

Modelo Relacional

Especificación de una base de datos relacional utilizando una definición formal

Definiciones

- Introducido en 1970
- Base teórica y práctica en estudios, investigaciones e implementaciones de sistemas de bases de datos
- Descripciones formales de:
 - Relaciones
 - Esquemas
 - Instancias
 - Base de datos relacional

ESQUEMAS DE RELACION

Esquema de relación primitivo

$$PRS = (ATR, DMN, dom)$$

- *ATR* es un conjunto finito de *atributos*. Los atributos serán los cabecales de las columnas “de la tabla”.
- *DMN* es un conjunto finito de *dominios*. Cada dominio es un conjunto de valores los cuales pueden ser infinitos.
- *dom: ATR-->DMN* es una función que asocia a cada atributo un dominio. Para cada atributo solamente los valores del correspondiente dominio pueden aparecer en la columna que es encabezada por ese atributo.

Esquema de relación

$$RS = (PRS, M, SC)$$

- PRS es un esquema de relación primitivo
- M es el significado de la relación
- SC es el conjunto de restricciones o condiciones de la relación.

Relación:

$$RS = (ATR, DMN, dom, M, SC)$$

Esquemas de relación compatibles

■ Dados dos esquemas de relación

$RS1 = (ATR1, DMN1, dom1, M1, SC1)$ y

$RS2 = (ATR2, DMN2, dom2, M2, SC2)$

■ Se dice que son *compatibles* si se cumple que

■ $ATR1 = ATR2$

■ $DMN1 = DMN2$

■ $dom1 = dom2$

Ejemplo

Considérese un hotel que desea guardar información sobre sus habitaciones.

De cada habitación interesa:

- Su número.
- El número de camas que posee.
- Si tiene o no baño privado.
- En qué piso del hotel se sitúa.
- Su precio.

Ejemplo

Un esquema de relación que puede ser usado para este propósito es ***HABITACIONES = (ATR, DMN, dom, M, SC)*** donde:

- ***ATR*** = {*NUM-HABITACION, CAMAS, BAÑO?, PISO, PRECIO*};
- ***DMN*** = {*conjunto de números de habitaciones, conjunto de enteros positivos, conjunto {true, false}, conjunto de números de pisos, conjunto de enteros positivos*}
- ***dom*** es claro.

Ejemplo

■ El significado ***M*** de esta relación es que la misma describe información sobre las habitaciones del hotel. Para cada habitación su número de habitación es dado, junto con su número de camas, si posee o no baño privado, en qué piso del hotel se sitúa y su precio por día.

- **SC** puede incluir diferentes condiciones tales como:
- Cada habitación posee un número diferente
 - Existen solamente 8 pisos y el primer dígito del número de pieza indica el piso.
 - Toda habitación del piso 2 tiene baño privado.
 - Una habitación con baño privado cuesta más de U\$ 20.
 - Ningún piso tiene más de 20 habitaciones.

Ejemplo: Restricciones

- ***Cada habitación tiene un número único de habitación.***

f1(prs) = true

iff

forall t1,t2 in prs holds

if t1[NUM-HABITACION] = t2[NUM-HABITACION]

then t1 = t2

- ***Cada habitación del piso 2 tiene baño.***

f3(prs) = true

iff

forall t in prs holds

if t[PISO] = 2

then t[BANO?] = true

- ***Una habitación con baño privado cuesta más de U\$ 25.***

f4(prs) = true

iff

forall t in prs holds

if t[BANO?] = true

then t[PRECIO] > 25

Instancias de relación

- El **esquema** de relación permite especificar la estructura de la relación.
- La información real almacenada en la relación se describe mediante **instancias** de relación.
- Una instancia de relación puede verse como una **tabla** en la que el orden de los renglones no es importante.
- Cada renglón tiene varias entradas (dato), **una por cada atributo** en el esquema y el valor almacenado en cada entrada debe pertenecer a un dominio adecuado.
- Un renglón de la tabla se denomina **tupla**

Tupla

■ Sea el esquema primitivo de relación: $PRS = (ATR, DMN, dom)$

Una **tupla** sobre el esquema primitivo de relación PRS es una función $t: ATR \rightarrow \prod_{\delta \in DMN} U_{\delta}$

que asocia a cada atributo con su correspondiente valor tal que:
Para todo atributo $A \in ATR$, $t(A) \in dom(A)$.

