

---

# Práctica: Base de Datos Distribuida de una Compañía Vinícola.

---



**UNIVERSIDAD  
DE GRANADA**

Bases de Datos Distribuidas (2020-2021)

Daniel Bolaños Martínez  
Julio José Prado Muñoz

# Índice

<b>1. Descripción del Problema</b>	<b>3</b>
<b>2. Diseño Conceptual y Lógico Global de la Base de Datos</b>	<b>5</b>
2.1. Diagrama Entidad-Relación . . . . .	5
2.2. Diseño lógico . . . . .	7
2.2.1. Descripción de las tablas y atributos . . . . .	7
2.2.2. Paso a tablas . . . . .	11
<b>3. Diseño de la Fragmentación y de la Asignación</b>	<b>12</b>
3.1. Fragmentación de tablas . . . . .	12
3.2. Replica de fragmentos . . . . .	15
3.3. Asignación de fragmentos, réplicas y relaciones . . . . .	15
3.4. Reconstrucción de las relaciones globales a partir de los fragmentos . .	16
<b>Referencias.</b>	<b>17</b>

# 1. Descripción del Problema

Una compañía vinícola española desea implementar una base de datos distribuida para gestionar, lo más eficientemente posible, la distribución de vinos españoles por todo el territorio nacional. La compañía dispone de cuatro delegaciones territoriales: **Madrid** (localidad 1), **Barcelona** (localidad 2), **La Coruña** (localidad 3) y **Sevilla** (localidad 4). La delegación de Madrid suministra vinos a clientes de la región centro de la península (Castilla-León, Castilla-La Mancha, Aragón, Madrid, La Rioja). La delegación de Barcelona suministra vinos a clientes del Levante Español (Cataluña, Baleares, País Valenciano y Murcia). La delegación de La Coruña suministra vinos a clientes del Norte de España (Galicia, Asturias, Cantabria, País Vasco y Navarra). Y, por último, la delegación de Sevilla suministra vinos a clientes de la zona Sur del país (Andalucía, Extremadura, Canarias, Ceuta y Melilla). Cada delegación realiza sus labores de gestión a través de sucursales ubicadas en diferentes ciudades de cada autonomía. Cada sucursal solamente puede distribuir directamente vinos producidos en las comunidades autónomas que pertenecen a su delegación. Un cliente puede solicitar suministros de cualquier vino, pero siempre tendrá que hacerlo a través de sucursales de la delegación a la que pertenece su comunidad autónoma. Si una sucursal no puede suministrar directamente un vino a un cliente porque su delegación no lo distribuye, esta sucursal solicitará el pedido de vino a cualquier sucursal de la delegación que sí lo distribuya. Por ejemplo, un cliente de Andalucía requiere suministros de vinos de La Rioja. Este cliente tendrá que solicitar su pedido a cualquier sucursal de la delegación de Sevilla. Puesto que esta delegación no distribuye vinos de la Rioja, la sucursal a la que se haya dirigido el cliente tendrá que solicitar el pedido a una sucursal de la delegación de Madrid que es la encargada de la distribución de los vinos de la Rioja.

Los principales requisitos de funcionamiento y de almacenamiento de la compañía se describen con la siguiente lista de especificaciones:

**LOS CLIENTES** de la compañía se identifican mediante un código de cliente y, para cada uno de ellos, se almacena el DNI (o CIF), nombre del cliente, dirección o domicilio social del cliente, tipo de cliente, comunidad autónoma en la que reside, los distintos suministros de vino que la distribuidora le va a proporcionar (o le ha proporcionado) y las sucursales que van a tramitar (o han tramitado) cada uno de esos suministros. Cada suministro consiste en un tipo de vino, cantidad que se va a suministrar (número de botellas) y la fecha en la que se solicitó el suministro. A efectos de almacenamiento, la solicitud de dos (o más) suministros del mismo vino en la misma fecha y requeridos a la misma sucursal por el mismo cliente, se consideran como un único suministro en el cual se suman las cantidades que se van a suministrar (o se han suministrado).

