hevo

Mastering ETL: 9 Best Practices for a High-Performing ETL Process

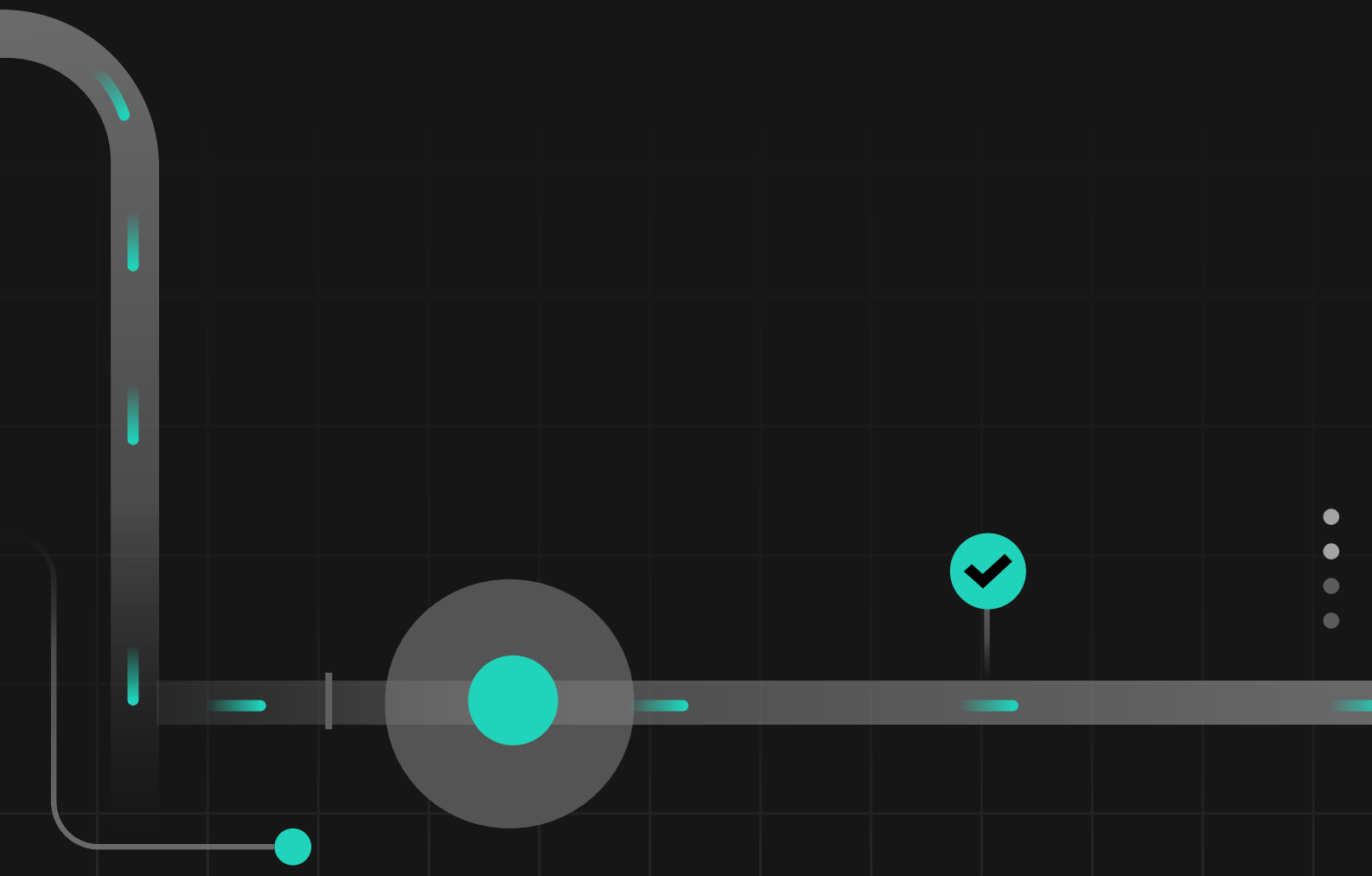

Table Of Contents

Introduction 01

Best Practices

01. Take Measures for Any Data Issues in the ETL Cycl 02

02. Improve Data Quality by Cleaning and Formatting Databases 03

03. Use Incremental Data Updates 07

04. Maintain Proper Logging of ETL Jobs 08

05. Adopt Automation in as Many Ways in Your ETL Process 09

06. Use Parallel Processing 10

07. Cache Data 11

08. Carry Out Data Audits 12

09. Adopt Modularization 14

Wrapping Up 15

hevo

Hevo is a no-code data pipeline platform that automates data integration across **150+** sources with zero maintenance. Trusted by 2,500+ companies, Hevo simplifies and scales data operations, ensuring reliable, real-time data for all users

Start your data-driven journey with Hevo today!

