

## 15. Bases de datos //Acabar apuntes de base de datos, no están listos.

### Niveles de abstracción

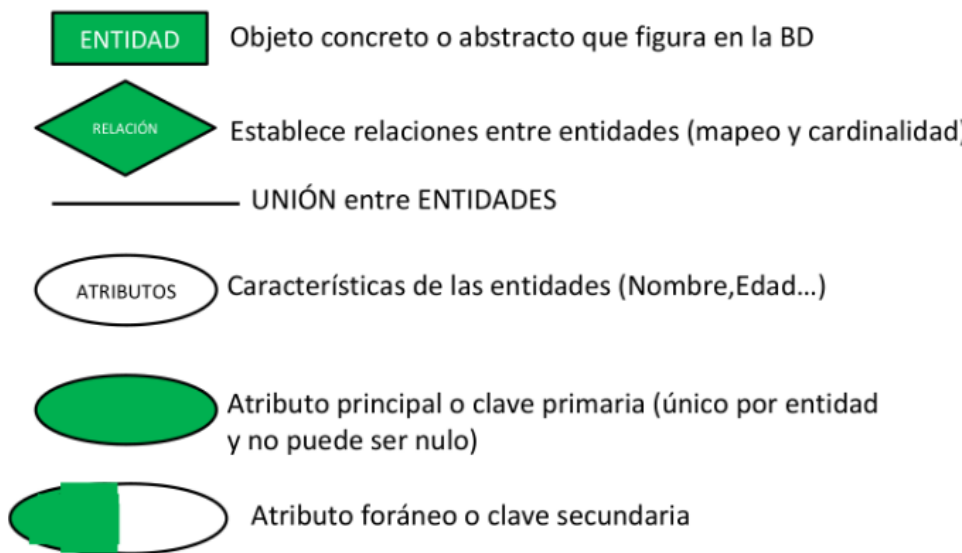
Simplifica usabilidad de usuarios con Bases de datos (BBDD) y Sistema.

- ➔ Nivel físico: Dispositivos de almacenamiento Local o remoto. Nivel más bajo de abstracción. Dónde tenemos nuestra BD.
- ➔ Nivel conceptual: Definición + Relación de DATOS: Interno, como está construido.
- ➔ Nivel visión: Usuario visualiza BD en el sistema. Nivel más alto de abstracción. Diferentes vistas

### Diagrama/modelo Entidad – Relación

La clave primaria es un identificador único. Solo lo tiene el esa clave primaria.

Elementos:



Clave foránea: Una que relaciona con otras tablas.

Cuando solo es uno se pone un 1:

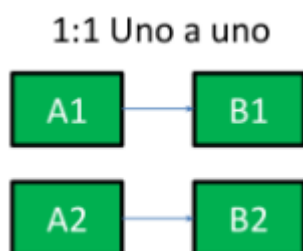
Cuando es muchos se pone una n:

Quien obtiene la clave foránea es quien tiene muchos.

### Diagrama Entidad – Relación: Grado de cardinalidad

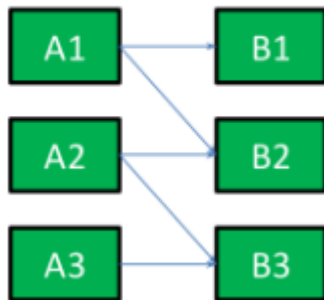
Unión de dos cardinalidades de dos entidades.

1:1. Uno a uno. Una entidad, solo puede tener otra entidad.



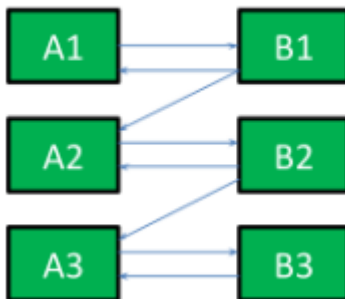
1:N Uno a muchos: Una entidad puede tener muchas entidades.

## 1:N Uno a muchos



N:M Mucho a muchos

N:M Muchos a muchos



### Diagrama Entidad – Relación



- **Ventajas:** Diseño de alto nivel, ya que refleja con bastante precisión el esquema conceptual. Los diagramas ER permiten mantener una visión global del diseño y favorecer la comunicación entre diseñadores.
- **Desventajas:** Falta de soporte formal, ya que los SGBD no lo implantan directamente. Casi siempre hay que transformarlo en un modelo de más bajo nivel, esto significa, más cercano a la comprensión del sistema – máquina:
  - **Un nivel intermedio:** El lenguaje SQL
  - **Un nivel alto:** el usuario lo puede entender más fácilmente pero la máquina que computa el código, no tanto.
  - **Un nivel bajo:** más cercano a la comprensión del sistema – máquina. Entiende bytes pero no los textos.

## Unified Modeling Language

Es un modelo internacional, que ayudaba al diseño e implementación de sistemas de software complejos. Cuando quieres hacer un flujo de bases de datos (como se hace un préstamo por parte de un usuario con entidades, Estudiantes-prestaos-libros). Se crea para los diagramas que definen la estructura y el comportamiento.

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

Los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

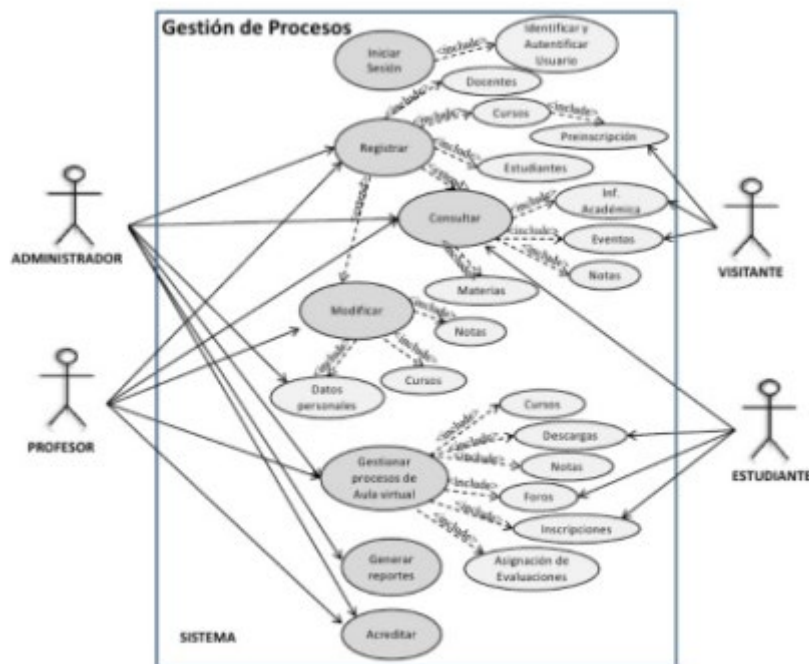
UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML.

Vamos a establecer información relacionada, se acerca a la base de datos, pero no va hacer que yo pueda insertar el préstamos de un libro en ningún sitio concreto. El UML no es un lenguaje de base de datos pero es más avanzado que el Modelo Entidad-Relación.

\*Cuando hacemos el modelo Entidad Relación ponemos atributos, pero hemos establecido la relación que tenga la forma en la que un libro es leído por un alumno. Que sepas la parte del libro que ha sido leído.

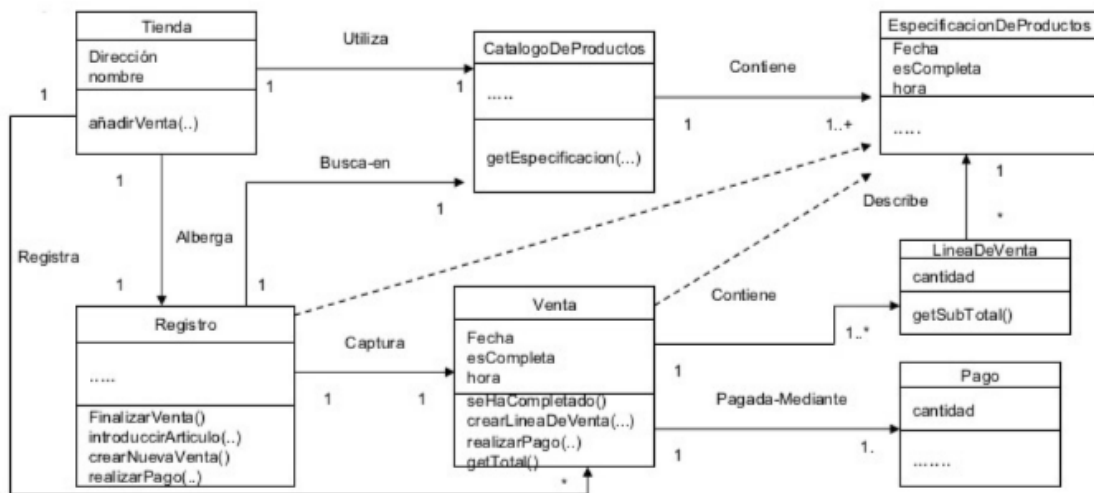
## UML: Diagrama de casos de usos

Representa gráficamente los casos de usos, interacción con el sistema.

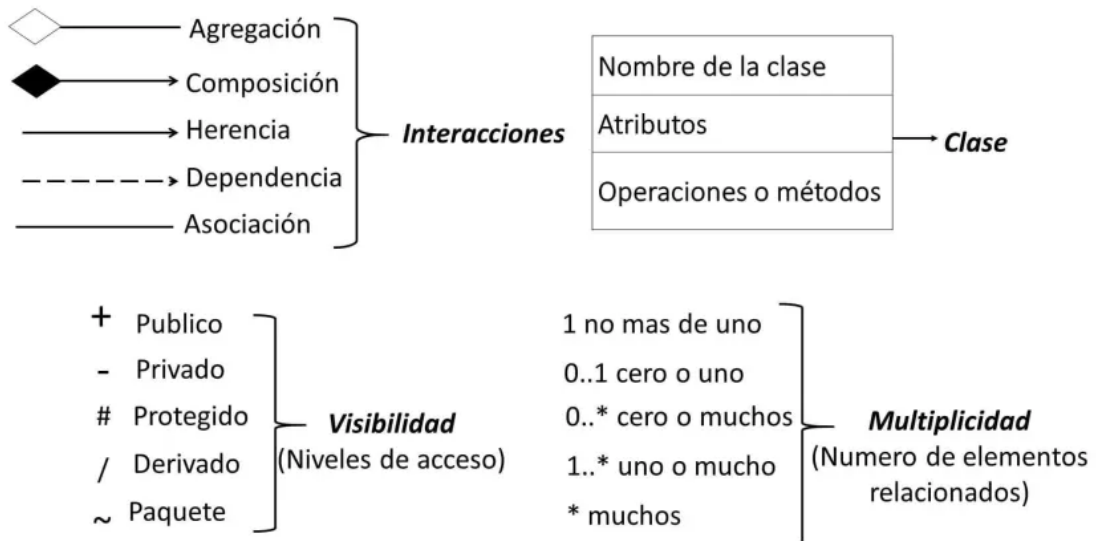


## UML: Diagrama de clases

Muestra un conjunto de clases interfaces y sus relaciones



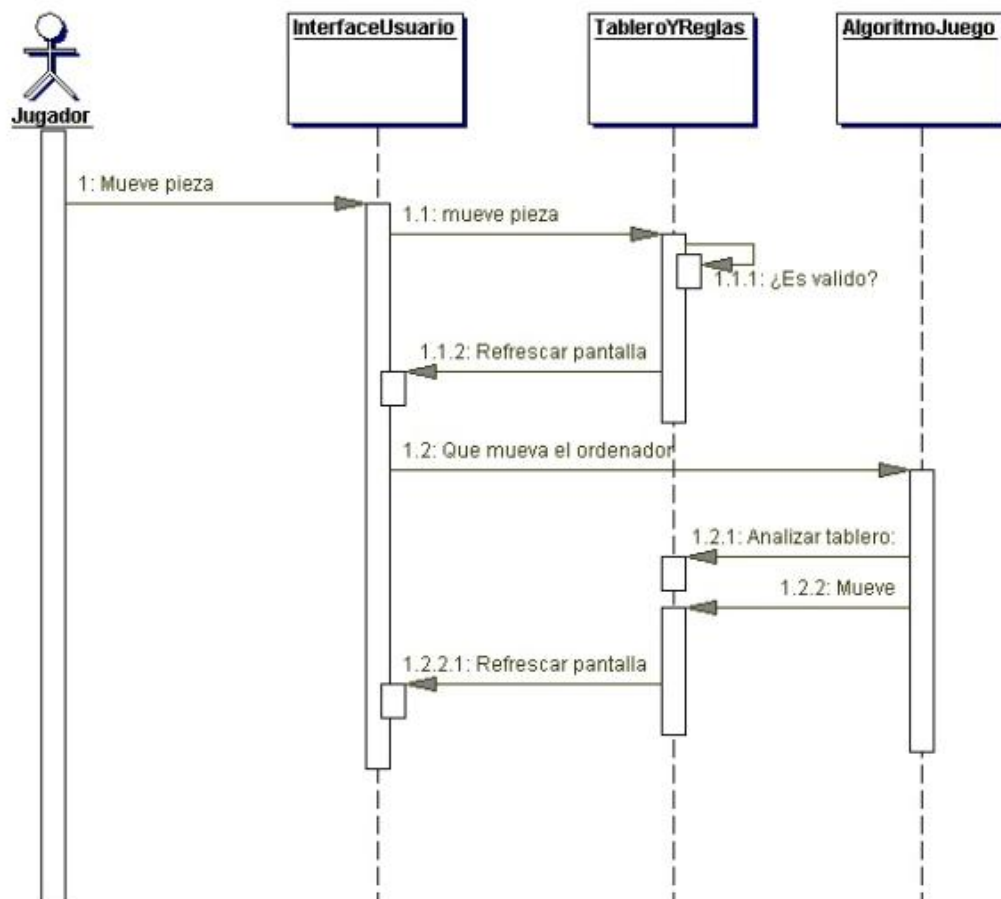
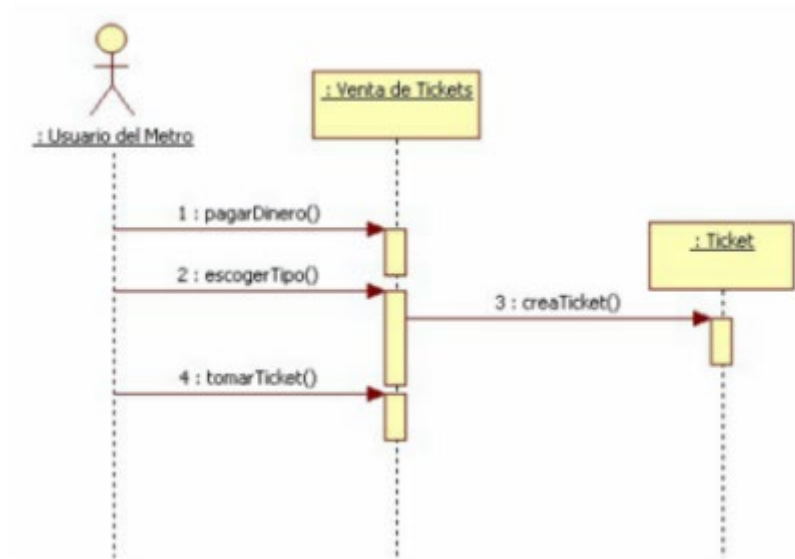
## Elementos y símbolos en los diagramas de clases UML



## UML: Diagrama de secuencia

Muestra la interacción de los objetos que componen un sistema de forma **temporal**.

