

5. Otras formas normales: FNBC, 4FN y otras

5.1. FNBC (forma normal de Boyce-Codd)

La forma normal de Boyce-Codd es una extensión de la tercera forma normal y también se conoce como forma normal 3,5.

Una tabla se dice que está en forma normal Boyce-Codd (FNBC) si:

- ✓ Está en 3FN.
- ✓ Para cualquier dependencia $A \rightarrow B$, A debería ser una superclave.

Por tanto, una base de datos estará en FNBC si todas sus tablas lo están.

El segundo punto suena un poco complicado, ¿verdad? En palabras simples, significa que para una dependencia $A \rightarrow B$, A no puede ser un atributo no principal si B es un atributo principal.

Vamos a intentar entender este concepto en el siguiente ejemplo:

| MATRÍCULAS | | |
|------------|--------|--------------|
| NIA | Módulo | Profesor |
| 0011 | PRG | José |
| 0011 | BDD | Fidel |
| 0022 | PRG | Raúl Vicente |
| 0033 | LMSGI | Alfredo |
| 0044 | PRG | José |

Fig. 5.28. Tabla MATRÍCULAS.

En la tabla de arriba:

- Un estudiante puede inscribirse en varios módulos. Por ejemplo, el estudiante con NIA 0011 ha optado por asignaturas PRG y BDD.
- Para cada módulo se asigna un profesor al alumno.
- Puede haber varios profesores enseñando un módulo, como tenemos para PRG.

¿Cuál crees que debería ser la clave principal?

En la tabla anterior, NIA y el módulo en conjunto forman la clave principal, porque al usar NIA y módulo podemos encontrar todas las columnas de la tabla. Otro punto importante es que un profesor enseña solo un módulo, pero un módulo puede tener dos profesores diferentes. Por lo tanto, existe una dependencia entre el módulo y el profesor, donde el módulo depende del nombre del profesor.

Esta tabla satisface la 1FN porque todos los valores son atómicos, los nombres de las columnas son únicos y todos los valores almacenados en una columna en particular son del mismo dominio. Esta tabla también satisface la 2FN, ya que cada atributo depende de la clave completa. Y todos los atributos no claves son independientes. Por lo tanto, la tabla también satisface la 3FN. Pero esta tabla no está en la FNBC.

¿Por qué esta tabla no está en FNBC?

En la tabla anterior, NIA y módulo forman la clave principal, lo que significa que la columna del módulo es un atributo principal. Pero hay una dependencia más: profesor \rightarrow módulo. Y mientras que el módulo es un atributo principal, el profesor es un atributo no principal, que no está permitido por FNBC.

¿Cómo satisfacer FNBC?

Para que esta relación (tabla) satisfaga la FNBC, descompondremos esta tabla en dos tablas, tabla de estudiantes y tabla de profesores. A continuación tenemos la estructura de ambas tablas.

| ALUMNADO | | PROFESORADO | | |
|----------|---------|-------------|--------------|--------|
| NIA | ID_Prof | ID_Prof | Profesor | Módulo |
| 0011 | 01 | 01 | José | PRG |
| 0011 | 02 | 02 | Fidel | BDD |
| 0022 | 03 | 03 | Raúl Vicente | PRG |
| 0033 | 04 | 04 | Alfredo | LMSGI |
| 0044 | 01 | | | |

Fig. 5.29. Tablas ALUMNADO y PROFESORADO.

5.2. 4FN

La 4FN entra en escena cuando la dependencia de valor múltiple o dependencia multivalor ocurre en cualquier relación. Para que exista una dependencia de valor múltiple, la tabla debe tener al menos 3 columnas.

Una tabla se dice que está en cuarta forma normal (4FN) si:

- ✓ Está en FNBC.
- ✓ No existe ninguna dependencia de valor múltiple.

Por tanto, una base de datos estará en 4FN si todas sus tablas lo están.

Intentemos entender qué es la dependencia de valores múltiples mediante el siguiente ejemplo.

Imaginemos una tabla donde vamos a almacenar datos acerca de la matrícula de alumnos de ciclos formativos en distintos módulos y en los que se pide también que especifiquen los lenguajes de programación que les gustaría aprender durante sus estudios en el ciclo formativo.

| MATRÍCULAS | | |
|------------|--------|----------|
| NIA | Módulo | Lenguaje |
| 0011 | PRG | Java |
| 0011 | BDD | C# |
| 0022 | EDD | Java |
| 0022 | LMSGI | C# |

Fig. 5.30. Tabla MATRÍCULAS.

Como puede verse en la tabla anterior, el estudiante con NIA 0011 ha optado por matricularse

en dos módulos, PRG y BDD, y le gustaría aprender dos lenguajes de programación, Java y C#.

Debes estar pensando qué problema puede llevar esto, ¿verdad?

Bueno, los dos registros para el estudiante con NIA 0011 darán lugar a dos registros más, como se muestra a continuación, porque para un estudiante existen dos lenguajes de programación; por lo tanto, estos lenguajes pueden darse a la vez junto con ambos módulos, lo que daría lugar a los siguientes registros.

| MATRÍCULAS | | |
|------------|--------|----------|
| NIA | Módulo | Lenguaje |
| 0011 | PRG | Java |
| 0011 | PRG | C# |
| 0011 | BDD | Java |
| 0011 | BDD | C# |

Fig. 5.31. Tabla MATRÍCULAS.

En la tabla anterior no hay relación entre las columnas módulo y lenguaje. Son independientes entre sí.

Por lo tanto, existe una dependencia de valores múltiples, lo que conduce a la repetición innecesaria de datos y también a otras anomalías.

¿Qué debemos hacer para que la tabla esté en 4FN?

Para hacer que la relación anterior satisfaga la cuarta forma normal debemos descomponer la tabla en 2 tablas.

| MATRÍCULA_MÓDULOS | | MATRÍCULA LENGUAJES | |
|-------------------|--------|---------------------|----------|
| NIA | Módulo | NIA | Lenguaje |
| 0011 | PRG | 0011 | Java |
| 0011 | BDD | 0011 | C# |
| 0022 | EDD | 0022 | Java |
| 0022 | LMSGI | 0022 | C# |

Fig. 5.32. TABLA MATRÍCULA_MÓDULOS y MATRÍCULA LENGUAJES.

Una tabla también puede tener una dependencia funcional junto con una dependencia de varios valores. En ese caso, las columnas funcionalmente dependientes se mueven en una tabla separada y las columnas dependientes de valores múltiples se mueven a tablas separadas.

Estos problemas se pueden evitar fácilmente si diseñamos con cuidado nuestra su base de datos.

5.3. 5FN y 6FN

Una tabla está en la 5FN solo si está en 4FN, y no se puede descomponer en ninguna cantidad de tablas más pequeñas sin pérdida de datos.

La 6FN de momento no está estandarizada y no será objeto de estudio en esta unidad didáctica.

La 5FN y 6FN, en la normalización de bases de datos, generalmente no se implementan en el diseño de bases de datos de la vida real.

