# ETL Quick Dive

A Guide by Ivan Sokolov with 💖

# CONTENTS

Hey there!
I'm Ivan.
I like automation, fixing data discrepancies, and piano music

**Tbilisi**
Current location

**3,5 years**
Overall experience

**SQL**
Main skill

**FMCG**
**INSURTECH**
Domain knowledge

Atria PERHETILOILTA VUODESTA 1903 → Symfa™

| **2019-2021** | **2021-now** |
|---|---|
| Forecasting | BI – Enterprise Reporting |
| Business Intelligence (BI) | ETL – International DWH |
| ETL | |
| SQL (SSMS) | SQL (SSMS, SSRS, SSIS) |
| IBM SPSS Modeler | Astera Centerprise |
| Python | Git |

ETL stands for ‘**Extract, Transform, Load**’

This technology is very popular among **large** enterprises

If you **collect** data from **different** counteragents, you need ETL

## Extract

DataBase

File / Document

ERP

## Transform

Cleaning / Grooming

Validation

Aggregation

Merging / Appending

Normalization

## Load

Data Warehouse (DWH)

Folder

Portal / Website

Cloud

Data Engineers, ETL Developers

## BI

## ML

BI Devs, Data Analysts, ML Devs

## The most popular ETL tools

Informatica

talend

hadoop

Apache Airflow

IBM DataStage

ORACLE DATA INTEGRATOR

Microsoft SQL Server

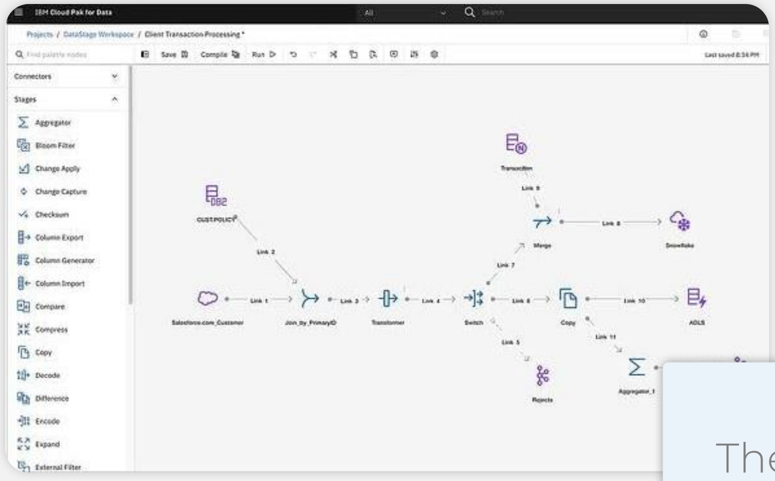
Azure Data Factory

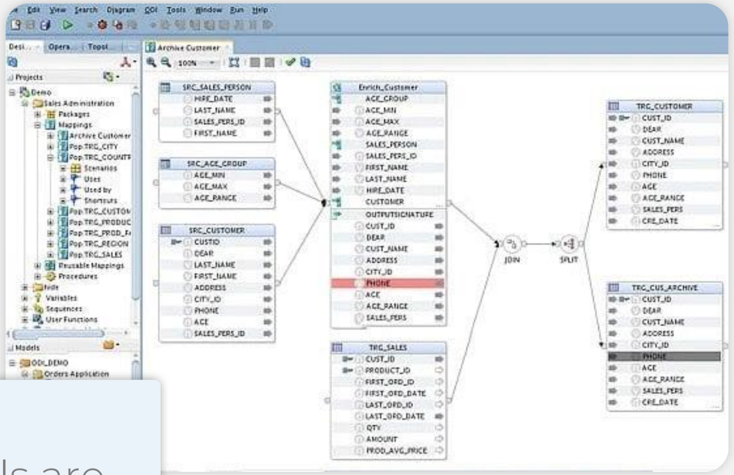
SAP Data Services

HEVO

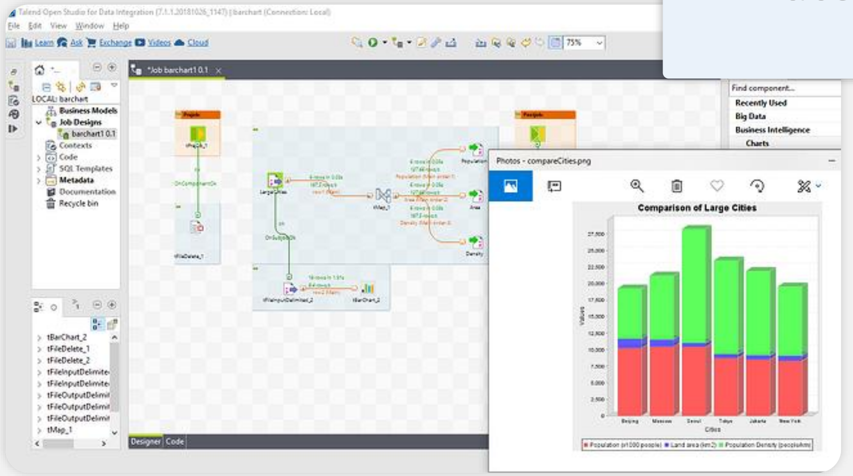
pentaho

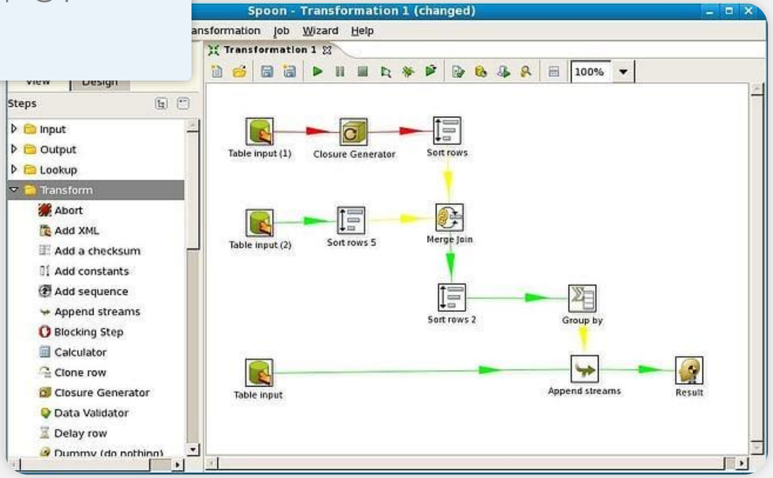
Qlik

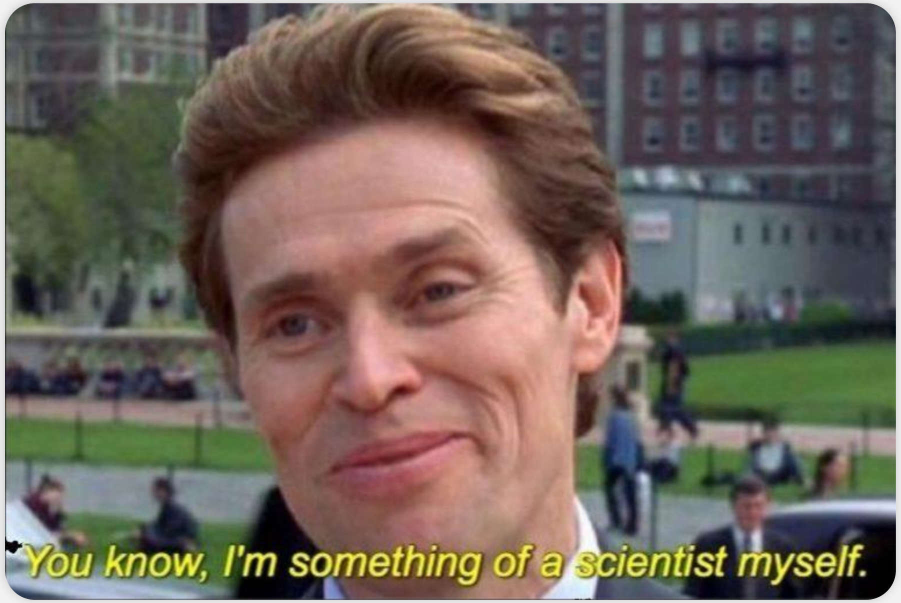




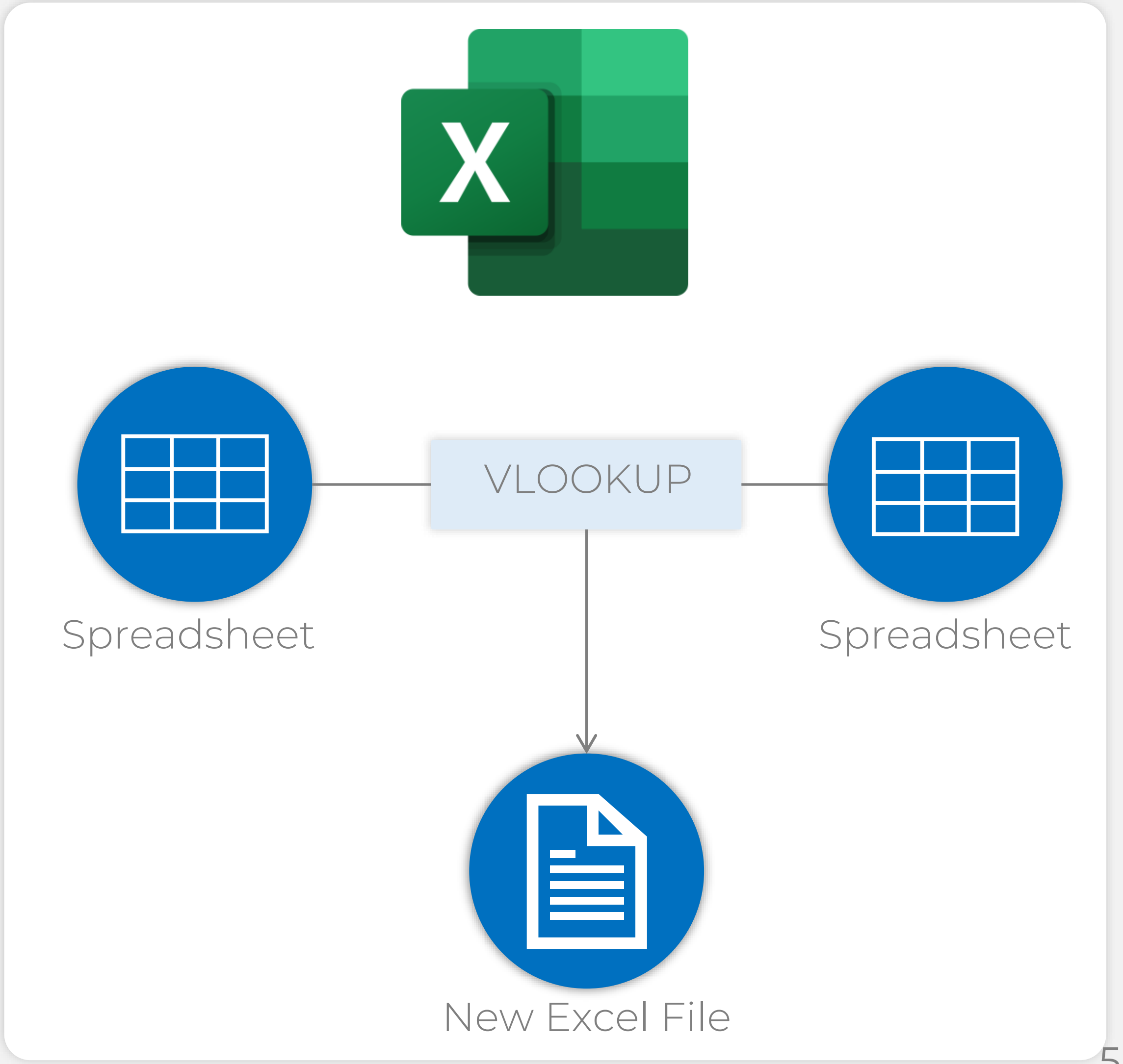
The major part of ETL tools are designed with similar UI


