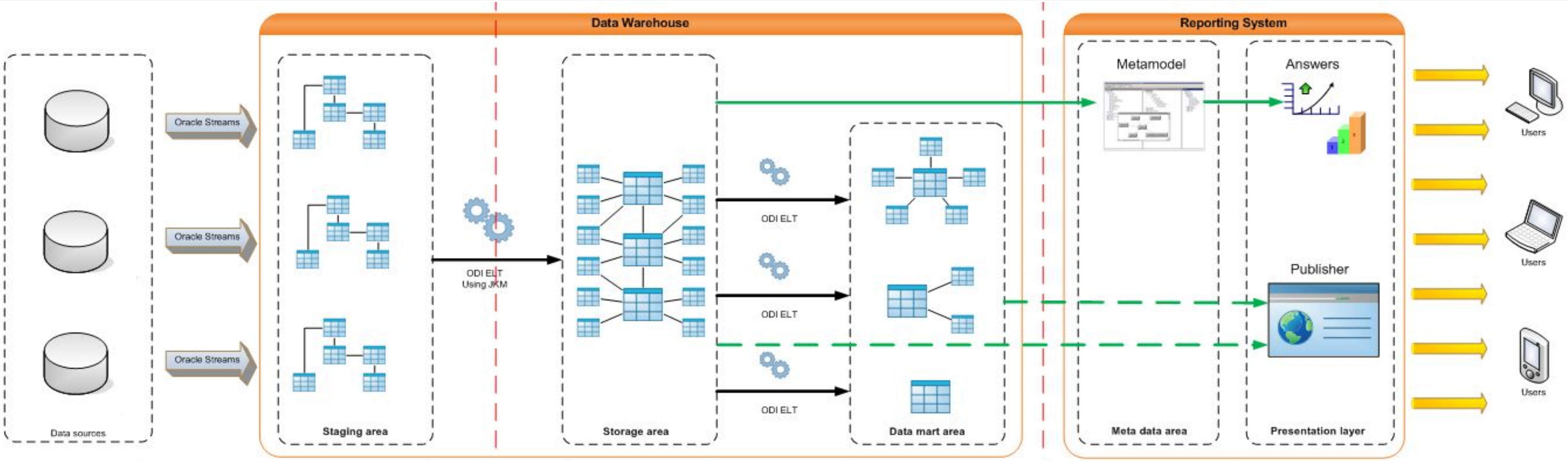
<epam>

Agenda

- DWH Testing
- Typical Data Issues

DWH TESTING



OLTP databases of different systems.

Real-time data capture using Oracle downstream CDC technology. In order to decrease DWH impact on production databases, data capture is carried out on DWH server side.

Staging area is represented by BISTG schema in DWH database. It contains data replicated from the source systems via streams.

ELT process catches changed data from the staging area, transform it and load into the storage area in near real-time manner. ODI is used here.

Storage area uses dimensional approach to keep historical and new coming data. It contains conformed data that is ready for further transformations and aggregations. This area is represented by BISTORE, BISYS and BIREC schemas.

ODI ELT processes populate pre-aggregated summary table sets aka mini data marts for reporting purposes. Each process has its own schedule to run. It depends on how often each particular mini data mart should be refreshed with more recent data (every hour, day or month).

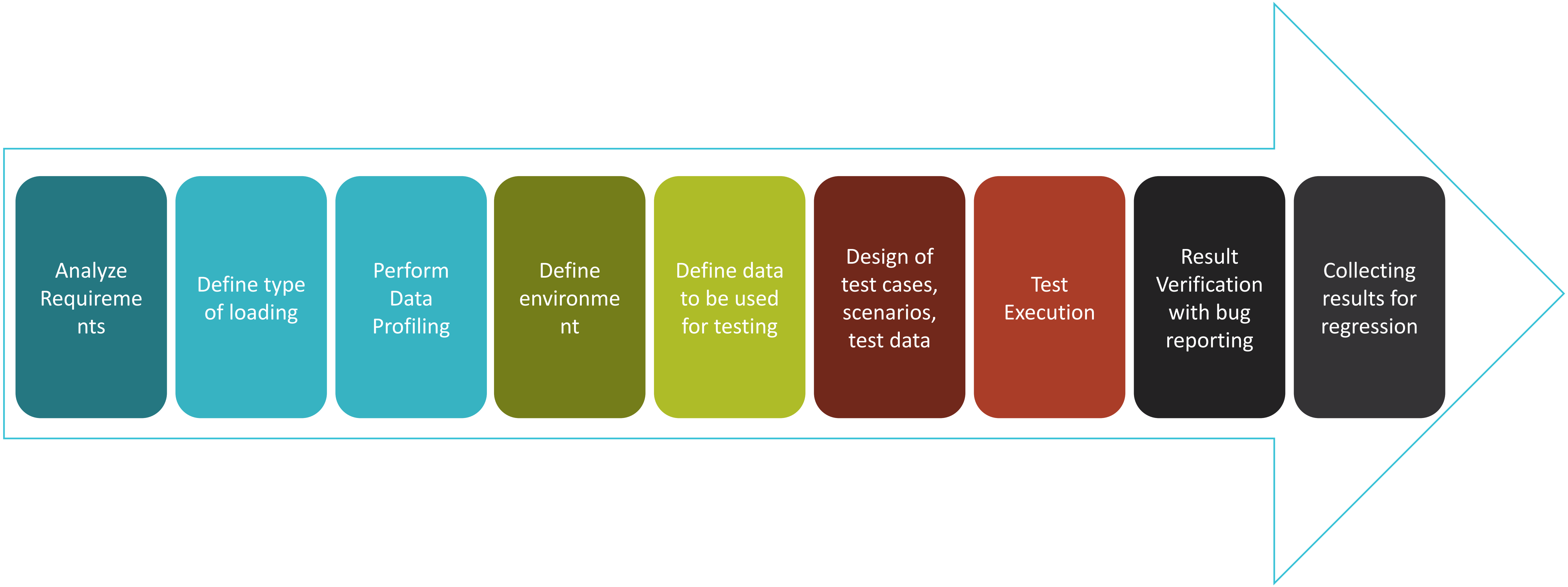
Data mart area contains mini data marts. Mini data marts may consist of several filtered dimension tables and an aggregated fact table with links to them or one aggregated fact table that includes all required dimension attributes.

Meta data area is represented by Metamodel object. Metamodel describes a business model with relationships between tables, with a list of dimensions and calculated members, which will be used for report creation.

Presentation layer consists of two major modules – Publisher (for operational reports) and Answers (for ad-hoc analysis) - which provide end users with ability to analyze structured and aggregated data from DWH using charts, pivots, tables and dashboards.

