U2M3.LW.Subqueries

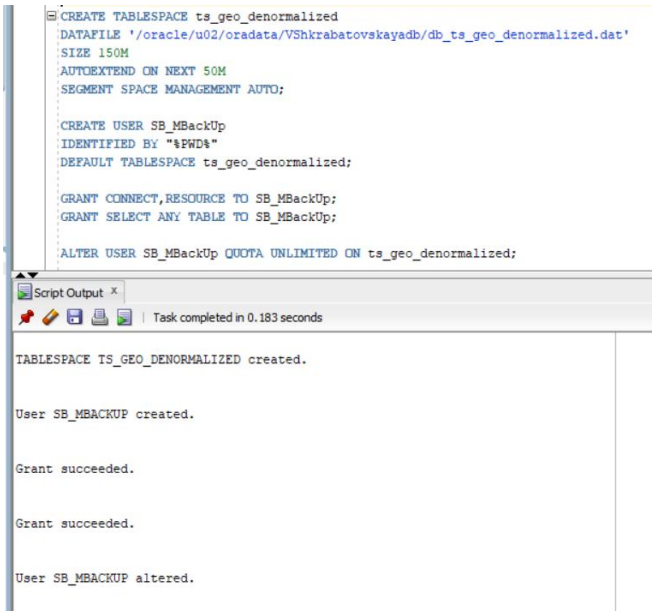
Shkrabatouskaya Vera

https://github.com/VeraShkrabatouskaya/DataMola_Data-Camping-2022

2. Business analyses tasks – Reports

2.1. Task 01: Export Geo Location Reference

Create tablespace ts_geo_denormalized and user SB_MBackUp.

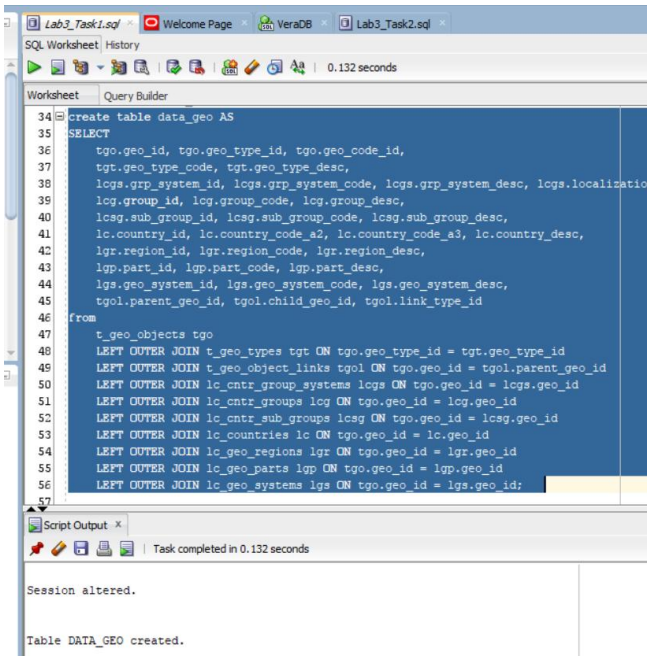


```
CREATE TABLESPACE ts_geo_denormalized
DATAFILE '/oracle/u02/oradata/VShkrabatovskayadb/db_ts_geo_denormalized.dat'
SIZE 150M
AUTOEXTEND ON NEXT 50M
SEGMENT SPACE MANAGEMENT AUTO;

CREATE USER SB_MBackUp
IDENTIFIED BY "%PWD%"
DEFAULT TABLESPACE ts_geo_denormalized;

GRANT CONNECT,RESOURCE TO SB_MBackUp;
GRANT SELECT ANY TABLE TO SB_MBackUp;

ALTER USER SB_MBackUp QUOTA UNLIMITED ON ts_geo_denormalized;
```

Script Output | Task completed in 0.183 seconds

```
TABLESPACE TS_GEO_DENORMALIZED created.

User SB_MBACKUP created.

Grant succeeded.

Grant succeeded.

User SB_MBACKUP altered.
```

Create the table data_geo using left join.

