

## Configuración de base de datos source y entorno para ETL en Python 6/25



Curso de Fundamentos de ETL con Python y Pentaho



→EXTRACCIÓN DE D...

**¡Hola!** En esta clase conocerás cómo configurar una base de datos con SQL, que será una de las 3 fuentes para extraer datos en el proyecto de ETL. Las otras dos fuentes son un archivo JSON y otro CSV que conocerás en clases posteriores.

Además, conocerás cómo conectarte a esta base de datos OLTP con un software de administración de bases de datos. Puede ser DataSpell, DBeaver o el de tu preferencia.

**Te sugiero usar DataSpell.** Más adelante de este tutorial verás cómo configurarlo.

💡 Algo que tenemos que destacar es que la base de datos SQL source no se tendría que crear en un proceso de ETL. Esta base de datos ya estaría creada en algún lado de la infraestructura de los sistemas y aplicaciones de la empresa donde estés colaborando.

En este caso lo estamos haciendo por fines educativos para que tengas una base de datos de donde tomar datos y conozcas el proceso de extracción.

Para la configuración de nuestra base de datos source usaremos **PostgreSQL**. Podemos utilizarlo de dos formas, una instalación local de PostgreSQL o una configuración por Docker. **Te sugiero hacerlo por Docker.**

### 1. Crear container en Docker

Recordemos que **Docker es un entorno de gestión de contenedores**, de manera que usaremos una imagen base con toda la configuración que requerimos sin instalar necesariamente en nuestra máquina. Solo utilizando los recursos del sistema para correr dicha imagen, algo similar a una máquina virtual.

Por ahora, solo necesitas haber tomado el [Curso de Python: PIP y Entornos Virtuales](#) para conocer lo esencial de cómo usar esta herramienta con Python. En ese curso encontrarás la [clase para saber cómo instalarlo en tu computador](#).

Una vez que tengas instalado Docker en tu computador, ejecuta este comando en tu terminal:

*WSL 2, Linux o macOS*

```
sudo docker run -d --name=postgres -p 5432:5432 -v postgres-volume:/var  
/lib/postgresql/data -e POSTGRES_PASSWORD=mysecretpass postgres
```

*Windows*

```
docker run -d --name=postgres -p 5432:5432 -v postgres-volume:/var  
/lib/postgresql/data -e POSTGRES_PASSWORD=mysecretpass postgres
```

Como podrás notar, en este comando se especifico lo siguiente para la creación de la base de datos con Docker:

- Nombre del container: --name=postgres
- Puerto a compartir con la máquina local: -p 5432:5432
- Volumen para el manejo de disco e información: -v postgres-volume:/var  
/lib/postgresql/data
- Password en PostgreSQL: POSTGRES\_PASSWORD=mysecretpass

## 1.5 Instalación local de PostgreSQL (opcional)

---

De no usar Docker podrías ver la [clase del curso de PostgreSQL](#) en donde aprendes a instalarlo localmente, pero te sugiero intentarlo con Docker ya que puede agilizar tu

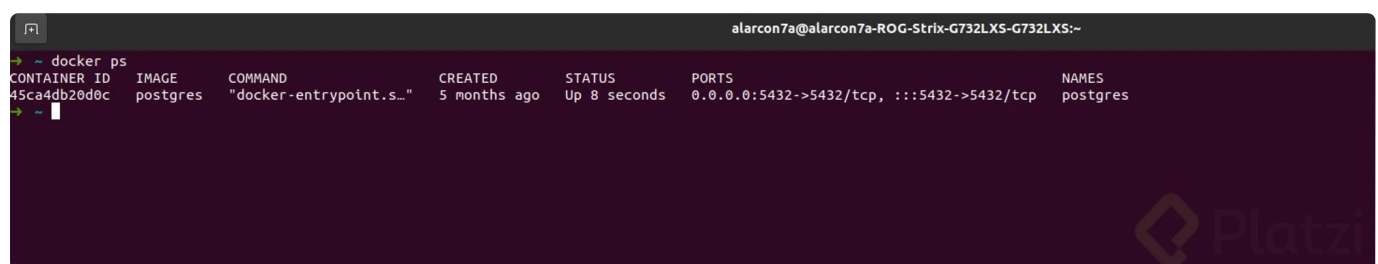
flujo de trabajo. 😊

## 2. Validar container creado

---

Una vez que hayas creado el container de Docker usa el comando `docker ps` en tu terminal. Podrás ver todos los contenedores que se encuentran en ejecución actualmente y una descripción.

Deberás ver la **IMAGE postgres**.



```
alarcon7a@alarcon7a-ROG-Strix-G732LX5-G732LX5:~  
→ ~ docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES  
45ca4db20d0c   postgres  "docker-entrypoint.s..." 5 months ago   Up 8 seconds   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   postgres
```

## 3. Configurar DataSpell

---

Para conectarte a la base de datos usarás un software de administración de bases de datos. Existen varios que puedes utilizar. Para el seguimiento del curso te sugiero utilizar **DataSpell** o, en su defecto, **DBeaver**.

DataSpell es un **IDE completo para ciencia de de datos** donde, además de conectarte y hacer consultas a bases de datos, podrás ejecutar Jupyter Notebooks. **¡Todo en el mismo lugar!** 💪



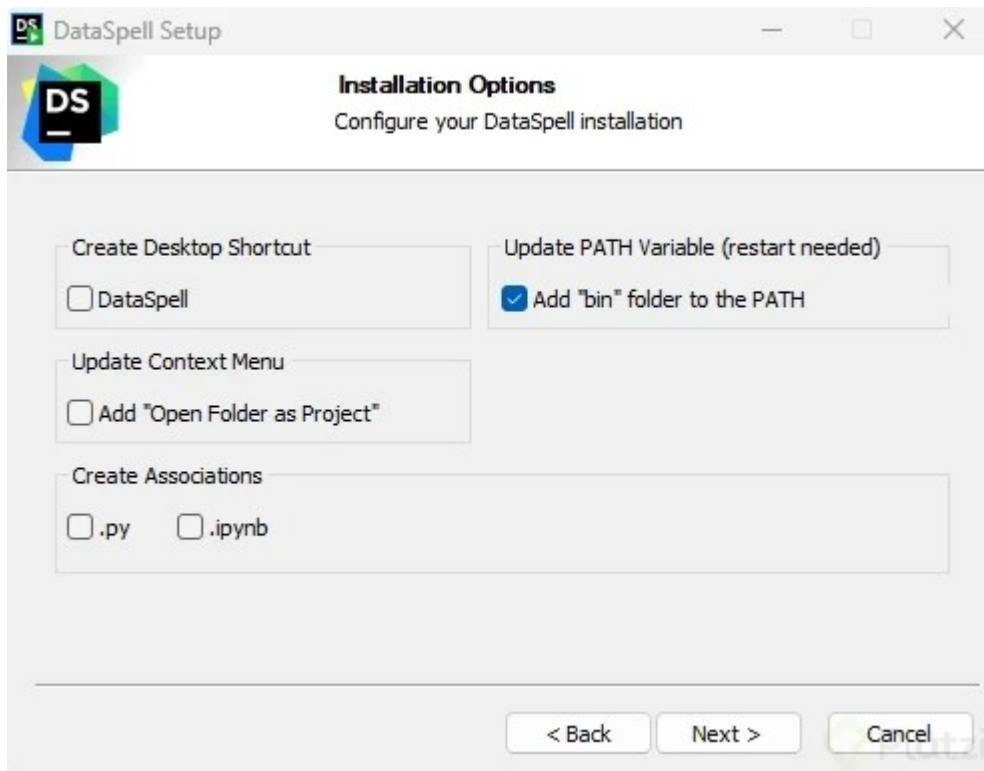
💡 Una de sus desventajas es que es de pago, pero tiene un período de prueba de 30 días para que lo pruebes con este curso. Además existen ciertas opciones para obtener [licencias para estudiantes de bachillerato y universidad](#).

⚠️👉 En caso de que decidas usar DBeaver en lugar de DataSpell, utiliza tu entorno local de **Jupyter Notebooks con Anaconda** para la ejecución del código Python de las siguientes clases. 🐍

## Instalación de DataSpell

1. Para instalar DataSpell ve a [su sitio web aquí](#) y descarga la versión para tu sistema operativo. 📁
2. Instálalo siguiendo las instrucciones que te aparezcan en el instalador.

