Microsoft

Module 1: Explore core data concepts

Agenda

Explore core data concepts

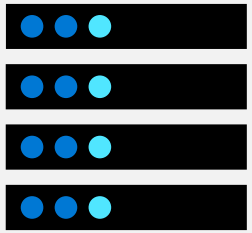
Explore roles and responsibilities in the world of data

Describe concepts of relational data

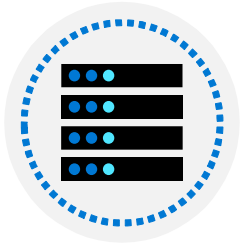
Explore concepts of non-relational data

Explore concepts of data analytics

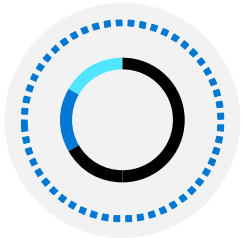
Lesson 1: Explore core data concepts



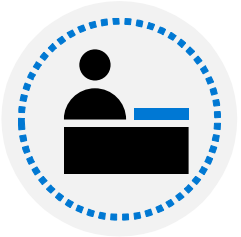
Lesson 1 objectives



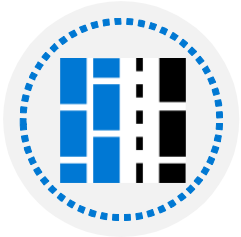
Identify how data is defined and stored



Identify characteristics of relational and non-relational data



Describe and differentiate data workloads

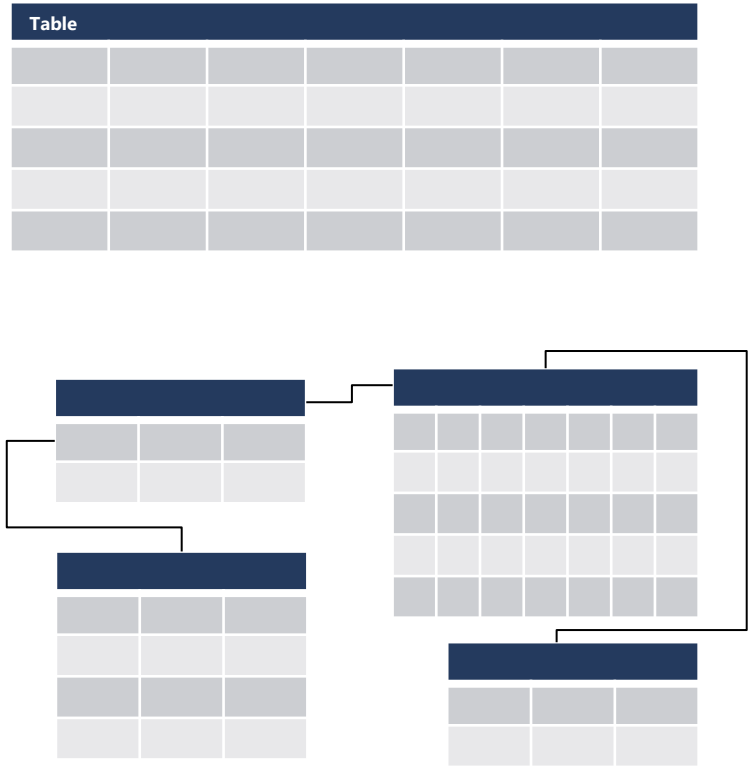


Describe and differentiate batch and streaming data

What is data?

Collection of facts, numbers, descriptions, objects , stored in a structured, semi-structured, unstructured way.

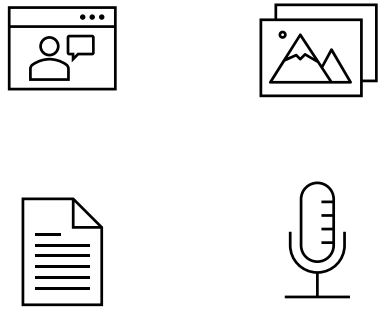
Structured



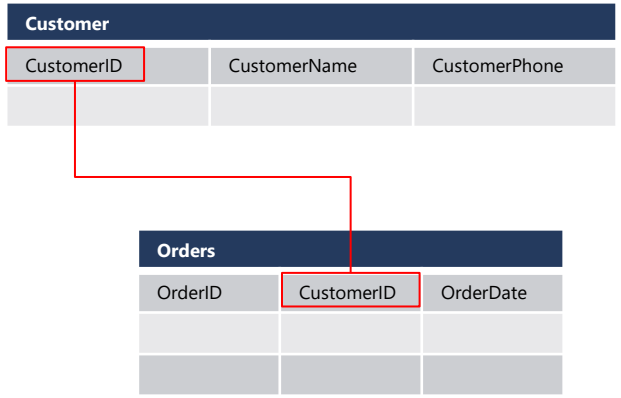
Semi-structured

```
## Document 1 ## {
"customerID": "103248",
"name": { "first": "AAA",
"last": "BBB" }, "address": {
"street": "Main Street",
"number": "101", "city":
"Acity", "state": "NY" },
"ccOnFile": "yes",
"firstOrder": "02/28/2003" }
## Document 2 ## {
"customerID": "103249",
"name": { "title": "Mr",
"forename": "AAA",
"lastname": "BBB" },
"address": { "street":
"Another Street", "number":
"202", "city": "Bcity",
"county": "Gloucestershire",
"country-region": "UK" },
"ccOnFile": "yes" }
```

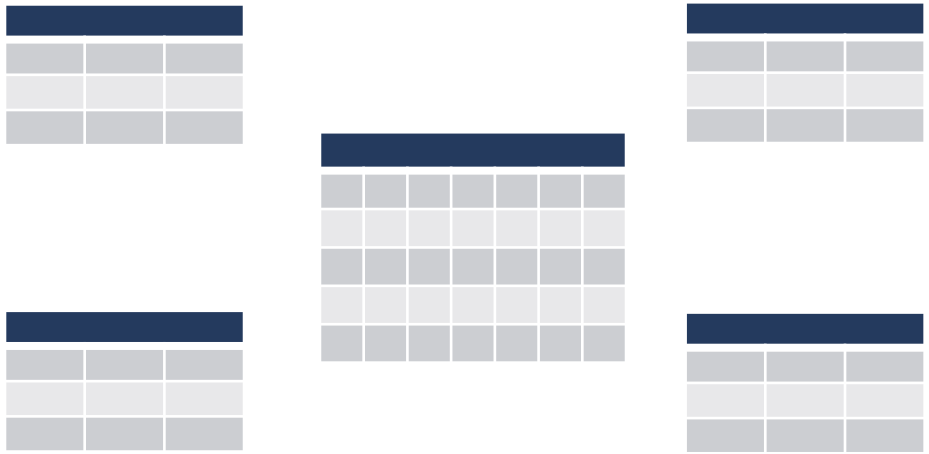
Unstructured



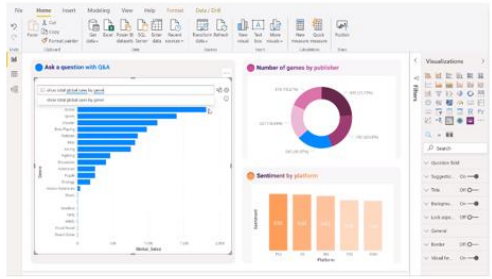
Transactional vs analytical data stores



Online Transactional Processing (OLTP)



Online Analytical Processing (OLAP)



Transactional workloads

Customer		
CustomerID	CustomerName	CustomerPhone

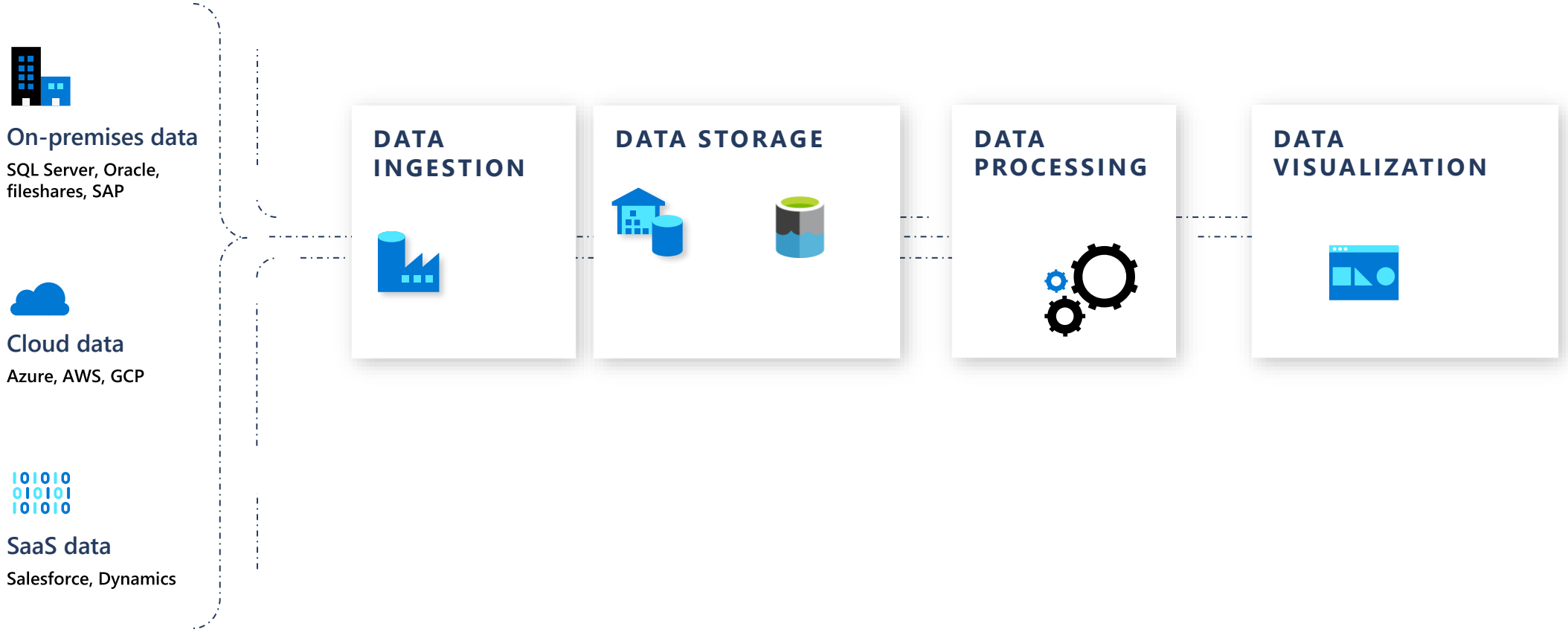
Orders		
OrderID	CustomerID	OrderDate

Account	
CustomerID	Balance
5558	1000
6023	1500

Transfers					
TransactionID	FromAccount	ToAccount	Transaction Amount	OrderDate	TransactionDescription
982801	6023	5558	500	DD/MM/YY	Transfer 500 from account 6023 to account

