



BASES DE DATOS

Tema 6

Lenguajes Relacionales

UCLM- E.S. de Informática

Coral Calero, Marcela Genero, Francisco Ruiz



Objetivos

- Conocer los fundamentos de la **dinámica** del **Modelo Relacional**, es decir, los aspectos fundamentales de los lenguajes relacionales formales:
 - **álgebra** y
 - **cálculo relacional**.
- Conocer los principales lenguajes comerciales de los SGBD relacionales:
 - **SQL**
 - **QBE**



Contenido

- **Tipos de lenguajes relacionales**
- **Álgebra relacional**
 - Expresiones
 - Operadores
- **Cálculo relacional**
 - Cálculo de tuplas
 - Cálculo de dominios
- **SQL**
- **QBE**



Bibliografía

- **Básica**
 - Piattini et al. (2006)
 - Caps. 7, 8
- **Complementaria**
 - Atzeni et al. (1999)
 - CapS. 3, 4
 - Elmasri y Navathe (2003)
 - 6, 8



Tipos de Lenguajes Relacionales

- El modelo relacional (MR), como todo modelo de datos (MD), lleva asociado a su parte estática (estructura y restricciones) una **dinámica** que permite la **transformación entre estados** de la base de datos.
- Esta transformación de un estado origen a un estado objetivo se realiza aplicando un **conjunto de operadores**, mediante los cuales se llevan a cabo los siguientes tipos de operaciones:
 - **inserción** de tuplas
 - **borrado** de tuplas
 - **modificación** de tuplas
 - **consultas**
- Si O es un operador, el paso de un estado origen de la base de datos (BD_i) a un estado objetivo (BD_j) se pueden expresar como:

$$O(BD_i) = BD_j$$

ambos estados deben satisfacer las **restricciones de integridad estáticas**, y la transformación ha de cumplir las **restricciones de integridad dinámicas** (entre estados).



Tipos de Lenguajes Relacionales

- Los lenguajes relacionales (LR) **operan sobre conjuntos de tuplas**, es decir, no son lenguajes navegacionales (que manipulan registros, como Pascal, Basic, Cobol, XBase, ...) sino de **especificación**.
- Se dividen en dos tipos:
 - **Algebraicos**: los cambios de estado se especifican mediante **operaciones**, cuyos operandos son relaciones y cuyo resultado es otra relación. Genéricamente se conocen como **álgebra relacional**.
 - **Predicativos**: los cambios de estado se especifican mediante **predicados** que definen el estado objetivo sin indicar las operaciones que hay que realizar para llegar al mismo. Genéricamente se conocen como **cálculo relacional**. Se dividen en dos subtipos:
 - orientados a **tuplas**.
 - orientados a **dominios**.
- Los **lenguajes comerciales** (SQL, QBE, etc.) están basados en los anteriores pero con sintaxis más amigable.



Álgebra Relacional

Expresiones

- Una expresión en AR está formada por una **colección de operadores** que, aplicados a relaciones, dan como resultado nuevas relaciones:

Sean R_1, R_2, \dots, R_i y R' relaciones cualesquiera y O un operador del AR, una operación consiste en aplicar O a la(s) relación(es) R_1, \dots, R_i , obteniéndose R' :

$$O (R_1 \dots R_i) = R'$$

Al ser el resultado de la operación otra relación, se cumple la **propiedad de cierre**, es decir, si O_1, \dots, O_n representan operadores, se cumple:

$$O_n (\dots (O_1 (R))) = R'$$

- Una expresión del AR se construye a partir de **subexpresiones**.

Siendo R_1 y R_2 relaciones, O un operador unario o binario, son expresiones válidas en el AR las siguientes:

- Una relación de la BD: R_i
- Una relación constante C .
- $R_1 \ O \ R_2$, $O(R_1)$.
- Las expresiones derivadas por aplicación repetida de las tres opciones anteriores.



Álgebra Relacional

Operadores

- Podemos **clasificar** los **operadores** del AR de tres formas:

a) Según su **origen**:

- Procedentes de la **teoría de conjuntos**: unión, intersección, diferencia y producto cartesiano.
- Relacionales especiales**: restricción, proyección, combinación y división.

b) Según la **completitud del lenguaje**:

- Primitivos**: operadores esenciales que no pueden obtenerse de otros (sin ellos, el AR no sería un lenguaje completo).
- Derivados**: se pueden obtener aplicando varios de los operadores primitivos. Aunque se puede prescindir de ellos, son útiles para simplificar muchas operaciones habituales.

c) Según el **número de operandos**:

- Unarios**: actúan sobre una única relación.
- Binarios**: el operador tiene dos relaciones como operandos.



Álgebra Relacional

Operadores - asignación

- **Asignación:**

- Operación auxiliar que se utiliza para almacenar el resultado de una consulta en una nueva relación o para denominar **resultados intermedios** cuando se desea dividir una operación compleja en una secuencia de operaciones más simples. El símbolo para representarla es una flecha:

$$\text{RELACION_NUEVA} \leftarrow \text{O(R)}$$

- También permite **copiar** una relación en otra nueva (siendo el operador O la identidad):

$$\text{R}' \leftarrow \text{R}$$

- O **renombrar los atributos** de la relación original:

$$\text{RELACION_NUEVA (A}_1, \text{A}_2, \dots, \text{A}_n) \leftarrow \text{O(R)}$$



Álgebra Relacional

Operadores – primitivos vs derivados

- **Operadores primitivos:**

- Selección (restricción)
- Proyección
- Unión
- Diferencia
- Producto Cartesiano

- **Operadores derivados:**

- Combinación
- Combinación natural
- Intersección
- División ...



Álgebra Relacional Operadores – selección

- La **selección** o **restricción** de una relación mediante una expresión lógica (**predicado de selección**) da como resultado una relación formada por el conjunto de tuplas que satisfacen dicha expresión.

Autor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan

$\sigma_{\text{nacionalidad="Española"}}$ (Autor)

Nombre	Nacionalidad	Institución
Saltor	Española	U.P.C.

En SQL

```
SELECT *  
FROM autor  
WHERE nacionalidad="Española"
```



Álgebra Relacional Operadores – proyección

- La **proyección** de una relación sobre un conjunto de sus atributos es otra relación definida sobre estos atributos, **eliminando las tuplas duplicadas** que hubieran podido resultar.

Autor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan

$\pi_{\text{nacionalidad}}$ (Autor)

Nacionalidad
Norteamericana
Española
Italiana

En SQL

```
SELECT nacionalidad  
FROM autor
```



Álgebra Relacional Operadores – unión

- La **unión** de dos relaciones r_1 y r_2 , con **esquemas compatibles** R_1 y R_2 , es otra relación definida sobre el mismo esquema de relación, cuya extensión estará constituida por el conjunto de tuplas que pertenezcan a r_1 , a r_2 o a ambas (**sin duplicar**).

Autor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan

Editor

Nombre	Nacionalidad	Institución
Chen	Norteamericana	ER Institute
Yao	Norteamericana	U.N.Y
Bertino	Italiana	U. Milan

Autor \cup Editor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan
Chen	Norteamericana	ER Institute
Yao	Norteamericana	U.N.Y



Álgebra Relacional Operadores – compatibilidad de esquemas

- Se dice que dos esquemas de relación $R_1(A_1i:D_1i)$ y $R_2(A_2i:D_2i)$, con cardinalidades m_1 y m_2 , son compatibles cuando ambos están definidos sobre el mismo conjunto de dominios, cumpliéndose:
 - $\forall A_1i \exists A_2j \mid \text{dom}(A_1i) = \text{dom}(A_2j)$ y
 - $\forall A_2j \exists A_1i \mid \text{dom}(A_2j) = \text{dom}(A_1i)$
 - es decir, cada atributo de R_1 tiene su equivalente de R_2 (con igual dominio) y viceversa; por tanto, el grado (número de atributos) de ambas relaciones debe ser el mismo.
- También se dice que R_1 y R_2 son **semánticamente equivalentes**, lo que no quiere decir que los nombres de los atributos sean los mismos (esto sería sólo un aspecto sintáctico).



