Bases de Datos

Parte 1: Conceptos Básicos y Modelo Entidad-Relación

Objetivo General

Dar a conocer al estudiante los conceptos básicos para diseñar e implementar bases de datos.

Prof. Fray León Osorio Rivera

Introducción

Es muy común que un sistema de información comience con la necesidad de guardar algunos datos de una empresa en determinado número de archivos y la generación de ciertas aplicaciones para manipularlos en respuesta a necesidades eventuales de la empresa. Lo más probable es que en dicho sistema con el correr del tiempo vayan interviniendo varios programadores con características de programación disímiles al igual que lo serán las herramientas de software y los formatos de archivo utilizados. A un sistema como éste, se le conoce en la jerga informática como **Sistema de Procesamiento de Archivos**, los cuales adolecen de lo siguiente:

- **Redundancia e inconsistencia de información** que genera alto consumo de medios de almacenamiento y de tiempo de acceso.
- **Dificultad en el acceso de la información** frente consultas fuera de las comunes y como resultado de la información repartida en distintos tipos de formatos.
- **Problemas de seguridad** al permitir el acceso a la información de todo tipo de usuario.
- **Problemas de integridad** al dificultarse la labor de satisfacer la consistencia de ciertas informaciones (sus limitantes).

Estos problemas son los que han fomentado el desarrollo de los sistemas de manejo bases de datos.

Modelos

Cuando los investigadores tratan de comprender un conjunto particular de fenómenos, con frecuencia hacen uso de un **modelo**. Un modelo, desde el punto de vista científico, es una especie de analogía o imagen mental de los fenómenos en términos de algo que les es familiar. El movimiento ondulatorio de la luz es un ejemplo. No podemos ver las ondas luminosas como podemos ver las ondas en el agua; pero es útil pensar que la luz está formada por ondas, porque los experimentos de óptica indican que en muchos aspectos la luz se comporta igual que las ondas del agua.

El objeto de un modelo es ofrecernos un cuadro mental o visual, algo que podamos captar, cuando no podemos ver qué esta sucediendo en realidad. Con frecuencia los modelos nos proporcionan una comprensión más profunda: la analogía con un sistema conocido (por ejemplo, de las ondas de agua en el párrafo anterior) nos puede sugerir nuevos experimentos para llevar a cabo, y nos puede dar ideas sobre otros fenómenos relacionados que podrían suceder.

Un modelo nunca es perfecto, y los investigadores tratan constantemente de afinar sus modelos o de establecer nuevos cuando los anteriores ya no son adecuados. El modelo atómico de la materia ha pasado por muchos refinamientos. Se imaginaba a los átomos como esferas diminutas con ganchos (para explicar los enlaces químicos), o bien como pequeñas bolas de billar rebotando continuamente entre sí. Más recientemente, el "modelo planetario" consideraba el átomo como un núcleo con electrones girando a su alrededor, igual que los planetas giran alrededor del sol. Sin embargo, este modelo resultó demasiado simplificado y falla en las pruebas definitivas.

Preguntará el lector cuál es la diferencia entre una teoría y un modelo. Algunas veces se usan esas palabras indistintamente. Sin embargo, en general, un modelo es bastante sencillo y proporciona una semejanza estructural con el fenómeno que estudia, mientras que una teoría es más amplia, más detallada, y trata de resolver un conjunto de problemas, frecuentemente con exactitud matemática. También con frecuencia, a medida que se desarrolla y modifica un modelo y corresponde más estrechamente a los experimentos en una amplia variedad de fenómenos, puede llegar a llamarse teoría. La teoría atómica es un ejemplo, al igual que la teoría ondulatoria de la luz.

Los modelos pueden ser muy útiles y con frecuencia conducen a teorías relevantes. Pero es importante no confundir un modelo, o una teoría, con el sistema real de los fenómenos mismos.

CODD Y LAS BASES DE DATOS RELACIONES

El modelo de las bases de datos relacionales ha dado vueltas desde 1970, cuando Edgar Codd, entonces un investigador de la IBM, lo presentó en sociedad en el clásico paper de comunications de la ACM titulado "*Un modelo relacional para enormes bancos de datos compartidos"*.

Actualmente, multitud de base de datos se distribuyen con el apellido de *relacionales* y de gran prestigio (DB2, por ejemplo. Inicialmente, la mayoría de las bases de datos que se ofrecían como relacionales dejaban mucho que desear, y en el año 1985 en un artículo publicado por la revista *Computerworld*, Codd daba 12 reglas que debían cumplir cualquier base de datos que desee considerarse relacional:

LOS MANDAMIENTOS DE CODD

1. Regla de información:
   Todos los datos de una base de datos relacional se representan explícitamente (al nivel lógico) como valores de tablas.

2. Reglas de acceso garantizado:
   Todos y cada uno de los datos (valor indisoluble, único o atómico) de una base de datos relacional se garantiza que sean lógicamente accesibles recurriendo a una combinación de nombres de tabla, valor de clave primaria y nombre de columna.

3. Tratamiento sistemático de valores Nulos:
   Los valores nulos (distinto de cadena de caracteres vacías o de una cadena de caracteres en blanco y distinta de cero o de cualquier otro número) se soporta en los SGBD completamente relacionales para representar la falta de información y la información inaplicable de un modo sistemático e independiente del tipo de datos.

4. Catalogo en línea dinámico basado en el modelo relacional:

La descripción de la base de datos o Catalogo se representa (o almacena) a nivel lógico como valores en tablas, del mismo modo que los datos ordinarios, de modo que los usuarios autorizados pueden aplicar a los datos regulares.

5. Regla de sublenguaje completo de datos:
Un sistema relacional puede soportar varios lenguajes de manipulación de datos (DML) y varios modos de uso terminal (por ejemplo, el modo de rellenar con blancos), sin embargo debe haber al menos un lenguaje cuyas sentencias sean expresables, mediante alguna sintaxis bien definida, como cadenas de caracteres, y que sea completa en cuanto al soporte de todos los puntos siguientes:

Definición de datos y definición de vistas
Manipulación de datos (interactiva y por programa)
Restricciones de integridad
Autorización
Fronteras de transacciones (comienzo, complementación y vuelta atrás)

6. Regla de actualización de vistas:
Todas las vistas que puedan utilizarse son también actualizables por el sistema.

7. Inserción, Modificación y Borrado de alto nivel:
Un SGBDR debe hacer más que recuperar conjuntos relacionales de datos, también debe ser capaz de insertar, actualizar y eliminar datos como un conjunto relacional.

8. Independencia física de los datos:
Los datos deben ser físicamente independiente de los programas de aplicación. Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cualquiera que sean los cambios efectuados ya sea las representaciones de almacenamiento o a los métodos de acceso.

9. Independencia lógica de los datos:
Cada vez que sea posible, el Software de aplicación debe ser independiente de los cambios hechos a las tablas Base. Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cuando se efectúan sobre las tablas de base de cambios preservadores de la información de cualquier tipo que teóricamente permita alteraciones.

10. Independencia de integridad:
La integridad de los datos debe ser definible en el lenguaje relacional y almacenarse en el Catálogo. Las restricciones de integridad específicas para una base de datos relacional y almacenables en el catálogo, no en los programas de aplicación.

11. Regla de no subversión:
Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro cada vez), ese bajo nivel no puede ser utilizado para subvertir o suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez)

12. Un SGBDR tiene independencia distributiva.

TEMA 1 Conceptos básicos de las bases de datos

Un **Sistema de Manejo de Bases de Datos** (en inglés **DBMS**, *DataBase Manegement System)* es un conjunto de archivos interrelacionados y una serie de programas que permiten a varios usuarios tener acceso a estos archivos ya sea para consultarlos o actualizarlos.

Concepto de Abstracción

Entre los objetivos de un DBMS está el de proporcionar a los usuarios una visión **abstracta** de la información, lo cual quiere decir que el sistema oculta ciertos detalles relativos a la forma como los datos se almacenan. Esto se debe a la necesidad de diseñar estructuras complejas de datos como consecuencia de la búsqueda de la eficiencia en el almacenamiento y el acceso de la información. La abstracción se da en tres niveles:

Nivel físico. El cual compete a la manera como realmente se almacenan los datos en los medios de almacenamiento.

Nivel conceptual. En el que se describen los datos que realmente se almacenan y las relaciones que existen entre ellos.

