

DOCUMENTATION

IICS PROJECT

- By Kshitij Chaube

Table of Contents:

DOCUMENTATION	1
IICS PROJECT	1
INTRODUCTION:.....	2
SCHEMA OVERVIEW:	3
OBJECTIVES:	3
PLAN:.....	4
1st Week:	4
2nd Week:.....	4
3rd Week - Final Week of the Project:	5
CHALLENGES:	5
Work Done:	6
Snowflake Procedures:	6
IICS Procedures:	10
Expression Transformer:.....	11
SUMMARY:.....	12
Conclusion:.....	13

My Services



Data Integration



Administrator



Monitor

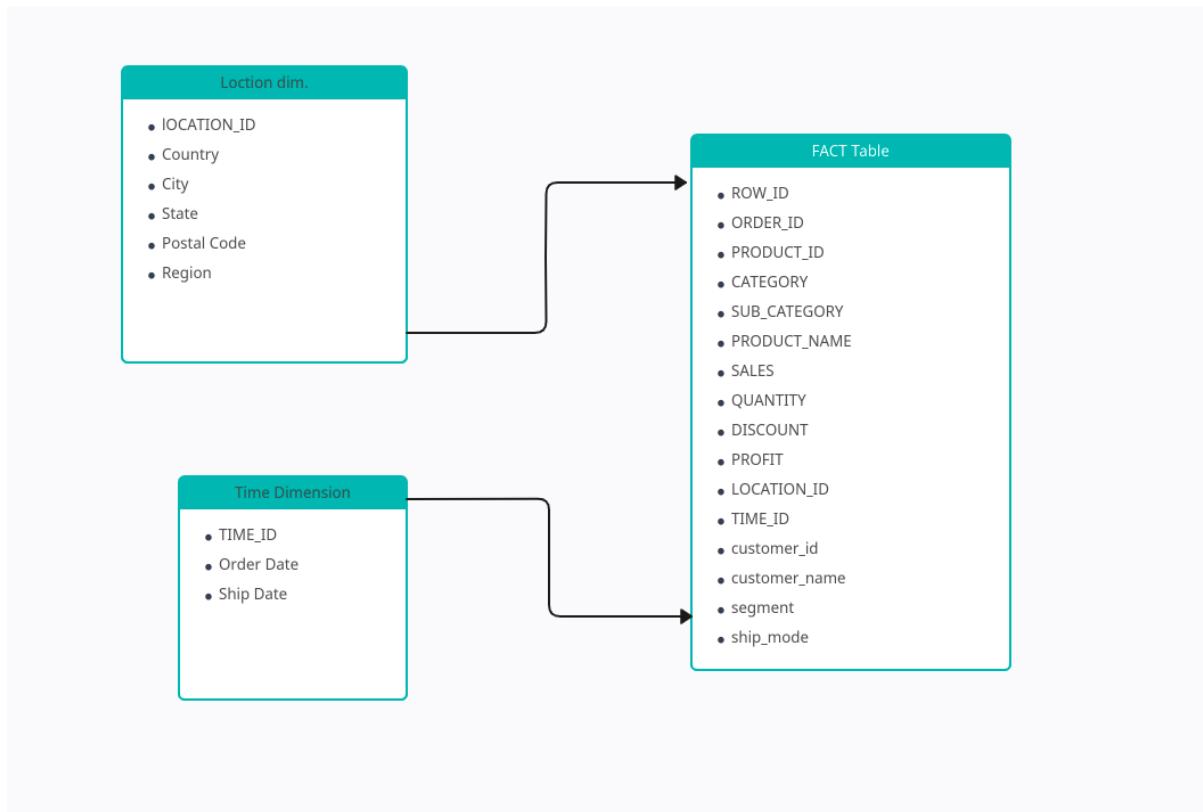
INTRODUCTION:

This project is all about managing and transforming data from the Superstore dataset. We're using Informatica Intelligent Cloud Services (IICS) for structuring the data into location, time dimensions, and a fact table. IICS also helps us handle insert and update flags, crucial for tracking changes in the data.

Our database is Snowflake, giving us a stable platform to build on. Lastly, we're creating a user-friendly Power BI dashboard to visualize and present the data.

Together, these tools and techniques will help us handle data efficiently and provide valuable insights.

SCHEMA OVERVIEW:



OBJECTIVES:

1. Efficiently manage and analyze data.
2. Utilize Snowflake and IICS for clean data processing.
3. Create a user-friendly Power BI dashboard for effective data visualization.
4. Support informed decision-making and streamline data management.

