

Tema

3

Modelo Relacional

En el capítulo anterior expusimos el modelado de datos empleando el modelo Entidad-Relación. En este capítulo introducimos el concepto del modelo relacional que está relacionado al modelo funcional de base de datos. Una enumeración de los tópicos que cubriremos en este capítulo son:

- Visión General del Modelo Relacional
- Estructura de Datos y Terminología
- Notación

3.1 Visión General del Modelo Relacional.

El modelo relacional fue formalmente introducido por el Dr. E. F. Codd en 1970, y ha evolucionado desde entonces debido a una serie de escritos. El modelo provee un concepto simple, pero rigurosamente definido, de cómo los usuarios perciben los datos. El modelo de datos representa la data en forma de tablas bidimensionales. Cada tabla representa alguna persona, lugar, cosa o evento del mundo real, acerca de la cual se recolecta información. Una base de datos relacional es una colección de tablas bi-dimensionales. La organización de la data en tablas relacionales se le conoce con el nombre de Logical View de la base de datos, es decir, la forma en la cual la base de datos relacional presenta datos al usuario y al programador. La forma en que el software de base de datos almacena físicamente en el disco de un sistema de computación se le llama Internal View. El Internal View varía de un producto a otro, y no nos concierne en este Tema III.

Es necesaria una comprensión básica del modelo relacional para efectivamente un software de base de datos relacional como Oracle, Microsoft SQL Server, o aún sistemas de bases de datos personales como Microsoft Access o FoxPro, los cuales están basados en el modelo relacional.

El contenido de este tema no constituye una completa descripción del modelo relacional, por lo que se le sugiere al estudiante investigar más profundamente sobre el tema.

3.2. Estructura de Datos y Terminología.

En el modelo relacional, una base de datos es una colección de tablas relacionales. Una tabla relacional es un archivo compuesto de un conjunto de columnas, cada una de ellas con su nombre, y un número arbitrario de filas o registros sin nombres. Los registros de la tabla representan ocurrencias de una “cosa” a la cual la tabla representa. Un valor de datos se almacena en la intersección de una fila y una columna. Cada columna tiene un dominio, el cual representa el conjunto de valores que pueden aparecer en la columna. A continuación les mostramos las tablas relacionales de una base de datos bibliográfica simple, que almacena información acerca de títulos de libros, autores de libros, y editores de libros.

A Relational Data Base

AUTHOR

| au_id | au_lname | au_fname | address | city | state |
|-------------|------------|----------|----------------------|---------------|-------|
| 172-32-1176 | White | Johnson | 10932 Bigge Rd. | Menlo Park | CA |
| 213-46-8915 | Green | Marjorie | 309 63rd St. #411 | Oakland | CA |
| 238-95-7766 | Carson | Cheryl | 589 Darwin Ln. | Berkeley | CA |
| 267-41-2394 | O'Leary | Michael | 22 Cleveland Av. #14 | San Jose | CA |
| 274-80-9391 | Straight | Dean | 5420 College Av. | Oakland | CA |
| 341-22-1782 | Smith | Meander | 10 Mississippi Dr. | Lawrence | KS |
| 409-56-7008 | Bennet | Abraham | 6223 Bateman St. | Berkeley | CA |
| 427-17-2319 | Dull | Ann | 3410 Blonde St. | Palo Alto | CA |
| 472-27-2349 | Gringlesby | Burt | PO Box 792 | Covelo | CA |
| 486-29-1786 | Locksley | Charlene | 18 Broadway Av. | San Francisco | CA |

TITLE

| title_id | title | type | price | pub_id |
|----------|-------------------------------------|--------------|-------|--------|
| BU1032 | The Busy Executive's Database Guide | business | 19.99 | 1389 |
| BU1111 | Cooking with Computers | business | 11.95 | 1389 |
| BU2075 | You Can Combat Computer Stress! | business | 2.99 | 736 |
| BU7832 | Straight Talk About Computers | business | 19.99 | 1389 |
| MC2222 | Silicon Valley Gastronomic Treats | mod_cook | 19.99 | 877 |
| MC3021 | The Gourmet Microwave | mod_cook | 2.99 | 877 |
| MC3026 | The Psychology of Computer Cooking | UNDECIDED | | 877 |
| PC1035 | But Is It User Friendly? | popular_comp | 22.95 | 1389 |
| PC8888 | Secrets of Silicon Valley | popular_comp | 20 | 1389 |
| PC9999 | Net Etiquette | popular_comp | | 1389 |
| PS2091 | Is Anger the Enemy? | psychology | 10.95 | 736 |

