

12.4: Data Partitioning

Partitioning is a concept in databases in which very large tables and data are partitioned into smaller, individual tables, and queries which helps the data process more quickly and efficiently. One form of partitioning is called horizontal partitioning. Horizontal partitioning is the classification of the rows based on common characteristics into several, separate tables. For example, students with scores less than 70 might be stored in the FAIL table, while students who scored greater than 70 might be stored in the PASS table. The two partitioned tables would be FAIL and PASS. In doing so, the database is partitioned into two tables, both of which will be processed more quickly than a single table. The two typical methods of horizontal partitioning are to partition a sole column value and to partition the date (as the date helps organize the query chronologically). The keyword SELECT represents a horizontal partition in SQL. Below is an example of what horizontal partitioning would look like in SQL in which the first name, last name, class number and grade are specifically chosen as desired values:

```
SELECT viewClassGradeReport.[FirstName],  
viewClassGradeReport.[LastName],  
viewClassGradeReport.[ClassName],  
viewClassGradeReport.[Grade] FROM viewClassGradeReport;
```

The Oracle DBMS (Database Management System) provides support for multiple forms of partitioning. The first one is range partitioning. In range partitioning, each partitioned portion is characterized by a range of values for one or multiple columns, such as IDs or dates. For instance, rows for a column titled COUNTRY containing India, Sri Lanka, Pakistan, Nepal, and Bangladesh could be a partition for South Asian countries

The next form of partitioning is hash partitioning. Hash partitioning is the spreading of data in even partitions autonomous of the key value. Hash partitioning outperforms the uneven distributions or rows.

The next partitioning is known as list partitioning. List partitioning is a technique where a list of distinct values is defined as the partitioning key in the characterization for each partition.

The best use of list partitioning is when one requires specifically to highlight rows established on discrete values. For instance, a global distributor may only want data regarding deliveries to China and Japan while ignoring deliveries to other nations.

Oracle provides another form of partitioning known as composite partitioning. Composite partitioning is a combination of the general data allocation methods in which a table is partitioned using one allocation method, and then each partition is subdivided into smaller partitions through another general allocation method.

The counterpart of horizontal partitioning, vertical partitioning is another form of partitioning. Vertical partitioning, as opposed to horizontal partitioning, is the distribution of columns based on their commonalities and separating them into independent tables. Vertical partitioning helps identify the dynamic data from the stable, immutable data. Vertical partitioning is represented with the keyword PROJECT.

Partitioning is a useful tool that helps with the design of the database and has many advantages. Partitioning is practical and helps manage the table because partitioning helps identify the area where maintenance is needed and saves storage space. Partitioning is also secure, as only the relevant and necessary data can be specifically chosen and accessed by the user. Another benefit of implementing partitioning is that backing up and securing files is easier due to their smaller size, and if one file is corrupted, the other is still accessible. Partitioning works similarly with cars and replaceable parts: if one piece of equipment is damaged, the rest of the car will still be functioning. Partitioning also helps with balancing the load. The partitioned files can be designated to different storage locations, reducing conflict and maximizing performance.

Although partitioning proves to be quite handy, it does have its drawbacks. The first drawback of partitioning is the inconsistency of the access speed that it provides. All partitions are not identical; therefore, depending on the data the specific partition consists, access speeds will differ. Another drawback is complexity. Due to the complex nature of partitions, the code required to program will need to be more complex and challenging to maintain. The final disadvantage of partitioning is the excess storage space and time. Data can be replicated multiple times, which in turn takes up storage and can affect the time taken to process.

Partitioning is a useful tool that helps with the design of the database and has many advantages. Partitioning is practical and helps manage the table because partitioning helps identify the area where maintenance is needed and saves storage

space. Partitioning is also secure, as only the relevant and necessary data can be specifically chosen and accessed by the user. Another benefit of implementing partitioning is that backing up and securing files is easier due to their smaller size, and if one file is corrupted, the other is still accessible. Partitioning works similarly with cars and replaceable parts: if one piece of equipment is damaged, the rest of the car will still be functioning. Partitioning also helps with balancing the load. The partitioned files can be designated to different storage locations, reducing conflict and maximizing performance.

Although partitioning proves to be quite handy, it does have its drawbacks. The first drawback of partitioning is the inconsistency of the access speed that it provides. All partitions are not identical; therefore, depending on the data the specific partition consists, access speeds will differ. Another drawback is complexity. Due to the complex nature of partitions, the code required to program will need to be more complex and challenging to maintain. The final disadvantage of partitioning is the excess storage space and time. Data can be

Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN
3	SELDA	BAGCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

Vertical Partitions

VP1

CUSTOMER ID	FIRST NAME	LAST NAME
1	TAEKO	OHNUKI
2	O.V.	WRIGHT
3	SELDA	BAGCAN
4	JIM	PEPPER

VP2

CUSTOMER ID	FAVORITE COLOR
1	BLUE
2	GREEN
3	PURPLE
4	AUBERGINE

Horizontal Partitions

HP1

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN

HP2

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
3	SELDA	BAGCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

Figure 12.4.1:

Sharding, by Digital Ocean, Copyright 2020 by Digital Ocean

The user view, in a database, is a tool that helps visualize the tables in a database. Using the user view, physical tables can be partitioned coherently. The main intention of utilizing the user view is that it simplifies query editing and develops a secure database. In Oracle, a form of user view is offered called partition view, which displays physically partitioned tables that can be logically merged into one using the SQL union operator. This manner of partitioning does have its limitations. Firstly, there should not be any global index. Second, the physical tables should be independently handled. Lastly, fewer choices are at your disposal with partition view.

12.4: Data Partitioning is shared under a [not declared](#) license and was authored, remixed, and/or curated by LibreTexts.