16

SQL Statements: DROP CONTEXT to DROP JAVA

This chapter contains the following SQL statements:

DROP CONTEXT

Purpose

Use the `DROP CONTEXT` statement to remove a context namespace from the database.

Removing a context namespace does not invalidate any context under that namespace that has been set for a user session. However, the context will be invalid when the user next attempts to set that context.

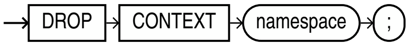
> **See Also:**
>
> CREATE CONTEXT and *Oracle Database Security Guide* for more information on contexts

Prerequisites

You must have the `DROP ANY CONTEXT` system privilege.

Syntax

***drop_context*::=**



Semantics

namespace

Specify the name of the context namespace to drop. You cannot drop the built-in namespace USERENV.

> **See Also:**
>
> SYS_CONTEXT for information on the USERENV namespace

Examples

Dropping an Application Context: Example

The following statement drops the context created in CREATE CONTEXT:

```
DROP CONTEXT hr_context;
```

DROP DATABASE

Purpose

> **Note:**
>
> You cannot roll back a DROP DATABASE statement.

Use the DROP DATABASE statement to drop the database. This statement is useful when you want to drop a test database or drop an old database after successful migration to a new host.

You can issue DROP DATABASE on True Cache to delete all the files that belong to this True Cache. The command only deletes files that belong to this True Cache. You must have started up the True Cache in mount mode.

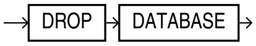
> **See Also:**
>
> *Oracle Database Backup and Recovery User's Guide* for more information on dropping the database

Prerequisites

You must have the SYSDBA system privilege to issue this statement. The database must be mounted in exclusive and restricted mode, and it must be closed.

Syntax

***drop_database*::=**



Semantics

When you issue this statement, Oracle Database drops the database and deletes all control files and data files listed in the control file. If the database used a server parameter file (spfile), then the spfile is also deleted.

Archived logs and backups are not removed, but you can use Recovery Manager (RMAN) to remove them. If the database is on raw disks, then this statement does not delete the actual raw disk special files.

Drop True Cache

To drop the True Cache using DROP DATABASE you must mount the database and set restricted mode as follows:

```
STARTUP MOUNT
ALTER SYSTEM ENABLE RESTRICTED SESSION;
DROP DATABASE
```

DROP DATABASE LINK

Purpose

Use the DROP DATABASE LINK statement to remove a database link from the database.

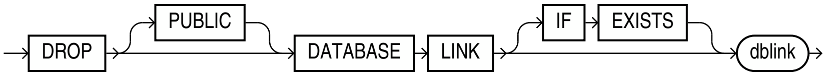
> **See Also:**
>
> CREATE DATABASE LINK for information on creating database links

Prerequisites

A private database link must be in your own schema. To drop a PUBLIC database link, you must have the DROP PUBLIC DATABASE LINK system privilege.

Syntax

***drop_database_link*::=**



Semantics

PUBLIC

You must specify PUBLIC to drop a PUBLIC database link.

dblink

Specify the name of the database link to be dropped.

Restriction on Dropping Database Links

You cannot drop a database link in another user's schema, and you cannot qualify *dblink* with the name of a schema, because periods are permitted in names of database links. Therefore, Oracle Database interprets the entire name, such as ralph.linktosales, as the name of a database link in your schema rather than as a database link named linktosales in the schema ralph.

IF EXISTS

Specify IF EXISTS to drop an existing object.

Specifying IF NOT EXISTS with DROP results in ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement.

Examples

Dropping a Database Link: Example

The following statement drops the public database link named remote, which was created in "Defining a Public Database Link: Example":

```
DROP PUBLIC DATABASE LINK remote;
```

DROP DIMENSION

Purpose

Use the DROP DIMENSION statement to remove the named dimension.

This statement does not invalidate materialized views that use relationships specified in dimensions. However, requests that have been rewritten by query rewrite may be invalidated, and subsequent operations on such views may execute more slowly.

