

## Unošenje podataka

### Unos podataka

```
INSERT INTO FAKULTET(IME_FAK, SJEDISTE) VALUES  
( 'PMF', 'PODGORICA')
```

```
INSERT INTO FAKULTET(IME_FAK, SJEDISTE) VALUES  
( 'ETF', NULL)
```

```
INSERT INTO FAKULTET(IME_FAK) VALUES ('MTF')
```

## Brisanje podataka

### Brisanje podataka

```
DELETE FROM FAKULTET  
WHERE IME_FAK ='PMF'
```

### Brisanje svih torki (OPASNO!)

```
DELETE FROM STUDENT
```

## Izmjena podataka (SQL)

### Izmjena podataka

```
UPDATE FAKULTET  
SET FAKULTET.IME_FAK = 'FIL'  
WHERE FAKULTET.IME_FAK='PMF'
```

```
UPDATE STUDENT  
SET IME_FAK = 'FIL',  
GODINA=1  
WHERE IME_FAK='PMF' AND  
BROJ_INDEKSA='34'
```

## Indeks

- Indeks je struktura podataka koja služi za efikasniji pristup podacima i sortiranje.
- Nedostatak je što usporava dodavanje i brisanje zapisa.
- Indeks može biti na jednoj ili više kolona.
- Indeks je najčešće neka varijanta B-stabla, recimo B+-stablo.

## UNIQUE indeks

- **Unique** indeks– ne dozvoljava duplikate
- Kod kreiranja primarnog ključa i unique constraint-a kreira se unique indeks.
- Podrazumijeva se da nije UNIQUE.

## Kreiranje i brisanje indeksa (SQL)

```
• Kreiranje i brisanje indeksa (SQL)  
CREATE UNIQUE INDEX indsjed  
ON FAKULTET (SJEDISTE ASC)
```

```
CREATE INDEX indgod  
ON STUDENT (GODINA ASC)
```

```
DROP INDEX STUDENT.indgod
```

## Klasterisani i neklasterisani indeks

- CLUSTERED indeks – kod koga redosled zapisa u listu odgovara fizičkom poretku.
- NONCLUSTERED indeks – kod koga redosled zapisa u listu ne mora odgovarati fizičkom poretku.
- Može biti samo jedan CLUSTERED indeks na tabeli!
- Indeks može biti
  - CLUSTERED - NONCLUSTERED
  - UNIQUE CLUSTERED - UNIQUE CLUSTERED

## Indeks

**CREATE UNIQUE CLUSTERED INDEX indstud ON STUDENT (JMBG ASC)**

**CREATE NONCLUSTERED INDEX indstud ON STUDENT (BRK, GRAD)**

## FILLFACTOR, PAD\_INDEX

- FILLFACTOR – procenat od 1 do 100 popunjenosti lista.
- Nije dobar ni preveliki i ni premali!
- PAD\_INDEX – ako je uključena ova opcija, FILLFACTOR se primjenjuje i na unutrašnje čvorove.

**CREATE CLUSTERED INDEX indstud ON STUDENT (JMBG ASC) WITH PAD\_INDEX, FILLFACTOR = 60**

## IGNORE\_DUP\_KEY

- IGNORE\_DUP\_KEY opcija dozvoljava da se, kada INSERT rečenicom unosimo torke koje imaju duplikate na indeksnim kolona, duplikati budu eliminisani. Ako ova opcija nije postavljena kompletna INSERT rečenica se odbija.

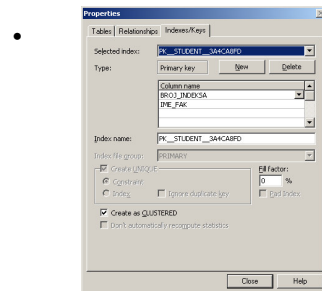
**CREATE INDEX indstud ON STUDENT (JMBG ASC) WITH IGNORE\_DUP\_KEY**

## STATISTICS\_NORECOMPUTE

- Da se ne računa automatski statistika vezana za indeks.
- Ima smisla samo ako se tabela ne mijenja, može loše uticati na optimizator upita.

