Baker Hughes

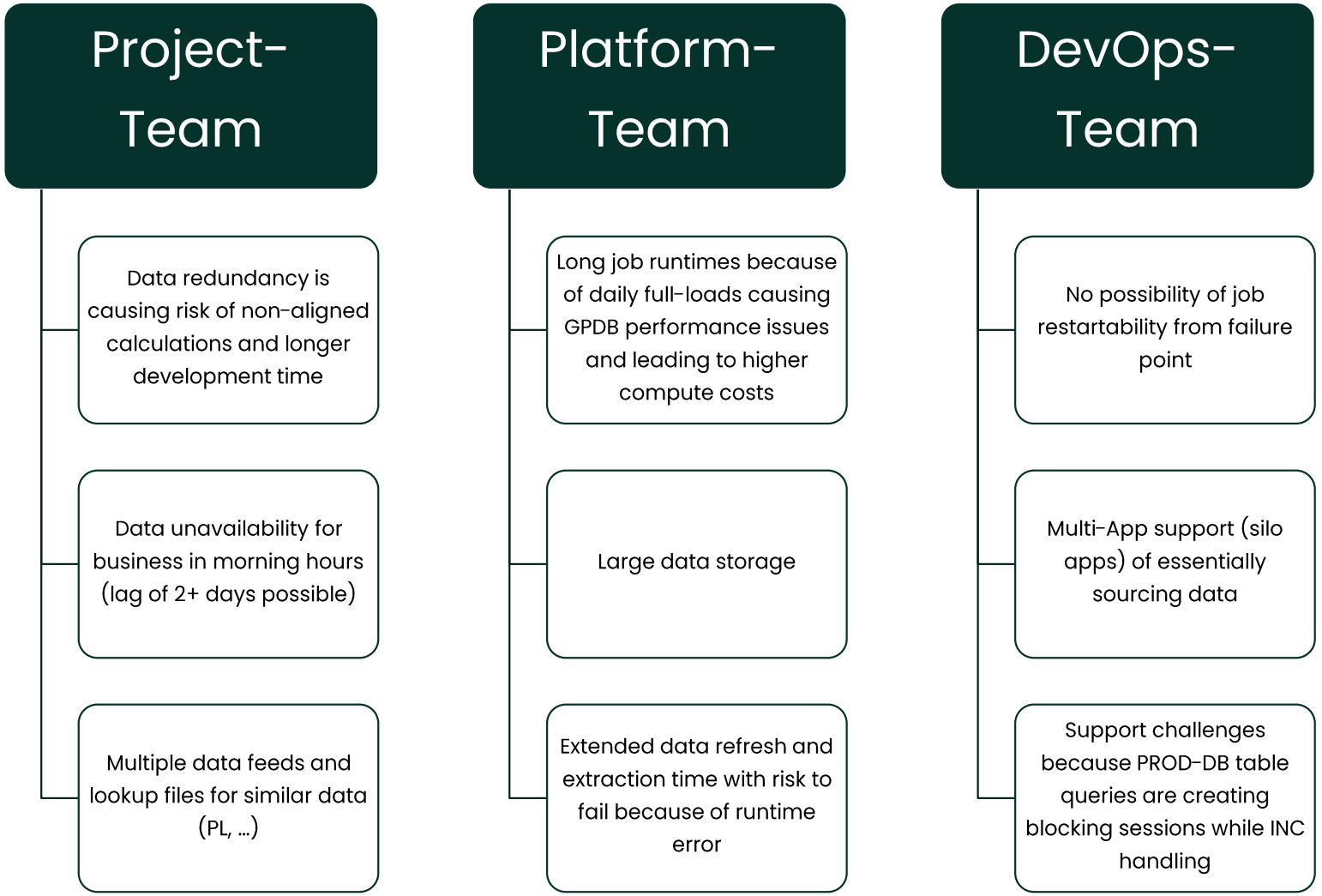
Global Sourcing Analytics

Re-architecture & Key Considerations

Marc Matthesius – Sr. Global Technical Program Manager
Suresha Jayappa – Sr. Solution Architect and Scrum Master