**LOS EMPLEADOS** de la compañía se identifican mediante un código de empleado, código que mantendrán mientras trabajen en dicha compañía independientemente de la sucursal en la que estén destinados en un momento determinado. La administración de la distribuidora almacena para cada empleado DNI, nombre del empleado, fecha de comienzo de contrato con la compañía (NO la fecha de comienzo de trabajo en una sucursal), salario, dirección y sucursal de destino. Cada empleado, en un momento dado, sólo puede estar asignado en una sucursal (no se mantendrá registro de las distintas sucursales a las que haya estado destinado el empleado).

**LOS VINOS** se identifican mediante un código de vino y, para cada uno de ellos, se almacena la marca del vino, año de la cosecha, denominación de origen (si la tiene), graduación, viñedo de procedencia, comunidad autónoma, cantidad producida (número de botellas), cantidad en stock (número de botellas) y productor. En cada momento, la cantidad en stock será la cantidad producida menos la cantidad total suministrada. Inicialmente, la cantidad en stock es la cantidad producida.

**LOS PRODUCTORES** se identifican mediante un código de productor y, para cada uno de ellos, se almacena el DNI (o CIF), nombre del productor, dirección o domicilio social del productor y los vinos que produce.

**LAS SUCURSALES** se identifican mediante un código de sucursal y, para cada una de ellas, se almacena el nombre de la sucursal, ciudad en la que se encuentra, comunidad autónoma en la que se ubica, director de la sucursal y los pedidos que ha realizado a otras sucursales para tramitar los suministros solicitados por sus clientes. Cada pedido consiste, en un tipo de vino, cantidad que se solicita y fecha del pedido. A efectos de almacenamiento, dos (o más) pedidos del mismo vino en la misma fecha y solicitados por la misma sucursal, se consideran como un único pedido en el cual se suman las cantidades solicitadas. Las aplicaciones, tanto de actualización como de consulta, que usan datos de sucursales (incluidos datos de empleados, clientes, suministros, vinos que distribuye, etc.), se pueden generar en cualquier localidad, pero referencian con una probabilidad del 95 % a sucursales cercanas.

## 2. Diseño Conceptual y Lógico Global de la Base de Datos

### 2.1. Diagrama Entidad-Relación

A continuación se presenta el diagrama **Entidad-Relación** elegido tras el estudio y análisis del problema, junto con posibles representaciones alternativas que podrían a su vez ser consideradas.

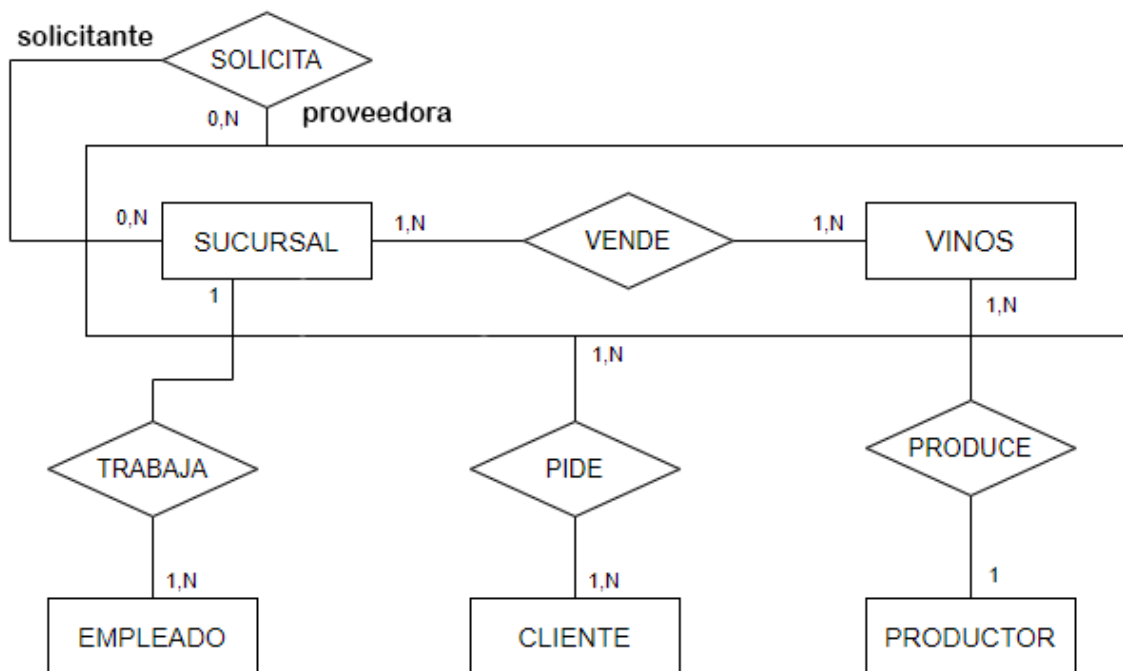


Figura 1: Diagrama E/R para nuestro problema.

