

METODOLOGIA DE DISEÑO DE BASES DE DATOS

Ing. Edgar T. Espinoza R.

- ✓ Es un atributo que sólo puede tener un solo valor por cada registro o instancia de una entidad.
- ✓ Observar que un atributo de valor simple no es necesariamente un atributo simple.



Atributo de valores múltiples..



- ✓ Llamado también atributo multivalorado.
- ✓ Es un atributo que puede tener más de un valor para cada registro o instancia de una entidad.
- ✓ En el ejemplo: Un trabajador puede tener varios números de teléfono. teléfono de casa, celular, teléfono de oficina, etc.

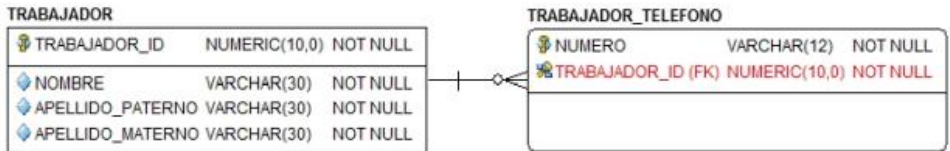
Diseño conceptual	Diseño lógico Crow's Foot e IDEFIX
Atributo multivalor	Atributo multivalor
	<ul style="list-style-type: none">• En el modelo relacional un atributo no puede tener múltiples valores para un mismo registro.• El modelo relacional no soporta de forma directa la representación de un atributo multivalorado. Por cada registro y columna debe existir un solo valor.• Para resolver este caso se emplean estrategias alternas que se muestran a continuación.• Algunos manejadores permiten la definición de colecciones como tipo de dato y así poder asociar una lista de valores a un registro. En un modelo relacional no debería emplearse el uso de colecciones ya que no es un elemento estándar o propio del modelo relacional.

3

Estrategia 1.



La solución típica para implementar un atributo multivalorado en diseño lógico es la incorporación de una nueva entidad que permita almacenar la lista de todos los valores relacionándola con la entidad original a través de una PK compuesta:



Los datos se verían así:

TRABAJADOR			
trabajador_id	nombre	apellido_paterno	apellido_materno
1	JUAN	LOPEZ	LARA
2	PABLO	LUNA	MONTES
3	MARIA	LARA	MARISCAL

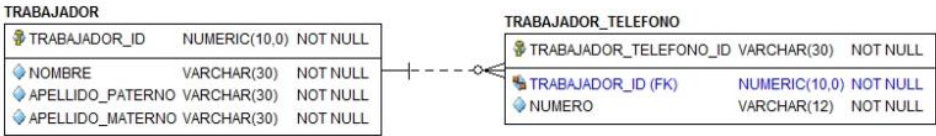
TRABAJADOR_TELEFONO	
trabajador_id (pk)	numero (pk)
1	55909023902
1	55029309233
1	33892898323
2	55902930212
2	55902899323

4



Estrategia 2.

Para resolver el inconveniente de la PK compuesta se puede hacer uso de una PK artificial. :

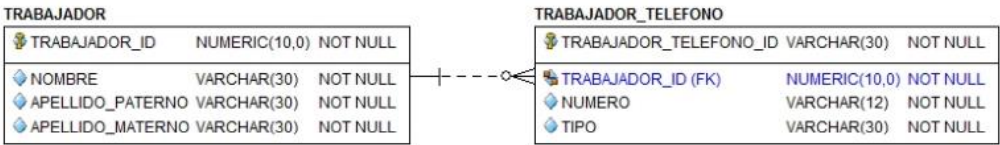


5



Ejemplo:

Suponer que se agrega el siguiente requerimiento al ejercicio anterior: Se desea saber el tipo de teléfono: teléfono de casa, de oficina, móvil, etc. ¿Qué cambios tendría que aplicarse al modelo?



Los datos se verían así:

TRABAJADOR			
trabajador_id	nombre	apellido_paterno	apellido_materno
1	JUAN	LOPEZ	LARA
2	PABLO	LUNA	MONTES
3	MARIA	LARA	MARISCAL

TRABAJADOR_TELEFONO		
tipo	trabajador_id	numero
CASA	1	55909023902
CELULAR	1	55029309233
OFICINA	1	33897898323
CASA	2	55902930212
CELULAR	2	55902899323

6

Observar que los valores del nuevo atributo tipo tienden a repetirse con una alta frecuencia. Se deberá tener un control adecuado de los valores que serán insertados ya que de lo contrario se podría generar el siguiente inconveniente



TRABAJADOR_TELEFONO

tipo	trabajador_id	numero
CASA	1	55909023902
CELULAR	1	55029309233
OFICINA	1	33892898323
HOME CEL	2	55902930212
MÓVIL	2	55902899323
TRABAJO	2	55982374834
DEPTO	3	55092849341

La redundancia y un mal control de los valores del campo tipo provoca inconsistencias.

