

* BIEN FORMADA \Rightarrow toda lectura y escritura tienen lock y unlock (después del acceso).

* PROTOCOLO LOCK EN 2 FASES \Rightarrow todos los bloqueos (lock) preceden a todos los desbloques (unlock):

1^a FASE: fase de crecimiento

2^a FASE: fase de contracción

NIVELES

- Nivel 0 \Rightarrow anarquía
- Nivel 1 \Rightarrow Bien formado respecto a escrituras y 2 fases para escritura.
- Nivel 2 \Rightarrow 2 fases para escritura y bien formado
- Nivel 3 \Rightarrow 2 fases y bien formado.

OBTENER ELEMENTOS ESPECÍFICOS DEL ÁRBOL DOM

document.getElementById("id")

document.getElementsByTagName("tag")

document.getElementsByClassName("class-name")

MÉTODOS PARA NODOS:

createElement createTextNode createAttribute

insertBefore

removeChild, removeAttribute

appendChild

ATRIBUTOS PARA ACCEDER/MODIFICAR HTML DEL NODO

innerHTML, outerHTML

innerText, outerText

setAttribute("class-name", estilo)

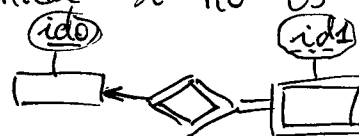
MODELO ENTIDAD-RELACIÓN

ELIPSES: atributos

- ELIPSES DOBLES: atributos multivaluados
- ELIPSES PUNTEADAS: atributos derivados
- SUBRAYADO: clave primaria

RELACIÓN ISA: relación entre dos entidades para indicar que una de ellas es un subconjunto de la otra.

ENTIDADES DÉBILES: entidad que no está bien definida si no es con relación a otra entidad (entidad fuerte)



MONGODB

Mostrar base de datos: show dbs

Creación y utilización de bases de datos: use nombre-db

Borrado de bases de datos en uso: db.dropDatabase

Mostrar colecciones: show collections

Crear colección en una base de datos en uso: db.createCollection(nombre, opciones)

Insertar un documento en una colección: db.nombre-collec.insert(nombre-doc)

Eliminar collec: db.nombre-collec.drop()

db.nombre-collec.insertOne(documento)

Búsqueda: db.nombre-collec.find(-)

db.nombre-collec.insertMany(documentos)

Actualizar documentos: db.nombre-collec.update(criterio-selección, datos)

Actualizar documento completo: db.nombre-collec.save({_id: ObjectId(), datos})

Eliminar documentos de una colección: db.nombre-collec.remove(criterio-selección)

Ordenar: db.nombre-collec.find(-).sort({key: i}) $i = -1$ (desc) $i = 1$ (asc)

Creación índices: db.nombre-collec.ensureIndex({key: i})

Agregación en la búsqueda: db.nombre-collec.aggregate(op-aggr)

ESTRATEGIAS DE ACCESO BD

- Directo / Constante (CONST.)
 - tablas con un solo registro
 - por valor en índice
- Cruce por clave única (EQ-REF)
- Clave no única (REF)
- Merge de índice (INDEX-MERGE)
- Clave única en subconsulta (UNIQUE-SUBQUERY)
- Clave no única en subconsulta (INDEX-SUBQUERY)
- Rango en índice (RANGE)
=, <, >, <=, >=, BETWEEN, LIKE, IN, IS NULL
- Full index scan (INDEX)
- Full table scan, secuencial (ALL)