PUBLISHER

| pub_id | pub_name | city |
|--------|-----------------------|------------|
| 736 | New Moon Books | Boston |
| 877 | Binnet & Hardley | Washington |
| 1389 | Algodata Infosystems | Berkeley |
| 1622 | Five Lakes Publishing | Chicago |
| 1756 | Ramona Publishers | Dallas |
| 9901 | GGG&G | München |
| 9952 | Scootney Books | New York |
| 9999 | Lucerne Publishing | Paris |

AUTHOR_TITLE

| au_id | title_id |
|-------------|----------|
| 172-32-1176 | PS3333 |
| 213-46-8915 | BU1032 |
| 213-46-8915 | BU2075 |
| 238-95-7766 | PC1035 |
| 267-41-2394 | BU1111 |
| 267-41-2394 | TC7777 |
| 274-80-9391 | BU7832 |
| 409-56-7008 | BU1032 |
| 427-17-2319 | PC8888 |
| 472-27-2349 | TC7777 |

Existen nombres alternativos usados para describir las tablas relacionales. Algunos manuales usan los términos tablas, campos, y registros para describir tablas, columnas y filas relacionales, respectivamente. La literatura formal de de base de datos tiende a usar términos matemáticos, relaciones atributos, y tuplas.

3.3 Notación.

Las tablas relacionales pueden ser expresadas concisamente eliminando la data de ejemplo, y mostrando tan sólo el nombre de la tabla y los nombres de columnas.

AUTHOR (au_id, au_lname, au_fname, address, city, state, zip)

TITLE (title_id, title, type, price, pub_id)

PUBLISHER (pub_id, pub_name, city)

AUTHOR_TITLE (au_id, title_id)

3.4 Tablas Relacionales.

Las tablas relacionales tienen 6 propiedades:

1. Los valores son atómicos.
2. Los valores de columnas son de un mismo tipo.
3. Cada fila es única.
4. La secuencia de columnas es insignificante.
5. Cada columna debe tener un nombre único.

LOS VALORES SON ATOMICOS.

Esta propiedad implica que las columnas en una tabla relacional no son grupos repetitivos o arreglos. A las tablas que cumplen con esta propiedad se dice que están en “primera forma normal” (1FN). La propiedad de valor atómico de las tablas relacionales es importante debido a que es una de las piedras angulares del modelo relacional.

El gran beneficio del valor atómico es que simplifica la lógica a emplear para manipulación de data.

LOS VALORES DE LAS COLUMNAS SON DEL MISMO TIPO.

En términos relacionales, esto significa que todos los valores en una columna vienen del mismo dominio. Un dominio es un conjunto de valores que una columna puede tener. Por ejemplo, una columna denominada “salario mensual” contiene sólo salarios mensuales específicos. Nunca contendrá otra información, tal como comentarios, salarios semanales, estatus, etc.

Esta propiedad simplifica el acceso de datos debido a que los desarrolladores y usuarios pueden estar seguros del tipo de datos contenido en una columna dada. Esto también simplifica la validación de data. Debido a que todos los

valores son del mismo dominio, el dominio puede ser definido e impuesto con el Data Definition Language (DDL) del software de base de datos.

CADA FILA ES UNICA.

Esta propiedad se asegura de que no haya dos filas idénticas en una tabla relacional; por lo menos existe una columna, o conjunto de columnas, cuyos valores identifica de manera única cada fila en la tabla.

LA SECUENCIA DE LAS COLUMNAS ES INSIGNIFICANTE.

Esta propiedad establece que el orden de las columnas en la tabla relacional no tiene significado. Las columnas pueden ser traídas o usadas en cualquier orden y en varias secuencias. El beneficio de esta propiedad es que habilita que múltiples usuarios puedan compartir la misma tabla sin preocuparse como la tabla está organizada. Esto también permite que la estructura física de la base de datos cambie sin afectar las tablas relacionales.

LA SECUENCIA DE FILAS ES INSIGNIFICANTE.

