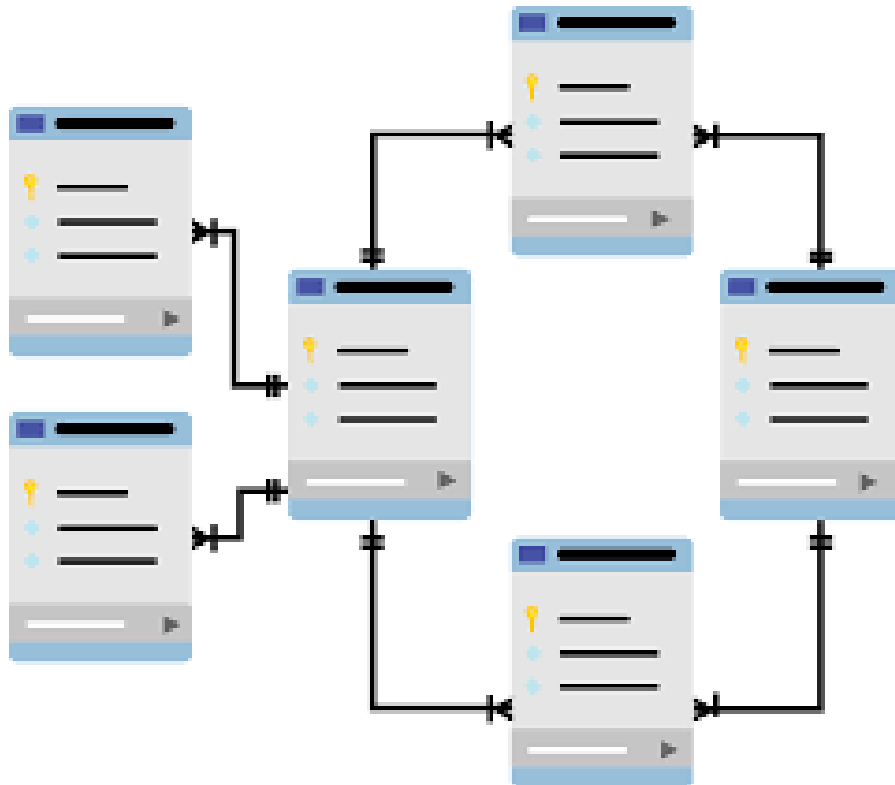


Base de Datos

Unidad 2 – Diseño lógico relacional



Índice

BLOQUE I: Modelo Conceptual.....	4
1. Introducción.....	4
2. Almacenamiento de la información.....	4
2.1. Entidad.....	4
2.2. Atributos.....	5
2.3. Claves.....	5
2.4. Representación gráfica.....	6
2.5. Dominio de un atributo.....	6
2.6. Relaciones.....	7
3. Modelo E/R Extendido.....	7
3.1. Restricciones en las relaciones.....	8
3.1.1. Exclusividad.....	8
3.1.2. Exclusión.....	8
3.1.3. Inclusividad.....	8
3.1.4. Inclusión.....	8
3.2. Jerarquías de generalización / especialización.....	8
3.3. Entidad Asociativa y Agregación.....	9
3.3.1. Entidad asociativa.....	9
3.3.2. Agregación.....	10
4. Guía para la creación del Modelo E/R.....	10
4.1. Redundancia en diagramas E/R.....	10
4.2. Propiedades deseables del diagrama E/R.....	11
BLOQUE II: Modelo Relacional.....	12
5. Modelo relacional.....	12
5.1. Terminología relacionales.....	12
5.2. Restricciones en base de datos relacionales.....	13
5.2.1. Restricciones implícitas (inherentes).....	13
5.2.2. Restricciones semánticas (definidas por el usuario).....	13
6. Representación del modelo relacional.....	14
6.1. Textual y gráficamente.....	14
BLOQUE III: Transformación del Modelo Conceptual E/R al Modelo Lógico Relacional.....	15
7. Conversión del E/R al Relacional.....	15
7.1. Conversión del E/R al Relacional.....	15
7.2. Transformación de dominios.....	15

