SQL

DML

dr Szymon Murawski

Comarch SA

November 19, 2019

Inserting rows into relation

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

```
INSERT INTO employees
  (employee_id, first_name, last_name, city, salary)
VALUES
  (6, 'Andy', 'Kauffman', 'Chicago', 300);

---
Query returned successfully: one row affected, 12 msec execution
    time.
```

Inserting rows into relation

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

Order of the columns can be changed, but also order of the attributes must change

```
INSERT INTO employees
  (salary, first_name, last_name, city, employee_id)
VALUES
  (300, 'Bill', 'Holgersson', 'Washington', 7);

---
Query returned successfully: one row affected, 12 msec execution
    time.
```

Inserting rows into relation

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

If column names are not specified, then it is assumed attributes are in the same order, as the columns in the database

```
INSERT INTO employees
VALUES
  (8, 'Paul', 'Doe', 'Los Angeles', 450);

---
Query returned successfully: one row affected, 12 msec execution
    time.
```

Inserting multiple rows into relation

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

```
INSERT INTO employees
VALUES
  (9, 'Jacob', 'Grimm', 'Berlin', 250),
  (10, 'Wilhelm', 'Grimm', 'Berlin', 250);

---
Query returned successfully: 2 rows affected, 12 msec execution
    time.
```

Inserting multiple rows into relation with SELECT

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

Instead of specifying VALUES, we can use SELECT to generate rows

```
INSERT INTO employees
SELECT
(SELECT MAX(employee_id) + 1 FROM employees) AS employee_id,
 'Jacob', 'Grimm', 'Berlin', 250

---
Query returned successfully: 1 row affected, 12 msec execution
    time.
```

Deleting rows from relation

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

```
DELETE FROM employess
WHERE employee_id = 5;

---
Query returned successfully: one row affected, 12 msec execution
    time.
```

Deleting rows from relation

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

If WHERE clause is omitted all rows are removed from table!

```
DELETE FROM employees;

---
Query returned successfully: 5 rows affected, 12 msec execution
    time.
```

Updating values in tables

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

```
UPDATE employees
SET salary = '150'
WHERE employee_id = 3

---
Query returned successfully: one row affected, 12 msec execution
    time.
```

Updating values in tables

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

If WHERE clause is omitted all rows are affected by the update!

```
UPDATE employees
SET salary = '150'

---
Query returned successfully: 5 rows affected, 12 msec execution
    time.
```

Updating multiple values in tables

Employees

employee_id	first_name	last_name	city	salary
1	John	Smith	New York	150
2	Ben	Johnson	New York	250
3	Louis	Armstrong	New Orleans	75
4	John	Lennon	London	300
5	Peter	Gabriel	London	100

If WHERE clause is omitted all rows are affected by the update!

```
UPDATE employees
SET salary = '3000', city = 'Miami'
WHERE employee_id = 1

---
Query returned successfully: one row affected, 12 msec execution
    time.
```