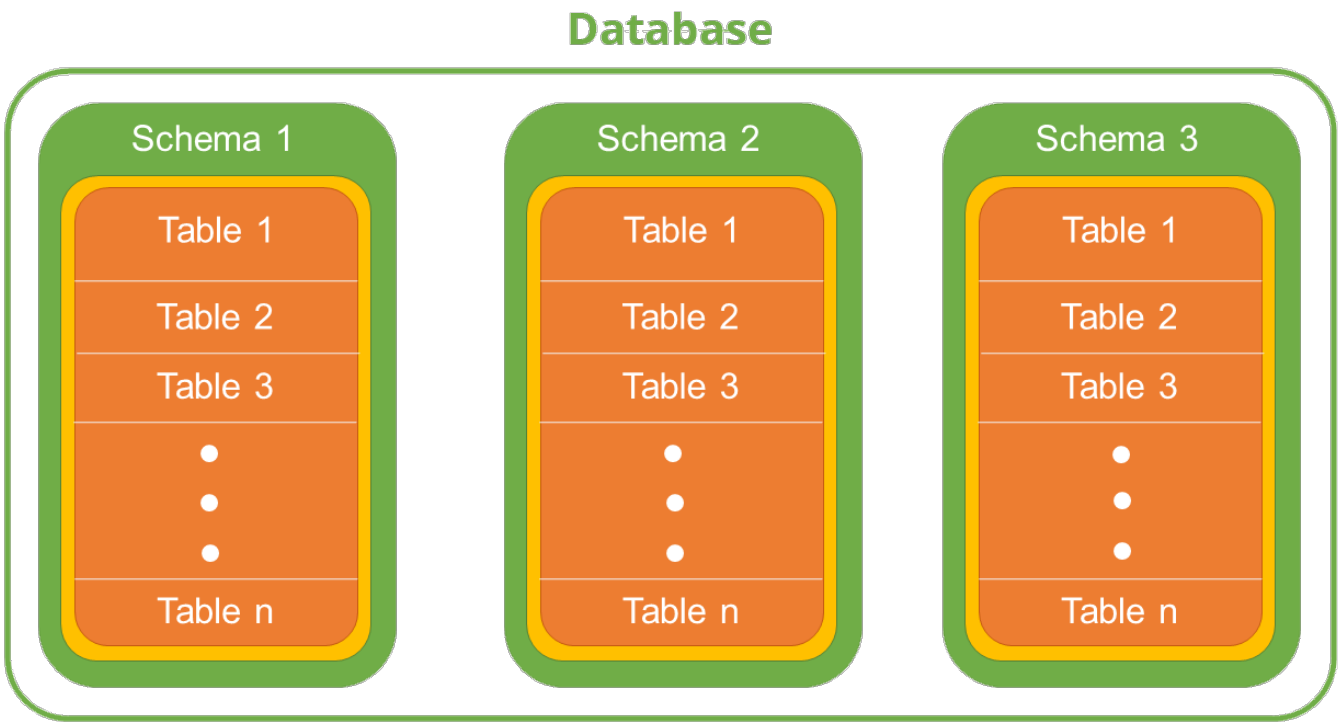
DB2 - SCHEMAS

This chapter introduces and describes the concept of Schema.

Introduction

A schema is a collection of named objects classified logically in the database.

In a database, you cannot create multiple database objects with same name. To do so, the schema provides a group environment. You can create multiple schemas in a database and you can create multiple database objects with same name, with different schema groups.



A schema can contain tables, functions, indices, tablespaces, procedures, triggers etc. For example, you create two different schemas named as “Professional” and “Personal” for an “employee” database. It is possible to make two different tables with the same name “Employee”. In this environment, one table has professional information and the other has personal information of employee. In spite of having two tables with the same name, they have two different schemas “Personal” and “Professional”. Hence, the user can work with both without encountering any problem. This feature is useful when there are constraints on the naming of tables.

Let us see few commands related to Schema:

Getting currently active schema

Syntax:

```
db2 get schema
```

Example: [To get current database schema]

```
db2 get schema
```

Setting another schema to current environment

Syntax:

```
db2 set schema=<schema_name>
```

Example: [To arrange ‘schema1’ to current instance environment]

```
db2 set schema=schema1
```

Creating a new Schema

Syntax: [To create a new schema with authorized user id]

```
db2 create schema <schema_name> authroization <inst_user>
```

Example: [To create “schema1” schema authorized with ‘db2inst2”]

```
db2 create schema schema1 authorization db2inst2
```

Exercise

Let us create two different tables with same name but two different schemas. Here, you create employee table with two different schemas, one for personal and the other for professional information.

Step 1: Create two schemas.

Schema 1: [To create schema named professional]

```
db2 create schema professional authorization db2inst2
```

Schema 2: [To create schema named personal]

```
db2 create schema personal authorization db2inst2
```

Step 2: Create two tables with the same name for Employee details

Table1: professional.employee

[To create a new table ‘employee’ in the database using schema name ‘professional’]

```
db2 create table professional.employee(id number, name
varchar(20), profession varchar(20), join_date date,
salary number);
```

Table2: personal.employee

[To create a new table ‘employee’ in the same database, with schema name ‘personal’]

```
db2 create table personal.employee(id number, name
varchar(20), d_birth date, phone bigint, address
varchar(200));
```

After executing these steps, you get two tables with same name ’employee’, with two different schemas.