# SQL

## DDL

dr Szymon Murawski

Comarch SA

December 3, 2019

# Creating new tables

```
CREATE TABLE students (
id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE);
```

```
SELECT *
FROM students

id | first_name | last_name | birthdate
---+------------+-----------+----------

(0 rows)
```

# Numeric data types in Postgres

- **INTEGER**
- **REAL**
- **DOUBLE PRECISION**
- **NUMERIC(precision, scale)** - `precision` is the number of digits in the whole number, `scale` is number of digits after the decimal point
- **DECIMAL(precision, scale)** - same as NUMERIC
- **SERIAL** - an autoincrementing integer field
- **MONEY** - for storing money values, that will be displayed using locale settings

# Text and date types

- **CHAR(n)** - stores text of length `n`. It is fixed-length column, if text is shorter then it is filled with gaps
- **VARCHAR(n)** - stores text of length `n`. It is variable-length column, shorter texts will occupy less memory space
- **TEXT** - stores text of any length

- **TIME**
- **DATE**
- **TIMESTAMP** - date and time
- **INTERVAL** - difference between two timestamps

- **BOOLEAN**

# Notation of values in data types

- `INTEGER`, i.e. 123
- `REAL, DOUBLE PRECISION`, i.e. 12.34
- `NUMERIC(5,3)`, i.e. 12.345
- `DATE`, i.e. '2019-01-01'
- `TIME`, i.e. '12:45:01'
- `TIMESTAMP`, i.e. '2019-11-25 13:11:40'
- `INTERVAL`, i.e. '1 year 2 months 3 days 4 hours 5 minutes 6 seconds'
- `BOOLEAN`, i.e. true, false

# Default values

```
CREATE TABLE students (
id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE,
active BOOLEAN DEFAULT true,
date_added TIMESTAMP DEFAULT now());
```

```
SELECT *
FROM students

id | first_name | last_name | birthdate | active | date_added
---+------------+-----------+-----------+--------+-----------

(0 rows)
```

# Default values

```
INSERT INTO students
VALUES (1, 'John', 'Smith', '2000-01-01');

SELECT *
FROM students

id | first_name | last_name | birthdate  | active | date_added
---+------------+-----------+------------+--------+-----------
1  | John       | Smith     | 2000-01-01 | true   | 2019-12-01
(1 row)
```

# Autoincrement column

```
CREATE TABLE students (
id SERIAL,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE,
active BOOLEAN DEFAULT true,
date_added TIMESTAMP DEFAULT now());
```

```
INSERT INTO students(first_name, last_name, birthday)
VALUES ('John', 'Smith', '2000-01-01');

SELECT *
FROM students

id | first_name | last_name | birthdate  | active | date_added
---+------------+-----------+------------+--------+-----------
1  | John       | Smith     | 2000-01-01 | true   | 2019-12-01
(1 row)
```

# Autoincrement column #2

```
CREATE SEQUENCE students_id_seq;
CREATE TABLE students (
id INTEGER DEFAULT nextval(students_id_seq),
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE,
active BOOLEAN DEFAULT true,
date_added TIMESTAMP DEFAULT now());
```

```
INSERT INTO students(first_name, last_name, birthday)
VALUES ('John', 'Smith', '2000-01-01');

SELECT *
FROM students

id | first_name | last_name | birthdate  | active | date_added
---+------------+-----------+------------+--------+-----------
1  | John       | Smith     | 2000-01-01 | true   | 2019-12-01
(1 row)
```

# Primary keys

## Method 1

```
CREATE TABLE students (
id SERIAL PRIMARY KEY,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE);
```

## Method 2

```
CREATE TABLE students (
id SERIAL,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE,
PRIMARY KEY(id));
```

# Removing table

```
CREATE TABLE students (
id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE);

DROP TABLE students;
```

```
SELECT *
FROM students;

ERROR:  relation "students" does not exists
```

# Altering table - columns

```
CREATE TABLE students (
id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE);

ALTER TABLE students ADD COLUMN address VARCHAR(30);
```

```
SELECT *
FROM students

id | first_name | last_name | birthdate | address
---+------------+-----------+-----------+---------

(0 rows)
```

# Altering table - columns #2

```
CREATE TABLE students (
id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE);

ALTER TABLE students DROP COLUMN birthdate;
ALTER TABLE students RENAME COLUMN last_name TO surname;
```

```
SELECT *
FROM students

id | first_name | surname
---+------------+---------

(0 rows)
```

# Altering columns

```
CREATE TABLE students (
id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
birthdate DATE);

ALTER TABLE distributors
    ALTER COLUMN birthdate TYPE TIMESTAMP,
    ALTER COLUMN birthdate SET DEFAULT now();
```

```
SELECT *
FROM students

id | first_name | last_name | birthdate
---+------------+-----------+----------

(0 rows)
```

# Information schema

- Information schema is SQL standard way of storing metadata information about database
- It consist of several tables, such as:
  - information_schema.tables
  - information_schema.columns
  - information_schema.table_constraints
- In PostgreSQL there is also pg_catalog schema available, that also stores database metadata, but organized in a different way

# Displaying tables metadata

```
SELECT table_catalog, table_schema, table_name FROM
    information_schema.tables;

table_catalog | table_schema | table_name
--------------+--------------+-----------
rentals       | public       | movies
rentals       | public       | actors
rentals       | pg_catalog   | pg_authid
...
```