



Luis Valencia Cabrera

lvalencia@us.es

(<http://www.cs.us.es/~lvalencia>)

Ciencias de la Computación e IA

(<http://www.cs.us.es/>)

Universidad de Sevilla

- Bases de Datos –
(2012/2013)

Tema 2:
Diseño lógico.
Modelo
Relacional

Indice

- Introducción al Modelo Relacional
- Definiciones
- Propiedades
- Claves primarias y ajenas
- Valores nulos y restricciones de integridad
- Paso del DER al esquema relacional
- Bibliografía

Introducción al Modelo Relacional

Introducción al Modelo Relacional (I)

- Fue introducido por **E.F. Codd** (Edgar Frank Codd, “Ted Codd”) en **1970**. Aunque los **primeros SGBD relacionales** no aparecieron hasta los **años 80**.
- Supuso una **revolución en el diseño lógico de BD**, dando lugar a la *segunda generación de SBGD*.
- Es el **modelo lógico más extendido** (ORACLE, SQL Server, PostgreSQL, MySQL, Access, dBaseIV, Base...)
- Los **datos se estructuran lógicamente en forma de relaciones** (tablas).
 - Ojo! Notar que en los DER hablamos de entidades y relaciones (o interrelaciones). No obstante, en el modelo relacional todo son relaciones.
- Intuitivamente, una BD relacional es un **conjunto de tablas bidimensionales enlazadas entre sí**.

Introducción al Modelo Relacional (II)

- Conceptos fundamentales:
 - **Relación:** tabla bidimensional
 - **Registro** o tupla: fila de la tabla
 - **Campo:** columna de la tabla
- Ejemplo:
 - Tabla ESCRITOR (2 registros de 4 campos)

DNI	Nombre	Dirección	Fecha
44345789	Ana Pérez	Sol, 17	9/5/1960
56123009	Luis Gómez	Feria,2	5/5/1961

- Las tablas se enlazan entre sí mediante campos con contenido común.

Definiciones

Definiciones (I)

- Una relación de grado m consta de dos partes:
 - Cabecera: conjunto fijo de m campos.
 - Cada campo esta definido por su Nombre y su Dominio (que indica el tipo de valores que contendrá dicho campo).
 - $\{(\text{Nombre}_1: \text{Dominio}_1), \dots, (\text{Nombre}_m: \text{Dominio}_m)\}$
 - Cuerpo: conjunto variable de registros (también denominados tuplas).
 - Un registro es un conjunto de m valores:
 - $\{(\text{Nombre}_1: \text{Valor}_1), \dots, (\text{Nombre}_m: \text{Valor}_m)\}$
 - ...
 - $\{(\text{Nombre}_1: \text{Valor}_1), \dots, (\text{Nombre}_m: \text{Valor}_m)\}$

Definiciones (II)

- **Notas:**

- Cada relación tiene asociado un Nombre que la identifica.
- Una relación de grado m puede representarse mediante una tabla bidimensional de m columnas y tantas filas como registros aparezcan en la relación.
- Cada valor de un registro debe pertenecer al correspondiente dominio especificado en la cabecera.

Definiciones (III). Ejemplo

DNI	Nombre	Dirección	Fecha
44345789	Ana Pérez	Sol,17	9/5/1960
56123009	Luis Rus	Feria,2	5/5/1961

- La cabecera de la relacion Escritor es:
 - {(DNI:Numérico), (Nombre:Texto), Direccion:Texto), (Fecha:Fecha/Hora)}
 - El cuerpo de la relación Escritor esta formado por 2 registros:
 - {(DNI:56123009),(Nombre:"Luis Rus"), (Direccion:"Feria,2"), (Fecha:5/5/1961)}
 - {(Fecha:9/5/1960), (DNI:44345789), (Direccion:"Sol,17"),(Nombre:"Ana Perez")}

Propiedades

Propiedades de la cabecera de una relación

- Cada relación tiene asociada una cabecera formada por un número fijo de campos.
 - Notación: Nombre1.Nombre2 denota el campo Nombre2 de la cabecera de la relación Nombre1.
 - Dos campos pertenecientes a la cabecera de la misma relación no pueden tener el mismo nombre.
- Campos de relaciones distintas pueden tener el mismo nombre:
 - Escritor.DNI denota el campo DNI de la relación Escritor.
 - Cliente.DNI denota el campo DNI de la relación Cliente.
- El orden de los campos en la cabecera de una relación no importa.

