


DISEÑO DE BASES DE DATOS RELACIONALES

Data BootCamp
Cámara Comercio Bilbao

Germán Alonso Lascurain
germanalonso@opendeusto.es

CÁMARABILBAO 

 C2B
campus to business

 ai
ai power

 tigloo

TABLA DE CONTENIDOS

1.Intro modelado de datos

2.Gestores de BBDD

3.Proceso de diseño de una BBDD

4.Modelo conceptual

5.Modelo relacional

6.Modelo lógico

1.Normalización

7.Modelo físico

INTRO MODELADO DE DATOS

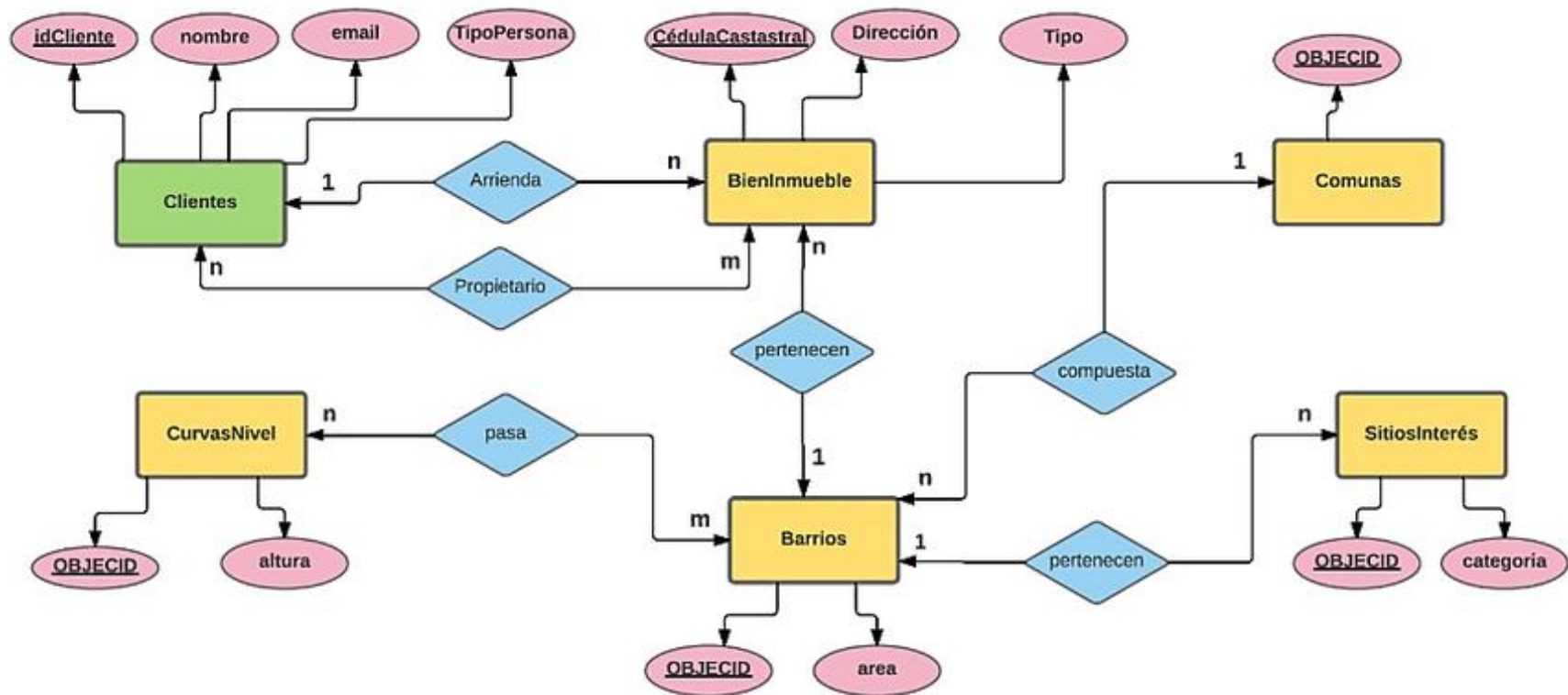
- Modelo de datos:
 - Un modelo de datos es una representación gráfica de la información consumida y producida por un sistema el cual permite analizar los datos presentes en el sistema y las relaciones entre ellos.
 - Un modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.
 - Tipos -> Relacional, Jerárquico, Grafo, etc.

INTRO MODELADO DE DATOS

- El modelado de datos es necesario para poder almacenar la información y luego trabajar con ella.
Si por ejemplo tenemos un dataset y alguna columna no guarda el formato correcto (texto en vez de número por ejemplo), no podríamos trabajar correctamente con él.

INTRO MODELADO DE DATOS

Modelo conceptual E/R



INTRO MODELADO DE DATOS

Modelo lógico

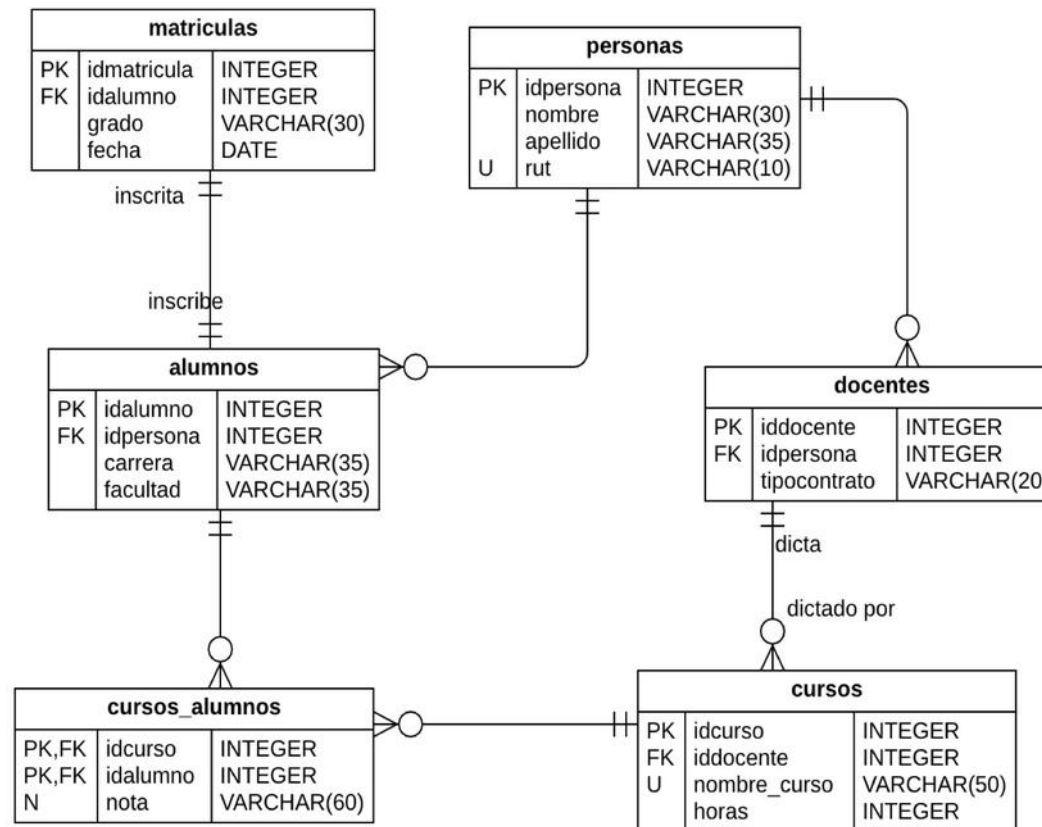


TABLA DE CONTENIDOS

- 1.Intro modelado de datos
- 2.Gestores de BBDD**
- 3.Proceso de diseño de una BBDD
- 4.Modelo conceptual
- 5.Modelo relacional
- 6.Modelo lógico
 - 1.Normalización
- 7.Modelo físico

GESTORES DE BBDD (RDBMS)

- Sistema de gestión de bases de datos relacionales.
 - Un RDBMS es un DBMS diseñado específicamente para bases de datos relacionales. Por lo tanto, los RDBMS son un subconjunto de DBMS.
 - **Almacenamiento estructurado** de los datos (filas y columnas fijas). Esto facilita la localización y el acceso a valores específicos dentro de la base de datos.
 - Es "relacional" porque **los valores dentro de cada tabla están relacionados entre sí**. Las tablas también pueden estar relacionadas con otras tablas. La estructura relacional permite ejecutar consultas en varias tablas a la vez.
- Ejemplo: Oracle, Db2, MariaDb, ...

GESTORES DE BBDD (RDBMS)

- Sistema de gestión de bases de datos relacionales.

