

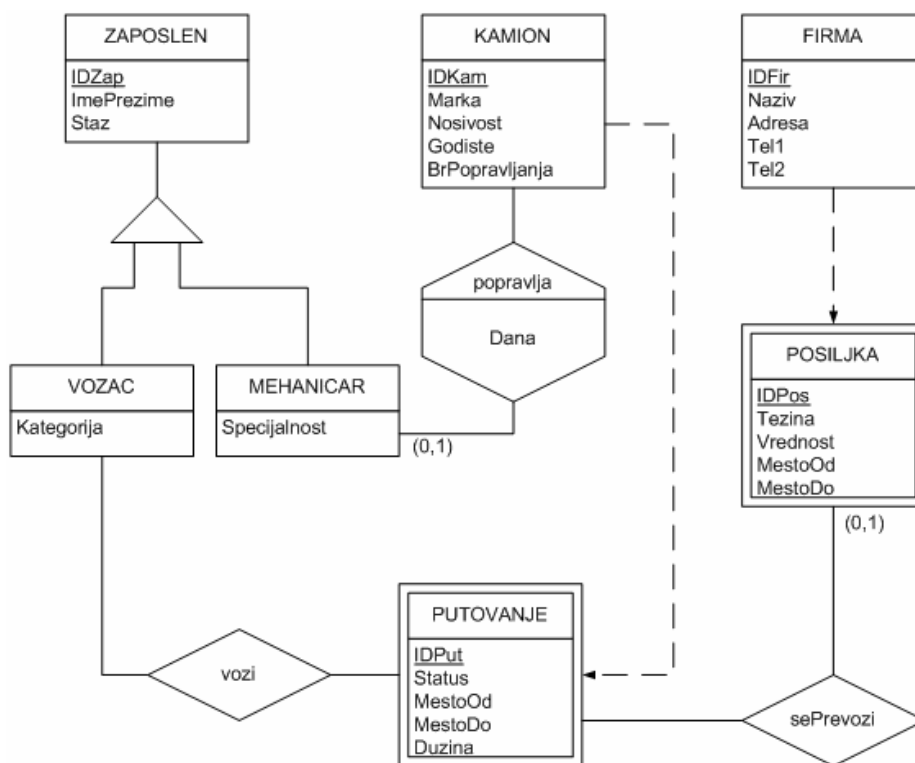


Базе података 1

- показна лабораторијска вежба -

Посматрани систем је једна компанија за друмски превоз терета. Компанија располаже са одређеним бројем камиона (прате се марка, носивост, годиште, број поправљања). У компанији раде запослени за које се прате име, презиме, стаж. Део запослених су возачи, за које се додатно прати и категорија а део су механичари за које се додатно прати марка возила за коју су специјализовани. Када је камион у квару, води се евиденција о томе који механичар га поправља као и предвиђено време за поправку (у данима). Компанија послује тако што од фирми (прати се назив, адреса и два броја телефона) прима пошиљке одређених тежина и вредности које треба пренети од полазног до одредишног места, за које се такође прати назив. При томе се евидентира свако путовање камиона, у смислу релације (од-до), учешћа возача (до два) и пошиљки које се преносе (једна или више њих).

У наставку је дата релациона шема посматраног дела базе податка.



Kamion (IDKam, Marka, Nosivost, Godiste, BrPopravljanja)

- IDKam - цео број, идентификује камион, аутоматско додељивање наредног идентификатора
- Marka - низ до 25 знакова, обавезно
- Nosivost - цео број, обавезно
- Godiste - цео број већи од 1900, није обавезно
- BrPopravljanja - цео број, обавезно, подразумевано има вредност 0

Zaposlen (IDZap, ImePrezime, Staz)

IDZap	- цео број, идентификује запосленог, аутоматско додељивање наредног идентификатора
ImePrezime	- низ до 40 знакова, обавезно
Staz	- цео број већи или једнак 0, није обавезно

Vozac (IDZap, Kategorija)

IDZap	- страни кључ (табела Zaposlen), идентификује механичара
Kategorija	- низ од 5 знакова, није обавезно

