

1.- Diagramas de estructuras de datos en el modelo E-R

Los diagramas Entidad-Relación representan la estructura lógica de una BD de manera gráfica. Los símbolos utilizados son los siguientes:

- Rectángulos para representar a las entidades.
- Elipses para los atributos. El atributo que forma parte de la clave primaria va subrayado.
- Rombos para representar las relaciones.
- Las líneas, que unen atributos a entidades y a relaciones, y entidades a relaciones.

Si la flecha tiene punta, en ese sentido está el uno, y si no la tiene, en ese sitio está el muchos. La orientación señala cardinalidad.

- Si la relación tiene atributos asociados, se le unen a la relación.
- Cada componente se etiqueta con el nombre de lo que representa.

En la Figura 1.4 se muestra un diagrama E-R correspondiente a PROVEEDORES-ARTÍCULOS. Un PROVEEDOR SUMINISTRA muchos ARTÍCULOS.



Figura 1.4. Diagrama E-R, un proveedor suministra muchos artículos.

2.- Grado y cardinalidad de las relaciones con ejemplos:

Se define **grado de una relación** como el número de conjuntos de entidades que participan en el conjunto de relaciones, o lo que es lo mismo, el número de entidades que participan en una relación. Las relaciones en las que participan dos entidades son binarias o de grado dos. Si participan tres serán ternarias o de grado 3. Los conjuntos de relaciones pueden tener cualquier grado, lo ideal es tener relaciones binarias.

Las relaciones en las que sólo participa una entidad se llaman anillo o de grado uno; relaciona una entidad consigo misma, se las llama relaciones reflexivas. Por ejemplo, la entidad EMPLEADO puede tener una relación JEFE DE consigo misma: un empleado es JEFE DE muchos empleados y, a la vez, el jefe es un empleado.

Otro ejemplo puede ser la relación DELEGADO DE los alumnos de un curso: el delegado es alumno también del curso. Ver Figura 1.5.



Figura 1.5. Relaciones de grado 1.

En la Figura 1.6 se muestra una relación de grado dos, que representa un proveedor que suministra artículos, y otra de grado tres, que representa un cliente de un banco que tiene varias cuentas, y cada una en una sucursal:

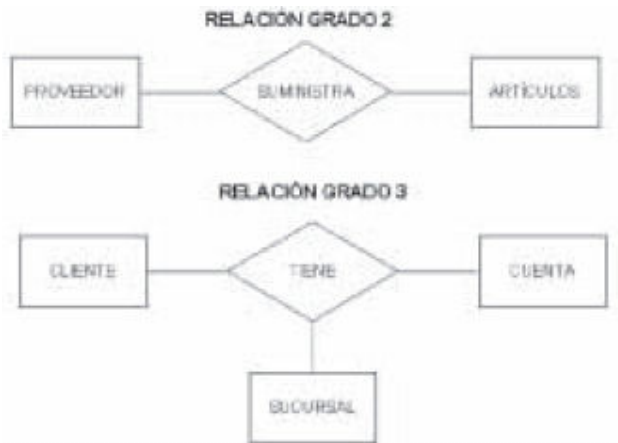


Figura 1.6. Relaciones de grados 2 y 3.

En el modelo E-R se representan ciertas restricciones a las que deben ajustarse los datos contenidos en una BD. Éstas son las restricciones de las cardinalidades de asignación, que expresan el número de entidades a las que puede asociarse otra entidad mediante un conjunto de relación.

Las cardinalidades de asignación se describen para conjuntos binarios de relaciones. Son las siguientes:

- **1:1, uno a uno.** A cada elemento de la primera entidad le corresponde sólo uno de la segunda entidad, y a la inversa. Por ejemplo, un cliente de un hotel ocupa una habitación, o un curso de alumnos pertenece a un aula, y a esa aula sólo asiste ese grupo de alumnos. Ver Figura 1.7:



Figura 1.7. Representación de relaciones uno a uno.