Nivel de Visión. Describe sólo una parte de la base de datos la cual se puede hacer de diferentes maneras para una misma base de datos.

Ejemplos:

Para un sistema de calificaciones, las siguientes descripciones corresponderían al *nivel conceptual*:

```
CREATE TABLE Alumno(
        [Id Alumno] int          IDENTITY(1,1)    PRIMARY KEY CLUSTERED,
        Carnet varchar(8)        NOT NULL,
        Apellidos varchar(50)    NOT NULL,
        Nombres varchar(50)      NOT NULL,
        [Id Programa] int        NOT NULL,
        Nivel int                NOT NULL
        )
CREATE TABLE Asignatura(
        [Id Asignatura] int      IDENTITY(1,1)    PRIMARY KEY CLUSTERED,
        Codigo varchar(10)       NOT NULL,
        Asignatura varchar(100)  NOT NULL,
        Creditos int             NOT NULL
        )
CREATE TABLE Programa(
        [Id Programa] int        IDENTITY(1,1)    PRIMARY KEY CLUSTERED,
        Programa varchar(100)    NOT NULL,
        )
CREATE TABLE Nota(
        [Id Alumno] int          NOT NULL,
        [Id Asignatura] int      NOT NULL,
        [Id Periodo int          NOT NULL,
        Nota real                NULL
        )
CREATE TABLE Periodo(
        [Id Periodo] int  IDENTITY(1,1)    PRIMARY KEY CLUSTERED,
        Periodo varchar(20)      NOT NULL
        )
```

En el *nivel físico*, un registro de *Alumno* estaría almacenado como un bloque de bytes consecutivos (igual pueden estarlo los demás).

En el *nivel de visión*, una opción que permita consultar las asignaturas que tiene matriculadas un alumno en un período determinado, implicaría extraer los siguientes datos:

Del registro *Alumno*: Apellidos, Nombres
Del registro *Asignatura:* Codigo, Asignatura

lo cual estaría regido por ciertas condiciones y las relaciones que haya entre las diferentes estructuras y sería consultado así:

```
Select Alumno.Apellidos, Alumno.Nombres, Asignatura.Codigo, Asignatura.Asignatura, Asignatura.Codigo
    From Alumno, Asignatura, Nota, Periodo
    Where Alumno.[Id Alumno] = Nota.[Id Alumno]
    And Nota.[Id Asignatura] = Asignatura.[Id Asignatura]
    And Nota.[Id Periodo] = Periodo.[Id Periodo]
    And Alumno.Carnet = '8711608'
    And Periodo.Periodo = '1990-1'
```

Modelos de Datos

Son un grupo de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes.

Para los niveles conceptual y de visión, surge una clasificación de estos modelos, así:

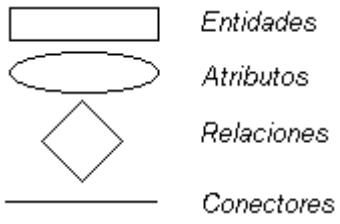
1. Modelos Lógicos Basados en Objetos.

Permiten una estructuración bastante flexible y hacen posible especificar claramente las limitantes de los datos. El más representativo es el **Modelo Entidad-Relación** (E-R) el cuál se basa en la percepción de un mundo real consistente en un conjunto de objetos básicos denominados **Entidades** y de las **Relaciones** que se dan entre estos. Una **Entidad** es un objeto que existe y puede distinguirse de otros, dado que cada uno posee ciertos **Atributos** que le identifican. Una **Relación** es una asociación entre varias entidades. Por ejemplo:

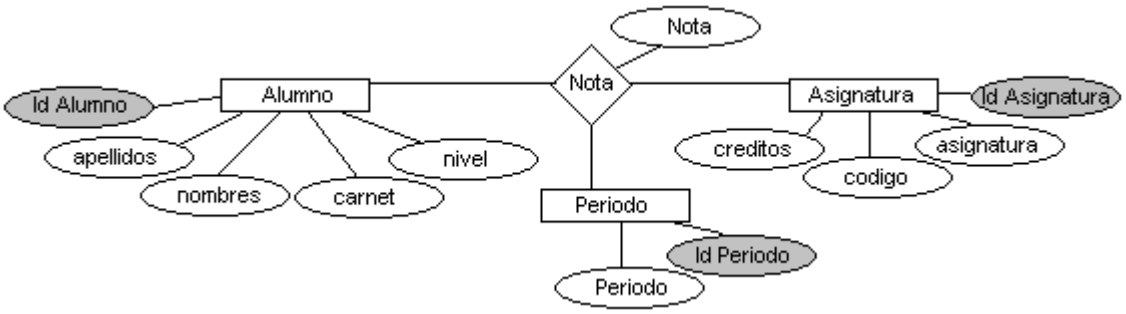
Los atributos **carnet, apellidos, nombres, nivel, Id alumno** describen al objeto **alumno**.

La relación **Nota** asocia a un alumno con cada una de las asignaturas que matriculó en el semestre.

La estructura lógica general de una base de datos se puede expresar gráficamente por medio de un **Diagrama E-R** el cual tiene los siguientes componentes:



Para el ejemplo anterior, parte del *Diagrama E-R* sería:

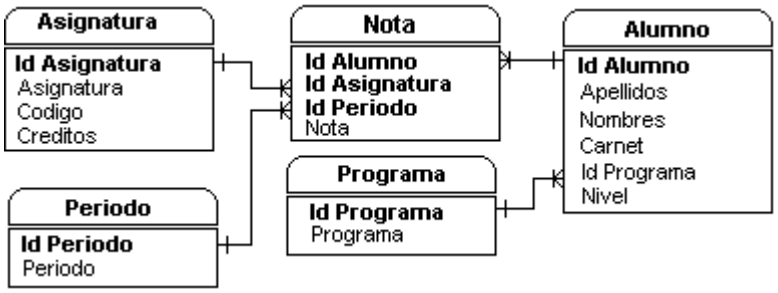


2. Modelos Lógicos Basados en Registros.

Estos, además de especificar la estructura lógica general de la base de datos, da una descripción en un nivel más alto de implementación, pero no permite especificar de un modo claro las limitantes de los datos.

Un modelo representativo es el **Modelo Relacional**, en el cual los datos y las relaciones entre los datos se representan por medio de una serie de **Tablas** cada una de las cuales tiene varias **Columna** con nombres únicos.

Para el ejemplo anterior, el *modelo relacional* sería:



Instancias y Esquemas

El conjunto de información guardado en una base de datos en un momento dado se denomina **Instancia** en la base de datos.

El diseño general de la base de datos se denomina **Esquema** de la base de datos.

Un *esquema* raras veces se altera, mientras la *instancia* se caracteriza por su inestabilidad.

En una base de datos existen esquemas para cada nivel de abstracción. De este modo, se tienen *esquema físico, esquema conceptual* y *subesquemas*, respectivamente.

En el ejemplo anterior, un esquema es el modelo relacional, mientras que una instancia podría ser:

Alumno

Id Alumno	Carnet	Apellidos	Nombres	Id Programa	Nivel
1	8620639	GARCIA LORCA	FEDERICO	1	02
2	8711608	ARCINIEGAS	GERMAN	1	03
3	8820890	ASUNCION SILVA	JOSE	2	02

Asignatura

Id Asignatura	Codigo	Asignatura	Creditos
1	MAT150	ALGEBRA LINEAL	4
2	SIS020	DIAGRAMACION	6
3	SIS021	LENGUAJES DE PROGRAMACION	4

Programa

Id Programa	Programa
1	INGENIERÍA CIVIL
2	CONTADURÍA

Independencia de los datos

Es la capacidad de modificar un esquema de un nivel sin modificar el esquema del nivel inmediatamente superior. De hecho existen dos *niveles de independencia* de los datos:

- **Independencia Física:** Capacidad de modificar el esquema físico sin obligar a la reedición de los programas de aplicaciones.
- **Independencia Lógica:** Capacidad de modificar el esquema conceptual sin obligar a la reedición de los programas de aplicaciones.

La *independencia lógica de los datos* es más difícil de lograr que la *independencia física*, puesto que los programas de aplicaciones dependen en alto grado de la estructura lógica de los datos.

Lenguaje de Definición de Datos

Un **Lenguaje de Definición de Datos** (en inglés **DDL**, *Data Definition Language*) es un lenguaje especial que permite la especificación de un esquema de una base de datos por medio de una serie de definiciones.

