



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

FUNDAMENTOS DE BASES DE DATOS - 7094

T A R E A 3

EQUIPO:

DEL MONTE ORTEGA MARYAM MICHELLE - 320083527

SOSA ROMO JUAN MARIO - 320051926

CASTILLO HERNÁNDEZ ANTONIO - 320017438

ERIK EDUARDO GÓMEZ LÓPEZ - 320258211

JULIO CÉSAR ISLAS ESPINO - 320340594

FECHA DE ENTREGA:

24 DE SEPTIEMBRE DE 2024

PROFESOR:

M. EN I. GERARDO AVILÉS ROSAS

AYUDANTES:

LUIS ENRIQUE GARCÍA GÓMEZ

KEVIN JAIR TORRES VALENCIA

RICARDO BADILLO MACÍAS

ROCÍO AYLIN HUERTA GONZÁLEZ



Tarea 3

Preguntas de repaso

1. ¿Qué es una relación y qué características tiene?

Comenzamos por notar que una BDR no es más que un conjunto de relaciones.

Una relación es un conjunto de tuplas, donde cada tupla es un conjunto de **atributos**. Cada atributo tiene un nombre y un **dominio**. Existe un conjunto no vacío de atributos que forman la **llave primaria** de la relación, esto garantiza que no existan tuplas repetidas; además de esta llave primaria, una relación puede tener **llaves foráneas** que hacen referencia a otras relaciones. Estas llaves deben coincidir con valores de la llave primaria de la relación referenciada; finalmente, existen las **llaves candidatas**, que son llaves primarias potenciales, es decir, que cumplen con las restricciones de llave primaria, pero no son seleccionadas como tal. Finalmente, una relación puede tener **restricciones de integridad**, que son condiciones que deben cumplir las tuplas de la relación. Hay 2 tipos de restricciones: inherentes y de usuario.

Además de esto, sabemos que una relación tiene **nombre único** y que los atributos de una relación son **no ordenados**, es decir, que el orden de los atributos no importa. Lo mismo aplica para las tuplas. **No hay atributos multivaluados**, es decir, no hay atributos que tengan más de un valor en una tupla; además, los atributos deben ser **atómicos**, es decir, que no pueden ser descompuestos en atributos más pequeños, y tienen **nombre único** en la relación.

Información tomada de la presentación de la clase: "03ModeloR_BD.pdf" de la materia "Fundamentos de Bases de Datos" páginas 3 a 6.

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

Las llaves primarias y foráneas dentro de un modelo relacional imponen algunas restricciones las cuales muchas veces son necesarias para mantener una buena coherencia y relación de los datos que están dentro del modelo.

- **Llave primaria (PK):** La llave primaria debe garantizar que cada registro en la tabla sea único, es decir, no puede haber dos registros con el mismo valor en los atributos que forman la llave primaria. Además, una PK no puede tener valores nulos, ya que su propósito es identificar de manera clara y sin ambigüedades a cada registro.
- **Llave foránea (FK):** La llave foránea, por otro lado, se usa para conectar dos tablas. Los valores de una FK deben coincidir con los de una llave primaria en otra tabla (*o en la misma tabla, si se trata de una relación recursiva*). La principal restricción que impone una FK es la integridad referencial: no se puede insertar un valor en una FK si no existe una correspondencia en la tabla referenciada. A diferencia de una PK, una FK puede contener valores nulos cuando la relación no es obligatoria.

Información tomada de la presentación de la clase: "03ModeloR_BD.pdf" de la materia "Fundamentos de Bases de Datos"

3. Investiga que cuáles son las Reglas de Codd y explica con tus propias palabras cada una de ellas. Indica por qué consideras que son importantes.

Bien se sabe que las Reglas de Codd fueron propuestas por Edgar F. Codd, son 12, las cuáles son principios que definen los requisitos que una Base de Datos debe cumplir para ser considerada una Base de Datos relacional en su totalidad. Estas reglas son importantes y piezas clave, ya que garantizan la consistencia y eficiencia de los sistemas de Bases de Datos relacionales.

