IE SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

Business Intelligence & Datawarehousing

DATA WAREHOUSE MODELLING

-

Data Integration Using ETL of Production Data on Oil Wells in the State of New Mexico

Submitted by:	Team 1 Group O2
Group Members:	Rajat Prasad Carolin Kröger Sara Krumpak Mohamad el Charif Andrea Salvati Dominik Thausing Alejandro Torres Pérez
Date:	November 25th, 2018

Table of Contents

Table of Tables

Table of Figures

Executive Summary

The following report describes the Data integration process using Extract, Transformation and Loading. The First step taken was to make necessary changes to the schema based on the recommendation received. Some of the changes were like merging the "d_county_id" and the "d_Area". Also, some columns that were proposed in the fact table were dropped as they could only be obtained by special mathematical transformation which is not a part of ETL process. After changes made to dimensions and the fact table, new foreign keys were chosen which were basically primary key of dimensions. They were created because of auto sequence. All these primary keys in dimension table represent a unique column or a set of unique columns. These could only be populated when we run the loading process on Pentaho. After fixing the bugs in MySQL workbench file, A strategy was developed to understand the data and fix the anomalies. Since we have the information that what columns / attributes belong to which dimension and which of these attributes should be unique and not null. A set of transformations were created to create dimensions and load them in the database. A final of transformation was created to load data in fact table.

However, the first step undertaken was to create a transformation called "TR_Cleaning" where we are deleting the rows with NULL values in key columns and sorting and removing their duplicates. For some of the columns Null values were replaces by 0. We made sure that source of data was added after every transformation, not only source by system log time and last modified time of transformation as well, for creating meta data and to resolve any data governance issues.

As the business use of case of the data is to analyze monthly oil gas and water of each well therefore the fact table contains these crucial facts along with the information of the entry. The fact table contains reference to all dimensions such as "d_operator" which contains information about operators. Thus, reducing redundancy and fast processing of queries and providing a more organized structure.

Requirements

Before proceeding in the explanation of the ETL process, it is pivotal to check 10 important requirements that the system needs to meet in order to be effective.

1. Business Needs

The business objective of the ETL processes is to give managers of oil wells in the state of New Mexico resources to perform data driven decisions, clustered into three directions of management:

- recognizing underperforming wells
- optimizing wells performance
- understanding where investments should be focused.

2. Compliance

We have treated the data in our model in an accurate and complete way. Moreover, we confirm that data were not tampered and that the model represents the real state of the business. For this reason our ETL process should meet the legal requirement of the sector. However, we suggest a deeper analysis from the legal department of the company.

3. Data Quality

The initial dataset presented some distortions and it was not ready to use for an ETL process. This is the reason why the first transformation of the process is about the cleaning of the data (see below for the detailed report of the cleaning)

4. Security

In order to maximize the security of the process we have considered mainly 2 measures:

- Physical backup of all the ETL processes and the MySql schema
- Report with carefully explained instruction to perform all the transformations

5. Data integration

We have designed our ETL model with just one database so we have not included common dimensional attributes across different databases and we have not created common business matrix. However, if in the future, the company decides to implement a second database the steps above will be required.

6. Data Latency

The dataset that we used for the ETL process is composed by 1,000,000 rows. This is why the ETL processes takes a considerable amount of time to be performed. (more details will be discussed in the Fact Table's transformation description)

7. Archiving and Lineage

In order to reach the best performance in terms of Archiving and Lineage the following actions have been taken:

- Backup of the output of the cleaning transformation, being the most important step of the process;
- Back up of every transformation in a separate file;
- Metadata details have been added to every transformation.

8. BI delivery Interfaces

The delivery platform for our ETL process is MySql Workbench. Every transformation has been uploaded to the schema designed in MySql Workbench and can be easily extract through the platform.

9. Available Skills

No programming skills were required in order to build the ETL processes. The skills that are needed to perform this task are related to the use of the PDI software and basic sql knowledge.

10. Legacy Licenses

The ETL process has been build using Pentaho Data Integration and performed using My Sql Workbench.

Instructions for Executing the ETL process

We have all the input files (csv format) in the input folder present in the GA2 folder. We also have all the transformation and the main job in the GA2 folder.

1. **Change the username and password while connecting to database in the "input/update" step of each transformation while making database connection.**
2. We shall now open a job called "JB_WELL" present in the GA2 folder. There is no need to make any change. Simply run the job.
3. You can check the loading output of all the transformation in the MySql workbench file called "well_status.mwb" also present in the GA2 folder.

Timing of the ETL process

transformation	Time
Cleaning	0,3s
Status	1,6s
Company	5,1s
Operator	5,3s
Date	9,6s
Area	19,8m
Fact_Table	NA (155,000 rows in 5.30h)

Table 1 -Timing of the ETL process

Transformations overview

1. **TR_Cleaning.ktr**.

   There are two sub-transformations in this file. The first transformation deals with "wel_mexico_all_new_format.csv" and the second deals with "production_mexico_all_new_format.csv". In the first step of the first sub-transformation the filename is

${Internal.Entry.Current.Directory}/input/wel_mexico_all_new_format.csv. In the last step of the first sub transformation called “output well” the output folder is ${Internal.Entry.Current.Directory}/output/dataset_well. Now let’s visit the second sub-transformation, the filename is “${Internal.Entry.Current.Directory}/input/production_mexico_all_new_format.csv” and the output folder is ${Internal.Entry.Current.Directory}/output/dataset_production.

2. **TR_DATE**
   the filename is “${Internal.Entry.Current.Directory}/output/dataset_production.csv” in “csv file input production” and in the last step of the transformation called “insert/update” already connection called “localhost” has been established. The target schema is “well_status” and the target table is “d__date”.
3. **TR_company**
   the filename is “${Internal.Entry.Current.Directory}/output/dataset_well.csv “ in csv file input well” and in the last step of the transformation called “insert/update” already connection called “localhost” has been established. The target schema is “well_status” and the target table is “d___company”.
4. **TR_operator**
   The filename is “${Internal.Entry.Current.Directory}/output/dataset_well.csv “ in csv file input well” and in the last step of the transformation called “insert/update” already connection called “localhost” has been established. The target schema is “well_status” and the target table is “d_Operator”. Click ok save and run the transformation.
5. **TR_status**
   The filename is “${Internal.Entry.Current.Directory}/output/dataset_well.csv “ in csv file input well” and in the last step of the transformation called “insert/update” already connection called “localhost” has been established. The target schema is “well_status” and the target table is “d_status”.

18. **TR_Area**
    The filename is “${Internal.Entry.Current.Directory}/output/dataset_well.csv “ in csv file input well” and in the last step of the transformation called “insert/update already connection called “localhost” has been established. The target schema is “well_status” and the target table is “d_area”.

6. **TR_FactTable**

There are two CSV file input stage called “production” and “well”. In production the filename is “${Internal.Entry.Current.Directory}/output/dataset_production.csv “.In WELL the filename is “${Internal.Entry.Current.Directory}/output/dataset_well.csv” In “Insert/Update” command set target table to “f_well”.

Implementation of Feedback

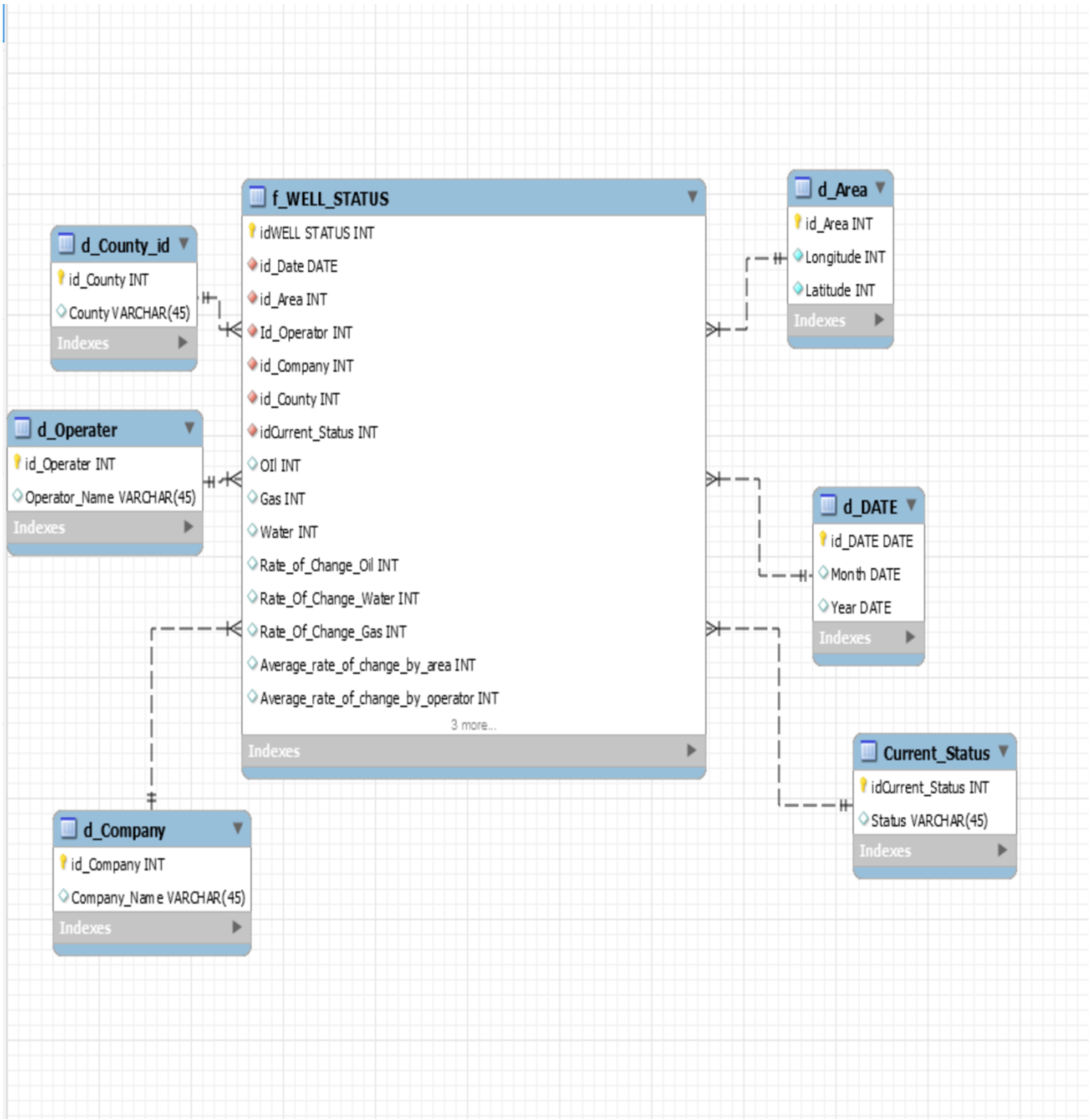


Figure 1 - Previously proposed Schema

In the previous schema we had 6 dimension tables called current_status, d_date, d_area, d_company, d_operator, d_county_id.

