DP-900 : Microsoft Azure

Azure Data Fundamentals

YOUR PRESENTER

Mohammed Arif, PhD

GenAI Architect & Data Scientist

Mohammed Arif has more than eighteen (18+) years of working experience in Information Communication and Technology (ICT) industry. The highlights of his career are more than nine (9) years of holding various senior management and/or C-Level and had six (6) years of international ICT consultancy exposure in various countries (APAC and Australia), specially on Big Data, Data Engineering, Machine Learning and AI arena.

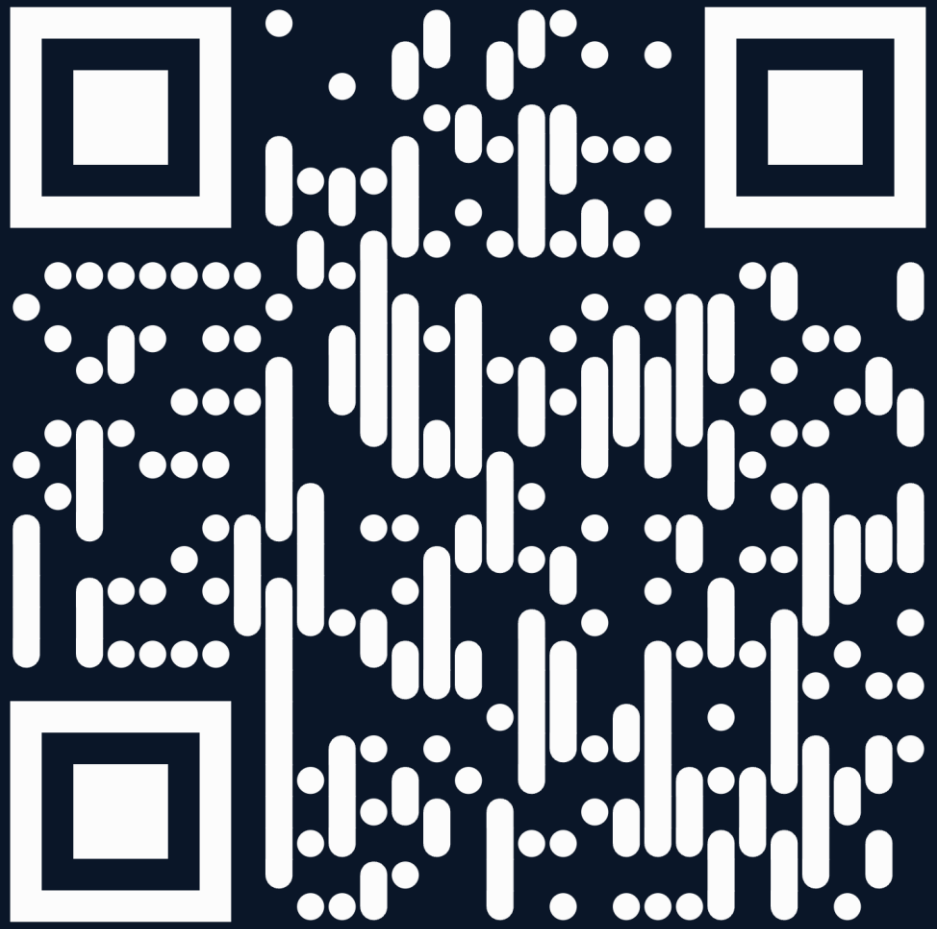
He is also Certified Trainer for Microsoft & Cloudera.

/arifmazumder

Contents

COURSE RESOURCES

https://arif.works/mbb/



Scan QR code or visit URL above for course materials

Microsoft

Explore fundamentals of data

Agenda

- Core data concepts
- Data roles and services

1: Core data concepts

What is data?

Values used to record information – Often representing *entities* that have one or more *attributes*

Structured

Customer				
ID	**FirstName**	**LastName**	**Email**	**Address**
1	Joe	Jones	joe@litware.com	1 Main St.
2	Samir	Nadoy	samir@northwind.com	123 Elm Pl.

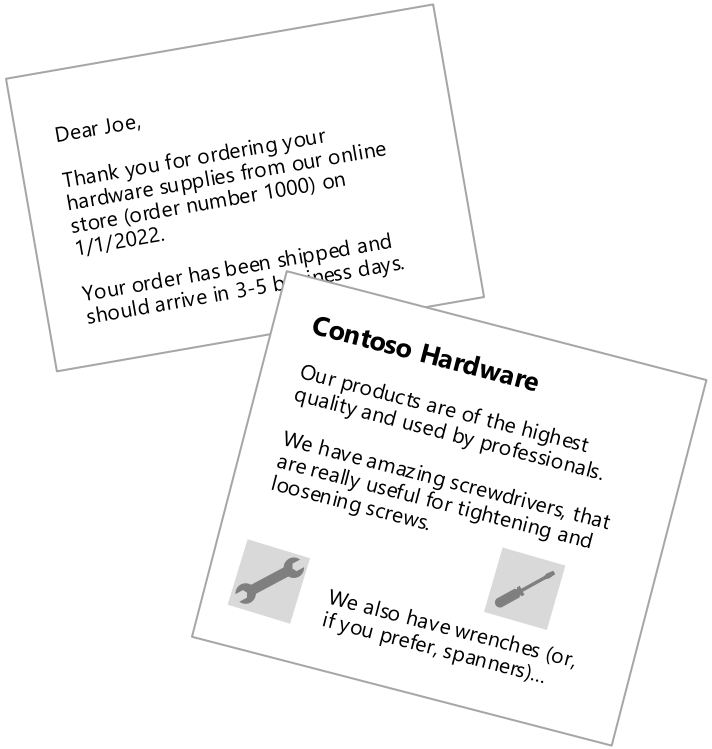
Product		
ID	**Name**	**Price**
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

Semi-structured

```
{
    "firstName": "Joe",
    "lastName": "Jones",
    "address":
    {
        "streetAddress": "1 Main
St.",
        "city": "New York",
        "state": "NY",
        "postalCode": "10099"
    },
    "contact":
    [
        {
          "type": "home",
          "number": "555 123-
        },
        {
          "type": "email",
          "address":
"joe@litware.com"
        }
    ]
}
```

```
{
    "firstName": "Samir",
    "lastName": "Nadoy",
    "address":
    {
        "streetAddress": "123 Elm
Pl.",
        "unit": "500",
        "city": "Seattle",
        "state": "WA",
        "postalCode": "98999"
    },
    "contact":
    [
        {
          "type": "email",
          "address":
"samir@northwind.com"
        }
    ]
}
```

Unstructured



How is data stored?

