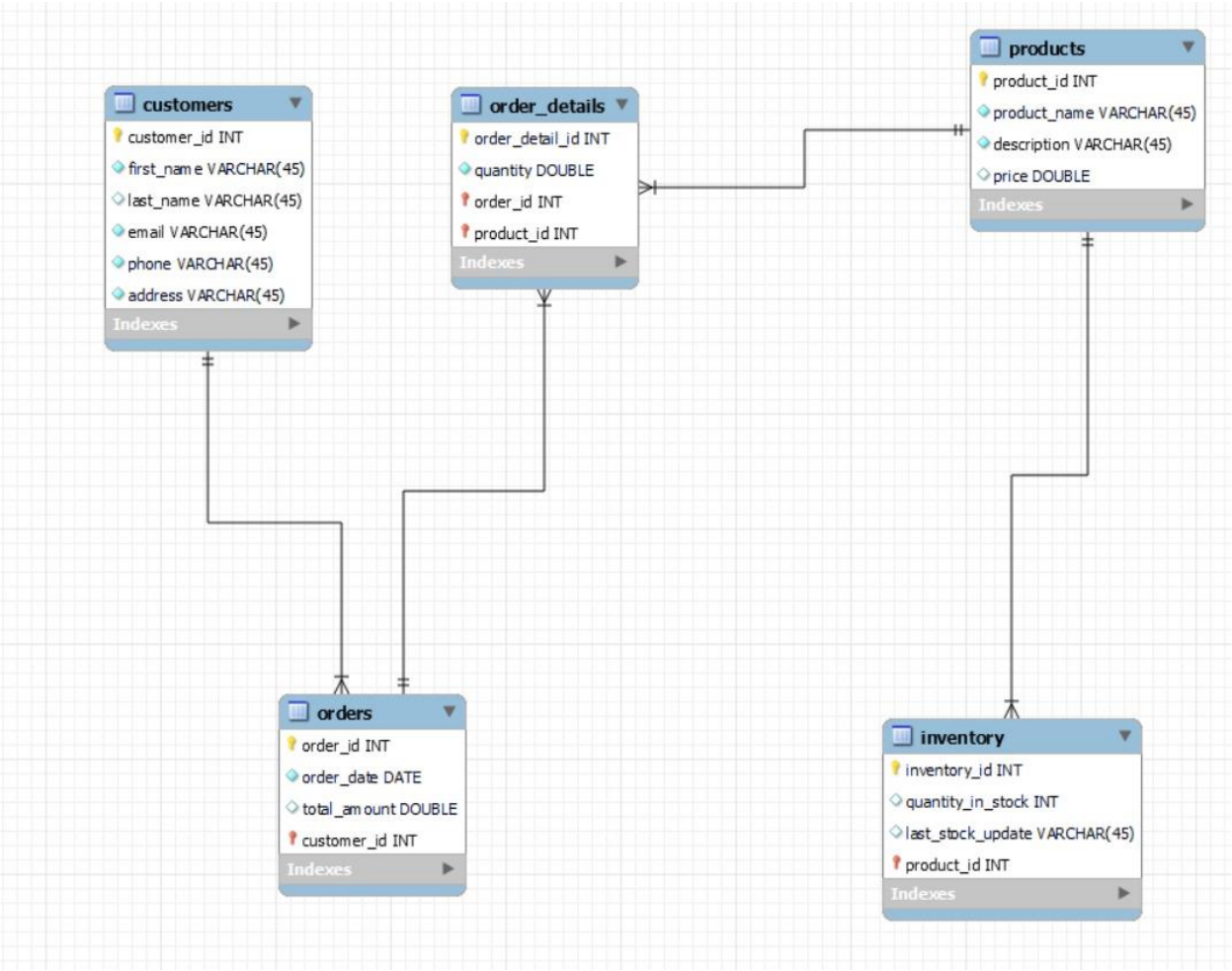
GADGETS

ER DIAGRAM



use gad;

show databases;

show tables;