- a) **Regla de la información:** Nos dice que los valores serán representados en tablas, esto quiere decir que lo que necesitamos en la Base de Datos debe estar en filas y columnas. Aparte es importante porque garantiza un modelo uniforme para almacenar datos.
- b) **Regla de acceso garantizado:** Nos dice que, para poder acceder a cualquier dato en la Base de Datos, se debe señalar sus tablas, columnas y la clave primaria. Esto nos asegura que no importa el tamaño de la Base de Datos, siempre se podrá encontrar los datos de una manera accesible.
- c) **Regla de valores nulos:** Nos dice que el sistema debe permitir los valores nulos, sin confundirlos con ceros o espacios vacíos. Esos valores nulos son necesarios cuando no hay información o esta no se aplica.
- d) **Regla del catálogo en línea:** Nos dice que el catálogo de la Base de Datos, el cual contiene toda la información de las tablas y otros elementos, debe estar disponible para el público, es decir se puede consultar y manipular por los usuarios y administradores.
- e) **Regla del sublenguaje de datos:** Nos dice que la Base de Datos debe tener un lenguaje propio que permita hacer todo tipo de operaciones (consultar, actualizar, eliminar, etc.). Esto es clave, pues sin un lenguaje de manipulación de datos, no se podría interactuar con la base de datos de manera eficiente.
- f) **Regla de actualización de vistas:** Nos dice que las vistas son consultas guardadas, deben ser actualizables. Esto nos permite que no solo se puedan ver los datos a través de una vista, sino también poder modificarlos si es necesario.
- g) **Regla de inserción, actualización y eliminación masiva:** Nos dice que se debe poder hacer operaciones sobre múltiples filas de datos a la vez, no solo una a la vez. Esto es esencial para la eficiencia, pues te permite trabajar con grandes cantidades de datos sin tener que procesarlos uno por uno.
- h) **Independencia física de los datos:** Nos dice que los cambios en la manera en que los datos se almacenan físicamente no deben afectar la forma en que se acceden o manejan. Esto significa que los usuarios no deben preocuparse por cómo se guardan los datos.
- i) **Independencia lógica de los datos:** Nos dice que, si se realizan cambios en la estructura lógica de la Base de Datos, como agregar nuevas tablas o modificar columnas, no debe afectar las aplicaciones existentes. Esto nos garantiza la estabilidad y flexibilidad, sin romper aplicaciones que ya funcionan.
- j) **Regla de integridad:** Nos dice que esas reglas deben definirse y mantenerse dentro de la Base de Datos, no en las aplicaciones. Esto nos garantiza que la integridad de la información no dependa de las aplicaciones externas.
- k) **Regla de independencia de la distribución:** Nos dice que la Base de Datos debe funcionar correctamente, aunque se encuentre en varios lugares. Esto significa que no importa dónde estén físicamente los datos, el sistema debe ser capaz de acceder a ellos; esto en un mundo con redes distribuidas y aplicaciones en la nube.
- l) **Regla de no subversión:** Nos dice que si el sistema permite acceder a los datos a un nivel más bajo (código), esto no debe violar las reglas de integridad y seguridad de la Base de Datos, aparte si hay acceso no deberías poder saltarte las reglas del sistema.

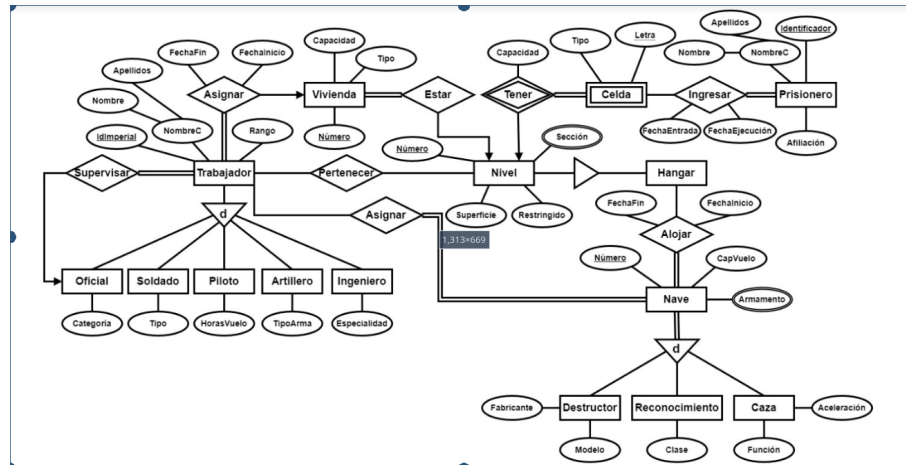
0.1. Conclusión

Por último, para mencionar, las Reglas de Codd son esenciales porque aseguran que los SBD relacionales funcionen de una manera eficiente y coherente, aparte todas esas reglas establecen una base sólida para el diseño y construcción de la Base de Datos, esto permite que sean modificables, confiables y seguras. Además, estas reglas hacen que haya una separación lógica del manejo de datos de las aplicaciones,