Representa gráficamente los casos de usos, interacción con el sistema. Es un diagrama de alto nivel.



EJERCICIO: Diagrama de secuencia de un juego "Buscaminas".

## **15.2 Concepto y origen de las Bases de Datos (DDBB) y de los Sistemas Gestores de Bases de Datos (DBMS). Evolución de la Base De Datos (BD).**

Si trataremos los datos solo con ficheros, no tendríamos información relacionada.

\* En el mundo actual es fundamental el procesamiento de datos para obtener información que permita tomar decisiones, cada vez se genera y se utilizan más datos, lo cual dificulta su procesamiento.

Etapas de evolución del tratamiento de información:

- 1ª etapa, principios de lo 1960, primeras aplicaciones integran datos en programas

en forma de CONSTANTES, se analizan partes del mundo científico

- 2ª etapa, mediados 60s, aparecen los FICHEROS que representan partes del mundo

real, solo se accedían de forma secuencial y poco después de forma aleatoria.

- 3ª etapa, finales 60s, para evitar problemas en ficheros aparecen las BASES DE
- DATOS, que pretenden modelar grandes partes del mundo real mediante la relación de varios archivos, evitando la redundancia e incoherencias.

Una base de datos es “una colección de datos que están relacionados entre sí, que tienen una definición y una descripción comunes y que están estructurados de forma particular”

Los elementos que forman las bases de datos.

1. **Los datos:** representación de hechos, conceptos o instrucciones, estructurados, para comunicar, interpretar y representar datos de informaciones almacenados y relacionados entre si:
  - Microdatos: datos simples, como por ejemplo nombres, direcciones, ...
  - Macrodatos: aglomeración de datos, como las tablas.
2. **El diccionario de datos (DD):** conjunto de metadatos que contiene datos sobre los datos, que da lugar a una metabase, una BD que describe otra BD
  - Esquemas externos, conceptuales e internos
  - Descripciones de registros, campos y relaciones a cada nivel
  - Procedimientos de autorización de accesos y validación de datos
  - Correspondencias externas/conceptual y conceptual/interna
  - Referencias cruzadas
    - Un programa qué datos utiliza
    - Unos datos y qué programas lo utilizan o lo modifican

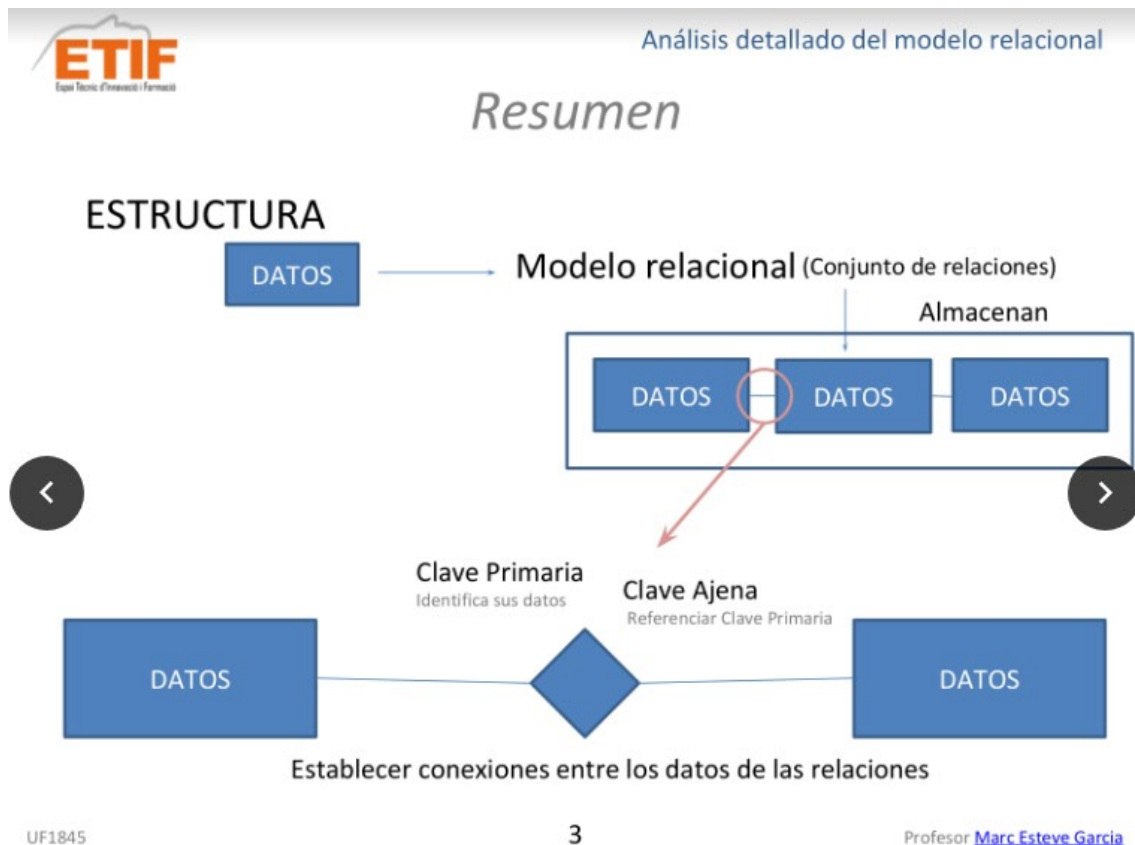
//Es lo que describe la base de datos. Sus esquemas internos. Todos los campos y relaciones describen la base de datos.
3. **El sistema de gestión de la base de datos (SGBD):** Software o conjunto de programas que gestionan las BD.El software encargado de administrar y elaborar bases de datos.
  - Funciones: Crear, actualizar, consultar, proteger las BBDD
4. **Los usuarios: todas las personas que utilizan las BD**
  - Informáticos (especializados y comprenden la parte interna)
    - Administradores (ABD): gestionan la base de datos y proporcionan herramientas para el resto de usuarios

- Desarrolladores: complementan a los administradores, analizan y programan aplicaciones para usuarios finales.
- No informáticos: serán los tipos de usuarios a los que va dirigida la Base de Datos, utilizarán datos para el trabajo cotidiano y accederán los datos según sus permisos para facilitar el acceso y manipulación

Una cosa es la biblioteca de datos y otra cosa es la definición de estas bases de datos.

### **5.3 Análisis detallado del modelo relacional**

#### **3.1 Estructura de los datos. Estructura.**



**Lenguaje de consulta SQL:** es un lenguaje declarativo de acceso a las bases de datos relacionales, que permite definir y manipular las bases de datos.

**El modelo relacional** es un modelod e datos, y, como tal, hay que tener en cuenta los aspectos de los datos:

- ➔ La estructura
- ➔ La manipulación
- ➔ La integridad

Las relaciones están formadas de:

- ➔ **Atributo:** Se trata de cada una de las columnas de la tabla. Vienen definidas por un nombre y pueden contener un conjunto de valores.

- ➔ **Tupla:** También llamado registro. Esa fila de valores. Se trata de cada una de las filas de la tabla, es importante señalar que no se pueden tener tuplas duplicadas en una tabla.

**\*Dominio:** conjunto de valores atómicos o indivisibles, es decir, que por muy largo que sea un valor, no tiene una estructuración interna para un sistema de gestión de base de datos relacional. Serán de una tipología determinada: (establece los valores que va a tener). Pueden ser de dos tipos:

- ➔ Dominio predefinido: Corresponde a los tipos de datos que normalmente proporcionan los lenguajes de base de datos. Como por ejemplo los enteros, las cadenas de texto, los reales, los numéricos, etc.
- ➔ Los dominios definidos por el usuario: que pueden ser más específicos, esta definición de dominio debe constar, como mínimo del nombre del dominio y de la descripción de los valores que la forman.

La relación se componen de:

- ➔ Esquema:
- ➔ Extensión:

Nosotros haremos relaciones entre tablas. Se relacionaran por sus atributos

**Hay una serie de características que las relaciones deben cumplir:**

- ➔ La atomicidad de los valores de los atributos:
- ➔ Las tuplas no se pueden ordenar:
- ➔ Los atributos no se pueden ordenar

**Hay diferentes claves que se pueden encontrar en las relaciones:**

- ➔ Las claves: conjuntos de atributos que identifica a una relación.
  - Super clave: Atributo o conjunto de atributos que identifica de forma única a una identidad. Es la capacidad de ser clave primaria en las diferentes relaciones.
    - La clave prima pase a convertirse en superclave ya que define directamente los atributos.
  - Clave candidata: "Todos aquellos atributos pertenecientes a una relación que pueden identificar inequívocamente a cada tupla".
  - Clave foránea:

**Operadores del modelo**

- ➔ La actualización de datos: consiste en hacer que los cambios que se producen en la vida real, queden reflejados en las relaciones de la base de datos. Hay **tres operaciones básicas de actualización:**
  - Inserción: sirve para añadir una o más tuplas a una relación.
  - Borrado: que sirve para eliminar una o más tuplas de una relación
  - Modificación: que sirve para cambiar a los valores que tienen una o más tuplas de una relación para uno o más de sus atributos.
- ➔ El valor NULO "Es un atributo que no contiene ningún valor"
- ➔ El **grado** de una relación " Es el número de atributos que pertenecen a su esquema".
- ➔ La **cardinalidad de una relación** \* Es el número de tuplas que pertenecen a su extensión.



- ➔ La consulta de datos: consiste en la obtención de datos a partir de las relaciones que contiene la base de datos.
- ➔
  - Lenguajes basados en el álgebra relacional
  - Los lenguajes basados en el cálculo relacional

## La integridad

Es la propiedad de los datos de corresponder a representaciones creíbles del mundo real. Es necesario que se cumplan varias condiciones, en general, estas condiciones garantizan la integridad de los datos y pueden ser de dos tipos.

- ➔ Las restricciones de integridad del usuario. Condiciones específicas que se deben cumplir en una base de datos concreta.
- ➔ Las reglas de integridad de modelo.

Existen reglas que todo sistema gestor de base de datos (SGBD) relacional debe garantizar que se cumpla:

- ➔ **Regla de integridad de unicidad de la clave primaria:** Toda clave primaria que se elija para una relación no debe tener valores repetidos, es decir, la clave primaria o el conjunto de atributos que forman la clave primaria de una relación no puede tener en ningún momento dos tuplas con la misma combinación de valores para los atributos de clave primaria. El SGBD relacional deberá garantizar el cumplimiento de esta regla de integridad en todas las inserciones y modificaciones que afecten a atributos que pertenecen a la clave primaria de la relación.
- ➔ **Regla de integridad de entidad de la clave primaria:** Establece que los atributos de la clave primaria de una relación no pueden tener valores nulos. La clave primaria o el conjunto de atributos que formen la clave primaria de una relación, no puede tener ninguna tupla con algún valor nulo en algún
- ➔ **Regla de la integridad referencial:** Esta regla determina que todos los valores que toma una clave ajena deben ser valores nulos o valores que existen en la clave primaria a la que referencia.

Los dominios pueden ser de dos tipos, predefinidos o definidos por el usuario. El último nos resulta muy útil, ya que nos permite determinar de forma más específica cuáles serán los valores administrados por el atributo.

Hay una serie de reglas para mantener la integridad del dominio. Relacionada con la idea de dominio y establece dos condiciones. Regla de integridad del dominio:

- ➔ La primera condición: Consiste en que un valor no nulo de un atributo debe pertenecer al dominio del atributo.
- ➔ La segunda condición: Es más difícil, sobretodo en el caso de dominios definidos por el usuario y en los sistemas gestores de bases de datos (SGBD) actuales que no lo soportan para este tipo de dominios. Esta condición sirve para establecer que los operadores que pueden aplicarse sobre los valores dependen de los dominios de estos valores, es decir, un operador determinado solo se puede aplicar sobre valores que tengan dominios que le sean adecuados.

## Álgebra relacional

El álgebra relacional se basa en la teoría de conjuntos para especificar consultas en una base de datos relacional. Es necesario definir varios pasos que sirven para ir

construyendo, mediante operaciones de álgebra relacional, una nueva relación que contenga los datos que responden a

Es el paso previo a un lenguaje SQL. Se pueden dividir **las operaciones de álgebra lineal**:

**Según se pueden expresar o no en términos de otras operaciones:**

- **Operaciones primitivas:** aquellas operaciones a partir de las cuales podemos definir el resto.
- **Operaciones no primitivas:** el resto de las operaciones del álgebra relacional que no son estrictamente necesarias, porque se pueden expresar en términos de las primitivas, sin embargo, las operaciones no primitivas permite formular algunas consulta de forma más cómoda. Existen distintas versiones del álgebra relacional, según las operaciones no primitivas que se incluyen.

\*Las que se utilizan con mayor frecuencia son la intersección y la combinación.

