**Resumen Bases de Datos**

Cap1: Conceptos de Bases de datos.

* 1. Base de datos.
* Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo de organización.
* Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos se integran con una misma cantidad de duplicidad. De este modo, la base de datos no pertenece a un solo departamento, sino que se comparte por toda la organización. Además, la base de datos no solo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción se denomina metadatos, se almacena en el diccionario de datos o catalogo y es lo que permite que exista independencia de datos lógica-física.

1.2: Sistema de gestión de base de datos.

* El sistema de gestión de base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma. Se denomina sistema de bases de datos al conjunto formado por la base de datos, el SGBD y los programas de aplicación que dan servicio a la empresa u organización.
* Los sistemas de bases de datos siguen el modelo conocido como abstracción de datos, donde los usuarios solo ven la especificación externa y no se deben preocupar de como esta implementado internamente. Los sistemas de bases de datos separan la definición de la estructura física de los datos de su estructura lógica y almacenan esta definición en la base de datos. Todo esto es gracias a la existencia del SGBD, que se sitúa entre la base de datos y los programas de aplicación.

* Generalmente, un SGBD proporciona los siguientes servicios:
* El SGBD permite la definición de la base de datos mediante un lenguaje de definición de datos, que permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos.
* El SGBD permite la inserción, actualización, eliminación y consulta de datos mediante un lenguaje de manejo de datos. Hay dos tipos de lenguaje de manejo de datos, que se distinguen por el modo en el que acceden a los datos:
* Procedurales: Manipulan la base de datos registro a registro.

Además, se especifica que operaciones se debe realizar para obtener los resultados.

* No Procedurales: Operan sobre conjuntos de datos de registro, y se especifica que datos deben obtenerse sin decir cómo. Por ejemplo: SQL.
* Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos.
* Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos.
* Un sistema de integridad que mantiene la integridad y la consistencia de los datos.
* Un sistema de control de concurrencia que permite el acceso compartido a la base de datos.
* Un sistema de control de recuperación que restablece la base de datos después de que se produzca una falla en el hardware o software.
* Un diccionario de datos o catálogo, accesible por el usuario, que contiene la descripción de los datos de la base de datos.

1.3: Personas en el entorno de la base de datos

* Administrador de la base de datos: Se encarga de la implementación física de la base de datos. Además, se encarga de establecer la política de seguridad y del acceso concurrente. También se debe preocupar de que el sistema se encuentre funcionando y con un buen rendimiento.
* Diseñadores de la base de datos: Realizan el diseño de la base de datos, debiendo identificar los datos, relaciones entre ellos las restricciones entre datos y relaciones. Además, deben tener un profundo conocimiento de los datos de la empresa y sus reglas de negocio.
* Programadores de aplicación: Se encargan de implementar los programas de aplicación que sería a los usuarios finales, que en los que permiten consultar datos, insertados, actualizados o eliminarlos.
* Usuarios finales: Son los clientes de la base de datos.

1.5: Ventajas e inconveniente de los sistemas de bases de datos.

* Ventajas:

1. Control sobre la redundancia de los datos: No se almacenan varias copias de los mismos datos.
2. Control sobre la consistencia de datos: si un dato esta almacenado una sola vez, cualquiera actualización se debe realizar solo una vez, y esta disponible para todos los usuarios que estén autorizados.
3. Compartición de datos: La base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.
4. Mantenimiento de estándares: gracias a la integración, es más fácil respetar los estándares.
5. Mejora en la integridad de los datos: Restricciones o reglas que no se pueden violar, tanto en los datos como en sus relaciones.
6. Mejora en la seguridad: Protección de la base de datos frente a usuarios no autorizados.
7. Mejora en la accesibilidad de los datos: Lenguajes de consulta o generadores de informes que permiten al usuario hacer cualquier tipo de consultas sobre los datos.
8. Mejora en la productividad: El SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.
9. Mejora en el mantenimiento: Los SGBD separan las descripciones de datos de las aplicaciones. Esto se conoce como independencia de datos.
10. Aumento de la concurrencia: La mayoría de los SGBD gestionan el acceso concurrente a la base de datos.
11. Mejora en los servicios de copias de seguridad y de recuperación ante fallas: Los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

* Inconvenientes:

1. Alta complejidad: Los SGBD son programas muy complejos con una gran funcionalidad.
2. Gran tamaño: Los SGBD requieren gran cantidad de espacio en disco y memoria.
3. Coste económico del SGBD: Los SGBD son muy costosos, aunque en los últimos años han surgido SGBD open source.
4. Coste de equipamiento adicional: Puede que sea necesario adquirir más espacio de almacenamiento o una maquina más grande.
5. Coste de conversión: Coste elevado para convertir la aplicación actual en un sistema de bases de datos.
6. Prestaciones: Los SGBD al tener muchas funcionalidades, pueden no ser tan rápidas.
7. Vulnerable a los fallos: Por el hecho de que todo se encuentre centralizado en el SGBD.

Cap2: Modelo Relacional

2.1. Modelo de datos

Unas de las características fundamentales de los sistemas de bases de datos es que proporcionan cierto nivel de abstracción de datos, ocultar las características sobre el almacenamiento físico que la mayoría de los usuarios no necesita conocer.

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos; los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos contienen un conjunto de operaciones básicas para la realización de consultas (lecturas) y actualizaciones de datos. Los modelos de datos modernos incluyen mecanismos para especificar acciones compensatorias o adicionales que se deben llevar a cabo ante las acciones habituales que se realizan sobre la base de datos. Los modelos de datos se pueden clasificar dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos, formando una jerarquía de niveles. Tanto los modelos de datos de alto nivel, o modelos conceptuales hacen que la mayoría de los usuarios perciban los datos, mientras que los modelos de bajo nivel o modelos físicos describen los detalles de como se almacenan los datos en el ordenador. Los modelos lógicos ocultan de como se almacenan los datos, pero pueden implementarse de manera directa en un SGBD.

Los modelos conceptuales utilizan conceptos como entidades, atributos y relaciones. Una entidad representa un objeto o concepto del mundo real, ejemplo: “un cliente de una empresa o una de sus facturas”. Un atributo representa alguna propiedad de interés de una entidad, ejemplo “el nombre o el domicilio del cliente”. Una relación describe una interacción entre dos o mas entidades, ejemplo “la relación que hay entre un cliente y las facturas que le han realizado”.

Cada SGBD soporta un modelo lógico, siendo los más común el relacional, la red y el jerárquico.

En el modelo relacional los datos se describen como un conjunto de tablas con referencias lógicas entre ellas, mientras que en los modelos jerárquicos y de red, los datos se describen como conjuntos de registros con referencias físicas entre ellos (punteros). Los modelos físicos describen como se almacenan los datos, el formato de los registros, la estructura de los ficheros (ordenado, desordenado, agrupados) y los métodos de acceso utilizados (índices, tablas de dispersión).

A la descripción de una base de datos mediante un modelo de datos se la denomina esquema de la base de datos. Los datos que se almacenan en las bases de datos pueden cambiar con mucha frecuencia: se insertan datos, se actualizan, se borran, etc. Los datos que contienen las bases de datos se denominan ocurrencia de la base de datos. Cuando definimos una nueva base de datos, sólo especificamos su esquema al SGBD. En ese momento, el estado de la base de datos es el estado vacío, sin datos. Cuando se cargan datos por primera vez, la base datos pasa al estado inicial. De ahí en adelante, siempre que se realice una operación de actualización de la base de datos, se tendrá un nuevo estado. Cuando se cargan datos por primera vez, la base datos pasa al estado inicial. De ahí en adelante, siempre que se realice una operación de actualización de la base de datos, se tendrá un nuevo estado. El SGBD se encarga, en parte, de garantizar que todos los estados de la base de datos sean estados válidos que satisfagan la estructura y las restricciones especificadas en el esquema.

El modelo relacional representa la segunda generación de los SGBD. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica.

2.2.1 Relaciones

El modelo relacional se basa en el concepto matemático de relación, que gráficamente se representa mediante una tabla. Una relación es una tabla con columnas y filas. Un atributo es el nombre de una columna de una relación. En el modelo relacional, las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos. Una relación se representa gráficamente como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas corresponden a los campos o atributos de esos registros. Un dominio es el conjunto de valores legales de uno o varios atributos. Los dominios constituyen una poderosa característica del modelo relacional. Cada atributo de una base de datos relacional se define sobre un dominio, pudiendo haber varios atributos definidos sobre el mismo dominio.

