Business Intelligence Solution Providers

Specialized in creating talent resource pool

Getting Started with Oracle 12c

Hierarchical Retrieval

Description:

BISP is committed to provide BEST learning material to the beginners and advance learners. In the same series, we have prepared a complete end-to end Hands-on Beginner's Guide for Oracle Analytic Functions. The document focuses Grouping and Aggregating data using SQL. Join our professional training program and learn from experts.

History:

Version	Description Change	Author	Publish Date
0.1	Initial Draft	Pawan Madanan	10th Oct 2013
0.1	Review#1	Amit Sharma	10th Oct 2013

Contents:

Hierarchical Queries:

About **Hierarchical** Queries:

- Retrieve data using hierarchical queries , based on hierarchical relationship between rows in a table.
- Oracle being a relational database, does not store records in hierarchical way , but if any parent-child relationship exists between rows , we can construct a hierarchy between the rows.
- With help of a process called Tree Walking , we can create hierarchy between rows in single table.
- A hierarchical query is a method of reporting, in order, the branches of a tree..
- Syntax of a Hierarchical query:

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

  - With the help of "**START WITH**" ,you specify the starting point , ie root row **(or rows)** of the tree.
  - With the help of "**CONNECT BY**",you specify the direction in which hierarchy is walked ,you specify the conditions of parent-child relationship in a hierarchy.
  - **PRIOR** refers the **parent** row

To execute Hierarchical queries , I am going to use table

Tables used : salesstaff

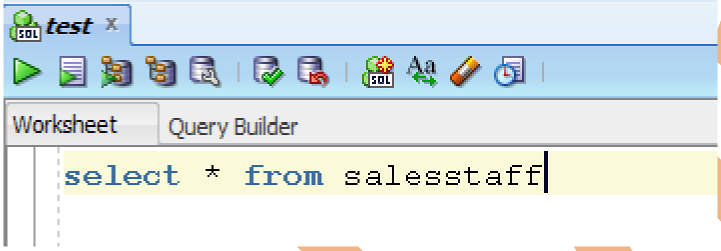
Salesstaff table basically contains data about the people in the sales departement.

Before using Hierarchical queries, lets see whole of the data present in **salesstaff** table.

For we will be executing the following query :

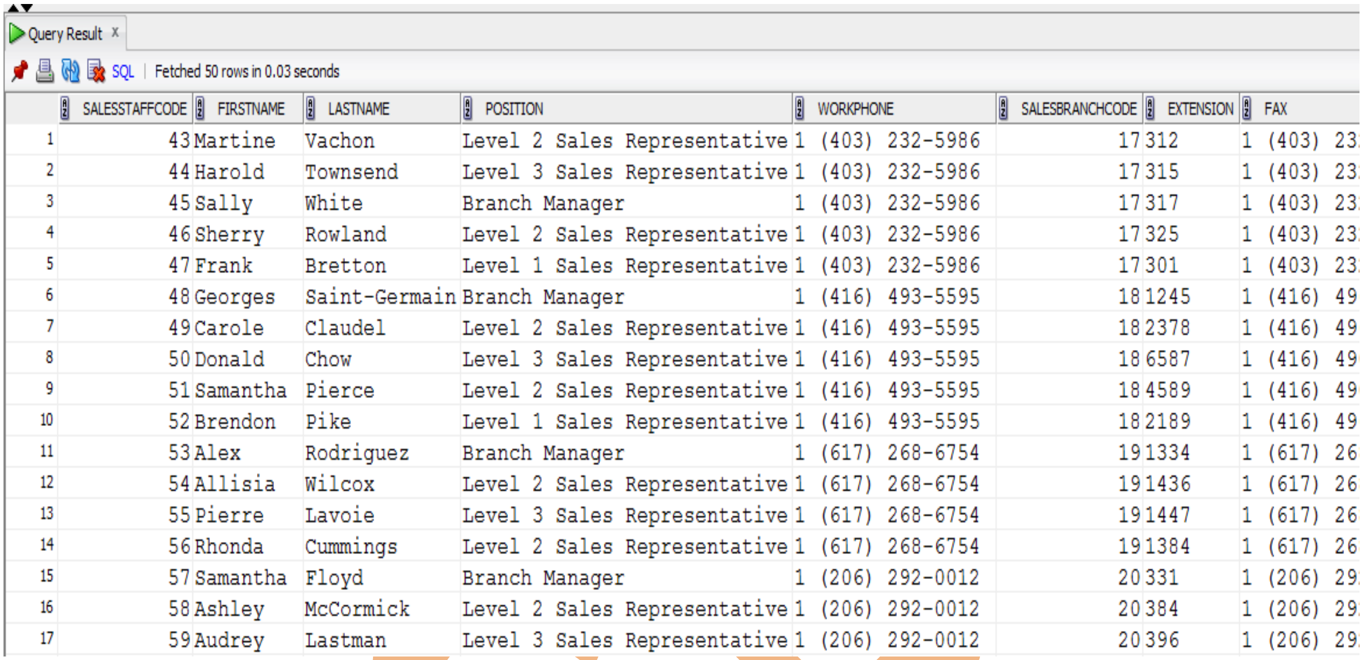
Select * from salesstaff;

Then click on "run" button specified in below image to execute the query .



```
select * from salesstaff
```

Output:



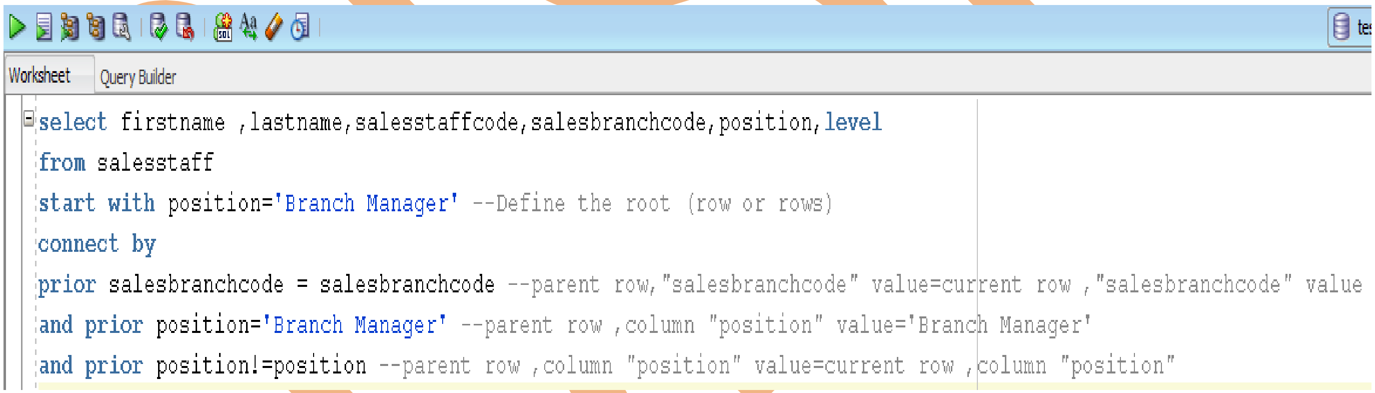
	SALESSTAFFCODE	FIRSTNAME	LASTNAME	POSITION	WORKPHONE	SALESBRANCHCODE	EXTENSION	FAX
1	43	Martine	Vachon	Level 2 Sales Representative	1 (403) 232-5986	17	312	1 (403) 23
2	44	Harold	Townsend	Level 3 Sales Representative	1 (403) 232-5986	17	315	1 (403) 23
3	45	Sally	White	Branch Manager	1 (403) 232-5986	17	317	1 (403) 23
4	46	Sherry	Rowland	Level 2 Sales Representative	1 (403) 232-5986	17	325	1 (403) 23
5	47	Frank	Bretton	Level 1 Sales Representative	1 (403) 232-5986	17	301	1 (403) 23
6	48	Georges	Saint-Germain	Branch Manager	1 (416) 493-5595	18	1245	1 (416) 49
7	49	Carole	Claudel	Level 2 Sales Representative	1 (416) 493-5595	18	2378	1 (416) 49
8	50	Donald	Chow	Level 3 Sales Representative	1 (416) 493-5595	18	6587	1 (416) 49
9	51	Samantha	Pierce	Level 2 Sales Representative	1 (416) 493-5595	18	4589	1 (416) 49
10	52	Brendon	Pike	Level 1 Sales Representative	1 (416) 493-5595	18	2189	1 (416) 49
11	53	Alex	Rodriguez	Branch Manager	1 (617) 268-6754	19	1334	1 (617) 26
12	54	Allisia	Wilcox	Level 2 Sales Representative	1 (617) 268-6754	19	1436	1 (617) 26
13	55	Pierre	Lavoie	Level 3 Sales Representative	1 (617) 268-6754	19	1447	1 (617) 26
14	56	Rhonda	Cummings	Level 2 Sales Representative	1 (617) 268-6754	19	1384	1 (617) 26
15	57	Samantha	Floyd	Branch Manager	1 (206) 292-0012	20	331	1 (206) 29
16	58	Ashley	McCormick	Level 2 Sales Representative	1 (206) 292-0012	20	384	1 (206) 29
17	59	Audrey	Lastman	Level 3 Sales Representative	1 (206) 292-0012	20	396	1 (206) 29

Let s say we want to know the details :

Requirement:

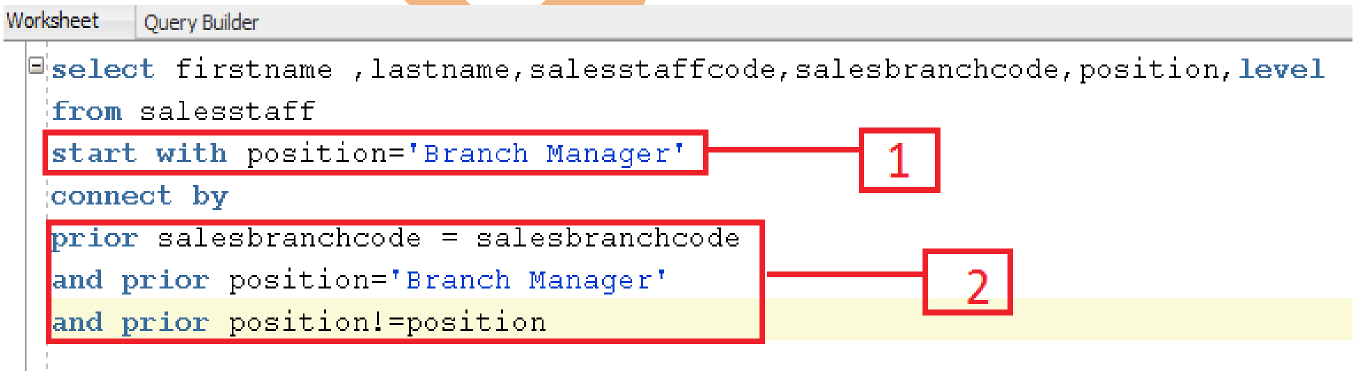
Display management reporting line ie display the mangers of different branches and the branch employees.

