

Tema 5.- EL MODELO RELACIONAL DE DATOS

- Codd 1970
- Modelo de Datos Lógico -de Representación o Implementación- (basado en registros)
- El modelo más usado en las aplicaciones comerciales de procesamiento de datos convencional
- Dividido en 3 partes:
 1. Estructura de Datos
 2. Integridad de Datos (características generales)
 3. Manipulación de Datos

Modelo Relacional de datos - 1

ESTRUCTURA DE DATOS RELACIONAL

BD = Conjunto de RELACIONES

- Forma sencilla de representar datos: **TABLA**
 - Estructura de datos fundamental del modelo
 - Representa una **entidad genérica**
 - Compuesta de COLUMNAS
 - Cada columna representa un **atributo** de la entidad genérica
 - **Conjunto** de FILAS
 - Cada fila representa una **entidad concreta**
- Definición **CREAR RELACION PELICULA**
(**titulo ...**
 duracion ...
 director ...
 estreno ...
 ...)

Modelo Relacional de datos - 2

ESTRUCTURA DE DATOS RELACIONAL

- **FILA = TUPLA**
 - Representa una *entidad concreta*
 - Es una relación entre elementos de datos
 - » Tabla = Conjunto de relaciones
 - Analogía RELACIÓN (concepto matemático) - TABLA
 - Modelo basado en Teoría matemática
 - » Sólida Base Formal
- **COLUMNA = ATRIBUTO / Propiedad de la entidad**
 - Todo Atributo pertenece a un DOMINIO
 - Dominio:
 - Colección de Valores donde un atributo toma su valor real

Modelo Relacional de datos - 3

E.D.RELACIONAL: términos básicos

- **RELACIÓN = TABLA** (conjunto de tuplas)
- **TUPLA = FILA** $t \in R$ si la tupla t está en la relación R
- **CARDINALIDAD**: nº de tuplas/filas en una relación
- **ATRIBUTO = COLUMNA** (nombres únicos dentro de cada relación)
- **GRADO**: nº atributos/columnas en una relación
- **DOMINIO**: Colección de Valores Permitidos para cierto(s) atributo(s)

Modelo Relacional de datos - 4

ED RELACIONAL Definiciones Formales: RELACIONES

Una RELACIÓN **R**, sobre un conjunto de dominios **D₁, D₂, ..., D_n** se compone de dos partes: *cabecera* y *cuerpo*

1. **Cabecera**: Conjunto fijo de atributos (pares *Atributo: Dominio*)

$$\{ (A_1:D_1), (A_2:D_2) \dots (A_n:D_n) \}$$

Cada A_j corresponde a un único D_j

2. **Cuerpo**: Conjunto de tuplas variable en el tiempo
tupla = conjunto de pares *Atributo: Valor*

$$\{ (A_1:v_{i1}), (A_2:v_{i2}) \dots (A_n:v_{in}) \}, \text{ donde } i=1..m$$

- *Cardinalidad* **m** (cambia con el tiempo)
- *Grado* **n** (no suele variar)

Modelo Relacional de datos - 7

ED RELACIONAL Definiciones Formales: RELACIONES

• Relación vs. Tabla

- Relación: Representación abstracta de un elemento o entidad de datos
- Tabla: Representación concreta de tal elemento abstracto

• Modelo Relacional

- Ventaja
 - Representación muy sencilla (tabla) de su *elemento abstracto* básico (*relación*)
 - Fácil de utilizar, entender, razonar...
- Inconveniente
 - **Aparente orden** entre filas y entre columnas de la tabla

Modelo Relacional de datos - 8

ED RELACIONAL Definiciones Formales: RELACIONES

- Definición de Datos

- Definición de cada Relación dentro de la Definición de la BD

- *Creación --- relación "vacía" (sin tuplas)*

CREAR RELACION PELICULA

(**titulo** **Titulos,**
 duracion **Tiempo,**
 director **Nombres,**
 estreno **Fechas**
 ...
);

- Eliminación de una Relación

- *Dstrucción --- relación, tuplas e información existente en el Catálogo*

DESTRUIR RELACION PELICULA ;

Modelo Relacional de datos - 9

ED RELACIONAL Definiciones Formales: RELACIONES

- Propiedades de una Relación

1. ***No existen tuplas repetidas***

2. ***Las tuplas no están ordenadas***

} Pq *cuerpo* = conjunto
matemático de tuplas

3. ***Los atributos no están ordenados***

Pq *cabecera* = conjunto de pares *Atributo: Dominio*

4. ***Los valores de atributos son Atómicos***

Pq *Dominio* = Conjunto de Valores Atómicos

- Intersección fila/columna = un solo valor (no una lista de valores)
- Si **R** cumple esta propiedad, **R** **está normalizada** (en 1FN)

Modelo Relacional de datos - 10

ED RELACIONAL Definiciones Formales: RELACIONES

- FORMAS NORMALES

R está en determinada **FN** si

cumple un conjunto de condiciones o **restricciones**,
necesarias para estar **bien diseñada**,
de acuerdo con el **modelo relacional** de datos.

- Toda relación ha de estar en 1FN (estructura de datos simple)

Nota: las Formas Normales se estudiarán con detalle en esta asignatura,
en el tema 9: “Diseño en el Modelo Relacional:Normalización”

ED RELACIONAL: BASE DE DATOS RELACIONAL

- Base de Datos Relacional (BDR)
 - BD percibida por el usuario como una *colección de relaciones normalizadas de diversos grados y que varía con el tiempo*
 - Analogía con los conceptos de los sistemas de procesamiento de ficheros:
 - Relación ----- Archivo
 - Tupla ----- Registro concreto
 - sin orden visible entre registros
 - Atributo ----- Campo
 - sin orden visible entre campos
 - valor atómico
 - Principio básico del MR:
Entidades representadas de una ÚNICA manera: por su VALOR
 - Conexión lógica entre Relaciones (**Interrelación**)
representada mediante valores. No existen punteros

ED RELACIONAL: ESQUEMA DE BASE DE DATOS

- En una BD distinguimos
 - ◆ Esquema: Diseño lógico de la BD
 - ◆ Instancia: Visión del contenido de la BD en determinado instante
- Esquema de una Relación
 - Conjunto con nombre de pares Atributo: **Dominio**

PELICULA (titulo:Titulos,duracion:Tiempo,director:Nombres,fecha:Fechas)

 - » Esquema de BD = conjunto de esquemas de relación
 - Esquema de Relación **no suele variar** (costoso)
 - reescritura de miles de tuplas
 - ¿valores de nuevos atributos para tuplas existentes?

