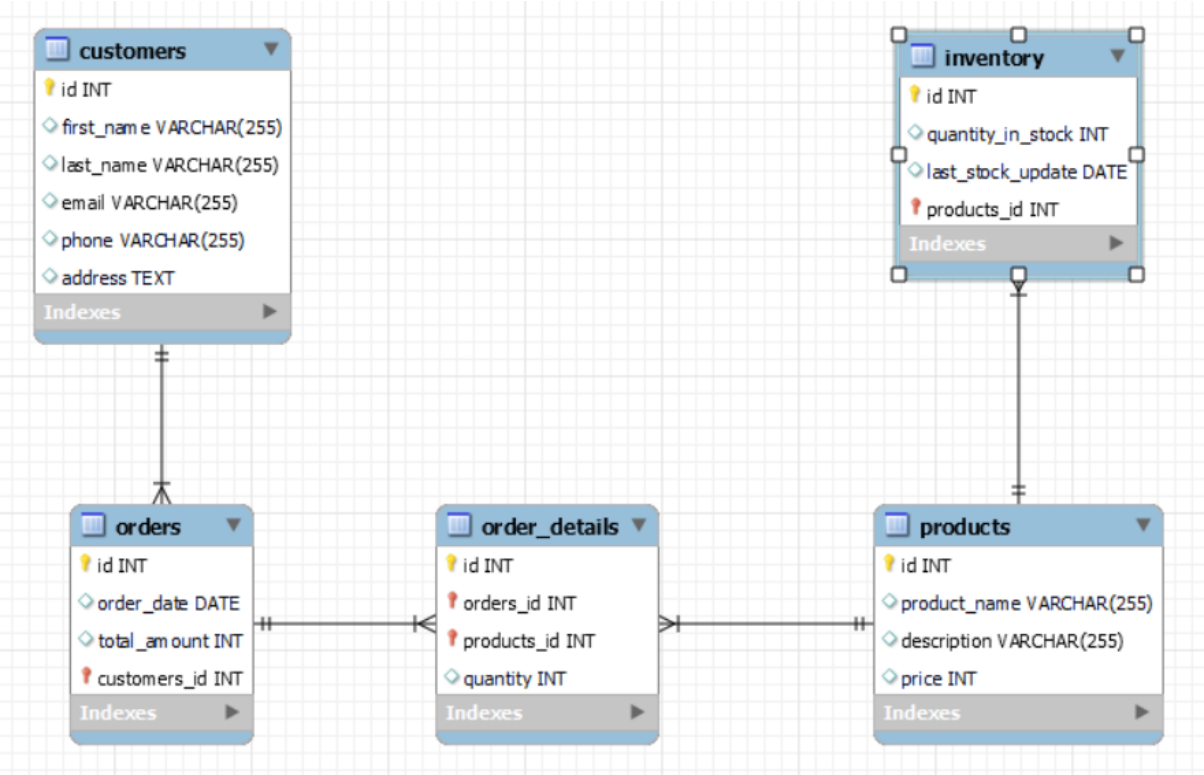
ASSIGNMENT NO : 1

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`.`customers`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TechShop`.`customers` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `first_name` VARCHAR(255) NULL,
  `last_name` VARCHAR(255) NULL,
  `email` VARCHAR(255) NULL,
  `phone` VARCHAR(255) NULL,
  `address` TEXT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `TechShop`.`products`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TechShop`.`products` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `product_name` VARCHAR(255) NULL,
  `description` VARCHAR(255) 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,
  `customers_id` INT NOT NULL,
  PRIMARY KEY (`id`, `customers_id`),
  INDEX `fk_orders_customers_idx` (`customers_id` ASC) ,
  CONSTRAINT `fk_orders_customers`
    FOREIGN KEY (`customers_id`)
    REFERENCES `TechShop`.`customers` (`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,
  `products_id` INT NOT NULL,
  PRIMARY KEY (`id`, `products_id`),
  INDEX `fk_inventory_products1_idx` (`products_id` ASC) ,
  CONSTRAINT `fk_inventory_products1`
    FOREIGN KEY (`products_id`)
    REFERENCES `TechShop`.`products` (`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,
  `products_id` INT NOT NULL,
  `quantity` INT NULL,
  PRIMARY KEY (`id`, `orders_id`, `products_id`),
  INDEX `fk_orders_has_products_products1_idx` (`products_id` ASC) ,
  INDEX `fk_orders_has_products_orders1_idx` (`orders_id` ASC) ,
  CONSTRAINT `fk_orders_has_products_orders1`
    FOREIGN KEY (`orders_id`)
    REFERENCES `TechShop`.`orders` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_orders_has_products_products1`
    FOREIGN KEY (`products_id`)
    REFERENCES `TechShop`.`products` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

