

MODELO LÓGICO

Modelo Relacional (MR)

Departamento de Sistemas
Facultad de Electrónica y Telecomunicaciones
Universidad del Cauca



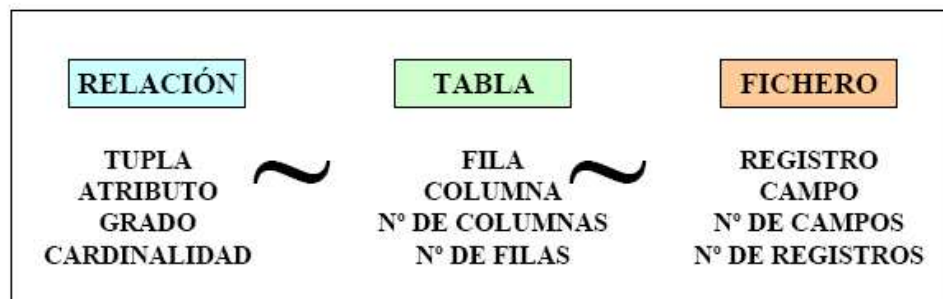
1

MODELO RELACIONAL



DEFINICIÓN:

- Presentado por Codd en 1970
- Conjunto de relaciones (tablas) con un nombre exclusivo.



**Modelo
Relacional**
(teoría)

**SGBD
Relacionales**
(implementación)

**Sistemas
de Ficheros
Clásicos**

MODELO RELACIONAL



RESEÑA HISTORICA:

En 1970 **Codd** publicó en ACM un trabajo proponiendo un nuevo MD que perseguía una serie de objetivos:

- **Independencia física:** El modo cómo se almacenan los datos no debe influir en su manipulación lógica y, por tanto, los usuarios que acceden a esos datos no han de modificar sus programas por cambios en el almacenamiento físico.
- **Independencia lógica:** Añadir, eliminar o modificar cualquier elemento de la BD no debe repercutir en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).
- **Flexibilidad:** Ofrecer a cada usuario los datos de la forma más adecuada a la correspondiente aplicación.
- **Uniformidad:** Las estructuras lógicas de los datos presentan un aspecto uniforme (tablas), lo que facilita la concepción y manipulación de la BD por parte de los usuarios.
- **Sencillez:** Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo relacional (MR) sea fácil de comprender y de utilizar por parte del usuario final.

MODELO RELACIONAL



RESEÑA HISTORICA:

Codd concedió mucha importancia al tema de la **independencia** de la representación lógica de los **datos** respecto a su almacenamiento interno, que concretó en tres tipos de independencia:

- de **ordenación**,
 - de **indización**, y
 - de los **caminos de acceso**.
- Importancia que Codd manifiesta explícitamente:
"... se propone un modelo relacional de datos como una base para proteger a los usuarios de sistemas de datos formateados de los cambios que potencialmente pueden alterar la representación de los datos, causados por el crecimiento del banco de datos y por los cambios en los caminos de acceso".

MODELO RELACIONAL



PRINCIPALES AVANCES:

Los avances más importantes que el MD relacional incorpora respecto a los MD anteriores fueron:

- **Sencillez y uniformidad:** Los usuarios ven la base de datos relacional como una colección de tablas, y al ser la tabla la estructura fundamental del modelo, éste goza de una gran uniformidad, lo que unido a unos lenguajes no navegacionales y muy orientados al usuario final, da como resultado la sencillez de los sistemas relacionales.
- **Sólida fundamentación teórica:** Al estar el modelo definido con rigor matemático, el diseño y la evaluación del mismo puede realizarse por métodos sistemáticos basados en abstracciones.
- **Independencia de la interfaz de usuario:** los lenguajes relacionales, al manipular conjuntos de registros, proporcionan una gran independencia respecto a la forma en la que los datos están almacenados.

MODELO RELACIONAL



EXITOS

Las ventajas citadas han contribuido a que desde mediados de los años 80, el MR sea utilizado por prácticamente la totalidad de los **SGBD comerciales**. Este éxito se refleja en:

- Algunas de las principales empresas informáticas del mundo, son en origen, empresas de SGBD: **ORACLE**, Sybase, INFORMIX, ...
- Los grandes fabricantes de software tienen “su” SGBD relacional: IBM DB2, Microsoft SQL Server, ...
- Existen bastantes SGBD diseñados para PC's y usuarios no expertos: Microsoft Access, etc.
- El tremendo éxito real del MR ha supuesto que el **cambio** tecnológico a la siguiente generación esté siendo **evolutivo** y no revolucionario:
Triunfan los SGBD Objeto-Relacionales, y
Fracasan, en general, los SGBD de Objetos puros.

MODELO RELACIONAL



GENERACIONES EN LA TECNOLOGIA DE BD:

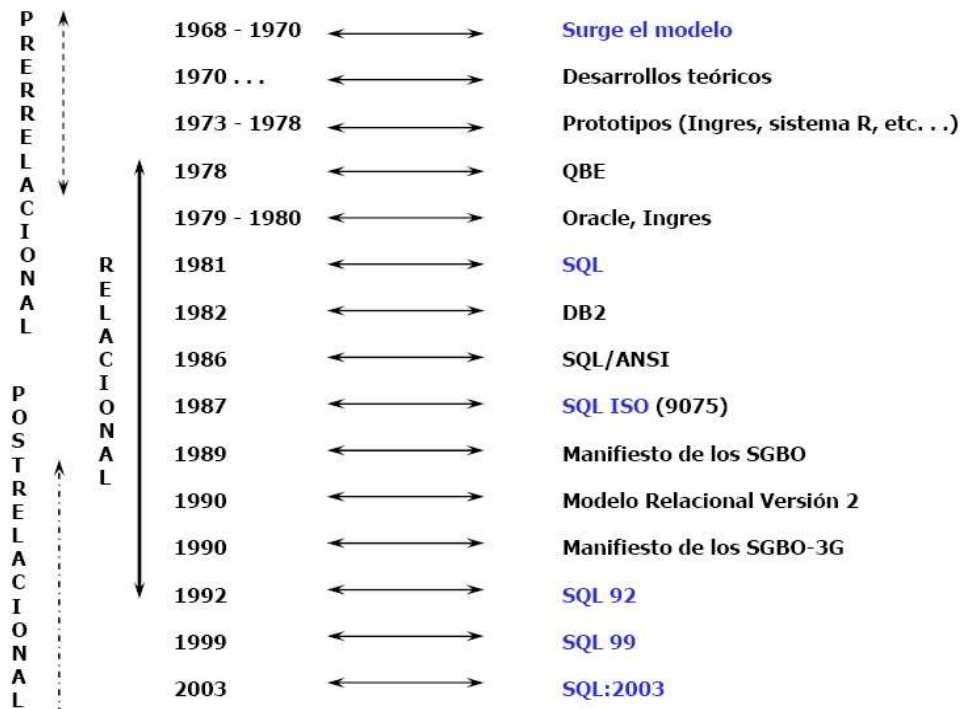
La aparición del MR representa un hito en el desarrollo de las BD, ya que ha marcado tres etapas diferentes, conocidas como **generaciones de los SGBD's**:

- **Prerrelacional (primera generación)**, en la cual los SGBD se soportan en los modelos Codasyl (en red) y Jerárquico.
- **Relacional (segunda generación)**, donde los sistemas relacionales se van aproximando a su madurez y los productos basados en este modelo van desplazando poco a poco a los sistemas de primera generación, hasta conseguir una mayor cuota en el mercado de las bases de datos.
- **Postrelacional (tercera generación)**, en la que aparecen otros MD, en especial los orientados al objeto, que están en estos momentos intentando abrirse un hueco en el mercado de las bases de datos e integrándose como extensiones en los SGBD's previos de la generación relacional.

MODELO RELACIONAL



EVOLUCION DEL MODELO RELACIONAL:



MODELO RELACIONAL



ESQUEMA DE UNA RELACION:

NombreRelación = (Atributo1, Atributo2, AtributoN)

ESQUEMA DE LA BD:

Es el conjunto de esquemas de una relación.

CONDICIONES DE LAS RELACIONES:

- No pueden existir tuplas iguales.
- El orden de las tuplas no es significativo.
- El orden de los atributos no es significativo.
- Cada atributo (campo) es univaluado y no contienen estructuras.

ESTRUCTURA DEL MODELO RELACIONAL



COMPONENTES ESTÁTICOS:

PERMITIDOS: Dominios, Relaciones (Tablas), Atributos

NO PERMITIDOS: Restricciones Inherentes y Semánticos

COMPONENTE DINÁMICO:

Álgebra Relacional
Cálculo Relacional
SQL

COMPONENTES ESTÁTICOS: Permitidos



DOMINIO:

- Es un conjunto de valores finito y todos del mismo tipo
- Se define por: Extensión (dando los posibles valores), Intensión (como un rango de valores)
- Un dominio tiene existencia propia y no depende de ninguna tabla.
- A veces se le asocia una unidad de medida (Kilos, metros, etc) y restricciones (como un rango de valores)

ATRIBUTO:

- Un atributo representa una propiedad de una tabla, y no existe sin ella.
- Un atributo toma valores de un dominio, donde el dominio pertenezca o no a la misma tabla.

RELACION:

- **GRADO:** Número de atributos de la relación.
- **CARDINALIDAD:** Numero de tuplas que participan en una relación. Varía con el tiempo.

COMPONENTES ESTÁTICOS: No Permitidos



RESTRICCIONES INHERENTES:

- Ponen limitaciones al modelar el mundo real, ya que solo utiliza un elemento (las relaciones, no se diferencia entre objetos y asociaciones).
- No pueden existir dos tuplas iguales → obligatoriedad de una clave primaria o identificador.
- Prohibido que en el cruce de una fila y columna haya más de un valor.
- Integridad de Identidad: Todas las tuplas de una relación sean distintas → "Ningún atributo que forme parte de la clave primaria puede tomar valores nulos"

COMPONENTES ESTÁTICOS: No Permitidos



RESTRICCIONES SEMÁNTICAS (INTEGRIDAD):

- Restricciones que define el diseñador, para que el esquema sea lo mas fiel posible al mundo real.

- El SGBD debe permitir:

En la fase de definición:

- Describir las restricciones con precisión y sencillez.
- Indicar las acciones ante un posible violación de una restricción.
- Verificar la consistencia de las restricciones entre sí mismas.

En la fase de manipulación:

- Comprobar que las actualizaciones cumplen las restricciones de integridad
- Poner en marcha las acciones indicadas en el caso de que las restricciones no se cumplan.

RESTRICCIONES SEMÁNTICAS: Según los elementos a los que afecta la condición



Generalmente, la acción es el rechazo de la operación.

RESTRICCIÓN INTERRELACIÓN SOBRE ATRIBUTOS:

- **UNICIDAD:** Los valores de uno o varios atributos no pueden repetirse en las distintas tuplas de una relación » UNIQUE
- **OBLIGATORIEDAD:** No se permite que un atributo sea Nulo (llave primaria) » NOT NULL

RESTRICCIÓN INTERRELACIÓN SOBRE TUPLAS: Hace referencia a las condiciones de cardinalidad. Revisar que una relación no sobrepasa un determinado valor.

RESTRICCIÓN INTERRELACIÓN SOBRE DOMINIOS:
Hace referencia a los dominios involucrados en la relación.

