

Modelo Relacional

UTN - FRBA
Ing. en Sistemas de Información
Gestión de Datos

Prof.: Ing. Juan Zaffaroni

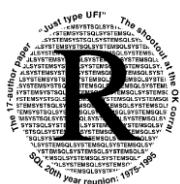
Un poco de historia

- En los últimos 25 años el mundo de la TI experimentó grandes cambios.
 - Nuevas Arquitecturas de aplicaciones.
 - Nuevos Paradigmas de programación.
 - Nuevas herramientas para desarrollo de Software.
 - Pero algo permaneció constante....

Un poco de historia

- **RDBMS** – Relational Data Base Managemet Systems.

En los 70's Edgar Codd **Modelo Relacional** **Algebra Relacional**



ORACLE

 INFORMIX

IBM DB2

**1974
IBM
System R
1er RDBMS
SEQUEL=SQL**

**1977/79
Lawrence
Ellison
SDL luego
Relational Sw
Inc.**

1980 Relational Database Systems

1981
IBM

Concepto

- Propuesto por el Dr. E.F. Codd en 1970, consta de:

Estructura

Recopilación de
objetos o relaciones

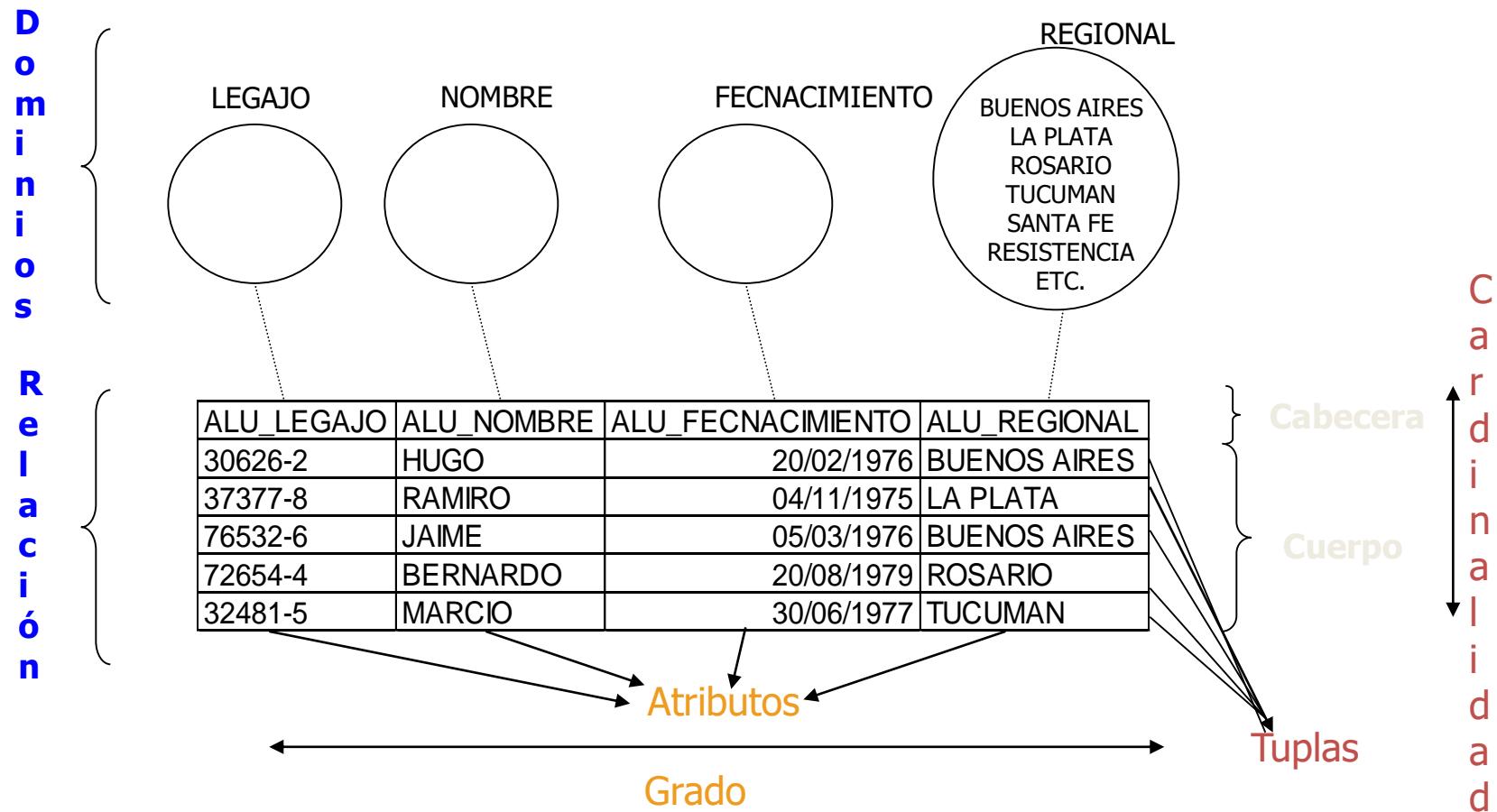
Manipulación
de datos

Operaciones sobre
las relaciones para
producir otras

Integridad

Para obtener precisión
y consistencia

Estructura



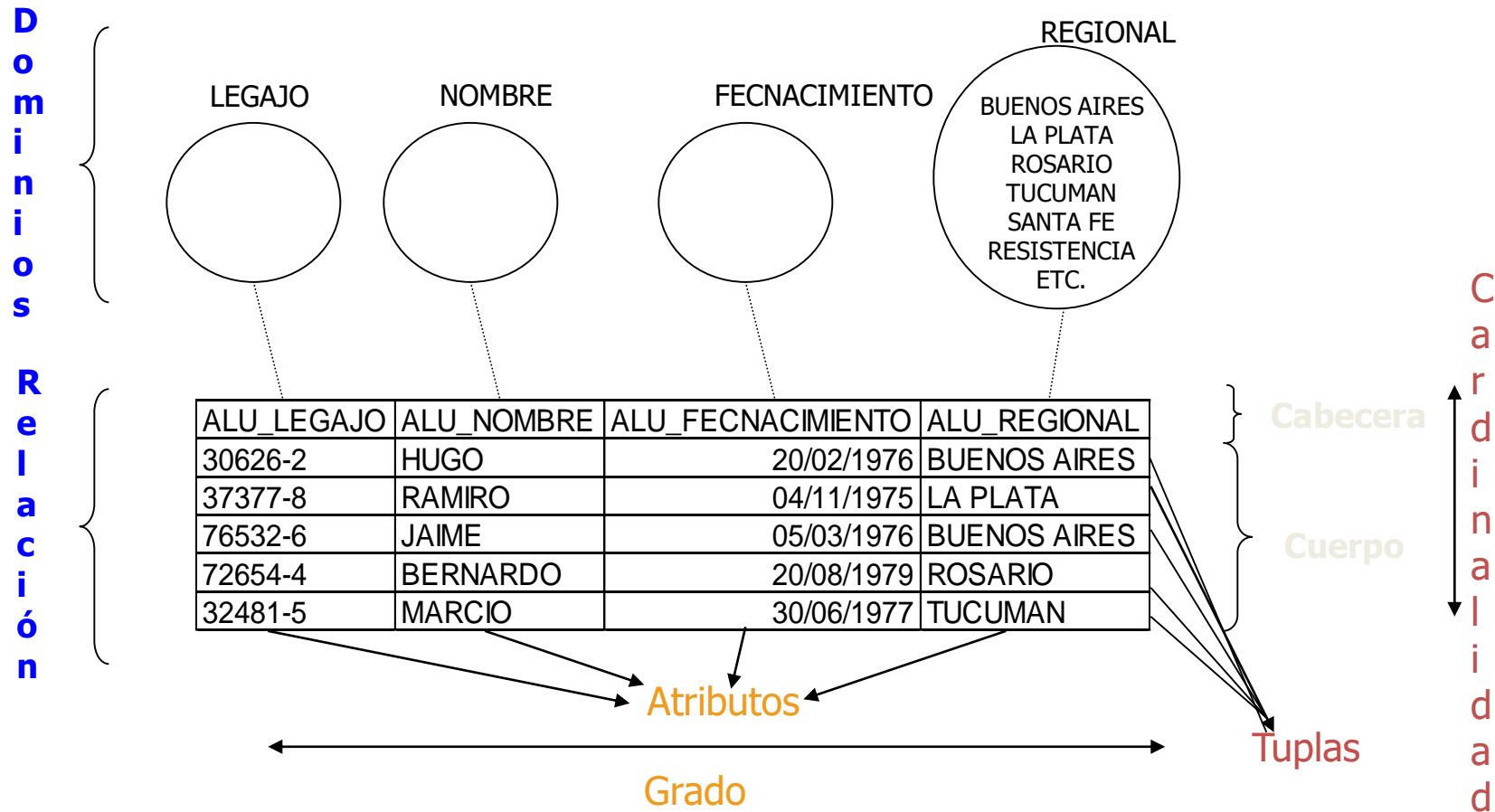
Dominio

Colección de valores, de los cuales los atributos obtienen sus valores reales.

- Menor unidad semántica de información.
- Atómicos (no se pueden descomponer sin perder significado).