DATA SHOULD BE VERIFIED AT EVERY DWH LAYER

DQE Workflow



Analyze Requirements

Define type of loading

Perform Data Profiling

Define environment

Define data to be used for testing

Design of test cases, scenarios, test data

Test Execution

Result Verification with bug reporting

Collecting results for regression

DATA PROFILING

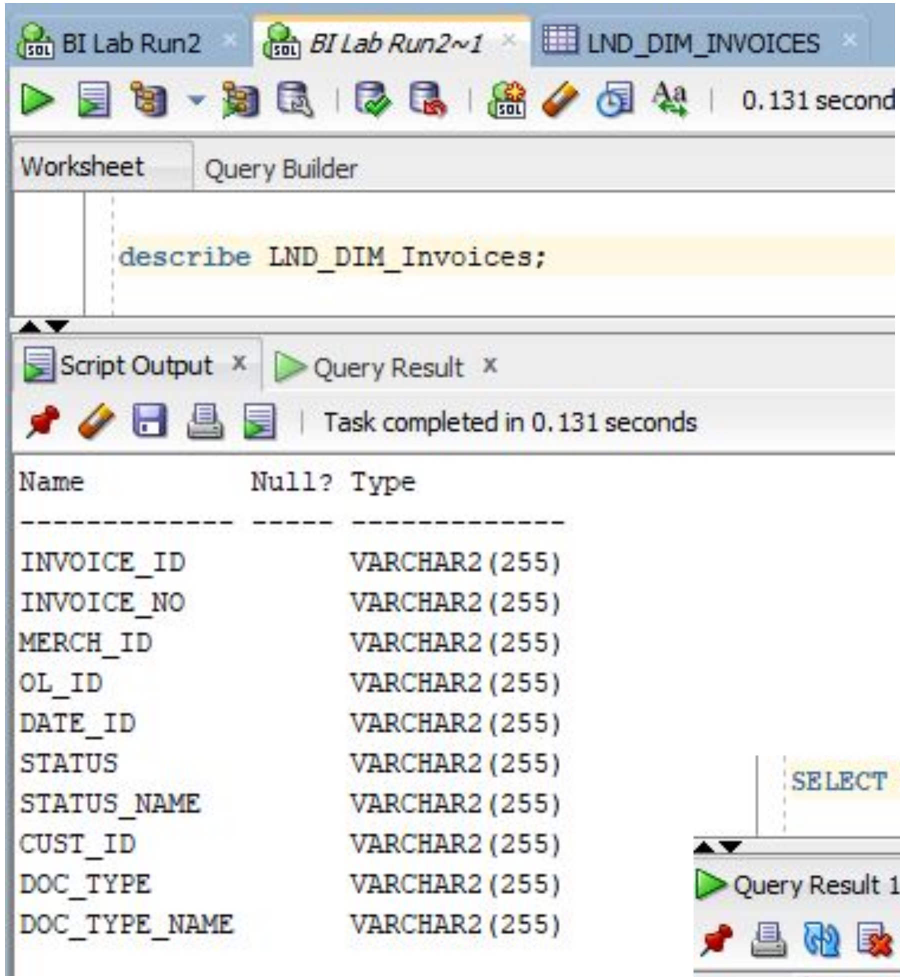
Analyze source data before and after extraction to landing

- extract representative data from each source file
- parse data for the purpose of profiling
- structure, relationship, data discovery
- check for unusual cases

DATA PROFILING – UNUSUAL CASES Examples

- When data loaded into DWH from 2 different databases (SQL Server, Maria DB) for one DB datetime will be extracting for client time (Europe), and for other DB for source DB time (India), which can cause discrepancies when uniting this datasets.
- Different values in DB for same business key. For example we have 20201231 and 20200101 periods. We need only year (and it's the same), but we also need to save MM and DD in the DB because it is standard for all period in DB. How we should handle this situation, add new logic to capture both periods as one or use **UPDATE** and make all values consistent?
- We have only int values in a column dataID in source, but company that provides source to us keep this column **VARCHAR**. How we make sure that it won't cause problems in future?
- In source we have phone number +4402012345678, +44(020)12345678, 44(020)12345678, +44(020)-1234-5678, +44(020)1234-5678, +44020-1234-5678, etc. And it is the **same** phone number.

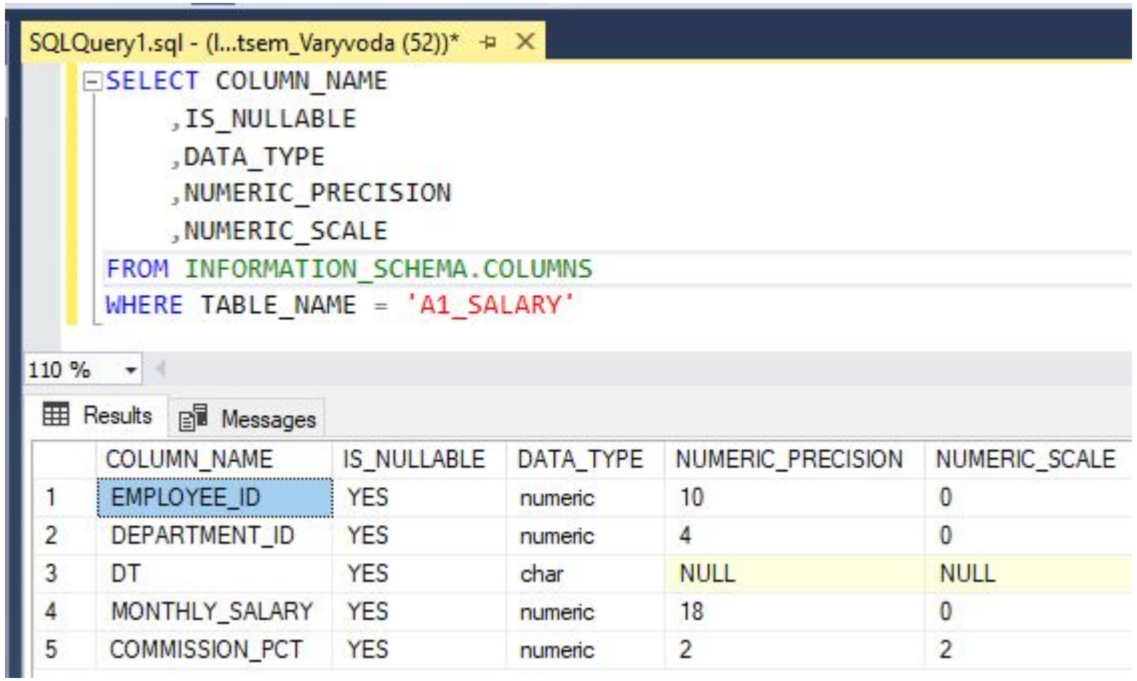
DATA PROFILING



```
describe LND_DIM_Invoices;
```

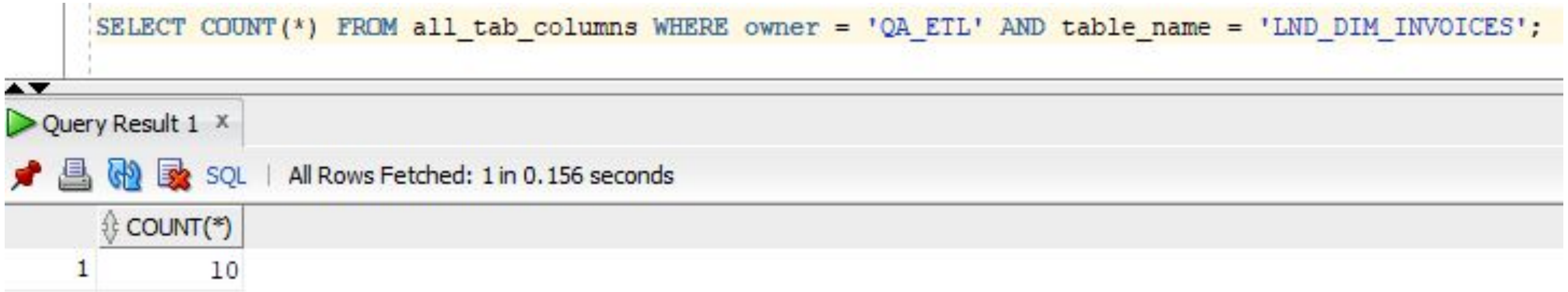
Task completed in 0.131 seconds

```
Name          Null? Type
------------- ----- -------------
INVOICE_ID          VARCHAR2(255)
INVOICE_NO          VARCHAR2(255)
MERCH_ID            VARCHAR2(255)
OL_ID               VARCHAR2(255)
DATE_ID             VARCHAR2(255)
STATUS              VARCHAR2(255)
STATUS_NAME         VARCHAR2(255)
CUST_ID             VARCHAR2(255)
DOC_TYPE            VARCHAR2(255)
DOC_TYPE_NAME       VARCHAR2(255)
```



```
SELECT COLUMN_NAME
      ,IS_NULLABLE
      ,DATA_TYPE
      ,NUMERIC_PRECISION
      ,NUMERIC_SCALE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'A1_SALARY'
```

