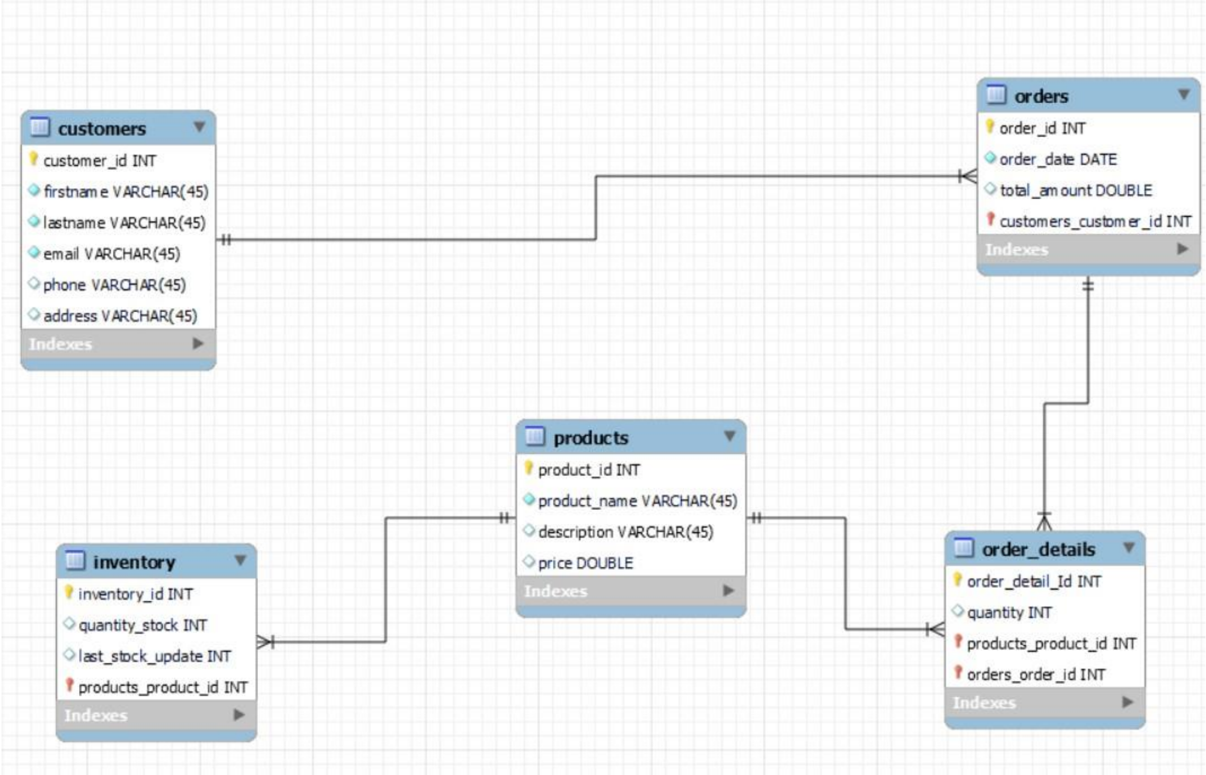
ELECTRONIC GADGET SHOP

ER DIAGRAM:



customers
customer_id INT
firstname VARCHAR(45)
lastname VARCHAR(45)
email VARCHAR(45)
phone VARCHAR(45)
address VARCHAR(45)

orders
order_id INT
order_date DATE
total_amount DOUBLE
customers_customer_id INT

products
product_id INT
product_name VARCHAR(45)
description VARCHAR(45)
price DOUBLE

inventory
inventory_id INT
quantity_stock INT
last_stock_update INT
products_product_id INT

order_details
order_detail_Id INT
quantity INT
products_product_id INT
orders_order_id INT

TASK – 1

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

USE `techshop` ;

