Guide to enhancing privacy and addressing GDPR requirements with the Microsoft SQL platform

Learn from Microsoft's experience with Microsoft SQL-based technologies

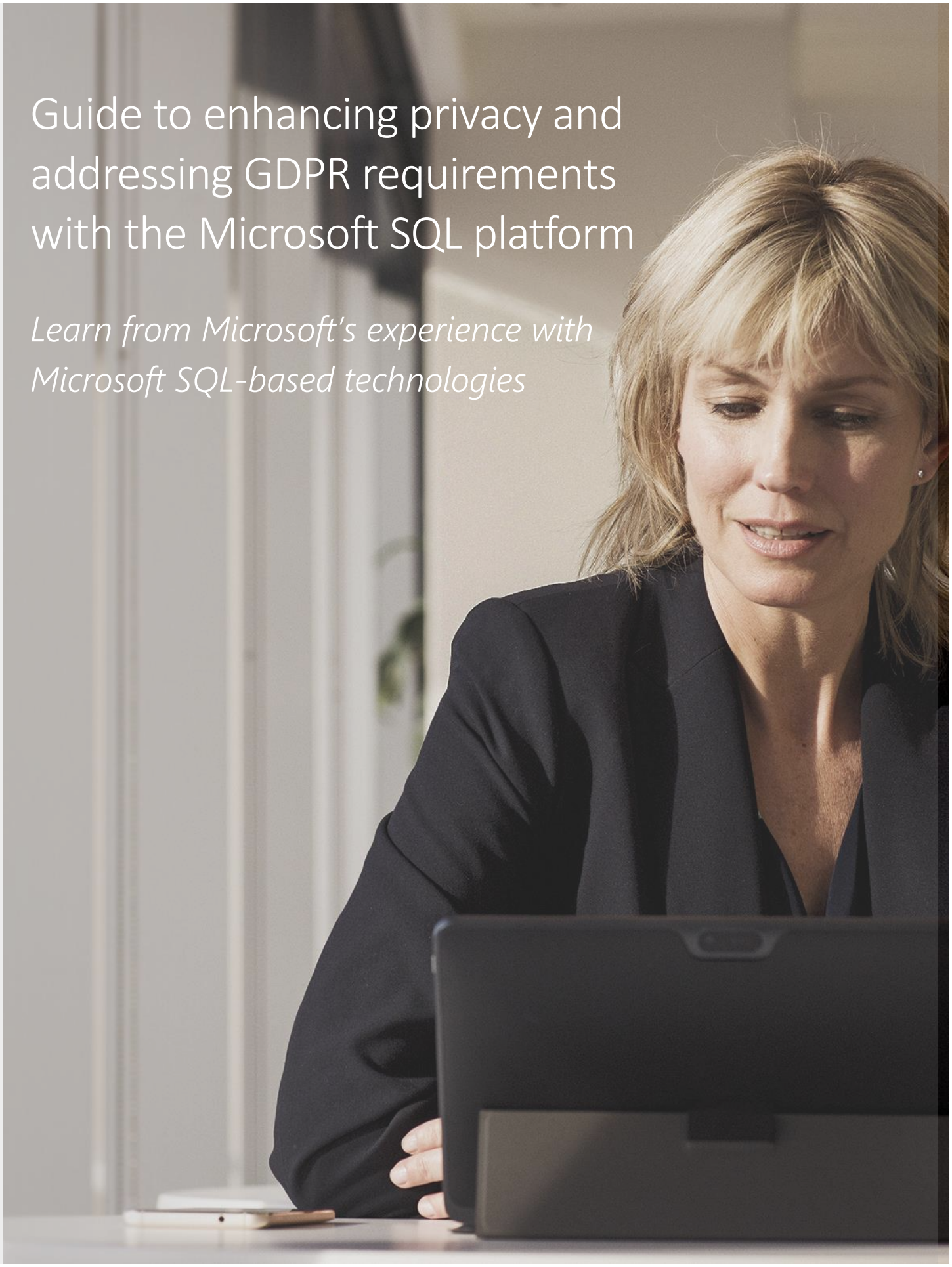


Table of Contents

Disclaimer

This white paper is a commentary on the EU General Data Protection Regulation (GDPR), as Microsoft interprets it, as of the date of publication. We've spent a lot of time with GDPR and like to think we've been thoughtful about its intent and meaning. But the application of GDPR is highly fact-specific, and not all aspects and interpretations of GDPR are well-settled.

As a result, this white paper is provided for informational purposes only and should not be relied upon as legal advice or to determine how GDPR might apply to you and your organization. We encourage you to work with a legally qualified professional to discuss GDPR, how it applies specifically to your organization, and how best to ensure compliance.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS WHITE PAPER. This white paper is provided "as-is". Information and views expressed in this white paper, including URL and other Internet website references, may change without notice.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this white paper for your internal, reference purposes only.

Published May 2017

Version 1.0

Abstract

In this age of digital transformation, protecting privacy and enhancing security have become top of mind. The upcoming EU General Data Protection Regulation (GDPR) sets a new bar for privacy rights, security, and compliance.

The GDPR mandates many requirements and obligations on organizations across the globe. Complying with this regulation will necessitate significant investments in data handling and data protection for a very large number of organizations.

Microsoft SQL customers who are subject to the GDPR, whether managing cloud-based or on-premises databases or both, will need to ensure that qualifying data in their database systems is aptly handled and protected according to GDPR principles. This means that many customers will need to review or modify their database management and data handling procedures, especially focusing on the security of data processing as stipulated in the GDPR.

Microsoft has significant experience in addressing data privacy principles and complying with complex regulations. Microsoft is committed to sharing this experience with customers, to help them achieve the privacy requirements of the GDPR with Microsoft SQL-based database and data warehouse products. With the most comprehensive set of compliance and security offerings of any cloud provider and a vast partner ecosystem, Microsoft is prepared to support customers' privacy and security initiatives now and in the future.

Introduction: The GDPR and its implications

On May 25, 2018, a European privacy law is due to take effect that sets a new global bar for privacy rights, security, and compliance.

The General Data Protection Regulation, or GDPR, is fundamentally about protecting and enabling the privacy rights of individuals. The GDPR establishes strict global privacy requirements governing how personal data is managed and protected, while respecting individual choice.

To achieve its objectives, the GDPR introduces several specific requirements related to the rights of individuals, such as the right to access their personal data, correct inaccuracies, erase data, object to processing of their data, and the right to obtain a copy of their data.

Personal data in scope of the regulation can include, but is not limited to, the following:

- Name
- Identification number
- Email address
- Online user identifier
- Social media posts
- Physical, physiological, or genetic information
- Medical information
- Location
- Bank details
- IP address
- Cookies

The GDPR also seeks to ensure personal data is protected no matter where it is sent, processed, or stored. As a result, data protection and security are key components in addressing the GDPR principles.