The fact table also contained a lot of variables which are obtained after complex mathematical calculations. We decided to drop as they are not needed at this stage of Data warehousing.

A new schema was developed taking into consideration the feedback and Business use case.

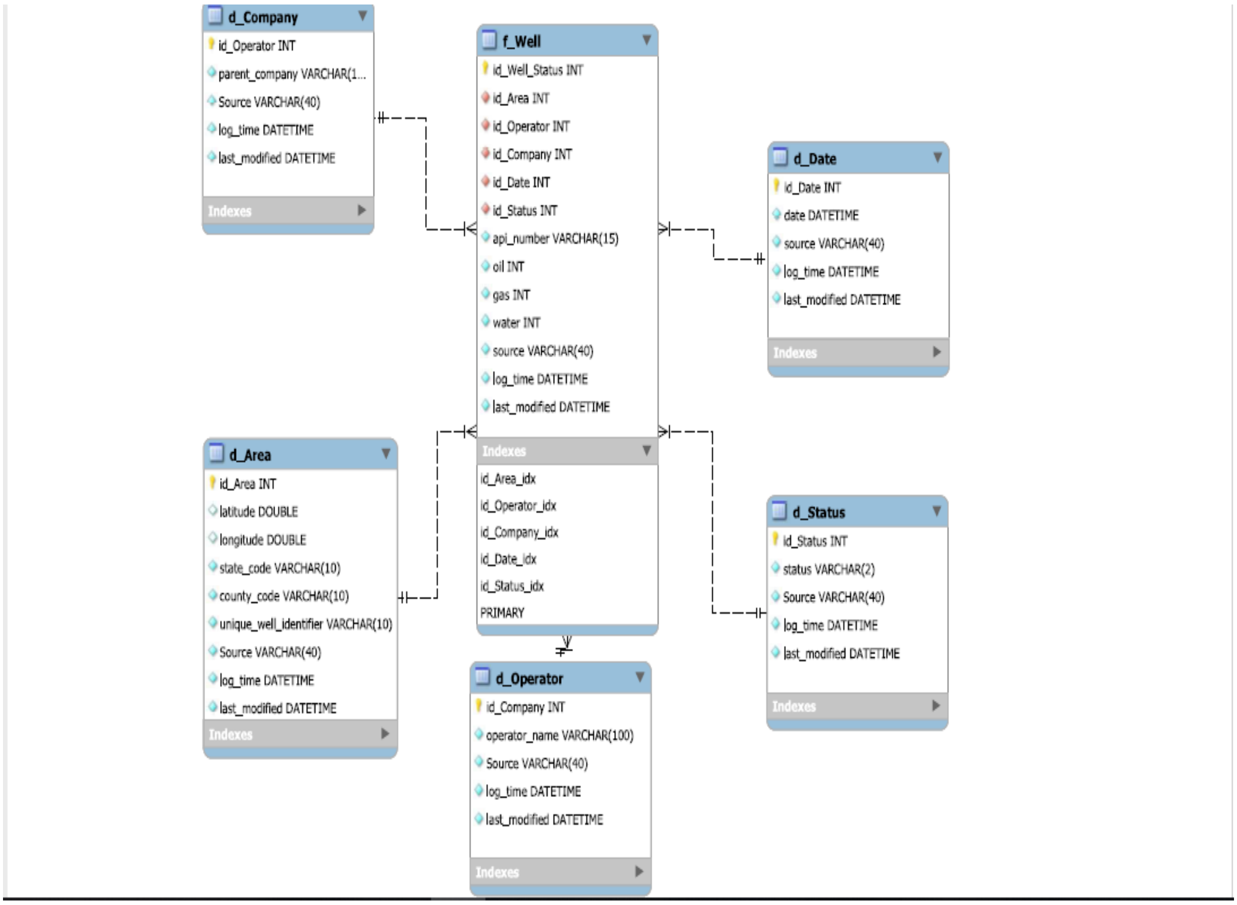


Figure 2 - Improved new Schema

Although the previous schema had major problems (like containing derived values which are not suitable at this stage of data warehousing), it had the design that was in conformance with the business need (the star schema). We decided to get rid of derived columns. We merged the “area” and “county” dimensions. We analyzed each variable in the dataset and calculated their max length so that in dimensions we have appropriate size for each attribute. We fixed the same for all dimensions according to standard. We came up with all these dimensions after thoroughly analyzing the need of the stakeholders. How will they benefit with these data mart.

Data Integration using Pentaho for ETL

We were provided data in two json files, one containing We manipulated these files to be read by dataiku which converted it into a csv file. These CSV files were then saved in Input folder present in the GA2 folder

Our first transformation in PDI was data cleaning- taking out null and duplicate values. Or reassigning a value to a Null value. The outputs of these transformation were then saved as a csv file in the output folder which is also present in the GA2 folder.

information about all the wells in New Mexico, United States and the other contained monthly production data of each well.

Overview Of All Steps

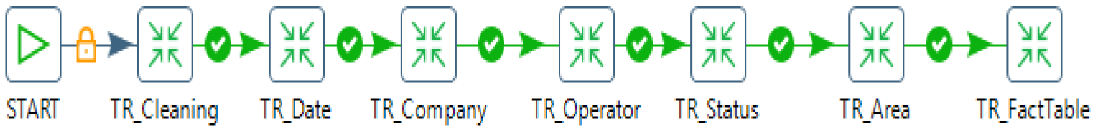


Figure 3 - Overview of all steps

We have 7 transformation in total, which operates on the two csv files obtained as a result of a series of transformation on json files that we were given in the first place.

The first transformation is **TR_Cleaning**, it takes care of the null values, it also removes duplicate values and the output is fed to a new CSV file called dataset_prodcution and dataset_well.

The second Transformation was to create a dimension for **dates** as shown in the ER diagram of MYSQL workbench. We inserted the new processed CSV file called dataset_ production as input file and extracted all unique dates in dimension d_date.

The third Transformation was to create a dimension for **Company** as shown in the ER diagram of MYSQL workbench. We inserted the new processed CSV file called dataset_ well as input file and extracted all unique Companies and feed into dimension called d_company.

The fourth Transformation was to create a dimension for Operator as shown in the ER diagram of MYSQL workbench. We inserted the new processed CSV file called dataset_ well as input file and extracted all unique Operators and feed into dimension called d_Operator.

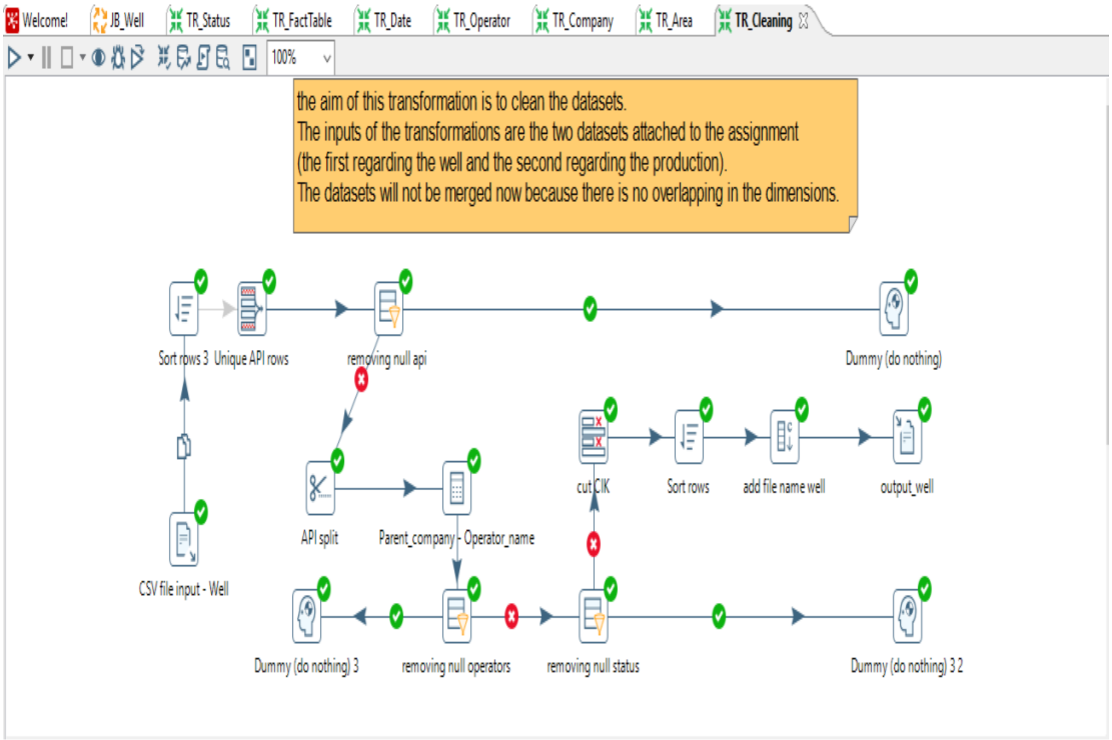
The fifth Transformation was to create a dimension for status as shown in the ER diagram of MYSQL workbench. We inserted the new processed CSV file called dataset_well as input and extracted all unique Operators and feed into dimension called d_Company.

