

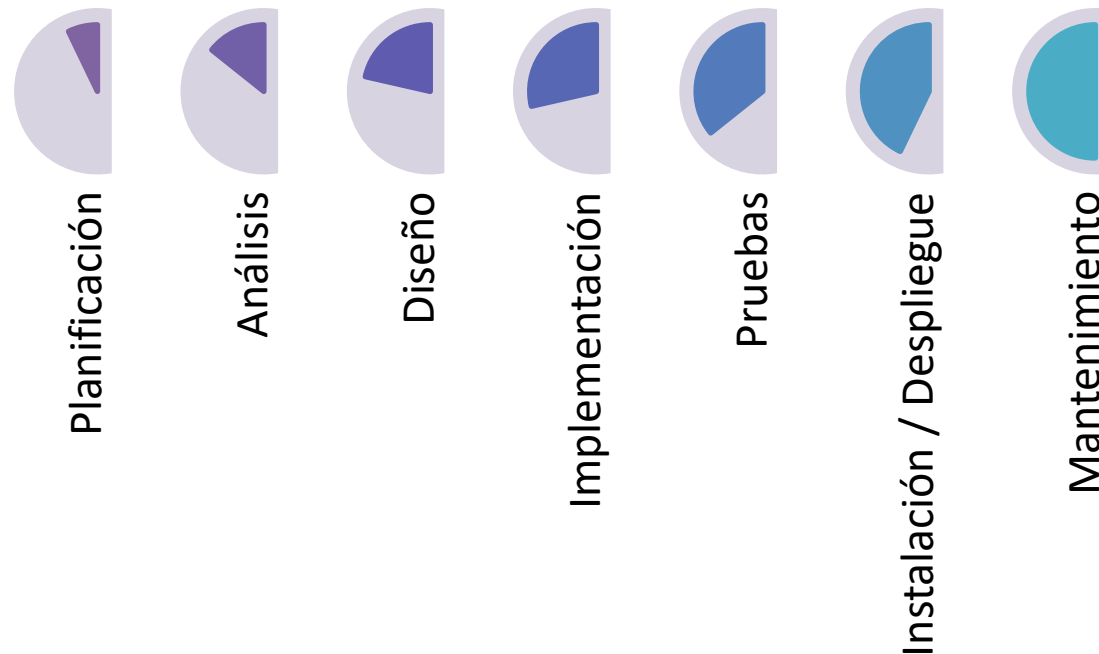
Diseño de bases de datos relacionales.

*Máster en Business Intelligence e
Innovación Tecnológica*

Índice de contenidos

1. Introducción
2. Diseño conceptual
3. Diseño lógico
4. Diseño físico
5. Normalización
6. Introducción a UML
7. Diagrama Relacional
8. Caso práctico de diseño.

El diseño de una BDD es una fase inicial en la que se analizan los requisitos de los clientes y se define la que será la BDD final de forma que cumpla de la forma más acurada con las necesidades de la organización.



Ciclo de vida de desarrollo de sistemas de la información

Por desgracia, es bastante frecuente encontrar proyectos dónde la fase de diseño no ha sido considerada, bien por temas de presupuesto, temas de timing u otros.

También es frecuente que en estos casos, los errores o carencias derivados de una fase de diseño deficiente acarreen al proyecto sobrecostes incluso superiores al que supondría el de un diseño correcto. Hay que tener en cuenta que, por ejemplo, cualquier cambio en la definición de una base de datos puede suponer modificar el código de las aplicaciones que se alimentan de ella: capas de acceso a datos, capas de visualización, etc.

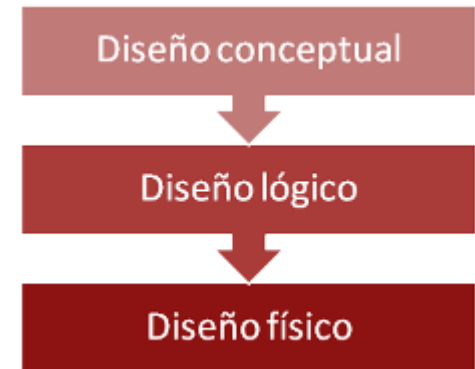
A pesar de poder disponer de herramientas, conocimientos o experiencia, es muy importante poder enfocar esta fase.

Uno de los procedimientos de diseño más comunes es el modelo ANSI /SPARC:

Conceptual.-realización de un esquema o diagrama conceptual representativo de las entidades y sus relaciones. Es el nivel de abstracción más alto. La herramienta típica utilizada para la representación de este modelo es el diagrama Entidad-Relación.

Lógico.-consiste en la representación de un esquema lógico de la estructura. Depende del tipo de SGBD o del modelo elegido (jerárquico, de red, relacional). Existen una serie de reglas que permiten transformar el diagrama Entidad-Relación del diseño conceptual al diseño lógico.

Físico.-es una descripción de la implementación de la base de datos. Incluye la descripción de las estructuras de almacenamiento y los métodos de acceso a los datos. Depende de un SGBD específico. La forma más sencilla es implementar el diseño lógico mediante el lenguaje SQL, obteniendo así el diseño físico.



Modelo Entidad - Relación

Es un modelo semántico de datos

El modelo E/R permite representar, utilizando los diagramas E/R,

- las estructuras que constituyen el contenido del SI
- y las restricciones que limitan las ocurrencias válidas de las mismas.

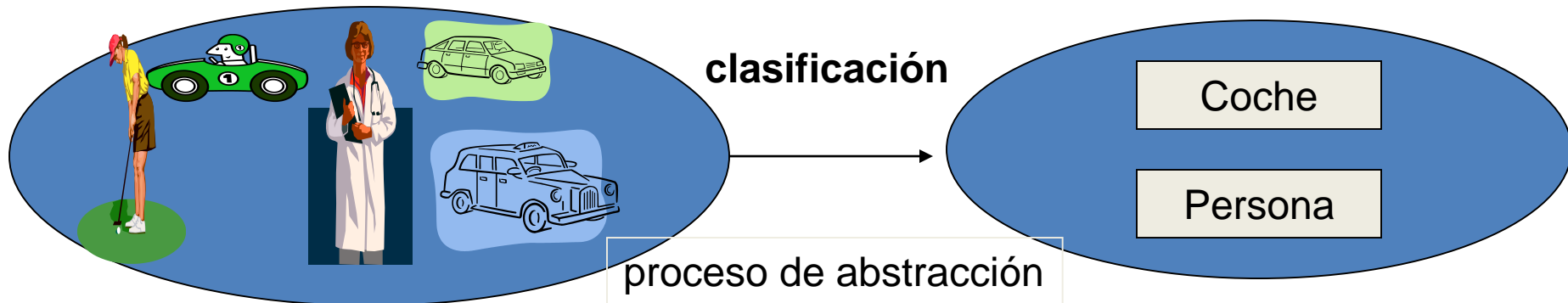
Utiliza tres conceptos:

- Entidad
- Atributo
- Relación

Modelo E-R: Entidad

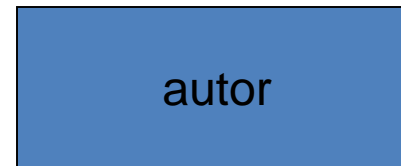
Definición: se define **entidad** como aquel objeto (real o abstracto) acerca del cual queremos almacenar información en la base de datos.

Denominaremos a la estructura genérica en su sentido abstracto **tipo de entidad**, mientras que **entidad** será cada una de las ocurrencias o instancias de este tipo de entidad.



Modelo E-R: Entidad

La representación gráfica de un tipo de entidad es un rectángulo etiquetado con el nombre del tipo de entidad:



Tres reglas generales que debe cumplir cualquier entidad:

- Tiene que tener existencia propia
- Cada ocurrencia de un tipo debe poder distinguirse de las demás
- Todas las ocurrencias de un tipo de entidad deben tener los mismos tipos de características (atributos).

Modelo E-R: Relaciones

Los objetos de un SI se asocian con otros, siendo también de interés modelar estas conexiones.

