Databases

Developing a Simple Database Application

N. Antunes, Marco Vieira

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2021/2022

Outline

- Goal: create a new (simple) database for managing the employees of a company; develop a application, using Python, with some features to access and manage data in the employees database
- Prepare the setup
- Requirements
- Database schema
- Architecture
- Create and fill the database
- Code the client application
- Test the client application

Prepare the Setup

- Install PostgreSQL
  - Follow the instruction from the first PL lesson
- Install Python3
- Install a driver to access PostgreSQL from Python3
  - Psycopg
  - PyCharm

Install Python3

- Download the installer from:
  - https://www.python.org/downloads/
- Follow the installation steps
  - In Windows OS, remember to select option "*Add Python to PATH*"
- Check the installation

```
python3 --version
```

  - Make sure that you have a updated version installed (current version is 3.9.1)

Install Psycopg

- Psycopg: PostgreSQL adapter for the Python programming language
  - https://pypi.org/project/psycopg2/
- Installing psycopg2 requires pip package manager:
  - pip is already installed if you are using Python 2 >=2.7.9 or Python 3 >=3.4 downloaded from python.org
  - Otherwise, install pip: https://pip.pypa.io/en/stable/installing/
- Install psycopg2:

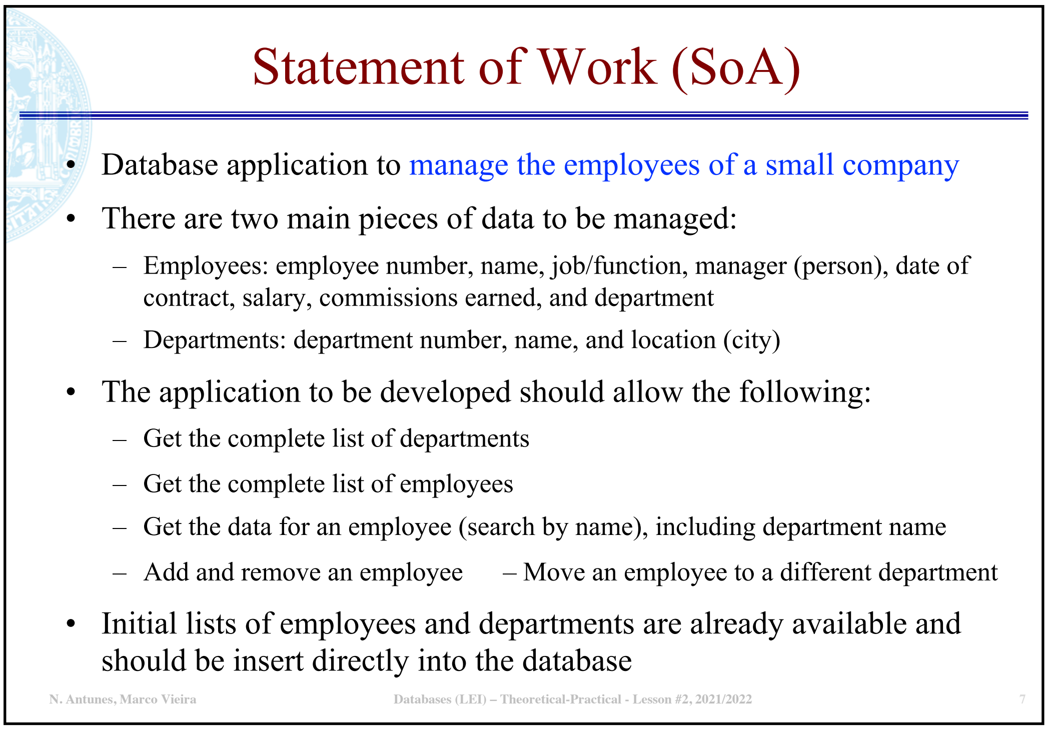
```
pip install psycopg2-binary
```

- If you are using Mac OS, you can specify the python distribution where you want to install psycopg2 using:

```
python3 -m pip install psycopg2-binary
```