	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	NUMERIC_PRECISION	NUMERIC_SCALE
1	EMPLOYEE_ID	YES	numeric	10	0
2	DEPARTMENT_ID	YES	numeric	4	0
3	DT	YES	char	NULL	NULL
4	MONTHLY_SALARY	YES	numeric	18	0
5	COMMISSION_PCT	YES	numeric	2	2



```
SELECT COUNT(*) FROM all_tab_columns WHERE owner = 'QA_ETL' AND table_name = 'LND_DIM_INVOICES';
```

All Rows Fetched: 1 in 0.156 seconds

	COUNT(*)
1	10

SOURCE DATA PROFILING

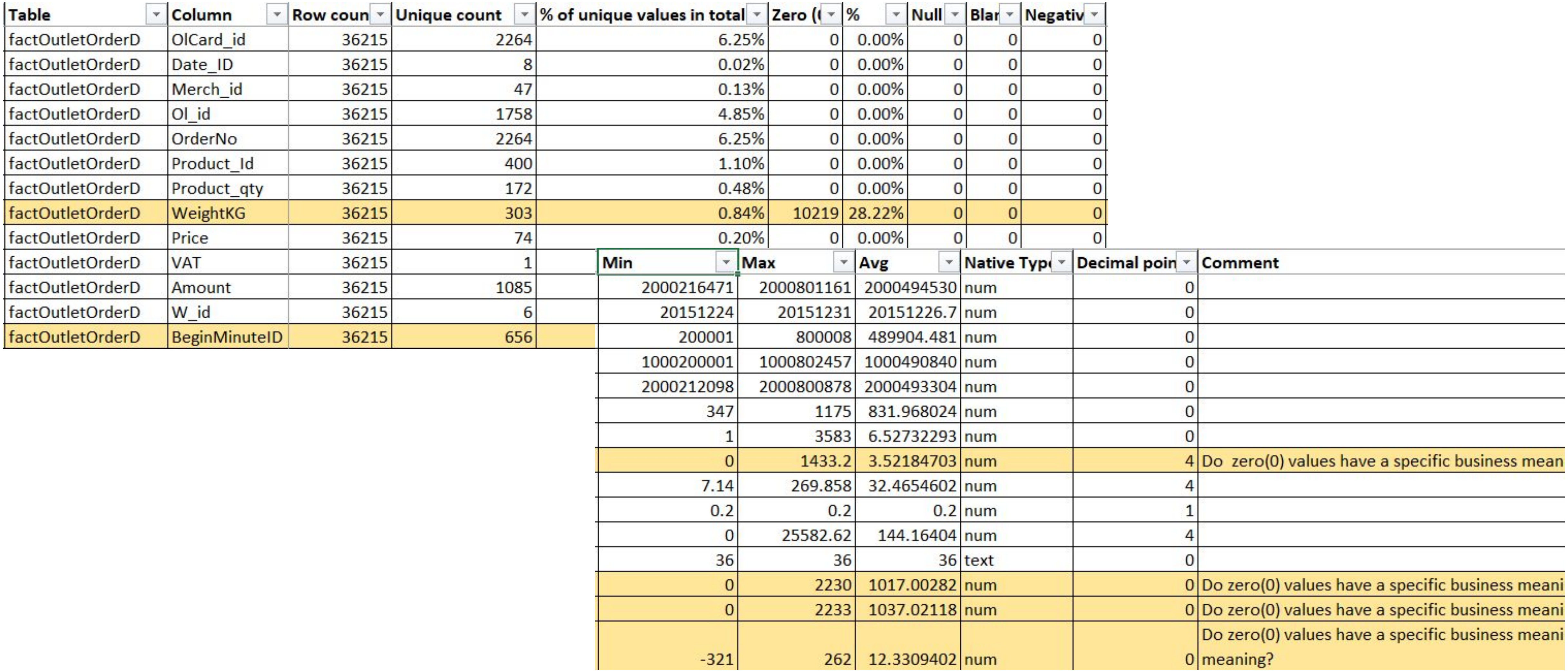


Table	Column	Row coun	Unique count	% of unique values in total	Zero ((	%	Null	Blar	Negativ
factOutletOrderD	OlCard_id	36215	2264	6.25%	0	0.00%	0	0	0
factOutletOrderD	Date_ID	36215	8	0.02%	0	0.00%	0	0	0
factOutletOrderD	Merch_id	36215	47	0.13%	0	0.00%	0	0	0
factOutletOrderD	Ol_id	36215	1758	4.85%	0	0.00%	0	0	0
factOutletOrderD	OrderNo	36215	2264	6.25%	0	0.00%	0	0	0
factOutletOrderD	Product_Id	36215	400	1.10%	0	0.00%	0	0	0
factOutletOrderD	Product_qty	36215	172	0.48%	0	0.00%	0	0	0
factOutletOrderD	WeightKG	36215	303	0.84%	10219	28.22%	0	0	0
factOutletOrderD	Price	36215	74	0.20%	0	0.00%	0	0	0
factOutletOrderD	VAT	36215	1						
factOutletOrderD	Amount	36215	1085						
factOutletOrderD	W_id	36215	6						
factOutletOrderD	BeginMinuteID	36215	656						

Min	Max	Avg	Native Typ	Decimal poin	Comment
2000216471	2000801161	2000494530	num	0	
20151224	20151231	20151226.7	num	0	
200001	800008	489904.481	num	0	
1000200001	1000802457	1000490840	num	0	
2000212098	2000800878	2000493304	num	0	
347	1175	831.968024	num	0	
1	3583	6.52732293	num	0	
0	1433.2	3.52184703	num	4	Do zero(0) values have a specific business mean
7.14	269.858	32.4654602	num	4	
0.2	0.2	0.2	num	1	
0	25582.62	144.16404	num	4	
36	36	36	text	0	
0	2230	1017.00282	num	0	Do zero(0) values have a specific business meani
0	2233	1037.02118	num	0	Do zero(0) values have a specific business meani
-321	262	12.3309402	num	0	Do zero(0) values have a specific business meani meaning?

SOURCE-LANDING DATA CHECK WITH DATA PROFLING

- MIN, MAX, AVG… numeric values check

SOURCE

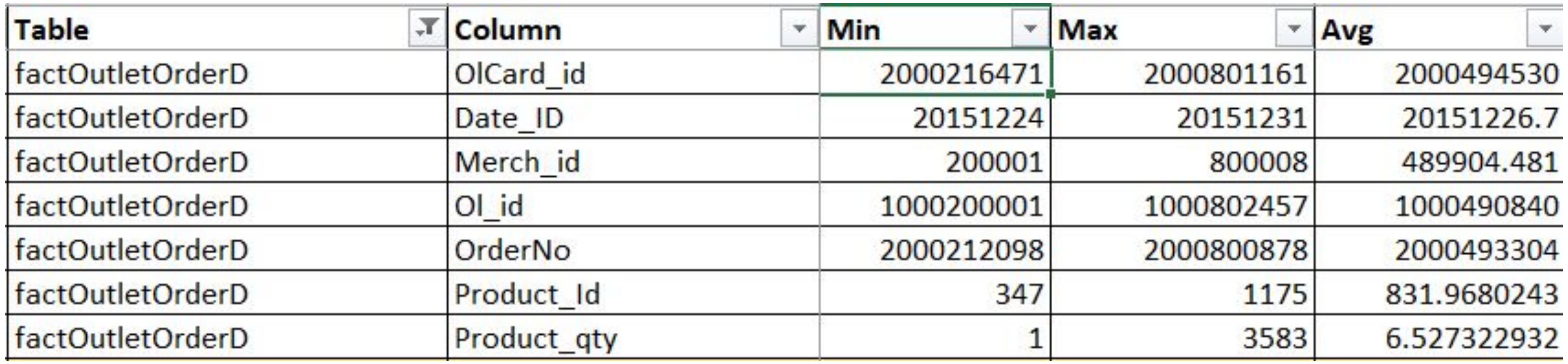


Table	Column	Min	Max	Avg
factOutletOrderD	OlCard_id	2000216471	2000801161	2000494530
factOutletOrderD	Date_ID	20151224	20151231	20151226.7
factOutletOrderD	Merch_id	200001	800008	489904.481
factOutletOrderD	Ol_id	1000200001	1000802457	1000490840
factOutletOrderD	OrderNo	2000212098	2000800878	2000493304
factOutletOrderD	Product_Id	347	1175	831.9680243
factOutletOrderD	Product_qty	1	3583	6.527322932