Mehanicar (IDZap, Specijalnost)

IDZap	- страни кључ (табела Zaposlen), идентификује механичара
Specijalnost	- низ до 15 знакова, није обавезно

Firma (IDFir, Naziv, Adresa, Tel1, Tel2)

IDFir	- цео број, идентификује фирму, аутоматско додељивање наредног идентификатора
Naziv	- низ до 20 знакова, јединствено, обавезно
Adresa	- низ до 60 знакова, није обавезно
Tel1	- низ од 15 знакова, није обавезно
Tel2	- низ од 15 знакова, није обавезно

Posiljka (IDPos, Tezina, Vrednost, MestoOd, MestoDo, IDFir)

IDPos	- цео број, идентификује пошиљку, аутоматско додељивање наредног идентификатора
Tezina	- цео број, обавезно
Vrednost	- цео број, није обавезно
MestoOd	- низ до 25 знакова, није обавезно
MestoDo	- низ до 25 знакова, није обавезно
IDFir	- страни кључ (табела Firma), обавезно

Putovanje (IDPut, Status, MestoOd, MestoDo, Duzina, IDKam)

IDPut	- цео број, идентификује путовање, аутоматско додељивање наредног идентификатора
Status	- знак, није обавезно
MestoOd	- низ до 25 знакова, обавезно
MestoDo	- низ до 25 знакова, обавезно
Duzina	- цео број, није обавезно
IDKam	- страни кључ (табела Kamion), обавезно

Popravljа (IDZap, IDKam, Dana)

IDZap	- страни кључ (табела Mehanicar), обавезно, део идентификације
IDKam	- страни кључ (табела Kamion), обавезно, део идентификације
Dana	- цео број, није обавезно

Vozi (IDZap, IDPut)

IDZap	- страни кључ (табела Vozac), обавезно, део идентификације
IDPut	- страни кључ (табела Putovanje), обавезно, део идентификације

Popravljа (IDPos, IDPut)

IDPos	- страни кључ (табела Posiljka), обавезно, део идентификације
IDPut	- страни кључ (табела Putovanje), обавезно, део идентификације

Задатак 1

Потребно је направити SQL упит који исписује информације о свим пошиљкама. Резултат сортирати по IDPos растуће.

Резултат дати у форми: IDPos, Tezina, Vrednost, MestoOd, MestoDo, IDFir

У Сactus-у користити таб: Zadatak 1

Није дозвољено коришћење погледа.

```
SELECT *  
FROM Posiljka  
ORDER BY IDPos
```

Задатак 2

Потребно је направити SQL упит који исписује тежину свих пошиљки фирме "Tropik".

Резултат дати у форми: UkupnaTezina

У Сactus-у користити таб: Zadatak 2

Није дозвољено коришћење погледа.

```
SELECT SUM(Tezina) AS UkupnaTezina  
FROM Posiljka P, Firma F  
WHERE F.IDFir = P.IDFir AND F.Naziv='Tropik'
```

Задатак 3

Потребно је направити SQL скрипту која додаје нов камион у базу података, а затим исписује све камионе. Камион је марке "Mercedes", носивости 15 и 2022. годиште. Узети да идентификатор камиона има вредност 20. Резултат сортирати по IdKam растуће.

Резултат дати у форми: Naziv

У Сactus-у користити таб: Zadatak 3

Није дозвољено коришћење погледа.

```
INSERT INTO Kamion (IDKam,Marka, Nosivost, Godiste, BrPopravljanja)  
VALUES (20, 'Mercedes', 15, 2022, 0);
```

```
SELECT *  
FROM Kamion  
ORDER BY IdKam
```

Задатак 4

Потребно је направити SQL скрипту која ажурира вредност свих пошиљки фирме "Petrolept" тако што им вредност увећа за 250, а затим је потребно исписати све пошиљке. Резултат сортирати по IdPos растуће.

