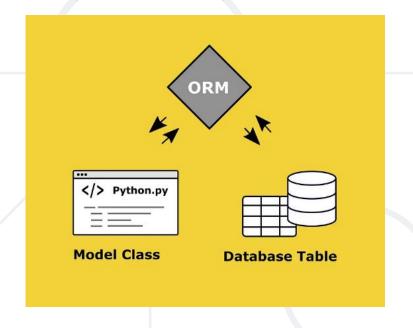
# **ORM Introduction**

Overview, Drivers, Django ORM



**SoftUni Team** Technical Trainers







**Software University** 

https://about.softuni.bg/

### **Table of Contents**



- 1. Introduction to ORM
- 2. Database Drivers
- 3. Django and Django ORM
  - Creating a Django project and application
  - Introducing the ready-to-use skeleton
  - Setting up a database
- 4. Django dbshell



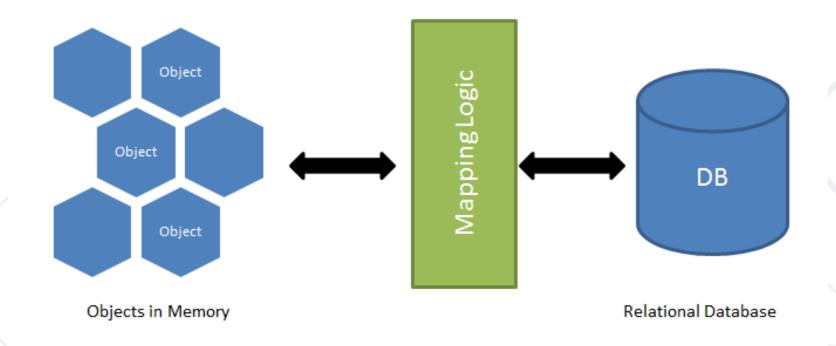
### Questions



sli.do

#python-db

#### O/R Mapping



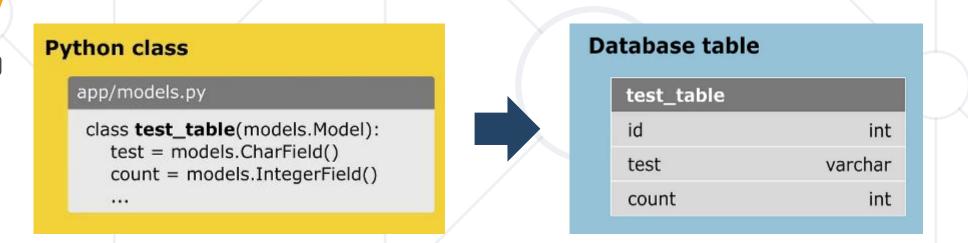
# Introduction to ORM

**Object-Relational Mapping** 

#### What is ORM?



- Object-Relational Mapping (ORM) allows manipulating databases using common classes and objects
  - Python Classes → Database Tables



### **Benefits of ORM**



- Enables you to write code in your preferred OOP language
- Provides a layer of abstraction that hides the complexity of the database schema and relationships
- Can automate some common tasks:
  - Creating, updating, and deleting records
  - Validating data
  - Managing transactions and connections

### **Drawbacks of ORM**



- When it comes to complex queries and aggregations that require high performance and flexibility
  - Can generate inefficient or suboptimal SQL queries
    - Poor performance
    - Excessive memory usage
    - Unexpected errors
- Limits control and customization over the SQL queries and operations
  - Harder to optimize
  - May not support some advanced features or functions

# **SQL** Injection



- SQL Injection is a type of an injection attack
  - Makes it possible to execute malicious SQL statements
  - Attackers can use SQL Injection vulnerabilities to bypass application security measures so they could:
    - go around authentication and authorization
    - retrieve the content of the entire database
    - add, modify, and delete records in the database
    - gain unauthorized access to sensitive data
- ORM reduces explicit SQL queries and is much less vulnerable to SQL injection

### **SQL Injection Example**



Input fields vulnerable to SQL Injection

```
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']
# SQL query vulnerable to SQLi
SELECT id FROM users WHERE username='" + uname + "' AND
password='" + passwd + "'
# Set the passwd field
password' OR 1=1
# Database server runs the following SQL query
SELECT id FROM users WHERE username='username' AND
password='password' OR 1=1'
```

