

# UVOD U BAZE PODATAKA

Predavanje 13-14

# Školska zadaća 4 - komentar

	Broj rješenja	Broj studenata	Prosječno bodova	Broj studenata s maksimalnim bodovima	Broj studenata s 0 bodova
I1	155 60%	257	1	13 8%	20 13%
I2	139 54%	257	1.26	50 36%	25 18%
I3	141 55%	257	1.26	25 18%	13 9%
I4	101 39%	257	1.1	30 30%	27 27%

# PODUPITI

# Uvod

- **Podupit** (engl. *subquery*) je upit koji se nalazi unutar nekog drugog upita
- Podupit je uvijek SELECT izraz
- Podupit se najčešće koristi:
  - U SELECT dijelu
  - U WHERE ili HAVING dijelu
- Podupit se **uvijek** stavlja u okrugle zagrade
- Podupit se često može zamijeniti **spajanjem**
  - Spajanje se uvijek može zamijeniti podupitom
  - Korištenje podupita ili spajanja? Prema želji i iskustvu...

# Primjer podupita

```
SELECT
    Naziv,
    Boja,
    CijenaBezPDV
FROM Proizvod
WHERE CijenaBezPDV >
    (SELECT AVG(CijenaBezPDV)
     FROM Proizvod)
ORDER BY
    CijenaBezPDV DESC
```

Vanjski upit  
(1. dio)

Podupit  
(unutar zagrada)

Vanjski upit  
(2. dio)

# Podjela podupita

- Podjela prema **načinu rada**:
  1. **Konstantni** podupiti (ne ovise o retku vanjskog upita)
  2. **Korelirani** podupiti (ovise o retku vanjskog upita)
- Podjela prema **mjestu pojavljivanja**:
  1. Podupiti u **SELECT** dijelu
  2. Podupiti u **WHERE/HAVING** dijelu
  3. Podupiti u **FROM** dijelu
- Podjela prema **povratnoj vrijednosti**:
  1. **Skalarni** podupiti (vraćaju **jednu** vrijednost)
  2. **Tabularni** podupiti (vraćaju **tablicu**)

# Konstantni i korelirani podupiti

- Podupit je koreliran ako koristi stupce vanjskog upita
  - Ako ih ne koristi, onda je konstantan
- Konstantni podupit se izvršava jedanput
  - Konstantan je za sve retke vanjskog upita
- **Korelirani podupit** se izvršava jedanput za svaki redak vanjskog upita
  - Što ako rezultat ima 500.000 redaka? Oprezno s koreliranim podupitima!
  - Vanjski upit ne može koristiti stupce podupita, već samo njegov rezultat
- Kao kod spajanja, preporuka je korištenje aliasa za tablice kako bi se izbjegle zabune ili nejasnoće

# Podupiti u SELECT dijelu

- Podupit u SELECT dijelu **mora vratiti jednu vrijednost**
  - Podupit u SELECT dijelu ne može vratiti tablicu
- Podupit može vratiti:
  - Istu vrijednost za sve retke (**konstantni**), ili
  - Za svaki redak drukčiju vrijednost (**korelirani**)
- Prema vraćenoj vrijednosti se odnosimo kao prema bilo kojem drugom stupcu
  - Možemo mu dati alias
  - Možemo ga koristiti za sortiranje u ORDER BY dijelu



# Primjeri podupita u SELECT dijelu

- **Konstantni:**

```
SELECT Naziv,  
    (  
        SELECT COUNT(*)  
        FROM Proizvod  
    ) AS BrojProizvoda  
FROM Proizvod
```

- **Korelirani:**

```
SELECT TOP 5 Naziv,  
    (  
        SELECT SUM(s.Kolicina)  
        FROM Stavka AS s  
        WHERE s.ProizvodID =p.IDProizvod  
    ) AS Prodano  
FROM Proizvod AS p  
ORDER BY Prodano DESC
```

# Podupiti u WHERE/HAVING dijelu

- Podupit u WHERE/HAVING dijelu **može vratiti jednu ili više vrijednosti**, ovisno o operatoru s kojim se koristi:
  - Operatori uspoređivanja =, <, <=, >, >=, !=
    - Podupit mora vratiti **jednu** vrijednost
  - Operatori IN i NOT IN
    - Podupit mora vratiti **tablicu s jednim stupcem**
  - Operatori EXISTS i NOT EXISTS
    - Podupit može vratiti **bilo što** (za EXISTS: ako podupit vrati **0 redaka**, rezultat je **false**, inače je **true**; za NOT EXISTS je obratno)
- Podupit može biti konstantni ili korelirani
- Vraćena vrijednost se koristi za filtriranje redaka (WHERE) ili grupa (HAVING)

# Primjeri podupita u WHERE/HAVING dijelu (1/2)

- **Konstatni:**

```
SELECT
    Naziv
FROM Proizvod
WHERE CijenaBezPDV >
(
    SELECT AVG(CijenaBezPDV)
    FROM Proizvod
)
```

- **Korelirani s operatorom uspoređivanja:**

```
SELECT *
FROM Kupac AS k
WHERE
(
    SELECT COUNT(*)
    FROM Racun AS r
    WHERE r.KupacID =k.IDKupac
) >= 27
```

