



**Universidad Nacional de Ingeniería  
Facultad de Ciencias**

**Bases de Datos**

# **Gestión de Bases de Datos**

**Víctor Melchor Espinoza**

# El cliente psql

Es el terminal interactivo de PostgreSQL, una aplicación cliente que nos permitirá conectarnos a una base de datos

psql [opciones]

Entre las opciones más usadas tenemos:

<b>-h</b>	<b>Host</b>	<b>Máquina del servidor de base de datos</b>
<b>-P</b>	<b>Puerto</b>	<b>Puerto del servidor de base de datos</b>
<b>-U</b>	<b>Usuario</b>	<b>Nombre del usuario de la base de datos</b>

# Creación de una Base de Datos

```
CREATE DATABASE nombre  
  [[WITH ][OWNER [=] propietario ]  
  [TEMPLATE[=] modelo ]  
  [TABLESPACE [=] tablespace]]  
  [CONNECTION LIMIT [=] limite_conex]]
```

Un servidor PostgreSQL puede administrar varias bases de datos. Normalmente, se usa una base de datos separada para cada usuario o proyecto. Para crear una nueva base de datos, que en este ejemplo estamos llamando **mibdatos** se utiliza el siguiente comando:

```
create database mibdatos;
```

Que producirá la siguiente respuesta:

```
CREATE DATABASE
```

Un nombre de base de datos deberá tener un primer carácter alfabético y están limitados a 63 caracteres como máximo.

# Pasos para la Creación de una Base de Datos

```
C:\WINDOWS\system32\cmd.exe - psql -U postgres

C:\Documents and Settings\Andres>cd\

C:\>cd C:\Archivos de programa\PostgreSQL\9.1\bin
```

```
C:\WINDOWS\system32\cmd.exe - psql -U postgres

C:\Archivos de programa\PostgreSQL\9.1\bin>psql -U postgres
Contraseña para usuario postgres:
psql (9.1.3)
Digite \help para obtener ayuda.
```

A continuación digite el siguiente comando para crear la base de datos **mibdatos**:

```
C:\WINDOWS\system32\cmd.exe - psql -U postgres

postgres=# create database mibdatos;
CREATE DATABASE
postgres=# _
```

# Eliminación de una Base de Datos

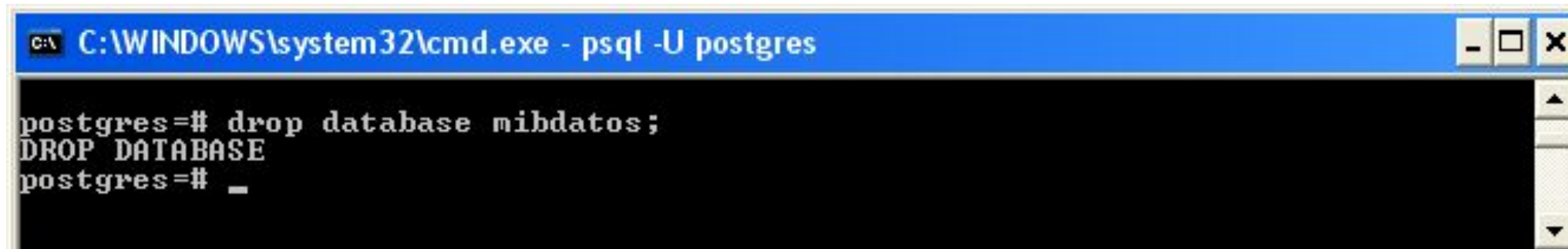
## Eliminación:

### Sintaxis:

drop database nombredb;

Para que el propietario de la base de datos pueda eliminar una base de datos, basta digitar:

drop database mibdatos;

A screenshot of a Windows command prompt window. The title bar is blue and contains the text "C:\WINDOWS\system32\cmd.exe - psql -U postgres". The command prompt area is black with white text. It shows the command "drop database mibdatos;" being entered, followed by the PostgreSQL prompt "postgres=#". The command is then executed, resulting in the output "DROP DATABASE" and a new prompt "postgres=# \_".

```
C:\WINDOWS\system32\cmd.exe - psql -U postgres  
postgres=# drop database mibdatos;  
DROP DATABASE  
postgres=# _
```

## Listado:

Para listar las bases de datos existentes basta digitar:


# Accediendo a una Base de Datos

Podemos acceder a una Base de Datos de diversas maneras:

1. Terminal interactivo del PostgreSQL (psql);
2. Herramientas gráficas para la creación, administración y manipulación de bases de datos (pgAdmin);
3. Aplicaciones desarrolladas por los usuarios utilizando conexiones vía ODBC o JDBC.