Para ellos se utilizan los tipos de relaciones:

- llamaremos tipo de relación a la estructura genérica del conjunto de relaciones existentes entre dos o más tipos de entidad.

El tipo de relación se representa mediante un rombo etiquetado con el nombre de la relación, unido mediante arcos a los tipos de entidad que asocia.

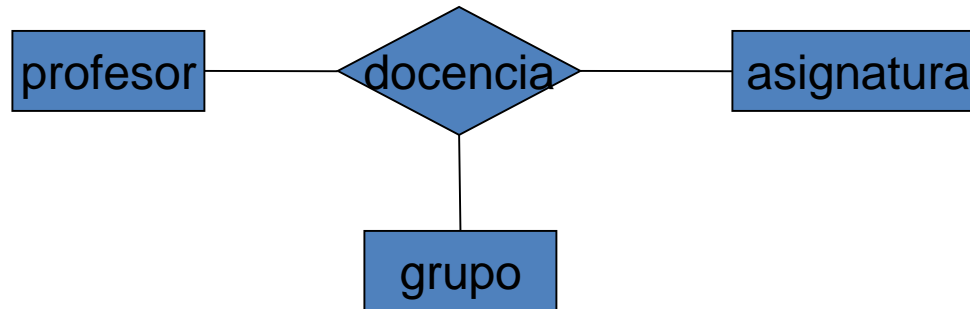
Modelo E-R: Relaciones

- Ejemplos de relaciones:

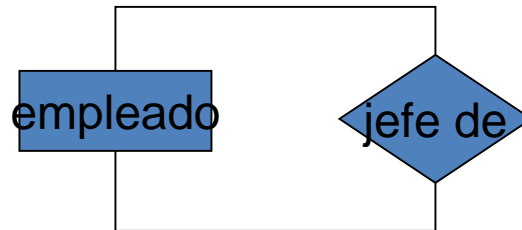
Relación binaria



Relación ternaria



Relación reflexiva

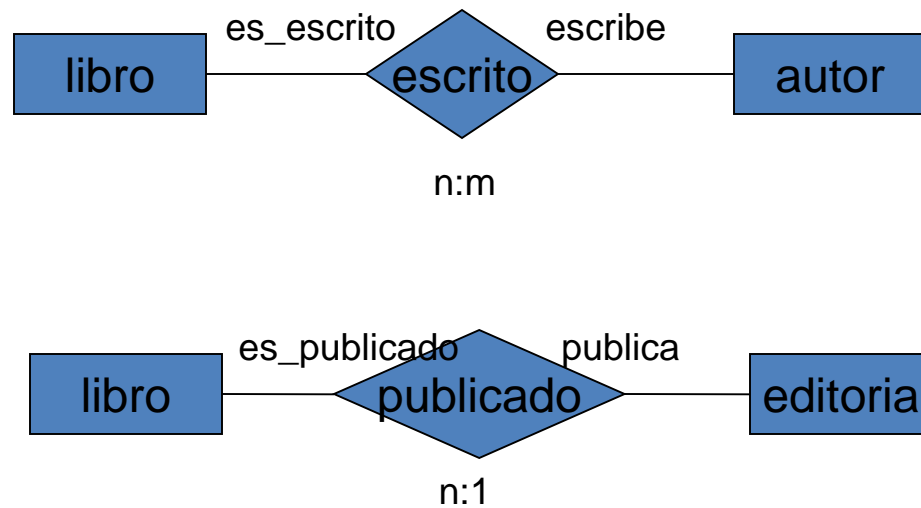


Modelo E-R: Relaciones

- Una relación se define por su nombre y por su grado.
 - El nombre es el identificador que se le da a la propia relación,
 - El grado equivale al número de tipos de entidad a los que asocia o relaciona.
 - Relación de grado 1 o reflexiva.
 - Relación de grado 2 o binaria.
 - Relación de grado 3 o ternaria.
- El tipo de correspondencia es el número máximo de ocurrencias de cada tipo de entidad que pueden intervenir en una ocurrencia del tipo de relación que se está tratando. Gráficamente, esto se representa con alguna de estas etiquetas textuales: 1:1, 1:N, N:M.

Modelo E-R: Relaciones

- Ejemplo de Correspondencia:



Modelo E-R: Atributos

- Los atributos permiten representar:
 - las propiedades de los objetos (entidades)
 - las propiedades de las asociaciones entre ellos.
- La representación gráfica de un atributo consiste en una elipse con el nombre del atributo en su interior.



nombre

- Entre todos los atributos de un tipo de entidad debemos elegir uno o varios que identifiquen unívocamente (clave primaria) cada entidad. Estos atributos se representarán de la misma forma, pero con el nombre del atributo subrayado.



nombre

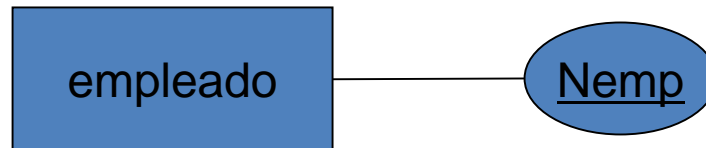
Modelo E-R: Restricciones

- El modelo E/R permite la inclusión de algunas restricciones que limitan el conjunto de ocurrencias válidas.
- Se pueden definir sobre:
 - Entidades
 - Restricción de identificación.
 - Restricción de unicidad.
 - Relaciones
 - Restricción de participación.

Modelo E-R: Restricciones sobre entidades

Restricción de identificación:

Exige que cada ocurrencia de un tipo entidad se identifique unívocamente de forma que se pueda diferenciar del resto de ocurrencias (*clave principal*).



Modelo E-R: Restricciones sobre entidades

Restricción de unicidad:

- Indica que las distintas ocurrencias de una entidad deben tomar valores distintos (o nulos) para el atributo (o conjunto de atributos) donde se define esta restricción. (*No se pueden repetir*)
- Representación: subrayando los atributos con una línea discontinua. En caso de que haya varios conjuntos de atributos con esta restricción se añade un subíndice en la línea.



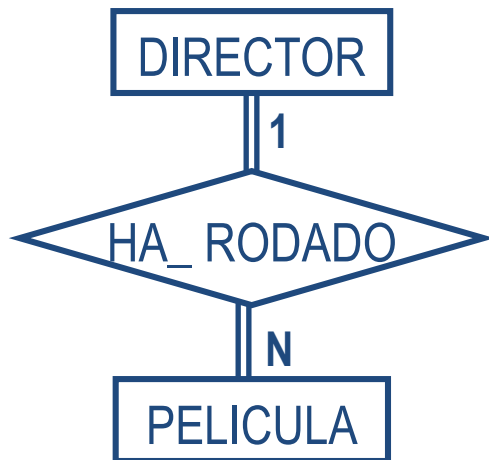
Modelo E-R: Restricciones sobre entidades

Restricción de participación:

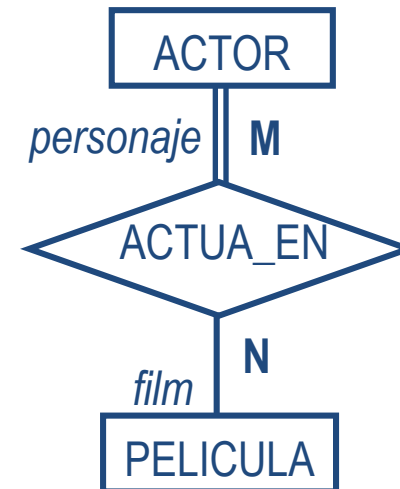
- Especifica si la existencia de una entidad depende de que esté relacionada con otra entidad a través del tipo de relación.

