

Proceso de Normalización

Formas Normales



1

1NF, 2NF, 3NF

Medidas informales para el diseño de esquemas de relaciones

- Hasta ahora se ha supuesto que los atributos agrupados para formar un esquema de relación **surgen del sentido común del diseñador de la base de datos**, y luego de la transformación de un esquema especificado en el modelo Entidad-Relación a un esquema relacional.
- Sin embargo, **no se había contado con una medida formal de *por qué una agrupación de atributos para formar un esquema de relación puede ser mejor que otra***, es decir, contar con la mera intuición del diseñador significa no fundarnos en ninguna medida de lo apropiado del diseño o de su calidad

Medidas informales para el diseño de esquemas de relaciones

Para intentar “elegir” buenos esquemas de relación se ha desarrollado una teoría que pueda **medir formalmente las razones por las que una agrupación de atributos es mejor que otra.**

Existen dos niveles en los que podemos medir la “bondad” de los esquemas de relación: lógico y de manipulación (o de almacenamiento).

- **Nivel Lógico:** Se refiere a la manera en que los usuarios interpretan los esquemas de relación y el significado de sus atributos.
- **Nivel Físico:** Se refiere a cómo se almacenan y actualizan las tuplas de una relación base.

Medidas informales para el diseño de esquemas de relaciones

- Semántica de los atributos
- Reducción de los valores redundantes en las tuplas
- Reducción de los valores nulos en las tuplas
- Prohibición de tuplas espurias

Semántica de los atributos

- El significado o semántica especifica cómo se han de interpretar los atributos almacenados en una tupla.
- Cuanto más fácil sea de explicar la semántica de la relación, mejor será el diseño del esquema correspondiente.

Redundancia de valores en tuplas

Los valores redundantes provocan ciertas anomalías, de acuerdo a la tarea que se desea realizar.

- Anomalías de Inserción: al insertar una nueva tupla, los valores de algunos atributos deben ser congruentes con los valores de un atributo igual en otras tuplas. Para el/los atributo/s redundantes será difícil insertar una tupla con valores nuevos.
- Anomalías de Eliminación: si se elimina una tupla con la última información de un atributo correspondiente a otra relación se perderá información de la base de datos.
- Anomalías de Modificación: si alteramos el valor de uno de los atributos redundantes deberemos actualizar todas las relaciones que los contengan.

Reducción de Valores nulos en tuplas

Por otro lado, la reducción de valores nulos en las tuplas es muy importante para evitar que ocurran los siguientes problemas:

- Desperdicio de espacio de almacenamiento
- Dificultad en la comprensión de la semántica de los atributos (varias interpretaciones)
- Dificultad en las especificaciones de las operaciones de Reunión en el nivel lógico.
- No saber cómo manejarlos cuando se aplican funciones agregadas.

Prohibición de tuplas espúreas

Las tuplas espúreas son tuplas adicionales resultantes de una operación de Join natural, que originalmente no existen en los datos registrados en la base de datos

- ▶ Las relaciones deben diseñarse, de tal forma que puedan “relacionarse” o “reunirse” mediante condiciones de igualdad sobre atributos que sean clave primaria y foránea, y no genere este tipo de tuplas.
- ▶ No se debe incluir relaciones que contengan atributos coincidentes que no estén definidos como claves primarias o foráneas.

Pautas de diseño

Pauta 1: Diseñe un esquema de relación de modo que sea fácil explicar su significado. No combine atributos de varios tipos de entidades y tipos de vínculos en una sola relación. Es decir, que el esquema de relación sea semánticamente claro y no confuso.

Pauta 2: Diseñe los esquemas de las relaciones de modo que no haya anomalías de actualización. Si las hay señálelas con claridad para que los programas que actualicen la base de datos operen correctamente.

Pautas de diseño

Pauta 3: Hasta donde sea posible, evite incluir en una relación base atributos cuyos valores puedan ser nulos. Si no es posible, asegúrese que se apliquen en casos excepcionales y no a la mayoría de las tuplas.

Pauta 4: Diseñe los esquemas de relación de modo que puedan reunirse mediante condiciones de igualdad sobre atributos que sean claves primarias o externas, a fin de garantizar de que no se formarán tuplas espurias

Diseño

Una tabla o esquema de relación está bien diseñado cuando cumple con las 6 (seis) formas normales, 1NF, 2NF, 3NF, BCNF, 5NF, 6NF, cuya base o fundamento es el concepto de Dependencia Funcional.

