

Acceso a Datos

Introducción y conceptos básicos

Índice

-
1. Objetivos
 2. Programas y datos
 3. Persistencia de datos
 4. Sistema de persistencia
 5. Almacenamiento de Datos
 6. Acceso a datos con Iteradores
 7. Control de accesos concurrentes y transiciones
 8. Persistencia de datos

1.- Objetivos

- Conceptos nuevos:
 - Persistencia de datos
 - Formas de Almacenamiento
 - Mecanismos para trabajar con datos persistentes
- Repaso de los conceptos más importantes de la asignatura de BD (1)
- Repaso de colecciones de Datos en Java

2.- Programas y Datos

- Un programa “gestiona” información
 - Información => Conjunto de datos estructurados
- Existe multitud de formas de trabajar con datos, pero depende :
 - Del tipo de aplicación que tengamos
 - Del tipo de “procesador” que maneje los datos
 - PC, Tablet, Movil etc
- Tipos de almacenamiento
 - Almacenamiento primario o en memoria principal
 - Almacenamiento secundario o permanente

2.- Programas y Datos

- **Almacenamiento primario o en memoria principal**
 - Datos de acceso inmediato por parte de la aplicación
 - Son volátiles : desaparecen cuando se apaga el equipo
 - Su tamaño viene limitado por el espacio
 - Velocidad de acceso alta
- **Almacenamiento Secundario o permanente**
 - Dispositivo físico donde se almacenan los datos de forma permanente
 - HD, memorias Flash
 - Gran capacidad de almacenamiento
 - Tiempos de acceso largos

3.- Persistencia de Datos

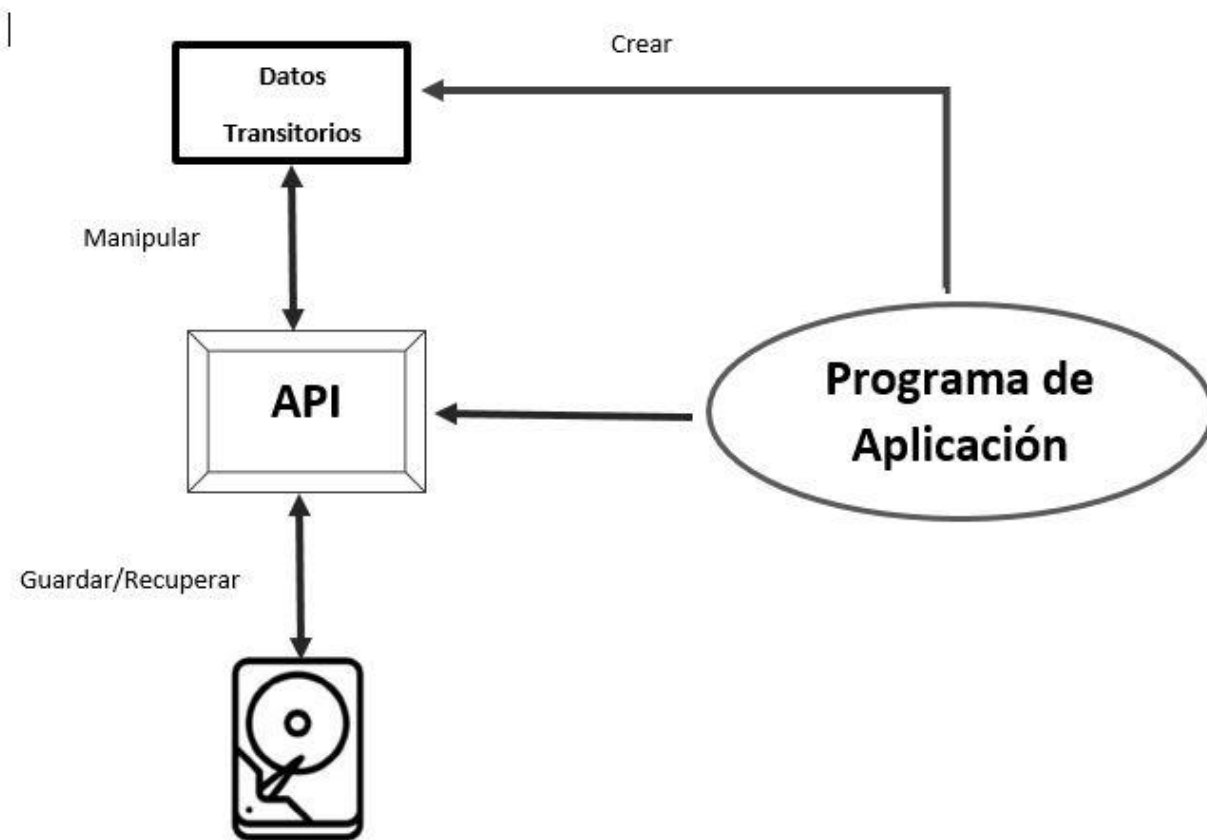
- Acciones de los programas sobre los datos:
 - **Extraer datos:**
 - Seleccionar un conjunto de información de un almacenamiento secundario
 - Colocarlo en memoria para que sea accesible por las aplicaciones
 - **Crear/modificar datos**
 - Los datos almacenados memoria principal son creados o modificados por las aplicaciones
 - Una vez manipulados se guardan en almacenamiento secundario.