⚠️ Cuando te solicite actualizar PATH Variable acepta marcando la opción que te indique. Esto es para evitar errores de ambientes en el futuro. En Windows se ve así:



Al finalizar te pedirá reiniciar el computador:

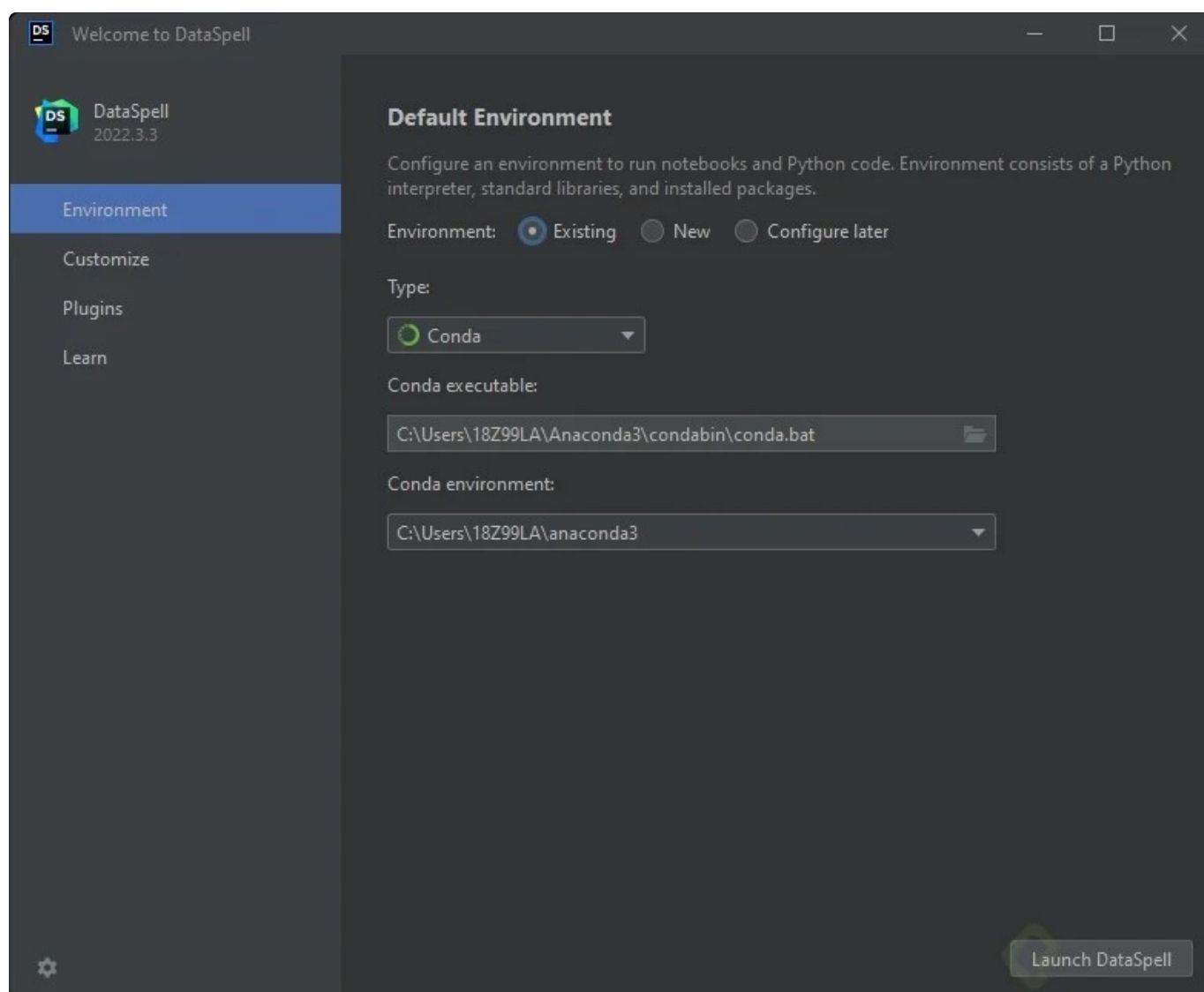


4. Abre DataSpell ya que se haya instalado. Al hacer esto por primera vez te pedirá iniciar sesión. Elige la versión free trial registrando tu cuenta para ello.

- Una vez que tengas tu cuenta configurada te pedirá elegir un intérprete de Python 🐍.

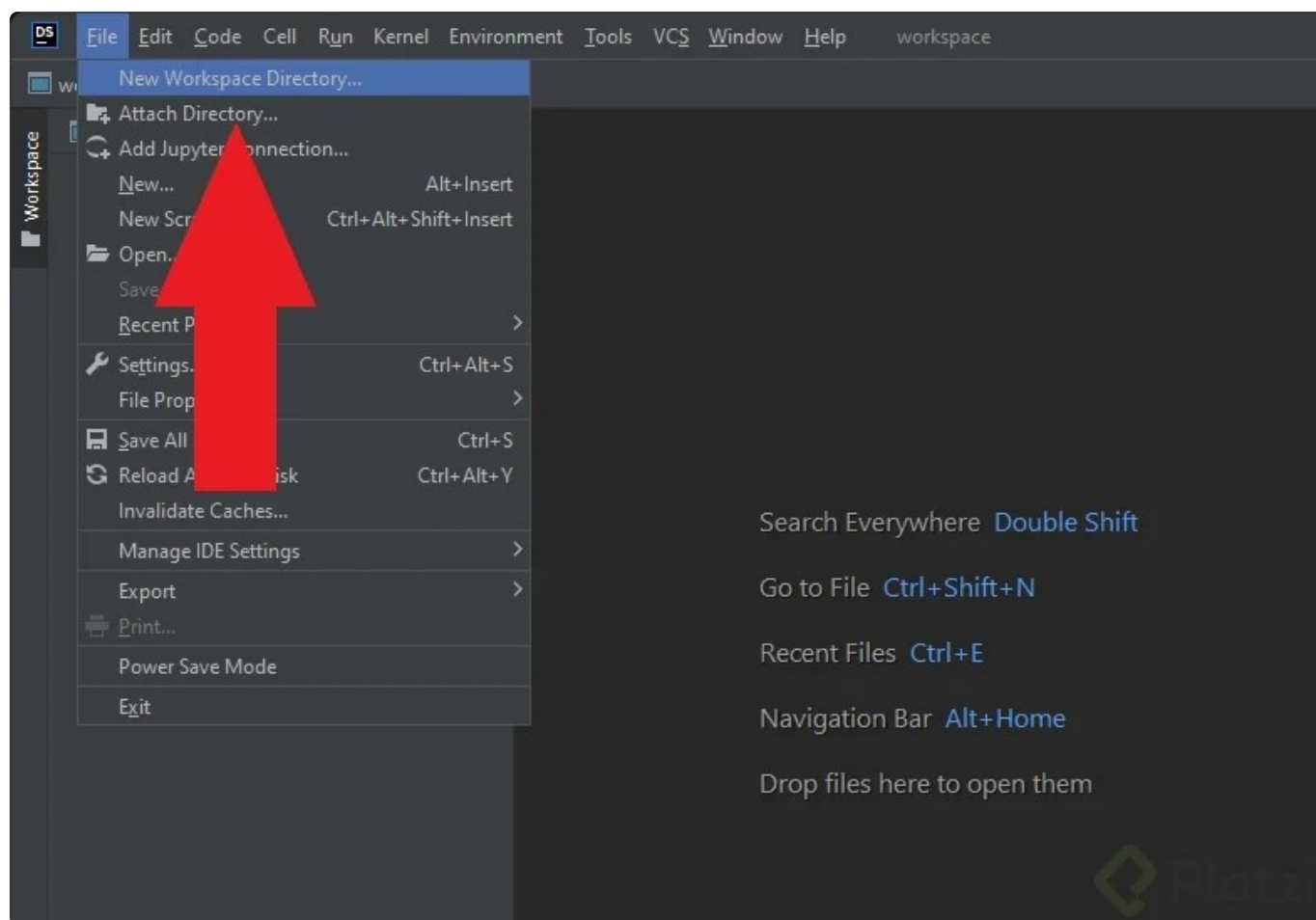
Previamente deberás tener instalado **Anaconda** en tu sistema operativo. Te recomiendo que crees un **ambiente de Anaconda (Conda environment)** único para el proyecto del curso. Llama al ambiente `fundamentos-etl`.

Elige el ambiente de Anaconda que usarás para el proyecto y presiona el botón **Launch DataSpell**.

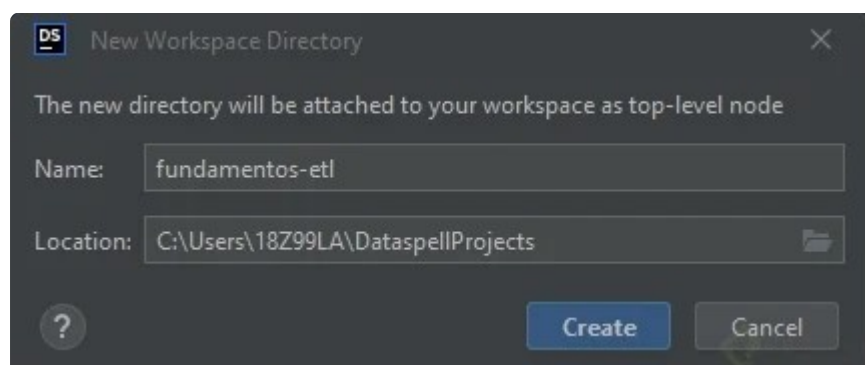


Elegir un intérprete de Anaconda servirá para ejecutar Jupyter Notebooks en DataSpell.

6. **Crea un nuevo Workspace en DataSpell.** Presiona el botón **File** en la barra superior y luego elige la opción **New Workspace Directory**.



Llama fundamentos-etl al workspace y presiona el botón azul **Create**.



