**Report “Lab 3”**

**Anton Tserakhau**

# Business analyses tasks – Reports

## Export Geo Location Reference

I have exported Geo Location Reference on Denormalized table:

SELECT

--country

table\_hier.country\_geo\_id as country\_geo\_id

, cntr.country\_id

, CNTR.COUNTRY\_CODE\_A2

, cntr.country\_code\_a3

, cntr.region\_desc AS country\_desc

--region

, NVL ( table\_hier.region\_geo\_id, -99 ) AS region\_geo\_id

, NVL ( reg.src\_continent\_id, -99 ) AS region\_id

, NVL ( reg.region\_code, 'n.d.' ) AS region\_code

, NVL ( reg.region\_desc, 'n.d.' ) AS region\_desc

-- part

, NVL ( table\_hier.part\_geo\_id, -99 ) AS part\_geo\_id

, NVL ( part.part\_id, -99 ) AS part\_id

, NVL ( part.part\_code, 'n.d.' ) AS part\_code

, NVL ( part.part\_desc, 'n.d.' ) AS part\_desc

-- geo\_systems

, NVL ( table\_hier.system\_geo\_id, -99 ) AS geo\_system\_geo\_id

, NVL ( g\_sys.src\_geo\_system\_id, -99 ) AS geo\_system\_id

, NVL ( g\_sys.geo\_system\_code, 'n.d.' ) AS geo\_system\_code

, NVL ( g\_sys.geo\_system\_desc, 'n.d.' ) AS geo\_system\_desc

-- sub\_groups

, NVL ( table\_hier.sub\_group\_geo\_id, -99 ) AS sub\_group\_geo\_id

, NVL ( sub\_grp.sub\_group\_id, -99 ) AS sub\_group\_id

, NVL ( sub\_grp.sub\_group\_code, 'n.d.' ) AS sub\_group\_code

, NVL ( sub\_grp.sub\_group\_desc, 'n.d.' ) AS sub\_group\_desc

-- groups

, NVL ( table\_hier.group\_geo\_id, -99 ) AS group\_geo\_id

, NVL ( grp.GROUP\_ID, -99 ) AS GROUP\_ID

, NVL ( grp.group\_code, 'n.d.' ) AS group\_code

, NVL ( grp.group\_desc, 'n.d.' ) AS group\_desc

-- group system

, NVL ( table\_hier.group\_system\_geo\_id, -99 ) AS group\_system\_geo\_id

, NVL ( grp\_sys.grp\_system\_id, -99 ) AS group\_system\_id

, NVL ( grp\_sys.grp\_system\_code, 'n.d.' ) AS group\_system\_code

, NVL ( grp\_sys.grp\_system\_desc, 'n.d.' ) AS group\_system\_desc

--

FROM ( SELECT country\_geo\_id

, SUM ( region\_geo\_id ) region\_geo\_id

, SUM ( part\_geo\_id ) part\_geo\_id

, SUM ( system\_geo\_id ) system\_geo\_id

, SUM ( sub\_group\_geo\_id ) sub\_group\_geo\_id

, SUM ( group\_geo\_id ) group\_geo\_id

, SUM ( group\_system\_geo\_id ) group\_system\_geo\_id

FROM ( SELECT DISTINCT

CONNECT\_BY\_ROOT ( links.child\_geo\_id ) country\_geo\_id

, links.parent\_geo\_id

, links.link\_type\_id

, CASE WHEN ( links.link\_type\_id = 3 ) THEN links.parent\_geo\_id ELSE NULL END AS region\_geo\_id

, CASE WHEN ( links.link\_type\_id = 2 ) THEN links.parent\_geo\_id ELSE NULL END AS part\_geo\_id

, CASE WHEN ( links.link\_type\_id = 1 ) THEN links.parent\_geo\_id ELSE NULL END AS system\_geo\_id

, CASE WHEN ( links.link\_type\_id = 6 ) THEN links.parent\_geo\_id ELSE NULL END AS sub\_group\_geo\_id

, CASE WHEN ( links.link\_type\_id = 5 ) THEN links.parent\_geo\_id ELSE NULL END AS group\_geo\_id

, CASE WHEN ( links.link\_type\_id = 4 ) THEN links.parent\_geo\_id ELSE NULL END

