

El Modelo Relacional

- ¿Qué es?
- Estructura de Datos relacional
- Restricciones de Integridad

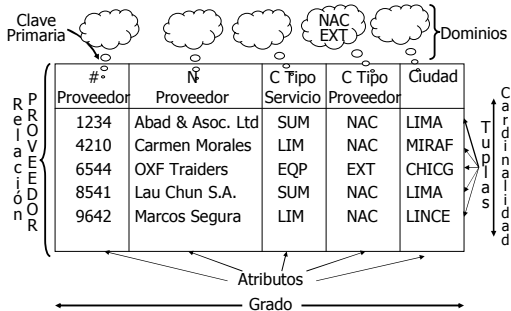
¿Qué es?

- Es el modelo teórico que constituye los cimientos de los manejadores de base de datos relacionales (RDBMS).
- Es el principal modelo de datos empleado para las aplicaciones de procesamiento de datos, cubriendo la mayor proporción de las ventas mundiales de DBMS.

¿Qué es?

- El modelo relacional es una forma de ver los datos que consta de:
 - Una representación de los datos mediante tablas llamadas "relaciones" (**Estructura**)
 - Un conjunto de restricciones que permiten la consistencia de los datos (**Integridad**)
 - Un conjunto de operadores que permiten manipular dichos datos (**Lenguaje de Manipulación**)

Estructura de Datos Relacional



Bases de Datos Relacionales Terminología

| Término Relacional | Término tabular |
|--------------------------|---|
| Base de datos relacional | Conjunto de tablas |
| Relación | Una sola tabla |
| Atributo | Encabezado de columna |
| Tupla | Fila de datos de una tabla |
| Cardinalidad | Número de fila de una tabla |
| Grado | Número de columnas de una tabla |
| Dominio | Lista de valores permitidos para los datos de una columna |

Estructura de Datos Relacional

- El **dominio** es el componente elemental de una BD relacional: un conjunto de valores (= tipo de dato).
 - $D_1 = \{\text{Juan, Maria, Pablo}\}$
 - $D_2 = \text{todas las cadenas de caracteres}$
 - $D_3 = \{x / x \text{ es un entero y } x > 0\}$
 - $D_4 = \{x / x \text{ es un entero negativo}\}$
- Es un conjunto nominado y homogéneo de valores atómicos
- Se especifica lógicamente mediante un nombre y un formato, por extensión (lista de valores), o por intención (tipo de dato o predicado).

Estructura de Datos Relacional

- Estos dominios o tipos pueden ser definidos por el sistema o definidos por el usuario
- Un **dominio o tipo** tiene asociado un conjunto de *operadores* válidos que se pueden aplicar a sus valores, los cuales dependen de su semántica y no de su representación física.
 - Todo valor *tiene* un tipo
 - El sistema siempre comprueba que los operandos son del tipo adecuado para la ejecución de cada operación

Estructura de Datos Relacional: Dominios

Los dominios restringen la manipulación:

- Las comparaciones tienen sentido cuando se plantean entre atributos que provienen del mismo dominio:
 $Q_Peso = Q_Unidades$
 $A.N_Ciudad = B.N_Ciudad$
- Las expresiones tienen sentido si emplean operadores definidos como válidos en los dominios y en sus combinaciones:
 $Q_Peso + Q_Unidades$
 $Q_Peso * Q_Unidades$

Estructura de Datos Relacional: Dominios

El soporte para dominios (tipos) en el modelo relacional implica que:

- El sistema siempre sabrá qué expresiones son válidas y el tipo del resultado de cada expresión
- El conjunto de tipos de una base de datos es un conjunto cerrado: el tipo del resultado de toda expresión será siempre un tipo conocido por el sistema
- El sistema conoce qué asignaciones y qué comparaciones son válidas

Fuente: C.J.Date, "Introducción a los sistemas de bases de datos"

Estructura de Datos Relacional

- Un **Atributo** es un par ordenado (N, D) , donde N es el nombre del atributo y D es el dominio del cual toma sus valores.
- Una **Relación** (tabla) es un conjunto de atributos con nombre (R, C) , donde
 R es el nombre de la relación, y
 $C = \{(N_1, D_1), (N_2, D_2), \dots, (N_m, D_m)\}$ es el conjunto finito de atributos;
 todos los N_i son distintos, aunque no necesariamente lo son los D_i

Estructura de Datos Relacional

- Una **asociación** basada en el atributo (N, D) es un par ordenado (N, x) , donde $x \in D$.
 Luego, una tupla es un **conjunto** de asociaciones (N, x) , una por cada atributo del esquema.
- Un cuerpo relacional es un **conjunto** de tuplas
- Una base de datos relacional es un **conjunto** de relaciones