La restricción de participación total se le conoce como dependencia de existencia (se puede representar con una doble línea) e implica la existencia de **integridad referencial**

Modelo E-R: Ejemplos de notación



*Un director lo es si ha rodado alguna película
(sin codirecciones)*



*Un actor lo es si ha actuado en alguna película
Pueden haber películas sin actores*

Introducción al diseño lógico

- El diseño lógico consiste en una transformación de modelos.
 - El modelo conceptual se convierte al modelo de la base de datos para el que se está diseñando, en nuestro caso modelo relacional
- El modelo relacional es una forma de representar los datos (mediante tablas) que se ocupa de tres aspectos de los datos: su estructura, su integridad y su manipulación.
- Una base de datos relacional estará compuesta por una o más tablas (“relaciones”)

El modelo relacional

- Una **tabla** (o relación) guarda una colección de información conteniendo registros (o filas) referidos a entidades de un mismo tipo
- Un **registro** (fila o tupla) es un dato compuesto de otros más simples que contiene la información de una entidad.
- Un **campo** (o atributo) es el contenedor de cada uno de los datos más simples que constituyen un registro.

El modelo relacional

- El tipo de información que un campo puede contener viene determinado por el tipo de **tipo de dato** (dominio), que puede ser:
 - texto (de longitud fija o variable)
 - numérico (entero, entero largo, con decimales)
 - fechas, horas
- Todos los registros de una misma tabla tienen la misma estructura de campos (obviamente con diferentes contenidos)

El modelo relacional

El modelo relacional incluye dos **reglas de integridad**, que se aplican a cualquier BD, basadas en el concepto de clave:

- Clave primaria: regla de integridad de las entidades.
- Clave ajena: regla de integridad referencial.

Clave primaria:

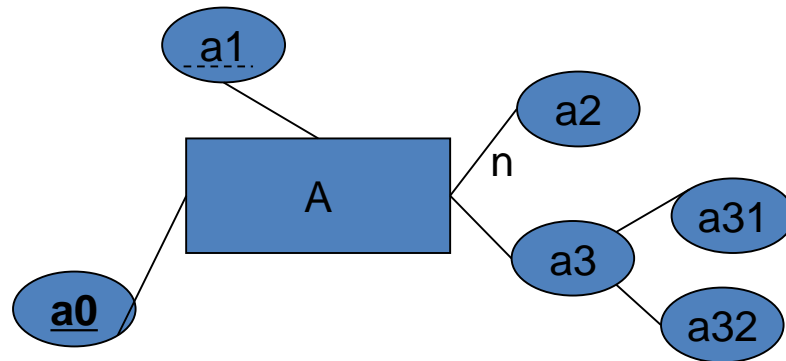
- Es un identificador único para la tabla, esto es, una columna o conjunto de columnas de forma que nunca existen dos filas en la misma tabla con el mismo valor en esa columna o conjunto de columnas.

Clave ajena:

- Una clave ajena es un atributo (o un compuesto de atributos) de una relación R1 cuyos valores deben concordar con los de la clave primaria de alguna fila de la relación R2 (donde R1 y R2 no son necesariamente distintas).

Transformación de entidades

- Se crea una relación para cada tipo de entidad presente en el esquema E/R.
- Se incluyen todos los atributos que caracterizan las entidades, indicando para cada uno el dominio (tipo de datos que puede contener).
- De los atributos compuestos se incluyen sólo sus componentes finales.
- Se indica la clave primaria
- Se “marcan” los campos “únicos”



A(a0:dom_a0, a1:dom_a1, a2:dom_a2, a31:dom_a31, a32:dom_a32)

Clave Primaria: a0

Único: a1

Transformación de relaciones

Relaciones N:M.

- Se transforman en una relación que tendrá como clave primaria la concatenación de los atributos principales de cada una de las entidades que relaciona.
- Estos atributos deben tener restricción de clave ajena respecto a cada una de las tablas donde ese atributo es clave primaria.
- Habrá que considerar lo que ocurre en los casos en los que se borre o modifique la clave primaria referenciada.

Relaciones binarias 1:n, con restricción de participación (integridad referencial).

- Las dos entidades dan lugar a dos relaciones y la relación R se representa con una clave ajena en la relación de lado de la n.

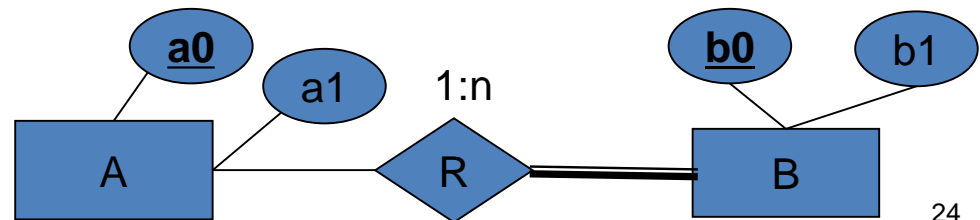
A(a0:dom_a0, a1:dom_a1, ...)

Clave Primaria: {a0}

B(b0:dom_b0, b1:dom_b1, ..., a0:dom_a0)

