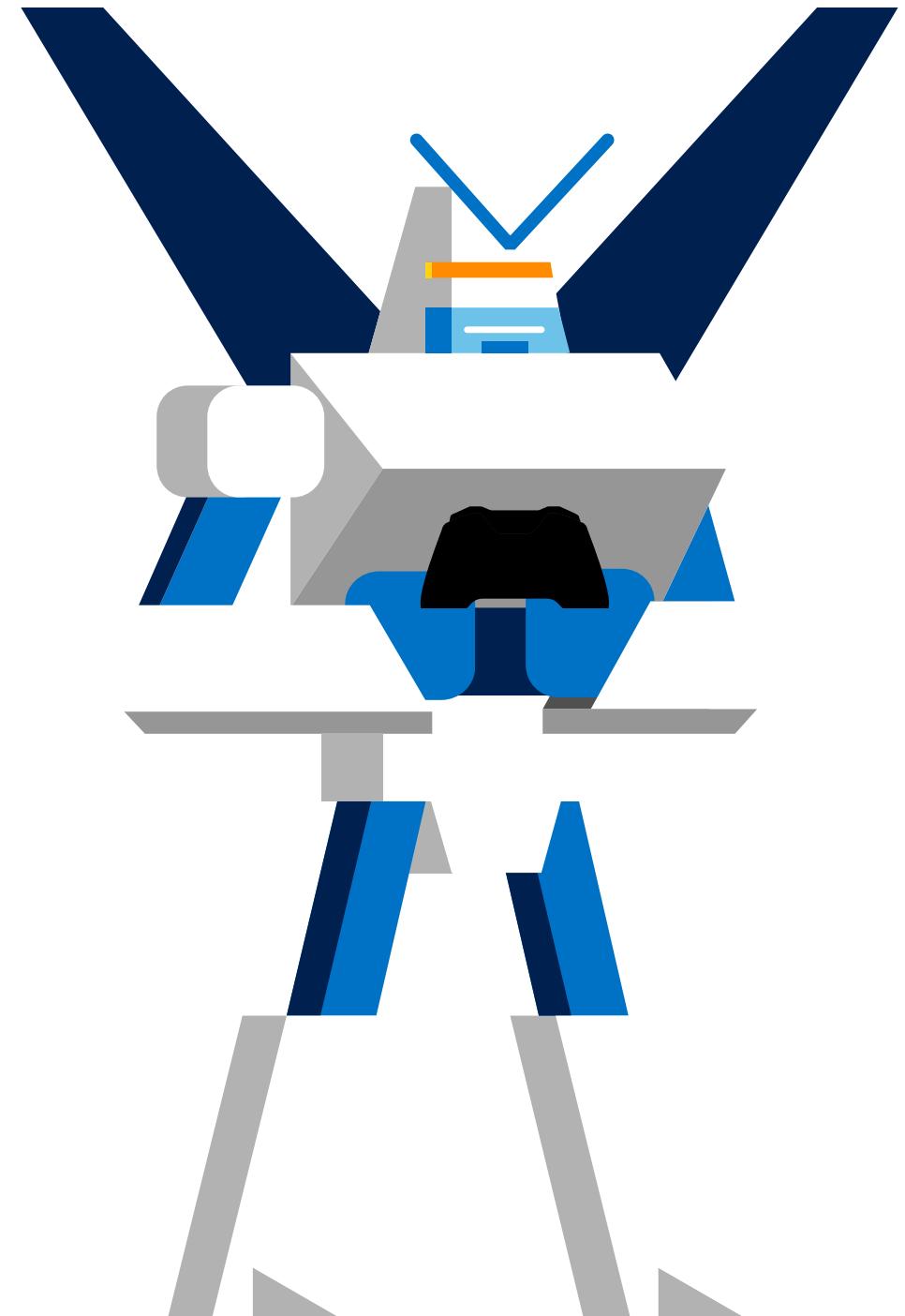


Azure SQL Database

José Manuel Jurado / Microsoft



About me..

Jose Manuel Jurado Diaz

Azure SQL DB Support Engineer

Subject Master Expert

Speaker SQL Saturday, SQL Nexus
channel9 and TechReady

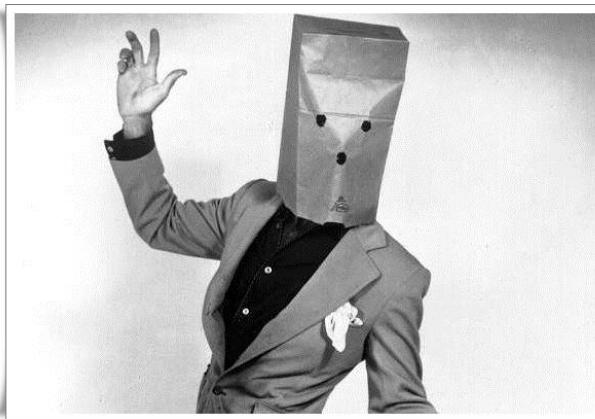


Email: jmjurado@microsoft.com

Before Getting Started - Questions



Did you work with SQL Azure previously? V11 or V12?



If not, do you know what it is?



If you know, did you have any trouble or issue working with SQL Azure previously?

Agenda

- ❑ Getting Started.
 - What is it?
- ❑ Changes and new improvements in Azure SQL Database V12
 - Engine
 - Security
- ❑ High Availability and Disaster Recovery
 - Geo-Replication and Geo-Restore
- ❑ The new database model and Elastic Tools
 - Elastic Database Pools
 - Elastic Query, Jobs, Scale and Transaction
- ❑ New Tools and connectivity improvements
 - Connectivity
- ❑ Performance with Azure SQL DB
 - Query Data Store, Query Performance Insights, Index Advisor, Extended Events, Alerts, DMVs.
- ❑ Q&A



Getting started with SQL Azure DB

Chapter



Azure SQL Database – What is it?

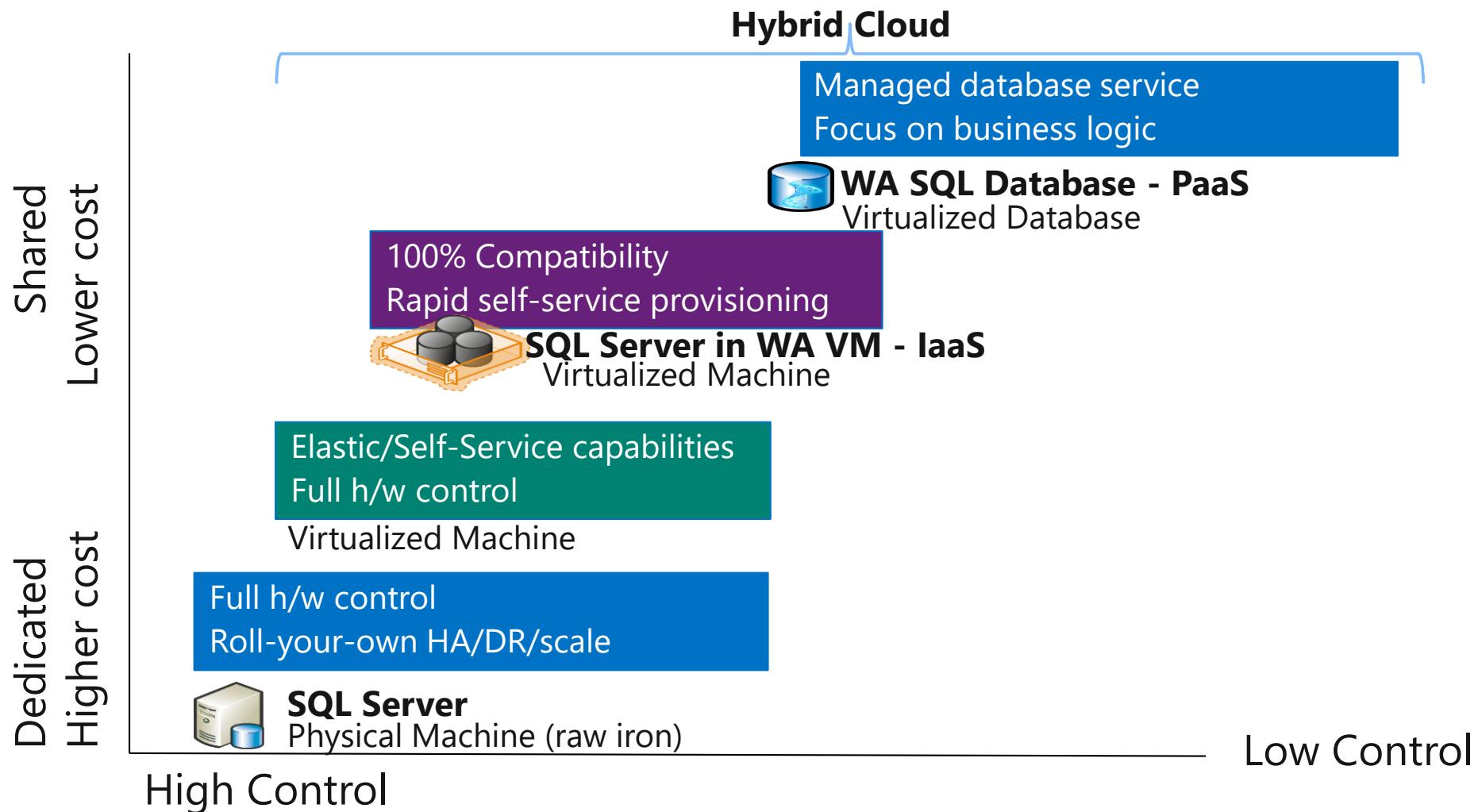


- SQL Server as Service
- Service managed by Microsoft
- Enterprise Edition options, limited only by hardware characteristics depending on the model.
- Almost 0 maintenance
- Resources oriented to use your database

Azure SQL Database – Cloud Models



Azure SQL Database – Database Platform



Azure SQL Database – Five Goals

Simple

Easy to create.
Dashboard metrics DB.
No need to change the application.
Multiple migration tools.

Scalability

Predictive performance.
Multiple database model depending on use.
Parallel Data-Warehouse

Trusted

Azure AAD or SQL Login
Multiple security mechanisms.
Compliance Auditing

Business Continuity

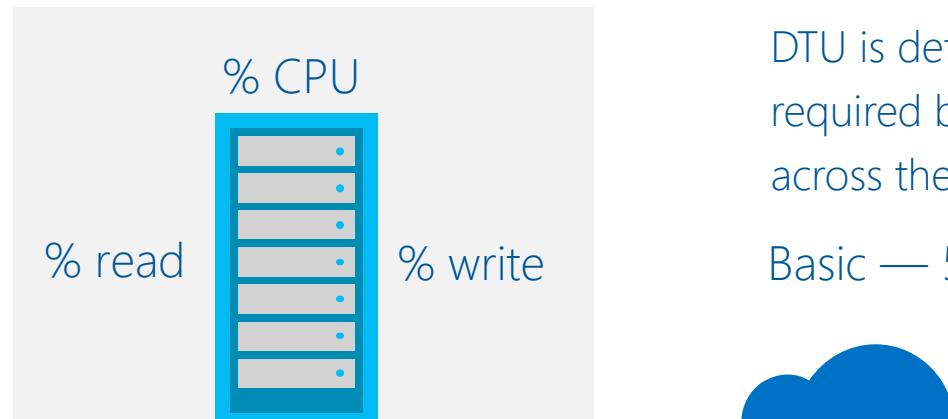
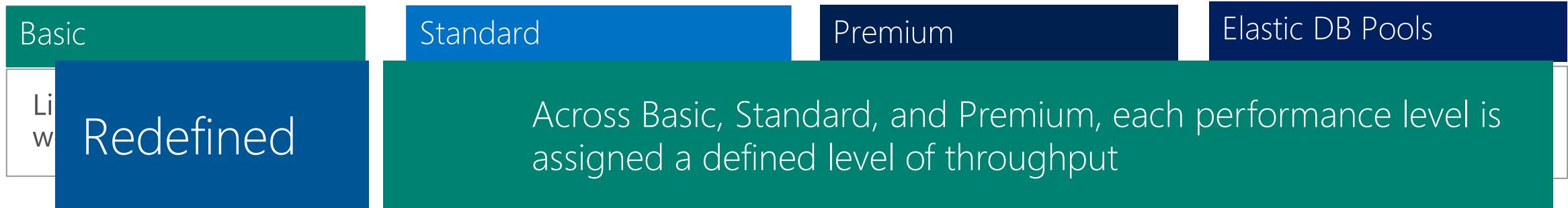
Disaster Recovery
Geo-Replication
Geo-Restore
Restore Point-in-Time.
Database Replicas.
Backups

Always Updated

Latest options available at company level.
Multiple monitoring tools.

Familiarization.

Azure SQL Database – Database Level/DTUs



DTU is defined by the bounding box for the resources required by a database workload and measures power across the six performance levels.

Basic — 5 DTU	S0 — 10 DTU	P1 — 125 DTU
	S1 — 20 DTU	P2 — 250 DTU
	S2 — 50 DTU	P4 — 500 DTU

Measure of power

Introducing the Database Throughput Unit (DTU) which represents database power and replaces hardware specs

New Improvements for Azure SQL DB V12 and SQL SERVER 2016

Chapter



What is new using V12 Azure SQL Engine

Engine

[Partitioning](#)

[Partial Full-Text Search](#)

[Parallel Execution for >P2](#)

[Contained database](#)

[Tables In-Memory](#)

[Temporal Tables \(Versioning\)](#)

[DBCC commands](#)

[CLR Integration – Stopped](#)

[Table as Heap](#)

[Data Compression](#)

[Enhanced Session Context](#)

[Setting changes Database](#)

[Change tracking](#)

[New functions and TSQL](#)

[Strech Database](#)

Compatibility Level to 130

[SQL Server 2016 \(Preview \)](#)

[Azure SQL DB V12](#)

Alter Database SCOPE Configuration

Available for:

[SQL Server 2016 \(Preview \)](#)

[Azure SQL DB V12](#)

Options:

[Procedure cache cleaning](#)

[Maxdop](#)

[Legacy Cardinality Estimation](#)

[Parameter sniffing](#)

Indexes

[Online Index.](#)

[Column Store Index.](#)

[XML Index.](#)

Replication

[Transactional](#)

[SQL Data-Sync](#)

New Product

[Azure SQL DataWarehouse – Parallel Data Warehouse](#)

High Availability

[Geo-Replication Offline/Online](#)

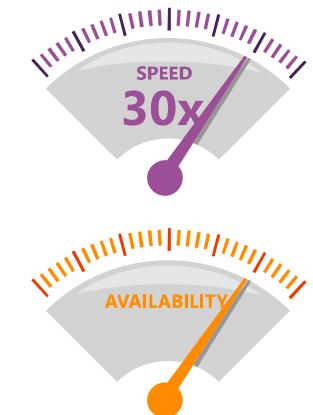


Max degree of parallelism

V11 all user databases is using MAXDOP = 1

V12 the Premium Databases are able to use a MAXDOP =>2

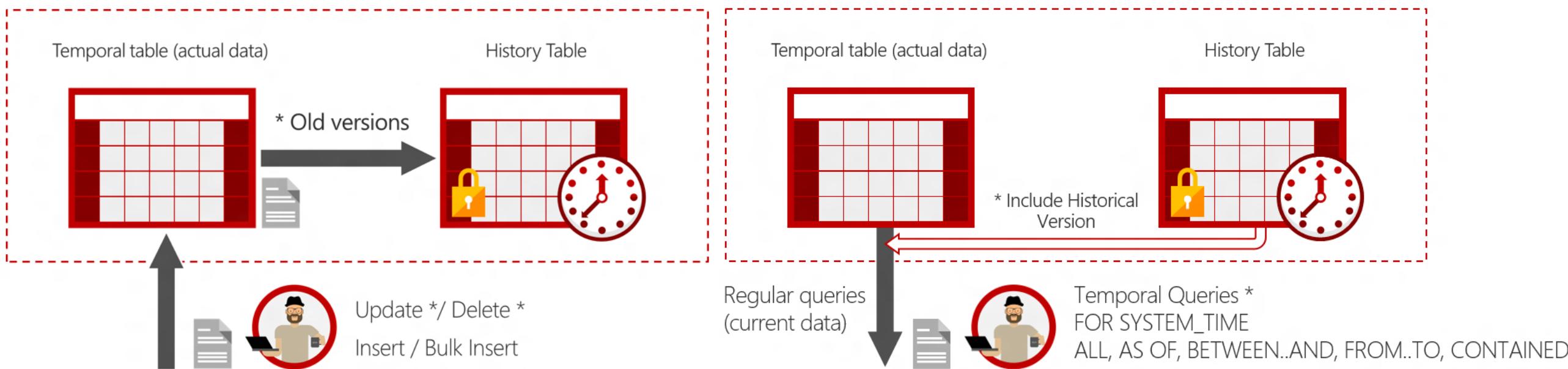
- **Benefit:** A Query execution will be use more than a core.
- **Issue:** A query execution could take more time. Using a [Hint](#) or plan guide in case that there is needed too much CXPACKET wait stat.



Temporal Tables

Maintains the versioning of the changes of a specific table and the range of date for this value.

- [Temporal Tables](#)
- [Getting Started with System-Versioned Temporal Tables:](#)
- [System-Versioned Temporal Tables with Memory-Optimized](#)
- [Tables Temporal Table Usage Scenarios](#)

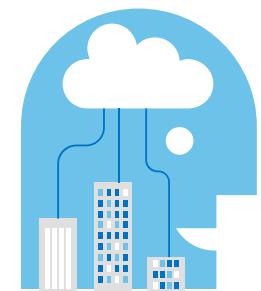


In-Memory Tables

- **Limited memory for In-Memory Tables depending the number of rows and layout**
- **The Isolation level is Snapshot.**

	Basic	Standard				Premium				
		S0	S1	S2	S3	P1	P2	P4	P6/P3	P11
Maximum database size	2 GB	250 GB				500 GB		1 TB		
DTUs	5	10	20	50	100	125	250	500	1,000	1,750
Point-in-time restore	Any point last 7 days	Any point last 14 days				Any point last 35 days				
Disaster recovery	Geo-Restore, restore to any Azure region	Standard Geo-Replication, offline secondary				Active Geo-Replication, up to 4 online (readable) secondary backups				
Max In-Memory OLTP storage	NA	NA	NA	NA	NA	1 GB	2 GB	3 GB*	8 GB	10 GB*
Max concurrent requests	30	60	90	120	200	200	400	800	1,600	2,400
Max concurrent logins	30	60	90	120	200	200	400	800	1,600	2,400
Max sessions	300	600	900	1,200	2,400	2,400	4,800	9,600	19,200	32,000

* In-Memory OLTP storage limits will soon adjust to 4 for P4 and 14 for P11.



