

## UT.3 Los esquemas relacionales y su transformación

Una vez hemos realizado la representación conceptual de las estructuras que almacenan los datos, sus características y sus relaciones, es necesario transformarlas a otras que sean directamente implementables por el SGBD que utilizaremos para crear y gestionar nuestra BD.

La mayoría de los SGBD se basan en tablas en las que se almacena la información, este modelo se conoce como **modelo relacional** y está basado en dos disciplinas matemáticas: la **lógica de predicados** y la **teoría de conjuntos**.

El modelo relacional fue propuesto por Edgar Frank Codd en su trabajo "A Relational Model of Data for Large Shared Data Banks" (Un modelo relacional de datos para grandes bancos de datos compartidos) en el año 1970, aunque no se implementó comercialmente hasta finales de esa década.

### 3.1. Del diseño conceptual al diseño lógico

Las tablas (también se llaman a veces relaciones) representan en el diseño lógico lo que las entidades en el diseño conceptual.

Una **relación** tiene un nombre, una serie de atributos y un conjunto de tuplas:

- El **nombre** debe ser **único**, es decir **no puede haber dos relaciones con el mismo nombre en la misma BD**
- Los **atributos** representan las propiedades de las relaciones
- Las **tuplas** son los valores que toman los diferentes atributos para cada elemento de la relación

<b>NOMBRE</b>				
<b>Atributo1</b>	<b>Atributo2</b>	...	<b>Atributo n</b>	
XXX	XXX	...	XXX	Tupla 1
XXX	XXX	...	XXX	Tupla 2
...	...	...	...	...
XXX	XXX	...	XXX	Tupla n

#### Ejemplo

<b>ASIGNATURA</b>			
<b>CodAsignatura</b>	<b>Nombre</b>	<b>NumHoras</b>	<b>CodCicloFormativo</b>
1	Bases de datos	165	1
2	Lenguajes de Marcas	120	1
3	Seguridad Informática	90	2
4	Entornos de desarrollo	95	1

#### 3.1.1. Clave primaria, claves candidatas y claves alternativas. Claves foráneas

Para que el almacenamiento de la información sea útil éste debe permitir la identificación de los datos. En el caso de las bases de datos relacionales las tuplas de las relaciones se identifican con las **superclaves**.

Una **superclave** es un subconjunto de los atributos que forman el esquema de una relación de manera que no sea posible que haya más de una tupla con la misma combinación de valores en los atributos que forman el subconjunto elegido.

Una superclave no puede admitir valores nulos en ninguno de sus atributos, porque si lo hiciera no podría garantizar la identificación inequívoca de las tuplas.

En una relación puede haber varias superclaves y las llamamos **claves candidatas**. De entre todas las **claves candidatas** el administrador o el diseñador lógico de la BD elegirá una que será la que distinguirá unívocamente cada tupla de una relación de las otras, a esta clave la llamaremos **clave primaria**.

Cuando en una relación ya se ha establecido la clave primaria, el resto de las claves que también podrían haber servido para identificar las distintas tuplas, reciben el nombre de **claves alternativas**.

**Clave externa, ajena o foránea** es un atributo cuyos valores coinciden con una clave candidata (normalmente primaria) de otra tabla.

### 3.1.2. Restricciones en el modelo relacional

En el modelo relacional existe una serie de restricciones que las podemos englobar en dos tipos,

**Restricciones definidas por el hecho de que la BD sea relacional:**

- En una relación, no puede haber 2 tuplas iguales

ASIGNATURA				
CodAsignatura	Nombre	CodCicloFormativo	NumHoras	
1	Bases de datos	1	165	
1	Bases de datos	1	165	
3	Seguridad Informática	2	90	
4	Entornos de desarrollo	1	95	

- El orden de las tuplas y el de los atributos es irrelevante

ASIGNATURA				
CodAsignatura	Nombre	CodCicloFormativo	NumHoras	
1	Bases de datos	1	165	
2	Lenguajes de Marcas	1	120	
3	Seguridad Informática	2	90	
4	Entornos de desarrollo	1	95	

ASIGNATURA				
CodAsignatura	Nombre	NumHoras	CodCicloFormativo	
1	Bases de datos	165	1	
2	Lenguajes de Marcas	120	1	
4	Entornos de desarrollo	95	1	
3	Seguridad Informática	90	2	

*Desde el punto de vista lógico, estas dos tablas son equivalentes a la del ejemplo anterior*

- Cada atributo sólo puede tomar un único valor, por ejemplo, la asignatura *Bases de datos* no puede tener a la vez los valores 1 y 2 en la columna *CodCicloFormativo*.

