

Libro Capitulo 1

miércoles, 19 de febrero de 2025 07:28 p. m.

Las bases de datos relacionales ofrecen independencia de datos, lo que significa que un usuario no necesita saber en qué disco duro y archivo se almacena una determinada información. El RDBMS proporciona a los usuarios consistencia e integridad de los datos.

Una base de datos relacional almacena datos en tablas, que son básicamente una matriz bidimensional compuesta por columnas y filas.

Es un conjunto organizado de informacion que permite almacenar, gestionar y recuperar informacion de manera eficiente.

TABLAS

Una tabla normalmente contiene datos sobre un único tema. Cada tabla tiene un nombre único que indica el contenido de los datos. Una base de datos normalmente consta de muchas tablas. Por ejemplo, puede almacenar datos sobre libros que lee en una tabla llamada LIBRO y almacenar detalles sobre autores en la tabla AUTOR.

COLUMNAS-ATRIBUTOS

Las columnas de una tabla organizan aún más los datos y una tabla consta de al menos una columna. Cada columna representa un único detalle de bajo nivel sobre un conjunto particular de datos. El nombre de la columna es único dentro de una tabla e identifica los datos que se encuentran en la columna. Por ejemplo, la tabla BOOK puede tener una columna para el título, la editorial, la fecha de publicación del libro, etc. El orden de las columnas no es importante porque SQL le permite mostrar los datos en cualquier orden que elija.

FILAS-REGLONES-REGISTROS

Cada fila suele representar un conjunto único de datos dentro de una tabla. Por ejemplo, la fila con el título “La fuerza invisible” es única dentro de la tabla LIBRO. Todas las columnas de la fila representan los datos respectivos de la fila. Cada intersección de una columna y una fila en una tabla representa un valor, y algunas no. Esta compuesto por una fila que contiene datos de un elemento determinado.

NULL

Se dice que el valor es nulo. Nulo es un valor desconocido, por lo que ni siquiera son espacios en blanco. Los valores nulos no se pueden evaluar ni comparar porque son desconocidos.

CLAVE PRIMARIA

Al trabajar con tablas, debe comprender cómo identificar de forma única los datos dentro de una tabla. Una tabla puede tener solo una clave principal, que consta de una o más columnas. Si la clave principal contiene varias columnas, se denomina clave principal compuesta o clave principal concatenada. Oracle no exige que todas las tablas tengan una clave principal y puede haber casos en los que no sea adecuado tenerla. Sin embargo, se recomienda encarecidamente que la mayoría de las tablas tengan una clave principal. A veces, la clave única es un número de secuencia generado por el sistema; este tipo de clave se denomina clave sintética o sustituta. La ventaja de utilizar una clave sustituta es que es única y no tiene ningún significado o propósito inherente; por lo tanto, no está sujeta a cambios. Es mejor evitar las claves principales que estén sujetas a actualizaciones, ya que generan una complejidad innecesaria. Por ejemplo, el número de teléfono de un cliente es un mal ejemplo de una elección de columna de clave principal. Aunque posiblemente sea único dentro de una tabla, los números de teléfono pueden cambiar y luego causar una serie de problemas con las actualizaciones de otras columnas que hacen referencia a esta columna. No permiten valores NULL.

Atributo que nos permite identificar de forma unica cada registro en nuestra tabla.

CLAVE FORANEA

Es una relación entre dos tablas, se crea esa relación cuando lo que se quiere hacer es crear una regla de integridad referencial, es decir que un campo que no pueda tener ciertos valores al menos si y solo si los valores existen en la segunda tabla. Esa regla se crea para que no se graben valores en la primera tabla que no estén definidos en la segunda, eso da una consistencia de datos.

paciente

id	nombre	Pais paciente	
	pedro	ARG	
	abraham	MEX	

pais	Ubicación
EU	202.54
ARG	

La clave foránea debe de ser del mismo tipo de dato que su campo relacionado, el valor del campo definido como Foreign key puede ser NULL, una tabla puede tener mas de un campo FK. Una clave foránea (FOREIGN KEY) en SQL es un campo o conjunto de campos en una tabla que establece una relación con la clave primaria (PRIMARY KEY) de otra tabla. Esta relación es fundamental para mantener la integridad referencial en una base de datos, asegurando que los datos sean consistentes y que no existan registros huérfanos.

