



Tarea 3: Modelo Relacional

* Alpuche Morales Elmer Alexis 415975945
elmer.alpuche@ciencias.unam.mx
* Cova Pacheco Felipe de Jesús 312030111
felipejde.fc@ciencias.unam.mx

Profesor: Gerardo Avilés Rosas
Ayudante: José Enrique Vargas Benítez
Ayudante de Laboratorio: Rodrigo Eduardo Colín Rivera

1. Preguntas de repaso

a. *¿Qué es una relación y qué características tiene?*

R. Las relaciones de bases de datos son asociaciones entre tablas que se crean utilizando sentencias de unión para recuperar datos. Los tipos de relaciones son:

- Unívoca. Las dos tablas pueden tener sólo un registro en cada lado de la relación.
Cada valor de clave primaria se relaciona con sólo un (o ningún) registro en la tabla relacionada.
La mayoría de las relaciones unívocas están impuestas por las reglas empresariales y no fluyen con naturalidad a partir de los datos. Sin este tipo de regla, generalmente podrá combinar ambas tablas sin incumplir ninguna regla de normalización.
- Uno a varios. La tabla de claves primaria sólo contiene un registro que se relaciona con ninguno, uno o varios registros en la tabla relacionada.
- Varios a varios. Cada registro en ambas tablas puede estar relacionado con varios registros (o con ninguno) en la otra tabla. Estas relaciones requieren una tercera tabla, denominada tabla de enlace o asociación, porque los sistemas relacionales no pueden alojar directamente la relación.

b. *¿Qué es un esquema de relación?*

R. El esquema de base de datos define la estructura de una base de datos, en un lenguaje formal soportado por un sistema de gestión de base de datos (DBMS). En una base de datos relacional, el esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.

El esquema es generalmente almacenado en un diccionario de datos. Aunque generalmente el esquema es definido en un lenguaje de base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de base de datos.

c. ***¿Qué es una llave primaria?, ¿Qué es una llave candidata?, ¿Qué es una llave mínima?, ¿Qué es una súper llave?***

R.

- Llave. Es un conjunto no vacío de atributos que identifican de manera única a cada dupla.
- Llave primaria. Es una llave candidata elegida, casi siempre, arbitrariamente.
- Llave candidata. Sea R una relación, entonces una llave candidata para R es un subconjunto del conjunto de atributos de R, digamos k tal que:
 - i) Unicidad. No existe par de tuplas distintas en R con el mismo valor para k.
 - ii) Irreducibilidad. Ningún subconjunto propio de k tiene la propiedad de unicidad.

Una llave candidata que involucra más de un atributo se llama compuesta, en otro caso se llama simple.

- Llave mínima. Es el mínimo conjunto de atributos que puedes agarrar de una tabla tal que cumpla que en una fila que agarres no haya elementos repetidos (llaves duplicadas).
- Súper llave. Es cualquier conjunto de atributos cuyos valores no se repiten en tuplas distintas de la relación.

d. ***¿Qué restricciones impone una llave primaria y una llave foránea al modelo de datos relacional?***

- a. Integridad de la clave. Ningún atributo de una clave candidata puede tomar valores nulos. Lógicamente, los atributos que forman una clave candidata han de tomar siempre valores distintos para cada posible tupla.
- b. Integridad de referencia o referencial. Sea T1.a un atributo de la tabla T1 que forma parte de una clave ajena para la

