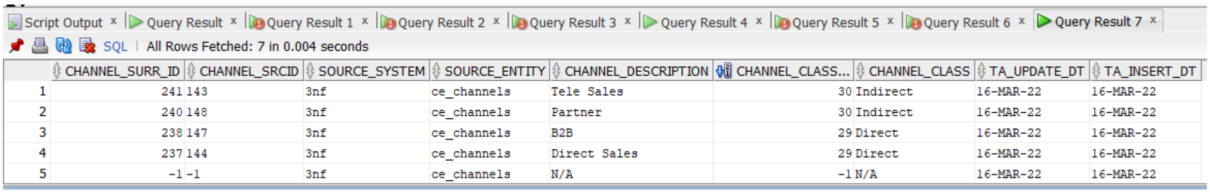
TASK 8 REPORT

1. CREATE PACKAGES TO LOAD DIMENSIONS FROM TO BL_DM LAYER

I CREATED ALL REQUIRED OBJECTS ON BL_DM LAYER ,GRANT ALL REQUIRED PRIVILEGES TO BL_CL, CREATED STRUCTURE OF PACKAGE AND REALIZED SOME OF PROCEDURES:

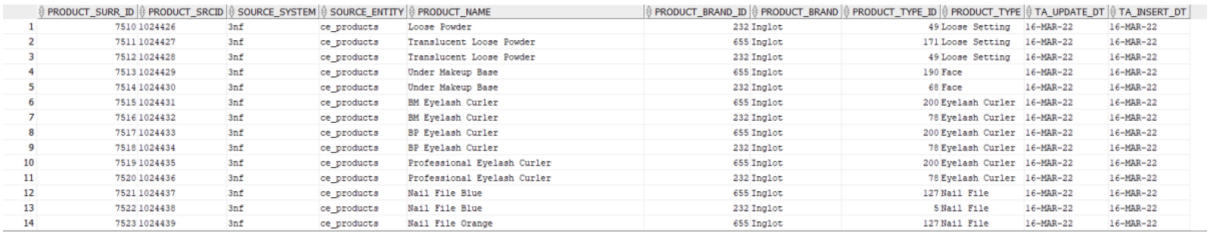
PROCEDURE PRC_LOAD_DIM_CHANNELS;



All Rows Fetched: 7 in 0.004 seconds

	CHANNEL_SURR_ID	CHANNEL_SRCID	SOURCE_SYSTEM	SOURCE_ENTITY	CHANNEL_DESCRIPTION	CHANNEL_CLASS...	CHANNEL_CLASS	TA_UPDATE_DT	TA_INSERT_DT
1	241	143	3nf	ce_channels	Tele Sales	30	Indirect	16-MAR-22	16-MAR-22
2	240	148	3nf	ce_channels	Partner	30	Indirect	16-MAR-22	16-MAR-22
3	238	147	3nf	ce_channels	B2B	29	Direct	16-MAR-22	16-MAR-22
4	237	144	3nf	ce_channels	Direct Sales	29	Direct	16-MAR-22	16-MAR-22
5	-1	-1	3nf	ce_channels	N/A	-1	N/A	16-MAR-22	16-MAR-22

PROCEDURE PRC_LOAD_DIM_PRODUCTS;

