

БАЗИРАНИ ПОДАТОЧКИ

2 КОЛОКВИУМ ТЕОРИЈА

Елегантен спидран за бка од **мацо <3**

Ако има некој пропуст тагнете ме во чет каналот за БнП
Најверојатно нема да ви одговорам ама вреде да се проба

Како и секогаш не сум крив ако паднеш
Viewer discretion is advised (срамотна содржина)

Олрајт лез гоу

1) Физичка организација, индекси и Вовед во SQL: DDL

Задача 1:

Важно: Научи ги на памет

ПРАШАЊЕ 1

Со кој од следните резервирали зборови **не дефинираме** ограничување во DDL изразите?

- NULL✓
- UNIQUE
- CHECK
- PRIMARY KEY

Mark 1.00 out of 1.00

ПРАШАЊЕ 2

Кое од наведените тврдења е точно?

- Со CHECK креираме ограничување на доменот
- Со UNIQUE дефинираме единственост на вредностите за множество на атрибути
- Атрибут за кој е дефинирано UNIQUE ограничување не може да има иста вредност за било кои две редици од табелата
- Сите претходно наведени тврдења се точни✓

Mark 1.00 out of 1.00

ПРАШАЊЕ 3

Кое од наведените ограничувања не наложува единственост на атрибутите за кои е дефинирано?

- UNIQUE
- PRIMARY KEY
- FOREIGN KEY✓
- Ниту едно од претходно наведените ограничување не наложува единственост

Mark 1.00 out of 1.00

ПРАШАЊЕ 4

Кое од наведените не е точно?

(одберете ги сите за кои важи)

- Ограничивање на ниво на атрибут се специфицира како дел од дефиницијата на атрибутот и важи само за тој атрибут
- Ограничивањето PRIMARY KEY дозволува NULL како една од уникатните вредности на клучот✓
- Кога во рамки на едно ограничување треба да се вклучи повеќе од еден атрибут се дефинира ограничување на ниво на табела кое може да биде именувано
- Ограничивање на ниво на табела се специфицира независно од дефинициите на атрибутите и може наеднаш да се однесува на повеќе од еден атрибут

Mark 1.00 out of 1.00

ПРАШАЊЕ 5

Кое од следните ограничување може да се искористи само еднаш во дефиницијата на една табела?
(одбери ги сите кои важи)

- NOT NULL
- PRIMARY KEY ✓
- CHECK
- FOREIGN KEY

Mark 1.00 out of 1.00

Задача 2:

Важно: Ако те прашат за **UPDATE** и **INSERT** сите ги заокружеш што се погоре,
НО ако те прашат за **DELETE** само заокружеш **Референцијален интегритет**

Кои ограничувања во една релациона база на податоци може да бидат нарушени при извршување на операцијата **INSERT**?

Select one or more:

- Ограничивање (единственост) на клучеви
 - ✓ On inserting a value in the new tuple of a relation which is already existing in another tuple of the same relation, can cause violation of Key Constraints.
- Ентитетен интегритет
 - ✓ On inserting NULL values to any part of the primary key of a new tuple in the relation can cause violation of the Entity integrity constraint.
- Референцијален интегритет
 - ✓ On inserting a value in the foreign key of relation 1, for which there is no corresponding value in the Primary key which is referred to in relation 2, in such case Referential integrity is violated.
- Ограничивање на доменот
 - ✓ Domain constraint gets violated only when a given value to the attribute does not appear in the corresponding domain or in case it is not of the appropriate datatype.

Your answer is correct.

Задача 3:

Важно: Која од следните наредби **НЕ МОЖЕ** да се изврши!

Нека се дадени табелите R(A, B) и S(A, B, C) и DDL изразите за нивно креирање:

```
CREATE TABLE R(
    A INT,
    B INT,
    PRIMARY KEY(A,B));

CREATE TABLE S(
    A INT,
    B INT,
    C INT,
    PRIMARY KEY (A,B,C),
    FOREIGN KEY (A,B) REFERENCES R(A,B));
```

Ако се извршени следните наредби:

```
INSERT INTO R VALUES (1,1),(1,2),(2,1),(2,2);
INSERT INTO S VALUES (1,1,1),(2,2,2);
```

Која следните наредби **не може** да се изврши?

Select one or more:

- INSERT INTO R VALUES (3,3);
- DELETE FROM R WHERE A=1; ✓
- UPDATE R SET B=B+1; ✓
- DELETE FROM S WHERE A=1;
- UPDATE S SET C=C+1;
- INSERT INTO S VALUES (3,3,3); ✓

Your answer is correct.



1. DELETE FROM R WHERE A=1; -> **не може**, бидејќи кај табелата S имаме FOREIGN KEY кој референцира кон табелата R(A,B)
2. UPDATE R SET B=B+1; -> **не може**, бидејќи кај табелата S имаме FOREIGN KEY кој референцира кон табелата R(A,B)
3. INSERT INTO S VALUES (3,3,3) -> -> **не може**, бидејќи во табелата R немаме А и Б што имат вредност 3.

Задача 4:

Важно: НЕМА ДА РЕЗУЛТИРА во целосно испразната табела С

Тип: Ќе треба да црташ за оваа задача за да не се испомешаш.

Дадени се SQL изразите за креирање на три табели А, В и С:

```
CREATE TABLE A(e INT PRIMARY KEY, f INT);
CREATE TABLE B(c INT PRIMARY KEY, d INT REFERENCES A(e) ON DELETE CASCADE);
CREATE TABLE C(a INT PRIMARY KEY, b INT REFERENCES B(c) ON DELETE CASCADE);
```

Претпоставете дека табелите ги содржат следните токки:

A	
e	f
5	6
4	3
3	5
2	1
1	0

B	
c	d
5	4
4	1
3	3
2	2
1	1

C	
a	b
0	2
1	2
2	1
3	5
4	4

Која од следните наредби, после сите акции за одржување на интегритетот, **нема да резултира** во целосно испразната табела С?

Select one or more:

- DELETE FROM A WHERE e < 5
- DELETE FROM A WHERE (e - f) > 0
- DELETE FROM A WHERE (e - f) = 1
- DELETE FROM A WHERE (e+f) >= 7



Your answer is correct.

Во табелата С: **b** референцира кон табелата В во **c**.

Во табелата В: **d** референцира кон табелата А во **e**.

Одеме наредба по наредба:

1. DELETE FROM A WHERE e < 5 -> **НЕ**
 - a. Ако се изврши оваа наредна редиците 4,3,2,1,0 во табелата А ќе се избришат бидејќи се помали од 5.
 - b. Бидејќи во табелата В **d** референцира кон А во **e** сега гледаме кои вредности се афектирани од оваа наредба.
 - c. 4,1,3,2,1 овие редици во табелата В ќе се избришат бидејќи сите се присутни во А.
 - d. Не мора да гледаме натаму бидејќи чим сите редици се избришани во табелата В очигледно дека и во табелата С ќе се избришат.

