

Application DataMart

By

Group 14

Hong Zhang

Abdul Bari Mohammed

Shravan Shankar Polisetty Venkataraao

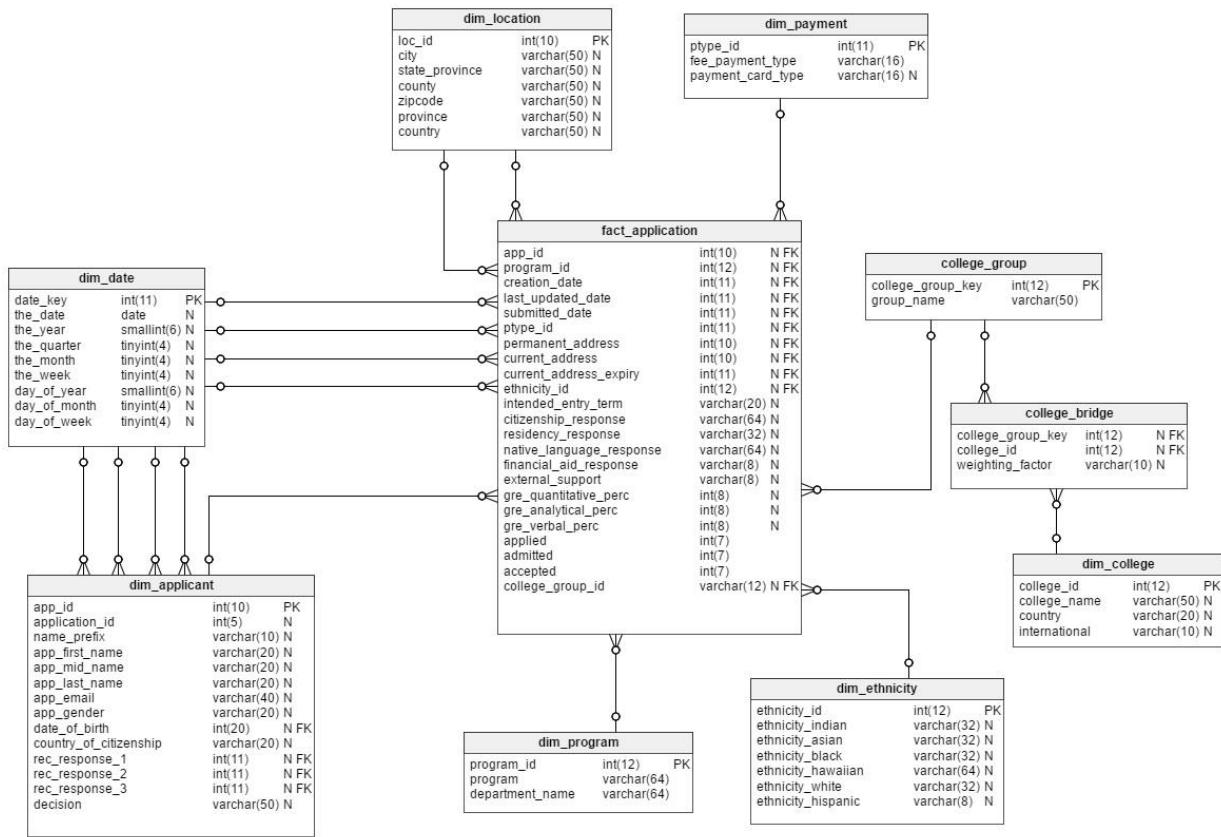
ITMD 526 – Data Warehousing

Illinois Institute of Technology

April 29, 2016

Daniel Lee

Entity Relationship Diagram for the Application datamart



Running all the DDL scripts available to create the structure of the DataMart and Source Table:

Filename: *ddl-scripts.sql*

```

4307
<
1 Messages 2 Table Data 3 Info
39 queries executed, 39 success, 0 errors, 17 warnings

Query: CREATE SCHEMA datamart_application

1 row(s) affected

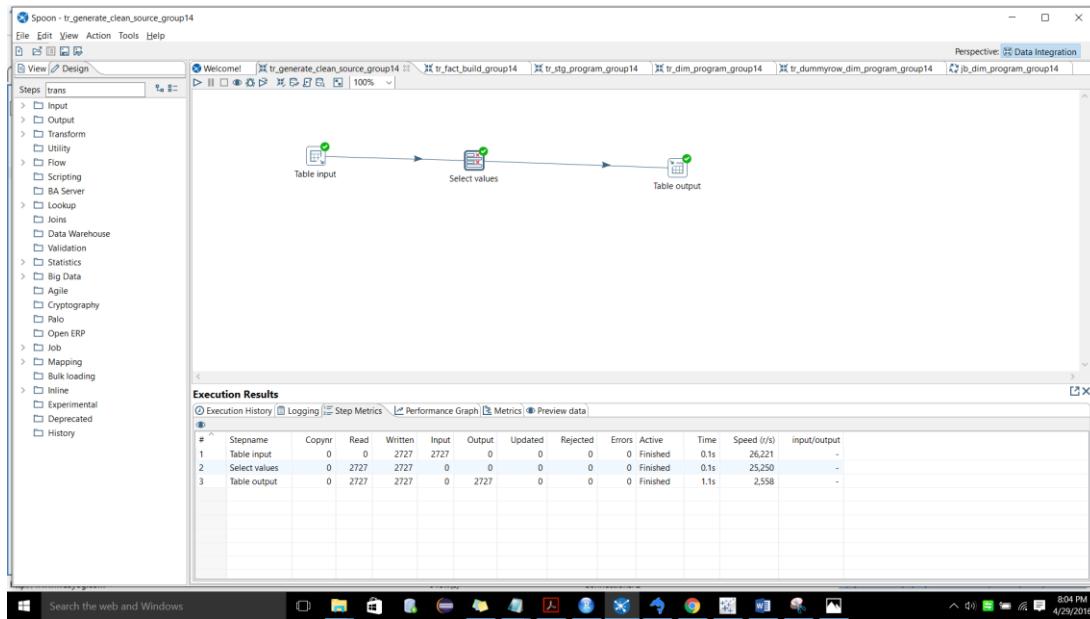
Execution Time : 0.018 sec
Transfer Time : 0 sec

```

Perform Steps in the specified Order for Valid Results.

Creating a clean version of admissions_info table and putting in admissions_info_clean table

Filename: tr_generate_clean_source_group14.ktr



Admissions_info_clean table

application_id	creation_date	last_updated_date	submitted_date	fee_payment_type	payment_card_type	name_prefix	applicant_last_name
1	2013-01-08	2015-01-18	2015-01-15	Credit Card	V	Mister	Lastname_3
2	2012-11-16	2014-11-29	2014-11-22	Credit Card	D	Mister	Lastname_6
3	2013-01-16	2014-10-16	2014-09-05	Credit Card	V	Mister	Lastname_7
4	2013-04-04	2015-02-27	2015-01-15	Credit Card	D	Mister	Lastname_8
5	2013-06-23	2014-11-02	2014-12-01	Credit Card	V	Miss	Lastname_9
6	2013-09-23	2014-12-01	2014-11-30	Credit Card	V	Mister	Lastname_14
7	2013-10-29	2015-01-12	2015-01-10	Credit Card	M	Mister	Lastname_16
8	2013-11-24	2014-11-24	2014-11-22	Credit Card	V	Miss	Lastname_24
9	2013-12-21	2015-01-16	2015-01-15	Check	NA	Mister	Lastname_25
10	2013-12-20	2015-01-16	2015-01-15	Credit Card	V	Mister	Lastname_27
11	2013-12-12	2015-02-11	2015-01-17	Check	NA	Mister	Lastname_28
12	2013-12-31	2014-12-03	2014-11-26	Credit Card	V	Miss	Lastname_32
13	2014-01-13	2014-01-28	2014-01-26	Check	NA	Mister	Lastname_33
14	2014-01-30	2014-11-22	2014-10-23	Credit Card	M	Mrs	Lastname_35
15	2014-02-13	2015-01-05	2014-12-29	Credit Card	V	Miss	Lastname_38
16	2014-02-21	2015-04-01	2014-09-12	Check	NA	Mister	Lastname_39
17	2014-02-21	2014-11-07	2014-11-03	Check	NA	Mister	Lastname_40
18	2014-02-26	2015-01-08	2015-01-07	Credit Card	M	Miss	Lastname_41
19	2014-03-04	2014-12-04	2014-11-30	Check	NA	Mister	Lastname_42
20	2014-04-02	2014-12-03	2014-12-02	Credit Card	V	Mrs	Lastname_45
21	2014-04-14	2014-11-14	2014-10-15	Credit Card	V	Mister	Lastname_49
22	2014-05-01	2015-03-15	2015-03-14	Credit Card	V	Miss	Lastname_50
23	2014-05-15	2014-11-24	2014-11-17	Credit Card	V	Mister	Lastname_52

Creating Date dimension

Changing the global variable values

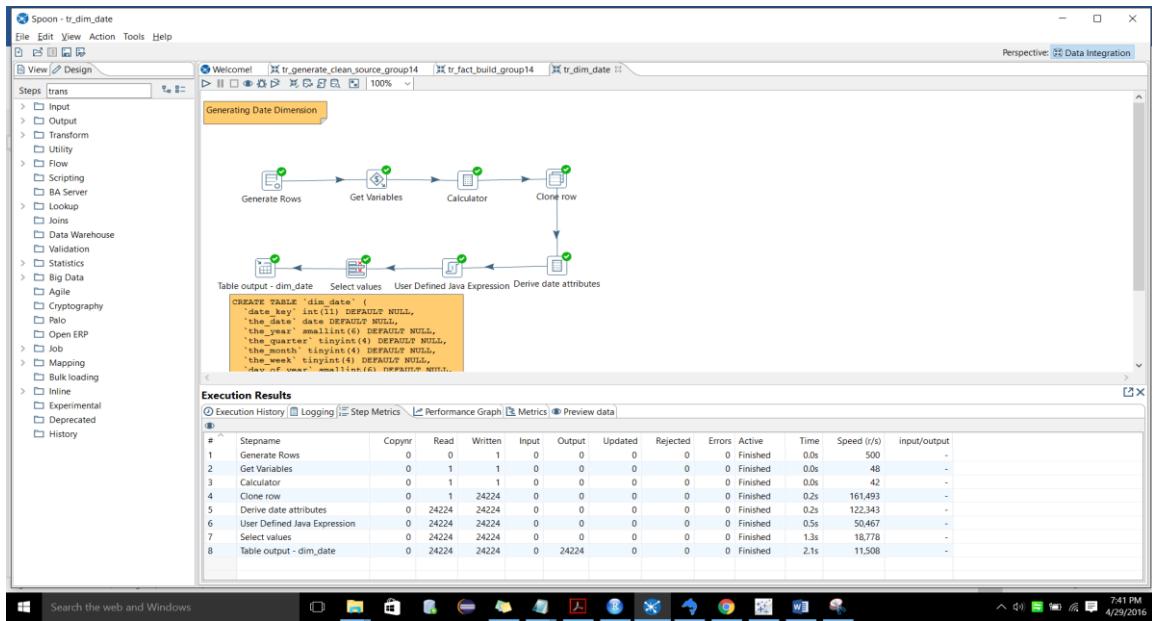
DIM_DATE_START = 1950-01-01

DIM_DATE_END = 2016-04-27

Kettle properties	
Enter the values for the kettle.properties file	
# ^	Variable name
1	DIM_CUSTOMER_TBL
2	DIM_DATE_END
3	DIM_DATE_START
4	DIM_PRODUCT_TBL
5	DIR_SOURCE_FILES
6	ETL_USER_NAME
7	ETL_USER_PASS
8	KETTLE_AGGREGATION_ALL_NULLS_ARE_ZERO
9	KETTLE_AGGREGATION_MIN_NULL_IS_VALUED
10	KETTLE_BATCHING_ROWSET
11	KETTLE_CARTRIDGE_ACCEPTORS