# **ORM Advantages and Disadvantages**





### **Advantages**

- Developer productivity –
   speeds up development
- SQL injection is a lot more difficult
- ORMs work very well with CRUD
- Better readability

### Disadvantages

- Reduced performance (N+1 issue or autogenerated SQL)
- Reduces flexibility (some operations are hard to implement)

# Popular ORM Tools for Python



#### Django

A great tool for building web applications rapidly

#### web2py

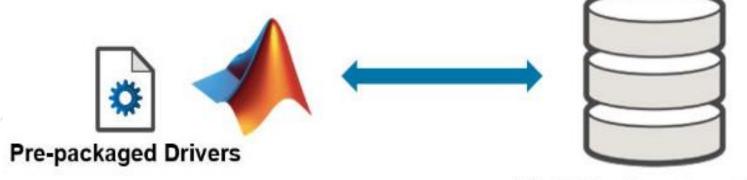
An open source full-stack Python framework

#### SQLObject

An ORM that provides an object interface to your database

#### SQLAlchemy

 Provides persistence patterns designed for efficient and highperforming database access



MySQL, PostgreSQL, SQLite Database

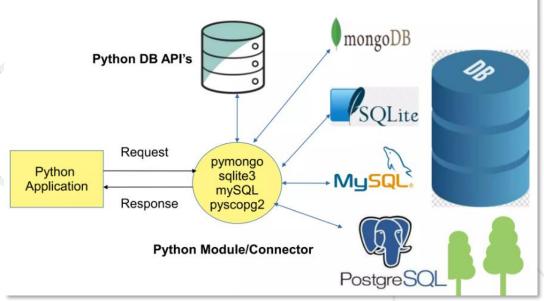
# **Database Drivers**

Psycopg2

### **Database Drivers**



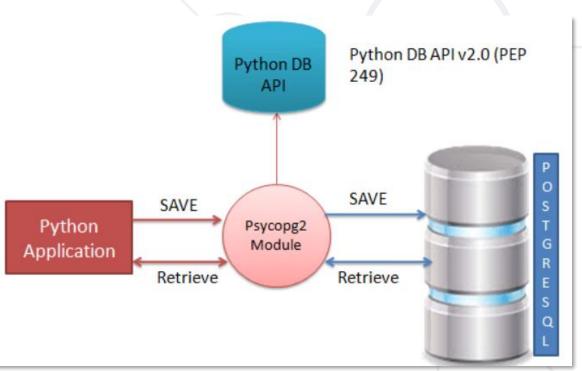
- Database Driver (module/connector) is a computer program that implements a protocol for a database connection
  - Works like an adapter that connects a generic interface to a specific database vendor implementation
  - Accesses the physical data through a stand-alone engine
  - Submits SQL statements to and retrieves results from the engine



# Psycopg2



- PostgreSQL database adapter for Python programming language
- Use Psycopg2 module to:
  - Connect to PostgreSQL
  - Perform SQL queries and database operations
- Psycopg2 is an external module



# **Psycopg2 Advantages**



- Preferred module for PostgreSQL interaction
- Trusted by Python and PostgreSQL frameworks
- Actively maintained
- Fully supports Python's primary versions, ensuring seamless integration
- Thread-safe design built to ensure safe and reliable operation
- Designed to handle heavy multi-threaded applications

# **Install Psycopg2**



- You need to install the current version of Psycopg2 on your machine to use PostgreSQL from Python
- Using the following pip command, you can install Psycopg2 on any operating system

pip install psycopg2





# Django & Django ORM

Django Project, Django App

### What is Framework?



- Platform for developing software applications
- Provides a foundation on which software developers can build programs for a specific platform
- A framework includes an API
- May include code libraries, a compiler, and other programs used in the software development process

# What is Django Framework?



- High-level Python Web Framework
  - Ridiculously fast
- Reassuringly secure
- Exceedingly scalable
- Free and Open Source



