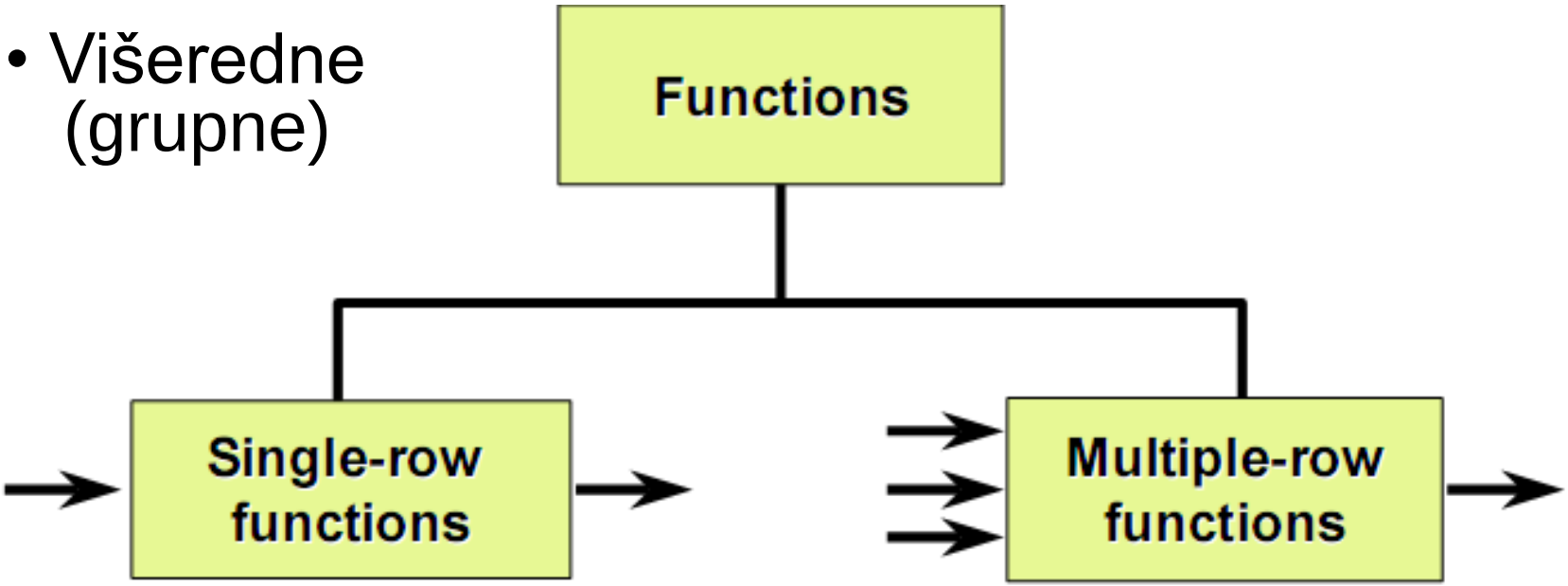
Višeredne funkcije

FUNKCIJE

Postoje dva različita tipa funkcija:

- Jednoredne
- Višeredne (grupne)



Group functions

Neke funkcije SQL-a se izvršavaju nad celom tabelom ili nad specifičnim grupama redova. Pišu se u klauzuli SELECT. Svaka funkcija vraća jedan rezultat. To su:

- AVG
- COUNT
- MIN
- MAX
- SUM
- VARIANCE
- STDDEV

```
SELECT group_function(column), …
FROM table
WHERE condition;
```

Group functions

- **MIN**: Koristi se nad kolonama sa podacima bilo kog tipa i vraća najmanju vrednost.
- **MAX**: Koristi se nad kolonama sa podacima bilo kog tipa i vraća najveću vrednost.
- **SUM:** Koristi se nad kolonama sa numeričkim podacima i vraća total ili sumu svih vrednosti.
- **AVG**: Koristi se nad kolonama sa numeričkim podacima i izračunava prosek.
- **COUNT**: Vraća broj redova

Grupne funkcije

ORACLE Academy

```
SELECT MAX(salary)
FROM employees;
```

DEPT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
…	…
60	11000
60	8600
	7000
10	4400

MAX (SALARY)
24000

ORACLE Academy

Group functions

Kolika je prosečna cena događaja u bazi podataka DJs on demand?

