ISEL

MERCM – ADEETC

AMD (MLDM – Machine Learning and Data Mining**) – Module of Practice**

Paulo Trigo Silva

1. Install the PostgreSQL object-relational DBMS

The software to install (for the MS Windows, 32bit platform) is already available within the "`distribution_01_PostgreSQL.zip`" (cf., "moodle", folder "`Aulas Práticas`"). Nevertheless, the steps for the download and installation of PostgreSQL 10.10 are as follows:

a) If you do not want to use the "`distribution*.zip`", then go to the PostgreSQL site:

`http://www.enterprisedb.com/products-services-training/pgdownload`

select version and operating system; I chose version "PostgreSQL 10.10" and "Windows x86-32"".

b) To install MS-Windows version execute file; keep all components selected. This may take a while!

Important: do not loose the information that you are provided with during the installation process; credentials, port, database name.

2. Explore additional PostgreSQL tools

Execute the "*Application Stack Builder*" (start \ PostgreSQL 10 \ Application Stack Builder).

a) Select the PostgreSQL server and then explore the available applications.

b) Note the "Database Drivers" and "Web Development" for, respectively, JDBC and PHP drivers. You do not need to install them for this course but they may be useful for other course(s)!

3. Test the installation using the graphical administration tool

Execute the application "*pgAdmin 4*" (start \ PostgreSQL 10 \ pgAdmin 4).

a) Connect to the server (double-click \ Servers).

b) Create a new database "`my_db`" (PostgreSQL 10 \ Databases \ right-click \ Create \ Database).

c) Create a new table, named "`t1`" in the public schema of "`my_db`" database (my_db \ Schemas \ public \ Tables \ right-click). Take a look at the <SQL> tab that is always available.

d) Create a column, of "`t1`", named "`c1`" (my_db \ Schemas \ public \ Tables \ t1 \ Columns \ right-click); set "`c1`" of type *integer* and "*not null*". Take a look at the <SQL> tab that is always available.

e) Create a primary key constraint on "`c1`" (my_db \ Schemas \ public \ Tables \ t1 \ Properties \ Constraints). Set the constraint name as "PK_T1".

f) Create another column, of "`t1`", named "`c2`" of type "*character varying(20)*".

g) Populate "`t1`" with tuples (my_db \ Schemas \ public \ Tables \ t1 \ right-click \ View/Edit Data).

h) Generate the "`t1`" create table script (my_db \ Schemas \ public \ Tables \ t1 \ right-click \ Scripts) and copy the script into a text file. You may need to use it later (instead of this interactive process!)

i) Drop the "`t1`" table (my_db \ Schemas \ public \ Tables \ t1 \ right-click \ Delete/Drop…).

4. Manipulate the DBMS using “command line” tools

Consider the “`d01_postgresql-10-A4.pdf`” document.

a) Follow and execute the statements described in section 1.3 (Creating a Database”). Notice that the commands (`createdb`, `dropdb`, etc) are in the `bin` installation folder; e.g., the `createdb` command can be executed as `XX\PostgreSQL\10\bin\createdb –U postgres my_db` whenever the installation folder is `XX`, user name is `postgres` and database to create is `my_db`.

b) Follow and execute the statements described in the section 1.4 (Accessing a Database”). Note that the “`psql`” tool can be launched from: start \ All Programs \ PostgreSQL 10 \ SQL Shell (psql)

c) Read the sections 5.3.4 and 5.5.3 and set a primary key for the table you have just created.

d) Read the section 5.3.5 about the foreign key constraint and specify a “on delete cascade” behavior (using the tables you have just created).

e) If you need to recall the SQL language explore sections 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9.

5. Automate the database construction (and destruction)

Make sure that your database, named “`my_db`”, already exists and contains some tables that you would like to preserve when creating a new database. The “`my_db`” will be used as a template for the creation of a new database, named “`db_operational`”. Consider the “`scripts`” folder.

a) Open the “`_go.bat`” script and adapt the path (`psqlPath` variable in the script).

b) Open the “`00_script_CREATE_DB.txt`” file and analyze the code.

c) Execute the “`_go00.bat`” script. Go to “*pgAdmin 4*” and analyze the created database.

6. Build the conceptual and logical model and populate database

a) Draw an entity-relationship diagram that could have originated that schema (R1, R2, R3, R4).

b) Open the “`01_script_CREATE_SCHEMA.txt`” file and analyze the code.

c) Execute the “`_go01.bat`” script. Go to “*pgAdmin 4*” and analyze the created schema. Change “`DROP TABLE`” order (“`01_script_CREATE_SCHEMA`”) and execute “`_go01.bat`”.

d) Open “`02_script_POPULATE_SCHEMA.txt`” and add code that enables to remove comments.

e) Extend the code to populate each table with, at least, 4 tuples.

f) Execute the “`_go02.bat`” script. Go to “*pgAdmin 4*” and analyze the populated schema.

g) Open “`03_script_CREATE_VIEW.txt`” and create a view that involves 3 tables. Execute the “`_go03.bat`” script. Go to “*pgAdmin 4*” and analyze the created view.

h) Complete “`10_script_DELETE_DATA.txt`” to delete all data in all tables.

i) Complete “`20_script_DROP_TABLE.txt`” to first drop all constraints and then drop all tables.