STAGING

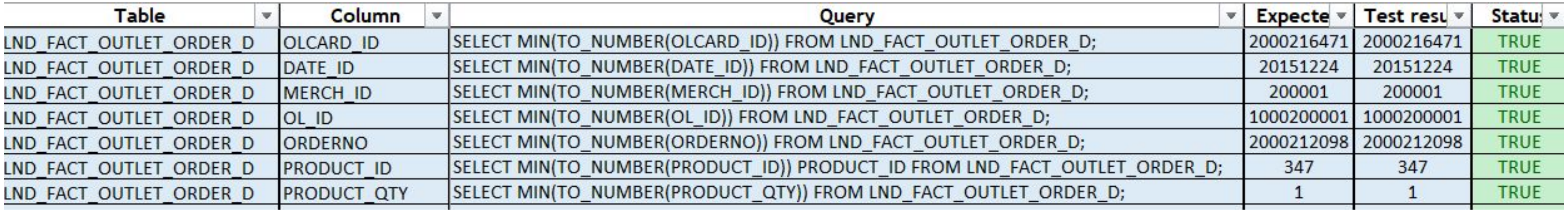


Table	Column	Query	Expecte	Test resu	Statu:
LND_FACT_OUTLET_ORDER_D	OLCARD_ID	SELECT MIN(TO_NUMBER(OLCARD_ID)) FROM LND_FACT_OUTLET_ORDER_D;	2000216471	2000216471	TRUE
LND_FACT_OUTLET_ORDER_D	DATE_ID	SELECT MIN(TO_NUMBER(DATE_ID)) FROM LND_FACT_OUTLET_ORDER_D;	20151224	20151224	TRUE
LND_FACT_OUTLET_ORDER_D	MERCH_ID	SELECT MIN(TO_NUMBER(MERCH_ID)) FROM LND_FACT_OUTLET_ORDER_D;	200001	200001	TRUE
LND_FACT_OUTLET_ORDER_D	OL_ID	SELECT MIN(TO_NUMBER(OL_ID)) FROM LND_FACT_OUTLET_ORDER_D;	1000200001	1000200001	TRUE
LND_FACT_OUTLET_ORDER_D	ORDERNO	SELECT MIN(TO_NUMBER(ORDERNO)) FROM LND_FACT_OUTLET_ORDER_D;	2000212098	2000212098	TRUE
LND_FACT_OUTLET_ORDER_D	PRODUCT_ID	SELECT MIN(TO_NUMBER(PRODUCT_ID)) PRODUCT_ID FROM LND_FACT_OUTLET_ORDER_D;	347	347	TRUE
LND_FACT_OUTLET_ORDER_D	PRODUCT_QTY	SELECT MIN(TO_NUMBER(PRODUCT_QTY)) FROM LND_FACT_OUTLET_ORDER_D;	1	1	TRUE

MAKING THE TEST ENVIRONMENT DECISION

A testing environment is a setup of software and hardware for the testing teams to execute test cases

- Do you need a separate QA env?
- How many environments do you really need?
- What is specific of these environments?
- Is it possible to satisfy your request?
- Working closely with DevOps team

MAKING THE TEST DATA DECISION

	What is it?	When we need it?	Advantages	Disadvantages
Synthetic data	Data that is artificially created rather than being generated by actual events.	To protect customer information Required data does not exist Required data has some gaps No access to prod data	✔ More efficient and cost effective ✔ Cover missing cases in real data/ specific cases/ conditions ✔ Increased flexibility ✔ You are the only owner of your data ✔ No secure risks ✔ Using less data	✔ If the system is complex it is a challenge to create high quality synthetic data
Production data	A subset of production data to represent a portion of the database that is relevant to a test case	Complicated logic and dependencies Historical data required Performance testing	✔ High quality software in case of complex systems and dependencies ✔ Ability to quickly reproduce client's issue	✔ Security violation: risk of exposing sensitive user data ✔ Email addresses, phone numbers, and the like can be accidentally reach users by integration tests ✔ data is changing all the time, so it's more difficult to write stable assertions
Production like data	Snapshot of production that has been masked or obfuscated	Only production sensitive data can cover requirements	✔ Same as production data	✔ legal or regulatory requirements mandate anonymizing PII, patient data, financials, and so on, which requires extra effort
Test data	End to end data created by test team in full integration environment	No access to UAT		✔ Extra efforts to create test data

MAIN PROCESSES IN DWH TESTING

- **Data Extraction** – the data in the warehouse can come from many sources and of multiple data format and types with may be incompatible from system to system. The process of data extraction includes formatting the disparate data types into one type understood by the warehouse. The process also includes compressing the data and handling of encryptions whenever this applies;

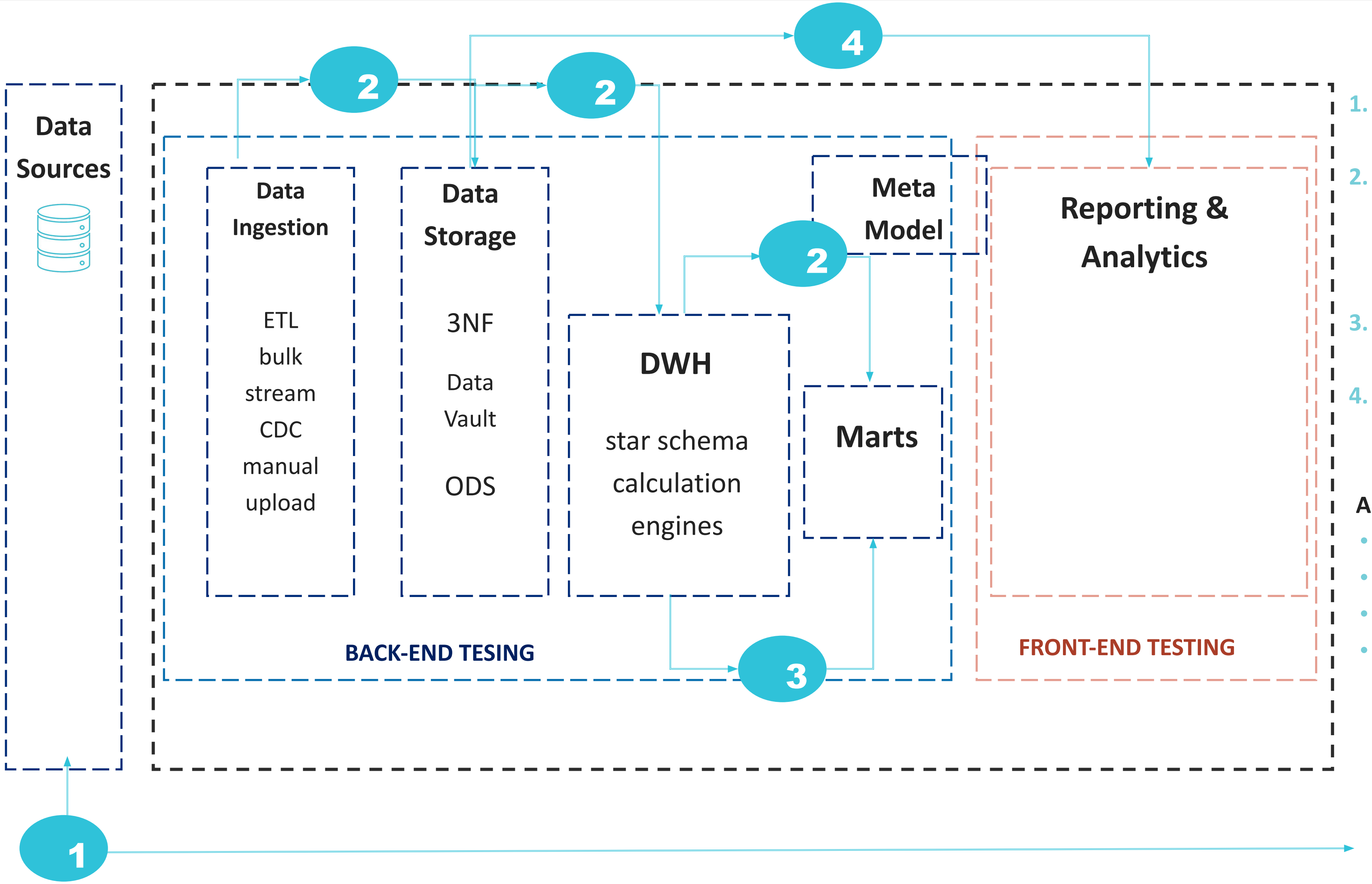
- **Data Transformation** – this processes include data integration, denormalization, surrogate key management, data cleansing, conversion, auditing and aggregation;

