



# Бази даних та інформаційні системи

## Тема 18. Ієрархічні запити.

СУМДУ, каф. КН  
2020

# Зміст

---

- ▶ Після завершення заняття ви повинні вміти і знати наступне:
  - ▶ Як створювати ієрархічні запити
  - ▶ Як використовувати псевдостовпці **LEVEL**, **PRIOR**

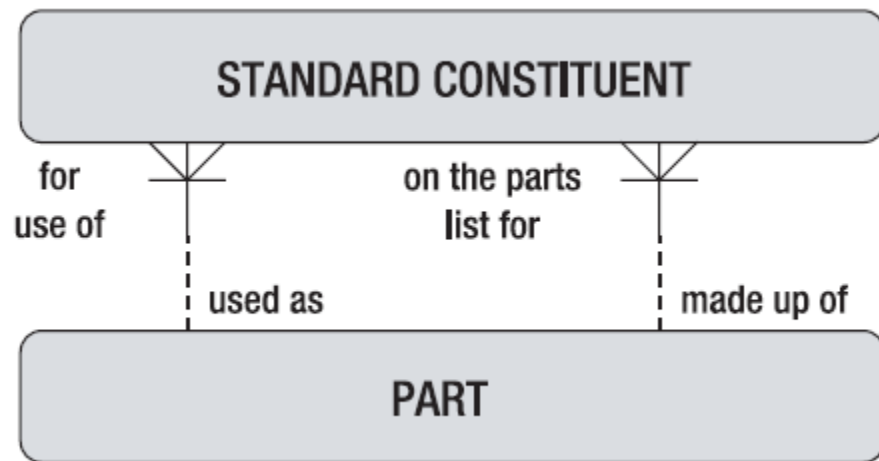
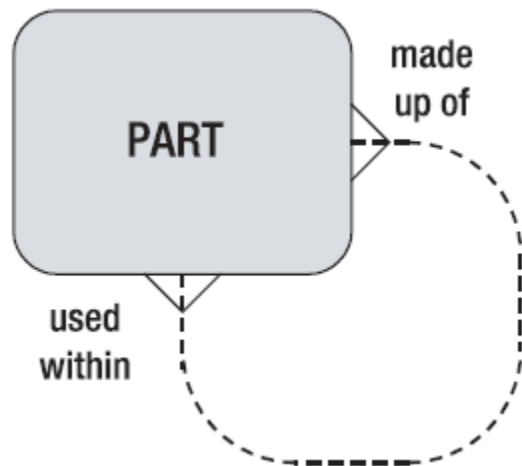
# Ієрархічні (рекурсивні) запити

---

```
ENAME
-----
KING
      JONES
            SCOTT
            ADAMS
      FORD
            SMITH
      BLAKE
            ALLEN
            WARD
            MARTIN
            TURNER
            JONES
      CLARK
            MILLER
14 rows selected.
```

# Ієрархічні запити

---



# Простий ієрархічний запит

---

```
SELECT select_list
FROM table_expression
[ WHERE ... ]
[ START WITH start_expression ]
CONNECT BY
{ PRIOR parent_expr = child_expr
| child_expr = PRIOR parent_expr }
[ ORDER SIBLINGS BY
column1 [ ASC | DESC ] [, column2 [ ASC | DESC ] ] ...
```

- ▶ **CONNECT BY** – визначає **за яким критерієм** з'єднувати записи;
- ▶ **START WITH** – визначає **з якого запису** починати цикл;
- ▶ **PRIOR** – надає доступ до попереднього запису;
- ▶ **SIBLINGS** – дозволяє здійснювати сортування **в межах рівня**.



## Приклад

```
SELECT ename, LEVEL  
FROM emp  
START WITH mgr is null  
CONNECT BY NOCYCLE PRIOR empno = mgr;
```

ENAME	LEVEL
-------	-------

-----	-----
-------	-------

KING	1
------	---

JONES	2
-------	---

SCOTT	3
-------	---

ADAMS	4
-------	---

FORD	3
------	---

SMITH	4
-------	---

...

14 rows selected.