**Según el número de relaciones que tienen como operando:**

- **Operaciones binarias:** Son aquellas que tienen dos relaciones como operando. Son binarias **todas las operaciones, excepto la selección y la proyección.**
- **Operaciones unarias:** Son aquellas que tienen una sola relación como operando. Son operaciones unarias la selección y la proyección.

**Según se parecen o no a las operaciones de la teoría de conjuntos**

- **Operaciones conjuntistas:** aquellas que se parece a las de la teoría de conjuntos. Se trata de la **unión, la intersección, la diferencia y el producto cartesiano.**
- **Operaciones específicamente relacionales:** Son el resto de las operaciones, como la selección, la proyección y la combinación.

\*Las operaciones del álgebra relacional obtienen como resultado una nueva relación.

Supongamos, que tenemos EMPLEADOS\_ADM UU EMPLEADOS-PROD para obtener la unión de las relaciones EMPLEADOS\_ADM y EMPLEADOS\_PROD, el resultado de la operación es una nueva relación que tiene la unión de las tuplas de las relaciones de originales.

### Operadores

- **Redenominar** :=Es uno de los principales operando. **Asigna un nombre** a la relación que resulta de una operación de álgebra relacional y se hace de la siguiente forma,  $R := E$ , siendo R la relación y E la operación del álgebra relacional. Estamos hablando de asignar un valor.
- **La Unión (U).** Es un operador binario. Tabla T unida de la tabl S  $\rightarrow T \cup S$ 
  - Deben ser **compatibles**. Decimos que dos relacionaes T y S son relaciones compatibles si:
    - Tienen el mismo grado
    - Tiene que tener el mismo dominio.
    - No se repiten. Si se repiten no se ponen.

### Tabla relación de Empleados de Administración

Esquema y extensión de EMPLEADOS\_ADM

DNI	nombre	apellido	edificiodesp	númerodesp
73519759p	Juan	García	Marina	120
75645339k	Marta	Roca	Marina	120

### Tabla relación de Empleados de Producción

Esquema y extensión de EMPLEADOS\_PROD

DNI	nombreemp	apellidoemp	edificiodesp	númerodesp
75645339k	Marta	Roca	Marina	120
57885791c	Carlos	Buendía	Diagonal	120
61527141m	Elena	Pla	Marina	230
40141287d	Salvador	García	NULO	NULO
80845987b	Pedro	González	NULO	NULO

$\text{EMPLEADOS\_ADM} \cup \text{EMPLEADOS\_PROD}$

El resultado será el siguiente:

DNI	nombreemp	apellidoemp	edificiodesp	númerodesp
73519759p	Juan	García	Marina	120
75645339k	Marta	Roca	Marina	120
57885791c	Carlos	Buendía	Diagonal	120
61527141m	Elena	Pla	Marina	230
40141287d	Salvador	García	NULO	NULO
80845987b	Pedro	González	NULO	NULO

Tienen el mismo grado. En la unión todo lo que no se repita lo coloca.

- ➔ **Intersección ( $\cap$  al revés):** En las relaciones las que son tuplas idénticas, las podremos encontrar en la intersección. Es binaria. Han de ser compatibles. Solo se encontrarán en la nueva relación aquellas tuplas cuyos atributos de los campos son comunes en ambas tuplas.

$R : = \text{EMPLEADOS\_ADM} \cap \text{EMPLEADOS\_PROD}$

**\*Ejemplo, en el caso de las dos tablas anteriores. Solo se mostrará la siguiente tabla:**

DNI	nombreemp	apellidoemp	edificiodesp	númerodesp
75645339k	Marta	Roca	Marina	120

→ **La diferencia (-):** es una operación en la que, a partir de dos relaciones, se obtiene una nueva relación formada por todas las tuplas que están en la primera relación y no están en la segunda. Solamente los que coinciden son los que se restan. La diferencia es binaria también.

**Solo tiene sentido si se aplica a relaciones que tengan tuplas similares.**

Es necesario que las relaciones sean compatibles.

\*Los atributos del esquema de la relación resultante de  $T - S$  coinciden con los atributos del esquema de la relación  $T$  y la extensión de la relación resultante de  $T - S$  es el conjunto de tuplas que pertenecen a la extensión de  $T$ , pero no a la de  $S$ .

$R := EMPLEADOS\_ADM(T) - EMPLEADOS\_PROD$

DNI	nombreemp	apellidemp	edificiodesp	numerodesp
73519759p	Juan	García	Marina	120

→ **Producto cartesiano (X):** A partir de dos relaciones obtiene una nueva relación formada por todas las tuplas que resultande concatenar tuplas de la priemra relación con tuplas de la segunda.

- No necesita ser del mismo grado para llevar a cabo eso.

//acabar de entender

Edificios empleados

Esquema y extensión de **EDIFICIOS\_EMP**

edificio	supmediadesp
Marina	15
Diagonal	10

DESPACHOS

Esquema y extensión de **DESPACHOS**

edificio	número	superficie
Marina	120	10
Marina	230	20
Diagonal	120	10
Diagonal	440	10

Escasamente se utiliza de forma explícita. El resultado que da no suele ser útil para resolver las consultas habituales. Es una operación primitiva del álgebra relacional a partir de la cual se define otra operación del álgebra → esta combinación se utiliza con mucha frecuencia.

Para realizar el producto cartesiano de las relaciones **DESPACHOS** y **EDIFICIOS\_EMP** es necesario red denominar atributos primeramente:

**EDIFICIOS(nombreedificio, supmediadesp) := EDICIOS\_EMP(edificio, supmediadesp)**

**R := EDIFICIOS x DESPACHOS**

El resultado da esto:

nombreedificio	supmediadesp	edificio	numero	superficie
Marina	15	Marina	120	10
Marina	15	Marina	230	20
Marina	15	Diagonal	120	10
Marina	15	Diagonal	440	10
Diagonal	10	Marina	120	10
Diagonal	10	Marina	230	20
Diagonal	10	Diagonal	120	10
Diagonal	10	Diagonal	440	10

#### Operadores específicamente relacionales

➔ **Selección.** Es un tipo de operador unario. De una sola tabla especificamos solo una tupla o varias.

- $\emptyset$  dentro del dominio.

Edificio = Marina y superficie > 12; se indicaría DESPACHOS (edificio = Marina y superficie > 12)

//esto es el proceso previo a escribirlo con el lenguaje SQL.

➔ **Proyección.** Una operación que sirve para elegir algunos atributos de una relación y eliminar el resto, es decir, la proyección es una operación que, a partir de una relación, se obtiene una nueva relación formada por todas las tuplas de la relación de origen, que resultan de eliminar unos atributos especificados.

- Sintaxis ➔ RELACIÓN [nombre, apellido]
- Atributos y tuplas.

EMPLEADOS\_ADM[nombre, apellido]

Elimina implícitamente todas las tuplas repetidas. El resultado de una proyección es una relación válida y no puede tener repeticiones de tuplas.

- Es una operación unaria. Y se extraen columnas.

Usuario 'COLUMNAS'

RELACIÓN[nombre, apellido] Atributos y tuplas.

➔ **Combinación.** Como una proyección pero la forma todas las tuplas que resultan de concatenar tuplas de la primera relación con tuplas de la segunda que cumplen una condición de combinación especificada. Concatena aquellas que cumplan una condición determinada:

- UNARIO 'COLUMNAS'
- RELACIÓN [nombre, apellido]

- Atributos y tuplas.

→  $REL1[atributo1=atributo2, condición]REL2$

### Limitaciones

### Reglas formales

\*Si tenemos la tercera forma normal significa que es más restrictiva que la primera. Incluye las otras dos.

→ Primera forma normal

Tabla que no cumple la primera Forma Normal 1FN

Trabajador		
DNI	nombre	departamento
10111213m	Andrés	Mantenimiento
25262722p	Andrea	Dirección Gestión

Es decir, tiene más de un valor en el campo departamento en la segunda tupla

**Marc** hoy a las 15:50

Entonces, en primer Forma Normal 1FN

Trabajador		
DNI	nombre	departamento
10111213m	Andrés	Mantenimiento
25262722p	Andrea	Dirección
25262722p	Andrea	Gestión

→ Segunda forma normal

Tabla que cumple la 1FN pero no la segunda:

Alumnos				
DNI	cod curso	nombre	apellido1	nota
12121219A	34	Pedro	Valiente	9
12121219A	25	Pedro	Valiente	8
3457775G	34	Ana	Fernández	6
5674378J	25	Sara	Crespo	7
5674378J	34	Sara	Crespo	6

Dividimos la tabla en dos, cuando tenemos dependencias funcionales completas

Ejemplo DNI: "nombre" y "apellidos"

Alumnos		
DNI	nombre	apellido1
12121219A	Pedro	Valiente
3457775G	Ana	Fernández
5674378J	Sara	Crespo

Asistencia		
DNI	cod curso	nota
12121219A	34	9
12121219A	25	8
3457775G	34	6
5674378J	25	7
5674378J	34	6

Esta tabla si que cumple la 2FN (segunda forma normal)

### → Tercera forma normal

- Hay una dependencia transitiva entre ambos valores. Provincia depende del código provincia. Lo que hace no esté en tercera forma normal. Cuesta más de entender que la segunda porque lo que se le está añadiendo aquí, además, es el componente de la dependencia transitiva entre ambos campos.

Ex:



Marc hoy a las 16:00

La siguiente tabla no cumple la 3FN

Alumnos				
DNI	nombre	apellido1	cod_provincia	provincia
12121349A	Salvador	Velasco	34	Palencia
12121219A	Pedro	Valiente	34	Palencia
3457775G	Ana	Fernández	47	Valladolid
5674378J	Sara	Crespo	47	Valladolid
3456858S	Marina	Serrat	08	Barcelona

Entonces para que se cumpla tenemos que dividir la tabla para la dependencia funcional transitiva que existe con cod\_provincia y provincia (redundante información) en la primera tabla

Alumnos			
DNI	nombre	apellido1	cod_provincia
12121349A	Salvador	Velasco	34
12121219A	Pedro	Valiente	34
3457775G	Ana	Fernández	47
5674378J	Sara	Crespo	47
3456858S	Marina	Serrat	08

Provincia	
cod_provincia	provincia
34	Palencia
47	Valladolid
08	Barcelona

➔ Forma normal de Boyce-Codd:

No hay una clave primaria clave y todas son candidatas.

Lo que hay que hacer es estrechar al máximo las relaciones. En este caso vemos que una persona es como máximo responsable de un departamento, por lo tanto solo con que indiquemos sus trabajadores. Y posteriormente los responsables de cada departamento, ya entendremos el departamento para el que trabaja cada trabajador. No hay una relación de uno a muchos que no haya separado.



Revisemos Forma Normal de Boyce Codd

Esta tabla no tiene clave candidata, no tiene una clave primaria entonces

(editado)

pero resulta que un responsable solo estará en un único departamento

Organización		
trabajador	departamento	responsable
Alex	Producción	Felipa
Arturo	Producción	Martín
Carlos	Ventas	Julio
Carlos	Producción	Felipa
Gabriela	Producción	Higinio
Luisa	Ventas	Eva
Luisa	Producción	Martín
Manuela	Ventas	Julio
Pedro	Ventas	Eva

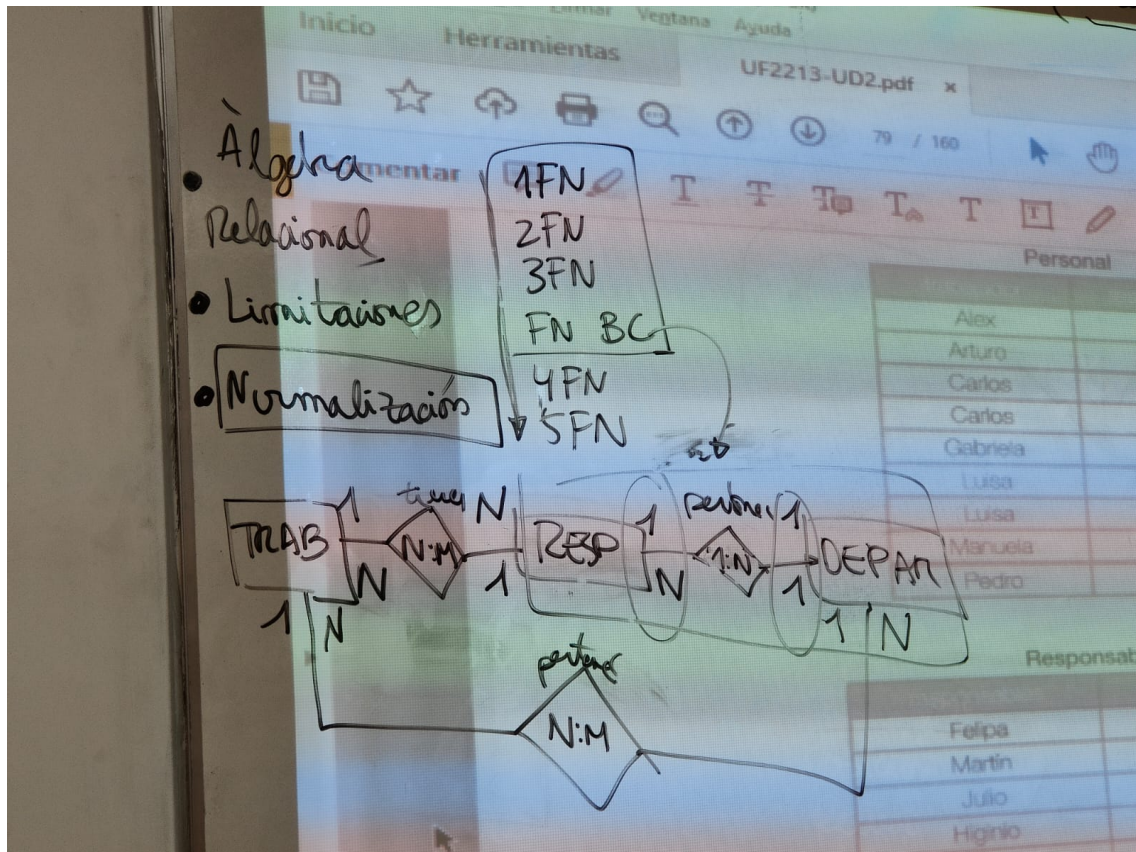
Esta es la clave

la relación funcional que existe entre un responsable solo esta en un solo departamento

podemos dividir la tabla con una clave primaria que sera clave foranea de la otra tabla

Personal	
trabajador	responsable
Alex	Felipa
Arturo	Martín
Carlos	Julio
Carlos	Felipa
Gabriela	Higinio
Luisa	Eva
Luisa	Martín
Manuela	Julio
Pedro	Eva

Responsables	
responsables	departamento
Felipa	Producción
Martín	Producción
Julio	Ventas
Higinio	Producción
Eva	Ventas



Hasta aquí ya tendríamos una base de datos bastante buenas.

