

## Dimensional Lookup / Tableau KPI

**Overview:** This brief series of exercises were done to put into practice various skills learned about in Data Warehousing. Essentially, I transformed an OLTP database into a dimensional model (OLAP) by building an ETL system. I took a simple dataset containing Sales Representative info, built an ETL Transformation to load and manage my data, created a Dimensional Lookup/Update table, tested various changes, then loaded the final change into Tableau and visualized two KPIs. **These exercises were done by following Richard Holowczak's tutorials.**

1. This step was to create a Dimensional Lookup table after importing the [ Salesrep ] data. I first added this new transformation, imported the data, made sure each field is in the correct Type, Format, and Length. Then added the [ Salesrep ] dimension, connected the output to input, set the first table as the target table, and set the field source data and target dimension. Then I ran and reviewed the data in SQL Server.

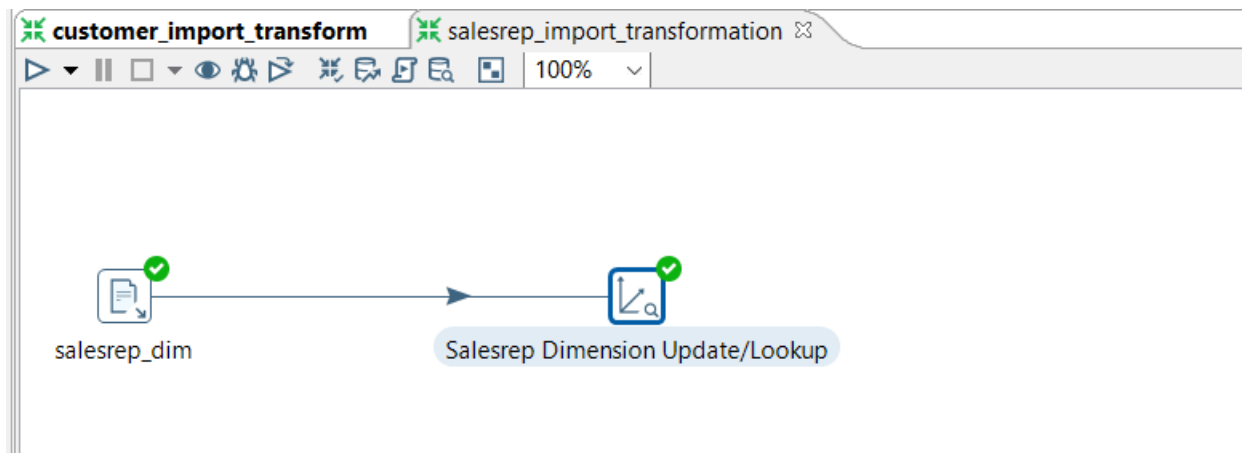


Figure 1 - Transformation Screen

Execution Results												
Logging Execution History Step Metrics Performance Graph Metrics Preview data												
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	salesrep_dim	0	0	33	34	0	0	0	0	Finished	0.0s	773
2	Salesrep Dimension Update/Lookup	0	33	33	33	33	0	0	0	Finished	3.1s	10