Modelo Relacional de datos - 13

ED RELACIONAL: ESQUEMA DE BASE DE DATOS

- Instancia de una Relación
 - Conjunto de tuplas que contiene en un instante dado (instantánea)
 - { ("Volaverunt", ..., "Luna, B.", ...)
 - ("Torrente", ..., "Segura, S.", ...)
 - ("Barrio", ..., "León, F.", ...) } ← instancia de la relación PELICULA
 - » Instancia de BD = conjunto de instancias de sus relaciones
 - Instancia de Relación **variable** en el tiempo
 - **nuevas** tuplas, **modificación** o **borrado** de existentes
- ESQUEMA de relación suele consistir en
 - Estructura Nominativa (cabecera)
 - Reglas de Integridad

Modelo Relacional de datos - 14

Características Generales de INTEGRIDAD de DATOS

- Instancia de BD **refleja la realidad**
es un MODELO de una PORCIÓN del mundo real (UoD)
- Algunas configuraciones de valores NO tienen SENTIDO
pues NO REPRESENTAN ningún estado posible del mundo real
- » Definición de la BD necesita REGLAS DE INTEGRIDAD...
 - ♦ Informan al DBMS de **restricciones del mundo real**
así el DBMS evita configuraciones de datos imposibles
 - ♦ Las **RI son específicas de la BD** considerada, PERO el MR incluye
DOS CARACTERÍSTICAS GENERALES DE INTEGRIDAD
IMPORTANTES y NECESARIAS en toda BD:
 - **Claves Candidatas y Primarias**
 - **Claves Ajenas** (foráneas, externas)

Modelo Relacional de datos - 15

CLAVES CANDIDATAS

- Sea R una relación $R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$
- Una **clave candidata** para R es un subconjunto **CK** de sus atributos tal que:
 1. (Prop. **Unicidad**) No existen dos tuplas distintas con igual valor para CK
 2. (Prop. **Irreductibilidad**) Ningún subconjunto de CK tiene la propiedad de unicidad
- Clave Candidata:
 - Simple: con 1 atributo
 - Compuesta: con más de 1 atributo
- Las claves candidatas deben ser irreductibles, pues cada clave candidata es una **RI** para el sistema
- Mecanismo Básico de Direccionamiento a nivel de tuplas
- **Clave Primaria** (*Primary Key, PK*) y **Claves Alternativas** (AK)

Modelo Relacional de datos - 16

CLAVES AJENAS (*Foreign Keys, FK*)

- Atributo (o conjunto de atributos) **FK** de una relación R2, tal que :
 1. Existe otra relación R1 con clave primaria PK , y
 2. Cada valor de FK en R2 es idéntico al valor de PK en alguna tupla de R1

» atributo (o conjunto de atributos) de una relación que hace referencia a una Clave Primaria de otra relación

Restricción de Integridad Referencial

Todo valor de una FK debe coincidir con un valor en la correspondiente PK

La BD no debe contener FK sin correspondencia:

Si B referencia a A, A debe existir

- Ojo: Puede existir valor de PK al que NO haga referencia ningún valor de la FK

Modelo Relacional de datos - 17

CLAVES AJENAS

- Clave Ajena
 - Simple
 - Compuesta

Cada componente de una FK debe estar **definido sobre el mismo DOMINIO que el correspondiente atributo de la PK** a la que referencia

- Uso de Claves Ajenas facilita:
 - Eliminación de la Redundancia: Integridad entre ficheros
 - Mecanismo del MR para establecer VÍNCULOS ENTRE RELACIONES

Diagrama Referencial

- Expresión de la existencia de Claves Ajenas

Camino Referencial

Ciclo Referencial

- Camino que empieza y acaba en la misma relación
- Caso especial: *Autorreferencia*

Modelo Relacional de datos - 18

CLAVES

- Definición de Datos

La **definición** de cada Relación incluirá...

- ✕ **clave primaria**
- ✕ **claves alternativas** y
- ✕ **claves ajenas**

CREAR RELACION PELICULA

```
(
  titulo    Titulos,
  duracion  Tiempo,
  director  Nombres,
  estreno   Fechas,
  ...
  CLAVE PRIMARIA (titulo),
  CLAVE AJENA(director) REFERENCIA DIRECTOR(nombre)
  ... );
```

CREAR RELACION DIRECTOR

```
(
  nombre    Nombres,
  nacionalidad Nacionalidades,
  ...
  CLAVE PRIMARIA (nombre)
  ... );
```

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL: REGLAS DE CLAVES AJENAS

- Las operaciones que no satisfacen (violan) la Integridad Referencial dejan la BD en un estado incorrecto

¿Cómo evita el Sistema esos Estados Incorrectos?

El Sistema puede...

1. **Rechazar** toda operación que pueda provocar un estado ilegal
2. **Aceptar** (y ejecutar) tales **operaciones**, pero realizar **acciones** que restauren la integridad de los datos

⇔ Diseñador define Reglas o

Políticas de Mantenimiento de la Integridad Referencial
para que la BD SIEMPRE alcance un Estado Final Legal

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL

R2 → R1

POLÍTICAS

1. **BORRADO** DE UNA TUPLA **t** DE **R1** A LA QUE REFERENCIA OTRA(S) DE **R2**

1. (Defecto) **Rechazar** la operación
Sólo permite borrar **t** si ninguna otra tupla hace referencia a **t**
2. **Cascada**. Propagar la eliminación
 - 1º Borrar todas las tuplas de **R2** que referencian a **t**
 - 2º Eliminar **t**

(* 3. *Borrar a nulos -- se verá después **)

2. **MODIFICACIÓN** DE UNA FK DE **R2** A UN VALOR NO EXISTENTE EN LA PK DE **R1**

- (SIEMPRE) **Rechazar** la operación

Modelo Relacional de datos - 21

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL

3. **MODIFICACIÓN** DEL VALOR DE LA CLAVE PRIMARIA DE UNA TUPLA **t** DE **R1** AL CUAL REFERENCIA UNA(S) TUPLA(S) DE **R2**

1. (Defecto) **Rechazar** la operación
Sólo permite modificar la PK de **t** si ninguna otra tupla referencia a **t**
2. **Cascada**. Propagar la modificación
 - Toda tupla de **R2** que referencia a **t** seguirá haciendolo:
cambiar su valor de clave ajena al nuevo de la clave primaria de **t**
 - Modificar el valor de la clave primaria de **t**

(* 3. *Modificar a nulos -- se verá después **)

4. **INSERCIÓN** DE UNA TUPLA **t** EN **R2** CUYO VALOR DE FK NO CORRESPONDE CON NINGÚNO DE LA PK EN **R1**

- (SIEMPRE) **Rechazar** la operación

Modelo Relacional de datos - 22

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL

- Diseñador de la BD indica (al SGBD) la política a seguir en la *Definición de cada Relación*, dentro de la *Definición de la BD*

```
CREAR RELACION PELICULA
(
  titulo      Titulos,
  director    Nombres,
  ...
  CLAVE PRIMARIA (titulo),
  CLAVE AJENA(director) REFERENCIA DIRECTOR(nombre)
                        AL BORRAR RECHAZO
                        AL MODIFICAR CASCADA
);
```

Modelo Relacional de datos - 23

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL

Operaciones Encadenadas

- Sean R1 y R2 relaciones tales que $R2 \rightarrow R1$, con Regla de Borrado en Cascada
- Borrado de una tupla de R1 \Rightarrow borrado de tuplas de R2 que la referencian
- Si existe R3, tal que $R3 \rightarrow R2$ (con Regla Borrado X) entonces $R3 \rightarrow R2 \rightarrow R1$
- Borrado **directo** de tuplas de R2 \equiv Borrado **provocado** por borrado en R1
 - a) Si X = CASCADA, no problema
 - b) Si X = RECHAZO y hay tuplas en R3 que las referencian


