DS684
Cloud Computing
Week 04

Regarding Labs and Assignments

- Class participation means more than Zoom attendance. You must actively participate in the discussion and labs, and answer questions.

- Must hit Submit button, otherwise no grade
- If you need extension in time, must send written request (***email***). Otherwise no grade and no makeup. Requests sent over Zoom chat do not count.
- For any technical difficulty (installation, Azure access, etc), you must send written explanation (***email***) before the deadline. Otherwise no grade and no makeup.

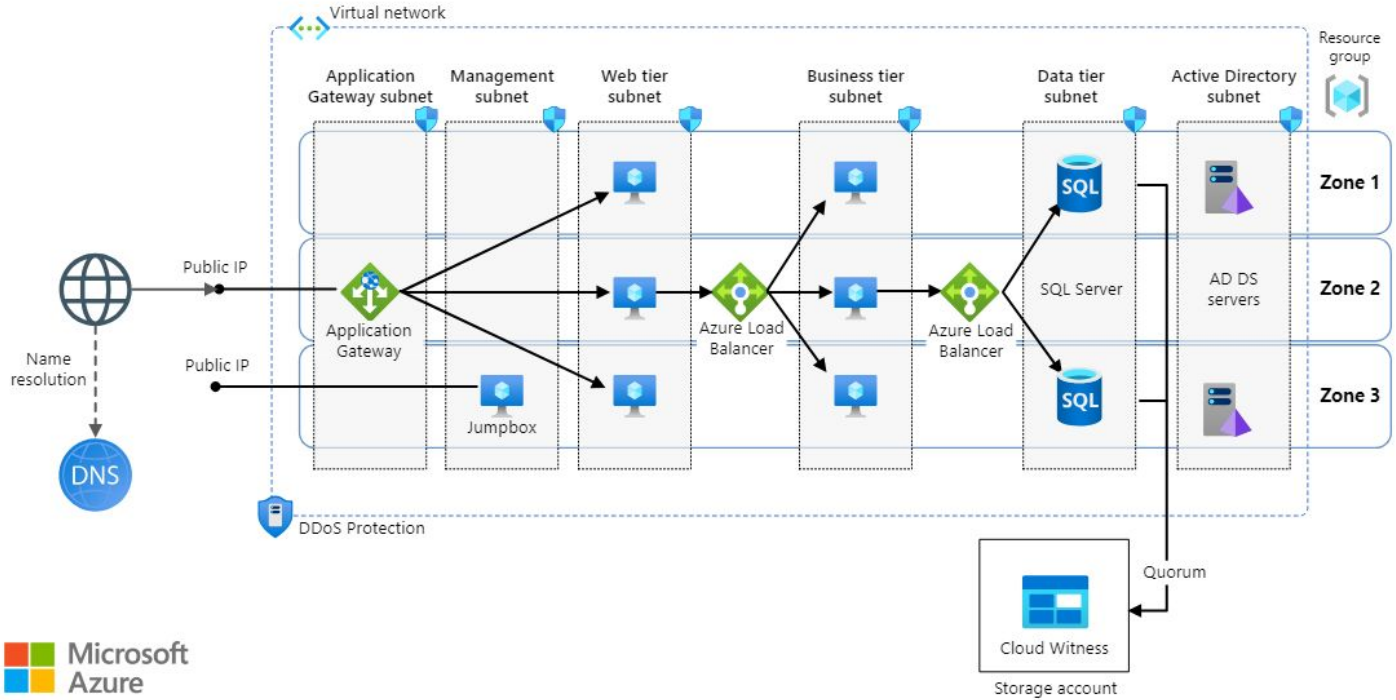
Management of Azure Cloud

- Management of resources
- Management of user
- Management of resource - user interaction

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Management of Azure Cloud



Management of Cloud Resources

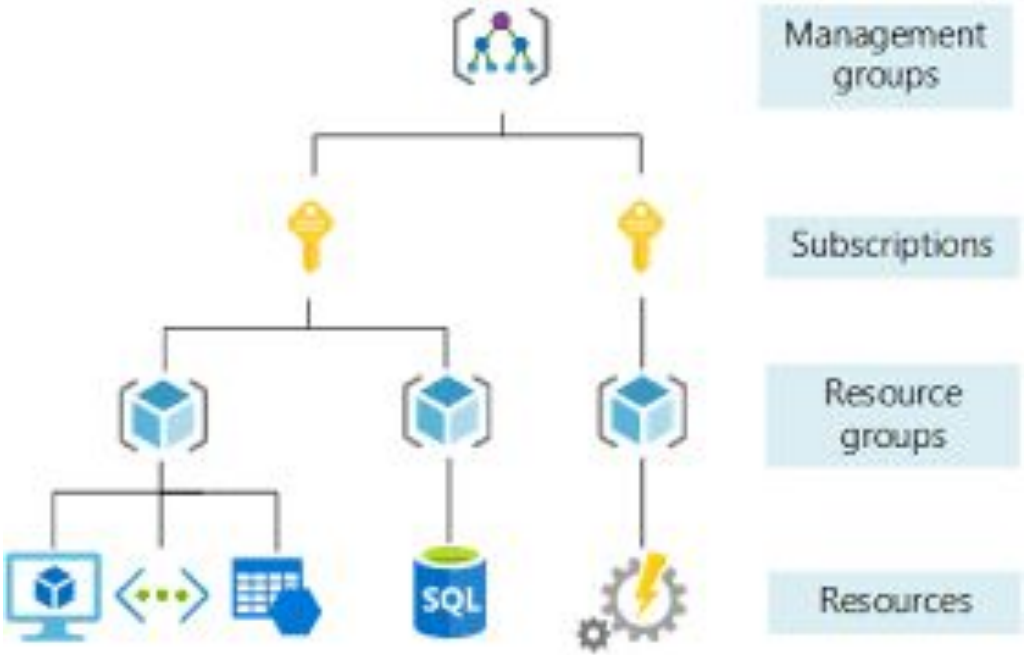
How to manage resources

- Group and hierarchy
- Tag
- Automation

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

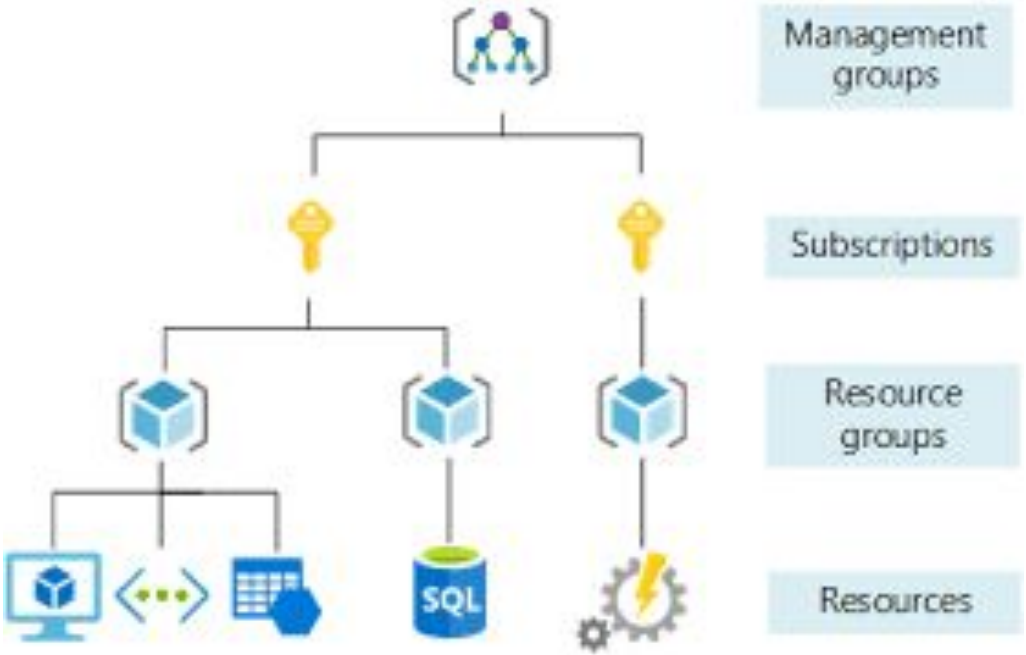
Grouping Hierarchy of Azure Cloud Resources



Resource Group

Resource Group: A resource group is a container that holds related resources for an Azure solution.