Files

Delimited Text

```
FirstName,LastName,Email
Joe,Jones,joe@litware.com
Samir,Nadoy,samir@northwind.com
```

JavaScript Object Notation (JSON)

```
{
  "customers":
  [
    { "firstName": "Joe", "lastName": "Jones"},
    { "firstName": "Samir", "lastName": "Nadoy"}
  ]
}
```

Extensible Markup Language (XML)

```
<Customer firstName="Joe" lastName="Jones"/>
```

Binary Large Object (BLOB)

```
101101011010101100010...
```

Optimized formats:

- Avro, ORC, Parquet

Databases

Relational

Customer

ID	Email	Address
1	joe@litware.com	1 Main St.
2	samir@northwind.com	123 Elm Pl.

Product

ID	Name	Price
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

Order

OrderNo	OrderDate	Customer
1000	1/1/2022	1
1001	1/1/2022	2

LineItem

OrderNo	ItemNo	ProductID	Quantity
1000	1	123	1
1000	2	201	2
1001	1	123	2

Non-Relational

Products

Key	Value
123	"Hammer ($2.99)"
162	"Screwdriver ($3.49)"
201	"Wrench ($4.25)"

Key-Value

Customers

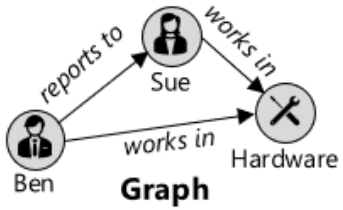
Key	Document
1	{ "name": "Joe Jones" }
2	{ "name": "Samir Nadoy" }

Document

Orders

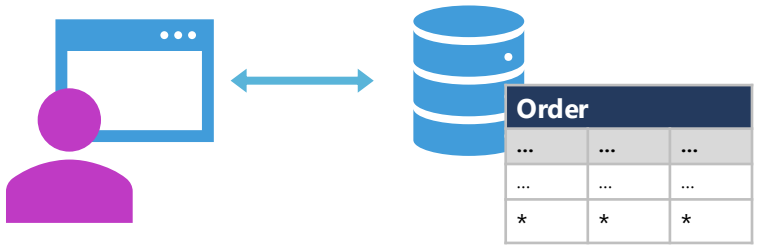
Key	Customer		Product	
	Name	Address	Name	Price
1000	Joe Jones	1 Main St.	Hammer	2.99
1001	Samir Nadoy	123 Elm Pl.	Wrench	4.25

Column Family



Graph

Operational data workloads



Data is stored in a database that is optimized for *online transactional processing* (OLTP) operations that support applications

A mix of *read* and *write* activity

For example:

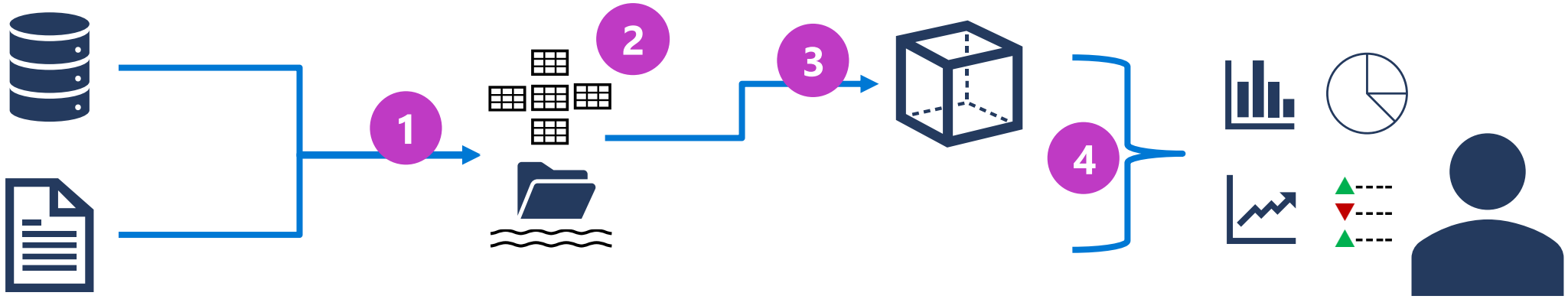
- Read the *Product* table to display a catalog
- Write to the *Order* table to record a purchase

Data is stored using *transactions*

Transactions are "ACID" based:

- **Atomicity** – Each transaction is treated as a single unit of work, which succeeds completely or fails completely
- **Consistency** – Transactions can only take the data in the database from one valid state to another
- **Isolation** – Concurrent transactions cannot interfere with one another
- **Durability** – When a transaction has succeeded, the data changes are persisted in the database

Analytical data workloads



1. Operational data is extracted, transformed, and loaded (ETL) into a *data lake* for analysis
2. Data is loaded into a schema of tables - typically in a Spark-based *data lakehouse* with tabular abstractions over files in the data lake, or a data warehouse with a fully relational SQL engine
3. Data in tables may be aggregated and loaded into an online analytical processing (OLAP) model, or cube
4. The files in the data lake, relational tables, and analytical model can be queried to produce *reports* and *dashboards*

2: Data roles and services

Data professional roles

Database Administrator

- Database provisioning, configuration and management
- Database security and user access
- Database backups and resiliency
- Database performance monitoring and optimization

Data Engineer

- Data integration pipelines and ETL processes
- Data cleansing and transformation
- Analytical data store schemas and data loads

Data Analyst

- Analytical modeling
- Data reporting and summarization
- Data visualization

Microsoft cloud services for data

Operational data workloads

Azure SQL

- Family of SQL Server based relational database services

Azure Database for open-source

- Maria DB, MySQL, PostgreSQL

Azure Cosmos DB

- Highly scalable non-relational database system

Azure Storage

- File, blob, and table storage
- Hierarchical namespace for data lake storage

Analytical data workloads

Software-as-a-Service (SaaS)

Microsoft Fabric

Unified, SaaS based analytics platform based on open and governed lakehouse:

- Data ingestion and ETL
- Data Lakehouse
- Data Warehouse
- Data Science and ML
- Realtime Analytics
- Data visualization
- Data governance and management

Microsoft Purview

Solution for enterprise-wide data governance and discoverability:

- Create a map of your data and track data lineage across multiple data sources.
- Enforce data governance across the enterprise and ensure the integrity of data.

Platform-as-a-Service (PaaS)

Azure Databricks

- Apache Spark analytics and data processing

others...