**CREATE INDEX indstud ON STUDENT (JMBG ASC) WITH STATISTICS\_NORECOMPUTE**

## Indeksi, interaktivno



## Primjer baze

```
CREATE TABLE SNABDJEVAC(
  S_NO VARCHAR(20),
  S_IME VARCHAR(20),
  STATUS INTEGER,
  GRAD VARCHAR(20),
  PRIMARY KEY (S_NO)
)

CREATE TABLE FABRIKA (
  F_NO VARCHAR(20),
  F_IME VARCHAR(20),
  GRAD VARCHAR(20),
  PRIMARY KEY (F_NO)
)

CREATE TABLE ISPORUKA (
  S_NO VARCHAR(20),
  P_NO VARCHAR(20),
  F_NO VARCHAR(20),
  KOL SINGLE,
  PRIMARY KEY (S_NO, P_NO, F_NO),
  FOREIGN KEY (S_NO) REFERENCES SNABDJEVAC,
  FOREIGN KEY (P_NO) REFERENCES PROIZVOD,
  FOREIGN KEY (F_NO) REFERENCES FABRIKA
)

CREATE TABLE PROIZVOD (
  P_NO VARCHAR(20),
  P_IME VARCHAR(20),
  BOJA VARCHAR(20),
  TEZINA DECIMAL(10,4),
  GRAD VARCHAR(20),
  PRIMARY KEY (P_NO)
)
```

### SNABDJEVAČ

S_NO	S_IME	STATUS	GRAD
S1	Stanko	20	Podgorica
S2	Ušvan	10	Beograd
S3	Bozo	30	Beograd
S4	Dejan	20	Podgorica
S5	Igor	30	Budva

### ISPORUKA

S_NO	P_NO	F_NO	KOL
S1	P1	F4	700
S2	P3	F1	400
S2	P3	F2	200
S2	P3	F3	200
S2	P3	F4	500
S2	P3	F5	600
S2	P3	F6	400
S2	P3	F7	800
S2	P5	F2	100
S3	P3	F1	200
S3	P4	F2	500
S4	P6	F3	300
S4	P6	F7	300
S5	P2	F2	200
S5	P2	F4	100
S5	P5	F5	500
S5	P5	F7	100
S5	P6	F2	200
S5	P1	F4	100
S5	P3	F4	200
S5	P4	F4	800
S5	P5	F4	400
S5	P6	F4	500

### PROIZVOD

P_NO	P_IME	BOJA	TEZINA	GRAD
P1	Saraf	Crvena	12	Podgorica
P2	Orponik	Zelena	17	Beograd
P3	Kondenzator	Plava	17	Pjevlja
P4	Kondenzator	Crvena	14	Podgorica
P5	Kabl	Plava	12	Beograd
P6	Utenica	Crvena	19	Podgorica

### FABRIKA

F_NO	F_IME	GRAD
F1	Video	Beograd
F2	Televizor	Pjevlja
F3	Muzički stub	Budva
F4	DVD Plejer	Budva
F5	Satelitska antena	Podgorica
F6	Kasetofon	Kragujvac
F7	Komputer	Podgorica

## Operacija SELECT

- Naći svu informaciju o snabdjevačima  
**SELECT S\_NO, S\_IME, STATUS, GRAD  
FROM SNABDJEVAC**
- Drugi način  
**SELECT \* FROM SNABDJEVAC**
- Naći šifre i imena svih snabdjevača (*projekcija*)  
**SELECT S\_NO, S\_IME  
FROM SNABDJEVAC**

## WHERE dio

- Naći imena proizvoda iz Podgorice.  
**SELECT P\_IME  
FROM PROIZVOD  
WHERE GRAD= 'Podgorica'**
- Naći snabdjevače iz Podgorice sa statusom većim od 10 .  
**SELECT \* FROM SNABDJEVAC  
WHERE GRAD= 'Podgorica' AND STATUS>20**

## And , Or

- Naći fabrike koje su u Beogradu ili Podgorici.  
**SELECT \* FROM FABRIKA  
WHERE GRAD= 'Podgorica' OR GRAD= 'Beograd'**
- Naći sve proizvode iz Beograda, kao i one iz Podgorice koji su crvene boje.  
**SELECT \* FROM PROIZVOD  
WHERE GRAD= 'Beograd' OR  
(GRAD= 'Podgorica' AND BOJA='Crvena')**

## Not

- Naći snabdjevače koji nisu iz Podgorice  
**SELECT \* FROM SNABDJEVAC  
WHERE NOT (GRAD='Podgorica')**
- ili  
**SELECT \* FROM SNABDJEVAC  
WHERE GRAD<>'Podgorica'**
- NULL uvijek daje vrijednost NETAČNO!!!

