



Base de datos Orientado a Objetos

(ODBMS, *Object Database Management System*).

Descripción

- En una base de datos orientada a objetos, la información se representa mediante objetos como los presentes en la programación orientada a objetos. Cuando se integra las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos (ODBMS, *Object Database Management System*).
- Un ODBMS hace que los objetos de la base de datos aparezcan como objetos de un lenguaje de programación en uno o más lenguajes de programación a los que dé soporte. Un ODBMS extiende los lenguajes con datos persistentes de forma transparente, control de concurrencia, recuperación de datos, consultas asociativas y otras capacidades.



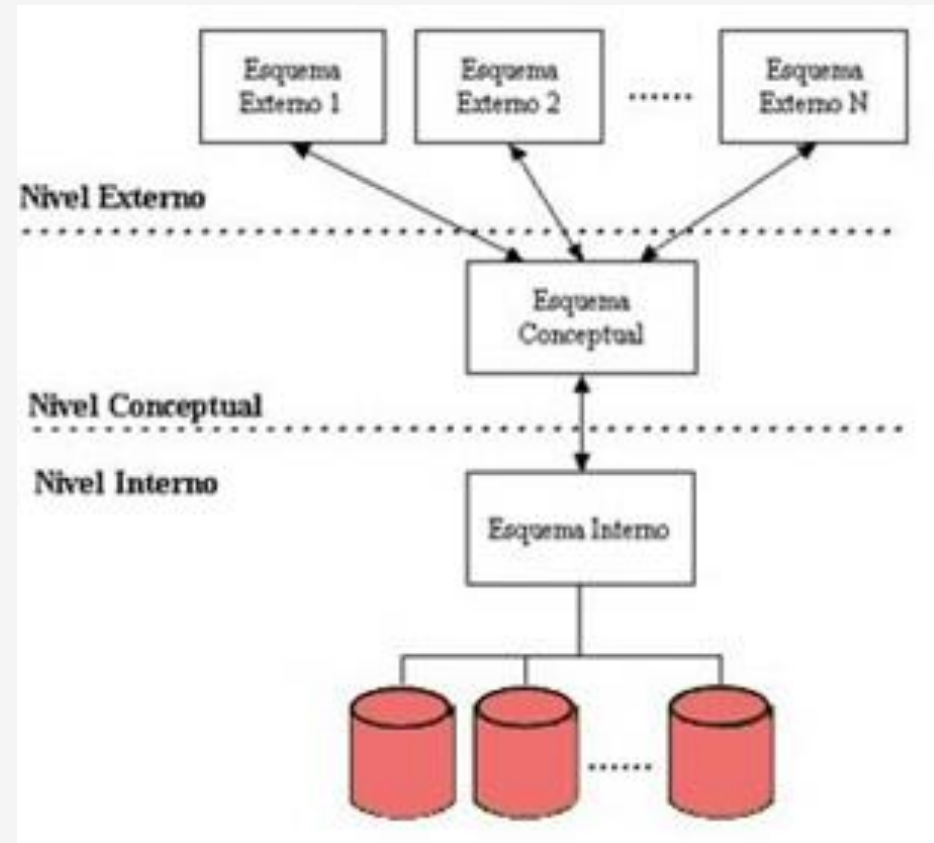
Descripción

- Las bases de datos orientadas a objetos se diseñan para trabajar bien en conjunción con lenguajes de programación orientados a objetos como Java, C#, Visual Basic.NET y C++. Los ODBMS usan exactamente el mismo modelo que estos lenguajes de programación.
- Los ODBMS proporcionan los costes de desarrollo más bajos y el mejor rendimiento cuando se usan objetos gracias a que almacenan objetos en disco y tienen una integración transparente con el programa escrito en un lenguaje de programación orientado a objetos, al almacenar exactamente el modelo de objeto usado a nivel aplicativo, lo que reduce los costes de desarrollo y mantenimiento.