- Ningún atributo que forme parte del atributo principal de una relación (clave primaria), puede tomar un valor nulo.

#### Restricciones definidas por los usuarios:

- **Clave primaria (*Primary Key*):** los atributos marcados como clave primaria no puedan repetir valores. Además obliga a que esos atributos no puedan estar vacíos. Si la clave primaria la forman varios atributos, ninguno de ellos podrá estar vacío.
- **Unicidad (*unique*):** los atributos marcados como únicos no pueden repetirse. Esta restricción debe indicarse en todas las claves alternativas.
- **Obligatoriedad (*not null*):** Un atributo marcado de esta forma está obligado a tener un valor, es decir impide que pueda contener el valor nulo *null*.
- **Integridad referencial (*foreign key*):** Sirve para indicar una clave externa. Cuando una clave se marca con integridad referencial, no se podrán introducir valores que no estén incluidos en los campos relacionados con esa clave.

#### 3.1.3. Dominio

En el modelo relacional, un dominio consiste en un conjunto finito de valores. Los atributos sólo pueden tomar los valores que estén incluidos en su dominio. Si no es así, los valores no serán válidos y el SGBD no permitirá su almacenamiento. Tenemos dos tipos de dominios:

- **Dominios predefinidos:** son el tipo de datos que admite cada SGBD: cadenas de caracteres, números enteros, números decimales, fechas, etc...
- **Dominios definidos por los usuarios:** son restricciones adicionales aplicadas sobre el dominio predefinido de un atributo que son establecidos por los diseñadores y los administradores de la BD

#### Ejemplo

En una relación en la que se quiere almacenar los aspirantes a oposiciones de Bombero de la Comunidad de Madrid, existe un atributo *Edad*, se puede establecer un dominio de este campo para que no permita aspirantes con una edad inferior a 18 años.

#### 3.1.4. Esquema de una relación

Para evitar confusiones es recomendable utilizar una notación concreta a la hora de representar un esquema de las relaciones que conforman una BD:

- El nombre de las relaciones se escribirá en mayúsculas y en singular.
- El nombre de los atributos se escribirá con la primera letra en mayúsculas y las demás en minúsculas
- Las siglas se escribirán en mayúsculas
- Para leer mejor los nombres compuestos, la primera letra de cada palabra se escribirá en mayúsculas (por ejemplo: TelefonoMovil, TelefonoFijo, ...)
- Los atributos de una relación son únicos, es decir, su nombre no se puede repetir dentro de una misma relación. Pero diferentes relaciones sí pueden tener atributos con el mismo nombre.
- A veces los atributos de una relación pueden no tener ningún valor o dicho de otra manera, pueden tener valores nulos.

El diagrama lógico se representa a través de tuplas compuestas por el nombre de la relación y el conjunto de atributos entre paréntesis y separados por comas.

#### Ejemplo

El esquema de la relación que hemos visto en el ejemplo anterior quedaría:

**ASIGNATURA**(CodAsignatura, Nombre, NumHoras, CodCicloFormativo)

El orden en el que se muestran los atributos en el esquema de la relación es indiferente.

El **grado** de una relación depende del nº de atributos que incluye en su esquema. En el ejemplo anterior, dado que tenemos 4 atributos, el grado de la relación ASIGNATURA es 4.

La **extensión** de una relación son los valores de los datos almacenados en todas las tuplas que contiene. En la relación ASIGNATURA su extensión sería la lista:

**Asignatura1:** 1, Bases de datos, 165, 1 **Asignatura 2:** 2, Lenguajes de Marcas, 120, 1 **Asignatura3:** 4, Entornos de desarrollo, 95, 1 **Asignatura4:** 3, Seguridad Informática, 90, 2

La **cardinalidad** de una relación viene dada por la cantidad de tuplas que forman la extensión. En el ejemplo que tenemos, la cardinalidad de la relación ASIGNATURA sería 4, porque su extensión contiene 4 tuplas correspondientes a las 4 asignaturas que de momento tenemos en la BD.

Una manera de diferenciar los atributos que forman la clave primaria de una relación de los otros atributos del esquema es subrayándolos. Por ese motivo, normalmente se colocan juntos y delante de los demás atributos. Las claves secundarias las señalaremos poniendo U. delante de su nombre.

