CREATE DATABASE

Creates a database with the specified name. If database with the same name already exists, an exception will be thrown.

Syntax:

**CREATE** { **DATABASE** | **SCHEMA** } [ IF **NOT** **EXISTS** ] database\_name

[ **COMMENT** database\_comment ]

[ **LOCATION** database\_directory ]

[ **WITH** DBPROPERTIES ( property\_name = property\_value [ , ... ] ) ]

* *Create database `customer\_db`. This throws exception if database with name customer\_db already exists.*
* **CREATE** **DATABASE** customer\_db;
* *Create database `customer\_db` only if database with same name doesn't exist.*
* **CREATE** **DATABASE** IF **NOT** **EXISTS** customer\_db;
* *Create database `customer\_db` only if database with same name doesn't exist with Comments`,`Specific Location` and `Database properties`.*
* **CREATE** **DATABASE** IF **NOT** **EXISTS** customer\_db **COMMENT** 'This is customer database' **LOCATION** '/user'

**WITH** DBPROPERTIES (ID=001, Name='John');

* *Verify that properties are set.*
* *Describe* customer\_db *DATABASE with EXTENDED option to return additional database properties*
* **DESCRIBE** **DATABASE** EXTENDED customer\_db;

|database\_description\_item|database\_description\_value|

**Database** Name| customer\_db

Description| This **is** customer **database**

**Location**| hdfs://hacluster/**user**

Properties| ((ID,001), (Name,John))

CREATE FUNCTION

The CREATE FUNCTION statement is used to create a temporary or permanent function in Spark.

Syntax

**CREATE** [ **OR** **REPLACE** ] [ **TEMPORARY** ] **FUNCTION** [ IF **NOT** **EXISTS** ]

function\_name **AS** class\_name [ resource\_locations ]

syntax:

* *Create a table called `test` and insert two rows.*
* **CREATE** **TABLE** test(c1 INT);

**INSERT** **INTO** test **VALUES** (1), (2);

* *Create a permanent function called `simple\_udf`.*

CREATE FUNCTION simple\_udf AS ‘SimpleUdf’

USING JAR ‘/tmp/Simpleudf.jar’;

* *Verify that the function is in the registry.*

**SHOW** **USER** FUNCTIONS;

* *Invoke the function. Every selected value should be incremented by 10.*

**SELECT** simple\_udf(c1) **AS** function\_return\_value **FROM** test;

function\_return\_value

11

12

* *Created a temporary function.*

**CREATE** **TEMPORARY** **FUNCTION** simple\_temp\_udf **AS** 'SimpleUdf'

**USING** JAR '/tmp/SimpleUdf.jar';

* **SHOW** **USER** FUNCTIONS;

**Function:**

**default**.simple\_udf

simple\_temp\_udf

* *Replace the implementation of `simple\_udf`*

**CREATE** **OR** **REPLACE** **FUNCTION** simple\_udf **AS** 'SimpleUdfR'

**USING** JAR '/tmp/SimpleUdfR.jar';

* *Invoke the function. Every selected value should be incremented by 20.*

SELECT simple\_udf(C!) AS function\_return\_value FROM test;

Function \_return\_value

21,22

22

CREATE DATASOURCE TABLE

* *Use data source*

CREATE TABLE student (id INT, name STRING, age INT) USING CSV;

Here CSV is the Data siurce

* *Use data from another table*

CRESTE TABLE student\_copy USING CSV AS

SELECT \* FROM student

* *Omit the USING clause, which uses the default data source (parquet by default)*

CREATE TABLE student (id INT, name STRING, age INT)

CREATE VIEW

* *Create or replace view for `experienced\_employee` with comments.*

**CREATE** **OR** **REPLACE** **VIEW** experienced\_employee

(ID **COMMENT** 'Unique identification number', Name)

**COMMENT** 'View for experienced employees'

**AS** **SELECT** id, name **FROM** all\_employee

**WHERE** working\_years > 5;

* *Create a global temporary view `subscribed\_movies` if it does not exist.*

**CREATE** **GLOBAL** **TEMPORARY** **VIEW** IF **NOT** **EXISTS** subscribed\_movies

**AS** **SELECT** mo.member\_id, mb.full\_name, mo.movie\_title

**FROM** movies **AS** mo **INNER** **JOIN** members **AS** mb

**ON** mo.member\_id = mb.id;