Estructura de Datos Relacional: Relaciones

Relación CURSO

| CoCurso | NoCurso | Qthoras | NoGrupo | } ESQUEMA C U R S O |
|---------|-------------------------------|---------|---------------------|--|
| 00012 | Diseño de Base de Datos | 50 | Base de Datos | |
| 00034 | Bases de Datos OO | 30 | Base de Datos | |
| 00167 | Sistemas Operativos Avanzados | 30 | Sistemas Operativos | |
| 01521 | Datawarehousing | 25 | Base de Datos | |
| 005142 | Introducción al C++ | 25 | Lenguajes | |

Estructura de Datos Relacional: Relaciones

D_1 el conjunto de los nombres de códigos de curso
 D_2 el conjunto de los nombres de curso
 D_3 el conjunto de los números naturales
 D_4 el conjunto de los grupos

Cada fila de CURSO consiste de una tupla (v_1, v_2, v_3, v_4) ,
 donde: v_i está en el dominio D_i

→ La tabla **CURSO** es un subconjunto de
 $D_1 \times D_2 \times D_3 \times D_4$

Estructura de Datos Relacional: Relaciones

En general:

Una tabla de "n" atributos debe ser un subconjunto de

$$D_1 \times D_2 \times D_3 \dots \times D_{n-1} \times D_n$$

Definición matemática de "Relación":

"Subconjunto del producto cartesiano de la lista de dominios"

→ Relación = Tabla
 Tupla = Fila

Propiedades de las Relaciones

- No existen tuplas repetidas:

Cada relación es un conjunto matemático; por ende, no tiene elementos repetidos.

- Las tuplas no están ordenadas (de arriba a abajo).
- Los atributos no están ordenados (de izquierda a derecha).
- Todos los valores de los atributos son atómicos. Es decir, cada tupla contiene exactamente un valor para cada atributo.

Restricciones de Integridad

Restricciones de Integridad

- El propósito de las reglas de integridad es informar al DBMS de ciertas restricciones en el mundo real.
- La mayor parte de las reglas de integridad de datos son específicas, en cuanto a que se aplican a una base de datos en particular (reglas de negocio).

Restricciones de Integridad (Constraints)

Aseguran que las modificaciones realizadas no originen pérdida de consistencia de los datos

- Restricciones de Dominio:
Especifican el conjunto de valores que se puede asociar a un atributo.
- Restricciones de Clave Primaria
Todas las tuplas de una relación son distintas, es decir, debe haber siempre una clave primaria que garantice esta unicidad

Restricciones de Integridad (Constraints)

■ Restricciones de Clave Primaria

- La *clave primaria* (PK) es un identificador único para una relación.
- La clave primaria puede ser *compuesta* (más de un atributo)
- Es posible que en una relación haya más de un identificador único o "*clave candidata*":
 - Unicidad: En cualquier momento dado, no existen dos tuplas en R con el mismo valor de K.
 - Minimalidad: Si K es compuesta, no será posible eliminar ningún componente de K, sin destruir la propiedad de unicidad.
- En estos casos una debe ser escogida como clave primaria y las otras se denominarán claves alternantes únicas.

Restricciones de Integridad (Constraints)

- Restricciones de Integridad de entidades:
Ningún componente de la clave primaria de una relación base puede aceptar nulos (un valor desconocido o no aplicable)
- Restricciones sobre nulos:
Especifican cuándo un atributo puede no tener un valor asociado por ser desconocido o no aplicable a alguna instancia en particular

Restricciones de Integridad (Constraints)

- Restricciones de Integridad Referencial:
Aseguran que un valor que aparece en una relación para un conjunto de atributos dado, aparezca también para un conjunto de atributos en otra relación:
"La base de datos no debe contener valores de clave foránea sin concordancia"

Restricciones de Integridad (Constraints)

■ Restricciones de Integridad Referencial:

- Una **clave foránea** o ajena es un atributo o conjunto de atributos (clave foránea compuesta) de una relación R2, cuyos valores deben concordar con los de la clave primaria de alguna relación R1 (donde: R1 y R2 no necesariamente son distintos).

Restricciones de Integridad (Constraints)

■ Restricciones de Integridad Referencial:

- El atributo FK (quizá compuesto) de una relación base R2 es una clave foránea si satisface dos propiedades:
 - Cada valor de FK es nulo del todo o bien no nulo del todo.
 - Existe una relación base R1 con clave primaria PK, tal que cada valor no nulo de FK es idéntico al valor de PK en alguna tupla de R1.