Una tupla es una fila de una relación. Los elementos de una relación son las tuplas o filas de la tabla. Las tuplas de una relación no siguen ningún orden. El grado de una relación no cambia con frecuencia.

La cardinalidad de una relación es el número de tuplas que contiene. Ya que en las relaciones se van insertando y borrando tuplas a menudo, la cardinalidad de estas varía constantemente.

Una base de datos relacional es un conjunto de relaciones normalizadas. Una relación está normalizada si en la intersección de cada fila con cada columna hay un solo valor.

2.2.2: Propiedades de las relaciones.

* Cada relación tiene un nombre, y éste es distinto del nombre de todas las demás.
* Los dominios sobre los que se definen los atributos son escalares, por lo que los valores de los atributos son atómicos. De este modo, en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están normalizadas.
* No hay dos atributos que se llamen igual.
* El orden de los atributos no importa: los atributos no están ordenados.
* Cada tupla es distinta de las demás: no hay tuplas duplicadas.
* El orden de las tuplas no importa: las tuplas no están ordenadas.

2.2.3: Tipos de las relaciones.

En un SGBD relacional hay dos tipos de relaciones:

* Relaciones base: Son relaciones reales que tienen nombre, y forman parte directa de la base de datos almacenada. Se dice que las relaciones base son relaciones autónomas.
* Vistas: También denominadas relaciones virtuales, son relaciones con nombre y derivadas (no autónomas). Que son derivadas significa que se obtienen a partir de relaciones: se representan mediante su definición en términos de esas otras relaciones. Las vistas no poseen datos almacenados propios, los datos que contienen corresponden a datos almacenados en relaciones base.

2.2.4: Claves.

Se denomina superclave a un atributo o conjunto de atributos que identifican de modo único las tuplas de una relación. Se denomina clave candidata a una superclave en la que ninguno de sus subconjuntos es una superclave de la relación.

* Uniciad: Nunca hay dos tuplas en la relación R con el mismo valor que K.
* Irreducibilidad: (Minimalidad): Ningún subconjunto de K tiene la propiedad de unicidad, no se pueden eliminar componentes de K sin destruir la unicidad.

Una clave formada por mas de un atributo es una clave compuesta. Una relación puede tener varias claves candidatas.

Para identificar las claves candidatas de una relación no hay que fijarse en un estado u ocurrencia de la base de datos. La presencia de duplicados si es útil ya que esto permite saber si es posible que aparezcan duplicados. Solo utilizando información semántica se puede saber con certeza si un conjunto de atributos forma una clave candidata. Se denomina clave primaria de una relación a aquella clave candidata que se escoge para identificar sus tuplas de modo único. una relación no tiene tuplas duplicadas, siempre hay una clave candidata, por lo tanto, la relación siempre tiene clave primaria.

2.3: Esquema de una base de datos relacional.

Una base de datos relacional es un conjunto de relaciones. Para representar el esquema de una base de datos relacional se debe dar el nombre de sus relaciones, los atributos de éstas, los dominios sobre los que se definen estos atributos, las claves primarias y las claves ajenas. (Ver ejemplo en el apunte)

2.4: Reglas de integridad.

Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo. A este tipo de restricciones se les denomina restricciones de dominios. Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados (las reglas se deben cumplir todo el tiempo). Estas reglas son la regla de integridad de entidades y la regla de integridad referencial.

2.4.1: Nulos.

Un nulo no representa el valor cero ni la cadena vacía ya que éstos son valores que tienen significado. El nulo implica ausencia de información, bien porque al insertar la tupla se desconocía el valor del atributo, o bien porque para dicha tupla el atributo no tiene sentido.

Ya que los nulos no son valores, deben tratarse de modo diferente, lo que causa problemas de implementación. De hecho, no todos los SGBD relacionales soportan los nulos.

2.4.2: Reglas de integridad de entidades.

La primera regla de integridad se aplica a las claves primarias de las relaciones base: ninguno de los atributos que componen la clave primaria puede ser nulo.

Por definición, una clave primaria es una clave irreducible que se utiliza para identificar de modo único las tuplas. Irreducible significa que ningún subconjuto de la clave primaria sirve para identificar las tuplas de modo único. Si se permitiera que parte de la clave primaria fuera nula, se estaría diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se estaría contradiciendo la irreductibilidad.

