

Ch 5. Advanced SQL Features (in Oracle) (parts of Ch. 24 and Ch. 25)

Sang-Won Lee

<http://icc.skku.ac.kr/~swlee>

SKKU VLDB Lab.

(<http://vldb.skku.ac.kr/>)

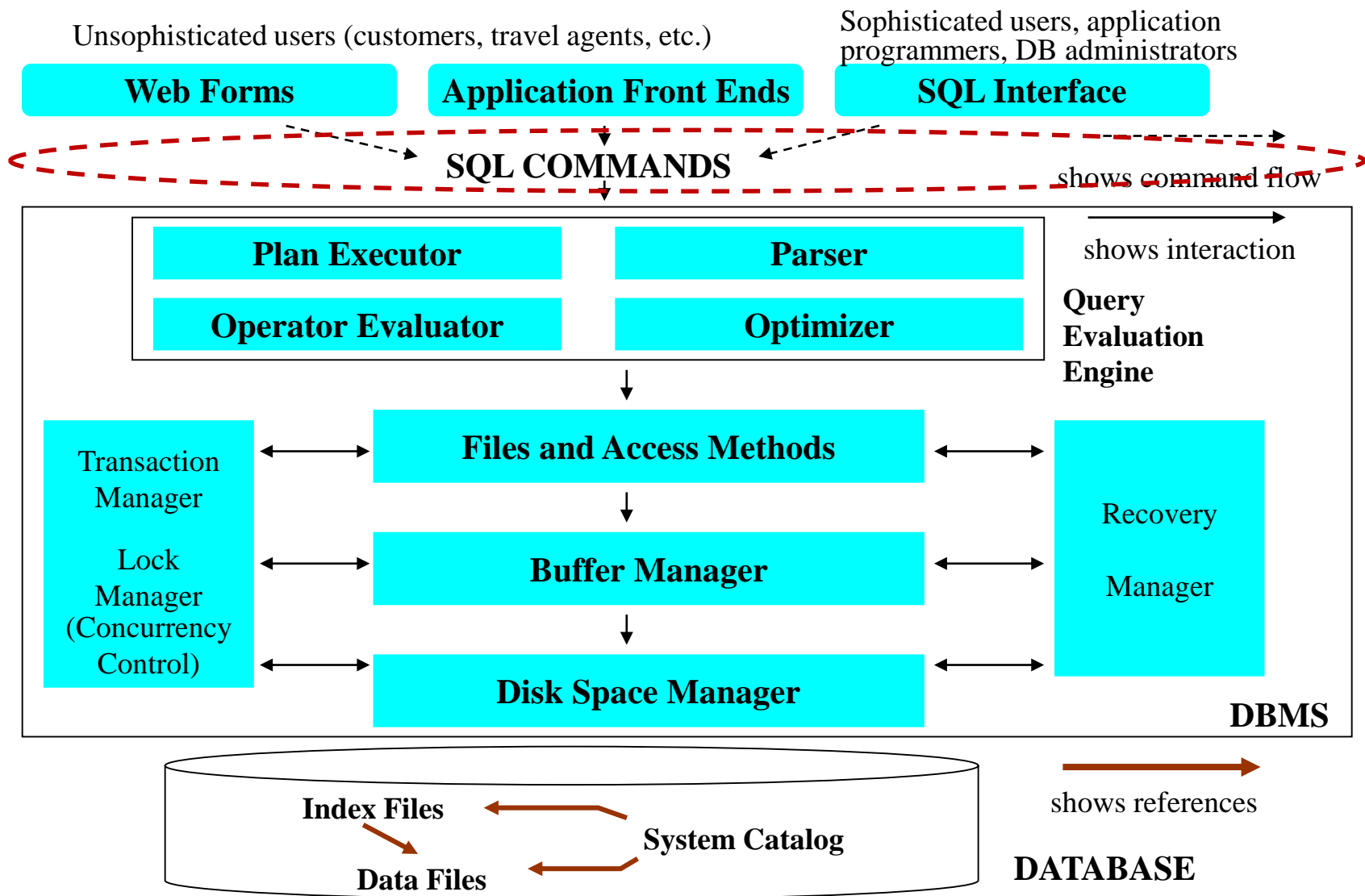


Figure 1.3 Anatomy of an RDBMS

Overview

- Tree traversal (Ch 24.1)
- Pivoting (Ch 25.3)
- Roll-up and Cube in SQL:1999(Ch 25.3.1)
- Window queries in SQL:1999 (Ch 25.4)