# **Django ORM**



- Django is equipped with an ORM
- One of the best ORMs available in the industry today
- Tightly coupled with the Django framework
- Very efficient
- Ability to handle medium to low complexity queries and medium to huge datasets
- Migrations are another useful feature





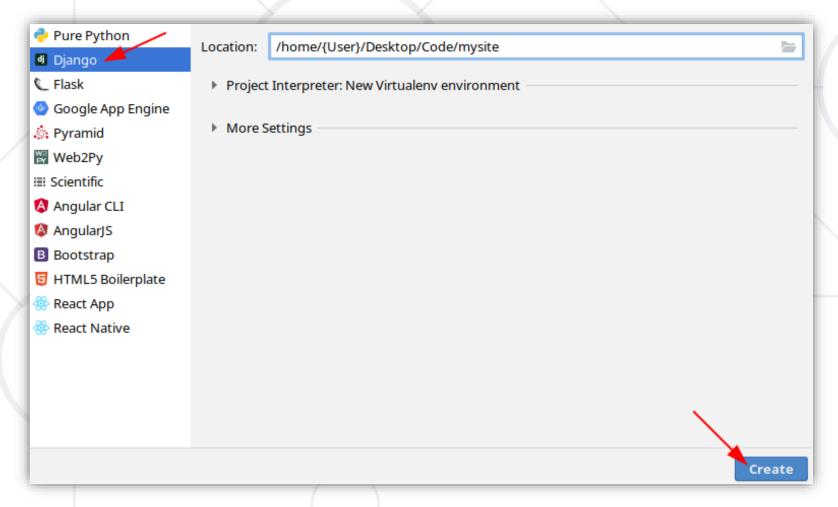
# Creating a Django Project

Where the Magic Happens

# Creating a Django Project



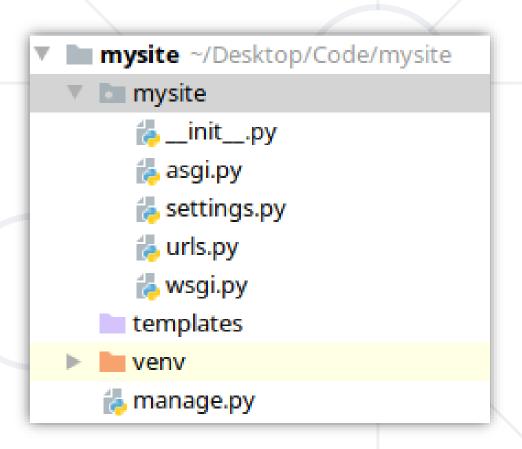
Open PyCharm Professional -> File -> New Project



# **Project Structure**



- \_\_init\_\_.py
  - The directory is a Python package
- settings.py
  - The configuration file for the Django Project
- urls.py
  - Contains the list of URLs
- manage.py
  - Tool for executing commands



# Running a Django Project (1)



Using Terminal command

python manage.py runserver

Using Keyboard Shortcut in PyCharm

Using PyCharm Run button



### Running a Django Project (2)



You'll see the following output on the command line:

```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 18, 2020 - 11:15:18

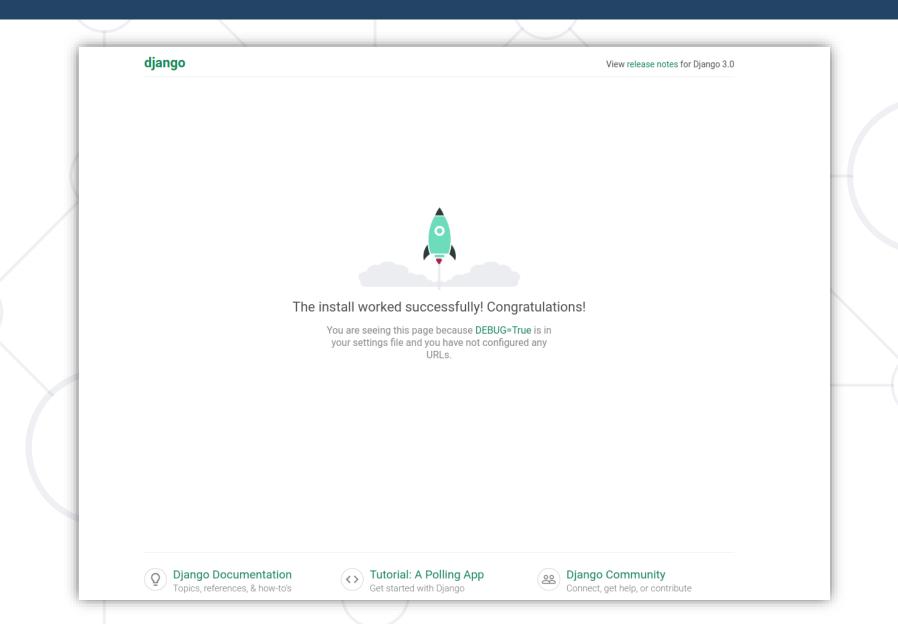
Django version 3.0.3, using settings 'mysite.settings'

Starting development server at <a href="http://127.0.0.1:8000/">http://127.0.0.1:8000/</a>
Quit the server with CONTROL-C.
```

