



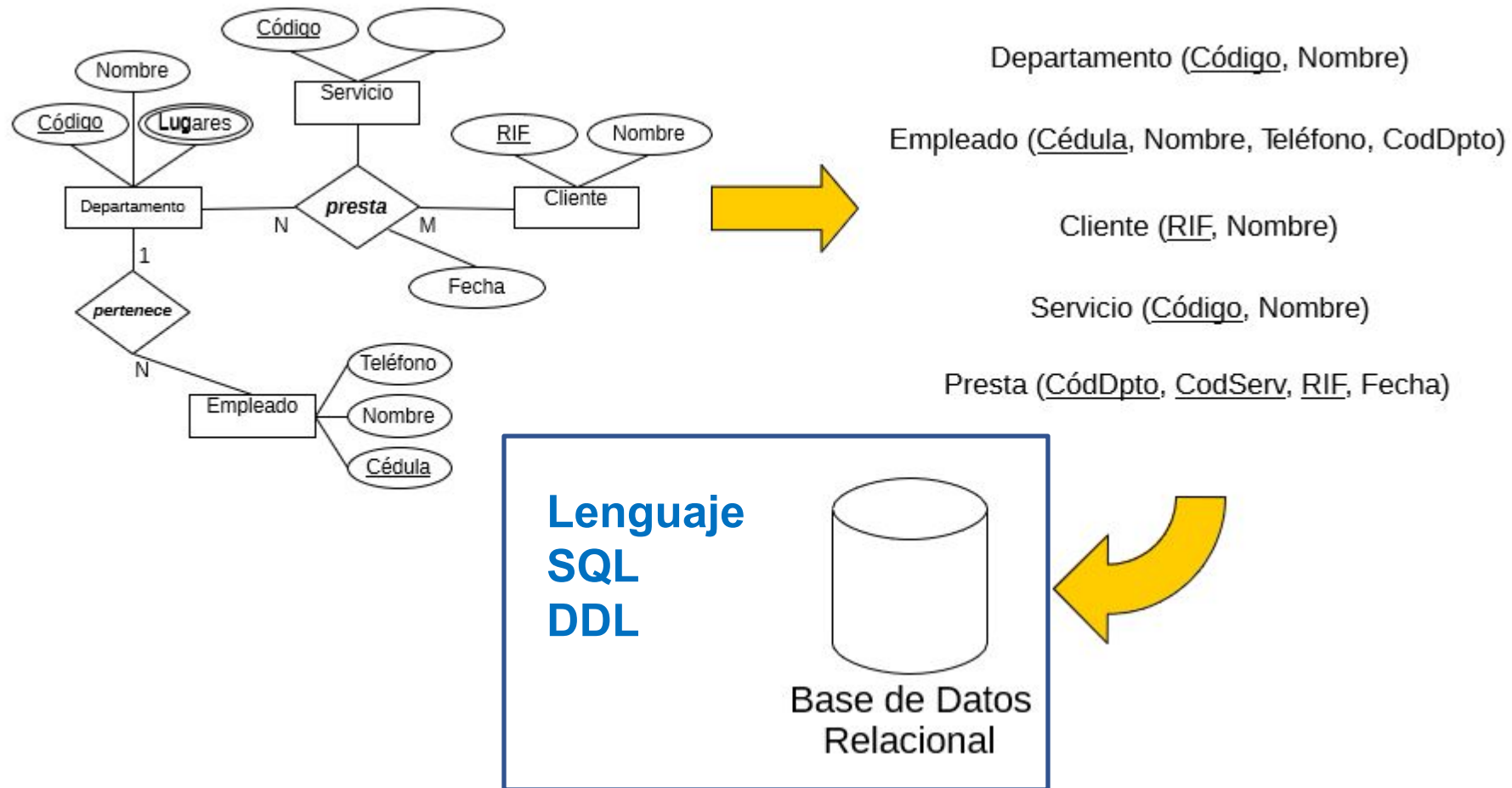
GeeksHubs
academy _

DDL / DML



DDL



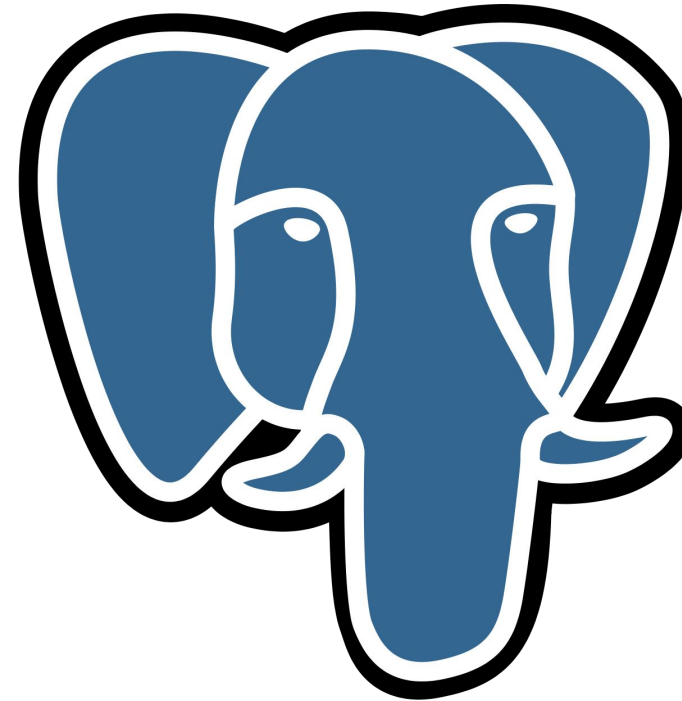


PostgreSQL

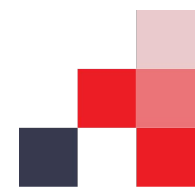
PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL.

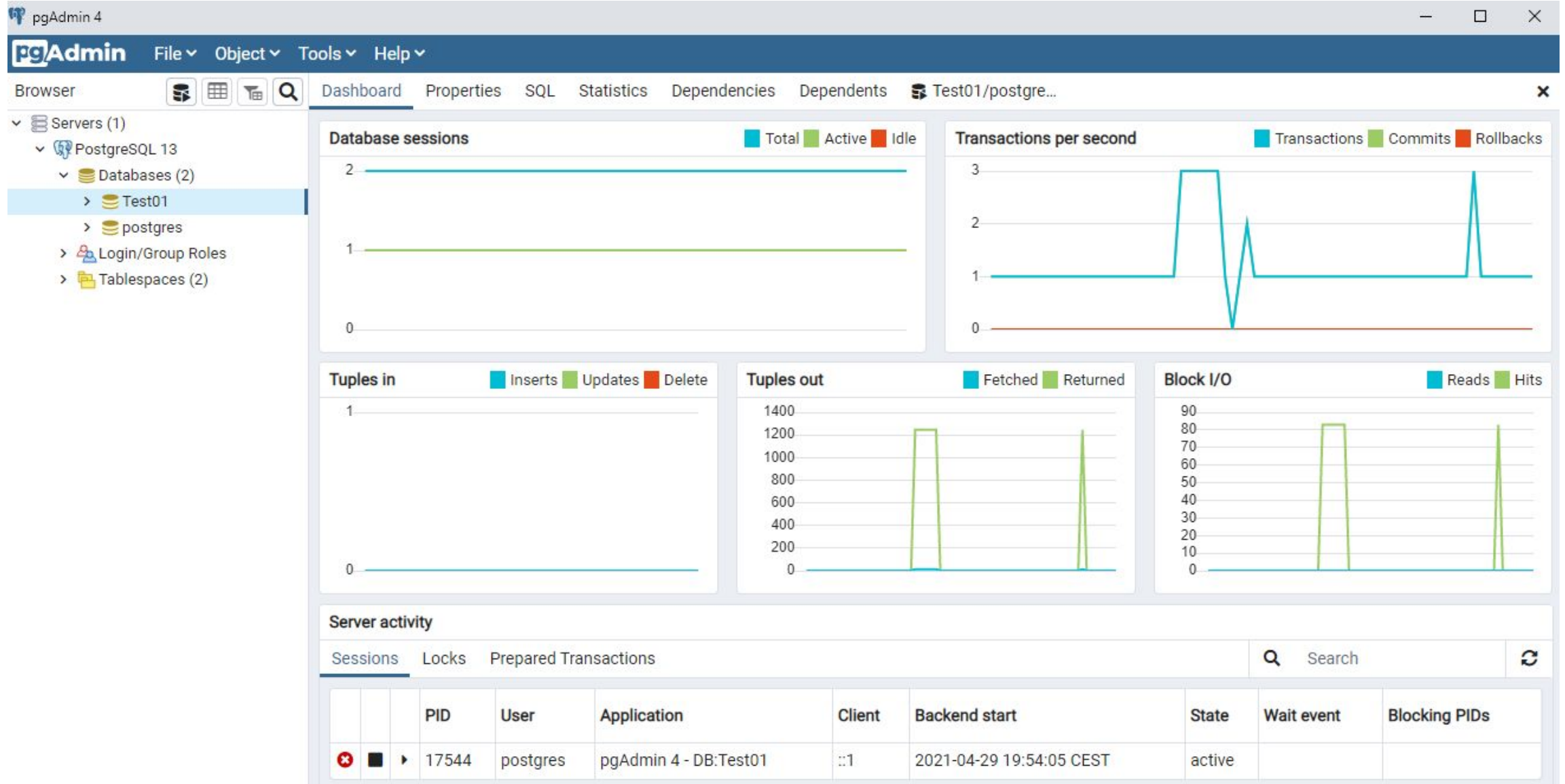
Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. La última versión es la 13.2

