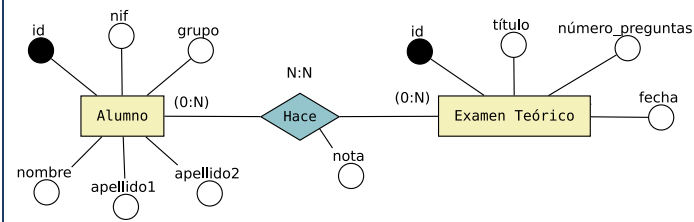


Entidad/Relación

Modelo Relacional

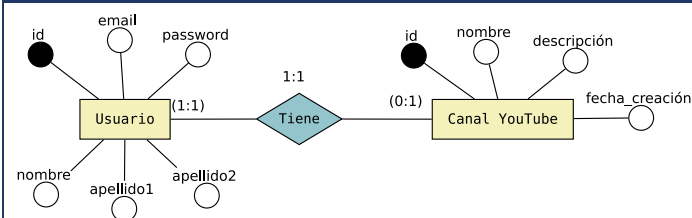
Entidades Fuertes y Débiles



Cada una de las entidades (fuertes y débiles) del diagrama E/R genera una tabla, donde cada uno de los atributos de la entidad pasa a ser una columna de la tabla.

ALUMNO(id, nombre, apellido1, apellido2, nif, grupo)
EXAMEN_TEÓRICO(id, título, número_preguntas, fecha)

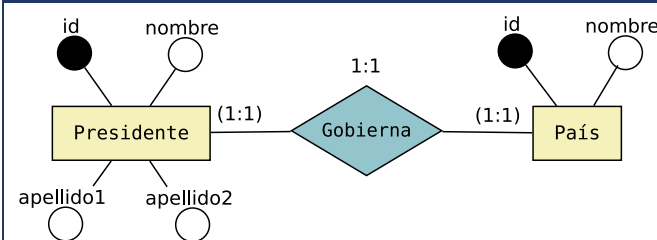
Relaciones con cardinalidad 1:1 - Participación (1,1)..(0,1)



Como la participación de **Usuario** es de (1,1) y la de **Canal YouTube** es de (0,1), la clave primaria de **Usuario** se almacena en la tabla de **Canal YouTube** como un atributo. El atributo *id_usuario* que se añade en la tabla **Canal YouTube** es una **clave ajena** o **foreign key** de la tabla **Usuario** columna *id*.

USUARIO(id, email, password, nombre, apellido1, apellido2)
CANAL_YOUTUBE(id, nombre, descripción, fecha_creación, *id_usuario*)
id_usuario: FOREIGN KEY de USUARIO(id)

Relaciones con cardinalidad 1:1 - Participación (1,1)..(1,1)



En este caso, como la participación de las dos entidades es de (1,1) podemos resolverlo de tres formas.

1. La clave primaria de **Presidente** se almacena en la tabla **País** como un atributo (*id_presidente*). El atributo *id_presidente* es una clave ajena o foreign key de la tabla **Presidente**.

PRESIDENTE(id, nombre, apellido1, apellido2)
PAÍS(id, nombre, *id_presidente*)
id_presidente: FOREIGN KEY de PRESIDENTE(id)

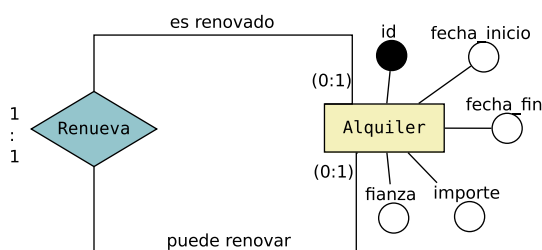
2. La clave primaria de **País** se almacena en la tabla **Presidente** como un atributo (*id_país*). Se dice que *id_país* es una clave ajena o foreign key de la tabla **País**.