La dependencia multivaluada es la base de la cuarta forma normal. Si las funcionales eran la base de la segunda, la tercera y Boyce-Codd, estas son la base de la cuarta forma normal.

#### → Cuarta forma normal

- Entre ellas dependen por separado. Todas son relaciones de muchos a muchos. En la Boyce-Codd alguno encontramos peor separado. Pero aquí no se puede dejar ninguna. Todas las relaciones son de muchos a muchos. Entre dos atributos no hay una dependencia clara. En Boyce-Codd si que tenemos la relación uno a muchos.
- Lo puedes dividir entre diferentes elementos, aunque lo suyo es ver cual es más reducido, en este caso por cada profesor, cuales son de cursos diferentes.

#### → Quinta forma normal: A veces se ha dudado de ella por que no se sabe si realmente mejora.

- Dependencia Join: la tabla resultante de tomar un subconjunto de los atributos con una tabla. Tabla formada por unas cuantas columnas de la tabla original. Si queremos normalizarla para reducir información podemos hacer una división pero a triple. Puede ser que sea más complejo. Se reduce tanto que se divide la tabla en tres.

1. B → La estructura, la manipulación y la integridad
2. D → Foráneas
3. B → La inserción, el borrado, y la modificación.
4. B → Las restricciones de integridad de usuario y las reglas de integridad del modelo.
5. B → Operaciones primitivas y no primitivas
6. A → Redenominar
7. D → Intercisión
8. Combinación y proyección. También puede ser combinación con corchetes.
9. C → También incluye la cuarta
10. D → Está en la página 84.

## **15. 4 Modelos avanzados de BD**

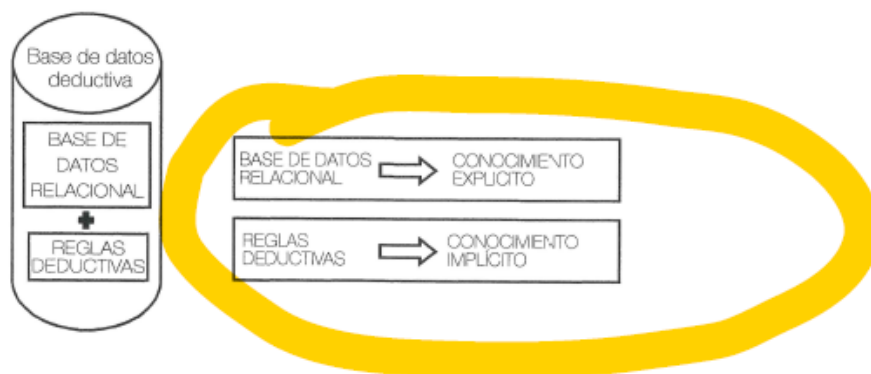
### **4.1 BD Deductivas**

Son un sistema de gestión de bases de datos que permiten mediante información **extensión al de reglas y hechos**, la **deducción de información extra que complementa la que nuestra base de datos tiene almacenada**.

Aparecen en la década de los 80 y su desarrollo ha contribuido a obtener mejoras en cuanto a los resultados y desarrollos fundamentales relativos al campo de la lógica y la inteligencia artificial.

→ Este tipo de base de datos se usa con frecuencia en el campo de la deducción automática.

A continuación veremos su esquema:



El lenguaje más utilizado es el datalog (no procedimental basado en un lenguaje de programación lógica de Prolog que simplifica la escritura de consultas simples y las hace más sencillas).

Esta base de datos presenta un alto grado de lógica matemática. A su vez, de computación. → Por la estrecha relación que existe entre ambos conceptos: por esto se puede identificar el término base de datos deductiva con el término base de datos lógicas.

- ➔ Almacenar el mayor número de datos mediante un programa lógico e intentando seleccionar los datos para guardar en la memoria principal, de modo que se almacene una determinada información.
  - Cuando seleccionas los datos para guardar en la memoria principal, se almacena una determinada información.
- ➔ Se utilizan estos tipos de especificaciones:
  - Hechos: Conocimientos explícitos y se conocen como la base de datos extensional → podemos identificar con las siglas EDB.
    - Son las tuplas en una base de datos relacional con la diferencia de que **se indica el nombre de tupla pero no se indican los nombres de los atributos**.
      - Una tupla intenta describir algún hecho de la realidad que dependen de los atributos. Por ello se puede concluir que los atributos dependen de la posición en la tupla.

\*Recordar que una **tupla** "es cada una de las filas de una tabla. Es importante señalar que no se puede tener filas duplicadas en una tabla".

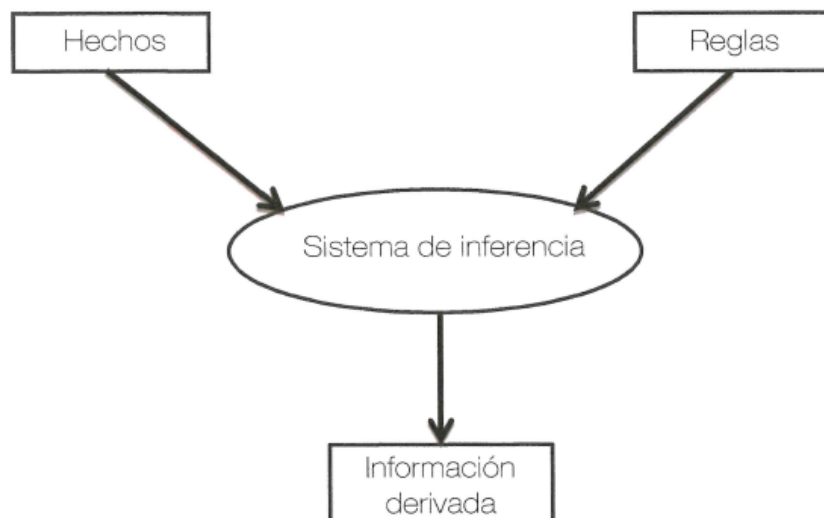
- Reglas (pagina 95): Son los conocimientos implícitos y son conocidas como la base de datos intencional, la cual podemos identificar con las siglas IDB.
  - Puede tener parecido a las vistas relacionales. Indican como obtener mediante los hechos guardados nuevos hechos y las relaciones indirectas entre las entidades.
  - Normalmente se escriben utilizando un lenguaje declarativo, se indica lo que se desea, pero no el algoritmo para llegar a esa meta. Las bases de datos deductivas combinan el modelo relacional para representar los datos con el modelo de la programación lógica para la representación y manejo de las reglas.

\*Hay que recordar que las **vistas** "son tablas cuyos datos derivan de la base de datos global, por estos datos no existen realmente, solo se visualizan como una tabla virtual, por eso no hay ningún archivo que la represente".

- Sistemas de inferencia: también llamado mecanismo de deducción puede deducir hechos nuevos a partir de la base de datos interpretando las reglas. El modelo empleado por la base de datos deductivos está íntimamente relacionado con el modelo de datos relacional y sobre todo con el formalismo del cálculo relacional.
  - Relacionado también con el campo de la programación lógica y el lenguaje Prolog. Este es el punto de partida de los trabajos sobre bases de datos basados en lógica.
  - Datalog es un lenguaje similar a Prolog, está formado por reglas que junto a un conjunto de relaciones existentes componen el lenguaje, aunque la estructura gramatical se parece a la del Prolog, su semántica es diferente y nos permite trabajar con datos que no estén en la memoria principal.
- Podemos establecer dos tipos de inferencia computacional basados en la interpretación de las reglas por la teoría de la demostración:
  - **Mecanismo de inferencia ascendente**: //encadenamiento hacia delante o resolución ascendente. El sistema de

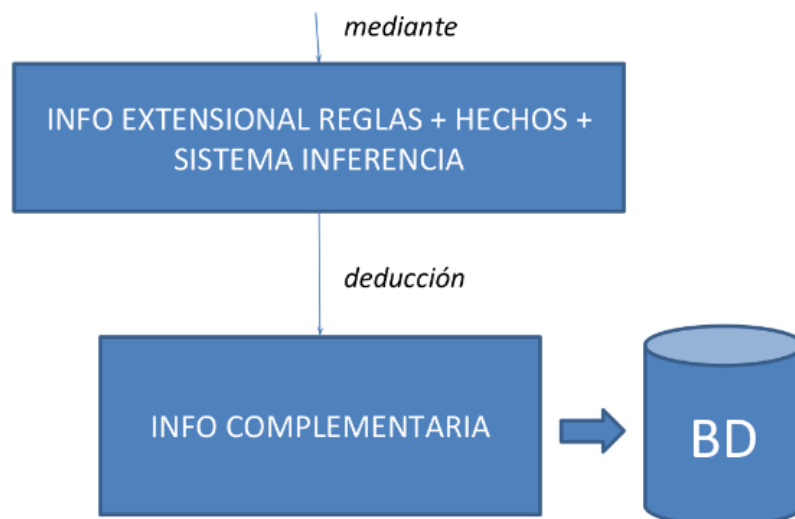
inferencia parte de los hechos y aplica las reglas para generar hechos nuevos.

- **Mecanismo de inferencia descendente:** //También llamado encadenamiento hacia atrás o resolución descendente.
  - Parte del predicado que es el objetivo de la consulta e intenta encontrar coincidencias con las variables que conduzcan a hechos válidos de la base de datos.
  - Retrocede desde el objetivo buscado para determinar hechos que lo satisfacen.
  - Si no existieran los hechos que buscamos, el sistema entonces buscará la primera regla cuya cabeza tenga el mismo nombre de predicado que la consulta.



## BD Deductivas

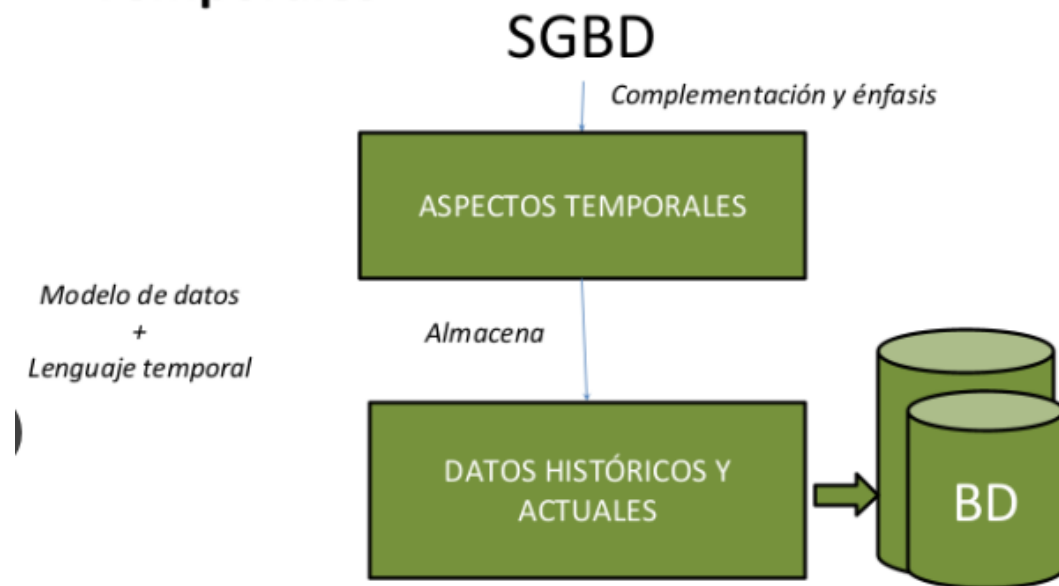
### SGBD



Son datos que no existen, sino que se generan a partir de las reglas.

### 4.2 BD Temporales

# BD Temporales



A partir de la página 98. Un sistema de gestión identificado con las siglas DBMS, es decir, data base management system.

Hay tres tipologías de bases de datos temporales, dependiendo de como utilizan o manejan el aspecto del tiempo y estas son::

- ➔ Base de datos temporales de tiempo transaccional:
  - Tiempo de acuerdo al momento en el que se almacena un hecho.
- ➔ Bases de datos temporales de tiempo válido: Registran el tiempo en el cual un hecho ocurrió en la realidad, → realizar correcciones sobre los datos reistrado.
- ➔ Bases de datos bitemporales: Cada estado se puede modificar para ctualziar el conocimiento de la realidad pasada, presente o futura. Se pueden actualizar generando nuevas versiones de los mismos estados.

Cada hecho almacenado posee una marca de tiempo asociada, y son:

- ➔ Tiempo valido: Este concepto hace referencia al conjunto de intervalos de tiempo durante los que el hecho es verdadero.
- ➔ Tiempo de transacción: Engloba el intervlaio de tiempo durante el cual ese hecho es cierto en el sistema ade base de datos.