Usually, these resources should have the same expectation of life cycles (created and deleted together).

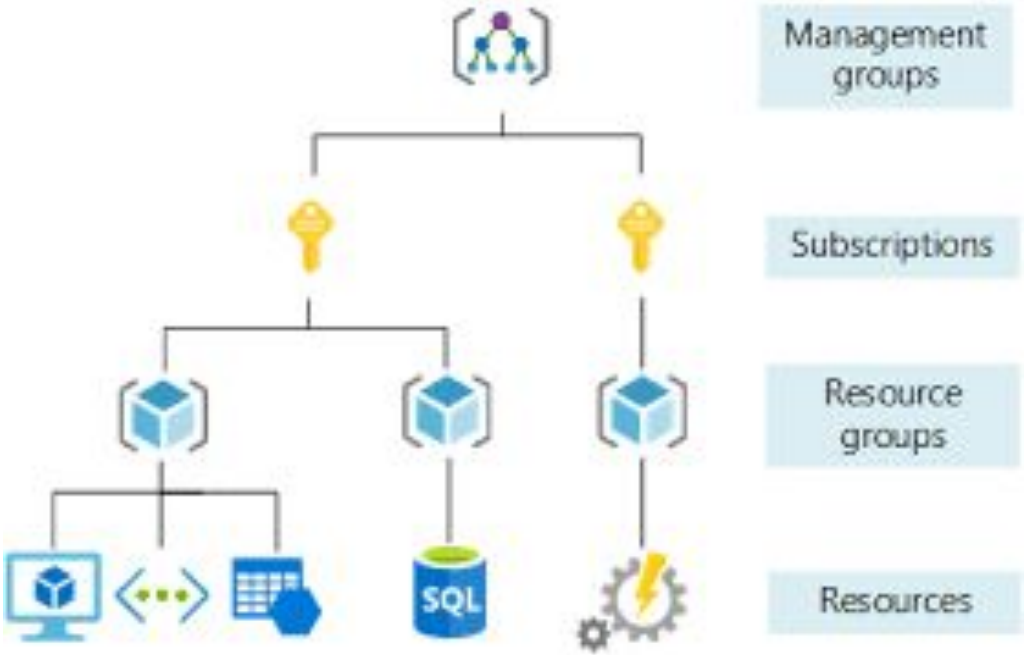


Subscription

Subscriptions logically associate user accounts with the resources that they create.

Subscriptions are used to manage costs and the resources serving same business activity.

Each subscription has limits or quotas on the amount of resources that it can create and use.



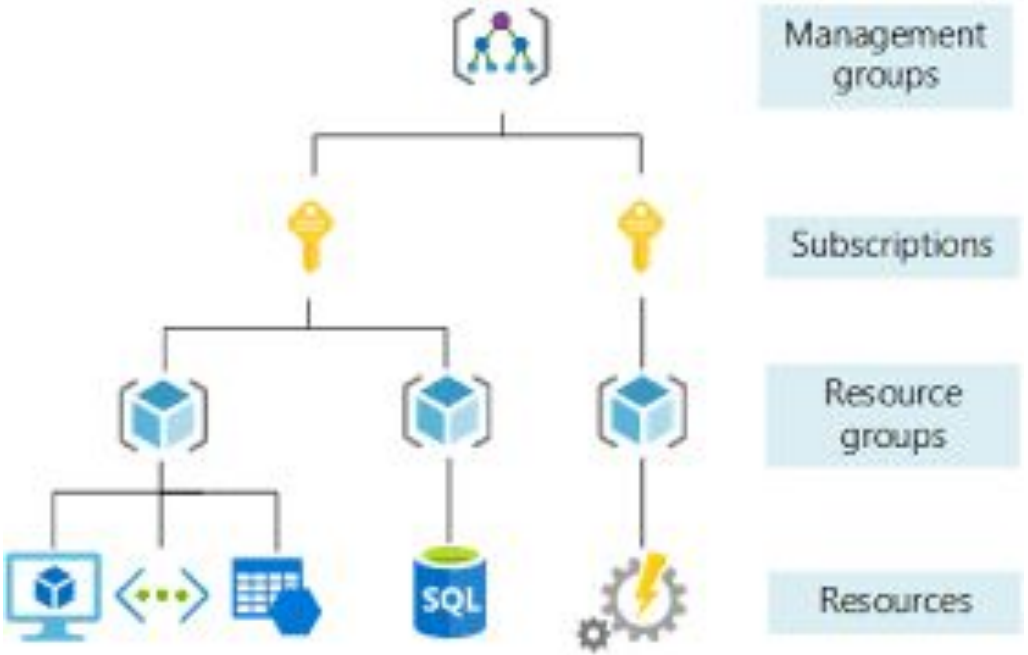
Grouping Size Limit of Azure Cloud Resources

Azure places limits on size of groups on different levels. You need to check it whenever applicable. Here are some examples:

Resource	Limit
Azure subscriptions associated with an Azure Active Directory tenant	Unlimited
Coadministrators per subscription	Unlimited
Resource groups per subscription	980
Azure Resource Manager API request size	4,194,304 bytes
Tags per subscription[1]	50
Unique tag calculations per subscription[2]	80,000
Subscription-level deployments per location	800[3]
Locations of Subscription-level deployments	10

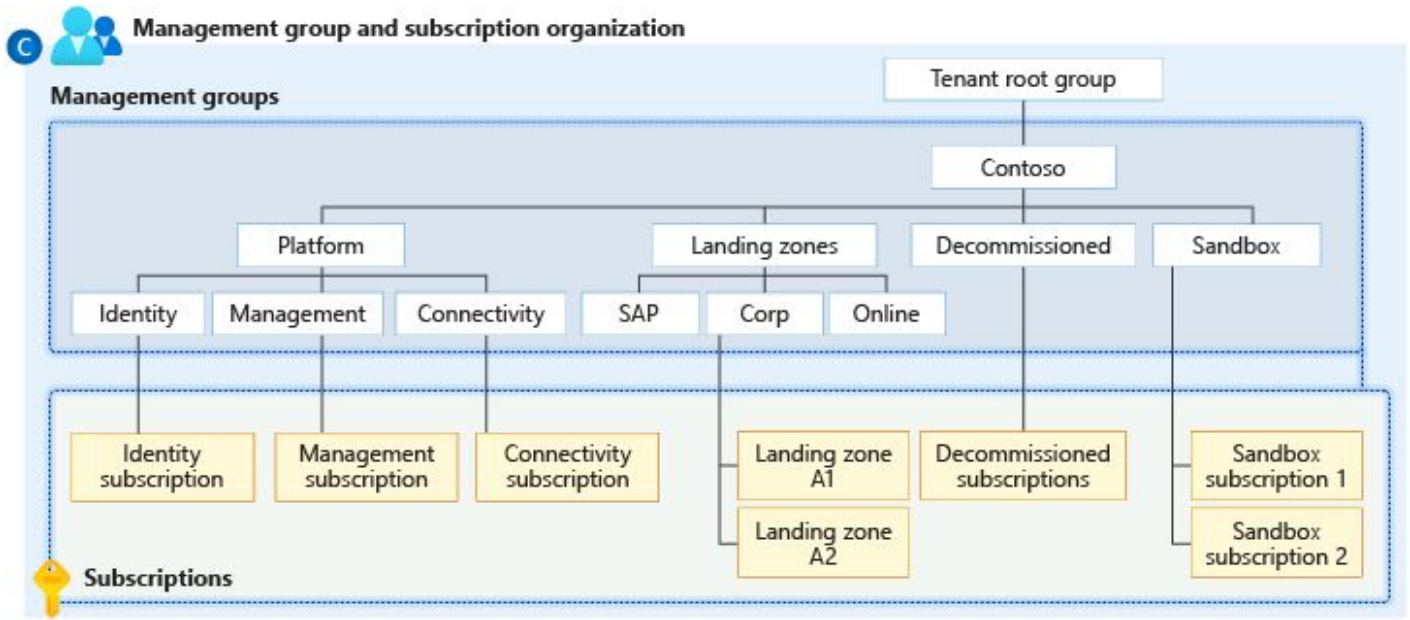
Management Group

Management groups help you manage access, policy, and compliance for multiple subscriptions.



