



# BASES DE DATOS AVANZADAS

## Tema 2

### *Repaso de SQL y Lenguajes Relacionales*

*Univ. Cantabria – Fac. de Ciencias*

*Francisco Ruiz*



## Objetivos

- Conocer los **tipos de lenguajes** para el modelo relacional de datos.
- Conocer los conceptos básicos del lenguaje estándar **SQL:2003**.
- Comprender las características generales del lenguaje **QBE**.



## Contenido

---

- Tipos de Lenguajes Relacionales
- SQL
  - Historia
  - Principales Conceptos
  - Definición de Datos
  - Manipulación de Datos
  - Consultas
- QBE



## Bibliografía

---

- Piattini et al. (2006): Tecnología y Diseño de Bases de Datos.
  - Cap. 8.
- Elmasri y Navathe (2007): Fundamentos de Sistemas de Bases de Datos.
  - Cap. 8.
- Connolly y Begg (2005): Sistemas de Bases de Datos.
  - Caps. 5, 6 y 7.



## 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.



## 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).



## SQL Historia

- **SQL-2003: ISO/IEC 9075-n:2003**
  - **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**). [*no disponible todavía*]
  - **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**.



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



- **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.



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;
```



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"));
```



## 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**)



## 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,
PRIMARY KEY (Codigo_E),
UNIQuE (Nombre_E));
```



## 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



## 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> ]



- **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.



- 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** ]



- **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.



<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>)



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 }



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> }



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> ]
```



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 UPDAE CASCADE  
ON DELETE NO ACTION));
```



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"));



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).



## SQL LDD – vistas

---

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



## 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)



## 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.



## 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



## 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	-



## 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



# 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-I	003	Mc GRAW	181 AV	NEW YORK
L	004	RELATIONAL DB	I	ADDISON	1995	0-201-55483-X	004	PARANINFO	SOL 3	MADRID



# 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-I	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-I	001	CTRA CANILLAS 144	MADRID
L	006	ANALISIS DE SI	E	ANAYA	1993	84-7900-080-I	005	GOYA 8	MADRID
L	007	COMPILADORES	E	PARANINFO	1992	84-7884-033-I	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;
```



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-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	-	-	-	-	-



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-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
							003	M: CRAW	181 AVENUE	NEWYORK



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 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-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



## 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:

<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>	



## 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



## 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 <sub>1</sub>	P._t <sub>2</sub>	p._t <sub>3</sub>

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



# QBE

## Consultas – producto comercial

- QBE implementado en MS Access con interfaz gráfico Windows