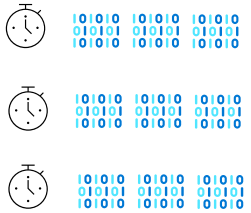
```
BEGIN TRANSACTION
UPDATE Account
SET Balance = Balance -500
WHERE CustomerID=6023;
UPDATE Account
SET Balance = Balance +500
WHERE CustomerID=5558;
INSERT INTO Transfers (Fromaccount, ToAccount, TransactionAmount,TransactionDescription)
 VALUES (6023,5558,500,'Transfer 500 from account 6023 to account 5558)
COMMIT TRANSACTION
```

Analytical System

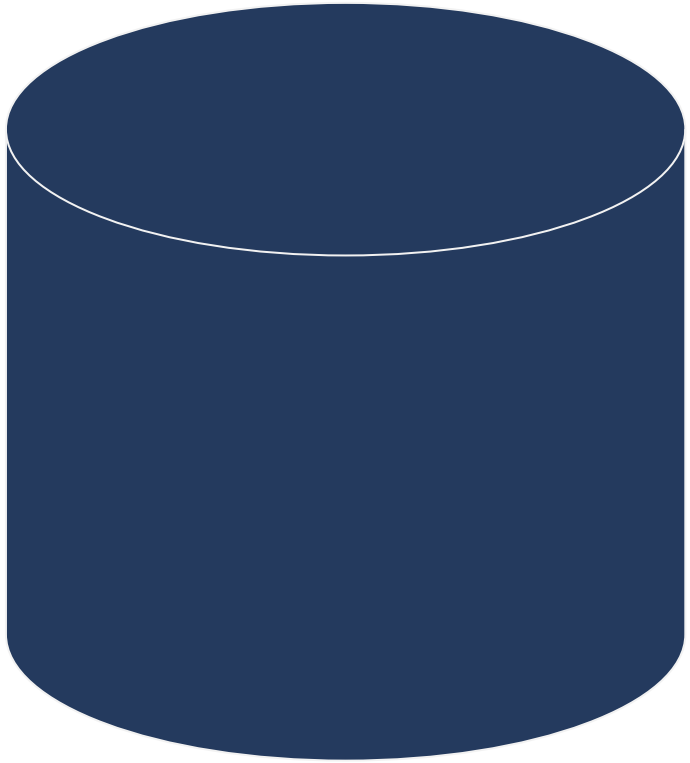


Batch Data / Streaming Data

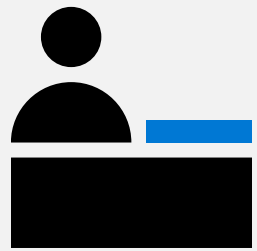
BATCH



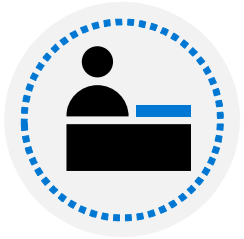
STREAMING



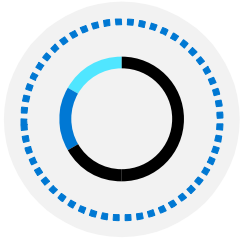
Lesson 2: Explore roles and responsibilities in the world of data



Lesson 2 objectives



Explore data job roles



Explore common tasks and tools for data job roles

Roles in Data

Database Administrator

Database Management

Implements Data Security

Backups

User Access

Monitors performance

Data Engineer

Data Pipelines and processes

Data Ingestion storage

Prepare data for Analytics

Prepare data for analytical processing

Data Analyst

Provides insights into the data

Visual Reporting

Modeling Data for Analysis

Combines data for visualization and analysis

Common Tools – Database Administrator

Azure Data Studio

- Graphical interface for managing on-premises and cloud-based data services
- Runs on Windows, macOS, Linux

SQL Server Management Studio

- Graphical interface for managing on-premises and cloud-based data services
- Runs on Windows
- Comprehensive Database Administration tool

Azure Portal / CLI

- Tools for management and provisioning of Azure Data Services
- Manual and automation of scripts using Azure Resource Manager or Command Line Interface scripting

Common Tools – Data Engineering

Azure Synapse Studio

- Azure Portal integrated to manage Azure Synapse
- Data Ingestion (Azure Data Factory)
- Management of Azure Synapse assets (SQL Pools / Spark Pool)

SQL Server Management Studio

- Graphical interface for managing on-premises and cloud-based data services
- Runs on Windows
- Comprehensive Database Administration tool

Azure Portal / CLI

- Tools for management and provisioning of Azure resources
- Manual and automation of scripts using Azure Resource Manager or Command Line Interface scripting

Common Tools – Data Analyst

Power BI Desktop

- Data Visualization tool
- Model and Visualize Data
- Management of Azure Synapse assets (SQL Pools / Spark Pool)

Power BI Portal / Power BI Service

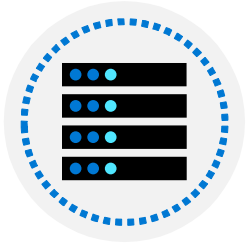
- Authoring and management of Power BI reports
- Authoring of Power BI dashboards
- Share Reports / Datasets

Power BI Report Builder

- Data Visualization tool for paginated reports
- Model and Visualize paginated reports

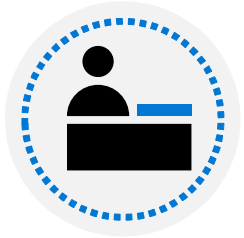
Lesson 3: Describe concepts of relational data

Lesson 3 objectives



Explore the characteristics of relational data

Define tables, indexes, and views



Explore relational data workload offerings in Azure

Tables

Customers		
CustomerID	CustomerName	CustomerPhone
100	Muisto Linna	*XXX-XXX-XXXX*
101	Noam Maoz	*XXX-XXX-XXXX*
102	Vanja Matkovic	*XXX-XXX-XXXX*
103	Qamar Mounir	*XXX-XXX-XXXX*
104	Zhenis Omar	*XXX-XXX-XXXX*
105	Claude Paulet	*XXX-XXX-XXXX*
106	Alex Pettersen	*XXX-XXX-XXXX*
107	Francis Ribeiro	*XXX-XXX-XXXX*

Data is stored in a table

Table consists of rows and columns

All rows have same # of columns

Each column is defined by a datatype

Normalization

Customers		
CustomerID	CustomerName	CustomerPhone
100	Muisto Linna	*XXX-XXX-XXXX*
101	Noam Maoz	*XXX-XXX-XXXX*
102	Vanja Matkovic	*XXX-XXX-XXXX*
103	Qamar Mounir	*XXX-XXX-XXXX*
104	Zhenis Omar	*XXX-XXX-XXXX*
105	Claude Paulet	*XXX-XXX-XXXX*
106	Alex Pettersen	*XXX-XXX-XXXX*

Orders		
OrderID	CustomerName	CustomerPhone
AD100	Noam Maoz	*XXX-XXX-XXXX*
AD101	Noam Maoz	*XXX-XXX-XXXX*
AD102	Noam Maoz	*XXX-XXX-XXXX*
AX103	Qamar Mounir	*XXX-XXX-XXXX*
AS104	Qamar Mounir	*XXX-XXX-XXXX*
AR105	Claude Paulet	*XXX-XXX-XXXX*
MK106	Muisto Linna	*XXX-XXX-XXXX*

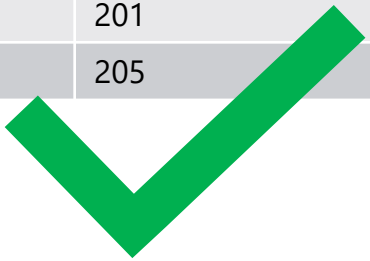
Data is normalized to:

- Reduce storage
- Avoid data duplication
- Improve data quality

Relations

Customers		
CustomerID	CustomerName	CustomerPhone
100	Muisto Linna	*XXX-XXX-XXXX*
101	Noam Maoz	*XXX-XXX-XXXX*
102	Vanja Matkovic	*XXX-XXX-XXXX*
103	Qamar Mounir	*XXX-XXX-XXXX*
104	Zhenis Omar	*XXX-XXX-XXXX*
105	Claude Paulet	*XXX-XXX-XXXX*
106	Alex Pettersen	*XXX-XXX-XXXX*

Orders		
OrderID	CustomerID	SalesPersonID
AD100	101	*200*
AD101	101	200
AD102	101	200
AX103	103	201
AS104	103	201
AR105	105	200
MK106	105	201
DB205	100	205



In a normalized database schema:

- Primary Keys and Foreign keys are used to define relationships
- No data duplication exists (other than key values in 3rd Normal Form (3NF)
- Data is retrieved by joining tables together in a query

Indexes

Customers		
CustomerID	CustomerName	CustomerPhone
100	Muisto Linna	*XXX-XXX-XXXX*
101	Noam Maoz	*XXX-XXX-XXXX*
102	Vanja Matkovic	*XXX-XXX-XXXX*
103	Qamar Mounir	*XXX-XXX-XXXX*
104	Zhenis Omar	*XXX-XXX-XXXX*
105	Claude Paulet	*XXX-XXX-XXXX*
106	Alex Pettersen	*XXX-XXX-XXXX*

IDX-CustomerRegion	
CustomerID	Region
100	France
101	Brazil
102	Croatia
103	Jordan
104	Spain
105	France
106	USA

An index

- Optimizes search queries for faster data retrieval
- Reduces the amount of data pages that need to be read to retrieve the data in a SQL Statement
- Data is retrieved by joining tables together in a query

View

Customers		
CustomerID	CustomerName	CustomerPhone
100	Muisto Linna	*XXX-XXX-XXXX*
101	Noam Maoz	*XXX-XXX-XXXX*
102	Vanja Matkovic	*XXX-XXX-XXXX*
103	Qamar Mounir	*XXX-XXX-XXXX*
104	Zhenis Omar	*XXX-XXX-XXXX*
105	Claude Paulet	*XXX-XXX-XXXX*
106	Alex Pettersen	*XXX-XXX-XXXX*

Orders		
OrderID	CustomerID	SalesPersonID
AD100	101	*200*
AD101	101	200
AD102	101	200
AX103	103	201
AS104	103	201
AR105		
MK106		
DB205		