Clave Primaria: {b0}

Clave ajena: {a0} de B con A



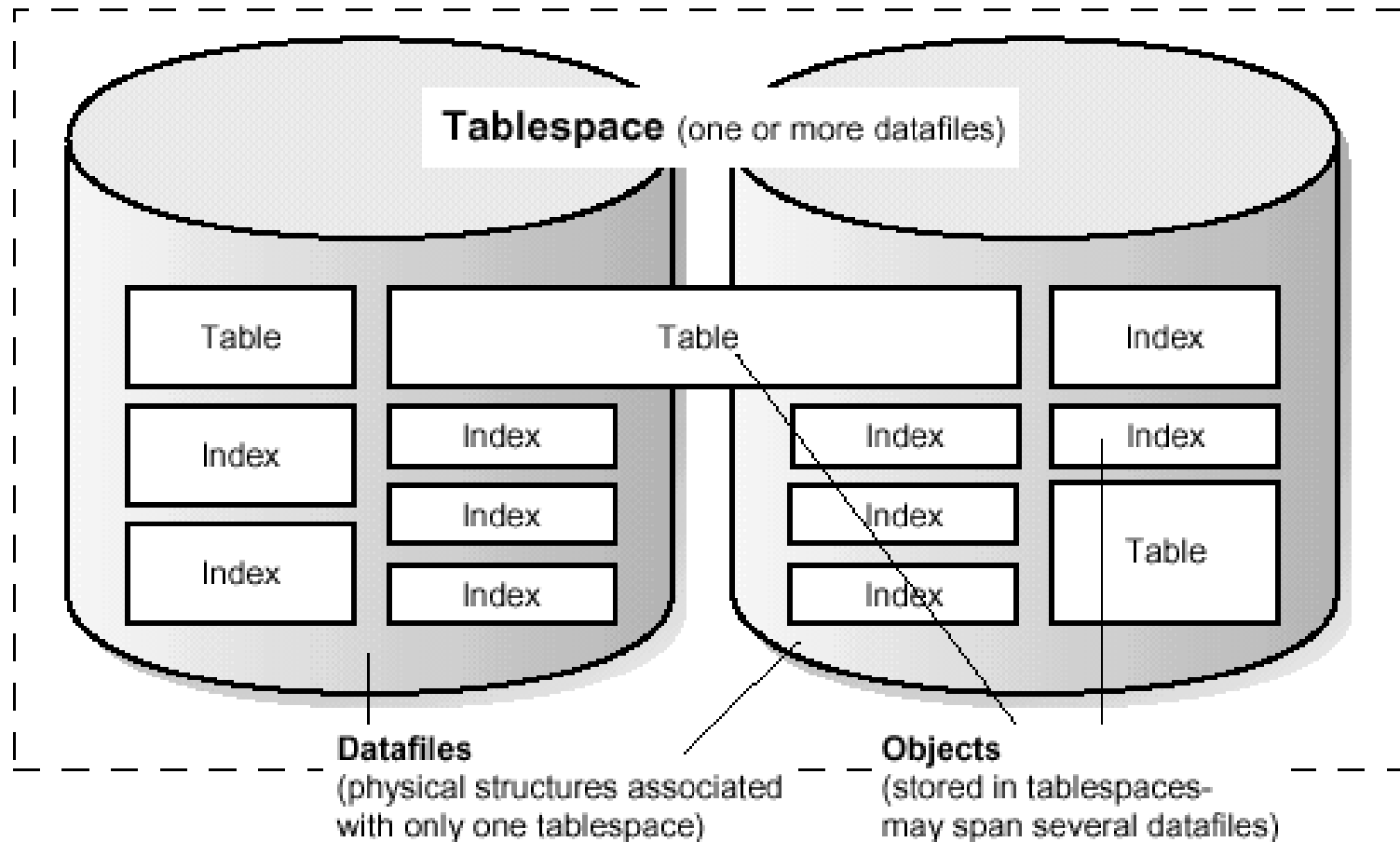
Introducción

- Especificación de estructuras de almacenamiento internas y caminos de acceso específicos para que las diversas aplicaciones que accedan a la BD tengan un buen rendimiento
- Cada SGBD ofrece varias opciones:
 - Diferentes tipos de ÍNDICES
 - Agrupamiento de registros (de distinto tipo) relacionados en los mismos bloques de disco (CLUSTER de ficheros)
 - Distintos tipos de técnicas de dispersión (HASHING)
 - Diferentes valores para los parámetros físicos (tamaño de bloque, de buffers, ...)
 - ...
- El diseño físico es muy dependiente del SGBD comercial seleccionado
- Una vez elegido el SGBD, el Diseño Físico consiste en la elección e implementación de las estructuras más apropiadas para los archivos de la BD, entre las opciones que ofrece el SGBD
- Diseñar e implementar los mecanismos de seguridad : vistas de usuario y reglas de acceso (privilegios/roles)

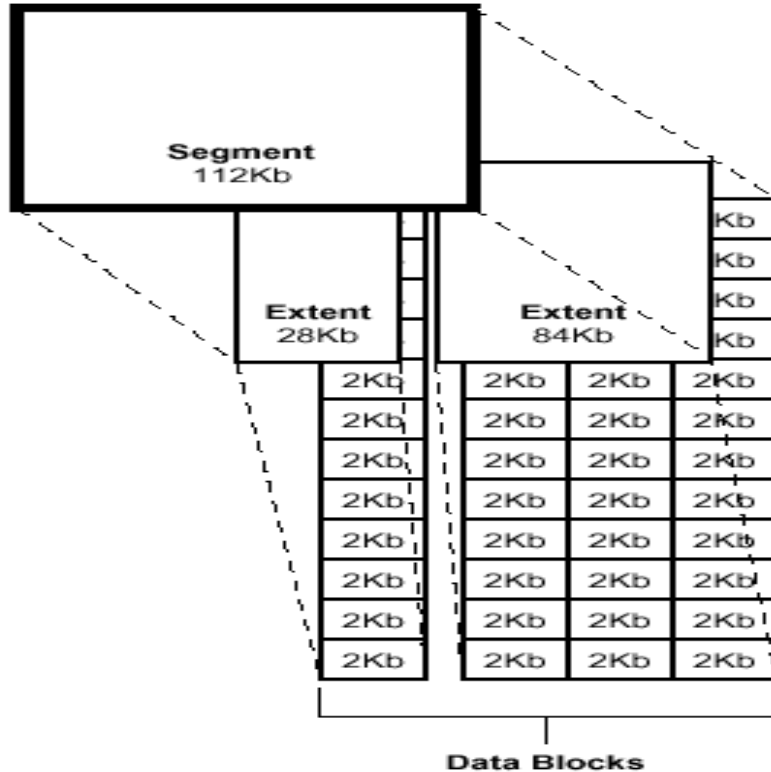
Objetivos

- Tiempo de respuesta (debe minimizarse)
 - Tiempo entre la introducción de una transacción de BD y la obtención de respuesta
 - Depende de...
 - Tiempo de acceso a la base de datos (bajo el control del SGBD) para obtener los datos que T necesita
 - Carga del sistema, planificación de tareas del SO, retrasos de comunicación (fuera del control del SGBD)
- Aprovechamiento del espacio (debe optimizarse)
 - Cantidad de espacio ocupado por archivos de la BD y sus estructuras de acceso
- Productividad de las transacciones (debe maximizarse)
 - Número promedio de transacciones que el SBD puede procesar por minuto
 - Parámetro crítico de los sistemas de procesamiento masivo de transacciones
 - Debe medirse en “condiciones pico” del sistema

Estructura lógica de una BD (Oracle)

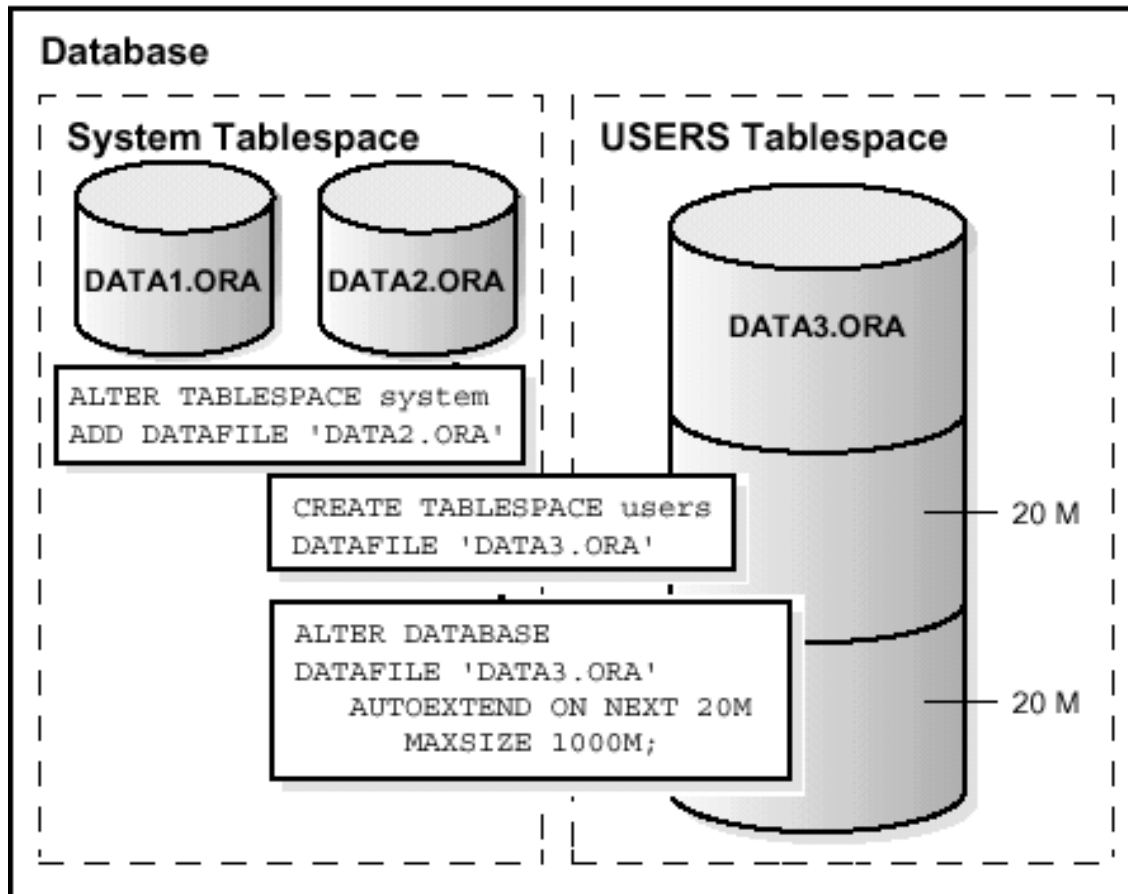


Estructuras lógicas de almacenamiento



- Segmento datos
- Segmento de índices
- Segmento de rollback
- Segmento temporal
 - SELECT ... ORDER BY...
 - CREATE INDEX.
 - SELECT ... GROUP BY...
 - SELECT ... UNION ...
 - SELECT DISTINCT ...
 - SELECT ... INSERT ...
 - SELECT ... MINUS ...

Estructura lógica de una BD (Oracle)



- Ficheros de datos
- Ficheros redo log
- Ficheros de control

¿Qué es **normalización**?