La operación completa FALLARÁ

****Comportamiento análogo para las Modificaciones****

**Las operaciones de actualización en una BD son siempre ATÓMICAS:
se realiza "TODO o NADA"**

Modelo Relacional de datos - 24

VALORES NULOS

- En el mundo Real existe INFORMACIÓN PERDIDA o AUSENCIA DE INFORMACIÓN
(cliente sin *número de teléfono*)
 - Necesaria una forma de *representar* estas situaciones *en los sistemas de BD* ⇒ **valor NULO** (NULL)
 - Si una tupla tiene un atributo con valor NULO, significa que el valor real de tal atributo es desconocido
-  **NULO no es un valor** en sí mismo,
sino un indicador o marca de ausencia de información
- * No hay dos NULOS iguales (*num_telefono NULL ≠ edad NULL*)

Modelo Relacional de datos - 25

VALORES NULOS

- En la *definición de cada Relación* de la BD se indica qué atributos pueden o no contener NULO

```
CREAR RELACION ALUMNO
(
  num_exp  Expedientes  NO NULO,
  nombre   Nombres      NO NULO,
  direccion Direcciones ,
  num_telef Telefonos    NULO,
  ...
  CLAVE PRIMARIA (num_exp)
  ...
);
```

- INSERCIÓN de una tupla SIN VALOR para un atributo A ...
 - si A puede contener NULO el sistema le asigna NULO
 - si A NO puede contener NULO el sistema RECHAZA la inserción

Modelo Relacional de datos - 26

VALORES NULOS: Consecuencias en la INTEGRIDAD de Datos

- NULO y Claves Primarias

Restricción de Integridad de Entidad:

Ningún atributo componente de una clave primaria puede contener nulo

CREAR RELACION EMPLEADO

```
( cod_emp  Codigos      NO NULO,
 nombre   Nombres      NO NULO,
 num_telef Telefonos,
 depto    Num_Departamentos,
 jefe     Codigos,
 CLAVE PRIMARIA (cod_emp),
 CLAVE AJENA depto REFERENCIA DEPARTAMENTO(numero),
 CLAVE AJENA jefe  REFERENCIA EMPLEADO(cod_emp)
 );
```

¿Qué pasaría si **cod_emp** pudiera contener NULO?

- NULO y Claves Ajenas

El Modelo Relacional permite NULO como valor de clave ajena

Ejemplo: Empleados no asignados a ningún departamento o empleados sin jefe

Modelo Relacional de datos - 27

VALORES NULOS: Consecuencias en la INTEGRIDAD de Datos

- Hemos de extender la Definición de Clave Ajena

“Sea R2 una relación. FK es una Clave Ajena en R2 si es un subconjunto de sus atributos, tal que:

1. Existe otra relación R1 con clave primaria PK y

2. En todo momento, cada valor de FK en R2

a) es NULO, o

b) es idéntico a un valor de PK en alguna tupla de R1”

Regla de Integridad Referencial

La Base de Datos no debe contener valores no nulos de clave ajena sin correspondencia

Modelo Relacional de datos - 28

VALORES NULOS: Consecuencias en la INTEGRIDAD de Datos

- Hay que extender algunas **POLÍTICAS** de mantto de la Integridad Referencial:

1. BORRADO DE UNA TUPLA t DE **R1** A LA QUE REFERENCIA OTRA(S) DE **R2**

1. (Defecto) **Rechazar** la operación
2. **Cascada**. Propagar la eliminación
3. **Borrar a nulos**. **Ojo: Si la FK de **R2** permite NULO
 - Las tuplas de R2 que referencian a t pasan a contener NULL
 - Eliminación de t

Ejemplo: borrado de un departamento con empleados

3. MODIFICACIÓN DEL VALOR DE LA CLAVE PRIMARIA DE UNA TUPLA t DE **R1** AL CUAL REFERENCIA UNA(S) TUPLA(S) DE **R2**

1. (Defecto) **Rechazar** la operación
2. **Cascada**. Propagar el cambio
3. **Modificar a nulos**. **Ojo: Si la FK de **R2** permite NULO
 - Toda tupla de R2 que referencia a t pasa a contener NULL
 - Modificar el valor de la clave primaria de t

Ejemplo: Cambio del **cod_emp** de un jefe que cambia de puesto de trabajo

Modelo Relacional de datos - 29

RESUMIENDO...

El **DBMS** se encarga de...

- **Comprobar las claves candidatas** (Primaria + Alternativas)
No existen dos tuplas distintas con igual valor para la Clave Candidata
→ *Definición de BD* : indicar los Atributos Componentes de las Claves Candidatas ←
- **Comprobar la Integridad de la Clave Primaria**
Un atributo componente de una clave primaria no puede contener NULO
→ *Definición de BD* : indicar los Atributos Componentes de la Clave Primaria ←
- **Comprobar la Integridad Referencial...**
El valor de la clave ajena en cualquier tupla, o coincide con un valor de clave primaria de alguna tupla en la relación referenciada, o es NULO
→ *Definición de BD* : indicar los Atributos Componentes de las Claves Ajenas ←
- ... y **mantenerla** frente operaciones que puedan violar la **Integridad**
→ *Definición de BD* : indicar Políticas de Mantenimiento de Integridad Referencial ←

Modelo Relacional de datos - 30

MANIPULACIÓN DE DATOS: ÁLGEBRA RELACIONAL

- Colección de **operadores** que toman **relaciones como operandos** y devuelven **relaciones como resultado**.
- Álgebra Relacional definida por *Codd* consiste de **8 operadores**:
 - » **OPERADORES TRADICIONALES SOBRE CONJUNTOS**
 - *unión*
 - *intersección*
 - *diferencia*
 - *producto cartesiano*
 - ** Los operandos son *relaciones*, y NO conjuntos arbitrarios
⇒ Operaciones *adaptadas a relaciones* (tipo especial de conjuntos)
 - » **OPERADORES RELACIONALES ESPECIALES**
 - *restricción*
 - *proyección*
 - *reunión (join)*
 - *división*

Modelo Relacional de datos - 31

ÁLGEBRA RELACIONAL: *CLAUSURA*

- Propiedad de **clausura relacional**
el resultado de cualquier operación relacional es otra relación

⇒ la salida de una operación puede ser entrada (operando) de otra
 - » **Expresiones Anidadas**
Expresiones cuyos operandos son expresiones
(en lugar de nombres de relación)

Modelo Relacional de datos - 32

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

- En matemáticas, $A \cup B = \{ e / e \in A \text{ y/o } e \in B \}$
- Relación = conjunto de tuplas,
⇒ **es posible hacer la UNIÓN de relaciones R y S**
- *Resultado: conjunto de todas las tuplas que están en R y/o en S*
Sin embargo...
PELICULA \cup DIRECTOR es un conjunto, pero NO ES una RELACIÓN
 - Las **relaciones no** pueden contener mezcla de tuplas de distintos tipos
Deben ser homogéneas
 - Ha de mantenerse la **Propiedad de Clausura:**
el **resultado** de la operación **DEBE ser una relación**
 - » **Las relaciones de entrada deben ser de tipos compatibles**