Talend Open Studio for Data Integration
Job barchart1 0.1
Photos - compareCities.png
Comparison of Large Cities





If you have ever merged / appended several Excel spreadsheets / built pivot tables and saved it to a new file, you already have some ETL experience



A **good** ETL Dev is a guy who:

- Understands and sees his data on the **granular** level
- Knows how to **manipulate** the data
- Understands what is going on behind **every step** of his flows

Create flows

Automate flows

Schedule flows

Update flows

## Required Skillset

**Tools / Software**
Microsoft, Amazon, IBM, Apache or any other tool

**SQL**
Querying the data

**Parametrization**
Dynamically change ETL aspects

**Scripting Language**
Juggle files, directories, users, and permission issues

**Organization**
Keep the work at hand organized and structured

**Creativity**
Source To Target Mapping

**Debugging**
Fix the broken

UX / UI

Functionality

Interface

**Application**

API

System Components

Server / DWH / Data

ETL is here!

ETL Development is **different** from other development types and has its **own features**

## Functional Development

**Software, app, web**
object of development

**CI/CD, Containers**
main tool / approach

**PL**
Ruby, JS, Python, Swift, C++, C, C#, Perl, R, SQL

**Coding Paradigm**
OOP, FP

## ETL Development

**Data**
object of development

**ETL Pipeline**
main tool / approach

**PL**
**SQL**, Python, R

**SQL Paradigm**
Declarative language

## Spot Check

Simple and fast queries to check the data at random places

```
SELECT TOP <N>
<…>
FROM <Table>
WHERE <...>
```

*Quick but not reliable*

## Integrity Check

Heavy queries to check the relations between the tables of a database

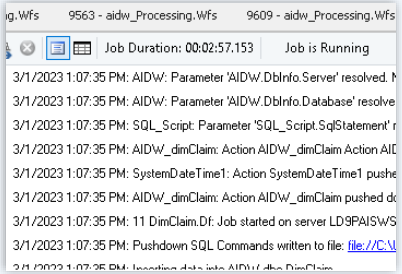
```
SELECT *
FROM <DimTable>
WHERE <Key> = -1

SELECT *
FROM <FactTable>
WHERE <Key> = -1
```

*Reliable but long*

## Empirical Test

Empirical analysis of the way the ETL works

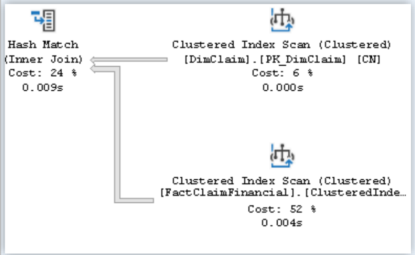


```
g.Wfs    9563 - aidw_Processing.Wfs    9609 - aidw_Processing.Wfs
Job Duration: 00:02:57.153    Job is Running
3/1/2023 1:07:35 PM: AIDW: Parameter 'AIDW.DbInfo.Server' resolved. N
3/1/2023 1:07:35 PM: AIDW: Parameter 'AIDW.DbInfo.Database' resolve
3/1/2023 1:07:35 PM: SQL_Script: Parameter 'SQL_Script.SqlStatement' r
3/1/2023 1:07:35 PM: AIDW_dimClaim: Action AIDW_dimClaim Action AID
3/1/2023 1:07:35 PM: SystemDateTime1: Action SystemDateTime1 pushe
3/1/2023 1:07:35 PM: AIDW_dimClaim: Action AIDW_dimClaim pushed do
3/1/2023 1:07:35 PM: 11 DimClaim.Df: Job started on server LD9PAISWS
3/1/2023 1:07:35 PM: Pushdown SQL Commands written to file: file://C:\U
3/1/2023 1:07:35 PM: Inserting data into AIDW.dbo.DimClaim
```