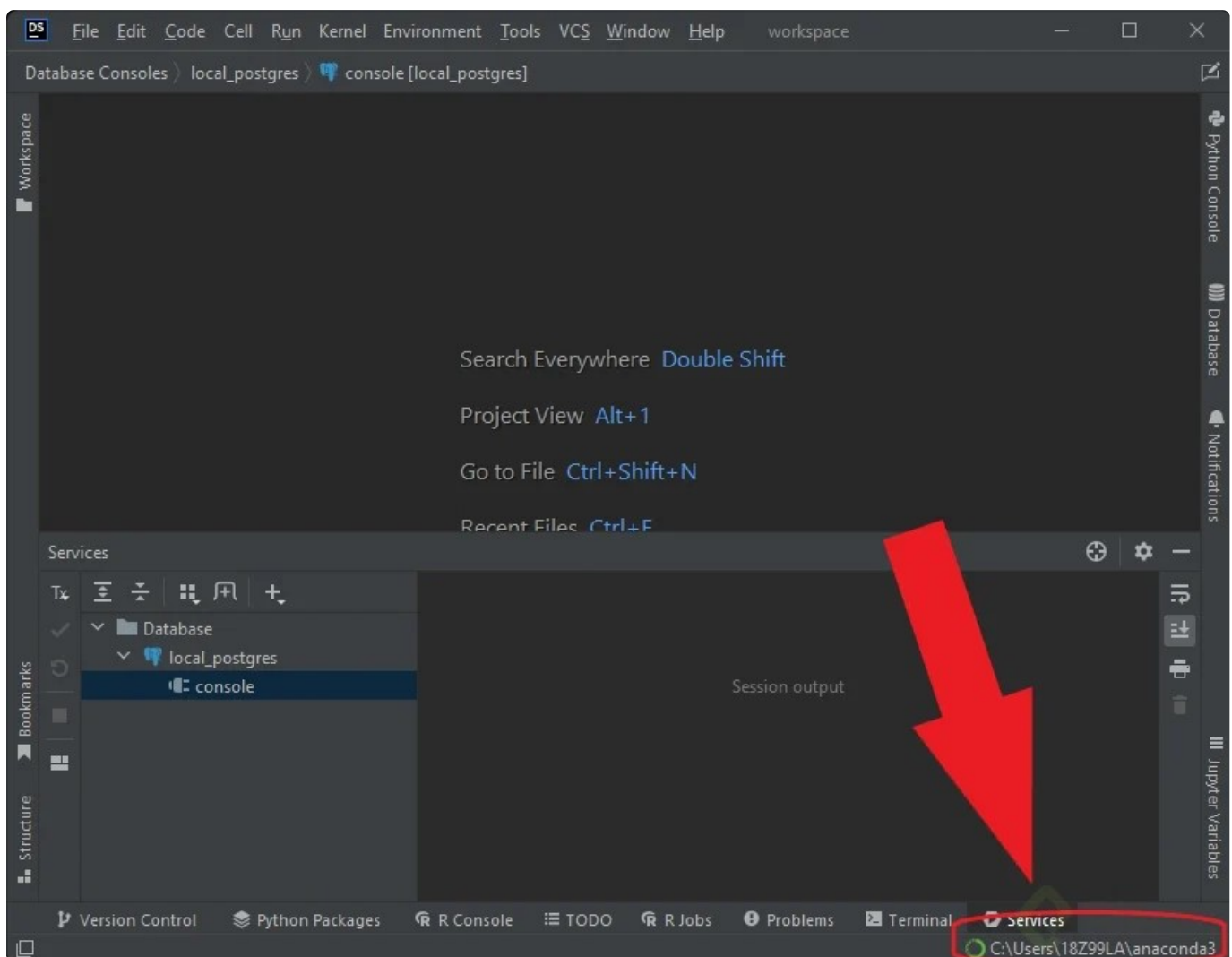
## Elegir ambiente de WSL2 (opcional si usas WSL)

Si quieres usar DataSpell con tu entorno en **Windows con WSL 2**, deberás conectar DataSpell al ambiente de Anaconda que tenga tu WSL. 🐍

0. Crea un ambiente de Anaconda en tu WSL dedicado al proyecto de tu curso si todavía no lo has hecho. Llámalo `fundamentos-etl`

```
conda create --name fundamentos-etl python=3.9
```

1. Después ve a DataSpell en su parte inferior donde aparece el intérprete. Presiona la dirección que aparece y elige la opción **Interpreter Settings**.

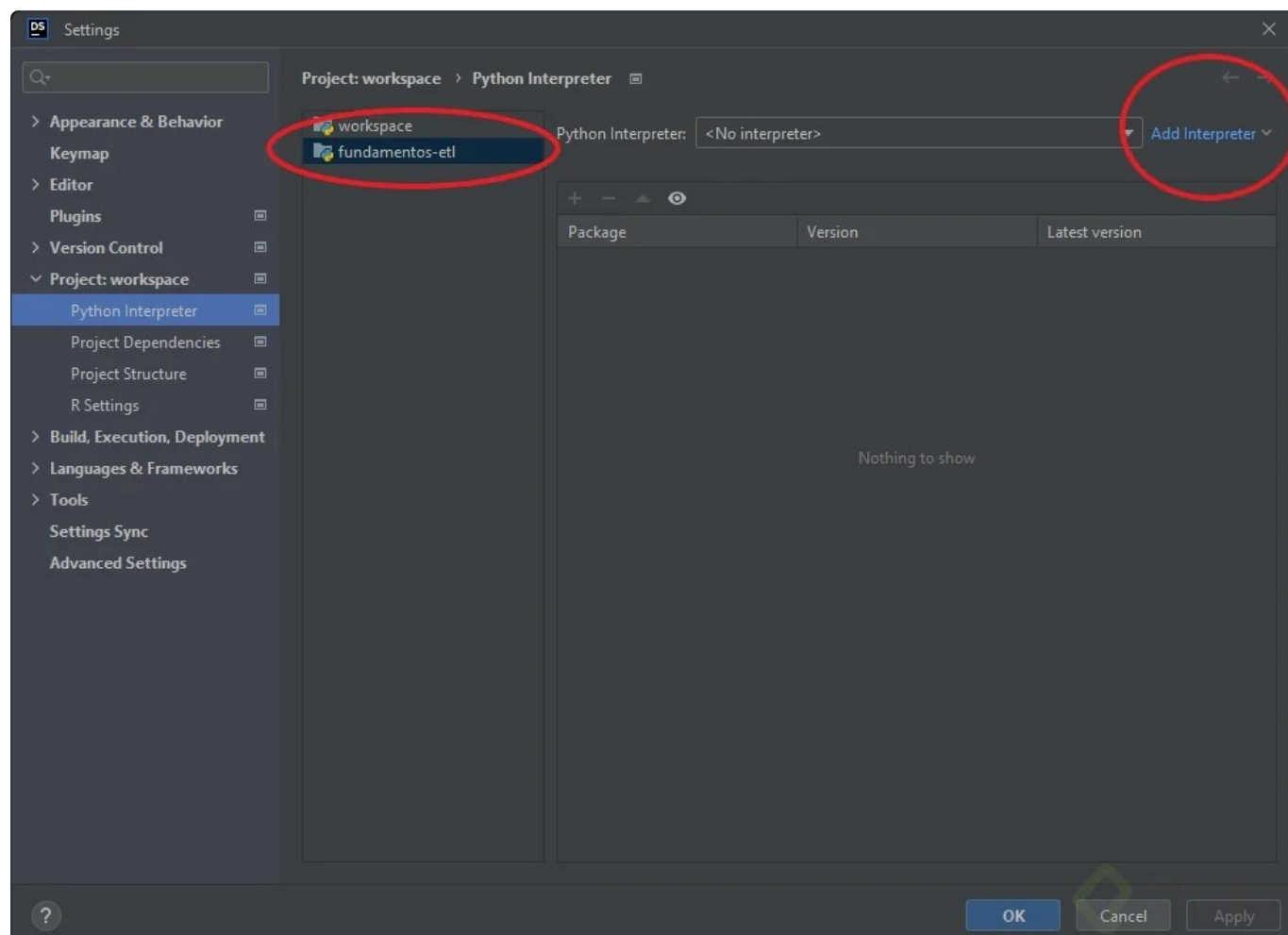


2. Escoge el workspace `fundamentos-etl` creado anteriormente en DataSpell.

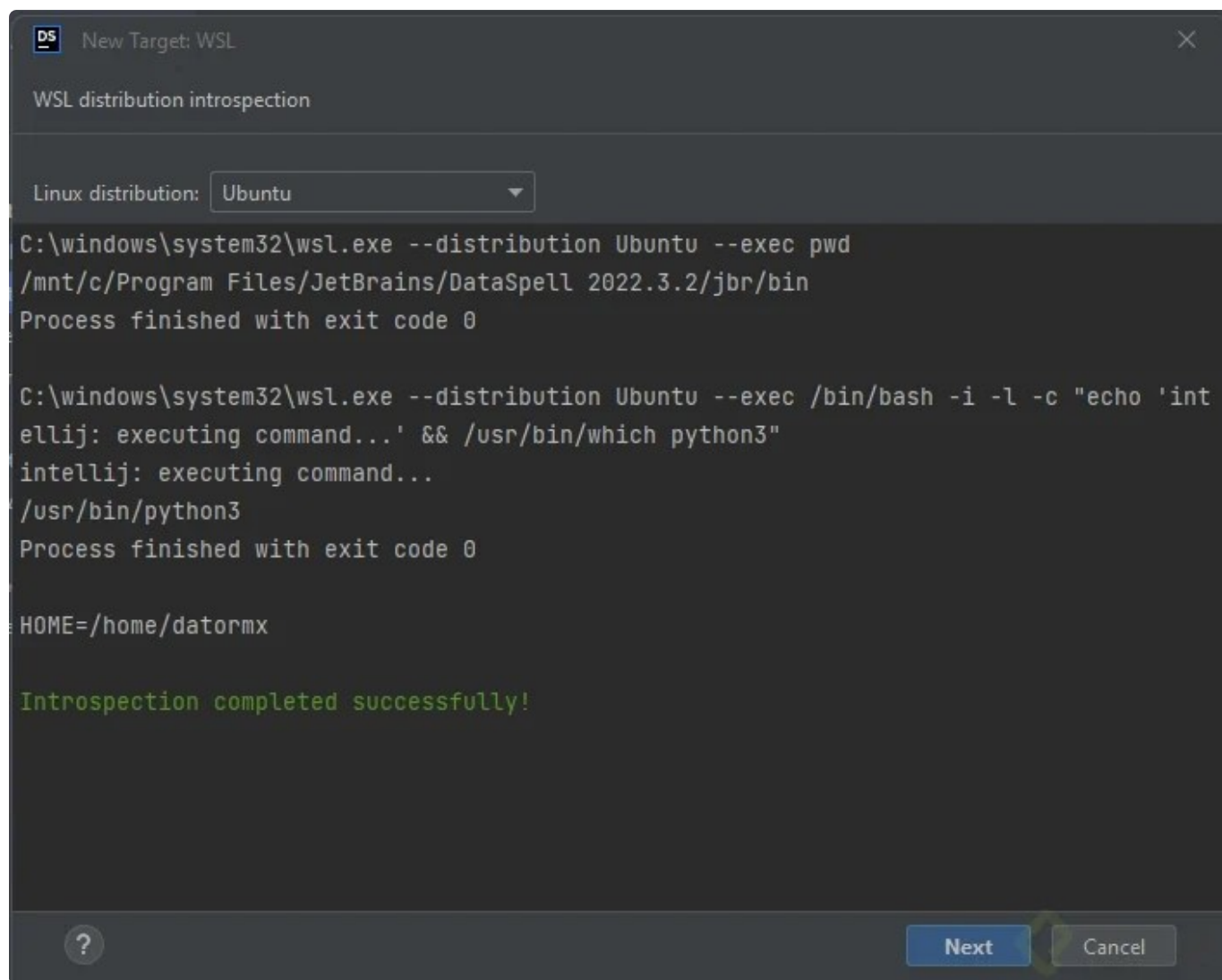
⚠ OJO: el workspace y el Anaconda Environment no son lo mismo. El Anaconda Environment lo vamos a cargar dentro del Workspace de DataSpell.



