

|  |
| --- |
| EXTRACT, TRANSFORM, LOAD  Labwork 3  Dimension Tables’ Techniques |

Оглавление

[1 gEO ИЕРАРХИЯ 3](#_Toc499054950)

[1.1 сОЗДАНИЕ VIEW GEO 3](#_Toc499054951)

[1.2 GEO ИЕРАРХИЯ С CONNECT\_BY 4](#_Toc499054952)

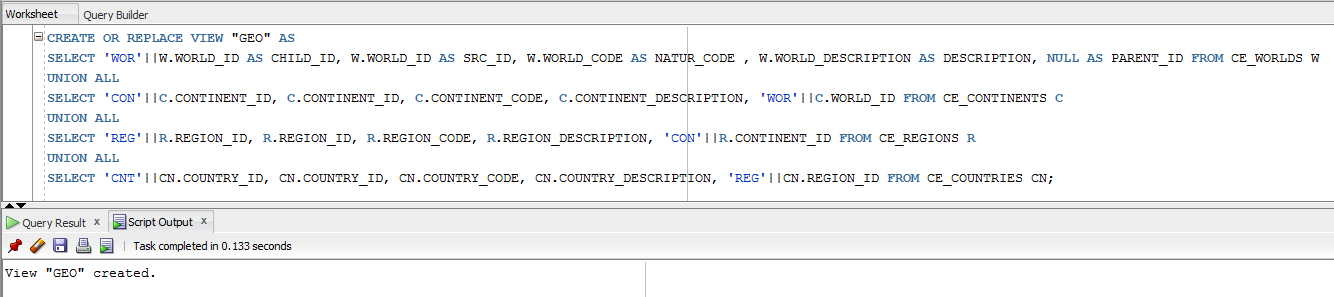
[1.3 GEO ИЕРАРХИЯ С WITH 5](#_Toc499054953)

# gEO ИЕРАРХИЯ

Все скрипты находятся на Git в папке [BI-Lab-2017](https://github.com/mikitakandratsiuk/BI-Lab-2017)/[\_2. ETL](https://github.com/mikitakandratsiuk/BI-Lab-2017/tree/master/_2.%20ETL)/[Tasks](https://github.com/mikitakandratsiuk/BI-Lab-2017/tree/master/_2.%20ETL/Tasks)/[Maryna\_Hlazunova](https://github.com/mikitakandratsiuk/BI-Lab-2017/tree/master/_2.%20ETL/Tasks/Maryna_Hlazunova)/Task 03/. Эта директория является домашней для данной лабораторной работы и далее упоминаться не будет.

## сОЗДАНИЕ VIEW GEO

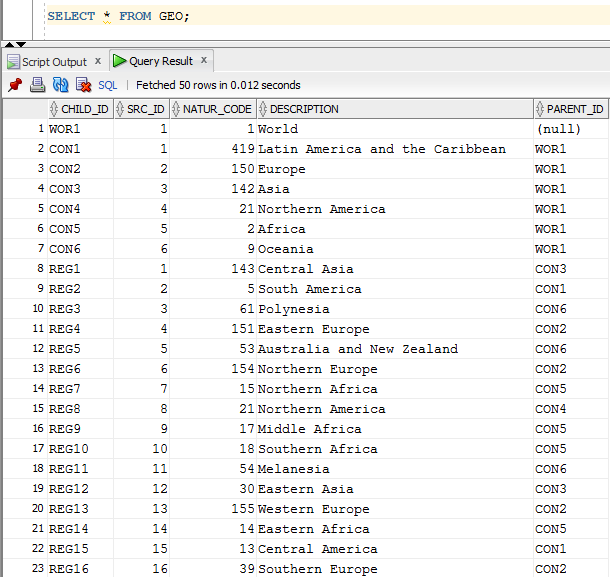
Создание вью на изначальных данных:



В дальнейшем будет использоваться как основа для рекурсивных запросов.

Скрипт располагается в файле create\_geo.sql.

