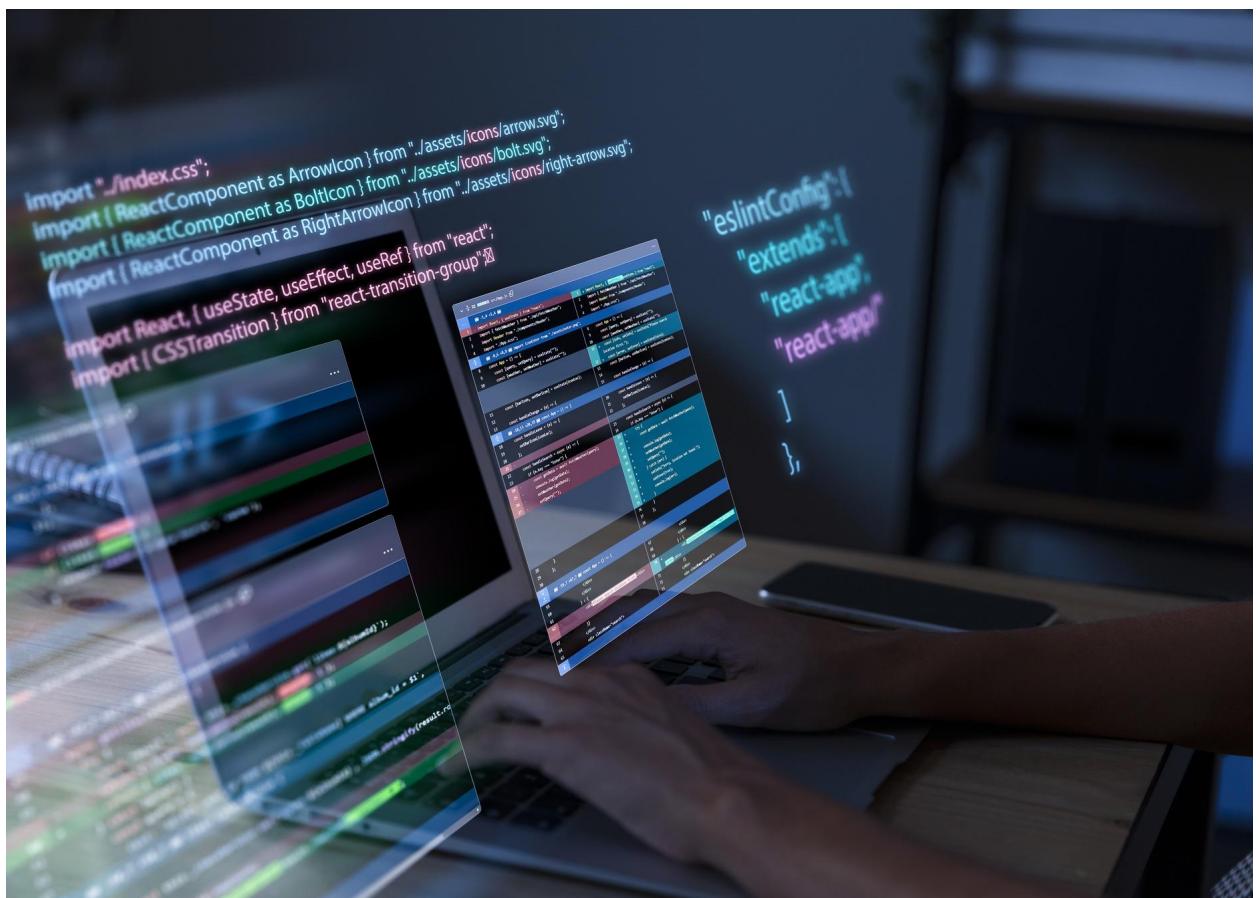


UNIDAD 3

Diseño lógico relacional



Autor: Luz María Álvarez Moreno

Fecha: 2/12/2025

Sesión 9: Transformación E-R al Modelo Relacional

+

Sesión 10: Restricciones de integridad: dominio, entidad y referencial

¿Qué es transformar un modelo E-R al modelo relacional?