- Técnica formal de análisis y organización de datos; trata de evitar la redundancia y anomalías de actualización.
 - Introduce formalización en el diseño lógico de BDR. Además permite mecanizar parte del proceso al disponer de instrumentos algorítmicos de ayuda al diseño.
 - Proceso de normalización: disminuye las anomalías de actualización, pero penaliza las consultas (combinación consume muchos recursos).

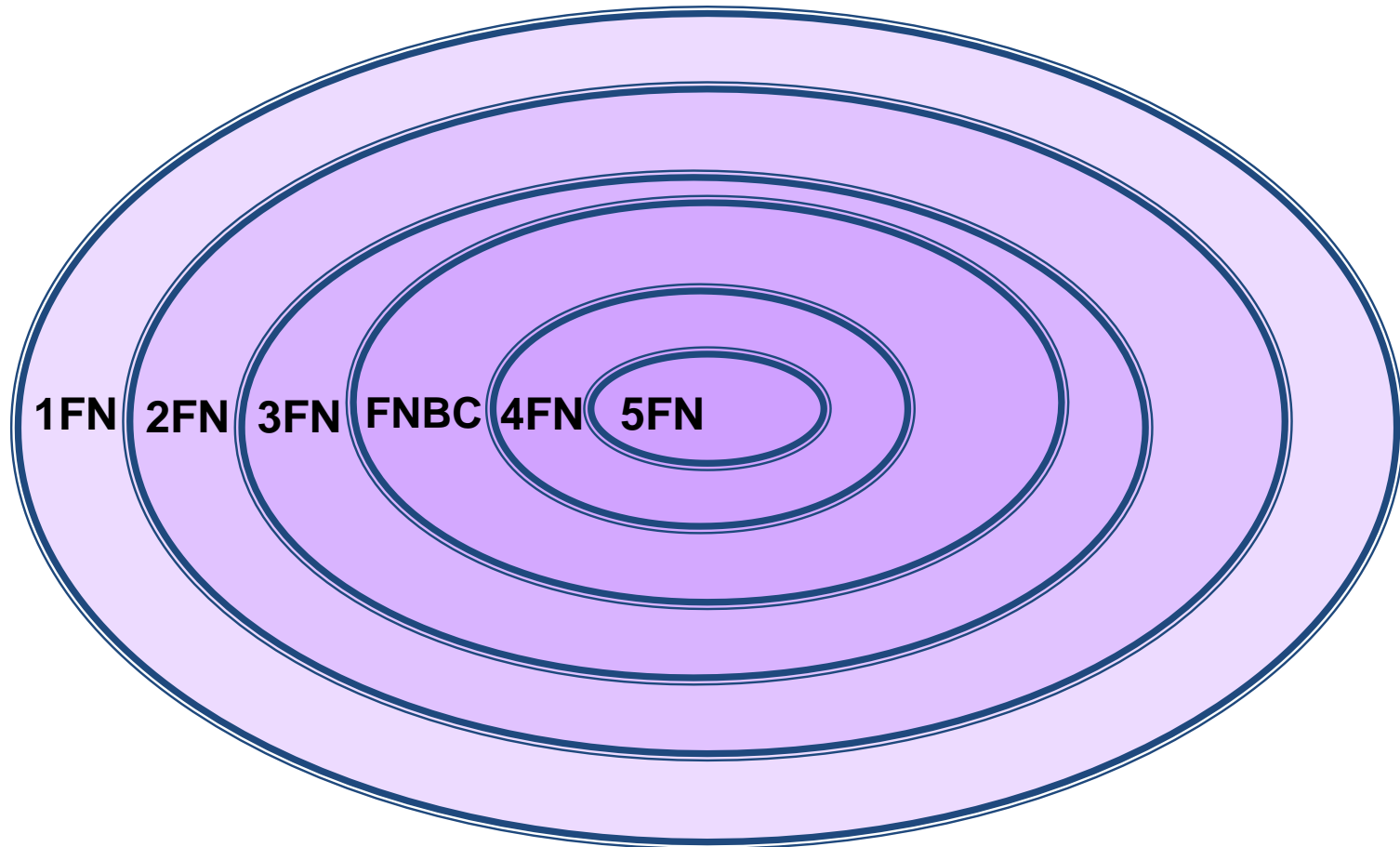
Pedidos

| <u>Artículo</u> | <u>cliente</u> | cantidad | precio | ciudad | distancia |
|-----------------|----------------|----------|--------|----------|-----------|
| A1 | C1 | 12 | 100 | Madrid | 400 |
| A1 | C2 | 30 | 100 | Valencia | 200 |
| A1 | C3 | 15 | 100 | Alicante | 80 |
| A2 | C1 | 35 | 250 | Madrid | 400 |
| A2 | C2 | 20 | 250 | Valencia | 200 |
| A2 | C4 | 10 | 250 | Madrid | 400 |
| A3 | C3 | 25 | 175 | Alicante | 80 |

Errores de Diseño:

- **Redundancia de información:** ciudad, distancia (ciudad); precio (artículo).
- **Anomalías de modificación:** !podemos tener el mismo artículo con dos precios! (igual argumento para ciudad y distancia). → **inconsistencias**
- **Anomalías de inserción:** ¿Podemos registrar nuevo artículo?, ¿Nuevo cliente?, ¿Nueva ciudad, distancia?
- **Anomalías de borrado:** Si eliminamos tupla de pedido de artículo A3 o cliente C4 → **pérdida de información.**

- Un esquema de relación está en una determinada forma normal si **satisface un determinado conjunto específico de restricciones** definidas sobre los atributos del esquema (dependencias).
- 1ª FN (Codd, 1970)
 - Concepto de relación normalizada.
- 2ª, 3ª FN (Codd, 1970), FNBC (Boyce/Codd, 1974)
 - Basadas en análisis de dependencias funcionales.
- 4ª FN. Fagin, 1977
 - Basada en análisis de dependencias multivaluadas.
- 5ª FN. Fagin, 1979
 - Basada en análisis de dependencias de proyección / combinación.



- **Primera Forma Normal:**

Una tabla está en Primera Forma Normal si:

1. Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.
2. La tabla contiene una clave primaria única.
3. La clave primaria no contiene atributos nulos.
4. No debe existir variación en el número de columnas.
5. Los Campos no clave deben identificarse por la clave
6. Debe Existir una independencia del orden tanto de las filas como de las columnas, es decir, si los datos cambian de orden no deben cambiar sus significados
7. Esta forma normal elimina los valores repetidos dentro de una Base de Datos.

- **Segunda Forma Normal:**

Una tabla 1NF está en 2NF si y solo si ninguno de sus atributos no-principales son funcionalmente dependientes en una parte de una clave primaria. **(dependencia funcional)**

Una dependencia funcional es una conexión entre uno o más atributos. Por ejemplo si se conoce el valor de DNI tiene una conexión con Apellido o Nombre .

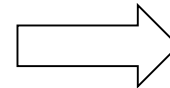
Las dependencias funcionales del sistema se escriben utilizando una flecha, de la siguiente manera:

FechaDeNacimiento → Edad.



Tabla: Habilidades de los empleados

| <u>Empleado</u> | <u>Habilidad</u> | Lugar actual de trabajo |
|-----------------|------------------|-------------------------|
| Jones | Mecanografía | 114 Main Street |
| Jones | Taquigrafía | 114 Main Street |
| Jones | Tallado | 114 Main Street |
| Bravo | Limpieza ligera | 73 Industrial Way |
| Ellis | Alquimia | 73 Industrial Way |
| Ellis | Malabarismo | 73 Industrial Way |
| Harrison | Limpieza ligera | 73 Industrial Way |



| <u>Empleado</u> | Lugar actual de trabajo |
|-----------------|-------------------------|
| Jones | 114 Main Street |
| Bravo | 73 Industrial Way |
| Ellis | 73 Industrial Way |
| Harrison | 73 Industrial Way |

| <u>Empleado</u> | <u>Habilidad</u> |
|-----------------|------------------|
| Jones | Mecanografía |
| Jones | Taquigrafía |
| Jones | Tallado |
| Bravo | Limpieza ligera |
| Ellis | Alquimia |
| Ellis | Malabarismo |
| Harrison | Limpieza ligera |

La única clave candidata de la tabla es {Empleado, Habilidad}.

