



# Базы данных и Информационные системы

5/15 Простые запросы к БД

Кузиков Б.О.  
Сумы, СумГУ  
2013

# Таблицы, используемые в курсе

## EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	1500		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7811	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7812	WARD	SALESMAN	7698	09-APR-81	1250	500	30
7813	MALCOLM	CLERK	7698	03-DEC-81	950		30
7814	DEWNEY	CLERK	7698	22-FEB-81	1250	500	30
7815	SMITH	CLERK	7566	03-DEC-81	3000		20
7902	BLAKE	CLERK	7902	17-DEC-80	800		20
7566	JONES	MANAGER	7566	09-DEC-82	3000		20
7788	SCOTT	ANALYST	7788	12-JAN-83	1100		20
7782	CLARK	MANAGER	7782	23-JAN-82	1300		10

## DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

# Простое выражение SELECT

---

```
SELECT    [DISTINCT] {*, column [alias], ...}  
FROM      table;
```

- SELECT определяет *какие* столбцы
- FROM определяет *из какой* таблицы

# Написание выражений SQL

---

- ▶ Выражения SQL не чувствительны к регистру.
- ▶ Выражения SQL могут состоять из одной или нескольких строк.
- ▶ Ключевые слова не могут переноситься и сокращаться.
- ▶ Предложения обычно пишутся в различных строках.
- ▶ Табуляции и отступы используются для улучшения читаемости.

```
SELECT    [DISTINCT] {*, column [alias],...}  
FROM      table;
```

# Выборка всех столбцов

---

```
SELECT *  
FROM dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

# Выборка конкретных столбцов

---

```
SELECT deptno, loc  
FROM dept;
```

DEPTNO	LOC
10	NEW YORK
20	DALLAS
30	CHICAGO
40	BOSTON

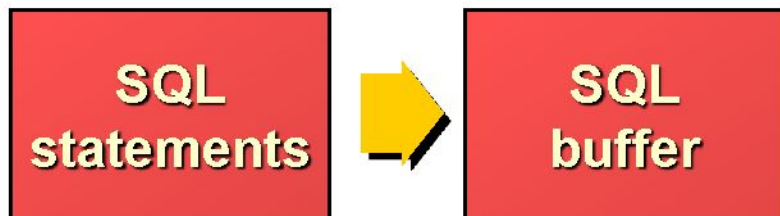


# Сравнение выражений SQL и команд SQL\*Plus

---

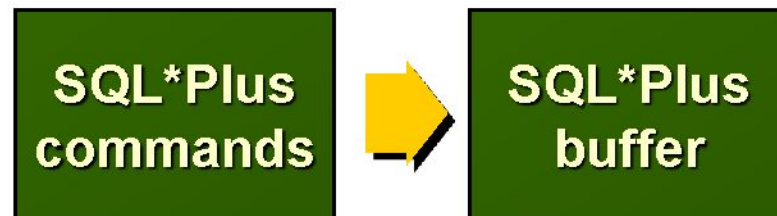
## SQL

- Язык
- Стандарт ANSI
- Ключевые слова не могут быть сокращены
- Выражения взаимодействуют с данными и таблицами в БД



## SQL\*Plus

- Среда
- Разработка Oracle
- Ключевые слова могут быть сокращены
- Команды не позволяют манипулировать данными БД





# Обзор SQL\*Plus

---

- ▶ Вход в SQL\*Plus.
- ▶ Просмотр структуры таблиц.
- ▶ Редактирование выражений SQL.
- ▶ Выполнение выражений SQL из SQL\*Plus.
- ▶ Запись и добавление выражений SQL в файлы.
- ▶ Запуск сохраненных файлов.
- ▶ Загрузка из файлов в буфер с возможностью редактирования.

# Вход SQL\*Plus

---

В командной строке:

```
sqlplus [username[/password[@database]]]
```

# Команды редактирования SQL\*Plus

---

- A[PPEND] *text*
- C[HANGE] / *old* / *new*
- C[HANGE] / *text* /
- CL[EAR] BUFF[ER]
- DEL
- DEL *n*
- DEL *m n*

# Команды редактирования SQL\*Plus

---

- I[NPUT]
- I[NPUT] *text*
- L[IST]
- L[IST] *n*
- L[IST] *m n*
- R[UN]
- *n*
- *n text*
- *0 text*

# Использование LIST, n, и APPEND

---

## **LIST**

```
1  SELECT  ename
2*  FROM    emp
```

## **1**

```
1*  SELECT  ename
```

## **A , job**

```
1*  SELECT  ename, job
```

## **LIST**

```
1  SELECT  ename, job
2*  FROM    emp
```

# Команда CHANGE

---

**LIST**

```
1* SELECT * from emp
```

**c/emp/dept**

```
1* SELECT * from dept
```

**LIST**

```
1* SELECT * from dept
```

# Файловые команды SQL\*Plus

---

- `SAVE filename`
- `GET filename`
- `START filename`
- `@ filename`
- `EDIT filename`
- `SPOOL filename`

# Команды SAVE, START, и EDIT

---

**LIST**

