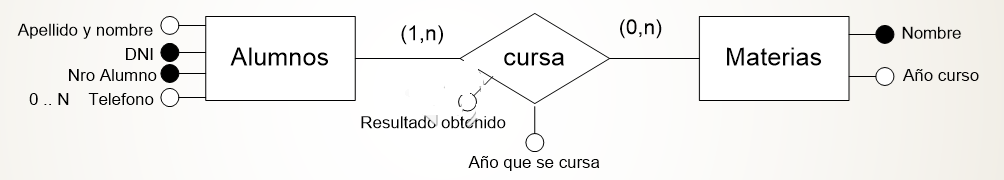
**Modelo Físico:**

***>> El modelo físico (relacional) representa la BD como una colección de relaciones:***

* Cada relación se parece a una tabla de valores, o a un archivo plano de registros.
* Un registro o un elemento de una relación (tabla) se denomina tupla.
* Un atributo mantiene su nombre.
* Cada tabla de valores resultante se llama relación.
  + Cada relación se obtiene a partir de una entidad o una relación ER 🡪 ósea que las entidades o las relaciones de ER son pasadas a tablas.
* El tipo de datos que describe los tipos de valores para un atributo se llama *dominio.*
* No se debe confundir la palabra “relación” del modelo físico con la del modelo conceptual ER, no son lo mismo.

**Pasos a seguir para convertir al modelo físico relacional 🡪 generación de tablas:**

1. Eliminación de identificadores externos 🡪 sí hay. La tabla solo puede tener identificadores internos.
2. Selección de claves (se vuelve a usar el concepto de clave):
   1. Primaría
   2. Candidata (unívoca)
3. Conversión de entidades.
4. Relaciones.



**Tablas:**

Alumnos = (apellidoynombre,dni,nroalumno) 🡪 teléfono no aparece xq se resolvió en el modelo lógico. 🡪 tiene dos identificadores, cuál elegimos como Clave Primaria (se trata con hash) y cuál como Clave Candidata (clave unívoca, se trata con arboles)?

* Podría preguntarme: ¿Cuál clave se va a buscar más?, y en base a la respuesta, elegir la respuesta como clave primaría (por el método de hash se va a conseguir más rápido el dato), el problema es que nos vamos a olvidar de hacer búsquedas secuenciales EFICIENTES usando dicha clave primaria.
* La clave primaria está propensa a errores (no está bueno).
* Desde fines del siglo pasado se propone que el atributo que sea clave primaría sea el atributo **AUTOINCREMENTAL.**
* ¿Qué significa esto?:
  + Significa definir en cada tabla un nuevo atributo que hasta ese momento no estaba presente en el modelo.
  + Dicho atributo va a ser manipulado por el DBMS (el DBMS le pone un valor NUNCA REPETIDO) 🡪 va a ser ideal porque: está manejado por la DBMS, permite búsquedas eficientes y está más allá del usuario (el usuario no sabe qué existe).
  + Por lo tanto, **LA CLAVE PRIMARÍA DEBE SER UN VALOR AUTOINCREMENTAL.**
  + Los identificadores restantes pasan a ser claves candidatas.

Materias = (nombre, añocurso)

**Después de haber explicado el atributo autoincremental:** (En azul subrayado: autoincremental, en rojo: candidata)

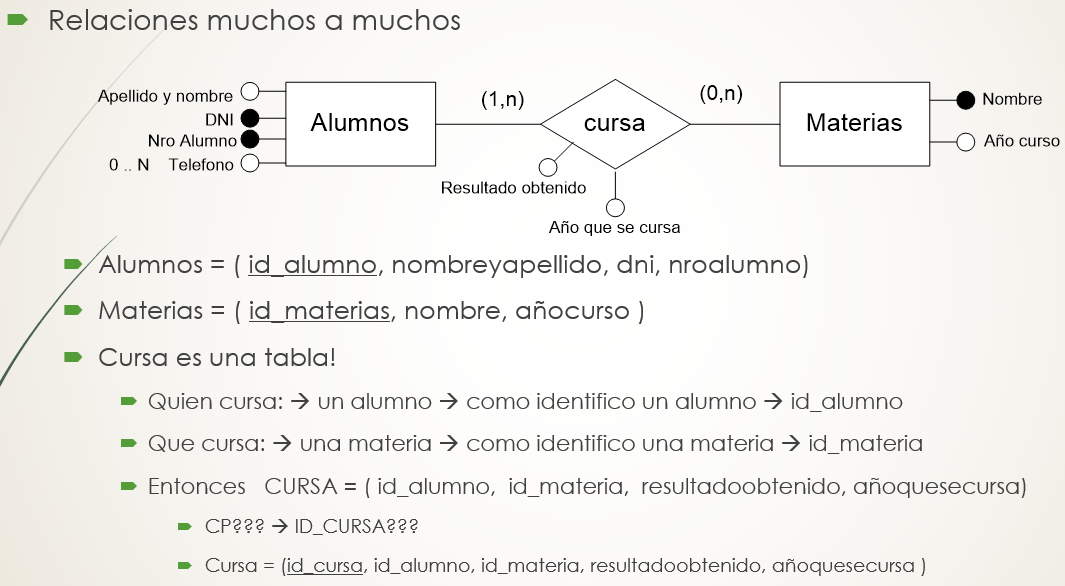
Alumnos = (id\_alumno, nombreyapellido, dni, nro\_alumno)