2.4.2: Reglas de integridad referencial.

Regla de integridad se aplica a las claves ajenas: si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos.

La regla de integridad referencial se enmarca en términos de estados de la base de datos: indica lo que es un estado ilegal, pero no dice cómo puede evitarse. Por lo tanto, una vez establecida la regla, hay que plantearse qué hacer si estando en un estado legal, llega una petición para realizar una operación que conduce a un estado ilegal. Existen dos opciones: rechazar o aceptar la operación y realizar operaciones adicionales compensatorias que conduzcan a un estado legal.

Para hacer respetar la integridad referencial se debe contestar, para cada clave ajena, a las tres preguntas que se plantean a continuación y que determinarán su comportamiento:

 Regla de los nulos: «¿Tiene sentido que la clave ajena acepte nulos?»

 Regla de borrado: «¿Qué ocurre si se intenta borrar la tupla referenciada por la clave ajena?»

* Restringir: no se permite borrar la tupla referenciada.
* Propagar: se borra la tupla referenciada y se propaga el borrado a las tuplas que la referencian mediante la clave ajena.
* Anular: se borra la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).
* Valor por defecto: se borra la tupla referenciada y las tuplas que la referenciaban ponen en la clave ajena el valor por defecto establecido para la misma.

Regla de modificación: «¿Qué ocurre si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave ajena?



* Restringir: no se permite modificar el valor de la clave primaria de la tupla referenciada.
* Propagar: se modifica el valor de la clave primaria de la tupla referenciada y se propaga la modificación a las tuplas que la referencian, mediante la clave ajena.
* Anular: se modifica la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).
* Valor por defecto: se modifica la tupla referenciada y las tuplas que la referenciaban ponen en la clave ajena el valor por defecto establecido para la misma.
* Regla de los nulos: la clave ajena acepta nulos, por lo que es posible encontrar facturas cuyo cliente se ignore (esto se ha decidido así porque lo impone un requisito del usuario).
* Regla de borrado: anular. Cuando se elimine un cliente de la base de datos y se proceda a borrarlo de la relación CLIENTES, se deberán anular todas las referencias que hubiera desde FACTURAS.codcli. De este modo, todas las facturas que tenía ese cliente pasarán a tener un nulo en el código del cliente. (EJEMPLO APUNTE).
* Regla de modificación: propagar. En caso de que se modifique el código a un cliente (quizá porque el sistema de codificación se cambie por parte de la empresa), todas las facturas de dicho cliente actualizarán el valor de FACTURAS.codcli para continuar haciendo referencia a la misma tupla. (EJEMPLO DEL APUNTE).

2.4.2: Reglas de negocio.

Estas reglas forman parte de la estrategia de funcionamiento de la empresa. El SGBD debe dar la posibilidad al usuario de definir una regla al respecto y debe hacerla respetar.

Cap6: Diseño Conceptual

6.1: Modelo entidad-relación.

* Los modelos conceptuales se utilizan para representar la realidad a un alto nivel de abstracción para armar una descripción de esta que sea fácil de entender.
* Deben poseer las siguientes cualidades:

1. Expresividad: Deben tener suficientes conceptos para expresar perfectamente la realidad.
2. Simplicidad: Deben ser simples para que los esquemas sean fáciles de entender.
3. Minimalidad: Cada concepto debe tener un significado distinto.
4. Formalidad: Todos los conceptos deben tener una interpretación única, precisa y bien definida.

* Imagen que contiene mujer, listo, sostener, competencia de atletismo

  Descripción generada automáticamenteConceptos

6.1.1: Entidades

* Para identificar las entidades se buscan conceptos importantes como personas, lugares o conceptos abstractos, excluyendo aquellos que solo son propiedades de otros objetos. También se pueden identificar buscando conceptos que existen por sí mismos.

