



## Базе података 1

(13E113БП1)

- јануарски испитни рок -

### Група Б

Посматра се део базе података једне туристичке агенције. У овој бази се прате аранжмани, путници, хотели, места и превоз. Аранжман карактерише назив, цена, датум поласка и доласка и место поласка. Путника карактеришу име, презиме и пол ('М' – мушки, 'З' – женски). Путник може да уплаћује аранжман, где се прати датум уплате. У систему се прате хотели који се могу наћи у понуди аранжмана. Аранжман има свој план који се састоји из листе места које обухвата. Тим планом је могуће у неком месту одсести у неком од хотела. Хотел карактерише назив хотела, број звездица (1 до 5), капацитет, цена ноћења. Места су дефинисана по називу држави и континенту на коме се налази. Туристичкој агенцији пристижу понуде за превоз које се прате и на основу којих се формирају аранжмани.

У наставку је дата релациона шема посматраног дела базе податка.

#### Putnik (IdPut, ImeIPrezime)

IdPut	- цео број, идентификује путника
ImeIPrezime	- низ до 50 знакова, обавезно
Pol	- један знак, обавезно поље, могуће вредности су 'М' и 'З' ( 'М' – мушки, 'З' – женски)

#### Aranzman (IdAra, Naziv, DatumOd, DatumDo, Cena)

IdAra	- цео број, идентификује аранжман
Naziv	- низ до 50 знакова, обавезно
DatumOd	- цео број, обавезно поље, датум у формату (ууууммдд)
DatumDo	- цео број, обавезно поље, датум у формату (ууууммдд)
Cena	- цео број, обавезно поље, вредности већа од 0

#### Uplatio (IdPut, IdAra, Datum)

IdPut	- страни кључ (табела Putnik), обавезно
IdAra	- страни кључ (табела Aranzman), обавезно
Datum	- цео број, обавезно поље, датум у формату (ууууммдд)

Напомена: Комбинација IdPut и IdAra је јединствена.

#### Drzava (IdDrz, Ime, Kontinent)

IdDrz	- цео број, идентификује државу
Ime	- низ до 50 знакова, обавезно, јединствено
Kontinent	- цео број, обавезно поље, могуће вредности у опсегу 1 до 7

Напомена: Значење атрибута Kontinent: 1-Азија, 2-Антарктик, 3-Аустралија, 4-Африка, 5-Европа, 6-Јужна Америка, 7-Северна Америка

Mesto (IdMes, Naziv, IdDrz)

IdMes	- цео број, идентификује место
Naziv	- низ до 50 знакова, обавезно, јединствено
IdDrz	- страни кључ (табела Drzava), обавезно

Hotel (IdHot, Naziv, BrojZvezdica, Kapacitet, Cena, IdMes)

IdHot	- цео број, идентификује хотел, аутоматско додељивање наредног идентификатора
Naziv	- низ до 50 знакова, обавезно
BrojZvezdica	- цео број, обавезно поље, могуће вредности у опсегу 1 до 5
Kapacitet	- цео број, обавезно поље, вредност већа од 0
Cena	- цео број, обавезно поље, вредност већа од 0
IdMes	- страни кључ (табела Mesto), обавезно

PlanAranzmana (IdAra, IdMes, RB, IdHot, BrojNocenja)

IdAra	- страни кључ (табела Aranzman), обавезно
IdMes	- страни кључ (табела Mesto), обавезно
RB	- цео број, обавезно
IdHot	- страни кључ (табела Hotel), није обавезно
BrojNocenja	- цео број, није обавезно поље, вредност већа од 0

Напомена: RB представља редни број по коме се обилази место у једном аранжману. RB се додељује редом почевши од 1. Комбинација IdAra, IdMes и RB је јединствена.

Ако је IdHot недефинисан (NULL), онда се у том месту не ноћи.

Сматрати да је на нивоу базе обезбеђено да се IdMes хотела поклапа са IdMes плана аранжмана.