Резултат дати у форми: IdPos, Tezina, Vrednost, MestoOd, MestoDo, IdFir

У Сactus-у користити таб: Zadatak 4

Није дозвољено коришћење погледа.

```
UPDATE Posiljka
SET Vrednost = Vrednost +250
WHERE IdFir = (SELECT MAX(IdFir) FROM Firma WHERE Naziv = 'Petrolept');
```

```
SELECT *
FROM Posiljka
ORDER BY IdPos
```

Задатак 5

Потребно је направити SQL скрипту која брише сву тренутне поправке које трају више од 2 дана, а затим исписује сву преостале поправке. Резултат сортирати по IdZap растуће, а затим растуће по IdKam.

Резултат дати у форми: IdZap, IdKam, Dana

У Сactus-у користити таб: Zadatak 5

Није дозвољено коришћење погледа.

```
DELETE FROM Popravlja
WHERE Dana >2;
```

```
SELECT *
FROM Popravlja
ORDER BY IdZap, IdKam
```

Задатак 6

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

У Cactus-у користити таб: Zadatak 6

```
DROP TABLE IF EXISTS Kamion;
```

```
CREATE TABLE Kamion (  
  IDKam INTEGER PRIMARY KEY AUTOINCREMENT,  
  Marka VARCHAR(25) NOT NULL,  
  Nosivost INTEGER NOT NULL CHECK (Nosivost>0),  
  Godiste INTEGER CHECK (Godiste>1900),  
  BrPopravljanja INTEGER DEFAULT 0 NOT NULL  
);
```

Задатак 7

Потребно је направити SQL упит који исписује фирме које су имале пошиљку највеће вредности. Резултат сортирати по Naziv растуће.

Резултат дати у форми: Naziv

У Cactus-у користити таб: Zadatak 7

Није дозвољено коришћење погледа.

```
SELECT F.Naziv  
FROM Posiljka P, Firma F  
WHERE F.IDFir = P.IDFir AND P.Vrednost=(SELECT MAX(Vrednost) FROM Posiljka)  
ORDER BY F.Naziv
```

Задатак 8

Потребно је направити SQL упит који за све фирме које су имале бар једну пошиљку исписује назива фирми и просечне вредности пошиљки које су имали. Резултат сортирати по Naziv опадајуће.

Резултат дати у форми: Naziv, ProsecnaVrednost

У Cactus-у користити таб: Zadatak 8

Није дозвољено коришћење погледа.

```
SELECT F.Naziv, AVG(P.Vrednost) AS ProsecnaVrednost  
FROM Posiljka P, Firma F  
WHERE F.IDFir = P.IDFir  
GROUP BY F.IDFir, F.Naziv  
ORDER BY F.Naziv DESC
```

Задатак 9

Потребно је направити SQL упит који приказује механичаре који тренутно не поправљају ни један камион. Резултат сортирати по IDZap опадајуће.

Резултат дати у форми: IDZap, ImePrezime

У Сactus-у користити таб: Zadatak 9

Није дозвољено коришћење погледа.

```
SELECT Z.IdZap, ImePrezime
FROM Zaposlen Z, Mehanicar M
WHERE Z.IDZap=M.IDZap
      AND NOT EXISTS( SELECT * FROM Popravlja P WHERE P.IDZap=M.IDZap)
ORDER BY Z.IdZap DESC
```

Задатак 10

Потребно је направити SQL упит за механичаре који тренутно поправљају камионе, примазује њихово име и презиме, марку и носивост камиона који поправљају, као и колико дана ће учествовати у поправци. Резултат сортирати по ImePrezime растуће.

Резултат дати у форми: ImePrezime, Marka, Nosivost, Dana

У Сactus-у користити таб: Zadatak 10

Није дозвољено коришћење погледа.

```
SELECT Z.ImePrezime, K.Marka, K.Nosivost, P.Dana
FROM Zaposlen Z, Mehanicar M, Popravlja P, Kamion K
WHERE Z.IDZap=M.IDZap AND P.IDZap=M.IDZap AND P.IDKam=K.IDKam
ORDER BY Z.ImePrezime
```

Задатак 11

Потребно је направити SQL упит који приказује фирме које имају пошиљке које нису нити послате, нити испланиране, уз информације о укупној тежини тих пошиљки. Резултат сортирати по Naziv растуће.