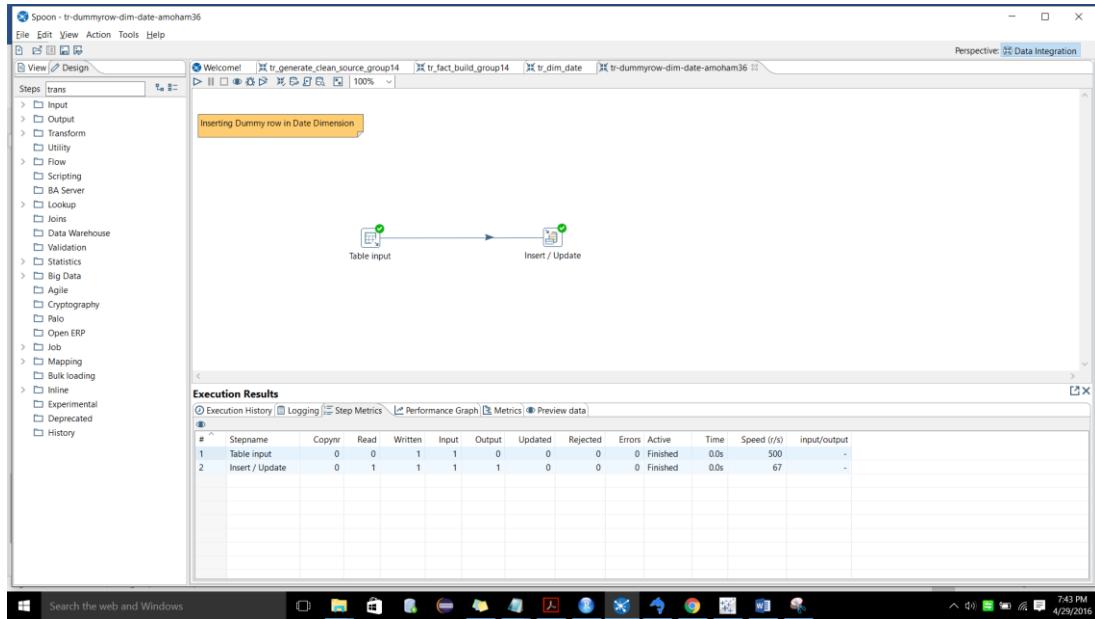
Running the transformation to populate date dimension

Filename: tr_dim_date-group14.ktr

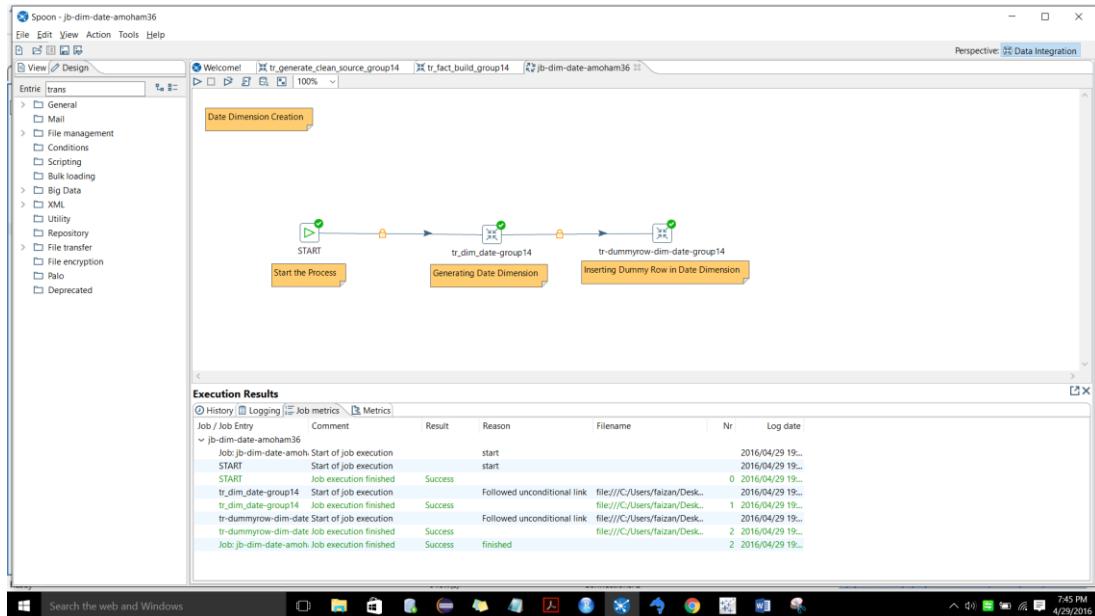


Inserting the dummy row in date dimension

Filename: tr-dummyrow-dim-date-group14.ktr

*Job for populating date dimension with dummy row in one go*

Filename: jb-dim-date-group14.kjb



Dim_date

SQLyog Community 32 - [itm526/datamart_application - root@localhost]

File Edit Favorites Database Table Others Tools Powertools Window Help

itm526 X itm526

Filter tables in datamart_application Filter Ctrl+Shift+B

Query Final.sql Query Query Query Final_dimensions.edl final_staging.sql test.sql ddi-scripts.sql +

Tables

datamart_application

Tables

admissions_info

admissions_info_clean

college_branch

college_group

dim_aplicant

dim_college

dim_ethnicity

dim_location

dim_payment

dim_program

fact_aplication

stg_aplication

stg_ethnicity

stg_location

stg_payment

stg_program

Variables

Stored Procs

Triggers

Events

datamart_application_source

dim_target_db

information_schema

mysql

performance_schema

source_db

target_db

date_key the_date the_year the_quarter the_month the_week day_of_year day_of_month day_of_week

19500101	1950-01-01	1950	1	1	1	1	1	1
19500102	1950-01-02	1950	1	1	1	2	2	2
19500103	1950-01-03	1950	1	1	1	3	3	3
19500104	1950-01-04	1950	1	1	1	4	4	4
19500105	1950-01-05	1950	1	1	1	5	5	5
19500106	1950-01-06	1950	1	1	1	6	6	6
19500107	1950-01-07	1950	1	1	1	7	7	7
19500108	1950-01-08	1950	1	1	2	8	8	1
19500109	1950-01-09	1950	1	1	2	9	9	2
19500110	1950-01-10	1950	1	1	2	10	10	3
19500111	1950-01-11	1950	1	1	2	11	11	4
19500112	1950-01-12	1950	1	1	2	12	12	5
19500113	1950-01-13	1950	1	1	2	13	13	6
19500114	1950-01-14	1950	1	1	2	14	14	7
19500115	1950-01-15	1950	1	1	3	15	15	1
19500116	1950-01-16	1950	1	1	3	16	16	2
19500117	1950-01-17	1950	1	1	3	17	17	3
19500118	1950-01-18	1950	1	1	3	18	18	4
19500119	1950-01-19	1950	1	1	3	19	19	5
19500120	1950-01-20	1950	1	1	3	20	20	6
19500121	1950-01-21	1950	1	1	3	21	21	7
19500122	1950-01-22	1950	1	1	4	22	22	1
19500123	1950-01-23	1950	1	1	4	23	23	2
19500124	1950-01-24	1950	1	1	4	24	24	3

Database: datamart_application Table: dim_date

Ready Search the web and Windows 1000 row(s) Connections: 2 Upgrade to SOLyog Professional/Enterprise/Ultimate 8:16 PM 4/29/2016

Creating Program Dimension*Staging for program dimension*

Filename: tr_stg_program_group14.ktr

Spoon - tr_stg_program_group14

File Edit View Action Tools Help

Welcome! tr_generate_clean_source_group14 tr_fact_build_group14 tr_stg_program_group14 tr_dim_program_group14 tr_dummyrow_dim_program_group14 jb.dim.program_group14

Perspective: Data Integration

Steps trans

- > Input
- > Output
- > Transform
- > Utility
- > Flow
- > Scripting
- > BA Server
- > Lookup
- > Joins
- > Data Warehouse
- > Validation
- > Statistics
- > Big Data
- > Agile
- > Cryptography
- > Palo
- > Open ERP
- > Job
- > Mapping
- > Bulk loading
- > Inline
- > Experimental
- > Deprecated
- History

Table input → Table output

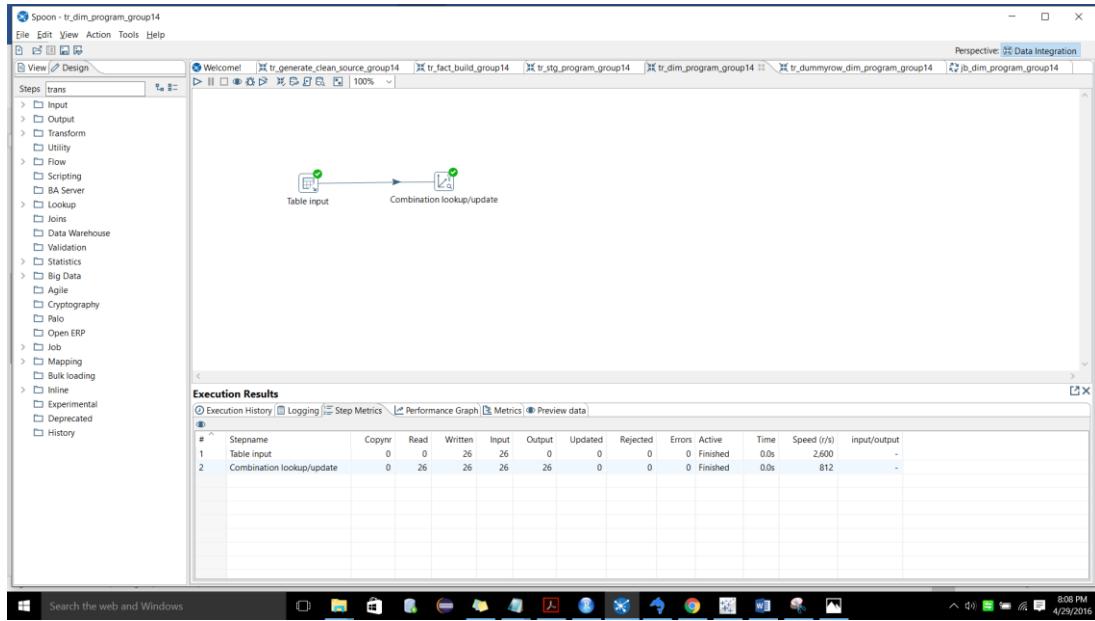
Execution Results

#	Stepname	Copysn	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Table input	0	0	26	26	0	0	0	0	Finished	0.0s	2,000	-
2	Table output	0	26	26	0	26	0	0	0	Finished	0.0s	1,130	-