Prevoz (IdMesOd, IdMesDo, Sredstvo, Cena, Razdaljina)

IdMesOd	- страни кључ (табела Mesto), обавезно
IdMesDo	- страни кључ (табела Mesto), обавезно
Sredstvo	- низ до 7 знакова, обавезно поље, могуће вредности су 'Autobus', 'Avion' и 'Brod'
Cena	- цео број, обавезно поље, вредност већа од 0
Razdaljina	- цео број, обавезно поље, вредност већа од 0

Напомена: Комбинација IdMesOd, IdMesDo и Sredstvo је јединствена.

### Задатак 1 [3 поена]

Потребно је направити SQL упит који исписује све хотеле који припадају *Continental* ланцу хотела, као и назив места где се налази. Овај ланац хотела именује своје хотеле на следећи начин. Назив почиње са речју “Continental”, након чега следи бланко знак и потом га прати звездица, а након чега следи број који представља број звездица хотела, а након тога следи бланко, па средња црта, па бланко и онда у наставку конкретно име хотела. *Continental* хотели имају број звездица хотела већи или једнак 4.

Резултат дати у форми: IdHot, Naziv, BrojZvezdica, Mesto

У Cactus-у користити таб: Zadatak 1s

---

```
SELECT Hotel.IdHot, Hotel.Naziv, Hotel.BrojZvezdica, Mesto.Naziv AS Mesto
FROM Hotel JOIN Mesto USING (IdMes)
WHERE (Hotel.Naziv LIKE 'Continental *4 - _%' AND BrojZvezdica = 4)
      OR (Hotel.Naziv LIKE 'Continental *5 - _%' AND BrojZvezdica = 5)
```

---

### Задатак 2 [3 поена]

Потребно је направити SQL упит који исписује план аранжмана за аранжман са називом 'Godina Nova i odmor iz snova', тако што исписује редни број града који се посећује, назив града, као и ако се у том граду ноћи, онда је потребно навести и име хотела и његов број звездица. Уколико се у граду не ноћи, треба изоставити информације о хотелу (NULL вредности). Резултат треба сортирати опадајуће по редном броју.

Резултат дати у форми: Grad, RB, Hotel, BrojZvezdica

У Cactus-у користити таб: Zadatak 2

---

```
SELECT Mesto.Naziv AS Grad, RB, Hotel.Naziv AS Hotel, Hotel.BrojZvezdica
FROM Aranzman NATURAL JOIN PlanAranzmana JOIN Mesto USING(IdMes)
      LEFT OUTER JOIN Hotel USING (IdHot)
WHERE Aranzman.Naziv='Godina Nova i odmor iz snova'
ORDER BY PlanAranzmana.RB DESC;
```

---

### Задатак 3 [3 поена]

Потребно је направити SQL упит који приказује аранжмане код којих се по плану аранжмана тачно два пута преноћи (у размацама) у истом хотелу и то оба пута по два дана.

Резултат дати у форми: IdAra, Naziv

У Сactus-у користити таб: Zadatak 3

---

```
SELECT IdAra, Naziv
FROM Aranzman
WHERE IdAra IN (
    SELECT IdAra
    FROM PlanAranzmana
    WHERE BrojNocenja IS NOT NULL
    GROUP BY IdAra, IdHot
    HAVING COUNT(*) = 2 AND SUM(BrojNocenja) = 4 AND MIN(BrojNocenja) = 2
)
```

---

### Задатак 4 [3 поена]

Потребно је направити SQL упит који исписује за свако место које је у Азији или у Европи колико има хотела са 3, 4 и 5 звездица. Сортирати резултат растуће по Naziv-у места.