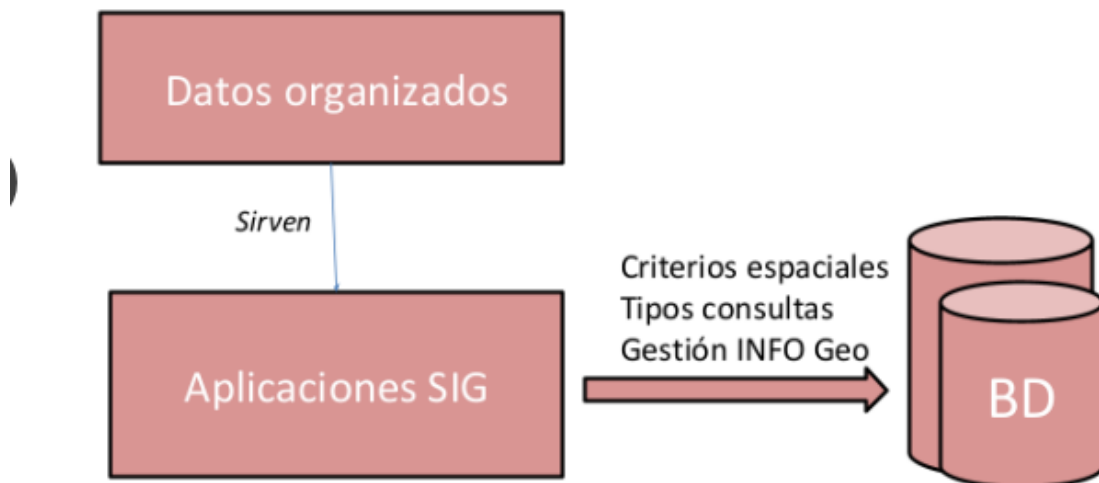
Tres tipos de datos más comunes son:

- ➔ Dato temporal: Es el más importante, y se pueden definir como una unión ifnita de intervalos. El dato temporal se considera la base sobre la que se construye la base d edatos temporal.
- ➔ Dato estático: Se considera como una constante sobre todo el univeros del tiempo. Se suele identificar este dato como eterno, en contraste con el tipo de ddato temporal es válido apra un periodo o intervalo específico de tiempo.
- ➔ Dato instantáneo: dato solo vigente en un momento o instante actual

BD Geográfica

Aguardaremos unos datos mediante datos geográficos. Pueden ser de cualquier tipo. Son una colección de datos que sirven para aplicaciones de sistemas de información geográficas. Permiten el almacenamiento estructurada de los datos de acuerdo a **criterios espaciales**, tipos de consultas y **gestión de información geográfica**.

## Base Datos **Geográficas** (BDG)



Quizá en una cinta de montaje no lo necesita toda la maquinaria pero quizá uno si que necesita datos geográficos. AGV → Robot que necesita una serie de datos geográficos. Puede ser un SIG (sistema de navegación geográfica) o bien puede ser global, e tc.

La lectura, la forma de acceder a estos datos de manera geográfica.





Interoperabilidad → Habilidad de dos o más sistemas o componentes para intercambiar información y utilizar información intercambiada.

Normalización → Se entiende como la utilización de forma adecuada de los procedimientos, datos y servicios.

El Instituto Geográfico Nacional, ha definido su Catálogo de Objetos Geográficos. Se puede definir como aspectos del mundo real relacionados con una ubicación concreta en el terreno. Se caracterizan por tener atributos que pueden ser representados mediante una geometría primitiva, tales como:

- ➔ Puntos: Se encuentran determinados por las coordenadas terrestres medidas por la latitud y longitud.
- ➔ Líneas: Objetos abiertos que cubren una distancia dada y comunican varios puntos o nodos.
- ➔ Polígonos: Figuras planas conectadas por distintas líneas.
- ➔ Celdas de un ráster: La rasterización, permite a través de una malla asociar datos a una imagen, es decir, se pueden relacionar paquetes de información con los píxeles de una imagen digitalizada.

En teoría tendremos un usuario que accede a esos datos de manera geográfica.

Los SIG modelan la realidad territorial para convertirla en datos geográficos que son manipulados en un entorno informatizado. Utilizan los modelos de representación raster y vectorial.

- ➔ El modelo ráster
- ➔ El modelo de representación vectorial

Las bases de datos distribuidas, tenemos unos datos en el sistema que se ejecutan en el aplicativo global pero se puede hacer una consulta de manera local. A su vez está



repartido entre ordenadores de una misma red, ya sea a nivel local, o cada uno en una diferente localización geográfica. A un sistema que da información a dispositivos locales.

- ➔ En las criptos, cada uno tiene un aplicativo que revisa que existan. No hay un sitio global. El blockchain es todo lo contrario a base de datos distribuidas. +

Es un híbrido entre la centralizada y la descentralizada.

### Base de Datos distribuida

Centralized: Una base de datos central a la que tienen acceso los diferentes usuarios.

Distributed: Tendrá un aplicativo global. Tendrán particiones. Tendremos centralizaciones.

Descentralized : Sería algo parecido al blockchain. No hay un sistema central que tenga ese control.

//Pendiente de repasar PDF.

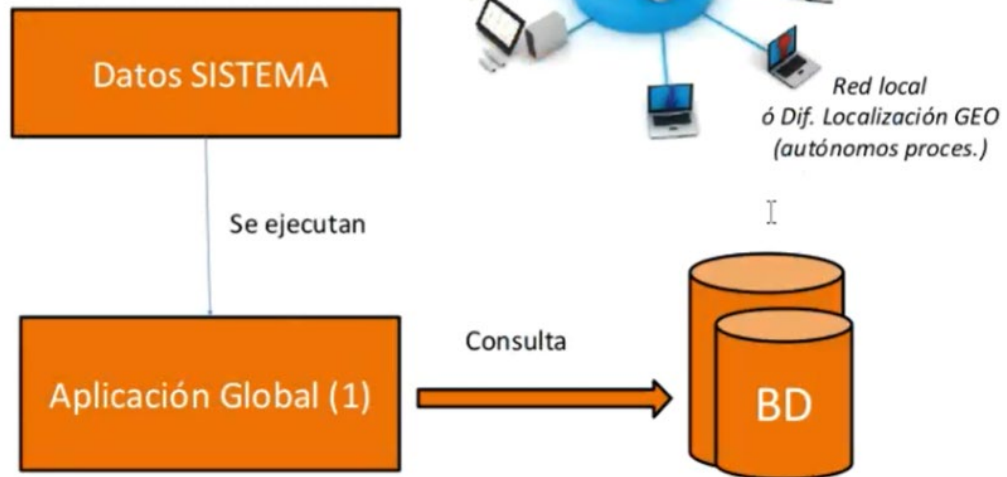
Autonomía: Se pueden presentar en diferentes niveles, como son:

- ➔ Autonomía de diseño: Está relacionada con el mismo diseño
- ➔ Autonomía de comunicación: Cuando se comunica con otros sistemas manejadores de bases de datos (SMBD). Ellos pueden comunicarse autónomamente con esa base de datos.
- ➔ Autonomía de ejecución. Ejecuta operaciones locales como quiera.
- ➔ Autonomía de ejecución: Ejecuta operaciones locales como quiera.

Se debe definir un modelo de referencia para un esquema de estandarización en las bases de datos, cuyo propósito es dividir el trabajo en piezas y esas piezas se relacionen unas con otras:

- ➔ BD Distribuidas

## BD Distribuidas



Que servicios vamos a ofrecer?. Eso es importante tenerlo en cuenta. Que datos vamos a ver en esa base de datos. Podemos tener diferentes tipologías de base de datos dentro de una arquitectura.

### BD Analíticas (OLAP=.

→ ROLAP

→ OLAP

Son bases de datos que almacenan datos e información que se extraen de bases de datos operacionales y externas seleccionadas. Se componen de los datos y la información resumida que más necesitan los usuarios finales.

No tiene porque ser locales. Las bases de datos analíticas también el nombre de bases de datos gerenciales o bases de datos de información. Es el acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Processing).

Se componen de la información resumida que más necesitan los usuarios finales.

## BD Analíticas (OLAP)

ROLAP, MOLAP y HOLAP



En la operacional podríamos tener una serie de datos. Y en las externas una serie de información que se extrae de esa base de datos operacional. La razón es la rapidez de su respuesta. Una base de datos relacional almacena entidades de tabla (falta penúltimo párrafo de la página 109).

- ➔ ROLAP: Implementación OLAP que almacena los datos en un motor relacional. Los datos son detallados, evitando las agregaciones y las tablas se encuentran **desnormalizadas**. Sacamos un dato concreto.
  - Tendríamos diferentes estructuras de diseño. Serían bases de datos relacionales.
- ➔ MOLAP: Podríamos obtener datos en una base de datos multidimensional. Pueden tener más de una información agregada en esa información. Almacena los datos en una base de datos multidimensional.
  - No sigue la primera forma normal porque tenemos más de un valor. Pero tenemos más de un valor.
- ➔ HOLAP. Almacena algunos datos en un motor relacional y otros en una base de datos multidimensional. Es Híbrido.

BD de Columnas: Son bases de datos orientadas a columnas, es decir, tienen sus datos organizados y almacenados por columnas separadas, debido a esto, el sistema puede evaluar las columnas que están siendo accedidas en una consulta y recuperar sólo los valores solicitados desde las columnas específicas.

- ➔ Están definidos por columnas y no por tablos.
- ➔ Ventajas:

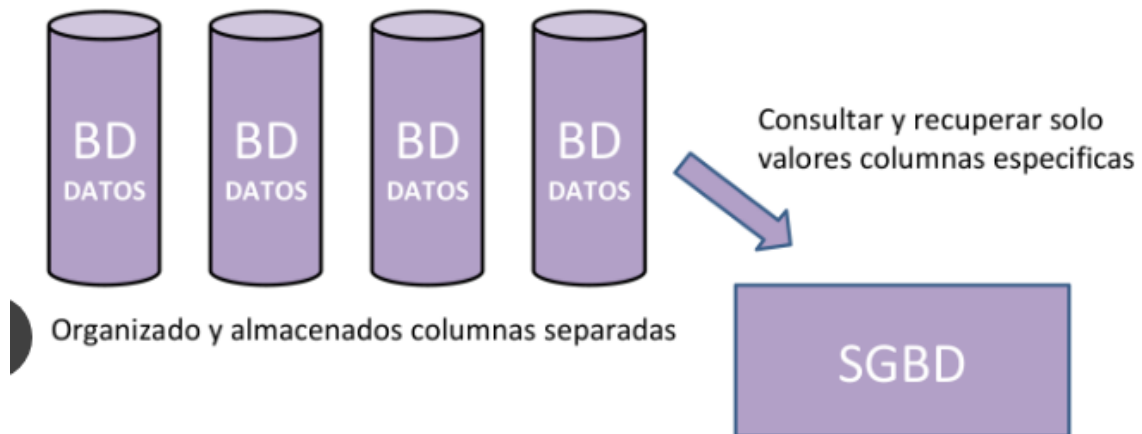
- Acceso rápido a los datos, como el modelo DSM, el cual nos permite consultar rápidamente los datos columna a columna, al guardarse físicamente de manera contigua.
- Lee solo los valores de las columnas necesarias para el procesamiento de una consulta determinada, por lo que tiene una mayor eficiencia en entornos de almacenes, donde las consultas, típicas incluyen los agregados realizados por un gran número de elementos de datos.
- Comprime la información asignable de cada columna con el fin de mejorar el procesamiento desde el ancho de banda del acceso a disco.
- Los cambios en el esquema tienen menor impacto y por lo tanto el costo de realizarlos es menor.

➔ Desventajas //

➔ La función es sobretodo para el acceso a esos campos de manera más eficiente posible.

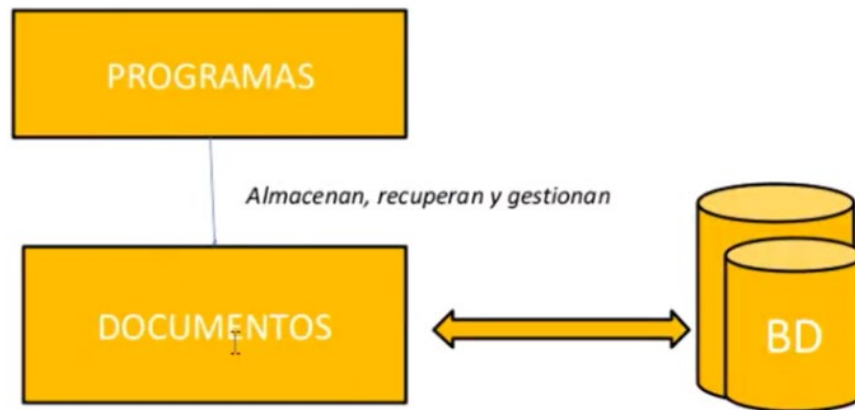
➔ Se considera un modelo avanzado de base de datos.

## BD de Columnas



### Bases de datos documentales

## BD de Documentales (Bases de datos NO SQL)



Filezilla almacena, recupera y gestiona una serie de documentos de una base de datos. La URL de Firebase (por ejemplo) tiene una URL, a una base de datos con una serie de archivos. Tenemos una serie de archivos en esa base de datos. Podemos tener una base de datos que guarde un archivo concreto. Se puede ir combinando.

A diferencia de las bases de datos relacionales, estas bases de datos están diseñadas alrededor de una noción abstracta de documento. Mientras cada implementación de base de datos orientada a documentos difiere en los detalles, en general todas ellas comparten el principio de que los documentos encapsulan y codifican datos.

Más de la sencilla correspondencia clave-documento o clave-valor usada para recuperar un documento. Ofrece un API o un lenguaje de interrogación para recuperar documentos según su contenido. Así poder, por ejemplo, preguntarle por todos los documentos que tienen un valor dado en un campo. El conjunto de características del API o del lenguaje de interrogación, así como lo que se obtiene, varía significativamente entre distintas implementaciones. Se utiliza para:

- ➔ Colecciones
- ➔ Etiquetas
- ➔ Metadatos ocultos.
- ➔ Jerarquía de los contenidos.

Se puede llegar a combinar. Podríamos tener Netflix con una base de datos documental, dentro de la cual tuviéramos una serie de particiones distribuidas. Entre las que los usuarios pueden acceder.

WEBANGULAR HTML ➔ Es un framework, para aplicaciones web. Es un sistema que te ayuda a crear aplicativos. Desarrollado por Typscript.

BD XML: Si os suena, XML. Esto es similar a copiar la bases de datos. Sirve para almacenar y recuperar datos poco estructurados. No tienen relación entre ellos a no ser que tengan estructuras anidadas en concreto.