INSERTION:

-- customer table

insert into customers (first_name, last_name, email, phone, address) values

('Ram','Prasad','ram@gmail.com','9024554745','123 main street'),

('Sandiya','Vishwanath','sandiya@gmail.com','9174543526','543 anna nagar'),

('Jayanthi','Selvam','selvam@gmail.com','9082707895','321 Ranipet'),

('Swetha','Seetharaman','swetha@gmail.com','7098645321','456 White Town'),

('Divya','Dharshini','divya@gmail.com','9123765480','890 Madagadipet'),

('Nisha','Vaithiyanathan','nisha@gmail.com','9865432178','698 Sowkarpet'),

('Darshini','Balamurali','darshnini@gmail.com','709834521','987 nehru nagar'),

('Agalya','Shanmugam','agalya@gmail.com','8143256790','678 Gandhi park'),

('Harini','Murugavel','harini@gmail.com','9024554745','234 Semmandalam'),

('Selva','Ramaiah','selva@gmail.com','9156473420','567 Manjakupam');

```
mysql> select*from customers;
+----+------------+----------------+------------------------+------------+-----------------+
| id | first_name | last_name      | email                  | phone      | address         |
+----+------------+----------------+------------------------+------------+-----------------+
|  1 | Ram        | Prasad         | ram@gmail.com          | 9024554745 | 123 main street |
|  2 | Sandiya    | Vishwanath     | sandiya@gmail.com      | 9174543526 | 543 anna nagar  |
|  3 | Jayanthi   | Selvam         | selvam@gmail.com       | 9082707895 | 321 Ranipet     |
|  4 | Swetha     | Seetharaman    | swetha@gmail.com       | 7098645321 | 456 White Town  |
|  5 | Divya      | Dharshini      | divya@gmail.com        | 9123765480 | 890 Madagadipet |
|  6 | Nisha      | Vaithiyanathan | nisha@gmail.com        | 9865432178 | 698 Sowkarpet   |
|  7 | Darshini   | Balamurali     | darshnini@gmail.com    | 709834521  | 987 nehru nagar |
|  8 | Agalya     | Shanmugam      | agalya@gmail.com       | 8143256790 | 678 Gandhi park |
|  9 | Harini     | Murugavel      | harini@gmail.com       | 9024554745 | 234 Semmandalam |
| 10 | Selva      | Ramaiah        | selva@gmail.com        | 9156473420 | 567 Manjakupam  |
```