- Conjunto de valores escalares de igual tipo.
- No contienen nulos.

ESTRUCTURA

Dominio



Dominio: Importancia

```
SELECT ALUMNO.NOMBRE  
      FROM ALUMNO, CURSO, MATERIA  
     WHERE CURSO.ALU_LEGATO = ALUMNO.ALU_LEGATO  
       AND CURSO.MAT_CODIGO = MATERIA.MAT_CODIGO  
       AND MATERIA.MAT_CODIGO = '85-1346';
```



Consulta bien formulada

```
SELECT ALUMNO.*  
      FROM ALUMNO, CURSO  
     WHERE CURSO.MAT_CODIGO = ALUMNO.ALU_LEGATO  
       AND CURSO.COD_CURSO = '3052';
```



*Consulta mal formulada
(aunque sin errores de sintaxis)*

Dominio: Implementación

-  **Restricciones de Chequeo**
-  **Claves Foráneas**
-  **Definición de "tipos de dato de usuario"**
-  **Definición de convenciones de nombre para los atributos**

Relación

Subconjunto del producto cartesiano de los dominios de valores involucrados.

■ Cabecera: Conjunto fijo de pares atributo-dominio.

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

■ Cuerpo: Conjunto de tuplas que varía con el tiempo.

■ Tupla: Conjunto de pares atributo-valor.

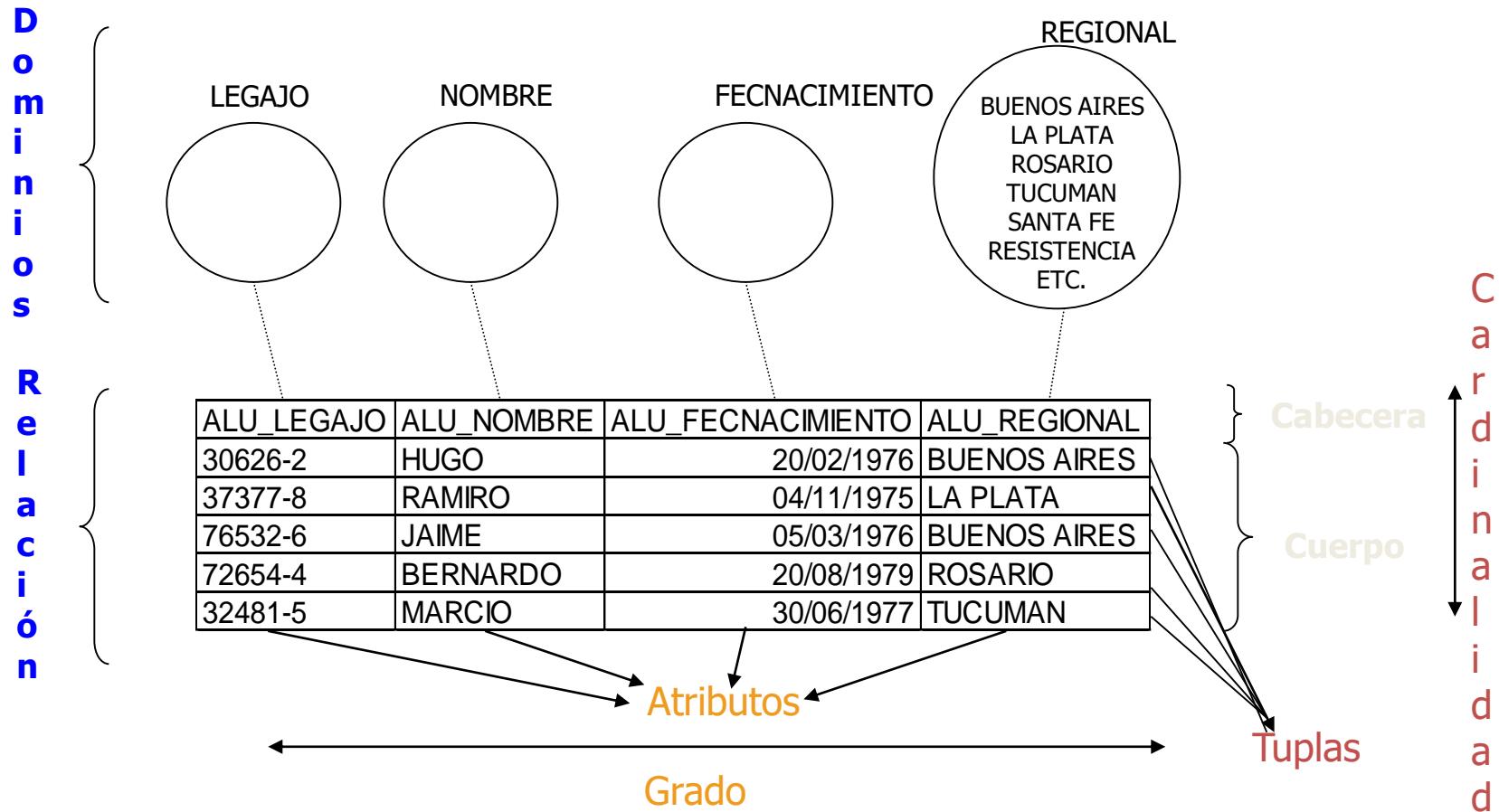
$\{(A_1:V_{i1}), \dots, (A_n:V_{in})\}$ $1 >= i >= m$, m: cardinalidad, n: grado

■ Tabla: Representación de una relación.

