



**Universidad
Europea de Madrid**

LAUREATE INTERNATIONAL UNIVERSITIES

Introducción a las bases de datos

DISEÑO DE BD. MODELO RELACIONAL. CONCEPTOS BÁSICO

Índice

Presentación.....	3
Motivación histórica.....	4
Conceptos de bases de datos relacionales. Definiciones informales.....	6
Conceptos de bases de datos relacionales. Definiciones formales I.....	7
Conceptos de bases de datos relacionales. Definiciones formales II.....	8
Modelos matemáticos de consultas.....	11
Álgebra relacional. Operaciones unarias.....	12
Álgebra relacional. Operaciones de combinación.....	13
Álgebra relacional. Operaciones de conjuntos.....	16
Cálculo relacional de tuplas.....	20
Cálculo relacional de dominios.....	22
Resumen.....	23

Presentación

Una vez realizado el diseño de la BD (consultar tema anterior), se escoge cuál será el SGBD que se utilizará en el proyecto en desarrollo y, utilizando el modelo de datos de éste, se obtiene el esquema conceptual de la base de datos. El modelo relacional es el modelo de datos más extendido en los sistemas comerciales actuales: sistemas como Oracle, MySQL y otros utilizan este modelo de datos.

En este tema se tratará cuál es la estructura de datos básica utilizada por este modelo así como las operaciones que están definidas. Se dará una visión general de los lenguajes matemáticos teóricos que emplea este modelo: álgebra relacional, cálculo relacional de tuplas y cálculo relacional de dominios.



Se tratará con más extensión el lenguaje relacional comercial estándar para los SGBD basados en el modelo relacional: SQL. Este lenguaje se tratará con profundidad en las unidades de aprendizaje 3 (LMD) y 4 (LDD) y será el lenguaje utilizado en las prácticas a lo largo del curso.

Motivación histórica

A continuación se muestra una pequeña introducción histórica de la aparición del modelo relacional y cómo ha ido incorporándose a los SGBD comerciales actuales.

A horizontal timeline bar at the top of the section is divided into three segments labeled 'Años 70', 'Años 80', and 'Años 90'. The 'Años 70' segment is highlighted in dark red, corresponding to the text below.

A principios de los años 70, Edgar Frank Codd, un científico informático inglés que trabajaba para IBM, estableció los fundamentos básicos teóricos del modelo relacional en el artículo titulado "A relational model of data for large shared data banks". Estos conceptos fueron ampliados a lo largo de toda la década de los 70. Se realizaron varios proyectos de sistemas experimentales, entre ellos:

- R de IBM que utilizaba el lenguaje SEQUEL.
- INGRES de la Universidad de California, Berkeley, que utilizaba QUEL como lenguaje del SGBD.
- ADABAS: Universidad Técnica de Darmstadt, Alemania.

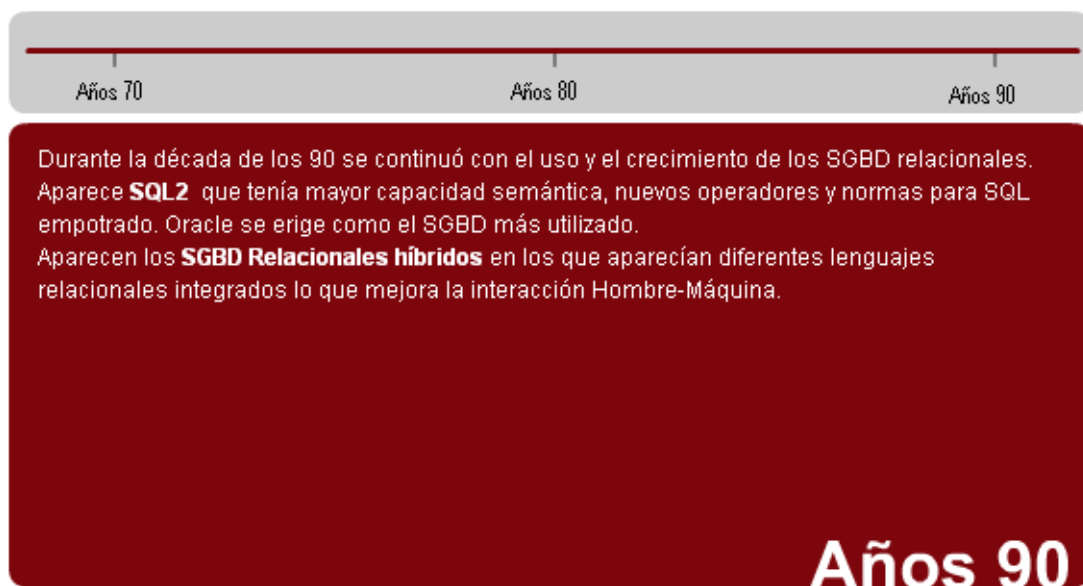
Se desarrollaron los primeros lenguajes relacionales implementados en los sistemas experimentales: SQUARE, SEQUEL (SQL), QBE, QUEL.

