DATABASE AUTOMATION EXPLAINED: CONCEPTS & BEST PRACTICES

Databases—spreadsheets containing data—are useful tools to store information. So useful, in fact, that they're integral to the operations of any modern organization.

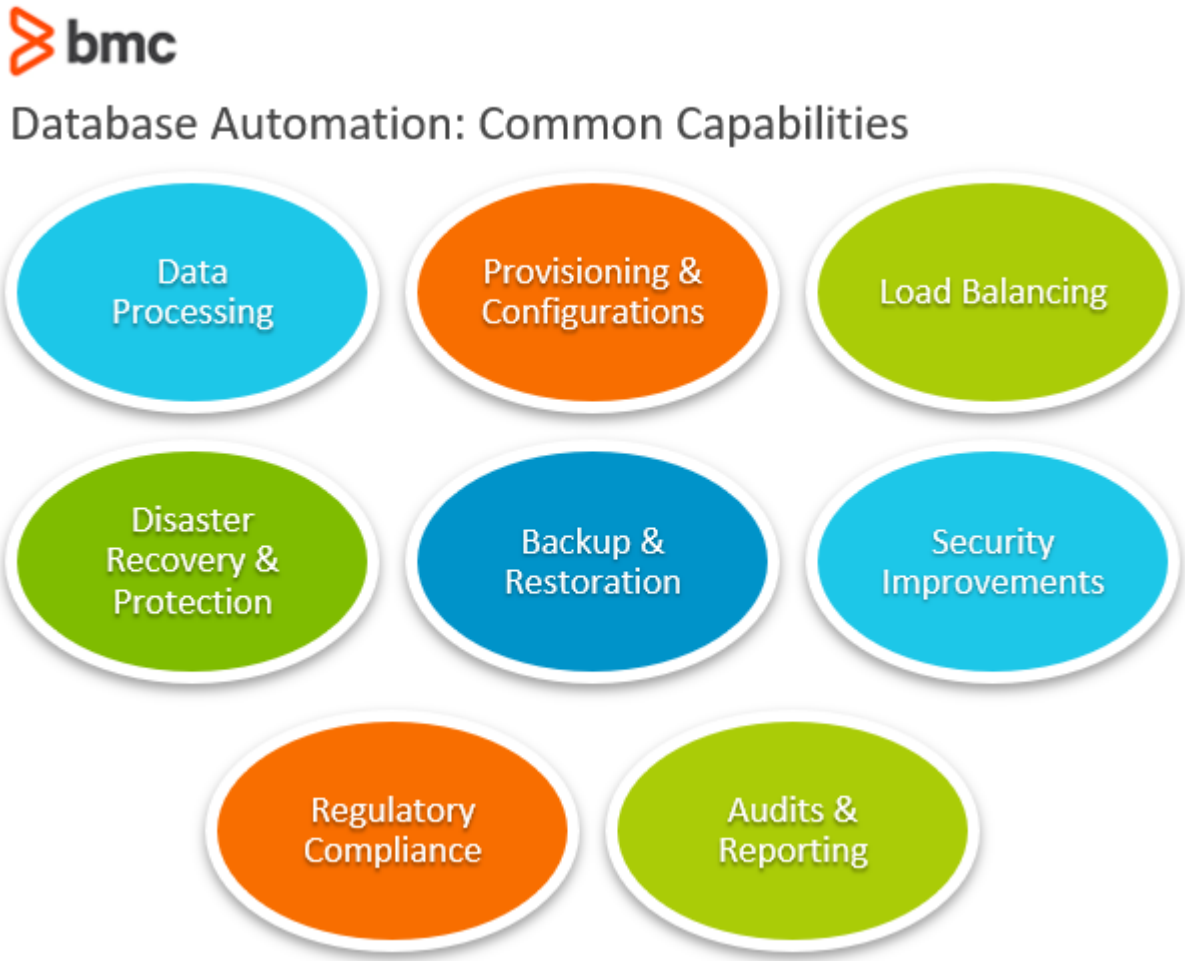
Databases often contain sensitive and profitable information assets on users, systems, and organizations. This information is maintained, processed, and updated through several complicated administrative operations assigned to a database administrator (DBA) who takes advantage of technology solutions such as database management system (DBMS) to perform these tasks.