3.- Persistencia de Datos

- **Almacenamiento de datos persistentes**

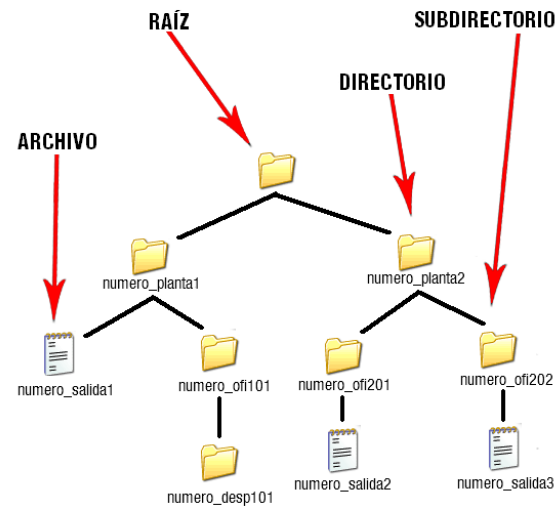
- Gran volumen de datos => Se necesita gran capacidad
- Datos compartidos
- **Soluciones: Almacenar datos en servidores**
 - Servidores dedicados a la persistencia de datos para multiples aplicaciones
 - Ubicados en diferentes equipos
 - La comunicación se realiza a través de protocolos de red
 - Dispone de funciones de alto nivel para consultas

3.- Persistencia de Datos



4.- Sistemas de Persistencia de datos

- Los datos se almacenan en ficheros
- Los ficheros se almacenan dentro de un sistema de ficheros



4.- Sistemas de Persistencia de datos

- Sistema de ficheros:
 - Estructura jerárquica basada en el concepto de **directorio**
 - **Directorio = Carpeta**
 - Puede contener en su interior
 - Ficheros
 - Directorios
- **Importante:**
 - Tener en cuenta las necesidades futuras para poder seleccionar el tipo de almacenamiento secundario

5.- Almacenamiento de Datos

- Tipos:
 1. Ficheros
 2. Bases de datos SQL
 - Bases de datos Relacionales
 3. Bases de datos NoSQL
 - Documentos XML
 - Bases de datos de Objetos
- Bases de datos NoSQL