Años 70

A horizontal timeline bar at the top of the section is divided into three segments labeled 'Años 70', 'Años 80', and 'Años 90'. The 'Años 80' segment is highlighted in dark red, corresponding to the text below.

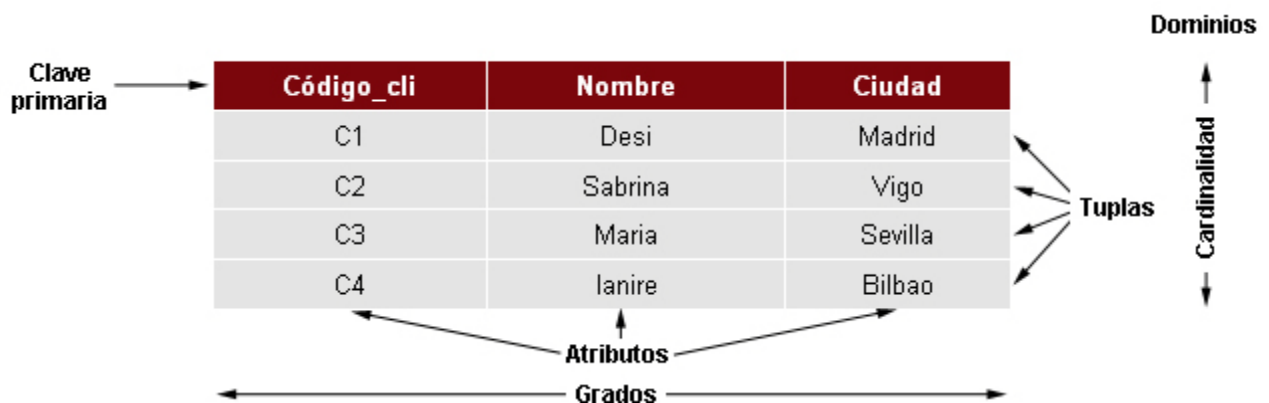
Se definió una versión estándar de SQL, SQL/ANSI, que contenía una intersección de las implementaciones existentes. Aparecieron los siguientes SGBD para sistemas corporativos: DB2, Oracle (1979), Sybase, Informix, INGRES y ADABAS. Estos dos últimos adoptan SQL como lenguaje del SGBD. Aparecieron también sistemas relacionales para sistemas personales: Dbase, Paradox, Access.

Años 80



Conceptos de bases de datos relacionales. Definiciones informales

La estructura de datos básica del modelo relacional es la relación. Una relación es una tabla con filas y columnas. De esta forma una **base de datos relacional** es una colección de relaciones normalizadas (correctamente estructuradas) en la que cada relación tiene un nombre diferente. A continuación se muestra una tabla que representa lo que es una relación en un modelo relacional. Esta figura contiene también las definiciones informales de los términos que maneja el modelo relacional.



- **Atributo:** una columna de una relación.
- **Dominio:** conjunto de valores permitidos para uno o más atributos.
- **Tupla:** una fila de una relación (una fila de una tabla).
- **Cardinalidad de una relación:** número de tuplas que contiene.
- **Grado de una relación:** número de atributos que contiene.
- **Base de datos relacional:** colección de relaciones normalizadas (correctamente estructuradas) en la que cada relación tiene un nombre diferente.
- **Clave primaria:** identificador único de las tuplas de una relación.