El atributo restante, *Lugar actual de trabajo*, es dependiente solo en parte de la clave candidata, llamada *Empleado*. Por lo tanto la tabla no está en 2NF.

- **Tercera Forma Normal:**

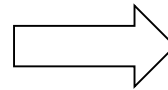
La tabla se encuentra en 3FN si es 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

Dependencia funcional transitiva: Sean X , Y , Z tres atributos (o grupos de atributos) de la misma entidad. Si Y depende funcionalmente de X y Z de Y , pero X no depende funcionalmente de Y , se dice entonces que Z depende transitivamente de X .



Tabla: Ganadores del torneo

| <u>Torneo</u> | <u>Año</u> | Ganador | Fecha de nacimiento del ganador |
|----------------------|------------|----------------|---------------------------------|
| Indiana Invitational | 1998 | Al Fredrickson | 21 de julio de 1975 |
| Cleveland Open | 1999 | Bob Albertson | 28 de septiembre de 1968 |
| Des Moines Masters | 1999 | Al Fredrickson | 21 de julio de 1975 |
| Indiana Invitational | 1999 | Chip Masterson | 14 de marzo de 1977 |



| <u>Torneo</u> | <u>Año</u> | Ganador |
|----------------------|------------|----------------|
| Indiana Invitational | 1998 | Al Fredrickson |
| Cleveland Open | 1999 | Bob Albertson |
| Des Moines Masters | 1999 | Al Fredrickson |
| Indiana Invitational | 1999 | Chip Masterson |

| <u>Ganador</u> | Fecha de nacimiento |
|----------------|--------------------------|
| Chip Masterson | 14 de marzo de 1977 |
| Al Fredrickson | 21 de julio de 1975 |
| Bob Albertson | 28 de septiembre de 1968 |

La violación de la 3NF ocurre porque el atributo no primario Fecha de nacimiento del ganador es dependiente transitivamente de {Torneo, Año} vía el atributo no primario Ganador. El hecho de que la Fecha de nacimiento del ganador es funcionalmente dependiente en el Ganador hace la tabla vulnerable a inconsistencias lógicas, pues no hay nada que impida a la misma persona ser mostrada con diferentes fechas de nacimiento en diversos registros.

- **Forma Normal de Boyce-Codd:**

Es una versión ligeramente más fuerte de la Tercera forma normal (3FN). La forma normal de Boyce-Codd requiere que no existan dependencias funcionales no triviales de los atributos que no sean un conjunto de la clave candidata.

En una tabla en 3FN, todos los atributos dependen de una clave, de la clave completa y de ninguna otra cosa excepto de la clave (excluyendo dependencias triviales, como $A \rightarrow A$).

Se dice que una tabla está en FNBC si y solo si está en 3FN y **cada dependencia funcional no trivial tiene una clave candidata como determinante**.

Solamente en casos raros una tabla en 3NF no satisface los requerimientos de la FNBC. Un ejemplo de tal tabla es (teniendo en cuenta que cada estudiante puede tener más de un tutor):

| ID Tutor | Número de seguro social del tutor | ID Estudiante |
|----------|-----------------------------------|---------------|
| 1078 | 088-51-0074 | 31850 |
| 1078 | 088-51-0074 | 37921 |
| 1293 | 096-77-4146 | 46224 |
| 1480 | 072-21-2223 | 31850 |

El propósito de la tabla es mostrar qué tutores están asignados a qué estudiantes.

Las claves candidatas de la tabla son:

{ID Tutor, ID Estudiante}

{Número de seguro social del tutor, ID Estudiante}

Denormalización

Una BDD compleja con un excesivo volumen de normalización puede acarrear un mal rendimiento en la manipulación, por ejemplo, de joins con gran cantidad de datos. Se realizan dos estrategias:

- Mantener el modelo lógico añadiendo una capa con los datos almacenados en un formato más manipulable: las vistas.
- Denormalizar el modelo lógico, creando reglas en el motor de la base de datos (restricciones, triggers) que permitan mantener la integridad. Puede provocar una pérdida de rendimiento en las operaciones de modificación de datos pero mejorará el rendimiento en la consulta.

Un modelo de datos denormalizado **no es lo mismo** que un modelo de datos que no ha sido normalizado, y la denormalización debe tomar lugar solamente **después** de que haya ocurrido un nivel satisfactorio de normalización y de que hayan sido creadas las restricciones y/o reglas requeridas para ocuparse de las anomalías inherentes en el diseño

Denormalización

«Desnormalizar, es decir, violar la normalización, sólo tiene una excusa: rendimiento ... y sólo en algunas situaciones » [Shasha-92]

La normalización hasta la 3FN, FNBC o superior, facilita la comprensión de los datos y relaciones entre los datos, que deben protegerse y mantenerse al crear las aplicaciones.

DESNORMALIZACIÓN REAL

- Consultas o Transacciones frecuentes que necesitan reunión de demasiadas tablas (tres o más), o dos tablas con muchas tuplas se ejecutarán muy lentamente
- Solución: Romper la 3FN (FNBC o superior)
- Almacenar atributos de una tabla en otra tabla (redundancia de datos)
- Importante: las transacciones que actualizan el atributo deben MANTENER LA CONSISTENCIA entre los duplicados (evitar anomalías)

DESNORMALIZACIÓN EXTREMA

- Almacenar un FICHERO con la REUNIÓN de las tablas → Resolver explícitamente las ANOMALÍAS DE ACTUALIZACIÓN

- **Consideraciones**

- ✓ Un correcto diseño es fundamental para asegurar la mantenibilidad de una base de datos, su escalabilidad y su rendimiento.
- ✓ Es muy importante concebir el diseño desde su fase inicial: con la base de datos ya en funcionamiento será mucho más difícil adaptarla que desde un inicio.
- ✓ No siempre es correcto aplicar el mayor grado de normalización...hay que encontrar un equilibrio. Es común llegar a niveles similares a 3FN o FNBC.
- ✓ Se suele aplicar un grado de normalización selectivo sobre cada tabla de la base de datos, adaptándolo a las necesidades concretas de cada una.