5.1 Ficheros

- Secuencia de bytes
- Proporciona una organización secuencial de bytes
- Representan cualquier información



```
00000000: 00 50 55 40 01 01 00 00 34 00 00 00 20 01 00 00 : .PGF.....
00000010: 0A 00 00 00 00 00 04 04 04 00 00 00 04 00 00 00 : .....
00000020: 00 00 00 00 09 00 04 02 03 00 00 00 04 00 00 00 : .....
00000030: 04 00 00 00 12 00 04 02 0A 00 00 00 10 00 00 00 : .....
00000040: 08 00 00 00 1A 00 04 04 04 00 00 00 04 00 00 00 : .....
00000050: 18 00 00 00 28 00 04 04 04 00 00 00 04 00 00 00 : .....
00000060: 1C 00 00 00 31 00 04 02 05 00 00 00 08 00 00 00 : .....
00000070: 20 00 00 00 32 00 04 04 04 00 00 00 04 00 00 00 : .....
00000080: 28 00 00 00 45 00 04 02 05 00 00 00 08 00 00 00 : .....
00000090: 2C 00 00 00 5C 00 04 04 04 00 00 00 04 00 00 00 : .....
000000A0: 34 00 00 00 63 00 04 02 0A 00 00 00 00 00 00 00 : .....
000000B0: 38 00 00 00 42 4F 6F 54 41 42 4C 45 30 42 41 54 : 6...BOOTABLE.CAT
000000C0: 45 47 4F 52 59 00 44 49 53 43 5F 49 44 00 44 49 : EXPORT.DISC_ID.DI
000000D0: 50 43 5F 4E 55 40 42 45 52 00 46 49 53 43 5F 54 : GC_NUMBER.DISC_T
000000E0: 4F 54 41 4C 00 46 49 53 43 52 56 45 52 53 49 47 : OTAL.DISC_VERSION
000000F0: 4E 00 00 41 52 45 4E 04 41 4C 5F 4C 43 56 45 4C : M.PARENTAL_LEVEL
00000100: 00 50 5C 50 5F 53 59 53 54 4C 4B 5F 55 45 53 00 : .PGF_SYSTEM_VERSION
00000110: 53 45 47 49 4F 4E 00 54 44 54 4C 45 00 00 00 00 : REGION_TITLE....
00000120: 01 00 00 00 55 47 00 00 55 4C 45 53 30 30 39 30 : ...EXE_LINKS0070
00000130: 31 00 00 00 00 00 00 00 01 01 00 00 01 00 00 00 : 3.....
00000140: 51 22 30 50 00 00 00 00 02 00 00 00 32 2E 58 32 : 1.00.....2.82
00000150: 00 00 00 00 00 80 00 00 4B 69 6E 64 20 51 78 69 : .....C..Kind Qui
00000160: 7A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : .....
00000170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : .....
00000180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : .....
00000190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : .....
000001A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : .....
000001B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : .....
000001C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : .....
```

