

Diseño de Bases de Datos y el modelo E-R.

2.1 El Proceso de Diseño.

Los diseñadores entrevistan a los futuros usuarios de la base de datos para recoger y documentar sus necesidades de información. En paralelo, conviene definir los requerimientos funcionales que consisten en operaciones (transacciones) que se aplicarán a la base de datos, e incluyen la obtención de datos y la actualización.

Diseño conceptual. Una vez recogidos todos los requerimientos, el siguiente paso es crear un esquema conceptual para la base de datos mediante un modelo de datos conceptual de alto nivel.

El esquema conceptual contiene una descripción detallada de los requerimientos de información de los usuarios, y contiene descripciones de los tipos de datos, relaciones entre ellos y restricciones.

Diseño lógico de la base de datos (transformación de modelo de datos) El siguiente paso en el proceso de diseño consiste en implementar de hecho la base de datos con un SGBD (Sistema Manejador de Base de Datos) comercial, transformando el modelo conceptual al modelo de datos empleados por el SGBD (entidad-relación, jerárquico, red o relacional).

Diseño físico de la base de datos En este paso se especifican las estructuras de almacenamiento internas y la organización de los archivos de la base de datos.

2.2 Modelo Entidad-Relación.

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre estos objetos.

Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos.

El modelo E-R (Entidad-Relación) además de entidades y relaciones representan las uniones que los contenidos de la base de datos deben cumplir. Una unión es la correspondencia de cardinalidades, que expresan el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones.

Conceptos básicos: Existen tres conceptos fundamentales que se emplean en el modelo de datos E-R (Entidad-Relación): conjunto de entidades, conjunto de relaciones y atributos, las cuales se definen a continuación.

Entidad: Se puede definir como Entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar información. Una entidad tiene propiedades y valores que identifican a un sujeto u objeto el cual existe y es distinguible de otros objetos, se representan por un conjunto de atributos, ejemplo entidad cliente: rfc, nombre, dirección, teléfono.

Un conjunto de entidades es un conjunto de entidades del mismo tipo que comparten las mismas propiedades, o atributos.

Atributos: Los Atributos son características o propiedades asociadas a la entidad que toman valor en una instancia particular. Ejemplo: nombre, cédula, teléfono. Cada entidad tiene un valor para cada uno de sus atributos.

Dominio del atributo: Para cada atributo hay un conjunto de valores permitidos, llamados el dominio, o el conjunto de valores, de ese atributo.

Un atributo, como se usa en el modelo E-R, se puede caracterizar por los siguientes tipos de atributo:

- **Atributos simples:** Un atributo simple es aquel que no se puede subdividir, por ejemplo la edad y el sexo de una persona.
- **Atributos compuestos:** Un atributo compuesto, es un atributo que puede ser subdividido en otros atributos adicionales, por ejemplo la dirección de una persona, puede subdividirse en calle, número, código postal, etc.
- **Atributos monovalorados y multivalorados.**
- **Atributos derivados.**

Llave o clave de la relación: Es el identificador único de cada tupla.

Clave primaria: clave candidata que el diseñador elige de la base de datos como el medio principal de identificar entidades dentro de un conjunto de entidades.

Clave compuesta: Una clave compuesta de más de un atributo.

Clave candidata: Cualquier conjunto de atributos que puede ser elegido como clave de una relación.

Clave externa: Un conjunto de atributos o un atributo, en una relación que constituyen una clave en alguna otra relación, usada para establecer enlaces lógicos entre relaciones.

Tupla: Conjunto de atributos que representan a una unidad. Valor nulo: El valor dado a un atributo en una tupla si el atributo es inaplicable o su valor es desconocido.

Cardinalidad: Numero específico de ocurrencias de una entidad, asociadas con una ocurrencia de la entidad relacionada, esto es el número máximo de instancias de un conjunto de objetos que puede estar relacionado con una sola instancia de otro conjunto de objetos.