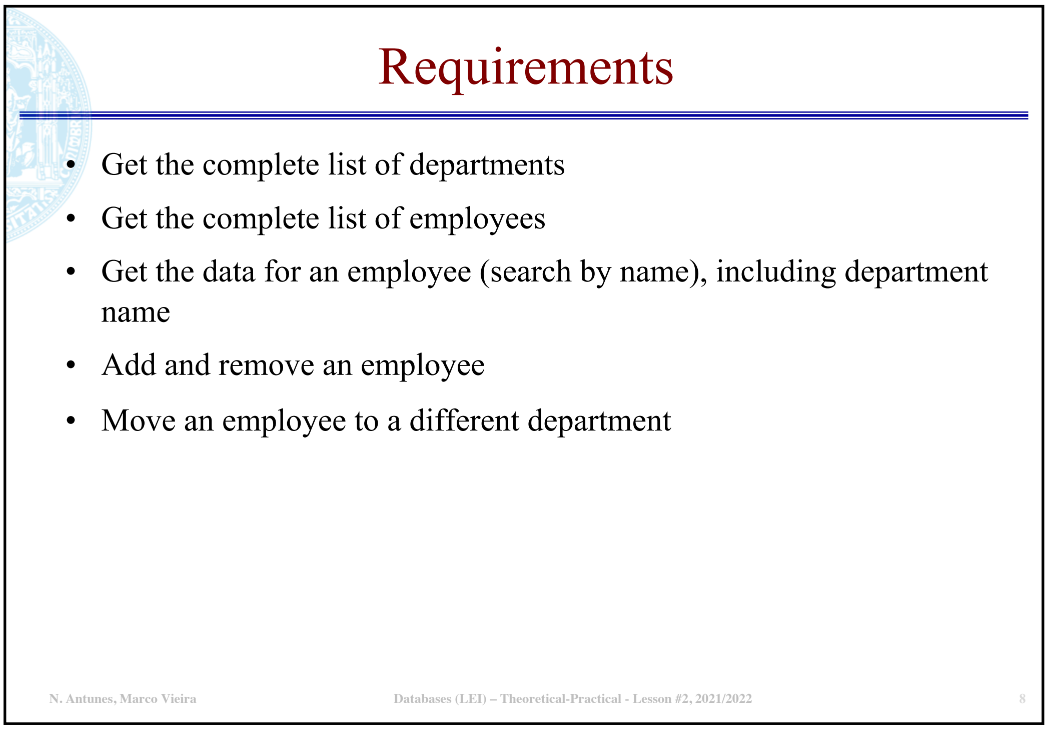
Install PyCharm

- Download PyCharm version 2020.2.
  - https://www.jetbrains.com/pycharm/download/
- Choose between Professional or Community version
  - Community version is free
  - Professional version is free for students. You can apply for education license at: https://www.jetbrains.com/student/
- Follow the installation instructions



Statement of Work (SoA)

- Database application to manage the employees of a small company
- There are two main pieces of data to be managed:
  - Employees: employee number, name, job/function, manager (person), date of contract, salary, commissions earned, and department
  - Departments: department number, name, and location (city)
- The application to be developed should allow the following:
  - Get the complete list of departments
  - Get the complete list of employees
  - Get the data for an employee (search by name), including department name
  - Add and remove an employee – Move an employee to a different department
- Initial lists of employees and departments are already available and should be insert directly into the database

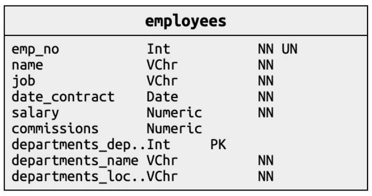


Requirements

- Get the complete list of departments
- Get the complete list of employees
- Get the data for an employee (search by name), including department name
- Add and remove an employee
- Move an employee to a different department

Database Schema

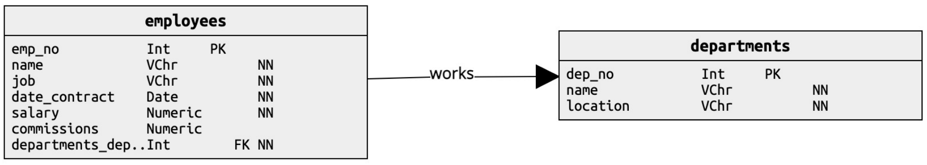
- Can this be represented with a single table?



```
employees
emp_no             Int          NN UN
name               VChr         NN
job                VChr         NN
date_contract      Date         NN
salary             Numeric      NN
commissions        Numeric
departments_dep..Int      PK
departments_name VChr           NN
departments_loc..VChr           NN
```

Database Schema

- Or does it work better with two tables?