Funciones de la Clave Foránea

Vinculación de Tablas: La clave foránea permite conectar dos tablas, donde una tabla (llamada tabla hija o referendo) contiene la clave foránea que apunta a la clave primaria de otra tabla (llamada tabla padre o referenciada). Esto facilita la organización de datos relacionados y su recuperación. Integridad Referencial: Al establecer una clave foránea, se asegura que los valores insertados en la columna de la clave foránea deben existir en la columna referenciada de la tabla padre. Esto evita la inserción de datos inválidos y mantiene la coherencia en las relaciones entre tablas.

NORMALIZACION

la normalización es eliminar la redundancia en las tablas, evitando así futuros problemas de manipulación de datos. Existen distintas reglas para minimizar la duplicación de datos, que se formulan en las distintas formas normales. Las reglas verifican que las columnas que usted colocó en las tablas pertenecen de hecho a ellas. Usted diseña sus tablas, las columnas apropiadas y las claves primarias y externas correspondientes para cumplir con estas reglas. Este proceso se denomina normalización.

Primera forma normal "No datos repetidos"	Para que una tabla esté en su primera forma normal, se deben eliminar todos los grupos repetitivos y colocarlos en una nueva tabla. Se dice que una tabla está en Primera Form al Normal si y solo si todos su Campos (Atributos) contienen valores atómicos. Esto quiere decir que cada atributo de la tabla deberá tener un único valor para una ocurrencia de la Entidad, no se permitirán grupos repetitivos, no se permitirán registros iguales. Se puede añadir una key primaria para que no haya registros similares en todas sus características.
Segunda forma normal "todas las columnas deben de estar relacionadas con la clave primaria.	La segunda forma normal establece que todas las columnas que no son clave deben depender de la clave principal completa. Se aplica únicamente a las tablas que tienen claves principales compuestas. Una tabla está en la Segunda Forma Normal si y solo si está en 1FN y todos los atributos no clave dependen por completo de la clave primaria.
Tercer forma normal "Las columnas no clave no pueden relacionarse entre si"	La tercera forma normal va un paso más allá que la segunda forma normal, establece que cada columna que no sea clave debe ser un hecho sobre la clave principal. Una tabla est en la tercera Form al Normal y si y solo si está en 2FN y los atributos no clave son independientes entre sí. Esto quiere decir que los valores de los atributos dependen solo de la clave primaria y no dependen de otro atributo no clave. El valor del atributo no debe depender del valor de otro atributo no clave. Cada campo no clave no debe de estar relacionado con otro no clave.
FORMA NORMAL BOYCE-CODD (BCNF), CUARTA FORMA NORMAL Y QUINTA FORMA NORMAL	La BCNF es una versión elaborada de la tercera forma normal y se ocupa de las anomalías de eliminación. La cuarta forma normal aborda los problemas potenciales cuando tres o más columnas forman parte del identificador único y sus dependencias entre sí. La quinta forma normal divide las tablas aún más para eliminar toda redundancia.

DESCRIPCION GENERAL DE SQL

Se trabaja con tablas, filas y columnas mediante el lenguaje SQL. SQL permite consultar datos, crear datos nuevos, modificar datos existentes y eliminar datos. Dentro del lenguaje SQL se pueden diferenciar sublenguajes individuales, que son una colección de comandos individuales.

Por ejemplo, los comandos del lenguaje de manipulación de datos (DML,Data Manipulation Language) permiten consultar, insertar, actualizar y eliminar datos. SQL permite crear nuevas estructuras de bases de datos, como tablas, o modificar las existentes; esta subcategoría de comandos del lenguaje SQL se denomina lenguaje de definición de datos (DDL,Data Definition Language). Con el lenguaje SQL, se puede controlar el acceso a los datos mediante comandos del lenguaje de control de datos (DCL, Data Control Language).

