

Sistemas de Gestión Empresarial

UD03 - Organización, consulta y tratamiento de la información

0. Consideraciones previas	2
1. Crear tabla	2
2. Insertar registros	3
3. Consulta sencilla	4
4. Subconsultas	5
5. Las tablas de Odoo	6
5.1. Ingreso de información	6
5.1.1. Información general de la empresa	6
5.1.2. Información bancaria de la empresa	7
5.1.3. Creación de perfiles institucionales y personales	7
5.1.4. Creación de actividades	8
5.2. Sistema de contabilidad	9
5.3. Consulta a la base de datos	10

0. Consideraciones previas

Para ejecutar los pasos de esta tarea, emplearemos la máquina virtual configurada en la UD2. Una vez iniciado el sistema, accedemos a la terminal y, desde el usuario `operador_odoo`, nos situamos en el directorio `/opt/odoo` y, ejecutando el archivo de configuración, arrancamos nuestro servidor.

```
odoo@odoo-VirtualBox: ~  
odoo@odoo-VirtualBox:~$ su operador_odoo  
Contraseña:  
operador_odoo@odoo-VirtualBox:/home/odoo$ cd /opt/odoo  
operador_odoo@odoo-VirtualBox:/opt/odoo$ ./odoo-bin
```

1. Crear tabla

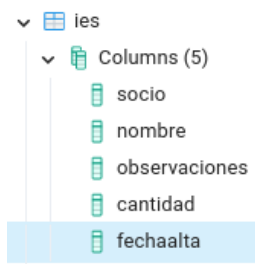
Empleando la herramienta de consulta de pgAdmin, crearemos una tabla en la base de datos.

La tabla llevará por nombre mis iniciales (IES) y contendrá cinco campos: socio, nombre, observaciones, cantidad y fechaAlta. El campo socio será la clave primaria.

```
Query Query History  
1 -- Si ya existe, la borramos  
2 DROP TABLE IF EXISTS ies;  
3  
4 -- Si no existe, la creamos  
5 CREATE TABLE IF NOT EXISTS ies  
6 ( -- [nombre] [tipo de datos] de los campos de la tabla  
7  socio serial,  
8  nombre varchar(20), -- max 65535 caracteres  
9  observaciones text, -- max 65535 bytes  
10  cantidad int,  
11  fechaAlta date,  
12  
13  primary key (socio) -- podría ir definido arriba  
14 );
```

Una vez ejecutado, nos devuelve el por consola el mensaje de la imagen: nos indica que la tabla no existe (no la elimina, como le habíamos solicitado en caso de que existiese) y procede a crearla con las indicaciones que le hemos dado.

```
Data Output Messages Notifications  
NOTICE: table "ies" does not exist, skipping  
CREATE TABLE  
  
Query returned successfully in 660 msec.
```

Esquema	Columna	Tipo de dato	Comentario
	socio	serial	Genera un valor autoincremental, es decir, no tenemos que insertar manualmente el número de socio. Se ha definido como clave primaria.
	nombre	varchar(20)	Cadena de longitud variable. Al poner el parámetro 20, definimos el máximo que puede ocupar (entre 0 y 65535 caracteres)
	observaciones	text	Cadena sin límite definido por nosotros. Admite hasta hasta 65535 bytes.
	cantidad	int	Guarda un número entero entre -2147483648 y 2147483647 (sin negativos, entre 0 y 4294967295)
	fechaAlta	date	Fecha en formato AAAA/MM/DD

2. Insertar registros

Con la tabla ya creada, el siguiente paso es insertar registros en ella. Mediante **INSERT INTO** definimos qué datos vamos a insertar y con **VALUES** especificamos qué información guardamos.

```
-- INSERCIÓN DE REGISTROS
INSERT INTO ies (nombre, observaciones, cantidad, fechaAlta)
VALUES
    ('Bach','Modelo: Stradivarius',10,'2023/03/01'),
    ('Yamaha','Modelo: XENO',20,'2023/03/02'),
    ('STOMVI','Modelo: Master',30,'2023/03/03'),
    ('Fides','Modelo: Symphony',40,'2023/03/04'),
    ('CONN','Modelo: CONNstellation 52BSP',50,'2023/03/05');
```

En este caso, estamos insertando en la tabla 5 marcas de trompetas, usando el campo de observaciones para especificar un modelo y añadiendo cantidades y fechas.