*Semi-reliable*

## Execution Plan

SSMS feature that graphically represents operations' execution. Helps identify bottlenecks


Hash Match
(Inner Join)
Cost: 24 %
0.009s
Clustered Index Scan (Clustered)
[DimClaim].[PK_DimClaim] [CN]
Cost: 6 %
0.000s
Clustered Index Scan (Clustered)
[FactClaimFinancial].[ClusteredInde…
Cost: 52 %
0.004s

*Powerful but needs rights access*

## Source-Destination Check (SDC)

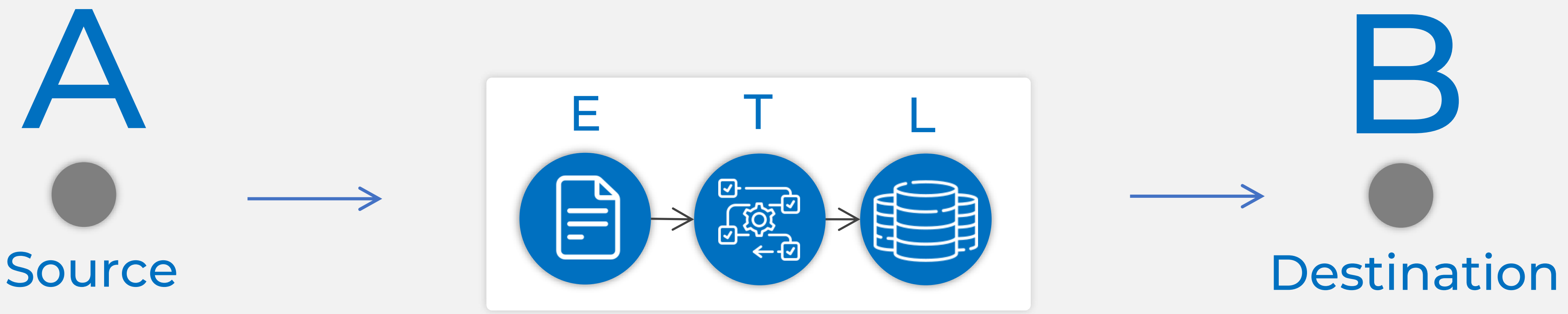
SQL queries to check destination tables against their source by a particular criteria

```
SELECT
,SUM(<Field>)
,COUNT(*)
FROM <Destination>

SELECT
,SUM(<Field>)
,COUNT(*)
FROM <Source>
```

