

BASES DE DATOS

Modelo Relacional

Modelo Relacional

Conceptos básicos

- El modelo relacional representa a una BD como una colección de tablas, las cuales se conforman por registros.
- Cada tabla se denomina relación y está formada por filas horizontales y columnas verticales. Cada fila representa un registro y se denomina tupla, mientras que cada columna representa un atributo del registro.

Modelo Relacional

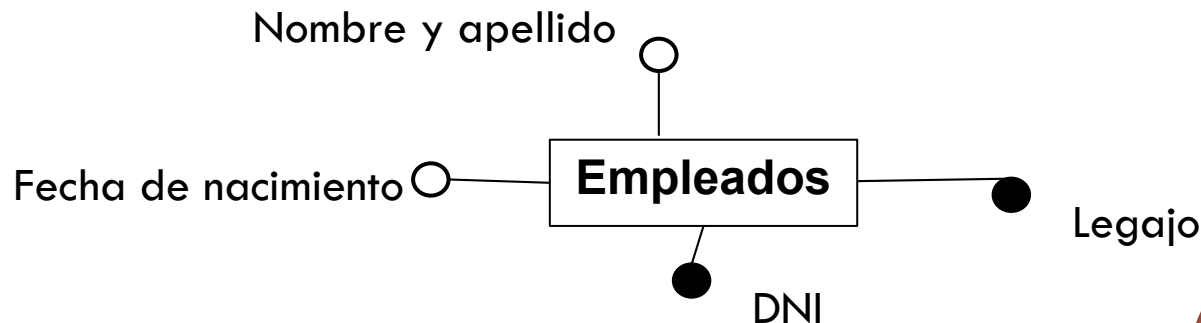
Selección de claves

- Si una entidad solo tiene un identificador, este identificador es clave primaria. Si la entidad tuviese varios identificadores, la selección de la **clave primaria (CP)** debe realizarse del siguiente modo:
 - Entre un identificador simple y uno compuesto, debe tomarse el simple.
 - Entre dos identificadores, se debe optar por aquel de menos tamaño físico.

Modelo Relacional

Selección de claves

- Las consideraciones anteriores definen, en general, el criterio más adecuado para elegir la CP. El resto de los identificadores será definido como **Clave Candidata (CC)**.



¿Qué
identificador será
clave primaria?

Modelo Relacional

Selección de claves

- Los DBMS actuales presentan una alternativa de tratamiento para las CP, a través del uso de un tipo de dominio denominado Autoincremental. El usuario sólo tiene permitida la operación de consulta sobre la CP autoincremental, no la puede generar, borrar ni modificar.

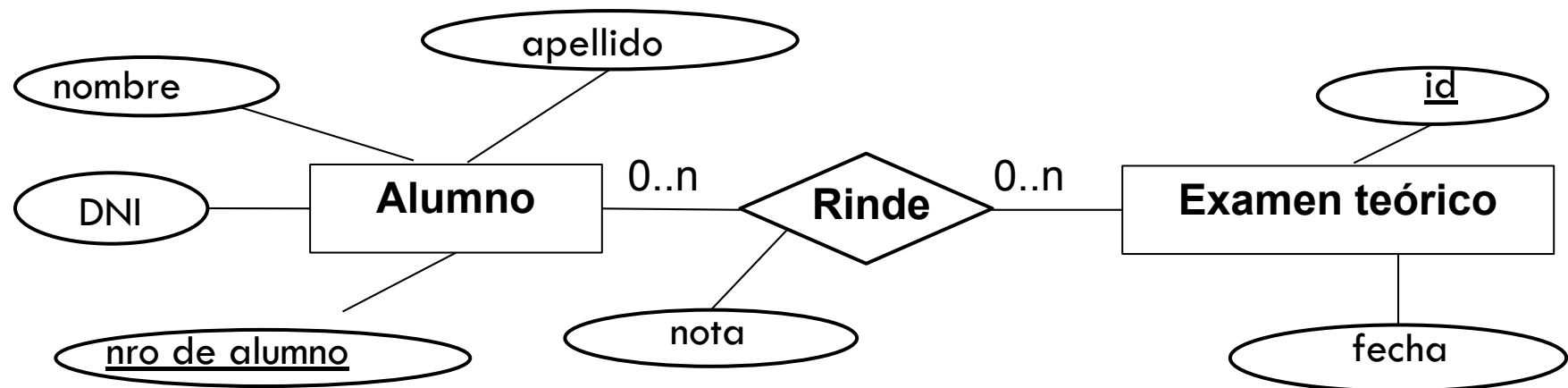
Modelo Relacional

Conversión de entidades

- El proceso de conversión para obtener el esquema físico de una BD comienza con el análisis de las entidades definidas en el modelo lógico.
- El proceso de conversión muestra que cada una de las entidades fuertes y débiles definidas deben transformarse en una tabla del modelo.