Contained Databases

- **Easier migration between servers because there is not needed to create the login.**
- **In failover reconfiguration for master databases the contained databases are no affected.**



DBCC commands supported

Brings the possibility for customers to run DBCC commands including CHECKDB, CHECKALLOC, and CHECKTABLE.

For example:

- `SELECT file_id, name ,size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS AvailableSpaceInMB FROM sys.database_files;`
- `DBCC SHRINKFILE(FileID, size)`
 - ?1 => database
 - ?2=> log

`DBCC TRACESTATUS`

`DBCC CHECKDB('ColumnStoreDB')`

`DBCC OPENTRAN`



CLR Implementation – In Review by Product Team

What we can do?

Stored Procedure

Triggers

Function

Aggregate

User Defined Type - UDT

Assemblies needed

System

System.Data.SqlTypes

System.Data.SqlClient

Microsoft.SqlServer.Server

Enhanced Context Session

Azure SQL DB Session Context implements a session-scoped key-value store that allows applications to store any information in the database connection context scope.

- Set values using a new system stored procedure: `sp_set_session_context 'key', 'value'`
Key names are stored as **sysname**
Values are stored as **sql_variant**
Maximum overall size of cache: 256 kb per connection. This includes both keys (sysname, or maximum of 128 bytes) and values (sql_variant, or maximum of 8,000 bytes + 16 byte overhead).
- Retrieve values using a new built-in function: `SESSION_CONTEXT('key')`
 - Returns NULL if no value has been set for the specified key
 - All keys and values are reset by `sp_reset_connection`
 - No support to delete a single entry
 - Like `CONTEXT_INFO`, setting keys and values cannot be rolled back with a transaction



DATABASE SETTINGS

- ALTER DATABASE [p1] SET PARAMETERIZATION FORCED
- ALTER DATABASE [p1] set auto_update_statistics_async on
- ALTER DATABASE [P1] SET COMPATIBILITY_LEVEL = 120;
 - Database compatibility level 120 uses a new cardinality estimator that is tuned for modern data warehousing and OLTP workloads.
 - Applications that are migrated from earlier versions of SQL Server should be carefully tested to confirm that good performance is maintained or improved. If performance degrades, you can set the database compatibility level to 110 or earlier to use the older query optimizer methodology.



Replication

2 main ideas for leading this work.

- Easy migration of SQL Server workloads to AzureDB with minimum downtime
- Existing features such as Sync framework and Export/Import don't achieve real-time data movement.

Sync Framework is out of support and no feature upgrades are planned

Export/Import does not copy the data real time and there will always be gaps unless there is downtime for migration.

Supported Versions

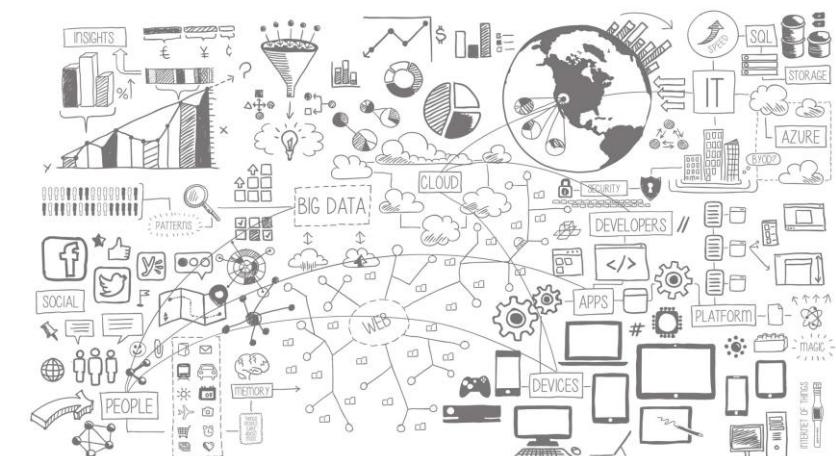
- **Publisher/Distributor**
 - SQL 14 RTM CU5, SQL11 SP2 CU4, SQL11 SP1 CU14, SQL16
- **Subscriber should be on V12**

Limitations/Restrictions

- Only [Transactional replication](#) is supported to AzureDB
- Pull subscription is not supported
- Peer-to-Peer, bi-directional or updateable subscriptions are not supported
- Distribution agent will use SQL auth and cannot use impersonate Dist agent process account
- No-sync subscribers is not supported (backup/restore for initializing subscription)

Azure SQL DataWarehouse

- Azure SQL Data Warehouse is a port of PDW technology to Azure.
 - It allows Multi Processing across many V12 Databases as a scale out technology for distributed databases.
 - It is designed as a distributed relational database for DW workloads, targeted at large data sets.
 - Designed to handle large data sizes, 10's to 100's of TB of data is typical. Azure Data Warehouse implementation can scale to Petabyte scale storage space.
 - Measured with DWU that is similar to DTU concept.
 - Scale-up/down in seconds.
 - It can scale from single node to hundreds of nodes.
 - Service can be paused to save billing costs.
 - Integrated with Azure family (ADF, HDInsight, SQL-IP, CloudML)
 - True T-SQL Support:
 - UDF
 - Stored Procedures,
 - Table Partitioning, Collations,
 - Indices
 - Column Store Support (but not 100% compatible Azure SQL DB)
 - Users can provision a new SQL DW database via the Azure portal
 - Users can use the existing PDW tools to upload data including dwloader.exe tool, SSIS and 3rd party software. As well as existing client tools, Excel, SSMS, RS, etc.



Security

Chapter



Security Approach

Azure Active Directory Security

TDE

Always Encrypted

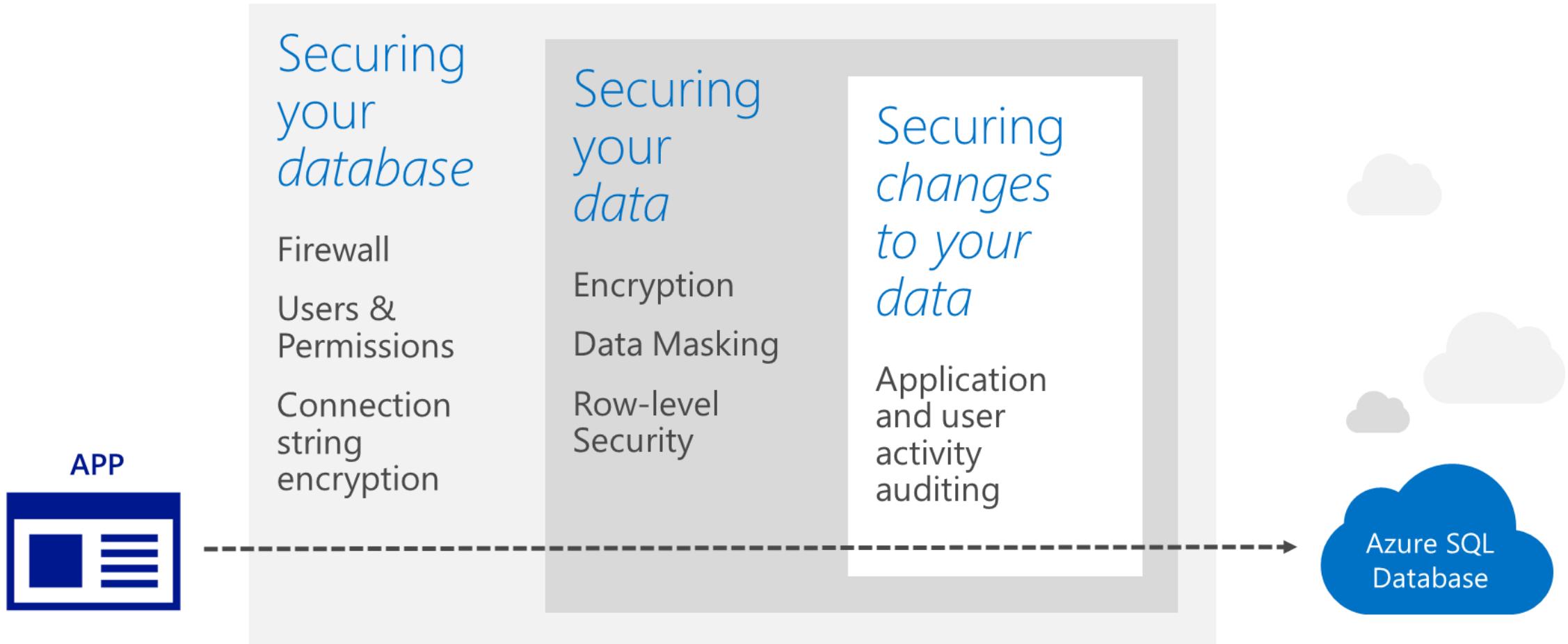
Row-level Security

Dynamic Data Masking

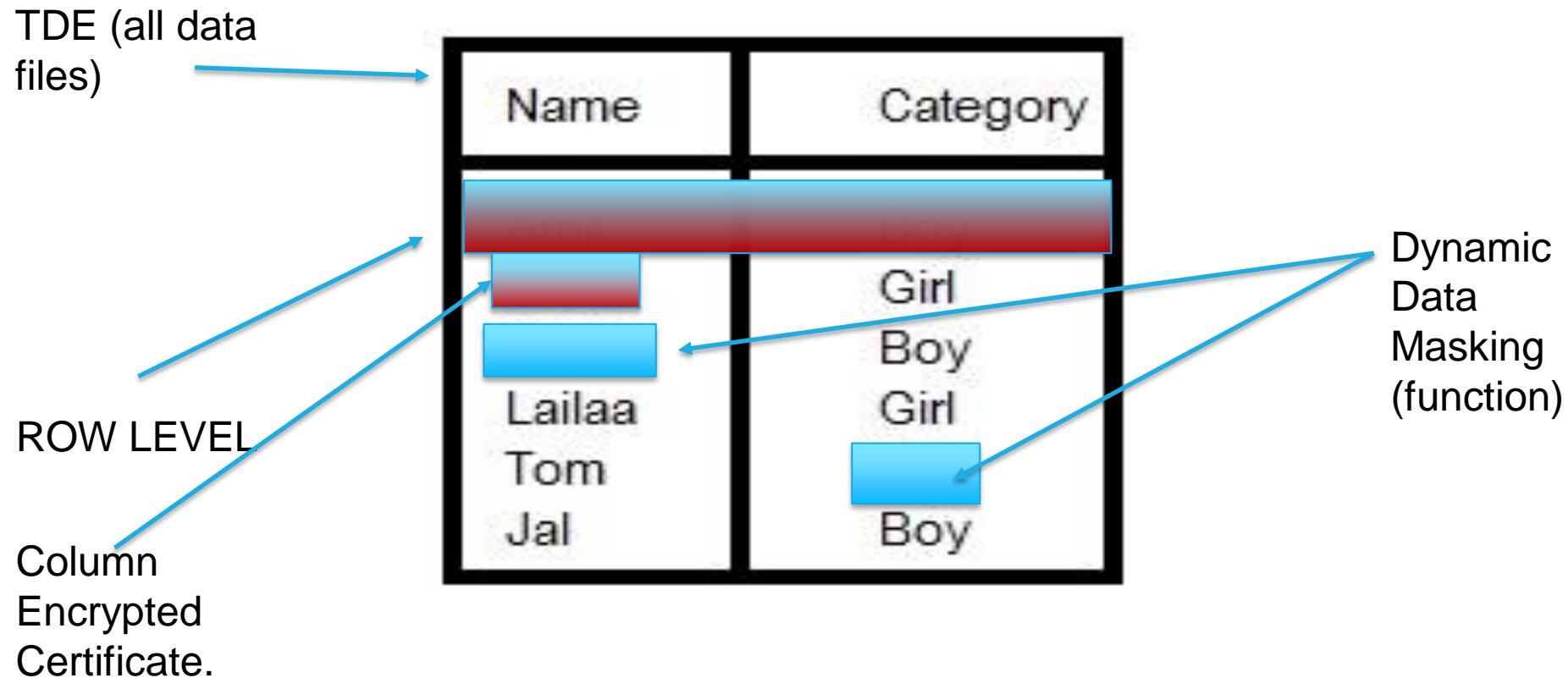
SQL Auditing

Thread Detection

Security approach



Security approach



Azure Active Directory Security

Azure Active Directory Security

Azure Active Directory authentication is a mechanism of connecting to Microsoft Azure SQL Database by using identities in Azure Active Directory (Azure AD)

Azure Active Directory Security

Centrally manage user permissions

Alternative to SQL Server authentication

Allows password rotation in a single place

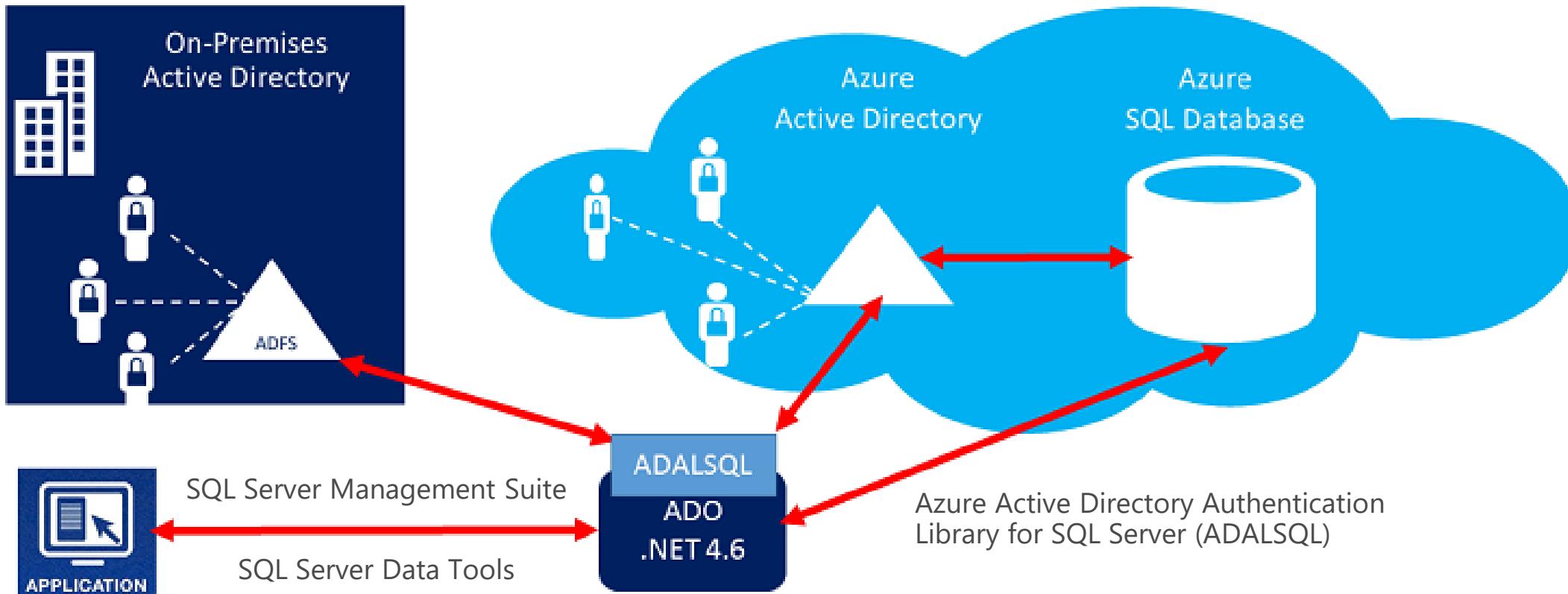
Enables management of database permissions using external Azure Active Directory groups

Stops password storing by using integrated Windows authentication and other forms of authentication supported by Azure Active Directory



Azure Active Directory Security

Azure Active Directory and Azure SQL Database



Azure Active Directory Security

The screenshot shows the Microsoft Azure portal interface. On the left, a sidebar lists various services: CDN, AUTOMATION, SCHEDULER, API MANAGEMENT, MACHINE LEARNING, STREAM ANALYTICS, OPERATIONAL INSIGHTS, NETWORKS, TRAFFIC MANAGER, REMOTEAPP, MANAGEMENT SERVICES, ACTIVE DIRECTORY, MARKETPLACE, STORSIMPLE, and SETTINGS. The SETTINGS item is highlighted with a red box. The main content area is titled 'settings' and contains tabs for SUBSCRIPTIONS, MANAGEMENT CERTIFICATES, ADMINISTRATORS, AFFINITY GROUPS, USAGE, and REMOTEAPP. The SUBSCRIPTION tab is selected, showing a table with one row. The row contains the following data:

SUBSCRIPTION	SUBSCRIPTION ID	ACCOUNT ADMINISTRATOR	DIRECTORY
123456.f...	123456...	account@outlook.com	Default Directory (account@out...)

At the bottom of the main content area, there is a 'NEW' button and an 'EDIT DIRECTORY' button, both enclosed in red boxes. The top right corner of the screen shows 'Subscriptions', the user's email 'account@outlook.com', and a profile icon.

Transparent Data Encryption

Real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application.