## Is Null, Is not null

- Naći snabdjevače kojima ne znamo grad.  
**SELECT \* FROM SNABDJEVAC  
WHERE GRAD IS NULL**
- Naći snabdjevače kojima *znamo* grad.  
**SELECT \* FROM SNABDJEVAC  
WHERE GRAD IS NOT NULL**

## Between

- Naći šifre i imena snabdjevača čiji je status između 10 i 20.

```
SELECT * FROM SNABDJEVAC  
WHERE STATUS >= 10 AND STATUS <= 20
```

ili

```
SELECT * FROM SNABDJEVAC  
WHERE STATUS BETWEEN 10 AND 20
```

## Not Between

- Naći šifre i imena snabdjevača čiji je status ili manji od 10 ili veći od 20.

```
SELECT * FROM SNABDJEVAC  
WHERE STATUS < 10 OR STATUS > 20
```

ili

```
SELECT * FROM SNABDJEVAC  
WHERE STATUS NOT BETWEEN 10 AND 20
```

## In

- Naći šifre i imena snabdjevača čiji je status 10 ili 20 ili 30.

```
SELECT * FROM SNABDJEVAC  
WHERE STATUS = 10 OR STATUS = 20 OR  
STATUS = 30
```

ili

```
SELECT * FROM SNABDJEVAC  
WHERE STATUS IN (10,20,30)
```

## Not In

- Naći šifre i imena snabdjevača čiji status nije ni 10 ni 20 ni 30.

```
SELECT * FROM SNABDJEVAC  
WHERE STATUS <> 10 AND STATUS <> 20 AND  
STATUS <> 30
```

ili

```
SELECT * FROM SNABDJEVAC  
WHERE NOT (STATUS = 10 OR STATUS = 20 OR  
STATUS = 30)
```

ili

```
SELECT * FROM SNABDJEVAC  
WHERE STATUS NOT IN (10,20,30)
```

## Like

- Naći imena proizvoda koji počinju slovom P.

```
SELECT P_IME FROM PROIZVOD  
WHERE P_IME LIKE 'P%'
```

	<b>Primjer</b>	<b>Tačno</b>	<b>Netačno</b>
% nula ili više znakova	a%a %ab%	aa, aBa, aBBBa	aBC
_ Jedan znak	a_a	abc, AABb, Xab	aZb, bac
Opseg ili skup znakova	[a-z] [fpj]	abcdefg, abc aaa, a3a, aBa	cab, aab aBBBa
Van opsega ili skupa	[^a-z]	f, p, j f, p, j 9, &, %	2, & 2, & b, a

## Izrazi i ugrađene funkcije

```
SELECT F_NO, KOL*1.3  
FROM ISPORUKA
```

```
SELECT F_NO, KOL*1.3 AS NOVO  
FROM ISPORUKA
```

```
SELECT SIN(STATUS), LEFT(GRAD,3)  
FROM SNABDJEVAC
```

## Order by

```
SELECT * FROM ISPORUKA  
ORDER BY S_NO
```

```
SELECT * FROM ISPORUKA  
ORDER BY S_NO, F_NO
```

```
SELECT * FROM ISPORUKA  
ORDER BY S_NO ASC, F_NO DESC
```

```
SELECT * FROM ISPORUKA  
ORDER BY S_NO DESC, F_NO DESC
```

## Agregatne funkcije

- Pravu snagu pokazuju sa *grupisanjem*.
- 5 osnovnih agregatnih funkcija:
  - MAX
  - MIN
  - AVG
  - SUM
  - COUNT

## Agregatne funkcije

- **SELECT MAX(KOL) FROM ISPORUKA**
- **SELECT MIN(KOL) FROM ISPORUKA**
- **SELECT SUM (KOL) FROM ISPORUKA**
- **SELECT AVG(KOL) FROM ISPORUKA**
- **SELECT COUNT(\*) FROM ISPORUKA**