Grouping Hierarchy of Azure Cloud Resources

Resource group and subscription cannot be embedded and thus will only form one flat layer. Management groups can be embedded and thus can form a hierarchy of their own.



Grouping Hierarchy of Azure Cloud Resources

Usually

- Management groups are used to map business units (departments, functions, teams).
- Subscriptions are used to map to *finance* accounts: cost centers and ledger accounts.
- Resource groups are used to group resources for a given workload. Resources within a resource group usually share the same lifecycle.

This is a best practice, not a mandatory rule. But it can make the administrator’s life easier.

Grouping Hierarchy of Azure Cloud Resources

Saint Peter's Univ	Management Group
Frank J. Guarini School of Business	Management Group
Master of Data Science Program	Management Group
DS684 Fall 2024 Session 1	Subscription
Week 2 Lab Virtual Machines	Resource Group

Grouping Hierarchy of Azure Cloud Resources

Saint Peter's Univ	Management Group
Frank J. Guarini School of Business	Management Group
Master of Data Science Program	Management Group
DS684 (or DS530)	Subscription
DS684 Fall 2024 Session 1	Resource Group

Grouping Hierarchy of Azure Cloud Resources

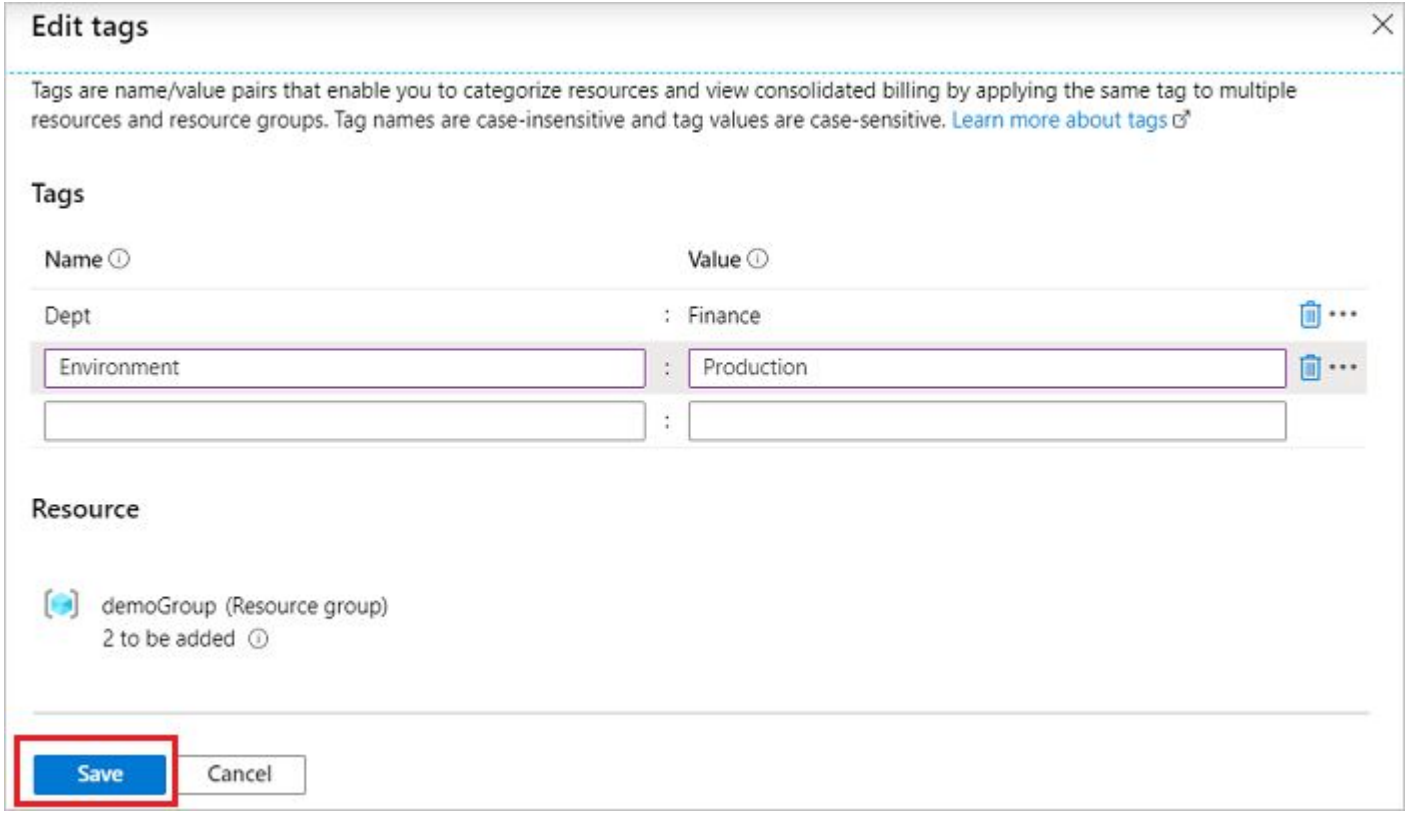
Saint Peter’s Univ	Management Group
Saint Peter’s Univ Subscription	Subscription
Saint Peter’s Univ Resource Group	Resource Group

Nothing stops you from doing something like this. But this is apparently not the best way to administrate your resources.