7.3. Transformación de entidades.....	15
7.4. Transformación de relaciones binarias.....	16
7.4.1. Relación N:M.....	16
7.4.2. Relación 1:N.....	16
7.4.3. Relación 1:1.....	16
7.4.4. Relaciones con entidades débiles.....	16
7.5. Transformaciones de relaciones ternarias.....	17
7.6. Relaciones reflexivas.....	17
7.7. Transformación de jerarquías en Generalización / Especialización.....	17
7.8. Transformación de dimensión temporal.....	18
7.9. Transformación de atributos derivados.....	18
8. Modelo relacional y los SGBD.....	18
BLOQUE IV: Normalización del Modelo Lógico Relacional.....	19
9. Normalización.....	19
9.1. Dependencia funcional simple.....	19
9.2. Dependencia funcional plena.....	19
9.3. Primera Forma Normal (1FN).....	20
9.4. Segunda Forma Normal (2FN).....	20
9.5. Tercera Forma Normal (3FN).....	20
10. Resumen de las formas normales.....	21

BLOQUE I: Modelo Conceptual

1. Introducción

Todo el desarrollo de un sistema de información comienza con el **análisis estructurado del problema**. Se busca **modelar la realidad** de forma precisa y simple para su implementación posterior. Se suele utilizar el **Modelo de Entidad-Relación** de Peter Chen como base del diseño conceptual.

2. Almacenamiento de la información

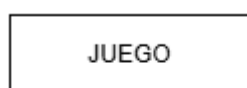
El objetivo de la **representación y definición de todos los datos de almacenamiento** que sean introducidos, almacenados, transformados y producidos dentro de un sistema de información sin tener en cuenta las necesidades de la tecnología existente ni otras restricciones.

2.1. Entidad

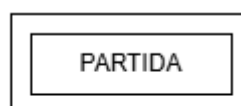
Representa un **objeto concreto o abstracto** del cual se desea guardar la información, como por ejemplo un coche, un cliente, una profesión, un departamento,...

Hay dos tipos de entidades:

1. **Fuerte:** tiene existencia propia, es decir por si sola se vale. Se representa con un rectángulo.
2. **Débil:** depende de otra entidad, es decir que debe existir otra para que esa entidad exista. Se representa con un doble rectángulo.



Entidad Fuerte



Entidad Débil

2.2. Atributos

Las entidades tienen cada unas **características que describen a una entidad o relación**.

Hay varios tipos de atributos, se clasifican en:

- **Compuesto:** pueden subdividirse, como una dirección (calle, ciudad).
- **Multivaluado:** pueden tener múltiples valores, como más de un número de teléfono.
- **Opcional:** puede estar vacío el campo.
- **Derivado:** se calcula a partir de otros valores, como la edad.
- **Clave:** se identifica como único, no puede ser nulo.