Sistema representativo del modelo

- La arquitectura más representativa basada en el modelo relacional es la ANSI/X3/SPARC, siendo referencia para la mayoría de los SGBD Relacionales comerciales



Representación gráfica de un modelo de datos orientado a objetos

Implementación

Diseño del Sistema

Análisis

Aplicación

Qué

Arquitectura del sistema
Subsistemas

Diseño de Objeto

Estructuras de datos
Algoritmos
Controles

Fases en el ciclo de desarrollo de un sistema orientado a objetos

Representación gráfica de un modelo de datos orientado a objetos

El ciclo de desarrollo de los sistemas orientados a objetos, como se ilustra en la figura anterior, consiste de la representación de objetos desarrollando progresivamente e iterativamente a través de 3 fases. ; análisis, diseño e implementación. En un modelo de desarrollo iterativo, el enfoque cambia desde los aspectos de desarrollo del proceso más abstractos (análisis) a los aspectos más concretos sobre el tiempo de vida de un proyecto. De esta forma, en las etapas tempranas de desarrollo, el modelo que tu desarrollaste es abstracto, enfocándose en las cualidades externas del sistema. Mientras el modelo evoluciona, se convierte en más y mas detallado, el enfoque cambia a como el sistema será construido y cómo debe funcionar. El énfasis en el modelaje debe estar en el análisis y el diseño, enfocándose en temas conceptuales desde el frente hacia el final en lugar de una implementación de atrás al final que innecesariamente restringe las opciones de diseño (Larman, 2004).

En la fase de análisis, tu desarrollas un modelo de una aplicación del mundo real, mostrando sus propiedades importantes. El modelo abstrae conceptos desde el dominio de la aplicación y describe lo que el sistema tiene la intención de hacer en lugar de como será hecho. Especifica el comportamiento funcional del sistema independiente de los asuntos relacionados al medio ambiente en el cuál será finalmente implementado.

Representación gráfica de un modelo de datos orientado a objetos

Usted necesita dedicar suficiente tiempo a claramente entender los requerimientos del problema, mientras se recuerda que en los modelos de desarrollo iterativos, las actividades de análisis serán revisadas múltiples veces durante un proyecto de desarrollo de tal forma que usted pueda aplicar las lecciones aprendidas desde las etapas tempranas de diseño y actividades de implementación a el análisis. Por favor note que durante las actividades de análisis, su enfoque debe estar en analizar y modelar el dominio de interés del mundo real. y no las características internas del sistema de software. En la fase de diseño orientado a objetos, usted define como el modelo de análisis enfocó el mundo real y se dará cuenta en el ambiente de implementación. Por lo tanto su enfoque se moverá a modelar el software del sistema, el cuál será muy fuertemente informado por los modelos que usted creó durante las actividades de análisis. Jacobson y otros (1992) citan tres razones para usar el diseño orientado a objetos :

- 1) El modelo de análisis no es lo suficientemente formal para ser implementado directamente en un lenguaje de programación. Moviendo impecablemente dentro del código fuente requiere refinar los objetos tomando decisiones acerca de qué operaciones un objeto proveerá, qué comunicación entre objetos deben parecerse, qué mensajes tienen que ser pasados, etc .

Representación gráfica de un modelo de datos orientado a objetos

2) El sistema debe adaptarse a el ambiente en cual el sistema será implementado. Para lograr eso,

El modelo de análisis tiene que ser transformado en un modelo de diseño, considerando diferentes factores tales como requerimientos de ejecución, requerimientos en tiempo real y concurrencia, el hardware objetivo y los sistemas de software, las DBMS y lenguaje de programación a ser adoptado y así sucesivamente.

3) Los resultados del análisis pueden ser validados usando un diseño orientado por objetos. En esta etapa, usted puede verificar si los resultados del análisis son apropiados para construir el sistema y hacer cualquier cambio necesario al modelo de análisis durante la siguiente iteración del ciclo de desarrollo.

Para desarrollar un modelo de diseño, usted tiene que identificar e investigar las consecuencias que la implementación del medio ambiente tendrá sobre el diseño. Todas las decisiones de diseño estratégicas, tales como las DBMS a ser incorporadas, cómo el proceso de comunicaciones y el manejo de errores se considera que sean alcanzados, qué componente de bibliotecas serán reusadas y así sucesivamente son hechas. A continuación, usted incorpora tales decisiones dentro de un modelo de

Representación gráfica de un modelo de datos orientado a objetos

diseño de primer corte que se adapte al ambiente de implementación. Finalmente, usted formaliza el modelo de diseño para describir cómo los objetos interactúan entre ellos para cada escenario concebible. Dentro de cada iteración, las actividades de diseño son seguidas por las actividades de implementación (es decir, implementación de diseño usando un lenguaje de programación y/o un sistema administrador de bases de datos). Si el diseño fue hecho bien, trasladarlo a un código de programa es un proceso relativamente sencillo, dado que el modelo de diseño ya incorpora los matices del lenguaje de programación y las DBMS.

