VIEWS

1. **Standard View (Cust_Order_View)**

- This view provides a simplified way to retrieve customer order details, including customer ID, name, order ID, order date, and total price by joining the `customer` and `orders` tables.

— standard views

```
mysql> CREATE VIEW Cust_Order_View AS
-> SELECT c.custid,
c.name,o.orderid,o.order_date,o.total_price
-> FROM customer c
-> JOIN orders o ON c.custid = o.custid;
Query OK, 0 rows affected (0.04 sec)

mysql>
```

```
mysql> SELECT * FROM Cust_Order_View;
+--------+-------------+---------+------------+-------------+
| custid | name        | orderid | order_date | total_price |
+--------+-------------+---------+------------+-------------+
| C0001  | Amit Sharma | OR0001  | 2025-03-20 |    75580.00 |
| C0001  | Amit Sharma | OR0004  | 2025-07-08 |    74490.00 |
| C0001  | Amit Sharma | OR0009  | 2026-07-15 |   103950.00 |
| C0002  | Rahul Verma | OR0002  | 2025-04-10 |    25000.00 |
| C0002  | Rahul Verma | OR0007  | 2026-03-22 |   103130.00 |
| C0003  | Vikram Singh | OR0003  | 2025-05-18 |    84700.00 |
| C0003  | Vikram Singh | OR0008  | 2026-06-05 |   105380.00 |
| C0004  | Sneha Kapoor | OR0005  | 2025-09-25 |   110000.00 |
| C0004  | Sneha Kapoor | OR0010  | 2026-12-31 |    57800.00 |
| C0005  | Priya Mehta | OR0006  | 2025-11-12 |   108400.00 |
```

```
+--------+-------------+---------+------------+-------------+
```
10 rows in set (0.00 sec)

mysql>

---

2. **Read-Only View (Monthly_Sales)**

- This view aggregates order data on a monthly basis, showing the number of orders and total sales per month. Since MySQL views are read-only by default unless explicitly updatable, this view does not allow modifications.
- In MySQL, a view becomes **non-updatable (read-only)** if it contains:

1. **Aggregations** (`SUM()`, `COUNT()`, `AVG()`, etc.)
2. **GROUP BY** clause
3. **DISTINCT keyword**
4. **Joins involving multiple tables** (in most cases)

Since **Monthly_Sales** contains `COUNT(orderid)`, `SUM(total_price)`, and `GROUP BY OrderMonth`, MySQL does not allow updates, inserts, or deletions on this view, making it **read-only**.

—Read only view

```
mysql> CREATE VIEW Monthly_Sales AS
-> SELECT DATE_FORMAT(order_date, '%Y-%m') AS OrderMonth,
->        COUNT(orderid) AS OrderCount,
->        SUM(total_price) AS TotalSales
-> FROM orders
-> GROUP BY OrderMonth;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from Monthly_Sales;
+------------+------------+------------+
| OrderMonth | OrderCount | TotalSales |
+------------+------------+------------+
```

```
| 2025-03    |         1 |   75580.00 |
| 2025-04    |         1 |   25000.00 |
| 2025-05    |         1 |   84700.00 |
| 2025-07    |         1 |   74490.00 |
| 2025-09    |         1 |  110000.00 |
| 2025-11    |         1 |  108400.00 |
| 2026-03    |         1 |  103130.00 |
| 2026-06    |         1 |  105380.00 |
| 2026-07    |         1 |  103950.00 |
| 2026-12    |         1 |   57800.00 |
+------------+-----------+------------+
10 rows in set (0.00 sec)
```

Materialized View (Not Supported in MySQL)

- MySQL does not have built-in support for materialized views, which store query results physically for performance optimization. Instead, temporary tables or scheduled jobs (e.g., using triggers or stored procedures) can be used to achieve similar functionality.

MYSQL DOES NOT DIRECTLY SUPPORT MATERIALIZED VIEW

Difference Between Types of Views

Type of View	Stores Data	Updatable	Used for Aggregation	Performance Impact
Standard View	No	Yes (if simple)	No	Minimal
Complex View	No	No	Yes	Moderate
Read-Only View	No	No	Sometimes	Minimal
Updatable View	No	Yes	No	Minimal
Materialized View	Yes	Yes (on refresh)	Yes	High (Improves Query Speed)