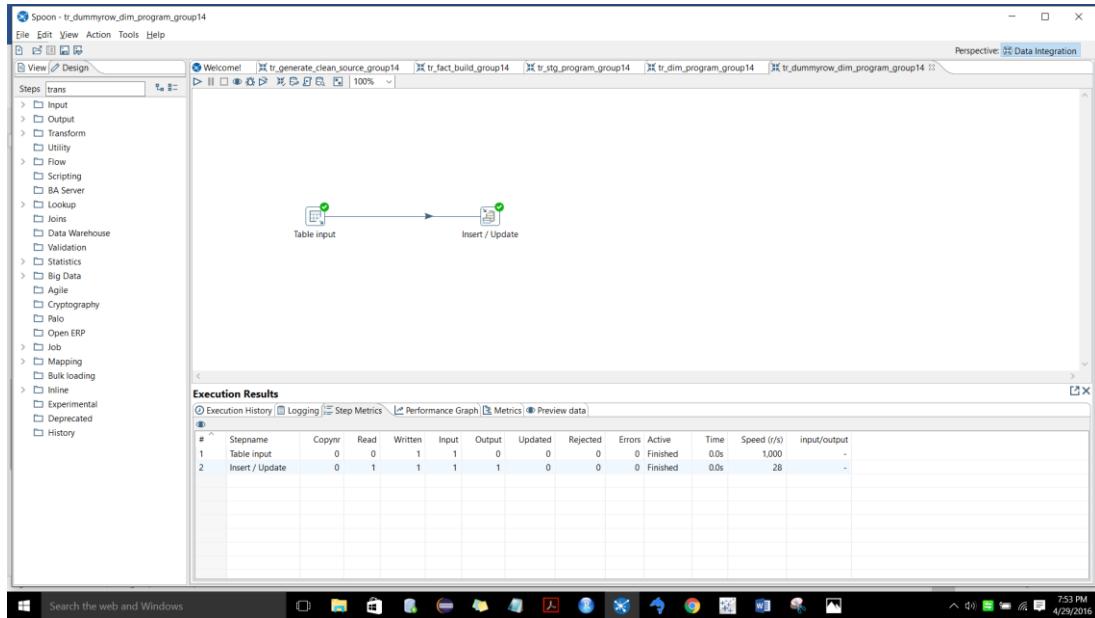
Search the web and Windows 8:07 PM 4/29/2016

Populating the program Dimension

Filename: tr_dim_program_group14.ktr

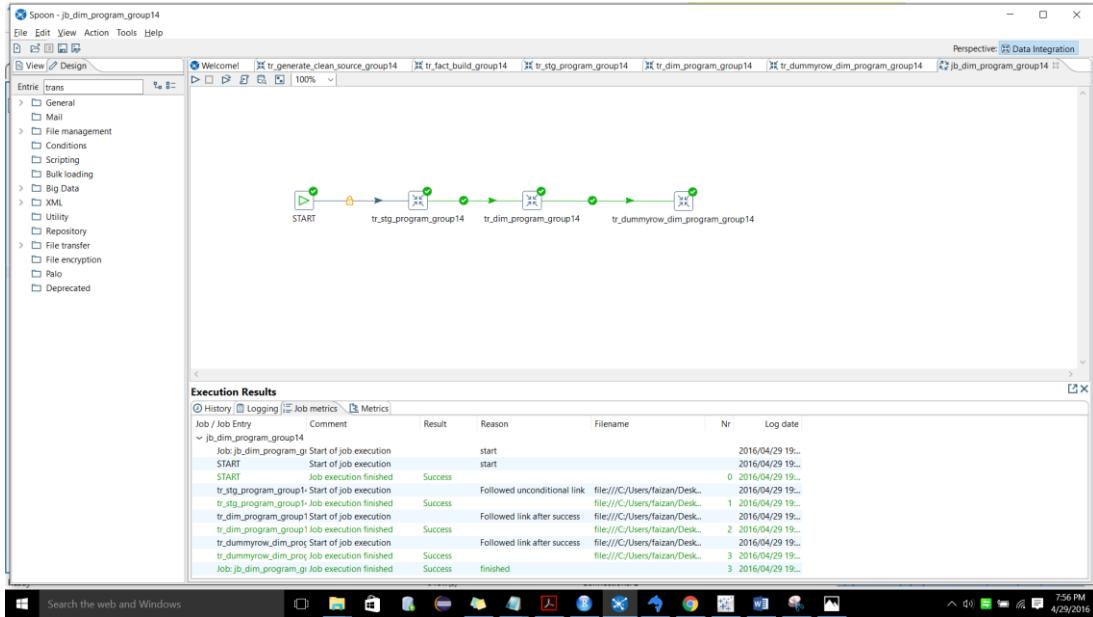
*Inserting dummy row in program dimension*

Filename: tr_dummyrow_dim_program_group14.ktr



Job to populate Program dimension with dummy row in one go

Filename: jb_dim_program_group14.kjb



Dim_program

The screenshot shows the SQLyog interface connected to the 'datamart_application' database. The 'dim_program' table is selected, showing the following data:

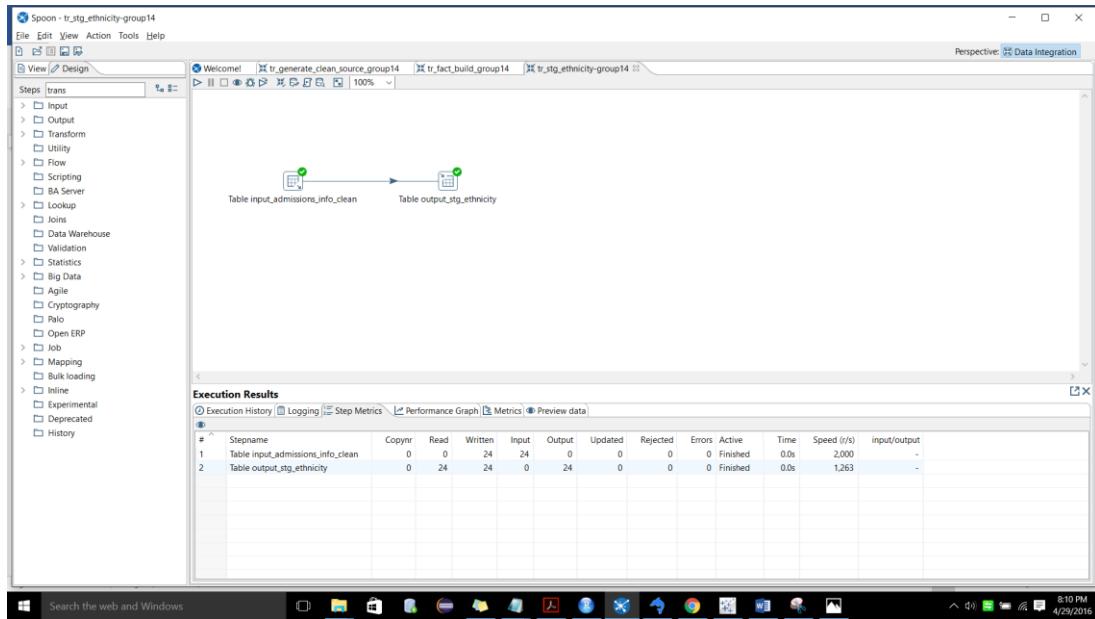
program_id	program	department_name
1	Statistics-PHD	Statistics
2	Psychology-PHD	Psychology
3	Mathematics-PHD	Mathematics
4	Computer Science-MS	Computer Science
5	Philosophy-PHD	Philosophy
6	Physics-PH	Physics and Astronomy
7	Neuroscience-PHD	Neuroscience
8	Physics-MS	Physics and Astronomy
9	Chemistry-BS	Chemistry
10	Computer Science-PHD	Computer Science
11	Law-LLM	Law
12	English-PHD	English
13	Applied Mathematics-MA	Mathematics
14	Statistics-MS	Statistics
15	Applied Statistics-MS	Statistics
16	Economics-PHD	Economics
17	Applied Statistics-MA	Statistics
18	Philosophy-MA	Philosophy
19	English-HFA	English
20	Statistics-MA	Statistics
21	Neuroscience-MS	Neuroscience
22	English-MA	English
23	Applied Mathematics-MS	Mathematics