7

Para resolver el problema, el modelo requiere aplicar un nivel de normalización mayor, es decir, se creará una nueva entidad (catálogo) para resolver el inconveniente.



Los datos se verían así:

TRABAJADOR_TELEFONO

telefono_trabador_id	trabajador_id	tipo_telefono_id	numero
1	1	1	55909023902
2	1	2	55029309233
3	1	3	33892898323
4	2	1	55902930212
5	2	2	55902899323

TIPO_TELEFONO

tipo_telefono_id	nombre
1	CASA
2	CELULAR
3	OFICINA



REPRESENTACION DE RELACIONES

- Representación general.
- Niveles de dependencia (débil y fuerte).
- Cardinalidad
- Dependencia de existencia.
- Participación de una entidad en una relación.
- Entidades débiles.
- Grado de una relación

9



Representación general de relaciones.

Diseño conceptual.

- ❖ Representación a través de un rombo con el nombre de la relación (verbo en 3ª persona).
- ❖ El nombre que se especifica en el rombo debe permitir leer la relación en 2 posibles sentidos dependiendo la forma en la que ambas entidades están organizadas en el diagrama.
- ❖ De izquierda hacia la derecha
- ❖ De arriba hacia abajo

Ejemplo:



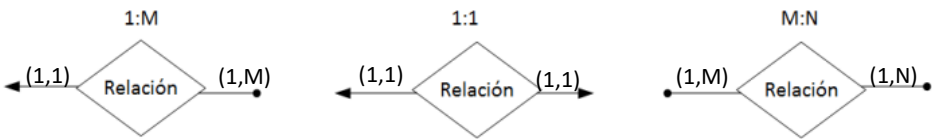
10

Papel o Rol:

- ❖ Es la función que realiza cada tipo de entidad en una relación.



Representación de tipos de relaciones:



- ❖ En una relación 1:M la flecha apunta a la tabla padre
- ❖ En una relación M:N se especifican 2 puntos finales en lugar de flechas.
- ❖ En una relación 1:1 se especifican ambas flechas.

11

Ejemplos:



Relación	Representación diseño conceptual
Sucursal y Ciudad (1:M)	
Factura y Renta (1:1)	
Sucursal y Empleado (Agente) (M:N)	

12

Diseño logico.



- ❖ Para relacionar 2 entidades se emplea el concepto de llave foránea (FK).
- ❖ Para asociar a las 2 entidades es necesario conocer el tipo de relación. 1:1, 1:M o M:N
- ❖ El siguiente paso es identificar la tabla que contendrá a la FK.

Niveles de dependencia.

El nivel de dependencia indica que tan fuerte es la relación entre 2 entidades.

- ❖ Relaciones suaves, débiles o no identificativas. -----
- ❖ Relaciones fuertes, duras o identificativas. _____

13

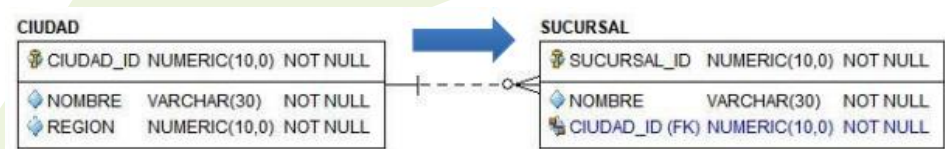
Relaciones no identificativas, suaves o débiles. :



- ✓ Consiste en relacionar 2 entidades a través de una **línea punteada**. Normalmente se emplea para representar relaciones 1:M aunque también puede emplearse para representar relaciones 1:1, y en algunos casos, relaciones M:N.
- ✓ Al asociar 2 tablas con una relación no identificativa (línea punteada) la llave primaria (PK) de la tabla padre pasa como un campo más de la tabla hija como llave foránea (FK)

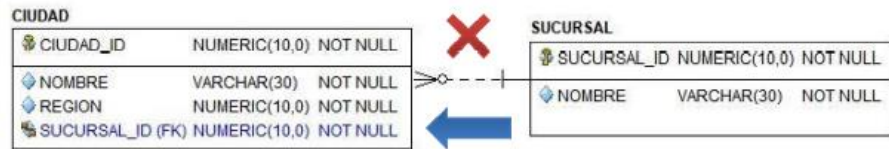
Ejemplo:

- ✓ Una sucursal se ubica en una ciudad. (1:1)
- ✓ En una ciudad pueden existir varias sucursales. (1:M)



14

¿Qué sucedería si la FK estuviera del lado de la tabla ciudad?