Propiedades del cuerpo de una relación

- Todos los registros de una relación deben tener el mismo número de campos, aunque alguno este vaco. En este caso, dicho campo vaco toma el valor NULL.
- Los valores de los campos son atómicos: fijado un registro, cada campo toma un único valor (no se admiten campos multivaluados).
- No se admiten registros duplicados. Dos registros de una relación deben diferir, al menos, en el valor de un campo.
- El orden de los registros en el cuerpo de una relación no importa.

Propiedades de las relaciones. Ejemplos (I)

DNI	Nombre	Dirección	Fecha
44345789	Ana Pérez	Sol, 17	9/5/1960
40876100	José Ruíz	Luna,1	1/1/1972
56123009	Luis Gómez	Feria,2	NULL

Dirección	Nombre	Fecha	DNI
Feria,2	Luis Gómez	NULL	56123009
Luna,1	José Ruíz	1/1/1972	40876100
Sol,17	Ana Pérez	9/5/1960	44345789

- En el modelo Relacional, las dos relaciones anteriores son idénticas. Solo difieren en el orden de los campos y los registros.

Propiedades de las relaciones. Ejemplos (II)

Nombre y Apellido	Edad	Estudios
Juan Pérez	41	Lcdo. Química
Ana Sánchez	37	Lcdo. Medicina Lcdo. Física
Juan Pérez	41	Lcdo. Química
Félix González	32	NULL

- La relación anterior NO es válida en el modelo relacional:
 - posee campos multivaluados,
 - posee registros repetidos.

Propiedades de los campos

- Cada campo debe poseer un Nombre (relacionado con los datos que contendrá) y debe tener asociado un Tipo de dato.
 - Texto: cadenas de caracteres, ya sean números (con los que no se vaya a realizar operaciones), letras o símbolos.
 - Numérico: números destinados a realizar operaciones.
 - Fecha/hora: almacena fechas y horas.
 - Sí/No: datos que solo tengan dos posibilidades (verdadero-falso).
 - Autonumérico: valor numérico (1,2,3,...) que el SGBD incrementa de modo automático cuando se añade un registro.
 - Memo: almacena texto largo.
 - Moneda: almacena valores de moneda.
 - Objeto OLE: almacena gráficos, imágenes o textos creados por otras aplicaciones.

Propiedades adicionales de los campos

- Opcionalmente, un campo puede poseer las siguientes propiedades:
 - Descripción: texto breve que aclara el contenido o la finalidad del campo.
 - Tamaño: indica el tamaño máximo permitido (solo es aplicable a campos de texto o numéricos).
 - Requerido o NOT NULL: no se permiten valores nulos para dicho campo.
 - Predeterminado: se fija un valor por defecto para el campo.

Descripción de una cabecera:

- Descripción gráfica:

NIF	Nombre	Fecha	Nacionalidad	Dirección
-----	--------	-------	--------------	-----------

- Descripción completa:

- {(NIF:Texto(9),NOT NULL), (Nombre:Texto(50),NOT NULL, Descripción="Nombre y apellidos del cliente"), (Fecha:Fecha, Descripción= "Fecha de nacimiento del cliente"), (Nacionalidad:Texto(20), Predeterminado="Española"),(Direccion:Texto)}

Claves primarias y ajenas

Clave primaria de una relación

- Clave: conjunto de campos cuyos valores determinan unívocamente a cada registro de la relación. Dicho conjunto de campos debe ser minimal, esto es, ningún subconjunto propio de la clave puede actuar también como clave.
- Clave candidata: cada uno de los campos o combinaciones de campos que pueden actuar como clave de la relación.
- Clave primaria(PK=Primary Key): clave candidata elegida por el diseñador de la BD para la relación.
- ***Nota: En el modelo relacional, toda relación posee clave primaria.***

Clave primaria de una relación. Ejemplos

- Relación ALUMNO:

Nombre	NIF	Código	Fecha	Dirección
--------	-----	--------	-------	-----------

- Claves candidatas:

- (Alumno.NIF)
- (Alumno.Codigo)

- Relación CURSAR:

NIF	Asignatura	Año	Repetidor
-----	------------	-----	-----------

- Claves candidatas:

- (Cursar.NIF, Cursar.Asignatura)

Claves ajenas de una relación

- Clave ajena o secundaria (FK=Foreign Key): campo o combinación de campos de una relación (relación hija) que funciona como clave primaria de otra relación de la BD (relación referenciada o relación padre para la clave ajena).

