Fivetran

A guide to modern database replication

Real-time data replication is a key aspect of enterprise data management to accelerating time to insights for analytic use cases, ensuring business continuity and more.

Learn about common database replication strategies, how to choose the right one for your business and best practices for implementation.

Table of contents

Introduction

Modern enterprises have growing workloads from applications that run critical processes and store important pieces of data. However, traditional on-premises servers are often unable to efficiently handle heightened workloads. These inefficiencies can affect your company's ability to scale processes and perform real-time analytics, transformations and visualizations on your data.

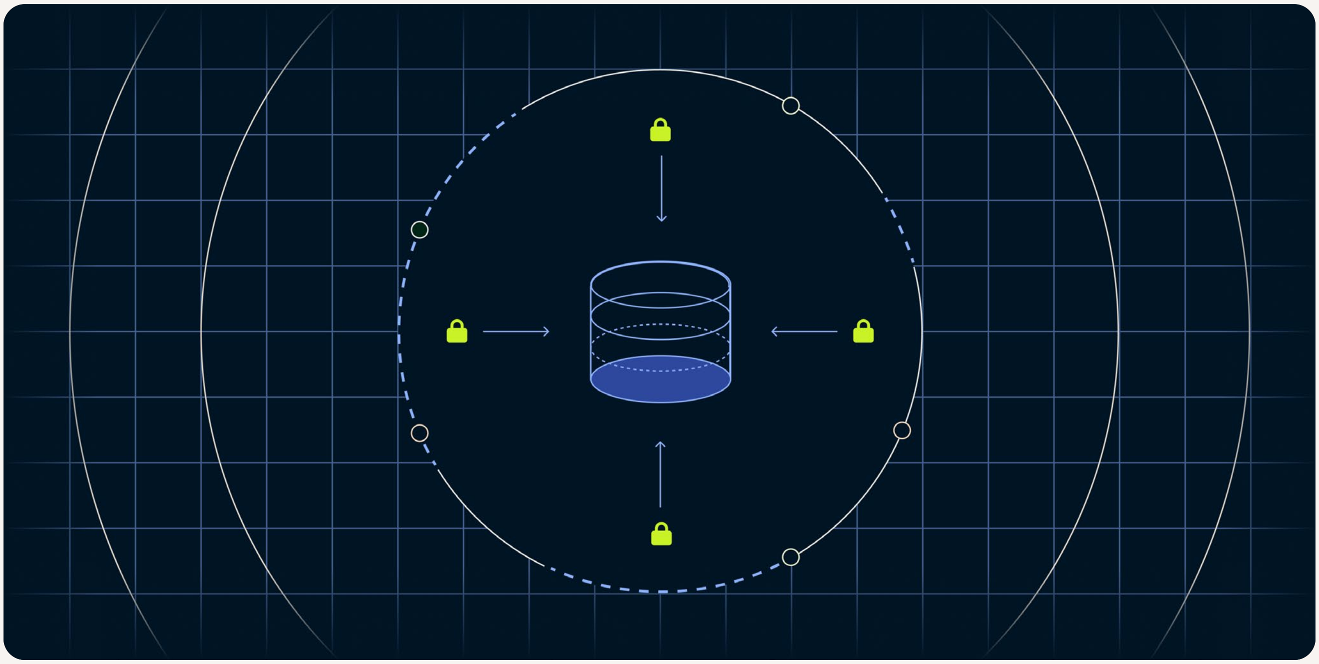
This is where database replication comes in — the process of creating and storing database copies in multiple locations, such as your cloud data warehouse, data lake or other databases. You can shift workloads and data storage from your database to a cloud environment. Gartner[1] predicts that 65 percent of application workloads will be optimal or ready for cloud delivery by 2027.

With typical IT organizations spending more than 30 percent of their budget on infrastructure, shifting to the cloud can save organizations anywhere from 10-20 percent[2] of their annual IT budget — savings that can be reinvested in growth and innovation.