Coad y Yourdon (1991) identifican varias motivaciones y beneficios de un modelo orientado a objetos :

- La habilidad para abordar dominios de problemas más retadores
- Comunicación mejorada entre usuarios, analistas, diseñadores y programadores
- Consistencia incrementada entre analistas, diseñadores y actividades de programación
- Representación explícita entre el común de la gente entre componentes de sistema
- Robustez del sistema
- Reusabilidad de análisis, diseño y resultados de programación
- Consistencia incrementada entre todos los modelos desarrollados durante el análisis, diseño y programación orientada a objetos.

Representación gráfica de un modelo de datos orientado a objetos

Nombre de los elementos de modelo de datos

- Tablas

Visor de datos - Employee (Dibujo1.dwg)

Emp_Id	Last_Name	First_Name	Gender	Title
1000	Torbati	Yolanda	F	Programmer
1001	Kleinn	Joel	M	Programmer
1002	Ginsburg	Laura	F	President
1003	Cox	Jennifer	F	Programmer
1005	Zlada	Maui	M	Product Designer
1006	Keyser	Cara	F	Account Executive
1010	Smith	Roxie	M	Programmer
1011	Nelson	Robert	M	Programmer
1012	Sachsen	Lars	M	Support Technician
1013	Shannon	Don	M	Product Designer

- Campos

Ver lista Ver simple Agregar entrada Plantillas Campos

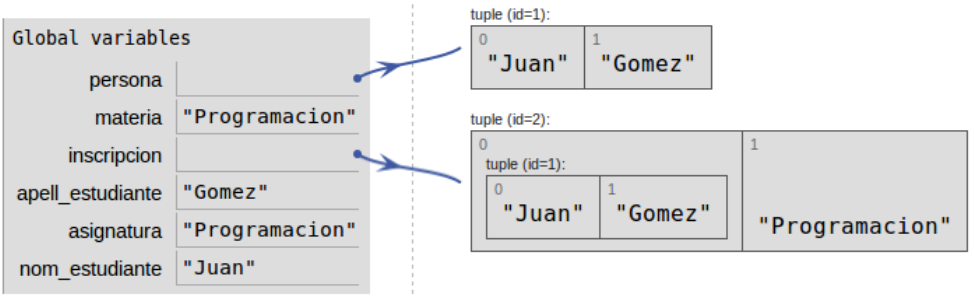
No hay campos definidos para esta base de datos. Por favor, agregue algunos

Crear un nuevo campo: Escoger...

Campo de ordenación por defecto: Escoger...

- Escoger...
- Archivo
- Botones de radio
- Caja de selección
- Dibujo
- Fecha
- Menú
- Menú (selección múltiple)
- Número
- Texto
- Url
- Área de texto

- Tuplas



- Registros

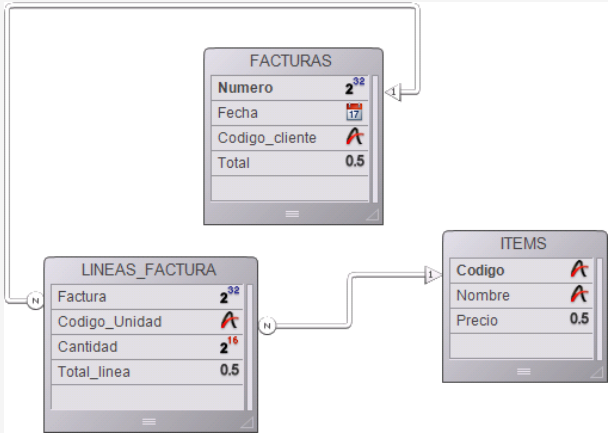
Campos

Nombre Proyecto	Contacto	Correo	Teléfonos
Eagle	anell y adrian	eagleog@cantv.net	0212 2349902 - 4162900
TLC	Ambar Teixeira	boutiquepoventas2003@gmail.com	945 07 65
Tedexis	Luis Poggi	lpoggi@gmail.com	4141039524
FYC	Carlos Marrero	carlos.marrero@ryccorp.com	2327911
NGS camcaroni	Alfonso Mora	amora@ngs.com.ve	4148095259
Tecnoquim	Lenin Marques	grupotecnoquim@gmail.com	7816267, 7629963, 0416 806 0708
Tuscomprasporinternet	Rafael Monserrat	bufetemontserrat@hotmail.com	4147999904