Пример данных:



## GEO ИЕРАРХИЯ С CONNECT\_BY

Создание новой вью с использованием connect\_by:

CREATE OR REPLACE VIEW geo\_connect\_by AS

SELECT

g.child\_id,

g.description,

g.parent\_id,

ltrim(

sys\_connect\_by\_path(description,'==>'),

'==>'

) AS path,

(

CASE

WHEN level = 1 THEN 'Y'

ELSE 'N'

END

) AS root,

(

CASE

WHEN

level <> 1

AND

CONNECT\_BY\_ISLEAF = 0

THEN 'Y'

ELSE 'N'

END

) AS branch,

(

CASE

WHEN CONNECT\_BY\_ISLEAF = 1 THEN 'Y'

ELSE 'N'

END

) AS leaf,

(

SELECT

COUNT(\*)

FROM

geo

START WITH

parent\_id = g.child\_id

CONNECT BY

PRIOR child\_id = parent\_id

) AS "Count of Children"

FROM

geo g

START WITH

parent\_id IS NULL

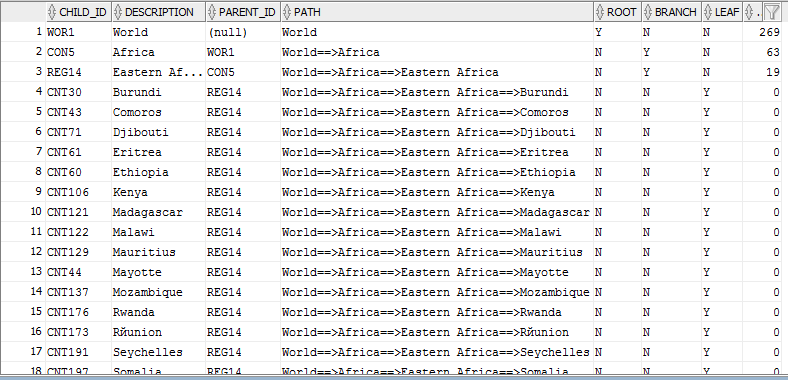
CONNECT BY

PRIOR child\_id = parent\_id

ORDER SIBLINGS BY 2;

Скрипт располагается в файле geo\_connect\_by.sql.

Результат выполнения:



## GEO ИЕРАРХИЯ С WITH

Создание новой вью с использованием with:

CREATE OR REPLACE VIEW geo\_with AS

WITH hier (

child\_id,

description,

lvl,

parent\_id,

path,

root,

branch,

leaf,

"Count of Children"

) AS (

SELECT

child\_id,

description,

0 AS lvl,

parent\_id,

description AS path,

'Y' AS root,

'N' AS branch,

'N' AS leaf,

(

SELECT

COUNT(\*) - 1

FROM

geo

START WITH

parent\_id IS NULL

CONNECT BY

PRIOR child\_id = parent\_id

) AS chl

FROM

geo

WHERE

parent\_id IS NULL

UNION ALL

SELECT

chld.child\_id,

chld.description,

prnt.lvl + 1,

chld.parent\_id,

prnt.path

|| '==>'

|| chld.description,

'N',

(

CASE

WHEN (

SELECT

COUNT(\*)

FROM

geo

WHERE

parent\_id = chld.child\_id

) <> 0 THEN 'Y'

ELSE 'N'

END

) AS branch,

(

CASE

WHEN (

SELECT

COUNT(\*)

FROM

geo

WHERE

parent\_id = chld.child\_id

) = 0 THEN 'Y'

ELSE 'N'

END

) AS leaf,

(

SELECT

COUNT(\*)

FROM

geo

START WITH

parent\_id = chld.child\_id

CONNECT BY

PRIOR child\_id = parent\_id

) AS chl

FROM

geo chld

JOIN hier prnt ON chld.parent\_id = prnt.child\_id

)

SEARCH DEPTH FIRST BY description SET q

SELECT

child\_id,

description,

parent\_id,

path,

root,

branch,

leaf,

"Count of Children"

FROM

hier

ORDER BY q;

Скрипт располагается в файле geo\_with.sql

Результат выполнения:

