



Inserting Data into a Database

Database Fundamentals

Name of presenter

Date

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Welcome to Inserting Data into a Database.

What you will learn

At the core of the lesson

You will learn how to:

- Insert rows into an existing table
- Identify potential problems that might require data to be cleansed before being imported into a database
- Import a CSV file into a table

Key terms:

- INSERT INTO statement
- DESCRIBE statement
- NULL value



2

aws re/start

In this module, you will learn how to:

- Create a new table in an established database
- Implement data types when you create a table



Activity: Discussion – Why Do Tables Need Rows?



4

Activity

- Discuss why you would need to insert a row into a table.



■ INSERT statement

INSERT

INSERT INTO statement

- This statement is fundamental to populating a database table with data.
- SQL INSERT statement is used to insert a single record or multiple records into a table.
- The SQL INSERT statement is referred to as a **data manipulation language (DML)** command.
- Note: The **order of the columns** is important when you select.



INSERT INTO statement

```
INSERT INTO tableName (col_1,col_2,col_3...col_n)
VALUES ("val_1","val_2","val_3","val_n");
```

7

aws re/start

When you insert a row, you must specify a column where the data will go. If the column is not specified, values are added to all the columns of the table, which results in adding a single row. When you insert values, you must enclose the values with single quotation marks (' ') for character or date values. If you insert values for one or more specific columns, then you must specify the names of the columns in structured query language (SQL).

SQL parameters

Name	Description
tableName	The table where data will be inserted
col_1,col_2,col_3,...	Each column of the table where the data is going
val_1,val_2,val_3,...	Values against each column

8

aws re/start

Remember:

- **Column** represents the field titles.
- **Values** represents the data that is being inserted into the fields.

Syntaxes for the INSERT statement

I. Syntax for inserting data into a single record in SQL

```
INSERT INTO tableName  
VALUES ("val_1","val_2","val_3","val_n"),  
       ("val_11","val_22","val_32","val_nn");
```

II. Syntax for inserting data into multiple records in SQL

```
INSERT INTO tableName (col_1,col_2,col_3...col_n)  
VALUES ("val_1","val_2","val_3","val_n"),  
       ("val_11","val_22","val_32","val_nn");
```

First, specify the **table name** and a list of **comma-separated columns** inside parentheses after the INSERT INTO clause. Then, put a **comma-separated list of values** of the corresponding columns inside the parentheses that follow the VALUES keyword.

You can use the INSERT statement in two ways. In the first example, when you add values for all the columns of the table, you do not need to specify the column names. In the second example, both the column name and the values are written. The number of columns and values must be the same. In addition, the positions of columns must correspond with the positions of their values.

The INSERT statement, continued

employees

employee id	family name	given name
EN1-10	wang	Xiulan
EN1-12	Ramirez	Diego
EN1-19	Major	Mary
EN1-22	Sarkar	Saanvi
EN1-27	Mansa	Akua
EN1-35	Jackson	Mateo

```
INSERT INTO employees  
VALUES ("EN1-35", "Jackson",  
"Mateo");
```

10

aws re/start

The example illustrates the syntax of the INSERT statement. This information is placed in the new row that was created.