- **Data Loading** – after the first two process, the data will then be ready to be optimally stored in the data warehouse;

- **Security Implementation** – data should be protected from prying eyes whenever applicable as in the case of bank records and credit card numbers. The data warehouse administrator implements access and data encryption policies;

- **Job Control** – this process is the constant job of the data warehouse administrator and his staff. This includes job definition, time and event job scheduling, logging, monitoring, error handling, exception handling and notification.

DWH TESTING



1. **Metadata checks + data formats, Profiling, LLD**
2. **Data**: COUNT, MIN, MAX, AVG, SUM, row-by-row comparison (subset)
   Integrity: COUNT DISTINCT, NOT NULL, Hash, data ranges
3. **Reconciliation**: Row-by-row comparison
4. **Reports data testing**

Additional checks:

- Performance
- Security
- Error handling
- ETL run times (errors, warnings)

MAIN FUNCTIONAL VALIDATIONS

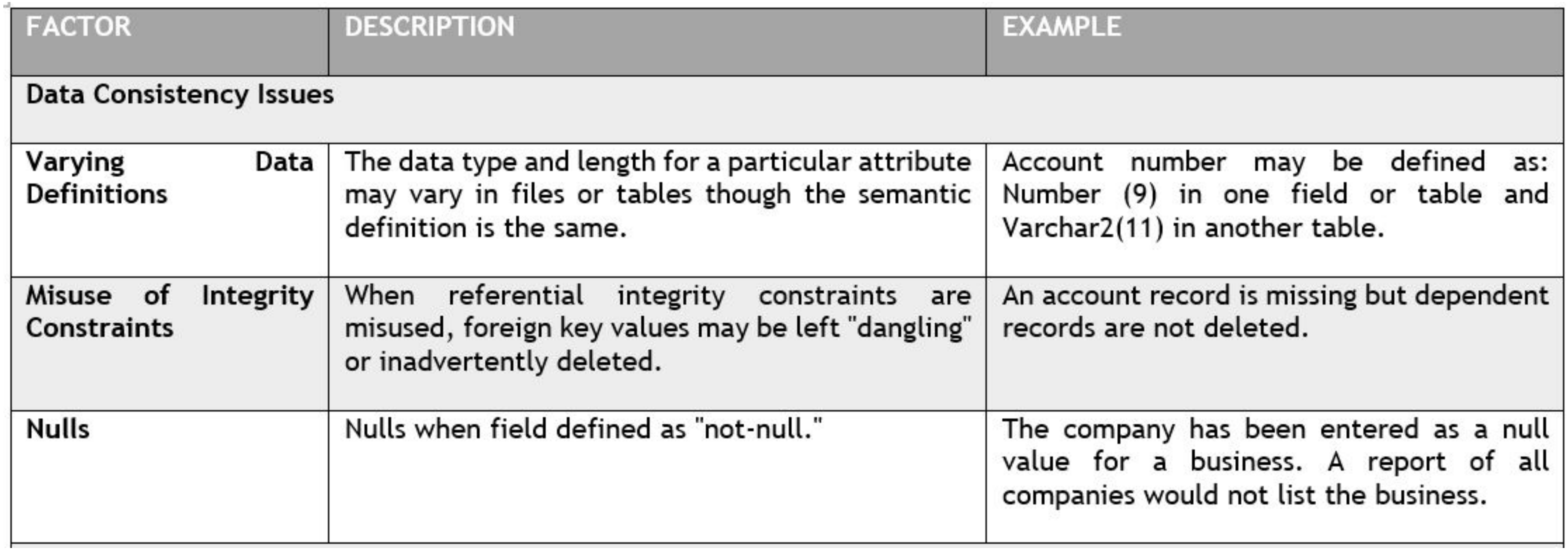
Standard Validation

- Profiling /LLD/ Data Validation
- Counts, Checksum Validation
- End to End testing

Business Validation

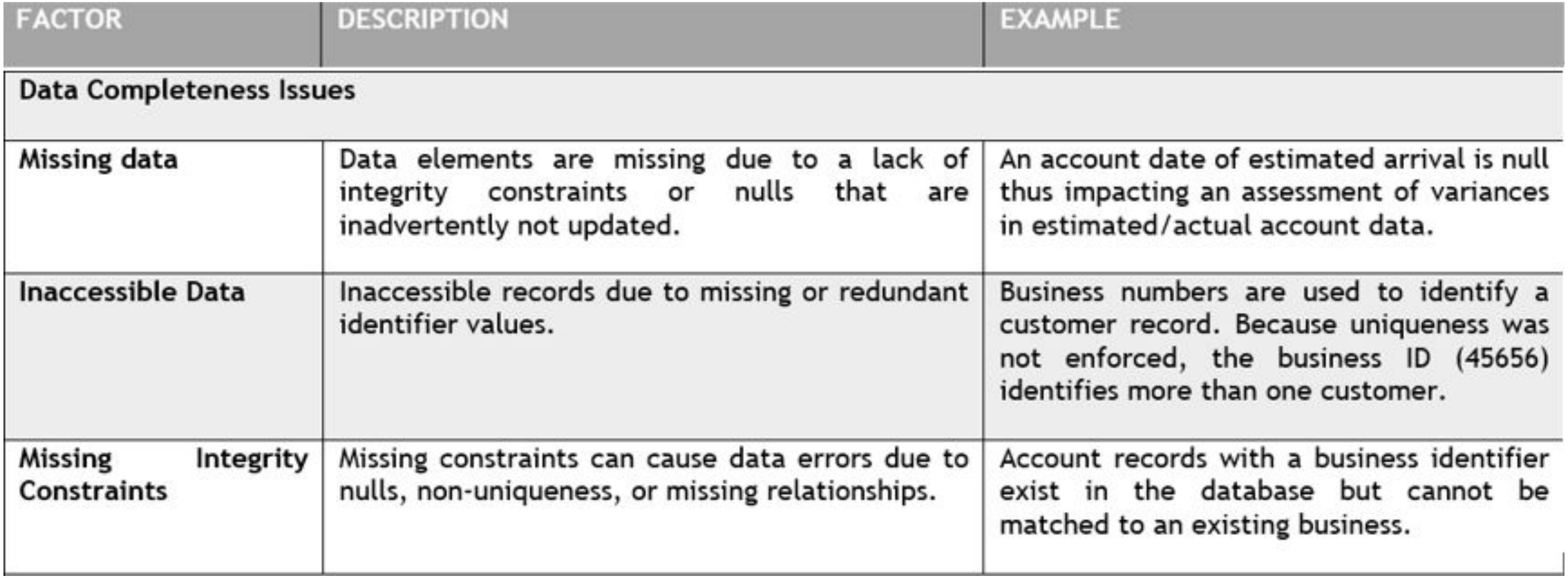
- Straight/Direct move
- Data transformation
- Look up validation
- Filtering
- Average Balance Calculation
- Data integrity validation
- External field validation

SOME EXAMPLES



FACTOR	DESCRIPTION	EXAMPLE
Data Consistency Issues		
Varying Data Definitions	The data type and length for a particular attribute may vary in files or tables though the semantic definition is the same.	Account number may be defined as: Number (9) in one field or table and Varchar2(11) in another table.
Misuse of Integrity Constraints	When referential integrity constraints are misused, foreign key values may be left "dangling" or inadvertently deleted.	An account record is missing but dependent records are not deleted.
Nulls	Nulls when field defined as "not-null."	The company has been entered as a null value for a business. A report of all companies would not list the business.