AS group\_system\_geo\_id

, LEVEL

FROM w\_geo\_object\_links links

CONNECT BY PRIOR links.parent\_geo\_id = links.child\_geo\_id

START WITH links.child\_geo\_id IN (SELECT DISTINCT geo\_id

FROM cu\_countries)

ORDER BY country\_geo\_id)

GROUP BY country\_geo\_id) table\_hier

LEFT JOIN u\_dw\_references.cu\_countries cntr

ON cntr.geo\_id = table\_hier.country\_geo\_id

LEFT JOIN u\_dw\_references.cu\_geo\_regions reg

ON reg.geo\_id = table\_hier.region\_geo\_id

LEFT JOIN u\_dw\_references.cu\_geo\_parts part

ON part.geo\_id = table\_hier.part\_geo\_id

LEFT JOIN u\_dw\_references.cu\_geo\_systems g\_sys

ON g\_sys.geo\_id = table\_hier.system\_geo\_id

LEFT JOIN u\_dw\_references.cu\_cntr\_group\_systems grp\_sys

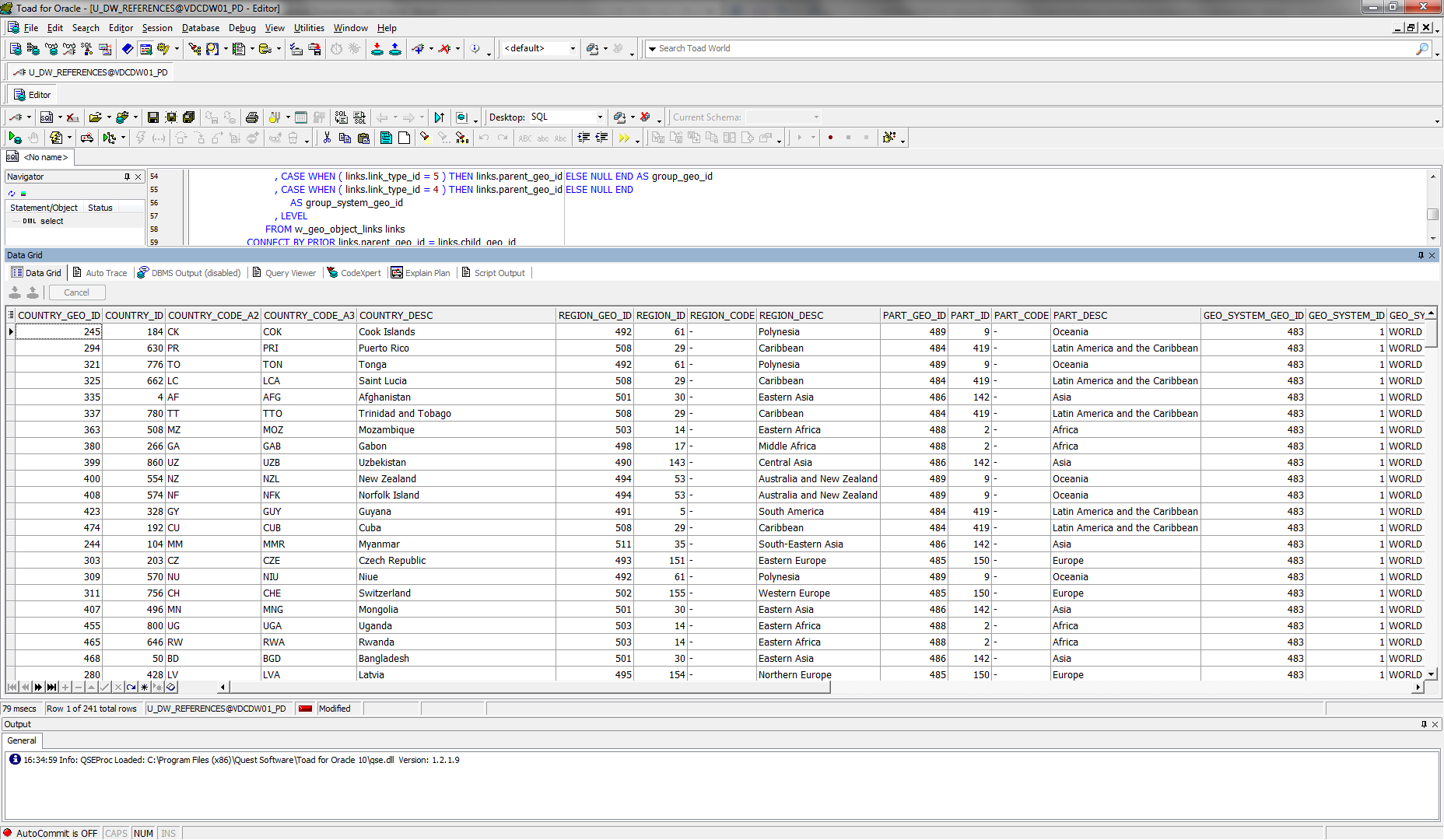
ON grp\_sys.geo\_id = table\_hier.group\_system\_geo\_id

