**MODELO ENTIDAD-RELACIÓN**

Es un modelo de datos, como el *jerárquico*, *en red*, *relacional*, *dirigido a objetos*, etc. Se basa en interrelacionar objetos de datos. Trabaja en el nivel conceptual de la base de datos (no sirve para documentar el tamaño de los registros, de los campos, etc., sirve para establecer objetos y sus interrelaciones).

**Diseño de una Base de Datos:** Proceso por el que se determina la organización de una BBDD. Incluidos su estructura y su contenido. Se realiza en tres fases:

* **Conceptual:** Representación abstracta y de alto nivel de la realidad. *Modelos Entidad-Relación, orientado a objetos, binario, semántico de datos...*
* **Lógico:** Convierte esta representación en especificaciones que pueden implantarse en un ordenador y ser procesadas por él. Básicamente define las tablas del sistema Conceptual. *Modelos jerárquico, en red, relacional…*
* **Físico:** Determina las estructuras de almacenamiento físico y métodos de consulta requeridos para un acceso óptimo a la base de datos. Es el nivel en el que trabajan los administradores de bases de datos. *Es la implementación a bajo nivel. Depende del SGBD en cuestión. MySQL usa (entre otros) InnoDB.*

**Modelos de datos:** Son un conjunto de herramientas que permiten diseñar la base de datos a partir de la observación del mundo real que se quiere representar.

Por tanto, antes de hacer el modelo *Relacional (Lógico)* basado en tablas, vamos a estructurar la información *conceptualmente* usando el modelo *Entidad-Relación*.

Este modelo es el más sencillo de entender, el más fácil de representar, parecido a la realidad, y sobre todo es el más extendido.

Representa los datos semánticamente, ya que no representa la estructura de los datos, solo su contenido. Está basado en la percepción del mundo con objetos (o entidades) y relaciones entre ellos.

**Entidad:** Objeto real o abstracto con características diferenciadoras capaces de hacerse distinguir de otros objetos. Si dos objetos son indistinguibles, no pueden ser entidades diferentes. Hay dos tipos de entidades:

* **Entidades fuertes:** Entidad con existencia por sí misma. No depende de otras entidades. Por ejemplo, la entidad *médico* es una entidad fuerte porque no depende de médico.Se representan con *un rectángulo*.
* **Entidades débiles:** Entidad con existencia dependiente de otras entidades. Por ejemplo, si no hay edificio, no hay habitación. Así que *habitación* es una entidad débil y depende de *edificio*. Se representan con *un rectángulo doble*. Hay dos tipos de dependencia:
  + **Dependencia en existencia (E):** Si desaparece una instancia de la entidad fuertes, desaparecerán las entidades débiles dependientes de ellas. Si desaparece el edificio, desaparecen las aulas.
  + **Dependencia en identificación (ID):** Aparece a partir de la dependencia de existencia. Una entidad débil depende de la clave de la entidad fuerte asociada. Un ejemplar depende de la ID (ISBN) de un libro.

**Atributos:** Son las propiedades o características que tienen un tipo de entidad o un tipo de relación. Toman valores de uno o varios dominios. Se representan con un óvalo, o con una raya seguida de un círculo. Los atributos clave se subrayan, o se colorea el círculo. El dominio es el conjunto de valores permitidos para dicho atributo. Hay varios tipos de atributos:

* **Obligatorios:** Como por ejemplo los campos clave. Deben tener el valor *NOT NULL*.
* **Opcionales:** Todos los campos que podrían existir o no para una entidad. Pueden tener valor *NULL*.
* **Atómicos:** Son atributos que no pueden dividirse. *Por ejemplo: Nombre*.
* **Compuestos:** Son atributos divisibles. *Por ejemplo: Domicilio*. Hay una jerarquía en los atributos compuestos, en este ejemplo sería: *Calle, Número y Piso*. El valor de un atributo compuesto es la *concatenación* de los valores que tienen sus atributos simples. Son útiles cuando el usuario se refiere a varios atributos simples como una unidad.
* **Monovaluados:** Es aquel que tiene un único valor para cada ocurrencia de la entidad.
* **Multivaluados:** Es aquel que puede tomar diferentes valores para cada ocurrencia de la entidad. *Ejemplo: Número(s) de teléfono*. Es el nombre que se da a las relaciones {0,n} entre entidades.
* **Derivados:** Atributo que se conoce en base a otro(s) atributo(s).*Ejemplo: Empleados tienen hijos, se sabe que Empleado1 tiene 4 hijos porque se cuenta el número de atributos Hijo que tiene.*
* **Almacenados:** Es cuando se guarda un atributo directamente en la BBDD.
* **Claves:** Sirven para identificar una entidad. Hay varios tipos:
  + **Clave primaria:** Son claves candidatas que han sido elegidas por el diseñador de la BBDD para ser la clave principal de la entidad. No puede contener valores *NULL*.
  + **Clave candidata:** Claves de *<repuesto>* o segundas claves, que se usan en caso de no encontrarse la clave primaria.

**Conjunto de entidades:** Se modeliza mediante una sola entidad. Funciona de la misma manera que la programación orientada a objetos, se crea una clase (entidad “padre”) con ciertos atributos. Y luego se crean los objetos (entidades “hijas”) que heredan los atributos de la clase.

**Relación:** Asociación entre distintas entidades. En una relación no pueden aparecer dos veces relacionadas las mismas ocurrencias de entidad. Se representa con un rombo y sus atributos, igual que los de las entidades. *Ejemplo: la fecha de emisión de una factura, porque vamos a tener muchas fechas distintas*.

* **Grado de la relación:** Si puede o no intervenir más de de una entidad en una relación.
  + **Relación Aria/de Grado 1:** Participa una sola entidad (*recursiva*).
  + **Relación Binaria/de Grado 2:** Participan dos entidades.
  + **Relación Ternaria/de Grado 3:** Participan tres entidades.
  + **Relación Enearia/de Grado N:** Participan N entidades.
  + **Relación Doble:** Dos entidades se relacionan de dos maneras distintas.

* **Cardinalidad de la relación:** Número máximo de ocurrencias de cada entidad que pueden intervenir en una ocurrencia de la relación. Se representa en mayúsculas y separado por “*:*”. Se usa la cardinalidad máxima de las dos entidades juntas.
  + **Relaciones 1 a 1:** Una instancia de la entidad A solo se relaciona con una instancia de la entidad B. *Ejemplo: Expediente ----- 1:1 ----- Alumno*.
  + **Relaciones 1 a N:** Una instancia de la entidad A se relaciona con varias instancias de la entidad B. *Ejemplo: Equipo ----- 1:N ----- Jugador*.
  + **Relaciones N a M:** Varias instancias de la entidad A se relacionan con varias instancias de la entidad B. *Ejemplo: Jugador ----- N:M ----- Partido*.
* **Cardinalidad de la entidad:** La cardinalidad mínima representa el número de asociaciones que aparecerán en cada ocurrencia de la entidad. La cardinalidad máxima representa el número máximo de asociaciones que aparecerán en cada ocurrencia de la entidad.
  + **Cardinalidad 0 a 1:** En una ocurrencia de la entidad A pueden haber 0 o 1 asociaciones con la entidad B. *Ejemplo: Equipo (0,1) ----- 1:Y ----- (x,y) Jugador.*
  + **Cardinalidad 1 a 1:** En una ocurrencia de la entidad A pueden haber 1 y solo 1 asociación con la entidad B. *Ejemplo: Padre (1,1) ----- 1,Y ----- (x,y) Hijo.*
  + **Cardinalidad 0 a N:** En una ocurrencia de la entidad A pueden haber 0 o varias asociaciones con la entidad B. *Ejemplo: Miembro (0,n) ----- N,Y ----- (x,y) Club.*
  + **Cardinalidad 1 a N:** En una ocurrencia de la entidad A pueden haber 1 o Varias asociaciones con la entidad B. *Ejemplo: Hijo (1,n) ----- N,Y ----- (x,y) Padre.*
  + **Cardinalidad N a M:** En una ocurrencia de la entidad A pueden haber varias asociaciones con la entidad B. *Ejemplo:*