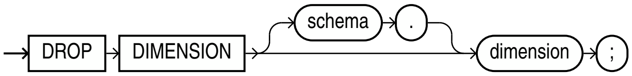
> **See Also:**
>
> - CREATE DIMENSION and ALTER DIMENSION for information on creating and modifying a dimension
> - *Oracle Database Concepts* for general information about dimensions

Prerequisites

The dimension must be in your own schema or you must have the DROP ANY DIMENSION system privilege to use this statement.

Syntax

***drop_dimension*::=**



Semantics

schema

Specify the name of the schema in which the dimension is located. If you omit *`schema`*, then Oracle Database assumes the dimension is in your own schema.

dimension

Specify the name of the dimension you want to drop. The dimension must already exist.

Examples

Dropping a Dimension: Example

This example drops the `sh.customers_dim` dimension:

```
DROP DIMENSION customers_dim;
```

> **See Also:**
>
> "Creating a Dimension: Examples" and "Modifying a Dimension: Examples" for examples of creating and modifying this dimension

DROP DIRECTORY

Purpose

Use the `DROP DIRECTORY` statement to remove a directory object from the database.

> **See Also:**
>
> CREATE DIRECTORY for information on creating a directory

Prerequisites

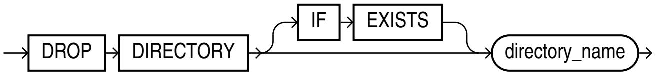
To drop a directory, you must have the `DROP ANY DIRECTORY` system privilege.

Note:

Do not drop a directory when files in the associated file system are being accessed by PL/SQL or OCI programs.

Syntax

***drop_directory*::=**



Semantics

IF EXISTS

Specify `IF EXISTS` to drop an existing object.

Specifying `IF NOT EXISTS` with `DROP` results in `ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement`.

directory_name

Specify the name of the directory database object to be dropped.

Oracle Database removes the directory object but does not delete the associated operating system directory on the server file system.

Examples

Dropping a Directory: Example

The following statement drops the directory object `bfile_dir`:

```
DROP DIRECTORY bfile_dir;
```

See Also:

"Creating a Directory: Examples"

DROP DISKGROUP

> **Note:**
>
> This SQL statement is valid only if you are using Oracle ASM and you have started an Oracle ASM instance. You must issue this statement from within the Oracle ASM instance, not from a normal database instance. For information on starting an Oracle ASM instance, refer to *Oracle Automatic Storage Management Administrator's Guide*.

Purpose

The DROP DISKGROUP statement lets you drop an Oracle ASM disk group along with all the files in the disk group. Oracle ASM first ensures that no files in the disk group are open. It then drops the disk group and all its member disks and clears the disk header.

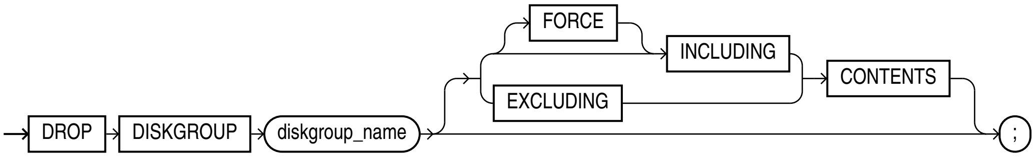
> **See Also:**
>
> - CREATE DISKGROUP and ALTER DISKGROUP for information on creating and modifying disk groups
> - *Oracle Automatic Storage Management Administrator's Guide* for information on Oracle ASM and using disks groups to simplify database administration

Prerequisites

You must have the SYSASM system privilege and you must have an Oracle ASM instance started, from which you issue this statement. The disk group to be dropped must be mounted.

Syntax

***drop_diskgroup*::=**



Semantics

diskgroup_name

Specify the name of the disk group you want to drop.

INCLUDING CONTENTS

Specify INCLUDING CONTENTS to confirm that Oracle ASM should drop all the files in the disk group. You must specify this clause if the disk group contains any files.

FORCE

This clause clears the headers on the disk belonging to a disk group that cannot be mounted by the Oracle ASM instance. The disk group cannot be mounted by any instance of the database.