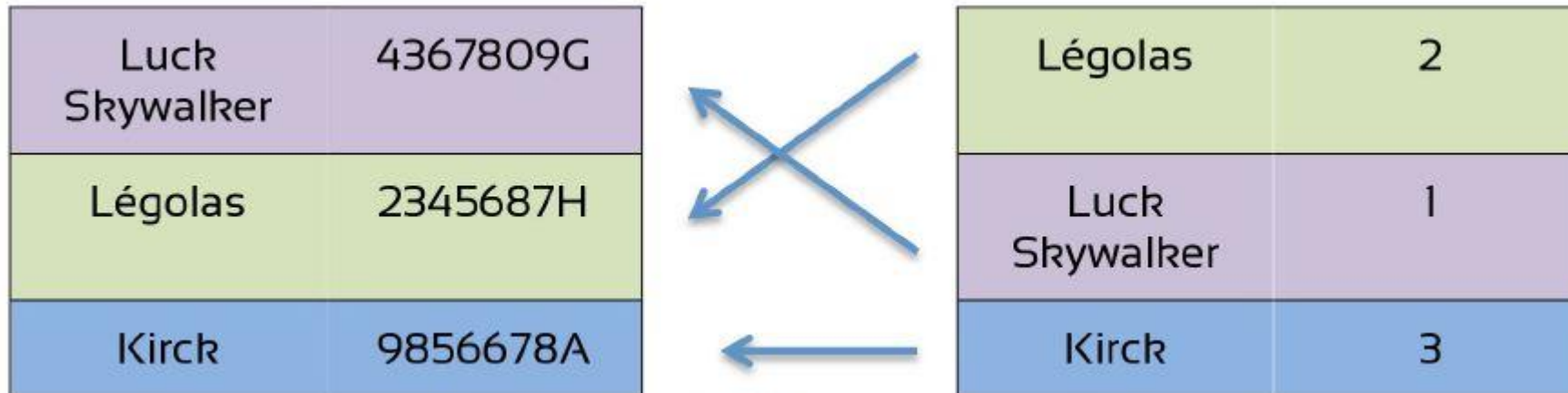
5.1 Ficheros

- Tipos de ficheros:
 - Secuencial:
 - La información se almacena de forma secuencial
 - Registros son de longitud fija
 - Los registros compuestos de campos de longitud fija
 - Secuencia fija de registros

Luck Skywalker	4367809G
Légolas	2345687H
Kirck	9856678A

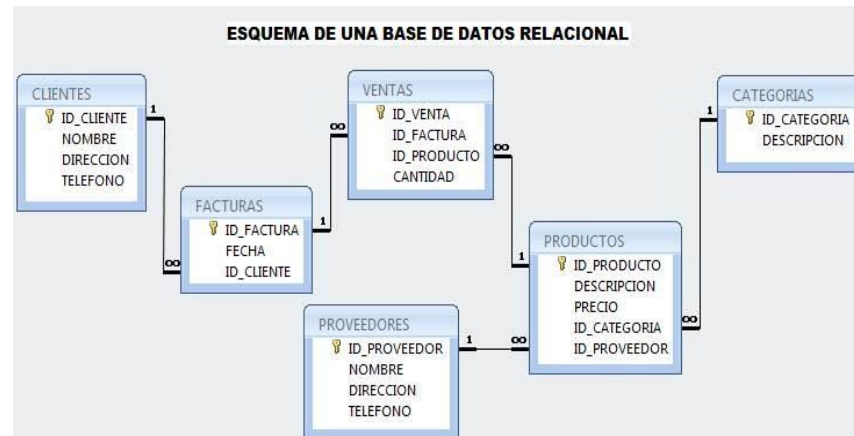
5.1 Ficheros

- Tipos de ficheros:
 - Indexados
 - Se crean ficheros índices
 - Formado por un campo ordenado y una posición del fichero de datos



5.2 Bases de datos Relacionales

- Los datos se organizan en tablas
- Existen relaciones entre ellos
 - Lenguaje estándar : **SQL**
 - Conocidas como bases de datos SQL