**Ejemplo**

Si tomamos CodAsignatura como clave primaria y CodCicloFormativo como secundaria:

**ASIGNATURA**(CodAsignatura, Nombre, NumHoras, U.CodCicloFormativo)**Ejemplo**

Fijándonos en la siguiente relación:

ALUMNO					
DNI	NumSS	NumMatricula	Nombre	Apellidos	Telefono
1111111S	12457892	4589956	José Luis	López García	914562569
2222222T	25469878	4589957	Esther	Benito Pérez	915578985
3333333D	54879565	4589960	Mariano	Bernal Martínez	917894652
8888888K	87985545	4599780	Victoria	Trujillo Sanz	NULL

El esquema sería:

**ALUMNO**(DNI, NumSS, NumMatricula, Nombre, Apellidos, Telefono)

Vemos fácilmente que los atributos *DNI*, *NumSS* y *NumMatricula* al ser personales e irrepetibles, nos podrían servir para identificar unívocamente a los alumnos, por tanto serían claves candidatas. En este momento el diseñador de la BD deberá decidirse por una clave candidata para que sea la clave primaria. Si por ejemplo, elige *DNI* como clave primaria, las antiguas claves candidatas que no han sido elegidas, pasarán a ser **claves alternativas o secundarias**. En este caso, el esquema resultante deberá indicar cuál es la clave primaria de la relación subrayando el atributo DNI, las claves secundarias o alternativas las indicamos igual que en el modelo conceptual con la letra U delante del nombre del atributo :

**ALUMNO**(DNI, U. NumSS, U. NumMatricula, Nombre, Apellidos, Telefono)

Si no hay ningún atributo que sea capaz de identificar las tuplas de la relación por sí mismo, tendremos que buscar un subconjunto de atributos, de manera que la combinación de los valores que adopten no se pueda repetir. Si esta posibilidad no existe, habrá que añadir a la relación un atributo adicional que haga de identificador.

### 3.1.5. Claves foráneas

Cualquier BD relacional, por pequeña que sea contiene normalmente más de una relación. Para poder reflejar correctamente los vínculos existentes entre algunos objetos del mundo real, las tuplas de las diferentes relaciones deben poder interrelacionarse. A veces incluso puede ser necesario relacionar las tuplas de una relación con otras tuplas de la misma relación.

Para poder interrelacionar las relaciones en una BD relacional se utilizan las **claves foráneas**.

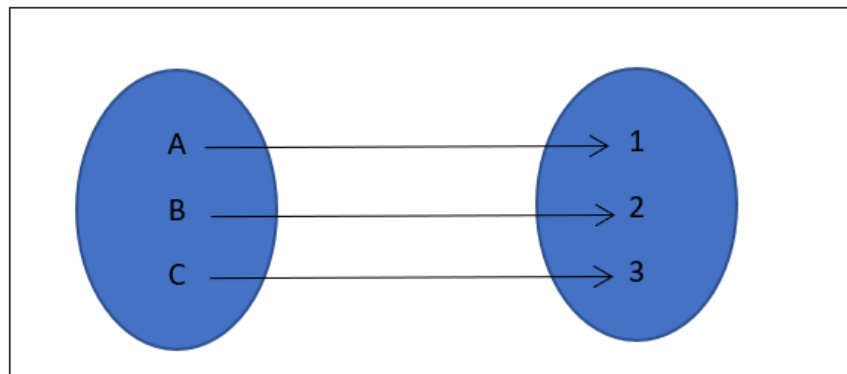
Una **clave foránea** está formada por un atributo o un conjunto de atributos del esquema de una relación que sirven para relacionar sus tuplas con las tuplas de otra relación de la BD, o con tuplas de la misma relación.

Para conseguir conectar las tuplas de una relación con las de otra (o con las suyas), la clave foránea utilizada debe referenciar a la clave primaria de la relación con la que se quiere relacionar.

Las diferentes combinaciones de valores de los atributos de cualquier clave foránea deben existir en la clave primaria a la que hacen referencia, o bien deben ser valores nulos. Si no fuera así, las referencias serían erróneas y por tanto los datos serían incorrectos.

