

# SQL - Structured Query Language

# Определение

- **SQL (structured query language — язык структурированных запросов)** — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД).
- Язык SQL представляет собой совокупность:
  1. Операторов;
  2. Инструкций;
  3. Вычисляемых функций.

# Стандарты SQL

Год	Название	Изменения
1986	SQL-86 (SQL-87)	Первый вариант стандарта, принятый институтом ANSI и одобренный ISO в 1987 году.
1989	SQL-89 (FIPS 127-1)	Немного доработанный вариант предыдущего стандарта.
1992	SQL-92 (SQL2, FIPS 27-2)	Значительные изменения (ISO 9075); уровень Entry Level стандарта SQL-92 был принят как стандарт FIPS 127-2.
1999	SQL:1999 (SQL3)	Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, нескаллярные типы данных и некоторые объектно-ориентированные возможности.
2003	SQL:2003	Введены расширения для работы с XML-данными, оконные функции (применяемые для работы с OLAP-базами данных), генераторы последовательностей и основанные на них типы данных.
2006	SQL:2006	Функциональность работы с XML-данными значительно расширена. Появилась возможность совместно использовать в запросах SQL и XQuery.
2008	SQL:2008	Улучшены возможности оконных функций, устранены некоторые неоднозначности стандарта SQL:2003

# Группы операторов SQL

- Операторы определения данных (Data Definition Language, **DDL**): **CREATE, ALTER, DROP**;
- Операторы манипуляции данными (Data Manipulation Language, **DML**): **SELECT, INSERT, UPDATE, DELETE**;
- Операторы управления транзакциями (Transaction Control Language, **TCL**): **COMMIT, ROLLBACK, SAVEPOINT**;
- Операторы определения доступа к данным (Data Control Language, **DCL**): **GRANT, REVOKE, DENY**.

# DDL: CREATE

<https://www.postgresql.org/docs/9.5/static/sql-createtable.html>

```
CREATE TABLE test (  
  id INTEGER NOT NULL,  
  id_category SMALLINT NOT NULL,  
  first_name VARCHAR(50) NOT NULL,  
  middle_name VARCHAR(50),  
  last_name VARCHAR(50) NOT NULL,  
  age SMALLINT DEFAULT 20 NOT NULL,  
  created TIMESTAMP(0) WITHOUT TIME ZONE DEFAULT now() NOT NULL,  
  updated TIMESTAMP(0) WITHOUT TIME ZONE,  
  active BOOLEAN DEFAULT true NOT NULL,  
  CONSTRAINT test_pkey PRIMARY KEY(id),  
  CONSTRAINT test_fk FOREIGN KEY (id_category)  
    REFERENCES category(id)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
    NOT DEFERRABLE  
)  
WITH (oids = false);
```