Transparent Data Encryption

Data is encrypted at rest, in flight, and while in use

Azure SQL Database service manages your keys

You can keep application changes to a minimum

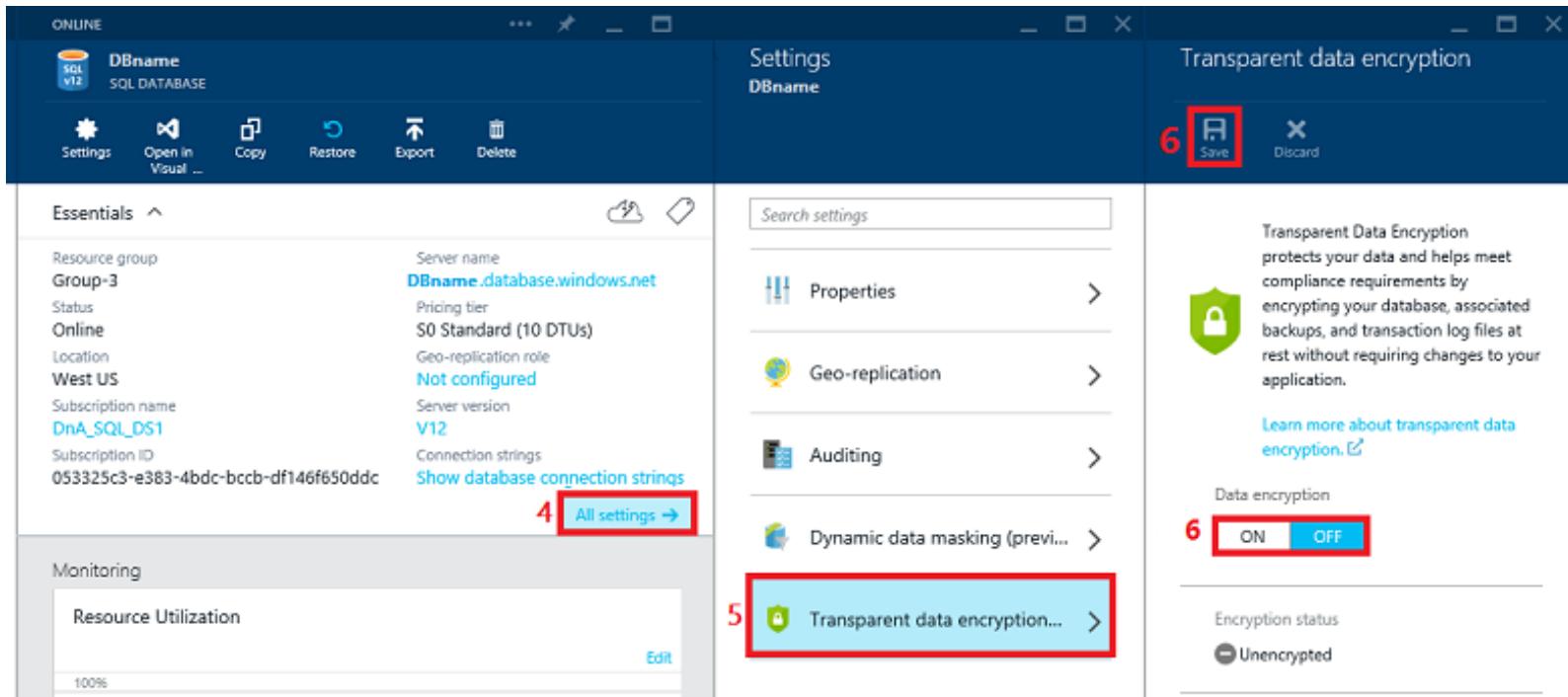
Encryption and decryption of data is done
transparently in a transparent data encryption
(TDE)-enabled client driver

Support for equality operations (including joins) on
encrypted data

Azure manages encryption keys



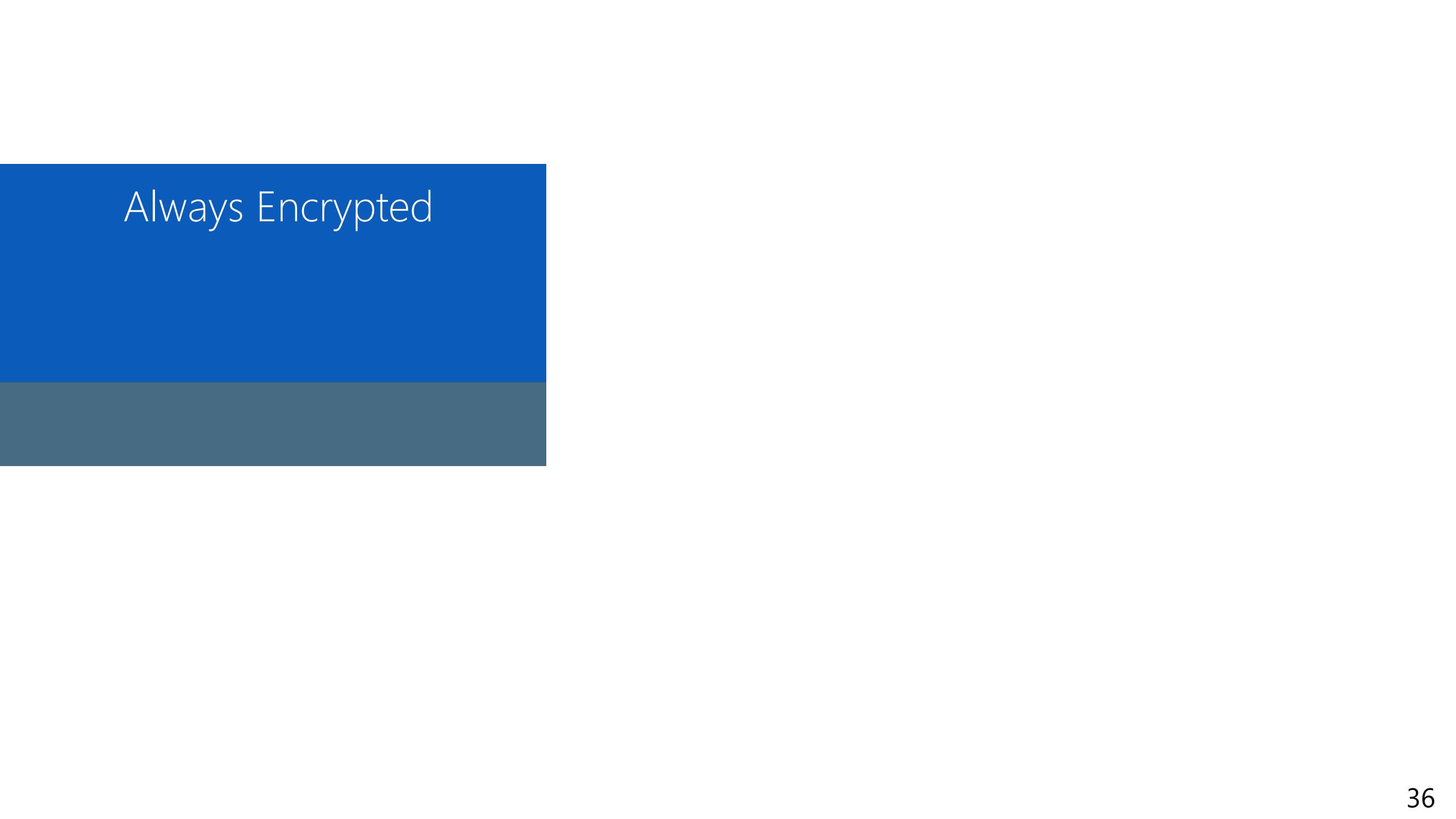
Transparent Data Encryption



Transparent Data Encryption

```
-- Enable encryption ALTER DATABASE [AdventureWorks] SET ENCRYPTION ON;  
GO
```

```
SELECT * from sys.dm_database_encryption_keys
```

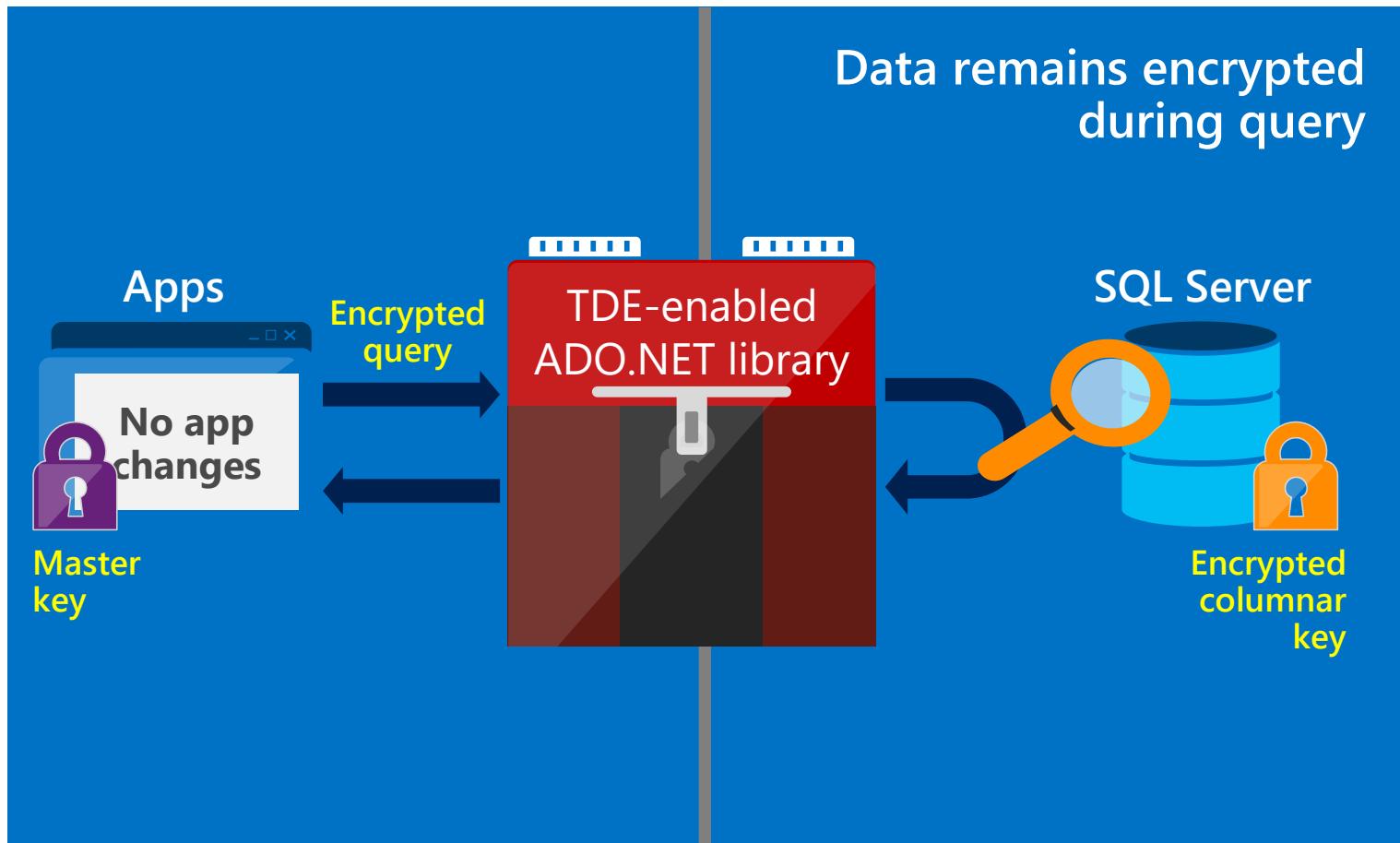


Always Encrypted

Always Encrypted

Always Encrypted allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to SQL Database. 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).

Always Encrypted



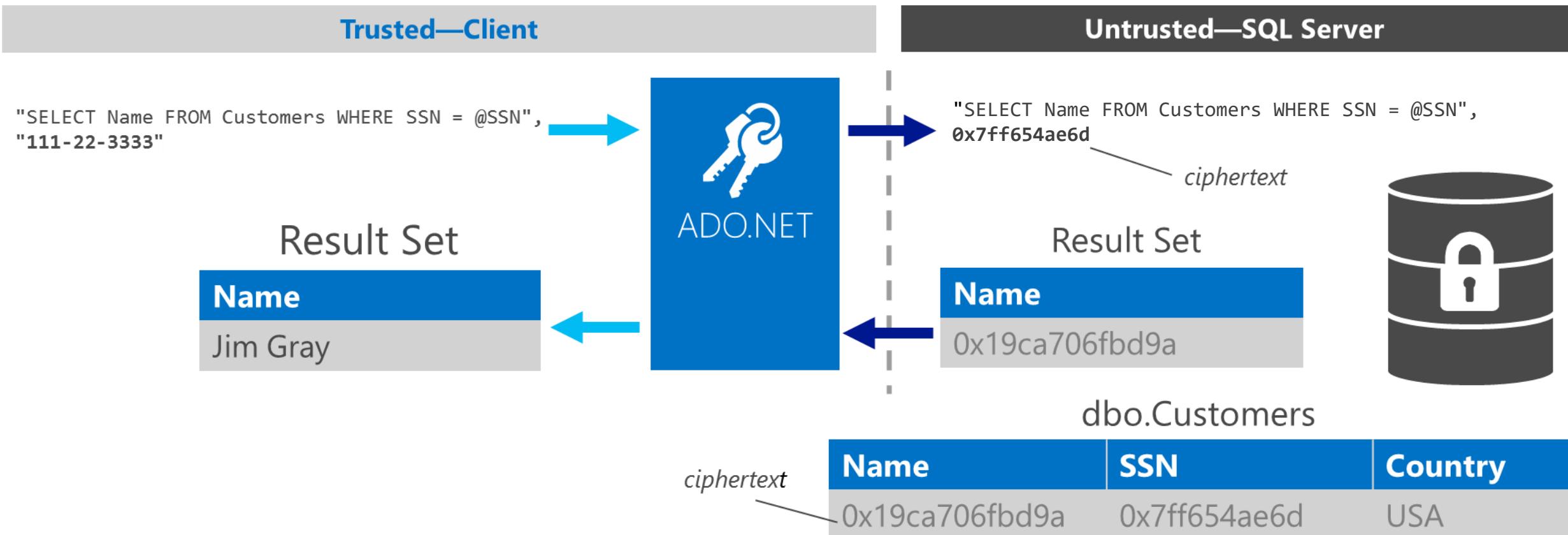
Capability

Transparent client-side encryption, while SQL Server executes T-SQL queries on encrypted data

Benefits

- Sensitive data remains encrypted and queryable at all times
- Unauthorized users never have access to data or keys
- No changes to applications are necessary

Always Encrypted



Always Encrypted

- * The key is always under control of the client and application, and is never on the server.
- * Neither server nor database administrators can recover data in plain text.
- * Encrypted columns of data are never sent to the server as plain text.
- * Limited query operations on encrypted data are possible.

Always Encrypted

Database side

Local stored certificate: **column master key** (CMK)

CMK will be used to protect **column encryption keys** (CEK), which encrypts the sensitive data.

Column master key definition object: information about the location of the CMK.

Column encryption key, encrypted with CMK

Column encryption key object: in the database uploading the encrypted value of the key.

Create a table with encrypted columns.

Always Encrypted

Client side

Same local stored **certificate**

"Column Encryption Setting=Enabled;" in connection string

".Net Framework 4.6" or higher

Always Encrypted

```
SELECT * FROM [dbo].[Patients]
```

100 %

	PatientId	SSN	FirstName	LastName	MiddleName	StreetAddress	City	ZipCode	State	BirthDate
1	28	0x01FA8C0E1560A78F779CFF041E78CE1C026F12007E0101E3...	John	Doe	NULL	NULL	NULL	NULL	NULL	0x01CA4ABF9F5D3203F63C67820185A223C3AC1BA98948E00...
2	29	0x01B05139055568621268279C46C7E887064F54216F81C812...	Joanne	Doe	NULL	NULL	NULL	NULL	NULL	0x01E1215A48290BF31163BE9C1C4A6CB4C9A3CB4BA088002...
3	30	0x014054BB44CEA3839AD42B684405DF7275B73EE42EDA98...	Michael	Park	NULL	NULL	NULL	NULL	NULL	0x01539807EB9CCC5D489EA30CD3F5F747FE2D30FC5F8FA9...

Row Level Security

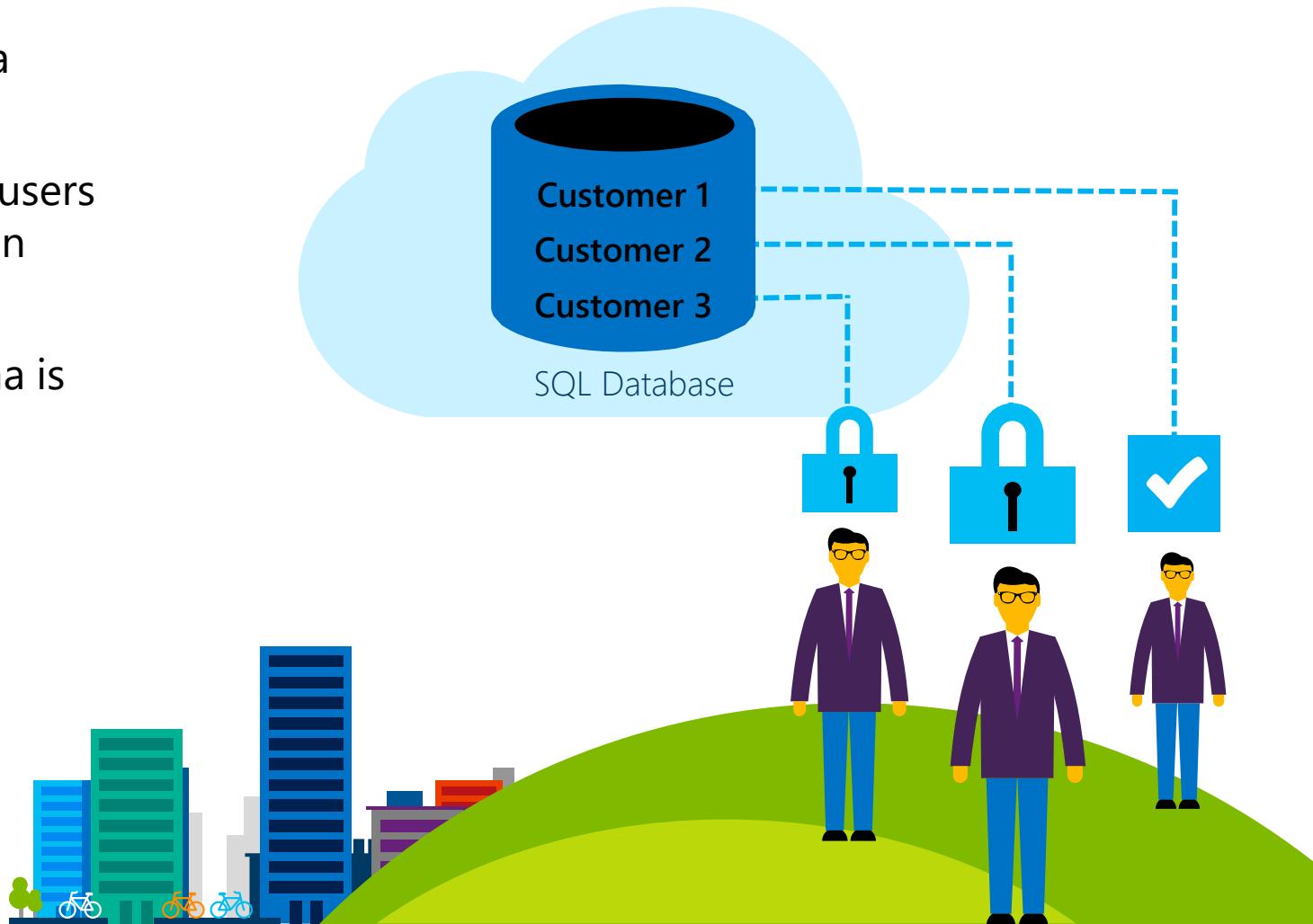
Row-Level Security enables customers to control access to rows in a database table based on the characteristics of the user executing a query

