SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

Data Integration Design Workgroup

CASE STUDY	ALASKA & ARKANSAS - OIL, GAS, WATER PRODUCTION
PROFESSOR	JOSÉ CURTO DÍAZ (jcurto@faculty.ie.edu)
TEAM MEMBERS	TEAM A; *Ckalib N., Abdulaziz A.i, Michael W., , Nabil M. & Sarang Zendhrooh*
DUE DATE	29-MAR-2020, 11:59 PM CET

Introduction

Before working on the Data Integration Design, Team A completed the Data Warehouse Modeling in SQL Workbench. To complete the Data Warehouse Modeling, Team A documented the data set analysis, approach selection process, and the final data warehouse design created in SQL Workbench. With the Data Warehouse Modeling complete, Team A began to work on the Data Integration Design. For this project, the team was asked to document the following:

1. Extraction
2. Data Mapping
    a. Data Mapping Process
    b. Transformation Approach
    c. Transformation
3. Data quality tracking
4. Metadata approach

As a result, this Data Mapping document is structured to discuss each of the aforementioned items with a conclusion on the BI Delivery Interface. Given that the next project, the Individual - Dashboard Design, is dependent on the team's ability to connect to a visualization software for presentation purposes, Team A thought it wise to briefly touch on this step after the completion of the Data Integration Design.

Extraction

Although the primary datasets were provided to the team, the original data sources were the following websites:

- Enigma.io
- Data.gov

After building the data model in SQL Workbench, Team A determined a strategy to best extract the data from SQL Workbench to PDI. The core element to this strategy was ensuring there was a high-level plan. This high-level plan illustrated the relationship between the sources and targets.

As mentioned earlier, the sources, or the place in which the data was obtained, were enigma.io and data.gov. These sources not only provided the Alaska Oil, Water & Gas Production and the Arkansas Oil, Water, and Gas Production, it also provided the team additional information on the wells. Given that the API No. in each dataset represented a unique number to each well, the team could extract information, such as the exact location of each well.

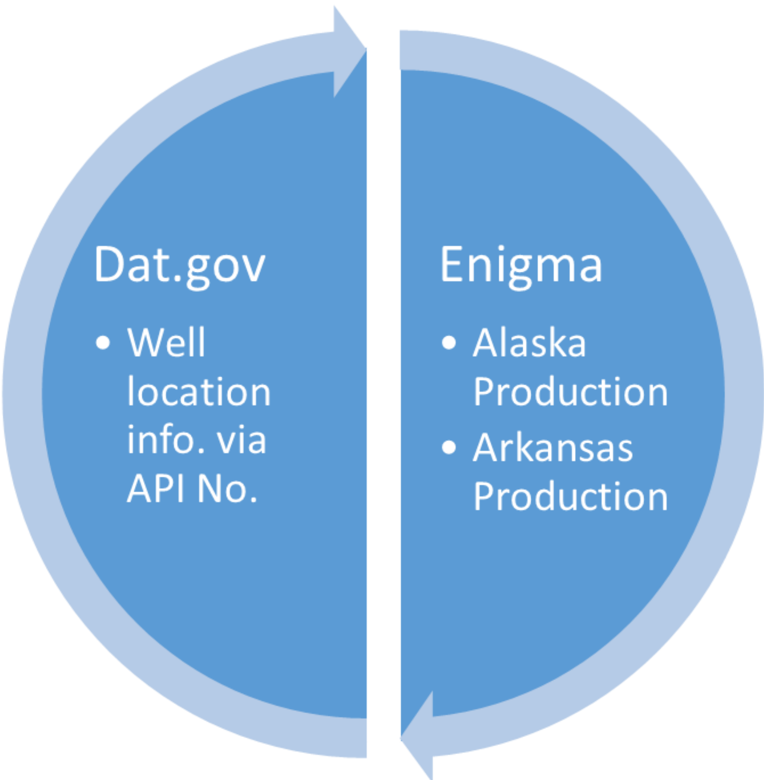


Illustration: Sources

On the other hand, the targets, or the final stages in which the team preferred the data, resembled the dimensions and fact tables in the Star Schema, as shown on page 9 of the team's Data Integration Design Report: Date, Wells, Location, and Production:

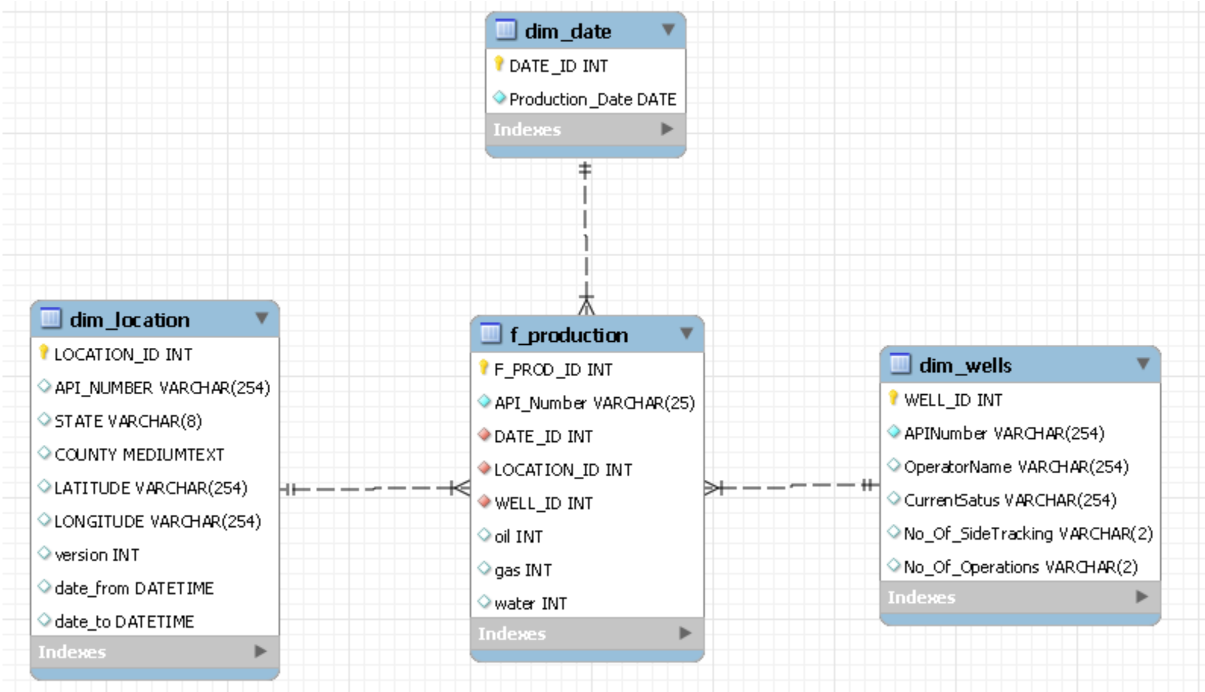


Illustration: Star Schema Model

Here is an illustration of the targets yielded from the sources:

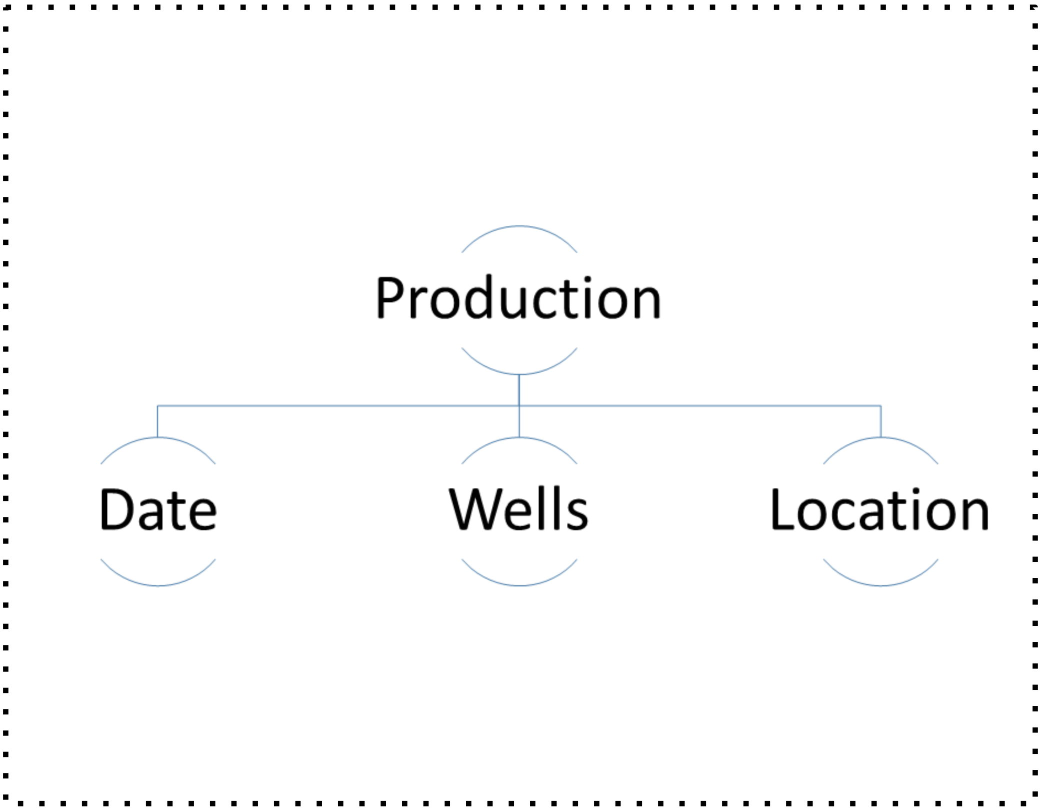


Illustration:Targets

Data Mapping

Data Mapping Process

Although there are no extravagant transformations, the team has detailed the load schematic for the fact table. Whereas other transformations include changes such as sorts of particular attributes or bucketing, the transformations required for this ETL process were not that extensive. Here is an illustration of the detailed load schematic for the fact table:

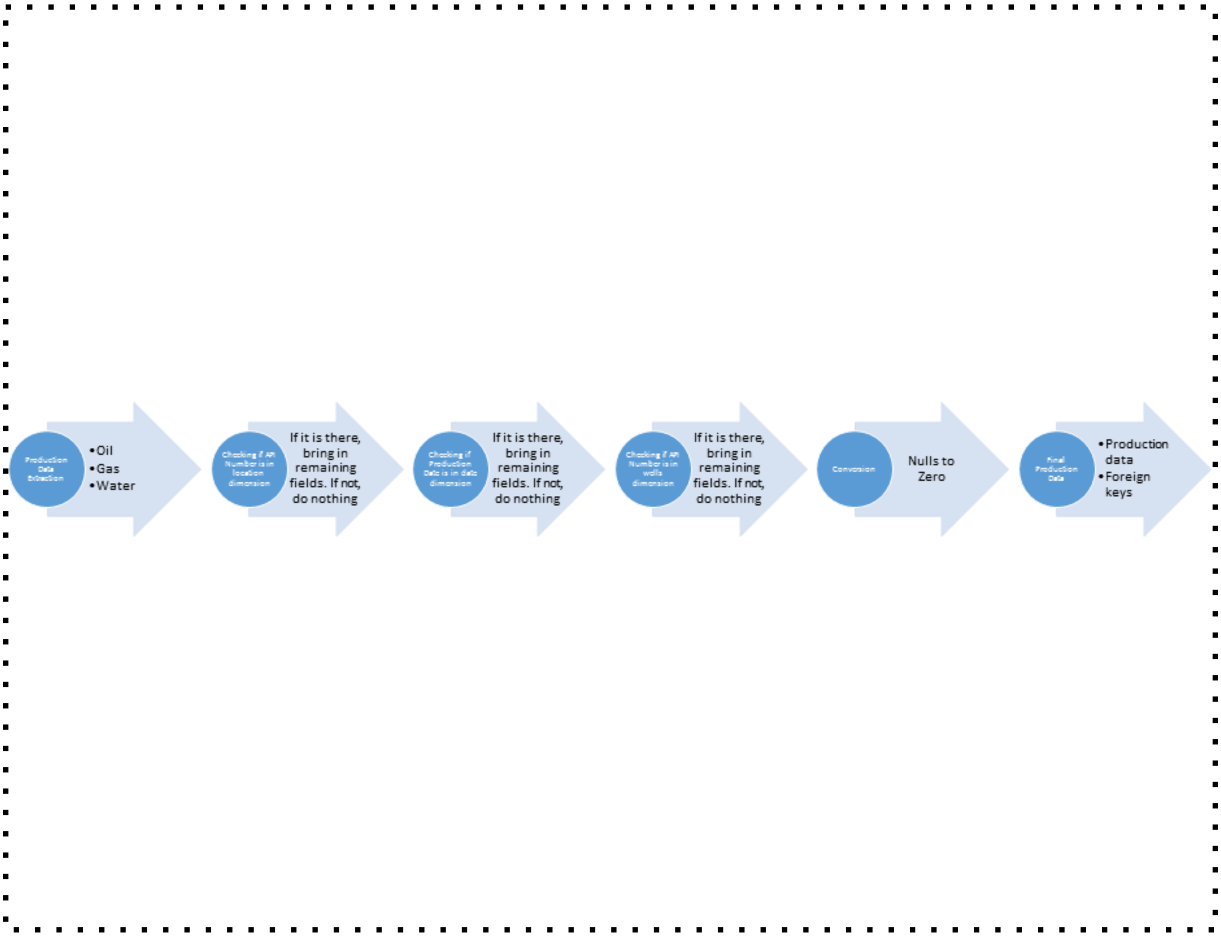


Illustration: Data Mapping Process

Transformation Approach

During the analysis process, Team A thought about two transformation approaches and pros and cons for each:

1. Dropping the indexes and disable the constraints before loading the data: drop all the indexes before loading the data then rebuild them after completion.
    a. Pros:
        i. Load all the data for dimension and facts in parallel at the same time.
        ii. Faster in data loading.
        iii. Loading order is not important.
        iv. Helpful in full loading (bulk insert).
    b. Cons:
        i. Risky in applying referential integrity after loading the whole data.
        ii. Rebuilding the indexes takes time to calculate the statistics.
        iii. Not helpful in incremental refreshment.
        iv. Useless if the data volume is small.
2. Keep the indexes and constraints during the transformation: Keep the indexes and constraints without changes during the data loading but the team had to be cognizant of the referential integrity and the data loading order
    a. Pros:
        i. It will guarantee the data referential integrity during and after the data loading.
        ii. It's more effective in incremental refreshment cases.
        iii. No need to rebuild the indexes in this case.
    b. Cons:
        i. Takes longer time in the data loading.
        ii. Not helpful in loading the full data (bulk loading).
        iii. Not recommended if the data volume is huge.
        iv. Be conscious of the data loading order, otherwise it will violate the data referential integrity.

Choosing the suitable approach: Since an incremental data refreshment as well as the loaded data on the periodic basis is not huge, there's no need to drop the indexes every time the data is refreshed and rebuild them after completion, as this is more expensive than loading with the indexes available.

Transformation

In PDI, two Transformations had to be completed before the completion of the Job. The first transformation, entitled “Dimension Loading,” yielded the three dimensions from SQL Workbench via a parallel loading transformation. The Dimension Loading transformation includes the extraction of the location, date, and wells data from SQL Workbench, a couple manipulations of each data set, and then prior to a load into their respective tables. Here is an illustration of the Dimension Loading transformation:

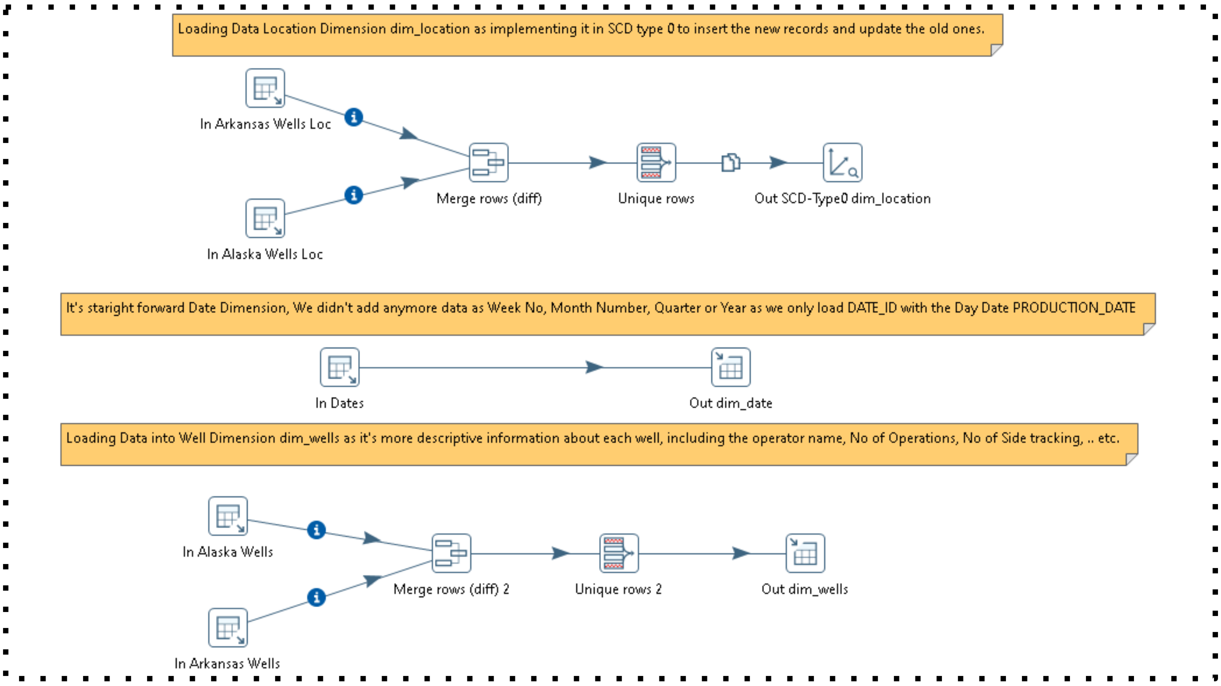


Illustration: Dimension Loading Transformation

The second transformation, entitled "Fact Loading," yields the content of the fact table. The Fact Loading transformation includes the extraction of the production data from SQL Workbench and the foreign keys to connect to the data from the dimensions. Here is an illustration of the Fact Loading Transformation:

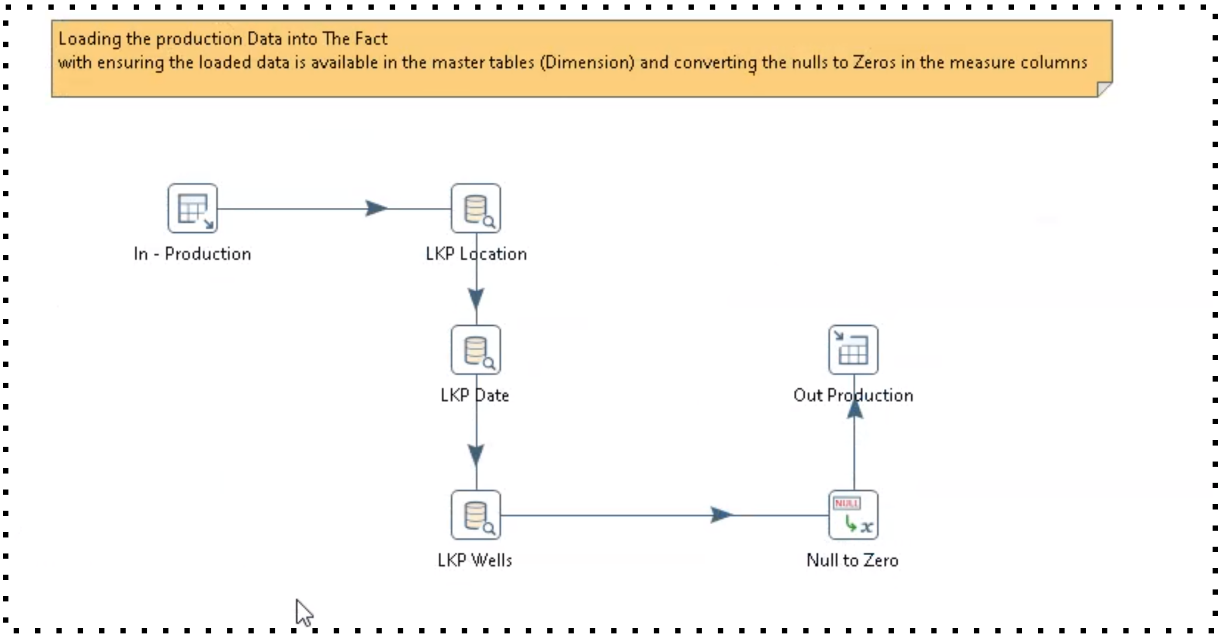


Illustration: Fact Loading

Following the completion of the Dimension Loading and Fact Loading transformations is the completion of the Job for the schema loading. Since a Job in PDI can include a series of transformations, the team leveraged the aforementioned transformations, Dimension Loading and Fact Loading, to complete the schema. Here is an illustration of the complete schema:

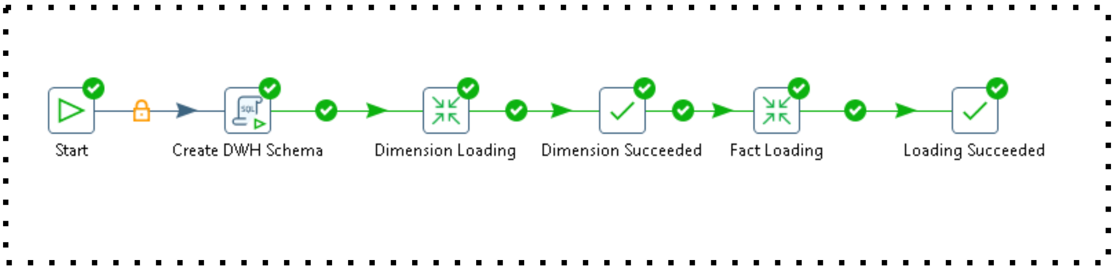


Illustration: Schema

With a successful Dimension Loading and Fact Loading, there were no complications with executing the ETL script, as shown in the illustration below:

Job / Job Entry	Comment	Result	Reason	Filename
Schema Loading Job				
Job: Schema Loading Job	Start of job execution		start	
Start	Start of job execution		start	
Start	Job execution finished	Success		
Create DWH Schema	Start of job execution		Followed unconditional link	
Create DWH Schema	Job execution finished	Success		
Dimension Loading	Start of job execution		Followed link after success	file:///D:/IE University/Study/DHW & BI/Assignment 02/Dimension Loadi
Dimension Loading	Job execution finished	Success		file:///D:/IE University/Study/DHW & BI/Assignment 02/Dimension Loadi
Dimension Succeeded	Start of job execution		Followed link after success	
Dimension Succeeded	Job execution finished	Success		
Fact Loading	Start of job execution		Followed link after success	file:///D:/IE University/Study/DHW & BI/Assignment 02/Fact Loading.ktr
Fact Loading	Job execution finished	Success		file:///D:/IE University/Study/DHW & BI/Assignment 02/Fact Loading.ktr
Loading Succeeded	Start of job execution		Followed link after success	
Loading Succeeded	Job execution finished	Success		
Job: Schema Loading Job	Job execution finished	Success	finished	

Illustration: Execution Results

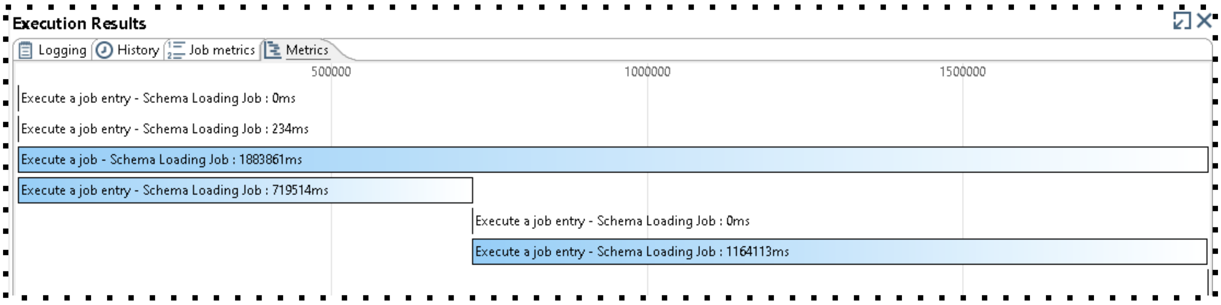
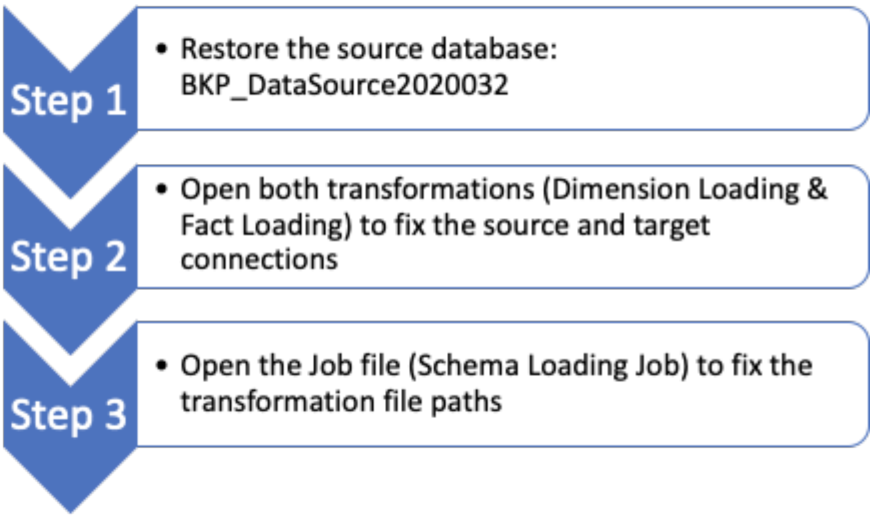


Illustration: Execution Metrics

Finally, here are the steps needed to execute the transformations:



For more information about the Data Mapping Process, please refer to the Mapping Sheet in the Appendix.

Data Quality Tracking

The first step in data quality tracking was performing simple queries into the data sets to check for null values and transform those null values to 0. Fortunately, the Alaska dataset did not have any quality issues; however on the Arkansas dataset, 1158 distinct API values were missing.

