

Entrega

- El nombre del fichero será `201.zip` y contendrá:
- Scripts SQL (`.sql`) que utilizaste para crear la BD MySQL y PostgreSQL y añadirle datos
- Los archivos `.java` con el código de la aplicación.

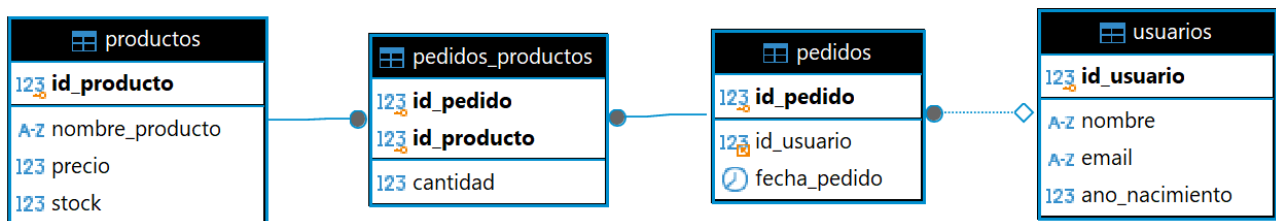
Se valorará

- El código no presenta problemas de ejecución.
- Las bases de datos son cargadas correctamente a partir de los scripts proporcionados.
- Los scripts de las bases de datos proporcionan datos de muestra para verificar la aplicación.
- Que el proyecto no sea entregado fuera de la fecha límite.

Descripción del proyecto

En este proyecto habrá que modelar dos bases de datos. Una base de datos MySQL y una base de datos PostgreSQL.

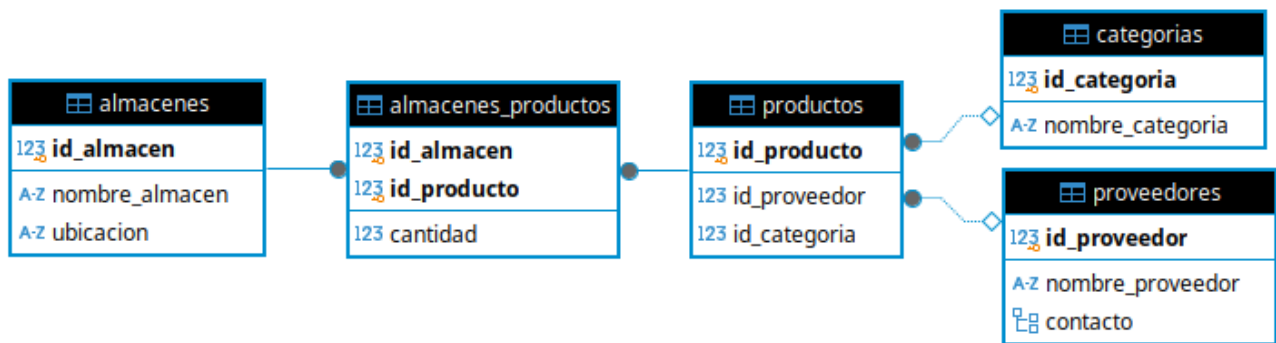
La base de datos MySQL que habrá que modelar tiene el siguiente diagrama relacional:



Donde existen las siguientes relaciones:

- Una relación varios a varios entre `Productos` y `Pedidos`.
- Una relación varios a uno entre `Pedidos` y `Usuarios`.

En cuanto a la base de datos PostgreSQL que habrá que modelar, esta tiene el siguiente diagrama relacional:



Donde existen las siguientes relaciones:

- Una relación varios a varios entre almacenes y productos.
- Una relación varios a uno entre productos y categorias.
- Una relación varios a uno entre productos y proveedores.

Además, en la tabla proveedores, el atributo contacto **representa un objeto** formado por los siguientes atributos: nombre_contacto, nif, telefono y email.