The Oracle ASM instance first determines whether the disk group is being used by any other Oracle ASM instance using the same storage subsystem. If it is being used, and if the disk group is in the same cluster, or on the same node, then the statement fails. If the disk group is in a different cluster, then the system further checks to determine whether the disk group is mounted by any instance in the other cluster. If it is mounted elsewhere, then the statement fails. However, this latter check is not as definitive as the checks for disk groups in the same cluster. Therefore, use this clause with caution.

EXCLUDING CONTENTS

Specify `EXCLUDING CONTENTS` to ensure that Oracle ASM drops the disk group only when the disk group is empty. This is the default. If the disk group is not empty, then an error will be returned.

Examples

Dropping a Diskgroup: Example

The following statement drops the Oracle ASM disk group `dgroup_01`, which was created in "Creating a Diskgroup: Example", and all of the files in the disk group:

```
DROP DISKGROUP dgroup_01 INCLUDING CONTENTS;
```

DROP DOMAIN

Purpose

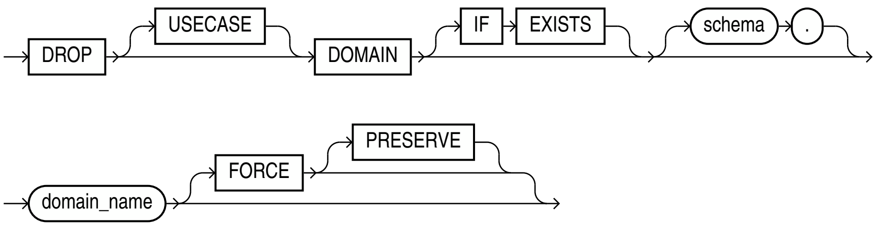
Use this statement to drop a domain, thereby disassociating the domain from all its dependent objects.

Prerequisites

The domain must be in your own schema or you must have the `DROP ANY DOMAIN` system privilege.

Syntax

***drop_domain*::=**



Semantics

USECASE

This keyword is optional and is provided for semantic clarity. It indicates that the domain is to describe a data use case.

IF EXISTS

Specify `IF EXISTS` to drop a domain that exists.

You cannot specify `IF NOT EXISTS` with `DROP DOMAIN`. This results in the following error: `ORA-11544:Incorrect IF EXISTS clause for ALTER/DROP statement`.

You can drop a domain by specifying its domain name and disassociate it from all its dependent columns. In the following example `email` is the name of a domain name:

```
DROP DOMAIN email;
```

If a domain is associated with columns, `DROP DOMAIN` *`domain_name`* fails with `ORA-11502:The domain <domain_name> to be dropped has dependent objects.`This includes any tables in the recyclebin.

You can additionally drop a domain by specifying two optional keywords: either `FORCE` or `FORCE PRESERVE` which have different meanings:

FORCE

`DROP DOMAIN` *`domain_name`* `FORCE` disassociates the domain from all its dependent columns. This includes dropping all constraints on columns that were inherited from the domain. Defaults inherited from the domain are also dropped unless these defaults were set specifically on columns.

If you must drop a domain and have dependent tables in the recyclebin, you must use the `FORCE` option.

FORCE PRESERVE

Use `DROP DOMAIN` *`domain_name`* `FORCE PRESERVE` if you want to preserve domain defaults and domain constraints on columns inherited from the domain. You must first specify `FORCE`, in order to specify the `PRESERVE` option. Use this option if you want to temporarily drop the domain and recreate it later with new data and the former dependent columns. In this case you want to ensure that the table data continues to be consistent with the domain definition with all the constraints and defaults on columns inherited from the domain preserved. If you drop a domain with `FORCE PRESERVE` and later recreate the domain and reassociate the column with it, you can end up with a second constraint. In this case, use `ALTER TABLE DROP CONSTRAINT` to drop the second constraint.

If there are no tables or materialized views with columns of the given domain, then `DROP DOMAIN` will invalidate any SQL dependent statements and remove the domain object from the catalog. In this case you can specify `FORCE` and `FORCE PRESERVE` without affecting the domain or the domain's dependent objects.

