

# Bases de Datos

Tema 4

## *Lenguajes Relacionales*

*Francisco Ruiz*

*dic-2000*

*documentación preparada con ayuda de Esperanza Marcos (Universidad Rey Juan Carlos) y Mario Piattini  
(Universidad de Castilla-La Mancha)*

## Tema 4

# *Lenguajes Relacionales*

*Complementar con:*

*\* capítulos 6 y 7 del libro “Fundamentos y Modelos de Bases de Datos” (2ª edición). De Miguel, A.; Piattini, M.; Ra-Ma, 1999.*

*\* apuntes de “Formalización del Algebra Relacional” y “Cálculo Relacional” de la asignatura.*

- *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 (versión 92), y*
  - *QBE.*

- **Principales:**

- [de Miguel y Piattini, 1999]
  - caps. 6 y 7
  - De Miguel, A.; Piattini, M.; Fundamentos y Modelos de Bases de Datos (2ª edición). Ra-Ma, 1999.

- **Otras:**

- Elmasri, R.; Navathe, S.B.; Sistemas de Bases de Datos: Conceptos fundamentales (2ª edición). Addison-Wesley, 1997. Capítulos 6, 7 y 8.
- Silberschatz, A.; Korth, H.F.; Sudarshan, S.; Fundamentos de Bases de Datos (3ª edición). McGraw-Hill, 1998. Capítulo 5.
- Ullman, J.D.; Widom, J.; Introducción a los Sistemas de Bases de Datos. Prentice Hall, 1999. Capítulos 4 y 5.

# *Índice*

1. Introducción.
2. Álgebra relacional.
  - 2.1. Operadores primitivos.
  - 2.2. Operadores derivados.
  - 2.3. Operadores de consulta adicionales.
  - 2.4. Operadores con valores nulos.
3. Cálculo relacional.
  - 3.1. Cálculo relacional de tuplas.
  - 3.2. Cálculo relacional de dominios.
4. Lenguajes relacionales comerciales.
  - 4.1. SQL.
  - 4.2. SQL embebido.
  - 4.3. SQL dinámico.
  - 4.4. QBE.

- 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).

- 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.

- En álgebra relacional (AR) la dinámica del MR la constituyen una colección de operadores de alto nivel que, aplicados a las 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:

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



Podemos clasificar los operadores del AR de tres formas diferentes:

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.

- **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:

$$\text{RELACION\_NUEVA} \leftarrow O(R)$$

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

$$R' \leftarrow R$$

- O renombrar los atributos de la relación original:

$$\text{RELACION\_NUEVA} (A_1, A_2, \dots, A_n) \leftarrow O(R)$$

## Álgebra Relacional

### Operadores primitivos + operadores derivados

Operadores primitivos:

- Restricción (selección)
- Proyección
- Unión
- Diferencia
- Producto Cartesiano

*También pueden ser otros, p.e., con la intersección en vez de la diferencia*

## Selección ( $\sigma$ )

La selección 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

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

Nombre	Nacionalidad	Institución
Saltor	Española	U.P.C.

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

## Proyección ( $\pi$ )

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

```
SELECT nacionalidad  
FROM autor
```

## Unión ( $\cup$ )

La unión de dos relaciones  $r_1$  y  $r_2$ , con esquemas  $R_1$  y  $R_2$  compatibles, 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

## 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.

## Producto Cartesiano ( $\times$ )

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  $r_2$ .



