

VJEŽBA 4 – FUNKCIJE, AGREGATNE FUNKCIJE I GRUPIRANJE

IZRAČUNATA POLJA

Izračunata polja predstavljaju attribute koji ne postoje u tablici, nego se kreiraju unutar SELECT izraza. Na primjer, iz atributa Ime i Prezime moguće je generirati novi atribut ImePrezime .

POVEZIVANJE VRIJEDNOSTI (ULANČAVANJE)

Ulančavanje vrijednosti (eng. Concatenation) je termin koji se koristi za povezivanje različitih riječi ili znakova u jednu cjelinu. Kod SQL servera, operator ulančavanje je znak plus (+).

```
-- Ispis punog imena svih zaposlenika.

SELECT FirstName + ' ' + LastName
FROM Employees;
```

	(No column name)
1	Nancy Davolio
2	Andrew Fuller
3	Janet Leverling
4	Margaret Peacock
5	Steven Buchanan
6	Michael Suyama
7	Robert King
8	Laura Callahan
9	Anne Dodsworth

MATEMATIČKI IZRAČUNI

Matematički izračuni u SQL jeziku slični su onima iz programskih jezika.

MATEMATIČKI OPERATORI	
Operator	Opis
+	Zbrajanje
-	Oduzimanje
*	Množenje
/	Dijeljenje
%	Modulo

```
SELECT OrderID, Freight, Freight * 1.1
FROM Orders
WHERE Freight >= 500;
```

	OrderID	Freight	(No column name)
1	10372	890.7800	979.85800
2	10479	708.9500	779.84500
3	10514	789.9500	868.94500
4	10540	1007.6400	1108.40400
5	10612	544.0800	598.48800
6	10691	810.0500	891.05500
7	10816	719.7800	791.75800
8	10897	603.5400	663.89400
9	10912	580.9100	639.00100
10	10983	657.5400	723.29400
11	11017	754.2600	829.68600
12	11030	830.7500	913.82500
13	11032	606.1900	666.80900

ALIASI

Uočimo da izračunata polja iz prethodnih primjera nemaju naziv. Ključna riječ AS se koristi za imenovanje polja.

```
SELECT OrderID, Freight, Freight * 1.1 AS FreightTotal
FROM Orders
WHERE Freight >= 500;
```

ODABIR JEDINSTVENIH ZAPISA

Izraz DISTINCT koristi se za dohvaćanje jedinstvenih kombinacija vrijednosti atributa iz tablice. Sljedeći primjer pokazuje kako dohvatiti jedinstvene nazive gradova u kojima Northwind ima zaposlenike.

```
-- Pronađi sve jedinstvene gradove u kojima Northwind ima zaposlenike
SELECT DISTINCT City
FROM Employees
ORDER BY City
```

DISTINCT se često koristi u kombinaciji s agregatnim funkcijama. Sljedeći primjer pokazuje kako se ključna riječ DISTINCT može koristiti kako bi se saznalo u koliko različitih gradova Northwind ima zaposlenike.

```
-- Pronađi u koliko različitih gradova Northwind ima zaposlenike.
```

```
SELECT COUNT(DISTINCT City) AS NumCities
FROM Employees
```

UGRAĐENE FUNKCIJE

Tablica prikazuje neke od uobičajenih matematičkih, string i funkcija za manipuliranje datumima. Funkcije uvelike variraju među različitim sustavima za upravljanje bazama podataka, pa je za svaki sustav potrebno znati sintaksu funkcija koju koristi.

MATEMATIČKE FUNKCIJE

Naziv	Opis
ABS	Apsolutna vrijednost
CEILING	Najmanja integer vrijednost \geq vrijednosti argument funkcije
FLOOR	Najveći integer \leq vrijednosti argument funkcije
POWER	Potencija
ROUND	Zaokruživanje
SQRT	Kvadratni korijen
CAST(num AS decimal(8,2))	Formatiranje broja kao decimalnog na dvije decimale

```
-- Prikaži trošarinu i njezinu zaokruženu vrijednost na jednu decimalu
SELECT Freight, ROUND(Freight,1) AS ApproxFreight
FROM Orders;
```

	Freight	ApproxFreight
1	32.38	32.40
2	11.61	11.60
3	65.83	65.80
4	41.34	41.30
5	51.30	51.30
6	58.17	58.20
7	22.98	23.00
8	148.33	148.30
9	13.97	14.00
10	81.91	81.90
11	140.51	140.50

```
-- Odaberi jediničnu cijenu, te je pretvori u decimalni broj sa dvije decimale
SELECT UnitPrice, CAST(UnitPrice AS Decimal(8,2))
FROM Products;
```

	UnitPrice	(No column name)
1	18.0000	18.00
2	19.0000	19.00
3	10.0000	10.00
4	22.0000	22.00
5	21.3500	21.35
6	25.0000	25.00
7	30.0000	30.00
8	40.0000	40.00
9	97.0000	97.00
10	31.0000	31.00
11	21.0000	21.00
12	38.0000	38.00

Uočite da se zaokruživanje na cjelobrojnu vrijednost vrši prosljeđivanjem 0 kao drugog parametra ROUND(field,0).