Modelo Relacional

Conversión de entidades



El proceso de conversión genera dos tablas, con las CP subrayadas:

Alumno = (nroAlumno, nombre, apellido, DNI)

Examen_teórico = (id, fecha)

Modelo Relacional

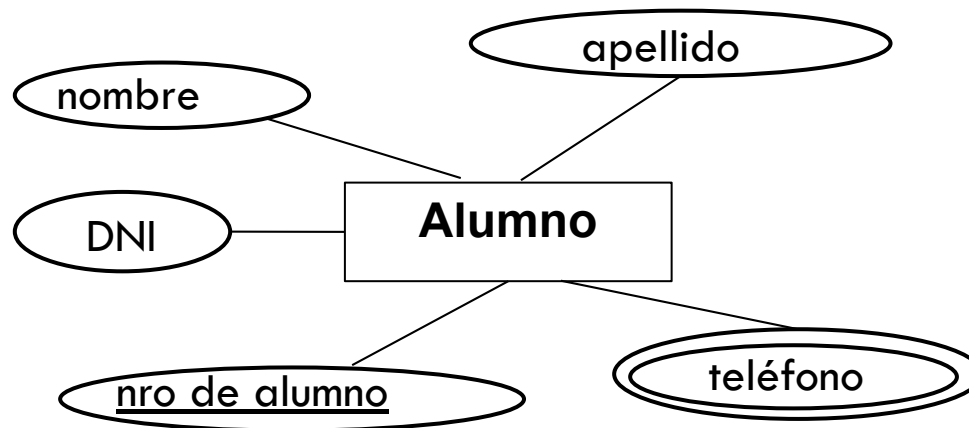
Conversión de atributos multivalorados

- Por cada atributo multivalorado (ya sea de entidad o de relación) se crea una tabla.
- Se agrega un atributo para el multivalorado.
- Se agrega como atributo la clave primaria de la tabla que modela la entidad o relación a la cual pertenece el multivalorado.
- La clave primaria se forma con todos sus atributos.

Modelo Relacional

9

Conversión de atributos multivalorados



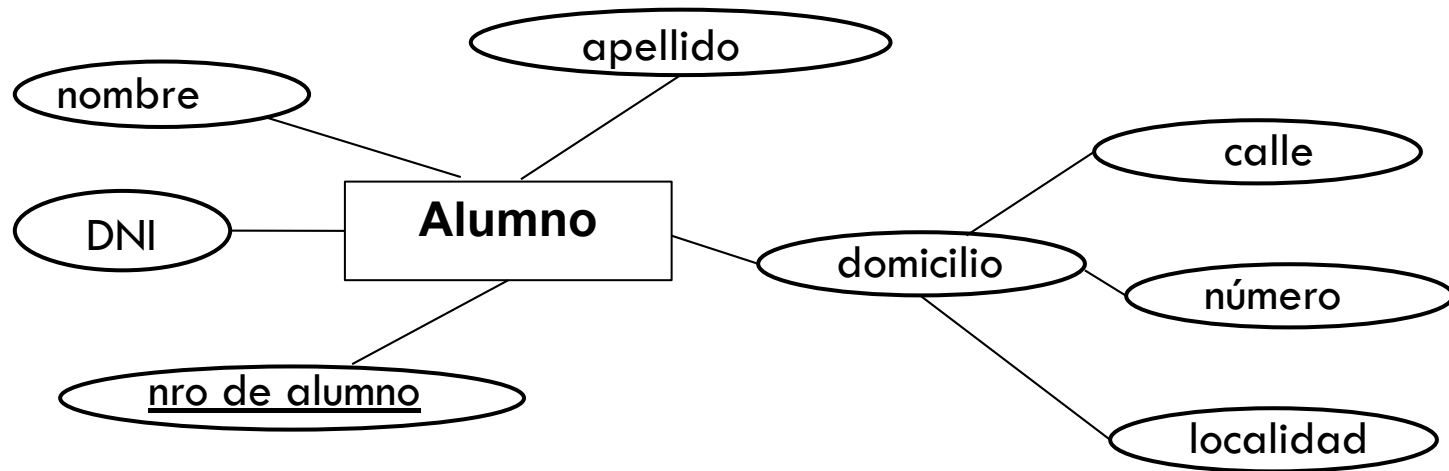
Alumno = (nroAlumno, nombre, apellido, DNI)

Teléfono = (nroAlumno, nroTeléfono)

Modelo Relacional

Conversión de atributos compuestos

- Se transforma en los atributos simples, desapareciendo este como tal de la entidad o relación.

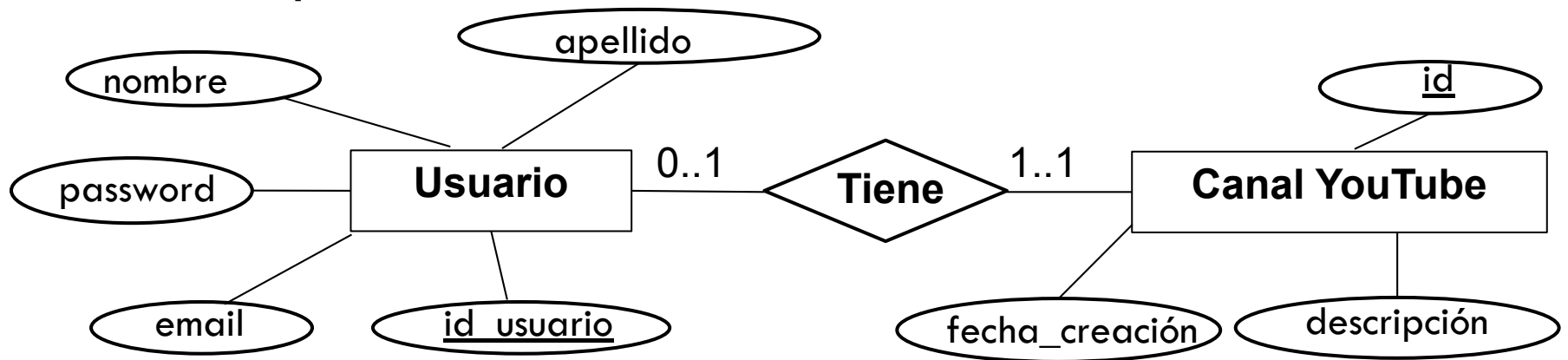


Alumno = (nroAlumno, nombre, apellido, DNI, calle, número, localidad)

Modelo Relacional

Conversión de relaciones

□ Participación 0..1 - 1..1



La clave primaria de Usuario se almacena en la tabla de Canal YouTube como un atributo. Se dice que el atributo `id_usuario` que se añade en la tabla `Canal_YouTube` es una clave ajena o foreign key (FK) de la tabla `Usuario`.