There are several obligations specifically introduced by the GDPR related to controls and security around the handling of personal data. The regulation obligates the data processor or controller to "implement appropriate technical and organizational measures" to address these. Some specific concepts include:

GDPR Article 25—"Data protection by design and default": Control exposure to personal data.

- Control accessibility—who is accessing data and how.
- Minimize data being processed in terms of amount of data collected, extent of processing, storage period, and accessibility.
- Include safeguards for control management integrated into processing.

GDPR Article 32—"Security of processing": Security mechanisms to protect personal data.

- Employ pseudonymization and encryption.
- Restore availability and access in the event of an incident.
- Provide a process for regularly testing and assessing effectiveness of security measures.

GDPR Article 33—"Notification of a personal data breach to the supervisory authority": Detect and notify of breach in a timely manner (72 hours).

- Detect breaches.
- Assess impact on and identification of personal data records concerned.

- Describe measures to address breach.

GDPR Article 30—"Records of processing activities": Log and monitor operations.

- Maintain an audit record of processing activities on personal data.
- Monitor access to processing systems.

GDPR Article 35—"Data protection impact assessment": Document risks and security measures.

- Describe processing operations, including their necessity and proportionality.
- Assess risks associated with processing.
- Apply measures to address risks and protect personal data, and demonstrate compliance with the GDPR.

It will be incumbent upon the organization to ensure that its entire IT environment complies with each of these principles and establishes appropriate measures.

Microsoft has outlined its commitment to the GDPR and its support for customers in the compliance process within the "Get GDPR compliant with the Microsoft Cloud" blog post by Chief Privacy Officer Brendon Lynch and the "Earning your trust with contractual commitments to the General Data Protection Regulation" blog post by Rich Sauer—Microsoft Corporate Vice President & Deputy General Counsel.

For additional information about the GDPR, Microsoft's commitments, and Microsoft's recommendations for beginning the compliance journey, please visit the GDPR section of the Microsoft Trust Center.

Journey to complying with the GDPR

A previously published Microsoft white paper, Beginning your General Data Protection Regulation (GDPR) Journey, introduces the GDPR, broadly describes how it will affect global organizations, and suggests ways to begin the compliance journey. The compliance process will span across the entire IT environment, including Microsoft SQL environments.

Microsoft recommends that organizations begin their journey to GDPR compliance by focusing on four key steps:

- **Discover**—identify what personal data is being managed and where it resides.
- **Manage**—govern how personal data is used and accessed.
- **Protect**—establish security controls to prevent, detect, and respond to vulnerabilities and data breaches.
- **Report**—keep required documentation, manage data requests, and provide breach notifications.

This document describes how these steps specifically apply to Microsoft SQL environments, whether on-premises, in the cloud, or hybrid environments. The paper also evaluates several existing and future-looking technological solutions built-in to Microsoft SQL-based technologies that can help organizations on their journey to GDPR compliance.

Data protection in Microsoft SQL to help address GDPR requirements

Microsoft SQL as a hub of private data and sensitive information

The database tends to be at the heart of the organization's IT infrastructure, where much of its data is stored and processed, and serves as a base from which data then flows throughout other systems of the organization. This means that using measures to protect the database and the data therein is fundamental to meeting the requirements of the GDPR. This is applicable for the entire range of Microsoft SQL-based technologies (Microsoft SQL), including Microsoft SQL Server (whether on-premises or hosted in a public cloud platform), Microsoft Analytics Platform System, Azure SQL Database, and Azure SQL Data Warehouse.

Within Microsoft, many technologies and services are built on Microsoft SQL-based technologies as the underlying data storage and processing layer. Therefore, many GDPR protection principles apply at the database level. Microsoft SQL offers many built-in security capabilities that can help reduce risks to data and improve the protection and manageability of data at the database level and beyond.

This document shares some of Microsoft's own approaches using Microsoft SQL for achieving the data privacy goals of the GDPR.

Four key steps to achieve GDPR compliance

Microsoft has significant experience addressing data privacy standards and principles across numerous products and services, many of which are built on top of Microsoft SQL-based technologies. Based on this experience, and the specific requirements outlined by the GDPR, Microsoft has established this methodology of focusing on four steps to achieving GDPR compliance: discover, control, protect, and report.

Discover

The first step is the discovery phase. In this step, it is important to locate systems storing sensitive data and identify which data qualifies as personal data according to the GDPR. A more detailed classification can also be performed according to data categories and levels of sensitivity.

It is then important to locate where exactly that data resides. *Which servers and/or databases contain personal data? Which columns or rows can be marked as containing personal data?* If there are multiple systems involved in storing or moving sensitive data, it is useful to create a map or data inventory indicating the locations of sensitive data in the database environment.

Another part of the discovery phase is understanding the attack surface area. *Who has access to what elements of data in the database system? What elements and features of the database system can be accessed and potentially exploited to gain access to the sensitive data?* It is helpful to create a map to visualize potential access to different resources.

Finally, while this document focuses on the Microsoft SQL-based data tier, it is also important to consider what happens beyond the borders of the database. *Where does the data go when it leaves*

the database? Here, the recommended approach is to review data flows and track data lineage—originating from the database and throughout the system. Microsoft teams perform this process by creating data flow diagrams for the relevant sensitive data for each potential access vector to the database as discovered in the surface area mapping.

Once these steps have been completed, it is possible to identify gaps with the privacy team to help achieve GDPR compliance.

Manage

Once the surface area and all data locations are understood, the next phase is to understand and limit what personal data is accessed, and by whom.

Ensuring the principle of least privilege is applied at the appropriate level of granularity is important here. Microsoft strongly recommends following best practices for highly secure authentication and centralized authorization management. More details on these best practices and how to implement these with existing Microsoft SQL technologies will be discussed later.

Protect

Once all sensitive data types are identified and current security practices understood, protection efforts can begin. These aim to reduce risk and minimize the impact to data through security and monitoring.

Protection requires proper security controls that are appropriate for the specific data types and usage scenarios. This includes various methods like data encryption at different levels (at rest, in transit, and in use), availability and resiliency mechanisms that prevent data loss, and auditing to continuously monitor activities taking place on the database.

An additional element of the protect phase is detecting and responding to data breaches involving personal data. Various capabilities are available in Microsoft SQL that can be used to help address these issues, as detailed later in this document.

Report

Proper reporting involves both transparency and creating and retaining records of all activities regarding personal data. These records are necessary to meet GDPR obligations to provide transparent information and communication with data subjects, as well as to have clear reports to meet the "Data protection impact assessment" requirement of the GDPR *(***GDPR Article 35***)*.