Los campos de texto van entre comillas mientras que los números admiten distintos formatos. Para las fechas, el ideal es AAAA/MM/DD, que introduciremos también entre comillas.

En Postgres, el comando **RETURNING** nos permite obtener información sobre los datos recién insertados. Por ejemplo, el número de socio junto con su nombre.

```
16 -- INSERCIÓN DE REGISTROS
17 INSERT INTO ies (nombre, observaciones, cantidad, fechaAlta)
18 VALUES
19     ('Bach','Modelo: Stradivarius',10,'2023/03/01'),
20     ('Yamaha','Modelo: XENO',20,'2023/03/02'),
21     ('STOMVI','Modelo: Master',30,'2023/03/03'),
22     ('Fides','Modelo: Symphony',40,'2023/03/04'),
23     ('CONN','Modelo: CONNstellation 52BSP',50,'2023/03/05')
24 RETURNING ies.socio, ies.nombre
25
```

Data Output Messages Notifications

	socio [PK] integer	nombre character varying (20)
1	1	Bach
2	2	Yamaha
3	3	STOMVI
4	4	Fides
5	5	CONN

3. Consulta sencilla

Un pensamiento rápido ahora mismo podría ser «no necesito consultar qué hay en la base de datos porque lo podría obtener en el paso anterior». Sin embargo, **RETURNING** únicamente me informa de los datos insertados en ese paso, mientras que mediante **SELECT** realizo la consulta sobre todos los datos almacenados, eligiendo qué y cómo quiero representar los datos obtenidos.

1

SELECT *

2

FROM ies

3

ORDER BY fechaAlta ASC;

Data Output

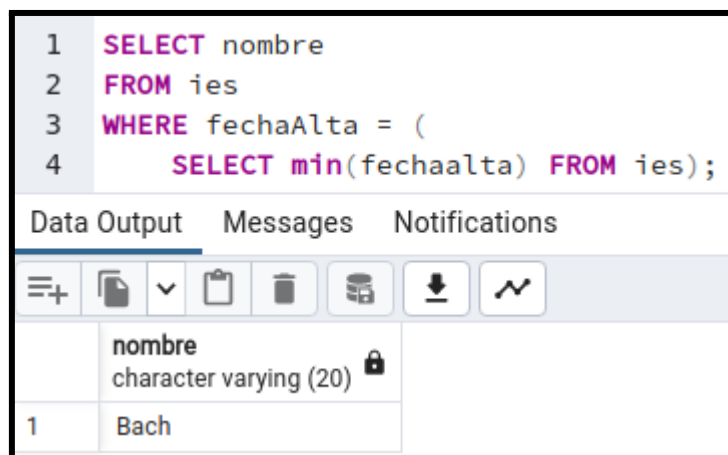
Messages

Notifications

	socio [PK] integer	nombre character varying (20)	observaciones text	cantidad integer	fechaalta date
1	1	Bach	Modelo: Stradivarius	10	2023-03-01
2	2	Yamaha	Modelo: XENO	20	2023-03-02
3	3	STOMVI	Modelo: Master	30	2023-03-03
4	4	Fides	Modelo: Symphony	40	2023-03-04
5	5	CONN	Modelo: CONNstellation 52BSP	50	2023-03-05

4. Subconsultas

Una subconsulta es una sentencia SELECT que aparece dentro de otra sentencia SELECT, de ahí que también se las conozca como consultas anidadas. Permiten obtener un subconjunto de datos sobre los que realizar la consulta principal.



```
1 SELECT nombre
2 FROM ies
3 WHERE fechaAlta = (
4     SELECT min(fechaalta) FROM ies);
```