SQL Expressive Powers

1. Relational Algebra or Calculus
2. Aggregation / Grouping
3. → Deductive Logics /
Analytic Functions (Windowing)
1. Data Mining Features

Back Ground

- Tree/Graph Data becomes more popular (e.g. XML, Social Network, RDF)
- Data Warehouse, OLAP (vs. OLTP), Data Mining, Big Data (since 1990s)
 - Nothing is more difficult, and therefore more precious, than to be able to decide.
(Napoleon Bonaparte)
- In order to support these new applications using RDBMS, traditional SQL should be extended to be equipped with new weapons for better productivity and higher performance.

1. Relational Algebra or Calculus
2. Aggregation / Grouping
3. **Deductive Logics** / Analytic Functions (Windowing)
4. Data Mining Features

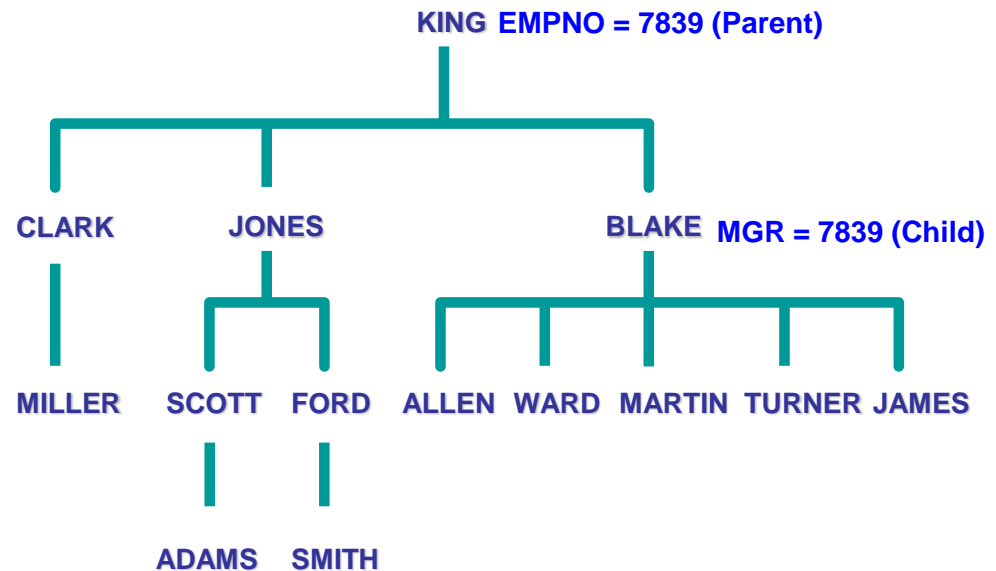
Tree Traversal (in Oracle) (Read Section 24.1 @ Textbook)

1. The concept of a hierarchical query
2. How to generate a tree-structured report
3. *How to format hierarchical data*
4. *Exclude branches from the tree structure*

Table, Tree Structure, and Hierarchical Query

- Relation looks simple, but is general enough to express complex data structure
 - Tree: hierarchical structure, part-of, lineage, ...
 - Graph: map in Car navigation

