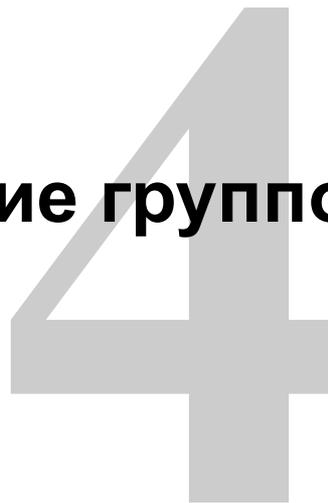


# Использование групповых функций



# Objectives

После завершения этого урока вы должны знать :

- Что такое групповые функции и как их использовать
- Как производить группировку с помощью GROUP BY
- Как производить включение или исключение сгруппированных строк с помощью HAVING

# Что такое групповая функция?

Групповые функции работают с наборами строк, чтобы дать один результат в каждой группе.

**EMPLOYEES**

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
	7000
10	4400

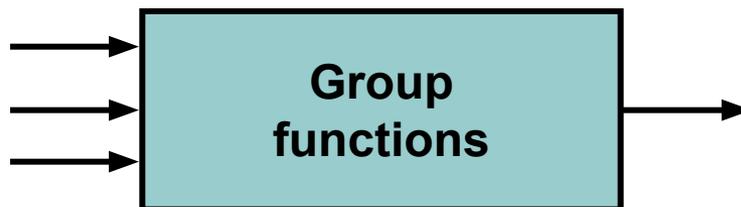
**Maximum salary in  
EMPLOYEES table**

MAX(SALARY)
24000

...  
20 rows selected.

# Типы групповых функций

- **AVG**
- **COUNT**
- **MAX**
- **MIN**
- **SUM**
- ...



# Функции Группа: Синтаксис

```
SELECT      [column,] group_function(column), ...  
FROM        table  
[WHERE      condition]  
[GROUP BY  column]  
[ORDER BY  column];
```

# Использование функций AVG и SUM

Вы можете использовать AVG и SUM для числовых данных.

```
SELECT AVG(salary) , MAX(salary) ,  
       MIN(salary) , SUM(salary)  
FROM   employees  
WHERE  job_id LIKE '%REP%';
```

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8150	11000	6000	32600

# Использование функций MIN и MAX

Вы можете использовать MAX и MIN для типов **numeric, character, date**

```
SELECT MIN(hire date), MAX(hire date)
FROM   employees;
```

MIN(HIRE_	MAX(HIRE_
17-JUN-87	29-JAN-00

# Использование COUNT функции

**COUNT (\*) возвращает количество строк в таблице :**

1

```
SELECT COUNT (*)  
FROM employees  
WHERE department_id = 50;
```

COUNT(\*)

5

**COUNT (expr) возвращает количество строк с ненулевых значений для expr:**

2

```
SELECT COUNT (commission_pct)  
FROM employees  
WHERE department_id = 80;
```

COUNT(COMMISSION\_PCT)

3

# Использование DISTINCT

- COUNT (DISTINCT expr) возвращает число различных ненулевых значениях *expr*.
- Для того, чтобы отобразить количество различных значений отдела в таблице EMPLOYEES :

```
SELECT COUNT(DISTINCT department id)  
FROM employees;
```

```
COUNT(DISTINCTDEPARTMENT_ID)
```

```
7
```

# Групповые функции и значения Null

Групповые функции игнорируют столбцы со значением null :

1

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

AVG(COMMISSION\_PCT)

.2125

Функция NVL позволяет включать нулевые значения:

2

```
SELECT AVG (NVL (commission_pct, 0))
FROM employees;
```

AVG(NVL(COMMISSION\_PCT,0))

.0425

# Создание групп данных

## EMPLOYEES

DEPARTMENT_ID	SALARY
10	4400
20	13000
20	6000
50	5800
50	3500
50	3100
50	2500
50	2600
60	9000
60	6000
60	4200
80	10500
80	8600
80	11000
90	24000
90	17000

...

20 rows selected.

4400

9500

3500

6400

10033

**Average  
salary in  
EMPLOYEES  
table for each  
department**

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

# Создание групп данных: Синтаксис предложения GROUP BY

```
SELECT    column, group_function(column)
FROM      table
[WHERE    condition]
[GROUP BY group_by_expression]
[ORDER BY column];
```

Вы можете разделить строки в таблице на более мелкие группы при помощи предложения GROUP BY.

# Использование предложения GROUP BY

Все столбцы в списке выбора, к которым не применяются групповые функции должны быть описаны в предложении 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	10033.3333
90	19333.3333
110	10150
	7000

8 rows selected.