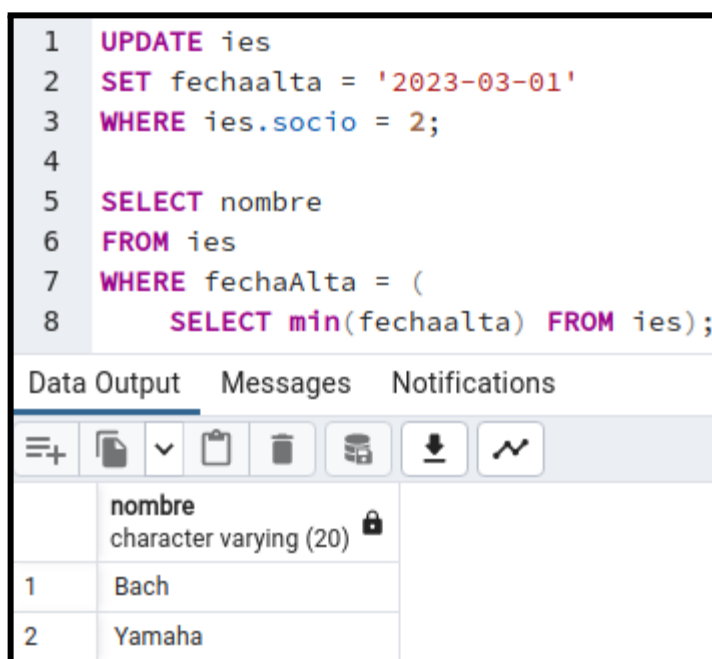
	nombre character varying (20)
1	Bach

En el caso de ejemplo, la subconsulta busca en la tabla **ies** el registro con la fecha de alta más antigua. Una vez localizado, la consulta principal muestra el nombre de los registros cuya fecha de alta coincida con la obtenida en la subconsulta.

Como veíamos antes, Bach tenía como fecha 2023-03-01, lo que la hace la más antigua.

Para comprobar su funcionamiento, podemos probar a realizar un **UPDATE** sobre alguno de los registros existentes.

En el ejemplo se ha actualizado el socio 2 para que su fecha de alta coincida con la del socio 1. Al realizar la consulta anterior, vemos que nos devuelve todos aquellos registros que cumplan la subconsulta: Bach y Yamaha.



```
1 UPDATE ies
2 SET fechaalta = '2023-03-01'
3 WHERE ies.socio = 2;
4
5 SELECT nombre
6 FROM ies
7 WHERE fechaAlta = (
8     SELECT min(fechaalta) FROM ies);
```

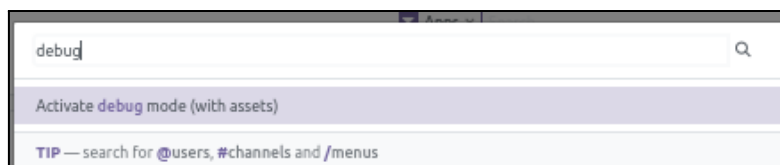
	nombre character varying (20)
1	Bach
2	Yamaha

5. Las tablas de Odoo

En este apartado reutilizo la información que había generado en la tarea descartada, ya que aparece información adicional sobre cambios realizados en los datos (asociación de empresas y trabajadores, límites de crédito, etc) y sirve como guía para entender de dónde provienen. La consulta solicitada se encuentra en el [apartado 5.3](#).

5.1. Ingreso de información

En el navegador, accedemos a Odoo desde `localhost:8089`. Una vez dentro, nuestro



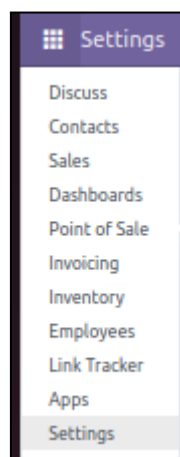
primer paso será activar el modo desarrollador: mediante el comando **Control + K**, desplegamos la paleta de comandos, donde escribimos «debug». Nos aparecerá la opción «Activate debug mode

(with assets)». Tras unos segundos, comprobamos que se ha activado el modo desarrollador al observar que aparece su icono en la barra superior.