Cuando diseñamos una base de datos, **el modelo E-R (Entidad-Relación)** es como un boceto conceptual: explica qué es importante, pero **no se puede implementar** en un SGBD.

La transformación al modelo relacional es el proceso que toma ese dibujo y lo convierte en **tablas reales**, con **columnas, claves primarias, foráneas y restricciones**.

Es un proceso **sistemático y sin ambigüedades**.

2. Reglas básicas de transformación

Regla 1: Cada entidad → una tabla

Ejemplo:

Entidad **Usuario** → Tabla **USUARIOS**

Regla 2 : Atributos → columnas

Si la entidad Usuario tiene:

- nombre_usuario
- email
- fecha_registro

Se convierten en columnas dentro de la tabla USUARIOS.

Regla 3: El identificador → Clave Primaria (PK)

Si Usuario tiene *id_usuario* como identificador → **PK(id_usuario)**

Regla 4: Las relaciones → Claves Foráneas (FK)

Si un usuario publica fotos (1:N), entonces la tabla FOTO lleva:

id_usuario_fk

Regla 5: Relaciones N:M → tabla intermedia

Ejemplo típico: Usuarios dan like a Fotos → tabla LIKES con dos claves foráneas.

Ejemplo práctico Instagram**ENTIDADES**

- Usuario
- Foto
- Comentario

Transformación

USUARIOS(id_usuario PK, nombre_usuario, email, fecha_registro, biografia)

FOTOS(id_foto PK, url_imagen, descripcion, fecha_publicacion, id_usuario_fk)

COMENTARIOS(id_comentario PK, texto, fecha_comentario, id_usuario_fk, id_foto_fk)

Tabla para la relación N:M

LIKES(id_usuario_fk, id_foto_fk, fecha_like)

PK compuesta: (id_usuario_fk, id_foto_fk)

¿Qué conseguimos?

- Obtener fotos por usuario: SELECT * FROM FOTOS WHERE id_usuario_fk = X
- Comentarios de una foto: SELECT * FROM COMENTARIOS WHERE id_foto_fk = Y

3. Restricciones de Integridad

Las restricciones de integridad son **las reglas que protegen la base de datos de errores**.

1. Restricciones de Dominio: qué valores son válidos

Ejemplos:

Columna	Restricción de dominio
edad	CHECK (edad BETWEEN 0 AND 120)
email	debe contener @
fecha	no puede estar en futuro
nombre	NOT NULL

Ejemplo práctico (USUARIOS):

email VARCHAR(255) NOT NULL UNIQUE,
edad INT CHECK (edad >= 0 AND edad <= 120)

2. Restricciones de Entidad: aseguran que cada fila es única

Incluye:

- Claves primarias (PK)
- Atributos únicos (email único)
- Comprobaciones internas (CHECK)

Ejemplo:

"Cada usuario debe tener un email único"
email VARCHAR(255) UNIQUE

3. Restricciones Referenciales: coherence entre tablas

Son las **claves foráneas (FK)**.

Ejemplo:

```
FOREIGN KEY (id_usuario_fk)
  REFERENCES USUARIOS(id_usuario)
  ON DELETE CASCADE
```

Ejemplo real PayPal

PayPal valida:

- emails con expresiones regulares,
- tarjetas con algoritmo de Luhn,
- fechas que no pueden ser futuras,
- PK globales únicas,
- que ninguna transacción apunte a usuarios inexistentes.

Este caso es perfecto para explicar la importancia de las restricciones.

4. Práctica guiada: De E-R al Modelo Relacional completo

Escenario: Plataforma tipo RRSS Lectores

Autores publican artículos, lectores comentan, usuarios siguen a autores, artículos tienen etiquetas.

FASE 1: Requisitos

Se desea diseñar la base de datos para una plataforma similar a un blog donde diversos usuarios interactúan entre sí.

En el sistema existen autores, que pueden publicar artículos con un *título, contenido, fecha de publicación y estado* (borrador o publicado).

Los lectores pueden comentar los artículos publicados por los autores. Además, los usuarios pueden seguir a otros usuarios, permitiendo así recibir actualizaciones de sus publicaciones.

Los artículos pueden clasificarse mediante etiquetas, existiendo una relación de tipo N:M entre artículos y etiquetas, ya que un artículo puede tener múltiples etiquetas y una etiqueta puede estar asociada a varios artículos.