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

-- Table `techshop`.`customers`

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

CREATE TABLE IF NOT EXISTS `techshop`.`customers` (

  `customer_id` INT NOT NULL AUTO_INCREMENT,
  `firstname` VARCHAR(45) NOT NULL,
  `lastname` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `phone` VARCHAR(45) NULL,
  `address` VARCHAR(45) NULL,
  PRIMARY KEY (`customer_id`))
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `techshop`.`products`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `techshop`.`products` (
  `product_id` INT NOT NULL AUTO_INCREMENT,
  `product_name` VARCHAR(45) NOT NULL,
  `description` VARCHAR(45) NULL,
  `price` DOUBLE NULL,
  PRIMARY KEY (`product_id`))
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `techshop`.`orders`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `techshop`.`orders` (
  `order_id` INT NOT NULL AUTO_INCREMENT,
  `order_date` DATE NOT NULL,
  `total_amount` DOUBLE NULL,
  `customers_customer_id` INT NOT NULL,
  PRIMARY KEY (`order_id`, `customers_customer_id`),
  INDEX `fk_orders_customers_idx` (`customers_customer_id` ASC) ,
  CONSTRAINT `fk_orders_customers`
    FOREIGN KEY (`customers_customer_id`)
    REFERENCES `techshop`.`customers` (`customer_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `techshop`.`order_details`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `techshop`.`order_details` (
  `order_detail_Id` INT NOT NULL AUTO_INCREMENT,
  `quantity` INT NULL,
  `products_product_id` INT NOT NULL,
  `orders_order_id` INT NOT NULL,
  PRIMARY KEY (`order_detail_Id`, `products_product_id`, `orders_order_id`),
  INDEX `fk_order_details_products1_idx` (`products_product_id` ASC) ,
  INDEX `fk_order_details_orders1_idx` (`orders_order_id` ASC) ,
  CONSTRAINT `fk_order_details_products1`
    FOREIGN KEY (`products_product_id`)
    REFERENCES `techshop`.`products` (`product_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_order_details_orders1`
    FOREIGN KEY (`orders_order_id`)
    REFERENCES `techshop`.`orders` (`order_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `techshop`.`inventory`
-- -----------------------------------------------------
```

```
CREATE TABLE IF NOT EXISTS `techshop`.`inventory` (
  `inventory_id` INT NOT NULL AUTO_INCREMENT,
  `quantity_stock` INT NULL,
  `last_stock_update` INT NULL,
  `products_product_id` INT NOT NULL,
  PRIMARY KEY (`inventory_id`, `products_product_id`),
  INDEX `fk_inventory_products1_idx` (`products_product_id` ASC) ,
  CONSTRAINT `fk_inventory_products1`
    FOREIGN KEY (`products_product_id`)
    REFERENCES `techshop`.`products` (`product_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


use techshop;
 show tables;
 desc customers;
 desc inventory;
 desc order_details;
 desc orders;
 desc products;
insert into customers(firstname,lastname,email,phone,address)
values('dolu','kappor','dolu@gmail.com','8745961254','mumbai')
,('bolu','bhatt','bolu@gmail.com','8745961254','mumbai'),('chutki','shetty','chutki@gmail.com','9745964254','chennai'),
('chotta','bheem','chotta@gmail.com','7548548796','chennai');
select * from customers;
insert into products(product_name,description,price)
values('motherboard','computer',3500),('speaker','4d effect',6500),
('headphones','wired',1000),('cameras','digital',7500);
```

```
select * from products;

insert into orders(order_date,total_amount,customers_customer_id) values('2024-01-
08',3500,1),('2024-03-05',6000,4),

('2024-04-14',8500,1),('2024-02-11',7000,3),('2024-01-21',14000,2);

select * from orders;

insert into inventory(quantity_stock,last_stock_update,products_product_id)

values(40,25,2),(75,35,1),(50,15,3),(60,12,4),(85,30,2);

select * from inventory;

insert into order_details(quantity,products_product_id,orders_order_id)
values(3,2,4),(5,1,3),(1,3,5),(2,4,2),(3,1,4);

select * from order_details;
```

TASK - 2

#1.sql query to retrieve the names and emails of all customers

```
select firstname,lastname,email from customers;
```

#2.query to list all orders with their order dates and corresponding customer names

```
select o.order_id,o.order_date,c.firstname

from orders o,customers c

where c.customer_id=o.customers_customer_id;
```

#3.query to insert a new customer record into the customers table.Include customer information such as name,email, and address.

