

Базы данных

Лекция 7

Язык SQL

Функции

- Однострочные: возвращают результат для каждой строки;
- Многострочные: возвращают результат для нескольких строк;

Однострочные функции

- СИМВОЛЬНЫЕ:
 - LOWER(string_value)
 - UPPER(string_value)
 - SUBSTRING(string_value, first_ch_number, ch_count)
 - LEN(string_value)
 - LEFT(string_value, ch_count)
 - RIGHT(string_value, ch_count)
 - LTRIM(string_value)
 - RTRIM(string_value)
 - REPLACE(string_value, pattern, replacement)
 - ...

Примеры

```
SELECT employee_id, last_name, department_id  
FROM employees  
WHERE LOWER(last_name) = 'higgins';
```

```
SELECT employee_id, last_name, job_id  
FROM employees  
WHERE UPPER(SUBSTRING(job_id, 4, 3)) = 'REP';
```

Однострочные функции

- Числовые:
 - ROUND(value, number_of_digits)
 - ABS(value),
 - RAND([seed])
 - SIN(value)
 - COS(value)
 - POWER(value, degree)
 - ...

Примеры

```
SELECT ROUND(45.923, 2), ROUND(45.923, 0), ROUND(45.923,  
-1)  
FROM Table_1;
```

```
SELECT employee_id, last_name, job_id  
FROM employees  
WHERE UPPER(SUBSTRING(job_id, 4, 3)) = 'REP';
```

Однострочные функции

- Дата/время:
 - SYSDATETIME()
 - SYSDATETIMEOFFSET()
 - GETDATE()
 - DATENAME(datepart, date)
 - DAY(date)
 - MONTH(date)
 - YEAR(date)
 - DATEADD (datepart , number , date)
 - ...

Приведение типов

- Неявное
- Явное
 - CAST(expression AS type)
 - CONVERT(type, expression[, style])

Однострочные функции

- Обработка NULL

- NULLIF(expression1, expression2)
- COALESCE(expression1, expression2, ...)

```
SELECT last_name, salary, COALESCE(commission_pct, 0),  
(salary*12) + (salary*12*COALESCE(commission_pct, 0)) AS  
AN_SAL  
FROM employees;
```

```
SELECT first_name, LEN(first_name), last_name, LEN(last_name),  
NULLIF(LEN(first_name), LEN(last_name)) result  
FROM employees;
```

```
SELECT last_name, employee_id, COALESCE(CONVERT(varchar,  
commission_pct), CONVERT(varchar, manager_id), 'No  
commission or manager')  
FROM employees;
```

Условные выражения

CASE expression

```
WHEN comparison_expr1 THEN return_expr1  
[WHEN comparison_expr2 THEN return_expr2 ...]  
[ELSE else_return_expr]  
END
```

```
SELECT last_name, job_id, salary,  
CASE job_id  
WHEN 'IT_PROG' THEN 1.10*salary  
WHEN 'IT_CLERK' THEN 1.15*salary  
WHEN 'SA_REP' THEN 1.20*salary  
ELSE salary  
END AS "Revised Salary"  
FROM employees;
```

Многострочные функции

- AVG
- COUNT
- MAX
- MIN
- SUM

```
SELECT group_function([DISINCT] expression), ...  
FROM table_name
```

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

Группировки

GROUP BY group_by_expression

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

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

Группировки

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

Примеры ошибочных запросов:

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

```
SELECT department_id, job_id, COUNT(last_name)
FROM employees
GROUP BY department_id;
```

Группировки

```
SELECT department_id, job_id, SUM(salary)
FROM employees
WHERE department_id > 40
GROUP BY department_id, job_id;
```

Некорректный запрос:

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

Группировки. HAVING

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

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