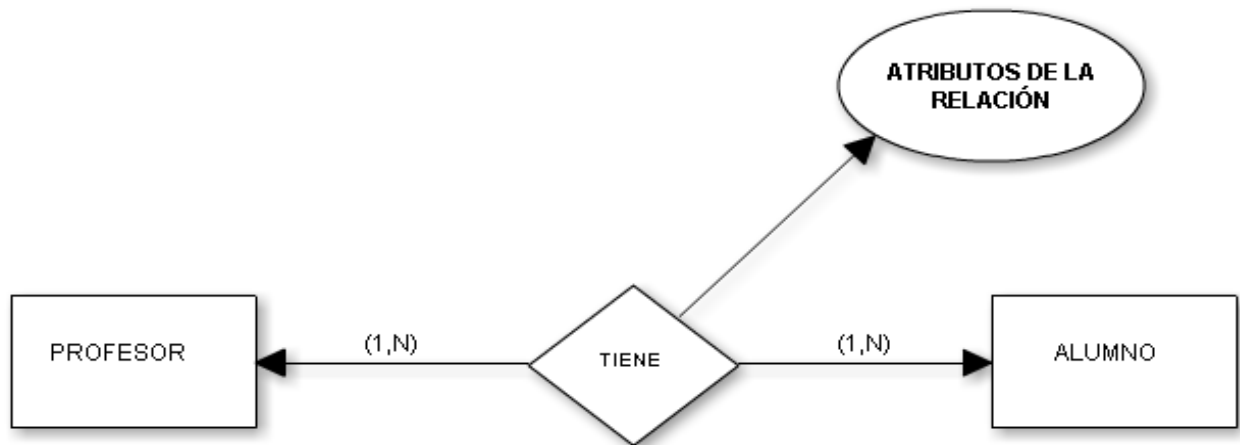
tabla T2. Es decir, que en T2 existe un atributo definido con el mismo dominio, aunque no obligatoriamente con igual nombre, y que es parte de su clave primaria. Entonces, T1.a debe ser siempre igual a algún valor ya contenido en el atributo referenciado en la tabla T2, o bien tomar un valor nulo.

- c. Otras restricciones de acuerdo con la semántica concreta del problema. Pueden ser sencillas, como la especificación de valores mínimos o máximos que puede tomar un atributo numérico, lista de valores permitidos de un atributo, o más complejas: condiciones sobre valores de los atributos en función de valores de otros atributos de esa u otras tablas.

e. Investiga como se traducen las categorías (Presentes en el modelo E/R) al modelo relacional. Da un ejemplo.

- Cada entidad corresponde a una tabla
- Cada atributo corresponde a un campo dentro de la tabla
- Se crea un atributo en una de las tablas que corresponde a la llave primaria de la otra. Esta es la llave foránea de la relación.
- En el caso de las relaciones 1:1 también pueden darse dos casos:
 - a. Si las entidades poseen cardinalidades (0,1), la relación se convierte en una tabla.
 - b. Si una de las entidades posee cardinalidad (1:1) de la tabla resultante de la entidad con cardinalidad (0,1). Si ambas entidades poseen cardinalidades (1,1) se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra.

Ejemplo:



PROFESOR	
PK	idProfesor
Nombre Apellidos Asignatura Aula	

PROFESOR_ALUMNO	
PK,FK	idProfesor
PK,FK	idAlumno
ATRIBUTOS DE LA RELACIÓN	

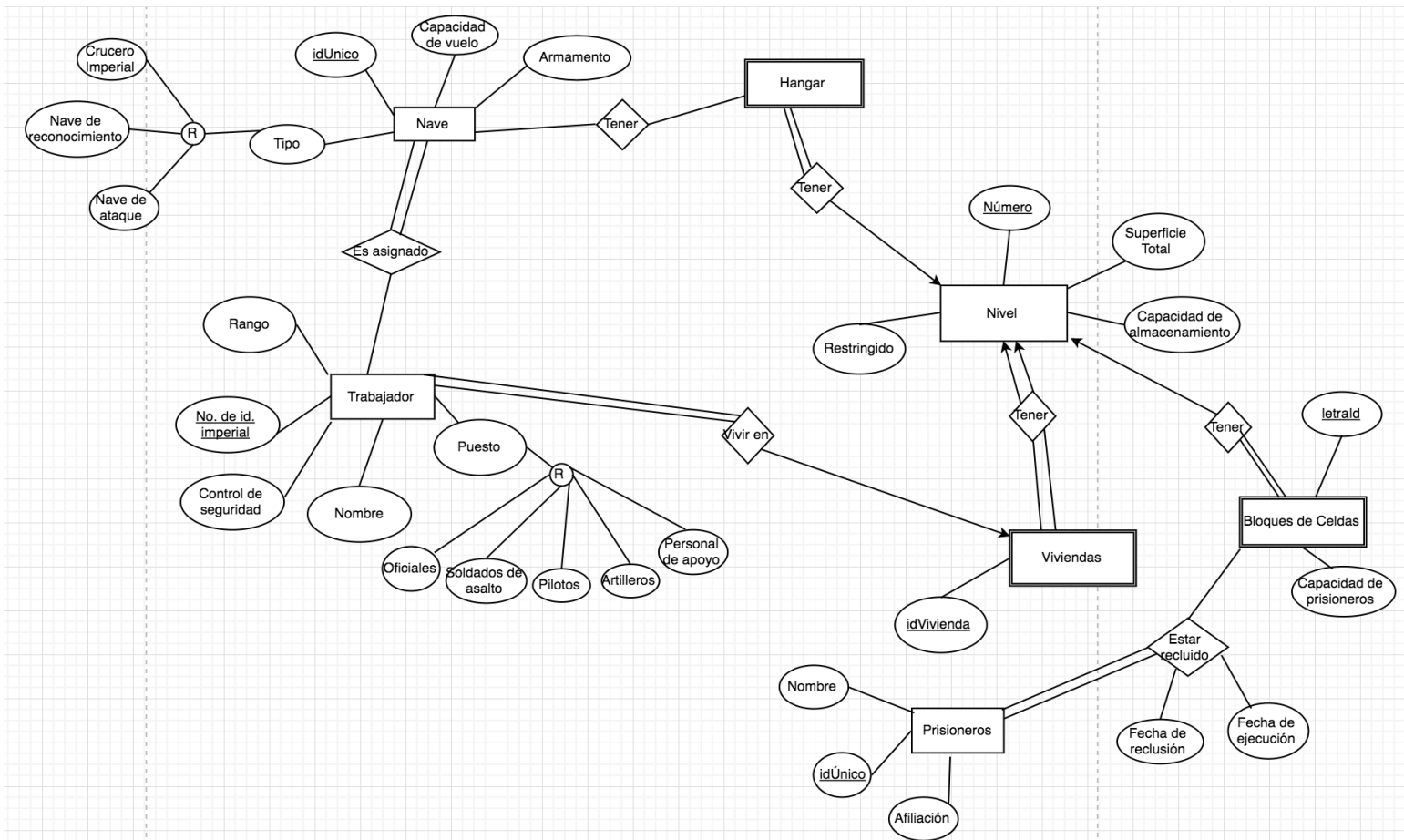
ALUMNO	
PK	DNI
Nombre Apellidos Experiencia CV	



2. Modelo relacional

- Obra (IdObra, IdExhibición, Nombre, IdMuseo, nombreArtista, Descripción, Título, Año)
- Pintura (IdObra, IdExhibición, Nombre, IdMuseo, nombreArtista, Descripción, Título, Año)
- TipoPintura (IdObra, TipoPintura, Óleo, Acuarela)
- MaterialPintura (IdObra, MaterialPintura)
- EstiloPintura (IdObra, EstiloPintura)
- Escultura (IdObra, IdExhibición, Nombre, IdMuseo, Ancho, Largo, Peso, nombreArtista, Descripción, Título, Año)
- MaterialEscultura (IdObra, MaterialEscultura)
- TipoEscultura (IdObra, TipoEscultura, Escultura, Estatua)
- Miscelanea (IdObra, IdExhibición, Nombre, IdMuseo, nombreArtista, Descripción, Título, Año)
- EstiloMiscelanea (IdObra, EstiloMiscelanea)
- TipoMiscelanea (IdObra, TipoMiscelanea, Video, Fotografía)
- Permanentes (IdObra, IdExhibición, IdMuseo, nombreArtista, Descripción, Título, Año, Costo, Fecha_Adq)
- Prestadas (IdObra, IdExhibición, IdMuseo, nombreArtista, Descripción, Título, Año, Prestador, Fecha_Pres, Fecha_Dev)
- Exhibiciones (IdExhibición, Título, Periodo)
- Artistas (Nombre, Fecha_Nac, Fecha_def, País, Época, Descripción)
- EstilosArtistas (Nombre, EstilosArtistas)
- Museo (IdMuseo, Nombre)
- DirecciónMuseo (IdMuseo, DirecciónMuseo)
- TeléfonosMuseo (IdMuseo, TeléfonosMuseo)
- Colecciones (Nombre, IdMuseo, Descripción, Contacto)
- TipoColecciones (Nombre, TipoColecciones, Personal, Museo)
- DirecciónColecciones (Nombre, DirecciónColecciones)
- TipoTeléfono (Nombre, TipoTeléfono)