DESCRIPTION	SQL COMMANDS
Data Manipulation	SELECT, INSERT, UPDATE, DELETE, MERGE
Data Definition	CREATE, ALTER, DROP,TRUNCATE, RENAME
Data Control	GRANT, REVOKE
Transaction Control	COMMIT, ROLLBACK, SAVEPOINT

Una de las primeras instrucciones que se utilizará en los ejercicios es el comando SELECT, que le permite consultar datos. El comando INSERT le permite agregar nuevas filas a una tabla. Para crear nuevas tablas, utilice el comando CREATE TABLE.

Puede manipular las definiciones de columnas de una tabla con el comando ALTER TABLE. Esto le permite agregar o eliminar columnas. También puede PRIMARY KEY y FOREIGN KEY en una tabla. Las restricciones imponen reglas de negocio dentro de la base de datos.

RELACIONES ENTRE TABLAS

Cuando dos tablas tienen una o más columnas en común, se dice que tienen una relación entre ellas. La cardinalidad de una relación es el número real de ocurrencias entre ellas.

RELACION DE UNO A MUCHOS (1:M)

Ocurre cuando un registro en una tabla A puede estar relacionado con ninguno o muchos registros en la tabla B, pero un registro en la tabla B solo puede estar relacionado con un registro en la tabla . Por ejemplo, una sucursal tiene muchos empleados, pero cada empleado pertenece solo a una sucursal. En una base de datos de seguimiento de pedidos, un cliente puede realizar varios pedidos. Un cliente individual puede tener muchas filas de pedidos en la tabla PEDIDO. Esta relación representa la regla de negocio que establece que “un cliente puede realizar uno o varios pedidos (o ningún pedido)”. Si leemos la relación en la otra dirección, un pedido se asocia con una sola fila de cliente (o ninguna fila de cliente). En otras palabras, “cada pedido puede ser realizado por un solo cliente”.

RELACION UNO A UNO (1:1)

Implica que por cada registro en una tabla, existe un único registro correspondiente en otra tabla. Es una relación relativamente rara, ya que si existe este tipo de relación, es posible que los datos de ambas tablas debieran estar en una sola tabla, ya que comparten la misma llave primaria. Este tipo de relación se puede usar para dividir una tabla con muchos campos, para aislar parte de una tabla por razones de seguridad o para almacenar información que solo se aplica a un subconjunto de la tabla principal. En el mundo de las bases de datos existen relaciones uno a uno, pero no son típicas porque, con frecuencia, los datos de ambas tablas se combinan en una sola tabla para simplificar.

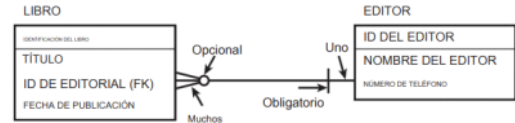
RELACION DE MUCHOS A MUCHOS (M:M)

Ocurre cuando un registro en una tabla A puede estar relacionado con ninguno o muchos registros en la tabla B, pero un registro en la tabla B solo puede estar relacionado con un registro en la tabla . Por ejemplo, una sucursal tiene muchos empleados, pero cada empleado pertenece solo a una sucursal. En una base de datos de seguimiento de pedidos, un cliente puede realizar varios pedidos. Un libro puede tener uno o más autores, y un autor puede escribir uno o más libros. El modelo de base de datos relacional requiere la resolución de relaciones de muchos a muchos en tablas de relaciones de uno a muchos. La solución más común se logra creando una tabla asociativa (también llamada tabla de intersección)

DIAGRAMAS DE ESQUEMAS DE BASE

Para mayor claridad de significado y coherencia conceptual, resulta útil mostrar las relaciones entre tablas mediante diagramas de esquema de base de datos. Existen varias notaciones estándar para este tipo de diagrama.

La convención utilizada para una relación de uno a muchos es una línea con una “pata de gallo” (bifurcación) en un extremo que indica el lado “de muchos” de la relación; en el otro extremo, una “línea simple” representa el lado “uno” de la relación



Se traduciría como: un editor puede publicar cero, uno o muchos libros, y, un libro debe ser publicado por un único editor.

El símbolo “(FK)” que aparece junto a la columna ID_DE EDITORIAL en la tabla BOOK indica que se trata de una columna de clave externa.