El resultado de la compilación de las proposiciones en DDL es un conjunto de tablas que se almacenan en un archivo especial conocido como **Diccionario de Datos**.

Lenguaje de Manejo de Datos

Un **Lenguaje de Manejo de Datos** (en inglés **DML**, *Data Manipilation Language*) permite a los usuarios manejar o tener acceso a los datos que estén organizados por medio del modelo apropiado. Existen dos tipos de DML:

- **Procedimentales**, los cuales requieren que el usuario especifique *Qué* datos quiere y *Cómo* los quiere.
- **No procedimentales**, que requieren que el usuario especifique *Qué* datos quiere sin especificar cómo obtenerlos.

Manejador de Base de Datos

Un **Manejador de Base de Datos** es un módulo de programa que constituye la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicaciones y las consultas hechas al sistema. Sus tareas básicas son :

- **Interacción con el manejador de archivos**. Es quién se encarga realmente del almacenamiento, recuperación y actualización de los datos en la base de datos.
- **Implantación de la integridad**. Realiza las operaciones que limitan la consistencia de los datos.
- **Puesta en práctica de la seguridad**. Hace cumplir los requisitos de seguridad para tener acceso a los datos.

- **Respaldo y recuperación**. Es responsable de detectar fallas y restaurar la base de datos al estado que existía antes presentarse ésta.
- **Control de Concurrencia**. Vela por la consistencia de la información cuando varios usuarios actualizan a la vez la base de datos.

Administrador de Base de Datos

La persona que ejerce un control centralizado sobre un DBMS se le conoce como el **Administrador de Base de Datos** (en inglés **DBA**, *Data Base Administrator*). Sus funciones principales son:

- **Definición de Esquema**. Es el creador del esquema original de la base de datos en un DDL.
- **Modificación del esquema y de la organización física.**
- **Administración de la seguridad**. Concede autorizaciones para el acceso a los datos a los distintos usuarios de la base de datos.
- **Especificación de las limitantes de Consistencia**.

Usuarios de la Base de Datos

Existen tres tipos de usuarios de un DBMS, los cuales se distinguen por el modo como interactúan con el sistema:

- **Programadores de Aplicaciones :** Son profesionales en computación que interactúan con el sistema mediante instrucciones en DML, las cuales son traducidas en un programa escrito en un **lenguaje huésped** (Cobol, Pascal, C, Visual Basic, Java). Estos programas se denominan **Programas de Aplicaciones**.
- **Usuarios Casuales** : Usuarios que interactúan con el sistema sin escribir programas, pero haciendo consultas en un **Lenguaje de Consulta de Datos**.
- **Usuarios Ingenuos :** Usuarios que interactúan con el sistema llamando alguno de los programas de aplicaciones permanentes escritos previamente por los *programadores de Aplicaciones*.

Componentes de un Sistema de Base de Datos

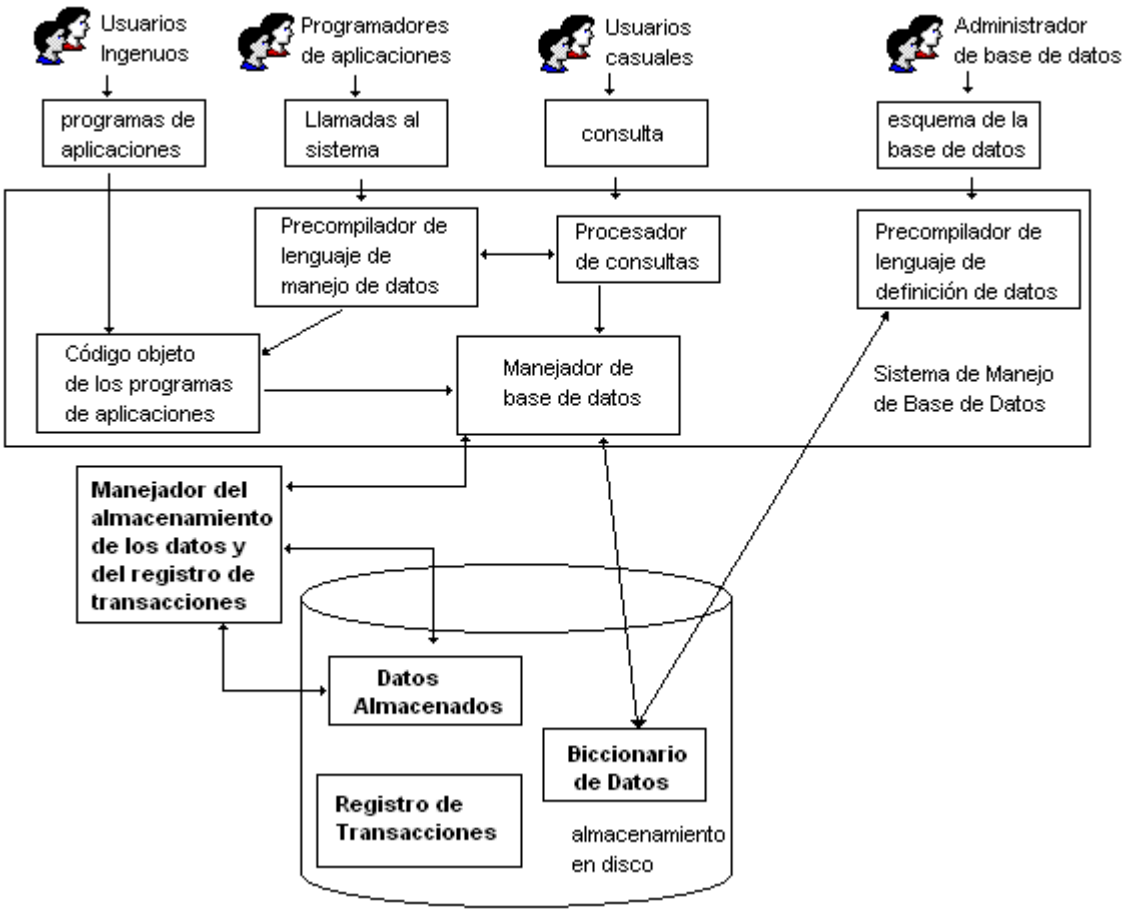
Un sistema de base de datos se divide en una serie de módulos que se encargan de cada una de las tareas del sistema general.

Básicamente, un sistema de base de datos consiste en varios componentes funcionales, entre los que se cuentan:

- **El manejador de archivos :** encargado de asignar espacio en el disco y de las estructuras de datos que se van a emplear para representar la información almacenada en disco.
- **El manejador de base de datos :** que constituye la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicaciones y las consultas que se hacen al sistema.
- **El procesador de consultas :** que traduce las proposiciones en lenguaje de consulta a instrucciones de bajo nivel que pueda entender el manejador de la base de datos.
- **El precompilador de DML:** que convierte las proposiciones hechas en lenguaje DML en procedimientos del *lenguaje huésped*.
- **El compilador de DDL :** que convierte las proposiciones DDL en un conjunto de tablas que contienen metadatos. Estas quedan almacenadas en el *Diccionario de Datos*.

Se tienen además, unas estructuras de datos como parte de la implementación física del sistema:

- **Archivos de datos:** que almacenan la base de datos.
- **Diccionario de datos :** que almacena información respecto a la estructura de la base de datos.
- **Indices :** que permiten un acceso ordenado a la información.



TEMA 2 | Modelo Entidad-Relación

El modelo de datos **entidad-relación** (E-R) es la percepción de un mundo real que consiste en un conjunto de objetos básicos llamados **entidades** y de unas **relaciones** entre estos objetos. Se le utiliza para esquematizar la estructura lógica general de la base de datos.

Entidades

Una **entidad** es un objeto que existe y puede distinguirse de otros objetos. Por ejemplo, *Ana María* con carnet *8620639* es una entidad, ya que identifica a una persona en una institución educativa.

Las entidades pueden ser de dos tipos: **Concretas**, cuando representan algo tangible (personas, equipos, libros) o **Abstractas**, cuando se utilizan para cosas intangibles (conceptos de pago, asignaturas).

Un **Conjunto de entidades** es una agrupación de entidades del mismo tipo. Un ejemplo sería el conjunto de estudiantes de una universidad el cual se denominaría *alumno*.

