# Solution: Using Partitioning and Bucketing in Hive

## 1. Partitioning

Partition the table based on sale\_date and country. This makes it easier to run queries like daily sales reports or sales filtered by country. Partitioning reduces the amount of data scanned when querying specific partitions.

* - Partition on: sale\_date, country

## 2. Bucketing

Use bucketing on product\_category and customer\_id. Bucketing divides data within partitions into smaller manageable files, which helps in efficiently querying specific product categories or customers.

* - Bucket on: product\_category, customer\_id

## 3. Create Table with Partitioning and Bucketing

Hive Query to create the table with partitioning and bucketing:  
CREATE TABLE sales\_data (  
 sale\_id INT,  
 product\_id INT,  
 product\_category STRING,  
 customer\_id INT,  
 sale\_amount FLOAT,  
 region STRING  
)  
PARTITIONED BY (sale\_date STRING, country STRING)  
CLUSTERED BY (product\_category, customer\_id) INTO 10 BUCKETS  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ','  
STORED AS ORC;

## 4. Load Data into Partitioned and Bucketed Table

When loading data, specify the partition values:  
LOAD DATA INPATH '/path/to/data/sales\_2023-08-01\_US.csv'  
INTO TABLE sales\_data  
PARTITION (sale\_date='2023-08-01', country='US');

## 5. Query Optimizations Using Partitioning and Bucketing

### a. Daily Sales by Country

To query daily sales for a specific country (e.g., USA) for a particular date:  
SELECT SUM(sale\_amount) AS total\_sales  
FROM sales\_data  
WHERE sale\_date = '2023-08-01' AND country = 'US';

### b. Top Products by Category in a Region

To find the top-selling products in a specific product category in a particular region:  
SELECT product\_id, SUM(sale\_amount) AS total\_sales  
FROM sales\_data  
WHERE product\_category = 'Electronics' AND region = 'North America'  
GROUP BY product\_id  
ORDER BY total\_sales DESC  
LIMIT 10;

### c. Customer Purchase Trends

To analyze purchase trends for a specific customer, say customer\_id = 12345, across different product categories:  
SELECT product\_category, SUM(sale\_amount) AS total\_spent  
FROM sales\_data  
WHERE customer\_id = 12345  
GROUP BY product\_category  
ORDER BY total\_spent DESC;

## 6. Advantages of Partitioning and Bucketing

- \*\*Partitioning\*\*: Limits the amount of data scanned, especially for time-based or country-based queries, speeding up reports like daily sales.  
- \*\*Bucketing\*\*: Divides large datasets into smaller, manageable chunks (buckets), improving performance when querying by product category or customer ID. This is ideal for targeted queries such as customer behavior analysis or product category trends.

# Conclusion

Partitioning and bucketing in Hive improve query performance by reducing the amount of data scanned and optimizing data storage. In this e-commerce use case, partitioning by sale\_date and country helps generate fast sales reports, while bucketing by product\_category and customer\_id speeds up targeted queries, such as top-selling products and customer segmentation.