Ilustra una relación de uno a muchos entre el EDITORIAL (padre) y el LIBRO (hijo). Al examinar la línea de relación en el extremo “muchos”, se observa un círculo que identifica la relación opcional y una pata de gallo que indica “muchos”. Los símbolos indican que un editor puede publicar cero, uno o muchos libros. Se utiliza la palabra puede para indicar que la relación es opcional, y esto permite que exista un editor sin un valor correspondiente en la tabla LIBRO. La línea de relación también se lee al revés. La línea sólida en el extremo de la línea que corresponde al EDITORIAL indica el lado “uno” y una barra vertical la cruza. Esta barra identifica una relación obligatoria. Esta dirección de la relación se lee como “Un libro debe ser publicado por un único editor”. Esto significa que una fila en la tabla LIBRO siempre debe tener el valor ID DE EDITORIAL completo

En la siguiente figura se muestra una relación opcional en ambos lados; un libro puede ser publicado por cero o por un editor. En efecto, esto significa que el valor de la columna ID DE EDITORIAL en LIBRO es opcional. Al leer la relación desde EDITOR, se puede decir que “un editor puede publicar cero, uno o muchos libros”

CARDINALIDAD Y OPCIONALIDAD

La cardinalidad expresa la relación entre una tabla principal y una secundaria desde la perspectiva de la tabla principal. Describe cuántas filas puede encontrar entre las dos tablas para un valor de clave principal determinado. La cardinalidad se refiere a la cantidad de instancias de una entidad que pueden estar asociadas con instancias de otra entidad.

-Uno a uno (1:1): Cada registro en una tabla se relaciona con un único registro en otra tabla. Por ejemplo, una persona puede tener solo un pasaporte y cada pasaporte pertenece a una sola persona.

-Uno a muchos (1:n): Un registro en una tabla puede relacionarse con múltiples registros en otra tabla. Por ejemplo, un autor puede haber escrito varios libros, pero cada libro tiene un solo autor.

-Muchos a uno (n:1): Es la inversa de la relación uno a muchos. Varios registros en una tabla pueden relacionarse con un único registro en otra tabla.

-Muchos a muchos (n:m): Varios registros en una tabla pueden relacionarse con varios registros en otra tabla. Por ejemplo, los estudiantes pueden inscribirse en múltiples cursos, y cada curso puede tener múltiples estudiantes. Para manejar esta relación, se utiliza una tabla intermedia que contiene las claves foráneas de ambas tablas involucradas

Las líneas de relación gráfica también indican la opcionalidad de una relación, ya sea que una fila sea obligatoria o no (obligatoria u opcional). En concreto, la opcionalidad muestra si una fila de una tabla puede existir sin una fila en la tabla relacionada. La opcionalidad indica si la participación de una entidad en una relación es obligatoria o no. Se clasifica generalmente como:

- Obligatoria: La existencia de un registro en una tabla requiere que haya un registro correspondiente en la tabla relacionada. Por ejemplo, si un pedido debe estar asociado a un cliente, entonces la relación es obligatoria.
- Opcional: La existencia de un registro en una tabla no requiere necesariamente que haya un registro correspondiente en la tabla relacionada. Por ejemplo, si un cliente puede realizar pedidos o no, la relación es opcional.

NOTA: Cuando una columna se define como NOT NULL, siempre debe contener un valor. Cuando una columna de una fila se define como que permite valores NULL, significa que una columna no necesita contener un valor.

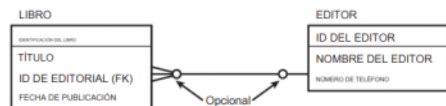
Normalmente se ven solo dos tipos de relaciones: primero, obligatoria en el lado “uno” y opcional en el extremo “muchos”; y segundo, opcional en ambos extremos. Solo en raras ocasiones se encuentran otros tipos de relaciones. Por ejemplo, las relaciones obligatorias en ambos lados se implementan con poca frecuencia; esto significa que las filas deben insertarse en ambas tablas simultáneamente. Ocasionalmente, se encuentran relaciones uno a uno, pero la mayoría de las veces, las columnas de ambas tablas se combinan en una sola tabla. Las relaciones de varios a varios no están permitidas en la base de datos relacional; deben resolverse mediante una tabla asociativa o de intersección en relaciones de uno a varios.

