Create and Drop Indexes

by Sophia

WHAT'S COVERED

This lesson explores using the CREATE INDEX and DROP INDEX commands to create and remove indexes, in two parts. Specifically, this lesson will cover:

1. Creating an Index
2. Dropping an Index

1. Creating an Index

So far in this challenge, you have learned the potential benefits of creating indexes, and about the different types of indexes you can create. Now it's time to learn how to actually create an index as well as how to drop (delete) one.

The syntax for creating an index—using the **CREATE INDEX** statement—is as follows:

```
CREATE [UNIQUE] INDEX <indexname>
ON<tablename> (<columnname>) [USING method];
```

Notice the optional UNIQUE clause in that syntax. When you create an index, you can optionally enforce having unique values in that column. For example, the following statement creates an index on the customer_email column and also sets a constraint requiring the values in the customer_email column to be unique for each record.

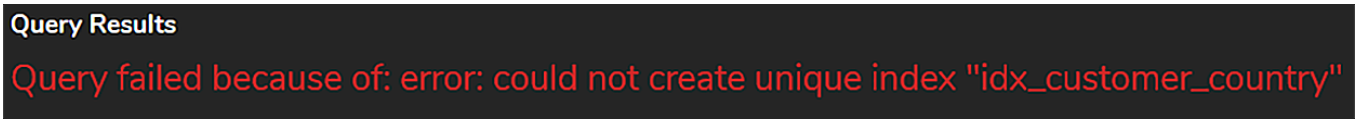
```
CREATE UNIQUE INDEX idx_customer_country
ON customer(country);
```

If there are already duplicates in that column, the index creation will fail.

In the above examples, note that the index name is written as “idx” plus the table name and then the country name. This naming convention is common but not required; you can name the index anything you like.

The UNIQUE constraint would not be appropriate to use in a column where there are some duplicates—or might be in the future. For example, if you tried to create an index on the country column in the customer table, an error would appear if there were already multiple customers in the same country.

```
CREATE UNIQUE INDEX idx_customer_country
ON customer(country);
```



You could, however, create an index on the country in general:

```
CREATE INDEX idx_customer_country
ON customer(country);
```

By default, PostgreSQL creates a B-tree index. However, the PostgreSQL versions 10 and later support specifying an index type with the USING clause. For example, to use a hash index, you would specify it at creation like this:

```
CREATE INDEX idx_customer_country USING hash
ON customer(country);
```

TERM TO KNOW

CREATE INDEX

A statement that enables you to create a new index for a column in a table.

2. Dropping an Index

If an index isn't helping with query performance, we typically would want to remove it to ensure it does not impair performance. The syntax for the **DROP INDEX** statement is as follows:

```
DROP INDEX [CONCURRENTLY] [IF EXISTS] <indexname> [CASCADE or RESTRICT or FORCE];
```

Let's examine the statement's options.

The CONCURRENTLY option allows you to drop the index without blocking concurrent operations on the table. That's useful for large tables where blocking other operations for the duration of the index drop would not be acceptable.

The IF EXISTS option prevents an error from occurring if the index does not exist. For example, the following would execute whether or not the index exists:

```
DROP INDEX IF EXISTS idx_customer_country;
```

The CASCADE option automatically drops objects that depend on the index. For example, if there are foreign key constraints or views dependent on the index, they will be dropped too. For example:

```
DROP INDEX idx_customer_country CASCADE;
```

The RESTRICT option prevents the drop operation if there are any dependent objects on the index. RESTRICT is the default behavior if you do not specify CASCADE or FORCE. For example:

```
DROP INDEX idx_customer_country RESTRICT;
```

The FORCE option drops the index without checking if any objects depend on it. Use this option with caution, as it can lead to data integrity issues.

TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then, enter one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.

TERM TO KNOW

DROP INDEX

A statement that enables you to drop (delete) an index.

SUMMARY

In this lesson, you learned that the CREATE INDEX statement is used to **create an index** in a database table using specific SQL columns. This command enhances data retrieval efficiency by creating a data structure that improves query performance. With customized indexes, queries involving indexed columns can be processed more quickly. Based on the data and query patterns, different types of indexes, such as B-trees, hashes, and bitmaps, offer varying benefits.

A DROP INDEX command, on the other hand, removes an existing index from a database table. **Dropping an index** can be necessary when the index becomes redundant due to changes in query patterns or data characteristics. It can also be required when reclaiming storage space is necessary. Before dropping indexes, however, it is important to consider their impact on query performance. Dropping an index might cause queries to execute slower, especially if the dropped index is used in the query. The database's requirements and performance implications should be thoroughly considered when dropping an index.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND FAITHE WEMPEN (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR **TERMS OF USE**.

TERMS TO KNOW

CREATE INDEX

A statement that enables you to create a new index for a column in a table.

DROP INDEX

A statement that enables you to drop (delete) an index.