**1. Business analyses tasks – Reports**

**1.1 Export Geo Location Reference**

*/\* Formatted on 7/31/2013 4:11:48 PM (QP5 v5.139.911.3011) \*/*

CREATE TABLESPACE ts\_sb\_mbackup

DATAFILE 'SB\_MBackUp\_data\_01.dat'

SIZE 20M

AUTOEXTEND ON NEXT 10M

SEGMENT SPACE MANAGEMENT AUTO;

CREATE USER sb\_mbackup

IDENTIFIED BY "SB"

DEFAULT TABLESPACE ts\_sb\_mbackup;

GRANT CONNECT,RESOURCE TO sb\_mbackup;

GRANT SELECT ANY TABLE TO sb\_mbackup;

ALTER USER sb\_mbackup QUOTA UNLIMITED ON ts\_sb\_mbackup;

*/\* Formatted on 7/31/2013 4:11:36 PM (QP5 v5.139.911.3011) \*/*

CREATE TABLE denormal\_export\_table

AS

SELECT *--+USE\_HASH(src reg part g\_sys sub\_grp grp grp\_sys)*

country\_geo\_id

, cntr.country\_id

, cntr.country\_code\_a3

, cntr.region\_desc country\_desc

, NVL ( g\_region\_id, -99 ) region\_geo\_id

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

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

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

, NVL ( g\_part\_id, -99 ) part\_geo\_id

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

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

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

, NVL ( g\_system\_id, -99 ) geo\_system\_geo\_id

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

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

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

, NVL ( grp\_item, -99 ) sub\_group\_geo\_id

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

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

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

, NVL ( grp\_group, -99 ) group\_geo\_id

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

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

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

, NVL ( grp\_sys, -99 ) grp\_system\_geo\_id

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

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

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

FROM ( SELECT country\_geo\_id

, SUM ( g\_region\_id ) AS g\_region\_id

, SUM ( g\_part\_id ) AS g\_part\_id

, SUM ( g\_system\_id ) AS g\_system\_id

, SUM ( grp\_item ) AS grp\_item

, SUM ( grp\_group ) AS grp\_group

, SUM ( grp\_sys ) AS grp\_sys

FROM ( SELECT CONNECT\_BY\_ROOT ( child\_geo\_id ) AS country\_geo\_id

, parent\_geo\_id

, DECODE ( link\_type\_id, 3, parent\_geo\_id ) AS g\_region\_id

, DECODE ( link\_type\_id, 2, parent\_geo\_id ) AS g\_part\_id

, DECODE ( link\_type\_id, 1, parent\_geo\_id ) AS g\_system\_id

, DECODE ( link\_type\_id, 6, parent\_geo\_id ) AS grp\_item

, DECODE ( link\_type\_id, 5, parent\_geo\_id ) AS grp\_group

, DECODE ( link\_type\_id, 4, parent\_geo\_id ) AS grp\_sys

FROM u\_dw\_references.w\_geo\_object\_links

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

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

FROM u\_dw\_references.cu\_countries))