Microsoft keeps records of activities related to personal data for its own end users, and these records can also serve as evidence of compliance. Microsoft customers are encouraged to maintain similar records pertaining to personal data under their control.

Another element of this phase is to ensure all processes and procedures remain current and are being followed. Ongoing tracking and management of a data protection policy is important in reducing risk, and it is a key element of continuously adhering to data privacy principles and maintaining compliance.

Built-in Microsoft SQL technologies that can help address GDPR compliance

Microsoft SQL provides built-in solutions that can help in each of the recommended steps to achieving GDPR compliance.

Microsoft is making use of these built-in security capabilities internally to help many of its own products and services meet GDPR requirements. SQL Server on-premises (and SQL Server on Azure Virtual Machines), Azure SQL Database, Microsoft Analytics Platform System, and Azure SQL Data Warehouse have some shared and some unique security capabilities that can be applied for the different scenarios.

Customers can take advantage of Microsoft's experience in implementing these features and best practices in internal systems. The following examples taken from Microsoft's own use of built-in capabilities in Microsoft SQL may be beneficial for customers who are facing the challenges of meeting GDPR requirements.

Discovering and classifying personal data and its access vectors

Microsoft SQL provides mechanisms to help identify personal data. Initially, it is possible to query metadata, such as analyzing all column names via querying `sys.columns`, to identify column names which potentially contain personal data such as "Name", "Birthdate", "ID number", etc. These identified columns should be added into a data map for further analysis and to identify data flows throughout an organization.

For more advanced discovery capabilities, it is possible to use Full-Text Search in Microsoft SQL to search for keywords located within freeform text. Additionally, sensitive data can be tagged using Extended Properties to add sensitivity labels to relevant columns.

Many capabilities exist in Microsoft SQL today to help in completing this personal data discovery phase on database systems, and Microsoft is developing more capabilities to make this even easier for customers going forward (see the Looking Forward section, later).

Beyond discovering the personal data in the system, it is necessary to identify and understand the current data access policies being applied to that data. Furthermore, a security review is needed to map the attack surface area of the SQL database system by understanding what features and capabilities are enabled that can be used to gain access to data in non-obvious ways.

Generally, it is a recommended best practice to disable all features that are not in use to reduce the attack surface area. These can generally be disabled via T-SQL queries or via a management console like SQL Server Management Studio (SSMS). Example features that should be checked and, if possible, disabled include (but are not limited to): `XP_CMDSHELL`, `CLR`, `Filestream`, `Cross DB Ownership Chaining`, `OLE AUTOMATION`, `External Scripts`, `Ad-hoc Distributed Queries`, and disabling the `Trustworthy` bit. Additional recommendations are to disable network protocols that are not in use, turn off the SQL Server browser service and to uninstall sample databases.

For more information on security best practices that help protect data, please see the white paper SQL Server 2012 Security Best Practices - Operational and Administrative Tasks.

Managing access and controlling how data is used and accessed

Once personal data is located and gaps in policies of data governance are identified, control mechanisms should be implemented to minimize risks to that data from unauthorized access or loss.

The Microsoft SQL family of products provide multiple means of controlling access to the database and data at a granular level.

Fundamental mechanisms that address this are the Microsoft SQL built-in Authentication and Authorization mechanisms, which can help manage database permissions in a centralized and highly secure way.

Authentication in SQL Server

SQL Server Authentication helps ensure that only authorized users with valid credentials can access the database server.

SQL Server supports two authentication modes, Windows authentication mode and mixed mode. Windows authentication is often referred to as integrated security because this SQL Server security model is tightly integrated with Windows. Windows user and group accounts are trusted to log in to SQL Server, based on authentication credentials directly within Windows. Mixed mode supports authentication both by Windows and by SQL Server, using user names and passwords.

The use of Windows authentication on SQL Server is considered a best practice—it enables centralized management of SQL Server principals. (Azure Active Directory (Azure AD) authentication naturally extends the support for Windows authentication to cloud databases; see the following section on Azure AD for more details.)

Here are a few additional recommended best practices for SQL Server authentication:

Recommendation	Benefits
Use Windows authentication	• Enables centralized management of SQL Server principals via Active Directory. • Uses Kerberos security protocol to authenticate users. • Supports built-in password policy enforcement including complexity validation for strong passwords, password expiration, and account lockout.
Use separate accounts to authenticate users and applications	• Enables limiting the permissions granted to users and applications. • Reduces the risks of malicious activity such as SQL injection attacks.
Use contained database users	• Isolates the user or application account to a single database. • Improves performance, as contained database users authenticate directly to the database without an extra network hop to the master database. • Supports both SQL Server and Azure SQL Database, as well as Azure SQL Data Warehouse.

When Windows authentication is implemented in the SQL Server environment, permissions can be managed for individuals via Active Directory. Individuals or groups can be mapped to roles in the database, and assigned permissions according to specific needs of performing certain functions—such as connecting from certain applications. Since the roles are mapped to Windows users or groups, they can easily be managed in one central location.

Why is this important for the GDPR?

The GDPR talks about ensuring the security of personal data, "including for preventing unauthorized access to or use of personal data and the equipment used for the processing."

GDPR Recital 39.

Authorization

An important mechanism to manage access to data within the database is to define a proper authorization policy—including database role memberships and object-level permissions. This enables implementing a proper separation of duties model, where separate accounts/roles are defined for high-privileged operations, which are isolated from the roles used by applications or users to perform day-to-day tasks.

Access policies should abide by the principle of least privilege—which stipulates that users and applications always log in with the minimum privileges required to perform their task. Microsoft SQL-based technologies support this principle by providing mechanisms to define granular object-level permissions, and simplify the process by implementing role-based security. Granting permissions to roles rather than users simplifies security administration. Permissions assigned to roles are inherited by all members of the role, and users can be added or removed to a role to include them in a permission set.

It is a best practice to use server-level roles for managing server-level access and security, and database roles for managing database level access.

Role-based security provides the flexibility to define permissions at a high level of granularity in Microsoft SQL, thus greatly reducing the attack surface area of the database system.

Microsoft Case Study: Digital Crimes Unit team applies authorization best practices

The Microsoft Digital Crimes Unit (DCU) is responsible for safeguarding Microsoft customers and Microsoft assets from digital threats. It makes use of cutting-edge data analytics to fight cybercrime internationally.

The DCU team works to identify fraud on the Internet, which requires analyzing copious amounts of data including machine identifiers, personal user identifiers, and user account information. Several of these activities entail managing personal data in an Azure SQL Data Warehouse environment, and the DCU is fully committed to protecting the privacy of individual and enterprise data.

The team uses a variety of methods to control and secure its data warehouse environment—one of which is limiting access to a minimal set of authorized users. The team performs a continuous process of evaluating the existing set of permissions on its Azure SQL Data Warehouse, to better adhere to the best practice principle of least privilege.