<https://www.postgresql.org/>



Stack Necesario





Usuarios

En **PostgreSQL** existen cinco niveles distintos de privilegios:

1. **Globales:** se aplican al conjunto de todas las bases de datos en un servidor. el nivel más alto de privilegio, en el sentido de que su ámbito es el más general.
2. **De base de datos:** se refieren a bases de datos individuales, y por extensión, a todos los objetos que contiene cada base de datos.
3. **De tabla:** se aplican a tablas individuales, y por lo tanto, a todas las columnas de esas tabla.
4. **De columna:** se aplican a una columna en una tabla concreta.
5. **De rutina:** se aplican a los procedimientos almacenados.



Sentencias SQL DDL

```
CREATE DATABASE databasename;  
  
DROP DATABASE databasename;  
  
BACKUP DATABASE databasename TO DISK = 'filepath';  
  
CREATE SEQUENCE table_name_id_seq;  
  
CREATE TABLE Persons (  
    PersonID int NOT NULL DEFAULT nextval('table_name_id_seq'),  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255),  
    PRIMARY KEY (PersonID)  
);  
  
DROP TABLE table_name;  
  
ALTER TABLE table_name ADD column_name datatype;
```



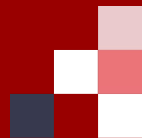
FOREIGN KEY

```
CREATE TABLE Persons (  
    ID int NOT NULL PRIMARY KEY,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);
```

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID)  
    REFERENCES Persons(PersonID)  
);
```



Kata III



KATA III: Modelo Relacional a DDL

En esta Kata vamos a definir en base de datos las siguientes entidades definidas en la base de datos. Consiste en modelar una biblioteca a partir del siguiente modelo relacional:

```
Libro(ClaveLibro, Título, Idioma, Formato, ClaveEditorial)
Tema(ClaveTema, Nombre)
Autor(ClaveAutor, Nombre)
Editorial(ClaveEditorial, Nombre, Dirección, Teléfono)
Ejemplar(ClaveEjemplar, ClaveLibro, NúmeroOrden, Edición, Ubicación, Categoría)
Socio(ClaveSocio, Nombre, Dirección, Teléfono, Categoría)
Préstamo(ClaveSocio, ClaveEjemplar, NúmeroOrden, Fecha_préstamo,
        Fecha_devolución, Notas)
Trata_sobre(ClaveLibro, ClaveTema)
Escrito_por(ClaveLibro, ClaveAutor)
```



DML

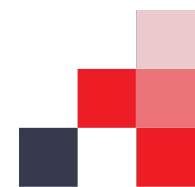


INSERT INTO

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway



DELETE FROM

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden



SELECT