Una entidad está representada mediante un conjunto de **atributos**. Para el conjunto de entidades *alumno*, por ejemplo, los atributos podrían ser *id alumno, carnet*, *nombres, apellidos*, y *nivel*.

Para cada atributo existe un rango de valores permitidos conocido como **Dominio** del atributo. Así, por ejemplo, el dominio del atributo *carnet* son todas las cadenas de caracteres de 7 cifras.

Un atributo es una función que mapea un conjunto de entidades dentro de un dominio, para lo cual cada entidad se describe por medio de un conjunto de parejas **(atributo, valor del dato)**, una por cada atributo del conjunto de entidades. Por ejemplo: Una entidad *alumno* se describiría:
{(***id alumno***, *23*), (***nombre***, *Ana María*), (***apellidos***, *Jimenez Suárez*), (***carnet***, *8620639*), (***nivel***, *1*)}

Una base de datos es, por consiguiente, una agrupación de conjunto de entidades, cada uno de los cuales contiene cualquier número de entidades del mismo tipo.

En el ejemplo que se manejará en este documento, se tendrán los siguientes conjuntos de entidades para una base de datos de calificaciones de una universidad.

Entidad	Descripción	Atributos
alumno	El conjunto de los estudiantes de una universidad determinada	***id alumno*** ***carnet*** ***apellidos*** ***nombres*** ***nivel***
asignatura	El conjunto de las asignaturas que se dictan en la universidad	***id asignatura*** ***código*** ***asignatura*** ***créditos***
programa	El conjunto de los programas académicos que se dictan en la universidad	***id programa*** ***programa***
periodo	El conjunto de los períodos académicos que han transcurrido en la universidad	***id programa*** ***periodo***

En la siguiente figura se pueden observar los conjuntos de entidades ***alumno*** y ***programa***.

Alumno

1	GUTIERREZ LOPEZ	ADRIANA MARIA	9019451	2
2	MENESES AGUIRRE	OLGA LUCIA	9020678	3
3	TOBON VELASQUEZ	ELKIN DARIO	9021023	3
4	MUNOZ PEREZ	JORGE ELIECER	9110345	4
5	DIAZ HINESTROSA	ANA MARIA	8620639	8
6	IDARRAGA GÓMEZ	JAIME ELADIO	8711608	7

Programa

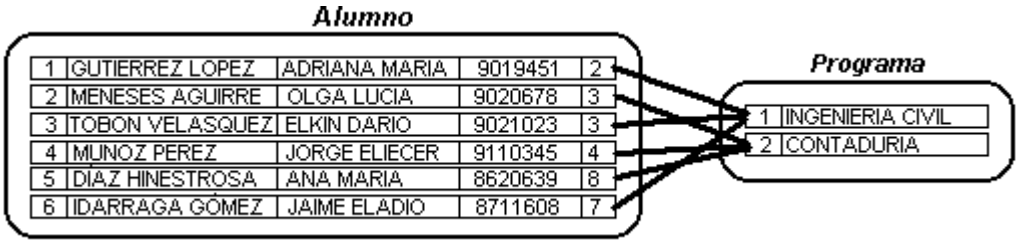
1	INGENIERIA CIVIL
2	CONTADURIA

Entidades

Una **relación** es una asociación entre varias entidades. Por ejemplo, se puede definir una relación que asocie al *alumno* "ANA MARIA MUÑOZ" con el *programa* "INGENIERA CIVIL", lo cual indicaría que el alumno está matriculado en ese programa. Llamaríamos al conjunto de tales relaciones *Alumno Programa.*

Un **conjunto de relaciones** es un grupo de relaciones del mismo tipo.

Para el caso de la figura anterior se ilustraría la relación *Alumno Programa* de la siguiente manera:



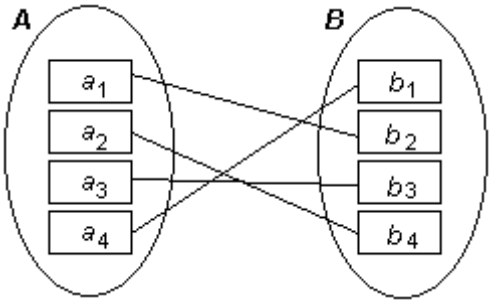
La relación ***Alumno Programa*** es un ejemplo de relación binaria, lo cual quiere decir que implica a dos entidades. También pueden existir relaciones que implican a más de dos entidades.

Cardinalidades de Mapeo

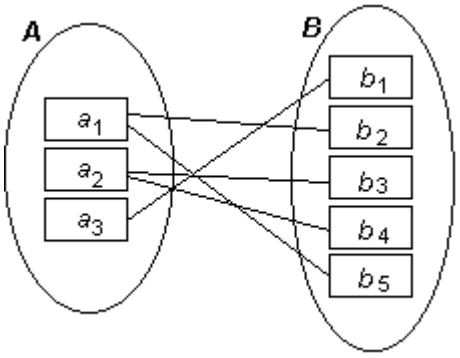
Limitante de la base de datos que expresa el número de entidades con las que puede asociarse otra entidad mediante una relación.

Para un conjunto binario de relaciones ***R*** entre los conjuntos de entidades ***A*** y ***B***, la **cardinalidad de mapeo** debe ser una de las siguientes:

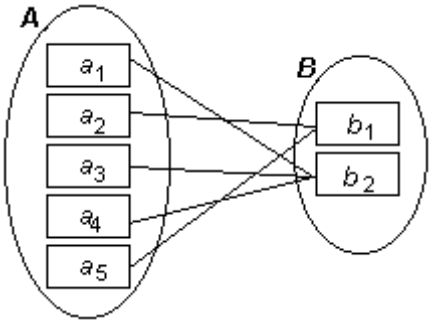
Una a una. Una entidad de ***A*** sólo puede estar asociada con una entidad de ***B***, y una entidad de ***B*** sólo puede estar asociada con una entidad de ***A***.



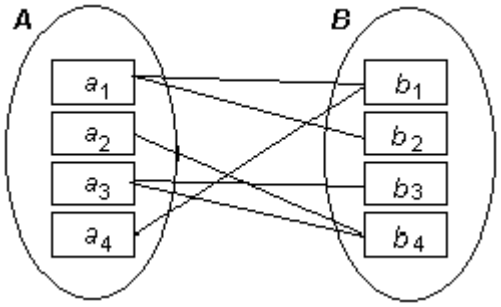
Una a muchas. Una entidad de ***A*** está relacionada con cualquier número de entidades de ***B***, pero una entidad de ***B*** sólo puede estar asociada con una entidad de ***A***.



Muchas a una. Una entidad de ***A*** está relacionada sólo con una entidad de ***B***, pero una entidad de ***B*** puede estar asociada con cualquier número de entidades de ***A***.



Muchas a muchas. Una entidad de ***A*** está relacionada con cualquier número de entidades de ***B***, y una entidad de ***B*** puede estar asociada con cualquier número de entidades de ***A***



En el ejemplo, la relación *Alumno Programa* es de cardinalidad *muchas a una* puesto que un alumno puede tomar sólo un programa, y a su vez, un programa puede ser tomada por varios alumnos.

Dependencia de Existencia

La **dependencia de existencia** constituye otra clase importante de limitante. Esta implica que la existencia de una entidad depende de la existencia de otra. Si la existencia de la entidad ***a*** depende de la existencia de entidad ***b***, se dice entonces que ***a es dependiente por existencia de b***. Esto implicaría a su vez que si se elimina ***b***, también se eliminará ***a***. En este orden de eventos, ***a*** es una **entidad subordinada** y ***b*** es una **entidad dominante**.

En nuestro ejemplo la existencia de la entidad *Alumno* depende de que exista la entidad *Programa.*

Llaves primarias

Desde el punto de vista de las bases de datos, las entidades y relaciones deben diferenciarse unas de otras en términos de sus atributos. Para hacer tales distinciones, se asigna una **Superllave** a cada conjunto de entidades.

La **Superllave** es un conjunto de uno o más atributos, lo cuales, juntos, permiten identificar en forma única a una entidad dentro del conjunto de entidades. La entidad *Alumno*, puede ser identificada por el atributo *carnet*, por lo tanto esta es una de su superllaves. El atributo *nombres* no es suficiente para identificar a un sólo estudiante pero si se combina con el atributo *carnet*, también lo identificaría. De esta manera cualquier combinación del atributo *carnet* con otro de los atributos, serviría para indentificar a una entidad. Por esta razón, el concepto superllave no es suficiente para el modelado de una base de datos. Lo que se busca entonces es la superllave más pequeña posible. Esta se conoce como **Llave Candidato**. Es posible que existan varios conjuntos de atributos que puedan servir como llaves candidato.