## Grupisanje

Položio

Br_ind	Predmet	Profesor	Ocjena
100	Mat	P1	6
130	Mat	P2	7
100	Fiz	P2	10
121	Hem	P1	8
150	Fiz	P2	9

## Grupisano po predmetu

100	Mat	P1	6
130	Mat	P2	7

100	Fiz	P2	10
150	Fiz	P2	9

121	Hem	P1	8
-----	-----	----	---

## Grupisano po profesoru

100	Mat	P1	6
121	Hem	P1	8

100	Fiz	P2	10
150	Fiz	P2	9
130	Mat	P2	7

## Grupisano po predmetu i profesoru

100	Mat	P1	6
-----	-----	----	---

121	Hem	P1	8
-----	-----	----	---

100	Fiz	P2	10
150	Fiz	P2	9

130	Mat	P2	7
-----	-----	----	---

## GROUP BY

Naći ukupnu i prosječnu isporučenu količinu po fabrikama.

```
SELECT F_NO, SUM(KOL), AVG(KOG)
FROM ISPORUKA
GROUP BY F_NO
```

- ❖ U rezultat ulazi po jedan red iz svake grupe
- ❖ Zato u SELECT dijelu mogu učestvovati samo atributi iz GROUP BY dijela i agregatne funkcije

## GROUP BY

Naći maksimalnu isporučenu količinu po snabdjevačima i fabrikama.

```
SELECT S_NO, F_NO, MAX(KOL)
FROM ISPORUKA
GROUP BY S_NO, F_NO
```

## HAVING

- Dodatni kriterijum na grupu

Naći maksimalnu isporučenu količinu po snabdjevačima i fabrikama koji imaju više od dvije isporuke.

```
SELECT S_NO, F_NO, MAX(KOL)
FROM ISPORUKA
GROUP BY S_NO, F_NO
HAVING COUNT(*)>2
```

- ❖ U HAVING dijelu se ispituju grupe i zato u tom dijelu mogu učestvovati samo atributi iz GROUP BY dijela i agregatne funkcije

## Upiti sa više tabela

- *Dekartov proizvod*

```
SELECT *
FROM SNABDJEVAC, ISPORUKA
```

- Dekartov proizvod sa where uslovom je **SPAJANJE**.

- Naći imena snabdjevača koji imaju bar jednu isporuku

```
SELECT S_IME
FROM SNABDJEVAC, ISPORUKA
WHERE SNABDJEVAC.S_NO=ISPORUKA.S_NO
```

## Upiti sa više tabela

- Naći imena snabdjevača iz Podgorice koji imaju bar jednu isporuku.

```
SELECT S_IME
FROM SNABDJEVAC, ISPORUKA
WHERE SNABDJEVAC.S_NO=ISPORUKA.S_NO
AND SNABDJEVAC.GRAD='Podgorica'
```

## Upiti sa više tabela

- Naći imena snabdjevača iz Podgorice koji isporučuju bar jedan crveni proizvod.

```
SELECT S_IME
FROM SNABDJEVAC, ISPORUKA, PROIZVOD
WHERE SNABDJEVAC.S_NO=ISPORUKA.S_NO
AND PROIZVOD.P_NO=ISPORUKA.P_NO
AND SNABDJEVAC.GRAD='Podgorica'
AND PROIZVOD.BOJA='Crvena'
```

## Upiti nad više tabela

- Naći imena snabdjevača iz Podgorice koji isporučuju bar jedan crveni proizvod fabrici iz Beograda.

```
SELECT S_IME
FROM SNABDJEVAC, ISPORUKA, PROIZVOD, FABRIKA
WHERE SNABDJEVAC.S_NO=ISPORUKA.S_NO
AND PROIZVOD.P_NO=ISPORUKA.P_NO
AND FABRIKA.F_NO=ISPORUKA.F_NO
AND SNABDJEVAC.GRAD='Podgorica'
AND PROIZVOD.BOJA='Crvena'
AND FABRIKA.GRAD='Beograd'
```

## INNER JOIN

- Spajanje se može izvesti i samo u FROM dijelu

```
SELECT S_IME
FROM SNABDJEVAC INNER JOIN ISPORUKA
ON SNABDJEVAC.S_NO=ISPORUKA.S_NO

SELECT * FROM
(SNABDJEVAC INNER JOIN ISPORUKA ON
SNABDJEVAC.S_NO=ISPORUKA.S_NO)
INNER JOIN FABRIKA ON
ISPORUKA.F_NO=FABRIKA.F_NO
```

