O objetivo é criar a base de dados da universidade, começando por testar a manipulação básica do modelo de uma base de dados.

Scroll down for the english version

A documentação PostgreSQL está disponível em: http://www.postgresql.org/docs/

Escolha a documentação correspondente à versão que tem instalada. Esta ficha vai usar a versão 9.x, em http://www.postgresql.org/docs/9.6/static/index.html

1. Entre na base de dados que criou com o cliente "psql"

Alguns commandos: liste todas as tabelas existentes:

```
bd=> \d
```

crie uma nova tabela:

```
bd=> create table disciplina (id serial, numero int, nome text, descricao text, ects int, primary key(id));

-- serial: contador auto-incrementado

-- text: texto de comprimento variável

-- int: número inteiro em 4-bytes

        veja os tipos em http://www.postgresql.org/docs/9.1/static/datatype-numeric.html

-- date: data no formato ISO (aaaa-mm-dd)

-- para outros tipos de dados, veja: http://www.postgresql.org/docs/9.1/static/datatype.html

bd=> create table aluno (id serial, numero int, nome text, apelido text, data_nascimento date, primary key(id));
```

Veja a estrutura da tabela

```
bd=>\d disciplina
```

2. Insira dados na tabela (por enquanto evite usar caracteres acentuados e cedilhados)

Insira linhas na tabela:

```
bd=>insert into disciplina (nome, descricao, ects) values ('Bases de Dados', 'Fundamentos dos SGBD', 5);

bd=>insert into aluno (numero, nome, apelido, data_nascimento) values (123, 'paulo', 'moreira', '1992-01-01');

-- repare que o ID não é inserido, mas é automaticamente gerado

-- repare nas plicas ' para os valores do tipo TEXT e DATE
```

-- insira mais linhas, com valores diferentes

bd=>select * from aluno;

3. Apague uma ou duas linhas. Tem de escolher um valor que permita identificar a linha sem margem para dúvida:

bd=>delete from aluno where numero=123;

-- todos os alunos com numero=123 serão apagados. Em princípio queremos que seja só um aluno.

-- liste agora os que ficaram, ordenados por "numero"

bd=>select * from aluno order by numero;

4- Atualize um valor numa das linhas (supondo que tem um aluno com o numero 456. Senão use outro):

bd=>update aluno set nome='paula' where numero=456;

bd=>select * from aluno order by numero;

5- Liste apenas algumas linhas:

bd=>select * from aluno where data_nascimento > '1992-01-01';

-- mude o critério de restrição da data para incluir mais ou menos linhas

6- Vamos alterar a estrutura da tabela. Juntamos uma coluna para um número de curso:

bd=>alter table aluno add column curso integer;

7- Veja a nova estrutura da tabela:

bd=>\d aluno

8- Preencha valores de curso diferentes nas linhas que tiver

9- Guarde a base de dados num ficheiro: pg_dump

10- Apague a tabela (e o seu conteúdo):

bd=>drop table aluno;

11- Abra o ficheiro que criou em 9). Use o Notepad++, NoteTab, ou outro editor. Junte mais linhas de "insert" (faça copiar/colar e mude alguns valores para ficarem diferentes).

12- Volte a criar o conteúdo da base de dados a partir do ficheiro que criou em 9) e modificou em 11)

13- Faça variantes de interrogações. Alunos do curso X, alunos com número Y, alunos que nasceram antes ou depois de uma data data.

Utilize uma tabela que tenha criado.

1. Adicione restrições do tipo:

NOT NULL, UNIQUE, CHECK, ON DELETE RESTRICT, ON DELETE CASCADE

a) Verifique o efeito de cada uma dessas restrições, tentando ultrapassá-las e verificando os erros.

2. Utilize ALTER TABLE para adicionar colunas, remover colunas, adicionar restrições, remover restrições, renomear colunas, mudar o tipo de uma coluna.

3. Utilize o valor por omissão de uma coluna para especificar a marca temporal (timestamp) em que uma linha foi inserida

4. Considere os privilégios SELECT, INSERT, UPDATE, DELETE. Utilize o comando GRANT para restringir o acesso com estas operações de um dado utilizador a uma dada tabela.

5. Crie triggers para situações de inserção/atualização, testando as verificações linha a linha, e por consulta.

Alguns exemplos: um jogador só pode jogar se estiver convocado para essa edição do mundial (FIFA), uma equipa não pode jogar contra ela própria, um docente não pode estar em duas salas diferentes no mesmo horário, um aluno só se pode inscrever em disciplinas do seu curso,...

The goal is to create the university database, starting with basic schema manipulation commands.

Documentation PostgreSQL at: http://www.postgresql.org/docs/

check the docs online for your database version

1. Use the psql client

list all tables of your database:

bd=> \d

create a new table:

bd=> create table disciplina (id serial, numero int, nome text, descricao text, ects int, primary key(id));

-- serial: counter, auto-increment

-- text: variable length text

-- int: 4-byte integer

see http://www.postgresql.org/docs/9.1/static/datatype-numeric.html

-- date: date in ISO format (yyyy-mm-dd)

-- for other data types, visit: http://www.postgresql.org/docs/9.1/static/datatype.html

bd=> create table aluno (id serial, numero int, nome text, apelido text, data_nascimento date, primary key(id));

See the structure of the table

bd=>\d disciplina

2. Insert some rows (do not use accented characters):

Insert rows in the table

bd=>insert into disciplina (nome, descricao, ects) values ('Bases de Dados', 'Fundamentos dos SGBD', 5);

bd=>insert into aluno (numero, nome, apelido, data_nascimento) values (123, 'paulo', 'moreira', '1992-01-01');

-- note the ID is not supplied, but it is automatically created

-- the quote ' is used for TEXT and DATE

-- insert more lines, as you like

bd=>select * from aluno;

3. Delete a couple of lines, choosing clear criteria (in the WHERE clause):

bd=>delete from aluno where numero=123;

-- delete all students with numero=123. It will be only one student.

-- list all students, ordered by "numero"

bd=>select * from aluno order by numero;

4- Update a student, with number=456:

bd=>update aluno set nome='paula' where numero=456;

bd=>select * from aluno order by numero;

5- List some rows respecting a given clause:

bd=>select * from aluno where data_nascimento > '1992-01-01';

-- change the date and see how the number of lines in the result changes

6- Alter the structure of the table, adding a column for the degree (curso):

bd=>alter table aluno add column curso integer;

7- See the new structure:

bd=>\d aluno

8- Add curso number to the existing rows

9- Use pg_dump to dump the database to a text file

10- Drop the table (this is irreversible!):

bd=>drop table aluno;

11- Open the dump file created in 9). Use Notepad++, NoteTab, or other editor. Add more “insert” lines, use copy paste and change at will.

12- Recreate the database from the file created

13- Perform different queries on the table, using different criteria

Use any table you have created.

1. Add constraints: NOT NULL, UNIQUE, CHECK, ON DELETE RESTRICT, ON DELETE CASCADE

a) Check what each one does, and try to override them

2. Use ALTER TABLE to add columns, remove columns, add/remove restrictions, rename columns, change column type

3. Use a timestamp default value to specify for example an insert time

4. Change permissions, SELECT, INSERT, UPDATE, DELETE. Use the GRANT command

5. Create triggers to fire in update/insert situations, testing once per row and once per statement.

Some examples: a football player can only play in games if it is in the call list of the team, a team cannot play against itself, an instructor cannot teach in two different classes at the same time