

UVOD U BAZE PODATAKA

Predavanje 11-12

Sistemske funkcije

Uvod

- Svaki RDBMS sadržava skup ugrađenih ili **sistemskih funkcija**
- Većina ih je **skalarna** (vraćaju **jednu** vrijednost)
- Sistemske funkcije služe za obavljanje raznih operacija, a možemo ih podijeliti na:
 - Matematičke funkcije
 - Funkcije za rad s datumom i vremenom
 - Funkcije za rad sa stringovima
 - Ostale funkcije
- Mogu se koristiti u bilo kojem dijelu SELECT naredbe
- Najjednostavniji oblik SELECT naredbe: **SELECT *izraz***

Matematičke funkcije

- Najvažnije matematičke funkcije su:
 - $\text{ROUND}(\text{izraz}, \text{duljina})$ – zaokružuje zadani broj
 - $\text{FLOOR}(\text{izraz})$, $\text{CEILING}(\text{izraz})$ – vraća prvi manji/veći cijeli broj
 - $\text{ABS}(\text{izraz})$ – vraća apsolutnu vrijednost broja
 - $\text{SQUARE}(\text{izraz})$ – vraća kvadrat broja
 - $\text{SQRT}(\text{izraz})$ – vraća drugi korijen broja
 - $\text{SIGN}(\text{izraz})$ – vraća **-1** za negativni broj, **0** za 0 i **1** za pozitivni broj
 - $\text{RAND}()$ – vraća slučajni broj između 0 i 1

Primjeri

1. Vratite sve proizvode s cijenama zaokružanim na dvije decimale.
2. Vratite sve stavke s cijenama zaokružanim na dvije decimale te vratite prvi najmanji i najveći cijeli broj svake cijene.
3. Vratite apsolutni iznos broja -91 .
4. Vratite sve stavke i uz svaki ID stavke vratite njegov korijen i kvadrat.
5. Vratite slučajni cijeli broj između 1 i 100.

Funkcije za rad s datumom i vremenom

- Najvažnije funkcije za rad s datumom i vremenom su:
 - **GETDATE()** – vraća današnji datum i vrijeme
 - **DATEPART(*mjera*, *datum*)** – vraća dio datuma i vremena
 - year, quarter, month, dayofyear, day, week, weekday, hour, minute, second
 - **YEAR(*datum*)**, **MONTH(*datum*)**, **DAY(*datum*)** – ekvivalentno DATEPART funkciji s prvim parametrom jednakim year, month, odnosno day
 - **DATEDIFF(*mjera*, *početak*, *kraj*)** – vraća koliko je vremena proteklo između zadanih datuma i vremena u zadanoj mjeri (gledaju se granice mjere)
 - **DATEADD(*mjera*, *količina*, *datum*)** – zadanom datumu i vremenu dodaje zadanu količinu mjere

Primjeri

6. Vratite trenutno vrijeme i datum.
7. Vratite sve račune za kupca 377 i umjesto datuma izdavanja ispišite godinu i mjesec izdavanja.
8. Vratite sve račune za kupca 377 i ispišite koliko je godina, mjeseci i dana proteklo od izdavanja svakog računa do danas.
9. Vratite datum koji će biti za točno 12.000 dana.

Funkcije za rad sa stringovima

- Najvažnije funkcije za rad sa stringovima:
 - **CHARINDEX(*traženi_string*, *string*, *od*)** – traži string unutar stringa od zadanog položaja i vraća broj znaka na kojem je string pronađen (ili 0 ako ga nema)
 - **LEN(*string*)** – vraća broj znakova u stringu
 - **LOWER(*string*)**, **UPPER(*string*)** – pretvara string u mala/velika slova
 - **LTRIM(*string*)**, **RTRIM(*string*)**, **TRIM(*string*)** – uklanja razmake
 - **REVERSE(*string*)** – okreće string
 - **SUBSTRING(*string*, *od*, *koliko*)** – uzima dio stringa počevši od zadanog znaka i uzevši zadano znakova

Primjeri

10. Vratite sve proizvode i ispišite na kojem mjestu se pojavljuje slovo 'A'.
11. Vratite sve proizvode i ispišite duljinu naziva svakog.
12. Vratite sve proizvode tako da nazive pretvorite u velika slova, boje u mala.
13. Ako imate string ' tekst ', vratite ga bez razmaka.
14. Vratite sve proizvode tako da naziv ispišete od kraja prema početku.
15. Vratite sve proizvode tako da uzmete samo prvih 5 slova naziva.