- Relación ALUMNO (PK = Alumno.NIF):

Nombre	NIF	Código	Fecha	Dirección
--------	-----	--------	-------	-----------

- Relación CURSAR (PK = (Cursar.NIF,Cursar.Asignatura)):

NIF	Asignatura	Año	Repetidor
-----	------------	-----	-----------

- El campo Cursar.NIF es una clave ajena de la relación CURSAR y enlaza dicha relación con la relación ALUMNO.

Claves ajenas de una relación. Ejemplo

- Relación Editorial

Nombre	Dirección	Ciudad	País
LaÑ	Sol,5	Sevilla	España

- Relación Escritor

Nombre	DNI	Nacionalidad
Ana Ruíz	56234111	Chilena

- Relación Libro

Código	Título	Autor	Nombre-Ed
1256AB	Volver	56234111	LaÑ

- Relación Editorial: PK = (Nombre:Texto)
- Relación Escritor: PK = (DNI:Texto)
- Relación Libro: PK = (Codigo:Texto)
- FK = (Nombre-Ed:Texto) (→ Editorial)
- FK = (Autor:Texto) (→ Escritor)

Claves ajenas. Propiedades

- Las claves ajenas son esenciales en el Modelo Relacional, ya que permiten enlazar tablas de la BD.
- Una clave ajena y la clave primaria de la relación referenciada asociada han de estar definidas sobre los mismos dominios.
- Una relación puede poseer mas de una clave ajena (tendrá una clave ajena por cada relación referenciada de la cual dependa).
- Una relación puede no poseer ninguna clave ajena.
- Una clave ajena puede enlazar una relación consigo misma (relaciones reflexivas).

Valores nulos y restricciones de integridad

Valores nulos en el modelo Relacional

- Valor nulo (NULL): marca utilizada para representar información desconocida o no aplicable. El valor de un campo puede ser nulo por dos razones distintas:
 - Existencia de registros con ciertos campos desconocidos en ese momento.
 - Existencia de campos inaplicables a ciertos registros.
- Ejemplo: Relación OBRA

Código	Título	Tipo	Editorial	Año
123A	La huida	Libro	LaÑ	2002
678V	El infinito	Libro	NULL	NULL
564B	Azul	Cuadro	NULL	1975

- Los valores nulos del registro '678V' lo son por información desconocida, mientras que el valor nulo del registro '564B' representa un campo no aplicable (un cuadro no posee editorial).

Restricciones de integridad (I)

- **Integridad de entidad:**

- Diremos que una relación cumple la restricción de integridad de entidad si ningún campo que forme parte de la clave primaria de la relación puede tomar valores nulos.
- Nota: Para conseguir la integridad de entidad, basta declarar como 'Requerido' todos los campos que formen parte de la PK de cada relación de la BD.
- Por convenio, fijamos que cualquier campo que forme parte de una PK posee la propiedad adicional 'Requerido' y no será necesario declararlo explícitamente.

Restricciones de integridad (II)

- Si una relación R1 posee una clave ajena que la enlaza con la relación padre R2, entonces diremos que cumple la restricción de integridad referencial si todo valor de dicha clave ajena de R1:
 - coincide con algún valor de la clave primaria en la relación R2; o bien
 - toma el valor nulo (NULL).

Restricciones de integridad.

Ejemplo

- Relación ESCRITOR, PK = (DNI:Texto).

DNI	Nombre	Fecha	País
67543198	Luis Ruíz	1/1/1965	Chile
89564123	Ana Pérez	2/7/1977	España

- Relación OBRA, PK = (Código:Texto).

Código	Título	Autor	Fecha
345	La huida	67543198	1993
111	El fin	33567900	1982
654	NULL	NULL	2001

- FK = (Autor:Texto) (\rightarrow Escritor)
- La BD anterior NO cumple la restricción de integridad referencial. El valor del campo Autor del segundo registro de la tabla Obra (33567900) NO se corresponde con ningún valor del campo DNI de la tabla Escritor.

¿Como mantener la integridad referencial? (I)

- La relación R1 esta enlazada con la relación padre R2 mediante una clave ajena C. Para mantener la integridad referencial...
- Inserción:
 - El SGBD sólo permite insertar un nuevo registro en la relación R1 cuando el valor del campo C para ese registro coincida con algún valor de la PK de R2 que aparezca en la relación.

¿Como mantener la integridad referencial? (II)

- La relación R1 esta enlazada con la relación padre R2 mediante una clave ajena C. Para mantener la integridad referencial...
- Borrado (eliminación en cascada):
 - Si eliminamos un registro de la relación padre R2, el SGBD elimina automáticamente todos los registros de la relación R1 que están relacionados con dicho registro.

