

Desarrollo de aplicaciones web / multiplataforma



M02A: Bases de datos

UF1c. Modelo relacional

Índice

1. Introducción
2. El modelo de datos
3. Modelo Entidad/Relación

1. Introducción.

- El modelo relacional es el siguiente nivel de abstracción al que debemos enfrentarnos a la hora de modelar nuestra base de datos.
- En el modelo relacional tratamos la base de datos como una colección de relaciones, relaciones que pueden ser operadas mediante el Álgebra Relacional.

1. Introducción.

- Relación → Conjunto de registros compuestos de parámetros.
- Se representa gráficamente en forma de tabla con atributos dispuestos en columnas y filas que representan las tuplas.
- El conjunto de tuplas representa el cuerpo de la relación, mientras que el conjunto de atributos junto con el nombre definen el esquema de la misma.

ISBN	Título	Nombre Autor	Apellido Autor	Precio
978-0062511409	El Alquimista	Paulo	Coelho	\$8.46
978-0307744593	Aleph	Paulo	Coelho	\$12.23
978-034580704	El peregrino	Paulo	Coelho	\$12.20

2. Conceptos Fundamentales

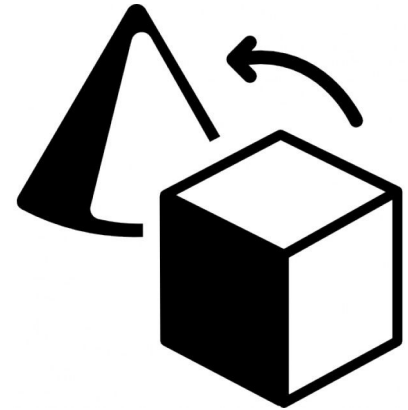
- Atributo → Cada una de las propiedades de una entidad o relación.
- Dominio → Valores que puede adoptar un determinado atributo.
- Restricciones de semántica:
 - De clave → Conjunto de atributos que identifican a una entidad.
 - De valor único → Indica que no pueden existir valores repetidos para un atributo.
 - De integridad referencial → Relacionado con los valores existentes en varias tablas. Se tratará más adelante.
 - Restricciones de dominio → Implican que un atributo debe pertenecer a un dominio concreto.
 - Restricciones de valor nulo → En caso de que un atributo sea obligatorio, este no admitirá valor NULL.

2. Conceptos Fundamentales

- Clave → Definimos clave como el atributo o conjunto de atributos que identifican de forma unívoca una ocurrencia dentro de una relación.
 - Clave candidata → Cualquier atributo que sea susceptible de identificar una tupla en concreto. Por ejemplo, si hablásemos de vehículos, podríamos afirmar que, tanto el número de matrícula como el número de bastidor son claves candidatas, pues ambos son únicos.
 - Clave primaria → De entre las claves candidatas, la clave primaria es el campo o atributo que finalmente identificará a la tupla. En muchas ocasiones la elección de una de las claves candidatas como clave primaria depende de la visión subjetiva del diseñador.
 - Clave foránea (o clave ajena) → Definimos clave ajena como un atributo dentro de nuestra relación, que hace referencia a la clave primaria de otra relación diferente. Se emplea para establecer interrelaciones entre diferentes tablas.

3. Transformación de E/R a Relacional

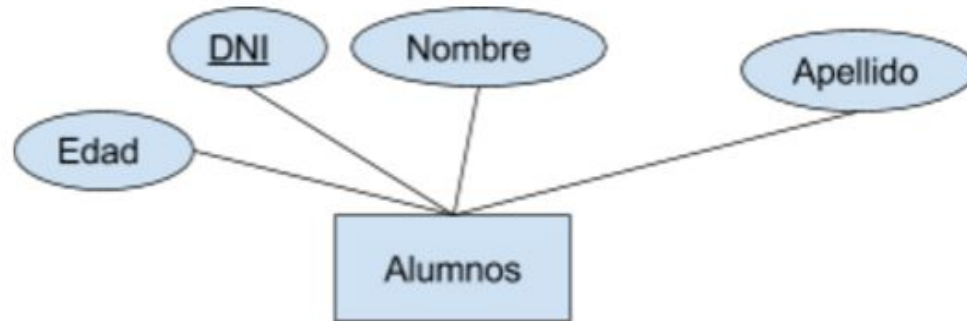
- El modelo relacional es el paso intermedio entre el modelo Entidad/Relación y el modelo físico computerizado.
- Para obtenerlo (A partir de un E/R) es necesario seguir una serie de normas y pautas de conversión para obtener las relaciones (Tablas)



