

Tema 2-Diseño lógico

Es el proceso mediante el cual sabemos si el diseño de una base de datos es lo suficientemente bueno. Su finalidad es eliminar la redundancia y las anomalías de inserción, eliminación y borrado. La idea es dividir en esquemas de relación más pequeños aquellos esquemas que no cumplan determinadas condiciones.

La forma normal en la que se encuentra una relación es la forma normal más alta que cumple. Además se tiene que evitar la pérdida.

Un atributo de R es primo si pertenece a alguna clave candidata de la relación.

1ª forma normal

Es la propia definición de relación. Simplemente dice que el dominio de un atributo debe incluir valores indivisibles y que en cada tupla debe aparecer un valor individual. Por ejemplo:

Deptno	Loc
10	New York, A Coruña
20	Dallas, Las Cruces

No está en 1ª forma normal porque Loc no es atómico. Se arreglaría así:

Deptno	Loc
10	New York
20	Dallas
10	A Coruña
20	Las Cruces

2ª forma normal

Un esquema está en 2ª forma normal si todo atributo no primo depende funcionalmente de todas las claves candidatas con dependencias funcionales completas. Por ejemplo en esta relación:

Deptno	Dname	Loc
10	Accounting	New York

Deptno	Dname	Loc
20	Research	Dallas
10	Accounting	A Coruña
20	Research	Las Cruces

Si matizamos en que dos departamentos pueden tener el mismo nombre, la única dependencia que se puede extraer es $\{Deptno\} \rightarrow \{Dname\}$. La clave candidata es (Deptno, Loc). No está en 2ª forma normal porque Dname no depende de (Deptno, Loc), solo depende de Deptno. Si la separamos en 2 relaciones:

Deptno	Dname
10	Accounting
20	Research

Deptno	Loc
10	New York
20	Dallas
10	A Coruña
20	Las Cruces

En la primera relación Dname tiene una dependencial funcional completa con Deptno (clave candidata). En la otra relación no hay dependencias funcionales ni atributos no primos. Por lo tanto ambas están en 2ª forma normal.

3ª forma normal

Una dependencia funcional $\{X\} \rightarrow \{Y\}$ es transitiva si existe un conjunto de atributos Z que no sea subconjunto de una clave candidata que cumpla que $\{X\} \rightarrow \{Z\}$ y $\{Z\} \rightarrow \{Y\}$. Un esquema está en 3ª forma normal si está en 2ª y ningún atributo no primo depende transitivamente de una clave candidata.

Otra manera de saberlo es si:

- Todas las dependencias funcionales son triviales.
- La parte izquierda es superclave de R en todas.
- Cada atributo de la derecha que no esté en la izquierda está contenido en alguna clave candidata.

En el ejemplo de los empleados, Sueldo tenía una dependencia transitiva con CodEmp a través de NombreCat. Por tanto, no está en 3ª forma normal.

Forma normal de Boyce-Codd

Nació porque en ciertos casos la 3ª forma normal tenía redundancias. Un esquema está en Boyce-Codd si para toda dependencia $\{X\} \rightarrow \{Y\}$, X es superclave de R. Por ejemplo:

Un empleado trabaja en una sucursal. Un cliente puede tener cuentas en varias sucursales, pero en cada sucursal tiene asignado un “empleado gestor”.

$F = \{$
 $\{ID_cliente, ID_sucursal\} \rightarrow \{ID_empleado\}$
 $\{ID_empleado\} \rightarrow \{ID_sucursal\}$
 $\}$

Todos los atributos son primos, ya que las claves candidatas son (ID_cliente, ID_sucursal) y (ID_cliente, ID_empleado). Está en 3ª forma normal, pero no en Boyce-Codd porque ID_empleado no es superclave. Debemos descomponerla en las relaciones empleado+sucursal y cliente+empleado (ambas estarán en Boyce-Codd).

Las dependencias muchas veces no se conservan al pasar a Boyce-Codd. Por ejemplo la dependencia $\{ID_cliente, ID_sucursal\} \rightarrow \{ID_empleado\}$ se pierde.