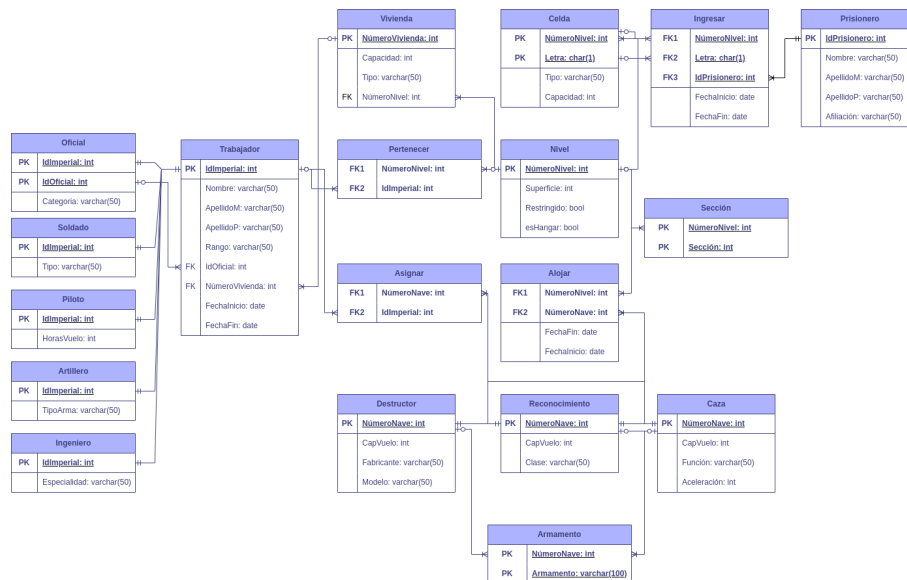
esto nos dice que los datos pueden evolucionar y adaptarse a nuevas necesidades sin tener que reescribir todo el código de las aplicaciones.

Conversión de Modelo E/R a Modelo relacional

a. Traduce el siguiente modelo Entidad – Relación a su correspondiente Modelo Relacional:



Tras traducir, nos queda el siguiente modelo relacional:



La verdad es que hay varias que son tanto PK como FK pero me comentaron que no era correcto ponerlas como PK y FK a la vez, así que seguí el pdf.

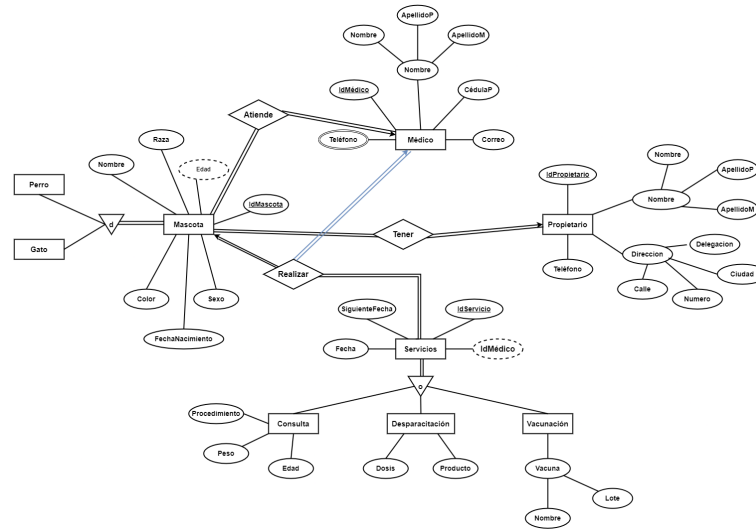
Para la parte de **Supervisar** entre Trabajador y Oficial, tuve que agregarle un atributo **IdOficial** a la tabla Trabajador, ya que si no era guardar en Trabajador un atributo IdTrabajador que no se entiende tanto y se pierde la semántica de la relación.

Además de esto, varias relaciones son del mismo tipo entonces sus conexiones se sobrelapan lo que hace un poco difícil seguir el diagrama pero por lo demás, creo que está bien.

Finalmente, parti los **Apellidos** en **ApellidoP** y **ApellidoM** para que sea atómico.

- b. Traduce a su correspondiente Modelo Relacional, el problema de la Clínica Veterinaria. Si realizaste alguna modificación a tu diseño original (para mejorarlo), indica los cambios hechos y la justificación de éstos. Deberás mostrar el modelo E-R y su correspondiente traducción. Es importante que muestres tanto las restricciones de entidad como las de integridad referencial.

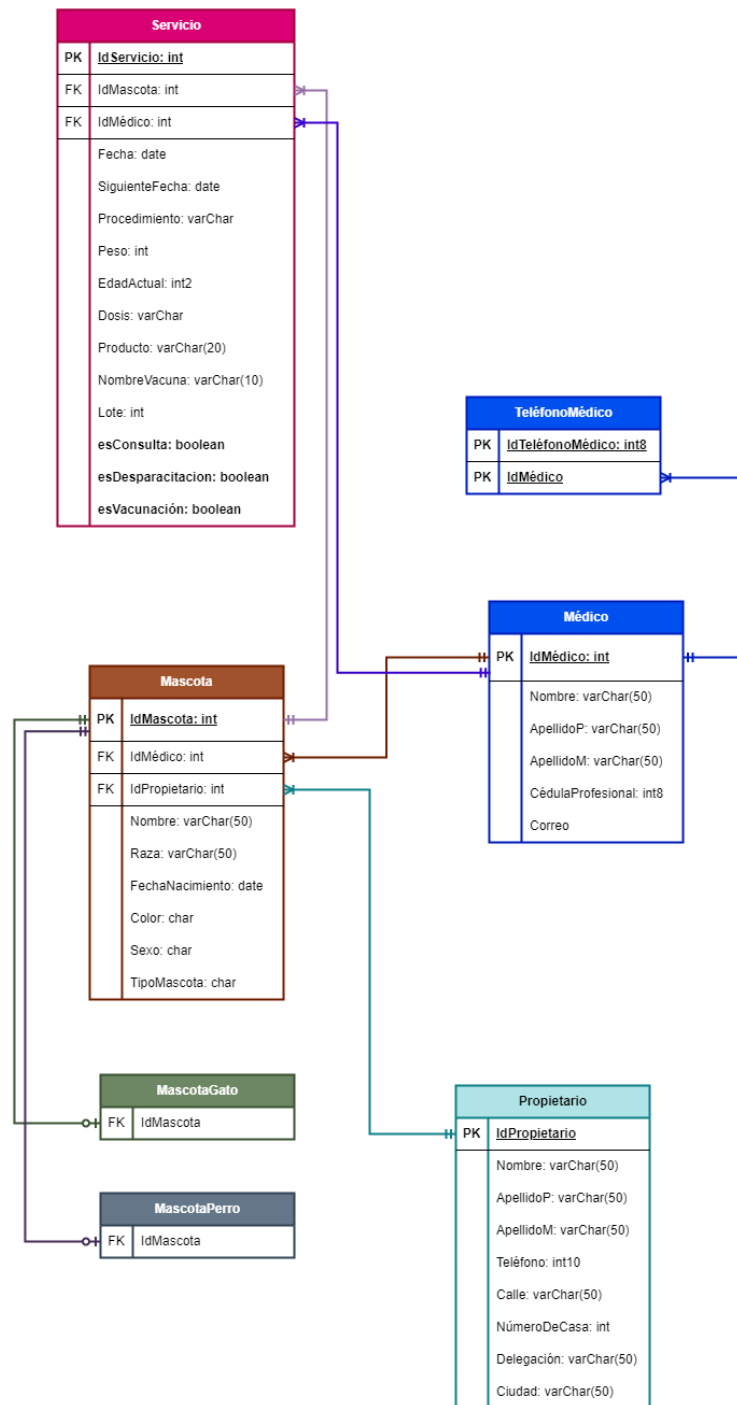
Modelo Entidad-Relacion



Cambios:

- En Mascota hemos cambiado la Edad a un atributo calculado
- En Servicio cambiamos IdMédico a un atributo calculado
- En Propietario cambiamos Teléfono de calculado a normal, ya que en la foto del carnet dado solo se permite un solo teléfono

Modelo Relacional



Para Mascota, a pesar de ser herencia disyuntiva obligatoria y no se considera al padre, en este caso decidimos dejarlo para no tener que hacer tantas relaciones para los dos tipos de mascotas que atiende la clínica, esto como el ejemplo de Metro del profe:D

Claves Foráneas (integridad referencial):

- Servicio: IdMascota referencia a Mascota(IdMascota)
- Servicio: IdMédico referencia a Médico(IdMédico)
- Mascota: IdMédico referencia a Médico(IdMédico)
- Mascota: IdPropietario referencia a Propietario(IdPropietario)
- TeléfonoMédico: IdMédico referencia a Médico(IdMédico)

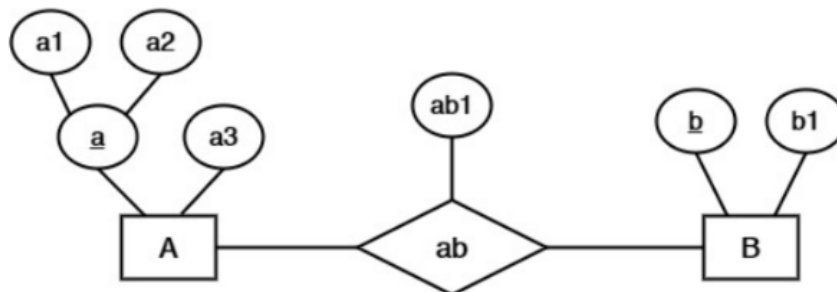
Estas relaciones aseguran que no se puedan ingresar datos inconsistentes, como un servicio para una mascota que no existe.

- Para esConsulta, esDesparacitacion, esVacunación estos campos solo pueden tener dos estados (verdadero o falso), por lo que boolean es el tipo más apropiado.

- La entidad MascotaGato y MascotaPerro en el modelo relacional es una herencia, lo cual es una buena práctica para manejar atributos específicos de cada tipo de mascota.

Modelo relacional e inserción de tuplas.

