Step 13: Group Project Requirement

Your goal in this project is to continue the data migration process. Design and create 5 Pentaho transformation by using the Sakila database. Your transformation steps are not limited to the steps I introduced above. You are encouraged to explore much more possibilities.

For each of the transformations

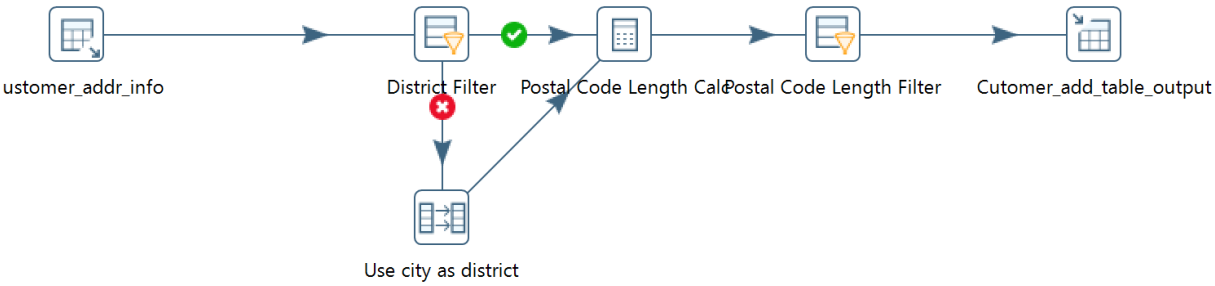
Use a Microsoft Word file to

1. Describe the business value why the output table is important.

2. Clearly state your assumptions and rules used in the transformation process.

Use a screenshot software to

3. Provide a screenshot of the transformation steps. For example,

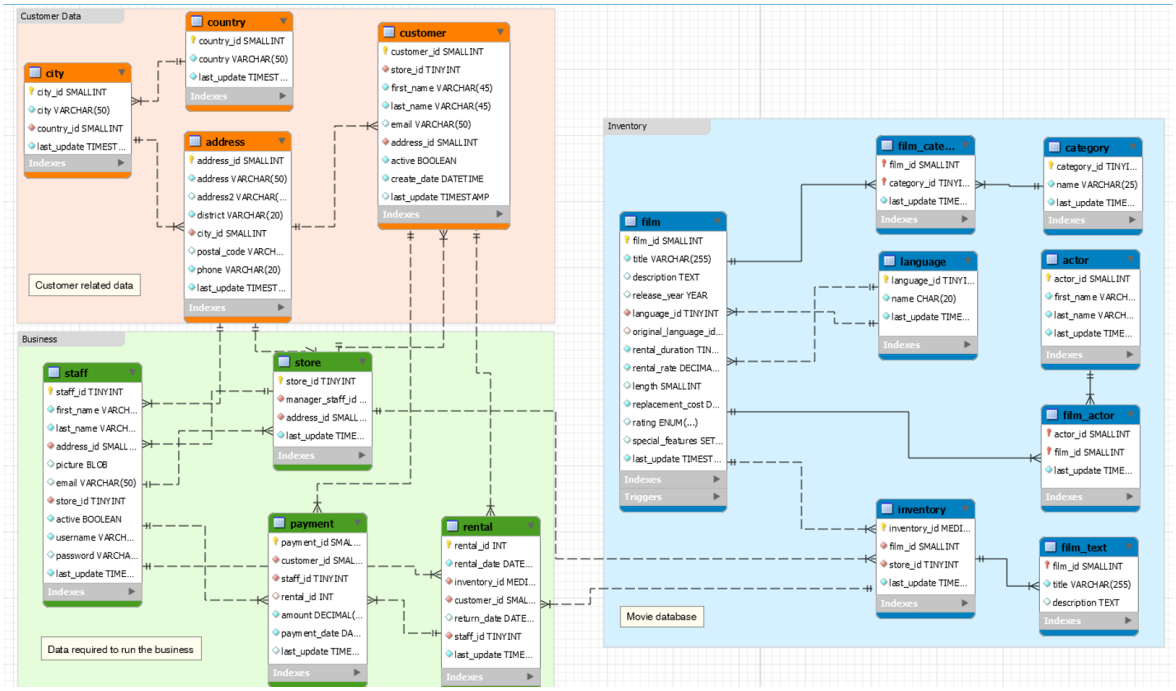


4. Show your result table in PostgreSQL.

5. You will create a Pentaho transformation file with a file extension .ktr. Remember to include this file as well in your deliverables.

You should submit one ZIP or RAR file containing all the files required above.