```
1  SELECT ename, mng, dept
2* FROM emp
```

**SAVE my\_query**

```
Created file my_query
```

**START my\_query**

ENAME	MNG	DEPNO
King		90
Kochhar	100	90
...		

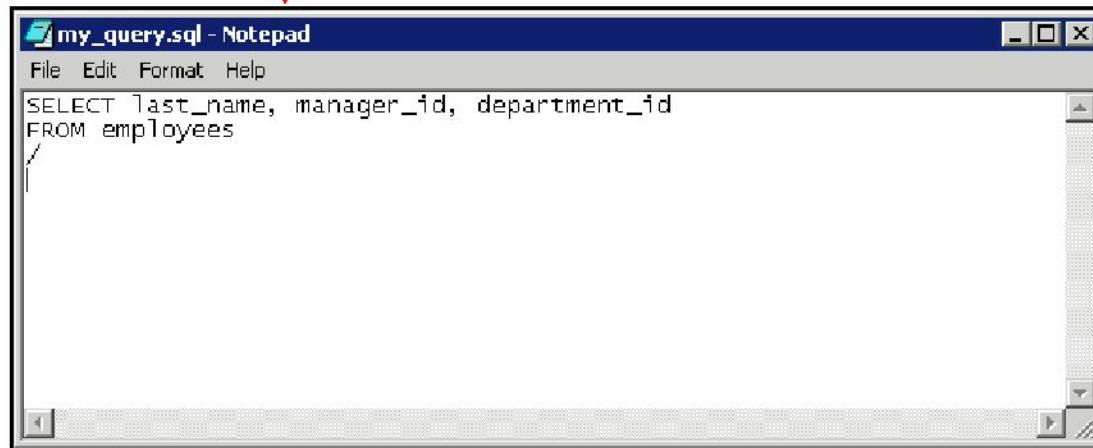
107 rows selected.



# Команды SAVE, START, и EDIT

---

```
EDIT my_query
```



A screenshot of a Notepad window titled "my\_query.sql - Notepad". The window contains the following SQL query:

```
SELECT last_name, manager_id, department_id  
FROM employees
```

A red arrow points from the text "EDIT my\_query" in the box above to the Notepad window.

# Команда SPOOL

---

```
SPO[OL] [file_name[.ext] [CRE[ATE] | REP[LACE] |  
APP[END]] | OFF | OUT]
```

Аргумент	Значение
file_name[.ext]	Запись результатов в файл
CRE[ATE]	Создает новый файл с заданным именем
REP[LACE]	Перезаписывает файл, если он существует. Если не существует – создает и перезаписывает
APP[END]	Добавляет вывод в конец файла
OFF	Прекращает вывод в файл
OUT	Прекращает вывод и пытается распечатать на принтере

# Запуск SQL\*plus

---

1. `rem` Задание параметров экрана для вывода информации в консольном режиме
2. `SET LINESIZE 200`
3. `SET PAGESIZE 40`
4. `SET NULL (null)`
5. `rem` Для корректного отображения кириллицы
6. `@set nls_lang=RUSSIAN_RUSSIA.ru8pc866`
7. `rem` Запуск xe-экземпляра SQL\*Plus \*/
8. `sqlplus student/student@xe`

# Вывод результатов в формате html

---

```
1.  -- =====
2.  -- htmldemoscript.sql
3.  -- =====
4.  SET ECHO off TERMOUT OFF
5.  set markup html on spool on -
6.  preformat off entmap on -
7.  head "<title>HTML Demo Report</title> -
8.  <link rel='stylesheet' href='x.css'>"

9.  spool htmdemo.htm replace

10. select empno, ename, init, msal
11. from employees
12. where deptno = 20;

13. spool off
14. set markup html off
15. set echo on
```

# Определение псевдонима столбца

---

- ▶ Переименовывает заголовок столбца
- ▶ Полезно при вычислениях
- ▶ Следует сразу за именем колонки; ключевое слово **AS** опционально и может находиться между именем колонки и псевдонимом
- ▶ Требуется заключения в двойные кавычки, если содержит пробелы, специальные символы или чувствителен к регистру

## Использование псевдонимов столбцов

```
SQL>SELECT ename AS name,sal salary
2 FROM emp;
```

```
NAME SALARY
-----
...
```

```
SQL> SELECT ename "Name",
2 sal*12 "Annual Salary"
3 FROM emp;
```

```
Name Annual Salary
-----
...
```

# Дублирующиеся строка

---

По умолчанию выводятся все строки, полученные в результате выборки, включая дублирующиеся

```
SQL> SELECT deptno  
2 FROM emp;
```

```
DEPTNO  
-----  
10  
30  
10  
20  
  
...  
14 rows selected.
```

# Исключение дублирующихся строк

---

- ▶ Исключение дубликатов производится путем включения ключевого слова **DISTINCT**