## INNER JOIN

- Naći imena snabdjevača iz Podgorice koji isporučuju bar jedan crveni proizvod fabrici iz Beograda.

```
SELECT S_IME FROM
( (SNABDJEVAC INNER JOIN ISPORUKA ON
SNABDJEVAC.S_NO=ISPORUKA.S_NO)
INNER JOIN FABRIKA ON ISPORUKA.F_NO=FABRIKA.F_NO)
INNER JOIN PROIZVOD ON ISPORUKA.P_NO=
PROIZVOD.P_NO
WHERE SNABDJEVAC.GRAD='Podgorica'
AND PROIZVOD.BOJA='Crvena'
AND FABRIKA.GRAD='Beograd'
```

## Spajanje tabele sa sobom

- Naći sve moguće parove (P\_IME, P\_IME)

```
SELECT A.P_IME, B.P_IME
FROM PROIZVOD A, PROIZVOD B
```

## Upiti nad više tabela sa grupisanjem

- Naći šifre snabdjevača kao i broj isporuka crvenih proizvoda.

```
SELECT S_NO, COUNT(*)
FROM ISPORUKA, PROIZVOD
WHERE PROIZVOD.P_NO=ISPORUKA.P_NO
      AND PROIZVOD.BOJA='Crvena'
GROUP BY S_NO
```

## OUTER JOIN

- Uključuje u rezultat i one koji nisu spojeni, sa lijeve, desne strane ili obje strane.

```
SELECT * FROM
SNABDJEVAC INNER JOIN PROIZVOD
ON SNABDJEVAC.GRAD= PROIZVOD.GRAD
```

```
SELECT * FROM
SNABDJEVAC LEFT OUTER JOIN PROIZVOD
ON SNABDJEVAC.GRAD= PROIZVOD.GRAD
```

## OUTER JOIN

```
SELECT * FROM
SNABDJEVAC RIGHT OUTER JOIN PROIZVOD
ON SNABDJEVAC.GRAD= PROIZVOD.GRAD
```

```
SELECT * FROM
SNABDJEVAC FULL OUTER JOIN PROIZVOD
ON SNABDJEVAC.GRAD= PROIZVOD.GRAD
```

## Podupiti

- To su SELECT upiti unutar drugog upita
- Postoje dvije vrste podupita
  - Nevezani podupiti (oni koji se računaju nezavisno od spoljašnjeg upita)
  - Vezani podupiti

## Podupiti kao konstante

Podupit koji vraća jednu vrijednost se može koristiti svuda gdje i konstanta!  
Ispisati podatke o snabdjevačima uz informaciju o ukupnom broju isporuka

```
SELECT *,
      (SELECT COUNT(*) FROM ISPORUKA) AS BR_ISPORUKA
FROM SNABDJEVAC
```

Ispisati podatke o snabdjevačima uz informaciju o ukupnom broju **njihovih** isporuka

```
SELECT *,
      (SELECT COUNT(*) FROM ISPORUKA
       WHERE ISPORUKA.S_NO=SNABDJEVAC.S_NO) AS
      BR_ISPORUKA
FROM SNABDJEVAC
```

## Podupiti kao konstante

Naći one proizvode čijih je isporuka više nego 1/5 ukupnog broja isporuka

```
SELECT ISPORUKA.P_NO, COUNT(*)
FROM ISPORUKA
GROUP BY P_NO
HAVING COUNT(*) >
      (SELECT COUNT(*) FROM ISPORUKA)/5
```

- ovako mogu da se koriste podupiti koji vraćaju tačno jednu vrijednost!!!