■ Base de datos Relacional: BD percibida por el usuario como una colección de relaciones normalizadas de diversos grados que varía con el tiempo.

ESTRUCTURA

Relación



Relación: Propiedades

- || → No hay tuplas repetidas
 - ↳ Toda relación tiene clave primaria
- || → Las tuplas no están ordenadas
- || → Los atributos no están ordenados
- || → Los valores de los atributos son atómicos
 - ↳ La relación está normalizada

Relación: Tipos

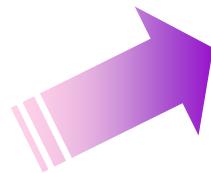
- ★ Tablas – Relaciones Base
- ★ Resultados de consultas
- ★ Resultados intermedios de consultas
- ★ Vistas
- ★ Instantáneas o snapshots
- ★ Tablas Temporales

Integridad

Clave Primaria



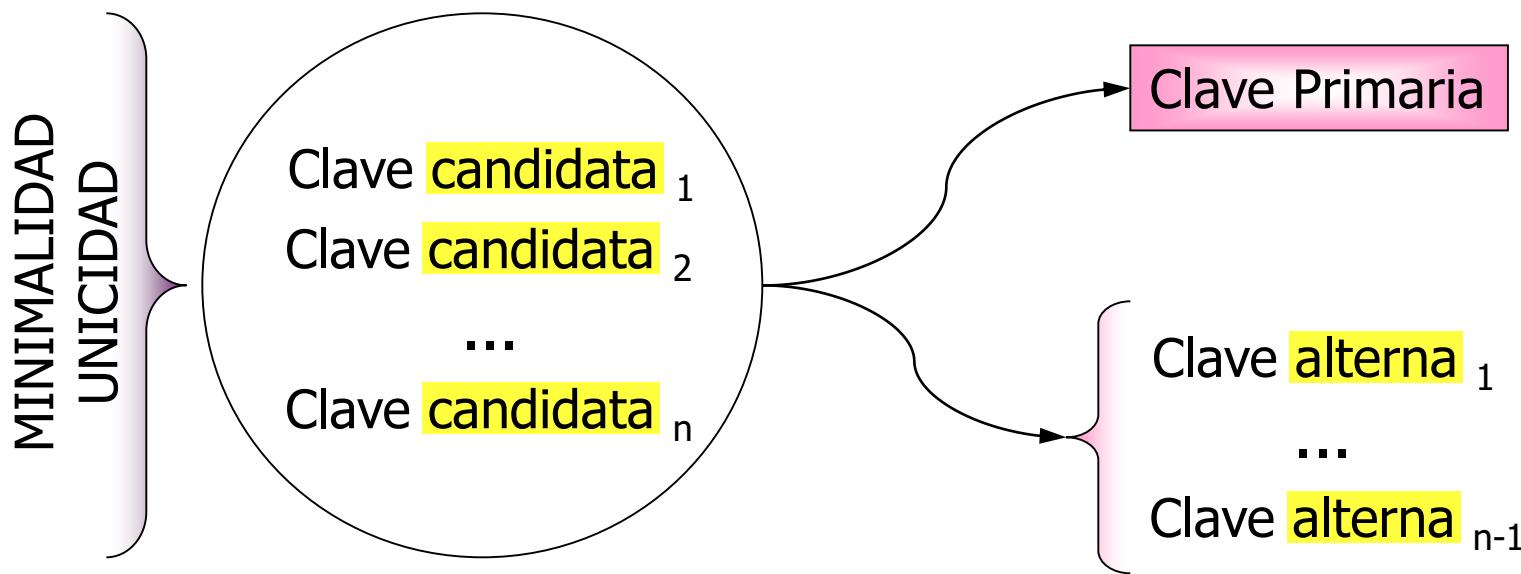
Clave Foránea



Reglas de
Integridad

Clave Primaria

Atributo (o conjunto de atributos) identificador único para la relación.



Clave Primaria

REGLA DE INTEGRIDAD DE LAS ENTIDADES

Ningún componente de la clave primaria de una relación base puede aceptar nulos.

- ⇒ Una tupla es un elemento de una relación, que representa al mundo real.
- ⇒ Las claves primarias, son el mecanismo de identificación en el modelo relacional.
- ⇒ Una base de datos relacional no admite registrar información acerca de algo que no se puede identificar.

Clave Foránea

Atributo (o conjunto de atributos) de una relación (R2) cuyos valores (no nulos) deben coincidir con los de la clave primaria de una relación (R1).

- ➡ La clave primaria en R1 y foránea en R2 están definidas sobre el mismo dominio
- ➡ La clave foránea puede o no ser parte de la clave primaria de R2.
- ➡ R1 y R2 no necesariamente son distintas. Si R1=R2 existe una *relación autorreferencial*.
- ➡ Las claves foráneas deben en ciertas ocasiones aceptar nulos.

Clave Foránea

REGLA DE INTEGRIDAD REFERENCIAL

La base de datos no debe contener valores no nulos de clave foránea para los cuales no existe un valor concordante de clave primaria en la relación referenciada.

- ➲ Mantener un estado consistente de la base de datos.
- ➲ Determinar acciones a llevar a cabo ante operaciones que puedan violar la integridad referencial
 - ✓ Eliminación: RESTRICT / CASCADE / SET NULL
 - ✓ Modificación: RESTRICT / CASCADE / SET NULL
 - ✓ Inserción: RESTRICT

Independencia de Datos

Independencia lógica. Cada aplicación requiere una vista diferente de los mismos datos y no requiere saber todos los atributos de una tabla o el orden real de dichos atributos, ni la distribución de atributos a través de las tablas. Se podrían hacer cambios en tablas sin que afecten a las aplicaciones que no los requieren.

Independencia física. Es posible modificar la estructura de almacenamiento, la distribución física o la técnica de acceso sin afectar las aplicaciones. (crear un índice por nombre para que se resuelva el listado más rápidamente, crear un objeto determinado asociado a una tabla , etc.)

Reglas de Codd

(aseguran que un motor de BD es relacional)

- Independencia entre el motor de base de datos y los programas que acceden a los datos (es posible modificar el motor de base de datos o los componentes de aplicación en forma independiente).
- Representar la información de la estructura de las tablas y sus relaciones mediante un catálogo dinámico basado en el modelo relacional.
- Las reglas de integridad deben guardarse en el catálogo de la base, no en programas de aplicación.