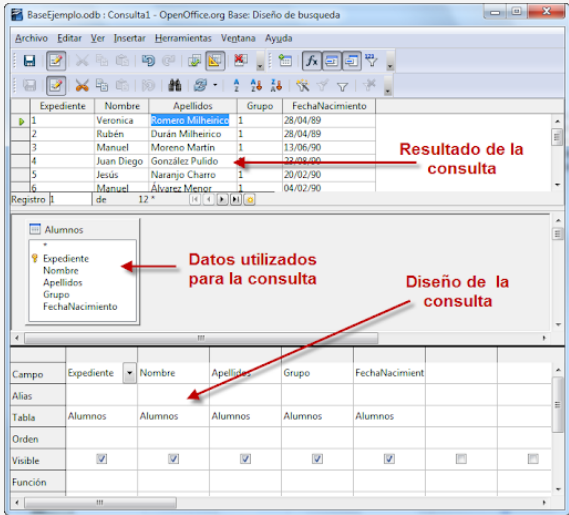
Registros

Nombre de los elementos de modelo de datos

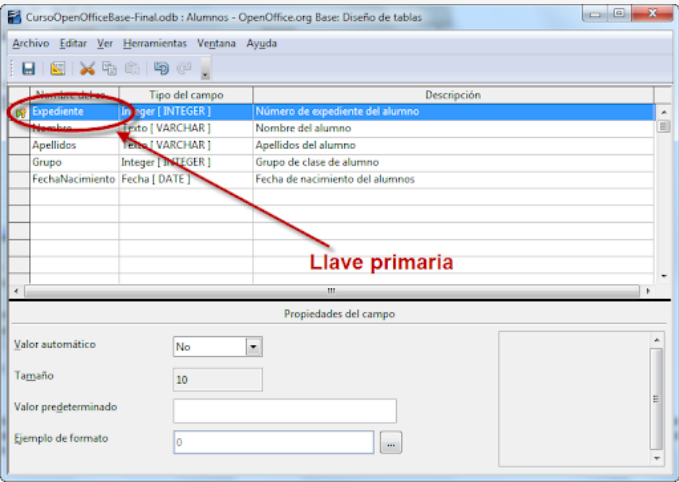
•Relaciones



•Consulta



•Claves



•Formularios



Nombre de los elementos de modelo de datos

- Informes

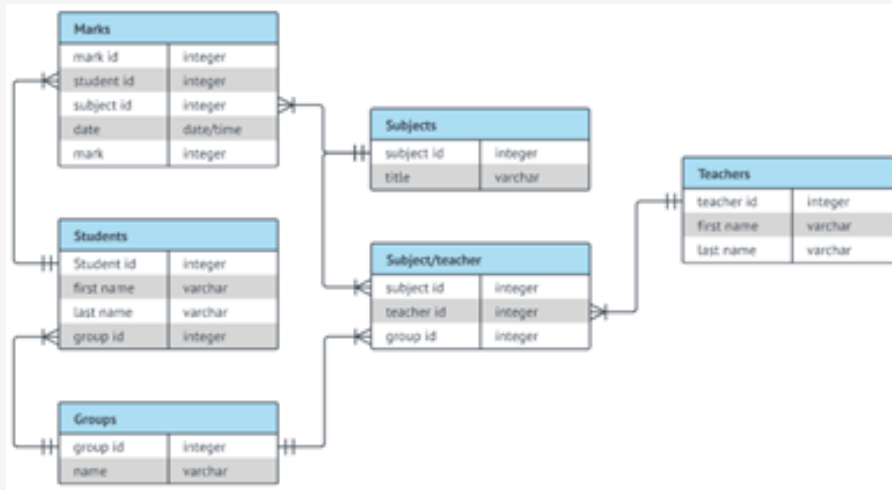
ALUMNOS
30 de marzo de 2011

Grupo	Expediente	Nombre	Apellidos	FechaNacimiento
0	1	Veronica	Romero Milhenico	25/04/89

