Data Distribution Plan

1. Normalization:

Firstly, we performed Normalization as a part of data distribution strategy. To eliminate the redundancy, we broke down the order to Orders and Orders details. We are performing Horizontal, Vertical Fragmentation, Replication as a part of the database distribution plan.

2. Horizontal Fragmentation on Product categories:

1. **Data Localization:**
    - Horizontal fragmentation of product categories involves dividing the Products table based on the categories ('Electronics', 'Clothing and Fashion', 'Books').
    - This leads to data localization, where products of a specific category are stored together on a node.
    - It Improves query performance for category-specific searches or analytics.
2. **Reduced Data Transfer:**
    - Queries related to a particular product category can be executed on the node containing the relevant data, reducing the need for data transfer between nodes.
    - It Lowers the network overhead and faster query execution for category-specific operations.
3. **Isolation of Workloads:**
    - Different nodes can handle queries related to different product categories independently.
    - Which results in Isolation of workloads, preventing contention for resources when dealing with distinct categories.

3.Vertical Fragmentation on Warehouse Location:

1. **Reduced Data Volume:**
    - Vertical fragmentation involves splitting columns of a table and placing them on different nodes. For example, warehouse locations may be stored separately from other warehouse-related information.
    - It Reduces data volume on each node, which can lead to more efficient storage and retrieval.
2. **Improved Query Performance:**
    - Queries that involve only certain columns (e.g., WarehouseName, Location) can be executed on the node containing those columns, reducing the amount of data that needs to be accessed.
    - It Improves query performance for warehouse-related searches.
3. **Flexibility in Scaling:**

- Vertical fragmentation allows for scalability at the column level. You can scale specific columns independently based on their growth patterns or access frequency.
- It provides Flexible scaling to accommodate changes in data distribution patterns over time.

4.Replication:

1. Timestamped Replication:

- A mechanism has been established to timestamp database operations, allowing for the tracking of updates and changes.
- Each transaction within the primary database triggers the creation of a corresponding entry in the secondary database, capturing the state of the database at that specific moment.

2. Secondary Database Creation:

- The secondary database serves as a versioned snapshot of the primary database, maintaining a historical record of changes.
- This replication approach ensures that in the event of data loss or corruption, a specific version of the database can be easily restored from the secondary copy.

3. Fault Tolerance:

- The secondary database contributes to fault tolerance by providing a fallback option in case of primary database failures.
- In the event of a primary database outage, the secondary database can be seamlessly promoted to the primary role, minimizing downtime, and ensuring continuous operation.