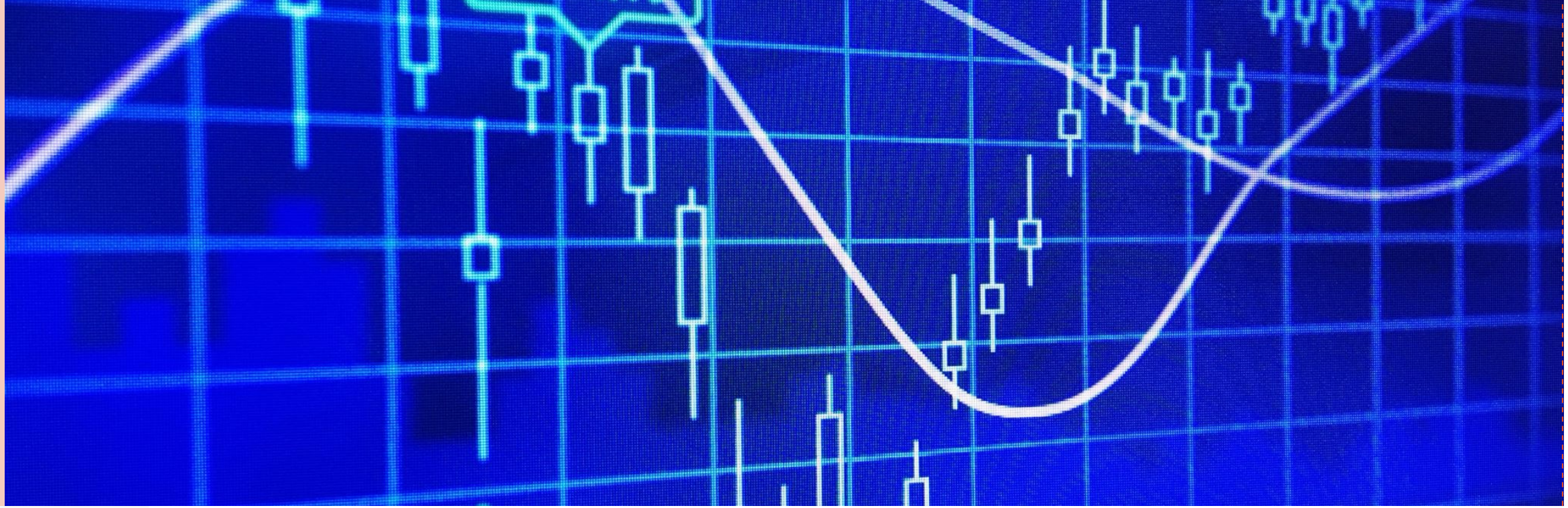


Ramani's POV
QA Script Generation
Relational Data Projects
Jan 2024

Ramasubramaniam Ramani
https://www.linkedin.com/in/rramani44/

Data Project Observations

Focusing on Data centric projects with relational data stores– data lakes , migration , conversion

50% or more data movements involve no or minimal transformations - identifiers , dates , amounts get moved from layer to layer.