SOME EXAMPLES

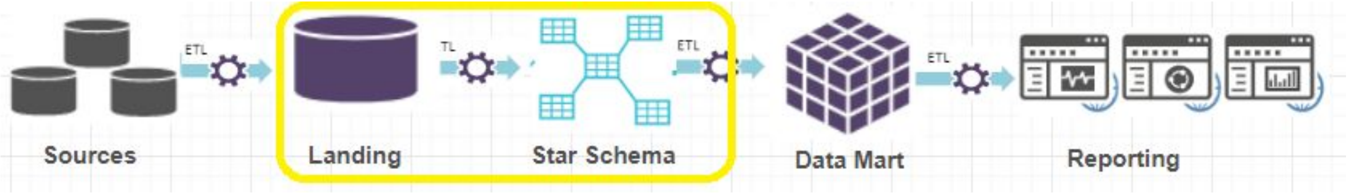


FACTOR	DESCRIPTION	EXAMPLE
Data Completeness Issues		
Missing data	Data elements are missing due to a lack of integrity constraints or nulls that are inadvertently not updated.	An account date of estimated arrival is null thus impacting an assessment of variances in estimated/actual account data.
Inaccessible Data	Inaccessible records due to missing or redundant identifier values.	Business numbers are used to identify a customer record. Because uniqueness was not enforced, the business ID (45656) identifies more than one customer.
Missing Integrity Constraints	Missing constraints can cause data errors due to nulls, non-uniqueness, or missing relationships.	Account records with a business identifier exist in the database but cannot be matched to an existing business.

VERIFY CLEANED DATA

Verify corrected, cleaned, merged data

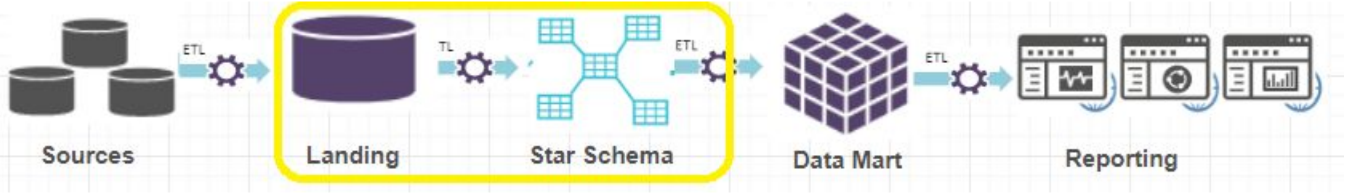
- verify cleansing rules (check error tables, rejected records)
- verify data merge, lookups
- verify data integrity (check for duplicates, orphaned data)
- verify data for renaming/reformatting
- verify data transformations



VERIFY CONSOLIDATED DATA

Verify matched and consolidated data

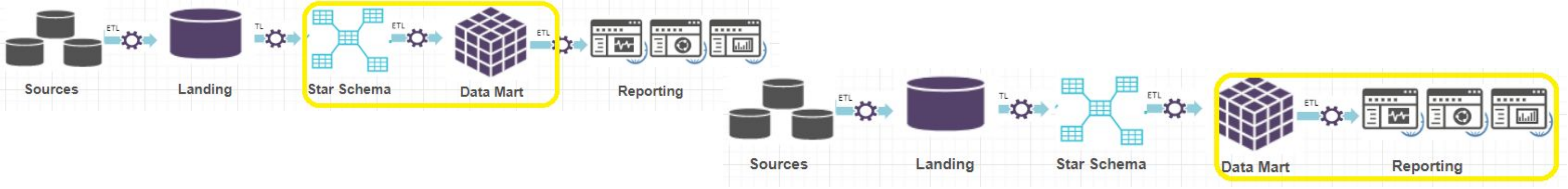
- verify pivoting or loading data
- verify data completeness, quality
- verify joining data from multiple sources (e.g., lookup, merge)



VERIFY DATA ON REPORTS LEVEL

Verify transformed/enhanced/calculated data

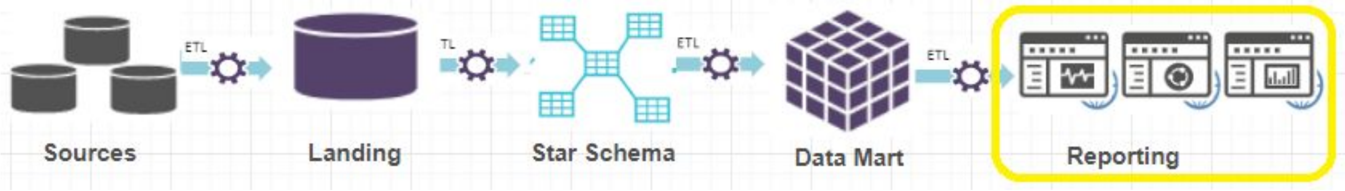
- verify sorting, pivoting, computing subtotals, adding view filters, etc. (Reporting)
- verify that dimension and fact tables mapped correctly, therefore SQL generated correctly (DM-Reporting)
- validate calculation logic against business requirements (write SQL for data mart using calculation rules and compare data set (DM-Reporting)



FRONT-END VERIFICATION

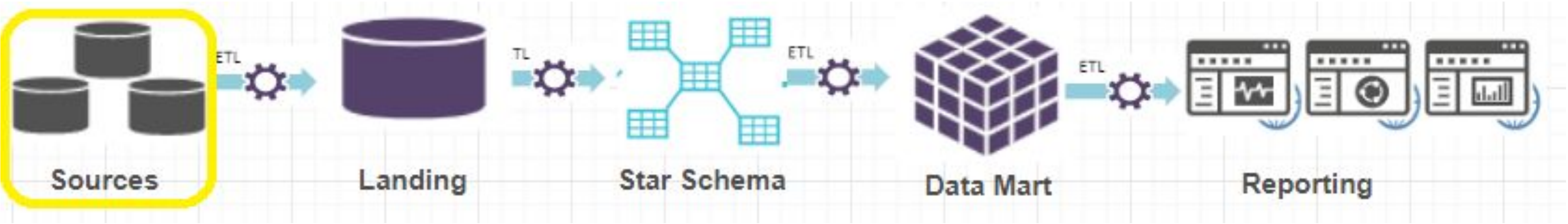
Verify front-end data

- verify main functionality (export, scheduling, filters, etc.)
- verify data on UI
- verify presentation
- verify performance (speed, availability, response time, recovery time, etc.);