2. DELETE FROM A WHERE ($e - f > 0$) \rightarrow **НЕ**
 - a. Ако се изврши оваа наредна редиците 4,2,1 во табелата А ќе се избришат бидејќи ($e - f > 0$).
 - b. Во Табелата В ќе се избришат 4,1,2,1
 - c. Во Табелата С ќе се избришат 2,2,1,5,4 тоа се сите редици
 - d. Внимавај во кои **колони** гледаш!
3. DELETE FROM A WHERE ($e - f = 1$) \rightarrow **НЕ**
 - a. Ако се изврши оваа наредна редиците 4,2,1 во табелата А ќе се избришат бидејќи ($e - f = 1$).
 - b. Исто како претходната наредба
4. DELETE FROM A WHERE ($e + f \geq 7$) \rightarrow **ДА**
 - a. Ако се изврши оваа наредна редиците 5,4,3 во табелата А ќе се избришат бидејќи ($e + f \geq 7$).
 - b. Во В ќе се избришат 4,3
 - c. Во С ќе се избришат 5,2,2
 - d. Не е целосно испразнета бидејќи редиците 1,4 не се избришани!

Задача 5:

Важно: Ова задача се решава со формулите:

Изминување на индексот до листовите: **висина + 1**

Изминување на сите листови: **(2d + 1)^висина - 1**

Пристап до записите: **(2d+1)^висина * 2d**

$$1. \text{ висина} + 1 = 2 + 1 = 3$$

$$2. (2d + 1)^{\text{висина}} - 1 = (2 * 3 + 1)^2 - 1 = 48$$

$$3. (2d+1)^{\text{висина}} * 2d = (2 * 3 + 1)^2 * 2 * 3 = 294$$

Дадена е релацијата Colleges чии записи се зачувани во датотека, но за побрз пристап до записите на релацијата е креиран индекс над атрибутот *happiness*.

Индексот е некластерирачки , организиран како B+ стебла Алтернатива 2, со степен d=3 и висина 2.

```
CREATE TABLE Colleges {
    name varchar[255] PRIMARY KEY,
    ranking int,
    city varchar[255],
    state varchar[255],
    size int,
    happiness int
}
```

Да се претпостави дека сите јазли во индексот (внатрешните и листовите) се максимално пополнети.

Да се процени бројот на пристапи потребен за изминување на сите записи според атрибутот *size*? Одредети ги проценките за секој чекор од изминувањето!

(Совет: одредете го бројот на листови и бројот на записи во листовите бидејќи влегуваат во пресметките)

За изминување на индексот (до листовите) потребни се:

За изминување на листовите потребни се:

За пристап до сите записи во датотеката потребни се:

Задача 6:

ПРАШАЊЕ 1:

Кога е INSERT, UPDATE, DELETE секојпат е без индекс

ПРАШАЊЕ 2:

Кога е SELECT имаш 3 опции да биде:

1. Кластерирачки индекс Алтернатива 1

- Ова скоро никојпат не е точно (не се замарај шефе/шефице) ?

2. Кластерирачки индекс Алтернатива 2

- Ова е точно доколку нема дадено опсег

3. Кластерирачки индекс Алтернатива 3

- Ова е точно доколку има дадено опсег од до каде се движат

вредностите [x-y]

Прашање 1

Кој од понудените одговори е соодветно решение за индексна структура доколку се знае дека се очекува значително голем број на внесувања на нови записи во релацијата Users (пример како со следниот SQL израз):

```
INSERT INTO Users (name) VALUES (David);
```

Перформансите на операцијата INSERT би се подобрile:

- со кластерирачки индекс Алтернатива 2
- со некластерирачки индекс Алтернатива 2
- без индекс✓
- со кластерирачки индекс Алтернатива 3
- со индекс Алтернатива 1

Mark 1.00 out of 1.00

Прашање 2

Кој од понудените одговори е соодветно решение за подобрување на перформансите на следниото SQL прашање, доколку оценките (score) се движат од [1-12] и тековната состојба на базата содржи по 50 напреварувачи за секоја оценка.

```
SELECT * FROM Competitors WHERE score >= 10;
```

Перформансите на даденото SQL прашање би се подобрile:

- со кластерирачки индекс Алтернатива 3✓
- со некластерирачки индекс Алтернатива 2
- со кластерирачки индекс Алтернатива 2
- без индекс
- со индекс Алтернатива 1

Mark 1.00 out of 1.00

2) Вовед во SQL: DML

Задача 1:

A	B	C
1	1	3
1	2	3
2	1	4
null	3	5
2	4	1
3	2	4
null	3	6

Колку торки ќе има во резултатот од извршувањето на следниве SQL прашања над табелата R?
(Во соодветното поле внесете нумеричка вредност.)

1) SELECT * FROM R WHERE C>4

✓ торки

2) SELECT * FROM R WHERE B BETWEEN 2 AND 3

✓ торки

3) SELECT * FROM R WHERE C IN (2,4,6,8,10)

✓ торки

4) SELECT * FROM R WHERE C NOT IN (2,4,6,8,10) AND A IS NOT NULL

✓ торки

5) SELECT * FROM R WHERE C IN (SELECT A FROM R WHERE A<>3)

✓ торки

6) SELECT * FROM R R1 WHERE A IN (SELECT C FROM R R2 WHERE R1.A=R2.A)

✓ торки

7) SELECT * FROM R R1 WHERE C IN (SELECT C FROM R R2 WHERE R2.A IS NOT NULL AND R1.A<>R2.A)

✓ торки

Оваа задача се решава со читање на командата и пишување колку торки(редици) ќе се селектират

- 1) Во колоната С има само **2** торки што се поголеми од 4: 5 и 6
- 2) Во колоната В има само **4** торки што се измеѓу 2 и 3: 2,3,2,3
- 3) Во колоната С има само **3** торки што се (2,4,6,8,10): 4,4,6

- 4) Обратните од претходната команда и плус гледаш да немат NULL во колоната A, тоа се 3 торки: 3,3,1
- 5) Мало објаснување за ова:
- Прво го гледаш вгнездениот SELECT
 - $\text{SELECT } * \text{ FROM R WHERE C IN } (\text{SELECT A FROM R WHERE A} <> 3)$
 - Ова се претвара во $\text{SELECT } * \text{ FROM R WHERE C IN } (1,2,\text{null})$
 - Исто како претходно сега се селектира само 1 торка: 1
- 6) $\text{SELECT } * \text{ FROM R R1 WHERE A IN } (\text{SELECT C FROM R R2 WHERE R1.A=R2.A})$
- 0 торки и ќе ти кажам зошто само не ме пукай
 - Како претходната стратегија прво го разработуваме вгнездениот селект мјај
 - Мора да правеме копи пејст помошна табела бидејќи кучето селектира од P и P2
 - Најубаво нацртај си tabela

The diagram shows two identical tables, R and R, side-by-side. Both tables have columns A, B, and C. The first table (R) has rows labeled 1, 2, null, 2, 3, and null. The second table (R) also has rows labeled 1, 2, null, 2, 3, and null. Arrows point from the first row of the first table to the first row of the second table, indicating a self-join where the condition R1.A = R2.A is met.