Materias = (id\_materia, nombre, añocurso)

¿Por qué se llama id\_nombre\_entidad\_singular? Id = identificador, \_ (espaciado).

¿Qué hacemos con las relaciones?

* Analizaremos:
  + Cardinalidad muchos a muchos.
  + Cardinalidad uno a muchos.
    - Clave foránea: atributo/s de una tabla que en otra tabla es/son CP y que sirven para establecer una unión entre ambas estructuras.
    - Cobertura total.
    - Cobertura parcial.
  + Relaciones recursivas.
  + Relaciones ternarias.



* Como cursa no tiene una clave unívoca, podemos concatenar en el modelo físico: Un alumno está en una materia en un año determinado (no se repite) 🡪 puedo formarlo como clave candidata: id\_alumno,id\_matería,añoquesecursa.
* ¿Es un buen nombre “Cursa” para una tabla?
  + Las relaciones cuando las pasamos a tabla podemos denominarlas concatenando las entidades que componen la relación.
  + Por lo tanto:
    - Alumnos\_Materias = (id\_alumnomateria, id\_alumno, id\_materia, resultadoobtenido, añoquesecursa) 🡪 anterior nombre: cursa.

**Clave foránea e integridad referencial**:

CF 🡪 atributo/s de una tabla que en otra tabla es/son CP y que sirven para establecer una unión entre ambas estructuras.

Ejemplo:

Alumnos = (ID\_ALUMNO, nombre y apellido, dni, nro alumno).

Materias = (ID\_MATERIA, nombre, añocurso).

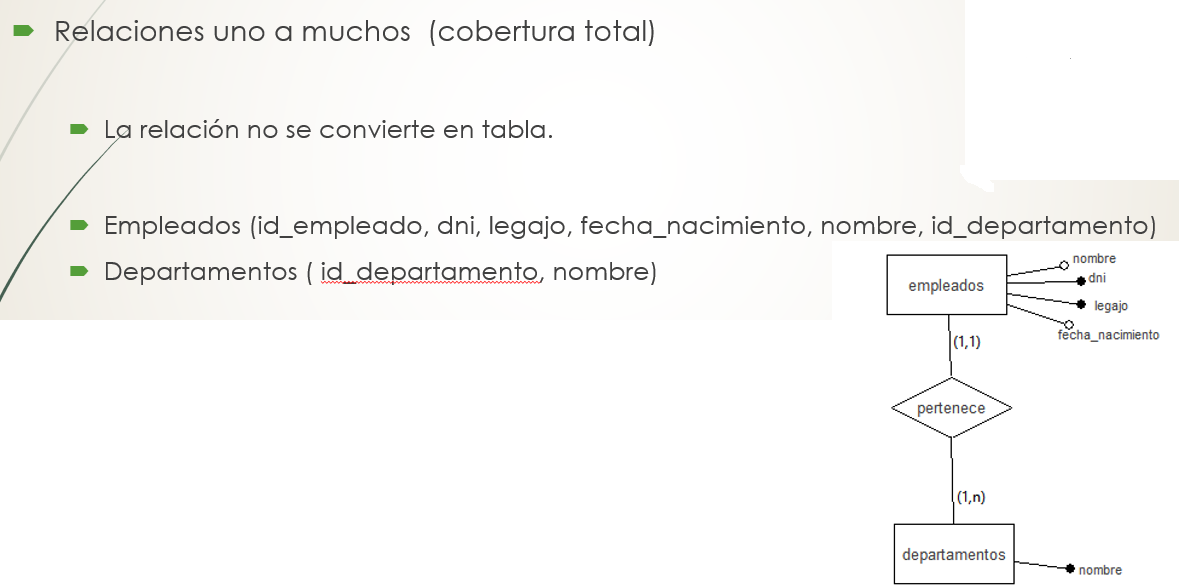
Alumnos\_Materias = (ID\_ALUMNOMATERIA, ID\_ALUMNO, ID\_MATERIA, resultadoobtenido, añoquesecursa) 🡪 ID\_ALUMNO, ID\_MATERIA SON CLAVES FORÁNEAS. Las claves foráneas suelen ser claves secundarias.