STRING FUNKCIJE

Naziv	Opis
LOWER	Prebaci znakove u mala slova (lowercase)
UPPER	Prebaci znakove u velika slova (uppercase)
RTRIM	Ukloni prazan znak s kraja
LTRIM	Ukloni prazan znak s početka
SUBSTRING	Podskup znakova

```
--Odaberi ime i prezime zaposlenika pisano velikim slovima
SELECT UPPER(FirstName), UPPER(LastName)
FROM Employees;
```

	(No column name)	(No column name)
1	NANCY	DAVOLIO
2	ANDREW	FULLER
3	JANET	LEVERLING
4	MARGARET	PEACOCK
5	STEVEN	BUCHANAN
6	MICHAEL	SUYAMA
7	ROBERT	KING
8	LAURA	CALLAHAN
9	ANNE	DODSWORTH

```
-- Odaberi prvih 10 znakova adrese svakog kupca
SELECT SUBSTRING(Address,1,10)
```

FROM Customers;

	(No column name)
1	Obere Str.
2	Avda. de l
3	Mataderos
4	120 Hanove
5	Berguvsväg
6	Forsterstr
7	24, place
8	C/ Araquil
9	12, rue de
10	23 Tsawass
11	Fauntleroy

DATUMSKE FUNKCIJE

Naziv	Opis
DATEADD	Dodavanje vrijednosti datumu
DATEDIFF	Oduzimanje vrijednosti datumu
DATENAME	Pretvaranje datuma u string
DATEPART	Pretvaranje datuma u broj
GETDATE	Dohvaćanje trenutnog datuma i vremena

-- Pronaći prije koliko godina je svaki od zaposlenika zaposlen.

```
SELECT LastName, BirthDate, HireDate, DATEDIFF(year,HireDate, GETDATE()) AS HireAge
FROM Employees
ORDER BY HireDate;
```

	LastName	BirthDate	HireDate	HireAge
1	Leverling	1963-08-30 00:00:00.000	1992-04-01 00:00:00.000	20
2	Davolio	1959-01-01 00:00:00.000	1992-05-01 00:00:00.000	20
3	Fuller	1952-02-19 00:00:00.000	1992-08-14 00:00:00.000	20
4	Peacock	1937-09-19 00:00:00.000	1993-05-03 00:00:00.000	19
5	Buchanan	1955-03-04 00:00:00.000	1993-10-17 00:00:00.000	19
6	Suyama	1963-07-02 00:00:00.000	1993-10-17 00:00:00.000	19
7	King	1960-05-29 00:00:00.000	1994-01-02 00:00:00.000	18
8	Callahan	1958-01-09 00:00:00.000	1994-03-05 00:00:00.000	18
9	Dodsworth	1966-01-27 00:00:00.000	1994-11-15 00:00:00.000	18
10	ert	1966-01-27 00:00:00.000	1994-11-15 00:00:00.000	18

```
-- Pronaći mjesec rođenja svakog zaposlenika
```

```
SELECT FirstName, LastName, DATENAME(month,BirthDate) AS BirthMonth
FROM Employees
ORDER BY DATEPART(month,BirthDate);
```

	FirstName	LastName	BirthMonth
1	Laura	Callahan	January
2	Anne	Dodsworth	January
3	Andrew	Fuller	February
4	Steven	Buchanan	March
5	Robert	King	May
6	Michael	Suyama	July
7	Janet	Leverling	August
8	Margaret	Peacock	September
9	Nancy	Davolio	December

AGREGATNE FUNKCIJE I GRUPIRANJE

AGREGATNE FUNKCIJE

Agregatne funkcije koriste se za izračunavanje rezultata korištenjem vrijednosti atributa iz više zapisa.

Ako želimo postaviti upit koji traži maksimalnu, minimalnu, prosječnu vrijednost nekog atributa u retcima ili jednostavno prebrojava sve retke ili uz određeni uvjet, koristimo agregatne funkcije.

U najosnovnijem obliku agregatne funkcije vraćaju skalarnu vrijednost:

```
SELECT AgregatnaFunkcija(atribut) FROM tablica
```

Dakle, agregatne funkcije se izvršavaju nad skupom vrijednosti, ali vraćaju jednu, skalarnu vrijednost.