Figure 2 - Step Metrics

Query Result: \* All Rows Fetched: 34 in 0.028 seconds

	SALESREP_DIM_ID	VERSION	DATE_FROM	DATE_TO	SALESREP_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	0	1	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
2	1	1	101-JAN-00	31-DEC-99	147	Alberto	Erasmus	AERASUR	011.44.1344.429276	10-MAR-06	SA_REP	12000	0.3	100	80
3	2	1	101-JAN-00	31-DEC-99	148	Gerald	Cambraut	GCAMBRAU	011.44.1344.619269	15-OCT-07	SA_REP	11000	0.3	100	80
4	3	1	101-JAN-00	31-DEC-99	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-08	SA_REP	10500	0.2	100	80
5	4	1	101-JAN-00	31-DEC-99	150	Peter	Tucker	PTUCKER	011.44.1344.129269	30-JAN-05	SA_REP	10000	0.3	145	80
6	5	1	101-JAN-00	31-DEC-99	151	David	Bernstein	DBERNSTE	011.44.1344.345269	24-MAR-05	SA_REP	9500	0.25	145	80
7	6	1	101-JAN-00	31-DEC-99	152	Peter	Hall	PHALL	011.44.1344.478968	20-AUG-05	SA_REP	9000	0.25	145	80
8	7	1	101-JAN-00	31-DEC-99	153	Christopher	Olsen	COLSEN	011.44.1344.498718	30-MAR-06	SA_REP	8000	0.2	145	80
9	8	1	101-JAN-00	31-DEC-99	154	Janette	Cambraut	NCAMBRAU	011.44.1344.987668	09-DEC-06	SA_REP	7500	0.2	145	80
10	9	1	101-JAN-00	31-DEC-99	155	Oliver	Tuvault	OTUVAULT	011.44.1344.486508	23-NOV-07	SA_REP	7000	0.15	145	80
11	10	1	101-JAN-00	31-DEC-99	156	Janette	King	JKING	011.44.1345.429268	30-JAN-04	SA_REP	10000	0.35	146	80
12	11	1	101-JAN-00	31-DEC-99	157	Patrick	Sully	PSULLY	011.44.1345.929268	04-MAR-04	SA_REP	9500	0.35	146	80
13	12	1	101-JAN-00	31-DEC-99	158	Allan	McEwen	AMCEWEN	011.44.1345.829268	01-AUG-04	SA_REP	9000	0.35	146	80
14	13	1	101-JAN-00	31-DEC-99	159	Lindsey	Smith	LSMITH	011.44.1345.729268	10-MAR-05	SA_REP	8000	0.3	146	80
15	14	1	101-JAN-00	31-DEC-99	160	Louise	Doran	LDORAN	011.44.1345.629268	15-DEC-05	SA_REP	7500	0.3	146	80
16	15	1	101-JAN-00	31-DEC-99	161	Sarah	Sewall	SSEWALL	011.44.1345.529268	03-NOV-06	SA_REP	7000	0.25	146	80
17	16	1	101-JAN-00	31-DEC-99	162	Clara	Viahney	CVIAHNEY	011.44.1346.129268	11-NOV-05	SA_REP	10500	0.25	147	80
18	17	1	101-JAN-00	31-DEC-99	163	Danielle	Greene	DGREENE	011.44.1346.229268	19-MAR-07	SA_REP	9500	0.15	147	80

Figure 3 - Resulting Data

- After importing the data file and establishing the proper Dimension Table, this step was to implement a Type 1 SCD. I performed the same steps as before but changed the Type of Dimension Update to “Punch through”. This allows a Type 1 SCD on these fields.

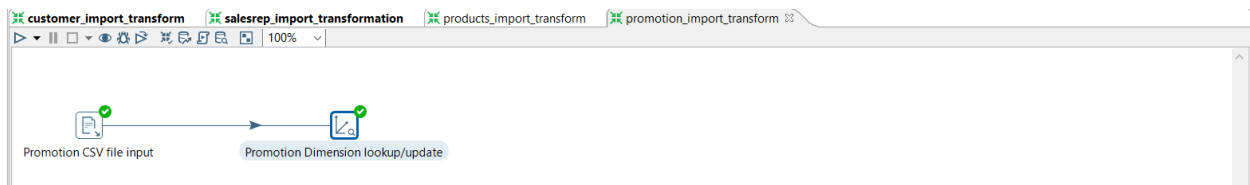


Figure 4 – Transformation Screen

Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Promotion CSV file input	0	0	2	3	0	0	0	0	Finished	0.0s	250	-
2	Promotion Dimension lookup/update	0	2	2	2	2	0	0	0	Finished	0.3s	6	-

