

Испит из Релационих база података, јануар 2025. год. (И смер), I група

Задаци се раде 180 минута. Максималан број поена је 40.

1. (8п) Написати SQL упит којим се издвајају студенти на буџету са студијског програма Математика чије се име или презиме састоји из више речи и који су полагали неки испит у петак. Издвојити следеће податке: индекс, име, презиме, назив предмета, ознаку рока и оцену добијену на испиту који је полагао у петак. Колону за издвајање оцене назвати "Оцена/Порука". Издвојити и студенте који нису полагали ни један испит у петак – у том случају уместо оцене исписати поруку *Није полагао/ла ни један испит петком*. Резултат уредити у растући редослед према броју индекса.
2. (12п) Написати SQL упит којим се издвајају студенти треће године основних академских студија који имају положено више од 90 ЕСПБ поена и једини су на тој години студија из свог места. Издвојити следеће податке: индекс, име, презиме, место рођења у облику <прво слово, последње слово> (нпр. за Ужице - УЕ), назив студијског програма и број положених ЕСПБ.
3. (20п)
 - (a) Написати SQL наредбу којом се прави табела **Studentska_organizacija** која има наредне колоне:
 - **ID** - индекс студента, примарни кључ. Додати страни кључ ка одговарајућој табели.
 - **ime** - име студента, ниска максималне дужине 50 карактера.
 - **prezime** - презиме студента, ниска максималне дужине 50 карактера.
 - **tim** - назив тима којем студент припада унутар организације. Ниска која може имати једну од следећих вредности: *CR, PR, HR* или *IT*.
 - **datum_uclanjenja_u_org** - датум учлањења студента у организацију. Вредност овог атрибута мора бити позната.
 - **datum_uclanjenja_u_tim** - датум учлањења студента у тренутни тим (током чланства у организацији могу се мењати тимови).
 - **upravni_odbor** - ознака да ли је студент члан управног одбора. Подразумевана вредност је *False*.
 - (b) Направити кориснички дефинисану функцију **clan_organizacije** која у зависности од задатог индекса студента:
 - Уколико је студент члан организације, враћа поруку: *Student jeste clan organizacije*.
 - Уколико студент није члан организације, враћа поруку: *Student nije clan organizacije*.
 - (c) Написати SQL наредбу којом се уносе подаци о новим члановима у табелу **Studentska_organizacija**, користећи податке из табеле *dosije*. Унети податке о студентима чије је место рођења Нови Сад. Уколико студирају Информатику, постају чланови ИТ тима, уколико студирају Астрономију и астрофизику, постају део CR тима, уколико студирају Математику-основне студије, тада постају део HR тима, а иначе постају део PR тима. Као датум учлањења у организацију поставити датум уписа на факултет, а за датум учлањења у тим поставити вредност 15.3.2025.
 - (d) Написати SQL наредбу којом се ажурирати вредности колоне *upravni_odbor*. Поставити вредност на *True* за све чланове који су се у оквиру свог тима најраније учланили у организацију.
 - (e) Написати SQL наредбу којом се креира окидач **prekvalifikacija**, помоћу којег се члану организације дозвољава промена тима само уколико је прошло више од 6 месеци од његовог учлањења у тим. Тада вредност колоне *datum_uclanjenja_u_tim* поставити на тренутни датум.
 - (f) Написати SQL наредбу којом се из студентске организације бришу сви чланови који су дипломирали.