**Ejercicios:**

**1.- Diseña una base de datos de juegos, escribe las entidades con sus respectivos atributos.**

Juegos:

Juego1: ID, Nombre, Precio, Descripción, Puntuación

Desarrollador: ID, Nombre

Publisher: ID, Nombre

**2.- Tenemos la entidad Trabajador con los atributos: Nombre, Apellido1, Apellido2, DNI, Número de la Seguridad Social, Fecha de nacimiento y Código de empresa.**

**Buscar super claves, claves candidatas, claves primarias y claves alternativas.**

Superclaves: DNI + Nombre + Apellido1 + Apellido2, DNI + CodEmpresa, NumSS + Nombre

Claves candidatas: DNI, NumSS

Claves primarias: NumSS

Claves alternativas: DNI

**3.- Una empresa vende productos a varios clientes, se necesita saber los datos personales de los clientes (nombres, apellidos, dirección, DNI, fecha de nacimiento). Cada producto que vende la compañía tiene un nombre y un código, así como un precio unitario.**

**Los productos son suministrados por diferentes proveedores. Se debe tener en cuenta que un producto solo puede ser suministrado por un proveedor, y que un proveedor puede suministrar diferentes productos. Un producto puede ser comprado por uno o varios clientes, y un cliente puede comprar ninguno, o varios productos.**

**De un proveedor vamos a tener su NIF, nombre, y dirección.**

**4.- Se desea informatizar la gestión de una empresa de transportes que reparte paquetes por toda España. Los encargados de llevar los paquetes son los camioneros, de los que se quiere guardar DNI, Nombre, Teléfono, Dirección, Salario, Población. Un camionero distribuye muchos paquetes, y un paquete sólo puede ser distribuido por un camionero. De las provincias a las que llegan los paquetes interesa guardar el Código de Provincia y el Nombre. Un paquete sólo puede llegar a una provincia, y a una provincia pueden llegar varios paquetes. De los camiones que llevan los camioneros se guardan: Matrícula, Modelo, Tipo y Potencia. Un camionero puede conducir diferentes camiones en fechas diferentes, y un camión puede ser conducido por varios camioneros. De los paquetes transportados se guardan: Código Paquete, Descripción, Destinatario, Dirección Destinatario.**

**5.- Se desea diseñar la bbdd de un instituto. Se van a guardar los datos de los profesores: DNI, Nombre, Dirección y TLFN. Los profesores imparten módulos, los cuales tienen un Código y un Nombre. Cada alumno está matriculado en uno o varios módulos. De cada alumno se quiere guardar el Nº de expediente, Nombre, Apellidos, Fecha de nacimiento.**

**Los profesores pueden impartir varios módulos, pero un módulo puede ser impartido por solo un profesor. Cada curso tiene un grupo de alumnos, uno de los cuales es el delegado del grupo.**

**6.- Se va a hacer una bbdd para una empresa de automóviles. Hay una serie de coches para la venta. Sus datos son: Matrícula, Marca, Modelo, Color, Precio. Los datos de los clientes son: NIF, Nombre, Dirección, Ciudad, Teléfono, ID. El concesionario también lleva a cabo las revisiones de cada coche. Cada revisión lleva un código de revisión que se auto-incrementa con cada una. De cada revisión se desea saber el cambio de filtro, cambio de aceite, cambio de frenos u otros. Los coches pueden pasar varias revisiones en el concesionario.**

