

2-1 Comparison Operators

- Uslovi komparacije (comparison conditions) se koriste za nalaženje podatka u tabeli kada se poklope traženi kriterijumi

Comparison Operators

- SQL ima druge operatore koji dodaju funkcionalnost za vraćanje posebnih setova podataka (BETWEEN...AND, IN, LIKE)

BETWEEN...AND

- Operator BETWEEN...AND se koristi za selektovanje i prikaz redova baziranim na opsegu vrednosti
- Ako je unutar WHERE rečenice, BETWEEN...AND uslov će vratiti opseg vrednosti između i uključujući donji i gornji limit
- U primeru iz Employees db, vrednosti vraćene uključuju donji i gornji limit vrednosti
- Za vrednosti specificirane sa BETWEEN uslovom se kaže da su inclusive
- Takođe, donja limit vrednost mora biti prva prikazana:

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 9000 AND 11000;
```

- Primetiti da je izlaz uključio donju i gornju limit vrednost
- Korišćenje BETWEEN...AND je isto kao i : WHERE salary >= 9000 AND salary <=11000;

LAST_NAME	SALARY
Zlotkey	10500
Abel	11000
Hunold	9000

IN

- IN uslov se takođe naziva “membership condition”; koristi se za testiranje da li vrednost u IN ima specificirani set vrednosti
- Npr, IN može da se koristi za identifikaciju studenata čiji identifikacioni brojevi su 2349, 7354 ili 4333 ili ljudi čiji internacionalni telefonski pozivni kod je 1735, 82 ili 10

```
SELECT city, state_province, country_id
FROM locations
WHERE country_id IN('UK', 'CA');
```

CITY	STATE_PROVINCE	COUNTRY_ID
Toronto	Ontario	CA
Oxford	Oxford	UK

- U ovom primeru, WHERE rečenica se takođe može napisati kao set od OR uslova:

```
SELECT city, state_province, country_id
FROM locations
WHERE country_id IN('UK', 'CA');
...
WHERE country_id = 'UK' OR country_id = 'CA';
```

LIKE

- Menadžer može znati da prezime zaposlenog počinje sa “S” ali ne zna celo ime zaposlenog; ipak u SQL, LIKE uslov omogućava da se izabere red koji odgovara ili karakteru, datumu ili šemi brojeva

- Dva simbola, % i _, se nazivaju wildcard karakteri, se mogu koristiti za konstrukciju stringa potrage
- Simbol (%) se koristi za predstavljanje bilo koje sekvence od nula ili više karaktera
- Simbol underscore (_) se koristi za predstavljanje jednog karaktera
- U sledećem primeru, svi zaposleni sa prezimenom koje počinje sa bilo kojim slovom posle kojeg je "o" a zatim sledi bilo koji broj slova će biti vraćeni

```
SELECT last_name
FROM employees
WHERE last_name LIKE '_o%';
```

LAST_NAME
Kochhar
Lorentz
Mourgos

- Koje prezime će se vratiti iz prethodnog upita ?
 - 1.Sommersmith
 - 2.Oog
 - 3.Fong
 - 4.Mo
 Tačna su sva četiri odgovora
- Dodatne opcije: Kada vam treba tačno podudaranje za string koji ima % ili _ karakter, treba ukazati da % ili _ nije wildcard ali jeste deo stvari koje tražite.
- ESCAPE opcija se može koristiti za ukazivanje da _ ili % je deo imena, ne wildcard vrednost
- Npr, ako želimo vratiti employee JOB_ID iz employee tabele koja sadrži šemu _R, treba koristiti escape karakter da bi se reklo da tražimo za _ a ne bilo koji drugi karakter

```
SELECT last_name, job_id
FROM EMPLOYEES
WHERE job_id LIKE '%\_R%' ESCAPE '\';
```

- Ovaj primer koristi \ kao escape karakter ali bilo koji karakter se može koristiti
- Bez ESCAPE opcije, svi zaposleni koji imaju R u njihovom JOB_ID će se vratiti:

```
SELECT last_name, job_id
FROM EMPLOYEES
WHERE job_id LIKE '%_R%'
```

LAST_NAME	JOB_ID
Abel	SA_REP
Davies	ST_CLERK
Ernst	IT_PROG
Fay	MK_REP
Fay	MK_REP
Grant	SA_REP
Higgins	AC_MGR
Hunold	IT_PROG
King	AD_PRES
Lorentz	IT_PROG
Matos	ST_CLERK
Rajs	ST_CLERK
Taylor	SA_REP
Vargas	ST_CLERK

IS NULL, IS NOT NULL

- NULL je vrednost koja je nedostupna, nedodeljena, nepoznata ili neprimenjiva
- Često je poželjno testirati za NULL

- Možda želiš znati sve datume u Junu kojima nije zakazan koncert ili sve klijente koji nemaju telefonske brojeve zapisane u db
- IS NULL uslov se testira za nedostupne, nedodeljene ili nepoznate podatke
- IS NOT NULL testira za podatke kojima se može prići u db
- U sledećem primeru, WHERE rečenica je napisana da bi vratila sva prezimena onih zaposlenih koji nemaju menadžera

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL;
```

LAST_NAME
King

- Employee King je predsednik kompanije pa zato nema menadžera

```
SELECT last_name, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL;
```

LAST_NAME	COMMISSION_PCT
Zlotkey	.2
Abel	.3
Taylor	.2
Grant	.15

- IS NOT NULL vraća redove koji imaju vrednost u commission_pct koloni
- Kada se upoređuju NULLS ne mogu se koristiti = ili != operatori
- Upit će raditi ali neće vratiti podatke, pošto prava vrednost od NULL nije poznata, pa kako da proverimo da li je nešto jednako ili ne sa vrednosti koju neznamo.