Edgar Codd, [Is your DBMS really relational?](#), Computer World, 1985

Reglas de Codd (Cont.)

(aseguran que un motor de BD es relacional)

- Soportar información faltante mediante valores nulos (NULL)
- Proveer lenguajes para:
 - Definición de datos
 - Manipulación de datos, donde debe haber operaciones de alto nivel para insertar, eliminar, actualizar o buscar.
 - Definición de restricciones de seguridad, restricciones de integridad, autorización y delimitar una transacción.

Edgar Codd, [Is your DBMS really relational?](#), Computer World, 1985

Álgebra relacional

El álgebra relacional es un conjunto de operaciones sobre las relaciones. Cada operación del álgebra relacional toma 1 ó 2 tablas como operandos y produce como resultado una nueva relación.

Se compone de dos grupos de operadores:

Operadores Tradicionales: son los operadores utilizados en álgebra.

Son ellos:

- ∪ Unión,
- ∩ intersección
- diferencia
- y X producto cartesiano.

Operadores Especiales: son operadores orientados al manejo de relaciones.

Los operadores σ (select)

π (project),

\bowtie (join)

y % (división), constituyen el álgebra relacional.

Álgebra relacional

Operador Unión (\cup)

A

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto

B

B#	NomB	Ciudad
B1	José	Miami
B2	Jorge	Orlando
X1	Angel	Toronto

$A \cup B$

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto
B1	José	Miami
B2	Jorge	Orlando

Álgebra relacional

Operador Unión (\cup)

Unión: La unión de dos relaciones compatibles A y B es el conjunto de todas las tuplas que pertenecen a ambas relaciones. Ejemplo: sean las relaciones A y B.

A

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto

B

B#	NomB	Ciudad
B1	José	Miami
B2	Jorge	Orlando
X1	Angel	Toronto

$$A \cup B$$

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto
B1	José	Miami
B2	Jorge	Orlando

Álgebra relacional

Operador Intersección (\cap)

Intersección: La intersección de dos relaciones compatibles en la unión A y B es el conjunto de todas las tuplas que pertenecen tanto a A como a B.

A

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto

B

B#	NomB	Ciudad
B1	José	Miami
B2	Jorge	Orlando
X1	Angel	Toronto

$A \cap B$

A#	NomA	CiudadA
X1	Angel	Toronto

Álgebra relacional

Operador Diferencia (-)

Diferencia: La diferencia entre dos relaciones A y B es el conjunto de las tuplas que pertenecen a A y no pertenecen a B

A

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto

B

B#	NomB	Ciudad
B1	José	Miami
B2	Jorge	Orlando
X1	Angel	Toronto

A - B

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.

Álgebra relacional

Operador Producto Cartesiano(X)

Producto cartesiano : El producto cartesiano extendido de dos relaciones **A** y **B** es el conjunto de las tuplas **t** tales que **t** es la concatenación de una tupla **a** perteneciente a **A** y una tupla **b** perteneciente a **B**. Dicho de otra manera, dada una serie de conjuntos **D1, D2, ..., Dn**; el **producto cartesiano** de estos **n** conjuntos, es el conjunto de las **n** tuplas posibles

A

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto

B

B#	NomB	Ciudad
B1	José	Miami
B2	Jorge	Orlando
X1	Angel	Toronto

A X B

A#	NomA	CiudadA	B#	NomB	Ciudad
A1	Marina	París	B1	José	Miami
A1	Marina	París	B2	Jorge	Orlando
A1	Marina	París	X1	Angel	Toronto
A2	Martin	Londres	B1	José	Miami
A2	Martin	Londres	B2	Jorge	Orlando
A2	Martin	Londres	X1	Angel	Toronto
A3	Graciela	Bs. As.	B1	José	Miami
A3	Graciela	Bs. As.	B2	Jorge	Orlando
A3	Graciela	Bs. As.	X1	Angel	Toronto
X1	Angel	Toronto	B1	José	Miami
X1	Angel	Toronto	B2	Jorge	Orlando
X1	Angel	Toronto	X1	Angel	Toronto

Álgebra relacional

Operador Select (σ)

Construye una nueva tabla al tomar un subconjunto **horizontal** de la tabla existente. Produce un subconjunto horizontal de una relación específica.

El resultado de la selección es otra tabla con los mismos atributos que la tabla original

A

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto

B

B#	NomB	Ciudad
B1	José	Miami
B2	Jorge	Orlando
X1	Angel	Toronto

σ (A)

CiudadA='París'

A#	NomA	CiudadA
A1	Marina	París

σ (B)

NomB='Jorge'

B#	NomB	Ciudad
B2	Jorge	Orlando

Álgebra relacional

Operador Project(π)

Construye una nueva tabla al tomar un subconjunto **vertical** de la tabla existente. Produce un subconjunto vertical de una relación dada, es decir el subconjunto obtenido de seleccionar los atributos especificados.

A

A#	NomA	CiudadA
A1	Marina	París
A2	Martin	Londres
A3	Graciela	Bs. As.
X1	Angel	Toronto

π (A)

NomA, CiudadA

NomA	CiudadA
Marina	París
Martin	Londres
Graciela	Bs. As.
Angel	Toronto

B

B#	NomB	Ciudad
B1	José	Miami
B2	Jorge	Orlando
X1	Angel	Toronto

π (B)

Ciudad

Ciudad
Miami
Orlando
Toronto

Álgebra relacional

Operador División(%)

Sea una relación **A** de grado **m + n** donde **A** puede definirse como un conjunto de pares de valores **< x,y >**. Sea una relación **B** de grado **n**, donde **B** puede definirse como un conjunto de valores **< y >** simples.

Al aplicar el operador división **A % B** el resultado será una relación **C** de grado **m** donde **C** puede definirse como el conjunto de valores **x** tales que el par **<x,y>** aparece en **A** para todos los valores **y** que aparecen en **B**.

Los atributos de la relación resultado, tienen los mismos nombres que los primeros **m** atributos de **A**.

A	Atrib A1	Atrib A2	Atrib A3	...	Atrib Am	Atrib Am+1	...	Atrib Am+n
	x							y
B	Atrib B1	Atrib B2	...	Atrib Bn				
	y							
A % B	Atrib C1	Atrib C2	Atrib C3	...	Atrib Cm			
	x							

Álgebra relacional

Operador División(%)

A

S#	P#
S1	P1
S1	P2
S1	P3
S1	P4
S1	P5
S1	P6
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4
S4	P5

B

P#
P1
P2
P3
P4
P5
P6

A % B

S#
S1

D

P#
P1
P2

A % D

S#
S1
S2

E

P#
P2

A % E

S#
S1
S2
S3
S4

Álgebra relacional

Operador Join(\bowtie)

El resultado de aplicar un JOIN sobre dos tablas es una nueva tabla donde cada renglón se forma concatenando dos renglones que tengan el mismo valor de atributo. Se puede definir un join **mayor que** de la relación **A** sobre el atributo **X** con la relación **B** sobre el atributo **Y** como el conjunto de todas las tuplas **t** tales que, **t** es la concatenación de una tupla **a** tal que **a** pertenece a **A** y una tupla **b** perteneciente a **B** donde **x > y** y **x** es el componente **X** de **A** e **y** es el componente **Y** de **B**

Esta operación es equivalente a tomar el producto cartesiano de las dos relaciones dadas y luego realizar una selección adecuada sobre ese producto. En el caso de que el join se defina de manera tal que la condición **se fundamenta en la igualdad** entre valores de la columna común, la tabla resultante contiene por fuerza dos columnas idénticas.