<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda
67.789.901	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsain
32.112.312	Santos	Mayor	Peguerinos
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

(a) La tabla *cliente*

<i>número-cuenta</i>	<i>saldo</i>
C-101	500
C-215	700
C-102	400
C-305	350
C-201	900
C-217	750
C-222	700

(b) La tabla *cuenta*

<i>id-cliente</i>	<i>número-cuenta</i>
19.283.746	C-101
19.283.746	C-201
01.928.374	C-215
67.789.901	C-102
18.273.609	C-305
32.112.312	C-217
33.666.999	C-222
01.928.374	C-201

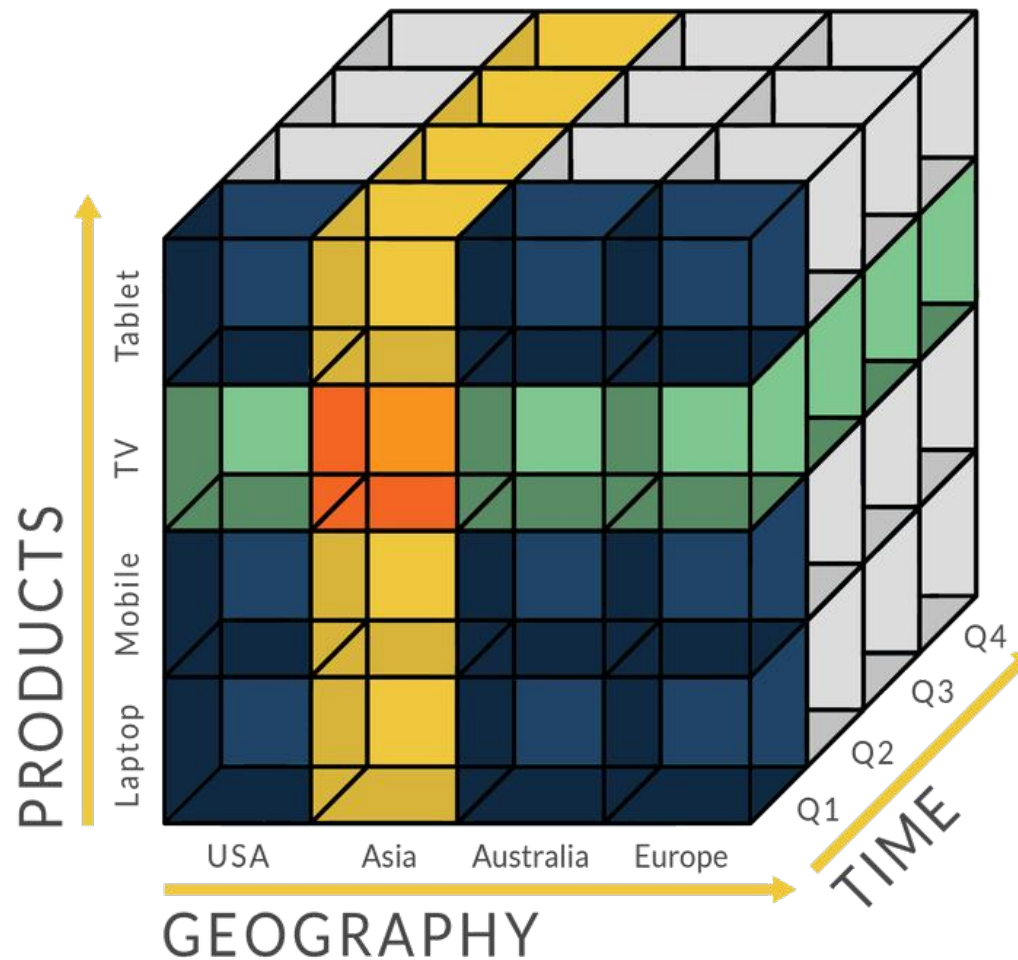
(b) La tabla *impositor*

GESTORES DE BBDD (MDDDB)

- Multidimensional database.
 - Es un aplicativo que trabaja con **BBDD multidimensionales en lugar de relacionales**, es decir los datos son guardados en celdas, la posición de cada celda se define por las dimensiones.
 - Son bases de datos optimizadas para aplicaciones de almacén de datos y procesamiento analítico en línea (**OLAP**).
 - Las bases de datos multidimensionales suelen crearse **a partir de bases de datos relacionales existentes**.
 - Acceso a los datos mediante consultas del tipo: "¿Cuántas vehículos se han vendido en Euskadi en lo que va de año?"
- Ejemplo: Essbase.

GESTORES DE BBDD (MDDDB)

- Multidimensional database.



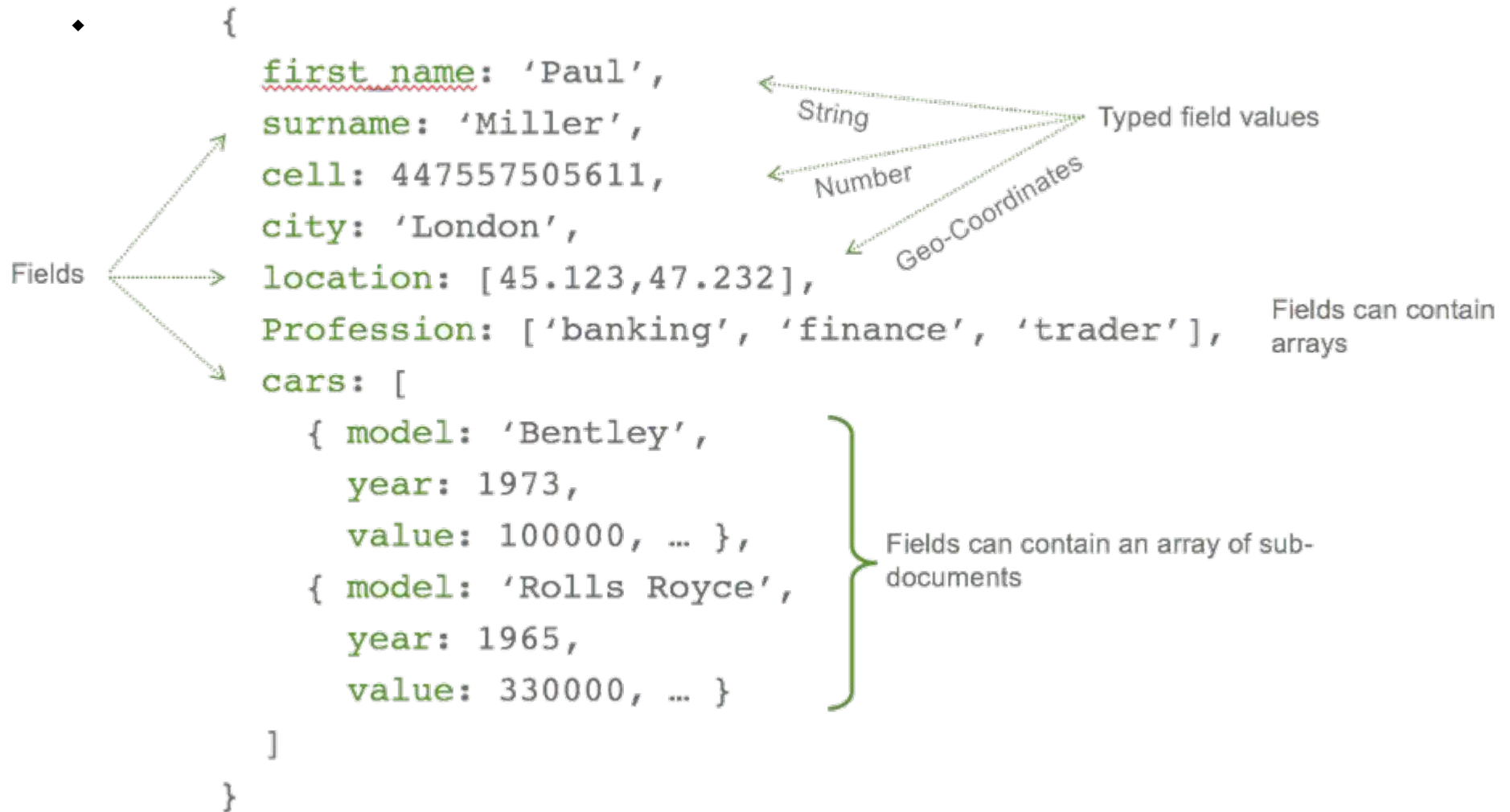
GESTORES DE BBDD (NOSQL DOCUMENTOS)

- NoSQL Database

- Las bases de datos NOSQL, difieren de las relacionales de toda la vida en que **no usan SQL** ya que no son bases de datos relacionales.
- Se adaptan mejor al cambio de estructura que las SQL.
- No usan tablas, columnas, etc.
- La información se guarda en formato par clave-valor (JSON).
- Ejemplos : MongoDB, Couch-Db

