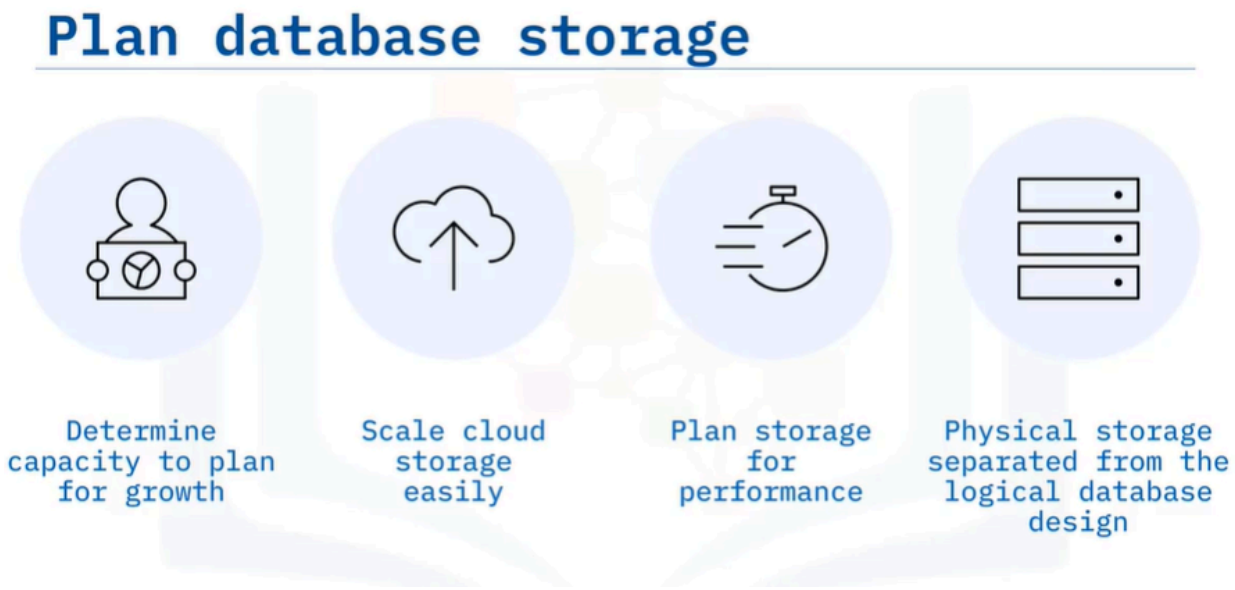
# Database Storage



Plan database storage

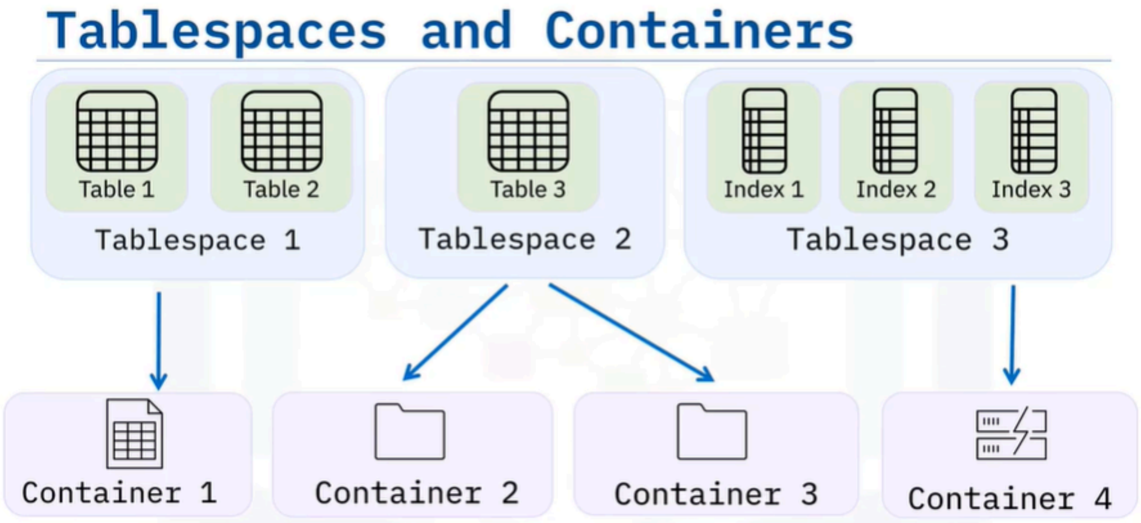
Determine capacity to plan for growth

Scale cloud storage easily

Plan storage for performance

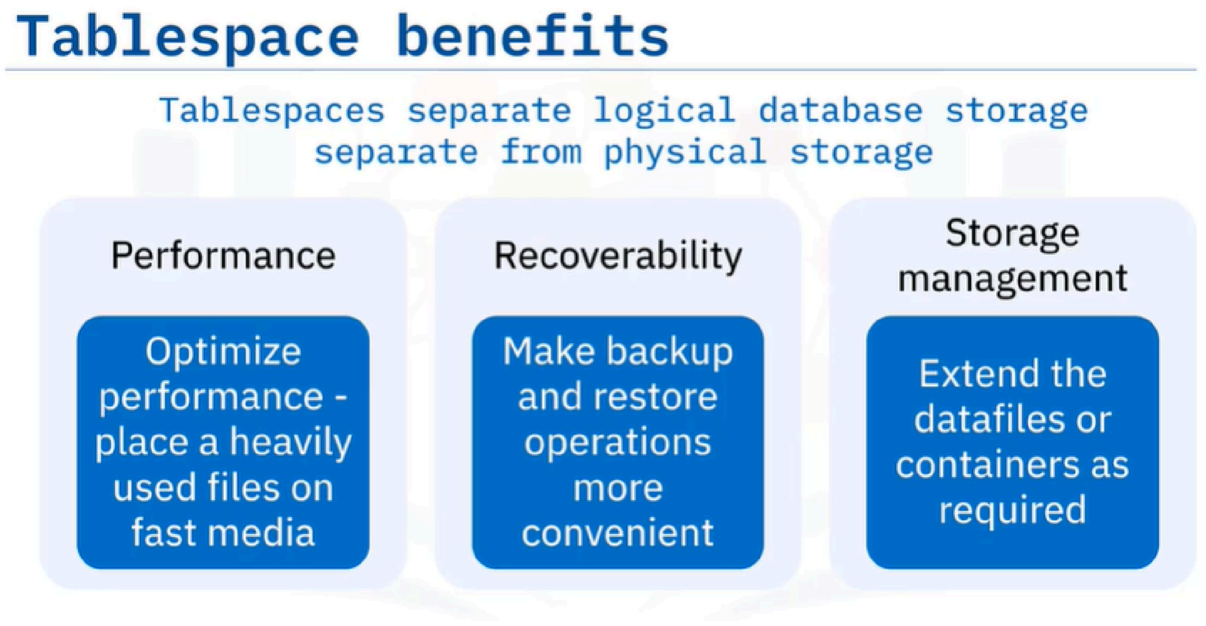
Physical storage separated from the logical database design

- As a DBA, you'll be responsible for ensuring that your database has enough storage space for all the data it needs to store. You must determine the capacity required for the database and plan for growth. In cloud-based databases expanding your storage space is performed through an API or a Graphical Console. One of the advantages of using a cloud-based database is that it is so easy to scale up and scale down in terms of storage space. In a self-managed or on-premises environment, you can also plan storage space for improved performance, storing competing resources on different disks. RDBMSes separate the physical storage of database files on disk from the logical design of the database, allowing more flexibility in managing the data files. You can manage the data through a logical object without being concerned about the nature of the physical storage. For example, you can issue a command to back up a database without having to specify all the physical disks that store the database.



**Storage Containers:** Files / Directories / Raw Devices

- Tablespaces are structures that contain database objects such as tables, indexes, large objects, and long data. DBAs use tablespaces to logically organize database objects based on where their data is stored. Tablespaces define the mapping between logical database objects and the physical storage containers that house the data for these objects. A storage container can be a data file, a directory, or a raw device. Tablespaces can contain one or more database objects.
- In this example, Tablespace 1 contains multiple small tables, whereas Tablespace 2 only houses a single large table. Tablespace 3 contains frequently used indexes. DBAs configure one or more storage containers to store each tablespace. In this example, Container 1 stores Tablespace 1, Containers 2 and 3 store Tablespace 2, and Container 4 stores Tablespace 4.