April 11, 2020

Sourcing Analytics: Considerations – Challenges



Baker Hughes

Sourcing Analytics: Solution & Benefits

Solution: Re-architecture

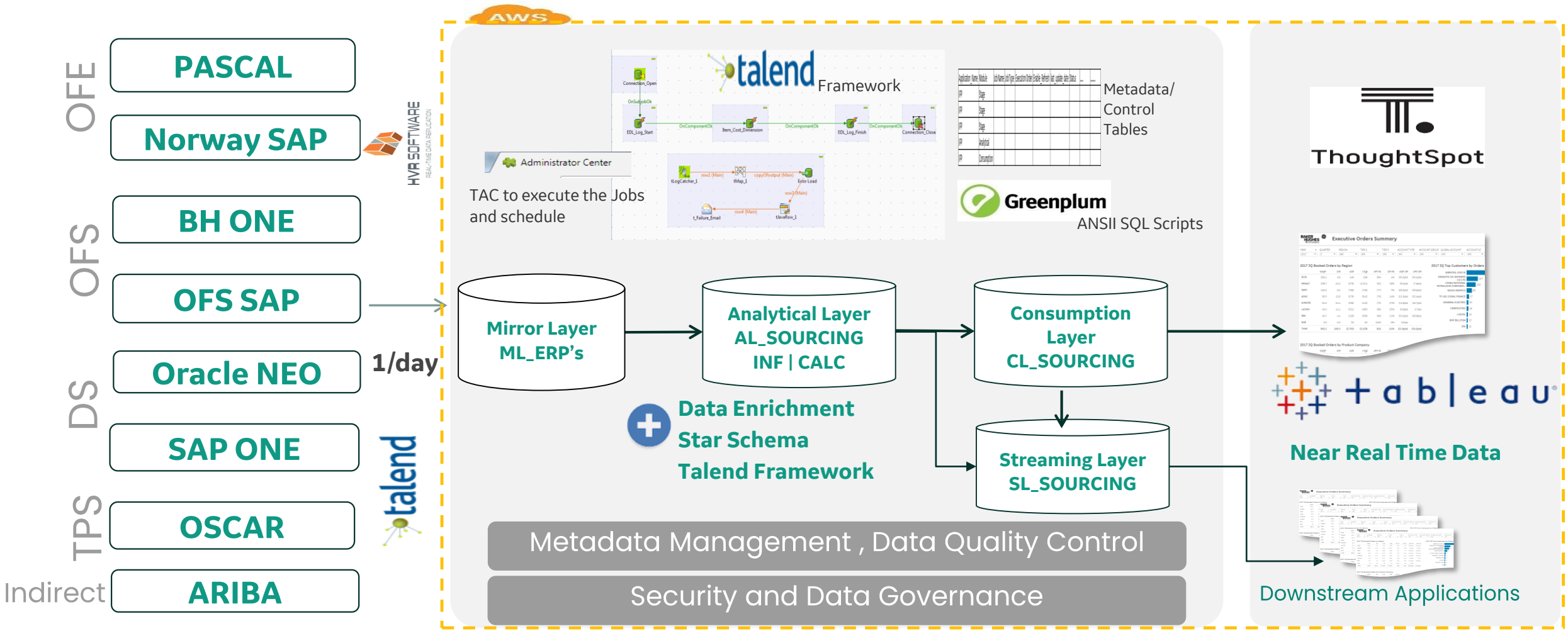
- Move from daily full loads to incremental loads
- Move from a de-normalized model to a star-schema model
- Introduce interface layer for script consolidation (Oracle, ERP, Other)
- Talend Framework and ANSII SQL with job restartability at failure point
- Further parallelization and table independency
- ERP decommisioning based on ERP integration strategy
- Integrate existing dimensions from other domains like Materials/Finance (e.g. Item Master)
- Attribute mapping and data lineage for key silo app-reports to integrate in GSA solution

Benefits:

- Cost reduction for data storage
- Cost reduction for data processing
- Cost reduction for visualization tools
- Readiness for next-gen datalake (technolgy independent code)
- Quicker/Faster data availability and reduced refreshed times
- Elimination of GE dependencies
- Single source of truth for sourcing analytics (silo app reduction)
- Reduced future development effort
- Restartability of jobs at failure point
- Easy connect to other data models (plug and analyze)
- Usage increase/adoption of GSA

Baker Hughes

Data Flow Design – Talend Framework and ANSII SQL



Baker Hughes

GSA – Database Schema Layers And Naming Convention

Mirror Layer (ML)

- Schema Name:
  - GOG_ERPNAME
  - OG_ERPNAME
  - GOG_DATAFORMS
  - **ML**_xxxxx (New)
- Table Names
  - Same as ERP

Analytics Layer (AL)

- Schema Name:
  - **AL_SOURCING**
- Table Names
  - Dimensions : GSA_**DIM**_XXXXX
  - Facts : GSA_**FCT**_XXXXX
  - Interface : GSA_**INF**_XXXXX
  - Lookup : GSA_**LKP**_XXXXX
  - Temporary : GSA_**TMP**_XXXXX
  - Views : GSA_**VW**_XXXXX
  - Snapshot : GSA_**SNP**_XXXXX
  - Control : GSA_**CTL**_XXXXX

Consumption Layer (CL)

- Schema Name:
  - **CL_SOURCING**
- Table Names
  - Interface : GSA_**INF**_XXXXX
  - Consumption: GSA_**RPT**_XXXXX
  - Views : GSA_**VW**_XXXXX
  - Snapshot : GSA_**SNP**_XXXXX

Streaming Layer (SL)

- Schema Name:
  - **SL_SOURCING**
- Table Names
  - Views : GSA_**VW**_XXXXX

Acronyms :

- ML : Mirror Layer
- AL : Analytical Layer
- CL : Consumption Layer
- SL : Streaming Layer (Downstream)
- DIM : Dimension
- FCT : Fact
- INF : Interface
- LKP : Lookup
- TMP : Temporary
- RPT : Reporting
- VW : Views
- SNP : Snapshot (Incremental copies of data)

Columns Naming Conventions :

- _ID : Key /unique Identifier
- _DT : Date Field
- _NAME : Name of the Attributes
- _DESC : Description of the Attributes
- _CD : Code Field
- _FLAG : Flag

- GOG / OG : Oil & Gas
- GSA : Global Sourcing Analytics

Refer : https://devcloud.swcoe.ge.com/devspace/display/NPGBW/Data+Modeling+Standards

Baker Hughes

Naming Conventions – Talend ETL FRAMEWORK

Control Tables

Schema Name:

- AL_SOURCING

Table Names:

GSA_**MTD**_ETL_JOBS
GSA_**MTD**_ETL_SCRIPTS
GSA_**MTD**_ETL_LOG
GSA_**MTD**_ETL_STAT_HIST

Acronyms : **JOB** : Talend Jobs , **HIST** : History , GSA : Global Sourcing Analytics

Talend Project Name :

AWS_DEV_GLOBAL_SOURCING_ANALYTICS
AWS_QA_GLOBAL_SOURCING_ANALYTICS
AWS_PROD_GLOBAL_SOURCING_ANALYTICS

Talend Folder Name :

GLOBAL_SOURCING_ANALYTICS /GSA_ETL_ANALYTICAL_LAYER
GLOBAL_SOURCING_ANALYTICS/GSA_ETL_CONSUMPTION_LAYER
GLOBAL_SOURCING_ANALYTICS/GSA_ETL_STREAMING_LAYER

Talend Job Names :

Master Job : JOB_GSA_<<AL/CL/SL>>_ETL_MASTER
Child Job : JOB_GSA_<<AL/CL/SL>>_ETL_CHILD
Notify Job : JOB_GSA_<<AL/CL/SL>>_ETL_NOTIFY
Reset Job : JOB_GSA_<<AL/CL/SL>>_ETL_RESET