# DDL: ALTER

<https://www.postgresql.org/docs/9.5/static/sql-altertable.html>

**ALTER TABLE** test ADD COLUMN address varchar(30);

**ALTER TABLE** test RENAME COLUMN address TO city;

**ALTER TABLE** test DROP COLUMN city RESTRICT;

**ALTER TABLE** test RENAME TO suppliers;

# DDL: DROP

<https://www.postgresql.org/docs/9.5/static/sql-droptable.html>

**DROP TABLE suppliers**

**DROP TABLE IF EXISTS suppliers**

**DROP TABLE IF EXISTS suppliers CASCADE**

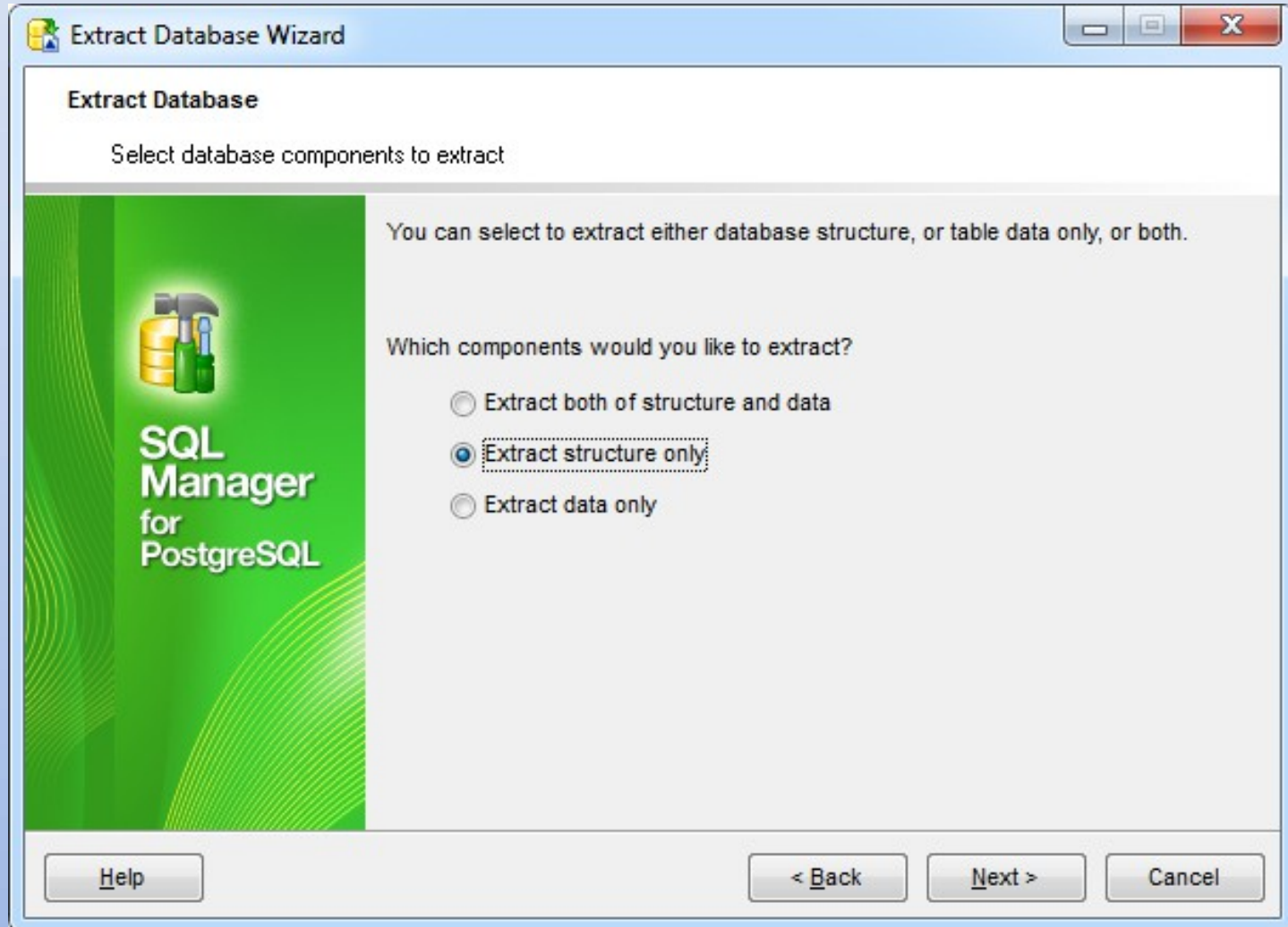
**DROP TABLE suppliers CASCADE**

Кроме создания, изменения структуры и удаления таблиц операторы CREATE, ALTER, DROP используются для других объектов базы данных:

CREATE DATABASE, CREATE FUNCTION, CREATE INDEX, CREATE MATERIALIZED VIEW, CREATE ROLE, CREATE SCHEMA, CREATE SEQUENCE, CREATE **TABLE**, CREATE TABLESPACE, CREATE TRIGGER, CREATE TYPE, CREATE USER, CREATE VIEW, и другие.