- The runserver command starts the development server on the internal IP at port 8000 by default
- Note: this server is intended for use only while in Development Mode

# Running a Django Project (3)







# Django Application

The Bread and Butter of a Django Project

# **App vs Project**



- Django App:
  - A Web application that does something - e.g., a blog system or a small task app
  - One app can be used in multiple projects

- Django Project:
  - A collection of configuration and apps for a particular website
  - The project can contain multiple apps



### **Creating a Django App**



The app is created in the same directory as the manage.py file

Use the terminal command

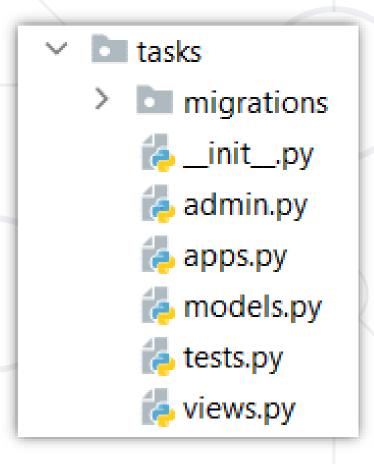
python manage.py startapp tasks

- Django automatically generates the basic directory structure of an app
- The app can be moved inside the root directory of your
   Django project (one good way of project management)

### **Directory Structure**



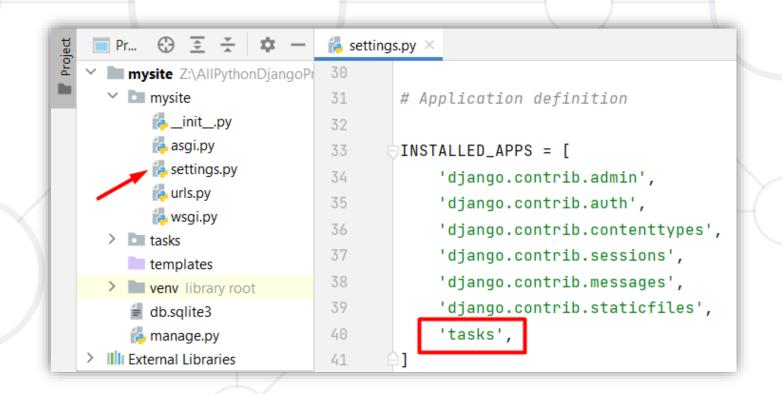
- admin.py
  - The admin site module
- models.py
  - The models of the app
- views.py
  - The views of the app
- migrations
  - Command-line utility for propagating changes in models



# Including an App



 To include an app in a project, add a reference to the app in the INSTALLED\_APPS setting





# Ready-to-Use Skeleton

# Ready-to-Use Skeleton (1)



- You will be provided with a ready-to-use skeleton for the purposes of the ORM Course
  - No need to create a project and apps from scratch
- Running the skeleton
  - Open it with Pycharm Professional
  - Install dependencies
  - Install psycopg2
  - Run the project