Microsoft

2: Explore fundamentals of relational data in Azure

Agenda

- Explore relational data concepts
- Explore Azure services for relational data

1: Explore relational data concepts

Relational tables

- Data is stored in tables
- Tables consists of rows and columns
- All rows have the same columns
- Each column is assigned a datatype

Customer

ID	FirstName	MiddleName	LastName	Email	Address	City
1	Joe	David	Jones	joe@litware.com	1 Main St.	Seattle
2	Samir		Nadoy	samir@northwind.com	123 Elm Pl.	New York

Product

ID	Name	Price
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

Order

OrderNo	OrderDate	Customer
1000	1/1/2022	1
1001	1/1/2022	2

LineItem

OrderNo	ItemNo	ProductID	Quantity
1000	1	123	1
1000	2	201	2
1001	1	123	2

Normalization

Sales Data				
OrderNo	**OrderDate**	**Customer**	**Product**	**Quantity**
1000	1/1/2022	Joe Jones, 1 Main St, Seattle	Hammer ($2.99)	1
1000	1/1/2022	Joe Jones- 1 Main St, Seattle	Screwdriver ($3.49)	2
1001	1/1/2022	Samir Nadoy, 123 Elm Pl, New York	Hammer ($2.99)	2
…	…	…	…	…

- Separate each *entity* into its own table
- Separate each discrete *attribute* into its own column
- Uniquely identify each entity instance (row) using a *primary key*
- Use *foreign key* columns to link related entities

Customer				
ID	**FirstName**	**LastName**	**Address**	**City**
1	Joe	Jones	1 Main St.	Seattle
2	Samir	Nadoy	123 Elm Pl.	New York

Order		
OrderNo	**OrderDate**	**Customer**
1000	1/1/2022	1
1001	1/1/2022	2

Product		
ID	**Name**	**Price**
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

LineItem			
OrderNo	**ItemNo**	**ProductID**	**Quantity**
1000	1	123	1
1000	2	201	2
1001	1	123	2

Structured Query Language (SQL)

- SQL is a standard language for use with relational databases
- Standards are maintained by ANSI and ISO
- Most RDBMS systems support proprietary extensions of standard SQL

Data Definition Language (DDL)	Data Control Language (DCL)	Data Manipulation Language (DML)
CREATE, ALTER, DROP, RENAME	*GRANT, DENY, REVOKE*	*INSERT, UPDATE, DELETE, SELECT*

`CREATE TABLE Product`
`(`
`  ProductID INT PRIMARY KEY,`
`  Name VARCHAR(20) NOT NULL,`
`  Price DECIMAL NULL`
`);`
	`GRANT SELECT, INSERT, UPDATE`
	`ON Product`
	`TO user1;`
		`SELECT Name, Price`
		`FROM Product`
		`WHERE Price > 2.50`
		`ORDER BY Price;`

Product

ID	Name	Price

Product

ID	Name	Price
123	Hammer	2.99
162	Screwdriver	3.49
201	Wrench	4.25

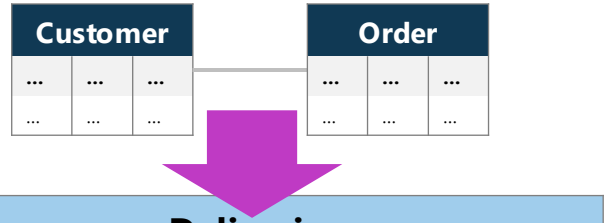
Name	Price
Hammer	2.99
Screwdriver	3.49
Wrench	4.25

Other common database objects

Views

Pre-defined SQL queries that behave as virtual tables

```
CREATE VIEW Deliveries
AS
SELECT o.OrderNo, o.OrderDate,
       c.Address, c.City
FROM Order AS o JOIN Customer AS c
ON o.Customer = c.ID;
```



Deliveries

OrderNo	OrderDate	Address	City
1000	1/1/2022	1 Main St.	Seattle
1001	1/1/2022	123 Elm Pl.	New York

Stored Procedures

Pre-defined SQL statements that can include parameters

```
CREATE PROCEDURE RenameProduct
          @ProductID INT,
          @NewName VARCHAR(20)
AS
UPDATE Product
SET Name = @NewName
WHERE ID = @ProductID;
...
EXEC RenameProduct 201, 'Spanner';
```

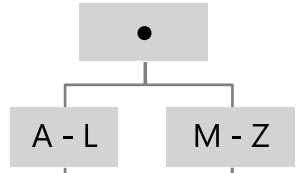
Product

ID	Name	Price
201	~~Wrench~~ **Spanner**	4.25

Indexes

Tree-based structures that improve query performance

```
CREATE INDEX idx_ProductName
ON Product(Name);
```



Product

ID	Name	Price
123	Hammer	2.99
162	Screwdriver	3.49
201	Spanner	4.25

2: Explore Azure services for relational data

Azure SQL

Family of SQL Server based cloud database services

SQL Server on Azure VMs

- Guaranteed compatibility to SQL Server on premises
- Customer manages everything – OS upgrades, software upgrades, backups, replication
- Pay for the server VM running costs and software licensing, not per database
- Great for hybrid cloud or migrating complex on-premises database configurations

Azure SQL Managed Instance

- Near 100% compatibility with SQL Server on-premises
- Automatic backups, software patching, database monitoring, and other maintenance tasks
- Use a single instance with multiple databases, or multiple instances in a pool with shared resources
- Great for migrating most on-premises databases to the cloud

Azure SQL Database

- Core database functionality compatibility with SQL Server
- Automatic backups, software patching, database monitoring, and other maintenance tasks
- *Single database* or *elastic pool* to dynamically share resources across multiple databases
- Great for new, cloud-based applications

IaaS

PaaS

Azure Database services for open-source

Azure managed solutions for common open-source RDBMSs

Azure Database for MySQL

- PaaS implementation of MySQL in the Azure cloud, based on the MySQL Community Edition
- Commonly used in Linux, Apache, MySQL, PHP (LAMP) application architectures

Azure Database for MariaDB

- An implementation of the MariaDB Community Edition database management system adapted to run in Azure
- Compatibility with Oracle Database

Azure Database for PostgreSQL

- Database service in the Microsoft cloud based on the PostgreSQL Community Edition database engine
- Hybrid relational and object storage

PaaS

Microsoft

3: Explore fundamentals of non-relational data in Azure

Agenda

- Fundamentals of Azure Storage
- Fundamentals of Azure Cosmos DB

1: Fundamentals of Azure Storage

Azure Blob Storage

Storage for data as binary large objects (BLOBs)

Block blobs

- Large, discrete, binary objects that change infrequently
- Blobs can be up to 4.7 TB, composed of blocks of up to 100 MB
  - A blob can contain up to 50,000 blocks

