

A continuación, se presenta un documento que detalla cómo ciertos elementos del Diseño Entidad-Relación (DER) se traducen al diseño relacional, con un enfoque específico en los casos que generan nuevas tablas o atributos en el modelo relacional.

Traducción de Modelos Entidad-Relación (DER) a Diseño Relacional: Casos que Generan Nuevas Tablas o Atributos

El Modelo Entidad-Relación (E/RM), propuesto por Peter Chen en 1976, es un modelo conceptual que representa la información de manera independiente del Sistema Gestor de Bases de Datos (DBMS). Este modelo conceptual organiza la realidad en "Entidades" y las "Relaciones" entre ellas, donde las entidades y las relaciones pueden poseer "atributos". El paso del diseño conceptual (DER) al diseño lógico y físico es una etapa fundamental en el ciclo de vida de una base de datos, donde se implantan las decisiones de diseño en un DBMS específico.

El modelo relacional, basado en la teoría de conjuntos, organiza los datos en "relaciones" o "tablas". A continuación, se describen los casos en la traducción del DER al modelo relacional que resultan en la creación de nuevas tablas o atributos.

1. Entidades y Atributos

-

Entidades: Cada "entidad" en el DER se traduce directamente en una nueva "tabla" en el diseño relacional.

-

Ejemplo: La entidad "ALUMNOS" se convierte en la tabla ALUMNOS.

-

Atributos Simples: Los atributos simples de una entidad pasan a ser atributos (columnas) de la tabla correspondiente.

-

Ejemplo: Atributos como nombre, apellidos, fecha_nacimiento de la entidad "ALUMNOS" serán columnas en la tabla ALUMNOS.

-

Atributos Compuestos: Estos atributos, que pueden descomponerse en otros más simples (ej., una dirección compuesta por calle, número, ciudad), se descompone y cada componente pasa a ser un atributo de la tabla principal.

-

Ejemplo: Si "DOMICILIO" es un atributo compuesto en la entidad "ALUMNOS", en la tabla ALUMNOS se crearán columnas como calle_domicilio, numero_domicilio, ciudad_domicilio.

-

Atributos Multivaluados: Son atributos que pueden tomar varios valores al mismo tiempo (ej., varios números de teléfono para una persona). Estos atributos no se incluyen directamente en la tabla de la entidad, sino que generan una nueva tabla. Esta nueva tabla tendrá el atributo multivaluado y una clave foránea (FK) que referencia a la clave primaria (PK) de la entidad original.

-

Ejemplo: Si la entidad "JUGADORES" tiene un atributo multivaluado "TELEFONOS", se creará una nueva tabla TELEFONOS con las columnas telefono y id_jugador (FK a la tabla JUGADORES).

-

Identificadores (Claves): El identificador principal (id) de una entidad se convierte en la PRIMARY KEY de la tabla. Otros identificadores candidatos se definen como UNIQUE en la tabla.

◦

Ejemplo: id_jugador de la entidad JUGADORES se convierte en la PK de la tabla JUGADORES. DNI o nro_licencia pueden ser UNIQUE.

2. Entidades Débiles o Dependientes

Una entidad débil depende de la existencia de otra entidad (fuerte o dominante).

•

Dependencia por Existencia: La entidad dependiente tiene un identificador principal propio. Se traduce en una nueva tabla para la entidad débil, y una clave foránea (FK) que referencia a la entidad fuerte.

◦

Ejemplo: Si E-MAILS es una entidad débil que depende de PROFESORES por existencia, E-MAILS tendrá su propia PK (id_email) y una FK (id_profesor) a PROFESORES.

•

Dependencia por Identificación: La entidad débil no tiene un identificador principal propio; su identificación depende de un atributo discriminante y de la clave de la entidad fuerte. Se traduce en una nueva tabla para la entidad débil. La clave primaria de esta nueva tabla será una clave compuesta formada por el atributo discriminante de la entidad débil y la clave foránea (FK) que referencia a la clave primaria de la entidad fuerte.

◦

Ejemplo: Si TELEFONOS es una entidad débil de PROFESORES por identificación, la tabla TELEFONOS tendrá como PK compuesta (telefono, id_profesor), donde id_profesor es FK a PROFESORES.

3. Relaciones Binarias (entre dos entidades)

•

Relaciones 1 a N (uno a muchos): En este tipo de relación, un ejemplar de una entidad se relaciona con muchos ejemplares de otra, pero un ejemplar de la segunda solo se relaciona con uno de la primera. La traducción implica añadir un atributo de clave foránea (FK) en la tabla del lado "N" (muchos) que referencia a la clave primaria de la tabla del lado "1" (uno). Si la cardinalidad mínima del lado "1" es "1", la FK debe ser NOT NULL; si es "0", puede aceptar NULL.

