id
group by o.customer_id) as money_spent
from customer c
```

```
order by money_spent desc limit 0,1;
```

```
/* 9. Write an SQL query to calculate the average order value (total
revenue divided by the number of orders) for all customers.*/
```

```
select concat(c.first_name," ",c.last_name) as name,
(select avg(o.total_amount) from orders o
where o.customer_id=c.id
group by o.customer_id)
as average_order_value from customer c
order by average_order_value desc;
```

```
/* 10. Write an SQL query to find the total number of orders placed by each
customer and list their names along with the order count.*/
```

```
select concat(c.first_name," ",c.last_name) as name,
(select count(o.customer_id) from orders o
where o.customer_id=c.id
group by o.customer_id)
as total_number_of_orders from customer c
order by total_number_of_orders desc;
```