PLAN:

Project Timeline: 13 Days (Close to 3 Weeks)

1st Week:

In the initial week, starting from September 26th, we commenced our project journey. The following key tasks were accomplished:

- Receiving the dataset and creating accounts on Snowflake and IICS.
- Drafting preliminary schemas for both dimensions and facts tables.
- Uploading data into Snowflake, which involved troubleshooting errors.
- Connecting Snowflake and IICS, along with an initial exploration of IICS functionality.
- Establishing a staging area in IICS with stringent checks to prevent null values from passing through.

2nd Week:

The second week was marked by significant progress. Here's an overview of the key activities:

- Creation of fact and dimension tables within Snowflake.
- Designing mappings for loading data into these tables.
- Implementing a sequence generator to add primary keys in both dimension tables.
- Applying an update strategy through lookup and expression transformers.
- Exploring the concept of Slowly Changing Dimensions (SCD).
- Implementing a data-driven strategy for table inserts and updates.

3rd Week - Final Week of the Project:

In the last week of the project, we achieved significant milestones and addressed critical activities. Here is a concise summary of the key tasks and accomplishments during this period:

- Designed an indirect lookup table for both dimension tables, streamlining the data lookup process. Created a final mapping that inserts and updates data in all three tables, ensuring data integrity.
- Implemented an update flag in all three tables to enable tracking of data changes. Conducted testing to verify its functionality, although some challenges were encountered, especially with columns not present in their respective tables.
- Successfully addressed the flag problem by introducing a centralized expression transformer, allowing for consistent flag handling across the tables. This solution provided a more robust and uniform data management process.
- Continued the project by setting up Power BI integration and addressed any issues that arose during this process. Created an Initial Power BI dashboard with basic visualizations, providing a preliminary overview of data presentation and insights.
- Conducted thorough testing by inserting and updating rows, verifying the results within Power BI to ensure data accuracy and integrity.
- Refined and finalized the Power BI dashboard by incorporating various visualizations, enhancing the presentation of data insights.

CHALLENGES:

Some of the challenges I faced are as follows

- While loading data into snowflake it was unable to recognize \n in the end of each line this was fixed by using record delimiter in the worksheet as \n.
- Another one was when update flag was being added it wasn't working for all 3 dim tables at once this was fixed by adding a centralized expression transformer.
- I was attaching sequence transformer directly to target table I was having to use multiple sequence generator to generate same sequence and I was getting multiple error because of it. This was solved by using only one sequence generator for all tables.
- IICS automatically cuts length of its object to match it with length set in snowflake, because of this we were getting mixed flags, this was fixed by increasing the length of columns in snowflake.

Work Done:

Snowflake Procedures:

Step 1:

Loading table into dataset via csv file:

Create Table:

```
create or replace TABLE KOKO.HOP.SALESDATA (
ROW_ID NUMBER(38,0) NOT NULL,
ORDER_ID VARCHAR(255),
ORDER_DATE DATE,
SHIP_DATE DATE,
SHIP_MODE VARCHAR(255),
CUSTOMER_ID VARCHAR(255),
CUSTOMER_NAME VARCHAR(255),
SEGMENT VARCHAR(255),
COUNTRY VARCHAR(255),
CITY VARCHAR(255),
STATE VARCHAR(255),
POSTAL_CODE NUMBER(38,0),
REGION VARCHAR(80),
PRODUCT_ID VARCHAR(255),
CATEGORY VARCHAR(60),
```

```
SUB_CATEGORY VARCHAR(60),
PRODUCT_NAME VARCHAR(255),
SALES FLOAT,
QUANTITY NUMBER(38,0),
DISCOUNT FLOAT,
PROFIT FLOAT,
primary key (ROW_ID)
);
```

Inserting Data:

```
COPY INTO "MYDATA"."KOKO"."SALESDATA"
FROM '@"MYDATA"."KOKO"."%SALESDATA"/__snowflake_temp_import_files__/'
FILES = ('output.csv')
FILE_FORMAT = (
    TYPE=CSV,
    SKIP_HEADER=1,
    FIELD_DELIMITER=',',
    record_delimiter='\n'
    TRIM_SPACE=FALSE,
    FIELD_OPTIONALLY_ENCLOSED_BY='',
    DATE_FORMAT=AUTO,
    TIME_FORMAT=AUTO,
    TIMESTAMP_FORMAT=AUTO
)
ON_ERROR=ABORT_STATEMENT
PURGE=TRUE
```

Step 2:

Creating Tables:

Fact_Sales

```
create or replace TABLE MYDATA.KOKO.FACT_SALES (  
  
  ROW_ID NUMBER(38,0) NOT NULL,  
  
  ORDER_ID VARCHAR(255),  
  
  PRODUCT_ID VARCHAR(255),  
  
  CATEGORY VARCHAR(20),  
  
  SUB_CATEGORY VARCHAR(20),  
  
  PRODUCT_NAME VARCHAR(255),  
  
  SALES FLOAT,  
  
  QUANTITY NUMBER(38,0),  
  
  DISCOUNT FLOAT,  
  
  PROFIT FLOAT,  
  
  LOCATION_ID NUMBER(38,0),  
  
  TIME_ID NUMBER(38,0),  
  
  SEGMENT VARCHAR(255),  
  
  SHIP_MODE VARCHAR(255),  
  
  CUSTOMER_ID VARCHAR(255),  
  
  CUSTOMER_NAME VARCHAR(255),  
  
  STATUS NUMBER(38,0),  
  
  primary key (ROW_ID)  
  
);
```

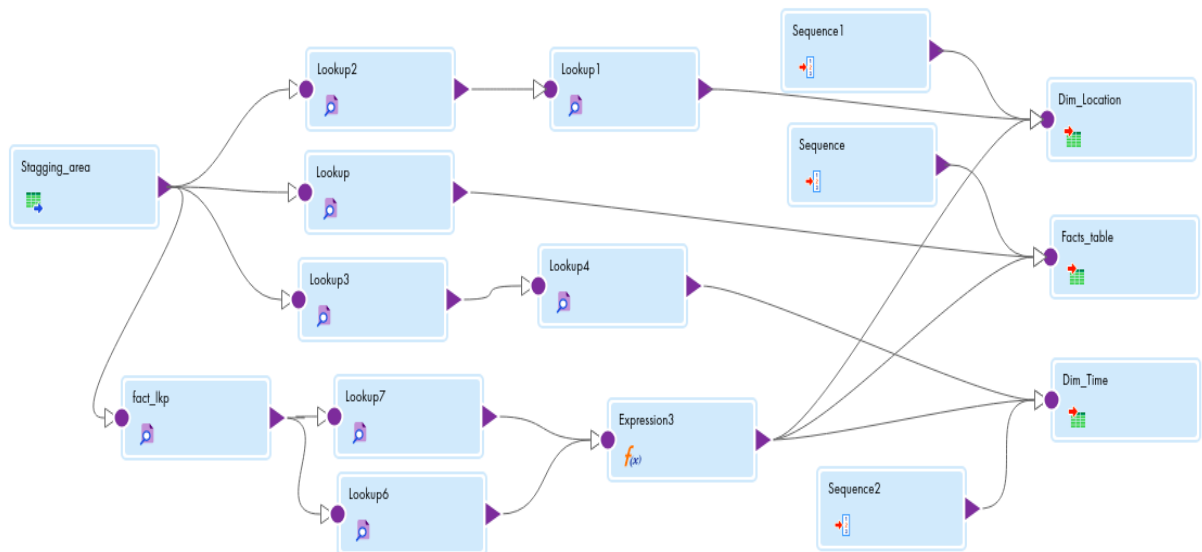
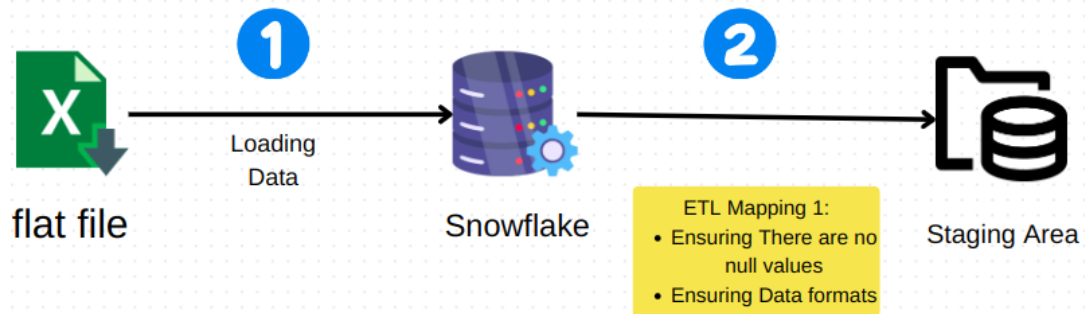

Dim_location:

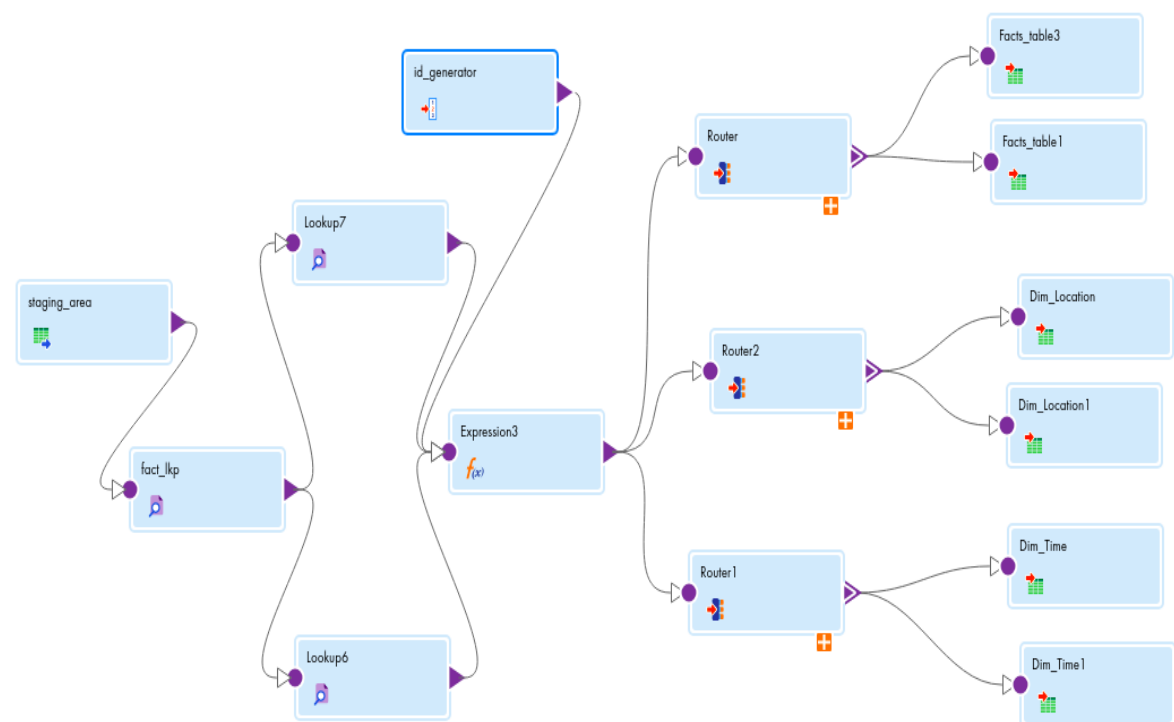
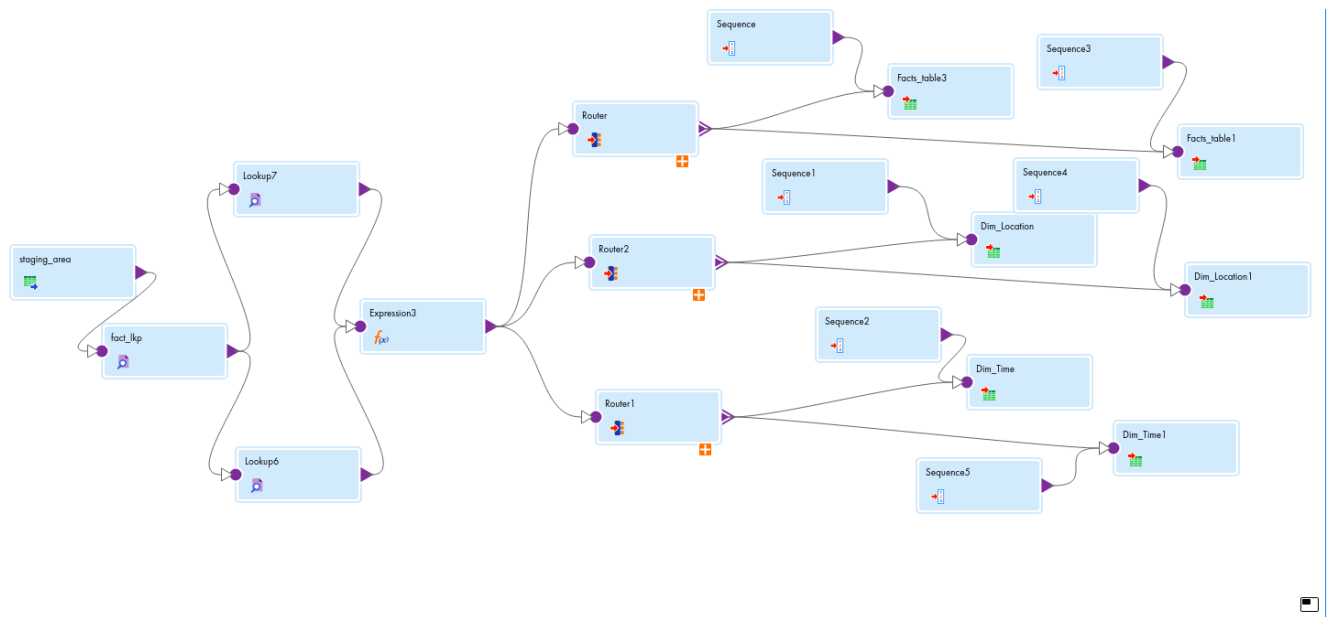
```
create or replace TABLE MYDATA.KOKO.DIM_LOCATION (  
LOCATION_ID NUMBER(38,0) NOT NULL,  
COUNTRY VARCHAR(255),  
CITY VARCHAR(255),  
STATE VARCHAR(255),  
POSTAL_CODE NUMBER(38,0),  
REGION VARCHAR(20),  
STATUS NUMBER(38,0),  
primary key (LOCATION_ID)  
);
```

