Teórico

martes, 12 de diciembre de 2023 15:25

1

Definir DB → Especificar tipos de datos, estructuras y restricciones.

Construir DB → Almacenar en algún lado con DBMS

DBMS → Ayuda a definir, construir y mantener BDs.

- recuperación e integridad.
- Acceso a datos.
- Seguridad (restringir accesos)
- integridad de datos.
- Backups.

DDL → Da el esquema de la BD (Diccionario de datos)

DML → Recuperar, agregar, modificar y borrar data

Son componentes del DBMS

DML Procedimental → Hay q' decirle cómo conseguir los datos (ej. usando bucles)

2

Acceso concurrente → Abstracción

→ q' datos se almacenan y los recuperan el ellos.

Lógica → representación en una comp.

Físico → Cómo se almacenan archivos, hardware

Subconjunto → Jerarquía con solo 1 conjunto hijo.

↳ Sí o sí parcial exclusiva.

Atributos derivados

↳ info q' se puede obtener de otra forma.

- Nivel conceptual
  - Generico
  - Alejado del tipo de DBMS
  - Alejado del producto particular
- Nivel logico
  - Mas especifico
  - Orientado a un tipo de DBMS
  - Aljeado del producto particular
- Nivel fisico
  - Especifico
  - Orientado a un producto

3 Cosas del conceptual

→ ciclo de relaciones, atributos derivados.

Minimalidad → Cada cosa aparece 1 sola vez.

Expresividad → Cosas expresadas de manera natural, se entienden ex

Autoexplicación →

Legibilidad

Extensibilidad → fácil de cambiar.

Diferencias modelo lógico.

- Atributos derivados
- Atributos polivalentes
- Atributos compuestos
- Ciclo de relaciones
- Jerarquías

4

- Eliminación de identificadores externos
- Selección de claves
  - Primaria
  - Candidata
- Conversión de entidades
- Relaciones

Si entidad no tiene id interno, es débil

Clave foránea: atributo/s de una tabla que en otra tabla es/son CP y que sirven para establecer un nexo entre ambas estructuras

Integridad referencial

- Propiedad deseable de las BD
- Asegura que un valor que aparece para un atributo en una tabla, aparezca además en otra tabla
- Como se inscribiria un alumno en una materia??? (alta)
- Que otras operaciones podria haber??
- Eliminación
- Modificación

- Tipos de IR:
  - Restringir la operación
  - Realizar la operación en cascada
  - Establecer la clave Foránea en nulo

- No hacer nada

Restricciones de dominio

- Especifican que el valor de c/atributo A debe ser un valor atómico del dominio de A.

Restricciones de clave

- Evita que el valor del atributo clave genere valores repetidos

Restricciones sobre nulos

- Evita que un atributo tome *nulo* en caso de no ingresarle valor

Restricciones de integridad

- Ningún valor de la clave primaria puede ser nulo.

Restricción de integridad referencial

- Se especifica entre dos relaciones y sirve para mantener la consistencia entre tuplas de la dos relaciones
- Establece que una tupla en una relación que haga referencia a otra relación deberá referirse a una tupla existente en esa relación
- *Clave foránea* está representada por un atributo de una relación que en otra es clave primaria.

Las operaciones de Alta, Baja y Modificación (ABM) pueden generar violaciones a las restricciones anteriores.

- Alta
  - Puede violar: valor nulo para clave, repetición de la clave, integridad referencial, restricciones de dominio.
  - Si se viola la regla, la operación se rechaza
- Baja
  - Puede violar: integridad referencial (se procede como en el caso anterior)
- Modificación
  - Puede violar: cualquiera de las operaciones.

- **Total Exclusiva (T, E)**: Tres posibilidades, dejar todo, dejar sólo los hijos o dejar sólo al padre.
- **Total Superpuesta (T, S)**: Dos posibilidades, dejar todo o dejar sólo al padre. No se puede eliminar al padre.
- **Parcial Exclusiva (P, E)**: Dos posibilidades, dejar todo o dejar sólo al padre. No se puede eliminar al padre.
- **Parcial Superpuesta (P, S)**: Dos posibilidades, dejar todo o dejar sólo al padre. No se puede eliminar al padre.

4

Dependencia funcional → restricción e/ 2 conjuntos de atributos

si una restricción en R dice que no puede haber más de una tupla con un valor X en r (convirtiendo a X en clave primaria) entonces X → Y para cualquier Y de R
Si X → Y en R, no se puede afirmar ni negar que Y → X.
Cuando si y cuando no de esta afirmación???

- Empleado = (NroEmpl, Nombre, DNI, Sexo)
  - Nroempl → nombre
  - Nroempl → dni
  - Nroempl → sexo
  - DNI → nroempl??
    - Cuando si?
    - Que otras dependencias pueden surgir?

parcial → si se puede sacar la dependencia y sigue siendo

transitiva → A → B y B → C

Nro_empleado → nombre, posición, salario, nro_depto, nombre_depto
Nro_depto → nombre_depto.
En este ejemplo
A = nro_empleado
B = nro_depto
C = nombre_depto

completa → necesitas todo.