The DCU team also implements role-based security by defining separate roles for each application or service type that connects to the data warehouse system, and ensuring that each of these have exactly the required permissions to support their tasks. This includes granting only READ and EXECUTE permissions on certain schemas or objects, for example, and no permissions at all to other database objects.

Azure SQL Database Firewall

When creating a database server in Azure, a firewall is automatically set up to help protect the data. The firewall initially prevents all access to the database server until explicit access permissions are specified, based on the originating IP address of each request. By default, there is an exception to allow all Azure services to connect to the database, to ensure full functionality across Azure services. However, there is an option to disable this exception in the firewall settings.

The following guidelines should be used to properly configure firewall settings for Azure SQL Database or Azure SQL Data Warehouse:

- In the "allowed IP addresses" entry field, enter the IP address range that allows connections to trusted users and services.
- Avoid using broad IP ranges as this defeats much of the protection provided by firewall rules.
- Follow the least privilege principle by restricting firewall settings for the IP address range to trusted IP addresses.

These guidelines ensure that only authorized connections have access to the database, and align with the GDPR requirements.

For more details on configuring Azure SQL Database firewall rules and best practices, please see this article on Azure SQL Database firewall rules.

Why is this important for the GDPR?

The GDPR specifically addresses the need for mechanisms which limit access to data, requiring "measures [that] shall ensure that by default personal data are not made accessible without the individual's intervention to an indefinite number of natural persons."

GDPR Article 25(2)—"Data protection by design and by default."

Authentication in Azure SQL Database using Azure Active Directory

As for authentication in Azure SQL Database, here too there are two types of authentication modes supported. Azure SQL Database supports SQL Authentication, which uses username and password to authenticate, and Azure Active Directory (AD) Authentication. Azure AD authentication uses identities managed by Azure Active Directory and is supported for managed and integrated domains.

Microsoft recommends using Azure AD authentication whenever possible as the more secure model. With Azure AD authentication, the identities of database and service users can be managed in one central location, which greatly simplifies permission management. There are also several additional benefits:

Recommendation	Benefits
Use Azure AD Authentication	• Reduces the proliferation of user identities across database servers. • Allows password rotation in a single place. • Enables managing database permissions using external (Azure AD) groups. • Enables integrated Windows authentication and other forms of authentication supported by Azure Active Directory. • Azure AD authentication uses contained database users to authenticate identities at the database level (considered more secure). • Supports token-based authentication for applications connecting to SQL Database. • Supports Active Directory Federation Services (ADFS) or native user/password authentication for a local Azure Active Directory without domain synchronization.

The use of federated authentication is also recommended, in which subscriptions are associated with the Azure Active Directory which is integrated with the on-premises Active Directory. This integration can make users more productive by providing a common identity for accessing both cloud and on-premises resources. This also greatly simplifies permissions management, as everything is centralized in the single integrated Active Directory.

To set up Azure Active Directory authentication for Azure SQL Database and Data Warehouse, use the following steps:

- Create an Azure AD and integrate with Active Directory (see the article Integrate your on-premises directories with Azure Active Directory for detailed instructions). Another option is to populate the Azure AD with users and groups.
- Create and provision an Azure AD administrator for Azure SQL Server.
- Configure client computers so that they have .NET 4.6 and the Azure Active Directory Authentication Library for SQL Server installed.
- Create contained database users in the database mapped to Azure AD identities.
- Connect to the database by using Azure AD identities or by using a token obtained from Azure Active Directory.

For further details on the process of setting up Azure Active Directory authentication, see the article Configure and manage Azure Active Directory authentication with SQL Database or SQL Data Warehouse. This tutorial can also help in getting started with the Azure Active Directory implementation.

This is helpful in addressing the GDPR requirements around managing access to

Microsoft Case Study: ISRM team enforces policies for database access management in Azure

The Microsoft Information Security and Risk Management (ISRM) team is responsible for overseeing the security for all internal Microsoft services. One of its primary charters is to define security control procedures that are designed to protect sensitive data, including personal data identified by the GDPR. These security procedures then lead to policies and best practices which are adopted by Microsoft internal teams when designing and implementing their services. These include requirements and controls for services using Azure SQL Database and Azure SQL Data Warehouse.

One of ISRM's most crucial policies states that: "Access to SQL Servers and Databases must be controlled on a need-to-know basis." This aligns with the GDPR principles relating to processing of personal data. They guide internal Microsoft services to use several built-in security controls like Azure AD-based authentication, firewall setup, encryption and Auditing and Threat Detection capabilities.

As an example, the ISRM policy requires the use of Azure AD Authentication to Azure SQL Database and Data Warehouse:

- All users must authenticate using Azure AD-backed credentials.
- SQL server authentication is not permitted.
- An Active Directory (AD)-contained user/group must exist in the user database.

ISRM also advocates the use of federated authentication, which simplifies permissions management as everything is centralized in the single integrated Active Directory.

Dynamic Data Masking

Dynamic Data Masking (DDM) limits sensitive data exposure by masking the data to non-privileged users or applications. DDM allows the database administrator to select a particular table-column that contains sensitive data, add a mask to it (there are a few available built-in masks that can be applied, as well as a customizable mask), and designate which DB users are privileged and should have access to the real data. Once configured, any query on that table/ column will contain masked results, except for queries run by privileged users.

DDM works by masking the data on the fly, at query time, in the result set of the query itself—with minimal impact on the application layer. Meanwhile the data in the database remains intact—and thus can still be accessed by users with appropriate privileges. In this way, access to the actual data is limited to only those who really have a need to access it.

In addition, for users of Azure SQL Database, DDM can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This can help with the identification of personal data qualifying for GDPR protection, and for reducing access such that it does not exit the database via unauthorized access.

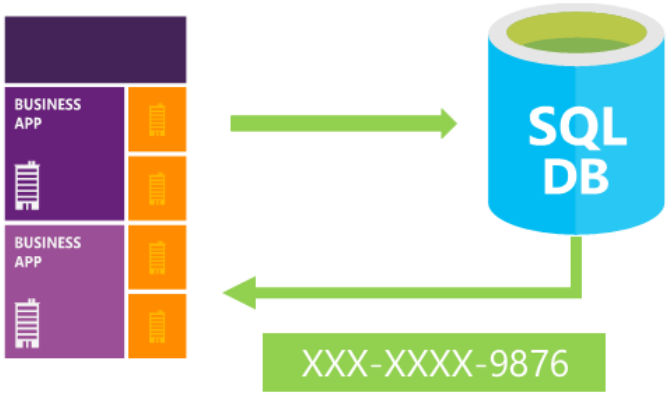
Applying DDM does not explicitly require any application changes, so it is easy to use with existing applications.