Database: datamart_application Table: dim_program

Creating Ethnicity dimension which is a junk dimension

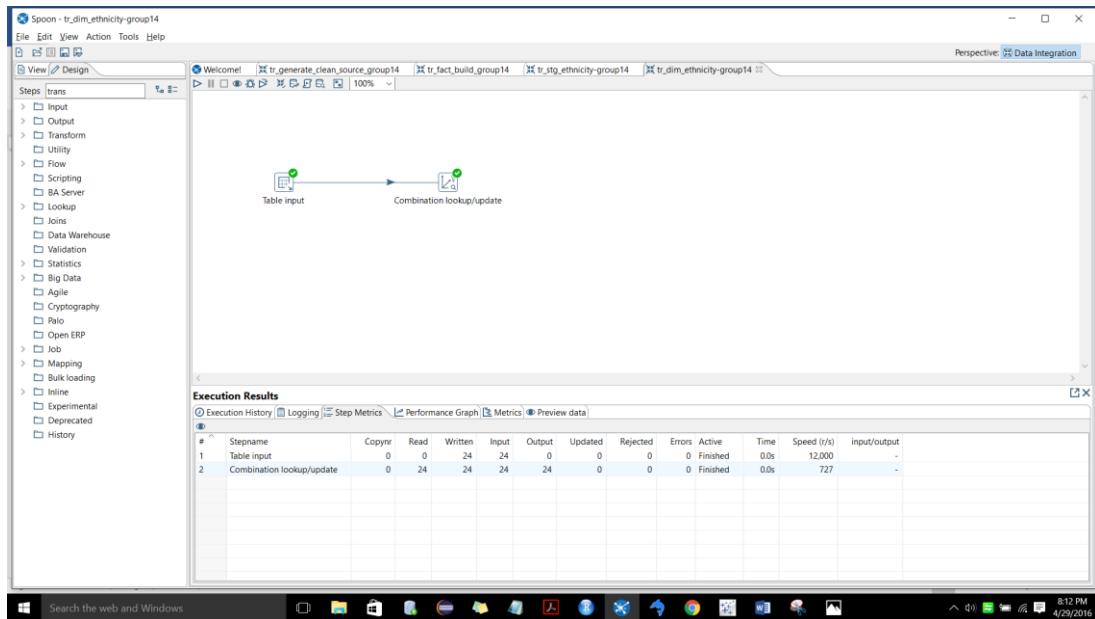
Staging for ethnicity dimension

Filename: tr_stg_ethnicity-group14.ktr



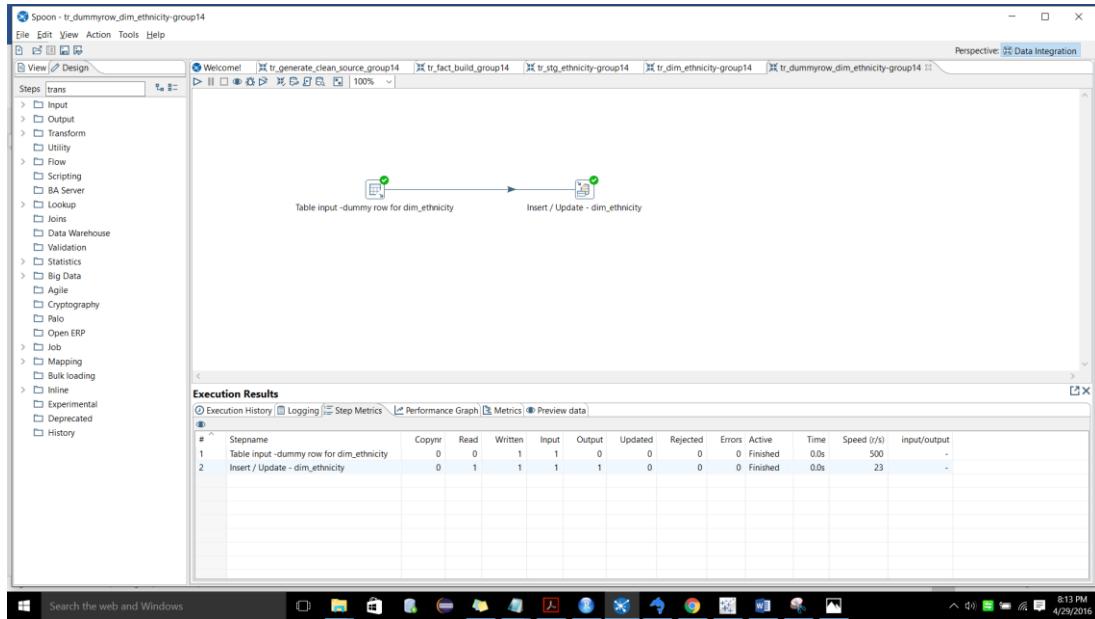
Populating ethnicity dimension

Filename: tr_dim_ethnicity-group14.ktr

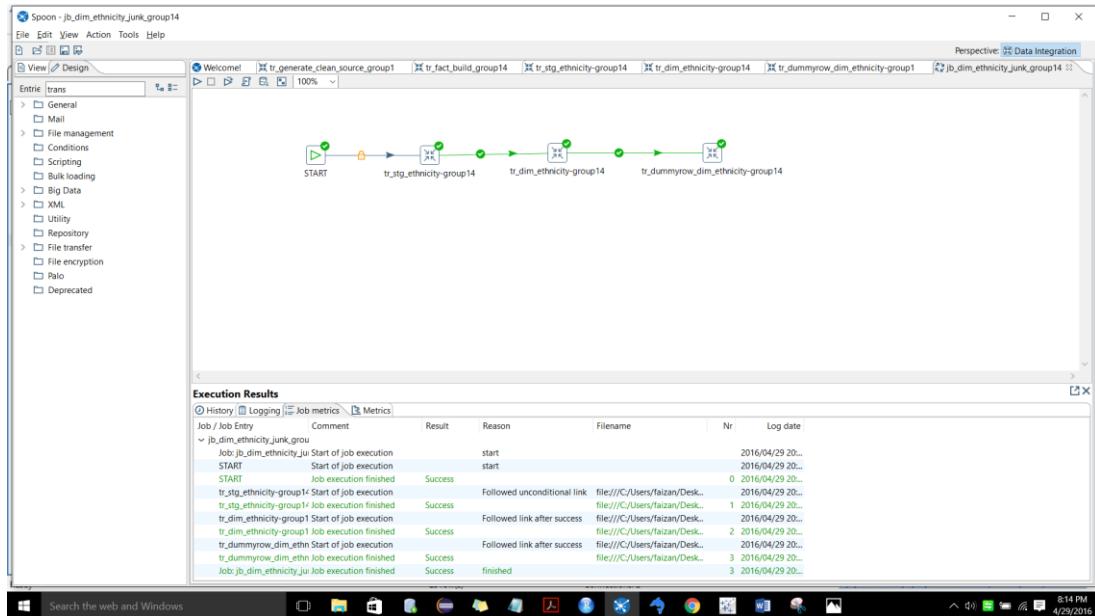


Inserting dummy row in ethnicity dimension

Filename: tr_dummyrow_dim_ethnicity-group14.ktr

*Job to populate ethnicity dimension with dummy row in one go*

Filename: jb_dim_ethnicity_junk_group14.kjb

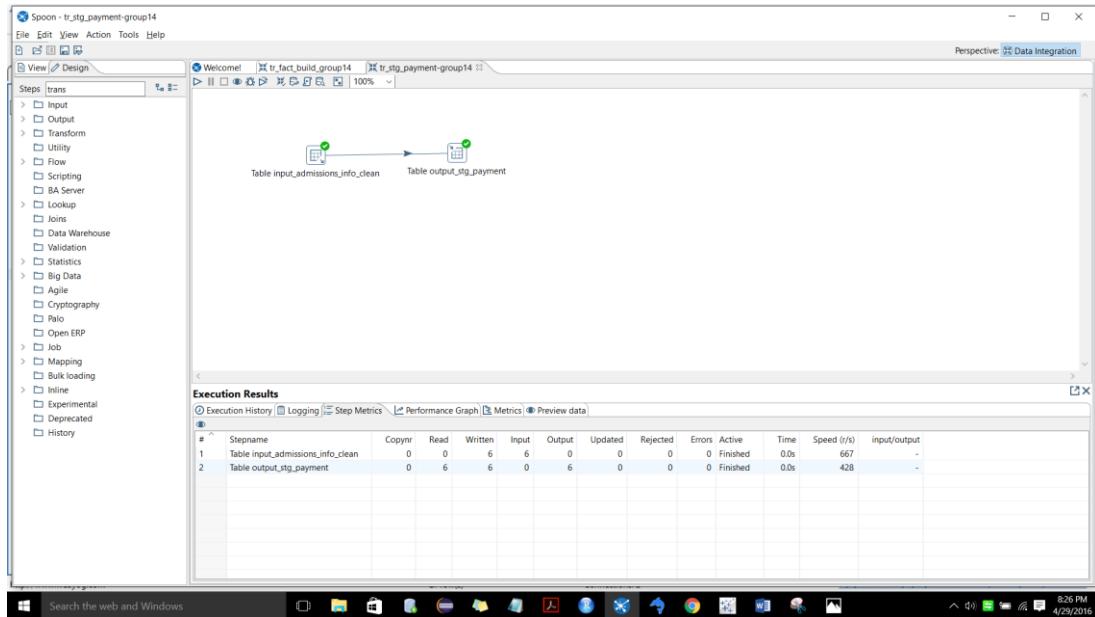


Dim_ethnicity

Creating Payment dimension

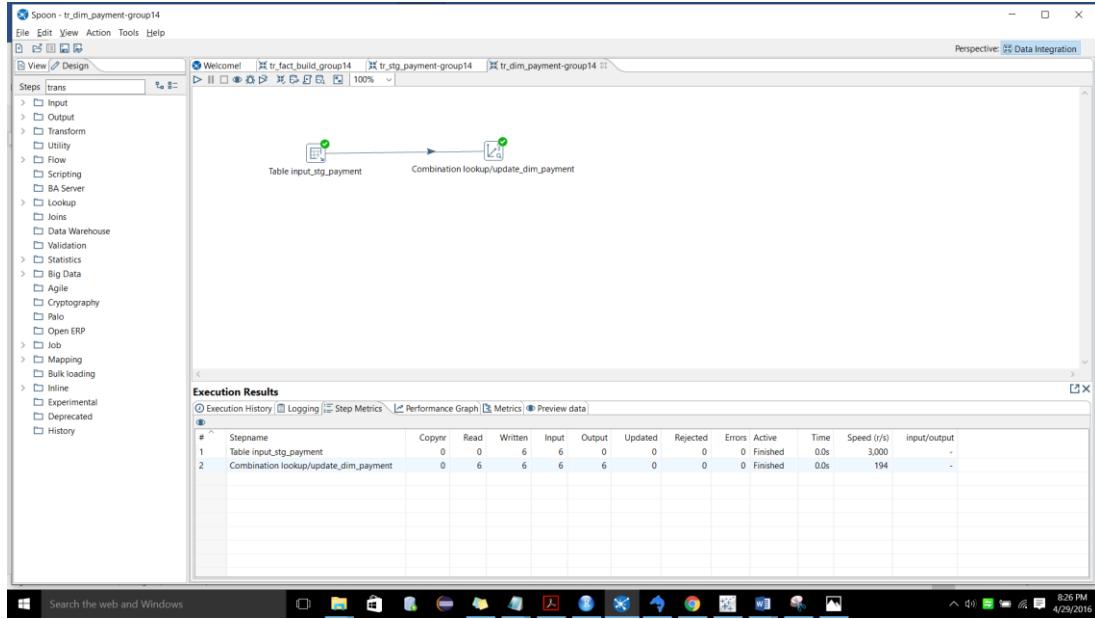
Staging for payment dimension

Filename: tr_stg_payment-group14.ktr

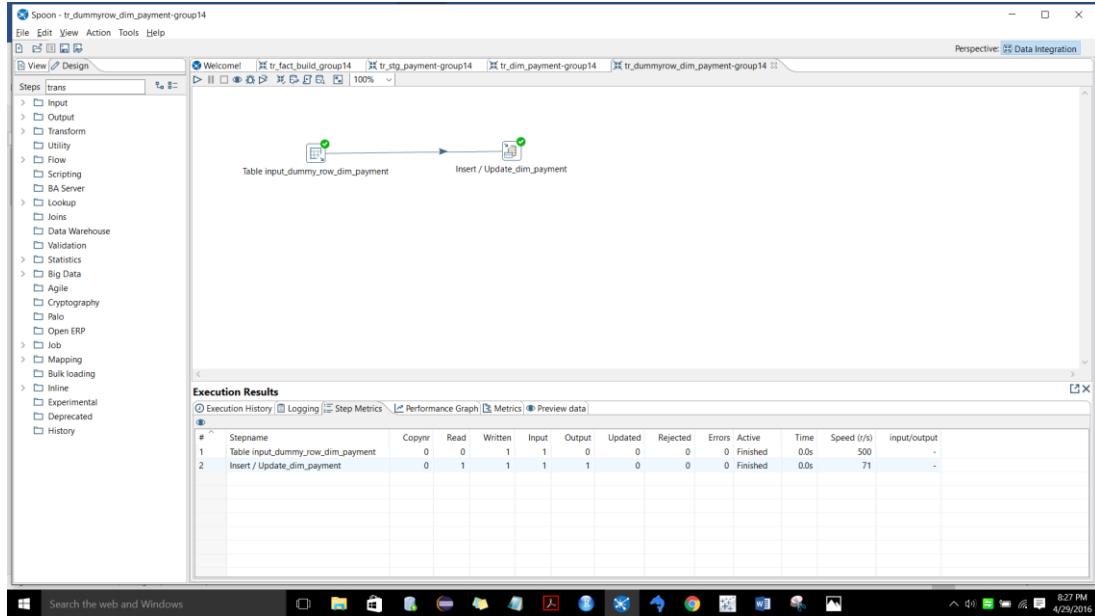