Conceptos de bases de datos relacionales. Definiciones formales I

Ahora que conocemos de forma informal los conceptos utilizados en el sistema relacional, vamos a redefinirlos formalmente:

Dados los dominios D_1, D_2, \dots, D_k , se definen el **producto cartesiano de dominios** como el conjunto de todas las k -tuplas (v_1, v_2, \dots, v_k) tales que $v_1 \in D_1, v_2 \in D_2, \dots, v_k \in D_k$ y se nota como $D_1 \times D_2 \times \dots \times D_k$.

Se define una **relación** como un subconjunto del producto cartesiano de una lista de dominios.

El concepto de relación tiene dos significados: el esquema y la instancia de la relación. El **esquema de la relación** está compuesto por los dominios de los atributos que la forman y la **instancia de la relación** por el subconjunto de tuplas concreto del producto cartesiano que la forman.



Se define la **aridad o grado de una relación** como el número de dominios que intervienen en el producto cartesiano de su definición (k).

Esto es equivalente a decir que es el número de componentes de sus tuplas (número de columnas).

Se definen los **atributos** de una relación como los dominios que intervienen en el producto cartesiano de su definición.



[Ejemplo de esquema e instancia de relación](#)

Conceptos de bases de datos relacionales. Definiciones formales II

Se define el **esquema de una relación** como el conjunto de los nombres de los atributos de una relación. El esquema de una relación se nota así: **Nombre_relación (atributos)**

De esta forma el esquema de una base de datos relacional estará compuesto por los esquemas del conjunto de relaciones que las forman. En el ejemplo de la empresa comercial que venimos desarrollando, el esquema relacional será éste.

Esquema relacional

clientes (nif_o_dni, nombre, dirección, cuenta_bancaria, num_pedido)

pedidos (num_pedido, fecha, importe_bruto, iva, importe_netto)

artículos (num_referencia, descripción, precio_venta, num_unid)

contiene (num_pedido, num_referencia, cantidad)

proveedores (nif, dirección, teléfono)

suministra (nif, num_referencia, precio_coste)

Se denomina **relación finita** a aquella obtenida a partir de dominios finitos.

Se define **tupla** como los elementos de una relación y se nota: $\mu = [234, Cervera, Alcala]$, $\mu \in clientes$

Se puede acceder a una componente de una tupla de dos formas:

- Por el nombre del atributo. $\mu(DNI) = 234$
- Por el orden de la componente. $\mu(1) = 145$, $\mu(2) = Cervera$

El acceso a varios componentes de una tupla se realiza:

- $\mu[DNI, DIR] = (234, Alcalá)$
- $\mu[1, 3] = (234, Alcalá)$

Se denomina **clave de una relación** (definición análoga a conjunto de entidades) al conjunto de atributos S de una relación R que cumple 2 condiciones básicas:

1. R no tiene dos tuplas que coincidan en todos los valores de S. Es decir, así.
2. Ningún subconjunto propio de S tiene la propiedad 1.

así

$$S = clave(R)$$

$$\mu_1, \mu_2 \in R$$

$$\mu_1 \neq \mu_2 \Rightarrow \mu_1[k] \neq \mu_2[k]$$



Ejemplo

Relación clientes



Ejemplo

Relaciones



Ejemplo

Clave de una relación

Relación clientes

Dada la relación clientes (dni, nombre, dirección) donde el dominio dni está compuesto por 8 dígitos del 1 al 10 y los dominios nombre y dirección por 20 caracteres de la A a la Z podemos establecer:

Cientes \subset dni x nombre x dirección

El número de tuplas de este producto cartesiano será: $10 \cdot 27^{20}$

Relaciones

Ejemplos de relaciones

A	N
0	b
0	b
1	c

Ejemplo1:

$$R = \{(0,b), (0,c), (1,c)\}$$

Ejemplo 2

Clientes(DNI,NOMBRE,DIR)

CLIENTES = { (234, 'Sanz', 'Alcala'), (456, 'Garcia', 'Serrano'),
(897, 'Fernandez', 'Princesa') }

DNI	NOMBRE	DIR
234	Sanz	Alacala
456	Garcia	Serrano
897	Fernandez	Princesa

Clave de una relación

Ejemplo 1

- CLIENTES(DNI,NOMBRE,DIR)
- $\{DNI\} = clave(CLIENTES)$
- $cl_1, cl_2 \in CLIENTES$
- $cl_1 \neq cl_2 \Rightarrow cl_1[DNI] \neq cl_2[DNI]$

Ejemplo 2

- $s = \{DNI, NOMBRE\} = clave(CLIENTES)$
- No, porque el subconjunto $\{DNI\}$ cumple la 1ª propiedad

Modelos matemáticos de consultas

Existen tres modelos matemáticos para la construcción de sentencias que permiten recuperar información de la base de datos. A partir de ahora llamaremos a estas sentencias consultas.