# Accediendo a una Base de Datos

El psql permite al usuario ingresar, editar y ejecutar interactivamente comandos SQL.



```
C:\WINDOWS\system32\cmd.exe - psql -U postgres mibdatos

C:\Documents and Settings\Andres>psql -U postgres mibdatos
Contraseña para usuario postgres:
psql (9.1.3)
Digite \help para obtener ayuda.

mibdatos=#
```

# Accediendo a una Base de Datos

El psql permite al usuario ingresar, editar y ejecutar interactivamente comandos SQL.



```
C:\WINDOWS\system32\cmd.exe - psql -U postgres mibdatos

C:\Documents and Settings\Andres>psql -U postgres mibdatos
Contraseña para usuario postgres:
psql (9.1.3)
Digite \help para obtener ayuda.

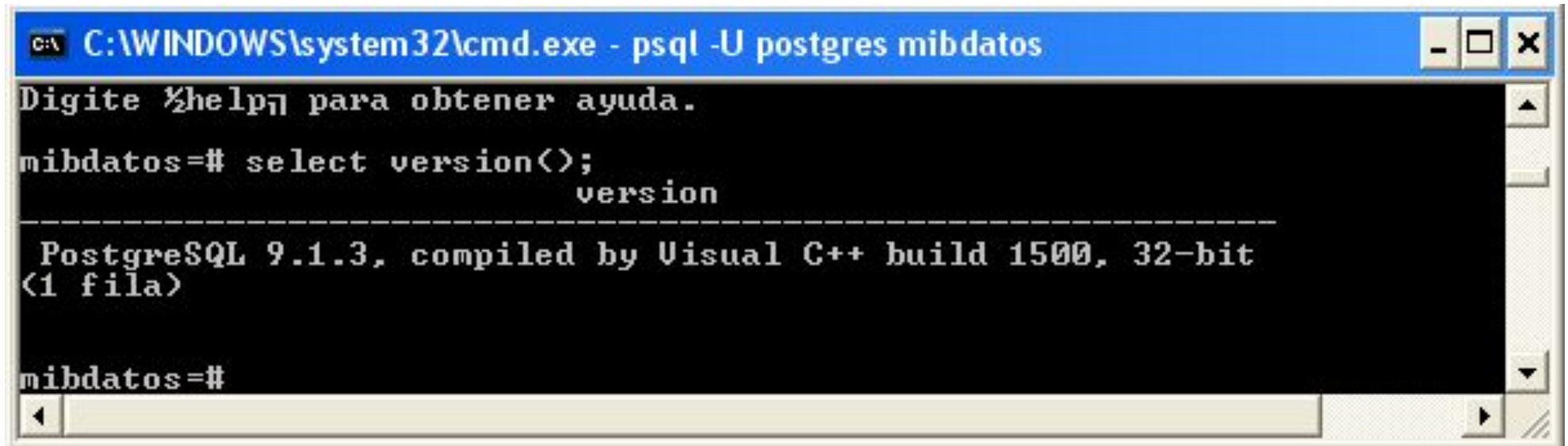
mibdatos=#
```



# Accediendo a una Base de Datos

## Verificando la versión del PostgreSQL:

SELECT version();

A screenshot of a Windows command prompt window. The title bar reads 'C:\WINDOWS\system32\cmd.exe - psql -U postgres mibdatos'. The window content shows a PostgreSQL session. It starts with the prompt 'Digite \help para obtener ayuda.' followed by the user input 'mibdatos=# select version();'. The output shows the word 'version' centered under a dashed line, followed by 'PostgreSQL 9.1.3, compiled by Visual C++ build 1500, 32-bit' and '<1 fila>'. The session ends with the prompt 'mibdatos=#' and a blank line.

```
C:\WINDOWS\system32\cmd.exe - psql -U postgres mibdatos
Digite \help para obtener ayuda.
mibdatos=# select version();
              version
-----
PostgreSQL 9.1.3, compiled by Visual C++ build 1500, 32-bit
<1 fila>
mibdatos=#
```