**7.- Clínica: Se necesita un control de la gestión de pacientes médicos. La información que se guarda de los pacientes es: Código, Nombre, Apellidos, Dirección, Población, Provincia, CP y Fecha de Nacimiento. De cada médico queremos: Código, Nombre, Apellidos, Teléfono y Especialidad. Se desea llevar el control de cada uno de los ingresos que el paciente hace en el hospital. Cada ingreso se registra. De cada ingreso se guardan: Código de ingreso, nº de habitación, cama y fecha de ingreso. Un médico puede atender varios ingresos, pero el ingreso de un paciente solo puede ser atendido por un médico. Un paciente puede hacer varios ingresos en el hospital.**

**8.- Tienda de informática: La tienda dispone de una serie de productos que se pueden vender a los clientes. De cada producto se guardan: Código, Descripción, Precio, Existencias. De cada cliente se guardan: Código, Nombre, Apellidos, Dirección, Teléfono. Un cliente puede comprar varios productos en la tienda, y un mismo producto puede ser comprado por varios clientes. Cada vez que se compre un artículo se registrará la compra con su fecha de compra. La tienda tiene contactos con varios proveedores. Un mismo producto puede ser suministrado por varios proveedores. De cada proveedor se guarda: Código, Nombre, Apellidos, Dirección, Teléfono.**

**MODELO ENTIDAD-RELACIÓN EXTENDIDO**

**Restricción de exclusividad:** Si hay dos entidades relacionadas por dos o más relaciones distintas, se producirá una relación o la otra, pero nunca ambas a la vez. Se dice que *un músico puede tocar y dirigir* una orquesta, pero nunca tocará y dirigirá a la vez. Se representa en el modelo E-R cortando con un arco las relaciones.

**Restricción de exclusión:** Se produce cuando las ocurrencias de las entidades solo pueden asociarse usando una única ocurrencia de la relación. Un *entrenador personal* no puede asistir a un curso para el perfeccionamiento de entrenadores personales, si es él mismo quien imparte el curso para otros entrenadores personales. O lo imparte, o asiste. Se representa en el modelo E-R relacionando las dos relaciones con una línea discontínua.

**Restricción de inclusividad:** Se aplica cuando es necesario modelar situaciones en las que para que dos ocurrencias de entidad se asocien a través de una relación, antes se ha tenido que dar otra relación entre ellas. Para que *un entrenador impartiera cursos a otros monitores, este antes tendría que haber dado un curso para ser entrenador*. Se representa en el modelo E-R cortando las relaciones con un arco acabado en flecha.

**Restricción de inclusión:** N/A

**Relaciones de generalización y especialización:** Relaciones jerárquicas y de herencia. Por ejemplo, un *bombero forestal* es una especialización de *bombero*. Se representa en el modelo E-R mediante una flecha triangular en la intersección de la relación entre las entidades especializadas.

**9.- Suponemos la existencia de dos entidades, turismo y camión. Los atributos de turismo son: Nº de bastidor, Fecha fabricación, Precio, Nº de puertas. Los atributos de camión son: Nº de bastidor, Fecha de fabricación, Precio, Nº de ejes, Tonelaje.**

**Agregaciones:** Se pueden encerrar conjuntos de entidades y relaciones dentro de una sola entidad, de forma que se comporte como tal a efectos prácticos. De esta forma podemos juntar dos entidades y una relación dentro de una sola entidad para relacionarlo con una entidad distinta.