GESTORES DE BBDD (NOSQL DOCUMENTOS)

- NoSQL Database



GESTORES DE BBDD (NOSQL GRAFOS)

- **DBOG**

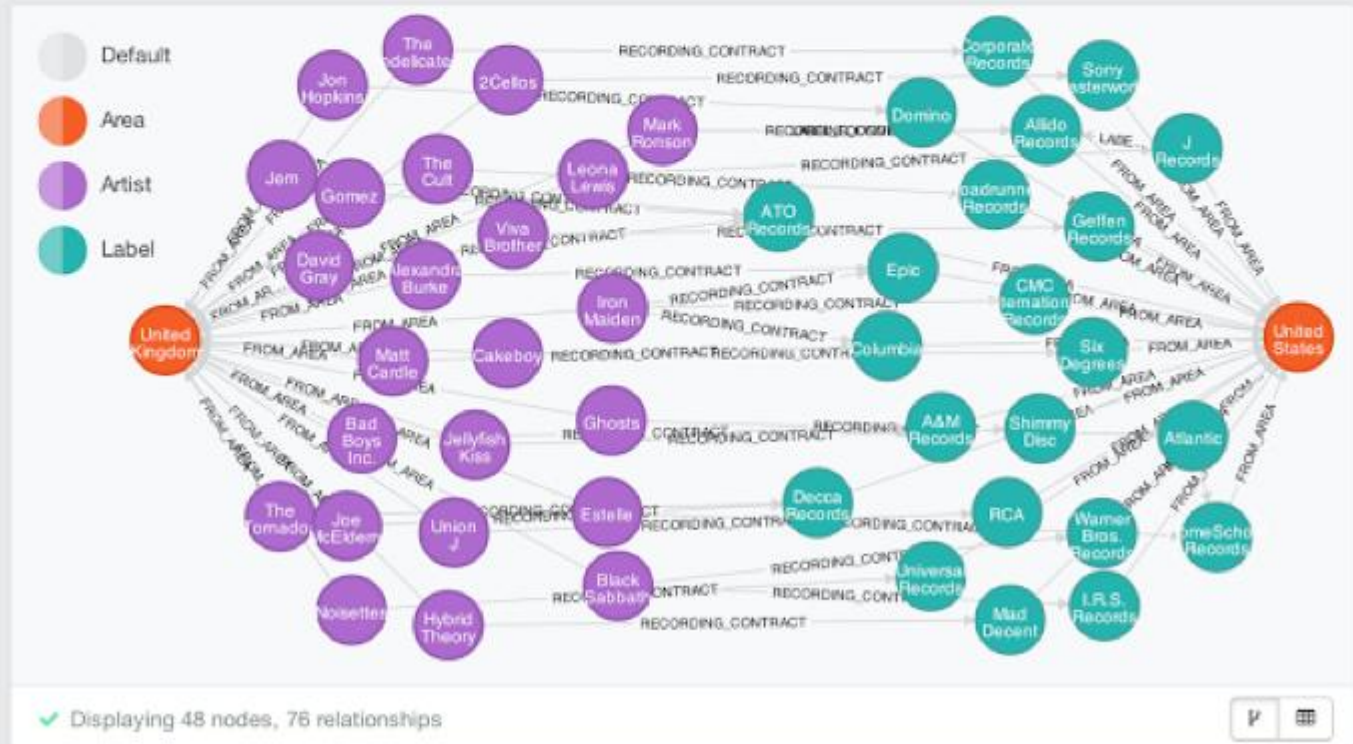
- Las bases de datos orientada a grafos.
- Representan la información como nodos de una grafo y sus relaciones con las aristas del mismo
- Ejemplos : Neo4j, TitanDb, OrientDb

GESTORES DE BBDD (NOSQL GRAFOS)

- DBOG

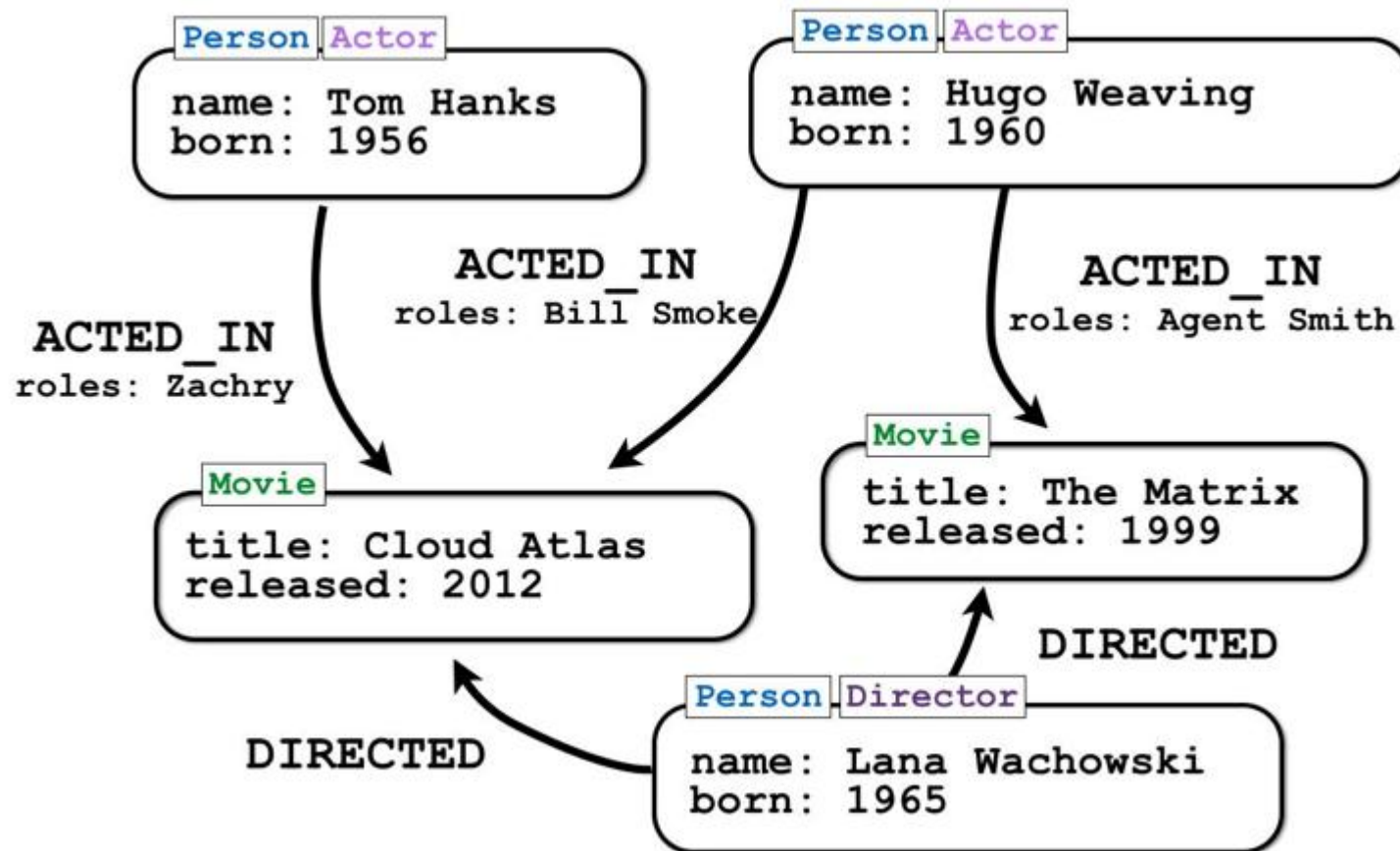
```
1 // British artists signed on American record labels
2 START usa=node:mb_fulltext(name="United States"),
3 gb=node:mb_fulltext(name="United Kingdom")
4 MATCH (gb:Country), (gb:Country),
5 (a:Artist)-[:FROM_AREA]-(gb),
6 (a:Artist)-[:RECORDING_CONTRACT]-(l:Label),
7 (l)-[:FROM_AREA]-(usa)
```

CYPHER START usa=node:mb_fulltext(name="United States"), gb=node:mb_fulltext(name="United



GESTORES DE BBDD (NOSQL GRAFOS)