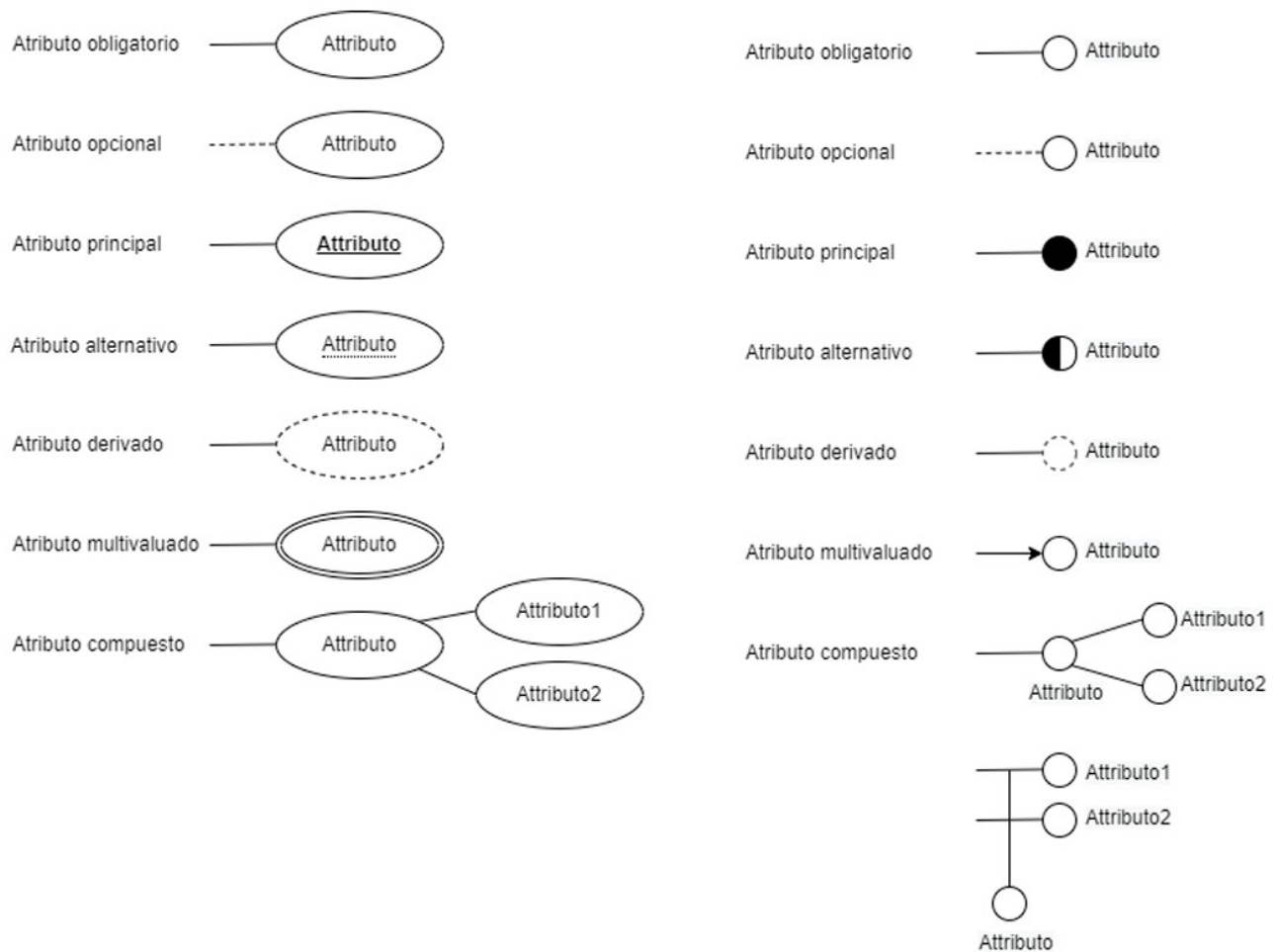
2.3. Claves

Toda entidad debe identificarse a través de una única clave, sin embargo es posible determinar según el atributo que tipo de clave es necesaria:

- **Clave candidata:** es posible que los identificadores sean únicos.
- **Clave principal:** es una clave seleccionada.
- **Clave alternativa:** no está seleccionada.
- **Entidades débiles:** necesitan tener una clave parcial y la clave principal de la entidad fuerte.

2.4. Representación gráfica

La **representación** varía según la notación elegida.



2.5. Dominio de un atributo

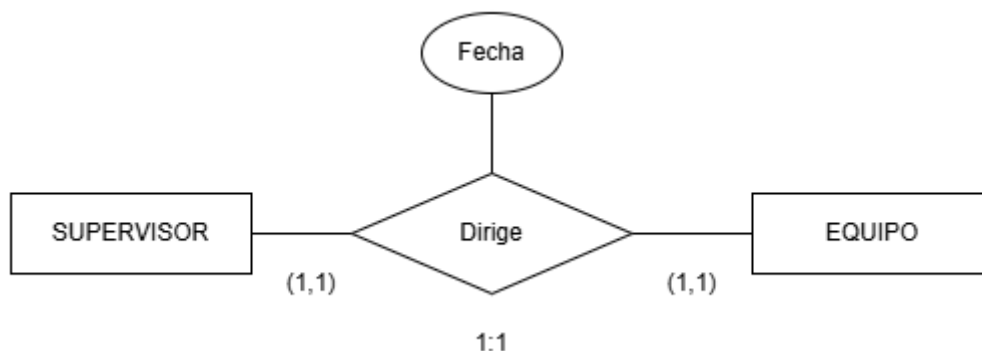
El conjunto de valores permitidos de un atributo se hace **referencia al tipo de datos con el que se almacenará** la fecha, el número, las ciudades,... Aunque no siempre es representado gráficamente, se suelen hacer notaciones específicas del contenido.