◦

Ejemplo: En la relación INTEGRAR entre JUGADORES (N) y EQUIPOS (1), se añade ID_EQUIPO como FK en la tabla JUGADORES.

•

Relaciones 1 a 1 (uno a uno): La traducción de relaciones 1:1 tiene varias opciones:

◦

Caso (1,1) – (1,1) (ambas entidades son obligatorias en la relación): Se puede crear una sola tabla combinando todos los atributos de las dos entidades, o tratarlo como un caso especial de 1-N, agregando una FK en cualquiera de las dos tablas. Esta FK debe ser UNIQUE NOT NULL.

◦

Caso (1,1) – (0,1) (una entidad es obligatoria, la otra opcional): Se trata como un caso especial de 1-N. Se agrega una FK en la tabla del lado (0,1), la cual debe ser UNIQUE NOT NULL.

◦

Caso (0,1) – (0,1) (ambas entidades son opcionales): Se trata como un caso especial de 1-N, agregando una FK en cualquiera de las dos tablas que debe ser solo UNIQUE.

-

Excepción: Muy pocas interrelaciones: Si la relación es esporádica (pocos ejemplares se relacionan), se debe crear una nueva tabla para registrar estas interrelaciones. Esta tabla contendrá las FKs a las dos tablas originales. Se elegirá una de ellas como PK (no se repetirá) y la otra como UNIQUE (tampoco se repetirá).

-

Ejemplo: Si la relación entre EMPLEADOS e INMUEBLES (para indicar quién tiene asignado qué inmueble) es muy escasa, se crearía una tabla ASIGNACION_INMUEBLES con FKs id_empleado y id_inmueble, donde una sería PK y la otra UNIQUE.

-

Relaciones N a N (muchos a muchos): Cuando varios ejemplares de una entidad se relacionan con varios ejemplares de otra. Este tipo de relación siempre se convierte en una nueva tabla en el diseño relacional.

-

Esta nueva tabla incluirá las claves foráneas (FK) de las dos tablas participantes. Conjuntamente, estas FKs formarán la clave primaria compuesta de la nueva tabla.

-

Si la relación en el DER tiene atributos (ej., puesto en la relación INTEGRAR entre JUGADORES y EQUIPOS), estos también se convierten en atributos de la nueva tabla.

-

Ejemplo: Una relación INTEGRAR N:N entre JUGADORES y EQUIPOS (donde un jugador puede haber jugado en varios equipos y un equipo tiene varios jugadores), se transforma en una nueva tabla JUGADORES_POR_EQUIPO con id_jugador y id_equipo como PK compuesta, y el atributo puesto como columna.

4. Relaciones Reflexivas

Son relaciones donde dos ejemplares de la misma entidad se relacionan.

-

Relaciones Reflexivas 1 a N: Se siguen las mismas reglas que una relación 1-N, pero la referencia es a la misma tabla. Se añade una columna de FK en la propia tabla que referencia a su PK.

-

Ejemplo: Un EMPLEADO SER JEFE de otros EMPLEADOS (1:N). La tabla EMPLEADOS tendría una columna id_jefe (FK) que referencia a id_empleado (PK) en la misma tabla.

-

Relaciones Reflexivas N a N: Se siguen las mismas reglas que una relación N-N. Se crea una nueva tabla que contendrá dos claves foráneas que referencian a la misma tabla original, y estas dos FKs juntas formarán la clave primaria compuesta de la nueva tabla.

-

Ejemplo: AEROPUERTOS que CONECTAR con otros AEROPUERTOS (N:N). Se crearía una tabla CONEXIONES_AEROPUERTO con id_aeropuerto_origen y id_aeropuerto_destino como PK compuesta, ambas referenciando a id_aeropuerto en la tabla AEROPUERTOS.

5. Relaciones Ternarias (y de Grado Superior)

Involucran tres o más entidades.

-

Caso N:M:P (muchos a muchos a muchos): Cuando cada ejemplar de una entidad se relaciona con muchos de las otras dos. La relación ternaria genera una nueva tabla en el diseño relacional.

- Esta nueva tabla contendrá una clave foránea (FK) a cada una de las tres entidades participantes. Las tres FKs juntas conformarán la clave primaria compuesta de la nueva tabla.

