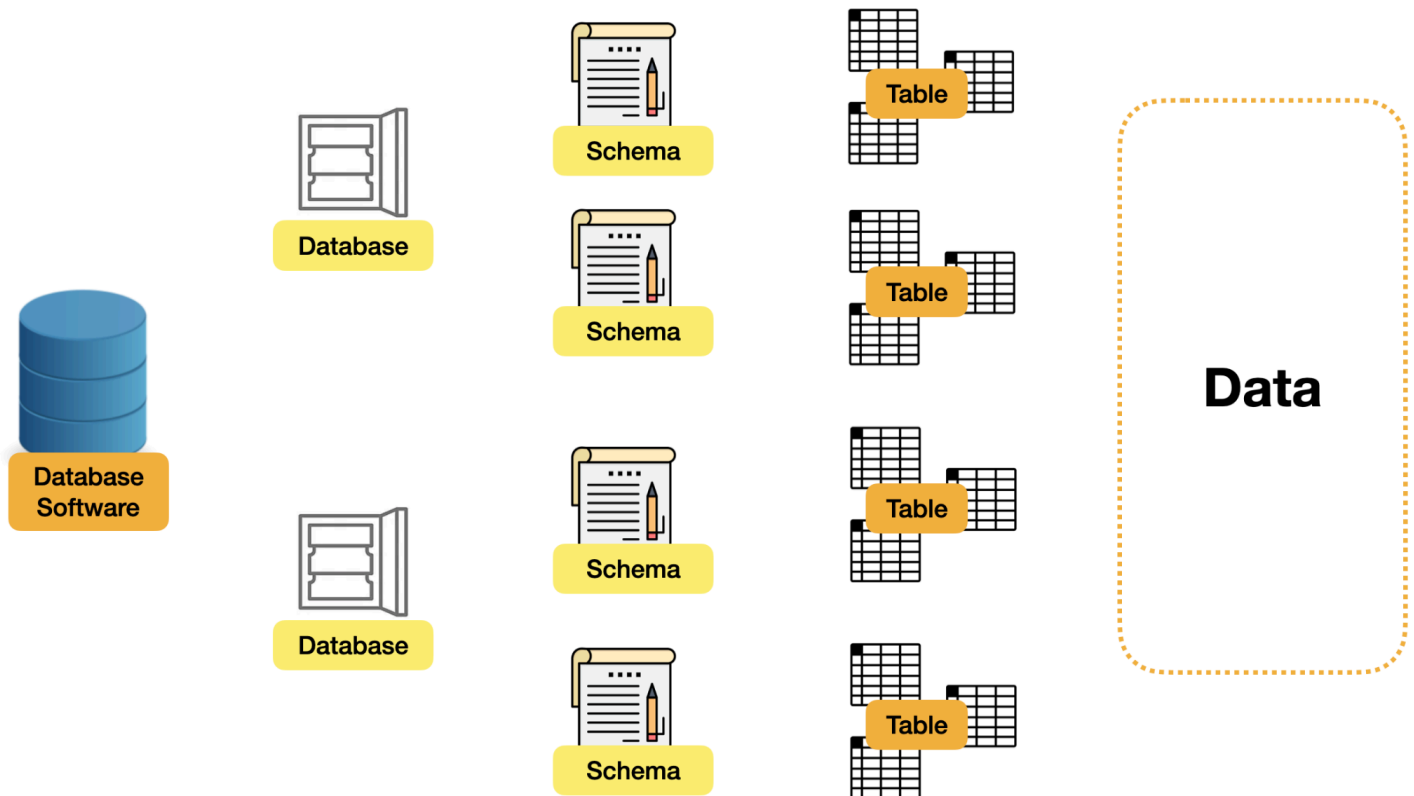


General Database Structure



The image above describes what the architecture looks like in a typical database software ecosystem. 1. **Database Software**: It can be regarded as common database software, such as MySQL, Postgre SQL, MsSQL, etc. We will use MySQL here because it's one of the most widely used database software in the world. 2. **Database**: We can divide different databases' structures into database software. In general, we can separate data entities with different commercial meanings. For example, the US stock data can be set in one database, and the London stock data can be set in another database. 3. **Schema**: This is a collection of **Tables**, which can be split and grouped according to logic, and some table details meta settings can be set on this layer, like a blueprint. But in some database software, such as MySQL, **Schema** and **Database** are integrated into one. 4. **Table**: The key section of the database. **Tables** record different data entities, such as `users`, `products`, etc. Each **table** is responsible for recording the column size, type, and default value... of each row data in the database. 5. **Data**: The most valuable part of the database is the data itself. Different types of data can be stored.

Schema Syntax

Basic SQL Syntax

Usually in the database, when we want to create a Schema, the corresponding SQL syntax structure is as follows:

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

Here are a few basic points of SQL syntax: - The all-capital part is the so-called 'keyword', which is the vocabulary that the SQL programming language needs to use. Because it has special meanings in SQL language, we cannot use it to name our `schema`, `table`..etc. - A pair of ' ` ' symbols, the word ' new_schema ' enclosed together, this is part of our own input, declaring that the name of the schema we want to create is new_schema

Create Schema

The above statement can be divided into three parts to make it easier to understand:

1. The first component is the simplest; it creates a Schema and gives it a name.

```
CREATE SCHEMA `new_schema`
```

1. Set the character encoding because the symbols used for human language are very diverse. For example, Japanese and Arabic will have different symbol systems, so the corresponding software also has many character encoding methods; we set a common character 4-Byte [UTF-8 Unicode Encoding](#) series here.

```
DEFAULT CHARACTER SET utf8mb4
```

1. Third point, in the character encoding, there will also be some derivatives. The example here uses `utf8mb4_unicode_ci`, which is a derivative that can use emoji-related symbols. This is only shown as an example here.

```
COLLATE utf8mb4_unicode_ci;
```

To learn more about the encoding, please refer to [What is UTF-8 Encoding](#).

Most importantly, after the SQL statement is described, you need a semicolon 『 ; 』 to indicate the end point of program execution! After the statement has been executed and creation is complete, the following SQL statements can be executed:

```
SELECT * FROM INFORMATION_SCHEMA.SCHEMATA;
```

And as we can see from the result of this statement (on the result tab, below), we now have a new schema called "new_schema":

As for `INFORMATION_SCHEMA.SCHEMATA`, it is the place where `SCHEMA` data is stored in MySQL. The basic data of Schema can be seen here.