If there are tables or materialized views with columns of the given domain, `DROP DOMAIN` without the `FORCE` option will fail without affecting the domain or domain's dependent objects.

If there are tables or materialized views with columns of the given domain, `DROP DOMAIN FORCE` will do the following:

- Remove the default expression from any dependent column, if the column has a default expression that was only set as a domain default. If the default expression was added on the column and set as domain default, then default on the column is preserved.
- Remove the domain annotations from all dependent columns

- Preserve collation on any domain dependent columns
- Invalidate all SQL dependent statements in the cursor cache.
- Materialized views (MVs) that reference domain functions like DOMAIN_DISPLAY, DOMAIN_ORDER, DOMAIN_NAME will be invalidated so they can be fully refreshed. MVs that reference columns of a given domain will not be invalidated .
- Remove the domain successfully.

If there are flexible domains referencing the domain, then DROP DOMAIN without the FORCE option will raise an error, while DROP DOMAIN FORCE will drop all flexible dependent domains in FORCE mode also.

Examples

The following example drops the domain day_of_week. If there are any columns associated with the domain the statement will raise ORA-11502 and the domain will still be present:

```
DROP DOMAIN day_of_week;
```

The following statement drops the domain day_of_week. If there are any columns associated with it, the columns will inherit any defaults and constraints from the domain:

```
DROP DOMAIN day_of_week FORCE PRESERVE;
```

DROP EDITION

Purpose

Use the DROP EDITION statement to drop an edition, along with all actual editionable objects it contains. An actual editionable object is an editionable object that has been created or modified in an edition.

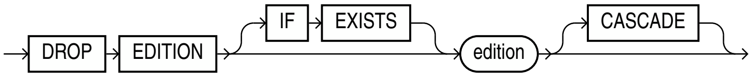
> **See Also:**
>
> CREATE EDITION for a listing of editionable object types

Prerequisites

You must have the DROP ANY EDITION system privilege, granted either directly or through a role.

Syntax

***drop_edition*::=**



Semantics

IF EXISTS

Specify IF EXISTS to drop an existing edition.

Specifying `IF NOT EXISTS` with `DROP` results in `ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement`.

Objects that are not editionable, or that are editionable but have not been actualized in the current edition, are not dropped.

You must specify `CASCADE` if the specified edition contains any actual editionable objects.

Restrictions

This statement is subject to the following conditions and restrictions:

- The specified edition cannot have both a parent edition and a child edition.
- `DROP EDITION` will fail if you attempt to drop the default edition.
- `DROP EDITION` will fail if you attempt to drop the root edition and the recycle bin contains at least one object that used to be in that edition before it was dropped. Under these circumstances, even `DROP EDITION CASCADE` will fail. In this case, you can purge all objects from the recycle bin with the `PURGE DBA_RECYCLEBIN` statement and then drop the edition. Refer to PURGE for more information.

  `DROP EDITION` will also fail if you attempt to drop the leaf edition and the recycle bin contains at least one object that used to be in that edition before it was dropped. However, under these circumstances, `DROP EDITION CASCADE` will succeed.

  The only type of editioned object that might be in the recycle bin is a trigger.

> **See Also:**
>
> - *Oracle Database Development Guide*
> - *Oracle Database PL/SQL Packages and Types Reference*

Examples

For examples that use this statement, refer to CREATE EDITION .

DROP FLASHBACK ARCHIVE

Purpose

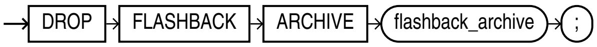
Use the `DROP FLASHBACK ARCHIVE` clause to remove a flashback archive from the system. This statement removes the flashback archive and all the historical data in it, but does not drop the tablespaces that were used by the flashback archive.

Prerequisites

You must have the `FLASHBACK ARCHIVE ADMINISTER` system privilege to drop a flashback archive.

Syntax

***drop_flashback_archive*::=**



Semantics

flashback_archive

Specify the name of the flashback archive you want to drop.