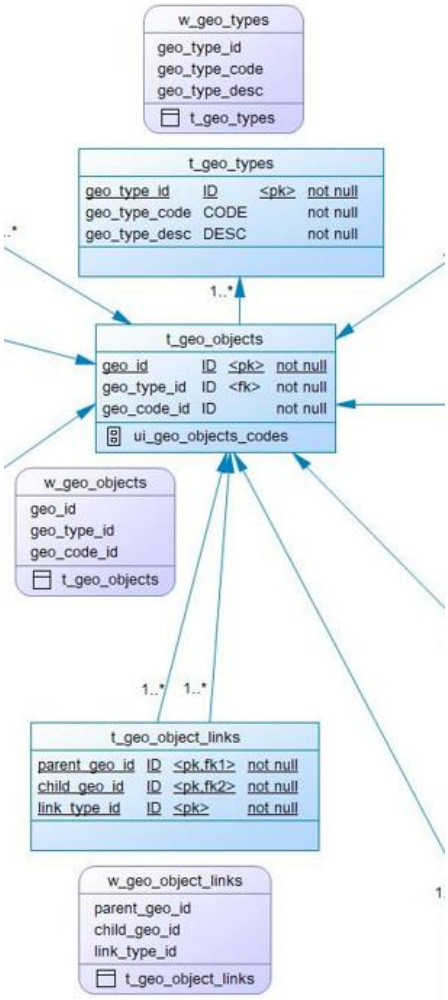


```
create table data_geo AS
SELECT
    tgo.geo_id, tgo.geo_type_id, tgo.geo_code_id,
    tgt.geo_type_code, tgt.geo_type_desc,
    lcgs.grp_system_id, lcgs.grp_system_code, lcgs.grp_system_desc, lcgs.localizatio
    lcg.group_id, lcg.group_code, lcg.group_desc,
    lcsg.sub_group_id, lcsg.sub_group_code, lcsg.sub_group_desc,
    lc.country_id, lc.country_code_a2, lc.country_code_a3, lc.country_desc,
    lgr.region_id, lgr.region_code, lgr.region_desc,
    lgp.part_id, lgp.part_code, lgp.part_desc,
    lgs.geo_system_id, lgs.geo_system_code, lgs.geo_system_desc,
    tgol.parent_geo_id, tgol.child_geo_id, tgol.link_type_id
from
    t_geo_objects tgo
    LEFT OUTER JOIN t_geo_types tgt ON tgo.geo_type_id = tgt.geo_type_id
    LEFT OUTER JOIN t_geo_object_links tgol ON tgo.geo_id = tgol.parent_geo_id
    LEFT OUTER JOIN lc_cntr_group_systems lcgs ON tgo.geo_id = lcgs.geo_id
    LEFT OUTER JOIN lc_cntr_groups lcg ON tgo.geo_id = lcg.geo_id
    LEFT OUTER JOIN lc_cntr_sub_groups lcsg ON tgo.geo_id = lcsg.geo_id
    LEFT OUTER JOIN lc_countries lc ON tgo.geo_id = lc.geo_id
    LEFT OUTER JOIN lc_geo_regions lgr ON tgo.geo_id = lgr.geo_id
    LEFT OUTER JOIN lc_geo_parts lgp ON tgo.geo_id = lgp.geo_id
    LEFT OUTER JOIN lc_geo_systems lgs ON tgo.geo_id = lgs.geo_id;
```

Script Output | Task completed in 0.132 seconds

```
Session altered.

Table DATA_GEO created.
```

As we can see, the table t_geo_object_links has parent_geo_id and child_geo_id.