- DBOG



GESTORES DE BBDD

SQL VS NOSQL

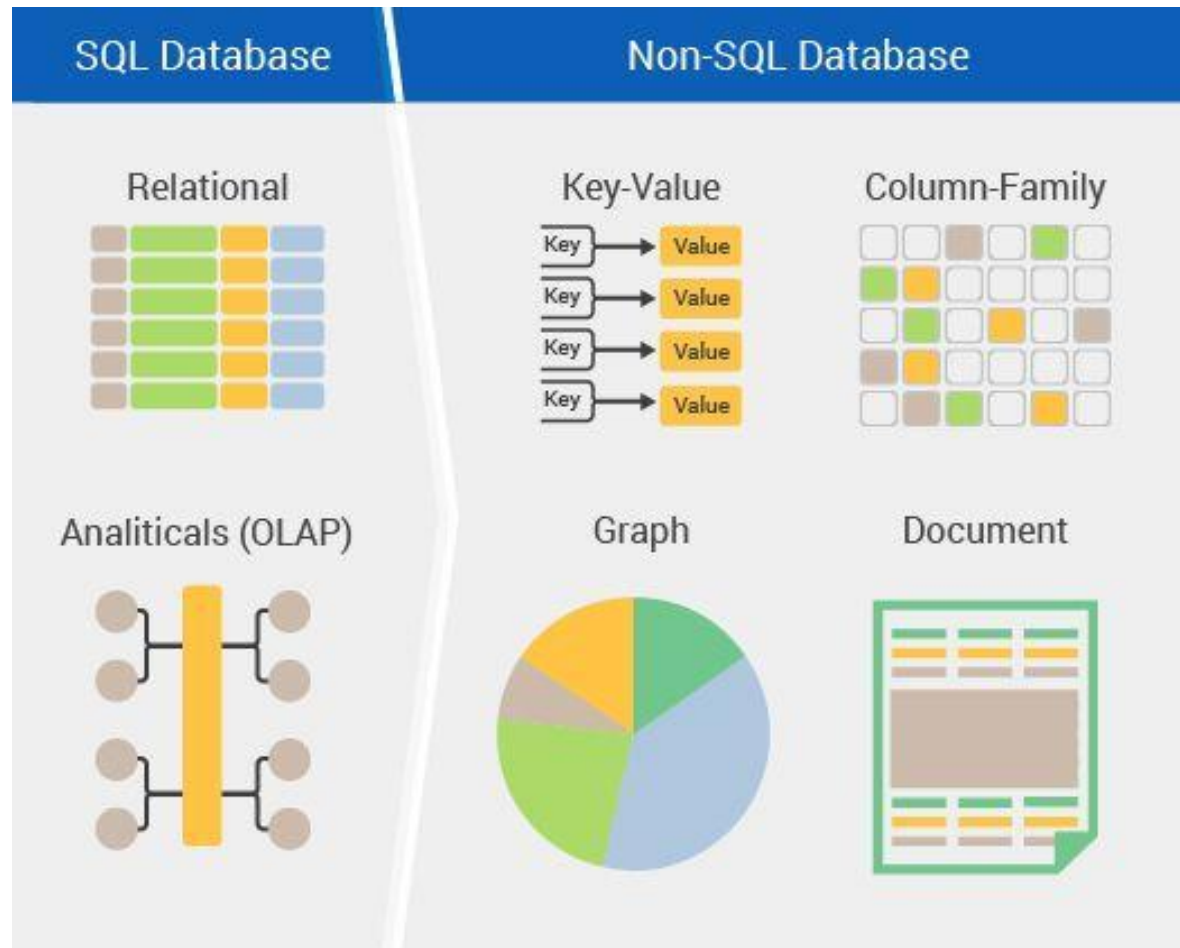


TABLA DE CONTENIDOS

1. Intro modelado de datos
2. Gestores de BBDD
- 3. Proceso de diseño de una BBDD**
4. Modelo conceptual
5. Modelo relacional
6. Modelo lógico
 1. Normalización
7. Modelo físico

PROCESO DE DISEÑO DE UNA BBDD

- Fase 1: Análisis de requisitos
 - Recabar información sobre el uso que se piensa dar a la base de datos. Recogida de requisitos del sistema.
- Fase 2: Modelo conceptual
 - Creación de un esquema conceptual de la base de datos independiente del gestor de BBDD que se vaya a utilizar.
- Fase 3: Elección del gestor de BBDD
 - Elección del modelo de datos, así como el gestor concreto. Por ejemplo, relacional (Oracle, DB2), multidimensional (Essbase, Gentium)...

PROCESO DE DISEÑO DE UNA BBDD

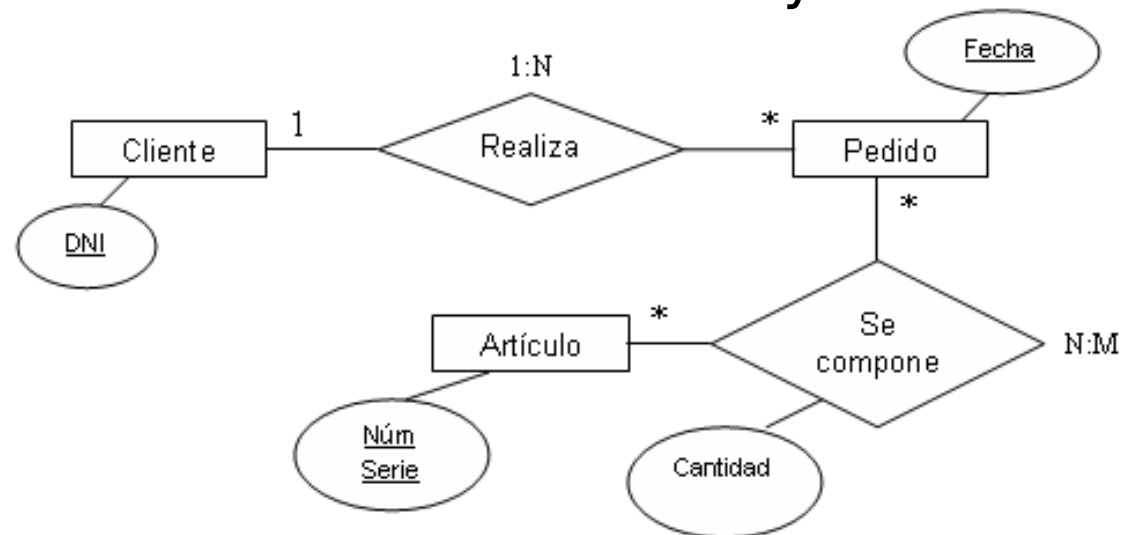
- Fase 4: Modelo lógico
 - Creación del esquema conceptual para el modelo de datos del DBMS elegido.
 - (p.ej. paso del modelo E/R a un conjunto de tablas)
- Fase 5: Modelo físico
 - Creación de la base de datos utilizando DDL (lenguaje de definición de datos del DBMS).
- Fase 6: Uso y mantenimiento
 - Gestión de los datos utilizando el DML (lenguaje de manipulación de datos del DBMS).

TABLA DE CONTENIDOS

- 1.Intro modelado de datos
- 2.Gestores de BBDD
- 3.Proceso de diseño de una BBDD
- 4.Modelo conceptual**
- 5.Modelo relacional
- 6.Modelo lógico
 - 1.Normalización
- 7.Modelo físico

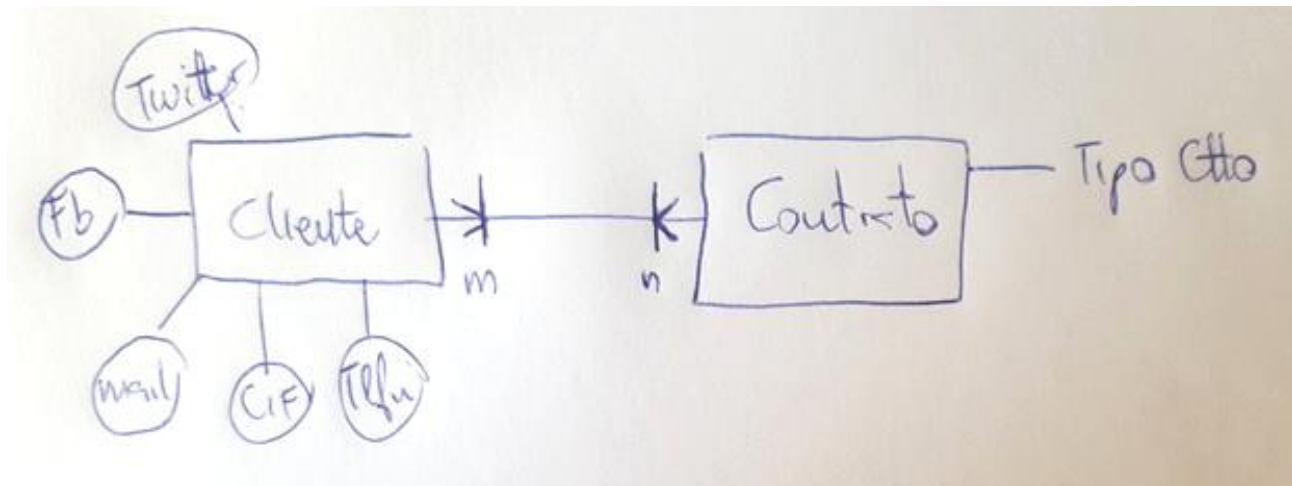
MODELO CONCEPTUAL

- Modelo Entidad/Relación (E/R):
 - Constituye una forma de representar **conceptualmente** la realidad mediante su abstracción en entidades y relaciones.
 - De esta forma intentamos representar el mundo que nos rodea, los datos de nuestro problema mediante una serie de entidades que representan objetos o conceptos así como las relaciones que se dan entre ellos tales como su uso y composición.