D_EVENTS

ID	NAME	…	…	COST	…
100	Peters Graduation			8000	
105	Vigil wedding			10000	

```
SELECT AVG(cost)
FROM d_events;
```

AVG(COST)
9000

Group functions

Odredite:

broj zaposlenih,

najmanju platu,

datum najkasnijeg zaposlenja,

prezime lica koje je na vrhu abecednog spiska

prezime koje je na kraju spiska

zaposlenih iz tabele EMPLOYEES.

Uzmite u obzir samo zaposlene iz odeljenja 50 i 60.

ORACLE Academy

Group functions

```
SELECT COUNT(employee_id) "Ukupno radnika",
       MIN(salary) "Najmanja plata",
       MAX(hire_date) "Najnoviji",
       MIN(last_name) "Prvi na spisku",
       MAX(last_name) "Na kraju spiska"
FROM employees
WHERE department_id IN (50, 60);
```

Ukupno radnika	Najmanja plata	Najnoviji	Prvi na spisku	Na kraju spiska
8	2500	16-NOV-99	Davies	Vargas

Pravila

- Grupne funkcije zanemaruju null-vrednosti.
- Grupne funkcije se ne mogu koristiti u klauzuli WHERE.
- MIN, MAX i COUNT se mogu koristiti sa bilo kojim tipom podatka.
- SUM, AVG, STDDEV i VARIANCE se mogu koristiti samo sa numeričkim tipom podataka.

COUNT DISTINCT

COUNT(kolona_za_brojanje) broji redove u kojima kolona _za_brojanje ima ne-null vrednost. Ako se ista vrednost kolone kolona _za_brojanje javlja u više redova, svaki red će biti uračunat.

Ako želimo da izbrojimo samo različite vrednosti kolone kolona _za_brojanje, koristićemo COUNT(DISTINCT kolona_za_brojanje)

PRIMER

kol_1	kol_2	kol_3	kol-4
3	-	aba	-
6	a	-	-
-	a	-	-
3	a	aca	-

COUNT(kol_1) = 3
COUNT(DISTINCT kol_1) = 2
COUNT(kol_2) = 3
COUNT(DISTINCT kol_2) = 1
COUNT(kol_3) = 2
COUNT(DISTINCT kol_3) = 2

ORACLE Academy

COUNT DISTINCT

D_CDS

CD_NUMBER	TITLE	PRODUCER	YEAR
90	The Celebrants Live in Concert	Old Town Records	1997
91	Party Music for All Occasions	The Music Man	2000
92	Back to the Shire	Middle Earth Records	2002
93	Songs from My Childhood	Old Town Records	1999
94	Carpe Diem	R & B Inc.	2000
95	Here Comes the Bride	The Music Man	2001
96	Graduation Songbook	Tunes Are Us	1998
98	Whirled Peas	Old Town Records	2004

```
SELECT COUNT(year)
FROM d_cds;
```

COUNT(YEAR)
8

ORACLE Academy

COUNT DISTINCT

D_CDS

CD_NUMBER	TITLE	PRODUCER	YEAR
90	The Celebrants Live in Concert	Old Town Records	1997
91	Party Music for All Occasions	The Music Man	2000
92	Back to the Shire	Middle Earth Records	2002
93	Songs from My Childhood	Old Town Records	1999
94	Carpe Diem	R & B Inc.	2000
95	Here Comes the Bride	The Music Man	2001
96	Graduation Songbook	Tunes Are Us	1998
98	Whirled Peas	Old Town Records	2004

```
SELECT COUNT(DISTINCT year) as "Years"
FROM d_cds;
```

Years
7

ORACLE Academy

COUNT(*)

COUNT(kolona_za_brojanje) broji redove u kojima kolona _za_brojanje ima ne-null vrednost i koji zadovoljavaju WHERE-uslov ako je dat.

COUNT(*) koristimo ako želimo da izbrojimo sve redove, ili sve redove koji zadovoljavaju dati uslov ako je data klauzula WHERE.

