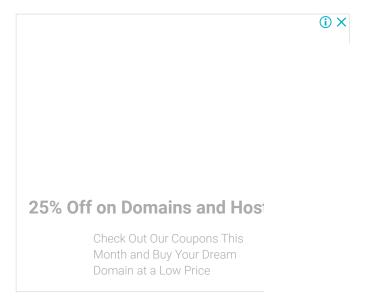HOME   START HERE   BASIC   ADVANCED ⌄   FUNCTIONS ⌄   PL/SQL

Home / Oracle Basics / Oracle DROP TABLE

Oracle DROP TABLE

Summary: in this tutorial, you will learn how to use the Oracle `DROP TABLE` statement to remove an existing table.

Introduction to Oracle `DROP TABLE` statement

To move a table to the recycle bin or remove it entirely from the database, you use the `DROP TABLE` statement:

```
1  DROP TABLE schema_name.table_name
2  [CASCADE CONSTRAINTS | PURGE];
```

First, indicate the table and its schema that you want to drop after the `DROP TABLE` clause. If you don't specify the schema name explicitly, the statement assumes that you are removing the table from your own schema.

Second, specify `CASCADE CONSTRAINTS` clause to remove all referential integrity constraints which refer to primary and unique keys in the table. In case such referential integrity constraints exist and you don't use this clause, Oracle returns an error and stops removing the table.

Third, specify `PURGE` clause if you want to drop the table and release the space associated with it at once. By using the `PURGE` clause, Oracle will not place the table and its dependent objects into the recycle bin.

Notice that the `PURGE` clause does not allow you to roll back or recover the table that you dropped. Therefore, it is useful if you don't want the sensitive data to appear in the recycle bin.

Oracle `DROP TABLE` examples

Let's look at some examples of using the `DROP TABLE` statement.

Basic Oracle `DROP TABLE` example

The following `CREATE TABLE` statement creates `persons` table for the demonstration:

```
1  CREATE TABLE persons (
2      person_id NUMBER,
3      first_name VARCHAR2(50) NOT NULL,
4      last_name VARCHAR2(50) NOT NULL,
5      PRIMARY KEY(person_id)
6  );
```

```
1  DROP TABLE persons;
```

Oracle `DROP TABLE CASCADE CONSTRAINTS` example

The following statements create two new tables named `brands` and `cars` :

```
 1  CREATE TABLE brands(
 2      brand_id NUMBER PRIMARY KEY,
 3      brand_name varchar2(50)
 4  );
 5
 6  CREATE TABLE cars(
 7      car_id NUMBER PRIMARY KEY,
 8      make VARCHAR(50) NOT NULL,
 9      model VARCHAR(50) NOT NULL,
10      year NUMBER NOT NULL,
11      plate_number VARCHAR(25),
12      brand_id NUMBER NOT NULL,
13
14      CONSTRAINT fk_brand
15      FOREIGN KEY (brand_id)
16      REFERENCES brands(brand_id) ON DELETE CASCADE
17  );
```

In these tables, each brand has 1 or more cars while each car belongs to only one brand.

The following statement tries to drop the `brands` table:

```
1  DROP TABLE brands;
```

Oracle issued the following error:

the `cars` table.

The following statement returns all foreign key constraints of the `cars` table:

```
 1  SELECT
 2      a.table_name,
 3      a.column_name,
 4      a.constraint_name,
 5      c.owner,
 6      c.r_owner,
 7      c_pk.table_name r_table_name,
 8      c_pk.constraint_name r_pk
 9  FROM
10      all_cons_columns a
11  JOIN all_constraints c ON
12      a.owner = c.owner
13      AND a.constraint_name = c.constraint_name
14  JOIN all_constraints c_pk ON
15      c.r_owner = c_pk.owner
16      AND c.r_constraint_name = c_pk.constraint_name
17  WHERE
18      c.constraint_type = 'R'
19      AND a.table_name = 'CARS';
```

To drop the `brands` table, you must use the `CASCADE CONSTRAINTS` clause as follows:

```
 1  DROP TABLE brands CASCADE CONSTRAINTS;
```

This statement dropped not only the `brands` table but also the foreign key constraint `fk_brand` from the `cars` table.

↑

Oracle `DROP TABLE PURGE` example

The following statement drops the `cars` table using the `PURGE` clause:

```
1  DROP TABLE cars purge;
```

Drop multiple tables at once

Oracle provides no direct way to drop multiple tables at once. However, you can use the following PL/SQL block to do it:

```
1  BEGIN
2    FOR rec IN
3      (
4        SELECT
5          table_name
6        FROM
7          all_tables
8        WHERE
9          table_name LIKE 'TEST_%'
10     )
11   LOOP
12     EXECUTE immediate 'DROP TABLE  '||rec.table_name || ' CASCADE CONSTRAINTS'
13   END LOOP;
14 END;
15 /
```

This block deletes all tables whose names start with `TEST_`.

To test this code, you can first create three tables: `test_1`, `test_2` and `test_3` as follows:

Then, execute the PL/SQL block above.

In this tutorial, you have learned how to use the Oracle `DROP TABLE` statement to drop a table from the database.

Was this tutorial helpful? 👍 yes 👎 no

HOME   START HERE   BASIC   ADVANCED ⌄   FUNCTIONS ⌄   PL/SQL

Search this website

GETTING STARTED

What Is Oracle Database

Install Oracle Database Server

Download Oracle Sample Database

Create Oracle Sample Database

Connect To Oracle Database Server

ORACLE DATA MANIPULATION

HOME   START HERE   BASIC   ADVANCED ⌄   FUNCTIONS ⌄   PL/SQL

ORDER BY

SELECT DISTINCT

WHERE

Table & Column Aliases

AND

OR

FETCH

BETWEEN

IN

LIKE

Joins

INNER JOIN

LEFT JOIN

RIGHT JOIN

CROSS JOIN

Self Join

GROUP BY

HAVING

UNION

INSERT

INSERT INTO SELECT

INSERT ALL

UPDATE

DELETE

MERGE

Subquery

Correlated Subquery

EXISTS

NOT EXISTS

ANY

ALL

ORACLE DATA TYPES

Oracle Data Types

NUMBER

FLOAT

BINARY_FLOAT

CHAR

NVARCHAR2

DATE

INTERVAL

TIMESTAMP

TIMESTAMP WITH TIME ZONE

ORACLE DATA DEFINITION

CREATE TABLE

Identity Column

ALTER TABLE

ALTER TABLE ADD Column

DROP TABLE

TRUNCATE TABLE

RENAME Table

ORACLE CONSTRAINTS

PRIMARY KEY

FOREIGN KEY

UNIQUE

CHECK

NOT NULL

ORACLE VIEWS

CREATE VIEW

DROP VIEW

Updatable Views

Inline Views

WITH CHECK OPTION

↑

HOME   START HERE   BASIC   ADVANCED ⌄   FUNCTIONS ⌄   PL/SQL

ABOUT ORACLE TUTORIAL

OracleTututorial.com website provides
Developers and Database Administrators with
the updated Oracle tutorials, scripts, and tips.

SEARCH

Search this website

RECENT TUTORIALS

Oracle Joins

Oracle Bitmap Index

Oracle Function-based Index

Oracle UNIQUE Index

Oracle DROP INDEX

SITE LINKS

About

Contact

Privacy Policy

⊤