RELACIONES DE ETIQUETADO

Para aclarar y explicar la naturaleza de una relación en un diagrama, es útil agregar en la línea de relación una etiqueta o nombre con un verbo. La Figura muestra un ejemplo de una relación etiquetada. Para mayor claridad, una relación debe estar etiquetada en ambos lados. Entonces se lee como: “Un EDITORIAL puede publicar cero, uno o muchos LIBROS; y un LIBRO debe ser publicado por un solo EDITORIAL”.



O, en el ejemplo siguiente que se muestra, la etiqueta debería decir “Un EDITORIAL puede publicar cero, uno o muchos LIBROS; y un LIBRO puede ser publicado por un solo EDITORIAL”. Este tipo de etiquetado deja la relación perfectamente clara y la establece en términos que un usuario comercial pueda entender.



RELACIONES IDENTIFICATIVAS

En una relación de identificación, la clave principal se propaga a la entidad secundaria como parte de la clave principal. Esto contrasta con una relación no identificativa, en la que la clave externa se convierte en una de las columnas que no son clave. Una relación no identificativa puede aceptar un valor nulo en la columna de clave externa.

Una relación identificativa se establece entre una entidad fuerte y una entidad débil. En este tipo de relación, la entidad débil no puede ser identificada de manera única por sus propios atributos y, por lo tanto, depende de la entidad fuerte para su identificación. Una relación identificativa es aquella en la que la existencia de una entidad depende de otra, y la clave primaria de la entidad dependiente incluye la clave primaria de la entidad a la que está relacionada. En otras palabras, la entidad dependiente no puede existir sin la entidad principal, y su identificación está ligada a ella.

Características:

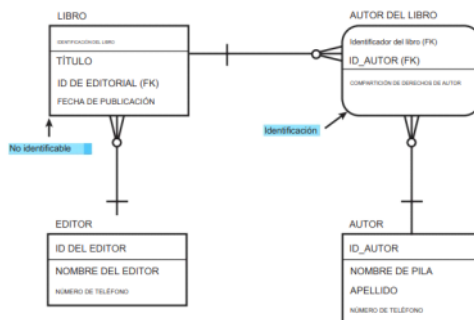
- Dependencia: La entidad dependiente no puede existir sin la entidad principal.
- Clave Primaria: La clave primaria de la entidad dependiente incluye la clave primaria de la entidad principal.
- Ejemplo: En un sistema de gestión de pedidos, una línea de pedido (entidad dependiente) no puede existir sin un pedido (entidad principal). La línea de pedido podría tener una clave primaria compuesta que incluya el ID del pedido y un número de línea.

Relaciones No Identificativas

Una relación no identificativa se da entre dos entidades fuertes. En este caso, cada entidad tiene su propia clave primaria y no depende de otra para su identificación. Por otro lado, una relación no identificativa es aquella en la que una entidad puede existir independientemente de otra. En este caso, la clave primaria de la entidad dependiente no incluye la clave primaria de la entidad a la que está relacionada. Características:

- Independencia: Ambas entidades pueden existir sin depender una de la otra. Por ejemplo, un tipo de flor puede existir sin necesidad de estar asociado a una flor específica. La entidad dependiente puede existir sin la entidad principal.
- Clave Primaria: La clave primaria de la entidad dependiente es independiente y no incluye ninguna parte de la clave primaria de la entidad principal.
- Un cliente (entidad independiente) puede tener múltiples pedidos (entidad dependiente), pero los pedidos pueden existir sin necesidad de estar vinculados a un cliente específico.

Si una representación gráfica del cuadro de una tabla tiene bordes redondeados, significa que la relación es identificativa. En efecto, una de las claves externas se convirtió en la clave principal o parte de la clave principal, si es no significativa tiene bordes redondeados, EJEMPLO:

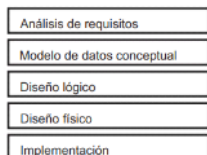


La relación de varios a varios entre las tablas BOOK y AUTHOR se resolvió con la tabla asociativa denominada AUTOR LIBRO. La combinación de las dos claves externas forma la clave principal única. Esta clave principal también garantiza que no habrá entradas duplicadas.

La relación entre las tablas EDITOR y LIBRO no es identificable, como lo indican los bordes afilados. La columna de clave externa ID DEL EDITOR no es parte de la PRIMARY KEY de la tabla BOOK. Las columnas de la clave foránea de una relación no identificable pueden ser NULL o NOT NULL. En este caso, puede determinar si se permite un valor nulo al verificar si la columna de clave externa ID DEL EDITOR es parte de la clave principal de la tabla BOOK.

NOTA: Una clave foránea puede ser NULL si la relación entre las tablas permite que no haya un registro correspondiente en la tabla referenciada. Esto significa que no es obligatorio que cada registro en la tabla que contiene la clave foránea tenga un valor asociado en la tabla referenciada.

CONTEXTO DE DESARROLLO DE BASE DE DATOS



-Análisis de requisitos: El proceso de diseño y desarrollo de bases de datos comienza con la recopilación de los requisitos de datos que identifican las necesidades y deseos de los usuarios. Uno de los resultados de esta fase es una lista de elementos de datos individuales que deben almacenarse en la base de datos.

-Modelo de datos conceptual: El modelo de datos conceptual agrupa de forma lógica los principales elementos de datos del análisis de requisitos en entidades individuales. Una entidad es algo de importancia para lo cual es necesario almacenar datos. Por ejemplo, todos los datos relacionados con los libros, como el título, la fecha de publicación y el precio de venta al público, se colocan en la entidad del libro. Los elementos de datos como el nombre y la dirección del autor forman parte de la entidad del autor. Los elementos de datos individuales se denominan atributos. Usted designa un identificador único, o clave candidata, que distingue de forma única una fila en la entidad. En este modelo de datos conceptual, utilizamos los términos entidad, atributo y clave candidata o identificador único en lugar de tabla, columna y clave principal, respectivamente. Los atributos no críticos no se incluyen en el modelo para enfatizar el significado comercial de esas entidades, atributos y relaciones. Las relaciones de muchos a muchos son aceptables y no se resuelven. El diagrama del modelo conceptual es útil para comunicar la comprensión inicial de los requisitos a los usuarios comerciales. El modelo conceptual no tiene en cuenta la plataforma de implementación o software de base de datos. Muchos proyectos se saltan el modelo conceptual y pasan directamente al modelo lógico. El modelo de datos conceptual es una representación abstracta y de alto nivel de los datos que se gestionarán dentro de un sistema de información.

Proporciona una vista general sin entrar en detalles técnicos o específicos sobre la implementación. Se centra en las entidades y sus relaciones, dejando de lado atributos específicos y claves primarias.

-Modelo de datos lógicos: El objetivo del modelo de datos lógicos es mostrar que todas las entidades, sus respectivos atributos y la relación entre entidades representan los requisitos de negocio, sin considerar cuestiones técnicas. El enfoque se centra por completo en los problemas de negocio y considera un diseño que se adapte al crecimiento y al cambio. Las entidades y los atributos requieren nombres descriptivos y documentación de su significado. El etiquetado y la documentación de las relaciones entre entidades aclaran las reglas de negocio entre ellas. Describe todas las entidades, atributos y relaciones entre ellos. Esto incluye la especificación de claves primarias y foráneas, así como los tipos de datos para cada atributo. En este nivel, se aplica la normalización para eliminar redundancias y asegurar la integridad de los datos. Esto incluye resolver relaciones de muchos a muchos y definir claramente las relaciones entre entidades. El diagrama puede mostrar el tipo de datos de un atributo en términos generales, como texto, número y fecha. En muchos modelos de diseño lógico, se encuentran columnas de clave externa identificadas; en otros, están implícitas. El modelo completo se denomina modelo lógico de datos o diagrama de relación entre entidades (ERD). Al final de la fase de análisis, las entidades están completamente normalizadas, se determina el identificador único para cada entidad y las relaciones de varios a varios se resuelven en entidades asociativas.