#### Entidades:

- Sucursal
- Vino
- Productor
- Empleado
- Cliente

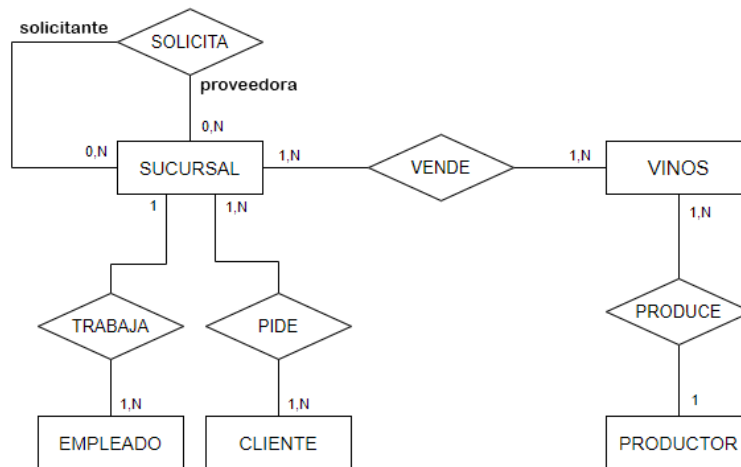
### Relaciones:

Relación	Tipo	Cardinalidad	Definición
Vino - Productor	Binaria	N:1	Un productor produce uno más vinos, siendo un vino producido por un único productor.
Empleado - Sucursal	Binaria	N:1	Un empleado trabaja para una sucursal en un momento determinado, teniendo cada sucursal uno o más de empleados
Sucursal - Vino	Binaria	N:N	Una sucursal distribuye uno o más tipos de vinos, pudiendo un vino ser distribuido por una o más sucursales
Sucursal - Sucursal	Unaria	N:N	Una sucursal solicita vinos a otra sucursal que distribuye los vinos solicitados.
Cliente - Sucursal	Binaria	N:N	Un cliente solicita una cantidad de uno o más tipos de vinos a una sucursal

Tabla 1: Relaciones diagrama E/R

### Alternativas de diseño:

- Eliminar la agregación entre Sucursal y Vinos.



**Problema:** eliminar esta agregación no nos impediría relizar nuestro diseño. Sin embargo, al proceder de manera natural, la relación pedido entre Cliente y Sucursal acarrea implícitamente la solicitud de uno o más tipos de vinos a una o varias sucursales, que tarde o temprano terminaría siendo plasmada en el diseño físico. La agregación pues, es una forma de estructurar y clarificar el comportamiento de esta relación.

## 2.2. Diseño lógico

Una vez explicado el diagrama E/R diseñaremos el modelo lógico de la base de datos, realizando el paso a tablas del modelo de datos relacional.

### 2.2.1. Descripción de las tablas y atributos

#### Cliente

- Identificador de cliente: Dato de tipo numérico que sirve como identificador para cada cliente dentro del sistema.
- DNI: Dato alfanumérico formado por 8 números y 1 caracter alfabético.
- Nombre: Dato de tipo String o cadena de texto.
- Dirección o domicilio social: Dato de tipo String.
- Tipo: Dato de tipo String que hace referencia al tipo de cliente en cuestión.
- Comunidad Autónoma: Dato de tipo String.

#### Empleado

- Identificador de empleado: Dato de tipo numérico que sirve como identificador para cada empleado dentro del sistema.
- DNI: Dato alfanumérico formado por 8 números y 1 caracter alfabético.
- Nombre: Dato de tipo String.
- Fecha de comienzo de contrato: Dato de tipo Date. Hace referencia al inicio del contrato con la compañía que es diferente a la de inicio de contrato con la sucursal.
- Salario: Dato de tipo numérico en coma flotante.
- Dirección: Dato de tipo String.
- Sucursal de destino: Dato de tipo numérico que hace referencia al identificador de la sucursal en la que trabaja el empleado en cuestión.