Page blobs

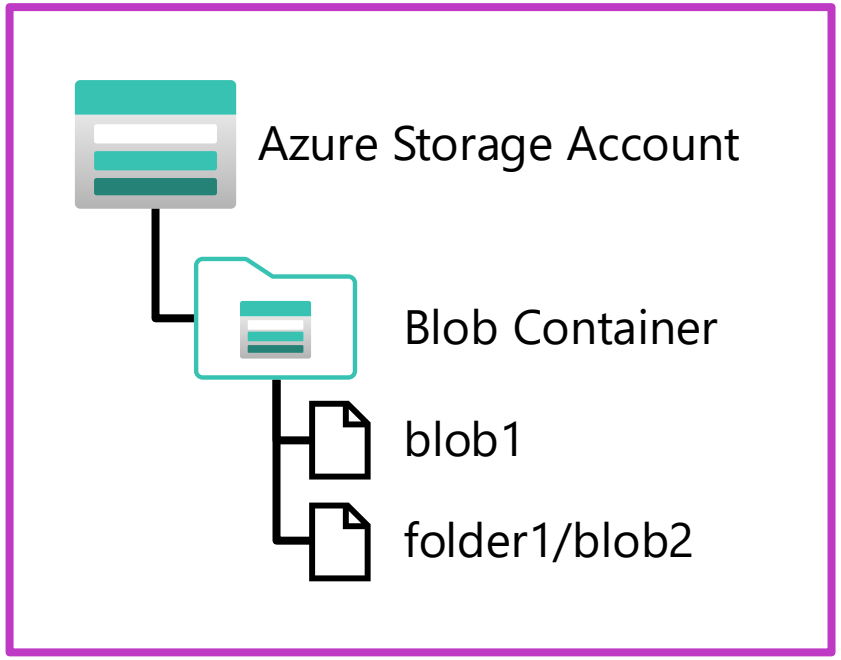
- Used as virtual disk storage for VMs
- Blobs can be up to 8 TB, composed of fixed sized-512 byte pages

Append blobs

- Block blobs that are used to optimize append operations
- Maximum size just over 195 GB – each block can be up to 4 MB

Per-blob storage tiers

- Hot – Highest cost, lowest latency
- Cool – Lower cost, higher latency
- Archive – Lowest cost, highest latency



Blobs can be organized in virtual directories, but each path is considered a single blob in a flat namespace – folder level operations are not supported

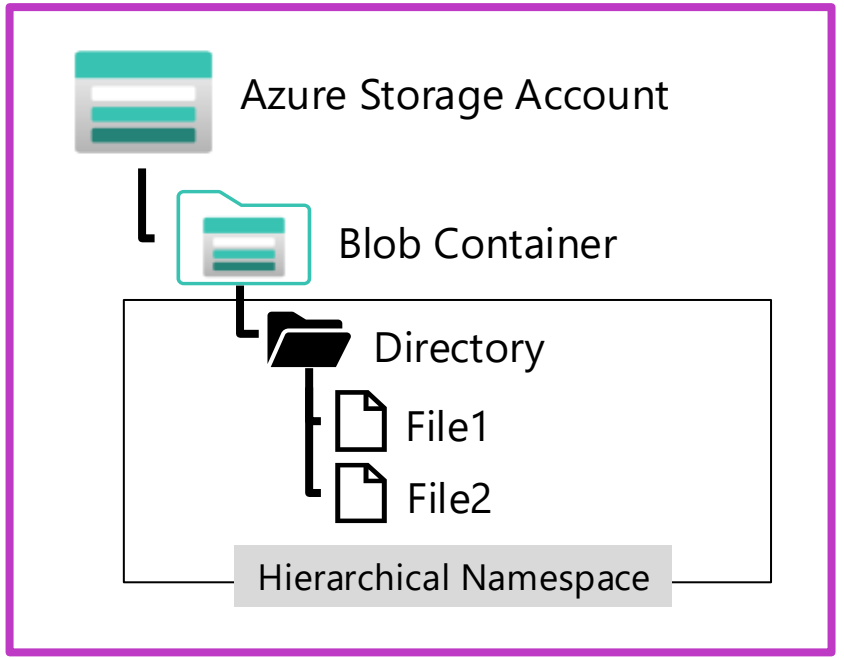
Azure Data Lake Store Gen 2

Distributed file system built on Blob Storage

- Combines Azure Data Lake Store Gen 1 with Azure Blob Storage for large-scale file storage and analytics
- Enables file and directory level access control and management
- Compatible with common large scale analytical systems

Enabled in an Azure Storage account through the *Hierarchical Namespace* option

- Set during account creation
- Upgrade existing storage account
  - One-way upgrade process

