Base de datos relacionales

En una computadora existen diferentes formas de almacenar información. Esto da lugar a distintos modelos de organización de la base de datos: jerárquico, red, relacional y orientada a objeto.

Los sistemas relacionales son importantes porque ofrecen muchos tipos de procesos de datos, como: simplicidad y generalidad, facilidad de uso para el usuario final, períodos cortos de aprendizaje y las consultas de información se especifican de forma sencilla.

Las tablas son un medio de representar la información de una forma más compacta y es posible acceder a la información contenida en dos o más tablas. Más adelante explicaremos que son las tablas.

Las bases de datos relacionales están constituidas por una o más tablas que contienen la información ordenada de una forma organizada. Cumplen las siguientes leyes básicas:

- Generalmente, contendrán muchas tablas.
- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no está determinados.
- Para cada campo existe un conjunto de valores posible.

Diseño de las bases de datos relacionales

El primer paso para crear una base de datos, es planificar el tipo de información que se quiere almacenar en la misma, teniendo en cuenta dos aspectos: la información disponible y la información que necesitamos.

La planificación de la estructura de la base de datos, en particular de las tablas, es vital para la gestión efectiva de la misma. El diseño de la estructura de una tabla consiste en una descripción de cada uno de los campos que componen el registro y los valores o datos que contendrá cada uno de esos campos.

Los campos son los distintos tipos de datos que componen la tabla, por ejemplo: nombre, apellido, domicilio. La definición de un campo requiere: el nombre del campo, el tipo de campo, el ancho del campo, etc.

Los registros constituyen la información que va contenida en los campos de la tabla, por ejemplo: el nombre del paciente, el apellido del paciente y la dirección de este. Generalmente los diferente tispos de campos que su pueden almacenar son los siguientes: Texto (caracteres), Numérico (números), Fecha / Hora, Lógico (informaciones lógicas si/no, verdadero/falso, etc., imágenes.

En resumen, el principal aspecto a tener en cuenta durante el diseño de una tabla es determinar claramente los campos necesarios, definirlos en forma adecuada con un nombre especificando su tipo y su longitud.

Bases de datos orientadas a objetos (BDOO)

¿Qué es O.O.?

En esos mundos OO, el conocimiento se descentraliza en todos los objetos que lo componen, cada objeto sabe hacer lo suyo y no le interesa saber cómo el vecino hace su trabajo, pero sabe que lo hace y qué es lo que puede hacer. Como bien lo definió Dan Ingalls de Smalltalk con las siguientes palabras:

"La orientación a objetos proporciona una solución que conduce a un Universo de Objetos 'bien educados' que se piden de manera cortés, concederse mutuamente sus deseos".

¿Por qué O.O.?

La meta es dejar la etapa en la que la construcción del software es una labor de artesanos, y pasar a la etapa en la que se pueda tener fábricas de software, con gran capacidad de reutilización de código y con metodología eficientes y efectivas que se apliquen al proceso de producción.

¿Qué es una BDOO?

A finales de los 80's aparecieron las primeras BDOO, es una base de datos inteligente. Soporta el paradigma orientado a objetos almacenando datos y métodos, y no sólo datos. Está diseñada para ser eficaz, desde el punto de vista físico, para almacenar objetos complejos. Evita el acceso a los datos; esto es mediante los métodos almacenados en ella. Es más segura ya que no permite tener acceso a los datos (objetos); esto debido a que para poder entrar se tiene que hacer por los métodos que haya utilizado el programador.

Un Modelo Conceptual Unificado

Las técnicas OO utilizan los mismos modelos conceptuales para el análisis, diseño y construcción. La tecnología de las BDOO da un paso más hacia la unificación, el modelo conceptual de la base de datos OO es igual al del resto del mundo OO, en lugar de utilizar tablas por relación independientes como SQL.

El uso del mismo modelo conceptual para todos los aspectos del desarrollo simplifica éste, particularmente con las herramientas CASE OO; mejora la comunicación entre usuarios, analistas y programadores, además de que reduce las posibilidades de error.

Para ver las figuras faltantes haga click en el menú superior "Bajar Trabajo"

(Figura No.2)

El desarrollo tradicional tiene cuatro modelos conceptuales

Arquitectura de Una BDOO

En la siguiente (TablaNo1) se muestran algunos de los principales productos de BDOO y sus vendedores.

