Improving Hierarchy LEVEL Access Performance Using Recursion Function

Kuber Vikram Singh [1] ,Inderjeet Yadav [2]
[1]M.Tech Scholar, Department of Computer Science & Engineering, NGFCET,
Maharshi Dayanand University, Palwal, INDIA.
[2]Assistant Professor, Department of Computer Science & Engineering, NGFCET,
Maharshi Dayanand University, Palwal, INDIA.
E-mail:[1]KuberVikramSingh@gmail.com, [2]yadavcse@gmail.com , www.ngfcet.com

Abstract:-In the Current Data base sql query for fetching Hierarchical data like supervisor hierarchy or organization hierarchy etc standard connect by prior command used for fetching data so connect by prior command take longer time for fetching the hierarchical data level for supervisor hierarchy or organization hierarchy etc.

As a workaround, we will provide solutions for fast access for hierarchical data level for supervisor hierarchy or organization hierarchy etc by using recursive function in sql statement

Keywords*: oracle sql connect by prior, Hierarchal Data, supervisor hierarchy level, organization hierarchy , hierarchy level , recursive function.*

____*****____

1. Introduction

In relational Data base it’s not store data in hierarchical fashion Then big problem is how to fetch data in hierarchical fashion for supervisor hierarchy or organization hierarchy etc. Oracle Data base provide standard features Connect by Prior concept for getting data in hierarchical fashion and hierarchy level

Using connect by prior we can fetch data in hierarchy fashion.

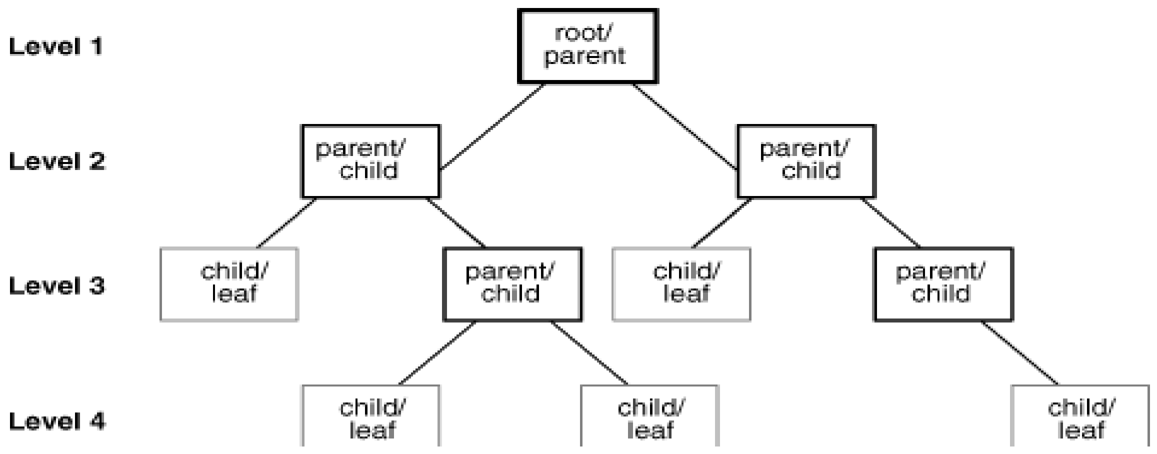
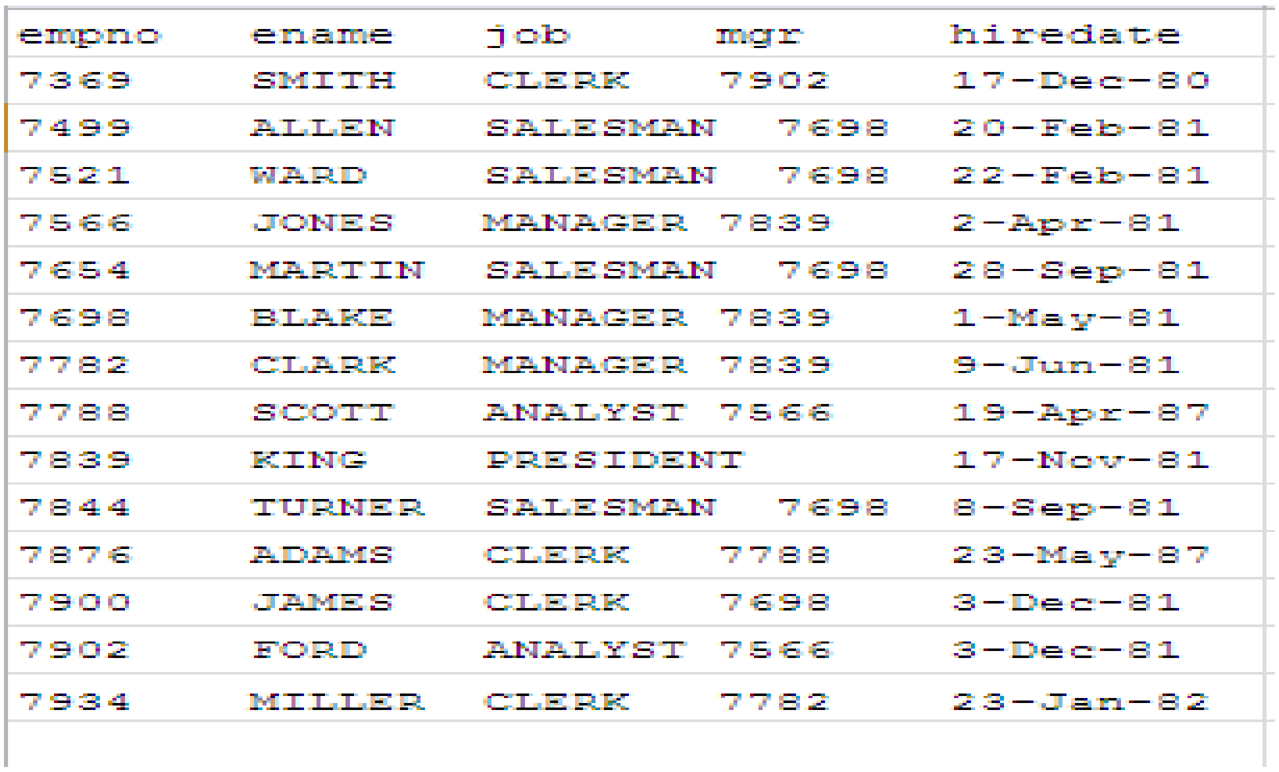


