

Configuración de herramientas para Data Warehouse y ETL 13/26



Curso de Data Warehousing y Modelado OLAP



→MODELADO DIME...

¡Hola, te doy la bienvenida a este tutorial! Configurarás las bases de datos y herramientas que usaremos para el ETL y crear un data warehouse.

Usaremos **PostgreSQL** con la base de datos **Adventureworks**. Será nuestra base de datos transaccional y la fuente de información para llevar al data warehouse.

Ejecuta las siguientes instrucciones para configurar esto:

Ruby

Instalación de Ruby en Ubuntu o WSL con Ubuntu

1. Abre la terminal de Ubuntu
2. Ejecuta el siguiente comando en la terminal para actualizar la lista de paquetes disponibles:

```
sudo apt-get update
```

3. Una vez actualizada la lista de paquetes, instala Ruby ejecutando el siguiente comando en la terminal:

```
sudo apt-get install ruby-full
```

4. Verifica que Ruby se haya instalado correctamente ejecutando `ruby -v` en la terminal.

Instalación de Ruby en Windows

1. Descarga el instalador de Ruby desde la página oficial de Ruby para Windows:
<https://rubyinstaller.org/downloads/>
2. Selecciona la versión de Ruby que deseas instalar.
3. Ejecuta el instalador y sigue las instrucciones del asistente de instalación.

4. Una vez completada la instalación, abre la línea de comandos de Windows (cmd.exe) y escribe `ruby -v` para verificar que la instalación se haya realizado correctamente.

Instalación de Ruby en macOS

1. Abre la terminal de macOS.
2. Instala Homebrew ejecutando el siguiente comando en la terminal:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```



3. Una vez instalado Homebrew, ejecuta el siguiente comando en la terminal para instalar Ruby:


```
brew install ruby
```

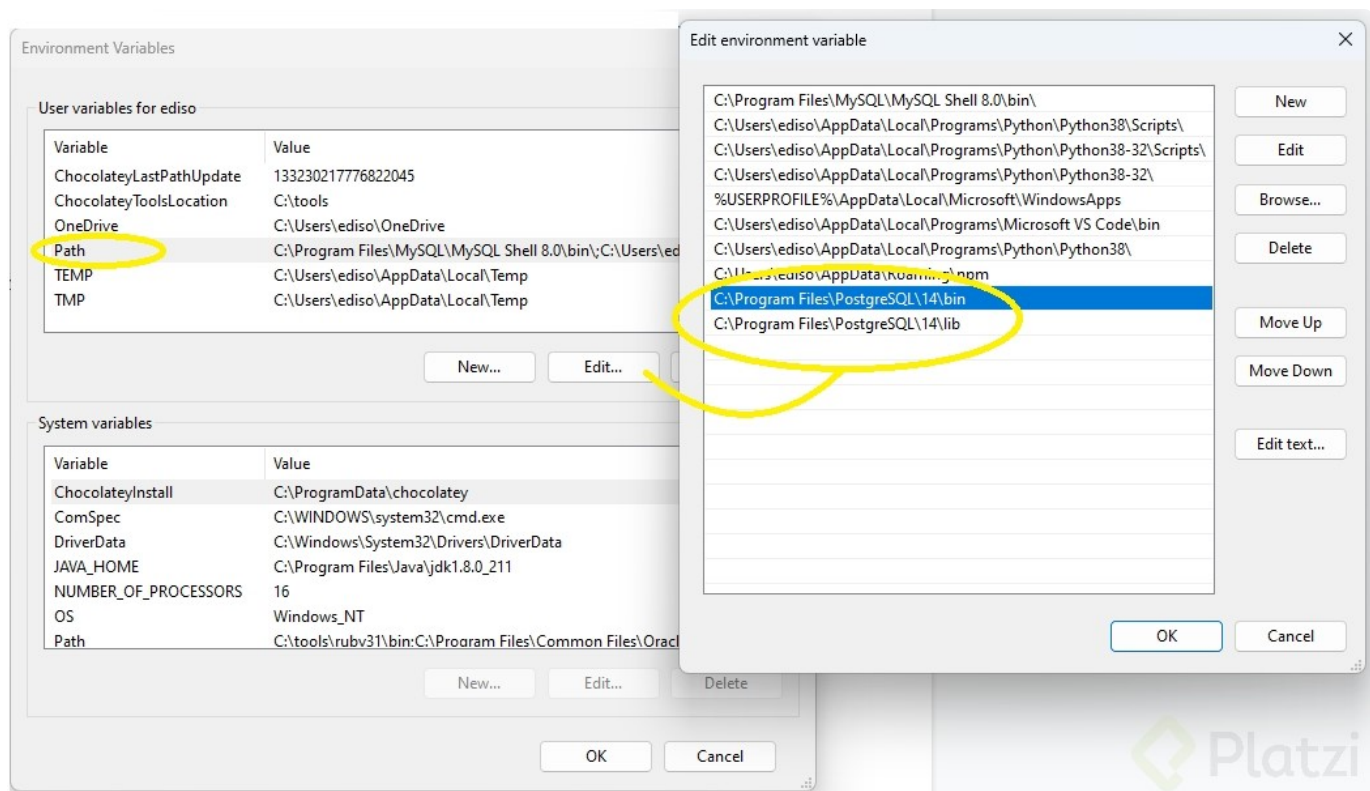
4. Verifica que Ruby se haya instalado correctamente ejecutando `ruby -v` en la terminal.

