

Испит из Релационих база података, јун 2024. год. (И смер)

Број индекса	Име и презиме

Задаци се раде 240 минута. Максималан број поена је 100. Број поена на испиту се израчунава тако што се саберу освојени поени по задацима, збир подели са два и заокружи. Број поена по задацима је:

Задатак	1	2	3	4	Збир 1-4		5	6	7	8	Збир 5-8		Укупно
Поена	16	20	28	16	80	Збир/2	30	20	40	30	120	Збир/2	
Освојено													

- Написати SQL упит који за сваког студента издваја податке о предметима које су уписали, а нису положили. Приказати само оне студенте који су у последњих шест година положили најмање три предмета са оценом 9 или 10. Издвојити име и презиме студента (у једној колони), назив предмета и назив месеца у коме је студент дипломирао. У случају да студент још увек није дипломирао, исписати "Није дипломирао". Резултат упита уредити према имену и презимену студента у растућем поретку, а затим према називу предмета у опадајућем поретку.
- Издвојити испите које су студенти пали или желе да пониште. Студент жели да поништи испит уколико је регуларно полагао и фали му 1 поен до више оцено, односно освојио је 60, 70, 80 или 90 поена. Потребно је издвојити и студенте који немају испит који испуњава услове. Потребно је издвојити име и презиме студента раздвојене размаком (колону назвати *student*), освојене поене (колону назвати *poeni*), назив предмета (колону назвати *predmet*), школску годину полагања (колону назвати *godina*), и колону са називом *komentar*. Уколико је студент освојио 50 поена коментар треба да буде *Student je pao*, уколико му фали поен до више оцено треба да буде *Student zeli visu ocenu*, иначе *Student ne zeli nista da ponisti*. Ово није потребно издвајати за све испите, већ само за оне испите које је студент полагао у школској години у којој је положио више од 30 еспб поена и више од 5 испита. У резултату један студент може да се нађе више пута, по једном за сваку годину која испуњава услов, у неким може да не жели ништа да поништи, а у другој да жели. Резултат је потребно уредити према индексу студента растуће и школској години опадајуће.
- Написати SQL наредбу којом се прави табела *apsolvent* која има наредне колоне:
 - indeks* - цео број, примарни кључ,
 - poenadokraja* - цео број,
 - godinastudija* - цео број,
 - vremeunosa* - датум, подразумевано данашњи датум.
 - Направити кориснички дефинисану функцију *is_polozeno*. Функција прима два аргумента:
 - student* - цео број који представља индекс студента.
 - predmet* - цео број који представља идентификатор предмета.
Функција треба да врати индикатор да ли је студент положио предмет.
 - Написати наредбу која уноси вредности у табелу из дела под а. Потребно је унети *indeks*, *poenadokraja* и колону *godinastudija*. Унети студенте који имају мање од 30 бодова до краја студија. Број бодова до краја студија се рачуна као разлика обима еспб бодова на смеру и збира еспб бодова предмета које је студент положио. Година студија се рачуна као број година који је прошао од када је студент уписао студије до данас.
 - Написати наредбу којом се формира окидач *unos_ispita* који ће модификовати табелу из дела по а када студент положи испит. Када се у табелу испита унесе положен испит, за студента који је положио тај испит је потребно изменити његове број поена до краја у табели из дела под а. Број поена до краја се умањује за број поена колико носи унети положен предмет.
За тестирање се могу унети наредни подаци:
(индекс, оцена, поени, идпредмета, статус, школска година, ознака)
(20160369, 10, 100, 2482, о, 2019, јан1)
(20160370, 10, 100, 2352, о, 2019, јан1).
 - Написати наредбу којом се из табеле под (а) бришу студенти који више нису апсолвенти, односно студенти који имају 0 поена до краја.
- Написати упит који издваја име и презиме студента који је рођен у Београду или је положио испит у свим испитним роковима у којима је добијена оцена 8. Задатак решити на:

За решавање задатка користити релације:

`dosije(indeks, ime, prezime, mestorodjenja, datupisa)`

`predmet(idpredmeta, oznaka, naziv, espb)`

`ispitnirok(skgodina, oznakaroka, naziv)`

`ispit(indeks, idpredmeta, skgodina, oznakaroka, ocena, datpolaganja, bodovi)`

5.
 - a) Шта значи принцип релационог затворења? Наведите његове последице.
 - б) Набројити и објаснити бар пет основних функција које обавља СУБП. Напомена - само навођење функција без описа неће бити признати као решење задатка.
 - в) Прецизно опишите ефекат извршавања OLAP функција `row_number()`, `rank()` и `dense_rank()`.
 - г) Које услове мора да испуни поглед да би могао да се ажурира у систему DB2? Шта означава опција *WITH CHECK* приликом дефинисања погледа?

6.
 - a) Нека је дата табела:

```
create table dosije_xml (podatak xml not null)
```

```
<?xml version="1.0" encoding="UTF-8" ?>
  <student broj_indeksa="20210021">
    <ime>Miloš</ime>
    <prezime>Perić</prezime>
    <datum_upisa>"06.07.2021"</datum_upisa>
    <datum_rodjenja>"20.01.2001"</datum_rodjenja>
    <telefon tip="mobilni">064-123-456</telefon>
    <telefon tip="fiksni">011-123-456</telefon>
    <prosek>9.45</prosek>
  </student>
```

Написати упит који враћа број индекса, име, презиме, датум уписа и просек оцена студената чије је име 'Veselin'.

- б) Написати SQL упит којим се издвајају име, презиме, број индекса студената и позиција трећег појављивања ниске 'ад' од краја у месту рођења студената или 0 уколико таква ниска не постоји. На пример, ако је место рођења 'Ада над градом Нови Сад', тада треба вратити вредност 18, док за ниску 'Ада над Новим Садом' треба вратити 0.
7.
 - a) Навести дефиниције Бојс-Кодове, IV, и V нормалне форме.
 - б) Нека је дата релациона променљива $R = \{A, B, C, D, E, F\}$ и скуп F ФЗ:

1) $A \longrightarrow BF$

6) $D \longrightarrow C$

2) $ACD \longrightarrow E$

7) $DE \longrightarrow A$

3) $BD \longrightarrow AE$

8) $DF \longrightarrow BC$

4) $BCF \longrightarrow E$

9) $EF \longrightarrow D$

5) $CE \longrightarrow D$

a) Одредити нередуцибилни покривач скупа функционалних зависности F .

б) Одредити све кандидате за кључ релације R .

в) Одредити скуп ФЗ које важе у релацији $R_1 = \{A, B, C, D\}$ која је пројекција релације R .

ОБАВЕЗНО ОБРАЗЛОЖИТИ СВЕ СВЕ КОРАКЕ У РАДУ. Навођење само резултата појединих корака (нпр. затворења скупа атрибута без објашњења поступка како се до њега дошло) неће бити признато као делимично урађен задатак.

8.
 - a) Детаљно објаснити двофазни протокол закључавања - шта је, када се примењује и које су последице његове примене.
 - б) Објаснити улогу *log* датотека у релационим базама. Детаљно описати ARIES алгоритам и навести његове предности у односу на алгоритме опоравка претходне генерације.
 - в) Навести бар 5 трансформација израза које се примењују при оптимизацији упита.