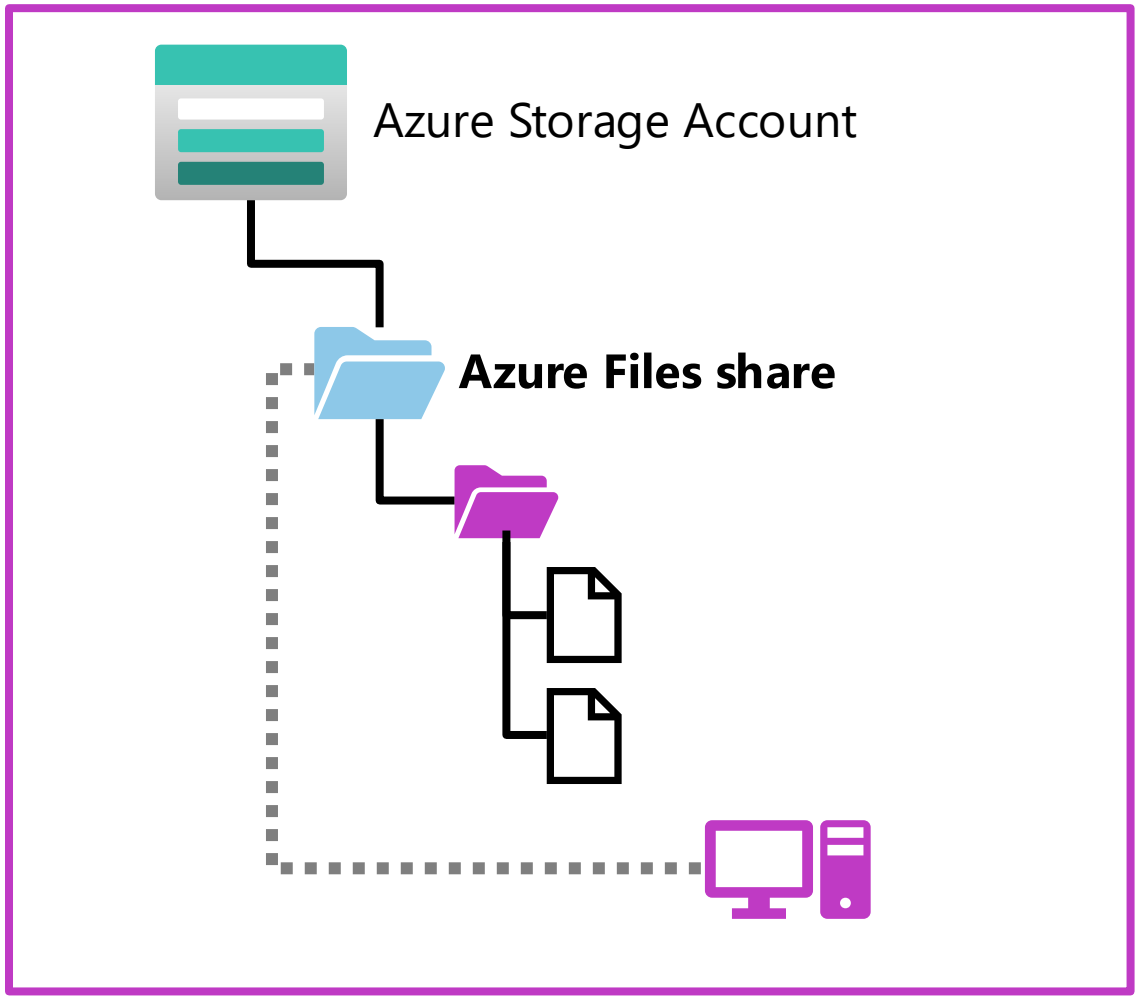


File system includes directories and files, and is compatible with large scale data analytics systems like Databricks

Azure Files

Files shares in the cloud that can be accessed from anywhere with an internet connection

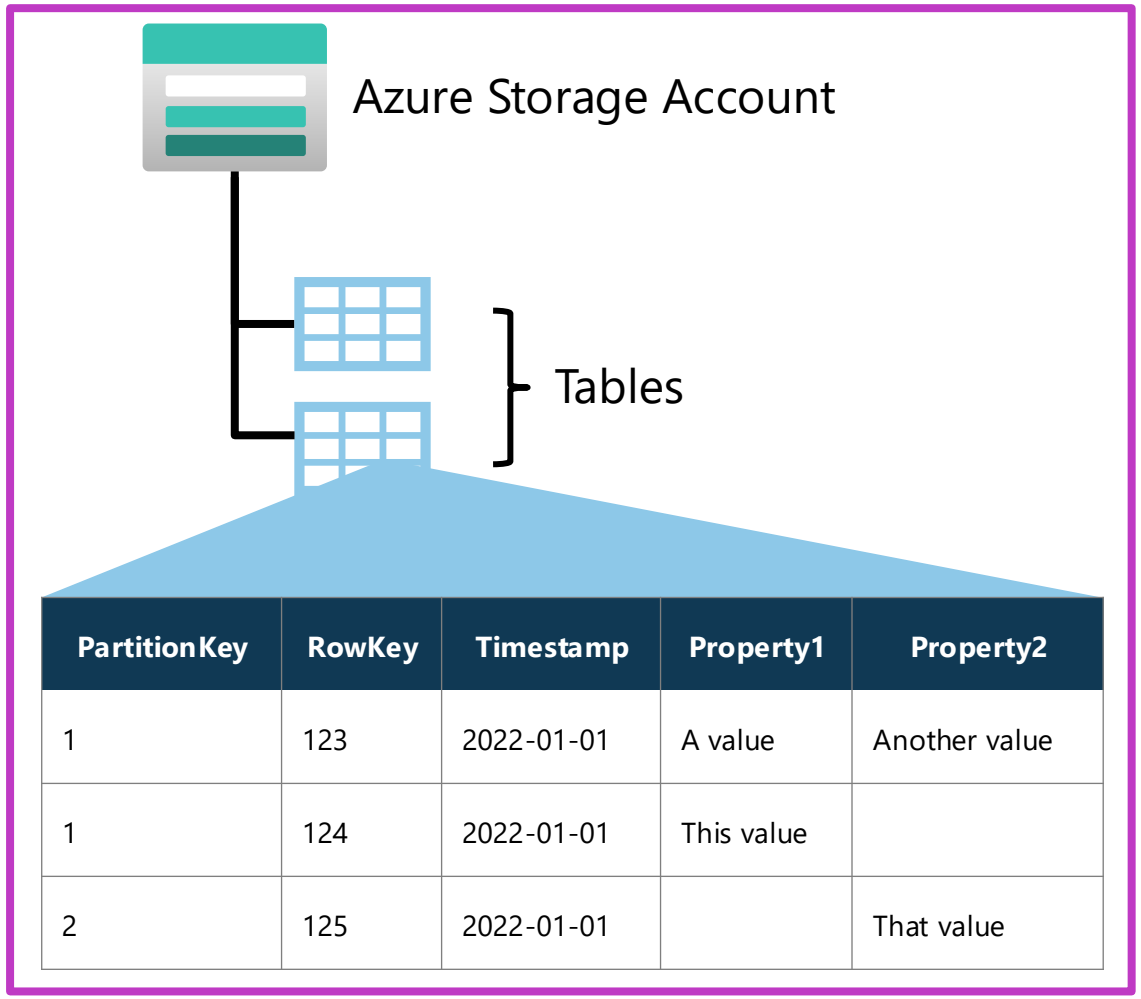
- Support for common file sharing protocols:
  - Server Message Block (SMB)
  - Network File System (NFS) – *requires premium tier*
- Data is replicated for redundancy and encrypted at rest



Azure Table Storage

Key-Value storage for application data

- Tables consist of *key* and *value* columns
  - Partition and row keys
  - Custom property columns for data values
    - A *Timestamp* column is added automatically to log data changes
- Rows are grouped into partitions to improve performance
- Property columns are assigned a data type, and can contain any value of that type
- Rows do not need to include the same property columns



