

# Apuntes para Examen – Modelo Relacional

Basado en TEMA 2. Parte 2. Modelo Relacional.pdf

---

## 1. TERMINOLOGÍA RELACIONAL

Término relacional formal	Equivalente informal
Relación	Tabla
Tupla	Fila o registro
Cardinalidad	Número de filas
Atributo	Columna o campo
Grado	Número de columnas
Clave primaria	Identificador único
Dominio	Conjunto de valores permitidos

**Condiciones que debe cumplir una relación:**

1. **Un solo tipo de registro** por tabla → cada tipo será una tabla diferente.
  2. **No existen tuplas repetidas** (el cuerpo es un conjunto matemático).
  3. **El orden de las tuplas no está determinado** (por ser conjunto).
  4. **No existen atributos repetidos** (la cabecera también es un conjunto).
  5. **El orden de los atributos no está determinado** (por ser conjunto).
-

## 2. ESTRUCTURA Y NORMALIZACIÓN

### 2.1. Definición de Base de Datos Relacional

- Colección de **relaciones normalizadas** de diversos grados que varía con el tiempo.
- Los sistemas relacionales operan sobre **tablas completas**, mediante operaciones sobre ellas.

### 2.2. Normalización

Problemas sin normalizar:

- **Redundancia** (datos repetidos).
- **Problemas de actualización** (modificaciones inconsistentes).
- **Problemas de inserción** (no se puede insertar sin ciertos datos).
- **Problemas de borrado** (se pierde información relacionada).

**Definición:** Técnica para agrupar la información en diferentes conjuntos facilitando la manipulación.

#### Dependencias

##### 1. Dependencia Funcional (DF)

$X \rightarrow Y \rightarrow$  cada valor de X tiene un único valor de Y asociado.

- Ejemplo:  $CodEmp \rightarrow NomEmp$

##### 2. Dependencia Funcional Plena

$(X1, X2) \rightarrow Y$  y ni  $X1$  ni  $X2$  por separado determinan Y.

- Ejemplo:  $(NumFac, CodArt) \rightarrow Cantidad$

##### 3. Dependencia Transitiva

$X \rightarrow Y$  y  $Y \rightarrow Z \rightarrow$  entonces  $X \rightarrow Z$  (transitivamente).

- Ejemplo:  $CodCiudad \rightarrow CodComunidad$  y  $CodComunidad \rightarrow NomComAut$

## FORMAS NORMALES

### Primera Forma Normal (1FN)

- Todos los dominios contienen **valores atómicos** (no conjuntos).
- Cada intersección fila-columna contiene **un solo valor**.

### Segunda Forma Normal (2FN)

- Debe estar en 1FN.
- Si la clave primaria es compuesta, **todos los atributos no primarios deben tener dependencia funcional plena** con la clave primaria.
- Si la clave primaria no es compuesta → automáticamente en 2FN.

### Tercera Forma Normal (3FN)

- Debe estar en 2FN.
- **Ningún atributo no primario tiene dependencias transitivas** respecto a la clave.

**Objetivo final:** Todas las tablas en 3FN.

### 3. REGLAS DE INTEGRIDAD RELACIONAL

#### 3.1. Reglas Específicas (Semánticas)

- **Propias de cada base de datos** → también se conocen como **Reglas de Integridad Semánticas**.
- Ejemplos: cantidades múltiplos de 100, no negativas, formatos *Snnn*, *Pnnn*.
- **Cómo se controlan al crear la tabla:** con cláusulas como:
  - *UNIQUE* (valores únicos)
  - *DEFAULT* (valor por defecto)
  - *CHECK* (condición que debe cumplirse)
  - También mediante *TRIGGERS*

Ejemplo SQL:

```
CREATE TABLE coche (
    CodCoc INT NOT NULL,
    MatCoc VARCHAR(8) NOT NULL UNIQUE,
    ColCoc VARCHAR(50) NOT NULL DEFAULT 'blanco',
    NumPueCoc TINYINT NOT NULL CHECK (NumPueCoc >= 2),
    PRIMARY KEY (CodCoc)
);
```

#### 3.2. Reglas Generales

##### a) Regla de Integridad de la Entidad

- Ningún componente de la clave primaria puede aceptar nulos.

##### b) Regla de Integridad de la Clave (Regla de Integridad de la Entidad en sentido amplio)

- No pueden existir valores de clave primaria repetidos en la tabla.
- **Cómo se controla:** al crear la tabla, indicando qué atributo es la clave primaria.

Ejemplo SQL:

```
CREATE TABLE departamento (
    CodDep INT NOT NULL,
    NomDep VARCHAR(100) NOT NULL,
    PRIMARY KEY (CodDep)
);
```

##### c) Regla de Integridad Referencial

- La base de datos no debe contener valores de clave ajena sin concordancia con la clave primaria referenciada.

# CLAVES PRIMARIAS Y AJENAS

## Claves Candidatas

- Atributo(s) que identifican una tupla de forma única.
- Puede haber varias.

## Clave Primaria

- Cualquier elección de una clave candidata.
- Las demás son **claves alternativas**.

## Clave Ajena (Foránea)

- Atributo(s) de una relación R2 cuyos valores deben concordar con la clave primaria de R1.
- **Puede aceptar nulos** (a diferencia de la primaria).
- Si es compuesta: todos sus componentes son nulos o todos no nulos.
- Puede ser **autorreferencial** (referenciar la misma tabla).