A view is a virtual table based on the result set of query

- Views are created to simplify the query
- Combine relational data into a single pane view

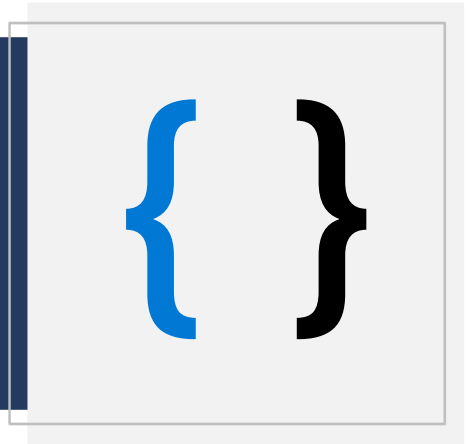
Create the definition of a view:

```
CREATE VIEW vw_customerorders AS
SELECT Customers.CustomerID,
Customers.CustomerName, Orders.OrderID FROM
Customers JOIN Orders on Customers.CustomerID
= Orders.CustomerID
```

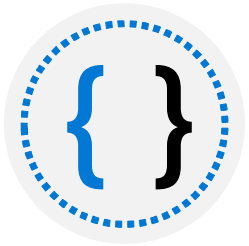
Retrieve the orders placed by customer 102 using the view:

```
SELECT CustomerName, OrderID from
vw_customerorders WHERE CustomerID=102
```

Lesson 4: Explore concepts of non-relational data

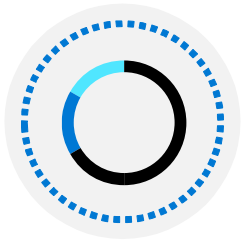


Lesson 4 objectives



Explore the characteristics of non-relational data

Define types of non-relational data



Describe NoSQL, and the types of non-relational databases

Explore characteristics of non-relational data

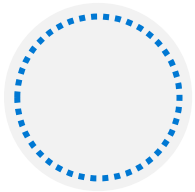
Entities

```
## Customer 1 ID: 1
Name: Mark Hanson
Telephone: [ Home: 1-999-9999999, Business: 1-888-8888888, Cell: 1-777- 7777777 ]
Address: [ Home: 121 Main Street, Some City, NY, 10110,
           Business: 87 Big Building, Some City, NY, 10111 ]
## Customer 2 ID: 2
Title: Mr
Name: Jeff Hay
Telephone: [ Home: 0044-1999-333333, Mobile: 0044-17545-444444 ]
Address: [ UK: 86 High Street, Some Town, A County, GL8888, UK,
           US: 777 7th Street, Another City, CA, 90111 ]
```

Non-relational collections can have

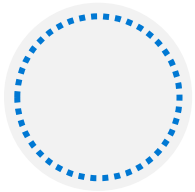
- Multiple entities in the same collection or container with different fields
- Have a different, non-tabular schema
- are often defined by labeling each field with the name it represents

Identify non-relational database use cases



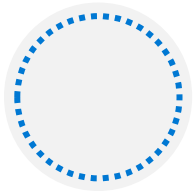
IoT and Telematics

Often require to ingest large amounts of data in frequent burst of activity, data is either semi structured or structured, often requires real time processing



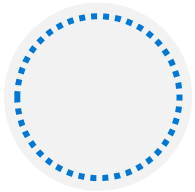
Retail and Marketing

Common scenarios for globally distributed data, document storage



Gaming

In-game stats, social media integration, leaderboards, low-latency applications



Web and Mobile

Common used with web click analytics, modern applications including bots

Types of non-relational data

What is semi-structured data?

Data structure is defined within the actual data by fields.
Format / file types include:

JSON

AVRO

ORC

Parquet

What is unstructured data?

- Does not naturally contain fields
  Examples: video, audio, media streams, documents

- Often used to extract data form and categorize or identify “structures”
- Frequently used in combination with Machine Learning or Cognitive Services capabilities to “extract data” by using:
  - Text Analytics
  - Sentiment Analysis with Cognitive APIs
  - Vision API

What is NoSQL?

Loose term, to describe non-relational

- Key-value stores
- Document based
- Column family databases
- Graph Databases

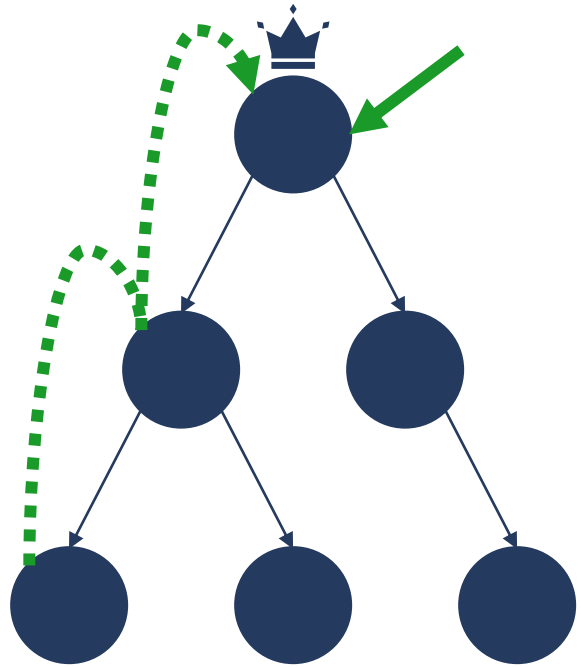
What is a graph database?

- Stores entities centric around relationships
- Enable applications to perform queries traversing a network of nodes and edges

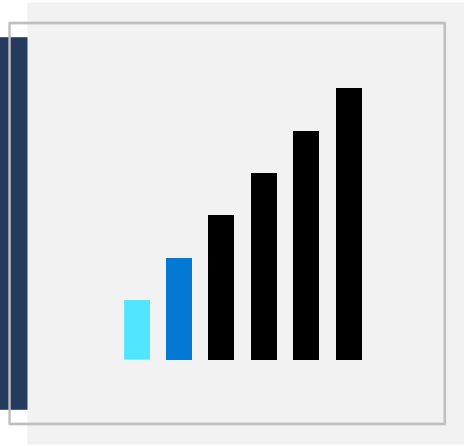
What applications require a graph database?

Business requirements:

- OLTP apps with highly correlated data.
- Easy updates to single or many objects.
- Flexible data modelling.
- Data requirements that evolve.
- Hierarchical data structures.