Figure 5 - Step Metrics

Query Result: \* All Rows Fetched: 3 in 0.023 seconds

	PROMOTION_DIM_ID	VERSION	DATE_FROM	DATE_TO	PROMO_ID	PROMO_NAME
1	0	1	(null)	(null)	(null)	(null)
2	1	1	101-JAN-00	31-DEC-99	1	everyday low price
3	2	1	101-JAN-00	31-DEC-99	2	blowout sale

Figure 6 - Resulting Data

- For this step I had to embellish a created Date Dimension with additional date attributes. To generate the Date Dimension it required first to create a table with the Start Date, link to a table that incrementally generated dates, link to an additional table that calculated the dates, then to a table that selected the values. After creating the various tables, I created one more table to calculate the additional fields and linked that to the final Dimensional Lookup.

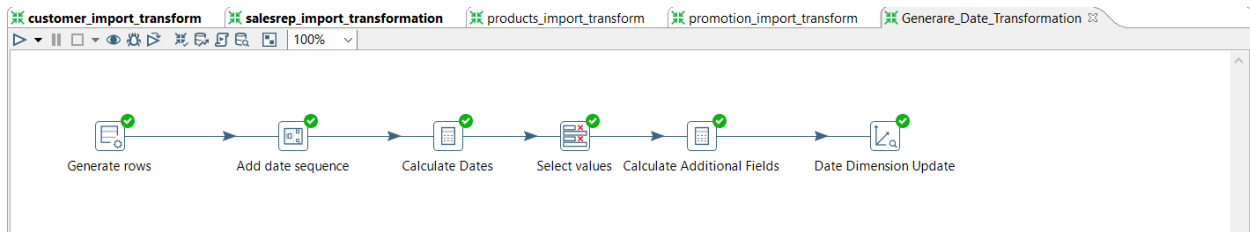


Figure 7 – Transformation Screen

Execution Results												
Logging Execution History Step Metrics Performance Graph Metrics Preview data												
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Generate rows	0	0	1100	0	0	0	0	0	Finished	0.0s	73,333
2	Add date sequence	0	1100	1100	0	0	0	0	0	Finished	0.0s	45,833
3	Calculate Dates	0	1100	1100	0	0	0	0	0	Finished	0.1s	12,941
4	Select values	0	1100	1100	0	0	0	0	0	Finished	0.2s	7,285
5	Calculate Additional Fields	0	1100	1100	0	0	0	0	0	Finished	0.2s	5,392
6	Date Dimension Update	0	1100	1100	1100	1100	0	0	0	Finished	1mn 3s	17

Figure 8 - Step Metrics

Execution Results									
Logging Execution History Step Metrics Performance Graph Metrics Preview data									
First rows Last rows Off									
#	Sales_Date	sales_day_of_year	sales_month	sales_year	sales_quarter	sales_month_name	sales_day_of_week_name	sales_day_of_week	sales_day_of_month
1	01/02/2006	2	1	2006	1	January	Monday	2	1101
2	01/03/2006	3	1	2006	1	January	Tuesday	3	1102
3	01/04/2006	4	1	2006	1	January	Wednesday	4	1103
4	01/05/2006	5	1	2006	1	January	Thursday	5	1104
5	01/06/2006	6	1	2006	1	January	Friday	6	1105
6	01/07/2006	7	1	2006	1	January	Saturday	7	1106
7	01/08/2006	8	1	2006	1	January	Sunday	1	1107
8	01/09/2006	9	1	2006	1	January	Monday	2	1108
9	01/10/2006	10	1	2006	1	January	Tuesday	3	1109
10	01/11/2006	11	1	2006	1	January	Wednesday	4	1110

Figure 9 - Resulting Data

- Here I had to embellish the Date Dimension with additional Holiday info. I loaded this data into a table, connected it separately to a created Stream Lookup table to actually check the various dates, then implemented these into the screen.