- ✓ Como se puede observar, a cada registro de ciudad solo se le puede asociar un registro de sucursal, pero, ¿Qué sucede si 2 sucursales están en la misma ciudad?
- ✓ Esta condición puede ocurrir, pero con la FK de esta forma, no es posible registrar los datos correspondientes.

15

Relaciones identificativas, fuertes o duras. :



- ✓ Consiste en relacionar 2 entidades a través de una línea **continua**. Se emplea para representar relaciones 1:1 y para modelar relaciones M:N
- ✓ Al asociar 2 tablas con una relación identificativa (línea continua) la llave primaria (PK) de la tabla padre pasa como llave primaria (PK) y también como llave foránea (FK) en la tabla hija.
- ✓ La diferencia con la relación no identificativa es que la (FK) forma parte de la PK de la tabla hija.
- ✓ Si la tabla hija no tiene una PK propia, la PK de la tabla padre y de la tabla hija **es compartida**.
- ✓ Por lo anterior y por la definición de PK, la relación entre ambas tablas es 1:1, un registro de la tabla padre, solo puede asociarse con un registro de la tabla hija.

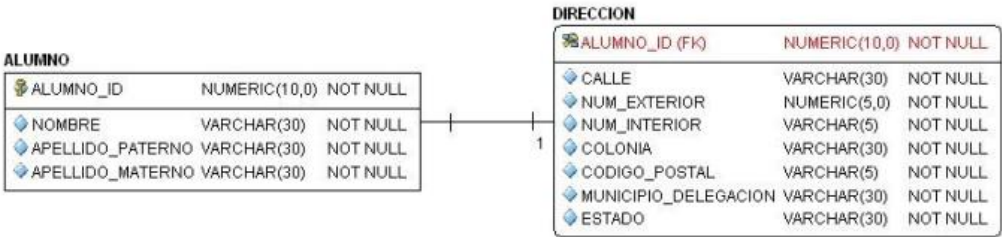
16

Ejemplo:



- ✓ Un alumno tiene una dirección. (1:1)
- ✓ Una dirección le pertenece a un alumno. (1:1)

Estrategia 1:

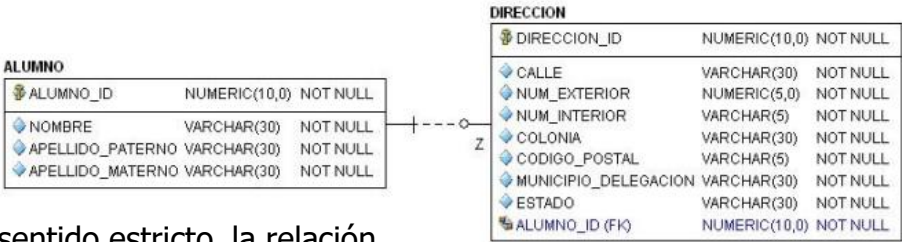


17

Estrategia 2:



El hecho de que la tabla dirección tenga una PK y FK alumno_id pareciera verse un poco extraña, no tan natural. Lo común es ver su propio identificador: direccion_id. Para respetar este estilo, se puede aplicar la siguiente variante:

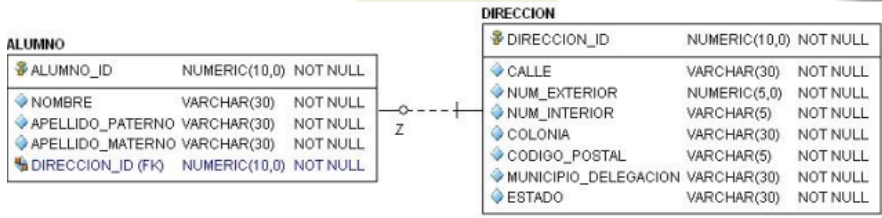


En sentido estricto, la relación anterior es un 1:M. :

DIRECCION	
direccion_id(pk)	alumno_id (fk)
1	1001
2	1001
3	1002
4	1002

18

¿Qué pasa si la FK se incluye del lado de la tabla alumno?



- ✓ Ahora podríamos decir que el campo direccion_id se puede duplicar, es decir, en una dirección pueden vivir varios alumnos

ALUMNO	
alumno_id (pk)	direccion_id (fk)
1001	1
1002	1
1003	2
1004	2

19

Estrategia 3



Otra solución es eliminar la tabla direccion y agregar todos sus campos a la tabla alumno. Esto es válido al tratarse de una relación 1:1. En realidad se decide separar en 2 tablas, cuando se desee tener una clara definición y separación de entidades. Si para el caso de estudio es relevante manejar ambas entidades por separado, se recomienda separarlas, de otra forma pueden fusionarse.