The sixth Transformation was to create a dimension for Area as shown in the ER diagram of MYSQL workbench. We inserted the new processed CSV file called dataset_ well as input and extracted all unique combination of area, state and unique well identifier and feed all the relevant data in dimension d_area.

The seventh Transformation was to fill the fact table with columns important to business needs like oil, gas and water production along with source of the data and the log time, not only these but primary key of other five dimensions were also inserted as foreign keys as shown in the ER diagram of MYSQL workbench. We inserted the two new processed CSV files as input called dataset_prodcution and dataset_ well and extracted anecessary features like api_number, monthly oil, gas, water production and feed it in table called f_well present in schema called well_status.

Overview of individual steps

I STEP. TR_Cleaning



the aim of this transformation is to clean the datasets.
The inputs of the transformations are the two datasets attached to the assignment (the first regarding the well and the second regarding the production).
The datasets will not be merged now because there is no overlapping in the dimensions.

Figure 4 - Cleaning Transformation

In this transformation we cleaned two CSV files, But before we indulge into getting to know the working of this transformation, It should be duly noted that as input two json files were provided. These files were manipulated to enable them to be converted into CSV files. After converting them to CSV files they were used as input in this transformation.

As already mentioned we cleaned two files and therefore we have two sets of transformation. The first set of SUB-Transformation deals with cleaning the wel_mexico_all_new_format.csv, which contains attributes called API Number, latitude, longitude, operator name, well name, status, parent, CIK.

The first step was to read the input file and sort it according to API number, the second step was to remove the duplicate API number because one well cannot have more than one API number, in a nutshell it has to be unique in this table, It is not possible to two or more columns having same API number but separate well name, operator name etc.

After removing the duplicate API number, all the rows that don’t have an API were removed using filter row step.

Next, we use String cut command to split the API number into 3 substrings called- 1) the state code which is the first 3 digits of API number. 2) the county code fourth to sixth digit. 3) the third is the unique well identifier and it comprises of last four digits.

These three substrings provide us information about the state, county and the unique id of a certain well.

As there were many null values in parent company columns, these null values were replaced by the corresponding operator name column. This was done using the calculator tool.

Next, all the rows in which column "operator name" was null were removed because our stakeholders are people who own these wells and that the wells without operator name are not useful for our final goal.

Next, all the rows in which column "status" was null were removed using filter rows.

Column "CIK CODE" was removed as it does not provide any knowledge to achieve the stakeholders´ final goal.

Before creating a new CSV file and inserting all the data into it, a new column called source was added and for each row "file name" was added. This column will tell the stakeholder the source of data for each row.

All this data was inserted into CSV file called dataset_well.csv

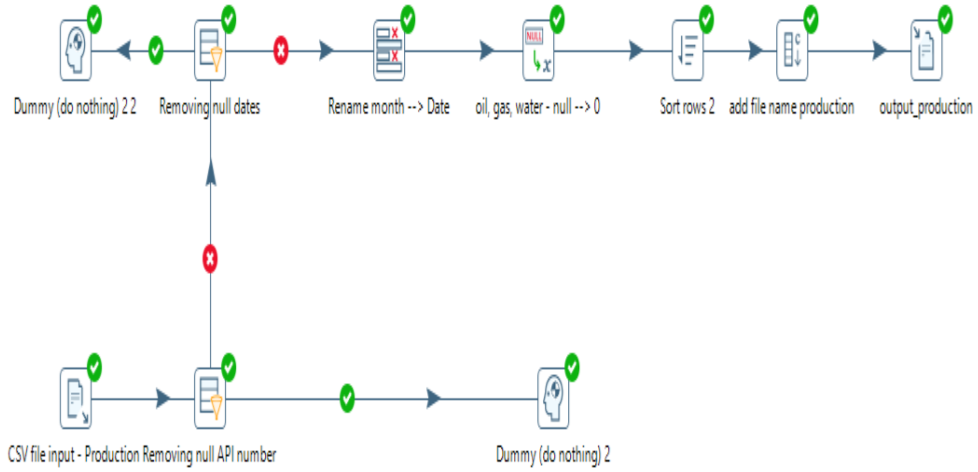
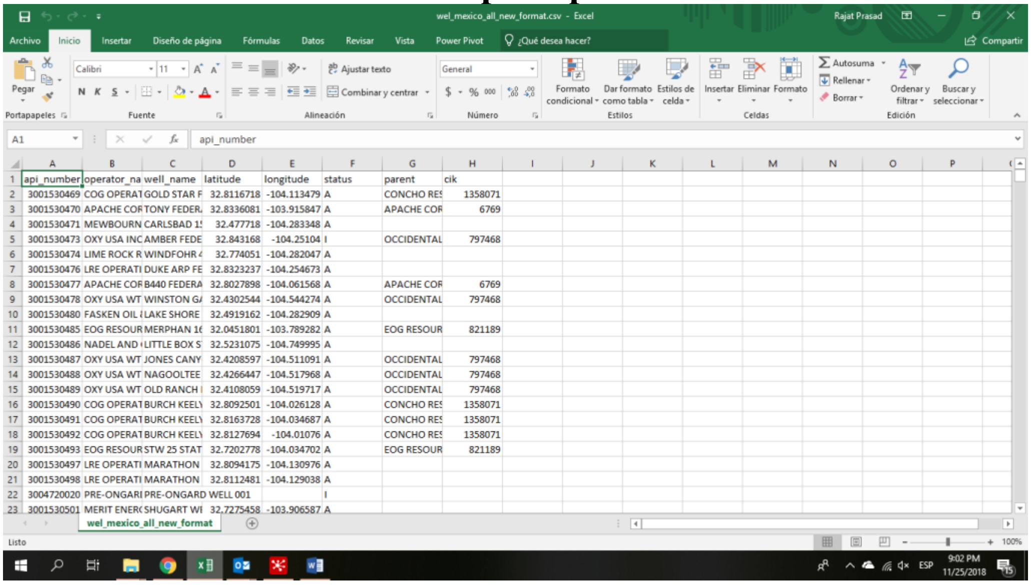


Figure 5 - Cleaning Flow Overview

In this transformation we took production_mexico_all_new_format.csv as the input file, removed all the rows that lacked the API number null using row filter. After this step all the rows that did not had their month column null were removed using filter row function. Next, we renamed the column month to date. All the rows in which oil, gas, water were null were filled with 0. All these rows were sorted according to API NUMBER. The source file was added to each row in a new column called SOURCE. This tells the stakeholder the origin of the data.

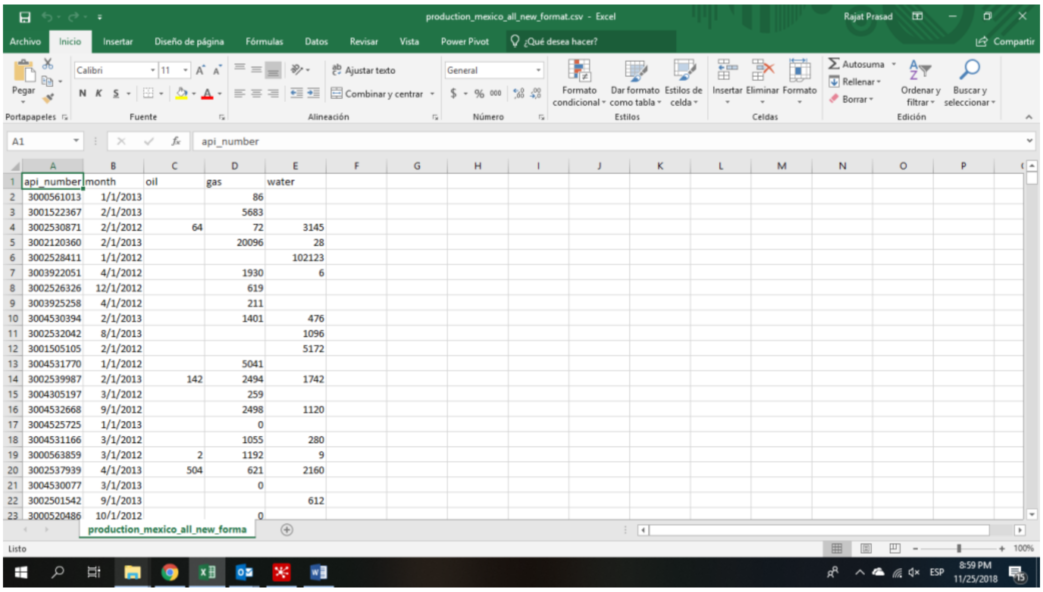
All this data was filled in a newly created CSV file called dataset_production.csv