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

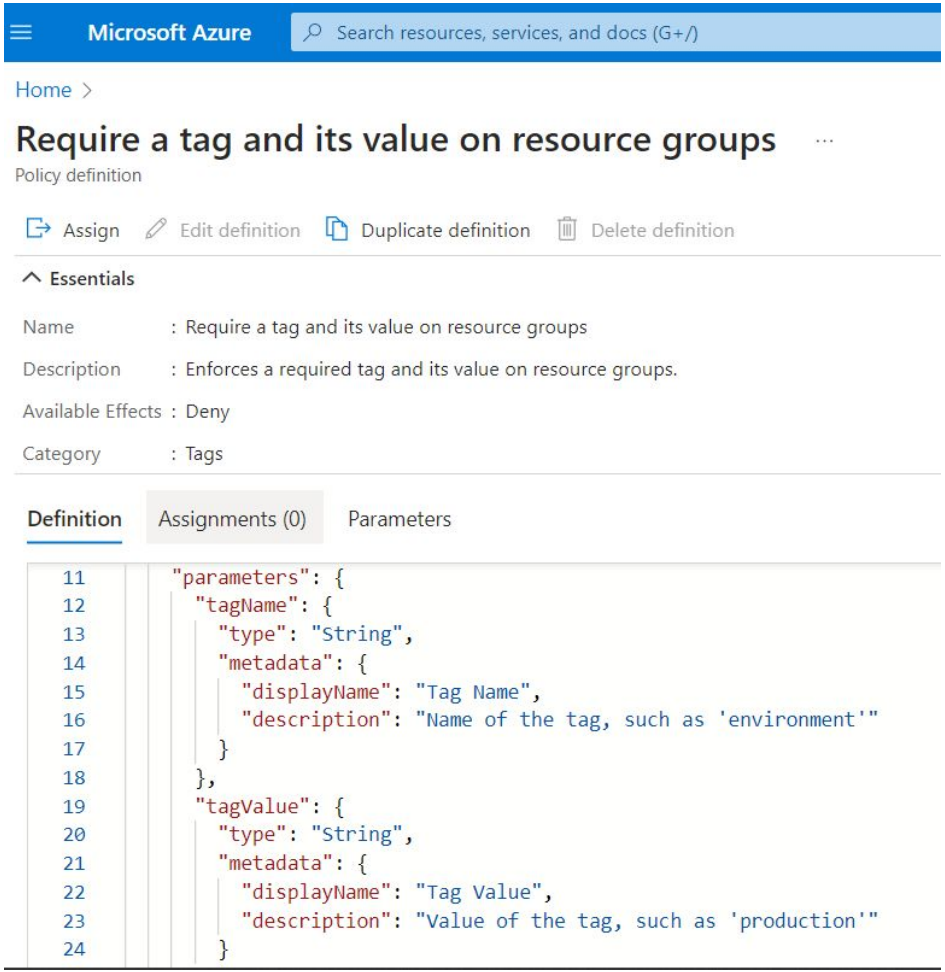
Tag


Edit tags
Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. Tag names are case-insensitive and tag values are case-sensitive. Learn more about tags
Tags
Name
Value
Dept : Finance
Environment : Production
Resource
demoGroup (Resource group)
2 to be added
Save
Cancel

Tag

Tagging can be enforced by administer

- Mandatory to have a certain tag
- Mandatory to have the tag value from a specified list
- Can be inherited from hierarchy/group



Usage of Tag

Provides a more flexible way of organizing resources

Different levels of statistics:

- Cost
- Usage (time and hours)

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Automation

Infrastructure as Code (IaC): Use code to generate infrastructure (resources), such as virtual machine, storage, database, etc.

Quote from Azure:

IaC model generates the same environment every time it deploys.

With IaC, DevOps teams can work together with a unified set of practices and tools to deliver applications and their supporting infrastructure rapidly and reliably at scale.

IaC Sample

```
# Create the Azure resource group
az group create `
  --name Ex_ResourceGroup `
  --location eastus

# Create the Azure virtual machine
az vm create `
  --resource-group Ex_ResourceGroup `
  --name Ex_VMFlask `
  --location eastus `
  --image UbuntuLTS `
  --public-ip-sku Standard `
  --admin-username vmadmin `
  --generate-ssh-keys
```

IaC Sample

```
# Open up the Flask http port
az vm open-port `
  --port 5000 `
  --resource-group Ex_ResourceGroup `
  --name Ex_VMFlask

# Get the public IP address of the virtual machine
az vm list-ip-addresses `
  --resource-group Ex_ResourceGroup `
  --name Ex_VMFlask

# One way of running command in the virtual machine
az vm run-command invoke `
  --resource-group Ex_ResourceGroup `
  --name Ex_VMFlask `
  --command-id RunShellScript `
  --scripts "sudo apt update"
```

IaC Sample Noted

- Scripts can be reused
- Can have variables and conditional branches
- Tags and groups can be inherited from hierarchy, or can also be specified in code

Automation

Benefit of IaC

- Consistency across environments
- Completeness of deployment (especially security)
- Speed of deployment
- Accountability (tracking changes)
- Lifecycle management (clean removal of footprints)

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Management of Users

How to manage users

- Group and hierarchy
- Tag
- Automation

Tag and automation for user management is very similar to those for resource management. Here we will only introduce hierarchical management of user by using Microsoft Entra ID.

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Azure Active Directory (Microsoft Entra ID)

Azure Active Directory

Azure Active Directory is a cloud-based user management service.

- It is the native user management service for Azure
  - There are other services that Azure can use but they are outside of the scope of this class
- It is where the administrator defines the users of the system
- It manages user creation/update/deletion
- It groups user into user groups for easier management

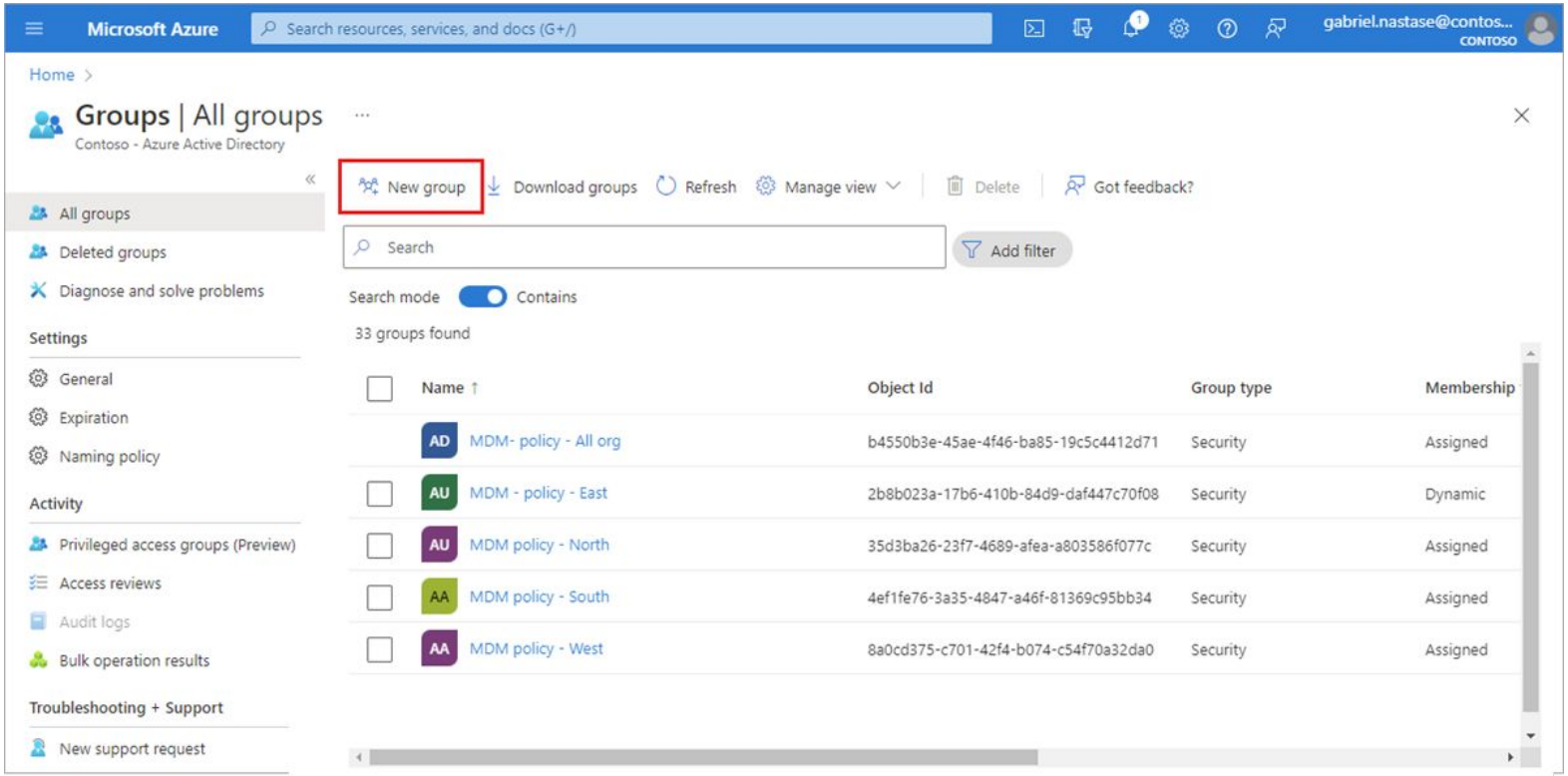
Directory in Azure Active Directory

Azure Active Directory is hosted as multiple Directories. Each directory consists of users, groups, and apps and is used to perform identity and access management functions for resources

