Usually, when you use SQL, you must log in first, and you will use the SQL user account when logging in. The management of these user accounts is an integral part of ensuring database security. This chapter will show you how to manage user accounts in SQL.

Create a User

Before you can manage an account, you must first have an account managed. The syntax for creating a user is as follows:

```
CREATE USER 'john'@'localhost' IDENTIFIED BY 'password';
```

This statement can be split into 3 parts for analysis: 1. `CREATE USER`: A keyword used by SQL to create a user, just like the concept of `CREATE SCHEMA`, `CREATE TABLE`. 2. `'john'@'localhost'`: The former `john` is our user's account, just like the default user `root` common in SQL; the latter `localhost` refers to the network context where the account can use the database from. We will cover this in detail later. 3. `IDENTIFIED BY 'password'`: `IDENTIFIED BY` is the keyword used by SQL to set the password, `'password'` is the password we set for the user `john`.

Network Context

Regarding the network context mentioned in the previous section, it refers to the context of how users connect to the database. There are three main types: 1. `localhost`: Let me introduce it first. There is often a website system situation where the website server and the database server run on the same machine. `localhost` means that the database can only be connected from the same machine. Users can execute SQL statements only through the functions provided by the website application, but cannot directly connect to the database in their homes or other places. 2. `110.78.9.12` (or any IP): Here 110.78.9.12 is just an arbitrary IP. If you specify the IP, you restrict that the user is only allowed to connect database from that specific IP. 3. `%`: The percent symbol means the user can connect to the database from anywhere. This is a very powerful setting, and it should be used with great care. Otherwise, it is possible that strangers could connect to our database when the password is leaked.

Show a User

If we want to see our newly added users, we can execute:

```
SELECT * FROM `mysql`.`user`
```

We will get results like the following:

Host	User	Select_priv	...other priv
localhost	root	Y	Y or N...
localhost	john	N	Y or N...

You may wonder: what is `priv`? The next section will explain this!

Grant a User

After the user is established, we will set the permissions again. Let's take a look at the SQL statements we will use when setting permissions:

```
GRANT ALL ON `new_schema`.`orders` TO 'john'@'localhost';
GRANT SELECT ON `new_schema`.* TO 'root'@'150.10.12.1';
```

Here we can focus on the following syntax: 1. `GRANT [privilege] ON`: This is the syntax for setting privileges in SQL. `GRANT` and `ON` are fixed, and `[privilege]` can be replaced with the privilege name provided by SQL. For example, `ALL` refers to all privileges, and `SELECT` refers to the permissions of the `SELECT` syntax for the user. 2. `TO`: The latter is directly connected to the user who has been set the permission.

The names of privileges are not the same in different databases. Regarding the MySQL used in this chapter, if you would like to learn more, you can refer to MySQL privileges-provided.