## Help:

\h

## Salir del PostgreSQL:

\q

# Comandos Básicos del psql

Se puede crear un archivo de texto con todas las operaciones de creación de tablas y de consultas, y posteriormente ejecutarlo de la siguiente forma:

```
\i camino/NombreDelArchivosql
```

Para ver la descripción de la Tabla:

```
\d nombreDeTabla
```

Para ver todas las tablas creadas

```
\d
```

```
C:\WINDOWS\system32\cmd.exe - psql -U postgres P
Contraseña para usuario postgres:
psql (9.1.3)
Digite \help para obtener ayuda.

Pearson=# \d
Listado de relaciones

```

Esquema	Nombre	Tipo	Dueño
public	autores	tabla	postgres
public	cargos	tabla	postgres
public	edi_info	tabla	postgres
public	editoriales	tabla	postgres
public	empleados	tabla	postgres
public	tiendas	tabla	postgres
public	tituloautor	tabla	postgres
public	titulos	tabla	postgres
public	ventas	tabla	postgres

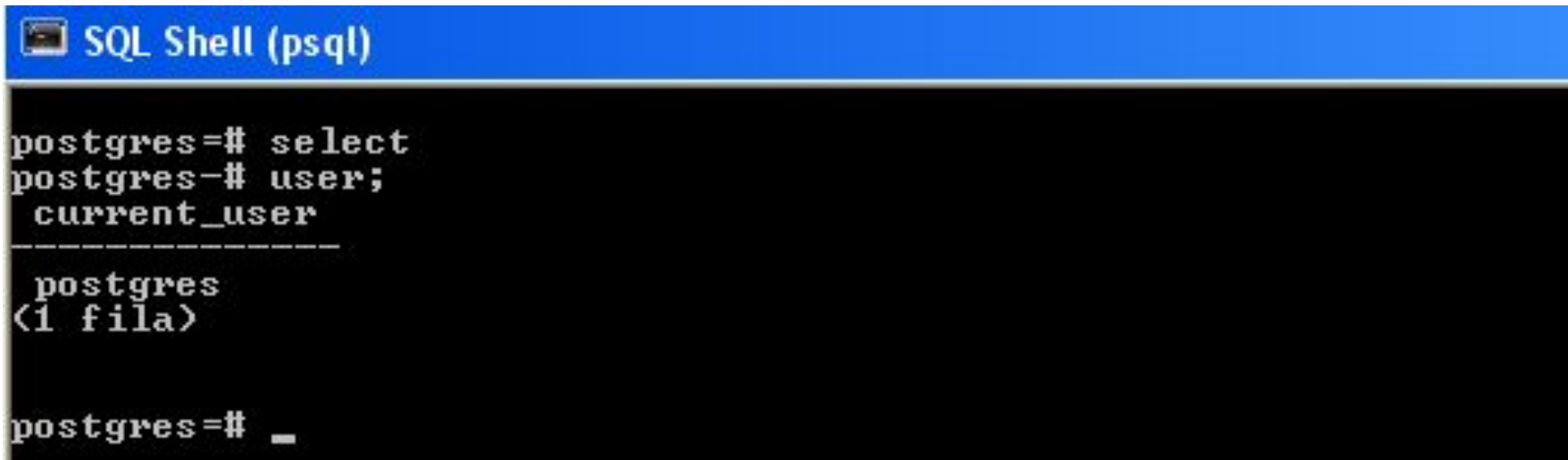
```
<9 filas>
```

# Ingreso de sentencias

Las sentencias que escribamos en el cliente deben terminar con ; o \g

```
postgres=# select user;  
current_user  
-----  
postgres  
<1 fila>  
  
postgres=# _
```

Podemos ingresar una instrucción en mas de una línea, sin embargo el indicador cambia de forma y señala el elemento que aun no ha sido completado:



```
SQL Shell (psql)  
  
postgres=# select  
postgres-> user;  
current_user  
-----  
postgres  
<1 fila>  
  
postgres=# _
```