Una columna se podría eliminar aplicando un project, pero para evitar esta operación se utiliza el **natural join**, operación mediante la cuál una de las columnas idénticas es eliminada

Álgebra relacional

Operador Join(\bowtie)

S: Proveedores

S#	NombreS	Estado	Ciudad
S1	Smith	30	Bs.As.
S2	Jones	10	Rosario
S3	Blake	30	Rosario
S4	Clark	20	Bs. As.
S5	Adams	30	Córdoba

P: Productos

P#	NombreP	Color	Stock	Ciudad
P1	Monitor Led 19"	Rojo	100	Bs.As.
P2	Notebook 14" I5	Verde	70	Rosario
P3	HDD 500 Gb SATA	Azul	200	Córdoba
P4	HDD 1 Tb SATA	Rojo	100	Bs.As.
P5	CD / DVD Externo	Azul	200	Rosario
P6	UltraBook Corei7	Rojo	150	Bs.As.

SP: Relaciones

S#	P#	Cantidad
S1	P1	20
S1	P2	10
S1	P3	40
S2	P3	20
S2	P4	30
S2	P3	50
S2	P6	60
S2	P3	40
S2	P6	80
S2	P5	10
S3	P3	20
S3	P4	50
S4	P6	30
S4	P6	50
S5	P2	20
S5	P2	10
S5	P5	50
S5	P5	10
S5	P6	20
S5	P1	9
S5	P3	9
S5	P4	8
S5	P5	4
S5	P6	5

Obtener una nueva relación con los datos de la relación SP y todos los datos de cada producto de la relación P

$$SP \bowtie P$$

p#

S#	P#	Cantidad	NombreP	Color	Stock	Ciudad
S1	P1	20	Monitor Led 19"	Rojo	100	Bs.As.
S1	P2	10	Notebook 14" I5	Verde	70	Rosario
S1	P3	40	HDD 500 Gb SATA	Azul	200	Córdoba
S2	P3	20	HDD 500 Gb SATA	Azul	200	Córdoba
S2	P4	30	HDD 1 Tb SATA	Rojo	100	Bs.As.
S2	P3	50	HDD 500 Gb SATA	Azul	200	Córdoba
S2	P6	60	UltraBook Corei7	Rojo	150	Bs.As.
.....						

Álgebra relacional

Ejemplos con Operadores Combinados

S: Proveedores

S#	NombreS	Estado	Ciudad
S1	Smith	30	Bs.As.
S2	Jones	10	Rosario
S3	Blake	30	Rosario
S4	Clark	20	Bs. As.
S5	Adams	30	Córdoba

P: Productos

P#	NombreP	Color	Stock	Ciudad
P1	Monitor Led 19"	Rojo	100	Bs.As.
P2	Notebook 14" I5	Verde	70	Rosario
P3	HDD 500 Gb SATA	Azul	200	Córdoba
P4	HDD 1 Tb SATA	Rojo	100	Bs.As.
P5	CD / DVD Externo	Azul	200	Rosario
P6	UltraBook Corei7	Rojo	150	Bs.As.

SP: Relaciones

S#	P#	Cantidad
S1	P1	20
S1	P2	10
S1	P3	40
S2	P3	20
S2	P4	30
S2	P3	50
S2	P6	60
S2	P3	40
S2	P6	80
S2	P5	10
S3	P3	20
S3	P4	50
S4	P6	30
S4	P6	50
S5	P2	20
S5	P2	10
S5	P5	50
S5	P5	10
S5	P6	20
S5	P1	9
S5	P3	9
S5	P4	8
S5	P5	4
S5	P6	5

Obtener una nueva relación con todos los productos de color Rojo, informar solamente el P# y su NombreP.

$$\pi_{P\#, \text{NombreP}} (\sigma_{\text{Color}='Rojo'} (P))$$

P#	NombreP
P1	Monitor Led 19"
P4	HDD 1 Tb SATA
P6	UltraBook Corei7

Álgebra relacional

Ejemplos con Operadores Combinados

S: Proveedores

S#	NombreS	Estado	Ciudad
S1	Smith	30	Bs.As.
S2	Jones	10	Rosario
S3	Blake	30	Rosario
S4	Clark	20	Bs. As.
S5	Adams	30	Córdoba

P: Productos

P#	NombreP	Color	Stock	Ciudad
P1	Monitor Led 19"	Rojo	100	Bs.As.
P2	Notebook 14" I5	Verde	70	Rosario
P3	HDD 500 Gb SATA	Azul	200	Córdoba
P4	HDD 1 Tb SATA	Rojo	100	Bs.As.
P5	CD / DVD Externo	Azul	200	Rosario
P6	UltraBook Corei7	Rojo	150	Bs.As.

SP: Relaciones

S#	P#	Cantidad
S1	P1	20
S1	P2	10
S1	P3	40
S2	P3	20
S2	P4	30
S2	P3	50
S2	P6	60
S2	P3	40
S2	P6	80
S2	P5	10
S3	P3	20
S3	P4	50
S4	P6	30
S4	P6	50
S5	P2	20
S5	P2	10
S5	P5	50
S5	P5	10
S5	P6	20
S5	P1	9
S5	P3	9
S5	P4	8
S5	P5	4
S5	P6	5

Obtener una nueva relación con los datos de la relación SP y el Nombre de cada producto

$$SP \bowtie \pi(P)$$

p# P#,NombreP

S#	P#	Cantidad	NombreP
S1	P1	20	Monitor Led 19"
S1	P2	10	Notebook 14" I5
S1	P3	40	HDD 500 Gb SATA
S2	P3	20	HDD 500 Gb SATA
S2	P4	30	HDD 1 Tb SATA
S2	P3	50	HDD 500 Gb SATA
S2	P6	60	UltraBook Corei7
.....			

Álgebra relacional

Ejemplos con Operadores Combinados

S: Proveedores

S#	NombreS	Estado	Ciudad
S1	Smith	30	Bs.As.
S2	Jones	10	Rosario
S3	Blake	30	Rosario
S4	Clark	20	Bs. As.
S5	Adams	30	Córdoba

P: Productos

P#	NombreP	Color	Stock	Ciudad
P1	Monitor Led 19"	Rojo	100	Bs.As.
P2	Notebook 14" I5	Verde	70	Rosario
P3	HDD 500 Gb SATA	Azul	200	Córdoba
P4	HDD 1 Tb SATA	Rojo	100	Bs.As.
P5	CD / DVD Externo	Azul	200	Rosario
P6	UltraBook Corei7	Rojo	150	Bs.As.

