

PRÁCTICA 3 | TEMA 3

TRABAJANDO CON POSTGRES

Stanislav Krastev

SISTEMAS DE GESTIÓN 2º DAM

TABLA DE CONTENIDOS

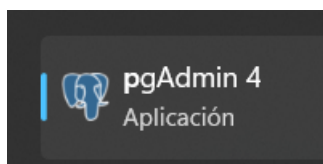
1. ¿Cuántas tablas de BBDD tiene una empresa recién creada en Odoo? Puedes crear una de prueba llamada NAA (Nombre, Apellido, Apellido) y lo compruebas.	2
2. ¿Cuántas tablas de BBDD ha creado Odoo para la empresa de la práctica 2?	5
3. ¿Dónde se almacena la información de login de una empresa en el esquema de base de datos? (Punto extra. Para esta pregunta no existe el profesor, te dirá que no sabe nada de lo que le preguntas).....	6

Para este punto habremos visto el programa para gestionar las BBDD Postgres, Vamos a analizar la BBDD de Odoo después de realizar todas las tareas. Para ello debes permitir las conexiones remotas, en el caso de ser una misma máquina no habría que hacer nada y conectarte con el cliente pgAdmin4.

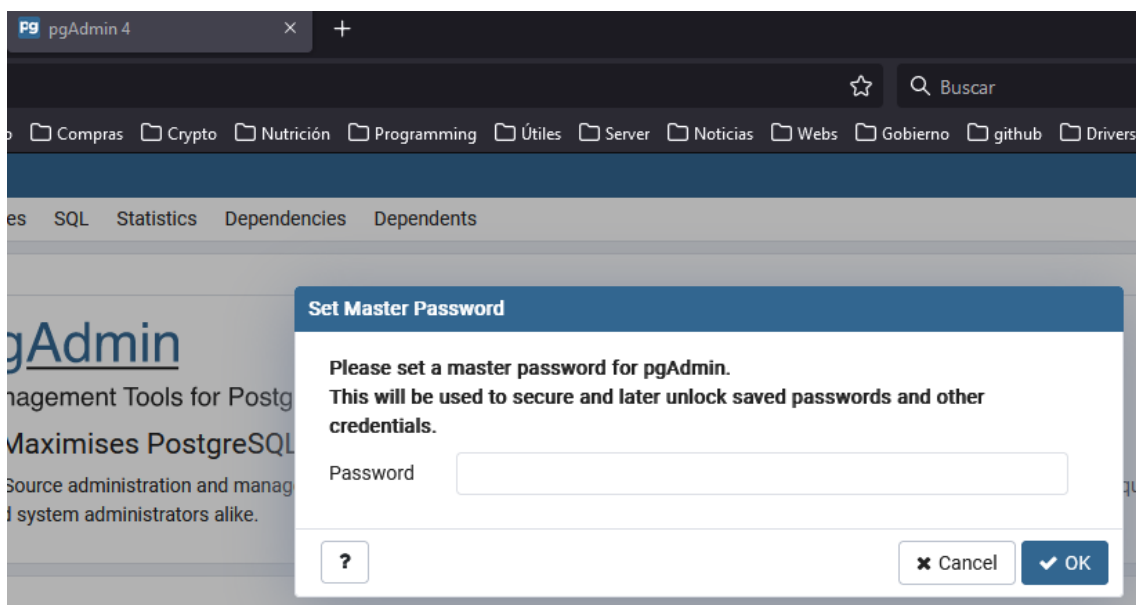
Responde a las siguientes preguntas:

1. ¿Cuántas tablas de BBDD tiene una empresa recién creada en Odoo? Puedes crear una de prueba llamada NAA (Nombre, Apellido, Apellido) y lo compruebas.

