

Anton Slizh's

U2M9.LW.ETL Overview - Transportation

GitHub: <https://github.com/drapejny/DataCamp2022>

Task 1

2.1. Task 01: Transportation Description

The transportation method is highly depends on the source system being used. The most common and efficient way to transfer data is to use flat files and mechanism such FTP or other remote files system access protocols. Data is unloaded or exported from the source system into flat files and then transporting to the target platform using FTP or similar mechanism.

Because source systems and data warehouses often use different operating systems and database systems, using flat files is often the simplest way to exchange data between heterogeneous systems with minimal transformations. However, even when transporting data between homogeneous systems, flat files are often the most efficient and most easy-to-manage mechanism for data transfer.

Task 2

3.1. Task 02: Prepare Table of Facts to DW Layer

I have already created the package for load my fact table in Lab 4. Let's look at the loading fact table (*fct_sales*) again.

Defining procedure and variables:

```

CREATE OR REPLACE PACKAGE BODY pkg_load_sales
IS
    PROCEDURE load_sales
    IS
        TYPE sales_rows_t IS TABLE OF dw_data.fct_sales%ROWTYPE;

        sales sales_rows_t;

        CURSOR c IS
            SELECT l,
                   cl.date_id,
                   pr.product_id,
                   cu.customer_id,
                   st.store_id,
                   geo.geo_id,
                   cl.amount,
                   cl.pos_transaction
            FROM dw_cl.dw_cl_sale_data cl
            JOIN dw_data.dim_products_scd pr
            ON cl.sku_num = pr.sku_num AND pr.exp_time IS NULL
            JOIN dw_data.dim_customers cu
            ON cl.phone = cu.phone
            JOIN dw_data.dim_stores st
            ON cl.store_address = st.address
            JOIN dw_data.dim_geo_locations geo
            ON cl.country = geo.country_desc;

```

I have used the cursor to iterate throw the data. The cursor was created as select statement on the sales data from cleansing layer and joined data from dimensions (just to convert natural keys to surrogate).

The procedure body contains bulk collecting to the sales variable and further bulk insertion into fact table.

```

BEGIN
    OPEN c;
    LOOP
        FETCH c
        BULK COLLECT INTO sales;
        FORALL i in 1 .. sales.COUNT()
            INSERT INTO dw_data.fct_sales
                (
                    sale_id,
                    date_id,
                    product_id,
                    customer_id,
                    store_id,
                    geo_id,
                    amount,
                    pos_transaction
                )
            VALUES
                (
                    seq_sales.NEXTVAL,
                    sales(i).date_id,
                    sales(i).product_id,
                    sales(i).customer_id,
                    sales(i).store_id,
                    sales(i).geo_id,
                    sales(i).amount,
                    sales(i).pos_transaction
                );
        EXIT WHEN c%NOTFOUND;
    END LOOP;
    CLOSE c;

    COMMIT;

```

The load sales procedure executing after updating data in the dimensions:

```

BEGIN
    pkg_load_dates.load_dates;
    pkg_load_geo_locations.load_geo_locations;
    pkg_load_products.load_products;
    pkg_load_stores.load_stores;
    pkg_load_customers.load_customers;
    pkg_load_sales.load_sales;
END;

```

Result data:

	SALE_ID	DATE_ID	PRODUCT_ID	CUSTOMER_ID	STORE_ID	GEO_ID	AMOUNT	POS_TRANSACTION
1	62623568	02.02.22	2769955	57783	306	412	3	20220202560201
2	62623569	24.04.21	2769955	57783	306	412	1	20210424458301
3	62623570	02.05.21	2769977	57783	306	412	2	20210502566001
4	62623571	31.03.21	2769977	57783	306	412	2	20210331457401
5	62623572	09.11.21	2769977	57783	306	412	1	20211109374501
6	62623573	04.01.21	2769977	57783	306	412	2	20210104524501
7	62623574	23.06.22	2769977	57783	306	412	2	20220623149701
8	62623575	04.01.22	2769977	57783	306	412	2	20220104528201
9	62623576	09.07.22	2769942	57783	306	412	1	20220709549001
10	62623577	01.05.21	2769942	57783	306	412	1	20210501234701
11	62623578	03.11.21	2769942	57783	306	412	3	20211103376201
12	62623579	28.01.22	2769942	57783	306	412	1	20220128355701
13	62623580	21.11.21	2769942	57783	306	412	2	20211121113001
14	62623581	17.03.22	2769942	57783	306	412	3	20220317371901
15	62623582	21.09.21	2769942	57783	306	412	1	20210921173801
16	62623583	22.06.22	2769942	57783	306	412	2	20220622368401
17	62623584	29.06.22	2769942	57783	306	412	1	20220629019601
18	62623585	11.11.21	2769955	57783	306	412	1	20211111300401
19	62623586	15.03.22	2769977	57783	306	412	2	20220315072101
20	62623587	12.07.21	2769977	57783	306	412	2	20210712195801
21	62623588	10.01.22	2769977	57783	306	412	3	20220110455501
22	62623589	26.02.21	2769977	57783	306	412	1	20210226114901
23	62623590	31.03.22	2769961	57783	306	412	2	20220331450801
24	62623591	19.02.22	2769961	57783	306	412	2	20220219059501
25	62623592	04.02.21	2769961	57783	306	412	2	20210204543101