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

('kareena','kappoor','kareena@gmail.com',9843567555,'England'),

('Alia','Butt','alia@gmail.com',9847777555,'Germany'),

('Amir','khan','amirgmail.com',8881567555,'England'),

('Salman','khan','salman@gmail.com',9843197555,'Canada'),

('Radhika',null,'radhika@gmail.com',9843567663,'Germany');

```
select * from customers;
insert into orders(order_date,total_amount,customer_id)values
('2024-01-01',10000,1),('2024-01-28',15000,2),('2024-02-14',25000,3),
('2024-03-01',10000,4),('2024-02-02',20000,5);
select * from orders;
insert into products(product_name,description,price) values ('Laptop','ABC with Graphic card',60000),
('TV','ACB with high performance of 3D Quality',90000),
('Coffee Maker', 'Programmable, 6-cup capacity', 25000),
('Digital Camera', '20MP resolution, 4K video recording', 15000),
('Fitness Tracker', 'Heart rate monitor, step counter, sleep tracker', 5500);
select * from products;
insert into order_details(quantity,order_id,product_id) values (5,1,2),
(10,2,3),(5,3,4),(15,4,2),(20,5,5);
select * from order_details;
insert into inventory(quantity_in_stock,last_stock_update,product_id)values(10, '2024-03-07', 1),
(15, '2024-02-10', 2),(5, '2024-01-20',2),
(20, '2024-03-08', 3),(10, '2024-03-04',4);
select * from inventory;
#---------------------------------#



-- Tasks 2
-- 1. Write an SQL query to retrieve the names and emails of all customers.
 SELECT first_name,last_name, email FROM customers;
-- 2. Write an SQL query to list all orders with their order dates and corresponding customer
-- names.
SELECT o.order_id, o.order_date, c.first_name FROM orders o
JOIN customers c ON o.customer_id = c.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 customers (first_name,last_name, email, address)
VALUES ('Jack', 'Mark', 'jack@example.com','Europe');
```

```
-- 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 * 1.1;
-- 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 FROM order_details WHERE order_id =3;
-- 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
('2024-01-17',10500,4);
-- 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 = 'radhi@gmail.com', address = 'Canada',phone=9878987898
WHERE customer_id = 5;
-- 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.
-- 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
delete from orders where customer_id=3;
-- 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
('ABC','With Graphic card',60685);
 --- Task 3
-- 1. Write an SQL query to retrieve a list of all orders along with customer information (e.g.,
-- customer name) for each order.
SELECT o.order_id, o.order_date, o.total_amount, c.first_name,c.last_name
FROM orders o JOIN customers c ON o.customer_id = c.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.
```

```
-- 3. Write an SQL query to list all customers who have made at least one purchase. Include their
-- names and contact information.
SELECT c.customer_id,c.first_name,c.email,c.address
FROM customers c JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id,c.first_name,c.email,c.address;
-- 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.
-- 5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding
-- categories.
SELECT product_name, description
FROM products WHERE description = '%electronic%';
-- 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 average_order_value
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.first_name;
-- 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.order_id,c.first_name,c.last_name,c.email,c.address,
SUM(od.quantity * p.price) AS total_revenue
FROM orders o JOIN customers c ON o.customer_id = c.customer_id
JOIN order_details od ON o.order_id = od.order_id
JOIN products p ON od.product_id = p.product_id
GROUP BY o.order_id,c.first_name,c.last_name,c.email,c.address
ORDER BY total_revenue DESC LIMIT 1;
-- 8. Write an SQL query to list electronic gadgets and the number of times each product has been
-- ordered.
SELECT p.product_name, COUNT(od.product_id) AS order_count
FROM products p
JOIN order_details od ON p.product_id = od.product_id
```

```
JOIN orders o ON od.order_id = o.order_id
WHERE p.product_name = 'electronic'
GROUP BY p.product_name;
-- 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 DISTINCT c.first_name,c.last_name,c.email,c.address
FROM customers c JOIN orders o ON c.customer_id = o.customer_id
JOIN order_details od ON o.order_id = od.order_id
WHERE o.order_id IN (SELECT o.order_id FROM orders o
JOIN order_details od ON o.order_id = od.order_id
JOIN products p ON od.product_id = p.product_id
WHERE p.product_name = 'TV');
-- 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(od.quantity * p.price) AS total_revenue FROM orders o
JOIN order_details od ON o.order_id = od.order_id
JOIN products p ON od.product_id = p.product_id
WHERE o.order_date BETWEEN '2024-01-01' AND '2024-03-10';
-- Task 4.
-- 1. Write an SQL query to find out which customers have not placed any orders.
SELECT c.customer_id, c.first_name,c.email,c.address
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id
WHERE o.customer_id IS NULL;
-- 2. Write an SQL query to find the total number of products available for sale.
SELECT COUNT(*) AS total_products_available
FROM products;
-- 3. Write an SQL query to calculate the total revenue generated by TechShop.
SELECT SUM(total_amount) AS total_revenue_generated
FROM orders;
-- 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.
```

```
-- 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 SUM(od.quantity * p.price) AS total_revenue_generated
FROM order_details od
JOIN orders o ON od.order_id = o.order_id
JOIN products p ON od.product_id = p.product_id
WHERE o.customer_id = 4;
-- 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.
-- 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 c.first_name,SUM(od.quantity * p.price) AS total_spending
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_details od ON o.order_id = od.order_id
JOIN products p ON od.product_id = p.product_id
WHERE p.product_name = 'electronic'
GROUP BY c.customer_id, c.first_name
ORDER BY total_spending 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.
-- 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 c.first_name,COUNT(o.order_id) AS order_count
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.first_name;
```