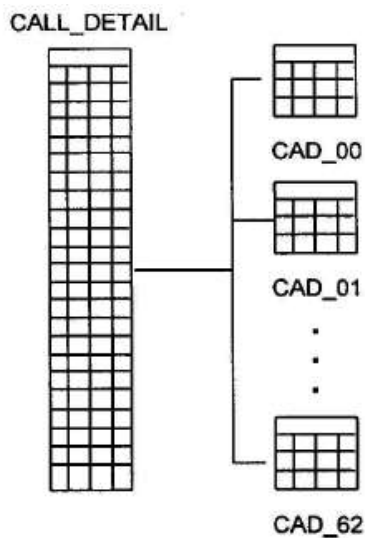


What Are Partitioned Tables?

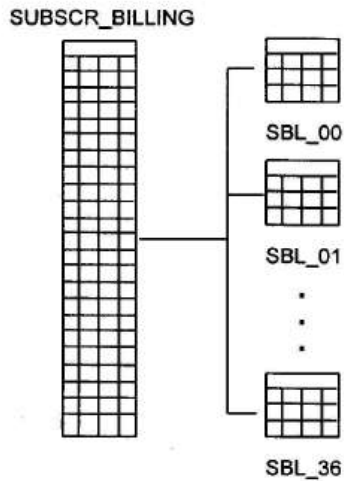
Partitioned tables are database tables that have the exact same structure, but contain different records. Partitioned tables logically segment the records that would otherwise be contained in one very, large table. Using smaller, partitioned tables makes database administration much easier. In addition, using partitioned tables increases response time for users.

In ABC, partitioned tables are used for the large fact tables. For example, the `CALL_DETAIL` table contains two months of call data, or approximately 90 million records. To make this table easier to manage and query, `CALL_DETAIL` is partitioned into 63 tables, each containing a day's worth of call records ($63 = 31$ maximum days in a month \times 2 months $+ 1$ for the current day's table). Each partitioned table is named `CAD_##`, where `##` is a number from 00 to 62.



CALL_DETAIL Partitioning

The SUBSCR_BILLING table is another fact table that is partitioned in ABC. SUBSCR_BILLING contains three years of subscriber billing records, or approximately 45 million records. To make this table easier to manage and query, SUBSCR_BILLING is partitioned into 37 tables, each containing a month's worth of billing records (12 months x 3 years + 1 for the current month's table). These partitioned tables are named SBL_##, where ## is a number from 00 to 36.

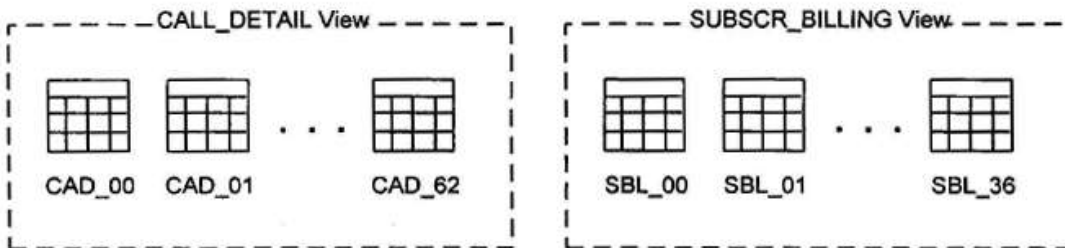


SUBSCR_BILLING Partitioning

How Are Views Used With Partitioned Tables?

An Oracle view is a custom-tailored presentation of the data from one or more database tables. Views enable the user to examine the data in arrangements different from the actual arrangements in the tables.

In ABC, views are used to group the data in a set of partitioned tables. For example, the CALL_DETAIL view groups the data from all the CAD tables. The SUBSCR_BILLING view groups the data from all the SBL tables.



Partitioned Tables and Views

How Are Aliases Used With Partitioned Tables?

Aliases, or synonyms, are simple, descriptive names for tables. Aliases make database administration much easier.

In ABC, aliases reflecting the date of the data are used for each of the partitioned tables. For example:

- CAD_JAN_02 is the alias for the table containing call records from January 2
- SBL_JAN is the alias for the table containing billing records from January

How Are Check Constraints Used With Partitioned Tables?

A constraint is an object in an Oracle database that defines a business rule for a column of a table. A check constraint is a specific type of constraint that only allows values in a column if those values satisfy the logical expression of the constraint. Check constraints improve performance by ensuring user queries go against only the relevant table in a view.

In ABC, there is a check constraint for each partitioned table. The format of this check constraint is:

partition_table_CK_alias

Where:

partition_table = the partitioned table that is constrained

alias = the alias of the partitioned table

These check constraints limit the value in the partitioned table's PERIOD_KEY column. This ensures that the partitioned table only contains data for a particular date.