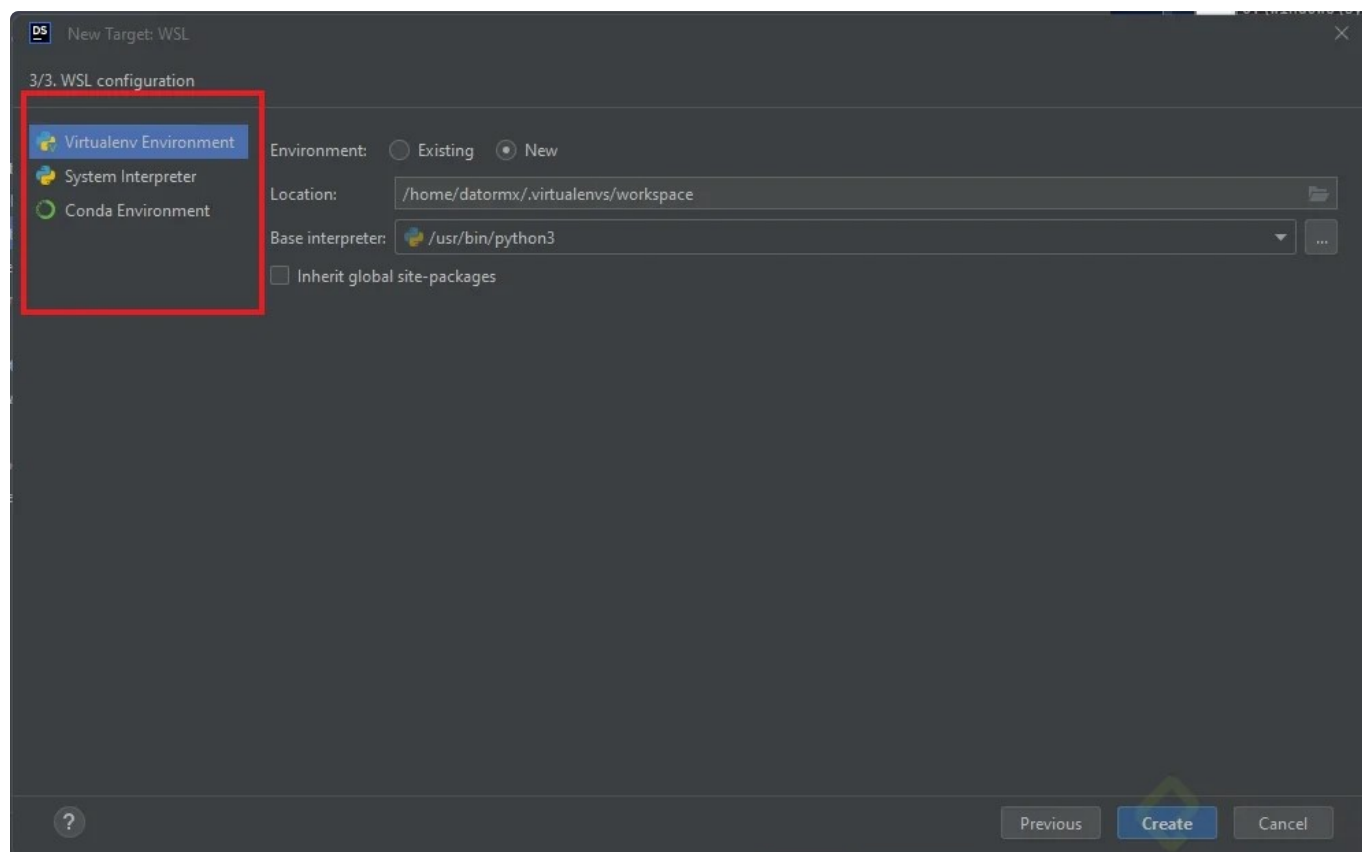
Después presiona el botón **Add Interpreter** e inmediatamente selecciona la opción **On WSL**.



3. Elige la distribución de Linux a usar y da clic en el botón **Next** cuando aparezca el mensaje "Introspection completed successfully!"

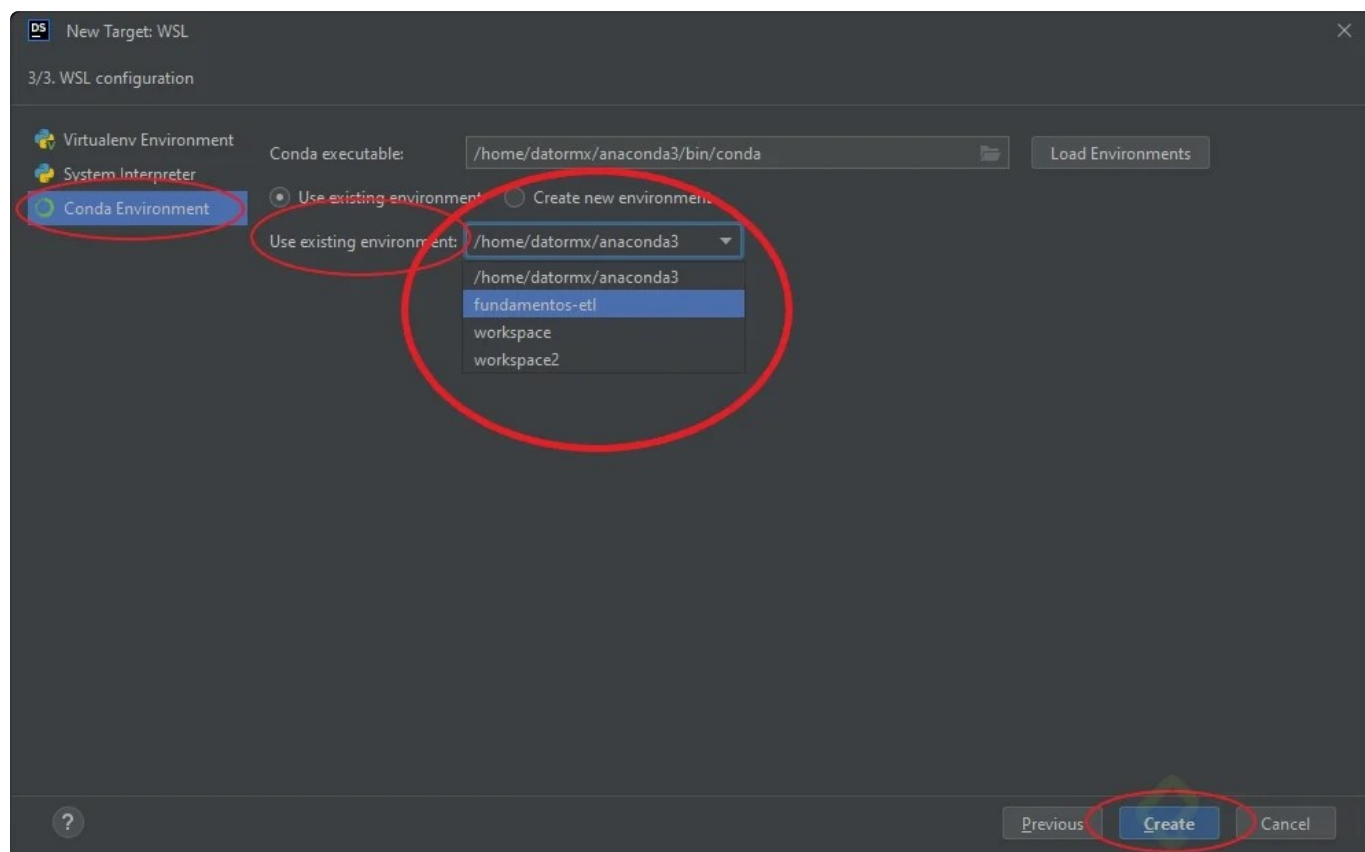


4. Elige el intérprete a usar. Este puede ser un **Virtualenv Environment**, el **System Interpreter** o un **Conda Environment**. Elige la opción de **Conda Environment**.