RESTRICCIONES SEMÁNTICAS:

Restricciones de Condición y Acción Especifica

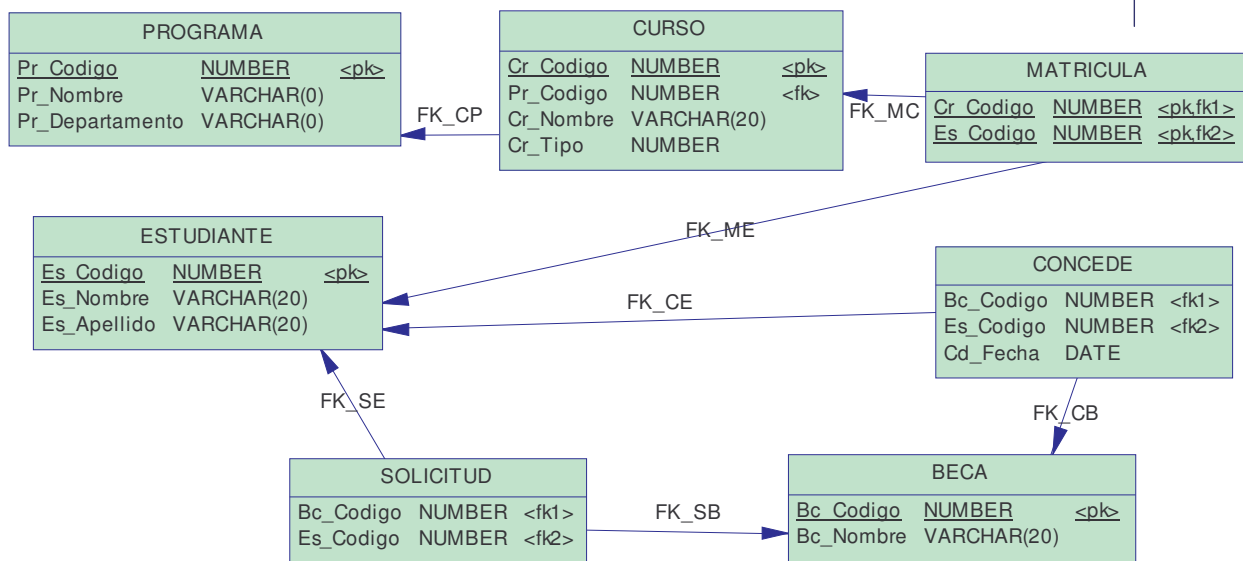


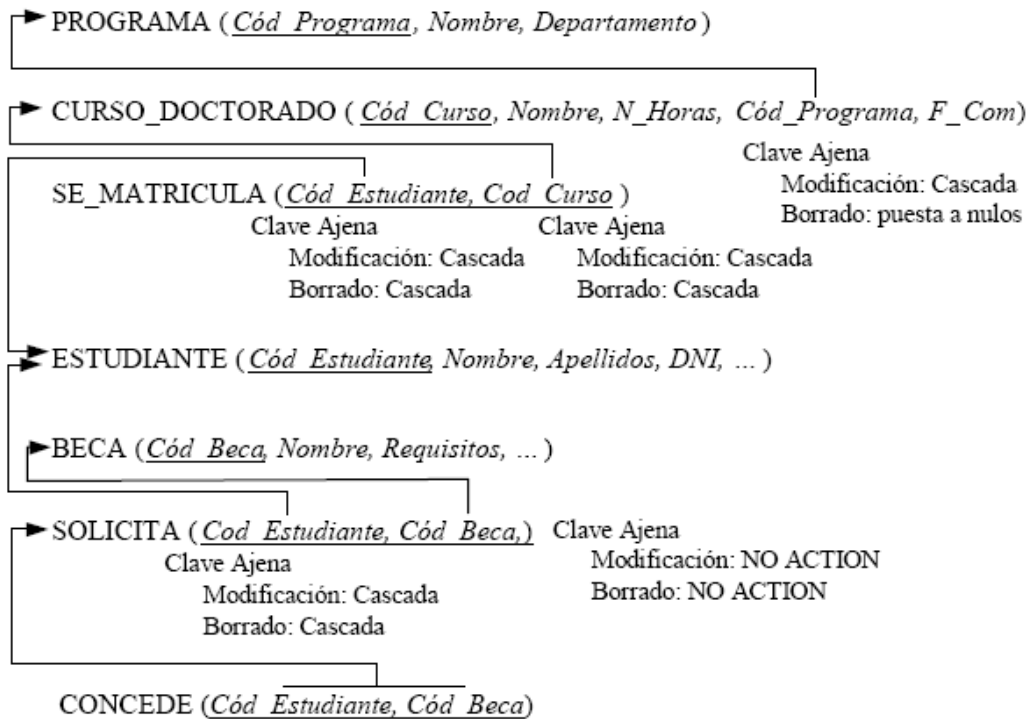
LLAVES CANDIDATAS: Atributos no vacíos que identifican cada tupla de una relación. La acción es de rechazo.

- **LLAVE PRIMARIA:** Identifica las tuplas de tabla (Restricción de identidad) » PRIMARY KEY
- **LLAVE ALTERNATIVA:** Claves candidatas que no fueron elegidas llaves primarias.

LLAVES AJENAS: Atributos cuyos valores hacen referencia a la llave primaria de otra tabla o son Nulos (Integridad Referencial). La acción no es de rechazo en todos los casos:

- NO ACTION: Rechazar la operación. Es por defecto.
- CASCADE: Borrar o modificar las tuplas de las tablas que referencia.
- SET NULL: Poner en Nulo las tuplas de las tablas que referencia.
- SET DEFAULT: Poner el valor que se ha definido por defecto las tuplas de las tablas que referencia.