*Powerful and my favorite*

## Testing methods in a general dataflow

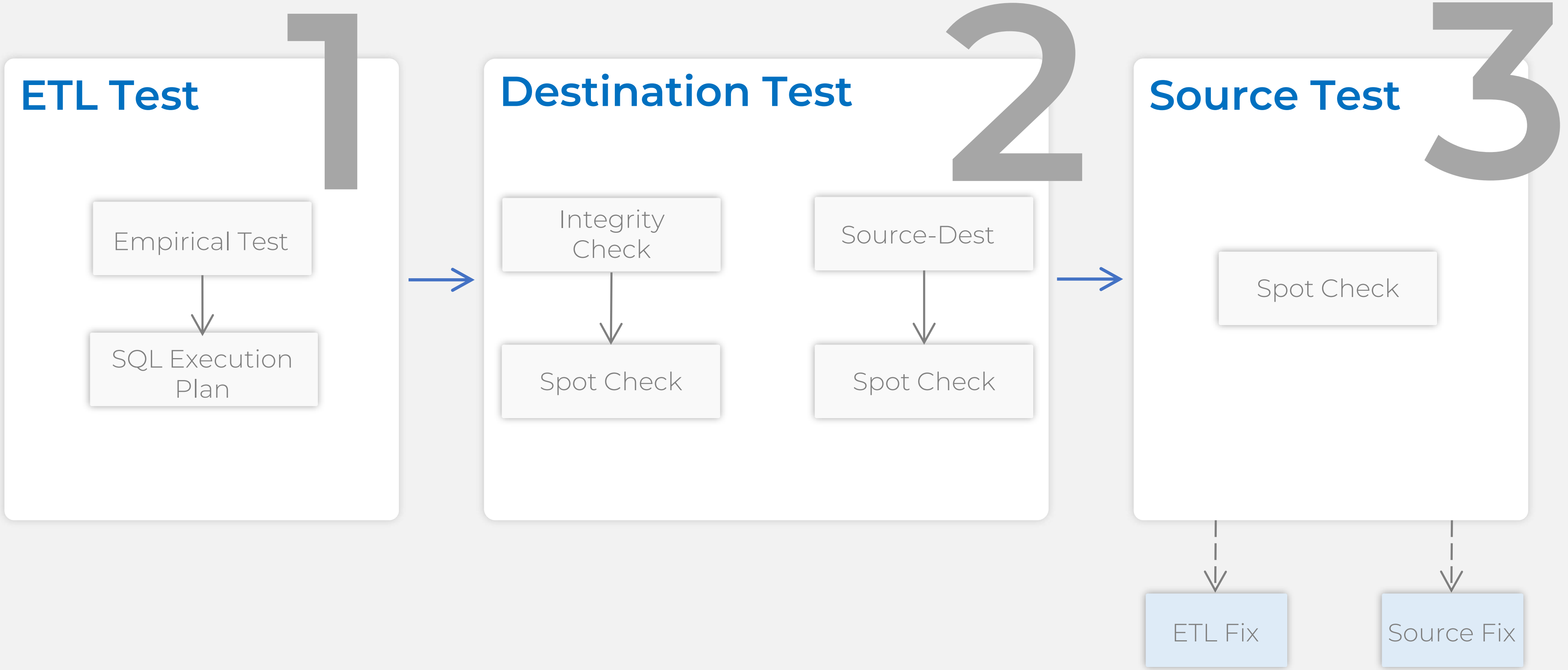


**Source**
- Spot Check
- Integrity check

**ETL**
- Empirical Test
- SQL Execution Plan

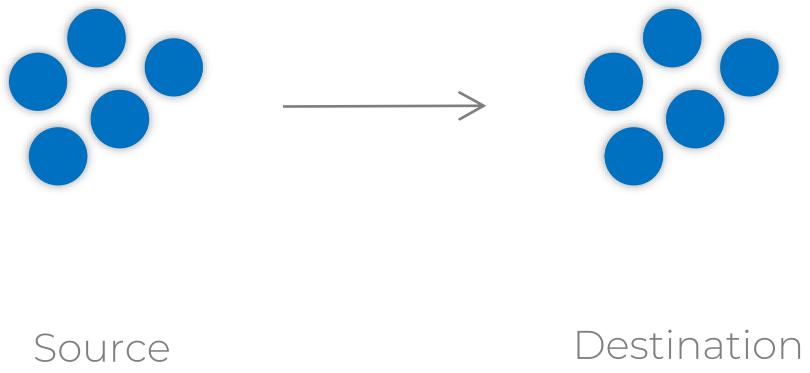
**Destination**
- Spot Check
- Integrity Check
- Source-Dest Check