LEFT JOIN u\_dw\_references.cu\_cntr\_groups grp

ON grp.geo\_id = table\_hier.group\_geo\_id

LEFT JOIN u\_dw\_references.cu\_cntr\_sub\_groups sub\_grp

ON sub\_grp.geo\_id = table\_hier.sub\_group\_geo\_id



I have added Additional Columns to table:

o Geo\_id types: Branch, ROOT, Leaf

o Count of childs of Branch or Root, for Leafs this Field you have fill by NULL

o Full path of Dependencies by Example: ROOT -> BRANCH -> BRANCH -> LEAF

SELECT LPAD ( ' '

, LEVEL \* 2

, ' ' )

|| SYS\_CONNECT\_BY\_PATH ( links.child\_geo\_id

, '/' )

path\_with\_id

, LPAD ( ' '

, LEVEL \* 2

, ' ' )

|| SYS\_CONNECT\_BY\_PATH ( links.description

, '/' )

path\_with\_names

, links.geo\_id

, links.description

, LEVEL

, CASE

WHEN LEVEL = 1 THEN 'ROOT'

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

ELSE 'BREANCH'

END

lvl

, ( SELECT COUNT ( \* )

FROM t\_geo\_object\_links

CONNECT BY parent\_geo\_id = PRIOR child\_geo\_id

START WITH parent\_geo\_id = links.geo\_id )

AS count\_of\_childs

FROM (SELECT \*

FROM ( ( SELECT \*

FROM w\_geo\_object\_links )

UNION

( SELECT DISTINCT NULL AS parent\_geo\_id

, geo\_id AS child\_geo\_id

, NULL AS link\_type\_id

FROM cu\_geo\_systems )) idents

LEFT JOIN

( ( SELECT DISTINCT geo\_id

, region\_desc AS description

FROM cu\_countries )

UNION

( SELECT DISTINCT geo\_id

, region\_desc

FROM cu\_geo\_regions )

UNION

( SELECT DISTINCT geo\_id

, part\_desc

FROM cu\_geo\_parts )

