

BASES DE DATOS



EL MODELO

RELACIONAL

1 Introducción

El modelo relacional fue introducido a finales de los 60 por Edgar Frank **Codd** basándose en los trabajos de los matemáticos Cantor y Childs cuya teoría de conjuntos es la base del Modelo Relacional. Con este modelo Codd intenta que los usuarios no tengan que conocer el funcionamiento físico de la base de datos y puedan trabajar independientemente de él. Oracle llevó a cabo sus teorías y hoy casi todas las bases de datos siguen este modelo

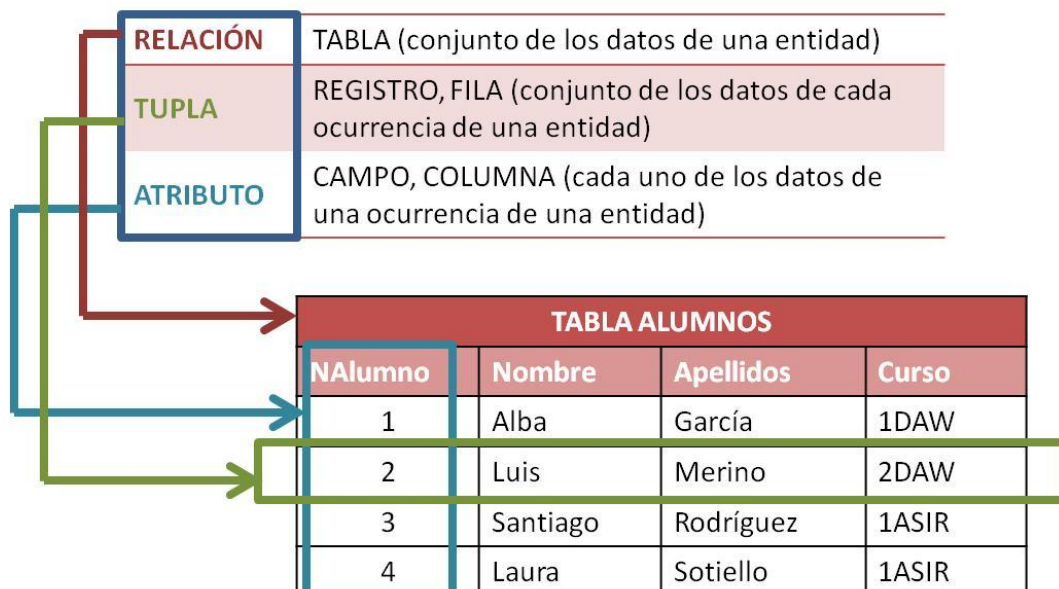
OBJETIVOS:

- **Independencia física:** que la forma en que se vayan a almacenar físicamente los datos no influya en la forma en que se va a trabajar con ellos lógicamente. De esta forma puede cambiar el almacenamiento físico sin que tengamos que cambiar el diseño lógico.
- **Independencia lógica:** que un cambio en la base de datos no implique que haya que cambiar las aplicaciones que utilizan esa base de datos.

Representación lógica	Representación física	Modelo relacional
Tabla	Archivo secuencial	Relación
Fila	Registro	Tupla
Columna	Campo	Atributo

- **Flexibilidad:** la BD tendrá distintas vistas en función de los usuarios y aplicaciones.
- **Uniformidad:** siempre se usa la misma forma conceptual (las tablas) para las estructuras lógicas.
- **Sencillez.** todo lo anterior junto con unos lenguajes de usuario muy sencillos, tienen como resultado que el modelo de datos relacional sea fácil de comprender y de utilizar por parte del usuario final

2 Terminología del Modelo Relacional



2.1 Relación o tabla

El modelo de datos relacional organiza y representa los datos en forma de tablas o relaciones, así una **Base de Datos Relacional** es una colección de relaciones (tablas).

id_trabajador	nombre	tarifa_hr	tipo_de_oficio	id_supv
1235	F. Aguilera	12,50	Electricista	1311
1412	A. Calvo	13,75	Fontanero	1540
2920	N. Marín	10,00	Carpintero	null
3231	O. Pons	17,40	Albañil	null
1540	J.M. Medina	11,75	Fontanero	null
1311	J.C. Cubero	15,50	Electricista	null
3001	D. Sánchez	8,20	Albañil	3231

CARACTERÍSTICAS

- Cada tabla tiene un nombre distinto de las demás.
- Todas las tuplas o filas tienen las mismas columnas.
- No hay filas o tuplas repetidas.
- Cada columna (atributo) es única y con nombre único en una tabla (si puede repetirse en otra tabla distinta).
- En cada celda de la tabla, intersección fila con columna, sólo puede haber un valor contenido en el dominio. Es válido el valor nulo o vacío (NULL) en algunos casos.
- No es significativo el orden de las tuplas (filas) ni el de los atributos (columnas).

