

Бази даних та інформаційні системи

Лекція 5. Прості запити до бази даних

СумДУ, каф. КН
2020

Таблиці, що використовуються в курсі

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
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7902	WATSON	CLERK	7698	03-DEC-81	950		30
7566	SCOTT	ANALYST	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	DEAN	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

Оператор SELECT

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

- **SELECT** визначає *які* стовпці;
- **FROM** визначає з *якої* таблиці.

Створення запитів мовою SQL

- ▶ Вирази SQL не чутливі до регістру.
- ▶ Вирази SQL можуть складатися з одного або декількох рядків.
- ▶ Ключові слова не можуть переноситися та скорочуватися.
- ▶ Табуляції та відступи використовуються для зручності при читанні.

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

Вибірка всіх стовпців

```
SELECT *  
FROM dept;
```

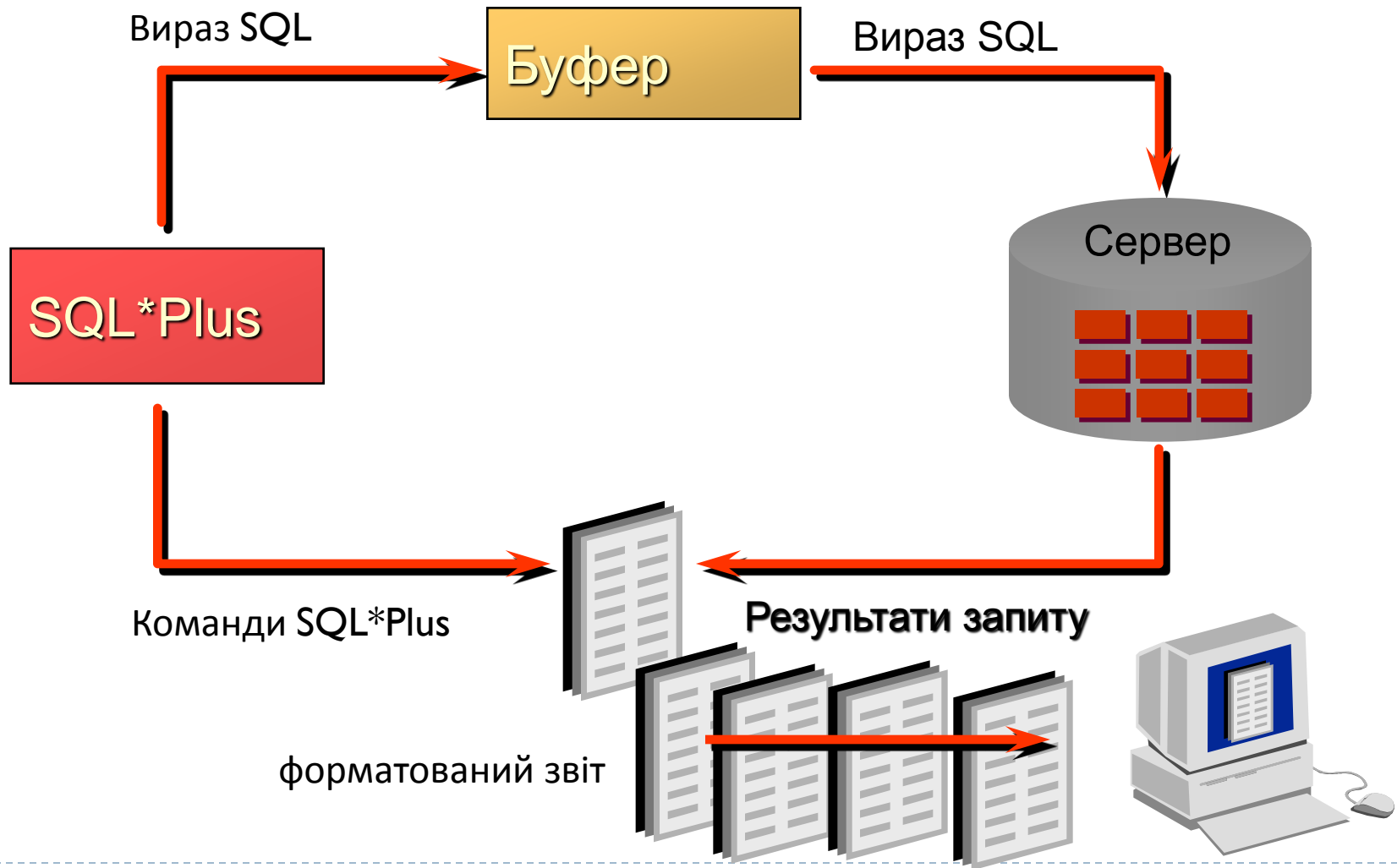
DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Вибірка конкретних стовпців

