

PODUPITI

V auditorne vježbe

Teme

- ▶ SQL naredbe koje sadrže uslove s podupitom
 - ▶ Usporedba izraza iz vanjskog upita sa rezultatom podupita
 - ▶ Ispitivanje je li neki izraz iz vanjskog upita sadržan u rezultatu podupita
 - ▶ Ispitivanje da li se kao rezultat podupita pojavljuje barem jedna n-torka
- ▶ Korelirani podupiti
- ▶ Kombiniranje uslova s podupitima
- ▶ Određivanje presjeka i razlike relacija pomoću SELECT naredbe



Podupiti (*Subqueries*)

sale

oznSala	nazSala	kapacitet
A101	Mali amfiteatar 101	64
Stelekt	Zgrada Stelekta	60
RC15	Računarska sala 15	16
A008	Veliki amfiteatar 008	120
102	Učionica 102	35

rezultat:

oznSala	nazSala	kapacitet
A101	Mali amfiteatar 101	64
Stelekt	Zgrada Stelekta	60
A008	Veliki amfiteatar 008	120

- ▶ Ispisati sale čiji je kapacitet veći od prosječnog kapaciteta svih sala

- ▶ Pogrešna rješenja:

nije dozvoljeno

SELECT * FROM sala WHERE kapacitet > AVG(kapacitet)

- ▶ Obaviti upit kojim se određuje prosječni kapacitet svih sala

SELECT AVG(kapacitet) FROM sala

- ▶ Zapamtiti dobijeni rezultat (59) te napisati novi upit

SELECT * FROM sala WHERE kapacitet > 59



Podupiti (*Subqueries*)

► Ispravno rješenje:

- Upit kojim se određuje prosječni kapacitet svih sala uvrstiti u uslov dohvata

```
SELECT * FROM sala
```

```
WHERE kapacitet > (SELECT AVG(kapacitet) FROM sala)
```

59

- Rezultat ovog podupita je jednak za svaku n-torku iz relacije sala sa kojom se poredi kapacitet te je rezultat podupita dovoljno izračunati samo jednom tokom obavljanja upita

sale

oznSala	nazSala	kapacitet
A101	Mali amfiteatar 101	64
Stelekt	Zgrada Stelekta	60
RC15	Računarska sala 15	16
A008	Veliki amfiteatar 008	120
102	Učionica 102	35

64>(SELECT AVG ...) → true

60>(SELECT AVG ...) → true

16>(SELECT AVG ...) → false

120>(SELECT AVG ...) → true

35>(SELECT AVG ...) → false

rezultat:

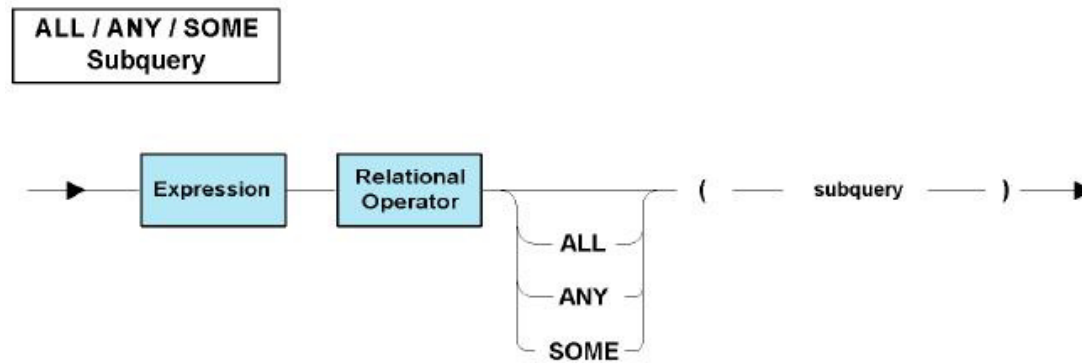
oznSala	nazSala	kapacitet
A101	Mali amfiteatar 101	64
Stelekt	Zgrada Stelekta	60
A008	Veliki amfiteatar 008	120

