## **ACTIVIDAD 1 – introducción y conceptos básicos**

**ACTIVIDAD 1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Tipo de sistema de persistencia** | **Estructura básica** | **Soporte para transacciones** | **Grado de estandarización** | **Madurez** | **Implantación actual** | **Observaciones relevantes** | | --- | --- | --- | --- | --- | --- | --- | | **Bases de datos relacionales (SQL)** | Tablas (filas/columnas) | Siempre (ACID) | Muy alta (SQL estándar) | Muy alta (décadas de uso) | Muy alta | Excelentes para datos estructurados y consistencia; limitaciones en escalabilidad horizontal | | **Bases de datos orientadas a objetos** | Objetos y relaciones entre ellos | A veces | Baja (pocos estándares fuertes) | Media (existen desde los 90) | Baja | Utiles en nichos con OOP muy fuerte; escasa adopción comercial | | **Bases de datos XML nativas** | Documentos XML jerárquicos | A veces | Media (XQuery, XPath del W3C) | Media | Baja | Buen soporte para datos jerárquicos y documentos; menos relevantes frente a JSON | | **Bases de datos NoSQL** | Clave-valor, documentos (JSON), grafos, columnas | A veces | Baja-media (no hay un estándar único) | Media (desde 2000s) | Alta (Big Data, web apps, escalabilidad) | Alta flexibilidad y escalabilidad, sacrifican consistencia fuerte en algunos casos | | **Archivos planos / ficheros** | Texto o binario lineal | No | Nula | Muy alta (desde inicios de la informática) | Media-baja (para casos simples) | Simples, portables, pero limitados para datos complejos o multiusuario | | **Bases de datos en memoria** | Clave-valor en RAM | A veces | Baja | Media | Media (usadas en cachés y sistemas de alta velocidad) | Muy rápidas, pero persistencia limitada o dependiente de snapshots | |

**ACTIVIDAD 2**

Test de autoevaluación Examen de Conceptos de Almacenamiento y Acceso a Datos

1. La persistencia de datos: a) Se utiliza para datos cuyo valor muy rara vez cambia. El resto se mantiene siempre en memoria principal. b) Comprende un conjunto de técnicas que permiten transferir datos de memoria principal a almacenamiento secundario y viceversa. c) Solo es posible para algunos tipos de datos. d) Permite utilizar siempre las mismas operaciones para manejar los datos, tanto si están en memoria principal como si están en almacenamiento secundario.

2. Los programas de aplicación acceden a datos persistentes: a) Bien directamente si están almacenados en el mismo ordenador, bien mediante API si están almacenados en otro. b) Mediante API, que permiten utilizar los servicios de persistencia. c) Mediante protocolos de red, que dan acceso a los servicios de persistencia. d) Previo volcado binario en memoria principal desde almacenamiento secundario.

3. Un documento de XML: a) Se almacena normalmente en varios directorios organizados jerárquicamente. b) Tiene estructura jerárquica. c) Al tener estructura jerárquica, no se puede almacenar en una estructura lineal como un fichero. d) Se puede almacenar en un fichero, pero debe ser necesariamente binario.

4. Las restricciones de integridad: a) Solo se pueden definir para bases de datos relacionales. b) Solo se pueden preservar en una base de datos utilizando transacciones. c) Se pueden definir para bases de datos de cualquier tipo. d) Se definen para garantizar la consistencia de datos relacionados entre sí.

5. Los iteradores: a) Permiten recuperar datos persistentes, pero no modificarlos. b) Solo permiten recorrer estructuras lineales o tabulares. No documentos de XML, por ejemplo. c) Obtienen los datos siempre uno a uno del almacenamiento secundario. d) Permiten obtener uno a uno los resultados de una consulta.

6. Una transacción: a) Podría fallar en caso de interbloqueo, es decir, cuando esa transacción y otras estén paradas esperando a que alguna de ellas termine. b) Solo terminará con éxito si lo hacen todas las operaciones que incluye. c) Puede ejecutarse en paralelo con otras transacciones. d) Todas las respuestas anteriores son correctas.

7. Las bases de datos relacionales: a) Almacenan los datos en tablas. b) Tienen, en general, un excelente soporte para transacciones. c) Se manejan con el lenguaje SQL. d) Todas las respuestas anteriores son correctas.

8. Las bases de datos de objetos: a) Permiten almacenar directamente objetos, pero no objetos complejos. b) No suelen tener soporte para transacciones. c) Son las únicas que permiten almacenar directamente objetos. d) Ninguna de las opciones anteriores es correcta.

9. Las bases de datos de XML nativas: a) Incluyen soporte para colecciones de acuerdo a estándares del W3C. b) Son las únicas que permiten almacenar directamente documentos de XML, a la vez que proporcionan soporte para el estándar XQuery de W3C. c) Suelen tener soporte para transacciones. d) No tienen un lenguaje estándar universalmente adoptado para operaciones de modificación de datos.

10.Las bases de datos NoSQL: a) Priman la disponibilidad sobre la consistencia e integridad de los datos. b) Son importantes para las aplicaciones de big data. c) Suelen utilizar estructuras de almacenamiento muy sencillas y flexibles. d) Todas las opciones anteriores son correctas.

## **ACTIVIDAD 1 – ficheros**

**ACTIVIDAD 1**

## **ACTIVIDAD 2 – ficheros**

**ACTIVIDAD 1**

## **ACTIVIDAD 3 – ficheros**

**ACTIVIDAD 1**

## **ACTIVIDAD 4 – ficheros**

**ACTIVIDAD 1**