## **Vino**

- Identificador de vino: Dato de tipo numérico que sirve como identificador para cada vino dentro del sistema.
- Marca: Dato de tipo String.
- Año de la cosecha: Dato de tipo numérico formado por 4 dígitos.
- Denominación de origen: Dato de tipo String.
- Graduación: Dato de tipo numérico en coma flotante.
- Viñedo de procedencia: Dato de tipo String.
- Comunidad Autónoma: Dato de tipo String.
- Cantidad producida: Dato de tipo numérico que hace referencia al total de botellas de vino producidas.
- Cantidad en stock: Dato de tipo numérico que hace referencia al número de botellas de vino que quedan en stock en el almacén en cuestión.
- Productor: Dato de tipo numérico que hace referencia al identificador del productor del que procede el vino en cuestión.

## **Productor**

- Identificador de productor: Dato de tipo numérico que sirve como identificador para cada productor dentro del sistema.
- DNI o CIF: Dato alfanumérico formado por 8 números y 1 carácter alfabético.
- Nombre: Dato de tipo String.
- Dirección o domicilio social: Dato de tipo String.



### Sucursal

- Identificador de sucursal: Dato de tipo numérico que sirve como identificador para cada sucursal dentro del sistema.
- Nombre: Dato de tipo String.
- Ciudad: Dato de tipo String.
- Comunidad Autónoma: Dato de tipo String.
- Director: Dato de tipo numérico que hace referencia al identificador del empleado que es director de la sucursal en cuestión.

### Suministro Pedido Cliente

Representación en la base de datos de la relación **PIDE** entre las entidades Cliente y Sucursal en el diagrama E/R.

- Identificador pedido: Dato de tipo numérico que sirve como identificador para cada pedido de un cliente dentro del sistema.
- Identificador de cliente: Dato de tipo numérico que hace referencia al identificador del cliente que solicita el pedido en cuestión.
- Identificador de sucursal: Dato de tipo numérico que hace referencia al identificador de la sucursal a la que se le solicita el pedido en cuestión.
- Identificador de vino: Dato de tipo numérico que hace referencia al identificador del vino del suministro en cuestión.
- Cantidad: Dato de tipo numérico entero que hace referencia al número de botellas de vino del suministro.
- Fecha de solicitud: Dato de tipo Date. Hace referencia a la fecha de solicitud del pedido.

### Almacén

Representación en la base de datos de la relación **VENDE** entre las entidades Sucursal y Vinos en el diagrama E/R. Para identificar cada Almacén usaremos una clave compuesta con los identificadores de la Sucursal y el Vino.

- Identificador de sucursal: Dato de tipo numérico que hace referencia al identificador de la sucursal que contiene el almacén en cuestión.
- Identificador de vino: Dato de tipo numérico que hace referencia al identificador del vino del almacén en cuestión.

### Suministro Pedido Sucursal

Representación en la base de datos de la relación **SOLICITA** entre las entidades Sucursal (solicita) y Sucursal (suministra) en el diagrama E/R.

- Identificador de pedido: Dato de tipo numérico que sirve como identificador para cada pedido de una sucursal dentro del sistema.
- Identificador de sucursal solicitante: Dato de tipo numérico que hace referencia al identificador de la sucursal que solicita el pedido en cuestión.
- Identificador de sucursal suministradora: Dato de tipo numérico que hace referencia al identificador de la sucursal a la que se le solicita el pedido en cuestión.
- Identificador de vino: Dato de tipo numérico que hace referencia al identificador del vino del suministro en cuestión.
- Cantidad: Dato de tipo numérico entero que hace referencia al número de botellas de vino del suministro.
- Fecha de solicitud: Dato de tipo Date. Hace referencia a la fecha de solicitud del pedido.

