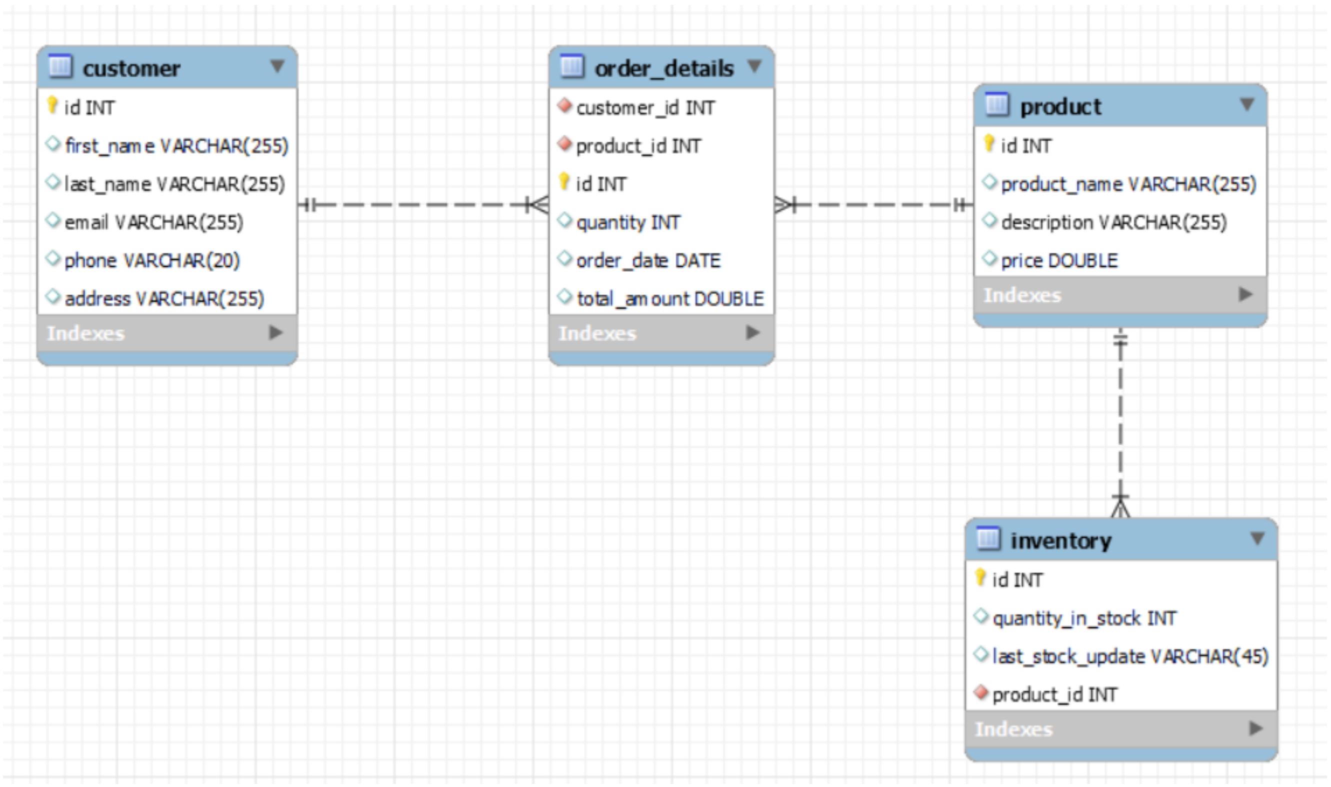
ASSIGNMENT 5 – TECHSHOP (ELECTRONIC GADGETS)

ER DIAGRAM:



Queries:

-- Task 1:.....

```
-- MySQL Workbench Forward Engineering
-- -----------------------------------------------------
-- Schema tech_shop
-- -----------------------------------------------------
-- -----------------------------------------------------
-- Schema tech_shop
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `tech_shop` DEFAULT CHARACTER SET utf8 ;
USE `tech_shop` ;

-- -----------------------------------------------------
-- Table `tech_shop`.`customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `tech_shop`.`customer` (
  `id` INT NOT NULL,
  `first_name` VARCHAR(255) NULL,
  `last_name` VARCHAR(255) NULL,
  `email` VARCHAR(255) NULL,
  `phone` VARCHAR(20) NULL,
  `address` VARCHAR(255) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
```

```sql
-- -----------------------------------------------------
-- Table `tech_shop`.`product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `tech_shop`.`product` (
  `id` INT NOT NULL,
  `product_name` VARCHAR(255) NULL,
  `description` VARCHAR(255) NULL,
  `price` DOUBLE NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `tech_shop`.`inventory`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `tech_shop`.`inventory` (
  `id` INT NOT NULL,
  `quantity_in_stock` INT NULL,
  `last_stock_update` VARCHAR(45) NULL,
  `product_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_inventory_product1_idx` (`product_id` ASC) ,
  CONSTRAINT `fk_inventory_product1`
    FOREIGN KEY (`product_id`)
    REFERENCES `tech_shop`.`product` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `tech_shop`.`order_details`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `tech_shop`.`order_details` (
  `customer_id` INT NOT NULL,
  `product_id` INT NOT NULL,
  `id` INT NOT NULL,
  `quantity` INT NULL,
  `order_date` DATE NULL,
  `total_amount` DOUBLE NULL,
  INDEX `fk_customer_has_product_product1_idx` (`product_id` ASC) ,
  INDEX `fk_customer_has_product_customer1_idx` (`customer_id` ASC) ,
  PRIMARY KEY (`id`),
  CONSTRAINT `fk_customer_has_product_customer1`
    FOREIGN KEY (`customer_id`)
    REFERENCES `tech_shop`.`customer` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_customer_has_product_product1`
    FOREIGN KEY (`product_id`)
    REFERENCES `tech_shop`.`product` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

insert into customer (id, first_name, last_name, email, phone, address)
values
(1, 'John', 'Doe', 'john.doe@example.com', '1234567890', 'Mumbai'),
(2, 'Jane', 'Smith', 'jane.smith@example.com', '9876543210', 'Delhi'),
(3, 'Mike', 'Johnson', 'mike.johnson@example.com', '5551234567', 'Kolkata'),
(4, 'Emily', 'Williams', 'emily.williams@example.com', '7779876543', 'Chennai'),
(5, 'Daniel', 'Brown', 'daniel.brown@example.com', '1237894560', 'Banglore'),
(6, 'Eva', 'Taylor', 'eva.taylor@example.com', '5557891234', 'Pondicherry'),
(7, 'Alex', 'Clark', 'alex.clark@example.com', '9876543210', 'Karaikal'),
(8, 'Sophia', 'Moore', 'sophia.moore@example.com', '4445678901', 'Mumbai'),
(9, 'Carter', 'Anderson', 'carter.anderson@example.com', '2223456789', 'Gujarat'),
(10, 'Olivia', 'Turner', 'olivia.turner@example.com', '3336789012', 'Kolkata');

insert into product (id, product_name, description, price)
values
(1, 'Laptop', 'Powerful laptop with high performance', 999.99),
(2, 'Smartphone', 'Latest model with advanced features', 599.99),
(3, 'Headphones', 'High-quality over-ear headphones', 149.99),
(4, 'Tablet', 'Slim and lightweight tablet', 299.99),
(5, 'Camera', 'Professional DSLR camera', 899.99),
(6, 'Keyboard', 'Mechanical gaming keyboard', 79.99),
(7, 'Mouse', 'Wireless optical mouse', 29.99),
(8, 'Monitor', '27-inch IPS display monitor', 399.99),
(9, 'Printer', 'All-in-one laser printer', 199.99),
(10, 'External Hard Drive', '1TB portable external hard drive', 79.99);

insert into inventory (id, quantity_in_stock, last_stock_update, product_id)
values
(1, 50, 'March', 1),
(2, 100, 'December', 2),
(3, 25, 'January', 3),
(4, 30, 'February', 4),
(5, 15, 'March', 5),
(6, 50, 'Novemmber', 6),
(7, 75, 'February', 7),
(8, 20, 'March', 8),
(9, 10, 'January', 9),
(10, 30, 'March', 10);
```

```sql
insert into order_details (customer_id, product_id, id, quantity, order_date, total_amount)
values
(1, 1, 1, 2, '2024-02-07', 1999.98),
(2, 2, 2, 1, '2024-01-17', 599.99),
(3, 3, 3, 3, '2024-02-27', 449.97),
(4, 4, 4, 1, '2024-01-15', 299.99),
(5, 5, 5, 2, '2024-02-21', 1799.98),
(6, 6, 6, 1, '2024-01-02', 79.99),
(7, 7, 7, 2, '2024-02-05', 59.98),
(8, 8, 8, 1, '2024-02-26', 399.99),
(9, 9, 9, 3, '2024-03-05', 599.97),
(10, 10, 10, 1, '2024-03-07', 79.99);

-- Task 2: ....

-- Q1. Write an SQL query to retrieve the names and emails of all customers.
select first_name, email from customer;

-- Q2. Write an SQL query to list all orders with their order dates and corresponding customer names.
select o.id as order_id, o.order_date, c.first_name as customer_name
    from order_details o join customer c
    where c.id=o.customer_id;

-- Q3. 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(id, first_name, last_name, email, phone, address)
    values(11,'Swethaa','Gayathri','swetha@exxample.com','9876543217','Chennai');

-- Q4. 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);

-- Q5. 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 from order_details
    where id=7;

-- Q6. 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 order_details(customer_id,product_id,id,quantity,order_date,total_amount)
    values(2,2,11,1,'2024-03-07',5000.00);
```

```
-- Q7. 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='john@gmail.com' , address='Delhi'
    where id=1;

-- Q8. 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 order_details o
    join product p on o.product_id = p.id
    set o.total_amount = p.price * o.quantity;

-- Q9. 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 from order_details
    where customer_id=3;

-- Q10. 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 values(21,'Earpods','High noise cancellation earpods',2999.00);

-- Q11. 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.
-- no such column exists

-- Q12. 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.
-- no such column exists in customer table like number of orders

-- Task 3: ....

-- Q1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.
select c.first_name as customer_name, o.*
    from customer c join order_details o
    on c.id=o.customer_id;

-- Q2. 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 Total_revenue
    from order_details o join product p
    on p.id=o.product_id
    group by o.product_id;
```