Relación: Una relación es una asociación entre entidades, se denomina de igual modo a una tabla que se genera a partir de la relación o asociación de dos o más tablas o entidades existentes.

2.3 Restricciones.

Un esquema de desarrollo E-R puede definir ciertas restricciones a las que los contenidos de la base de datos se deben adaptar. En este apartado se examina la correspondencia de cardinalidades y las restricciones de participación, que son dos de los tipos más importantes de restricciones.

La correspondencia de cardinalidades, o razón de cardinalidad, expresa el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.

Reglas de cardinalidad:

Cardinalidad de uno a uno: Una cardinalidad de la interrelación que es 1 en ambas direcciones. Cuando un registro de una tabla sólo puede estar relacionado con un único registro de la otra tabla y viceversa. En este caso la clave foránea se ubica en alguna de las 2 tablas.

Cardinalidad de uno a muchos: Cuando un registro de una tabla (tabla secundaria) sólo puede estar relacionado con un único registro de la otra tabla (tabla principal) y un registro de la tabla principal puede tener más de un registro relacionado en la tabla secundaria. En este caso la clave foránea se ubica en la tabla secundaria.

Cardinalidad de muchos a muchos: Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos (Tabla débil o de vinculación) que incluya los pares de valores relacionados entre sí.

El nombre de tabla débil deviene de que con sus atributos propios no se puede encontrar la clave, por estar asociada a otra entidad. La clave de esta tabla se conforma por la unión de los campos claves de las tablas que relaciona.

Reglas que determinan las interrelaciones (cardinalidad).

Regla 1. Si dos tablas tienen una interrelación de uno a uno (1 a 1), entonces el campo clave de una de las tablas debe aparecer en la otra tabla.

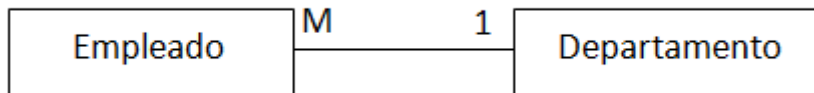
Regla 2. Si dos tablas tienen una interrelación de uno a muchos (1 a *), entonces el campo clave de la tabla del (1) debe aparecer en la tabla del muchos (*).

Regla 3. Si dos tablas tienen una interrelación de muchos a muchos (* a *), entonces debe crearse una tabla que tenga los campos claves de las dos tablas.

Ejemplos:

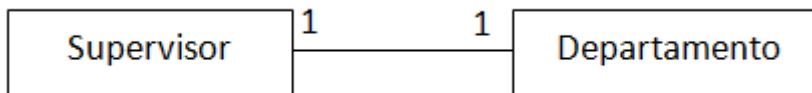
Las relaciones entre entidades se generan en dos direcciones, 1. Si se quiere definir la relación entre un empleado y un departamento se especifica de la siguiente manera:

- Un empleado pertenece a un departamento.
- En un departamento están asignados muchos empleados.
- Esta relación es de tipo 1:M



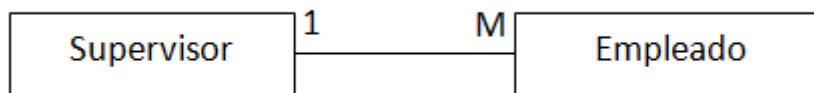
2. Para una interrelación entre un supervisor y un departamento, la cardinalidad es de una a una, en donde el 1 es representado por 1.

- Un supervisor, supervisa un departamento.
- Un departamento es supervisado por una persona.
- En este caso la relación es de uno a uno.



3. Para una interrelación entre un supervisor y empleados, la cardinalidad es de una a mucho, en donde el muchos puede ser representado por un (*), o la letra m.

