

PAC 1. UF6.

*Diseño de programas con
lenguajes de
programación orientados a
objetos para la gestión de
bases de datos.*

UF6_PAC1

1. ¿Qué es una base de datos relacional?

Una base de datos relacional es un tipo de base de datos que cumple con el modelo relacional, que se basa fundamentalmente en el uso de relaciones, las cuales podrían considerarse como conjuntos de datos llamados tuplas. Es el modelo mas utilizado en la actualidad para administrar datos de forma dinámica.

Almacena la información de mundo real a través de tablas que se relacionan entre sí, para organizar mejor la información y poder llegar de un dato a otro por relación entre sus tablas. Todos los datos son almacenados en la base de datos en forma de relaciones que se visualizan como una tabla, la cual contiene filas y columnas. Las primeras son los atributos de la tabla, mientras que las segundas son los registros donde almacenaremos la información.

2. ¿Qué es una base de datos objeto-relacional?

Una base de datos objeto-relacional es una base de datos relaciona a la cual se le añade una extensión para poder programar sus tablas o relaciones, orientándola a objetos. Se puede guardar un objeto en una tabla. Es una base de datos híbrida que alberga dos modelos, el relacional y el orientado a objetos.

Sus características entre otras son que pueden definir tipos de datos más complejos, usar colecciones o conjuntos, atributos compuestos y almacenamiento de objetos de gran tamaño.

3. ¿Qué es una base de datos orientada a objetos?

En una base de datos orientada a objetos la información se representa mediante objetos. El resultado es un sistema gestor de base de datos orientada a objetos, que hace que los objetos de la base de datos aparezcan como objetos de un lenguaje de programación en uno o mas lenguajes a los que de soporte.

Se diseñan de la misma forma que los POO. Cada tabla se convertirá en objeto de la base de datos. Los objeto pueden heredar unos de otros, así como tendrán un identificador único que los diferencie del resto. El usuario se encargará de decidir los elementos que van a formar parte de la base de datos. Además tendremos como características de la programación orientada a objetos, la sobrecarga y el poliformismo.

4. ¿Qué es ODMG?

Es un estándar Object Data Management Group, utilizado para definir el grupo de empresas que forman parte del desarrollo de un nuevo modelo de objetos, que permite realizar el diseño de bases de datos implementadas desde lenguajes de programación orientada a objetos.

Su función es definir los elementos y la persistencia entre ellos en las Bases de datos Orientadas a Objetos, permitiendo que sean portarles en los sistemas de Programación Orientada a Objetos que las soportan. Los componentes básicos son los objetos y lo literales.

5. ¿Qué es el lenguaje ODL? ¿Qué es el lenguaje OQL? Explica las diferencias entre ellos.

El ODL es Object Definition Language, es el lenguaje de especificación para definir los tipos de objetos para sistemas compatibles con ODBMS. Es el equivalente al DDL en los SGBF. Define los atributos y las relaciones entre tipos y especifica la signatura de las operaciones. Es un lenguaje de definición.

El OQL es Object Query Language, es un lenguaje de consulta estándar para bases de datos orientadas a objetos modelado a partir de SQL. Permite realizar consultas de modo eficiente sobre bases de datos orientadas a objetos.

ODL sirve para crear tipos de objetos, relaciones, es decir, es un lenguaje de definición. OQL es un lenguaje de consulta de objetos, con el que consultaremos valores de esos objetos creados y la estructura de la base de datos.

6. ¿Qué es SGBDOO? Explica sus características.

SGBDOO significa Sistema Gestor de Bases de Datos Orientadas a Objetos. Es un sistema gestor con la característica de almacenar objetos. Se puede tratar directamente con objetos y no se tiene que hacer la traducción de registros o tablas. Combina el sistema gestor de bases de datos y el sistema orientado a objetos.

Sus características principales son:

- En la parte de bases de datos: Persistencia, gestión de almacenamiento secundario, concurrencia, recuperación y facilidad de consultas.
- Para la parte de objetos: objetos complejos, identidad de objetos, encapsulamiento, tipos, clases, herencia, sobrecarga de métodos, extensionalidad y completitud computacional.