Query:



```
select firstname ,lastname,salesstaffcode,salesbranchcode,position,level
from salesstaff
start with position='Branch Manager' --Define the root (row or rows)
connect by
prior salesbranchcode = salesbranchcode --parent row,"salesbranchcode" value=current row ,"salesbranchcode" value
and prior position='Branch Manager' --parent row ,column "position" value='Branch Manager'
and prior position!=position --parent row ,column "position" value=current row ,column "position"
```

Same query:

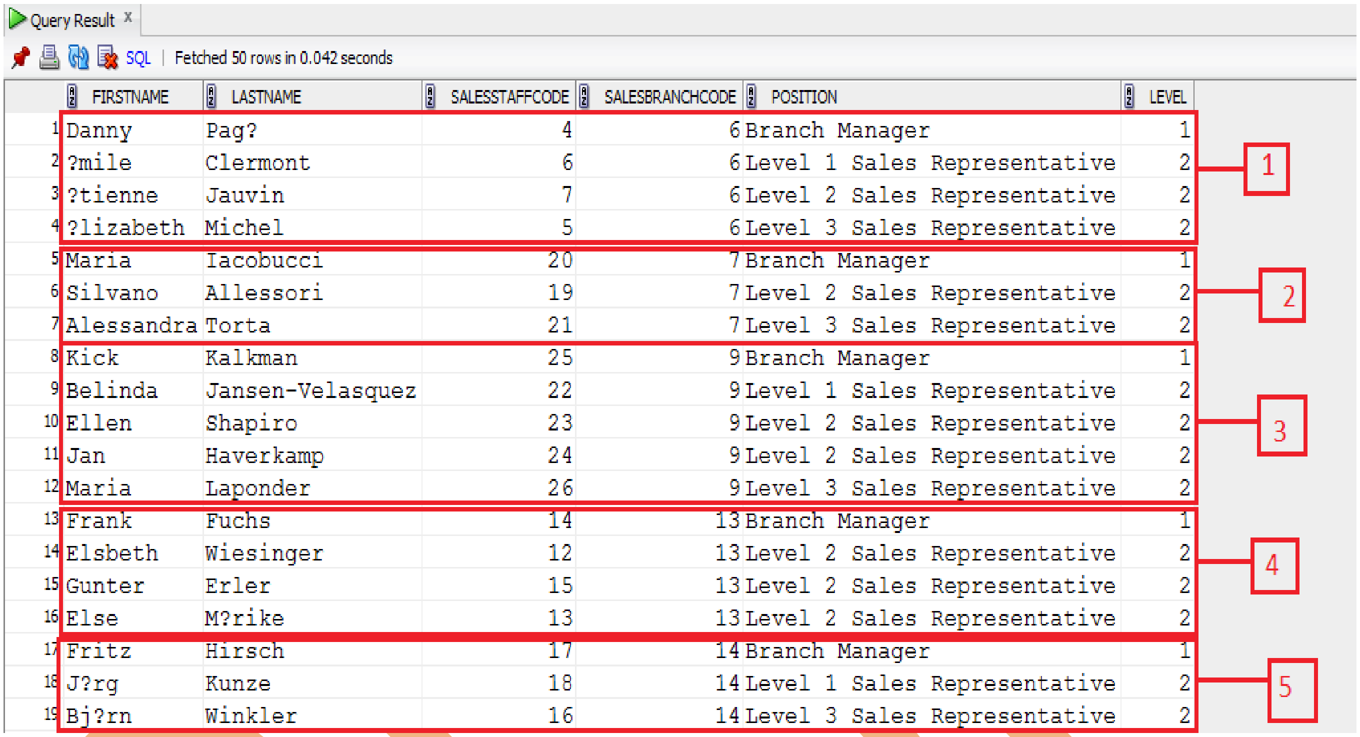


```
select firstname ,lastname,salesstaffcode,salesbranchcode,position,level
from salesstaff
start with position='Branch Manager'
connect by
prior salesbranchcode = salesbranchcode
and prior position='Branch Manager'
and prior position!=position
```

Line labelled 1, has “start with” clause which defines the root of the hierarchical structure.
Line labelled 2, defines the parent-child relationship and defines the direction in which hierarchy is walked .

So , whichever rows **satisfies the connect by conditions** are retrieved as part of the hierarchy.

Output :



Fetched 50 rows in 0.042 seconds

	FIRSTNAME	LASTNAME	SALESSTAFFCODE	SALESBRANCHCODE	POSITION	LEVEL
1	Danny	Pag?	4	6	Branch Manager	1
2	?mile	Clermont	6	6	Level 1 Sales Representative	2
3	?tienne	Jauvin	7	6	Level 2 Sales Representative	2
4	?lizabeth	Michel	5	6	Level 3 Sales Representative	2
5	Maria	Iacobucci	20	7	Branch Manager	1
6	Silvano	Allessori	19	7	Level 2 Sales Representative	2
7	Alessandra	Torta	21	7	Level 3 Sales Representative	2
8	Kick	Kalkman	25	9	Branch Manager	1
9	Belinda	Jansen-Velasquez	22	9	Level 1 Sales Representative	2
10	Ellen	Shapiro	23	9	Level 2 Sales Representative	2
11	Jan	Haverkamp	24	9	Level 2 Sales Representative	2
12	Maria	Laponder	26	9	Level 3 Sales Representative	2
13	Frank	Fuchs	14	13	Branch Manager	1
14	Elsbeth	Wiesinger	12	13	Level 2 Sales Representative	2
15	Gunter	Erler	15	13	Level 2 Sales Representative	2
16	Else	M?rike	13	13	Level 2 Sales Representative	2
17	Fritz	Hirsch	17	14	Branch Manager	1
18	J?rg	Kunze	18	14	Level 1 Sales Representative	2
19	Bj?rn	Winkler	16	14	Level 3 Sales Representative	2

In the above image, you can see the management reporting line , ie Branch Managers of different branches along with the branch employees details.
Block labeled as “1”, are the details of the branch employees along with their Branch Manager, for the Branch with code “6”.
Block labeled as “2”, are the details of the branch employees along with their Branch Manager, for the branch code “7”.
Block labeled as “3”, are the details of the branch employees along with their Branch Manager, for the branch with code “9”.
Block labeled as “4”, are the details of the branch employees along with their Branch Manager, for the branch with code “13”.
Block labeled as “5”, are the details of the branch employees along with their Branch Manager, for the branch with code “14”.

