

E-COMMERCE DATABASE QUERIES AND OUT IN MYSQL

SMARTPHONE OF THE YEAR



CREDIT CARD



FREE SHIPPING ALL
OVER AMERICA



100% RELIABLE
STORE



FREE
RETURN


```
30
31  -- Create orders table
32  ● ○ CREATE TABLE IF NOT EXISTS orders (
33      order_id INT PRIMARY KEY AUTO_INCREMENT,
34      customer_id INT NOT NULL,
35      order_date DATE NOT NULL,
36      total_amount DECIMAL(10, 2) NOT NULL CHECK (total_amount >= 0),
37      FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
38  );
39
```

Result Grid



Filter Rows:

Edit:



Export/Import:





	order_id	customer_id	order_date	total_amount
	1	1	2023-04-05	1099.98
	2	2	2023-04-10	699.99
*		NULL	NULL	NULL

```

55 • INSERT IGNORE INTO customers (name, email, registration_date) VALUES
56     ('John Doe', 'john@example.com', '2023-01-15'),
57     ('Jane Smith', 'jane@example.com', '2023-02-20'),
58     ('Alice Brown', 'alice@example.com', '2023-03-10'),
59     ('Bob Johnson', 'bob@example.com', '2023-04-01');
60
61 -- Insert products (with IGNORE to skip duplicates)
62 • INSERT IGNORE INTO products (name, category, price) VALUES
63     ('Laptop', 'Electronics', 999.99),
64     ('Smartphone', 'Electronics', 699.99),
65     ('Headphones', 'Accessories', 99.99),

```

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	customer_id	name	email	total_orders	total_spent	last_order_date
▶	1	John Doe	john@example.com	2	1119.97	2023-05-15
	2	Jane Smith	jane@example.com	1	699.99	2023-04-10
	3	Alice Brown	alice@example.com	0	0.00	NULL
	4	Bob Johnson	bob@example.com	1	124.95	2023-05-20

MONTHLY REVENUE GROWTH RATE

```
193      -- 9. Monthly revenue growth rate
194  • WITH monthly_revenue AS (
195      SELECT
196          DATE_FORMAT(order_date, '%Y-%m') AS month,
197          SUM(total_amount) AS revenue
198      FROM orders
199      GROUP BY month
200  )
201
202      -- 8. Find repeat customers (ordered more than once)
203  SELECT
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: 

	Tables_in_ecommerce
▶	customer_purchase_history
	customers
	order_items
	orders
	products









COMPLEX MONTHLY SALES REPORT

```
149 -- Query 7: Complex monthly sales report
150 • SELECT 'QUERY 7: Monthly sales report' AS description;
151 • SELECT
152     DATE_FORMAT(o.order_date, '%Y-%m') AS month,
153     c.customer_id,
154     c.name AS customer_name,
155     p.category,
156     COUNT(DISTINCT o.order_id) AS orders_count,
157     SUM(oi.quantity) AS items_sold,
158     SUM(oi.quantity * oi.price) AS monthly_revenue
159 FROM orders o
160 JOIN customers c ON o.customer_id = c.customer_id
161 JOIN order_items oi ON o.order_id = oi.order_id
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	month	customer_id	customer_name	category	orders_count	items_sold	monthly_revenue
▶	2023-04	1	John Doe	Electronics	1	1	999.99
	2023-04	2	Jane Smith	Electronics	1	1	699.99
	2023-04	1	John Doe	Accessories	1	1	99.99
	2023-05	4	Bob Johnson	Electronics	1	5	124.95
	2023-05	1	John Doe	Clothing	1	1	19.99


```
128 • SELECT 'QUERY 5: Creating customer_purchase_history view' AS description;
129 • DROP VIEW IF EXISTS customer_purchase_history;
130 • CREATE VIEW customer_purchase_history AS
131 SELECT
132     c.customer_id,
133     c.name,
134     c.email,
135     COUNT(o.order_id) AS total_orders,
136     COALESCE(SUM(o.total_amount), 0) AS total_spent,
137     MAX(o.order_date) AS last_order_date
138 FROM customers c
139 LEFT JOIN orders o ON c.customer_id = o.customer_id
```

Result Grid |   Filter Rows: | Edit:    | Export/Import:   | Wrap Cell Content: 

	product_id	name	price
▶	1	Laptop	999.99
	2	Smartphone	699.99
•	NULL	NULL	NULL

SUMMARY OF THIS QUERY:

- **TO SOLVE THIS PROBLEM USE MYSQL WORKBENCH.**
-
- **USE SELECT TO FIND MONTHLY SALES.**
-
- **USE COUNT ,MAX TO FIND PRICE.**
-
- **USE ORDER BY GROUP BY.**
-
- **USE JOIN TO FIND SALES REPORT.**

THANK YOU