

Informatika 2 vizsgadolgozat, 2013. jún. 11.

1. Milyen tranzakciós izolációs alapproblémákat ismer? Ismertesse őket röviden!

- Elveszett módosítás: Két párhuzamos módosítás, egyik elveszik.
- Piszkos olvasás: A tranzakció egy másik tranzakció egy nem commitált adatait használja, pedig előfordulhat rollback
- Nem megismételhető olvasás: A lekérdezés eredménye függ attól, hogy mikor adták ki egy tranzakcióban
- Fantom rekord: kb =nem megismételhető olvasás, de rekord szinten: A lekérdezés több vagy kevesebb rekordot ad vissza a végrehajtás idejétől függően

2. Írjon SQL lekérdezést, amely ABC sorrendben megjeleníti azon ügyfelek nevét és ID-ját, akiknek van olyan folyószámlája, aminek negatív az egyenlege!

Client(Id INT, name NVARCHAR(100), CreatedAt DATETIME, IsActive BIT)

Account(Id INT, Number CHAR(26), ClientId INT, Balance INT, CreatedAt DATETIME)

Összekapcsoljuk a két táblát úgy, hogy csak azok(!) legyenek felsorolva (egyszer) akiknek van folyószámlája, szűkítünk Balance alapján, majd sorba rakjuk:

```
SELECT DISTINCT Client.Id, Client.Name FROM  
Client INNER JOIN Account ON Client.Id=Account.ClientId  
WHERE Account.Balance<0  
ORDER BY name ASC
```

3. Írjon SQL utasításokat amelyek kitörlik valamennyi inaktív (IsActive=0) ügyfél valamennyi folyószámláját és magát az ügyfelet is az adatbázisból!

Összekapcsoljuk a két táblát, majd töröljük mindkettőből azokat a sorokat, ahol Client.IsActive=0

```
DELETE Client, Account FROM  
Client INNER JOIN Account ON Client.Id=Account.ClientId  
WHERE Client.IsActive=0
```

Alternatív, két utasításos megoldás:

```
DELETE FROM Account WHERE ClientId = (SELECT Id FROM Client WHERE IsActive=0);  
DELETE FROM Client WHERE IsActive=0;
```

4. Funkcionális függőség axiómái (Armstrong) (Szabó N. megoldása)

- Reflexivitás: Ha R reláción Y valódi részhalmaza X, akkor $X \rightarrow Y$
- Tranzitivitás: Ha R reláción $X \rightarrow Y$ és $Y \rightarrow Z$, akkor $X \rightarrow Z$
- Bővíthetőség: Ha R reláción $X \rightarrow Y$, akkor $XZ \rightarrow YZ$

Pszudotranzitivitás bizonyítása:

$X \rightarrow Y$ és $YW \rightarrow Z$, akkor $XW \rightarrow Z$

1. $X \rightarrow Y \rightarrow XW \rightarrow YW$ (bővíthetőség)
2. $XW \rightarrow YW$ és $YW \rightarrow Z \rightarrow XW \rightarrow Z$ (tranzitivitás)

5. Ismertesse a nyelvek osztályozásának Chomsky féle hierarchiáját! Adjon példát az egyes nyelvosztályokba tartozó nyelvekre!

3. Reguláris (szabályos) nyelvek

Pl. változók, számok, kulcsszavak (tokenizáló).

$S \rightarrow a \mid aS$

$a^i \quad i > 0$

2. Környezetfüggetlen (context free) nyelvek

Pl. aritmetikai kifejezések, egymásba ágyazható szerkezetek

$S \rightarrow ab \mid aSb$

$a^i * b^i \quad i > 0$

1. Környezetfüggő nyelvek (context sensitive)

$\beta A \gamma \rightarrow \beta \alpha \gamma$

$a^i * b^i * c^i \quad i > 0$

0. Rekurzívan felsorolható nyelvek

Nincs megkötés.

$\alpha \rightarrow \beta$

6. E-mail címre reguláris kifejezés (Eke M. megoldása)

$^([a-z0-9-]+\backslash.)*[a-z0-9-]+\@([a-z0-9-]+\backslash.)*[a-z]{2,4}\$$

7. Adott az alábbi nyelvtan:

$S \rightarrow aSa \mid bSb \mid aa \mid bb$

Elemezze a „bbaabb” szöveget Cocke-Younger-Kasim (CYK) algoritmussal!

Példa: Legyen nyelvtanunk:

$S \rightarrow aSa \mid bSb \mid aa \mid bb$

Hozzuk Chomsky normál formára:

$S \rightarrow AS_A \quad S \rightarrow BS_B \quad S \rightarrow AA \quad S \rightarrow BB$

$S_A \rightarrow SA \quad S_B \rightarrow SB$

$A \rightarrow a \quad B \rightarrow b$

Elemezzük az

aabb

szöveget CYK algoritmussal:

S					
	S_A				
	S				
S_B		S_A			
S		S		S	
A	A	B	B	A	A
a	a	b	b	a	a

A fenti táblázat alapján látható, hogy a kiinduló szövegből a fenti példában csak egyféleképpen lehet eljutni a mondat-szimbólumig. Általános esetben azonban a táblázat bármely négyzetében több nemterminális is előfordulhat.

Bővebben a megoldás menetéről (angol nyelven):

<https://www.youtube.com/watch?v=b98Uyj7JHIU>

8. Mutassa be, hogyan történik a TCP protokoll esetén a kapcsolat felépítése és bontása: pontosan milyen tartalmú üzeneteket küldenek egymásnak a kommunikáló felek? (típus, sorszám, nyugta). (Szabó N. megoldása)

Nyitás:

- Kliens kiválasztja a kezdő sorszámot (x), TCP SYN üzeneteket küld (SYNbit=1, Seq=x)
- Szerver nyugtáz, TCP SYNACK, kiválasztja a kezdősorszámot (y) (SYNbit=1, Seq=y, ACKbit=1, ACKnum=x+1)
- Kliens megkapja SYNACK(x)-et, nyugtáz (ACKbit=1, ACKNUM=y+1)
- Szerver megkapja ACK(y)-t, már adat is jöhet

Lezárás (EZ NEM KELLŐEN RÉSZLETES! ELŐADÁSDIÁK!):

- Kliens TCP szegmenst küld, amelyben FIN bit=1
- Szerver nyugtázza az ünetetet (ACK), elküldi saját FIN üzenetét
- Kliens nyugtáz (ACK=1)

9. Csomagkésleltetés összetevői (4)

$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$

d_{proc} : feldolgozási késleltetés (fejrész vizsgálata, bitszintű hibák keresése)

d_{queue} : sorbanállási késleltetés (kimeneti összeköttetésre várás)

d_{trans} : átviteli késleltetés ($=L/R$ =bitek/átvitel)

d_{prop} : jelterjedési késleltetés ($=d/s$ = kábelhossz/sebesség)

10. Datagram mérete: 7000byte, maximális átvihető (MTU): 1500 byte. (Szabó N. megoldása)

IPv4 feldarabolja

20 byte fejléc -> 6980 byte adat

fej 20, adat 1480

$1480/8 = 185$

hossz = 1500	ID=x	fragflag = 1	eltolás = 0
hossz = 1500	ID=x	fragflag = 1	eltolás = 185
...	...	...	...
hossz = 1080	ID=x	fragflag = 0	eltolás = 740

IPv6: nem darabol és küld egy ICMP6 üzenetet, hogy Packet too large, és ebből tudja a kliens, hogy az a csomag elveszett