Introduction

Is your organization struggling to capitalize data for making fast and accurate decisions?

A highly possible reason could be that you have a scattered data collection process or a fragile data architecture that is untimely. But, it's not just you!

According to New Vantage Partner's Data and AI Leadership Executive Survey, many organizations are struggling to become data-driven. Barely 26.5% of them responded that they believe they have created a data-driven organization. And, just 19.3% said that they have established a data culture within the firm.

So, how can you make it as one of those best data-driven organizations?

Here is where a high-performance ETL with low latency and accurate replication matters! You can consolidate and format data from all sources into a single repository and easily access them for analysis and additional processing.

In this guide, we are going to break down all the problems that hamper your ETL performance and identify easily actionable steps to tackle them once and forever.

Under each best practice section, we will cover

- What's the problem
- How does it impact your ETL performance
- How can the best practice fix it?

We'll also throw in some pro tips as we go along, to help you crush your goals.

Let's jump right into it. 😄

Best Practices

Take Measures for Any Data Issues in the ETL Cycle

What's the problem?

- Manual data entry or data transformation errors caused by humans.

How does it impact your ETL performance?

- **Data quality issues** could arise that can include missing, inconsistent, duplicate, or inaccurate data.

- When data is extracted from multiple sources it becomes **challenging to integrate** and align the data that may need to be merged, transformed, or cleansed.

- **Data validation issues** may include data that fails validation checks or does not meet defined business rules.

- **Data can be mapped incorrectly**, resulting in data being loaded into the wrong fields or not being loaded at all.

How can the best practice fix it?

- **Add autocorrect tasks** for predictable errors and add data validation constraints by talking to source partners if the error persists.

- **Set up an alert system and push notifications** that help you correct the error immediately. This will be most helpful in case of unauthorized access or any other security breach.

- **Add checkpoints** that can help to keep track of unexpected errors.

Pro tip: Understand the type and amount of data you will be handling. Knowing the data types, schema, and other specifics of your data is always important.

Just a glimpse into how an automated pipeline like Hevo can help you address the errors:-

> “What I love the most about Hevo is how it automatically fixes most of the data errors, and when Hevo can't solve the issues, it sends out a notification. Hevo allows me to run transformations like date format conversion, concatenating the values, dimensionality reduction on the fly”
>
> - **Kashyap Shreepathi,**
> *Data Engineer*
>
> Groww

Improve Data Quality by Cleaning and Formatting Databases

What's the problem?

- Manual data entry or data transformation errors caused by humans.

How does it impact your ETL performance?

- Dirty or unclean data, with missing values, inconsistent formats, or duplicate records, can cause ETL processes to take longer and **trigger batch failures.**

- It results in **inaccurate models or analysis after ETL**, as data inconsistencies can lead to errors in data transformations, aggregations, or calculations.

- It can lead to increased data volume, which can **strain system resources**, such as CPU, memory, and storage, and negatively impact scalability. This can cause performance issues, such as slower data processing, longer ETL job execution times, and reduced system responsiveness.

- It can increase the **complexity of data integration tasks**, such as data mapping, data validation, and data transformation.

How can the best practice fix it?

You can clean/format databases with the following techniques.

- **Divide large database tables into smaller ones**
  When the index tree is shallower and each partition has its own indices, the data may be accessed more quickly. Additionally, it enables rapid metadata operations to switch data in and out of a table rather than the actual insertion or deletion of data entries.

- **Delete redundant entries**
  Instead of wasting time transforming data just to cull it later, clear away any unnecessary items you find in a database. You can use data normalization technique for this. Also, use consistent naming conventions for tables and columns to make the database easier to understand and use.

- **Use automated data quality tools**
  You can find missing and inconsistent data in your data set using a data quality tool. Different options are available.

05 Data Staging and Cloning

Snowflake houses a feature called **Stages** that assists in carrying out bulk loads. Here, the data remains in a **hold position** until it is loaded. To stage your data, you can either use the **internal Snowflake** or **external storage** like **S3** and **Azure**. You'll only have to pay for the basic data storage cost with internal storage. There's **no extra cost** for the Fail-safe and Time Travel features of internal storage. As a best practice, you can always stage data while carrying out ETL for Snowflake. You can stage data as partitions based on **Logical Paths.** Staging your data will also allow you to execute **concurrent COPY statements** on multiple subsets in parallel.

Snowflake further supports a feature, **Zero-Copy Cloning**. This will help create a clone of any schema, table, etc. Hence, it can be helpful for Data Backups and Replication. If you don't make any changes to the cloned objects, you **don't incur** extra costs. You can create derived objects with ease.

