

SQL

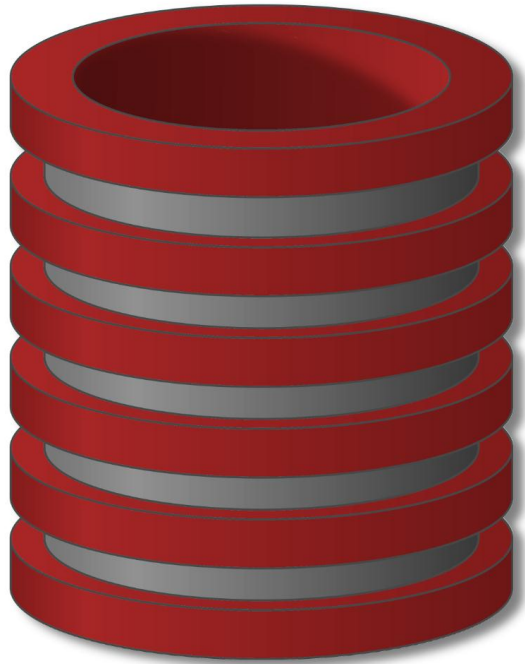
Podupiti

Mogućnosti podupita

- Podupit je upit unutar upita
- Ima zadatak da **razloži kompleksnu logiku**
- Mozemo da prikažemo podatke koje nam **where klauzula ne bi dozvolila**, to se pre svega odnosi na funkcije agregacije koje se primenjuju nad svim vrstama a koje nisu dozvoljene u where klauzuli.
- Uz pomoć podupita, **funkcije agregacije mozemo da sakrijemo u where klauzuli**

Osobine podupita

- Podupiti se mogu navesti u **SELECT**, **FROM** i **WHERE** klauzuli.
- Rezultat podupita može biti **konkretna vrednost** (upit vraća samo jednu vrednost)
 - Podupit je povezan sa glavnim upitom jednim od relacionih operatora (**=**, **>**, **<**, **!=**, **<=**, **>=**)
- Podupit može da vrati skup vrednosti i tada se koristi operator **IN** ili uz relacione operatore doda ključna reč **ANY** ili **ALL** (npr. **=ALL** ili **>ALL**)
- U podupitu se može koristiti operator **EXISTS**.
 - To je unarni logički operator
 - Prikazaće vrednost ako pod upit izdvaja bar jedan slog i često se može zameniti operatorom **IN**




SQL

Podupiti u WHERE klauzuli

```
select A1, A2, ..., An  
From   R1, R2, ..., Rm  
where  uslov
```

Iskaz uključuje “podupit“:
ugnježdeni u **select** iskaz



UGNJEŽDANI UPITI U WHERE ISKAZU

ODELJENJE <brod#, imeod, mesto, sefod\$>

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

UCESCE <idbr#, brproj#, brsati, funkcija>

PROJEKAT <brproj#, imeproj, sredstva, rok>

1. Prikazati ime i posao svih radnika koji rade odeljenju Palilula

```
SELECT ime, posao  
FROM RADNIK  
WHERE brod =
```

Podupit vraća samo jednu vrednost, broj odeljenja za odeljenje Palilula

```
(SELECT brod  
FROM ODELJENJE  
WHERE odeljnje = 'Palilula')
```

2. Prikazati ime i posao svih radnika koji rade u Nisu

```
SELECT ime, posao  
FROM RADNIK  
WHERE brod IN
```

Podupit vraća više vrednosti, brojeve odeljenja za odeljenja koja su u Nišu

```
(SELECT brod  
FROM ODELJENJE  
WHERE mesto = 'Nis')
```

UGNJEŽDANI UPITI U WHERE ISKAZU

ODELJENJE <brod#, imeod, mesto, sefod\$>

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

UCESCE <idbr#, brproj#, brsati, funkcija>

PROJEKAT <brproj#, imeproj, sredstva, rok>

3. Prikazati ime, posao i kvalifikaciju svih radnika koji imaju istu kvalifikaciju kao Mitar

```
SELECT ime, posao, kvalif
FROM RADNIK
WHERE kvalif IN
      (SELECT kvalif
       FROM RADNIK
       WHERE ime = 'Mitar')
```

