

Neka su date relacije koje čuvaju podatke o artefaktima:

Nalaziste(id, naziv, lokacija, površinaHa) - Spisak arheoloških nalazišta.

Artefakt(id, naziv, tipId, vrednost, nalazisteId, istrazivacId, datum) - Spisak pronađenih artefakata. Vrednost je procenjena novčana vrednost.

TipArtefakta(id, naziv) - Spisak svih tipova artefakata.

Istrazivac(id, ime, prezime, specijalnost) - Spisak istraživača.

Istrazivanje(id, nalazisteId, datumPocetka, datumZavrsetka) - Spisak sprovedenih istraživanja. Ukoliko je istraživanje i dalje u toku, *datumZavrsetka* ima vrednost *NULL*. Na jednom nalazištu nije moguće istovremeno sprovođenje više istraživanja.

Uloga(id, naziv) - Spisak svih uloga koje istraživač može imati na istraživanju.

Ucestvovanje(id, istrazivacId, istrazivanjeId, ulogaId) - Istraživači i uloge koje su imali na istraživanjima. Istraživač može imati više uloga na jednom istraživanju.

Napisati SQL upite kojim se prikazuje:

1. **(4 poena)** Kreirati tabelu **VrhunskIstrazivaci**, koja sadrži sledeće kolone: **istrazivacId**, **brojVodjstava**, **ukupnaVrednost**. Da bi se istraživač našao u tabeli, moraju da važe sledeći uslovi:

- Istraživač je bio vođa istraživanja bar 5 puta.
- Ukupna vrednost artefakata koje je pronašao veća je od prosečne vrednosti svih artefakata u bazi.

Kreirati triger na tabeli *Ucestvovanje*, kao i na tabeli *Artefakt* koji automatski ažurira ovu tabelu kada se doda novo učešće, odnosno artefakt. *Pomoć*: možete iskoristiti proceduru iz 4. zadatka.

2. **(4 poena)** Kreirati funkciju koja vraća tabelu sa istraživanjima koja su bila u toku na nekom nalazištu za prosleđene parametre **nalazisteId**, **datumPocetka** i **datumZavrsetka**. Tabela treba da sadrži sledeće kolone: **istrazivanjeId**, **period**, gde *period* predstavlja dužinu trajanja tog istraživanja.
3. **(6 poena)** Kreirati proceduru koja za prosleđenu **godinu** ispisuje koliko je kog meseca bilo nalazišta na kojima se odvijalo istraživanje. Za izradu zadatka je neophodno koristiti kursore. Procedura treba da **ispisuje rezultate** u formatu:
Mesec | BrojNalazišta
4. **(3 poena)** Kreirati proceduru koja za prosleđeni **istrazivacId** vraća broj vodstava, kao i ukupnu vrednost artefakata koje je istraživač pronašao.
5. **(6 poena)** Kreirati triger na tabeli **Istrazivanje** koji proverava da nijedno istraživanje na istom nalazištu ne sme vremenski da se preklapa sa drugim istraživanjem. Ako se pokuša dodavanje ili ažuriranje takvog istraživanja, ispisati poruku "**GREŠKA: Nije moguće sprovođenje više istraživanja na istom nalazištu istovremeno!**". U suprotnom, izvršiti operaciju. *Pomoć*: možete iskoristiti funkciju iz 2. zadatka.

Napomena: Za svaku kreiranu rutinu je potrebno napisati i primer poziva. Zadaci bez primera poziva neće biti bodovani!

Kursori

```
declare @indeks int
declare @upisan int
declare @spred int
declare @ocena int

-- Deklarisi kursor za rezultat zeljenog upita
declare kursorkrozPrijave cursor for
select indeks, upisan, spred, ocena from prijave

open kursorkrozPrijave -- Otvori kursor za citanje

FETCH NEXT FROM kursorkrozPrijave INTO @indeks, @upisan, @spred, @ocena
-- Ucitaj prvi red rezultata u promenljive

WHILE @@FETCH_STATUS = 0
BEGIN
    --select @indeks, @upisan, @spred, @ocena
    print concat(@indeks, ', ', @upisan, ', ', @spred, ', ', @ocena)

    FETCH NEXT FROM kursorkrozPrijave INTO @indeks, @upisan, @spred, @ocena
END

CLOSE kursorkrozPrijave
DEALLOCATE kursorkrozPrijave
```

/* TRIGERI - SINTAKSA

```
CREATE TRIGGER [ schema_name.]trigger_name
ON { table|view }
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS { sql_statement [ ; ] [ ... ] } */
```