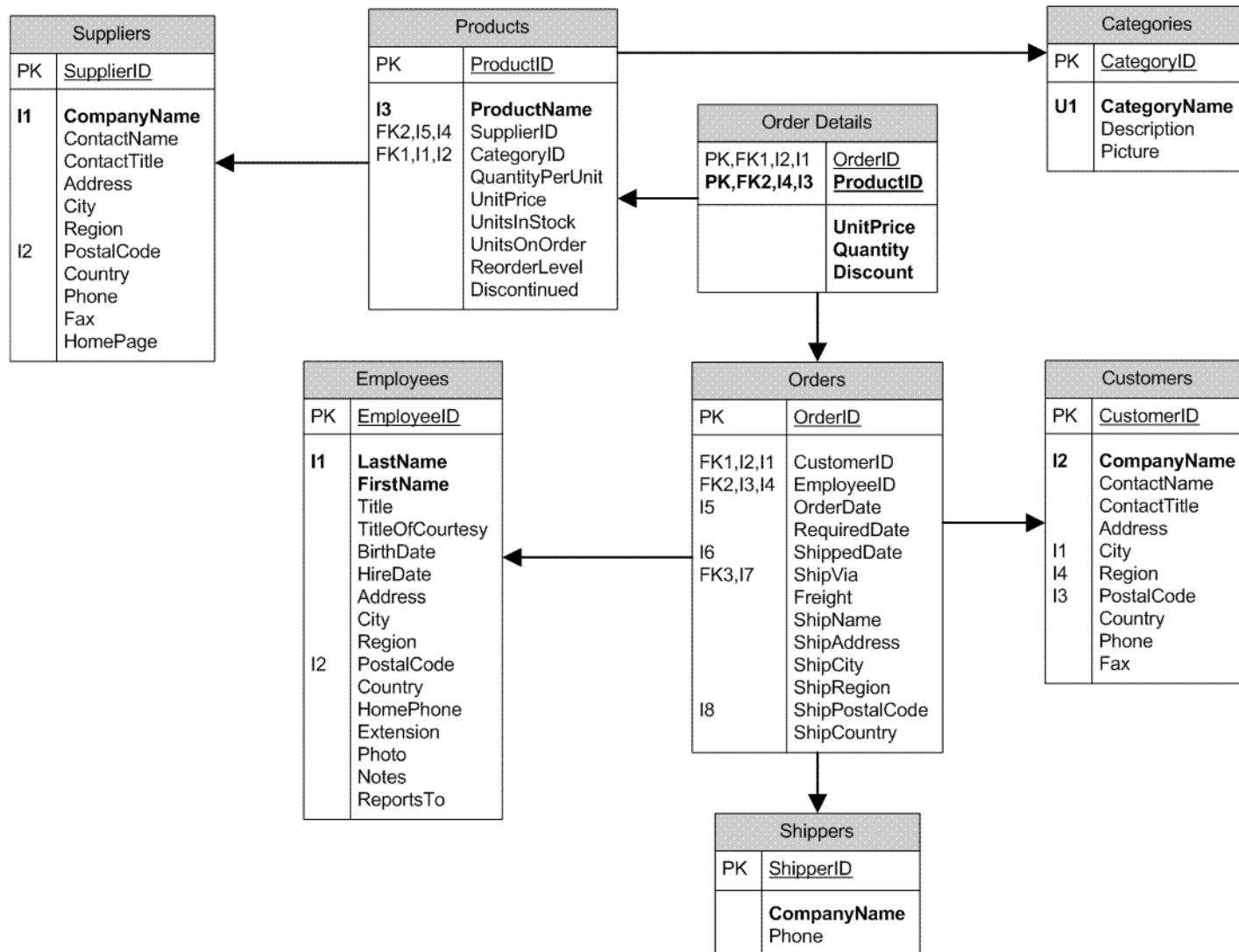
UML

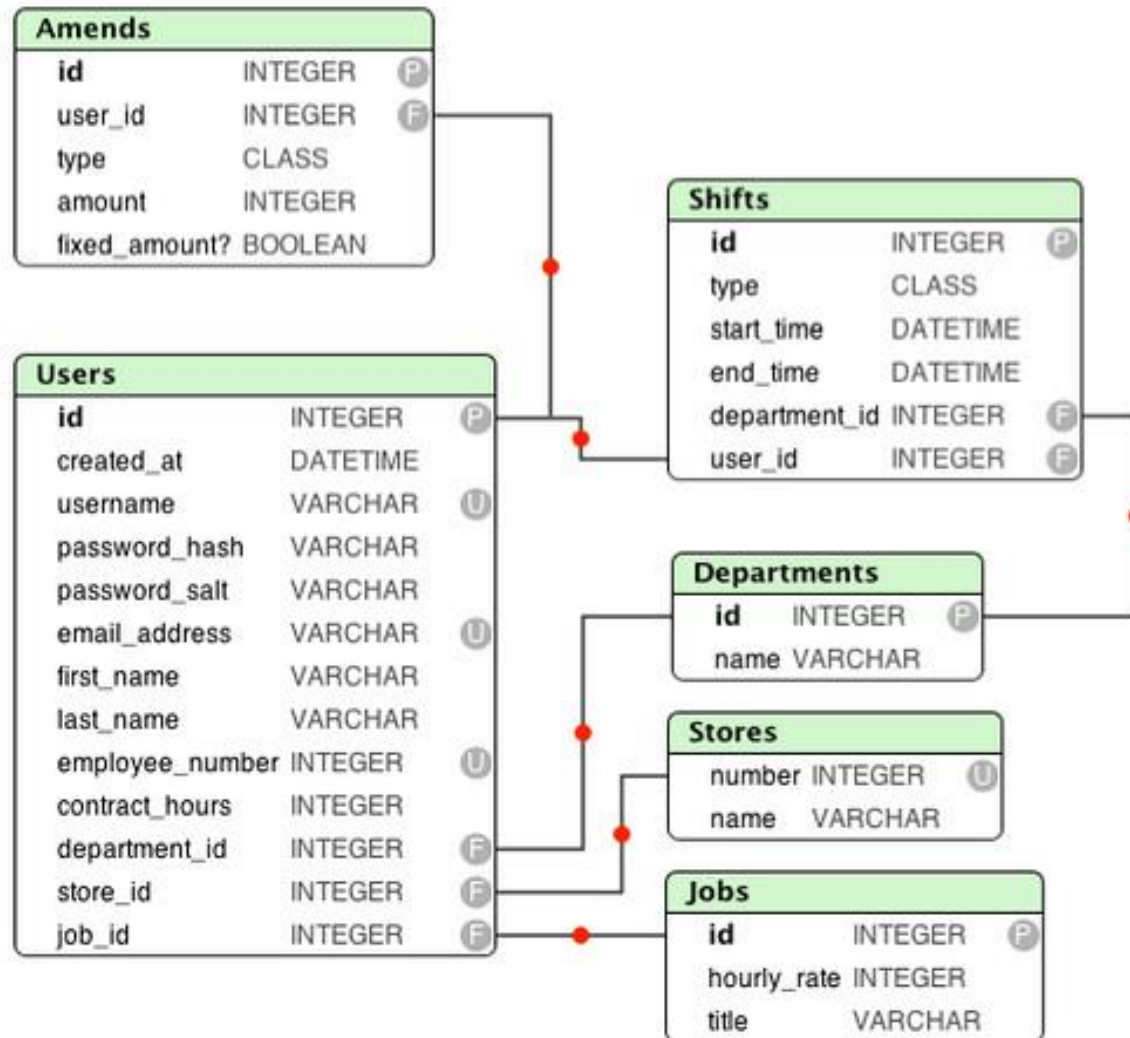
- Lenguaje Unificado de Modelado: estándar en el mercado para modelado de software.
- Incluye herramientas de modelado para software, redes, definición de objetos, procesos y secuencias y también los referentes a las bases de datos y modelo ER.
- La definición se realiza de un modo gráfico altamente intuitivo.
- Gracias a múltiples herramientas comerciales y gratuitas, incluso permite la traducción de los campos definidos
- Aunque el diagrama entidad – relación no forma parte de esta especificación, si se integra de forma conceptual como parte de la misma.

Diagrama Relacional

- Diagrama que representa el modelo entidad – relacion de una base de datos, de forma que se pueda plasmar el modelo conceptual de una forma gráfica.
- En ella definimos las entidades en unos objetos cuyos sub-elementos son:
 - Relaciones (FKs, pero también PKs y Uks)
 - Atributos (que se suelen complementar con el tipo de datos utilizado)
 - Por lo tanto, incluye elementos del modelo lógico y modelo físico

