MYONGJI UNIVERSITY

# 고급 SQL: Recursion

# WITH Clause

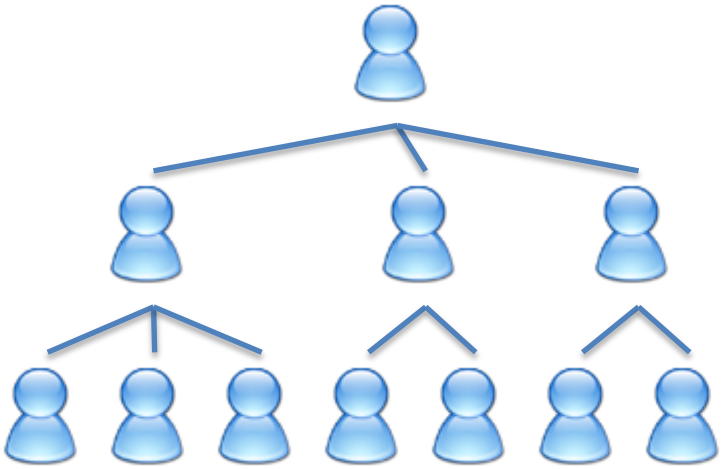
- 복잡한 Subquery를 단순화

```
SELECT e.ename AS employee_name,
       dc.dept_count AS emp_dept_count
FROM   emp e,(SELECT deptno, COUNT(*) AS dept_count
        FROM   emp GROUP BY deptno) dc
WHERE  e.deptno = dc.deptno;
```

```
WITH dept_count AS (
  SELECT deptno, COUNT(*) AS dept_count FROM   emp
  GROUP BY deptno)
SELECT e.ename AS employee_name,
       dc.dept_count AS emp_dept_count
FROM   emp e, dept_count dc
WHERE  e.deptno = dc.deptno;
```

# Recursive Query

- 트리/그래프 형태의 자료를 테이블로 저장?



| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|---|---|---|---|---|---|---|
| 7369 | SMITH | CLERK | 7902 | 80/12/17 | 800 | | 20 |
| 7499 | ALLEN | SALESMAN | 7698 | 81/02/20 | 1600 | 300 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 81/02/22 | 1250 | 500 | 30 |
| 7566 | JONES | MANAGER | 7839 | 81/04/02 | 2975 | | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 81/09/28 | 1250 | 1400 | 30 |
| 7698 | BLAKE | MANAGER | 7839 | 81/05/01 | 2850 | | 30 |
| 7782 | CLARK | MANAGER | 7839 | 81/06/09 | 2450 | | 10 |
| 7788 | SCOTT | ANALYST | 7566 | 87/04/19 | 3000 | | 20 |
| 7839 | KING | PRESIDENT | | 81/11/17 | 5000 | | 10 |
| 7844 | TURNER | SALESMAN | 7698 | 81/09/08 | 1500 | 0 | 30 |
| 7876 | ADAMS | CLERK | 7788 | 87/05/23 | 1100 | | 20 |
| 7900 | JAMES | CLERK | 7698 | 81/12/03 | 950 | | 30 |
| 7902 | FORD | ANALYST | 7566 | 81/12/03 | 3000 | | 20 |
| 7934 | MILLER | CLERK | 7782 | 82/01/23 | 1300 | | 10 |

- 트리 형태의 질의 처리?
  - "A"의 모든 부하 직원을 출력하라.
  - Boss 부터 각 직원들을 차례로 출력하라.

# WITH (ANSI 표준)

- Recursive Subquery를 활용하는 법

```
WITH emp_mgr(empno, ename, mgr, lvl) AS
( SELECT empno, ename, mgr, 1
  FROM emp WHERE mgr IS NULL
  UNION ALL
  SELECT e.empno, e.ename, e.mgr, m.lvl + 1
  FROM emp e JOIN emp_mgr m ON e.mgr = m.empno
)
SELECT * FROM emp_mgr;
```

# Hierarchical Query (ORACLE)

- 트리 형태 구조를 추출하기 위한 질의
- START WITH (ROOT조건), CONNECT BY PRIOR (연결조건)
- LEVEL: 트리의 레벨을 나타내는 Pseudo Column

```
SELECT empno, ename, mgr, level
FROM emp
START WITH mgr IS NULL
CONNECT BY PRIOR empno = mgr;
```

# Examples: Subquery Factoring

```
WITH dept_costs AS (
      SELECT department_name, SUM(salary) dept_total
         FROM employees e, departments d
         WHERE e.department_id = d.department_id
      GROUP BY department_name),
     avg_cost AS (
      SELECT SUM(dept_total)/COUNT(*) avg
      FROM dept_costs)
SELECT * FROM dept_costs
   WHERE dept_total >
       (SELECT avg FROM avg_cost)
       ORDER BY department_name;

DEPARTMENT_NAME                DEPT_TOTAL
------------------------------ ----------
Sales                              304500
Shipping                           156400
```

# Examples: Subquery Factoring

```
WITH reports_to_101 (eid, emp_last, mgr_id, reportLevel) AS
  (  SELECT employee_id, last_name, manager_id, 0 reportLevel
     FROM employees
     WHERE employee_id = 101
   UNION ALL
     SELECT e.employee_id, e.last_name, e.manager_id, reportLevel+1
     FROM reports_to_101 r, employees e
     WHERE r.eid = e.manager_id
  )
SELECT eid, emp_last, mgr_id, reportLevel
FROM reports_to_101 ORDER BY reportLevel, eid;

       EID EMP_LAST                      MGR_ID REPORTLEVEL
---------- ------------------------- ---------- -----------
       101 Kochhar                          100           0
       108 Greenberg                        101           1
       200 Whalen                           101           1
       203 Mavris                           101           1
       204 Baer                             101           1
       205 Higgins                          101           1
       109 Faviet                           108           2
       110 Chen                             108           2
       111 Sciarra                          108           2
       112 Urman                            108           2
       113 Popp                             108           2
       206 Gietz                            205           2
```