Fig 1.1: Hierarchical Tree

LEVEL shows the level or rank of the particular row in the hierarchical tree.

For example employee/Manger hierarchy tree, Employee/manager tree create a relationship between employee and manger. Employee and manager data lies in same table

We can take Employee table and in this employee table King is top most in the hierarchy. Find below Employee Table Data.

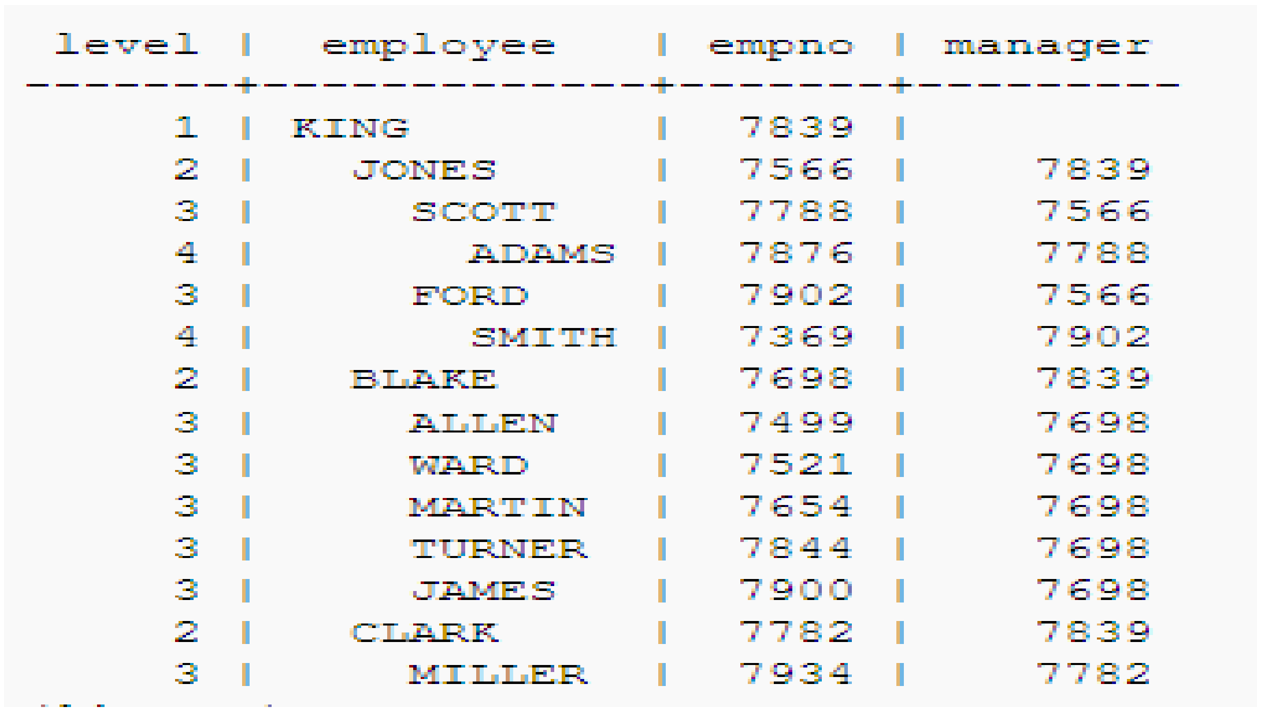


empno	ename	job	mgr	hiredate
7369	SMITH	CLERK	7902	17-Dec-80
7499	ALLEN	SALESMAN	7698	20-Feb-81
7521	WARD	SALESMAN	7698	22-Feb-81
7566	JONES	MANAGER	7839	2-Apr-81
7654	MARTIN	SALESMAN	7698	28-Sep-81
7698	BLAKE	MANAGER	7839	1-May-81
7782	CLARK	MANAGER	7839	9-Jun-81
7788	SCOTT	ANALYST	7566	19-Apr-87
7839	KING	PRESIDENT		17-Nov-81
7844	TURNER	SALESMAN	7698	8-Sep-81
7876	ADAMS	CLERK	7788	23-May-87
7900	JAMES	CLERK	7698	3-Dec-81
7902	FORD	ANALYST	7566	3-Dec-81
7934	MILLER	CLERK	7782	23-Jan-82

Fig 1.2: Employee Table Data

SELECT LEVEL, LPAD (' ', 2 * (LEVEL - 1)) || ename "employee", empno, mgr "manager"
FROM employee START WITH mgr IS NULL
CONNECT BY PRIOR empno = mgr;

Find below output of the above query:



```
 level |   employee    | empno  | manager
-------+---------------+--------+---------
     1 | KING          |  7839  |
     2 |   JONES       |  7566  |    7839
     3 |     SCOTT     |  7788  |    7566
     4 |       ADAMS   |  7876  |    7788
     3 |     FORD      |  7902  |    7566
     4 |       SMITH   |  7369  |    7902
     2 |   BLAKE       |  7698  |    7839
     3 |     ALLEN     |  7499  |    7698
     3 |     WARD      |  7521  |    7698
     3 |     MARTIN    |  7654  |    7698
     3 |     TURNER    |  7844  |    7698
     3 |     JAMES     |  7900  |    7698
     2 |   CLARK       |  7782  |    7839
     3 |     MILLER    |  7934  |    7782
```

Fig 1.3 Employee Hierarch data with Hierarchy Level

So using connect by prior for finding the level its take more time to fetch data, We can reduce for finding hierarchy Level by Recursive functions.

2. Background

As per new design of writing sql query using recursive functions. This will increase the fast access for hierarchy level.

When a functions call itself dynamic run time, this functions call as recursive functions.
This requires creating local copies of its memory structure for each call, Recursive functions use stack concepts for storing data. Recursive functions use exit conditions in code for returning from the functions or stopping conditions of recursive functions.

3. Proposed Work

As per proposed work we will have one recursive functions for finding the Hierarchy level and this recursive functions we can call in sql statement for finding the level data.