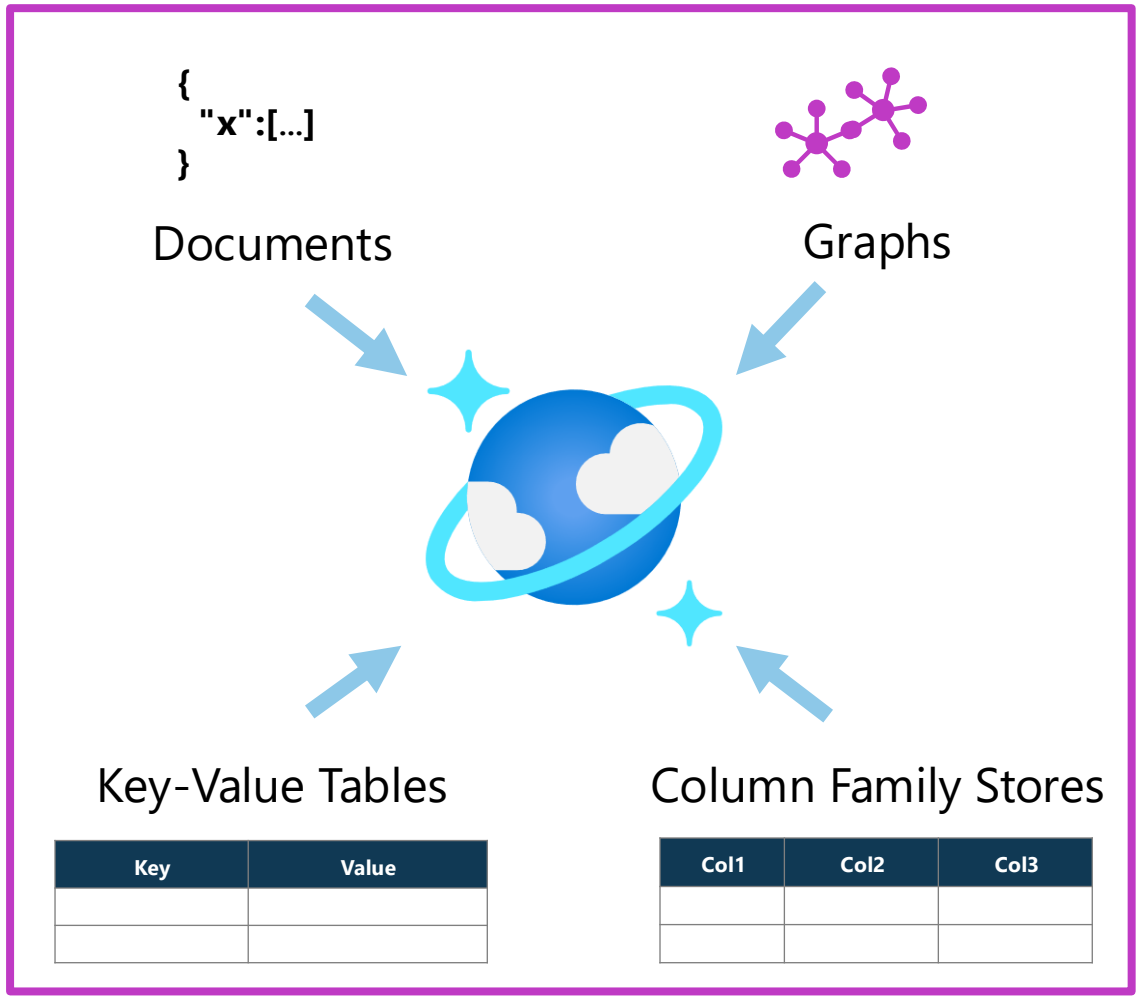
PartitionKey	RowKey	Timestamp	Property1	Property2
1	123	2022-01-01	A value	Another value
1	124	2022-01-01	This value	
2	125	2022-01-01		That value

2: Fundamentals of Azure Cosmos DB

What is Azure Cosmos DB?

A multi-model, global-scale *NoSQL* database management system

- Support for multiple storage APIs
- Real time access with fast read and write performance
- Enable *multi-region writes* to replicate data globally; enabling users in specified regions to work with a local replica



Azure Cosmos DB APIs

Azure Cosmos DB for NoSQL

- Native API for Cosmos DB

```
SELECT *
FROM customers c
WHERE c.id =
"joe@litware.com"
```

```
{
   "id": "joe@litware.com",
   "name": "Joe Jones",
   "address": {
        "street": "1 Main St.",
        "city": "Seattle"
    }
}
```

Azure Cosmos DB for MongoDB

- Compatibility with MongoDB

```
db.products.find({
id: 123})
```

```
{
   "id": 123,
   "name": "Hammer",
   "price": 2.99}
}
```

Azure Cosmos DB for PostgreSQL

- Compatibility with PostgreSQL

id	name	dept	manager
1	Sue Smith	Hardware	Joe Jones
2	Ben Chan	Hardware	Sue Smith

Azure Cosmos DB for Table

- Key-value storage API
- Compatible with Azure Table Storage

PartitionKey	RowKey	Name
1	123	Joe Jones
1	124	Samir Nadoy

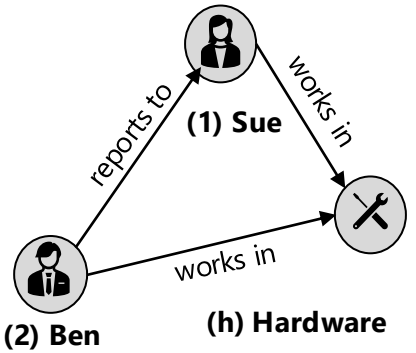
Azure Cosmos DB for Apache Cassandra

- Compatibility with Apache Cassandra

id	name	dept	manager
1	Sue Smith	Hardware	
2	Ben Chan	Hardware	Sue Smith

Azure Cosmos DB for Apache Gremlin

- Used to work with graph data
- vertices are connected via relationships (edges)



Microsoft

4: Explore fundamentals of data analytics

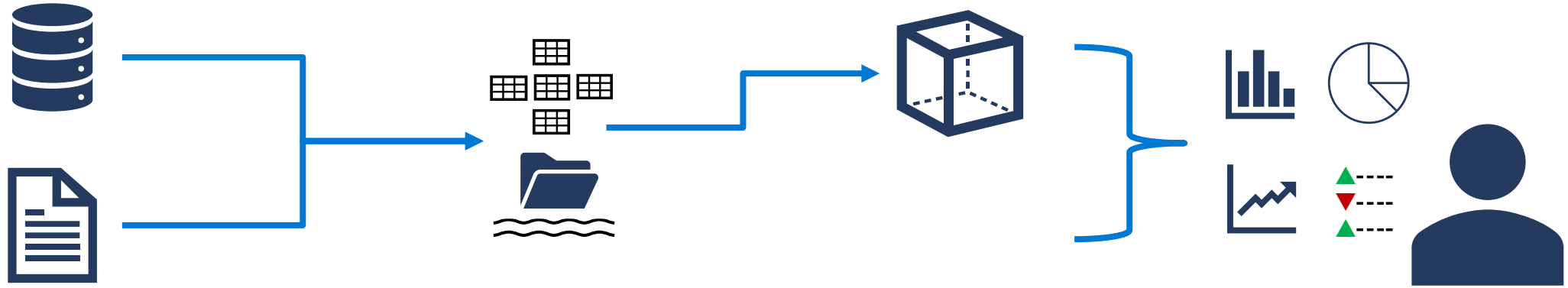
Agenda

- Large-scale data analytics
- Streaming and real-time analytics
- Data visualization

1: Large-scale data analytics

Elements of a large-scale data analytics solution

Data ingestion and processing	Analytical data store	Analytical data model	Data visualization



