105
DBMS_IMMUTABLE_TABLE

Immutable tables are read-only tables that protect data against unauthorized modification. They also prevent against accidental data modifications that may be caused by human errors. The `DBMS_IMMUTABLE_TABLE` package allows you to delete the expired rows in an immutable table and add interval partitioning.

This chapter contains the following topics:

- DBMS_IMMUTABLE_TABLE Overview
- DBMS_IMMUTABLE_TABLE Security Model
- Summary of DBMS_IMMUTABLE_TABLE Subprograms

> **See Also:**
>
> - *Oracle Database Administrator's Guide*
> - *Oracle Database SQL Language Reference*
> - *Oracle Database Reference*

DBMS_IMMUTABLE_TABLE Overview

Immutable tables are `read-only` tables that protect data against unauthorized modification. Immutable tables also prevent against accidental data modifications that may be caused by human errors. You must specify a retention period for the immutable table and for rows within the immutable table. An immutable table can be dropped if it contains no rows or if the specified retention period for the table has elapsed. Rows can be deleted from an immutable table only after the specified retention period for rows in the table has elapsed. Except for increasing the retention periods and adding interval partitioning, you cannot modify the definition of an immutable table with this package.

DBMS_IMMUTABLE_TABLE Security Model

The `DBMS_IMMUTABLE_TABLE` package is owned by `SYS` and is installed as part of database installation. The routines in the package are run with invoker's rights and hence run with the privileges of the current user.

Summary of DBMS_IMMUTABLE_TABLE Subprograms

This table lists the `DBMS_IMMUTABLE_TABLE` subprograms in alphabetical order and briefly describes them.

Table 105-1 DBMS_IMMUTABLE_TABLE Package Subprograms

Subprogram	Description
ADD_INTERVAL_PARTITIONING Procedure	This procedure adds interval partitioning to an existing, non-partitioned, V1 or V2 immutable table.
DELETE_EXPIRED_ROWS Procedure	This procedure deletes the expired rows.

ADD_INTERVAL_PARTITIONING Procedure

This procedure adds interval partitioning to an existing, non-partitioned, V1 or V2 immutable table.

Syntax

```
DBMS_IMMUTABLE_TABLE.ADD_INTERVAL_PARTITIONING(
    schema_name               IN    VARCHAR2,
    table_name                IN    VARCHAR2,
    interval_number           IN    NUMBER,
    interval_frequency        IN    VARCHAR2,
    first_high_timestamp      IN    TIMESTAMP);
```

Parameters

Table 105-2 ADD_INTERVAL_PARTITIONING Parameters

Parameter	Description
`schema_name`	The name of the schema.
`table_name`	The name of the immutable table.
`interval_number`	Sets how often the database creates partitions for the immutable table.
`interval_frequency`	Sets the frequency for the value that was set in the `interval_number` setting. Supported values are `YEAR`, `MONTH`, `DAY`, `HOUR`, and `MINUTE`.
`first_high_timestamp`	A timestamp that determines the upper boundary of the first partition in the immutable table.

Usage Notes

- Composite partitioning (that is, sub-partitioning) is not supported with the above interval partitioning.

DELETE_EXPIRED_ROWS Procedure

This procedure deletes some or all of the expired rows from the immutable table. This procedure commits before deleting any expired rows and commits after deleting any expired rows.

Syntax

```
DBMS_IMMUTABLE_TABLE.DELETE_EXPIRED_ROWS(
   schema_name                 IN VARCHAR2,
   table_name                  IN VARCHAR2,
```

```
  before_timestamp             IN TIMESTAMP WITH TIME ZONE DEFAULT NULL,
  number_of_rows_deleted       OUT NUMBER);
```

Parameters

Table 105-3 DELETE_EXPIRED_ROWS Procedure Parameters

Parameter	Description
schema_name	The name of the schema.
table_name	The name of the immutable table.
before_timestamp	If the parameter is NULL, all expired rows in the table are deleted. If the parameter is not NULL and older than the timestamp calculated based on current time and row retention time, rows with timestamps less than the parameter value are deleted. If the parameter is younger than the timestamp calculated based on the current time and row retention time, the calculated timestamp is used, and all expired rows are deleted. The default value is NULL.
number_of_rows_deleted	The number of rows deleted.