Transformación del MERE al Modelo Relacional e Implantación en MySQL

Adrián Losada Álvarez
Simón Suárez Rosende

TABLA DE CONTENIDO

TRANSFORMACIÓN A MODELO RELACIONAL (MR)

Para la transformación del Modelo Entidad-Relación Extendido al Modelo Relacional se diseñaron inicialmente las tablas para las entidades y relaciones correspondientes dependiendo del tipo de restricción de cardinalidad, tipo de entidad, etc. A continuación, se muestran todas las tablas creadas y su justificación:

CLIENTE						
id	nombre	edad	genero	idSeguro	idViaje	fecha
PK						
				FK.SEGURO_VIA.	FK.VIAJE	

Esta tabla se creó a partir de la entidad "*CLIENTE*", en la que cada columna indica los atributos de esta entidad (id, nombre, edad y género), además se establece el atributo "*id*" como la **clave primaria** (***PK***). A esta tabla posteriormente se le agregaron las **claves foráneas** (***FK***) de las entidades "*SEGURO DE VIAJE*" y "*VIAJE*", "*idSeguro*" y "*idViaje*" respectivamente, ya que se tratan de **entidades fuertes** con relación **1:N**, además como la relación "*Compra*" entre el cliente y el viaje tiene un atributo descriptivo "*fecha*" este también se agrega a la tabla después de la **FK** del viaje.

RECOMIENDA	
idCliente	idRecomendado
PK	
FK.CLIENTE	FK.CLIENTE

La tabla "*RECOMIENDA*" surge de la relación recursiva: CLIENTE->RECOMIENDA->CLIENTE.
En este caso se trata de una relación reflexiva de tipo N:M, por lo tanto, se crea una tabla para la entidad ("*CLIENTE*", creada anteriormente) y otra tabla para la relación en la cual la **PK** será una **clave primaria compuesta** formada por las dos claves principales de las entidades participantes y serán además **FK** de las entidades de las que provienen.

SEGURO_VIAJE				
id	caducidad	precio	cancelación	tipo
PK				

La tabla para "*SEGUROS DE VIAJE*" está compuesta por sus atributos (id, caducidad, precio, cancelación), además de un atributo añadido "*tipo*", que surge de la **relación de especialización** de tipo **disjunta total** (**disjunta**: ya que un "*SEGURO DE VIAJE*" **no puede ser** "*BÁSICO*" y "*A TODO RIESGO*" al mismo tiempo, **total**: un "*SEGURO DE VIAJE*" **obligatoriamente será** de tipo "*BÁSICO*" o "*A TODO RIESGO*" **exclusivamente**), además los subtipos no se diferencian en ningún atributo y no tienen clave primaria propia. Por lo tanto, se opta por el método 3 de los apuntes de teoría para crear la tabla.

VIAJE			
id	precio	duración	transporte
PK			

Se trata de una tabla simple creada a partir de los atributos de la entidad formada por agregación "*VIAJE*", en donde la **PK** es el "*id*" de este.

NACIONAL	
idViaje	dni
PK	
FK.VIAJE	

INTERNACIONAL		
idViaje	pasaporte	visado
PK		
FK.VIAJE		

Las tablas anteriores surgen de la **relación de especialización** de tipo **solapada parcial** (**solapada**: ya que un "VIAJE" **puede ser** "*NACIONAL*" e "*INTERNACIONAL*" al mismo tiempo, **parcial**: ya que un "*VIAJE*" **puede ser** de tipo "*NACIONAL*" o "*INTERNACIONAL*", **además de otros**), además como todos los atributos de estos subtipos son distintos y se quiere mantener los atributos comunes a ambos, se optó por el método 1 de conversión de MERE a MR de los apuntes, en el cual se crea una tabla para cada especialización en la cual la **PK** será la **FK** de la entidad padre.