5.1.1. Información general de la empresa

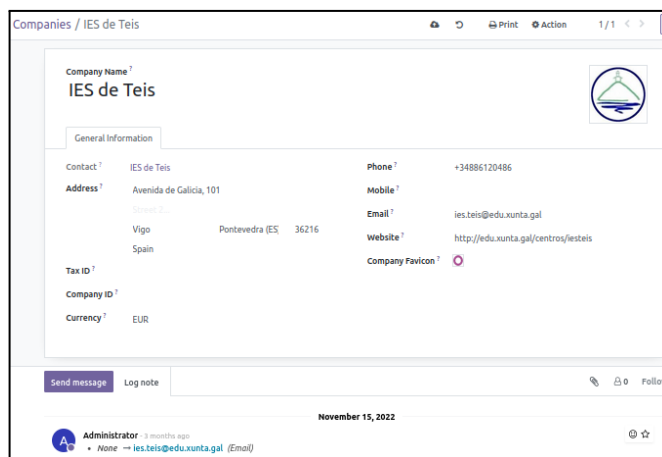
Desde el menú de configuración acceso a Usuarios y Compañías > Compañías, donde se puede editar la información de la empresa.

Ciertos datos ya se habían incorporado en el momento de la instalación, por lo que relleno los restantes, manteniendo mínimas diferencias con el material del profesor (IES Teis - IES de Teis).



Una vez guardados los cambios, compruebo que en la instalación inicial había optado por instalar ya la aplicación de contactos, por lo que tengo tanto Contactos

(Contacts) como Conversaciones (Discuss) operativas, entre otras. Desde Contactos > Configuración> Bancos, creo una nueva entidad con datos inventados salvo el código SWIFT/BIC.



5.1.2. Información bancaria de la empresa

Una vez creado el banco, desde el menú de Bank Accounts podemos crear una cuenta bancaria. Para dotar al ejercicio de un mínimo de realismo, busco el código bancario de ING, que es 1465. A partir de él, y gracias a la página «generadordni.es», creo una cuenta ficticia con el IBAN

Name	Caixa Ribadavia
Bank Identifier Code	INGDESMXXX
Bank Address	Praza Maior, 1 Ribadavia Ourense (Oren) 32400 Spain
Phone	988471154
Email	caixa@ribadavia.gal

ES8614650390195396918906. Tras cubrir el resto de datos, guardo la cuenta.

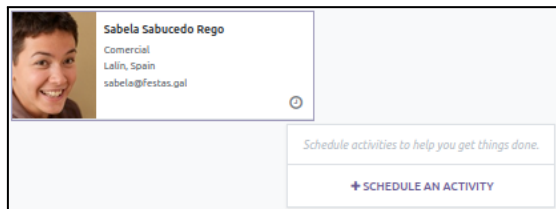
Account Number	ES8614650390195396918906	Bank	Caixa Ribadavia - INGDESMXXX
Account Holder	IES de Teis	Send Money	<input type="checkbox"/>
Account Holder Name	Iván		

5.1.3. Creación de perfiles institucionales y personales

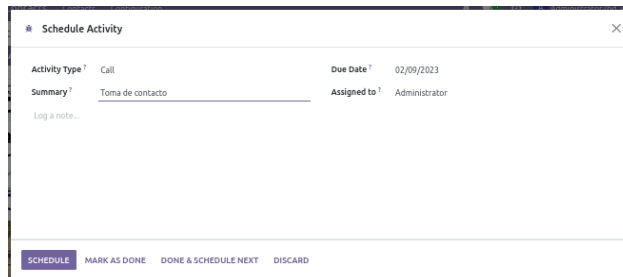
A continuación, crearemos 3 perfiles de empresa y 3 de persona, empleando información pseudoreal junto con imágenes y perfiles inventados o adaptados de la realidad, quedando la pantalla de contactos de la siguiente forma:

Name	Location	Email
A Lira de Ribadavia	Ribadavia, Spain	banda@lira.gal
Administrator		admin@example.com
Festas Gal...		hai@festas.gal
IES de Teis	Vigo, Spain	ies.teis@edu.xunta.gal
Marisol Arias Rego	Ourense, Spain	marisol@charangueo.gal
Queremos...		queremos@...
Ramón Estévez Casasnovas	A Arnoia, Spain	ramon@lira.gal
Sabela Sabucedo Rego	Lalín, Spain	sabela@festas.gal

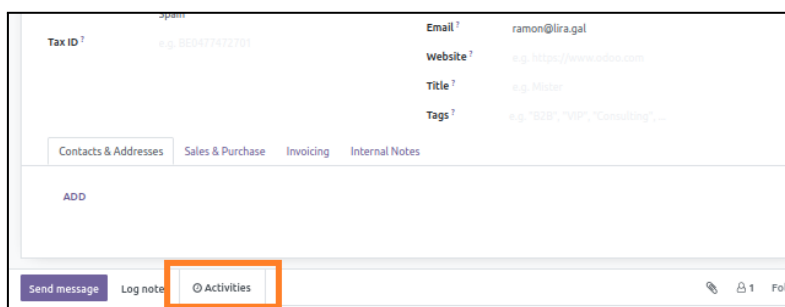
5.1.4. Creación de actividades



El siguiente paso será crear actividades. Desde el icono del reloj nos aparecerá la opción para crear una, junto con una nota informativa.

