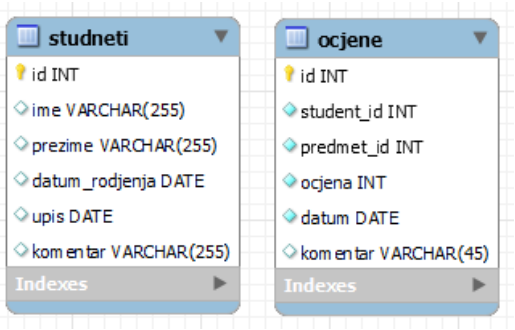
Predavanje 2-4



SQL operatori AND, OR, NOT

U prethodnim slučajevima, kada se selektovala vrijednost iz tabele, to se radilo po samo jednom uslovu. Npr. *sve ocjene za studenta pod id=3*. Ili npr. ako je potrebno vratiti sve studente sa ocjenom 5:

```
SELECT * FROM ocjene WHERE ocjena=5
```

Međutim, nekad želimo da dobijemo sve unose gdje ocjena ***nije*** 5. Za tu svrhu se koristi operator `NOT` koji se stavlja prije uslova.

```
SELECT * FROM ocjene WHERE NOT ocjena=5
```

Pored NOT operatora, postoji niz drugih operatora. `AND` i OR operatori daju mogućnost dodatnog filtriranja po jednoj ili više kolona. Evo nekoliko primjera:

```
SELECT * FROM ocjene WHERE ocjena>3 AND student_id=3
SELECT * FROM ocjene WHERE ocjena=4 OR ocjena=5 AND predmet_id=1
SELECT * FROM ocjene WHERE NOT ocjena=5 AND student_id=2
SELECT COUNT(*) FROM ocjene WHERE ocjena=5 AND student_id=3 AND
predmet_id=2
SELECT * FROM studenti WHERE ime='Alisa' AND NOT prezime='Alisovic'
```

U prevodu, ovi upiti će uraditi sljedeće:

1. Vrati sve ocjene veće od 3 za studenta pod id=3
2. Vrati sve petice i četvrtice na predmetu pod id=1
3. Vrati sve ocjene koje nisu petice za studenta pod id=2
4. Vrati broj petica za studenta pod id=3 na predmetu pod id=2
5. Vrati sve studente sa imenom Alisa ali koja nema prezime Alisovic

Na ovaj način se mogu dodatno filtrirati redovi iz tabele i dobiti željene informacije. Pošto logički operatori mogu biti zbunjujući, moguće je (kao u matematici) grupisati logičke cjeline koristeći zagrade. Zagrade imaju veći prioritet od samih operatora. Npr.:

```
SELECT * FROM studenti WHERE grad='Sarajevo' AND (ime='Alisa' OR ime='Bakir')
```

Ovaj upit će vratiti sve studente koji se zovu Alisa ili Bakir a koje za grad imaju Sarajevo.

SQL Operatori BETWEEN i IN

Ako je potrebno vratiti sve ocjene između 1 i 5 onda možemo napisati:

```
SELECT * FROM ocjene WHERE NOT ocjena=1 AND NOT ocjena=5
SELECT * FROM ocjene WHERE ocjena>1 AND ocjena<5
SELECT * FROM ocjene WHERE ocjena>=2 AND ocjena<=4
```

Svi ovi upiti će dati iste rezultate. Međutim, postoji operator `BETWEEN` koji se može ovako iskoristiti:

```
SELECT * FROM ocjene WHERE ocjena BETWEEN 2 AND 4
```

Ovaj upit vraća unose između dvije vrijednosti (uključivo na oba kraja) i daje iste rezultate kao prethodna tri upita. `BETWEEN` operator selektuje vrijednosti između dva opsega i te vrijednosti mogu biti brojevi, tekst ili datumi, i može se koristiti u kombinaciji i sa drugim operatorima:

```
SELECT * FROM ocjene WHERE ocjena NOT BETWEEN 2 AND 4
SELECT * FROM ocjene WHERE ocjena BETWEEN 2 AND 4 AND (student_id=3 OR student_id=4)
```

Prvi upit vraća sve unose ocjene gdje ocjena nije 2, 3 ili 4. Drugi upit vraća sve unose za studenta pod id=3 ili 4 gdje su ocjene 2, 3 ili 4. Međutim, ovaj upit se može elegantnije napisati koristeći operator `IN`. Ovaj operator potvrđuje da li se traženi pojam nalazi u datom skupu. Tako, drugi upit se može napisati i kao:

```
SELECT * FROM ocjene WHERE ocjena BETWEEN 2 AND 4 AND student_id IN (3,4)
```

Na ovaj način, može se dodati mnogo veća lista studenata, sa mnogo manje teksta. Ovo radi na istom principu i sa nenumeričkim kolonama. Npr.:

```
SELECT * FROM studenti WHERE ime IN ('Alisa', 'Bakir')
```

Ovaj upit će vratiti sve studente koje za ime imaju jedno od imena u datom skupu. Na ovaj način ne moramo pisati duge OR izraze. Svi ovi operatori se mogu koristiti u kombinaciji jedni sa drugima, a pošto može postojati logička nepreciznost, mogu se koristiti zagrade kako bi se logički izrazi grupisali.

Operator LIKE

U prethodnim primjerima, pretraživani su unosi studenata koji za ime imaju Alisa ili Bakir. Međutim recimo da u tabeli postoje imena poput Alice, Amra, Admir, Bob, Bruno itd. Koristeći = u `WHERE` klauzuli, mogu se dobiti samo precizna podudaranja, tj. podaci moraju u potpunosti odgovarati traženom pojmu. Ali ako je potrebno izvući sve studente koji za prvo slovo imena

imaju A onda se to ne može uraditi na taj način. Tu može pomoći operator `LIKE`. `LIKE` operator je operator koji vrši šablonsko pretraživanje. Radi samo na tekstualnim podacima i može odgovoriti na pitanja, da li podatak počinje ili završava ili možda sadrži određen tekst. Npr.:

```
SELECT * FROM studenti WHERE ime LIKE 'A%'
SELECT * FROM studenti WHERE ime LIKE '%kir'
```

Prvi upit vraća sve studente čije ime počinje sa slovom A. Drugi vraća sve studente čije ime završava sa 'kir'. Npr. Bakir. Znak % predstavlja nula, jedan ili više karaktera. Ako treba tražiti podudaranje samo jednog karaktera, onda se u tu svrhu može koristiti simbol _. Npr.:

```
SELECT * FROM studenti WHERE ime LIKE 'Ali__'
```

Ovdje je rečeno da se vrate svi studenti kojima ime počinje sa Ali a završava sa bilo koja, tačno dva karaktera. Pa će tako ovaj upit pronaći imena poput Alisa, Alice, Alina itd. Mogao se koristiti i simbol % npr. 'Ali%' međutim ovo bi vratilo i neko ime, recimo, Ali, što nije traženo.

Ova dva simbola se mogu i kombinovati. Tako, ako je potrebno pronaći sve unose sa imenom koje ima slovo k na drugom mjestu rekli bi:

```
SELECT * FROM studenti WHERE ime LIKE '_k%'
```

I ovaj operator se može koristiti sa ostalim operatorima. Tako ako želimo npr. sve studente čija imena ne počinju sa Ali, možemo napisati:

```
SELECT * FROM studenti WHERE ime NOT LIKE 'Ali%'
```

ili ako želimo sve studente koji počinju sa A ili B:

```
SELECT * FROM studenti WHERE ime LIKE 'A%' OR ime LIKE 'B%'
```