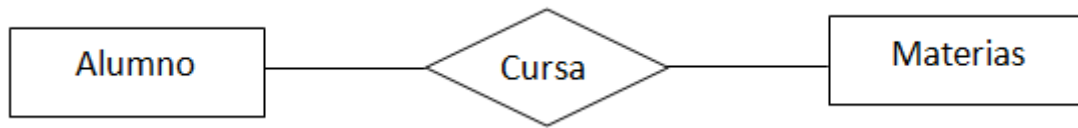
- Un supervisor supervisa a muchos empleados.
- Un empleado es supervisado por un supervisor.
- En este caso la relación es de uno a muchos.



4. En una relación entre alumnos y materias cursadas, la cardinalidad es de muchos a muchos.

- Un alumno cursa muchas materias.
- Una materia la cursan muchos alumnos.
- En este caso es una interrelación de muchos a muchos.

- Una materia la cursan muchos alumnos.
- En este caso es una interrelación de muchos a muchos.

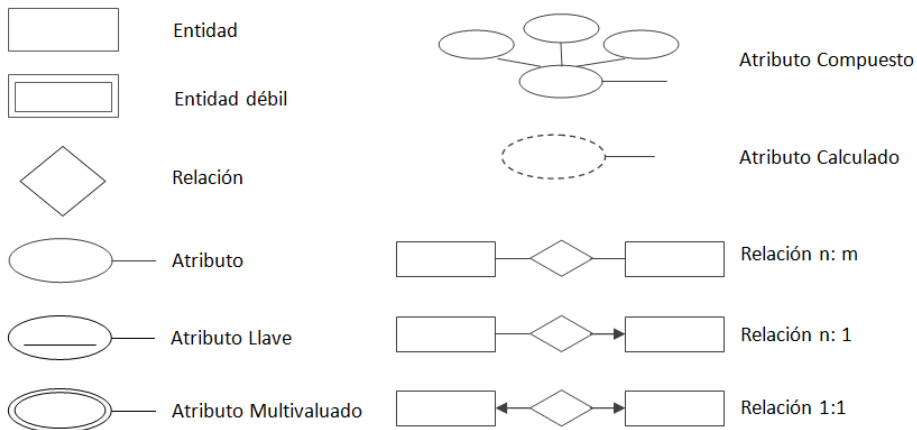


Como en este ejemplo se tiene una relación de muchos a muchos, se genera una tercera entidad débil (Cursa), que se forma con las llaves primarias de la entidad Alumno y Materias.

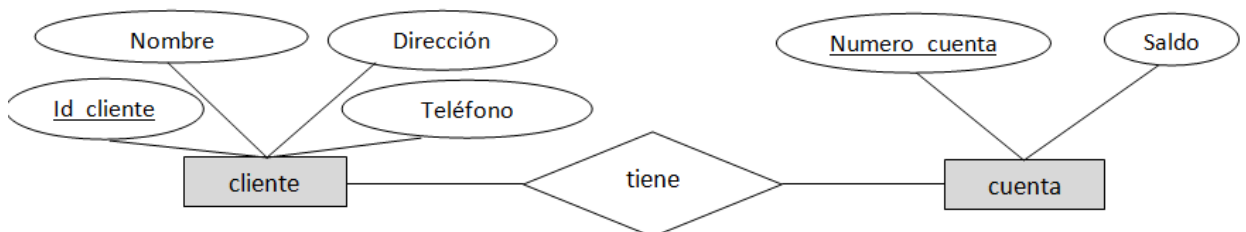
2.4 Diagramas E-R.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama E-R. Los diagramas son simples y claros, cualidades que pueden ser responsables del amplio uso del modelo E-R. Tal diagrama consta de los siguientes componentes principales:

Rectángulos, que representan conjuntos de entidades. **Elipses**, que representan atributos. **Rombos**, que representan relaciones. **Líneas**, que unen atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones. **Elipses dobles**, que representan atributos multivalorados. **Elipses discontinuas**, que denotan atributos derivados. **Líneas dobles**, que indican participación total de una entidad en un conjunto de relaciones. **Rectángulos dobles**, que representan conjuntos de entidades débiles.