EMPNO	ENAME	JOB	MGR
7839	KING	PRESIDENT	
7698	BLAKE	MANAGER	7839
7782	CLARK	MANAGER	7839
7566	JONES	MANAGER	7839
7654	MARTIN	SALESMAN	7698
7499	ALLEN	SALESMAN	7698
7844	TURNER	SALESMAN	7698
7900	JAMES	CLERK	7698
7521	WARD	SALESMAN	7698
7902	FORD	ANALYST	7566
7369	SMITH	CLERK	7902
7788	SCOTT	ANALYST	7566
7876	ADAMS	CLERK	7788
7934	MILLER	CLERK	7782



Hierarchical Queries in Oracle

- Queries against hierarchy? Self-join is enough?
- Hierarchical queries in Oracle

```
SELECT [LEVEL], column, expr...  
FROM   table  
[WHERE condition(s)]  
[START WITH condition(s)]  
[CONNECT BY PRIOR condition(s)];
```

Walking the Tree

Starting Point

- Specifies the condition that must be met
- Accepts any valid condition

```
START WITH column1 = value
```

- Using the EMP table, start with employee Blake.

```
...START WITH ename = 'BLAKE'
```


Walking the Tree

Direction

Top down	→	Column1 = Parent Key Column2 = Child Key
Bottom up	→	Column1 = Child Key Column2 = Parent Key

```
CONNECT BY PRIOR column1 = column2
```

- Walk **top down** using the EMP table.

```
... CONNECT BY PRIOR empno = mgr
```

Walking the Tree

```
SQL> SELECT empno, ename, job, mgr
2  FROM emp
3  START WITH empno = 7698
4  CONNECT BY PRIOR mgr = empno;
```

Bottom-up

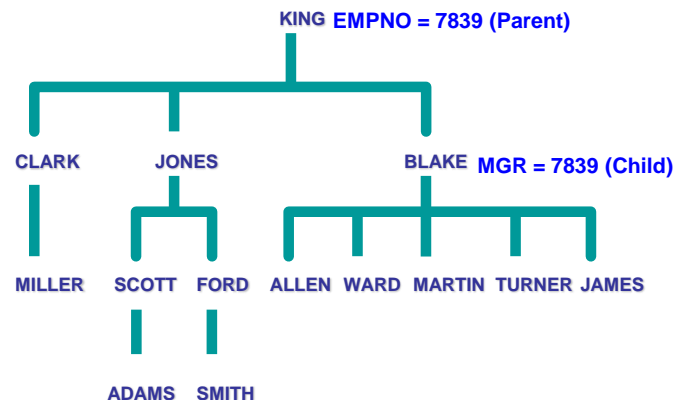
EMPNO	ENAME	JOB	MGR
7698	BLAKE	MANAGER	7839
7839	KING	PRESIDENT	

```
SQL> SELECT empno, ename, job, mgr
2  FROM emp
3  START WITH empno = 7698
4  CONNECT BY empno = PRIOR mgr;
```

Walking the Tree

```
SQL> SELECT ename || ' reports to ' || PRIOR ename "Walk"
2   FROM   emp
3   START WITH ename = 'KING'
4   CONNECT BY PRIOR empno = mgr;
```

EMPNO	ENAME	JOB	MGR
7839	KING	PRESIDENT	
7698	BLAKE	MANAGER	7839
7782	CLARK	MANAGER	7839
7566	JONES	MANAGER	7839
7654	MARTIN	SALESMAN	7698
7499	ALLEN	SALESMAN	7698
7844	TURNER	SALESMAN	7698
7900	JAMES	CLERK	7698
7521	WARD	SALESMAN	7698
7902	FORD	ANALYST	7566
7369	SMITH	CLERK	7902
7788	SCOTT	ANALYST	7566
7876	ADAMS	CLERK	7788
7934	MILLER	CLERK	7782