Concepto de integridad referencial:

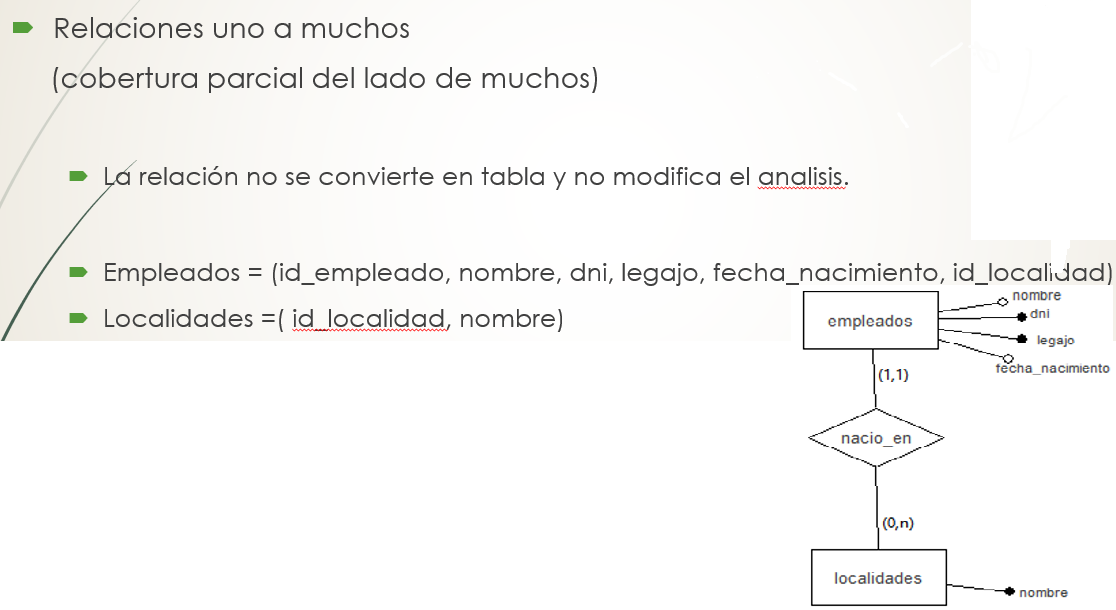
* Propiedad deseable de las BD.
* Asegura que un valor que aparece para un atributo en una tabla, aparezca además en otra tabla.
* La misma significa que la clave externa de una tabla de referencia siempre debe aludir a una fila válida de la tabla a la que se haga referencia. La integridad referencial garantiza que la relación entre dos tablas permanezca sincronizada durante las operaciones de actualización y eliminación.
* Se encarga de definir restricciones a lo que se puede y no se puede hacer.
* 4 tipos de IR:
  + Restringir operación.
  + Realizar operación en cascada.
  + Clave foránea en nulo.
  + No hacer nada.

Borrado en la integridad referencial (modificación en ANEXO):