### Relaciones:

- **Referencial**: contiene la clave ajena.
- **Objetivo**: contiene la clave primaria referenciada.

# REGLAS PARA CLAVES AJENAS

Para cada clave ajena se debe especificar:

1. **¿Puede aceptar nulos?** (depende del modelo del mundo real).
2. **Regla de Eliminación (DELETE):**
  - **RESTRICT / NO ACTION**: no permite eliminar si hay referencias.
  - **CASCADE**: elimina también las filas referenciadas.
  - **SET NULL**: asigna NULL a la clave ajena (si acepta nulos).
3. **Regla de Modificación (UPDATE)**: mismas opciones que DELETE.

### Sintaxis SQL:

```
FOREIGN KEY (atributos) REFERENCES tabla_objetivo(atributos)
ON DELETE efecto
ON UPDATE efecto
```

### Ejemplo:

```
CREATE TABLE empleado (
    CodEmp INT NOT NULL,
    NomEmp VARCHAR(100) NOT NULL,
    CodDep INT,
    PRIMARY KEY (CodEmp),
    FOREIGN KEY (CodDep) REFERENCES departamento(CodDep)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);
```

## 4. LENGUAJES RELACIONALES

(Basado en Operaciones Relacionales.pdf)

### 4.1. Operaciones Relacionales

#### A. Operaciones Tradicionales de Conjuntos

**Condición:** Tablas compatibles (mismo grado y mismos dominios).

##### 1. UNIÓN ( $R \cup S$ )

- **Resultado:** Todas las filas de R o S (sin duplicados).
- **Ejemplo:** *TablaA UNION TablaB*

##### 2. INTERSECCIÓN ( $R \cap S$ )

- **Resultado:** Filas comunes a ambas tablas.
- **Ejemplo:** *TablaA INTERSECCIÓN TablaB*

##### 3. DIFERENCIA ( $R - S$ )

- **Resultado:** Filas de R que NO están en S.
- **Importante:** El orden importa.
- **Ejemplo:** *TablaA MINUS TablaB*

##### 4. PRODUCTO CARTESIANO ( $R \times S$ )

- **Resultado:** Todas las combinaciones posibles de filas.
- **SQL equivalente:** *FROM R, S* (sin condición WHERE)
- **Ejemplo:** *TablaA TIMES TablaB*

#### B. Operaciones Relacionales Básicas

##### 1. SELECCIÓN ( $\sigma$ )

- **Resultado:** Filas que cumplen una condición.
- **SQL equivalente:** *WHERE condición*
- **Ejemplo:** *SELECT \* FROM empleado WHERE CodDep = 1*

##### 2. PROYECCIÓN ( $\pi$ )

- **Resultado:** Algunas columnas de la tabla original.
- **SQL equivalente:** Lista de atributos en SELECT.
- **Ejemplo:** *SELECT NomEmp, ApeEmp FROM empleado*

##### 3. CONCATENACIÓN (JOIN)

- **Resultado:** Tabla que combina filas de dos tablas relacionadas.
- **SQL equivalente:** *INNER JOIN ... ON condición*
- **Ejemplo:** *SELECT \* FROM emp e INNER JOIN dep d ON e.CodDep = d.CodDep*

##### 4. DIVISIÓN

- **Resultado:** Tuplas de R que contienen todos los valores de S.
- **Ejemplo:** Encontrar proveedores que suministran TODOS los tipos de productos.

## 4.2. Ejemplos Prácticos de Álgebra Relacional

- **Seleccionar empleados del departamento 1:**

*SELECT emp WHERE CodDep = 1 GIVING R1*

- **Obtener nombres de empleados:**

*PROJECT emp OVER NomEmp, ApeEmp GIVING R1*

- **Combinar empleados con sus departamentos:**

*JOIN emp AND dep OVER CodDep GIVING R1*

## 4.3. Inserción y Borrado en Álgebra

- **Insertar:** *T UNION ((nuevos\_datos)) GIVING T*

- **Borrar:** *T MINUS ((datos\_a\_borrar)) GIVING T*

- **Borrar con comodín:** *T MINUS ((?, '2', ?)) GIVING T* → borra todas las filas con talla '2'
- 

## 5. RESUMEN PARA EL EXAMEN

1. **Terminología:** relación, tupla, atributo, clave, dominio, cardinalidad, grado.
2. **Normalización:** 1FN, 2FN, 3FN (definiciones + dependencias).
3. **Integridad:**
  - **Específicas (semánticas):** *UNIQUE, DEFAULT, CHECK, TRIGGERS.*
  - **Generales:** entidad (PK no nula), clave (PK única), referencial (FK concordante).
4. **Claves:** candidata, primaria, alternativa, ajena (reglas DELETE/UPDATE/SET NULL).
5. **Álgebra Relacional:**
  - **Conjuntos:** UNIÓN, INTERSECCIÓN, DIFERENCIA, PRODUCTO.
  - **Relacionales:** SELECCIÓN, PROYECCIÓN, CONCATENACIÓN, DIVISIÓN.

**Consejo:** Practicar ejercicios de normalización y operaciones de álgebra relacional con ejemplos concretos.