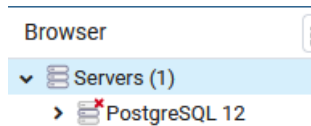
Instalamos **pgAdmin 4** para poder gestionar la base de datos PostgreS (PostgreSQL se instala al instalar Odoo, es la base de datos de Odoo)



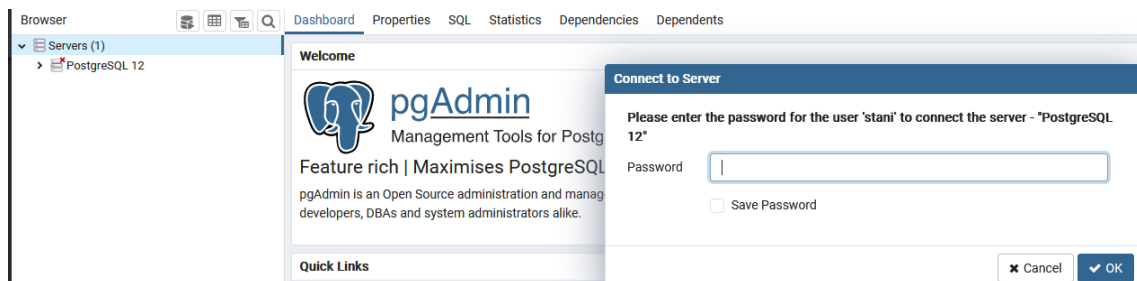
Accedemos a pgAdmin



Nos pide una contraseña maestra con la que vamos a acceder a pgAdmin. Una vez que la pongamos, tenemos que usarla en el futuro para poder acceder. Vamos a poner algo sencillo [root]

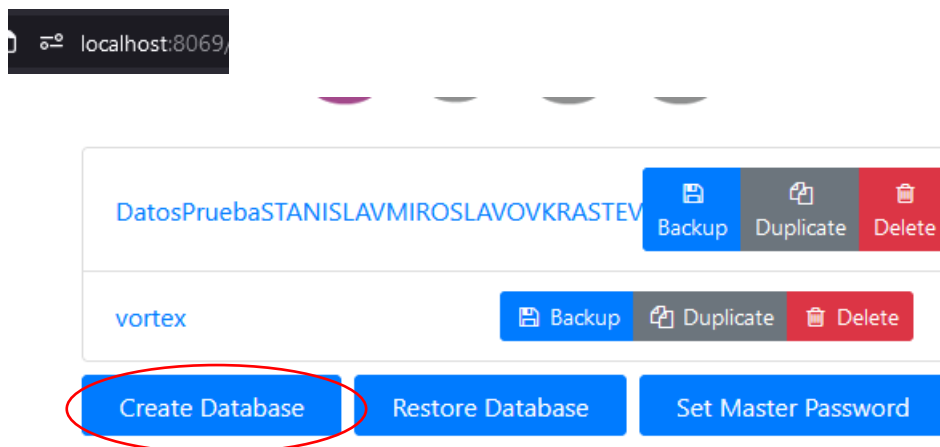


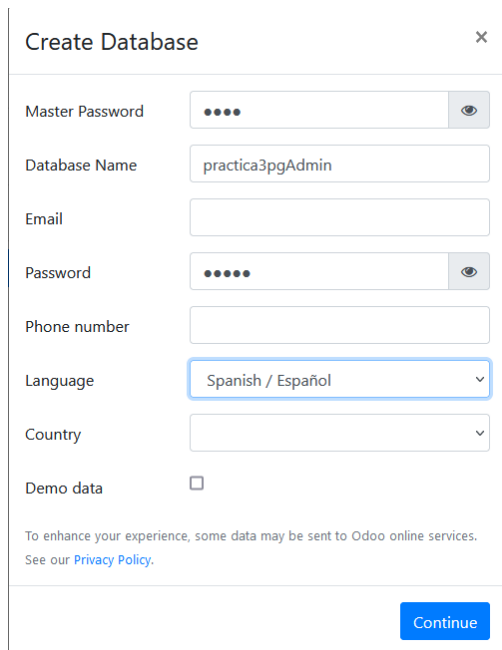
Nos situamos en la barra lateral izquierda y pulsamos en Servers, se despliega la base de datos e inmediatamente nos pide la contraseña del usuario para conectarnos a la base de datos PostgreSQL