2.6. Relaciones

Según que relaciones asocian dependiendo de la entidad con la que participen.

Hay varios tipos de relaciones:

- **Grado:** es el número de entidades que participan en ella, que pueden ser binaria, ternaria o n-aria.
- **Cardinalidad:** número máximo de ocurrencias como son 1:1, 1:N o M:N.
- **Modalidad:** son el mínimo y máximo de ocurrencias en la relación.
- **Atributos de relación:** entre las entidades débiles y fuertes siempre de grado 2 y modalidad (1,1).



3. Modelo E/R Extendido

Ampliar la capacidad semántica del **Modelo E/R original de Peter Chen** para **representar realidades más complejas en el diseño** de bases de datos, usando nuevas estructuras como restricciones avanzadas, jerarquías y agregaciones.

3.1. Restricciones en las relaciones

3.1.1. Exclusividad

Una entidad solo puede participar en **una de varias relaciones excluyentes**. Por ejemplo, un profesor o imparte un curso o realiza una investigación, pero no ambos. Se representa con un arco que engloba las relaciones exclusivas.

3.1.2. Exclusión

Dos relaciones **no pueden cumplirse al mismo tiempo** entre dos entidades. Por ejemplo, un profesor no puede impartir y recibir un curso al mismo tiempo. Se representa con una línea discontinua entre relaciones.

3.1.3. Inclusividad

Participar en una relación implica **obligatoriamente participar en otra**. Por ejemplo: un jugador debe haber jugado en un equipo antes de entrenarlo. Se representa con una flecha arqueada desde la relación obligatoria hacia la dependiente.

3.1.4. Inclusión

Las ocurrencias en una relación deben **también estar en otra relación**. Por ejemplo, para que un profesor imparta un curso, debe haberlo recibido antes. Se representa con una flecha recta discontinua.

3.2. Jerarquías de generalización / especialización

Es un **supertipo** comparte atributos y relaciones con varios subtipos. Y los subtipos **heredan** del supertipo (no al revés).

- **Generalización:** se parte de varios tipos similares los cuales son agrupados en un supertipo.
- **Especialización:** se parte de un tipo general y se divide en subtipos más específicos.

Se representan con un **triángulo invertido** con líneas que conectan al supertipo y los subtipos. Si hay un **atributo discriminante**, también se conecta al triángulo.

Los tipos de restricciones semánticas son las siguientes:

1. Exclusividad vs Solapamiento

- **Exclusividad:** una instancia que solamente pertenece a un subtipo.
- **Solapada:** una instancia que puede pertenecer a varios subtipos.

2. Totalidad vs Parcialidad

- **Total:** toda instancia del supertipo debe pertenecer a un subtipo.
- **Parcial:** puede no pertenecer a ningún subtipo.

Combinaciones posibles respecto a las restricciones semánticas:

- Exclusiva / Parcial → Ej. documento puede ser libro o artículo (o ninguno).
- Exclusiva / Total → Ej. una persona es menor de edad o mayor de edad (sí o sí).
- Solapada / Parcial → Ej. un empleado puede ser docente, investigador o ninguno.
- Solapada / Total → Ej. una vivienda puede tener garaje, trastero o ambos.

3.3. Entidad Asociativa y Agregación

3.3.1. Entidad asociativa

Se crea una nueva entidad a partir de una relación entre otras entidades para asociarla con más entidades. Es de utilidad cuando no se puede relacionar directamente una relación con otra entidad.

Por ejemplo, una entrevista como entidad entre la empresa y el candidatos se relaciona luego con la oferta.

3.3.2. Agregación

Se permite representar las relaciones como parte de una entidad compuesta. Hay varios tipos:

- Composición: las partes distinta con roles diferentes, como un coche con motor, ruedas,...
- Colección: a partes iguales con el mismo rol, como un equipo con jugadores.