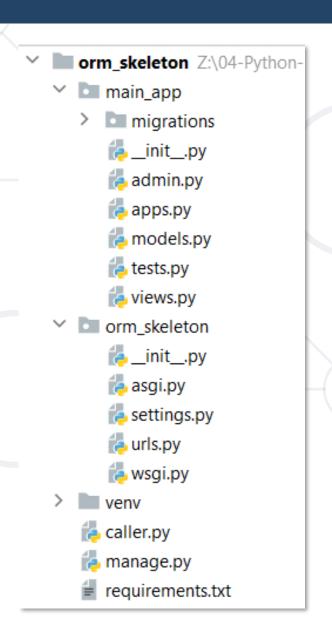
```
py -m pip install -r requirements.txt
```

py -m pip install psycopg2

# Ready-to-use Skeleton (2)



- Skeleton structure
  - settings.py
    - database settings
  - main\_app
    - migrations folder
    - admin.py, models.py
  - caller.py
    - Python code and Django queries
  - requirements.txt
    - dependencies



# Ready-to-use Skeleton (3)



- Activate Python Virtualenv for Windows
  - Open a Command Prompt terminal in the project root directory
  - Execute the following command

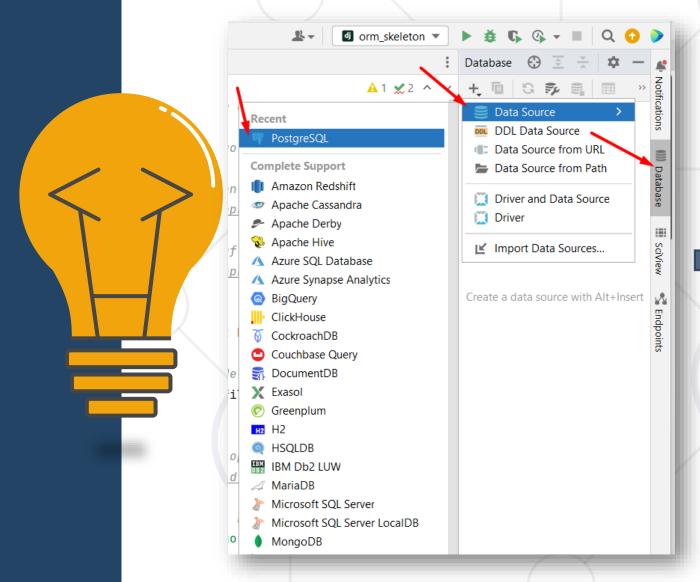
venv\Scripts\activate

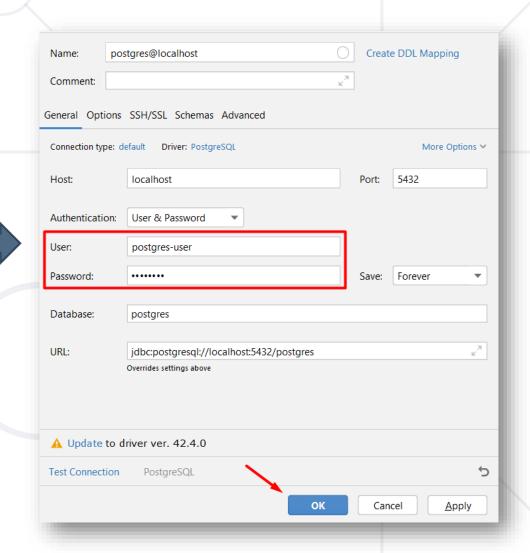


# Setting up a Database

#### Connect to PostgreSQL



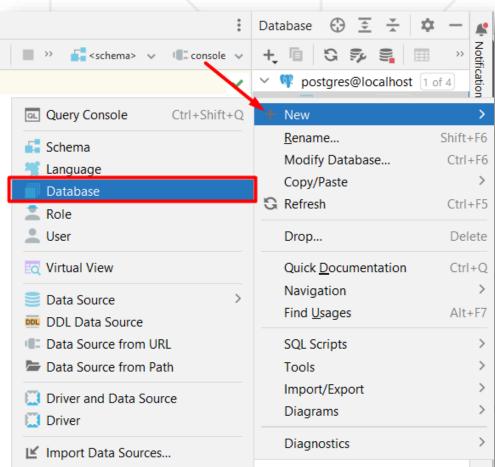


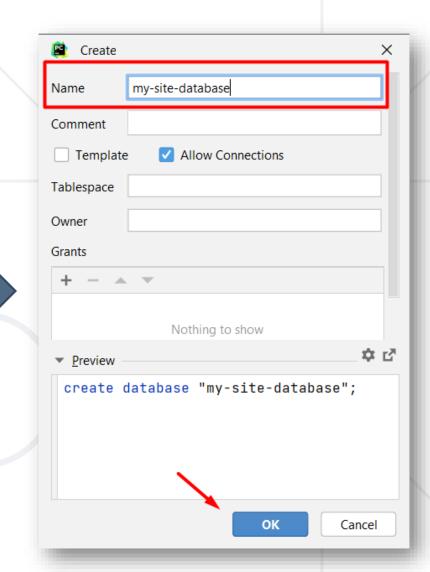


#### **Create a Database**









#### Set up PostgreSQL



 To configure our project to work with PostgreSQL, we need to set it up in the settings.py file

```
Use PostgreSQL
        DATABASES = {
             'default':
                 'ENGINE': 'django.db.backends.postgresql',
Name of the
                 'NAME': 'my-site-database',
                 'USER': 'postgres-user',
 database