PRIMER

kol_1	kol_2	kol_3	kol-4
3	-	aba	-
6	a	-	-
-	a	-	-
3	a	aca	-

COUNT(kol_1) = 3
COUNT(kol_2) = 3
COUNT(kol_3) = 2
COUNT(kol_4) = null
COUNT(*) = 4

Koje vrednosti daju sledeći upiti?

SELECT COUNT(kol_2) FROM primer WHERE kol_1< 6;	**1**
SELECT COUNT(*) FROM primer WHERE kol_1< 6;	**2**

ORACLE Academy

AVG, NVL

```
SELECT AVG(kolona)
FROM tabela;
```

Prosek će biti izračunat samo za one redove u kojima posmatrana kolona nije null.

PRIMER

kol_1	kol_2	kol_3	kol-4
3	-	aba	-
6	a	-	-
-	a	-	-
3	a	aca	-

```
SELECT AVG(kol_1)
FROM primer;
```

Rezultat je
(3+6+3) / 3 = 4

Međutim, nekada je potrebno da pri izračunavanju proseka uzmemo u obzir i redove u kojima je posmatrana kolona null. Recimo, ako imamo četiri radnika u grupi, od kojih dvojica naprave po 3 proizvoda, jedan napravi 6, a jedan ništa, po koliko proizvoda su oni napravili u proseku?

Pri izračunavanju proseka moramo uzeti u obzir svu četvoricu, a ne samo onu trojicu koji su napravili nešto. U SQL-u za takve slučajeve koristimo funkciju NVL pomoću koje zamenjujemo null-vrednost nulom.

```
SELECT AVG(NVL(kol_1, 0))
FROM primer;
```

Rezultat je
(3+6+0+3) / 4 = 3

ORACLE Academy

AVG, NVL

SELECT id, first_name, last_name, auth_expense_amt FROM d_partners

ID	FIRST_NAME	LAST_NAME	AUTH_EXPENSE_AMT
11	Jennifer	cho	-
22	Jason	Tsang	-
33	Allison	Plumb	300000
12	Anna	Smith	20
13	Jan	Smith	-

SELECT AVG(auth_expense_amt)
FROM d_partners;

AVG(AUTH_EXPENSE_AMT)
150010

(300000+20) / 2 = 150010

Zanemareni su redovi sa null-vrednostima

SELECT
AVG(NVL(auth_expense_amt,0))
FROM d_partners;

AVG(AUTH_EXPENSE_AMT)
60004

(0+0+300000+20+0) / 5 = 60004

Sve null-vrednosti su zamenjene sa 0 pa su uključeni svi redovi.

ORACLE Academy

GROUP BY

```
SELECT AVG(salary)
FROM employees;
```

AVG(SALARY)
8775

Ovaj upit nam daje informaciju o prosečnoj plati na nivou cele tabele EMPLOYEES. A ako nas interesuje prosek po pojedinim odeljenjima (departments), mogli bismo to da preciziramo klauzulom WHERE:

```
SELECT department_id, AVG(salary)
FROM employees WHERE department_id=10
```

DEPARTMENT_ID	AVG(SALARY)
10	4400

```
SELECT department_id, AVG(salary)
FROM employees WHERE department_id=20
```

DEPARTMENT_ID	AVG(SALARY)
20	9500

I tako dalje.