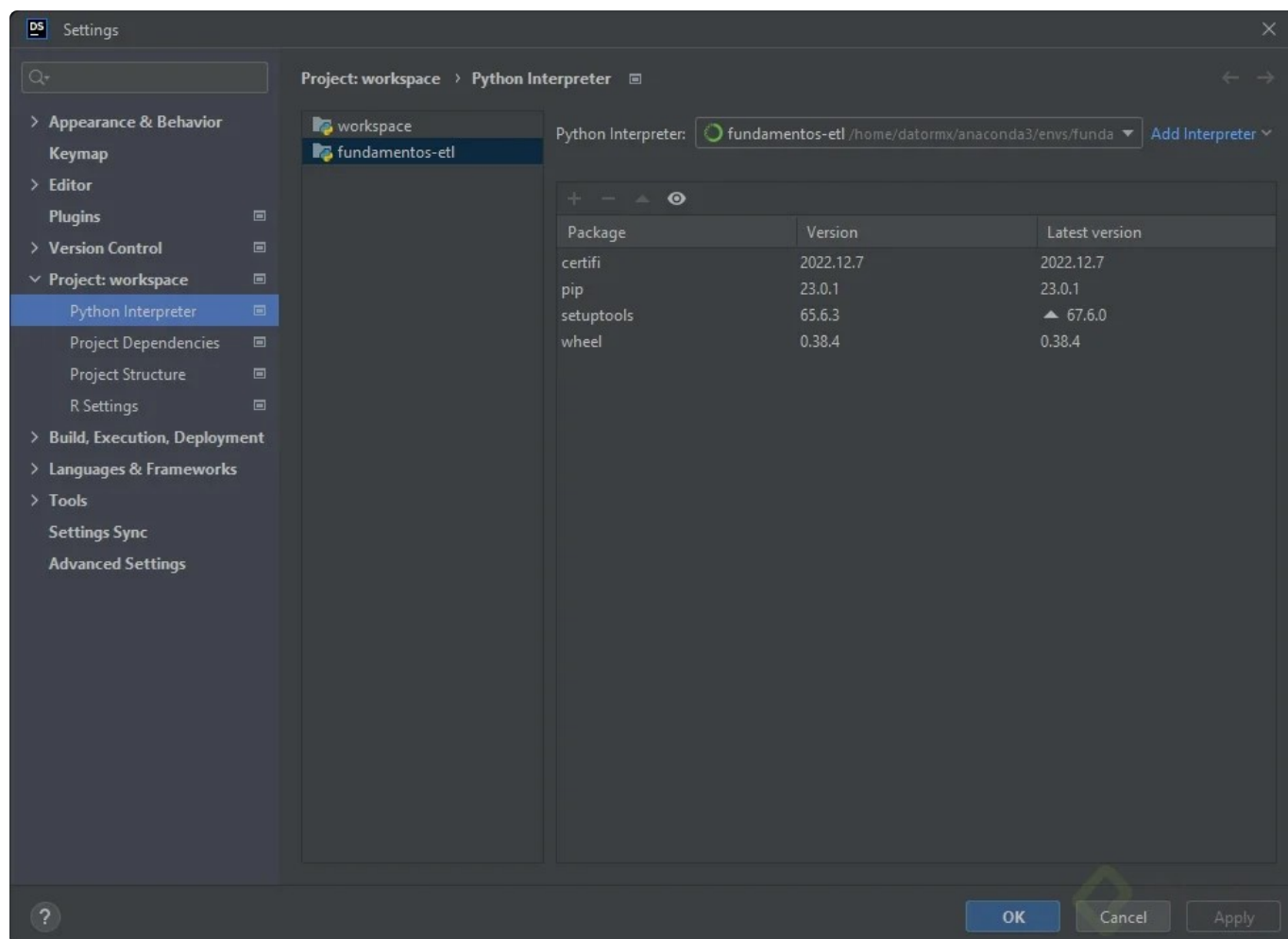


5. Marca la casilla **Use existing environment**. Elige el **Conda Environment** de WSL que usarás para tu proyecto. Anteriormente debiste crearlo desde tu terminal en WSL y llamarlo `fundamentos-etl`.

Finalmente, presiona el botón azul **Create**.



6. Para terminar el proceso presiona el botón azul **OK** en la parte inferior.



7. Listo, ya deberá aparecer tu entorno de Anaconda en WSL cargado en la parte inferior de DataSpell.



⚠ Si te aparece un error que indique que el ambiente no puede ser usado como el intérprete del workspace es porque estás intentando cargar el ambiente en el workspace general y no en un workspace de DataSpell que creaste.

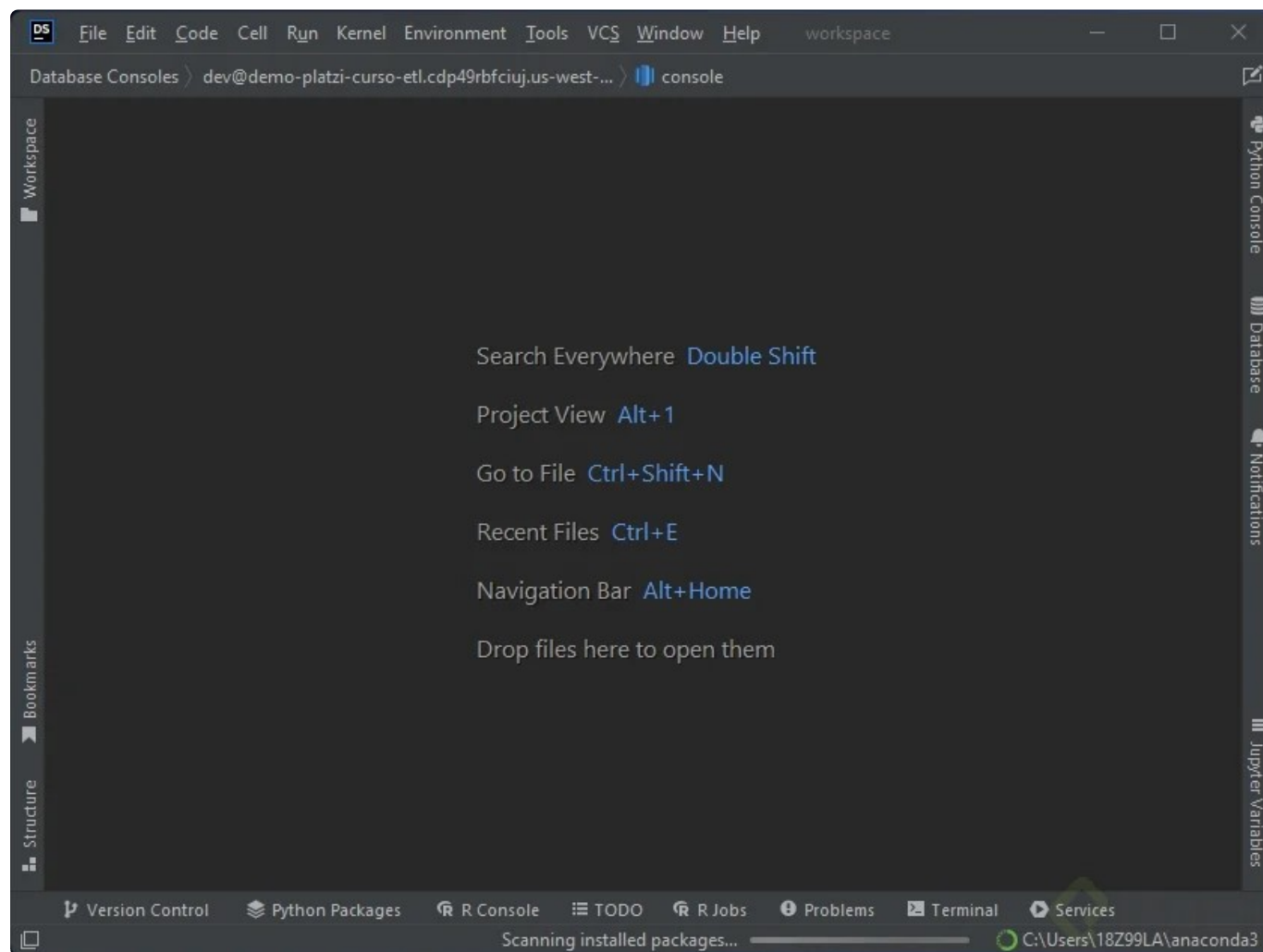
[Aquí](#) encuentras la guía oficial de cómo conectar tu DataSpell al intérprete de Python o Anaconda en WSL, por si necesitas aprender a configurarlo a detalle.

Recuerda que otra alternativa en Windows es instalar [Anaconda para Windows](#) y conectar DataSpell directamente a esta versión.