UNION

( SELECT DISTINCT geo\_id

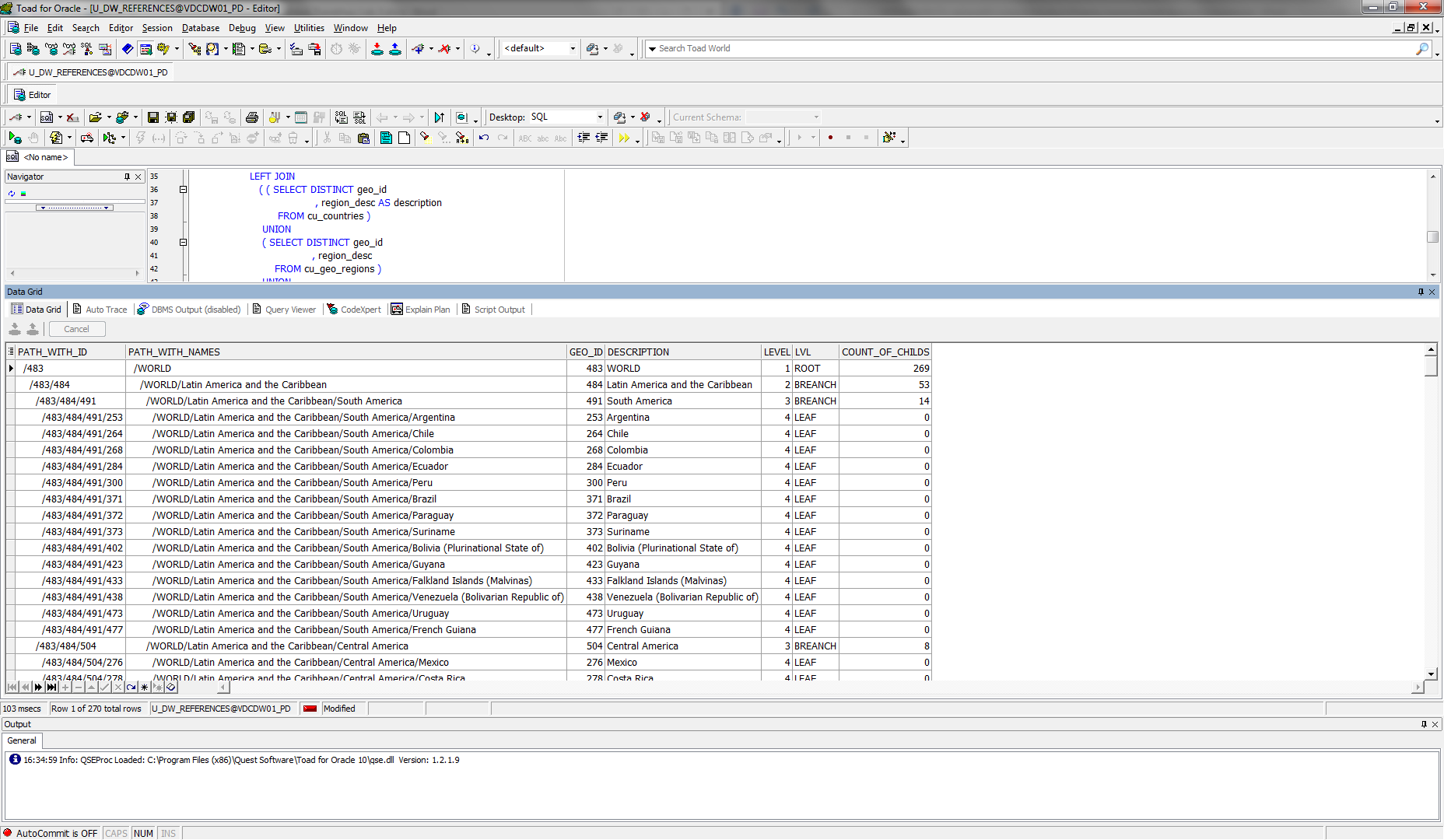
, geo\_system\_code

FROM cu\_geo\_systems )) names

ON idents.child\_geo\_id = names.geo\_id) links

CONNECT BY PRIOR links.child\_geo\_id = links.parent\_geo\_id

START WITH links.parent\_geo\_id IS NULL;



I have created new Schema SB\_MBackUp and new Default TableSpace, prepared load script:

CREATE TABLESPACE ts\_dw\_sb\_mbackup\_01

DATAFILE 'db\_qpt\_dw\_SB\_MBackUp\_01.dat'

SIZE 50M

AUTOEXTEND ON NEXT 50M

SEGMENT SPACE MANAGEMENT AUTO;

CREATE USER u\_dw\_sb\_mbackup

IDENTIFIED BY "%PWD%"

DEFAULT TABLESPACE ts\_dw\_sb\_mbackup\_01;

GRANT CONNECT,RESOURCE TO

u\_dw\_sb\_mbackup;

ALTER USER u\_dw\_sb\_mbackup QUOTA UNLIMITED ON ts\_dw\_sb\_mbackup\_01;

GRANT SELECT ON u\_dw\_references.w\_geo\_object\_links TO u\_dw\_sb\_mbackup;

GRANT SELECT ON u\_dw\_references.cu\_countries TO u\_dw\_sb\_mbackup;

GRANT SELECT ON u\_dw\_references.cu\_geo\_regions TO u\_dw\_sb\_mbackup;

