

# BASES DE DATOS. TEMA 5.

**El modelo de datos relacional**

# 5.1. Introducción.

- Es el modelo de datos más importante. Los nuevos modelos surgidos (orientado a objetos o multidimensional) se plantean como extensiones de este.
- Fue desarrollado por E. F. Codd en 1970, y utiliza el Algebra Relacional y el Calculo Relacional como formalismos de consulta.
- Inicialmente surge como lenguaje de consulta QUEL, y posteriormente impulsado por IBM y ORACLE aparece SEQUEL (luego SQL). Hoy en día todo gestor relacional ofrece SQL como lenguaje de consulta.
- En este tema estudiaremos los elementos de representación y las reglas de integridad asociadas al modelo relacional.

# 5.2. El modelo relacional.

## 5.2.1. Definiciones iniciales.

### □ Atributo:

Cualquier elemento de información susceptible de tomar valores.

Notación:  $A_i, i=1,2,\dots$

### □ Dominio:

Conjunto de valores que puede tomar un atributo concreto. Se considera finito.

Notación:  $D_i, i=1,2,\dots$

En la práctica: Dominio  $\rightarrow$  Tipo de datos

### □ Relación:

Dados los atributos  $A_i, i=1,2,\dots,n$  con dominios  $D_i, i=1,2,\dots,n$  definimos relación asociada a  $A_1..A_n$ , y notaremos por  $R(A_1..A_n)$  a cualquier subconjunto del producto cartesiano  $D_1 \times D_2 \times \dots \times D_n$ .

En una relación hay dos aspectos diferentes:

- **Esquema de la relación:** los atributos  $A_1..A_n$
- **Instancia de la relación:** el 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

# 5.2.1. Definiciones iniciales.

Representación Física		Representación Intuitiva		Modelo Relacional
Archivo secuencial	↔	Tabla	↔	Relación
Registros	↔	Filas	↔	Tuplas
Campos	↔	Columnas	↔	Valores atributos

- Base de datos relacional: Conjunto finito de relaciones
- Restricciones o reglas de integridad:

Condiciones sobre el valor de los atributos que preservan la semántica de una base de datos:

☞ Asociadas a tablas:

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

☞ Asociadas a la base de datos:

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

( un profesor inexistente no puede impartir una asignatura)

# 5.2.1. Definiciones iniciales.

- Esquema de una base de datos relacional:  
Colección de esquemas de relaciones junto con restricciones de integridad.
- Instancia o estado (válido) 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.
- 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  $R$ :  $r$ .
  - Tuplas de una instancia:  $x_1, x_2, \dots \in r$
  - Valores de un atributo  $A_i$  en una tupla  $x_i$ :  $x_i[A_i]$

## 5.2.2. Propiedades de las relaciones.

- Una relación es un conjunto de tuplas.

### Propiedades

1. No existe orden en las tuplas, ni en los atributos.  
El acceso es por valor, no por orden.  
Varias tablas representan la misma relación.
2. No hay tuplas duplicadas.

- Clave primaria.

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 *unicidad*.

Una *clave primaria* es una clave candidata elegida por el diseñador.

Si  $PK$  verifica *unicidad* y *no minimalidad* se denomina *superclave*.

El esquema de toda relación incluye una clave primaria.

## 5.2.2. Propiedades de las relaciones.

### □ Condición de normalización (1ª F.N.).

*Todos los valores de los atributos de una relación son atómicos.*

- Un 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 y pueden producir redundancia.

## 5.3. Reglas de integridad en el modelo relacional.

### 5.3.1. Conceptos generales.

#### □ Condiciones de integridad:

*Normas que mantienen la corrección semántica de una base de datos.*

#### □ Tipos de integridad:

➤ **Integridad específica:** depende de la semántica del problema.

- Asociadas a atributos simples:  $0 \leq \text{edad} \leq 100$
- Asociadas a conjuntos de atributos:  $\text{suelo} \geq \text{impuestos} + \text{ss}$

➤ **Integridad genérica:** depende del papel que juegue un atributo en la estructura de la base datos. Por ejemplo, si forma parte de la clave primaria.

- Son metareglas (generan reglas de integridad aplicables a cualquier base de datos concreta).
- Existen la *integridad de entidad* y la *integridad referencial*.

## 5.3. Reglas de integridad en el modelo relacional.

### 5.3.2. Formulación de las reglas de integridad.

- Las reglas de integridad se basan en los conceptos de clave primaria y clave externa.

- La **integridad de entidad**:

*Los atributos de una clave primaria no pueden tomar valores nulos.*

- Ideas básicas:

- ☞ Un valor nulo en un atributo significa “valor desconocido”.
- ☞ Una clave primaria identifica unívocamente cada tupla de forma minimal (menos atributos no identifican).
- ☞ Un valor nulo en un atributo de la clave primaria hace indistinguible la tupla.

- Ejemplo:

NOMBRE	APE1	APE2	....
Juan	Lopez	Perez	...
Juan	Lopez	Martinez	...
Juan	Lopez	null é	...

## 5.3. Reglas de integridad en el modelo relacional.

### 5.3.2. Formulación de las reglas de integridad.

La **integridad referencial** sirve para mantener coherente el estado de la BBDD.

Conceptos básicos:

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  $\text{card}(PK) = \text{card}(FK)$ ,  $Fk$  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.

Ejemplo:

	<b>FK</b>
Cod-as.	NRP

Imparte

	<b>PK</b>			
NRP	Nomb.	Categ.	Area	

Profesor

## 5.3. Reglas de integridad en el modelo relacional.

### 5.3.2. Formulación de las reglas de integridad.

#### □ 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:

	<b>PK</b>			<b>FK</b>
NRP	Nomb.	Categ.	Area	NRP

*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 bien completamente nulo (si la semántica lo permite).*

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

## 5.3. Reglas de integridad en el modelo relacional.

### 5.3.3. El mantenimiento de las reglas de integridad.

Principio básico: *La integridad debe ser mantenida por el sistema.*

- **Integridad de entidad.**

Se debe comprobar que los atributos que forman parte de una clave primaria son no nulos y que el valor conjunto de ellos no está repetido en procesos de :

***Inserción y Actualización***

- **Integridad referencial.**

***En inserción:*** comprobar que el valor de la clave externa es nulo o concuerda con un valor de la clave primaria.

***En actualización:***

Si se actualiza la clave externa: comprobar las condiciones de clave externa.

Si se actualiza la clave primaria: actualizar en cadena la clave externa.

***En borrado:***

Si se borra la clave primaria: borrado en cadena o valor nulo en cadena de la clave externa.