Row Level Security

Fine-grained access control over specific rows in a database table

Help prevent unauthorized access when multiple users share the same tables, or to implement connection filtering in multitenant applications

Enforcement logic inside the database and schema is bound to the table



Row Level Security

Predicate function

User-defined inline table-valued function (iTVF) implementing security logic
Can be arbitrarily complicated, containing joins with other tables



Performance?
Inline functions get optimized to provide comparable performance to views, as if the logic were directly embedded in the original query statement

Security predicate

Binds a predicate function to a particular table, applying it for all queries
Two types: filter predicates and blocking predicates (coming soon)

Security policy

Collection of security predicates for managing security across multiple tables

```
CREATE SECURITY POLICY mySecurityPolicy
    ADD FILTER PREDICATE dbo.fn_securitypredicate(wing, startTime, endTime)
    ON    dbo.patients
```

Row Level Security

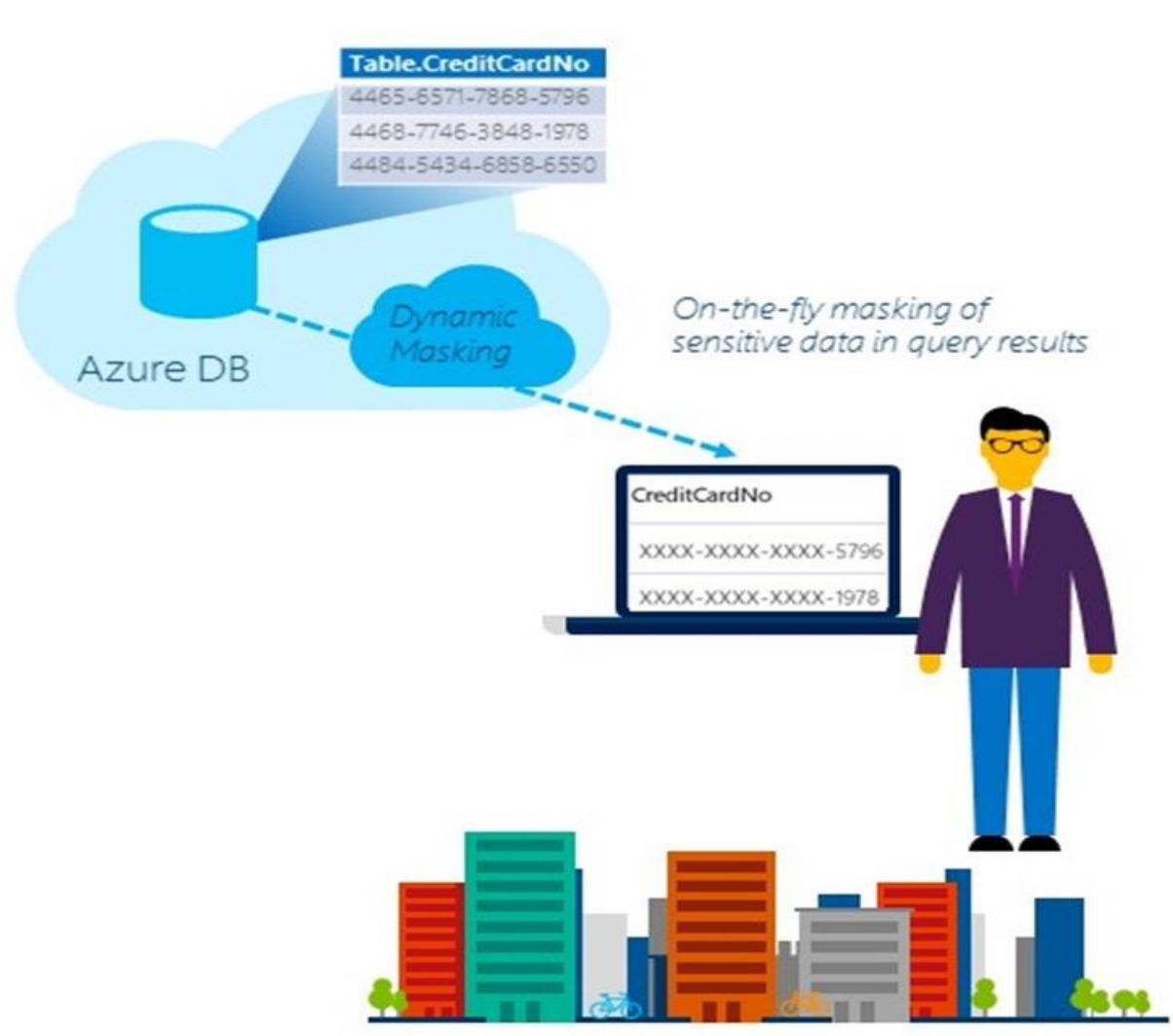
// Row Level Security

```
CREATE FUNCTION SecPred(@userId int)  
RETURNS TABLE  
WITH SCHEMABINDING  
AS  
RETURN SELECT 1 as valor WHERE @userId = user_id()
```

Dynamic Data Masking

Dynamic Data Masking is a policy-based security feature that helps limit the exposure of data in a database by returning masked data to non-privileged users who run queries over designated database fields

Dynamic Data Masking



Dynamic Data Masking

// Data Masking

```
CREATE TABLE Contacto  
  (ID int IDENTITY PRIMARY KEY,  
   Nombre varchar(100) MASKED WITH (FUNCTION = 'partial(1,"XXXXXXX",0)'),  
   Apellido varchar(100) NOT NULL,  
   NrTlf varchar(12) MASKED WITH (FUNCTION = 'default()'),  
   Email varchar(100) MASKED WITH (FUNCTION = 'email()'))
```

```
CREATE SECURITY POLICY [secpol] ADD FILTER PREDICATE  
[dbo].[SecPred]([UserId]) on [dbo].[Protegido]
```

Dynamic Data Masking

Compliance policies: When accessing customer or employees data

Dev & test with production data: To ensure compliance with sensitive data

SQL Auditing

Azure SQL Database Auditing tracks database events and writes audited events to an audit log in your Azure Storage account



SQL Auditing

Events

Plain SQL

Parameterized SQL

Stored procedure

Logins

Transaction management

SQL Auditing

The image displays three overlapping windows from the Azure portal, illustrating the configuration of SQL Auditing:

- Settings (SecurityDemo):** Shows a list of security features with "Auditing" selected.
- Auditing (SecurityDemo):** Shows the auditing configuration. It includes a note to enable auditing via Security Enabled Access, a checkbox for inheriting settings, and an "ENABLE AUDITING" switch set to "ON".
- Audit Logs Storage:** Shows the storage configuration for audit logs. It includes a "STORAGE ACCOUNT" section with "jdpassdemoazure", a "Retention (Days)" slider set to 730, a "Table Name" field containing "SQLDAuditLogs Jbcx642sb7Secur YYYYMMDD", and a "Storage Access Key" section with "Primary" selected.

SQL Auditing

ONLINE

SecurityDemo
SQL DATABASE

Settings Open in Visual ... Copy Restore Export Delete

Essentials

Resource group: Default-SQL-EastUS2
Status: Online
Location: East US 2
Subscription name: Visual Studio Ultimate with MSDN
Subscription ID: 424d0f78-5980-4d31-98ec-624616db8e74

Server name: jbcx642sb7.database.windows.net
Pricing tier: S0 Standard (10 DTUs)
Geo-replication role: Not configured
Server version: V12
Connection strings: Show database connection strings

All settings →

Auditing: 7 events

Dynamic Data Masking (preview)

Events: 8 events

Auditing SECURITYDEMO

Open In Excel Configure

Audit events last 24 hours

14.3% DATA CHANGES 1

SCHEMA CHANGES: 0

LOGIN FAILURE: 0

STORED PROCEDURE: 0

DATA CHANGES: 1

LOGIN SUCCESS: 0

DATA ACCESS: 0

SQL Auditing

Audit Records (Preview)

Showing Audit records up to Thu, 29 Oct 2015 08:46:58 GMT.

EVENT TIME	EVENT ID	PRINCIPAL NAME	EVENT TYPE	ACTION STATUS
Thu, 29 Oct 2015 08:46:58...	48ce7f75-3f02-49a0-8b11...	testlogin	DataAccess	Failure
Thu, 29 Oct 2015 08:46:58...	04e8fbb7-b6e0-4fb2-a23a...	testlogin	Login	Success
Thu, 29 Oct 2015 07:36:39...	8bae2f31-aec0-44aa-af63...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:36...	4b0f6bd9-71c9-4d25-9a44...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:34...	9a8b2c5b-ee02-4e28-8db...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:34...	78bf75f4-4f02-410f-b6ff-c...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:34...	0bb78eb5-6699-463e-9d8...	testlogin	Login	Success
Thu, 29 Oct 2015 07:36:27...	49ca2098-ab10-4999-995f...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:20...	2ad9b130-0730-4754-83d...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:19...	747847f9-0d46-4bb7-8b2...	testlogin	Login	Success

Load more

SQL Auditing

The screenshot shows a split interface in SQL Server Management Studio. On the left, a dark blue pane titled "Audit Record" displays a list of event details. On the right, a white pane titled "Query" shows the executed SQL statement.

Audit Record Details:

Event Time	Thu, 29 Oct 2015 08:46:58 GMT
Event ID	48ce7f75-3f02-49a0-8b11-561a176e6ee4
Event Type	DataAccess
Server Name	demo-server.sqltest-eq1.mscds.com
Database Name	demo-db
Principal Name	testlogin
Client IP	79.182.223.251
Application Name	.Net SqlClient Data Provider
Action Status	Failure
Failure Reason	Err 208, Level 16, State 1, Server demo-server, Line 1 I...
Response Rows	0
Response Volume	0
Affected Rows	0
Server Duration	00:00:00.0156359

Query Executed:

```
1 select * from employee where '1'='1' -- and '1'='1'
```

SQL Auditing

Source	Destination	Protocol
	NCPP	DNS
NCPP	westeurope1-a.control.database.wind...	TCP
westeurope1-a.control.data...	NCPP	TCP
NCPP	westeurope1-a.control.database.wind...	TCP
NCPP	westeurope1-a.control.database.wind...	TDS
westeurope1-a.control.data...	NCPP	TDS
NCPP	westeurope1-a.control.database.wind...	TLS
westeurope1-a.control.data...	NCPP	TLS
westeurope1-a.control.data...	NCPP	TCP
westeurope1-a.control.data...	NCPP	TCP
NCPP	westeurope1-a.control.database.wind...	TCP
NCPP	westeurope1-a.control.database.wind...	TLS
westeurope1-a.control.data...	NCPP	TLS
		SSDP
NCPP	westeurope1-a.control.database.wind...	TDS
westeurope1-a.control.data...	NCPP	TDS
NCPP	westeurope1-a.control.database.wind...	TCP
NCPP		DNS
westeurope1-a.control.data...	NCPP	TCP
westeurope1-a.control.data...	NCPP	TCP
NCPP	westeurope1-a.control.database.wind...	TCP
NCPP		DNS
NCPP	databsec-weu-2-a3.cloudapp.net	TCP

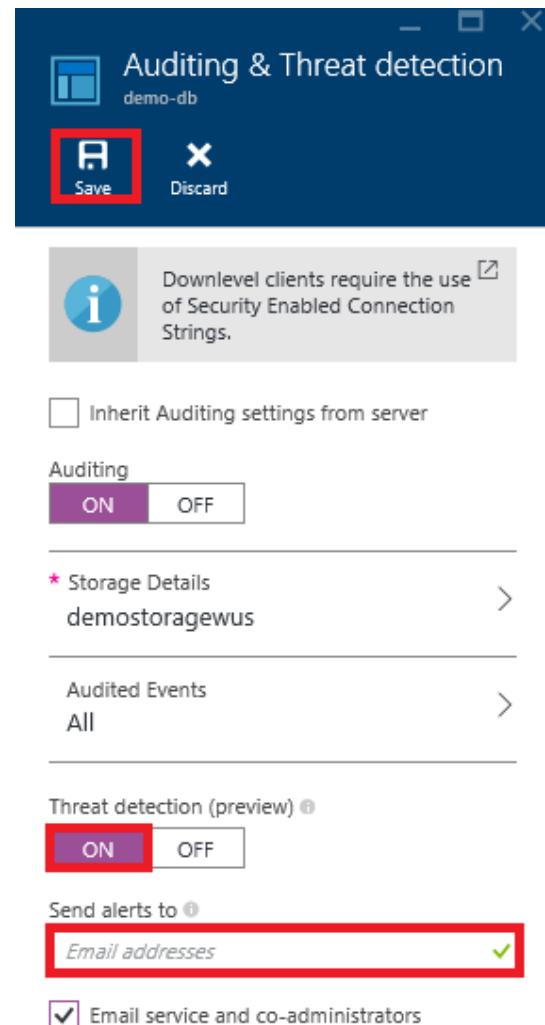


Thread Detection

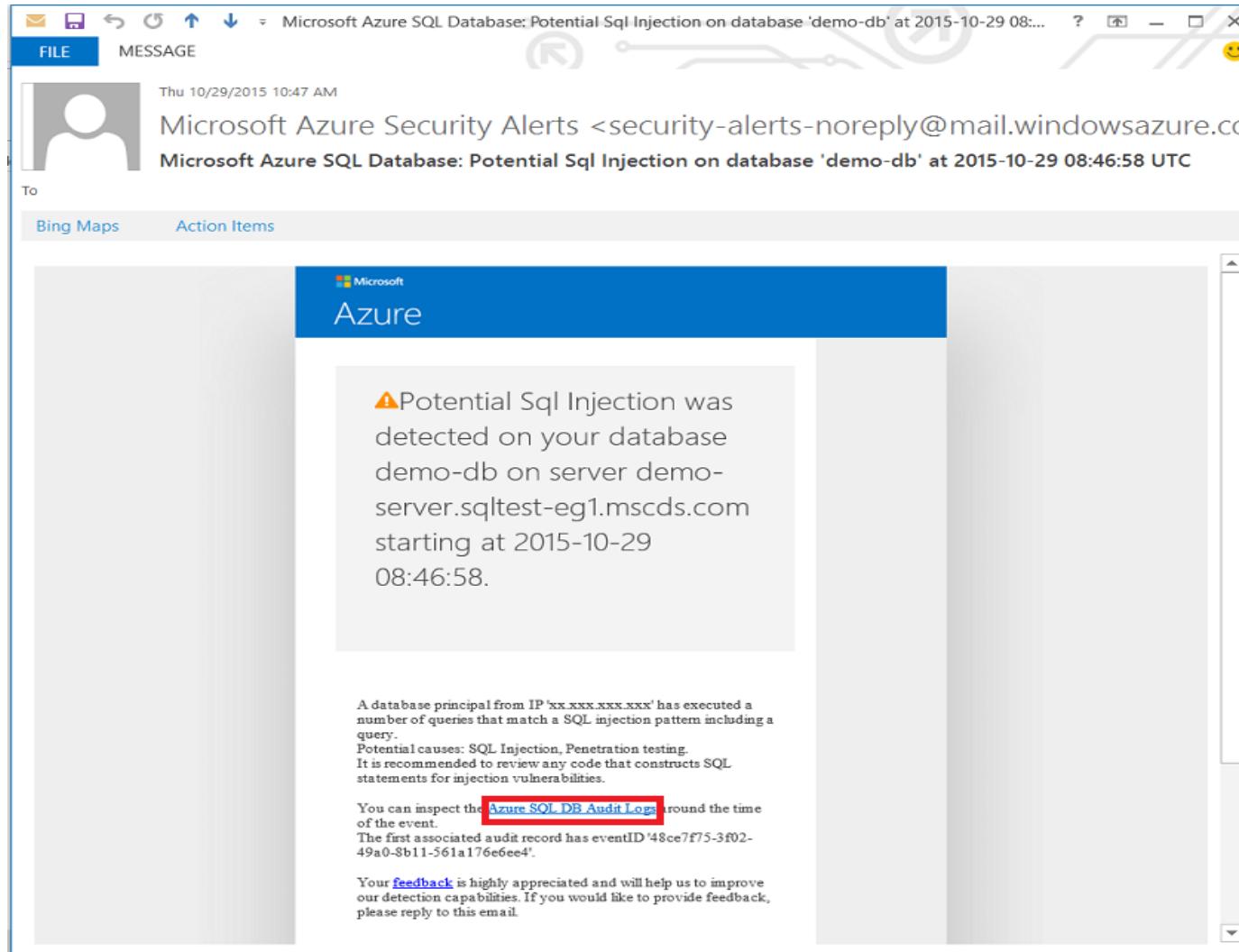
Threat Detection detects anomalous database activities indicating potential security threats to the database.