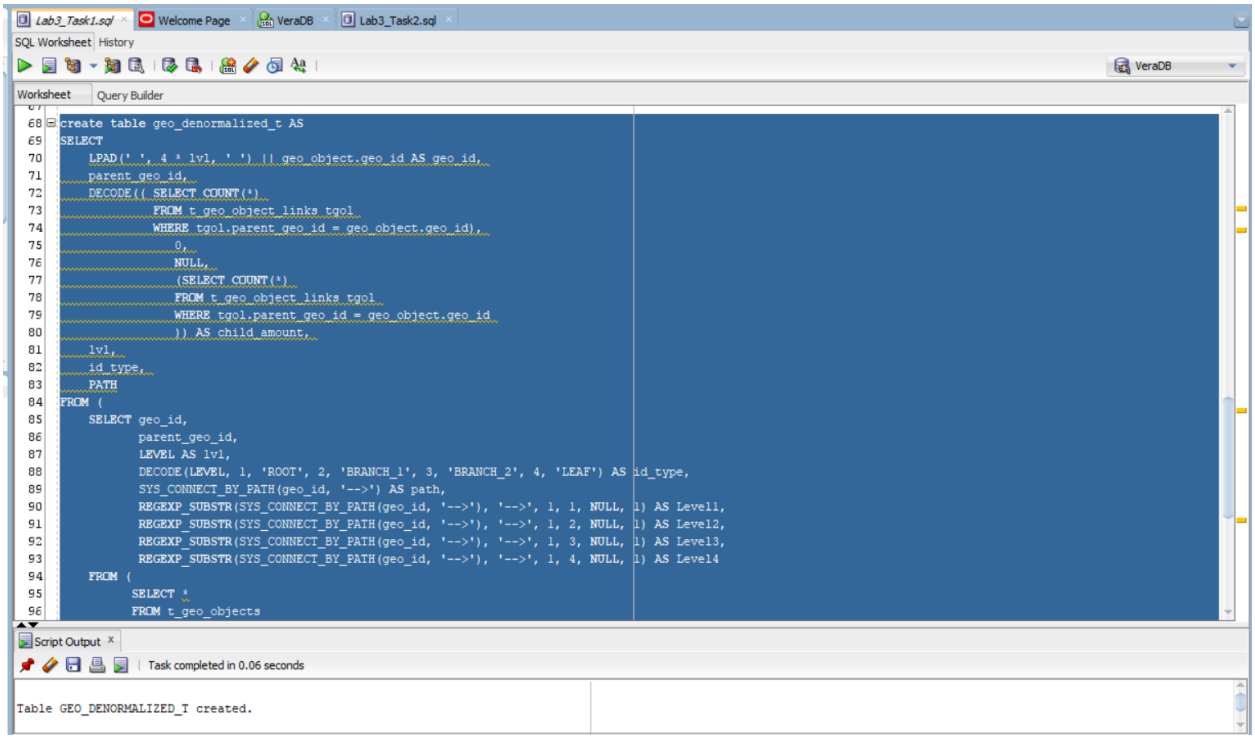



```
SELECT
    geo_id, geo_type_id, geo_code_id,
    geo_type_code, geo_type_desc,
    parent_geo_id, child_geo_id, link_type_id
FROM
    data_geo;
```

Script Output | Query Result | SQL | Fetched 50 rows in 0.01 seconds

	GEO_ID	GEO_TYPE_ID	GEO_CODE_ID	GEO_TYPE_CODE	GEO_TYPE_DESC	PARENT_GEO_ID	CHILD_GEO_ID	LINK_TYPE_ID
1	247	10	9	CONTINENT	Referene: List of All Continets	247	249	2
2	247	10	9	CONTINENT	Referene: List of All Continets	247	256	2
3	247	10	9	CONTINENT	Referene: List of All Continets	247	265	2
4	247	10	9	CONTINENT	Referene: List of All Continets	247	270	2
5	245	10	21	CONTINENT	Referene: List of All Continets	245	260	2
6	244	10	2	CONTINENT	Referene: List of All Continets	244	250	2
7	244	10	2	CONTINENT	Referene: List of All Continets	244	251	2
8	244	10	2	CONTINENT	Referene: List of All Continets	244	254	2
9	244	10	2	CONTINENT	Referene: List of All Continets	244	259	2
10	244	10	2	CONTINENT	Referene: List of All Continets	244	262	2
11	248	10	150	CONTINENT	Referene: List of All Continets	248	258	2
12	248	10	150	CONTINENT	Referene: List of All Continets	248	264	2
13	248	10	150	CONTINENT	Referene: List of All Continets	248	268	2
14	248	10	150	CONTINENT	Referene: List of All Continets	248	269	2
15	246	10	142	CONTINENT	Referene: List of All Continets	246	252	2
16	246	10	142	CONTINENT	Referene: List of All Continets	246	253	2
17	246	10	142	CONTINENT	Referene: List of All Continets	246	255	2
18	246	10	142	CONTINENT	Referene: List of All Continets	246	257	2