GROUP BY country\_geo\_id) src

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

ON cntr.geo\_id = src.country\_geo\_id

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

ON reg.geo\_id = src.g\_region\_id

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

ON part.geo\_id = src.g\_part\_id

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

ON g\_sys.geo\_id = src.g\_system\_id

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

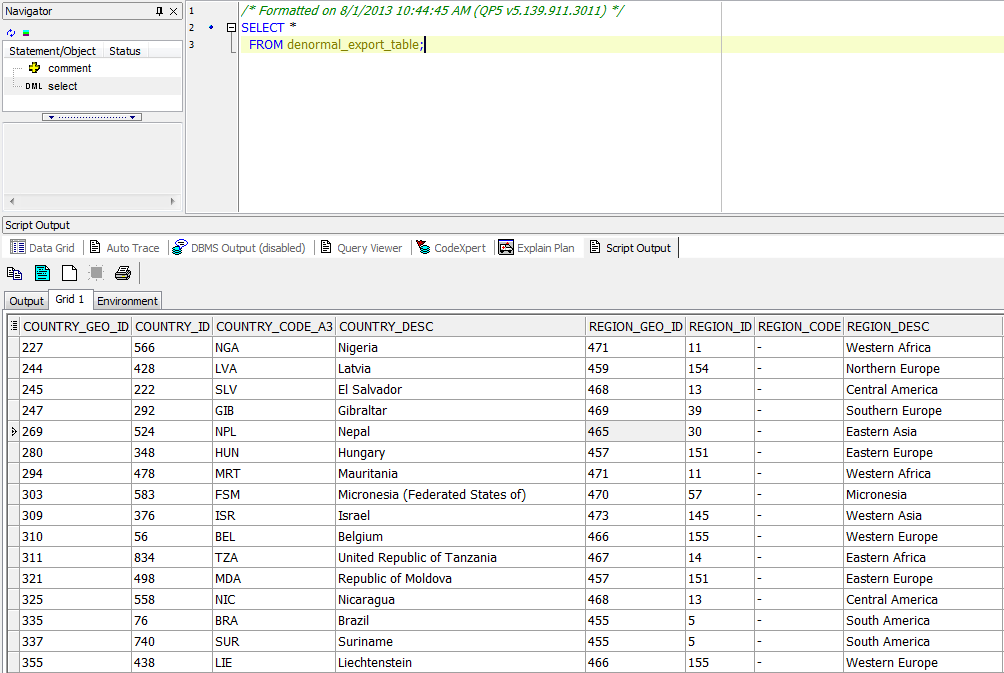
ON grp\_sys.geo\_id = src.grp\_sys

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

ON grp.geo\_id = src.grp\_group

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

ON sub\_grp.geo\_id = src.grp\_item;



*/\* Formatted on 8/1/2013 10:46:08 AM (QP5 v5.139.911.3011) \*/*

SELECT LPAD ( ' '

, LEVEL \* 2 - 1

, ' ' )

|| child\_geo\_id

AS geo\_id

, SYS\_CONNECT\_BY\_PATH ( child\_geo\_id

, ':' )

AS PATH

, link\_type\_id

, CASE

WHEN LEVEL = 1 THEN 'ROOT'

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

ELSE 'BRANCH'

END

AS "LEVEL"

, CASE

WHEN CONNECT\_BY\_ISLEAF = 1 THEN

NULL

ELSE

( SELECT COUNT ( child\_geo\_id )

FROM u\_dw\_references.t\_geo\_object\_links

START WITH parent\_geo\_id = t1.child\_geo\_id

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

END

count\_of\_childs

FROM (SELECT \*

FROM u\_dw\_references.t\_geo\_object\_links

UNION

SELECT NULL

, geo\_id

, NULL

FROM u\_dw\_references.lc\_cntr\_group\_systems

UNION

SELECT NULL

, geo\_id

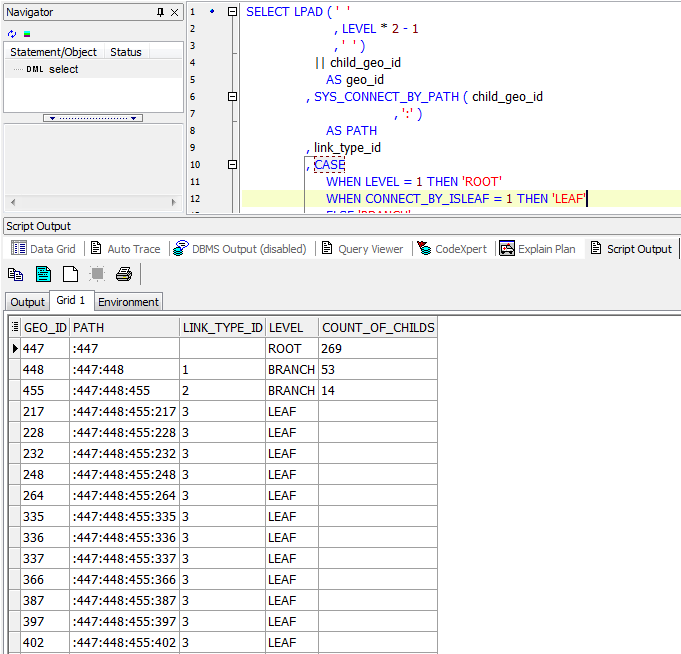
, NULL

FROM u\_dw\_references.lc\_geo\_systems) t1

START WITH parent\_geo\_id IS NULL

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

ORDER SIBLINGS BY child\_geo\_id;



**1.2 Analyze Business** **hierarch Reference Analyses**

