POGLAVLJE 1

Učitavanje zapisa

U ovom poglavlju bavićemo se osnovnim naredbama SELECT. Važno je da dobro upoznate osnove – ne samo što je više tema koje ćemo ovde obraditi deo težih recepata, već ćete na njih nailaziti i pri svakodnevnoj upotrebi SQL-a.

Učitavanje svih redova i kolona iz tabele

Problem

Imate tabelu i želite da vidite sve podatke koji se u njoj nalaze.

Rešenje

Upotrebite specijalni znak „*“ (zvezdica) i izvršite komandu SELECT nad tabelom:

```
1 select *
2   from emp
```

Objašnjenje

Simbol „*“ ima posebno značenje u SQL-u. Kada ga upotrebite, dobićete vrednosti iz svih kolona tabele koju navedete. Budući da u primeru nije zadata odredba WHERE, upit će učitati i sve redove tabele. Druga mogućnost bila bi da pojedinačno navedete ime svake kolone:

```
select empno,ename,job,sal,mgr,hiredate,comm,deptno
  from emp
```

U ad hok upitima koje izvršavate u interaktivnom načinu rada, jednostavnije je da upotrebite SELECT *. Međutim, kada pišete programski kôd, bolje je da kolone navodite pojedinačno. Performanse će biti jednake, ali ako su imena kolona eksplicitno navedena, uvek ćete znati koje kolone upit učitava. Osim toga, tako napisane upite lakše će razumeti drugi ljudi (oni možda znaju, ali možda i ne znaju koje sve kolone postoje u tabelama na koje se upit odnosi).

Učitavanje podskupa redova iz tabele

Problem

Imate tabelu i hoćete da vidite samo redove koji ispunjavaju određeni uslov.

Rešenje

Upotrebite odredbu WHERE da biste zadali koje redove hoćete da izdvojite. Na primer, da biste prikazali sve zaposlene u odeljenju čija je šifra 10:

```
1 select *
2   from emp
3  where deptno = 10
```

Objašnjenje

Odredba WHERE omogućava da učitate samo redove koji vas zanimaju. Ako za određeni red izraz u odredbi WHERE ima logičku vrednost true, taj red se učitava.

Većina sistema za rad s relacionim bazama podataka podržava uobičajene operatore, na primer: =, <, >, <=, >=, !, <>. Može se dogoditi da vam budu potrebni redovi koji ispunjavaju više uslova; zahtev ćete napisati pomoću logičkih operatora AND, OR i zagrada, što je prikazano u narednom receptu.

Pronalaženje redova koji ispunjavaju više uslova

Problem

Želite da učitate redove koji ispunjavaju više uslova.

Rešenje

U odredbi WHERE upotrebite operatore OR i AND. Na primer, ako hoćete da pronađete sve zaposlene iz odeljenja čija je šifra 10, zajedno sa svim zaposlenima koji rade za proviziju i svim zaposlenima u odeljenju 20 čija je plata jednaka 2000 dolara ili manja od toga:

```
1 select *
2   from emp
3  where deptno = 10
4     or comm is not null
5     or sal <= 2000 and deptno=20
```

Objašnjenje

Uslove možete kombinovati pomoću operatora AND i OR, te zagrada, kako biste učitali redove koji ispunjavaju više uslova. U navedenom primeru rešenja, odredba WHERE zadaje da:

- polje DEPTNO sadrži vrednost 10, ili
- polje COMM je NULL, ili
- iznos plate je 2000 dolara ili manji za zaposlenog koji radi u odeljenju (DEPTNO) 20.

Zagrade čine da se uslovi navedeni unutar njih tretiraju kao jedna celina.