Microsoft Case Study: Azure Marketplace team restricts access using Dynamic Data Masking

The Microsoft Azure Marketplace team also manages databases with personal information, and the application logic has a clear division between who needs to see certain pieces of data to support the service, and who does not.

The Microsoft Azure Marketplace team creates and manages an online market for buying and selling finished software as a service (SaaS) applications and premium datasets. A specific team manages a platform called the Azure Marketplace Seller Insights Reporting platform, which enables ISVs, publishers, and developers to see their customers' orders, usage, payout and customer identity details.

The Azure Marketplace team restricts access to sensitive data to the minimal set of individuals with a legitimate need to access it. The Azure Marketplace team has accomplished this in part by implementing Dynamic Data Masking on columns that contain personal data such as name, email, company name, and geographic information. This allows the team to pseudonymize this data within the application, and thus limit access to the data in accordance with GDPR principles.



DDM is supported starting from SQL Server 2016 (all editions) and for Azure SQL Database.

Why is this important for the GDPR?

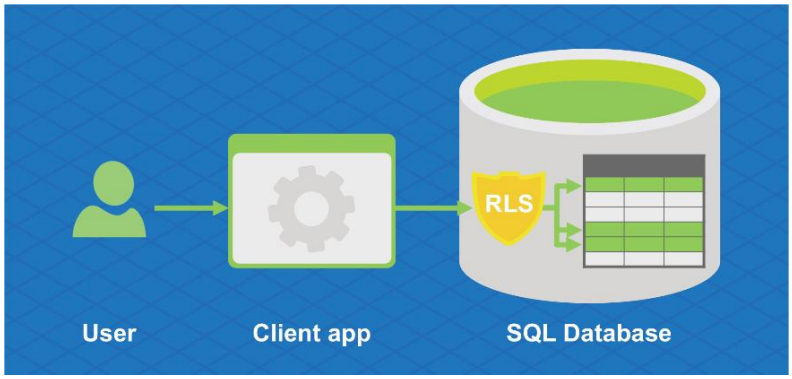
The GDPR calls for implementing "appropriate technical and organizational measures, such as pseudonymization, which are designed to implement data-protection principles, such as data minimization, in an effective manner…"

GDPR Article 25(1)— "Data protection by design and by default."

Row-Level Security

Another built-in means to restrict access according to specific user entitlements is via Row-Level Security (RLS). Use RLS to control access to rows in a database table based on the characteristics of the user executing a query. In this way, only database users that have a specific need to access data in a database row will be granted that access. For example, workers can access only those data rows that are pertinent to their department, or a nurse's access can be restricted to relevant data only for her assigned patients.

RLS can greatly simplify the design and coding of this type of access management and security within the application. The access restriction logic is in the database tier rather than away from the data in another application tier. The restrictions are applied every time that data access is attempted from any tier. This makes the security system more reliable and robust by reducing the system's surface area.



This built-in access control solution can be very helpful in managing the application access model.

Why is this important for the GDPR?

Users have access only to the data that is pertinent to them, thus reducing the risk of "unauthorized disclosure of, or access to personal data."

GDPR Article 32(2)—"Security of processing."

RLS is supported starting from SQL Server 2016 (all editions) and for Azure SQL Database.

Protecting personal data against security threats

In previous phases, the focus was on discovery of where the sensitive data is located and how it can be accessed. The objective was to create access controls to the system in accordance with the principle of least privilege, enabling only authorized access to the database system and data. In this phase, the focus shifts to the data itself, and to the application of protection and monitoring mechanisms in Microsoft SQL-based technologies to thoroughly protect the data.

Microsoft SQL-based technologies provide a powerful set of built-in capabilities that safeguard data and identify when a data breach occurs.

Why is this important for the GDPR?

Protecting personal data against security threats is specifically declared as a core requirement of the GDPR. The GDPR requires that organizations implement **"Data protection by design and by default" GDPR Article 25.**

GDPR Article 32(1)—"Security of processing" states that an organization must implement appropriate technical and organizational measures to ensure a level of security appropriate to the risk, including inter alia as appropriate:

- The pseudonymization and encryption of personal data.
- The ability to ensure the ongoing confidentiality, integrity, availability, and resilience of processing systems and services.
- The ability to restore the availability and access to personal data in a timely manner in the event of a physical or technical incident.
- A process for regularly testing, assessing, and evaluating the effectiveness of technical and organizational measures for ensuring the security of the processing.

There is also a requirement of "notification of a personal data breach" within a specified window of 72 hours, where feasible, after the organization becomes aware of it.

GDPR Article 33(1)—"Notification of a personal data breach to the supervisory authority."

Transport Layer Security

Microsoft SQL encryption technologies can be applied at different levels. It is a best practice to always use connections secured with Transport Layer Security (TLS). This ensures that data is encrypted in transit to and from the database, and reduces susceptibility to "man-in-the-middle"

Why is this important for the GDPR?

GDPR talks about integrating the "necessary safeguards into the processing" (**GDPR Article 25(1)**, "Data protection by design and by default"), and accounting for risks presented by processing, "in particular from accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to personal data transmitted, stored or otherwise processed" (**GDPR Article 32(2),** "Security of processing"). Protecting data during transmission is explicitly called out in this context—to avoid possible leakage and minimize these risks.

attacks. SQL Server and Azure SQL Database have TLS1.2 support enabled, and this is the recommended protocol to use for highly secure communication.

To enable encrypted connections in SQL Server, provision a certificate on the server, configure the server to accept encrypted connections, and configure the client to request encrypted connections. To enforce that all client connections must be encrypted, set the *ForceEncryption* flag to **Yes**.

For Azure SQL Database and Azure SQL Data Warehouse, all connections require encryption at all times. To securely connect with a client, specify connection string parameters (for ADO.NET driver) *Encrypt=True* and *TrustServerCertificate=False*.

Transparent Data Encryption

Another level of encryption is encryption of data at rest, using a built-in Microsoft SQL feature called Transparent Data Encryption (TDE). TDE addresses the scenario of protecting the data at the physical storage layer. TDE performs real-time encryption and decryption of the database, associated backups, and transaction log files without requiring changes to the application.

Why is this important for the GDPR?

This relates to the GDPR obligation to take into account "risks that are presented by processing, in particular from accidental or unlawful destruction, loss, alteration, [or] unauthorized disclosure of" that data. In this case, the protection is at the level of the physical device—and prevents the risk of compromising the storage itself, for example via copying the physical data out to another server.

GDPR Article 32(2)—"Security of processing."