Proposed Example: Find Organization LEVEL in Organization Hierarchy

AS-IS : Standard Feature for finding Organization Level in Organization Hierarchy

```
SELECT
    organization_id
    ,haou.name
    ,Level Org_Level
FROM
    apps.HR_ALL_ORGANIZATION_UNITS haou
    ,apps.PER_ORG_STRUCTURE_ELEMENTS OSE
Where
    1=1
    and haou.organization_id = :organization_id -- Input Parameter Pass Organization at run time
    and haou.business_group_id = 101
    and trunc(sysdate) between trunc( haou.date_from) and  nvl( haou.date_to , trunc(sysdate))
    AND haou.ORGANIZATION_ID = OSE.ORGANIZATION_ID_CHILD
    Start With OSE.ORGANIZATION_ID_PARENT = '101'
    CONNECT BY PRIOR OSE.ORGANIZATION_ID_CHILD = OSE.ORGANIZATION_ID_PARENT ;
```

TO-BE Proposed work: Recursions Features for finding Organization Level in Organization Hierarchy

```
FUNCTION org_level(
    p_org_no NUMBER )
  RETURN NUMBER
IS
  /*  l_org_no number ;  begin if p_org_no = 101 then return 1 ; else select OSE.ORGANIZATION_ID_PARENT into
l_org_no from apps.PER_ORG_STRUCTURE_ELEMENTS OSE where OSE.organization_id_child = p_org_no ; return
1 + org_level (l_org_no ); end if; */
  l_org_level NUMBER ;
BEGIN – Starting of Begin
  SELECT ( 0 + org_level_rec ( p_org_no , p_org_no ) )
  INTO l_org_level
  FROM dual ; -- Dual is dummy table
  IF l_org_level >= 0 THEN
    RETURN l_org_level;
  ELSE
    RETURN -1; -- else -1 return some issue with data
  END IF;
```

```
END;
```

Recursion Functions:

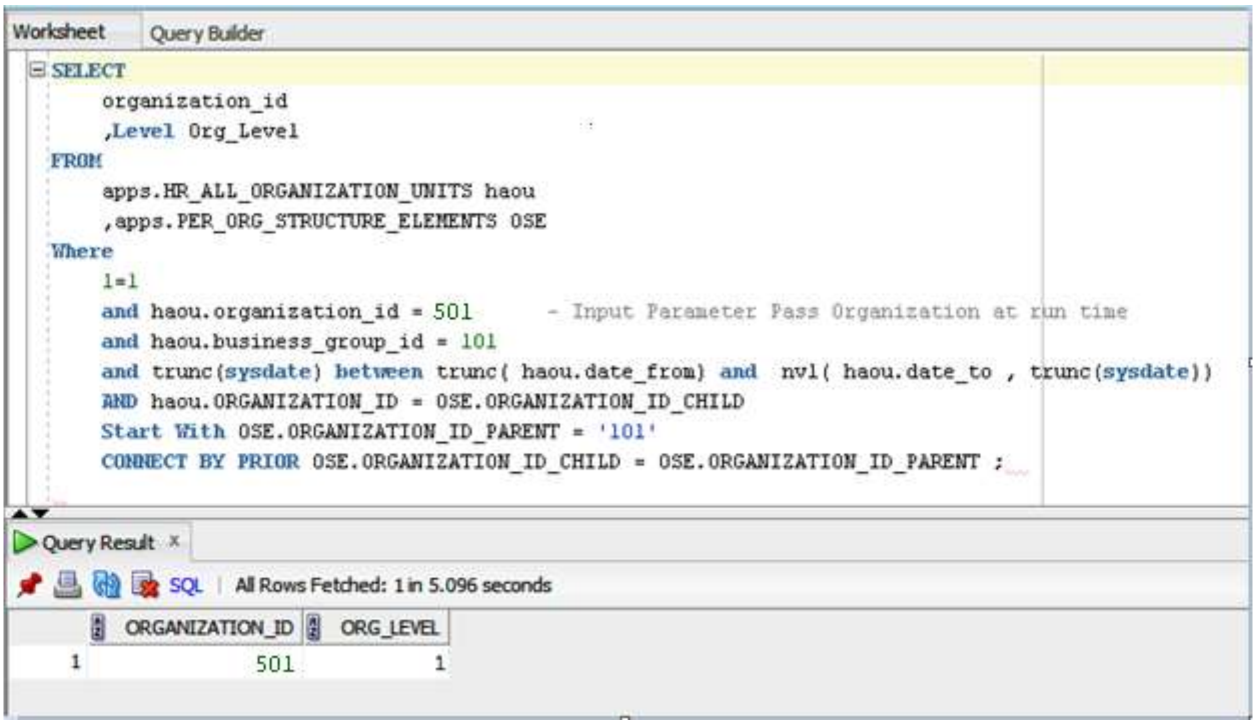
```
FUNCTION org_level_rec(
    p_child_org_no NUMBER,
    p_org_no       NUMBER )
  RETURN NUMBER
IS
  l_org_no NUMBER ;
BEGIN
  IF p_org_no = 101 THEN
    RETURN 0 ;
  ELSE
    BEGIN
      SELECT OSE.ORGANIZATION_ID_PARENT
      INTO l_org_no
      FROM apps.PER_ORG_STRUCTURE_ELEMENTS OSE
      WHERE OSE.organization_id_child = p_org_no ;
    EXCEPTION
    WHEN OTHERS THEN
      RETURN -99;
    END;
    IF p_child_org_no = l_org_no THEN
      RETURN -88;
    END IF;
    RETURN 1 + org_level_rec ( p_child_org_no , l_org_no );
  END IF;
END org_level_rec ;
```

sql Query:

```
select
   organization_id ,
   name ,
 ( 0 + ORG_LEVEL ( org.ORGANIZATION_ID ))  Org_level
from apps.HR_ALL_ORGANIZATION_UNITS   org
where
org.business_group_id = 101
and organization_id = :organization_id  ;  -- Input Parameter Pass Organization at run time
```

Result Response Time screen shot for Reference:

As-Is Connect by Prior: Screen shot for Input Organization Id - 501



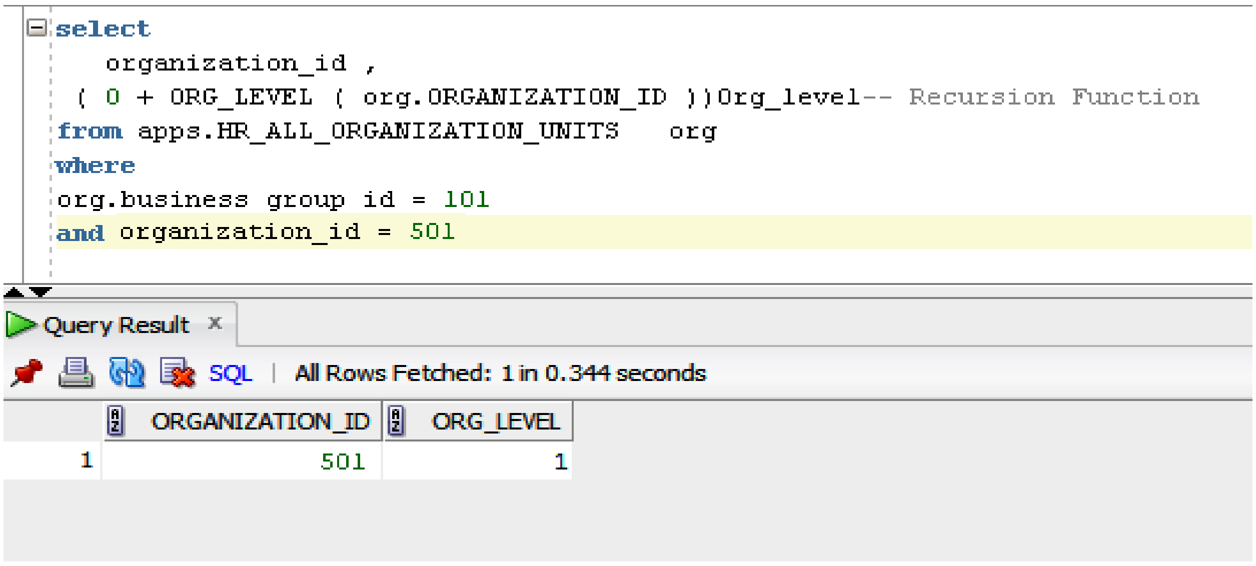
```
SELECT
    organization_id
    ,Level Org_Level
FROM
    apps.HR_ALL_ORGANIZATION_UNITS haou
    ,apps.PER_ORG_STRUCTURE_ELEMENTS OSE
Where
    1=1
    and haou.organization_id = 501       - Input Parameter Pass Organization at run time
    and haou.business_group_id = 101
    and trunc(sysdate) between trunc( haou.date_from) and  nvl( haou.date_to , trunc(sysdate))
    AND haou.ORGANIZATION_ID = OSE.ORGANIZATION_ID_CHILD
    Start With OSE.ORGANIZATION_ID_PARENT = '101'
    CONNECT BY PRIOR OSE.ORGANIZATION_ID_CHILD = OSE.ORGANIZATION_ID_PARENT ;
```

```
All Rows Fetched: 1 in 5.096 seconds

     ORGANIZATION_ID   ORG_LEVEL
1                501           1
```