In this guide, we will discuss common database replication strategies, how to choose the right one for your business and best practices for implementation.

[1] Gartner says four trends are shaping the future of cloud, data center and edge infrastructure

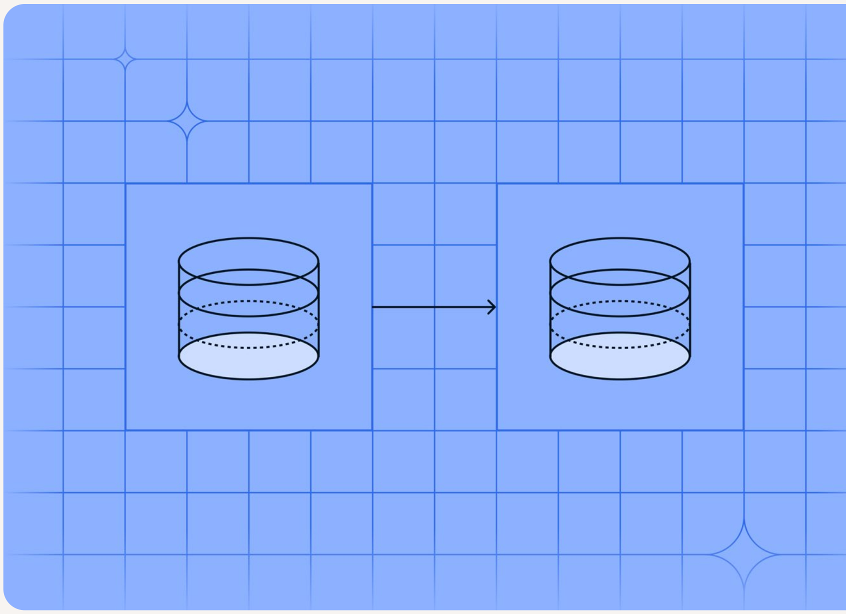
[2] KPMG: Cloud economics: Making the business case for cloud



What is database replication?

Database replication is the process of creating copies of a database and storing them across various on-premises or cloud destinations. This ensures every authorized user connected to the system can access copies of the same (up-to-date) data, improving data availability and accessibility.

Database replication is an ongoing process. If a user accesses and changes data in the source database, those changes are synced to the replicated databases. This ensures users are always working with the latest and most accurate data.



There are two types of database replication:

1. **Active/active replication:** Every system can process changes and the databases are synced in all directions. Some preparation is generally required for active/active replication applications to avoid conflicts and this setup is generally used for load balancing or high availability scenarios.
2. **Read-only replication:** Replicated databases only receive changes from the primary database. Users can read but not modify any data. Read-only databases are used for data democratization giving users access to data without the ability to make changes.

How is database replication different from data replication?

Data replication refers to the data you're replicating. It's what you're copying over to your data warehouse or data lake. Examples can include structured data like customer information and operational data produced by your organization from various applications.

In some cases, it makes sense to replicate specific types of data. As an example, if you operate a chain of stores, you might replicate data from your transactional database or a CRM application to a cloud destination where it can be analyzed and mined for useful insights.

Database replication involves making an identical copy of a table or whole database that gives organizations the full context for analytics. It can also support operations – for instance, ensuring that servers that support operations in different regions are synchronized with each other, or using redundant copies of a database for disaster recovery or as failover instances.

Databases can be replicated once, in scheduled batches or continuously.

- A **one-time replication** is usually performed to migrate data from one platform to another. For example, an organization must perform a one-time replication when they migrate data from an on-premises environment to a cloud-based platform.
- **Scheduled batches** are the norm when some amount of latency is acceptable (i.e. daily reporting).
- **Continuous replication** is critical for time-sensitive operational uses.

Despite the importance of database replication, it can be extremely challenging to move data in a performant, reliable and secure manner. Database replication allows two operational databases in different locales to perfectly mirror one another or provide an analytics environment for accurate, real-time insights.