ALUMNO		
ALUMNO_ID	NUMERIC(10,0)	NOT NULL
NOMBRE	VARCHAR(30)	NOT NULL
APELLIDO_PATERNO	VARCHAR(30)	NOT NULL
APELLIDO_MATERNO	VARCHAR(30)	NOT NULL
CALLE	VARCHAR(30)	NOT NULL
NUM_EXTERIOR	NUMERIC(5,0)	NOT NULL
NUM_INTERIOR	VARCHAR(5)	NOT NULL
COLONIA	VARCHAR(30)	NOT NULL
CODIGO_POSTAL	VARCHAR(30)	NOT NULL
MUNICIPIO_DELEGACION	VARCHAR(30)	NOT NULL
ESTADO	VARCHAR(30)	NOT NULL

20

Cardinalidad



La cardinalidad expresa el número mínimo y máximo de ocurrencias de una entidad (instancias o registros) asociados con una ocurrencia (instancia o registro) de otra entidad.

A nivel general se pueden aplicar las siguientes preguntas para determinar los valores de cardinalidad:

- ¿Cuántas instancias de A como mínimo y máximo se asocian con una sola instancia de B?
- ¿Cuántas instancias de B como mínimo y máximo se asocian con una sola instancia de A?

Las preguntas anteriores se responden con base a las reglas de negocio del caso de estudio.

21

Ejemplo:

- Un profesor imparte como máximo 4 cursos.
- Un curso es impartido por un profesor.



A. Determinar los valores de cardinalidad para esta relación

- Para cada entidad se realizan las siguientes preguntas:

Entidad profesor:

- ¿Cuántas instancias de la entidad curso (min y máx.) se asocian con una instancia de profesor?

Respuesta: (1,4), un profesor imparte como mínimo 1 curso, máximo 4.

- El resultado se escribe del lado de la entidad contraria, en este caso al lado de la entidad curso.

Entidad curso:

- ¿Cuántas instancias de la entidad profesor (min y máx.) se asocian con una instancia de curso?

Respuesta: (1,1). Un curso es impartido mínimo por un profesor, máximo por 1.

- El resultado se escribe del lado de la entidad contraria, en este caso al lado de la entidad profesor.

22

Los datos se verían así:

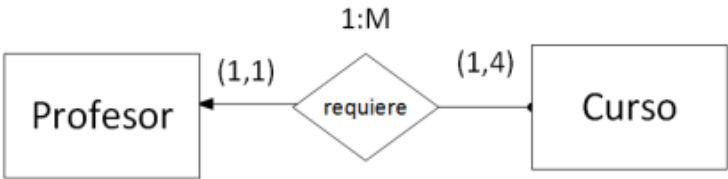


PROFESOR		
profesor_id	nombre	
1	JUAN	
2	LORENA	
3	MARCOS	

CURSO		
curso_id	nombre	profesor id
1	ALGEBRA	1
2	CALCULO	1
3	GEOMETRIA	1
4	ESTADISTICA	3

23

B. Construir el modelo ER incluyendo los valores de cardinalidad.



- Un profesor imparte mínimo 1, máximo 4 cursos **(1,4)**
- Un curso lo imparte mínimo , máximo 1 profesor **(1,1)**

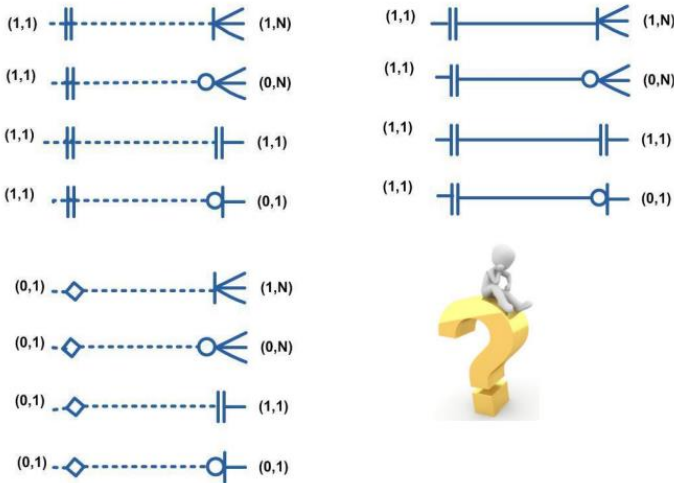
24

Cardinalidad en diseño lógico

- Similar al diseño conceptual, se emplea la notación (min, max) en ambos lados de las entidades.
- Adicionalmente las relaciones identificativas y no identificativas se complementan con las siguientes notaciones:

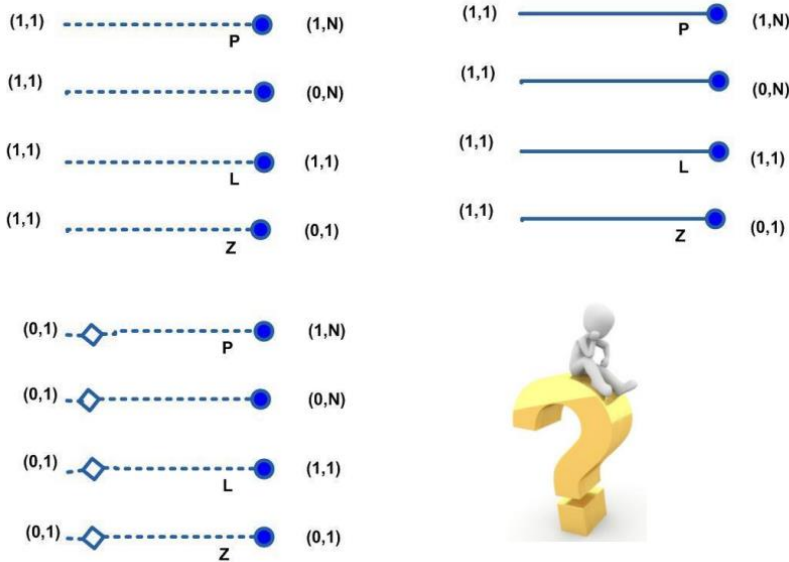


Cardinalidad con notación crow's Foot



25

Cardinalidad con notación IDEF1X

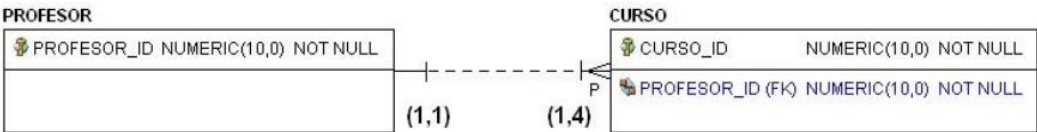


26

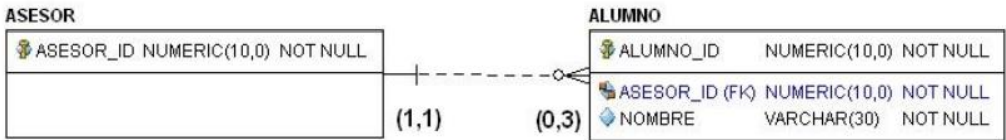
Ejemplo:



- **Un** profesor imparte como máximo **4** cursos.
- **Un** curso es impartido por **un** profesor.



- **Un** asesor si lo desea, puede ser asesor de hasta **3** alumnos.
- **Un** alumno debe contar con su asesor y es **uno** solo durante su estancia en la escuela.



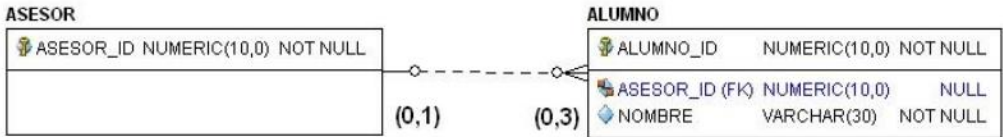
27

Ejemplo:



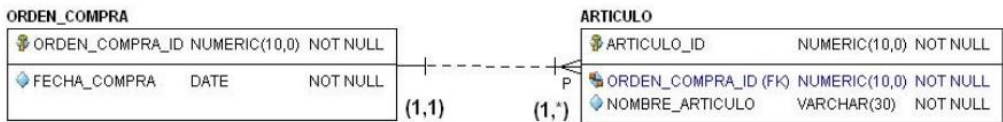
Suponer que se modifica la regla anterior:

- **Un** asesor si lo desea, puede ser asesor de hasta **3** alumnos.
- **Un** alumno puede solicitar **un** único asesor durante su estancia en la escuela.



Ejemplo:

- **Una** orden de compra está integrada por **varios** artículos.
- **Un** artículo pertenece a **una** orden de compra.



28

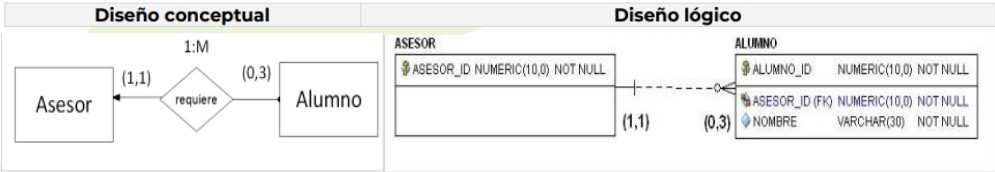
Dependencia de existencia



- Una entidad se considera dependiente de existencia cuando sus instancias o registros requieren la existencia de una instancia de la entidad con la que se relaciona.
- La dependencia de existencia siempre se verifica del lado de la entidad hija, la cual puede ser dependiente o independiente de existencia con respecto a la tabla padre.

Ejemplo:

- **Un** profesor, si lo desea, puede ser asesor de hasta **3** alumnos.
- **Un** alumno debe contar con su asesor y es **uno** solo durante su estancia en la escuela.

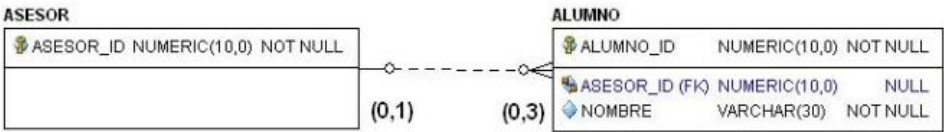


29

Ejemplo:



- **Un** profesor, si lo desea, puede ser asesor de hasta **3** alumnos.
- **Un** alumno puede solicitar **un** único asesor durante su estancia en la escuela.



La tabla alumno es independiente de existencia con respecto a asesor.

30

Participación de una entidad en una relación .



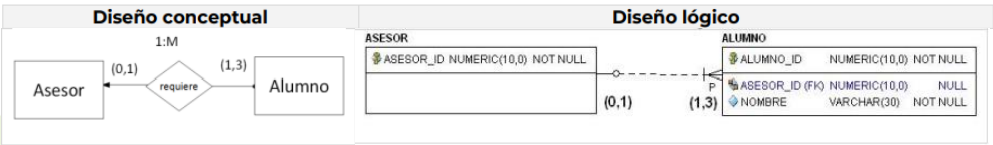
A diferencia de la dependencia de existencia, este concepto se verifica observando la cardinalidad en la tabla padre. La participación de una entidad en una relación puede ser obligatoria u opcional.

Participación obligatoria.

Una entidad padre tiene participación obligatoria en la relación con su entidad hija, cuando cada una de las instancias de la entidad padre se asocia con al menos una instancia de la entidad hija.

Ejemplo:

- Un profesor debe ser asesor de uno o hasta 3 alumnos.
- Un alumno puede solicitar un único asesor durante su estancia en la escuela.

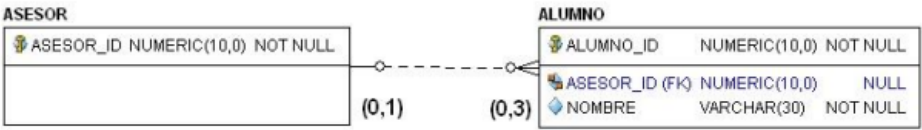


31

Ejemplo:



- Un profesor, si así lo desea, puede ser asesor máximo de 3 alumnos.
- Un alumno puede solicitar un único asesor durante su estancia en la escuela.



Observar que el valor mínimo de la cardinalidad del lado derecho en una participación opcional es 0. Nuevamente, no confundir con el concepto de dependencia de existencia, no importa la forma en la que se define la FK (**not null o null**).

32



Dependencia de identificación.

- Representa una extensión del concepto de dependencia de existencia.
- Este tipo de relación se verifica en la tabla hija al igual que en dependencia de existencia.
- La tabla hija define su propia PK, pero sus valores no son suficientes para poder garantizar unicidad, es decir, la tabla hija requiere de la PK de la tabla padre para funcionar correctamente, formando una PK compuesta.

33



Ejemplo:.

- Una librería cuenta con un catálogo de libros.
- Un libro se identifica de manera única con su ISBN
- Por cada ejemplar que se compra de dicho libro, se guarda un registro en base de datos identificándose por su numero iniciando en 1, por ejemplo, el ejemplar 1, del libro con ISBN 12, el ejemplar 2 del libro con ISBN 12, el ejemplar 1 del libro con el ISBN 13, etc.
- Se observa que el número de ejemplar se puede repetir por cada libro, es decir, el ejemplar 1 del libro 12, el ejemplar 1 del libro 13, etc.



34

Grado de una relación.



Consiste en clasificar a las relaciones por el número de entidades (no instancias) que participan en una relación:

- Relaciones unarias
- Relaciones binarias
- Relaciones ternarias

Relaciones unarias.

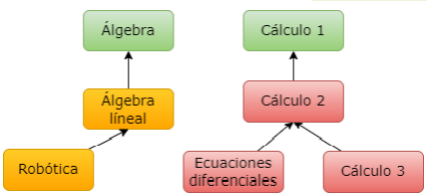
Ocurre al existir una relación empleando una sola entidad, es decir, una instancia de una entidad se relaciona con otra instancia de la misma entidad.
A este tipo de relaciones también se les conoce como relaciones recursivas.