TDE is often required by compliance regulations and various industry guidelines as a core requirement, as it ensures the full protection of data at the physical layer. Note that with TDE, data tables are not directly encrypted. Specifically, if a user was given permissions to a database with TDE enabled, the user can see all data. Instead, TDE protects the physical data files and transaction log files. If these are moved to another server, for instance, they cannot be opened and viewed on that server.

TDE is straightforward to enable—and applies to the entire database.

In SQL Server, keys and certificates must first be created, and then Encryption must be set to **On** in the database. The keys required are a master key, a certificate protected by the master key, and a database encryption key that is protected by the certificate. For more details, please see the TDE documentation.

In Azure SQL Database, enabling TDE is even simpler. Currently the keys are entirely managed by the Azure SQL Database service, making encryption-at-rest as simple as enabling the feature on a database. However, to enable greater flexibility for customers that require it, the service will also support customer-managed keys in future versions.

Always Encrypted

For some classes of highly sensitive data, encryption at rest and on the wire may not be sufficient. The GDPR addresses a few different classes of sensitive data that warrant heightened protection (see **GDPR Article 9**), and consequently a customer may want to implement data governance policies that allow differentiating how different types of personal data are classified, and thus protected.

Microsoft SQL databases (Azure SQL Database and SQL Server 2016) offer an industry-first security feature called Always Encrypted, which is designed to protect highly sensitive data.

Always Encrypted allows customers to encrypt sensitive data inside client applications and never reveal the encryption keys to the database engine (SQL Database or SQL Server). As a result, Always Encrypted provides a separation between those who own the data (and can view it) and those who manage the data (but should have no access). By ensuring that on-premises database administrators, cloud database operators, or other high-privileged but unauthorized users cannot access the encrypted data, Always Encrypted enables customers to confidently store sensitive data outside of their direct control.

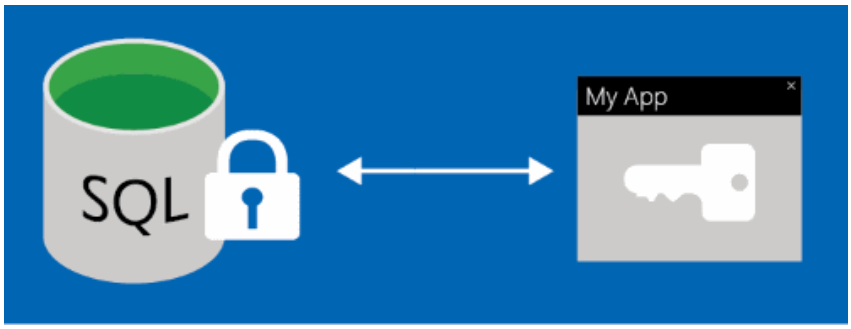
Microsoft Case Study: Dynamics 365 uses Always Encrypted to protect sensitive data

The Dynamics 365 team at Microsoft manages databases that include personal information such as customer contact information and service bus connection strings in the database—which is classified as sensitive data and protected under the GDPR. To protect this data, the Dynamics 365 team employs the Always Encrypted technology.

Dynamics 365 uses Always Encrypted on database columns that are classified as highly confidential—which the database owner (DBO) does not have a business need to see. The Dynamics 365 team described its Always Encrypted onboarding:

"The motivation was to remove encryption from our service and rely on a trusted platform to do this for us. Since we were already using Azure SQL Database and KeyVault, it was a perfect fit."

An advantage of Always Encrypted is that it makes encryption transparent to applications. An Always Encrypted-enabled driver installed on the client computer achieves this by automatically encrypting and decrypting sensitive data in the client application. The driver encrypts the data in sensitive columns before passing the data to the database engine, and automatically rewrites queries so that the semantics to the application are preserved. Similarly, the driver transparently decrypts data, stored in encrypted database columns, contained in query results.



Always Encrypted can be configured using a wizard in SQL Server Management Studio (SSMS). The wizard guides through the process of creating (or configuring) the Always Encrypted keys, creating key metadata, and determining which encryption type should be applied to a column. There are two encryption types supported: deterministic encryption and randomized encryption. Deterministic encryption always generates the same encrypted value for any given plain text value, and supports point lookups, equality joins, grouping, and indexing on encrypted columns. Randomized encryption uses a method that encrypts data in a less predictable manner. Randomized encryption is more secure, but prevents searching, grouping, indexing, and joining on encrypted columns.

One of the challenges of configuring Always Encrypted is to determine which type of encryption is most appropriate. Randomized encryption has a greater potential to break existing queries and application logic—but it provides a more secure option. This means that selecting this option may require some redesign of the application. Selecting deterministic encryption allows more operations to be performed on the encrypted data—though here, too, there may be some application logic modifications required. It is necessary to evaluate what is most appropriate for the particular use case and environment.

Why is this important for the GDPR?

The use of this powerful encryption feature which better ensures that highly sensitive data is encrypted on the server side, even in-memory, can significantly help in meeting the GDPR requirements around "Security of processing." This particularly applies to the requirement to "implement appropriate technical and organizational measures to ensure a level of security appropriate to the risk" including, as appropriate, "the pseudonymization and encryption of personal data".

GDPR Article 32(2) – "Security of processing."

Auditing for Azure SQL Database

Auditing for Azure SQL Database tracks database activities by writing events to an audit log. It enables the customer to understand ongoing database activities, as well as analyze and investigate historical activity to identify potential threats or suspected abuse and security violations.

Auditing can be enabled in the Auditing pane in the Azure portal, by selecting a storage account to save the logs to, and turning the feature on. A retention period can be enabled for audit logs according to specific requirements. This can be done using the retention slider in the storage configuration of the Audit settings.

Why is this important for the GDPR?

The GDPR, as part of the data protection requirement, stipulates a requirement that "Each controller… shall maintain a record of processing activities under its responsibility."

GDPR Article 30(1)—"Records of processing activities."

The audit data may sometimes need to be analyzed to investigate specific issues that come up. Audit

logs can be read using the `fn_get_audit_file` function, or viewed via the Azure Portal Audit Records blade, SQL Server Management Studio (SSMS), or Azure Storage Explorer. However, with substantial amounts of data these tools can become quite unwieldy.

An additional option is to transmit audit logs into Microsoft Operations Management Suite (OMS) Log Analytics using this integration guidance. OMS Log Analytics can then be used to support advanced log consumption and analysis. Since the audit logs are in machine-readable format, a customized integration can also be created with a log analytics tool of choice.