### 2.2.2. Paso a tablas

- *Cliente* (id\_cliente, dni, nombre, direccion, tipo, ccaa)
- *Empleado* (id\_empleado, dni, nombre, direccion, fecha\_inicio, salario, id\_sucursal\*)  
\*\*Uso de propagación: inclusión del campo id\_sucursal en la tabla Empleado (un empleado trabaja en una única sucursal)
- *Vino* (id\_vino, marca, anio\_cosecha, denominacion, graduacion, procedencia, ccaa, cantidad\_producida, cantidad\_stock, id\_productor\*)  
\*\*Uso de propagación: inclusión del campo id\_productor en la tabla Vino (un vino es producido por un único productor)
- *Productor* (id\_productor, cif, nombre, direccion)
- *Sucursal* (id\_sucursal, nombre, ciudad, ccaa, director)  
\*\*Indicación de diseño: en lugar de crear una llave externa con el campo *director*, usaremos un disparador para comprobar que el identificador de director introducido se corresponde con un *id* de empleado en la tabla *Empleado*
- *SuministroPedidoCliente* (id\_pedido, id\_cliente\*, id\_sucursal\*, id\_vino, cantidad, fecha\_solicitud)  
\*\*Indicación de diseño: la referencia externa para *id\_vino* la implementaremos con un disparador que compruebe si hace referencia a un *Vino* de la tabla *Vino*.
- *Almacen* (id\_sucursal, id\_vino)  
\*\*Indicación de diseño: como las relaciones *Almacen* se crean en el momento de llamar al procedimiento para dar de alta un *Vino*, se asegura en ese momento que los identificadores de sucursal y vino son válidos.
- *SuministroPedidoSucursal* (id\_pedido, id\_sucursal\_solicitante, id\_sucursal\_suministradora, id\_vino, cantidad, fecha\_solicitud)  
\*\*Indicación de diseño: puesto que *SuministroPedidoSucursal* no será fragmentado, las referencias externas las implementaremos con un disparador que compruebe si *id\_sucursal\_solicitante*, *id\_sucursal\_suministradora* e *id\_vino* hacen referencia a *Sucursales* y *Vinos* existentes en la base de datos respectivamente.

### 3. Diseño de la Fragmentación y de la Asignación

#### 3.1. Fragmentación de tablas

Inicialmente, siguiendo la lógica de nuestra base de datos, fragmentaremos la relación *Sucursal* de forma **horizontal primaria** a partir del atributo **Comunidad Autónoma** (ccaa). Una opción alternativa consistiría en fragmentar la relación a partir de un nuevo atributo "localidad", sin embargo seguiremos la aproximación inicial a efectos prácticos. Para ello, obtenemos un conjunto minimal y completo de predicados simples:

$$P = \{ccaa = "Castilla-León", ccaa = "Castilla-La Mancha", ccaa = "Aragón", \\ ccaa = "La Rioja", ccaa = "Cataluña", ccaa = "País Valenciano", \\ ccaa = "Baleares", ccaa = "Murcia", ccaa = "Galicia", ccaa = "Asturias", \\ ccaa = "Cantabria", ccaa = "País Vasco", ccaa = "Navarra", ccaa = "Madrid", \\ ccaa = "Andalucía", ccaa = "Extremadura", ccaa = "Canarias", ccaa = "Ceuta", \\ ccaa = "Melilla"\}$$

Para simplificar, usaremos la siguiente notación:

$$\begin{aligned} p_1 &\equiv ccaa = "Madrid" \\ p_2 &\equiv ccaa = "Aragón" \\ p_3 &\equiv ccaa = "Castilla-León" \\ p_4 &\equiv ccaa = "Castilla-La Mancha" \\ p_5 &\equiv ccaa = "La Rioja" \\ p_6 &\equiv ccaa = "Cataluña" \\ p_7 &\equiv ccaa = "País Valenciano" \\ p_8 &\equiv ccaa = "Murcia" \\ p_9 &\equiv ccaa = "Baleares" \\ p_{10} &\equiv ccaa = "Galicia" \\ p_{11} &\equiv ccaa = "Asturias" \\ p_{12} &\equiv ccaa = "Cantabria" \\ p_{13} &\equiv ccaa = "Navarra" \\ p_{14} &\equiv ccaa = "País Vasco" \\ p_{15} &\equiv ccaa = "Andalucía" \\ p_{16} &\equiv ccaa = "Extremadura" \\ p_{17} &\equiv ccaa = "Canarias" \\ p_{18} &\equiv ccaa = "Ceuta" \\ p_{19} &\equiv ccaa = "Melilla" \end{aligned}$$

