

# FUNDAMENTOS DE BASES DE DATOS

## TEMA 5

### El Modelo Relacional

#### T5 – El Modelo Relacional

#### Contenido

- 5.1. Introducción y objetivos**
- 5.2. Estructura del Modelo Relacional**
- 5.3. Restricciones del Modelo Relacional**
- 5.4. Grafo Relacional**
- 5.5. Estática del modelo. Lenguaje de definición**
- 5.6. Dinámica del modelo. Lenguaje de manipulación**

T5 – El Modelo Relacional

5.1. Introducción y Objetivos

---

- ◆ El *Modelo Relacional* fue presentado por **Codd** a finales de los años sesenta
- ◆ **Codd** propuso un modelo de datos basado en la *teoría matemática de las relaciones*, en el que los datos se estructuran lógicamente en forma de relaciones

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.1. Introducción y Objetivos

---

- ◆ Los objetivos perseguidos por **Codd** en la presentación de su modelo son los siguientes:
  - ❑ **Independencia Física**
  - ❑ **Independencia Lógica**
  - ❑ **Flexibilidad**
  - ❑ **Uniformidad**
  - ❑ **Sencillez**

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.1. Introducción y Objetivos

- ◆ Para conseguir los objetivos, **Codd** introduce el concepto de **relación** como estructura básica del modelo en su *Parte Estática*.
- ◆ Con respecto a la *Parte Dinámica* propone un **conjunto de operadores** que se aplican a las relaciones.

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.2. Estructura del Modelo Relacional

- ◆ El elemento básico es la **relación**, que puede representarse como una tabla.

NOMBRE

Atributo 1	Atributo 2	.....	Atributo N	
Valor 11	Valor 12	.....	Valor 1N	<i>Tupla 1</i>
.....	.....	.....	.....	.....
Valor N1	Valor N2	.....	Valor NN	<i>Tupla N</i>

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.2. Estructura del Modelo Relacional

---

♦ **Características de una relación:**

- **Grado:** Número de atributos. Poco variable en el tiempo
- **Cardinalidad:** Número de tuplas. Normalmente variable en el tiempo

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.2. Estructura del Modelo Relacional

---

DNI  
 Números  
 < 60 Mill.

NOMBRES  
 25  
 Caracteres

NOMBRES  
 25  
 Caracteres

PUESTOS  
 Técnico  
 Secretaria  
 Director

Dominios

EMPLEADO

Atributos

DNI	Apellido	Nombre	Puesto
111111	García	Manuel	Técnico
333333	Perez	María	Secretaria
555555	Sanchez	Belén	Director

Cardinalidad  
 (3)

} **Grado (4)**

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.2. Estructura del Modelo Relacional

♦ **Dominios y atributos**

- Un **dominio** es un conjunto finito de valores homogéneos y atómicos:  $V_1, V_2, \dots, V_n$
- Todo **dominio** tiene un **nombre** y un **tipo de datos**
- Un **atributo** toma sus valores de un **dominio** y está asociado a la relación. El dominio existe por sí mismo
- Es habitual que el **atributo** y el **dominio** tengan el mismo nombre
- Distintos **atributos** pueden definirse sobre el mismo **dominio**. En este caso los **atributos** tendrán distinto nombre porque una relación no puede tener dos **atributos** iguales

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.2. Estructura del Modelo Relacional

INTENSIÓN DE UNA RELACIÓN

♦ Representa la parte definitoria y estática de una relación. También se denomina *esquema de relación*.

*EMPLEADO* (dni:DNIs, Apellido: NOMBRES, Nombre: NOMBRES, Puesto: PUESTOS)

♦ Notación para un *esquema de relación*:

$$R(\{A_i:D_i\}_{i=1 \dots n})$$

Donde **R** es el nombre de la relación, **A<sub>i</sub>** cada atributo, **D<sub>i</sub>** el dominio y **n** es el grado de la relación

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.2. Estructura del Modelo Relacional

## EXTENSIÓN DE UNA RELACIÓN

- ◆ Definición de la relación a través del conjunto de **tuplas** que, en un momento determinado, cumplen el **esquema de relación**.