- **Álgebra relacional (AR)**: es un modelo matemático algebraico.
- **Cálculo relacional de tuplas (CRT)**: es un modelo matemático lógico.
- **Cálculo relacional de dominios (CRD)**: es un modelo matemático lógico.

Los **lenguajes relacionales comerciales** utilizados por los SGBD proponen un soporte para implementar la estructura de datos fundamental, la relación; e implementan operaciones de acceso a los datos, tomadas del AR, CRT o CRD. Así, **ISBL** está basado en AR; **QUEL** está basado en CRT, **QBE** es un Lenguaje visual basado en CRD y **SQL** es una mezcla de AR, CRT y CRD, con sintaxis inspirada en el lenguaje natural.

En el tema 2 ya vimos que una de las propiedades que tiene que tener el lenguaje del SGBD es la expresividad. De esta forma, el álgebra relacional es más expresivo que los lenguajes relacionales comerciales como sql y éste, a su vez, más expresivo que los lenguajes de programación de propósito general.

Ejemplo:

Si quisiéramos recuperar el nombre del cliente con dni 234 de nuestra base de datos de la empresa comercial, tendríamos que realizar las siguientes consultas:

- **En Álgebra relacional:** $\pi_{nombre, direccion}(\sigma_{dni=234}(clientes))$
- **En SQL:** Select nombre from cliente where dni=234
- **En Pascal:** Necesitaríamos conocer con detalle las estructuras de datos físicas utilizadas para almacenar los datos puesto que el código dependerá de la estructura utilizada. De esta forma el desarrollo es costoso y de difícil modificación y optimización posterior.

Álgebra relacional. Operaciones unarias

El **álgebra relacional** es un lenguaje teórico con operaciones aplicables a una o más relaciones. El resultado obtenido es otra relación. No se modifican, por tanto, las relaciones originales. Permite la construcción de expresiones que representan las principales operaciones que se pueden hacer sobre una BD Relacional.

Las operaciones del Álgebra relacional pueden ser:

- **Unarias:** cuando se aplican a una sola relación. En este tipo de operaciones se encuentran la selección y la proyección.
- **Entre conjuntos:** aquellas que están basadas en las operaciones de conjuntos. Son: unión, diferencia, intersección.
- **De combinación:** producto cartesiano y producto natural.

Selección: esta operación unaria selecciona de la relación R a la que se aplica aquellas tuplas que cumplen una determinada condición F, donde F es una expresión lógica que puede incluir operadores de comparación (<, >, <=, >=, <>, etc.) y operadores lógicos (\wedge , \vee). Se nota por: $\sigma_F(R)$

Se puede decir que esta operación selecciona filas de la tabla de la relación a la que se aplica: **selección horizontal**.

Proyección: esta operación selecciona columnas de la tabla que representa la relación a la que se aplica. Realiza **selección vertical**.

Si R es una relación de aridad k, la proyección sobre las componentes i_1, i_2, \dots, i_m ($m \leq k$) se representa: $\pi_{i_1, i_2, \dots, i_m}(R)$, dando como resultado otra relación que representa el conjunto de m-tuplas (a_1, a_2, \dots, a_m) , tales que hay una k-tupla en R, (b_1, b_2, \dots, b_k) , para la que $a_j = b_{i_j}$ para $j = 1, 2, \dots, m$. (Elimina duplicados).



Formulación de consultas mediante expresiones algebraicas

Álgebra relacional. Operaciones de combinación

Dentro del grupo de las operaciones de combinación se encuentran el producto cartesiano y el producto natural.