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

DEPTNO	LOC
10	NEW YORK
20	DALLAS
30	CHICAGO
40	BOSTON

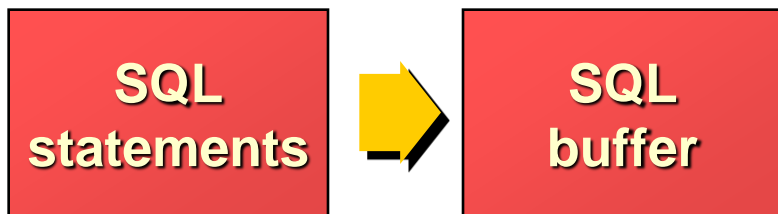
Взаємодія SQL та SQL*Plus



Порівняння виразів SQL і команд SQL * Plus

SQL

- Мова
- Стандарт ANSI
- Ключові слова не можуть бути скорочені
- Вирази взаємодіють з даними і таблицями в БД



SQL*Plus

- середа
- Розробка Oracle
- Ключові слова можуть бути скорочені
- Команди не дозволяють маніпулювати даними БД



Огляд SQL*Plus

- ▶ Вхід в SQL*Plus.
- ▶ Перегляд структури таблиць.
- ▶ Редагування виразів SQL.
- ▶ Виконання виразів SQL з SQL * Plus.
- ▶ Запис і додавання виразів SQL в файли.
- ▶ Запуск збережених файлів.
- ▶ Завантаження з файлів в буфер з можливістю редагування.
- ▶ Додаткове читання:
<http://citforum.ru/database/oracle/sqlplus>

Вхід SQL*Plus

У командному рядку:

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

У командному рядку: SQL*Plus

Команда	Дія
A[PPEND] <i>текст</i>	Додати текст в кінець поточного рядка
C[HANGE] <i>/старый/новый/</i>	Замінити в поточному рядку <i>старий</i> текст на <i>новий</i> .
C[HANGE] <i>/текст/</i>	Видалити вказаний <i>текст</i>
CL[EAR] BUFF[ER]	Видалити всі рядки з буферу SQL
DEL	Видалити поточний рядок
DEL <i>n</i>	Видалити рядок, який заданий параметром <i>n</i>
DEL <i>m n</i>	Видалити рядки від <i>m</i> до <i>n</i>

Команди редагування SQL*Plus

Команда	Опис
I[INPUT]	Вставити невизначену кількість рядків
I[INPUT] <i>text</i>	Вставити рядок, що складається з тексту
L[IST]	Вивести список всіх рядків з буфера SQL
L[IST] <i>n</i>	Вивести один рядок (з номером <i>n</i>)
L[IST] <i>m n</i>	Вивести діапазон рядків від <i>n</i> до <i>m</i>
R[UN]	Вивести і виконати команду з буфера SQL
/	Синонім RUN
<i>n</i>	Вказати рядок, який повинен стати поточним
<i>n text</i>	Замінити рядок <i>n</i> текстом
0 <i>text</i>	Вставити рядок перед рядком 1