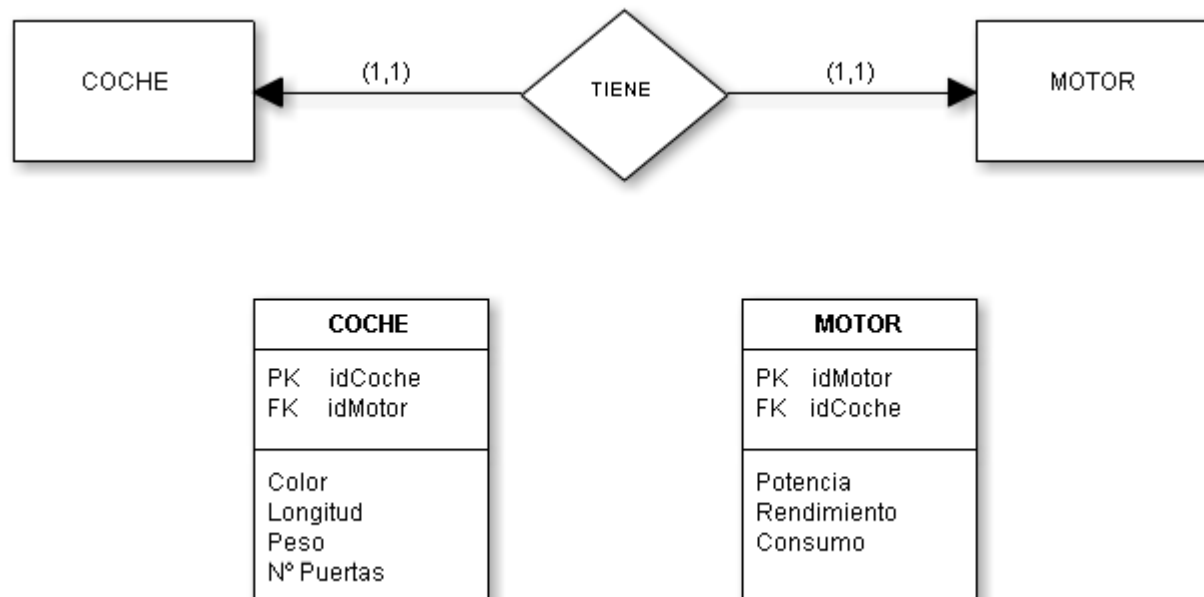
| VAGÓN |
|-------------------------------|
| PK,Fk idVagon PK,Fk idTren |
| Peso Longitud Asientos |





Ejemplos de transformaciones

- Relación 1:1



Ejemplos de transformaciones

- Relación 1: N

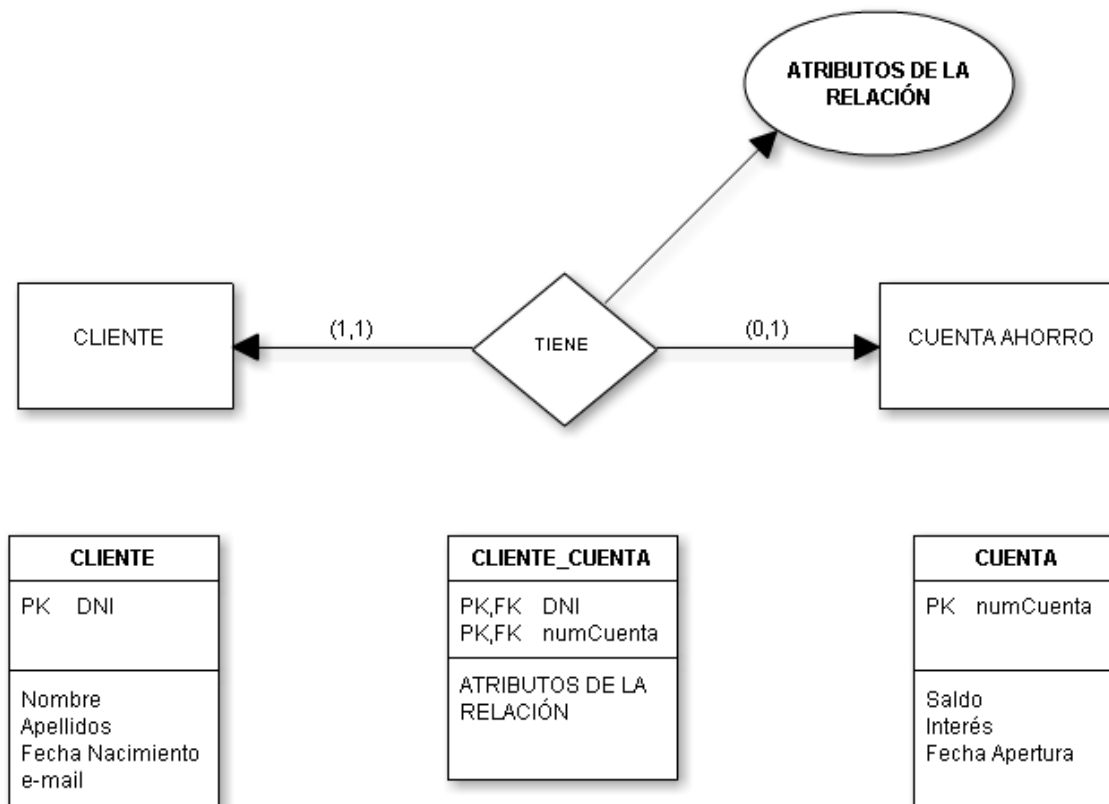


| PERIODISTA | |
|--|-------------|
| PK | DNI |
| FK | idPeriodico |
| Nombre Domicilio Fiscal Dirección Facturación | |

| PERIÓDICO | |
|-----------------|-------------|
| PK | idPeriodico |
| Nombre Fecha | |

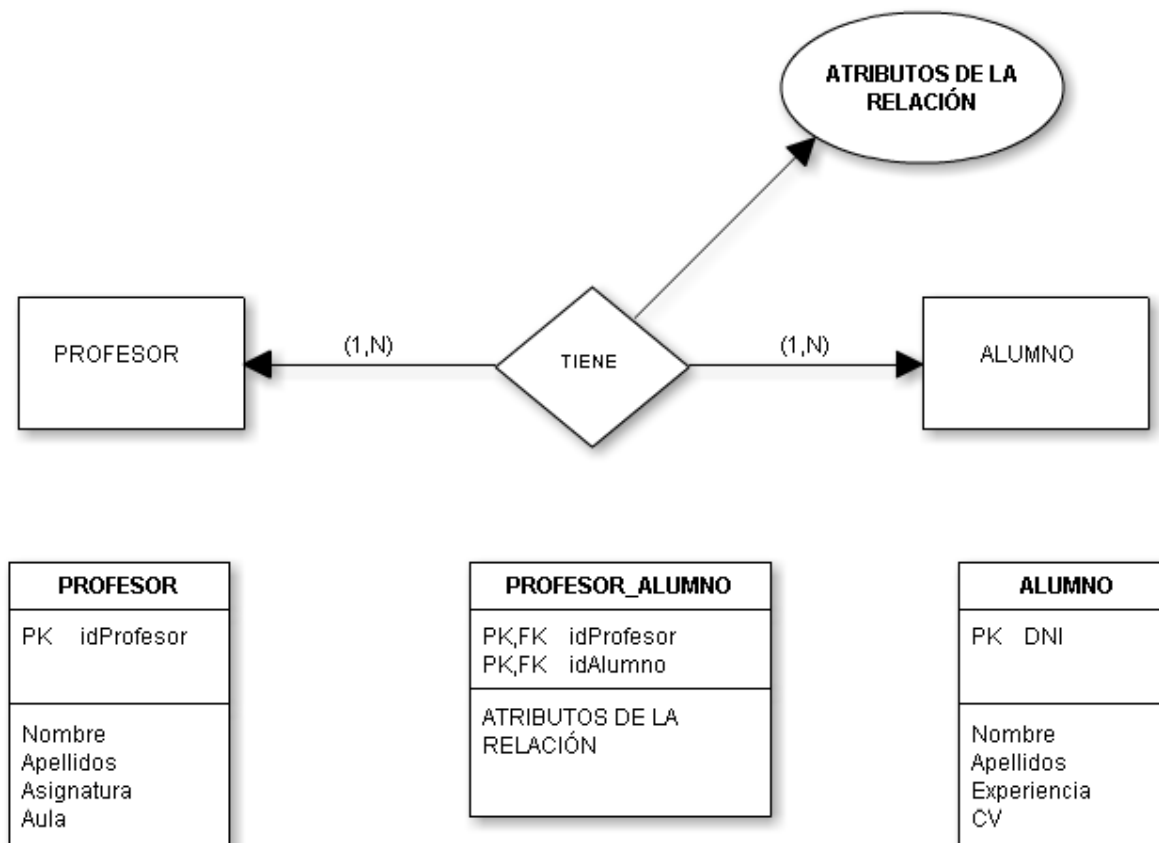
Ejemplos de transformaciones

- Cardinalidad mínima 0



Ejemplos de transformaciones

- Relación N:N



Ejemplos de transformaciones

- Relación reflexiva

