International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 6 Issue 1 PP 71-81, 2025
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V6I1P110

Original Article

Top Skills Every ETL Developer Needs in 2025

Bhavitha Guntupalli[1], Surya Vamshi ch[2]
[1]ETL/Data Warehouse Developer at Blue Cross Blue Shield of Illinois, USA.
[2]QualityEngineer at Bank of New York Mellon, USA.

Received On: 19/01/2025 ***Revised On: 03/03/2025*** ***Accepted On: 07/03/2025*** ***Published On: 11/03/2025***

Abstract: *As companies in many different sectors get increasingly data-driven in 2025, ETL (Extract, Transform, Load) engineers will be absolutely critical. Modern ETL has evolved from beyond conventional relational systems and set batch processes into a dynamic, real-time process supporting intelligent and responsive decision-making. Rising use of cloud computing, the development of real-time streaming data, growing concerns about data security, and growing importance of artificial intelligence and machine learning in analytical processes drive this change. Modern ETL professionals must thus have a complete and flexible skill set combining exceptional technical knowledge with required interpersonal skills. With cloud-native integration tools including AWS Glue, Google Cloud Dataflow, Azure Data Factory, and open-source orchestrators like Apache Airflow and Apache NiFi, ETL experts must be exceptional. Absolutely essential is knowledge of distributed systems, container orchestration using Kubernetes, Apache Kafka, real-time data platforms, and sophisticated data modeling techniques. Of course, these days one expects scaled processes to contain privacy protections, transformation tools, and quality checks. Simultaneously, data strategy alignment with business goals, good communication, agile and cross-functional team collaboration, and competency for successful communication are of great importance. Often serving as a link between data engineering, business intelligence, and regulatory compliance, ETL engineers have quite valuable people skills. Maintaining safe and compliant data flows requires whole awareness of data governance covering knowledge of GDPR, HIPAA, and data localization rules. Modern ETL engineers need both technical knowledge and strategic awareness since digital companies rely more and more on data to spark creativity. This paper presents the most sought-after technical and interpersonal skills for ETL professionals in 2025, providing a whole guide for everyone wishing to improve their careers in the dynamic data environment and offer continuous corporate value via intelligent, strong, and compliant ETL systems.*

Keywords: *ETL Developer, Data Engineering, Data Pipelines, Cloud Integration, Data Governance, SQL, Python, DataOps, Data Security, Real-Time Processing, API Integration, Apache Airflow, Azure Data Factory, Data Architecture, Agile, Communication Skills, Machine Learning, Automation, Scalability, Metadata Management.*

1. Introduction

From their traditional constraints, the responsibilities of ETL (Extract, Transform, Load) developers have grown significantly in the modern, fast-changing digital context. Originally only a means of data transfer between source and target systems, current data architecture now plays a very major role. From simple pipeline builders, ETL developers have developed into architects of data flow, quality enforcers, compliance custodians, and active players in business intelligence, analytics, and machine learning projects. The demand for competent ETL professionals able to provide fast, consistent, and safe data has skyrocketed to hitherto unheard-of heights as companies embrace data-driven transformation more and more. 2025 promises many powerful industrial catalysts changing the expectations of ETL roles. Growing artificial intelligence and machine learning presence in business analytics calls for ETL pipelines to provide quick model training and inference on scale. Supported ETL operations run simply across hybrid and multi-cloud configurations, and cloud-native systems defined by scalable and distributed architectures have become the standard. Technologies such as Apache Kafka, Flink, and streaming databases provide real-time analytics that demand ETL systems able to consume and translate low-latency data. Moreover, extending worldwide requirements on data protection and governance such as those of GDPR, HIPAA, and cross-border data residency policieshelps to position ETL developers as absolutely essential in preserving compliance inside data operations.

This post tries to enumerate all the key skills differentiating a first-class ETL developer in 2025. It emphasizes not just the technical masteryincluding tools, platforms, and engineering approachesbut also the interpersonal skills needed for success in cooperative, cross-functional, and compliance-oriented environments. Knowing these qualities regardless of your rolethat of a team leader defining hiring criteria or developer trying to enhance their skillshelps you build productive and future-ready data teams. This stuff is presented in a sensible, realistic manner. First addressing understanding in cloud-native technologies, real-

time data engineering, data quality management, and security-oriented design, it then looks at the technical skills needed for 2025. It then looks at, among others, soft skills needed for ETL successcommunication, agile teamwork, and regulatory awareness. The last paper offers helpful guidance and learning opportunities to help professionals connect their growth in skill level with organizational requirements and market developments.

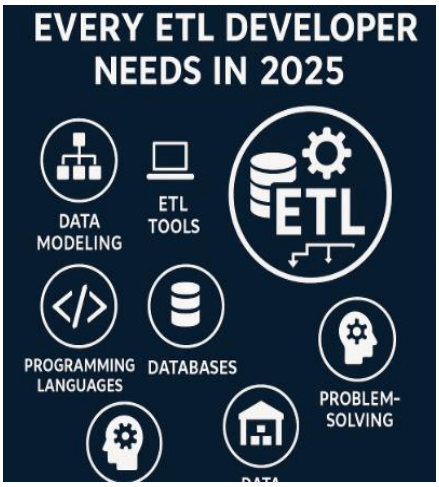

Fig 1: Essential Skills for ETL Developers in 2025

2. Foundational Technical Skills

An ETL (Extract, Transform, Load) developer still needs to have the same basic technical skills as before, but they need to be more up-to-date. By 2025, ETL experts who are skilled must be able to show that they know a lot about basic technology and acquire training on new systems. This study looks at the key skills that ETL engineers need to learn in order to create data pipelines that can grow, work well, and last a long time.