Álgebra Relacional Operadores – diferencia

- La **diferencia** de dos relaciones r_1 y r_2 , **compatibles en su esquema**, es otra relación definida sobre el mismo esquema de relación, cuya extensión estará constituida por el conjunto de tuplas que pertenecen a r_1 y no pertenecen a r_2 .

Autor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan

Editor

Nombre	Nacionalidad	Institución
Chen	Norteamericana	ER Institute
Yao	Norteamericana	U.N.Y
Bertino	Italiana	U. Milan

Autor - Editor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.



Álgebra Relacional Operadores – producto cartesiano

- El **producto cartesiano** de dos relaciones r_1 y r_2 , de cardinalidades m_1 y m_2 respectivamente, es una relación definida sobre la unión de los atributos de ambas relaciones y cuya extensión esta constituida por las $m_1 \times m_2$ tuplas formadas **concatenando** cada tupla de r_1 con cada una de las tuplas de r_2 .

LIBRO

Código	Título	Idioma	Nombre_e
001	Bases de Datos	Español	Ra-ma
003	Diseño de BD	Español	Ra-ma

EDITORIAL

Nombre_e	Dirección	Ciudad	País
Ra-ma	Pez, 20	Madrid	España
Addison-Wesley	24 Lennon	London	UK

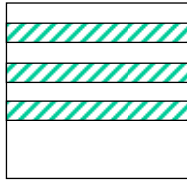
LIBRO x EDITORIAL

Código	Título	Idioma	Nombre_e	Nombre_e	Dirección	Ciudad	País
001	BD	Español	Ra-ma	Ra-ma	Pez, 20	Madrid	España
001	BD	Español	Ra-ma	Addison-Wesley	24 Lennon	London	UK
003	Diseño de BD	Español	Ra-ma	Ra-ma	Pez, 20	Madrid	España
003	Diseño de BD	Español	Ra-ma	Addison-Wesley	24 Lennon	London	UK

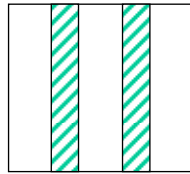


Operadores – resumen operadores primitivos

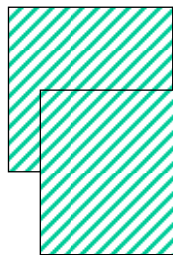
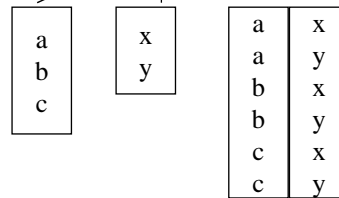
Selección (σ)



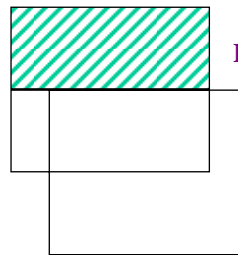
Proyección (π)



Producto (\times)



Unión (\cup)



Diferencia ($-$)



Operadores – combinación

- La **combinación θ** de dos relaciones respecto a una cierta **condición de combinación**, es otra relación constituida por todos los pares de tuplas t_i y t_j concatenadas, tales que, en cada par, las correspondientes tuplas satisfacen la condición especificada.
- También se llama **JOIN** o **PRODUCTO THETA**.
- También se representa mediante el símbolo \bowtie
- Se corresponde con un **producto cartesiano seguido de una restricción**.

$$R_1 \theta_p R_2 \equiv \sigma_p(R_1 \times R_2)$$

- Si se trata de una condición de combinación simple por igualdad también se denomina **equi-combinación**.
- La **condición de combinación**, en el caso más sencillo, está referida a dos atributos A_{1i} y A_{2j} , cada uno de los cuales pertenece a una de las relaciones, unidos por un operador de comparación. En el caso más general, varios pares de atributos pueden combinarse mediante el operador lógico AND.



Operadores – combinación natural

- La **combinación natural (*)** es una combinación por igualdad donde se ha eliminado, en la relación resultante, un atributo de cada pareja cuyos valores son idénticos. Es el caso más utilizado de combinación para relaciones que tienen atributos comunes.
- También se llama **PRODUCTO NATURAL** o **NATURAL JOIN**.
- También se representa mediante el símbolo \bowtie
- Se corresponde con un **producto cartesiano** seguido de una **restricción** por igualdad, y después de una **proyección** (para quitar la duplicación en los atributos emparejados).

$$R_1 * R_2 \equiv \Pi_{(A_1 \cup A_2)} \sigma_p (R_1 \times R_2)$$

- siendo p la condición de igualdad entre todas las parejas de atributos correspondientes: P = (A1i1=A2j1) and (A1i2=A2j2)
 - y A1∪A2 el conjunto de todos los atributos eliminando uno de cada pareja
- Cuando el atributo común tiene el mismo nombre en ambas relaciones, se suele omitir la condición de combinación.



Operadores – combinación natural

AUTOR

NOMBRE	NACIONALIDAD	INSTITUCION
Date, C.J.	Norteamericana	Relational Inst.
Saltor, F.	Española	U.P.C.
Ceri, S.	Italiana	Politéc. Milán

LIBRO

LIBRO	AUTOR	EDITORIAL
DB Systems	Date, C.J.	Addison
Basi di Dati	Ceri, S.	Clup
SQL Standard	Date, C.J.	Addison

AUTOR * LIBRO

(AUTOR.nombre = LIBRO.autor)

NOMBRE	NACIONALIDAD	INSTITUCION	LIBRO	EDITORIAL
Date, C.J.	Norteamericana	Relational Inst.	DB Systems	Addison
Date, C.J.	Norteamericana	Relational Inst.	SQL Standard	Addison
Ceri, S.	Italiana	Politéc. Milán	Basi di Dati	Clup



Álgebra Relacional

Operadores – intersección

- La **intersección** (\cap) de dos relaciones r_1 y r_2 , **compatibles en su esquema**, es otra relación definida sobre el mismo esquema de relación, cuya extensión esta formada por el conjunto de tuplas que pertenecen a r_1 y a r_2 a la vez.
- La intersección se puede definir **en función de la diferencia**:

$$R_1 \cap R_2 = R_1 - (R_1 - R_2)$$

AUTOR

NOMBRE	NACIONALIDAD	INSTITUCION
Date, C.J.	Norteamericana	Relational Inst.
Saltor, F.	Española	U.P.C.
Ceri, S.	Italiana	Politéc. Milán

EDITOR

NOMBRE	NACIONALIDAD	INSTITUCION
Chen, P.	Norteamericana	ER Institute
Yao, L.	Norteamericana	U.N.Y.
Ceri, S.	Italiana	Politéc. milán

AUTOR \cap EDITOR

NOMBRE	NACIONALIDAD	INSTITUCION
Ceri, S.	Italiana	Politéc. Milán

UCLM-ESI-BDa

6.21



Álgebra Relacional

Operadores – división

- La **división** ($:$) de una relación R_1 (dividendo) por otra relación R_2 (divisor) es una relación R (cociente) tal que, al realizarse su combinación con el divisor, todas las tuplas resultantes se encuentran en el dividendo.
- Se puede expresar en función de la **proyección**, el **producto cartesiano** y la **diferencia**:

$$R_1 : R_2 = \Pi_C(R_1) - \Pi_C(R_2 \times \Pi_C(R_1) - R_1)$$

AUTOR_EDITORIAL

NOMBRE	NACIONALIDAD	EDITORIAL
Date, C.J.	Norteamericana	Addison
Cervera, J.	Española	Rama
Saltor, F.	Española	Paraninfo
Ceri, S.	Italiana	Clup
Costilla, C.	Española	Diaz de Santos
Codd, E.	Norteamericana	Prentice Hall
Cervera, J.	Española	Addison

EDITORIAL

EDITORIAL
Addison
Rama

NOMBRE	NACIONALIDAD
Cervera, J.	Española

AUTOR_EDITORIAL : EDITORIAL

UCLM-ESI-BDa

6.22



Álgebra Relacional Operadores – agrupación

- La **agrupación (group by)** permite agrupar tuplas en subconjuntos que poseen valores comunes de ciertos atributos. A dichos subconjuntos se les pueden aplicar **funciones de agregación** (media, frecuencia, varianza, etc.)