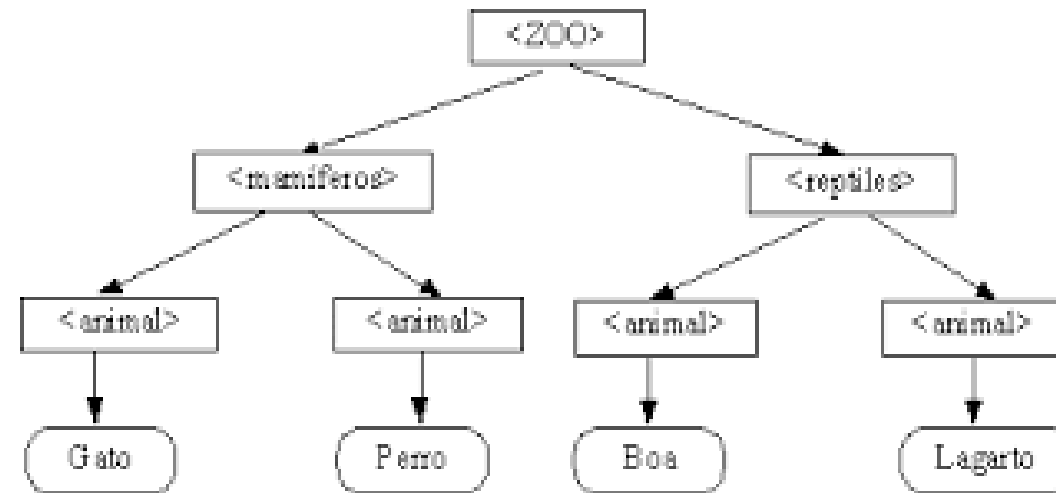


5.4 Bases de datos NoSQL

- Almacenan la información como documento usando una estructura
 - JSON
 - XML
- Manejan grandes volúmenes de datos
- Desarrollo de metodologías ágiles scrum
- Volumen creciente de datos y dispositivos
- Necesidades de gran escalabilidad

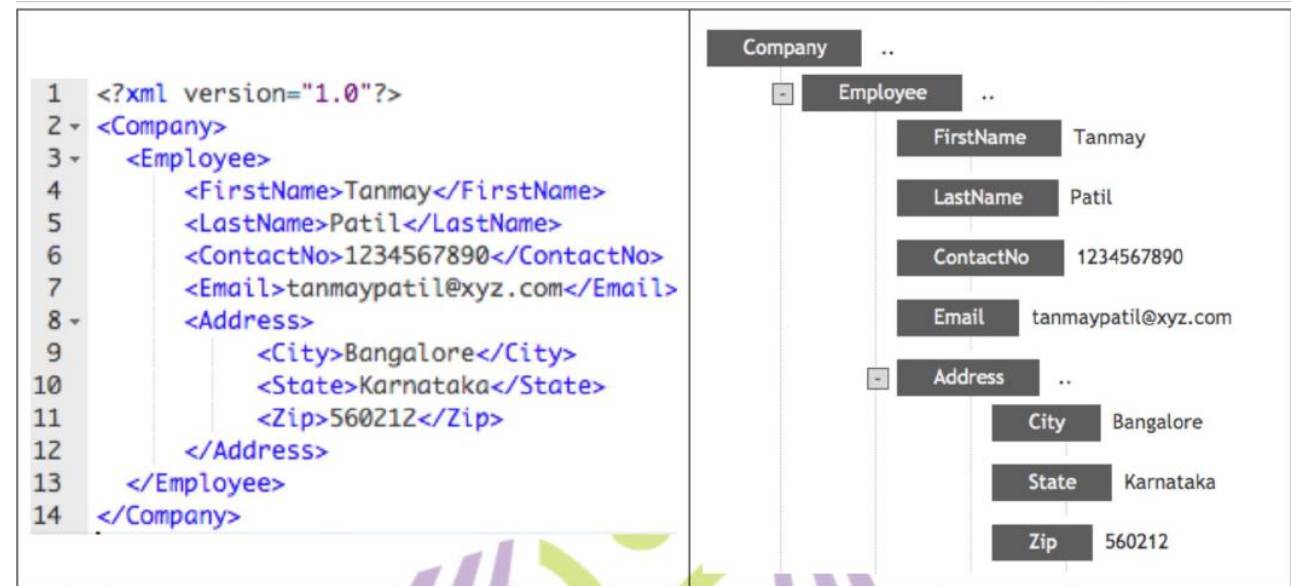
5.3 Documentos XML

- Un documento XML organiza la información de forma jerárquica
 - Forma de Árbol
 - Modelo DOM : representa el contenido del un documento XML árbol
 - Jerarquía de nodos