R	A	B	C
1	1	3	
2	2	3	
null	3	5	
2	4	1	
3	2	4	
null	3	6	

R	A	B	C
1	1	3	
2	2	3	
null	3	5	
2	4	1	
3	2	4	
null	3	6	

- e)
- Одеме редица по редица
 - Тражеме каде A од P1 = A од P2
 - Во овај случај бидејќи табелата е идентична тоа значи сите редици се еднакви
 - Сега ја споредува С колоната со A колона, ако е еднаква тоа е 1 торка
 - Никаде не еднакво значи 0 торки

- 7) `SELECT * FROM R R1 WHERE C IN (SELECT C FROM R R2 WHERE R2.A IS NOT NULL AND R1.A<>R2.A)`
- Иста постапка како погоре само сега А на Р2 да не е NULL и А од Р1 да е различна тоест да не е еднаква со А на Р2
 - Има само 2 торки

Задача 2:

Нека е дадена табелата R(A, B, C), изразите за креирање на погледите V1, V2 и V3 над табелата R.

R		
A	B	C
1	1	3
1	2	3
2	1	null
null	3	5
2	4	1
3	2	3
null	3	6

`CREATE VIEW V1 as
SELECT COUNT(*) as A1, COUNT(A) as A2,
COUNT(DISTINCT A) as A3, COUNT(B) as A4,
COUNT(C) as A5
FROM R`

`CREATE VIEW V2 as
SELECT C, COUNT(*) as BR
FROM R
GROUP BY C`

`CREATE VIEW V3 as
SELECT B, C, COUNT(A) as D, SUM(A) as E
FROM R
GROUP BY B, C`

Прашање 1:

Која ќе биде содржината на погледот V1?

A1	A2	A3	A4	A5
7	5	3	7	6
✓	✓	✓	✓	✓

Прашање 2:

Во содржината на погледот V2 ќе има торки.

Прашање 1:

Гледај ги командите кај V1:

- A1 = 7, бидејќи ги бори сите редици
- A2 = 5, бидејќи ги брои сите редици освен тие што имат null
- A3 = 3, не ги брои дупликатите и null
- A4 = 7, бидејќи ги брои сите редици освен тие што имат null
- A5 = 6, бидејќи ги брои сите редици освен тие што имат null

Прашање 2:

Селектирај ги сите од С и тие што имат null и после кога ќе ги групираш во С тргни ги дупликатите и ќе ти останат 5 торки.

Прашање 3:

Кои од понудените торки ќе се најдат во содржината на погледот V2?

- (null,null)
- (null,1) ✓
- (null,0)
- (5,1) ✓
- (1,2)
- (5,3)
- (3,3) ✓

Mark 2.00 out of 2.00

Прашање 4:

Query1: SELECT BR, SUM(C) FROM V2 GROUP BY BR

Колку торки ќе се најдат во резултатот на извршувањето на SQL прашањето Query1 над погледот V2? 2 ✓

Прашање 5:

Query2: SELECT * FROM V2 WHERE BR = 1

Колку торки ќе се најдат во резултатот на извршувањето на SQL прашањето Query2 над погледот V2? 4 ✓

Прашање 6:

Query3: SELECT BR, SUM(C) FROM V2 GROUP BY BR HAVING COUNT(*) <> 1

Колку торки ќе се најдат во резултатот на извршувањето на SQL прашањето Query3 над погледот V2? 1 ✓

Прашање 3:

Броеш колку пати се појавила таа вредност во колоната.

Null се појави 1

5 се појави 1

3 се појави 3 пати

Прашање 4:

Бидејќи ги групираш ќе треба да нацрташ посебна табела

С колоната ја препишеш, тргнеш дупликатите.

ВР колоната ги ставаш колку пати се појавиле тие вредности во С колоната

Сега твојта табела треба да изгледа како ова подоле

C	BR
3	3
NULL	1
5	1
1	1
6	1

3	3
1	12

На крај гледаш ја ВР колоната оти групираш по неа и ги ставаш во помошна табела сите вредности и ги тргаш дупликатите како погоре

Сега треба само да имаш колона со 3 и 1

На крај сумираш ги вредностите во С колоната колку пати се појавила таа вредност

3 се појави 1 значи $SUM = 3$

1 се појави 4 пати сега гледаш во другата колона

$NULL + 5 + 1 + 6 = 12$

Крајната табела има 2 торки

Прашање 5:

Гледаш ги вредностите што се појавиле само 1 во помошната табела

Null се појави 1

5 се појави 1

1 се појави 1

6 се појави 1

Тоа се 4 торки

Прашање 6:

Ако гледаме во горната помошна табела што нацртавме имаме само 1 торка што е различна од 1.

Прашање 7:

Query4: SELECT BR, SUM(C) FROM V2 GROUP BY BR HAVING COUNT(DISTINCT BR) <> 1

Колку торки ќе се најдат во резултатот на извршувањето на SQL прашањето Query4 над погледот V2? 0 ✓

Прашање 8:

Во содржината на погледот V3 ќе има 6 ✓ торки.

Прашање 9:

Кои од понудените торки ќе се најдат во содржината на погледот V3?

- (3,6,0,NULL) ✓
- (1,3,2,2)
- (1,3,1,1) ✓
- (3,6,2,0)
- (1,3,2,3)
- (3,5,1,0)
- (2,3,2,4) ✓

Mark 3.00 out of 3.00

Прашање 7:

C	BR
3	3
NULL	1
5	1
1	1
6	1

3	3
1	12

Нема Distinct што се различни од 1 значи 0 торки

Прашање 8:

CREATE VIEW V3 as

SELECT B, C, COUNT(A) as D, SUM(A) as E

FROM R

GROUP BY B, C

B	C	D	E
1	3	1	1
2	3	2	4
1	NULL	1	2
3	5	0	NULL
4	1	1	2
3	6	0	NULL

B и **C** ги препишуваш.