- **1:N, uno a muchos.** A cada elemento de la primera entidad le corresponde uno o más elementos de la segunda entidad, y a cada elemento de la segunda entidad le corresponde uno sólo de la primera entidad. Por ejemplo, un proveedor suministra muchos artículos (ver Figura 1.8).



Figura 1.8. Representación de relaciones uno a muchos.

- **N:1, muchos a uno.** Es el mismo caso que el anterior pero al revés; a cada elemento de la primera entidad le corresponde un elemento de la segunda, y a cada elemento de la segunda entidad, le corresponden varios de la primera.

- **M:N, muchos a muchos.** A cada elemento de la primera entidad le corresponde uno o más elementos de la segunda entidad, y a cada elemento de la segunda entidad le corresponde uno o más elementos de la primera entidad. Por ejemplo, un vendedor vende muchos artículos, y un artículo es vendido por muchos vendedores (ver Figura 1.9).



Figura 1.9. Representación de relaciones muchos a muchos.

La cardinalidad de una entidad sirve para conocer su grado de participación en la relación, es decir, el número de correspondencias en las que cada elemento de la entidad interviene. Mide la obligatoriedad de correspondencia entre dos entidades.

La representamos entre paréntesis indicando los valores máximo y mínimo: (máximo, mínimo). Los valores para la cardinalidad son: (0,1), (1,1), (0,N), (1,N) y (M,N). El valor 0 se pone cuando la participación de la entidad es opcional.

En la Figura 1.10, que se muestra a continuación, se representa el diagrama E-R en el que contamos con las siguientes entidades:

- **EMPLEADO** está formada por los atributos N°. Emple, Apellido, Salario y Comisión, siendo el atributo N°. Emple la clave principal (representado por el subrayado).

- **DEPARTAMENTO** está formada por los atributos N°. Depart, Nombre y Localidad, siendo el atributo N°. Depart la clave principal.

- Se han definido dos relaciones:

- La relación «**PERTENECE**» entre las entidades **EMPLEADOS** y **DEPARTAMENTO**, cuyo tipo de correspondencia es 1:N, es decir, a un departamento le pertenecen cero o más empleados (0,N). Un empleado pertenece a un departamento y sólo a uno (1,1).

- La relación «**JEFE**», que asocia la entidad **EMPLEADO** consigo misma. Su tipo de correspondencia es 1:N, es decir, un empleado es jefe de cero o más empleados (0,N). Un empleado tiene un jefe y sólo uno (1,1). Ver Figura 1.10:

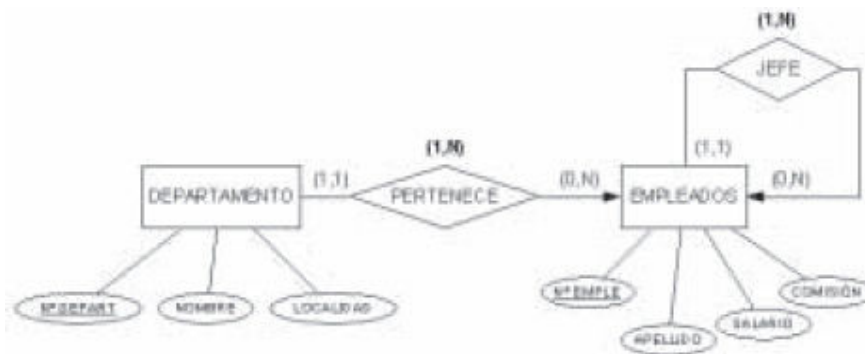


Figura 1.10. Diagrama E-R de las relaciones entre departamentos y empleados

3.-Modelo de red y cardinalidad

Este modelo utiliza estructuras de datos en red, también conocidas como estructuras plex. Las entidades se representan como registros o nodos, y las relaciones como enlaces o punteros.