2.2 Tuplas o filas de la relación

Referido a cada elemento de la relación. Por ejemplo si una relación almacena datos de personas, una tupla representaría la información una persona en concreto.

Cada una de las filas de la relación. Se corresponde con la idea clásica de registro. Representa cada elemento individual de esa relación. Tiene que cumplir que:

- Cada tupla se debe corresponder con un elemento del mundo real.
- No puede haber dos tuplas iguales (con todos los valores iguales).

2.3 Atributos o columnas de la relación

Cada propiedad de los datos que se almacenan en la relación (dni, nombre, dirección,...).

DOMINIO: Un dominio contiene todos los posibles valores que puede tomar un atributo. Hay dos tipos de dominios: generales y restringidos.

Generales o continuos: Sus valores suelen estar comprendidos entre un máximo y un mínimo. Por ejemplo:

- Para el campo NOTA el dominio podría estar comprendido entre 0 y 10.
- Para el campo SALDO DE CUENTA el dominio podría ser cualquier cantidad positiva, negativa ó 0.

Restringidos o discretos: Sus valores están incluidos en un conjunto de valores previamente definidos. Por ejemplo:

- Para el campo SEXO, los valores posibles serían hombre o mujer.
- Para el campo CATEGORÍA, se podrían limitar los valores a infantil, juvenil y sénior.

Hay dos formas de indicar el contenido de un dominio:

Por **definición**. Se define el dominio con la descripción exacta de sus posibles valores. Por ejemplo se definir el dominio de edades de los trabajadores como: *números enteros entre el 16 y el 65 (un trabajador sólo podría tener una edad entre 16 y 65 años)*.

Por **extensión**. Se indican algunos valores y se sobrentiende el resto. Por ejemplo el dominio localidad se podría definir por extensión con: *Palencia, Valladolid, Gijón,...*

GRADO: Indica el tamaño de una relación. Es el número de atributos (columnas) de la misma. En la tabla alumnos anterior el grado es 4 y en la de trabajadores el grado es 5.

CARDINALIDAD: Es el número de tuplas de una relación o el número de filas de la tabla. En el ejemplo anterior de los alumnos la cardinalidad sería 4 y para la tabla trabajadores sería 7.

2.4 Claves

Clave candidata

La componen uno o más atributos de la relación. No admite valores nulos e identifica inequívocamente a cada tupla de la relación, es decir, no puede haber más de una fila con el mismo valor de la clave primaria.

Toda tabla en el modelo relacional debe tener al menos una clave candidata (puede haber más).

Ejemplos: NIF, matrícula de un coche, número de bastidor de un motor, etc.

Clave primaria

Es la clave candidata que se elige como identificador principal, en la fase de diseño de la tabla. El diseñador de la base de datos, elige como primaria la candidata que identifique mejor a cada tupla en el contexto de la organización que se esté representando. Una vez escogida no se cambia. Debe existir siempre.

Por ejemplo, en una tabla de clientes que tuviera un atributo con el NIF y un atributo de código de cliente, ambos serían clave candidata. Elegiríamos entre ambas aquella que fuera la mejor para el contexto en el que estuviéramos trabajando.

Clave alternativa

Cada una de las claves candidatas que no fueron elegidas como clave primaria.

Clave externa, ajena o foránea

Clave ajena en una tabla (tabla hijo) es conjunto de atributos de una tabla que es clave primaria en otra tabla (tabla padre).

Las claves ajenas permiten RELACIONAR tablas. Cada vez que se define una clave ajena, se está asegurando la aplicación de restricciones de integridad referencial en sus valores:

- Su dominio tiene que ser el mismo que el de la clave primaria en la otra tabla, la tabla padre.
- Muy importante: NO puede haber en la tabla hijo un valor de la clave ajena distinto a los que ya tiene la clave primaria en la tabla padre.
- Sus valores pueden estar duplicados o ser nulos.
- NO es obligatorio que los nombres de los atributos sean iguales en las dos tablas (aunque es conveniente por claridad).