R	A	B	C
1	1	1	3
1	2	2	3
2	1	1	null
null	3	3	5
2	4	4	1
3	2	2	3
null	3	3	6

D се добива бројќи колку пати се појавува парот на **B** и **C**

Пример 1 и 3 се појавува само 1 значи прва редица во D е 1

2 и 3 се појавува 2 пати

1 и нул се појавува 1

3, 5 се појавува 0 пати бидејќи COUNT(A) праввиме значи чим во колоната A имаме нул одма опаѓа

2, 3 се појавува 1

3, 6 се појавува 0

E се добива сумирајќи ги бројките во колоната A што имат исти вредности во колоните **B** и **C**

1 има 1, 3, се појавуват еднаш тоа е сумата од 1

1 има 2,3 се појавуват два пати тоа е сумата од $1 + 3 = 4$

3	2	3
---	---	---

Ова тројка ја сумираме **пази!**

2 има 1 и нул, се појавува еднаш значи сумата е 2

Нулл има 3 и 5, се појавува еднаш значи сумата е нулл

2 има 4, 1 се појавува еднаш значи сумата е 2

Итн итн за последните 2

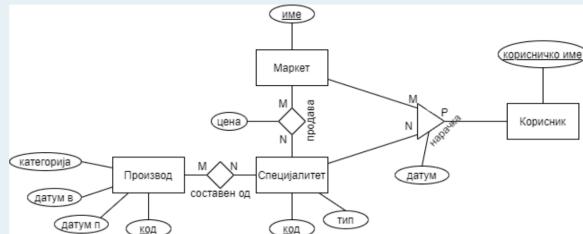
Прашање 9:

Гледаш кои торки ги имаш во табелата што ја доби и ги заокружуваш

Задача 3:

Важно: Нема многу филозофија тука едно треба да научеш како да претвараш ЕСкуЕл изрази во природен јазик

Дадена е база на податоци за маркети каде што се дадени следните табели: Korisnik, Market, Specijalitet, Narachka, Prodazhba.



Market(ID, ime, adresa, grad, reiting, raketodolitel, rabotnoVremeOd, rabotnoVremeDo)

Korisnik(ID, korisnickoIme, lozinka, email, adresaDostava)

Specijalitet(kod, ime, tip)

Prodazhba(marketID*, specijalitetKod*, cena)

Narachka(ID, marketID*, specijalitetKod*, korisnikID*, datum)

Proizvod(kod, kategorija, datum_v, datum_p)

Sostaven_od(proizvodKod*, specijalitetKod*)

Што ќе вратат како резултат следните DML изрази?

1) SELECT s.* FROM Specijalitet s, Produkt p, Sostaven_od sostav WHERE s.kod=sostav.specijalitetKod AND sostav.proizvodKod=p.kod AND p.kategorija='mlečni proizvodi'

- Ќе се вратат производите од категоријата 'mlečni proizvodi' кои влегуваат во состав на барем еден специјалитет.
- Ќе се вратат производите од категоријата 'mlečni proizvodi'.
- Ќе се вратат специјалитетите кои во својот состав содржат производ од категоријата 'mlečni proizvodi'.
-
- Ќе се вратат специјалитетите кои во својот состав не содржат производ од категоријата 'mlečni proizvodi'.
- Ќе се вратат специјалитетите кои во својот состав содржат само производи од категоријата 'mlečni proizvodi'.

Mark 2.00 out of 2.00

2) SELECT n.* FROM Narachka n, Specijalitet s WHERE n.specijalitetKod=s.kod AND datum='2021-01-01' AND s.ime LIKE '_%meso%'

- Ќе се вратат специјалитетите чие име завршува на 'meso' доколку е нарачан на 1.1.2021.
- Ќе се вратат нарачките направени на 1.1.2021 кои се однесуваат на специјалитет кој во своето име содржи 'meso' и не почнува со 'meso'.
-
- Ќе се вратат нарачките направени на 1.1.2021 кои се однесуваат на специјалитет чие име содржи 'meso'.
- Ќе се вратат сите информации за специјалитетот со име 'meso' доколку е нарачан на 1.1.2021.

Mark 2.00 out of 2.00

3) SELECT marketID FROM Prodazhba WHERE cena=(SELECT MAX(cena) FROM Prodazhba)

- Ќе се врати шифрата на маркетот во кој е направена најскапата нарачка.
- Ќе се врати шифрата на маркетот во кој се продаваат најмногу производи.
- Ќе се врати шифрата на маркетот во кој се продава најскапиот продукт.
-
- Ќе се врати шифрата на маркетот во кој сите производи се со највисока цена

Тип: Пробај да ги читаш изразите во обратен редослед, тоа обично помага

- 4) SELECT korisnikID, COUNT(*) FROM Narachka GROUP BY korisnikID
- Ке се вратат шифрите на корисниците заедно со бројот на нарачки кои се направени.
 - За секој маркет ќе се вратат шифрите на корисниците кои имаат направено нарачка заедно со бројот на нарачки кои ги имаат направено.
 - За секој корисник ќе се врати шифрата и бројот на маркети во кои има направено нарачка.
 - За секој корисник ќе се врати шифрата и бројот на нарачки кои ги има направено.

Mark 2.00 out of 2.00

- 5) SELECT korisnikID, COUNT(DISTINCT marketID) FROM Narachka GROUP BY korisnikID
- Ке се вратат шифрите на корисниците заедно со бројот на различни маркети во кои се направени нарачки.
 - За секој маркет ќе се вратат шифрите на корисниците кои имаат направено нарачка заедно со бројот на нарачки кои ги имаат направено.
 - За секој корисник ќе се врати шифрата и бројот на нарачки кои ги има направено.
 - За секој корисник ќе се врати шифрата и бројот на маркети во кои има направено нарачка.

Mark 2.00 out of 2.00

- 6) SELECT korisnikID, specijalitetKod, COUNT(*) FROM Narachka GROUP BY korisnikID, specijalitetKod
- За секој корисник ќе се вратат шифрите на специјалитетите за кои има нарачка и бројот на специјалитети за кои има нарачка.
 - За секој корисник ќе се врати бројот на специјалитети кои има нарачано заедно со бројот на нарачки кои ги има направено.
 - За секој корисник ќе се вратат шифрите на специјалитетите за кои има направено нарачка и бројот на нарачки што ги има направено.
 - За секој корисник ќе се врати колку пати го нарачал секој од специјалитетите за кои има направено нарачка.

✓

Задача 4:

Важно: Нема многу филозофија тутка едно треба да научеш како да претвараш ЕСкуЕл изрази во природен јазик

Тип: A% значи почнува со А, а %A значи завршува

Прашање 1: Да се спарат дадените DML изрази со соодветното барање.