UGNJEŽDANI UPITI U WHERE ISKAZU

ODELJENJE <brod#, imeod, mesto, sefod\$>

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap, premija, plata, brod\$>

UCESCE <idbr#, brproj#, brsati, funkcija>

PROJEKAT <brproj#, imeproj, sredstva, rok>

4. Prikazati ime, datum zaposlenja i posao zaposlenih koji imaju kvalifikaciju kao Marko i zaposleni su 2015.

```
SELECT ime, datzap, posao
FROM RADNIK
WHERE kvalif IN (SELECT kvalif
                 FROM RADNIK
                 WHERE ime = 'Marko')
AND datzap BETWEEN '01/01/2015' AND '31/12/2015'
```

YEAR(datzap)=2015

UGNJEŽDENE UPITI U WHERE ISKAZU

ODELJENJE <brod#, imeod, mesto, sefod\$>

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

UCESCE <idbr#, brproj#, brsati, funkcija>

PROJEKAT <brproj#, imeproj, sredstva, rok>

5. Prikazati ime, posao i ukupna primanja radnika koji rade u Nišu ne uzimajući u obzir upravnike i prodavce. Rezultate sortirati po ukupnim primanjima u opadajućem redosledu

```
SELECT ime, posao, plata + IFNULL(premija,0) as 'Ukupna primanja'  
FROM RADNIK  
WHERE brod IN (SELECT brod  
                FROM ODELJENJE  
                WHERE mesto = 'Nis')  
AND posao NOT IN ('upravnik' , 'prodavac')  
ORDER BY 3 DESC
```

UGNJEŽDANI UPITI U WHERE ISKAZU

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

UCESCE <idbr#, brproj#, brsati, funkcija>

6. Prikazati ime I kvalifikaciju svih radnika koji rade na istim projektima kao i Marko

```
SELECT ime, kvalif
FROM RADNIK
WHERE idbr IN
      (SELECT idbr
       FROM UCESCE
       WHERE brproj IN
            (SELECT brojproj
             FROM UCESCE
             WHERE idbr IN
                  (SELECT idbr
                   FROM RADNIK
                   WHERE ime='Marko'))));
```

UGNJEŽDENE UPITI U WHERE ISKAZU

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

7. Prikazati idbr, ime, platu i kvalifikaciju zaposlenih koji imaju istu platu kao bilo koji zaposleni čija je kvalifikacija VSS

```
SELECT idbr, ime, platu, kvalif
```

```
FROM RADNIK
```

```
WHERE plata = ANY
```

```
    (SELECT plata
```

```
      FROM RADNIK
```

```
      WHERE kvalif = 'VSS')
```

```
AND kvalif <> 'VSS';
```

- Kada se koriste u podupitu, **IN** operator i = **ANY** imaju potpuno istu ulogu
- ANY ključna reč uvek ide uz operator poređenja (=, >, <, <>, >=, <=, !=)
- Vraća TRUE ukoliko je poređenje tačno za bilo koju vrednost koju podupit vraća
- NOT IN nije jednako sa <> **ANY**
- NOT IN je jednako sa <> **ALL**.

Umesto ključne reči ANY može da se koristi i SOME koja je za razumevanje lakša od ANY

UGNJEŽDENE UPITI U WHERE ISKAZU

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

8. Prikazati idbr, ime, platu i kvalifikaciju zaposlenih koji imaju platu manju od svih zaposlenih čija je kvalifikacija VSS

```
SELECT idbr, ime, platu, kvalif
FROM RADNIK
WHERE plata < ALL
                (SELECT plata
                 FROM RADNIK
                 WHERE kvalif = 'VSS');
```

- **ALL** ključna reč uvek ide uz operator poređenja (=, >, <, <>, >=, <=, !=)
- Vraća **TRUE** ukoliko je poređenje tačno za sve vrednosti koje podupit vraća
- **NOT IN** je jednako sa <> **ALL**.
- Rezultat je FALSE (unknown) ukoliko podupit sadrži NULL vrednost

UGNJEŽDENI UPITI U WHERE ISKAZU

ODELJENJE <brod#, imeod, mesto, sefod\$>

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

9. Prikazati sve podatke o odeljenjima u kojima ima zaposlenih radnika

10. Prikazati sve podatke o odeljenjima u kojima nema zaposlenih radnika

```
SELECT *  
FROM ODELJENJE  
WHERE EXISTS  
    (SELECT *  
     FROM RADNIK  
     WHERE RADNIK.brod=ODELJENJE.brod);
```

```
SELECT *  
FROM ODELJENJE  
WHERE NOT EXISTS  
    (SELECT *  
     FROM RADNIK  
     WHERE RADNIK.brod=ODELJENJE.brod);
```