Резултат дати у форми: Naziv, UkupnaTezina

У Сactus-у користити таб: Zadatak 11

Није дозвољено коришћење погледа.

```
SELECT Naziv, SUM(Tezina) AS UkupnaTezina
FROM Firma F JOIN Posiljka P USING(IDFir)
WHERE NOT EXISTS(SELECT * FROM sePrevozi SP WHERE SP.IDPos=P.IDPos)
GROUP BY IDFir, Naziv
ORDER BY Naziv
```

Задатак 12

Потребно је направити SQL упит који исписује који камиони су коришћени за превоз пошиљки у укупној тежини до 7000 кг. Резултат сортирати по Naziv растуће, а затим растуће по Nosivost.

Резултат дати у форми: Marka, Nosivost

У Сactus-у користити таб: Zadatak 12

Није дозвољено коришћење погледа.

```
SELECT DISTINCT Marka, Nosivost
FROM Kamion Kam JOIN Putovanje PUT USING(IDKam)
WHERE (
    SELECT SUM(P.Tezina)
    FROM Posiljka P JOIN sePrevozi SP USING(IDPos)
    WHERE PUT.IDPut=SP.IDPut
) <=7000
ORDER BY Marka, Nosivost
```

Задатак 13

Потребно је направити SQL упит који приказује све возаче који су некада возили пошиљку од фирме "Petrolept" као и марку камиона који је коришћен том приликом и колико пута је та марка коришћена у том случају. Резултат сортирати по ImePrezime растуће, а затим растуће по Marka.

Резултат дати у форми: ImePrezime, Marka, Broj posiljki

У Сactus-у користити таб: Zadatak 13

Није дозвољено коришћење погледа.

```
SELECT ImePrezime, Marka, COUNT(*) AS "Broj posiljki"
FROM Zaposlen JOIN Vozac USING(IDZap) JOIN Vozi USING(IDZap)
    JOIN Putovanje USING(IdPut) JOIN sePrevozi USING(IDPut)
    JOIN Kamion USING(IDKam) JOIN Posiljka USING(IDPos)
    JOIN Firma USING(IdFir)
WHERE Firma.Naziv='Petrolept'
GROUP BY ImePrezime, Marka
```

Задатак 14

Потребно је направити SQL упит који исписује колико терета су превезли возачи са највећим стажом међу свим возачима.

Резултат дати у форми: Ukupna tezina

У Сactus-у користити таб: Zadatak 14

Није дозвољено коришћење погледа.

```
SELECT SUM(Tezina) AS "Ukupna tezina"
FROM Zaposlen JOIN Vozac USING(IDZap) JOIN Vozi USING(IDZap)
      JOIN Putovanje USING(IDPut) JOIN sePrevozi USING(IDPut)
      JOIN Posiljka USING(IDPos)
WHERE Staz = (SELECT MAX(Staz) FROM Vozac JOIN Zaposlen USING(IDZap))
```

Задатак 15

Потребно је направити SQL упит који приказује колика је укупна вредност и укупна тежина терета коју су превезли камиони који се тренутно поправљају, а који су већ имали мање од просечног броја поправки свих постојећих камиона, а да је те камионе у неком тренутку возио Драган Милутиновић.

Резултат дати у форми: Vrednost, Tezina

У Сactus-у користити таб: Zadatak 15

Није дозвољено коришћење погледа.

```
SELECT SUM(Posiljka.Vrednost) AS Vrednost, SUM(Posiljka.Tezina) AS Tezina
FROM Kamion JOIN Putovanje USING (IdKam)
      JOIN sePrevozi USING(IDPut) JOIN Posiljka USING(IDPos)
WHERE IDKam IN (
      SELECT IDKam
      FROM Zaposlen JOIN Vozac USING (IDZap) JOIN Vozi USING(IDZap)
      WHERE ImePrezime='Dragan Milutinovic'
)
AND BrPopravljanja<( SELECT AVG(BrPopravljanja) FROM Kamion)
AND IdKam IN ( SELECT IDKam FROM Popravljaj)
```