Query1: SELECT * FROM Specijalitet s1 WHERE ime LIKE 'A%' AND EXISTS (SELECT * FROM Specijalitet s2 WHERE ime LIKE '%A')

Query2: SELECT * FROM Specijalitet s1 WHERE ime LIKE 'A%' AND EXISTS (SELECT * FROM Specijalitet s2 WHERE ime NOT LIKE '%A')

Query3: SELECT * FROM Specijalitet s1 WHERE ime LIKE 'A%' AND EXISTS (SELECT * FROM Specijalitet s2 WHERE ime LIKE '%A' AND s1.kod<>s2.kod)

Query4: SELECT * FROM Specijalitet s1 WHERE ime LIKE 'A%' AND EXISTS (SELECT * FROM Specijalitet s2 WHERE ime NOT LIKE '%A' AND s1.kod<>s2.kod)

Query5: SELECT * FROM Specijalitet s1 WHERE ime LIKE 'A%' AND NOT EXISTS (SELECT * FROM Specijalitet s2 WHERE ime LIKE '%A' AND s1.kod<>s2.kod)

Query6: SELECT * FROM Specijalitet s1 WHERE ime LIKE 'A%' AND NOT EXISTS (SELECT * FROM Specijalitet s2 WHERE ime NOT LIKE "%A' AND s1.kod<>s2.kod)

Query7: SELECT * FROM Specijalitet s1 WHERE ime LIKE 'A%' AND EXISTS (SELECT * FROM Specijalitet s2 WHERE s1.kod=s2.kod AND ime LIKE '%A')

Query8: SELECT * FROM Specijalitet s1 WHERE ime LIKE 'A%' AND EXISTS (SELECT * FROM Specijalitet s2 WHERE s1.kod=s2.kod AND ime NOT LIKE '%A')

Query 5 ✓ - Да се вратат специјалитетите чие име почнува на А доколку не постои друг специјалитет чие име завршува на А.

Query 2 ✓ - Да се вратат специјалитетите чие име почнува на А доколку постои барем еден специјалитет чие име не завршува на А.

Query 8 ✓ - Да се вратат специјалитетите чие име почнува на А и не завршува на А.

Query 4 ✓ - Да се вратат специјалитетите чие име почнува на А доколку постои друг специјалитет чие име не завршува на А.

Query 6 ✓ - Да се вратат специјалитетите чие име почнува на А доколку не постои друг специјалитет чие име не завршува на А.

Query 1 ✓ - Да се вратат специјалитетите чие име почнува на А доколку постои барем еден специјалитет чие име завршува на А.

Query 7 ✓ - Да се вратат специјалитетите чие име почнува на А и завршува на А.

Query 3 ✓ - Да се вратат специјалитетите чие име почнува на А доколку постои друг специјалитет чие име завршува на А.

Важно: За првиот израз еден тип, ако се траже да вратиш се или **ВРАТИ ГИ СИТЕ** обично тоа се прави со двојна негација. Првиот израз има 2 NOT EXISTS тоа е двојната негација и ке се селектираат сите маркети

Прашање 2: Да се спарат дадените DML изрази со соодветното барање.

Query1: SELECT * FROM Specijalitet s
WHERE NOT EXISTS (SELECT * FROM Market m
WHERE NOT EXISTS (SELECT * FROM Prodazhba p WHERE p.specijalitetKod=s.kod AND p.marketId=m.ID))

Query2: SELECT * FROM Specijalitet s
WHERE EXISTS (SELECT * FROM Market m
WHERE NOT EXISTS (SELECT * FROM Prodazhba p WHERE p.specijalitetKod=s.kod AND p.marketId=m.ID))

Query3: SELECT * FROM Specijalitet s
WHERE NOT EXISTS (SELECT * FROM Market m
WHERE EXISTS (SELECT * FROM Prodazhba p WHERE p.specijalitetKod=s.kod AND p.marketId=m.ID))

Query4: SELECT * FROM Specijalitet s
WHERE EXISTS (SELECT * FROM Market m
WHERE EXISTS (SELECT * FROM Prodazhba p WHERE p.specijalitetKod=s.kod AND p.marketId=m.ID))

Query 4 ✓ - Да се вратат сите информации за специјалитетите кои се продаваат во барем еден од marketите.

Query 2 ✓ - Да се вратат сите информации за специјалитетите кои не се продаваат во барем еден од marketите.

Query 3 ✓ - Да се вратат сите информации за специјалитетите кои не се продаваат во ниту еден од marketите.

Query 1 ✓ - Да се вратат сите информации за специјалитетите кои се продаваат во сите marketи.

Многу од прашањата се решават исто

Имаш израз претвори го во природен јазик или обратно.

Тука не можам многу да ти помогнам ако не знаеш ЕскуЕл.

Па нема да губам време на сите оти се огромни

Сори мејт :/

Задача 5:

Важно: Сите ѝоинс прават дупликати освен Натурал

Тип: Научи ѝоинс

За крос ѝоин се множат колоните и редиците

Инер ѝоин ако се исти редиците ќе се спојат заедно со колоните ама треба да се исполне условот даден

За лефт оутер ѝоин се додават нуллс ако нема редици тоест празни редици

Исто и за рајт оутер ѝоинс

Дадени се табелите R(A,B), S(B,C) и T(B,D).

Состојбата на табелите R, S и T е

R	A	B
a1	b1	
a2	b2	
a3	b3	
a4	b4	

S	B	C
b1		c1
b2		c2
b3		c3

T	B	D
b1		d1
b2		d2

Колку редици и колони ќе има во резултатот од извршувањето на следните DML изрази?

1) SELECT * FROM R CROSS JOIN T

редици и колони.

2) SELECT * FROM R INNER JOIN T ON R.B=T.B

редици и колони.

3) SELECT * FROM S LEFT OUTER JOIN T ON S.B=T.B

редици и колони.

4) SELECT S.* FROM S RIGHT OUTER JOIN T ON S.B=T.B

редици и колони.

5) SELECT * FROM (SELECT * FROM R LEFT OUTER JOIN S ON R.B=S.B WHERE C IS NULL) AS RS FULL OUTER JOIN T ON RS.B=T.B

редици и колони.

6) SELECT * FROM R NATURAL RIGHT OUTER JOIN T

редици и колони.

1) Р крос Т = Р

- Р има 4 редици а Т има $2 = 4 * 2 = 8$
- Р има 2 колони а Т $2 = 2 * 2 = 4$

2) Р Инер Т на Р.В = Т.В

- Во Р.В има 2 редици што се исит б1 и б2 = 2 редици
- Кога ќе се спојат тие 2 редици ќе имам 4 колони

- 3) Лефт оутер Џоин
- Т има 2 редици а S има 3 значи се додава нул редица во Т и пишуваме 3 редици
 - Колоните се спојуват од 2те табели т.е имаме 4
- 4) Рајт оутер Џоин
- Условот испаѓа за последната редица затоа ја бришеме = 2 редици
 - Сега тука се спојуват само колоните што го исполнуваат условот значи ќе имаме 2 колони
- 5) Итн за сите натаму само да напоменам НАТУРАЛ џоин ги трга дупликатите

3) Вовед во SQL: Погледи, тврдења, тригери и Вовед во трансакции

Задачи за директно преведување на EcKyEl ќе ги скипнам

Задача 1:

Важно: Кога има **ASSERTION** секојпат е **CHECK** па **NOT_EXISTS** (финта)!