AUTOR_ARTIC

NOMBRE	NACIONALIDAD	NUM. ART.
Date, C.J.	Norteamericana	27
Cervera, J.	Española	10
Saltor, F.	Española	14
Ceri, S.	Italiana	9
Costilla, C.	Española	8
Codd, E.	Norteamericana	29

AUTOR_ARTIC AGRUPACION POR nacionalidad, MEDIA (NUM_ART)

NACIONALIDAD	MEDIA (Nº ART.)
Norteamericana	28
Española	11
Italiana	9

UCLM-ESI-BDa

6.23



Álgebra Relacional Operadores – cierre transitivo

- Operación unaria definida sobre **dos atributos compatibles** (que comparten el mismo dominio), que se obtiene por **sucesivas operaciones de combinación, proyección y unión**, consistente en añadir a la relación de origen todas las tuplas que se deducen, sucesivamente, por transitividad hasta la saturación:

si existen las tuplas (a,b) y (b,c) se añade (a,c)

TEMAS

TEMA P	TEMA S
Bases de Datos	Fundamentos
Bases de Datos	Diseño
Diseño	Diseño conceptual
Diseño	Diseño lógico
Diseño lógico	Diseño relacional

Cierre transitivo de Temas

TEMA P	TEMA S
Bases de Datos	Fundamentos
Bases de Datos	Diseño
Diseño	Diseño conceptual
Diseño	Diseño lógico
Diseño lógico	Diseño relacional
Bases de Datos	Diseño conceptual
Bases de Datos	Diseño lógico
Diseño	Diseño relacional
Bases de Datos	Diseño relacional

UCLM-ESI-BDa

6.24



Álgebra Relacional Operadores – combinación externa

- También llamado **OUTER JOIN**.
- Al combinar R1 con R2, evita que las tuplas de una relación que no casan con ninguna tupla de la otra desaparezcan en el resultado (tal como ocurre en la combinación normal o interna).
- Según que las tuplas a conservar sean las de R1, o las de R2, o ambas se tienen tres tipos:
 - por la **izquierda**: /*
 - por la **derecha**: */
 - simétrica**: /*/

AUTOR

NOMBRE	NACIONALIDAD	INSTITUCION
Date, C.J.	Norteamericana	Relational Inst.
Saltor, F.	Española	U.P.C.
Ceri, S.	Italiana	Politéc. Milán

LIBRO

LIBRO	AUTOR	EDITORIAL
DB Systems	Date, C.J.	Addison
Basi di Dati	Ceri, S.	Clup
SQL Standard	Date, C.J.	Addison

Autor /* Libro
(AUTOR.nombre=LIBRO.autor)

Autor_Libro

NOMBRE	NACIONALIDAD	INSTITUCION	LIBRO	EDITORIAL
Date, C.J.	Norteamericana	Relational Inst.	DB Systems	Addison
Date, C.J.	Norteamericana	Relational Inst.	SQL Standard	Addison
Saltor, F.	Española	U.P.C.	Nulo	Nulo
Ceri, S.	Italia	Politéc. Milán	Basi di Dati	Clup

UCLM-ESI-BDa

6.25



Álgebra Relacional Operadores – versiones trivaluadas

- También llamados **Operadores MAYBE**.
- Son un grupo de operadores que tienen en cuenta la **lógica trivaluada**, incorporando en el resultado las tuplas que pueden ser ciertas.
- Existe una versión MAYBE para cada operador de los ya estudiados donde puede producirse una comparación en la que intervienen valores nulos.

R1

a	b
1	5
1	Nulo
2	4
Nulo	2
3	Nulo

R2

c	d
5	3
2	4
Nulo	6
Nulo	5

Combinación MAYBE => **MAYBE R1 $\theta_{b=c}$ R2**

a	b	c	d
1	5	5	3
1	5	Nulo	6
1	5	Nulo	5
1	Nulo	5	3
1	Nulo	2	4
1	Nulo	Nulo	6
1	Nulo	Nulo	5
2	4	Nulo	6
2	4	Nulo	5
Nulo	2	2	4
Nulo	2	Nulo	6
Nulo	2	Nulo	5
3	Nulo	5	3
3	Nulo	2	4
3	Nulo	Nulo	6
3	Nulo	Nulo	5

UCLM-ESI-BDa

6.26



Álgebra Relacional Operadores – versiones trivaluadas

Selección **MAYBE**

$$\sigma_{\text{editorial}=\text{"Addison"}}(\text{AUTOR_LIBRO})$$

NOMBRE	NACIONALIDAD	INSTITUCION	LIBRO	EDITORIAL
Date, C.J.	Norteamericana	Relational Inst.	DB Systems	Addison
Date, C.J.	Norteamericana	Relational Inst.	SQL Standard	Addison

$$\text{MAYBE } \sigma_{\text{editorial}=\text{"Addison"}}(\text{AUTOR_LIBRO})$$

NOMBRE	NACIONALIDAD	INSTITUCION	LIBRO	EDITORIAL
Date, C.J.	Norteamericana	Relational Inst.	DB Systems	Addison
Date, C.J.	Norteamericana	Relational Inst.	SQL Standard	Addison
Salto, F.	Española	U.P.C.	Nulo	Nulo



Cálculo Relacional

- Mientras que en un lenguaje *algebraico* hay que especificar los operadores que se tienen que aplicar a las relaciones para obtener el resultado, en el cálculo relacional (CR) sólo es preciso indicar cuál es el resultado que se quiere obtener.
- Este tipo de lenguajes se llaman *predicativos* porque utilizan el **cálculo de predicados** para construir expresiones.
- DATE (1990) dice que los lenguajes basados en el CR son *descriptivos*, mientras que los algebraicos son *prescriptivos*.
- Los lenguajes de cálculo relacional pueden ser de dos tipos:
 - **orientados a tuplas**, en los que una variable se interpreta como si representase las tuplas de una relación.
 - **orientados a dominios**, en los que una variable se interpreta como si representase los valores de un dominio.



Cálculo Relacional De Tuplas

- Las consultas a una BD relacional en CRT son expresiones del tipo

$$\{ x \mid F(x) \}$$

donde **F** es una *fórmula bien formada* que tiene como única *variable-tupla libre* a **x** (según las reglas de la lógica de predicados de primer orden).