Modelo Relacional de datos - 33

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

*** COMPATIBILIDAD (con la unión) DE TIPO

Relaciones A y B compatibles en tipo si tienen el "mismo" esquema:

$$A(a_1, a_2, \dots, a_n), B(b_1, b_2, \dots, b_n)$$

1) Igual número de atributos:

$$\text{grado}(A) = \text{grado}(B) = n$$

2) Los atributos correspondientes están definidos sobre el mismo dominio:

$$\text{dom}(a_i) = \text{dom}(b_i) \text{ , } i = 1, 2, \dots, n$$

Ejemplo DIRECTOR \cup DIR_FOTOG

- **UNIÓN, INTERSECCIÓN, DIFERENCIA** entre operandos compatibles en tipo
- **EL PRODUCTO CARTESIANO** no necesita compatibilidad de tipo en sus operandos

Modelo Relacional de datos - 34

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

UNIÓN

- $A \cup B$, con A y B compatibles en tipo, es una **relación** tal que:
Cabecera: la de A (o B)
Cuerpo: conjunto de todas las tuplas que están en A, en B o ambas
Las tuplas repetidas se eliminan (por definición)

INTERSECCIÓN

- $A \cap B$, con A y B compatibles en tipo, es una **relación** tal que:
Cabecera: la de A (o B)
Cuerpo: conjunto de todas las tuplas que están a la vez en A y en B
Ejemplo $DIRECTOR \cap DIR_FOTOG$

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

DIFERENCIA

- $A - B$, con A y B compatibles en tipo, es una **relación** tal que:
Cabecera: la de A (o B)
Cuerpo: conjunto de todas las tuplas que están en A pero NO en B
Ejemplo $DIRECTOR - DIR_FOTOG$
Operación con *cierta direccionalidad* (como la resta aritmética)

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

*** SECUENCIAS DE OPERACIONES del Álgebra Relacional

- *Propiedad de Clausura* permite aplicar una operación tras otra

A, B, C compatibles en tipo

- Única expresión: **expresiones anidadas**

$$A \cap (B \cup C)$$

o bien

- **Nombrar Relaciones Intermedias**

$$T \leftarrow B \cup C$$

$$R \leftarrow A \cap T$$

*** RENOMBRAMIENTO DE ATRIBUTOS

- POR DEFECTO: Atributos de Relación Resultado con **nombres de la 1ª Relación**

$$DIR \leftarrow DIRECTOR \cup DIR_FOTOG$$

- Se puede indicar una **lista con nuevos nombres de atributos**:

$$DIR(codDir, nomDir, apeDir, nacDir, fechaNac, pelic) \leftarrow DIRECTOR \cup DIR_FOTOG$$

Modelo Relacional de datos - 37

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

Producto Cartesiano (ampliado)

- En matemáticas, $A \times B = \{ (a,b) / a \in A \text{ y } b \in B \}$
- Relación = conjunto de tuplas,
⇒ **es posible el PRODUCTO CARTESIANO de relaciones R y S**

» *Resultado*: conjunto de pares ordenados de tuplas

PERO ha de CONSERVARSE la *Propiedad de Clausura*:

el **resultado** DEBE ser un **conjunto de tuplas (no de pares de)**

» *Producto Cartesiano* del Álgebra Relacional *ampliado*, pues
cada par ordenado es sustituido por la tupla resultante de la combinación de las dos tuplas "origen"

Modelo Relacional de datos - 38

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

- **Cabecera** de Relación Resultante **A X B** debe estar Bien Formada
(nombres de atributos únicos)

Es la combinación (unión) de cabeceras de A y de B

- *Problema:* **nombres de atributos iguales** en las dos relaciones
 $ACTOR \times AGENCIA \Rightarrow$ "colisión" de nombres en *nombre*

Solución: **renombrar atributos** antes del producto

$AGENCIA_OK(codAge, nomAge, direccion, telefono) \leftarrow AGENCIA$

$RESULTADO \leftarrow ACTOR \times AGENCIA_OK$

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

PRODUCTO CARTESIANO

- **A X B**, con A, B relaciones cualesquiera, es una **relación** tal que:
Cabecera: **combinación (unión) de las de A y B**
Cuerpo: **conjunto de todas las tuplas formadas por las posibles combinaciones de cada tupla de A con cada tupla de B**

Ejemplo $PELICULA \times DIRECTOR$

- Operación SIN demasiada IMPORTANCIA PRÁCTICA
 \Leftarrow **NO se tiene más información a la salida que a la entrada**
pero es **necesaria para definir** la operación **REUNIÓN (JOIN)**

ÁLGEBRA RELACIONAL: OPERACIONES TRADICIONALES

PROPIEDADES A, B, C relaciones compatibles en tipo

- ASOCIATIVA

$$(A \cup B) \cup C \equiv A \cup (B \cup C) \equiv A \cup B \cup C$$

$$(A \cap B) \cap C \equiv A \cap (B \cap C) \equiv A \cap B \cap C$$

$$(A \times B) \times C \equiv A \times (B \times C) \equiv A \times B \times C (*)$$

- CONMUTATIVA

$$A \cup B \equiv B \cup A$$

$$A \cap B \equiv B \cap A$$

$$A \times B \equiv B \times A (*)$$

- La DIFERENCIA **no** cumple ninguna de esas Propiedades

(*) El producto cartesiano "normal" no las cumple, pero SÍ el "ampliado"

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

RESTRICCIÓN

- Obtener un subconjunto de tuplas de una relación para las cuales se **satisface** una condición de selección

$$\sigma_{\langle \text{condición} \rangle} (\langle \text{relación} \rangle)$$

- **Resultado:** conjunto de tuplas con los atributos de $\langle \text{relación} \rangle$
- $\langle \text{condición} \rangle$
 - Expresión booleana, especificada en términos de atributos de $\langle \text{relación} \rangle$
 - Compuesta por una o más **cláusulas**, del tipo:
 - $\langle \text{nomAtrib} \rangle \langle \text{opComp} \rangle \langle \text{cte} \rangle$ o bien $\langle \text{nomAtrib} \rangle \langle \text{opComp} \rangle \langle \text{nomAtrib} \rangle$
 - $\langle \text{opComp} \rangle$ operador de comparación $\in \{=, <, \leq, >, \geq, \neq\}$
 - $\langle \text{cte} \rangle$ valor constante \in dominio del atributo $\langle \text{nomAtrib} \rangle$
 - Cláusulas conectadas mediante operadores booleanos **AND, OR, NOT**

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

* Subconjunto de tuplas de actores representados por la agencia número 2

$$\sigma_{\text{agencia}=2}(\text{ACTOR})$$

* Actores cuyo caché rebasa los 30.000\$

$$\sigma_{\text{cache}>30000}(\text{ACTOR})$$

* Actores representados por la agencia número 2, cuyo cache rebasa los 25.000\$, o bien por la agencia 4 y con caché superior a 35.000\$

$$\sigma_{(\text{agencia}=2 \text{ AND } \text{cache}>25000) \text{ OR } (\text{agencia}=4 \text{ AND } \text{cache}>30000)}(\text{ACTOR})$$

- **Mecanismo de selección del sistema**

- Aplica <condición> a cada *tupla individual* de <relación> sustituyendo cada atributo por su valor en la tupla
- Si <condición> es *TRUE*, la tupla se selecciona (para el resultado)