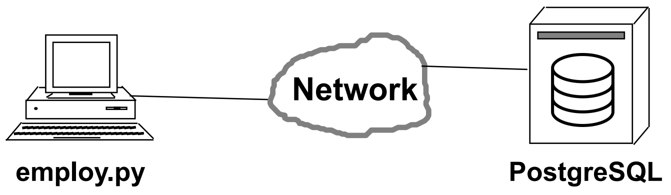


```
employees
emp_no             Int      PK
name               VChr            NN
job                VChr            NN
date_contract      Date            NN
salary             Numeric         NN
commissions        Numeric
departments_dep..Int           FK NN
```

```
departments
dep_no             Int      PK
name               VChr            NN
location           VChr            NN
```

Next classes we will study how to design the database schema…

Architecture



Create the Database

- Create *empdb* database:

```
psql -h localhost -p 5432 -d postgres -U postgres
create database empdb;
```

- Create *empdb* user:

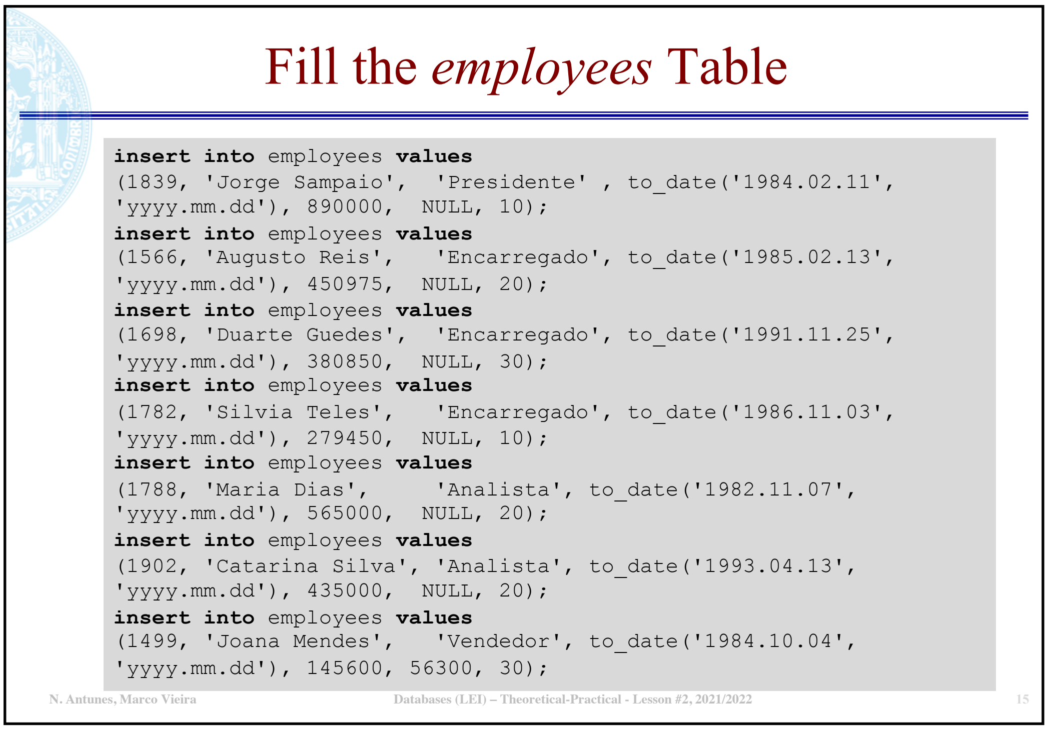
```
create user empuser password 'empuser';
exit
```