Ostale funkcije

- Od ostalih funkcija ćemo spomenuti:
 - **CAST(*izraz AS tip_podataka*)** – pretvara zadani izraz u zadani tip podataka
 - **ISDATE(*izraz*)** – vraća je li zadani izraz ispravan datum
 - **ISNUMERIC(*izraz*)** – vraća je li zadani izraz ispravan broj (cijeli ili decimalni)
 - **ISNULL(*izraz, zamjena*)** – vraća izraz ako nije NULL; ako jest, vraća zamjenu

Primjeri

16. Vratite sve proizvode tako da cijene pretvorite u cijele brojeve i u stringove.
17. Vratite je li 'danas' ispravan datum. Vratite je li '2011-08-15' ispravan datum.
18. Vratite je li 'abcd' ispravan broj. Vratite je li '67.55' ispravan broj.
19. Vratite nazive svih proizvoda i uz svaki ispišite boju ako je definirana, odnosno "NIJE DEFINIRANA" ako nije.

Agregatne funkcije

Pregled načina korištenja SELECT naredbe

1. Dohvaćanje stupaca iz tablica

```
SELECT * FROM Racun
```

- Smijemo vratiti bilo koji stupac i n redaka

2. Dohvaćanje agregiranih vrijednosti

```
SELECT MAX(DatumIzdavanja) FROM Racun
```

- Smijemo vratiti samo agregirane stupce, uvijek su u 1 retku

3. Dohvaćanje agregiranih vrijednosti po grupama

```
SELECT KupacID, MAX(DatumIzdavanja)  
FROM Racun GROUP BY KupacID
```

- Smijemo vratiti agregirane stupce i stupce po kojima grupiramo, uvijek su u onoliko redaka koliko ima grupa

Agregatne funkcije

- **Agregatne funkcije** rade na zadanom skupu redaka i **od više vrijednosti računaju jednu vrijednost**
- ISO standard definira sljedeće agregatne funkcije:
 - **AVG** – vraća prosječnu vrijednost
 - **COUNT** – vraća broj elemenata
 - **MAX** – vraća najveću vrijednost
 - **MIN** – vraća najmanju vrijednost
 - **SUM** – vraća zbroj vrijednosti
- Navedene funkcije su dostupne u svakom RDBMS-u
 - Svaki RDBMS može imati prošireni skup funkcija

Korištenje





- Agregatne funkcije se mogu koristiti u:
 - SELECT dijelu
 - HAVING dijelu
 - ORDER BY dijelu

- Primjer:

```
SELECT
    MIN(CijenaBezPDV) AS NajnizaCijena,
    MAX(CijenaBezPDV) AS NajvisaCijena,
    AVG(CijenaBezPDV) AS ProsjecnaCijena
FROM Proizvod
WHERE CijenaBezPDV > 0
```

Česta pogreška

- Ako koristimo agregatne funkcije u SELECT dijelu, **ne možemo vratiti podatke koji nisu agregirani**
- Primjer **neispravnog** upita:

```
SELECT
    MIN(CijenaBezPDV) AS NajnizaCijena, 
    MAX(CijenaBezPDV) AS NajvisaCijena, 
    AVG(CijenaBezPDV) AS ProsjecnaCijena, 
    Boja 
FROM Proizvod
WHERE CijenaBezPDV > 0
```


NULL vrijednosti

- Većina agregatnih funkcija **ignorira NULL vrijednosti**
 - `SELECT MIN(PotkategorijaID) FROM Kupac`
 - Uzimat će u obzir samo retke bez NULL vrijednosti
- Ponekad **želimo računati i s NULL vrijednostima**
 - Moramo NULL vrijednosti pretvoriti u neku vrijednost, primjerice u 0
 - Zgodno koristiti **funkciju ISNULL()**:
`ISNULL(stupac, zamjena)`
 - Ako u stupcu piše NULL, funkcija vraća *zamjena*
 - Ako u stupcu nije NULL, funkcija vraća ono što piše u stupcu

Napomene

- Neke agregatne funkcije rade s **više tipova podataka**
 - Ako AVG radi s `int` podacima, vratit će `int` rezultat
 - Ako AVG radi s `money` podacima, vratit će `money` rezultat
- Za pretvaranje tipova podataka koristimo funkciju **CAST()**:
 - `CAST(vrijednost AS tip)`
 - Primjerice: `CAST(Broj as float)`
- Agregatna funkcija **COUNT** ima specifično ponašanje:
 - `COUNT(stupac)` će vratiti broj redaka koji nemaju `NULL` vrijednost upisanu u *stupac*
 - `COUNT(*)` će vratiti ukupni broj redaka, bez obzira na `NULL` vrijednosti

Primjeri

1. Vratite broj svih proizvoda.
2. Vratite broj proizvoda koji imaju definiranu boju.
3. Vratite najvišu cijenu proizvoda.
4. Vratite prosječnu cijenu proizvoda iz potkategorije 16.
5. Vratite datume najstarijeg i najnovijeg računa izdanog kupcu 131.