Con estos pasos ya has instalado Ruby.

PostgreSQL y pgAdmin o DBeaver

Estas herramientas ya deberías tenerla instaladas. Si no las tienes, vuelve a revisar [esta clase tutorial](#) o sigue la [documentación de PostgreSQL](#).  

 **Nota:** si usas Windows recuerda asignar las variables de entorno para PostgreSQL.



Descarga y configuración de la base de datos AdventureWorks

1. Descarga el repositorio en <https://github.com/lorint/AdventureWorks-for-Postgres>

Ejecuta el siguiente comando de Git:

```
git clone https://github.com/lorint/AdventureWorks-for-Postgres.git
```

Este repositorio contiene los archivos para crear las tablas y vistas de la base de datos.

2. Descarga [Adventure Works 2014 OLTP Script](#).

Contiene los archivos para llenar las tablas de la base de datos.

3. Copia y pega el archivo ***AdventureWorks-oltp-install-script.zip*** en el directorio ***AdventureWorks-for-Postgres***.

4. En tu terminal úbate en el directorio ***AdventureWorks-for-Postgres*** y descomprime ***AdventureWorks-oltp-install-script.zip***:

```
cd AdventureWorks-for-Postgres/  
unzip AdventureWorks-oltp-install-script.zip
```

5. En la terminal, ubicándote en el directorio ***AdventureWorks-for-Postgres***, ejecuta el siguiente comando para convertir los archivos csv:

```
ruby update_csvs.rb
```

6. Activa la conexión con postgresql:

```
sudo service postgresql start
```

7. Crea la base de datos con el siguiente comando de PostgreSQL:

```
psql -c "CREATE DATABASE \"Adventureworks\";"
```

O

```
psql -c "CREATE DATABASE \"Adventureworks\";" -U postgres -h localhost
```