- Ejemplo: La relación Impartir entre PROFESORES, ASIGNATURAS y CURSOS (N:M:P). Se crearía una tabla IMPARTICION con id_profesor, id_asignatura y id_curso como PK compuesta.

- Caso N:M:1 (uno a muchos a muchos): Similar al anterior, pero una de las entidades tiene cardinalidad "uno" con respecto a la combinación de las otras dos. La relación genera una nueva tabla.

- Contiene una FK a cada entidad. Las dos FKs de los lados a "muchos" conformarán la PK compuesta de la nueva tabla. La FK que referencia al lado "1" debe ser NOT NULL.

- Ejemplo: Si una asignatura en un curso solo puede ser impartida por un profesor. La tabla IMPARTICION tendría id_asignatura e id_curso como PK compuesta, y id_profesor como FK NOT NULL.

- Caso N:1:1 y 1:1:1: Estos casos también generan una nueva tabla para la relación. Las claves primarias de estas nuevas tablas se definen a partir de combinaciones de las claves foráneas, asegurando unicidad y no nulidad para las claves candidatas elegidas.

----- 6. Relaciones de Herencia (IsA) / Generalización-Especialización

Se utilizan para agrupar atributos comunes de varias entidades.

- Estrategia: Tablas Relacionadas (la más común): Esta estrategia implica la creación de varias tablas. Se crea una tabla para la entidad generalizada (padre) con los atributos comunes. Adicionalmente, se crea una tabla para cada entidad especializada (hija) con sus atributos propios. Cada tabla de especialización contendrá una clave foránea (FK) que referencia a la clave primaria (PK) de la tabla de generalización, y esta FK será a su vez la clave primaria de la tabla de especialización.

- Ejemplo: La entidad PERSONAS (generalización) con atributos como nombre, apellidos, dni, f_nac. Las especializaciones JUGADORES, ENTRENADORES, MANAGERS tendrían sus propias tablas. JUGADORES tendría su id_jugador (PK y FK a PERSONAS) y atributos propios como puesto, activo.

- Estrategia: Una Sola Tabla: Se crea una única tabla para toda la jerarquía de herencia. Esta tabla contiene todos los atributos comunes, los atributos de cada especialización, y un atributo "discriminante" que indica a qué tipo de especialización pertenece cada tupla. Se utilizan restricciones CHECK para controlar qué atributos pueden ser NULL y cuáles no, basándose en el valor del discriminante.

- Ejemplo: Una única tabla FEDERADOS que contendría id_federado, nombre, apellidos, licencia, f_nac, puesto, activo, nivel, dedicacion, y un campo tipo_federado (jugador, entrenador, manager).

-

Estrategia: Tabla Independiente por Entidad: Esta estrategia implica la creación de una tabla separada por cada especialización, replicando los atributos heredados junto con los propios. Si la relación de herencia es parcial, también puede haber una tabla para la generalización. Sin embargo, las fuentes indican explícitamente NO UTILIZAR esta estrategia porque las tablas no están relacionadas, lo que resulta en la pérdida de controles de integridad referencial.

7. Agregaciones

Las agregaciones en el DER son una forma de tratar una relación como una entidad.

-

En el diseño relacional, una agregación no genera una nueva tabla adicional por sí misma. En su lugar, la agregación "representa" la tabla que ya se ha creado para la relación subyacente (por ejemplo, una relación N-N que ya se ha transformado en una tabla). Los atributos propios de la agregación se incorporan como columnas en esta tabla ya existente de la relación. Las relaciones posteriores con la agregación se establecen con esta tabla.

-

Ejemplo: Si la relación interpretar entre CANCIONES y GRUPOS se agrega, y luego ALBUMES se relaciona con esta agregación INTERPRETACIONES. La relación interpretar (N:N) ya habría generado una tabla INTERPRETACIONES con id_cancion y id_grupo como PK. La relación de ALBUMES con esta agregación se traduciría añadiendo una FK (id_interpretacion o la PK compuesta) en la tabla ALBUMES o una nueva tabla para la relación entre ALBUMES y INTERPRETACIONES.

En resumen, la traducción de un DER a un diseño relacional es un proceso sistemático donde las entidades se convierten en tablas, y las relaciones y tipos especiales de atributos/entidades (multivaluados, débiles, N-N, ternarias, reflexivas N-N y herencias) a menudo resultan en la creación de nuevas tablas o la adición de claves foráneas y restricciones en las tablas existentes para mantener la integridad de los datos y las relaciones definidas en el modelo conceptual.