La cardinalidad siempre será (1,1). Se representará con un rombo pequeño junto a la entidad compuesta.

4. Guía para la creación del Modelo E/R

1. Identificar **entidades** y sus atributos (incluir claves y multivaluados).
2. Establecer **relaciones**, cardinalidades y modalidades.
3. Determinar **atributos de relaciones**.
4. Analizar posibles **jerarquías**.
5. Detectar **entidades débiles** y tipo de dependencia.
6. Identificar **restricciones entre relaciones** (exclusión, inclusión, etc.).
7. Analizar **relaciones de grado >2**: ¿pueden simplificarse?
8. Revisar **redundancias** y eliminarlas si es posible.

4.1. Redundancia en diagramas E/R

Los **problemas de pérdida** de rendimiento, espacio y consistencia. Los tipos de **atributos derivados** son aceptables si están marcados y calculados correctamente. Las **relaciones redundantes** pueden obtenerse la información desde otras relaciones.

4.2. Propiedades deseables del diagrama E/R

Un buen diagrama debe ser:

- **Completo:** todos los requerimientos están representados.
- **Correcto:** sintáctica (elementos bien usados) y semánticamente (significados correctos).
- **Mínimo:** sin redundancia innecesaria.
- **Simple:** fácil de entender.
- **Legible:** bien organizado visualmente.
- **Escalable:** adaptable a cambios futuros.

BLOQUE II: Modelo Relacional

5. Modelo relacional

Es una **representación lógica** de la información, basada en la teoría de conjuntos y lógica de predicados. Fue propuesto por **Edgar F.Codd** en 1970. Sustituye los **modelos anteriores** como el **jerárquico y de red** al **eliminar las dependencias entre los datos y los programas**.

5.1. Terminología relacionales

- **Relación:** es un conjunto de tuplas, que es equivalente a una tabla.
- **Tupla:** Fila de una tabla; conjunto de pares atributo-valor.
- **Atributo:** Columna de la tabla (nombre + dominio).
- **Dominio:** Conjunto de valores válidos para un atributo.
- **Grado:** Número de atributos (columnas).
- **Cardinalidad:** Número de tuplas (filas).

Para establecer un esquema visual del contenido anterior:

• Relación	Tabla
• Tupla	Fila o registro
• Cardinalidad	Número de filas
• Atributo	Columna o campo
• Grado	Número de columnas
• Dominio	Conjunto de valores permitidos

5.2. Restricciones en base de datos relacionales

5.2.1. Restricciones implícitas (inherentes)

Están derivadas del modelo relacional, lo que significa que:

- No hay tuplas duplicadas.
- La **clave primaria** debe ser única y no nula.
- Cada atributo tiene un solo valor, es decir que no se permiten grupos repetitivos.
- El orden es irrelevante entre las tuplas y los atributos.
- El nombre de los atributos son únicos dentro de la relación.

5.2.2. Restricciones semánticas (definidas por el usuario)

- **UNIQUE:** es un atributo o conjunto que no puede repetirse el valor.
- **NOT NULL:** el atributo debe tener obligatoriamente un valor.
- **PRIMARY KEY:** el identificador es único en una tupla, no nulo, ni repetido.
- **FOREIGN KEY:** el atributo hace referencia a una clave primaria en otra relación. En la que permitir enlazar tablas. Además no es necesario que tenga el mismo nombre, pero si el mismo dominio.
- La integridad es referencial a la que asegura valores de la clave foránea existente en la tabla referenciada, las acciones posibles son las siguientes:
 - **CASCADE:** permite borrar o actualizar en cascada.
 - **SET NULL:** establece el valor en NULL.
 - **SET DEFAULT:** establece por defecto un valor.

- **NO ACTION / RESTRICT:** prohíbe la operación si hay referencias.
- **CHECK:** es la condición que debe cumplir el atributo.
- **ASSERTION:** es como CHECK, pero incluye a más de una tabla.
- **TIGGER (DISPARADOR):** ejecuta las acciones automáticas si cumple una condición.