```
SELECT column_name,column_name  
FROM table_name;
```

```
SELECT * FROM table_name;
```



```
SELECT CustomerName,City FROM Customers;
```

CustomerName	City
Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå
Blauer See Delikatessen	Mannheim
Blondel père et fils	Strasbourg
Bólido Comidas preparadas	Madrid



SELECT ALL

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden



Select Where

```
SELECT column_name,column_name
FROM table_name
WHERE column_name operator value;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Select all Where

```
SELECT * FROM Customers  
WHERE Country='Mexico';
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
13	Centro comercial Moctezuma	Francisco Chang	Sierras de Granada 9993	México D.F.	05022	Mexico
58	Pericles Comidas clásicas	Guillermo Fernández	Calle Dr. Jorge Cash 321	México D.F.	05033	Mexico

Select All Where id

```
SELECT * FROM Customers  
WHERE CustomerID=1;
```



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany



Operadores de Búsqueda

Operator	Description
=	Equal
<>	Not equal. Note: In some versions of SQL this operator may be written as !=
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column



AND

```
SELECT * FROM Customers  
WHERE Country='Germany'  
AND City='Berlin';
```



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany



OR

```
SELECT * FROM Customers  
WHERE City='Berlin'  
OR City='München';
```



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
25	Frankenversand	Peter Franken	Berliner Platz 43	München	80805	Germany

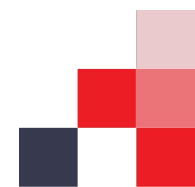


AND / OR

```
SELECT * FROM Customers  
WHERE Country='Germany'  
AND (City='Berlin' OR City='München');
```



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
25	Frankenversand	Peter Franken	Berliner Platz 43	München	80805	Germany



Sentencia Order By

```
SELECT * FROM Customers  
ORDER BY Country;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
64	Rancho grande	Sergio Gutiérrez	Av. del Libertador 900	Buenos Aires	1010	Argentina
54	Océano Atlántico Ltda.	Yvonne Moncada	Ing. Gustavo Moncada 8585 Piso 20-A	Buenos Aires	1010	Argentina
12	Cactus Comidas para llevar	Patricio Simpson	Cerrito 333	Buenos Aires	1010	Argentina
59	Piccolo und mehr	Georg Pipps	Geislweg 14	Salzburg	5020	Austria
20	Ernst Handel	Roland Mendel	Kirchgasse 6	Graz	8010	Austria
50	Maison Dewey	Catherine Dewey	Rue Joseph-Bens 532	Bruxelles	B-1180	Belgium
76	Suprêmes délices	Pascale Cartrain	Boulevard Tirou, 255	Charleroi	B-6000	Belgium

Sentencias Order By ASC DESC

Ordenar filas

- ASC orden ascendente, por default.
- DESC orden descendente.

```
SELECT column_name, column_name  
FROM table_name  
ORDER BY column_name ASC|DESC, column_name ASC|DESC;
```

```
SELECT * FROM Customers  
ORDER BY Country DESC;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
47	LINO-Delicateses	Felipe Izquierdo	Ave. 5 de Mayo Porlamar	I. de Margarita	4980	Venezuela
33	GROSELLA-Restaurante	Manuel Pereira	5ª Ave. Los Palos Grandes	Caracas	1081	Venezuela
35	HILARIÓN-Abastos	Carlos Hernández	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	5022	Venezuela
46	LILA-Supermercado	Carlos González	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	Barquisimeto	3508	Venezuela
32	Great Lakes Food	Howard Snyder	2732 Baker Blvd.	Eugene	97403	USA

Sentencias Order By DESC