PAÍS(id, nombre)
PRESIDENTE(id, nombre, apellido1, apellido2, *id_país*)
id_país: FOREIGN KEY de PAÍS(id)

3. Las claves primarias de ambas entidades se guardan en la tabla de la otra entidad. Esta solución puede presentar redundancia, pero puede ser interesante en algunas ocasiones, dependiendo de las consultas que se vayan a realizar sobre estas tablas a nivel de aplicación. En este caso los atributos *id_país* y *id_presidente* serían claves_ajenas o foreign key.

PRESIDENTE(id, nombre, apellido1, apellido2, *id_país*)
id_país: FOREIGN KEY de PAÍS(id)
PAÍS(id, nombre, *id_presidente*)
id_presidente: FOREIGN KEY de PRESIDENTE(id)

Relaciones con cardinalidad 1:1 – Participación (0,1)..(0,1)

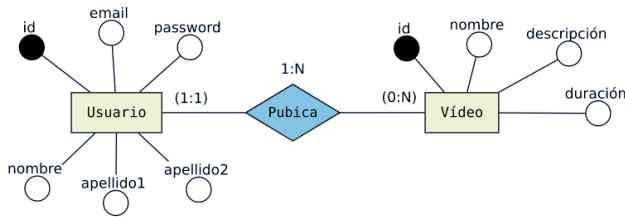


Cuando la participación de las dos entidades es de (0,1), se puede crear una nueva tabla donde se almacenan las claves primarias de las dos entidades que participan en la relación. La clave primaria de la nueva tabla será una de las dos claves ajenas que se reciben.

En este ejemplo tendríamos:

ALQUILER(id, fecha_inicio, fecha_fin, importe, fianza)
ALQUILER_RENUEVA_ALQUILER(id_alquiler, *id_alquiler_anterior*)
id_alquiler: FOREIGN KEY de ALQUILER(id)
id_alquiler_anterior: FOREIGN KEY de ALQUILER(id)

Relaciones con cardinalidad 1:N



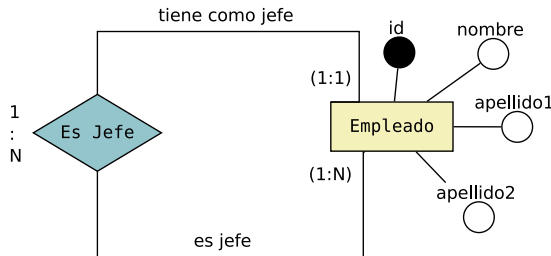
Las relaciones con cardinalidad 1:N no generan una tabla, lo que haremos será que la clave primaria de la entidad que participa con cardinalidad 1 pasará a formar parte de la tabla de la entidad que participa con cardinalidad N.

USUARIO(id, email, password, nombre, apellido1, apellido2)

VÍDEO(id, nombre, descripción, duración, id_usuario)

id_usuario: FOREIGN KEY de USUARIO(id)

Relaciones reflexivas con cardinalidad 1:N



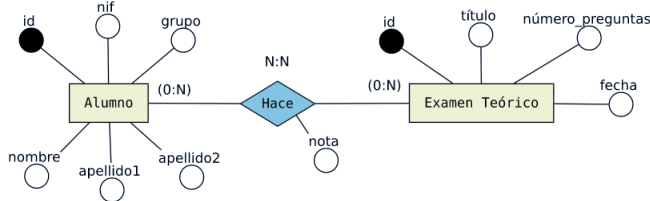
En este caso la clave primaria se almacena en la misma tabla como atributo.

La tabla Empleado vuelve a guardar su clave primaria como atributo haciendo referencia al id del jefe, le llamaremos id_jefe.

EMPLEADO(id, nombre, apellido1, apellido2, id_jefe)

id_jefe: FOREIGN KEY de EMPLEADO(id)

Relaciones con cardinalidad N:N



Las relaciones con cardinalidad N:N van a generar una nueva tabla. En este caso se crea una nueva tabla donde se almacenan las claves primarias de las dos entidades que participan en la relación. Las claves primarias de las entidades también serán claves primarias de la nueva tabla. Si la relación contiene algún atributo, se deberán añadir a la nueva tabla.

