

## Clase=Entidad

Algo con significado para lo que queremos, de lo que tenemos información, objetos, eventos. De ese algo hay casos (objetos). Una categoría de cosas (no la cosa en concreto).

Pueden ser consideradas como conjuntos

## Ejemplos

Entidad/Clase: Persona	Caso: Tú
Entidad/Clase: Arma	Caso: Fusil ametrallador LUXMAN-26 chapado en oro con culata de caoba
Entidad/Clase: Contrato	Caso: Esclavitud
Entidad/Clase: Nivel	Caso: Cinturón negro, vigésimo dan
Entidad/Clase: Entrada	Caso: Palco real, butaca 1, para "Aida" en el Teatro Real esta noche a las 10
Entidad/Clase: Compra	Caso: 1 contenedor de Coca, a International Trade Unlimited
Entidad/Clase: Referendum	Caso: Elección de garbanzo negro de la familia

...

## Atributo

Un dato de un caso de una entidad, una propiedad, un detalle. Algo que nos interesa de la entidad. Tendrá un valor, de cierto tipo.

Tiene que tener un tipo y no más de un valor, aunque ese valor puede cambiar.

## Ejemplos

Entidad/Clase: Empleado      Atributos: apodo, cumpleaños, talla de zapato, lugar de vacaciones, ...

Entidad/Clase: Coche      Atributos: velocidad punta, altura de salto, reflectancia al radar, ...

Entidad/Clase: Pedido      Atributos: hora, año de envío, ...

Aunque algunos atributos parezcan tener más de un valor (un nombre puede ser doble), pero pueden ser considerados como una entidad única (un texto).

Huir de los atributos que van a cambiar necesariamente por sí mismos (como la edad)

## Relaciones

Son entre entidades o clases. Se consideran desde el punto de vista de ambas entidades relacionadas.

## Ejemplos

Las Personas compran Entradas

Las Entradas son compradas por Personas

Interesa la cardinalidad/inyectividad/suprayectividad

¿Todas las personas compran entradas? ¿Pueden comprar más de una?

¿Toda entrada es comprada por una persona? ¿Puede una entrada ser comprada por varias personas?

# Diagramas ER/UML

Entidades: nodos que contienen un listado de sus atributos. Los atributos tienen marcas diferentes si son obligatorios u opcionales.

Las relaciones son arcos entre los nodos, con un nombre a cada lado, indicando qué relación es. Hay distintas escuelas en ER de cómo reflejar la cardinalidad/inyectividad/suprayectividad en el final de arco y en el tipo de línea, pero de una u otra forma, se refleja. Es importante.

En UML sí está normalizado.

## Identificadores

Para distinguir e identificar cada caso de una entidad. Puede ser un valor o una combinación de ellos.

## Ejemplos

Ordenador: dirección IP

Empleado: código O nombre y fecha de nacimiento

Lista de correo: nombre y propietario

Elemento de esa lista: lista y destinatario

Observar que no tienen que ser atributos de esa entidad

Si hay varias posibilidades, se prefiere la más compacta

## Ejercicios

1. Emparejar entidades/clases y atributos  
Entidades: huésped, hotel, habitación  
Atributos: dirección, fecha de llegada, nombre, número, piso, número de camas, número de plazas de aparcamiento, precio, disponibilidad de TV
2. Posibles relaciones entre personas y ciudades
3. Diagrama ER/UML para pronóstico meteorológico, recetas

## Paso a base de datos

Entidad/clase>tabla

Atributo>columna

Identificador>clave primaria

Identificador alternativo>clave única

Relación>clave externa

Restricciones > chequeos

No todas las restricciones se pueden hacer efectivas en una declaración relacional. Algunas requieren programación.

Las relaciones uno a muchos conducen a una clave externa en el lado muchos.

Las relaciones que son muchos a muchos conducen a una tabla de intersección, cuyas únicas columnas son las claves externas de las dos tablas que se relacionan, más posibles atributos de la relación.

Las relaciones uno a uno conducen a una clave externa en la tabla donde sea obligatoria, o en cualquiera de ellas si ambas son opcionales u obligatorias.

Las relaciones reflexivas, también conducen a tabla intersección.

Las relaciones ternarias o más, conducen a tablas.

## Ejercicios

1. Diagrama ER/UML y tablas de Empleados y Departamentos

## SQL

NOT NULL: tiene que tener valor

UNIQUE: Identificador secundario, típicamente con lo anterior

PRIMARY KEY: Identificador principal elegido para esta tabla (pueden ser varios atributos)

FOREIGN KEY: Clave externa, identificador de otra tabla. En este caso, especificamos que hay que hacer en caso de que cambie o se borre en la tabla suya original:

ON DELETE CASCADE: Se borran también todos los ítems con esa clave externas

ON DELETE SET NULL: Al borrarse se ponen vacías

ON DELETE SET DEFAULT: Al borrarse se ponen al valor por defecto

ON UPDATE CASCADE: Al cambiar la tabla original, se cambia en las que usen esa clave externas

ON UPDATE SET NULL ó SET DEFAULT: equivalente

CASCADE se usa para relaciones obligatorias y las otras para las opcionales

## Ejemplos

```
CREATE TABLE DEPARTAMENTOS (NUMERO INTEGER, NOMBRE CHAR(20),  
IDIRECTOR CHAR(10) NOT NULL UNIQUE, PRIMARY KEY (NUMERO), FOREIGN KEY  
(IDIRECTOR) REFERENCES EMPLEADOS(ID) ON DELETE SET DEFAULT ON UPDATE  
CASCADE);
```

```
CREATE TABLE EMPLEADOS (ID CHAR(10), NOMBRE CHAR(20), IDPAREJA CHAR(10),  
ORDENADOR INTEGER, NUMDEPARTAMENTO INTEGER NOT NULL, PRIMARY  
KEY(ID), FOREIGN KEY (NUMDEPARTAMENTO) REFERENCES  
DEPARTAMENTOS(NUMERO) ON DELETE SET DEFAULT ON UPDATE CASCADE,  
FOREIGN KEY (IDPAREJA) REFERENCES EMPLEADOS(ID) ON DELETE SET NULL ON  
UPDATE CASCADE);
```

```
CREATE TABLE ORDENADORES(NUMERO INTEGER, IDEMPLEADO CHAR(10), PRECIO  
NUMERIC, CHECK(PRECIO>=100), PRIMARY KEY(NUMERO), FOREIGN KEY  
(IDEMPLEADO) REFERENCES EMPLEADOS(ID) ON DELETE SET NULL ON UPDATE  
CASCADE);
```

```
CREATE TABLE INFORMES(NUMERO INTEGER, IDDEPARTAMENTO INTEGER  
DEFAULT 1, PRIMARY KEY(NUMERO), FOREIGN KEY (IDDEPARTAMENTO)  
REFERENCES DEPARTAMENTOS(NUMERO) ON DELETE SET NULL ON UPDATE  
CASCADE);
```

```
CREATE TABLE AUTORES(IDINFORME INTEGER, IDEMPLEADO CHAR(10), PRIMARY  
KEY (IDINFORME, IDEMPLEADO), FOREIGN KEY (IDEMPLEADO) REFERENCES  
EMPLEADOS(ID) ON DELETE CASCADE ON UPDATE CASCADE);
```