

UNIVERSIDAD DE CASTILLA-LA MANCHA  
ESCUELA SUPERIOR DE INFORMÁTICA



**Modelos Avanzados de Base de Datos**

LISTADO DE PREGUNTAS DEL TRABAJO FUNCIONALIDAD 1:  
BASES DE DATOS ORIENTADAS A OBJETOS Y BASES DE  
DATOS OBJETO-RELACIONALES

Juan Andrada Romero  
Juan José Antequera Flores  
Jose Domingo López López

8 de marzo de 2010

## 1. PREGUNTAS SELECCIONADAS

**Pregunta:** En el trabajo se comentan los lenguajes de definición de objetos y de consulta de objetos que define el estándar ODMG, pero no se hace referencia a ningún lenguaje para poder manipular objetos, es decir, insertarlos en la base de datos, modificarlos o eliminarlos. ¿Dicho estándar define un lenguaje de manipulación de datos específico? ¿Cómo se realizarían las manipulaciones sobre objetos en una base de datos orientada a objetos?

**Respuesta:** El estándar ODMG-93 sugiere que el lenguaje de manipulación de objetos (OML) sea la extensión de un lenguaje de programación, de forma que se pueden realizar entre otras las siguientes operaciones sobre la base de datos: creación, borrado, y modificación de un objeto. [1].

Por tanto, los objetos se manipulan dentro del lenguaje de programación orientado a objetos, utilizando el concepto de transacción y de clases persistentes. De este modo, como indica el ODMG API ([3]), para realizar las operaciones de inserción, modificación o eliminación de un objeto dentro del lenguaje de programación, se procedería de la siguiente forma:

1. Crear una transacción.
2. Obtener la base de datos en uso.
3. Ejecutar la operación de inserción, modificación o eliminación.

Un ejemplo de inserción sería el siguiente:

```
public static void storeNewProduct(Product product)
    Transaction tx = odmng.newTransaction();
    tx.begin();
    // get current used Database instance
    Database db = odmng.getDatabase(null);
    // make persistent new object
    db.makePersistent(product);
    tx.commit();
```

**Pregunta:** En el documento se comentan algunas características de los SGBDOR, pero no se mencionan las de un SGBDOO. ¿Cuáles son las características que debe un cumplir un SGBDOO para que sea considerado como tal?

**Respuesta:** En 1989 se hizo el Manifiesto [2] de los sistemas de base de datos orientados a objetos el cual propuso trece características obligatorias para un SGBDOO y cuatro opcionales, las cuales son:

- Características obligatorias de orientación a objetos:
  1. Deben soportarse objetos complejos
  2. Deben soportarse mecanismos de identidad de los objetos
  3. Debe soportarse la encapsulación
  4. Deben soportarse los tipos o clases
  5. Los tipos o clases deben ser capaces de heredar de sus ancestros
  6. Debe soportarse el enlace dinámico
  7. El DML debe ser computacionalmente complejo
  8. El conjunto de todos los tipos de datos debe ser ampliable
- Características obligatorias de SGBD:
  1. Debe proporcionarse persistencia a los datos
  2. El SGBD debe ser capaz de gestionar bases de datos de muy gran tamaño
  3. El SGBD debe soportar a usuarios concurrentes
  4. El SGBD debe ser capaz de recuperarse de fallos hardware y software
  5. El SGBD debe proporcionar una forma simple de consultar los datos.
- Características opcionales:
  1. Herencia múltiple
  2. Comprobación de tipos e inferencia de tipos
  3. Sistema de base de datos distribuido
  4. Soporte de versiones

## 2. PREGUNTAS DESCARTADAS

**Pregunta:** Si las bases de datos orientadas a objetos surgieron para combatir las limitaciones de las bases de datos relacionales, ¿por qué siguen siendo las bases de datos relacionales las más utilizadas?

**Respuesta:** Se siguen usando porque es un estándar muy arraigado y extendido, y en muchas empresas es muy costoso hacer un cambio a nuevas tecnologías aunque aporten más ventajas que las antiguas. Además, el estándar de las bases de datos orientadas a objetos no está todo lo maduro que debiera para poder hacer la migración sin correr ningún riesgo. Por otra parte, las bases de datos orientadas a objetos suelen utilizarse en aplicaciones de diseño y fabricación por computador (CAD), que no están tan extendidas como otras aplicaciones comerciales que hacen uso de bases de datos relacionales. Por esta razón, se siguen utilizando en mayor medida, por estar más extendidas en las aplicaciones comerciales.

**Pregunta:** ¿Qué ventajas ofrece un sistema gestor de bases de datos orientadas a objetos frente a un sistema clásico relacional? [4]

**Respuesta:**

- Los objetos en la OODMBS pueden almacenar un número arbitrario de tipos atómicos, así como otros objetos.
- Los datos en el mundo real tienen habitualmente características jerárquicas.
- No es necesario un lenguaje de consultas para acceder a datos de la OODB, al contrario que ocurre con RDB, ya que la interacción con la base de datos es transparente al acceder a los objetos.
- En una aplicación típica que usa un lenguaje de programación orientado a objetos y un RDMS se necesita bastante tiempo en «mapear» las tablas a partir de los objetos.
- No hay claves primarias.
- Un modelo de datos debería modelar entidades y sus relaciones, restricciones y operaciones que cambian el estado de los datos en el sistema.

## REFERENCIAS

- [1] Bases de Datos Orientadas a Objetos. <http://www.elrinconcito.com/articulos/BaseDatos/BasesDatos.htm#3.2>.
- [2] Manifiesto Malcolm Atkinson: características de un BDOO. <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/clamen/OODBMS/Manifiesto/htManifiesto/Manifiesto.html>.
- [3] ODMG API. <http://db.apache.org/ojb/docu/tutorials/odmg-tutorial.html#Persisting+New+Objects>.
- [4] Why aren't you using an oodbms. <http://www.25hoursaday.com/WhyArentYouUsingAnOODBMS.html>.