Demo

- Tenant/account
- Group
- User

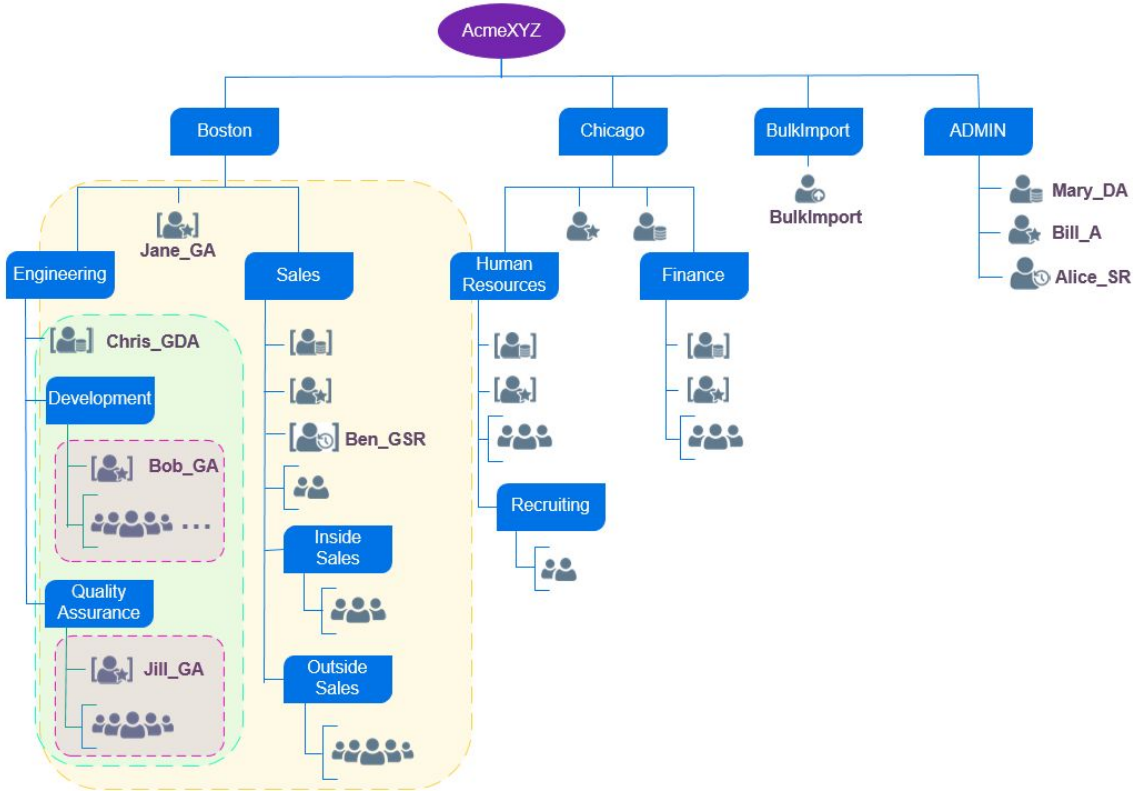
Azure Active Directory Group



Azure Active Directory Group Hierarchy

Groups can be embedded, thus forming a multi-level hierarchy.

Usually, there is a set of backbone hierarchy based on the organization hierarchy, then other sets based on access privilege.



Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Users - Resource Interaction

- In order for a user to use a cloud resource, two things must happen
  - Authentication
  - Authorization
- IAM: Identity and Access Management
  - Cloud IAM: The cloud service that performs IAM

Authentication

Authentication is the process of proving that you're who you say you are. This is achieved by verification of the identity of a person or device.

- Knowing who you are
- Authentication mechanism
  - In is simplest form: username + password:
  - Multi-factor authentication (MFA): Microsoft Authenticator
  - Federated login: ask another identity service provider to verify your identity

Authorization

Authorization is a process that grants or denies access to a system by verifying whether the accessor has the permissions to perform the requested action.

- Allowing a user or all users of a group to perform certain actions on a certain resource
  - Best practice is to perform authorization on group level to reduce management cost
- Role assignment
  - User group role list
  - Storage account role assignment

Principal

- User and group
- Service principal: An Azure service principal is a security identity used by applications, services, and automation tools to access specific Azure resources
- Managed identity: An identity to your applications and services running on Azure
    - Managed Identities are actually Service Principals. They are always linked to an Azure Resource, not to an application or 3rd party connector

Role Assignment

- Azure roles are a fundamental component of the Role-Based Access Control (RBAC) system, which is used to control and manage access to Azure resources.
- A collection of permissions that define what actions can be performed on Azure resources.

Entra ID Demo

- Entra ID User and Group
- Service Principal
- Managed Identity
- Authentication mechanism
  - In is simplest form: username + password: Demo
  - Multi-factor authentication (MFA): Microsoft Authenticator demo
- Role assignment
  - User group role list demo
  - Storage account role assignment demo
- Combining principals and roles

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

What does all these mean to a Data Scientist

- Usually you will not be involved in provisioning or managing any of these resource management or access management activities.
- You will request the administrators to provide you with access to certain resources, or privileges to provision/terminate certain resources.
- However it is very useful to understand what is behind all these requests

Summary of Prev Weeks

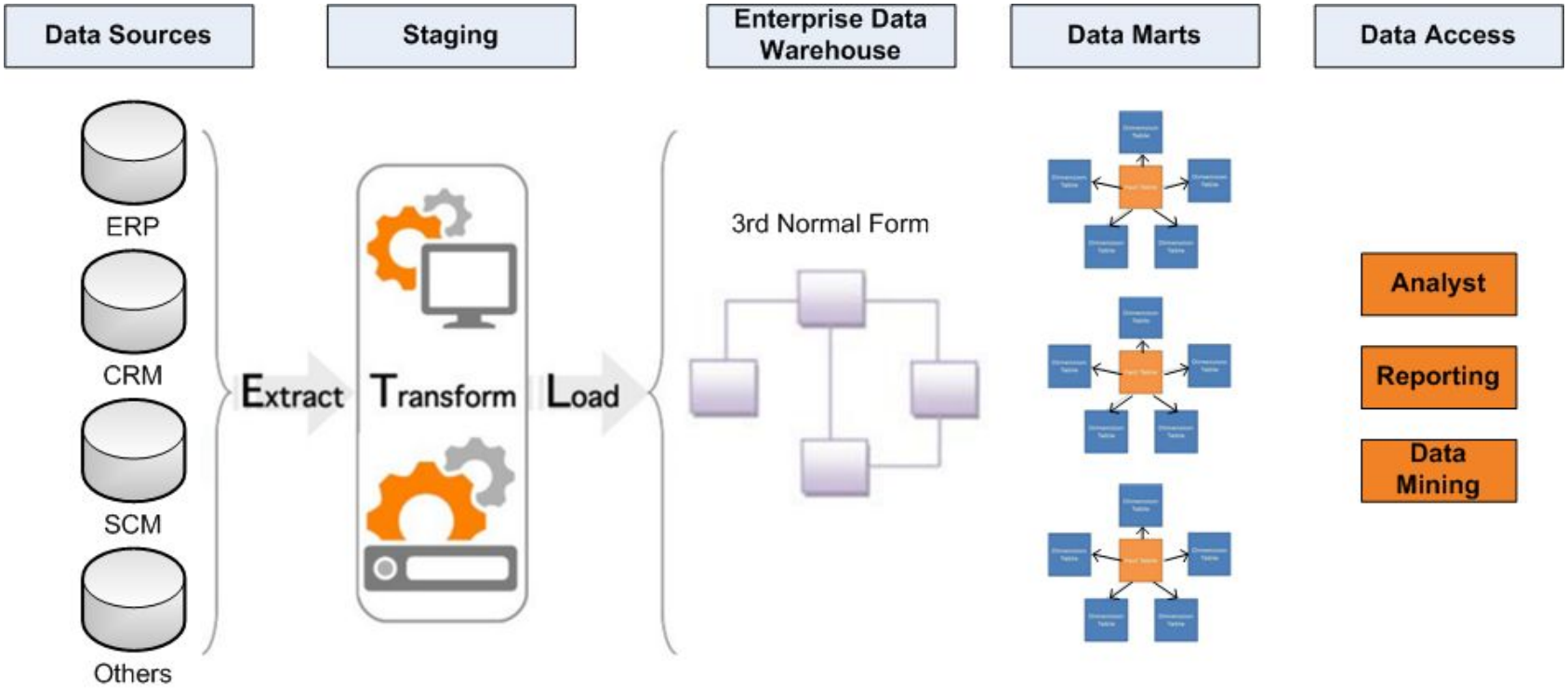
- Introduction to Cloud
- Virtual network and virtual machine
- Serverless computing: Container, Function as a Service
- Blob storage and file storage
- Cloud resource management

Summary of Prev Weeks

How are these related to data engineers and data scientists?