Використання LIST, n, 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
```

Файлові команди SQL*Plus

Команда	Опис
SAV[E] <i>ім'я_файлу</i> [.ext] [REP[LACE] APP[END]]	Зберігає поточний зміст буферу SQL у файлі. APPEND використовується для додавання інформації в наявний файл. REPLACE перезаписує наявний файл. За замовчуванням файл має розширення .sql.
GET <i>ім'я_файлу</i> [.ext]	Викликає вміст збереженого раніше файлу в буфер SQL. За замовчуванням файл має розширення .sql.
START <i>filename</i>	Запускає виконання збереженого раніше файлу команд.
@ <i>filename</i>	Синонім START
EDIT	Викликає редактор і зберігає вміст буфера в файлі <i>afiedt.buf</i> .
EDIT <i>filename</i>	Викликає редактор для редагування збереженого файлу.
SPOOL <i>filename</i>	Записує результати запиту в файл. OFF закриває буферний файл (спула-файл). OUT закриває буферний файл і надсилає результати з файлу на системний принтер.
EXIT	Вихід з SQL * Plus (Синонім QUIT)

Команди SAVE, START, EDIT

LIST

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

SAVE my_query

Created file my_query

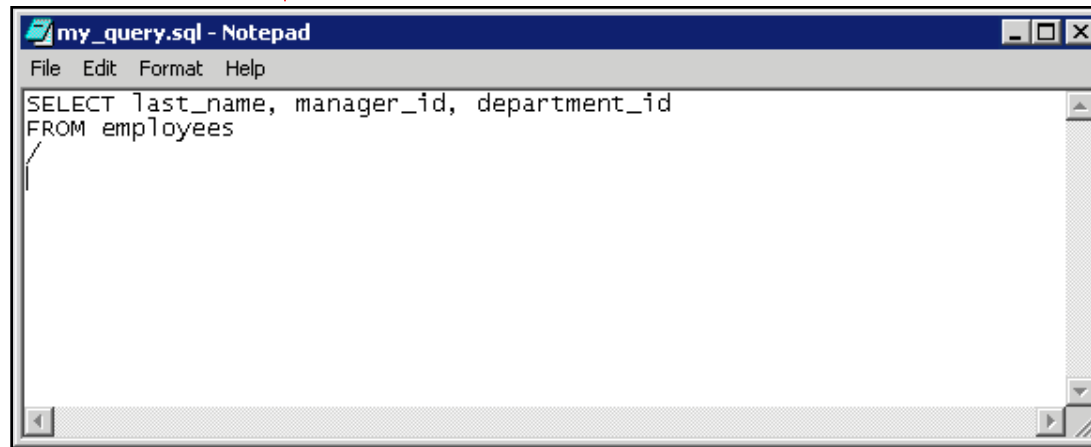
START my_query

ENAME	MNG	DEPNO
King		90
Kochhar	100	90
...		

107 rows selected.

Команди SAVE, START, EDIT

EDIT my_query



Команда SPOOL

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

Аргумент	Значення
file_name[.ext]	Запис результатів у файл
CRE[ATE]	Створює новий файл із заданим ім'ям
REP[LACE]	Перезаписує файл, якщо він існує. Якщо не існує - створює і перезаписує
APP[END]	Додає виведенні дані в кінець файлу
OFF	Припиняє виведення в файл
OUT	Припиняє виведення і намагається роздрукувати на принтері

Запуск SQL*plus

1. `rem` - задання параметрів екрану для виведення інформації в консольному режимі.
2. `ECHO SET LINESIZE 200 > startup.sql`
3. `ECHO SET PAGESIZE 40 >> startup.sql`
4. `ECHO SET NULL (null) >> startup.sql`
5. `@set nls_lang=RUSSIAN_RUSSIA.ru8pc866`
6. `rem запуск xe-екземпляру SQL*Plus */`
7. `sqlplus student/student@xe @startup`

Виведення результатів у форматі html

```
1.  -- =====
2.  -- htmldemo.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 htmldemo.htm replace

10. select empno, ename, sal from empl;

11. spool off
12. set markup html off
13. set echo on
```

Визначення псевдоніма стовпця

Псевдонім:

- ▶ Перейменовує заголовок стовпця;
- ▶ Корисний при обчисленнях;
- ▶ Слідуює відразу за ім'ям колонки;
Ключове слово **AS** – це опція, яка може перебувати між ім'ям колонки і псевдонімом;
- ▶ Вимагає укладення в подвійні лапки, якщо містить пробіли, спеціальні символи або чутливий до регістру.

Використання псевдонімів стовпців

```
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

...	

Рядки, що дублюються

За замовчуванням виводяться всі рядки, отримані в результаті вибірки, включно з тими, що дублюються.