- Autores: escriben artículos
- Artículos: títulos, contenido, fecha, estado
- Lectores: pueden comentar
- Usuarios: siguen a otros usuarios
- Artículos + Etiquetas : relación N:M

FASE 2: Modelo E-R

Entidades:

- Usuario
- Artículo
- Comentario
- Etiqueta
- Seguimientos

Relaciones:

- Usuario escribe Artículo (1:N)
- Usuario comenta Artículo (N:M)
- Artículos tienen Etiquetas (N:M)
- Usuario sigue Usuario (N:M)

FASE 3: Esquema Relacional

USUARIOS(id_usuario PK, nombre_usuario, email UNIQUE, fecha_registro, biografia)

ARTICULOS(id_articulo PK, titulo, contenido, fecha_publicacion, estado, id_autor_fk)

COMENTARIOS(id_comentario PK, texto, fecha_comentario, id_usuario_fk, id_articulo_fk)

ETIQUETAS(id_etiqueta PK, nombre_etiqueta, descripcion)

ARTICULO_ETIQUETA(id_articulo_fk, id_etiqueta_fk) PK compuesta

SEGUIMIENTOS(id_seguidor_fk, id_seguido_fk, fecha_seguimiento) PK compuesta

FASE 4: Restricciones

CHECK (estado IN ('borrador', 'publicado'))

FOREIGN KEY (id_autor_fk)

 REFERENCES USUARIOS(id_usuario)

 ON DELETE CASCADE

Solo se permiten valores borrador o publicado en la columna estado.

Si intentas hacer algo como:

INSERT INTO ARTICULOS (estado, ...) VALUES ('archivado', ...);
 la base de datos **rechaza** la operación porque 'archivado' no está en la lista permitida.

Sirve para:

- evitar errores de escritura ('borrador', 'publicado'...),
- garantizar que el estado siempre sea uno de los valores válidos.

FOREIGN KEY (id_autor_fk)
 REFERENCES USUARIOS(id_usuario)
 ON DELETE CASCADE

Esto define una **clave foránea** + qué pasa al borrar.

- FOREIGN KEY (id_autor_fk): indica que la columna id_autor_fk de la tabla actual apunta a...
- REFERENCES USUARIOS(id_usuario): ...la columna id_usuario de la tabla USUARIOS.

Eso obliga a que **todo id_autor_fk exista antes en USUARIOS**.

No puedes tener un artículo con autor 999 si no existe el usuario 999.
 La parte importante:

ON DELETE CASCADE

Significa:

Si se borra un usuario en la tabla USUARIOS, **se borran automáticamente** todas las filas de la tabla que lo referencien (por ejemplo, sus artículos).

Ejemplo:

- Borras al usuario con id_usuario = 5.
- La base de datos borra automáticamente todos los registros donde id_autor_fk = 5 en la tabla de ARTICULOS.

FASE 5: Validación

Preguntas que deben cumplirse:

- ¿Puedo obtener todos los artículos de un autor?
- ¿Puedo obtener todos los comentarios de un artículo?
- ¿Puedo saber qué autores sigo?
- ¿Puedo saber qué etiquetas tiene un artículo?

Si NO puedes responderlas, el esquema está mal.

Ejercicio para realizar en clase: Plataforma de Cursos y Recursos Educativos

Se desea diseñar la base de datos para una plataforma de aprendizaje online donde distintos usuarios pueden participar de varias formas.

En el sistema existen **profesores**, que pueden **crear cursos**.

Cada curso incluye un *título, descripción, fecha de creación y estado* (activo o inactivo).

Los **estudiantes** pueden **inscribirse en los cursos** y también pueden **dejar reseñas** sobre ellos, indicando un comentario y una valoración numérica.

Además, los usuarios de la plataforma pueden **seguir a profesores**, con el fin de recibir notificaciones sobre nuevos cursos que publiquen.

Los cursos pueden organizarse mediante **categorías**, existiendo una relación **N:M** entre cursos y categorías, ya que un curso puede pertenecer a varias categorías, y una categoría puede agrupar varios cursos.