¿Como mantener la integridad referencial? (III)

- La relación R1 esta enlazada con la relación padre R2 mediante una clave ajena C. Para mantener la integridad referencial...
- Modificación (actualización en cascada):
 - Si modificamos el valor de la PK de un registro de la relación padre R2, el SGBD modifica automáticamente dicho valor en todos los registros de la relación R1 que estén relacionados con el.

Paso del DER al esquema relacional

Paso del DER al Esquema Relacional

- Existe un procedimiento general para obtener el Esquema Relacional (conjunto de Tablas Relacionales) a partir del Diagrama Entidad/Relación.
- Veremos como tratar...
 - entidades,
 - relaciones, y
 - jerarquías de generalización.

Entidades fuertes

- Por cada entidad fuerte del diagrama E-R, se creará una nueva tabla en el esquema relacional con tantos campos como atributos posea la entidad.
- La PK de la tabla creada es la misma que la PK de la entidad.
- Ejemplo: La entidad fuerte Alumno(DNI,Nombre,Dirección,Fecha) genera la tabla Alumno definida por:

DNI	Nombre	Dirección	Fecha
-----	--------	-----------	-------

- PK = DNI

Entidades Débiles

- **Débiles en Existencia:**

- Se tratan como entidades fuertes.

- **Débiles en Identificación:**

- Se creara una nueva tabla con los campos: un campo por cada atributo de la entidad, y se añaden los campos que forman la PK de la entidad padre de la cual depende.

$$PK = \left\{ \begin{array}{l} \text{discriminador de la entidad débil} + \\ \text{PK de la entidad padre} \end{array} \right.$$

- Se añade además una clave ajena a la tabla:
 - FK=PK de la entidad padre(→ Relación padre)

Entidades Débiles. Ejemplo

- Cuenta(Código,Titular,Fecha,Saldo)
- Operación(Número,Descripción,Cantidad)
- La entidad fuerte Cuenta genera la tabla:

Código	Titular	Fecha	Saldo
--------	---------	-------	-------

- PK = Código
- La entidad débil en identificación Operación genera la tabla:

Número	Código-cuenta	Descripción	Cantidad
--------	---------------	-------------	----------

- PK = (Número,Código-Cuenta)
- FK = (Código-Cuenta) (→ Cuenta)

Relaciones (I)

- Suponemos que R asocia las entidades E1,E2. La relación se tratara de forma distinta según el tipo: (N:M), (1:N), (1:1).
- Relaciones de tipo (N:M)
 - Creamos una nueva tabla con los siguientes campos:
 - los campos de la PK de la entidad E1,
 - los campos de la PK de la entidad E2,
 - los campos correspondientes a los atributos propios de la relación (si los hubiese).
 - $PK \supseteq (PK \text{ de } E1) + (PK \text{ de } E2)$
 - Se añaden dos claves ajenas a la nueva tabla:
 - FK = PK de E1 (\rightarrow Relación E1)
 - FK = PK de E2 (\rightarrow Relación E2)

Relaciones (II)

- Relaciones de tipo (1:N)
 - NO se creara ninguna tabla nueva. En su lugar, modificaremos la tabla asociada a la entidad que participa con cardinalidad máxima "muchos". Suponemos que E1 participa con cardinalidad (1,n).
- Modificamos la tabla asociada a la entidad E1 como sigue:
 - añadimos los campos que forman la PK de la entidad E2,
 - añadimos los campos correspondientes a los atributos propios de la relación (si los hubiese),
 - añadimos una nueva clave ajena:
 - FK = PK de la entidad E2 (→ Relación E2)

Relaciones (III)

- Relaciones de tipo (1:1)
 - NO se creara ninguna relación nueva. Se tratan como las relaciones (1:N). Puesto que las dos entidades participan con cardinalidad (1,1), tenemos dos opciones:
 - añadir a la tabla asociada a E1 la PK de E2 y los atributos propios de la relación (1:1), o bien
 - añadir a la tabla asociada a E2 la PK de E1 y los atributos propios de la relación (1:1).
- Nota: Si una entidad participa con cardinalidad (1,1) y la otra con cardinalidad (0,1), optaremos por modificar la tabla correspondiente a la entidad que participa con cardinalidad (1,1). (Ventaja: Evitamos valores nulos).

Relaciones especiales (I)

- Relaciones débiles en identificación
 - No se creará ninguna tabla nueva ni se añadirán claves ajenas.
 - Basta añadir los atributos propios de la relación débil en identificación (si los hubiese) a la tabla previamente creada para la entidad débil.