                                                  Database user
                 'PASSWORD': 'postgres',
                 'HOST': '127.0.0.1',
                                                    credentials
                 'PORT': '5432'
```



## Django dbshell



 An interactive command-line interface shell environment

Runs the command-line client for specified database,
 or the default database

 A very useful tool for SQL database debugging when working on a Django application



#### Running Django dbshell



#### python manage.py dbshell

#### py -m manage dbshell

## Running Django dbshell (2) - Error



- In case you receive an Error like:
  - "CommandError: You appear not to have the 'psql' program installed or on your path"
- Follow the next steps:
  - Find your PostgreSQL binary path
    - C:\Program Files\PostgreSQL\<version>\bin
    - Where <version> is your current PostgreSQL version(e.g., 15)
  - Add this path to Windows PATH environment variables

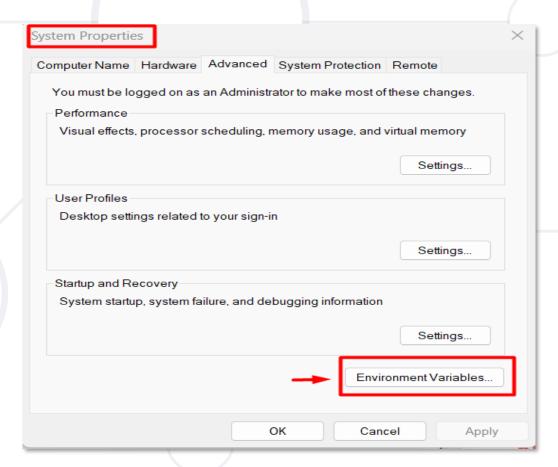
## Add Path to Windows Path Environment (1)



C:\Program Files\PostgreSQL\<version>\bin - copy your path!

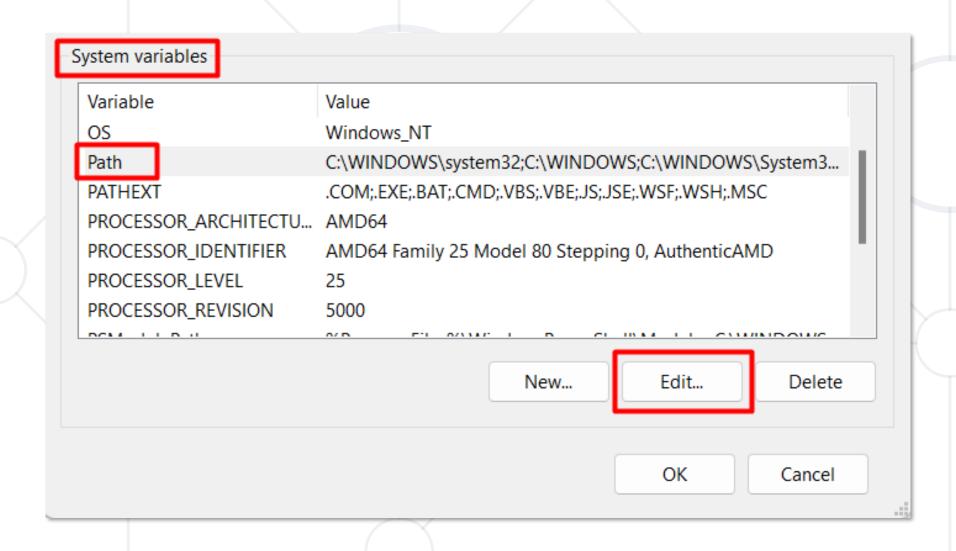
Add it to the Windows PATH environment variables – View Advanced