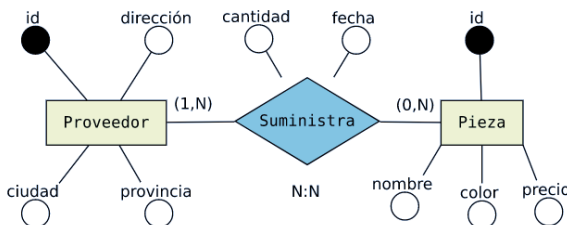
ALUMNO(id, nombre, apellido1, apellido2, nif, grupo)

EXAMEN_TEÓRICO(id, título, número_preguntas, fecha)

ALUMNO_HACE_EXAMEN_TEÓRICO(id_alumno, id_examen, nota)

id_alumno: FOREIGN KEY de ALUMNO(id)

id_examen: FOREIGN KEY de EXAMEN(id)



Habrán casos donde los atributos de la relación también formarán parte de la clave primaria de la nueva tabla. Estos casos aparecerán cuando en la relación existan atributos de tipo fecha y sea necesario almacenar un histórico de las relaciones entre las dos entidades en función de las fechas. Estos casos también pueden resolverse añadiendo un nuevo identificador de tipo entero con autoincremento en lugar de utilizar una clave primaria compuesta por varias columnas.

Solución 1

PROVEEDOR_SUMINISTRA_PIEZA(id_proveedor, id_pieza, fecha, cantidad)

id_proveedor: FOREIGN KEY de PROVEEDOR(id)

id_pieza: FOREIGN KEY de PIEZA(id)

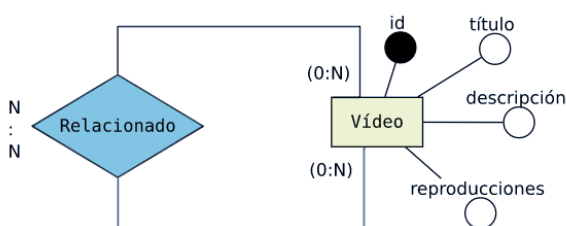
Solución 2

PROVEEDOR_SUMINISTRA_PIEZA(id, id_proveedor, id_pieza, fecha, cantidad)

id_proveedor: FOREIGN KEY de PROVEEDOR(id)

id_pieza: FOREIGN KEY de PIEZA(id)

Relaciones reflexivas con cardinalidad N:N



En este caso tendremos dos tablas en el modelo relacional:

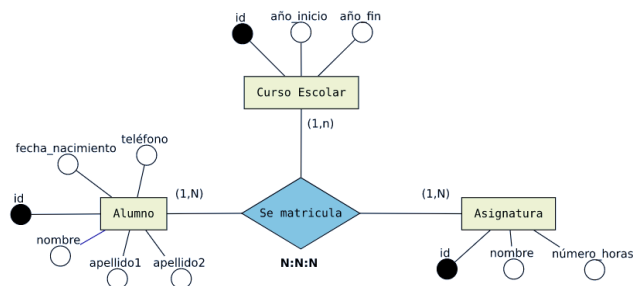
VÍDEO(id, título, descripción, reproducciones)

VÍDEOS_RELACIONADOS(id_video, id_video_relacionado)

id_video: FOREIGN KEY de VÍDEO(id)

id_video_relacionado: FOREIGN KEY de VÍDEO(id)

Relaciones grado 3 con cardinalidad N:N:N



Siempre que sea posible se recomienda convertir las relaciones de grado 3 en dos relaciones de grado 2. Las relaciones de grado 3 pueden generar una nueva tabla dependiendo de la cardinalidad de la relación.

En este caso creamos una tabla. La clave primaria de la nueva tabla estará formada por las tres claves de las entidades que participan en la relación.