Thread Detection



Thread Detection



Thread Detection

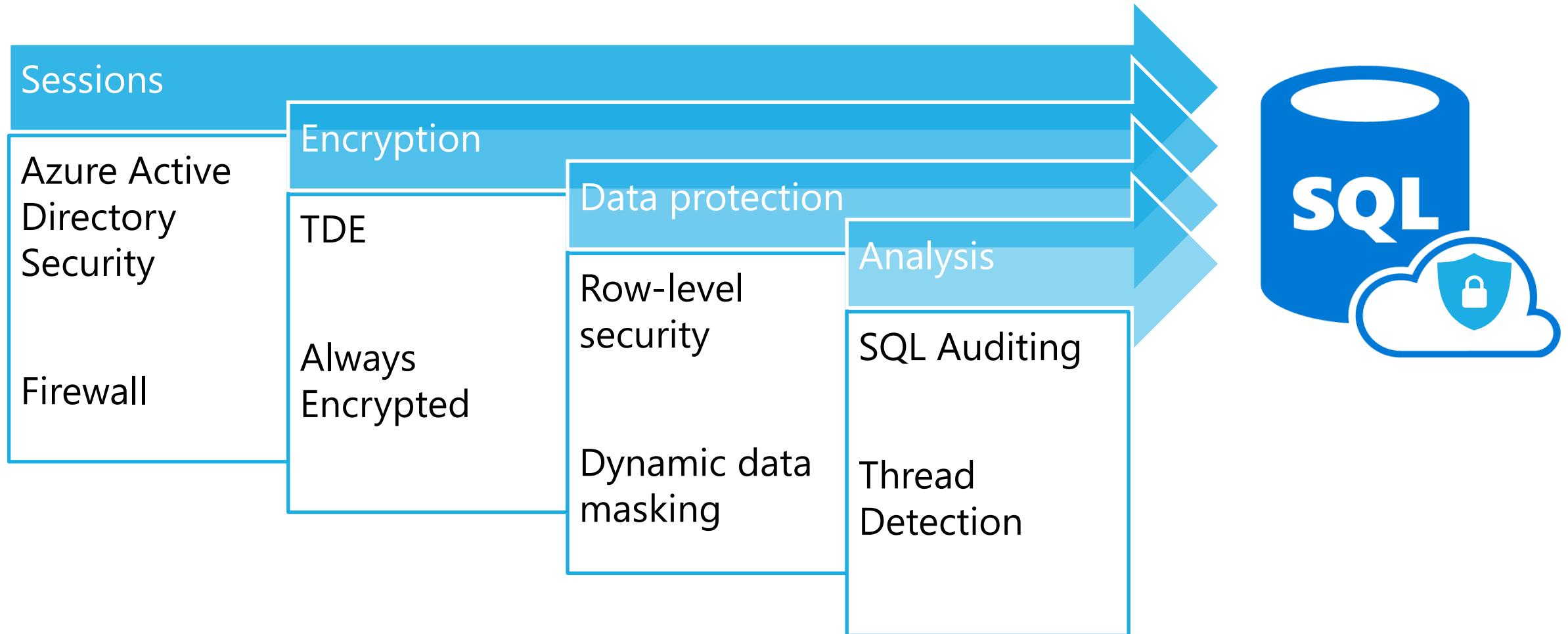
Audit Records (Preview)

Showing Audit records up to Thu, 29 Oct 2015 08:46:58 GMT.

EVENT TIME	EVENT ID	PRINCIPAL NAME	EVENT TYPE	ACTION STATUS
Thu, 29 Oct 2015 08:46:58...	48ce7f75-3f02-49a0-8b11...	testlogin	DataAccess	Failure
Thu, 29 Oct 2015 08:46:58...	04e8fbb7-b6e0-4fb2-a23a...	testlogin	Login	Success
Thu, 29 Oct 2015 07:36:39...	8bae2f31-aec0-44aa-af63...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:36...	4b0f6bd9-71c9-4d25-9a44...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:34...	9a8b2c5b-ee02-4e28-8db...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:34...	78bf75f4-4f02-410f-b6ff-c...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:34...	0bb78eb5-6699-463e-9d8...	testlogin	Login	Success
Thu, 29 Oct 2015 07:36:27...	49ca2098-ab10-4999-995f...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:20...	2ad9b130-0730-4754-83d...	testlogin	DataAccess	Success
Thu, 29 Oct 2015 07:36:19...	747847f9-0d46-4bb7-8b2...	testlogin	Login	Success

[Load more](#)

Summary



Geo-Replication

Chapter



GEO-Replication

- The GeoDR feature of Windows Azure SQL Database is a database-level disaster recovery solution.
 - Enables you to configure any user database for full redundancy using different SQL Database servers within the same region or in different regions.
 - Uses the continuous copy mechanism to provide redundancy on a different SQL Database server within the same Windows Azure region or in different regions (geo-redundancy).
 - Enables applications to recover from a temporary outage or even the permanent loss of a datacenter.
 - GeoDR asynchronously replicates committed transactions from an original database to up to four continuous copies of the database.
 - The original database becomes the source database of the continuous copy. Each continuous copy is known as a replica database.
 - Then SQL Database continues to asynchronously replicate the committed transactions from the source database.
 - While this relationship exists, automatic data synchronization occurs between the source and replica databases. At any given point the replica data might be slightly behind of the source database (less fresh), but the replica data is guaranteed to always be transactionally consistent.
 - Like the source database, the replica database comprises a primary copy with two secondary copies. However, this primary is directly updated by the continuous copy mechanism and cannot accept any application-initiated updates.
 - If the source database becomes unavailable, terminating the continuous copy relationship for a given replica database makes the replica database a standalone database. At this point, the

GEO-Replication

The login on the secondary must be created with the same SID as on the primary. For example:

On the primary in the context of the user database (and assuming that the database user name matches the corresponding login name):

```
SELECT sid  
FROM sys.sql_logins  
WHERE name = 'querysvcreader';
```

On the secondary:

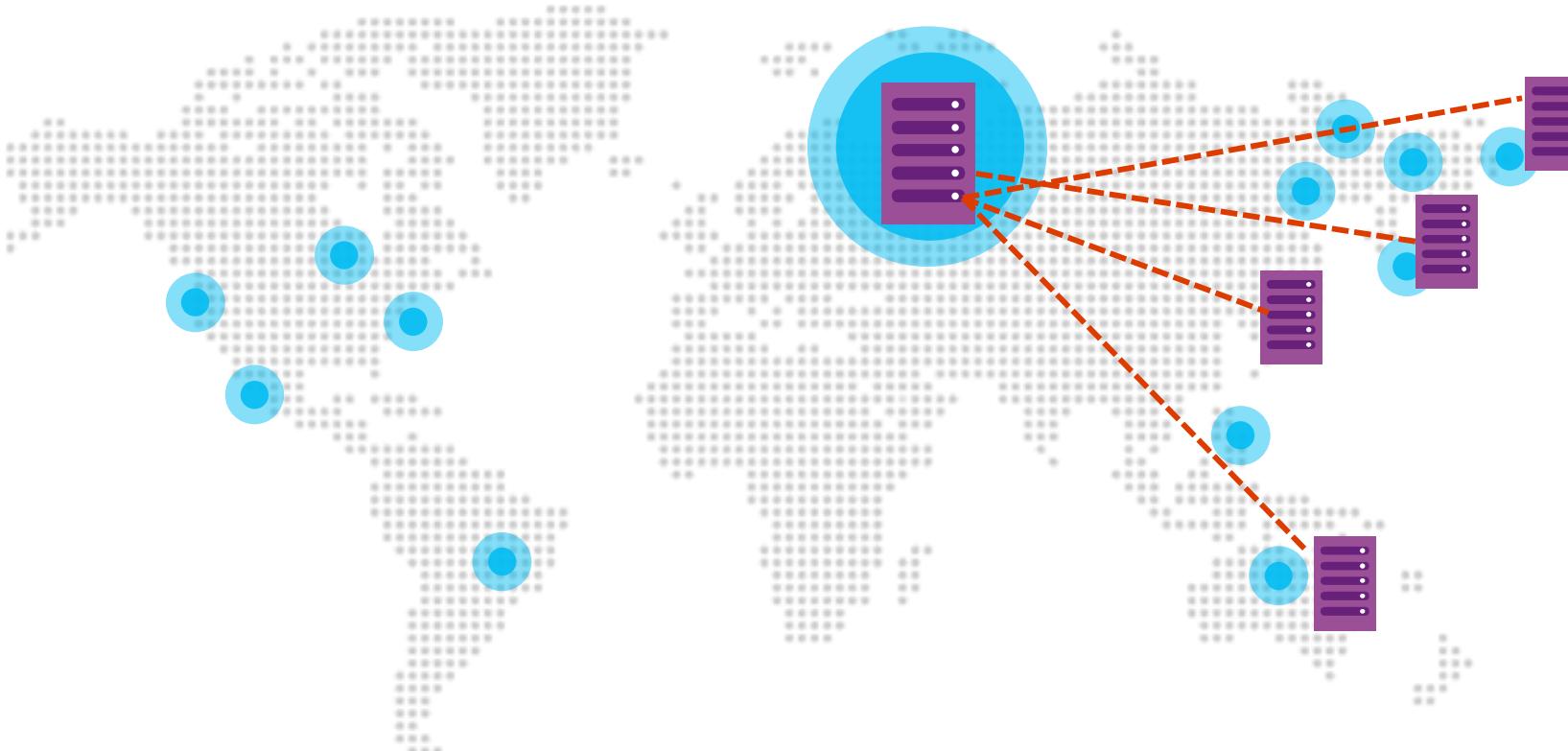
```
CREATE LOGIN querysvcreader WITH PASSWORD = 'pwd', SID =  
0xFEDA06D0AA3A3046A965BF7DD9469531;
```

Where the SID value is the output from the first query.

[Information here](#)



GEO-Replication



- [Script para realizar Failover Premium en Azure SQL](#)
- [Recuperación de una base de datos Azure SQL tras un outage](#)
- [Geo-replicación estándar en bases de datos Azure SQL](#)
- [Geo-restore de bases de datos Azure SQL](#)
- [Configure geo-replication for Azure SQL Database with the Azure Portal](#)

Backup and Restore

- **Backup Full, Differential and Transaction Log every 3-5 minutes.**
- Situations with **Long-Transaction with long RollBack process.**
- We have a **limited retention period**, in our roadmap, we planed an unlimited retention period.

	Basic	Standard				Premium				
		S0	S1	S2	S3	P1	P2	P4	P6/P3	P11
Maximum database size	2 GB	250 GB				500 GB				1 TB
DTUs	5	10	20	50	100	125	250	500	1,000	1,750
Point-in-time restore	Any point last 7 days	Any point last 14 days				Any point last 35 days				
Disaster recovery	Geo-Restore, restore to any Azure region	Standard Geo-Replication, offline secondary				Active Geo-Replication, up to 4 online (readable) secondary backups				
Max In-Memory OLTP storage	NA	NA	NA	NA	NA	1 GB	2 GB	3 GB*	8 GB	10 GB*
Max concurrent requests	30	60	90	120	200	200	400	800	1,600	2,400
Max concurrent logins	30	60	90	120	200	200	400	800	1,600	2,400
Max sessions	300	600	900	1,200	2,400	2,400	4,800	9,600	19,200	32,000

* In-Memory OLTP storage limits will soon adjust to 4 for P4 and 14 for P11.

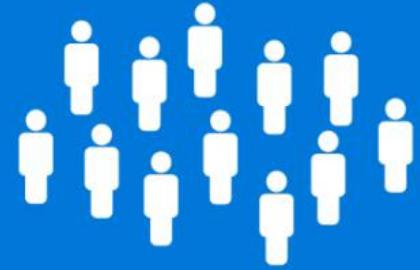
- **Point-In-Time Recovery:**

- Servers, Databases and deleted databases.

Elastic Database Pool

Chapter





↑

↑

↑

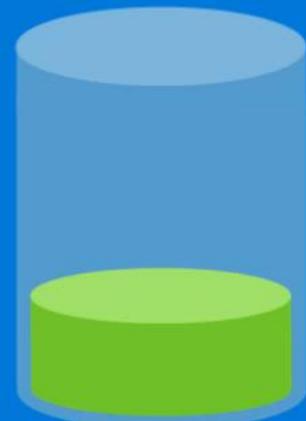
↑



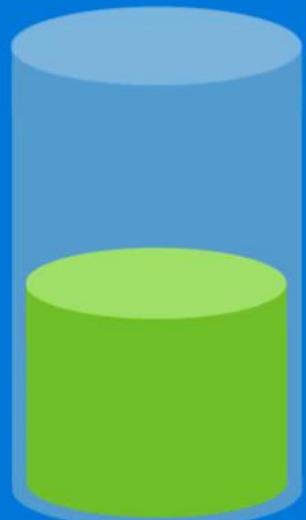
S0



S1



S2



S3

ELASTIC DATABASE POOL

MAX
100 DTUs

MIN
10 DTUs



Elastic Databases

A pool is given a set number of eDTUs, for a set price.

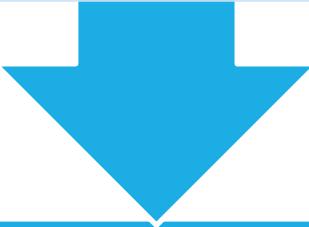
Elastic Databases

eDTU consumption

Individual databases are given the flexibility to auto-scale within set parameters.

Under heavy load a database can consume more eDTUs to meet demand.

Databases under light loads consume less, and databases under no load don't consume any eDTUs.



Provisioning resources for the entire pool rather than for single databases simplifies your management tasks. Plus you have a predictable budget for the pool.

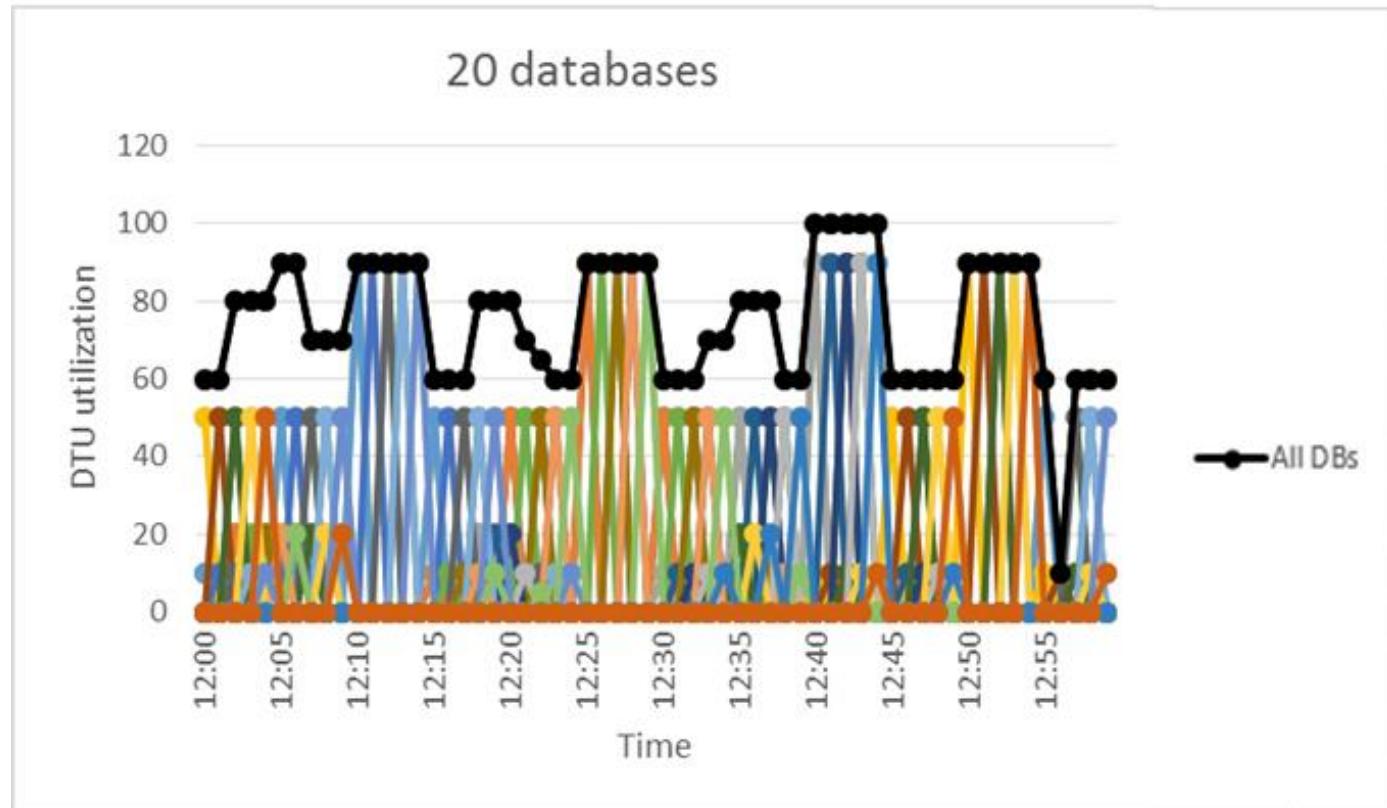
Elastic Databases

PROPERTY	BASIC	STANDARD	PREMIUM
dtu	100, 200, 400, 800, 1200	100, 200, 400, 800, 1200	125, 250, 500, 1000, 1500
databaseDtuMax	5	10, 20, 50, 100	125, 250, 500, 1000
databaseDtuMin	0, 5	0, 10, 20, 50, 100	0, 125, 250, 500, 1000
storageMB*	10000 MB, 20000 MB, 40000 MB, 80000 MB, 120000 MB	100 GB, 200 GB, 400 GB, 800 GB, 1200 GB	62.5 GB, 125 GB, 250 GB, 500 GB, 750 GB
storage per DTU	100 MB	1 GB	.5 GB
max databases per pool	200	200	50