Los primeros se diseñaron como una extensión de los lenguajes de programación como Smalltalk ó C++. El LMD (lenguaje para el manipulación de datos; también conocido como DML) y el LDD (lenguaje para la definición de los datos; también conocido como DDL) construían un lenguaje OO común.

El diseño de las BDOO actuales debe aprovechar al máximo él CASE e incorporar métodos creados con cualquier técnica poderosa, incluyendo enunciados declarativos, generadores de códigos e inferencias con base en reglas.

Para ver la tabla faltante haga click en el menú superior "Bajar Trabajo"

(Tabla No.1) (Figura No.3)

Seis Productos de BDOO y sus Proveedores.

Algunas características son independientes de la arquitectura fundamental de una BDOO pero son comunes a

la mayoría de ellas:

* Versiones.− La mayoría de los sistemas de bases de datos sólo permiten que exista una representación de un ente de la base de datos dentro de esta. Las versiones permiten que las representaciones alternas existan en forma simultánea.

* Transacciones compartidas.− Las transacciones compartidas soportan grupos de usuarios en estaciones de trabajo, los cuales desean coordinar sus esfuerzos en tiempo real, los usuarios pueden compartir los resultados intermedios de una base de datos. La transacción compartida permite que varias personas intervengan en una sola transacción

Desarrollo con Bases de Datos OO

Las BDOO se desarrollan al describir en primer lugar los tipos de objetos importantes del dominio de aquellos tipos de objetos. Estos tipos de objetos determinan las clases que conformarán la definición de la BDOO.

Por ejemplo: Una base de datos diseñada para almacenar la geometría de ciertas partes mecánicas incluiría clases como CILINDRO, ESFERA Y CUBO. El comportamiento de CILINDRO podría incluir información relativa a sus dimensiones, volumen área superficial:

Clase de CILINDRO{

ALTURA FLOTANTE ();

RADIO FLOTANTE ();

VOLUMEN FLOTANTE ();

AREA DE SUPERFICIE FLOTANTE ();

Se puede llegar a definiciones similares para el cubo y la esfera. En la definición anterior, ALTURA,RADIO y ÁREA representan los mensajes que se pueden enviar a un objeto CILINDRO .

La Implantación se lleva a cabo en el mismo lenguaje, escribiendo funciones correspondientes a las solicitudes OO:

CILINDRO::ALTURA () {RETORNA CILINDRO_ALTURA;}

CILINDRO::VOLUMEN () {RETORNA PI*RADIO ()*ALTURA ();}

En este caso, la Altura se almacena como un elemento de los datos, mientras que volume se calcula mediante la fórmula apropiada. Observe que la implantación interna de volume utiliza solicitudes para obtener altura y radio. Sin embargo, el aspecto más importante es la sencillez y uniformidad que experimentan los usuarios de CILINDRO. Sólo necesitan conocer la forma de enviar una solicitud y las solicitudes disponibles.

Tres Enfoques de Construcción de Bases de Datos OO

Las BDOO se pueden construir mediante alguno de los tres enfoques siguientes:

El Primero.− se puede utilizar el código actual altamente complejo de los sistemas de administración de las bases de datos, de modo que una BDOO se implante más rápido sin tener que iniciar de cero. Las técnicas orientadas a objetos se pueden utilizar como medios para el diseño sencillo de sistemas complejos. Los

sistemas se construyen a partir de componentes ya probados con un formato definido para las solicitudes de las operaciones del componente.

* El Segundo: considera a la BDOO como una extensión de la tecnología de las bases de datos por relación. De este modo, las herramientas, técnicas, y vasta experiencia de la tecnología por relación se utilizan para construir un nuevo SABD. Se pueden añadir apuntadores a las tablas de relación para ligarlas con objetos binarios de gran tamaño (BLOB). La base de datos también debe proporcionar a las aplicaciones clientes un acceso aleatorio y por partes a grandes objetos, con el fin de que sólo sea necesario recuperar a través de la red la parte solicitada de los datos.

* El Tercero: reflexiona sobre la arquitectura de los sistemas de bases de datos y produce una nueva arquitectura optimizada, que cumple las necesidades de la tecnología OO. Las compañías como Versant, Objectivity, Itasca, etc. Utilizan esté enfoque y afirman que la tecnología de relación es un subconjunto de una capacidad más general. Además que las BDOO no de relación son aproximadamente dos veces más rápidas que las bases de datos por relación para almacenar y recuperar la información compleja. Por lo tanto, son esenciales en aplicaciones como CAD y permitirían que un depósito CASE fuera una facilidad de tiempo real en vez de una facilidad por lotes.