# Indicadores de PostgreSQL

Las sentencias que escribamos en el cliente deben terminar con ; o \g

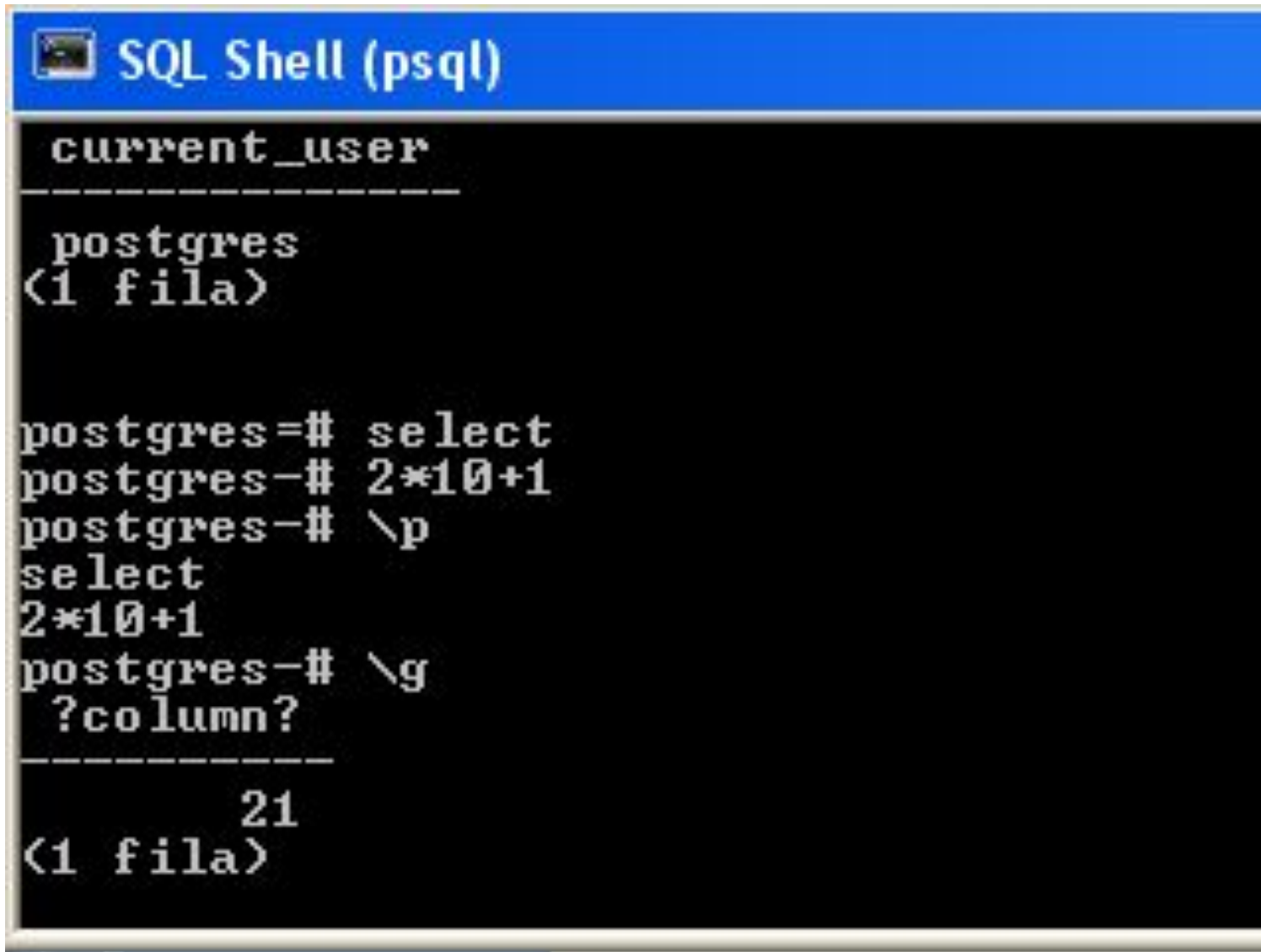
A continuación presentamos un cuadro con los indicadores que envía psql :

Indicador	Significado
=#	Espera una nueva instrucción
-#	Espera que complete la sentencia con ; o \g
"#	Espera que se cierre una cadena de comillas dobles
'#	Espera que se cierre una cadena de comillas simples
(#	Espera que se cierre un paréntesis