Sample input



api_number	operator_na	well_name	latitude	longitude	status	parent	cik
3001530469	COG OPERAT	GOLD STAR F	32.8116718	-104.113479	A	CONCHO RES	1358071
3001530470	APACHE COR	TONY FEDER.	32.8336081	-103.915847	A	APACHE COR	6769
3001530471	MEWBOURN	CARLSBAD 1!	32.477718	-104.283348	A		
3001530473	OXY USA INC	AMBER FEDE	32.843168	-104.25104	I	OCCIDENTAL	797468
3001530474	LIME ROCK R	WINDFOHR 4	32.774051	-104.282047	A		
3001530476	LRE OPERATI	DUKE ARP FE	32.8323237	-104.254673	A		
3001530477	APACHE COR	B440 FEDERA	32.8027898	-104.061568	A	APACHE COR	6769
3001530478	OXY USA WT	WINSTON G/	32.4302544	-104.544274	A	OCCIDENTAL	797468
3001530480	FASKEN OIL {	LAKE SHORE	32.4919162	-104.282909	A		
3001530485	EOG RESOUR	MERPHAN 1(	32.0451801	-103.789282	A	EOG RESOUR	821189
3001530486	NADEL AND (	LITTLE BOX S	32.5231075	-104.749995	A		
3001530487	OXY USA WT	JONES CANY	32.4208597	-104.511091	A	OCCIDENTAL	797468
3001530488	OXY USA WT	NAGOOLTEE	32.4266447	-104.517968	A	OCCIDENTAL	797468
3001530489	OXY USA WT	OLD RANCH I	32.4108059	-104.519717	A	OCCIDENTAL	797468
3001530490	COG OPERAT	BURCH KEEL\	32.8092501	-104.026128	A	CONCHO RES	1358071
3001530491	COG OPERAT	BURCH KEEL\	32.8163728	-104.034687	A	CONCHO RES	1358071
3001530492	COG OPERAT	BURCH KEEL\	32.8127694	-104.01076	A	CONCHO RES	1358071
3001530493	EOG RESOUR	STW 25 STAT	32.7202778	-104.034702	A	EOG RESOUR	821189
3001530497	LRE OPERATI	MARATHON	32.8094175	-104.130976	A		
3001530498	LRE OPERATI	MARATHON	32.8112481	-104.129038	A		
3004720020	PRE-ONGARI	PRE-ONGARD WELL 001			I		
3001530501	MERIT ENER(	SHUGART WI	32.7275458	-103.906587	A		

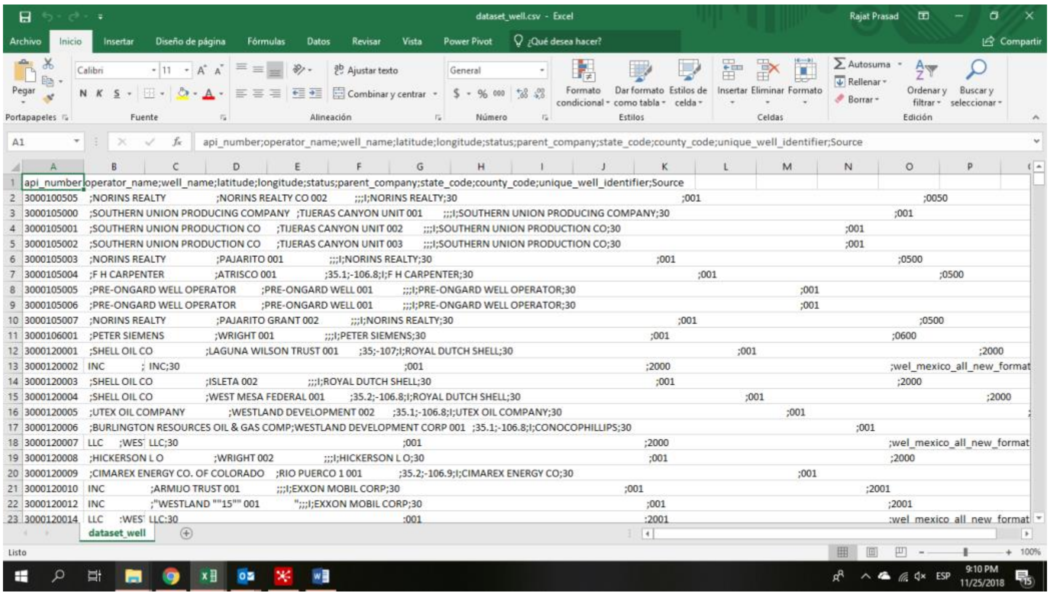
Figure 6 - wel_mexico_all_new_format.csv



api_number	month	oil	gas	water
3000561013	1/1/2013		86	
3001522367	2/1/2013		5683	
3002530871	2/1/2012	64	72	3145
3002120360	2/1/2013		20096	28
3002528411	1/1/2012			102123
3003922051	4/1/2012		1930	6
3002526326	12/1/2012		619	
3003925258	4/1/2012		211	
3004530394	2/1/2013		1401	476
3002532042	8/1/2013			1096
3001505105	2/1/2012			5172
3004531770	1/1/2012		5041	
3002539987	2/1/2013	142	2494	1742
3004305197	3/1/2012		259	
3004532668	9/1/2012		2498	1120
3004525725	1/1/2013		0	
3004531166	3/1/2012		1055	280
3000563859	3/1/2012	2	1192	9
3002537939	4/1/2013	504	621	2160
3004530077	3/1/2013		0	
3002501542	9/1/2013			612
3000520486	10/1/2012		0	

Figure 7 - Production_mexico_all_new_format.csv

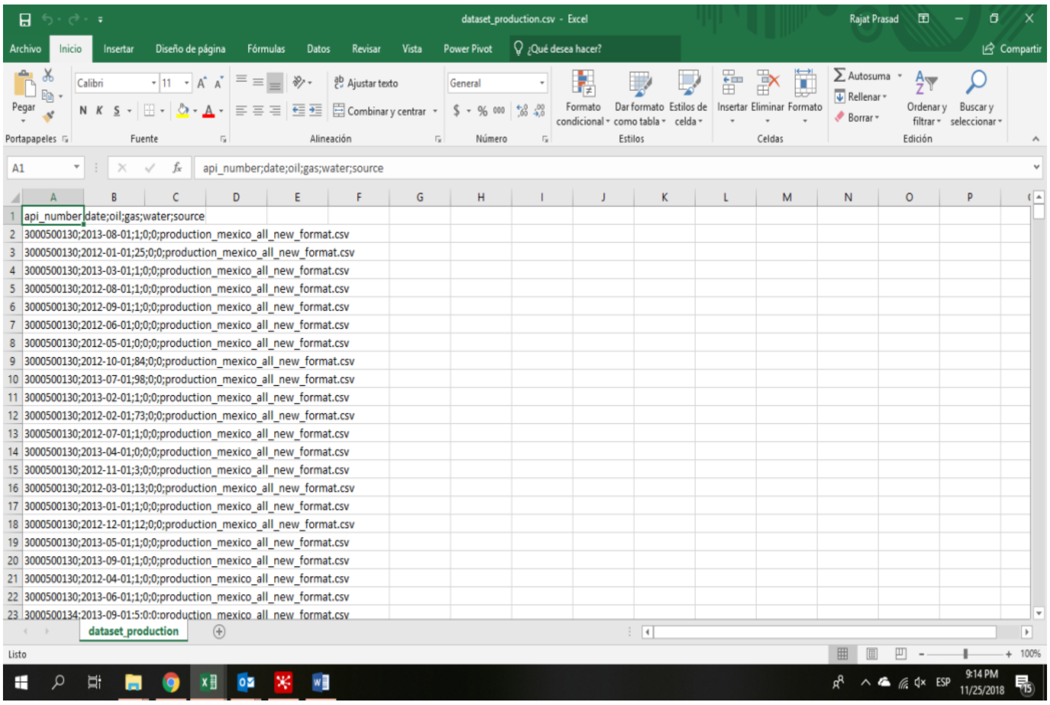
Sample Output



```
api_number;operator_name;well_name;latitude;longitude;status;parent_company;state_code;county_code;unique_well_identifier;Source

1  api_number operator_name;well_name;latitude;longitude;status;parent_company;state_code;county_code;unique_well_identifier;Source
2  3000100505 ;NORINS REALTY           ;NORINS REALTY CO 002      ;;;I;NORINS REALTY;30                         ;001                    ;0050
3  3000105000 ;SOUTHERN UNION PRODUCING COMPANY ;TIJERAS CANYON UNIT 001  ;;;I;SOUTHERN UNION PRODUCING COMPANY;30          ;001
4  3000105001 ;SOUTHERN UNION PRODUCTION CO   ;TIJERAS CANYON UNIT 002   ;;;I;SOUTHERN UNION PRODUCTION CO;30        ;001
5  3000105002 ;SOUTHERN UNION PRODUCTION CO   ;TIJERAS CANYON UNIT 003   ;;;I;SOUTHERN UNION PRODUCTION CO;30        ;001
6  3000105003 ;NORINS REALTY           ;PAJARITO 001          ;;;I;NORINS REALTY;30                   ;001                  ;0500
7  3000105004 ;F H CARPENTER           ;ATRISCO 001           ;35.1;-106.8;I;F H CARPENTER;30                ;001                  ;0500
8  3000105005 ;PRE-ONGARD WELL OPERATOR    ;PRE-ONGARD WELL 001      ;;;I;PRE-ONGARD WELL OPERATOR;30          ;001
9  3000105006 ;PRE-ONGARD WELL OPERATOR    ;PRE-ONGARD WELL 001      ;;;I;PRE-ONGARD WELL OPERATOR;30          ;001
10 3000105007 ;NORINS REALTY           ;PAJARITO GRANT 002      ;;;I;NORINS REALTY;30                    ;001                 ;0500
11 3000106001 ;PETER SIEMENS           ;WRIGHT 001            ;;;I;PETER SIEMENS;30                   ;001                  ;0600
12 3000120001 ;SHELL OIL CO          ;LAGUNA WILSON TRUST 001   ;35;-107;I;ROYAL DUTCH SHELL;30                ;001                  ;2000
13 3000120002 INC         ; INC;30                                    ;001       ;2000             ;wel_mexico_all_new_format
14 3000120003 ;SHELL OIL CO          ;ISLETA 002           ;;;I;ROYAL DUTCH SHELL;30                  ;001                  ;2000
15 3000120004 ;SHELL OIL CO          ;WEST MESA FEDERAL 001     ;35.2;-106.8;I;ROYAL DUTCH SHELL;30              ;001                 ;2000
16 3000120005 ;UTEX OIL COMPANY          ;WESTLAND DEVELOPMENT 002    ;35.1;-106.8;I;UTEX OIL COMPANY;30              ;001                 ;
17 3000120006 ;BURLINGTON RESOURCES OIL & GAS COMP;WESTLAND DEVELOPMENT CORP 001 ;35.1;-106.8;I;CONOCOPHILLIPS;30       ;001
18 3000120007 LLC   ;WES  LLC;30                                    ;001       ;2000             ;wel_mexico_all_new_format
19 3000120008 ;HICKERSON L O          ;WRIGHT 002            ;;;I;HICKERSON L O;30                   ;001                  ;2000
20 3000120009 ;CIMAREX ENERGY CO. OF COLORADO  ;RIO PUERCO 1 001        ;35.2;-106.9;I;CIMAREX ENERGY CO;30            ;001
21 3000120010 INC          ;ARMIJO TRUST 001        ;;;I;EXXON MOBIL CORP;30                ;001                  ;2001
22 3000120012 INC          ;"WESTLAND ""15"" 001        ";;;I;EXXON MOBIL CORP;30               ;001                  ;2001
23 3000120014 LLC   ;WES  LLC;30                                    ;001       ;2001             ;wel_mexico_all_new_format
```