Considera el siguiente Modelo E/R:



- a. Completa la tabla que se presenta a continuación, convirtiendo el Modelo E-R en un Modelo Relacional, para todas las opciones de cardinalidad (considera en todos los casos, participación parcial). Indica las relaciones resultantes, su llave primaria y la integridad referencial. Considera el formato `Tabla(llavePk1, llavePK2, ..., atr1, atr2, ..., llaveFk1,...)`

Modelo E-R	Modelo Relacional
M:N	<code>A(<u>a1</u>,<u>a2</u>,a3), B(<u>b</u>,b1), AB(a1,a2,b,ab1)</code>
1:N	<code>A(<u>a1</u>,<u>a2</u>,a3), B(<u>b</u>,b1,a1,a2,ab1)</code>
N:1	<code>A(<u>a1</u>,<u>a2</u>,a3,b,ab1), B(<u>b</u>,b1)</code>
1:1	<code>A(<u>a1</u>,<u>a2</u>,a3), B(<u>b</u>,b1), AB(a1,a2,b,ab1)</code>

- b. Del inciso a) toma el MR que obtuviste para la cardinalidad M : N. Asume que los atributos a1, b y ab1 son de tipo entero, mientras que a2, a3 y b1 son de tipo cadena. Supón que la relación A tiene 4 tuplas con los siguientes valores (2,'ww','a'), (4,'xx','b'), (6,'yy','c'), (8,'zz','d') y la relación B tiene 5 tuplas identificadas por los valores 17, 27, 37, 47, 57. Los incisos que se presentan a continuación, representan un conjunto de tuplas a insertar (en ese orden) en la relación AB, indica cuál conjunto se puede insertar completamente en dicha relación. Justifica tu respuesta en cada caso.

- a) (8,'zz',17,5); (6,'yy',57,10); (4,'xx',27,15); (2,'ww',37,20); (4,'xx',27,15)

Podemos insertar (8,'zz',17,5); (6,'yy',57,10); (4,'xx',27,15); (2,'ww',37,20) y ya, insertar **otra vez** (4,'xx',27,15) es innecesario ya que existiría más de una tupla con la misma llave.

- b) (17,'zz',2,'m'); (27,'yy',4,'n'); (37,'xx',6,'o'); (47,'ww',8,'p'); (57,'zz',4,'q')

Ninguna se puede insertar ya que **ab1** es de tipo entero según las especificaciones y para este conjunto de tuplas, en todas se representa a **ab1** como cadena o caracter.

- c) (2,'a',17,23); (4,'b',27,24); (6,'c',37,25); (8,'d',47,26); (2,'a',57,27)

Todas se pueden insertar, ya que los tipos de datos están correctos y no hay tuplas duplicadas ya que todas tienen más de un atributo distinto para los identificadores.

- d) (2,'ww',57,'a'); (4,'xx',37,'b'); (6,'yy',17,'c'); (8,'zz',37,'d'); (4,'xx',47,'a')

Nuevamente no podemos insertar ninguna ya que **ab1** es de tipo entero y se le pasan cadenas.

- c. Del inciso a) toma como base el MR que obtuviste para la cardinalidad 1 : N. Los incisos que se presentan a continuación representan un conjunto de tuplas a insertar (en ese orden) en la relación B, indica cuál conjunto se puede insertar completamente en dicha relación. Justifica tu respuesta en cada caso.

- a) (2,'f',57,'zz'); (4,'g',47,'yy'); (6,'h',37,'xx'); (8,'i',27,'ww'); (2,'j',17,'yy')

- b) (57,8,'zz','f'); (47,6,'yy','g'); (37,4,'xx','h'); (27,2,'ww','i'); (17,6,'yy','j')

- c) (57,'f',8,'zz'); (47,'g',6,'yy'); (37,'h',4,'xx'); (27,'i',2,'ww'); (17,'j',6,'yy')

- d) (57,'f',8,'a'); (47,'g',6,'b'); (37,'h',4,'c'); (27,'i',2,'d'); (17,'j',6,'c')

Tomando como base el modelo relacional obtenido para la relación 1:N del inciso a), procederemos a analizar cada conjunto de tuplas para determinar cuál de ellos se puede insertar completamente en la relación B.

- a) (2,'f',57,'zz'); (4,'g',47,'yy'); (6,'h',37,'xx'); (8,'i',27,'ww'); (2,'j',17,'yy')

En este caso, los valores de b (57, 47, 37, 27, 17) son todos únicos, por lo que no hay problemas con la clave primaria en B. Además, las combinaciones de a1 y a2 ((2,'zz'), (4,'yy'), (6,'xx'), (8,'ww'), (2,'yy')) no generan conflictos en términos de las claves foráneas. Si estas combinaciones existen en la tabla A, se pueden insertar sin problemas.

∴ El conjunto de tuplas se puede insertar completamente.

- b) (57,8,'zz','f'); (47,6,'yy','g'); (37,4,'xx','h'); (27,2,'ww','i'); (17,6,'yy','j')

Los valores de b (57, 47, 37, 27, 17) son diferentes, por lo que no hay conflictos con la clave primaria en B. Las combinaciones de a1 y a2 ((8,'zz'), (6,'yy'), (4,'xx'), (2,'ww'), (6,'yy')) muestran una repetición de (6,'yy'), pero mientras esa combinación sea válida en A, no debería haber problemas para insertar las tuplas en B.