- Ejemplos:

- La unión de dos relaciones **R** \cup **S** puede expresarse en CRT así:

$$\{ t \mid R(t) \vee S(t) \}$$

- La diferencia de dos relaciones **R** - **S** se expresa de la siguiente forma:

$$\{ t \mid R(t) \wedge \neg S(t) \}$$

UCLM-ESI-BDa

6.29



Cálculo Relacional De Tuplas – ejemplos de consultas

- Dado el esquema de BD siguiente:

sucursal(nombre, activo, ciudad)

cliente(nombre, calle, ciudad)

depósito(sucursal, n-cuenta, cliente, saldo)

préstamo(sucursal, n-prestamo, cliente, importe)

- Encontrar la sucursal, n-prestamo, cliente e importe para los préstamos mayores de 3000 €:

$$\{ t \mid \text{préstamo}(t) \wedge t.\text{importe} > 3000 \}$$

- Obtener todos los clientes que tienen préstamos de más de 3000 €:

$$\{ t \mid \text{cliente}(t) \wedge \exists s(\text{préstamo}(s) \wedge t.\text{nombre} = s.\text{cliente} \wedge s.\text{importe} > 3000) \}$$

- Encontrar a todos los clientes que tienen una cuenta en la sucursal 'Paz', pero que no han sacado un préstamo en esa sucursal:

$$\{ t \mid \text{cliente}(t) \wedge \exists s(\text{depósito}(s) \wedge t.\text{nombre} = s.\text{cliente} \wedge s.\text{sucursal} = \text{'Paz'}) \wedge \neg \exists u(\text{préstamo}(u) \wedge t.\text{nombre} = u.\text{cliente} \wedge u.\text{sucursal} = \text{'Paz'}) \}$$

UCLM-ESI-BDa

6.30



Cálculo Relacional De Dominios

- La forma de expresar en este lenguaje una consulta a la BD es:

$$\{ x_1, x_2, \dots, x_k \mid F(x_1, x_2, \dots, x_k) \}$$

tal que x_1, x_2, \dots, x_k son las únicas **variables libres** en la **fórmula bien formada F**.

- Ejemplos:

- La unión de dos relaciones $R \cup S$ se expresa en CRD así:

$$\{ t_1, t_2, \dots, t_k \mid R(t_1, t_2, \dots, t_k) \vee S(t_1, t_2, \dots, t_k) \}$$

- La diferencia de dos relaciones $R - S$ se expresa de la siguiente forma:

$$\{ t_1, t_2, \dots, t_k \mid R(t_1, t_2, \dots, t_k) \wedge \neg S(t_1, t_2, \dots, t_k) \}$$

UCLM-ESI-BDa

6.31



Cálculo Relacional De Dominios – ejemplos de consultas

- Para el esquema de BD utilizado antes en CRT:

sucursal(nombre, activo, ciudad)

cliente(nombre, calle, ciudad)

depósito(sucursal, n-cuenta, cliente, saldo)

préstamo(sucursal, n-prestamo, cliente, importe)

- Encontrar la sucursal, n-prestamo, cliente e importe para los préstamos mayores de 3000 €:

$$\{ s, t, u, v \mid \text{préstamo}(s, t, u, v) \wedge v > 3000 \}$$

- Obtener todos los clientes que tienen préstamos de más de 3000 €:

$$\{ t_1, t_2, t_3 \mid \text{cliente}(t_1, t_2, t_3) \wedge \exists s_1 \exists s_2 \exists s_3 (\text{préstamo}(s_1, s_2, t_1, s_3) \wedge s_3 > 3000) \}$$

- Encontrar a todos los clientes que tienen una cuenta en la sucursal 'Paz', pero que no han sacado un préstamo en esa sucursal:

$$\{ t_1, t_2, t_3 \mid \text{cliente}(t_1, t_2, t_3) \wedge \exists s_1 \exists s_2 \text{depósito}('Paz', s_1, t_1, s_2) \wedge \neg \exists u_1 \exists u_2 \text{préstamo}('Paz', u_1, t_1, u_2) \}$$

UCLM-ESI-BDa

6.32



SQL Historia

- 1975: Prototipo creado por IBM, conocido como SEQUEL.
- 1977: Cambia el nombre a **SQL (Structured Query Language)**.
- 1979: **Primer SGBDR** comercial basado en el (ORACLE).
- 1986: Aprobada la norma SQL/ANSI.
- 1987: **Primer estándar internacional** de **ISO-9075**.
- 1989: Nueva versión **ISO SQL-89** que añade integridad referencial básica.
- 1992: Versión **SQL-92** que amplía notablemente la anterior.
- 1999: Versión **SQL-99** que incorpora aspectos de **orientación a objetos** y amplía notablemente el lenguaje. Ya está incorporado a los principales SGBD-OR. (1126 pgs).
- 2003: Versión **SQL:2003** (estándar de 2000 páginas).

UCLM-ESI-BDa

6.33



SQL Historia

- **SQL-2003: ISO/IEC 9075-n:2003**
 - Consta de varias partes. Algunas no están concluidas todavía.
 - **1- Framework:** Introducción con el **marco de trabajo conceptual general**.
 - **2- Foundation:** Define las estructuras de datos y operaciones para trabajar con BD relacionales y especifica la **semántica y sintaxis del lenguaje**.
 - **3- Call-Level Interface (SQL/CLI):** **SQL embebido**, uso de órdenes SQL desde otros lenguajes de programación generales.
 - **4- Persistent Stored Modules (SQL/PSM):** **Procedimientos almacenados**.
 - **5- Host Language Bindings (SQL/Bindings):** entre otros, incluye "On-Line Analytical Processing" (**SQL/OLAP**).
 - **9- Management of External Data (SQL/MED):** gestión de datos externos a una BD relacional mediante el uso de "wrappers" y "datalinks".
 - **10- Object Language Bindings (SQL/OLB):** SQL embebido en Java. También conocido como **SQLJ**.
 - **11- Information and Definition Schemas (SQL/Schemata):** Establece las estructuras y contenido del **DEFINITION_SCHEMA**, es decir, los **metadatos** internos.
 - **13- SQL Routines and Types Using the Java TM Programming Language (SQL/JRT):** utilizar **métodos y clases Java** como si fuesen rutinas-SQL y tipos estructurados SQL.
 - **14- XML-Related Specifications (SQL/XML):** Maneras de utilizar SQL en conjunción con **XML**.

UCLM-ESI-BDa

6.34



SQL

Principales Conceptos

En **SQL-99** y **SQL-2003**: (sin orientación a objetos)

- Catálogo
- Esquema
- Dominio
- Tabla
- Columna
- Restricción de integridad
 - De tabla
 - De dominio
 - Aserción
- Vista
- Disparador

Se estudia en detalle en las prácticas de laboratorio.

Ver apéndice con la sintaxis de SQL.

UCLM-ESI-BDa

6.35



SQL

LDD – catálogo y esquema

- **Catálogo** (catalog)
 - **Conjunto de esquemas**. Proporciona un mecanismo adicional para calificar nombres de elementos:
 - **<catalogo>.<esquema>.<elemento>**
 - En cada catálogo existe un esquema especial (**DEFINITION_SCHEMA**) que contiene una colección de tablas base con los **metadatos**:
 - **USERS, TABLES, COLUMNS, DOAMINS, KEY_COLUMN_USAGE, ...**
- **Esquema** (schema)
 - Representa la estructura de **una base de datos**.
 - Tiene un nombre, un identificador del propietario, y descriptores de **elementos de esquema** (schema objects).
 - Los elementos son **dominios, tablas, vistas, restricciones, disparadores** y otros constructores.

UCLM-ESI-BDa

6.36



SQL

LDD – catálogo y esquema

<definición de esquema> ::=

```
CREATE SCHEMA <cláusula de nombre de esquema>
[ <elemento de esquema> ... ]
```

<cláusula de nombre de esquema> ::=

```
<nombre de esquema>
| AUTHORIZATION <id. de autorización de usuario>
| <nombre de esquema> AUTHORIZATION <id. de autorización de
usuario>
```

<elemento de esquema> ::=

```
<definición de tabla>
| <definición de vista>
| <definición de dominio>
| <definición de aserción>
| <definición de disparador>
| <asignación de privilegio> | ....
```

```
CREATE SCHEMA Biblioteca
AUTHORIZATION UCLM;
```

UCLM-ESI-BDa

6.37



SQL

LDD – dominios

- **Dominio** (domain)

- Tienen un nombre, un tipo de datos y, opcionalmente, un valor por defecto y una restricción.
- La restricción limita los valores del tipo de datos.