Until recently, many organizations had no choice but to homebrew data pipelines to replicate databases between different platforms or simply go without. In the present day, these challenges can be addressed through fully managed data replication.

DID YOU KNOW?

Fivetran's fully managed database replication:

✓ **Optimizes** for the underlying storage mechanisms of databases

✓ **Captures** schema changes and historical data

✓ **Uses** insights gleaned from hundreds of customers to optimize the performance of database connectors

What are the benefits of database replication?

Database replication involves storing copies of your database across various databases, data warehouses and data lakes. Here's how this practice can benefit your company.

Improved disaster recovery

Relying on a single source database leaves your company vulnerable, as any malfunctions or downtime can prevent access to critical data and disrupt your workflows.

Replicating multiple copies of your database can power a high availability environment and ensure your data is always easily accessible. In the event of a natural or technological disaster causing a data center to go offline, users still have access to their data via a database in an unaffected region and are able to resume operations with minimal disruptions.

LEARN MORE

For more than **90% of mid-sized and large enterprises**, the cost of downtime exceeds **$300,000 an hour.**

Source: ITIC 2021 Hourly Cost of Downtime survey

Lower data latency

If you have a database hosted in North America, users trying to access the data from a region like Europe or Asia may experience delays. High data latency can have a direct impact on their work.

Database replication allows for localized data access to reduce latency and improve data democratization. With databases replicated in multiple locations, users are able to access the database that's closest to them.

Reduced server load

Network performance can take a hit when capacity on the database server is lower due to significant data storage or CPU processing. With replication from your database to your destination, you can free space on your production database to keep performance levels optimal. Database replication allows for efficient data processing while maintaining integral and accurate historical data for audit or analytics purposes.

Challenges of database replication

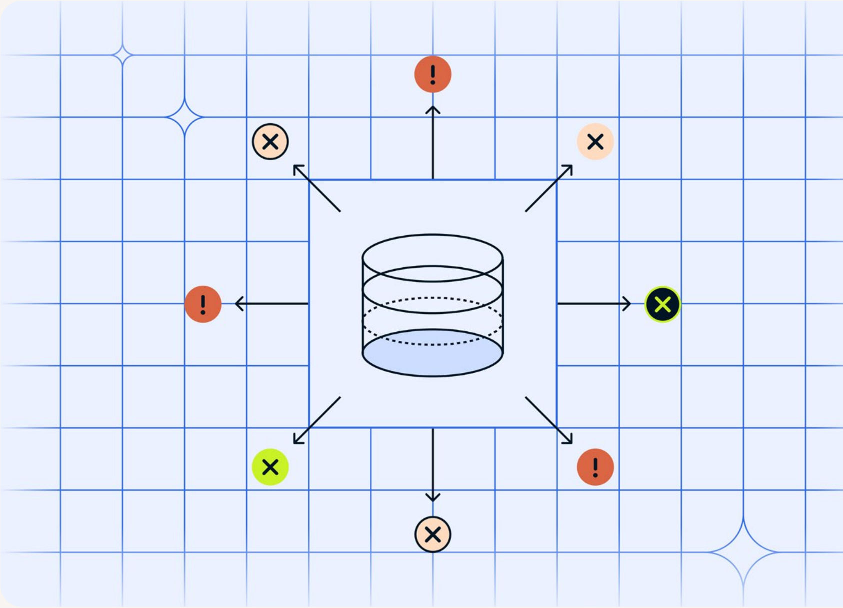
Database replication poses several challenges – ensuring data consistency, managing movement between multiple points of origin and destinations, and addressing complex technical requirements.

Ensuring data consistency: Keeping databases and their replicates consistent across all locations can be a challenge. Poor data governance, fragile manual data pipelines and ineffective use of change data capture can lead to issues with data accuracy and integrity between your source database and destination. Data loss is also possible if database objects are incorrectly configured or if the primary key used to verify data integrity is malfunctioning or incorrect.