UGNJEŽDANI UPITI U WHERE ISKAZU

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

UCESCE <idbr#, brproj#, brsati, funkcija>

11. Prikazati sve podatke o zaposlenima koji rade na nekom projektu

```
SELECT *  
FROM RADNIK  
WHERE EXISTS (SELECT *  
               FROM UCESCE  
               WHERE RADNIK.idbr=UCESCE.idbr);
```

UGNJEŽDENI UPITI + GRUPISANJE

ODELJENJE <brod#, imeod, mesto, sefod\$>

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap. premija, plata, brod\$>

12. Prikazati imena odeljenja u kojima su ukupna primanja svih radnika u odeljenju veća od 10000

```
SELECT imeod
FROM ODELJENJE
WHERE brod IN
      (SELECT brod
       FROM RADNIK
       GROUP BY brod
       HAVING SUM( plata + IFNULL(premija,0)) > 10000)
```

UGNJEŽDANI UPITI + GRUPISANJE + AGREGATNE FUNKCIJE

13. Prikazati ime i primanja svih zaposlenih čija su primanja veća od prosečnog primanja u preduzeću.

```
SELECT ime, plata + IFNULL(premija,0) as Ukupna Primanja
FROM RADNIK
WHERE plata + IFNULL(premija,0) >
                                (SELECT AVG(plata + IFNULL(premija,0))
                                FROM RADNIK);
```


UGNJEŽDANI UPITI + GRUPISANJE + AGREGATNE FUNKCIJE

ODELJENJE <brod#, imeod, mesto, sefod\$>

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap, premija, plata, brod\$>

14. Prikazati imena svih radnika čija su ukupna primanja manja od prosečnih primanja u odeljenju čije je sedište na Paliluli

```
SELECT ime
FROM RADNIK
WHERE plata + IFNULL(premija,0) < (SELECT AVG(plata + IFNULL(premija,0))
FROM RADNIK
WHERE brod= (SELECT brod
FROM ODELJENJE
WHERE imeod='Palilula')));
```

UGNJEŽDENE UPITI + GRUPISANJE + AGREGATNE FUNKCIJE

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap, premija, plata, brod\$>

UCESCE <idbr#, brproj#, brsati, funkcija>

15. Prikazati ime, datum zaposlenja i primanja zaposlenih koji su angažovani na dva projekta.

```
SELECT ime, datzap, plata + IFNULL(premija,0)
FROM RADNIK
WHERE idbr IN
                (SELECT idbr
                FROM UCESCE
                GROUP BY idbr
                HAVING count(*) = 2);
```

UGNJEŽDENE UPITI + GRUPISANJE + AGREGATNE FUNKCIJE

16. Ko su najbolje plaćeni radnici u celom preduzeću?

```
SELECT idbr,ime,plata  
FROM RADNIK  
WHERE plata = (SELECT MAX(plata)  
                FROM RADNIK);
```

- Spoljašnji i unutrašnji upit mogu biti povezani po vrednostima više atributa.
- U tom slučaju ako se upoređuju argumenti oba argumenta moraju imati jednaki broj atributa

17. Ko su najbolje plaćeni radnici u svakom odeljenju?

```
SELECT idbr,ime,plata,brod  
FROM RADNIK  
WHERE plata IN (SELECT MAX(plata)  
                FROM RADNIK  
                GROUP BY brod);
```

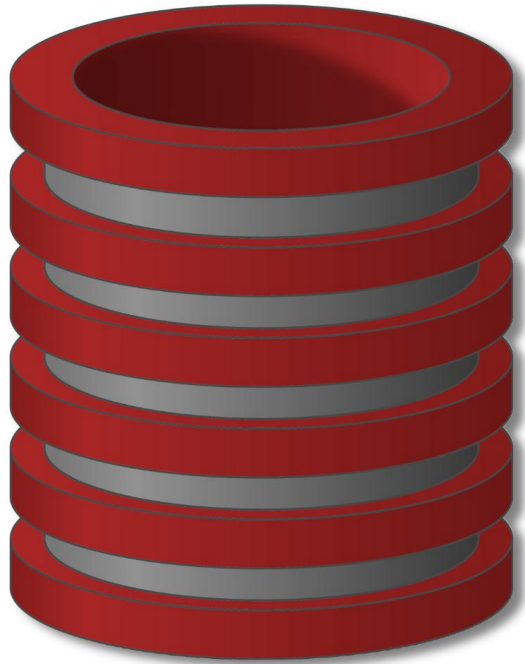
```
SELECT idbr,ime,plata,brod  
FROM RADNIK  
WHERE (brod, plata) IN (SELECT brod, MAX(plata)  
                       FROM RADNIK  
                       GROUP BY brod);
```