Create Tables

```
psql -h localhost -p 5432 -d empdb -U empuser
create table employees (
        emp_no                  integer,
        name                    varchar(40) not null,
        job                     varchar(20) not null,
        date_contract           date not null,
        salary                  numeric(8,2) not null,
        commissions             numeric(8,2),
        departments_dep_no      integer not null,
        primary key(emp_no)
);
create table departments (
        dep_no          integer,
        name            varchar(40) not null,
        location        varchar(20) not null,
        primary key(dep_no)
);
alter table employees add constraint employees_fk1
foreign key (departments_dep_no) references departments(dep_no);
```

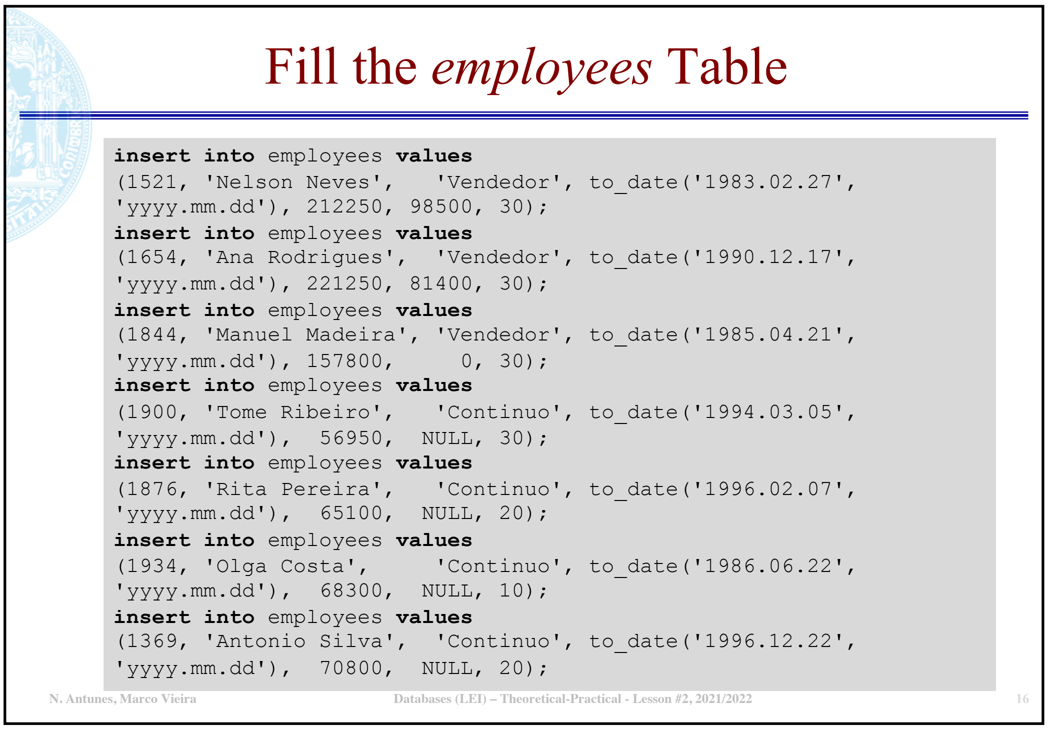
Fill the *departments* Table

```
insert into departments values (10, 'Contabilidade', 'Condeixa');
insert into departments values (20, 'Investigacao', 'Mealhada');
insert into departments values (30, 'Vendas', 'Coimbra');
insert into departments values (40, 'Planeamento', 'Montemor');
```