En una estructura red cualquier componente puede vincularse con cualquier otro. Es posible describirla en términos de padres e hijos, pero, a diferencia del modelo jerárquico, un hijo puede tener varios padres.

Las representaciones lógicas basadas en árboles o en estructuras plex, a menudo, limitan el cambio que el crecimiento de la BD exige, hasta tal punto que las representaciones lógicas de los datos pueden variar afectando a los programas de aplicación que usan esos datos.

Los conceptos básicos de este modelo son los siguientes:

- **Elemento:** es un campo de datos. Ejemplo: DNI.

- **Agregados de datos:** conjunto de datos con nombre. Ejemplo: Fecha (día, mes, año).
- **Tipos de registro:** representa un nodo, un conjunto de campos. Cada campo contiene elementos. Es la unidad básica de acceso y manipulación. Se asemeja a los registros en archivos o las entidades en el modelo E-R.
- **Conjunto:** colección de dos o más tipos de registro que establece una vinculación entre ellos. Uno de ellos se llama propietario y el otro, miembro. Tienen una relación muchos a muchos (M:M), que para representarla se necesita un registro conector. Los conjuntos están formados por un solo registro propietario y uno o más registros miembros. Un registro propietario no puede ser a la vez miembro de sí mismo. Una ocurrencia del conjunto está formada por un registro propietario y el resto son registros miembros. Una ocurrencia de registro no puede pertenecer a varias ocurrencias del mismo conjunto.
- **Ciclo:** se forma cuando un registro miembro tiene como descendientes a uno de sus antepasados.
- **Bucle, lazo o loop:** es un ciclo en el que los registros propietarios y miembros son el mismo.

4.-Diagramas de estructura de datos en un modelo en red

El **diagrama** es el esquema que representa el diseño de una BD de red. Se basa en representaciones entre registros por medio de enlaces. Existen relaciones en las que participan solo dos entidades (binarias) y relaciones en las que participan más de dos entidades (generales) ya sea con o sin atributo descriptivo en la relación. Se utilizan cuadros o celdas para representar los registros y líneas para representar los enlaces.

Una ocurrencia del esquema son los valores que toman los elementos del esquema en un determinado momento. El modelo es muy flexible pues no hay restricciones. Esto implica la gran dificultad a la hora de implementarlo físicamente a la larga es poco eficiente. Para representar físicamente este modelo se pueden usar punteros y listas. Este modelo es sólo teórico, a la hora de llevarlo a la práctica se introducen las restricciones necesarias.

El diagrama de estructura de datos de red especifica la estructura lógica global de la BD; su representación gráfica se basa en la colocación de los campos de un registro en un conjunto de celdas que se ligan con otro(s) registro(s).

Vamos a suponer que tenemos una BD de red con datos de alumnos y asignaturas.

Registro ALUMNO:

```
struct ALUMNO
{ int NMatricula;
  char Nombre[35];
  int Curso;
  char Direccion[30];
};
```

Registro ASIGNATURA:

```
struct ASIGNATURA
{ int Codigo;
  char NomAsig[35];
};
```

Entre **ALUMNOS** y **ASIGNATURA** existe la relación de **ALUMNO-CURSA-ASIGNATURA**, que en el modelo **E-R** se representa como muestra la Figura 1.16:

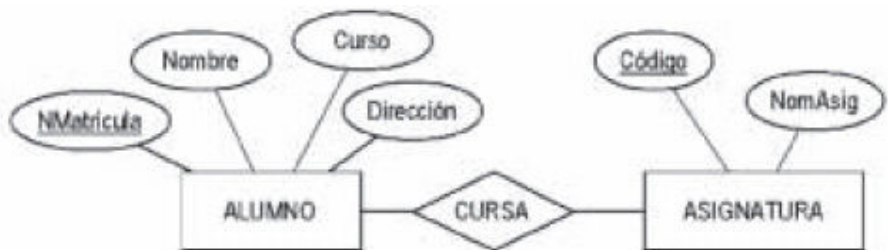


Figura 1.16 Modelo E-R de ALUMNO-CURSA-ASIGNATURA.