■ Una **instancia de relación posible** del esquema de relación primitivo PRS es un **conjunto de tuplas sobre PRS** .

Como notación usaremos $t[A]$ para expresar el valor correspondiente al atributo A en una tupla t .

Restricción de relación – Instancia de relación

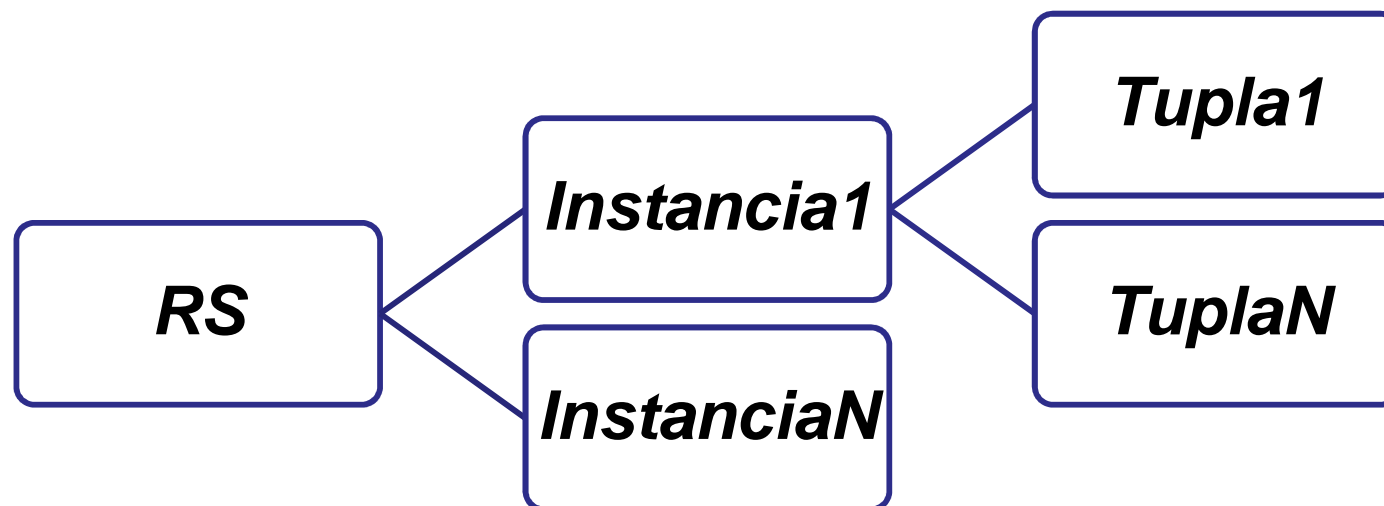
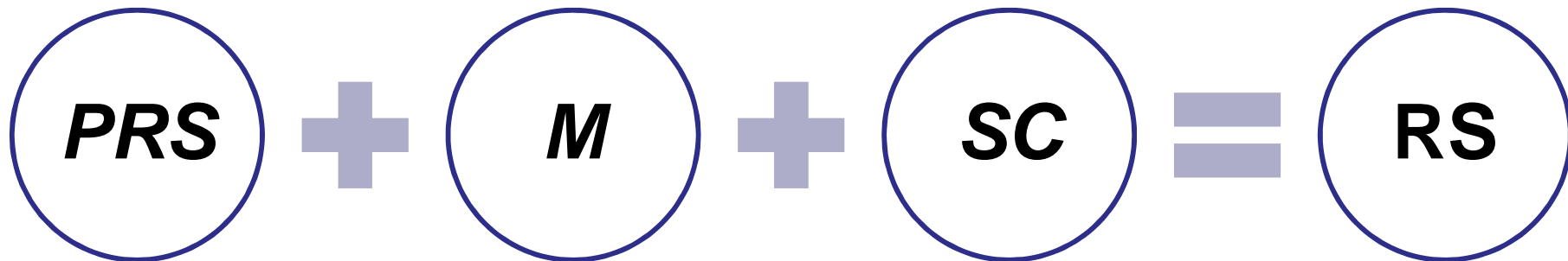
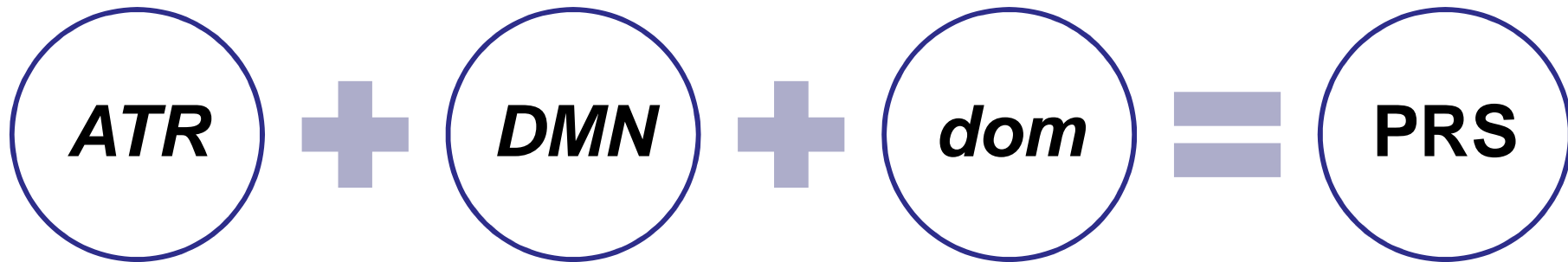
■ Sea el esquema de relación: $RS = (PRS, M, SC)$