Question 2
Correct
Mark 10.00 out of 10.00
Flag question

Дадени се табелите
PREVOZ(p.id, datum, odGrad, doGrad, trgnuvanje, prevoznik*)
PREVOZNIK(id, naziv, grad)

Да се напише SQL израз преку кој нема да се дозволи внес на запис во табелата PREVOZ во кој и градот на појање и градот на пристигнување се различни од градот на превозникот. (Да се дополни SQL изразот)

CREATE ASSERTION proverka **CHECK** ✓
(NOT EXISTS ✓
(SELECT * FROM PREVOZ Pz, PREVOZNIK Pr WHERE
Pz.prevoznik=Pr.id ✓ **AND** ✓ **Pz.odGrad<>Pr.grad** ✓ **AND** ✓ **Pz.doGrad>Pr.grad** ✓
 $)$

Тип: Кога имаме ваква задача што траже **ASSERTION** доле во EcKyEl изразот обично треба да го ставеш обратното од тоа што се траже бидејќи **NOT_EXISTS** го негира.

Задача 2:

Дадени се табелите R(a,b) и S(a,c).

Питање 1:

Дадено е следнovo ограничување.

```
CREATE ASSERTION Mystery1 CHECK (
NOT EXISTS
(SELECT * FROM R, S WHERE R.a = S.a AND R.b <> S.c)
);
```

Кои од наведените комбинации торки не е дозволено истовремено да постојат?

- (1,1) во R и (2,2) во S
- (1,2) во R и (1,2) во S
- (1,2) во R и (2,1) во S
- (1,1) во R и (1,2) во S ✓

Mark 3.00 out of 3.00

НЕ Е ДОЗВОЛЕНО, читај убаво љубави

(1,1) (1,2) само бидејќи го исполнуваат R.B <> S.C правилото

НЕГИРА СЕ ОВА ПРАВИЛО ЗНАЧИ КОГА ТРАЖЕМЕ ДА НЕ Е ДОЗВОЛЕНО ТРЕБА ДА СЕ ИПОЛНЕ!!

Прашање 2:

Дадено е следнovo ограничување.

```
CREATE ASSERTION Mystery2 CHECK (
    NOT EXISTS
    (SELECT * FROM R, S WHERE R.a = S.c AND R.b = S.a)
);
```

Кои од наведените комбинации токки не е дозволено истовремено да постојат?

- (1,2) во R и (2,1) во S ✓
- (1,1) во R и (1,2) во S
- (1,1) во R и (2,2) во S
- (1,2) во R и (1,2) во S



Mark 3.00 out of 3.00

Доунт мајнд да стрелка од маус знам дека е аутистична малце како мене ама битно се труде!

Исто како горе се исполнува условот а бидејќи се негира значи се крше!

(1,2) и (2,1) 1 со 1 и 2 со 2 бидејќи $R.a = S.c$ AND $R.b = S.a$

ЧЕКАЈ МАЛЦЕ

Ако си стигнал до овде да знаеш дека без разлика од каков успех ќе постигнеш или
нема да постигнеш, колку пари ќе имаш или нема да имаш,

дека вредиш <3

Овие глупости не те вреднуват тебе
и верувам дека си прекрасна личност

- мацо

Олрајт лез гоу

Задача 3:

Дадени се табелите T(A, B*) и S(B, C). Нека моменталната состојба на табелите е дадена со:

T		S	
A	B	B	C
1	1	1	2
2	2	2	2
3	1	3	1
4	3	4	1
5	4	5	4
6	2	6	3
7	3	7	1

Даден е следниот тригер:

```
CREATE TRIGGER UNKNOWN
BEFORE UPDATE OF C ON S
FOR EACH ROW
WHEN NEW.C=NEW.B AND NEW.B=OLD.B
BEGIN
    DELETE FROM T WHERE B=(SELECT B FROM S WHERE B=OLD.B) ;
END
```

Питање 1:

Колку торки ќе содржи табелата T по извршување на следниве SQL наредби? (Внесете нумеричка вредност)

Напомена: Наредбите не се извршуваат секвенцијално туку само наредбата во даденото подпрашање!

1) UPDATE S SET C=3 WHERE B=1

✓ торки

2) UPDATE S SET C=3 WHERE B=3

✓ торки

За вакви задачи е битно условот дали е исполнет или и да гледаш каде брише.

Друго битно е да пазеш во која табела се брише.

- 1) Никаде нема да се брише бидејќи условот WHEN NEW.C=NEW.B AND NEW.B=OLD.B не е исполнет
- 2) Е сега е исполнет
 - a) Во табелата S редицата 3, 1 ќе стане 3, 3 значи дека сега условот е исполнет и кога е исполнет гледаш во табелата T кои ќе се избришат

- b) Редиците 7,3 и 4, 3 ќе се избришат остануват 5 редици
c) Бидејќи таа Зка е старото Б и тоа старо Б е еднакво на новот Б
каде што се исполнит условот

3) Итн За другите

3) UPDATE S SET B=8, C=3 WHERE B=3

✓ торки

4) UPDATE S SET C=B WHERE C=2

✓ торки

5) UPDATE S SET C=B WHERE C=1

✓ торки

6) UPDATE S SET C=B WHERE C<3

✓ торки

Прашање 2:

Што прави тригерот даден погоре?

- Доколку каде дадена торка од табелата T се постави вредноста на B да е еднаква на вредноста на C која ја има записот од табелата S за таа вредност на B, тогаш ќе се избришат сите торки од S кај кои ја има дадената вредност за B.
- Доколку каде дадена торка од табелата T се постави вредноста на B да е еднаква на вредноста на C која ја има записот од табелата S за таа вредност на B, тогаш ќе се избришат сите торки од T кај кои ја има дадената вредност за B.
- Доколку каде дадена торка од табелата S се постави вредноста на C да е еднаква на B, а при тоа вредноста на B да не се промени, тогаш ќе се избришат сите торки од T кај кои ја има дадената вредност за B.
- Доколку каде дадена торка од табелата S атрибутите B и C си ги променат вредностите, тогаш ќе се избришат сите торки од T кај кои ја има старата вредност за B од табелата S.