Project name	Sub Folder	Job name	Frequency	Status	Environment	Email Subject
SC (GSA)	APCOE	Job_ML_Sourcing_SFTP2Datalake_Invoice_holds	Daily	SUCCESS	DEV	SC (GSA): Job_ML_Sourcing_SFTP2Datalake_Invoice_holds Daily SUCCESS *** DEV ***
SC (GSA)	APCOE	Job_ML_Sourcing_Box2Datalake_Perceptive	Daily	FAILED	QA	SC (GSA): Job_ML_Sourcing_Box2Datalake_Perceptive Daily FAILED *** QA ***
SC (GSA)	APCOE	Job_ML_Sourcing_Sharepoint2Datalake_Bank	Daily	FAILED	PROD	SC (GSA): Job_ML_Sourcing_Sharepoint2Datalake_Bank_Details Daily FAILED *** PROD ***
SC (GSA)	Strategic ERP	Job_AL_Sourcing_ Strategic ERPs	Daily	SUCCESS	PROD	SC (GSA): Job_AL_Sourcing_ Strategic ERPs Daily SUCCESS *** PROD ***
SC (GSA)	Independent	Job_CL_Sourcing_Independent	Weekly	SUCCESS	PROD	SC (GSA): Job_CL_Sourcing_Independent Daily SUCCESS *** PROD ***
Domain (Project Abbreviation)	Sub Folder Name	Job_</MLL>_<Project Name>_<Source>2<Target>_<Name> Job_<AL/CL/SL>_<Project Name>_<Name>	<Daily/Monthly/Weekly/Fortnightly/Quarterly>	<SUCCESS/FAILURE>	<DEV/QA/PROD>	<Project Name> : <Job Name> <Frequency> <Status> <*** Environment ***>

Baker Hughes

GSA : ERP list and Plan for Migration to Strategic

Source ID	ERP Name	Major Business	Req.	PO	Receipts	Invoice	Payments	Holds	Disbursement	Migration plan to Strategic ERP	Tentative Migration Date	Plan for Remodeling (98 % Spend)	% Spend (FY 2019)
30	BH SAP	OFS	✓	✓	✓	✓	✓	✓	✓			✓	43.1 %
2	ORACLE OSCAR	TPS	✓	✓	✓	✓	✓	✓	✓			✓	33.2 %
1	ORACLE PASCAL	OFE	✓	✓	✓	✓	✓	✓	✓			✓	9.1 %
23	SAP ONE	DS	✓	✓	✓	✓	✓	✓	✓			✓	5.3 %
16	SSS**	ALL	✓	✓	✓					ARIBA	Q3-2020	✓	4.7 %
4	OFS SAP	OFS	✓	✓	✓	✓	✓	✓	✓			✓	1.5 %
6	ORACLE FT**	TPS		✓	✓	✓	✓	✓	✓	SAP ONE	DEC-2020 H2-2021		0.6 %
27	NORWAY SAP	OFE	✓	✓	✓	✓	✓	✓				✓	0.8 %
31	ALSTOM**	TPS		✓	✓	✓	✓	✓		GE Power	Q1-2020		0.7 %
28	ORACLE NEO	DS	✓	✓	✓	✓	✓	✓				✓	0.4 %
9	SYTELINE-AL	OFS		✓	✓	✓	✓	✓				✓	0.3 %
7	ORACLE NA**	TPS		✓	✓	✓	✓	✓	✓	SAP ONE	OCT-2020		0.2 %
11	ORACLE DVI**	TPS		✓	✓	✓	✓	✓		SAP ONE	Q1-2019		-
13	ERPLX	TPS		✓	✓					Sold Business			0.05 %
8	ORACLE S&I**	DS		✓	✓	✓	✓	✓		SAP ONE	Q4-2019		-
18	VISUAL MFG	TPS		✓	✓					SAP ONE	H2 - 2021		0.04 %
24	MAPICS	TPS		✓	✓					No data in 2020			0.01 %
14	SAGE	TPS		✓	✓					No Data in 2020			-
	KARIWA	TPS								SAP ONE	AUG-2020		0.9 %

** Historical data for marked ERP will move moved to ml_sourcing schema as per current structure

Baker Hughes

GSA: Key Considerations (1/2)

