|  |
| --- |
|  |
| Dimension Table Techniques |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status | Arina Marchenko | 23-NOV-2017 |  |  |

Contents

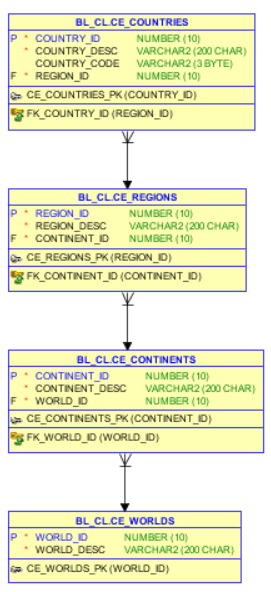
[1. Geo Hierarchy 3](#_Toc499234959)

[2. Dimension Hierarchy Analysis 4](#_Toc499234960)

# Geo Hierarchy

For this task I used the information from the task 1.

There are 4 tables: countries, regions, continents and worlds.



First of all I created with all this tables.

CREATE VIEW DIM\_LOCATIONS AS

SELECT TO\_NUMBER (NULL) AS CHILD\_ID,

'WORLD' AS LEVEL\_NAME,

WORLD\_ID AS PARENT\_ID,

WORLD\_DESC AS NAME

FROM CE\_WORLDS

UNION ALL

SELECT W.WORLD\_ID AS CHILD\_ID,

'CONTINENT' AS LEVEL\_NAME,

C.CONTINENT\_ID AS PARENT\_ID,

CONTINENT\_DESC AS NAME

FROM CE\_CONTINENTS C JOIN CE\_WORLDS W ON C.WORLD\_ID=W.WORLD\_ID

UNION ALL

SELECT C.CONTINENT\_ID AS CHILD\_ID,

'REGION' AS LEVEL\_NAME,

R.REGION\_ID AS PARENT\_ID,

R.REGION\_DESC AS NAME

FROM CE\_REGIONS R JOIN CE\_CONTINENTS C ON R.CONTINENT\_ID=C.CONTINENT\_ID

UNION ALL

SELECT R.REGION\_ID AS CHILD\_ID,

'COUNTRY' AS LEVEL\_NAME,

C.COUNTRY\_ID AS PARENT\_ID,

C.COUNTRY\_DESC AS NAME

FROM CE\_COUNTRIES C JOIN CE\_REGIONS R ON C.REGION\_ID=R.REGION\_ID;



# Dimension Hierarchy Analysis

There are two options how to build the hierarchy. One of them is connect by.

We need to write start with. Then connect by prior, which means our start point of hierarchy.

From the beginning we have root, after this branch and in the end leaf.

Here is the script and screenshot that shows the process of hierarchy.

SELECT PARENT\_ID, NAME,

LTRIM ( SYS\_CONNECT\_BY\_PATH ( NAME, '==>' ), '==>' ) AS PATH,

CASE

WHEN LEVEL = 4 AND CONNECT\_BY\_ISLEAF = 1

THEN 'LEAF'

WHEN LEVEL = 3 AND CONNECT\_BY\_ISLEAF = 0

THEN 'BRANCH'

WHEN LEVEL = 2 AND CONNECT\_BY\_ISLEAF = 0

THEN 'BRANCH'

WHEN LEVEL = 1 AND CONNECT\_BY\_ISLEAF = 0

THEN 'ROOT'

END AS DEFINITION,

(SELECT COUNT(\*)

FROM DIM\_LOCATION DL

WHERE DIM.PARENT\_ID=CHILD\_ID

) AS AMOUNT\_OF\_CHILDREN

FROM DIM\_LOCATION DIM

START WITH CHILD\_ID IS NULL

CONNECT BY PRIOR PARENT\_ID = CHILD\_ID

ORDER SIBLINGS BY 2;

# 

Another option is with as. But in this case we decided to find the information without countries. Here is the script and screenshot that show how it works

WITH WITH\_QUERY AS

(SELECT PARENT\_ID, NAME,

LTRIM ( SYS\_CONNECT\_BY\_PATH ( NAME, '==>' ), '==>' ) AS PATH,

CASE

WHEN LEVEL = 4 AND CONNECT\_BY\_ISLEAF = 1

THEN 'LEAF'

WHEN LEVEL = 3 AND CONNECT\_BY\_ISLEAF = 0

THEN 'BRANCH'

WHEN LEVEL = 2 AND CONNECT\_BY\_ISLEAF = 0

THEN 'BRANCH'

WHEN LEVEL = 1 AND CONNECT\_BY\_ISLEAF = 0

THEN 'ROOT'

END AS DEFINITION,

(SELECT COUNT(\*)

FROM DIM\_LOCATION DL

WHERE DL.PARENT\_ID=CHILD\_ID

) AS AMOUNT\_OF\_CHILDREN

FROM DIM\_LOCATION DIM

START WITH CHILD\_ID IS NULL

CONNECT BY NOCYCLE PRIOR PARENT\_ID= CHILD\_ID

ORDER SIBLINGS BY 2

)

SELECT \* FROM WITH\_QUERY WHERE AMOUNT\_OF\_CHILDREN!=0;

