Relacionales eran las más utilizadas en la mayoría de los ámbitos de aplicación la estructura de datos básicos que ofrece la tabla relacional es apropiada para muchas aplicaciones habituales en el ámbito empresarial sin embargo existen casos en los que presenta serios inconvenientes prácticos en generalmente si se requiere gestionar datos muy complejos o no convencionales por ejemplo imágenes o documentos extensos en para ellos la estructura relacional resulta ineficiente algunos ejemplos de gran importancia práctica son las bases de datos multimedia las bases de datos científicas y los sistemas de apoyo al diseño industrial sistemas cad cam en tenemos así una primera generación que fue las bases de datos jerárquicas luego una segunda generación de bases de datos relacionales y como consecuencia de lo que dijimos aparece 1/3 generación de base de datos orientadas a objetos las bases de datos orientadas a objetos soportan el paradigma orientado a objetos almacenando datos y métodos y no sólo los datos como eran las relacionales esta característica Evita el acceso a los datos mediante el control que ejerzan los métodos almacenados en el objeto permiten el desarrollo de aplicaciones utilizando el mismo modelo conceptual de análisis diseño y programación orientada a objetos sin necesidad de mapearlo a un modelo conceptual diferente reduciendo el problema de la conversión

una base orientada a objetos es una colección de datos modelados en objetos de los lenguajes de programación orientados a objetos que tienen la capacidad de encapsular los atributos y las operaciones definidas sobre ellos y mediante las cuales interactúan los objetos se agrupan en clases que son las mismas utilizadas en los lenguajes de programación orientada a objetos y que se estructura en una jerarquía de súper clases y subclases se definen por datos y comportamientos llamados atributos y métodos como un todo un atributo es una propiedad o característica de la clase y describe un rango de valores que podrán contener los objetos o instancias de esa clase un método es una acción que la clase puede realizar un objeto de la clase de persona por ejemplo tendrá como atributos el DNI el nombre la edad la empresa donde trabaja etc el valor de los atributos es propio de cada instancia pero entre algunos objetos se comparten el comportamiento LA orden para ejecutar un comportamiento se efectiviza mediante un mensaje se efectiviza mediante un mensaje si el comportamiento y los datos son aceptados por el protocolo del objeto se ejecutará si no es rechazado ahí tenemos la responsabilidad y ejecución del control por parte del objeto para nuestro objeto persona se han definido comportamientos como actuliazar la edad, cambiar la empresa, etc. Si cambia la empresa se ejecutará el método cambiar empresa, nombre de la empresa.

Hay 3 conceptos que son fundamentales en el marco teórico de las bases de datos orientadas a objetos encapsulamiento herencia y polimorfismo correcto el encarcelamiento podemos decir que las metodologías de desarrollo clásicas datos y procesos se tratan por separado hay un modelo de datos donde en el diccionario figuran las estructuras y las relaciones y hay un modelo de procesos donde se representa en el intercambio entre entidades almacenamientos y procesos el hecho de que un objeto sea una estructura indivisible de atributos que lo describen y métodos de comportamiento permiten por un lado el tratamiento de ambos aspectos desde el inicio del modelado del sistema y no en etapas como en las metodologías tradicionales y por otro lado permite reducir el potencial de errores ya que la responsabilidad de control y ejecución reside en el mismo objeto cada objeto oculta su funcionalidad mediante la propiedad de encapsulamiento estableciendo atributos y métodos públicos y privados para permitir la comunicación entre ellos presenta una interfaz que recibe o envía mensajes a modo de peticiones para realizar operaciones con las aplicaciones de usuario.

Herencia eres es el mecanismo que permite la reutilización y extensibilidad del software una clase implementa el tipo de objeto y una subclase hereda las propiedades de su clase padre agregando nuevos atributos y métodos este concepto lleva a una organización jerárquica que facilita el mantenimiento se modifican los niveles superiores y los inferiores heredan la modificación.

Polimorfismo es la propiedad de una operación mediante la cual puede ser aplicada a distintos objetos siendo el sistema quien define qué acción exactamente debe ejecutarse un ejemplo clásico de polimorfismo es el siguiente tenemos 3 clases diferentes círculo rectángulo y cuadrado que heredan la superclase forma la clase forma tiene el método calcular área que se implementa de distinta forma en cada una de las subclases en la subclase círculo será el algoritmo el algoritmo tipo radio al cuadrado en la subclase de rectángulo será base por altura y en la subclase cuadrado será lado al cuadrado.

En el sistema de gestión de base de datos orientada a objetos es un software específico dedicado a servir de interfaz entre la base de objetos el usuario y las aplicaciones que la utilizan con la particularidad de almacenar objetos permitiendo persistencia concurrencia y recuperación de los mismos el hecho de que las bases de datos orientadas a objetos y los lenguajes de programación orientada a objetos compartan el mismo modelo de datos permite desarrollar aplicaciones sobre la base de datos orientada a objetos de manera casi transparente sin embargo hay que tener en cuenta que desde el lenguaje de programación orientado a objetos es necesario indicar qué objetos se guardarán en la base de datos es decir serán persistentes y cuáles no la propiedad de persistencia define qué objetos deberán permanecer en la base de datos una vez que el programa haya terminado y cuáles no los objetos transitorios desaparecen una vez que el programa que los creó termina de ejecutarse mientras que los objetos persistentes permanecen almacenados en la base de datos

El conjunto de reglas para los sistemas orientados a objetos se conoce como el manifiesto de las bases de datos orientadas a objetos y fue definido en 1009 92 consta de las 3:00 reglas obligatorias para considerar que un sistema gestor de base de datos es orientado a objetos según las 13 reglas del manifiesto orientado a objetos llamado manifiesto de atkinson las características son unos debe soportar objetos complejos debe ser posible construir objetos complejos conformados por objetos simples aplicando constructores a objetos básicos sin necesidad de desagregarlo en objeto simple en 2 identidad del objeto todos los objetos deben tener un identificador llamado o ID o a identifier el cual es independiente de los valores de sus atributos es único inmutable generado por el sistema y no es visible para el usuario externo 3 encapsulamiento los programadores sólo tienen acceso a la especificación de interfaz de los métodos y los datos e implementación de estos métodos están ocultos en los objetos cuarto tipos o clases el esquema de una base de datos orientada a objetos contiene un conjunto de clases o tipos que especifican una estructura de datos y los métodos operativos permisibles que se aplican a cada uno de los objetos 5 jerarquía de tipo o clases deben ser capaces de heredar por especialización generalización los atributos y los métodos el subtipo hereda todas las funciones del súper tipo dando lugar a una jerarquía de tipos o clases.

Sobrecargue polimorfismo distintas clases podrán proporcionar distintos métodos para una misma operación el sistema determinará qué método corresponde ejecutar el lenguaje de manipulación de datos de esta manipulación language debe ser completo en los sistemas de gestor de base de datos orientados a objetos debe ser un lenguaje de programación de propósito general que puede expresar cualquier algoritmo el conjunto de tipos de datos debe ser extensible debe permitir la creación de nuevos tipos de datos a partir de los existentes y no habrá distinción entre los tipos definidos por el usuario y los tipos definidos por el sistema asistencia de datos los datos deben mantenerse después que la aplicación que lo escribió haya finalizado el usuario no tiene que hacer una copia explícitamente el sistema gestor de base de datos debe ofrecer funciones de gestión de almacenamiento eficiente incluyendo el uso de índices almacenamiento en memoria intermedia selección de rutas de acceso y optimización de consultas el sistema gestor de base de datos debe soportar la concurrencia debe disponer del mecanismo para el control de sincronización el acceso de más de un usuario al mismo recuperación el sistema que esto debe proveer mecanismos para mantener el nivel de servicio en caso de fallas y el sistema gestor de base de datos orientado a objetos debe de proveer un lenguaje de consulta específico para objetos.

El manifiesto propone también 5 opciones de implementación no obligatorias herencia múltiple un objeto tiene características de padres diferentes y herencia selectiva hereda solo algunas de las características del padre verificación e inferencia de tipos le quita al programador la tarea de especificar los tipos de datos distribución de los datos en diferentes servidores transacciones complejas entre usuarios interactivos cooperantes o secuencias de sus transacciones y racionado de objetos algunos sistemas de orientados a objetos cuentan con capacidades para manejar múltiples versiones del mismo objeto esta característica es útil para conservar una versión anterior hasta que la nueva se haya probado y verificado por ejemplo.