**EMPLEADO**

DNI	Apellido	Nombre	Puesto
111111	García	Manuel	Técnico
333333	Perez	María	Secretaria
555555	Sanchez	Belén	Director

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.2. Estructura del Modelo Relacional

- ◆ **Clases de relación**
  - **Relaciones Básicas:** Se definen con independencia de las demás relaciones. Se corresponden con el *nivel conceptual*. Siempre tienen un nombre
  - **Relaciones Derivadas:** Se definen como resultado de efectuar operaciones sobre otras relaciones básicas o derivadas. Pueden no tener existencia física, bastando con almacenarse su definición. Pueden no tener nombre

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.2. Estructura del Modelo Relacional**

---

◆ **Claves**

- **Clave Candidata:** Es un conjunto no vacío de atributos que identifican unívoca y mínimamente cada tupla de la relación
  - ❑ **Clave Primaria:** Es aquella clave candidata que se escoge para identificar las tuplas
  - ❑ **Clave Alternativa:** Son las claves candidatas que no han sido elegidas como clave primaria
- **Clave Ajena** de una relación R2 es un conjunto no vacío de atributos cuyos valores deben coincidir con los valores de la clave candidata de una relación R1 (R1 y R2 pueden ser iguales). Los dominios de la clave candidata y de la clave ajena deben ser iguales

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.3. Restricciones**

---

◆ **RESTRICCIONES INHERENTES**

- En el **MR** se derivan de la definición matemática de relación. Son las siguientes:
  - ❑ No hay tuplas duplicadas (implica la existencia de una clave primaria)
  - ❑ El orden de las tuplas no es relevante
  - ❑ El orden de los atributos no es significativo
  - ❑ Cada atributo sólo puede tomar un valor de su dominio. No se permiten grupos repetitivos

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.3. Restricciones

---

◆ **RESTRICCIONES INHERENTES**

- A estas restricciones se añade la **Regla de Integridad de Entidad** que impone lo siguiente:
  - ✓ *Ningún atributo de la clave primaria de una relación puede ser desconocido o inexistente, es decir, tomar un valor nulo*

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.3. Restricciones

---

◆ **RESTRICCIONES SEMÁNTICAS**

- Las principales son:
  - **Clave primaria (PRIMARY KEY):** Permite declarar un atributo/s como clave primaria, por lo que sus valores no pueden repetirse ni admitir nulos (valores desconocidos)
  - **Unicidad (UNIQUE):** Indica que los valores del atributo/s no pueden repetirse. Permite definir claves alternativas
  - **Obligatoriedad (NOT NULL):** Indica que el atributo/s no admite valores **nulos**

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.3. Restricciones**

◆ **RESTRICCIONES SEMÁNTICAS**

- **Integridad Referencial (FOREIGN KEY):** Si en una relación R2 hay un conjunto de atributos que son una *clave candidata* de una relación R1, los valores de ese conjunto de atributos deben coincidir con un valor de la *clave candidata* (de R1) o ser *nulos*. El conjunto de atributos es, por tanto, una *clave ajena* de la relación R2. R1 y R2 pueden ser iguales y la clave ajena en R2 puede ser parte (o totalidad) de la clave primaria de R2)

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.3. Restricciones**

**EDITORIAL**  
(R1)

Nombre	Dirección	Provincia	.....
Alfaguara	Alfacalle, 3	Madrid	.....
Gredos	Gredos, 4	Madrid	.....
Planeta	Planecalle, 2	Barcelona	.....