Populating payment dimension

Filename: tr_dim_payment-group14.ktr

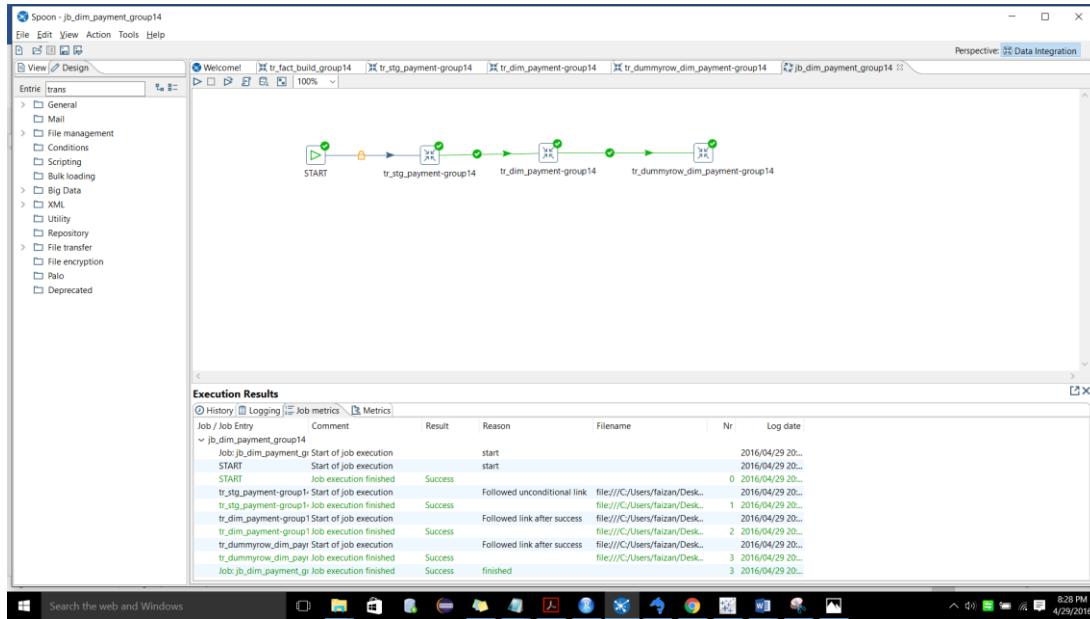
*Inserting dummy row in payment dimension*

Filename: tr_dummyrow_dim_payment-group14.ktr



Job to populate payment dimension with dummy row in one go

Filename: jb_dim_payment_group14.kjb



Dim_payment

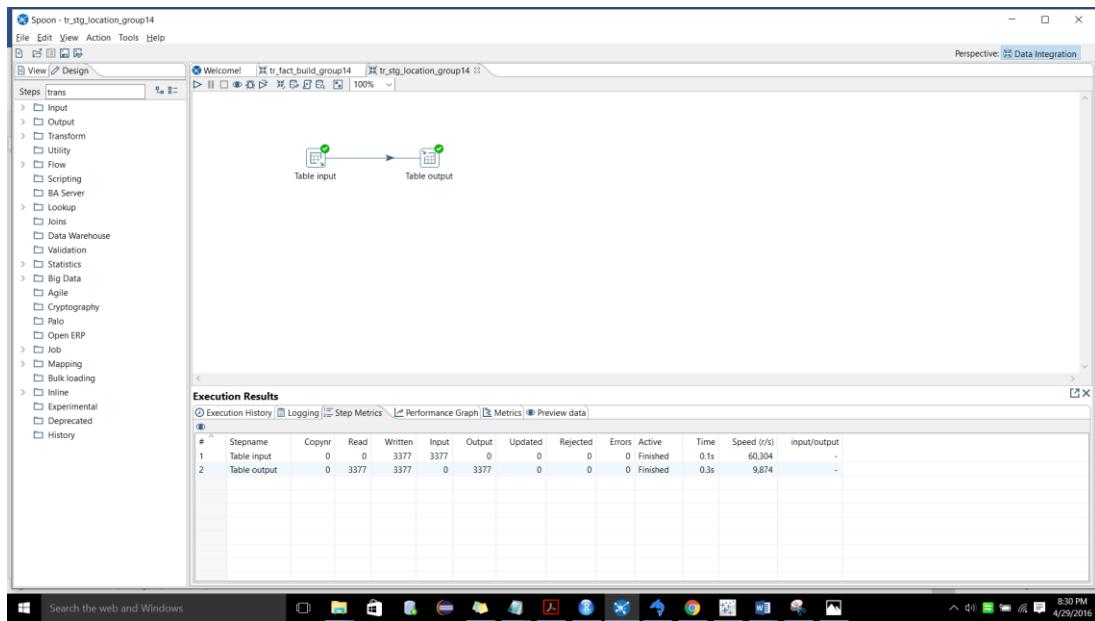
The screenshot shows the SQLyog interface connected to the 'dataset_application' database. The left pane displays the database structure with various tables such as 'dim_payment', 'stg_payment', 'dim_location', etc. The right pane shows the data for the 'dim_payment' table, which has the following structure and data:

dim_payment_id	fee_payment_type	payment_card_type
1	Credit Card	A
2	Credit Card	D
3	Credit Card	M
4	Check	NA
5	Credit Card	C
6	Credit Card	NA
0	Unknown	Unknown