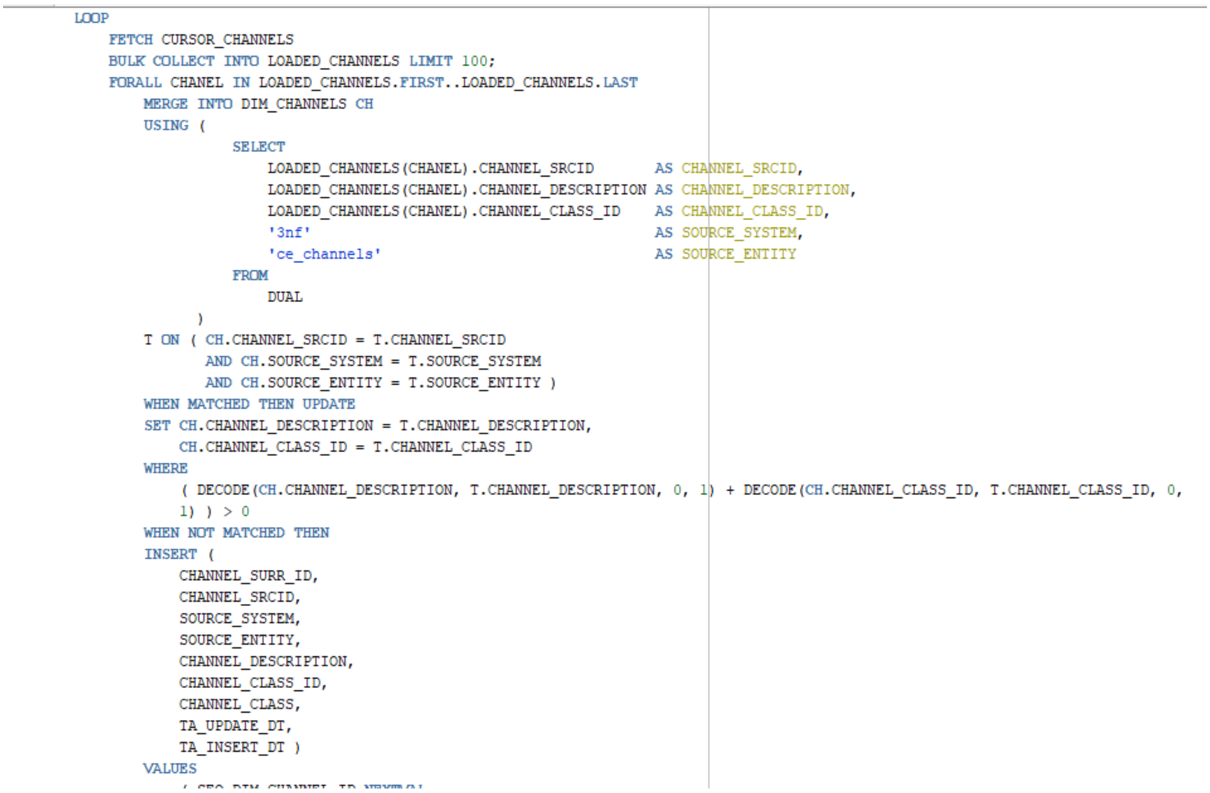


	PRODUCT_SURR_ID	PRODUCT_SRCID	SOURCE_SYSTEM	SOURCE_ENTITY	PRODUCT_NAME	PRODUCT_BRAND_ID	PRODUCT_BRAND	PRODUCT_TYPE_ID	PRODUCT_TYPE	TA_UPDATE_DT	TA_INSERT_DT
1	7510	1024426	3nf	ce_products	Loose Powder	232	Inglot	49	Loose Setting	16-MAR-22	16-MAR-22
2	7511	1024427	3nf	ce_products	Translucent Loose Powder	655	Inglot	171	Loose Setting	16-MAR-22	16-MAR-22
3	7512	1024428	3nf	ce_products	Translucent Loose Powder	232	Inglot	49	Loose Setting	16-MAR-22	16-MAR-22
4	7513	1024429	3nf	ce_products	Under Makeup Base	655	Inglot	190	Face	16-MAR-22	16-MAR-22
5	7514	1024430	3nf	ce_products	Under Makeup Base	232	Inglot	68	Face	16-MAR-22	16-MAR-22
6	7515	1024431	3nf	ce_products	BM Eyelash Curler	655	Inglot	200	Eyelash Curler	16-MAR-22	16-MAR-22
7	7516	1024432	3nf	ce_products	BM Eyelash Curler	232	Inglot	78	Eyelash Curler	16-MAR-22	16-MAR-22
8	7517	1024433	3nf	ce_products	BP Eyelash Curler	655	Inglot	200	Eyelash Curler	16-MAR-22	16-MAR-22
9	7518	1024434	3nf	ce_products	BP Eyelash Curler	232	Inglot	78	Eyelash Curler	16-MAR-22	16-MAR-22
10	7519	1024435	3nf	ce_products	Professional Eyelash Curler	655	Inglot	200	Eyelash Curler	16-MAR-22	16-MAR-22
11	7520	1024436	3nf	ce_products	Professional Eyelash Curler	232	Inglot	78	Eyelash Curler	16-MAR-22	16-MAR-22
12	7521	1024437	3nf	ce_products	Nail File Blue	655	Inglot	127	Nail File	16-MAR-22	16-MAR-22
13	7522	1024438	3nf	ce_products	Nail File Blue	232	Inglot	5	Nail File	16-MAR-22	16-MAR-22
14	7523	1024439	3nf	ce_products	Nail File Orange	655	Inglot	127	Nail File	16-MAR-22	16-MAR-22