```
SQL> SELECT DISTINCT deptno  
2 FROM emp ;
```

DEPTNO
10
20
30



# DUAL

---

- ▶ DUAL – виртуальная таблица с одним столбцом типа `varchar2(1)` и одной строкой.
- ▶ Использование:

```
SQL> SELECT 5 FROM DUAL
```

```
SQL> SELECT 5+5, 'text' FROM DUAL
```

# Арифметические выражения

---

Записывайте выражения для ЧИСЛОВЫХ и ТЕМПОРАЛЬНЫХ данных с использованием арифметических операторов.

Оператор	Действие
+	Сложение
-	Вычитание
*	Умножение
/	Деление

# Использование арифметических операторов

---

```
SQL> SELECT ename, sal, sal-sal*0.25  
2 FROM emp;
```

ENAME	SAL	SAL-SAL*0.25
KING	5000	5300
BLAKE	2850	3150
CLARK	2450	2750
JONES	2975	3275
MARTIN	1250	1550
ALLEN	1600	1900
...		

14 rows selected.

# Приоритеты операторов

---



- ▶ Умножение и деление имеют более высокий приоритет, чем сложение и вычитание.
- ▶ Операторы с одинаковым приоритетом вычисляются слева направо.
- ▶ Скобки используются для указания приоритетов вычисления и повышения читаемости выражений

# Приоритет операторов

```
SQL> SELECT ename, sal, 12*sal+100  
2 FROM emp;
```

ENAME	SAL	12*SAL+100
KING	5000	60100
BLAKE	2850	34300
CLARK	2450	29500
JONES	2975	35800
MARTIN	1250	15100
ALLEN	1600	19300
...		

14 rows selected.

## Использование скобок

```
SQL> SELECT ename, sal, 12*(sal+100)
      2 FROM emp;
```

ENAME	SAL	12*(SAL+100)
-----	-----	-----
KING	5000	61200
BLAKE	2850	35400
CLARK	2450	30600
JONES	2975	36900
MARTIN	1250	16200
...		

14 rows selected.

# Определение значения Null

- ▶ Null – это значение, которое неизвестно, не определено или неприменимо.
- ▶ Null – это не то же самое, что ноль или пробел
- ▶ **У Oracle нет пустых строк – вместо них автоматически вставляется null**

```
SQL> SELECT  ename, job, comm
2 FROM      emp;
```

ENAME	JOB	COMM
KING	PRESIDENT	
BLAKE	MANAGER	
...		
TURNER	SALESMAN	0
...		

14 rows selected.

# NULL в логических выражениях

---

Условие	Значение A	Результат
a IS NULL	10	FALSE
a IS NOT NULL	10	TRUE
a IS NULL	NULL	TRUE
a IS NOT NULL	NULL	FALSE
a = NULL	10	UNKNOWN
a != NULL	10	UNKNOWN
a = NULL	NULL	UNKNOWN
a != NULL	NULL	UNKNOWN
a = 10	NULL	UNKNOWN
a != 10	NULL	UNKNOWN



# Null в арифметических выражениях

---

Значением выражений, содержащих null, является null.

```
SQL> select ename NAME, 12*sal+comm  
2 from emp;
```

NAME	12*SAL+COMM
-----	-----
KING	

# Оператор конкатенации

---

- ▶ Выполняет конкатенацию значений в столбцах или строк к значениям других столбцов
- ▶ Записывается двумя вертикальными чертами (||)
- ▶ Создает результирующий столбец, значения которого являются символьными выражениями

# Использование оператора конкатенации

---

```
SQL> SELECT      ename||job AS "Employees"  
      2 FROM      emp;
```

```
Employees  
-----  
KINGPRESIDENT  
BLAKEMANAGER  
CLARKMANAGER  
JONESMANAGER  
MARTINSALESMAN  
ALLENSALESMAN  
  
...  
14 rows selected.
```

# Строковые литералы

---

- ▶ Литерал – это символ, строка или число, включенные в список `SELECT`.
- ▶ Темпоральные и символьные данные должны быть заключены в одинарные кавычки.
- ▶ Каждая строка выводится один раз для каждой возвращаемой строки таблицы.

# Использование литералов

---

```
SQL> SELECT ename || ' ' || 'is a' || ' ' || job
2          AS "Employee Details"
3 FROM    emp;
```

```
Employee Details
-----
KING is a PRESIDENT
BLAKE is a MANAGER
CLARK is a MANAGER
JONES is a MANAGER
MARTIN is a SALESMAN
...
14 rows selected.
```

# Вопрос

---

```
SELECT 'O'Reilly' FROM DUAL
```

```
SELECT 'O'Reilly' FROM DUAL
```

Q или q (и), [и], {и}, <и>, Любой одиночный символ

```
SELECT q'!O'Reilly!' FROM DUAL
```

```
SELECT q'(O'Reilly)' FROM DUAL
```

```
SELECT Q'[O'Reilly]' FROM DUAL
```