<definición de dominio> ::=

```
CREATE DOMAIN <nombre de dominio> [ AS ] <tipo de dato predefinido>
[ <cláusula de defecto> ]
[ <restricción de dominio> ]
```

<cláusula de defecto> ::= DEFAULT

```
{ <literal>
| <función de valor tiempo/fecha>
| USER | CURRENT_USER
| CURRENT_ROLE | SESSION_USER
| SYSTEM_USER | CURRENT_PATH | ... }
```

```
CREATE DOMAIN Tipos_Doc CHAR(1)
CHECK (VALUE("A", "L"));
```

UCLM-ESI-BDa

6.38



SQL LDD – tablas

- **Tabla Base** (base table)
 - Cada **descriptor de una tabla base** contiene:
 - Nombre de la tabla.
 - Lista de **descriptores de columnas**.
 - Indicador del **tipo de tabla**.
 - Opcionalmente, descriptores de **restricciones de tabla**.
 - Conjunto de una o varias **dependencias funcionales**.
 - Conjunto de una o varias **claves candidatas**.
 - La clave candidata principal (**clave primaria**).
 - Los **tipos de tablas base** son:
 - Persistentes
 - Temporales:
 - Globales (**GLOBAL TEMPORARY**)
 - Locales (**LOCAL TEMPORARY**)

UCLM-ESI-BDa

6.39



SQL LDD – tablas

<definición de tabla> ::=
CREATE [<alcance>] **TABLE** <nombre>
 (<elemento de tabla> [{, <elemento de tabla>} ...])

<alcance> ::= { **GLOBAL** | **LOCAL** } **TEMPORARY**

<elemento de tabla> ::=
{ <definición de columna>
 | <definición de restricción de tabla> | .. }

```
CREATE TABLE Editorial  
(Codigo_E Codigos,  
Nombre_E Nombres NOT  
NULL,  
Direccion Dirs NOT NULL,  
Ciudad Lugares NOT NULL,  
PRIMAY KEY (Codigo_E),  
UNIQuE (Nombre_E));
```

UCLM-ESI-BDa

6.40



SQL LDD – columnas

- **Columna** (column)

- Tienen un nombre, un **tipo de dato** o dominio asociado, un **valor por defecto** (opcional) y un indicador de si pueden **admitir valores nulos**.
- También existen **columnas generadas**, cuyo valor se obtiene a partir de las demás columnas de la tabla evaluando una expresión.
- Los **tipos de datos predefinidos** son:
 - CHARACTER
 - CHARACTER VARYING
 - CHARACTER LARGE OBJECT
 - BINARY LARGE OBJECT
 - NUMERIC
 - DECIMAL
 - SMALLINT
 - INTEGER
 - BIGINT
 - FLOAT
 - REAL
 - DOUBLE PRECISION
 - BOOLEAN
 - DATE
 - TIME
 - TIMESTAMP
 - INTERVAL

UCLM-ESI-BDa

6.41



SQL LDD – columnas

<definición de columna> ::=

<nombre> [<tipo de dato> | <dominio>]
[<cláusula de defecto> | ... | <cláusula de generación>]
[<definición de restricción de columna>...]

<cláusula de generación> ::=

GENERATED ALWAYS AS (<expresión valuada>)

<definición de restricción de columna> ::=

[**CONSTRAINT** <nombre de restricción>] <restricción de columna>
[<modo de restricción>]

UCLM-ESI-BDa

6.42



SQL

LDD – tipos de restricciones de integridad

- **Restricción de Integridad** (integrity constraint)
 - De **tabla**
 - Unicidad (unique).
 - Clave primaria (primary key).
 - Integridad referencial (foreign key - references).
 - Control (check).
 - De **dominio**
 - Se aplican a cada columna (de cualquier tabla) que está basada en ese dominio, actuando como una restricción de tabla.
 - Incluyen una **condición de búsqueda** referida a los valores del dominio.
 - **Aserción**
 - Una restricción de control (check constraint) definida independientemente de cualquier tabla.
 - Incluyen una **condición de búsqueda** que puede referir al contenido de filas individuales de una o varias tablas, al contenido de tablas enteras o a un estado determinado por las relaciones entre varias tablas.

UCLM-ESI-BDa

6.43



SQL

LDD – comprobación de restricciones de integridad

- Todas las restricciones son comprobadas con los mismos mecanismos y procedimientos:
 - Cada restricción tiene una **condición** que se evalúa a cierto o a falso.
 - Cada restricción tiene un **modo** (constraint mode):
 - **DEFERRED** => diferido
 - **IMMEDIATE** => inmediato
 - Las restricciones con el modo **IMMEDIATE** son comprobadas **después de la ejecución de cada sentencia SQL**.
 - Las restricciones con el modo **DEFERRED** son comprobadas **al final de la ejecución de un grupo lógico de sentencias SQL (transacción)**.
 - Cada restricción tiene un **modo por defecto** que es el utilizado al iniciar una nueva sesión SQL o una nueva transacción.

```
<modo de restricción> ::=  
{ INITIALLY DEFERRED | INITIALLY IMMEDIATE }  
[ [ NOT ] DEFERRABLE ]
```

UCLM-ESI-BDa

6.44



SQL

LDD – restricciones de tabla

- **Unicidad** (**unique**).
 - Dos filas de la tabla no pueden tener el mismo valor no nulo en las columnas indicadas.
 - Puede aplicarse al conjunto completo de columnas (**UNIQUE (VALUE)**).
- **Clave primaria** (**primary key**).
 - Es una restricción de unicidad pero en la cual las columnas no pueden tomar valor nulo (**integridad de entidades**).
- **Integridad referencial** (**foreign key- references**).
 - Representa una **clave ajena**.
 - Incluye una o más columnas referenciantes (referencing columns) y las correspondientes columnas referenciadas (referenced columns) de una tabla base referenciada (referenced table), que puede ser la referenciante.
 - Las columnas referenciadas deben tener una restricción de unicidad en la tabla referenciada.
 - Se puede establecer un modo de comparación total (**MATCH FULL**) o parcial (**MATCH PARTIAL**).
- **Control** (**check**).
 - Incluye una **condición de búsqueda** que sólo puede referir a elementos de la tabla que la incluye.

UCLM-ESI-BDa

6.45



SQL

LDD – restricciones de tabla

```
<definición de restricción de tabla> ::=
  [ CONSTRAINT <nombre de restricción> ] <restricción de tabla>
  [ <modo de restricción> ]

<restricción de tabla> ::=
  { <definición de restricción de unicidad>
  | <definición de restricción referencial>
  | <definición de restricción de control> }

<definición de restricción de unicidad> ::=
  { { UNIQUE | PRIMARY KEY } ( <lista de columnas> )
  | UNIQUE ( VALUE ) }

<definición de restricción de control> ::=
  CHECK (<condición de búsqueda>)
```

UCLM-ESI-BDa

6.46



SQL

LDD – restricciones de tabla

```
<definición de restricción referencial> ::=
    FOREIGN KEY ( <lista columnas referenciantes> )
    <especificación de referencia>

<especificación de referencia> ::=
    REFERENCES <nombre tabla referenciada> [ ( <lista columnas referenciadas> ) ]
    [ MATCH { FULL | PARTIAL | SIMPLE } ]
    [ <acciones referenciales> ]

<acciones referenciales> ::=
    { <regla de modificación> [ <regla de borrado> ]
    | <regla de borrado> [ <regla de modificación> ] }

<regla de modificación> ::=
    ON UPDATE { CASCADE | SET NULL | SET DEFAULT | RESTRICT | NO ACTION }

<regla de borrado> ::=
    ON DELETE { CASCADE | SET NULL | SET DEFAULT | RESTRICT | NO ACTION }
```

