MYSQL Workbench

Created Database & Tables

#	Time	Action	Message
1	13:13:55	CREATE DATABASE ecommerce	1 row(s) affected
2	13:13:56	USE ecommerce	0 row(s) affected
3	13:14:18	CREATE TABLE customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, customer_name VARCHAR(100), email VARCHAR(100...	0 row(s) affected
4	13:14:21	CREATE TABLE products (product_id INT PRIMARY KEY AUTO_INCREMENT, product_name VARCHAR(100), category VARCHAR(50), ...	0 row(s) affected
5	13:14:23	CREATE TABLE orders (order_id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, order_date DATE, total_amount DECIMAL...	0 row(s) affected
6	13:14:25	CREATE TABLE order_items (order_item_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, product_id INT, quantity INT, sub...	0 row(s) affected

Customer Table

customer_id	customer_name	email	country
1	Amit Patel	amit@gmail.com	India
2	Sarah Smith	sarah@gmail.com	USA
3	Mohammed Ali	ali@yahoo.com	UAE
NULL	NULL	NULL	NULL

Products Table

product_id	product_name	category	price
1	Laptop	Electronics	55000.00
2	Smartphone	Electronics	20000.00
3	Shoes	Fashion	3000.00
NULL	NULL	NULL	NULL

Order Table

order_id	customer_id	order_date	total_amount
1	1	2024-04-01	75000.00
2	2	2024-04-02	20000.00
NULL	NULL	NULL	NULL

Order_items Table

order_item_id	order_id	product_id	quantity	subtotal
1	1	1	1	55000.00
2	1	3	2	6000.00
3	2	2	1	20000.00
NULL	NULL	NULL	NULL	NULL

➔ Select, Where, Groupby, Orderby queries

```
-- Get all Indian customers sorted by name
SELECT * FROM customers WHERE country = 'India' ORDER BY customer_name;
```

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

customer_id	customer_name	email	country
1	Amit Patel	amit@gmail.com	India
NULL	NULL	NULL	NULL

MYSQL Workbench

```
-- Total orders and amount spent by each customer
SELECT customer_id, COUNT(order_id) AS total_orders, SUM(total_amount) AS total_spent
FROM orders GROUP BY customer_id;
```

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

customer_id	total_orders	total_spent
1	1	75000.00
2	1	20000.00

```
-- Monthly order summary with total revenue
SELECT
    DATE_FORMAT(order_date, '%Y-%m') AS order_month,
    COUNT(order_id) AS total_orders,
    SUM(total_amount) AS total_revenue
FROM orders GROUP BY order_month ORDER BY order_month;
```

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

order_month	total_orders	total_revenue
2024-04	2	95000.00

➔ JOINS (INNER, LEFT, RIGHT)

```
-- INNER JOIN: Orders with customer names
SELECT o.order_id, c.customer_name, o.order_date, o.total_amount
FROM orders o
INNER JOIN customers c ON o.customer_id = c.customer_id;
```

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

order_id	customer_name	order_date	total_amount
1	Amit Patel	2024-04-01	75000.00
2	Sarah Smith	2024-04-02	20000.00

MYSQL Workbench

```sql
-- LEFT JOIN: All customers with or without orders
SELECT c.customer_name, o.order_id
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id;
```

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

customer_name	order_id
Amit Patel	1
Sarah Smith	2
Mohammed Ali	NULL

```sql
-- RIGHT JOIN: All orders with customer names
SELECT o.order_id, c.customer_name
FROM orders o
RIGHT JOIN customers c ON o.customer_id = c.customer_id;
```

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

order_id	customer_name
1	Amit Patel
2	Sarah Smith
NULL	Mohammed Ali

➔ Subqueries

```sql
-- Customers who spent more than average total order amount
SELECT customer_name FROM customers
WHERE customer_id IN (
    SELECT customer_id
    FROM orders
    GROUP BY customer_id
    HAVING SUM(total_amount) > (
        SELECT AVG(total_amount) FROM orders
    )
);
```

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

customer_name
Amit Patel

MYSQL Workbench

```sql
-- Find customers who placed the highest value single order
SELECT customer_name
FROM customers
WHERE customer_id IN (
    SELECT customer_id
    FROM orders
    WHERE total_amount = (
        SELECT MAX(total_amount) FROM orders
    )
);
```

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

customer_name
Amit Patel

➔ Aggregate Functions (Sum, Average, Minimum, Maximum)

```sql
-- Average order amount per customer
SELECT customer_id, AVG(total_amount) AS avg_order_value
FROM orders
GROUP BY customer_id;
```

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

customer_id	avg_order_value
1	75000.000000
2	20000.000000

MYSQL Workbench

```
-- Total quantity sold per product
SELECT p.product_name, SUM(oi.quantity) AS total_quantity_sold
FROM order_items oi
JOIN products p ON oi.product_id = p.product_id
GROUP BY p.product_id;
```

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

product_name	total_quantity_sold
Laptop	1
Shoes	2
Smartphone	1

```
SELECT
    MAX(total_amount) AS highest_order,
    MIN(total_amount) AS lowest_order
FROM orders;
```

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

highest_order	lowest_order
75000.00	20000.00

➔ Views

```
CREATE VIEW customer_summary AS
SELECT c.customer_id, c.customer_name, COUNT(o.order_id) AS total_orders, SUM(o.total_amount) AS total_spent
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id;
```

```
select * from customer_summary;
```

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

customer_id	customer_name	total_orders	total_spent
1	Amit Patel	1	75000.00
2	Sarah Smith	1	20000.00
3	Mohammed Ali	0	NULL

customer_summary 16

MYSQL Workbench

➔ Indexes

```
-- Add index on customer_id in orders table
CREATE INDEX idx_customer_id_orders ON orders(customer_id);
SHOW INDEX FROM orders;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
orders	0	PRIMARY	1	order_id	A	2	NULL	NULL		BTREE			YES	NULL
orders	1	idx_customer_id_orders	1	customer_id	A	2	NULL	NULL	YES	BTREE			YES	NULL

```
-- Add index on order_id in order_items
CREATE INDEX idx_order_id_items ON order_items(order_id);
SHOW INDEX FROM order_items;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
order_items	0	PRIMARY	1	order_item_id	A	3	NULL	NULL		BTREE			YES	NULL
order_items	1	product_id	1	product_id	A	3	NULL	NULL	YES	BTREE			YES	NULL
order_items	1	idx_order_id_items	1	order_id	A	2	NULL	NULL	YES	BTREE			YES	NULL

➔ Case statement for conditional logic

```
SELECT
  customer_name,
  total_amount,
  CASE
    WHEN total_amount >= 1000 THEN 'High Value'
    WHEN total_amount >= 500 THEN 'Medium Value'
    ELSE 'Low Value'
  END AS order_category
FROM orders
JOIN customers ON orders.customer_id = customers.customer_id;
```

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

customer_name	total_amount	order_category
Amit Patel	75000.00	High Value
Sarah Smith	20000.00	High Value

MYSQL Workbench

➔ Date based analysis

```
SELECT
  DATE_FORMAT(order_date, '%M %Y') AS month,
  COUNT(order_id) AS total_orders
FROM orders
GROUP BY month;
```

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

month	total_orders
April 2024	2

➔ Ranking

```
SELECT
  customer_id, order_id, total_amount,
  RANK() OVER (PARTITION BY customer_id ORDER BY total_amount DESC) AS rank_in_customer
FROM orders;
```

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

customer_id	order_id	total_amount	rank_in_customer
1	1	75000.00	1
2	2	20000.00	1