CASSANDRA - CQL USER DEFINED DATATYPES

CQL provides the facility of creating and using user-defined data types. You can create a data type to handle multiple fields. This chapter explains how to create, alter, and delete a user-defined data type.

Creating a User-defined Data Type

The command **CREATE TYPE** is used to create a user-defined data type. Its syntax is as follows:

```
CREATE TYPE <keyspace name>. <data typename>
( variable1, variable2).
```

**Example**

Given below is an example for creating a user-defined data type. In this example, we are creating a **card_details** data type containing the following details.

<table>
<thead>
<tr>
<th>Field</th>
<th>Field name</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>credit card no</td>
<td>num</td>
<td>int</td>
</tr>
<tr>
<td>credit card pin</td>
<td>pin</td>
<td>int</td>
</tr>
<tr>
<td>name on credit card</td>
<td>name</td>
<td>text</td>
</tr>
<tr>
<td>cvv</td>
<td>cvv</td>
<td>int</td>
</tr>
<tr>
<td>Contact details of card holder</td>
<td>phone</td>
<td>set</td>
</tr>
</tbody>
</table>

CQLsh:tutorialspoint> CREATE TYPE card_details (
    ... num int,
    ... pin int,
    ... name text,
    ... cvv int,
    ... phone set<int>
    ... );

**Note:** The name used for user-defined data type should not coincide with reserved type names.

**Verification**

Use the **DESCRIBE** command to verify whether the type created has been created or not.

```
CREATE TYPE tutorialspoint.card_details (num int, pin int, name text, cvv int, phone set<int>);
```

Altering a User-defined Data Type

**ALTER TYPE** command is used to alter an existing data type. Using ALTER, you can add a new field or rename an existing field.

**Adding a Field to a Type**
Use the following syntax to add a new field to an existing user-defined data type.

```sql
ALTER TYPE typename
ADD field_name field_type;
```

The following code adds a new field to the Card_details data type. Here we are adding a new field called email.

```sql
cqlsh:tutorialspoint> ALTER TYPE card_details ADD email text;
```

**Verification**

Use the `DESCRIBE` command to verify whether the new field is added or not.

```sql
cqlsh:tutorialspoint> describe type card_details;
CREATE TYPE tutorialspoint.card_details (  
    num int,  
    pin int,  
    name text,  
    cvv int,  
    phone set<int>,  
    email text
);
```

**Renaming a Field in a Type**

Use the following syntax to rename an existing user-defined data type.

```sql
ALTER TYPE typename
RENAME existing_name TO new_name;
```

The following code changes the name of the field in a type. Here we are renaming the field email to mail.

```sql
cqlsh:tutorialspoint> ALTER TYPE card_details RENAME email TO mail;
```

**Verification**

Use the `DESCRIBE` command to verify whether the type name changed or not.

```sql
cqlsh:tutorialspoint> describe type card_details;
CREATE TYPE tutorialspoint.card_details (  
    num int,  
    pin int,  
    name text,  
    cvv int,  
    phone set<int>,  
    mail text
);
```

**Deleting a User-defined Data Type**

`DROP TYPE` is the command used to delete a user-defined data type. Given below is an example to delete a user-defined data type.

**Example**

Before deleting, verify the list of all user-defined data types using `DESCRIBE TYPES` command as shown below.

```sql
cqlsh:tutorialspoint> DESCRIBE TYPES;
card_details card
```
From the two types, delete the type named **card** as shown below.

cqlsh:tutorialspoint> drop type card;

Use the **DESCRIBE** command to verify whether the data type dropped or not.

cqlsh:tutorialspoint> describe types;
card_details