SQL naredbe koje sadrže uslove s podupitom

- ▶ Osim uslova usporedbe, uslov dohvata se može graditi i iz uslova s podupitima
- ▶ SELECT naredba ugniježdena unutar *WHERE clause* neke druge SELECT, UPDATE ili DELETE naredbe naziva se podupit
- ▶ Svaki podupit mora se nalaziti unutar okruglih zagrada
- ▶ Podupit se može ugraditi u vanjski upit na tri osnovna načina (načini ugradnje podupita):
 - ▶ Usporedba izraza iz vanjskog upita sa rezultatom podupita
 - ▶ Ispitivanje je li neki izraz iz vanjskog upita sadržan u rezultatu podupita
 - ▶ Ispitivanje da li se kao rezultat podupita pojavljuje barem jedna n-torka



Usporedba izraza iz vanjskog upita sa rezultatom podupita



- ▶ *Relational Operator* može biti bilo koji od relacionih operatora ($>$, $<$, $=$, itd.)
- ▶ Rezultat SELECT naredbe koja predstavlja podupit, može biti jedna n-torka, više n-torki ili niti jedna n-torka.
- ▶ Da bi n-torka vanjskog upita zadovoljila uslov selekcije napisan na ovaj način, mora biti zadovoljen uslov usporedbe izraza *Expression* s jednom ili više n-torki koje se dobiju kao rezultat podupita.



Usporedba izraza iz vanjskog upita sa rezultatom podupita

- ▶ Ključna riječ **ALL** – **sve n-torke** dobivene podupitom moraju zadovoljiti uslov usporedbe s *Expression*, da bi se izraz evaluirao kao istinit. Ako podupit ne vrati niti jednu vrijednost, izraz se uvijek evaluira kao istinit (TRUE).
- ▶ Ključna riječ **ANY** (**SOME** je sinonim) – biće dovoljno da uslov usporedbe bude zadovoljen za **barem jednu n-torku** iz podupita. Ako podupit ne vrati niti jednu vrijednost, izraz se uvijek evaluira kao lažan (FALSE).
- ▶ Podupit upotrijebljen na ovaj način mora vraćati tačno jednu kolonu
- ▶ U suprotnom, upit završava u pogrešci.
- ▶ **SELECT** naredba koja se koristi za podupit ne smije sadržavati **ORDER BY clause**.



Usporedba izraza iz vanjskog upita sa rezultatom podupita

- ▶ **Primjer:** Odrediti predmete na kojima je upisan tako mali broj studenata da se nastava može odjednom (bez podjele po dvoranama) održati u bilo kojoj dvorani

```
SELECT * FROM pred
      WHERE upisanoStud <= ALL (SELECT kapacitet FROM dvorana)
```

- ▶ Ovaj se upit (konceptualno) obavlja na slijedeći način:
 - ▶ Prvo se evaluiira podupit kojim se formira skup čiji su elementi kapaciteti dvorana.
 - ▶ Uzima se prva n-torka iz relacije **pred** pa se vrijednost atributa **upisanoStud** uporedi sa svakom vrijednošću iz skupa kapaciteta.
 - ▶ Ukoliko je vrijednost **upisanoStud** manja ili jednaka od svake vrijednosti s kojom se usporedila, prva n-torka iz relacije **pred** će se pojaviti u rezultatu.
 - ▶ Postupak se ponavlja za svaku n-torku relacije **pred**.



