|  |
| --- |
| EPAM Systems, RD Dep. |
| Dimension Table Techniques |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status | Valeryia\_Lupanava | 21-NOV-2017 |  |  |

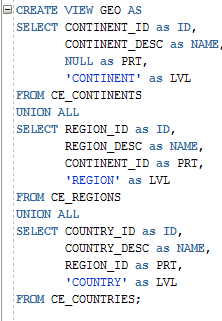
Содержание

[1. Использование CONNECT BY 3](#_Toc499054809)

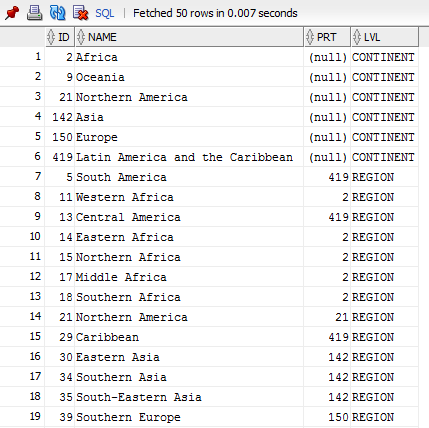
[2. Использование WITH AS 5](#_Toc499054810)

# Использование CONNECT BY

* Сначали создали представление, денормализовав нашу 3nf-структуру.



* Результат.



* Затем написали следующий запрос.

SELECT

g.ID

, g.NAME

, ltrim ( SYS\_CONNECT\_BY\_PATH ( NAME, '==>' ), '==>' ) as PATH

, level lvl

, CONNECT\_BY\_ISLEAF

, (CASE WHEN level = 1 THEN NAME WHEN level = 2 THEN ' ' || NAME WHEN level = 3 THEN ' ' || NAME END) as HIERARCHY

, (CASE WHEN level = 1 THEN 'Branch' WHEN level = 2 THEN 'Root' WHEN level = 3 THEN 'Leaf' END) as Entity

, (SELECT COUNT(\*) FROM GEO START WITH PRT = g.ID CONNECT BY PRIOR ID = PRT) as Children

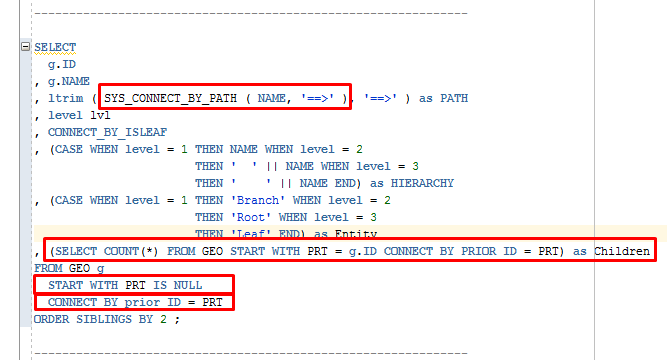
FROM GEO g

START WITH PRT IS NULL

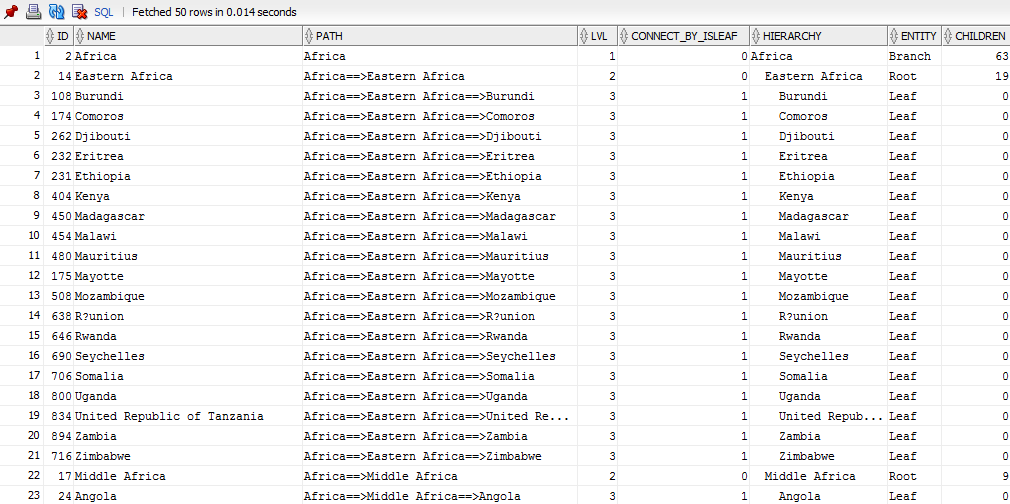
CONNECT BY prior ID = PRT

ORDER SIBLINGS BY 2 ;