## Tablespace benefits

Tablespaces separate logical database storage separate from physical storage

Performance

Optimize performance - place a heavily used files on fast media

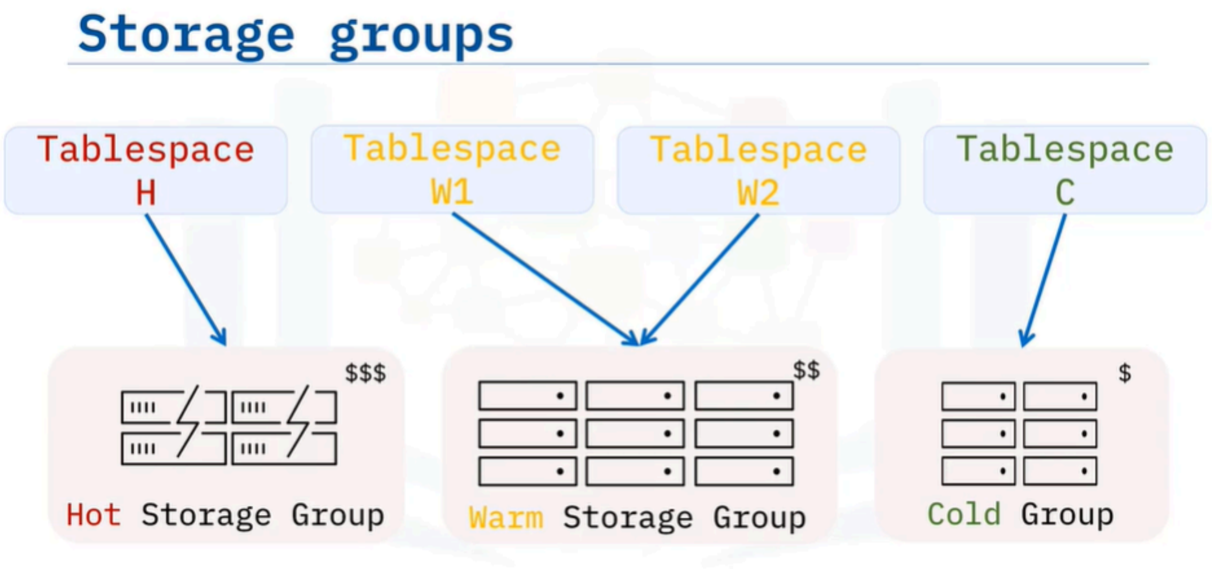
Recoverability

Make backup and restore operations more convenient

Storage management

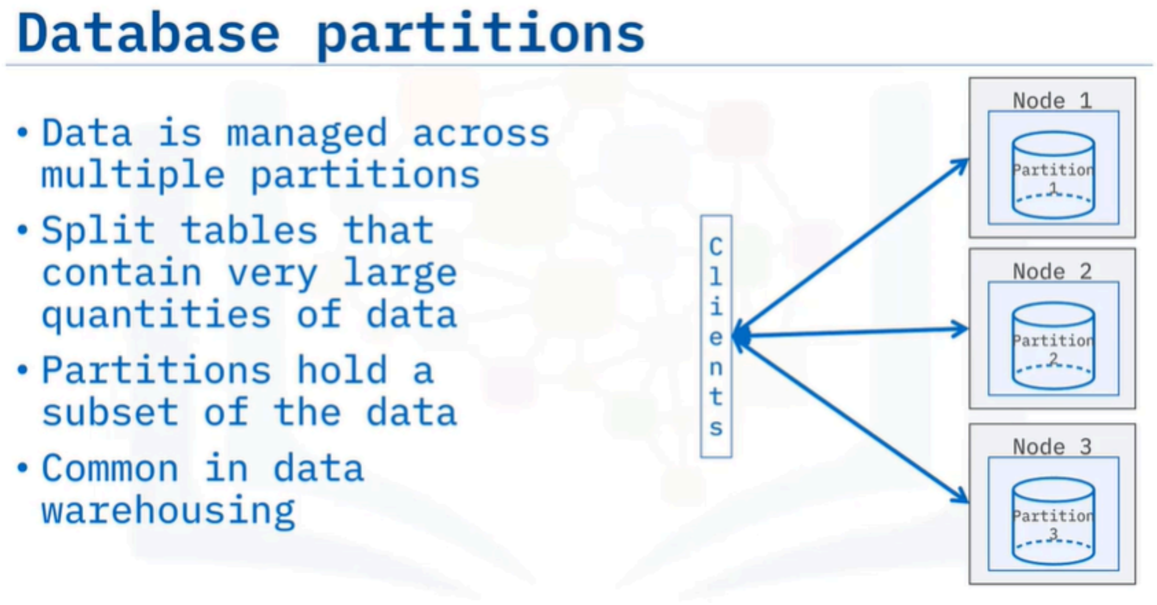
Extend the datafiles or containers as required

- By using the combination of tablespaces and containers, you can keep logical database storage separate from physical storage and manage the disk layout of a database and its objects. This results in several benefits: Performance: You can use tablespaces to optimize performance. For example, you can place a heavily used index on a fast SSD. Alternatively, you can store tables containing rarely accessed or archived data on a less expensive but slower magnetic hard drive. Recoverability: Tablespaces make backup and restore operations more convenient. Using a single command, you can make a backup or restore all the database objects without worrying about which storage container each object or tablespace is stored on. Storage Management: The RDBMS creates and extends the datafiles or containers depending on the need. When necessary, you can also manually expand the storage space by adding another storage path or container to the tablespace.



## Storage groups

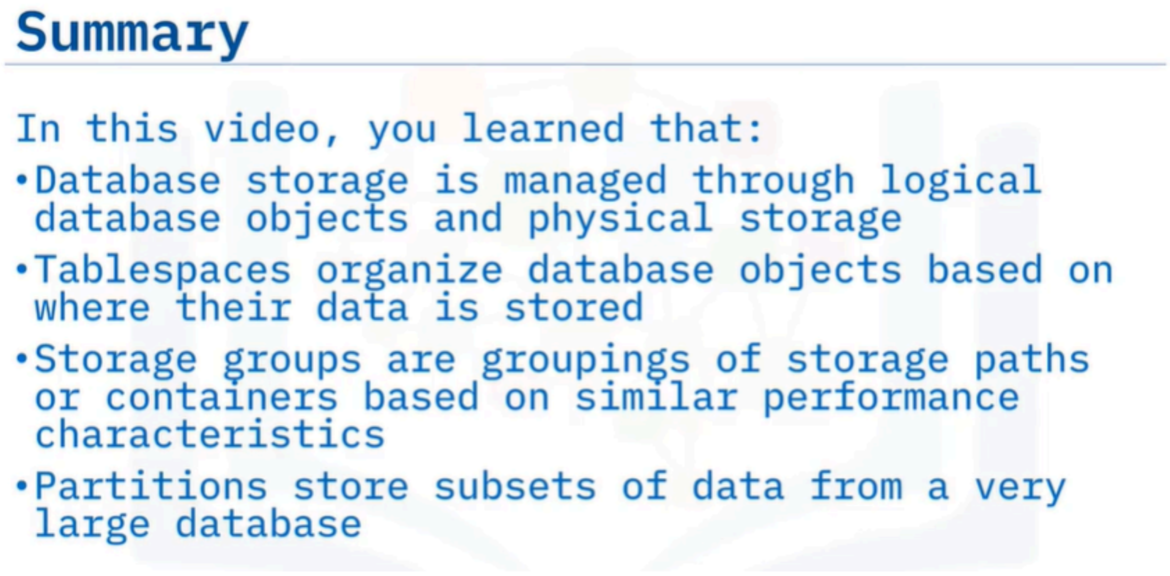
- Some RDBMSes provide Storage Groups. A storage group is a grouping of storage paths or containers based on similar performance characteristics. This allows you to perform Multi-Temperature Data Management more easily. In this context, temperature refers to the frequency of data access. Hot data is accessed very frequently, Warm data is accessed somewhat frequently, and Cold data is accessed infrequently. By using storage groups, you can organize your data and storage based on temperature. In this example, very frequently accessed hot tables can be placed in Tablespace H, which is distributed on a group of fast storage devices. Somewhat frequently accessed tables in Tablespaces W1 and W2 can be stored on a Warm Storage Group. And the least frequently accessed tables in Tablespace C can be stored on slower and less expensive storage devices in a Cold Group. Using storage groups helps with optimizing performance for frequently accessed data and reducing costs for storing infrequently accessed data.



## Database partitions

- Data is managed across multiple partitions
- Split tables that contain very large quantities of data
- Partitions hold a subset of the data
- Common in data warehousing

- A partitioned relational database is a relational database whose data is managed across multiple database partitions. You can partition tables that need to contain very large quantities of data into multiple logical partitions, with each partition containing a subset of the overall data. Database partitioning is used in scenarios that involve very large volumes of data, such as data warehousing and data analysis for business intelligence.



## Summary

In this video, you learned that:

- Database storage is managed through logical database objects and physical storage
- Tablespaces organize database objects based on where their data is stored
- Storage groups are groupings of storage paths or containers based on similar performance characteristics
- Partitions store subsets of data from a very large database