UCLM-ESI-BDa

6.47



SQL

LDD – restricciones de tabla

- Las **restricciones de columna** son restricciones asociadas a una única columna de una tabla:

```
<restricción de columna> ::=
    { NOT NULL
    | { UNIQUE | PRIMARY KEY }
    | <especificación de referencia>
    | <definición de restricción de control> }
```

UCLM-ESI-BDa

6.48



SQL

LDD – restricciones de dominio y aserciones

```
<restricción de dominio> ::=  
  [ CONSTRAINT <nombre de restricción> ]  
  CHECK (<condición de búsqueda>)  
  [ <modo de restricción> ]  
  
<definición de aserción> ::=  
  CREATE ASSERTION <nombre de restricción>  
  CHECK (<condición de búsqueda>)  
  [ <modo de restricción> ]
```

UCLM-ESI-BDa

6.49



SQL

LDD – restricciones de dominio y aserciones

```
CREATE TABLE Documento  
(Tipos Tipos_Doc,  
Cod_Doc CHAR(4),  
Titulo CHAR(25) NOT NULL,  
Idioma Idiomas,  
Nombre_E Nombres,  
Año INTEGER(4) CHECK (Año > 1950),  
ISBN INTEGER(10),  
PRIMARY KEY (Tipo, Cod_Doc),  
UNIQUE(ISBN)  
CHECK ((Tipo = "A" AND  
        ISBN IS NULL AND  
        Nombre_E IS NULL  
        OR  
        (Tipo = "L" AND  
        ISBN IS NOT NULL  
        NOMBRE_E IS NOT NULL)),  
FOREIGN KEY (Nombre_E) REFERENCES TO Editorial  
ON UPDATE CASCADE  
ON DELETE NO ACTION));
```

UCLM-ESI-BDa

6.50



SQL

LDD – restricciones de dominio y aserciones

```
CREATE ASSERTION
  Idiomas_No_Usados_Por_Editoriales_En_Madrid
CHECK (NOT EXISTS
  (SELECT * FROM
  Documento NATURAL JOIN Editorial
  WHERE Idioma IN ("F", "A")
  AND Ciudad = "Madrid"));
```

UCLM-ESI-BDa

6.51



SQL

LDD – vistas

- **Vista** (view)
 - Las vistas pueden ser **modificables** (**WITH CHECK OPTION**) dependiendo de diversos factores.
 - Un **descriptor de vista** incluye:
 - El **nombre** de la vista.
 - La **expresión de la consulta**.
 - Un indicador de si es modificable.

<definición de vista> ::=

```
CREATE [ RECURSIVE ] VIEW <nombre de tabla> ( <lista de columnas> )
AS <expresión de consulta> [ WITH CHECK OPTION ]
```

- Además de las vistas normales o regulares existe otro tipo de vistas basadas en el uso de tipos de datos definidos por el usuario llamadas **referenciables** (referenceable view).

UCLM-ESI-BDa

6.52



SQL LDD – vistas

```
CREATE VIEW Libro  
AS SELECT *  
FROM Documento  
WHERE Tipo = "L";
```



SQL LDD – disparadores

- **Disparador** (trigger)
 - Está asociado a una única tabla base.
 - Es el concepto clave para implementar **BD activas**.
 - Tiene tres partes:
 - Un **evento**: indica la acción sobre la tabla base que causará que se active el disparador.
 - **INSERT**, **DELETE**, o **UPDATE**
 - Un **tiempo de acción**: indica cuando se activará el disparo.
 - **BEFORE** => antes del evento.
 - **AFTER** => después del evento.
 - Una **acción**: Se llevan a cabo si ocurre el evento. Puede ser de dos tipos:
 - Una sentencia SQL ejecutable (SQL executable statement).
 - Un bloque atómico de sentencias SQL ejecutables.



SQL

LDD – disparadores

```
<definición de disparador> ::=
CREATE TRIGGER <nombre del disparador> { BEFORE | AFTER } <evento>
ON <nombre tabla> [ REFERENCING <lista de tablas/variables de transición> ]
<acción disparada>

<evento> ::=
{ INSERT | DELETE | UPDATE [ OF <lista de columnas> ] }

<lista de tablas/variables de transición> ::=
{ OLD [ ROW ] [ AS ] <nombre de variable>
| NEW [ ROW ] [ AS ] <nombre de variable>
| OLD TABLE [ AS ] <nombre de tabla>
| NEW TABLE [ AS ] <nombre de tabla> } ...

<acción disparada> ::=
[ FOR EACH { ROW | STATEMENT } ]
[ WHEN (<condición de búsqueda>) ]
{ <sentencia SQL ejecutable>
| BEGIN ATOMIC { <sentencia SQL ejecutable> ; } ... END
```

UCLM-ESI-BDa

6.55



SQL

LDD – disparadores

Sean las dos tablas siguientes:

```
DEPT (N_DEPT, LOCALIDAD, ..., NUM_EMP)
EMP (COD_EMP, NOMBRE, N_DEPT)
```

Clave ajena EMPLEADO.N_DEPT→DEPT

```
CREATE TRIGGER borrar_emp
AFTER DELETE ON emp
FOR EACH ROW
(UPDATE dep.num_emp = det.num_emp -1
WHERE dept.n_dept = emp.n_dept);

CREATE TRIGGER insertar_emp
AFTER INSERT ON emp
FOR EACH ROW
(UPDATE dept.num_emp = dept.num_emp + 1
WHERE dept.n_dept = emp.n_dept);
```

UCLM-ESI-BDa

6.56



SQL

LDD – disparadores

```
CREATE TRIGGER modificar_emp
AFTER UPDATE OF n_dept ON emp
REFERENCING OLD_TABLE AS v_emp
NEW_TABLE AS n_emp
FOR EACH ROW
(UPDATE dept
SET num_emp = num_emp + 1
WHERE dept.n_dept = n_emp.n_dept);
(UPDATE DEPT
SET num_emp = num_emp -1
WHERE Dept.n_dept = v_emp.n_dept);
```

UCLM-ESI-BDa

6.57



SQL

Manipulación de Datos

- Existen tres formas de manipular BD con SQL:
 - **Interactivamente**, invocando escribiendo las órdenes directamente en algún programa que hace de interfaz y las envía al SGBD.
 - Mediante el uso de **SQL Embebido**, que consiste en insertar órdenes SQL dentro de programas escritos en algún lenguaje genérico de programación que hace de anfitrión (C, Java, etc.).
 - Creando **módulos SQL** que se invocan desde programas escritos en otros lenguajes.
- Las principales sentencias del **LMD interactivo** son:
 - **SELECT** (consultar)
 - **INSERT** (añadir)
 - **DELETE** (eliminar)
 - **UPDATE** (modificar)

Se estudian en detalle en las prácticas de laboratorio

UCLM-ESI-BDa

6.58



SQL Consultas

- Una consulta básica en SQL tiene la forma:

```
select A1, A2, ... , An
from R1, R2, ... , Rm
where P
```

Siendo A_i atributos, R_i relaciones y P un predicado construido de forma similar a los explicados en el AR para el operador de selección, pero utilizando AND, OR y NOT como operadores lógicos.