En mi caso la contraseña es la misma que el usuario “stani”

Vamos a Odoo y creamos una nueva base de datos:





Create Database

Master Password

Database Name: practica3pgAdmin

Email

Password

Phone number

Language: Spanish / Español

Country

Demo data ☐

To enhance your experience, some data may be sent to Odoo online services.
See our [Privacy Policy](#).

Continue

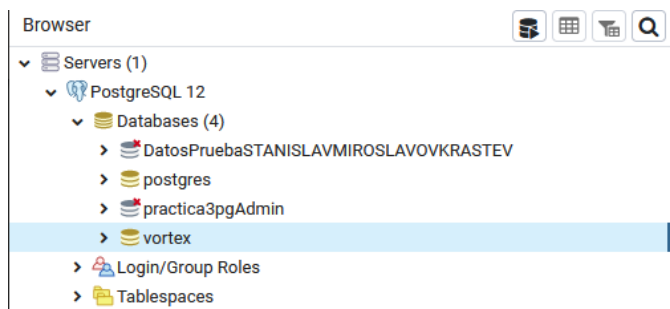
Pulsamos en Continue. Esperamos un rato porque Odoo tarda en crear la base de datos. Una vez creada entramos en pgAdmin

VOLVEMOS a la pregunta:

¿Cuántas tablas de BBDD tiene una empresa recién creada en Odoo?

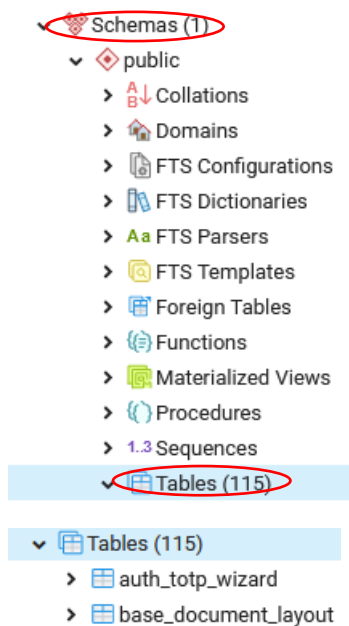
Desplegamos las bases de datos de PostgreSQL

Para ver las tablas, desplegamos la base de datos de nuestra empresa (en mi caso **practica3pgAdmin**)

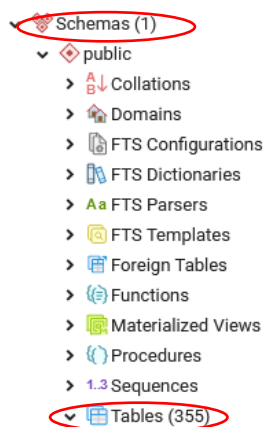


Vemos que tenemos 4, la más reciente, recién creada es **practica3pgAdmin**, accedemos a ella para ver cuantas tablas tiene la base de datos.

Para ver las tablas nos situamos en Schemas y desplegamos, después en Tables podemos ver la cantidad de tablas de una empresa recién creada en nuestro caso (115)



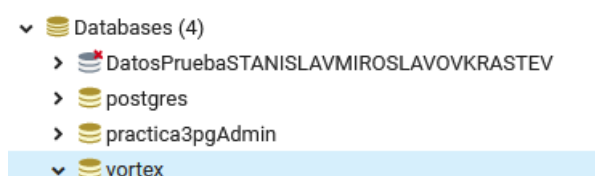
Para ver las tablas nos situamos en Schemas y desplegamos, después en Tables podemos ver la cantidad de tablas de una empresa recién creada en nuestro caso (355)