> **See Also:**
>
> CREATE FLASHBACK ARCHIVE for information on creating flashback archives and for some simple examples of using flashback archives

DROP FUNCTION

Purpose

Functions are defined using PL/SQL. Refer to *Oracle Database PL/SQL Language Reference* for complete information on creating, altering, and dropping functions.

Use the `DROP FUNCTION` statement to remove a standalone stored function from the database.

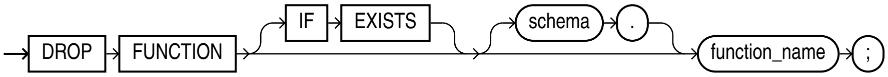
> **Note:**
>
> Do not use this statement to remove a function that is part of a package. Instead, either drop the entire package using the `DROP PACKAGE` statement or redefine the package without the function using the `CREATE PACKAGE` statement with the `OR REPLACE` clause.

Prerequisites

The function must be in your own schema or you must have the `DROP ANY PROCEDURE` system privilege.

Syntax

***drop_function*::=**



Semantics

IF EXISTS

Specify `IF EXISTS` to drop an existing object.

Specifying `IF NOT EXISTS` with `DROP` results in `ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement`.

schema

Specify the schema containing the function. If you omit `schema`, then Oracle Database assumes the function is in your own schema.

function_name

Specify the name of the function to be dropped.

Oracle Database invalidates any local objects that depend on, or call, the dropped function. If you subsequently reference one of these objects, then the database tries to recompile the object and returns an error if you have not re-created the dropped function.

If any statistics types are associated with the function, then the database disassociates the statistics types with the `FORCE` option and drops any user-defined statistics collected with the statistics type.

> **See Also:**
>
> - *Oracle Database Concepts* for more information on how Oracle Database maintains dependencies among schema objects, including remote objects
> - ASSOCIATE STATISTICS and DISASSOCIATE STATISTICS for more information on statistics type associations

Examples

Dropping a Function: Example

The following statement drops the function `SecondMax` in the sample schema `oe` and invalidates all objects that depend upon `SecondMax`:

```
DROP FUNCTION oe.SecondMax;
```

> **See Also:**
>
> *Oracle Database PL/SQL Language Reference* for the example that creates the `SecondMax` function

DROP HIERARCHY

Purpose

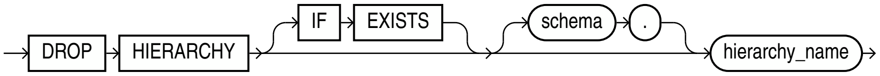
Use the `DROP HIERARCHY` statement to drop a hierarchy. A `HIERARCHY` object is a component of analytic views.

Prerequisites

To drop a hierarchy in your own schema, you must have the `DROP HIERARCHY` system privilege. To drop a hierarchy in another user's schema, you must have the `DROP ANY HIERARCHY` system privilege.

Syntax

***drop_hierarchy*::=**



Semantics

IF EXISTS

Specify `IF EXISTS` to drop an existing object.

Specifying `IF NOT EXISTS` with `DROP` results in `ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement`.

schema

Specify the schema in which the hierarchy exists. If you do not specify a schema, then Oracle Database looks for the hierarchy in your own schema.

hierarchy_name

Specify the name of the hierarchy to drop.

Example

The following statement drops the specified hierarchy object:

```
DROP HIERARCHY product_hier;
```

DROP INDEX

Purpose

Use the `DROP INDEX` statement to remove an index or domain index from the database.

When you drop a global partitioned index, a range-partitioned index, or a hash-partitioned index, all the index partitions are also dropped. If you drop a composite-partitioned index, then all the index partitions and subpartitions are also dropped.

In addition, when you drop a domain index:

- Oracle Database invokes the appropriate routine.
- If any statistics are associated with the domain index, then Oracle Database disassociates the statistics types with the `FORCE` clause and removes the user-defined statistics collected with the statistics type.