Ejemplo de un diagrama Entidad-Relación:



En este modelo se representa a las entidades cliente y cuenta, además de una tabla generada por la relación denominada tiene.

La tabla cliente contiene los atributos: Id_cliente, Nombre, Dirección, Teléfono. La tabla cuenta contiene los atributos: Numero_cuenta, Saldo.

2.5 Diseño con diagramas E-R.

El modelo de datos E-R da una flexibilidad sustancial en el diseño de un esquema de bases de datos para modelar una empresa dada. En este apartado se considera cómo un diseñador de bases de datos puede seleccionar entre el amplio rango de alternativas. Entre las decisiones que se toman están las siguientes:

- Si se usa un atributo o un conjunto de entidades para representar un objeto.
- Si un concepto del mundo real se expresa más exactamente mediante un conjunto de entidades o mediante un conjunto de relaciones.
- Si se usa una relación ternaria o un par de relaciones binarias.
- Si se usa un conjunto de entidades fuertes o débiles; un conjunto de entidades fuertes y sus conjuntos de entidades débiles dependientes se pueden considerar como un objeto en la base de datos, debido a que la existencia de las entidades débiles depende de la entidad fuerte.

Un modelo de datos de alto nivel sirve al diseñador de la base de datos para proporcionar un marco conceptual en el que especificar de forma sistemática los requisitos de datos de los usuarios de la base de datos que existen, y cómo se estructurará la base de datos para completar estos requisitos. La fase inicial del diseño de bases de datos, por tanto, es caracterizar completamente las necesidades de datos esperadas por los usuarios de la base de datos. El resultado de esta fase es una especificación de requisitos del usuario. Esta estructura general se puede expresar gráficamente mediante un diagrama E-R.

A continuación, el diseñador elige un modelo de datos y, aplicando los conceptos del modelo de datos elegido, traduce estos requisitos a un esquema conceptual de la base de datos. El esquema desarrollado en esta fase de **diseño conceptual** proporciona una visión detallada del desarrollo. Debido a que sólo se ha estudiado el modelo E-R hasta ahora, se usará éste para desarrollar el esquema conceptual. En términos del modelo E-R, el esquema especifica todos los conjuntos de entidades, conjuntos de relaciones, atributos y restricciones de correspondencia.

El diseñador revisa el esquema para confirmar que todos los requisitos de datos se satisfacen realmente y no hay conflictos entre sí. También se examina el diseño para eliminar características redundantes. Lo importante en este punto es describir los datos y las relaciones, más que especificar detalles del almacenamiento físico.

Un esquema conceptual completamente desarrollado indicará también los requisitos funcionales de la empresa. En una **especificación de requisitos funcionales** los usuarios describen los tipos de operaciones (o transacciones) que se realizarán sobre los datos. Algunos ejemplos de operaciones son la modificación o actualización de datos, la búsqueda y recuperación de datos específicos y el borrado de datos. En esta fase de diseño conceptual se puede hacer una revisión del esquema para encontrar los requisitos funcionales.

El proceso de trasladar un modelo abstracto de datos a la implementación de la base de datos consta de dos fases de diseño finales. En la **fase de diseño lógico**, el diseñador traduce el esquema conceptual de alto nivel al modelo de datos de la implementación del sistema de base de datos que se usará. El diseñador usa el esquema resultante específico a la base de datos en la siguiente **fase de diseño físico**, en la que se especifican las características físicas de la base de datos.

2.6 Conjunto de entidades débiles.

Las entidades que no tienen atributos llave se conocen como entidades débiles. Las entidades de este tipo se identifican relacionándolas con otras entidades en combinación con algunos de sus atributos. Esa otra entidad se denomina entidad fuerte o propietaria.

Una entidad débil siempre tiene una dependencia de existencia (restricción de participación total) con respecto a la entidad fuerte.