El proceso de normalización

- ▶ La normalización de los datos es un proceso durante el cual los esquemas de relación insatisfactorios se descomponen repartiendo sus atributos en esquemas de relación más pequeños que tienen las propiedades deseables.
- ▶ En el proceso de normalización, se somete un esquema de relación para certificar si pertenece o no a cierta forma normal

El proceso de normalización

Es un **estándar** que consiste, básicamente, en un proceso de conversión de las relaciones entre las entidades, evitando:

La redundancia de los datos: repetición de datos en un sistema.

Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.

Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos

Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos

Las formas normales

Proveen a los diseñadores de un **marco formal** para analizar los esquemas de relación **en base a sus claves y a las dependencias funcionales entre sus atributos**, y de una serie de pruebas que pueden efectuarse sobre esquemas de relación individuales de modo que la base de datos relacional pueda normalizarse **hasta el grado deseado**.

Primera forma normal

Se dice que una tabla se encuentra en primera forma normal (**1NF**) si y solo si cada uno de los campos contiene un único valor para un registro determinado.

Es decir, la misma establece que “los dominios de los atributos deben incluir sólo valores atómicos (simples, indivisibles) y que el valor de cada atributo en una tupla debe ser un valor individual proveniente del dominio de ése atributo”.

Primera Forma Normal (1FN)

- Todos los atributos contienen valores atómicos (indivisibles).
- No hay grupos repetidos o multivaluados.

ID_Cliente	Nombre	Teléfonos
1	Juan	12345, 67890
2	María	54321

Aquí, la columna "Teléfonos" contiene varios números en una sola celda para el cliente Juan, lo que viola la 1FN.

ID_Cliente	Nombre	Teléfono
1	Juan	12345
1	Juan	67890
2	María	54321

Ahora cada valor en "Teléfono" es atómico y no hay grupos multivaluados.

Ejemplo

NroDpto	NombreDpto	CodLugar	NombreLugar
110	Administración	20, 30, 40	Tucumán, Salta, Jujuy
215	Gerencia	50	Córdoba
354	Compras	90, 20	Santa Fe, Tucumán

Ejemplo (solución 1)

NroDpto	NombreDpto	CodLugar	NombreLugar
110	Administración	20, 30, 40	Tucumán, Salta, Jujuy
215	Gerencia	50	Córdoba
354	Compras	90, 20	Santa Fe, Tucumán

NroDpto	NombreDpto	CodLugar	NombreLugar
110	Administración	20	Tucumán
110	Administración	30	Salta
110	Administración	40	Jujuy
215	Gerencia	50	Córdoba
354	Compras	90	Santa Fe
354	Compras	20	Tucumán

Ejemplo (solución 2)

NroDpto	NombreDpto	CodLugar	NombreLugar
110	Administración	20, 30, 40	Tucumán, Salta, Jujuy
215	Gerencia	50	Córdoba
354	Compras	90, 20	Santa Fe, Tucumán

NroDpto	NombreDpto
110	Administración
215	Gerencia
354	Compras

NroDpto	CodLugar	NombreLugar
110	20	Tucumán
110	30	Salta
110	40	Jujuy
215	50	Córdoba
354	90	Santa Fe
354	20	Tucumán

Dependencia Funcional

Las dependencias funcionales son propiedades inherentes al contenido semántico de los datos, que se han de cumplir para cualquier extensión del esquema de relación y forman parte de las restricciones del usuario.

La principal utilidad de las dependencias funcionales es describir mejor un esquema de relación mediante la especificación de restricciones sobre sus atributos que deben cumplirse siempre.

Dependencia Funcional

Definición formal:

Sea R una relación y sean x e y atributos de R , $x \rightarrow y$ se cumple en R , entonces para cualquier par de tuplas t_1 y t_2 de R se cumple:

$$t_1[x] = t_2[x] \rightarrow t_1[y] = t_2[y]$$

Así pues, x determina funcionalmente a y en un esquema de relación R si y sólo si, siempre que dos tuplas $r(R)$ coincidan en su valor x , necesariamente deben coincidir en su valor y .

Si x es una clave candidata de R (no se puede repetir el valor de x en otra tupla) entonces $x \rightarrow y$ para cualquier subconjunto de atributos y de R . Que $x \rightarrow y$ en R no dice absolutamente nada de $y \rightarrow x$ en R .

Segunda forma normal

La segunda forma normal compara *todos y cada uno* de los campos de la tabla con la clave definida. Si todos los campos dependen directamente de la clave se dice que la tabla está en segunda forma normal (2NF).

- Se basa en el concepto de Dependencia Funcional Total.
- Una dependencia funcional $x \rightarrow y$ es total si la eliminación de cualquier atributo "a" de X hace que la dependencia deje de ser válida.
- Un esquema de relación R está en 2NF si todo atributo *no primo* a en R depende funcionalmente de manera total de la clave primaria de R.

Primera Forma Normal (1FN)

- Todos los atributos contienen valores atómicos (indivisibles).
- No hay grupos repetidos o multivaluados.