# Ingreso de sentencias

## Visualización del contenido del buffer:

Para revisar el contenido del buffer use la orden \p



```
SQL Shell (psql)

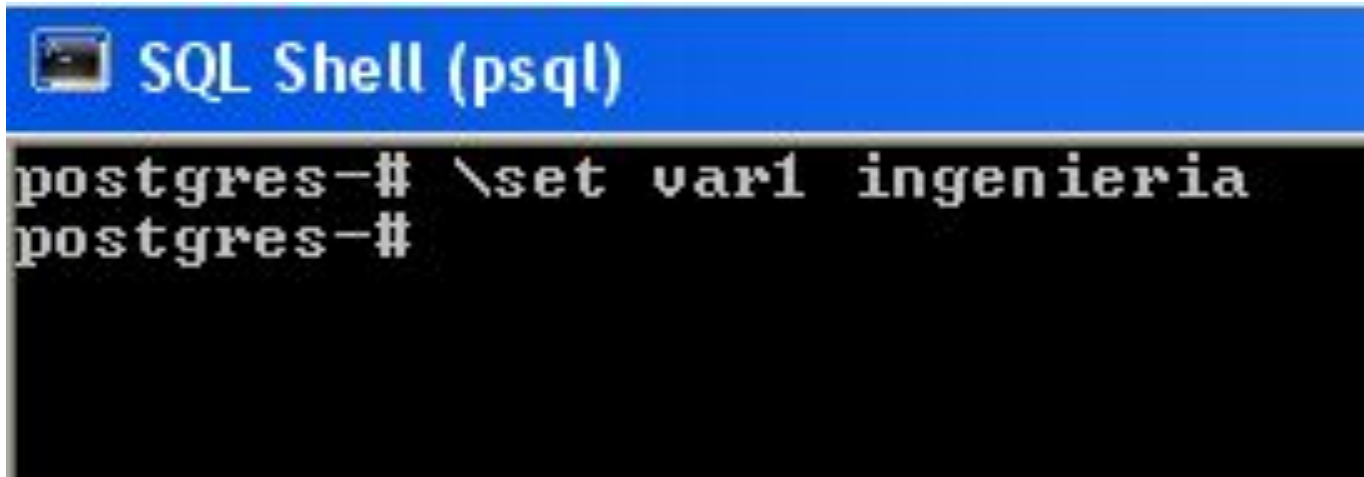
current_user
-----
 postgres
<1 fila>

postgres=# select
postgres=# 2*10+1
postgres=# \p
select
2*10+1
postgres=# \g
?column?
-----
          21
<1 fila>
```

# Expresiones y variables

## Sustitución de variables

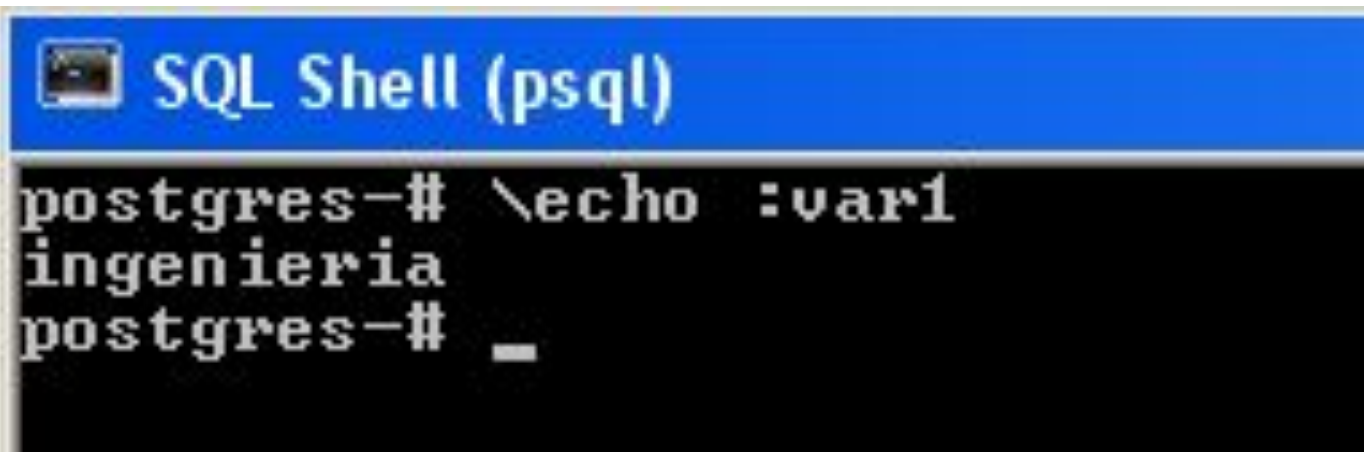
Puede asignar un valor a una variable usando \set



```
SQL Shell (psql)
postgres-# \set var1 ingenieria
postgres-#
```