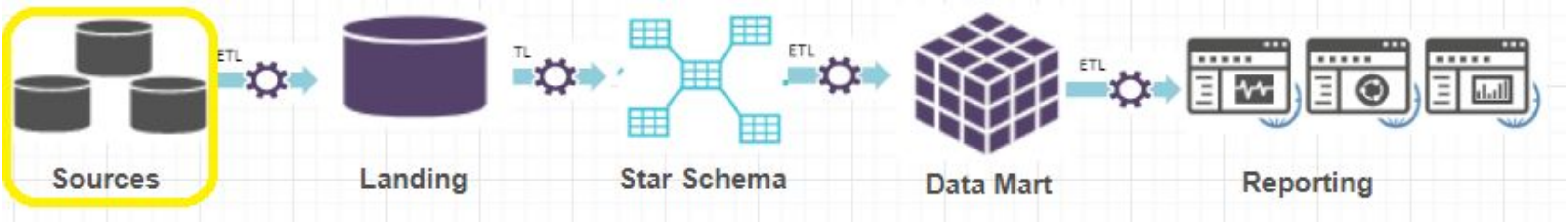
TYPICAL DATA ISSUES

DATA SOURCE LEVEL



DATA SOURCE - TYPICAL DATA ISSUES

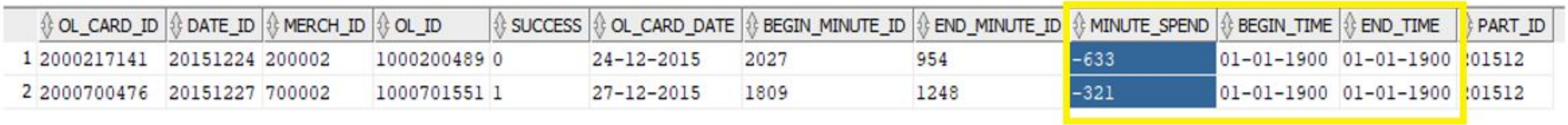
- Inappropriate selection of candidate data sources
- Unanticipated changes in source application
- Conflicting information present in data sources
- Inappropriate data entity relationships among tables
- Different data types for similar columns (for example, addresslD is stored as a number in one table and a string in another)
- Different data representation (The day of the week is stored as M, or Mon, and Monday in other separate columns)



DATA SOURCE - ISSUE EXAMPLE

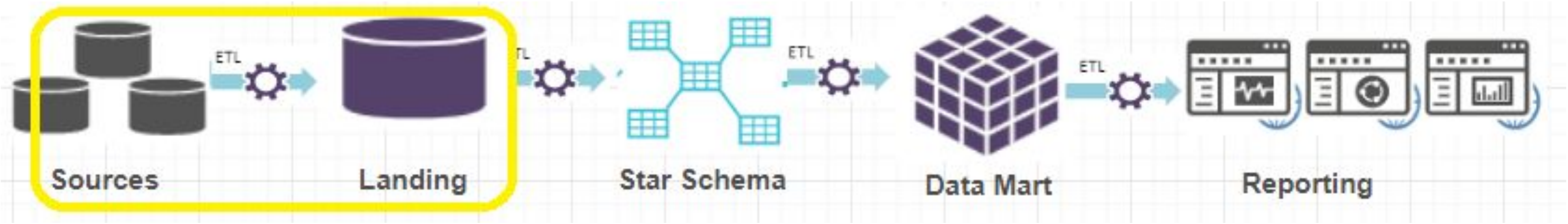
All values in columns BEGIN_TIME and END_TIME are '01-01-1900'. Is it correct? If yes, how It should be interpreted?

There are records with negative value in MINUTE_SPEND column. Screenshot below:



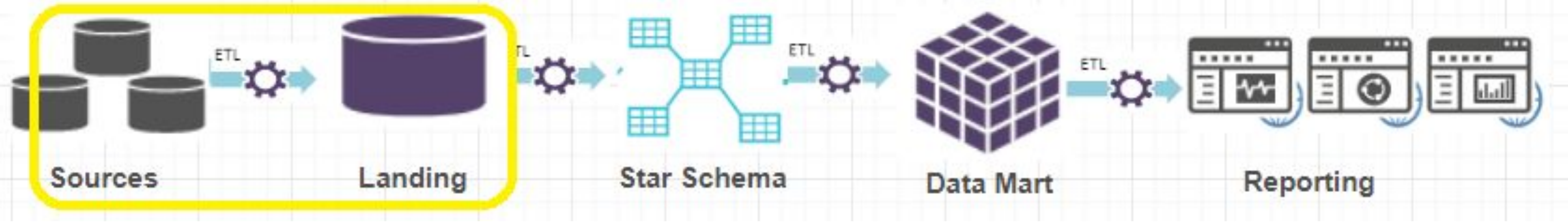
	OL_CARD_ID	DATE_ID	MERCH_ID	OL_ID	SUCCESS	OL_CARD_DATE	BEGIN_MINUTE_ID	END_MINUTE_ID	MINUTE_SPEND	BEGIN_TIME	END_TIME	PART_ID
1	2000217141	20151224	200002	1000200489	0	24-12-2015	2027	954	-633	01-01-1900	01-01-1900	01512
2	2000700476	20151227	700002	1000701551	1	27-12-2015	1809	1248	-321	01-01-1900	01-01-1900	01512

SOURCE - LANDING LEVEL

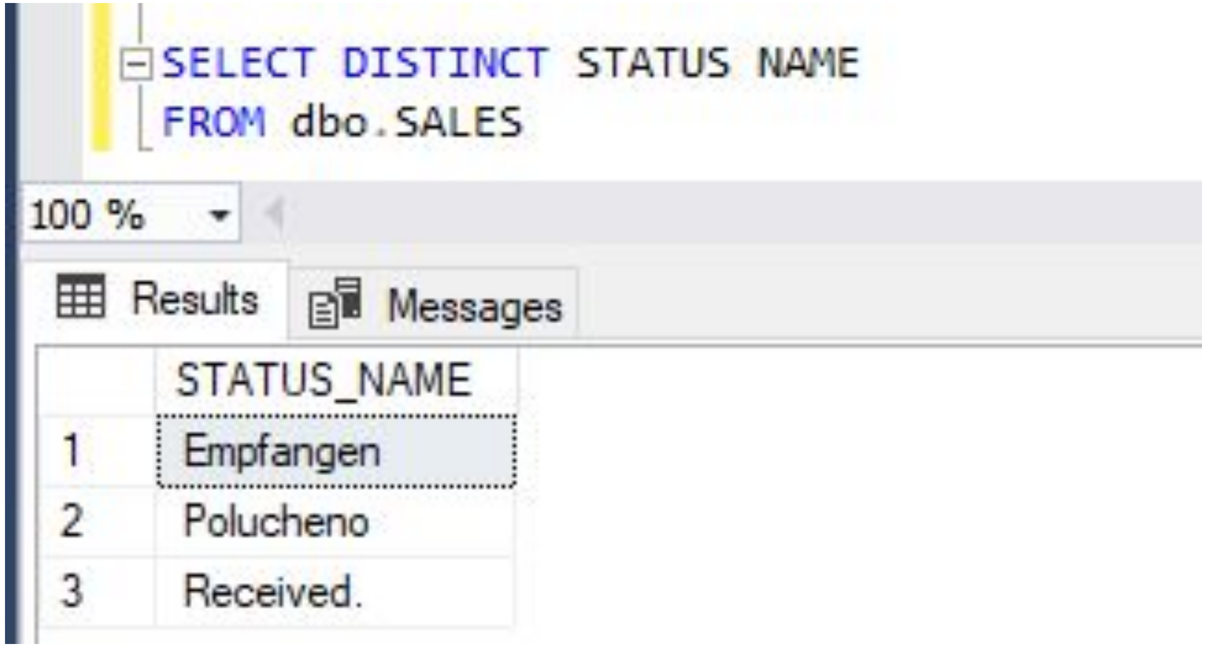


SOURCE-LANDING - TYPICAL DATA ISSUES

- Different data formats, column names
- Some data can be missed or corrupted while capturing from data sources
- Data comes in real-time
- Performance - incremental and initial download