3. Modelo relacional



- Trabajador (**No. de Id. Imperial**, **Id Único**, Rango, Control de Seguridad, Nombre, Capacidad de vuelo, Armamento)
- PuestoTrabajador (**No. de Id. Imperial**, **Puesto Trabajador**, Oficiales, Soldados de asalto, Pilotos, Artilleros, Personal de apoyo)
- Nave (**IdÚnico**, Capacidad de Vuelo, Armamento)
- TipoNave(**IdÚnico**, **TipoNave**, Crucero Imperial, Nave de Reconocimiento, Nave de Ataque)
- Hangar (**NúmeroNivel**, **IdHangar**)
- Nivel (**IdHangar**, **NúmeroNivel**, **Id Vivienda**, Superficie Total, Capacidad de almacenamiento, Restringido)
- Viviendas (**NúmeroNivel**, **Id Vivienda**)

- Bloques de Celdas(**LetraId**, **NúmeroNivel**, Capacidad de Prisioneros Restringido, Superficie Total, Capacidad de Almacenamiento)
- Prisioneros (**IdÚnico**, **LetraId** Nombre, Afiliación, Fecha de Ejecución, Fecha de Reclusión)

4. Lectura

Lo que principalmente el Artículo “Codd’s 12 Rules for an RDBMS” nos quiere dar a entender es que para que un sistema se denomine como Sistema de Gestión de Bases de Datos Relacionales, éste sistema debe usar (exclusivamente) sus capacidades relacionales para gestionar la base de datos.

Las doce reglas que él postuló en éste artículo son:

1. La regla de la información. Nos dice que toda la información en la base de datos es representada unidireccionalmente, por valores en posiciones de las columnas dentro de filas de tablas. Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera, con valores en tablas.

Ejemplo: Toda la información necesaria para un correcto funcionamiento de la base de datos debe de ir dentro de una tabla, ya que si no se puede generar información incompleta que no tendría significado alguno dentro de la base de datos.

2. Regla de acceso garantizado. Todos los datos deben ser accesibles sin ambigüedad. Esta regla es esencialmente una nueva exposición del requisito fundamental para las llaves primarias. Dice que cada valor escalar individual en la base de datos debe ser lógicamente direccionable especificando el nombre de la tabla, la columna que lo contiene y la llave primaria.

Ejemplo: Cualquier almacenado tiene que poder ser direccionado unívocamente. Para ello hay que indicar en qué

tabla está, cuál es la columna y cuál es la fila (mediante la clave primaria).

Un buen ejemplo de esta regla es cuando hacemos una búsqueda dentro de la base de datos y como resultado nos muestra mucha información repetida además de información que en realidad no buscamos, esto se debe a que las llaves primarias no apuntan de forma correcta hacia su tabla o dato que dependan de ella.

3. Tratamiento sistemático de valores nulos. Los valores nulos (que son distintos de la cadena vacía, blancos, 0s, ...) se soportan en los SGBD totalmente relacionales para representar información desconocida o no aplicable de manera sistemática, independientemente del tipo de datos.

Ejemplo: Cuando la base no puede manejar la entrada de datos nulos, lo cual podría producirnos un error, por eso es conveniente tener en cuenta los dominios de cada tipo de atributo o campo de la tabla, así como la inexistencia del dato requerido.

4. Catálogo dinámico en línea basado en el modelo relacional. El sistema debe soportar un catálogo en línea, el catálogo relacional debe ser accesible a los usuarios autorizados. Es decir, los usuarios autorizados deben poder tener acceso a la estructura de la base de datos (catálogo).

Ejemplo: El uso de clavel para controlar lo que cada usuario puede ver o manejar dentro de la base de datos es muy importante y debe poder ser accesible en cualquier momento por quien así lo requiera.

Debe de ser posible el acceso a datos y metadatos.

5. La regla comprensiva del sublenguaje de los datos. Un sistema relacional debe soportar varios lenguajes y varios modos de uso de terminal (ej. Rellenar formularios, etc.). Sin

embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante una sintaxis bien definida, como cadenas de caracteres, y que sea completo, soportando:

- Definición de datos
- Definición de vistas
- Manipulación de datos (interactiva y por programa)
- Límites de integridad
- Limitantes de transacción (iniciar, realizar, deshacer)

Ejemplo: Independientemente de que el SGBD ofrezca interfaces amigables para el usuario o administrador, se debe disponer de un lenguaje relacional con las mismas (o más) funciones que las interfaces. Cuando una base de datos no puede funcionar de forma correcta dentro de cierto tipo de lenguajes. Debe de ser multiplataforma por así decirlo.