Creating location dimension

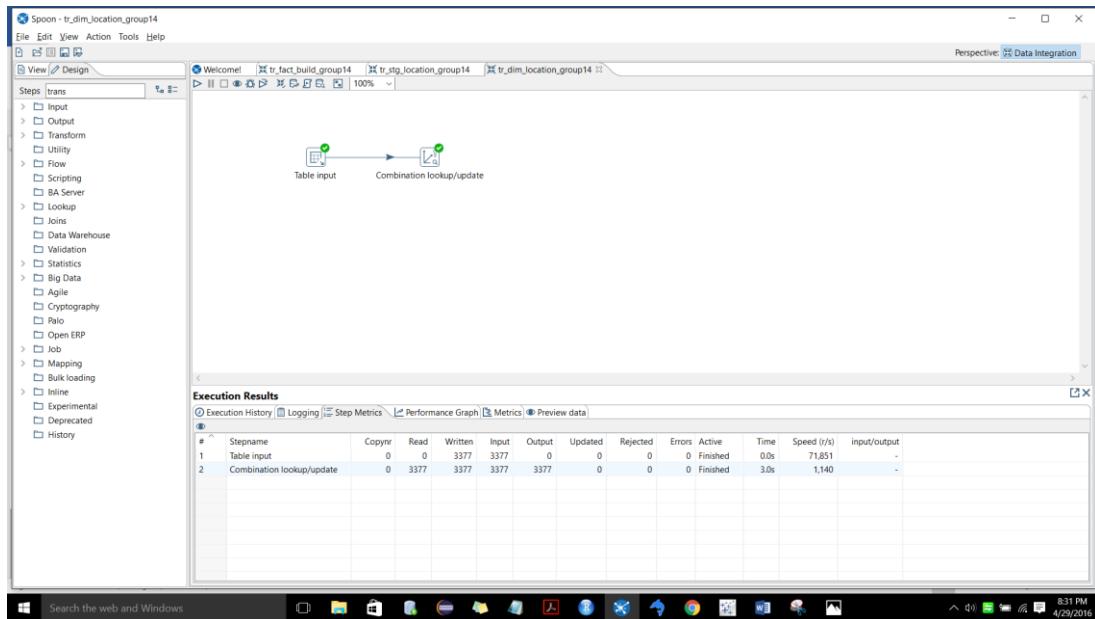
Staging for location dimension

Filename: tr_stg_location_group14.ktr



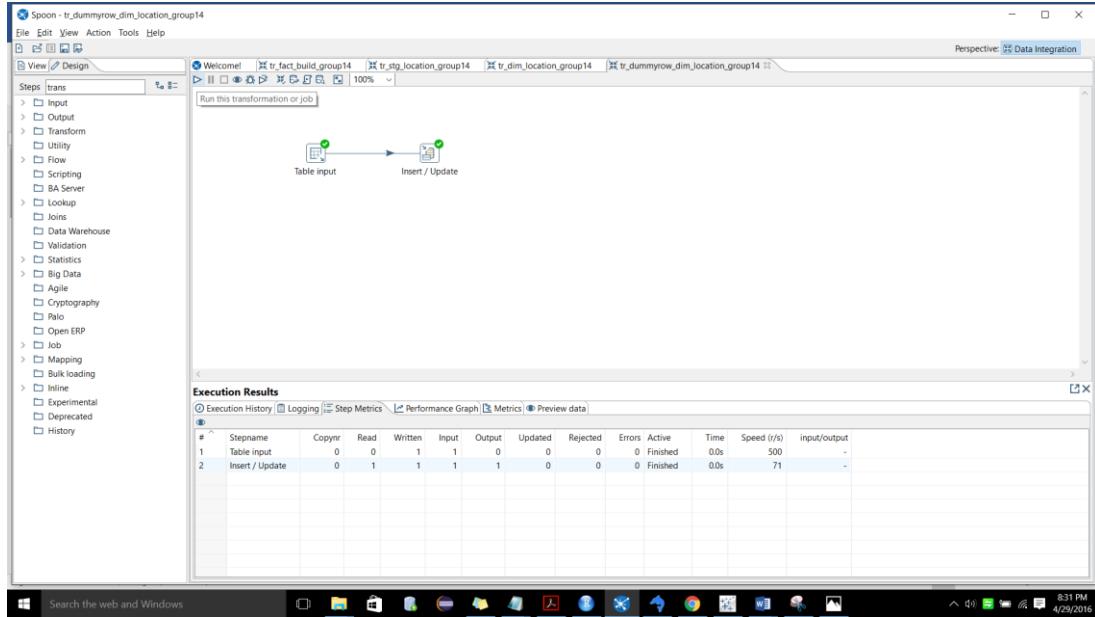
Populating location dimension

Filename: tr_dim_location_group14.ktr

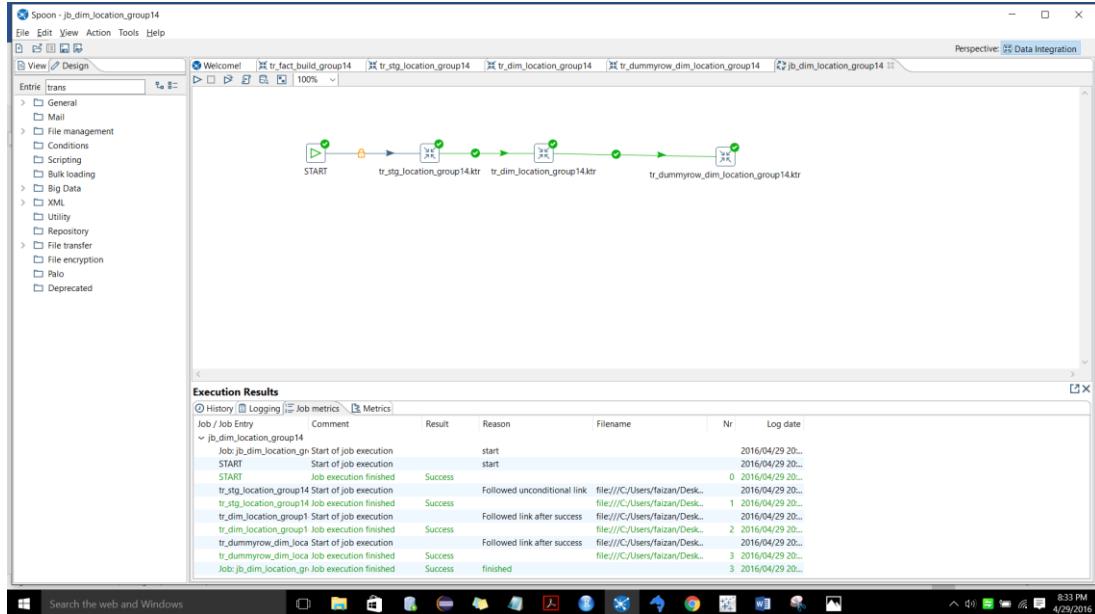


Inserting dummy row in location dimension

Filename: tr_dummyrow_dim_location_group14.ktr

*Job to populate location dimension with dummy row in one go*

Filename: jb_dim_location_group14.kjb



Dim_location

The screenshot shows the SQLyog interface with the 'datamart_application' database selected. The 'dim_location' table is open, displaying 1000 rows of data. The columns are: loc_id, city, state_province, county, zipcode, province, and country. The data includes various US cities like Harrison, Flemington, Tequicuilapa, Union Park, Queen Creek, Bethesda, Frohocke, Austin, Fayetteville, New York City, Hamilton, Latrobe, Muncie, Ligonier, Chemung, Harrison City, Madrid, Galena, Coraopolis, Monroeville, Irvine, Fayetteville, El Paso, and Hyderabad, along with their corresponding states/provinces, counties, zipcodes, provinces, and countries.

loc_id	city	state_province	county	zipcode	province	country
1	Harrison	New Jersey	NA	07029	NA	United States
2	Flemington	New Jersey	NA	08822	NA	United States
3	Tequicuilapa	NA	NA	1111	NA	Mexico
4	Union Park	Pennsylvania	Allegheny	15101	NA	United States
5	Queen Creek	Arizona	NA	85142	NA	United States
6	Bethesda	Maryland	NA	20817	NA	United States
7	Frohocke	NA	NA	MEZ 2HS	NA	Canada
8	Austin	Texas	NA	78705	Travis	United States
9	Fayetteville	Pennsylvania	Franklin	17222	NA	United States
10	New York City	New York	NA	11231	NA	United States
11	Hamilton	New Jersey	NA	08610	NA	United States
12	Latrobe	Pennsylvania	Westmoreland	15650	NA	United States
13	Muncie	Pennsylvania	Allegheny	15120	NA	United States
14	Ligonier	Pennsylvania	Westmoreland	15658	NA	United States
15	Chemung	PA	NA	NA	PA	United States
16	Harrison City	Pennsylvania	Westmoreland	15636	PA	United States
17	Madrid	NA	NA	28221	NA	Spain
18	Galena	Ohio	NA	43021	NA	United States
19	Coraopolis	Pennsylvania	Allegheny	15108	NA	United States
20	Monroeville	Pennsylvania	Allegheny	15146	NA	United States
21	Irvine	California	NA	92612	NA	United States
22	Fayetteville	Arkansas	NA	72701	Arkansas	United States
23	El Paso	Texas	NA	79902	Texas	United States
24	Hyderabad	NA	NA	500089	NA	India

Creating Applicant dimension*Staging for applicant dimension*

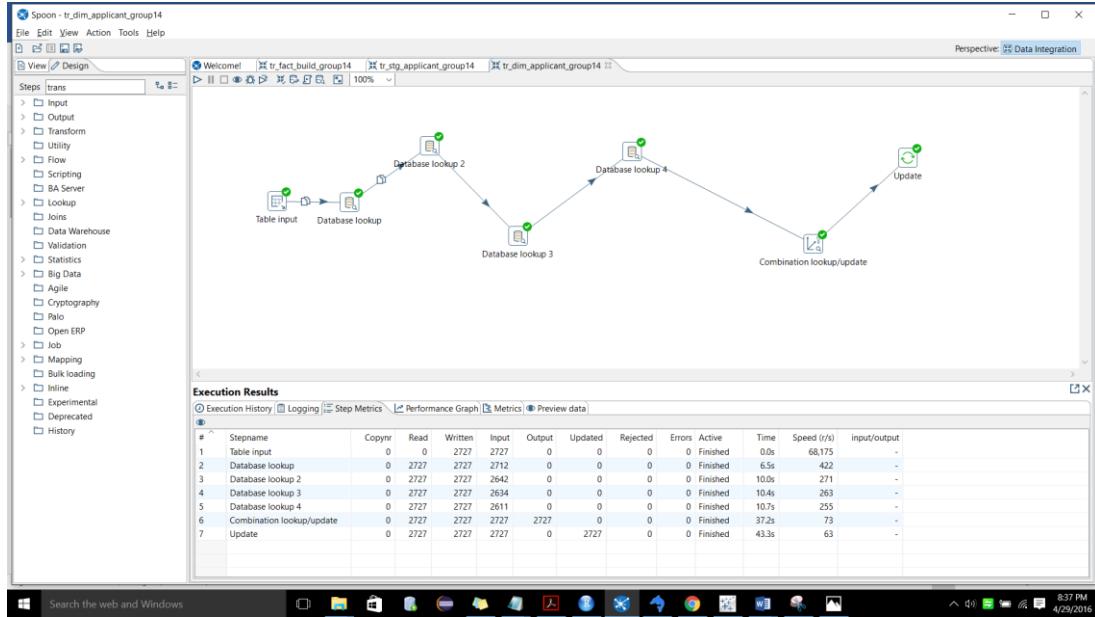
Filename: tr_stg_applicant_group14.ktr

The screenshot shows the Spoon interface for a Data Integration job named 'tr_stg_applicant_group14'. The job consists of a single step: 'Table input' followed by 'Table output'. The execution results show one row being processed from the 'Table input' step and one row being output from the 'Table output' step. The execution time was 0.3s and the speed was 101,000 input/output.

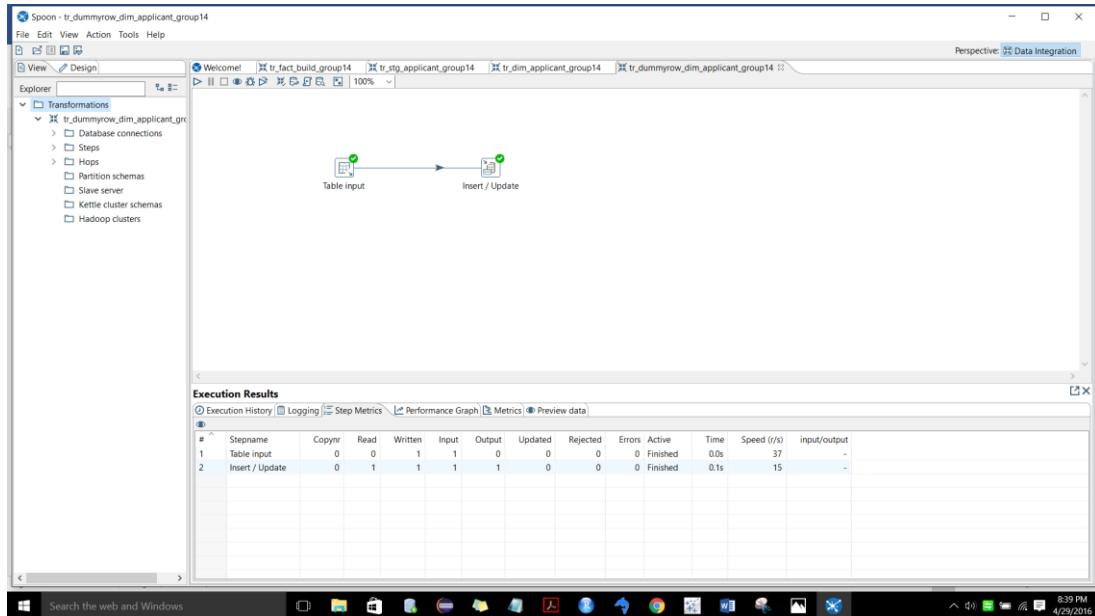
#	Stepname	Copysn	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Table input	0	0	2727	2727	0	0	0	0	0	0.0s	101,000	-
2	Table output	0	2727	2727	0	2727	0	0	0	0	0.3s	8,140	-

Populating applicant dimension

Filename: tr_dim_applicant_group14.ktr

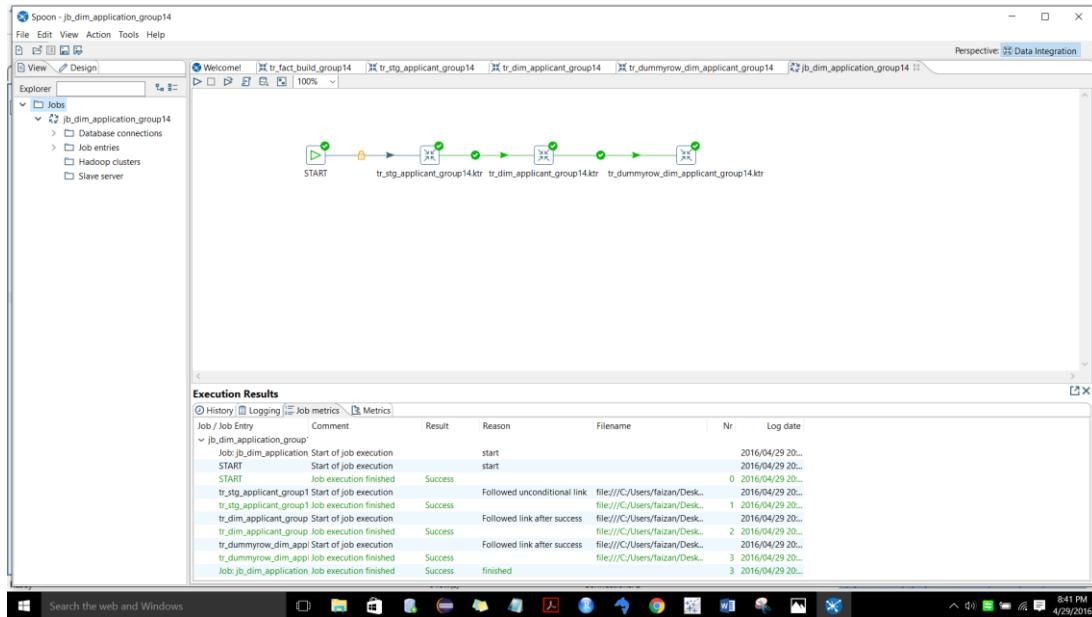
*Inserting dummy row in applicant dimension*

