



Universidad de Granada

decsai.ugr.es

Fundamentos de Bases de Datos

Grado en Ingeniería Informática

Tema 4: El modelo relacional



DECSAI

**Departamento de Ciencias de la
Computación e Inteligencia Artificial**

- 1. La estructura de datos relacional**
- 2. Definiciones iniciales**
- 3. Propiedades de la estructura de datos relacional**
- 4. Notación**
- 5. Restricciones o reglas de integridad**



- 1. La estructura de datos relacional**
2. Definiciones iniciales
3. Propiedades de la estructura de datos relacional
4. Notación
5. Restricciones o reglas de integridad



Introducido por E.F. Codd en 1970.



El modelo relacional abarca tres ámbitos distintos de los datos:

1. **Las estructuras para almacenarlos:** El usuario percibe la información de la base de datos estructurada en **tablas**.
2. **La integridad:** Las **tablas** deben **satisfacer** ciertas **condiciones** que **preservan la integridad y la coherencia** de la **información** que contienen.
3. **Consulta y manipulación:** Los **operadores** empleados por el modelo se aplican **sobre tablas y devuelven tablas**.

La tabla es la estructura lógica de un sistema relacional. A nivel físico, el sistema es libre de almacenar los datos en el formato más adecuado (archivo secuencial, archivo indexado, listas con apuntadores,...).

1. La estructura de datos relacional
2. **Definiciones iniciales**
3. Propiedades de la estructura de datos relacional
4. Notación
5. Restricciones o reglas de integridad



Atributo: Cualquier elemento de información susceptible de tomar valores. Notación: A_i , $i=1,2,\dots$

Dominio: Rango de valores donde toma sus datos un atributo. Se considera finito. Notación: D_i , $i=1,2,\dots$

Relación: Dados los atributos A_i , $i=1,2,\dots,n$ con dominios D_i , $i=1,2,\dots,n$, no necesariamente distintos, definimos relación asociada a $A_1..A_n$, y lo notaremos por $R(A_1..A_n)$, a **cualquier subconjunto del producto cartesiano $D_1 \times D_2 \times \dots \times D_n$.**

Tupla: Cada una de las **filas** de una **relación**.

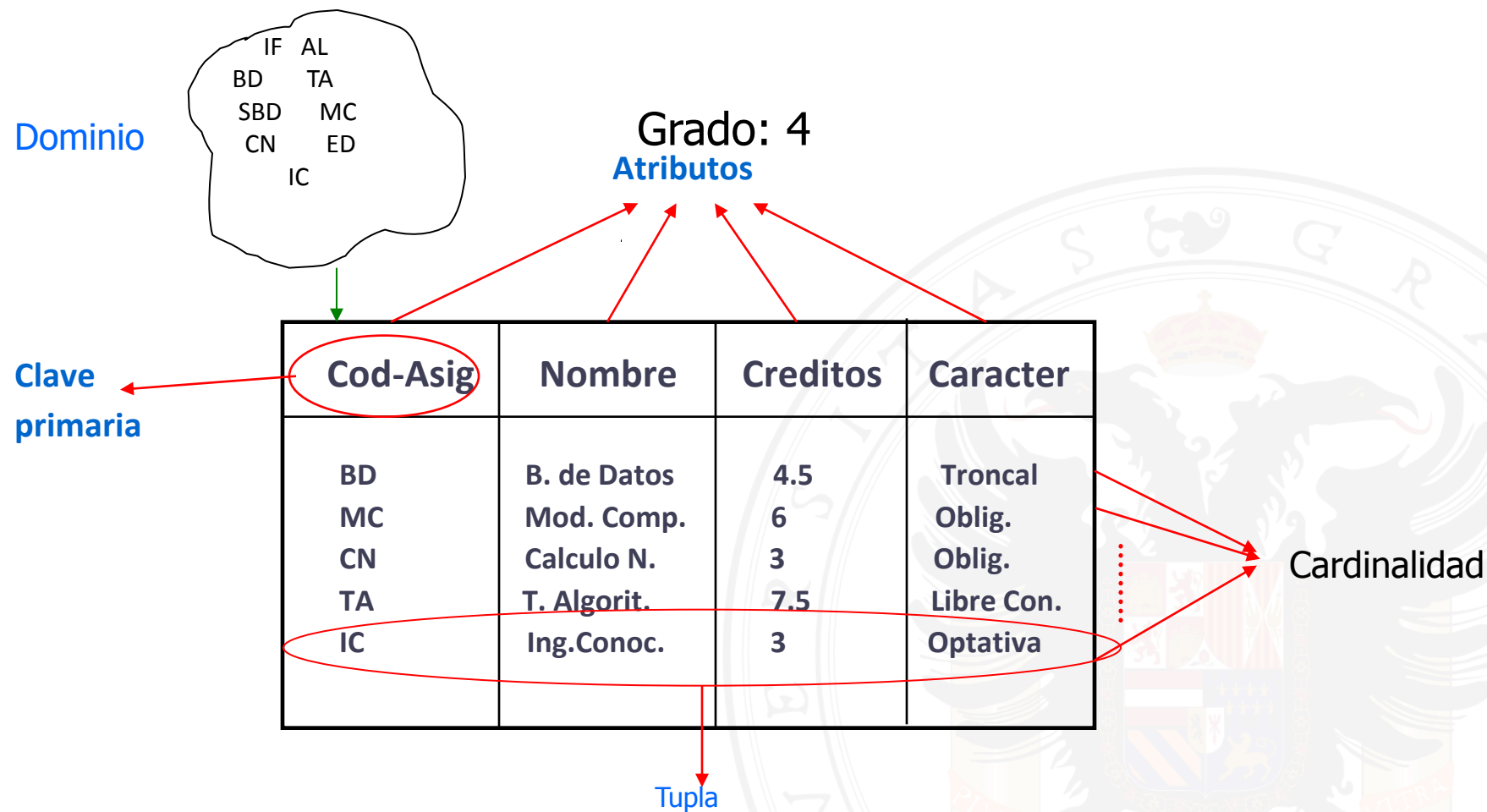
Cardinalidad de una relación: Número de **tuplas** que contiene. Es variable en el tiempo.

Esquema de una relación R : Atributos de la relación junto con su dominio, $A_1:D_1, \dots, A_n:D_n$

Grado de una relación: Número de **atributos** de su **esquema** (n). Invariable en el tiempo.

Instancia de una relación: Conjunto de **tuplas** $\{(x_1, x_2, \dots, x_n)\} \subseteq D_1 \times D_2 \times \dots \times D_n$ que la componen en cada **momento**.

Ejemplo: Asignaturas de una titulación



1. La estructura de datos relacional
2. Definiciones iniciales
3. **Propiedades de la estructura de datos relacional**
4. Notación
5. Restricciones o reglas de integridad



Condición de normalización

- Todos los **valores** de los atributos de una relación son **atómicos**.
- Valor **atómico** es un valor **no estructurado**.
- Cuando una relación cumple la primera condición de normalización se dice que está en **Primera Forma Normal**.

Consecuencias

- No hay valores **tipo conjunto**.
- No hay valores **tipo registro**.
- No hay valores **tipo tablas**.

Problema

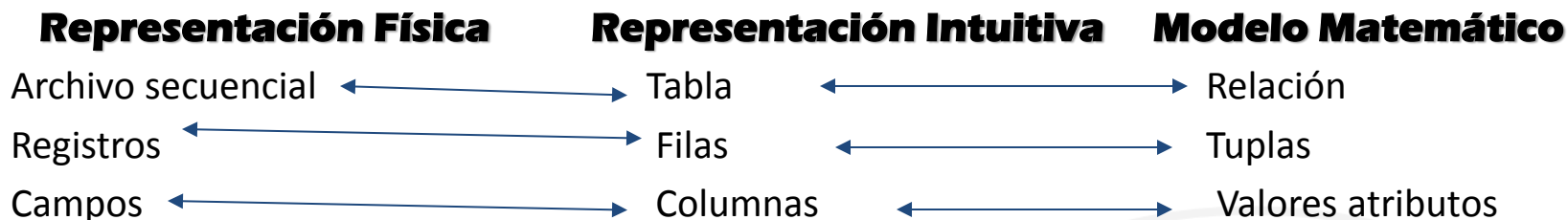
Todas las **representaciones** son **extensivas** (no se puede representar información del tipo “el valor de este atributo en esta tupla es el mismo que el valor de tal atributo en tal otra tupla”).