Na primer, pogledajmo kako se menja skup rezultata ako upit napišete sa zagradama, kao što sledi:

```
select *
 from emp
where (    deptno = 10
        or comm is not null
        or sal <= 2000
      )
  and deptno=20

EMPNO ENAME  JOB       MGR HIREDATE       SAL       COMM DEPTNO
----- ------ ------ ----- ----------- ----- ---------- ------
 7369 SMITH  CLERK   7902 17-DEC-1980   800                20
 7876 ADAMS  CLERK   7788 12-JAN-1983  1100                20
```

Učitavanje podskupa kolona iz tabele

Problem

Imate tabelu i želite da vidite vrednosti samo iz određenih kolona, a ne iz svih.

Rešenje

Navedite pojedinačno kolone koje vas zanimaju. Na primer, da biste videli ime, šifru odeljenja i iznos plate zaposlenih:

```
1 select ename,deptno,sal
2   from emp
```

Objašnjenje

Kada u odredbi SELECT pojedinačno navedete kolone, time obezbeđujete da upit ne učitava suvišne podatke. To može biti naročito važno kada podatke učitavate preko mreže jer tako izbegavate gubljenje vremena na učitavanje podataka koji vam ne trebaju.

Privremeno dodeljivanje drugačijih imena kolonama

Problem

Želeli biste da imena kolona koje upit učitava izmenite tako da budu čitljivija i razumljivija. Pogledajte sledeći upit koji učitava plate i provizije svih zaposlenih:

```
1 select sal,comm
2   from emp
```

Šta bi bilo „sal“? Da li je to skraćenica za „sale“ (prodaja)? Je li to nečije ime? A šta je „comm“? Da li je u pitanju nekakva komunikacija? Odgovaralo bi vam da rezultati imaju smislenija zaglavlja.

Rešenje

Da biste izmenili imena kolona u rezultatima upita, upotrebite rezervisanu reč AS u obliku: *`izvorno_ime AS novo_ime`*. U nekim bazama podataka AS nije obavezno, ali ga sve prihvataju:

```
1 select sal as salary, comm as commission
2   from emp

SALARY COMMISSION
------- ----------
    800
   1600        300
   1250        500
   2975
   1250       1300
   2850
   2450
   3000
   5000
   1500          0
   1100
    950
   3000
   1300
```

Objašnjenje

Upotreba rezervisane reči AS radi privremenog dodeljivanja drugačijih imena kolonama koje upit učitava, zove se *dodeljivanje alijasa* (engl. *aliasing*) tih kolona. Imena koja tako dajete poznata su kao *alijasi* (engl. *aliases*). Dodeljivanje dobrih alijasa može značajno doprineti da upit i njegovi rezultati budu razumljiviji drugima.

Referenciranje kolone sa alijasima pomoću odredbe WHERE

Problem

Zadali ste alijase da biste u skupu rezultata upita imali kolone sa smislenim imenima, a pomoću odredbe WHERE želite da izostavite određene redove. Ipak, vaš pokušaj da u odredbi WHERE referencirate kolonu pomoću njenog alijasa završava se neuspehom:

```
select sal as salary, comm as commission
  from emp
 where salary < 5000
```

Rešenje

Ako upit prepravite tako da upotrebite lokalni prikaz (engl. *inline view*), možete referencirati kolone sa alijasima:

```
1 select *
2   from (
3 select sal as salary, comm as commission
4   from emp
5        ) x
6  where salary < 5000
```

Objašnjenje

U ovom jednostavnom primeru, možete izbeći upotrebu lokalnog prikaza i umesto toga referencirati kolone COMM ili SAL direktno u odredbi WHERE i postići ćete isti rezultat. Ovo rešenje vas dovodi do pitanja šta bi trebalo uraditi kada u odredbi WHERE pokušavate da referencirate bilo šta od sledećeg:

- agregatne funkcije (engl. *aggregate functions*)
- skalarne podupite
- analitičke funkcije (engl. *window functions*)
- alijase.