Usuario = (id_usuario, nombre, apellido, email, password)

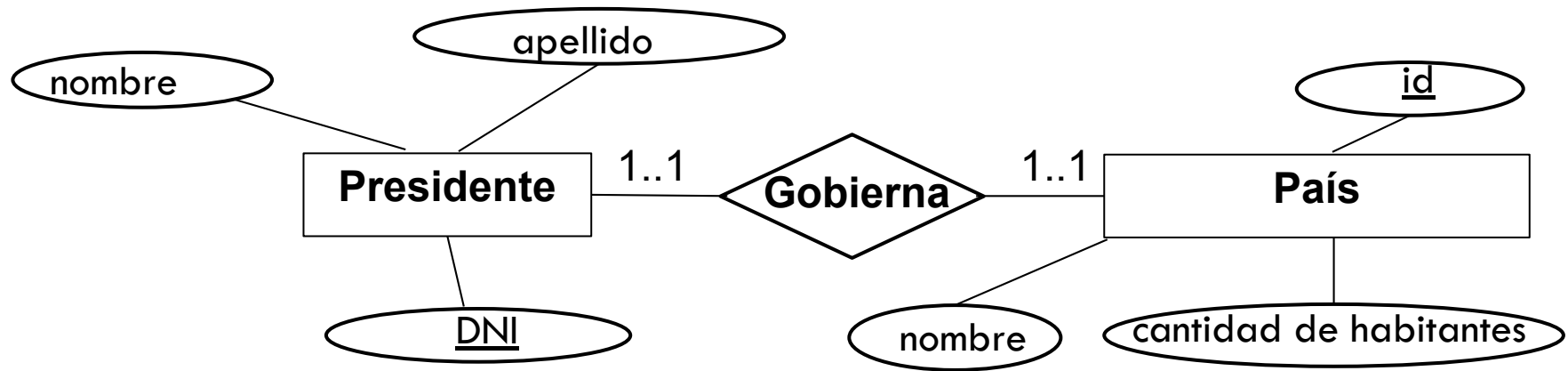
Canal_YouTube = (id, fecha_creación, descripción, *id_usuario*)

○ *id_usuario*: FOREIGN KEY de Usuario

Modelo Relacional

Conversión de relaciones

□ Participación 1..1 - 1..1



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

Modelo Relacional

Conversión de relaciones

1. La clave primaria de **Presidente** se almacena en la tabla **País** como un atributo y pasa a ser foreign key.

Presidente = (DNI, nombre, apellido)

País = (id, nombre, cantidad_habitantes, DNI)

- *DNI*: FOREIGN KEY de Presidente

2. La clave primaria de **País** almacena en la tabla **Presidente** como un atributo y pasa a ser foreign key.

País = (id, nombre, cantidad_habitantes, DNI)

Presidente = (DNI, nombre, apellido, id_país)

- *id_país*: FOREIGN KEY de País

Modelo Relacional

Conversión de relaciones

3. Las claves primarias de ambas entidades se guardan en la tabla de la otra entidad. Es decir, la tabla **Presidente** guardaría la clave primaria de **País** y la tabla **País** guardaría también la clave primaria de **Presidente**. 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 foreign key (FK).

Presidente = (DNI, nombre, apellido, id_país)

- `id_país`: FOREIGN KEY de País

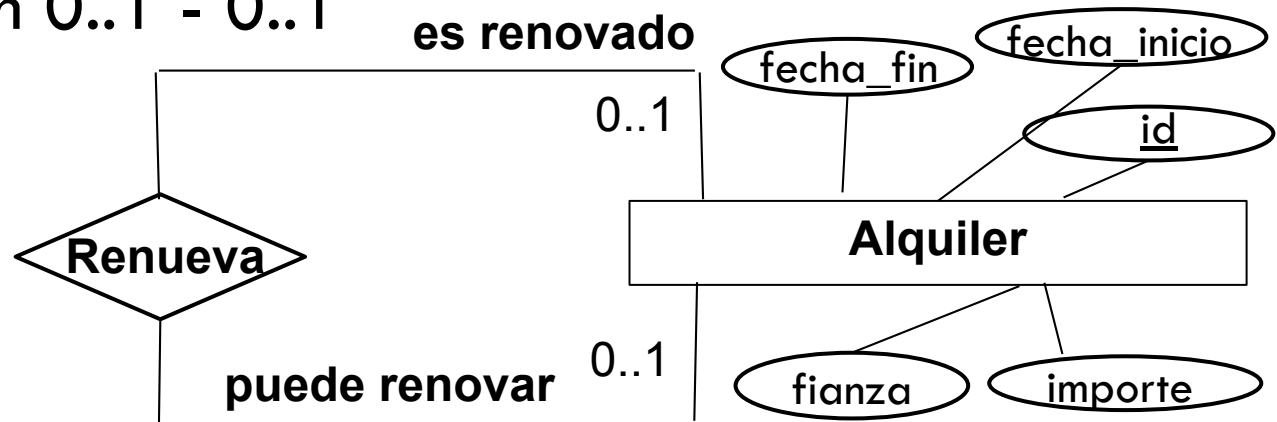
País = (id, nombre, cantidad_habitantes, DNI)

