



## Databases: Proyecto

## PROYECTO MÓDULO DATABASES

*El proyecto final del módulo tiene como objetivo demostrar todos los conocimientos adquiridos durante las clases. Estará compuesto por tres partes:*

1. Cuestionario (10%)
2. Consultas sobre base de datos (30%)
3. Modelado de una base de datos (60%)

### 1. CUESTIONARIO

1. Explicar la diferencia entre Base de datos Relacional y SQL
2. ¿Por qué es necesario para la tablas definir una primary key?
3. ¿Cómo se denomina la relación que se hace entre una columna de un tabla y la primary key de otra tabla?
4. ¿Qué es lo que necesitamos hacer para poder tener una relación n:m entre dos tablas?

### 2. CONSULTAS SQL SOBRE UNA BASE DE DATOS

Vamos a partir de una base de datos ya definida y poblada con datos: [https://github.com/pthom/northwind\\_psql](https://github.com/pthom/northwind_psql)

En este repositorio podéis ver el modelo entidad-relación y en el fichero con extensión sql todas las consultas necesarias para que os podáis construir una copia de esta base de datos.

Lo primero es tener operativa esta base de datos, para ello podemos hacerlo de dos formas:

#### Método 1

1. Crear una base de datos nueva que se llame ProyectoFinal
2. Copiar todas las sentencias definidas en el fichero sql y ejecutarlas
3. Aseguraros que no os da ningún error y que se han creado las tablas con registros

#### Método 2

1. Crear una base de datos nueva que se llame ProyectoFinal
2. Descargamos el fichero northwind.sql en nuestro ordenador
3. Copiamos el fichero dentro del docker de postgres para luego poder usar el fichero sql dentro
4. Entramos dentro del docker de postgres y ejecutamos el fichero sql para que cargue todo en la base de datos

```
● iperez@MacBook-Air-de-Itziar Downloads % docker cp northwind.sql postgres_container:/.  
Successfully copied 353kB to postgres_container:/.  
○ iperez@MacBook-Air-de-Itziar Downloads % docker exec -ti postgres_container bash  
root@6ceacfed5e8:/# ls  
bin boot data dev docker-entrypoint-initdb.d etc home lib media mnt northwind.sql opt proc root run sbin srv s  
root@6ceacfed5e8:/# psql -U postgres -d project -a -f northwind.sql
```

Una vez tengáis la base de datos disponible, hay que realizar las siguientes consultas:

1. Buscar todos los clientes (*customers*) con el código postal 1010.
2. Buscar el número de teléfono que tiene el proveedor (*supplier*) con id 11.
3. Listar los primeros 10 pedidos (*orders*) ordenados de manera descendente por la fecha de pedido.
4. Buscar todos los clientes (*customers*) que vivan en *London*, *Madrid* o *Brazil*.
5. Añadir un nuevo registro en la tabla clientes (*customers*) con la siguiente información (indicada en el mismo orden de las columnas): "XYZ", "The Shire", "Bilbo Baggins", "1 Hobbit-Hole", "Bag End", "111" y "Middle Earth".
6. Actualizar el código postal a "11122" del cliente "Bilbo Baggins".
7. Mostrar "ProductName" y "CategoryName" de todos los productos (*products*).
8. Mostrar "OrderID" y "CompanyName" del expedidor (*shippers*) de todos los pedidos (*orders*) realizados antes del 9 de agosto de 2012.
9. Mostrar el número de pedidos (*orders*) realizados por cada expedidor (*shipper*).

**Entregar todas las consultas sql realizadas y los resultados de ellas para poder validar que están correctas.**

### **3. MODELADO BASE DE DATOS**

Siguiendo los pasos que hemos ido haciendo en los ejercicios de clase:

- Plantear un caso de uso (el que vosotros queráis o podemos realizarlo sobre proyectos entregados en otros módulos) en el que quede bien explicado los requisitos y el caso de uso que queremos implementar en nuestra base de datos. Mínimo debe tener 3 entidades con 3-4 atributos cada una de ellas; también deben existir relaciones 1:N y N:M.
- Crear el modelo entidad-relación correspondiente
- Crear la base de datos con sus tablas y relaciones entre ellas
- Poblar la base de datos con una serie de registros y hacer al menos 5 consultas para demostrar que vuestro modelo cumple con los requisitos indicados en el paso 1.

**Entregar todas las consultas sql realizadas y los resultados de ellas para poder validar que están correctas junto con el diagrama ER.**