Consecuencias de la definición

- No hay tuplas duplicadas
 - Por la definición conjuntista de relación.
- No hay orden en las filas ni en los atributos
 - Al no estar ordenados ni los atributos ni las filas (conjuntos) el acceso es por Nombre de Atributo y Valor.
- Varias instancias representan la misma relación

A	B	C	D	E
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>d1</i>	<i>e1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>	<i>d2</i>	<i>e1</i>
<i>a2</i>	<i>b1</i>	<i>c3</i>	<i>d3</i>	<i>e1</i>
<i>a2</i>	<i>b1</i>	<i>c4</i>	<i>d3</i>	<i>e1</i>
<i>a3</i>	<i>b2</i>	<i>c5</i>	<i>d1</i>	<i>e1</i>

A	B	C	D	E
<i>a2</i>	<i>b1</i>	<i>c4</i>	<i>d3</i>	<i>e1</i>
<i>a2</i>	<i>b1</i>	<i>c3</i>	<i>d3</i>	<i>e1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>	<i>d2</i>	<i>e1</i>
<i>a3</i>	<i>b2</i>	<i>c5</i>	<i>d1</i>	<i>e1</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>d1</i>	<i>e1</i>



Esquema de una base de datos relacional

- Colección de esquemas de relaciones junto con sus restricciones de integridad.

Instancia o estado de una base de datos

- Colección de instancias de relaciones que verifican las restricciones de integridad.

Base de datos relacional

- Instancia de una base de datos junto con su esquema.

1. La estructura de datos relacional
2. Definiciones iniciales
3. Propiedades de la estructura de datos relacional
4. **Notación**
5. Restricciones o reglas de integridad



Notación a utilizar:

- Relación: R, S, T, \dots
- Atributos: A, B, \dots
- Esquema de relación: $R[A_1, A_2, \dots, A_n]$
- Instancia de relación R : r, \dots
- Tuplas de una instancia: $x_1, x_2, \dots \in r$
- Valor de un atributo A_i en una tupla x_j : $x_j[A_i]$ ó A_{ij}

Algunas veces no se conoce el valor de un atributo para una determinada tupla. En esos casos a ese atributo de esa tupla se le asigna un valor nulo (*NULL* o *Nulo*).

- Un valor nulo puede ser un valor desconocido.
- Un valor nulo puede ser un atributo no aplicable.
- En cualquier caso, ese valor es un valor más de todos los dominios de la base de datos.

1. La estructura de datos relacional
2. Definiciones iniciales
3. Propiedades de la estructura de datos relacional
4. Notación
5. **Restricciones o reglas de integridad**



Restricciones o reglas de integridad

Condiciones para preservar la semántica de una base de datos

Asociadas a tablas:

$0 \leq \text{edad} \leq 100$

$\text{créditos} > 0$

$\text{carácter} \in (\text{'troncal'}, \text{'obligatoria'}, \text{'optativa'}, \dots)$

Asociadas a la base de datos:

$\text{imparte.NRP} \in \text{profesor.NRP}$

(un profesor inexistente no puede impartir una asignatura)

$\text{cod_asig} \neq \text{nulo}$

(siempre debe conocerse el código de una asignatura)

Superclave: Cualquier conjunto de atributos que identifica unívocamente a cada tupla de una relación.

Clave de una relación: superclave minimal.

- Por ejemplo, en la relación Asignaturas, el conjunto de atributos {Cod_Asig, Nombre} identifica unívocamente cada tupla. Sin embargo, **no es minimal** y **no puede considerarse como una clave**. Cod_Asig por sí sólo, es una clave.

En una relación dada puede que más de un conjunto de atributos puedan ser elegidos como clave. Estos conjuntos de atributos se llaman claves candidatas.

Cuando hay más de una clave candidata, hay que seleccionar una como principal. Esta clave recibe el nombre de clave primaria de la tabla.

Clave candidata (CC) y primaria (CP) (definición formal).

- Sea $R[A_1, A_2, \dots, A_n]$, $PK \subseteq \{A_1, A_2, \dots, A_n\}$ se denomina **clave candidata** sii:
 - **Unicidad**: $\forall r$ instancia de R y $\forall t_1, t_2 \in r$ $t_1 \neq t_2 \Rightarrow t_1[PK] \neq t_2[PK]$
 - **Minimalidad**: No existe $PK' \subset PK$ que verifique la unicidad.
- Osea, una **clave candidata** es un atributo o conjunto de atributos que **identifican a cada tupla** en la relación y que, además, **no existe un subconjunto de ellos** que también **identifiquen a cada tupla de la relación**.
- Una clave primaria es una clave candidata elegida por el diseñador.
- Si PK verifica la **unicidad** y **no la minimalidad** se denomina **superclave**.

Conceptos generales:

- Condiciones de **integridad**:
 - Normas que **mantienen la corrección semántica** de una **base de datos**.
- Nos centramos en **Integridad Genérica**: depende del papel que juegue un atributo en el diseño de la tabla.
 - Son metarreglas (generan reglas de integridad aplicadas a una base de datos concreta).
 - Existen la **integridad de entidad** y la **integridad referencial**.