Filename: tr_dummyrow_dim_applicant_group14.ktr



Job to populate applicant dimension with dummy row in one go

Filename: jb_dim_application_group14.kjb

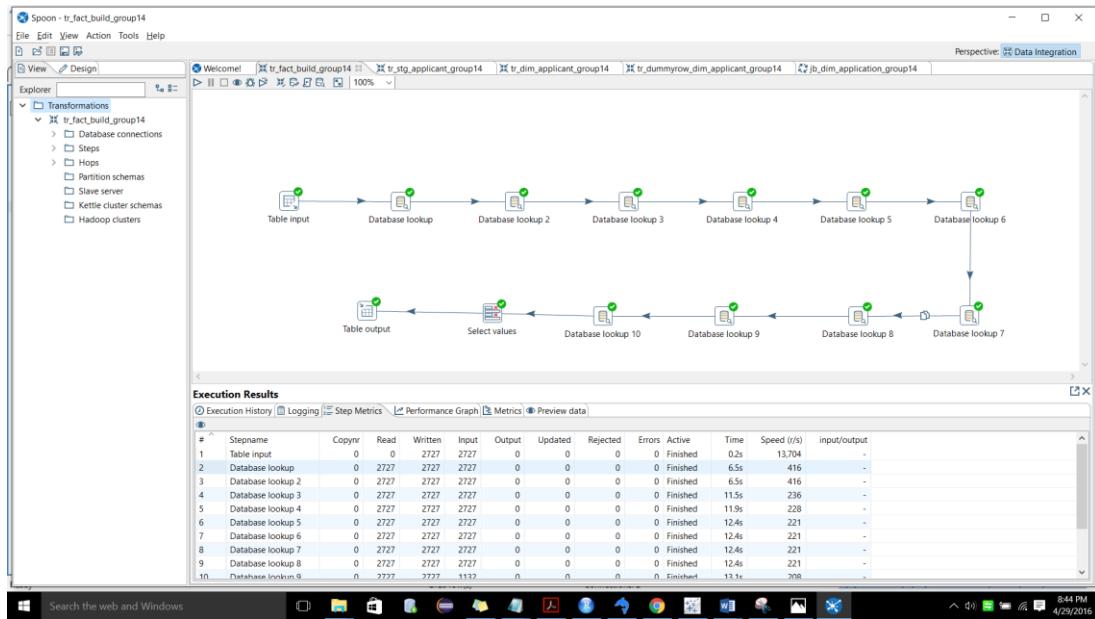


Dim_applicant

Creating Application Fact

Populating application fact

Filename: tr_fact_build_group14.ktr

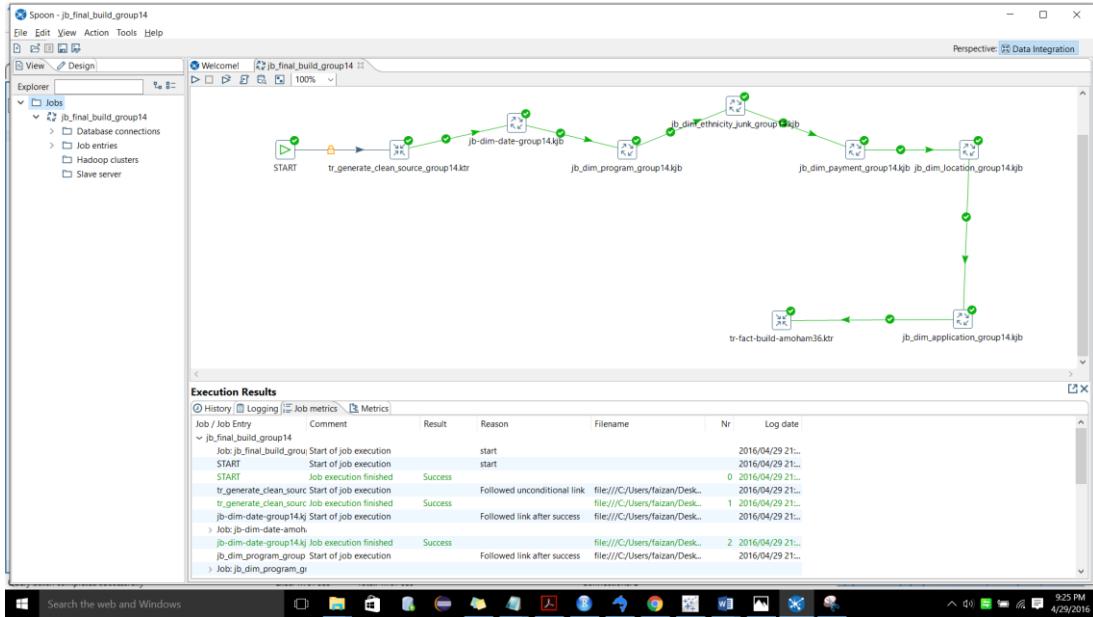


Application_fact

program_id	creation_date	last_updated_date	submitted_date	ptype_id	permanent_address	current_address	current_address_expiry	ethnicity_id	intended_entry
1	20130108	20130119	20130115	1	1	1	20150530	1	1 Fall 2015
2	20121116	20141129	20141122	2	2	2	0	0	2 Fall 2015
3	20130108	20141016	20140905	1	3	3	0	0	3 Fall 2015
4	20130404	20150227	20150115	2	4	4	0	0	2 Fall 2015
5	20130623	20141202	20141201	1	5	5	20141231	4	4 Fall 2015
6	20130923	20141201	20141130	1	6	6	0	0	2 Fall 2015
7	20131029	20150112	20150110	3	2104	7	0	0	1 Fall 2015
8	20131121	20141122	20141122	1	2105	8	20150531	1	1 Fall 2015
9	20131127	20150114	20150113	4	9	9	20140401	2	2 Fall 2015
10	20140105	20141226	20141226	1	1546	10	20140715	2	2 Fall 2015
11	20131212	20150211	20150117	4	2106	11	0	0	2 Fall 2015
12	20131221	20141203	20141206	1	6	6	0	0	2 Fall 2015
13	20140113	20140128	20140126	6	2107	12	20140515	2	2 Spring 2015
14	20140130	20141122	20141123	5	13	13	0	0	5 Fall 2015
15	20140213	20150105	20141229	1	14	14	0	0	2 Fall 2015
16	20140221	20150401	20140912	4	2108	15	0	0	1 Spring 2015
17	20140221	20141107	20141103	4	16	16	0	0	2 Fall 2015
18	20140226	20150108	20150107	3	2109	17	20150515	2	2 Fall 2015
19	20140304	20141204	20141130	4	18	18	0	0	2 Summer 2015
20	20140402	20141203	20141202	1	19	19	0	0	2 Fall 2015
21	20140428	20150124	20150123	1	20	20	0	0	2 Spring 2015
22	20140501	20150115	20150114	1	1411	21	20150430	1	1 Fall 2015
23	20140515	20141124	20141117	1	22	22	0	0	2 Fall 2015

Job to Complete the whole Process in one go

Filename: jb_final_build_group14.kjb



ANALYSIS REPORTS

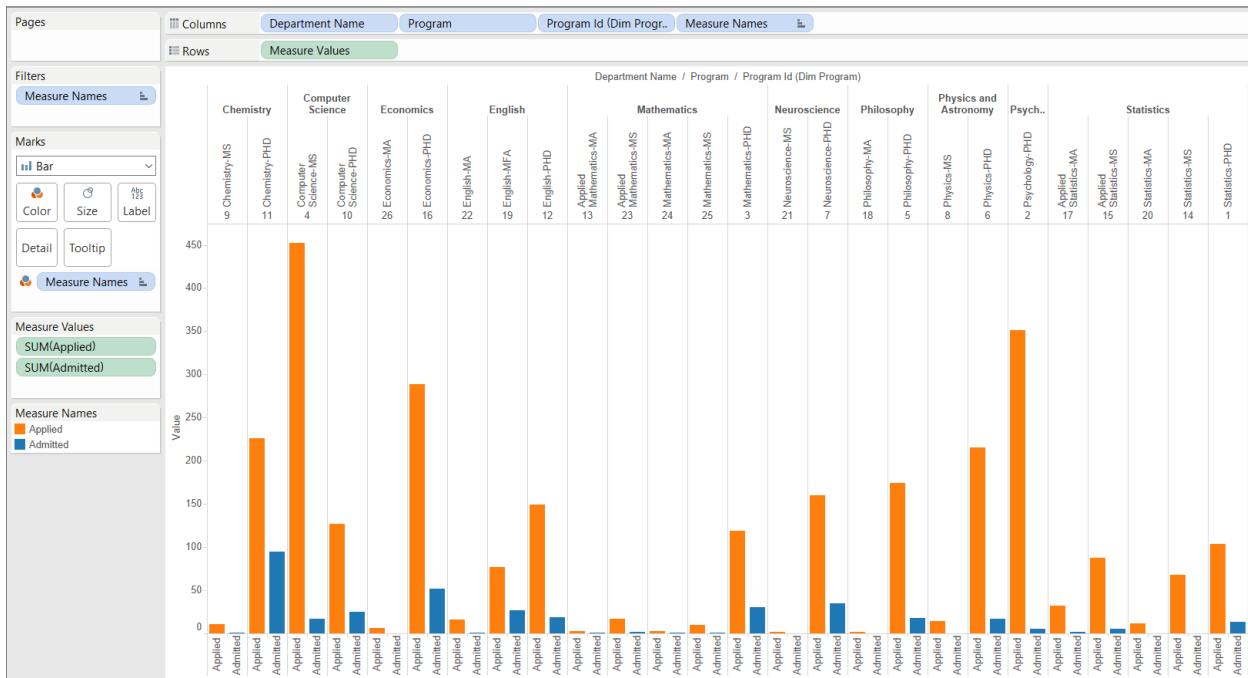
In the submitted Tableau worksheet, you have to give your localhost's Username and Password, after executing the DDLs and performing all the ETL jobs, to see the reports.