Ali, ovo se efikasnije rešava klauzulom **GROUP BY**

ORACLE Academy

GROUP BY

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id;
```

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.333333333333333333333333333333
90	19333.333333333333333333333333333333
110	10150
-	7000

Da bismo izlaznoj informaciji dali željeni oblik, koristićemo ROUND, NVL i druge funkcije i klauzule SQL-a.

Pravila

- Ako u klauzulu SELECT uključite grupnu funkciju (AVG, SUM, COUNT, MAX, MIN, STDDEV, VARIANCE) i još neku pojedinačnu kolonu, svaka tako navedena pojedinačna kolona mora biti navedena i u klauzuli GROUP BY.
- Alijasi kolona se ne mogu koristiti u klauzuli GROUP BY.
- Klauzula WHERE isključuje vrste (redove) pre njihove podele na grupe. Zato se grupne funkcije ne mogu koristiti u klauzuli WHERE. Za definisanje uslova selekcije na osnovu vrednosti grupnih funkcija koristi se klauzula HAVING

ORACLE Academy

HAVING

Kao što se klauzula WHERE koristi da se ograniči izbor redova, tako se klauzula HAVING koristi za ograničen izbor grupa.

Ako upit koristi klauzule GROUP BY i HAVING, prvo se grupišu redovi, zatim se izračunavaju grupne funkcije, a zatim se prikazuju samo one grupe koje su saglasne sa klauzulom HAVING.

Klauzula WHERE se koristi da ograniči izbor redova koji se grupišu; klauzula HAVING ograničava izbor grupa dobijenih klauzulom GROUP BY.

```
SELECT department_id, MAX(salary)
FROM employees
WHERE department_id < 80
GROUP BY department_id
HAVING COUNT(*) > 1;
```

DEPARTMENT_ID	MAX(SALARY)
20	13000
50	5800
60	9000

Klauzula HAVING

ORACLE Academy

```
SELECT department_id, MAX(salary),
                    COUNT(*) "Broj zaposlenih"
FROM employees
GROUP BY department_id;
```

DEPARTMENT_ID	MAX(SALARY)	Broj zaposlenih
-	7000	1
90	24000	3
20	13000	2
110	12000	2
80	11000	3
50	5800	5
10	4400	1
60	9000	3

```
SELECT department_id, MAX(salary)
FROM employees
WHERE department_id
                    IN (10, 20, 80)
GROUP BY department_id
HAVING COUNT(*)
                    BETWEEN 2 AND 5
ORDER BY MAX(salary);
```

DEPARTMENT_ID	MAX(SALARY)
80	11000
20	13000

Klauzula HAVING

ORACLE Academy

HAVING

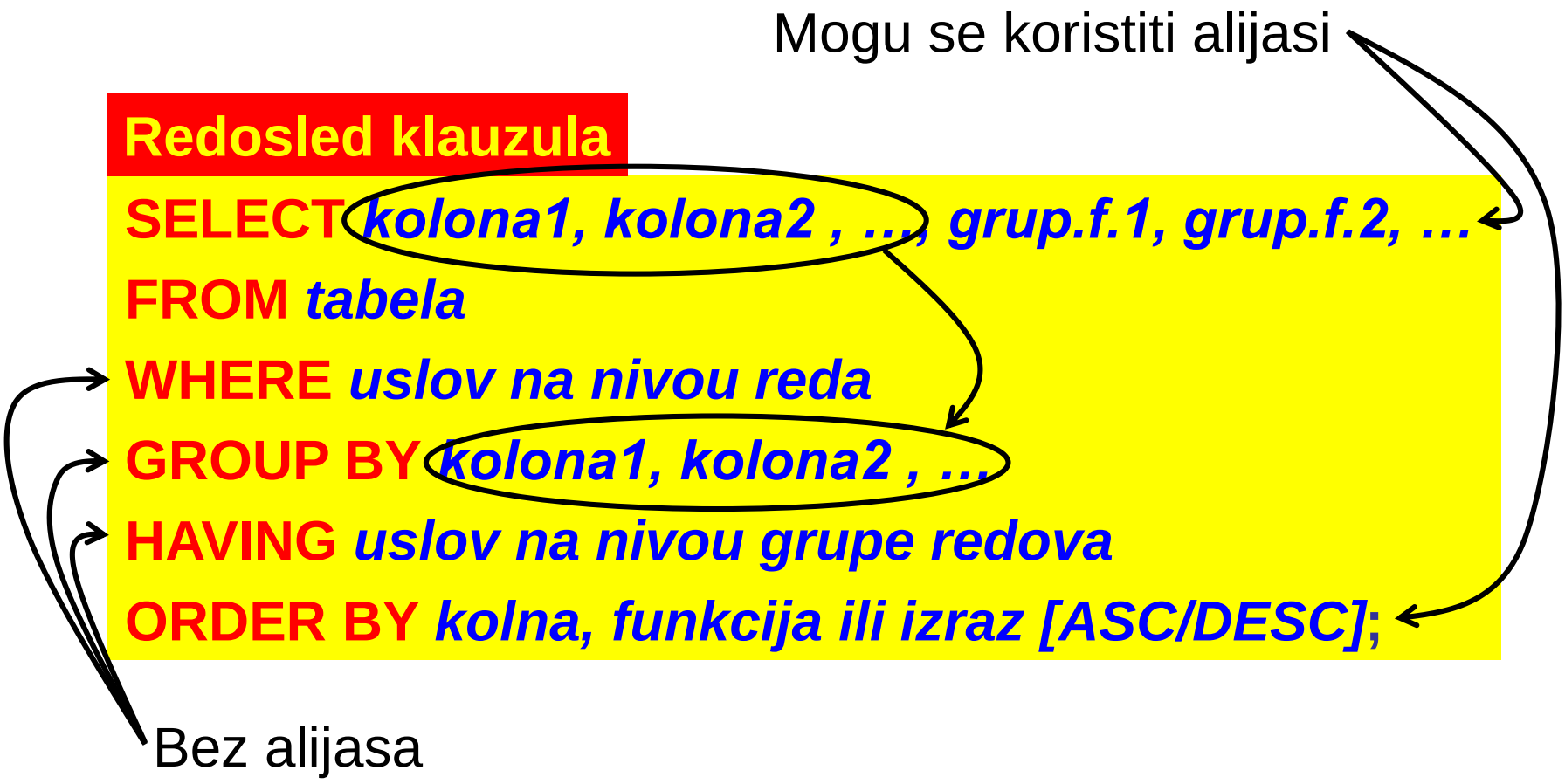
```
SELECT department_id, AVG(salary)
FROM employees
WHERE AVG(salary)<10000
GROUP BY department_id;
```

ORA-00934:
group function is not allowed here

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
HAVING AVG(salary)<10000;
```