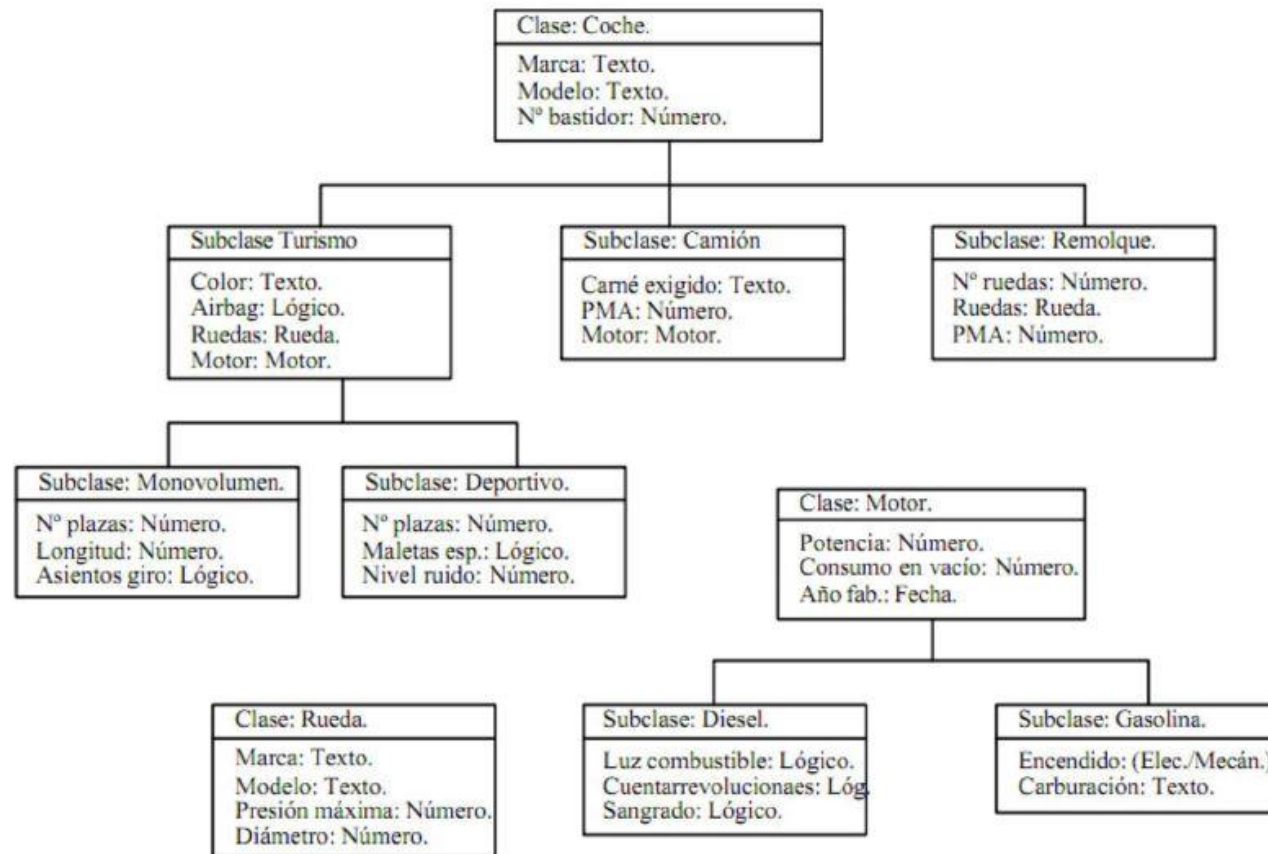


5.3 Documentos XML

- Un fichero de XML se almacena como fichero de texto
- Colecciones de ficheros XML : almacenamiento
 - Basada en directorios:
 - Jerarquía de árbol de directorios
 - Basada en Bases de datos
 - Jerarquía de colecciones



5.3 Bases de datos de Objetos



6.- Restricciones de Integridad

- Condiciones que se imponen a los datos y a las relaciones
 - Ejemplo
 - DNI longitud 9 dígitos y una letra
 - No incluir un producto que no existe en un pedido
- Funcionamiento básico:
 - Si no se cumplen las condiciones no se puede crear/modificar datos
 - Si no se cumplen las condiciones no podremos realizar una operación

7.- Acceso a datos con iteradores

- Iteradores
 - Permiten el acceso a datos mediante una consulta a la base de datos
 - Limita el numero de datos que procesamos en una aplicación
 - Ejemplo
 - Una empresa de telecomunicaciones con miles de registros
 - Podemos tener diferentes tipos de almacenamiento
 - BD relacionales => acceso es fila a fila
 - BD XML => acceso es nodo a nodo
 - BD Objetos => acceso objeto a objeto
- Extraemos bloques de datos de la base de datos
- Iteradores: recorren el bloque dato a dato