Las estructuras de datos según la cardinalidad se representan en los siguientes casos:

- **Representación del diagrama de estructura cuando el enlace o la relación no tiene atributos descriptivos**

- **Caso 1.** Cardinalidad uno a uno (ver Figura 1.17).

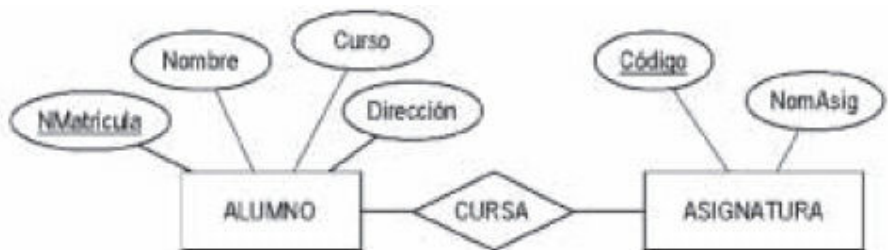


Figura 1.17. Relación uno a uno sin atributos, en el modelo en red.

- **Caso 2.** Cardinalidad uno a muchos, un alumno cursa muchas asignaturas (ver Figura 1.18).

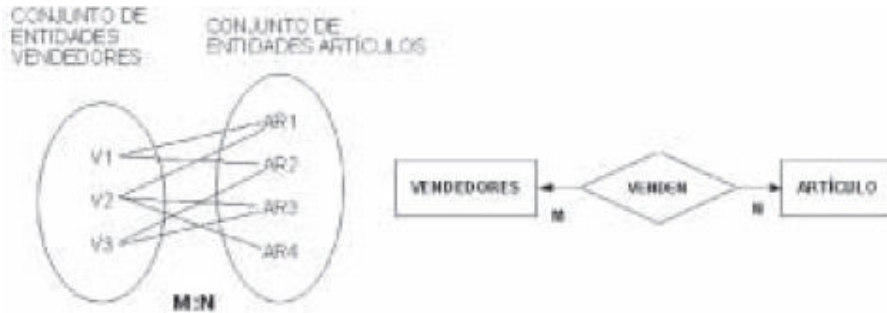


Figura 1.18. Relación uno a muchos, en el modelo en red.

- **Caso 3.** Cardinalidad muchos a muchos, muchos alumnos cursan muchas asignaturas (ver Figura 1.19).

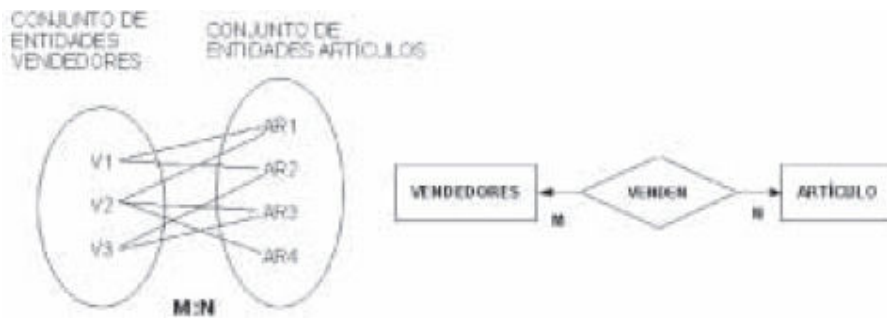


Figura 1.19. Relación muchos a muchos en el modelo en red.

En la Figura 1.20 se muestra un ejemplo de ocurrencia M:N:

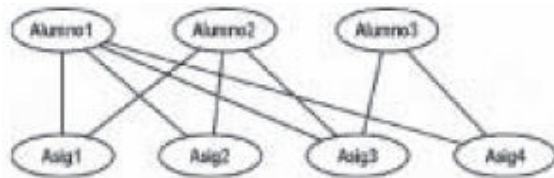


Figura 1.20. Ejemplo de ocurrencias M:N en el modelo en red.