TABLA ALUMNOS				TABLA CICLOS		
<u>NAlumno</u>	Nombre	Apellidos	Curso	<u>CodCurso</u>	Nombre	Nivel
1	Alba	García	1DAW	1DAW	Diseño Aplicaciones Web	Primero
2	Luis	Merino	2DAW	2DAW	Diseño Aplicaciones Web	Segundo
3	Santiago	Rodríguez	1DAM	1DAM	Diseño Aplicaciones Multimedia	Primero
4	Laura	Sotiello	1DAM	2DAM	Diseño Aplicaciones Multimedia	Segundo

3 Restricciones del Modelo Relacional

- Restricciones inherentes al modelo relacional

Son aquellas que NO son definidas por los usuarios, sino que son determinadas por el hecho de que la base de datos sea relacional. Son las condiciones que han de cumplirse obligatoriamente y que diferencian una relación o tabla del modelo relacional de otro tipo de tabla. Son las definidas en “Características de una relación o tabla”.

- Restricción de dominio

Asegura que los valores que puede tomar un atributo pertenezcan a su dominio.

- Restricción de integridad de clave primaria

Asegura que se van a cumplir las condiciones que una clave primaria lleva asociadas cuando ésta se define en una tabla. Son las que definimos en “clave candidata” y “clave primaria”. Para cada clave primaria, el SGBD crea un índice principal con los valores de la clave primaria que tiene la tabla en cada momento. Este índice permite localizar cada fila de la tabla de una forma rápida a través de un valor de la clave primaria.

- Restricción de integridad de clave alternativa

Asegura que se van a cumplir las condiciones que una clave alternativa lleva asociadas cuando ésta se defina en una tabla. Son las que definimos en “clave candidata” y “clave primaria”. Para cada clave alternativa, el SGBD crea un índice secundario con los valores de la clave alternativa que tiene la tabla en cada momento. Este índice permite localizar cada fila de la tabla de una forma rápida a través de un valor de la clave alternativa.

- Restricción de obligatoriedad (NOT NULL)

Asegura que un atributo no pueda tener valor nulo, o lo que es lo mismo, sea obligatorio proporcionar un valor a ese atributo.

- Restricción de integridad referencial o restricción de clave ajena

Asegura que NO puede haber en la tabla hijo un valor de la clave ajena distinto a los que ya tiene la clave primaria en la tabla padre. También se permite que una clave ajena tenga el valor nulo (NULL). Esta restricción se comprueba cada vez que se introduce algún valor para la clave ajena.

Por ejemplo, en la relación anterior ALUMNOS y CICLOS, si se intenta introducir en ALUMNOS un código de curso que no está ya metido en la relación CICLOS, como pueda ser 1ASIR, se está incumpliendo la restricción de integridad referencial dada por esa clave ajena CURSO y el SGBD no dejaría hacer la inserción, notificándolo con un error.

Además de definir las claves ajenas, tenemos que tener en cuenta las posibles operaciones de borrado y modificación de los valores de la clave en la tabla padre (CICLOS), pues podríamos tener filas en la tabla hija (ALUMNOS) haciendo referencia a un valor que ya no existe.

Para solventar esta situación, si se produce el borrado y/o modificación de una tupla en la relación padre (relación con la clave primaria) se puede hacer uso de estas opciones:

- **Borrado y/o modificación restringido** es decir, prohibir la operación (NO ACTION O RESTRICT) en la tabla padre, si existen tuplas relacionadas en la tabla hija (la que contiene la clave ajena). En el caso de ALUMNOS y CICLOS, no se podrá borrar un CICLO o modificar su clave, si tiene ALUMNOS.
- **Borrado y/o modificación en cascada** (CASCADE). En el caso de ALUMNOS y CURSOS, si se borra un ciclo en la tabla CICLOS, se borrarán todas las filas de la tabla ALUMNOS que tengan en la clave ajena el ciclo borrado. Igualmente ocurrirá si se modifica el código de ciclo de la tabla CICLOS, esa modificación se arrastra a los alumnos de ese ciclo.
- **Borrado y/o modificación colocando nullos** (SET NULL). Si se borra un ciclo de la tabla CICLOS, a los alumnos de ese ciclo en la tabla ALUMNOS se les asignará NULL en el atributo de la clave ajena (Curso). Quedarán como alumnos sin ciclo.
- **Borrado y/o modificación colocando el valor por defecto** (DEFAULT). En este caso, el valor que se pone en las claves ajenas de la tabla referenciada es un valor por defecto que se habrá especificado en la creación de la tabla.