Figure 8 - dataset_well.csv



```
api_number;date;oil;gas;water;source

1  api_number date;oil;gas;water;source
2  3000500130;2013-08-01;1;0;0;production_mexico_all_new_format.csv
3  3000500130;2012-01-01;25;0;0;production_mexico_all_new_format.csv
4  3000500130;2013-03-01;1;0;0;production_mexico_all_new_format.csv
5  3000500130;2012-08-01;1;0;0;production_mexico_all_new_format.csv
6  3000500130;2012-09-01;1;0;0;production_mexico_all_new_format.csv
7  3000500130;2012-06-01;0;0;0;production_mexico_all_new_format.csv
8  3000500130;2012-05-01;0;0;0;production_mexico_all_new_format.csv
9  3000500130;2012-10-01;84;0;0;production_mexico_all_new_format.csv
10 3000500130;2013-07-01;98;0;0;production_mexico_all_new_format.csv
11 3000500130;2013-02-01;1;0;0;production_mexico_all_new_format.csv
12 3000500130;2012-02-01;73;0;0;production_mexico_all_new_format.csv
13 3000500130;2012-07-01;1;0;0;production_mexico_all_new_format.csv
14 3000500130;2013-04-01;0;0;0;production_mexico_all_new_format.csv
15 3000500130;2012-11-01;3;0;0;production_mexico_all_new_format.csv
16 3000500130;2012-03-01;13;0;0;production_mexico_all_new_format.csv
17 3000500130;2013-01-01;1;0;0;production_mexico_all_new_format.csv
18 3000500130;2012-12-01;12;0;0;production_mexico_all_new_format.csv
19 3000500130;2013-05-01;1;0;0;production_mexico_all_new_format.csv
20 3000500130;2013-09-01;1;0;0;production_mexico_all_new_format.csv
21 3000500130;2012-04-01;1;0;0;production_mexico_all_new_format.csv
22 3000500130;2013-06-01;1;0;0;production_mexico_all_new_format.csv
23 3000500134;2013-09-01;5;0;0;production_mexico_all_new_format.csv
```

Figure 9 - dataset_production.csv

II STEP. TR_DATE

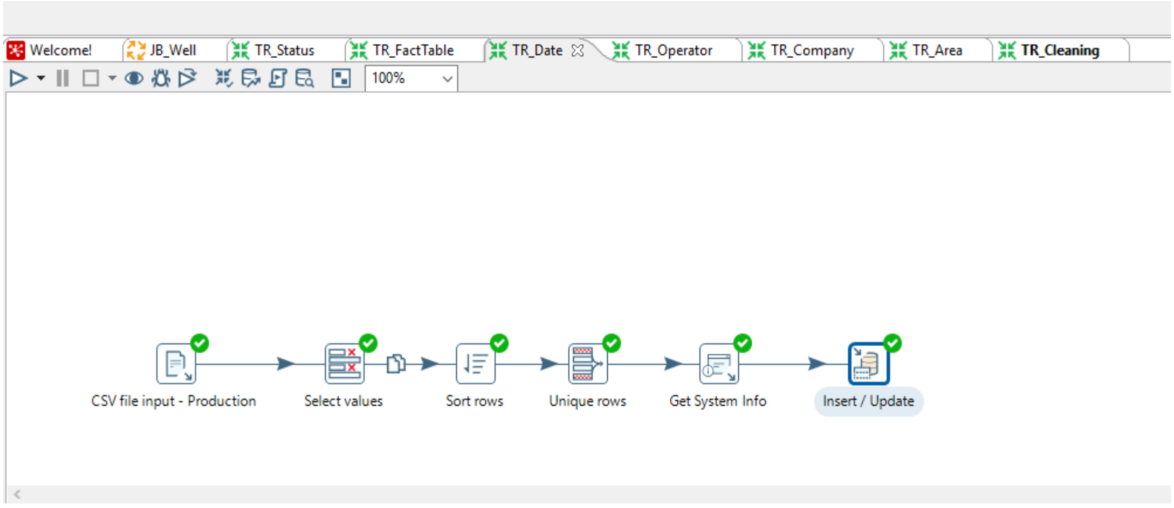
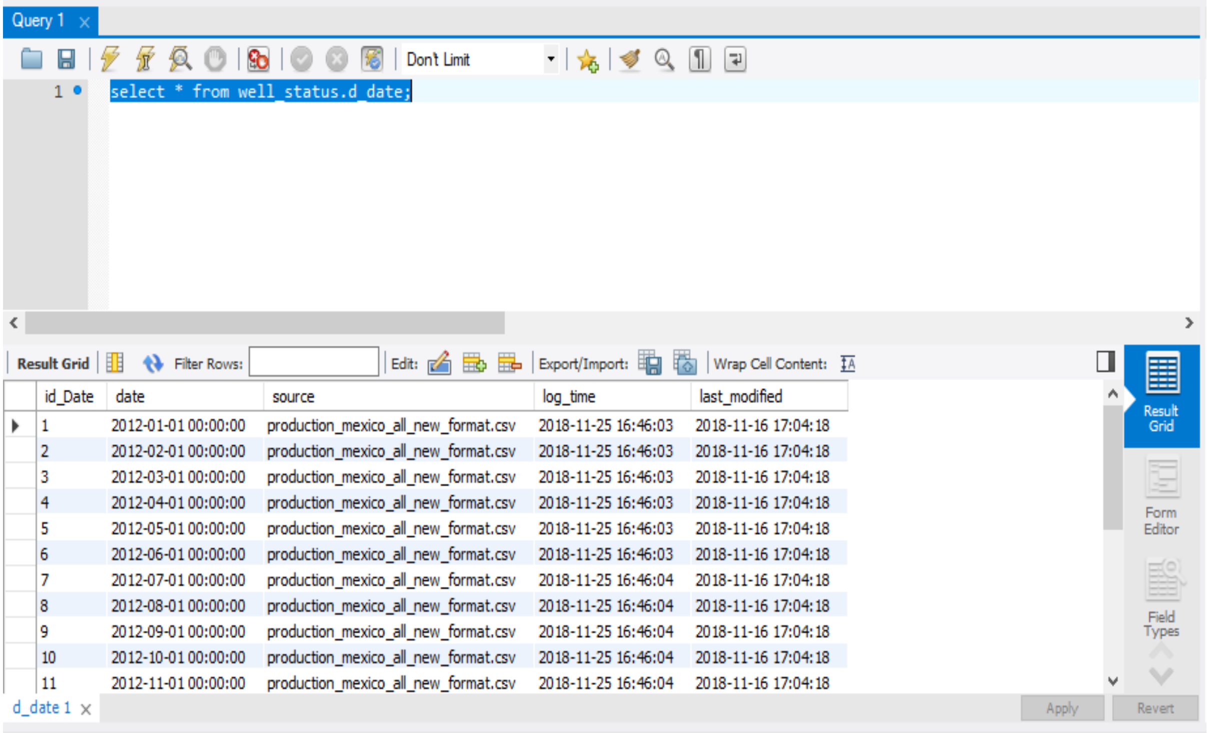


Figure 10 - Date Transformation

This is the second transformation in the ETL process, in first process we read CSV input file from dataset_production.csv, then we remove all the rows that we don't want to keep in our dimension. We want a dimension that only has date, source and meta data. So, in select values step we remove all columns except date and source. Then we sort all the rows according to date in Ascending order. We append log data such as last login, last modified. The last step is to insert all this information using insert/update in a table called **d_date** found in schema called **well_status**.