ID_Cliente	Nombre	Teléfonos
1	Juan	12345, 67890
2	María	54321

Aquí, la columna "Teléfonos" contiene varios números en una sola celda para el cliente Juan, lo que viola la 1FN.

ID_Cliente	Nombre	Teléfono
1	Juan	12345
1	Juan	67890
2	María	54321

Ahora cada valor en "Teléfono" es atómico y no hay grupos multivaluados.

Segunda forma normal

Clave compuesta

NroEmple	NroDpto	Empleado	Dpto	Ingreso
10	110	María	Administración	20-01-2010
20	354	Pedro	Compras	30-03-2014
30	110	Elena	Administración	04-08-2013
40	350	Pablo	Ventas	07-06-2016
20	215	Pedro	Gerencia	01-06-2020
40	110	Pablo	Administración	01-11-2021

El campo empleado **no depende funcionalmente de toda la clave**, sólo depende del número del empleado. El campo Dpto **no depende funcionalmente de toda la clave**, sólo del número del departamento.

El campo Ingreso sí que **depende funcionalmente** de la clave ya que depende del número del empleado y del departamento (representa la fecha de ingreso de cada empleado en cada departamento)

Segunda forma normal

Tabla A

NroEmple	Empleado
10	María
20	Pedro
30	Elena
40	Pablo

Tabla B

NroDpto	NombreDpto
110	Administración
354	Compras
350	Ventas
215	Gerencia

Tabla C

NroEmple	NroDpto	Ingreso
10	110	20-01-2010
20	354	30-03-2014
30	110	04-08-2013
40	350	07-06-2016
20	215	01-06-2020
40	110	01-11-2021

Podemos observar que ahora si se encuentran las tres tablas en segunda forma normal, considerando que la tabla A tiene como índice el campo NroEmple, la tabla B NroDpto y la tabla C una clave compuesta por los campos NroEmple y NroDpto.

Tercera forma normal

Se dice que una tabla está en tercera forma normal si y solo si los campos de la tabla dependen únicamente de la clave, dicho en otras palabras los campos de las tablas no dependen unos de otros.

- Se basa en el concepto de Dependencia Transitiva.
- Una dependencia funcional $x \rightarrow y$ es una dependencia transitiva si existe un conjunto de atributos Z **que no sea** un subconjunto de cualquier clave de R , y se cumplen tanto $x \rightarrow z$ como $z \rightarrow y$.
- De acuerdo a la definición original de Codd, un esquema de relación R está en 3NF si está en 2NF y ningún atributo no primo de R depende transitivamente de la clave primaria.

Tercera forma normal

Tomando como referencia el ejemplo anterior, supongamos que un empleado puede estar asignado solo a un departamento a la vez y que queremos guardar el numero de Empleado del gerente del departamento. Se puede plantear la siguiente estructura.

NroEmple	Empleado	FechaN	Direccion	NroDpto	Dpto	NroEmpleGte
-----------------	-----------------	---------------	------------------	----------------	-------------	--------------------

Estudiamos la dependencia con respecto a la clave codigo

- Dpto depende directamente de NroDpto
- Empleado depende del NroEmple
- El numero de empleado del gerente, aunque en parte, también interesa el empleado, depende del departamento en el que trabaja

Tercera Forma Normal (3FN)

- Está en 2FN.
- No existen dependencias transitivas (un atributo no clave no depende de otro atributo no clave).

ID_Empleado	Nombre	Departamento	Jefe_Departamento
1	Juan	IT	Carlos
2	María	Marketing	Laura

Aquí, "Jefe_Departamento" depende de "Departamento", lo que introduce una dependencia transitiva, violando la 3FN

Tercera Forma Normal (3FN)

Empleado

ID_Empleado	Nombre	Departamento
1	Juan	IT
2	María	Marketing

Departamento

Departamento	Jefe_Departamento
IT	Carlos
Marketing	Laura

Ahora no hay dependencias transitivas, cada atributo depende solo de la clave primaria.

Solución

NroEmple	Empleado	FechaN	Direccion	NroDpto
----------	----------	--------	-----------	---------

NroDpto	Dpto	NroEmpleGte
---------	------	-------------

Forma normal de Boyce-Codd

Una tabla está en cuarta forma normal si y sólo si para cualquier combinación clave - campo no existen valores duplicados. Es más estricta que la 3NF, lo que significa que toda relación que está en FNBC también está en 3NF; sin embargo, una relación en 3NF no está necesariamente en FNBC.

Un esquema de relación R está en BCNF si siempre que una dependencia funcional $x \rightarrow A$ es válida en R, entonces x es una superclave de R.



Muchas gracias