- **Cuando el enlace tiene atributos descriptivos**

Suponemos ahora que en la relación entre ALUMNOS y ASIGNATURAS necesitamos saber la Nota de un alumno en una asignatura, es decir, la relación lleva un atributo descriptivo. El diagrama del modelo E-R se muestra en la Figura 1.21:



Figura 1.21. Modelo E-R de ALUMNO-CURSA-ASIGNATURA con el atributo NOTA.

Para convertirlo al diagrama de estructura de datos realizaremos lo siguiente:

1. Se representan los campos del registro como antes.
2. Se crea un nuevo registro, que llamaremos NOTAS, con el campo Nota.
3. Se crean los enlaces indicando la cardinalidad. A los enlaces les daremos los nombres: **ALUMNOTAS** para la relación entre **ALUMNO** y **NOTAS**, y **ASIGNOTAS**, para la relación entre **ASIGNATURAS** y **NOTAS**.

- **Dependiendo de la cardinalidad los diagramas de estructuras de datos se transforman como sigue**

- **Caso 1.** Cardinalidad uno a uno. Un alumno cursa una asignatura con una sola nota, en la Figura 1.22 se representa el diagrama.

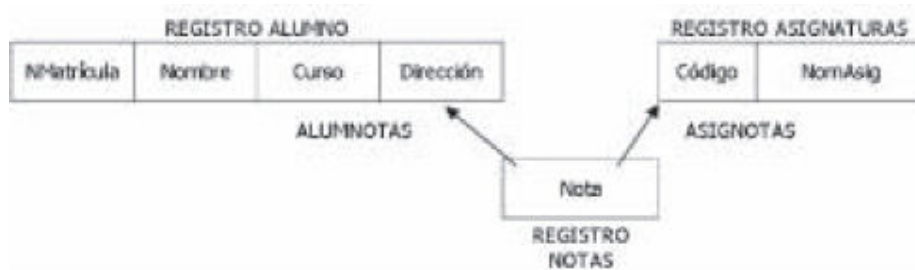


Figura 1.22. Representación de la cardinalidad 1:1 en el modelo en red.

- **Caso 2.** Cardinalidad uno a muchos. Un alumno cursa muchas asignaturas que tienen una nota (ver Figura 1.23).

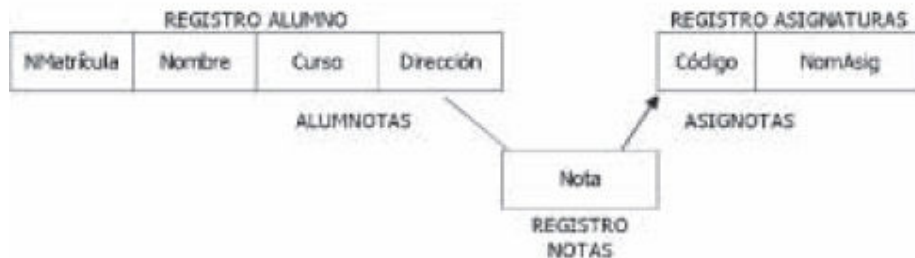


Figura 1.23. Representación de la cardinalidad 1:M en el modelo en red.

- **Caso 3.** Cardinalidad muchos a muchos. Muchos alumnos cursan muchas asignaturas con muchas notas (ver Figura 1.24).