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

- **Operador Restricción**

- **Unario:**

- Sólo se aplica a UNA relación

Nunca puede seleccionar tuplas de más de una relación

- Se aplica a UNA única tupla a la vez

<condición> **nunca se refiere a más de una tupla**

- **Grado**(Relación Resultado) = **Grado**(Relación Origen)

Tienen los mismos atributos

- **Nº Tuplas**(Relación Resultado) ≤ **Nº Tuplas**(Relación Origen)

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

- Es conmutativa

$$\sigma_{\text{cond1}}(\sigma_{\text{cond2}}(R)) \equiv \sigma_{\text{cond2}}(\sigma_{\text{cond1}}(R))$$

⇒

- Secuencia de Restricciones (selecciones) en **cualquier orden**
- **Combinación** de una **secuencia de restricciones** en una **única restricción** con *condición conjuntiva*:

$$\sigma_{\text{cond1}}(\sigma_{\text{cond2}}(\dots(\sigma_{\text{condn}}(R))\dots)) \equiv \sigma_{\text{cond1 AND cond2 AND...AND condn}}(R)$$

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

PROYECCIÓN

- RESTRICCIÓN σ selecciona algunas **filas** de la tabla y **desecha** otras
- PROYECCIÓN π selecciona ciertas **columnas** y **desecha** las demás
 - Sólo interesan algunos atributos de la relación
 - Se **proyecta** la relación sobre esos atributos

$$\pi_{\langle \text{listAtrib} \rangle}(\langle \text{relación} \rangle)$$

- $\langle \text{listAtrib} \rangle$ lista de nombres de atributos de $\langle \text{relación} \rangle$
- *Resultado*: **conjunto de tuplas** cuyos **atributos** son **sólo** los de $\langle \text{listAtrib} \rangle$ y **en ese orden**

Ejemplo: Obtener el código, nombre y el caché de todos los actores

$$\pi_{\text{codA, nombre, caché}}(\text{ACTOR})$$

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

- Ojo: si $\langle \text{listAtrib} \rangle$ **no** contiene **atributos clave** \Rightarrow tuplas repetidas ¡!
Ejemplo: Obtener la agencia y la nacionalidad de todos los actores

$$\Pi_{\text{agencia, nacionalidad}}(\text{ACTOR})$$

- » **Eliminación implícita de duplicados** \Rightarrow resultado \equiv relación válida

- **Grado**(Relación Resultado) = N° atributos($\langle \text{listAtrib} \rangle$)
- N° **Tuplas**(Relación Resultado) \leq N° Tuplas(Relación Origen)
y es igual (=) si $\langle \text{listAtrib} \rangle$ contiene una clave candidata

- $\Pi_{\text{lista1}} \left(\Pi_{\text{lista2}} (R) \right) = \Pi_{\text{lista1}} (R)$
siempre que $\text{lista1} \subseteq \text{lista2}$

- **NO** es **conmutativa**

$$\Pi_{\text{lista1}} \left(\Pi_{\text{lista2}} (R) \right) \neq \Pi_{\text{lista2}} \left(\Pi_{\text{lista1}} (R) \right)$$

Modelo Relacional de datos - 47

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

REUNIÓN (JOIN)

- **Combina tuplas relacionadas** de dos relaciones **en una sola tupla**
- Permite procesar *vínculos entre relaciones*

Ejemplo: Datos de películas junto con datos de su correspondiente director

- Necesario combinar cada tupla de PELÍCULA tp con la tupla DIRECTOR td , tal que el valor de *codDir* en td coincida con el de *director* en tp
- Se consigue aplicando la operación REUNIÓN a las dos relaciones

$$\text{PEL_DIR} \leftarrow \text{PELICULA} \bowtie_{\text{director=codDir}} \text{DIRECTOR}$$

Ejemplo: Títulos de películas junto con nombre y apellidos de su director

- Se consigue aplicando la operación REUNIÓN a las dos relaciones
- Y proyectando el resultado sobre los atributos requeridos

$$\text{PEL_DIR} \leftarrow \Pi_{\text{titulo,nombre,apellidos}} \left(\text{PELICULA} \bowtie_{\text{director=codDir}} \text{DIRECTOR} \right)$$

Modelo Relacional de datos - 48

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

- Forma General para relaciones $A(a_1, a_2, \dots, a_n)$ y $B(b_1, b_2, \dots, b_m)$:

$$A \bowtie_{\langle \text{condición de reunión} \rangle} B$$

- *Resultado:* **relación R** con $n+m$ atributos $(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m)$

Cabecera: **unión de las cabeceras de A y B**

Cuerpo: **conjunto de tuplas**, tal que existe **una tupla por cada combinación** de tuplas (una de A y otra de B), que **satisface la <condición de reunión>**

» **REUNIÓN vs. PRODUCTO CARTESIANO**

En el Producto Cartesiano aparecen **TODAS las posibles combinaciones**

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

- **<condición de reunión>**

– Expresión **booleana** especificada en términos de **atributos de A y B**

– **Evaluada para cada combinación (par) de tuplas:**

Si la cumplen, forman **una** nueva tupla de la relación resultado

– Es de la forma:

<condición> AND <condición> AND... AND <condición>

donde:

<condición> tiene la forma $a_i \theta b_j$, y

- a_i es un atributo de A; b_j es un atributo de B,
- $\text{Dominio}(a_i) = \text{Dominio}(b_j)$,
- θ (theta) cumple que $\theta \in \{=, <, \leq, >, \geq, \neq\}$

» Una **REUNIÓN** con **condición de reunión general** es una **REUNIÓN THETA**

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

Ejemplo: Nombre de los actores y las agencias que los representan

- Problema: COLISIÓN DE NOMBRES DE ATRIBUTOS

Solución: Previo **renombramiento** de atributos

$AGENC(codAg, nomAg, dirAg, tel) \leftarrow AGENCIA$

$ACT_AGEN \leftarrow \prod_{nombre, nomAg} (ACTOR \bowtie_{agencia=codAg} AGENC)$

- **Ojo: tuplas** cuyos **atributos de reunión** son **nulos**, **NO aparecen** en el **resultado**

Para actores que se auto-representan \Rightarrow NULL en atributo *agencia*
¿qué relación se obtiene como resultado?