MODELO CONCEPTUAL

Esquema conceptual



MODELO CONCEPTUAL

- Entidad: Representan cosas u objetos, que se diferencian claramente entre sí.

Dentro del modelado de datos, cuando veamos un requerimiento de negocio o estemos analizando algo para modelarlo, **cuando nos encontremos con un sustantivo, dicho sustantivo será una entidad**. Cuidado porque no todos los sustantivos son entidades, habrá que tener en cuenta que determinados sustantivos que están catalogados dentro de otros (p.ej un producto es un contrato (a nivel financiero) y por tanto si nos están hablando de contratos y productos, no definiríamos 2 entidades, sino 1 y contratos y productos serían la misma entidad).

MODELO CONCEPTUAL

‣ Ejemplo Entidad:

En un taller mecánico se podrían crear las siguientes entidades:

Coches (objeto físico): contiene la información de cada coche.

Empleado (objeto físico): información de los trabajadores.

Cargo del empleado (cosa abstracta): información de la función del empleado.

Estas entidades se representan en un diagrama con un rectángulo.

MODELO CONCEPTUAL

- Dominio (Business Rule): Definen el conjunto de valores para los cuales un atributo es válido (tipo de dato, longitud, precisión, rango de valores,...).

P.ej, en el caso de una entidad Provincia, el dominio sería el conjunto de valores que puede adoptar la Provincia (Bizkaia, Araba, Burgos, ...). Sirve para homogeneizar los datos de las diferentes entidades del modelo. **Es una muy buena práctica definir los dominios.**

MODELO CONCEPTUAL

‣ Ejemplo Dominio:

Nombre: AUDITORIA HORA

Descripción: Dato reservado para contener la fecha y hora en la que se realizó una operación con vistas a su revisión por Auditoría

Tipo de dato: Timestamp

MODELO CONCEPTUAL

➤ Atributo: Propiedad de una entidad.

Definen o identifican las características de entidad (**es el contenido de la entidad**). Cada entidad, contiene distintos atributos, que dan información sobre la entidad.

MODELO CONCEPTUAL

➤ Ejemplo Atributo:

Posibles atributos para la entidad **Coches**

Número de chasis, matrícula, NIF propietario, marca, ...

MODELO CONCEPTUAL

- Data Item: Sólo aparece en el modelo conceptual avanzado. **Representa un hecho o una definición en un sistema de información y el cual puede o no existir como objeto dentro del modelo.** Un dataitem puede estar o no acoplado a una entidad para crear un atributo, es decir podemos crear el Dataitem y no tenerlo posteriormente asociado a ningún atributo.

MODELO CONCEPTUAL

➤ Ejemplo Data Item:

Matrícula: Combinación de caracteres alfanuméricos o numéricos que identifica e individualiza el vehículo respecto a los demás.

MODELO CONCEPTUAL

- Relaciones entre entidades (cardinalidad): Las relaciones entre entidades son vínculos que nos permiten definir una dependencia entre varias entidades.

Es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable. **Normalmente los verbos son las relaciones entre entidades.**

MODELO CONCEPTUAL

➤ Ejemplo Relaciones entre entidades:

Los empleados del taller (de la entidad "Empleados") tienen un cargo (según la entidad "Cargo del empleado").

Es decir, un atributo de la entidad "Empleados" especificará que cargo tiene en el taller, y tiene que ser idéntico al que ya existe en la entidad "Cargo del empleado".

MODELO CONCEPTUAL

- Clave: Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad).
- Tipos de claves (ver modelo lógico).

MODELO CONCEPTUAL

- Cardinalidad: Define el número máximo de ocurrencias de una entidad que participan en una relación.

MODELO CONCEPTUAL

Las relaciones pueden ser:

- Uno a uno:

Cada instancia de una entidad se relacionará exactamente con una instancia de la entidad asociada

- Uno a muchos:

Una instancia de la primera entidad se relacionará con muchas instancias de la segunda, pero cada instancia de la segunda se relacionará exactamente con una instancia de la primera.

- Muchos a muchos:

Las instancias de ambos lados de la relación de asociarán con muchas instancias de la otra entidad

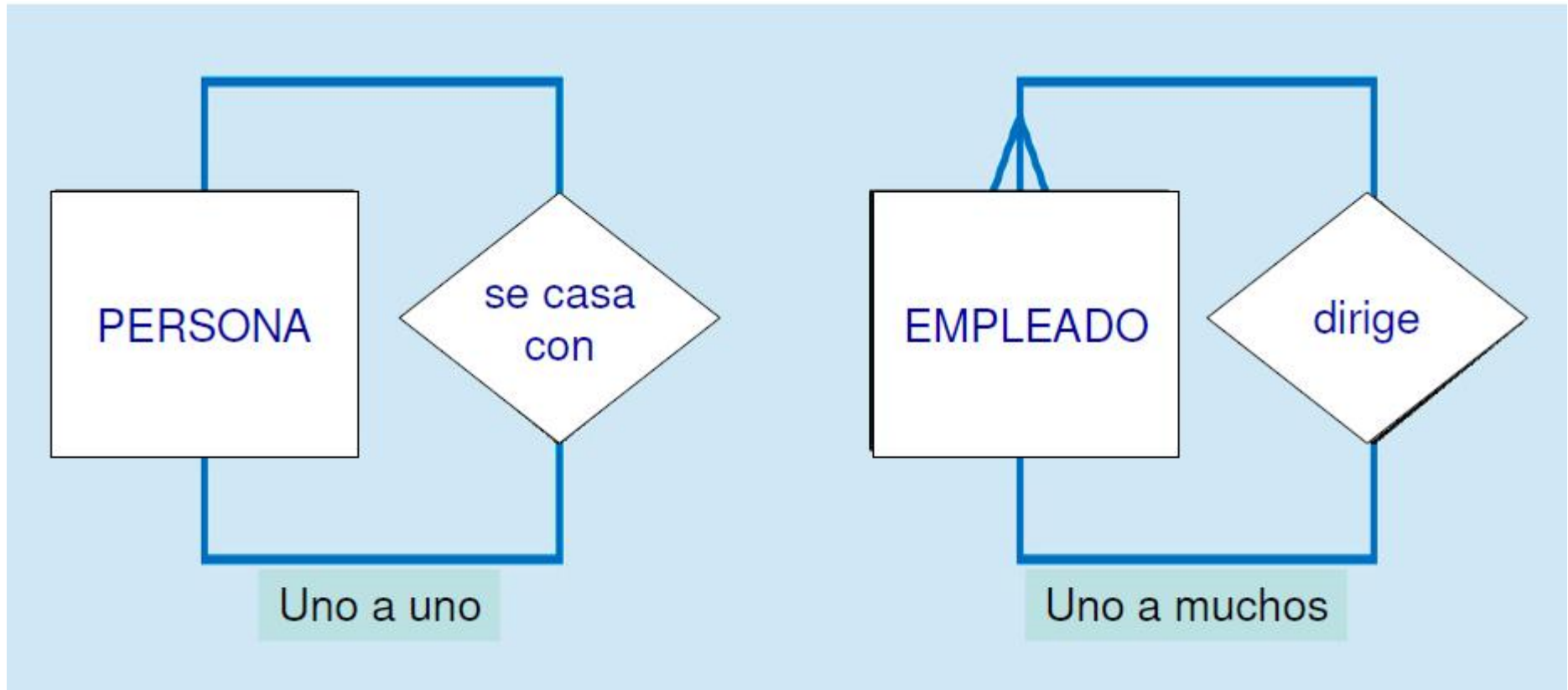
MODELO CONCEPTUAL

- ALCANCE.

Corresponderá a las “cosas que son de interés” para nuestro modelo.

- P.ej., si queremos hacer un modelo sobre venta de comida para llevar en un establecimiento, los datos relativos a **número de ventanas, o servicios en el establecimiento** quedarían fuera del alcance ya que no serían de interés.