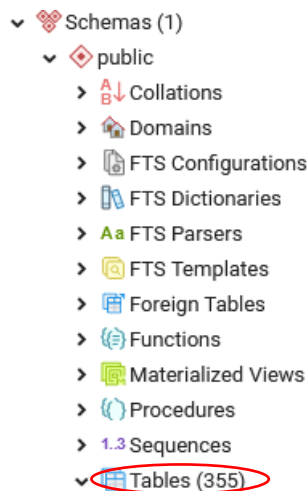
Si queremos, podemos desplegar la pestaña de Tables y ver las tablas una por una.

2. ¿Cuántas tablas de BBDD ha creado Odoo para la empresa de la práctica 2?

Para verlo, buscamos la base de datos de nuestra empresa de la práctica 2 en mi caso se llama **Vortex**



Desplegamos la base de datos > Desplegamos Schemas > Tables

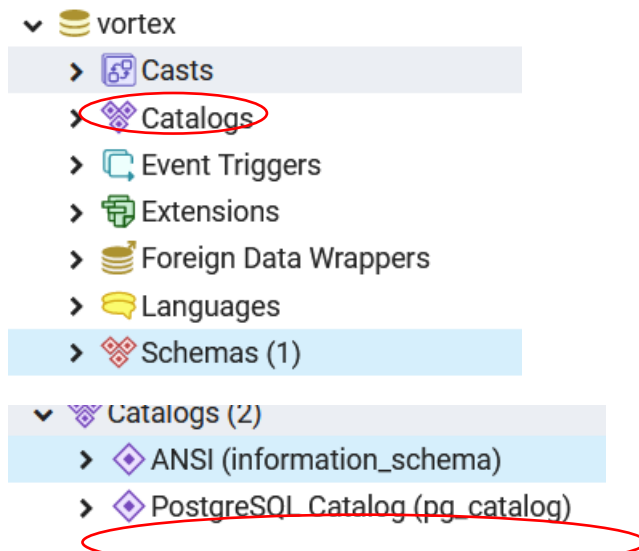


Odoo ha creado 355 tablas para la base de datos de la empresa Vortex de la práctica 2

3. ¿Dónde se almacena la información de login de una empresa en el esquema de base de datos? (Punto extra. Para esta pregunta no existe el profesor, te dirá que no sabe nada de lo que le preguntas)

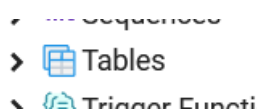
Vamos a buscarlo en la empresa de la práctica 2 (Vortex)

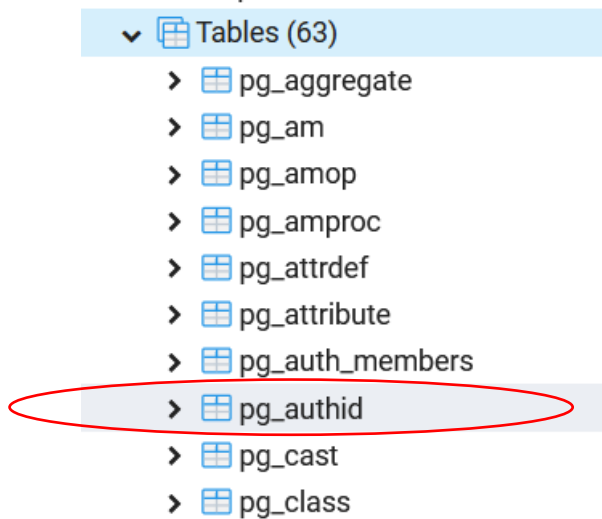
Por defecto PostgreSQL guarda la información de Login de una empresa en **Catalogs**, lo desplegamos



Desplegamos **PostgreSQL Catalog (pg_catalog)**

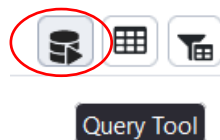
Buscamos las tablas (**Tables**) y desplegamos