A TENER EN CUENTA

Se deberá tener en cuenta que las tablas productos de ambas bases de datos permiten establecer el enlace entre ambas bases de datos. La información del producto se guardará principalmente en la base de datos MySQL donde se almacenará junto toda la información relacionada con los pedidos y los usuarios que los hacen. En cuanto a la base de datos PostgreSQL en ella se almacenará toda la información del producto que tenga que ver con su almacenamiento, categoría y proveedor.

DATOS

Los script SQL de ambas bases de datos deberán añadir datos a modo de ejemplo que permitan probar las consultas.

Tareas a realizar

- Diseñar una base de datos en MySQL y PostgreSQL que cumplan con los diagramas mostrados anteriormente.
- Insertar en las Bases de datos, datos ficticios que permitan probar los apartados de la aplicación.
- Desarrollar la aplicación Java usando JDBC para conectarte a las bases de datos.

La aplicación debe permitir:

- Establecer las conexiones con la base de datos MySQL y PostgreSQL de la forma más eficiente que conozcas.
- Crear un menú que permita probar cada uno de los siguientes apartados.
- Apartados a implementar:
 - i. Crear una nueva categoría (PostgreSQL)
 - ii. Crear un nuevo proveedor (PostgreSQL)
 - iii. Eliminar un nuevo proveedor (PostgreSQL)
 - iv. Crear un nuevo usuario (MySQL)
 - v. Eliminar un usuario (MySQL)
 - vi. Crear nuevo producto (nombre, precio, stock, categoría, proveedor) (MySQL + PostgreSQL)
 - vii. Eliminar un producto por su nombre (MySQL + PostgreSQL)
 - viii. Listar los productos con bajo stock (menos de X unidades disponibles) (MySQL)
 - ix. Obtener el total de pedidos realizados por cada usuario (MySQL)
 - x. Obtener la cantidad de productos almacenados por cada almacén (PostgreSQL)
 - xi. Listar todos los productos con sus respectivas categorías y proveedores (PostgreSQL)
 - xii. Obtener todos los Usuarios que han comprado algún producto de una categoría dada (MySQL + PostgreSQL).

sobre las consultas

Los apartados del 8 al 10 consisten en la realización de algún tipo de consulta. Salvo que el apartado indique lo contrario el resultado de la consulta tendrá que obtenerse ya sea con lenguaje SQL o PostgreSQL. Java solo podrá utilizarse para realizar la iteración para mostrar los datos por pantalla. O para juntar las tablas producto de ambas bases de datos.

Descripción técnica

A continuación, se detallan los aspectos técnicos que habrán que tener en cuenta a la hora de implementar los apartados anteriores:

1.Crear una nueva categoría (PostgreSQL).

- Se implementará una función con la siguiente cabecera: `void`

```
crearCategoria(String nombreCategoria).
```

- Se recibirá un String que será el `nombreCategoria` y se añadirá a la base de datos.

2.Crear un nuevo proveedor (PostgreSQL)

- Se implementará una función con la siguiente cabecera: `void`

```
crearNuevoProveedor(String nombreProveedor, String nif, int telefono, String email).
```

- Se recibirá todos los datos del proveedor y se añadirán en la base de datos.

3.Eliminar un nuevo proveedor (PostgreSQL)

- Se implementará una función con la siguiente cabecera: `void`

```
eliminarProveedor(int id).
```

- Se tendrá que comprobar si el `id` indicado existe y si es así, eliminarlo de la base de datos.

4.Crear un nuevo usuario (MySQL)

- Se implementará una función con la siguiente cabecera: `void`

```
crearUsuario(String nombre, String email, int anho_nacimiento).
```

- Se recibirán todos los datos del usuario.

5.Eliminar un usuario (MySQL)

- Se implementará una función con la siguiente cabecera: `void`

```
eliminarUsuario(int id).
```

- Se tendrá que comprobar si el `id` indicado existe y si es así, eliminarlo de la base de datos.

6. Crear nuevo producto (nombre, precio, stock, categoría, proveedor) (MySQL + PostgreSQL)

- Se implementará una función con la siguiente cabecera: void

```
crearProducto(String nombre, Double precio, int stock, String  
nombre_categoria, String nif).
```

- Se tendrá que obtener el id de la categoría y el id del proveedor a partir del nombre y del nif.
- Se añadirá en la base de datos MySQL y en la base de datos PostgreSQL.
- El identificador del producto tendrá que ser el mismo en ambas bases de datos.

