



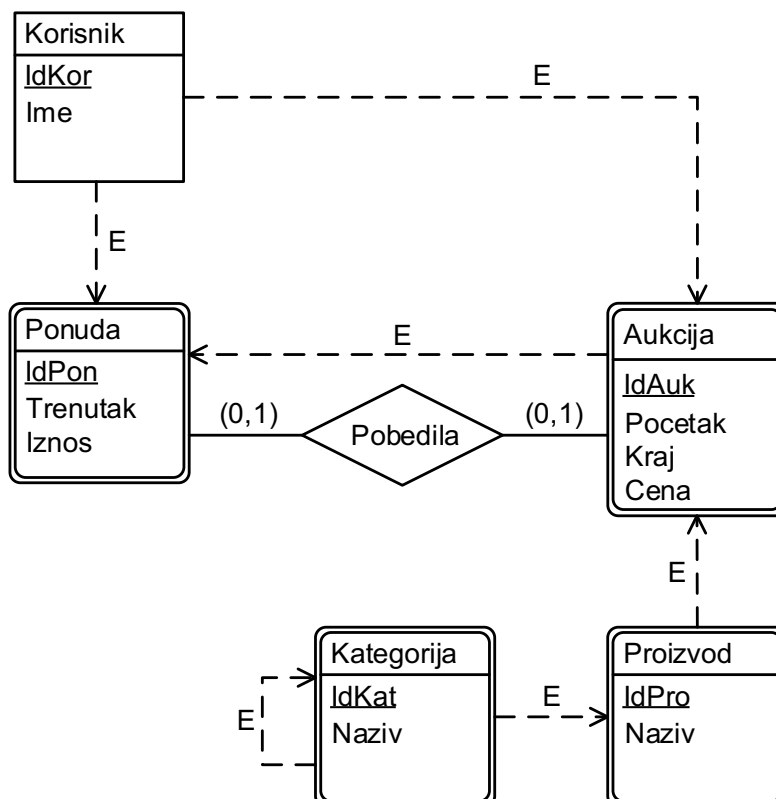
# Базе података 1

(13С112БП1, 13Е113БП1)  
- фебруарски испитни рок –

## Група А

Посматра се систем за аукције у коме корисници могу да креирају аукције на којима продају производе. Приликом креирања аукције, познати су производ, време почетка аукције и почетна цена, док се време краја аукције дефинише у моменту проглашења краја аукције. Сваки корисник може да даје већи број понуда док аукција траје. Износ сваке понуде мора да буде строго већа од износа претходно дате понуде за исту аукцију. У тренутку када се прогласи крај, уколико је била бар једна понуда за ту аукцију аукција се сматра успешном и проглашава се победник те аукције. У супротном, аукција се сматра неуспешном и победник не постоји. Сваки производ припада некој од категорија у хијерархији категорија које су евидентиране у систему. Сви датуми у систему су из периода од 1941. године до 2040. године.

У наставку је дата релациона шема посматраног дела базе податка.



#### Kategorija (IdKat, Naziv, IdKatNad)

IdKat	- ceo broj, identifikuje kategoriju, automatsko dodeljivanje narednog identifikatora
Naziv	- niz do 50 znakova, obavezno
IdKatNad	- strani kljuc (tabela Kategorija)

#### Напомена:

Атрибут IdKatNad представља категорију изнад у хијерархији.

Уколико је вредност атрибута IdKatNad једнака NULL, то значи да је то основна категорија.

#### Proizvod (IdPro, IdAuk, Naziv, IdKat)

IdPro	- ceo broj, identifikuje proizvod, automatsko dodeljivanje narednog identifikatora
Naziv	- niz do 50 znakova, obavezno
IdKat	- strani kljuc (tabela Kategorija), obavezno

#### Korisnik (IdKor, Ime)

IdKor	- ceo broj, identifikuje korisnika, automatsko dodeljivanje narednog identifikatora
Ime	- niz do 50 znakova, obavezno

#### Aukcija (IdAuk, Pocetak, Kraj, Cena, IdPro, IdKor)

IdAuk	- ceo broj, identifikuje aukciju, automatsko dodeljivanje narednog identifikatora
Pocetak	- ceo broj, obavezno, vreme u formatu (ууmmddhhmm)
Kraj	- ceo broj, vreme u formatu (ууmmddhhmm)
Cena	- ceo broj, obavezno, vrednost veћа od 0
IdPro	- strani kljuc (tabela Proizvod), obavezno
IdKor	- strani kljuc (tabela Korisnik), obavezno

#### Напомена:

Атрибут IdKor представља корисника који је иницирао аукцију.