* SYS\_CONNECT\_BY\_PATH – возвращает путь от корня до узла.
* Ltrim – убирает '==>' в начале строки.
* level – показывает к какому уровню принадлежит на объект: BRANCH, ROOT или LEAF.
* (SELECT COUNT(\*) FROM GEO START WITH PRT = g.ID CONNECT BY PRIOR ID = PRT) as Children – подсчитывает количество дочерних элементов для каждого элемента.
* START WITH PRT IS NULL – указываем верхнее значение иерархии.
* CONNECT BY prior ID = PRT – находит все записи ID(child) для PRT(parent).



* Результат.



# Использование WITH AS

* Был написан следующий запрос.

WITH m (id,

name,

prt,

lvl)

AS

(SELECT id,

name,

prt,

0 lvl

FROM geo

WHERE prt IS NULL

UNION ALL

SELECT e.id,

e.name,

m.id,

m.lvl + 1

FROM geo e, m

WHERE e.prt = m.id)

SEARCH DEPTH FIRST by id ASC SET ordseq

CYCLE id SET is\_cycle TO 1 DEFAULT 0

SELECT name,

id as child\_is,

prt as parent\_is,

lvl as level\_is

, (CASE WHEN lvl = 0 THEN NAME

WHEN lvl = 1 THEN ' ' || NAME

WHEN lvl = 2 THEN ' ' || NAME END) as HIERARCHY

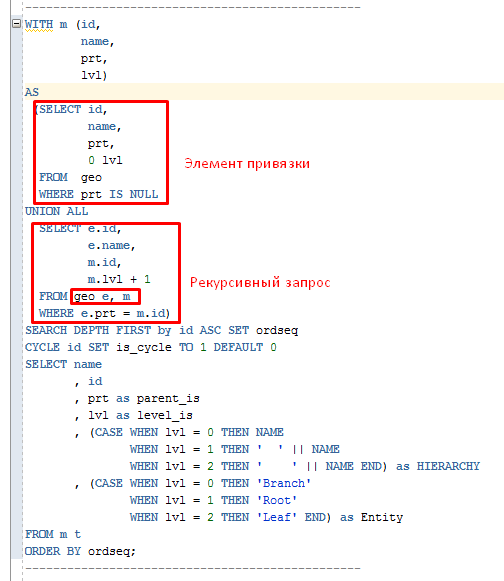
, (CASE WHEN lvl = 0 THEN 'Branch'

WHEN lvl = 1 THEN 'Root'

WHEN lvl = 2 THEN 'Leaf' END) as Entity

FROM m t

ORDER BY ordseq;



* Первый SELECT называется элементом привязки. Это часть, которая определяет верхний уровень иерархии нашего запроса.
* Второй блок, который должен быть соединен с первым с помощью UNION ALL, является рекурсивным. **Он присоединяется не к первому блоку, а к псевдониму запроса WITH AS (в нашем случае m)**.
* SEARCH DEPTH FIRST by id ASC SET ordseq – определяет псевдоколонку **ORDSEQ** как набор данных для **ID** (child\_is) для каждого parent\_id (parent\_is). В последствии по этой колонке проходит сортировка результата.
* Lvl – эквивалень LEVEL, показывает уровень в иерархии нашего объекта: BRANCH, ROOT или LEAF.
* CYCLE - эквивалент для connect\_by\_iscycle - указывает Oracle возвращать строки из запроса, даже если в данных существует CONNECT BY LOOP (строка является и родительской и дочерней для других строк).
* Результат.