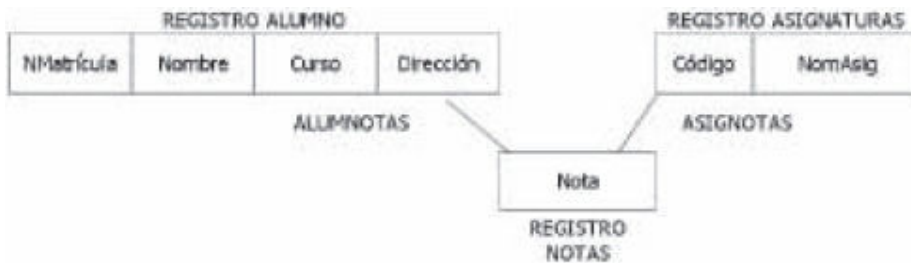


Figura 1.24. Representación de la cardinalidad N:M en el modelo en red.

- Consideramos ahora que en la relación intervienen más de dos entidades y no hay atributos descriptivos en la relación.

Suponemos que nos interesa saber los profesores que imparten las asignaturas y agregamos a la relación **ALUMNO-CURSA-ASIGNATURA** la entidad **PROFESOR**. El diagrama **E-R** queda como se muestra en la Figura 1.25:

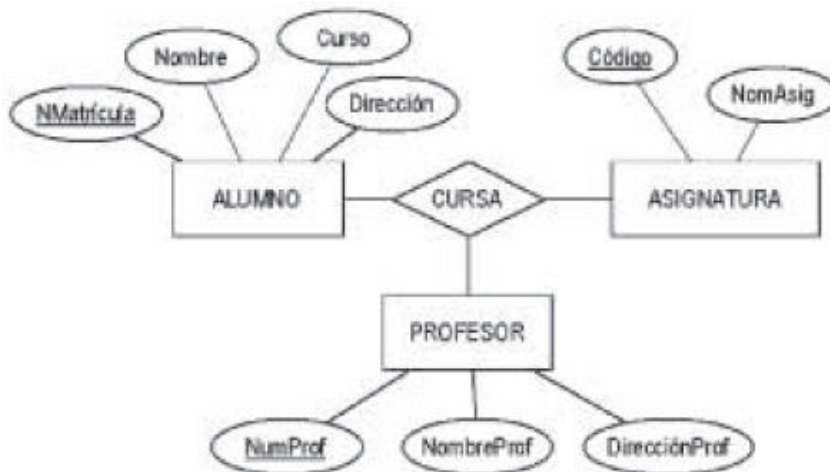


Figura 1.25. Representación en el modelo E-R de la relación entre ALUMNO-ASIGNATURA-PROFESOR.

Para realizar la transformación a diagramas de estructura de datos seguimos los siguientes pasos:

1. Se crean los registros para cada una de las entidades que intervienen.
2. Se crea un nuevo tipo de registro que llamaremos ENLACE, que puede no tener campos o tener sólo uno que contenga un identificador único, el identificador lo proporcionará el sistema y no lo utiliza directamente el programa de aplicación. A este registro se le denomina también como registro de enlace o conector.

- Considerando una relación con cardinalidad uno a uno, el diagrama de estructuras se muestra en la Figura 1.26:

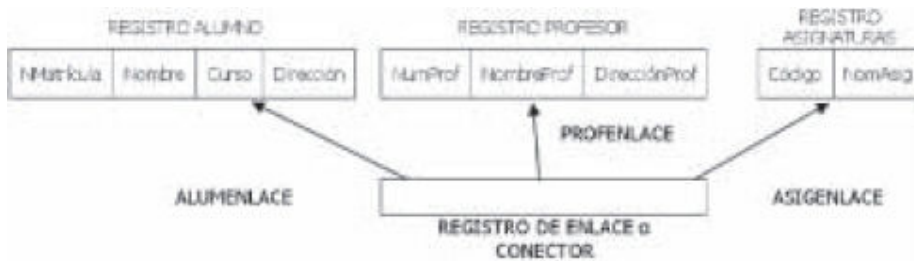


Figura 1.26. Diagrama de estructuras de la relación uno a uno entre 3 entidades.