SP: Relaciones

S#	P#	Cantidad
S1	P1	20
S1	P2	10
S1	P3	40
S2	P3	20
S2	P4	30
S2	P3	50
S2	P6	60
S2	P3	40
S2	P6	80
S2	P5	10
S3	P3	20
S3	P4	50
S4	P6	30
S4	P6	50
S5	P2	20
S5	P2	10
S5	P5	50
S5	P5	10
S5	P6	20
S5	P1	9
S5	P3	9
S5	P4	8
S5	P5	4
S5	P6	5

Obtener una nueva relación con los datos de la relación SP y el nombre de cada Producto, se solicita mostrar solo las relaciones que involucren productos de color Verde.

$$SP \bowtie \pi(\sigma(P))$$

p# Color='Verde'
 P#,NombreP

S#	P#	Cantidad	NombreP
S1	P2	10	Notebook 14" I5
S5	P2	20	Notebook 14" I5
S5	P2	10	Notebook 14" I5

Álgebra relacional

Ejemplos con Operadores Combinados

S: Proveedores

S#	NombreS	Estado	Ciudad
S1	Smith	30	Bs.As.
S2	Jones	10	Rosario
S3	Blake	30	Rosario
S4	Clark	20	Bs. As.
S5	Adams	30	Córdoba

P: Productos

P#	NombreP	Color	Stock	Ciudad
P1	Monitor Led 19"	Rojo	100	Bs.As.
P2	Notebook 14" I5	Verde	70	Rosario
P3	HDD 500 Gb SATA	Azul	200	Córdoba
P4	HDD 1 Tb SATA	Rojo	100	Bs.As.
P5	CD / DVD Externo	Azul	200	Rosario
P6	UltraBook Corei7	Rojo	150	Bs.As.

SP: Relaciones

S#	P#	Cantidad
S1	P1	20
S1	P2	10
S1	P3	40
S2	P3	20
S2	P4	30
S2	P3	50
S2	P6	60
S2	P3	40
S2	P6	80
S2	P5	10
S3	P3	20
S3	P4	50
S4	P6	30
S4	P6	50
S5	P2	20
S5	P2	10
S5	P5	50
S5	P5	10
S5	P6	20
S5	P1	9
S5	P3	9
S5	P4	8
S5	P5	4
S5	P6	5

Obtener una nueva relación con los datos de la relación SP y el nombre de cada Producto, se solicita mostrar solo las relaciones que involucren productos de color Verde.

Preguntas

Devuelven lo mismo?

Si

Son ambas performantes?

No

Por qué?

Cant. De tuplas Joineadas.

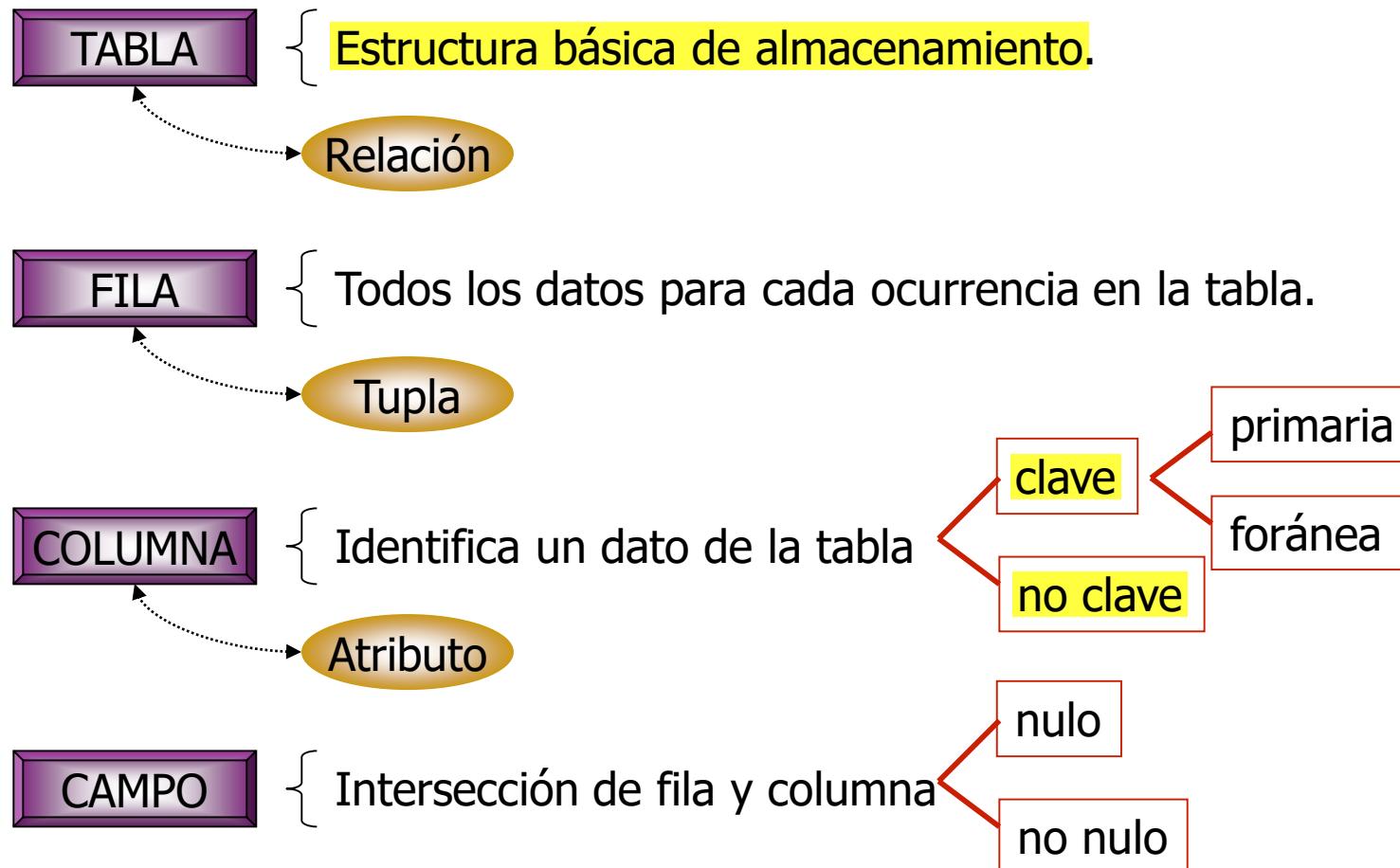
$$SP \bowtie \pi(\sigma(P))$$

p# Color='Verde'
 P#, NombreP

$$\pi(\sigma(SP \bowtie P))$$

Color='Verde' p#
S#, P#, Cantidad, NombreP

Terminología



Definición de Base de Datos

Que es una Base de Datos?

Una BD es un **conjunto de datos persistentes e interrelacionados que es utilizado por los sistemas de aplicación** de una empresa, los mismos se encuentran almacenados en un **conjunto independiente y sin redundancias o con redundancias mínimas**.