RESTRICCIONES SEMÁNTICAS: Restricción de Condición General y de Acción Específica (Rechazo)



Las dos restricciones se diferencian en la cantidad de relaciones que afectan.

RESTRICCIONES DE VERIFICACIÓN: Condición que afecta solo a un atributo de la tabla » CHECK
Ejemplo: Curso con mínimo 30 horas. CHECK N_horas >= 30

RESTRICCIÓN DE ASERCIÓN: Condición que afecta a uno o varios atributo de una o varias tablas. Tienen nombre
Ejemplo: Solo se puede conceder una beca a uno de los estudiantes que ha solicitado dicha beca (restricción de inclusión entre Concede y solicita)

```
CREATE ASSERTION CONCEDE_SOLICITA AS
CHECK(SELECT Cod_Estudiante, Cod_Beca
FROM CONCEDE) IN
(SELECT Cod_Estudiante, Cod_Beca
FROM SOLICITA));
```

Impide que cualquier actualización deje en la relación concede tuplas que no estaban en solicita

RESTRICCIONES SEMÁNTICAS:

Restricción de Condición General y de Acción General



Se utilizan cuando no se puede definir una acción específica para una restricción. Procedimentales

DISPARADORES (TRIGGERS) : Se dispara cuando una condición es verdadera. Ejm: Solo se pueden asignar hasta 50 becas

```
CREATE TRIGGER Comprobar_Matriculados
AFTER INSERT ON SOLICITA
DECLARE
  NroSolicitudes NUMBER;
BEGIN
  SELECT COUNT(*) INTO NroSolicitudes FROM SOLICITA;
  IF NroSolicitudes > 50 THEN
    INSERT INTO MENSAJES VALUES ('Hay más de 50
    solicitudes');
  END IF;
END Comprobar_Matriculados;
```

PROCEDIMIENTO ALMACENADO

GENERACIÓN DE UN MODELO LÓGICO ESTANDAR



REGLAS DE TRANSFORMACION GENERICAS:

- Cada entidad e interrelación de un MER se convierte en una **Relación**.
- Cada Relación tiene varias columnas, cada una con nombre único.
- La pérdida de semántica se reconstruye con las restricciones de Integridad en el MR.

ENTIDADES FUERTES:

Una entidad E con N atributos, se convierte en una relación E con N campos (Columnas).

ENTIDADES DEBILES:

Se crea una relación a la cual se le adicionan los campos que forman la llave primaria de la entidad de la cual pertenece.



GRUPO
Codigo
CodGru
CantMax

ATRIBUTOS COMPUESTOS UNIVALUADO:

Se convierten en campos adicionales a la relación todos los componentes del atributo.



ATRIBUTOS COMPUESTOS MULTIVALUADOS:

Se crea una nueva relación con los atributos correspondientes si es compuesto. Se le adiciona un campo que es la llave primaria de la relación a la que pertenece.

DOMINIOS:

- Si es de valores que pueden cambiar en el tiempo, se crea una entidad agregando un campo que se convierte en llave primaria de la misma (una secuencia).
- En caso contrario, se convierte en una condición CHECK (Restricción de Verificación).

LLAVES CANDIDATAS:

Se convierten en campos no NULOS o UNIQUE.

INTERRELACIONES 1-1:

Pasar todos los atributos de la relación y la llave primaria de una de las entidades a la otra entidad.

- Si es de (0,1) a (0,1) se crea una relación con el nombre de la interrelación en el MER para evitar nulos, propagando ambas llaves a esta.
- Si es de (0,1) a (1,1) propagar la llave de (1,1) a (0,1).
- Si es de (1,1) a (1,1) es indiferente.



INTERRELACIONES 1-N:

Se puede pasar la llave primaria de la entidad con cardinalidad 1 a la entidad con cardinalidad N, los atributos de la interrelación pasan a la entidad con cardinalidad N.

INTERRELACIONES N-M:

Se crea una nueva relación con las llaves primarias de las entidades participantes, así esta interrelación no tenga atributos.

INTERRELACIONES RECURSIVAS:

Se adiciona un nuevo campo con la llave primaria de la entidad que participa en la interrelación. Se debe de cambiar el nombre del campo de acuerdo al rol correspondiente.



INTERRELACIONES N-ARIAS:

Se crea una nueva relación con el nombre de la interrelación con todas las llaves primarias de todas las entidades participantes y los atributos de la interrelación.

GENERALIZACIÓN:

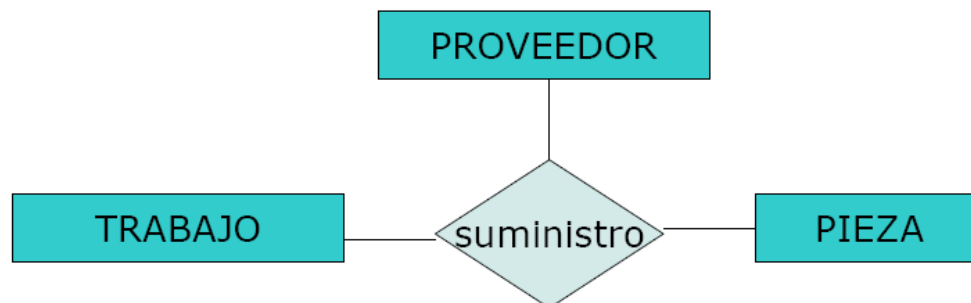
-Si es Solapada y Parcial: Se crea una nueva relación para el supertipo y su llave primaria pasa a los subtipos.

-Si es total: Se crean relaciones de las entidades subtipo, donde cada una tiene los atributos del supertipo. No se crea una relación para el supertipo.