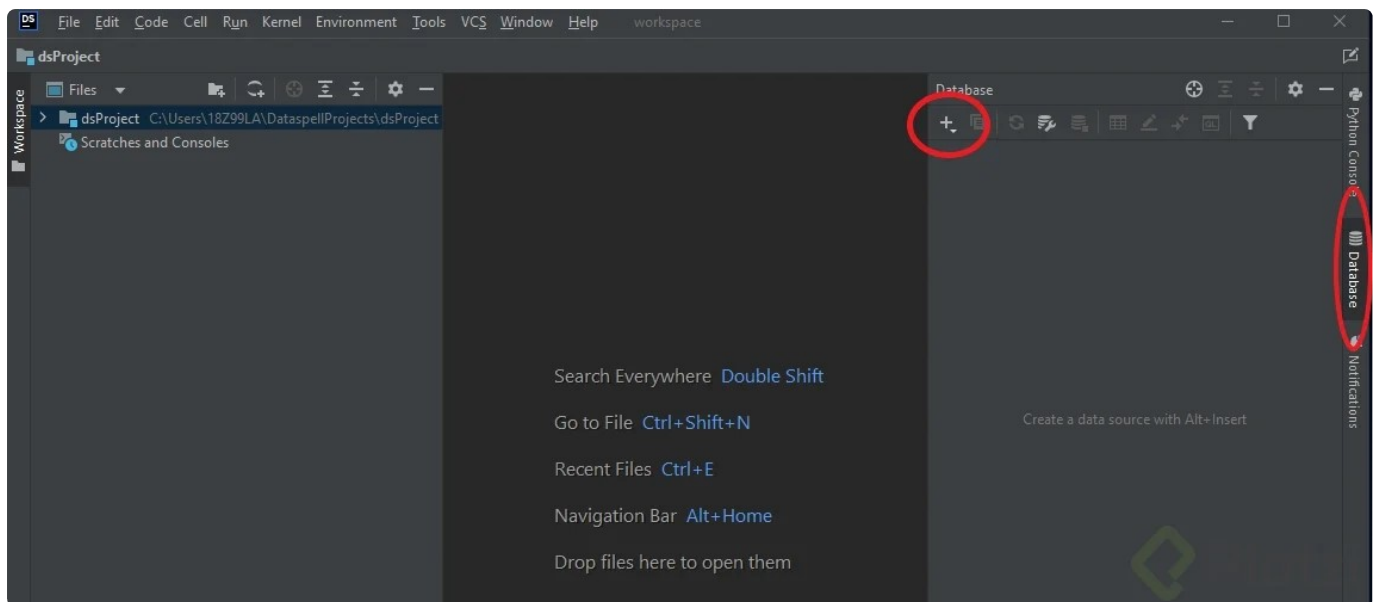
## 4. Conexión a la base de datos PostgreSQL

Sigue estos pasos para conectarte a la base de datos postgres desde DataSpell.

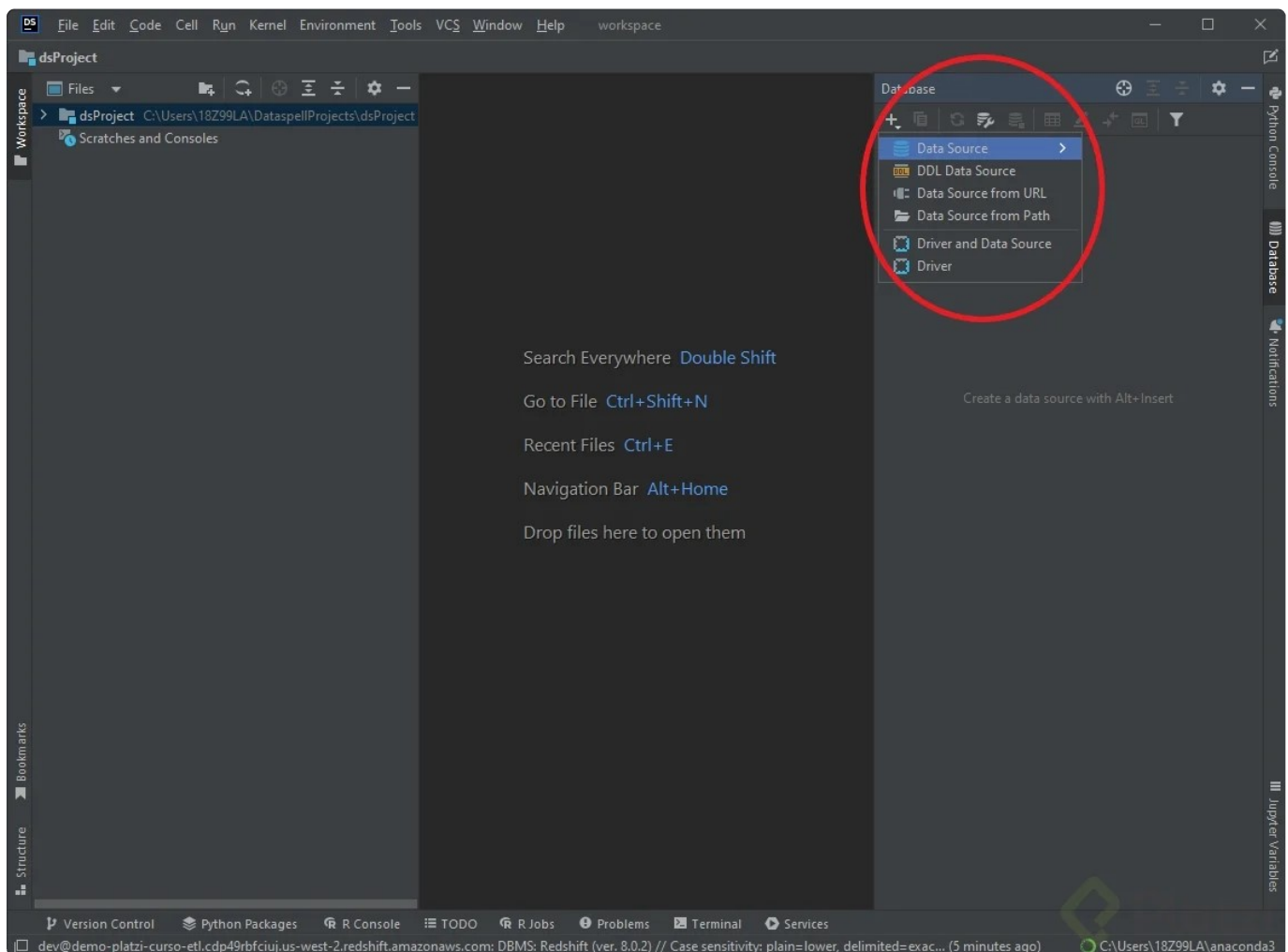
1. Abre DataSpell en tu computador.



2. Ve a la pestaña de **Database** y en ella da clic en el **botón de signo de +**.



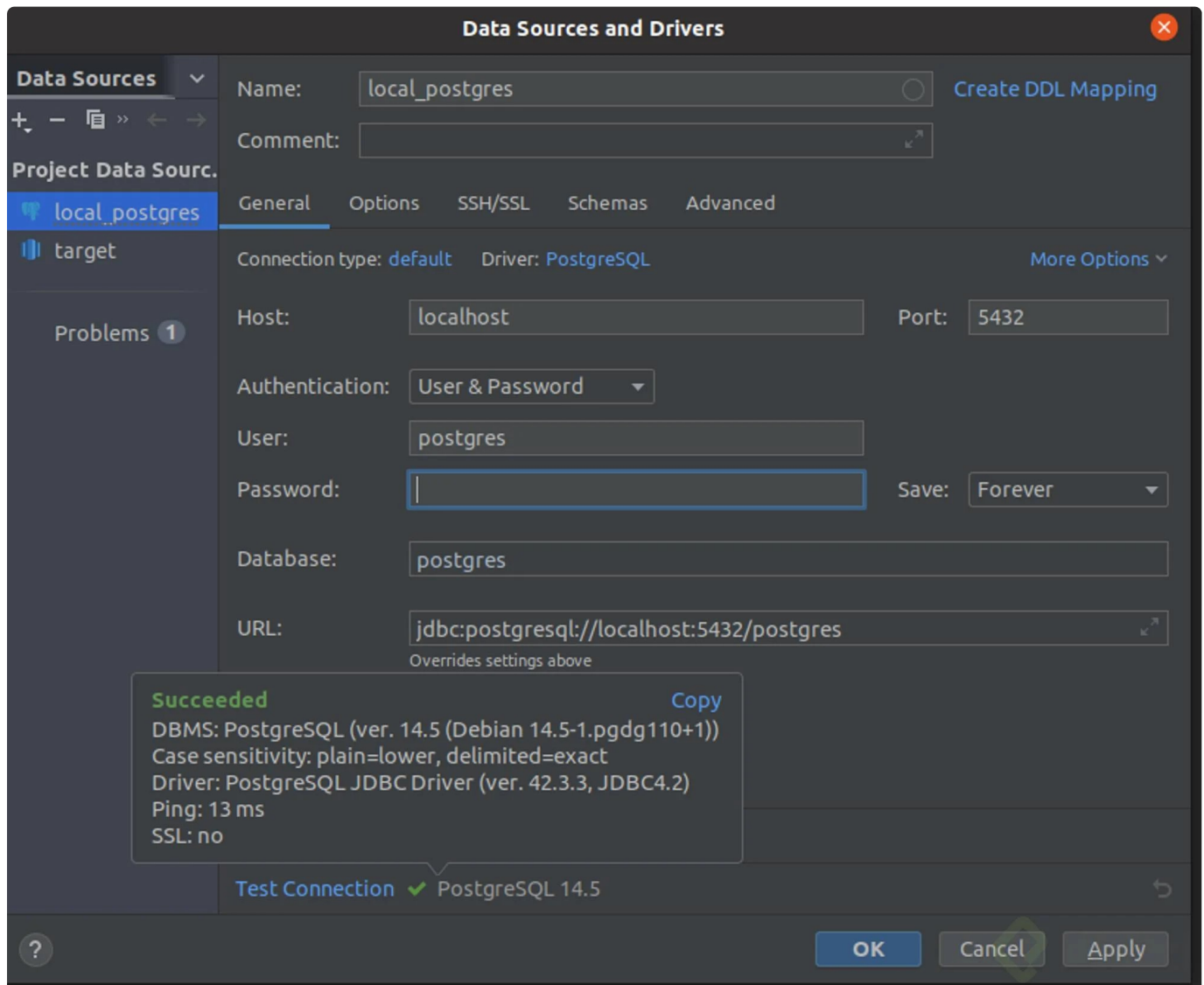
3. Selecciona la opción de **Data Source** y dentro del menú desplegable elige la opción de **PostgreSQL**.



