

Feladatok – 2023.05.31

I. rész

Definiálja a következő fogalmakat!

- Koncepcionális adatmodell
- Gyenge egyedtípus
- Atomikusság (ACID)
- Projekció
- Kulcs

II. rész

1. Definiálja az Armstrong-axiómákat, és azok közül bizonyítsa be a tranzitivitás szabályát!
2. Képezze le az alábbi ER diagramot a LÉGITÁRSASÁG adatbázisról relációs modellé. Indokolja is meg lépéseit

III. rész

1. Definiálja a program-adat függetlenséget!
2. Ismertesse a lehetséges integritási megszorítási sérüléseket!
3. Definiálja az uniókompatibilitás fogalmát és ismertesse a relációalgebra halmazműveleteit és azok tulajdonságait!
4. Sorolja fel az ER modell leképezésének lépéseit relációs modellé!
5. Ismertesse azokat a problémákat, amelyek tranzakciók egyidejű feldolgozásakor léphetnek fel!
6. Hogyan örököltethet felhasználó által definiált típust és mik ennek a tulajdonságai?
7. Definiálja a gyenge és az erős konzisztenciát NoSQL adatbázisokban

Megoldások – 2023.05.31

I. rész

Konceptcionális adatmodell

Véges számú tulajdonságtípussal megadott véges számú egyedtípus és a közöttük fennálló véges számú kapcsolattípus összessége.

A kapcsolattípus is rendelkezhet véges számú tulajdonságtípussal.

Gyenge egyedtípus

Azokat az egyedtípusokat, amelyek nem rendelkeznek saját kulcsattribútumokkal, gyenge egyedtípusnak nevezzük.

A gyenge egyedtípusoknak részleges kulcsuk (diszkriminátoruk) van, amely azon attribútumok halmaza, amelyek egyértelműen azonosítják azokat a gyenge egyedeket, amelyek ugyanazon tulajdonos egyed(ek)hez kapcsolódnak.

Atomikusság (ACID)

A tranzakció a feldolgozás atomi egysége; vagy teljes egészében végrehajtódik, vagy egyáltalán nem.



Projekció

Általános alakja: $\pi_{\langle \text{attribútumlista} \rangle}(R)$, ahol az $\langle \text{attribútumlista} \rangle$ az R reláció lekérdezni kívánt attribútumainak listája.

A projekció megvalósítása SQL-ben: $\text{SELECT attribútumlista FROM } R$; ahol az attribútumlista elemeit vesszővel választjuk el és a tulajdonos reláció azonosítására, ha több relációt sorolunk fel, alkalmazható a pontozott jelölés $R.A$

Kulcs

Egy R relációséma K kulcsa R -nek egy olyan superkulcsa, amelyből bármelyik A attribútumot elhagyva, az így kapott $K' = K \setminus A$ attribútumhalmaz már nem superkulcsa R -nek. Egy kulcs kielégíti a következő feltételeket:

-  Bármilyen relációt tekintve, a reláció két különböző rekordjának nem lehetnek azonosak a kulcsban szereplő attribútumokhoz tartozó értékek.
-  Minimális superkulcs, azaz egy olyan superkulcs, amelyből nem tudunk úgy eltávolítani egyetlen attribútumot sem, hogy az egyediségre vonatkozó feltétel továbbra is fennálljon.

II. rész

1. feladat

Armstrong-axióma definíciója

A reflexivitás, az augmentitívás és a tranzitivitás szabályait együtt Armstrong-Axiómának nevezzük. William Ward Armstrong 1974-ben bizonyította be, hogy a reflexivitás, az augmentitívás és a tranzitivitás szabálya együtt helyes és teljes.

Tranzitivitás szabálya

Tegyük fel, hogy (1) $X \rightarrow Y$ és (2) $Y \rightarrow Z$ fennáll egy r relációban. Ekkor tetszőleges t_1 és t_2 r -beli rekordokra, melyekre igaz, hogy $t_1[X] = t_2[X]$, (1) miatt kapjuk, hogy $t_1[Y] = t_2[Y]$; így (3)-ból és a (2)-es feltevésünkből azt is kapnunk kell, hogy $t_1[Z] = t_2[Z]$; ezért $X \rightarrow Z$ -nek fenn kell állnia r -ben.

2. feladat

III. rész

1. feladat

Program-adat függetlenség

A programok és adatok elkülönítését program-adat függetlenségnek nevezzük. Ez lehetővé teszi az adatszerkezetek és a tárolás módjának megváltoztatását anélkül, hogy a DBMS-t elérő programot meg kellene változtatni.

Logikai adatfüggetlenség

Annak képessége, hogy a koncepcionális séma anélkül változzon meg, hogy a külső sémának és a hozzájuk rendelt alkalmazói programoknak meg kellene változnia.

Fizikai adatfüggetlenség

Annak képessége, hogy a belső séma anélkül változzon meg, hogy a koncepcionális sémának meg kellene változnia. Pl. a belső séma megváltozhat anélkül, hogy bizonyos fájl szerkezeteket átszervezünk vagy új indexeket hozunk létre az adatbázis hatékonyságának a javítása miatt.

Alacsonyabb, magasabb szintű változás

Amikor egy alacsonyabb szintű séma megváltozik, akkor csak ez és az eggyel magasabb szintű sémák közötti leképezésnek kell változnia.

A magasabb szintű sémák változatlanok maradnak. Ezért az alkalmazói programoknak nem szükséges módosulniuk, mivel azok a külső sémákra hivatkoznak.

2. feladat

Integritási megszorítási sérülések

INSERT művelet

- ✎ tartomány megszorítás, ha az új rekord egyik attribútum értéke nem a megadott tartományba esik
- ✎ kulcs megszorítás, ha az új rekord kulcs attribútum értéke már létezik a reláció egy másik rekordjánál
- ✎ hivatkozási integritás, ha a külső kulcs érték az új rekordban egy olyan elsődleges kulcs értékre hivatkozik, amelyen nem létezik a hivatkozott relációban
- ✎ egyedintegritás, ha az elsődleges kulcs érték NULL az új rekordban

DELETE művelet

Csak hivatkozási integritási megszorításnál okozhat sérülést, ha olyan elsődleges kulcs értékkel bíró rekordot törölünk, amelyre más relációból hivatkozás van.

UPDATE művelet

A tartomány megszorítás és a NULL érték megszorítás sértheti meg.

3. feladat

Uniókompatibilitás fogalma

Az $R(A_1, A_2, \dots, A_n)$ és $S(B_1, B_2, \dots, B_n)$ relációkat egymással uniókompatibilisnek (típuskompatibilisnek) mondjuk, ha azonos fokszámuk és $\text{dom}(A_i) = \text{dom}(B_i)$ minden $1 \leq i \leq n$ esetén. Azaz az uniókompatibilitás azt jelenti, hogy a két relációnak ugyanannyi attribútuma van, és attribútumaik tartományai páronként megegyeznek egymással.

Relációalgebra halmazműveletei és tulajdonságai

Halmazműveletek

- ✎ unió $R \cup S$: azok a rekordok, amelyek szerepelnek valamelyik relációban.
- ✎ metszet $R \cap S$: azok a rekordok, amelyek mindkét relációban szerepelnek.
- ✎ különbség $R - S$: azok a rekordok, amelyek szerepelnek az első relációban de nem szerepelnek a másodikban

Tulajdonságok

Az unió, a metszet és a különbség bináris műveletek. Az eredményül kapott reláció sémája – megállapodás szerint – az első (R) reláció sémájával egyezik meg.

Az unió és a metszet műveletek kommutatívak: $R \cup S = S \cup R$ és $R \cap S = S \cap R$.

Az unió és a metszet műveletek asszociatívak: $R \cup (S \cup T) = (R \cup S) \cup T$ és $R \cap (S \cap T) = (R \cap S) \cap T$.

A különbség művelet általában nem kommutatív: $R - S \neq S - R$.

4. feladat

ER modell -> relációs modell

1. Erős egyed típusok leképezése
2. Gyenge egyed típusok leképezése
3. Bináris 1:1 számosságú kapcsolattípusok leképezése
 - (a) külső kulcs használata
 - (b) összevonás
 - (c) kereszthivatkozás v. kapcsoló reláció használata
4. Bináris 1:N számosságú kapcsolattípusok leképezése
5. Bináris M:N számosságú kapcsolattípusok leképezése
6. Többértékű attribútumok leképezése
7. N-edfokú kapcsolattípusok leképezése

5. feladat

Problémák amelyek tranzakciók egyidejű feldolgozásakor léphetnek fel

Elvesztett frissítés

Akkor fordul elő, amikor két tranzakció, amely ugyanazokat az adatbázis elemeket éri el, úgy fészülődik össze, hogy egyes adatbázis elemek hibásakká válnak.

Időleges frissítés (dirty read)

Akkor fordul elő, amikor egy tranzakció frissít egy adatbázis elemet, ami után valamilyen oknál fogva a tranzakció hibásan fejeződik be. Ezt a frissített elemet más tranzakció is eléri mielőtt az még visszaállna az eredeti értékre.

Helytelen összegzés

Amikor egy tranzakció rekordok egy összegző függvényét számolja, amíg egy másik tranzakció ezen rekordok közül néhányat frissít. Ekkor az összegző függvény olyan értékekkel számolhat, amelyek még a frissítés előtt vannak, míg mások már a frissítés után.

6. feladat

Felhasználó által definiált típus célja

Összetett szerkezetű (a relációs modell rekordjainál bonyolultabb) objektumok létrehozása. Egy típus deklarációjának elválasztása a tábla (reláció) létrehozásától. Rekord típusú konstruktor a ROW kulcsszóval rekord típusú attribútumok létrehozására.

4 féle kollekció típus: ARRAY, MULTiset, LIST és SET

Öröklődés

SQL-ben UNDER kulcsszó alatt működik

A NOT FINAL kulcsszót kell használni ha egy UDT-nek további altípusát szeretnénk deklarálni.

Minden attribútum öröklődik.

A szupertípusok sorrendje az UNDER kulcsszó után határozza meg az öröklődési sorrendet.

Egy altípus példánya minden olyan kontextusban használható ahol szupertípusának példánya használható.

Egy altípus minden a szupertípuson definiált függvényt újradefiniálhat feltéve hogy a szignatúra nem változhat.

Egy függvény hívásakor a legjobban illeszkedő implementáció kerül alkalmazásra az összes argumentum típusát figyelembe véve.

Dinamikus kötéskor a paraméterek futáskori típusait veszi figyelembe

7. feladat

Gyenge és erős konzisztencia NoSQL-ben

Gyenge konzisztencia

A rendszer nem garantálja, hogy a későbbi hozzáférések a módosított értéket adják eredményül. Több feltételnek is teljesülnie kell, mielőtt az érték visszaadásra kerül. A módosítás megtörténte és azon pillanat közötti időszakot, amelyre már garantált, hogy minden megfigyelő mindig a módosított értéket látja, inkonzisztencia ablaknak (inconsistency window) nevezzük.

Erős konzisztencia

Miután a módosítás végrehajtódott, minden (akár A, akár B, akár C által végzett) hozzáférés a módosított értéket adja eredményül.