**System Settings** 



## Add Path to Windows Path Environment (2)

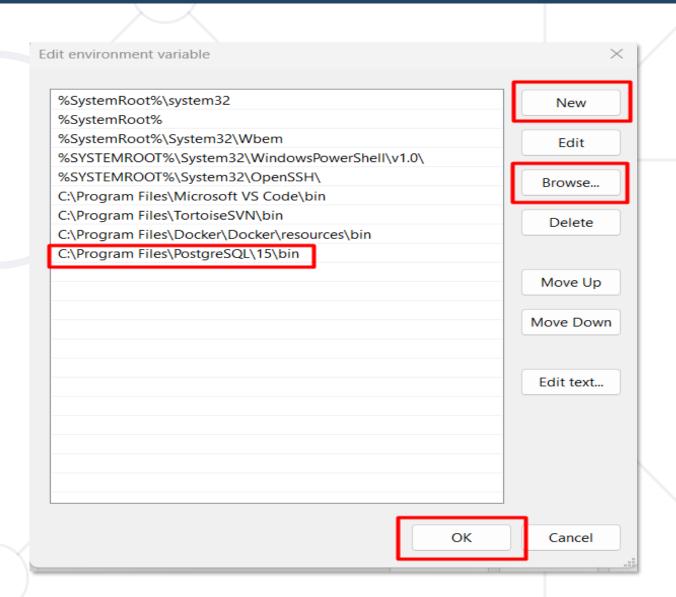




#### Add Path to Windows Path Environment (3)



- Navigate to your path or paste the copied one
- Press **OK** in all windows and restart your **IDE**



## Using Django dbshell (1)



dt command shows all tables in the current database

```
Terminal:
     Local ×
fruitipedia_db=# \dt
          List of relations
Schema l
                     Type
           Name
                           0wner
public | auth_group
                   | table | postgres
public | auth_group_permissions | table | postgres
public | auth_user
                    | table | postgres
public | auth user groups | table | postgres
public | auth_user_user_permissions | table | postgres
public | django_admin_log
                | table | postgres
public | fruit_fruit
             | table | postgres
(12 rows)
```

## Using Django dbshell (2)



\d <table\_name> command shows a specific table

```
fruitipedia_db-# \d fruit_fruit
                                Table "public.fruit_fruit"
    Column
                                     | Collation | Nullable |
                      Type
                                                                      Default
id
      | bigint
                                               | not null | generated by default as identity
fruit_name | character varying(30) |
                                               | not null |
fruit_image_url | character varying(200) | | not null |
                                                | not null |
 description | text
nutrition_info | text
Indexes:
   "fruit_fruit_pkey" PRIMARY KEY, btree (id)
fruitipedia_db-#
```

## Using Django dbshell (3) - Queries



SELECT \* FROM <table\_name>;



## **Practice**

Live Demo in Class

#### Summary



- ORM
  - Object-Relational Mapping
  - Python Classes -> DB Tables
- DB Drivers
  - Psycopg2
- Django and Django ORM
  - Django Project, Django App
- Django dbshell





# Questions?

















#### **SoftUni Diamond Partners**



SUPER HOSTING .BG

























## Trainings @ Software University (SoftUni)



- Software University High-Quality Education,
   Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg









#### License



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is copyrighted content
- Unauthorized copy, reproduction or use is illegal
- © SoftUni <a href="https://about.softuni.bg">https://about.softuni.bg</a>
- © Software University <a href="https://softuni.bg">https://softuni.bg</a>