Una ***restricción de relación*** del esquema de relación RS se representa por una **función booleana que asocia con cada instancia de relación posible de PRS , el valor *true* o *false*.**

■ Si esa función asocia el valor *true*, decimos que la instancia de relación posible *satisface a la restricción de relación*.

■ Una ***instancia de relación*** del esquema de relación RS es una instancia de relación posible de PRS , que satisface todas las restricciones de relación de SC .

Resumen 1



Esquema primitivo de una Base de datos

$$PDS = \{ RSi = (ATRi, DMNi, domi, Mi, SCi) \mid i \in I \}$$

donde RSi son esquemas de relación tales que para todo $i, j \in I$, y $A \in ATR$,
 $A \in ATRi \cap ATRj$ implica que $domi(A) = domj(A)$.

Todas las funciones $domi$ pueden ser extendidas a una función $dom: \bigcup_{i \in I} ATRi \rightarrow DMNi$.

Esquema de la base de datos

$$DS = (PDS, DM, SDC)$$

Siendo:

- **PDS** un esquema primitivo de la base de datos.
- **DM** el significado de la base de datos.
- **SDC** el conjunto de las restricciones de la base de datos.

Instancia posible de una base de datos

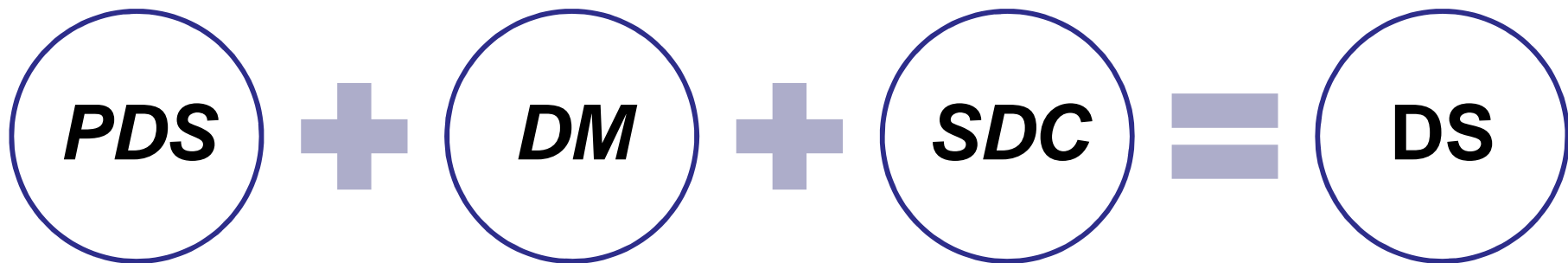
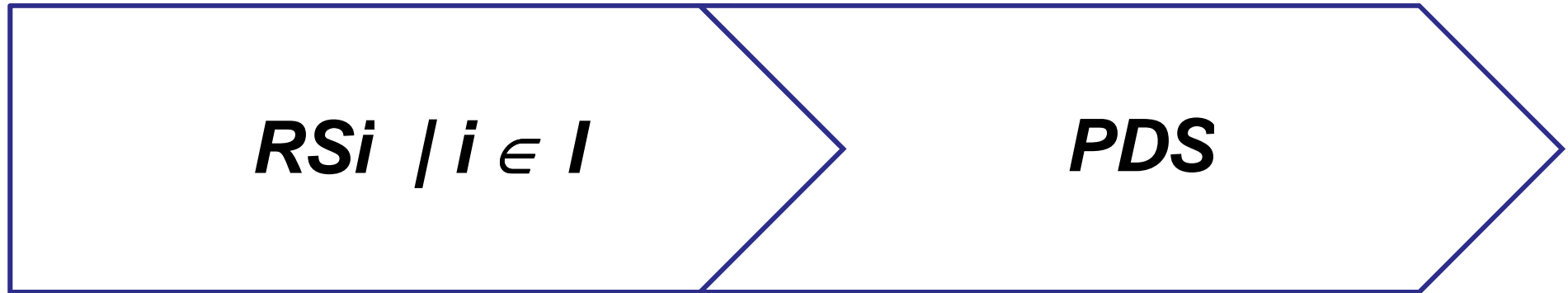
Una *instancia posible de una base de datos* es el conjunto *pds* de instancias de relaciones tal que hay exactamente una instancia de relación en *pds* por cada esquema en el esquema primitivo de la base de datos (PDS).

Restricciones de la base de datos

Las restricciones de la base de datos son representadas por funciones booleanas que asocian *true* o *false* a cada posible instancia de la base de datos.

Si asocia *true* decimos que la instancia de la base de datos satisface las restricciones.

Resumen 2



ESQUEMAS DINAMICOS

Esquema de relación dinámico

Par ordenado $DYRS = (RS, SDYC)$

Donde:

- **RS** es un esquema de relación
- **SDYC** es el conjunto de las restricciones dinámicas de la relación.

Cada una de estas restricciones está representada por una función que asocia *true* o *false* con cada secuencia de instancias de la relación RS.

Decimos que la secuencia de instancias de la relación *satisface una restricción dinámica de la relación* si la función que la representa asocia *true* para esa secuencia.

Evolución

La *evolución de un esquema dinámico de **relación*** es una secuencia rs_0, rs_1, rs_2, \dots de instancias de la relación para las cuales cada restricción dinámica de la relación en SDYC es satisfecha.

Esquema dinámico de una base datos

Par ordenado $DYDS = (DS, SDYDC)$

Donde:

- DS es el esquema de la base de datos
- $SDYDC$ es el conjunto de las restricciones dinámicas de la base de datos.