-- products table

insert into products (product_name, description, price) values

('Laptop', 'High-performance laptop with Intel Core i7', 67000),

('Smartphone', 'Latest smartphone with 6.5-inch display', 30000),

('Headphones', 'Wireless noise-canceling headphones', 300),

('Smartwatch', 'Fitness tracker and smartwatch combo', 5000),

('Camera', 'Mirrorless camera with 24MP sensor', 20000),

('Printer', 'Color inkjet printer with wireless capability', 50000),

('Tablet', '10-inch tablet with high-resolution display', 70000),

('Desktop', 'Powerful desktop computer for gaming', 80000),

('Monitor', 'High-speed wireless router for home network', 75000),

('Mouse', '2TB external hard drive for backup', 8000);

```
mysql> select * from products;
+----+--------------+-------------------------------------------------+--------+
| id | product_name | description                                     | price  |
+----+--------------+-------------------------------------------------+--------+
|  1 | Laptop       | High-performance laptop with Intel Core i7      |  73700 |
|  2 | Smartphone   | Latest smartphone with 6.5-inch display         |  33000 |
|  3 | Headphones   | Wireless noise-canceling headphones             |    330 |
|  4 | Smartwatch   | Fitness tracker and smartwatch combo            |   5500 |
|  5 | Camera       | Mirrorless camera with 24MP sensor              |  22000 |
|  6 | Printer      | Color inkjet printer with wireless capability   |  55000 |
|  7 | Tablet       | 10-inch tablet with high-resolution display     |  77000 |
|  8 | Desktop      | Powerful desktop computer for gaming            |  88000 |
|  9 | Monitor      | High-speed wireless router for home network     |  82500 |
| 10 | Mouse        | 2TB external hard drive for backup              |   8800 |
```

-- orders table

insert into orders (order_date, total_amount,customers_id) values

('2020-12-01',134000,1),

('2024-01-02',30000,2),

('2023-04-03',1500,3),

('2021-11-04', 30000,4),

('2020-09-05',140000,5),

('2022-04-06',120000,6),

('2021-03-07',100000 ,7),

('2022-10-08',320000 ,8),

('2023-11-09', 150000,10),

('2024-02-11', 335000,1),

('2024-05-21',700000 ,3),

('2022-06-30',1500 ,2),

('2023-08-14',32000 ,8),

('2024-09-07',490000 ,7),

('2024-03-04',67000 ,7);

```
mysql> select*from orders;
+----+------------+--------------+-------------+---------------+
| id | order_date | total_amount | customers_id | orders_status |
+----+------------+--------------+-------------+---------------+
|  1 | 2020-12-01 |       147400 |           1 | pending       |
|  2 | 2024-01-02 |        33000 |           2 | shipped       |
|  3 | 2023-04-03 |         1650 |           3 | pending       |
|  4 | 2021-11-04 |        33000 |           4 | shipped       |
|  6 | 2022-04-06 |       132000 |           6 | shipped       |
|  7 | 2021-03-07 |       110000 |           7 | pending       |
|  8 | 2022-10-08 |       352000 |           8 | shipped       |
|  9 | 2023-11-09 |       165000 |          10 | pending       |
| 10 | 2024-02-11 |       368500 |           1 | shipped       |
| 11 | 2024-05-21 |       770000 |           3 | pending       |
| 12 | 2022-06-30 |         1650 |           2 | shipped       |
| 13 | 2023-08-14 |        35200 |           8 | pending       |
| 14 | 2024-09-07 |       539000 |           7 | shipped       |
| 15 | 2024-03-04 |        73700 |           7 | pending       |
+----+------------+--------------+-------------+---------------+
```

-- order_details table

insert into order_details (orders_id, products_id,quantity) VALUES

(1, 1,2),

(2, 2,1),

(3, 3,5),

(4, 4,6),

(5, 5,7),

(6, 5,6),

(7, 6,2),

(8, 8,4),

(9, 9,2),

(10, 1,5),

(11, 7,10),

(12, 3,5),

(13, 10,4),

(14, 7,7),

(15, 1,1);

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

-- inventory table

insert into inventory(quantity_in_stock, last_stock_update,products_id) values

(20,'2020-12-01',1),

(5,'2024-01-02',2),

(30,'2023-04-13',3),

(25,'2020-09-05',4),

(10,'2024-02-11',5),

(15,'2023-11-27',6),

(8,'2021-04-11',7),

(8,'2022-05-20',8),

(35,'2022-11-09',9),

(20,'2024-07-03',10);

```
mysql> select*from inventory;
+----+-------------------+-------------------+-------------+
| id | quantity_in_stock | last_stock_update | products_id |
+----+-------------------+-------------------+-------------+
|  1 |                20 | 2020-12-01        |           1 |
|  2 |                 5 | 2024-01-02        |           2 |
|  3 |                30 | 2023-04-13        |           3 |
|  4 |                25 | 2020-09-05        |           4 |
|  5 |                10 | 2024-02-11        |           5 |
|  6 |                15 | 2023-11-27        |           6 |
|  7 |                 8 | 2021-04-11        |           7 |
|  8 |                 8 | 2022-05-20        |           8 |
|  9 |                35 | 2022-11-09        |           9 |
| 10 |                20 | 2024-07-03        |          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 names, email
from customers;
```