## Podupiti koji vraćaju više vrijednosti

- Uvede se pomoću riječi
  - ALL
  - ANY (SOME)
  - IN
  - NOT IN
  - EXISTS
  - NOT EXISTS
- Svi osim EXISTS i NOT EXISTS rade isključivo sa podupitima koji vraćaju jednu kolonu

## Podupit ALL

- Naći šifre onih snabdjevača koji imaju takvu isporuku koja ima količinu veću od svake količine koju isporučuje 'S1'.

```
SELECT ISPORUKA.S_NO
FROM ISPORUKA
WHERE ISPORUKA.KOL >
      ALL (SELECT X.KOL
           FROM ISPORUKA X
           WHERE X.S_NO='S1')
```

## Podupit ALL

- Naći šifre onih snabdjevača koji isporučuju bar jedan detalj koji ne isporučuje 'S1'.

```
SELECT ISPORUKA.S_NO
FROM ISPORUKA
WHERE ISPORUKA.P_NO <>
      ALL (SELECT X.P_NO
           FROM ISPORUKA X
           WHERE X.S_NO='S1')
```

- Ako podupit ne vraća vrijednost onda se upoređenje smatra TAČNIM (probajte sa 'S7')!

## Podupit ANY

- Naći šifre onih snabdjevača koji isporučuju bar jedan detalj u količini većoj od bar jedne količine koju isporučuje 'S1'.

```
SELECT ISPORUKA.S_NO
FROM ISPORUKA
WHERE ISPORUKA.KOL >
      ANY (SELECT X.KOL
           FROM ISPORUKA X
           WHERE X.S_NO='S1')
```

- Ako podupit ne vraća vrijednost onda se upoređenje smatra NETAČNIM (probajte sa 'S7')!
- SOME je isto što i ANY

## Podupit IN

- Naći šifre onih snabdjevača koji snabdjevaju neku fabriku iz Beograda.

```
SELECT ISPORUKA.S_NO
FROM ISPORUKA
WHERE ISPORUKA.F_NO
      IN (SELECT F_NO
          FROM FABRIKA
          WHERE GRAD='Beograd')
```

## Podupit NOT IN

- Naći šifre onih snabdjevača koji isporučuju bar jedan proizvod koji nije crven.

```
SELECT ISPORUKA.S_NO
FROM ISPORUKA
WHERE ISPORUKA.P_NO
      NOT IN (SELECT P_NO
              FROM PROIZVOD
              WHERE BOJA='Crvena')
```

## Podupit EXISTS

- Naći one snabdjevače koji isporučuju bar nešto.

```
SELECT *
FROM SNABDJEVAC
WHERE EXISTS
  (SELECT *
   FROM ISPORUKA
   WHERE SNABDJEVAC.S_NO= ISPORUKA.S_NO)
```

## Podupit NOT EXISTS

- Naći šifre onih snabdjevača koji isporučuju sve crvene proizvode.

```
SELECT ISPORUKA.S_NO
FROM ISPORUKA
WHERE NOT EXISTS
  (SELECT P_NO
   FROM PROIZVOD
   WHERE BOJA='Crvena' AND NOT EXISTS
     (SELECT *
      FROM ISPORUKA X
      WHERE X.S_NO= ISPORUKA.S_NO
        AND X.P_NO= PROIZVOD.P_NO))
```

## SLOŽENIJI INSERT

- INSERT INTO može da bude sa podupitom

```
INSERT INTO SNABDJEVAC(S_NO, S_IME, STATUS, GRAD)
SELECT S_NO+'d', S_IME, STATUS, GRAD
FROM SNABDJEVAC
```

## SELECT - INTO

- Kreiranje tabele pomoću SELECT upita
- SELECT \* INTO POMOCNA FROM SNABDJEVAC

## Složeniji UPDATE

- ALTER TABLE ISPORUKA ADD  
IME\_SNAB VARCHAR(20)
- UPDATE ISPORUKA  
SET IME\_SNAB = SNABDJEVAC.S\_IME  
FROM SNABDJEVAC  
WHERE SNABDJEVAC.S\_NO= ISPORUKA.S\_NO
- UPDATE ISPORUKA  
SET IME\_SNAB = (SELECT SNABDJEVAC.S\_IME  
FROM SNABDJEVAC  
WHERE SNABDJEVAC.S\_NO=  
ISPORUKA.S\_NO)  
WHERE ISPORUKA.KOL>500

## SLOŽENIJI DELETE

- DELETE FROM POMOCNA  
FROM SNABDJEVAC  
WHERE SNABDJEVAC.S\_NO= POMOCNA.S\_NO
- DELETE FROM POMOCNA  
WHERE EXISTS(SELECT \*  
FROM SNABDJEVAC  
WHERE SNABDJEVAC.S\_NO=  
POMOCNA.S\_NO)