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

- **REUNIÓN más común:** la que implica **comparaciones de igualdad**
 - » **EQUIRREUNIÓN** ($\theta \in \{=\}$)
 - **Relación Resultado con 1 o más pares de atributos con valores idénticos para todas las tuplas** (*agencia* y *codAg* en *ACT_AGEN*)
 - Sobra uno de estos atributos \Rightarrow **REUNIÓN NATURAL**

- **REUNIÓN NATURAL (NATURAL JOIN)** **$A * B$**

- Operación de REUNIÓN **más** importante
- Es una EQUIRREUNIÓN + eliminación de atributos superfluos
- **La definición estándar de REUNIÓN NATURAL exige:**
 - Los **atributos de reunión** deben tener **nombre idéntico**
 - **Si no** es así, aplicar antes un **renombramiento** de atributos
- Sólo **conserva UN atributo de reunión**
- Iguala **TODOS** los pares de **atributos** con **igual nombre** en **A** y **B** (ojo: deben tener el **mismo dominio**)

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

Ejemplos

1. Título de todas las películas junto con el título y descripción de su guión

$GUIONES(\text{guion}, \text{titGuion}, \text{descrip}) \leftarrow GUION$

$RESUMEN \leftarrow \prod_{\text{titulo}, \text{titGuion}, \text{descrip}} (PELICULA * GUIONES)$

2. Títulos de películas junto con el nombre y apellidos de su director

$DIREC(\text{director}, \text{nombre}, \text{apellidos}, \text{nacio}, \text{fechNacim}, \text{opPrima}) \leftarrow DIRECTOR$

$PELI_DIREC \leftarrow \prod_{\text{titulo}, \text{nombre}, \text{apellidos}} (PELICULA * DIREC)$

3. Nombre de actores y de las agencias que los representan

$AGENC(\text{agencia}, \text{nomAg}, \text{dirAg}, \text{tel}) \leftarrow AGENCIA$

$ACT_AGEN \leftarrow \prod_{\text{nombre}, \text{nomAg}} (ACTOR * AGENC)$

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

- Si **ninguna combinación de tuplas cumple** <condición de reunión>

Relación Resultado = **Relación VACÍA** (cero tuplas)

– En general, sea **A** con n_A tuplas, **B** con n_B tuplas

$R \leftarrow A \bowtie_{\langle \text{condición de reunión} \rangle} B$ cumple que $0 \leq n_R \leq n_A * n_B$

- Si **NO hay** <condición de reunión> (A y B SIN atributos en común), entonces

– la <condición de reunión> es TRUE para todas las tuplas,

– $\bowtie \equiv X$ (REUNIÓN \equiv PROD. CARTESIANO \equiv REUNIÓN CRUZADA)

ÁLGEBRA RELACIONAL: OPERACIONES ESPECIALES

DIVISIÓN

- Sean las relaciones $A(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)$ y $B(y_1, y_2, \dots, y_n)$
 $A \div B$ es una relación tal que:
Cabecera: $R(x_1, x_2, \dots, x_m)$ **Atributos no comunes**
Cuerpo: **Conjunto de tuplas (X) tal que existe en A una tupla (X,Y) para TODAS las tuplas (Y) de B**
- Poco común. Útil para consultas especiales ocasionales
 Nombres de actores que trabajan en todas las películas dirigidas por Alex de la Iglesia (Santiago Segura, por ejemplo...)
- Para que una tupla t aparezca en el resultado, los valores de t deben aparecer en A en combinación con TODAS las tuplas de B

Modelo Relacional de datos - 55

A.R. OPERACIONES ESPECIALES. DIVISIÓN

A	x	y
	x1	y1
	x2	y1
	x3	y1
	x4	y1
	x1	y2
	x3	y2
	x2	y3
	x3	y3
	x4	y3
	x1	y4
	x2	y4
	x3	y4

B	x
	x1
	x2
	x3

R	y
	y1
	y4

y_1, y_4 aparecen en A en combinación con las 3 tuplas de B , por eso están en el resultado

$$R = A \div B$$

El resto de valores de y en A , no aparecen con todas las tuplas de B y no son seleccionadas: y_2 no aparece con x_2 , e y_3 no aparece con x_1

Modelo Relacional de datos - 56

OTRAS OPERACIONES DEL ÁLGEBRA RELACIONAL

- Algunas consultas comunes NO pueden expresarse con operaciones estándar del Álgebra Relacional
- Ampliación del poder expresivo del AR mediante **operaciones adicionales**
- Incluidas en la mayoría de los Lenguajes de Consulta Relacionales comerciales

FUNCIONES DE AGREGADOS

- Funciones Matemáticas de agregados sobre colecciones de valores de la BD

Ejemplos:

- Valor medio del caché de todos los actores
- Número de películas (almacenadas en la BD)
- Máximo porcentaje de comisión de las distribuidoras de películas
- Mínima recaudación en taquilla
- Cantidad total pagada a los actores de cierta película

Modelo Relacional de datos - 57

OTRAS OPERACIONES: FUNCIONES DE AGREGADOS

- Funciones **aplicadas a un conjunto de tuplas**
 - SUMA
 - PROMEDIO
 - MÁXIMO
 - MÍNIMO
 - CUENTA (número de tuplas en una relación)

AGRUPACIONES DE TUPLAS

- Según valor de algunos atributos
- Puede aplicarse una Función Agregada a cada grupo por separado

Ejemplo: Media del caché de los actores agrupados por agencias

¿Solución?

- Agrupar actores según su agencia representante (valor de atributo *agencia*)
- » Cada grupo incluye tuplas de actores representados por la misma agencia
- Cálculo del caché medio de cada grupo (función PROMEDIO)
- El *Resultado* es una RELACIÓN $R(\text{agencia, caché-medio})$

Modelo Relacional de datos - 58

OTRAS OPERACIONES: FUNCIONES DE AGREGADOS

- **Definición** de la Operación **FUNCIÓN F**

$\langle \text{atributos de agrupación} \rangle \mathbf{F} \langle \text{lista funciones} \rangle (\langle \text{relación} \rangle)$

donde:

• **$\langle \text{atributos de agrupación} \rangle$**

- Lista de nombres de atributos de $\langle \text{relación} \rangle$
- Indican atributos con los que construir los grupos
- Puede estar vacía \Rightarrow la relación es un (único) grupo

• **$\langle \text{lista funciones} \rangle$**

- Lista de pares $\langle \text{función} \rangle \langle \text{atributo} \rangle$
donde $\langle \text{función} \rangle \in \{ \text{SUMA, PROMEDIO, MÁXIMO, MÍNIMO, CUENTA} \}$
y $\langle \text{atributo} \rangle$ es uno de los atributos de $\langle \text{relación} \rangle$

- **Resultado:** relación R, tal que

Cabecera: atributos de $\langle \text{atributos de agrupación} \rangle$ +
un atributo por cada elemento de $\langle \text{lista funciones} \rangle$

Cuerpo: conjunto de tuplas (una por cada grupo)

OTRAS OPERACIONES: FUNCIONES DE AGREGADOS

Ejemplos

1. **Códigos de Películas, número de actores en cada película y su paga media**

$R(\text{codpeli}, \text{numActores}, \text{pagaMedia}) \leftarrow_{\text{film}} \mathbf{F}_{\text{CUENTA actor, PROMEDIO paga}} (\text{ACTUA_EN})$

2. **Códigos de agencias, número de actores en cada agencia y caché medio**

$R(\text{codAg}, \text{numActores}, \text{cacheMedio}) \leftarrow_{\text{agencia}} \mathbf{F}_{\text{CUENTA codA, PROMEDIO cache}} (\text{ACTOR})$

- Si **no** se indican **nombres** para los atributos de la relación **resultado** R,

- Atributo correspondiente a $\langle \text{lista funciones} \rangle$ denominado **función_atributo**
- Atributo-Agrupación denominado como en $\langle \text{atributos de agrupación} \rangle$

Para los ejemplos anteriores...