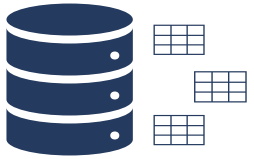
- *Extract, Transform, and Load* (ETL) or *Extract, Load, and Transform* (ELT) orchestration
- Distributed processing to cleanse and restructure data at scale
- Batch and real-time data processing

- Flexible, scalable file storage in a *data lake*
- Relational tables in a *data lakehouse* or *data warehouse*

- Semantic models for analytical entities
- Often in the form of aggregated *cubes* that summarize numeric values across one or more *dimensions*

- Reports
- Charts
- Dashboards

Data processing in large-scale analytics



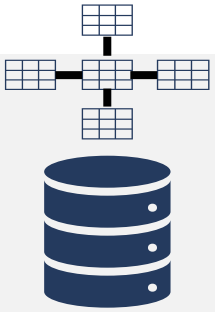
Relational Database

- Well established model for relational data storage and processing
- Comprehensive SQL language support for querying and data manipulation

Apache Spark

- Open-Source platform for scalable, distributed data processing
- Multi-language data processing code (Python, Scala, Java, SQL, …)

Analytical data store architectures



Data Warehouse

- Data is stored in a relational database and queried using a SQL query engine
- Tables are *denormalized* for query optimization
  - Typically as a star or snowflake schema of numeric *facts* that can be aggregated by *dimensions*

Data Lakehouse

- Data files are stored in a distributed file system (a *data lake*) and typically processed using Apache Spark
- Metadata is used to define tables that provide a relational SQL interface to the file data
  - Commonly, a *delta lake* format is used to provide transactional database functionality

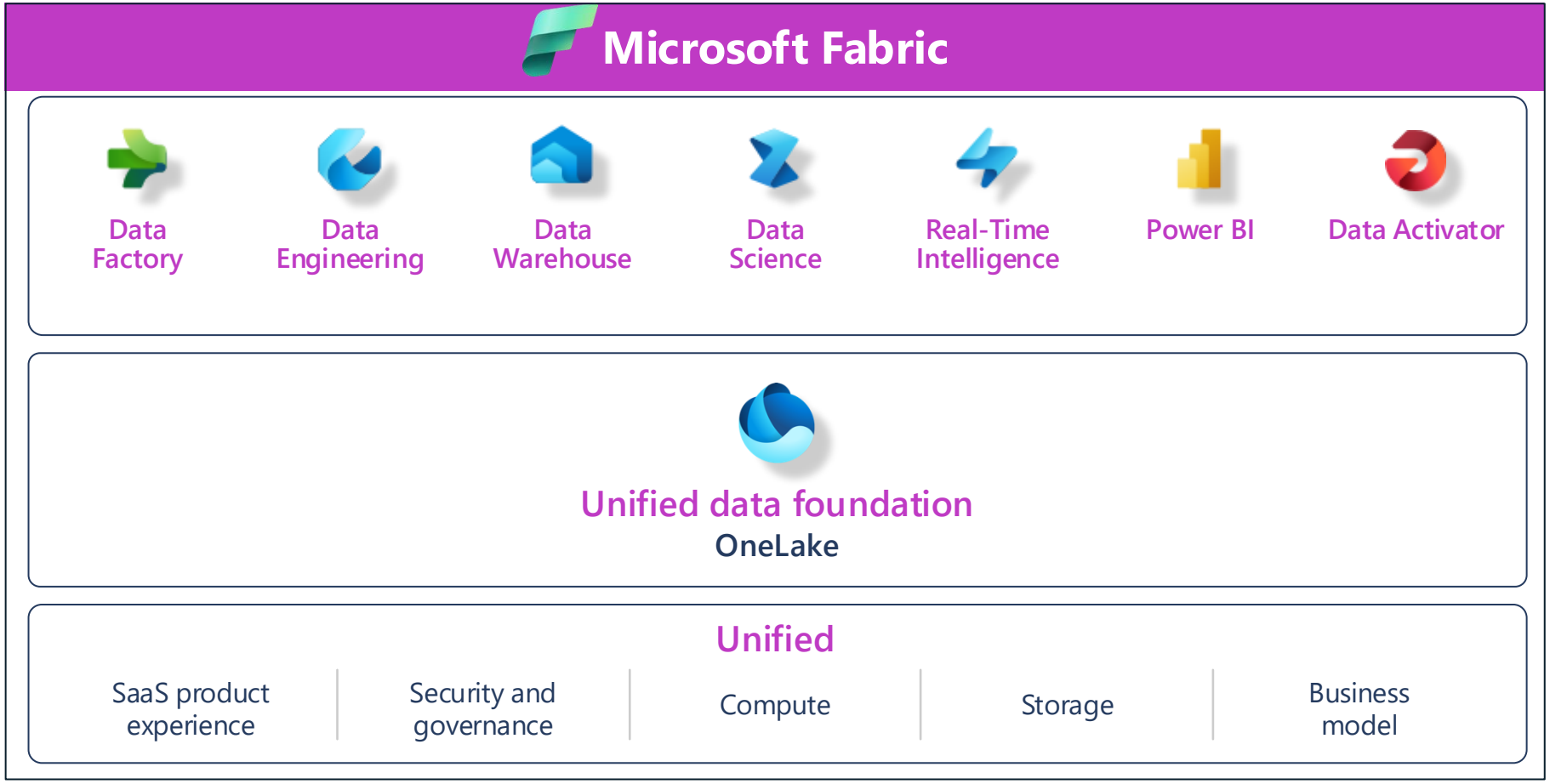
PaaS data analytics with Azure Databricks

Azure Databricks

- Azure-based implementation of Databricks cloud analytics platform
- Scalable Spark and SQL querying for data lake analytics
- Interactive experience in Azure Databricks workspace
- Use Azure Data Factory to implement data ingestion and processing pipelines

Use to leverage Databricks skills and for cloud portability

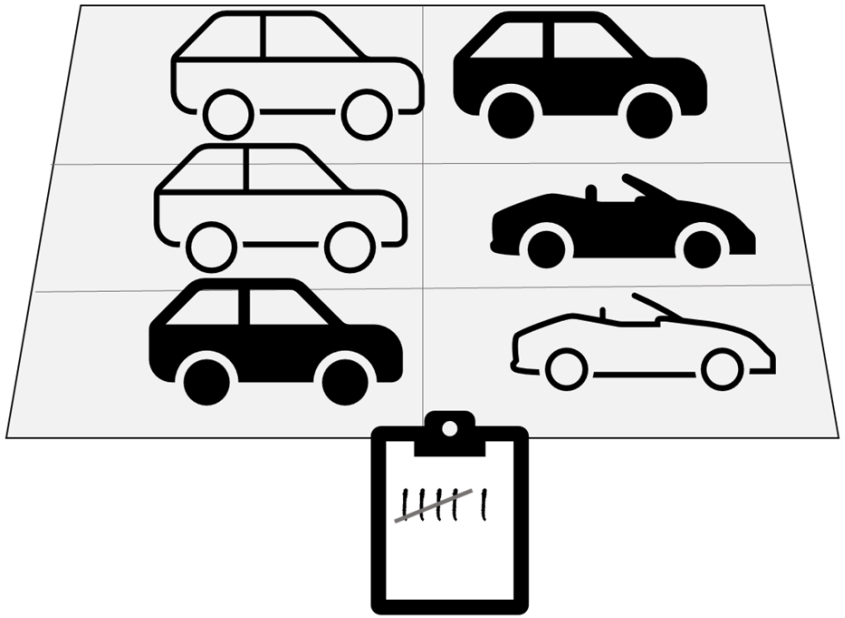
SaaS data analytics with Microsoft Fabric