# Primjeri podupita u WHERE/HAVING dijelu (2/2)

- **Konstantni s operatorom IN:**

```
SELECT *
FROM Proizvod AS p
WHERE IDProizvod IN
(
    SELECT DISTINCT
    s.ProizvodID
    FROM Stavka AS s
)
```

- **Korelirani s operatorom EXISTS:**

```
SELECT *
FROM Proizvod AS p
WHERE EXISTS
(
    SELECT *
    FROM Stavka AS s
    WHERE s.ProizvodID =p.IDProizvod
)
```

# Primjeri

1. Ispišite sve potkategorije i za svaku ispišite broj proizvoda u njoj.
2. Riješite prethodni zadatak pomoću spajanja.
3. Ispišite sve kategorije i za svaku ispišite broj proizvoda u njoj.
4. Ispišite sve proizvode i pokraj svakog ispišite zarađeni iznos, od najboljih prema lošijim.
5. Dohvatite sve proizvode, uz svaki proizvod ispišite prosječnu cijenu svih proizvoda te razliku prosječne cijene svih proizvoda i cijene tog proizvoda. U obzir uzmite samo proizvode s cijenom većom od nule.

# Primjeri

6. Dohvatite imena i prezimena 5 komercijalista koji su izdali najviše računa.
7. Dohvatite imena i prezimena 5 najboljih komercijalista po broju realiziranih računa te uz svakog dohvatite i iznos prodane robe.
8. Dohvatite sve boje proizvoda. Uz svaku boju **pomoću podupita** dohvatite broj proizvoda u toj boji.
9. Dohvatite imena i prezimena svih kupaca iz Frankfurta i uz svakog ispišite broj računa koje je platio karticom, od višeg prema nižem.

# Primjeri

10. Vratite sve proizvode čija je cijena pet ili više puta veća od prosjeka.

11. Vratite sve proizvode koji su prodavani, ali u količini manjoj od 5.

12. Vratite sve proizvode koji nikad nisu prodani:

- Pomoću IN ili NOT IN
- Pomoću EXISTS ili NOT EXISTS
- Pomoću spajanja

13. Vratite količinu zarađenog novca za svaku boju proizvoda.

# Primjeri

14. Vratite količinu zaradenog novca za svaku boju proizvoda, ali samo za one boje koje su zaradile više od 20.000.000.
15. Vratiti sve proizvode koji imaju dodijeljenu potkategoriju i koji su prodani u količini većoj od 5000. Uz svaki proizvod vratiti prodanu količinu i naziv kategorije.



# Primjeri (Loš film)

16. Ispišite popis mjesta u koja su posuđivani filmovi te uz svako mjesto ispišite i broj posuđenih filmova.

17. Ispišite popis članova te uz svakoga koliko je posudbi napravio.

18. Ispišite popis medija te uz svaki medij koliko filmova je snimljeno na takav medij.

19. Ispišite popis žanrova te uz svaki žanr koliko je filmova tog žanra posuđeno.

# Primjeri (Loš film)

20. Ispišite popis žanrova te uz svaki žanr koliko članova je posudilo film tog žanra.
21. Ispišite popis članova koji su posudili više filmova od člana s oznakom 5.
22. Ispišite popis članova koji su posudili više filmova od prosjeka posudbi svih članova.
23. (AdventureworksOBP) Ispišite za svakog komercijalistu koliko je napravio svojih računa, koliki je općenito prosjek kreiranih računa po komercijalisti i neka budu prikazani samo oni koji imaju broj kreiranih računa veći od prosjeka.

# Podupiti u FROM dijelu

- Podupit u FROM dijelu vraća privremenu tablicu
- Obavezno je korištenje aliasa za privremenu tablicu dobivenu podupitom
- Vanjski upit potom koristi rezultat podupita
- Podupit ne može biti korelirani

# Primjeri (LosFilm)

24. Ispišite popis članova koji su posudili više filmova od prosjeka posudbi svih članova.

```
SELECT Clan.ImePrezime, COUNT(Posudba.FilmID) 'Broj filmova'
FROM Clan INNER JOIN Posudba ON Clan.ID = Posudba.ClanID
GROUP BY Clan.ImePrezime HAVING COUNT(*) >
(
    SELECT AVG(temp.broj_filmova_po_clanu) FROM
    (
        SELECT COUNT(Posudba.FilmID) AS broj_filmova_po_clanu
        FROM Clan INNER JOIN Posudba ON Clan.ID = Posudba.ClanID
        GROUP BY Clan.ImePrezime
    ) AS temp
)
```

# Primjeri (AdventureWorksOBP)

25. Ispišite popis kupaca koji imaju više računa od prosjeka svih kupaca.

```
select k.Ime, k.Prezime, COUNT(*) as BrojRacuna
from Kupac as k inner join Racun as r on r.KupacID=k.IDKupac
group by k.Ime, k.Prezime having COUNT(*) >
(
    select AVG(t.BrojRacunaPoClanu)
    from (
        select COUNT(*) as BrojRacunaPoClanu
        from Kupac as k inner join Racun as r on r.KupacID=k.IDKupac
        group by k.Ime, k.Prezime ) as t
    )
```