- `DNI`: FOREIGN KEY de Presidente

Modelo Relacional

Conversión de relaciones

□ 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

Alquiler = (id, fecha_inicio, fecha_fin, apellido, importe, fianza)

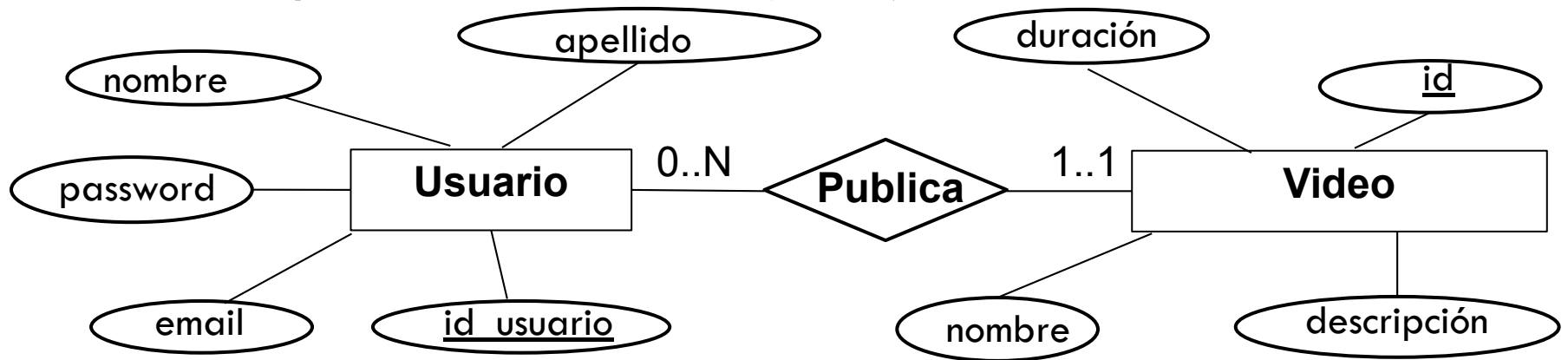
Alquiler_renueva_alquiler = (id_alquiler, id_alquiler_anterior)

- *id_alquiler*: FOREIGN KEY de Alquiler
- *id_alquiler_anterios*: FOREIGN KEY de Alquiler

Modelo Relacional

Conversión de relaciones

□ Participación 1..1 - 0..N (1..N)



En este caso se almacena la clave primaria en la tabla con participación 1..1

Usuario = (id_usuario, nombre, apellido, DNI, email, password)

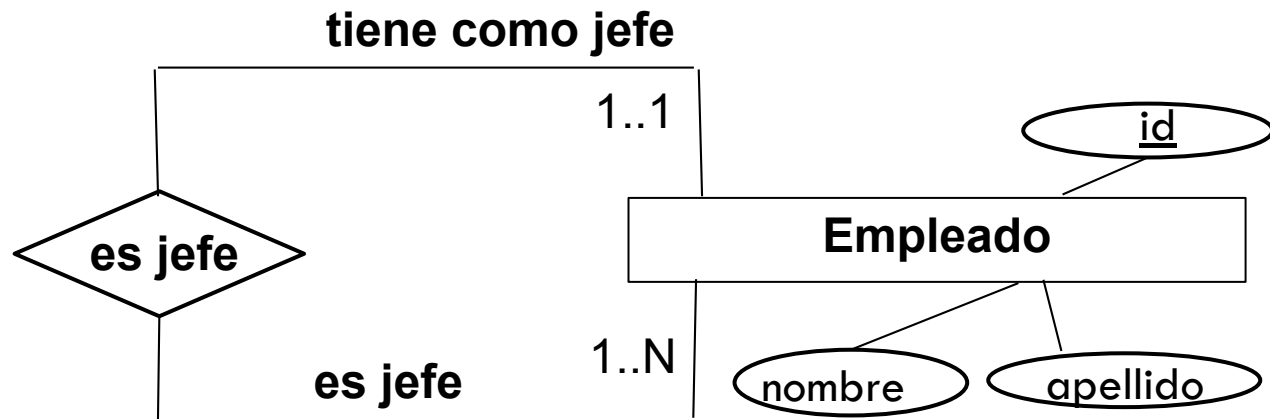
Video = (id, nombre, descripción, duración, id_usuario)

○ *id_usuario*: FOREIGN KEY de Usuario

Modelo Relacional

Conversión de relaciones

- Participación 1..1 - 0..N (1..N)