06 Design Efficient Queries

While writing queries for specific use cases, ensure you implement certain best practices. These will help you make your SQL Queries more efficient & deliver better performance:

- When bulk loading data from your S3 bucket, always sort your data **before** the load occurs. Avoid using the **ORDER BY statement** while loading data using the COPY command.

- Avoid executing simple queries multiple times. For example, SELECT (1), SELECT CURRENT_DATE, SELECT COUNT (1), etc. Executing these often can turn out to be quite costly as Snowflake **bills you per second**. With multiple small queries, you can quickly **shoot or add up** to your credit usage.

- Employ the **DISTINCT or GROUP BY command**s to avoid working with duplicate data.

OpenRefine:

Features	Pros	Cons
Runs on the user's local machine.	Has a large community for help.	Requires installation on the local machine.
Provides data parsing, clustering, and transformation operations.	Can handle large datasets.	Require some technical skills to set up and configure.
Has a undo/redo feature for easily reverting changes.	Has a user friendly interface with a lot of customization options.	

Trifacta:

Features	Pros	Cons
Provides data profiling, parsing, formatting, and validation operations.	Provides intelligent suggestions based on the data being cleaned.	Has a paid license, which may not be suitable for users with budget constraints.
Smart data wrangling feature automates the data cleaning process.	Can handle large datasets.	There is a learning curve to fully utilize its advanced features.
Has collaboration features for teams working on data cleaning tasks.	Integrates with popular data storage systems.	

Pro tip: Back up your data before doing the replication or making any changes to prevent any data loss.

Use Incremental Data Updates

What's the problem?

- Increased processing time due to large volumes of data.
- Untimely delivery of streaming data.

How does it impact your ETL performance?

- The **transformations and cleansing rules** can not be scaled efficiently to large data sets.
- There is a risk of **overwriting existing data** in the target system, especially in scenarios where data integrity is critical.
- You **can't enable real-time** or near-real-time ETL processing. This is important when the timeliness of data is crucial, such as in analytics or reporting applications that require up-to-date data.

How can the best practice fix it?

- During the incremental update, you only add the new data into your ETL pipeline instead of replicating historical data and starting over from scratch whenever your data sets are updated.

Pro tip: Allocate enough time for the incremental extract process evaluation and testing. The data warehouse's data will be wrong if the extracts from the source systems aren't accurate. The user/business trust in the data warehouse can be impacted by even a few lost records as a result of some data anomaly or unforeseen incident.

Maintain Proper Logging of ETL Jobs

What's the problem?

- Insufficient logging of all ETL jobs.

How does it impact your ETL performance?

- Without proper logging of the ETL process, it becomes harder to pinpoint the **root cause of errors** or exceptions, resulting in longer resolution times and increased downtime. This in turn results in critical bandwidth loss of the data team.

- It can result in **limited visibility** into the multiple steps involved in the ETL process including data extraction, transformation, and loading.

- It can lead to incomplete or **inadequate auditing** of ETL process. This can cause compliance violations, financial penalties, and reputational damage. It requires additional time and effort to manually review and verify ETL activities.

How can the best practice fix it?

- A good ETL solution can help you log your ETL jobs that take place before, during, and following an ETL activity.

- It improves the **visibility of metrics** such as data volume, processing time, error rates, and resource utilization.

Adopt Automation in as Many Ways in Your ETL Process

What's the problem?

- Slow data integration process due to human involvement.

- Manual human errors and risks of data corruption which increases cost of testing and leads to inaccurate data transformations.

How does it impact your ETL performance?

- Without automation, ETL processes may require **significant manual effort**, such as manually extracting, transforming, and loading the data into the target system. This can be time-consuming, error-prone, and lead to inconsistencies in data quality.

- Manual ETL processes **may not be scalable**, making it difficult to handle large volumes of data. This can lead to issues such as system crashes, data loss, and other performance-related issues.

How can the best practice fix it?

- You can automate the steps in ETL including data ingestion, testing, loading, and transformation.

The steps involved are:

- Identify the data sources that need to be integrated and choose an ETL tool that meets your organization s needs. There are many reliable and high-performance ETL tools available, such as Hevo Data.
- Define the data flow by specifying which data sources will be extracted, the transformations that will be applied, and the target system where the transformed data will be loaded.
- Develop the ETL process using the chosen tool. This typically involves creating data mapping, designing transformations, and configuring the ETL tool to automate the process. Let s say you are opting for Hevo Data. It takes just 3 easy steps to set up the whole process.
- Test-run the ETL process with sample data and verify that the transformed data meets the required quality standards.
- Schedule, automate, and monitor the ETL process to ensure it works correctly.

Use Parallel Processing

What's the problem?

- Inability to run multiple integrations at once.

How does it impact your ETL performance?