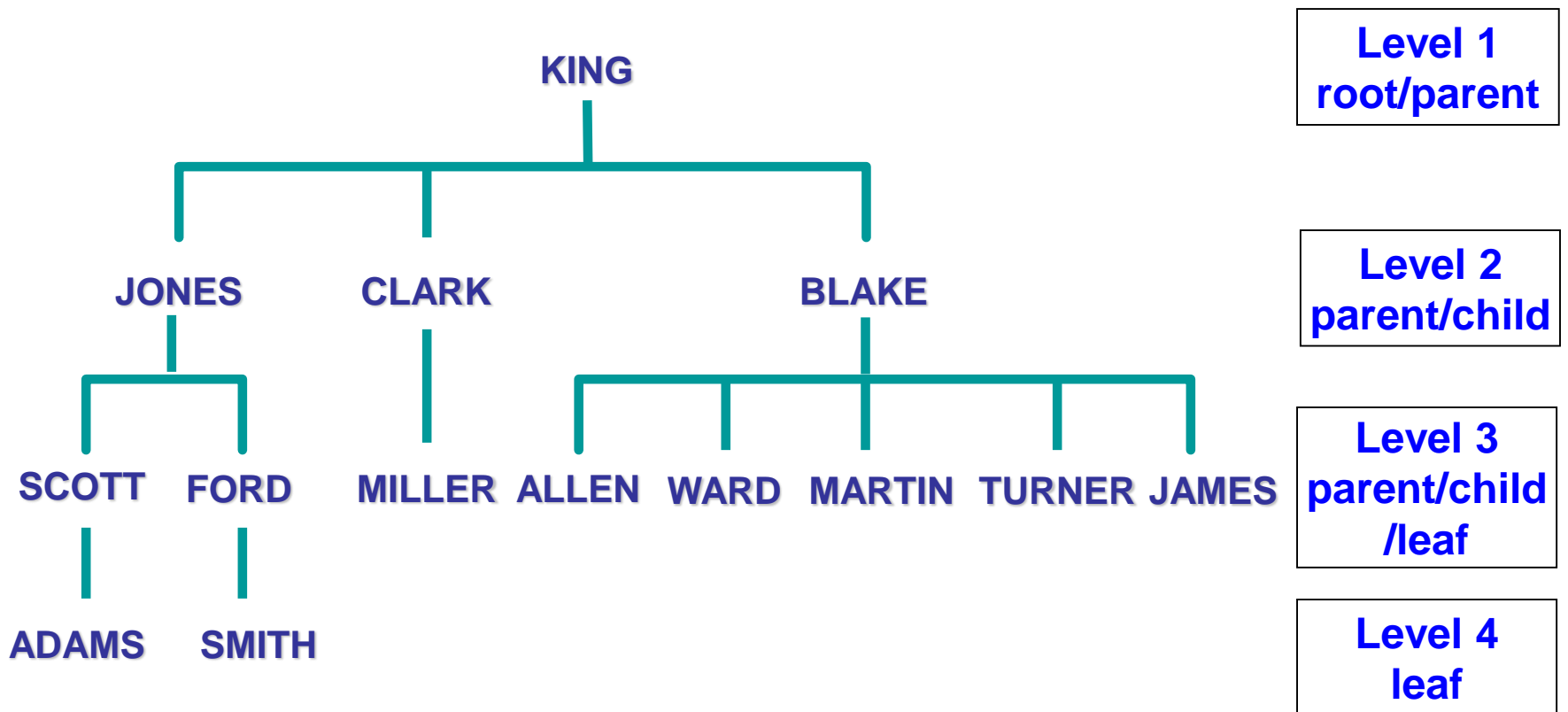


Walk

```

-----
KING reports to
BLAKE reports to KING
MARTIN reports to BLAKE
ALLEN reports to BLAKE
TURNER reports to BLAKE
JAMES reports to BLAKE
...
14 rows selected.
```