GSB 520
Data Integration Project
Nathan Diekema, Arash Akhavi, Jared Mcmullen, Thomas Roske



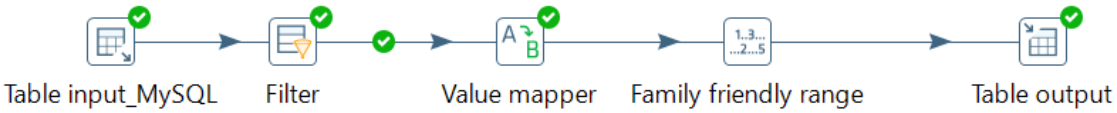
Transformation 1:

This transformation labels movies as “kid-friendly” and “not kid-friendly” based on movie rating. Both ‘G’ and ‘PG’ are considered kid-friendly whereas ‘PG-13’, ‘NC-17’ and ‘R’ are considered not kid-friendly. This is important because some customers (such as parents) may ask for specifically kid friendly movies. This transformation will make this easier.

Assumptions

- Film.title is a string and is not null
- Film.rating is not null and is one of the values: G, PG, PG-13, NC-17, or R
- Category.category is a string and is not null

The first step of this transformation is to filter out any data entry that is missing a title. The next step is to map the rating values to integers from 1-5 on a scale where G is 1 and R is 5. Next, we will use the range function to assign a ‘Yes’ under the kid_friendly column if the rating value is 1 or 2, and assign ‘No’ for values 3-5. If the movie does not have a rating associated with it, that row will be set to ‘Unknown’ in the kid_friendly column.



	title character varying (128)	rating character varying (5)	category character varying (25)	kid_friendly character varying (255)
1	AMADEUS HOLY	PG	Action	Yes
2	AMERICAN CIRCUS	R	Action	No
3	ANTITRUST TOMATOES	NC-17	Action	No
4	ARK RIDGEMONT	NC-17	Action	No
5	BAREFOOT MANCHURIAN	G	Action	Yes
6	BERETS AGENT	PG-13	Action	No
7	BRIDE INTRIGUE	G	Action	Yes
8	BULL SHAWSHANK	NC-17	Action	No
9	CADDYSHACK JEDI	NC-17	Action	No
10	CAMPUS REMEMBER	R	Action	No
11	CASUALTIES ENCINO	G	Action	Yes
12	CELEBRITY HORN	PG-13	Action	No
13	CLUELESS BUCKET	R	Action	No
14	CROW GREASE	PG	Action	Yes

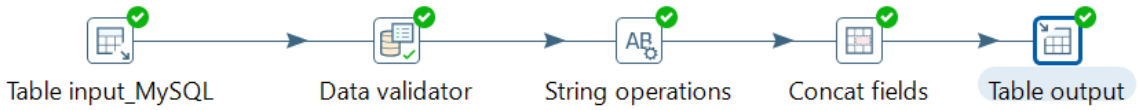
Transformation 2:

This transformation concatenates actors and actresses first name and last name into a column named Full_Name. Each actor is attached to one to many movie category names such as “Action” or “Comedy”. It can be useful when wanting to search up what actors star in what genre of movies. For instance, if you were to search JONNY LOLLOBRIGIDA, multiple genres would be attached to his name, giving customers an understanding of what types of movies an actor is in.

Assumptions:

- first_name & last_name only have spaces surrounding their string value when performing the string operation.
- first_name & last_name and both strings
- An actor/actress is attached to at least one genre field

The first step in this transformation is to validate both first and last name as string values. If they are not, an error will occur. Next, spaces are taken out of the first_name and last_name fields so they can concatenate. The next step is performing the concatenation between those two fields.



	Category character varying (25)	Full_Name text
354	Action	OPRAH KILMER
355	Action	NICK STALLONE
356	Action	CHRISTOPHER BERRY
357	Action	ELLEN PRESLEY
358	Action	JIM MOSTEL
359	Action	MICHAEL BENING
360	Action	PARKER GOLDBERG
361	Action	KIRSTEN AKROYD
362	Action	SPENCER DEPP
363	Action	MATTHEW CARREY
364	Animation	REESE KILMER
365	Animation	JADA RYDER
366	Animation	ANGELA WITHERSPOON
367	Animation	OPRAH KILMER

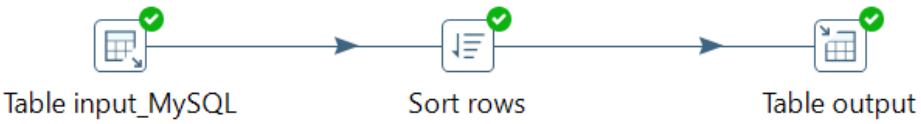
Transformation 3:

This transformation returns a sorted list in terms of the “most lucrative” movie categories which is determined by total sales with respect to each category. This provides valuable business information because it allows the company to observe which film categories are generally more popular and makes more money for the company. This may aid in business decisions that concern which movies would likely sell better and to forecast sales for certain movies.