- These are fundamental services provided by Azure
  - Widely used in data engineering and data science as the fundamental services
  - Good enough to create a data science project (or any projects)
- There are also data specific services provided by Azure
  - Provide you with out of box services that you can utilize
  - Azure manages the infrastructure for you (think about serverless)

Typical Enterprise Data Flow



Agenda

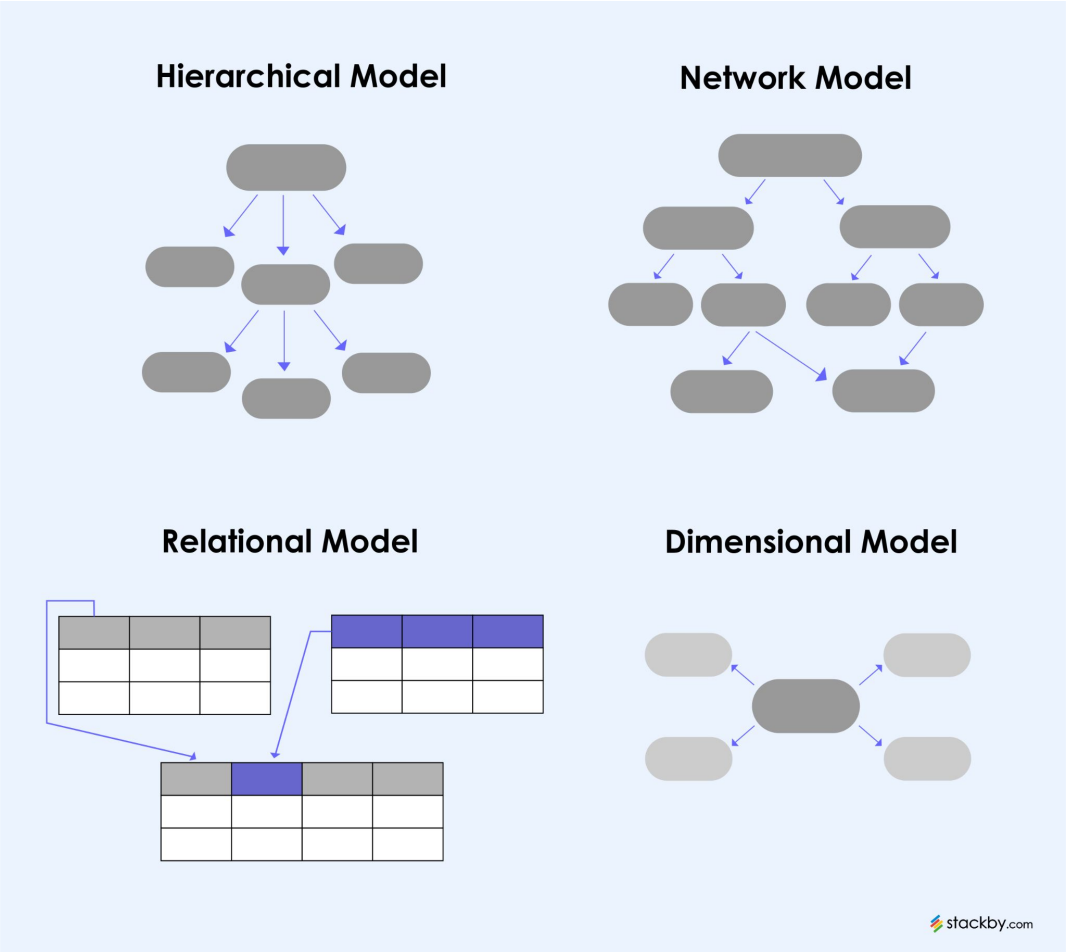
- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Data Architecture and Data Model

- A data architecture describes how data is managed – from collection through to transformation, distribution, and consumption.
- To collect, store, and analyze the data in a computer system, a well designed data model is necessary.

Data Modeling

- Hierarchical
- Network
- Relational
- Dimensional



Data Sources

Transactional Systems or Online Transaction Processing (OLTP) Systems

- Source Systems:
  - Enterprise applications (ERP, CRM, Finance, etc.)
  - eCommerce systems
  - Social networking sites
  - Internet of Things (IoT)

Hierarchical Database

Many source datasets are hierarchical based on nature of dataflow:

- Order -> Line Items
- Course -> Students

Hence the hierarchical model is a natural design pattern

- Record -> Subrecords

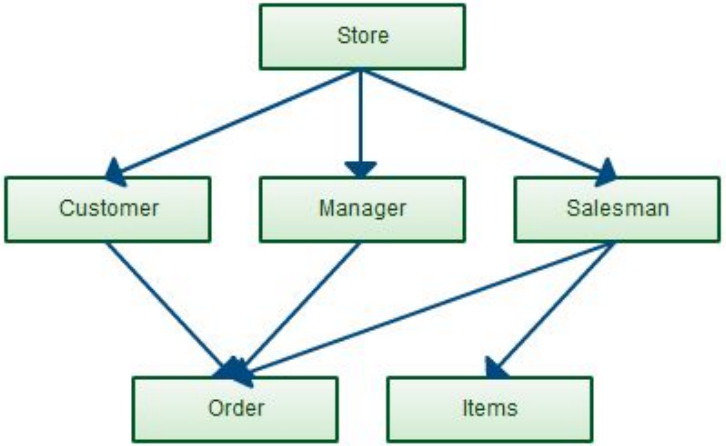
Network Database

However, each sub-record may have its own real world mapping, which is beyond the hierarchy:

- Products exist with or without an order
- Students exist with or without a course

Hence, network model was invented.

But, this model is complex and hard to maintain.



Relational Database

- Created in the 70’s
- Define all data as sets.
- Each set has a fixed schema, with format and order defined
- Easy to think about a data set as a 2 dimensional table

Name	Dry/Wet Food	Good Boy (Y/N)
Fido	Dry	Y
Rex	Wet	N
Bubbles	Dry	Y
Cujo	Wet	N

Tag #	Height (in)	Weight (lbs)
1573	15	21
2684	9	7
3795	27	130
4806	6	5

Tag #	Name	Breed	Color	Age
1573	Fido	Beagle	Brown/White	1.5
2684	Rex	Pekingese	White	9
3795	Bubbles	Rottweiler	Black	5
4806	Cujo	Chihuahua	Gold	4

Types of Data

Based on the way data is structured and stored:

- Semi-structured: do not have rigidly defined structure as traditional relational data but still retains some form of structure.
  - Examples: JSON data, XML documents, log files – Usually data coming from sources
- Structured: highly organized and formatted. typically fits into rows and columns, following a predefined schema or data model.
  - Examples: Relational databases – Usually data that has been processed
- Unstructured: no clear organization or format
  - Examples: Free text documents (e.g., Word files, PDFs), social media posts, multimedia content (image, audio, video)

Relational Database

Till this day, relational database is still the most important database in any enterprise setup (70%+ of database market). It is widely used for many tasks, including both transactional tasks and analytical tasks.