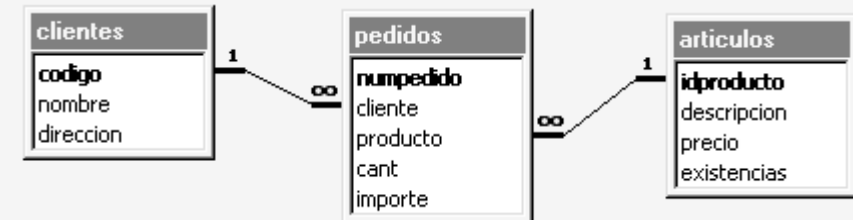
Grupo	Expediente	Nombre	Apellidos	FechaNacimiento
1	7	Cristian	Aguiar Caro	14/11/89
	6	Manuel	Álvarez Meror	04/02/90
	8	Alejandro	Antonio González	15/03/90
	9	Maria	Caballo Santos	18/01/90
	10	Gerardo	Correa Morán	04/05/91
	11	Sheila	Cosme Almeida	29/08/92
	12	Rubén	Crespo Bonilla	04/08/90
	2	Rubén	Durán Milhenico	28/04/89
	13	Maria Dolores	Durán Rasero	22/08/89
	14	Sheila	Eduardo Sánchez	04/07/87
	15	Tamara	Florez Hernández	08/02/88
	16	José Daniel	García López	10/04/89

Página 1/9

- Modelo entidad relación



- Integridad referencial



Independencia de datos

- La capacidad de modificar el esquema físico sin cambiar el esquema de Bases de Datos lógico
 - Las aplicaciones dependen del esquema lógico
 - En general, las interfaces entre los diferentes niveles y componentes deben definirse adecuadamente de modo que los cambios en algunas partes no influyeran seriamente



Beneficios de uso de los DBMS

- Independencia de datos y acceso eficaz
 - Reduce la redundancia de información
 - 3 Niveles de abstracción

Las vistas describen como ven los usuarios los datos

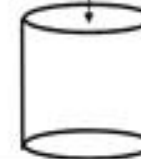
Vista 1 Vista 2 Vista 3

Esquema Conceptual

El esquema conceptual define la estructura lógica

El esquema físico describe los archivos, estructuras e índices usados

Esquema Físico



Seguridad en los modelos para bases de datos orientadas a objetos

Los lenguajes de programación orientados a objetos (OO), los análisis OO y las técnicas de diseño influyen el diseño y desarrollo de las bases de datos.

Muchos de los proveedores de bases de datos están incorporando conceptos orientados a objetos dentro de sus productos en un esfuerzo por facilitar el diseño y desarrollo de las bases de datos.

La incorporación de principios orientados a objetos, ofrece nuevas herramientas para asegurar la información almacenada en las bases de datos.

Este artículo explica los conceptos básicos en seguridad de bases de datos y las diferencias entre seguridad relacional y los sistemas orientados a objetos y algunos temas específicos relacionados a la seguridad de la siguiente generación de DBMS orientadas a objetos (OODBMS).

Conceptos básicos de seguridad en los datos

La seguridad en las bases de datos está relacionada con la secrecía de los datos. La Secrecía de los

Seguridad en los modelos para bases de datos orientadas a objetos

datos significa proteger una base de datos de accesos no autorizados por los usuarios y las aplicaciones de software.

La secrecía, en el contexto de seguridad en las bases de datos, incluye una variedad de amenazas incurridas a través de accesos no autorizados. Estas amenazas varían de robo intencional a la destrucción de datos hasta la adquisición de información a través de medidas más sutiles tal como la deducción. Hay 3 categorías generalmente aceptadas de problemas relacionados con la secrecía :

- **La liberación de información inadecuada desde la lectura de datos que fue intencionalmente o accidentalmente accesada por usuarios no autorizados.** La seguridad de las bases de datos de accesos no autorizados es más difícil que controlar el acceso a archivos administrados por los sistemas operativos. Este problema emerge de la granularidad más fina que se usa en las bases de datos cuando se manejan archivos, atributos y valores. Este tipo de problema también incluyen las violaciones a la secrecía que resultan del problema de inferencia, la cual es la deducción de información no autorizada desde la observación de información autorizada. La inferencia es uno de los factores más difíciles de controlar en cualquier intento de asegurar datos.