## The workflow of applying testing methods

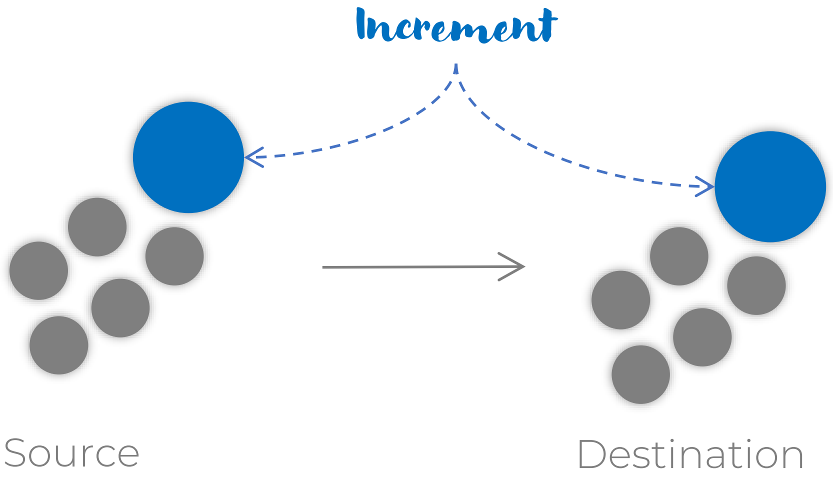


When dealing with **large historical** data, at some point every ETL should use **incremental** loads

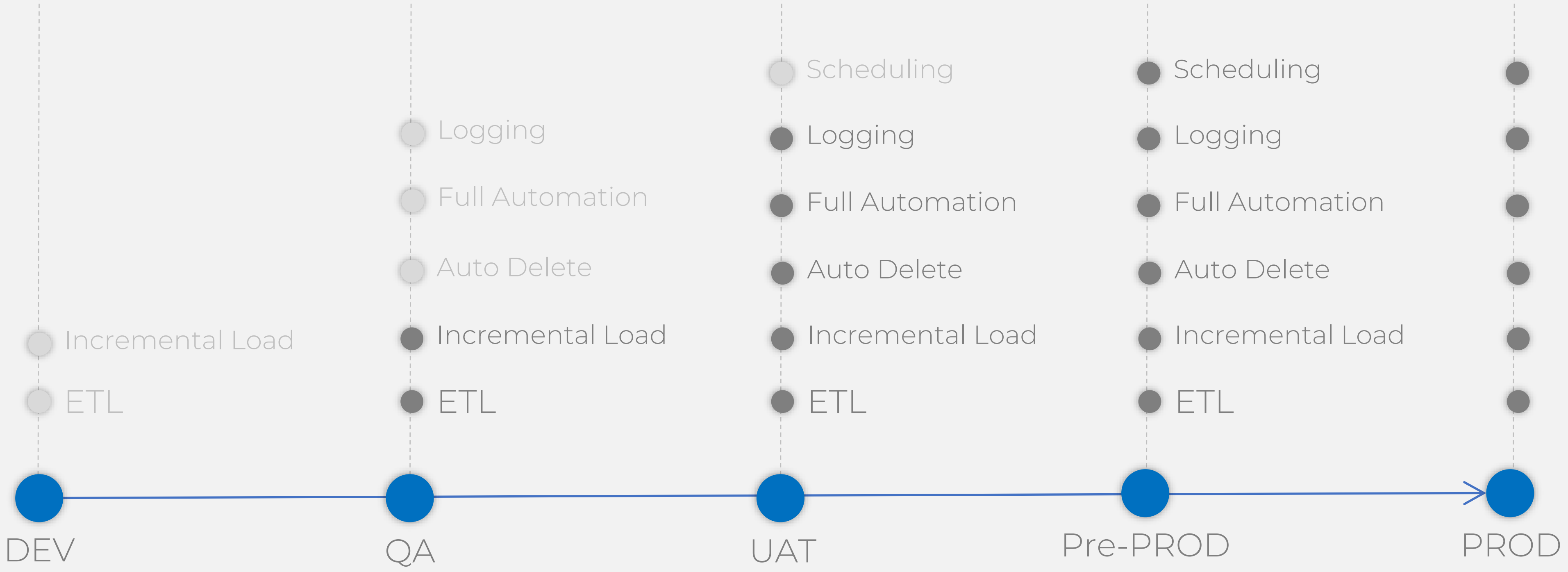
## Time Period 1



## Time Period 2



The closer you get to **Production**, the more **features** should be implemented and tested


Scheduling
Scheduling
Logging
Logging
Logging
Full Automation
Full Automation
Full Automation
Auto Delete
Auto Delete
Auto Delete
Incremental Load
Incremental Load
Incremental Load
Incremental Load
ETL
ETL
ETL
ETL
DEV
QA
UAT
Pre-PROD
PROD