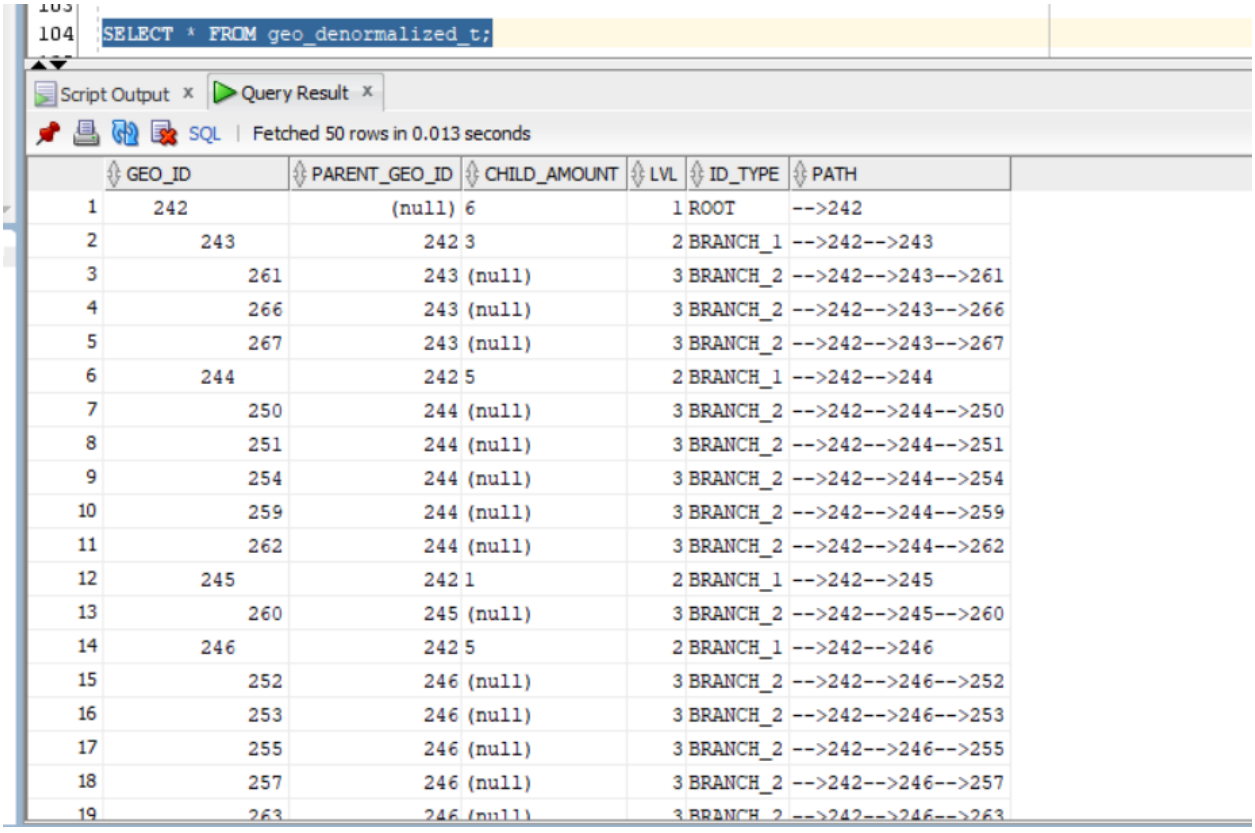
Let's create a hierarchy for these tables.



```
create table geo_denormalized_t AS
SELECT
    LPAD(' ', 4 * lvl, ' ') || geo_object.geo_id AS geo_id,
    parent_geo_id,
    DECODE(( SELECT COUNT(*)
             FROM t_geo_object_links tgol
             WHERE tgol.parent_geo_id = geo_object.geo_id),
              0,
              NULL,
              (SELECT COUNT(*)
               FROM t_geo_object_links tgol
               WHERE tgol.parent_geo_id = geo_object.geo_id
               )) AS child_amount,
    lvl,
    id_type,
    PATH
FROM (
    SELECT geo_id,
           parent_geo_id,
           LEVEL AS lvl,
           DECODE(LEVEL, 1, 'ROOT', 2, 'BRANCH_1', 3, 'BRANCH_2', 4, 'LEAF') AS id_type,
           SYS_CONNECT_BY_PATH(geo_id, '-->') AS path,
           REGEXP_SUBSTR(SYS_CONNECT_BY_PATH(geo_id, '-->'), '-->', 1, 1, NULL, 1) AS Level1,
           REGEXP_SUBSTR(SYS_CONNECT_BY_PATH(geo_id, '-->'), '-->', 1, 2, NULL, 1) AS Level2,
           REGEXP_SUBSTR(SYS_CONNECT_BY_PATH(geo_id, '-->'), '-->', 1, 3, NULL, 1) AS Level3,
           REGEXP_SUBSTR(SYS_CONNECT_BY_PATH(geo_id, '-->'), '-->', 1, 4, NULL, 1) AS Level4
    FROM (
        SELECT *
        FROM t_geo_objects
```

Script Output | Task completed in 0.06 seconds

```
Table GEO_DENORMALIZED_T created.
```