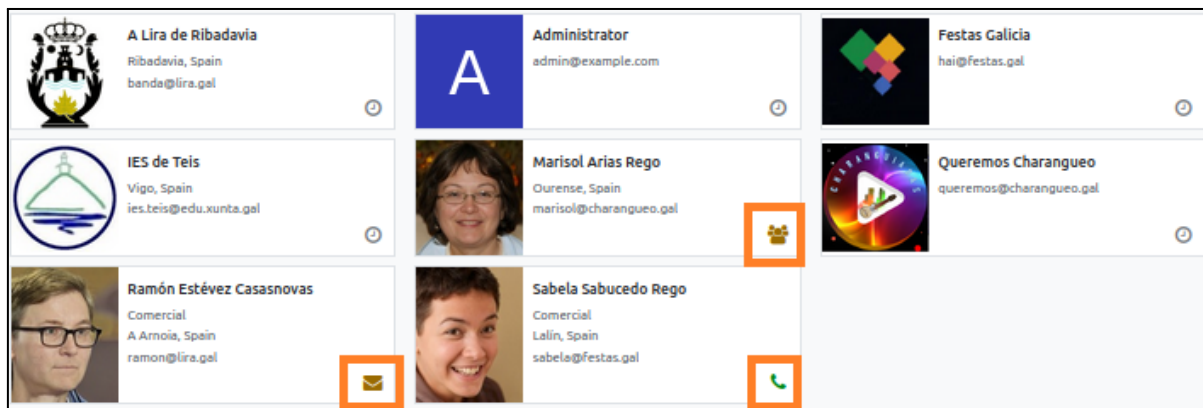


Al acceder, vemos una ventana donde se puede elegir el tipo de actividad, la fecha límite y la persona encargada; también existe la posibilidad de añadir un pequeño resumen e incluso notas.



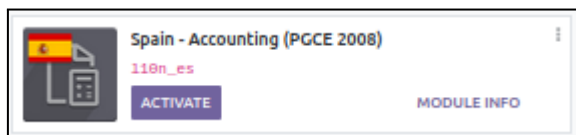
A esta sección también se puede acceder desde la parte inferior de la ficha de contacto.

Una vez creada una actividad en cada persona, se puede observar que se añade un icono identificativo donde antes figuraba el reloj.

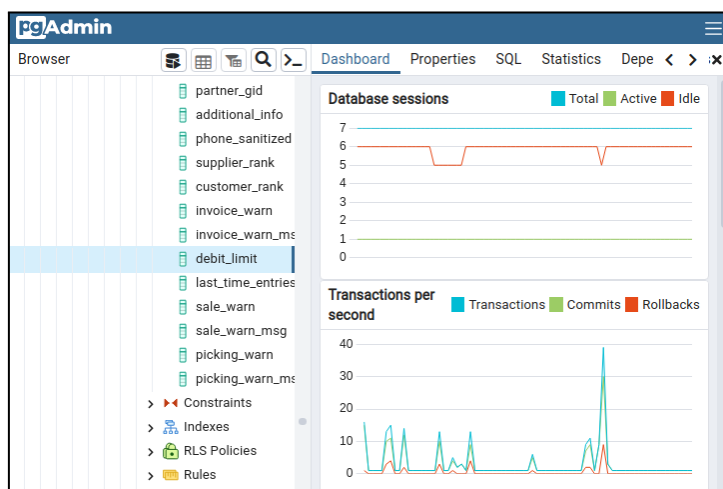
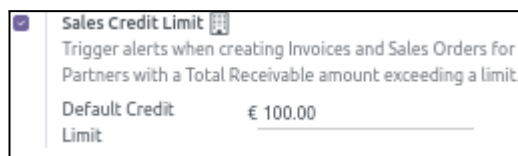


5.2. Sistema de contabilidad

Para añadir el módulo de contabilidad adaptado al sistema fiscal español, desde el buscador de aplicaciones únicamente habrá que escribir PGCE y esperar a que aparezca el módulo de la imagen. Una vez instalado, se añadirán más secciones al menú general de Odoo.