Explanation:

Internally first ,the table is **joined** with itself, then we use the **“start with”** clause to determine the **starting point** ie which are the root rows of the hierarchy , then we apply **“connect by”** conditions(parent-child relationship) to rows ,which decides rows that are going to be the part of the hierarchy .

Note:

“Level” column in the above table is a **Pseudocolumn**, which represents the level of the tree. All the managers of different branches are at level “1” and rest employees in level “2”.

Note:

Don’t use GROUP BY or ORDER BY with hierarchical queries, otherwise it would override the hierarchical order.

2) Applying ORDER SIBLINGS BY clause:

Used to order child rows of the same parent .

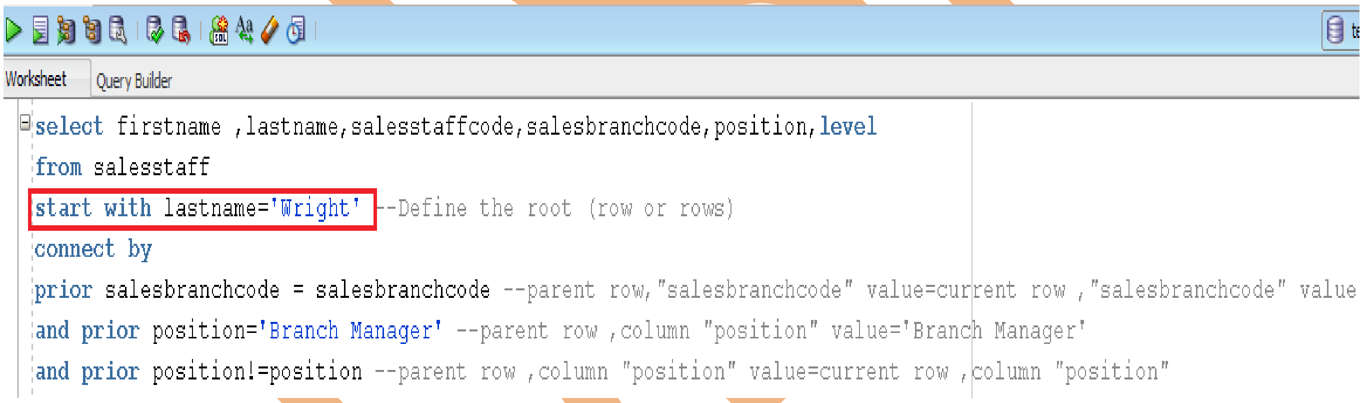
It first preserves the order specflied in the hierarchical query clause and then applies the order siblings clause to the siblings or(children)

Requirement:

Details of the branch manager having last name as “Wright” along with employees of the corresponding branch.

ie Display the management levels beginning with branch manager(having lastname as “Wright”) and below levels.

Query :

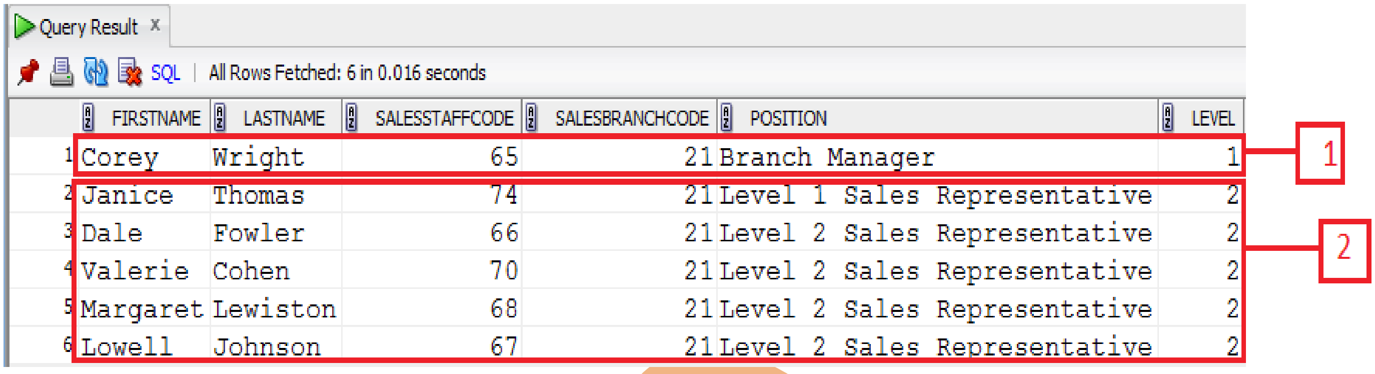


```
select firstname ,lastname,salesstaffcode,salesbranchcode,position,level
from salesstaff
start with lastname='Wright' --Define the root (row or rows)
connect by
prior salesbranchcode = salesbranchcode --parent row,"salesbranchcode" value=current row ,"salesbranchcode" value
and prior position='Branch Manager' --parent row ,column "position" value='Branch Manager'
and prior position!=position --parent row ,column "position" value=current row ,column "position"
```

In the above query , we have defined the root of the tree as person having lastname as “Wright”.(it happens that this person is actually a branch manager , we know it beforehand)

And connect by clause provides the employees present in that branch.