Restricciones de Integridad (Constraints)

■ Restricciones de integridad Semántica - Disparadores (triggers):

Son órdenes que el sistema ejecuta de manera automática como efecto secundario de la modificación de la base de datos. Su diseño debe cumplir dos requisitos:

- ✧ Especificar las condiciones de ejecución.
- Especificar las acciones a realizar.

Restricciones de Integridad (Constraints)

Ejemplo:

"Los sobregiros en las cuentas corrientes se tratan dejando en cero el saldo de la cuenta y creando un préstamo por el importe sobregirado"

- Condición:
 - Actualización de la relación *CUENTA* que da un valor negativo en el atributo *saldo*
- Acciones:
 - Insertar una nueva tupla en la relación *PRESTAMO*
 - Cambiar el valor de *saldo* a cero en la relación *CUENTA*

Restricciones de Integridad (Constraints)

- Restricciones de integridad Semántica – Asertos (afirmaciones):

Son expresiones declarativas que expresan predicados que deben ser verdaderos. Es decir, condiciones que siempre se deben cumplir en la BD como un todo, que pueden involucrar varias tuplas de varias tablas.

Ejemplo:

"La suma de los importes de los préstamos de un banco debe ser menor a la suma de los saldos de las cuentas".

Integridad Referencial – Reglas para Claves Foráneas

Para cada clave foránea es necesario responder tres preguntas:

- ☞ ¿La clave foránea puede aceptar nulos?
- ✕ ¿Que deberá suceder si hay un intento de eliminar el registro de referencia de una clave foránea?
- ☞ ¿Que deberá suceder si hay un intento de modificar la clave foránea?

Integridad Referencial – Reglas para Claves Foráneas

Si la relación R2 tiene una clave foránea FK que lo relaciona con R1:

- **Insertar:**

Si se inserta una tupla en R2, debe asegurarse que
 \exists una instancia en R1 con el valor correspondiente como PK

- **Borrar:**

Si se borra una tupla en R1, identificar el conjunto de tuplas en relacionadas mediante una FK. Si las hay, se puede rechazar la operación de borrado, o borrarlas en cascada o alterar su valor.

Integridad Referencial – Reglas para Claves Foráneas

- **Actualizar:**

- Si se modifica una tupla de R2

Se procede a verificar como en el caso de la inserción

- Si se modifica una tupla en R1

Se procede a verificar como en el caso del borrado

Integridad Referencial – Reglas para Claves Foráneas

REGLAS:

- *Cascada*

Si se elimina una instancia del padre, todas las instancias hijo relacionadas también se eliminarán.

- *Restricción*

No se puede eliminar una instancia del padre si tiene hijos asociados, o no se puede eliminar un hijo si existe una instancia relacionada en la entidad padre.

Integridad Referencial – Reglas para Claves Foráneas

■ *Set Null*

Si se elimina una instancia del padre, las claves foráneas (FK) en las instancias hijo relacionadas cambian al valor nulo.

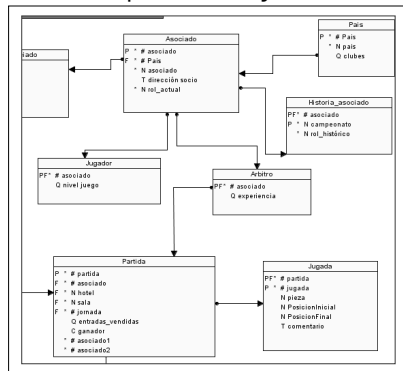
■ *Set Default*

Si se elimina una instancia del padre, las claves foráneas (FK) en las instancias hijo relacionadas cambian al valor por omisión establecido.

■ *<None>*

No se toma ninguna acción específica.

Campeonato de Ajedrez



Integridad Referencial – Reglas para Claves Foráneas

Para cada relación existen seis posibles acciones para las cuales es necesario establecer las reglas de integridad referencial:

| | |
|---------------|--------------|
| Parent Insert | Child Insert |
| Parent Update | Child Update |
| Parent Delete | Child Delete |



RESUMEN

- El modelo relacional establece una estructura para las relaciones (tablas), que se basa en el concepto de dominios
- Existen diferencias marcadas entre la propuesta del modelo relacional y sus implementaciones en los DBMS basados en este modelo
- El modelo relacional también nos ofrece herramientas para asegurar que la información es correcta en cualquier estado de la base de datos