8. Ejecuta el script que llena las tablas de la base de datos:

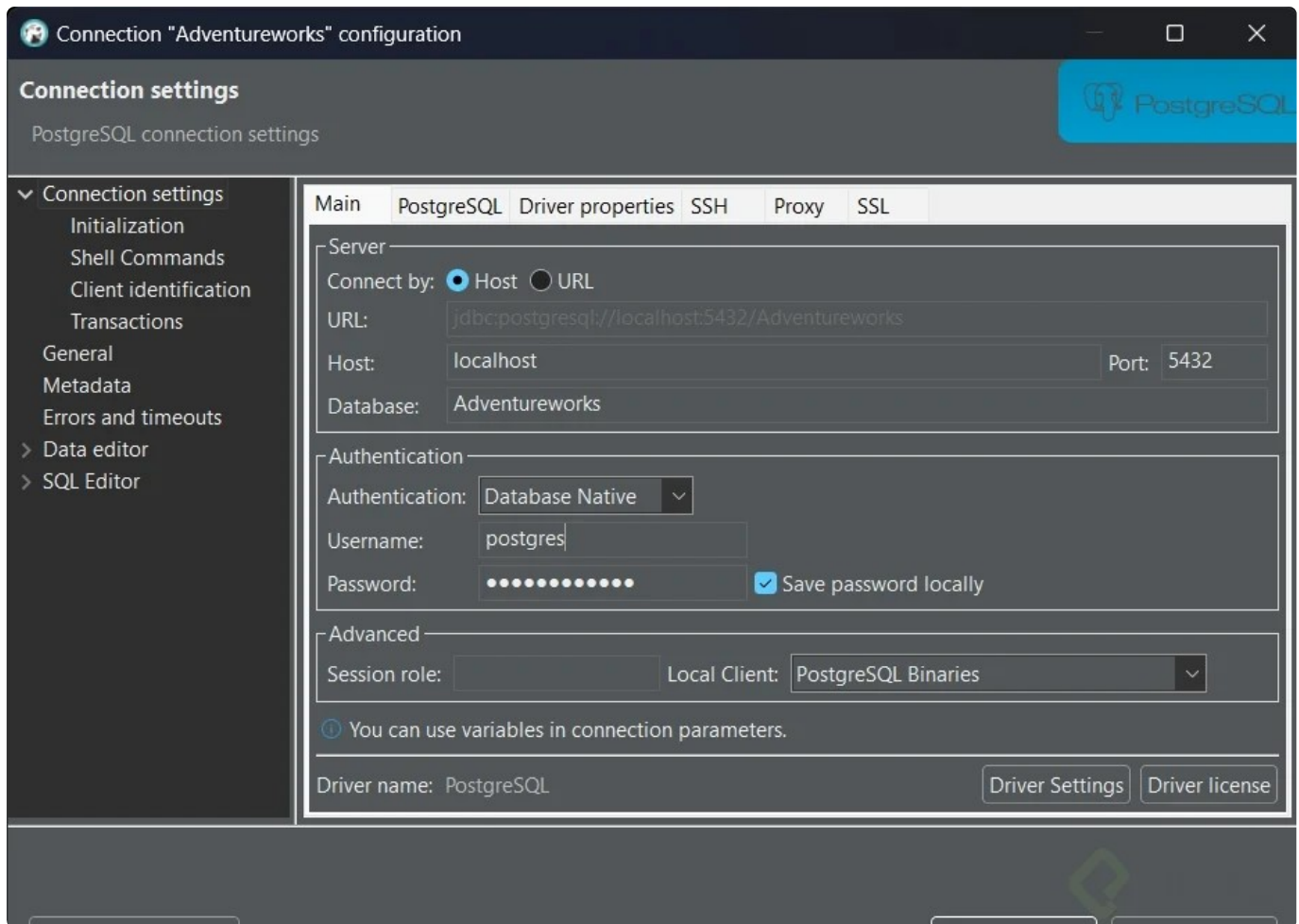
```
psql -d Adventureworks < install.sql
```

O

```
psql -d Adventureworks < install.sql -U postgres -h localhost
```

9. Conecta tu base de datos en DBeaver o pgAdmin.


1. Abre DBeaver o pgAdmin.
2. Selecciona la opción para crear una nueva conexión.
3. Selecciona **PostgreSQL** en la lista de bases de datos.
4. Ingresa la información de conexión necesaria en la pestaña.
 - Host: **localhost**
 - Port: **5432**
 - Base de datos: **Adventureworks**
 - Nombre de usuario: **postgres**
 - Password: la que tengas de tu user de postgresql.



5. Haz clic en ****Test Connection**** para asegurarte de que los detalles de conexión sean correctos y que puedas conectarte a la base de datos.
6. Si la prueba de conexión es exitosa, haz clic en "Finalizar" para guardar la configuración de la conexión.


Configuración de Pentaho

Esta herramienta la utilizaremos para crear las ETL de los datos transaccionales (DB Adventureworks) en Postgres a el Data Warehouse en AWS Redshift.

Esta herramienta deberías tenerla instalada del [Curso de Fundamentos de ETL con Python y Pentaho](#). Si no la tienes revisa [esta clase tutorial](#). 

Instalación y configuración de AWS CLI

Este servicio lo usarás para realizar la conexión a S3 y cargar archivos planos que luego serán cargados a AWS Redshift con el comando COPY.

Esta herramienta la configuraste en el [Curso Práctico de AWS: Roles y Seguridad con IAM](#) en su módulo **SDK, CLI y AWS Access Keys**. 

Vuelve a ver esas clases o sigue la siguiente documentación de AWS si no lo tienes configurado:


- Instalar AWS CLI: <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>
- Configurar AWS CLI: <https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html>

Configuración de AWS Redshift

AWS Redshift será utilizado como data warehouse. Será el lugar donde construiremos las dimensiones, tablas de hechos y llevaremos los datos modelados y limpios que se obtuvieron del sistema transaccional.

1. Crea un nuevo clúster de AWS Redshift de manera similar al **Curso de Fundamentos de ETL con Python y Pentaho**. Puedes seguir las clases tutoriales de ese curso:

- [Configuración de clúster en AWS Redshift](#).

 Recuerda **nombrar diferente** al **clúster de AWS Redshift** y al **bucket de AWS S3** que usarás para el proyecto de este curso.

Con esto has completado la configuración de herramientas a usar en las siguientes clases del curso.