Managing multiple sources and destinations: Managing multiple servers and destinations to store data from your database and its replicates requires additional resources. This can be a time-consuming and resource intensive operation and requires an analysis of the value that your database replication provides.

Addressing technical requirements: When initially setting up database replication, you must hand-pick and provision a wide range of tools, including both software and hardware, to ensure that you can sustain your required level of data processing. Meeting these technical requirements can require a complex and costly setup.

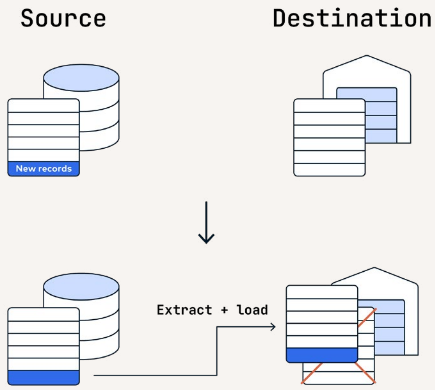
The rise of cloud-based data platforms sidesteps these issues by allowing you to outsource database replication to third parties, who can scale resources up and down as needed. Fully managed services obviate the need to provision your own cloud infrastructure (or worse, build and maintain an on-premises data center).



In most cases, a straightforward way to overcome these challenges is to outsource and automate database replication by engaging the services of a data movement provider. Whether through a provider or home-brewed, successfully implementing database replication requires several important technical considerations and best practices.

Types of database replication: Full vs. incremental

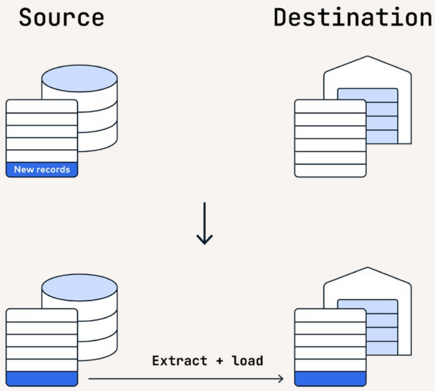
Full-table replication copies every piece of data within a table from the database to the cloud destination; this includes new, existing and updated data.



✓ **Advantages:** Because this replicates the entire table, you will always have the correct data set after each sync and can ensure that all inserts, updates and deletes are captured.

✗ **Disadvantages:** This is the least efficient type of database replication and rather resource intensive as you are copying every piece of data within a table whether it has changed or not. This can also lead to overloading your destination causing it to crash, depending on the size and volume of data within the tables.

Incremental replication involves identifying and replicating new or updated records based on either a unique identifier or log entry.



✓ **Advantages:** It is the most efficient approach to database replication, as it only transfers updates rather than full datasets.

✗ **Disadvantages:** It can be much more technically difficult than full-table replication and depends on whether the data source makes logs or timestamps and other keys available.

This is the preferred approach to database replication after an initial sync, and is considerably more efficient than copying and loading full tables.

However, incremental replication can be far more technically difficult than full replication. It can be difficult to comprehensively identify the changes made at the source. Incremental database replication requires change data capture, a set of methods that detect and replicate changes at a source.

Methods of change data capture

Change data capture (CDC) is commonly used for replication from databases and processing data from various data sources, such as SaaS applications or other systems only accessible through APIs.

Real-time CDC enables data to be transferred in as changes happen. That makes it invaluable for organizations operating critical systems 24/7 with no convenient time for batch windows, especially where the impact on database processing directly hurts the primary business process. Because critical systems contain the most important data to help drive decisions, real-time access to this data is essential.