Fig 3.1: Response time connect by Prior Options in sql query

To-Be Recursion Function: Screen shot for Input Organization Id – 501

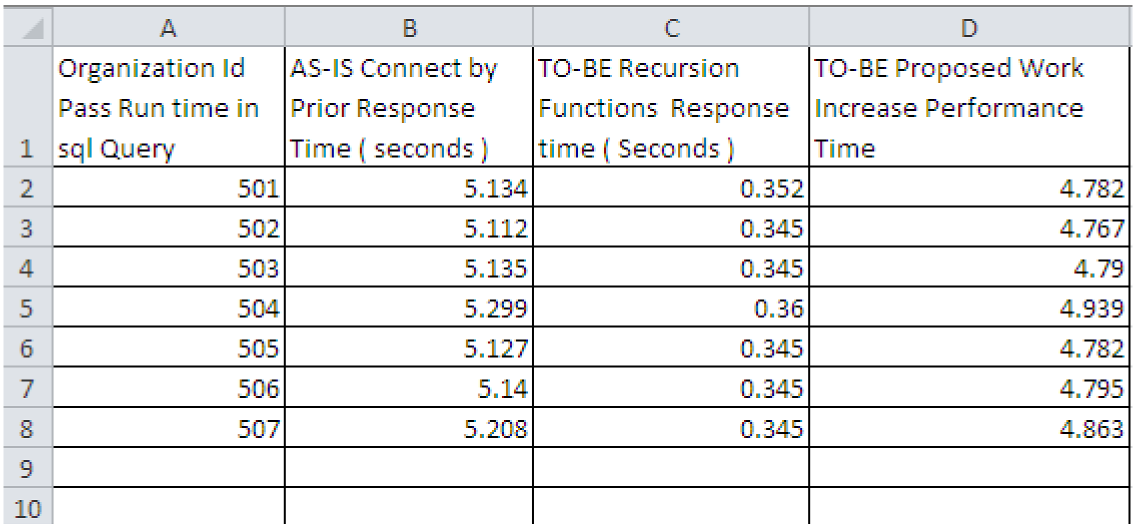


```
select
   organization_id ,
 ( 0 + ORG_LEVEL ( org.ORGANIZATION_ID ))Org_level-- Recursion Function
from apps.HR_ALL_ORGANIZATION_UNITS   org
where
org.business group id = 101
and organization_id = 501
```

```
All Rows Fetched: 1 in 0.344 seconds

     ORGANIZATION_ID   ORG_LEVEL
1                501           1
```

Fig 3.2: Response time Recursion Functions Options in sql query



	A	B	C	D
1	Organization Id Pass Run time in sql Query	AS-IS Connect by Prior Response Time (seconds)	TO-BE Recursion Functions Response time (Seconds)	TO-BE Proposed Work Increase Performance Time
2	501	5.134	0.352	4.782
3	502	5.112	0.345	4.767
4	503	5.135	0.345	4.79
5	504	5.299	0.36	4.939
6	505	5.127	0.345	4.782
7	506	5.14	0.345	4.795
8	507	5.208	0.345	4.863
9				
10				

Fig 3.3: Response Time Compare of Connect by Prior and Recursion Functions

As per above table for getting Level of Organization Response time taken by recursions functions is very less as compare to Connect by Prior standard features.

So Recursions gives better fast response time as compare to Connect by Prior options.

4. Conclusion Future work

Standard Features connect by Prior is very long time taking query so fast response time is always a challenge in sql query is always required.

References

[1] Oracle Database 11g SQL
[2] Mastering Oracle SQL