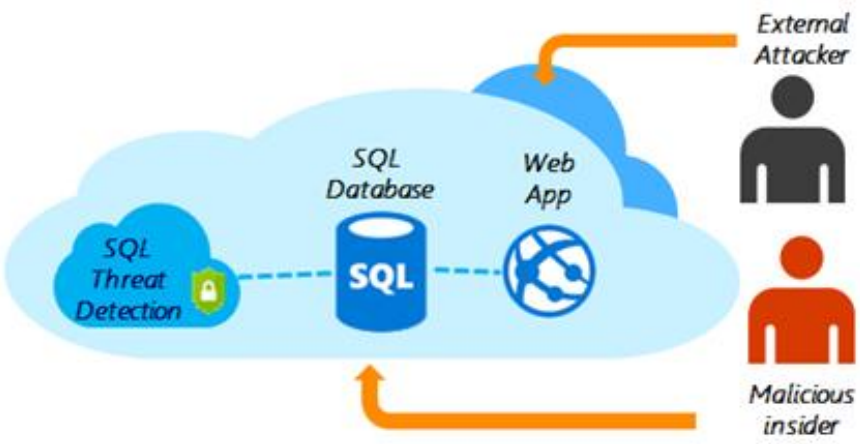
SQL Threat Detection

Another built-in feature of Azure SQL Database to help secure the database environment is SQL Threat Detection. SQL Database Threat Detection detects anomalous database activities indicating potential security threats to the database.

> Why is this important for the GDPR?
>
> The GDPR has a clear requirement regarding data breaches: "In the case of a personal data breach, the controller shall without undue delay and, where feasible, not later than 72 hours after having become aware of it, notify the personal data breach to the supervisory authority."
>
> **GDPR Article 33(1)—**"Notification of a personal data breach to the supervisory authority."

SQL Threat Detection on Azure SQL Database can greatly help in meeting this GDPR requirement.

The essence of this service is to proactively notify the Azure database administrator (or subscription owners) of any suspicious activity that could indicate a possible malicious intent to access, breach, or exploit data in the database.



Threat Detection operates by continuously profiling and monitoring application behavior, and employing machine learning and behavioral analytics methodologies to detect anomalies and unusual behavior. Upon detection of an unusual behavior pattern or a potential SQL injection attack, Threat Detection immediately notifies the database administrator of the possible breach. Threat detection alerts can be viewed in Azure Security Center. They provide details of the suspicious activity and recommend actions on how to investigate and mitigate the threat.

The proactive notifications and associated actionable information in Threat Detection enable the mitigation of security threats and enhanced protection for the database system. Such notifications may also be helpful in meeting the data breach notification requirement of the GDPR.

To learn more about Azure SQL Database Threat Detection, including pricing, visit the Azure blog.

Threat Detection can be enabled in the Azure Portal once Auditing is enabled, in the common Auditing & Threat Detection settings pane. All that is required is to turn it on, and specify an email address to receive alert notifications.

Auditing and Threat Detection are supported for both Azure SQL Database and Azure SQL Data Warehouse, on the database and server level.

Microsoft Case Study: WDATP team making use of Auditing and Threat Detection

The Windows Defender Advanced Threat Protection (WDATP) team manages a security service that enables enterprise customers to detect, investigate, and respond to advanced threats on their networks.

The team manages several different types of databases in its service, some of which contain personal data such as customer contact information which qualifies for protection under the GDPR. WDATP is built on Azure, and manages data in Azure SQL Database.

As part of its data protection strategy, the WDATP team tracks all activity on its databases and receives proactive notifications of suspected threats by enabling Azure SQL Database Auditing and Threat Detection in its database environment.

WDATP generates large amounts of audit data, and makes use of Microsoft Operations Management Suite (OMS) to consume and analyze audit data.

The WDATP team also benefited from a Threat Detection notification warning of a vulnerability to SQL injection in their SQL code. The team was able to quickly fix the vulnerability thanks to the guidelines provided by the notification, and thus avoid a data breach.

SQL Server Audit

SQL Server also contains an auditing capability that enables tracking activities on an on-premises database, and to monitor those activities that specifically affect sensitive personal data.

> Why is this important for the GDPR?
>
> The GDPR requires that "Each controller … shall maintain a record of processing activities under its responsibility."
>
> **GDPR Article 30(1)**—"Records of processing activities."

SQL Server Audit enables the customer to understand ongoing database activities, and analyze and investigate historical activity to identify potential threats or suspected abuse and security violations. SQL Server Audit enables the creation of server audits, which can contain server audit specifications for server level events, and database audit specifications for database level events. Audited events can be written to the event logs or to audit files. Granular control is available to specify exactly what events to audit, aligning with specific needs.

Audits can be defined using SQL Server Management Studio (SSMS) or using Transact-SQL commands. The following general process can be used for creating and using an audit:

1. Create an audit and define the target.
2. Create either a server audit specification or database audit specification that maps to the audit. Enable the audit specification.
3. Enable the audit.
4. Read the audit events by using the Windows Event Viewer, Log File Viewer, or the `fn_get_audit_file` function.

SQL Server Audit is an efficient and effective mechanism to help with coverage of the GDPR requirement to maintain a record of processing activities.

Business continuity—SQL Server Always On

One additional important element of protecting data concerns ensuring its resiliency and availability in the event of an adverse incident. This is another area where Microsoft SQL-based technologies have several built-in capabilities that can help in the creation of an effective compliance strategy.

> Why is this important for the GDPR?
>
> The GDPR explicitly refers to this important aspect of protecting data, by requiring that the organization "implement appropriate technical and organizational measures" that take into consideration, as appropriate, "the ability to restore the availability and access to personal data in a timely manner in the event of a physical or technical incident".
>
> **GDPR Article 32(1)**—"Security of processing."

In SQL Server, Always On Availability Groups can be used to maximize the availability of a set of user databases for an enterprise. An *availability group* supports a failover environment for a discrete set of user databases, known as *availability databases*, that fail over together. An availability group supports a set of read-write primary databases and one to eight sets of corresponding secondary databases. Optionally, secondary databases can be made available for read-only access and/or some backup operations.

There is a rich set of options that can be configured, such as alternative availability modes (asynchronous versus synchronous commit mode), numerous availability replicas, different forms of availability-group failover (like automatic, manual, and forced failover) and a flexible failover policy, and varying active-secondary capabilities. See the article about Always On Availability Groups for more information.

Always On Failover Cluster Instances is another part of the SQL Server Always On offering for business continuity. Always On Failover Cluster Instances uses Windows Server Failover Clustering (WSFC) functionality to provide local high availability through redundancy at the server-instance level—a failover cluster instance (FCI). An FCI is a single instance of SQL Server that is installed across Windows Server Failover Clustering (WSFC) nodes and, possibly, across multiple subnets. On the network, an FCI appears to be an instance of SQL Server running on a single computer, but the FCI provides failover from one WSFC node to another if the current node becomes unavailable. See the article about Always On Failover Cluster Instances for more information.

For more options and details on implementing a high availability and disaster recovery strategy for SQL Server on Azure VM, see the article High availability and disaster recovery for SQL Server in Azure Virtual Machines.