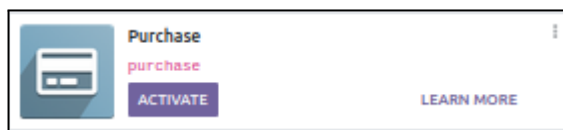


Para la realización del primer punto de la tarea se solicita, entre otros puntos, el crédito concedido. Para visualizarlo, desde el apartado de Facturas únicamente hay que dirigirse a Configuración y activar el apartado Límite de Crédito de Ventas, añadiendo una cantidad.

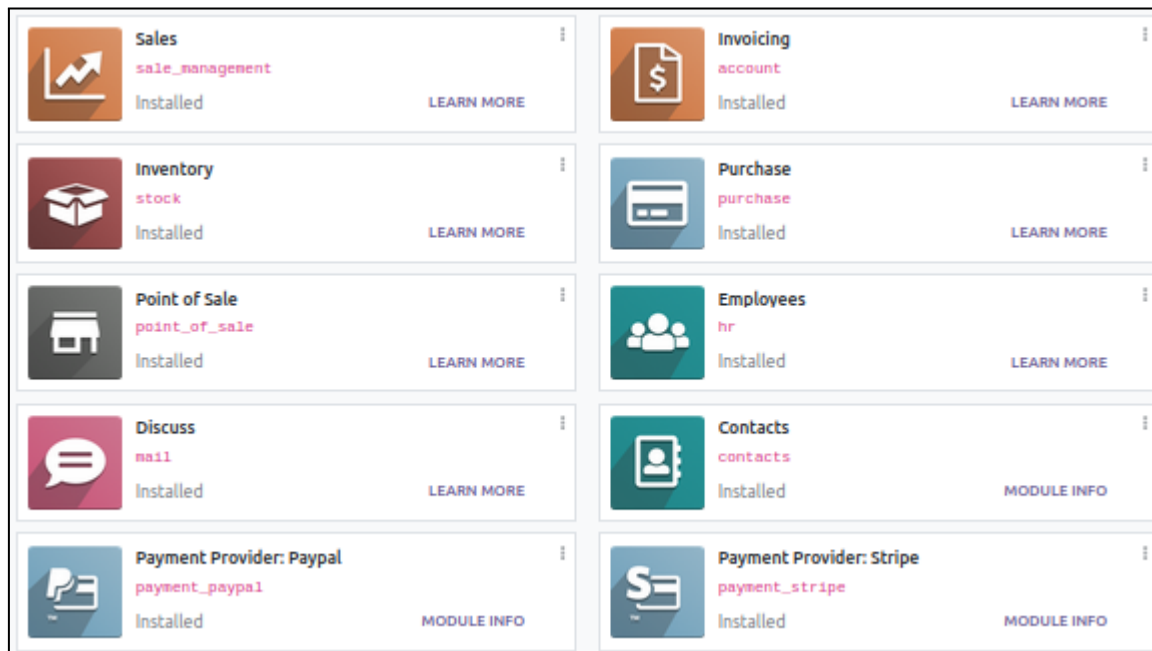


Con los anteriores cambios guardados, en la base de datos se puede comprobar que existe la tabla `res_partner` y la columna `debit_limit`.

Para añadir más funcionalidades a Odoo, desde el panel de aplicaciones se pueden instalar las que se consideren apropiadas. En este caso, se ha optado por Compras, Ventas e Inventario, tres módulos que añaden versatilidad por medio de un gran número de funcionalidades.



El módulo de compras, y tras seleccionar el método de pagos, gestiona la instalación de los módulos necesarios para dicha acción. En este caso, dos proveedores como PayPal y Stripe.



