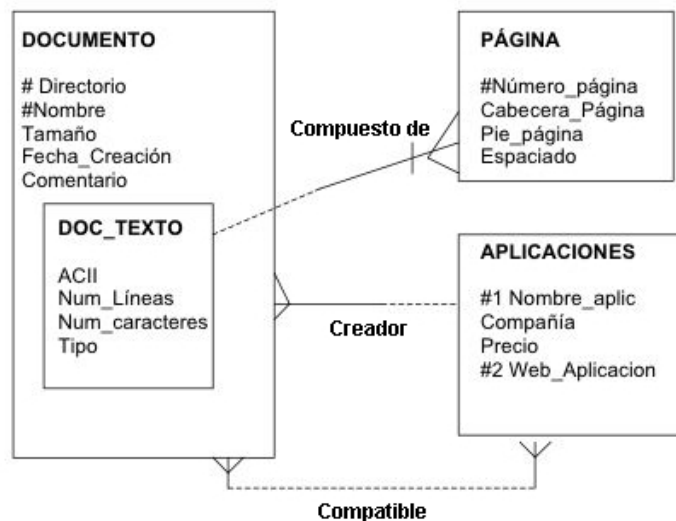




Introducción a la unidad

La tecnología de bases de datos vive un momento de lenta transición del modelo relacional a otros novedosos modelos con aplicaciones interesantes. Entre estos modelos están el multidimensional para sistemas OLAP, el semiestructurado para bases de datos XML de intercambio electrónico de información y el modelo dimensional para creación de *data warehouses*. Un modelo que llama mucho la atención por sus ventajas tecnológicas es el orientado a objetos y será producto de la exposición en esta unidad.



La orientación a objetos llegó para quedarse y desde los años 80 se convierte día a día en la tecnología de desarrollo de aplicaciones por excelencia. Este hecho bastaría para entender el surgimiento de los sistemas manejadores de bases de datos orientados a objetos (OODBMS), pero más adelante revisaremos algunas motivaciones adicionales para el origen de esta tecnología.



Unidad III. Modelo Orientado a Objetos



Algo importante que debemos resaltar es que a pesar de las ventajas para el desarrollo de aplicaciones empresariales que tienen los OODBMS, hoy en día las empresas siguen utilizando los manejadores de bases de datos relacionales y no se ve aún para cuándo serán suplantadas por completo.

En los apartados siguientes revisaremos las situaciones tecnológicas que dieron origen a los manejadores de bases de datos orientados a objetos. Repasaremos los fundamentos de la orientación a objetos. Detallaremos la evolución de este tipo de sistemas de bases de datos. Finalmente te proporcionaremos una definición y expondremos sus principales características.

Objetivo particular de la unidad

Distinguir un manejador de bases de datos orientado a objetos de uno relacional; enumerar las razones por las que surgieron estos manejadores así como identificar las generaciones de manejadores orientados a objetos y sus principales características.



Unidad III. Modelo Orientado a Objetos



Lo que sé:

Cree la Base de Datos “Libro”, empleando comandos de SQL o con el manejador de Bases de Datos de Access, con las siguientes tablas:

AUTOR			
ID_AUTOR	NOMB_AUTOR	NACIONALIDAD	ID_LIBRO
LIBRO			
ID_LIBRO	TITULO_LIBRO	ISBN	ID_AUTOR
EDITORIAL			
ID_EDIT	NOMB_EDIT	PAIS_EDIT	ID_LIBRO
CLIENTE			
ID_CTE	NOMB_CTE	RFC_CTE	ID_EDIT

Realiza tu actividad en un procesador de textos, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma.



Unidad III. Modelo Orientado a Objetos



Temas de la unidad III

1. Antecedentes

- 1.1 Retos actuales de los sistemas manejadores de bases de datos
- 1.2 Tendencias actuales en la tecnología de bases de datos
- 1.3 Orientación a objetos
- 1.4 Persistencia

2. Sistemas de administración de bases de datos orientadas a objetos

- 2.1 Historia
- 2.2 Primera generación
- 2.3 Segunda generación
- 2.4 Tercera generación
- 2.5 Definición
- 2.6 Características

3. Estándar ODMG

Resumen de la unidad

Los DBMS surgieron para responder a las necesidades de información de las organizaciones. Se trataron de un conjunto de datos persistentes y de programas para acceder a ellos y actualizarlos.

Surgieron diversos sistemas administradores de archivos hasta llegar a la generación de las bases de datos relacionales que utilizaban lenguajes más accesibles y poderosos en la manipulación de datos como el SQL, QUEL y QBE. Estos deben contar con un modelo de datos, es decir, estructuras lógicas para describir los datos, y operaciones para manipularlos (recuperación y actualización).



