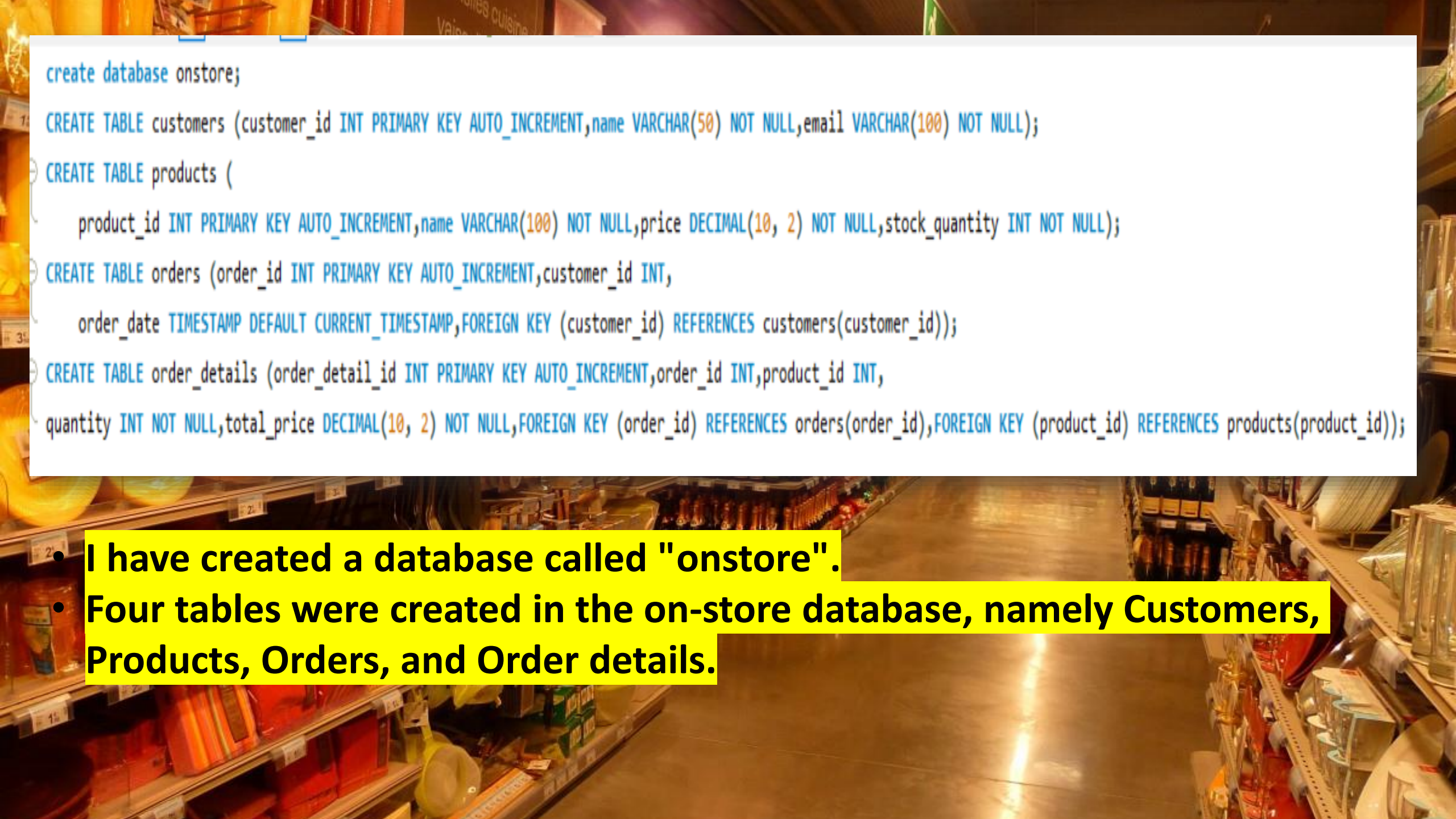


# Online Store Database Management –MYSQL Project





```
create database onstore;

CREATE TABLE customers (customer_id INT PRIMARY KEY AUTO_INCREMENT,name VARCHAR(50) NOT NULL,email VARCHAR(100) NOT NULL);

CREATE TABLE products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,name VARCHAR(100) NOT NULL,price DECIMAL(10, 2) NOT NULL,stock_quantity INT NOT NULL);

CREATE TABLE orders (order_id INT PRIMARY KEY AUTO_INCREMENT,customer_id INT,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,FOREIGN KEY (customer_id) REFERENCES customers(customer_id));

CREATE TABLE order_details (order_detail_id INT PRIMARY KEY AUTO_INCREMENT,order_id INT,product_id INT,
    quantity INT NOT NULL,total_price DECIMAL(10, 2) NOT NULL,FOREIGN KEY (order_id) REFERENCES orders(order_id),FOREIGN KEY (product_id) REFERENCES products(product_id));
```

- I have created a database called "onstore".
- Four tables were created in the on-store database, namely Customers, Products, Orders, and Order details.





```
INSERT INTO customers (name, email) VALUES  
('Abhi shaw', 'abhishaw@gmail.com'),  
('Sam malhotra', 'samm@gmail.com');
```

```
INSERT INTO customers (name, email) VALUES  
('Abhishek shaw', 'abhi20shaw@gmail.com'),  
('Samir singh', 'samsingh@gmail.com');
```

```
INSERT INTO products (name, price, stock_quantity) VALUES  
('Laptop', 1200.00, 10),  
('Headphones', 99.99, 50),  
('Smartphone', 699.99, 20);
```

```
INSERT INTO orders (customer_id) VALUES  
(1),  
(2);
```

```
INSERT INTO orders (customer_id) VALUES(3),(5);
```

```
INSERT INTO order_details (order_id, product_id, quantity, total_price) VALUES  
(1, 1, 2, 2400.00),  
(1, 2, 1, 99.99),  
(2, 3, 3, 2099.97);
```