Fill the *employees* Table

```
insert into employees values
(1839, 'Jorge Sampaio',  'Presidente' , to_date('1984.02.11',
'yyyy.mm.dd'), 890000,  NULL, 10);
insert into employees values
(1566, 'Augusto Reis',   'Encarregado', to_date('1985.02.13',
'yyyy.mm.dd'), 450975,  NULL, 20);
insert into employees values
(1698, 'Duarte Guedes',  'Encarregado', to_date('1991.11.25',
'yyyy.mm.dd'), 380850,  NULL, 30);
insert into employees values
(1782, 'Silvia Teles',   'Encarregado', to_date('1986.11.03',
'yyyy.mm.dd'), 279450,  NULL, 10);
insert into employees values
(1788, 'Maria Dias',     'Analista', to_date('1982.11.07',
'yyyy.mm.dd'), 565000,  NULL, 20);
insert into employees values
(1902, 'Catarina Silva', 'Analista', to_date('1993.04.13',
'yyyy.mm.dd'), 435000,  NULL, 20);
insert into employees values
(1499, 'Joana Mendes',   'Vendedor', to_date('1984.10.04',
'yyyy.mm.dd'), 145600, 56300, 30);
```



Fill the *employees* Table

```
insert into employees values
(1521, 'Nelson Neves',   'Vendedor', to_date('1983.02.27',
'yyyy.mm.dd'), 212250, 98500, 30);
insert into employees values
(1654, 'Ana Rodrigues',  'Vendedor', to_date('1990.12.17',
'yyyy.mm.dd'), 221250, 81400, 30);
insert into employees values
(1844, 'Manuel Madeira', 'Vendedor', to_date('1985.04.21',
'yyyy.mm.dd'), 157800,     0, 30);
insert into employees values
(1900, 'Tome Ribeiro',   'Continuo', to_date('1994.03.05',
'yyyy.mm.dd'),  56950,  NULL, 30);
insert into employees values
(1876, 'Rita Pereira',   'Continuo', to_date('1996.02.07',
'yyyy.mm.dd'),  65100,  NULL, 20);
insert into employees values
(1934, 'Olga Costa',     'Continuo', to_date('1986.06.22',
'yyyy.mm.dd'),  68300,  NULL, 10);
insert into employees values
(1369, 'Antonio Silva',  'Continuo', to_date('1996.12.22',
'yyyy.mm.dd'),  70800,  NULL, 20);
```

Python Application

```
import psycopg2
def get_option():
    …
def connect_db():
    …
def list_departments():
    …
def list_employees():
    …
def get_employee():
    …
def add_employee():
    …
def remove_employee():
    …
def move_emp_department():
    …
function_list=(list_departments, list_employees, get_employee,
add_employee, remove_employee, move_emp_department)
option = -1
while (option!=0):
    option=get_option()
    if (option!=0):
        function_list[option-1]()
```

get_option()

```
def get_option():
    option=-1

    while (option not in [0,1,2,3,4,5,6]):
        print('1 - List Departments')
        print('2 - List Employees')
        print('3 - Get Employee')
        print('4 - Add Employee')
        print('5 - Remove Employee')
        print('6 - Move Employee to Department')
        print('0 - Exit')

        try:
            option=int(input('Option: '))
        except:
            option=-1

    return option
```

connect_db() & list_departments()

```
def connect_db():
    connection = psycopg2.connect(user = "empuser",
        password = "empuser",    # password should not be visible
        host = "localhost",
        port = "5432",
        database = "empdb")
    # parameters should be changable - later on the course

    return connection
```

```
def list_departments():
    print('--- List of Departments ---')

    connection=connect_db()
    cursor = connection.cursor()
    cursor.execute('select * from departments')

    for row in cursor:
        print(row[0],'\t',row[1],'\t',row[2])

    print('---------------------------\n')

    return
```

list_employees()

```
def list_employees():
    print('--- List of Employees ---')

    ## To Be Completed

    print('-------------------------\n')
```

get_employee()

```
def get_employee():
    print('\n--- Get Employee ---')

    name=''
    while (len(name)==0):
        name=input('Name: ')

    connection=connect_db()
    cursor = connection.cursor()
    cursor.execute("select emp_no, employees.name, job, \
    date_contract, salary, commissions, departments.name \
    from employees,departments \
    where departments_dep_no = dep_no \
    and employees.name='"+name+"'") # insecure version

    if (cursor.rowcount==0):
        print('Employee does not exist!')
```