Seguridad en los modelos para bases de datos orientadas a objetos

Ya que la información en una base de datos está semánticamente relacionada, es posible determinar el valor de un atributo sin accederlo directamente. Los problemas de inferencia son más serios en bases de datos estadísticas, en donde los usuarios pueden localizar información en entidades individuales desde datos estadísticos agregados.

- **La inadecuada modificación de los datos** : Esta amenaza incluye violaciones de la seguridad de los datos a través del mal manejo o modificaciones por usuarios no autorizados. Estas violaciones pueden resultar de errores, virus, sabotajes o fallas en los datos que emergen de accesos no autorizados por los usuarios.
- **Amenazas de negación del servicio** : Las acciones que podrían prevenir a los usuarios de usar recursos del sistema o acceder datos están entre las más serias. Esta amenaza ha sido demostrada a un grado significativo recientemente con los “ Ataques de Inundaciones SYN “ contra proveedores de servicio de redes. La inundación SYN, es un ataque de protocolo. Una inundación SYN es una forma de ataque de negación del servicio, en el que un atacante envía una sucesión de solicitudes SYN al sistema de un objetivo en un intento de consumir suficientes recursos del servidor para que

Seguridad en los modelos para bases de datos orientadas a objetos

el sistema no responda al tráfico legítimo.

Políticas de control de acceso obligatorio vs discrecional : Los modelos de seguridad de las RDBMS (sistemas manejadores de bases de datos relacionales) y los modelos de bases de datos orientadas por objetos hacen uso de 2 tipos generales de políticas de control de acceso para proteger la información en sistemas de multinivel.

La primera de estas políticas es la discrecional (DAC). En la política discrecional, el acceso es restringido basado en las autorizaciones expedidas por el propietario.

La política de control de acceso obligatorio (MAC) asegura la información asignando niveles de sensibilidad, o etiquetas, a las entidades de datos.

Las políticas MAC son más seguras que las políticas DAC y ellas son usadas en sistemas en los cuales la seguridad es crítica, tales como aplicaciones militares. Sin embargo, el precio a pagar por esta seguridad reforzada es la reducción del desempeño del sistema de administración de bases de datos.

Seguridad en los modelos para bases de datos orientadas a objetos

Características de seguridad de los BDMS orientados a objetos : A diferencia de las bases de datos relacionales la seguridad de las DBMS orientadas a objetos (OODBMS) tienen ciertas características que las hacen únicas. Además, sólo un número limitado de modelos de seguridad han sido diseñados específicamente para bases de datos. Los modelos de seguridad propuestos hacen uso de los conceptos de encapsulado, herencia, información escondida, métodos y la habilidad a el modelo de las entidades del mundo real que están presentes en los ambientes orientados a objetos.

El modelo de base de datos orientado a objetos también permite la clasificación de una sensibilidad del objeto a través del uso de clase (de entidades) y petición o instancia. Cuando una instancia de una clase es creada, el objeto puede automáticamente heredar el nivel de sensibilidad de la superclase. Aunque la habilidad para pasar clasificaciones a través de la herencia es posible en OODBMS. La clase de instancias son usualmente clasificadas al nivel más alto dentro de la jerarquía de clase del objeto. Esto previene un problema de control de flujo, en donde la información pasa desde los niveles de clasificación más altos a los más bajos. Las OODBMS también usan características únicas que permiten a estos modelos controlar el acceso a los datos en la base de datos. Ellos incorporan características tales como estructura de datos flexible, herencia y ataduras tardías. El control de acceso de los modelos OODBMS tiene que ser consistente con tales características.

Seguridad en los modelos para bases de datos orientadas a objetos

Los usuarios pueden definir métodos, algunos de los cuales son abiertos para otros usuarios como métodos públicos. Además, las OODBMS pueden encapsular una serie de comandos de acceso básicos dentro de un método y hacerlo público para los usuarios mientras los comandos básicos se mantienen lejos de los usuarios.