## Customer table

customer_id	name	email
1	Abhi shaw	abhishaw@gmail.com
2	Sam malhotra	samm@gmail.com
3	Alex patra	patraalex@gmail.com
4	Abhishek shaw	abhi20shaw@gmail.com
5	Samir singh	samsingh@gmail.com
NULL	NULL	NULL

## Order details table

order_detail_id	order_id	product_id	quantity	total_price
1	1	1	2	2400.00
2	1	2	1	99.99
3	2	3	3	2099.97
5	4	3	2	1399.98
NULL	NULL	NULL	NULL	NULL



## Orders table

order_id	customer_id	order_date
1	1	2023-12-18 18:29:48
2	2	2023-12-18 18:29:48
4	3	2023-12-18 18:36:15
5	3	2023-12-18 18:40:44
6	5	2023-12-18 18:40:44
NULL	NULL	NULL

## Products table

product_id	name	price	stock_quantity
1	Laptop	1200.00	10
2	Headphones	99.99	30
3	Smartphone	699.99	20
NULL	NULL	NULL	NULL



#Q.1 Retrieve the total revenue for each order?

```
SELECT order_id, SUM(total_price) AS order_total FROM order_details GROUP BY order_id;
```

Output:

order_id	order_total
1	2499.99
2	2099.97
4	1399.98



#Q.2 Find the customer who made the highest total purchase?

```
SELECT c.name AS customer_name, SUM(od.total_price) AS total_purchase FROM customers c  
JOIN orders o ON c.customer_id = o.customer_id  
JOIN order_details od ON o.order_id = od.order_id GROUP BY c.customer_id ORDER BY total_purchase DESC LIMIT 1;
```

Output:

customer_name	total_purchase
Abhi shaw	2499.99



#Q.3 List all products with low stock (quantity less than 5)?

```
SELECT * FROM products WHERE stock_quantity < 5;
```

Output:

product_id	name	price	stock_quantity
NULL	NULL	NULL	NULL



#Q.4 Calculate the average order total price?

```
SELECT AVG(total_price) AS avg_order_total FROM order_details;
```

Output:

avg_order_total
-----------------

1499.985000
-------------





#Q.5 Update the stock quantity of 'Headphones' to 30?

```
UPDATE products SET stock_quantity = 30 WHERE name = 'Headphones';
```

#Q.6 Insert a new customer and an order for a 'Smartphone' (product\_id = 3) with a quantity of 2?

```
INSERT INTO customers (name, email) VALUES ('Alex patra', 'patraalex@gmail.com');
```

```
SET @new_customer_id = LAST_INSERT_ID();
```

```
INSERT INTO orders (customer_id) VALUES (3);
```

```
SET @new_order_id = LAST_INSERT_ID();
```

```
INSERT INTO order_details (order_id, product_id, quantity, total_price)
```

```
VALUES (4, 3, 2, (SELECT price FROM products WHERE product_id = 3) * 2);
```

#Q.7 Delete the order with order\_id = 3?

```
DELETE FROM orders WHERE order_id = 3;
```



#Q.8 Find the top 3 customers who have spent the most money in total?

```
SELECT c.customer_id, c.name AS customer_name, SUM(od.total_price) AS total_spent
```

```
FROM customers c JOIN orders o ON c.customer_id = o.customer_id
```

```
JOIN order_details od ON o.order_id = od.order_id GROUP BY c.customer_id, c.name ORDER BY total_spent DESC LIMIT 3;
```

Output:

customer_id	customer_name	total_spent
1	Abhi shaw	2499.99
2	Sam malhotra	2099.97
3	Alex patra	1399.98



#Q.9 Retrieve the customer who has made the highest single order?

```
SELECT c.customer_id, c.name AS customer_name, MAX(od.total_price) AS highest_order FROM customers c  
JOIN orders o ON c.customer_id = o.customer_id JOIN order_details od ON o.order_id = od.order_id  
GROUP BY c.customer_id, c.name ORDER BY highest_order DESC LIMIT 1;
```

Output:

customer_id	customer_name	highest_order
1	Abhi shaw	2400.00



#Q.10 List the products that have never been ordered?

```
SELECT p.product_id, p.name AS product_name FROM products p  
LEFT JOIN order_details od ON p.product_id = od.product_id WHERE od.order_detail_id IS NULL;
```

Output:

product_id	product_name
------------	--------------



#Q.11 Calculate the percentage of total revenue contributed by each product?

```
SELECT p.product_id, p.name AS product_name,  
       (SUM(od.total_price) / (SELECT SUM(total_price) FROM order_details)) * 100 AS revenue_percentage  
FROM products p LEFT JOIN order_details od ON p.product_id = od.product_id GROUP BY p.product_id, p.name;
```

Output:

product_id	product_name	revenue_percentage
1	Laptop	40.000400
2	Headphones	1.666517
3	Smartphone	58.333083



#Q.12 Identify customers who have ordered the same product more than once?

```
SELECT c.customer_id, c.name AS customer_name, od.product_id, COUNT(od.product_id) AS order_count  
FROM customers c JOIN orders o ON c.customer_id = o.customer_id JOIN order_details od ON o.order_id = od.order_id  
GROUP BY c.customer_id, c.name, od.product_id HAVING order_count > 1;
```

Output:

customer_id	customer_name	product_id	order_count
-------------	---------------	------------	-------------