Esta propiedad es análoga a la anterior, solo que aplica a las filas en vez de a las columnas. El beneficio está en que las filas de una tabla relacional pueden ser llamadas en distinto orden y secuencias. Adicionar información a una tabla relacional se simplifica y no afecta queries existentes.

CADA COLUMNA TIENE UN NOMBRE UNICO.

Debido a que las secuencia de columnas son insignificantes, las columnas deben ser referenciadas por su nombre y no por su posición. En general, un nombre de columna debe ser único dentro de la tabla a que pertenece, pero no tiene que ser único en la base de datos completa.

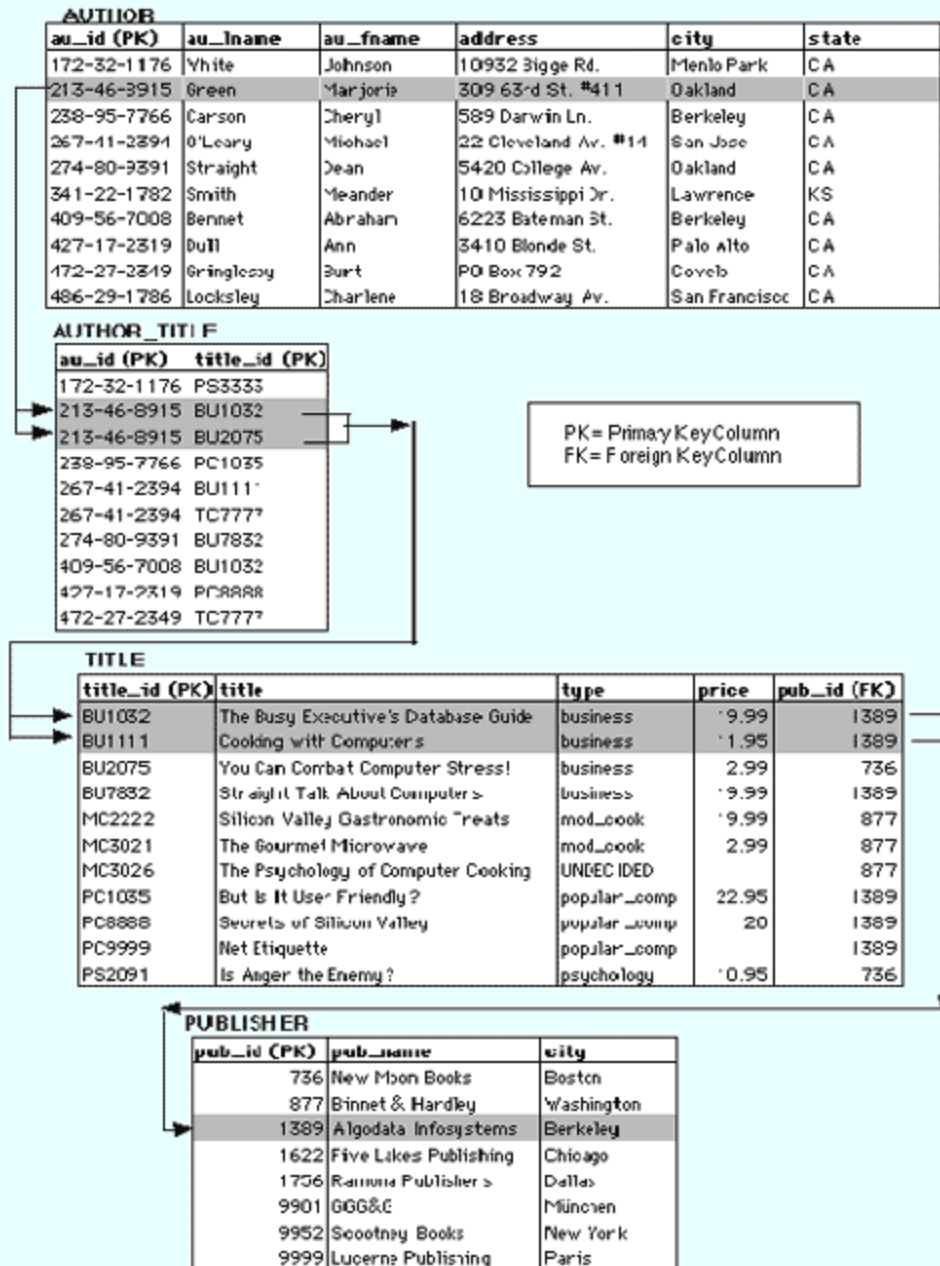
2.4 Relaciones y Llaves.