Dim_time:

```
create or replace TABLE MYDATA.KOKO.DIM_TIME (  
TIME_ID NUMBER(38,0) NOT NULL,  
ORDER_DATE DATE,  
SHIP_DATE DATE,  
STATUS NUMBER(38,0),  
primary key (TIME_ID)  
);
```

IICS Procedures:





Expression Transformer:

For update flag:

```
iif((timefct_ROW_ID != locfts_src_ROW_ID or locfts_src_ORDER_ID != timefct_ORDER_ID or  
locfts_src_ORDER_DATE != locORDER_DATE or locfts_src_SHIP_DATE != locSHIP_DATE or  
locfts_src_SHIP_MODE != locfts_SHIP_MODE or locfts_src_SEGMENT != locfts_SEGMENT or  
locfts_src_COUNTRY != timeCOUNTRY or  
locfts_src_CITY != timeCITY or  
locfts_src_STATE != timeSTATE or  
locfts_src_POSTAL_CODE != timePOSTAL_CODE or locfts_src_REGION != timeREGION or  
locfts_src_PRODUCT_ID != timefct_PRODUCT_ID or locfts_src_CATEGORY != timefct_CATEGORY or  
locfts_SUB_CATEGORY != locfts_SUB_CATEGORY or  
locfts_src_PRODUCT_NAME != locfts_PRODUCT_NAME  
or locfts_src_SALES != locfts_SALES or  
locfts_src_QUANTITY != locfts_QUANTITY or  
locfts_src_DISCOUNT != locfts_DISCOUNT or  
locfts_src_PROFIT != locfts_PROFIT OR  
timefct_CUSTOMER_ID != locfts_src_CUSTOMER_ID OR  
locfts_src_CUSTOMER_NAME!= timefct_CUSTOMER_NAME),1,0)
```

For insert flag:

```
iif(isnull(timefct_ROW_ID),2,0)
```

SUMMARY:

This project involved the transformation of a complex dataset from a Superstore into a structured and accessible format. Key phases included dataset acquisition and setup, data transformation and integration, and finalization with Power BI integration.

Initial Phase: Dataset Acquisition and Setup

The project began with the receipt of the Superstore dataset. Accounts on Snowflake and IICS were set up. The dataset was uploaded into Snowflake after creating preliminary schemas.

Data Transformation and Integration

Data integration and transformation occurred in the second phase. A staging area was established in IICS, ensuring data quality by eliminating null values.

Structured Data and Snowflake Schema

Structured data was organized into fact and dimension tables within Snowflake. This involved creating mappings, adding primary keys, and implementing advanced data strategies.

Finalization and Power BI Integration

In the final phase, an indirect lookup table was created, and update flags were introduced to track data changes. Integration with Power BI resulted in the creation of an initial dashboard.

Overall, the project successfully transformed raw data into informative dashboards, providing valuable insights for data analysis and decision-making. It established a strong foundation for future data-driven initiatives.

Conclusion:

This project successfully transformed complex Superstore data into user-friendly dashboards. It encompassed data setup, integration, and Power BI integration.

Data Transformation Success

We kickstarted the project by getting the Superstore data and setting up Snowflake and IICS accounts. Efficient data management was achieved with data uploaded to Snowflake.

Structured Data and Insights

We organized data into fact and dimension tables, creating an initial Power BI dashboard to showcase insights.

Value and Future Outlook

In a nutshell, this project turned raw data into practical insights for decision-making. It lays the foundation for future data-driven projects, highlighting the importance of organized data in generating valuable knowledge.