- Restricción de verificación o de validación (CHECK)

Obliga a que un valor concreto de un atributo cumpla una determinada condición para darlo por válido. Por ejemplo restringir el atributo sueldo para que siempre sea mayor de 1000, sería una regla de validación. También, por ejemplo, que la fecha de inicio sea mayor que la fecha final. Cada vez que se realiza una inserción o actualización de datos, se comprueba si los valores cumplen la condición.

- Aserciones

Son parecidas a la anterior, pero en lugar de afectar a los atributos de una relación, las condiciones se establecen sobre atributos de distintas relaciones.

- Disparadores, desencadenadores o triggers

Las restricciones anteriores son declarativas, pero esta es de tipo procedimental. Los disparadores son pequeños programas grabados en la base de datos, que se ejecutan automáticamente antes, después o en vez de que se produzca un cierto evento (cuando se añada una nueva fila, cuando se elimine una fila o cuando se actualice una fila de una determinada tabla). Permiten realizar restricciones muy potentes; aunque son más difíciles de crear.

Las 12 reglas de Codd

Codd se percató de que existían bases de datos en el mercado que decían ser relacionales pero lo único que hacían era guardar la información en tablas, sin estar estas tablas literalmente normalizadas. Entonces publicó 12 reglas que un verdadero sistema relacional debería tener aunque en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse “más relacional” cuanto más siga estas reglas.

Regla 0: el sistema debe ser relacional, base de datos y administrador de sistema. Ese sistema debe utilizar sus facilidades relacionales (exclusivamente) para manejar la base de datos.

1. Información. Toda la información de la base de datos debe estar representada mediante valores en tablas. Un valor posible es el valor nulo, con sus dos interpretaciones:

- Valor desconocido, por ejemplo una dirección desconocida.
- Valor no aplicable, por ejemplo un empleado soltero no tiene esposa.

2. Acceso garantizado. Todos los datos de una base de datos relacional son lógicamente accesibles a través de una combinación de nombre de tabla, valor de clave principal y nombre de columna (nombre de atributo).

3. Tratamiento sistemático de los valores nulos. El Gestor de Base de Datos debe permitir el tratamiento adecuado de estos valores. El valor nulo representa la ausencia de información de modo sistemático e independiente del tipo de dato.

4. Catálogo en línea basado en el modelo relacional. El catálogo o diccionario de la base de datos (metadatos) se guardan en tablas y se accede a ellos de la misma forma que los datos ordinarios, de modo que los usuarios autorizados para ello, pueden accederlos utilizando el mismo lenguaje relacional.

5. Sublenguaje de datos completo. Al menos debe de existir un lenguaje que permita el manejo completo de la base de datos. Este lenguaje, por lo tanto, debe permitir realizar cualquier operación sobre la misma.

6. Actualización de vistas. El Gestor de la Base de Datos debe encargarse de que las vistas muestren la información actualizada.

7. Inserciones, modificaciones y eliminaciones de alto nivel. El Gestor de Base de Datos debe permitir manejar una tabla o parte de ella, como un único operando en operaciones de consulta y actualización de datos.

8. Independencia física. Los datos deben de ser accesibles desde la lógica de la base de datos aun cuando se modifique el almacenamiento. La forma de acceder a los datos no varía porque el esquema físico de la base de datos cambie.

9. Independencia lógica. Los programas no deben verse afectados por cambios, a nivel lógico, en las tablas siempre y cuando se mantenga el esquema lógico.

10. Independencia de integridad. Las reglas de integridad deben almacenarse en la base de datos (en el diccionario de datos), no en los programas de aplicación.

11. Independencia de la distribución. El lenguaje que se utiliza para acceder a los datos, debe permitir que sus instrucciones funcionen de la misma forma, en una base de datos distribuida y en una que no lo es.

12. No subversión. Si un sistema relacional tiene un lenguaje de bajo nivel (un registro de cada vez), ese bajo nivel no puede ser usado para saltarse (subvertir) las reglas de integridad y las limitaciones expresadas en los lenguajes relacionales de más alto nivel (una relación, conjunto de registros, de cada vez).

Índice de contenidos

1 Introducción	2
2 Terminología del Modelo Relacional	2
2.1 Relación o tabla	3
2.2 Tuplas o filas de la relación	3
2.3 Atributos o columnas de la relación	3
2.4 Claves	4
3 Restricciones del Modelo Relacional	5