Una relación es una asociación entre 2 o más tablas. Las relaciones son expresadas en los valores de datos de las llaves primarias y foráneas.

Una llave primaria es una columna o columnas en una tabla cuyos valores identifican de forma única cada fila en una tabla. Una llave foránea es una columna o columnas cuyos valores son los mismos que la llave primaria de otra tabla. Usted puede pensar en una llave foránea como una copia de la llave primaria de otra tabla relacional. La relación se hace entre 2 tablas relacionales al igualar los valores de la llave foránea en una tabla con los valores de la llave primaria en otra tabla.

Las llaves son fundamentales en el concepto de base de datos relacional debido a que habilitan que las tablas de una base de datos se relacionen unas con otras. La navegación por una base de datos relacional depende de la habilidad de la llave primaria de identificar cada fila de una tabla sin ambigüedad. Navegar entre tablas requiere que la llave foránea sea capaz de referenciar valores de las llaves primarias de la tabla relacionada, de forma correcta y consistente.

Navigating Between Tables Using Keys



2.5 Integridad de Datos

Integridad de datos significa, en parte, que usted puede navegar y manipular las tablas de una en la base de datos de forma correcta y consistente. Existen dos

reglas básicas para asegurar la integridad de datos: integridad de entidad e integridad referencial.

La integridad de entidad establece que el valor de la llave primaria nunca podrá ser un valor null. Debido a que la llave primaria es usada para identificar una fila como única en una tabla relacional, su valor debe ser siempre especificado y nunca deberá ser desconocido. Esta regla requiere mantener el carácter único de todas las llaves primarias.

La integridad referencial establece que si una tabla relacional tiene una llave foránea, entonces cada valor de la llave foránea debe o ser null o ser igual a uno de los valores en la tabla relacional en la cual la llave foránea es una llave primaria.

3.6. Manipulación de Data Relacional.

Las tablas relacionales son conjuntos. Las filas (registros) pueden ser considerados como elementos del conjunto. Las operaciones que son efectuadas sobre conjuntos pueden también ser ejecutadas en tablas relacionales. Las 8 operaciones relacionales son:

UNION.

La operación Unión de 2 tablas relacionales está formada por la adición de los registros de una tabla a los registros de la otra tabla para producir una tercera tabla. Las filas duplicadas son eliminadas. La notación para la unión de las tablas A y B es A UNION B.

Las tablas relacionales usadas en la operación Unión deben ser compatibles para la unión: deben tener el mismo número de columnas, y las columnas correspondientes deben ser del mismo dominio.

| A | B | A UNION B | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|-----------|---|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|---|---|---|---|---|---|----|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|
| <table><tr><th>k</th><th>x</th><th>y</th></tr><tr><td>1</td><td>A</td><td>2</td></tr><tr><td>2</td><td>B</td><td>4</td></tr><tr><td>3</td><td>C</td><td>6</td></tr></table> | k | x | y | 1 | A | 2 | 2 | B | 4 | 3 | C | 6 | <table><tr><th>k</th><th>x</th><th>y</th></tr><tr><td>1</td><td>A</td><td>2</td></tr><tr><td>4</td><td>D</td><td>8</td></tr><tr><td>5</td><td>E</td><td>10</td></tr></table> | k | x | y | 1 | A | 2 | 4 | D | 8 | 5 | E | 10 | <table><tr><th>k</th><th>x</th><th>y</th></tr><tr><td>1</td><td>A</td><td>2</td></tr><tr><td>2</td><td>B</td><td>4</td></tr><tr><td>3</td><td>C</td><td>6</td></tr><tr><td>4</td><td>D</td><td>8</td></tr><tr><td>5</td><td>E</td><td>10</td></tr></table> | k | x | y | 1 | A | 2 | 2 | B | 4 | 3 | C | 6 | 4 | D | 8 | 5 | E | 10 |
| k | x | y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | A | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | B | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | C | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| k | x | y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | A | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | D | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | E | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| k | x | y | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | A | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | B | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | C | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | D | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | E | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

DIFERENCIA.