Source ID	Considerations	Major Business
1	Schema Design	○ Mirror Layer : Access or Select only incremental data based on Last update date or HVR Update date . Latest update date available in Control tables for respective interface tables ○ Interface Layer : Union/**Insert** on Oracle or SAP tables to load the respective dimensions incrementally ○ Analytica Layer : Snowflake Data Model ○ Consumption Layer : Star Schema Or De-Normalized Based on the requirements , Views to be created for the smaller tables rather storing data in tables ○ Streaming Layer : Views/tables with required columns will be created to provide access to downstream users ○ New Schema Creations [AL_SOURCING, CL_SOURCING, SL_SOURCING] ○ Resource Queue allocation based on data load strategy and storage and # of records processing daily basis ○ Resource Queue requirements for the historic data load
2	Table Design	○ Standard Naming convention (Interface, temporary, Dimension, Facts, Views, Reporting Tables and Columns _DT,NAME, DESC, ID, NUM etc,.) ○ Define right Distribution key , Natural Key is the right columns for distribution (Source ID ~ Table Primary Key) ○ Define Right Datatype (TEXT, VARCHAR, NUMERIC, DATE) , also consider data type use least space ○ Use Same Data type of columns considered in JOIN condition ○ Natural key definition to join with right columns set ○ Insert 1 row to each table with UNDEFINDED Description to preform inner join and avoid full outer join scenarios ○ Partition large Fact tables ○ Partition larger table to improve I/O Operations
3	Load Strategy	○ Incremental data load , Delete changed/new records and insert , This is for delta processing ○ Design and build an insert-only model, truncating a daily partition before load ○ Avoid Update statements ○ Mirror - > Interface -> Analytical (Snowflake : Dim/Facts) -> Consumption (Star Schema : Dim/Facts - Views) -> Streaming Layer ○ Restartability from failure point ○ Update control tables with from and to date for periodic data loads ○ Parallelization of data loads (Dim / Facts) independent irrespective an any sources
4	Talend Framework	○ New Project Folder and recommended structure , Centralized document repository ○ ANSII SQL ○ Metadata Tables to store labels, load type, Scripts and Audit log information ○ Configuration of Incremental or Full load strategy ○ Restartability of Jobs ○ Define dependencies if any ○ Email notifications ○ Talend Naming conversions (Job Name, Label , Name, email subject, body, metadata configurations)
5	Performance Best Practices	○ Skewness Check and define right distribution key ○ Explain plan and review cost, spilling and broadcast motion ○ Join tables with right columns and avoid data type mismatch ○ Avoid Subquery ○ Create temp table if necessary ○ ANALYZE large table before or after processing / **BLOAT Clearance** ○ Ref Additional Slides 8 and 9

GSA: Key Considerations (2/2)

Source ID	Considerations	Major Business
6	Functional Data Model Mapping And Transformation	○ Data Lineage and Model mapping of each attributes ○ Eliminate the Redundant information from all the models (example PO related information should be only from PO Models) ○ Transformation / Business logics only in Analytics layer model ○ Select only required columns from the interface table ○ Combine lookup tables to avoid duplicate data mapping other than ERP tables ○ Aggregation of data in Consumption table ○ Key columns rightly define to join the smaller table (example Supplier and Country Master, Purchase Org and Country, Item master and material type) ○ ThoughtSpot Catalog updates ○ Leverage Dimensions/Facts from other analytics models (Ex: Material Maser from Supply Chain Analytics) from the ○ Build master data tables from the de-normalized tables to avoid more joins, long run process and Volume of data ○ Strategic ERP's for data processing ○ Combine multiple models into one wherever applicable ○ Data Reconciliation between, Mirror, Analytical will be easier not filtering any records, in fact flag physically deleted or cancelled records or external filter using look-up table ○ 3 Box folder for the Input files – DEV , QA and Prod
7	Automation of Tableau Refresh	○ Build consumption layer tables with only required columns ○ Refresh notifications ○ Manual refresh dependencies
8	Automation of ThoughtSpot refresh and Star Schema Model	○ Star Schema Data Model ○ Automation of data extract and refresh ○ Incremental refresh using delete and upload mechanism for the dim/ facts tables ○ Write back incremental refresh details into metadata tables in bh_thoughtspot schema ○ Automated email notification
9	Interface layer Tables	○ Interface tables will be leveraged for the consolidation/Join of Mirror tables no Business logic will be applied here ○ Interface tables should be same naming conversions of fact table (History data restrictions + Filters like Document type + Incremental data) ○ Business logic should be defined at one place which is FACT load (SQL/Scripts/Job) ○ Truncate Interface Table but keep entire FACT table
10	Priority Data Loads	○ Data Availability to User Europe Morning Hours in ThoughtSpot ○ Parallelize the data load for Strategic Erp's (Oracle and SAP's)
11	Operations and Monitoring	○ FMEA Documents ○ Restartbaility procedure and incremental table updates ○ Execution Plan and Dependencies ○ Validation checks and procedures ○ Email notifications and details ○ ERP wise and Product Company wise IT and Business Stakeholder respectively