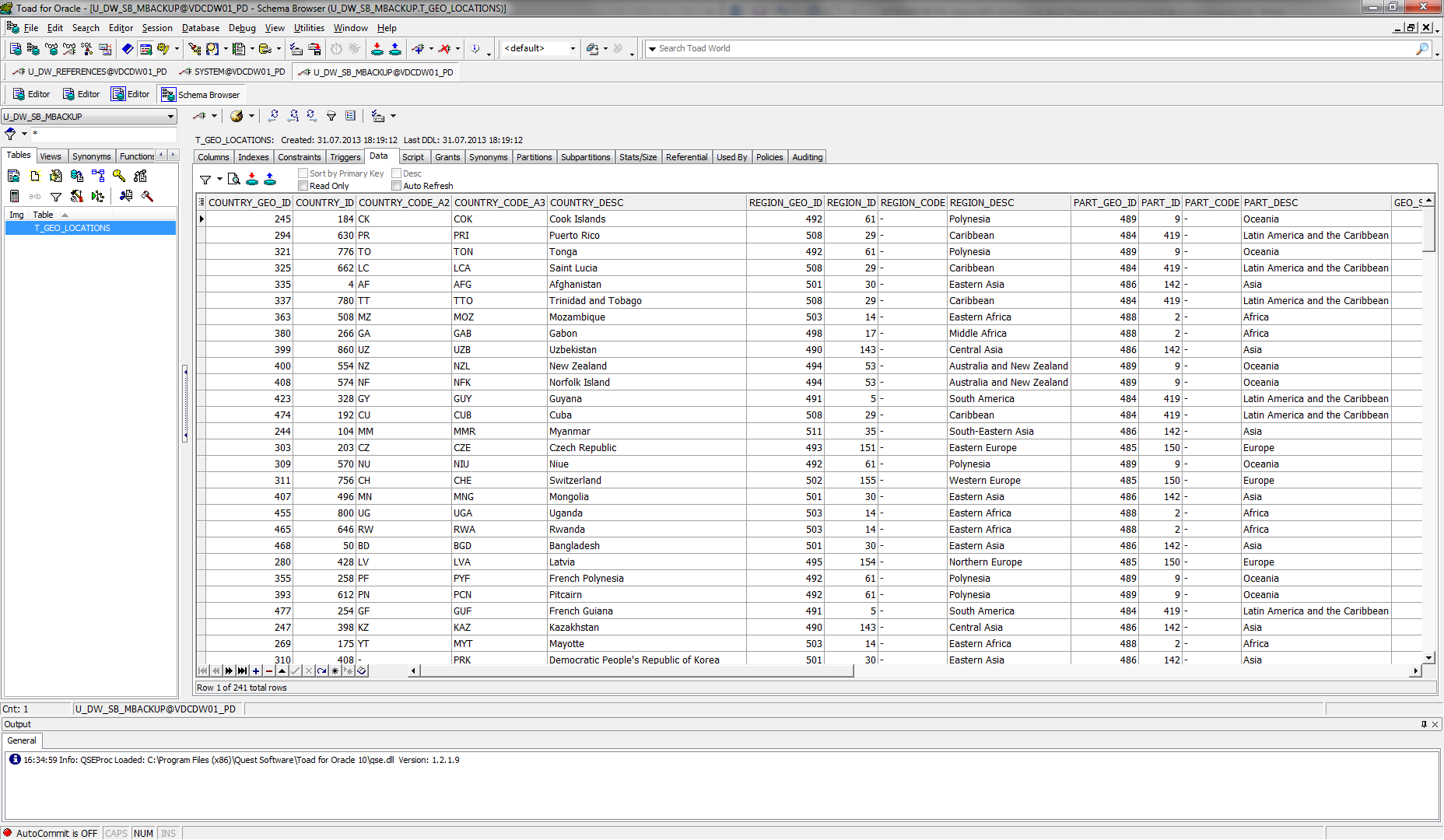
GRANT SELECT ON u\_dw\_references.cu\_geo\_parts TO u\_dw\_sb\_mbackup;

GRANT SELECT ON u\_dw\_references.cu\_geo\_systems TO u\_dw\_sb\_mbackup;

GRANT SELECT ON u\_dw\_references.cu\_cntr\_group\_systems TO u\_dw\_sb\_mbackup;

GRANT SELECT ON u\_dw\_references.cu\_cntr\_groups TO u\_dw\_sb\_mbackup;

GRANT SELECT ON u\_dw\_references.cu\_cntr\_sub\_groups TO u\_dw\_sb\_mbackup;



## Task 02: Analyse Business hierarch Reference Analyses

I have added cities into geo\_locations scheme.