-- Q3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.

```
 select c.*
    from customer c right join order_details o
    on c.id=o.customer_id;
```

-- Q4. 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(o.quantity) as total_quantity
    from product p join order_details o
    on p.id=o.product_id
    group by o.product_id
    order by total_quantity desc
    limit 1;
```

-- Q5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.

-- no specific column related to category

-- Q6. 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) as avg_order_value
    from customer c join order_details o
    on c.id=o.customer_id
    group by o.customer_id;
```

-- Q7. 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.first_name as customer_name, sum(o.total_amount) as total_revenue
    from customer c join order_details o
    on c.id=o.customer_id
    group by o.customer_id
    order by total_revenue desc
    limit 1;
```

-- Q8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.

```
select p.product_name, count(o.product_id) as no_of_times
    from product p join order_details o
    on p.id=o.product_id
    group by o.product_id;
```

-- Q9. 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 c.first_name, p.product_name
    from customer c join order_details o
```

```sql
    on c.id=o.customer_id join product p
    on o.product_id=p.id
    where p.product_name='Laptop';

-- Q10. 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 order_details
    where order_date between '2024-03-01' and '2024-03-07';

-- Task 4:....

-- Q1. Write an SQL query to find out which customers have not placed any orders.
select c.first_name
    from customer c
    join order_details o
    on c.id=o.customer_id
    where o.customer_id is null;

-- Q2. Write an SQL query to find the total number of products available for sale.
select count(p.id) as count_of_products
    from product p;

-- Q3. Write an SQL query to calculate the total revenue generated by TechShop.
select sum(total_amount) as total_revenue
    from order_details;

-- Q4. 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.
-- no specific column related to category

-- Q5. 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 c.first_name, sum(o.total_amount) as total_revenue
    from customer c join order_details o
    on c.id=o.customer_id
    where c.id=2;

-- Q6. 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 no_of_orders
    from customer c join order_details o
    on c.id=o.customer_id
    group by o.customer_id
    order by no_of_orders desc
    limit 1;
```

```
-- Q7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.
-- no specific column named category

-- Q8. 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 c.first_name as customer_name, sum(o.total_amount) as total_spending
    from customer c join order_details o
    on c.id=o.customer_id
    group by o.customer_id
    order by total_spending desc
    limit 1;

-- Q9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.
select c.first_name, avg_order_value
    from customer c join order_details o
    on c.id=o.customer_id
    where avg_order_value=(sum(o.total_amount)/count(o.product_id))
    group by o.customer_id;

-- Q10. 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 c.first_name, count(o.product_id) as no_of_orders
    from customer c join order_details o
    on c.id=o.customer_id
    group by o.product_id;
```