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

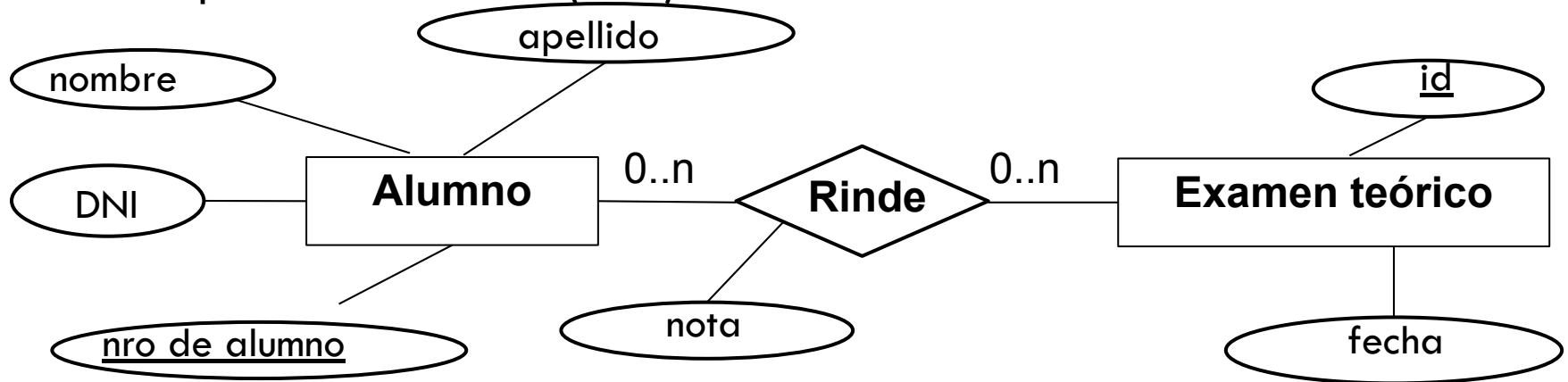
Empleado = (id, nombre, apellido, id_jefe)

- *id_jefe*: FOREIGN KEY de Empleado

Modelo Relacional

Conversión de relaciones

- Participación 0..N - 0..N (1..N)



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 en conjunto serán la clave primaria de la nueva tabla. Si la relación contiene algún atributo, se deberán añadir a la nueva tabla.

Alumno = (nroAlumno, nombre, apellido, DNI)

Examen_teorico = (id, fecha)

Alumno_hace_examen_teorico (nroAlumno, idExamen, nota)

- *nroAlumno*: FOREIGN KEY de Alumno
- *idExamen*: FOREIGN KEY de Examen_teorico

Modelo Relacional

Conversión de relaciones

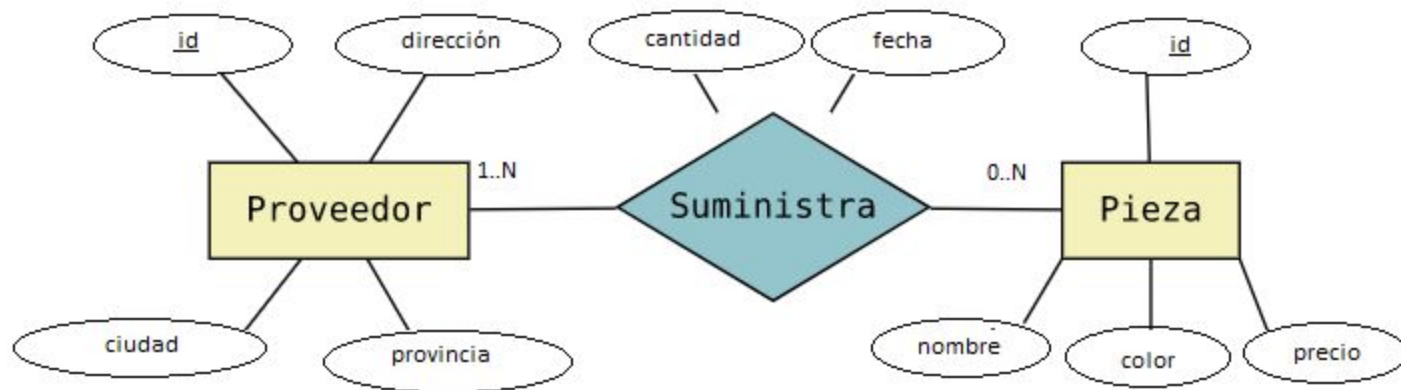
□ Participación 0..N - 0..N (1..N)

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.

Modelo Relacional

Conversión de relaciones

- Participación 0..N - 0..N

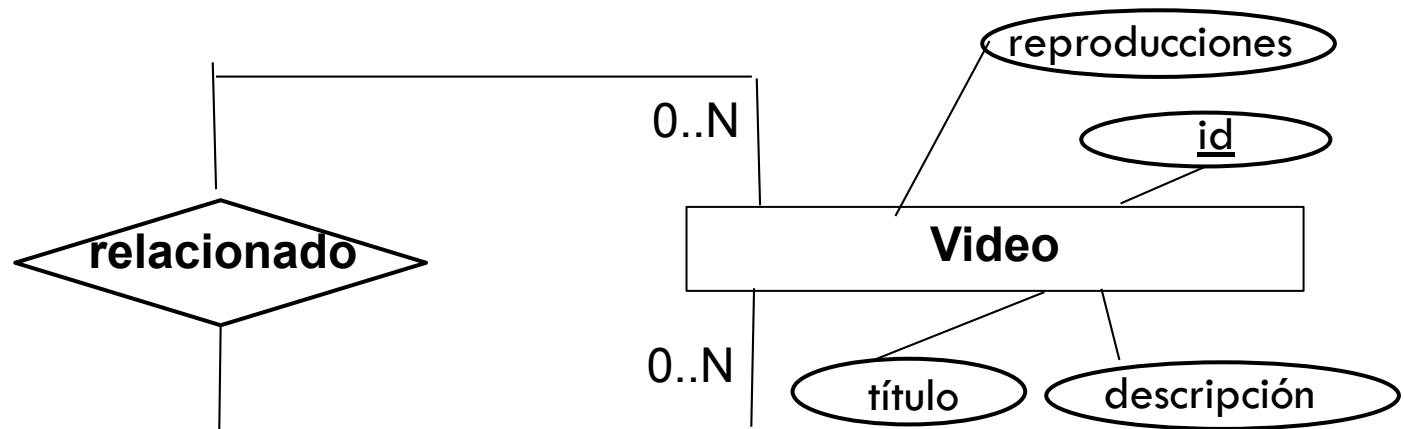


Cómo quedará formada la clave primaria para la tabla **Suministra** para poder permitir que un proveedor suministre piezas con el mismo id en fechas diferentes?