Резултат дати у форми: IdMes, Naziv, 3 Zv, 4 Zv, 5 Zv

У Сactus-у користити таб: Zadatak 4

---

```
SELECT IdMes, Naziv,
    (SELECT COUNT(*) FROM Hotel WHERE BrojZvezdica=3 AND Hotel.IdMes=m.IdMes)
    AS "3 Zv",
    (SELECT COUNT(*) FROM Hotel WHERE BrojZvezdica=4 AND Hotel.IdMes=m.IdMes)
    AS "4 Zv",
    (SELECT COUNT(*) FROM Hotel WHERE BrojZvezdica=5 AND Hotel.IdMes=m.IdMes)
    AS "5 Zv"
FROM Mesto m NATURAL JOIN Drzava
WHERE Kontinent=1 OR Kontinent=5
ORDER BY Naziv;
```

---

#### Задатак 5 [4 поена]

Потребно је направити SQL скрипту која ако постоји табела **Hotel** избацује табелу **Hotel** из шеме, а затим формира нову табелу **Hotel** која треба да има одговарајућу структуру и ограничења.

У Cactus-у користити таб Задатак 5.

---

```
DROP TABLE IF EXISTS Hotel;
```

```
CREATE TABLE Hotel (  
    IdHot      INTEGER PRIMARY KEY AUTOINCREMENT,  
    Naziv      VARCHAR(50) NOT NULL,  
    BrojZvezdica INTEGER NOT NULL CHECK (BrojZvezdica BETWEEN 1 AND 5),  
    Kapacitet  INTEGER NOT NULL CHECK (Kapacitet > 0),  
    Cena       INTEGER NOT NULL CHECK (Cena > 0),  
    IdMes      INTEGER NOT NULL,  
    FOREIGN KEY(IdMes) REFERENCES Mesto(IdMes)  
);
```

---

#### Задатак 6 [4 поена]

Потребно је направити SQL упит који исписује хотеле заједно са местом, државом и пуним називом континента на коме се налази. Сортирати по IdHot опадајуће.

Резултат дати у форми: IdHot, Naziv, BrojZvezdica, Mesto, Drzava, Kontinent

У Cactus-у користити таб: Задатак 6

**Није дозвољено коришћење погледа.**

---

```
SELECT IdHot, Hotel.Naziv, BrojZvezdica, Mesto.Naziv AS Mesto, Drzava.Ime AS Drzava,  
    CASE Kontinent  
        WHEN 1 THEN 'Azija'  
        WHEN 2 THEN 'Antartik'  
        WHEN 3 THEN 'Australija'  
        WHEN 4 THEN 'Afrika'  
        WHEN 5 THEN 'Evropa'  
        WHEN 6 THEN 'Juzna Amerika'  
        ELSE 'Severna Amerika'  
    END AS Kontinent  
FROM Hotel JOIN Mesto USING (IdMes) NATURAL JOIN Drzava  
ORDER BY IdHot DESC;
```

---

Задатак 7 [4 поена]

Потребно је направити SQL упит који исписује аранжмане у оквиру којег се преноћи (барем једном) у хотелима са барем три различите категорије (различит број звездица). Сортирати по IdAra растуће.

Резултат дати у форми: IdAra, Naziv

У Сactus-у користити таб: Zadatak 7

**Није дозвољено коришћење погледа.**

---

```
SELECT IdAra, Naziv
FROM Aranzman
WHERE IdAra IN (
    SELECT IdAra
    FROM PlanAranzmana NATURAL JOIN Hotel
    GROUP BY IdAra
    HAVING COUNT( DISTINCT BrojZvezdica ) >= 3
);
```

---

Задатак 8 [4 поена]

Потребно је направити SQL упит којим се увећава цена за 7.5% оних аранжмана чији је укупан број ноћења већи од просечног броја ноћења по аранжману, за аранжмане који имају бар неко ноћење.

Након ажурираних података, исписати све податке из табеле *Aranzman*.