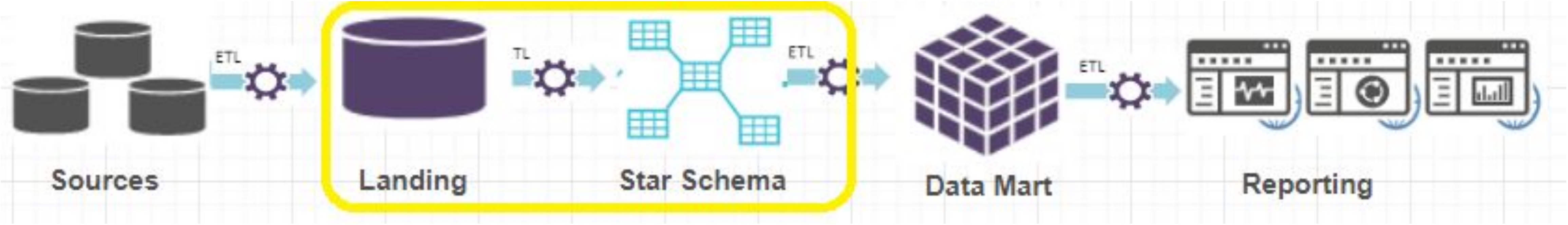
SOURCE-LANDING - DATA ISSUE EXAMPLE



```sql
SELECT DISTINCT STATUS NAME
FROM dbo.SALES
```

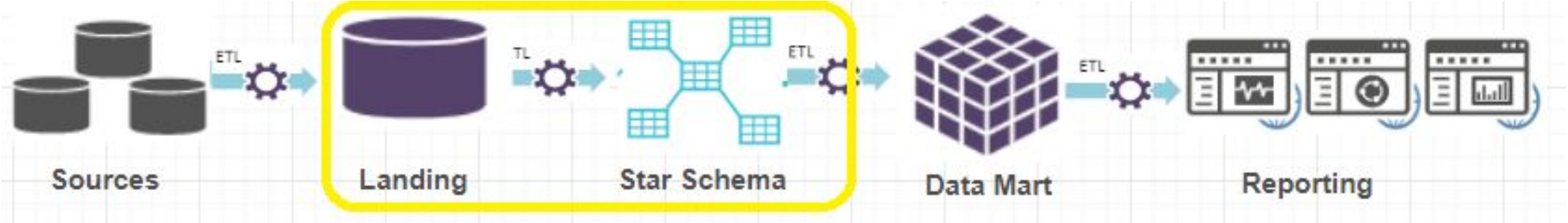
	STATUS_NAME
1	Empfangen
2	Polucheno
3	Received.

LND - DWH LEVEL

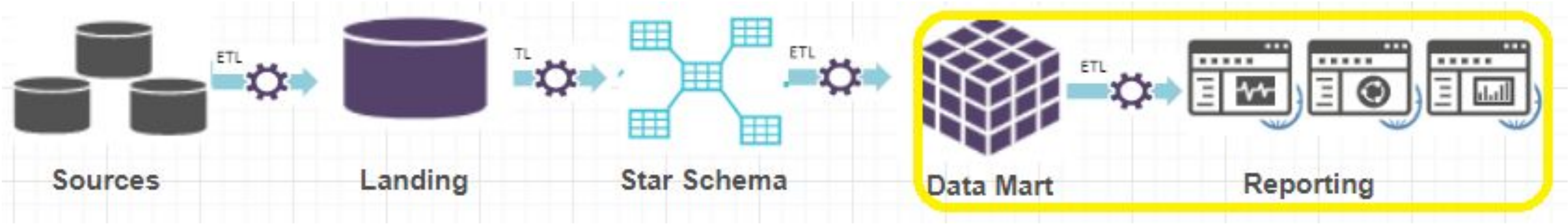


DWH - TYPICAL DATA ISSUES

- Incorrect **business rules** for data consolidation and merging: data inconsistency and data incompleteness
- Loss of data during the ETL process (rejected records, refused data records in the ETL process)
- Missed errors

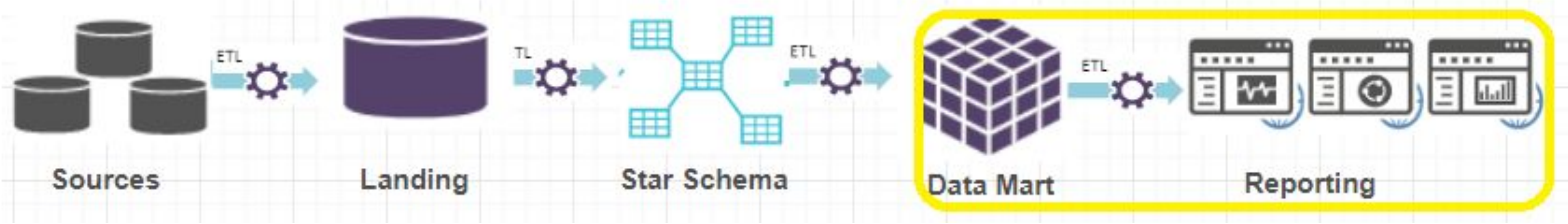


DM LEVEL

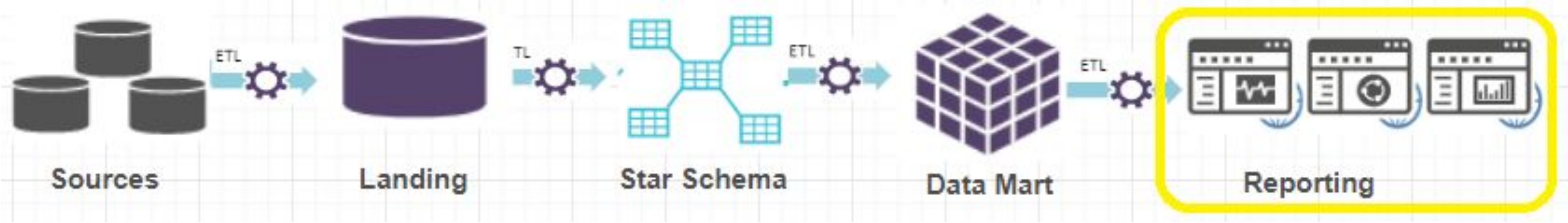

ETL
TL
ETL
ETL
Sources
Landing
Star Schema
Data Mart
Reporting

DATA MART - TYPICAL DATA ISSUES

- Errors in aggregation, calculation logic
- Incorrect data filtering

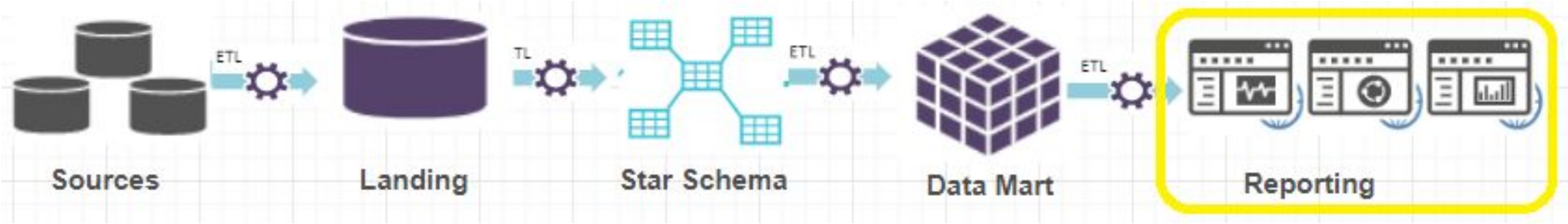


REPORT LEVEL



DATA ANALYSIS LAYER – TYPICAL DATA ISSUES

- Dimension and fact tables mapped incorrectly, therefore SQL generated incorrectly
- Incorrect calculation of subtotals (especially if derived metrics used), KPIs, metrics, etc.
- Incorrect behavior of some report manipulation techniques (drilling, sorting, export functions, etc.)
- Performance issues (speed, availability, response time, recovery time, etc.)



<epam>

Q & A