```
SELECT * FROM Customers  
ORDER BY Country, CustomerName;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
12	Cactus Comidas para llevar	Patricio Simpson	Cerrito 333	Buenos Aires	1010	Argentina
54	Océano Atlántico Ltda.	Yvonne Moncada	Ing. Gustavo Moncada 8585 Piso 20-A	Buenos Aires	1010	Argentina
64	Rancho grande	Sergio Gutiérrez	Av. del Libertador 900	Buenos Aires	1010	Argentina
20	Ernst Handel	Roland Mendel	Kirchgasse 6	Graz	8010	Austria
59	Piccolo und mehr	Georg Pipps	Geislweg 14	Salzburg	5020	Austria
50	Maison Dewey	Catherine Dewey	Rue Joseph-Bens 532	Bruxelles	B-1180	Belgium
76	Suprêmes délices	Pascale Cartrain	Boulevard Tirou, 255	Charleroi	B-6000	Belgium



Sentencias Between

```
SELECT * FROM Products  
WHERE Price BETWEEN 10 AND 20;
```

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5
16	Pavlova	7	3	32 - 500 g boxes	17.45
21	Sir Rodney's Scones	8	3	24 pkgs. x 4 pieces	10
25	NuNuCa Nuß-Nougat-Creme	11	3	20 - 450 g glasses	14
31	Gorgonzola Telino	14	4	12 - 100 g pkgs	12.5

Uso de la Condición BETWEEN para mostrar las filas basadas en un rango de valores



Sentencias Between

```
SELECT * FROM Products  
WHERE (Price BETWEEN 10 AND 20)  
AND NOT CategoryID IN (1,2,3);
```

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
31	Gorgonzola Telino	14	4	12 - 100 g pkgs	12.5
36	Inlagd Sill	17	8	24 - 250 g jars	19
40	Boston Crab Meat	19	8	24 - 4 oz tins	18.4
42	Singaporean Hokkien Fried Mee	20	5	32 - 1 kg pkgs.	14
46	Spegesild	21	8	4 - 450 g glasses	12
57	Ravioli Angelo	26	5	24 - 250 g pkgs.	19.5
58	Escargots de Bourgogne	27	8	24 pieces	13.25
73	Röd Kaviar	17	8	24 - 150 g jars	15



Sentencias IN

```
SELECT * FROM Customers  
WHERE City IN ('Paris','London');
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
11	B's Beverages	Victoria Ashworth	Fauntleroy Circus	London	EC2 5NT	UK
16	Consolidated Holdings	Elizabeth Brown	Berkeley Gardens 12 Brewery	London	WX1 6LT	UK
19	Eastern Connection	Ann Devon	35 King George	London	WX3 6FW	UK
53	North/South	Simon Crowther	South House 300 Queensbridge	London	SW7 1RZ	UK
57	Paris spécialités	Marie Bertrand	265, boulevard Charonne	Paris	75012	France
72	Seven Seas Imports	Hari Kumar	90 Wadhurst Rd.	London	OX15 4NB	UK
74	Spécialités du monde	Dominique Perrier	25, rue Lauriston	Paris	75016	France

Uso de la condición IN para evaluar los valores en una lista



Sentencias Alias AS

```
SELECT CustomerName AS Customer, ContactName AS [Contact  
Person]  
FROM Customers;
```

Customer	Contact Person
Alfreds Futterkiste	Maria Anders
Ana Trujillo Emparedados y helados	Ana Trujillo
Antonio Moreno Taquería	Antonio Moreno
Around the Horn	Thomas Hardy
Berglunds snabbköp	Christina Berglund
Blauer See Delikatessen	Hanna Moos
Blondel père et fils	Frédérique Citeaux
Bólido Comidas preparadas	Martín Sommer



Sentencias de consulta múltiple

```
SELECT o.OrderID, o.OrderDate, c.CustomerName  
FROM Customers AS c, Orders AS o  
WHERE c.CustomerName="Around the Horn" AND  
c.CustomerID=o.CustomerID;
```

OrderID	OrderDate	CustomerName
10383	12/16/1996	Around the Horn
10355	11/15/1996	Around the Horn



SQL COUNT(), AVG() and SUM()

```
SELECT COUNT(ProductID)  
FROM Products;
```



COUNT(ProductID)
77

```
SELECT AVG(Price)  
FROM Products;
```



AVG(Price)
28.8663636363637

```
SELECT SUM(Quantity)  
FROM OrderDetails;
```



SUM(Quantity)
12743



Sentencia HAVING