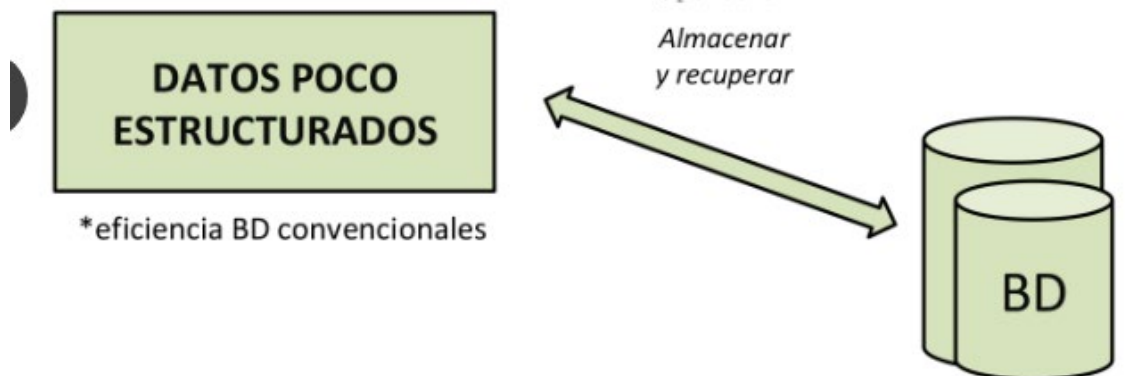
- Title tiene algo que ver con link? En HTML, en verdad no. Porque están en el mismo canal, pero entre ellos no tienen relación, realmente.
- Centrados en datos
- Centrados en documentos.

Hacemos una exportación de una base de datos como una fotografía, como datos anidados.

El XML es el documento que guarda esa bases de datos. Las otras tienen otras extensiones, depende del sistema. No es un archivo en concreto.

8 TALLER DE TRANSFORMACIÓN Y PATRONES

## BD XML



### BD Incrustadas (embedded)

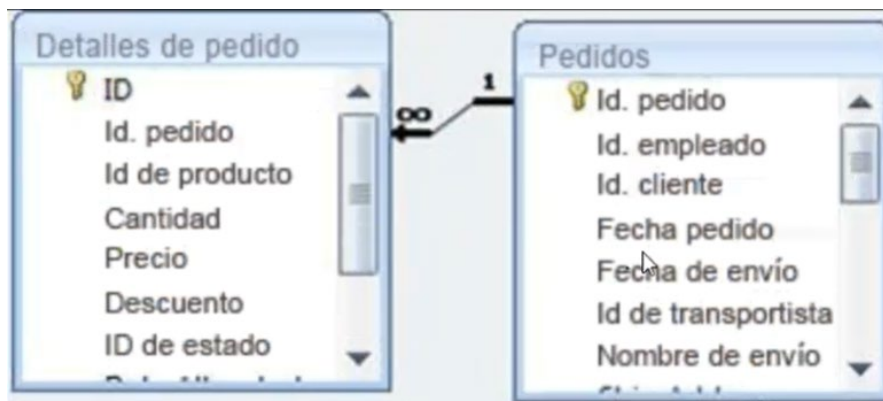
No inicia un servicio en el equipo

Código fuente – Librería

La base de datos está incrustado en un servicio. El próximo sistema gestor de la base de datos está creando esos archivos. Con su propio lenguaje está creando esa base de datos. Nosotros mediante un fichero generaremos la base de datos. Nosotros veremos esa relación mediante el propio sistema. Tendremos un Excel que en si, no será una base de datos, podemos importar esa base de datos que incrustalo dentro del sistema gestor de base de datos. En ese momento pasará a ser una base de datos.

El próximo sistema gestor de base de datos. De esta tabla y esta tabla voy a generar esa base de datos. Podremos modificar la base de datos, desde el propio sistema. Podemos acceder a esa relación y añadir más registros extras.

## BD Incrustadas (embedded) No inicia un servicio en el equipo



1. ¿Cuáles son los tipos de factores que utilizan las bases de datos deductivas?

b) Los hechos, las reglas y el sistema de inferencia.

\*Teníamos una serie de reglas que nos darán hechos. Nos darán una serie de datos. Son deductivos.

2.

d) El tiempo de validez y el tiempo de transición

3.

c) El dato temporal (el más importante), el dato estático y el dato instantáneo.

4.

b) Los puntos, las líneas, los polígonos y las celdas de un raster (esas representaciones primitivas a la hora de guardar una base de datos geográfica).

5.

c) En tres niveles

\*Vimos tres niveles:

El nivel de autonomía, etc. Lo vemos en tres niveles.

6.

b. ROLAP, MOLAP y HOLAP

7. ¿De qué está compuesta las filas de BD de columnas?

c) Las filas de una columna son las tuplas. Serían los registros.

8.

c) Colecciones, Etiquetas, Metadatos ocultos y Jerarquías de directorios.

9.

d)

\*Películas quizá si que podría estar dentro de documentos.

10.

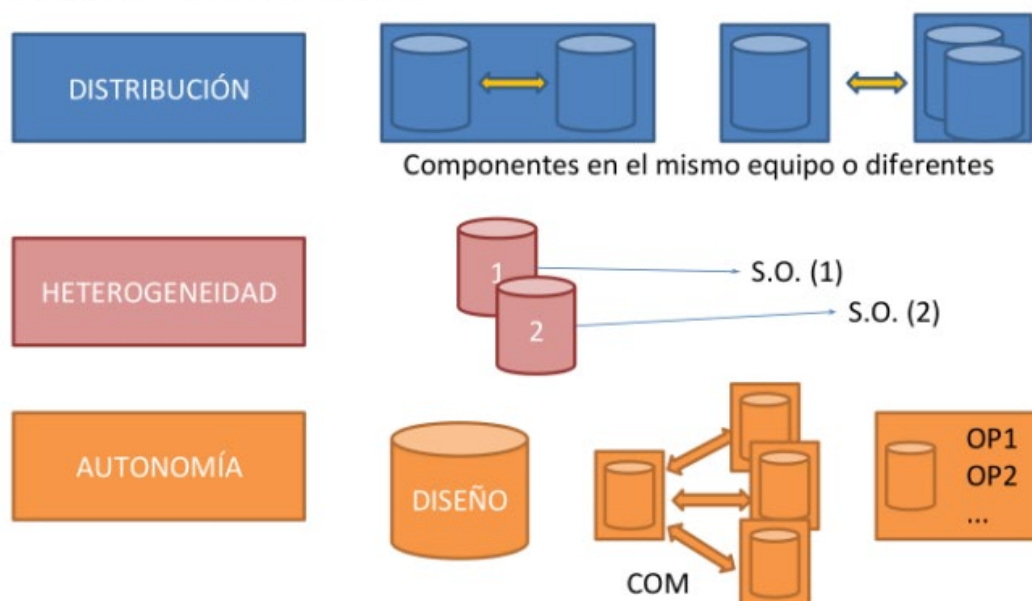
a)

Página 116. La Scimore DB, no es la más importante hoy en día dentro. Si buscas los tipos de base de datos incrustadas, casi que no sale ScimoreDB.

### **T.15.5 Análisis detallado de la distribución de la BD**

#### **5.1 Formas de distribución**

FACTORES (distribuir un sistema)





Distribución: Los componentes del sistema están localizados en un mismo equipo o en diferentes equipos.

Heterogeneidad: Cuando existen componentes que se ejecutan en diversos sistemas operativos.

Autonomía: Se puede presentar en diferentes niveles, como son:

- ➔ Autonomía de diseño: Está relacionada a su propio diseño.
- ➔ Autonomía de comunicación: Cuando se comunica con otros sistemas manejadores de bases de datos (SMBD).
- ➔ Autonomía de ejecución: Ejecuta operaciones locales como quiera.

Modelo de referencia para un esquema de estandarización en las bases de datos:  
propósito: dividir el trabajo en piezas y esas piezas se relacionan unas con otras.

- ➔ Basado en componentes: Se definen las componentes del sistema junto con las relaciones entre ellas.
- ➔ Basado en funciones: Se identifican las diferentes clases de usuarios junto con la funcionalidad que el sistema ofrecerá para cada clase.
- ➔ Basado en datos: Se identifica los diferentes tipos de descripción de datos y se especifica un marco de trabajo arquitectural el cual define las unidades funcionales que realizarán y usarán los datos de acuerdo con las diferentes vistas.

\*ANSI/SPARC.

### La transparencia

Es la división del nivel semántico y la implementación del sistema.

- ➔ Ocultar al usuario los detalles del diseño. No tiene que saber que se encuentra trabajando sobre un sistema distribuido, por ejemplo, la independencia de los datos es una forma de transparencia.
- ➔ El **propósito** de la transparencia en el ambiente distribuido es la independencia de datos:
  - Manejo de la red de comunicación
  - Manejo de copias repetidas
  - En la distribución o fragmentación de la información

La independencia de datos: inmunidad de las aplicaciones de usuarios a los cambios en la definición y organización de los datos y viceversa.

- ➔ **Independencia lógica** de datos: inmunidad de las aplicaciones de usuario a los cambios en la estructura lógica de la base de datos, permite un cambio en la definición de un esquema no debe afectar a las aplicaciones de usuario. Cuando cambie la estructura, no puede afectar a las aplicaciones.
  - Agregar un nuevo atributo a una relación, creación de una nueva relación, reordenamiento lógico de algunos atributos.
- ➔ **Independencia física de datos:** ocultamiento de los detalles sobre las estructuras de almacenamiento a las aplicaciones de usuario, es decir, **la descripción física de datos puede cambiar sin afectar a las aplicaciones de usuario.**
  - Los datos pueden ser movidos de un disco a otro, o la organización de los datos puede cambiar.

Hay niveles de transparencia: con el propósito de establecer una arquitectura de un sistema de bases de datos distribuidas es ofrecer un nivel de transparencia adecuado para el manejo de la información.

- ➔ **Primer nivel:** se soporta la transparencia de red
- ➔ **Segundo nivel:** se permite la transparencia de replicación de datos.
- ➔ **Tercer nivel:** se permite la transparencia de la fragmentación
- ➔ **Cuarto nivel:** se permite transparencia de acceso, por medio de un lenguaje de manipulación de datos.

Que son la transparencia de localización de fragmentación, de replicación de datos y de localización.

- ➔ La **transparencia de localización:** permite acceder al usuario a los datos sin tener en cuenta la ubicación de estos. Debe ser transparente al usuario ya que no necesita saber donde está el dato para utilizarlo. Se consigue cuando los administradores de transacciones distribuidas pueden determinar la localización de los datos y emitir acciones a los administradores apropiados.
  - Se puede ejecutar cuando los administradores de transacciones tienen acceso a los directorios de localizaciones de datos. Los administradores de transacciones necesitan conocer si los datos cambian de lugar, ya que las transacciones ignoran la modificación en la localización.
- ➔ La **transparencia de fragmentación:** El acceso a una base de datos distribuida debe hacerse de forma transparente, los usuarios deben comportarse como si los datos en realidad no estuvieran fragmentados, lo cual es necesario por razones de rendimiento.
  - El sistema maneja la conversión de consultas de usuario definidas sobre relaciones globales a consultas definidas sobre fragmentos. Las respuestas a las consultas fragmentadas para obtener una sola respuesta a una consulta global.
- ➔ La **transparencia de réplica:** si existen réplicas de objetos de la base de datos, esta debe ser controlada por el sistema manejador de base de datos (SMBD).
  - La función principal de la transparencia de replicación es la de mantener la consistencia entre las copias, esta funciona de forma transparente a las aplicaciones.
  - La replicación puede mejorar el funcionamiento y proteger la disponibilidad de las aplicaciones, porque alternas opciones de acceso de los datos existentes.
  - Existe una **copia principal** y varias **copias secundarias**, las que se extienden a lo largo de las modificaciones en forma asíncrona.
  - Hay dos tipos de propagación de modificaciones:
    - Incremental: la información que se envía desde la copia principal a las secundarias son las variaciones en los datos.
    - Total: Se envía toda la copia principal completa

Utilidad: hacer que el sistema sea menos sensible a los fallos. Ya que si la copia principal no está disponible, se puede seguir usando alguna de las copias secundarias. La replicación es necesaria por las siguientes razones.

- Mayor rendimiento: Debido a que se dispone de **copias locales**.
- Mayor disponibilidad: Debido a que los datos son accesibles siempre al tenerse varias copias.

Principal desventaja: hay que mantener actualizadas todas las copias de ese objeto o dato replicado. Esto nos lleva al problema de la propagación de las actualizaciones.

## 5.2 Arquitectura ANSI/X3/SPARC (130-145)

### 5.3 Transacciones distribuidas

### 5.4 Mecanismos de distribución de datos

#### **Mecanismos de distribución de datos transacciones:**

1. Distribución de datos
2. Durabilidad
3. Características BD
4. Duración
5. Estructura

#### Autoevaluación

- 1.
- b) La distribución, la heterogeneidad y la autonomía
- 2.
- d) La independencia lógica de datos y la independencia física de datos.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.

## T.16 Introducción SQL

\*En si, no es un lenguaje de programación. No tiene funciones, por ejemplo.

El SQL es el Structured Query Language. Lenguaje estructura de consultas. Inventado por IBM (SEQUEL).

- ➔ No solo consultas, no confundirse con el nombre, si no conoces lenguajae peudes pensar que solo se realizan consultas pero SQL sirve para crear bd de cero, crear campos nuevos, eliminarlos, modificar propiedades campos, establecer relación entre tablas de una base de datos, etc.

\*Cual es la consulta. Lo que sea. Nosotros haremos una acción concreta, en esa base de datos. Que tablas de esa base de datos estamos modificando. Que campos vamos a mencionar.

- ➔ Que tablas estamos modificando.
- ➔ Que campos vamos a mencionar
- ➔ Cual va a ser la relación de la bases de datos.

\***Vamos a pedir:** “dime un dato concreto”. Puede ser un campo, puede ser una consulta.

\*No solo es un lenguaje de consultas.