Атрибут Cena представља почетну цену производа.

Уколико је вредност атрибута Kraj једнака NULL, то значи да је време краја аукције непознато и да је аукција активна, уколико је вредност атрибута Kraj дефинисан, аукција је завршена.

Сматрати да је вредност атрибута Kraj увек исправна (мања од тренутног времена).

#### Ponuda (IdPon, Trenutak, Iznos, IdKor, IdAuk)

IdPon	- ceo broj, identifikuje ponudu, automatsko dodeljivanje narednog identifikatora
Trenutak	- ceo broj, vreme u formatu (ууmmddhhmm)
Iznos	- ceo broj, obavezno, vrednost veća od 0
IdKor	- strani kljuc (tabela Korisnik), obavezno
IdAuk	- strani kljuc (tabela Aukcija), obavezno

#### Напомена:

Атрибут IdKor представља корисника који даје понуду.

Гарантује се да ће Trenutak понуде бити исправан и да ће припадати интервалу у коме је аукција активна. Гарантује се да ће износ сваке касније понуде за исту аукцију бити строго већи.

#### Pobedila (IdAuk, IdPon)

IdAuk	- strani kljuc (tabela Aukcija), primarni kljuc, obavezno
IdPon	- strani kljuc (tabela Ponuda), obavezno

Задатак 1 [4 поена]

Потребно је направити SQL упит који исписује статистику везану за аукције. Потребно је приказати колико има аукција излицитираних за цену мању или једнаку од дупло, затим за цену већу дупло и мању или једнаку тродупло, па за цену већу од тродупло и мању или једнаку четвородупло и за цену већу од четвородупло у односу на почетну. Потребно је исписати и статистике за групе у којима нема ни једне аукције. Резултат треба да буде формиран у четири реда – групе. У првом реду треба да буде група “ $\leq 200\%$ ”, у другом реду “ $> 200\% \text{ i } \leq 300\%$ ”, у трећем реду “ $> 300\% \text{ i } \leq 400\%$ ”, а у четвртном реду “ $> 400\%$ ”.

Резултат дати у форми: Naziv grupe, Број аукција  
У Сactus-у користити таб: Zadatak 1

---

Задатак 2 [4 поена]

Потребно је написати SQL упит који исписује све аукције које су успешне, а које су почеле и завршиле се истог дана и које су трајале мање од 6 сати. Потребно је исписати и ко је победник аукције и по којој цени је производ купљен. Атрибут Pobednik представља IdКог корисника који је победио на аукцији. Сортирати по IdAuk опадајуће.

Резултат дати у форми: IdAuk, Pocetak, Крај, Pocetna cena, Крајња цена, Pobednik  
У Сactus-у користити таб: Zadatak 2

---

Задатак 3 [4 поена]

Потребно је направити SQL упит који исписује кориснике које су након свих аукција у којима су учествовали остварили строго више новца него што су имали пре аукција (разлика између укупне цене по којој су производе продавали и укупне цене по којој су производе куповали је позитивна). Активне аукције се не посматрају. Сортирати по IdKor растуће.

Резултат дати у форми: IdKor, Име  
У Сactus-у користити таб: Zadatak 3

---

Задатак 4 [4 поена]

Потребно је направити SQL скрипту која ако постоји табела **Aukcija** избацује табелу **Aukcija** из шеме, а затим формира нову табелу **Aukcija** која треба да има одговарајућу структуру и ограничења. Није потребно да се реализује ограничење да је вредност поља Rosetak увек мања од вредности поља Крај, као ни ограничење да је датум у исправном формату.

У Сactus-у користити таб: Zadatak 4

---

Задатак 5 [5 поена]

Потребно је направити SQL упит који исписује све парове корисника код којих је била узајамна купопродаја на аукцијама (први корисник је купио неки производ који је други корисник ставио на аукцију и други корисник је купио неки производ који је први корисник ставио на аукцију). Потребно је исписати и укупан број аукција у којима су један од другог нешто откупили. Резултат треба сортирати растуће по IdKor1, а затим растуће по IdKor2. IdKor1 је потребно да има мању вредност од IdKor2.