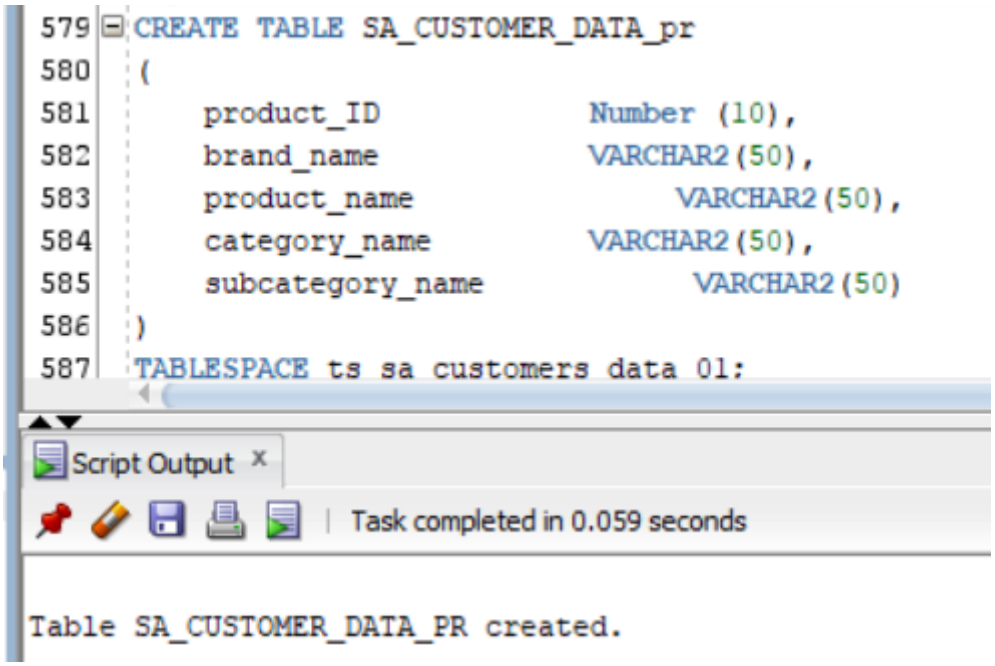
```
SELECT * FROM geo_denormalized_t;
```

Query Result | SQL | Fetched 50 rows in 0.013 seconds

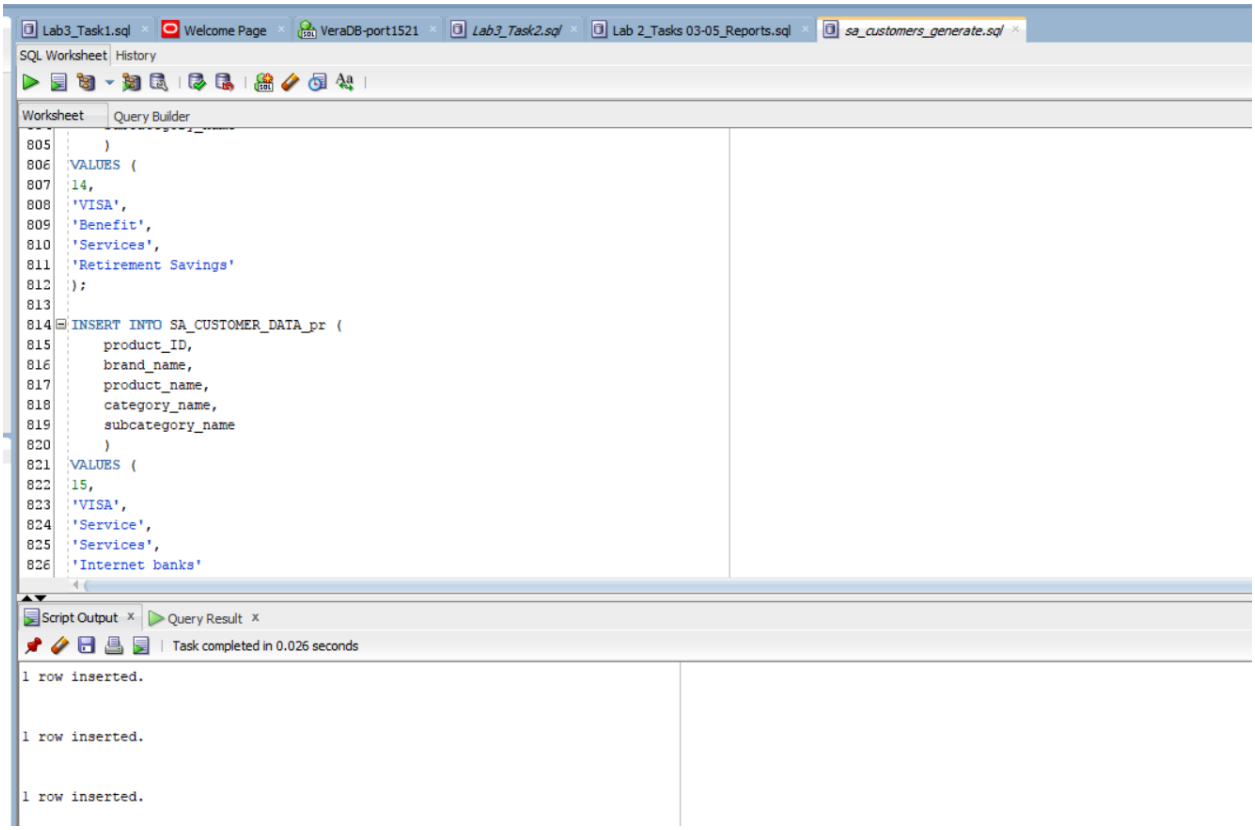
	GEO_ID	PARENT_GEO_ID	CHILD_AMOUNT	LVL	ID_TYPE	PATH
1	242	(null)	6	1	ROOT	-->242
2	243	242	3	2	BRANCH_1	-->242-->243
3	261	243	(null)	3	BRANCH_2	-->242-->243-->261
4	266	243	(null)	3	BRANCH_2	-->242-->243-->266
5	267	243	(null)	3	BRANCH_2	-->242-->243-->267
6	244	242	5	2	BRANCH_1	-->242-->244
7	250	244	(null)	3	BRANCH_2	-->242-->244-->250
8	251	244	(null)	3	BRANCH_2	-->242-->244-->251
9	254	244	(null)	3	BRANCH_2	-->242-->244-->254
10	259	244	(null)	3	BRANCH_2	-->242-->244-->259
11	262	244	(null)	3	BRANCH_2	-->242-->244-->262
12	245	242	1	2	BRANCH_1	-->242-->245
13	260	245	(null)	3	BRANCH_2	-->242-->245-->260
14	246	242	5	2	BRANCH_1	-->242-->246
15	252	246	(null)	3	BRANCH_2	-->242-->246-->252
16	253	246	(null)	3	BRANCH_2	-->242-->246-->253
17	255	246	(null)	3	BRANCH_2	-->242-->246-->255
18	257	246	(null)	3	BRANCH_2	-->242-->246-->257
19	263	246	(null)	3	BRANCH_2	-->242-->246-->263