La Arquitectura de Versant está designada al soporte Cliente/Servidor con acercamiento a la computación distribuida; cualquier aplicación de Cliente el servidor la procesa, usa las EDT y las máquinas servidoras que pueden cooperar en una BD distribuida de Versant. Las BD pueden estar levantadas como un sistema m−Cliente/n−Servidor.

Un servidor en el medioambiente de Versant es una máquina que está corriendo los procesos del servidor, esta soporta accesos concurrentes por usuarios múltiples de una o más BD. Un cliente es un proceso de aplicación este tiene acceso a espacios de trabajo de BD persistentes privadas y en adición puede accesar diversas BD sobre servidores concurrentes con otras aplicaciones de cliente.

Fig. No. 4

Arquitectura de Versant

Para ver las figuras faltantes haga click en el menú superior "Bajar Trabajo"

Impacto de la Orientación a Objetos en la Ingeniería del Software.

En las BDOO, la organización "Gestión Manejadora de Datos Objeto (ODMG)" representa el 100% de las BDOO industriales y ha establecido un estándar de definición (ODL − Lenguaje de Definición de datos) y manipulación (OQL − Lenguaje de consulta) de bases de datos equivalente a SQL.

Respecto a las relacionales, todas (Oracle, Informix, etc.) están añadiendo en mayor o menor grado algunos aspectos de la orientación a objetos. ANSI(Instituto Nacional Estadounidense de Estándar), por su parte, está definiendo un SQL−3 que incorpora muchos aspectos de la orientación a objetos. El futuro del SQL−3 es sin embargo incierto, ya que ODMG ha ofrecido a ANSI su estándar para que sirva de base para un nuevo SQL, con lo que solo habría un único estándar de base de datos.

El grupo ODMG (Grupo Manejador de Datos Objeto) nació de un grupo más grande, llamado "Grupo Manejador de Objetos (OMG)", donde están representados todas las cosas con alguna influencia en el sector. Este grupo esta definiendo un estándar universal por objetos. Este estándar permitirá que un objeto sea programado en cualquier lenguaje y sistema operativo. Esto facilitará enormemente el desarrollo de sistemas abiertos cliente−servidor.

Ventajas en BDOOs

Está su flexibilidad, y soporte para el manejo de tipos de datos complejos. Por ejemplo, en una base de datos convencional, si una empresa adquiere varios clientes por referencia de clientes servicio, pero la base de datos existente, que mantiene la información de clientes y sus compras, no tiene un campo para registrar quién proporcionó la referencia, de qué manera fue dicho contacto, o si debe compensarse con una comisión, sería necesario reestructurar la base de datos para añadir este tipo de modificaciones. Por el contrario, en una BDOO, el usuario puede añadir una "subclase" de la clase de clientes para manejar las modificaciones que representan los clientes por referencia.

La subclase heredará todos los atributos, características de la definición original, además se especializará en especificar los nuevos campos que se requieren así como los métodos para manipular solamente estos campos. Naturalmente se generan los espacios para almacenar la información adicional de los nuevos campos. Esto presenta la ventaja adicional que una BDOO puede ajustarse a usar siempre el espacio de los campos que son necesarios, eliminando espacio desperdiciado en registros con campos que nunca usan.

La segunda ventaja de una BDOO, es que manipula datos complejos en forma rápida y ágilmente. La estructura de la base de datos está dada por referencias (o apuntadores lógicos) entre objetos.

Posibles Desventajas

Al considerar la adopción de la tecnología orientada a objetos, la inmadurez del mercado de BDOO constituye una posible fuente de problemas por lo que debe analizarse con detalle la presencia en el mercado del proveedor para adoptar su producto en una línea de producción sustantiva. Por eso, en este artículo se propone que se explore esta tecnología en un proyecto piloto.