There are multiple methods for CDC. They are usually chosen based on the constraints imposed by the database that is acting as a source:

Log-based change data capture: Relational databases usually have transaction logs that record every modification performed on the data. In log-based change data capture, a log reader parses the transaction log directly to determine changes that have been made to the database. This database replication method allows all changes to be captured including inserts, updates and deletes due to the replication tool or solution having access to the database's logs. Log-based change data capture is comprehensive, enables a very high volume of throughput and is the preferred approach for performant database replication.

CHECK THIS OUT

Oldcastle Infrastructure, Pitney Bowes and JetBlue are Fivetran customers who leverage log-based CDC to drive real-time analytics.

jetBlue pitney bowes

Oldcastle Infrastructure™
A CRH COMPANY

Read the case studies

Timestamp-based change data capture: Changes are extracted via a mark of the most recent extraction time and replicating every item in the database with a timestamp from that point forward. This will replicate inserts and updates but does not detect when a row in the database has been deleted.

Trigger-based change data capture: Every insert, update and delete operation at the source database not only performs its respective change, but also triggers the recording of the change in a separate change table that can then be replicated to the destination. This is a resource-intensive database replication method and could potentially require significant amounts of storage depending on the frequency of changes.

Difference-based change data capture: This is a brute force comparison of all data within the database and the destination, where a compressed snapshot is used to compare the two systems and identify changes to each. This method captures inserts, updates and deletes but only runs optimally with smaller data sets.

Best practices for database replication

1 Don’t copy and paste, use incremental updates

Change data capture is arguably the single most important best practice for database replication. With the explosion in the volume, velocity and variety of modern corporate datasets – to the tune of hundreds of terabytes, the need for real-time analytics and dozens of data sources, respectively – it simply isn’t practical to copy and paste the contents of an entire database every time an update is required.

Without change data capture, none of the analytical or operational uses for database replication are possible except at very small volumes of data.

CUSTOMER SPOTLIGHT

Lufthansa

Lufthansa Systems, a division of Lufthansa Airlines, is one of the world’s leading providers of IT services in the airline industry. It serves roughly 300 national and international airlines comprising more than one-third of all airlines worldwide.

CHALLENGE: Lufthansa Systems needed to synchronize data about weather, air traffic, flight schedules and more between a central repository and hundreds of customer data warehouses.

SOLUTION: Lufthansa Systems uses Fivetran to provide fast bi-directional replication between its central data repository, customer data warehouses and backup instances. As a result, Lufthansa Systems can ensure that its airline customers have the optimized flight plans they need, when they need them.

2 Be alert to changing schemas

Schemas drift when an application's data model evolves and its columns, tables and data types change. Faithfully preserving the original values of data and ensuring its smooth passage from source to destination, even as the source schema changes, is a particularly vexing data movement challenge. Schema changes are one of the chief failure conditions for traditional extract, transform, load (ETL), often causing the data pipeline to fail and require an adjustment that requires scarce and highly specialized expertise.

Schema drift handling can be accomplished in two ways: net-additive data movement and live updating. Net additive data movement behaves like so:

- When a column or table is added to the source schema, it is added at the destination as well.
- When a column or table is removed, it is kept in the destination but no longer updated.
- When a column or table is renamed, it is duplicated at the destination under its new name, so that both the old version with the old name and the new version with the new name are present. The old version is subsequently no longer updated but retained for the sake of completeness.

By contrast, live updating perfectly replicates all additions, deletions and edits at the source in the destination. This extends to individual records as well. Rows that are deleted at the source are deleted at the destination.

Another consideration is preserving historical values of a given record over time. One method is through "history tables" that maintain "start" and "end" timestamps for every version of a record.

DID YOU KNOW?