Modelo Relacional

Conversión de relaciones

- Participación 0..N - 0..N (1..N)



En este caso tendremos dos tablas:

Video = (id, título, descripción, reproducciones)

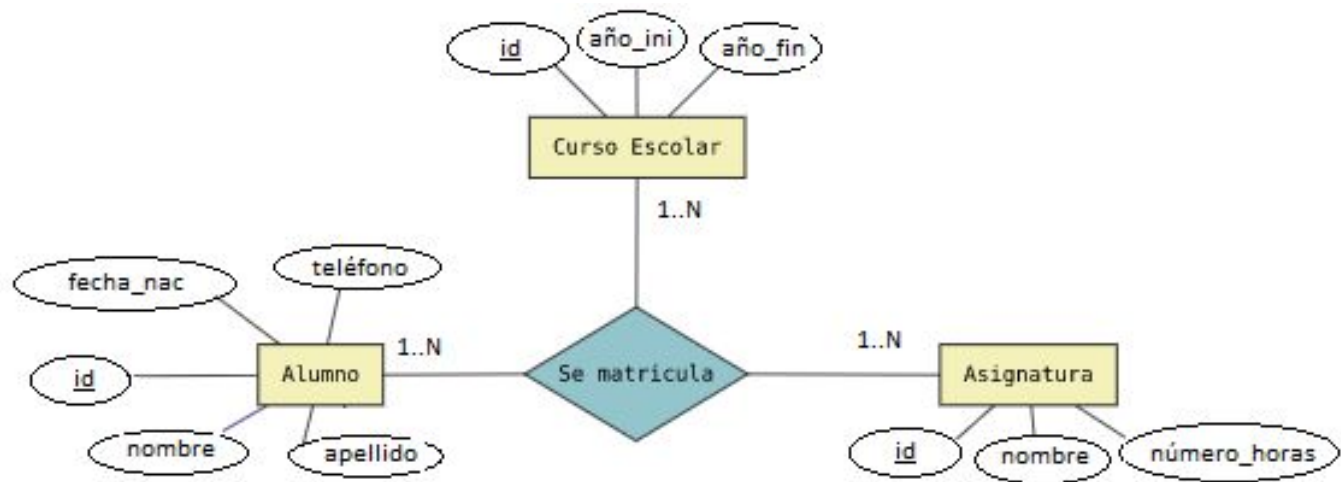
Video_relacionados (idVideo, idVideoRelacionado)

- *idVideo*: FOREIGN KEY de Video
- *idVideoRelacionado*: FOREIGN KEY de Video

Modelo Relacional

Conversión de relaciones

□ Relaciones de grado 3



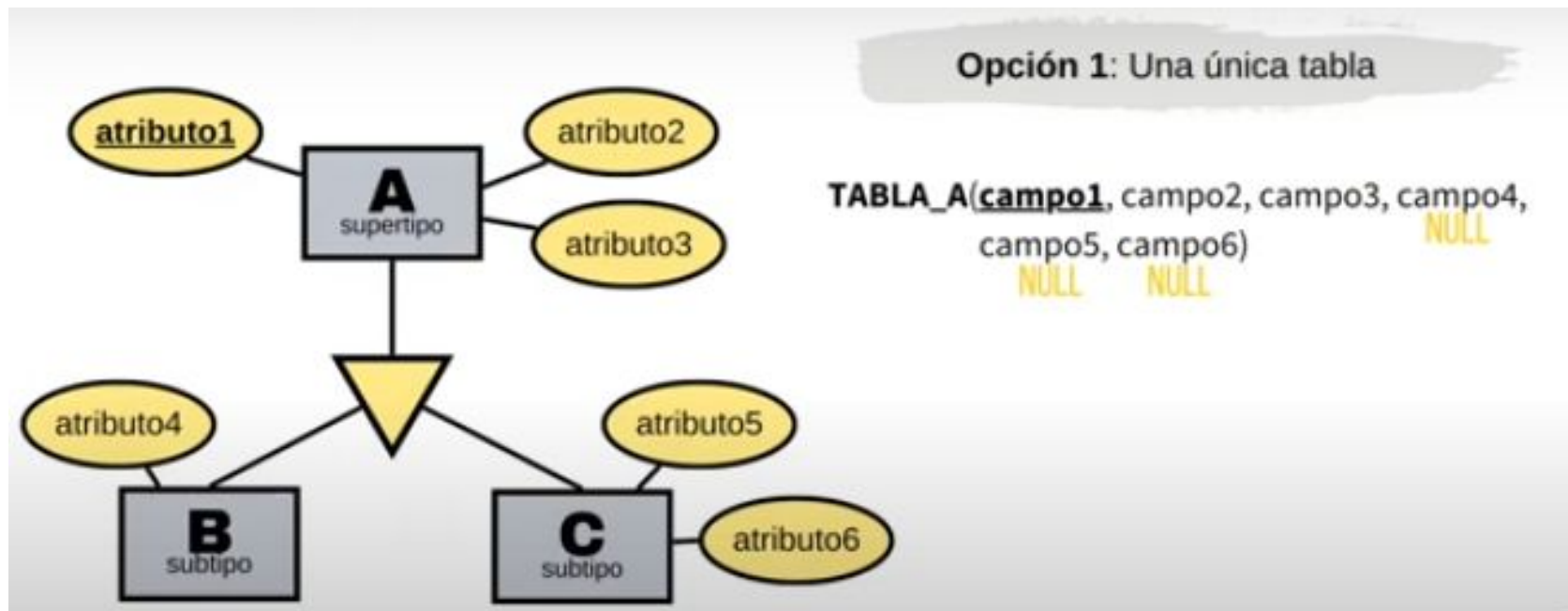
En este caso creamos una tabla. La clave primaria de la nueva tabla estará formada por las tres claves de las entidades que participan en la relación.