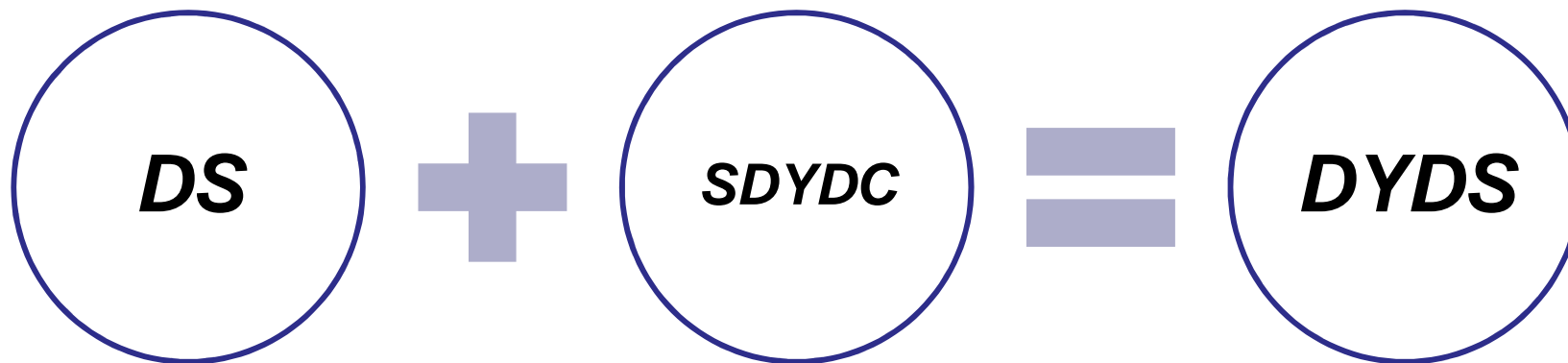
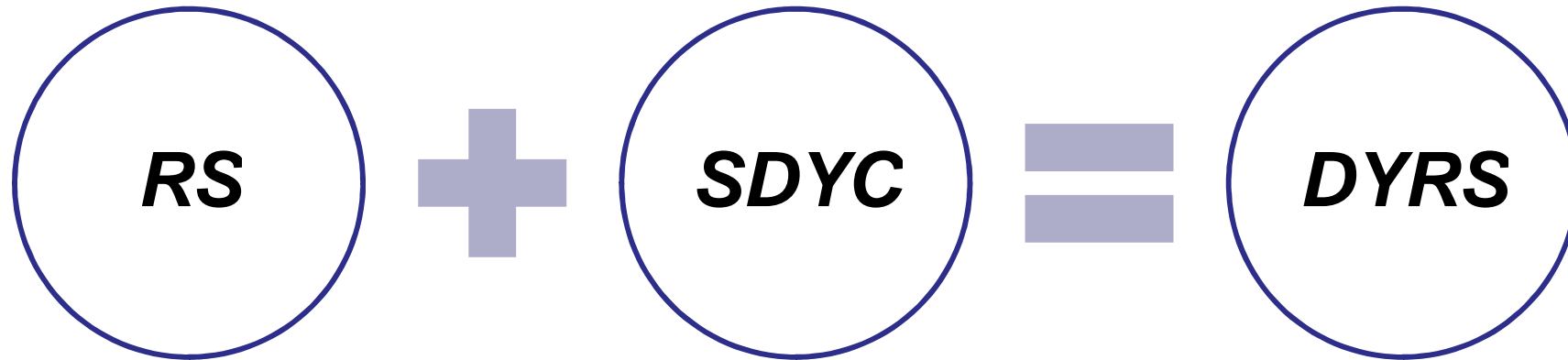
Cada una de estas restricciones está representada por una función que asocia *true* o *false* con cada secuencia de instancias de la base de datos DS .

Decimos que la secuencia de instancias de la base de datos satisface una restricción dinámica de la base de datos si la función que la representa retorna *true* para esa secuencia.

Evolución de un esquema dinámico de la base de datos

La evolución de un esquema dinámico de la base de datos es una secuencia ds_0, ds_1, ds_2, \dots de instancias de la base de datos para las cuales cada restricción dinámica de la base de datos en $SDYDC$ es satisfecha.