- Esta consulta es equivalente a la expresión siguiente del AR:

$$\Pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_m))$$

- Es decir, SQL forma el **producto cartesiano** de las relaciones que se indican en la cláusula from, a continuación realiza una **selección** utilizando el predicado de la cláusula where, y **proyecta** el resultado a los atributos de la cláusula select.
- Cualquier operador del AR tiene una expresión equivalente en SQL utilizando las múltiples maneras de la sentencia SELECT.

UCLM-ESI-BDa

6.59



SQL Consultas – diferencia

```
SELECT* FROM Editorial
EXCEPT
SELECT * FROM Librería;
```

EDITORIAL

CODIGO	NOMBRE	DIRECCION	CIUDAD
001	RAMA	CARRETERA DE CANILLAS 144	MADRID
002	ADDISON-WESLEY	SUNSET ST. 4	READING
003	McGRAW-HILL	181 AVENUE	NEW YORK
004	PARANINFO	SOL 3	MADRID
005	ANAYA	GOYA 8	MADRID

LIBRERIA

CODIGO	NOMBRE	DIRECCION	CIUDAD
001	RAMA	CARRETERA DE CANILLAS 144	MADRID
002	ADDISON-WESLEY	SUNSET ST. 4	READING
004	PARANINFO	SOL 3	MADRID

EDITORIAL - LIBRERIA

CODIGO	NOMBRE	DIRECCION	CIUDAD
003	McGRAW-HILL	181 AVENUE	NEW YORK
005	ANAYA	GOYA 8	MADRID

UCLM-ESI-BDa

6.60



SQL Consultas – unión

PROYECTOS

TÍTULO	IDIOMA	AÑO
MIMO	E	1996
ENEAS	E	1996
HERACLITO	I	1995

DOCUMENTO

TIPO	COD_DOC	TÍTULO	IDIOMA	NOMBRE_E	AÑO	ISBN
L	001	CONCEPCION BD	E	RAMA	1993	84-7894-083-S
L	002	AN INTRODUCTION DBS	I	ADDISON-WESLEY	1995	0-201-54329-X
L	003	A GUIDE TO SQL	I	ADDISON-WESLEY	1996	0-201-58432-I
L	004	RELATIONAL DB	I	ADDISON-WESLEY	1995	0-201-55483-X
L	005	ANALISIS SI	E	RAMA	1996	84-7897-233-I
L	006	ANALISIS DE SI	E	ANAYA	1993	84-7900-080-I
L	007	COMPILADORES	E	PARANINFO	1992	84-7884-033-I
A	001	ER MODEL	I	-	1976	-
A	002	RELATIONAL MODEL	I	-	1970	-
A	003	LENGUAJE SQL3	E	-	1995	-
A	004	SQL3 TRADEOFFS	I	-	1995	-
A	005	BASES DE DATOS	E	-	1996	-

UCLM-ESI-BDa

6.61



SQL Consultas – unión

```
SELECT * FROM Proyecto
UNION CORRESPONDING (titulo, idioma, año)
SELECT * FROM Documento;
```

PROYECTOS UNION DOCUMENTO

TÍTULO	IDIOMA	AÑO
CONCEPCION Y DISEÑO DE BD	E	1993
AN INTRODUCTION TO DATABASE SYSTEMS	I	1995
A GUIDE TO SQL STANDARD	I	1996
RELATIONAL DATABASE : SELECTED WRITINGS	I	1995
ANALISIS Y DISEÑO DE APLICACIONES	E	1996
ANALISIS DE SI	E	1993
COMPILADORES E INTERPRETES	E	1992
ENTITY RELATIONSHIP MODEL	I	1976
THE RELATIONAL MODEL	I	1970
LENGUAJE SQL3	E	1995
SQL3 TRADEOFFS	I	1995
BASES DE DATOS	E	1996
MIMO	E	1996
ENEAS	E	1996
HERACLITO	I	1995

UCLM-ESI-BDa

6.62



SQL Consultas – producto cartesiano

SQL-89

```
SELECT Documento.*, Editorial.*
FROM Documento, Editorial;
```

SQL-99

```
SELECT *
FROM Documento CROSS JOIN Editorial;
```

TIPO	COD_DOC	TITULO	IDIOMA	NOMBRE_E	AÑO	ISBN	COD	NOMBRE_E	DIR	CIUDAD
L	001	CONCEPCION BD	E	RAMA	1993	84-7894-083-S	001	RAMA	CTRA CANILLAS 144	MADRID
L	002	INTRODUCTION DBS	I	ADDISON	1995	0-201-54329-X	002	ADDISON	SUNSET 4	READING
L	003	A GUIDE TO SQL	I	ADDISON	1996	0-201-58432-1	003	Mc GRAW	181 AV	NEW YORK
L	004	RELATIONAL DB	I	ADDISON	1995	0-201-55483-X	004	PARANINFO	SOL 3	MADRID

UCLM-ESI-BDa

6.63



SQL Consultas – combinación natural

SQL-89

```
SELECT Documento.*, Codigo, Dir, Ciudad
FROM Documento, Editorial
WHERE Documento.Nombre_E =
Editorial.Nombre;
```

SQL-99

```
SELECT *
FROM Documento NATURAL JOIN Editorial
ON Documento.Nombre_E = Editorial.Nombre;
```

TIPO	COD_DOC	TITULO	IDIOMA	NOMBRE_E	AÑO	ISBN	CODIGO	DIRECCION	CIUDAD
L	001	CONCEPCION BD	E	RAMA	1993	84-7894-083-S	001	CTRA CANILLAS 144	MADRID
L	002	INTRODUCTION DBS	I	ADDISON	1995	0-201-54329-X	002	SUNSET ST. 4	READING
L	003	A GUIDE TO SQL	I	ADDISON	1996	0-201-58432-1	002	SUNSET ST. 4	READING
L	004	RELATIONAL DB	I	ADDISON	1995	0-201-55483-X	002	SUNSET ST. 4	READING
L	005	ANALISIS SI	E	RAMA	1996	84-7897-233-1	001	CTRA CANILLAS 144	MADRID
L	006	ANALISIS DE SI	E	ANAYA	1993	84-7900-080-1	005	GOYA 8	MADRID
L	007	COMPIADORES	E	PARANINFO	1992	84-7884-033-1	004	SOL 3	MADRID

Si las dos columnas que representan nombre de editorial (Documento.Nombre_E y Editorial.Nombre) se llamaran igual:

```
SELECT *
FROM Documento NATURAL JOIN Editorial;
```

UCLM-ESI-BDa

6.64



SQL Consultas – combinación natural externa

por la
izquierda

```
SELECT *
FROM Documento LEFT OUTER JOIN Editorial
ON Documento.Nombre_E = Editorial.Nombre;
```

TIPO	COD_DOC	TITULO	IDIOMA	NOMBRE_E	AÑO	ISBN	CODIGO	NOMBRE_E	DIR	CIUDAD
L	001	CONCEPCION BD	E	RAMA	1993	84-7894-083-S	001	RAMA	CTRA. CANILLAS, 144	MADRID
L	002	AN INTRODUCTION DBS	I	ADDISON-WESLEY	1995	0-201-54329-X	002	ADDISON	SUNSET ST., 4	READING
L	003	A GUIDE TO SQL	I	ADDISON-WESLEY	1996	0-201-58432-1	002	ADDISON	SUNSET ST., 4	READING
L	004	RELATIONAL DB	I	ADDISON-WESLEY	1995	0-201-55483-X	002	ADDISON	SUNSET ST., 4	READING
L	005	ANALISIS SI	E	RAMA	1996	84-7897-233-1	001	RAMA	CTRA. CANILLAS, 4	MADRID
L	006	ANALISIS DE SI	E	ANAYA	1993	84-7900-080-1	005	ANAYA	GOYA, 8	MADRID
L	007	COMPILADORES	E	PARANINFO	1992	84-7884-033-1	004	PARANINFO	SOL, 3	MADRID
A	001	ER MODEL	I	-	1976	-	-	-	-	-
A	002	RELATIONAL MODEL	I	-	1970	-	-	-	-	-
A	003	LENGUAJE SQL3	E	-	1995	-	-	-	-	-
A	004	SQL3 TRADEOFFS	I	-	1995	-	-	-	-	-
A	005	BASES DE DATOS	E	-	1996	-	-	-	-	-