Es importante conocer las siguientes características de las claves foráneas:

- Cualquier clave foránea debe tener el mismo número de atributos que la clave primaria a la que hace referencia.
- Entre los atributos del esquema de una clave foránea y los de la clave primaria respectiva se debe poder establecer una correspondencia



- Los dominios de los atributos de cualquier clave foránea deben coincidir con los dominios de los atributos de la clave primaria respectiva (o por lo menos deben ser compatibles dentro de un rango)
- Una relación puede tener más de una clave foránea o no tener ninguna. En sentido inverso, la clave primaria de una relación puede estar referenciada por una o más claves foráneas, o puede que no esté referenciada por ninguna.
- Puede que un mismo atributo forme parte tanto de la clave primaria como de alguna de sus claves foráneas.

## Ejemplo

CICLO	
CodCiclo Formativo	Nombre
1	Administración de sistemas informáticos en red
2	Administración de aplicaciones Web

ASIGNATURA			
CodAsignatura	Nombre	NumHoras	CodCiclo Formativo
1	Bases de datos	165	1
2	Lenguajes de Marcas	120	1
4	Entornos de desarrollo	95	1
3	Seguridad Informática	90	2

El atributo *CodCicloFormativo* de la relación **ASIGNATURA** sería la clave foránea que apunta a la clave primaria *CodCicloFormativo* de la relación **CICLO**. En este ejemplo, ambas tienen el mismo nombre, pero no tiene por qué coincidir.

Las claves foráneas apuntan a las claves primarias de otras relaciones, por eso usaremos una flecha después del nombre del atributo indicando cuál es la relación a la que se refiere. Utilizaremos la letra **F.** delante del atributo para indicar que es una clave foránea:

**ASIGNATURA**( CodAsignatura, Nombre, NumHoras, **F.** CodCicloFormativo    Ciclo)  
**CICLO**(CodCicloFormativo, Nombre)

Evidentemente, una relación puede tener varias claves foráneas, cada una apuntando a la clave primaria de la relación de la que depende. Si por ejemplo, tenemos que una asignatura la imparte un profesor y queremos saber a qué departamento pertenece:

**ASIGNATURA**( CodAsignatura, Nombre, NumHoras, **F.** CodCicloFormativo    Ciclo)

**CICLO**(CodCicloFormativo, Nombre)

**DEPARTAMENTO**(CodDepartamento, Nombre, Ubicacion)

**PROFESOR**(DNI, Nombre, Apellido1, **N.** Apellido2, **F.** CodDepartamento    Departamento, **F.** CodAsignatura    Asignatura)

### 3.2. Transformación de las entidades y los atributos

- Como regla general, cada entidad se transforma en una tabla.
- Cada atributo de una entidad se transforma en una columna de una tabla.
- Los atributos principales pasan a formar parte de la clave primaria de la tabla
- Los atributos multivaluados se convierten en una nueva tabla (débil) cuyos únicos atributos serán una clave ajena que hace referencia a la tabla principal y el valor del atributo
- Los atributos derivados se indican con una **D** y son atributos con un cálculo asociado a cada ocurrencia
- Para los atributos compuestos se crea una columna para cada uno de los atributos que los componen.
- Los atributos codificados (los que indicábamos con la letra C) se convierten en una nueva tabla. El nombre de esta tabla es el nombre del atributo con un campo clave y un campo descripción. También puede usarse la descripción como clave. La tabla que posea el atributo codificable hereda como clave foránea la clave primaria de esta nueva tabla.

#### Ejemplo

Tenemos la entidad:

ALUMNO	
<u>DNI</u>	DNI es el atributo principal
U. NumSS	NumSS es atributo secundario
Nombre	
Apellido1	
N. Apellido2	Apellido2 es opcional
MN. Telefono	Telefono es multivaluado y opcional
C. Localidad	Localidad es un atributo codificado

La transformación de la entidad ALUMNO al esquema relacional quedaría:

**LOCALIDAD** (CodLocalidad, Descripcion)

**ALUMNO**(DNI, U. NumSS, Nombre, Apellido1, N. Apellido2, F. CodLocalidad, Localidad)

**TELEFONO**(NumTelefono, F.DNI, Alumno)

#### Ejercicio

Obtener el esquema relacional de las siguientes entidades:

EMPLEADO
<u>DNI</u>
Nombre
Apellido1
N. Apellido2
C. País

**PAIS**(CodPaís, Nombre)

**EMPLEADO**(DNI, Nombre, Apellido1, N. Apellido2, F.CodPaís, País)