# Examples: Subquery Factoring

```
WITH reports_to_101 (eid, emp_last, mgr_id, reportLevel, mgr_list) AS (
     SELECT employee_id, last_name, manager_id, 0 reportLevel,
            CAST(manager_id AS VARCHAR2(2000))
     FROM employees WHERE employee_id = 101
  UNION ALL
     SELECT e.employee_id, e.last_name, e.manager_id, reportLevel+1,
            CAST(mgr_list || ',' || manager_id AS VARCHAR2(2000))
     FROM reports_to_101 r, employees e WHERE r.eid = e.manager_id
)
SELECT eid, emp_last, mgr_id, reportLevel, mgr_list
FROM reports_to_101 ORDER BY reportLevel, eid;
```

| EID | EMP_LAST | MGR_ID | REPORTLEVEL | MGR_LIST |
|---|---|---|---|---|
| 101 | Kochhar | 100 | 0 | 100 |
| 108 | Greenberg | 101 | 1 | 100,101 |
| 200 | Whalen | 101 | 1 | 100,101 |
| 203 | Mavris | 101 | 1 | 100,101 |
| 204 | Baer | 101 | 1 | 100,101 |
| 205 | Higgins | 101 | 1 | 100,101 |
| 109 | Faviet | 108 | 2 | 100,101,108 |
| 110 | Chen | 108 | 2 | 100,101,108 |
| 111 | Sciarra | 108 | 2 | 100,101,108 |
| 112 | Urman | 108 | 2 | 100,101,108 |
| 113 | Popp | 108 | 2 | 100,101,108 |
| 206 | Gietz | 205 | 2 | 100,101,2052 |

# Examples: 조직도

```
WITH
  org_chart (eid, emp_last, mgr_id, reportLevel, salary, job_id)
AS
  (
    SELECT employee_id, last_name, manager_id,
           0 reportLevel, salary, job_id
    FROM employees
    WHERE manager_id is null
  UNION ALL
    SELECT e.employee_id, e.last_name, e.manager_id,
           r.reportLevel+1 reportLevel, e.salary, e.job_id
    FROM org_chart r, employees e
    WHERE r.eid = e.manager_id
  )
  SEARCH DEPTH FIRST BY emp_last SET order1
SELECT lpad(' ',2*reportLevel)||emp_last emp_name, eid, mgr_id,
       salary, job_id
FROM org_chart
ORDER BY order1;
```

# Examples: 조직도 (결과)

| EMP_NAME | EID | MGR_ID | SALARY | JOB_ID |
|---|---|---|---|---|
| King | 100 | | 24000 | AD_PRES |
| Cambrault | 148 | 100 | 11000 | SA_MAN |
| Bates | 172 | 148 | 7300 | SA_REP |
| Bloom | 169 | 148 | 10000 | SA_REP |
| Fox | 170 | 148 | 9600 | SA_REP |
| Kumar | 173 | 148 | 6100 | SA_REP |
| Ozer | 168 | 148 | 11500 | SA_REP |
| Smith | 171 | 148 | 7400 | SA_REP |
| De Haan | 102 | 100 | 17000 | AD_VP |
| Hunold | 103 | 102 | 9000 | IT_PROG |
| Austin | 105 | 103 | 4800 | IT_PROG |
| Ernst | 104 | 103 | 6000 | IT_PROG |
| Lorentz | 107 | 103 | 4200 | IT_PROG |
| Pataballa | 106 | 103 | 4800 | IT_PROG |
| Errazuriz | 147 | 100 | 12000 | SA_MAN |
| Ande | 166 | 147 | 6400 | SA_REP |

. . .

# Examples: Subquery Factoring

```
WITH emp_count (eid, emp_last, mgr_id, mgrLevel, salary, cnt_employees) AS
  (
    SELECT employee_id, last_name, manager_id, 0 mgrLevel, salary, 0 cnt_employees
    FROM employees
  UNION ALL
    SELECT e.employee_id, e.last_name, e.manager_id,
           r.mgrLevel+1 mgrLevel, e.salary, 1 cnt_employees
    FROM emp_count r, employees e
    WHERE e.employee_id = r.mgr_id
  )SEARCH DEPTH FIRST BY emp_last SET order1
SELECT emp_last, eid, mgr_id, salary, sum(cnt_employees), max(mgrLevel) mgrLevel
FROM emp_count
GROUP BY emp_last, eid, mgr_id, salary
HAVING max(mgrLevel) > 0
ORDER BY mgr_id NULLS FIRST, emp_last;

EMP_LAST                       EID     MGR_ID     SALARY SUM(CNT_EMPLOYEES)    MGRLEVEL
------------------- ---------- ---------- ---------- ------------------- ----------
King                       100                 24000                 106          3
Cambrault                  148        100      11000                   7          2
De Haan                    102        100      17000                   5          2
Errazuriz                  147        100      12000                   6          1
Fripp                      121        100       8200                   8          1
Hartstein                  201        100      13000                   1          1
Kaufling                   122        100       7900                   8          1
. . .
```