Una de las motivaciones para el desarrollo de las bases de datos orientadas a objetos fue el deseo de representar objetos complejos puede ser de 2 tipos estructurados y no estructurados un objeto complejo estructurado está constituido por componentes y se define aplicando recurrentemente en diversos niveles los constructores de tipos disponibles como mínimo listas y arreglos uno no estructurado casi siempre es un tipo de datos que requiere una gran cantidad de espacio de almacenamiento como por ejemplo un tipo de datos que representa una imagen se conocen como objetos binarios extensos o bailar estos objetos son no estructurados en el sentido de que el sistema gestor de la base de datos no sabe qué estructura tienen sólo la aplicación que los usa sabe cómo interpretarlos se consideran complejos porque necesitan un área de almacenamiento sustancial y no forman parte de los tipos de datos estándar que suelen ofrecer los sistemas gestores de base de datos como el tamaño de los objetos es considerable el sistema gestor podría obtener una porción del objeto y proporcionarla al programa antes de recuperar todo el objeto también podría usar técnicas de papers y de almacenamiento memoria caché para obtener por anticipado partes del objeto cómo un sistema gestor de bases de datos orientado a objetos permite a los usuarios crear nuevos tipos y como un tipo incluye tanto la estructura como las operaciones podemos considerar que un sistema gestor de base de datos tiene un sistema de tipo extensibles se pueden crear bibliotecas de nuevos tipos definiendo estructura y operaciones otras aplicaciones pueden utilizar estos tipos e incluso modificarlo creando subtipos de los tipos provistos en la biblioteca.

Existen 2 tipos de semántica para la referencia entre un objeto complejo y sus componentes en cada nivel el primer tipo es conocido como agregación y es una relación conocida como parte de que denota una jerarquía del todo a la parte el segundo tipo se lo conoce como generalización especialización y denota una relación del tipo es un en el ejemplo vemos la relación de agregación entre el objeto perro y sus componentes hocico oreja cola es parte de el objeto perro en el objeto mariposa con los suyos a la antena son parte de mariposa en el ejemplo también está conceptualizado una relación generalización especialización entre la clase de animal y sus componentes perro y mariposa es un animal.

BMG es el estándar de facto para los modelos de objetos o MG que es object data management group es un grupo de representantes de la industria de bases de datos que fue creado en 1009 91 con el objeto de definir estándares para los sistemas gestores de base de datos ya que las bases de datos orientadas a objetos no contaban con un estándar como sí lo tiene el análisis y diseño orientado a objetos que es UML unified modeling language es el modelo para la semántica de los objetos en una base de datos la versión 3.0 propone los siguientes componentes principales de la arquitectura de un sistema gestor de base de datos un lenguaje de definición de objetos o de l object definition language la definición de una base de datos está contenida en un esquema que sea creado mediante el lenguaje de definición de datos o de l qué es el lenguaje de manejo de datos es el equivalente al del DL reina definition language de los sistemas gestores de bases de datos tradicionales un lenguaje de consulta de objetos o QL object query language con sintaxis similar a la de sql pero con características adicionales como identidad de objetos objetos complejos operaciones herencia polimorfismo y relaciones y conexión con los lenguajes c más más small dog y java al menos okay no es limitante esta lista.

Los componentes básicos de una base de datos proyectada objetos son los objetos es un objeto es una representación abstracta de una entidad del mundo real que tiene una identidad o identificador único dentro de la base de datos y propiedades que incluyen sus atributos y un comportamiento que le proporciona la capacidad de interaccionar con otros objetos y corporal los objetos que comparten propiedades y comportamientos forman clases NG los elementos que componen un objeto son identidad y la propiedad que permite diferenciar a cada objeto de los restantes se representa por medio de un proyecto ahí que es el único parece objeto es asignado por el sistema en el momento en que el objeto es creado y no puede ser alterado bajo ninguna circunstancia eh no es visible para el usuario externo sino que el sistema lo utiliza internamente para identificar al objeto y para referenciarlo desde otros objetos constructores de tipos y objetos los lenguajes de definición de datos orientados a objetos permiten definir los tipos de datos y los tipos de objetos necesarios para construir los objetos de una aplicación concreta referencia entre objetos los atributos cuyo dominio sea un tipo de objetos simplemente como referencia usando el objeto a identificar y representar relaciones entre los objetos estado de un objeto es el valor que tiene asignado dicho objeto en un momento determinado esto significa que cuando las propiedades de un objeto cambian su valor debe ser actualizado en la base de datos haciendo pasar el objeto a un nuevo estado comportamiento de los objetos determinan las tareas que puede realizar a partir de mensajes recibidos de otros objetos se representa por medio de operaciones que se ejecutan entre ellos toda operación que deba realizarse sobre un objeto tiene que estar previamente declarada las operaciones se usan para crear y eliminar objetos para cambiar los valores de sus atributos para conocer su valor o para realizar operaciones clasificación de objetos la base de datos orientada a objetos clasifican los objetos de acuerdo Asus similitudes una clase es un conjunto de objetos que comparten unas propiedades objeto y un comportamiento operaciones la distanciación es el proceso inverso al de clasificación y consiste en generar los objetos de una clase.