∴ El conjunto de tuplas se puede insertar completamente.

- c) (57,'f',8,'zz'); (47,'g',6,'yy'); (37,'h',4,'xx'); (27,'i',2,'ww'); (17,'j',6,'yy')

En este conjunto, los valores de b (57, 47, 37, 27, 17) son únicos, lo cual es correcto para la clave primaria de B. Las combinaciones de a1 y a2 ((8,'zz'), (6,'yy'), (4,'xx'), (2,'ww'), (6,'yy')) incluyen una duplicación de (6,'yy'), pero si esta combinación es válida en A, no habría problemas para que las tuplas se inserten en B.

∴ Este conjunto de tuplas se puede insertar completamente.

d) (57,'f',8,'a'); (47,'g',6,'b'); (37,'h',4,'c'); (27,'i',2,'d'); (17,'j',6,'c')

Aquí los valores de **b** (57, 47, 37, 27, 17) siguen siendo únicos, por lo que no hay conflicto con la clave primaria en **B**. Sin embargo, la combinación de **a1** y **a2** (6, 'c') aparece en dos tuplas, lo cual es un problema, ya que **a1** y **a2** forman parte de la clave primaria compuesta en **A**. Esta duplicación no es válida, lo que viola la integridad referencial y, por lo tanto, no es posible insertar todas las tuplas.

∴ No se puede insertar completamente este conjunto de tuplas.

- d. Considera el mismo escenario del inciso b para las relaciones **A** y **B**. Toma como base el Modelo Relacional que obtuviste para la cardinalidad 1:1. Supón que tu modelo tiene participación parcial de ambos lados. Propón un conjunto de 4 tuplas que se pueda insertar en **ab** y un conjunto que no se pueda insertar (también de 4 tuplas). Justifica tu respuesta en cada caso.

Conjunto de 4 tuplas que se pueden insertar en **AB**

Supongamos que las tuplas presentes en **A** y **B** son las siguientes:

- **A** : (2, 'ww', 'a'), (4, 'xx', 'b'), (6, 'yy', 'c'), (8, 'zz', 'd')
- **B** : (17, 'e'), (27, 'f'), (37, 'g'), (47, 'h')

Propuestas de tuplas para **AB**:

- a) (2, 'ww', 17, 10)
- b) (4, 'xx', 27, 20)
- c) (6, 'yy', 37, 30)
- d) (8, 'zz', 47, 40)

Estas tuplas se pueden insertar porque:

- Cada $(a1, a2)$ de **AB** corresponde a una combinación válida de claves primarias de **A**.
- Cada **b** en **AB** coincide con una clave primaria en **B**.
- Las combinaciones de $(a1, a2)$ y **b** son únicas, respetando la restricción de 1:1, sin repetir la asociación entre **A** y **B**.

Conjunto de 4 tuplas que no se pueden insertar en **AB**

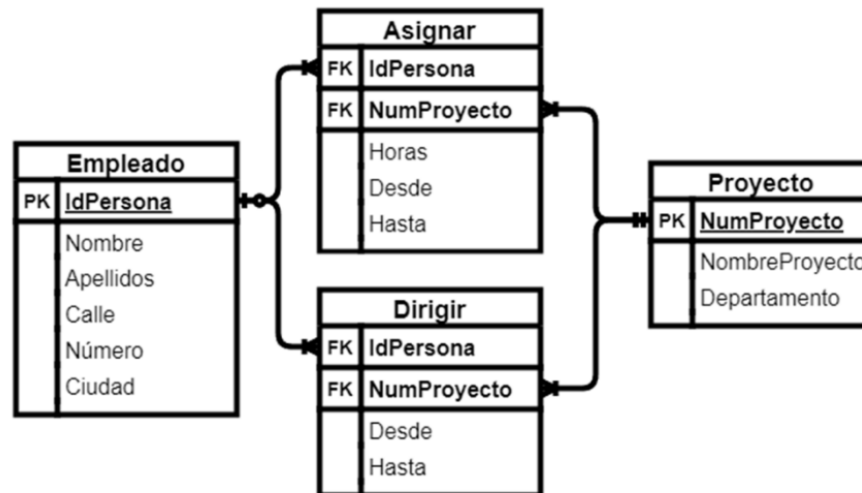
- a) (2, 'ww', 17, 10)
- b) (2, 'ww', 27, 20)
- c) (4, 'xx', 27, 30)
- d) (4, 'xx', 47, 40)

Estas tuplas no se pueden insertar debido a que:

- Las primeras dos tuplas intentan asociar la misma tupla de **A** (2, 'ww') con diferentes tuplas de **B** (17 y 27), violando la restricción de 1:1.
- Las últimas dos tuplas intentan asociar la misma tupla de **A** (4, 'xx') con diferentes tuplas de **B** (27 y 47), lo cual también infringe la relación 1:1.
- La relación 1:1 implica que una tupla de **A** solo puede estar asociada con una única tupla de **B** y viceversa, y estas tuplas propuestas intentan establecer múltiples asociaciones.