Lesson 5: Explore concepts of data analytics



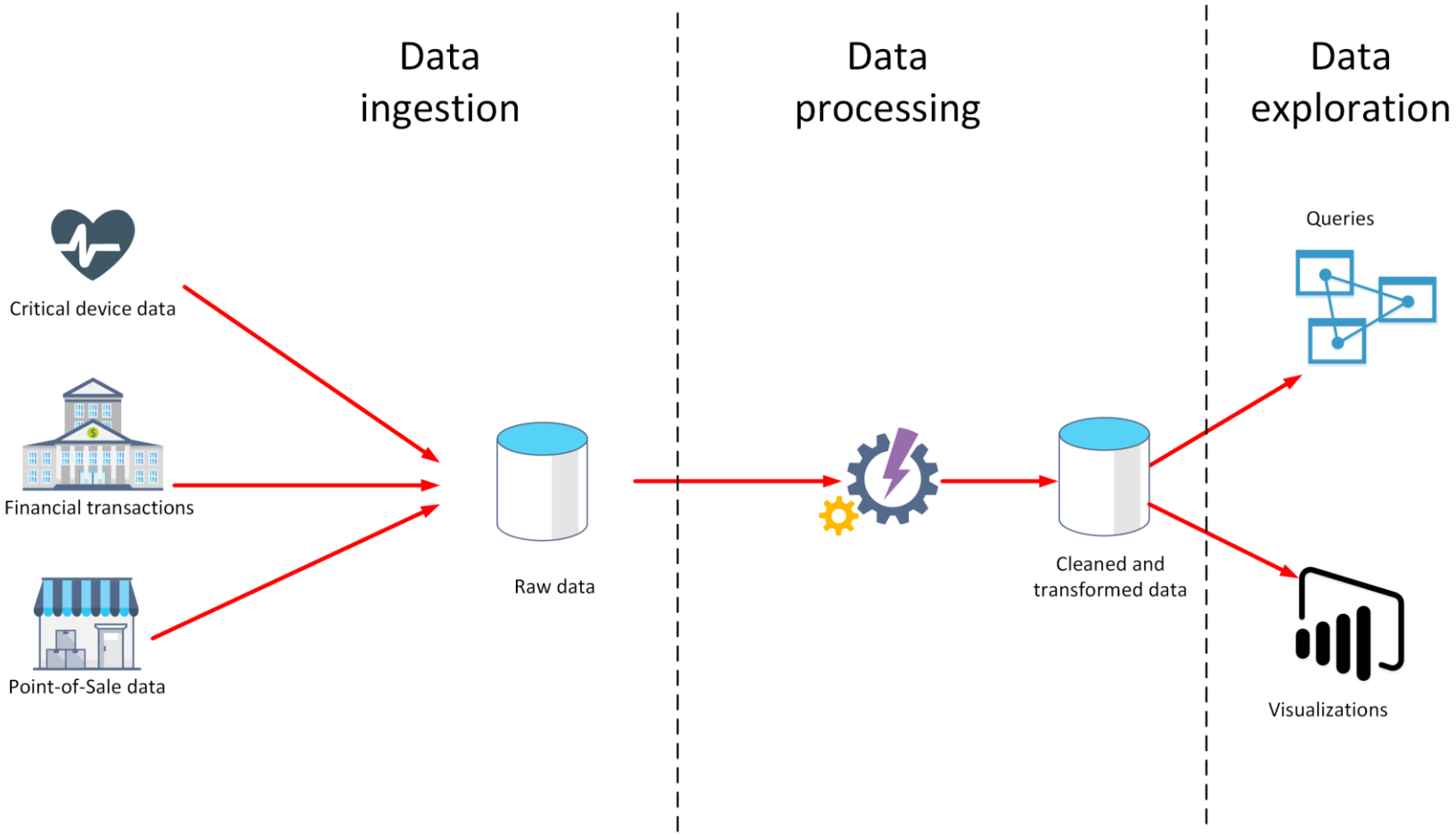
Lesson 5 objectives

Learn about data ingestion and processing

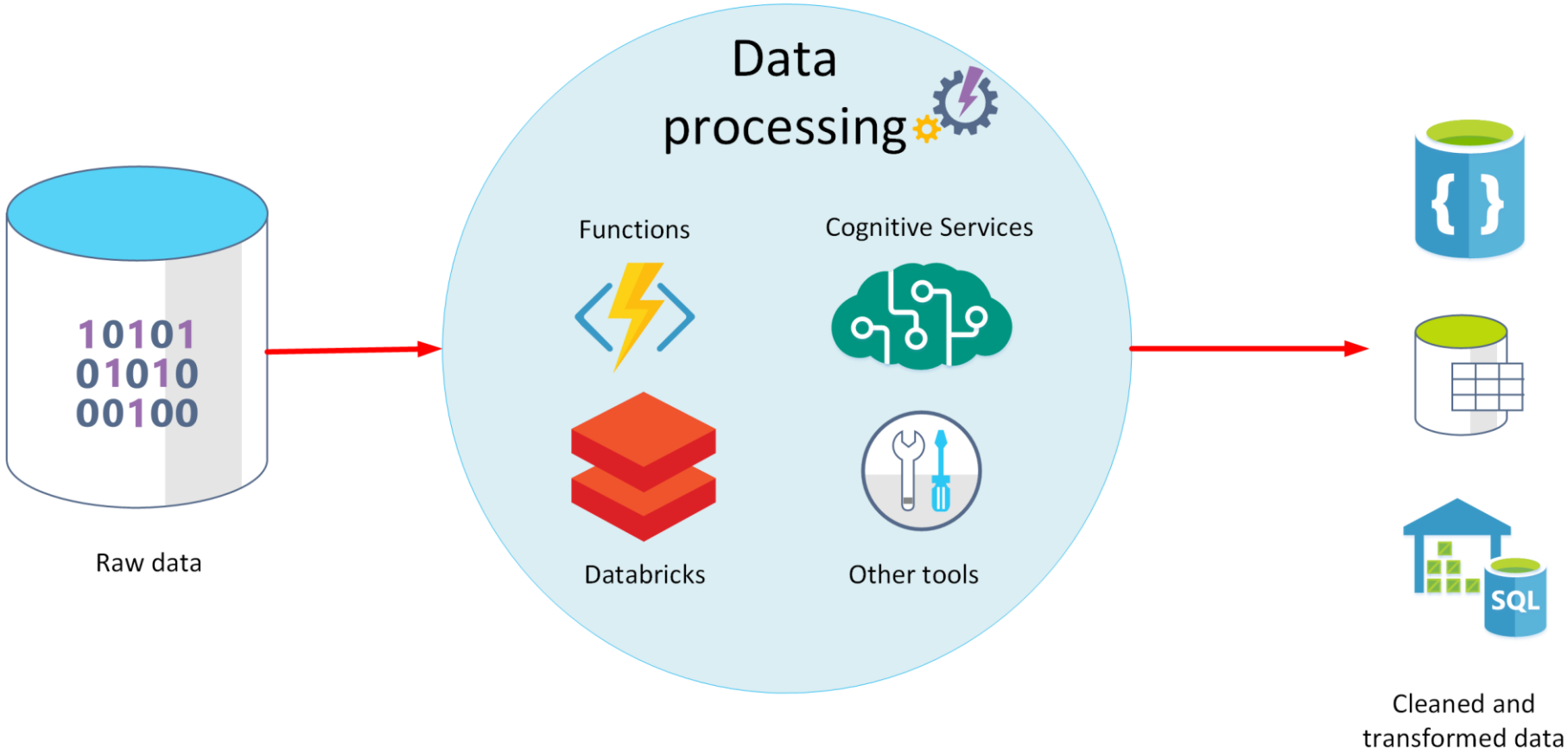
Explore data visualization

Explore data analytics

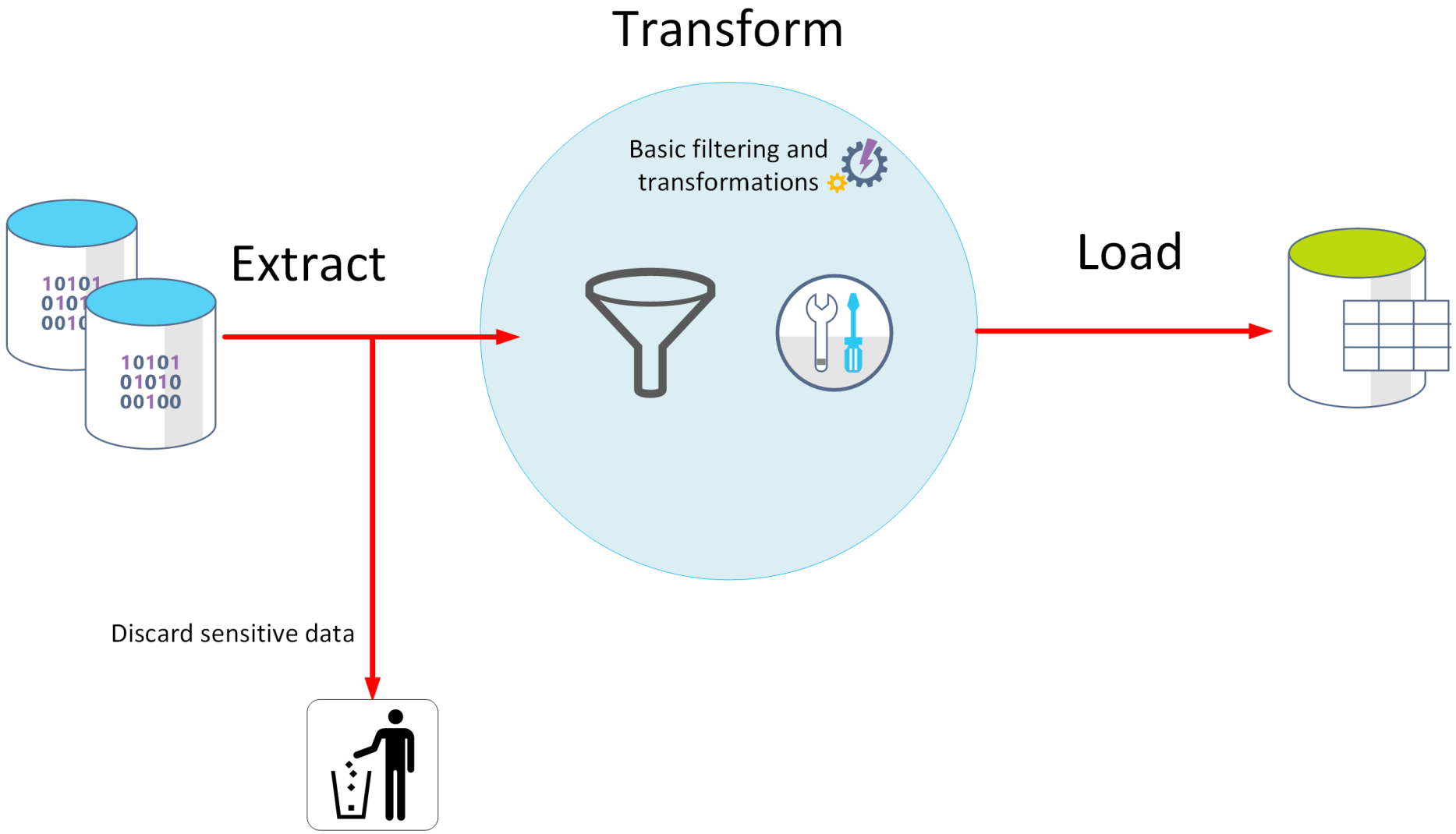
What is data ingestion?



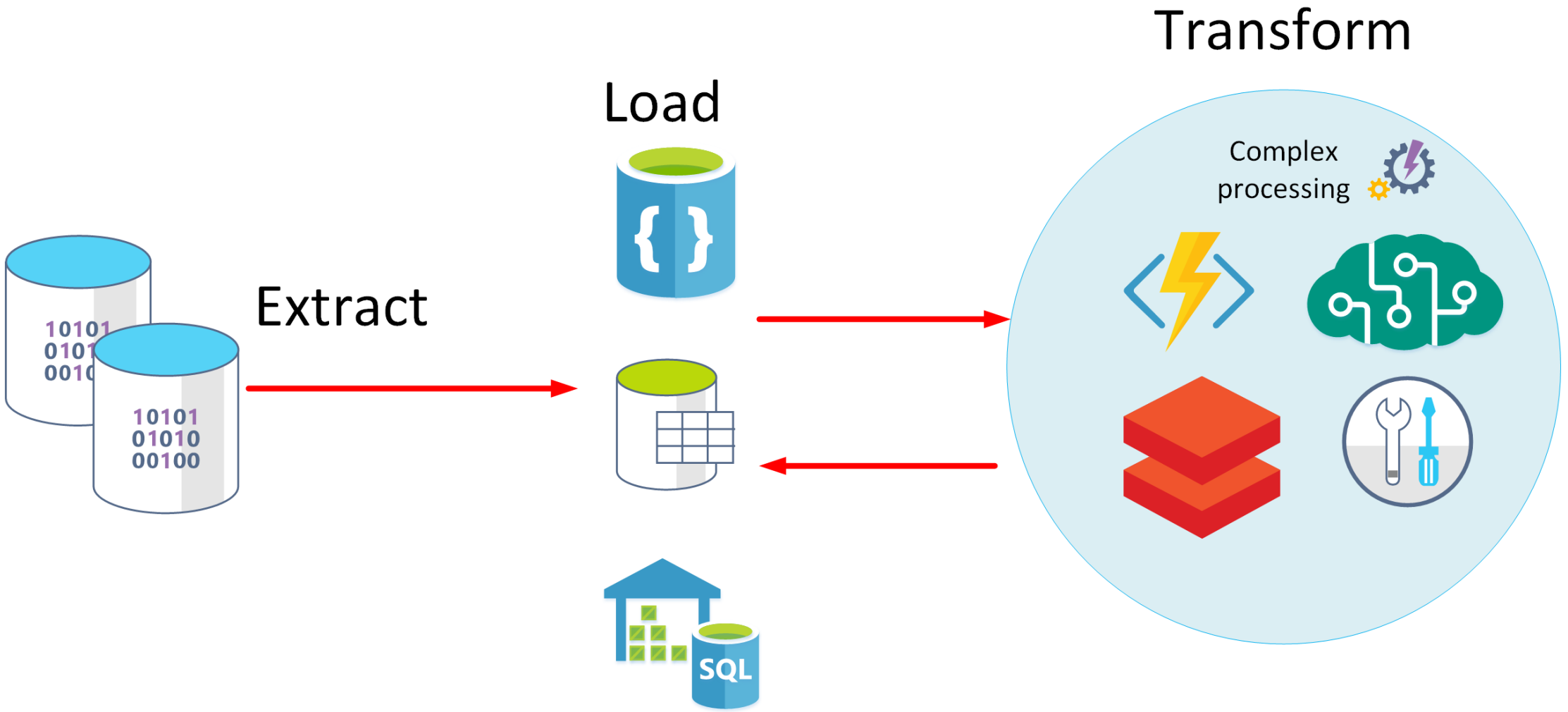
What is data processing?