Usporedba izraza iz vanjskog upita sa rezultatom podupita

- ▶ *Expression* **relOp** ALL (*subquery*)
 - ▶ Ako se podupitom dobije skup vrijednosti $\{x_1, x_2, \dots, x_n\}$ efektivno se uslov evaluira na sljedeći način:
 - ▶ *Expression* **relOp** x_1 AND *Expression* **relOp** x_2 AND ... AND *Expression* **relOp** x_n
 - ▶ true ako je *Expression* **relOp** x_i true za svako i
- ▶ *Expression* **relOp** { ANY | SOME } (*subquery*)
 - ▶ Ako se podupitom dobije skup vrijednosti $\{x_1, x_2, \dots, x_n\}$ efektivno se uslov evaluira na sljedeći način:
 - ▶ *Expression* **relOp** x_1 OR *Expression* **relOp** x_2 OR ... OR *Expression* **relOp** x_n
 - ▶ true ako je *Expression* **relOp** x_i true za barem jedno i
- ▶ gdje je **relOp** jedan od { < | <= | = | <> | > | >= }



Usporedba izraza iz vanjskog upita sa rezultatom podupita

- ▶ **Primjer:** Odrediti predmete za koje postoji barem jedna dvorana sa dovoljnim kapacitetom da se u njoj nastava održi odjednom za sve studente (bez dijeljenja studenata u više dvorana):

```
SELECT * FROM pred
WHERE upisanoStud <= ANY (SELECT kapacitet FROM
dvorana)
```

- ▶ **Primjer:** Odrediti predmete za koje se nastava ne može odjednom za sve upisane studente održati niti u jednoj dvorani:

```
SELECT * FROM pred
WHERE upisanoStud > ALL (SELECT kapacitet FROM
dvorana)
```



Usporedba izraza iz vanjskog upita sa rezultatom podupita

- ▶ Ako podupit pouzdano vraća samo jednu n-torku mogu se ključne riječi **ALL**, **ANY** i **SOME** ispustiti
- ▶ Ukoliko bi se pri tako postavljenom podupitu dogodilo da podupit vrati više od jedne n-torke, upit bi završio u pogrešci.
- ▶ Primjer: Ispisati sve dvorane u kojima je broj mjesta veći od prosječnog broja mjesta:

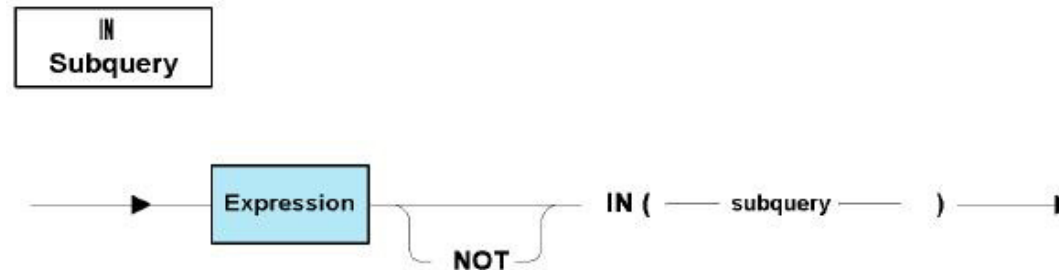
```
SELECT * FROM dvorana  
WHERE kapacitet >= (SELECT AVG(kapacitet) FROM dvorana)
```

- ▶ Primjer: Ispis dvorane (ili dvorana, jer ih može biti nekoliko sa jednakim brojem mjesta) koje imaju najveći broj mjesta:

```
SELECT * FROM dvorana  
WHERE kapacitet = (SELECT MAX(kapacitet) FROM dvorana)
```



Ispitivanje je li neki izraz iz vanjskog upita sadržan u rezultatu podupita



- ▶ Da bi n-torka vanjskog upita zadovoljila uslov (tj. pojavila se u rezultatu):
 - ▶ ukoliko se NE koristi modifikator NOT – *Expression* mora biti sadržan u skupu dobivenom obavljanjem podupita
 - ▶ ukoliko se koristi modifikator NOT – *Expression* ne smije biti sadržan u skupu dobivenom obavljanjem podupita
- ▶ Podupit upotrijebljen na ovaj način mora vraćati tačno jednu kolonu. U suprotnom, upit završava u pogrešci.
- ▶ SELECT naredba koja se koristi za podupit ne smije imati *ORDER BY clause*.



