

Unidad 4 (Semana 4) – Bases de datos I - Práctica

TP Semana 2 (de la unidad 4) – Diseño de bases de datos (denormalización) |

Estudiante: Varela Santiago Octavio - Email Institucional: santiago.varela@tupad.utn.edu.ar

(Comisión 14 - DNI 38011156)

*Este archivo PDF también se encontrará en el repositorio de la materia:

https://github.com/santiagovOK/UTN-TUPaD-bd_1

Parte I - Evaluación del diseño normalizado

Análisis de claves primarias y foráneas:

Utilizaré como ejemplo la estructura sugerida para analizar. Teniendo esa base, así quedarían las claves primarias y foráneas:

Tablas	PK	FK
CLIENTES	id_cliente	Id_localidad, de LOCALIDADES(id_localidad)
VENTAS	id_venta	Id_cliente, de CLIENTES(id_cliente)
DETALLES_VENTA	id_venta, id_producto (clave compuesta)	Id_venta, de VENTAS(id_venta) Id_producto, de PRODUCTOS(id_producto)
PRODUCTOS	id_producto	-
LOCALIDADES	id_localidad	-

Identificación de posibles violaciones a la 2FN y 3FN:

Comenzando por la 2FN, el criterio a considerar frente a la identificación de posibles violaciones es el de la no existencia de dependencias parciales, principalmente de atributos no-clave respecto a una clave compuesta. En el caso de las tablas CLIENTES, LOCALIDADES, PRODUCTOS y VENTAS la clave primaria es simple y no hay dependencias parciales. En el caso de la única tabla con clave primaria compuesta, que es DETALLES_VENTA, tanto `cantidad` como `precio_unitario` dependen de ambas claves, no solo de una sola. Si solo se relacionaran con una sola, no tendrían un sentido en sí mismas como atributo.

Respecto a la 3FN, el criterio que servirá como parámetro será el hecho de que no debe haber dependencias transitivas de atributos no-clave respecto a la clave primaria. Siguiendo este criterio, no se encuentran dependencias transitivas en LOCALIDADES, PRODUCTOS, DETALLES_VENTA y CLIENTES, pero sí en VENTAS, en donde `total` podría derivarse de la operación `cantidad * precio_unitario` dentro de DETALLES_VENTA, lo que genera una dependencia transitiva. Entonces, una sugerencia sería solucionar esa dependencia para que se cumpla con la 3FN.

Propuesta de la estructura corregida

Tabla	Atributos	PK	FK	Notas
CLIENTES	Id_cliente, nombre, email, id_localidad	id_cliente	Id_localidad, de LOCALIDADES(id_localidad)	-
VENTAS	Id_venta, fecha, id_cliente	id_venta	Id_cliente, de CLIENTES(id_cliente)	Eliminamos la columna `total` de la tabla
DETALLES_VENTA	Id_venta, id_producto, cantidad, precio_unitario	(id_venta, id_producto)	Id_venta, de VENTAS (id_venta); Id_producto, de PRODUCTOS (id_producto)	-
PRODUCTOS	Id_producto, descripcion, precio	id_producto	-	-
LOCALIDADES	Id_localidad, nombre_provincia	id_localidad	-	-

Finalmente, la estructura corregida cumple con las tres primeras formas normales (1FN, 2FN y 3FN) sin cambiar radicalmente su estructura.

Parte II - Normalización inversa guiada

Tablas a fusionar o simplificar

Considerando que cualquier modificación a las tablas normalizadas previas puede afectar el rendimiento y la redundancia de los datos, elegí focalizar en las tablas VENTAS y DETALLES_VENTA con la finalidad de fusionarlas. Esta fusión reduciría la cantidad de joins en

reportes frecuentes, que es uno de los objetivos establecidos en la consigna, pero también mejoraría el rendimiento, no solo por la reducción de joins sino por la simplificación de la estructura en general. VENTAS y DETALLES_VENTA se fusionará en VENTAS_DETALLES, que contendrá los atributos id_venta, fecha, id_cliente, id_producto, cantidad y precio_unitario. Entre estos atributos, la PK será la combinación de id_venta y id_producto (como lo era anteriormente en DETALLES_VENTA), mientras que las FK serán id_cliente (de CLIENTES) y id_producto (de PRODUCTOS). Resumen de la tabla VENTAS_DETALLES:

Tabla	Atributos	PK	FK
VENTAS_DETALLES	Id_venta, fecha, id_cliente, id_producto, cantidad, precio_unitario	Id_venta, id_producto (clave compuesta)	id_cliente, de CLIENTES; Id_producto, de PRODUCTOS

Justificación de la decisión desde el punto de vista de rendimiento

La reducción de los joins es directamente proporcional a la mejora de rendimiento, de allí ya se explica una mejora. VENTAS_DETALLES, la nueva tabla propuesta, reduciría la cantidad de joins necesarios en las consultas que requieren información de lo que eran las dos tablas anteriores, VENTAS y DETALLE_VENTA. Esto, como ya mencionamos, también simplifica la estructura respecto a la realización de consultas.

Sin embargo, esto no quiere decir que no haya desventajas. Es posible que haya casos de redundancia de datos debido a estas modificaciones, ya que se almacenarán la información de las dos anteriores tablas en una sola, pero se puede balancear gracias a la mejora de rendimiento ya retomada en el párrafo anterior. A su vez, es posible que haya complicaciones a la hora de realizar actualizaciones en los registros, ya que haber unido esas dos tablas en VENTAS_DETALLES complejiza el mantenimiento de esa tabla puntual. Esto último podría requerir más esfuerzo y tiempo para mantener la integridad de los datos.