6.1.2: Relaciones

* Para identificar las relaciones entre entidades, se suelen buscar expresiones verbales. Las relaciones suelen ser binarias, es decir, entre dos entidades, aunque también puede haber relaciones en las que participen más de dos entidades.
* Una vez identificadas todas las relaciones, hay que determinar la cardinalidad mínima y máxima con la que participa cada entidad en cada una de ellas. La cardinalidad es un tipo de restricción que se utiliza para comprobar y mantener la calidad de los datos.
* La cardinalidad mínima indica si la participación de la entidad en la relación es opcional (se indica con 0) o si es obligatoria (se indica con 1).
* La cantidad máxima indica si cada ocurrencia de la entidad solo puede relacionarse con una ocurrencia de la entidad con varias a la vez (se indica con n).

6.1.3: Atributos

* Los atributos son los nombres que identifican propiedades, cualidades, identificadores o características de entidades o relaciones.

6.1.4: Dominio

* El dominio de un atributo es el conjunto de valores que puede tomar el atributo, además de su tamaño y formato.

6.1.5: Identificadores

* Cada entidad tiene al menos un identificador y de cada entidad se escogerá uno de los identificadores como calve primaria.
* Si una entidad tiene al menos un identificador, es una entidad fuerte. De lo contrario, se considera una entidad débil.

6.1.6: Jerarquías de generalización

* Para identificarlas es necesario reflejar las diferencias entre distintas ocurrencias de una entidad, con lo que surgirán nuevas subentidades de esta entidad genérica; o bien, si hay entidades que tienen características en común y que realmente son subentidades de una nueva entidad genérica.

6.2: Restricciones

1. Dos entidades no se pueden conectar directamente con una línea.
2. No puede haber conexiones entre dos relaciones.
3. Los atributos se asocian a entidades y a relaciones, pero no se asocian a las líneas que las conectan.
4. Cuando una entidad participa en una relación, se debe indicar siempre la cardinalidad con la que participa.
5. Un atributo se dibuja solo una vez en el esquema.
6. Puede haber nombres de atributos iguales en distintas entidades siempre que tengan significados diferentes
7. Los atributos simples se representan mediante círculos pequeños conectados directamente a la entidad o relación con una línea en la que se especifica la cardinalidad.
8. Los atributos compuestos se representan mediante un ovalo y tendrán atributos simples conectados directamente a él.
9. La cardinalidad de un atributo no expresa su rango de valores, sino el numero de valores distintos que puede tener el atributo a la vez. El rango de valores posibles está indicando a través dl dominio.
10. Si un atributo tiene un número fijo posible de valores estos no se dibujan como componentes de un atributo compuesto. Al especificar el dominio del atributo será cuando se especifiquen los posibles valores.
11. Todas las entidades deben tener al menos un identificador. Los atributos con cardinalidad máxima “n” no pueden ser identificadores.
12. Cuando un identificador esta formado por vario atributos, estos no tendrán su círculo coloreado. Los atributos que forman el identificador se deben dejar sin colorear, se conectan con una línea y al final de la misma se dibuja un circulo coloreado.
13. Las relaciones no tienen identificadores.
14. Una entidad débil es aquella que depende de otra para identificarse. El identificador de la entidad débil estará formado por uno o varios de sus atributos, en combinación con el identificador de la entidad de la que depende. Esto expresa conectando con una línea dichos atributos, y la línea que conecta a la entidad con la relación de dependencia. Al final de la línea se dibujará un círculo coloreado, expresando aso el identificador. Los demás atributos que lo forman no deben colorearse.

Cap7: Diseño Lógico-Racional

* El diseño lógico es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo de información especifico e independiente del SGBD que vaya a utilizar.
* El objetivo del diseño lógico es obtener una representación que use, del modo mas eficiente posible, los recursos que el modelo de SGBD posee para estructurar datos y modelos las restricciones.
* El fin de esta etapa es obtener el esquema lógico, que estará formado por las tablas de la base de datos en tercera forma normal, a partir de la especificación realizada en la etapa del diseño conceptual.
* Para cada tabla del esquema lógico se debe especificar:

1. Nombre y descripción de la información que almacena.
2. Para cada columna, indicar: nombre, tipos de datos, si admite nulos, valor por defecto y rango de valores.
3. Clave primaria.
4. Claves alternativas
5. Claves foráneas y sus reglas de comportamiento ante el borrado y la modificación de la clave primaria a la que referencia.
6. Si alguna columna es un dato derivado indicar como se obtiene su valor.
7. Indicar las restricciones a nivel de fila de cada tabla, si las hay.
8. Indicar las restricciones no expresadas antes.
9. Especificar las reglas de negocio, que serán aquellas acciones que se deba llevar a cabo de forma automática como consecuencia de actualizaciones que se realicen sobre la base de datos.
10. Introducir tablas de referencia para establecer listas de valores para columnas que las necesiten.