See Also:

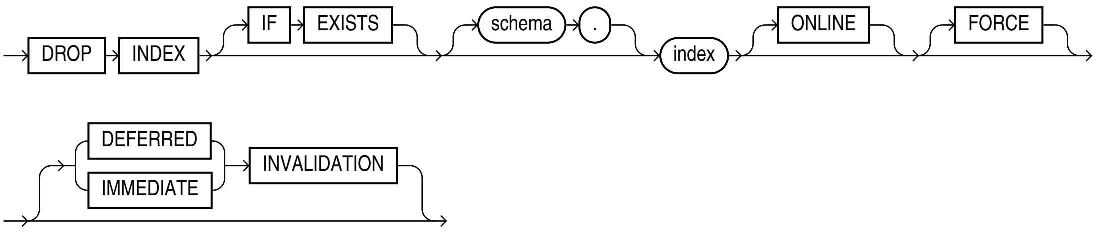
- *Oracle Database Data Cartridge Developer's Guide* for information on the routines
- CREATE INDEX and ALTER INDEX for information on creating and modifying an index
- The `domain_index_clause` of CREATE INDEX for more information on domain indexes
- ASSOCIATE STATISTICS and DISASSOCIATE STATISTICS for more information on statistics type associations

Prerequisites

The index must be in your own schema or you must have the `DROP ANY INDEX` system privilege.

Syntax

drop_index::=



Semantics

IF EXISTS

Specify `IF EXISTS` to drop an existing object.

Specifying `IF NOT EXISTS` with `DROP` results in `ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement`.

schema

Specify the schema containing the index. If you omit *`schema`*, then Oracle Database assumes the index is in your own schema.

index

Specify the name of the index to be dropped. When the index is dropped, all data blocks allocated to the index are returned to the tablespace that contained the index.

ONLINE

Specify `ONLINE` to indicate that DML operations on the table or partition will be allowed while dropping the index.

FORCE

FORCE applies only to domain indexes. This clause drops the domain index even if the indextype routine invocation returns an error or the index is marked IN PROGRESS. Without FORCE, you cannot drop a domain index if its indextype routine invocation returns an error or the index is marked IN PROGRESS.

> **Note:**
>
> When dropping a domain index with FORCE option, the index will be dropped regardless of any errors happening in the indextype routine. The errors raised by the indextype routine are not reported.
>
> Only use the FORCE option when the index or index partitions are marked IN PROGRESS or when DROP INDEX has already failed.

{ DEFERRED | IMMEDIATE } INVALIDATION

This clause lets you control when the database invalidates dependent cursors while dropping the index. It has the same semantics here as for the ALTER INDEX statement, with the following addition: When you drop an index with DEFERRED INVALIDATION, Oracle database will immediately invalidate any DML statement or query that references the dropped index in its plan.

See { DEFERRED | IMMEDIATE } INVALIDATION in the documentation on ALTER INDEX for the full semantics of this clause.

Restrictions on Dropping Indexes

The following restrictions apply to dropping indexes:

- You cannot drop a domain index if the index or any of its index partitions is marked IN_PROGRESS.
- You cannot specify the ONLINE clause when dropping a domain index, a cluster index, or an index on a queue table.

Examples

Dropping an Index: Example

This statement drops an index named ord_customer_ix_demo, which was created in "Compressing an Index: Example":

```
DROP INDEX ord_customer_ix_demo;
```

DROP INDEXTYPE

Purpose

Use the DROP INDEXTYPE statement to drop an indextype as well as any association with a statistics type.

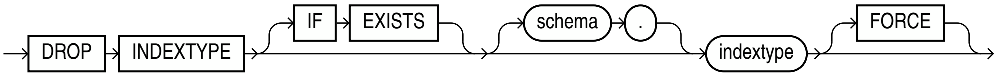
> **See Also:**
>
> CREATE INDEXTYPE for more information on indextypes

Prerequisites

The indextype must be in your own schema or you must have the `DROP ANY INDEXTYPE` system privilege.

Syntax

***drop_indextype*::=**



Semantics

IF EXISTS

Specify `IF EXISTS` to drop an existing object.

Specifying `IF NOT EXISTS` with `DROP` results in `ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement`.

schema

Specify the schema containing the indextype. If you omit *`schema`*, then Oracle Database assumes the indextype is in your own schema.

indextype

Specify the name of the indextype to be dropped.