4. Introduce los datos siguientes en la conexión:

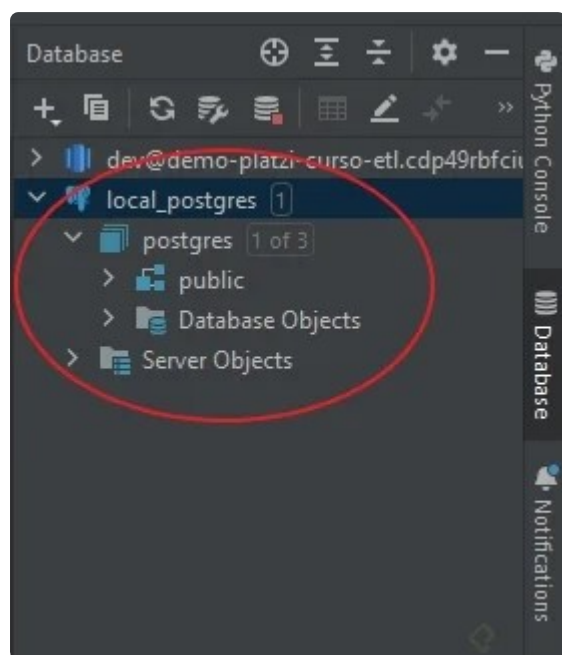
- **Name:** local\_postgres
- **Host:** localhost
- **Port:** 5432
- **User:** postgres
- **Database:** postgres
- **Url (opcional):** jdbc:postgresql://localhost:5432/postgres
- **Password:** mysecretpass

5. Da clic en el botón de **Test Connection** para probar la conexión. Puede que te solicite actualizar unos drivers, acéptalos. Una vez que indique que la conexión es exitosa, da clic en el **botón OK**.





6. Listo, ya tienes tu base de datos conectada en DataSpell.



## 4. Cargar datos en la base de datos Postgres

Dentro de DataSpell, ya con la conexión a la base de datos previamente creada, ejecutarás el script *postgres\_public\_trades.sql*.

Descárgalo [aquí de Google Drive](#). 

**⚠ Este archivo pesa cerca de 500 MB**, por lo que puede demorar su descarga.

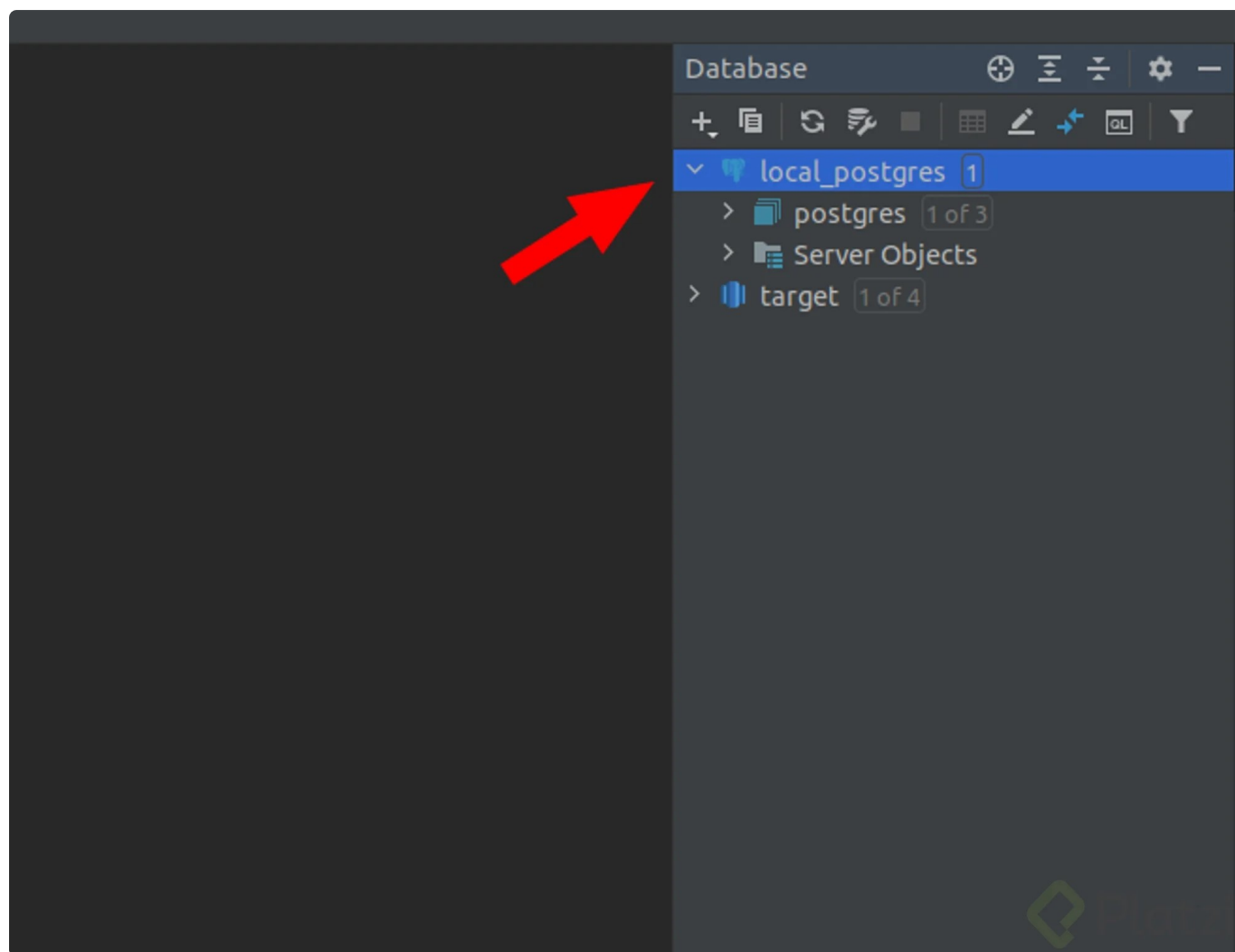
Contiene la creación de una tabla llamada *trades* y los insert de registros de la tabla.

**⚠** Es posible que al intentar **correr este script en DBeaver** no sea posible por falta de memoria. Te sugerimos cortarlo en varias partes y cargar cada script independientemente.