Ako upit koji sadrži alijase umetnete u obliku lokalnog prikaza, njegove kolone sa alijasima moći ćete da referencirate u spoljnom upitu. Zašto biste to morali da radite? Budući da se odredba WHERE obrađuje pre odredbe SELECT, to znači da kolone SALARY i COMMISSION još ne postoje kada se obrađuje odredba WHERE „problematičnog“ upita. Ti alijasi ne važe dok se ne završi obrada odredbe WHERE. Međutim, odredba FROM obrađuje se pre odredbe WHERE. Kada izvorni upit umetnete u odredbu FROM, rezultati tog upita generišu se pre obrade spoljne odredbe WHERE i ona zato „vidi“ alijase. Ova tehnika je naročito korisna kada kolone tabela nemaju baš najprikladnije izabrana imena.

Lokalni prikaz u ovom rešenju ima alijas X. Ne zahtevaju sve baze podataka da lokalni prikaz ima alijas, ali je u nekima obavezan. Sve ga prihvataju.

Nadovezivanje vrednosti iz kolona

Problem

Želite da vrednosti iz više kolona tabele učitate u jednu kolonu rezultata. Na primer, želeli biste da upit koji se odnosi na tabelu EMP daje sledeće rezultate:

```
CLARK WORKS AS A MANAGER
KING WORKS AS A PRESIDENT
MILLER WORKS AS A CLERK
```

Međutim, podaci koji vam trebaju da biste generisali ovaj skup rezultata potiču iz dve kolone tabele EMP – ENAME i JOB:

```
select ename, job
  from emp
 where deptno = 10

ENAME      JOB
---------- ---------
CLARK      MANAGER
KING       PRESIDENT
MILLER     CLERK
```

Rešenje

Pronađite i upotrebite ugrađenu funkciju koja u vašem DBMS-u omogućava nadovezivanje vrednosti iz više kolona.

DB2, Oracle, PostgreSQL

U ovim bazama podataka operator nadovezivanja je dvostruka uspravna crta:

```
1 select ename||' WORKS AS A '||job as msg
2   from emp
3  where deptno=10
```

MySQL

Ova baza podataka podržava funkciju čije je ime CONCAT:

```
1 select concat(ename, ' WORKS AS A ',job) as msg
2   from
3  where deptno=10
```

SQL Server

Upotrebite operator „+“ da biste nadovezali podatke:

```
1 select ename + ' WORKS AS A ' + job as msg
2   from emp
3  where deptno=10
```

Objašnjenje

Upotrebite funkciju CONCAT da biste nadovezali vrednosti iz više kolona. Simbol || je zamena za funkciju CONCAT u sistemima DB2, Oracle i PostgreSQL, dok je + zamena u SQL Serveru.

Upotreba uslovne logike u naredbi SELECT

Problem

Želite da u naredbi SELECT izvršavate operacije tipa IF-ELSE. Na primer, hoćete da formirate skup rezultata u kome se, ako je plata zaposlenog 2000 dolara ili manje od toga, pojavljuje reč „UNDERPAID“ (premalo plaćen); ako je plata zaposlenog 4000 dolara ili više, pojavljuje se reč „OVERPAID“ (previše plaćen); ako je plata između navedenih granica, pojavljuje se reč „OK“. Trebalo bi da skup rezultata izgleda ovako:

```
ENAME             SAL STATUS
---------- ---------- ---------
SMITH              800 UNDERPAID
ALLEN             1600 UNDERPAID
WARD              1250 UNDERPAID
JONES             2975 OK
MARTIN            1250 UNDERPAID
BLAKE             2850 OK
CLARK             2450 OK
SCOTT             3000 OK
KING              5000 OVERPAID
TURNER            1500 UNDERPAID
ADAMS             1100 UNDERPAID
JAMES              950 UNDERPAID
FORD              3000 OK
MILLER            1300 UNDERPAID
```

Rešenje

Upotrebite izraz CASE da biste ugradili uslovnu logiku direktno u naredbu SELECT:

```
1 select ename,sal,
2        case when sal <= 2000 then 'UNDERPAID'
3             when sal >= 4000 then 'OVERPAID'
4             else 'OK'
5        end as status
6   from emp
```

Objašnjenje

Izraz CASE omogućava da primenite uslovnu logiku na podatke koje upit učitava. Izrazu CASE možete dodeliti alijas da bi upit vraćao razumljiviji skup rezultata. U navedenom rešenju, vidite da je rezultatu izraza CASE dodeljen alijas STATUS. Odredba ELSE nije obavezna. Ako izostavite ELSE, izraz CASE će vraćati NULL za svaki red koji ne ispunjava uslov.