Unidad III. Modelo Orientado a Objetos



Los nuevos DBMS deben tomar en cuenta varias operaciones para llevar con eficiencia la Administración, Validación, Creación y Eliminación de las Bases de Datos y poseer características de Orientación a Objetos y Persistencia.

Existen organismos en pro de la estandarización de este tipo de sistemas manejadores de bases de datos como el OMG (Object Management Group), la CAD Framework Initiative y el grupo de trabajo de ANSI. Las características de los OODBMS que se estudiaron fueron son: objetos complejos, identidad de objetos, encapsulamiento, herencia, sobrescritura, sobrecarga y completa capacidad computacional.



Unidad III. Modelo Orientado a Objetos



Tema 1. Antecedentes

Objetivo del tema

Reconocer la tecnología aplicada a los sistemas de administración de Bases de Datos y las características de los nuevos modelos.

Desarrollo

Los DBMS (sistemas manejadores de bases de datos) surgieron para responder a las necesidades de manejo de información de las organizaciones en grandes volúmenes. Se trata de un conjunto de datos persistentes y de programas para acceder a ellos y actualizarlos.

La tecnología de bases de datos tiene más de treinta años. Primero surgieron sistemas administradores de archivos de tipo ISAM (*Indexed Secuencial Access Mode*), que trabajaban con archivos separados. Después vinieron sistemas que centralizaban los archivos en una colección llamada base de datos, quienes utilizaron el modelo jerárquico (sistemas IMS y 2000). Posteriormente surgieron los desarrollados por la CODASYL (Conference on Data Systems Languages), como IDS, TOTAL, ADABAS e IDMC. La siguiente generación fue la de bases de datos relacionales. Éstas utilizaron lenguajes más accesibles y poderosos en la manipulación de datos como el SQL, QUEL y QBE.

Los DBMS cuentan con un modelo de datos, es decir, estructuras lógicas para describir los datos y operaciones para manipularlos (recuperación y actualización). Las operaciones sobre los datos se hacen por medio de **tres lenguajes**: un **DDL** (Lenguaje de definición de datos, Data Definition Language por sus siglas en inglés) para definir el esquema y la integridad, un **DML** (Lenguaje de Manipulación de Datos, Data Manipulation Language) para la actualización de los datos y un **DCL**



Unidad III. Modelo Orientado a Objetos



(Lenguaje de Control de Datos, DCL por sus siglas en inglés: Data Control Language) para el manejo de las autorizaciones en la base de datos.

Adicionalmente un DBMS incluye mecanismos de seguridad, acceso a los datos, recuperación, control de concurrencia y optimización de consultas.

Los DBMS evolucionan con el afán de satisfacer nuevos requerimientos tecnológicos y de información. En seguida describiremos lo que ha motivado el surgimiento de sistemas orientados a objetos.

1. Retos actuales de los sistemas manejadores de bases de datos

Aunque los DBMS relacionales (RDBMS) son actualmente líderes del mercado y brindan las soluciones necesarias a las empresas comerciales, existen aplicaciones que necesitan funciones con las que no cuentan. Ejemplos de ellas son las CAD/CAM y CASE. Adicionalmente, los sistemas multimedia, como los geográficos y de medio ambiente, sistemas de gestión de imágenes y documentos, y los sistemas de apoyo a las decisiones necesitan de modelos de datos complejos difíciles de representar como tuplas de una tabla.

En general, estas aplicaciones necesitan manipular objetos y los modelos de datos deben permitirles expresar su comportamiento y las relaciones entre ellos. Los nuevos DBMS deben tomar en cuenta las siguientes operaciones:

- ❖ Ser capaces de definir sus propios tipos de datos.
- ❖ Manejar versiones de objetos y estados de evolución.
- ❖ El tamaño de los datos puede ser muy grande.
- ❖ La duración de las transacciones puede ser muy larga.
- ❖ Recuperar rápidamente objetos complejos.



Unidad III. Modelo Orientado a Objetos



- ❖ Ofrecer comunicación efectiva a los clientes del sistema, principalmente en desarrollos grupales.
- ❖ Permitir cambios en el esquema de la base.
- ❖ Manejar objetos completos y sus componentes.
- ❖ Lenguajes de consulta de objetos y lenguajes computacionalmente complejos.
- ❖ Mecanismos de seguridad basados en la noción de objeto.
- ❖ Funciones para definir reglas deductivas y de integridad.
- ❖ Tener la capacidad para comunicarse con las aplicaciones ya existentes y manipular sus datos.