## Recuperación de valores

Para mostrar el valor almacenado en una variable use la instrucción \echo

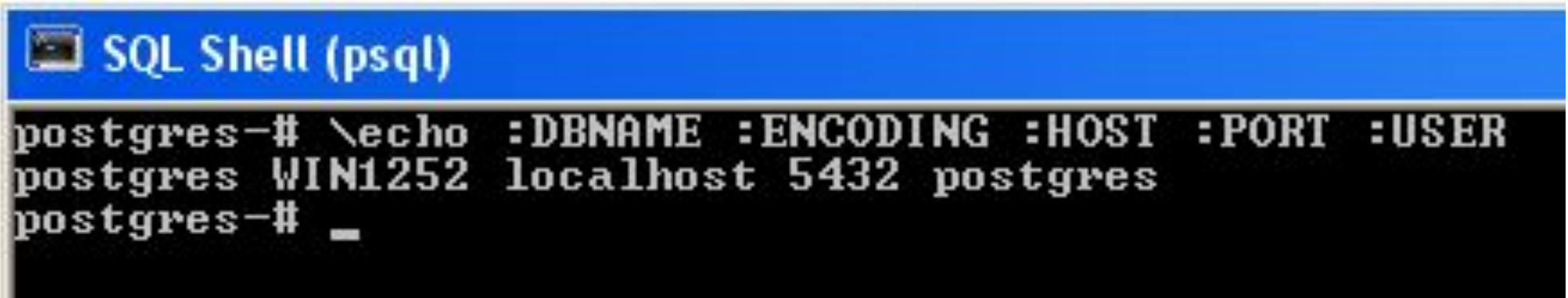


```
SQL Shell (psql)
postgres-# \echo :var1
ingenieria
postgres-# _
```

# Expresiones y variables

## Variables especiales

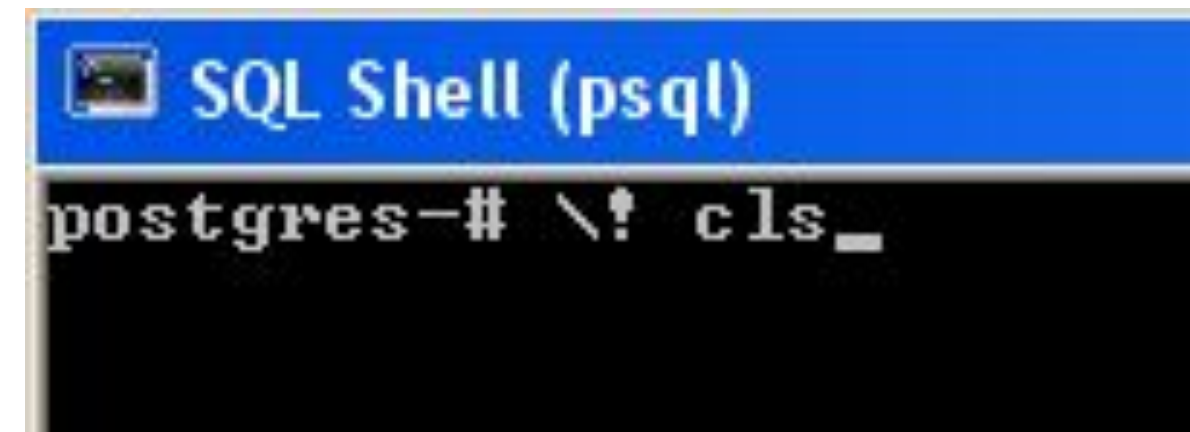
Podemos mostrar detalles del servidor al que estamos conectados

A screenshot of a terminal window titled "SQL Shell (psql)". The prompt is "postgres-#". The user has entered the command "\echo :DBNAME :ENCODING :HOST :PORT :USER". The output is displayed on the next line: "postgres WIN1252 localhost 5432 postgres". The prompt is now "postgres-# \_".

```
SQL Shell (psql)
postgres-# \echo :DBNAME :ENCODING :HOST :PORT :USER
postgres WIN1252 localhost 5432 postgres
postgres-# _
```

## Limpiar la pantalla en psql

Use el siguiente comando

A screenshot of a terminal window titled "SQL Shell (psql)". The prompt is "postgres-#". The user has entered the command "\! cls\_".

```
SQL Shell (psql)
postgres-# \! cls_
```