El número de expresiones conjuntivas resultante es de  $2^{19} = 524288$ . Dado el alto número de expresiones conjuntivas procederemos a indicar únicamente las verdaderas:

$$\begin{aligned}
y_1 &= p_1 \wedge \neg p_2 \wedge \dots \wedge \neg p_{19} \implies \textit{Verdadero} \\
y_2 &= \neg p_1 \wedge p_2 \wedge \neg p_3 \wedge \dots \wedge \neg p_{19} \implies \textit{Verdadero} \\
y_3 &= \neg p_1 \wedge \neg p_2 \wedge p_3 \wedge \neg p_4 \wedge \dots \wedge \neg p_{19} \implies \textit{Verdadero} \\
&\dots \\
y_{19} &= \neg p_1 \wedge \neg p_2 \wedge \dots \wedge \neg p_{18} \wedge p_{19} \implies \textit{Verdadero}
\end{aligned}$$

Cualquier expresión que involucre la pertenencia simultánea de una sucursal a dos o más comunidades autónomas distintas se evaluará como falsa. A su vez, se da por hecho que una sucursal debe pertenecer a una comunidad autónoma. El resto de expresiones se evalúan como verdaderas, es decir, aquellas en las que todas las expresiones aparecen negadas excepto una. En consecuencia, los términos de predicado serán  $y_1, y_2, y_3, \dots, y_{19}$ .

Teniendo en cuenta el proceso anterior, los esquemas de fragmentación para la relación *Sucursal*, *Cliente* y *Vino* son las siguiente:

$$\begin{aligned}
Sucursal_1 &= SL_{y_1}(Sucursal) \\
Sucursal_2 &= SL_{y_2}(Sucursal) \\
Sucursal_3 &= SL_{y_3}(Sucursal) \\
&\dots \\
Sucursal_{19} &= SL_{y_{19}}(Sucursal)
\end{aligned}$$

$$\begin{aligned}
Cliente_1 &= SL_{y_1}(Cliente) \\
Cliente_2 &= SL_{y_2}(Cliente) \\
Cliente_3 &= SL_{y_3}(Cliente) \\
&\dots \\
Cliente_{19} &= SL_{y_{19}}(Cliente)
\end{aligned}$$

$$\begin{aligned}
Vino_1 &= SL_{y_1}(Vino) \\
Vino_2 &= SL_{y_2}(Vino) \\
Vino_3 &= SL_{y_3}(Vino) \\
&\dots \\
Vino_{19} &= SL_{y_{19}}(Vino)
\end{aligned}$$

Puesto que las aplicaciones que requieren información sobre los empleados, pedidos y almacenes serán aplicaciones que se ejecutarán con frecuencia, habrá que facilitar las operaciones de producto natural entre estas relaciones.

De esta forma, las relaciones *Empleado*, *SuministroPedidoCliente* y *Almacén* serán fragmentadas de manera **horizontal derivada** a partir de la fragmentación horizontal primaria de *Sucursal*.

- Empleado y Sucursal

$$\begin{aligned} Empleado_1 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_1) \\ Empleado_2 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_2) \\ Empleado_3 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_3) \\ &\dots \\ Empleado_{19} &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_{19}) \end{aligned}$$

- SuministroPedidoCliente y Sucursal

$$\begin{aligned} SuministroPedidoCliente_1 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_1) \\ SuministroPedidoCliente_2 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_2) \\ SuministroPedidoCliente_3 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_3) \\ &\dots \\ SuministroPedidoCliente_{19} &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_{19}) \end{aligned}$$

- Almacén y Sucursal

$$\begin{aligned} Almacen_1 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_1) \\ Almacen_2 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_2) \\ Almacen_3 &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_3) \\ &\dots \\ Almacen_{19} &= SJ_{id\_sucursal=id\_sucursal}(Sucursal_{19}) \end{aligned}$$

Las relaciones *Productor* y *SuministroPedidoSucursal* no serán fragmentadas puesto que no hay un criterio lógico suficientemente claro para hacerlo.