El segundo problema es la falta de estándares en la industria orientada a objetos. Sin embargo, el "Grupo Manejador de Objetos" (OMG), es una organización Internacional de proveedores de sistemas de información y usuarios dedicada a promover estándares para el desarrollo de aplicaciones y sistemas orientados a objetos en ambientes de cómputo en red. La implantación de una nueva tecnología requiere que los usuarios iniciales acepten cierto riesgo. Aquellos que esperan resultados a corto plazo y con un costo reducido quedarán desilusionados. Sin embargo, para aquellos usuarios que planean a un futuro intermedio con una visión tecnológica avanzada, el uso de tecnología avanzada, el uso de tecnología orientada a objetos, paulatinamente compensará todos los riesgos.

Rendimiento

Las BDOO permiten que los objetos hagan referencia directamente a otro mediante apuntadores suaves. Esto hace que las BDOO pasen más rápido del objeto A al objeto B que las BDR, las cuales deben utilizar comandos JOIN para lograr esto. Incluso el JOIN optimizado es más lento que un recorrido de los objetos. Así, incluso sin alguna afinación especial, una BDOO es en general más rápida en esta mecánica de caza−apuntadores.

* Las BDOO hacen que el agrupamiento sea más eficiente. La mayoría de los sistemas de bases de datos permiten que el operador coloque cerca las estructuras relacionadas entre sí, en el espacio de almacenamiento en disco. Esto reduce en forma radical el tiempo de recuperación de los datos relacionados, puesto que todos los datos se leen con una lectura de disco en vez de varias. Sin embargo, en una BDR, los objetos de la implantación se traducen en representaciones tabulares que generalmente se dispersan en varias tablas. Así, en una BDR, estos renglones relacionados deben quedar agrupados, de modo que todo el objeto se pueda recuperar mediante una única lectura del disco. Esto es automático en una BDOO. Además, el agrupamiento de los datos relacionados, como todas las subpartes de un ensamble, puede afectar radicalmente el rendimiento general de una aplicación. Esto es relativamente directo en una BDOO, puesto que representa el

primer nivel de agrupamiento. Por el contrario, el agrupamiento físico es imposible en una BDR, puesto que esto requiere un segundo nivel de agrupamiento: un nivel para agrupar las hileras que representan a los objetos individuales y un segundo para los grupos de hileras que representan a los objetos relacionados.

Características mandatorias ó reglas de oro

Un sistema de BDOO debe satisfacer dos criterios:

* Debe tener un BDMS

* Debe ser un sistema OO

Por ejemplo: para la extensión posible este debe ser consistente en los actuales cortes de lenguajes de programación OO

El primer criterio se traduce en 5 características como son:

Persistencia, Manejador de almacenamiento secundario, Concurrencia, Recuperación, y Facilidad de Query,

La Segunda se traduce en 8 características: Objetos Complejos, Identidad del objeto, Encapsulación, Tipos ó Clases, Sobre paso con combinación retrasada, Extensibilidad y Completación Computacional.

Manifiesto de sistema de gestión de BDOO

Esta publicación intenta definir un sistema de BDOO y describe las principales características.

Hemos separado estas características en 3 grupos:

* Mandatorias.− Son las que el Sistema debe satisfacer a orden de tener un sistema de BDOO y estos son: Objetos complejos, Identidad de objetos, Encapsulación, Tipos ó Clases, Sobre paso combinado con unión retardada, Extensibilidad, Completación Computacional, Persistencia y Manejador de almacenamiento secundario, Concurrencia, Recuperación y Facilidad de Query.

* Opcional.− Son las que pueden ser añadidas para hacer el sistema mejor pero que no son Mandatorias estas son de: herencia múltiple, chequeo de tipos e inferencia distribución y diseño de transacciones y versiones.

* Abiertas.− Son los puntos donde el diseñador puede hacer un número de opciones y estas son el paradigma de la programación la representación del sistema ó el tipo de sistema y su uniformidad. Hemos tomado una posición no muy a la expectativa para tener una palabra final más bien para proveer un punto de orientación para un debate futuro.

Características obligatorias

Este es un punto que no debe faltar en una BD.

* Predominancia combinada con enlace retardado.− Se puede definir que sea Excel, Autocad, etc. desde la programación.

* Extensibilidad.− Proporciona los tipos de datos como: Caracter, booleano, String, etc.

* Concurrencia.− Permite que varios usuarios tengan acceso a una BD al mismo tiempo.

* Recuperación.− Cuando se hace una transacción pero no se puede realizar y se regresa al mismo estado.

* Facilidad de "Consultas a Modo".− Esto es que se tienen diferentes estándares