Mark 3.00 out of 3.00

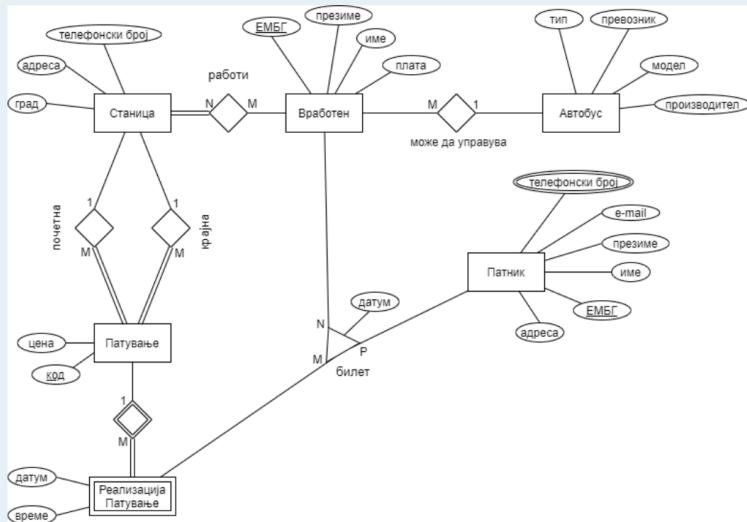
Lichtschat

Прашање 2: Самата реценица се објаснува нема многу филозофија

AJDE TEKELAAA

Задача 4:

Дадена е база на податоци за автобуски станици каде што се дадени следните табели: Stanica, Vraboten, Raboti, Avtobus, Patuvanje, Realizacija_patuwanje, Patnik, Patnik_tel_broj, Bilet.



```
Stanica(id, grad, adresa, tel_br)
Vraboteni(embg, ime, prezime, plata, avtobus_id)
Raboti(stanica_id*, vraboteni_embg*)
Avtobus(id, tip, prevoznik, model, proizvoditelj)
Patuvanje(kod, cena, pocetna_s*, krajna_s*)
Realizacija_patuvanje(kod*, datum, vreme)
Patnik(embg, ime, prezime, adresa, email)
Patnik_tel_broj(pat_embg*, broj)
Bilet(id, patuv_kod*, patuv_datum*, patuv_vrem)
```

Важно: Пази какви релации има и клучевите

Во табелата Vraboten е креиран изведен атрибут „**brBileti**“ во кој е зачуван вкупниот број на билети што ги продал вработениот. Дополнително, пресметана е моменталната вредност на овој атрибут за секој вработен. Да се напишат тригери за одржување на консистентноста на атрибутот „**brBileti**“ при додавање, бришење и ажурирање на записи од кои зависи неговата вредност.

CREATE TRIGGER triggerInsert

AFTER INSERT ON Bilet ✓

FOR EACH ROW ✓

BEGIN

Изберете кои од понудените наредби ќе се најдат во телото на тригерот.

- UPDATE Vraboten SET brBileti=brBileti+1 WHERE embg=OLD.vraboten_embg;
- UPDATE Vraboten SET brBileti=brBileti+1 WHERE embg=NEW.vraboten_embg; ✓
- UPDATE Vraboten SET brBileti=brBileti-1 WHERE embg=OLD.vraboten_embg;
- UPDATE Vraboten SET brBileti=brBileti-1 WHERE embg=NEW.vraboten_embg;

Mark 3.00 out of 3.00

END;

CREATE TRIGGER triggerDelete

AFTER DELETE ON Bilet ✓

FOR EACH ROW ✓

BEGIN

Изберете кои од понудените наредби ќе се најдат во телото на тригерот.

- UPDATE Vraboten SET brBileti=brBileti+1 WHERE embg=OLD.vraboten_embg;
- UPDATE Vraboten SET brBileti=brBileti-1 WHERE embg=OLD.vraboten_embg; ✓
- UPDATE Vraboten SET brBileti=brBileti-1 WHERE embg=NEW.vraboten_embg;
- UPDATE Vraboten SET brBileti=brBileti+1 WHERE embg=NEW.vraboten_embg;

Mark 3.00 out of 3.00

Lightshot

Your screenshot is copied to clipboard

CREATE TRIGGER triggerUpdate

AFTER UPDATE OF vraboten_embg ON Bilet ✓

FOR EACH ROW ✓

BEGIN

Изберете кои од понудените наредби ќе се најдат во телото на тригерот.

- UPDATE Vraboten SET brBileti=brBileti+1 WHERE embg=NEW.vraboten_embg; ✓
- UPDATE Vraboten SET brBileti=brBileti+1 WHERE embg=OLD.vraboten_embg;
- UPDATE Vraboten SET brBileti=brBileti-1 WHERE embg=NEW.vraboten_embg;
- UPDATE Vraboten SET brBileti=brBileti-1 WHERE embg=OLD.vraboten_embg; ✓

Mark 3.00 out of 3.00

END;

Прашање 2:

Во табелата Vraboten е креиран изведен атрибут „**brPatnici**“ во кој е зачуван број на патници на кои вработениот им продал билети. Дополнително, пресметана е моменталната вредност на овој атрибут за секој вработен. Да се напишат тригери за одржување на консистентноста на атрибутот „**brPatnici**“ при додавање на записи од кои зависи неговата вредност.

CREATE TRIGGER triggerInsert2

AFTER INSERT ON Bilet ✓

FOR EACH ROW ✓

BEGIN

Изберете кои од понудените наредби ќе се најдат во телото на тригерот.

- UPDATE Vraboten SET brPatnici=(SELECT COUNT(*) FROM Bileti WHERE vraboten_embg=OLD.vraboten_embg) WHERE embg=NEW.vraboten_embg;
- UPDATE Vraboten SET brPatnici=brPatnici+1 WHERE embg=NEW.vraboten_embg;
- UPDATE Vraboten SET brPatnici=(SELECT COUNT(DISTINCT patnik_embg) FROM Bileti WHERE vraboten_embg=NEW.vraboten_embg) WHERE embg=NEW.vraboten_embg; ✓

Пазливо ја читај реченицата и што се бара

Тип: Секојпат е **FOR EACH ROW** (финта)!

Треба да се одрже конзистентноста значи треба да се апдејтира и старата и новата вредност.

Задача 5:

Да се напишат тригери за одржување на референтниот интегритет на базата при ажурирање на запис во табелата за патниците.

CREATE TRIGGER triggerUpdateP

AFTER UPDATE OF embg ✓ ON Patnik ✓

FOR EACH ROW ✓

BEGIN

--UPDATE Vrabot: (Изберете дали и која од понудените наредби е точна - ќе се најде во телото на тригерот).

- UPDATE Vrabot SET embg= OLD.embg WHERE embg= NEW.embg;
- UPDATE Vrabot SET NEW.embg= embg;
- UPDATE Vrabot SET embg= NEW.embg;
- Ниту едно од понудените.✓
- UPDATE Vrabot SET embg= NEW.embg WHERE embg= OLD.embg;
- UPDATE Vrabot SET OLD.embg= NEW.embg;

Mark 2.00 out of 2.00

--UPDATE Bilet: (Изберете дали и која од понудените наредби е точна - ќе се најде во телото на тригерот).