Navegación de la información

- Como un lenguaje independiente OQL permite consultar objetos iniciando por sus nombres, los cuales actúan como puntos de acceso a la base de datos, un nombre debe denotar de que tipo de objeto se trata.
- Como un Lenguaje embebido OQL permite consultar objetos que tengan soporte por las expresiones a través del lenguaje nativo, una consulta OQL es una función que retorna un objeto el cual corresponde al tipo solicitado a través de la consulta.

Navegación de la información

- Algunas Características:

- OQL Es muy parecido a SQL-92. Demanda conocimiento de modelo orientado a objetos
- OQL proporciona un lenguaje de alto nivel que permite trabajar con conjuntos de objetos, pero no es restrictivo, también puede ser utilizado para trabajar con estructuras como listas o arreglos con la misma eficiencia.
- OQL Es un lenguaje de fácil uso.
- OQL puede ser invocado desde el interior del un lenguaje de programación, desde un lenguaje para el que está definido un ODMG
- OQL puede invocar operaciones programadas in aquellos lenguajes.
- OQL provee acceso declarativo a los objetos.
- Las consultas OQL pueden ser fácilmente optimizadas dado su naturaleza declarativa

Navegación de la información

- Algunas Reglas:
 - Todas las instrucciones deben terminar con Punto y coma.
 - Una lista de instrucciones usualmente es separada por comas pero no debe terminarse con punto y coma.
 - Las cadenas de texto deben ser escritas entre comillas

Navegación de la información

- Algunas Reglas:
 - Todas las instrucciones deben terminar con Punto y coma.
 - Una lista de instrucciones usualmente es separada por comas pero no debe terminarse con punto y coma.
 - Las cadenas de texto deben ser escritas entre comillas

Navegación de la información

Ejemplo :

Solicitar la velocidad del CPU con mas de 64 MB de RAM desde una base de datos de prueba

```
SELECT pc.cpuspeed  
FROM PCs pc  
WHERE pc.ram > 64;
```

Navegación de la información

Ejemplo :

El siguiente ejemplo ilustra como la información solicitada se agrupa por Manufacturera.

```
SELECT manufacturer, AVG(SELECT part.pc.ram FROM partition part)
FROM PCs pc
GROUP BY manufacturer: pc.manufacturer;
```

Navegación de la información

Ejemplo :

Esta consulta selecciona un conjunto de edades para todas las personas llamadas “Pat”, esta consulta recibirá valores enteros.

```
select distinct x.age  
from Persons x  
where x.name = "Pat"
```

Navegación de la información

Ejemplo :

Esta consulta hace prácticamente lo mismo que la anterior pero en este caso el resultado será una estructura que contendrá la edad y el sexo de las personas llamadas "Pat"

```
select distinct struct(a: x.age, s: x.sex)
from Persons x
where x.name = "Pat"
```

Referencias:

The Object Data Estándar ODMG 3.0 , Morgan Kauffman Plublishers 1° Edición 2000

Artículo : Security Models for Object-Oriented Data Bases ; Autor : James Cannady ; 2005

Introduction to Database Management System ; Autor : Gupta, Satinder Bal, Mittal, Aditya
2017

Referencias

-Bases de datos

https://personales.unican.es/corcuerp/BD/Slides/BD_intro.pdf

-Modelo entidad relación

<https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion#:~:text=Un%20diagrama%20entidad%2Drelaci%C3%B3n%2C%20tambi%C3%A9n,s%C3%AD%20dentro%20de%20un%20sistema.>

-Tecnólogo el administración documental

<http://yulipane.blogspot.com/2010/09/elementos-de-una-base-de-datos.html>

-Sistemas de gestión de bases de datos

<https://sites.google.com/site/basededatosjosemata/sistemas-de-gestion-de-bases-de-datos>

The Object Data Estándar ODMG 3.0 , Morgan Kauffman Plublishers 1º Edición 2000

Artículo : Security Models for Object-Oriented Data Bases ; Autor : James Cannady ; 2005

Introduction to Database Management System ; Autor : Gupta, Satinder Bal, Mittal, Aditya
2017