2.2. Task 02: Analyze Business hierarch Reference Analyses

Create table SA_CUSTOMER_DATA_pr on the tablespace ts_sa_customers_data_01.



Insert values into the table SA_CUSTOMER_DATA_pr.



```
829 select * from sa_customer_data_pr;
```

Script Output | Query Result | SQL | All Rows Fetched: 16 in 0.01 seconds

	PRODUCT_ID	BRAND_NAME	PRODUCT_NAME	CATEGORY_NAME	SUBCATEGORY_NAME
8	8	SAMSUNG	Smart watches	Computerized wristwatch	Smart watch for kids
9	9	SAMSUNG	TVs	Appliances	OLED
10	10	SAMSUNG	Smartphones	Phones	Android Phones
11	11	SAMSUNG	Headphones	Accessories	Earbuds
12	12	SAMSUNG	Tablets	Tablet computer	Galaxy Tab series
13	13	VISA	Card	Services	Contactless smart card
14	14	VISA	Benefit	Services	Retirement Savings

Let's join 2 tables SA_CUSTOMER_DATA_total and sa_customer_data_pr using a left join by product_name in the ts_sa_customers_data_01 tablespace.

```
834
835 select * from SA_CUSTOMER_DATA_total left outer join sa_customer_data_pr on SA_CUSTOMER_DATA_total.product_name=sa_customer_data_pr.product_name
836 order by 1;
```

Script Output | Query Result | SQL | Fetched 150 rows in 0.396 seconds

	CY_MOBILE_PHONE	AGENCY_FEE_PERCENT	AGENCY_VAT_PERCENT	PROMOTION_MEDIA_TYPE	PRODUCT_ID	BRAND_NAME_1	PRODUCT_NAME_1	CATEGORY_NAME	SUBCATEGORY_NAME
25	493-09-99	8	23	TV	2	SAMSUNG	Ovens	Appliances	Conventional ovens
26	-96-88-00	9	21	TV	2	SAMSUNG	Ovens	Appliances	Conventional ovens
27	66-798-2	2	22	TV	2	SAMSUNG	Ovens	Appliances	Conventional ovens
28	4-367-6411	4	5	TV	2	SAMSUNG	Ovens	Appliances	Conventional ovens
29	969-2011	7	20	TV	2	SAMSUNG	Ovens	Appliances	Conventional ovens
30	-577-43-17	5	20	TV	2	SAMSUNG	Ovens	Appliances	Conventional ovens
31	468-3588	5	15	TV	3	SAMSUNG	Cooking ovens	Appliances	Electric cooktop
32	66-8010	1	25.8	TV	3	SAMSUNG	Cooking ovens	Appliances	Electric cooktop
33	8-74-78-66	2	20	TV	3	SAMSUNG	Cooking ovens	Appliances	Electric cooktop
34	366-66-66	2	7.7	TV	3	SAMSUNG	Cooking ovens	Appliances	Electric cooktop