2. Tendencias actuales en la tecnología de bases de datos

Con miras a superar los retos antes mencionados, las bases de datos están tomando varias tendencias. En general se están auxiliando de los lenguajes de programación orientados a objetos, los lenguajes lógicos y la inteligencia artificial. En este sentido, podemos determinar cuatro tendencias actuales:

1	Sistemas relacionales extendidos. Incluyen manejo de objetos y <i>triggers</i> .	2	Sistemas de bases de datos orientadas a objetos. Integran el paradigma de la orientación a objetos a la tecnología de bases de datos.
3	Sistemas de bases de datos deductivas. Unen las bases de datos con la programación lógica. Cuentan con mecanismos de inferencia, basados en reglas, para generar información adicional a partir de los datos almacenados en la base.	4	Sistemas de bases de datos inteligentes. Incorporan técnicas desarrolladas en el campo de la inteligencia artificial.



Unidad III. Modelo Orientado a Objetos



3. Orientación a objetos

La orientación a objetos representa el mundo real y resuelve problemas a través de objetos, ya sean tangibles o digitales. Este paradigma tecnológico considera un sistema como una entidad dinámica formada de componentes. Un sistema sólo se define por sus componentes y la manera en que éstos interactúan. Las principales características de la orientación a objetos son:

- ❖ Es una tecnología para producir modelos que reflejen un dominio de negocio y utiliza la terminología propia de tal dominio.
- ❖ Cuenta con cinco conceptos subyacentes: objeto, mensajes, clases, herencia y polimorfismo.
- ❖ Un objeto tiene un estado, un comportamiento y una identidad.
- ❖ Los mensajes brindan comunicación entre los objetos.
- ❖ Las clases son un tipo de plantilla usada para definir objetos, los cuales son instancias del mundo real.
- ❖ Cada objeto tiene un nombre, atributos y operaciones.

4. Persistencia

La persistencia es una característica necesaria de los datos en un sistema de bases de datos. Recordemos que consiste en la posibilidad de recuperar datos en el futuro. Esto implica que los datos se almacenan a pesar del término del programa de aplicación. En resumen, todo manejador de base de datos brinda persistencia a sus datos.



Unidad III. Modelo Orientado a Objetos



En el caso de los OODBMS, la persistencia implica almacenar los valores de atributos de un objeto con la transparencia necesaria para que el desarrollador de aplicaciones no tenga que implementar ningún mecanismo distinto al mismo lenguaje de programación orientado a objetos.

Lo anterior traería como ventaja que no sería necesario el uso de dos lenguajes de programación para construir una aplicación. Es decir, actualmente, el desarrollo de aplicaciones se hace con lenguajes de programación orientada a objetos almacenando datos en bases relacionales, por lo que el desarrollador debe utilizar un lenguaje para la aplicación (Java, PHP, C++) y otro para la base de datos (SQL).

El objetivo básico de un OODBMS es entonces darle persistencia a los objetos. Por lo anterior algunos autores ven estos sistemas sólo como lenguajes de orientación a objetos con persistencia y no como manejadores completos. Para ver una discusión acerca de si los OODBMS son en realidad DBMS puedes leer a C. J. Date (2001: 845-847).



ACTIVIDAD 1

Investiga las características de los Administradores de Bases de Datos:

- a) ISAM
- b) RDBMS
- c) OODBMS

Realiza tu actividad en un procesador de textos, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma.

ACTIVIDAD 2

Contesta el siguiente cuestionario.

1. Enuncia las Tecnologías de Administración de Bases de Datos.
2. ¿Cuáles fueron las necesidades tecnológicas que dieron pie al surgimiento de los sistemas de bases de datos orientados a objetos?
3. Los nuevos DBMS contemplan ciertas operaciones, menciónalas.
4. Enuncia las cuatro tendencias actuales de las Tecnologías de Bases de Datos.
5. ¿Qué características propone el ODMG para un OODBMS?

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**.



Unidad III. Modelo Orientado a Objetos



Autoevaluación

Selecciona si las aseveraciones son verdaderas (V) o falsas (F). Al concluir tendrás tu calificación de manera inmediata.

	Verdadera	Falsa
1. Los Lenguajes de Programación Orientados a Objetos auxilian a la Tecnología de Bases De Datos.	()	()
2. Los Sistemas Relacionales Extendidos emplean triggers.	()	()
3. Los Sistemas de Bases de Datos Deductivas unen las Bases De Datos con programación funcional.	()	()
4. Los Sistemas de Bases de Datos Inteligentes incorporan mecanismos de Actualización en tiempo real.	()	()
5. Las Tecnologías de Bases de Datos deben tomar en cuenta la persistencia.	()	()

Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción





Unidad III. Modelo Orientado a Objetos



Tema 2. Sistemas de administración de bases de datos orientados a objetos

Objetivo del tema

Identificar las características de los sistemas administradores de bases de datos orientados a objetos y los ejemplos más representativos de estas tecnologías.

Desarrollo

Los sistemas de bases de datos orientados a objetos parecen ser la tecnología más prometedora para los próximos años, aunque carecen de un modelo de datos común y de fundamentos formales, además de que su comportamiento en seguridad y manejo de transacciones no están a la altura de los programas actuales de manejadores de Bases de Datos.