I have created table LC\_CITIES:

Drop table lc\_cities;

CREATE TABLE lc\_cities

(

geo\_id NUMBER ( 22 ) NOT NULL

, city\_id NUMBER ( 22 ) NOT NULL

, city\_desc VARCHAR2 ( 200 CHAR ) NOT NULL

, localization\_id NUMBER ( 22 ) NOT NULL

);

I have inserted data into table LC\_CITIES:

INSERT INTO lc\_cities

SELECT ROWNUM

+ ( SELECT MAX ( geo\_id )

FROM t\_geo\_objects )

AS geo\_id

, ROWNUM AS city\_id

, restaurant\_city AS city

, 1 AS loc

FROM (SELECT DISTINCT restaurant\_city

, restaurant\_country\_iso\_code

, restaurant\_country\_name

FROM u\_dw\_ext\_references.cls\_restaurants) city

LEFT JOIN

cu\_countries cntr

ON ( city.restaurant\_country\_iso\_code = cntr.country\_code\_a2 )

AND ( city.restaurant\_country\_name = cntr.region\_desc );

COMMIT;

I have inserted data into tables T\_GEO\_TYPES, T\_GEO\_OBJECTS, T\_GEO\_OBJECT\_LINKS and added partition to table T\_GEO\_OBJECT\_LINKS:

INSERT INTO t\_geo\_types

VALUES ( 55

, 'CITY'

, 'List all cities' );

INSERT INTO t\_geo\_objects

SELECT geo\_id

, 55

, city\_id

FROM lc\_cities;

ALTER TABLE t\_geo\_object\_links ADD PARTITION p\_city2countries VALUES (7) TABLESPACE ts\_references\_data\_01;

INSERT INTO t\_geo\_object\_links

SELECT DISTINCT cntr.geo\_id AS paren\_geo\_id

, lc\_cities.geo\_id

, 7 AS lvl

FROM u\_dw\_ext\_references.cls\_restaurants city

LEFT JOIN cu\_countries cntr

ON ( city.restaurant\_country\_iso\_code = cntr.country\_code\_a2 )

LEFT JOIN lc\_cities

ON lc\_cities.city\_desc = city.restaurant\_city

WHERE cntr.region\_desc NOT IN

( 'United States Virgin Islands'

, 'China, Hong Kong Special Administrative Region'

, 'China, Macao Special Administrative Region' );

I have created script with cities

SELECT LPAD ( ' '

, LEVEL \* 2

, ' ' )

|| SYS\_CONNECT\_BY\_PATH ( links.child\_geo\_id

, '/' )

path\_with\_id

, LPAD ( ' '

, LEVEL \* 2

, ' ' )

|| SYS\_CONNECT\_BY\_PATH ( links.description

, '/' )

path\_with\_names

, links.geo\_id

, links.description

, LEVEL

, CASE

WHEN LEVEL = 1 THEN 'ROOT'

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

ELSE 'BREANCH'

END

lvl

, ( SELECT COUNT ( \* )

FROM t\_geo\_object\_links

CONNECT BY parent\_geo\_id = PRIOR child\_geo\_id

START WITH parent\_geo\_id = links.geo\_id )

AS count\_of\_childs

FROM (SELECT \*

FROM ( ( SELECT \*

FROM w\_geo\_object\_links )

UNION

( SELECT DISTINCT NULL AS parent\_geo\_id

, geo\_id AS child\_geo\_id

, NULL AS link\_type\_id

FROM cu\_geo\_systems )) idents

LEFT JOIN

( ( SELECT DISTINCT geo\_id

, region\_desc AS description

FROM cu\_countries )

UNION

( SELECT DISTINCT geo\_id

, region\_desc

FROM cu\_geo\_regions )

UNION

( SELECT DISTINCT geo\_id

, part\_desc

FROM cu\_geo\_parts )

UNION

( SELECT DISTINCT geo\_id

, geo\_system\_code

FROM cu\_geo\_systems )

UNION

( SELECT DISTINCT geo\_id

, city\_desc

FROM lc\_cities )) names

ON idents.child\_geo\_id = names.geo\_id) links

CONNECT BY PRIOR links.child\_geo\_id = links.parent\_geo\_id

START WITH links.parent\_geo\_id IS NULL;