Ograničavanje broja učitanih redova

Problem

Želite da ograničite broj redova koje upit učitava. Redosled podataka je nevažan; odgovara vam svaki skup od *n* redova.

Rešenje

Upotrebite ugrađenu funkciju iz vaše baze podataka da biste zadali broj redova koje upit učitava.

DB2

U sistemu DB2 upotrebite odredbu FETCH FIRST:

```
1 select *
2   from emp fetch first 5 rows only
```

MySQL i PostgreSQL

Isto možete postići u MySQL-u i PostgreSQL-u pomoću odredbe LIMIT:

```
1 select *
2   from emp limit 5
```

Oracle

U Oracleu, ograničite broj redova koje upit učitava tako što ćete u odredbi WHERE zadati uslov za vrednost funkcije ROWNUM:

```
1 select *
2   from emp
3  where rownum <= 5
```

SQL Server

Pomoću rezervisane reči TOP ograničite broj redova koje upit učitava:

```
1 select top 5 *
2   from emp
```

Objašnjenje

U mnogim sistemima za upravljanje bazama podataka postoje odredbe tipa FETCH FIRST i LIMIT, koje omogućavaju da zadate broj redova koje upit treba da čita. Oracle se razlikuje po tome što morate da upotrebite funkciju ROWNUM, koja vraća redni broj svakog učitanog reda (rastuća vrednost koja počinje od 1).

Evo šta se događa kada zadate ROWNUM <= 5 da biste učitali prvih pet redova:

1. Oracle izvršava upit.
2. Oracle učitava prvi red i dodeljuje mu redni broj 1.
3. Da li smo premašili red broj 5? Ako nismo, Oracle uključuje taj red u skup rezultata zato što ispunjava uslov da njegov redni broj mora biti jednak ili manji od broja 5. Ako jesmo, Oracle ne uključuje red u skup rezultata.
4. Oracle učitava sledeći red i povećava redni broj (na 2, pa na 3, zatim na 4 itd.).
5. Vraćamo se na korak 3.

Kao što se vidi iz navedenog postupka, vrednosti koje vraća Oracleova funkcija ROWNUM dodeljuju se *posle* učitavanja reda. To je veoma važna činjenica. Mnogi programeri koji pišu kôd za Oracle pokušavaju da učitaju, na primer, samo peti red iz rezultata upita, tako što zadaju ROWNUM = 5. Upotreba uslova jednakosti u kombinaciji s funkcijom ROWNUM nije dobro rešenje. Evo šta se događa kada pokušate da učitate, na primer, samo peti red pomoću izraza ROWNUM = 5:

1. Oracle izvršava upit.
2. Oracle učitava prvi red i dodeljuje mu redni broj 1.
3. Da li smo došli do reda broj 5? Ako je odgovor odričan, Oracle odbacuje učitani red jer ne ispunjava uslove. Ako je odgovor potvrdan, Oracle uključuje red u skup rezultata. Ali odgovor neće nikad biti potvrdan!
4. Oracle učitava sledeći red i dodeljuje mu redni broj 1 zato što prvi red koji je upit učitao mora biti numerisan kao red broj 1.
5. Vraćamo se na korak 3.

Pažljivo proučite opisani postupak i shvatićete zašto se upotreba izraza ROWNUM = 5 da biste učitali samo red broj 5 završava neuspehom. Ne možete imati peti red ako prethodno ne učitate redove od jedan do četiri!