- UPDATE Bilet SET embg= OLD.embg WHERE embg= NEW.embg;
- Ниту едно од понудените.
- UPDATE Bilet SET patnik_embg= OLD.embg WHERE patnik_embg= NEW.embg;
- UPDATE Bilet SET embg= NEW.embg WHERE embg= OLD.embg;
- UPDATE Bilet SET NEW.embg= embg;
- UPDATE Bilet SET patnik_embg= NEW.embg;
- UPDATE Bilet SET patnik_embg= NEW.embg WHERE patnik_embg= OLD.embg;

Mark 3.00 out of 3.00

Овде е исто како претходната само ќе напоменам да пазеш што апдејтираш

Кога се одржува **референтниот интегритет** како што пишува треба да апдејтираш некој примарен клуч!!! **ЗНАЧИ UPDATE OF Embg** во овај пример а не само UPDATE

Задача 6:

Важно: За **dirty read** треба да е нагласано дека една од трансакциите пропаѓа поради некоја магична причина

За **lost update** практично би го објаснил како дека се губе трансакција бидејќи не се зачувало вредноста пред да се чита друга вредност

Како задачата подоле

Прво ја чита X мене м уја вредноста = $10 - 3 = 7$

Сега X е 7 ама оваа вредност не се зачува

Сега другата табела ја чита вредноста како 10 а не 7

И тоа е лост апдејт

За **unrepeatable read** најубаво објаснето е тако:

„Ти ако кажеш на некој другар дека ќе одеш за Париз и тај другар каже на трет, а ти меѓувреме се предомислеш испаѓа дека другарот лаже.“

Нека се дадени трансакциите T1 и T2 кои се извршуваат конкурентно.

Page 36

T_1	T_2
read_item(X);	
$X := X - N;$	
write_item(X);	read_item(X);
read_item(Y);	$X := X + M;$
	write_item(X);
$Y := Y + N;$	
write_item(Y);	

Кој проблем настанува при конкурентното извршување на овие две трансакции?

Select one:

- нема никаков проблем при извршувањето на трансакциите
- изгубено ажурирање (lost update)
- неповторливо читање (unrepeatable read)
- валкано читање (dirty read)



Your answer is correct.

Задача 7:

Важно: Цртај!

Question 10
Answer saved
Marked out of 15.00
Flag question

Даден е следниот распоред на извршување на три трансакции.
 $S = r_2(Z) \ r_2(Y) \ w_2(Y) \ r_3(Z) \ r_1(X) \ w_1(X) \ w_3(Y) \ w_2(Z) \ r_2(Y) \ w_1(Y) \ w_2(Y)$
Дали дадениот распоред е сериабилен или не?

Select one:
 True
 False

Previous page

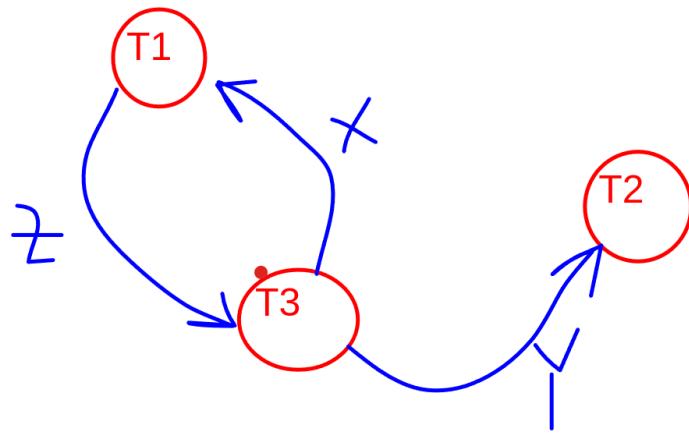
Test 5 - втор дел

Jump to...

ПРИМЕР

$S_1: r_1(X); r_2(Z); r_1(Z); r_3(X); r_3(Y); w_1(X); w_3(Y); r_2(Y); w_2(Z); w_2(Y);$

T ₁	T ₂	T ₃
r ₁ (X)		
r ₁ (Z)	r ₂ (Z)	
w ₁ (X)		r ₃ (X)
	r ₂ (Y)	r ₃ (Y)
	w ₃ (Y)	w ₁ (Y)
		w ₂ (Y)



Нацртај си вака табела со сите трансакции

И по ред си ги ставај почнувајќи од нагоре кон надоле

Тие што се маркирани со 1 одат во T1 тие со 2 одат во T2 итн итн

Нацртај топки што ги означува трансакциите

Почнувајќи од последниот врајт (тај најдоле)

Тражеш во другите табели (не во твојта) конфликт.

Конфликт се пронајдува со иста вредност во заградите **пример: $W1/R1(x) = W2/R2(x)$**

Значи во T3 има прв конфликт во T2 и цртаме од T3 кон T2

Конфликт може да биде од W или R

Одеснагоре кон следниот Write

Има во T1 значи од T1 кон T3 цртаме конфликт

Следен врв е W(Y) нема конфликт

Следен врв е W(X) има конфликт во T3 кај R(X)

Опа имаме циклус

Чим имаме циклус распоредот **НЕ Е** серијабилен

Ако продолжеш до крај и не најдеш циклу тогаш **Е** серијабилен

4) Колоквиумски/Испитни/Рандом

Задача 1:

Во дадена база на податоци постои релацијата **S(a, b, c)** за која ги имате следните информации:

- Датотеката која ги содржи записите (торките) од релацијата **S** зафаќа 300 блока и во секој блок се сместени по 4 записи (торки).
- За релацијата **S** креиран е индекс врз атрибутот **S.c** од типот **B^{*} Алтернатива 2** (во листовите се чуваат само покажувачи кон блоковите во кои се смеѓа релацијата) **некластерирачки индекс** (датотеката со записи е подредена според атрибутот **S.a**)
- Моменталната состојба на индексот ги има следните карактеристики: **висина 3 и 35 терминални јазли** (листови).

Прашање А

Колкав е најголемиот број на пристапи до блоковите (низ индексот и датотеката со записи) доколку е потребно да се пронајде запис од релацијата кој зададен пример **S.c = 20**?

Напомена: Најголемиот број пристапи се однесува на најлошиот случај кога треба да се проверат сите записи од релацијата!

Потребно е да се пристапи до вкупно 1200 блокови

Прашање Б

Да се направи споредба помеѓу операцијата на изминување на сите записи од релацијата без индекс и преку креираниот индекс. Кој од следните искажи е правилен?

Пефикасно е изминување на сите записи од релацијата без индекс.

Нема разлика Ѓуријо

Прашање А:

Формулата е: **блоковите * записите**

300 блока * 4 записи = 1200 записи

Прашање Б:

Не знам искрено