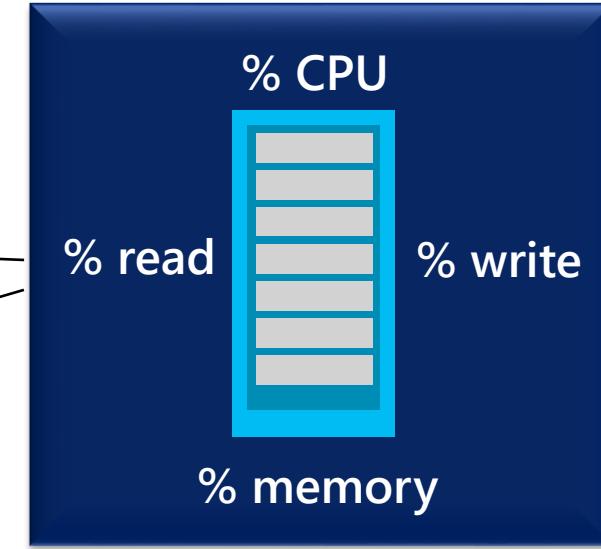
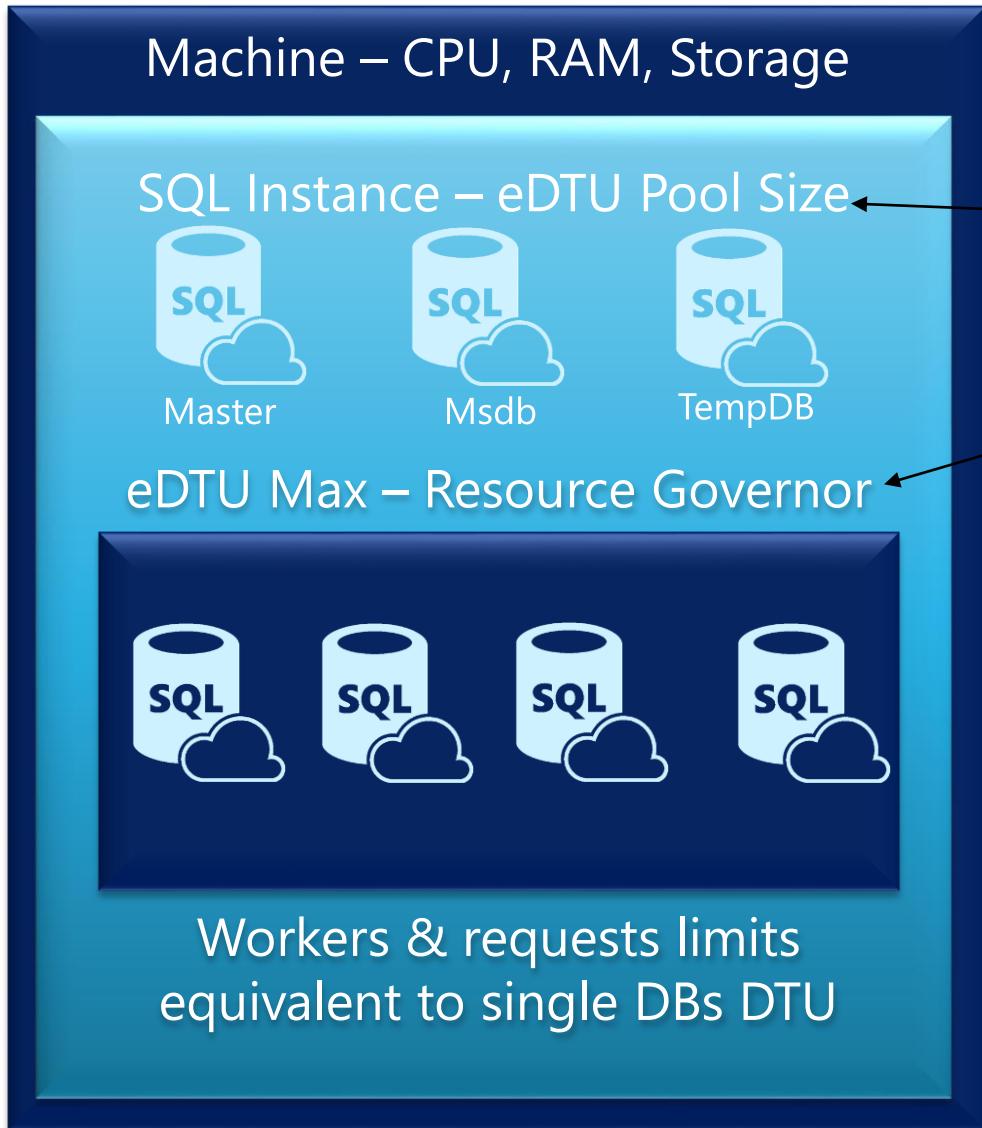
Default values are **bold**.

*units in API are MB, not GB.

Elastic Databases



Elastic Databases



Elastic Databases

```
SELECT end_time , (SELECT Max(v) FROM (VALUES (avg_cpu_percent)
, (avg_data_io_percent) , (avg_log_write_percent) ) AS value(v)) AS
[avg_DTU_percent] FROM sys.dm_db_resource_stats ORDER BY
end_time DESC;
```

To find a resource usage:

- a) sys.dm_db_resource_stats. Has 15 second granularity.
- b) sys.resource_stats has 5 minute granularity.
- c) sys.elastic_pool_resource_stats for elastic pool databases.

Elastic Database Query

Chapter



Elastic database query

The elastic database query feature enables you to run a Transact-SQL query that spans multiple databases in Azure SQL Database



Elastic Database query

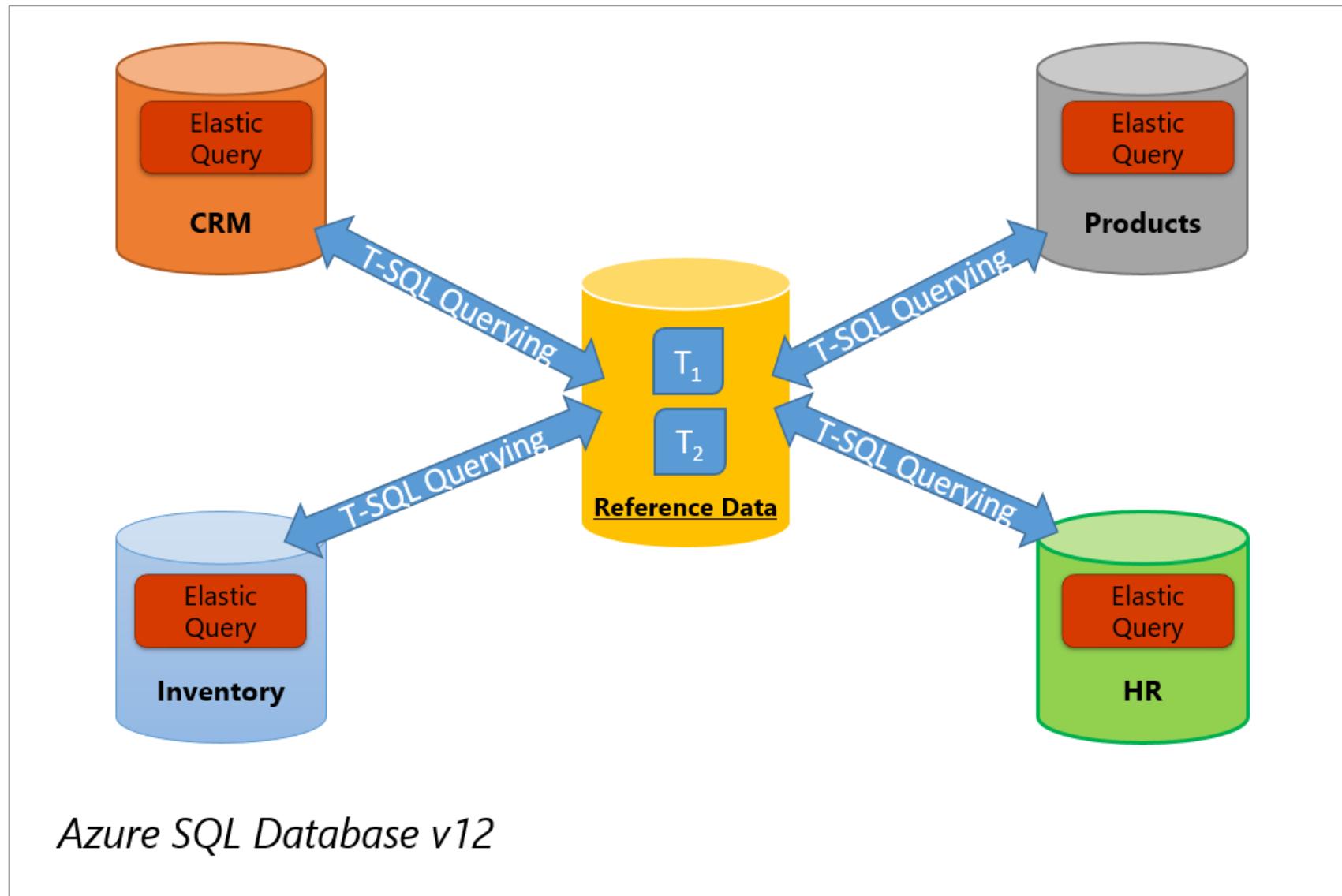
Uses

Elastic database query

- **Cross-database** query: read-only querying of remote databases.
- **Migrate** applications: using three- and four-part names or linked server to SQL DB.
- **Push SQL parameters:** to remote databases
- **Sp_execute_fanout:** parameters similar to sp_executesql
 - a) [Announcement blog post](#)
 - b) [Step-by-step tutorial](#)
 - c) [Overview documentation](#)

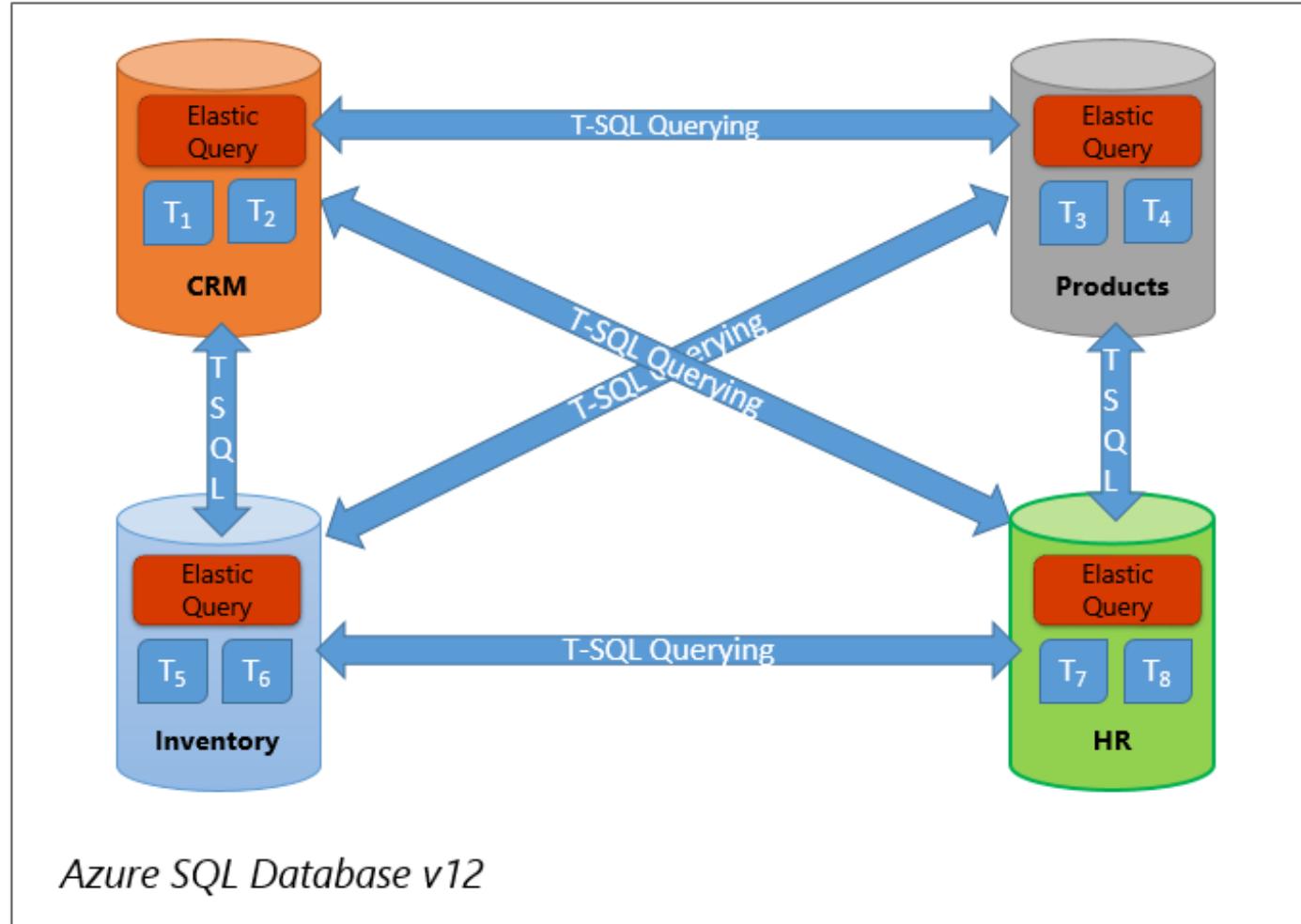
Elastic database query

Vertical partitioning



Elastic database query

Vertical partitioning



Elastic database query

Vertical partitioning

CREATE MASTER KEY mymasterkey

CREATE DATABASE SCOPED CREDENTIAL mycredential

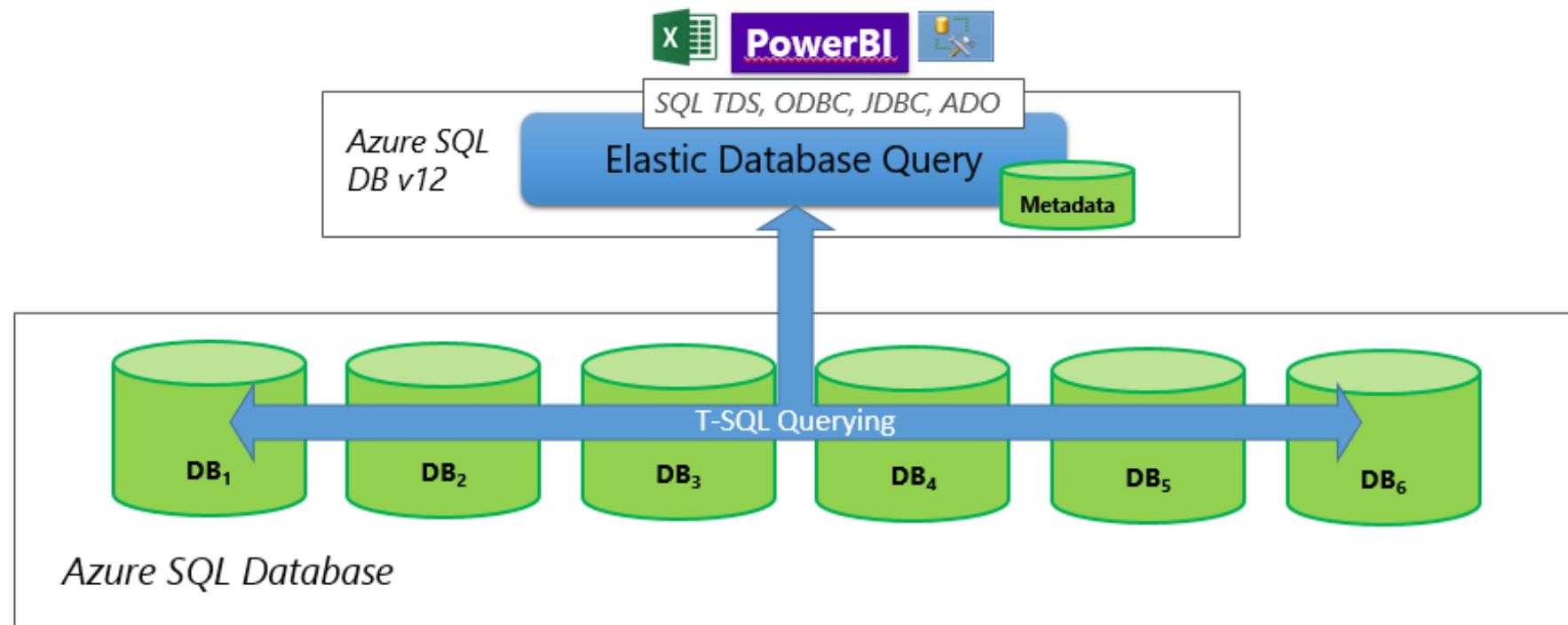
CREATE/DROP EXTERNAL DATA SOURCE mydatasource (**RDBMS**)

CREATE/DROP EXTERNAL TABLE mytable

Query the table: Read-only

Elastic database query

Horizontal partitioning



Elastic database query

Horizontal partitioning

CREATE MASTER KEY mymasterkey

CREATE DATABASE SCOPED CREDENTIAL mycredential

Create a shard map (elastic database client library)

CREATE/DROP EXTERNAL DATA SOURCE
mydatasource (**SHARD_MAP_MANAGER**)

CREATE/DROP EXTERNAL TABLE mytable

Query mytable: Read only

Elastic Jobs

Chapter



Elastic jobs

Elastic Database jobs is a customer-hosted Azure Cloud Service that enables the execution of ad-hoc and scheduled administrative tasks, which are called jobs

Elastic jobs



Elastic jobs

- **Administrative:** Deploy new schema
- **Update** data: product information common across all databases
- **Schedule:** automate updates after hours
- **Rebuild** indexes: regularly across a collection of databases
- **Centralize** query results: from many databases into a central table
- **Reference:** <https://azure.microsoft.com/en-us/documentation/articles/sql-database-elastic-jobs-powershell/>
- **Service:** <https://azure.microsoft.com/en-us/documentation/articles/sql-database-elastic-jobs-service-installation/>
- **Nuget:** <https://www.nuget.org/packages/Microsoft.Azure.SqlDatabase.Jobs/>

Elastic jobs

- Script must be able to run multiple times, even if it has failed before
- When a script fails, the job will be automatically retried until it succeeds (within limits, as the retry logic will eventually cease the retrying)

Elastic jobs

```
IF EXISTS (SELECT name FROM sys.indexes  
          WHERE name = N'IX_ProductVendor_VendorID')  
DROP INDEX IX_ProductVendor_VendorID ON Purchasing.ProductVendor;  
GO  
CREATE INDEX IX_ProductVendor_VendorID  
ON Purchasing.ProductVendor (VendorID);
```