Output: (**BEFORE USING “ORDER SIBLINGS BY”**):



All Rows Fetched: 6 in 0.016 seconds

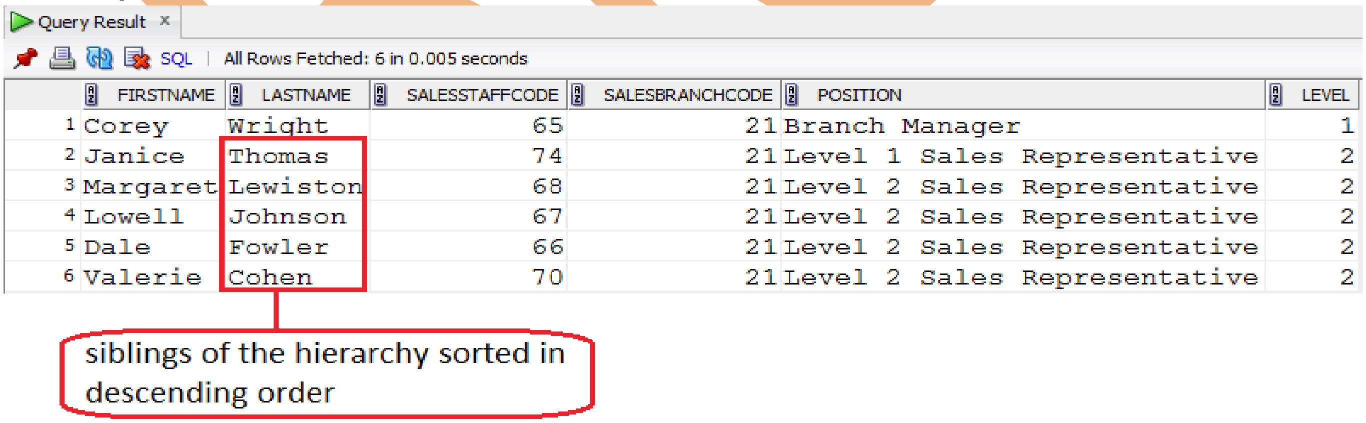
	FIRSTNAME	LASTNAME	SALESSTAFFCODE	SALESBRANCHCODE	POSITION	LEVEL
1	Corey	Wright	65	21	Branch Manager	1
2	Janice	Thomas	74	21	Level 1 Sales Representative	2
3	Dale	Fowler	66	21	Level 2 Sales Representative	2
4	Valerie	Cohen	70	21	Level 2 Sales Representative	2
5	Margaret	Lewiston	68	21	Level 2 Sales Representative	2
6	Lowell	Johnson	67	21	Level 2 Sales Representative	2

Line labeled as 1 , is the root of the hierarchy,as they have level 1 and
Lines labeled as 2, are the children of the parent root row as they have level 2
Notice that the children of the root row are not in any order , if we want to sort them we can use **ORDER**

SIBLINGS BY
NOW USING "ORDER SIBLINGS BY"
Modified query:

```
select firstname ,lastname,salesstaffcode,salesbranchcode,position,level
from salesstaff
start with lastname='Wright' --Define the root (row or rows)
connect by
prior salesbranchcode = salesbranchcode --parent row,"salesbranchcode" value=current row ,"salesbranchcode" value
and prior position='Branch Manager' --parent row ,column "position" value='Branch Manager'
and prior position!=position --parent row ,column "position" value=current row ,column "position"
order siblings by lastname desc
```

New output :



All Rows Fetched: 6 in 0.005 seconds

	FIRSTNAME	LASTNAME	SALESSTAFFCODE	SALESBRANCHCODE	POSITION	LEVEL
1	Corey	Wright	65	21	Branch Manager	1
2	Janice	Thomas	74	21	Level 1 Sales Representative	2
3	Margaret	Lewiston	68	21	Level 2 Sales Representative	2
4	Lowell	Johnson	67	21	Level 2 Sales Representative	2
5	Dale	Fowler	66	21	Level 2 Sales Representative	2
6	Valerie	Cohen	70	21	Level 2 Sales Representative	2

siblings of the hierarchy sorted in descending order

Siblings are sorted in descending order ASCII-batically.

3) Applying the SYS_CONNECT _BY_PATH :

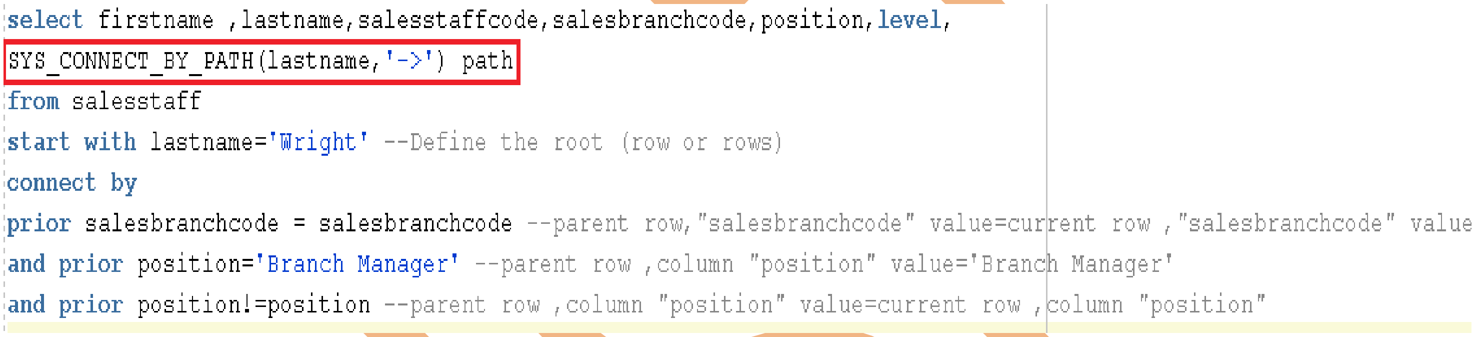
Used in Hierarchical queries

It returns the path of a column value (specified in the first argument) from root to node ,with column values separated by char (specified in the second argument)

Example Syntax:

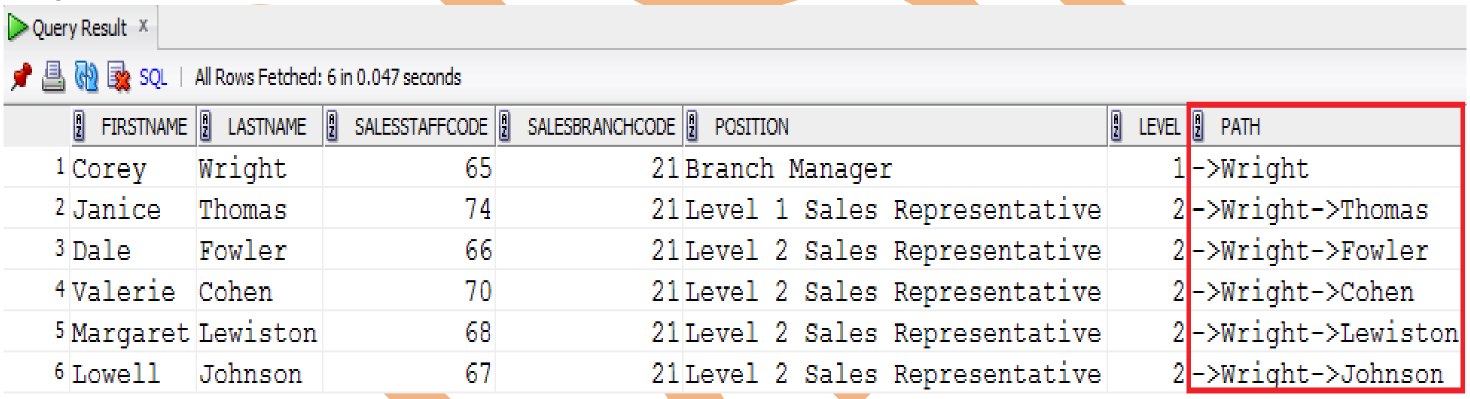
SYS_CONNECT_BY_PATH (column , char)

Query:



```
select firstname ,lastname,salesstaffcode,salesbranchcode,position,level,
SYS_CONNECT_BY_PATH(lastname,'->') path
from salesstaff
start with lastname='Wright' --Define the root (row or rows)
connect by
prior salesbranchcode = salesbranchcode --parent row,"salesbranchcode" value=current row ,"salesbranchcode" value
and prior position='Branch Manager' --parent row ,column "position" value='Branch Manager'
and prior position!=position --parent row ,column "position" value=current row ,column "position"
```

Output :



All Rows Fetched: 6 in 0.047 seconds

	FIRSTNAME	LASTNAME	SALESSTAFFCODE	SALESBRANCHCODE	POSITION	LEVEL	PATH
1	Corey	Wright	65	21	Branch Manager	1	->Wright
2	Janice	Thomas	74	21	Level 1 Sales Representative	2	->Wright->Thomas
3	Dale	Fowler	66	21	Level 2 Sales Representative	2	->Wright->Fowler
4	Valerie	Cohen	70	21	Level 2 Sales Representative	2	->Wright->Cohen
5	Margaret	Lewiston	68	21	Level 2 Sales Representative	2	->Wright->Lewiston
6	Lowell	Johnson	67	21	Level 2 Sales Representative	2	->Wright->Johnson

Path displays the path from root to other nodes.

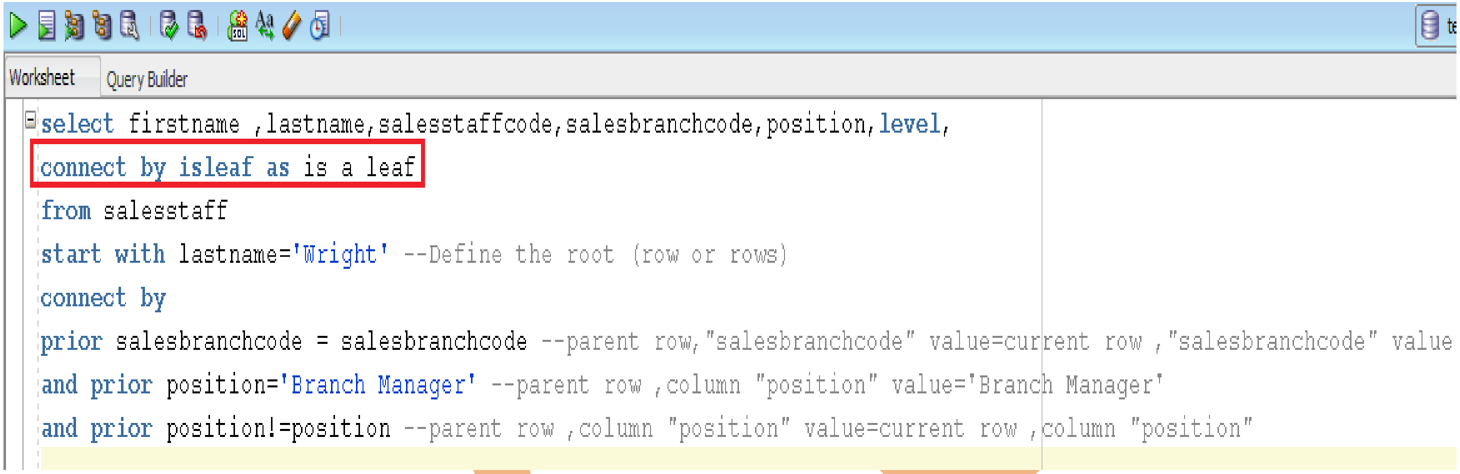
4) Applying CONNECT_BY_IS LEAF :

Is a Pseudocolumn .

Returns 1 if the current row is a leaf otherwise 0.