35

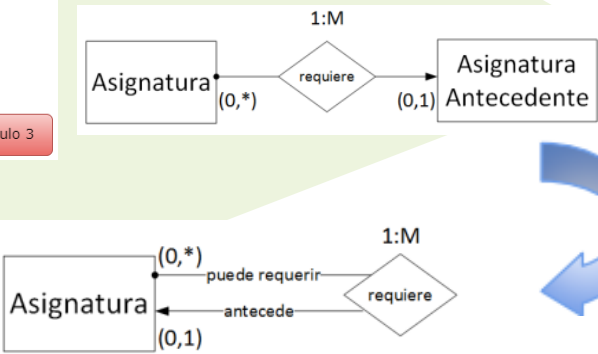
Ejemplo:.



- Una asignatura puede requerir de una asignatura antecedente para poder cursarla. (1:1)
- Una asignatura puede ser antecedente de varias asignaturas. (1:M)



Diseño conceptual:.



36

Diseño lógico:



- En una relación recursiva siempre habrá independencia de existencia. No todas las asignaturas (tabla hija) tienen asignatura antecedente (tabla padre). La FK es null

37

Diseño logico:

- ¿Qué pasaría si la FK se declara como not null? Toda asignatura deberá tener una asignatura antecedente. Sin embargo, ¿qué pasará con asignaturas como Álgebra o Cálculo 1 ? Ambas asignaturas no tienen una asignatura antecedente
- En una relación recursiva se puede tener ambos tipos de participación. En este caso, existe participación obligatoria. Toda asignatura antecedente (tabla padre) debe asociarse con al menos una asignatura subsecuente.

Los datos se verían así:

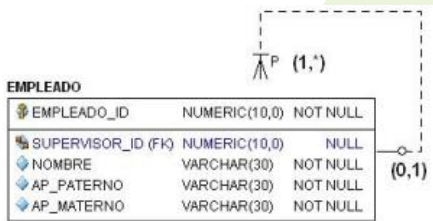
asignatura_id	nombre	creditos	asignatura_requerida_id
1	ALGEBRA	6	NULL
2	ALGEBRA LINEAL	7	1
3	CALCULO 1	6	NULL
4	CALCULO 2	7	3
5	CÁLCULO 3	7	4
6	ECUACIONES DIFERENCIALES	7	4
7	ROBÓTICA	8	2

38

Ejemplo:



- A cada empleado se le asigna su supervisor el cual también es un empleado.
- Un supervisor puede tener asignados a varios empleados.



- En este caso se tiene una independencia de existencia. No todos los empleados tienen supervisor, es decir, los supervisores en si no tienen supervisor.
- Participación: obligatoria. Un supervisor debe tener al menos un empleado asignado.

empleado_id	nombre	ap_paterno	ap_materno	supervisor_id
1	JUAN	MARTINEZ	AGUIRRE	4
2	LILIANA	LOPEZ	HURTADO	NULL
3	GERARDO	JIMENEZ	BENITEZ	2
4	MARIA	MONDRAGON	MENDEZ	NULL
5	JORGE	LUNA	AGUILAR	2

39

Relaciones binarias:

Ocurren entre 2 entidades diferentes.

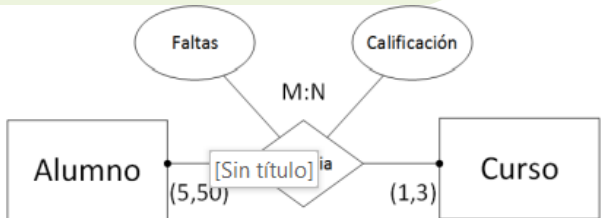


Relaciones ternarias:

Ocurren cuando se relacionan 3 entidades entre sí para poder cumplir una regla de negocio.

Suponer el siguiente escenario:

- Un curso está integrado mínimo de 5 alumnos, máximo de 50 (1:M)
- Un alumno puede tomar uno o hasta 3 cursos. (1:M)
- Se requiere registrar la calificación y el número de faltas que obtuvieron los alumnos.



40

Estrategia 1: PK compuesta:



- Observar en este caso la cardinalidad en ambos lados de la tabla intermedia:
 - Un alumno toma como mínimo un curso (participación obligatoria), y como máximo 3.
 - Un curso está formado por mínimo 5 alumnos (participación obligatoria), máximo 50.
- Observar que la dependencia de existencia no tiene sentido verificarla, las FKs forman parte de la llave primaria de la tabla intermedia, por lo tanto, no pueden ser definidas como NULL.
- En una relación M:N siempre existirá dependencia de existencia.
- La participación puede ser opcional u obligatoria.

41

Ocurren entre 2 entidades diferentes.