I WILL CREATE ALL REST TABLES IN NEXT COMMIT

USE BULK COLLECT AND FORALL

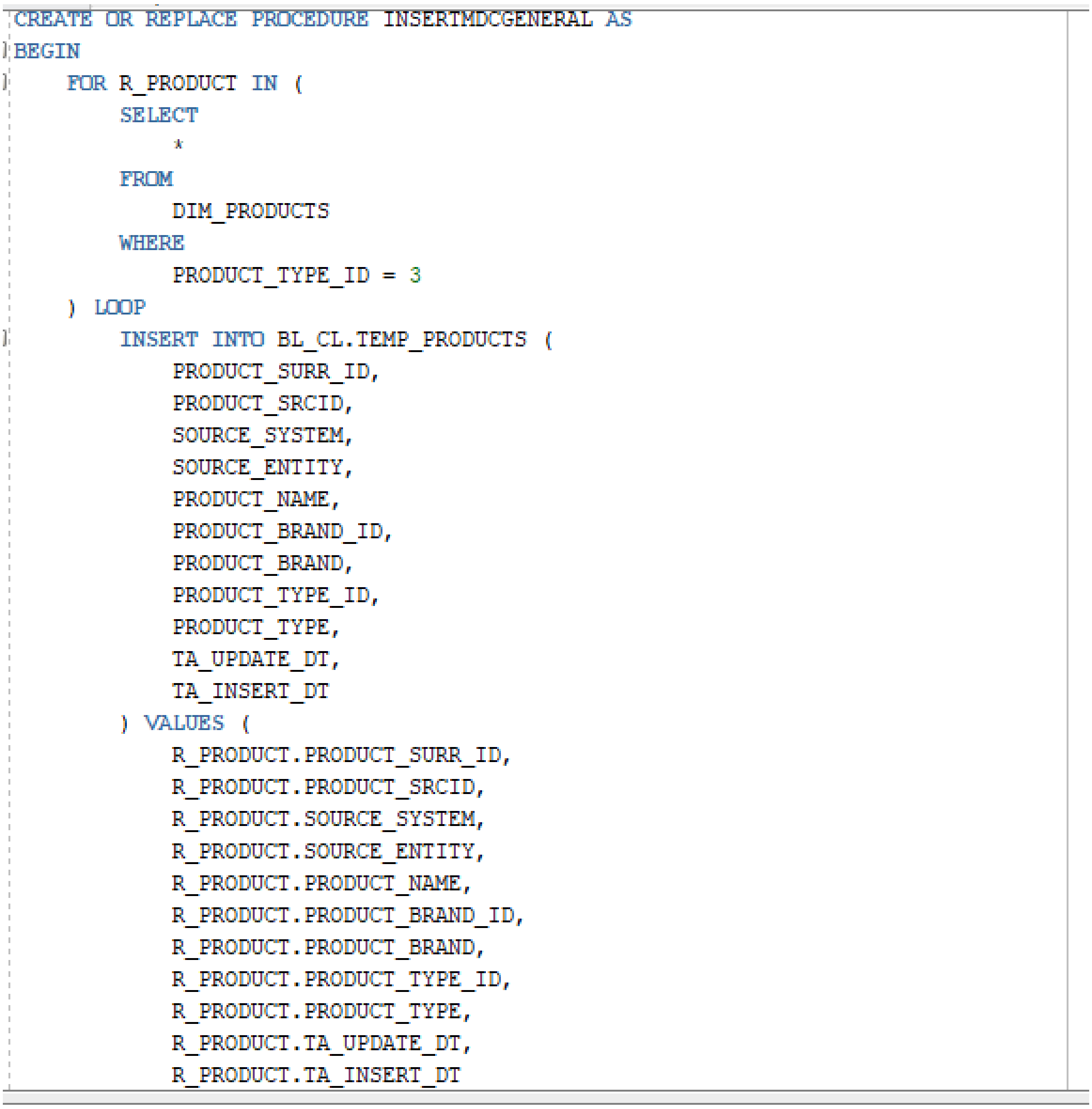
I USED BULK COLLECT AND FORALL IN PROCEDURE PRC_LOAD_DIM_CHANNELS;



```
LOOP
    FETCH CURSOR_CHANNELS
    BULK COLLECT INTO LOADED_CHANNELS LIMIT 100;
    FORALL CHANEL IN LOADED_CHANNELS.FIRST..LOADED_CHANNELS.LAST
        MERGE INTO DIM_CHANNELS CH
        USING (
                  SELECT
                      LOADED_CHANNELS(CHANEL).CHANNEL_SRCID       AS CHANNEL_SRCID,
                      LOADED_CHANNELS(CHANEL).CHANNEL_DESCRIPTION AS CHANNEL_DESCRIPTION,
                      LOADED_CHANNELS(CHANEL).CHANNEL_CLASS_ID    AS CHANNEL_CLASS_ID,
                      '3nf'                                       AS SOURCE_SYSTEM,
                      'ce_channels'                               AS SOURCE_ENTITY
                  FROM
                      DUAL
              )
        T ON ( CH.CHANNEL_SRCID = T.CHANNEL_SRCID
               AND CH.SOURCE_SYSTEM = T.SOURCE_SYSTEM
               AND CH.SOURCE_ENTITY = T.SOURCE_ENTITY )
        WHEN MATCHED THEN UPDATE
        SET CH.CHANNEL_DESCRIPTION = T.CHANNEL_DESCRIPTION,
            CH.CHANNEL_CLASS_ID = T.CHANNEL_CLASS_ID
        WHERE
            ( DECODE(CH.CHANNEL_DESCRIPTION, T.CHANNEL_DESCRIPTION, 0, 1) + DECODE(CH.CHANNEL_CLASS_ID, T.CHANNEL_CLASS_ID, 0,
            1) ) > 0
        WHEN NOT MATCHED THEN
        INSERT (
            CHANNEL_SURR_ID,
            CHANNEL_SRCID,
            SOURCE_SYSTEM,
            SOURCE_ENTITY,
            CHANNEL_DESCRIPTION,
            CHANNEL_CLASS_ID,
            CHANNEL_CLASS,
            TA_UPDATE_DT,
            TA_INSERT_DT )
        VALUES
            ( SEQ_DIM_CHANNEL_ID.NEXTVAL
```