Advantages of Relational Database

- Built on top of a solid foundation of relational algebra
- A generic model that can fit into many different scenarios
- A simple but universal operating tool - Structured Query Language (SQL)
- Full ecosystem with established vendors and community
- Well understood. Large user base.

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Data Model for Analytics

- Relational schemas can be used for analytics, but they are not always the best choice in all analytics scenarios
  - optimized for transactional workloads. not always well-suited for complex analytical queries that involve aggregations, joins, and filtering.
- Dimensional Model is a better model for analytics
  - Organizes data based on business activities, which is the main subject of data analysis
  - Data is centered around the measurable attributes of the activities, such as amount, quantity, etc.
  - Data is aligned based on dimensions, attribute hierarchies that are descriptive regarding a certain aspect of the data

Star Schema

- A star schema is a type of data warehouse schema used in the design of data warehouses and business intelligence systems. It organizes data according to the business activity that it analyzes. It is a specific way of organizing data to optimize query performance, reporting, and analytics.
- It is a variation of relational data model. All data in star schema are still relational tables, but not necessarily 3NF compliant.

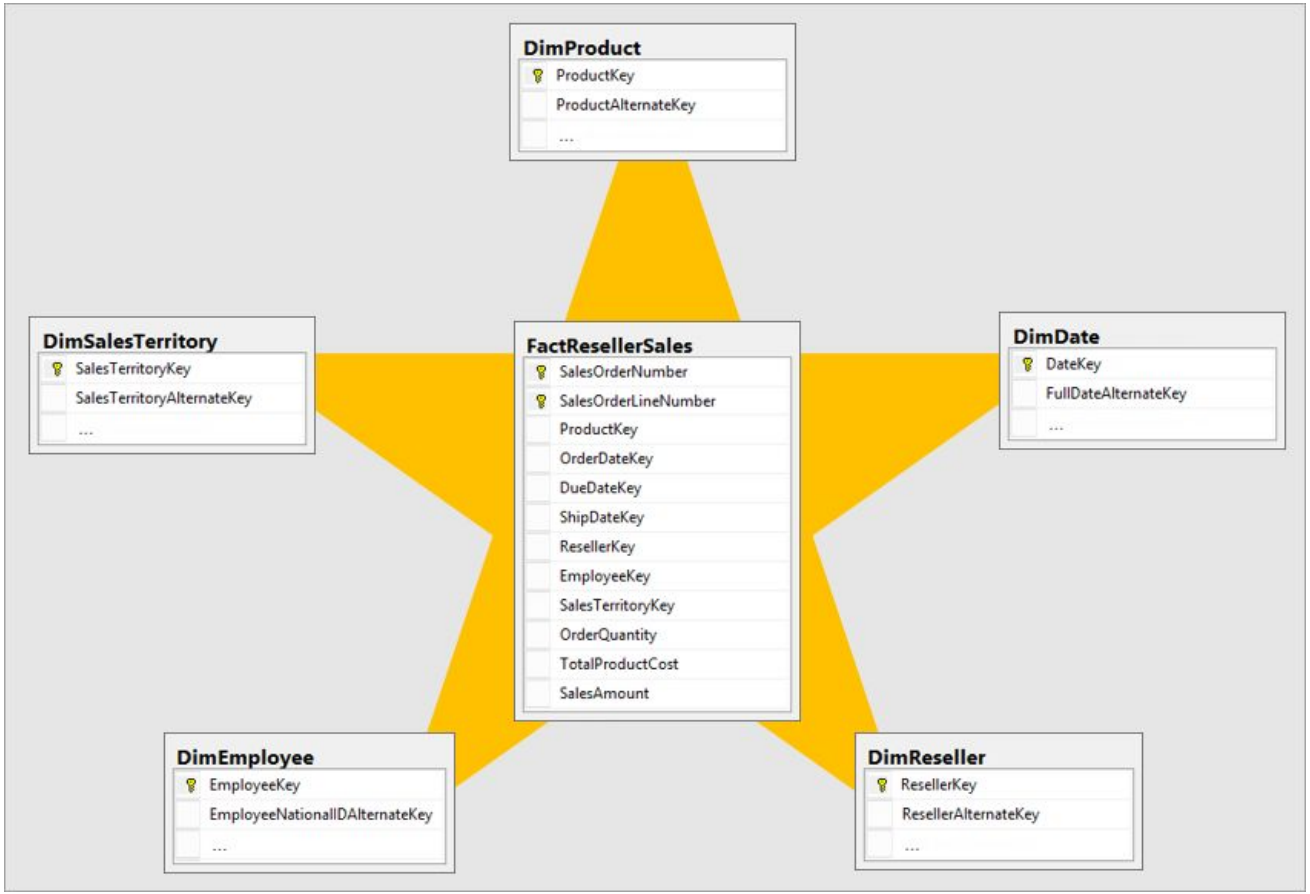
Star Schema

- The star schema consists of two primary types of tables: fact tables and dimension tables, organized in a way that simplifies data retrieval and analysis.
  - The fact table is the central table in the schema and contains measures or metrics, which are quantitative data points
  - Dimension tables represent descriptive attributes or characteristics related to the measures in the fact table

Star Schema

Designed around a business activity (sales).

Looks like a star (hence the name)



Design Star Schema using Kimball Enterprise Bus

- A concept and framework for organizing and managing data within an organization, especially in the context of a data warehousing and business intelligence (BI) environment
  - Identify business activities
  - For each business activity, define the facts and dimensions involved
  - Organize the facts and dimensions to build a matrix of fact-dimension relationship
  - (Optionally) consolidate facts with same dimensions

Kimball Enterprise Bus Matrix

Each row is a star schema

Business Process / Event	Time	Customer	Service	Rate Category	Local Svc Provider	Calling Party	Called Party	Long Dist Provider	Internal Organization	Employee	Location	Equipment Type	Supplier	Item Shipped	Account Status
Customer Billing	X	X	X	X	X			X			X				X
Service Orders	X	X	X		X			X	X	X	X	X			X
Trouble Reports	X	X	X		X	X		X	X	X	X	X	X	X	X
Yellow Page Ads	X	X		X		X			X	X	X				X
Customer Inquiries	X	X	X	X	X	X		X	X	X	X				X
Promotions & Communication	X	X	X	X	X	X		X	X	X	X	X	X	X	X
Billing Call Detail	X	X	X	X	X	X	X	X	X		X	X	X	X	X
Network Call Detail	X	X	X	X	X	X	X	X	X		X	X	X	X	X
Customer Inventory	X	X	X	X	X			X	X		X	X	X	X	X
Network Inventory	X		X						X	X	X	X	X	X	
Real Estate	X								X	X	X	X			
Labor & Payroll	X								X	X	X				
Computer Charges	X	X	X		X			X	X	X	X	X	X	X	
Purchase Orders	X								X	X	X	X	X	X	
Supplier Deliveries	X								X	X	X	X	X	X	

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Azure Relational Database

Azure offers several options of relational database management systems (RDBMS) out of box