2.1. SQL Proficiency: The Lingua Franca of Data Transformation

Often referred to as structured query language, SQL provides the fundamental foundation for data manipulation and transformation. Even while current data tools and scripting languages are now readily available, SQL is still the most dependable and often used language for querying, aggregating, and manipulating datasets.

Modern ETL developers have to overcome basic synthesis of simple SELECT lines. Advanced mastery consists of running totals, moving averages, rank assignments, and cumulative sumsachieved without subqueriesall of which challenge computations across partitions.

- Designed to manage modular searches and hierarchical data structures, Common Table Expressions (CTEs) and Recursive Queries
- Following rigorous SLAs in ETL systems can be achieved by producing effective SQL that eliminates whole table scans, makes use of indexes, and reduces joins in huge datasets.
- By means of improved transformations, knowledge of execution techniques and query profiling technologies helps developers to lower computing costs in cloud systems.

SQL expertise in hybrid and cloud-native systems (Snowflake, BigQuery, Redshift) searches for mastery of platform-specific syntax, query cost implications, and caching dynamics.

2.2. Scripting Languages: Python and Scala for Flexibility and Automation

SQL excels in declarative data operations even while scripting languages like Python and Scala provide the tools essential for orchestration, transformation logic, and automation in ETL systems.

Python is the main programming language in data engineering thanks in great part to its simplicity, large ecosystem (Pandas, PySpark, SQLAlchemy), and straightforward integration with cloud platforms and orchestration tools. Python ETL programmers for:

- Analyzing and organizing data
- API integration to take in outside data
- Pipelines integrity monitoring scripts for automation
- Developing unique transformation logic when SQL proves inadequate

Especially in systems built on Spark, Scala provides excellent support for distributed computing. It is sought for in low-latency, great parallelism-dependent, high-performance data pipelines. Mastery of Scala is absolutely critical in systems such as Apache Spark or Flink. Scripting tools enable control of edge conditions, connection with metadata management systems and logging frameworks, and application of tailored logic to enhance data.

2.3. Data Modeling: Designing for Analytics and Performance

Great data modeling is basically the building block for ETL development, which is a very important but sometimes overlooked skill. The data itself directly affects the way it's formatted, the ease of its access, the speed of the query, and the flexibility of the analysis.

- **Star and Snowflake Schemas:** These are the main multidimensional modeling approaches that are the basis of data systems, data warehouses, and OLAP. A well-organized star schema not only makes searches easier but also improves performance, whereas snowflake schemas offer consistency and flexibility.
- **Dimensional Modeling:** One of these methods is dimensional modeling, which is the design of fact and dimension tables that represent business processes. It

also makes sure that analytic models of business intelligence technologies are understandable for non-technical users.

- **Denormalization Strategies:** If ETL developers decide to go for better efficiency by giving up standardization, they need to make that choice carefully. Most of the time, denormalization is employed in systems that perform high-velocity querying, especially for dashboards and reporting tasks.
- **Slowly Changing Dimensions (SCD):** The accuracy in the historical data and the ability to trace the changes over time are totally dependent on the careful administration of the modifications in the dimension tables, which means, for example, customer addresses.

2.4. ETL Tools Mastery: From Traditional Giants to Modern Frameworks

The tooling space is ever-evolving; in 2025 it will be a combination of traditional and innovative ETL techniques. Going deep into the subjects of mastering those areas allows tech people to adapt quickly to new things, navigate different ecosystems, and pick the right tech for certain tasks.

Traditional ETL Tools:

- **Informatica PowerCenter:** Indicatively, Informatica PowerCenter drives its remarkable reliability, profound legacy system compatibility, metadata-driven pipeline, and traditional ETL enterprise data integration skills of the highest-level industry.
- **Talend:** Performance via open-source and commercial software, Talend ensures flexibility and the comprehensiveness of networking capacity. Talend's graphic user interface accelerates ETL design.
- **Microsoft SSIS (SQL Server Integration Services):** Services for Microsoft SQL Server Integration (SSIS) Common among businesses heavily relying on Microsoft-focused software, SSIS offers SQL Server Agent-based complex data flow transformation and scheduling functionalities.

They do not have scalable or agile characteristics, which are inherent in cloud-native; however, graphical designing and significant logging are the features that these systems grant.

Modern ETL and ELT Tools:

- **Dbt (data build tool): A** transforming power in the dbt (data build tool) allows analytics engineers in the ELT sector to produce modular SQL transformations, Git version control management, and orchestration platform implementation. In modern analytics systems, its perfect interaction with Redshift, BigQuery, and Snowflake, among other tools, makes it absolutely essential.
- **Apache NiFi and Airbyte:** Visual, flow-oriented technologies that facilitate data intakeApache NiFi and Airbyte aid especially for streaming or real-time applications.
- **Airflow and Prefect:** Emphasizing orchestration over transformation, these systems let ETL developers use scheduling, retries, and observability to track demanding data activities.

Developers must be knowledgeable in at least one platform from every field and know how to adapt across technologies as business needs change. Changing traditional ETL systems from Informatica to dbt and Airflow is one often-modernizing project.

3. Cloud and Platform-Specific Expertise

Cloud systems tightly entwine ETL development through 2025. The capacity to build, control, and maximize ETL pipelines inside scalable and cost-effective cloud-native systems has become increasingly important as companies transfer their operations to these frameworks. ETL developers not only possess tools but also have to understand architectural ideas and operational models of cloud services. This part stresses the fundamental knowledge about clouds and platforms defining a current ETL practitioner.