Si el registro enlace tuviese atributos, estos se añadirán y se enlazarán indicando el tipo de cardinalidad de que se trate. Por ejemplo, añadimos el atributo Nota a la relación CURSA (ver Figura 1.27).

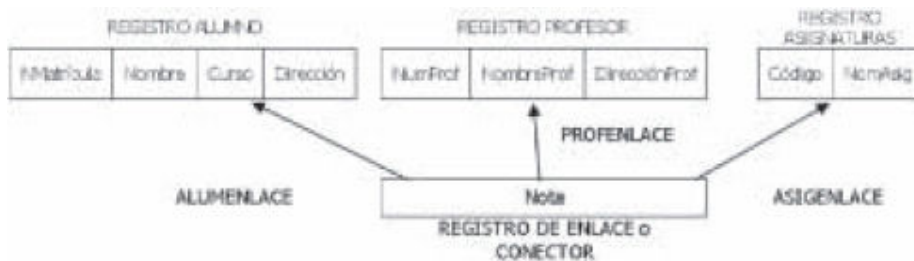


Figura 1.27. Diagrama de estructuras de la relación uno a uno con registro de enlace y atributos.

Si consideramos que un profesor imparte una sola asignatura y los alumnos tienen nota en una sola asignatura y esa asignatura sólo es cursada por un alumno, una instancia de este modelo sería la que se muestra en la Figura 1.28:



Figura 1.28. Representación de una ocurrencia de la relación PROFESOR-ALUMNO-ASIGNATURA.

El registro enlace contendrá los punteros para acceder a cada uno de los registros.

- Si ahora consideramos una cardinalidad muchos a muchos, es decir, un alumno cursa muchas asignaturas y el profesor imparte muchas asignaturas: una asignatura es impartida por muchos profesores y una asignatura es cursada por muchos alumnos, el diagrama de estructuras resultante se muestra en la Figura 1.29.

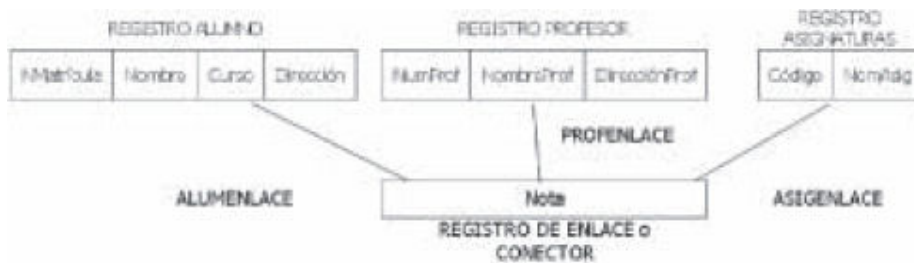


Figura 1.29. Diagrama de estructuras para una relación M:M.

Los **SGDB** que utilizan este modelo de red deben añadir restricciones a la hora de implementar físicamente la **BD** y obtener un mayor rendimiento del sistema. En 1971, un grupo conocido como **CODASYL** (Conference on Data System Languages) desarrolla el modelo **DBTG** (Data Base Task Group). Este grupo es el que desarrolló los estándares para **COBOL**. El modelo **CODASYL** ha evolucionado durante los últimos años y existen diversos productos **SGDB** orientados a transacciones, sin embargo actualmente estos productos se utilizan muy poco. El modelo es bastante complejo de implementar y los diseñadores y programadores deben de tener mucho cuidado al elaborar **BBDD** y aplicaciones **DBTG**. Además este modelo está enfocado al **COBOL**. Gran parte de las deficiencias detectadas son atribuye a que este modelo fue desarrollado antes de que se establecieran los conceptos esenciales de la tecnología de bases de datos. Ejemplos de **BBDD** en red son el DMS 1100 de **UNIVAC**, el **EDMS** de Xerox, el **PHOLAS** de Philips y el **DBOMP** de **IBM**.