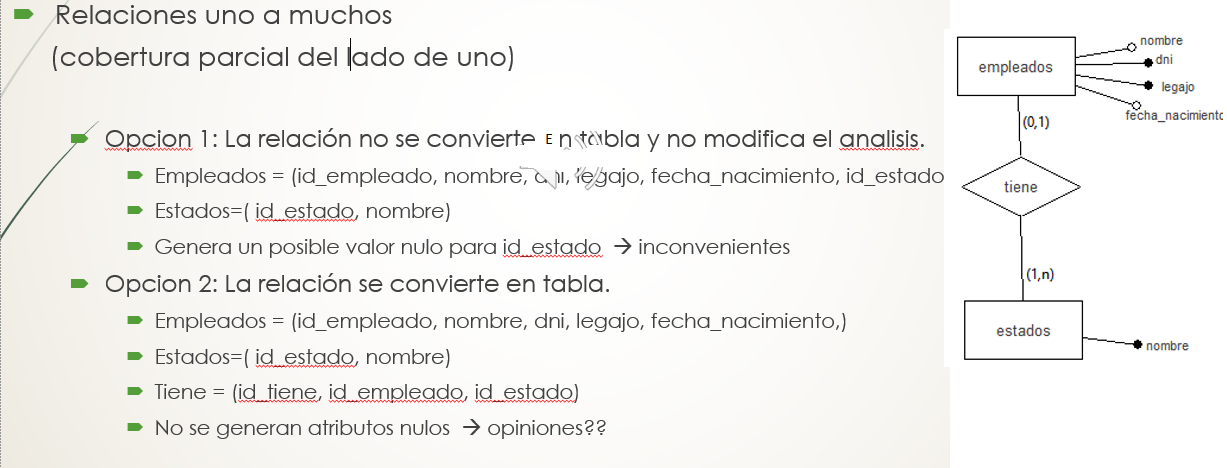
* Bajo los tipos de IR (son excluyentes una de otra, solo se pueden optar por una sola):
  + Restringir operación: si intento borrar una instancia de una entidad, y se prueba que dicha instancia corresponde por relación a datos existentes en otra entidad, entonces NO puedo realizar el borrado. Ejemplo: Borro la materia DBD a la cuál están inscriptos alumnos, no se puede realizar xq DBD está asociado a datos existentes en otra entidad.
  + Operación en cascada: no se restringe la operación de borrado (es una operación delicada 🡪 pero mantiene la integridad referencial). Imagina que borras la materia DBD a la cuál están inscriptos muchos alumnos. Entonces borras DBD y en cascada borras las inscripciones a la materia. Quitar algo y si éste tiene algo relacionado, ese algo se vá con él.
  + Establecer clave foránea en nulo: consiste en aceptar el borrado pero establecer los valores de clave foránea en nulo. Si borras DDB, todas las referencias relacionadas a DBD con clave foránea se ponen en nulo.
  + No hacer nada: permite operar como se desee, esto rompe la integridad referencial.
* *Anexo*: Si yo mantengo ID autoincremental 🡪 no puedo modificar nada, por ende no es necesario definir nada.



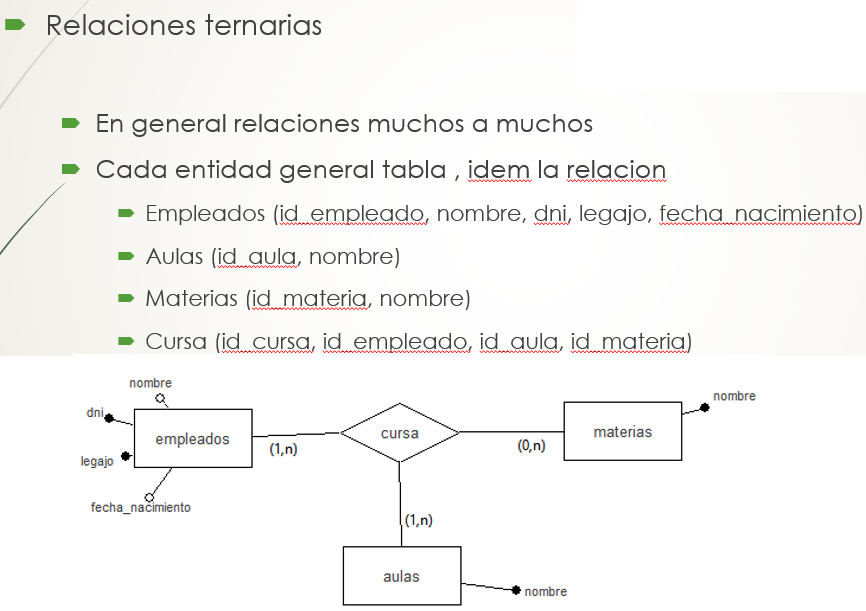
* El id\_departamento es único, la tupla de empleado ya indica a donde pertenece el empleado. La cobertura total indica (1,n) del lado de muchos (departamentos).