What is ETL?


Transform
Basic filtering and transformations
Extract
Load
10101
01010
00100
Discard sensitive data

What is ELT?


Transform
Load
Extract
Complex processing
10101
01010
00100
SQL

Explore Data Visualization

Power BI: A collection of software, services, apps, and connectors.

Explore Data Analytics

Microsoft Azure

Microsoft

Module 2: Explore relational data in Azure- Segment 1

Agenda

Explore relational data offerings in Azure

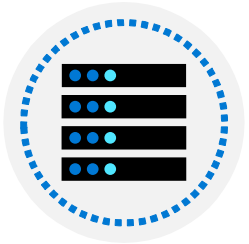
Explore provisioning and deploying relational database offerings in Azure

Query relational data in Azure

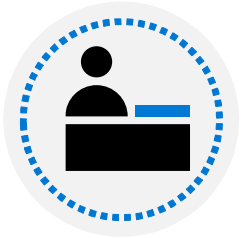
Lesson 1: Explore relational data offerings in Azure

Lesson 1 objectives

Explore relational data offerings in Azure



Explore provisioning and deploying of relational database offerings in Azure



Query relational data in Azure

What are Azure Data Services?

SQL Server on Azure Virtual Machines

Best for re-hosting and apps requiring OS-level access and control

Automated manageability features and OS-level access

Azure SQL Managed Instance

Best for modernizing existing apps

Offers high compatibility with SQL Server and native VNET support

Azure SQL Database

Best for building new apps in the cloud

Pre-provisioned or serverless compute and Hyperscale storage to meet demanding workload requirements

Infrastructure as a Service

Platform as a Service

SQL Server on Azure virtual machines

Customer challenge

I want to migrate to the cloud as fast as possible but maintain operating system control and complete SQL Server functionality

Solution

Get the combined performance, security, and analytics of SQL Server, backed by the flexibility, security, and hybrid connectivity of Azure

Key features

SQL Server and OS server access

Expansive SQL and OS versions

Windows, Linux, Containers

File stream, DTC, and Simple Recovery model

SSAS, SSRS, and SSIS

Azure differentiators

Free Extended Security Updates for SQL Server 2008/R2

Automated Backups and Security Updates

Point in Time Restore with Azure Backup

Accelerated storage performance with Azure Blob Caching

435 percent overall return on an Azure IaaS investment over five years[1]

SQL Server on Azure VM Deployment choices

Deployment Choices

- Marketplace pre-installed SQL Server on Windows or Linux
- Install your own SQL Server
- Lift and Shift with Azure Migrate (Azure Site Recovery)

Resource Provider

- Unlock Licensing and Edition Flexibility
- Automated Backups and Security Updates
- Manage VMs through Azure SQL in portal

Sizes and Storage Performance

- Memory or Storage optimized sizes for best performance
- Data and log on Premium Storage Managed Disks
- Azure Blob Read Caching for data disks
- Tempdb on local SSD
- Ultra disks for extremely low latency needs

Networking and Security

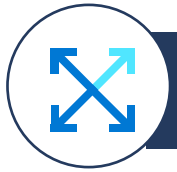
- Virtual Networks to integrate with on-premises
- Advanced Data Security services (Preview)

HADR

- Azure VM built-in HA
- Azure Storage built-in DR
- Azure Backup and Automated backups to Azure Blob Storage
- File-Snapshot Backups
- Failover Cluster Instance with Azure Premium File Share
- Always On Availability Groups with Cloud Witness
- Hybrid Availability Group Secondary replicas
- HADR on RedHat Linux with Pacemaker and fencing

IaaS vs PaaS

Business continuity



High availability

Automated backups

Long term backup retention

Geo-replication

Scale

Advanced security

Version-less

Built-in monitoring

Built-in intelligence

Azure SQL DB

Customer challenge

I want to build modern apps, potentially multi-tenanted, with the highest uptime and predictable performance

Solution

Azure SQL Database is a highly scalable cloud database service with built-in high availability and machine learning

Key features

Single database or elastic pool

Hyperscale storage (100TB+)

Serverless compute

Fully managed service

Private link support

High availability with AZ isolation

Azure differentiators

Industry highest availability SLA of 99.995%

Industry only business continuity SLA with 5 second RPO and 30 second RTO

Price-performance leader for mission-critical workloads while costing up to 86 percent less than AWS RDS (GigaOm)

Azure SQL DB Service Tiers

General purpose	Business critical	Hyperscale*
Most business workloads	Workloads that require low latency, fast recovery, and a readable secondary	Most business workloads with highly scalable storage and read-scale requirements
Remote storage IOPS $ Serverless* P	Local storage IOPS++ $$$ In-memory P S S R	Local + remote storage IOPS+ $$ Unlimited storage P R R R R

*Not in managed instance

Azure SQL DB Managed Instance

Customer challenge

I want to migrate to the cloud, remove management overhead, but I need instance-scoped features (Service Broker, SQL Server Agent, CLR…)

Solution

Managed instance combines leading security features with SQL Server compatibility and business model designed for on-premises customers

Key features

Single instance or instance pool

SQL Server surface area (vast majority)

Native virtual network support

Fully managed service

On-premise identities enabled with Azure AD and AD Connect

Azure differentiators

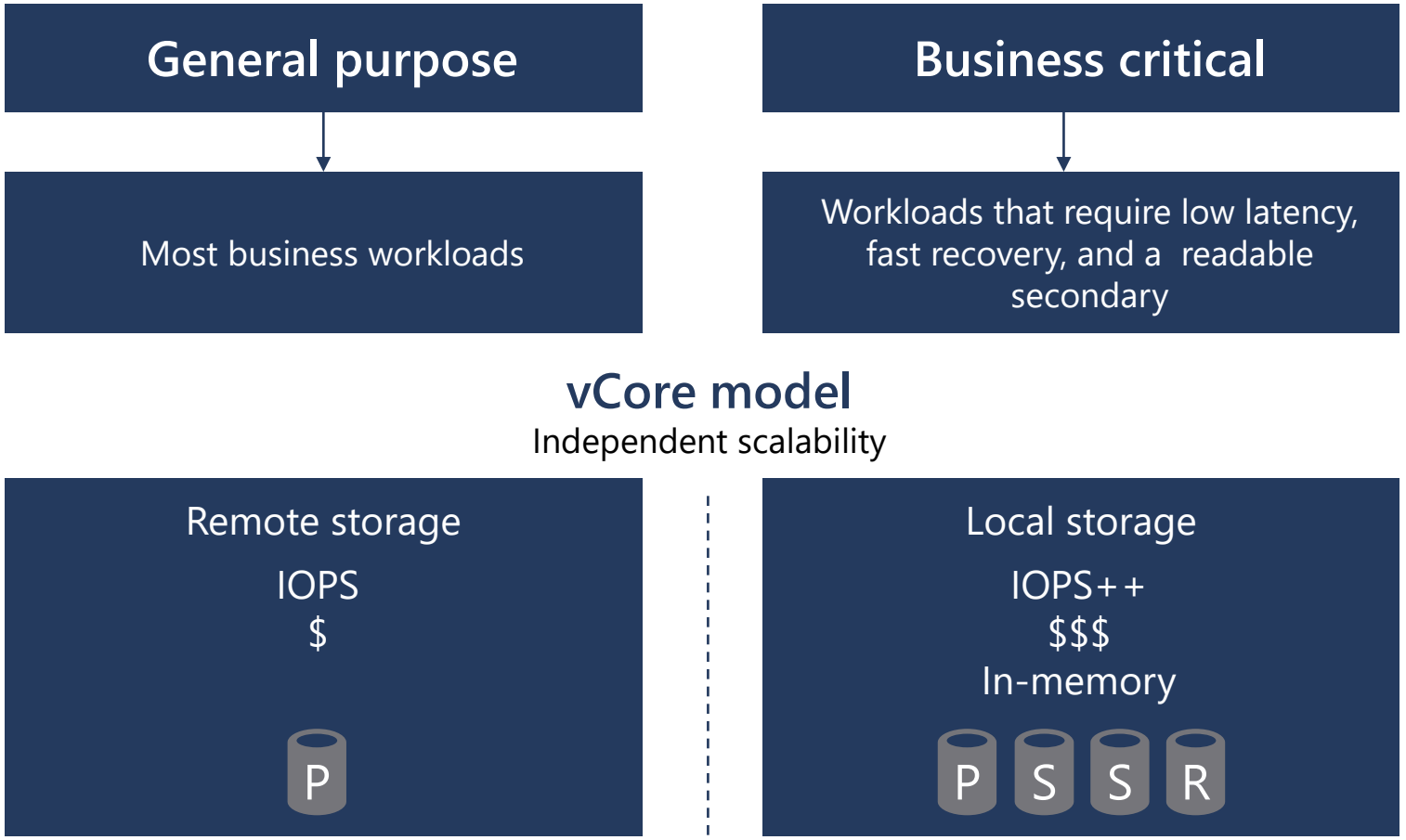
Near zero downtime migration using log shipping

Fully managed business continuity with failover groups

Projected return on investment of 212 percent over three years[1]

The best of SQL Server with the benefits of a managed service

Managed Instance Service Tiers



Azure SQL Managed instance or DB

Azure SQL managed instance

Single instance

SQL Server surface area (vast majority)

Native virtual network support

Fully managed service

Instance pool

Pre-provision compute resources for migration

Enables cost-efficient migration.