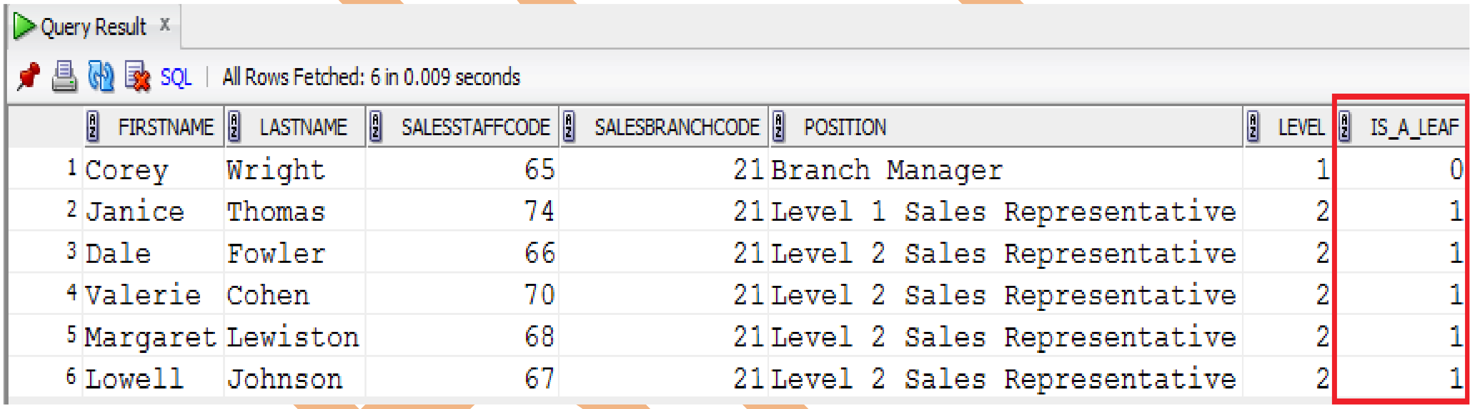
USE: can be used to check if a particular row can be expanded further or not .

Query:



```
select firstname ,lastname,salesstaffcode,salesbranchcode,position,level,
connect by isleaf as is a leaf
from salesstaff
start with lastname='Wright' --Define the root (row or rows)
connect by
prior salesbranchcode = salesbranchcode --parent row,"salesbranchcode" value=current row ,"salesbranchcode" value
and prior position='Branch Manager' --parent row ,column "position" value='Branch Manager'
and prior position!=position --parent row ,column "position" value=current row ,column "position"
```

Output:



All Rows Fetched: 6 in 0.009 seconds

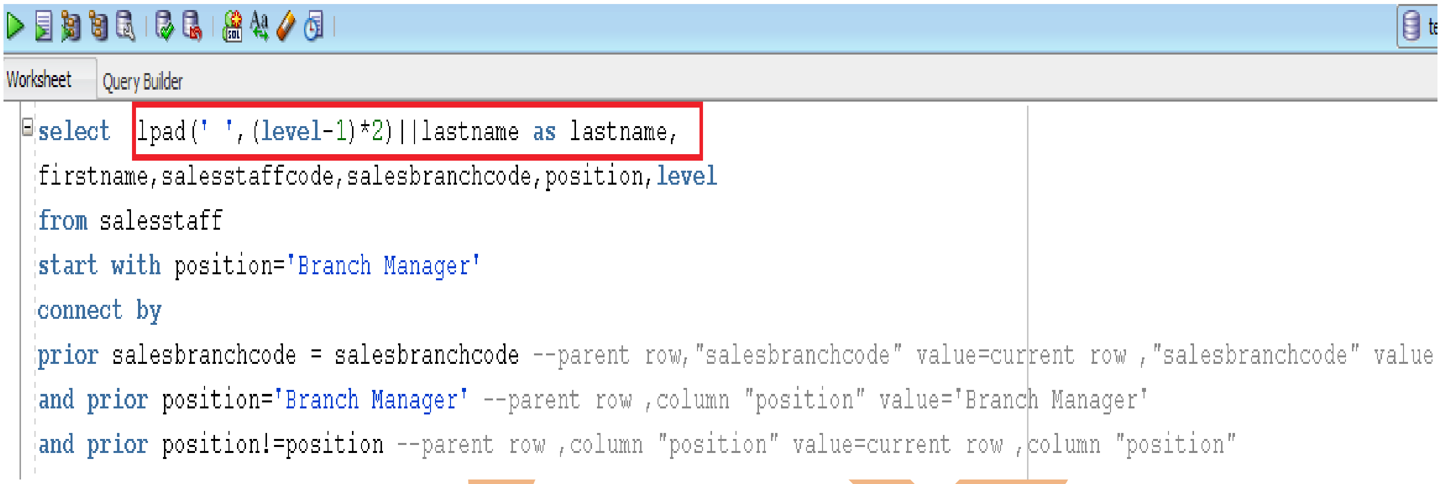
	FIRSTNAME	LASTNAME	SALESSTAFFCODE	SALESBRANCHCODE	POSITION	LEVEL	IS_A_LEAF
1	Corey	Wright	65	21	Branch Manager	1	0
2	Janice	Thomas	74	21	Level 1 Sales Representative	2	1
3	Dale	Fowler	66	21	Level 2 Sales Representative	2	1
4	Valerie	Cohen	70	21	Level 2 Sales Representative	2	1
5	Margaret	Lewiston	68	21	Level 2 Sales Representative	2	1
6	Lowell	Johnson	67	21	Level 2 Sales Representative	2	1

In the above output , except the root ,the rest rows (as they are the leafs in hierarchy) have value 1 in the "**is_a_leaf**" column.