Today, however, these operations are often predictable and repetitive, and performed at such a large scale that renders manual DBA skills insufficient. That's why more and more companies are turning to database automation: the process of automating database management operations.

In this article, we will discuss considerations and best practices for database automation.

How database automation works

Database automation tools offer a variety of purpose-built automation capabilities that apply to the DBMS and the associated infrastructure operations tasks. Here are the most common database automation capabilities.



Data processing

Data processing tasks includes automation data collection, replication, cleanup, and migration tasks that help make the data more meaningful, secure, dependable, and available for immediate processing when necessary.

Provisioning & configurations

Here, you can automate the setup of database environments and repositories for various tasks and stages during the software development lifecycle (SDLC). The provisioned database clusters should be secure, high performing, and dependable, according to the requirements of development, quality assurance (QA), and IT operations teams.

Load balancing

By automating load balancing, you optimize database performance in terms of:

- Throughput
- Latency
- Resource utilization

The workloads are balanced across servers in multi-cloud and hybrid infrastructure environments to optimize for the overall system performance, security, and cost investments.

Disaster recovery & protection

Losing critical information assets can compromise the ability of an organization to operate within regulatory guidelines and customer expectations. Therefore, organizations must introduce risk mitigation strategies such as:

- Database redundancy
- Distribution across geographic disparate server regions
- Automatic-trigger defense systems against compromise and cyber-attacks

Backup & restoration

The systems should be programmed to automatically backup and restore at specific instances in order to reduce the risk of data loss, especially in the event of a network intrusion that compromises the integrity of sensitive and mission-critical databases.

Security improvements

Deploying strong authentication and access management processes. Enrolling specific database systems to comply with specific organization policies for access privilege can be challenging and lead to several security issues unless appropriate security policies are enforced.

Regulatory compliance

Most organizations are subject to a variety of compliance requirements. For instance, the GDPR regulation requires anonymity of user data before engaging external partners without adequate user consent.

In this case, all database entries should be anonymized—perfect for automation—before a third-party vendor is allowed to process or utilize the data. Failure to comply can lead to:

- Steep fines
- Legal troubles
- Loss of trust in an increasingly privacy-aware user market

Audits & reporting

Use automation to monitor and track how databases and the information contained within changes. Auditing databases can help organizations understand how operations, systems, and data assets comply with organizational and regulatory policies.

Any discrepancy can be tracked and trigger immediate corrective measures before it impacts the business.

How to automate databases

While the lack of automation leaves organizations to follow a reactive approach to database management, DBAs and engineering teams should strategically approach database automation.

For instance, consider these factors *before* taking the automation route:

- **Waste processes.** Automating waste process only introduces new bugs, bottlenecks, and cascading waste processes.
- **Data inconsistencies.** Automation generalizes rules for information assets. If the data is inconsistent or deviates from expected structure, automation will introduce additional inconsistencies into the database.
- **Database tuning.** Parameters associated with dynamic environments should be carefully understood, investigated, tested, and benchmarked before applying it to the database. This process is especially challenging when a highly available cluster must be provisioned across multi-cloud and hybrid infrastructure environments.
- **Simple tasks.** It may be more efficient to perform a manual process instead of writing and testing an automation script for a simple database administration task.
- **Schema changes.** The process of database update, including dependencies, can change drastically during database automation if conflicts, configurations, permissions, and connectivity with the wider systems is not managed correctly.

In order to reduce the potential risk of automating databases that may contain anomalies and inconsistencies, it's important to devise and run extensive tests on the automation technology or script. Test early and often to identify the scope limitations and find new candidates for automation.

Another important consideration for database automation is to track and address dependencies associated with databases and the underlying infrastructure. Automation scripts that do not account for all dependencies may end up as a single point of failure, potentially leading to performance issues, security loopholes, and compliance violations.

Related reading

- BMC Workload Automation Blog
- An Introduction to Database Reliability
- CAP Theorem for Databases: Consistency, Availability & Partition Tolerance
- Data Storage Explained: Data Lake vs Warehouse vs Database
- What Is DBaaS? Database-as-a-Service Explained