```
SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders
FROM (Orders
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID)
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 10;
```

Number of Records: 8

LastName	NumberOfOrders
Buchanan	11
Callahan	27
Davolio	29
Fuller	20
King	14
Leverling	31
Peacock	40
Suyama	18



SUBCONSULTAS

```
SELECT nombre FROM empleados  
WHERE cuota <= (SELECT SUM(importe) FROM pedidos WHERE rep = numemp);
```

```
SELECT oficina, ciudad FROM oficinas  
WHERE objetivo > (SELECT SUM(ventas) FROM empleados WHERE oficina = oficinas.oficina);
```

```
SELECT nombre FROM empleados  
HAVING AVG(cuota) <= (SELECT SUM(importe) FROM pedidos WHERE rep = numemp);
```



Sentencias JOIN

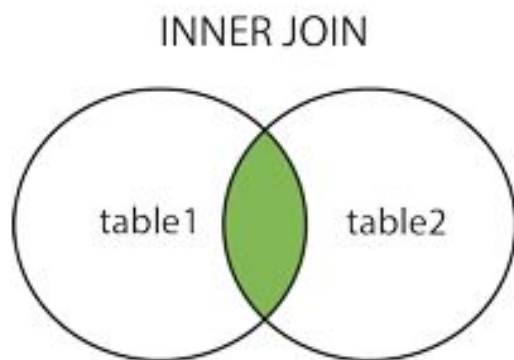
Una cláusula SQL JOIN se utiliza para combinar las filas de dos o más tablas, basado en un campo común entre ellos.

- **INNER JOIN.**
- **LEFT JOIN.**
- **RIGHT JOIN.**
- **FULL JOIN.**

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
INNER JOIN Customers  
ON Orders.CustomerID=Customers.CustomerID;
```

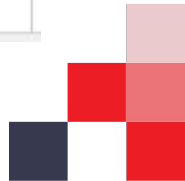


INNER JOIN



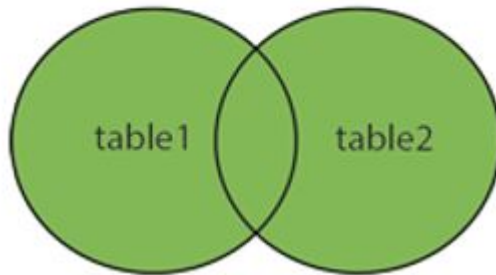
```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
INNER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

CustomerName	OrderID
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
Around the Horn	10383
Around the Horn	10355
Berglunds snabbköp	10278
Berglunds snabbköp	10280
Berglunds snabbköp	10384
Blondel père et fils	10265



FULL OUTER JOIN

FULL OUTER JOIN



```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
FULL OUTER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

CustomerName	OrderID
Alfreds Futterkiste	
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
	10382
	10351



Sentencia UNION

```
SELECT City FROM Customers  
UNION  
SELECT City FROM Suppliers  
ORDER BY City;
```

City
Aachen
Albuquerque
Anchorage
Ann Arbor
Annecy
Århus
Barcelona
Barquisimeto



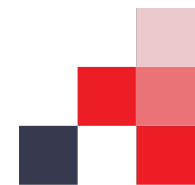
Sentencia GROUP BY

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country;
```

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;
```

Number of Records: 21

COUNT(CustomerID)	Country
3	Argentina
2	Austria
2	Belgium
9	Brazil
3	Canada
2	Denmark