3. Transformación de E/R a Relacional

Transformación de entidades fuertes

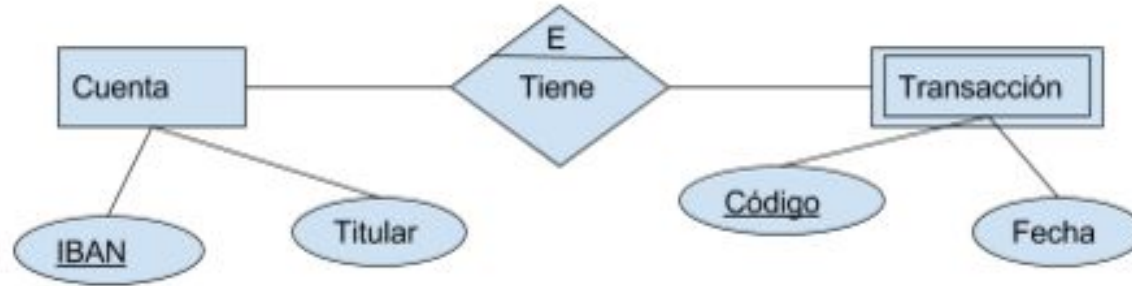
- Una entidad fuerte del diagrama E/R se transforma en una tabla en el modelo relacional
- Cada atributo de la entidad se traduce en una columna de la tabla.
- La clave primaria de la tabla será el atributo clave o la combinación de atributos clave de la entidad.



3. Transformación de E/R a Relacional

Transformación de entidades débiles (Existencia)

- La transformación se realiza de forma similar a como transformaríamos una entidad fuerte.

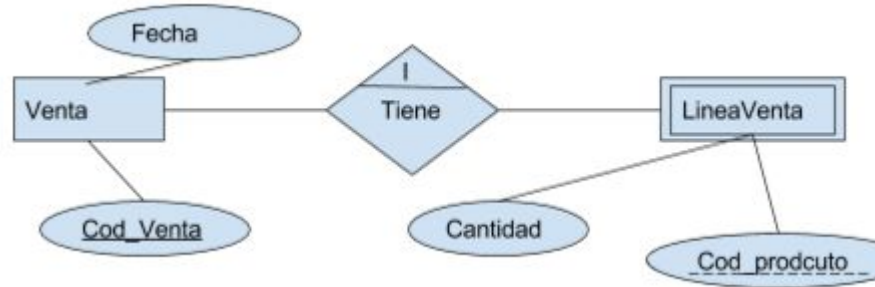


Cuenta (<u>IBAN</u> , Titular)
Transacción (<u>Código</u> , Fecha, <u>IBAN</u> _Cuenta)

3. Transformación de E/R a Relacional

Transformación de entidades débiles (Identificación)

- La clave primaria de la entidad fuerte deberá pasar a formar parte de la clave primaria de la entidad débil, a fin de dotarla de identidad.



Venta (Cod_Venta, Fecha)

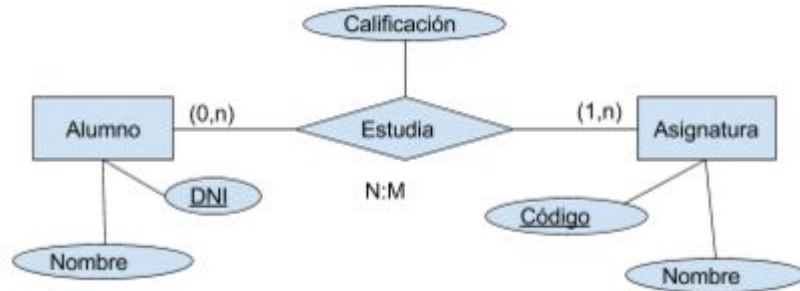
LineaVenta(Cod_venta, Cod_producto, Cantidad)

3. Transformación de E/R a Relacional

Transformación de relaciones (N:M)

- Cada relación N:M genera una nueva tabla.
- Esta nueva tabla contendrá las claves primarias de las entidades que participan en la relación a modo de clave primaria propia.
- En caso de que la relación tuviera atributos se incluirán también en esta nueva tabla.