Grupiranje

Grupiranje

- **Grupiranje je operacija kojom odabrane retke dijelimo u grupe (razrede, klase)**
 - Grupiranje se radi prema zadanim **kriterijima grupiranja**
 - Svaki **redak pripada samo jednoj grupi**
- Grupiranje se radi pomoću naredbe **GROUP BY**
- Kod grupiranja možemo koristiti i naredbu za filtriranje grupa **HAVING**
 - Služi da bismo uključili samo grupe koje zadovoljavaju zadani uvjet

Izgled SELECT naredbe pri grupiranju

- Proširena sintaksa SELECT naredbe je sljedeća:

```
SELECT 6 DISTINCT TOP 8 količina
      5 grupirajući_stupac1,
      agregatna_funkcija1, ...
FROM 1 izvor_podataka
WHERE 2 uvjet
GROUP BY 3 stupci
HAVING 4 uvjet
ORDER BY 7 stupac1 smjer, ...
```

Primjer grupiranja

```
SELECT  
  Boja, COUNT(*) AS BrojProizvoda  
FROM Proizvod  
WHERE Boja IS NOT NULL  
GROUP BY Boja
```

- Tijek izvršavanja naredbe:
 1. Iz tablice Proizvod (FROM) se uzmu samo reci koji imaju definiranu boju (WHERE)
 2. Rade se grupe redaka prema boji (GROUP BY)
 3. Za svaku grupu se uzme Boja i broj proizvoda u grupi (SELECT)

Napomene

- Prilikom grupiranja u SELECT listi mogu biti navedeni:
 - Stupci po kojima se radi grupiranje
 - Agregatne funkcije
 - **Kod grupiranja, agregatne funkcije se računaju za svaku grupu posebno**
 - Primjerice, sljedeći upit će vratiti onoliko redaka koliko ima različitih boja:

```
SELECT Boja, COUNT(*) AS BrojProizvoda  
FROM Proizvod  
GROUP BY Boja
```


Grupiranje po više kriterija

- U listi za grupiranje može biti navedeno više stupaca
 - Grupiranje će se raditi prema svim zadanim stupcima
 - Primjerice:

```
SELECT
  Boja, PotkategorijaID, COUNT(*) AS Kolicina
FROM Proizvod
GROUP BY Boja, PotkategorijaID
ORDER BY Boja, PotkategorijaID
```

- Dobit ćemo po jednu grupu za svaku kombinaciju boje i potkategorije

Primjeri

1. Ispišite sve boje proizvoda i pokraj svake napišite koliko proizvoda ima tu boju.
2. Promijenite prethodni upit tako da sortirate padajuće prema broju proizvoda.
3. Promijenite prethodni upit tako da isključite nedefiniranu boju.
4. Ispišite koliko proizvoda svake boje ima u svakoj od potkategorija. Sortirajte prema potkategoriji i prema boji.
5. Promijenite prethodni upit tako da ispišete 10 kategorija i boja s najviše proizvoda.

Primjeri

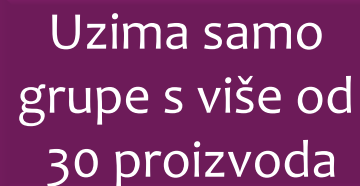
6. Promijenite prethodni upit tako da umjesto ID potkategorije ispišete njen naziv.
7. Ispišite nazive svih kategorija i pokraj svake napišite koliko potkategorija je u njoj.
8. Ispišite nazive svih kategorija i pokraj svake napišite koliko proizvoda je u njoj.
9. Ispišite sve različite cijene proizvoda i napišite koliko proizvoda ima svaku cijenu.
10. Ispišite koliko je računa izdano koje godine.
11. Ispišite brojeve svih račune izdane kupcu 377 i pokraj svakog ispišite ukupnu cijenu po svim stavkama.

Naredba HAVING

- Naredba **HAVING** je opcionalna naredba za **filtriranje prema** napravljenim **grupama, nakon što je obavljeno grupiranje**
 - **WHERE** služi za filtriranje **redaka**
 - **HAVING** služi za filtriranje **grupa**
- Navodi se iza **GROUP BY** naredbe i sadržava **agregatne funkcije**

```
SELECT  
    Boja, COUNT(*) AS BrojProizvoda  
FROM Proizvod  
GROUP BY Boja  
HAVING COUNT(*) > 30
```

Uzima samo
grupe s više od
30 proizvoda



Primjeri

12. Ispišite sve boje koje imaju više od 40 proizvoda.
13. Ispišite nazive svih potkategorija koje sadržavaju više od 10 proizvoda.
14. Ispišite ukupno zarađene iznose i broj prodanih primjeraka za svaki od ikad prodanih proizvoda.
15. Ispišite ukupno zarađene iznose za svaki od proizvoda koji je prodan u više od 2000 primjeraka.
16. Ispišite ukupno zarađene iznose za svaki od proizvoda koji je prodan u više od 2.000 primjeraka ili je zaradio više od 2.000.000 dolara.