3.1. Cloud Services: Leveraging Platform-Native ETL Tools

Every leading cloud vendor has a great array of robust ETL and data orchestration capabilities. Mastery of at least one of these platforms helps ETL engineers to operate in multi-cloud and hybrid systems, together with a functional awareness of others.

3.1.1. AWS Glue

Designed as a whole managed ETL tool, AWS Glue closely interacts with the AWS environment. It addresses automatic schema inference, data cataloging, and serverless Spark tasks.

- Important tools are Glue Studio for visual task creation and debugging.
- Glue Jobs let Python and Scala scripts handle intricate transformations.
- Using Glue Data Catalog, understand metadata across S3, Redshift, and Athena.
- Integrating with AWS Lake Formation, Step Functions, and EventBridge helps one to safely and clearly handle processes.

Expert knowledge in AWS Glue enables ETL engineers to create extremely scalable pipelines free from infrastructure installations for sporadic and burst workloads.

3.1.2. Azure Data Factory (ADF)

At an enterprise level, ADF offers code-free as well as code-based data integration options. It facilitates data movement across on-site and cloud environments as well as more than 90 connectors.

Key features include

- Data flows are anchored on Spark in the background for visible transformation logic.
- Branching, conditional logic, retries, and parameterizing pipeline orchestration.
- Perfect harmony with Azure Functions, Blob Storage, and Azure Synapse.

Especially helping developers in Microsoft-centric environments to maintain consistency across their data ecosystem is ADF.

3.1.3. Google Cloud Dataflow

Built upon Apache Beam, Dataflow is a strong tool for batch and stream processing.

- Two main strengths are a coherent batch and streaming ETL programming framework.
- For large, unforeseen data operations, autoscaling and dynamic workload balance are ideal.
- Easy connection with Cloud Storage, Pub/Sub, and Big Query.

For real-time ETL applications, dataflow is perfect since it helps developers to handle IoT pipelines, large event streams, and machine learning preparation chores. Automation, scalability, and observability run across all platforms as uniting themes. Using the inherent logging and alerting features, developers have to understand how to monitor job execution, limit expenses, and handle fault tolerance.

3.2. Serverless Architectures: Scalability without the Overhead

ETL workload distribution and scalability rules have altered with serverless computing. Under a serverless design, the cloud provider manages all infrastructure, therefore freeing developers to focus only on coding or process specification. This is especially useful for ETL tasks motivated by events, variable in scale, or uneven frequency.

Benefits of Serverless for ETL:

- **Scalability:** Depending on task intensity, automatically changes resources to guarantee optimal performance free from human involvement.
- **Cost Optimization:** Pay-per-use pricing helps to save money on seldom-needed projects by removing the requirement for over-provisioning resource**s.**
- **Reduced Maintenance:** Developers free from the responsibilities of provisioning, patching, and server management can concentrate more on data logic and quality.

Among these are AWS Glueserverless Spark, Google Cloud Functions applied in orchestration, and Azure Durable Functions for extended ETL processes. ETL developers have to know how to monitor execution timings and retries, safely create environment variablese.g., new files in S3 or Blob storageand employ serverless triggersthat is, new files in S3 or Blob storage.

3.3. Infrastructure as Code (IaC): Consistent, Reproducible Deployments

Infrastructure as Code (IaC) will help DevOps-era installations, setups, and deployments of cloud resources consumed in ETL pipelines to be automated. Infrastructure as Code (IaC) promises to support version control, scalable deployments inside Continuous Integration/Continuous Deployment (CI/CD) processes, and homogeneity across environments (development, testing, and production).

3.3.1. Terraform

Developed by HashiCorp, Terraform is an open-source Infrastructure as Code (IaC) application widely used for the definition and provisioning of cloud infrastructure across several providers (AWS, Azure, GCP) via declarative configuration files.

- Terraform lets ETL developers create Lambda functions, Azure Data Factory pipelines, IAM roles, or Glue jobs.
- Apply interdependent services, including storage, computing, and orchestration.
- To enable rollback and create audit trails, keep store configuration in source controlthat is, Git.

Terraform's provider-agnostic design makes it ideal for companies running many cloud providers.

3.3.2. AWS CloudFormation

CloudFormation allows you to define and provision infrastructure by using AWS-specific Infrastructure as Code (IaC) templates. It helps to enable major interactions with AWS products and guarantees automatic and safe resource management.

Main advantages consist in:

- Stack controls and drift detection help to stop accidental modifications.
- Python or TypeScript enhanced Infrastructure as Code (IaC) by means of AWS Cloud Development Kit (CDK) integration.
- Structural changes set to preview before execution are facilitated.

Eliminating manual provisioning, reducing human error, and enabling developer, infrastructure engineer, and security team collaboration is the aim whether utilizing Terraform or CloudFormation.

4. Real-Time and Streaming Data Skills

Since companies rely more on fast insights for competitive advantage, real-time and streaming data capabilities have become rather crucial for ETL developers. Applications include fraud detection, IoT monitoring, dynamic pricing, and real-time personalization, which call for a more complex model of batch processing than the current one. If ETL engineers are to provide speedy and scalable decisions, they must be masters in event-driven systems, low-latency designs, and streaming platforms by 2025. This section investigates the main tools, architectural techniques, and optimization methodologies defining real-time ETL excellence.

4.1. Apache Kafka, Spark Streaming, and Flink: Tools and Use Cases

Strong data entry systems and processing form modern streaming ETL. Apache Kafka, Apache Spark Streaming, and Apache Flink are among well-known options with varied advantages for various uses.