Modelo relacional y restricciones de integridad

A continuación, se encuentra el Modelo Relacional de un departamento de recursos humanos de alguna empresa. En este esquema, supón que desde es inclusivo, mientras que hasta es exclusivo, definiendo el período [desde,hasta). Indica cuáles de las siguientes afirmaciones se cumplen y por qué razón (sin considerar restricciones adicionales):



- a. Dos compañías con el nombre ‘Panaphonics’ podrían existir al mismo tiempo.

Voy a asumir que la relación **Proyecto** es donde se pondría el nombre de la compañía (que no se que tanto sentido tenga hacer esto para el departamento de RH de una empresa pero bueno). Vemos que en esta relación el único atributo que no se puede repetir es **NumProyecto** pues es la clave primaria. Por lo tanto, si se puede tener dos compañías con el nombre ‘Panaphonics’ al mismo tiempo.

Si se quisiera que solo existiera una compañía con el nombre ‘Panaphonics’ al mismo tiempo, se podría definir el atributo **NombreCompañía** como clave primaria o una llave primaria compuesta con el atributo **NumProyecto**.

- b. Dos o más empleados pueden dirigir la compañía ‘Sorny’ al mismo tiempo.

En el esquema **Dirigir** no se especifica que el atributo **IdPersona** sea clave primaria, por lo tanto, se puede repetir. Por lo tanto, dos o más empleados pueden dirigir la compañía ‘Sorny’ al mismo tiempo.

Si se quisiera que solo un empleado pudiera dirigir la compañía ‘Sorny’ al mismo tiempo, se podría definir el par de atributos como clave primaria compuesta.

- c. Un empleado puede trabajar en ‘Compumundo Hipermegared’ y dirigir ‘El Bar de Moe’ al mismo tiempo.

Verdadera. Vemos que hay dos entidades separadas relacionadas con los proyectos: Asignar y Dirigir. Ambas tienen una relación con Empleado a través de la llave foránea IdPersona, y con Proyecto a través de FK NumProyecto. No hay ninguna restricción visible que impida a un empleado estar asignado a

un proyecto y dirigir otro al mismo tiempo. Las tablas *Asignar* y *Dirigir* permiten múltiples registros para el mismo *IdPersona* con diferentes *NumProyecto*.

- d. **Para dirigir ‘Leftorium’ un empleado debe trabajar en dicha compañía.** Falso. No es necesario

que un empleado esté asignado a trabajar en un proyecto para poder dirigirlo. En el modelo, la relación *Dirigir* no depende de *Asignar*, por lo que un empleado puede dirigir un proyecto sin estar registrado como trabajador en dicho proyecto.

- e. **Un empleado podría dirigir ‘Krusty Burgers’ en dos períodos de tiempo diferentes.** Ver-

dadero. La relación *Dirigir* permite que un empleado dirija un proyecto en diferentes períodos de tiempo, ya que los campos *Desde* y *Hasta* permiten registrar múltiples intervalos sin restricciones de solapamiento.

- f. **Se puede almacenar ‘Laramie Cigarettes’ sin necesidad de definir a un director** Verdadero.

La existencia de un proyecto en la tabla *Proyecto* no depende de la relación *Dirigir*, por lo que un proyecto puede ser almacenado sin que se defina un director.

- g. **Los empleados y/o directores deben vivir en la misma Ciudad que la Compañía para la que laboran/dirigen.**

Para que esta afirmación fuera verdadera, debería existir una relación entre la ciudad donde reside un empleado o director y la ubicación del proyecto o compañía para la que trabajan o dirigen. Sin embargo, dado que no hay una tabla llamada *compañía*, podemos suponer que esta información se encuentra en las otras tablas como *proyecto*, *asignar*, o *dirigir*. Con esto en mente, podemos analizar dos situaciones:

Caso 1: Si hay un campo Ciudad tanto en proyecto, asignar o dirigir, como en empleado:

Si alguna de las tablas relacionadas con proyectos (*proyecto*, *asignar*, o *dirigir*) incluye un campo para la ubicación del proyecto, y la tabla *empleado* también tiene un campo *Ciudad* (que sabemos que tiene), podríamos crear una restricción que asegure que las ciudades coincidan. Si esto se cumple, la afirmación sería válida.

Caso 2: Si solo la tabla empleado tiene un campo Ciudad:

Si únicamente la tabla *empleado* incluye un campo *Ciudad* y ninguna otra tabla contiene información sobre la ubicación de los proyectos o compañías, entonces no habría forma de verificar si las ciudades coinciden. En este caso, la afirmación no sería válida, ya que faltaría la información necesaria para hacer la comparación.