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

```
DEPTNO  
-----  
10  
30  
10  
20
```

...

```
14 rows selected.
```

Виключення рядків, які дублюються

- ▶ Виключення дублікатів здійснюється за допомогою ключового слова **DISTINCT**.

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

DEPTNO
10
20
30

DUAL

- ▶ **DUAL** - віртуальна таблиця з одним стовпцем типу `varchar2 (1)` і одним рядком.
- ▶ Використання:

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

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

```
SQL> SELECT user FROM DUAL;
```

Звернення до каталогу даних

Опис таблиць користувача:

```
SQL> SELECT *  
2      FROM user_tables;
```

Перегляд типів даних об'єктів користувача:

```
SQL> SELECT DISTINCT object_type  
2      FROM user_objects;
```

Перегляд таблиць, представлень, синонімів і послідовностей:

```
SQL> SELECT *  
2      FROM user_catalog;
```



Додавання коментарів

- ▶ Ви можете додати коментар до таблиці або стовпця за допомогою оператора COMMENT:

```
SQL> COMMENT ON TABLE emp  
2 IS 'Employee Information';  
Comment created.
```

- ▶ Коментарі можуть бути переглянуті за допомогою каталогу даних:
 - ▶ ALL_COL_COMMENTS;
 - ▶ USER_COL_COMMENTS;
 - ▶ ALL_TAB_COMMENTS;
 - ▶ USER_TAB_COMMENTS.



Арифметичні вирази

Записуйте вирази для *числових* і *темпоральних* даних з використанням арифметичних операторів.

Оператор	Дія
+	Додавання
-	Віднімання
*	Добуток
/	Ділення

Використання арифметичних операторів

```
SQL> SELECT  ename, sal, sal+300  
2 FROM      emp;
```

ENAME	SAL	SAL+300
KING	5000	5300
BLAKE	2850	3150
CLARK	2450	2750
JONES	2975	3275
MARTIN	1250	1550
ALLEN	1600	1900

...

14 rows selected.

Пріоритети операторів



- ▶ Добуток і ділення мають більш високий пріоритет, ніж додавання і віднімання.
- ▶ Оператори з однаковим пріоритетом обчислюються зліва направо.
- ▶ Дужки використовуються для зазначення пріоритетів обчислення і для зручності читання.

Пріоритети операторів

```
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.

Використання дужок

```
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.

Значення Null

- ▶ **Null** - це значення, яке невідомо, чи не визначено або не можна застосувати.
- ▶ **Null** - це не те ж саме, що нуль або пробіл.
- ▶ У Oracle немає порожніх рядків - замість них автоматично вставляється **null**.

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

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

14 rows selected.

NULL в логічних виразах

Умова	Значення 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 арифметичних виразах

Значенням виразів, що містять `null`, є `null`.

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

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

Оператор конкатенації

- ▶ Конкатенація використовується тоді, коли потрібно з'єднати значення декількох стовпців або приєднати до значень стовпців текст;
- ▶ Записується двома вертикальними рисами (| |);
- ▶ Оператор конкатенації можна використовувати для строкових значень , для чисел і дат . Ці значення будуть автоматично конвертовані в строкові.
- ▶ Пробіли при конкатенації зберігаються.

Використання оператора конкатенації

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

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

Строкові літерали

- ▶ Літерал - це символ, рядок або число, що включені в список `SELECT`.
- ▶ Темпоральні і символльні дані повинні бути укладені в одинарні лапки.
- ▶ Кожен рядок виводиться один раз для кожного рядка таблиці, повертається .

Використання літералів

```
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.
```

Квотування символів

~~SELECT 'O'Reilly' FROM DUAL~~

SELECT 'O''Reilly' FROM DUAL

Використовуйте альтернативне квотування, яке починається з символу **Q** або **q**.

Обмежувачем може бути будь-який символ відмінний від пробілу.

(i), [i], { i }, < i >, будь-який одиночний символ:

SELECT q'!O'Reilly!' FROM DUAL

SELECT q'(O'Reilly)' FROM DUAL

SELECT Q'[O'Reilly]' FROM DUAL