Postoji nekoliko uobičajenih agregatnih funkcija:

UOBIČAJENE AGREGATNE FUNKCIJE	
Agregatna funkcija	Opis
COUNT(*)	Ukupan broj redaka koji imaju ne-NULL vrijednosti zadanih polja
COUNT(atribut)	Ukupan broj redaka (bez NULL vrijednosti) navedenog atributa
COUNT(DISTINCT atribut)	Ukupan broj različitih vrijednosti atributa
SUM()	Suma ne-NULL vrijednosti zadanih polja.
AVG()	Prosjek ne-NULL vrijednosti zadanih polja.
MAX()	Maksimalna vrijednost među ne-NULL vrijednostima zadanih polja.
MIN()	Minimalna vrijednost među ne-NULL vrijednostima zadanih polja.

```
-- Nađi broj zaposlenika
SELECT COUNT(*) AS NumEmployees
FROM Employees;

-- Nađi ukupnu količinu koja je naručena za ProductID = 3
SELECT SUM(Quantity) AS TotalUnits
FROM "Order Details"
WHERE ProductID=3;

-- Nađi prosječnu jediničnu cijenu artikala
SELECT AVG(UnitPrice) AS AveragePrice
FROM Products;

-- Nađi najraniji i najnoviji datum zaposlenja zaposlenika
SELECT MIN(HireDate) AS FirstHireDate,
MAX(HireDate) AS LastHireDate
FROM Employees;
```

GRUPIRANJE PODATAKA

GROUP BY

Korištenjem GROUP BY izraza, agregatne funkcije se mogu primijeniti na grupe zapisa kreirane temeljem zajedničkih vrijednosti određenih polja.

```
-- Ispiši atribut1 te njegovu prosječnu vrijednost
SELECT atribut1, AVG(atribut2) FROM Tablica GROUP BY atribut1
```

Važno:

Ako navodimo nakon SELECT izraza više atributa te agregatnu funkciju **neizostavno je** u GROUP BY izrazu dodati **sve attribute koji se nalaze nakon SELECT naredbe**.

```
SELECT atribut1, atribut2, atribut3, AGR_FUNKCIJA(atribut4)
FROM tablica
GROUP BY atribut1, atribut2, atribut3
```

Bitno je spomenuti da redoslijedom u GROUP BY izrazu ne mijenjamo odabir redaka nego samo njihov poredak u rezultirajućoj relaciji.

Zaključak: GROUP BY je dodan u SQL zato što agregatne funkcije vraćaju agregiranu vrijednost svih zapisa te je bez GROUP BY-a nemoguće naći primjerice, prosječnu vrijednost svake grupe atributa zasebno.

```
--Broj zaposlenika po svakom gradu.  
SELECT City, COUNT(EmployeeID) AS NumEmployees  
FROM Employees  
GROUP BY City;
```

	City	NumEmployees
1	Kirkland	1
2	London	4
3	Redmond	1
4	Seattle	2
5	Tacoma	1

HAVING

Izraz HAVING koristi se za filtriranje grupiranih podataka.

Želimo ispisati sve vrijednosti atributa1 kod kojih je $SUM(atribut2) > 1000$.

Promotrimo sljedeći upit

```
SELECT atribut1, SUM(atribut2) FROM tablica GROUP BY atribut1 WHERE atribut2 > 1000
```

Takav kod će dati **neispravan rezultat**, tj. tražit će one rezultate koje inicijalno u tablici sadrže vrijednosti atribut2 veću od 1000, a ne traženo. HAVING uvjet je dodan u SQL iz razloga što WHERE ne može biti korišten nad agregatnim funkcijama te bi **bez HAVING bilo nemoguće vršiti uvjete nad agregatnim funkcijama**. Ispravan izraz glasi:

```
SELECT atribut1, SUM(atribut2) FROM tablica GROUP BY atribut1 HAVING SUM(atribut2) > 1000
```

U HAVING bloku možemo izvršavati uvjete isključivo nad atributima navedenim u GROUP BY bloku i nad agregatnim funkcijama.

```
/* Broj zaposlenika po gradovima za one gradove koji imaju barem dva zaposlenika */  
SELECT City, COUNT(EmployeeID) AS NumEmployees
```



```
FROM Employees

GROUP BY City

HAVING COUNT(EmployeeID) > 1;
```

	City	NumEmployees
1	London	4
2	Seattle	2

REDOSLIJED NAREDBI

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. HAVING
6. ORDER BY

```
/* Pronađi broj agenata prodaje u svakom gradu koji ima barem dva agenta.
Sortiraj prema broju agenata. */
```

```
SELECT City, COUNT(EmployeeID) AS NumEmployees
FROM Employees
WHERE Title = 'Sales Representative'
GROUP BY City
HAVING COUNT(EmployeeID) > 1
ORDER BY NumEmployees;
```

	City	NumEmployees
1	London	3

PRAVILA GRUPIRANJA

- Svaki ne-agregirani atribut koji se pojavljuje u SELECT izrazu mora biti naveden liu GROUP BY izrazu.
- U HAVING izrazu ne smiju se koristiti aliasi.
- U ORDER BY izrazu mogu se koristiti aliasi.
- U HAVING izrazu smiju se koristiti samo izračunate (agregirane) vrijednosti.
- U ORDER BY izrazu mogu se koristiti aliasi za agregirana ili ne-agregirana polja.