Como ya se mencionaba anteriormente cuando se tiene una relación de muchos a muchos entre dos entidades se genera una tercera entidad denominada "débil", una entidad débil no tiene llave primaria y sus atributos principales son las llaves primarias de las entidades que se están relacionando.

2.7 Modelo E-R extendido.

Aunque los conceptos básicos de E-R pueden modelar la mayoría de las características de las bases de datos, algunos aspectos de una base de datos pueden ser más adecuadamente expresados mediante ciertas extensiones del modelo E-R básico. A continuación se definen las características E-R extendidas de especialización, generalización, conjuntos de entidades de nivel más alto y más bajo, herencia de atributos y agregación.

La **especialización y la generalización** definen una relación de contenido entre un conjunto de entidades de nivel más alto y uno o más conjuntos de entidades de nivel más bajo. La especialización es el resultado de tomar un subconjunto de un conjunto de entidades de nivel más alto para formar un conjunto de entidades de nivel más bajo. La generalización es el resultado de tomar la unión de dos o más conjuntos disjuntos de entidades (de nivel más bajo) para producir un conjunto de entidades de nivel más alto. Los atributos de los conjuntos de entidades de nivel más alto los heredan los conjuntos de entidades de nivel más bajo.

La **agregación** es una abstracción en la que los conjuntos de relaciones (junto con sus conjuntos de entidades asociados) se tratan como conjuntos de entidades de nivel más alto, y pueden participar en las relaciones.

2.8 Otros aspectos del diseño de bases de datos.

Al momento de diseñar un modelo de base de datos se deben considerar los siguientes aspectos:

- Obtener los Requisitos de datos.
- Designación de los conjuntos de entidades
- Designación de los conjuntos de relaciones
- Realizar el diseño del Diagrama E-R

2.9 La Notación E-R con UML.

Los diagramas entidad-relación ayudan a modelar el componente de representación de datos de un sistema software. La representación de datos, sin embargo, sólo forma parte de un diseño completo de un sistema.

Otros componentes son modelos de interacción del usuario con el sistema, especificación de módulos funcionales del sistema y su interacción, etc. El lenguaje de modelado unificado (UML, Unified Modeling Language) es un estándar propuesto para la creación de especificaciones de varios componentes de un sistema software. Algunas de las partes de UML son:

- **Diagrama de clase.** Un diagrama de clase es similar a un diagrama E-R.
- **Diagrama de caso de uso.** Los diagramas de caso de uso muestran la interacción entre los usuarios y el sistema, en particular los pasos de las tareas que realiza el usuario.
- **Diagrama de actividad.** Los diagramas de actividad describen el flujo de tareas entre varios componentes de un sistema.
- **Diagrama de implementación.** Los diagramas de implementación muestran los componentes del sistema y sus interconexiones tanto en el nivel del componente software como el hardware.

3 Diseño de bases de datos relacionales.

3.1 Características del diseño relacional.

El modelo relacional, está basado en las relaciones lógicas entre los datos, este modelo organiza y representa a los datos en forma de tablas de dos dimensiones, consistente en filas y columnas de datos.

El concepto de base de datos relacional fue escrito por primera vez por el Dr. Codd en 1970 el cual publico un artículo en que aplicaba los conceptos de una rama de las matemáticas llamada algebra relacional, a los problemas de almacenar enormes cantidades de datos. Este artículo dio inicio a un movimiento en la comunidad de las bases de datos que en muy poco tiempo condujo a la definición del modelo de base de datos relacionales.

El modelo relacional surge como un intento de simplificar la estructura de las bases de datos, eliminando estructuras padre/hijo del modelo jerárquico de la base de datos y en su lugar representar todos los datos en la base de datos como tablas conformada a su vez por renglones y columnas con valores de datos.

Este modelo, es un modelo simple, poderoso y formal de representar la realidad, que facilita la construcción de consultas de usuario.