La diferencia de dos tablas relacionales es una tercera tabla que contiene filas que ocurren en la primera tabla pero no en la segunda. La operación diferencia requiere que las tablas sean compatibles para Unión. Tal como ocurre en aritmética, el orden de la substracción si importa: $A - B$, no es lo mismo que $B - A$.

| A - B | | |
|-------|---|---|
| k | x | y |
| 2 | B | 4 |
| 3 | C | 6 |

| B - A | | |
|-------|---|----|
| k | x | y |
| 4 | D | 8 |
| 5 | E | 10 |

INTERSECCIÓN.

La intersección de dos tablas relacionales es una tercera tabla que contiene filas comunes. Ambas tablas deben ser compatibles para Unión. La notación para la intersección de A y B es $A \text{ [intersección] } B = C$ o A intersección B.

| A | | |
|---|---|---|
| k | x | y |
| 1 | A | 2 |
| 2 | B | 4 |
| 3 | C | 6 |

| B | | |
|---|---|----|
| k | x | y |
| 1 | A | 2 |
| 4 | D | 8 |
| 5 | E | 10 |

| A INTERSECT B | | |
|---------------|---|---|
| k | x | y |
| 1 | A | 2 |

PRODUCTO.

El producto de dos tablas relacionales (que también recibe el nombre de producto cartesiano, es la concatenación de cada fila en una tabla con cada fila en la segunda. El producto de la tabla A y la tabla B es la tabla C (tendrá $m \times n$ filas). El producto se denota como $A \times B$ o A VECES B.

La operación producto es por si misma no muy útil. Sin embargo, es a menudo usada como un proceso intermedio para un Join.

| A | | | B | | | A TIMES B | | | | | |
|---|---|---|---|---|----|-----------|----|----|----|----|----|
| k | x | y | k | x | y | ak | ax | ay | bk | bx | by |
| 1 | A | 2 | 1 | A | 2 | 1 | A | 2 | 1 | A | 2 |
| 2 | B | 4 | 4 | D | 8 | 1 | A | 2 | 4 | D | 8 |
| 3 | C | 6 | 5 | E | 10 | 1 | A | 2 | 5 | E | 10 |
| | | | | | | 2 | B | 4 | 1 | A | 2 |
| | | | | | | 2 | B | 4 | 4 | D | 8 |
| | | | | | | 2 | B | 4 | 5 | E | 10 |
| | | | | | | 3 | C | 6 | 1 | A | 2 |
| | | | | | | 3 | C | 6 | 4 | D | 8 |
| | | | | | | 3 | C | 6 | 5 | E | 10 |

PROYECCIÓN.

Una proyección de una tabla trae un subconjunto de columnas de una tabla, eliminando filas duplicadas del resultado.

SELECCIÓN.

Este operador trae un subconjunto de filas de una tabla relacional basándose en un valor (valores) en una columna o columnas. A esta operación a veces se le llama Restringir para evitar confusión con el comando SQL SELECT.

JOIN.

Una operación JOIN combina el producto, selección, y posiblemente, la proyección. El operador Join combina horizontalmente (concatena) data de una fila de una tabla con filas de otra tabla o de la misma tabla cuando ciertos criterios se cumplen. Si el criterio se basa en la igualdad del valor de una columna, el resultado se le conoce como equijoin. Un join natural es un equijoin con las columnas redundantes eliminadas.

Los Joins también pueden efectuarse con criterios de selección distintos de la igualdad.

| D | | | E | | Equijoin | | | | |
|---|---|----|---|----|----------|---|----|---|----|
| k | x | y | k | z | k | x | y | k | z |
| 1 | A | 2 | 1 | 20 | 1 | A | 2 | 1 | 20 |
| 2 | B | 4 | 4 | 24 | 4 | D | 8 | 4 | 24 |
| 3 | C | 6 | 5 | 28 | 5 | E | 10 | 5 | 28 |
| 4 | D | 8 | 7 | 32 | | | | | |
| 5 | E | 10 | 9 | 36 | | | | | |

| Natural Join | | | |
|--------------|---|----|----|
| k | x | y | z |
| 1 | A | 2 | 20 |
| 4 | D | 8 | 24 |
| 5 | E | 10 | 28 |

DIVISIÓN.