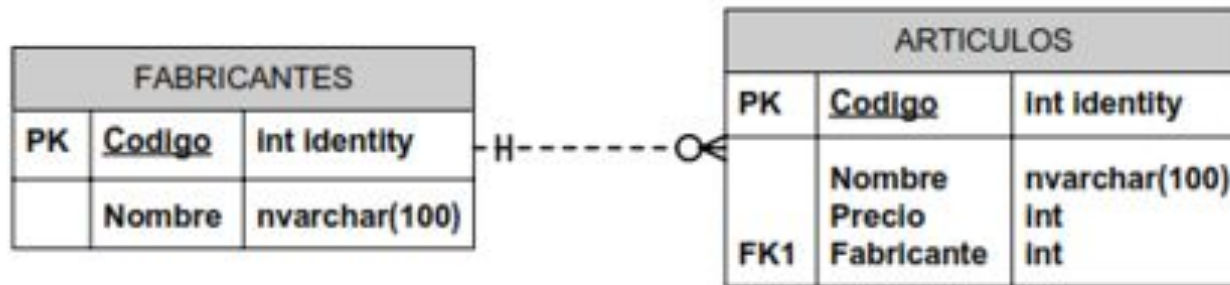


RETO 2

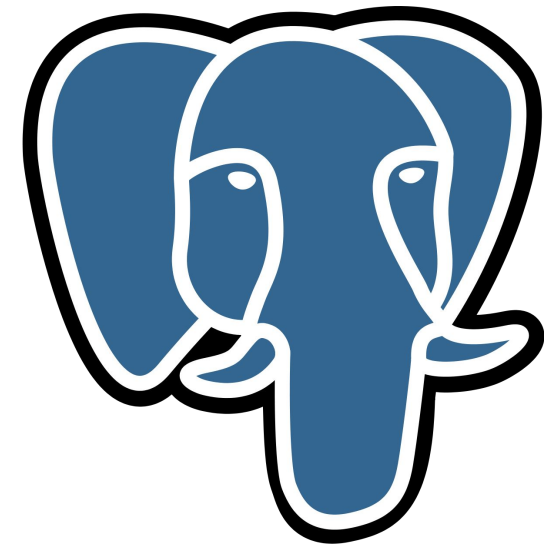


RETO 2:

Importa la base de datos desde el archivo “actividades_sql”. Una vez importado a tu sistema postgres implementa las sentencias SQL necesarias para resolver cada uno de las consultas planteadas en las siguientes actividades.



Añade diez registros por cada una de las entidades definidas en el siguiente diagrama ER.





1

- 1.1. Obtener los nombres de los productos de la tienda.
- 1.2. Obtener los nombres y los precios de los productos de la tienda.
- 1.3. Obtener el nombre de los productos cuyo precio sea menor o igual a 200 €.
- 1.4. Obtener todos los datos de los artículos cuyo precio esté entre los 60 € y los 120 € (ambas cantidades incluidas).
- 1.5. Obtener el nombre y el precio en pesetas (es decir, el precio en euros multiplicado por 166'386)





2

- 1.6. Seleccionar el precio medio de todos los productos.
- 1.7. Obtener el precio medio de los artículos cuyo código de fabricante sea 2.
- 1.8. Obtener el numero de artículos cuyo precio sea mayor o igual a 180 €.
- 1.9. Obtener el nombre y precio de los artículos cuyo precio sea mayor o igual a 180 € y ordenarlos descendientemente por precio, y luego ascendentemente por nombre.
- 1.10. Obtener un listado completo de artículos, incluyendo por cada articulo los datos del articulo y de su fabricante.
- 1.11. Obtener un listado de artículos, incluyendo el nombre del artículo, su precio, y el nombre de su fabricante.
- 1.12. Obtener el precio medio de los productos de cada fabricante, mostrando solo los códigos de fabricante.





3

- 1.13. Obtener el precio medio de los productos de cada fabricante, mostrando el nombre del fabricante.
- 1.14. Obtener los nombres de los fabricantes que ofrezcan productos cuyo precio medio sea mayor o igual a 150 €.
- 1.15. Obtener el nombre y precio del artículo más barato.
- 1.16. Obtener una lista con el nombre y precio de los artículos más caros de cada proveedor (incluyendo el nombre del proveedor).
- 1.17. Añadir un nuevo producto: Altavoces de 70 € (del fabricante 2)
- 1.18. Cambiar el nombre del producto 8 a 'Impresora Laser'
- 1.19. Aplicar un descuento del 10 % (multiplicar el precio por 0'9) a todos los productos.
- 1.20. Aplicar un descuento de 10 € a todos los productos cuyo precio sea mayor o igual a 120 €.



Jose Marín

www.geekshubsacademy.com

@geekshubs