### 3.2. Replica de fragmentos

Nuestra aproximación no incluirá replicación alguna de fragmentos. Consideramos que los costes de actualización y mantenimiento no se equiparan a los posibles *downtimes* que podrían sucederse al tener una única entidad por fragmento.

Una posible alternativa a esta aproximación podría incluir la réplica de la tabla *Productor* (teóricamente esta tabla no debería actualizarse demasiado) o también de *SuministroPedidoSucursal* (teniéndose por ejemplo, una copia en cada localidad) en vista a un acceso más directo al flujo de pedidos entre sucursales.

Una última aproximación mucho más conservadora y preventiva sería la replicación de cada fragmento en dos o más localidades distintas. Esto conllevaría un gasto en actualizaciones muy elevado. Sin embargo, la seguridad de la base de datos frente a *downtimes* y pérdidas de información sería más fiable.

### 3.3. Asignación de fragmentos, réplicas y relaciones

Una vez estructurada la fragmentación, realizaremos la asignación de los diferentes fragmentos en cada localidad teniendo en cuenta la división por localización geográfica que nos piden en el enunciado.

- $Sucursal_i, Cliente_i, Vino_i, Empleado_i, SuministroPedidoCliente_i, Almacen_i, SuministroPedidoSucursal$  y  $Productor$  con  $i \in \{1, 2, 3, 4, 5\}$   
 $\implies$  Se asignan a la localidad de **Madrid**

(*SuministroPedidoSucursal* y *Productor* pueden inicialmente asignarse a cualquier localidad)

- $Sucursal_i, Cliente_i, Vino_i, Empleado_i, SuministroPedidoCliente_i$  y  $Almacen_i$  con  $i \in \{6, 7, 8, 9\} \implies$  Se asignan a la localidad de **Barcelona**
- $Sucursal_i, Cliente_i, Vino_i, Empleado_i, SuministroPedidoCliente_i$  y  $Almacen_i$  con  $i \in \{10, 11, 12, 13, 14\} \implies$  Se asignan a la localidad de **La Coruña**
- $Sucursal_i, Cliente_i, Vino_i, Empleado_i, SuministroPedidoCliente_i$  y  $Almacen_i$  con  $i \in \{15, 16, 17, 18, 19\} \implies$  Se asignan a la localidad de **Sevilla**

Gráficamente podemos ver la asignación a cada una de las cuatro localidades de cada fragmento de la siguiente forma.

<b>Madrid LOCALIDAD1</b>	<b>Barcelona LOCALIDAD2</b>	<b>La Coruña LOCALIDAD3</b>	<b>Sevilla LOCALIDAD4</b>
Castilla-León	Cataluña	Galicia	Andalucía
Castilla-La Mancha	Baleares	Asturias	Extremadura
Aragón	País Valenciano	Cantabria	Canarias
Madrid	Murcia	País Vasco	Ceuta
La Rioja		Navarra	Melilla

Figura 2: Asignación de fragmentos a las localidades.

### 3.4. Reconstrucción de las relaciones globales a partir de los fragmentos

Podemos reconstruir las relaciones globales a partir de los diferentes fragmentos. En la base de datos esta relación global vendrá dada por la vista correspondiente.

- $Sucursal = Sucursal_1 \cup Sucursal_2 \cup \dots \cup Sucursal_{19}$
- $Cliente = Cliente_1 \cup Cliente_2 \cup \dots \cup Cliente_{19}$
- $Vino = Vino_1 \cup Vino_2 \cup \dots \cup Vino_{19}$
- $Empleado = Empleado_1 \cup Empleado_2 \cup \dots \cup Empleado_{19}$
- $SuministroPedidoCliente = SuministroPedidoCliente_1 \cup \dots \cup SuministroPedidoCliente_{19}$
- $Almacen = Almacen_1 \cup Almacen_2 \cup \dots \cup Almacen_{19}$



## Referencias

- [1] Guión de la práctica de la asignatura.
- [2] <https://pradogrado2021.ugr.es/auth/saml/login.php>
- [3] <https://www.w3schools.com/sql/default.asp>
- [4] [https://docs.oracle.com/cd/B14117\\_01/appdev.101/b10807/13\\_elems002.htm](https://docs.oracle.com/cd/B14117_01/appdev.101/b10807/13_elems002.htm)