

DROP INDEX Statement

The **SQL DROP INDEX** statement removes an existing **Index** from a database table.

SQL DROP INDEX

The **SQL DROP INDEX Command** is used to remove an index from the table. Indexes occupy space, which can cause extra time consumption on table modification operations.

Benefits of Using DROP INDEX

- **Removing unused indexes can improve INSERT, UPDATE, and DELETE operations on the table** – Unused indexes can slow down data modification operations, so removing them can enhance performance.
- **It will also free up some space** – Unused indexes consume storage space, and removing them helps to reclaim that space.
- **Deleting an index can have a significant impact on database queries** – Be cautious when dropping indexes, as it can affect query performance. Only drop indexes that are confirmed to be unused or unnecessary.

Note: Indexes created by **PRIMARY KEY** or **UNIQUE constraint** can not be deleted with just a **DROP INDEX statement**. To delete such indexes, we need to first drop the constraints using the **ALTER TABLE** statement, and then drop the index.

Syntax

DROP INDEX Syntax differs in different database systems.

MySQL

```
ALTER TABLE table_name  
DROP INDEX index_name;
```

MS Access

```
DROP INDEX index_name  
ON table_name;
```

SQL Server

```
DROP INDEX table_name.index_name;
```

DB2/Oracle

```
DROP INDEX index_name;
```

PostgreSQL

```
DROP INDEX index_name;
```

SQL DROP INDEX Example

Let's look at some examples of how to drop an index in SQL.

First, let's create a table and add an index using the **CREATE INDEX Statement**. We will be using SQL database in the examples.

```
CREATE DATABASE GEEKSFORGEEKS;  
USE GEEKSFORGEEKS;  
  
CREATE TABLE EMPLOYEE(  
    EMP_ID INT,  
    EMP_NAME VARCHAR(20),  
    AGE INT,  
    DOB DATE,  
    SALARY DECIMAL(7,2)  
);  
CREATE INDEX EMP  
ON EMPLOYEE(EMP_ID, EMP_NAME);
```

Output

```
mysql> CREATE INDEX EMP  
-> ON EMPLOYEE(EMP_ID,EMP_NAME);  
Query OK, 0 rows affected (0.02 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> |
```

Creating an index on two columns

Now let's look at some examples of DROP INDEX statement and understand its workings in SQL. We will learn different use cases of the SQL DROP INDEX statement with examples.

We can drop the index using two ways either with **IF EXISTS** or with **ALTER TABLE** so we will first drop the index using if exists.

SQL DROP INDEX with IF EXISTS Example

Removing an index using SQL DROP INDEX statement with **IF EXISTS clause**, allows the user to remove the index only if it exists in the table.

Query

```
DROP INDEX IF EXISTS EMP  
ON EMPLOYEE;
```

```
mysql> DROP INDEX IF EXISTS EMP  
-> ON EMPLOYEE;
```

Dropping index

Output

Since there are no indexes in the database with the supplied name, the aforementioned query simply ends execution without returning any errors.

Commands Executed Successfully;

SQL DROP index with ALTER TABLE Example

Query

```
ALTER TABLE EMPLOYEE  
DROP INDEX EMP;
```

Output

```
mysql> ALTER TABLE EMPLOYEE  
-> DROP INDEX EMP;  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> |
```

Dropping the index

Verify DROP INDEX

To verify if the DROP INDEX statement has successfully removed the index from the table, we can check the indexes on the table. If the index is not present in the list, we know it has been deleted.

Syntax

The syntax for viewing the index on a table differs for different databases, for example:

SQL Server

```
SELECT * FROM sys.indexes WHERE object_id = ( SELECT object_id FROM sys.objects WHERE name = 'YOUR_TABLE_NAME' );
```

MySQL

```
SHOW INDEXES FROM YOUR_TABLE_NAME;
```

PostgreSQL

```
SELECT * FROM USER_INDEXES;
```

Oracle

```
SELECT indexname, indexdef FROM pg_indexes WHERE tablename = 'your_table_name';
```

Important Points About SQL DROP INDEX Statement

- **The SQL DROP INDEX statement** – This statement is used to remove an existing index from a database table.
- **It optimizes database performance by reducing index maintenance overhead** – Removing unnecessary indexes reduces the overhead of maintaining indexes, which can improve overall database performance.
- **It improves the speed of INSERT, UPDATE, and DELETE operations on the table** – Dropping unused indexes can enhance the speed of data modification operations by reducing the maintenance burden on the database.
- **Indexes created by PRIMARY KEY or UNIQUE constraints cannot be dropped using just the DROP INDEX statement** – These indexes are integral to the table's structure and must be managed through other statements or constraints.
- **The IF EXISTS clause can be used to conditionally drop the index only if it exists** – This clause prevents errors by ensuring that the index is dropped only if it is present.
- **To verify if the DROP INDEX statement has successfully removed the index from the table** – Check the list of indexes on the table to confirm the removal.

SQL DROP INDEX Statement – FAQs

How to create an index in SQL?

To create an index in MYSQL we use the CREATE INDEX command.

How to drop an index in SQL?

To drop an index in SQL we use the ALTER TABLE DROP INDEX command.

What is the need to drop an index?

Generally we drop an index and then recreate it because it increases the data insertion speed.