Integridad de entidad:

- No se debe permitir que una entidad sea representada en la base de datos si no se tiene una información completa de los atributos que son claves de la entidad → la **clave primaria**, o una parte de la misma, **no puede ser un valor nulo**.

Un atributo que forma parte de la clave primaria de una tupla en una relación no puede tener un valor nulo.

Clave externa: conjunto de atributos en una relación que es una clave en otra (o incluso en la misma) relación.

- Podemos ver una clave externa como un conjunto de atributos de una relación cuyos valores en las tuplas deben coincidir con valores de la clave primaria de las tuplas de otra relación.

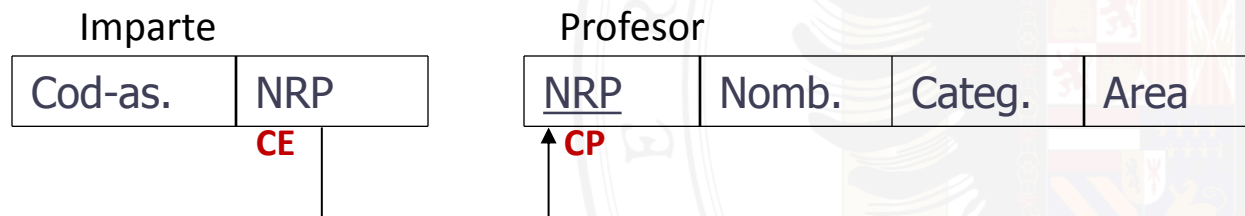
Formalmente:

– Clave externa

- Sean $R[A_1, A_2, \dots, A_n]$, y $PK \subseteq \{A_1, A_2, \dots, A_n\}$ su clave primaria, sea $S[B_1, B_2, \dots, B_n]$, y $FK \subseteq \{B_1, B_2, \dots, B_n\}$ de manera que $\text{grado}(PK) = \text{grado}(FK)$. **Fk es clave externa de S con respecto a R si se verifica:**

- $\forall r$ instancia de R y $\forall s$ instancia de S, $\forall x \in s \Rightarrow \exists y \in r / x[FK] = y[PK]$.

Es decir, el “**dominio activo**” de FK debe estar **incluido** en el “**dominio activo**” de PK para cualquier instancia de la base de datos.



Integridad referencial:

- Una base de datos en la que **todos los valores no nulos** de una **clave externa** referencian valores reales de la **clave referenciada** en la otra relación **cumple** la regla de **integridad referencial**.

Si una relación incluye una clave externa conectada a una clave primaria, el valor de la clave externa debe ser, bien igual a un valor ya existente en el dominio activo de la clave primaria, o bien completamente nulo (si la semántica lo permite).

La integridad referencial mantiene las conexiones en las bases de datos relacionales.

La integridad referencial

- Puede haber **más de una clave externa** en una **relación**.
- Puede haber una **clave externa** a la **clave primaria** de la propia relación.

Ejemplo:



EL SGBD debe encargarse de mantener las siguientes restricciones:

- La unicidad de la clave primaria y de las claves candidatas:
 - Frente a operaciones de **Inserción y Actualización**, el SGBD debe **rechazar** los valores introducidos **que sean iguales a los presentes en la BD** para los atributos que el diseñador ha definido como **clave primaria** y como **claves candidatas**.

EL SGBD debe encargarse de mantener las siguientes restricciones:

- La restricción de integridad de identidad:

Frente a operaciones de **Inserción y Actualización**, el SGBD debe **rechazar** las modificaciones que vulneren la **unicidad en la clave primaria y/o** que asignen un **valor NULO** a algún **atributo de la clave primaria**

EL SGBD debe encargarse de mantener las siguientes restricciones:

Integridad referencial (I)

En inserción:

Rechazar la tupla insertada si el valor de la clave externa **no concuerda** en la **relación referenciada** para alguna tupla en el valor su **clave primaria**.

Si el valor para la **clave externa** es **NULO** y el diseño **no lo permite** habrá de **rechazar** también esa inserción.

EL SGBD debe encargarse de mantener las siguientes restricciones:

Integridad referencial (II)

En actualización:

Si se **actualiza la clave externa**: rechazar la modificación si se produce alguna de las circunstancias descritas en punto anterior.

Si se **actualiza la clave primaria** de la relación referenciada: Actualizar en cadena las claves externa que la referencien (o impedir la actualización mientras existan referencias a valor anterior).

En borrado:

Si se **borra la clave primaria** en la relación referenciada: **borrado en cadena** de todas las tuplas que la referencian o **poner valor nulo** en la clave externa de todas esas tuplas