1. $R(\text{film}, \text{CUENTA_actor}, \text{PROMEDIO_paga})$

2. $R(\text{agencia}, \text{CUENTA_codA}, \text{PROMEDIO_cache})$

OTRAS OPERACIONES: FUNCIONES DE AGREGADOS

- Si **no** se especifican **atributos** de agrupación
 - Toda la RELACIÓN es un ÚNICO GRUPO
 - Las funciones se aplican a TODAS las tuplas
 - La relación RESULTADO tendrá una ÚNICA TUPLA

Ejemplo Número de películas y recaudación media

F CUENTA codP,PROMEDIO taquilla(PELICULA)

- El resultado de aplicar una función agregada SIEMPRE ES una RELACIÓN, no un nº escalar, aunque tenga un único valor

Ejemplo Recaudación máxima obtenida

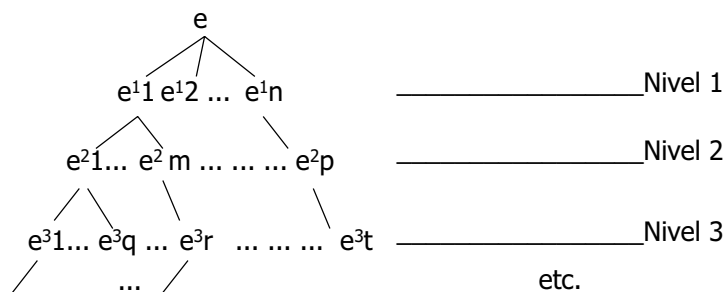
F MÁXIMO taquilla(PELICULA)

OTRAS OPERACIONES DEL ÁLGEBRA RELACIONAL

OPERACIONES DE CIERRE RECURSIVO

- No pueden expresarse en el Álgebra Relacional
- Se aplican a una referencia recursiva entre tuplas del mismo tipo (*empleado* y *jefe* en la relación EMPLEADO)

Ejemplo: Obtener todos los empleados (códigos) cuyo jefe es "e" en todos los niveles



OTRAS OPERACIONES: CIERRE RECURSIVO

- En Álgebra Relacional es **sencillo especificar** empleados cuyo jefe es "e" en cierto **nivel conocido**, pero no en todos los niveles

Ejemplo (nivel 2)

$$\text{EMPL_JEF}(\text{codE}, \text{codJ}) \leftarrow \prod_{\text{codemp}, \text{codjefe}} (\text{EMPLEADO})$$

$$\text{EMPL_1}(\text{cod}) \leftarrow \prod_{\text{codE}} (\sigma_{\text{codJ}=\text{"e"}} (\text{EMPL_JEF})) \longrightarrow \text{Empleados de nivel 1}$$

$$\text{EMPL_2}(\text{cod}) \leftarrow \prod_{\text{codE}} (\text{EMPL_JEFE} \bowtie_{\text{codJ}=\text{cod}} (\text{EMPL_1})) \longrightarrow \text{Empleados de nivel 2}$$

$$\text{RESULTADO} \leftarrow \text{EMPL_1} \cup \text{EMPL_2}$$

OTRAS OPERACIONES DEL ÁLGEBRA RELACIONAL

OPERACIONES DE REUNIÓN EXTERNA

- Extensión de la REUNIÓN
- Permiten conservar TODAS las tuplas en **A** o **B** o en ambas, aunque:
 - **No** tengan **tuplas coincidentes**
 - Contengan **nulos** en los **atributos de reunión**

Ejemplo: Nombres de actores y de sus agencias representantes, si tienen

$$\text{AGEN}(\text{codAg}, \text{nomAg}, \text{direccion}, \text{telefono}) \leftarrow \text{AGENCIA}$$

$$\text{TEMP} \leftarrow (\text{ACTOR} \bowtie_{\text{agencia}=\text{codAg}} \text{AGEN})$$

$$\text{RESULTADO} \leftarrow \prod_{\text{nombre}, \text{nomAg}} (\text{TEMP})$$

OTRAS OPERACIONES DEL A.R. REUNIÓN EXTERNA

REUNIÓN EXTERNA IZQUIERDA $R = A \bowtie B$

- Conserva en R todas las tuplas de A
- Si no encuentra una tupla coincidente en B, cada atributo de R (correspondiente a B) es NULO

REUNIÓN EXTERNA DERECHA $R = A \ltimes B$

- Conserva en R todas las tuplas de B
- Si no encuentra una tupla coincidente en A, cada atributo de R (correspondiente a A) es NULO

Ejemplo: Nombres de agencias y de los actores a los que representan; incluyendo las agencias que no representan a ningún actor

REUNIÓN EXTERNA COMPLETA $R = A \ltimes B$

- Conserva en R todas las tuplas de A y B
- cuando no encuentra tuplas coincidentes, rellenando con NULO

HERENCIA DE CLAVES CANDIDATAS

- Una relación no sólo tiene CABECERA y CUERPO también tiene un conjunto de CLAVES CANDIDATAS (**CK**)
(nota: ya hemos visto la *Herencia de Atributos* desde las relaciones-origen a la resultado)
- El sistema necesita **Reglas de Herencia de Claves Candidatas**, así, dadas las CK de las relaciones-operandos (de una operación relacional), será capaz de **deducir** las **CK de la relación resultado**

es decir...

- La **propiedad de Clausura obliga** al sistema a conocer:
 - **Atributos** y
 - **Claves Candidatas****de cada relación resultado** de una operación del AR

MANIPULACIÓN DE DATOS: CÁLCULO RELACIONAL

- Lenguaje formal para BD Relacionales
- Basado en Cálculo de Predicados de 1er Orden (rama de Lógica Matemática)

CÁLCULO RELACIONAL

vs

ÁLGEBRA RELACIONAL

- | | |
|--|---|
| <ul style="list-style-type: none">- Expresiones Declarativas (lenguaje no procedimental)» No se indica CÓMO evaluar la consulta, sino QUÉ se desea obtener» Describe la información deseada sin dar un procedimiento específico para obtenerla | <ul style="list-style-type: none">- Secuencias de Operaciones» Aunque se anidan para formar una sola expresión, SIEMPRE se indica explícitamente cierto ORDEN de las operaciones» Estrategia parcial de evaluación de la consulta |
|--|---|

Modelo Relacional de datos - 67

CÁLCULO RELACIONAL

- **Poder Expresivo IDÉNTICO del Álgebra y Cálculo Relacionales:**
Cualquier obtención de datos especificada en Álgebra Relacional puede expresarse en Cálculo Relacional (restringido a expresiones seguras) y viceversa
- Definición: **Lenguaje Relacionalmente Completo**
Lenguaje en el que es posible expresar cualquier consulta que pueda especificarse en el Cálculo Relacional
- DOS formas de *adaptar* el Cálculo de Predicados de Primer Orden, para crear un **Lenguaje de Consultas para BDR:**
 - ✓ Cálculo Relacional de **Tuplas** (CRT) ←
 - ✓ Cálculo Relacional de **Dominios** (o Cálculo de Dominios)

Modelo Relacional de datos - 68

CÁLCULO RELACIONAL DE TUPLAS

VARIABLES DE TUPLA

- CRT basado en la especificación de **variables de tupla**
- **Cada variable de tupla *abarca* una relación**
 - ⇒ puede tomar como valor cualquier tupla de esa relación
- Consulta sencilla: $\{ t \mid COND(t) \}$
 - donde: t : variable de tupla
 - $COND(t)$: expresión condicional en la que interviene t
- Resultado: conjunto de tuplas t que satisfacen la condición $COND(t)$

