Theory: Basic UPDATE statement

11 minutes 0 / 5 problems solved

Typically, we don't just store data: we need it to reflect the current state of things in real life as closely as possible. One can get a promotion at work, so their title needs to be changed. You can buy a gift for your friend and the amount of money in your bank account has to be adjusted accordingly. For cases like these, SQL has a special operation that helps to change values in cells of the already existing rows – UPDATE.

§1. General form

What information is necessary for making an update? Name of a **table** where we want to change data, **column name(s)** where the data resides and an **expression** to calculate a new value for each specified column:

```
UPDATE table_name SET col1 = expr1, col2 = expr2, …, colN = expr;
```

"Column name-expression" pairs are separated by commas. Generally, it is allowed to use any valid SQL expression. You can type a correct combination of literals, operators, functions and column references here, just remember about **type consistency**; trying to update an integer column with a text is never a good idea.

Imagine that you work as a developer for ABC Industries Ltd. The company has a lot of data and uses SQL to work with it. Information about their personnel is stored in a table named *employees*. For each employee there is a department id (integer), their last name (text), their salary (integer) and its upper limit (integer):

Table "employees"

department_id	last_name	salary	upper_limit
12	Johnson	80000	130000
78	Lee	90000	120000
34	Fowler	70000	140000
65	Owen	60000	110000

If for some reason all workers need to be moved to department #14, we could write the following:

```
UPDATE employees SET department_id = 14;
```

Since we used an integer value in the column *department_id* of integer type, the query is correct.

Here's what the table looks like after running the query:

Table "employees"

department_id	last_name	salary	upper_limit
14	Johnson	80000	130000
14	Lee	90000	120000
14	Fowler	70000	140000
14	Owen	60000	110000

§2. Column references

As mentioned earlier, new values don't have to be constant literals. Oftentimes, they are composed based on data that is already present in the table cells. Each column reference on the right side of the assignment represents the current value stored in the corresponding cell of a row.

Current topic:

Basic UPDATE statement

Topic depends on:

✕ Expressions

Topic is required for:

Updating selected rows

JDBC Statements

Table of contents:

↑ Basic UPDATE statement

§1. General form

§2. Column references

Feedback & Comments

What if we want to celebrate such a massive change in the company's structure and give our employees a raise?
Absolute values won't do here, so their current salaries should be used:

```
UPDATE employees SET salary = salary + 10000;
```

Table "employees"

department_id	last_name	salary	upper_limit
14	Johnson	90000	130000
14	Lee	100000	120000
14	Fowler	80000	140000
14	Owen	70000	110000

The addition of an integer value to an integer column produces a value of integer type as a result, which means that type consistency requirement is met.

Pay attention: during the execution of UPDATE, every row of a table is considered individually. If we want to use old value(s) to compute a new value for a cell, only cell(s) from the same row will be taken into account.

It's possible to update multiple columns simultaneously, so we can achieve the same result using only one query instead of two:

```
UPDATE employees SET department_id = 14, salary = salary + 10000;
```

Let's try to come up with something more elaborate: set new salaries to 80 percent of their upper limits and omit the fractional part that might appear. For the last part of the requirement, we can use the floor() function that takes a real value and returns an integer value.

```
UPDATE employees SET salary = floor(0.8 * upper_limit);
```

Table "employees"

department_id	last_name	salary	upper_limit
14	Johnson	104000	130000
14	Lee	96000	120000
14	Fowler	112000	140000
14	Owen	88000	110000

As you see, the update is a fairly simple yet useful operation. In practice, it often comes in handy.
Speaking of practice: are you ready for some tasks?

Report a typo

172 users liked this theory. 3 didn't like it. What about you?

Start practicing

Comments (4) Hints (0) Useful links (0) Show discussion