AGREGACION:

Se toma la llave primaria de las entidades compuestas y la llave primaria de la relación de la entidades componentes y se agregan a una nueva relación.

Ejemplos

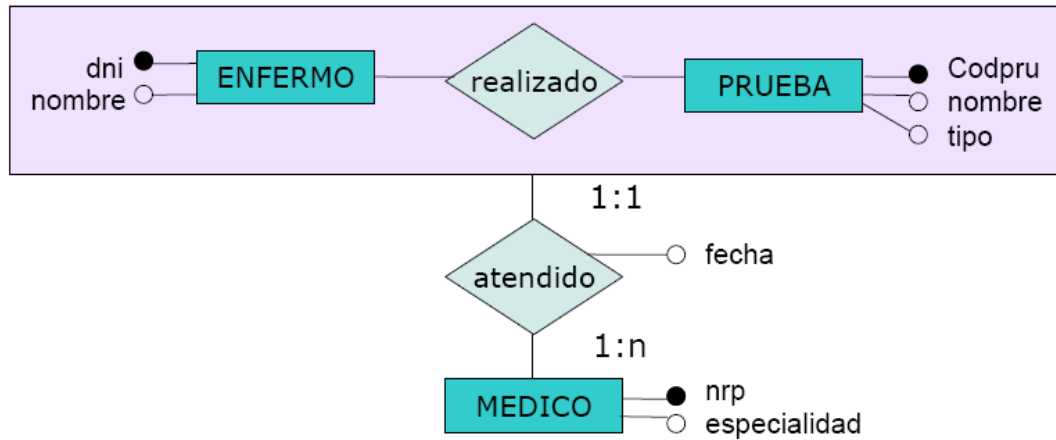


N:M:S SUMINISTRO (CodProv, CodPieza, CodTrab)

N:M:1 SUMINISTRO (CodProv, CodPieza, CodTrab)

N:1:1 SUM1 (CodPieza, CodProv)
SUM2 (CodPieza, CodTrab)

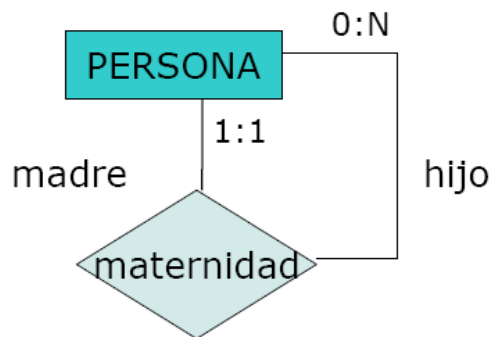
Ejemplos



REALIZADO(dni,codpru)

ATENDIDO(dni,codpru,nrp, fecha)

Ejemplos



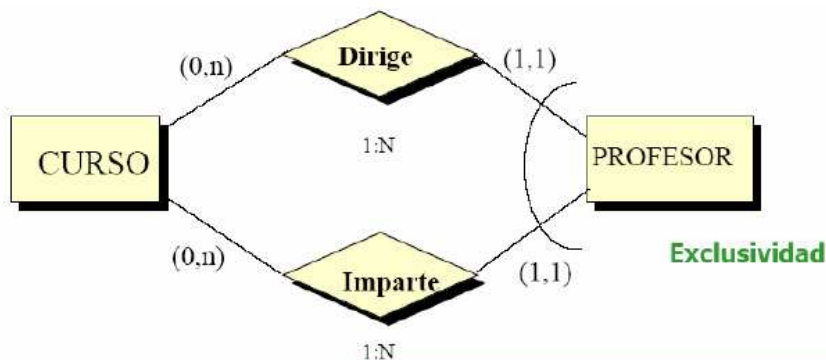
Borrado: null
Modificación: cascada

a) persona (Codper, nombre, ..., codper_madre)

b) persona (Codper, nombre, ...)
madre (Codper, codper_madre)