Assumptions:

- TotalSales is a numeric value
- Every category has at least one sale associated with it

This transformation begins with a SQL query that returns the total sales with respect to each category. Next, the rows are sorted by total sales and output to the table seen below. As you can see, sports films are the most profitable for this company.



	categoryname character varying (25)	totalsales numeric (29,2)
1	Sports	5314.21
2	Sci-Fi	4756.98
3	Animation	4656.30
4	Drama	4587.39
5	Comedy	4383.58
6	Action	4375.85
7	New	4351.62
8	Games	4281.33
9	Foreign	4270.67
10	Family	4226.07
11	Documentary	4217.52
12	Horror	3722.54
13	Children	3655.55
14	Classics	3639.59

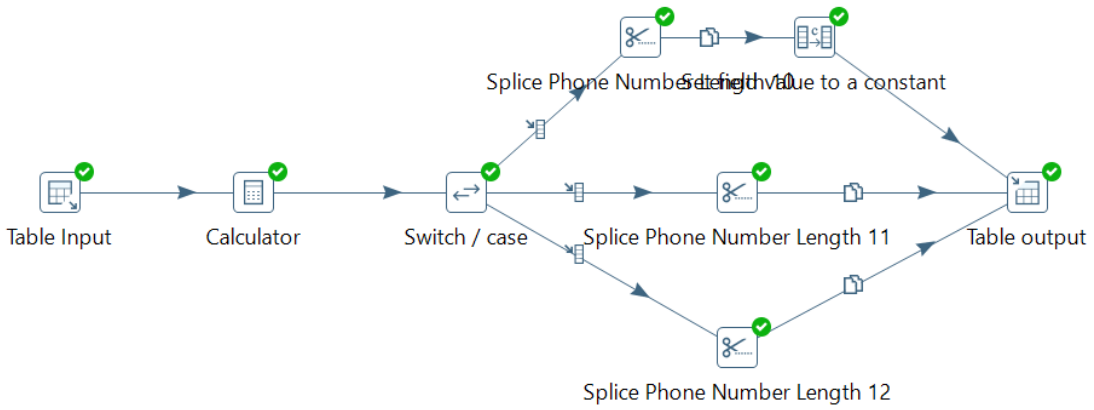
Transformation 4:

In this transformation the length of a customer's phone number will be calculated to determine the customer's unique country code. From here as an employer you can easily find where a customer is located, instead of using their district or street information. It also makes it easier to call a customer for marketing purposes.

Assumptions:

- First_name, Last_name, and district are not null
- Phone_number is not null and is a varchar of either length 10, 11, or 12

This transformation begins by calculating the length of every phone number. Next, it is passed through a switch/case object which will decide which step is next based on the length of the phone number. If the phone number is of standard length 10, it will return 'NA' for the country_code. If the phone number is of length 11, or 12, it will be split into two columns: area code, and phone number. The final output can be seen below.



	first_name character varying (45)	last_name character varying (45)	district character varying (20)	phone_number character varying (100)	country_code character varying (100)
1	PATRICIA	JOHNSON	California	8635286649	83
2	LINDA	WILLIAMS	Attika	8477190408	44
3	BARBARA	JONES	Mandalay	5814003527	70
4	JENNIFER	DAVIS	Texas	0452626434	86
5	MARIA	MILLER	Central Serbia	6571220373	71
6	SUSAN	WILSON	Hamilton	7282285970	65
7	MARGARET	MOORE	Masqat	0657522649	38
8	DOROTHY	TAYLOR	Esfahan	8856936185	64
9	LISA	ANDERSON	Kanagawa	5297277345	63
10	NANCY	THOMAS	Haryana	5887807014	46
11	KAREN	JACKSON	Osmaniye	5479687538	69
12	BETTY	WHITE	California	7338314235	51
13	HELEN	HARRIS	Madhya Pradesh	0911107354	99
14	SANDRA	MARTIN	England	9312333307	94

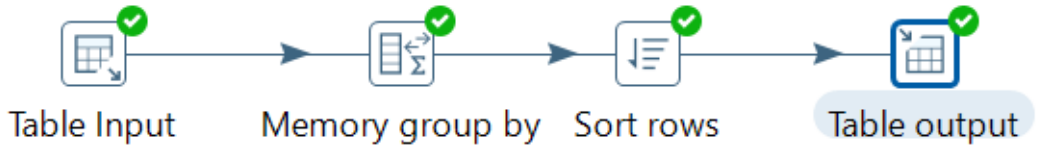
Transformation 5:

This transformation will take the amount of money generated by each genre of film in the data set. The transformation will group the amount of money generated by each respective genre then sort the data into descending order from the genre generating the highest amount of money to the lowest. This is important as it gives analysts an idea of which movie genres are generating the most amount of money.

Assumptions:

- Memory group by function will sort the amount data to each respective genre associated with the amount.
- Amount data is a numeric value and not null.

The first step of this transformation is to utilize the memory group by function to group by the names which correspond to the genre of the movies in the dataset. This step will also group the amount of money generated to the genre of movie by using the sum option to sum the “amount” values for each respective genre. Then we use the sort rows function to arrange the data in descending order from most money generated to least.



	name character varying (25)	amount double precision
1	Sports	138295.47
2	Animation	137116.48
3	Action	130684.74
4	Family	129048.71
5	Sci-Fi	127768.37
6	Drama	122052.28
7	Documentary	122046.18
8	Foreign	120780.5
9	Games	114451.03
10	New	110441.67
11	Comedy	110384.27
12	Children	109727.41
13	Classics	108921.82
14	Horror	98382.66