Baker Hughes

GSA: GPDB General Guidelines (1/2)

Area	Guidelines
Physical Data Model Design	De-normalization is the key for Greenplum database. More the flattened structure of the tables, better the IO throughput and query response time
	Highly normalized data models are not well supported . Star Schema and Dimensional models are supported as long the dimension tables are small.
Distribution Key (Do)	Ensure balanced distribution of data across the cluster by explicitly specifying DITRIBUTION KEY (DK)
	Preferably use the Primary key or a Unique key as the DK.
	If no Unique key is available for a table , choose DISTRIBUTED RANDOMLY option which will send the data in a round-robin manner to the segment instances
Distribution Key (Don't)	Never choose the same column for both Partition Key and Distribution Key
	Never use Date timestamp column as the DK
	Never choose a column as Distribution Key which is used in the WHERE clause as predicates.
Table Storage Orientation	Use APPEND ONLY tables where no DELETE/UPDATE operations are expected and data will be inserted in large batches
	Use HEAP storage for tables where Singleton (Row Operation and small batches) INSERT/UPDATE/DELETE operation will be performed
Row vs Column Oriented Table	Use ROW oriented table where UPDATE / DELETE operations are needed with frequent INSERTS
	Use ROW oriented table where majority of the columns are frequently accessed in Queries
	Use Column oriented tables where only a small subset of columns for a wide table are frequently accessed in queries along with aggregation functions..
Table Compression	Compression is recommended to be used for large APPEND ONLY tables to improve IO performance . Primarily useful for infrequently accessed table (Tables with Historical or Archival data).
	There is some CPU overhead while accessing data from Compressed table which needs to be considered carefully while using Compression option on a table.
Secondary Indexes	Indexes are NOT generally recommended and needed for GPDB tables
	Sometimes indexes are found useful for point queries on extremely large tables , however the impact of indexes on data loads and query plan must be carefully examined.
	If indexes are absolutely needed, drop indexes before the data is loaded and recreate the indexes after the load.
	Never create indexes on the partition key of a partitioned table.

GSA: GPDB General Guidelines (2/2)

Area	Guidelines
Table Partitioning	Partition key must be defined on column(s) which are commonly used in Query Predicates to ensure partition pruning (i.e. accessing the specific partitions or a small subset of the data)
	Avoid creating too many partitions on Column Oriented tables
	Never choose the same column for both Partition Key and Distribution Key
SQL Coding	Avoid writing complex SQL code or functions . GPDB optimizer can take better and consistent optimization decision on simple code.
General	Ensure the intermediate staging (temporary) tables are distributed evenly
	ANALYZE the intermediate staging tables with large volume of data to ensure GPDB optimizer can take the right decision.
	Avoid writing complex Views to support the consumption layer. Materialize them if needed. Data duplication is acceptable in GPDB to achieve required performance.
Architectural Guidelines	MPP databases like GPDB are NOT designed for large number of concurrent connections . It's designed to provide high throughput for large batch jobs , NOT to support large number of interactive queries