El resultante de una operación de división son valores de columnas en una tabla para los que existen otras columnas cuyos valores se corresponden con cada fila de otra tabla.

| A | | | B (divisor) | | Result | |
|----|------|---|-------------|---|--------|--|
| k | x | y | x | y | k | |
| 10 | 1101 | A | 1101 | A | 10 | |
| 10 | 1201 | B | 1201 | B | 30 | |
| 10 | 1301 | C | 1301 | C | | |
| 20 | 1201 | B | | | | |
| 30 | 1101 | A | | | | |
| 30 | 1201 | B | | | | |
| 30 | 1301 | C | | | | |

3.7. Normalización.

Normalización es una técnica de diseño ampliamente usada como guía en el diseño relacional de base de datos. La normalización es esencialmente un proceso de 2 pasos que pone data en forma tabular al eliminar grupos repetitivos y luego eliminando duplicados de las tabla relacionales.

La teoría de normalización se basa en los conceptos de formas normales. Una tabla relacional se dice que está en una forma normal particular si cumple ciertos constreñimientos. Existen en la actualidad de 5 formas normales definidas, las 3 primeras fueron creadas por E. F. Codd.

OBJETIVO DE LA NORMALIZACION.

El objetivo de la normalización es crear un conjunto de tablas relacionales que están libres de data redundante y pueden ser actualizadas de manera correcta y consistente.

PRIMERA FORMA NORMAL.

Una tabla relacional se encuentra en primera forma normal si cumple lo siguiente:

- No existen grupos repetitivos
- Todos los atributos claves están definidos
- Todos los atributos dependen de la llave primaria.
- Los valores de los campos son atómicos.

Si tenemos una estructura de datos como la que sigue:

- Número de proyecto
- Nombre de proyecto
- 1-n número de empleados (1-n significa varias ocurrencias de este campo)
- 1-n nombres de empleados
- Categoría empleado
- Paga por hora

Para llevar esta estructura a Primera Forma Normal (1FN) debemos eliminar los grupos repetitivos y definir la llave primaria. En 1FN tendríamos:

Tabla Empleado-Proyecto

- Número de proyecto *
- Nombre proyecto
- Número empleado *
- Nombre empleado
- Categoría empleado
- Paga por hora

SEGUNDA FORMA NORMAL.

Una se encuentra en 2FN si está en 1FN y no incluye dependencias parciales (atributos que dependen de sólo parte de la llave primaria).

Siguiendo con nuestro ejemplo el nombre del proyecto depende sólo del número del proyecto. Nombre de empleado, categoría y paga por hora son dependientes

únicamente del número de empleado. Por esta razón eliminamos estos campos y los colocamos en tablas separadas, y en dichas tablas los campos claves son aquellos en que realmente ellos dependen.

Tabla Empleado-Proyecto

- Número de proyecto *
- Número empleado *

Tabla Empleado

- Número empleado *
- Nombre empleado
- Categoría
- Paga por hora

Tabla de proyecto

- Número proyecto
- Nombre de proyecto

TERCERA FORMA NORMAL.

Una tabla se encuentra en 3FN si se encuentra en 2FN, y no tiene dependencias transitivas (no existen atributos que no son claves dependientes de otros atributos no claves).

Siguiendo con nuestro ejemplo, la única tabla con más de un atributo no clave es Empleado. El nombre de empleado no es dependiente de la categoría ni de la paga por hora, y lo mismo aplica al atributo categoría. Sin embargo, la paga por hora depende de la categoría de empleado, por lo que lo eliminamos y tal como hicimos con los campos anteriores se coloca en una tabla aparte.

Tabla Empleado-Proyecto

- Número de proyecto *
- Número empleado *

Tabla Empleado

- Número empleado *
- Nombre empleado
- Categoría

Tabla Categoría

- Categoría *
- Paga por hora

Tabla de proyecto

- Número proyecto
- Nombre de proyecto

Con esto nuestro ejemplo queda en 3FN, y listo para implementarse como base de datos. Existen otras formas normales, tales como la 4FN de Boyce-Codd, pero son rara vez usadas en aplicaciones de negocios, y en muchos casos las tablas en 3FN ya cumplen con las 4FN y 5FN.