4.1.1. Apache Kafka

Real-time data pipelines execute their basic messaging system via Kafka. It provides a distributed, fault-tolerant event log separating consumers from data providers, therefore enabling low-latency data streaming and high throughput.

Use Cases:

- Obtaining clickstream information from internet and mobile applications
- Collecting logs and telemetry from scattered systems
- Financial transaction real-time streaming for fraud prevention
- Designing microservices driven by events

Kafka gives the basic framework for event intake for ETL developers. Mastery of Kafka topics, partitions, consumer groups, and Kafka Connect for source/sink interactions is vital.

4.1.2. Apache Spark Streaming

Spark Streaming and its sequel Structured Streaming gives almost real-time stream processing using the current Spark engine and DataFrame API. It shines in settings demanding sophisticated data in transit computing.

Use Cases:

- Windows of computation and real-time aggregations
- Combining flowing and stationary data
- Identification of discrepancies in sensor data
- Feeding machine learning models processed data

Since Spark can mix batch and streaming activities under one engine, organizations requiring both scalability and flexibility will find it ideal.

4.1.3. Apache Flink

Flink guarantees low latency, event time processing, and stateful operationsa truly streaming-first engine. It is designed for high throughput and long computations.

Use Cases:

- Cep, sometimes known as complex event processing:
- Variations in stateful streams
- Notifications right away and identification of irregularities
- Improving event streams loaded with additional data

Strong support of precisely-once semantics and backpressure control by Flink provides a competitive edge in mission-critical, high-frequency activities.

4.2. Lambda and Kappa Architectures: Understanding Real-Time vs Batch Trade-Offs

The ETL designers must know the two main architectural models, Lambda and Kappa, if they want to build efficient real-time data systems.

4.2.1. Lambda Architecture

- The Lambda architecture is a design that incorporates both batch and real-time elements. The speed layer deals with real-time data for quick analysis, while the batch layer takes care of high-latency, complete datasets (like nighttime operations). The serving layer then provides the end users with a view that combines both.

Pros:

- Data accuracy is assured by the batch reprocessing.
- Protection against delayed events or data loss.

Cons:

- It is a necessity to keep track of two codebases.
- The increased complexity of the system and the heavy maintenance load.

4.2.2. Kappa Architecture

Simplifying the pipeline, kappa design meets reprocessing needs as well as real-time processing with a single streaming layer. Among the tools allowing you to progressively reprocess data streams are Kafka and Flink.

Advantages:

- Simplified construction with one processing stream
- More appropriate for modern, stream-centric systems

Challenges:

- Demand full control over systems of ordering, status, and recovery.
- Perhaps not ideal for any historical reprocessing plant

Knowing the trade-offs of various architectures helps ETL designers to choose designs in line with system complexity, fault tolerance, and latency.

4.3. Data Freshness and Latency Reduction Strategies

Attaching low-latency ETL requires deliberate methods at every stage of input, processing, and distribution.

- **Micro-batching and Windowing:** it is the clock of small data intervals rather than waiting for bigger ones that result in lower latency.
- **Stateful Stream Processing:** the process keeps the context throughout; thus, it is possible to make aggregate computations without accessing the storage.
- **Backpressure Management:** Thus Flink and Kafka together create a unity that gives them the potential of regulating the flow and preventing the system from being overloaded.
- **Incremental Processing:** The info that has just been brought in or refreshed is processed only. Such a technique leads to fewer transformations, which in turn result in higher efficiency.
- **Optimized Serialization:** The smaller and lighter models, such as Avro or Protobuf, allow for faster data flow between services.

ETL builders can realize vast improvements in performance by tuning a number of parameters like batch size, checkpoint intervals, and watermarking policies.

5. Data Governance and Security Skills

ETL developers will go beyond their traditional duties of data transmission and transformation to become major actors in the safe, ethical, legally acceptable administration of data by 2025 as data quantities rise and needs are more constrained. Dependent on strong systems of security and data governance, reliable data pipelines are Today's ETL developers have to be absolutely aware of access control systems, metadata management, lineage tracking, and data security rules. This section addresses the fundamental security and governance techniques every modern ETL practitioner should pick up.

5.1. Compliance: GDPR, HIPAA, and Regional Data Privacy Norms

The basic need of today is regulatory compliance; it is not a choice. ETL engineers actively conduct compliance data processing activities, especially in industries including healthcare, banking, and e-commerce.

5.1.1. GDPR (General Data Protection Regulation)

Businesses under GDPR have to ensure personally identifiable data (PII) is acquired under user authorization and treated openly and transparently. This translates for ETL developers into:

- Locating and noting sensitive information (such as names, addresses, email addresses).
- Allowing verifiable systems to assist in fulfilling requests for the right to be forgotten (data deletion).
- As necessary, anonymizing or pseudonymizing data.
- Recording every step of data processing for audit preparation.

5.1.2. HIPAA (Health Insurance Portability and Accountability Act)

HIPAA is a law that requires very strict safeguards for electronic protected health information (ePHI). ETL workflows, however, have to adhere to the following:

- Data encryption both while stored and being transferred
- Recording all the accesses and changes made to the healthcare data
- Making sure that only authorized systems and users have access to PHI
- Retaining the audit trails as evidence of compliance during security checks.

5.1.3. Regional Privacy Norms

Developers need to keep in mind that there are also other frameworks, such as:

- CCPA/CPRA (California).
- LGPD (Brazil).
- PIPEDA (Canada)
- Data Residency Requirements in the EU, UAE, India, etc.

Creating ETL processes that are aware of compliance with the laws and regulations means much more than just ticking boxesit is about creating flexible pipelines that have the capability to adjust to new regulations and privacy that customers expect.

5.2. Data Lineage and Cataloging: Enhancing Visibility and Trust

Knowing the origins, the processing that happens along the way and the destination of data is very important for governance, auditing, and troubleshooting. Data lineage and cataloging are the main aspects of this.

5.2.1. Data Lineage

Lineage gives a visual and programmatic representation of the journey of the data from the source to the destination. It allows:

- **Impact analysis:** Recognizing the downstream sectors that will be influenced by changes in the upstream sources

- **Auditing and debugging:** Locating the place where the fault or the anomaly first occurred by following the trail very quickly
- **Compliance reporting:** Supplying evidence to regulators that data is handled responsibly

Several applications, such as Apache Atlas, OpenLineage, and DataHub, are utilized for lineage tracking both in traditional and cloud-native environments. ETL developers should go far in making sure that transformations, joins, and aggregations are clearly documented and easily accessible in these systems.

5.2.2. Data Cataloging

Data catalogs such as Alation, Collibra, and Google Data Catalog enable users to browse inventories of data assets that have metadata, ownership information, and quality scores. ETL developers carry out jobs that help cataloging:

- It allows users to automatically tag and classify data during the pipeline run.
- Providing metadata like data types, source systems, and update frequencies.
- Enables data stewards and analysts to find reliable datasets without any difficulty.

Lineage and cataloging not only offer more transparency in the data, but they also create trust between the engineering, compliance, and analytics teams by establishing collaboration.

6. DataOps, CI/CD, and Automation

The modern ETL environment demands not only functioning pipelines but also robust, verifiable, version-controlled, entirely automated systems that fit very well with corporate DevOps methods. ETL developers are anticipated to be as technical and precise as software engineers in 2025 using DataOps concepts to design agile, high-quality data systems. Key elements of this transition are automated testing, continuous integration and deployment, version control, and enhanced orchestration. This section provides the fundamental engineering and automation methods defining a modern ETL professional.

6.1. Pipeline Testing: Unit, Integration, and Regression Testing for ETL

Testing in ETL makes ensuring that the data pipelines offer you outputs that are correct, consistent, and dependable. On the other hand, data pipelines operate with datasets and schemas that change, which makes testing harder and less clear.

6.1.1. Unit Testing

Unit testing within an ETL concentrates on a single transformation function or script. For instance:

- Checking if a date-parsing function is able to standardize different date formats in a correct way.
- Ensuring that the filter logic is given the records that meet the particular criteria only, thus testing the logic of records.

These tests are typically implemented with the help of tools such as PyTest, unittest (Python), and Great Expectations.

6.1.2. Integration Testing

Integration tests confirm that the parts function properly together throughout the systems. Such tests may verify:

- Whether data reaches the staging area correctly from the source.
- Whether the transformations are implemented uniformly even if there are changes in the schema.
- Whether the APIs that are used for the ingestion process respond as they should in different cases.

While carrying out these tests, the use of a mocking or sandbox environment is very important, as it ensures that the production systems are not corrupted.

6.1.3. Regression Testing

ETL regression testing confirms that modifications have not adversely affected the existing logic or made unexpected changes. Some of the frequently used methods are:

- Snapshots were taken of the data acquired before and after the changes and then compared.
- Checking if the table structure has the same number of fields or the names of the fields are consistent to catch any unexpected changes.
- Employing data diffing tools such as Datafold or Soda.

Integrating these tests into automated workflows will definitely help in identifying data quality issues at earlier stages of the development lifecycle.

6.2. Version Control and DevOps Practices: Git, CI/CD Pipelines

Version control and CI/CD, which are automation tools, make up the core of the DataOps of today. ETL developers rely on these tools to record their changes, work together efficiently, and deploy the pipelines in a trustworthy manner.

6.2.1. Git for ETL

They can utilize Git for ETL code in any form, for instance, SQL scripts, Python jobs, or YAML pipeline definitions. This allows them to trace every change and, if needed, to undo the changes. The teams receive the following benefits:

- Branching strategies (e.g., feature branches, GitFlow) to isolate changes.
- Code reviews and pull requests to enforce quality.
- Commit tagging to mark production releases.

Additionally, when it comes to the use of these visual ETL tools (such as Informatica or ADF), one can either export pipelines as code or metadata. This, in turn, enables the integration of these pipelines with the version control systems.

6.2.2. CI/CD Pipelines

Continuous Integration and Continuous Deployment are the techniques of using automation for the testing and delivery of ETL code. The most commonly used tools are:

- **Jenkins:** A highly configurable automation server that can manage ETL deployments of any complexity, carry out tests, and initiate builds when a code change happens.
- **GitHub Actions:** Best suited for small pipelines, GitHub Actions is a natural part of repositories and it also provides reusable workflows for test automation and deployment.
- **GitLab CI/CD,** Bitbucket Pipelines, and Azure DevOps are still frequently used by people who work on cloud-native ETL projects.

CI/CD equips teams to deploy changes at a faster pace, identify defects in the earlier phases, and be assured that the environments are the same in dev, test, and production.

7. Business and Analytical Skills

ETL development is based on technical skills, but being able to comprehend and interact with the business world makes a technically skilled developer into a highly strong data specialist. In addition to working on the pipeline after 2025, ETL developers will also need to figure out what data is needed, make sure it fits with the company's goals, and use analysts, stakeholders, and leaders to help them make decisions. This portion looks at the analytical and business skills that ETL developers need to be able to combine real insights with raw data.