5) Formatting Hierarchical Reports by using LEVEL and LPAD :

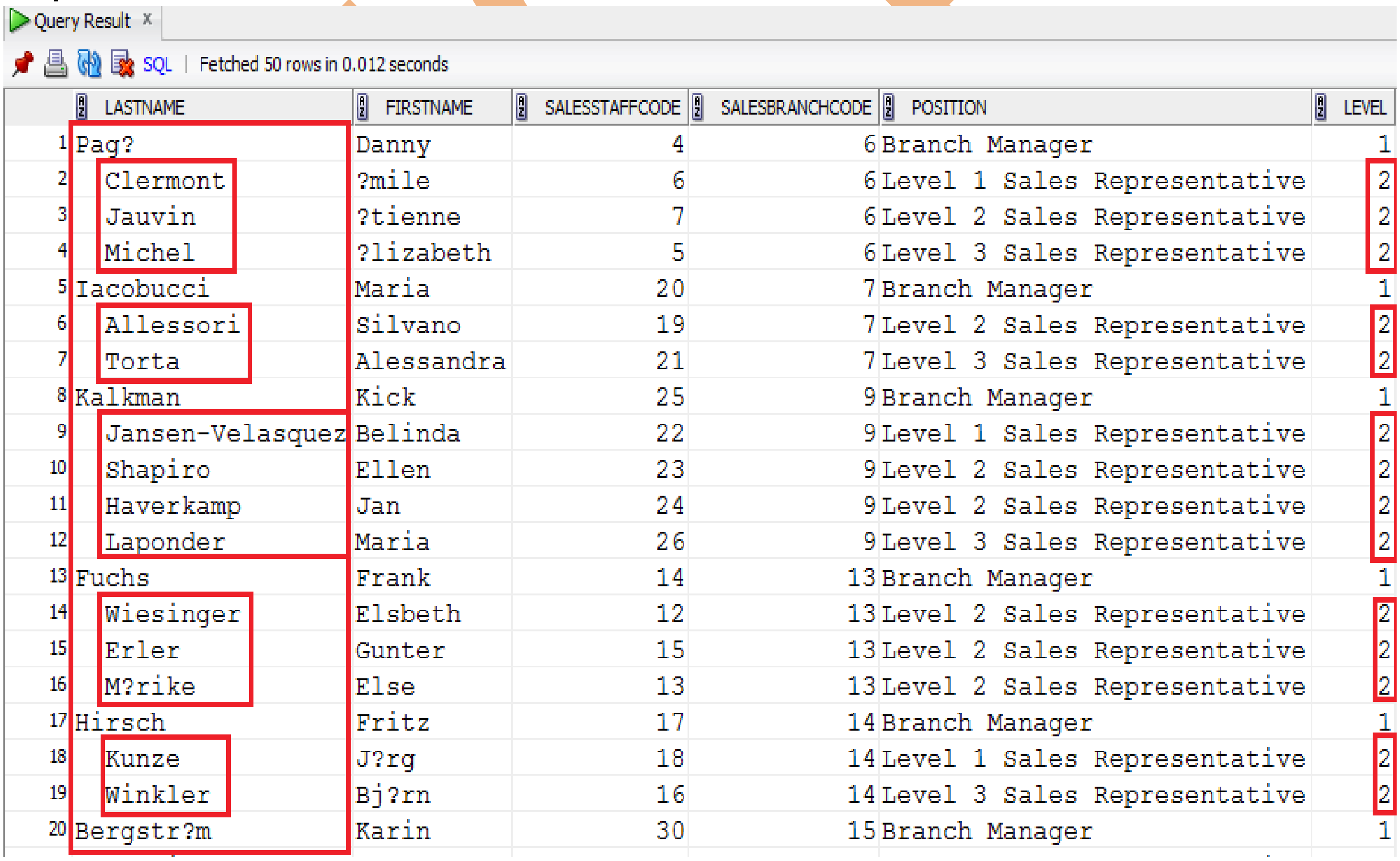
Displaying a hierarchical report displaying the management levels starting with the highest level and the subsequent levels indented.

Query:



```
select lpad(' ',(level-1)*2)||lastname as lastname,
firstname,salesstaffcode,salesbranchcode,position,level
from salesstaff
start with position='Branch Manager'
connect by
prior salesbranchcode = salesbranchcode --parent row,"salesbranchcode" value=current row ,"salesbranchcode" value
and prior position='Branch Manager' --parent row ,column "position" value='Branch Manager'
and prior position!=position --parent row ,column "position" value=current row ,column "position"
```

Output :



Fetched 50 rows in 0.012 seconds

	LASTNAME	FIRSTNAME	SALESSTAFFCODE	SALESBRANCHCODE	POSITION	LEVEL
1	Pag?	Danny	4	6	Branch Manager	1
2	Clermont	?mile	6	6	Level 1 Sales Representative	2
3	Jauvin	?tienne	7	6	Level 2 Sales Representative	2
4	Michel	?lizabeth	5	6	Level 3 Sales Representative	2
5	Iacobucci	Maria	20	7	Branch Manager	1
6	Allessori	Silvano	19	7	Level 2 Sales Representative	2
7	Torta	Alessandra	21	7	Level 3 Sales Representative	2
8	Kalkman	Kick	25	9	Branch Manager	1
9	Jansen-Velasquez	Belinda	22	9	Level 1 Sales Representative	2
10	Shapiro	Ellen	23	9	Level 2 Sales Representative	2
11	Haverkamp	Jan	24	9	Level 2 Sales Representative	2
12	Laponder	Maria	26	9	Level 3 Sales Representative	2
13	Fuchs	Frank	14	13	Branch Manager	1
14	Wiesinger	Elsbeth	12	13	Level 2 Sales Representative	2
15	Erler	Gunter	15	13	Level 2 Sales Representative	2
16	M?rike	Else	13	13	Level 2 Sales Representative	2
17	Hirsch	Fritz	17	14	Branch Manager	1
18	Kunze	J?rg	18	14	Level 1 Sales Representative	2
19	Winkler	Bj?rn	16	14	Level 3 Sales Representative	2
20	Bergstr?m	Karin	30	15	Branch Manager	1

In the above output, column “lastname“ has been indented according to the level of that particular row .

Explanation:

To do proper indentation, we used LPAD function and LEVEL pseudocolumn .

LPAD(' ',(level-1)*2)   : it left pads the ‘ ’ (first argument) with number of spaces until the padding has length equal to the number returned by the second argument.

 1) returns an empty string for level=1 (ie for root ,no indention is applied for the root row)

 2) returns a string consisting of 2 whitespaces, for level=2(ie for child rows , indentation is applied to the left side according to level, here 2 space indentation is applied to the left side)