Alumno(<u>DNI</u> , Nombre)
Asignatura (<u>Código</u> , Nombre)
Estudia(<u>DNI</u> , <u>Alumno</u> , <u>Código</u> , <u>Asignatura</u> , Calificación)

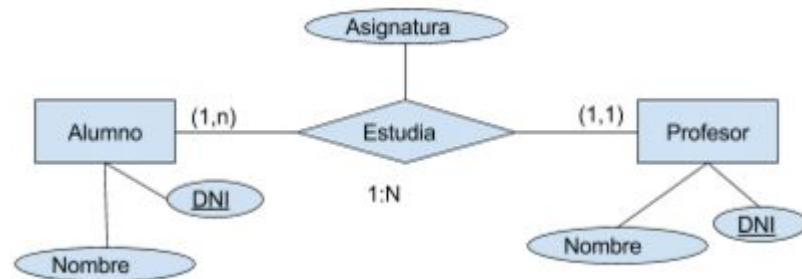


3. Transformación de E/R a Relacional

Transformación de relaciones (1:N)

- La clave primaria de la entidad con menor participación migra hacia la tabla generada por la otra entidad a modo de clave ajena.
- En caso de que la relación tuviera atributos estos se incluyen en la misma tabla hacia la que hemos migrado.

Profesor(<u>DNI</u> , Nombre)
Alumno (<u>DNI</u> , Nombre, DNI_Profesor, Asignatura)

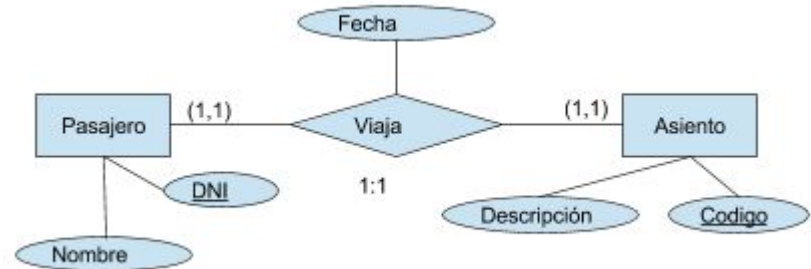


3. Transformación de E/R a Relacional

Transformación de relaciones (1:1)

- En este caso se procede del mismo modo que con las relaciones 1:N, pero en este caso es el diseñador quien decide hacia qué tabla migraremos claves y atributos de la relación.

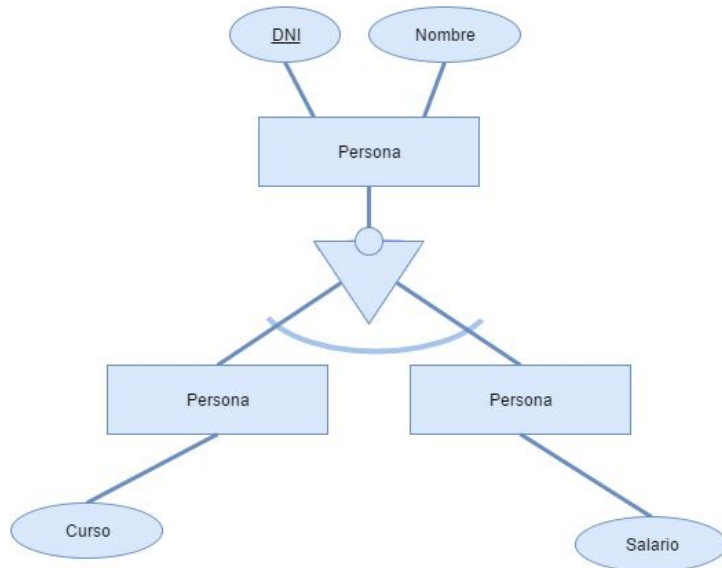
Pasajero(<u>DNI</u> , Nombre)
Asiento(<u>Código</u> , Descripción, <u>DNI_Pasajero</u> , Fecha)



3. Transformación de E/R a Relacional

Transformación de relaciones (Especialización)

- Para este tipo de relaciones se pueden adoptar diferentes estrategias dependiendo del tipo de especialización del que se trate.
- Partiremos del siguiente supuesto y veremos diferentes formas de transformarlo.



3. Transformación de E/R a Relacional

Transformación de relaciones (Especialización)

- a) Crear una tabla para la superclase y otras por cada subclase, de modo que la clave primaria de todas ellas sea la clave de la superclase.

Persona(<u>DNI</u> , Nombre)
Estudiante (<u>DNI</u> , Curso)
Profesor(<u>DNI</u> , Salario)

- b) Crear una tabla por cada subclase, incorporando a ellas la información de la superclase.

Estudiante(<u>DNI</u> , Nombre, Curso)
Profesor (<u>DNI</u> , Nombre, Salario)

3. Transformación de E/R a Relacional

Transformación de relaciones (Especialización)

c) Crear una única tabla en la que incluyamos tanto los campos de la superclase como los correspondientes a las subclasses, añadiendo un último campo que indique de qué tipo de entidad estamos hablando.