7.1. Understanding Business Context: KPIs and Domain-Specific Knowledge

To build data pipelines that offer analytics of real sense, a profound understanding of business goals is vital. Key performance indicators (KPIs) in the respective domain should be familiar to ETL developerswhether it is customer churn in subscription business, on-time delivery in logistics, or claims processing time in insurance. This recognition empowers developers to:

- Focus on the proper data sources and the right transformation logic.
- Build dimensional models (e.g., star schemas) that illustrate the business' nature.
- Come up with the possible stakeholder inquiries and make sure the data structure is compatible with the flexible exploration.

In addition, a developer's familiarity with a particular domainlike healthcare language (ICD codes, ePHI), the financial reporting standards, or retail seasonalitygreatly improves the developer's ability to spot anomalies, explain the trends, and identify data quality problems that are not yet obvious.

7.2. Collaboration with Analysts and Stakeholders

ETL development now goes beyond a reasonable backend requirement. Developers need all of the data analysts, business intelligence teams, product managers, compliance officials, and operational partners to be actively involved. This requires:

- Getting requirements from people through interviews, workshops or reading documents.
- Co-creating data solutions by going through sample dashboards or discussing the analytic workflow.
- Handling conversations about what data can be accessed, the schedule of data transformation, and dealing with unusual cases.

Excellent cooperation guarantees that data pipelines are technically solid as well as useful for corporate objectives. Like sprint planning and retrospectives, agile ceremonies are essential turning points in connecting dynamic corporate needs with ETL responsibilities. In high-performance teams, regular data discovery projects let ETL engineers help analysts probe new metrics and find latent linkages among the data.

7.3. Communicating Data Insights Effectively

Generally, analysts are the owners of data storytelling; however, ETL developers are the ones who are usually the first to come across data anomalies, trends, and opportunities for optimization. Even a short, informal conversation with them can result in a large change in data quality, system design, and business decisions.

ETL developers should:

- Take advantage of visual tools (e.g., Metabase, Looker, Tableau) to confirm the outputs and communicate the summaries.
- Write down the documentation and pipeline metadata in a clear manner that explains the assumptions, shortcomings, and data logic.
- Signal violations in the source data at the initial stage, along with the description of the changes necessary to solve the problem.

Trust-building through clear and concise communication with stakeholders positions ETL developers as key contributors in the data value chain, not only as implementers but also as informed collaborators.

8. Soft Skills for Cross-Functional Impact

Technical expertise is just one measure of an ETL developer's output; their ability to lead cross-functional teams,

manage complexity and communicate effectively is also highly valued. Data-driven organizations rely heavily on ETL pipelines to run their business, and thus, personal skills are becoming increasingly important to maintain alignment, decrease conflict, and ensure consistent corporate value. The major soft skills highlighted in this article allow ETL engineers to be the brightest stars in team settings at lightning speed.

8.1. Problem-Solving and Adaptability

Almost seldom are data ecosystems fixed in nature. Schema changes, missing data, upstream discrepancies, and shifting business needs are all things that ETL engineers must routinely and consistently negotiate. Modern data engineering is distinguished by the ability to spot issues, test several solutions, and quickly respond. Solving a problem requires not only error correction but also foresight of potential failures, development of a recovery strategy, and continuous data workflow upgrading. Flexibility also means using fresh tools and ideaslike switching from batch to real-time ETL or including machine learning workflows.

8.2. Agile Methodology Participation

Agile teamswhich include fast feedback loops, cross-functional collaboration, and brief cyclesare gradually embracing ETL developers. Agile's daily stand-ups, sprint planning, and backlog creation, among other methods, ensure that ETL projects are in alignment with corporate goals. Developers need to have the skills of turning difficult projects into understandable stories, assessing work, and sharing progress at sprint meetings. Agile fluency is a way of connecting technical execution to corporate agility.

8.3. Written and Verbal Communication Skills

Effective communication goes hand in hand with successful collaboration. ETL developers obviously have to express technical details in simple words to non-technical people who cannot comprehend. Clarifying is essential in any case, like explaining data transformations, conveying out-of-range cases, or presenting pipeline status. During group discussions, strong verbal communication plays a vital role in solving ambiguities quickly, while short written updates and documentation make it easier to maintain and share the knowledge between teams.

8.4. Continuous Learning Mindset

Data engineering is moving swiftly because new tools, frameworks, and best practices are always being produced. ETL developers may stay up to date and adaptable by always learning new stuff. This means taking classes online, reading books, visiting to events in your area, and trying out new technologies like data contracts or observability systems.

9. Case Study: ETL Developer Transformation in a FinTech Company

9.1. Background: From Legacy to Cloud-Native

At the beginning of 2023, a digital finance FinTech company of medium size embarked on a change to become a data-centric company. Besides other things, the company was mainly using data for underwriting, fraud detection and performance analytics; however, the very old ETL system was just the beginning of the problems that had already appeared. At present, the system is composed of batch scripts, manual work in Excel and on-site SQL server integration services (SSIS). As data volume and velocity increased, these solutions showed the lack of scalability, adaptability, and speed necessary for real-time decision-making and regulatory reporting. The government realized these limitations and pledged to install a new data system that would be cloud-native. The use of AWS features while still keeping with modern orchestration technologies meant providing a data environment that was more agile and under control. Indeed, technology was a crucial factor in this success, yet the success of this transformation still depended on the ETL development group raising their capability and execution methods.

9.2. Challenge: Bridging the Skills Gap

The existing ETL team of the company was competent in traditional tools, but they did not have knowledge of modern data engineering concepts such as serverless architecture, DataOps, and cloud governance.