- Data transformations such as data cleansing, validation, aggregation, and enrichment can be **computationally intensive** and time-consuming.
- **Concurrent extraction of data** from multiple sources takes time.
- **Scaling horizontally** by distributing data becomes difficult.

- The **utilization of available computing resources,** such as CPUs, memory, and storage would be inefficient.

How can the best practice fix it?

- During parallel processing, an ETL solution records each running time and compares them on a regular basis.

- Processing is stalled by sort and aggregate operations (count, sum, etc.), which must finish before the subsequent task may start. Processing in parallel maximizes resources as opposed to serially.

- Making simple changes like disabling check and foreign key constraints, or separating triggers into a complete task, reduces the running time of an ETL cycle.

- By distributing data processing across multiple nodes, parallel processing can also reduce the risk of system failure resulting in **superior fault tolerance.**

Cache Data

What's the problem?

- Difficulty in accessing recently used data for ETL processes.

How does it impact your ETL performance?

- Repeatedly accessing and loading data from the original data sources becomes time-consuming and **resource-intensive.**

- The data transformations may need to be performed repeatedly on the same data, leading to **increased processing time.**

- ETL system may generate an **increased load** on these data sources, as it needs to repeatedly fetch data from them.

How can the best practice fix it?

- Data caching **stores previously used data** in memory so that it can be accessed efficiently again. This reduces the need to **recompute or reapply** the same transformations, leading to faster data processing times.

- You can break down a pipeline into phases that each ingest and emit files. If a stage of the pipeline malfunctions, you only need to fix it and run it again. This can be orchestrated in numerous ways, including using Makefiles and Airflow.

Pro tip: Cache data expiration is critical to avoid stale data. Set an appropriate cache expiration time based on the data freshness requirement to ensure data correctness.

Carry Out Data Audits

What's the problem?

- Abnormalities in data such as missing or corrupt records that arise even when there are no errors which can lead to compliance issues with HIPAA regulations.

How does it impact your ETL performance?

- Data integrity issues, such as **data corruption, data loss**, or unauthorized changes to data may arise which can go undetected.

- Failure to meet various **regulations and compliance requirements**, such as GDPR, HIPAA, or SOX by validating data against predefined rules and requirements.

How can the best practice fix it?

- Data auditing is the process of examining, validating, and verifying data to ensure its accuracy, integrity, consistency, and compliance with relevant rules, regulations, and policies.

The steps in a data audit are:

- Identify the specific data elements, systems, or processes to be audited and determine the desired outcomes and expectations of the audit.
- Identify the audit team and methodology, and set a timeline.
- Collect data from the identified sources.
- Use data analysis techniques (statistical analysis, data profiling, data visualization) to identify data quality issues.
- Verify the accuracy, integrity, and consistency of the data by comparing it against predefined criteria, such as business rules.
- Document the audit findings in a formal report.
- Take corrective actions to address the identified issues, such as data cleansing, data validation, or process improvements, and monitor the progress.
- Conduct follow up audits to verify the effectiveness of the implemented corrective actions.

Adopt Modularization

What's the problem?

- **Repetition of tasks** in ETL.

How does it impact your ETL performance?

- It is not easy to **debug and fix** issues that arise.

- Changes to the process can affect multiple areas, rather than just a single module.

How can the best practice fix it?

- The technique of abstracting ETL processes into smaller, **reusable blocks** is known as modularization.

- By breaking the pipeline down into smaller modules, the pipeline becomes more resilient, allowing for faster iteration and easier debugging. In this way, a single block of code for several different procedures streamlines the process.

- This can be accomplished by segmenting the code into several functions and utilizing various **object-oriented programming concepts**. It facilitates unit testing and creates a standard that all processes must adhere to.

Wrapping Up

In this guide, we have presented **9 key practices** that can help you execute ETL processes with the highest efficiency. These are:

Take measures like autocorrect tasks for any data issues in the ETL cycle.	Clean/format databases by dividing large databases into smaller ones, and deleting redundant entries.	Clean/format databases by dividing large databases into smaller ones, and deleting redundant entries.
Take measures like autocorrect tasks for any data issues in the ETL cycle.	Conduct audits for removing abnormalities in data.	Cache data for easily accessing data used recently for ETL processes.
Modularize tasks.	Use parallel processing	Adopt automation.

That's it! Now it s your turn to take action based on these.

If you want smooth and efficient ETL, you should try Hevo Data's no-code, zero-maintenance data pipeline solution. With Hevo, you can integrate data from more than 150 sources into a destination of your choice. ETL your data stress free in just a few minutes with this scalable, reliable solution

hevo

ETL Data Seamlessly in Real-time with Hevo Data

2000+ customers ❤️

Hevo Data's reliable pipelines to replicate data from **150+ sources** in near realtime to their preferred destinations. Schedule a call with our solutions expert to get started today!

Book a Demo!