**1.- Obtener el diagrama E/R:**

* **REGIÓN (Nombre\_Región)**
* **PROVINCIA (Código\_Provincia, Nombre\_Provincia)**
* **LOCALIDAD (Código\_Localidad, Nombre\_Localidad)**
* **EMPLEADO (ID\_Empleado, DNI\_Empleado, Nombre\_Empleado, Tlfn, Salario)**

**Se quiere guardar la información de la localidad donde ha nacido cada uno de los empleados teniendo en cuenta que:**

* **Un empleado ha nacido en una sola localidad.**
* **Cada localidad pertenece a una única provincia.**
* **Cada provincia pertenece a una única región del país.**

**2.- Obtener el diagrama E/R:**

* **ALUMNO (Nº\_Matrícula, Nombre, Fecha\_Nac, Tlfn)**
* **ASIGNATURA (Cod\_Asignatura, Nombre)**
* **PROFESOR (ID\_Profesor, NIF\_Profesor, Nombre\_Profesor, Especialidad, Tlfn)**

**A tener en cuenta:**

* **Un alumno puede estar matriculado de una o varias asignaturas.**
* **Además, puede estar matriculado en la misma asignatura más de un curso escolar (si repite).**
* **Se quiere saber el curso escolar en el que cada alumno está matriculado de cada asignatura.**
* **En una asignatura habrá como mínimo 10 y como máximo 25 alumnos.**
* **Una asignatura es impartida por un único profesor.**
* **Un profesor podrá impartir varias asignaturas.**

**3.- Diseña una BD para una consultora sabiendo que:**

* **Empleado: NIF, Nombre, Apellidos, F\_nacimiento, tlfno y titulación**
* **Cada empleado pertenece a un solo departamento de la empresa. Departamento: Código y nombre**
* **Cada departamento tiene al menos 3 empleados**
* **Cada empleado trabajará en diferentes Proyectos: Código y Descripción**
* **Cada vez que un empleado trabaje en un proyecto, es necesario almacenar la fecha de incorporación al mismo, así como si hace funciones de coordinación en el mismo.**

**4.- Sobre el ejercicio 3 realiza las siguientes modificaciones:**

* **Cada departamento podrá tener Secciones: Cod\_seccion, Nombre, Area\_Trabajo.**
* **Los hijos de los empleados pueden obtener ayudas por estudio. Para ello almacenamos el nombre de cada hijo y su fecha de nacimiento.**
* **Ayudas: Código, Descripción, Cuantía. También se almacena el año en que se da cada ayuda.**
* **Los hijos de los empleados podrán recibir varias ayudas, pero siempre en diferentes años.**

**5.- Sobre el ejercicio 4 realiza las siguientes modificaciones:**

* **Si el empleado no es español, se almacena su nacionalidad y nombre del país en que tributa.**
* **En caso de que el empleado sea español, debes almacenar un nº de cuenta.**
* **Un proyecto puede ser internacional o local. En caso de que sea internacional, habrá que poner el país y la localidad donde se realiza, así como el nombre de la aseguradora de dicho país. En caso de que el proyecto sea local, hay que tener en cuenta, que en dicho proyecto no pueden trabajar más de 5 empleados.**

**6.- Crear un diseño entidad relación que permita gestionar los datos de una biblioteca de modo que:**

**● Las personas socias de la biblioteca disponen de un código de socio y además se necesita almacenar su dni, dirección, teléfono, nombre y apellidos**

**● La biblioteca almacena libros que presta a los socios y socias, de ellos se almacena su título, su editorial, el año en el que se escribió el libro, el nombre completo del autor principal, el año en que se editó y en qué editorial fue y el ISBN.**

**● Necesitamos poder indicar si un volumen en la biblioteca está deteriorado o no**

**● Queremos controlar cada préstamo que se realiza almacenando la fecha en la que se realiza y la fecha real en la que se devuelve el libro.**

**UML**

Hay varios modelos de diagramas:

* Diagramas de clase: Nombre de la tabla, Atributos, Funciones de clase. La parte de funciones de clase no la vamos a usar.

Cardinalidades

|  |  |  |
| --- | --- | --- |
| **Multiplicidad** | **Abreviatura** |  |
| 0..1 |  |  |
| 0..\* | \* |  |
| 1..1 | 1 | Se da por hecho |
| 1..\* |  |  |
| n..\* |  | n>=0 |
| n..m |  | m>=n>=0 |