Ispitivanje je li neki izraz iz vanjskog upita sadržan u rezultatu podupita

- ▶ **Primjer:** Ispis svih podataka o dvoranama u kojima se ne održava nastava

```
SELECT * FROM dvorana
      WHERE oznDvorana NOT IN
      (SELECT DISTINCT oznDvorana FROM
rezervacija)
```

- ▶ **Primjer:** Ispis svih nastavnika koji su u toku prošle godine dali barem jednu negativnu ocjenu

```
SELECT * FROM nastavnik
      WHERE sifNastavnik IN
      (SELECT DISTINCT sifNastavnik FROM ispit
      WHERE YEAR(datIspit)=YEAR(CURRENT_DATE)-1
      AND ocjena = 1)
```

(može li se ovaj upit riješiti bez upotrebe podupita?)



Ispitivanje da li se kao rezultat podupita pojavljuje barem jedna n-torka



- ▶ Rezultat podupita je jedna, više ili niti jedna n-torka.
- ▶ Broj atributa u listi za selekciju ovog podupita nije ograničen niti je bitan za rezultat.
- ▶ Da bi n-torka vanjskog upita zadovoljila uslov (i pojavila se u rezultatu):
 - ▶ ukoliko se NE koristi modifikator NOT – podupit mora rezultirati s bar jednom n-torkom
 - ▶ ukoliko se koristi modifikator NOT – podupit ne smije vratiti niti jednu n-torku
- ▶ **Primjer:** Ispis svih dvorana ukoliko DANAS nije obavljen niti jedan ispit

```
SELECT * FROM dvorana
WHERE NOT EXISTS
      (SELECT * FROM ispit WHERE datIspit = TODAY)
```



Korelirani podupiti

- ▶ Podupit koji pri evaluaciji koristi vrijednosti vanjskog upita naziva se korelirani podupit
- ▶ **Primjer:** Ispisati sve predmete za koje je sedmično opterećenje predmeta veće od broja različitih dana u sedmici u kojima se odvija nastava iz tog predmeta

```
SELECT pred.* FROM pred
    WHERE pred.brojSatiTjedno >
        (SELECT COUNT(DISTINCT oznVrstaDan) FROM
            rezervacija
            WHERE rezervacija.sifPred = pred.sifPred)
```

- ▶ Ovaj upit je koreliran jer se za njegovu evaluaciju koristi vrijednost atributa pred.sifPred koja je rezultat vanjskog upita
-



Korelirani podupiti

- ▶ Upit se (konceptualno) obavlja na slijedeći način:
 - ▶ Uzima se prva n-torka iz relacije **pred** pa se koristeći vrijednost **pred.sifPred** izračunava broj različitih dana u sedmici za koje je obavljena rezervacija za taj predmet.
 - ▶ Ukoliko je broj različitih dana u sedmici iz vanjskog upita veći od dobivenog broja, n-torka iz vanjskog upita se pojavljuje u rezultatu.
 - ▶ Postupak se ponavlja za svaku n-torku relacije **pred**.
- ▶ Svaki **nekorelirani** podupit se obavlja samo jednom, a zatim se njegov rezultat koristi za usporedbu s n-torkama dobivenim vanjskim upitom.
- ▶ **Korelirani** podupiti se moraju obaviti jednom za svaku n-torku iz vanjskog upita.



Korelirani podupiti

- ▶ I korelirani podupiti se mogu ugrađivati u upite na sva tri opisana načina
- ▶ **Primjer:** Ispisati sve predmete za koje ne postoji niti jedna rezervacija dvorane.

```
SELECT pred.sifPred, pred.nazPred FROM pred
WHERE NOT EXISTS
    (SELECT * FROM rezervacija
     WHERE rezervacija.sifPred = pred.sifPred)
```

- ▶ **Primjer:** Ispisati podatke o dvoranama za koje ne postoji neka druga dvorana s jednakim kapacitetom.

```
SELECT oznDvorana, kapacitet FROM dvorana
WHERE kapacitet NOT IN
    (SELECT kapacitet FROM dvorana drugaDvorana
     WHERE drugaDvorana.oznDvorana <>
     dvorana.oznDvorana)
```