En conclusión, esta afirmación no se cumple de manera automática, ya que no hay una relación directa que conecte la ciudad del empleado o director con la ubicación de la compañía o proyecto. Para que la afirmación sea verdadera, sería necesario agregar un campo de ubicación en las tablas que gestionan proyectos y establecer una relación con los empleados.

∴ La afirmación no se cumple en el modelo relacional actual.

- h. **Ningún empleado puede cobrar más de un Salario al mismo tiempo.**

Falso o indeterminado, ya que no muestra ninguna información sobre salarios. No hay una tabla o atributo específico para salarios en ninguna de las entidades mostradas. Sin esta información, no podemos hacer ninguna afirmación sobre cómo se manejan los salarios o si un empleado puede o no cobrar más

de uno al mismo tiempo, ni siquiera podríamos calcular un Salario en base a las horas trabajadas por proyecto ya que no tenemos un dato que nos diga cuánto es el salario asignado para cada proyecto.

- i. **Algunas tuplas en Trabaja podrían no tener valor para el atributo desde y ningún empleado asociado a ellas.**

0.2. Afirmación i)

La afirmación sugiere que en la relación **Trabaja** (que podría corresponder a la tabla **Asignar** en el modelo relacional), algunas tuplas podrían tener los siguientes problemas:

- No contener un valor definido para el atributo **desde**.
- No tener asociado un empleado a través de la **clave foránea** correspondiente.

Analizemos ambos:

0.2.1. Falta de valor en el atributo desde

En el modelo relacional que se describe, se utiliza el período de asignación (**desde, hasta**) para representar el tiempo durante el cual un empleado está asignado a un proyecto.

El valor **desde** representa el **inicio** de dicha asignación, y según el enunciado, es **inclusivo**, lo que significa que el empleado comienza su trabajo en el proyecto a partir de esa fecha.

Si alguna tupla no tuviera un valor en el campo **desde**, no sería posible definir cuándo comenzó la asignación de un empleado a un proyecto. Esto genera un problema en términos de **consistencia de los datos**, ya que una asignación sin una fecha de inicio es **incompleta** y difícil de interpretar.

En un modelo relacional bien diseñado, el atributo **desde** debe ser **obligatorio**, ya que es un clave para definir el período de asignación. Permitir que este atributo sea nulo implicaría una **inconsistencia en tiempos** en el registro de las asignaciones.

0.2.2. Ausencia de empleado relacionado a la tupla

El campo **IdPersona** en la relación **Trabaja** (o **Asignar**) es una **clave foránea** que se refiere a la tabla **Empleado**. Esto significa que cada registro en **Trabaja** debería estar asociado a un empleado a través de esta **clave foránea**.

Si no existe un valor para **IdPersona** en alguna tupla, entonces dicha asignación no estaría asociada a ningún empleado, lo que **violaría la integridad referencial** del modelo. En este sistema de gestión de recursos humanos, no tendría sentido que existiera una asignación de trabajo sin un empleado asignado, sería algo ilógico.

En un **modelo relacional correcto**, este tipo de relaciones se define para garantizar que las tuplas de la tabla **Trabaja** siempre estén asociadas a un empleado válido. Es decir, no debería permitirse que el campo **IdPersona** sea nulo, ya que esto rompería la relación entre la tabla **Empleado** y **Trabaja**, y resultaría en una asignación **inválida**.

Para terminar, se concluye que la afirmación i) **no se cumple**. Permitir tuplas en la relación **Trabaja** sin un valor definido para el atributo **desde** o sin un empleado asociado violaría las reglas de **integridad** y **consistencia** de un modelo relacional.

- El valor **desde** es un atributo esencial para definir el inicio del período de trabajo, por lo que no debe permitirse que sea nulo.
- El campo **IdPersona** es una **clave foránea** que asegura que cada asignación esté relacionada con un empleado. Su ausencia implicaría un error de **integridad referencial**.

En conclusión, la afirmación i) no se ajusta a las reglas lógicas de un modelo relacional consistente.

j. **‘Mr. Plow’ no requiere tener definido algún empleado que la dirija.**

Esta afirmación es **verdadera**, debido a la naturaleza opcional de la relación entre **Empleado** y **Dirigir**. Según el modelo relacional, un empleado no necesariamente tiene que estar vinculado a una tupla en **Dirigir**. Esto significa que **Mr. Plow** puede existir sin que haya un director asignado, ya que la relación es opcional para el empleado, y el campo **id_director** podría ser **NULL**.

Sin embargo, si el modelo exigiera que todos los empleados tengan asignada una dirección de proyecto, entonces **Mr. Plow** tendría que tener un director definido.

∴ La afirmación se cumple porque la participación opcional de **Empleado** en la relación con **Dirigir** permite que algunas compañías no tengan un director asignado.