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

```
select  o.id,o.order_date,concat( c.first_name,' ', c.last_name) as names
from orders o,customers c
where o.customers_id = c.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 customers(first_name,last_name,email,address)
values('Dhana', 'lakshmi','dhana@gmail.com','IG');
```

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

```
update products
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=5;
```

/* 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
('2020-09-05','140000', 5);
```

/* 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 customers
set email='dhanalakshmi@gmail.com',address='567 IG'
where id=14;
```

/* 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 products p join order_details od on p.id=od.products_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. */

```
delete o,s
from order_details o,orders s
where s.id=o.orders_id and s.id=5;
```

/* 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 products(product_name, description, price) VALUES
('Apple iphone', 'High-performance and Security', 150000);
```

/* 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 orders_status varchar(255);

update orders
set orders_status = case when id % 2 = 1 then 'shipped' else 'pending' end;
```

```
update orders
set orders_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 customers
add num_of_orders int;

update customers c
set num_orders=(select count(*)
                from orders o
                where c.id=o.customers_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 names,c.phone,
c.email,c.num_of_orders,o.order_date,o.total_amount,o.orders_status
from customers c join orders o on c.id=o.customers_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.product_name,sum(o.total_amount) as revenue
from products p join order_details od on p.id=od.products_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) as names,c.phone,c.address
```

```
from customers c
group by c.id
having count(c.num_orders)>=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.product_name,sum(od.quantity) as popular_gadget
from products p join order_details od on p.id=od.products_id
group by p.id
order by popular_gadget limit 0,1;
```

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

```
alter table products
add category varchar(255);

update products
set category=case
        when id=1 then 'computing device'
    when id=2 then 'communication device'
    when id=3 then 'audio device'
    when id=4 then 'wearable'
    when id=5 then 'imaging device'
    when id=6 then 'printing device'
    when id=7 then 'computing device'
    when id=8 then 'computing device'
    when id=9 then 'i/o device'
    when id=10 then 'input device'
    else 'gadget' end;
```

```
select category,product_name
from products
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,c.last_name,avg(o.total_amount) as avg_order_value
from customers c join orders o on c.id=o.customers_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 c.*,o.id,o.total_amount
from customers c join orders o on c.id=o.customers_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.product_name,count(p.id)as times_orders
from products p join order_details od on p.id=od.products_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.product_name,c.first_name,c.last_name
from customers c join orders o on c.id=o.customers_id
        join order_details od on o.id=od.orders_id
        join products p on p.id=od.products_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 id,sum(total_amount) as revenue
from orders
where order_date between '2023-11-09' and '2024-05-21'
group by id;
```

-- 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 names
from customers
where id not in(select customers_id from orders);
```

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

```
select i.products_id,(i.quantity_in_stock- (select sum(od.quantity)
from order_details od
where od.products_id=i.products_id)) as total_products
from inventory i ;
```

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

```
select sum(total_amount) as revenue_by_techshop
from (
select total_amount from orders) as revenue;
```

/* 4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter. */

```
select p.category,(
        select avg(od.quantity)
        from order_details od
```

```
        where od.id in(
                select id
                from products
                where category=p.category))as avg_quantity
from products 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.customers_id=c.id
group by o.customers_id)as total_revenue
from customers 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 c.first_name,count(o.id)as num_orders
from customers c join orders o on c.id = o.customers_id
group by c.id,c.first_name
order by num_orders desc
limit 1;
```

/* 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.product_name,p.category,(
        select sum(od.quantity)
        from order_details od
        where p.id=od.products_id
        group by od.products_id)as total_quantity
```

```
from products p
order by product desc
limit 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 name,(
        select sum(total_amount) from orders o
        where c.id=o.customers_id
        group by o.customers_id) as money_spent
from customers c
order by money_spent desc limit 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 c.id=o.customers_id)as total_value
from customers c
order by total_value;
```

/* 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.id)  from orders o
        where c.id=o.customers_id
        group by o.customers_id) as number_of_orders
from customers c;
```