7.2: Metodología de diseño

* A cada tabla se le dará un nombre, y el nombre de sus atributos aparecerá a continuación, entre paréntesis. El o los atributos que forman la clave primaria se subrayan. Las claves foráneas se especificarán indicando la tabla a la que hacen referencia.

7.3: Entidad fuertes

* Se debe crear una tabla para cada entidad fuerte, incluyendo todos sus atributos simples con cardinalidad máxima 1. De los atributos compuestos con cardinalidad máxima 1, incluir solo sus componentes.
* Cada atributo con cardinalidad máxima n se incluirá como una tabla dentro de la tabla correspondiente a la entidad. Si el atributo es simple, la tabla interna tendrá una sola columna. Si el atributo es compuesto, tendrá tantas columnas como componentes tenga.
* Cada uno de los identificadores de la entidad será una calve candidata. De entre las claves candidatas, hay que escoger la clave primaria; el resto serán claves alternativas.

7.2.2: Entidades Débiles

* Se debe crear una tabla para cada entidad débil teniendo en cuenta todos sus atributos. Una entidad débil participa en una relación con la entidad fuerte de la que depende y su cardinalidad siempre será (1,1).
* Se incluye la clave primaria de la tabla que representa a la entidad fuerte en la nueva tabla creada para la entidad débil. Luego se debe determinar la clave primaria de la nueva tabla.

7.2.3: Relaciones Binarias

* Una relación binaria es aquella en la que participan dos entidades o una entidad cuyas ocurrencias se relacionan entre sí.
* Según su cardinalidad, las relaciones binarias se clasifican en:

1. Uno a uno: Ambas entidades participan con cardinalidad máxima 1.
2. Uno a muchas: Una entidad participa con cardinalidad máxima n.
3. Muchos a muchos: Ambas entidades participan con cardinalidad máxima n.

7.2.4: Jerarquía de generalización

* Dependiendo del tipo de jerarquía (total/parcial, exclusiva/superpuesta), se pueden representar de 3 formas distintas:

1. Crear una tabla por cada entidad -> cualquier tipo de jerarquía.
2. Crear una tabla por cada entidad hija, heredando los atributos de la entidad madre -> totales o exclusivos.
3. Integrar todas las entidades en una sola tabla, incluyendo en los atributos de la entidad madre, las de sus hijas y un atributo discriminativo para indicar el subconjunto al cual pertenece la entidad en consideración -> cualquier tipo de jerarquía.

7.2.5: Normalización

* La normalización se utiliza para mejorar el esquema lógico, de modo que satisfaga ciertas restricciones que eviten la duplicidad de estos.
* Una base de datos normalizada proporciona los siguientes beneficios:

1. Organiza los datos de acuerdo con sus relaciones lógicas.
2. Obliga a entender completamente cada uno de los atributos que se han de representar en la base de datos.
3. Esta libre de anomalías que las redundancias pueden provocar cuando se actualiza la base de datos.
4. Base de datos más fáciles de manejar.
5. Esquemas flexibles que pueden extenderse con facilidad.

* Dependencia Funcional: Una dependencia funcional es una restricción y representa una relación de uno a muchos, o de uno a uno.
* Formas Normales:

1. Forma normal (primer):

* Todos los atributos son atómicos.
* Clave primaria única si no contiene no es 1° FN.
* No debe existir variación en el número de columnas.
* Los campos no clave deben identificarse por la clave.
* Debe existir una independencia del orden de filas y columnas.

1. Forma normal (segunda): Una tabla esta en 2° FN si delante de estar en 1°FN, cumple que los atributos no clave dependen de toda la clave principal.
2. Forma normal(tercera): Una tabla esta en 3°FN si además de estar en 2°FN, no existe ninguna dependencia transitiva entre los atributos que no son clave.
3. Forma normal (Boyee Codd): Una tabla está en FN de Boyee Codd si está en 3°FN y los únicos determinantes (atributos que dependen de otros) son claves candidatas.