get_employee()

```
def get_employee():
    print('\n--- Get Employee ---')

    name=''
    while (len(name)==0):
        name=input('Name: ')

    connection=connect_db()
    cursor = connection.cursor()
    cursor.execute("select emp_no, employees.name, job, \
    date_contract, salary, commissions, departments.name \
    from employees,departments \
    where departments_dep_no = dep_no \
    and employees.name=%s",(name,)) # secure version

    if (cursor.rowcount==0):
        print('Employee does not exist!')
```

get_employee()

```
    for row in cursor:
        print('No:',row[0])
        print('Name:',row[1])
        print('Job:',row[2])
        print('Date Contract:',row[3])
        print('Salary:',row[4])
        print('Commissions:',row[5])
        print('Department:',row[6])

    print('--------------------\n')
```

add_employee()

```
def add_employee():
    print('--- Add Employee ---')

    ## To Be Completed

    print('--------------------\n')
```

remove_employee()

```
def remove_employee():
    print('\n--- Remove Employee ---')

    name=''
    while (len(name)==0):
        name=input('Name: ')

    connection=connect_db()
    cursor = connection.cursor()
    cursor.execute("delete from employees \
    where name=%s",(name,))

    if (cursor.rowcount==0):
        print('Employee does not exist!')
    else:
        print(cursor.rowcount, 'employee(s) deleted!')
        connection.commit()

    print('-----------------------\n')
```

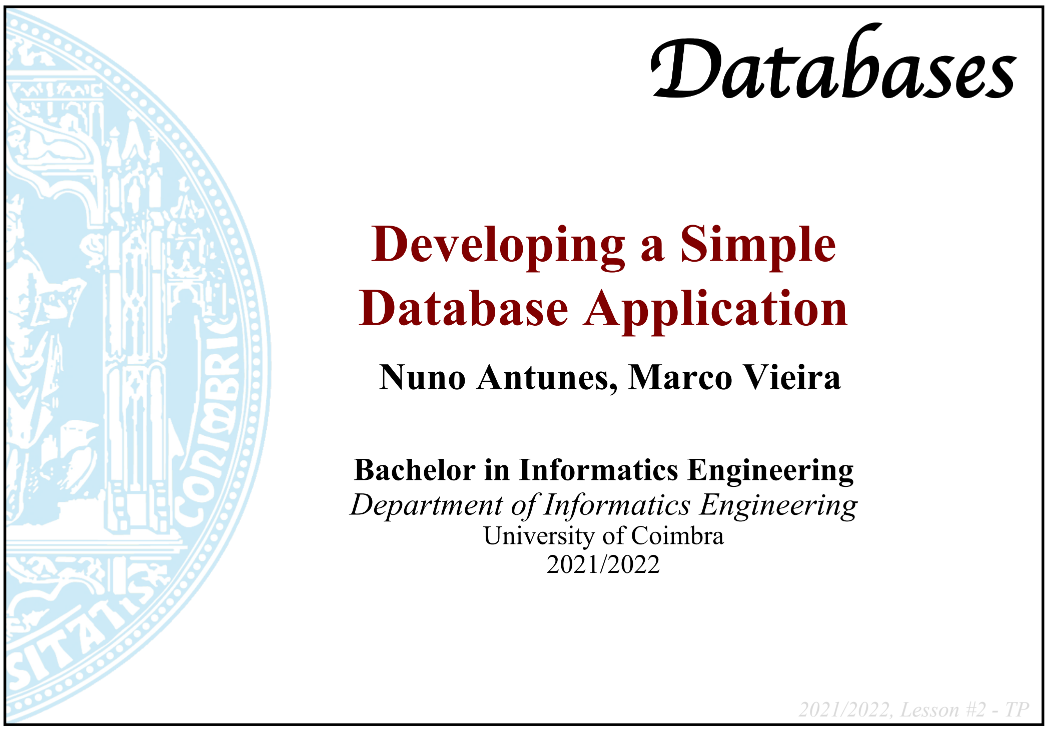
move_emp_department()

```
def move_emp_department():
    print('--- Move Employee to Department ---')

    ## To Be Completed

    print('-----------------------------------\n')
```

Q&A



Databases

Developing a Simple Database Application

Nuno Antunes, Marco Vieira

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2021/2022