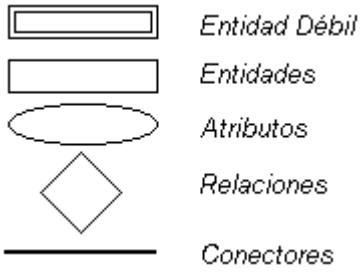
Una **Llave primaria** es la llave candidato que el diseñador de la base de datos elija para identificar una entidad dentro de un conjunto de identidades.

Es posible que una entidad no cuente con los suficientes atributos para definir una llave primaria. Se les conoce como **entidades débiles**. En consecuencia, a las entidades con llave primaria se les denomina **entidades fuertes**. Generalmente una entidad fuerte es dominante, mientras las débiles son subordinadas.

Los conjuntos de relaciones también tienen sus llaves primarias. Estas se forman tomando los atributos que constituyen las llaves primarias de los conjuntos de entidades que definieron el conjunto de relaciones.

Diagrama Entidad-Relación (E-R)

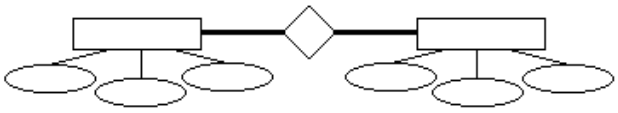
La estructura lógica general de una base de datos se puede expresar gráficamente por medio de un **Diagrama E-R** el cual tiene los siguientes componentes:



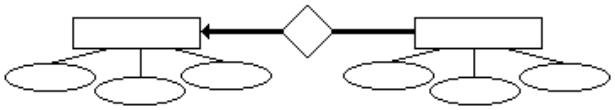
Cada componente se etiqueta con su nombre correspondiente.

Para distinguir la cardinalidad en los diagramas, se añada a los conectores una flecha según sea esta:

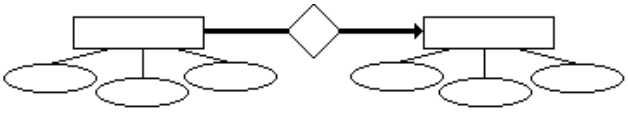
Muchas a Muchas



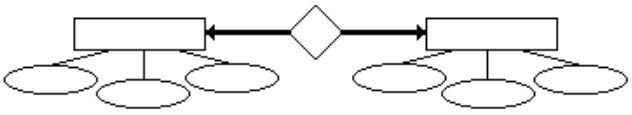
Una a Muchas



Muchas a Una

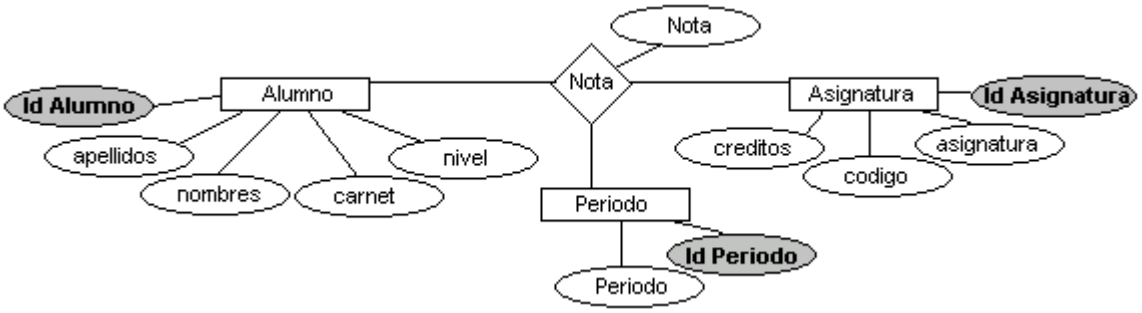


Una a Una

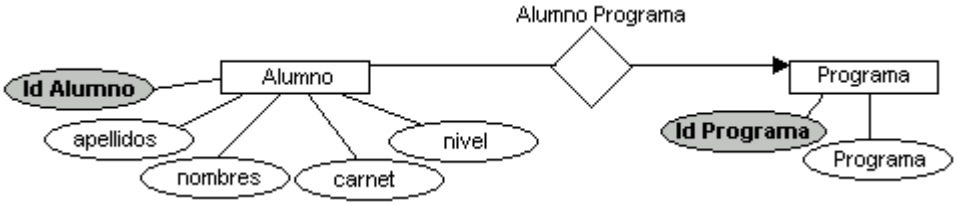


Para el ejemplo anterior podríamos tener los siguientes diagramas E-R:

Relación *Nota*: Asignaturas que un estudiante toma durante un período.



Relación *Alumno Programa*: Programa académico que cursa el estudiante.



Reducción a tablas

Una base de datos que se ajuste a un diagrama E-R puede representarse por medio de un conjunto de tablas. Para conjuntos de entidades y de relaciones en la base de datos, existe una tabla única que recibe el nombre del conjunto de entidades o de relaciones correspondiente. Cada tabla tiene un número de columnas las cuales también tienen nombres únicos. Estas columnas corresponden a los atributos. Las tablas a su

vez, presentan renglones con los valores de los atributos para cada una de las entidades. Veamos los ejemplos:

Tablas correspondientes a las entidades:

Alumno

Id Alumno	Apellidos	Nombres	Carnet	Nivel
1	GUTIERREZ LOPEZ	ADRIANA MARIA	9019451	2
2	MENESES AGUIRRE	OLGA LUCIA	9020678	3
3	TOBON VELASQUEZ	ELKIN DARIO	9021023	3
4	MUNOZ PEREZ	JORGE ELIECER	9110345	4
5	DÍAZ HINESTROSA	ANA MARIA	8620639	8
6	IDARRAGA GOMEZ	JAIME ELADIO	8711608	7

Programa

Id Programa	Programa
1	INGENIERIA CIVIL
2	CONTADURIA

Asignatura

Id Asignatura	Codigo	Asignatura	Creditos
1	MAT051	MATEMATICAS OPERATIVAS	4
2	MAT020	MATEMATICAS DISCRETAS	4
3	MAT021	GEOMETRIA VECTORIAL	4
4	MAT025	CALCULO I	6
5	MAT026	CALCULO II	4

Periodo

Id Periodo	Periodo
1	1991 - 1
2	1991 - 2
3	1992 - 1

Tablas correspondientes a las relaciones:

Nota

Id Alumno	Id Asignatura	Id Periodo	Calificación
1	1	1	**3,5**
1	3	1	**4,5**
1	4	2	**5,0**
2	1	1	**2,5**
2	3	1	**3,0**
3	1	1	**4,0**
3	3	1	**2,7**
4	1	2	**3,8**
4	3	2	**3,2**
5	1	2	**3,0**
6	1	2	**4,0**

Alumno Programa

Id Alumno	Id Programa
1	1
2	1
3	1
4	2
5	2
6	2

Observando detenidamente la tabla que surge de la relación *Alumno Programa*, ésta presenta el mismo número de filas que la tabla *Alumno*. Esto se debe a que la relación es muchas a una. En este tipo de relaciones, la tabla producto de la relación, puede ser absorbida por la tabla que corresponde a la entidad con cardinalidad "muchas". Por tanto, la tabla *Alumno Programa* desaparece y la tabla *Alumno* queda así:

Alumno

Id Alumno	Apellidos	Nombres	Carnet	Nivel	Id Programa
1	GUTIERREZ LOPEZ	ADRIANA MARIA	9019451	2	1
2	MENESES AGUIRRE	OLGA LUCIA	9020678	3	1
3	TOBON VELASQUEZ	ELKIN DARIO	9021023	3	1
4	MUNOZ PEREZ	JORGE ELIECER	9110345	4	2
5	DÍAZ HINESTROSA	ANA MARIA	8620639	8	2
6	IDARRAGA GOMEZ	JAIME ELADIO	8711608	7	2

La anterior "absorción" sólo puede hacerse en tablas producto de relaciones "una a muchas" o "muchas a una".