Business continuity in Azure SQL technologies

In Azure SQL Database, there are several options available for business continuity and disaster recovery solutions. The appropriate solution can be selected per data store, depending on varying business requirements of availability.

The Azure SQL Database service performs automated database backups periodically, and the service guarantees Point-in-Time Restore from these backups for a certain scope of recovery. Long-term retention for backups is also available, by storing Azure SQL Database backups in an Azure Recovery Services vault for up to ten years.

Azure SQL Database also offers an Active Geo-Replication feature, which provides a database-level recovery solution with low recovery time. Active Geo-Replication enables the configuration of up to four readable secondary databases in the same or different regions. Beyond offering a complete business continuity and disaster recovery solution, Active Geo-Replication also enables load-balancing using the secondary database, and offloading read-only workloads. It also supports user-controlled failover and failback and configurable performance levels for the secondary databases.

In its business continuity solutions, Azure respects data residency requirements. While Microsoft may replicate data to other regions for data resiliency, it will not replicate or move customer data outside

the defined geo boundary. Customers, however, may move, copy, or access their customer data from any location globally.

See the Overview of business continuity with Azure SQL Database for more details on evaluating these business continuity solutions.

Reporting on data protection policies and reviewing regularly

The final phase of the methodology for GDPR compliance addresses the need to maintain a record of personal data processing activities under the controller's responsibility, and to make the record available to the supervisory authority upon request. This phase also deals with the continuous process of reviewing the controls and security of the system, to better ensure ongoing compliance with GDPR principles.

Since the basis for reporting relies on maintaining documentation and records, the Microsoft SQL Auditing capabilities can serve as an essential component for fulfilling these requirements. Maintaining audit logs for all Microsoft SQL activities ensures the existence of a persistent record of database access and processing activities at all times. These records can then be analyzed and packaged to provide evidence needed for various record-keeping requirements. Furthermore, the records can be analyzed specifically for the processing of personal data, and reported specifically on this for GDPR requirements. And of course, as discussed previously in this document, maintaining an audit trail is a critical data protection component, which provides the ability to investigate historical records to identify any unusual or suspect activity.

Why is this important for the GDPR?

The GDPR requires an assessment of the impact of processing operations on the protection of personal data where such processing is likely to result in high risk. This includes an assessment on "the measures envisaged to address the risks, including safeguards, security measures and mechanisms to ensure the protection of personal data."

GDPR Article 35(7)— "Data protection impact assessment."

An additional built-in capability in Microsoft SQL that can assist with reporting procedures is Temporal Tables. System-versioned temporal tables are a new type of user table in SQL Server 2016, designed to keep a full history of data changes and allow easy point in time analysis. This can be a vital tool in providing reports on the state of the data at a particular point in time, as temporal tables enable reconstructing the state of the data as of any given point in time. An added benefit is the ability to perform data forensics when necessary, as all data changes are audited. Temporal tables are supported for SQL Server (starting with 2016) and Azure SQL Database.

For Microsoft SQL-based technologies, the best way to create such an assessment is to document all security and availability mechanisms being used—to clearly demonstrate the data protection policy being implemented in the environment. This can serve as evidence that these security controls are effectively mitigating the risks associated with processing personal data qualifying for protection under the GDPR.

Review—consistently analyze data and systems to reduce risk

An important element of achieving and maintaining GDPR compliance is to regularly review the security state of data and systems, to ensure that they meet the standards expected by the organization. Within a database system, the customer may want to consider performing periodic assessments, covering the state of database permissions, surface area exposure, protection mechanisms on sensitive data, and monitoring methods and procedures.

This type of assessment process can be run automatically today on Microsoft SQL Server, whether on-premises or installed on an Azure VM, by making use of the Operations Management Suite SQL Assessment. Operations Management Suite (OMS) is a cloud-based management system for hybrid environments, enabling management, monitoring, and performing advanced analytics on data gathered from the IT environment for many purposes. OMS includes a solution known as the SQL Assessment, which runs a periodic assessment of the risk and health of the SQL Server environment. This is primarily used for monitoring purposes. It also performs a limited set of security checks to review the current security status of the database environment.

Looking forward, these capabilities will be enhanced as Microsoft will introduce some additional offerings that can ease the efforts of meeting GDPR requirements. The goal of these offerings is to help in assessing the security state of the database environment, and map and trace sensitive data. These can be quite significant in facilitating the work of data protection in the context of the GDPR.

Looking forward

The Microsoft SQL product team is actively investing in advanced, intelligent capabilities that can help facilitate the journey to GDPR compliance. Some forward-looking tools and features in development include:

- Helping discover, classify, and protect sensitive data.
- Tracking personal or sensitive data throughout the database system and beyond.
- Assessing database vulnerabilities and overall security state.
- Helping meet security best practices with controls and hardening recommendations.

These Microsoft SQL intelligent offerings are being designed to help address and simplify many of the processes needed to attain and maintain GDPR compliance. The goal is to facilitate these processes by providing built-in tooling and automation. These tools can help efficiently manage key elements of the GDPR process like discovering where the personal data resides, tracing data lineage throughout other elements of the IT environment beyond the database, and enforcing strict security controls on access to and protection of susceptible systems. The aim is to also help in the continuous review of the protection status of the database environment, giving customers the peace of mind that their security posture is maintained.

As these capabilities become available, they will make the journey to GDPR compliance with Microsoft SQL-based technologies even smoother and more straightforward.

In closing

Organizations will need to invest significantly to ensure the GDPR principles are effectively implemented and sustained in their environments. Microsoft itself is going through the substantial process of validating that all its data systems meet GDPR compliance readiness—and is approaching this task methodically and meticulously across the company.

The Microsoft SQL platform provides many built-in capabilities to help meet various requirements of the GDPR, and helps to significantly ease this process. Ranging from granular controls that can be defined, to integration with centralized authentication management services and industry-leading methods to protect and maintain the availability of data, Microsoft SQL-based technologies offer a wide set of powerful capabilities to address data privacy principles in the data platform tier. This document has studied examples of how several teams within Microsoft itself are making use of these capabilities to help ensure the fortification and security of their environments, playing a major part in addressing some of the major principles of the GDPR.

Microsoft hopes customers can benefit from its experience and learnings in approaching data privacy and compliance processes, in order to help them meet GDPR requirements in their Microsoft SQL environments.

For more information on Microsoft SQL Security, visit the Security Center for SQL Server Database Engine and Azure SQL Database.

For an in-depth guide on how Microsoft can help customers begin their journey to GDPR compliance, see Beginning your General Data Protection Regulation (GDPR) Journey.

To learn more about Microsoft's commitment to privacy and GDPR principles, visit the Trust Center site at www.microsoft.com/GDPR.