7. Eliminar un producto por su nombre (MySQL + PostgreSQL)

- Se implementará una función con la siguiente cabecera: void

```
eliminarProductoPorNombre(String nombre).
```

- Se tendrá que eliminar el producto de ambas bases de datos.

8. Listar los productos con bajo stock (menos de X unidades disponibles) (MySQL)

- Se implementará una función con la siguiente cabecera: void

```
listarProductosBajoStock(int stock) .
```

- Mediante una única consulta se tendrá que obtener el conjunto de filas resultante y mostrar el nombre de los productos junto con su stock.

9. Obtener el total de pedidos realizados por cada usuario (MySQL)

- Se implementará una función con la siguiente cabecera: void

```
obtenerTotalPedidosUsuarios().
```

- Mediante una consulta se tendrá que obtener toda la información e imprimir por pantalla: el nombre del usuario y el total de pedidos que ha hecho.

10. Obtener la cantidad de productos almacenados por cada almacén (PostgreSQL)

- Se implementará una función con la siguiente cabecera: void

```
obtenerCantidadProductosEnCadaAlmacen().
```

- Mediante una consulta se tendrá que obtener toda la información solicitada y pintar por pantalla el nombre del almacén y el total de productos de los que dispone.

11. Listar todos los productos con sus respectivas categorías y proveedores (PostgreSQL)

- Se implementará una función con la siguiente cabecera: void

```
listarTodosProductosConCategoriaYProveedor().
```

- Se realizará una primera consulta en PostgreSQL que permita obtener toda la información del producto: id, nombre, nif, teléfono e email del proveedor así como el nombre de la categoría.
- Se realizará una consulta en MySQL para obtener el nombre, precio y stock del producto
- Se concatenará y se mostrarán todos los datos indicados anteriormente usando Java.

12. Obtener todos los usuarios que han comprado algún producto de una categoría dada (MySQL + PostgreSQL).

• Se implementará una función con la siguiente cabecera: `void`

`obtenerUsuariosCompraronProductosCategoria(int idCategoria).`

• Se recibirá el id de la categoría y se obtendrá en PostgreSQL el id de los productos que pertenezcan a esa categoría.

• En MySQL se obtendrá el nombre de los usuarios que han comprado algún producto de los indicados anteriormente.

• Se mostrará por pantalla el nombre de los usuarios

sobre las especificaciones

Si no se respetan las especificaciones de un apartado:

- Definición de la cabecera exacta,
- Número de consultas,
- Uso excesivo de Java para obtener los datos,
- etc.

El apartado se evaluará con un 0 aunque el funcionamiento y el resultado sea el correcto.

Puntuación

Actividad	Puntuación
Scripts de PostgreSQL y MySQL más datos para probar el programa	0.1 puntos
Conexiones eficientes con las bases de datos	0.1 puntos
Crear nueva categoría	0.1 puntos
Crear nuevo proveedor	0.1 puntos
Eliminar un proveedor	0.1 puntos
Crear nuevo usuario	0.1 puntos
Eliminar un usuario	0.1 puntos
Crear nuevo producto	0.25 puntos
Eliminar un producto por nombre	0.25 puntos
Listar los productos con bajo stock	0.1 puntos
Obtener el total de pedidos realizados por cada usuario	0.1 puntos
Obtener la cantidad de productos almacenado por cada almacén	0.2 puntos
Listar todos los productos con categoría y proveedor	0.2 puntos
Obtener usuarios que han comprado algún producto de una categoría	0.2 puntos
TOTAL	2 puntos

SOBRE LAS ESPECIFICACIONES

Si no se respetan las especificaciones de un apartado:

- Definición de la cabecera exacta,
- Número de consultas,
- Uso excesivo de Java para obtener los datos,
- etc.

El apartado se evaluará con un 0 aunque el funcionamiento y el resultado sea el correcto.