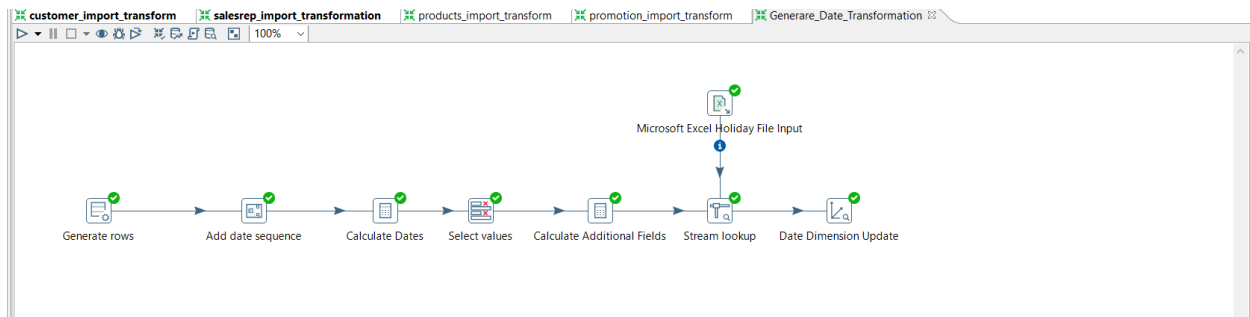


Figure 10 - Transaction Screen

**Execution Results**

Logging Execution History Step Metrics Performance Graph Metrics Preview data

#	Stepname	Copypnr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Generate rows	0	0	1100	0	0	0	0	0	Finished	0.0s	110,000	-
2	Add date sequence	0	1100	1100	0	0	0	0	0	Finished	0.1s	14,667	-
3	Calculate Dates	0	1100	1100	0	0	0	0	0	Finished	0.1s	12,088	-
4	Select values	0	1100	1100	0	0	0	0	0	Finished	1.3s	863	-
5	Microsoft Excel Holiday File Input	0	0	33	33	0	0	0	0	Finished	1.3s	26	-
6	Calculate Additional Fields	0	1100	1100	0	0	0	0	0	Finished	1.3s	861	-
7	Stream lookup	0	1133	1100	0	0	0	0	0	Finished	1.4s	785	-
8	Date Dimension Update	0	1100	1100	1100	0	1100	0	0	Finished	43.1s	26	-

Figure 11 - Step Metrics

**Execution Results**

Logging Execution History Step Metrics Performance Graph Metrics Preview data

First rows Last rows Off

#	Sales_Date	sales_day_of_year	sales_month	sales_year	sales_quarter	sales_month_name	sales_day_of_week_name	sales_day_of_week	sales_day_of_month	HOLIDAY_DESCRIPTION	IS_A_HOLIDAY	dt
1	01/02/2006	2	1	2006	1	January	Monday	2	2	New Year's Day	1.0	
2	01/03/2006	3	1	2006	1	January	Tuesday	3	3	<null>	0.0	
3	01/04/2006	4	1	2006	1	January	Wednesday	4	4	<null>	0.0	
4	01/05/2006	5	1	2006	1	January	Thursday	5	5	<null>	0.0	
5	01/06/2006	6	1	2006	1	January	Friday	6	6	<null>	0.0	
6	01/07/2006	7	1	2006	1	January	Saturday	7	7	<null>	0.0	
7	01/08/2006	8	1	2006	1	January	Sunday	1	8	<null>	0.0	
8	01/09/2006	9	1	2006	1	January	Monday	2	9	<null>	0.0	
9	01/10/2006	10	1	2006	1	January	Tuesday	3	10	<null>	0.0	

Figure 12 - Resulting Data

- This step required updating the set of transformations with new data. This happening after I fit the various transformations and dimensions together and creating a central fact table. I uploaded the data since I already set the transformations properly, it didn't require additional changes.