Резултат дати у форми: IdAra, Naziv, DatumOd, DatumDo, Cena

У Сactus-у користити таб: Zadatak 8

**Није дозвољено коришћење погледа.**

---

```
UPDATE Aranzman
SET Cena = Cena * 1.075
WHERE IdAra IN (
    SELECT IdAra
    FROM PlanAranzmana
    WHERE IdHot IS NOT NULL
    GROUP BY IdAra
    HAVING SUM( BrojNocenja ) > (
        SELECT SUM(BrojNocenja) * 1.0 / COUNT( DISTINCT IdAra)
        FROM PlanAranzmana
        WHERE IdHot IS NOT NULL
    )
);

SELECT * FROM Aranzman;
```

---

Задатак 9 [5 поена]

Потребно је направити SQL скрипту који формира нови аранжман 'Exit 2019' који би кретао и завршавао би се 27. јула 2019. Релација која обухвата аранжман је Beograd-Novi Sad-Beograd. За идентификацију аранжмана треба узети за једну већу у односу на тренутно највећу идентификацију. Туристичка агенција би за цену ставила 10% већу у односу на трошкове превоза. За превоз би се тражила прво најјефтинија понуда бродом, па онда најјефтинија понуда аутобусом. Ако не постоји ни једна понуда бродом или аутобусом, за цену поставити 80.

У Сactusu-у користити таб Zadatak 9 за додавање новог аранжмана и приказивање новоформираног садржаја табеле *Aranzman*.

---

```
INSERT INTO Aranzman VALUES (  
    (SELECT MAX(IdAra)+1 FROM Aranzman),  
    'Exit 2019',  
    20190727,  
    20190727,  
    (SELECT COALESCE ((  
        SELECT MIN(p1.Cena+p2.Cena)*1.1  
        FROM Prevoz p1, Prevoz p2, Mesto BG, Mesto NS  
        WHERE BG.Naziv='Beograd' AND NS.Naziv='Novi Sad'  
              AND p1.IdMesOd=BG.IdMes AND p1.IdMesDo=NS.IdMes  
              AND p2.IdMesOd=NS.IdMes AND p2.IdMesDo=BG.IdMes  
              AND p1.Sredstvo='Brod' AND p2.Sredstvo='Brod'  
    ),(  
        SELECT MIN(p1.Cena+p2.Cena)*1.1  
        FROM Prevoz p1, Prevoz p2, Mesto BG, Mesto NS  
        WHERE BG.Naziv='Beograd' AND NS.Naziv='Novi Sad'  
              AND p1.IdMesOd=BG.IdMes AND p1.IdMesDo=NS.IdMes  
              AND p2.IdMesOd=NS.IdMes AND p2.IdMesDo=BG.IdMes  
              AND p1.Sredstvo='Autobus' AND p2.Sredstvo='Autobus'  
    ),80)  
    )  
);  
  
SELECT * FROM Aranzman;
```

---

Задатак 10 [5 поена]

Потребно је направити SQL упит који исписује имена места и преосталу количину новца које је могуће посетити кренувши из *Beograd*-а са 350 EUR. За превозно средство се користи искључиво аутобус. Ако је могуће на више начина доћи до неког места, потребно је исписати онај резултат код којег остаје већа количина новца. Резултат сортирати опадајуће по преосталој количини новца, а затим по називу места. У резултату не треба да се нађе *Beograd*.

Резултат дати у форми: IdMes, Naziv, Preostalo

У Сactus-у користити таб: Zadatak 10

**Није дозвољено коришћење погледа.**

---

```
WITH RECURSIVE Putanje(Id, Novac) AS
(
    SELECT (SELECT IdMes FROM Mesto WHERE Naziv='Beograd'), 350
    UNION
    SELECT Pr.IdMesDo, (Pu.Novac-Pr.Cena)
    FROM Prevoz Pr JOIN Putanje Pu ON (Pr.IdMesOd = Pu.Id)
    WHERE Pr.Sredstvo='Autobus' AND (Pu.Novac-Pr.Cena)>=0
)
SELECT P.Id AS IdMes, M.Naziv, MAX(P.Novac) AS Preostalo
FROM Putanje P JOIN Mesto M ON P.Id = M.IdMes
WHERE M.Naziv != 'Beograd'
GROUP BY P.Id, M.Naziv
ORDER BY 3 DESC, 2 DESC;
```

---



Задатак 11 [6 поена]

Потребно је направити SQL упит који треба да прикаже којим превозним средством је могуће доћи у *Бес*, ако се полази из *Београд*-а и притом се у сваком граду који се на путу посети наизменично промени превозно средство: *Autobus* → *Avion* → *Brod* → *Autobus* → ... . У случају да није могуће на овај начин доћи до *Бес*-а, резултат треба да садржи један податак у коме пише "*Nemoguće*". Ако је могуће доћи до *Бес*-а, онда је потребно у сваком реду исписати последње коришћено превозно средство (без понављања).