6. Regla de actualización. Todas las vistas que son teóricamente actualizables deben ser actualizables por el sistema.

Ejemplo: La actualización debe ser automática, sin necesidad de que el usuario tenga que estar actualizando manualmente. Una vista puede ser el conjunto de socios de la biblioteca que viven en Actopan. Si quiero añadir un socio que vive en Actopan a la vista (sería actualizable), debo poder hacerlo sin notarlo, debe encargarse el SGBD de manejarlo.

7. Alto nivel de inserción, actualización y borrado. La capacidad de manejar una relación base o derivada como un solo operando se aplica no solo a la recuperación de los datos (consultas), sino también a la inserción, actualización y borrado de datos.

Ejemplo: La base de datos relacional debe de poder ser capaz de realizar con éxito operaciones de inserción, actualización y borrado a través de sus capacidades relacionales.

Si quiero eliminar a los socios de la biblioteca que sean de Actopan, no tengo que ir uno a uno. El sistema me debe proporcionar mecanismos para borrarlos todos de golpe.

8. Independencia física de los datos. Los programas de aplicación y actividades de la terminal permanecen inalterados a nivel lógico cuando quiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

Ejemplo: El modelo relacional es un modelo lógico de datos, y oculta las características de su representación física.

Indicamos al SGBD que la base de datos ya no se guarda en un solo disco duro, sino que ahora se guarda en varios discos duros en RAID. ¿Tenemos que modificar la estructura de la base de datos y los programas que acceden ella por ese cambio?

No deberíamos. El SGBD se debe encargar.

9. Independencia lógica de los datos. Los cambios al nivel lógico (tablas, columnas, filas, etc.) no deben requerir un cambio a una solicitud basada en la estructura. La independencia de datos lógica es más difícil de lograr que la independencia física de datos.

Ejemplo: Cambios que preservan la información:

- Añadir un atributo a una tabla base.
- Sustituir dos tablas base por la unión de las mismas. Usando vistas de la unión puedo recrear las tablas anteriores.

10. Independencia de integridad. Las limitaciones de la integridad se deben especificar por separado de los programas de la aplicación y se almacenan en la base de

datos. Debe ser posible cambiar esas limitaciones sin afectar innecesariamente las aplicaciones existentes.

Ejemplo: El objetivo de las bases de datos no es sólo almacenar los datos, sino también sus relaciones y evitar que estas (limitantes) se codifiquen en los programas. Por tanto, en una BDR se deben poder definir limitantes de integridad.

- Una BDR tiene integridad de entidad. Es decir, toda tabla debe tener una clave primaria.
- Una BDR tiene integridad referencial. Es decir, toda clave externa no nula debe existir en la relación donde es primaria.

11. Independencia de la distribución. La distribución de las personas de la base de datos a las varias localizaciones debe ser invisible a los usuarios de la base de datos. Los usos existentes deben continuar funcionando con éxito:

- Cuando una versión distribuida del SGBD se introdujo por primera vez
- Cuando se distribuyen los datos existentes se redistribuyen en todo el sistema.

Ejemplo: Las mismas órdenes y programas se ejecutan igual en una BD centralizada que en una distribuida.

Voy a sacar dinero de un cajero. El dinero de mi cuenta se guarda en un servidor. Saco un extracto de movimientos. Y esa información se guarda ¿En el mismo servidor de antes? No tiene porqué y yo no tengo porqué saberlo (Transparencia).

En un servidor, no tiene porqué guardarse todos los datos de todos los clientes de un banco. Es más lógico que se guarden en varios servidores, uno por zona geográfica, por ejemplo. Los clientes no tienen porqué saberlo. ¿Ellos lo notan?, No, y además se tiene que encargar el SGBD de que sea transparente.

12. La regla de la no subversión. Si el sistema proporciona una interfaz de bajo nivel de registro, a parte de una interfaz relacional, que esa interfaz de bajo nivel no se pueda utilizar para subvertir el sistema.

Ejemplo: No debe ser posible saltarse los limitantes de integridad impuestos al tratar las tuplas a ese nivel.

Si se puede, con la facilidades que da el SGBD utilizar un sistema para acceder a los registros (desde aplicaciones externas al SGBD), éste sistema debe respetar todas las reglas anteriores. Debe seguir manteniendo todas las integridades de los datos.

Un sistema podrá considerarse “más relacional” cuanto más siga estas reglas.