Hay organismos en pro de la estandarización de este tipo de sistemas manejadores de bases de datos, como el OMG (Object Management Group)¹ la CAD Framework Initiative y el grupo de trabajo de ANSI (American National Standards Institute)²

Algo que apoya esta tendencia es que a pesar de que la ingeniería de software orientada a objetos requiere mucho tiempo de análisis, la mayoría de los proyectos de desarrollo son más cortos y requieren menos personas, además de que la cantidad de código es menor.

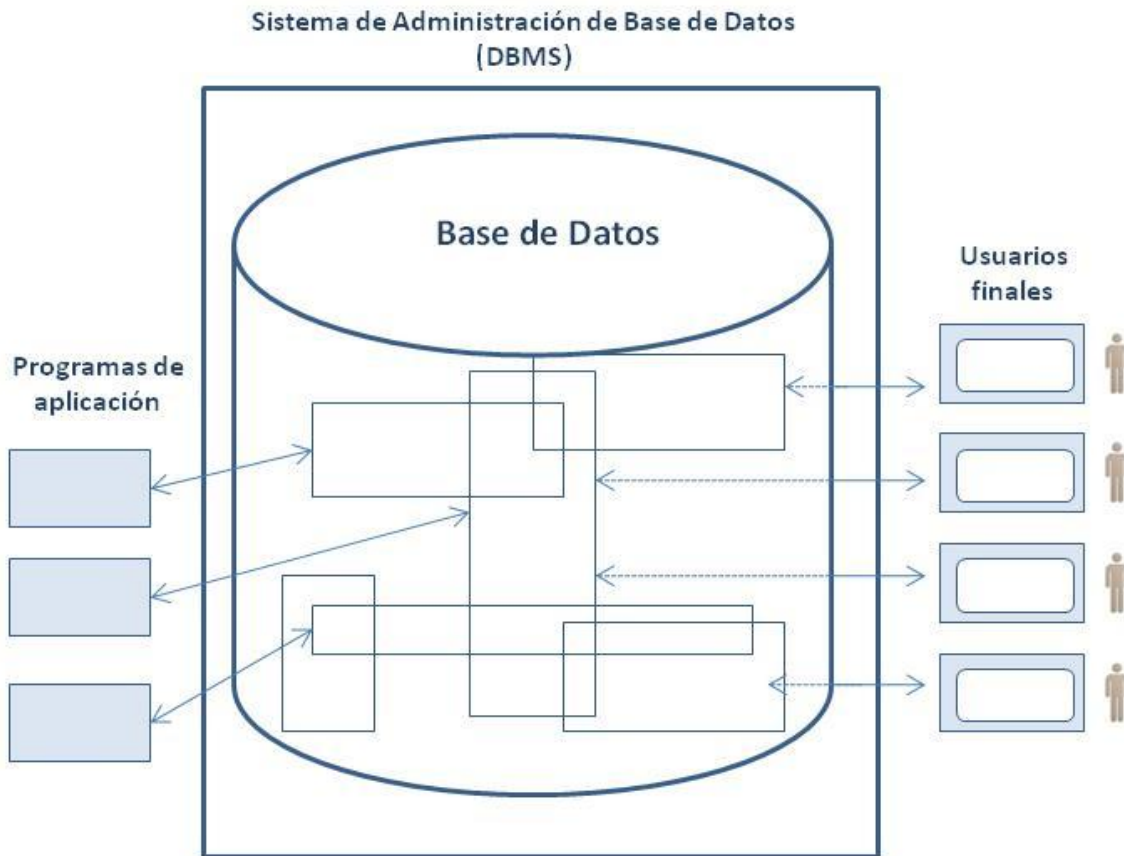
A pesar de lo dicho anteriormente, sería difícil para las empresas dejar de un día para otro los sistemas actuales, debido principalmente, a la falta de personal calificado, al efecto sobre la continuidad de sus operaciones y a la ausencia de garantías en la reutilización de los datos.

¹ <http://www.omg.org/>,

² www.ansi.org.



Unidad III. Modelo Orientado a Objetos



Antecedentes

La evolución de los sistemas de bases de datos orientados a objetos está muy ligada al mercado de manejadores y a las compañías que han apostado por este tipo de bases de datos. Como mencionamos, este tipo de DBMS no tiene estrictos fundamentos teóricos como el modelo relacional y por tanto no se puede establecer una historia de su concepción.

A continuación te describimos brevemente las tres generaciones de OODBMS según Bertino y Martino (1995).



Unidad III. Modelo Orientado a Objetos



1. Primera generación

Comienza en 1986 cuando el sistema G-Base fue lanzado por la compañía francesa Grápale. En 1987 Servio Corp introduce GemStone y en 1988 Ontologic promueve su Vbase, seguido de Statics por la empresa Symbolics. Estos sistemas estaban basados en lenguajes propios y plataformas independientes del mercado. Estos sistemas fueron considerados lenguajes orientados a objetos con persistencia.

