CMPT 321
FALL 2017

PostgreSQL

Lecture 05.01

By Marina Barsky

PostgreSQL

- Powerful database management system
- Open source, originally developed at the University of California at Berkeley CS Department
- Pioneered many concepts that only became available in some commercial database systems much later
- Because of the liberal license, PostgreSQL can be used, modified, and distributed by anyone free of charge for any purpose, be it private, commercial, or academic

2-tier client-server architecture

The Postgre DBMS software is running on Database server.

Your interaction with database consists of 2 processes:

- A server process: manages the database files, maintains connection pool, performs database actions on behalf of clients
- The client (frontend) application: a text-oriented tool, a graphical application, a web server that accesses the database to display web pages, or a specialized database maintenance tool

Note: The client and the server can be on different hosts. They communicate over a TCP/IP network connection. The files that can be accessed on a client machine might not be accessible on the database server machine.

PostgreSQL – SQL standards

- PostgreSQL supports most of the major features of SQL:2003. (No current version of any database management system claims full conformance to Core SQL:2003).
- Out of 164 mandatory features required for full Core conformance, PostgreSQL conforms to at least 150.
- In addition, there is a long list of supported optional features.

SQL syntax is very similar to MySQL and Oracle

How to connect to PostgreSQl server

- Using your SSH tool, ssh to *src-code.simons-rock,edu*
- PostgreSQL is installed
- If you want to create your own database instance, please contact me after this lecture
  - Do it only if you are really planing to use PostgreSQL or follow the examples in the lectures

Interactive shell client

- Connect to your specific database:

```
psql db_name
```

- You see the following prompt:

```
db_name=>
```

- You are now connected and you can enter sql commands

Schema in PostgreSQL

- A database contains one or more named schemas, which in turn contain tables
- To create or access objects in a schema, write a *qualified name* consisting of the schema name and table name separated by a dot:

schema.table

- There is a default schema called *public*, for which you don't need to specify the qualified name, only the name of the table

Documentation: http://www.postgresql.org/docs/9.1/static/ddl-schemas.html

Creating schema in Postgre

```
DROP SCHEMA IF EXISTS movies_db CASCADE;
CREATE SCHEMA movies_db;
SET SEARCH_PATH TO movies_db;
```

- Now you can use regular syntax without prefixing each object by movies_db

Some useful commands

- To execute sql script in file *moviesdb_postgre.sql*

```
\i moviesdb_postgre.sql
```

- to quit database shell:

```
\q
```

- To change current schema (to avoid typing qualifying schema name):

```
alter role mgbarsky set search_path = 'pizza','public';
```

Main data types

- **NUMERIC (precision, scale) :**
  - scale - count of decimal digits in the fractional part, to the right of the decimal point.
  - precision - the total count of significant digits in the whole number
- **CHAR(n)** allocates a fixed space, and if the string that we store is shorter than **n**, then it is padded with blanks.
- Differently, **VARCHAR(n)** denotes a string of up to n characters.
- CHAR has better performance. Use CHAR(n) for frequently used fields, and use VARCHAR(n) otherwise.
- **DATE**: Default date format: '1994-11-28'

Additional data types

- **INT**
- **BOOLEAN**
- **ENUM**

```
CREATE TYPE mood AS ENUM ('sad', 'ok', 'happy');
CREATE TABLE person (
    name text,
    current_mood mood
);
```

And see more here:
https://www.techonthenet.com/postgresql/datatypes.php

Getting information about tables

- Describe all tables:

\dt

```
  List of relations
 Schema |    Name    | Type  |  Owner
--------+------------+-------+----------
 public | movie      | table | mgbarsky
 public | movie_exec | table | mgbarsky
 public | movie_star | table | mgbarsky
 public | starsin    | table | mgbarsky
 public | studio     | table | mgbarsky
(5 rows)
```

Describe columns of table *movie*

\d+ movie;

```
 Table "public.movie"
  Column      |       Type          | Modifiers | Storage
-------------+---------------------+-----------+----------+-------------
 title        | character varying(30) | not null  | extended |
 year         | integer             | not null  | plain    |
 length       | integer             |           | plain    |
 incolor      | integer             |           | plain    |
 studioname   | character varying(20) |         | extended |
 producerc    | character varying(3)  |         | extended |
Indexes:
    "movie_pkey" PRIMARY KEY, btree (title, year)
```