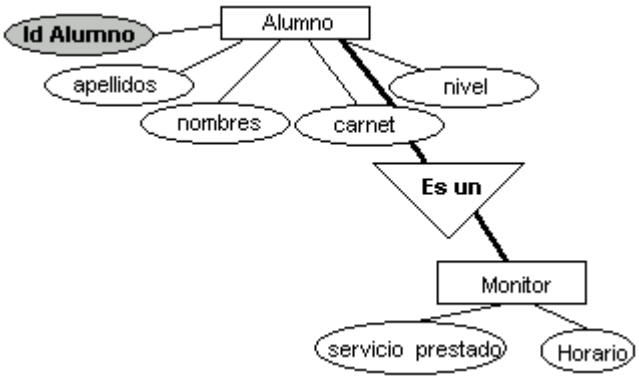
Generalización y Especialización

La **Generalización** es el resultado de la unión de dos o más conjunto de entidades de bajo nivel para producir un conjunto de entidades de más alto nivel. Cada entidad de alto nivel deberá ser también de bajo nivel.

La **Especialización** es el resultado de tomar un subconjunto de un conjunto de entidades de alto nivel para formar un conjunto de entidades de más bajo nivel.

La generalización se utiliza para resaltar las semejanzas entre los tipos de entidades de bajo nivel y para ocultar sus diferencias. La especialización hace todo lo contrario, resalta la distinción entre los conjuntos de entidades de alto y bajo nivel. Esto se realiza mediante la **herencia de atributos**. Se dice que los conjuntos de entidades de bajo nivel ***heredan*** los atributos de los conjuntos de entidades de alto nivel.

Para nuestro caso, supóngase que se da una nueva entidad de estudiantes *monitor*, los cuales además de estudiar, prestan un servicio a la universidad. Esta entidad heredaría de la entidad *alumno* todos los atributos que distinguen a un estudiante y se le añadirían los atributos de *servicio prestado* y *horario*. El diagrama entidad relación sería el siguiente:



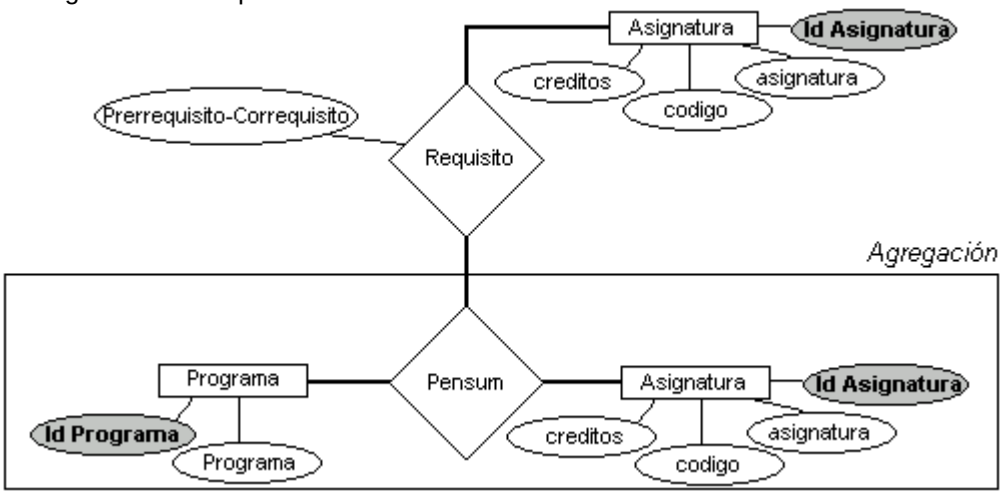
En términos de un diagrama E-R, tanto la generalización como la especialización se representan por medio de un componente triángulo marcado "**Es un**" el cual representa, por ejemplo, que *monitor* "es un" *alumno*.

Agregación

Una limitación del modelo E-R es que no es posible expresar relaciones entre relaciones. La solución es utilizar la **agregación**, la cual es una abstracción por medio de la cual las relaciones se tratan como entidades de alto nivel. Así, puede considerarse a un conjunto de relaciones y los conjuntos de entidades a los que asocia como una entidad de alto nivel que se maneja de la misma manera que cualquier entidad.

En nuestro ejemplo, se desea crear la relación *Requisito*, la cual asocia la relación *Pensum* con la entidad *Asignatura* para determinar que asignaturas son prerrequisitos o correquisitos de otras.

El diagrama E-R respectivo sería:



Normalización

La Normalización es un concepto que hace referencia a las relaciones. Básicamente, el principio de **Normalización** indica que las tablas de las bases de datos eliminarán las incoherencias y minimizarán la ineficacia.

Las bases de datos se describen como **incoherentes** cuando sus datos se introducen de forma incoherente o cuando los datos de una tabla no coinciden con los datos introducidos en otra tabla. Por ejemplo, se la mitad del personal piensa que el municipio de "Tame" se encuentra en la zona Andina de Colombia y la otra mitad que en la Orinoquía, y si ambas partes del personal tratan sus datos de acuerdo con esto, los informes de la base de datos sobre las estadísticas de la Zona Andina no serán significativos.

Una base de datos **ineficaz** no permite aislar los datos exactos que desea. Una base de datos que almacene todos sus datos en una tabla obligará a pasar por innumerables campos simplemente para recuperar un dato. Por otra parte, una base de datos completamente normalizada almacena cada información en su propia tabla e identifica cada información con su propia clave principal.

Un administrador de base de datos puede encontrarse con bases de datos que no han sido normalizadas, por un motivo u otro. La falta de normalización puede ser intencionada o resultado de una falta de experiencia o cuidado del diseñador original de la base de datos. En cualquier caso, si se elige normalizar una base de datos existente, se debe hacer en una etapa temprana del desarrollo de las aplicaciones, ya que este depende de la estructura de las tablas.

La teoría matemática que hay detrás de la normalización es rigurosa y compleja, aunque las comprobaciones que pueden aplicarse para determinar si un diseño tiene sentido son sencillas.

Regla 1: Unicidad de campo. Cada campo de una tabla debe contener un único tipo de información.

Ejemplo: Supongamos que la tabla *Alumno* tiene un campo *Dirección* en el cual se especifica la información donde se localiza el estudiante:

Alumno

Id Alumno	Apellidos	Nombres	Carnet	Nivel	Id Programa	Direccion
1	GUTIERREZ LOPEZ	ADRIANA MARIA	9019451	2	1	CLLE 73 # 80-10 ROBLEDO MEDELLIN TEL 2563426
2	MENESES AGUIRRE	OLGA LUCIA	9020678	3	1	CRA 55 # 20-21 GUAYABAL MEDELLIN TEL 265 34 21
3	TOBON VELASQUEZ	ELKIN DARIO	9021023	3	1	CLLE 94 # 50-13 ARANJUEZ MEDELLIN TEL 2361278
4	MUNOZ PEREZ	JORGE ELIECER	9110345	4	2	CRA 57 # 32-23 BARRIO OBRERO BELLO TEL 4527845
5	DÍAZ HINESTROSA	ANA MARIA	8620639	8	2	CRA 20 # 50-18 EL DORADO ENVIGADO TEL 2768910
6	IDARRAGA GOMEZ	JAIME ELADIO	8711608	7	2	CRA 70 #108-27 FLORENCIA MEDELLIN TEL 2734561

En este ejemplo, el campo *Direccion* deber ser dividido en campos más sencillos, tales como *Dirección, Barrio, Ciudad* y *Telefono*.

Alumno

Id Alumno	Apellidos	Nombres	Carnet	Nivel	Id Programa	Direccion	Barrio	Ciudad	Telefono
1	GUTIERREZ LOPEZ	ADRIANA MARIA	9019451	2	1	CLLE 73 # 80-10	ROBLEDO	MEDELLIN	2563426
2	MENESES AGUIRRE	OLGA LUCIA	9020678	3	1	CRA 55 # 20-21	GUAYABAL	MEDELLIN	265 34 21
3	TOBON VELASQUEZ	ELKIN DARIO	9021023	3	1	CLLE 94 # 50-13	ARANJUEZ	MEDELLIN	2361278
4	MUNOZ PEREZ	JORGE ELIECER	9110345	4	2	CRA 57 # 32-23	BARRIO OBRERO	BELLO	4527845
5	DÍAZ HINESTROSA	ANA MARIA	8620639	8	2	CRA 20 # 50-18	EL DORADO	ENVIGADO	2768910
6	IDARRAGA GOMEZ	JAIME ELADIO	8711608	7	2	CRA 70 #108-27	FLORENCIA	MEDELLIN	2734561

Regla 2: Clave principal. Cada tabla de tener un único identificador, o clave principal, que esté formado por uno o más campos de la tabla.