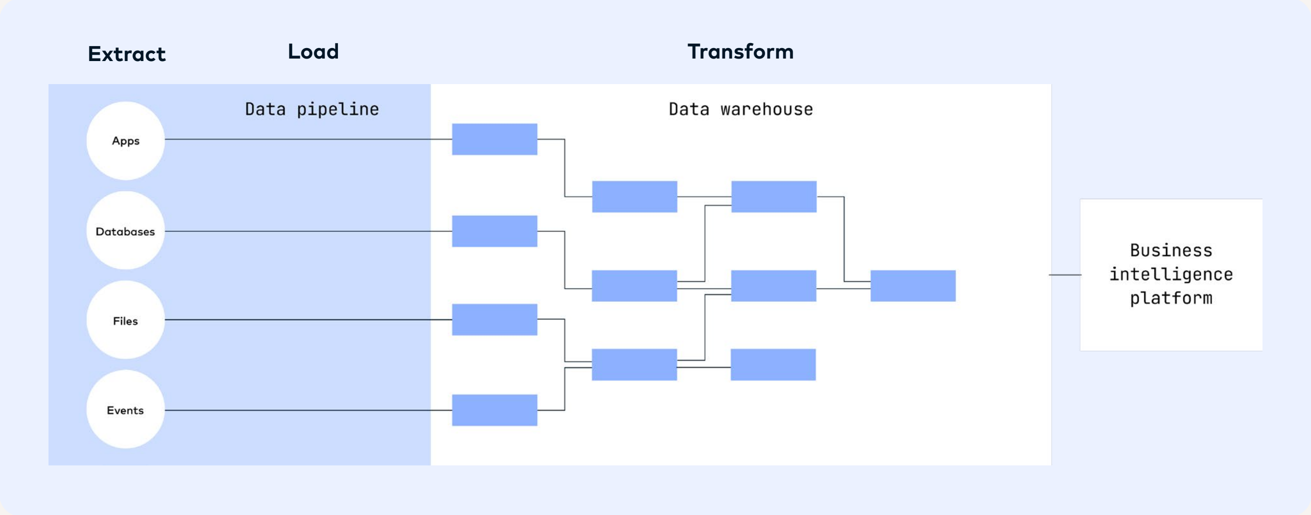
Fivetran uses live updating to ensure perfect replication between source and destination.

3 You need an ELT architecture

When it comes to data movement, there are two architectures – ELT (Extract, Load, Transform) and the more outdated ETL (Extract, Transform, Load). ELT has a number of clear advantages over ETL that become more pronounced as an organization's data sets grow in size:

1. **Simplifying data replication** – In ELT, the data is moved directly from the source to the destination to preserve the granularity and quality of the original data. There are no in-transit transformations.

2. **Lower failure rates** – By performing transformations in the destination, changes to both upstream schemas and downstream business requirements no longer cause extractions and loads to fail, requiring rebuilds of the data pipeline.
3. **Enables automation** – By decoupling extraction and loading from transformations, the data pipeline can produce a standardized output. This eliminates the need to constantly build and maintain pipelines to output custom data models. It also allows derivative products, like templated analytics, to be produced and layered on top of the destination.
4. **Enables outsourcing** – Since the ELT pipeline can produce standardized outputs, it's easier to outsource your data integration to third parties.
5. **Flexible scaling** – Organizational data needs can quickly change based on prevailing conditions. When data processing loads increase, a cloud-based ELT platform can quickly provision additional resources.
6. **SQL transformation support** – ELT shifts transformation from an engineering-intensive process that requires careful scripting to one that is performed in the destination by analysts. Transformations can be written in SQL rather than scripting languages such as Python.



4 Choose an appropriate database connection method

There are several ways to connect a database to a pipeline and a destination:

- Safelisting your data pipeline's IP address
- SSH and reverse SSH tunnels
- PrivateLink between platforms on virtual private clouds
- Using a VPN tunnel

All of these methods ensure that third parties cannot observe data that is in transit. Fivetran supports all of the listed connection methods.