8.- Control de accesos concurrentes y transiciones

- Acceso concurrente:
 - Muchos usuarios
 - Forma simultanea
 - Diferentes Operaciones (Lectura/Escritura)
- Evitar conflictos
- Arbitrar en el acceso a los datos
- Transición:
 - Conjunto de datos sobre los que aplicamos unas operaciones de forma
 - Conjunta
 - Aislada

8.- Control de accesos concurrentes y transiciones

- DEF:
 - Una secuencia de operaciones de lectura y actualización de datos que forman una unidad lógica y se ejecutan como un todo
 - Conjunto de operaciones sobre un bloque de datos ejecutadas de forma indivisible o atómica
- Los datos siempre deben ser :
 - Consistentes
 - Actualizados
 - Correctos
- Importante:
 - Sincronización en operaciones Lectura/escritura y transiciones

8.- Control de accesos concurrentes y transiciones

- Características (ACID)
 - Atomic (atómica)
 - Debe ejecutarse completamente y sin errores
 - Si hay error, los datos deben de volver a su valor anterior
 - Consistent (consistente)
 - Se deben respetar todas las restricciones de integridad durante las operaciones
 - Isolated (aislada)
 - Cada transición se ejecuta de forma aislada
 - Las transiciones son “serializables”, se ejecuta una detrás de otra de forma secuencial
 - Si una transición depende de otra podemos encontrarnos con **interbloqueo**
 - Durable (duradera)
 - Una vez ejecutadas todas las operaciones, se confirman los cambios y se guardan de forma permanente

9.- Persistencia en los datos de ficheros

- Fichero:
 - Secuencia de bytes que contiene información, posee un nombre y una ubicación dentro de un directorio del sistema de almacenamiento de ficheros
 - Solo los usaremos en aplicaciones
 - Pocos datos
 - Poco volumen
 - Rendimiento escaso
 - No permite acceso a varios usuarios
- El uso de ficheros no es productivo

10.- Persistencia de datos en bases de datos relacionales

- Bases de datos relacionales son las mas usadas
 - Modelo relacional :
 - Estructura básica : tabla
 - Lenguaje SQL
 - Creamos tabals
 - Establecemos relaciones
 - Imponemos restricciones
 - Formas normales: relaciones de dependencias y claves candidatas
 - (1FN) claves candidatas no son multivaluados
 - (2FN) campo no principal depende de forma completa de la clave
 - (3FN) Ningún campo no principal depende transitivamente de una clave candidata
 - (Boyce-Cod) Todo campo depende de la clave candidata

