



SQL

Data Manipulation Language

André Restivo

Index

Introduction

Inserting

Deleting

Updating

Introduction

SQL

- Structured Query Language.
- A special purpose language to manage data stored in a **relational** database.
- Based on **relational algebra**.
- Pronounced *Sequel*

History

- Early 70's SEQUEL Developed at IBM
- 1986 SQL-86 and SQL-87 Ratified by ANSI and ISO.
- 1989 SQL-89
- 1992 SQL-92 Also know as SQL2.
- 1999 SQL:1999 Also known as SQL3 Includes regular expressions, recursive queries, triggers, non-scalar data types and some object-oriented expressions.
- 2003 SQL:2003 XML support and auto-generated values.
- 2006 SQL:2006 XQuery support.
- 2008 SQL:2008.
- 2011 SQL:2011.

Standard

- Although SQL is an ANSI/ISO standard, every database system implements it in a slightly different way.
- These slides will try to adhere to the standard as much as possible.
- Sometimes we'll deviate and talk specifically about PostgreSQL.

Inserting

Inserting

To insert values into a table we use the **INSERT** command:

```
INSERT INTO <tablename> (<col1>, <col2>, ...) VALUES (<val1>, <val2>, ...);
```

Example:

```
INSERT INTO employee (id, name, salary) VALUES (1, 'John Doe', 1000);
```


Inserting

We can omit the column names if we insert the values in the same order we used to create the table columns.

```
INSERT INTO employee VALUES (1, 'John Doe', 1000);
```

Deleting

Deleting

- To delete value from a table, we use the **DELETE** command.
- The delete command can receive a **condition** specifying which rows to delete.
- If no **condition** is given **all rows are deleted** from the table.
- If there are foreign keys, the rule set by the **ON DELETE** clause is used.
- The condition can be as complex as those used on the **SELECT** command.

```
DELETE FROM <tablename> WHERE <condition>;
```

Examples

Delete employee with id 1

```
DELETE FROM employee WHERE id = 1;
```

Delete all employees

```
DELETE FROM employee;
```

Delete employees with salary larger or equal to 1200

```
DELETE FROM employee WHERE salary >= 1200;
```

Updating

Updating

- To modify values from a table, we use the **UPDATE** command.
- The update command can receive a **condition** specifying **which** rows to update.
- If **no condition** is given **all rows** are **updated** from the table.
- If there are foreign keys, the rule set by the **ON UPDATE** clause is used.
- The condition can be as complex as those used on the **SELECT** command.
- New values can be calculated on the fly.

```
UPDATE <tablename> SET <col1> = <val1>, <col2> = <val2>, ... WHERE <condition>;
```

Examples

Change the salary of employee 1 to 1500

```
UPDATE employee SET salary = 1300 WHERE id = 1;
```

Increase the salary of employee 1 by 10%

```
UPDATE employee SET salary = salary * 1.1 WHERE id = 1;
```

Increase the salary of all employees by 10%

```
UPDATE employee SET salary = salary * 1.1;
```

Decrease the salary and taxes of all employees with a salary larger than 1200 by 10%

```
UPDATE employee SET salary = salary * 0.9, taxes = taxes * 0.9  
WHERE salary > 1200;
```