UCLM-ESI-BDa

6.65



SQL Consultas – combinación natural externa

por la
derecha

```
SELECT *
FROM Documento RIGHT OUTER JOIN Editorial
ON Documento.Nombre_E = Editorial.Nombre;
```

TIPO	COD_DOC	TITULO	IDIOMA	NOMBRE_E	AÑO	ISBN	CODIGO	NOMBRE_E	DIR	CIUDAD
L	001	CONCEPCION BD	E	RAMA	1993	84-7894-083-S	001	RAMA	CTRA CANILLAS, 144	MADRID
L	002	AN INTRODUCTION DBS	I	ADDISON-WESLEY	1995	0-201-54329-X	002	ADDISON	SUNSET ST., 4	READING
L	003	A GUIDE TO SQL	I	ADDISON-WESLEY	1996	0-201-58432-1	002	ADDISON	SUNSET ST., 4	READING
L	004	RELATIONAL DB	I	ADDISON-WESLEY	1995	0-201-55483-X	002	ADDISON	SUNSET ST., 4	READING
L	005	ANALISIS SI	E	RAMA	1996	84-7897-233-1	001	RAMA	CTRA. CANILLAS, 4	MADRID
L	006	ANALISIS DE SI	E	ANAYA	1993	84-7900-080-1	005	ANAYA	GOYA, 8	MADRID
L	007	COMPILADORES	E	PARANINFO	1992	84-7884-033-1	004	PARANINFO	SOL, 3	MADRID
							003	M-CRAW	181 AVENUE	NEW YORK

UCLM-ESI-BDa

6.66



SQL Consultas – combinación natural externa

por ambos
lados

```
SELECT *
FROM Documento FULL OUTER JOIN Editorial
ON Documento.Nombre_E = Editorial.Nombre;
```

TIPO	COD_DOC	TITULO	IDIOMA	NOMBRE_E	AÑO	ISBN	CODIGO	NOMBRE_E	DIR	CIUDAD
L	001	CONCEPCION BD	E	RAMA	1993	84-7894-083-S	001	RAMA	CTRA. CANILLAS, 144	MADRID
L	002	AN INTRODUCCION DBS	I	ADDISON-WESLEY	1995	0-201-54329-X	002	ADDISON	SUNSET ST., 4	READING
L	003	A GUIDE TO SQL	I	ADDISON-WESLEY	1996	0-201-58432-I	002	ADDISON	SUNSET ST., 4	READING
L	004	RELATIONAL DB	I	ADDISON-WESLEY	1995	0-201-55483-X	002	ADDISON	SUNSET ST., 4	READING
L	005	ANALISIS SI	E	RAMA	1996	84-7897-233-I	001	RAMA	CTRA. CANILLAS, 4	MADRID
L	006	ANALISIS DE SI	E	ANAYA	1993	84-7900-080-I	005	ANAYA	GOYA, 8	MADRID
L	007	COMPILADORES	E	PARANINFO	1992	84-7884-033-I	004	PARANINFO	SOL, 3	MADRID
A	001	ER MODEL	I	-	1976	-	-	-	-	-
A	002	RELATIONAL MODEL	I	-	1970	-	-	-	-	-
A	003	LENGUAJE SQL3	E	-	1995	-	-	-	-	-
A	004	SQL3 TRADEOFFS	I	-	1995	-	-	-	-	-
A	005	BASES DE DATOS	E	-	1996	-	-	-	-	-
							003	Mc GRAW	181 AVENUE	NEW YORK

UCLM-ESI-BDa

6.67



QBE

- **Query-By-Example** (QBE) es el ejemplo más característico de lenguaje comercial basado en el cálculo relacional de dominios.
- Fue desarrollado por IBM en 1977.
- Su característica más especial es su *bidimensionalidad*: asistencia de un editor de pantallas que ayuda a formular las consultas en dos dimensiones, y que se ha convertido en un paradigma de interfaz de usuario.
- Para hacer consultas se emplean esqueletos de tablas, que son representaciones gráficas del esquema de las relaciones:

Zona para nombre de tabla		Zona para nombres de columna	
Zona para operador de fila		Zona para datos	

UCLM-ESI-BDa

6.68



QBE

- Sobre los esqueletos de las tablas, el usuario puede realizar operaciones:

TIPO	OPERADOR	OPERACIÓN REALIZADA
SISTEMA	P.	Visualización en pantalla o en impresora
	U.	Modificación
	I.	Inserción
	D.	Borrado
	G.	Agrupamiento de filas
	ALL.	Todas las filas
	UNQ.	No repite filas duplicadas
	AO (n) DO (n)	Clasificación ascendente Clasificación descendente
FUNCIONES	SUM.	Suma
	CNT.	Cuenta
	AVG.	Promedio
	MAX.	Valor máximo
	MIN.	Valor mínimo
COMPARACION	=, >, <	Comparaciones sencillas
	>=, <=, <>	y combinadas
	IN	Compara con lista de valores
	LIKE BETWEEN	Compara con una máscara Compara con un rango
CALCULO	+, -, *, /	Cálculo
LOGICA	AND, OR, NOT	Lógica

UCLM-ESI-BDa

6.69



QBE

Consultas - ejemplo

- Obtener todos los clientes que tienen préstamos de más de 3000 €:
 $\{ t_1, t_2, t_3 \mid \text{cliente}(t_1, t_2, t_3) \wedge \exists s_1 \exists s_2 \exists s_3 (\text{préstamo}(s_1, s_2, t_1, s_3) \wedge s_3 > 300000) \}$

cliente	nombre	calle	ciudad
	P._t ₁	P._t ₂	p._t ₃

préstamo	sucursal	n-préstamo	cliente	importe
			_t ₁	> 300000

UCLM-ESI-BDa

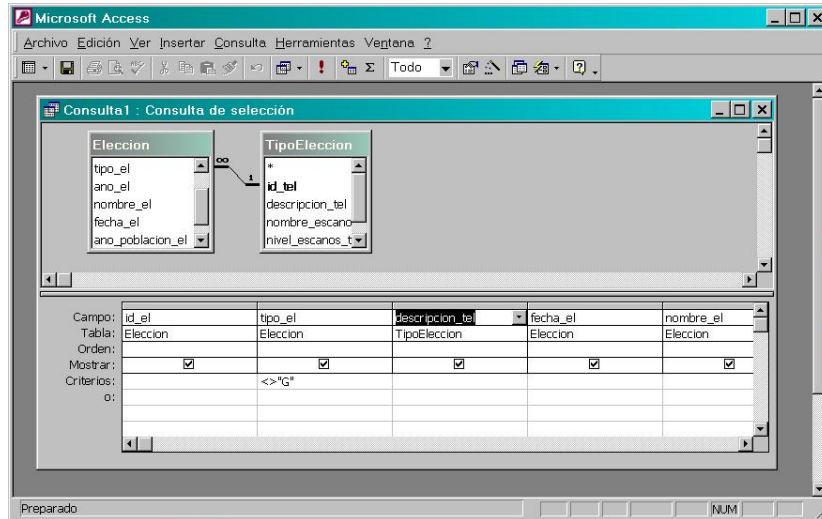
6.70



QBE

Consultas – producto comercial

- QBE implementado en MS Access con interfaz gráfico Windows



UCLM-ESI-BDa

6.71