En un buen diseño de base de datos relacional, cada uno de los registros de cualquier tabla debe ser identificado de forma única. Es decir, algún campo (o combinación de campos) de la tabla debe albergar un valor único para cada registro de la tabla. Este identificador único se denomina clave principal.

Ejemplo: Se tiene la siguiente tabla para almacenar la información de los grupos de una institución universitaria.

Grupo

Apellidos	Nombres	Documento	Id Asignatura	Grupo	Id Periodo
GUTIERREZ PEREZ	JORGE	CC 8567123	1	1	1
CARDONA LOPEZ	PEDRO	CC 71890278	1	2	1
CARDONA LOPEZ	PEDRO	CC 71890278	1	1	2
JIMENEZ ZULUAGA	MIGUEL ANGEL	CC 98543189	3	1	1
JIMENEZ ZULUAGA	MIGUEL ANGEL	CC 98543189	3	1	2
HERNANDEZ GIL	ELENA	CC 43278156	4	1	1
HERNANDEZ GIL	ELENA	CC 43278156	4	1	2

Se puede observar que se repite la información del docente ya que un docente puede dictar cátedra en varios grupos, igual sucede con la asignatura, el período o el consecutivo del grupo. Se puede decir que la clave primaria es la combinación de los campos *Id Asignatura, Grupo* e *Id Período*.

Grupo

Apellidos	Nombres	Documento	Id Asignatura	Grupo	Id Periodo
GUTIERREZ PEREZ	JORGE	CC 8567123	1	1	1
CARDONA LOPEZ	PEDRO	CC 71890278	1	2	1
CARDONA LOPEZ	PEDRO	CC 71890278	1	1	2
JIMENEZ ZULUAGA	MIGUEL ANGEL	CC 98543189	3	1	1
JIMENEZ ZULUAGA	MIGUEL ANGEL	CC 98543189	3	1	2
HERNANDEZ GIL	ELENA	CC 43278156	4	1	1
HERNANDEZ GIL	ELENA	CC 43278156	4	1	2

Esta es una clave algo compleja que podría ser tratada mejor como un índice y en lugar de ella, utilizar un nuevo campo *Id Grupo* que sería un valor entero único de naturaleza autoincrementable (como lo ofrecen los motores de bases de datos actuales).

Grupo

Id Grupo	Apellidos	Nombres	Documento	Id Asignatura	Grupo	Id Periodo
1	GUTIERREZ PEREZ	JORGE	CC 8567123	1	1	1
2	CARDONA LOPEZ	PEDRO	CC 71890278	1	2	1
3	CARDONA LOPEZ	PEDRO	CC 71890278	1	1	2
4	JIMENEZ ZULUAGA	MIGUEL ANGEL	CC 98543189	3	1	1
5	JIMENEZ ZULUAGA	MIGUEL ANGEL	CC 98543189	3	1	2
6	HERNANDEZ GIL	ELENA	CC 43278156	4	1	1
7	HERNANDEZ GIL	ELENA	CC 43278156	4	1	2

Regla 3: Dependencia Funcional. Para cada valor único de la clave principal, los valores de las columnas de datos deben estar relacionados y deben describir completamente el contenido de la tabla.

Esta regla opera de dos formas. En primer lugar, no debería existir ningún dato en la tabla que no sea relevante para el asunto, tema o materia (tal como se definió en la clave principal) de la tabla. En segundo lugar, los datos de la tabla deberían describir completamente la materia de la que trata la tabla.

Ejemplo: En la anterior tabla, aunque es necesaria la información del profesor para cada grupo, los profesores serían de hecho un asunto independiente y tendrían su propia tabla.

Grupo

Id Grupo	Id Profesor	Id Asignatura	Grupo	Id Periodo
1	1	1	1	1
2	2	1	2	1
3	2	1	1	2
4	3	3	1	1
5	3	3	1	2
6	4	4	1	1
7	4	4	1	2

Profesor

Id Profesor	Apellidos	Nombres	Documento
1	GUTIERREZ PEREZ	JORGE	CC 8567123
2	CARDONA LOPEZ	PEDRO	CC 71890278
3	JIMENEZ ZULUAGA	MIGUEL ANGEL	CC 98543189
4	HERNANDEZ GIL	ELENA	CC 43278156

Observe como a la tabla profesor se le agrega el campo *Id Profesor* para identificar el registro de cada docente.

Regla 4: Independencia de los campos. Debe ser posible realizar cambios en cualquier campo que no forme parte de la clave principal sin que para ello se vea afectado cualquier otro campo.

En el anterior ejemplo, se puede observar como al aplicar la regla 3, también se aplicó la regla 4, ya que si se necesitara corregir la escritura de un nombre de docente, se podría hacer sin afectar a otros campos del registro. Sin embargo, si se cometiesen errores en el mismo nombre del docente en muchos grupos, se tendrían que modificar muchos registros Además, no se podría cambiar el nombre del docente sin tener que cambiar los datos de los apellidos y el documento de identidad.

Si se observa la tabla *Alumno* del ejemplo en la regla 1, el campo *Ciudad* es funcionalmente dependiente del campo *Barrio*, por lo que deben ir en tablas aparte.

Alumno

Id Alumno	Apellidos	Nombres	Carnet	Nivel	Id Programa	Direccion	Id Barrio	Telefono
1	GUTIERREZ LOPEZ	ADRIANA MARIA	9019451	2	1	CLLE 73 # 80-10	1	2563426
2	MENESES AGUIRRE	OLGA LUCIA	9020678	3	1	CRA 55 # 20-21	2	265 34 21
3	TOBON VELASQUEZ	ELKIN DARIO	9021023	3	1	CLLE 94 # 50-13	3	2361278
4	MUNOZ PEREZ	JORGE ELIECER	9110345	4	2	CRA 57 # 32-23	4	4527845
5	DIAZ HINESTROSA	ANA MARIA	8620639	8	2	CRA 20 # 50-18	5	2768910
6	IDARRAGA GOMEZ	JAIME ELADIO	8711608	7	2	CRA 70 #108-27	6	2734561

Barrio

Id Barrio	Barrio	Id Ciudad
1	ROBLEDO	1
2	GUAYABAL	1
3	ARANJUEZ	1
4	BARRIO OBRERO	2
5	EL DORADO	3
6	FLORENCIA	1

Ciudad

Id Ciudad	Ciudad
1	MEDELLIN
2	BELLO
3	ENVIGADO

Una alternativa para comprobar la condición de independencia es ver si se tiene la misma información repetida en varios registros.

TEMA 3 Ejercicios de aplicación

1. Defina las entidades y relaciones e ilustre el modelo E-R para almacenar la información para un **sistema contable**.

   Tenga en cuenta:
   - Los planes de cuenta tienen imputaciones que son sumatoria de un conjunto de imputaciones.
   - Los estados financieros son resúmenes de los movimientos (consultas).
   - Se deben registrar los movimientos diarios de caja.

2. Defina las entidades y relaciones e ilustre el modelo E-R para almacenar la información para un **sistema de cuentas bancarias**.

   Tenga en cuenta:
   - La ciudad de documento y residencia de los clientes del banco (Cuentahabientes) son distintas.
   - La sucursal de inscripción de la cuenta es diferente de las de los movimientos.
   - Una persona puede tener varias cuentas y una cuenta puede ser de varias personas.
   - Aprovechar la herencia en la estructura de los tipos de documento, cuenta y movimiento.

3. Defina las entidades y relaciones e ilustre el modelo E-R para almacenar la información para un **sistema de inventario de una empresa comercializadora**.

Tenga en cuenta:

- Las entradas se hacen mediante compras a proveedores.
- Las salidas se hacen mediante ventas a clientes, por devoluciones o por pérdida.
- Los productos tienen precios de compra y de venta.