GRADOS DE RELACIONES: UNITARIAS



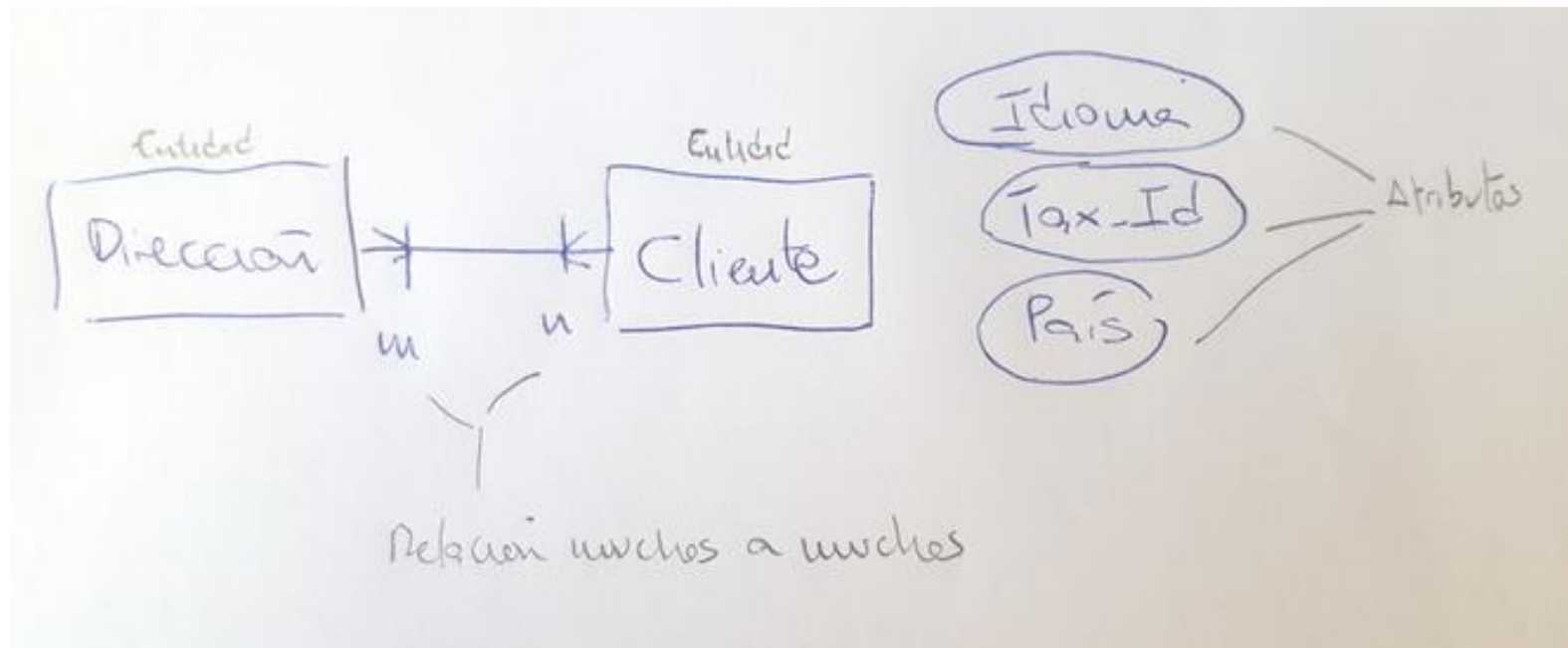
GRADOS DE RELACIONES

EJEMPLOS

- Relación entre entidades de muchos a muchos (m,n)

Verbo → Tiene

Ej: Un cliente tiene varias direcciones. ¿Podría ser una dirección tiene varios clientes?



GRADOS DE RELACIONES

EJEMPLOS

- Relación entre entidades de muchos a muchos (m,n)
Un cliente puede tener n contratos y a su vez, un contrato puede tener m clientes (puede haber varios clientes firmando el mismo contrato), por lo que es una relación de muchos a muchos.
- Semánticamente el esquema de relación se reduce al verbo TIENE.

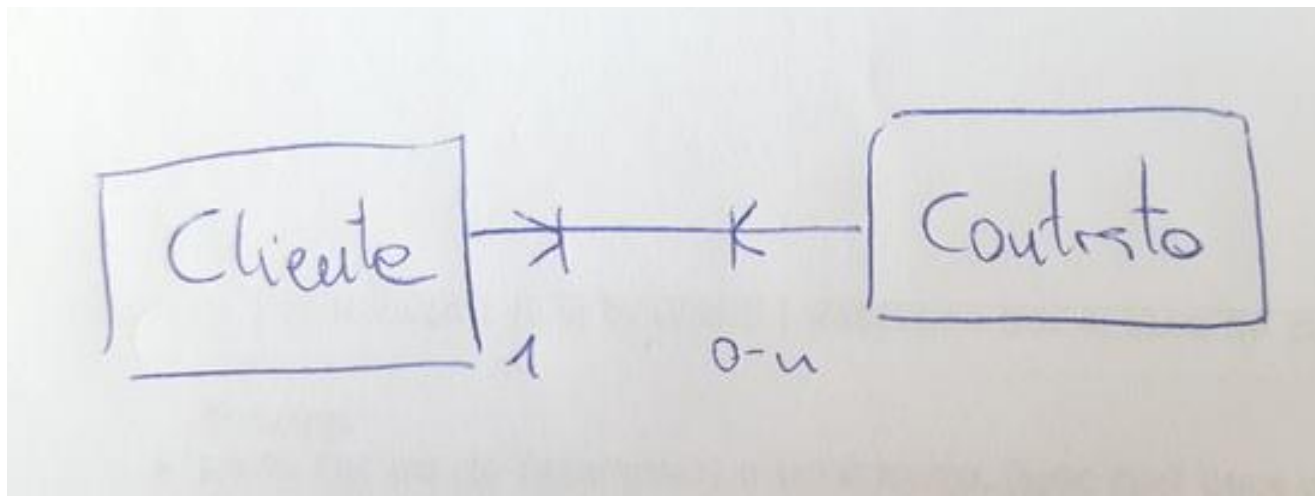
GRADOS DE RELACIONES

EJEMPLOS

- Relación entre entidades de 1 a varios (o no) (1,0-n)

Verbo → Puede tener

Ej: Un cliente puede tener varios contratos.

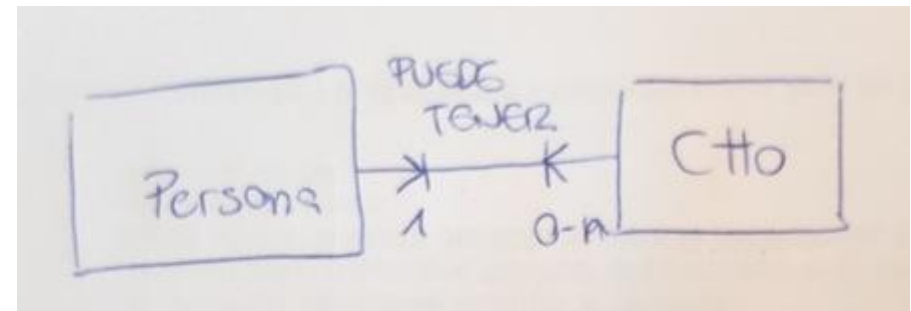
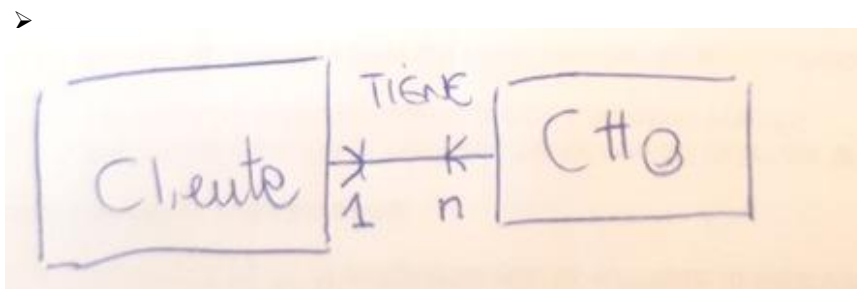


GRADOS DE RELACIONES

EJEMPLOS

- Relación entre entidades de 1 a varios (o no) (1,0-n)

Supongamos el caso de un cliente y una persona física en una tienda de ropa. Todo cliente por definición debe tener al menos un contrato (del tipo que sea, habrá hecho una compra, sino no sería cliente), sin embargo una persona física, no tiene porqué tener ningún contrato (todavía no ha comprado nada y está ojeando la tienda). Todo esto se representaría de la siguiente manera:



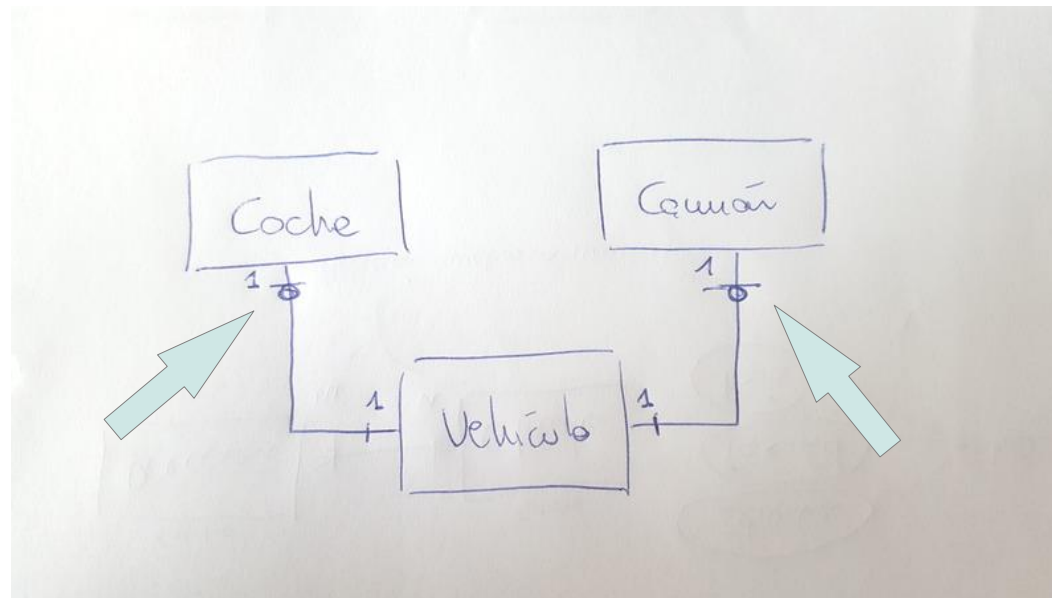
GRADOS DE RELACIONES

EJEMPLOS

➤ Relación de exclusividad

Son relaciones del tipo, puede ser éste o aquello).

Se representan de la siguiente manera:



Un vehículo puede
Ser coche o camión,
pero no las 2 cosas.

MODELO CONCEPTUAL

Un ejemplo más elaborado

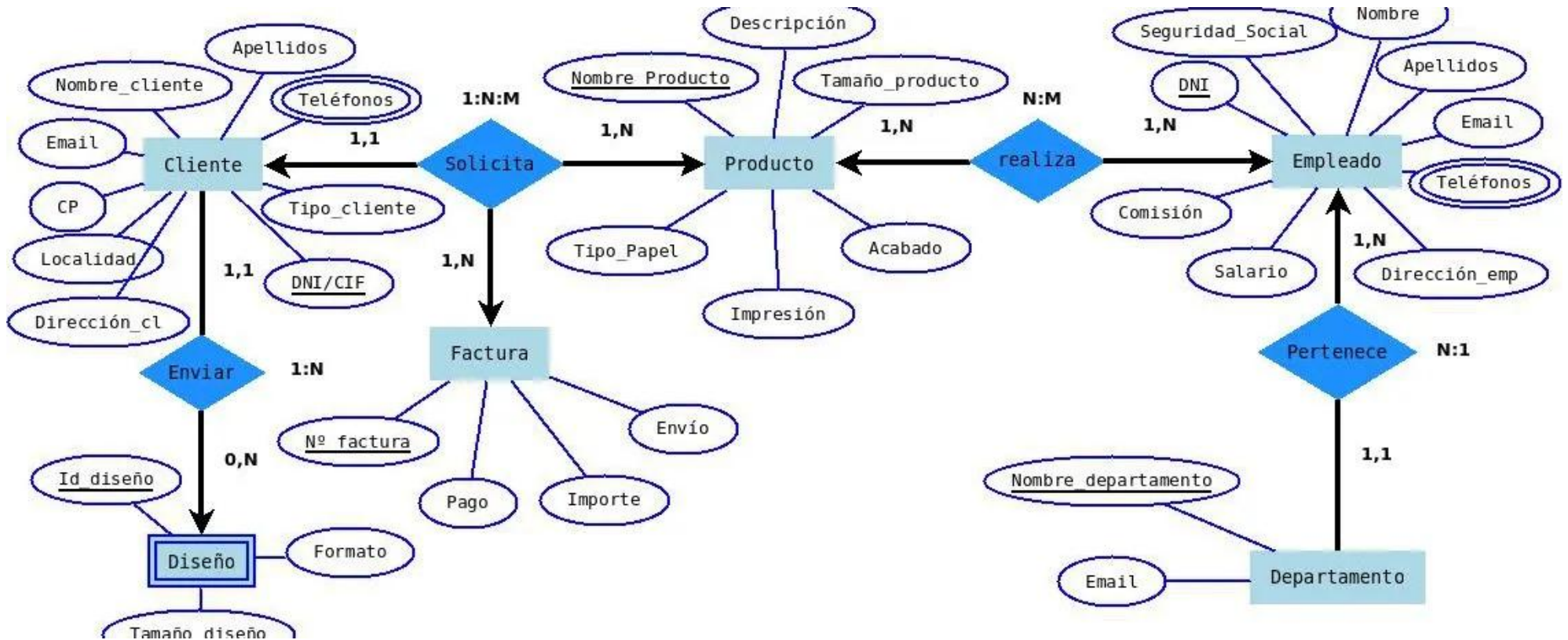


TABLA DE CONTENIDOS

- 1.Intro modelado de datos
- 2.Gestores de BBDD
- 3.Proceso de diseño de una BBDD
- 4.Modelo conceptual
- 5.Modelo relacional**
- 6.Modelo lógico
 - 1.Normalización
- 7.Modelo físico

MODELO RELACIONAL

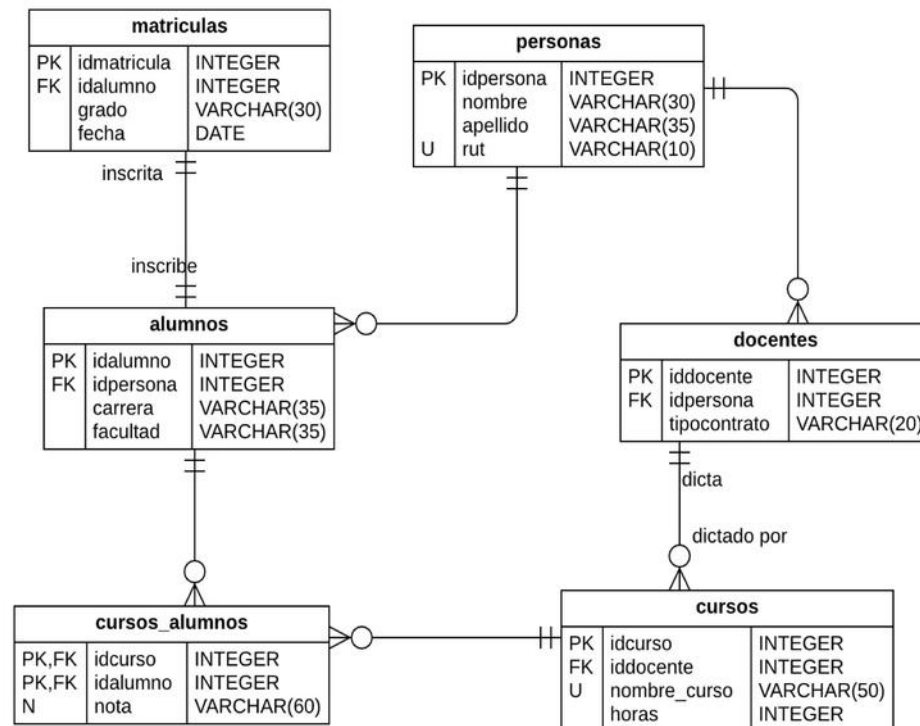
- Modelo Relacional:
 - Es el más empleado.
 - Todos los datos visibles al usuario están organizados estrictamente como tablas de valores.
 - Todas las operaciones sobre la base de datos operan sobre esas tablas.
 - Cada fila de una tabla es una instancia de los datos.
 - Cada columna de una tabla es un atributo (valor indivisible que tiene significado por sí solo).
 - Es el modelo de datos más sencillo y cercano a la forma humana de organizar la información.

TABLA DE CONTENIDOS

- 1.Intro modelado de datos
- 2.Gestores de BBDD
- 3.Proceso de diseño de una BBDD
- 4.Modelo conceptual
- 5.Modelo relacional
- 6.Modelo lógico**
 - 1.Normalización
- 7.Modelo físico

MODELO LÓGICO

- Una vez tengamos nuestro modelo relacional y/o conceptual definidos, podemos pasar a la fase de modelado lógico.



MODELO LÓGICO

- En el modelo lógico, las claves pasan a ser:
 - Primarias: Identifican inequívocamente un sólo atributo no permitiendo que se repita en la misma entidad. Formada por un campo o grupo de campos.