A number of core problems appeared:

- **Limited cloud expertise:** The developers didn't have knowledge about AWS-native tools like Glue, S3, and IAM policies.
- **Lack of automation:** The situation with no version control, lack of formal testing, and the presence of manual deployments made delivery slow and risk high.
- **Governance concerns:** The team didn't have practices, tools for lineage tracking, access control, and audit readiness, so they were non-compliant with the stricter rules like SOC 2 and GDPR.
- **Siloed collaboration:** Data engineering, analytics, and business teams were working separately and hence, they were duplicating the work and were having misaligned priorities.

These problems have ultimately created a great divide for the upskilling and modernizing of the process to meet the changing data objectives of the company.

9.3. Outcome: Quantifiable Gains and Cultural Shift

Within nine months, the overhaul has brought forth tremendous impact:

- **Pipeline Efficiency**: The time needed for ETL jobs was reduced by 40% because of Glue’s serverless infrastructure and Airflow’s efficient dependency tracking, which resulted in fewer tasks to execute.

- **Cross-Team Collaboration:** The sharing of sprint schedules and the use of integrated development environments not only made it easier for the teams to innovate and create together but also reduced the problems caused by misunderstandings or unclear requirements.
- **Compliance Readiness:** Data lineage tracking, encryption, and access controls were all part of the company's csr setup hence the organization was able to pass the major SOC 2 audit without any critical findings.

- **Developer Growth:** The team members have been very enthusiastic and confident; hence, they got cloud certifications and did open-source data tooling.

10. Conclusion

In 2025, ETL development will be different from what it is currently. The ETL environment has changed because it no longer uses batch processing or solutions that are set up on-site. Instead, it focuses on cloud-native designs, real-time data streams, and tight guidelines on how to manage data. These days, ETL engineers have to be able to handle a number of various things. They should know a lot about both the business and the technical aspects of things. They should, for example, know how to safeguard data, use cloud services, optimize SQL, and orchestrate. They also need to be able to adjust how they talk to and work with each other. You need to continuously learn and get better at your job if you want to achieve success in this exciting area, as this case study and skill analysis show. Developers should always be learning about new tools and methods, such as AWS Glue and Apache Airflow, as well as new concepts, such as Infrastructure as Code (IaC), DataOps, and data observability. You also need to be able to talk to analysts and make sure that data pipelines are lawful and help the organization reach its goals. You also need to be able to work with people from various parts of the company. It's really vital these days to be honest, open-minded, and willing to work with others. Safe for the future ETL developers are also quite good at what they do and can accomplish a lot of different things. They make sure that the appropriate individuals get the right information at the right time. People are more likely to trust data when they know it's accurate, secure, and helpful. When data is both helpful for business and a legal issue, these producers become even more vital to society. Backend implementers are very important for business intelligence and data-driven innovation since they work together and employ a number of different technologies.

References

[1] Patel, Monika, and Dhiren B. Patel. "Progressive growth of ETL tools: A literature review of past to equip future." *Rising Threats in Expert Applications and Solutions: Proceedings of FICR-TEAS 2020* (2020): 389-398.

[2] Kiran, Neelakanta Sarvashiva, et al. "Danio rerio: A Promising Tool for Neurodegenerative Dysfunctions." *Animal Behavior in the Tropics: Vertebrates*. Singapore: Springer Nature Singapore, 2025. 47-67.

[3] Veluru, Sai Prasad. "Dynamic Loss Function Tuning via Meta-Gradient Search." *International Journal of Emerging Research in Engineering and Technology* 5.2 (2024): 18-27.

[4] Khan, Bilal, et al. "An Overview of ETL Techniques, Tools, Processes and Evaluations in Data Warehousing." *Journal on Big Data* 6 (2024).

[5] Datla, Lalith Sriram. "Optimizing REST API Reliability in Cloud-Based Insurance Platforms for Education and Healthcare Clients". *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 4, no. 3, Oct. 2023, pp. 50-59

[6] Allam, Hitesh. "Developer Portals and Golden Paths: Standardizing DevOps with Internal Platforms". *International Journal of AI, BigData, Computational and Management Studies*, vol. 5, no. 3, Oct. 2024, pp. 113-28

[7] Goldfedder, Jarrett. "Choosing an ETL tool." *Building a Data Integration Team: Skills, Requirements, and Solutions for Designing Integrations*. Berkeley, CA: Apress, 2020. 75-101.

[8] Mohammad, Abdul Jabbar. "Chrono-Behavioral Fingerprinting for Workforce Optimization". *International Journal of AI, BigData, Computational and Management Studies*, vol. 5, no. 3, Oct. 2024, pp. 91-101

[9] Bussa, Santhosh, and E. Hegde. "Evolution of Data Engineering in Modern Software Development." *Journal of Sustainable Solutions* 1.4 (2024): 116-130.

[10] Chaganti, Krishna. "Adversarial Attacks on AI-driven Cybersecurity Systems: A Taxonomy and Defense Strategies." *Authorea Preprints*.

[11] Balkishan Arugula. "Personalization in Ecommerce: Using AI and Data Analytics to Enhance Customer Experience". *Artificial Intelligence, Machine Learning, and Autonomous Systems*, vol. 7, Sept. 2023, pp. 14-39

[12] Goldfedder, Jarrett. *Building a Data Integration Team: Skills, Requirements, and Solutions for Designing Integrations*. Apress, 2020.

[13] Talakola, Swetha. "Microsoft Power BI Performance Optimization for Finance Applications". *American Journal of Autonomous Systems and Robotics Engineering*, vol. 3, June 2023, pp. 192-14

[14] Jani, Parth. "FHIR-to-Snowflake: Building Interoperable Healthcare Lakehouses Across State Exchanges." *International Journal of Emerging Research in Engineering and Technology* 4.3 (2023): 44-52.