6. Representación del modelo relacional

6.1. Textual y gráficamente

Se listan las relaciones con sus **atributos y claves**. Al lado de los atributos se suelen poner las etiquetas según sean. La **clave primaria [pk]**, **clave foránea [fk]** o **clave alternativa [ak]**.

Además usan etiquetas de manera **gráfica** para determinadas restricciones:

- | | | |
|-----------------------------|-------|----|
| • Clave primaria | | PK |
| • Clave alternativa | | AK |
| • Clave foránea | | FK |
| • Clave primaria ajena | | PF |
| • Clave alternativa foránea | | AK |
| • Obligatorio (NOT NULL) | | * |

BLOQUE III: Transformación del Modelo Conceptual E/R al Modelo Lógico Relacional

7. Conversión del E/R al Relacional

Convertir un modelo conceptual E/R en un modelo lógico relacional, adecuado para ser implementado en un Sistema de Gestión de Bases de Datos (SGBD) relacional.

La transformación se realiza aplicando reglas sistemáticas según el tipo de entidad, relación, cardinalidad y modalidad.

7.1. Conversión del E/R al Relacional

Los **atributos compuestos** deberán eliminarse **dividiéndolos en atributos** en atributos **simples** como dirección → calle, ciudad,...

Los atributos multievaluados serán los valores posibles **se descomponen en varios atributos simples**. Si son muchos y desconocidos, **se crea una nueva entidad y una relación asociada**.

7.2. Transformación de dominios

Los dominios del modelo E/R se conservan en el modelo relacional. Se definen por extensión o por tipo de dato.

7.3. Transformación de entidades

Cada **entidad** se transforma en una **tabla** de la relación la cual dispone de varios tipos de atributos, como son la **clave principal** – PRIMARY KEY, **clave alternativa** – UNIQUE + NOT NULL y **otros atributos**, los cuales pueden ser opcionales NULL u obligatorios NOT NULL.

7.4. Transformación de relaciones binarias

7.4.1. Relación N:M

Se crea una nueva **tabla intermedia** con:

- Las claves de ambas entidades como **clave primaria compuesta**.
- Los atributos de la relación, si los hay.

7.4.2. Relación 1:N

Se pueden transformar de dos formas:

- **Propagación de clave:** la entidad del lado N recibe la clave del lado 1 como clave foránea.
- **Nueva tabla intermedia:** si hay muchos valores nulos al propagar, se prefiere crear una relación nueva.

7.4.3. Relación 1:1

- **Ambas entidades con modalidad (0,1):** se crea una tabla nueva.
- **Una entidad con (1,1):** se propaga la clave como FK + índice único.
- **Ambas con (1,1):** se puede propagar clave de una a otra, crear una tabla nueva o unificar ambas tablas.

7.4.4. Relaciones con entidades débiles

- Se propaga la clave de la entidad fuerte.

- La FK **no permite valores nulos** y se aplica **CASCADE** en **modificaciones y borrados**.
- La **clave primaria** incluye la clave de la entidad fuerte + clave parcial.

7.5. Transformaciones de relaciones ternarias

Todas se convierten en una **nueva tabla** que contienen:

- Claves de las entidades participantes.
- Atributos de la relación.

Las **variaciones de las cardinalidades** son:

- **N:M:P** → clave primaria = suma de todas las claves.
- **N:M:1** → clave primaria = las dos entidades con cardinalidad N.
- **N:1:1** → clave primaria = entidad con cardinalidad N + una de las otras.
- **1:1:1** → tres claves candidatas; se elige una como clave primaria y las otras como alternativas.

7.6. Relaciones reflexivas

Se transforman igual que las relaciones binarias, pero cambiando el nombre de los atributos propagados para evitar la duplicidad de los nombres.

7.7. Transformación de jerarquías en Generalización / Especialización

Hay 3 formas de **transformar las jerarquías**:

1. **Eliminación de los subtipos:** conservando solo el supertipo. Trasladando los atributos y las relaciones de los subtipos al supertipo.
2. **Mantenimiento del supertipo y los subtipos:** se crea una clave en la **relación de 1:1** entre cada subtipo y el supertipo. La clave del supertipo se propaga al subtipo como **clave foránea** (fk) y puede ser como **clave primaria** (pk).
3. **Eliminación el supertipo:** cada subtipo se transforma en una tabla. Se duplican los atributos comunes del supertipo en cada subtipo.

7.8. Transformación de dimensión temporal

Cuando la relación tiene un atributo temporal multievaluado, el atributo temporal se convierte en parte de la clave primaria. Con el tiempo se modela como una entidad y la relación se vuelve ternaria.

7.9. Transformación de atributos derivados

Hay dos alternativas para tratar a los atributos. Una de ellas es almacenar en una tabla en un disparador (trigger) o procedimiento para mantener su valor actualizado. Otra forma es no almacenarlo y calcularlo cuando se necesite, (más limpio, pero menos eficiente si se usa con frecuencia).

8. Modelo relacional y los SGBD

Almacenan algunas restricciones del Modelo E/R al modelo relacional. Se deben implementar mediante restricciones simples, entre varias tablas o acciones automáticas. También se aplican técnicas de aplicaciones de cardinalidades mínimas o máximas, exclusividades y de control de atributos derivados o multievaluados.

BLOQUE IV: Normalización del Modelo Lógico Relacional

Evitar **redundancias, anomalías y problemas de actualización** en una base de datos relacional, a través de un proceso estructurado, es denominado **normalización**.

Una base de datos que esté **mal normalizada puede tener**:

- Repetición de información.
- Inconsistencias en los datos.
- Problemas al insertar, modificar o borrar información.

9. Normalización

La **normalización** es un proceso de análisis y descomposición de relaciones complejas en las relaciones más simples. Se basa en el concepto de dependencias funcionales. Cada paso en la normalización se busca la estructura sin pérdida de información.

9.1. Dependencia funcional simple

La **dependencia funcional simple**, es cuando un **atributo B depende funcionalmente de A** si, para cada valor de A, hay un único valor de B. Al representarlo en notación $A \rightarrow B$.

Por ejemplo, si $DNI \rightarrow \text{nombre}$, cada DNI le corresponde un **solo nombre**.

9.2. Dependencia funcional plena

Una **dependencia plena** significa que el atributo depende de **todos** los atributos que forman parte la clave. Si dependen solo de parte de la clave, se llama **dependencia parcial**, por lo tanto debe eliminarse para alcanzar la 2FN.

9.3. Primera Forma Normal (1FN)

- Todos los atributos deben contener **valores atómicos**, invisibles.
- No se permiten **grupos repetitivos** ni listas de valores en un solo campo.
- Se **eliminan** los **atributos mutievaluados o compuestos**.
- La relación es estructurada con **celdas únicas y simples**.

9.4. Segunda Forma Normal (2FN)

- Debe estar en 1FN.
- **Elimina las dependencias parciales** donde todo atributo debe depender de la **clave completa**.
- Se **divide la tabla si hay atributos** que dependen **solo** de una **parte de la clave compuesta**.
- La relación es **más limpia y sin redundancia** basada en claves parciales.

9.5. Tercera Forma Normal (3FN)

- Debe estar en 2FN.
- **Elimina las dependencias transitivas**, ocurre cuando $A \rightarrow B$ y $B \rightarrow C$ entonces $A \rightarrow C$, de forma indirecta.
- Se separan los atributos que dependen de otro atributo **no clave**.
- Cada atributo no clave depende **únicamente de la clave primaria**, no de otro atributo.

10. Resumen de las formas normales

Forma	Requisitos principales	Qué elimina
1FN	Atributos atómicos	Repeticiones y agrupaciones
2FN	1FN + dependencias clave	Dependencias parciales
3FN	2FN + eliminación de dependencias transitivas	Dependencias entre no claves

Realizando la **normalización** se consiguen varios **beneficios**:

- Eliminar **redundancias innecesarias**.
- Aumentar la **coherencia** y **materialidad** de la base de datos.
- Reducir errores en inserciones, actualizaciones y eliminaciones de **anomalías**.
- Mejorar el rendimiento a largo plazo, aunque puede **generar más tablas** y requerir **más uniones (JOIN)**.