2. Segunda generación

Se da con la salida al mercado de Ontos en 1989. Siguieron los productos Object Design, Objectivity y Versant Object Technology. Todos utilizaron una arquitectura cliente/servidor y una plataforma en C++, X Windows y UNIX.

3. Tercera generación

La generación comienza con Itasca, lanzado en agosto de 1990 por Microelectronics and Computer Corporation. Le siguieron O2 producido por la compañía francesa Altair, y después Zeitgeist por Texas Instruments. Estos ya son sistemas administradores de bases de datos con características avanzadas, un DDL y DML orientados a objetos.

Definición

No obstante, en la actualidad hay mucha atención hacia los OODBS, tanto en el terreno de desarrollo como en el teórico, no hay una definición estándar de lo que estos sistemas significan.

Existen tres problemas principales que impiden una definición generalizada:



Unidad III. Modelo Orientado a Objetos



- 1** La falta de un modelo de datos común entre los diferentes sistemas. Los sistemas de bases de datos relacionales cuentan con especificaciones claras dadas por Codd, pero los orientados a objetos no tienen algo así. Se pueden encontrar muchos textos que describen diferentes modelos, pero no hay uno como estándar.
- 2** La carencia de fundamentos formales. El fundamento teórico de la programación orientada a objetos es escaso en comparación con otras áreas como la programación lógica. Además se carece de definiciones de diversos conceptos.
- 3** Una actividad experimental muy fuerte. Existe mucho trabajo experimental, la mayoría de los desarrollos son sistemas prototipo no comerciales, no hay trabajo de conceptualización y definición de estándares. El diseño de estos sistemas está orientado por las aplicaciones que los requieren y no por un modelo común

El problema de estos sistemas es similar al de las bases de datos relacionales a mitad de los setenta. La gente se dedicaba a desarrollar implementaciones en lugar de definir las especificaciones para luego hacer la tecnología que permitiera implementarlas.

Se espera que de los prototipos y desarrollos actuales de los OODBs surja un modelo. Aunque también se corre el riesgo de que alguno de estos se convierta en el estándar por su demanda en el mercado.



Unidad III. Modelo Orientado a Objetos



A manera de definición podemos decir que un OODBMS debe satisfacer dos criterios:

Debe ser un DBMS
El primer criterio incluye características de cualquier DBMS, que podemos listar como: persistencia, administración de almacenamiento secundario, concurrencia, recuperación y facilidad de consultas personalizadas.
Debe ser un sistema orientado a objetos (consistente con los lenguajes de programación orientada a objetos)
El segundo criterio corresponde a características que se comparten con la programación orientada a objetos: objetos complejos, identidad de objetos, encapsulación, herencia, sobreescritura y sobrecarga, y completa capacidad computacional (<i>computational completeness</i>).

Características

A continuación exponemos las características de un OODBMS.

1. Objetos complejos

Los objetos complejos son creados a partir de objetos (tipos de datos) simples. Estos objetos simples son: enteros, caracteres, cadenas de bytes, booleans y números de punto flotante. Los objetos complejos pueden ser por ejemplo: tuplas, conjuntos (sets), listas y arreglos. Un OODBMS debe tener como mínimo conjuntos (set), listas y tuplas.

Los conjuntos (sets) son la manera natural de representar colecciones del mundo real. Las tuplas permiten representar de manera natural las propiedades de una entidad y son importantes por la aceptación ganada con el modelo relacional.



Unidad III. Modelo Orientado a Objetos



Finalmente, las listas o arreglos resultan importantes porque capturan orden, cosa que ocurre en el mundo real; además de que ayudan a representar matrices y series de datos en el tiempo.

2. Identidad de objetos

La identidad de objetos ha existido desde hace mucho tiempo en los lenguajes de programación, pero en las bases de datos es más reciente. El objetivo es contar con objetos que tengan una existencia independiente de sus valores. Así, dos objetos pueden ser idénticos si son el mismo objeto o pueden ser iguales si tienen los mismos valores.

La identidad de objetos cobra relevancia cuando un objeto se comparte con otros y cuando se actualiza. En un modelo basado en identidad, dos objetos pueden compartir un objeto hijo. Por ejemplo, pensemos en dos personas, Arturo y Valeria, y cada uno tiene un hijo llamado Jaime. En la vida real hay dos posibles situaciones:

- ♦ Arturo y Valeria son padres de Jaime, por lo que existe identidad del objeto Jaime, es decir, se trata en realidad del mismo objeto.
- ♦ Hay dos niños del mismo nombre. Esto implica igualdad de dos objetos Jaime, porque tienen el mismo nombre, pero no son el mismo objeto.