El objetivo del diseño de las bases de datos relacionales es la generación de un conjunto de esquemas relacionales que nos permita almacenar la información sin redundancias innecesarias, pero que también nos permita recuperar fácilmente esa información.

Normalización:

La normalización de bases de datos relacionales toma un esquema relacional y le aplica un conjunto de técnicas para producir un nuevo esquema que representa la misma información pero contiene menos redundancias y evita posibles anomalías en las inserciones, actualizaciones y borrados. La técnica de normalización es semejante a lo que comúnmente se dice de que un párrafo debe tener un solo tema, si un párrafo tiene más de un tema, debe dividirse en tantos párrafos como temas se consideren. La lógica que se aplica a la normalización es cada afinidad normalizada tiene un solo tema, si tiene dos o más, deberá fragmentarse en afinidades, cada una de las cuales tendrá un solo tema.

Estas clases de afinidades y las técnicas para prevenir las anomalías son llamadas formas normales. Dependiendo de su estructura, una afinidad puede estar en primera forma normal, segunda forma normal o alguna otra. En su artículo Ted Codd, estableció la primera, segunda y tercera forma normal. Cada una de estas formas están anidadas, esto es una afinidad que está en tercera forma, debe estar en primera y segunda forma normal.

3.2 Dominios atómicos y la primera forma normal.

La primera de las formas normales que se van a estudiar, la primera forma normal, impone un requisito muy elemental a las relaciones; a diferencia de las demás formas normales, no exige información adicional como las dependencias funcionales.

Un dominio es atómico si se considera que los elementos del dominio son unidades indivisibles.

Una afinidad esta en primera forma normal, si la tupla tiene un campo definido como campo clave y todos sus valores son atómicos para cada atributo en la relación. Esto es que los valores de los atributos no pueden ser un conjunto de valores o un grupo repetitivo.

Cualquier tabla de datos que cumpla con la definición de una afinidad, se dice que está en primera forma normal.

Si en la tabla encontramos grupos repetitivos es necesario crear una tabla de relación que interrelacione a las tablas determinadas.

4 Modelo relacional.

El modelo relacional se ha establecido actualmente como el principal modelo de datos para las aplicaciones de procesamiento de datos. Ha conseguido la posición principal debido a su simplicidad, que facilita el trabajo del programador en comparación con otros modelos.

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada tabla tiene una estructura parecida a la presentada en la unidad 2, donde se representaron las bases de datos E-R mediante tablas.

4.1 Estructura básica.

Una tabla en el modelo relacional viene a ser como una de las listas descritas anteriormente. Una tabla o relación es una matriz rectangular que almacena líneas con una estructura concreta:

- La primera línea de una tabla, es una cabecera que indica el nombre de cada columna. O sea, cada columna tiene asignado un nombre único.
- Cada línea (excepto la primera) recibe el nombre de tupla, y almacena ítemes concretos para cada columna.
- Todas las filas deben ser diferentes entre sí.
- El orden de las filas y de las columnas carece de importancia a efectos del S.G.B.D. Este hecho es el que verdaderamente diferencia las tablas relacionales del concepto matemático de relación, en el que el orden de las columnas es fundamental.