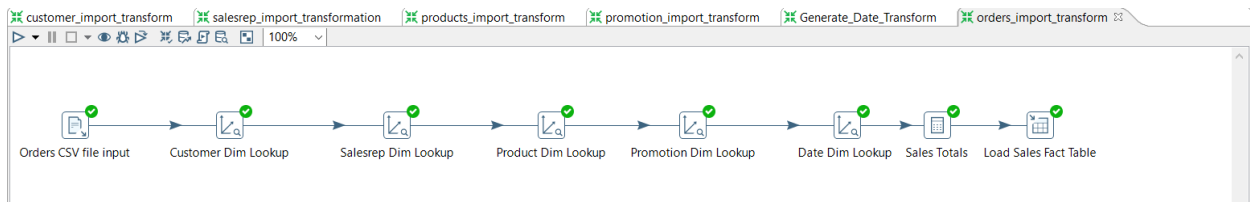


Figure 13 - Transaction Screen

#	Stepname	Copyn	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Orders CSV file input	0	0	18	19	0	0	0	0	Finished	0.0s	3,167	-
2	Customer Dim Lookup	0	18	18	18	0	0	0	0	Finished	0.4s	48	-
3	Salesrep Dim Lookup	0	18	18	18	0	0	0	0	Finished	0.4s	41	-
4	Product Dim Lookup	0	18	18	18	0	0	0	0	Finished	0.5s	36	-
5	Promotion Dim Lookup	0	18	18	18	0	0	0	0	Finished	0.6s	32	-
6	Date Dim Lookup	0	18	18	18	0	0	0	0	Finished	0.6s	31	-
7	Sales Totals	0	18	18	0	0	0	0	0	Finished	0.5s	36	-
8	Load Sales Fact Table	0	18	18	0	18	0	0	0	Finished	0.6s	28	-

Figure 14 - Step Metrics

#	ORDER_ID	ORDER_DATE	CUSTOMER_ID	ORDER_STATUS	ORDER_TOTAL	SALES_REP_ID	PROMO_ID	LINE_ITEM_ID	PRODUCT_ID	UNIT_PRICE	QUANTITY	customer_dim_id	salesre
1	2491	25-Oct-08	107	3	31574	160	0	1	3106	46	36	164	
2	2520	11-Nov-08	146	3	29249.1	<null>	0	1	2322	22	22	320	
3	2531	11-Nov-08	169	8	15760.5	156	1	1	3112	72	5	24	
4	2563	1-Nov-08	107	3	31574	160	0	1	3114	99	30	164	
5	2601	25-Sep-08	159	2	69286.4	161	0	1	2986	123	3	16	
6	2615	27-Oct-08	143	3	27132.6	<null>	0	1	3187	2.2	25	195	
7	2642	17-Nov-08	144	6	62303	159	0	1	2311	86.9	5	1	
8	2689	7-Oct-08	101	8	33893.6	161	1	1	2308	54	30	158	
9	2724	17-Nov-08	169	8	15760.5	156	0	1	3124	84	14	24	