Ability to host smaller instances (2Vcore)

Currently in public preview

Azure SQL Database

Single database

Hyperscale storage (up to 100TB)

Serverless compute

Fully managed service

Elastic pool

Resource sharing between multiple databases to price optimize

Simplified performance management for multiple databases

Fully managed service

PostgreSQL, MariaDB, MySQL

PostgreSQL is the most popular and wanted database for modern apps

MySQL is a leading open source relational database for LAMP stack apps

MariaDB is a community-developed fork of MySQL with strong focus on the user community

Benefits of Azure Database for MySQL, PostgreSQL, MariaDB

Fully managed community database

Take advantage of a fully managed service while still using the tools and languages you're familiar with

Built-in high availability for lowest TCO

Ensure your data is always available without the need for additional costs

Intelligent performance and scale

Improve performance with built-in intelligence and up to 16TB storage and 20K IOPs

Industry-leading security and compliance

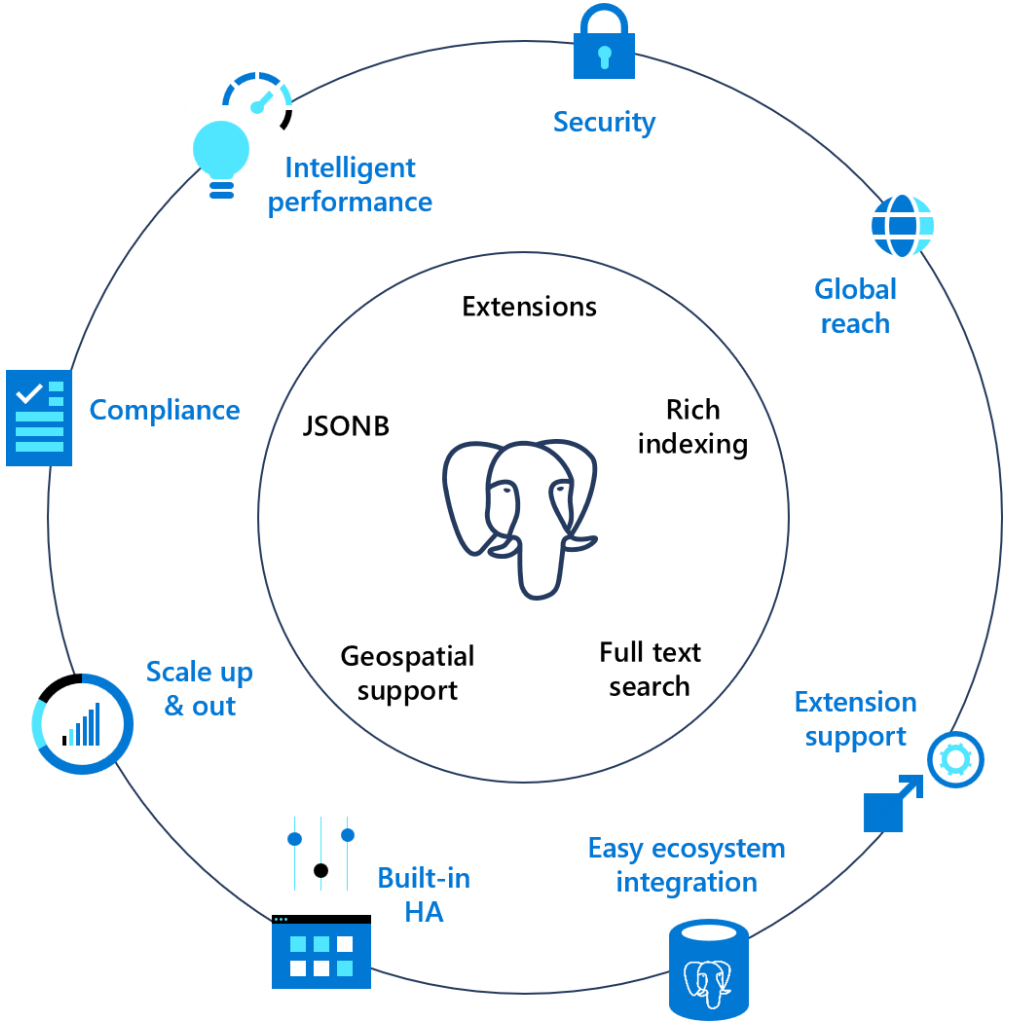
Protect your data with enhanced security features including Advanced Threat Protection

Integration with the Azure ecosystem

Build apps faster with Azure services and safeguard your innovation with Azure IP Advantage

Azure Database for PostgreSQL

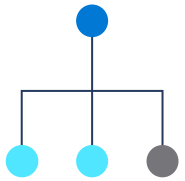
Azure builds upon the core benefits of PostgreSQL and Open Source



Azure Database for PostgreSQL is fully-managed, community PostgreSQL

The benefits of Azure Database for PostgreSQL

Build or migrate your workloads with confidence and optimized for value

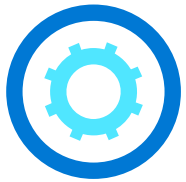


Fully managed and secure

Focus on your apps while Azure manages resource-intensive tasks, supports a large variety of Postgres versions and provides best-in industry indemnification coverage

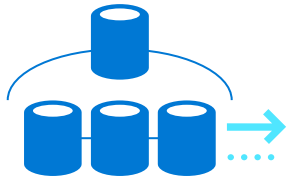
Intelligent performance optimization

Improve performance and reduce cost with customized recommendations



Flexible and open

Stay productive with your favorite Postgres extensions and leverage Microsoft's contributions to the Postgres community



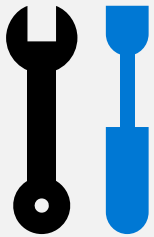
High performance scale-out with Hyperscale

Break free from the limits of single-node Postgres and scale out across 100s of nodes

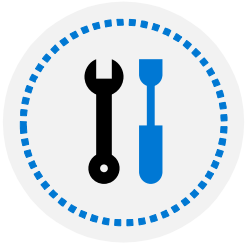
Single Server

Hyperscale

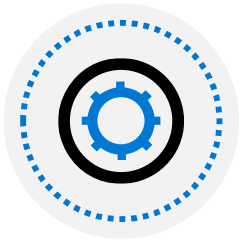
Lesson 2: Explore provisioning and deploying relational database offerings in Azure