For example:

- The check constraint CAD_00_CK_JAN_02 limits the data in PERIOD_KEY column to 1320 for the CAD_00 table with the alias CAD_JAN_02. In other words, the CAD_00 table can only contain call records for January 2.
- The check constraint SBL_00_CK_SBL_JAN limits the data in the PERIOD_KEY column to 1200 for the SBL_00 table with the alias SBL_JAN. In other words, the CAD_00 table can only contain billing records for January.

How are Partitioned Tables and their Constraints Maintained?

First, this section first provides a high-level description of the partition table maintenance process. Second, this section describes the PARTITION_INFO table. This table is located on the ABC database and is used during the maintenance process. Third, the maintenance process is detailed.

High-Level Description

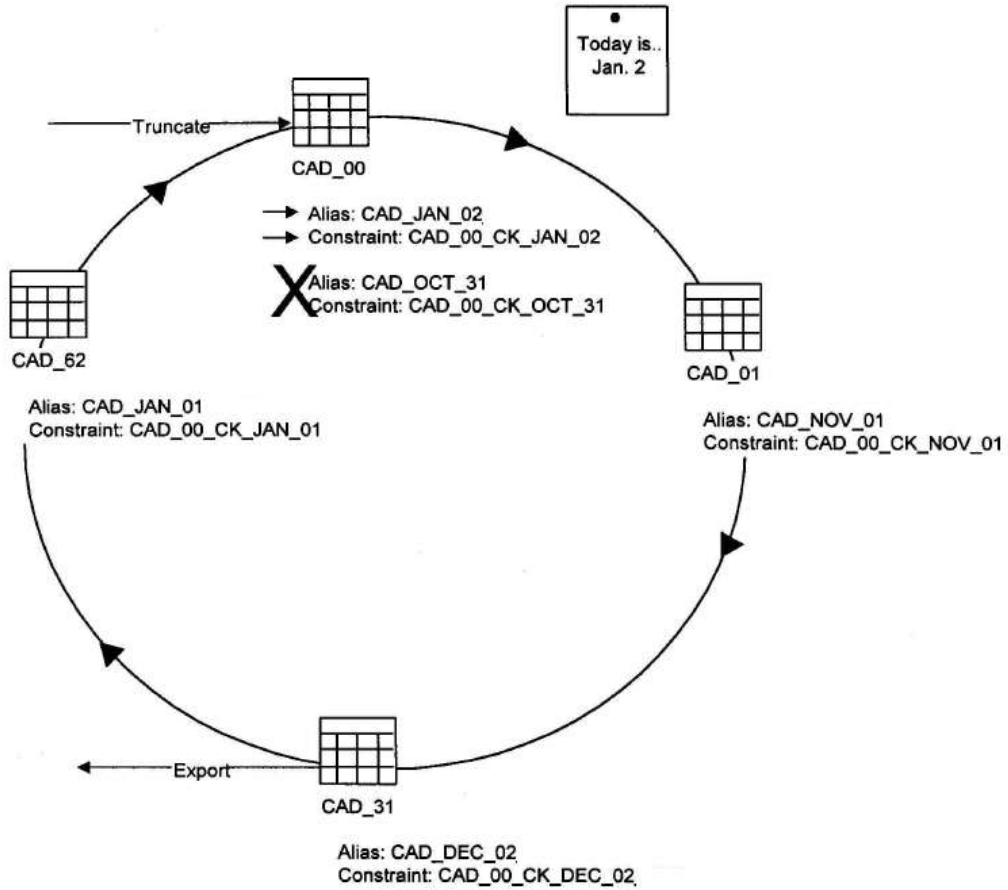
The ABC database always contains the same number of partitioned tables. However, the data these tables contain changes over time.

For example, the 63 CAD partitioned tables always contain call records for the past 63 days. So that the CAD tables always contain the most current 63 days of data, the following maintenance process is performed on the partitioned tables each night. This process is cyclical and new tables are maintained each night.

1. The data that is 31 days old is exported and backed up. Exporting data half way through the partitioned table cycle allows you time to troubleshoot any export problems.
2. The partitioned table containing data that is 64 days old is truncated. Truncating deletes the table's rows in the fastest possible way. Note that Steps 1 and 2 can run in parallel, in sequence, or be scheduled at different times.
3. Check constraints are created for the current date and associated with the partitioned table that was truncated.
4. The partitioned table that was truncated is loaded with data during the DEF to ABC Extract and Load process. For more information on this process, see the document titled "DEF to ABC Extract and Load Design." Note that check constraints can be created before or after the DEF to ABC Extract and Load process runs.
5. The partitioned table is analyzed.

The partitioned table maintenance process for the SBL tables is similar to the maintenance process for the CAD tables. The SBL tables, however, are maintained on a monthly basis instead of a daily basis.

The following figure illustrates the cyclical nature of the partitioned table maintenance process.



Partitioned Table Maintenance Process