## NO CYCLE

---

- ▶ Ключове слово **NOCYCLE** у фразі **CONNECT BY** запобігає нескінченні цикли в ієрархічних запитах.
- ▶ У нашому прикладі співробітники не утворюють циклів, але обережність не завадить.

## LEVEL, CONNECT\_BY\_ISCYCLE, CONNECT\_BY\_ISLEAF

---

- ▶ **LEVEL** – псевдостовпець містить рівень рядка в ієрархічному запиті по відношенню до кореневого.
- ▶ **CONNECT\_BY\_ISCYCLE** – псевдостовпець містить 1 для всіх рядків, які зв'язні циклічним зв'язком (інакше 0).
- ▶ **CONNECT\_BY\_ISLEAF** – псевдостовпець містить 1 якщо рядок є нижнім елементом ієрархії (листом) інакше - 0.



# Приклад с Level

```
SELECT LPAD(' ', 2*LEVEL-1) ||ename AS ename  
FROM emp  
START WITH mgr IS NULL  
CONNECT BY NOCYCLE PRIOR empno = mgr;
```

```
ENAME  
-----  
KING  
      JONES      SCOTT  
                  ADAMS  
      FORD  
                  SMITH  
      BLAKE  
                  ALLEN  
                  WARD  
                  MARTIN  
                  TURNER  
                  JONES  
      CLARK  
                  MILLER  
14 rows selected.
```

# Приклад с CONNECT\_BY\_ISLEAF

```
SELECT  ename , CONNECT_BY_ISLEAF as IsLeaf
FROM    emp
WHERE    deptno = 10
START WITH  ename='KING'
CONNECT BY PRIOR empno = mgr;
```

ENAME	ISLEAF
KING	0
CLARK	0
MILLER	1

# CONNECT\_BY\_ROOT, SYS\_CONNECT\_BY\_PATH

---

- ▶ **CONNECT\_BY\_ROOT** – оператор дозволяє зіставити рядку (незалежно від його рівня) кореневий елемент ієрархії.
- ▶ **SYS\_CONNECT\_BY\_PATH** – функція дозволяє отримати повний шлях від кореня до поточного елементу.

# Приклади

```
SELECT  ename,  
CONNECT_BY_ROOT ename as manager,  
SYS_CONNECT_BY_PATH(ename, ' > ') AS full_path  
FROM emp  
START WITH job = 'MANAGER'  
CONNECT BY PRIOR empno = mgr;
```

ENAME	MANAGER	FULL_PATH
-----	-----	-----
JONES	JONES	> JONES
SCOTT	JONES	> JONES > SCOTT
ADAMS	JONES	> JONES > SCOTT > ADAMS
FORD	JONES	> JONES > FORD
...		
MILLER	CLARK	> CLARK > MILLER

13 rows selected.

# Сортування результатів ієрархічних запитів

```
select ename
, sys_connect_by_path(ename, '|')
as path
from emp
start with mgr is null
connect by prior empno = mgr
order SIBLINGS by ename;
ENAME  PATH
```

```
-----
KING    |KING
BLAKE   |KING|BLAKE
ALLEN   |KING|BLAKE|ALLEN
JONES   |KING|BLAKE|JONES
MARTIN  |KING|BLAKE|MARTIN
TURNER  |KING|BLAKE|TURNER
```

...

14 rows selected.

```
select ename
, sys_connect_by_path(ename, '|')
as path
from emp
start with mgr is null
connect by prior empno = mgr
order by ename;
ENAME PATH
```

```
-----
ADAMS   |KING|JONES|SCOTT|ADAMS
ALLEN   |KING|BLAKE|ALLEN
BLAKE   |KING|BLAKE
CLARK   |KING|CLARK
FORD    |KING|JONES|FORD
JONES   |KING|JONES
```

...

14 rows selected

Опція SIBLINGS не руйнує ієрархію

# Висновки

---

- ▶ Ієрархічні запити слід використовувати якщо:
  - ▶ необхідно відобразити ієрархію об'єктів
  - ▶ здійснити аналіз даних у батьків або нащадків певного об'єкта
- ▶ Додаткові матеріали:
- ▶ <http://habrahabr.ru/blogs/sql/43955/>
- ▶ <http://citforum.ru/database/oracle/recursive/>