Código	Título	.....	Editorial
11111	El Quijote	.....	Gredos
22222	100 Años ...	.....	Alfaguara
33333	La Colmena	.....	NULO

**LIBRO**  
(R2)

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid



T5 – El Modelo Relacional

5.3. Restricciones

---

- ◆ **INTEGRIDAD REFERENCIAL: Terminología**
  - **Relación que referencia**: Es la relación donde se incluye la **clave ajena**. En la definición sería **R2**.
  - **Relación referenciada**: Es la relación que contiene la **clave candidata**, cuyos valores coincidirán con los de la **clave ajena**. En la definición sería **R1**

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.3. Restricciones

---

- ◆ **INTEGRIDAD REFERENCIAL: Consecuencias de las operaciones de borrado/modificación**
  - **Operación restringida (NO ACTION)**: El borrado de tuplas en la relación (R1) con la **clave referenciada** (o la modificación de esa clave) sólo es posible si no existen tuplas, con ese valor en la **clave ajena**, en la relación que referencia (R2)

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.3. Restricciones

◆ **INTEGRIDAD REFERENCIAL: Consecuencias de las operaciones de borrado/modificación**

- *Operación en cascada (CASCADE)*: El borrado de tuplas en la relación (R1) con la **clave referenciada** (o la modificación de esa clave) desencadena el borrado o la modificación en cascada de las tuplas, con ese valor en la **clave ajena**, de la relación que referencia (R2)

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.3. Restricciones

◆ **INTEGRIDAD REFERENCIAL: Consecuencias de las operaciones de borrado/modificación**

- *Operación con puesta a nullos (SET NULL)*: El borrado de tuplas en la relación (R1) con la **clave referenciada** (o la modificación de esa clave) implica **poner nullos** en las tuplas, con ese valor en la **clave ajena**, de la relación que referencia (R2). La **clave ajena** debería permitir valores nullos para que esta opción tenga sentido

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.3. Restricciones

---

- ◆ **INTEGRIDAD REFERENCIAL: Consecuencias de las operaciones de borrado/modificación**
  - *Operación con puesta a valor por defecto (SET DEFAULT):* El borrado de tuplas en la relación (R1) con la **clave referenciada** (o la modificación de esa clave) implica **poner un valor por defecto** en las tuplas, con ese valor en la **clave ajena**, de la relación que referencia (R2). El **valor por defecto** se establece al crear la tabla

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

T5 – El Modelo Relacional

5.3. Restricciones

---

- ◆ **RESTRICCIONES SEMÁNTICAS: Rechazo**
  - Es un tipo de restricción semántica que consiste en **formular una condición** sobre un conjunto de atributos, tuplas o dominios. La **condición** debe verificarse para que la operación de actualización sea admitida. En caso de no cumplirse se rechaza la operación.

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.3. Restricciones**

◆ **RECHAZO: Tipos**

- En el modelo relacional descrito por SQL92 hay dos tipos de rechazos:
  - **Verificación (CHECK):** Comprueba en toda actualización si se cumple la condición (*definida sobre un único elemento*). Si no se cumple se rechaza la actualización.
  - **Aserción (ASSERTION):** Igual que la anterior con la diferencia de que la condición *puede definirse sobre varios elementos*, por ejemplo, dos relaciones. Debe tener un nombre porque tienen existencia propia.

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.3. Restricciones**

◆ **RESTRICCIONES SEMÁNTICAS: Disparadores (Triggers)**

- Permiten, además de indicar una condición, especificar la acción que se llevará a cabo si la condición se hace verdadera.
- Pueden interpretarse como reglas del tipo **evento-condición-acción (ECA)** que especifican que: *cuando se produce un evento, si se cumple una condición, entonces se realiza una determinada acción*
- Los **Triggers** amplían la funcionalidad de la BD ya que permiten establecer reglas que deben cumplirse cuando se realicen operaciones INSERT, UPDATE o DELETE contra una relación específica.

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.4. Grafo Relacional**

---

- ◆ Un esquema relacional se representa mediante un grafo conocido como **Grafo Relacional**
- ◆ El **Grafo Relacional** representa gráficamente las *relaciones* de la BD y las *restricciones de integridad referencial* (clave ajena)
- ◆ En el **Grafo Relacional** se incluyen, además de las relaciones y restricciones de integridad referencial, otras restricciones como *clave primaria*, *claves alternativas* y *obligatoriedad*