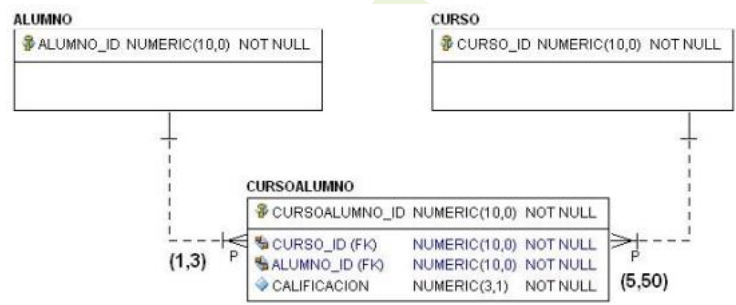
ALUMNO		CURSO	
alumno_id	nombre	curso_id	nombre
1	JUAN	1000	PROGRAMACIÓN
2	MARIO	1001	ALGORITMOS
3	LAURA	1002	INTEGRACION
4	LORENA	1003	COCINA

CURSO_ALUMNO		
curso_id(pk)(fk)	alumno_id(pk)(fk)	calificacion
1000	1	10
1001	1	9
1002	2	8
1002	3	7
1003	4	9

- Observar que la tabla intermedia permite implementar correctamente las reglas de negocio al incorporar una PK compuesta.
 - Un alumno puede tomar varios cursos (primeros 2 registros)
 - Un curso está formado por varios alumnos (registro 3 y 4)

42

Estrategia 2: PK artificial:



En esta estrategia se emplea una PK simple y artificial, y las 2 tablas se relacionan a través de relaciones no identificativas

Ventajas de esta estrategia:

- Al tener una PK artificial el desempeño tiende a mejorar ya que al realizar consultas y relacionar tablas, el procesamiento con llaves compuestas siempre es mayor al requerido para ligar tablas con PKs simples.
- Las consultas SQL requeridas para explotar los datos de esta tabla intermedia se simplifican con PKs simples (esto se verá en la parte de SQL)

43

Desventajas de esta estrategia:

- Se agrega un nuevo campo (la PK artificial), sin embargo, no representa un problema mayor en cuanto a espacio de almacenamiento.
- Posibilidad de insertar datos inconsistentes, situación que no ocurre con la estrategia anterior. En este caso, se le delega a la aplicación o al usuario insertar datos consistentes.



Ejemplo:

Los siguientes registros son inconsistentes:

CURSO_ALUMNO

curso_alumno_id(pk)	curso_id(fk)	alumno_id(fk)	calificacion
1	1000	1	10
2	1000	1	9

- En este caso, el registro es duplicado, y lo peor es que se tienen 2 calificaciones distintas para el mismo alumno y curso.
- Una solución para resolver este inconveniente es agregar un índice tipo **UNIQUE** que aplique a ambos campos, tanto a **curso_id** como a **alumno_id**

44



PRACTICA 2:

FECHA ENTREGA :

JUEVES 14/09/2023

HRS. 16:00 (hora de clase)

1. Para el enunciado de la ferretería FERRITER (Capítulo 3.1) realizar:

- A. Propuesta de diseño conceptual empleando un modelo ER
- B. Propuesta de un diseño lógico empleando un modelo relacional

45



2. Para el siguiente enunciado realizar:

- A. Propuesta de diseño conceptual empleando un modelo ER
- B. Propuesta de un diseño lógico empleando un modelo relacional

Se desea construir una pequeña base de datos para almacenar el registro de solicitudes de profesores candidatos para ocupar las vacantes de un colegio privado. Como parte inicial del proceso se requiere almacenar nombre y apellidos del candidato, número de cédula profesional, descripción de su título, año de titulación. Para establecer una comunicación segura, a cada aspirante se le permite registrar entre uno y hasta 3 correos electrónicos.

Se requiere registrar la lista de asignaturas que el profesor puede o desea impartir (álgebra, cálculo, etc.). Finalmente, en caso que el aspirante haya trabajado anteriormente en el colegio, se registran los números de contrato (folios de 5 dígitos) realizados en el pasado

46

Tipos de Datos empleados para realizar modelos relacionales.



Un punto importante dentro del diseño lógico es la selección del tipo de dato. Cuando se agrega cada una de las columnas de una tabla es indispensable indicar el tipo de dato que debe contener la columna.

Adicional a las restricciones que pudiera tener una columna en una tabla, es importante definir el tipo de dato y su dominio. Dicha definición debe reflejar de forma correcta la naturaleza de los datos:

- Valores monetarios
- Fechas
- Cantidades
- Nombres, etc

47

Clasificación de los tipos de datos SQL:



- Cadenas
 - Cadenas de caracteres
 - De longitud fija
 - De longitud variable
 - Cadenas binarias
- Numéricos
 - Exactos
 - Aproximados
- Tiempo y fechas
- Otros:
 - boolean (Estándar,MySQL, PostgreSQL)
 - ROWID (Oracle)
 - UROWID (Oracle) ○ BFILE (Oracle)
 - XML

48