Elastic Scale

Chapter

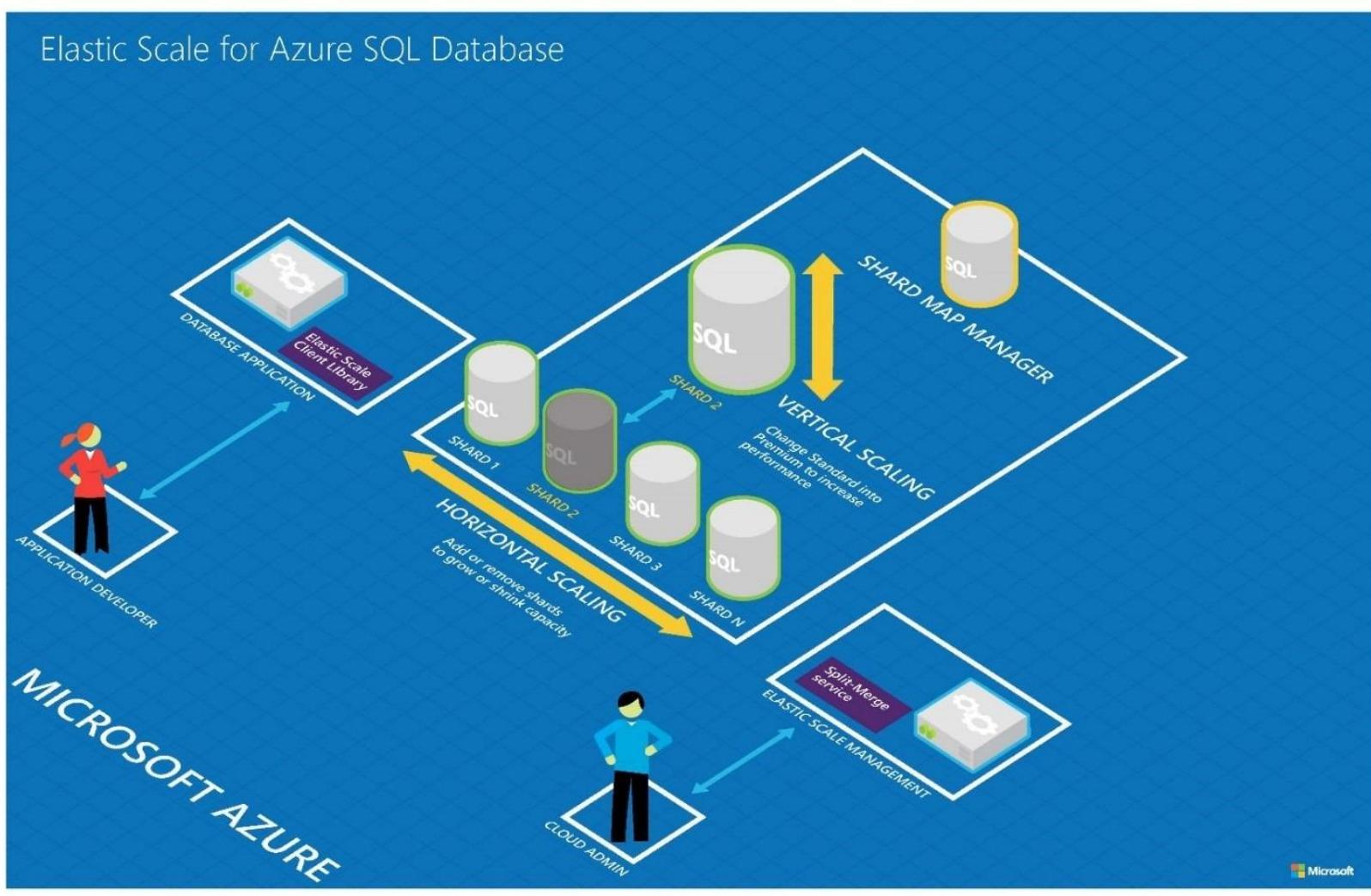


Elastic Escalation

The new Elastic Scale capabilities simplify the process of scaling out (and in) a cloud application's data tier.

Elastic Scale offers functionality to help you build and maintain applications that can span a few databases or thousands.

Elastic Escalate



Elastic Escalation

High-volume OLTP: Single application process massive data volumes by employing #number of database shards.

Multi-tenant SaaS: Each customer is assigned their own database for higher isolation

Continuous data collection: New shards for new date ranges. Newer shards can use higher service tiers, and scale down over time as usage is reduced

Elastic Escalation

Global Shard Map (GSM): Database as the repository for all of its shard maps and mappings.

The tables are in a special schema named **_ShardManagement**.

Local Shard Map (LSM): Every database in a shard contain metadata of the shard Redundant with the information in the GSM

Application cache: Application maintains a local in-memory cache Stores routing information that has recently been retrieved.

Elastic Escalation

Key	Shard Location
1	Database_A
3	Database_B
4	Database_C
6	Database_B
...	...

Elastic Database Transaction

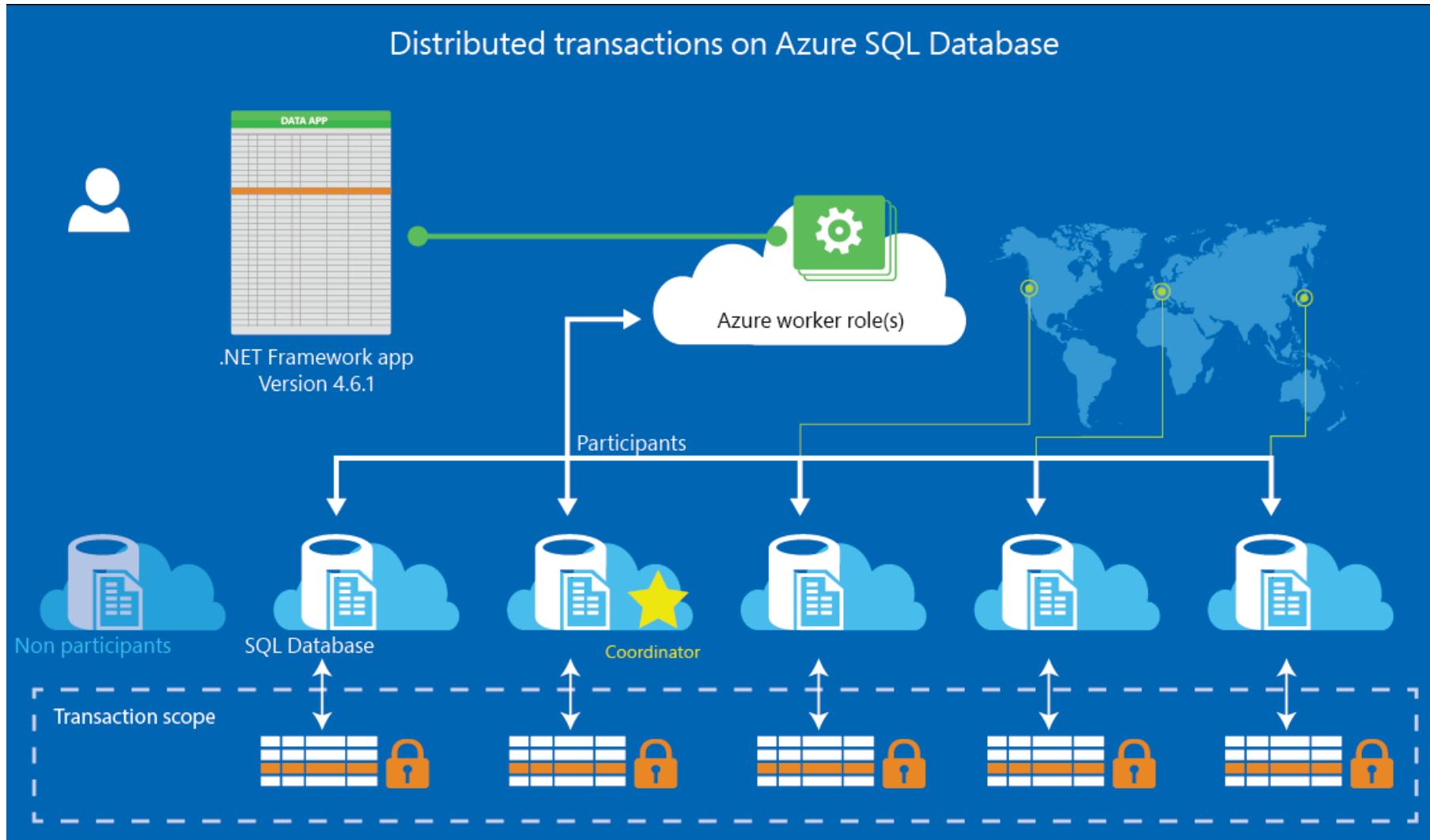
Chapter



Elastic Database Transactions

Elastic database transactions for Azure SQL Database (SQL DB) allow you to run transactions that span several databases in SQL DB

Elastic Database Transactions



Elastic Database Transactions

Available for **.NET 4.6.1** applications: ADO .NET ->System.Transaction classes

Coordinate distributed transactions is **directly integrated** into SQL DB As MSDTC is not an Azure PaaS.

Applications: Connect to any SQL Database-> Launch distributed transactions-> one of the databases will coordinate **transparently**

Elastic Database Transactions

Multi-database applications: Vertically partition

Sharded database applications: Horizontal partitioning

Elastic Databases Transactions

Only SQL DB supported

Not Other X/Open XA resource providers

Not SQL Server

Only **client-coordinated** transactions

Only databases on **Azure SQL DB V12** are supported.

Elastic Databases Transactions

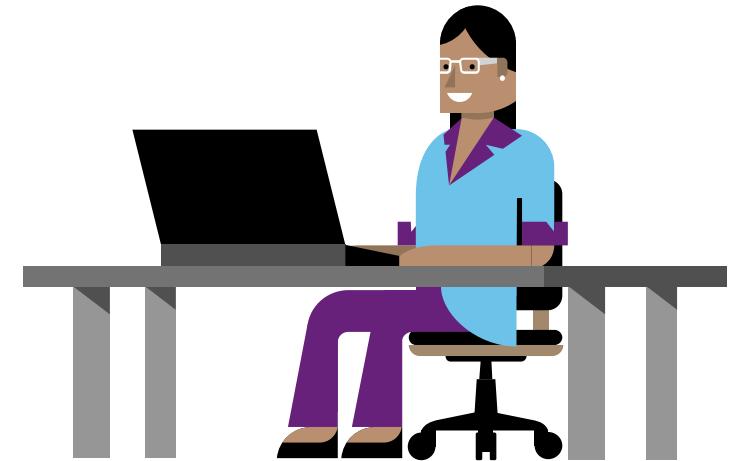
```
using (var scope = new TransactionScope())          {  
    using (var conn1 = new SqlConnection(connStrDb1))  
    {  
        conn1.Open();  
        SqlCommand cmd1 = conn1.CreateCommand();  
        cmd1.CommandText = string.Format("insert into T1 values(1)");  
        cmd1.ExecuteNonQuery();  
    }  
  
    using (var conn2 = new SqlConnection(connStrDb2))          {  
        conn2.Open();  
        var cmd2 = conn2.CreateCommand();  
        cmd2.CommandText = string.Format("insert into T2 values(2)");  
        cmd2.ExecuteNonQuery();  
    }  
  
    scope.Complete();
```

Connectivity & Performance using .NET

Chapter



Connectivity & Performance using .NET



Connectivity - Improvements

- **Connectivity**

- Retry-Logic Policy, incremental, custom timeout and exponential.
- Error Handling and custom retry-logic.

.NET 4.5.1 or superior: tenemos: ConnectRetryCount y ConnectRetryInterval

.NET 4.6.1, política de reintentos incluido en SqlConnection.Open
Entity Framework 6.0: SqlAzureExecutionStrategy

- **Performance**

SqlConnection object properties

Stopwatch

EF: Optimizar query performance y modelo



Connectivity&Performance - Improvements

//Llamada al principio del procedimiento y/o función.

```
Stopwatch stopWatch = new Stopwatch();
    stopWatch.Start();
    stopWatch.Stop();
```

// Obtenemos el tiempo de la ejecución como un TimeSpan.

```
TimeSpan ts = stopWatch.Elapsed;
```

//Llamada al diccionario de propiedades de la conexión

```
IDictionary currentStatistics = awConnection.RetrieveStatistics();
long bytesReceived = (long)currentStatistics["BytesReceived"];
    long bytesSent = (long)currentStatistics["BytesSent"];
    long selectCount = (long)currentStatistics["SelectCount"];
    long selectRows = (long)currentStatistics["SelectRows"];
```

Performance using Azure SQL DB

Chapter



Common Performance Issues

Performance issues

- ✓ Query tuning, investigating execution plans.
- ✓ Query Timeouts.
- ✓ Blocking.
- ✓ Deadlocks?.
- ✓ Database Tier in Azure SQL Database
- ✓ Incorrect configuration in SQL Server.
- ✓ Hardware Issues in SQL Server.

Way to resolve

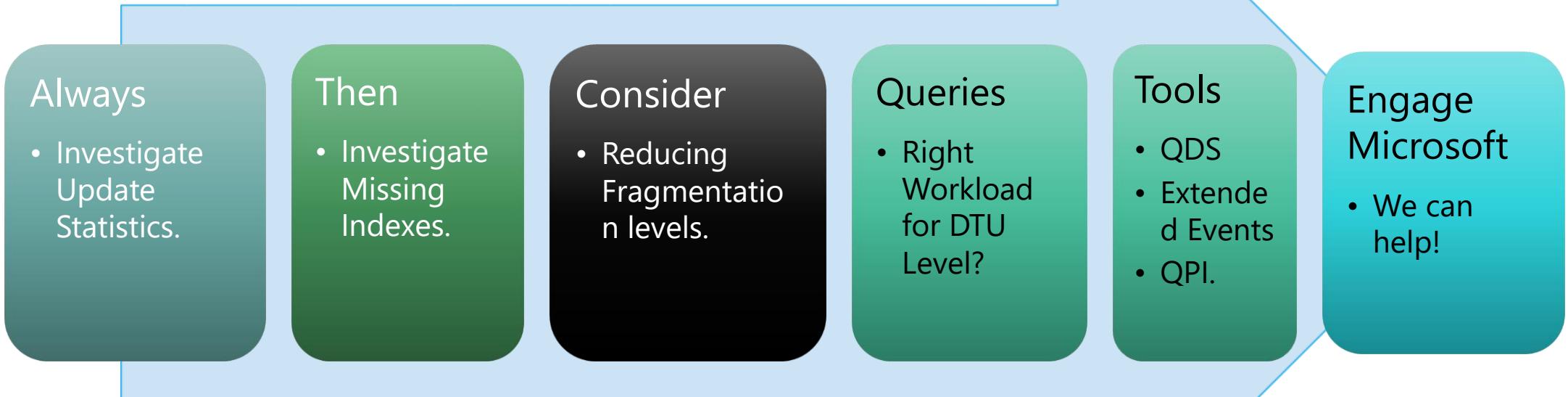
- ✓ Blocking - SDP, SQL Nexus
- ✓ Attentions, Errors, etc. – Xevents.
- ✓ Query behavior – QDS, DMVs

Frequently Simple

- ✓ Outdated statistics.
- ✓ Missing Indexes.
- ✓ Throttling for Azure SQL Database.



Troubleshooting Steps



Troubleshooting Steps

Monitoring Tools

SQL Nexus - SQL Perf Main

Overall Resource Consumption [basicdb] X SQLQuery1.sql - css...TER (rocavalc (59)) Batch_RPC_Only.sql - not connected

Overall resources consumption during the last month for database basicdb

Calls to Remote Dependencies

TYPE	DEPENDENCY	DURATION	SUCCESS
SQL	SQL: tcp://...database.windows.net,1433 S...L...	337 ms	✓
SQL	SQL: tcp://...database.windows.net,1433 S...L...	318 ms	✓
SQL	SQL: tcp://...database.windows.net,1433 S...L...	358 ms	✓
SQL	SQL: tcp://...database.windows.net,1433 S...L...	409 ms	✓
SQL	SQL: tcp://...database.windows.net,1433 S...L...	477 ms	✓
SQL	SQL: tcp://...database.windows.net,1433 S...L...	484 ms	✓

Duration_units

CPUTime_units

Ready



Query Performance Insight

basicdb - Past day - PREVIEW

Settings Refresh Index Advisor Feedback

CPU consumption: overall and top 5 queries



OVERALL DTU 3.15 %

Performance of top queries

#	QUERY ID	CPU[%]	DURATION[HH:MM:SS]	EXECUTIONS COUNT
1	1387	0.24	00:03:39.180	1
2	664	0	00:00:00.140	575
3	584	0.01	00:00:00.240	123
4	1386	0.13	00:01:58.620	1
5	1381	0	00:00:00.040	139

Index recommendations

basicdb

Advisor settings Feedback

Recommended indexes

IMPACT	RECOMMENDED ACTION	TARGET OBJECT	INDEXED COLUMNS
IMPACT	RECOMMENDED ACTION	TARGET OBJECT	INDEXED COLUMNS

We analyzed your database recently. There has been no activity in your database to make any recommendations at this time.

Getting Started with Query Data Store

Chapter



Query Data Store

✓ It is a ***persisted database with query execution information for***

- ✓ SQL Server 2016 (Preview)
- ✓ Azure SQL DB V12

✓ **Query Performance Tuning and Troubleshooting**

- ✓ SQL Profiler replacement in some parts.
- ✓ Minimum impact for SQL Engine in SQL Server or Azure SQL DB.
- ✓ Other tools like Query Performance Insight, Index Advisor are using QDS Information.
- ✓ SSMS Reports Supported for SQL Server and Azure SQL Database.
- ✓ Catalog Views all available.
- ✓ Active or disable this feature demand.
- ✓ Use Extended Events to capture the information.



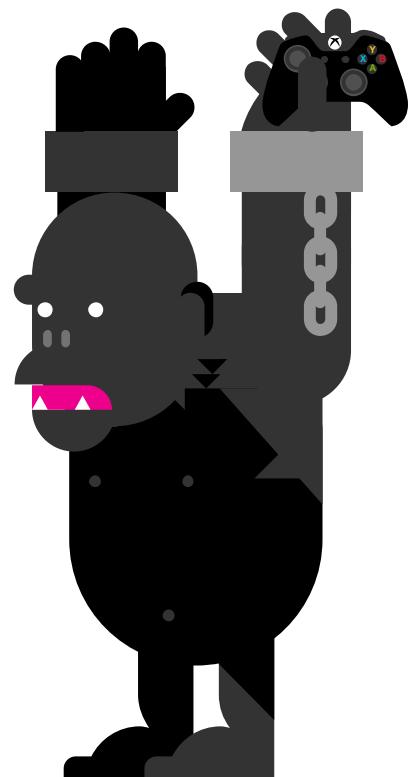
Information collected

✓ **A Query Flight Recorder to store:**

- ✓ Queries/statements, Query Plans, Runtime statistics with connection properties.
- ✓ Maintains the information even an instance restart.
- ✓ Just save DML: SELECT, DELETE, INSERT, UPDATE.
- ✓ Doesn't record details about: DDL: BULK INSERT or DBCC commands.
- ✓ The information will be recorded with a limited time.
- ✓ Details of queries whether they are cached or not.
- ✓ Analyze the history
- ✓ Look at properties and statistics for queries not available in DMVs
- ✓ Allows a simpler and more robust method to force plans vs plan guides
- ✓ Independent of what is cached
- ✓ Statement text appears in parameterized form
- ✓ Including cursors and SET assignment
- ✓ Persistence is async
- ✓ Works with In-Memory tables in Azure SQL DB and SQL SERVER 2016.