Figure 15 - Resulting Data

## SQL For Joining and Creating View

```

CREATE VIEW All_tables AS

SELECT customer_dim.customer_id, customer_dim.cust_first_name, customer_dim.cust_last_name, customer_dim.street_address, customer_dim.postal_code, customer_dim.city, customer_dim.state,
salesrep_dim.first_name, salesrep_dim.last_name, salesrep_dim.email, salesrep_dim.phone_number, salesrep_dim.hire_date, salesrep_dim.job_id, salesrep_dim.salary, salesrep_dim.commissio
promotion_dim.promo_id, promotion_dim.promo_name, promotion_dim.promotion_dim_id,
products_dim.product_id, products_dim.product_name, products_dim.language_id, products_dim.min_price, products_dim.list_price, products_dim.product_status, products_dim.supplier_id, pr
date_dim.sales_date, date_dim.sales_day_of_year, date_dim.sales_month, date_dim.sales_year, date_dim.sales_quarter, date_dim.sales_month_name, date_dim.sales_day_of_week_name, date_dim
sales_fact.order_id, sales_fact.order_date, sales_fact.order_status, sales_fact.order_total, sales_fact.line_item_id, sales_fact.unit_price, sales_fact.quantity

FROM customer_dim
FULL JOIN salesrep_dim
ON customer_dim.customer_dim_id = salesrep_dim.salesrep_dim_id FULL JOIN
promotion_dim
ON salesrep_dim.salesrep_dim_id = promotion_dim.promotion_dim_id FULL JOIN
products_dim
ON promotion_dim.promotion_dim_id = products_dim.products_dim_id FULL JOIN
date_dim
ON products_dim.products_dim_id = date_dim.date_dim_id FULL JOIN
sales_fact
ON date_dim.date_dim_id = sales_fact.date_dim_id;

```

- Finally, I loaded my data into Tableau and tested various visualizations for my below KPIs.

### KPI Visualization

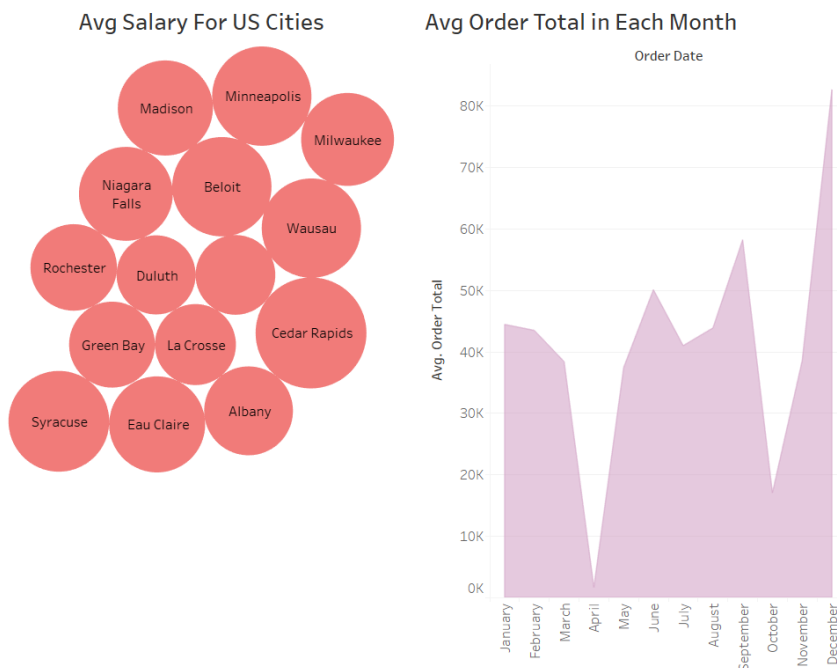


Figure 16 – KPIs \*Blank circle – Poughkeepsie (couldn't fit in screen)

**Conclusion:**

- Recently I started tracking my time using an app (Toggl) and decided it would be perfect to track every minute I worked on this set of exercises. What I didn't realize, until reviewing the data when I finally finished, was that this took way longer than I could've imagined. From opening the tutorial and starting, it took about 6hrs. This isn't including the Tableau portion which was about an additional 1hr. The most difficult part was painstakingly making sure every field, table, connection, target table, everything was perfect. Numerous errors can occur and stop your Transaction Run if something (no matter how small) doesn't sync up. The last step of creating a view took me way longer than I should admit because I kept encountering a "duplicate column error". After searching far and wide, I finally realized the duplication was occurring in my Sales\_Fact\_Table. This single issue sums up what's difficult about the ETL and Database Process. Finally seeing all green checkmarks, data displaying correctly, and proper KPI visualizations flushes someone with a sense of relief like none other.