5.3. Consulta a la base de datos

Como hemos visto, Odoo emplea Postgres como SGBD. En nuestro caso hemos empleado pgAdmin para obtener información de nuestra base de datos. Podemos hacerlo de forma visual, es decir, empleando el entorno gráfico de pgAdmin (Tools > Query Tool)

Query	Query History
1	<code>SELECT * FROM public.res_partner</code>
2	<code>ORDER BY id ASC</code>

```
bd_empresa1=# select name from res_partner;
          name
-----
OdooBot
Portal User Template
Public user
IES de Teis
Ramón Estévez Casasnovas
Sabela Sabucedo Rego
Marisol Arias Rego
A Lira de Ribadavia
Festas Galicia
Queremos Charangueo
Default User Template
Administrator
(12 rows)
bd_empresa1=#
```

pgAdmin nos permite hacer la consulta como si nos encontrásemos en la terminal, pero gestionando directamente la conexión. (Tools > PSQL Tool)

Por último, tenemos la opción de hacerlo desde la Terminal de Ubuntu. Para ello, debemos conectarnos a postgres, emplear el comando

`psql` y conectarnos a la base de datos que deseemos (`bd_empresa1`). Una vez establecida la conexión y realizada la consulta, la respuesta nos aparece directamente en la terminal.

```
bd_empresa1=# select name from res_partner;
          name
-----
OdooBot
Portal User Template
Public user
IES de Teis
Ramón Estévez Casasnovas
Sabela Sabucedo Rego
Marisol Arias Rego
A Lira de Ribadavia
Festas Galicia
Queremos Charangueo
Default User Template
Administrator
(12 rows)
bd_empresa1=#
```

Analizando los datos de la tabla `res_partner`, vemos que la columna `company_id` aparece en muchos casos con el valor 1. Para observar los cambios, vamos a asociar a los empleados que teníamos a sus respectivas compañías: en el menú de contactos, debajo del nombre de cada empleado, asociaremos a la empresa correspondiente. Una vez efectuado el cambio en Odoo, al actualizar los datos podemos observar que la columna de `company_id` pasa a estar en null para dichos usuarios.



company_id integer	create_date timestamp without time zone	name character varying
[null]	2022-11-15 21:32:07.130073	Public user
[null]	2022-11-15 21:32:07.130073	Default User Template
[null]	2022-11-15 21:32:07.130073	Portal User Template
[null]	2022-11-16 20:33:17.883792	Ramón Estévez Casasr
[null]	2022-11-16 20:33:17.883792	Sabela Sabucedo Rego
[null]	2023-02-07 21:08:46.952676	Marisol Arias Rego
[null]	2023-02-07 21:14:18.368295	A Lira de Ribadavia
[null]	2023-02-07 21:53:47.94303	Queremos Charangueo
[null]	2023-02-07 21:55:22.570095	Festas Galicia

Con los datos ya guardados, es momento de proceder al ejercicio 1 de la tarea, que consiste en realizar una consulta a la base de datos donde se muestren: id de cada tabla en la que se basa la consulta, nombre de la empresa, tipo de empresa (SL, SA, etc.), idioma, crédito concedido, calle, código postal, ciudad y teléfono:

```
select
  'res_partner' as tabla1,
  'ir_property' as tabla2,
  res_partner.id,
  res_partner.name as nombre_empresa,
  res_partner.title as tipo_empresa,
  res_partner.lang as idioma,
  ir_property.value_float as credito_concedido,
  res_partner.debit_limit as limite_credito,
  res_partner.street as calle,
  res_partner.zip as cp,
  res_partner.city as ciudad,
  res_partner.phone as tlf
from res_partner
left join ir_property on ir_property.res_id = concat('res.partner,',res_partner.id)
where is_company = true ;
```

La consulta nos devuelve los siguientes datos:

res_partner	ir_property	1	IES de Teis	[null]	en_US	200	[null]	Avenida de Galicia, 101	36216	Vigo	+34886120486
res_partner	ir_property	12	Festas Galicia	[null]	en_US	[null]	[null]	[null]	[null]	[null]	+34981987654
res_partner	ir_property	10	A Lira de Ribadavia	[null]	en_US	[null]	[null]	Rúa de Redondela, s/n	32400	Ribadavia	+34988658976
res_partner	ir_property	11	Queremos Charangueo	[null]	en_US	[null]	[null]	[null]	[null]	[null]	986123456