Lesson 2 objectives



Provision relational data services

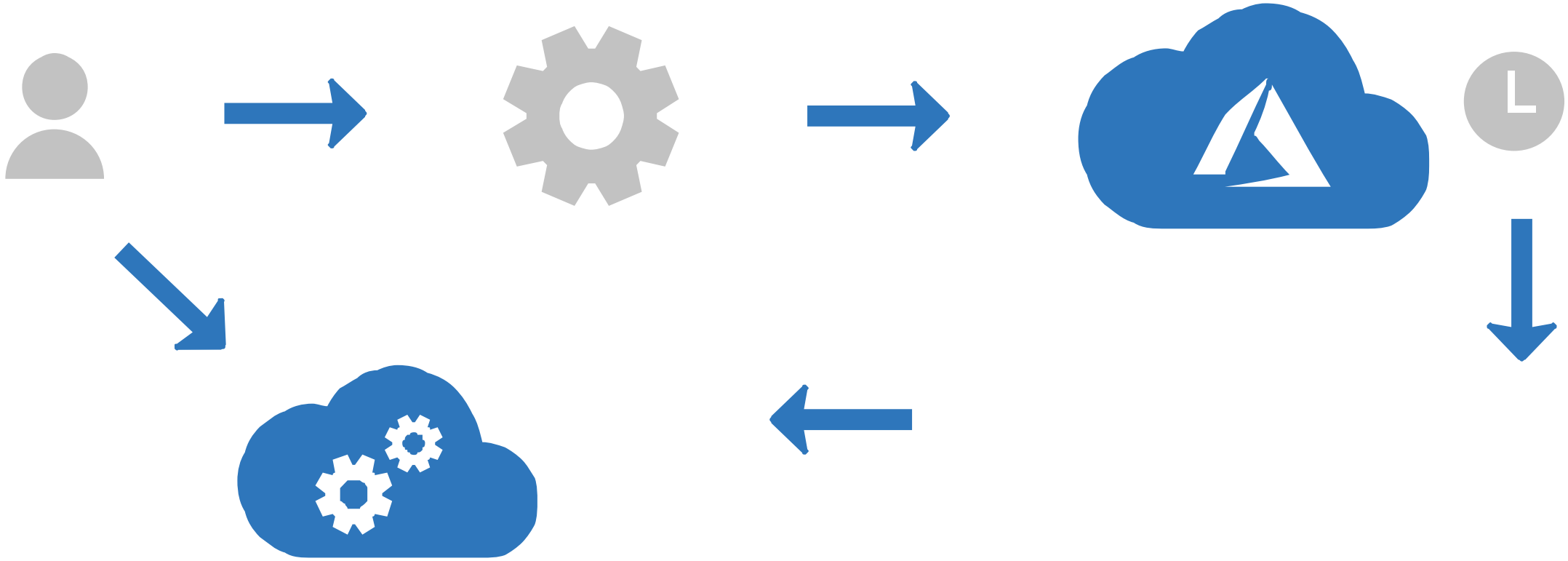


Configure relational data services

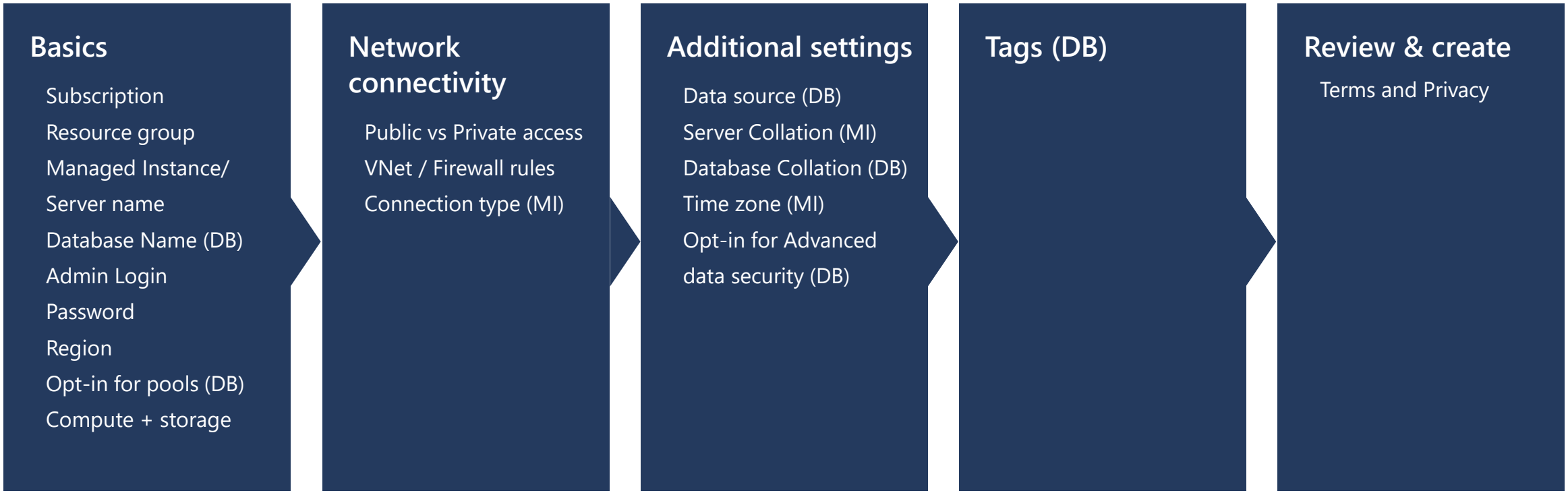
Explore basic connectivity issues

Explore data security

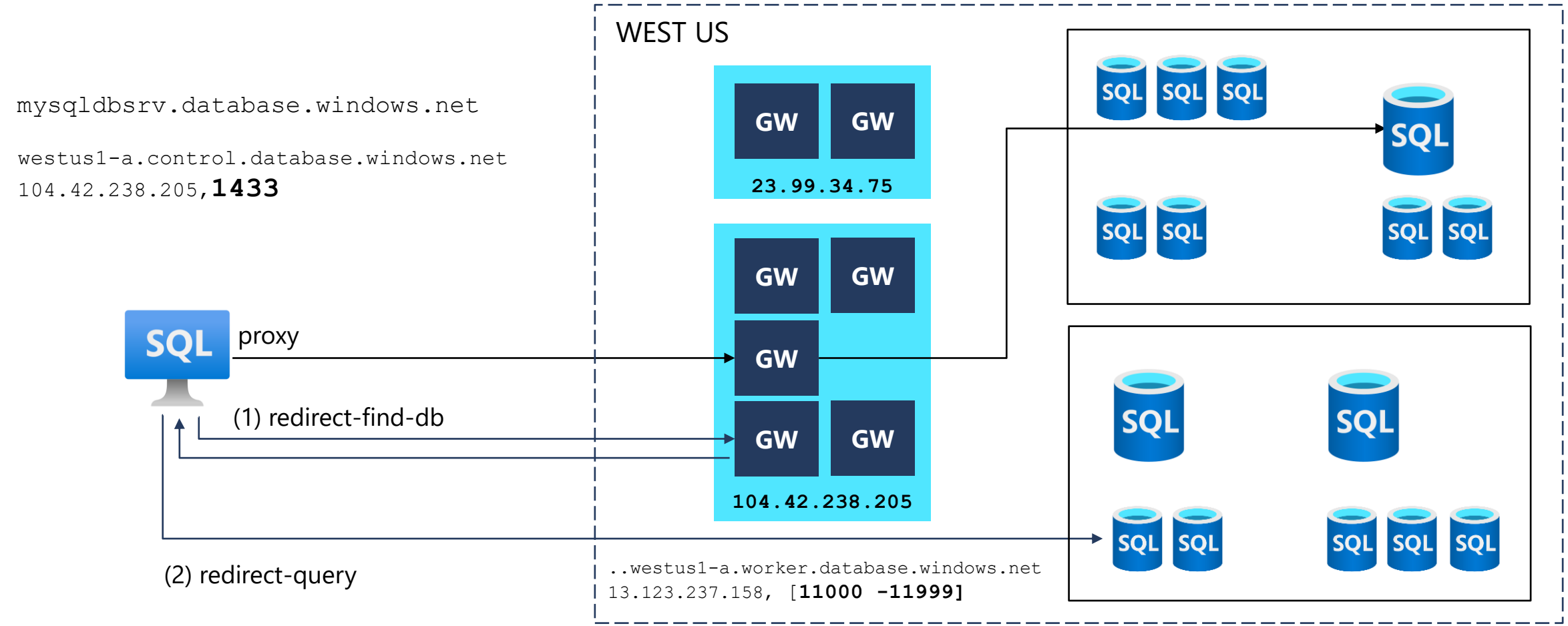
What is provisioning?



Configure Relational Data Services

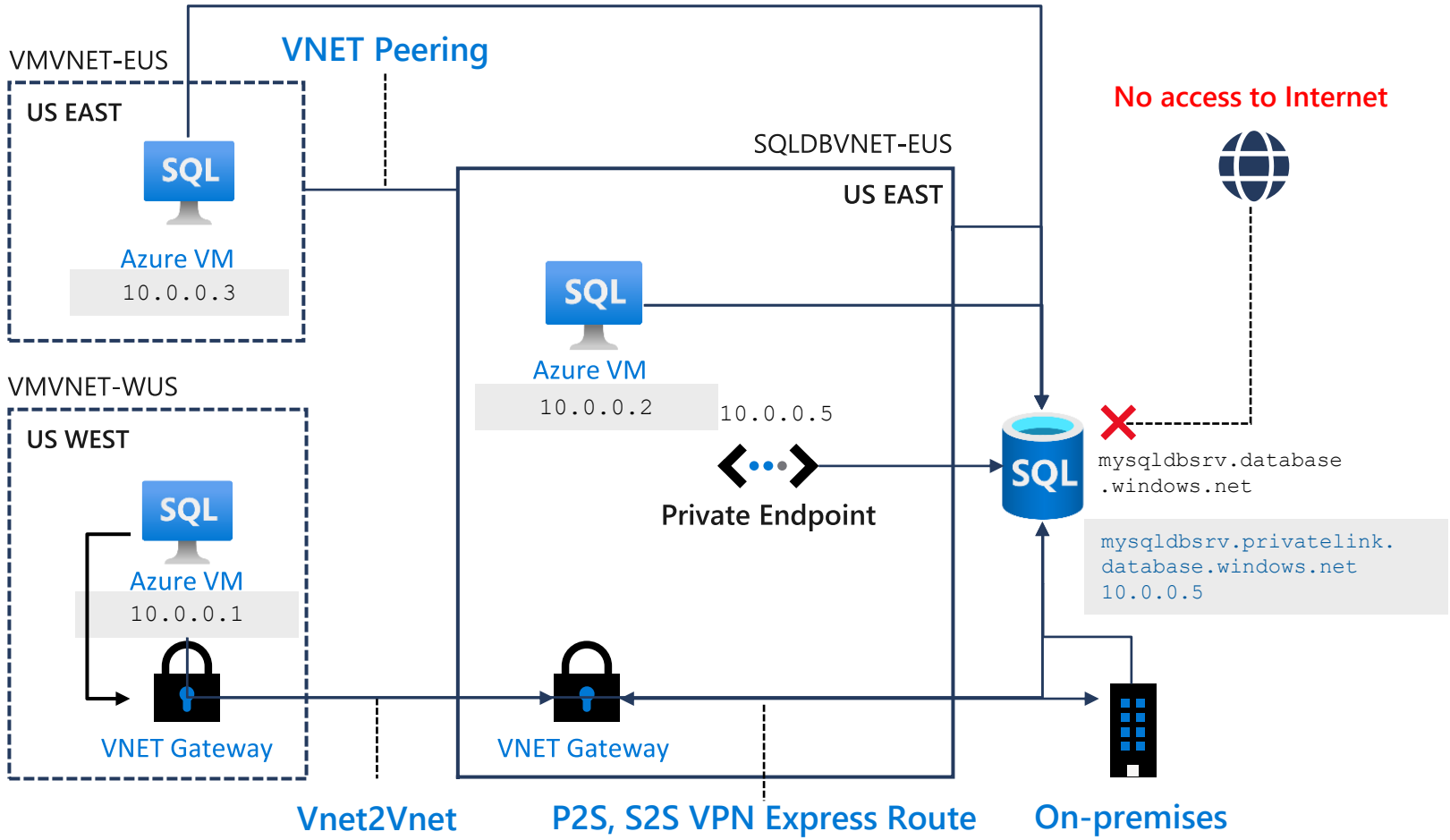


Connectivity and Firewalls



Network Security – SQL Database

- Allow access to Azure services
- Firewall Rules
- Virtual Network Rules
- Private Link



Authentication and Access Control

"Mixed Mode" authentication **forced**

SQL Auth for deployment: **server admin**

- Server-level principal for logical server for DB
- Member of sysadmin server role for MI

Need Windows Auth? Use Azure AD Authentication

Azure Managed Instance

- Azure AD Server Admin
- SQL or Azure AD Logins
- Database Users
- SQL Server Contained Database supported

Azure SQL Database

- Azure AD Server Admin
- SQL logins
- loginmanager and dbmanager roles for limited server admins
- Database Users
- Contained Database Users including Azure AD (recommended)

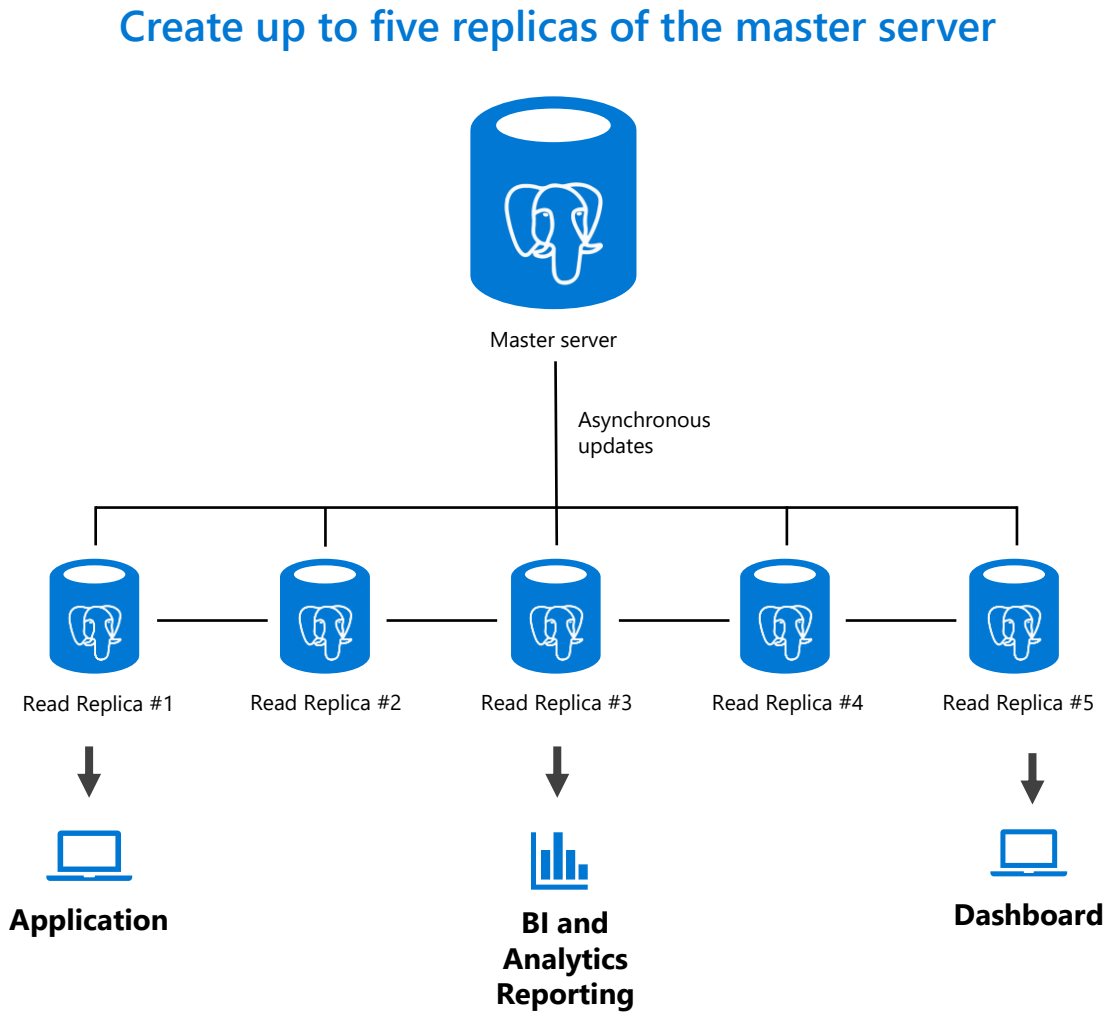
Azure Role Based Access Control (RBAC)