<https://www.postgresql.org/docs/9.5/static/bookindex.html>

# Extract database





# DML: INSERT

<https://www.postgresql.org/docs/9.5/static/sql-insert.html>

```
INSERT INTO contact VALUES (0, 1, 'Дмитрий', 'Дмитриевич', 'Дмитров', 22, now(), NULL, TRUE);
```

```
INSERT INTO contact VALUES (nextval('contact_seq'), 1, 'Илья', 'Ильич', 'Ильин', 22, now(), NULL, TRUE);
```

```
INSERT INTO contact VALUES (nextval('contact_seq'), 1, 'Дмитрий', 'Дмитриевич', 'Дмитров', 19, '10/26/2016 11:12:01 AM'::TIMESTAMPTZ, NULL, TRUE);
```

```
INSERT INTO contact (id, id_category, first_name, middle_name, last_name) VALUES (nextval('contact_seq'), 1, 'Дмитрий', 'Дмитриевич', 'Дмитров');
```

```
INSERT INTO contact VALUES (nextval('contact_seq'), 1, 'Дмитрий', 'Дмитриевич', 'Дмитров', DEFAULT, DEFAULT, DEFAULT, DEFAULT);
```

```
INSERT INTO contact (id, id_category, first_name, middle_name, last_name) VALUES (nextval('contact_seq'), 1, 'Дмитрий', 'Дмитриевич', 'Дмитров'), (nextval('contact_seq'), 1, 'Василий', 'Васильевич', 'Васильев'), (nextval('contact_seq'), 1, 'Анатолий', 'Анатолевич', 'Анатолев');
```

*Возвращает количество вставленных записей*

# DML: UPDATE

<https://www.postgresql.org/docs/9.5/static/sql-update.html>

**UPDATE** contact SET id\_category = 2 WHERE id > 3

**UPDATE** contact SET id\_category = 2, updated = now(), active = false WHERE id > 3

**UPDATE** contact SET created = DEFAULT, updated = now(), active = false WHERE id > 3

**UPDATE** contact SET first\_name = UPPER(first\_name), middle\_name = UPPER(middle\_name),  
last\_name = UPPER(last\_name) WHERE id > 3