CURSOR FOR LOOP AND CURSOR VARIABLES

I USED IT IN TEST PROCEDURE. YOU CAN SEE AN PART OF CODE EXAMPLE HERE:



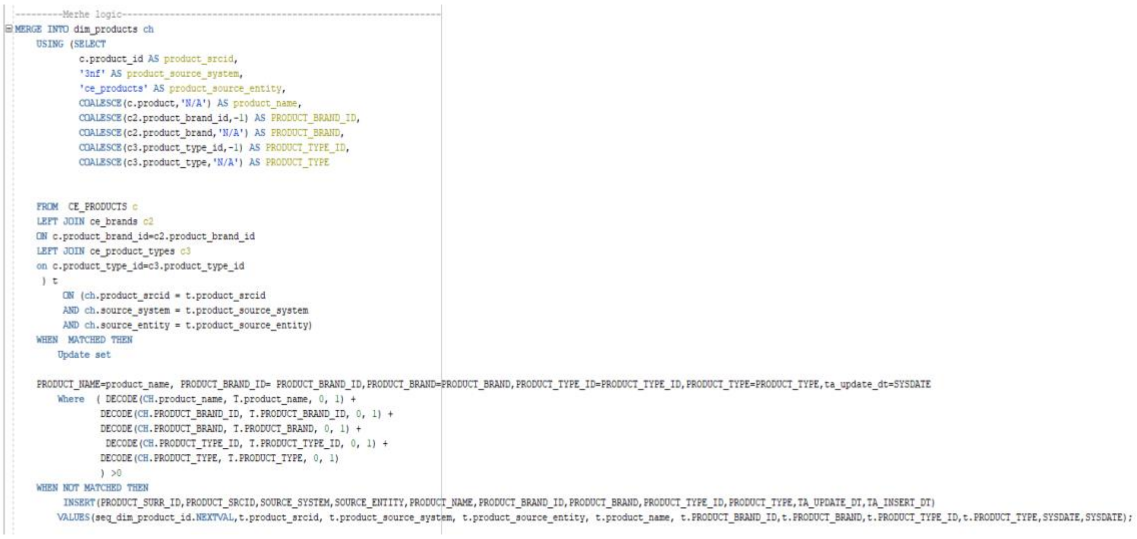
```
CREATE OR REPLACE PROCEDURE INSERTMDCGENERAL AS
BEGIN
    FOR R_PRODUCT IN (
        SELECT
            *
        FROM
            DIM_PRODUCTS
        WHERE
            PRODUCT_TYPE_ID = 3
    ) LOOP
        INSERT INTO BL_CL.TEMP_PRODUCTS (
            PRODUCT_SURR_ID,
            PRODUCT_SRCID,
            SOURCE_SYSTEM,
            SOURCE_ENTITY,
            PRODUCT_NAME,
            PRODUCT_BRAND_ID,
            PRODUCT_BRAND,
            PRODUCT_TYPE_ID,
            PRODUCT_TYPE,
            TA_UPDATE_DT,
            TA_INSERT_DT
        ) VALUES (
            R_PRODUCT.PRODUCT_SURR_ID,
            R_PRODUCT.PRODUCT_SRCID,
            R_PRODUCT.SOURCE_SYSTEM,
            R_PRODUCT.SOURCE_ENTITY,
            R_PRODUCT.PRODUCT_NAME,
            R_PRODUCT.PRODUCT_BRAND_ID,
            R_PRODUCT.PRODUCT_BRAND,
            R_PRODUCT.PRODUCT_TYPE_ID,
            R_PRODUCT.PRODUCT_TYPE,
            R_PRODUCT.TA_UPDATE_DT,
            R_PRODUCT.TA_INSERT_DT
```