UGNJEŽDENE UPITI + GRUPISANJE + AGREGATNE FUNKCIJE

RADNIK <idbr#, ime, prezime, posao, kvalif, rukovodilac\$, datzap, premija, plata, brod\$>

18. Prikazati imena radnika koji su se poslednji zaposlili u svakom odeljenju?

```
SELECT ime,datzap, brod
FROM RADNIK
WHERE datzap IN
                (SELECT MAX(datzap)
                FROM RADNIK
                GROUP BY brod);
```



SQL

Podupiti u

FROM i

SELECT Iskazima

SQL: Podupiti u FROM i SELECT

```
select A1, A2, ..., An
from R1, R2, ..., Rm
where uslov
```

Iskaz uključuje “podupiti“:
ugnježdeni u **select** iskaz

Primer: Prijava studenata baza podataka

Fakultet(Fime, Grad, BrojMesta)

Student(Sid, Sime, Prosek, Vskole)

Prijava(Sid, Fime, Smer, Odluka)

Student

Sid	Sime	Prosek	Vskole

Prijava

Sid	Fime	Smer	Odluka

Fakultet

Fime	grad	BrojMesata

SQL: Podupiti u FROM iskazu

Zadatak 1:

Prikazati sve podatke iz tabele student koji dolaze iz većih škola

```
select *
From (select Sid,Sime,Prosek,Prosek*(Vskole/1000,0) as 'SkaliranProsek'
      From Student) G
where G.SkaliranProsek-Prosek>1
```

- Podupit koji smo kreirali u from klauzuli ponasa se kao nova relacija.
- Bilo koji atribut tog podupita mozemo da pozovemo u glavnom upitu.
- U ovom slucaju mi smo rezultat podupita iskoristili u where klauzuli

Student

Sid	Sime	Prosek	Vskole
1	Marko	3.8	580
2	Darko	4.3	400
3	Jelena	4.9	620
4	Darko	3.6	300

Prijava

Sid	Fime	Smer	Odluka
1	VTS	SRT	Primljen
1	ETF	RTI	Odbijen
2	ELFAK	Grafika	Primljen
2	FON	SRT	Odbijen

Fakultet

Fime	grad	BrojMesata

SQL: Podupiti u WHERE iskazu

Zadatak 2:

Prikazati ime fakulteta, grad i prosek najboljeg studenata koji se prijavio za dati fakultet

```
Select Fakultet.Fime,Grad,Prosek
From Student,Prijava,Fakultet
Where Student.Sid=Prijava.Sid and Prijava.Fime=Fakultet.Fime and
       Prosek >= all (Select Prosek
                     From Student,Prijava
                     Where Student.Sid=Prijava.Sid
                     and Prijava.Fime=Fakultet.Fime)
```

Student

Sid	Sime	Prosek	Vskole
1	Marko	3.8	580
2	Darko	4.3	400
3	Jelena	4.9	620
4	Darko	3.6	300

Prijava

Sid	Fime	Smer	Odluka
1	VTS	S	Primljen
1	ETF	RTI	Odbijen
2	ELFAK	Grafika	Primljen
2	FON	SRT	Odbijen

Fakultet

Fime	grad	BrojMesata

SQL: Podupiti u SELECT iskazu

Zadatak 2:

Prikazati ime fakulteta, grad i prosek najboljeg studenata koji se prijavio za dati fakultet

```
select Fakultet.Fime,Grad, (select Prosek
                           from Student,Prijava
                           where Student.Sid=Prijava.Sid and Prijava.Fime=Fakultet.Fime
                              and Prosek >= all (select Prosek
                                                from Student,Prijava
                                                where Student.Sid=Prijava.Sid and
                                                Prijava.Fime=Fakultet.Fime) as Prosek
from Fakultet
```

Student

Sid	Sime	Prosek	Vskole
1	Marko	3.8	580
2	Darko	4.3	400
3	Jelena	4.9	620
4	Darko	3.6	300

Prijava

Sid	Fime	Smer	Odluka
1	VTS	S	Primljen
1	ETF	RTI	Odbijen
2	ELFAK	Grafika	Primljen
2	FON	SRT	Odbijen

Fakultet

Fime	grad	BrojMesata