



DIUM, University of Udine

Basi di dati

SQL and Postgres

Andrea Brunello

`andrea.brunello@uniud.it`



Database design

The process of designing a database is typically articulated into different phases:

- ① Brainstorming meetings with IT personnel and all interested stakeholders
 - Collection of requirements
 - Design of a conceptual model (e.g., E-R model)
 - The E-R model has a specific notation, the *E-R diagram*, that helps all involved parts to discuss about the future database
- ② Translation of the conceptual model into a logical model
 - Typically, a set of translation rules is followed
 - At this stage, a specific DBMS technology must be chosen (e.g., relational DB)
- ③ Addition to the logical model of details regarding the physical, low-level aspects (e.g., usage of indexes)
 - The physical schema is thus obtained



- SQL (Structured Query Language) is the (largely) most popular language used to interact with relational databases
- SQL is a very rich language, and it encompasses several functionalities:
 - DDL, Data Definition Language
 - definition and update of table schemas
 - definition of integrity constraints
 - DML, Data Manipulation Language
 - insertion, update, deletion of tuples
 - Administrative stuff
 - management of data access criteria
 - management of concurrent access
 - maintenance of the database



Introduction (Cont.)

- SQL was born in the '70s, with the development of language *Sequel* by IBM, within the *System R* project
- Standard ANSI and ISO:
 - first version, SQL-86
 - latest version, SQL:2019
- Relational DBMSs offer (almost) all functionalities defined in standard SQL-92, other than (typically) several other proprietary features
- Actually, SQL implementations offered by different DBMS vendors are slightly different from one another, at least for what concerns advanced queries
- This hinders code interoperability



- In this course, we will focus on the SQL language version offered by **PostgreSQL** (<https://www.postgresql.org/>)
- PostgreSQL (aka Postgres) is a **free and open-source** relational database management system (RDBMS)
- It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users
- It is the default database for macOS Server and is also available for Windows, Linux, FreeBSD, and OpenBSD



PostgreSQL

A little history

- PostgreSQL evolved from the *Ingres* project at the University of California, Berkeley, led by Michael Stonebraker (Turing Award in 2014)
- *POSTGRES*, a new project following Ingres, ran from 1986 to 1994, with the first public version of the system released in June 1989
- The project was then taken by two Berkeley graduate students, Andrew Yu and Jolly Chen, and finally renamed PostgreSQL in 1996
- Since then, developers and volunteers around the world have maintained the software as *The PostgreSQL Global Development Group*
- The project continues to make releases available under its free and open-source software PostgreSQL License



- PgAdmin is a multiplatform client application that allows users to connect to a PostgreSQL database running on a server (either a local or a remote one)
- It allows one to manage several database administration tasks by means of an intuitive graphical user interface
- SQL code can also be written and run in a dedicated console, to allow for a fine-grained control over the operations
- The first prototype of the software, named *pgManager*, dates back to 1998; current version is pgAdmin 4, written in Python language (<https://www.pgadmin.org/>)



Let us now download and install PostgreSQL on our machines!

- In the next part of the course, we will learn SQL language by directly using it in PgAdmin, implementing the logical schema of University DB of Exercise 2
- All the commented code is available on the GitHub page of the course (file *SQL_code_lecture.sql*)