✓ **Other Information**

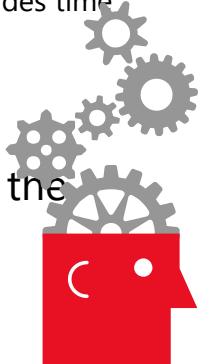
- ✓ Use the scenarios [page](#)
- ✓ The docs. Start [here](#)
- ✓ Query Store [Best Practices](#)



CAPTURE MODE

Query Capture Mode

- ✓ Query Store supports adjustable query capture.
 - ✓ The goal of capture option is to provide users with additional control over ingress of new queries with main value prop to enable collection of data that is highly relevant.
- ✓ This is the list of supported values:
 - ✓ ALL - all queries are captured. Infrequent queries and queries with insignificant compile and execution duration are ignored. Thresholds for execution count, compile and runtime duration are internally determined.
 - ✓ AUTO - capture relevant queries based on execution count and resource consumption. This is the default configuration value
 - ✓ NONE: stop capturing new queries. Query Store will continue to collect compile and runtime statistics for queries that were captured already.
- ✓ To change capture mode use one of the following statements:
 - ✓ ALTER DATABASE <database_name> SET QUERY_STORE (QUERY_CAPTURE_MODE = AUTO);
 - ✓ **Query capture mode** of AUTO (Default is ALL) = capture relevant queries based on execution count and resource consumption. NONE = Stop capture new queries but track and persist stats for existing.
 - ✓ ALTER DATABASE <database_name> SET QUERY_STORE (QUERY_CAPTURE_MODE = NONE);
 - ✓ **Sized based cleanup** of AUTO (Default is OFF) = Remove oldest queries and least expensive when 90% of max (until 80% reached). Overrides time retention (days kept in store)
- ✓ Query capture = AUTO; Query retention = 30 days; Sized based cleanup = ON
- ✓ We do limit memory for Query Store and may revert to read-only mode memory overcommitted or when the disk is full.



Troubleshooting with Query Data Store

Chapter



Dynamic Management Views

sys.database_query_store_options

- Exposes various Query Store configuration options for user database.

sys.query_store_query_text

- Exposes information about captured query text for a query

sys.query_context_settings

- Exposes information about content settings

sys.query_store_query

- Exposes information about queries in every execution

sys.query_store_plan

- Exposes information about different plans SQL Server used to execute queries in the system.

sys.query_store_runtime_stats_interval

- Returns all intervals created in Query Store

sys.query_store_runtime_stats

- Returns runtime statistics for executed query plans, aggregated on per-interval basis.

Extended Stored Procedures

Force plan

- **sp_query_store_force_plan @query_id, @plan_id**

Stop plan forcing

- **sp_query_store_unforce_plan @query_id, @plan_id**

Clear all data for a single query

- **sp_query_store_remove_query @query_id**

Remove a single plan

- **sp_query_store_remove_plan @plan_id**

Clear all stats

- **sp_query_store_reset_exec_stats @plan_id**

Flush data

- **sp_query_store_flush_db**

QDS vs DMVs

Using sys.query_context_settings

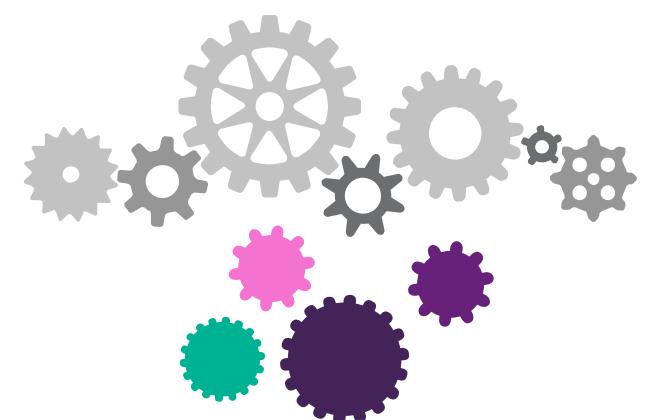
set_options

cursor_options - Go [here](#) to interpret bitmask

schema_id

Comparing DMVs vs Query Store

- ✓ Compilation statistics
- ✓ Plan properties including engine version and database compatibility level
- ✓ Details and stats for plan variations used for a query over time
- ✓ Recorded as queries are compiled
- ✓ Record query execution stats ***even if aborted***



Maintenance

Enabling, Clearing, and State

- ✓ **ON** = Enable; **OFF** = Disabled (existing state and data kept); **CLEAR** = TRUNCATE tables
- ✓ **READ_WRITE** = Default when ON; **READ_ONLY** = intentional or *problem* (desired != actual)

Size, limits, and retention

- ✓ Default **max size** = 100Mb (limited by overall database size). If you hit max state = READ_ONLY
- ✓ Default **max plans per query** = 200
- ✓ Default **days queries kept in store** = 367 days



SQL Azure Database V12 - Code Best Practices – Tips & Tricks

- **Batching considerations:** <https://azure.microsoft.com/en-us/documentation/articles/sql-database-use-batching-to-improve-performance/>
- **Performance considerations:** <https://azure.microsoft.com/en-us/documentation/articles/sql-database-performance-guidance/>
- **To avoid any parameter sniffing issue.**
<http://blogs.technet.com/b/mdegre/archive/2012/03/19/what-is-parameter-sniffing.aspx>
- **Link describing Performance Considerations with Entity Framework – with regards to optimize query performance and model.** <http://msdn.microsoft.com/en-us/library/cc853327.aspx>

Extended Events

Chapter



Extended Events

- You have the documentation in this [URL](#)
- In order to use Extended Events with your SQL Database server there are three basic steps:
 - Setup a storage account and container
 - Enable access to storage from SQL Database
 - Collect Data using Extended Events

Setup storage account and container

XEvents use blob storage as their target. You shall need access to a storage account that will be the target for writing out the XEL files. Typically, this shows up under the Storage node in Management Portal. For example, I have just created storage account named cssxestore



The screenshot shows the Azure Storage blade. A new storage account named "cssxestore" has been created and is listed under the "Storage" category. The account is currently "Online". The location is set to "South Central US".



The "Manage Access Keys" blade is open for the "cssxestore" account. It displays the "STORAGE ACCOUNT NAME" as "cssxestore" and the "PRIMARY ACCESS KEY" as a long, randomly generated string: "VtpfGJ...". Below the primary key is a "regenerate" button. There is also a "SECONDARY ACCESS KEY" field containing a different long string and a "regenerate" button next to it.

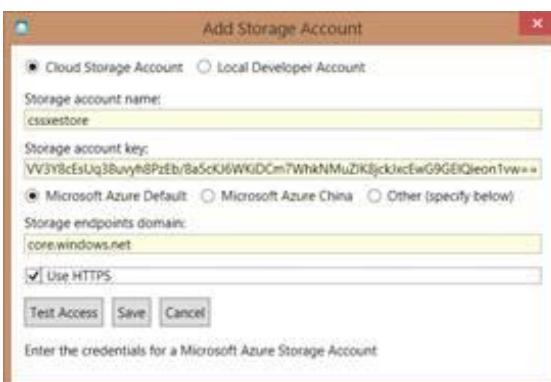
After storage is created, note the "Storage Account Name" and the "Primary Access Key" be needed to access the .xel output files within the storage.

After storage is created, note the "Storage Account Name" and the "Primary Access



Extended Events

- Enable access to storage from SQL Database.
- You shall use a tool called Azure Storage Explorer to download the XEL files from the container to your local machine. Download and install Azure Storage Explorer (<http://azurestorageexplorer.codeplex.com/>) on your client machine.
- We used the current version (Version 6) as of this writing; feel free to use the latest version if another one is available.
 - Here are the steps to do for the same:
 - Configure Azure Storage Explorer to point to your storage account using the "Storage Account Name" and "Storage Account Key" from previous section titled Setup storage account and container. Check the Use HTTPS" checkbox and then click on Test Access



Extended Events

- The “Test Access” button click should return a success message. Click Save to save the configuration



You should see a similar screen to this once you are done



The next step is to add a container within the storage account. Click Blob Containers and add a new container



In my case I created a container named xe_container (with access level=off)

Extended Events

In my case I created a container named xe_container (with access level=off)



that show up like this



Next click on the specific container and then click on Security



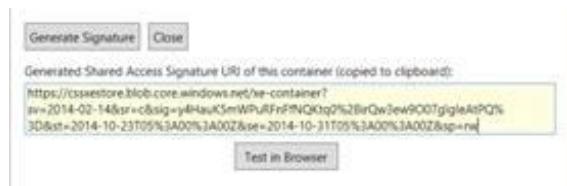
Go to Shared Access Signatures tab and choose Read and Write Permission and ensure that End Date is well in future so the access does not expire before data collection is done.

Extended Events

Go to Shared Access Signatures tab and choose Read and Write Permission and ensure that End Date is well in future so the access does not expire before data collection is done.



Select the "Generate Signature" button, then copy the output to notepad



The output of this step should be a SAS key for the container that looks like this:

http://your_storage_account.blob.core.windows.net/your_container_name?sr=c&sv=start_date&sr=c&sig=%signature

Extended Events

- Collecting Data using Extended Events

Open the SQL SERVER Management Studio and follow up the following instructions:

- Create a master key specifying a strong password.

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'xxxxxxxx!';
```

- Define the blob storage where the XEL will be save. Use the SAS key that was provided by the definition of the blob storage.

```
CREATE DATABASE SCOPED CREDENTIAL [https://xxxx.blob.core.windows.net/xe-container]
WITH IDENTITY='SHARED ACCESS SIGNATURE',
SECRET = 'sv=2014-02-14&sr=c&sig=Hz2n9vs%3D&st=2016-01-25T23%3A00%3A00Z&se=2016-02-02T23%3A00%3A00Z&sp=rw'
```

- Define the Extended Event. The filter result <> 0 will capture when the error gives an error.

```
CREATE EVENT SESSION ssEventoTimeout
ON DATABASE
ADD EVENT sqlserver.sql_batch_completed (
    ACTION(sqlserver.client_app_name,sqlserver.client_connection_id,sqlserver.client_hostname,sqlserver.client_
pid,sqlserver.database_id,sqlserver.database_name,sqlserver.session_id,sqlserver.sql_text,sqlserver.username)
WHERE ([result] <> (0)))
ADD TARGET package0.asynchronous_file_target(
    SET filename=https://xxx.blob.core.windows.net/xe-container/DemoPersistedEvents.xel)
```

- Start the event and wait to reproduce the issue.

```
ALTER EVENT SESSION [ssEventoTimeout] ON DATABASE STATE = START;
```

- Once the issue has been reproduced, stop the event.

```
ALTER EVENT SESSION [ssEventoTimeout] ON DATABASE STATE = STOP;
```

- You could delete it when you don't need it anymore.

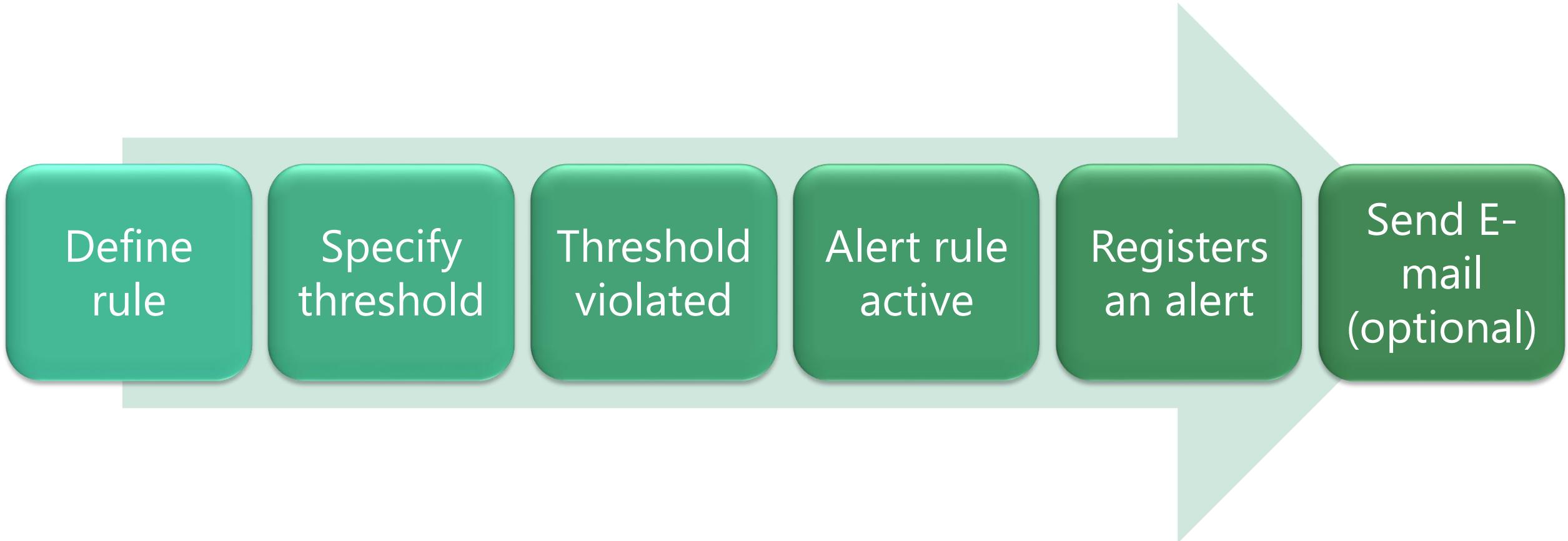
```
DROP EVENT SESSION ssEventoTimeout ON DATABASE
```

- You should see XEL files in the storage container in Azure Storage Explorer

- You can then download to your laptop/local machine

- (Note that the files are in the format server_name_database_name_session_name.– so you can tell which server & database they were collected from)

Alerts and events



DEADLOCK

- Is not showing correctly in the sys.event_log view, so we have a workaround to obtain it.
- Now all information is saving in Extended Events files per instance.

```
SELECT *,CAST(event_data as XML).value('(/event/@timestamp)[1]',  
'datetime2') AS timestamp  
,CAST(event_data as XML).value('(/event/data[@name="error"]/value)[1]',  
'INT') AS error  
,CAST(event_data as XML).value('(/event/data[@name="state"]/value)[1]',  
'INT') AS state  
,CAST(event_data as  
XML).value('(/event/data[@name="is_success"]/value)[1]', 'bit') AS  
is_success  
,CAST(event_data as  
XML).value('(/event/data[@name="database_name"]/value)[1]', 'sysname') AS  
database_name  
FROM sys.fn_xe_telemetry_blob_target_read_file('el', null, null, null)  
where object_name = 'database_xml_deadlock_report'
```

SQL Database Alert Rules

Blocked by
Firewall

Failed
Connections

Successful
Connections

CPU
Percentage

Deadlocks

DTU
Percentage

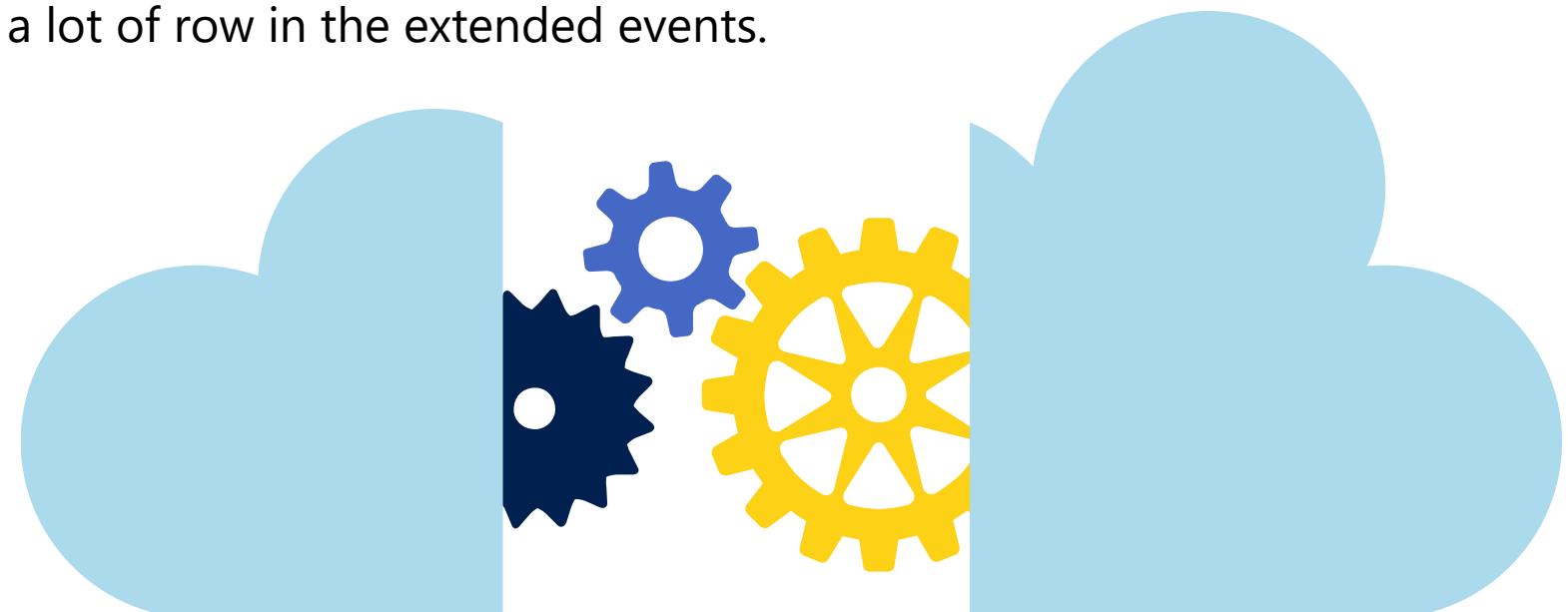
Log IO
Percentage

Data IO
Percentage

Total
Database
Size

Alerts – 3rd Party tool

- New Relic issue causing a high memory usage, must be to review the actual tool that customer has.
- Review with Nagios some timeouts and our customer needs to implement other alternatives.
- Performance issue with you have a lot of row in the extended events.
- Review the unit of measure.



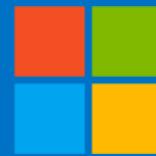
Q&A



If you have any question, please send an email

jmjurado@microsoft.com





Microsoft

Thanks!!