5 Transformations are essential to analytics

Data transformation is the process of revising, computing, separating and combining raw data into analysis-ready data models. Transformation prepares data for a range of important needs:

- **Analytics** – Your operational databases almost certainly have schemas suited for efficiently supporting operations rather than analytics. Once the data has arrived in your destination, you will need to transform it to support analytics instead.
- **Machine learning** – Transforming data to support machine learning models is called feature engineering and consists of turning raw data into numerical or categorical fields that algorithms can interpret.
- **Regulatory compliance** – Needlessly storing personal identifiable information (PII) leaves sensitive data vulnerable to a range of accidental and malicious data breaches. Transformations may be used to aggregate PII or block or hash it.
- **Other operational uses** – Data may be "activated," sent from an analytics platform like a data warehouse back into operational systems to support business activities.

In an ELT architecture, transformations are performed after data has arrived at its destination, rather than in transit. There are several reasons this is highly beneficial:

- It decouples extraction and loading from transformation, simplifying the data pipeline and making it generally faster and more efficient
- It leverages the scalability of a cloud data warehouse
- Most pertinently to database replication, when it is combined with CDC, it is possible to orchestrate transformations to trigger based on incremental updates, facilitating real-time analytics

WHAT ELSE TO KNOW

Fivetran enables orchestration after the data arrives in the destination through two key features:

1. Integrated scheduling automatically schedules and runs transformations in an automated, path-dependent manner as connectors finish syncs. This ensures data freshness, reduces latency and prevents unnecessary syncs, saving compute costs.
2. Data lineage graphs make it easy to track the flow of data, monitor performance and debug models in the pipeline.

6 Make sure your databases and pipelines are secure

In the context of database replication, security is all about preventing unauthorized access to pipelines and the uncontrolled exposure of data.

There are two ways to address security from a technological standpoint:

1. Data security, regarding the protection of data in and after transit
2. Platform security, regarding tools

Data security

Data security is fundamentally about protecting sensitive data such as personally identifiable information (PII). In the context of database replication, this means applying data security in the pipeline before data is loaded to the customer's destination through column masking, which takes two main forms:

1. **Blocking** data by excluding it from entering the destination altogether.
2. **Hashing** data by anonymizing and obscuring it while preserving its analytical value.

Platform security

Other security considerations have more to do with general access to the platform and include:

1. **End-to-end encryption**, ensuring data is encrypted while in transit and that all communication is conducted through a platform-specific PrivateLink, VPN or SSH.
2. **Deployment methods**, including region- and cloud-specific data residency.

SECURITY FEATURES

Fivetran offers:

- ✓ Column masking
- ✓ End-to-end encryption of data and credentials

Deployment options available through Fivetran include:

- ✓ Major cloud environments (AWS GovCloud, Azure and GCP)
- ✓ Over 20 regions across North America, Europe, Asia and the Pacific
- ✓ On-premises deployment for highly sensitive data

7 Governed data movement keeps database replication manageable

Governed data movement prevents unauthorized access to databases and pipelines, limiting the potential for alterations and changes to databases to lead to data conflicts and desynchronization. In the context of database replication, governance consists largely of maintaining access control over data before, during and after transit.

Data governance can be broken into three precepts:

1. Observing data – visualizing, knowing and understanding data and how it is handled
2. Controlling data – securing company, customer and employee data from unauthorized exposure
3. Scaling data programs – providing more data services without compromising compliance

Observing data

Security and legal teams need to audit data and monitor the access and handling of sensitive information. Analysts and decision makers need to understand the provenance of their data and what questions can and can't be answered. Data teams need to implement and enforce required security policies and also ensure that they are meeting their obligations to analysts and other stakeholders.

The following practices bring full observability into the data pipeline for data audits and use monitoring:

- Data lineage
- End-to-end audit trails logging all access, behaviors and changes to data in-flight
- Real-time metadata capture and logging of
    - Keys
    - Tables
    - Columns
    - Data types

IMPORTANT NOTE

Fivetran exposes all relevant metadata for import into dedicated catalog tools, such as Atlan, data.world, Alation and Collibra.