EXECUTE IMMEDIATE INTO FOR DYNAMIC QUERIES

I SOME PROCEDURES I USED: EXECUTE IMMEDIATE 'TRUNCATE TABLE BL_CL.MAP_CHANNELS';

USE MERGE

I USED MERGE ALMOS IN ALL PROCEDURES. YOU CAN SEE AN EXAMPLE HERE.



```
---------Merhe logic-----------------------------------------------------------
MERGE INTO dim_products ch
    USING (SELECT
            c.product_id AS product_srcid,
            '3nf' AS product_source_system,
            'ce_products' AS product_source_entity,
            COALESCE(c.product,'N/A') AS product_name,
            COALESCE(c2.product_brand_id,-1) AS PRODUCT_BRAND_ID,
            COALESCE(c2.product_brand,'N/A') AS PRODUCT_BRAND,
            COALESCE(c3.product_type_id,-1) AS PRODUCT_TYPE_ID,
            COALESCE(c3.product_type,'N/A') AS PRODUCT_TYPE


    FROM  CE_PRODUCTS c
    LEFT JOIN ce_brands c2
    ON c.product_brand_id=c2.product_brand_id
    LEFT JOIN ce_product_types c3
    on c.product_type_id=c3.product_type_id
     ) t
         ON (ch.product_srcid = t.product_srcid
         AND ch.source_system = t.product_source_system
         AND ch.source_entity = t.product_source_entity)
    WHEN  MATCHED THEN
        Update set

    PRODUCT_NAME=product_name, PRODUCT_BRAND_ID= PRODUCT_BRAND_ID,PRODUCT_BRAND=PRODUCT_BRAND,PRODUCT_TYPE_ID=PRODUCT_TYPE_ID,PRODUCT_TYPE=PRODUCT_TYPE,ta_update_dt=SYSDATE
        Where  ( DECODE(CH.product_name, T.product_name, 0, 1) +
                DECODE(CH.PRODUCT_BRAND_ID, T.PRODUCT_BRAND_ID, 0, 1) +
                DECODE(CH.PRODUCT_BRAND, T.PRODUCT_BRAND, 0, 1) +
                 DECODE(CH.PRODUCT_TYPE_ID, T.PRODUCT_TYPE_ID, 0, 1) +
                DECODE(CH.PRODUCT_TYPE, T.PRODUCT_TYPE, 0, 1)
                ) >0
    WHEN NOT MATCHED THEN
         INSERT(PRODUCT_SURR_ID,PRODUCT_SRCID,SOURCE_SYSTEM,SOURCE_ENTITY,PRODUCT_NAME,PRODUCT_BRAND_ID,PRODUCT_BRAND,PRODUCT_TYPE_ID,PRODUCT_TYPE,TA_UPDATE_DT,TA_INSERT_DT)
        VALUES(seq_dim_product_id.NEXTVAL,t.product_srcid, t.product_source_system, t.product_source_entity, t.product_name, t.PRODUCT_BRAND_ID,t.PRODUCT_BRAND,t.PRODUCT_TYPE_ID,t.PRODUCT_TYPE,SYSDATE,SYSDATE);
```