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.4. Grafo Relacional**

---

- ◆ **NOTACIÓN**
  - Las **relaciones** se escriben con mayúscula
  - Los **atributos** pueden expresarse al lado del nombre de la relación, bien entre paréntesis, bien en cajas rectangulares consecutivas
  - Las **claves primarias** se subrayan con trazo continuo
  - Las **claves alternativas** se subrayan en trazo discontinuo
  - Las **claves ajenas** referencian a la relación en la que está la *clave primaria* mediante una flecha dirigida hacia el nombre de la relación o hacia la propia clave primaria de la relación.
  - Los **atributos** que *pueden tomar valores nulos* (no son obligatorios) aparecen con un asterisco

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.4. Grafo Relacional**

---

◆ **NOTACIÓN**

- Para las **opciones de borrado y actualización relativas a la integridad referencial** pueden emplearse abreviaturas, por ejemplo, para el **borrado**:
  - *B.R* Borrado restringido
  - *B.C* Borrado en Cascada
  - *B.N* Borrado con puesta a nulos
  - *B.D* Borrado con puesta a valor por defecto
- Para la **modificación**:
  - *M.R* Modificación restringida
  - *M.C* Modificación en Cascada
  - *M.N* Modificación con puesta a nulos
  - *M.D* Modificación con puesta a valor por defecto

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.4. Grafo Relacional**

---

→ **ALUMNO** (Código Alumno, Apellidos, Nombre, Dirección, Población, CP, Teléfono, Fecha Nacimiento, Becario, Licenciatura)

| BC/MC

**MATRICULA** (*Alumno*, *Asignatura*)

| BC/MC

→ **ASIGNATURA** (Código Asignatura, Nombre, Créditos, Cuatrimestre, Obligatoria, Precio, *Curso*)

| BR/MC

→ **CURSO** (Código Curso, Nombre, Campus, Licenciatura)

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

### *Tipos de Datos principales*

- ◆ **VARCHAR2 (n)** : alfanumérico (hasta 240)
- ◆ **NUMBER (n,p)** : números con signo y punto decimal
  - ◆ n = número total de dígitos a almacenar
  - ◆ p = dígitos a la derecha del punto decimal (Si se omite solo se podrá almacenar hasta 40 dígitos)
- ◆ **DATE** :Las fechas en ORACLE se almacenan como números (facilidad de conversión a otros formatos).

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

### *Tipos de Datos principales*

- ◆ **LONG** : cadenas de caracteres de hasta 64 K (solo una por tabla). Normalmente para campos resumen.
- ◆ **Observaciones:**
  - \* No se pueden indexar
  - \* No se pueden utilizar en cláusulas WHERE
  - \* No pueden ser seleccionadas en consultas anidadas
  - \* No pueden aparecer en ningún tipo de expresiones
  - \* No pueden aparecer en consultas que estén unidas a otras mediante UNION|MINUS|INTERSECT

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

### *Tratamiento de Fechas*

- ◆ **SYSDATE** : fecha actual (SELECT SYSDATE FROM DUAL;)
- ◆ **Aritmética de fechas:**
  - Fecha + X*      Añade X días a la fecha seleccionada
  - Fecha - X*      Resta X días a la fecha seleccionada
  - Fecha - X/24*    Resta X horas a la fecha seleccionada (resultados de tipo fecha)
  - Fecha - Fecha*    Devuelve el número de días transcurridos entre dos fechas

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

### *Tratamiento de Fechas*

- ◆ *Conversión de fechas :*
  - TO\_CHAR (fecha, ‘formato’):** Convierte datos de tipo fecha en datos de tipo carácter con el formato indicado.
  - TO\_DATE (cadena, ‘formato’):** Convierte datos de tipo carácter con el formato indicado, en datos de tipo fecha.