Kombiniranje uslova s podupitima

- ▶ Budući da podupit (SELECT naredba koja se koristi kao podupit) može opet biti bilo koja SELECT naredba (osim što ne smije sadržavati ORDER BY *clause*), SQL naredba može sadržavati podupit koji je ugniježđen podupit, u kojem je opet ugniježđen podupit itd.
- ▶ **Primjer:** Ispisati podatke svih studenata koji nisu položili niti jedan ispit sa barem prosječnom ocjenom (prosjek se računa kao prosjek pozitivnih ocjena svih ispita)

```
SELECT * FROM stud
WHERE mbrStud NOT IN
  (SELECT DISTINCT mbrStud FROM ispit
   WHERE ocjena >=
     (SELECT AVG(ocjena) FROM ispit
      WHERE ocjena > 1))
```



Kombiniranje uslova s podupitima

- ▶ Uslov s podupitom je vrsta uslova koji se može povezivati s drugim uslovima pomoću logičkih operatora
- ▶ **Primjer:** Ispisati mjesta čiji se poštanski brojevi ne koriste u relaciji stud

```
SELECT * FROM mjesto
    WHERE pbr NOT IN (SELECT pbrStan FROM stud)
    AND pbr NOT IN (SELECT pbrRod FROM stud)
```

- ▶ **Drugo moguće rješenje je:**

```
SELECT * FROM mjesto
    WHERE NOT EXISTS
        (SELECT * FROM stud
            WHERE pbrRod = mjesto.pbr
            OR pbrStan = mjesto.pbr)
```



Kombiniranje uslova s podupitima

- Konačno, podupiti se mogu kombinovati i s uslovima usporedbe, npr.

```
SELECT ... FROM ...  
  WHERE izraz1 >= ALL  
    (SELECT ... FROM ...  
      WHERE ...  
        AND izraz2 NOT IN  
          (SELECT ... FROM ...  
            WHERE ...  
          )  
    )  
OR EXISTS  
  (SELECT ... FROM ...  
    WHERE ...  
    AND izraz3 < izraz4  
  )  
AND izraz5 > izraz6
```



Podupite ne treba koristiti bez razloga

- ▶ Podupite ne treba primjenjivati za rješavanje problema u kojima primjena podupita nije potrebna
- ▶ **Primjer:** Iskazati sve podatke o ispitima svih studenata koji stanuju u Koprivnici

```
SELECT * FROM ispit
  WHERE mbrStud IN
    (SELECT mbrStud FROM stud
      WHERE pbrStan IN
        (SELECT pbr FROM mjesto
          WHERE nazMjesto = 'Koprivnica'))
```

- ▶ Efikasnije i jasnije upit se rješava na slijedeći način:

```
SELECT ispit.* FROM ispit INNER JOIN stud
  ON ispit.mbrStud = stud.mbrStud
  INNER JOIN mjesto ON stud.pbrStan = mjesto.pbr
  WHERE nazMjesto = 'Koprivnica'
```



Određivanje presjeka relacija pomoću SELECT naredbe

MjestoRod

oznDrz	pbr	nazMjesto
HR	10000	Zagreb
HR	51000	Rijeka
HR	31000	Osijek
GB	10000	London
GB	47000	Glasgow

MjestoStan

oznDrz	pbr	nazMjesto
HR	10000	Zagreb
HR	47000	Karlovac
HR	31000	Osijek
GB	10000	London

- ▶ Ključ u obje relacije je (oznDrz, pbr)

```
SELECT * FROM mjestoRod
WHERE EXISTS
    (SELECT mjestoStan.pbr FROM mjestoStan
     WHERE mjestoStan.oznDrz = mjestoRod.oznDrz
      AND mjestoStan.pbr = mjestoRod.pbr)
```