## Producto Cartesiano (x)

### LIBRO

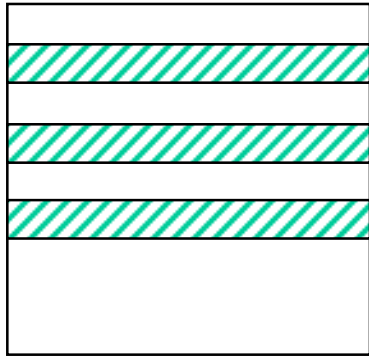
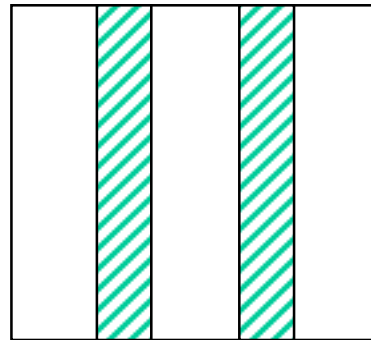
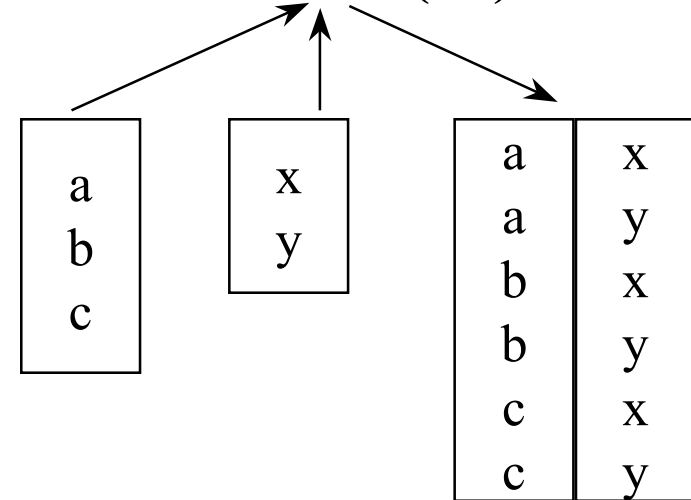
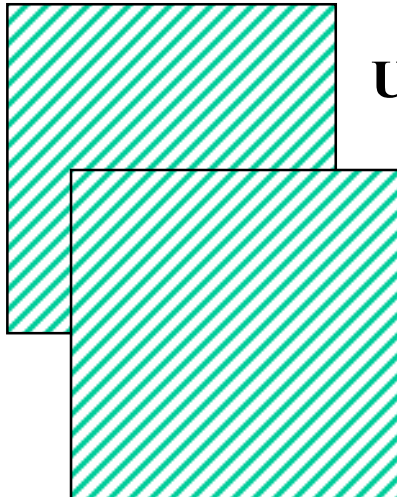
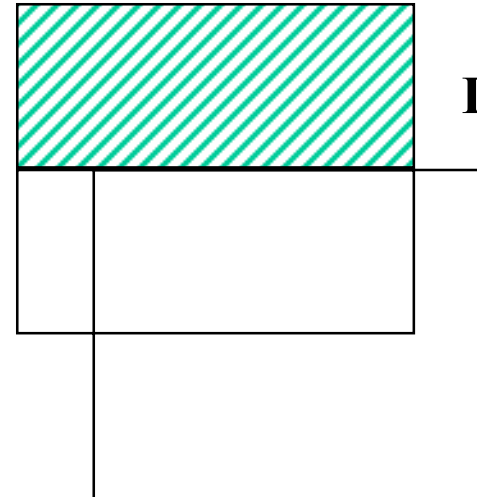
Código	Título	Idioma	Nombre_e
001	Bases de Datos	Español	Ra-ma
003	Diseño de BD	Español	Ra-ma

### 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

**Restricción ( $\sigma$ )****Proyección ( $\pi$ )****Producto (X)****Unión ( $\cup$ )****Diferencia (-)**

## Combinación ( $\theta$ )

La **combinación** 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.

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  $A1_i$  y  $A2_j$ , 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.

## 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).

Cuando el atributo común tiene el mismo nombre en ambas relaciones, se suele omitir la condición de combinación.

## 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

## Intersección ( $\cap$ )

La intersección 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.

***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

## División (:)

La división de una relación R1 (dividendo) por otra relación R2 (divisor) es una relación R (cociente) tal que, al realizarse su combinación con el divisor, todas las tuplas resultantes se encuentran en el dividendo.

### ***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

### ***AUTOR\_EDITORIAL : EDITORIAL***

NOMBRE	NACIONALIDAD
Cervera, J.	Española

## 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 (nº artículos)*

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



## 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

## 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 (*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***)

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

## 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

***MAYBE R1  $\theta$  R2***  
*b=c*

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

## Operadores MAYBE

También existe una versión MAYBE de la restricción

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

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

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
Salton, F.	Española	U.P.C.	Nulo	Nulo

- 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.

- 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:

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

- Ejemplos de consultas para 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)

- a) Encontrar la sucursal, n-prestamo, cliente e importe para los préstamos mayores de 300.000 pts:

$$\{ t \mid \text{préstamo}(t) \wedge t.\text{importe} > 300000 \}$$

- b) Obtener todos los clientes que tienen préstamos de más de 300.000 pts:

$$\{ t \mid \text{cliente}(t) \wedge \exists s(\text{préstamo}(s) \wedge t.\text{nombre} = s.\text{cliente} \wedge s.\text{importe} > 300000) \}$$

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

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

- La forma de expresar en este lenguaje una pregunta a la base de datos 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:

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



- Ejemplos de consultas para el esquema de BD utiliza antes:
  - a) Encontrar la sucursal, n-prestamo, cliente e importe para los préstamos mayores de 300.000 pts:  
 $\{ s, t, u, v \mid \text{préstamo}(s, t, u, v) \wedge v > 300000 \}$
  - b) Obtener todos los clientes que tienen préstamos de más de 300.000 pts:  
 $\{ t_1, t_2, t_3 \mid \text{cliente}(t_1, t_2, t_3) \wedge \exists s_1 \exists s_2 \exists s_3 (\text{préstamo}(s_1, s_2, t_1, s_3) \wedge s_3 > 300000) \}$
  - c) Encontrar a todos los clientes que tienen una cuenta en la sucursal 'Paz', pero que no han sacado un préstamo en esa sucursal:  
 $\{ t_1, t_2, t_3 \mid \text{cliente}(t_1, t_2, t_3) \wedge \exists s_1 \exists s_2 \text{depósito}('Paz', s_1, t_1, s_2) \wedge \neg \exists u_1 \exists u_2 \text{préstamo}('Paz', u_1, t_1, u_2) \}$

## Evolución Resumida:

- 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.
- 1989: Versión ISO SQL-1 o SQL-89 que añade integridad referencial básica.
- 1992: Versión SQL-2 o SQL-92 que amplía notablemente la anterior. Es la utilizada por casi todos los SGBDR.
- 1999: Versión SQL-3 que incorpora aspectos de orientación a objetos y amplía notablemente el lenguaje. Se está empezando a incorporar a los SGBDR.