Asumiendo que Arturo y Valeria son en realidad padres de un mismo niño Jaime, todas las actualizaciones al hijo de Arturo, también son aplicadas al hijo de Valeria, ya que se trata del mismo objeto, Jaime. Si el sistema no tuviera identidad, sino que se basara en igualdad de objetos, serían necesarias dos actualizaciones a los dos objetos Jaime.



Unidad III. Modelo Orientado a Objetos



Soportar identidad de objetos implica que el OODBMS ofrece operaciones como asignación de objetos, copiado de objetos y comprobación de la identidad o igualdad de objetos.

La principal manera de implementar la identidad de objetos es mediante un OID (*Object Identifier*) independiente de los valores de los atributos del objeto. Estos son implementados por el sistema y muchas veces a bajo nivel, lo que mejora el rendimiento.

Según Date (2001: 847), los OID son innecesarios e indeseables en el nivel del modelo. En comparación con las claves primarias, los OID están ocultos al usuario mientras que las claves no. El uso de estos *object identifiers* no elimina el uso de claves primarias ya que son necesarias para unir al sistema con la realidad en la que se inserta; pensemos, por ejemplo, en los folios de facturación.

3. Encapsulación

La idea de encapsulación es tomada de los lenguajes de programación en los que para todo objeto existe una parte visible que permite especificar el conjunto de operaciones que pueden ser realizadas sobre el objeto. Otra parte es no visible y contienen los datos que almacena ese objeto.

Traducido a bases de datos, un objeto encapsula programas y datos. Por ejemplo, en un sistema relacional, un empleado es representado por una tupla. Para ser consultado desde una aplicación es necesario usar un lenguaje de programación para programar procedimientos que serán almacenados fuera de la base datos; es más, ese lenguaje puede ser en realidad la combinación de un lenguaje de alto nivel con el lenguaje estándar relacional SQL.

En un sistema de bases de datos orientado a objetos, definimos al empleado como un objeto que tiene una parte de datos (probablemente muy similar al



Unidad III. Modelo Orientado a Objetos



registro que definiríamos en el sistema relacional) y una parte de operaciones, la cual consiste en operaciones de `aumentodeSalario()` y `bajaDefinitiva()`, por ejemplo, que accederían a los datos del empleado. Cuando se almacene un nuevo empleado, datos y operaciones serían almacenados en la base de datos y no fuera de ella.

4. Herencia

La herencia tiene dos ventajas: es una herramienta poderosa de modelado, ya que brinda una descripción precisa del mundo y ayuda a simplificar la implementación de las aplicaciones. Para entender el manejo de la herencia en los sistemas de bases de datos orientados a objetos vamos a establecer un ejemplo:

Asumamos que tenemos empleados y estudiantes. Cada empleado tiene un nombre, edad y salario, además podemos aumentar su sueldo. Por su parte, cada estudiante tiene edad, nombre y un conjunto de calificaciones con las cuales podemos obtener su promedio.

En un sistema relacional, el diseñador de bases de datos definiría una relación empleado y una estudiante, también escribiría el código para la operación de aumentar sueldo. Para la relación empleado tendría que escribir el código para la operación de obtener promedio.

En un sistema orientado a objetos, usando adecuadamente la herencia, nos daríamos cuenta de que *empleado* y *estudiante* son personas y comparten los atributos: *nombre* y *edad*. Entonces, declararíamos una clase empleado como un tipo especial de la clase persona, el cual incluiría una operación especial de `aumentarSueldo()` y un atributo de salario. De forma similar se declararía el estudiante como un tipo especial de la clase persona con el atributo adicional de conjunto de Grados y la operación especial `obtenerPromedio()`.



Unidad III. Modelo Orientado a Objetos



El modelo es más cercano a la realidad y nos permite ahorrar código de programación. Por esto se dice que la herencia ayuda a reutilizar código ya que cada programa está disponible para ser compartido.

5. Sobreescritura y sobrecarga

En la programación orientada a objetos tenemos la ventaja de poder reescribir métodos con el mismo nombre. Esto significa contar con varios métodos que se llamen igual, pero que realicen distintas operaciones. Para poder programar estos métodos llamados sobrecargados, es necesario que cambie algo en sus parámetros, como el número, orden o tipo de dato. Esto mismo es posible de una base de datos orientada a objetos.

6. Completa capacidad computacional (*Computational completeness*)

Los manejadores de bases de datos relacionales cuentan con un lenguaje para realizar procesos computacionales sobre los datos: el SQL. Además, adicionan un lenguaje *procedural* (“de procedimiento”) que permite la definición de variables, manejo de excepciones, ciclos y estructuras condicionales. Estos lenguajes son pl/sql para Oracle, pl/pgsql para PostgreSQL y Transact-SQL para SQL Server de Microsoft, por dar unos ejemplos.