NORMALIZACIÓN → 1FN → 2FN → 3FN → BOYCE CODD → 4FN → 5FN.

1 → sin redundancia

② → Sin dependencia parcial → Sacar datos y ponerlos en otras tablas

③ → Sacar transitivas.

BC → Todo determinante es clave candidata.

Si no hay dep. Boyce Codd → sacar redundancia.

④ Boyce todo + solo dependencia multivaluada trivial → q' no de muchos valores = q' transitiva

a → b y b → c

⑤ Sin relaciones con dependencias de combinación.

⑥

Operaciones de uno o dos relaciones (tablas) de entrada que generan una nueva relación (tabla) como resultado

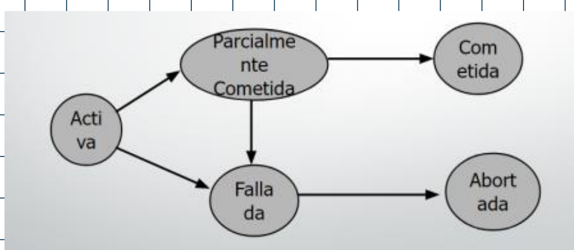
⋈ Producto Natural + pierda q' → Natural

X → inner Cartesiano → Todos con todos en SQL se usa ON (nombre = nombre)

⑦

FK ⋈ PK

⑧



Bitácora

- secuencia de actividades realizadas sobre la BD.
- Contenido de la bitácora
  - <T iniciada>
  - <T, E, Va, Vn>
    - Identificador de la transacción
    - Identificador del elemento de datos
    - Valor anterior
    - Valor nuevo
  - <T Commit>
  - <T Abort>

Modificación diferida

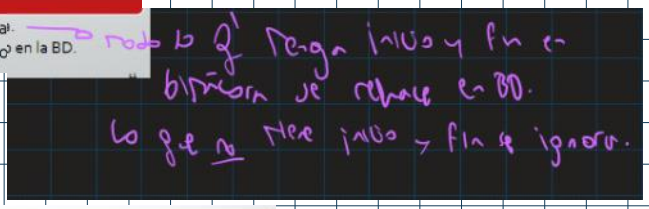
- Las operaciones write se aplazan hasta que la transacción esté parcialmente cometida, en ese momento se actualiza la bitácora y la BD

Recién con To parcialmente cometida, entonces se actualiza la BD.

- No se necesita valor viejo, se modifica la BD al final de la transacción o no se modifica.

Ante un fallo, y luego de recuperarse:

- REDO (Ti), para todo Ti que tenga un Start y un Commit en la Bitácora.
- Si no tiene Commit entonces se ignora, dado que no llegó a hacer algo en la BD.



Modificación inmediata:

- La actualización de la BD se realiza mientras la transacción está activa y se va ejecutando.
- Se necesita el valor viejo, pues los cambios se fueron efectuando.
- Ante un fallo, y luego de recuperarse:
  - REDO(Ti), para todo Ti que tenga un Start y un Commit en la Bitácora.
  - UNDO(Ti), para todo Ti que tenga un Start y no un Commit.

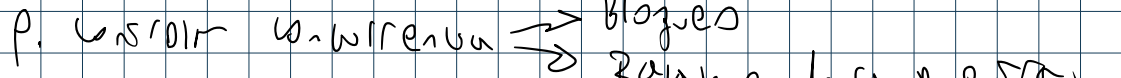
Ventajas:

- Elimina la sobrecarga de escrituras del log
- Recuperación más rápida (no existe el REDO o UNDO).

Desventajas:

- Sobrecarga en el compromiso: la técnica de paginación es por cada transacción.
- Fragmentación de datos: cambia la ubicación de los datos continuamente
- Garbage Collector: ante un fallo queda una página que no es mas referenciada.

El programa debe conservar la consistencia
La inconsistencia temporal puede ser causa de inconsistencia en planificaciones en paralelo
Una planificación concurrente debe equivaler a una planificación en serie
Solo las instrucciones READ y WRITE son importantes y deben considerarse



Bloqueo

- Compartido *Lock_c*(dato)(solo lectura)
- Exclusivo *Lock_e*(dato) (lectura/escritura)
- Las transacciones piden lo que necesitan.
- Los bloqueos pueden ser compatibles y existir simultáneamente (compartidos)

Protocolo basado en hora de entrada

- El orden de ejecución se determina por adelantado, no depende de quien llega primero
- C/transacción recibe una HDE
  - Hora del servidor
  - Un contador
- Si HDE(Ti) < HDE(Tj), Ti es anterior
- C/Dato
  - Hora en que se ejecutó el último WRITE
  - Hora en que se ejecutó el último READ
  - Las operaciones READ y WRITE que pueden entrar en conflicto se ejecutan y eventualmente fallan por HDE.