- All Azure operations for Azure SQL are controlled through RBAC
- Think of this as security rights outside the Managed Instance or Database
- Security principal and role-based system
- Scope includes subscription, resource group, and resource
- Decoupled from SQL Security (today)
- Applies to operations in Azure portal and CLI
- Allows for separation of duties for deployment, management, and usage
- Azure locks help protect resources from delete or read-only
- Built-in Azure SQL roles available to reduce need for owner

SQL DB Contributor	SQL Managed Instance Contributor	SQL Security Manager	SQL Server Contributor

Azure DB - read replicas

Read replicas help improve performance and scale of read-intensive workloads such as BI and analytics
Consider the read replica features in scenarios when delays in synching data between the master and replicas are acceptable
Create a replica in a different Azure region from the master for a disaster recovery plan, where a replica replaces the master in cases of regional disasters
Data storage on replica servers grows automatically without impacting workloads



Microsoft Azure

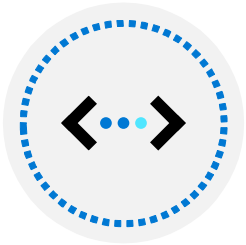
Microsoft

Module 2: Explore relational data in Azure- Segment 2

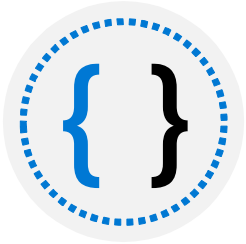
Demo: Provision an Azure SQL Database

Lesson 3: Query relational data in Azure

Lesson 3 objectives



Describe query techniques for data using the SQL language



Query relational data

Introduction to SQL

- SQL is a standard language for use with relational databases
- SQL standards are maintained by ANSI and ISO
- Proprietary RDBMS systems have their own extensions of SQL such as T-SQL, PL/SQL, pgSQL

SQL Statement types

DML	DDL	DCL
• Data Manipulation Language • Used to query and manipulate data • SELECT, INSERT, UPDATE, DELETE	• Data Definition Language • Used to define database objects • CREATE, ALTER, DROP, REMOVE	• Data Control Language • Used to manage security permissions • GRANT, REVOKE, DENY

Use DML statements

Statement	Description
SELECT	Select/read from a table
INSERT	Insert new rows in a table
UPDATE	Edit/Update existing rows in a table
DELETE	Delete existing rows in a table

Elements of the SELECT Statement

Clause	Expression
SELECT	<select list>
FROM	<table or view>
WHERE	<search condition>
GROUP BY	<group by list>
ORDER BY	<order by list>

Example of SELECT statement

```
SELECT EmployeeId, YEAR(OrderDate) AS OrderYear
FROM Sales.Orders
WHERE CustomerId = 71
GROUP BY EmployeeId, YEAR(OrderDate)
HAVING COUNT(*) > 1
ORDER BY EmployeeId, OrderYear;
```

Example of INSERT statement

- The INSERT … VALUES statement inserts a new row

```
INSERT INTO Sales.OrderDetails
       (orderid, productid, unitprice, qty, discount)
VALUES (10255,39,18,2,0.05);
```

- Table and row constructors add multirow capability to INSERT … VALUES

```
INSERT INTO Sales.OrderDetails
(orderid, productid, unitprice, qty, discount)

VALUES
(10256,39,18,2,0.05),
(10258,39,18,5,0.10);
```

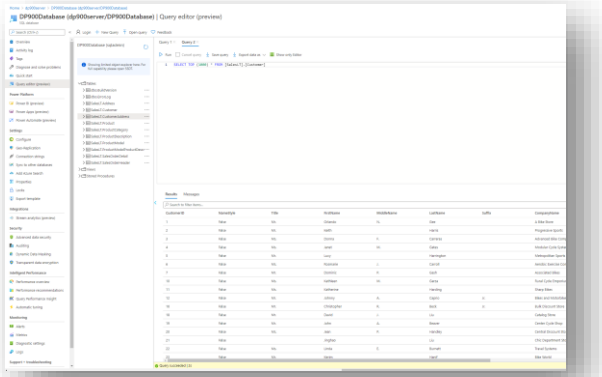
Use DDL statements

Statement	Description
CREATE	Create a new object in the database, such as a table or a view.
ALTER	Modify the structure of an object. For instance, altering a table to add a new column.
DROP	Remove an object from the database.
RENAME	Rename an existing object.

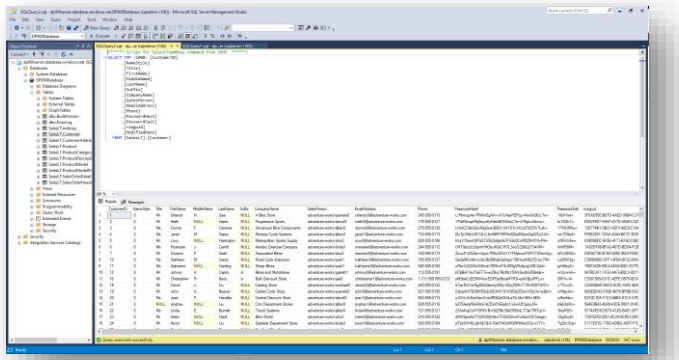
Example of CREATE statement

```
CREATE TABLE Mytable
(Mycolumn1 int NOT NULL PRIMARY KEY, Mycolumn2
VARCHAR(50) NOT NULL , Mycolumn2 VARCHAR(10) NOT NULL
```

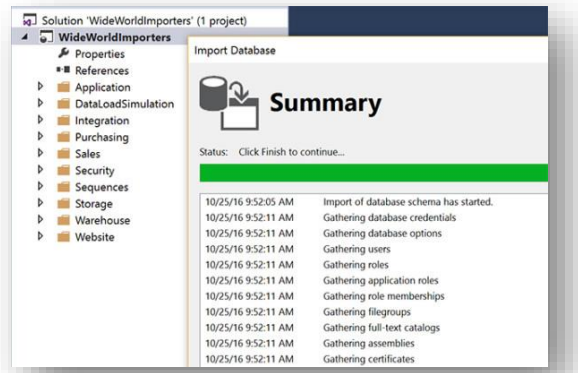
Query Tools



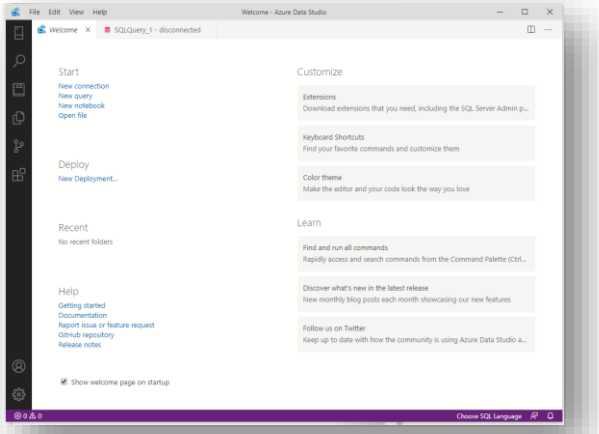
Azure Portal



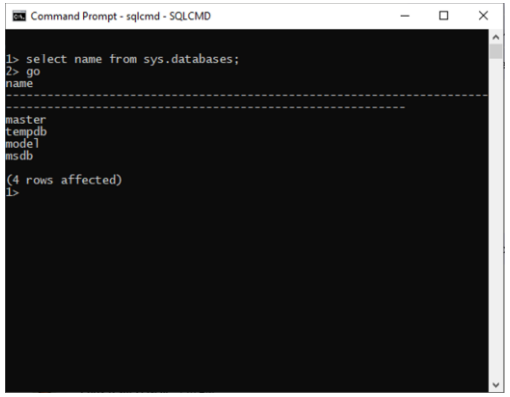
SQL Management Studio



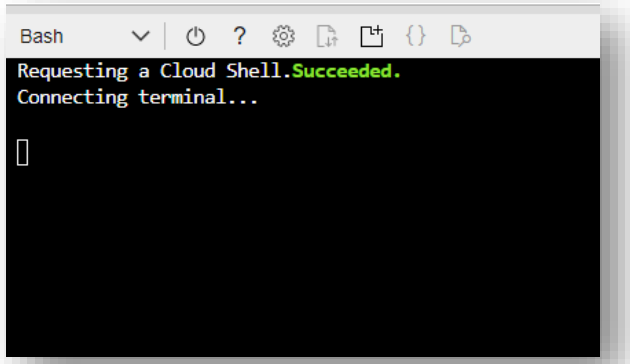
SQL Server Data Tools



Azure Data Studio



SQLCMD



Azure CLI / Cloud Shell

Query relational data in Azure SQL Database for PostgreSQL

Use PSQL to query a database

```
psql --host=<server-name>.postgres.database.azure.com --
username=<admin-user>@<server-name> --dbname=postgres
```

Query relational data in Azure SQL Database for MySQL

Use MySQL Workbench to query a database