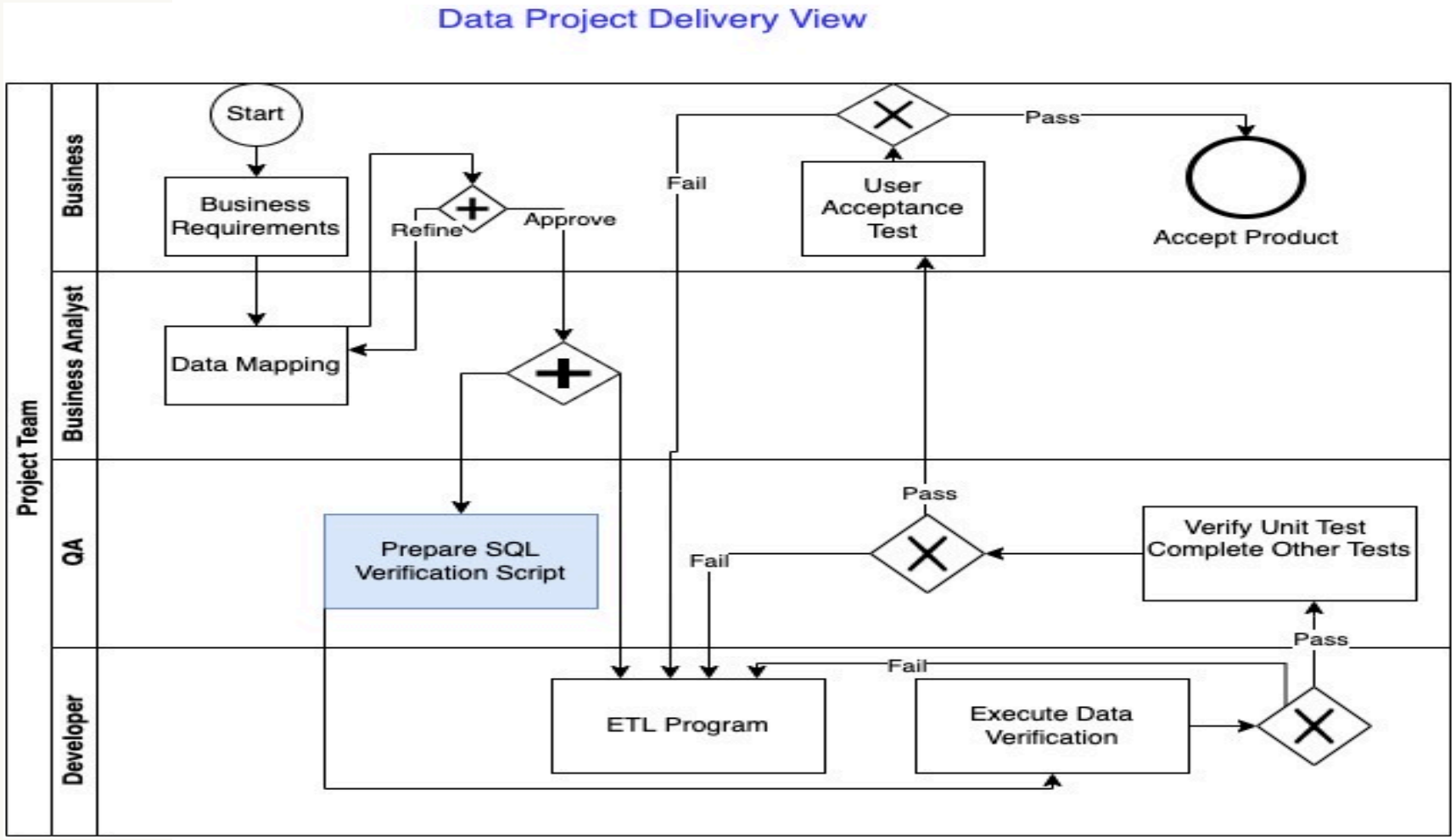
Summarizations - Rolled up dimensions

Business can test only a small percentage of the data during functional testing (UI , reports etc)

Technical team must verify the full dataset in the target – conformed (mapped) , complete , controlled (reconciled)

The QA team must certify the full dataset before business testing can begin

Illustrative Process



The mapping must be approved by business. They might delegate to BA team

BA team works with business and technical teams to complete business and technical maps

Automate the SQL generation as much as possible. Currently this is very manual

Developer will unit test with a few records

Solution Features

Used with a relational database to execute the SQL

Excel macro generates SQL

Compares all rows and columns and reports differences

Straight Pull

Defaults

Functions

Summarizations

Reference Table Lookup

Foundation for Operations dashboard

Could be incorporated into a Devops pipeline

Estimated saving between ½ hour to ½ day per ETL pipeline

Sample Output

Discover ! Experiment with the Excel worksheet at
https://github.com/ramanirepo/QASqlGen/blob/main/generateCompareSQL_v_1_0.xlsm

Simple pulls and transformations

interfaceName	RRTEST1	GroupNumber	1	sourceTableName	poc.table2	targetTableName	poc.table1	sourceWhere	id <> 3	targetWhere	
sourceColumn	targetColumn	sourceTransformation	primaryKeyFlag								
id	id		Y								
myChar	myChar										
newColumn	myDefault	'ab'									
myinteger	myinteger		10								

interfaceName	groupNumbe	s_id	t_id	s_myChar	t_myChar	s_newColumn	t_myDefault	s_myinteger	t_myinteger	d_t_myChar	d_t_myDefa	d_t_myinteger	
RRTEST1	1	1	1	RR	RR	ab	1	1	1	0	1	0	
RRTEST1	1	2	2	PR	PR	ab	1	2	2	0	1	0	

The difference column indicators are on far right

The columns are listed as pairs – source followed by Target

Sample Output

Discover ! Experiment with the Excel worksheet at https://github.com/ramanirepo/QASqlGen/blob/main/generateCompareSQL_GroupBy_v_1.xlsm

Summarizations

interfaceName	RRSUM1	GroupNumber	1	sourceTableName	stateSales	targetTableName	groupsumma	sourceWhere		targetWhere	
sourceColumn	targetColumn	sourceTransformation	primaryKeyFlag / GroupBy								
stateCode	stateCode		Y								
totSalesAmount	salesAmountT	sum(salesAmount)									

interfaceName	groupNumber	s_stateCode	t_stateCode	s_totSalesAr	t_salesAmountTot	d_t_salesAmountTotal	
RRSUM1	1	TX	TX	1001	1000	1	

The difference column indicators are on far right

The columns are listed as pairs – source followed by Target

Sample Output

Discover ! Experiment with the Excel worksheet at
hhttps://github.com/ramanirepo/QASqlGen/blob/main/generateCompareSQL_Ref_Table_v_1.xlsm

Reference Table Lookup

sourceTableName	employees.current_dept_emp	targetTableName	employees.v_full_employees_with_dept	referenceTableName	employees.departments		
interfaceName	RRREF1	GroupNumber	1	sourceWhere	emp_no between 10001 and 10100	targetWhere	emp_no between 10001 and 10100
sourceColumn	targetColumn	referenceColumn	sourceTransformation	primaryKeyFlag	Reference Key Flag	RefAsSourceFlag	
emp_no	emp_no			Y			
dept_no	dept_no				Y		
	department	dept_name				Y	

interfaceNar	groupNumber	c_emp_no	t_emp_no	c_dept_no	t_dept_no	c_dept_nam	t_departmer	d_t_dept_no	d_t_departm
RRREF1	1	10001	10001	d005	d005	Developmen	Developmen	0	0
RRREF1	1	10002	10002	d007	d007	Sales	Sales	0	0
RRREF1	1	10003	10003	d004	d004	Production	Production	0	0
RRREF1	1	10004	10004	d004	d004	Production	Production	0	0

The difference column indicators are on far right – In this example , there are no differences

The columns are listed as pairs – source followed by Target