CC105

Information Management

MySQL INSERT Statements

Objectives:

Understand the different types of MySQL insert statements. Learn the syntax and use cases for each insert method.

MySQL provides several ways to insert data into tables, each tailored to specific scenarios.

Whether you're inserting a single row, multiple rows, or bulk data, understanding these methods is crucial for efficient database management.

Basic INSERT INTO Statement

The INSERT INTO statement is the most straightforward way to add data to a table. It allows you to insert a single row or multiple rows at once.

Syntax

INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);

Example

```
INSERT INTO employees
(first_name, last_name, age,
department) VALUES ('John', 'Doe',
30, 'Sales');
```

When is the basic INSERT INTO statement sufficient?

How can you insert multiple rows in a single query?

Inserting Multiple Rows

Inserting multiple rows in a single query can significantly improve performance by reducing the number of round trips to the database.

Syntax

INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...),
(value4, value5, value6, ...),
(value7, value8, value9, ...);

Example

```
INSERT INTO employees
(first_name, last_name, age,
department) VALUES ('Jane', 'Smith',
25, 'HR'), ('Mike', 'Johnson', 40, 'IT'),
('Emily', 'Davis', 35, 'Finance');
```

INSERT ... SELECT

This method allows you to insert data into a table from the result of a SELECT query. This is particularly useful for copying data from one table to another.

Syntax

INSERT INTO table_name (column1, column2, column3, ...)
SELECT column1, column2, column3,
...
FROM another_table
WHERE condition;

Syntax

```
INSERT INTO former_employees
(first_name, last_name, age,
department)
SELECT first_name, last_name, age,
department
FROM employees
WHERE age > 60;
```

INSERT IGNORE

The INSERT IGNORE statement allows you to insert rows into a table while ignoring errors that would normally cause the query to fail, such as duplicate key errors.

Syntax

INSERT IGNORE INTO table_name
(column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);

Example

```
INSERT IGNORE INTO employees (first_name, last_name, age, department) VALUES ('John', 'Doe', 30, 'Sales');
```

LOAD DATA INFILE

For bulk data insertion, MySQL provides the LOAD DATA INFILE statement, which allows you to load data from a file into a table. This is much faster than inserting rows one by one.

Syntax

```
LOAD DATA INFILE 'file_path'
INTO TABLE table_name
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
(column1, column2, column3, ...);
```

Syntax

```
LOAD DATA INFILE
'/path/to/employees.csv'
INTO TABLE employees
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
(first_name, last_name, age,
department);
```

CC105

Information Management