Solución 1

ALUMNO_SE_MATRICULA_ASIGNATURA_CURSO(id_alumno, id_asignatura, id_curso_escolar)

id_alumno: FOREIGN KEY de ALUMNO(id)

id_asignatura: FOREIGN KEY de ASIGNATURA(id)

id_curso_escolar: FOREIGN KEY de CURSO_ESCOLAR(id)

Estos casos también pueden resolverse añadiendo un nuevo identificador de tipo entero con autoincremento en lugar de utilizar una clave primaria compuesta por varias columnas.

Solución 2

ALUMNO_SE_MATRICULA_ASIGNATURA_CURSO(id, *id_alumno*, *id_asignatura*, *id_curso_escolar*)

id_alumno: FOREIGN KEY de ALUMNO(id)

id_asignatura: FOREIGN KEY de ASIGNATURA(id)

id_curso_escolar: FOREIGN KEY de CURSO_ESCOLAR(id)

Relaciones grado 3 con cardinalidad 1:N:N

En este caso creamos una tabla. La clave primaria de la nueva tabla estará formada por las dos claves de las entidades que participan como N en la relación.

Relaciones de grado 3 con cardinalidad 1:1:N

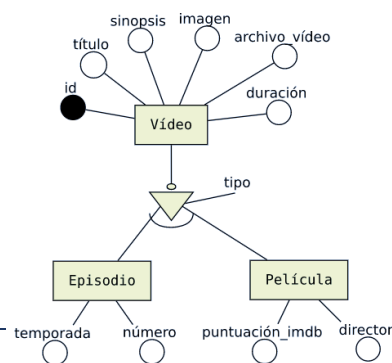
En este caso no es necesario crear una tabla. La entidad que participa como N recibe las claves de las dos entidades que participan como 1.

Generalización y Especialización (Relaciones ISA)

Existen varias soluciones para realizar el paso a tablas de una especialización. La solución que se elija en cada caso dependerá del tipo de especialización que estemos resolviendo: total, parcial, inclusiva o exclusiva.

Las 3 soluciones posibles que podemos aplicar son las siguientes:

1. Crear una única tabla para la superclase. En este caso todos los atributos de las subclases se guardarían en la superclase.
2. Crear una tabla sólo para las subclases. En este caso los atributos de la superclase habría que guardarlos en cada una de las subclases.
3. Crear una tabla para cada una de las entidades, tanto para la superclase como las subclases. En este caso las subclases tendrían que guardar la clave de la primaria de la superclase.



Solución 2: Crear una tabla sólo para las subclases.

EPISODIO(id, título, sinopsis, imagen, archivo_video, duración, temporada, número)

PELÍCULA(id, título, sinopsis, imagen, archivo_video, duración, puntuación_imdb, director)

Solución 3: Crear una tabla para cada una de las entidades.

VÍDEO(id, título, sinopsis, imagen, archivo_video, duración, **tipo**)

id: FOREIGN KEY de VÍDEO(id)

id: FOREIGN KEY de VÍDEO(id)

id: FOREIGN KEY de VÍDEO(id)

id: FOREIGN KEY de VÍDEO(id)

Solución 1: Crear una única tabla para la superclase.

LIBRO(id, título, isbn, año_publicación, descripción, tipo, lugar_impresión, fecha_impresión, precio_papel, tamaño_archivo, precio_ebook)

Solución 2: Crear una tabla para las subclases.

LIBRO_PAPEL(id, título, isbn, año_publicación, descripción, lugar_impresión, fecha_impresión, precio)

LIBRO_EBOOK(id, título, isbn, año_publicación, descripción, tamaño_archivo, precio)

id: FOREIGN KEY de LIBRO(id)

Solución 3: Crear una tabla para cada una de las entidades.

LIBRO(id, título, isbn, año_publicación, descripción)

LIBRO_PAPEL(id, lugar_impresión, fecha_impresión, precio)

id: FOREIGN KEY de LIBRO(id)

id: FOREIGN KEY de LIBRO(id)

id: FOREIGN KEY de LIBRO(id)