

# 24 Lecture - CS403

## Important Subjective

### 1. **What is Vertical Partitioning in database design?**

Vertical partitioning is the process of splitting a table into smaller sub-tables based on columns. Each sub-table contains a subset of the original table columns.

#### **What are the benefits of vertical partitioning?**

Vertical partitioning can improve performance by reducing the amount of data read from disk, increasing cache efficiency, and reducing contention for table-level locks. It can also simplify queries by reducing the number of columns that need to be accessed.

#### **What are the drawbacks of vertical partitioning?**

Vertical partitioning can make it more difficult to perform certain queries that involve multiple sub-tables. It can also increase the complexity of the database schema and make it more difficult to maintain.

#### **How do you decide which columns to partition vertically?**

The decision of which columns to partition vertically depends on the access patterns of the application. Columns that are frequently accessed together should be placed in the same sub-table to improve performance.

#### **How do you implement vertical partitioning in a database?**

Vertical partitioning can be implemented using a variety of techniques, including partitioned views, table inheritance, or custom partitioning schemes implemented in the application layer.

#### **What is table inheritance in vertical partitioning?**

Table inheritance is a technique in which a set of related tables share a common set of columns, with each table containing additional columns that are specific to that table.

#### **How can you measure the performance impact of vertical partitioning?**

The performance impact of vertical partitioning can be measured by comparing the execution time of queries against the original table to the execution time of queries against the partitioned tables.

#### **How does vertical partitioning differ from horizontal partitioning?**

Vertical partitioning involves splitting a table into smaller sub-tables based on columns, while horizontal partitioning involves splitting a table into smaller sub-tables based on rows.

#### **What are some common use cases for vertical partitioning?**

Vertical partitioning is commonly used in databases with large tables that are frequently queried, such as transactional systems or data warehouses.

#### **What are some best practices for implementing vertical partitioning?**

Best practices for implementing vertical partitioning include analyzing the access patterns of the application, using a consistent partitioning strategy, and maintaining referential integrity across

partitioned tables.