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
-	7000

Grupne funkcije



```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id ;
```

DEPARTMENT_ID	AVG(SALARY)
-	7000
90	19333.333333333333333333333333
20	9500
110	10150
80	10033.333333333333333333333333
50	3500
10	4400
60	6400

```
SELECT department_id, AVG(salary)
FROM employees
WHERE department_id>10
GROUP BY department_id;
```

DEPARTMENT_ID	AVG(SALARY)
20	9500
50	3500
60	6400
80	10033.333333333333333333
90	19333.333333333333333333
110	10150

Grupne funkcije

ORACLE Academy

```
SELECT department_id, AVG(salary)
FROM employees
WHERE department_id>10
GROUP BY department_id
HAVING AVG(salary)<10000;
```

DEPARTMENT_ID	AVG(SALARY)
20	9500
50	3500
60	6400

Grupne funkcije

ORACLE Academy

```
SELECT department_id, AVG(salary)
FROM employees
WHERE department_id>10
GROUP BY department_id
HAVING AVG(salary)<10000
ORDER BY AVG(salary) ;
```

DEPARTMENT_ID	AVG(SALARY)
50	3500
60	6400
20	9500

Grupne funkcije

ORACLE Academy

```
SELECT department_id, AVG(salary)
FROM employees
WHERE department_id>10
GROUP BY department_id
HAVING AVG(salary)<10000
ORDER BY AVG(salary) ASC;
```

DEPARTMENT_ID	AVG(SALARY)
50	3500
60	6400
20	9500

Grupne funkcije

ORACLE Academy

```
SELECT department_id, AVG(salary)
FROM employees
WHERE department_id>10
GROUP BY department_id
HAVING AVG(salary)<10000
ORDER BY AVG(salary) DESC;
```

DEPARTMENT_ID	AVG(SALARY)
20	9500
60	6400
50	3500

Grupne funkcije

ORACLE Academy

```
SELECT department_id odeljenje, AVG(salary) prosek
FROM employees
WHERE department_id>10
GROUP BY department_id
HAVING AVG(salary)<10000
ORDER BY prosek DESC;
```

ODELJENJE	PROSEK
20	9500
60	6400
50	3500