Borrado y
Modificación: cascada

Ejemplos

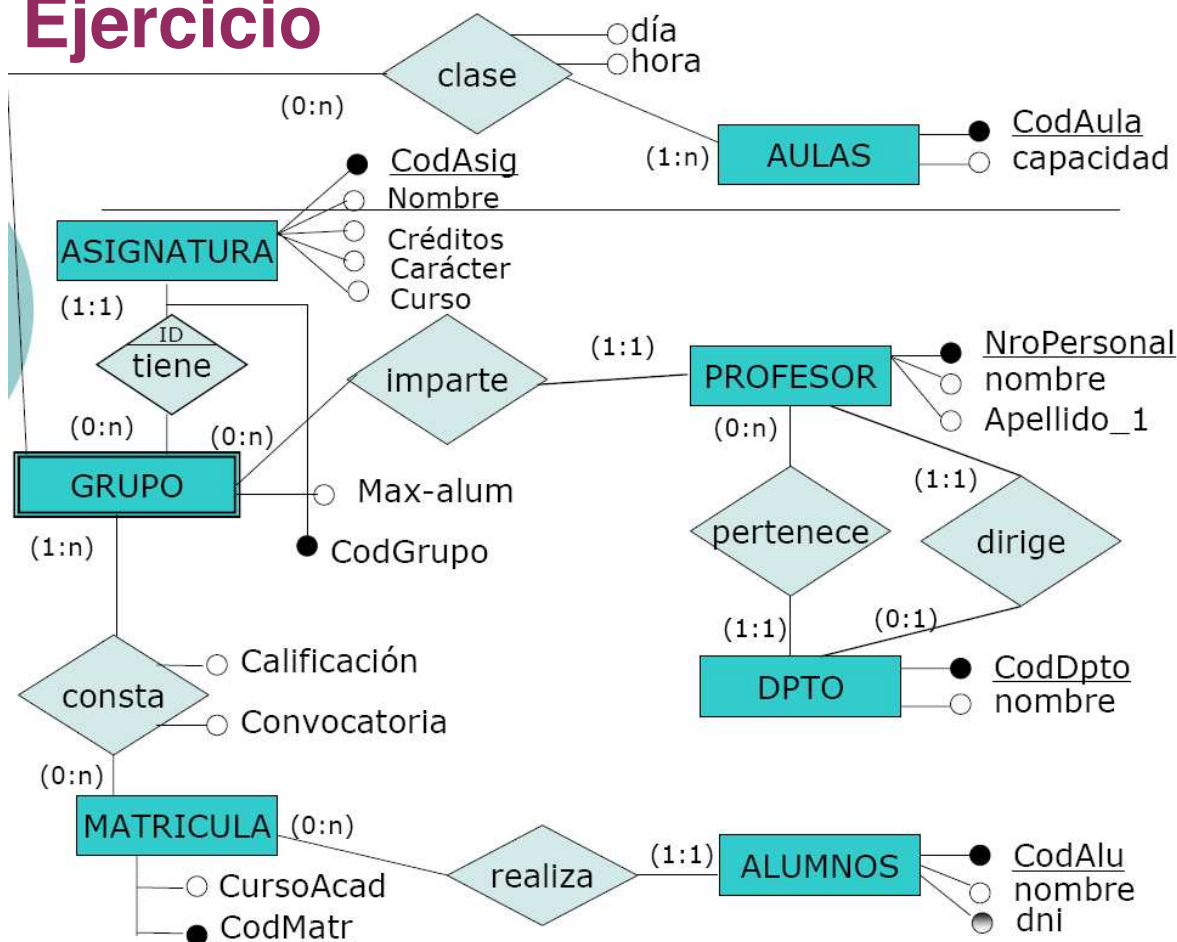


```
CREATE TABLE Curso (
    Cod_Curso Codigos_Cursos,
    Nombre Nombres, ...,
    Cod_prof_dirige Cods_profesores,
    Cod_prof_imparte Cods_profesores,
    PRIMARY KEY (Cod_Curso)
    FOREIGN KEY (Cod_prof_dirige) REFERENCES Profesor
        ON UPDATE CASCADE
    FOREIGN KEY (Cod_prof_imparte) REFERENCES Profesor
        ON UPDATE CASCADE
    CHECK (( Cod_prof_dirige NOT IN (SELECT Cod_prof_imparte FROM Curso))
        AND
        ( Cod_prof_imparte NOT IN (SELECT Cod_prof_dirige FROM Curso)));
```



Exclusividad

Ejercicio





Esquema obtenido

Asignatura (CodAsig, nombre, créditos, carácter, curso)

Alumno (CodAlu, nombre, dni)

Aula (CodAula, capacidad)

Profesor (NroPersonal, nombre, apellido_1)

Dpto (CodDpto, nombre)

Matricula (CodMatr, cursoacad)

Grupo (CodAsig, CodGrp, max_alum)

CONSTA (CodMatr, CodAsig, CodGrupo, Convocatoria, calificación)

PERTENECE (NroPersonal, CodDpto)

DIRIGE (NroPersonal, CodDpto)

IMPARTE (CodAsig, CodGrupo, NroPersonal)

CLASE (CodAsig, CodGrupo, CodAula, hora, dia)

REALIZA (NumMatr, CodAlum)

Esquema con reducción de tablas



Asignatura (CodAsig, nombre, credits, carácter, curso)

Alumno (CodAlu, nombre, dni)

Aula (CodAula, capacidad)

Profesor (NroPersonal, nombre, apellido1, CodDpto)

Profesor (NroPersonal, nombre, apellido1)

Dpto (CodDpto, nombre)

Dpto (CodDpto, nombre, CodProfDirige)

Matricula (CodMatr, cursoacad)

Matricula (CodMatr, cursoacad, CodAlu)

Grupo (CodAsig, CodGrp, max_alum)

Grupo (CodAsig, CodGrp, max_alum, NroPersonal)

CONSTA (CodMatr, CodAsig, CodGrp, Convocatoria, calificación)

PERTENECE (NroPersonal, CodDpto)

DIRIGE (NroPersonal, CodDpto)

IMPARTE (CodAsig, CodGrupo, NroPersonal)

CLASE (CodAsig, CodGrupo, CodAula, hora, dia)

REALIZA (CodMatr, CodAlu)



Esquema definitivo

Asignatura (CodAsig, nombre, créditos, carácter, curso)

Alumno (CodAlu, nombre, dni)

Aula (CodAula, capacidad)

Profesor (NroPersonal, nombre, apellido1, CodDpto)

Dpto (CodDpto, nombre, CodProfDirige)

Matricula (CodMatr, cursoacad, CodAlu)

Grupo (CodAsig, CodGrupo, max_alum, NroPersonal)

CONSTA (CodMatr, CodAsig, CodGrupo, Convocatoria, calificación)

CLASE (CodAsig, CodGrupo, CodAula, hora, dia)

TEORÍA DE LA NORMALIZACIÓN



PROBLEMAS AL PASAR EL MR AL MR:

- Incapacidad para almacenar ciertos hechos.
- Redundancias y, por tanto, posibilidad de inconsistencias.
- Ambigüedades.
- Pérdida de información.
- Pérdida de ciertas restricciones de integridad que dan lugar a interdependencias entre datos.
- Existencia de valores nulos (inaplicables).
- Aparición de estados inválidos en los procesos de manipulación.

DEFINICIÓN:

Es un método formal de análisis que ayuda a determinar si un MR está correcto y exento de errores: Encontrar fallos y corregirlos, eliminar dependencias entre atributos; evitando así introducir anomalías en las operaciones de manipulación de datos. Y, en tal caso, convertirlo es un mejor modelo: **Modelo Lógico Especifico** o **Modelo Lógico de datos Normalizado**.