**Возвращает количество обновленных записей**

# DML: DELETE

<https://www.postgresql.org/docs/9.5/static/sql-delete.html>

**DELETE FROM** contact WHERE id > 3

**DELETE FROM** contact -- Удалит все записи

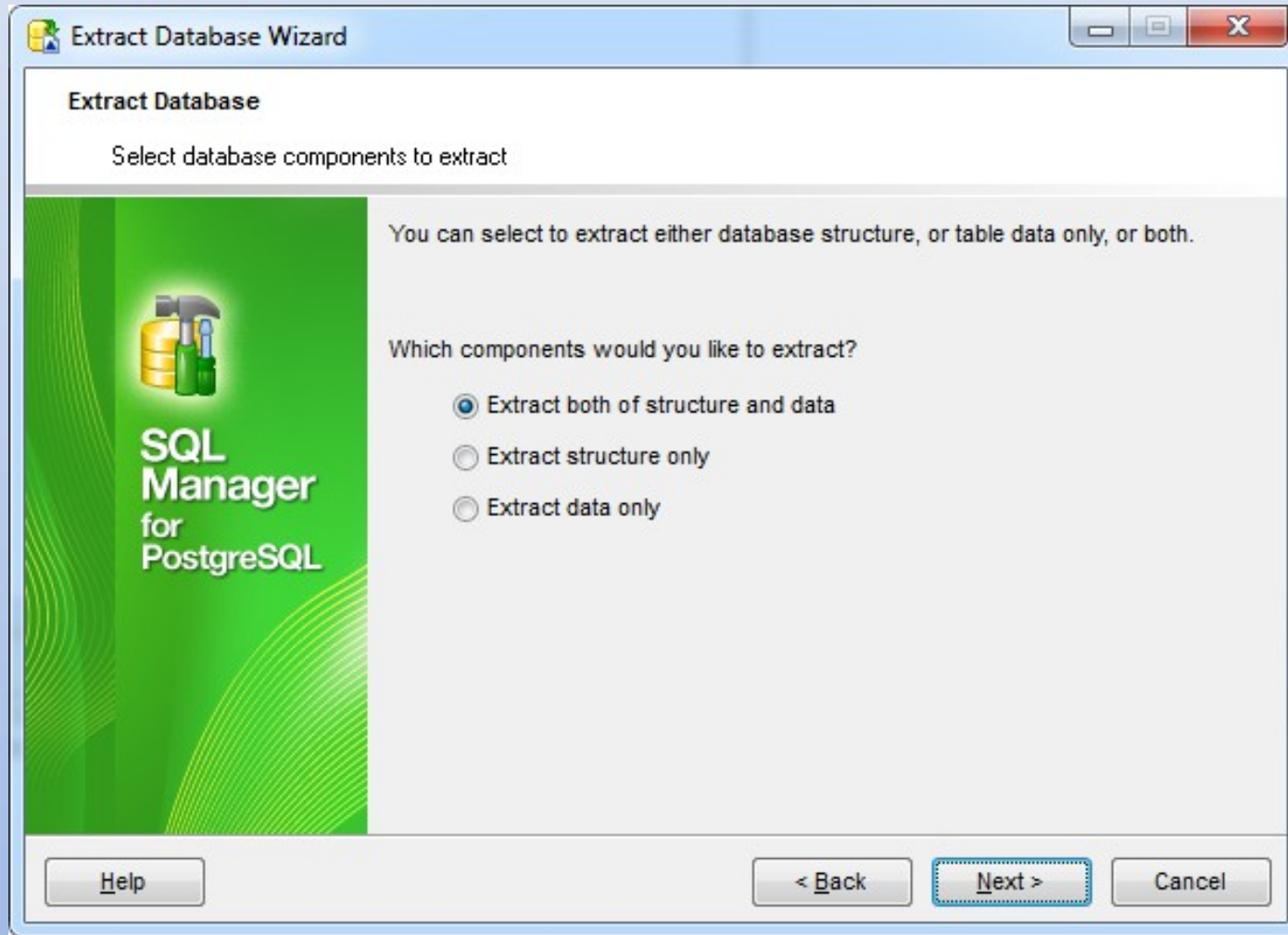
*Возвращает количество удаленных записей*

**FYI:**

Для удаления всех строчек из таблице с большим количеством записей рекомендуется использовать TRUNCATE, которая появилась в стандарте SQL:2008 – на больших объемах данных выполняется быстрее, за счет того, что удаляется вся таблица и данная команда не является транзакционной (см. дальше):

**TRUNCATE** contact CASCADE;

# Extract database with data



# Выводы

1. **SQL (structured query language** — язык структурированных запросов) — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД);
2. Группы операторов SQL:
  - a) Операторы определения данных (Data Definition Language, **DDL**): **CREATE, ALTER, DROP**;
  - b) Операторы манипуляции данными (Data Manipulation Language, **DML**): **SELECT, INSERT, UPDATE, DELETE**;
  - c) Операторы управления транзакциями (Transaction Control Language, **TCL**): **COMMIT, ROLLBACK, SAVEPOINT**;
  - d) Операторы определения доступа к данным (Data Control Language, **DCL**): **GRANT, REVOKE, DENY**.
3. Операторы **DDL** позволяют создавать таблицу (CREATE), изменять ее структуру (ALTER) и удалять таблицу вместе с данными (DROP);
4. Операторы **DML** позволяют манипулировать с данными: добавлять новые записи в таблицу (INSERT), модифицировать поля таблицы (UPDATE) и удалять данные из таблицы (DELETE) (В отличие от DROP удаляются только данные, структура таблицы остается);
5. Используя язык SQL можно создавать dump базы данных и переносить данные с одного сервера СУБД на другой сервер СУБД.
6. Операторы SELECT, DCL и TCL будут рассмотрены на следующих занятиях.