Introduction to PostgreSQL ALTER TABLE statement

To change the structure of an existing table, you use PostgreSQL `ALTER TABLE` statement.

The following illustrates the basic syntax of the `ALTER TABLE` statement:

```
ALTER TABLE table_name action;
```

Code language: SQL (Structured Query Language) (sql)

PostgreSQL provides you with many actions:

- Add a column
- Drop a column
- Change the data type of a column
- Rename a column
- Set a default value for the column.
- Add a constraint to a column.
- Rename a table

To add a new column to a table, you use `ALTER TABLE ADD COLUMN` statement:

```
ALTER TABLE table_name ADD COLUMN column_name datatype column_constraint;
```

Code language: SQL (Structured Query Language) (sql)

To drop a column from a table, you use `ALTER TABLE DROP COLUMN` statement:

```
ALTER TABLE table_name DROP COLUMN column_name;
```

Code language: SQL (Structured Query Language) (sql)

To rename a column, you use the `[ALTER TABLE RENAME COLUMN](https://www.postgresqltutorial.com/postgresql-rename-column/) TO` statement:

```
ALTER TABLE table_name RENAME COLUMN column_name TO new_column_name;
```

Code language: SQL (Structured Query Language) (sql)

To change a default value of the column, you use `ALTER TABLE ALTER COLUMN SET DEFAULT` or `DROP DEFAULT`:

```
ALTER TABLE table_name ALTER COLUMN column_name [SET DEFAULT value | DROP DEFAULT];
```

Code language: SQL (Structured Query Language) (sql)

To change the `NOT NULL` constraint, you use `ALTER TABLE ALTER COLUMN` statement:

```
ALTER TABLE table_name ALTER COLUMN column_name [SET NOT NULL| DROP NOT NULL];
```

Code language: SQL (Structured Query Language) (sql)

To add a `CHECK` constraint, you use `ALTER TABLE ADD CHECK` statement:

```
ALTER TABLE table_name ADD CHECK expression;
```

Code language: SQL (Structured Query Language) (sql)

Generailly, to add a constraint to a table, you use `ALTER TABLE ADD CONSTRAINT` statement:

```
ALTER TABLE table_name ADD CONSTRAINT constraint_name constraint_definition;
```

Code language: SQL (Structured Query Language) (sql)

To rename a table you use `ALTER TABLE RENAME TO` statement:

```
ALTER TABLE table_name RENAME TO new_table_name;
```

Code language: SQL (Structured Query Language) (sql)

PostgreSQL `ALTER TABLE` examples

Let's create a new table called `links` for practicing with the `ALTER TABLE` statement.

```
DROP TABLE IF EXISTS links; CREATE TABLE links ( link_id serial PRIMARY KEY, title VARCHAR (512) NOT NULL, url VARCHAR (1024) NOT NULL );
```

Code language: SQL (Structured Query Language) (sql)

To add a new column named `active`, you use the following statement:

```
ALTER TABLE links ADD COLUMN active boolean;
```

Code language: SQL (Structured Query Language) (sql)

The following statement removes the `active` column from the `links` table:

```
ALTER TABLE links DROP COLUMN active;
```

Code language: SQL (Structured Query Language) (sql)

To change the name of the `title` column to `link_title`, you use the following statement:

```
ALTER TABLE links RENAME COLUMN title TO link_title;
```

Code language: SQL (Structured Query Language) (sql)

The following statement adds a new column named `target` to the `links` table:

```
ALTER TABLE links ADD COLUMN target VARCHAR(10);
```

Code language: SQL (Structured Query Language) (sql)

To set `_blank` as the default value for the `target` column in the `links` table, you use the following statement:

```
ALTER TABLE links ALTER COLUMN target SET DEFAULT '_blank';
```

Code language: SQL (Structured Query Language) (sql)

If you insert the new row into the `links` table without specifying a value for the `target` column, the `target` column will take the `_blank` as the default value. For example:

```
INSERT INTO links (link_title, url) VALUES('PostgreSQL
Tutorial','https://www.postgresqltutorial.com/');
```

Code language: SQL (Structured Query Language) (sql)

The following statement selects data from the `links` table:

```
SELECT * FROM links;
```

Code language: SQL (Structured Query Language) (sql)

link_id	link_title	url	target
1	PostgreSQL Tutorial	http://www.postgresqltutorial.com/	_blank

The following statement adds a `CHECK` condition to the `target` column so that the `target` column only accepts the following values: `_self`, `_blank`, `_parent`, and `_top`:

```
ALTER TABLE links ADD CHECK (target IN ('_self', '_blank', '_parent', '_top'));
```

Code language: SQL (Structured Query Language) (sql)

If you attempt to insert a new row that violates the `CHECK` constraint set for the `target` column, PostgreSQL will issue an error as shown in the following example:

```
INSERT INTO links(link_title,url,target)
VALUES('PostgreSQL','http://www.postgresql.org/','whatever');
```

Code language: SQL (Structured Query Language) (sql)

```
ERROR: new row for relation "links" violates check constraint "links_target_check"
DETAIL: Failing row contains (2, PostgreSQL, http://www.postgresql.org/,
whatever).DETAIL: Failing row contains (2, PostgreSQL, http://www.postgresql.org/,
whatever).
```

Code language: Shell Session (shell)

The following statement adds a `UNIQUE` constraint to the `url` column of the `links` table:

```
ALTER TABLE links ADD CONSTRAINT unique_url UNIQUE ( url );
```

Code language: SQL (Structured Query Language) (sql)

The following statement attempts to insert the url that already exists:

```
INSERT INTO links(link_title,url)
VALUES('PostgreSQL','https://www.postgresqltutorial.com/');
```

Code language: SQL (Structured Query Language) (sql)

It causes an error due to the unique_url constraint:

```
ERROR: duplicate key value violates unique constraint "unique_url" DETAIL: Key
(url)=(https://www.postgresqltutorial.com/) already exists.
```

```
Code language: Shell Session (shell)
```

The following statement changes the name of the `links` table to `urls`:

```
ALTER TABLE links RENAME TO urls;
```

```
Code language: SQL (Structured Query Language) (sql)
```

In this tutorial, you have learned how to use the PostgreSQL `ALTER TABLE` statement to change the structure of an existing table.