If any statistics types have been associated with indextype, then the database disassociates the statistics type from the indextype and drops any statistics that have been collected using the statistics type.

> **See Also:**
>
> ASSOCIATE STATISTICS and DISASSOCIATE STATISTICS for more information on statistics associations

FORCE

Specify `FORCE` to drop the indextype even if the indextype is currently being referenced by one or more domain indexes. Oracle Database marks those domain indexes `INVALID`. Without `FORCE`, you cannot drop an indextype if any domain indexes reference the indextype.

Examples

Dropping an Indextype: Example

The following statement drops the indextype `position_indextype`, created in "Using Extensible Indexing ", and marks `INVALID` any domain indexes defined on this indextype:

```
DROP INDEXTYPE position_indextype FORCE;
```

DROP INMEMORY JOIN GROUP

Purpose

Use the `DROP INMEMORY JOIN GROUP` statement to remove a join group from the database.

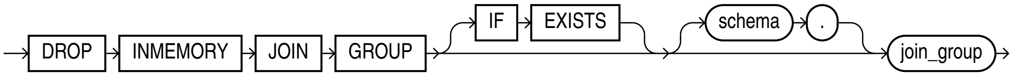
> **See Also:**
>
> - CREATE INMEMORY JOIN GROUP and ALTER INMEMORY JOIN GROUP
> - *Oracle Database In-Memory Guide* for more information on join groups

Prerequisites

If the join group is in another user's schema, then you must have the `DROP ANY TABLE` system privilege.

Syntax

***drop_inmemory_join_group*::=**



Semantics

IF EXISTS

Specify `IF EXISTS` to drop an existing object.

Specifying `IF NOT EXISTS` with `DROP` results in `ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement`.

schema

Specify the schema containing the join group. If you omit *`schema`*, then the database assumes the join group is in your own schema.

join_group

Specify the name of the join group to be dropped.

You can view existing join groups by querying the `DBA_JOINGROUPS` or `USER_JOINGROUPS` data dictionary view. Refer to *Oracle Database Reference* for more information on these views.

Examples

The following statement drops the join group `prod_id1`:

```
DROP INMEMORY JOIN GROUP prod_id1;
```

DROP JAVA

Purpose

Use the `DROP JAVA` statement to drop a Java source, class, or resource schema object.

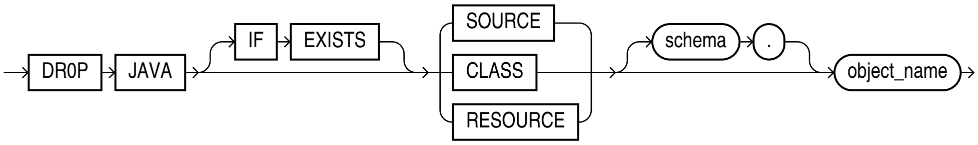
> **See Also:**
>
> - CREATE JAVA for information on creating Java objects
> - *Oracle Database Java Developer's Guide* for more information on resolving Java sources, classes, and resources

Prerequisites

The Java source, class, or resource must be in your own schema or you must have the `DROP ANY PROCEDURE` system privilege. You also must have the `EXECUTE` object privilege on Java classes to use this command.

Syntax

***drop_java*::=**



Semantics

IF EXISTS

Specify `IF EXISTS` to drop an existing object.

Specifying `IF NOT EXISTS` with `DROP` results in `ORA-11544: Incorrect IF EXISTS clause for ALTER/DROP statement`.

JAVA SOURCE

Specify `SOURCE` to drop a Java source schema object and all Java class schema objects derived from it.

JAVA CLASS

Specify `CLASS` to drop a Java class schema object.

JAVA RESOURCE

Specify `RESOURCE` to drop a Java resource schema object.

object_name

Specify the name of an existing Java class, source, or resource schema object. Enclose the *`object_name`* in double quotation marks to preserve lower- or mixed-case names.

Examples

Dropping a Java Class Object: Example

The following statement drops the Java class `Agent`, created in "Creating a Java Class Object: Example":

```
DROP JAVA CLASS "Agent";
```