MODELO LÓGICO

- En el modelo lógico, las claves pasan a ser:
 - Externas o foráneas: Este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad. Identifica una columna o grupo de columnas en una tabla (tabla hija) que se refiere a una columna o grupo de columnas en otra tabla (tabla padre). **Las columnas hija deber ser la clave primaria en la tabla padre.**
- Un ejemplo simple de una clave foránea en una base de datos es una tabla primaria "Estudiante" con Student_ID como su clave primaria. En la tabla secundaria relacionada "Curso_Matriculación" con Course_ID como clave primaria, para cada curso en el que un estudiante se haya matriculado aparece la clave foránea Student_ID de la tabla "Estudiante".

MODELO LÓGICO

- En el modelo lógico, las claves pasan a ser:
 - Clave subrogada: Campo numérico cuyo único requisito es almacenar un **valor numérico único** para cada fila de la tabla

TABLA DE CONTENIDOS

- 1.Intro modelado de datos
- 2.Gestores de BBDD
- 3.Proceso de diseño de una BBDD
- 4.Modelo conceptual
- 5.Modelo relacional
- 6.Modelo lógico
- 1.Normalización**
- 7.Modelo físico

MODELO LÓGICO NORMALIZACIÓN

➤ Normalización

- El proceso de normalización tiene como objetivo dejar una BBDD relacional con **poca o ninguna redundancia de datos**, es decir en 3ª forma normal o superior.

MODELO LÓGICO NORMALIZACIÓN

➤ Formas normales

→ **La primera forma normal:** Nos lleva a NO REPETIR los datos en nuestras tablas.

P.ej: Si nuestra tabla de ventas, repite una y otra vez (por cada venta), el nombre, domicilio y otros datos del cliente, NO habríamos aplicado esta normalización.

MODELO LÓGICO NORMALIZACIÓN

➤ Formas normales

→ La primera forma normal:

Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Estado	Num_art	nom_art	cant	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

PRIMERA FORMAL NORMAL (1FN)

Al examinar estos registros, podemos darnos cuenta que contienen un grupo repetido para NUM_ART, NOM_ART, CANT y PRECIO. La 1FN prohíbe los grupos repetidos, por lo tanto tenemos que convertir a la primera forma normal. Los pasos a seguir son:

- Tenemos que eliminar los grupos repetidos.
- Tenemos que crear una nueva tabla con la PK de la tabla base y el grupo repetido.

Los registros quedan ahora conformados en dos tablas que llamaremos ORDENES y ARTICULOS_ORDENES

MODELO LÓGICO NORMALIZACIÓN

➤ Formas normales

→ La primera forma normal:

ordenes (id_orden, fecha, id_cliente, nom_cliente, estado)

Articulos_ordenes (id_orden, num_art, nom_art, cant, precio)

Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Estado
2301	23/02/11	101	Martin	Caracas
2302	25/02/11	107	Herman	Coro
2303	27/02/11	110	Pedro	Maracay

Articulos_ordenes

Id_orden	Num_art	nom_art	cant	Precio
2301	3786	Red	3	35,00
2301	4011	Raqueta	6	65,00
2301	9132	Paq-3	8	4,75
2302	5794	Paq-6	4	5,00
2303	4011	Raqueta	2	65,00
2303	3141	Funda	2	10,00

MODELO LÓGICO NORMALIZACIÓN

➤ Formas normales

→ **La segunda forma normal:** Para aplicarla, necesariamente deberemos haber aplicado la Primera Forma Normal).

En etapa, CADA COLUMNA de la tabla, debe depender de una clave.

Esto significa que todo un registro debe depender únicamente de la clave principal, **si tuvieramos alguna columna que se repite a lo largo de todos los registros, dichos datos deberian atomizarse en una nueva.**

MODELO LÓGICO NORMALIZACIÓN

➤ Formas normales

→ La segunda forma normal:

Articulos_ordenes (id_orden, num_art, cant)

Articulos_ordenes

Id_orden	Num_art	cant
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

Articulos (num_art, nom_art, precio)

Articulos

Num_art	nom_art	Precio
3786	Red	35,00
4011	Raqueta	65,00
9132	Paq-3	4,75
5794	Paq-6	5,00
3141	Funda	10,00

MODELO LÓGICO NORMALIZACIÓN

➤ Formas normales

- ➔ **La tercera forma normal:** Tenemos que **eliminar cualquier columna no clave que sea dependiente de otra columna no clave.**

Pasos a seguir:

- Determinar las columnas que son dependientes de otra columna no clave.
- Eliminar esas columnas de la tabla base.
- Crear una segunda tabla con esas columnas y con la columna no clave de la cual son dependientes.

MODELO LÓGICO NORMALIZACIÓN

➤ Formas normales

➔ La tercera forma normal:

Al observar las tablas que hemos creado, nos damos cuenta que tanto la tabla ARTICULOS, como la tabla ARTICULOS_ORDENES se encuentran en 3FN. Sin embargo la tabla ORDENES no lo está, ya que NOM_CLIENTE y ESTADO son dependientes de ID_CLIENTE, y esta columna no es la llave primaria.

Para normalizar esta tabla, moveremos las columnas no llave y la columna llave de la cual dependen dentro de una nueva tabla CLIENTES. Las nuevas tablas CLIENTES y ORDENES se muestran a continuación.

ordenes (id_orden, fecha, id_cliente)

Ordenes

Id_orden	Fecha	Id_cliente
2301	23/02/11	101
2302	25/02/11	107
2303	27/02/11	110

Clientes(id_cliente, nom_cliente, estado)

Clientes

Id_cliente	Nom_cliente	Estado
101	Martin	Caracas
107	Herman	Coro
110	Pedro	Maracay

Por lo tanto la base de datos queda de la siguiente manera:

ordenes (id_orden, fecha, id_cliente)

Clientes (id_cliente, nom_cliente, estado)

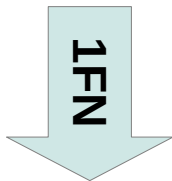
Articulos (num_art, nom_art, precio)

Articulos_ordenes (id_orden, num_art, cant)

MODELO LÓGICO NORMALIZACIÓN

Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Estado	Num_art	nom_art	cant	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00



ordenes (id_orden, fecha, id_cliente, nom_cliente, estado)
Articulos_ordenes (id_orden, num_art, nom_art, cant, precio)

Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Estado
2301	23/02/11	101	Martin	Caracas
2302	25/02/11	107	Herman	Coro
2303	27/02/11	110	Pedro	Maracay

Articulos_ordenes

Id_orden	Num_art	nom_art	cant	Precio
2301	3786	Red	3	35,00
2301	4011	Raqueta	6	65,00
2301	9132	Paq-3	8	4,75
2302	5794	Paq-6	4	5,00
2303	4011	Raqueta	2	65,00
2303	3141	Funda	2	10,00



Articulos_ordenes (id_orden, num_art, cant)

Articulos_ordenes

Id_orden	Num_art	cant
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

Articulos (num_art, nom_art, precio)

Articulos

Num_art	nom_art	Precio
3786	Red	35,00
4011	Raqueta	65,00
9132	Paq-3	4,75
5794	Paq-6	5,00
3141	Funda	10,00



ordenes (id_orden, fecha, id_cliente)

Ordenes

Id_orden	Fecha	Id_cliente
2301	23/02/11	101
2302	25/02/11	107
2303	27/02/11	110

Cientes(id_cliente, nom_cliente, estado)

Ordenes

Id_cliente	Nom_cliente	Estado
101	Martin	Caracas
107	Herman	Coro
110	Pedro	Maracay

Por lo tanto la base de datos queda de la siguiente manera

ordenes (id_orden, fecha, id_cliente)
Cientes (id_cliente, nom_cliente, estado)
Articulos (num_art, nom_art, precio)
Articulos_ordenes (id_orden, num_art, cant)

TABLA DE CONTENIDOS

- 1.Intro modelado de datos
- 2.Gestores de BBDD
- 3.Proceso de diseño de una BBDD
- 4.Modelo conceptual
- 5.Modelo relacional
- 6.Modelo lógico
 - 1.Normalización
- 7.Modelo físico**

MODELO FÍSICO

➤ Objetivo

- Determinar estructuras de almacenamiento y estructuras de acceso para que las aplicaciones que accedan a la BD obtengan un buen rendimiento.
- El Diseño Físico consiste en **elegir las estructuras más apropiadas para los ficheros de la BD**, de entre las opciones que ofrece el SGBD, y según el uso que se dará a los datos.
- Es muy dependiente del SGBD comercial seleccionado

EJEMPLO

- Convertir la tabla existente en el excel 'Modelo dimensional.xlsx' a forma 3FN.

Copyright (c) 2021 Germán Alonso Lascurain

This work (but the quoted images, whose rights are reserved to their owners*) is licensed under the Creative Commons “Attribution-ShareAlike” License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>



Germán Alonso Lascurain
germanalonso@opendeusto.es