Sample Output



```
select * from well_status.d_date;
```

id_Date	date	source	log_time	last_modified
1	2012-01-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:03	2018-11-16 17:04:18
2	2012-02-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:03	2018-11-16 17:04:18
3	2012-03-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:03	2018-11-16 17:04:18
4	2012-04-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:03	2018-11-16 17:04:18
5	2012-05-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:03	2018-11-16 17:04:18
6	2012-06-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:03	2018-11-16 17:04:18
7	2012-07-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:04	2018-11-16 17:04:18
8	2012-08-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:04	2018-11-16 17:04:18
9	2012-09-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:04	2018-11-16 17:04:18
10	2012-10-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:04	2018-11-16 17:04:18
11	2012-11-01 00:00:00	production_mexico_all_new_format.csv	2018-11-25 16:46:04	2018-11-16 17:04:18

Figure 11 - Sample Output Dimension Date

III STEP. TR_COMPANY

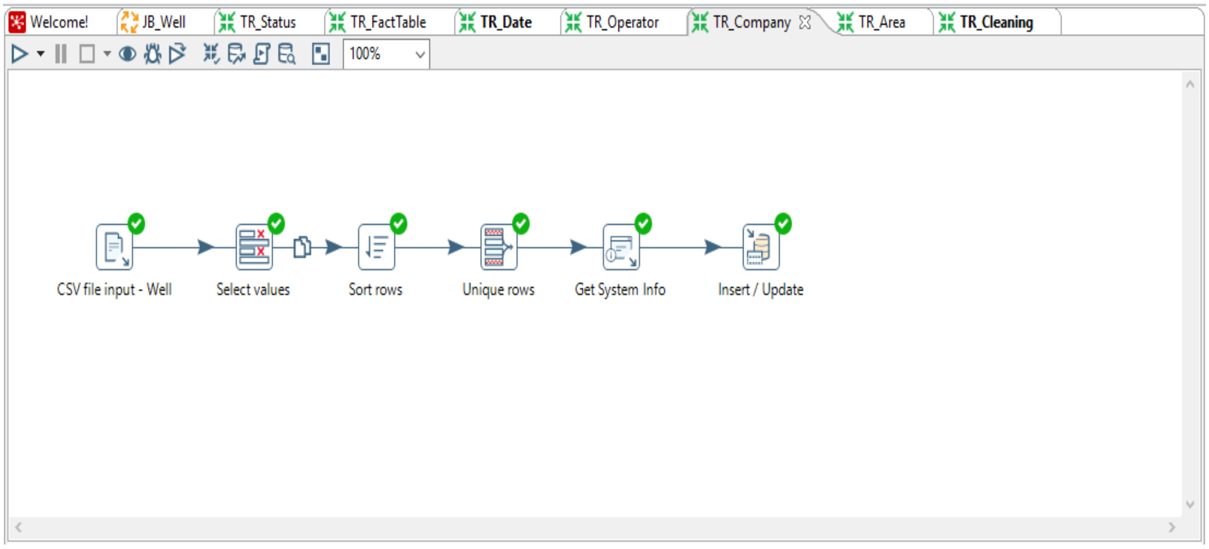
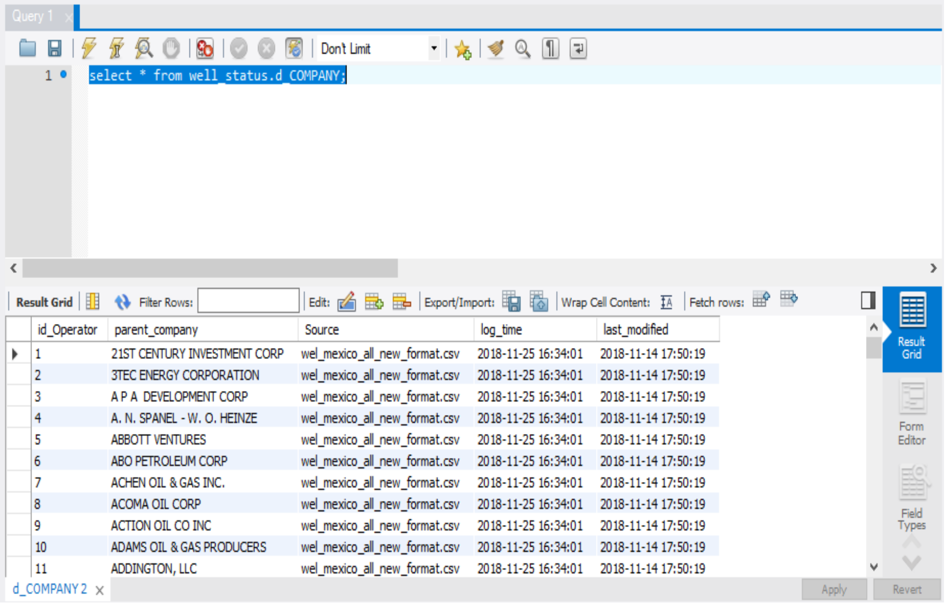


Figure 12 - Company Transformation

This is the third transformation in the ETL process, in first step we read CSV input file from dataset_well.csv, then we remove all the rows that we don't want to keep in our dimension. We want a dimension that only has **parent_company**, source and meta data. So, in select values step we remove all columns except **parent_compant** and source. Then we sort all the rows according to date in Ascending order and remove duplicate rows. We append log data such as last login, last modified. The last step is to insert all this information using insert/update in a table called **d_company** found in schema called **well_status**.

Sample Output



```
select * from well_status.d_COMPANY;
```

	id_Operator	parent_company	Source	log_time	last_modified
▶	1	21ST CENTURY INVESTMENT CORP	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	2	3TEC ENERGY CORPORATION	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	3	A P A DEVELOPMENT CORP	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	4	A. N. SPANEL - W. O. HEINZE	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	5	ABBOTT VENTURES	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	6	ABO PETROLEUM CORP	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	7	ACHEN OIL & GAS INC.	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	8	ACOMA OIL CORP	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	9	ACTION OIL CO INC	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	10	ADAMS OIL & GAS PRODUCERS	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19
	11	ADDINGTON, LLC	wel_mexico_all_new_format.csv	2018-11-25 16:34:01	2018-11-14 17:50:19

Figure 13 - Sample Output Dimension Company

IV STEP. TR_OPERATOR

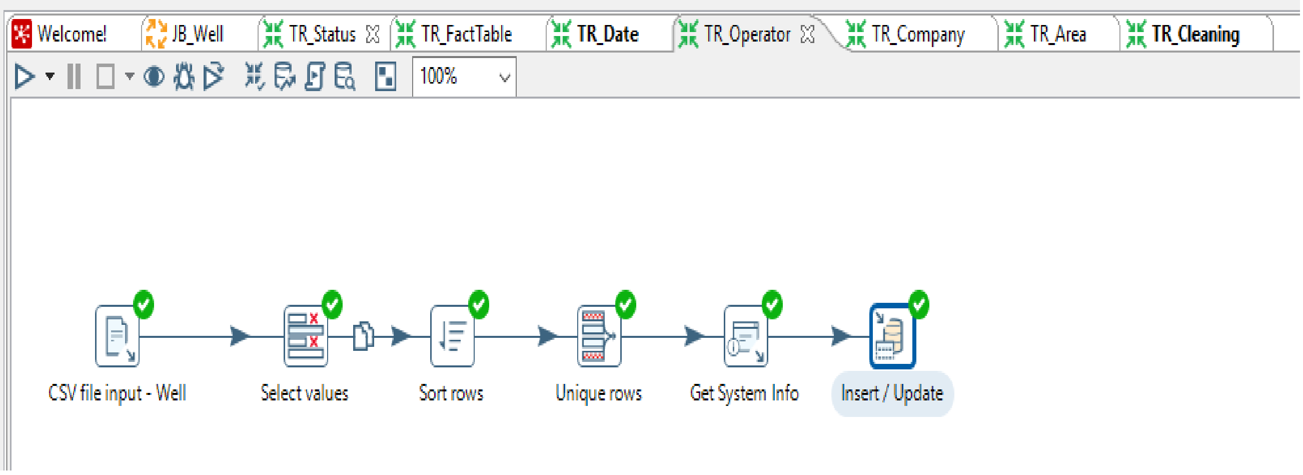
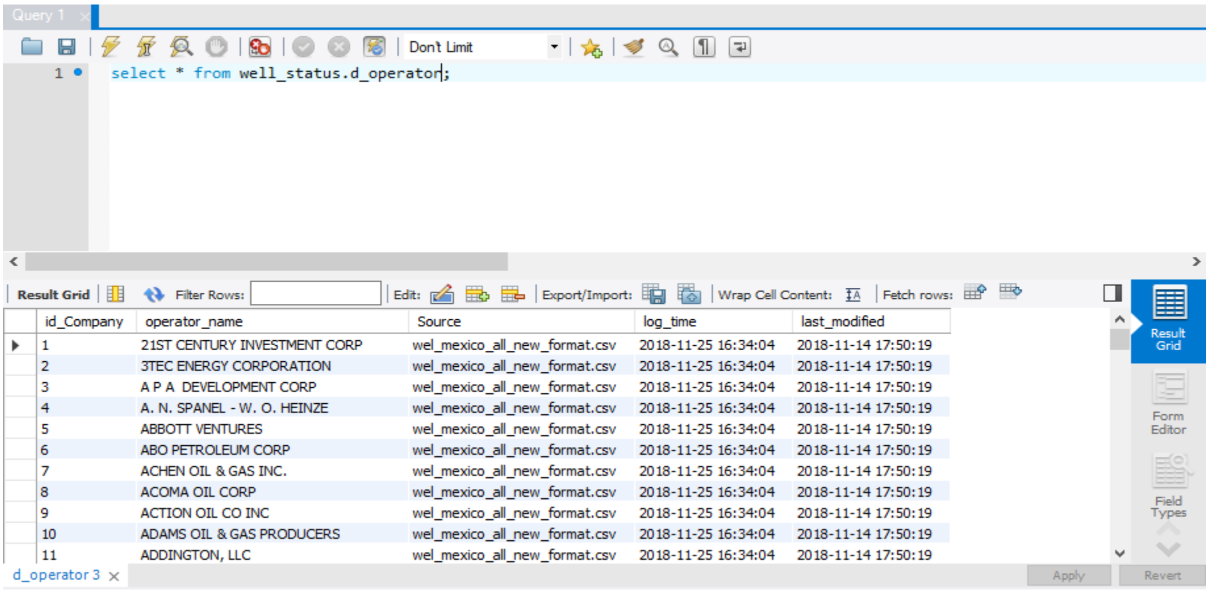


Figure 14 - Transformation Operator

This is the fourth transformation in the ETL process, in first step we read CSV input file from dataset_well.csv, then we remove all the rows that we don't want to keep in our dimension. We want a dimension that only has operator, source and meta data. So, in select values step we remove all columns except operator and source. Then we sort all the rows according to operator in Ascending order. We removed all the duplicate rows in operator column. We append log data such as last login, last modified. The last step is to insert all this information using insert/update in a table called **d_operator** found in schema called **well_status.**