Možda ste uočili da ROWNUM = 1, u stvari, ispravno radi i vraća prvi red, što vam se možda čini u suprotnosti s dosadašnjim objašnjenjima. ROWNUM = 1 ispravno radi i vraća prvi red zato što Oracle mora barem jedanput da pokuša da učita red da bi utvrdio ima li uopšte redova u tabeli. Pažljivo pročitajte prethodni postupak u kome ćete 5 zameniti sa 1 i shvatićete zbog čega je ispravno da zadate ROWNUM = 1 kao uslov (da bi upit vratio samo jedan red).

Učitavanje *n* nasumično izabranih zapisa iz tabele

Problem

Želite da učitate određeni broj nasumično izabranih zapisa iz tabele. Nameravate da izmenite sledeću naredbu tako da pri svakom izvršavanju vraća različit skup od pet zapisa:

```
select ename, job
  from emp
```

Rešenje

Upotrebite bilo koju ugrađenu funkciju iz vašeg DBMS-a koja omogućava generisanje nasumično izabranih vrednosti. Tu funkciju navedite u odredbi ORDER BY da biste sortirali redove nasumičnim redosledom. A zatim, primenite tehniku opisanu u prethodnom receptu da biste ograničili broj zapisa koji upit učitava.

DB2

Upotrebite ugrađenu funkciju RAND u kombinaciji sa odredbama ORDER BY i FETCH:

```
1 select ename,job
2   from emp
3  order by rand( ) fetch first 5 rows only
```

MySQL

Upotrebite ugrađenu funkciju RAND u kombinaciji sa odredbama LIMIT i ORDER BY:

```
1 select ename,job
2   from emp
3  order by rand( ) limit 5
```

PostgreSQL

Upotrebite ugrađenu funkciju RANDOM u kombinaciji sa odredbama LIMIT i ORDER BY:

```
1 select ename,job
2   from emp
3  order by random( ) limit 5
```

Oracle

Upotrebite ugrađenu funkciju VALUE, koja se nalazi u ugrađenom paketu DBMS_RANDOM, u kombinaciji sa odredbom ORDER BY i ugrađenom funkcijom ROWNUM:

```
1 select *
2   from (
```

```
3  select ename, job
4    from emp
6   order by dbms_random.value( )
7         )
8   where rownum <= 5
```

SQL Server

Upotrebite ugrađenu funkciju NEWID u kombinaciji sa odredbama TOP i ORDER BY da biste učitali skup nasumičnih rezultata:

```
1  select top 5 ename,job
2    from emp
3   order by newid( )
```

Objašnjenje

Odredba ORDER BY može da prihvati povratnu vrednost zadate funkcije i na osnovu nje izmeni redosled redova u skupu rezultata. Svi upiti navedeni u rešenju, ograničavaju broj učitanih redova *posle* izvršavanja funkcije navedene u odredbi ORDER BY. Korisnicima sistema koji nisu Oracle, može pomoći ako prouče rešenje za Oracle jer ono pokazuje (konceptualno) šta se dešava „ispod haube“ drugih rešenja.

Važno je da ne mešate upotrebu funkcije u odredbi ORDER BY sa upotrebom numeričke konstante. Kada u odredbi ORDER BY zadate numeričku konstantu, time zahtevate da se rezultati sortiraju po vrednostima kolone koju ste pod tim rednim brojem naveli na listi kolona u odredbi SELECT. Kada u odredbi ORDER BY zadate funkciju, sortiranje se obavlja na osnovu vrednosti rezultata funkcije za svaki učitani red.

Pronalaženje NULL vrednosti

Problem

Želite da pronađete sve redove koji u određenoj koloni sadrže vrednost NULL.

