Concept

A table is like the nanny of all data, and it is a cornerstone of the database. It can be said that a good table design can significantly improve the maintainability and performance speed of the database.

First, let’s introduce the four characteristics of tables: 1. Tables record the metadata of each column, including the data type, default value, comments, etc. 2. Tables are responsible for managing the index, which can improve the speed of searching for data in the table. 3. Provide relationship definitions with other tables. There will be associations between data tables. For example, a user in the `users` table may have many order records in the `orders` table, and this part of the affiliation needs to be managed in the table. 4. You can set the location where the data is actually stored in the computer hard drive.

SQL Syntax

Create Table

A practical table in SQL will look like this:

id	name	age
1	John	40
2	May	30
3	Tim	25

In order to create a table (to keep things simple, we'll start with an empty table), we can use the following statement:

```
CREATE SCHEMA `new_schema` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

CREATE TABLE `new_schema`.`new_table` (
  `id` INT NOT NULL COMMENT 'This is a primary index',
  PRIMARY KEY (`id`)
);
```

The statement can be split into three parts:

1. The first part is the easiest to understand, but the syntax changes from `SCHEMA` to `TABLE`; that is, the establishment of the table itself. Also, pay attention to the statement `new_schema`.`new_table` because there will be many schemas in our database software, and we must describe in which schema the database should create the table. Finally, `new_table` is the name of the table we want to create.

```
CREATE TABLE `new_schema`.`new_table` (
    -- other settings
);
```

2. The second part is to define the details of each field. We will explain this part in detail in the next chapter. Here, we will simply introduce the meaning of the syntax:

```
`id` INT NOT NULL COMMENT 'This is a primary index';
```

- `id` is the column name
- `INT` is the data type that will be stored in this column.
- `NOT NULL` is a kind of column attribute function. We will cover this in more detail in the next chapter.
- This column has a `COMMENT` with the display text `This is a primary index`.

3. The last part is to declare the metadata of this table (not for the column). For example, here is how to set the primary key of this table to be `id`.

```
PRIMARY KEY (`id`)
```

What is Primary Key?

The Primary Key is a vital part of each table in the database system. It can be any column or a group of multiple columns, and there can only be one primary key in a table. The value cannot be repeated or null.

It is a very important field for concatenating data tables and improving the efficiency of searching data. If you want to learn more about the technical aspects of the primary key, you can do so here: Primary Key.

After creating the table, we can check whether the column settings are as we expected, or if there is a typo in any of the column names, we can use the `SHOW FULL COLUMNS` statement to view the content of the defined table:

```
SHOW FULL COLUMNS FROM `new_schema`.`new_table`;
```

The result should be:

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
id	int(11)		NO	PRI			select, insert, update, references	This is a primary index

By the way, `SHOW` is a very common statement when retrieving information from the database system level. Here is a reference for different things you can show: MySQL SHOW Statements.

Destroy Table

The statement for deleting a table is as simple:

```
DROP TABLE `new_schema`.`new_table`;
```

The `DROP` keyword means to remove a table in the database, and it is also a very dangerous statement. If we are not careful, we may lose important data, so in practical applications, we rarely use this statement.

Clean Table

While we rarely delete a table, when we want to test our application, we can reset the data state inside the table.

In order to regenerate the test data, we clear the table frequently, then we will use the `TRUNCATE` statement, to delete all data, but not the table. Note that this statement should also be used with caution:

```
TRUNCATE `new_schema`.`new_table`;
```