

ПРЕПРЕМА - РЕШЕНИ ЗАДАЦИ

Дате су следеће релације базе података SPORTOVI.

- **SPORTISTA** (ID, IME, PREZIME, BRTEL, GOD_ROD, GRAD, PLATA)
- **KLUB** (ID, NAZIV, LOKACIJA, GOD_OSNIVANJA, SPORT)
- **CLAN** (ID_SPORTISTE, ID_KLUBA, DATUM_OD, DATUM_DO, TIM)

Релација **SPORTISTA** садржи информације о спортистима и то: јединствени идентификатор, име, презиме, број телефона, година рођења, град одакле је и плату (уколико зарађује).

У релацији **KLUB** чувају се информације о спортским клубовима: јединствени идентификатор, назив, локација (подразумевана вредност локације је “Beograd”), година оснивања и којим се спортом бави.

Релација **CLAN** поседује информације о томе који спортиста је члан ког спортског клуба, од када је члан тог клуба у једном тиму (ова информација је обавезна), до ког датума је био члан тог клуба у том тиму (уколико је још увек члан тима, ово поље има NULL вредност) и у ком је тиму. Тим може имати једну од следећих вредности: “skolski”, “prvi” или “juniori”. Могуће је да у бази постоје и спортисти који још увек немају клуб.

Задатак 1. [поена]

За описану базу података написати следеће SQL упите:

a) [поена]

Приказати име, презиме и број телефона свих спортиста чије презиме почиње словом ‘М’, ‘Р’ или ‘S’, а чији се број телефона завршава цифрама ‘0’ или ‘1’.

Решење:

```
SELECT IME, PREZIME, BRTEL
FROM SPORTISTA
WHERE PREZIME REGEXP '^[MPS]'
AND BRTEL REGEXP '[01]$';
```

b) [поена]

Приказати назив клуба и спорт којим се бави, али само за клубове чији назив садржи реч од тачно 5 слова (нпр. “Spart”, “Zemun”) или чија локација садржи цифру у називу (нпр. “Blok 45”). Приказати само оне клубове који имају барем једног члана у првом (“prvi”) тиму.

Решење:

```
SELECT DISTINCT k.NAZIV, k.SPORT
FROM KLUB k
JOIN CLAN c ON k.ID = c.ID_KLUBA
WHERE (k.NAZIV REGEXP '^[[:alpha:]]{5}$' OR k.LOKACIJA REGEXP '[0-9]')
AND c.TIM = 'prvi';
```

Задатак 2. [поена]

Креирати улогу (role) под називом `rukovodilac_podataka`. Овој улози доделити привилегије за читање, измену и унос података над табелама **SPORTISTA** и **CLAN**.

Након тога, креирати корисника `operater_sport` са лозинком “Sport2026!”, доделити му новокреирану улогу и поставити је као подразумевану за тог корисника.

Решење:

-- Kreiranje uloge

```
CREATE ROLE 'rukovodilac_podataka';
```

-- Dodela privilegija ulozi

```
GRANT SELECT, INSERT, UPDATE ON SPORTISTA TO 'rukovodilac_podataka'; GRANT SELECT, INSERT, UPDATE ON CLAN TO 'rukovodilac_podataka';
```

-- Kreiranje korisnika

```
CREATE USER 'operater_sport'@'localhost' IDENTIFIED BY 'Sport2026!';
```

-- Povezivanje korisnika i uloge

```
GRANT 'rukovodilac_podataka' TO 'operater_sport'@'localhost';
```

-- Postavljanje uloge kao aktivne

```
SET DEFAULT ROLE 'rukovodilac_podataka' TO 'operater_sport'@'localhost';
```

Задатак 3. [поена]

Написати тригер који, пре промене плате спортисте, проверава да ли је нова плата мања од старе. Уколико јесте, задржава стару вредност плате (не дозвољава смањење).

Решење:

```
DELIMITER //
```

```
CREATE TRIGGER zabrana_smanjenja_plate
```

```
BEFORE UPDATE ON SPORTISTA
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF NEW.PLATA < OLD.PLATA THEN
```

```
        SET NEW.PLATA = OLD.PLATA;
```

```
    END IF;
```

```
END //
```

```
DELIMITER ;
```

Задатак 4. [поена]

Написати ускладиштену процедуру `повецај_platu_sa_izvestajem`.

Процедура треба да прихвати параметар `id_kluba_param`, који представља идентификатор клуба.

Такође, процедура треба да има параметар `ukupni_trosak_bonusa` у који се уписује укупан износ свих повећања плата. [поена]

За све спортисте који су тренутни чланови задатог клуба (они код којих је `DATUM_DO` једнак `NULL`), потребно је повећати плату за 15%. [поена]

У процедури је потребно:

a) [поена]

користити локалну променљиву `procenat` која представља проценат повећања (15%),

b) [поена]

израчунати укупан износ повећања плата и уписати га у параметар `ukupni_trosak_bonusa`,

c) [поена]

након тога ажурирати плате спортиста тог клуба тако да имају плату повећану бонусом,

d) [поена]

користити сесијску променљиву `@log_operacija` која служи за бележење редног броја позива процедуре у оквиру једне сесије (увећава се при сваком позиву процедуре).

Решење:

DELIMITER //

```
CREATE PROCEDURE povecaj_platu_sa_izvestajem(
  IN id_kluba_param INT,
  OUT ukupni_trosak_bonusa DECIMAL(15,2)
)
BEGIN
  -- 1. Lokalna varijabla (traje samo dok traje procedura)
  DECLARE procenat_povecanja DECIMAL(5,2) DEFAULT 15.00;

  -- Inicijalizujemo OUT parametar na 0
  SET ukupni_trosak_bonusa = 0;

  -- 2. Računamo koliko će nas ukupno koštati повиšica pre samog UPDATE-a
  -- Koristimo SUM da izračunamo (plata * 0.15) za sve koji ispunjavaju uslov
  SELECT SUM(PLATA * (procenat_povecanja / 100)) INTO ukupni_trosak_bonusa
  FROM SPORTISTA s
  JOIN CLAN c ON s.ID = c.ID_SPORTISTE
  WHERE c.ID_KLUBA = id_kluba_param
  AND c.DATUM_DO IS NULL; -- Samo za trenutne članove

  -- 3. Izvršavamo stvarni Update
  UPDATE SPORTISTA s
  JOIN CLAN c ON s.ID = c.ID_SPORTISTE
  SET s.PLATA = s.PLATA + (s.PLATA * (procenat_povecanja / 100))
  WHERE c.ID_KLUBA = id_kluba_param
  AND c.DATUM_DO IS NULL;

  -- 4. Korišćenje sesijske varijable za brojanje poziva procedure
  IF @log_operacija IS NULL THEN
    SET @log_operacija = 1;
  ELSE
    SET @log_operacija = @log_operacija + 1;
  END IF;
END //

DELIMITER ;
```

ПРЕДМЕТНИ ПРОФЕСОР И ПРЕДМЕТНИ АСИСТЕНТ