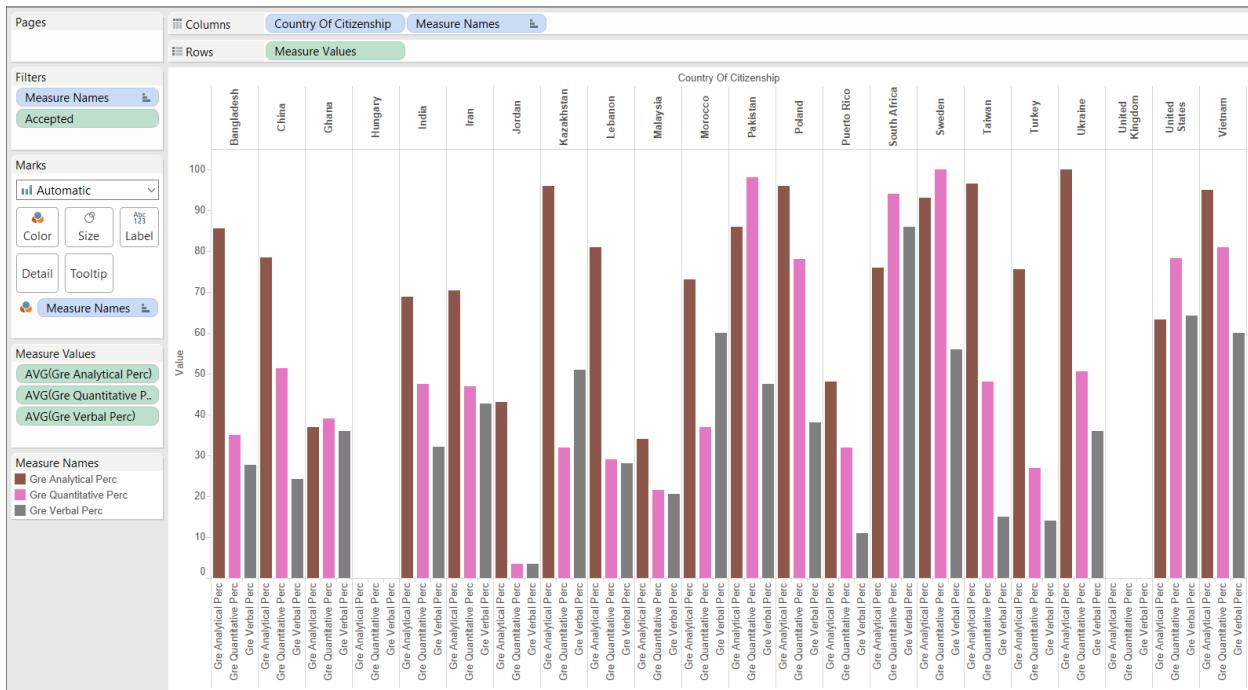
Selectivity & Yield Ration per Country

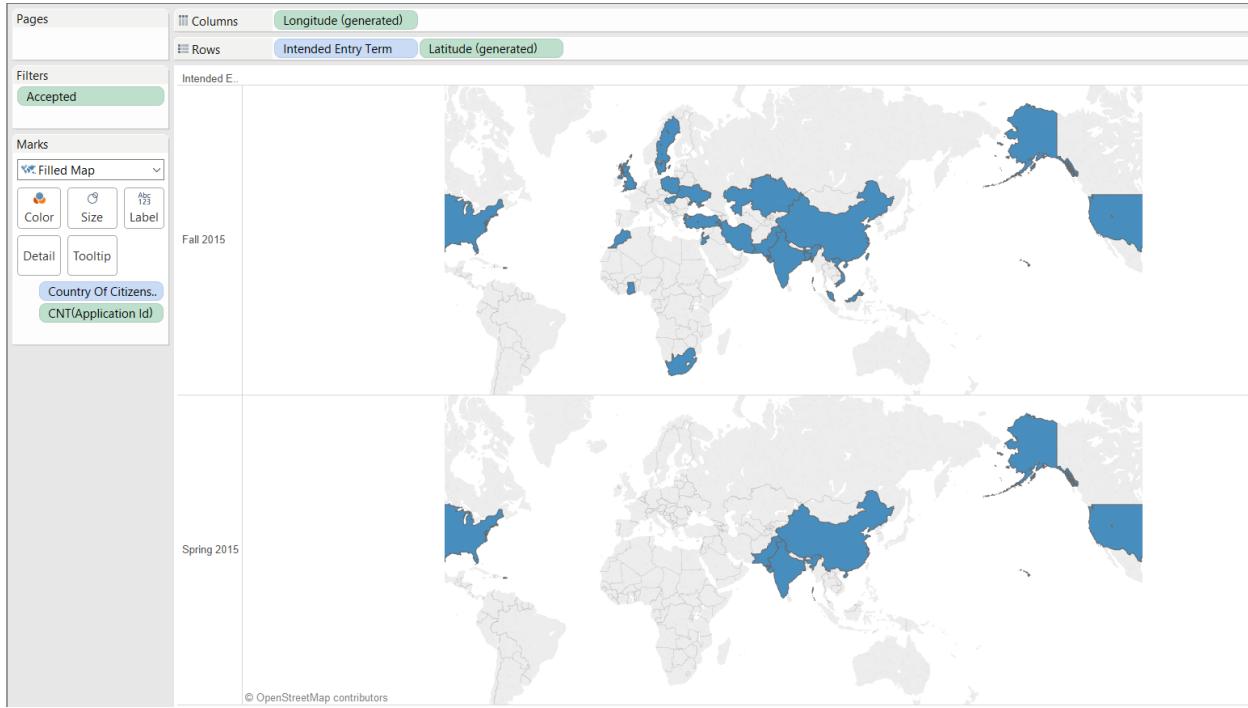
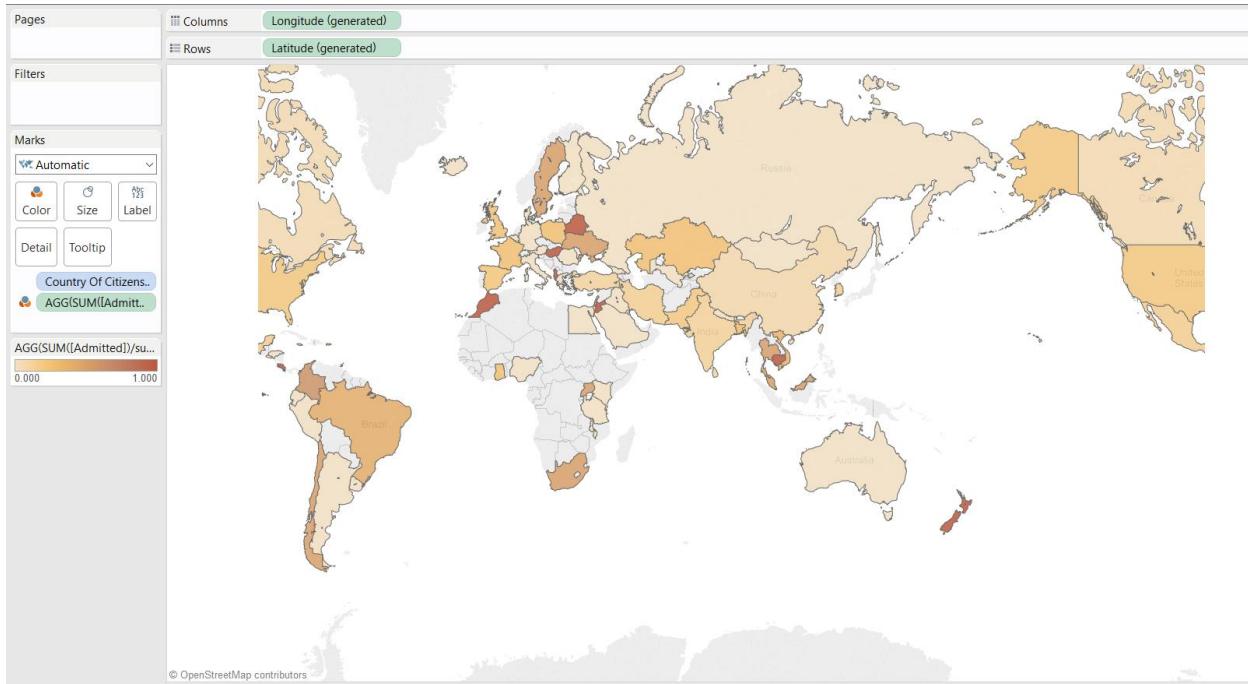
The screenshot shows a Tableau interface with the following details:

- Filters:** Measure Names (selected)
- Marks:** Automatic, with options for Color, Size, and Text.
- Measure Values:** SUM(Applied), SUM(Admitted), SUM(Accepted), AGG(sum([Admitted])/S..), and AGG(SUM([Accepted])/..).
- Table Headers:** Country Of Citizenship, Applied, Admitted, Accepted, Selectivity, Yield Ratio.
- Data Rows:** The table lists countries with their respective counts and metrics. Notable values include China (988 applied, 67 admitted, 28 accepted), India (271 applied, 32 admitted, 18 accepted), and Kenya (2 applied, 0 admitted, 0 accepted). The Selectivity column shows percentages like 100.00% for Argentina and 0.00% for Australia.

Country Of Citizenship	Applied	Admitted	Accepted	Selectivity	Yield Ratio
Albania	1	1	0	100.00%	0.00
Argentina	1	0	0	0.00%	
Australia	3	0	0	0.00%	
Austria	1	0	0	0.00%	
Bangladesh	11	3	3	27.27%	1.00
Belarus	1	1	0	100.00%	0.00
Belgium	1	0	0	0.00%	
Brazil	5	2	0	40.00%	0.00
Cambodia	1	1	0	100.00%	0.00
Canada	20	1	0	5.00%	0.00
Chile	2	1	0	50.00%	0.00
China	988	67	28	6.78%	0.42
Colombia	5	3	0	60.00%	0.00
Costa Rica	1	1	0	100.00%	0.00
Cyprus	1	0	0	0.00%	
Denmark	1	0	0	0.00%	
Ecuador	1	0	0	0.00%	
Egypt	1	0	0	0.00%	
El Salvador	1	0	0	0.00%	
Finland	1	0	0	0.00%	
France	4	1	0	25.00%	0.00
Georgia	1	0	0	0.00%	
Germany	9	0	0	0.00%	
Ghana	4	1	1	25.00%	1.00
Greece	4	0	0	0.00%	
Honduras	1	0	0	0.00%	
Hong Kong	9	0	0	0.00%	
Hungary	2	2	1	100.00%	0.50
Iceland	1	0	0	0.00%	
India	271	32	18	11.81%	0.56
Iran	29	4	3	13.79%	0.75
Iraq	2	0	0	0.00%	
Israel	6	1	0	16.67%	0.00
Italy	4	0	0	0.00%	
Jamaica	2	0	0	0.00%	
Jordan	2	2	2	100.00%	1.00
Kazakhstan	4	1	1	25.00%	1.00
Kenya	2	0	0	0.00%	
Korea, South	22	11	6	27.27%	0.27

Applicants per Country*Applied VS Admitted per Program*

Applicants per Country per Term*Average GRE Scores of candidates per Country who accepted the admission*

Applicants who Accepted admission per term*Selectivity per Country*

Yield Ratio per Country