Performance
Tuning Techniques

GSA : ThoughtSpot Model Fact Joins

Where is it done?
- Build views / tables in consumption layer
- Full outer join on facts to avoid data mismatch
- Introduce unified dummy row in each fact tables

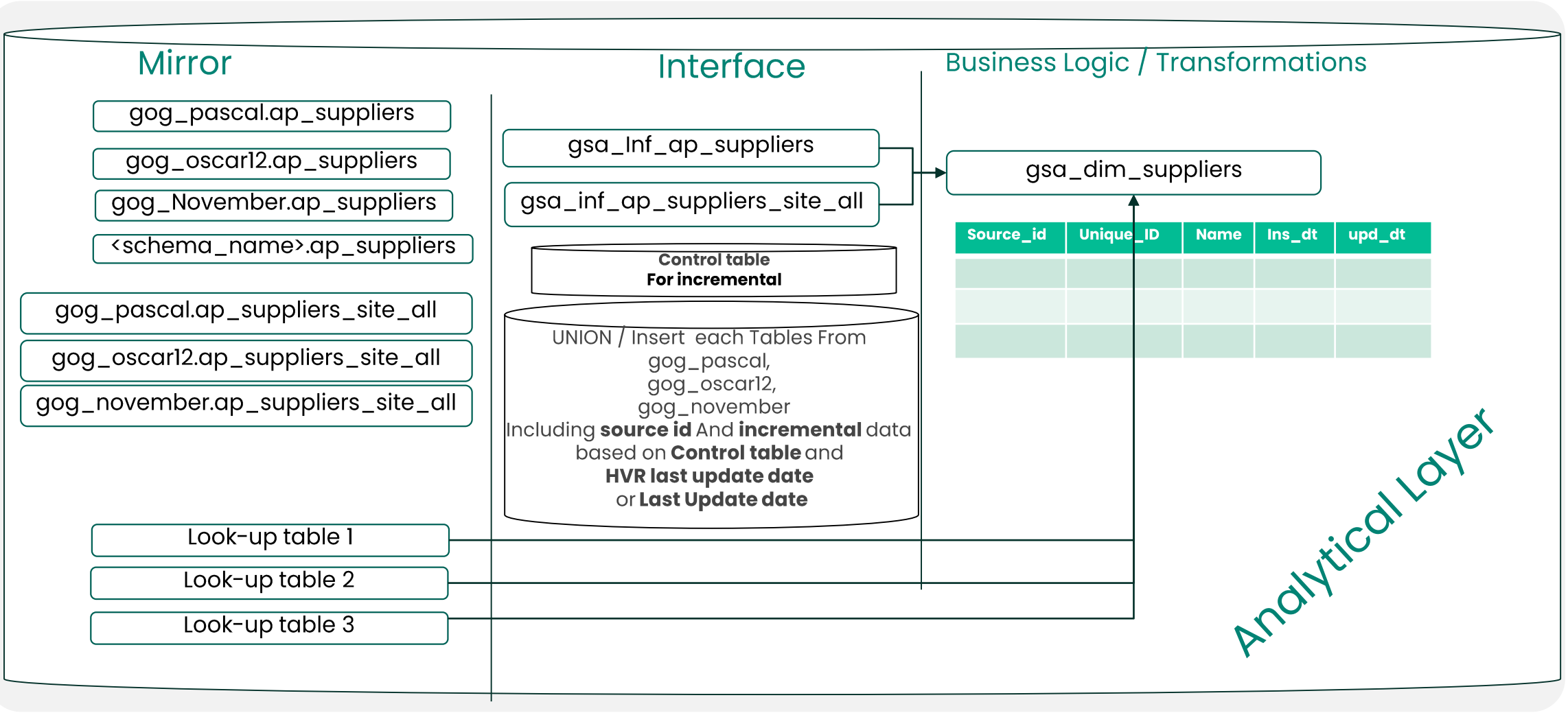
What are the Keys?
- Requisition table join with PO Schedule line join with PO Lines join with PO Headers using defined keys(PO Line Location ID [Oracle] / PO Line ID [SAP])
- Dimension table keys will be associated