Modelo Relacional

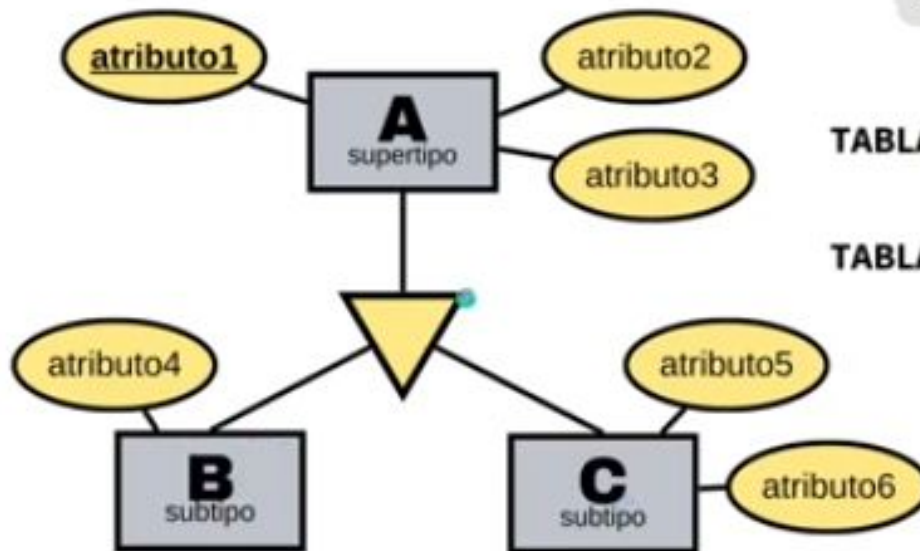
Generalizaciones/Especializaciones

- Básicamente, hay tres opciones para tratar una especialización.
 1. Eliminar las subentidades, dejando sólo la entidad padre a la cual se le incorporan todos los atributos de sus hijos. Cada uno de ellos deberá ser no obligatorio.
 2. Eliminar la entidad padre, dejando sólo las subentidades. Con esta solución, los atributos del padre deberán incluirse en cada uno de los hijos.
 3. Dejar todas las entidades de la jerarquía, convirtiéndolas en relaciones uno a uno entre el padre y cada uno de los hijos.

Modelo Relacional



Modelo Relacional

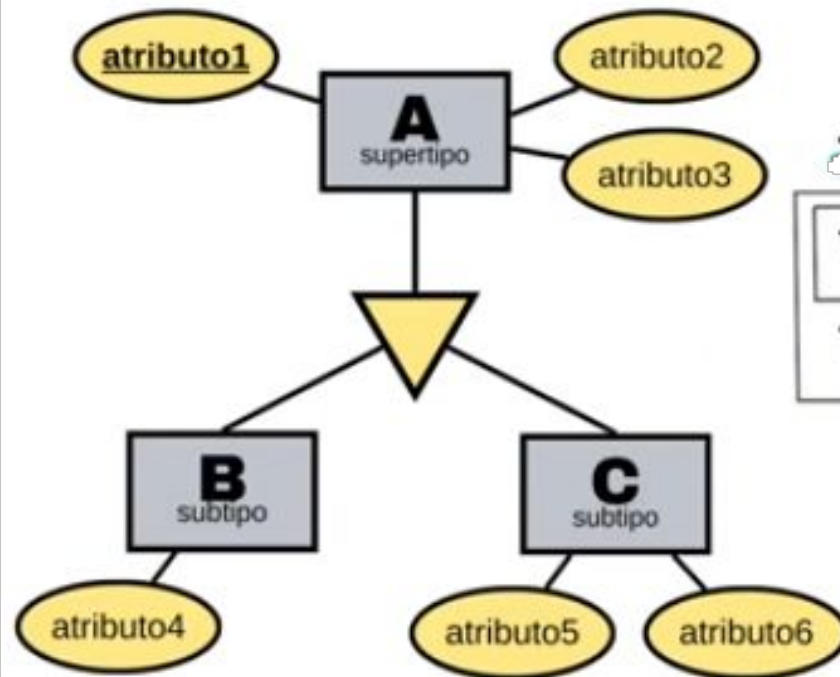


Opción 2: Una tabla para cada subtipo

TABLA_B (campo1, campo2, campo3, campo4)

TABLA_C(campo1, campo2, campo3, campo5, campo6)

Modelo Relacional



Opción 3: Una tabla para el supertipo y para cada subtipo

TABLA_A (campo1, campo2, campo3)

TABLA_B (campo1, campo4)

TABLA_C (campo1, campo5, campo6)

Modelo Relacional

Generalizaciones/Especializaciones

- Las tres soluciones no son aplicables en todos los casos.
- La cobertura es la que determina la solución viable en cada caso.
- Si la cobertura fuese parcial, la segunda solución no sería aplicable debido a que la conversión generaría un modelo no equivalente ya que se perdería información.

