

MySQL CASE 用法



如果源表和目标表的列名完全不同，你仍然可以通过指定列的名称和顺序来插入数据。SQL 中允许将不同名称的列匹配起来，只要确保源表中的列顺序与目标表中的列顺序相对应，并且数据类型兼容。

语法：

sql

```
INSERT INTO target_table (target_column1, target_column2, ...)
SELECT source_columnA, source_columnB, ...
FROM source_table
WHERE condition;
```

- ``target_table``：目标表的名称。
- ``target_column1, target_column2, ...``：目标表中要插入数据的列名。
- ``source_columnA, source_columnB, ...``：源表中对应的数据列。
- ``condition``：可选，用于限定要插入哪些记录。

示例：

假设有两个表，``old_employees`` 和 ``new_employees``，它们的列名称和结构如下：

- ``old_employees`` 表：
 - ``emp_id``
 - ``emp_name``
 - ``emp_age``
 - ``emp_department``
- ``new_employees`` 表：
 - ``id``
 - ``full_name``
 - ``age``
 - ``dept``

我们需要将 ``old_employees`` 表的数据插入到 ``new_employees`` 表中。虽然两个表的列名不同，但是可以通过明确指定列名来实现。

插入数据：

sql

```
INSERT INTO new_employees (id, full_name, age, dept)
SELECT emp_id, emp_name, emp_age, emp_department
FROM old_employees
WHERE emp_age > 30; -- 可以根据条件过滤要插入的数据
```

在这个例子中：

- `new_employees` 表的列 `id`, `full_name`, `age`, 和 `dept` 分别对应 `old_employees` 表的 `emp_id`, `emp_name`, `emp_age`, 和 `emp_department`。
- `SELECT` 子句中的列顺序与 `INSERT INTO` 子句中的目标列顺序严格对应。

注意事项：

1. **数据类型匹配**：确保目标表的列与源表的列数据类型兼容。例如，如果目标列是整数，源列也应该是整数或能够转换为整数的类型。
2. **列顺序匹配**：尽管列名不同，数据必须按顺序匹配。
3. **条件筛选**：可以使用 `WHERE` 子句来限制插入哪些记录。

插入带计算或常量值：

如果目标表中有一些列在源表中不存在，或者你想插入计算值或固定值，可以在 `SELECT` 子句中使用表达式或常量值。

例如，如果 `new_employees` 表有一列 `hire_date`，你想将所有记录的 `hire_date` 设置为当前日期：

sql

```
INSERT INTO new_employees (id, full_name, age, dept, hire_date)
SELECT emp_id, emp_name, emp_age, emp_department, CURRENT_DATE
FROM old_employees;
```

在这个例子中，`CURRENT_DATE` 是一个表达式，用于为所有新插入的记录指定当前日期作为 `hire_date`。

总结：

- 当源表和目标表列名不同时，通过显式指定目标表的列名和源表中的列顺序，可以插入数据。
- 数据类型和列顺序必须兼容，且可以使用常量值或表达式来填充不匹配的列。