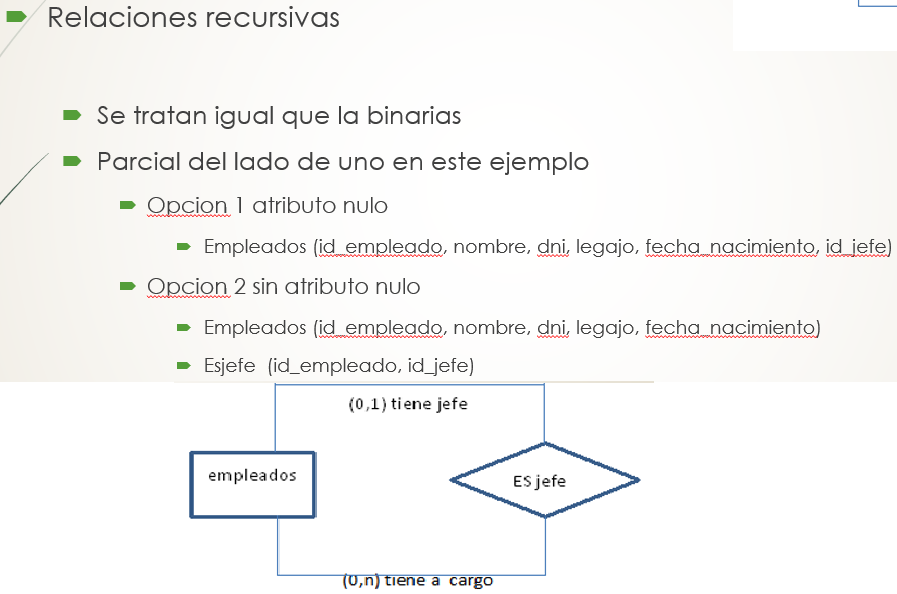


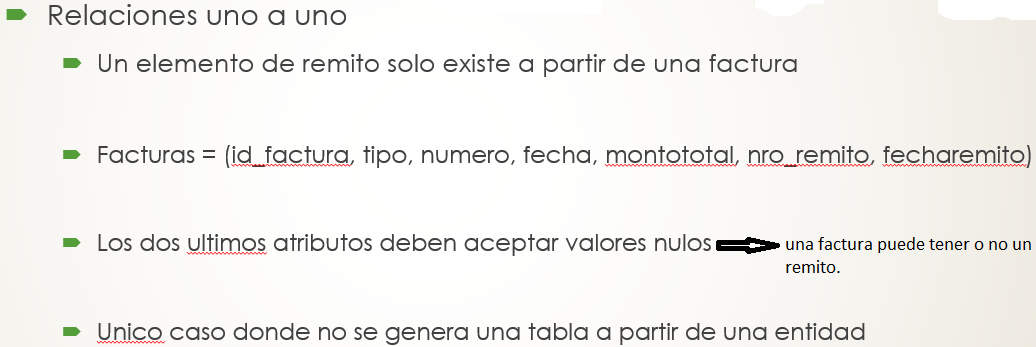
* Con parcial se refiere al 0 en el (0,n) de localidades.



* Opción 1, inconvenientes:
  + El atributo foraneo al ser opcional debe tener un posible valor nulo para id\_estado:
    - Los valores nulos si no son bien tratados pueden generar inconvenientes posteriores.
    - El nulo es nada, no tiene nada para comparar.
    - Es más eficiente.
* Opción 2:
  + Soluciona lo del valor nulo en la opción 1.
    - Genera más tablas, pero es más segura respecto de los valores nulos.







***Restricciones***

Restricciones de dominio:

* Especifican que el valor de cada atributo A debe ser un valor atómico del dominio de A. Ósea qué si mi atributo A es un tipo integer, me indica que sólo puedo ingresarle integers.

Restricciones de clave:

* Si yo defino un atributo como clave primaria o candidata, éste no puede tener valores repetidos.

Restricciones sobre nulos:

* Evita que un atributo tome nulo en caso de no ingresarle valor.

Restricciones de integridad:

* Ningún valor de la clave primaria puede ser nulo.

Restricción de integridad referencial:

* Especificada entre dos tablas, sirve para mantener consistencia entre ambas tuplas.
* Establece que una tupla en una tabla que haga referencia a otra relación deberá referirse a una tupla existente en esa tabla.
* Clave foránea está representada por un atributo de una relación que en otra es clave primaria.

Las operaciones de Alta,Baja y Modificación (ABM), pueden generar violaciones a las restricciones anteriores:

* Alta
  + Puede violar: valor nulo para clave, repetición de clave, integridad referencial, restricciones de dominio (insertar dato nuevo en un atributo con un tipo incompatible a lo insertado).
  + Si se viola la regla: la operación no se hace.
* Baja
  + Puede violar: integridad referencial (se procede como en caso anterior).
* Modificación
  + Puede violar: cualquiera de las operaciones.