Los manejadores de bases de datos orientadas a objetos, también deben contar con un lenguaje que puede realizar cualquier procesamiento. En este sentido, lo más común es que los OODBMS integren lenguajes computacionalmente completos dentro de la base de datos. Estos pueden ser los que ya existen en el mercado y que se usan como lenguajes aplicación general (Java, C++, etc.).



ACTIVIDAD 1

Investiga las características de los manejadores de Bases de Datos OMG y OID.

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**.

ACTIVIDAD 2

Responde las siguientes preguntas.

1. ¿En qué consiste la persistencia en los sistemas de bases de datos orientados a objetos?
2. ¿Qué factores han permitido el desarrollo de los Sistemas de Bases de Datos Orientados a Objetos?
3. ¿Con qué hecho se origina la Segunda Generación de los Sistemas de Bases de Datos Orientados a Objetos?
4. Anota los problemas para la Definición de Sistemas.
5. ¿Cuáles son los dos criterios a satisfacer en OODBS?

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**.



Unidad III. Modelo Orientado a Objetos



ACTIVIDAD 3

Realiza un cuadro sinóptico que resuma las razones por las que surgieron o fueron necesarios los OODBMS.

Realiza tu actividad en un procesador de textos, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma.



Unidad III. Modelo Orientado a Objetos



Autoevaluación

Selecciona si las aseveraciones son verdaderas (V) o falsas (F). Al concluir tendrás tu calificación de manera inmediata.

	Verdadera	Falsa
1. Un OODBMS emplea objetos relacionados.	()	()
2. Un OODBMS debe tener como mínimo conjuntos, listas y tuplas.	()	()
3. La Herencia ayuda en el modelado del mundo.	()	()
4. Los Métodos sobrecargados cambian sus parámetros.	()	()
5. Lenguaje + Base de Datos = Aplicación.	()	()

Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción



Unidad III. Modelo Orientado a Objetos



Tema 3. Estándar OMG

Objetivo del tema

Identificar las características de un Object DataBase Management Group.

Desarrollo

El *Object Database Management Group* (ODMG) surge en 1991 formado por un grupo de proveedores de OODBMS. Generaron un primer estándar en 1993 en donde exponían las características que consideraban necesarias en un sistema de bases de datos de este tipo. En 2001 aparece la nueva versión del estándar que se acomoda a las especificaciones de la tecnología de Java.

Los principales componentes de un OODBMS según el ODMG son:

- * Lenguaje de definición de objetos (ODL).
- * Lenguaje de consulta de objetos (OQL).
- * Conexión con los lenguajes C++, Smalltalk y Java.

En 1989, el Artículo "*The Object-Oriented Database System Manifesto*",] aunque con un enfoque demasiado limitado en temas de administración, Stonebraker et al. (1990), propone una definición compuesta de tres tipos de reglas que deben respetar un OODBMS:

I. Reglas Obligatorias: Las cuales, el sistema debe imperativamente seguir para merecer la calidad de OODBMS.

II. Reglas Facultativas: Lineamientos suplementarios del sistema.



Unidad III. Modelo Orientado a Objetos



El estándar ODMG-93: *Un estándar para bases de datos orientadas al objeto puras*

Es el resultado de trabajos que duraron 18 meses por los 5 principales distribuidores de OODBMS. Su objetivo fue asegurar la portabilidad de las aplicaciones de un sistema a otro. En este objetivo son definidas tres interfaz:

- 1) ODL (Lenguaje de Definición de Objetos) El lenguaje de definición del objeto permite definir el modelo de datos. Es compatible con IDL, el lenguaje del OMG (Grupo de Administración de Objetos). Permite la definición de objetos complejos, de relación entre esos objetos y de métodos asociados a dichos objetos.
- 2) OQL (Lenguaje de Consulta al Objeto) El lenguaje de requerimientos permite consultar los objetos de estructuras complejas, de enviar mensajes a objetos, efectuar join y otras operaciones de tipo asociativo. Su sintaxis es del tipo SQL.
- 3) Conexión vía C++ y Smalltalk. Esta interfaz ("bindings", enlazamientos), especifica como se debe hacer la programación en C++ o Smalltalk de una aplicación sobre una base de datos que ha sido declarada en ODL. La conexión es basada sobre la noción de "puntero inteligente" que permite manejar los objetos persistentes como objetos ordinarios vía punteros persistentes."

Hay una cosa más que podemos hacer con los campos, podemos asignar a las limitaciones, por ejemplo Si asigna ÚNICO obstáculo para cualquier columna, entonces no sería capaz de insertar en ella cualquier valor que ya está en su misma columna. Ayudar a las limitaciones en el mantenimiento de la integridad de los datos del sistema.