[15] Gurcan, Fatih, and Setenay Sevik. "Expertise roles and skills required by the software development industry." *2019 1st international informatics and software engineering conference (UBMYK)*. IEEE, 2019.

[16] Kupunarapu, Sujith Kumar. "AI-Driven Crew Scheduling and Workforce Management for Improved Railroad

Efficiency." *International Journal of Science And Engineering* 8 (2022): 30-37.

[17] Montandon, João Eduardo, et al. "What skills do IT companies look for in new developers? A study with Stack Overflow jobs." *Information and Software Technology* 129 (2021): 106429.

[18] Vasanta Kumar Tarra. "Claims Processing & Fraud Detection With AI in Salesforce". *JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING (JRTCSE)*, vol. 11, no. 2, Oct. 2023, pp. 37–53.

[19] Steffen, Don. "Setting the Standard for ETL Unit Testing." *Information Management* 19.8 (2009): 41.

[20] Syed, Mehdi, Ali Asghar, and Shujat Ali. "Kubernetes and AWS Lambda for Serverless Computing: Optimizing Cost and Performance Using Kubernetes in a Hybrid Serverless Model." *International Journal of Emerging Trends in Computer Science and Information Technology* 5.4 (2024): 50-60.

[21] Mali, Nilesh, and Sachin Bojewar. "A survey of ETL tools." *International Journal of Computer Techniques* 2.5 (2015): 20-27.

[22] Lalith Sriram Datla. "Cloud Costs in Healthcare: Practical Approaches With Lifecycle Policies, Tagging, and Usage Reporting". *American Journal of Cognitive Computing and AI Systems*, vol. 8, Oct. 2024, pp. 44-66

[23] Casters, Matt, Roland Bouman, and Jos Van Dongen. *Pentaho Kettle solutions: building open source ETL solutions with Pentaho Data Integration*. John Wiley & Sons, 2010.

[24] Balkishan Arugula. "Building Scalable Ecommerce Platforms: Microservices and Cloud-Native Approaches". *Journal of Artificial Intelligence & Machine Learning Studies*, vol. 8, Aug. 2024, pp. 42-74

[25] Machireddy, Jeshwanth Reddy. "Data quality management and performance optimization for enterprise-scale etl pipelines in modern analytical ecosystems." *Journal of Data Science, Predictive Analytics, and Big Data Applications* 8.7 (2023): 1-26.

[26] Talakola, Swetha. "Automated End to End Testing With Playwright for React Applications". *International Journal of Emerging Research in Engineering and Technology*, vol. 5, no. 1, Mar. 2024, pp. 38-47

[27] Jani, Parth. "Generative AI in Member Portals for Benefits Explanation and Claims Walkthroughs". *International Journal of Emerging Trends in Computer Science and Information Technology*, vol. 5, no. 1, Mar. 2024, pp. 52-60

[28] Abdul Jabbar Mohammad. "Dynamic Timekeeping Systems for Multi-Role and Cross-Function Employees". *Journal of Artificial Intelligence & Machine Learning Studies*, vol. 6, Oct. 2022, pp. 1-27

[29] Debortoli, S., O. Müller, and J. Vom Brocke. "Comparing Business Intelligence and Big Data Skills: A Text Mining Study Using Job Advertisements. Business & Information Systems Engineering." *The International Journal of WIRTSCHAFTSINFORMATIK ISSN* (2014): 2363-7005.

[30] Allam, Hitesh. "Declarative Operations: GitOps in Large-Scale Production Systems." *International Journal of Emerging Trends in Computer Science and Information Technology* 4.2 (2023): 68-77.

[31] Tarra, Vasanta Kumar. "Telematics & IoT-Driven Insurance With AI in Salesforce". *International Journal of AI, BigData, Computational and Management Studies*, vol. 5, no. 3, Oct. 2024, pp. 72-80

[32] 32. Ogunsola, Kolade Olusola, Emmanuel Damilare Balogun, and Adebanji Samuel Ogunmokun. "Developing an automated ETL pipeline model for enhanced data quality and governance in analytics." *International Journal of Multidisciplinary Research and Growth Evaluation* 3.1 (2022): 791-796.

[33] Frampton, Michael. "ETL with Hadoop." *Big Data Made Easy: A Working Guide to the Complete Hadoop Toolset*. Berkeley, CA: Apress, 2014. 291-323.

[34] Veluru, Sai Prasad, and Swetha Talakola. "Continuous Intelligence: Architecting Real-Time AI Systems With Flink and MLOps". *American Journal of Autonomous Systems and Robotics Engineering*, vol. 3, Sept. 2023, pp. 215-42

[35] Chaganti, Krishna Chaitanya. "AI-Powered Patch Management: Reducing Vulnerabilities in Operating Systems." *International Journal of Science And Engineering* 10 (2024): 89-97.

[36] Sangaraju, Varun Varma, et al. "REVIEW ON FOG COMPUTING–APPLICATIONS, SECURITY, AND SOLUTIONS." *Proceedings on Engineering* 7.1 (2025): 447-458.

[37] Entwistle, Noel. "Concepts and conceptual frameworks underpinning the ETL project." *Occasional report* 3 (2003): 3-4.

[38] R. Daruvuri, K. K. Patibandla, and P. Mannem, "Data Driven Retail Price Optimization Using XGBoost and Predictive Modeling", in *Proc. 2025 International Conference on Intelligent Computing and Control Systems (ICICCS), Chennai, India. 2025*, pp. 838–843.