PAQUETE_VACACIONAL	
id	n_destinos
PK	

DESTINO		
id	país	alojamiento
PK		

Al igual que en el caso de la tabla "*VIAJE*", estas dos tablas son simples ya que solamente están formadas por sus atributos correspondientes y su **PK** es su "*id*" correspondiente.

INCLUYE	
idPaquete	idDestino
PK	
FK.PAQUETE_VACACIONAL	FK.DESTINO

La tabla "*INCLUYE*" surge de la relación **N:M** entre las entidades "*PAQUETE VACACIONAL*" y "*DESTINO*", por lo cual, esta tabla contendrá como **PK** una **clave principal compuesta** formada por las dos claves principales de las entidades participantes y serán además **FK** de las entidades de las que provienen.

ACTIVIDAD			
idActividad	duración	precio	idViaje
PK			
			FK.VIAJE

Esta tabla pertenece al **caso 1 de entidades débiles con clave primaria para relaciones 1:N**, en las cuales se crea una tabla para la entidad débil de manera normal pero se le agrega una columna (a la tabla del lado **N**)(o más si la relación tuviera atributos descriptivos) en la cual se introduce como atributo el "*id*" de la entidad padre, que será convertirá en **FK**.

GUÍA			
idActividad	género	edad	nombre
PK			

Esta tabla, a diferencia del caso anterior, pertenece al **caso 2 de entidades débiles sin clave primaria para relaciones 1:1**, en las cuales se genera una tabla para la entidad débil y se coloca la clave principal de la entidad fuerte en esta como **PK** y **FK**.

IMPLEMENTACIÓN EN MYSQL

Para la implementación en MySQL de las tablas diseñadas en el apartado anterior, nos ayudamos de los comandos DDL:
- *CREATE TABLE IF NOT EXISTS (nombre_tabla) (atributos)*: Con este comando se crea la tabla si no existía previamente.
- *INSERT INTO (nombre tabla) (valores)*: Con este comando se introducen datos en las tablas creadas previamente

Además, se utilizan las siguientes definiciones de tipo de valores:
- *INT*: Tipo de datos para números enteros.
- *VARCHAR(int)*: Tipo de datos para cadenas de caracteres de longitud variable (*int*).
- *DATE*: Tipo de datos para almacenar fechas sin incluir información de tiempo.
- *DECIMAL(int1, int2)*: Tipo de datos para valores numéricos decimales con una precisión total de *int1* dígitos, incluidos *int2* dígitos a la derecha del punto decimal.
- *BOOLEAN*: Tipo de datos que representa valores de verdadero o falso.
- *ENUM*: Tipo de datos que permite especificar un conjunto de valores posibles para una columna.
- *TIMESTAMP*: Tipo de datos para almacenar fechas y horas con una resolución de segundos.

Por último, se utilizan las siguientes instrucciones para los atributos de las tablas:
- *AUTO_INCREMENT*: Se utiliza al definir una columna numérica, generalmente de tipo entero, en una tabla para asignar automáticamente un valor incremental **único** a esa columna para cada fila nueva que se inserte.
- *PRIMARY KEY*: Se utiliza para definir qué valor tomará el rol de clave primaria.
- *NOT NULL*: Se utiliza para indicar que esa columna no puede contener valores nulos, es decir, cada fila debe tener un valor válido en esa columna.
- *FOREIGN KEY (columna) REFERENCES (nombre_tabla)(nombre_tabla->columna)*: La primera parte del comando (*FOREIGN KEY ()*) indica que estamos estableciendo una restricción de clave foránea en la columna *columna* de la tabla actual. Mientras que la segunda parte (*REFERENCES ()()*) especifica la tabla y la columna a la que se hace referencia.
- *ON DELETE CASCADE*: Esta cláusula especifica qué sucede cuando se elimina (*ON DELETE*) una fila en la tabla actual. En este caso, indica que, si se elimina una fila de la tabla actual, todas las filas relacionadas en otra tabla que tienen el mismo valor en la columna de la **FK** también se eliminarán automáticamente (*CASCADE*).