Definición de RDBMS

**Que es un Sistema de Administración de Base de Datos?
(DBMS – Data Base Manager System)**

Es un programa que permite administrar los contenidos de una/s base/s de datos almacenada en disco. También llamado Motor de Base de Datos.

El DBMS ofrece a los usuarios una percepción de la base de datos que está en cierto modo por encima del nivel del hardware y que maneja las operaciones del usuario expresadas en el nivel más alto de percepción.

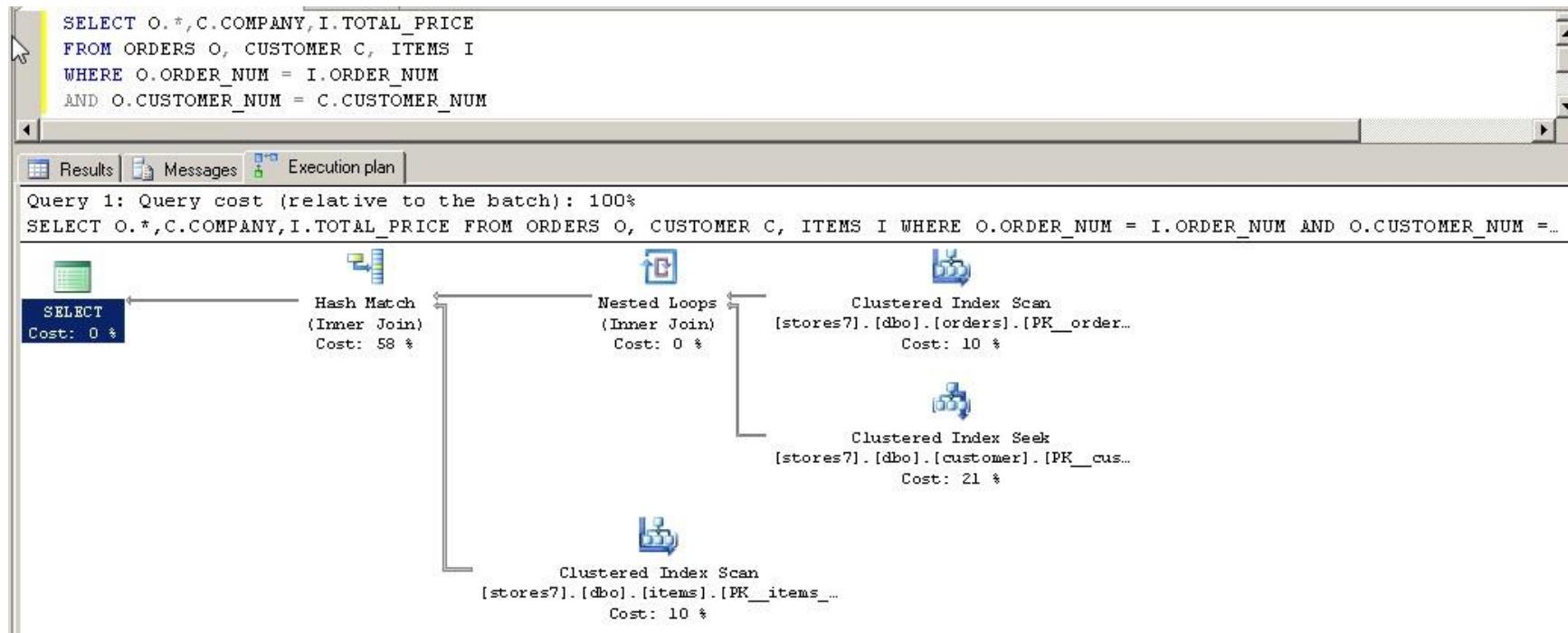
El DBMS también interpreta y ejecuta todos los comandos SQL que le son enviados.

Entre los motores de Base de Datos más utilizados podemos nombrar los siguientes: Oracle, MS SqlServer, MySql, PostgreSQL, DB2, Informix, Sybase, SQLite, entre otros.

Álgebra relacional

Para que nos podría servir el Algebra RelacionalP?

- Para poder entender como un Motor de Base de Datos resuelve una determinada consulta. Y poder inferirlo.



Ejercicios – Algebra Relacional

Ejercicios – Algebra Relacional

S : SUPPLIERS

S#	SNAME	Status	Ciudad
S ₁	Smith	30	London
S ₂	Jones	10	París
S ₃	Blake	30	París
S ₄	Clark	20	London
S ₅	Adams	30	Athenas

P : PARTS

P#	PNAME	Color	Peso	City
P ₁	Nut	Red	12	London
P ₂	Bolt	Green	17	Paris
P ₃	Screw	Blue	17	Rome
P ₄	Screw	Red	14	London
P ₅	Cam	Blue	12	Paris
P ₆	Cog	Red	19	London

J : JOBS

J#	JNAME	Ciudad
J ₁	Sorter	Paris
J ₂	Punch	Rome
J ₃	Reader	Athenas
J ₄	Console	Athenas
J ₅	Collator	London
J ₆	Terminal	Oslo
J ₇	Tape	London

SPJ : Relation

S#	P#	J#	Qty
S ₁	P ₁	J ₁	200
S ₁	P ₁	J ₄	100
S ₂	P ₃	J ₁	400
S ₂	P ₃	J ₂	200
S ₂	P ₃	J ₃	200
S ₂	P ₃	J ₄	500
S ₂	P ₃	J ₅	600
S ₂	P ₃	J ₆	400
S ₂	P ₃	J ₇	800
S ₂	P ₅	J ₂	100
S ₃	P ₃	J ₁	200
S ₃	P ₄	J ₂	500
S ₄	P ₆	J ₃	300
S ₄	P ₆	J ₇	300
S ₅	P ₂	J ₂	200
S ₅	P ₂	J ₄	100
S ₅	P ₅	J ₅	500
S ₅	P ₅	J ₇	100
S ₅	P ₆	J ₂	200
S ₅	P ₁	J ₄	900
S ₅	P ₃	J ₄	900
S ₅	P ₄	J ₄	800
S ₅	P ₅	J ₄	400
S ₅	P ₆	J ₄	500

- 1) Valores S# para proveedores que proveen el proyecto J1.

$$\pi_{S\#} (\sigma_{J\#='J1'} (SPJ))$$

- 2) Valores S# para proveedores que proveen el proyecto J1 c/la parte P1.

$$\pi_{S\#} (\sigma_{J\#= 'J1' \text{ AND } P\#= 'P1'} (SPJ))$$

$$\pi_{S\#} (\sigma_{J\#= 'J1'} (SPJ) \cap \sigma_{P\#= 'P1'} (SPJ))$$

Ejercicios – Algebra Relacional

S : SUPPLIERS

S#	SNAME	Status	Ciudad
S ₁	Smith	30	London
S ₂	Jones	10	París
S ₃	Blake	30	París
S ₄	Clark	20	London
S ₅	Adams	30	Athenas