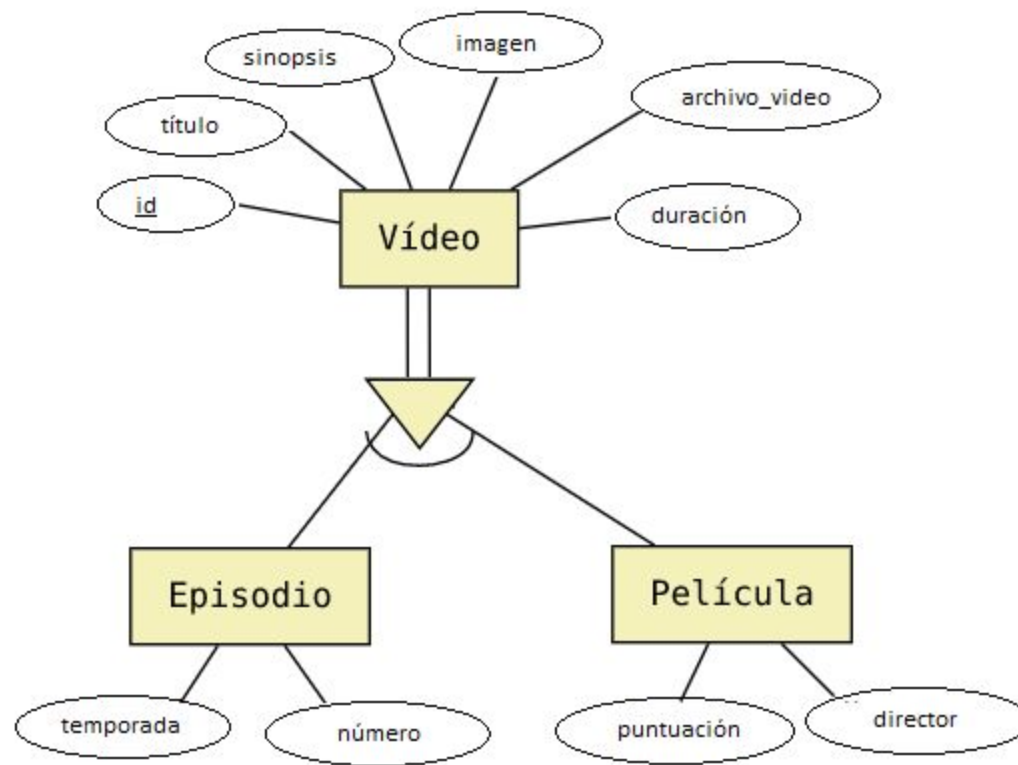
Modelo Relacional

Generalizaciones/Especializaciones

- Si se analiza la cobertura superpuesta, la segunda solución no resulta práctica. Algunos elementos del padre se repiten en varios hijos, esto significa que se deberá repetir información en las subentidades generadas.
- Se puede afirmar que la tercera alternativa de solución es la que capta mejor la esencia de la herencia y, por ende, la que resulta más interesante aplicar. Sin embargo, esta solución es la que genera mayor número de entidades y relaciones en el modelo final. Esto podría significar, a futuro, problema de performance en la utilización de la B.D.

Modelo Relacional

Generalizaciones/Especializaciones



En este caso sería viable aplicar la opción 2.

Modelo Relacional

Integridad referencial (IR)

- Propiedad deseable de las BD.
- Asegura que un valor que aparece para un atributo en una tabla, aparezca además en otra tabla para el mismo atributo.
- Plantea restricciones entre tablas y sirve para mantener la consistencia entre las tuplas de dichas tablas.

□ Ejemplo:

Facturas = (nroFactura, fecha, monto, nroCliente (CF))

Clientes = (nroCliente, nombre, dirección)

- En la tabla *Facturas*, nroCliente es una CF. Esta CF permite establecer IR entre las tablas *Facturas* y *Clientes*. Notar que nroCliente es CP en *Clientes*.

Modelo Relacional

Integridad referencial

- Cada DBMS presenta escenarios de definición de IR diferentes, pero en general cuando se la define se puede optar entre:
 - Restringir la operación: Si se intenta borrar o modificar una tupla que tiene IR con otra, la operación se restringe y no se puede llevar a cabo.
 - Realizar la operación «en cascada»: Si se intenta borrar o modificar una tupla donde está definida la CP de la IR, la operación se realiza en cadena sobre todas las tuplas de la tabla que tiene definida la CF.
 - Establecer la CF en nulo.
 - No hacer nada.

Modelo Relacional

Integridad referencial

- Establecer la CF en nulo: si se borra el valor del atributo que es CP, sobre la CF se establece valor nulo. Esta opción no es muy utilizada ni está presente en todos los DBMS.
- No hacer nada: en este caso se le indica al DBMS que no es necesario controlar la IR. Esta opción es equivalente a no definir restricciones de IR.