The steps to overcome this quality issue are solved while loading the data, sourced from the provided datasets and websites referenced. The team needed to make sure that the key is available in the dimensions, then the dimensions are validated. In case the team does not have parent or referential integrity, the record cannot be loaded. Therefore the slowly changing dimensions (SCD) attribute was developed, in this case type 0, adopted to the location since the quality data issue was observed on longitude and latitude parameters. This will perform a so called lookup on the dimensions against data extracted from the data source to check whether any location has changed in order to update it and if new, this will be inserted. This type of SCD has been selected because the dimension attribute value never changes, so facts are always grouped by this original value.

The quality check has been applied to the three below attributes:

- Whether the production date available in the fact is the same as the one available in the dimensions while loading the fact
- Whether the location available in the fact is the same as the one available in the dimensions while loading the fact. This will be achieved through the first five (5) digits of the API number
- Whether the well dimensions available in the fact is the same as the one available in the dimension

For the above three cases, if the values between the fact and dimensions do not match, a redirect is performed which moves the values to the rejected data/table in order to identify the unavailable data.

Metadata Approach

The ETL system by default is responsible for the creation and use of lots of metadata involved in the BI/DW environment. The team developed the metadata approach to capture business, technical as well as metadata processes.

Metadata is the basic information that references a specific data, enabling working with particular instances of data easier. Examples of metadata approach in the ETL is the description of each column from the business perspectives. Other metadata information include:

- Fact Datatype (The Foriegn Keys in fact table have the same data types of their parents)
- Forgein keys (All the foriegn key names starts with FK_)
- Primary Keys (All the primary keys are auto incremental)
- Date Fields (Unified the data formats)
- Null Values (All the numeric fields that have any null value have been replaced with 0s)

Team has a naming convention for indexes and constraints, for example foreign keys start with “FK_.” The primary objective of having a naming convention is to enable easy identification of the type and purpose of all objects contained in the index.

The following queries provides more information about the metadata in each table:

```
SELECT * FROM INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS WHERE
CONSTRAINT_SCHEMA='ENERGY_PROD_DWH';
```

The INFORMATION_SCHEMA .REFERENTIAL_CONSTRAINTS provides information about the foreign keys in the schema.

```
SHOW COLUMNS FROM ENERGY_PROD_DWH.F_PRODUCTION;
```

The SHOW COLUMNS query provides information regarding the associated columns in the production fact table. Including: Field(column name),Type(data type), Null(nullability), Key, Default values and extra which can contain additional information such as ‘auto_increment.’

```
SELECT COUNT(1) NO_OF_RECORDS, MAX(OIL) MAX_OIL, MAX(WATER) MAX_WATER,
MAX(GAS) MAX_GAS FROM ENERGY_PROD_DWH.F_PRODUCTION;
```

The query provides information regarding the number of records and maximum values of each oil, water and gas in the production fact table.

```
SHOW COLUMNS FROM ENERGY_PROD_DWH.DIM_LOCATION;
```

The SHOW COLUMNS query provides information regarding the associated columns in the location dimension table.

```
SELECT COUNT(1) NO_OF_RECORDS, STATE, COUNTY  FROM
ENERGY_PROD_DWH.DIM_LOCATION GROUP BY STATE, COUNTY;
```

The query provides information regarding the number of records for each county and associated state from the location dimension table.

```
SHOW COLUMNS FROM ENERGY_PROD_DWH.DIM_WELLS;
```

The SHOW COLUMNS query provides information regarding the associated columns in wells dimension table.

```
SELECT COUNT(1) NO_OF_RECORDS, OPERATORNAME  FROM ENERGY_PROD_DWH.DIM_WELLS
GROUP BY OPERATORNAME;
```

The query provides information regarding the number of records for each operator from the wells dimension table.

```
SELECT COUNT(1) NO_OF_RECORDS, CurrentStatus  FROM
ENERGY_PROD_DWH.DIM_WELLS  GROUP BY CurrentStatus;
```

The query provides information regarding the number of records for each Current status of wells from the wells dimension table.

```
SHOW COLUMNS FROM ENERGY_PROD_DWH.DIM_date;
```

The SHOW COLUMNS query provides information regarding the associated columns in the date dimension table.

```
SELECT COUNT(1) NO_OF_RECORDS, MIN(PRODUCTION_DATE) MIN_DATE,
MAX(PRODUCTION_DATE) MAX_DATE FROM ENERGY_PROD_DWH.DIM_DATE;
```

The query provides information regarding the number of records from the earliest date to the latest date in the date dimension table.