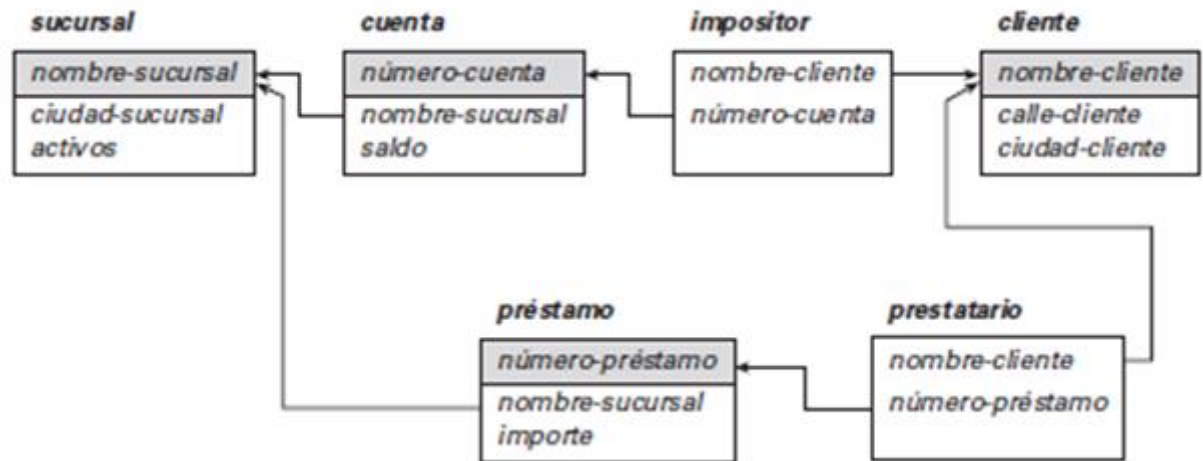
Dominio de Valores. Los dominios a que puede pertenecer un atributo, suelen depender de los que proporcione el S.G.B.D. que empleemos. Suelen ser comunes dominios como: Texto, Número entero, Número decimal, Fecha, Hora, Sí/No, etc.

Por otro lado, un dominio como pueda ser Número entero, es un dominio cuyo conjunto de valores es infinito, y dado que trabajamos con ordenadores, es imprescindible poner un límite que permita almacenar un valor concreto debido a las limitaciones de memoria, y sobre todo al hecho de que toda tupla debe poseer el mismo tamaño.

4.2 Esquema de las bases de datos.

Un esquema de bases de datos, junto con las dependencias de clave primaria y externa, se puede mostrar gráficamente mediante diagramas de esquema.

Ejemplo de un esquema de base de datos:



4.3 Claves.

- Una clave es un atributo o conjunto de atributos cuyo valor es único y diferente para cada tupla.
- Tenemos dos claves potenciales, también llamadas claves candidatas.
- De entre todas las claves candidatas, el administrador, cuando define la tabla, debe decidir cuál de ellas va a ser la clave principal o clave primaria, mientras que el resto de las claves pasan a denominarse claves alternativas o claves alternas.

Si el esquema de una base de datos relacional se basa en las tablas derivadas de un esquema E-R es posible determinar la clave primaria del esquema de una relación a partir de las claves primarias de los conjuntos de entidades o de relaciones de los que se deriva el esquema:

Conjunto de entidades fuertes. La clave primaria del conjunto de entidades se convierte en la clave primaria de la relación.

Conjunto de entidades débiles. La tabla y, por tanto, la relación correspondientes a un conjunto de entidades débiles incluyen:

- Los atributos del conjunto de entidades débiles.
- La clave primaria del conjunto de entidades fuertes del que depende el conjunto de entidades débiles.
- La clave primaria de la relación consiste en la unión de la clave primaria del conjunto de entidades fuertes y el discriminante del conjunto de entidades débil.

Conjunto de relaciones. La unión de las claves primarias de los conjuntos de entidades relacionados se transforma en una superclave de la relación. Si la relación es de varios a varios, esta superclave es también la clave primaria.

4.4 Lenguajes de consulta.

Un lenguaje de consulta es un lenguaje en el que un usuario solicita información de la base de datos. Estos lenguajes suelen ser de un nivel superior que el de los lenguajes de programación habituales. Los lenguajes de consulta pueden clasificarse como procedimentales o no procedimentales. En los lenguajes procedimentales el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos para calcular el resultado deseado. En los lenguajes no procedimentales el usuario describe la información deseada sin dar un procedimiento concreto para obtener esa información.

La mayor parte de los sistemas comerciales de bases de datos relacionales ofrecen un lenguaje de consulta que incluye elementos de los enfoques procedimental y no procedimental