10.- Persistencia de datos en bases de datos relacionales

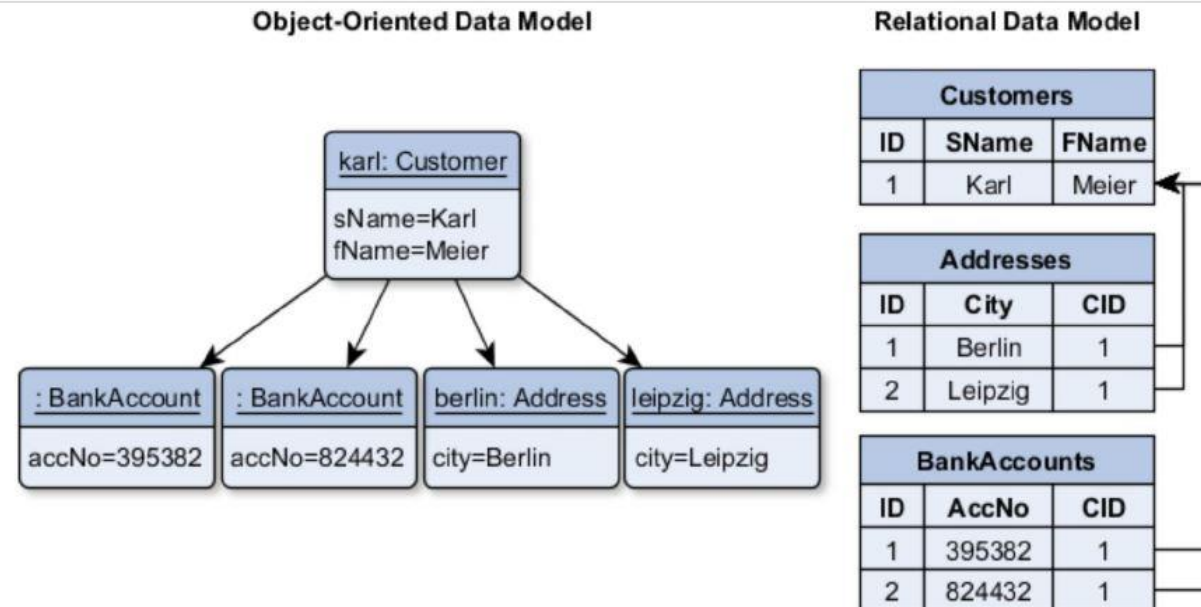
- SQL lenguaje de alto nivel
 - Permite especificar consultas para obtener datos
 - El propio sistema las gestiona
- Bases de datos actuales son:
 - Escalables
 - Permiten gestionar pequeños y grandes volúmenes de datos
 - Permiten operaciones sobre bloques de datos
 - Gestionan transiciones
 - Mecanismos de recuperación y copia ante fallos

11.- Persistencia de datos de XML en bases de datos relacionales

- XML : formato de datos muy utilizado
- Mecanismo SQL/XML
 - Tipos de datos XML
 - Permite extraer datos:
 - Tablas
 - Ficheros XML
- XML-enabled o bases de datos capacitadas para XML

12.- Persistencia de objetos en bases de datos relacionales

- Dificultad de representar un objeto en un modelo relacional
 - Complejidad en los objetos
 - Desfase entre objeto – modelo relacional
 - Problemas con la persistencia



12.- Persistencia de objetos en bases de datos relacionales

- Soluciones:
 - Bases de datos Objeto – relacionales
 - Bases de datos relacionales con capacidad de manejar objetos
 - Oracle
 - PostgreSQL
 - Correspondencia Objeto – relacional
 - Mapean objetos relacionales
 - Solución mas flexible
 - Hibernate

13.- Persistencia de bases de datos en bases de datos de objetos

- Bases de datos orientadas a objetos permiten guardar **directamente** los objetos en la base de datos.
- Su uso es limitado
 - No existe un modelo formal y aceptado
 - Falta de estándares

14.- Persistencia de datos en bases de datos XML Nativas

- Bases de datos nativas de XML están diseñadas y optimizadas para trabajar con XML
 - Consultas : Xpath y Xquery
 - Actualizaciones (Extensión de Xquery)
 - Transformaciones (XSL)
 - Validación (XML Schema y DTD)
- Xquery for java : implementación del API estándar para bases de datos XML
- No soportan transiciones :
 - El almacenamiento de los datos es en documentos enteros

15.- Persistencia de datos en cases de datos NoSQL

- BD NoSQL :
 - Nacen de la necesidad de gestionar grandes volúmenes de datos (Big Data)
 - Deben permitir el acceso ágil y rápido a gran volumen de datos
- Denominados BD NoSQL cuando no cumple:
 - Almacenamiento en forma de tablas
 - SQL como lenguaje
 - Soporte de restricciones de integridad y transiciones
- Características:
 - Almacenamiento a través de estructuras flexibles
 - Arrays asociados, mapas, documentos json,
 - No usan SQL como lenguaje de consultas
 - Lenguajes propios que especifican que conseguir no como)
 - Transiciones y persistencia
 - BASE (Basic availability, soft state eventual consistency)
 - Disponibilidad básica
 - Estado flexible
 - Consistencia en tiempo