Controlling data

Analysts depend entirely on access to data in order to support business stakeholders and perform their roles. Data teams are the main gatekeepers to analysts, managing approval workflows for interested parties. Highly regulated data requires complex configuration in order to meet security stipulations.

Controlling data is about managing access and proper handling of sensitive information.

Scaling data programs

In order to serve the competitive needs of the business, data teams need a way to safely and securely scale their data programs without compromising on compliance. Leveraging automation and standardization enables data teams to onboard new users and assign appropriate data access at scale. This obviates the need to manually create accounts and configure permissions, which can otherwise be prone to human error and delay.

CHECK THIS OUT

Fivetran offers special features for automating and standardizing data onboarding processes:

- ✓ Automatic tagging and categorizing of data, including PII
- ✓ Team assigned access control
- ✓ Automated, centralized user provisioning with granular permissions based on roles
- ✓ Programmatic configuration and troubleshooting via Terraform and Airflow providers

Getting started with database replication

Now that we have covered a number of important considerations around database replication, we can discuss concrete steps to properly implement a database replication process. Here are the database replication steps:

1 Identify your data source

The first step is to identify your primary data source where data from your organization originates. This could be any kind of database, on-premises or in the cloud. Next, determine the destination you'll replicate the data to. Potential destinations are major cloud data warehouses or data lakes for analytics or another database for operations.

2 Determine the scope of your database replication

The next step is to consider the data you need to replicate from your database.

If you need to replicate an entire database, you should opt for a full-table database replication scheme. This ensures that all of your data is available in your destination. However, if you only need certain aspects of a database replicated (e.g., analytical data), you would designate specific source tables and columns to replicate.

3 Decide on a database replication frequency

How often do you need the data replicated? Updates in real time are typically used for time-sensitive transactional applications. This approach uses more bandwidth, but ensures that data across the network is synchronized when it counts.

Longer cycles, ranging from every few minutes or hours to daily, are more cost-effective. However, there may be intervals where important data is inaccessible. Decide on a database replication frequency that fits your business needs.

4 Choose a database replication type and method

Decide on your database replication type: full or incremental. Do you need access to historical data? If so, you will need to initially make a full sync, and subsequently sync incrementally. Otherwise, you may sync incrementally to begin with.

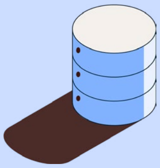
You will then need to determine what method of database replication you will set up: log-based CDC, trigger-based CDC, timestamp-based CDC or difference-based CDC. Each of these methods has their pros and cons. The choice will depend heavily on your access to the database's logs and tolerance for soft- vs hard-deletes.

5 Use a fully managed database replication tool

Database replication improves the availability of your data by storing it in multiple locations and potentially reducing the load on your source database. To ensure your data is properly replicated, you'll need to select the right database replication tool for your use case. This will keep your systems running smoothly and ensure you can get the greatest value out of your data.

Fivetran's automated data movement platform is the leading tool for enterprises to solve their database replication needs. Our comprehensive support for a wide range of database sources and destinations, multiple replication types and methods allows companies to replicate even the largest volumes of data.

Our high-volume replication solution uses log-based CDC to support large volumes of data and minimize replication latency, ensuring your analytics are run with the freshest of data.



To experience the power of automated database replication for yourself, request a demo.

Fivetran

Fivetran automates data movement out of, into and across cloud data platforms. We automate the most time-consuming parts of the ELT process from extract to schema drift handling to transformations, so data engineers can focus on higher-impact projects with total pipeline peace of mind.

With 99.9% uptime and self-healing pipelines, Fivetran enables hundreds of leading brands across the globe, including Autodesk, Conagra Brands, JetBlue, Lionsgate, Morgan Stanley, and Ziff Davis, to accelerate data-driven decisions and drive business growth.

Fivetran is headquartered in Oakland, California, with offices around the world.

For more info, visit Fivetran.com.