P : PARTS

P#	PNAME	Color	Peso	City
P ₁	Nut	Red	12	London
P ₂	Bolt	Green	17	Paris
P ₃	Screw	Blue	17	Rome
P ₄	Screw	Red	14	London
P ₅	Cam	Blue	12	Paris
P ₆	Cog	Red	19	London

J : JOBS

J#	JNAME	Ciudad
J ₁	Sorter	Paris
J ₂	Punch	Rome
J ₃	Reader	Athenas
J ₄	Console	Athenas
J ₅	Collator	London
J ₆	Terminal	Oslo
J ₇	Tape	London

SPJ : Relation

S#	P#	J#	Qty
S ₁	P ₁	J ₁	200
S ₁	P ₁	J ₄	100
S ₂	P ₃	J ₁	400
S ₂	P ₃	J ₂	200
S ₂	P ₃	J ₃	200
S ₂	P ₃	J ₄	500
S ₂	P ₃	J ₅	600
S ₂	P ₃	J ₆	400
S ₂	P ₃	J ₇	800
S ₂	P ₅	J ₂	100
S ₃	P ₃	J ₁	200
S ₃	P ₄	J ₂	500
S ₄	P ₆	J ₃	300
S ₄	P ₆	J ₇	300
S ₅	P ₂	J ₂	200
S ₅	P ₂	J ₄	100
S ₅	P ₅	J ₅	500
S ₅	P ₅	J ₇	100
S ₅	P ₆	J ₂	200
S ₅	P ₁	J ₄	900
S ₅	P ₃	J ₄	900
S ₅	P ₄	J ₄	800
S ₅	P ₅	J ₄	400
S ₅	P ₆	J ₄	500

3) Valores JNAME para proyectos suministrados por el proveedor S1.

$$\pi_{\text{JNAME}} (\text{J} \bowtie (\sigma_{\text{P}\#} (\text{SPJ})) \mid \text{P}\# = \text{S}\# = \text{'S1'})$$

4) Valores de Color para partes suministradas por el proveedor S1.

$$\pi_{\text{Color}} (\text{P} \bowtie (\sigma_{\text{S}\#} (\text{SPJ})) \mid \text{P}\# = \text{S}\# = \text{'S1'})$$

Ejercicios – Algebra Relacional

S : SUPPLIERS

S#	SNAME	Status	Ciudad
S ₁	Smith	30	London
S ₂	Jones	10	Paris
S ₃	Blake	30	Paris
S ₄	Clark	20	London
S ₅	Adams	30	Athenas

SPJ : Relation

S#	P#	J#	Qty
S ₁	P ₁	J ₁	200
S ₁	P ₁	J ₄	100
S ₂	P ₃	J ₁	400
S ₂	P ₃	J ₂	200
S ₂	P ₃	J ₃	200
S ₂	P ₃	J ₄	500
S ₂	P ₃	J ₅	600
S ₂	P ₃	J ₆	400
S ₂	P ₃	J ₇	800
S ₂	P ₅	J ₂	100
S ₃	P ₃	J ₁	200
S ₃	P ₄	J ₂	500
S ₄	P ₆	J ₃	300
S ₄	P ₆	J ₇	300
S ₅	P ₂	J ₂	200
S ₅	P ₂	J ₄	100
S ₅	P ₅	J ₅	500
S ₅	P ₅	J ₇	100
S ₅	P ₆	J ₂	200
S ₅	P ₁	J ₄	900
S ₅	P ₃	J ₄	900
S ₅	P ₄	J ₄	800
S ₅	P ₅	J ₄	400
S ₅	P ₆	J ₄	500

P : PARTS

P#	PNAME	Color	Peso	City
P ₁	Nut	Red	12	London
P ₂	Bolt	Green	17	Paris
P ₃	Screw	Blue	17	Rome
P ₄	Screw	Red	14	London
P ₅	Cam	Blue	12	Paris
P ₆	Cog	Red	19	London

J : JOBS

J#	JNAME	Ciudad
J ₁	Sorter	Paris
J ₂	Punch	Rome
J ₃	Reader	Athenas
J ₄	Console	Athenas
J ₅	Collator	London
J ₆	Terminal	Oslo
J ₇	Tape	London

5) Valores S# para proveedores que suministren a proyectos de London o Paris con alguna parte roja.

$$\pi_{S\#} (\sigma_{color='Red'}(P) \bowtie_{P\#} (\sigma_{Ciudad='London' \text{ OR Ciudad='Paris'}}(J))) \bowtie_{J\#} SPJ$$

S#
S ₁
S ₄

$$\pi_{S\#} (\sigma_{Ciudad='London'}(J) \bowtie_{P\#} (\sigma_{color='Red'}(P)) \bowtie_{J\#} SPJ)$$

Ing. Juan Zaffaroni / Ing. Cristina Chahin

Ejercicios – Algebra Relacional

S : SUPPLIERS

S#	SNAME	Status	Ciudad
S ₁	Smith	30	London
S ₂	Jones	10	París
S ₃	Blake	30	París
S ₄	Clark	20	London
S ₅	Adams	30	Athenas

SPJ : Relation

S#	P#	J#	Qty
S ₁	P ₁	J ₁	200
S ₁	P ₁	J ₄	100
S ₂	P ₃	J ₁	400
S ₂	P ₃	J ₂	200
S ₂	P ₃	J ₃	200
S ₂	P ₃	J ₄	500
S ₂	P ₃	J ₅	600
S ₂	P ₃	J ₆	400
S ₂	P ₃	J ₇	800
S ₂	P ₅	J ₂	100
S ₃	P ₃	J ₁	200
S ₃	P ₄	J ₂	500
S ₄	P ₆	J ₃	300
S ₄	P ₆	J ₇	300
S ₅	P ₂	J ₂	200
S ₅	P ₂	J ₄	100
S ₅	P ₅	J ₅	500
S ₅	P ₅	J ₇	100
S ₅	P ₆	J ₂	200
S ₅	P ₁	J ₄	900
S ₅	P ₃	J ₄	900
S ₅	P ₄	J ₄	800
S ₅	P ₅	J ₄	400
S ₅	P ₆	J ₄	500

P : PARTS

P#	PNAME	Color	Peso	City
P ₁	Nut	Red	12	London
P ₂	Bolt	Green	17	Paris
P ₃	Screw	Blue	17	Rome
P ₄	Screw	Red	14	London
P ₅	Cam	Blue	12	Paris
P ₆	Cog	Red	19	London

J : JOBS

J#	JNAME	Ciudad
J ₁	Sorter	Paris
J ₂	Punch	Rome
J ₃	Reader	Athenas
J ₄	Console	Athenas
J ₅	Collator	London
J ₆	Terminal	Oslo
J ₇	Tape	London

6) Valores S# para proveedores que suministren la misma parte a todos los proyectos.

$$\pi_{S\#, P\#, J\#}(\pi_{S\#, P\#, J\#}(SPJ)) \quad \% \quad \pi_{J\#}(J)$$

S#	P#
S ₂	P ₃

$$\pi_{S\#}(\pi_{S\#, P\#, J\#}(SPJ)) \quad \% \quad \pi_{J\#}(J)$$

S#
S ₂