Резултат дати у форми: IdKor1, Ime1, IdKor2, Ime2, БројАукција  
У Сactus-у користити таб: Zadatak 5

---

Задатак 6 [5 поена]

Потребно је направити SQL упит који дохвати све кориснике којима се исплатила продаја претходно купљеног производа. Продаја претходно купљеног производа се исплати уколико му износ по којој је производ продат у процентима није пала за више од броја година колико је производ био код њега. Број година се посматра као цео број и заокружује се на више. Процент за коју је вредност пала се рачуна као однос разлике износа и претходног износа. Производ може да буде максимално једном купљен и максимално једном продат од стране једног истог корисника. Резултат сортирати по IdKor растуће, а затим растуће по IdPro.

Резултат дати у форми: IdKor, Ime, IdPro, Naziv  
У Сactus-у користити таб: Zadatak 6  
**Није дозвољено коришћење погледа.**

---

Задатак 7 [5 поена]

Потребно је направити SQL упит који ажурира стања одговарајућих табела на начин да проглашава све активне аукције завршеним за које постоји бар један корисник који је дао понуду за ту аукцију. И том приликом корисника са највишом понудом проглашава за победника те аукције. Сматрати да је тренутно време 2002012005. Након тога приказати информације о свим аукцијама и њиховим победницима. Потребно је исписати и аукције које и даље немају победника. Атрибут Period се исписује у формату *Росетак - Крај*. Атрибут Pobednik представља IdKог корисника који је победио на аукцији. Резултат сортирати по IdAuk растуће.

Резултат дати у форми: IdAuk, Period, Pobednik

У Сactus-у користити таб: Zadatak 7

**Није дозвољено коришћење погледа.**

---

Задатак 8 [6 поена]

Потребно је направити SQL упит за испис три најбоље рангирана корисника на листи тактичара. Тактизирање се мери на следећи начин: Израчуна се колико је у просеку за сваку завршену аукцију на којој је корисник победио дао више пара у односу на износ претходно предложене понуде за ту аукцију. Ако је корисник био једини понуђач аукције (па самим тим и победник), онда се гледа колико је дао више пара у односу на почетну цену на аукцији. Што је добијена вредност мања, то је корисник боље рангиран на листи тактичара. Ранг представља редни број на листи и потребно је да буде написан речима (“prvi”, “drugi”, “treci”). Обратите пажњу да две особе могу имати исти ранг (деле исти ранг). Резултат треба сортирати по Rang растуће (по редном броју), па по идентификатору корисника растуће.

Резултат дати у форми: Rang, Име, Поени

У Sactus-у користити таб: Zadatak 8

**Није дозвољено коришћење погледа.**

---

Задатак 9 [6 поена]

Потребно је направити SQL упит који исписује колико има производа у свакој од основних категорија, као и колико њих је тренутно на аукцији, колико њих је било на аукцији, просечан износ производа по којој су последњи пут продавани (производи који су тренутно на аукцију и даље немају коначну цену па њих није потребно рачунати, већ цену по којој су продати на претходној аукцији уколико је она постојала). За категорије које немају ни један производ, потребно је исписати -2 за просечну цену. За категорије које немају ни један производ који је купљен на аукцији, потребно је исписати -1 за просечну цену. Категорија је основна уколико нема надкатегорију. Сортирати по *Bilo* на аукцији опадајуће, а затим по *Trenutno* на аукцији растуће и на крају по *IdKat* опадајуће.

Резултат дати у форми: *IdKat*, *Naziv*, *Broj proizvoda*, *Bilo na aukciji*, *Trenutno na aukciji*, *Prosecan iznos*  
У Сactus-у користити таб: *Zadatak 9*

**Није дозвољено коришћење погледа.**

---



Задатак 10 [7 поена]

Потребно је направити SQL упит који дохвата све категорије за које важи следеће:

- Та категорија има бар једну поткатегорију
- Сви производи који спадају у ту категорију налазе у некој од поткатегорија које нису даље дељиве на поткатегорије
- Не постоји поткатегорија те категорије, која није даље дељива на поткатегорије, таква да се у њој не налази ни један производ.

Сортирати по IdKat растуће.

Резултат дати у форми: IdKat, Naziv

У Sactus-у користити таб: Zadatak 10

**Није дозвољено коришћење погледа.**

---