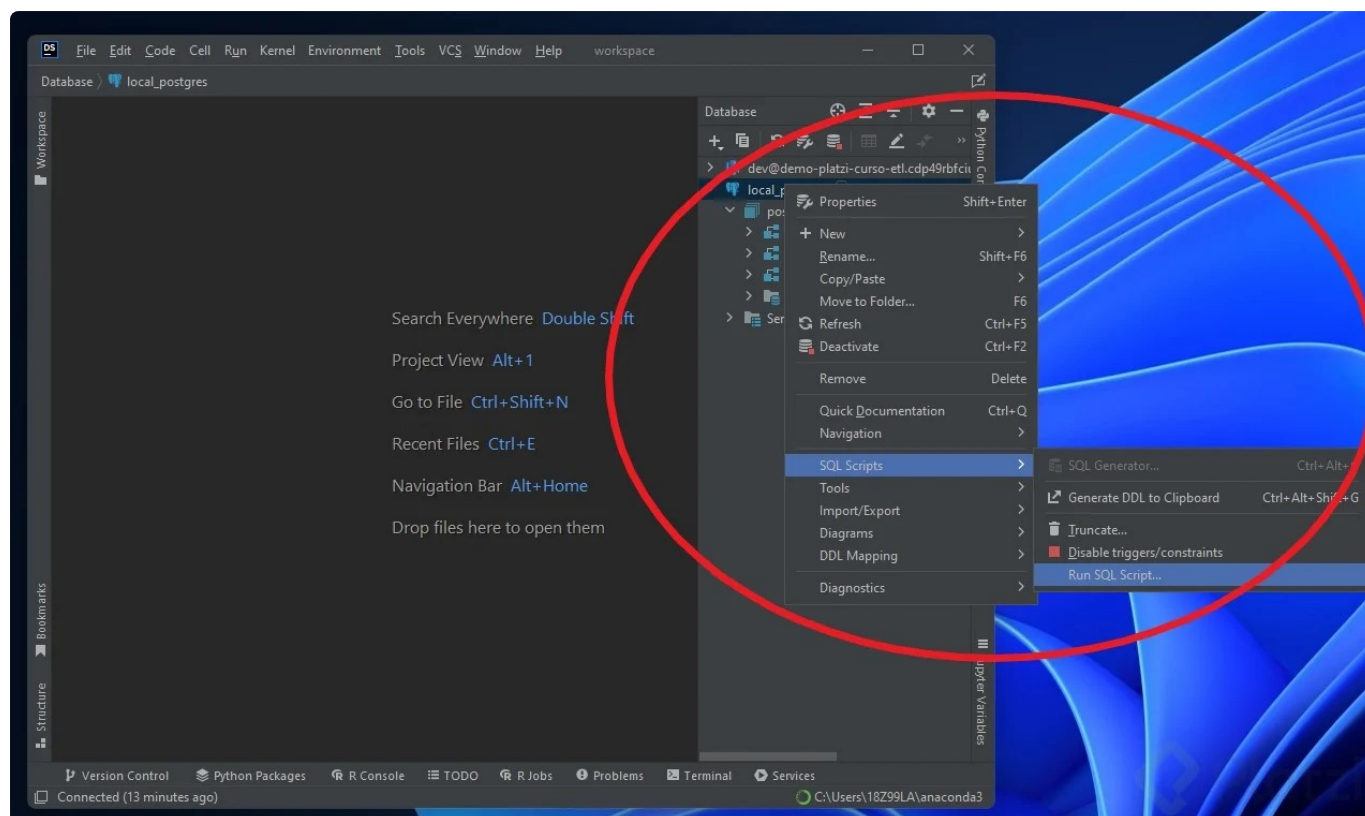
```
1 create table if not exists trades (  
2   country_code varchar(4),  
3   year int,  
4   comm_code int,  
5   flow varchar(10),  
6   trade_usd numeric(12,2),  
7   kg numeric(12,2),  
8   quantity numeric(12,2),  
9   quantity_name varchar(30)  
10 );  
11  
12 insert into public.trades (country_code, year, comm_code, flow, trade_usd, kg, quantity, quantity_name)  
13 values ('SYC', 1998, 890200, 'Import', 1431426.00, 0.00, 23000.00, 'Number of items'),  
14 ('SYC', 1998, 890310, 'Import', 31406.00, 0.00, 2545.00, 'Number of items'),  
15 ('SYC', 1998, 890310, 'Export', 950.00, 0.00, 300.00, 'Number of items'),  
16 ('SYC', 1998, 890310, 'Re-Export', 950.00, 0.00, 300.00, 'Number of items'),  
17 ('SYC', 1998, 890391, 'Import', 18251.00, 0.00, 450.00, 'Number of items'),  
18 ('SYC', 1998, 890392, 'Import', 1800859.00, 0.00, 58683.00, 'Number of items'),  
19 ('SYC', 1998, 890399, 'Import', 2100927.00, 0.00, 172909.00, 'Number of items'),  
20 ('SYC', 1998, 890520, 'Import', 201837.00, 0.00, 630000.00, 'Number of items'),  
21 ('SYC', 1998, 890710, 'Import', 19683.00, 0.00, 3160.00, 'Number of items'),  
22 ('SYC', 1998, 890790, 'Import', 221126.00, 0.00, 22169.00, 'Number of items'),  
23 ('SYC', 1998, 890800, 'Export', 9505.00, 0.00, 978.00, 'Number of items'),  
24 ('SYC', 1998, 890800, 'Re-Export', 9505.00, 0.00, 978.00, 'Number of items'),  
25 ('SYC', 1997, 890110, 'Import', 70791.00, 0.00, 5451.00, 'Number of items'),  
26 ('SYC', 1997, 890310, 'Import', 40882.00, 0.00, 3765.00, 'Number of items'),  
27 ('SYC', 1997, 890310, 'Export', 282.00, 0.00, 60.00, 'Number of items'),  
28 ('SYC', 1997, 890310, 'Re-Export', 282.00, 0.00, 60.00, 'Number of items'),  
29 ('SYC', 1997, 890392, 'Import', 1355051.00, 0.00, 56995.00, 'Number of items'),  
30 ('SYC', 1997, 890399, 'Import', 291178.00, 0.00, 11307.00, 'Number of items'),  
31 ('SYC', 1997, 890510, 'Import', 251488.00, 0.00, 4954.00, 'Number of items'),  
32 ('SYC', 1997, 890590, 'Import', 66642.00, 0.00, 5231.00, 'Number of items'),
```

Una vez descargado el archivo ***postgres\_public\_trades.sql*** sigue estos pasos para cargar los datos con DataSpell:

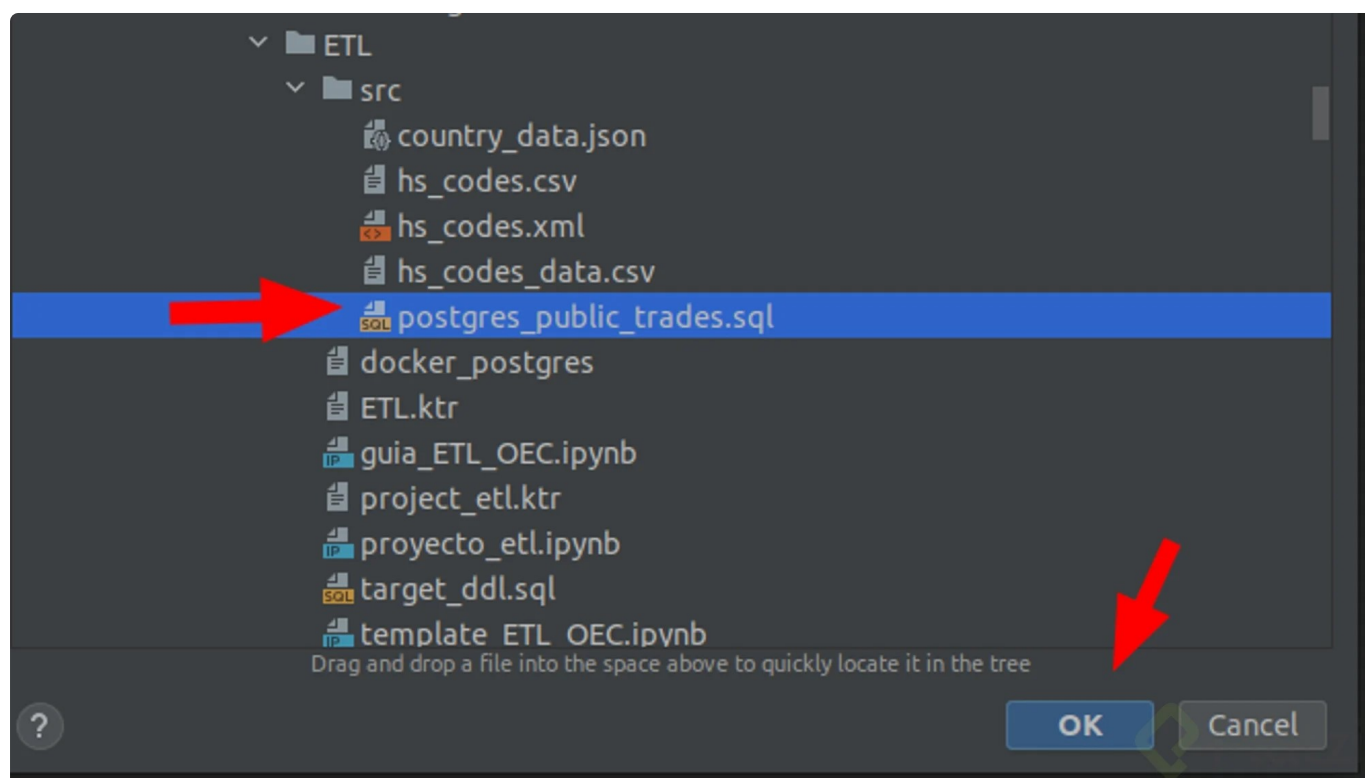
1. Da clic derecho sobre la base de datos de PostgreSQL.



2. Posteriormente da clic en **SQL Script** y luego en **Run SQL Scripts**.



3. Ubica el script descargado dentro de tu computador y da clic en **OK**.



⚠ La creación de la tabla y la carga de datos puede demorar cerca de 15-20 minutos en DataSpell.

```
Run: ▶ Run 'postgres_p..._trades.sql'... ×

[2023-03-08 12:24:57] 11587 row(s) affected in 339 ms

[2023-03-08 12:24:58] 11754 row(s) affected in 306 ms

[2023-03-08 12:24:59] 7335 row(s) affected in 204 ms

[2023-03-08 12:24:59] Summary: 567 of 567 statements executed in 12 min, 42 sec, 950 ms (579,364,865 s
```

## 5. Prueba la tabla trades

Una vez terminada la ejecución del script, consulta la tabla Trades ya cargada. Abre el editor de queries desde tu base de datos en DataSpell e ingresa la siguiente consulta:

```
SELECT * FROM trades;
```

	country_code	year	comm_code	flow	trade_usd	kg	quantity	quantity_name
1	SYC	1998	890200	Import	1431426.00	0.00	23000.00	Number of items
2	SYC	1998	890310	Import	31406.00	0.00	2545.00	Number of items
3	SYC	1998	890310	Export	950.00	0.00	300.00	Number of items
4	SYC	1998	890310	Re-Export	950.00	0.00	300.00	Number of items
5	SYC	1998	890391	Import	18251.00	0.00	450.00	Number of items
6	SYC	1998	890392	Import	1800859.00	0.00	58683.00	Number of items
7	SYC	1998	890399	Import	2100927.00	0.00	172909.00	Number of items
8	SYC	1998	890520	Import	201837.00	0.00	630000.00	Number of items
9	SYC	1998	890710	Import	19683.00	0.00	3160.00	Number of items
10	SYC	1998	890790	Import	221126.00	0.00	22169.00	Number of items
11	SYC	1998	890800	Export	9505.00	0.00	978.00	Number of items

**¡Listo!** Ya tienes lo esencial para comenzar a extraer datos de una base de datos OLTP y correr tus notebooks de Python.

**Avanza a la siguiente clase.** 