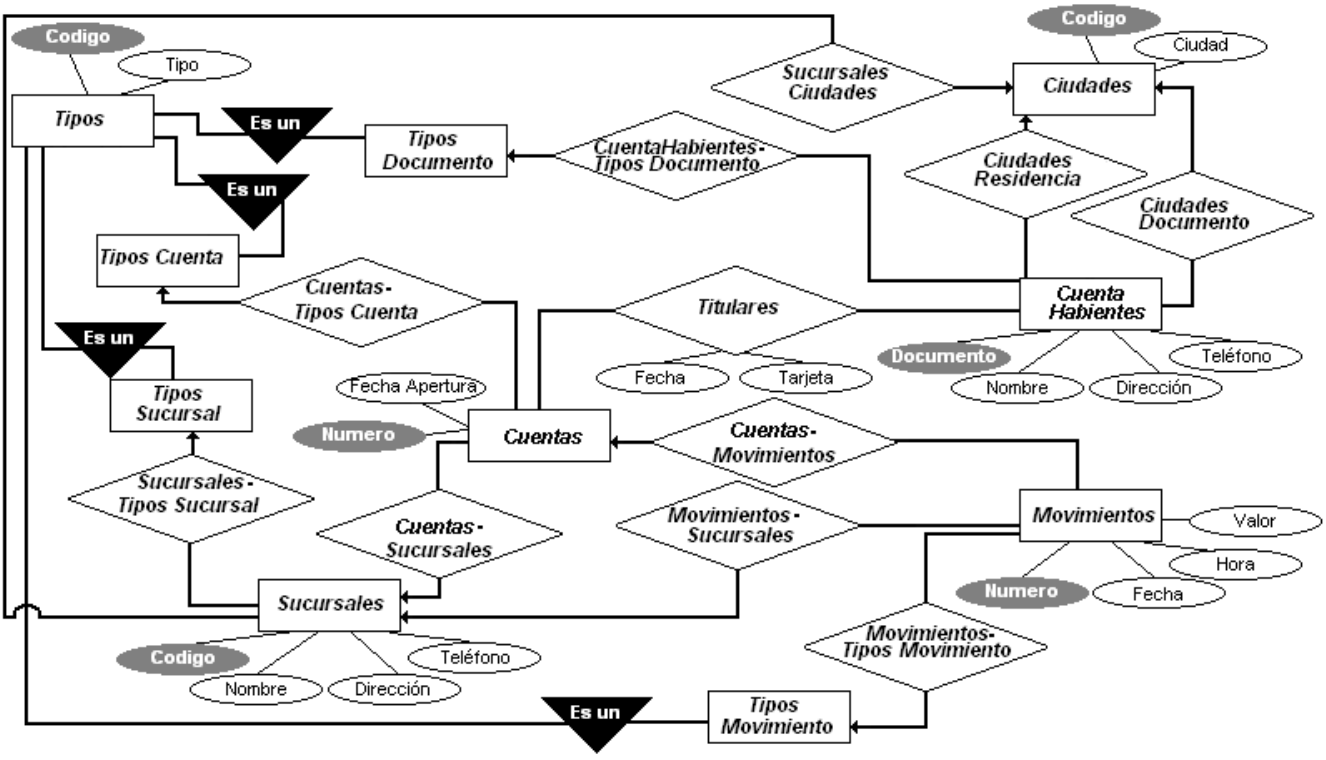
4. Defina las entidades y relaciones e ilustre el modelo E-R para almacenar la información para un **sistema de Registro Académico de un Colegio**.

   Tenga en cuenta:

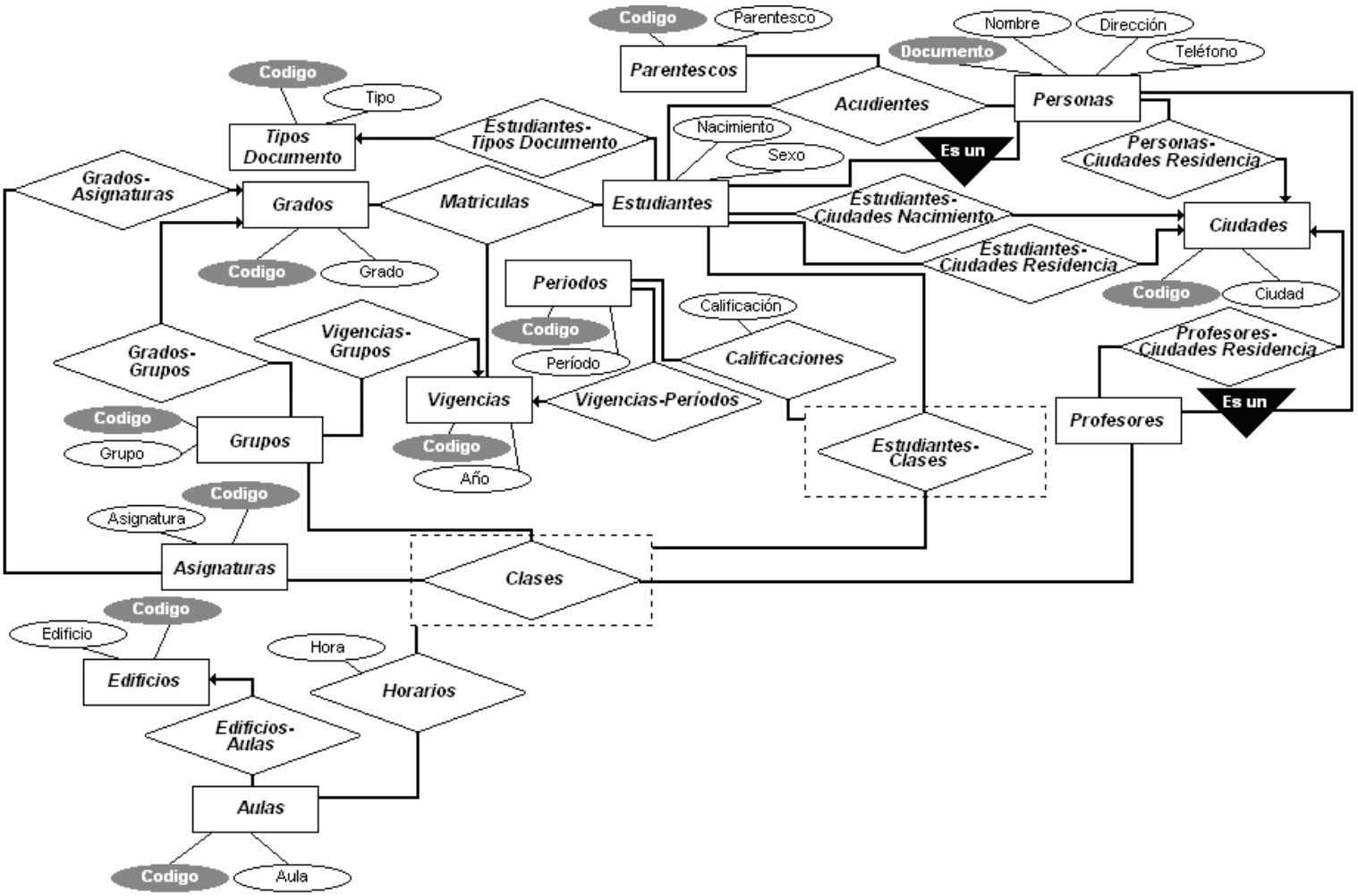
   - Los periodos académicos podrían de más de 1 por año (Educación básica adulta)
   - Los estudiantes pueden tener acudientes
   - Aprovechar la herencia en la estructura de los estudiantes, profesores, acudientes, etc.
   - Los cursos tienen horarios y utilizan espacio físico (Aulas, Bloques)

Respuestas a ejercicios pares

Ejercicio 2



Ejercicio 4


Codigo
Parentesco
Parentescos
Nombre
Dirección
Documento
Teléfono
Personas
Codigo
Tipo
Tipos Documento
Estudiantes-Tipos Documento
Acudientes
Nacimiento
Sexo
Es un
Personas-Ciudades Residencia
Grados-Asignaturas
Grados
Matriculas
Estudiantes
Estudiantes-Ciudades Nacimiento
Ciudades
Estudiantes-Ciudades Residencia
Codigo
Grado
Periodos
Codigo
Ciudad
Calificación
Profesores-Ciudades Residencia
Vigencias-Grupos
Grados-Grupos
Codigo
Período
Calificaciones
Vigencias
Vigencias-Periodos
Es un
Codigo
Grupo
Grupos
Codigo
Año
Profesores
Estudiantes-Clases
Codigo
Asignatura
Asignaturas
Clases
Codigo
Edificio
Hora
Edificios
Horarios
Edificios-Aulas
Aulas
Codigo
Aula