# Использование предложения GROUP BY

Столбец в GROUP не обязательно должен находиться в SELECT .

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

AVG(SALARY)	
	4400
	9500
	3500
	6400
	10033.3333
	19333.3333
	10150
	7000

# Группировка по нескольким столбцам

## EMPLOYEES

DEPARTMENT_ID	JOB_ID	SALARY
90	AD_PRES	24000
90	AD_VP	17000
90	AD_VP	17000
60	IT_PROG	9000
60	IT_PROG	6000
60	IT_PROG	4200
50	ST_MAN	5800
50	ST_CLERK	3500
50	ST_CLERK	3100
50	ST_CLERK	2600
50	ST_CLERK	2500
80	SA_MAN	10500
80	SA_REP	11000
80	SA_REP	8600

...

20	MK_REP	6000
110	AC_MGR	12000
110	AC_ACCOUNT	8300

20 rows selected.

**Add the salaries in the EMPLOYEES table for each job, grouped by department**

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

# Использование GROUP BY на несколько колонок

```
SELECT department_id dept_id, job_id, SUM(salary)
FROM employees
GROUP BY department_id, job_id ;
```

DEPT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

# Некорректные Запросы

## Использование групповых функций

Любой столбец или выражение в списке **SELECT**, который не является агрегатной функцией должен быть описан в предложении **GROUP BY**:

```
SELECT department_id, COUNT(last_name)
FROM employees;
```

```
SELECT department_id, COUNT(last_name)
      *
ERROR at line 1:
ORA-00937: not a single-group group function
```

**Колонка отсутствуют в списке предложения  
GROUP BY**

# Некорректные Запросы

## Использование групповых функций

- Вы не можете использовать групповые функции в **WHERE**.
- Для этой цели используйте **HAVING**.

```
SELECT    department_id, AVG(salary)
FROM      employees
WHERE     AVG(salary) > 8000
GROUP BY department_id;
```

```
WHERE    AVG(salary) > 8000
        *
ERROR at line 3:
ORA-00934: group function is not allowed here
```

**Cannot use the WHERE clause to restrict groups**

# Ограничение результатов группировки

## EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600

...

20	6000
110	12000
110	8300

20 rows selected.

The maximum salary per department when it is greater than \$10,000

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

# Ограничение результатов группировки с использованием HAVING

При использовании предложения HAVING, сервер Oracle ограничивает группы следующим образом:

1. Строки сгруппированы.
2. Применяется групповая функция.
3. Отображаются группы, соответствующие предложению HAVING.

```
SELECT      column, group_function
FROM        table
[WHERE      condition]
[GROUP BY  group_by_expression]
[HAVING    group_condition]
[ORDER BY  column];
```

# Использование предложения HAVING

```
SELECT    department_id, MAX(salary)
FROM      employees
GROUP BY  department_id
HAVING    MAX(salary) > 10000 ;
```

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

# Использование предложения HAVING

```
SELECT  job_id, SUM(salary) PAYROLL
FROM    employees
WHERE   job_id NOT LIKE '%REP%'
GROUP BY job_id
HAVING  SUM(salary) > 13000
ORDER BY SUM(salary);
```

JOB_ID	PAYROLL
IT_PROG	19200
AD_PRES	24000
AD_VP	34000

# Nesting Group Functions

Отображение максимальной средней заработной платы:

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

MAX(AVG(SALARY))
19333.3333

# Summary

Вы научились:

Использовать групповые функции COUNT, MAX, MIN и AVG

Писать запросы, которые используют GROUP BY

Писать запросы, которые используют HAVING

```
SELECT      column, group_function
FROM        table
[WHERE      condition]
[GROUP BY  group_by_expression]
[HAVING    group_condition]
[ORDER BY  column];
```

# Practice 4: Overview

**This practice covers the following topics:**

- **Writing queries that use the group functions**
- **Grouping by rows to achieve more than one result**
- **Restricting groups by using the `HAVING` clause**



1. Количество сотрудников по департаментам.
  - Оставить только департаменты с средней ЗП > 7000
  
2. Количество уникальных JOB\_ID в разрезе департаментов
  
3. Вывести одним запросом дату найма первого сотрудника и дату найма последнего сотрудника в компании.
  
4. Вывести количество сотрудников, являющихся руководителями.
  
- 5.\* Вывести среднюю ЗП руководителей.
  
- 6.\* Размер 2 по величине ЗП.

**\* Задачи «на вырост». Темы мы еще не проходили.  
Можно погуглить «использование подзапросов»**