Sample Output

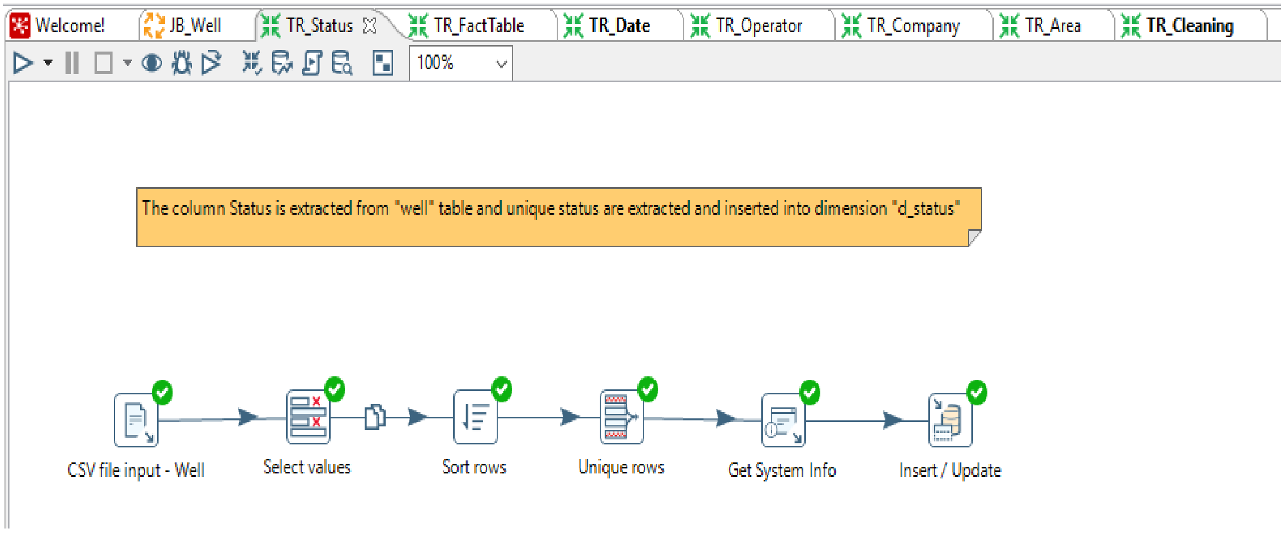


```
select * from well_status.d_operator;
```

id_Company	operator_name	Source	log_time	last_modified
1	21ST CENTURY INVESTMENT CORP	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
2	3TEC ENERGY CORPORATION	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
3	A P A DEVELOPMENT CORP	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
4	A. N. SPANEL - W. O. HEINZE	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
5	ABBOTT VENTURES	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
6	ABO PETROLEUM CORP	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
7	ACHEN OIL & GAS INC.	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
8	ACOMA OIL CORP	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
9	ACTION OIL CO INC	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
10	ADAMS OIL & GAS PRODUCERS	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19
11	ADDINGTON, LLC	wel_mexico_all_new_format.csv	2018-11-25 16:34:04	2018-11-14 17:50:19

Figure 15 - Sample Output Dimension Operator

V STEP. TR_STATUS

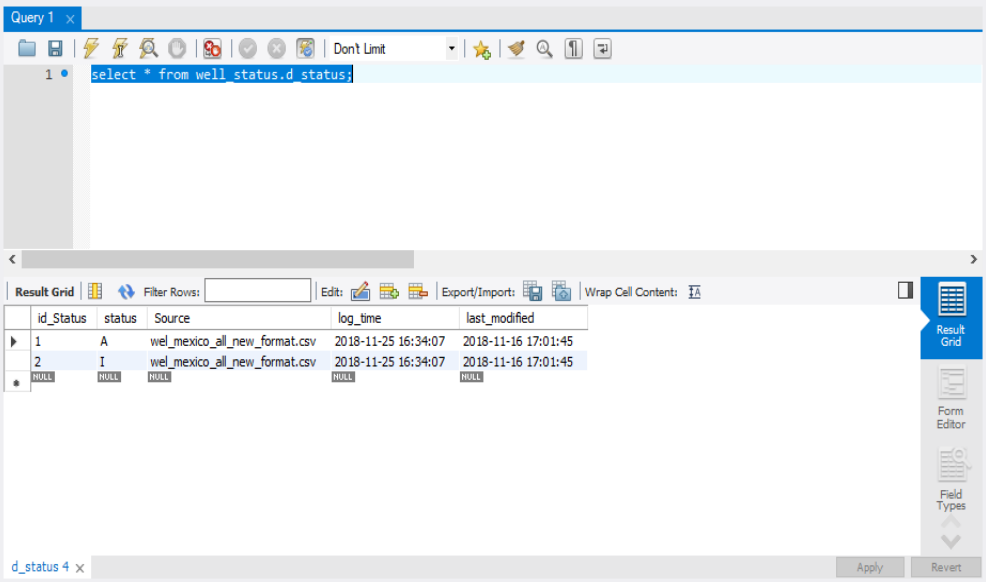


The column Status is extracted from "well" table and unique status are extracted and inserted into dimension "d_status"

Figure 16 - Transformation Status

This is the fifth transformation in the ETL process, in first step we read CSV input file from dataset_well.csv, then we remove all the rows that we don't want to keep in our dimension. We want a dimension that only has status, source and meta data. So, in select values step we remove all columns except status and source. Then we sort all the rows according to status in Ascending order. We remove duplicate rows in status column, We append log data such as last login, last modified. The last step is to insert all these information using insert/update in a table called d_status found in schema called well_status.

Sample Output

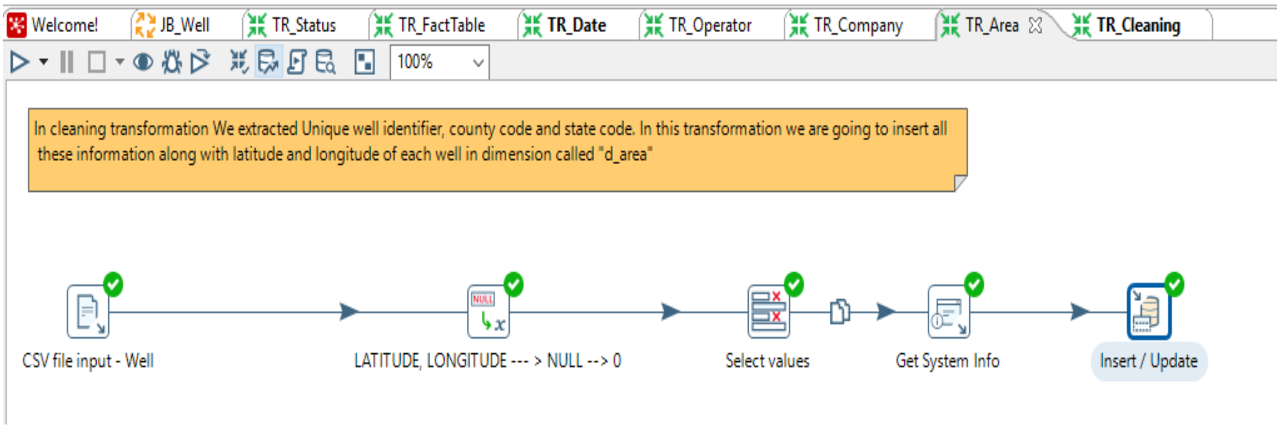


```
select * from well_status.d_status;
```

id_Status	status	Source	log_time	last_modified
1	A	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-16 17:01:45
2	I	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-16 17:01:45
NULL	NULL	NULL	NULL	NULL

Figure 17 - Sample Output Dimension Status

VI STEP. TR_AREA

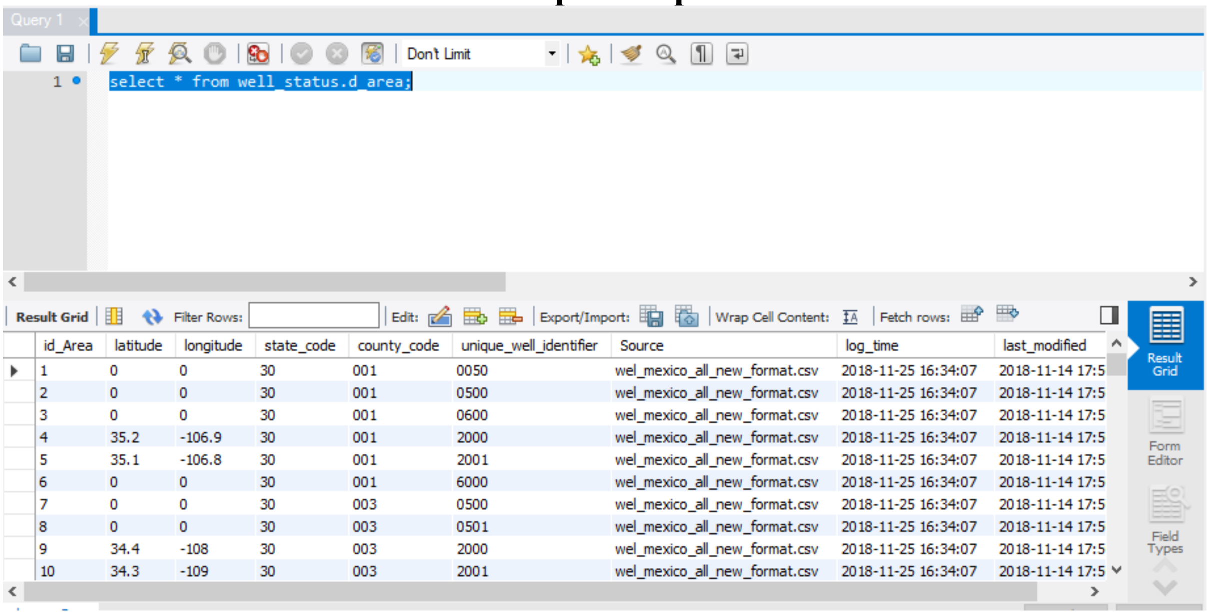


In cleaning transformation We extracted Unique well identifier, county code and state code. In this transformation we are going to insert all these information along with latitude and longitude of each well in dimension called "d_area"

Figure 18 - Transformation Area

This is the sixth transformation in the ETL process, in first step we read CSV input file from dataset_well.csv, then we remove all the rows that we don’t want to keep in our dimension. We want a dimension that only has latitude, longitude, state code, unique well identifier, source and meta data. So, in select values step we remove all columns except latitude, longitude, state code, unique well identifier, source and meta data. We append log data such as last login, last modified. The last step is to insert all these information using insert/update in a table called d_area found in schema called well_status.