Ejemplo: Actores cuyo caché rebasa los 50.000\$

$\{ t \mid ACTOR(t) \text{ and } t.cache > 50000 \}$

$ACTOR(t)$ indica que $ACTOR$ es la Relación de Intervalo abarcada por t

$t.cache$ hace referencia al atributo caché de la variable de tupla t

Modelo Relacional de datos - 69

CÁLCULO RELACIONAL DE TUPLAS

- Para obtener **algunos atributos** de las tuplas seleccionadas:

$\{ \langle \text{lista atributos} \rangle \mid COND(t) \}$

Ejemplo: Nombre y nacionalidad de los actores cuyo caché rebasa los 50.000\$

$\{ t.nombre, t.nacionalidad \mid ACTOR(t) \text{ and } t.cache > 50000 \}$

* Pto. vista informal: Una **Expresión del Cálculo Relacional** contiene:

- Para cada variable de tupla t , la **relación de intervalo** $R: R(t)$
- **Condición de Selección** de combinaciones de tuplas
 - ✓ Conforme las variables de tupla abarcan sus relaciones, la **condición se evalúa para cada combinación de tuplas**
 - ✓ Las **combinaciones** que dan **TRUE se seleccionan** para el resultado
- Lista de **atributos solicitados**
 - ✓ Se obtienen sus **valores para cada combinación seleccionada**

Modelo Relacional de datos - 70

CÁLCULO RELACIONAL DE TUPLAS

Ejemplo: Fecha de nacimiento y nombre real del actor "Javier Bardem"
{ t.fechaNacim, t.nombreReal | ACTOR(t) and t.nombre = "Javier Bardem" }

ESPECIFICACIÓN FORMAL DEL CÁLCULO RELACIONAL DE TUPLAS

- **Expresión General**

$\{ t_1.a_1, t_2.a_2, \dots, t_n.a_p \mid \text{COND}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m}) \}$

donde:

- $t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m}$ son variables de tupla
- a_i es un atributo de la relación que abarca t_j
- $\text{COND}()$: fórmula (*bien formada*) del Cálculo Relacional de Tuplas
Formada por **átomos** del Cálculo de Predicados

CÁLCULO RELACIONAL DE TUPLAS

ÁTOMOS

- $R(t_i)$ el intervalo de la variable de tupla t_i es la relación R
- $t_i.a \text{ op } t_j.b$,, $op \in \{ =, \neq, <, \leq, >, \geq \}$
 t_i y t_j variables de tupla
 a atributo de la relación que t_i abarca
 b atributo de la relación que t_j abarca
- $t_i.a \text{ op } c$,, $c \text{ op } t_j.b$,, $op \in \{ =, \neq, <, \leq, >, \geq \}$
 t_i y t_j variables de tupla
 a atributo de la relación que t_i abarca
 b atributo de la relación que t_j abarca
 c valor constante

✓ Los átomos están ligados mediante operadores Booleanos **and**, **or**, **not**

CÁLCULO RELACIONAL DE TUPLAS

- **Valor Lógico de un átomo**

Evaluación del átomo para una combinación específica de tuplas \Rightarrow TRUE o FALSE

- $R(t_i)$ es TRUE si se asigna una tupla de R a t_i . Si no, es FALSE
- $t_i.a \text{ op } t_j.b$,, $t_i.a \text{ op } c$,, $c \text{ op } t_j.b$ será TRUE si t_i y t_j se asignan a tuplas tales que los atributos especificados (a y b) satisfacen la condición. Si no es así, será FALSE

DEFINICIÓN (recursiva) **DE FÓRMULA** (o condición) **BIEN FORMADA**

D1. Todo **átomo** es una fórmula bien formada (fbf)

$R(t_i)$,, $t_i.a \text{ op } t_j.b$,, $t_i.a \text{ op } c$,, $c \text{ op } t_j.b$

D2. Si **F1** y **F2** son fbf, también lo son

(F1 and F2), **(F1 or F2)**, **not(F1)** y **not(F2)**

y sus valores lógicos se derivan de **F1** y **F2**, según la Lógica Booleana

Modelo Relacional de datos - 73

CÁLCULO RELACIONAL DE TUPLAS

CUANTIFICADORES

- Universal \forall
- Existencial \exists

VARIABLE DE TUPLA LIBRE Y LIGADA EN UNA FBF

-Def. informal-

t **ligada** si está **cuantificada** (aparece en cláusulas $(\forall t)$ o $(\exists t)$)
si **no**, está **libre**

-Def. formal-

- Si $F \equiv$ **átomo**, **cualquier ocurrencia** de una variable de tupla t , está **libre**
- En **(F1 and F2)**, **(F1 or F2)**, **not(F1)** y **not(F2)**, una **ocurrencia de t** está **libre o ligada según** lo esté en **F1 o F2**
- Toda **ocurrencia libre de t** en **F** está **ligada en F'**, si $F' = (\exists t)F$ o $F' = (\forall t)F$ y t estará **ligada al cuantificador** especificado en **F'**

Modelo Relacional de datos - 74

CÁLCULO RELACIONAL DE TUPLAS

Ejemplo

F1 : d.nombre = "Carmelo Gómez"

F2 : $(\exists t)$ (d.agencia = t.codAg)

La variable de tupla d está *libre* en F1 y en F2

t está *ligada* al cuantificador \exists en F2

(cont.) DEFINICIÓN DE **FÓRMULA BIEN FORMADA**

D3. Si F es una fbf, también lo es $(\exists t)F$, donde t es una variable de tupla.

$(\exists t)F$ es TRUE si F es TRUE para **al menos una tupla asignada a ocurrencias libres de t en F**, de lo contrario es FALSE

D4. Si F es una fbf, también lo es $(\forall t)F$, donde t es una variable de tupla.

$(\forall t)F$ es TRUE si F es TRUE **para toda tupla (en el universo) asignada a ocurrencias libres de t en F**, de lo contrario es FALSE

Modelo Relacional de datos - 75

CÁLCULO RELACIONAL DE TUPLAS

EXPRESIÓN SEGURA

Su resultado es un número finito de tuplas

$E = \{t \mid \text{not}(\text{ACTOR}(t))\} \rightarrow$ TODAS las tuplas del universo que **NO** son de ACTOR $\ddot{\text{!}} \infty \text{!!}$

\rightarrow cuidado al usar cuantificadores (\exists, \forall) o negación de predicados (**not**): la expresión ha de tener sentido (e.d. ser segura y no generar una relación infinita) \leftarrow

DOMINIO DE UNA EXPRESIÓN del CÁLCULO RELACIONAL DE TUPLAS

Todos los **valores constantes** en la expresión o que existen **en cq tupla de las relaciones** a las que se referencia **en la expresión**

Dominio de $E = \{t \mid \text{not}(\text{ACTOR}(t))\} =$ todos los valores de atributos de tuplas ACTOR

» **Expresión Segura** si todo valor del resultado \in dominio de expresión

E es insegura, ya que el resultado incluye tuplas (y, por tanto, valores) que NO están en la relación ACTOR (\notin su dominio)

Modelo Relacional de datos - 76