Todos los accesos, creación, modificación y borrado de objetos persistentes se debe realizar dentro de una transacción que son unidades lógicas de trabajo que llevan a la base de datos de un estado consistente a otro estado consistente el modelo asume una secuencia lineal de transacciones que se ejecutan de modo controlado la concurrencia se maneja en base a bloqueos de lectura y escritura el modelo especifica operaciones para iniciar terminar el cómic y abortar transacciones utilizando el bloqueo de 2 fases así como la operación de checkpoint esta última operación hacer permanentes los cambios realizados por la transacción en curso sin liberar ninguno de los bloqueos adquiridos.

Al igual que en la base de datos relacionales el modelo lógico de datos de las bases de datos orientadas a objetos es independiente de la localización física de los datos los datos también se almacenan en archivos emplazados en diferentes dispositivos físicos que el administrador de la base de datos tiene que gestionar muchos de los conceptos de diseño físico de las bases de datos relacionales se pueden aplicar a las bases de datos orientados a objetos sin embargo hay ciertas cuestiones que deben ser estudiadas en particular por ejemplo los índices de acceso a los datos en las principales tipos de estructuras de indexación que podemos encontrar en las bases de datos son los índices de atributos que se usan para indexar los objetos de una clase según alguno de sus atributos de esta manera se pueden recuperar rápidamente los objetos de una clase que cumplan cierta propiedad sobre el atributo indexado por ejemplo indexar los objetos de una clase empleados por su atributo y los índices de objetos que se usan para indexar los objetos de una clase según su identificador o idea esta índice contiene una entrada con el identificador de cada objeto de la clase y un puntero a su dirección de memoria esta alternativa resulta muy útil para navegar por los objetos ya que para pasar de un objeto de una clase a otra que referencia desde alguno de sus atributos se utiliza el índice.

Amor resumen veamos cuáles son las ventajas y convenientes de las bases de datos orientadas a objetos las bases de datos orientadas a objetos las ventajas que proporcionan un modelo de datos mucho más rico y extensible y así complementan aunque no sustituyen a las bases de datos relacional esto se debe a que un objeto permite encapsular tanto un estado como un comportamiento un objeto puede almacenar todas las relaciones que tenga con otros objetos y los objetos pueden agruparse para formar objetos complejos mediante herencia logran una equivalencia al de los lenguajes de programación orientados a objetos como c más más o java esto tiene la gran ventaja de que al compartir el modelo de datos se pueden integrar las bases de datos orientadas a objetos con el software usado para desarrollar aplicaciones de manera directa y casi transparente en las bases de datos relacionales es necesario usar 2 lenguajes uno de programación para la aplicación QL para el acceso a la base de datos disponen de nuevas características para el modelado de datos por ejemplo almacenar grandes objetos binarios como imágenes sonido o vídeo las herramientas de gestión de documentos índices y operadores específicos para buscar texto inclusive como soporte de xml y otras extensiones para soportar datos geográficos y además permiten la reusabilidad de clases lo que repercute en una mayor facilidad de mantenimiento y un menor tiempo de desarrollo inicialmente no podemos dejar de mencionar los inconvenientes que aún presentan los sistemas gestores de bases de datos por un lado la carencia de un modelo de datos universal no hay ningún modelo de datos que esté universalmente aceptado para los sistemas gestores de bases de datos y los modelos carecen en su mayoría de una base teórica por otra parte los sistemas gestores de bases de datos relacionales tienen una experiencia de uso considerable SQLl es un estándar aprobado y además el modelo relacional tiene una sólida base teórica los productos relacionales disponen de muchas herramientas de soporte que sirven tanto para desarrolladores como para usuarios finales que no es el caso de las bases de datos orientadas a objetos