Conclusion

Team A used a multi-step process to complete a one-time historic data load in PDI. First, the data had to be extracted from My SQL database. During this process, the team outlined both the sources and targets, which are the three dimensions and fact table as shown in the illustration entitled, “Star Schema.” Second, the team executed the Transformation and Job required to assemble the targets in PDI. The Transformation in PDI for the Fact Loading is outlined in the illustration entitled, “Data Mapping Process” (Data Mapping Process). A parallel loading transformation was required to build the three dimensions, as shown in the illustration entitled, “Dimension Loading Transformation” (Transformation). Fortunately, there were no complications running the final Job in PDI. Third, the team recorded the small subset of undefined data points in an effort to maintain data integrity. However, this was resolved by converting nulls to zeros, as the team, despite all efforts to find the missing data points from several sources, was not able to find an alternative solution for these data points. As for the metadata approach, the team utilised metadata to allow definition to be associated with the data which enables users to interact, identify and understand the underlying data. With both the data warehouse and data integration processes complete, each member of Team A is prepared to complete a dashboard using Tableau.

Appendix

Mapping Sheet

Target Schema	Target Table	Target Type	Target Columns	Source Table	Source Column	Expression
ENERGY_PROD_DWH	DIM_DATE	Dimension	DATE_ID	AUTO INCREMENTAL		
ENERGY_PROD_DWH	DIM_DATE	Dimension	PRODUCTION_DATE	ENERGY_PRODUCTION	PRODUCTION_DATE	
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	API_NUMBER	ENERGY_PRODUCTION	API_NUMBER	
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	COUNTY	COUNTIES	SHORT_NAME	
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	DATE_FROM	VERSION DATE - INSERTED THROUGH DATA INTEGRATION		SYSDATE()
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	DATE_TO	VERSION DATE - INSERTED THROUGH DATA INTEGRATION		
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	LATITUDE	ARKANSAS_WELL_NUMBERS & ALASKA_WELL_NUMBERS	LATITUDE & Y	
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	LOCATION_ID	AUTO INCREMENTAL		
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	LONGITUDE	ARKANSAS_WELL_NUMBERS & ALASKA_WELL_NUMBERS	LONGITUDE & X	
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	STATE	FIXED TEXT BEING INSERTED WITH EACH STATE DATA		ARKANSAS' & 'ALASKA'
ENERGY_PROD_DWH	DIM_LOCATION	Dimension	VERSION	VERSION NUMBER - INSERTED THROUGH DATA INTEGRATION		
ENERGY_PROD_DWH	DIM_WELLS	Dimension	APINUMBER	ENERGY_PRODUCTION	API_NUMBER	
ENERGY_PROD_DWH	DIM_WELLS	Dimension	CURRENT STATUS	ARKANSAS_WELL_NUMBERS & ALASKA_WELL_NUMBERS	CURRENT STATUS & STATUSSYM	
ENERGY_PROD_DWH	DIM_WELLS	Dimension	NO_OF_OPERATIONS	ENERGY_PRODUCTION	API_NUMBER	SUBSTR(APINUMBER,13,2)
ENERGY_PROD_D	DIM_WELLS	Dimension	NO_OF_SIDETRACKI	ENERGY_PRODUCTION	API_NUMBER	SUBSTR(APINUMBER,1

WH			NG			1,2)
ENERGY_PROD_DWH	DIM_WELLS	Dimension	OPERATORNAME	ARKANSAS_WELL_NUMBERS & ALASKA_WELL_NUMBERS	OPERATORNAME & OPERATOR	
ENERGY_PROD_DWH	DIM_WELLS	Dimension	WELL_ID	AUTO INCREMENTAL		
ENERGY_PROD_DWH	F_PRODUCTION	Fact	API_NUMBER	ENERGY_PRODUCTION	API_NUMBER	
ENERGY_PROD_DWH	F_PRODUCTION	Fact	DATE_ID	DIM_DATE	DATE_ID	
ENERGY_PROD_DWH	F_PRODUCTION	Fact	F_PROD_ID	AUTO INCREMENTAL		
ENERGY_PROD_DWH	F_PRODUCTION	Fact	GAS	ENERGY_PRODUCTION	GAS	
ENERGY_PROD_DWH	F_PRODUCTION	Fact	LOCATION_ID	DIM_LOCATION	LOCATION_ID	
ENERGY_PROD_DWH	F_PRODUCTION	Fact	OIL	ENERGY_PRODUCTION	OIL	
ENERGY_PROD_DWH	F_PRODUCTION	Fact	WATER	ENERGY_PRODUCTION	WATER	
ENERGY_PROD_DWH	F_PRODUCTION	Fact	WELL_ID	DIM_WELLS	WELL_ID	