Relaciones especiales (II)

- **Relaciones reflexivas**

- Tipo (N:M): se creará una nueva tabla siguiendo las instrucciones anteriores, pero la PK de la entidad que participa aparecerá dos veces (con nombres distintos según el rol con el que participe en la relación reflexiva).
- Tipo (1:N): NO se creará una nueva tabla. Se tratarán como se describió anteriormente. Ahora bien, en la tabla asociada a la entidad que participa en la relación reflexiva aparecerá dos veces su PK (con nombres distintos):
 - una vez como PK de la tabla, y
 - otra vez como FK de la tabla que la enlaza consigo misma.

Relaciones especiales (III)

- Relaciones de grado $k \geq 3$
 - Se debe analizar la relación y estudiar la mejor opción en cada caso.
- Solución General: Se trata como una relación binaria (N:M). Esto es, se crea una nueva tabla para la relación siguiendo los pasos descritos para las relaciones de tipo (N:M).
 - Ahora bien: en lugar de dos, habrá que añadir k claves ajenas en la tabla creada, la PK de la nueva tabla no tiene por que contener a la suma de las PK de las entidades participantes. Puede que haya que eliminar alguno de los campos.

Relaciones especiales (IV)

- Ejemplo: Relación Imparte entre Grupo, Asignatura, Aula y Profesor ; con atributo propio Horario.

Grupo	Cod-asig	Cod-aula	NIF-Prof	Horario
-------	----------	----------	----------	---------

- PK = (Cod-asig, Grupo)
- Claves ajenas:
 - FK = Cod-asig (\rightarrow Asignatura)
 - FK = Grupo (\rightarrow Grupo)
 - FK = Cod-aula (\rightarrow Aula)
 - FK = NIF-Prof (\rightarrow Profesor)

Jerarquías de generalización/especialización

- No existe una solución general para el paso de una jerarquía de generalización en el Diagrama Entidad/Relación a un conjunto de tablas en el Diagrama Relacional.
- Hay que analizar las ventajas e inconvenientes en cada caso.
- Propondremos tres soluciones:
 - Tabla única.
 - Orientada a Objetos.
 - Directo del Diagrama E/R.

Jerarquías. Tabla única

- Se crea una única tabla para representar la jerarquía con las siguientes características:
 - Nombre: nombre de la entidad padre.
 - Clave primaria: PK de la entidad padre
- Campos:
 - atributos de la entidad padre,
 - la unión de los atributos de los subtipos; y
 - un nuevo campo Tipo para indicar a que subtipo de la jerarquía pertenece cada registro.
- Inconvenientes:
 - Aparición de muchos valores nulos.
 - Pérdida de información si existen en el DER relaciones en las que no participa la entidad padre sino un cierto subtipo.

Jerarquías. Orientada a Objetos

- Se añade una nueva tabla por cada subtipo y se consideran que son entidades distintas (no es necesario incluir una tabla para la entidad padre a menos que la jerarquía sea parcial).
- Para cada subtipo, su tabla asociada contendrá los campos:
 - atributos de la entidad padre (la PK de la entidad padre será la PK de la tabla)
 - atributos propios del subtipo en cuestión.
- Inconvenientes:
 - Una relación del DER en la que participa la entidad padre ha de clonarse para cada subtipo (aparecen muchas tablas).
 - Información redundante (los atributos de la entidad padre se repiten para cada subtipo de la jerarquía).
 - Jerarquías solapadas.

Jerarquías. Directo del DER

- Solución intermedia.
 - Se añaden nuevas tablas para la entidad padre y los subtipos, y se relacionan mediante claves ajenas.
- Tabla entidad padre:
 - Campos: atributos de la entidad padre.
 - PK = PK de la entidad padre.
- Tabla para cada subtipo:
 - Campos: atributos del subtipo + PK de la entidad padre.
 - PK = PK de la entidad padre.
 - FK = PK entidad padre(→ Tabla entidad Padre).
- Inconvenientes:
 - Se repiten registros. Cada registro de la jerarquía aparece dos veces: una en la tabla padre y otra en el subtipo correspondiente.
 - Muchas claves ajenas. Puede ralentizar las consultas en la BD.

Bibliografía

- Diseño de Bases de Datos Relacionales, Adoración de Miguel, Mario Piattini, Esperanza Marcos, RA-MA Editorial (1999).
- Apuntes de Bases de Datos, Mercedes Marqués, Universidad Jaume I en Castellón (2011).
<http://www.uji.es/bin/publ/edicions/bdatos.pdf>
- Fundamentos de Sistemas de Bases de Datos, 5ta Edición – Ramez Elmasri & Shamkant B. Navathe (2011).