En el modelo **DBTG** sólo pueden emplearse enlaces uno a uno y uno a muchos. En este modelo, existen dos elementos principales que son el propietario y el miembro, donde sólo puede existir un propietario y varios miembros, y cada miembro depende sólo de un propietario.

Conjuntos

DBTG

En este modelo sólo se utilizan enlaces uno a uno y uno a muchos. Podemos representar este modelo como se muestra en la Figura 1.30:



Figura 1.30. Enlaces 1:1 y 1:M en el modelo DBTG. Conjunto DBTG.

En el modelo **DBTG** esta estructura se denomina conjunto **DBTG** o **SET**. El nombre que se le asigna al conjunto suele ser el mismo que el de la relación que une a las entidades. Cada conjunto **DBTG** puede tener cualquier número de ocurrencias. Puesto que no se permiten enlaces del tipo muchos a muchos, cada ocurrencia del conjunto tiene exclusivamente un propietario y cero o más registros miembros. Además, ningún registro puede participar en más de una

ocurrencia del conjunto en ningún momento. Sin embargo, un registro miembro puede participar simultáneamente en varias ocurrencias de diferentes conjuntos. Añadir que un mismo registro no puede ser miembro y propietario a la vez, no está admitida la reflexividad.

En la Figura 1.31 se representan varias ocurrencias de un conjunto DBTG. La tercera ocurrencia no es válida pues todas pertenecen al mismo conjunto, y el miembro 3 no puede tener dos propietarios (ver Figura 1.31).

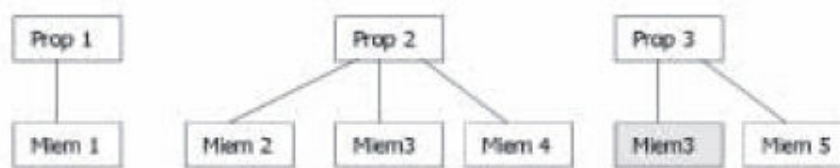


Figura 1.31. Ocurrencias de un conjunto DBTG.

Para declarar los conjuntos de un diagrama de estructuras como el que se muestra en la Figura 1.32 realizamos lo siguiente:

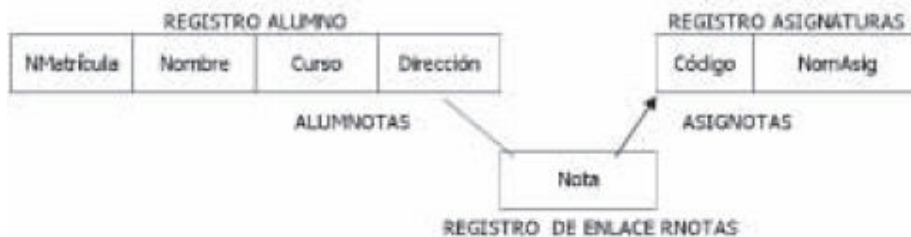


Figura 1.32. Ejemplo diagrama de estructuras. Relación 1:M.

En el diagrama existen dos conjuntos DBTG:

ALUMNOTAS, cuyo propietario es ALUMNO y cuyo miembro es RNOTAS.
ASIGNOTAS, cuyo PROPIETARIO es ASIGNATURAS y miembro RNOTAS.

La declaración de los conjuntos será:



Al igual que otros modelos de datos, el DBTG proporciona un lenguaje de comandos para la manipulación de datos, así como para la actualización y el procesamiento de conjuntos DBTG.

NOTA: Con este capítulo hemos llegado al final del curso. Recuerda que este trabajo es un fragmento de los alumnos del conalep 262 del grupo 4103 del equipo 4 integrado por los alumnos: PABLO REYLANDER ORTIZ DIAZ, MERCEDES RIVERA PINTO, ISABELA GUTIERREZ PEREZ, MARICELA OSORIO MALDONADO, YADIRA DEL CARMEN GUTIERREZ MENDEZ.