RESULTADOS DE APLICAR NORMALIZACIÓN:



- Mayor eficiencia en la utilización del LDD y LMD; reduciendo las posibles anomalías:
 - Las actualizaciones pueden llevarse a cabo con un número mínimo de operaciones, minimizando la inconsistencia de los datos.
 - Reduce el espacio de almacenamiento de las relaciones de la BD, menos costes.
 - Evita que no se pueda insertar un ejemplar en una tabla si no esta relacionado con otra. (Ejm: No dejar ingresar el curso BD si este no tiene al menos un estudiante, separar el detalle de una tabla de otra)
 - Gestiona la independencia entre las tablas. En caso de borrar un ejemplar de una de las tablas, evita que se borren datos importantes de otra tabla con la que tenga relación.
 - Garantiza que con la combinación de los Esquema Relacional Resultante (ERR), no se sufra pérdida de información.

RESULTADOS DE APLICAR NORMALIZACIÓN:



- Minimización de dependencias y de los atributos que participan en ella. Sin embargo, preserva las restricciones del UD.
- Minimización del número de tablas y el número de sus atributos.
- Captar mejor la semántica del mundo real.

AUTOR	NACIONALIDAD	COD_LIBRO	TITULO	EDITORIAL	AÑO
Date, C.	Norteamericana	98987	Database	Addison	1990
Date, C.	Norteamericana	97777	SQL Stan	Addison, W.	1986
Date, C.	Norteamericana	98987	Guide for	Addison, W.	1988
Codd, E.	Norteamericana	7890	Relational	Addison, W.	1990
Gardarin	Francesa	12345	Basi Dati	Paraninfo	1986
Gardarin	Francesa	67890	Comp BD	Eyrolles	1984
Valduriez	Francesa	67890	Comp BD	Eyrolles	1984
Kim, W.	Norteamericana	11223	BD OO	ACM	1989
Lochovsky	Canadiense	11223	BD OO	ACM	1989

ESCRIBE



Redundancia: La nacionalidad del autor se repite por cada ocurrencia del mismo. Cuando un libro tiene mas de un autor, se repite la editorial y el año de publicación.

Anomalías de modificación: Inconsistencias al cambio fácil del nombre de una editorial en una tupla sin modificar el resto de las que corresponden al mismo libro.

Anomalías de inserción: Si se quiere ingresar información de algún autor, del que no hubiera ningún libro en la BD, no sería posible, ya que cod_libro es parte de la clave primaria de la relación. La inserción de un libro, que tiene dos autores obliga a insertar dos tuplas en la relación.

Anomalías de borrado: Si se quiere eliminar un cierto libro, se pierden los datos de su autor y viceversa.

PROPIEDADES PARA QUE UN Esquema Relacional Origen (ERO) SEA IGUAL AL a un Esquema Relacional Resultante (ERR) :



- Conservación de la información.
 - *Conservación de los atributos:* Unión de atributos de ERO igual a la unión de atributos de ERR.
 - *Conservación del contenido:* La combinación (join) de las RR ha de producir la RO (descomposición).
- Conservación de las dependencias (Restricciones).
- Mínima redundancia de los datos (Normalización de las relaciones)

PROCESO:

El proceso se realiza a través de sucesivas proyecciones para pasar de un Esquema Relacional Origen (ERO) a un Esquema Relacional Resultante (ERR), tal que cumpla unas determinadas condiciones, para conseguir la mejor representación del Universo de Discurso.



Antes de definir las distintas formas normales se explican, muy brevemente, algunos conceptos necesarios para su comprensión:

Dependencia funcional

Un atributo Y se dice que depende funcionalmente de otro X si, y sólo si, a cada valor de X le corresponde **UN ÚNICO** valor de Y , lo que se expresa de la siguiente forma: $X \rightarrow Y$

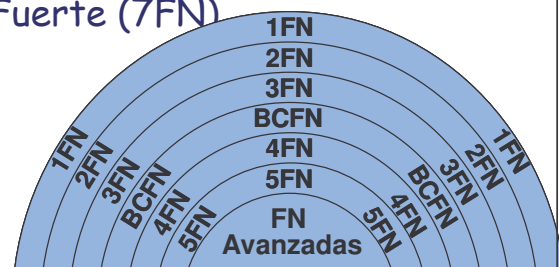
También se dice que X determina o implica a Y .

X se denomina implicante o determinante e Y es el implicado.

FORMAS NORMALES



- Se dice que un esquema relacional esta en una determinada forma si satisface un cierto conjunto específico de restricciones.
- Cuanto mas alta sea la forma normal en la que se encuentren los esquemas de relación, menores serán los problemas que aparecen en el mantenimiento de la BD.
- Inicialmente un esquema relacional esta en Forma No Normalizada (FNN), al pasar al formato de tablas.
- Formas normales:
 - Primera Forma Normal (1FN), segunda (2FN) y tercera (3FN). [Codd, 1970]
 - Forma Normal de Boyce-Codd (BCFN). [1974]
 - Cuarta Forma Normal (4FN) [Fagin, 1974, 79]
 - Forma Normal de Proyección-Unión (5FN)
 - Forma Normal de Proyección-Unión Fuerte (6FN)
 - Forma Normal de Proyección-Unión Extra Fuerte (7FN)
 - Forma Normal de Clave de Dominio.
- Cuando un esquema de relación esta en una Forma Normal, implícitamente también esta en las formas normales inferiores a ésta.



Primera Forma Normal (1FN):



- Su cumplimiento es obligatorio.
- Para que una tabla sea considerada una relación no debe de admitir grupos repetidos, es decir, un atributo sólo puede tomar un único valor de un dominio simple.