Резултат дати у форми: Sredstvo

У Cactus-у користити таб: Zadatak 11

---

```
WITH RECURSIVE Doseznost (IdMes, PSredstvo) AS
(
    SELECT IdMes, 2 FROM Mesto WHERE Naziv='Beograd'
    UNION
    SELECT IdMesDo, (PSredstvo + 1)%3
    FROM Prevoz, Doseznost
    WHERE IdMesOd= Doseznost.IdMes AND Sredstvo= (CASE (PSredstvo + 1) % 3
        WHEN 0 THEN 'Autobus'
        WHEN 1 THEN 'Avion'
        ELSE 'Brod' END
    )
)
SELECT DISTINCT CASE PSredstvo
    WHEN 0 THEN 'Autobus'
    WHEN 1 THEN 'Avion'
    ELSE 'Brod' END AS Sredstvo
FROM Doseznost
WHERE IdMes = (SELECT MAX(IdMes) FROM Mesto WHERE Naziv='Bec')
UNION
SELECT 'Nemoguće' AS Sredstvo
WHERE NOT EXISTS(
    SELECT *
    FROM Doseznost
    WHERE IdMes = (SELECT MAX(IdMes) FROM Mesto WHERE Naziv='Bec')
);
```

---

Задатак 12 [6 поена]

Потребно је направити SQL упит који треба да израчуна цену потенцијалног аранжмана (аранжман још није унет у базу). Цена се формира тако што се на све трошкове аранжмана (превоз и цена хотела) дода 10%. Аранжман би требало да обухвата релацију *Beograd-Budmpesta-Krakov-Beograd*. *Budimpesta* се само обилази – не одседа се у хотелу, док се у *Krakov*-у спава 10 дана у најјефтинијем хотелу који има више од 3 звезде. За цену превоза се тражи најјефтинија понуда, тако да се на релацији *Beograd-Budimpesta-Krakov* користи аутобус, а на релацији *Krakov-Beograd* користи авион. Претпоставити да се у бази сигурно налазе одговарајући подаци.

Резултат дати у форми: Cena

У Cactus-у користити таб: Zadatak 12

---

WITH

Beograd (IdMes) AS (

SELECT MAX(IdMes) FROM Mesto WHERE Naziv= 'Beograd'),

Budimpesta (IdMes) AS (

SELECT MAX(IdMes) FROM Mesto WHERE Naziv= 'Budimpesta'),

Krakov (IdMes) AS (

SELECT MAX(IdMes) FROM Mesto WHERE Naziv= 'Krakov')

SELECT SUM(Cena) \* 1.10 AS Cena

FROM (

SELECT 10 \* MIN(Cena) AS Cena

FROM Hotel NATURAL JOIN Krakov

WHERE BrojZvezdica>3

UNION ALL

SELECT MIN(Cena)

FROM Prevoz, Beograd, Budimpesta

WHERE Sredstvo='Autobus' AND IdMesOd =Beograd.IdMes AND IdMesDo =Budimpesta.IdMes

UNION ALL

SELECT MIN(Cena)

FROM Prevoz, Budimpesta, Krakov

WHERE Sredstvo='Autobus' AND IdMesOd = Budimpesta.IdMes AND IdMesDo = Krakov.IdMes

UNION ALL

SELECT MIN(Cena)

FROM Prevoz, Krakov, Beograd

WHERE Sredstvo='Avion' AND IdMesOd = Krakov.IdMes AND IdMesDo = Beograd.IdMes

) X;

---