- SQL Server
- PostgreSQL
- MySQL
- MariaDB
- Oracle

You can also run your own setup in a virtual machine

Azure SQL

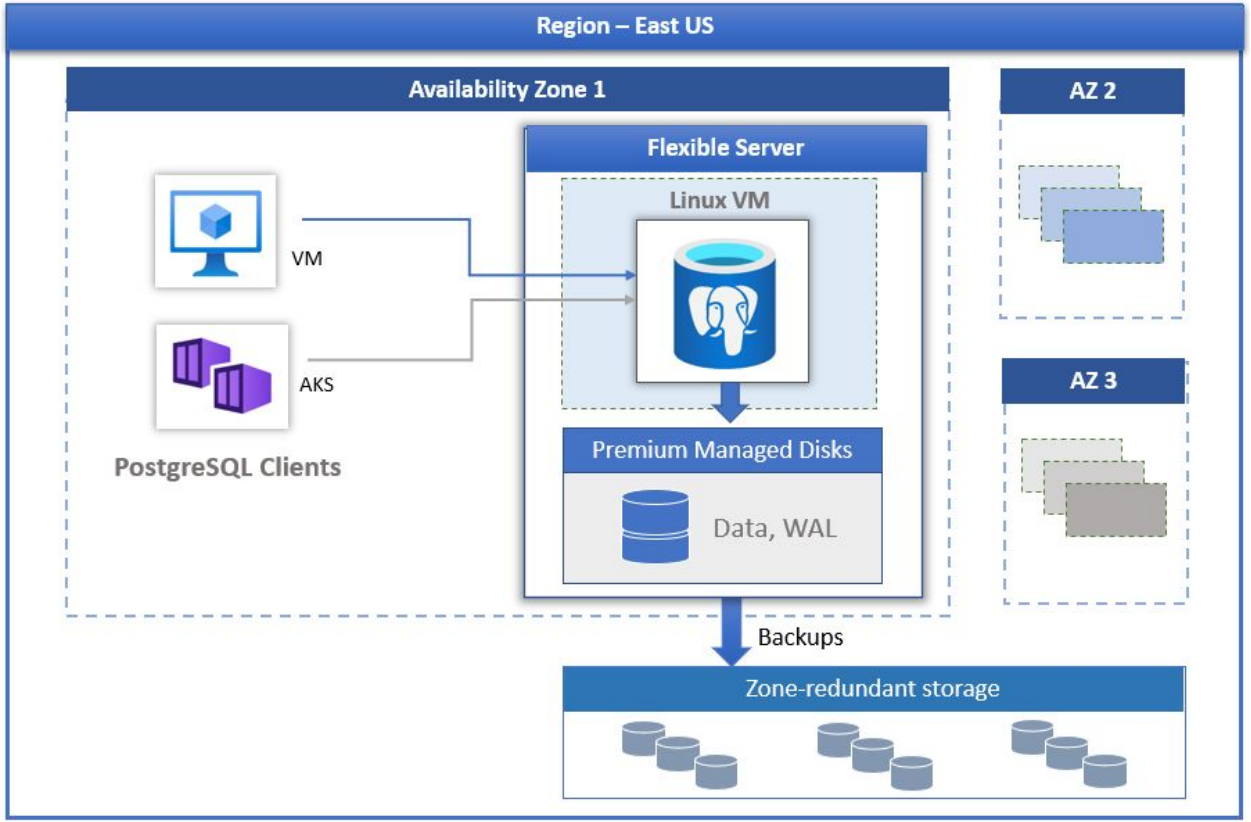
3 Types of SQL Server based on relational databases

- Azure SQL Database: More advanced, more automated, not 100% compatible with SQL Server.
- Azure SQL Managed Instance: Highly compatible with SQL Server. Requires some manual configuration and management.
- SQL Server on an Azure Virtual Machine: Most flexible. You run the operation and maintemamce.

Azure Database for PostgreSQL

Use PostgreSQL as an example:

- a virtual machine running a PostgreSQL database server
- data is stored redundantly in Azure storage



Azure Database for PostgreSQL

Advantages:

- Fully managed: backup, patch, monitoring and logging
- Redundant and highly available
- Pay as you go
- Security

Demo: Different options for Azure Database for PostgreSQL

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Azure Relational Database Lab

Purpose:

- Simulate the source data fetch activity in an enterprise data architecture

Instructor

- Create a PostgreSQL instance in Azure

Students:

- Use your own Jupyter Notebook to connect to the PostgreSQL
- Download the content of a table from PostgreSQL

Agenda

- Resource Management
  - Grouping hierarchy
  - Tags
  - Infrastructure as Code (IaC)
- User Management
  - Azure Entra ID
  - User Resource Interaction
- Data Architecture
  - Data model and relational database
  - Star schema
- Azure Relational Databases
  - Azure Relational Database Lab
  - Azure Key Vault

Azure Key Vault

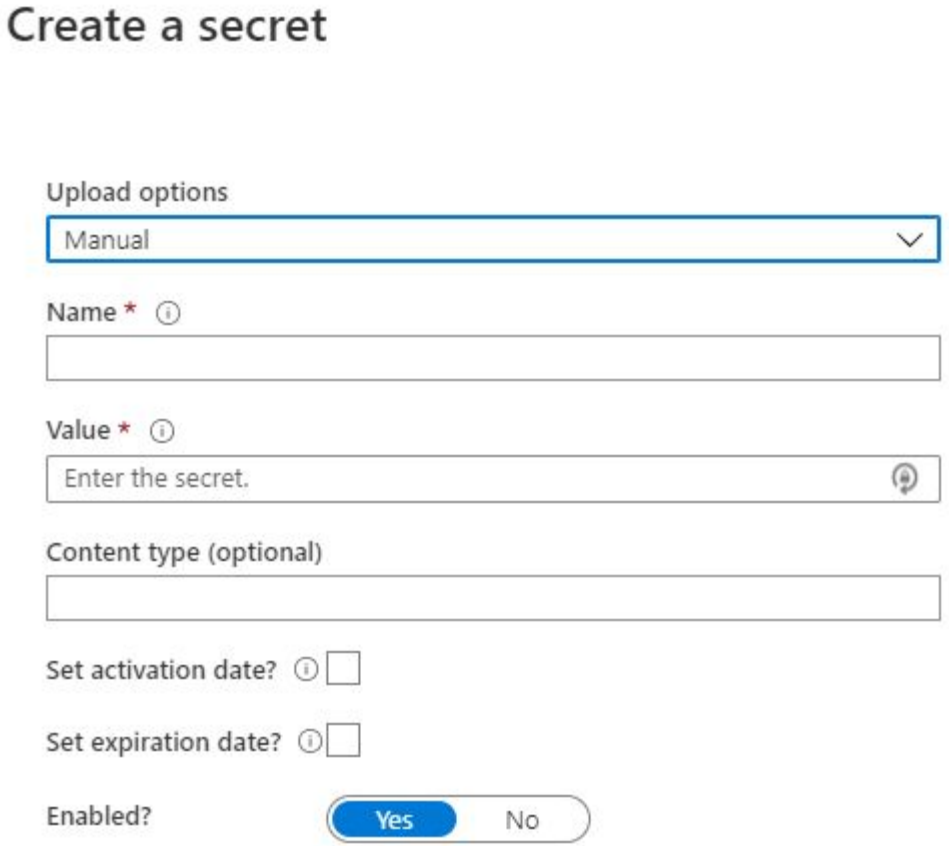
Securely storing and accessing secrets

- Don’t store your username and password in your code
- Don’t store your username and password in your code and upload the code to Github!!!
- Store in Key Vault

Demo of creating a key vault

Azure Key Vault

Create a secret in your key vault



Azure Key Vault

```
import os from azure.keyvault.secrets
import SecretClient from azure.identity
import DefaultAzureCredential

keyVaultName = os.environ["KEY_VAULT_NAME"]
KVUri = f"https://{keyVaultName}.vault.azure.net"
credential = DefaultAzureCredential()
client = SecretClient(
    vault url=KVUri,
    credential=credential)
secretName = input("Input a name for your secret > ")
secretValue = input("Input a value for your secret > ")
```

Azure Key Vault

```
print(f"Creating a secret in {keyVaultName} called '{secretName}'
with the value '{secretValue}' ...")
client.set_secret(secretName, secretValue)
print("Done.")

print(f"Retrieving your secret from {keyVaultName}.")
retrieved secret = client.get_secret(secretName)
print(f"Your secret is '{retrieved_secret.value}'.")

print(f"Deleting your secret from {keyVaultName} ...")
poller = client.begin_delete_secret(secretName)
deleted secret = poller.result()
print("Done.")
```

Assignment