Código	Nombre	FechaNac	Curso
001	Juan	12/05/80	BD
001	Juan	12/05/80	Ing.Soft.
002	Ana	20/03/82	BD
002	Ana	20/03/82	Ing.Soft

- El esquema de una relación R esta en 1FN si los dominios de todos los atributos son atómicos. No son atómicos:

Código	Nombre	Dirección
001	Juan	Cra 23 2-45. Ciudad: Popayán Calle 9 #2-45 Ciudad: Cali
002	Ana	Calle 1 #8-12. Ciudad: Bogotá

Los atributos compuestos.

Código	Materia
IS-BD-01	Bases de Datos I
IS-BD-02	Bases de Datos II

Número de identificación codificados.

- **Solución:** Descomponer el atributo en atributos si es cardinalidad 1. Colocar los datos repetitivos junto con una copia de los atributos que forman la llave primaria, en una tabla independiente si es cardinalidad n. Continuar el proceso hasta que se terminen los grupos repetitivos.

Segunda Forma Normal (2FN):



- Debe estar en 1FN.
- Se aplica principalmente a las relaciones donde la llave primaria es compuesta.
- Cada atributo no principal (no forma parte de las claves candidatas) debe depender funcionalmente de la llave primaria.
- La 2FN **NO** se cumple cuando un atributo no principal depende funcionalmente de algún subconjunto de la llave primaria.
- Cada uno de los atributos de una entidad depende de toda la clave. Todos los atributos que no forman parte de ninguna clave candidata deben suministrar información acerca de la llave principal completa.
- **Solución:** Una vez identificados los atributos que no dependen funcionalmente de toda la clave (Dependencia Parcial), se formará con ellos una nueva entidad y se eliminarán de la antigua. La clave principal de la nueva entidad estará formada por la parte de la antigua de la que dependen funcionalmente.



PRESTAMO (CodSocio, NombreSocio, CodLibro, NombreLibro, FechaPrestamo, Editorial, País)

Claves candidatas: (CodSocio, CodLibro), (NombreSocio, CodLibro) y (CodSocio, NombreLibro)

Las dependencias funcionales son:

CodSocio → NombreSocio	No 2FN
CodLibro → NombreLibro, Editorial, País	No 2FN
CodSocio, CodLibro → FechaPrestamo	2FN

La solución es descomponer esta relación en las siguientes:

PRESTAMO (CodSocio, CodLibro, FechaPrestamo)
LIBRO (CodLibro, NombreLibro, Editorial, País)
SOCIO (CodSocio, NombreSocio)

Tercera Forma Normal (3FN):



- Debe estar en 2FN.
- No debe existir algún atributo no principal que dependa transitivamente de alguna clave de la Relación ($X \rightarrow Y$, $Y \rightarrow Z$, entonces $X \rightarrow Z$)
- La 3FN no se cumple cuando existen atributos no principales que dependen funcionalmente de otros atributos no principales.
- Para estar en 3FN, los atributos que no forman parte de ninguna clave candidata deben facilitar información sólo acerca de la(s) clave(s) y no acerca de otros atributos.
- Se encuentran en 3FN toda relación cuyos atributos son todos principales.
- **Solución:** Una vez identificados los atributos que dependen de otro atributo distinto de la clave, se formará con ellos una nueva entidad y se eliminarán de la antigua. La clave principal de la nueva entidad será el atributo del cual dependen. Este atributo en la entidad antigua, pasará a ser una clave ajena.

En **PRESTAMO**, FechaPrestamo facilita información acerca de las claves. Por lo que está en 3FN.

En **LIBRO**, País entrega información acerca de la editorial que publica el libro y no del libro, por lo que NO está en 3FN. La solución es:

LIBRO (CodLibro, CodEdit) y **EDITORIAL** (CodEdit, NomEdit, País)

Forma Normal de Boyce-Codd (BCFN)



- También conocida como Forma Normal Óptima.
- Versión algo más restrictiva de la 3FN
- Es una de las formas normales **más deseables** para un **buen diseño**.
- No siempre se puede conseguir.
- Toda relación en BCFN esta también en 3FN.
- Si $X \rightarrow Z$ (X =Determinante de Z). Una relación esta en BCFN, si y solo si todo determinante es una clave candidata.
- Una relación está en FNBC si cualquier atributo sólo facilita información sobre claves candidatas, y no sobre atributos que no formen parte de ninguna clave candidata.
- Si la clave primaria está formada por un solo atributo y está en 3FN, ya está en FNBC.
- Las relaciones que siempre están en FNBC son las binarias.

PRESTAMO (CodSocio, CodLibro, FechaPrestamo)
LIBRO (CodLibro, CodEdit)
EDITORIAL (CodEdit, NomEdit, País)
SOCIO (CodSocio, NombreSocio)



Adicionalmente se tiene:

DIRECCIÓN (CodSocio, CodPostal, Número, Ciudad)

Claves Candidatas: (CodSocio, CodPostal) y (CodSocio, Ciudad)

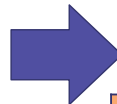
CodPostal \rightarrow Ciudad

CodSocio, CodPostal \rightarrow Número

DIRECCIÓN (CodSocio, CodPostal, Numero)

CODIGOPOSTAL (CosPostal, Ciudad)

CodSocio	CodPostal	Numero	Ciudad
001	30009	Cra 23 2-45	Cali
001	30009	Calle 9 #2-45	Cali
002	30023	Calle 1 #8-12	Popayán



CodPostal	Ciudad
30009	Cali
30023	Popayán

CodSocio	CodPostal	Numero
001	30009	Cra 23 2-45
001	30009	Calle 9 #2-45
002	30023	Calle 1 #8-12