Producto cartesiano

Sean las relaciones R, con Tr tuplas y Ar atributos, y S, con Ts tuplas y As atributos, el **producto cartesiano de R x S** es el conjunto de todas las posibles (Tr*Ts) tuplas, cuyas primeras Tr componentes forman una tupla de R, y sus Ts últimas forman una tupla en S. La aridad (grado) de la tabla resultado será (Ar + As).

En otras palabras, es la relación formada por todos los posibles pares de tuplas concatenando una de cada relación.

Producto natural (natural join)

Dadas dos relaciones R y S, el producto natural es el subconjunto de tuplas del producto cartesiano en las que coinciden los valores de los atributos comunes a ambas, es decir, que aparecen tanto en R como en S. Se nota: $R \bowtie S$

Para el cálculo de $R \bowtie S$:

- Calcular $R \times S$
- Para cada atributo A, común a R y S, seleccionar de $R \times S$ aquellas tuplas que cumplan $R.A = S.A$.
- Para cada atributo de los anteriores, eliminar los componentes S.A y conservar R.A.

Formalmente:

Si A1, A2, ..., Ak son los atributos comunes a R y S:

$$R \bowtie S = \pi_{i_1, i_2, \dots, i_m} \sigma_{R.A_1 = S.A_1 \wedge R.A_2 = S.A_2 \wedge \dots \wedge R.A_k = S.A_k} (R \times S)$$

siendo i1, i2, ..., im la lista de componentes de RxS en orden, exceptuando los componentes S.A1, S.A2, ..., S.Ak


[Producto cartesiano](#)

[Producto natural](#)

[Consultas](#)

Producto Cartesiano

R:

A	B	C
a	b	c
d	a	f
c	b	d

S:

A	B	C
b	g	a
d	a	f

R x S =

A	B	C	D	E	F
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f

Producto Natural

R:

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

S:

B	C	D
b	c	d
b	c	e
a	d	b

R |X| S:

A	B	C	D
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b

Álgebra relacional. Operaciones de conjuntos

Dentro de las operaciones de conjuntos del álgebra relacional se encuentran la unión, la intersección y la diferencia. Los significados de estas operaciones son equivalentes a los significados que tienen en la teoría de conjuntos.

Unión

Dadas dos relaciones R y S, la unión de ambas, $R \cup S$, se define como la relación que contiene todas las tuplas de R, de S o de ambas, eliminando las tuplas duplicadas. Sólo puede aplicarse sobre relaciones compatibles con respecto a la unión, es decir, si sus esquemas:

- Se corresponden
- Tienen el mismo número de atributos
- Cada pareja de atributos correspondientes tiene el mismo dominio

Diferencia

Define una relación formada por las tuplas que se encuentran en R pero no en S. R y S deben ser compatibles con respecto a la unión.

Intersección

Define una relación formada por las tuplas que se encuentran tanto en R como en S. R y S deben ser compatibles con respecto a la unión.



Ejemplo

Unión

Ejemplo

Diferencia

Ejemplo

Intersección

Documentos

Operaciones de conjuntos

Unión

R:

A	B	C
a	b	c
d	a	f
c	b	d

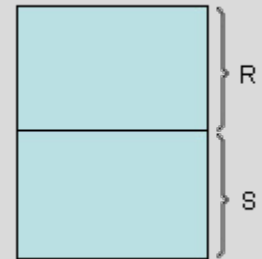
S:

B	C	D
b	g	a
d	a	f

 $R \cup S =$

A	B	C
a	b	c
d	a	f
c	b	d
b	g	a

RUS



Diferencia

R:

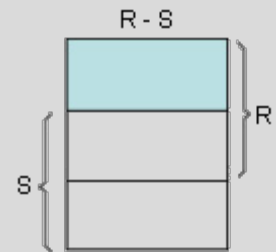
A	B	C
a	b	c
d	a	f
c	b	d

S:

B	C	D
b	g	a
d	a	f

R-S=

A	B	C
a	b	c
c	b	d



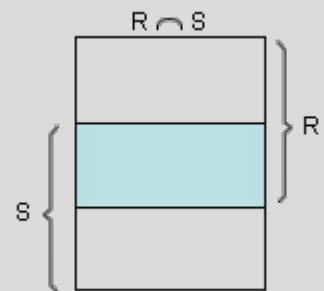
Intersección

R:

A	B	C
a	b	c
d	a	f
c	b	d

S:

B	C	D
b	g	a
d	a	f



$R \cap S =$

A	B	C
d	a	f

Cálculo relacional de tuplas

El cálculo relacional de tuplas es un modelo matemático basado en la lógica de predicados. En este caso, para establecer la condición de las tuplas de una relación que hay que seleccionar como resultado de una consulta, utiliza variables que almacenan tuplas (μ). Esta variable recorre secuencialmente la tabla dentro de la relación donde se aplica, compara en cada tupla si se cumple o no la condición de partida y, en ese caso, incluye la tupla en la relación resultado.

Veamos una consulta ejemplo:

DNI	NOMBRE	DIR
234	Sanz	Alcalá
456	García	Serrano
897	Fernandez	Princesa

Dada la tabla de clientes (dni, nombre, dir) con los datos que se muestran en la figura; si quisiéramos recuperar el dni, el nombre y la dir de los clientes con DNI=438 o DNI=234, tendríamos que utilizar la siguiente consulta:

$$\{ \nu | \text{clientes}(\nu) \wedge (\nu[\text{DNI}] = 438 \vee \nu[\text{DNI}] = 324) \}$$

Donde:

ν es la variable tupla que utilizamos.

ν

$\text{Clientes}(\nu)$ es la primera condición que establecemos para la selección e indica que la variable ν recorrerá la tabla de clientes.

$(\nu[\text{DNI}] = 438 \vee \nu[\text{DNI}] = 324)$ es la segunda condición que imponemos: que el valor de ν para la columna dni sea 438 ó que sea 324.



Cálculo relacional de tuplas

Ejemplo

Cálculo relacional de tuplas

DNI's de los clientes que tienen alguna cuenta con saldo superior a 1000€

$$\{\mu^1 | (\exists r)(\exists v)(tiene(r) \wedge cuentas(v) \wedge v[SALDO] > 1000 \wedge r[CCC] = v[CCC] \wedge r[DNI] = \mu^{(1)}[DNI])\}$$

DNI	CCC
456	12
456	8
897	8
234	10
897	35
CCC	SALDO
12	1200,7
35	800,4
10	7399,9
8	201,2

Cálculo relacional de dominios

El cálculo relacional de dominios es un modelo matemático para la definición de consultas basado en la lógica de predicados de primer orden. Es muy similar al cálculo relacional de tuplas pero en este caso se consideran variables de dominio (que toman sus valores en el dominio del atributo) en lugar de variables de tupla.

Veamos una consulta ejemplo: dada la tabla de clientes utilizada en el apartado anterior. La misma consulta en cálculo relacional de dominios sería:

DNI, NOM y DIR de los clientes con DNI=438 o DNI = 324.

$$\{ DNI, NOM, DIR \mid clientes(DNI, NOM, DIR) \wedge (DNI = 438 \vee DNI = 324) \}$$

Con el siguiente significado:

- *Dni, nom, dir*: vamos a utilizar tres variables de dominio.
- *clientes(DNI, NOM, DIR)*: Las tres variables anteriores almacenarán los valores de tuplas de la tabla clientes. Dni almacenará los valores de la primera columna, nom los de la segunda y dir los de la tercera.
- $(DNI = 438 \vee DNI = 324)$, la condición para seleccionar los valores de esta tupla como resultado de la consulta es que dni tenga como valor 438 o 324.



Resumen

Los SGBD están basados en un modelo de datos particular. El modelo de datos más utilizado en la actualidad es el **modelo relacional**.

El modelo relacional define como estructura de datos la relación. Una relación está compuesta por atributos y tuplas. Los atributos tienen asociado un dominio y algunos de ellos formarán la clave. Una relación se caracteriza por la aridad ó grado que tiene.

El modelo relacional utiliza para definir las operaciones que se pueden realizar modelos matemáticos teóricos. Éstos son el álgebra relacional (basada en el álgebra matemática), el cálculo relacional de tuplas y el cálculo relacional de dominios, estos últimos basados en lógica de predicados.

El álgebra relacional utiliza operaciones:

- Unarias: selección y proyección
- De combinación: producto cartesiano y producto natural
- De conjuntos: unión, intersección y diferencia.