Sample Output



```
select * from well_status.d_area;
```

id_Area	latitude	longitude	state_code	county_code	unique_well_identifier	Source	log_time	last_modified
1	0	0	30	001	0050	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
2	0	0	30	001	0500	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
3	0	0	30	001	0600	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
4	35.2	-106.9	30	001	2000	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
5	35.1	-106.8	30	001	2001	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
6	0	0	30	001	6000	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
7	0	0	30	003	0500	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
8	0	0	30	003	0501	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
9	34.4	-108	30	003	2000	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5
10	34.3	-109	30	003	2001	wel_mexico_all_new_format.csv	2018-11-25 16:34:07	2018-11-14 17:5

Figure 19 - Sample Output Dimension Area

VII STEP. TR_FACT_TABLE

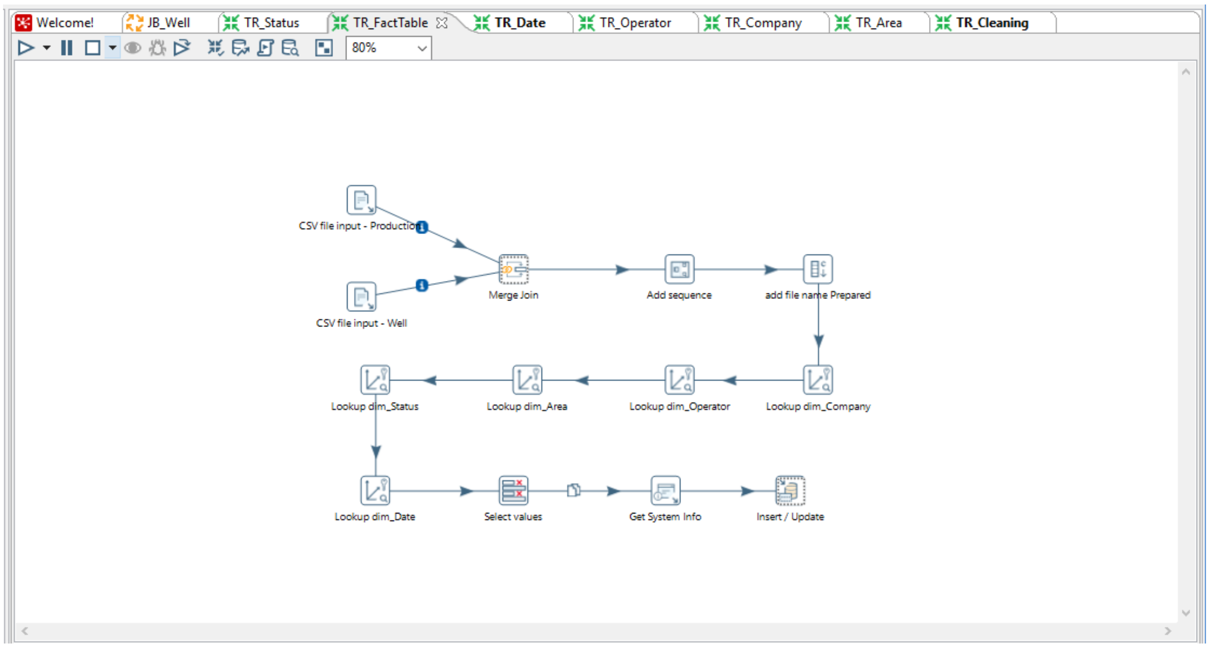


Figure 20 - Transformation Fact Table

In this Transformation we are going to populate the fact table with variable important to the Business needs and foreign keys pointing to other dimensions to provide us extra information. First, we read two CSV files and merged them using **merge join.** The two files we joined were dataset_production.csv and dateset_well.csv. As these two have API NUMBER common we were able to perform inner join. This step is important as we will need variables to look up primary keys in relevant dimensions. We added file name as source using "add filename prepared" this added column will help the stakeholders to know the source of data.

The next step was to perform lookup in the **D_company** table to map **"parent_company "** attribute which is present in both CSV table and **D_company** dimension and by doing that we can add the primary key of **d_company** to current CSV file called **id_operator**.

The next step was to perform lookup in the **D_operator** table to map **"operator_name"** attribute which is present in both CSV table and **D_operator** dimension and by doing that we can add the primary key of **d_operator** to current CSV file called **id_company**.

The next step was to perform lookup in the **D_AREA** table to map **“state_code“** and **“county_code”** attribute which is present in both CSV table and **D_AREA** dimension and by doing that we can add the primary key of **d_AREA** to current CSV file called **id_area**.

The next step was to perform lookup in the **D_STATUS** table to map “**STATUS**“ attribute which is present in both CSV table and **D_STATUS** dimension and by doing that we can add the primary key of **d_STATUS** to current CSV file called **id_status**.

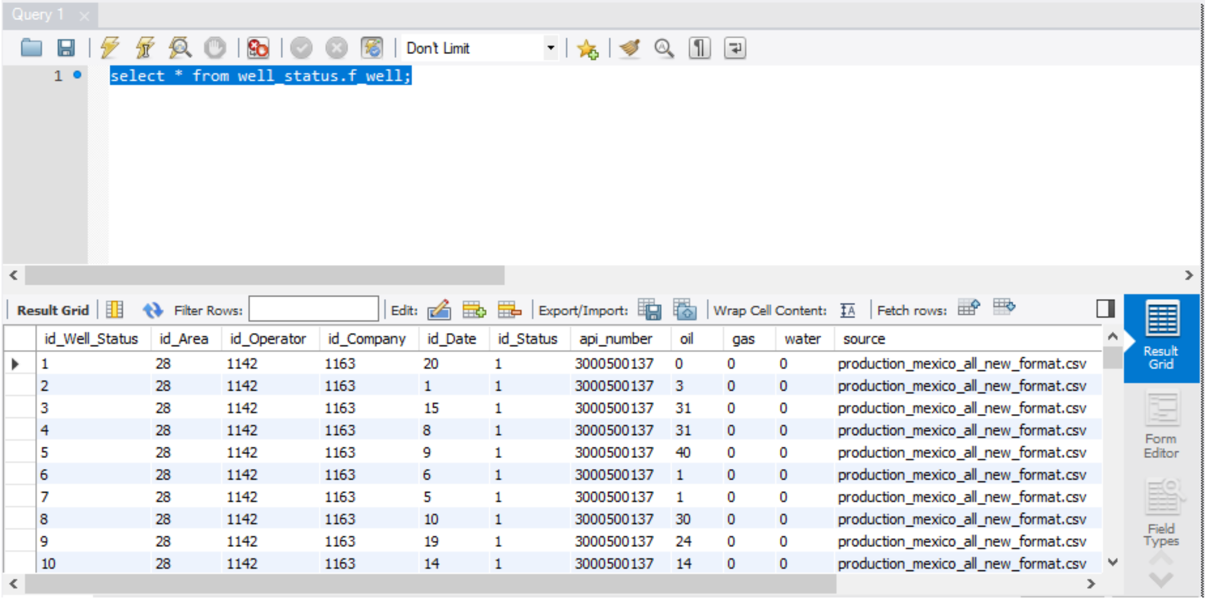
The next step was to perform lookup in the **D_DATE** table to map “**DATE**“ attribute which is present in both CSV table and **D_DATE** dimension and by doing that we can add the primary key of **d_DATE** to current CSV file called id_date.

In select/Rename values we only **select id_Area, id_company, id_date, id_operator, id_status, gas, oil, water, source, api_number**. We remove the other columns as they don’t help in achieving business goals.

We now append the system log time and time of modification of the transformation so that the stakeholder can know at what time the data was entered into the system.

The ultimate step is to connect to database and to fact table called f_well in the well_status schema and populate the values

Sample Output



```
select * from well_status.f_well;
```

id_Well_Status	id_Area	id_Operator	id_Company	id_Date	id_Status	api_number	oil	gas	water	source
1	28	1142	1163	20	1	3000500137	0	0	0	production_mexico_all_new_format.csv
2	28	1142	1163	1	1	3000500137	3	0	0	production_mexico_all_new_format.csv
3	28	1142	1163	15	1	3000500137	31	0	0	production_mexico_all_new_format.csv
4	28	1142	1163	8	1	3000500137	31	0	0	production_mexico_all_new_format.csv
5	28	1142	1163	9	1	3000500137	40	0	0	production_mexico_all_new_format.csv
6	28	1142	1163	6	1	3000500137	1	0	0	production_mexico_all_new_format.csv
7	28	1142	1163	5	1	3000500137	1	0	0	production_mexico_all_new_format.csv
8	28	1142	1163	10	1	3000500137	30	0	0	production_mexico_all_new_format.csv
9	28	1142	1163	19	1	3000500137	24	0	0	production_mexico_all_new_format.csv
10	28	1142	1163	14	1	3000500137	14	0	0	production_mexico_all_new_format.csv

Figure 21 - Sample Output Fact Table