Određivanje presjeka relacija pomoću SELECT naredbe

- ▶ Operacija presjeka se može riješiti i pomoću spajanja relacija

```
SELECT mjestoRod.*
```

```
FROM mjestoRod INNER JOIN mjestoStan
```

```
    ON mjestoStan.oznDrz = mjestoRod.oznDrz
```

```
    AND mjestoStan.pbr = mjestoRod.pbr
```

- ▶ U općem slučaju se operacija presjeka ne može riješiti pomoću podupita IN

```
SELECT * FROM mjestoRod
```

```
    WHERE mjestoRod.oznDrz IN
```

```
        (SELECT mjestoStan.oznDrz FROM mjestoStan)
```

```
    AND mjestoRod.pbr IN
```

```
        (SELECT mjestoStan.pbr FROM mjestoStan)
```



Određivanje razlike relacija pomoću SELECT naredbe

```
SELECT * FROM mjestoRod
WHERE NOT EXISTS
    (SELECT mjestoStan.pbr FROM mjestoStan
     WHERE mjestoStan.oznDrz = mjestoRod.oznDrz
     AND mjestoStan.pbr = mjestoRod.pbr)
```

- ▶ U općem slučaju se operacija razlike ne može riješiti pomoću podupita IN

```
SELECT * FROM mjestoRod
WHERE mjestoRod.oznDrz NOT IN
    (SELECT mjestoStan.oznDrz FROM mjestoStan)
AND mjestoRod.pbr NOT IN
    (SELECT mjestoStan.pbr FROM mjestoStan)
```



Primjeri

- ▶ Ispisati sve podatke o studentima koji nisu dobili niti jednu pozitivnu ocjenu na ispitu iz predmeta sa organizacione jedinice 'Zavod za primijenjeno računarstvo'

```
SELECT * FROM stud
WHERE mbrStud NOT IN
  (SELECT mbrStud FROM ispit INNER JOIN pred
    ON ispit.sifPred = pred.sifPred
    INNER JOIN orgjed
    ON pred.sifOrgjed = orgjed.sifOrgjed
    WHERE ocjena > 1
    AND nazOrgjed='Zavod za primijenjeno računarstvo')
```



Primjeri

- ▶ Ispisati kraticu i naziv predmeta za sve predmete na kojima je upisano više studenata nego što je položeno ispita u prvom kvartalu tekuće godine iz svih predmeta.

```
SELECT kratPred, nazPred FROM pred
WHERE upisanoStud > (SELECT COUNT(*) FROM ispit
                     WHERE YEAR(datIspit) = YEAR(CURRENT_DATE)
                     AND MONTH(datIspit) IN (1, 2, 3)
                     AND ocjena>1)
```



Primjeri

- ▶ Ispisati kraticu i naziv predmeta na kojima je upisano više studenata nego što je položeno ispita na svim ostalim predmetim koji se predaju na istoj organizacionoj jedinici.

```
SELECT kratPred, nazPred FROM pred
WHERE upisanoStud >
  (SELECT COUNT(*) FROM ispit INNER JOIN pred ostali
   ON ispit.sifPred = ostali.sifPred
   WHERE ocjena > 1
   AND ostali.sifOrgjed = pred.sifOrgjed
   AND ostali.sifPred <> pred.sifpred);
```



Primjeri

- ▶ Ispisati ime i prezime studenata koji su položili ispit iz predmeta 'Baze podataka' prije svih studenata koji su položili ispit iz predmeta 'Operativni sistemi'

```
SELECT imeStud, prezStud FROM stud INNER JOIN ispit
ON stud.mbrStud = ispit.mbrStud
INNER JOIN pred ON ispit.sifPred = pred.sifPred
WHERE ocjena > 1 AND nazPred = 'Baze podataka'
AND datIspit < ALL (SELECT datIspit FROM ispit
    INNER JOIN pred ON ispit.sifPred = pred.sifPred
    WHERE nazPred = 'Operativni sistemi' AND ocjena >1)
```