Rešenje

Da biste utvrdili da li je određena vrednost jednaka NULL, upotrebite IS NULL:

```
1 select *
2   from emp
3  where comm is null
```

Objašnjenje

Budući da NULL nije nikad jednako nečemu, niti različito od nečega, a nije jednako ni samom sebi, ne možete upotrebiti operator =, niti != da biste ispitali sadrži li određena kolona NULL. Kako biste utvrdili da li red sadrži NULL vrednosti, morate upotrebiti IS NULL. Možete takođe upotrebiti IS NOT NULL da biste pronašli redove koji ne sadrže NULL u datoj koloni.

Pretvaranje NULL vrednosti u stvarne vrednosti

Problem

Imate redove koji sadrže NULL vrednosti a želeli biste da umesto njih, skup rezultata upita sadrži neke druge vrednosti.

Rešenje

Upotrebite funkciju COALESCE kako biste NULL vrednosti zamenili stvarnim vrednostima:

```
1 select coalesce(comm,0)
2   from emp
```

Objašnjenje

Funkcija COALESCE prihvata jednu ili više vrednosti kao argumente. Ta funkcija vraća prvu vrednost sa liste argumenata koja nije NULL. U navedenom primeru rešenja, upit vraća vrednost iz kolone COMM kad god je ta vrednost različita od NULL. U ostalim slučajevima vraća nulu.

Kada radite s NULL vrednostima, najbolje je da iskoristite prednosti ugrađene funkcionalnosti koje vam pruža vaš DBMS; u mnogim slučajevima ustanovićete da posao možete obaviti podjednako dobro pomoću više funkcija. Funkcija COALESCE slučajno postoji u svim DBMS-ovima. Osim toga, izraz CASE je takođe podržan u svim DBMS-ovima:

```
select case
       when comm is null then 0
       else comm
       end
  from emp
```

Kada se uporedi prevođenje NULL vrednosti u stvarne, pomoću izraza CASE, uočljivo je da upotreba funkcije COALESCE daje razumljiviji i sažetiji kôd.

Pronalaženje uzoraka

Problem

Želite da učitate samo redove koji sadrže određeni niz znakova ili uzorak. Pogledajte sledeći upit i skup rezultata:

```
select ename, job
  from emp
 where deptno in (10,20)
```

```
ENAME      JOB
---------- ---------
SMITH      CLERK
JONES      MANAGER
CLARK      MANAGER
SCOTT      ANALYST
KING       PRESIDENT
ADAMS      CLERK
FORD       ANALYST
MILLER     CLERK
```

Od zaposlenih koji rade u odeljenjima 10 i 20, želite da učitate samo one čija imena sadrže slovo „I“ ili one čiji se naziv radnog mesta završava na „ER“:

```
ENAME      JOB
---------- ---------
SMITH      CLERK
JONES      MANAGER
CLARK      MANAGER
KING       PRESIDENT
MILLER     CLERK
```

Rešenje

Upotrebite operator LIKE u kombinaciji sa SQL-ovim džokerskim operatorom („%“):

```
1 select ename, job
2   from emp
3  where deptno in (10,20)
4    and (ename like '%I%' or job like '%ER')
```

Objašnjenje

Kada operator procenat („%“) u kombinaciji s operatorom LIKE upotrebite u operaciji utvrđivanja poklapanja sa određenim uzorkom, on daje poklapanje za bilo koji niz znakova. U većini izvedbi SQL-a podržan je i operator podvučeno („_“) koji utvrđuje poklapanje sa samo jednim znakom. Ako uzorak koji tražite, „I“, umetnete između dva operatora „%“, prihvata se svaki niz znakova koji sadrži slovo „I“ (na bilo kojem mestu). Ukoliko uzorak koji tražite ne umetnete između dva operatora „%“, rezultati upita zavisiće od mesta na koje ste postavili operator. Na primer, da biste pronašli nazive radnih mesta koji se završavaju na „ER“, ispred „ER“ postavite operator „%“; ako treba pronaći sve nazive radnih mesta koja počinju sa „ER“, dodajte operator „%“ iza „ER“.