Persona(<u>DNI</u> , Nombre, Curso, Salario, Tipo)

d) Crear una única tabla, del mismo modo que hacíamos en el caso anterior, pero reemplazando en este caso el campo tipo por campos binarios que indiquen el tipo.

Persona(<u>DNI</u> , Nombre, Curso, Salario, esProfe, esAlumno)
--

3. Normalización

- Una vez definido el modelo relacional se debe pasar por un proceso de “refinamiento” para garantizar que nuestro modelo cumple con las convenciones y normas establecidas.
- Existen una serie de condiciones que un modelo debe cumplir para ser correcto, estas se dividen en distintas *formas normales*.

Conceptos Previos

- **Dependencia funcional** → Sean X e Y subconjuntos de atributos de una relación. Diremos que Y tiene una dependencia funcional de X, o que X determina a Y, si cada valor de X tiene asociado siempre un único valor de Y.
- **Dependencia funcional completa** → En una dependencia funcional $X \rightarrow Y$, cuando X es un conjunto de atributos, decimos que la dependencia funcional es completa, si sólo depende de X, y no de ningún subconjunto de X.
- **Dependencia funcional transitiva** → Supongamos que tenemos una relación con tres conjuntos de atributos: X, Y y Z, y las siguientes dependencias $X \rightarrow Y$, $Y \rightarrow Z$, $Y \rightarrow IX$. Es decir X determina Y e Y determina Z, pero Y no determina X. En ese caso, decimos que Z tiene dependencia transitiva con respecto a X, a través de Y.

3. Normalización

Primera Forma Normal (1FN)

Para cumplir con esta forma normal debemos evitar que existan atributos multivaluados en nuestra tabla. En el siguiente ejemplo podemos la diferencia entre una misma tabla cumpliendo FN1 y sin hacerlo.

Tabla original

Película	Sala	Pases
Casablanca	1	16.30H 17.40H 19.50H
El Pianista	2	17.50H 20.00H
RoboCop	3	21.00H 23.00H



Tabla en 1FN

Película	Sala	Pases
Casablanca	1	16.30H
Casablanca	1	17.40H
Casablanca	1	19.50H
El Pianista	2	17.50H
El Pianista	2	20.00H
RoboCop	3	21.00H
RoboCop	3	23.00H

3. Normalización

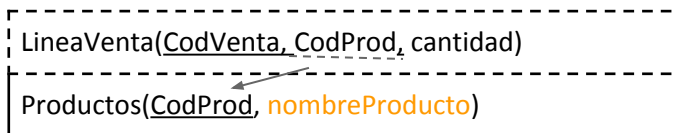
Segunda Forma Normal (2FN)

Diremos que un modelo de datos está en FN2 si y sólo si se encuentra en FN1 y cada atributo que no forma parte de la clave tiene dependencia completa de la misma.

Tabla original

LineaVenta(<u>CodVenta</u> , <u>CodProd</u> , cantidad, nombreProducto)
--

Como podemos ver, nombreProducto no depende directamente de la clave de LineaVenta sino que es un atributo de producto, por tanto debería quedar



3. Normalización

Tercera Forma Normal (3FN)

Diremos que un modelo se encuentra en FN3 si y sólo si se encuentra en FN2 y además, no existe ningún atributo no clave que dependa transitivamente de la clave.

Tabla original

Películas(<u>Nombre</u> , Género, Director, Fecha_Nacimiento)
--

Tanto género como director dependen completamente de la clave, sin embargo, la fecha de nacimiento presenta una dependencia transitiva con la película, pues realmente depende de Director. Por tanto:

Películas(<u>Nombre</u> , Género, Director)
--

Director(<u>NIE</u> , Nombre, Apellidos, Fecha_Nacimiento)

3. Normalización

Forma normal de Boyce-Codd (FNBC)

Diremos que un modelo se encuentra en FNBC si y sólo si se encuentra en 3FN y además no existe ningún determinante (Atributo con el que otro presenta una dependencia funcional completa) que no sea clave candidata.

Tabla original

ProductosProveedor(Cod_prod, cod_prov, CIF_Prov)
--

El propósito de esta tabla es mostrar información sobre qué proveedor nos proporciona cada producto, por tanto las claves candidatas serían {Cod_prod, Cod_Prov} o bien {Cod_prod, CIF_Prov} en cualquiera de los dos casos existe un determinante que no forma parte de la clave candidata. Para que esta relación quedase en FNBC:

ProductosProveedor(<u>Cod_Prod</u> , <u>Cod_Prov</u>)

Proveedores(<u>Cod_Prov</u> , CIF_Prov)
--