Resumen 3



Ejemplo

Vamos a suponer que no es posible remover un baño de una habitación del hotel.

El esquema dinámico de relación es:

$$DHABIT = (HABITACIONES, SDYC)$$

donde *SDYC* contiene por ejemplo la restricción dinámica que dice que ningún baño puede ser removido de una habitación. Esta restricción es representada por una función booleana ***NoRemover***

La función ***NoRemover: DS x DS --> {true,false}***
Se aplica a **dos** instancias de la base de datos ***ds1*** y ***ds2***.

Esta función retorna *true* cuando para cada tupla de *HABITACIONES* en la instancia *ds1*, que tenga baño, la tupla correspondiente (si existe) en la instancia *ds2* también tiene baño.

Clasificación de restricciones

- Restricciones dinámicas de la base de datos - **SDYDC**
- Restricciones dinámicas de la relación - **SDYC**
- Restricciones de la base de datos - **SDC**
- Restricciones de la relación - **SC**
- Restricciones de la tupla

En forma genérica, vamos a denominar a todas estas restricciones como ***Restricciones de integridad.***

Clasificación de restricciones

Las restricciones de **relación** determinan cuáles conjuntos de tuplas son instancias válidas de la relación. Para determinar esto debemos ver la relación en su totalidad.

Algunas veces es posible verificar tupla por tupla, en este caso decimos que es una *restricción de **tupla***.

Clasificación de restricciones

Toda restricción de tupla es una restricción de relación.

Toda restricción de relación de un esquema relación definido en un esquema de base de datos, es equivalente a una restricción de base de datos sobre dicho esquema de base de datos.

La restricción de relación sólo especifica las instancias de relación de la relación dada, y no dice nada acerca de las instancias de relación de las demás relaciones de la base de datos.

Resumen de restricciones

- Las restricciones de tupla son equivalentes a un subconjunto de las restricciones de relación.
- Las restricciones de relación son equivalentes a un subconjunto de las restricciones dinámicas de la relación y a otro subconjunto de las restricciones de la base de datos.
- Las restricciones de la base de datos son equivalentes a un subconjunto de las restricciones dinámicas de la base de datos.

NOCION MATEMATICA DEL MODELO RELACIONAL

- El modelo de datos relacional está basado en el concepto matemático de relación utilizado en la teoría de conjuntos.
- En esta teoría, una ***relación*** es un subconjunto de un **producto cartesiano de una lista de *dominios***.
- Un *dominio* es simplemente un conjunto de valores.
- El producto cartesiano de los dominios $D1, D2, \dots, Dk$ (que se escribe como $D1 \times D2 \times \dots \times Dk$), es el conjunto de todas las *k-tuplas* $\langle v1, v2, \dots, vk \rangle$ tales que $vi \in Di$, para todo $i=1\dots k$. El valor de k debe ser mayor que 0.

NOCION MATEMATICA DEL MODELO RELACIONAL

- Una *relación* (en sentido matemático) es cualquier subconjunto no estricto del producto cartesiano de uno o más dominios.
- En lo que respecta a las bases de datos, asumiremos que una relación es *finita*. Es decir, una instancia de una relación tiene una cantidad finita de tuplas.
- Cada relación subconjunto del producto cartesiano $D1 \times D2 \times \dots \times Dk$ se dice que tiene *aridad* o *grado* k . Una tupla $\langle v1, v2, \dots, vk \rangle$ tiene k componentes; la i -ésima componente es v_i .

NOCION MATEMATICA DEL MODELO RELACIONAL

- Podemos ver a una relación como una ***tabla*** donde **cada renglón es una tupla y cada columna se corresponde a una componente.**
- Se le dan nombres a las columnas, a los que se llaman ***atributos***. Si colocamos nombres de atributo en las columnas de una relación, entonces el orden de las columnas carece de importancia.
- En términos matemáticos vemos a las tuplas como **asociaciones entre los nombres de atributos y valores de los dominios de los atributos.**