Create the business_hierarch table.

```
16 --DROP TABLE business_hierarch
17 Create table business_hierarch as
18 select SA_CUSTOMER_DATA_total.total_ID, SA_CUSTOMER_DATA_total.brand_name, SA_CUSTOMER_DATA_total.product_name, sa_customer_data_pr.category_name, sa_customer_dat
19 order by 1;
```

Script Output | Task completed in 0.599 seconds

```
Table BUSINESS_HIERARCH created.
```

```
20
21 select * from business_hierarch;
```

Script Output | Query Result | SQL | Fetched 50 rows in 0.016 seconds

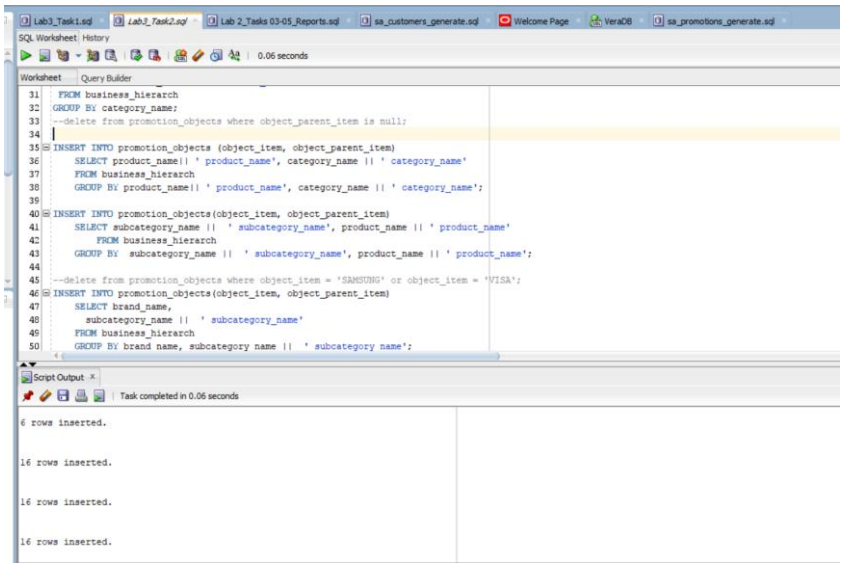
	TOTAL_ID	BRAND_NAME	PRODUCT_NAME	CATEGORY_NAME	SUBCATEGORY_NAME
10	10	SAMSUNG	Refrigerators	Appliances	Fridge
11	11	SAMSUNG	Refrigerators	Appliances	Fridge
12	12	SAMSUNG	Refrigerators	Appliances	Fridge
13	13	SAMSUNG	Refrigerators	Appliances	Fridge
14	14	SAMSUNG	Refrigerators	Appliances	Fridge
15	15	SAMSUNG	Refrigerators	Appliances	Fridge
16	16	SAMSUNG	Ovens	Appliances	Conventional ovens
17	17	SAMSUNG	Ovens	Appliances	Conventional ovens
18	18	SAMSUNG	Ovens	Appliances	Conventional ovens
19	19	SAMSUNG	Ovens	Appliances	Conventional ovens
20	20	SAMSUNG	Ovens	Appliances	Conventional ovens
21	21	SAMSUNG	Ovens	Appliances	Conventional ovens
22	22	SAMSUNG	Ovens	Appliances	Conventional ovens

Create the table promotion_objects with parent and child values.

```
24  CREATE TABLE promotion_objects(
25      object_item VARCHAR2(100),
26      object_parent_item VARCHAR2(100)
27  ) TABLESPACE ts_sa_customers_data_01;
```