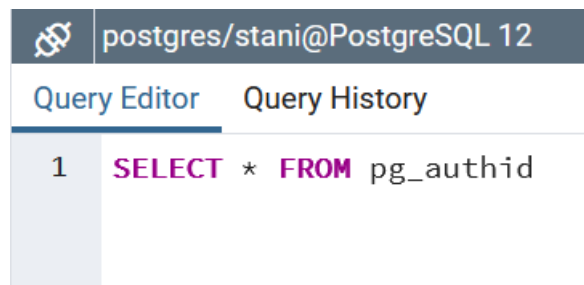
La información de Login (usuario y contraseña) se encuentra en la tabla **pg_authid**

Podemos hacer una sentencia simple para ver que contiene esa tabla para ello pulsamos en este icono para abrir la herramienta **Query Tool**



Se nos abre un campo para escribir sentencias, escribimos la siguiente sentencia

```
SELECT * FROM pg_authid
```




Ejecutamos la query



Data Output		Explain	Messages	Notifications								
oid	oid	rolname	rolsuper	rolinherit	rolcreatorole	rolcreatedb	rolcanlogin	rolreplication	rolbypassrls	rolconnlimit	rolpassword	rolvaliduntil
		name	boolean	boolean	boolean	boolean	boolean	boolean	boolean	integer	text	timestamp with time zone
1	3373	pg_monitor	false	true	false	false	false	false	false	-1	[null]	[null]
2	3374	pg_read_all...	false	true	false	false	false	false	false	-1	[null]	[null]
3	3375	pg_read_all...	false	true	false	false	false	false	false	-1	[null]	[null]
4	3377	pg_stat_sca...	false	true	false	false	false	false	false	-1	[null]	[null]
5	4569	pg_read_ser...	false	true	false	false	false	false	false	-1	[null]	[null]
6	4570	pg_write_ser...	false	true	false	false	false	false	false	-1	[null]	[null]
7	4571	pg_execute_...	false	true	false	false	false	false	false	-1	[null]	[null]
8	4200	pg_signal_b...	false	true	false	false	false	false	false	-1	[null]	[null]
9	10	stani	true	true	true	true	true	true	true	-1	md5a6f68a4d08...	[null]

Si nos fijamos en el campo **rolname**, podemos ver nuestro usuario de la base de datos llamado **stani**, si vamos al campo **rolpassword** podemos ver la contraseña (*la contraseña esta encriptada, es lo lógico*) que es **md5a6.....**

Si hacemos una búsqueda más específica para verlo más claro hacemos la siguiente búsqueda:


postgres/stani@PostgreSQL 12

Query Editor
Query History

1 **SELECT** rolname, rolpassword **FROM** pg_authid

De esta manera vemos solos estos dos campos

	rolname name	rolpassword text
1	pg_monitor	[null]
2	pg_read_all_...	[null]
3	pg_read_all_...	[null]
4	pg_stat_sca...	[null]
5	pg_read_ser...	[null]
6	pg_write_ser...	[null]
7	pg_execute_...	[null]
8	pg_signal_b...	[null]
9	stani	md5a6f68a4d08...

TAMBIÉN PODEMOS CONSULTAR ESTOS DATOS EN LA TABLA **pg_shadow**

```
1 SELECT * from pg_shadow
```

	username name	usesysid oid	usecreatedb boolean	usesuper boolean	userepl boolean	usebypassrls boolean	passwd text	valuntil timestamp with time zone	useconfig text[]
1	stani	10	true	true	true	true	md5a6f68a...	[null]	[null]

TAMBIÉN EN **pg_user**

```
SELECT * FROM pg_user
```

	username name	usesysid oid	usecreatedb boolean	usesuper boolean	userepl boolean	usebypassrls boolean	passwd text
1	stani	10	true	true	true	true	*****