Ranking Rows with the LEVEL Pseudocolumn



Formatting Hierarchical Reports Using **LEVEL** and **LPAD**

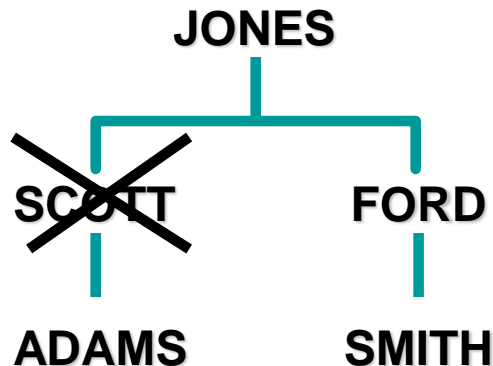
- Create a report displaying company management levels, beginning with the highest level and indenting each of the following levels to the lowest level.

```
SQL> COLUMN org_chart FORMAT A15
SQL> SELECT LPAD(' ', 3 * LEVEL-3) || ename AS org_chart,
2          LEVEL, empno, mgr, deptno
3  FROM    emp
4  START WITH mgr IS NULL
5  CONNECT BY PRIOR empno = mgr;
SQL> CLEAR COLUMN
```

Pruning Branches

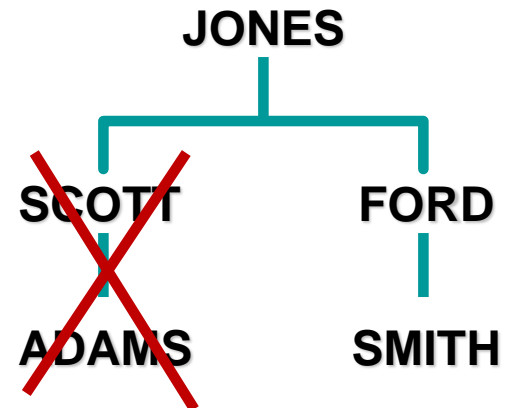
Use the WHERE clause
to eliminate a node.

```
WHERE ename != 'SCOTT'
```



Use the CONNECT BY clause
to **eliminate a branch**.

```
CONNECT BY PRIOR empno = mgr  
AND ename != 'SCOTT'
```



```
SELECT ename, 'reports to', prior ename as mgr_name  
FROM emp  
WHERE ename != 'SCOTT'  
START WITH ename = 'JONES'  
CONNECT BY PRIOR empno = mgr;
```

Ordering Data

- Create a hierarchical report sorted by department number.

```
SQL> BREAK ON deptno
SQL> SELECT LEVEL, deptno, empno, ename, job
  2 FROM emp
  3 START WITH mgr IS NULL
  4 CONNECT BY PRIOR empno = mgr
  5 ORDER BY deptno;
SQL> CLEAR BREAK
```

Advanced Connect-By in Oracle

- More about connect-by
 - Search “asktom.oracle.com” using keyword either “connect by”, “tree query”, or “child-parent relationship”
 - [Trees in SQL: Nested Sets and Materialized Path](#) – Good!!
 - [Trees in Oracle SQL](#)
 - Oracle 9i: Connect by in Joining, Sibling, Sys_Connect_By_Path
 - Oracle 10g: Connect_By_Root, Connect_By_Isleaf, Connect_By_Iscycle (See [New CONNECT-BY in Oracle 10G](#) for details)
- Advantages: performance, flexibility, productivity

Summary

- You can use hierarchical queries to view a hierarchical relationship between rows in a table.
- You specify the direction and starting point.
- You can eliminate nodes or branches by pruning.
- More general query: **DEDUCTIVE DB (recursive query)**

Recursion in SQL 2000 (Ch 24.4) – Example

```
WITH RECURSIVE ancestor (name,aname) AS (  
    SELECT name, father  
    FROM parent  
    UNION  
    SELECT p.name, a.aname  
    FROM parent p, ancestor a  
    WHERE p.father = a.name  
)  
SELECT *  
FROM ancestor;
```

SQL Expressive Powers

1. Relational Algebra or Calculus
2. Aggregation / Grouping
3. **Deductive Logics** / Analytic Functions (Windowing)
4. Data Mining Features

- IBM DB2 & MS SQL Server support standard recursion
- Recursive Subquery Factoring in Oracle (since 12c)
 - https://docs.oracle.com/database/121/SQLRF/statements_10002.htm#i2129904
 - See also “ch5.sql-ch24,25-1.tree” script