Script Output | Task completed in 0.038 seconds

```
Table PROMOTION_OBJECTS created.
```



```
31    FROM business_hierarch
32  GROUP BY category_name;
33  --delete from promotion_objects where object_parent_item is null;
34
35  INSERT INTO promotion_objects (object_item, object_parent_item)
36      SELECT product_name|| ' product_name', category_name || ' category_name'
37      FROM business_hierarch
38      GROUP BY product_name|| ' product_name', category_name || ' category_name';
39
40  INSERT INTO promotion_objects(object_item, object_parent_item)
41      SELECT subcategory_name ||  ' subcategory_name', product_name || ' product_name'
42          FROM business_hierarch
43      GROUP BY  subcategory_name ||  ' subcategory_name', product_name || ' product_name';
44
45  --delete from promotion_objects where object_item = 'SAMSUNG' or object_item = 'VISA';
46  INSERT INTO promotion_objects(object_item, object_parent_item)
47      SELECT brand_name,
48        subcategory_name ||  ' subcategory_name'
49      FROM business_hierarch
50      GROUP BY brand_name, subcategory_name ||  ' subcategory_name';
```

Script Output | Task completed in 0.06 seconds

```
6 rows inserted.

16 rows inserted.

16 rows inserted.

16 rows inserted.
```

```
51
52  SELECT * FROM promotion_objects;
53
```

Script Output | Query Result | SQL | All Rows Fetched: 54 in 0.022 seconds

	OBJECT_ITEM	OBJECT_PARENT_ITEM
1	Appliances category_name	(null)
2	Computerized wristwatch category_name	(null)
3	Phones category_name	(null)
4	Accessories category_name	(null)
5	Tablet computer category_name	(null)
6	Services category_name	(null)
7	Refrigerators product_name	Appliances category_name
8	Ovens product_name	Appliances category_name
9	Cooking ovens product_name	Appliances category_name

Let's create a table showing the hierarchy by category of products participating in promotions.

```
CREATE TABLE t_levels_by_category
AS
    SELECT LPAD(' ',4* LEVEL, ' ') || object_item AS levels_by_category,
        DECODE(( SELECT COUNT(*)FROM promotion_objects t1 WHERE t1.object_parent_item = t2.object_item),0,NULL,
                ( SELECT COUNT(*) FROM promotion_objects t1 WHERE t1.object_parent_item = t2.object_item)) AS count_child,
        DECODE(LEVEL, 1, 'ROOT', 4, 'LEAF', 'BRANCH') AS id_type,
        LEVEL AS lvl,
        CONNECT_BY_ROOT(object_item) AS root,
        SYS_CONNECT_BY_PATH(object_item, '-->') AS path,
        REGEXP_SUBSTR(SYS_CONNECT_BY_PATH(object_item, '-->'), '-->([^(-->)]+)', 1, 1, NULL, 1) AS Level1,
        REGEXP_SUBSTR(SYS_CONNECT_BY_PATH(object_item, '-->'), '-->([^(-->)]+)', 1, 2, NULL, 1) AS Level2,
        REGEXP_SUBSTR(SYS_CONNECT_BY_PATH(object_item, '-->'), '-->([^(-->)]+)', 1, 3, NULL, 1) AS Level3,
        REGEXP_SUBSTR(SYS_CONNECT_BY_PATH(object_item, '-->'), '-->([^(-->)]+)', 1, 4, NULL, 1) AS Level4
    FROM promotion_objects t2
    START WITH object_parent_item IS NULL
    CONNECT BY PRIOR object_item = object_parent_item
    ORDER SIBLINGS BY object_item;
```

Script Output | Task completed in 0.152 seconds

```
Table T_LEVELS_BY_CATEGORY created.
```

```
SELECT * FROM t_levels_by_category;
```

Script Output | Query Result | SQL | Fetched 50 rows in 0.015 seconds

	LEVELS_BY_CATEGORY	COUNT_CHILD	ID_TYPE	LVL	ROOT	PATH
1	Accessories category_name	1	ROOT	1	Accessories category_name	-->Accessories category_name
2	Headphones product_name	1	BRANCH	2	Accessories category_name	-->Accessories category_name-->Headphor
3	Earbuds subcategory_name	1	BRANCH	3	Accessories category_name	-->Accessories category_name-->Headphor
4	SAMSUNG	(null)	LEAF	4	Accessories category_name	-->Accessories category_name-->Headphor
5	Appliances category_name	8	ROOT	1	Appliances category_name	-->Appliances category_name
6	Cooking ovens product_name	1	BRANCH	2	Appliances category_name	-->Appliances category_name-->Cooking c
7	Electric cooktop subcategory_name	1	BRANCH	3	Appliances category_name	-->Appliances category_name-->Cooking c
8	SAMSUNG	(null)	LEAF	4	Appliances category_name	-->Appliances category_name-->Cooking c
9	Dishwashers product_name	1	BRANCH	2	Appliances category_name	-->Appliances category_name-->Dishwashe
10	Built-in Dishwasher subcategory_name	1	BRANCH	3	Appliances category_name	-->Appliances category_name-->Dishwashe