

## ПИТАЊА

### Питање 1. [2 поена]

Табела Student има ред величине 128 В.

Величина блока је 512 В. Колико блокова је потребно за чување 200 редова?

- A) 25
- B) 40
- C) 50
- D) 64

### Питање 2. [2 поена]

Табела има 1000 редова. Без индекса потребно је прегледати 250 блокова. Коришћењем вишеслојног индекса потребно је 3 приступа индексним блоковима и 1 приступ блоку са подацима. Колико максимално блокова се чита?

- A) 1
- B) 3
- C) 4
- D) 250

### Питање 3. [2 поена]

Ако табела садржи приближно 16 милиона елемената, колико је приближно потребно поређења коришћењем бинарне претраге?

- A) 16 милиона
- B) 16000
- C) 1024
- D) 24

### Питање 4. [2 поена]

В-stablo је реда 5. Колико максимално може имати кључева и „child” показивача?

- A) 4 кључа и 5 деце
- B) 5 кључева и 4 деце
- C) 5 кључева и 6 деце
- D) 6 кључева и 5 деце

### Питање 5. [2 поена]

Блок је величине 512 В. Кључ = 10 В, pointer = 6 В док заглавље занемарујемо. Који је максимални ред стабла?

- A) 16
- B) 32
- C) 24
- D) 64

**Питање 6. [2 поена]**

Табела има 10 колона и 10 милиона редова. Извршава се агрегациони упит. Која база ће вероватно имати мање IO операција?

- A) Row-oriented
- B) Column-oriented
- C) Оба система исто
- D) Ниједна

**Питање 7. [2 поена]**

Зашто су INSERT операције често спорије код column-oriented база?

- A) Зато што нема индекса
- B) Зато што се нови ред мора уписивати у више колонских блокова
- C) Зато што користе BST стабла
- D) Зато што немају ROWID

**Питање 8. [2 поена]**

Зашто column-oriented базе имају бољу компресију података?

- A) Зато што користе мање блокове
- B) Зато што не користе pointers
- C) Зато што имају мање табела
- D) Зато што су подаци истог типа груписани заједно

**Питање 9. [2 поена]**

Која колона је НАЈЛОШИЈИ избор за partition key?

- A) STATUS
- B) USER\_ID
- C) DATUM\_KREIRANJA
- D) REGION

**Питање 10. [2 поена]**

Зашто стриктно растући ID може бити проблем код партиционисања?

- A) Зато што повећава величину RAM-а
- B) Зато што сви нови подаци завршавају у последњој партицији
- C) Зато што смањује број партиција
- D) Зато што спречава индексирање

**Питање 11. [2 поена]**

Који је ГЛАВНИ циљ партиционисања?

- A) Повећање величине редова
- B) Смањење простора на диску
- C) Смањење броја података које треба претражити
- D) Уклањање индекса

**Питање 12. [2 поена]**

Шта DBMS мора да уради ако се промени partition key реда?

- A) Само да промени индекс
- B) Само да промени hash вредност
- C) Да обрише ред из старе партиције и убаци га у нову
- D) Ништа посебно

**Питање 13. [2 поена]**

Табела користи PARTITION BY HASH(USER\_ID) и има 4 партиције. Ако је  $\text{hash}(25)=9$  у коју партицију иде податак?

- A) P0
- B) P1
- C) P2
- D) P3

**Питање 14. [2 поена]**

Шта значи eventual consistency model код NoSQL база?

- A) Сви сервери увек имају идентичне податке
- B) Подаци никада нису синхронизовани
- C) Реплике ће након одређеног времена постати усклађене
- D) Нема репликације података

**Питање 15. [2 поена]**

Која комбинација особина најбоље описује NewSQL базе података?

- A) Нема SQL + нема ACID
- B) SQL + ACID + хоризонтална скалабилност
- C) Само вертикално скалирање
- D) Само key-value модел

*НАПОМЕНА: Погрешно одговорено питање носи -1 поен.*

## ЗАДАЦИ

Дате су следеће релације базе података **UNIVERZITET**.

- **STUDENTI**: Садржи основне податке, низ положених испита и низ историје уплата
- **PROFESORI**: Садржи податке о професорима, њиховој катедри и лиценци

Колекција **profesori** [ *\_id, ime, prezime, katedra, licenca, staz* ] садржи:

- 101, Jovan, Jovanovic, racunari, 7788, 10
- 102, Ana, Petrovic, matematika, 4455, 20
- 103, Igor, Simic, opsti\_predmeti, 1122, 4
- 104, Elena, Kostic, racunari, 3366, 7

Колекција **studenti** [ *ime, prezime, grad, status, polozeni\_ispiti, uplate ( profesor\_id, predmet, iznos )* ] садржи:

- Marko, Markovic, Nis, budzet, [ *Baze\_Podataka* ], [ { 102, Analiza, 4000 }, { 103, Engleski, 1500 } ]
- Milica, Stankovic, Beograd, samofinansiranje, [ *Matematika, Programiranje* ], [ { 101, Softveri, 3000 } ]
- Slobodan, Ilic, Beograd, master, [ ], [ { 103, Sociologija, 1200 } ]
- Sara, Nikolic, Kragujevac, budzet, [ ], [ ]

### Задатак 1. [8 поена]

Написати MongoDB Shell наредбе за креирање описане базе, колекција и за додавање докумената у колекције.

### Задатак 2. [13 поена]

Написати mongosh упите за следеће операције над колекцијама:

a) [3 поена]

Приказати име, презиме и град свих студената који су из Београда или Ниша, а чије се име завршава са "ог" или презиме у себи садржи "ан". Такође, студент не сме имати математику у низу положених испита.

b) [5 поена]

Свим студентима из Београда, додати нови положени испит "Baze\_Podataka" у низ, али тако да се не стварају дупликати унутар низа.

c) [5 поена]

Студенту Марку Марковићу поставити статус на "diplomirao" и увећати износ његове последње уплате за 200 динара.

**Задатак 3. [18 поена]**

Креирати колекцију inventar са валидацијом. Обавезна поља су naziv, tip и inventarski\_broj [15 поена].

- naziv мора бити string,
- tip може бити само једна од следећих вредности: "racunari", "namestaj" или "projekcija",
- inventarski\_broj мора бити цео број већи од 10000.

Написати скрипту која [3 поена]:

- Проналази све рачунаре чија је вредност мања од 130.000.
- Прескаче првих 5 резултата и приказује наредна 2.
- Из финалног приказа избацује поље \_id и tip.

**Задатак 4. [21 поена]**

Написати комплексне агрегационе оквире за следеће потребе:

a) [8 поена]

Израчунати просечну потрошњу по сваком студенту. Потребно је:

- Посматрати појединачне uplate.
- Груписати податке по имену студента и израчунати просечну вредност свих његових уплата.
- Резултате сортирати тако да они са највећим просеком буду први.

b) [8 поена]

Извршити спајање колекција, студената са професорима. Приказати:

- Име студента.
- Предмет са уплате.
- Име и презиме професора који држи предмет.
- Филтрирати резултате тако да се приказују само уплате код професора са катедре за рачунарство.

c) [5 поена]

Пронаћи најјефтинију појединачну уплату икада урађену у систему. Приказати само презиме студента и износ цене.