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

## Creación de Tablas

♦ **CREATE TABLE**

```
CREATE TABLE <nombre_de_la_tabla> (
  <nombre_de_la_columna1> <tipo_de_datos> [(tamaño)] [NOT NULL] [DEFAULT],
  .....
  <nombre_de_la_columnaN> <tipo_de_datos> [(tamaño)] [NOT NULL] [DEFAULT],
  [CONSTRAINT [UKnombre] UNIQUE ({<lista_de_columnas>})],
  [CONSTRAINT [PKnombre] PRIMARY KEY ({<lista_de_columnas>})],
  [CONSTRAINT [CKnombre] CHECK <condición>],
  [CONSTRAINT [FKnombre] FOREIGN KEY ({<lista_de_columnas>}) REFERENCES
    <nombre_de_tabla_referenciada> ({<lista_de_columnas>}) [ON DELETE
    CASCADE] ]
  );
```

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

## Creación de Tablas

♦ **OBSERVACIONES:**

- \* *Los nombres de las tablas de un usuario no se pueden repetir*
- \* *Los nombres de las columnas deben ser únicos dentro de cada tabla*
- \* *Los nombres pueden tener como máximo 30 caracteres*
- \* *Cada tabla puede contener hasta 254 columnas*

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

## Modificación de Tablas

- ◆ **ALTER TABLE**
  - \* *Cambiar la definición de una columna para permitir valores nulos o no, así como el tipo de datos*
  - \* *Aumentar o disminuir el tamaño de una columna*
  - \* *Añadir columnas*

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

## Modificación de Tablas

- ◆ **ALTER TABLE**

**{ADD|MODIFY|DROP|DISABLE|ENABLE}**

```

ALTER TABLE <nombre_de_la_tabla> ADD
(<nombre_de_la_nueva_columna1><tipo_de_datos> [NOT NULL] [constraint],
.....
<nombre_de_la_nueva_columnaN><tipo_de_datos> [NOT NULL] [constraint]);
                
```

```

ALTER TABLE <nombre_de_la_tabla> ADD
(CONSTRAINT <nombre_de_constraint><constraint> <columna|s>);
                
```

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

## Modificación de Tablas

- ◆ **ALTER TABLE**  
{ADD|MODIFY|DROP|DISABLE|ENABLE}

```

ALTER TABLE <nombre_de_la_tabla> MODIFY
(<nombre_de_la_columna1> <tipo_de_datos> [NULL|NOT NULL] [constraint],
.....
<nombre_de_la_columnaN> <tipo_de_datos> [NULL|NOT NULL] [constraint]);

ALTER TABLE <nombre_de_la_tabla> DROP
(<constraint> <nombre_de_la_columna(s)> CASCADE );

ALTER TABLE <nombre_de_la_tabla> {DISABLE|ENABLE}
(<constraint> <nombre_de_la_columna(s)> CASCADE );
    
```

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

## Modificación de Tablas

- ◆ **Observaciones:**

- \* *MODIFY*: para modificar el tipo o el tamaño de una columna ésta no debe contener valores

modificar una columna NULL a NOT NULL implica que no haya ninguna fila con valor NULL para esa columna, o bien que la tabla esté vacía

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

## Borrado de Tablas

- ◆ ***DROP TABLE <nombre\_de\_tabla>;***

**Cuando se ejecuta el comando DROP:**

- \* *La definición de la tabla se elimina del catálogo.*
- \* *El espacio que ocupaba la tabla se recupera para otros elementos de la base de datos*
- \* *Se borran todas las filas de la tabla sin ninguna alerta*
- \* *La ejecución de este comando elimina la tabla definitivamente*

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

## Renombrado de Tablas

- ◆ ***RENAME <antiguo\_nombre> TO <nuevo\_nombre>;***

**Cuando se ejecuta el comando RENAME:**

- \* *El nuevo nombre se almacena en el catálogo y la tabla no podrá volver a ser referenciada con el nombre antiguo*
- \* *No se produce ninguna repercusión sobre los datos*
- \* *El propietario de la tabla es el único que puede cambiar el nombre*

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

### Generación de Secuencias

- ◆ Generación de números secuenciales (autonuméricos) para las columnas

```

CREATE SEQUENCE <nombre_de_secuencia>
  [INCREMENT BY entero]
  [START WITH entero]
  [MAXVALUE entero| NOMAXVALUE]
  [MINVALUE entero| NOMINVALUE]
  [CYCLE|NOCYCLE]
  [CACHE entero|NOCACHE]
  [ORDER|NORDER]
    
```

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.5. Estática del modelo. Lenguaje de definición**

