

ETL vs. ELT

A Data Architect's Guide to Modern Data Integration



ProtoComet

The Paradigm Shift

1990s - 2010s

ETL Era

- Expensive on-prem warehouses
- Limited compute resources
- Transform before loading
- Minimize warehouse burden

2010s - Present

ELT Era

- Cloud-native warehouses
- Elastic, unlimited compute
- Load raw, transform in-DB
- Leverage MPP architecture



What is ETL?

Extract → Transform → Load

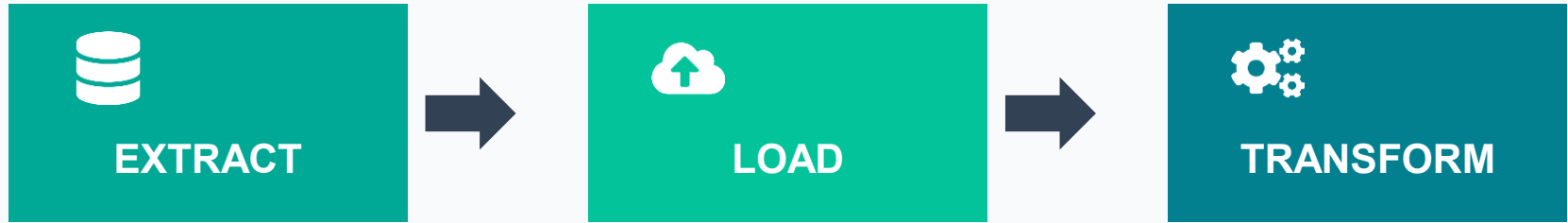


Data is transformed in external ETL servers before entering the warehouse



What is ELT?

Extract → Load → Transform



Raw data lands in the warehouse first, transformations happen in-database



Key Differences

Dimension	ETL	ELT
Processing Location	External ETL servers	Inside warehouse
Compute Model	Fixed capacity	Elastic, MPP
Data Flexibility	Schema-on-write	Schema-on-read
Time to Insight	Slower (redesign needed)	Faster (SQL iteration)
Raw Data Access	Limited/lost	Fully preserved
Best For	Legacy systems, PII	Cloud, analytics agility



Use ETL When...



Data Security & Compliance

PII must be masked/encrypted before warehouse entry



Legacy Systems

Limited warehouse compute or transformation capabilities



Heavy Pre-Processing

Complex external API integrations or ML scoring



Bandwidth Constraints

Massive datasets need filtering before transfer



Use ELT When...



Cloud-Native Architecture

Modern warehouses (Synapse, Snowflake, BigQuery)



Rapid Analytics Iteration

Frequently changing business requirements



Data Lake Strategy

Medallion architecture (bronze/silver/gold)



SQL-First Teams

Transparent, version-controlled transformations (dbt)



Trade-offs at a Glance

ETL

- ✓ Reduced warehouse footprint
- ✓ Data quality at ingestion
- ✓ Mature enterprise tooling
- ✗ Transformation bottleneck
- ✗ Inflexible, slow changes
- ✗ Raw data often lost

ELT

- ✓ Speed and agility
- ✓ Massive scalability (MPP)
- ✓ Complete data preservation
- ✗ Higher storage costs
- ✗ Governance challenges
- ✗ Potential compute spikes



Real-World Case Study

E-Commerce Analytics Platform on Azure

The Challenge

Consolidate data from multiple sources: Azure SQL (transactional), clickstream logs (JSON), ServiceNow API (support tickets), and Adobe Analytics (marketing campaigns).

The Solution

Hybrid ETL + ELT approach using Azure Data Factory, Synapse Analytics, Data Lake Gen2, and dbt for transformations.



Hybrid Architecture on Azure

Azure SQL (Transactional)

Pattern: ETL

- CDC extraction, PII masking

Clickstream (JSON Logs)

Pattern: ELT

- Raw to data lake, transform in Synapse

ServiceNow API

Pattern: ETL

- Flatten JSON, enrich with lookups

Adobe Analytics

Pattern: ELT

- Daily extract, transform in Synapse

Medallion Architecture: Bronze (raw) → Silver (cleansed) → Gold (business-ready)



Key Takeaways

- 1 ETL and ELT are tools, not religions—choose based on your context
- 2 Modern platforms benefit from hybrid approaches
- 3 Use ETL for sensitive data requiring pre-processing
- 4 Use ELT for raw data where flexibility and speed matter
- 5 Cloud platforms like Azure enable both patterns seamlessly



The Future is Hybrid

Build data platforms that are both robust and agile

Choose the right tool for each job



ProtoComet

<https://www.protocomet.com>