How will it work?
- Build summary at PO header or line level
- Views data to be extracted to file and upload into ThoughtSpot, it will act as summary fact
- Star schema created with multi fact tables join with all other dimension

Baker Hughes

GSA: Supplier and Sites Dimension - Oracle

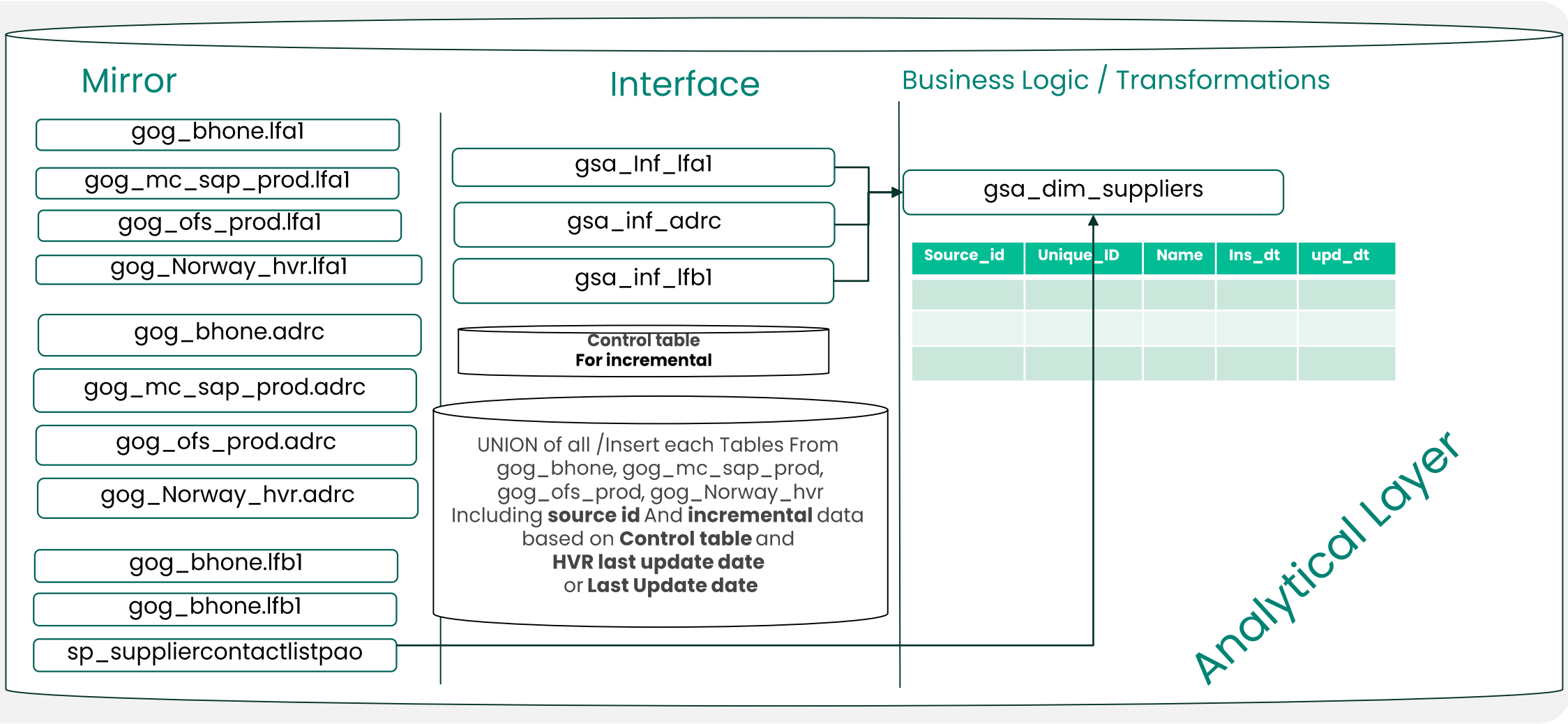
Example



Baker Hughes

GSA: Supplier and Sites Dimension – SAP

Example



Baker Hughes

Baker Hughes