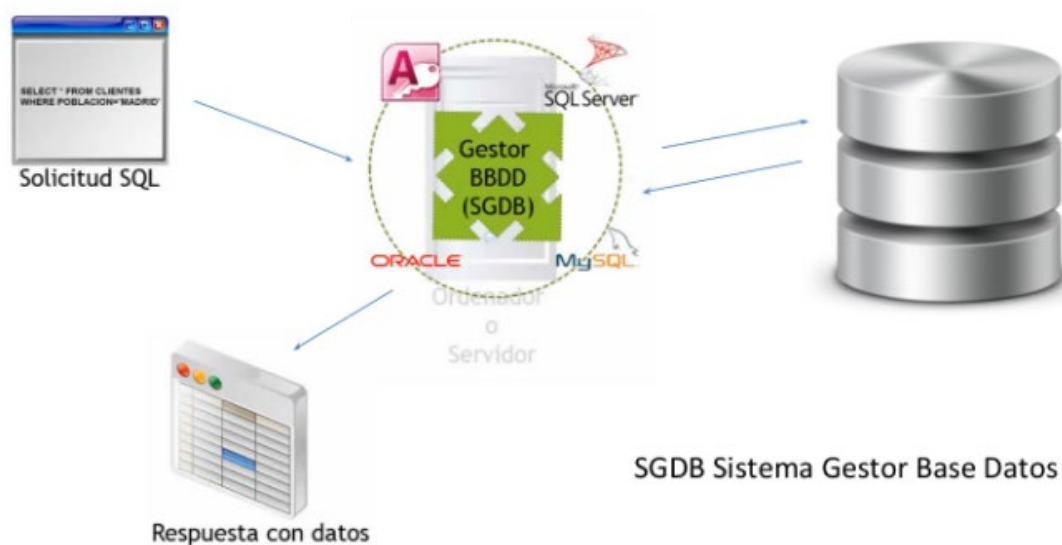
Más tablas, que se conectan con más tablas. Si yo tengo una tabla de clientes, con ventas. Puedo decirle las que ha tenido, y los datos de esa venta. SQL nos sirve para prácticamente todo en una base de datos.

\*La mayoría de **acciones** se llaman acciones **CRUD**

- ➔ Create
- ➔ Read
- ➔ Update (into) o insert
- ➔ Delete

\*Aunque se llame lenguaje estructurado de consulta, no solo hace consultas. Hace más casos.

Esquema básico de funcionamiento de este lenguaje de acceso y manipulación de datos en base de datos relacionales



Nosotros tendremos una solicitud concreto de SQL:

\*Select determinada tabla y mostrarme una serie de clientes.

En el propio SGDB haremos esa consulta. Puede ser como:

- ➔ Acces: En el mismo paquete office.
- ➔ AQL Server
- ➔ Oracle
- ➔ My SQL: Lo tenemos instalado con XAMPP.

\*Heide SQL, Terminal.

Tendremos una serie de datos, eso lo importamos al sistema gestor de base de datos.

## Estándar SQL



**ANSI SQL = ANSI + ISO**

Ese estándar se lo enviaremos a un Sistema Gestor de Base de Datos. Enviaremos un universal como es el caso del SQL.

Acces, normalmente entraran.

1. Select [E-mail Address], Company,
2. From Contacts
3. Where City = Seattle

Los cambios de un programa a otro pueden ser muy pequeños como es el caso de los corchetes. Tenemos diferentes formas (dialectos) de escribir el lenguaje estandarizado.

ANSI (Instituto de Estandarización). Establece una nomenclatura ANSI y una nomenclatura ISO para este tipo de lenguaje.

Dialecto como tal.

## Grupos de comandos SQL

- ➔ DDL: Lenguajes de definición de datos:
  - Crear y modificar la estructura de la base de datos BD.
  - Crear, eliminar, tablas...modificar estructura tablas agregando eliminados campos, etc.  
\*Crearía la estructura sobre la que se articula la base de datos. Puede eliminar los campos o los registros.
- ➔ DML: Lenguaje de manipulación de datos:
  - Seleccionar registros de una BD, consultas, insertar nuevos registros, actualizar, borrar información.
  - Consultas de selección y acción.
- ➔ DCL: Lenguajes de control de datos:
  - Proporcionan seguridad a la información en la base de datos.
- ➔ TCL: Lenguajes de control de transacciones:
  - Gestión de los cambios en los datos.

DDL: Create, Truncate, Drop, Alter

DML: Select, Delete, Update, Insert

DCL: Grant, Revoke

TCL: Commit, Rollback, Savepoint

## Cláusulas SQL:

- ➔ **From:** especifica la table de la que se quieren obtener los registros
- ➔ **Where:** especifica las condiciones o criterios de los registros seleccionados
- ➔ **Group By:** para agrupar los registros seleccionados en función de un campo
- ➔ **Having:** especifica las condiciones o criterios que deben cumplir los grupos
- ➔ **Order by:** ordena los registros seleccionados en función de un campo

Instrucción SQL:

## **COMANDO\* + CLÁUSULAS\* + OPERADORES + FUNCIONES**

Construimos mediante estos componentes una frase en inglés una instrucción SQL

Operadores SQL

➔ Comparación:

<	menor que
>	mayor que
=	Igual
>=	Mayor o igual que
<=	Menor o igual que
<>	Distinto
BETWEEN	Entre. Utilizado para especificar rango de valores
LIKE	Cómo. Utilizado con caracteres comodín
In	En. Para especificar registros en un campo en concreto

- ➔ Lógicos: (el and multiplica y el or solo esta sumando)
- And ----- Y lógico
  - Or ----- O lógico
  - Not ----- Negación lógica

Orden de escritura

1. Comando\*
2. Cláusulas\*

From, Where, Group By, Having, Order By.

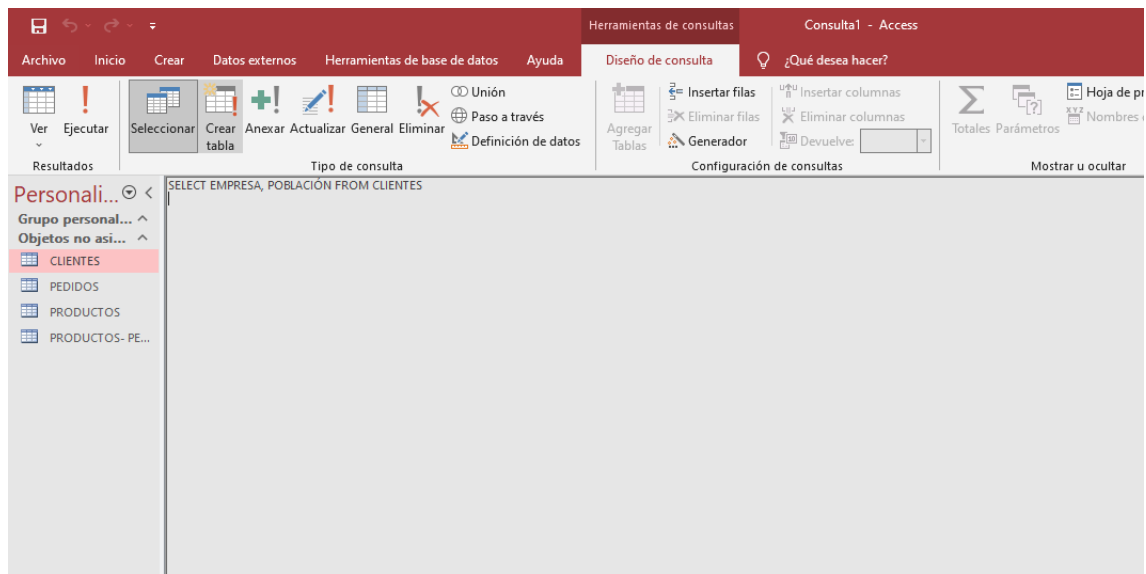
Ejercicios SQL

**Ejercicio 1: Realizar una consulta que muestre los campos “Empresa” y “Población” de la talba “Clientes”.**

SELECT EMPRESA, POBLACIÓN FROM CLIENTES

Procedimiento Acces:

En diseño de consulta ➔ Se crea una nueva opción

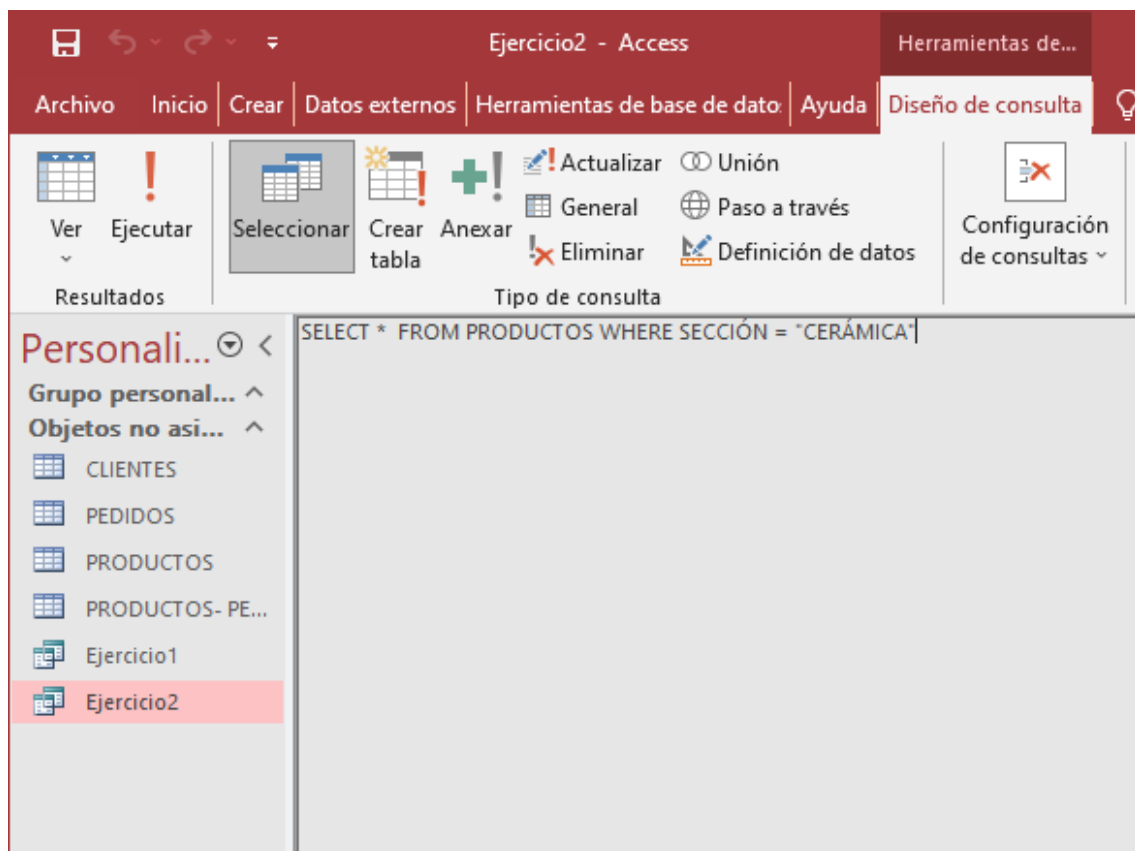


**Ejercicio 2: Realizar una consulta que muestre los artículos de la sección “Cerámica”.**

**SELECT \* FROM PRODUCTOS WHERE SECCIÓN = "CERÁMICA"**

\*No especifica que sea el nombre del artículo.

\*Para que muestre todo. Ponemos un \* y lo mostrará todo.



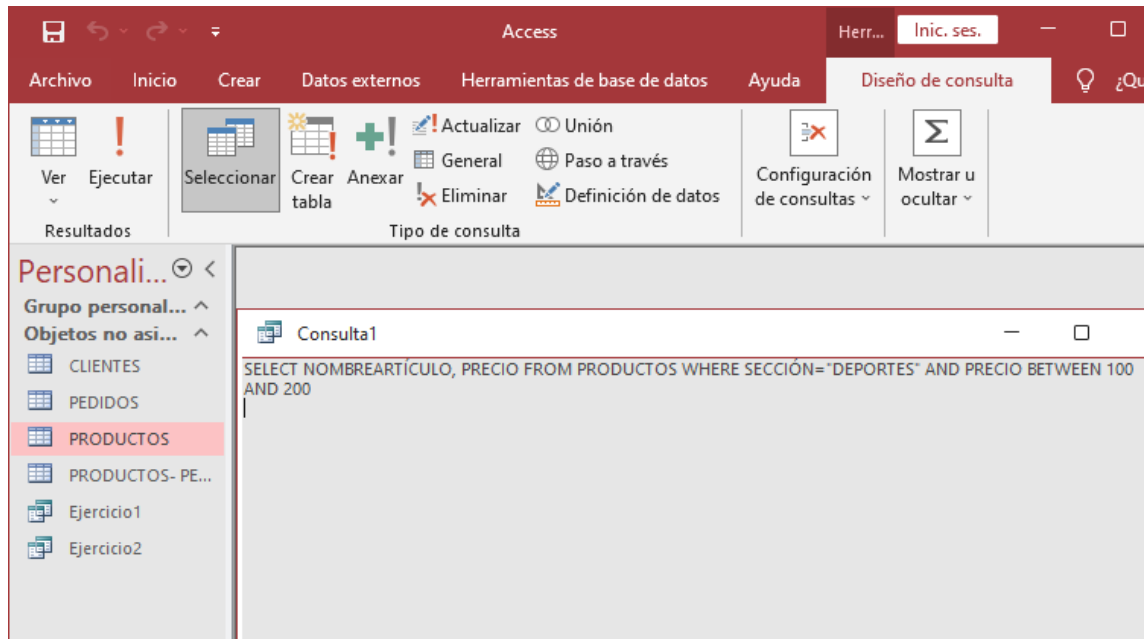
**Ejercicio 3. Realizar una consulta que muestre los productos que muestre los productos de la sección “Deportes” cuyo precio esté entre 100 y 200€. En la consulta solo se mostrarán los campos “Nombre de artículo” y “Precio”.**

```
SELECT NOMBREARTÍCULO, PRECIO FROM PRODUCTOS WHERE SECCIÓN="DEPORTES" AND PRECIO BETWEEN 100 AND 200
```

\*También podría ser `PRECIO >= 100 AND PRECIO <=200`

→ Con el BETWEEN, nos estamos ahorrando un PRECIO aquí.