---

### Generación de Secuencias

- ◆ **Observaciones:**

- \* *<nombre\_de\_columna>.CURRVAL y <nombre\_de\_columna>.NEXTVAL se utilizan para consultar e insertar los valores de la secuencia que contiene una columna*
- \* *Una vez definida la secuencia se puede alterar con ALTER SEQUENCE*

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.6. Dinámica del modelo. Lenguaje de manipulación**

---

*Inserción de tuplas*

*Borrado de tuplas*

*Modificación de tuplas*

**Consultas**

}

**Actualizaciones**

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.6. Dinámica del modelo. Lenguaje de manipulación**

---

***Actualización: Inserción***

◆ **Inserción de valores**

```
INSERT INTO <nombre_de_la_tabla>
[<nombre_de_la_columna1>, ....., <nombre_de_la_columnaN>]
VALUES (<valor1>, ....., <valorN>);
```

Tipos:

- \* Inserción total de columnas : - el orden de los valores debe coincidir con el de la definición de las columnas.  
- si algún valor es nulo se debe especificar con la constante NULL.
- \* Inserción parcial de columnas: - se deben especificar las columnas (orden).

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.6. Dinámica del modelo. Lenguaje de manipulación**

### Actualización: Inserción

- ◆ **Inserción tuplas mediante una subconsulta**

```

INSERT INTO <nombre_de_tabla>
[(<nombre_de_columna1>, ..., <nombre_de_columnaN>)]
SELECT <lista_de_columnas_y_expresiones>
FROM <lista_de_tablas>
WHERE <criterios_de_selección>;
                
```

  - \* Se insertan los datos que figuran en la lista de tablas y que cumplan los criterios de selección.
  - \* El orden de las columnas en la cláusula SELECT debe coincidir con el orden de las columnas en la cláusula INSERT.

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.6. Dinámica del modelo. Lenguaje de manipulación**

### Actualización: Inserción

- ◆ **Inserción de tuplas sin especificar el valor de las columnas**

```

INSERT INTO <nombre_de_la_tabla>
VALUES (&columna1', '&columna2' ..., '&columnaN');
                
```
- ◆ **Consideraciones generales**
  - \* Cuando se utilice el comando INSERT parcial, todas las columnas que no estén especificadas tomarán valores nulos.
  - \* Los valores de tipo CHAR y DATE deben especificarse con comillas simples.
  - \* La inserción de datos de tipo DATE cuyo formato no coincida con el que tiene por defecto el SGBD se insertarán utilizando la función TO\_DATE ('fecha\_a\_insertar', 'formato')

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.6. Dinámica del modelo. Lenguaje de manipulación**

---

### *Actualización: Borrado*

- ◆ Borrado de todas las tuplas
 

```
DELETE [FROM] <nombre_de_tabla>
```
- ◆ Borrado de tuplas con un criterio
 

```
DELETE [FROM] <nombre_de_tabla>
            [WHERE <criterio_de_selección>]
```
- ◆ Borrado de tablas
  - ◆ DROP TABLE <nombre\_de\_tabla>

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**T5 – El Modelo Relacional**

**5.6. Dinámica del modelo. Lenguaje de manipulación**

---

### *Actualización: Modificación*

- ◆ Modificación de una tupla
 

```
UPDATE <nombre_de_tabla>
            SET columna =<nuevo_valor> [, columna = <nuevo_valor>]
            [WHERE <criterio_de_selección_del_registro>];
```
- ◆ Modificación de varias tuplas
 

```
UPDATE <nombre_de_tabla>
            SET columna = (SELECT columna y/o expresión FROM
            [WHERE <criterio_de_selección_del_registro>]);
```

©LABDA – Laboratorio de Bases de Datos Avanzadas - Universidad Carlos III de Madrid

**Actualización: Modificación**

◆ **Consideraciones generales**

- \* Si se omite la cláusula WHERE se modifican todas las filas de la tabla.
- \* El nuevo valor de la columna se puede obtener mediante expresiones, constantes o subconsultas.
- \* Cuando se utilizan subconsultas, estas deben recuperar tantas columnas como se estén modificando.