```
insert into customers(firstname,lastname,email,phone,address)
values("kaliya","bhai","kaliya@gmail.com","7384569213","bangalore");
```

#4.query to update the prices od all eletectronic gadgets in the products table by increasing them by 10%

```
update products

set price=price+(price*0.1);
```

#5.query to inserrt a new order into the orders table.

```
insert into orders(order_date,total_amount,customers_customer_id) values("2024-03-15",2400,3);
```

#6.update the contact information of a secific customer

```
update customers

set phone="9944022631"

where customer_id=3;
```

#7.insert new electronic product into product table

```
insert into products(product_name,description,price) values("air pods","wireless",3500);
```

#8.calculate no of orders placed by each customers

```
select customers_customer_id,count(customers_customer_id)

from orders

group by customers_customer_id;
```

TASK - 3

#1.list of all orders along with customer info

```
select o.order_id,o.order_date,c.firstname,c.phone

from orders o join customers c

on c.customer_id=o.order_id;
```

#2.total revenue generated by each electronic gadget

```
select o.products_product_id,p.product_name,sum(o.quantity)*p.price as total_revenue from order_details o join products p

on p.product_id=o.order_detail_id

group by products_product_id;
```

#3.customers who have made atleast 1 purchase

```
select c.firstname,count(o.order_id) as orders
from customers c join orders o
on c.customer_id=o.order_id
group by order_id
having orders>=1;
```

#4.most popular gadget with highest quantity order

```
select p.product_name,o.quantity
from products p join order_details o
on p.product_id=o.products_product_id
order by o.quantity desc
limit 1;
```

#5.retrieve gadgets with their description

```
select product_name,description from products;
```

#6.customer name with average order value

```
select c.firstname,avg(o.total_amount)
from customers c join orders o
on c.customer_id=o.customers_customer_id
group by c.customer_id;
```

#7.order with highest total revenue

```
 select o.products_product_id,p.product_name,sum(o.quantity)*p.price as total_revenue
from order_details o join products p
on p.product_id=o.order_detail_id
group by products_product_id
order by total_revenue desc
limit 1;
```

#8. total revenue generated by all orders between specific time period

```
select sum(total_amount) as tot_revenue
from orders
where order_date between "2024-02-01" and "2024-03-31";
```

#9.customer along with purchased gadgets

```
select c.firstname,p.product_name
from customers c,orders o,order_details r,products p
where c.customer_id=o.customers_customer_id and o.order_id=r.orders_order_id and r.products_product_id=p.product_id;
```

TASK - 4

#1.customers have not placed any orders

```
select firstname from customers where customer_id not in(select customers_customer_id from orders);
```

#2.total no of products that are available for sale

```
select * from products;
select product_name from products where product_id in
(select products_product_id from inventory where last_stock_update>0);
```

#3.total revenue by tech shop

```
select(select sum(total_amount) from orders) as tot_revenue;
```

#4.customers who have placed most orders

```
select * from customers;
select * from orders;
-- alternative
```

```
select customers_customer_id,count(order_id) as most from orders group by customers_customer_id

order by most desc limit 1;
```

#5.total no of orders placed by each customer and list their names along with the order count

```
select c.firstname,o.order_id,count(o.order_id)

from customers c join orders o

on c.customer_id=o.customers_customer_id

group by c.firstname;
```

#6.customer who spent most money

```
select c.firstname,sum(o.total_amount) as revenue

from customers c join orders o

on c.customer_id=o.customers_customer_id

group by c.firstname

order by revenue desc;
```