**\*Se puede utilizar and con más de un operador.**

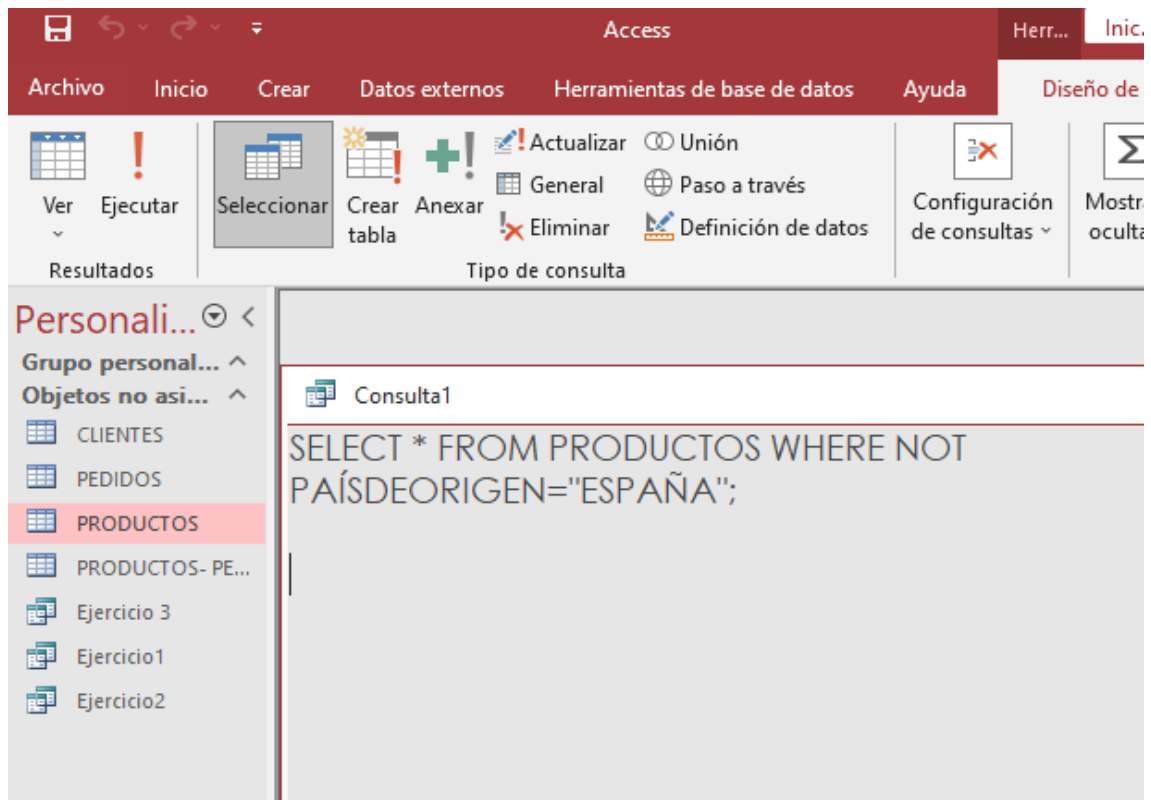


**Ejercicio 4. Realizar una consulta que muestre los productos cuyo país no sea España.**

Crear → Diseño de consulta →

```
SELECT * FROM PRODUCTOS WHERE NOT PAÍSDEORIGEN="ESPAÑA";
```





**Ejercicio 5. Realizar una consulta que muestre los artículos españoles de la sección “Deportes” o aquellos cuyo precio sea superior a 350€ independientemente de cual sea su sección o país de origen.**

- a) `SELECT * FROM PRODUCTS WHERE PAÍSDEORIGEN="ESPAÑA" AND SECCIÓN="DEPORTES" OR PRECIO<350 AND`

\*Cuidado con la conjunción i/o.

El or está englobando desde where país de origen hasta deportes. Porque lo primero que se hace es el producto/multiplicación que es paísdeorigen y sección=deportes.

0+0→0

0+1→1

1+0→1

1+1\_1

(1X1)+1

Importación de la base de datos en **PHPMyAdmin**

**En SQL el varchar es el tipo de dato.**

1. **Crea la base de datos**
2. **Inserta las diferentes tablas**
3. **Inserta las diferentes filas y tuplas aquí.** Las comas que utiliza para los valores son els accents oberts en català.

**a. La primera no la coge porque le hemos dicho que era el nombre del campo.**

\*Con el caso de los clientes hemos creado una base de datos y le hemos puesto nosotros el nombre. Luego hemos importado la base de datos y hemos vuelto a seleccionar la opción de que son los nombres del campo.

	A	B	C	D	E	F	G	H	I	J	K	L
1	CÓDIGOART	SECCIÓN	NOMBREART	PRECIO	FECHA	IMPORTADO	PAÍSDEORIGI	FOTO				
2	AR01	FERRETERÍA	DESTORNILL	6,63	2000-10-22	FALSO	ESPAÑA					
3	AR02	CONFECCIÓN	TRAJE CABAL	284,58	2002-03-11	#####	ITALIA					
4	AR03	JUGUETERÍA	COCHE TELE	159,45	2002-05-26	#####	MARRUECOS					
5	AR04	DEPORTES	RAQUETA TE	93,47	2000-03-20	#####	USA					
6	AR06	DEPORTES	MANCUERNA	60,00	2000-09-13	#####	USA					
7	AR07	CONFECCIÓN	SERRUCHO	30,20	2001-03-23	#####	FRANCIA					
8	AR08	JUGUETERÍA	CORREPASILI	103,34	2000-04-11	#####	JAPÓN					
9	AR09	CONFECCIÓN	PANTALÓN S	174,23	2000-01-10	#####	MARRUECOS					
10	AR10	JUGUETERÍA	CONSOLA VI	442,54	2002-09-24	#####	USA					
11	AR11	CERÁMICA	TUBOS	168,43	2000-02-04	#####	CHINA					
12	AR12	FERRETERÍA	LLAVE INGLE	24,40	2001-05-23	#####	USA					
13	AR13	CONFECCIÓN	CAMISA CAB	67,13	2002-08-11	FALSO	ESPAÑA					
14	AR14	JUGUETERÍA	TREN ELÉCTR	1.505,38	2001-07-03	#####	JAPÓN					
15	AR15	CERÁMICA	PLATO DECO	54,09	2000-06-07	#####	CHINA					

Donde pone general podemos cambiar y poner más formatos de número, vamos a fecha y allí podemos escoger el tipo de fecha que queremos.

Todos estos parámetros se guardan en el diccionario de datos. A su vez están en las diferentes relaciones entre los objetos.

\*Le interesa saber el tipo de archivo; o INT o DOBLE O VARCHAR.

### Instrucciones

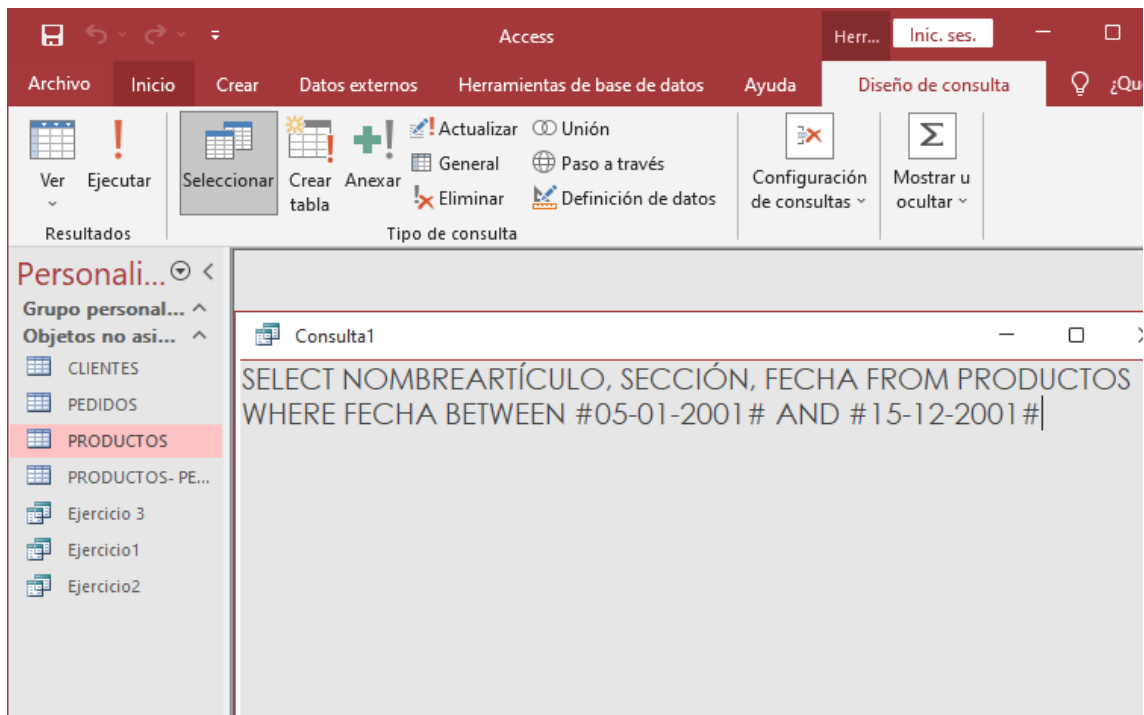
\*La **instrucción**

\*La **instrucción** que va a dar para destruir una base de datos: DROP DATABASE `ods\_db`;

**Ejercicio 6. Realizar una consulta que muestre los productos cuya fecha esté entre 1/05/2001 y 15/12/2001. En la consulta solo se visualizarán os campos “Nombre de artículo”, “Sección” y “Fecha”.**

Para Acces es:

```
SELECT NOMBREARTÍCULO, SECCIÓN, FECHA FROM PRODUCTOS WHERE FECHA BETWEEN #05-01-2001# AND #15-12-2001#
```



Para PHP My SQL es:

SELECT NOMBREARTÍCULO, SECCIÓN, FECHA FROM PRODUCTOS WHERE  
FECHA BETWEEN '2001-05-01' AND '2001-12-15'

Mostrar ventana de consultas SQL

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

✓ Mostrando filas 0 - 7 (total de 8. La consulta tardó 0,0006 segundos.)

SELECT NOMBREARTÍCULO, SECCIÓN, FECHA FROM PRODUCTOS WHERE FECHA BETWEEN '2001-05-01' AND '2001-12-15';

☐ Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

NOMBREARTÍCULO	SECCIÓN	FECHA
LLAVE INGLESA	FERRETERÍA	2001-05-23
TREN ELÉCTRICO	JUGUETERÍA	2001-07-03
CENICERO	CERÁMICA	2001-07-02
MARTILLO	FERRETERÍA	2001-09-04
CAZADORA PIEL	CONFECCIÓN	2001-07-10
BALÓN BALONCESTO	DEPORTES	2001-06-25
ABRIGO SRA	CONFECCIÓN	2001-05-03
PIE DE LÁMPARA	OFICINA	2001-05-27

En el caso de PHP my admin, se trata de un software diferente, y por lo tanto estamos hablando de un dialecto diferente. Es la manera como interpretan el lenguaje SQL.

En cifras las fechas es una comparativa. Podríamos tener  $\geq 100$  PASADO  $\leq 150$ . \*Para un mismo dialecto nosotros podemos tener diferentes maneras o diferentes vías de encontrar el resultado que ns está proponiendo.

### Cláusula ORDER BY

Ordena los registros seleccionados en función de un campo

Si queremos realizar una ordenación en nuestros registros, ejemplo:

SELECT \* FROM PRODUCTOS WHERE SECCIÓN='DEPORTES' OR SECCIÓN='CERÁMICA' + **ORDER BY SECCIÓN**

El texto por defecto sale de A a la Z, números de menos a mayor

➔ DESC: *invierte el orden por defecto*

SELECT \* FROM PRODUCTOS WHERE SECCIÓN='DEPORTES' OR SECCIÓN='CERÁMICA' ORDER BY SECCIÓN, PRECIO

No se puede ordenar por **dos campos** a la vez pero si por varios criterios de ordenación, primero uno, después los siguientes.

\*El desc se debe poner en cada uno de los campos, solo afecta a uno. Por ejemplo si queremos atribuirlo a sección, ponemos sección desc, precio.

➔ SELECT \* FROM PRODUCTOS WHERE SECCIÓN='DEPORTES' OR SECCIÓN='CERÁMICA' ORDER BY SECCIÓN, PRECIO DESC

CÓDIGOARTÍCULO	SECCIÓN ▾ 1	NOMBREARTÍCULO	PRECIO ▲ 2	FECHA	IMPORTADO	PAÍSDEORIGEN	FOTO
AR41	DEPORTES	PALAS DE PING PONG	21.6000	2002-02-02	FALSO	ESPAÑA	NULL
AR28	DEPORTES	BALÓN FÚTBOL	43.9147	2002-07-04	FALSO	ESPAÑA	NULL
AR18	DEPORTES	PISTOLA OLÍMPICA	46.7347	2001-02-02	VERDADERO	SUECIA	NULL
AR06	DEPORTES	MANCUERNAS	60.0000	2000-09-13	VERDADERO	USA	NULL
AR25	DEPORTES	BALÓN BALONCESTO	75.2731	2001-06-25	VERDADERO	JAPÓN	NULL
AR04	DEPORTES	RAQUETA TENIS	93.4694	2000-03-20	VERDADERO	USA	NULL
AR24	DEPORTES	BALÓN RUGBY	111.6440	2000-11-11	VERDADERO	USA	NULL
AR40	DEPORTES	BOTA ALPINISMO	144.0000	2002-05-05	FALSO	ESPAÑA	NULL
AR38	DEPORTES	CAÑA DE PESCA	270.0000	2000-02-14	VERDADERO	USA	NULL
AR32	DEPORTES	CRONÓMETRO	439.1764	2002-01-03	VERDADERO	USA	NULL
AR21	CERÁMICA	CENICERO	19.7468	2001-07-02	VERDADERO	JAPÓN	NULL
AR33	CERÁMICA	MACETA	29.0434	2000-02-23	FALSO	ESPAÑA	NULL
AR20	CERÁMICA	JUEGO DE TE	43.2728	2001-01-15	VERDADERO	CHINA	NULL
AR15	CERÁMICA	PLATO DECORATIVO	54.0911	2000-06-07	VERDADERO	CHINA	NULL
AR39	CERÁMICA	JARRA CHINA	127.7704	2002-09-02	VERDADERO	CHINA	NULL
AR11	CERÁMICA	TUBOS	168.4253	2000-02-04	VERDADERO	CHINA	NULL

//Quedaría guardar en Accés, l'exercici 4-5-6.