Unidad III. Modelo Orientado a Objetos



SQL no es sino un conjunto de comandos / declaraciones y se clasifican en cinco grupos a saber., Dql: Información de Query Language, LMD: Lenguaje de Manipulación de Datos, DDL: Data Definition Language, TCL: Lenguaje de control de las transacciones, DCL: Lenguaje de control de datos.

DQL: SELECT Dql: SELECT

DML: DELETE, INSERT, UPDATE LMD: SUPR, INSERT, UPDATE

DDL: CREATE, DROP, TRUNCATE, ALTER DDL: CREATE, DROP, TRUNCATE, ALTER

TCL: COMMIT, ROLLBACK, SAVEPOINT TCL: COMMIT, ROLLBACK, SAVEPOINT

DCL: GRANT, REVOKE DCL: GRANT, y REVOKE “



Unidad III. Modelo Orientado a Objetos



ACTIVIDAD 1

Desarrolla un mapa conceptual sobre las bases de datos orientadas a objetos que incluya su evolución, definición y características

Realiza tu actividad en un procesador de textos, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma.

ACTIVIDAD 2

Realizar un resumen del siguiente Artículo: **OODBMS (Base de datos de objetos) Ventajas**. Ver en:

http://translate.google.com.mx/translate?hl=es&sl=en&u=http://www.versant.com/en_US/products/oodbms.html&ei=_3nSSYOyFlyKtAOvzvnEAW&sa=X&oi=translate&resnum=4&ct=result&prev=/search%3Fq%3DOODBMS%26hl%3Des%26sa%3DG

Realiza tu actividad en un procesador de textos, guárdala en tu computadora y una vez concluida, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma.



Unidad III. Modelo Orientado a Objetos



Autoevaluación

Participa en el siguiente juego de serpientes y escaleras. Tira el dado, lee la pregunta e indica si la frase que se te presenta es **verdadera** o **falsa**. Para avanzar deberás contestar correctamente.



- 1 Los OODBMS surgen por la necesidad de tipos de datos de longitud variable.

V () F ()

- 2 Los sistemas de bases de datos inteligentes son una tendencia de la tecnología actual de bases de datos.

V () F ()



Unidad III. Modelo Orientado a Objetos



- 3 La orientación a objetos evita utilizar la terminología del dominio del negocio.

V () F ()

- 4 Los objetos tienen estado y comportamiento.

V () F ()

- 5 El modelo de base de datos orientado a objetos tiene un sustento fuertemente teórico y formal.

V () F ()

- 6 Un OODBMS debe contar con herencia.

V () F ()

- 7 Las tuplas, conjuntos y listas pueden considerarse objetos complejos.

V () F ()

- 8 Dos objetos con distinto OID, pero valores de atributos iguales son idénticos.

V () F ()

- 9 La sobrecarga y la sobrescritura son conceptos equivalentes.

V () F ()

- 10 La orientación a objetos se caracteriza por la poca reutilización de código.

V () F ()



Unidad III. Modelo Orientado a Objetos



Bibliografía básica

Autor	Capítulo	Páginas

Sitios electrónicos

Sitio	Descripción



Unidad III. Modelo Orientado a Objetos



Lo que aprendí

1. Elabora un Mapa Conceptual de la Unidad que incluya todos los elementos estudiados en esta unidad.
2. Observa las siguientes tablas.

¿Qué comandos emplearías para insertar dos registros entre las siguientes dos tablas? Anótalos.

Almacén		
Id_Producto	Nomb_Producto	Precio_producto

Proveedores		
Id_Proved	Nomb_proved	Direc_proved

Para enviar tu respuesta, pulsa el botón **Editar mi envío**; se mostrará un editor de texto en el cual puedes redactar tu información; una vez que hayas concluido, salva tu actividad pulsando el botón **Guardar cambios**.



Unidad III. Modelo Orientado a Objetos



GLOSARIO

Encapsulación

Es tomada de los lenguajes de programación en los que para todo objeto existe una parte visible que permite especificar el conjunto de operaciones que pueden ser realizadas sobre el objeto.

Herencia

Tiene dos ventajas: es una herramienta poderosa de modelado, ya que brinda una descripción precisa del mundo.

Identidad de objetos

El objetivo es contar con objetos que tengan una existencia independiente de sus valores.

Objetos complejos

Son creados a partir de objetos (tipos de datos) simples. Estos objetos simples son: enteros, caracteres, cadenas de bytes, booleans y números de punto flotante.

Orientación a objetos

Representa el mundo real y resuelve problemas a través de objetos, ya sean tangibles o digitales.

Persistencia

Es una característica necesaria de los datos en un sistema de bases de datos. Recordemos que consiste en la posibilidad de recuperar datos en el futuro. Esto implica que los datos se almacenan a pesar del término del programa de aplicación.



Unidad III. Modelo Orientado a Objetos



MESOGRAFÍA

Bibliografía básica

Bibliografía complementaria

Sitios electrónicos