RKIT Software Private Limited

Gupta Heer Dipak
Full Stack Developer

Pre-Joining Topics

Week 1: MYSQL Basics

DDL (Data Definition language)

DDL commands are those that can be used to define the database schema. It consists of metadata of the database schema and also create and modify the structure of the various objects within the database.

DDL commands:

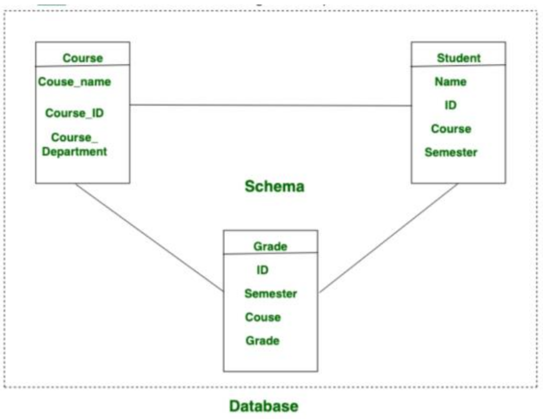
CREATE – is used to create the database or its objects (table index, function, views, store procedure and triggers).

Examples:

CREATE DATABASE Statement

CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] *db_name*

- A Database is an **organized collection of data**
- A **schema** is the structural design or **blueprint of a database**.



CREATE TABLE Statement

```
CREATE TABLE employee_table(

id int NOT NULL AUTO_INCREMENT,

name varchar(45) NOT NULL,

occupation varchar(35) NOT NULL,
age int NOT NULL,

PRIMARY KEY (id)

);
```

CREATE PROCEDURE Statements

```
CREATE   PROCEDURE [IF NOT EXISTS] sp_name ([proc_parameter[,...]])

proc_parameter:

   [ IN | OUT | INOUT] param_name type

CREATE PROCEDURE citycount (IN country CHAR(3), OUT cities INT)

     BEGIN

       SELECT COUNT(*) INTO cities FROM world.city

       WHERE CountryCode = country;

     END

CALL citycount('JPN', @cities);
```

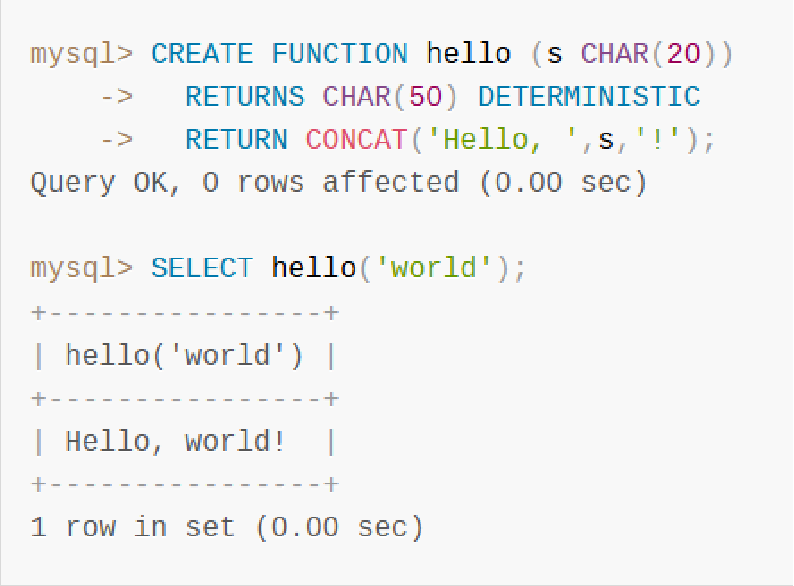
CREATE FUNCTION Statements

```
CREATE FUNCTION [IF NOT EXISTS] sp_name ([func_parameter[,...]])

   RETURNS type

func_parameter:

   param_name type
```



```
mysql> CREATE FUNCTION hello (s CHAR(20))
    ->   RETURNS CHAR(50) DETERMINISTIC
    ->   RETURN CONCAT('Hello, ',s,'!');
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT hello('world');
+----------------+
| hello('world') |
+----------------+
| Hello, world!  |
+----------------+
1 row in set (0.00 sec)
```

CREATE VIEW Statement

CREATE VIEW *view_name* [(*column_list*)]

AS *select_statement*

CREATE VIEW test.v AS SELECT * FROM t;

ALTER -is used to alter the structure of the database.

- **Adding a Column:** Used to add a new column to an existing table.

Code

```
ALTER TABLE table_name
ADD column_name datatype [constraints];
```

- **Dropping a Column:** Used to remove an existing column from a table.

Code

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

- **Modifying a Column:** Used to change the data type, size, or other characteristics of an existing column.

Code

```
ALTER TABLE table_name
MODIFY COLUMN column_name new_datatype [new_constraints];
```

- **Renaming a Column:** Used to change the name of an existing column.

Code

```
ALTER TABLE table_name
RENAME COLUMN old_column_name TO new_column_name;
```

- **Renaming a Table:** Used to change the name of the table itself.

Code

```
ALTER TABLE old_table_name
RENAME TO new_table_name;
```

- **Adding or Dropping Constraints:** Used to add or remove primary keys, foreign keys, unique constraints, check constraints, or default values.

Code

```
-- Add a primary key
ALTER TABLE table_name
ADD PRIMARY KEY (column_name);

-- Add a foreign key
ALTER TABLE table_name
ADD FOREIGN KEY (column_name) REFERENCES other_table (other_column);

-- Drop a constraint
ALTER TABLE table_name
DROP CONSTRAINT constraint_name;
```

- **Changing Table Options:** Used to modify table-level options like the storage engine (ENGINE), character set (CHARACTER SET), or collation (COLLATE).

Code

```
ALTER TABLE table_name
ENGINE = InnoDB;
```

DROP – is used to delete objects from the database.

Code

```
DROP TABLE table_name;
```

Code

```
DROP TABLE IF EXISTS table_name;
```

References

- Mysql Reference Manual
- Google