## Principales Conceptos manejados en SQL-92:

- Esquemas
- Dominios
- Tablas
- Columnas
- Restricciones y reglas de integridad
- Vistas
- Catálogos
- Privilegios
- Transacciones
- ....

*El lenguaje SQL-92 se estudia en detalle en las prácticas de laboratorio.*

*Ver apéndice con la sintaxis de SQL-92.*

- Una consulta en SQL tiene la forma:

**select A1, A2, ... , An**

**from R1, R2, ... , Rm**

**where P**

- Siendo Ai atributos, Ri relaciones y P un predicado construido de forma similar a los explicados en el AR para el operador de selección (s), 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 restricción utilizando el predicado de la cláusula where, y proyecta el resultado a los atributos de la cláusula select.

- El lenguaje SQL se puede utilizar de forma interactiva o como lenguaje huésped (embebido dentro de otro lenguaje de propósito general).

En **SQL Embebido**, las sentencias SQL se van intercalando entre las sentencias del lenguaje anfitrión, precedidas normalmente de la partícula EXEC SQL para que puedan ser distinguidas por los *precompiladores*.

Como puede verse en la figura 7.13 la declaración de variables se introduce en una sección que empieza con la sentencia

EXEC SQL BEGIN DECLARE SECTION

y finaliza con otra sentencia

EXEC SQL END DECLARE SECTION.

Sentencias del lenguaje anfitrión

```
EXEC SQL BEGIN DECLARE SECTION;
Año Date;
Isbn Char(12); ...
```

```
EXEC SQL END DECLARE SECTION;
```

Sentencias del lenguaje anfitrión

```
EXEC SQL SELECT Año
      INTO :Año
      FROM DOCUMENTO
      WHERE Isbn = :Isbn;
IF SQLSTATE = ...
```

?   
 Variables   
 anfitrión

Sentencias del lenguaje anfitrión

```
WHenever SQLERROR ...
```

Sentencias del lenguaje anfitrión

- Las sentencias SQL manejan conjuntos de filas mientras que el lenguaje anfitrión (Cobol, Basic, C) sólo puede manejarlas de una en una.
- Para resolver esta discordancia (conocida como *impedance mismatch*), el SQL soporta el concepto de **cursor**, ofreciendo sentencias para declararlos (DECLARE CURSOR FOR), abrirlos (OPEN), cerrarlos (CLOSE) y leerlos (FETCH).

- Declaración de un cursor:

<declaración de cursor>::=

```
DECLARE <nombre del cursor> [INSENSITIVE][SCROLL] CURSOR  
FOR <sentencia SELECT>
```

- Ejemplo:

```
DECLARE cursor_e CURSOR FOR  
SELECT Código, Nombre, Dir, Ciudad  
FROM Editorial  
ORDER BY Ciudad DESC;
```

- Con la opción SCROLL se permite realizar lecturas (FETCH) de diversas formas: NEXT, PRIOR, FIRST, LAST, ABSOLUTE n o RELATIVE n, siendo "n" el resultado de evaluar una expresión escalar.

- Otro avance del SQL92 lo constituye la posibilidad de utilizar sentencias SQL dinámicas, es decir, sentencias SQL cuyo contenido exacto se conoce sólo en tiempo de ejecución:

```
SELECT Nombre, Dirección  
      FROM Editorial  
      WHERE Ciudad = ?;
```

- donde mediante el símbolo "?" se indica el parámetro que habrá de pasarse a la sentencia durante su ejecución.

- **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:

<b>Zona para nombre de tabla</b>		<b>Zona para nombres de columna</b>	
<b>Zona para operador de fila</b>		<b>Zona para datos</b>	



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

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

## Ejemplo de consulta (ya vista en CRD):

- Obtener todos los clientes que tienen préstamos de más de 300.000 pts:  
 $\{ t_1, t_2, t_3 \mid \text{cliente}(t_1, t_2, t_3) \wedge \exists s_1 \exists s_2 \exists s_3 (\text{préstamo}(s_1, s_2, t_1, s_3) \wedge s_3 > 300000) \}$

cliente	nombre	calle	ciudad
	P._t <sub>1</sub>	P._t <sub>2</sub>	p._t <sub>3</sub>

préstamo	sucursal	n-préstamo	cliente	importe
			_t <sub>1</sub>	> 300000