2: Streaming and real-time analytics

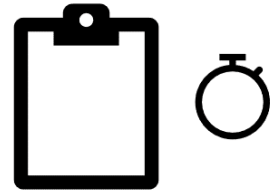
Batch vs stream processing

Batch processing



Data is collected and processed at regular intervals

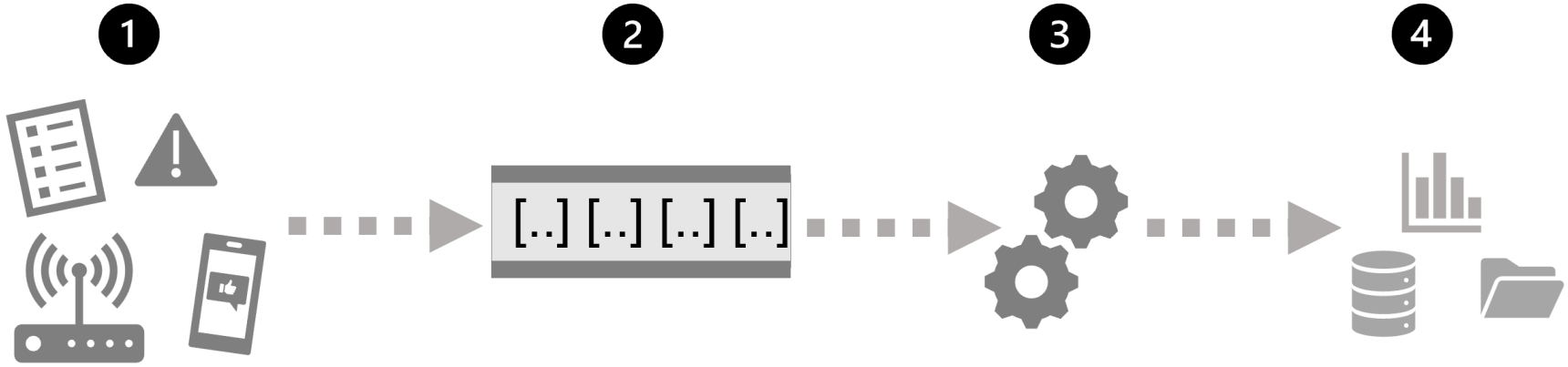
Stream processing



Data is processed in (near) real-time as it arrives

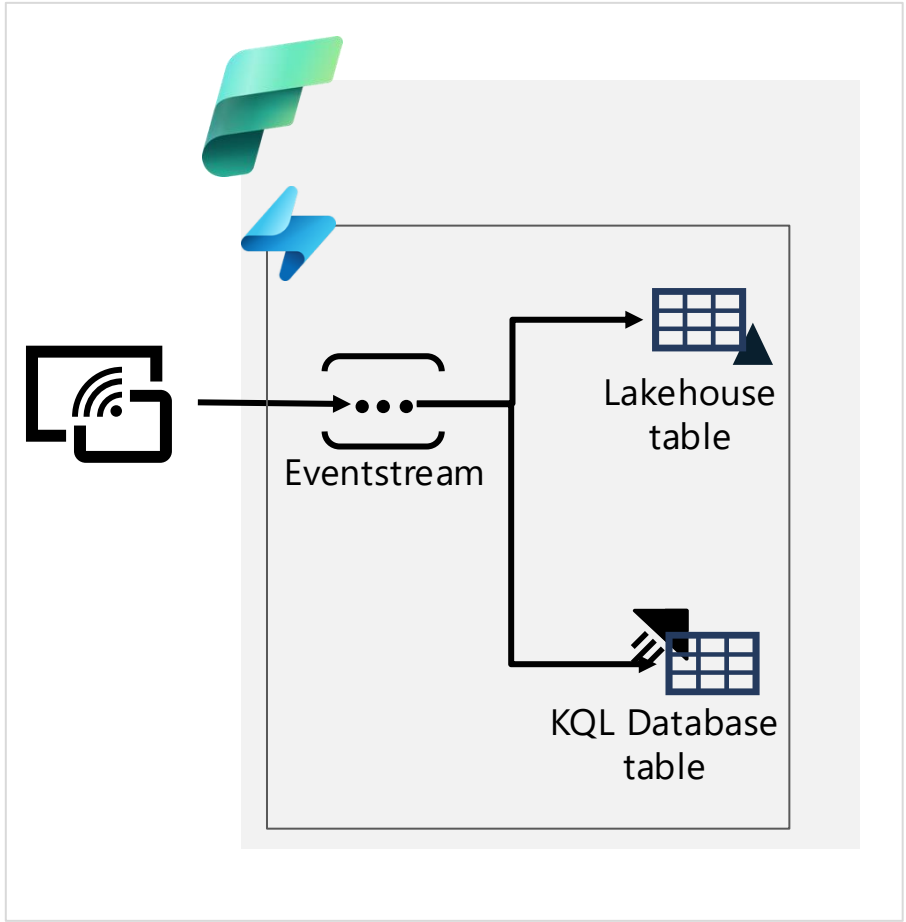
Common elements of stream processing

1. An event generates some data.
2. The generated data is captured in a streaming source for processing.
3. The event data is processed.
4. The results of the stream processing operation are written to an output (or sink).



Real-time analytics in Microsoft Fabric

- Support for continuous data ingestion from multiple sources
- Capture streaming data in an **eventstream**
- Write real-time data to a table in a Lakehouse or a KQL database
- Query real-time data using SQL or KQL
- Build real-time visualizations



Data analytics with Apache Spark

Apache Spark is a distributed processing framework for large scale data analytics. You can use Spark on Microsoft Azure in the following services:

- Microsoft Fabric
- Azure Databricks

Spark Structured Streaming

The Spark Structured Streaming library, which provides an application programming interface (API) for ingesting, processing, and outputting results from perpetual streams of data.

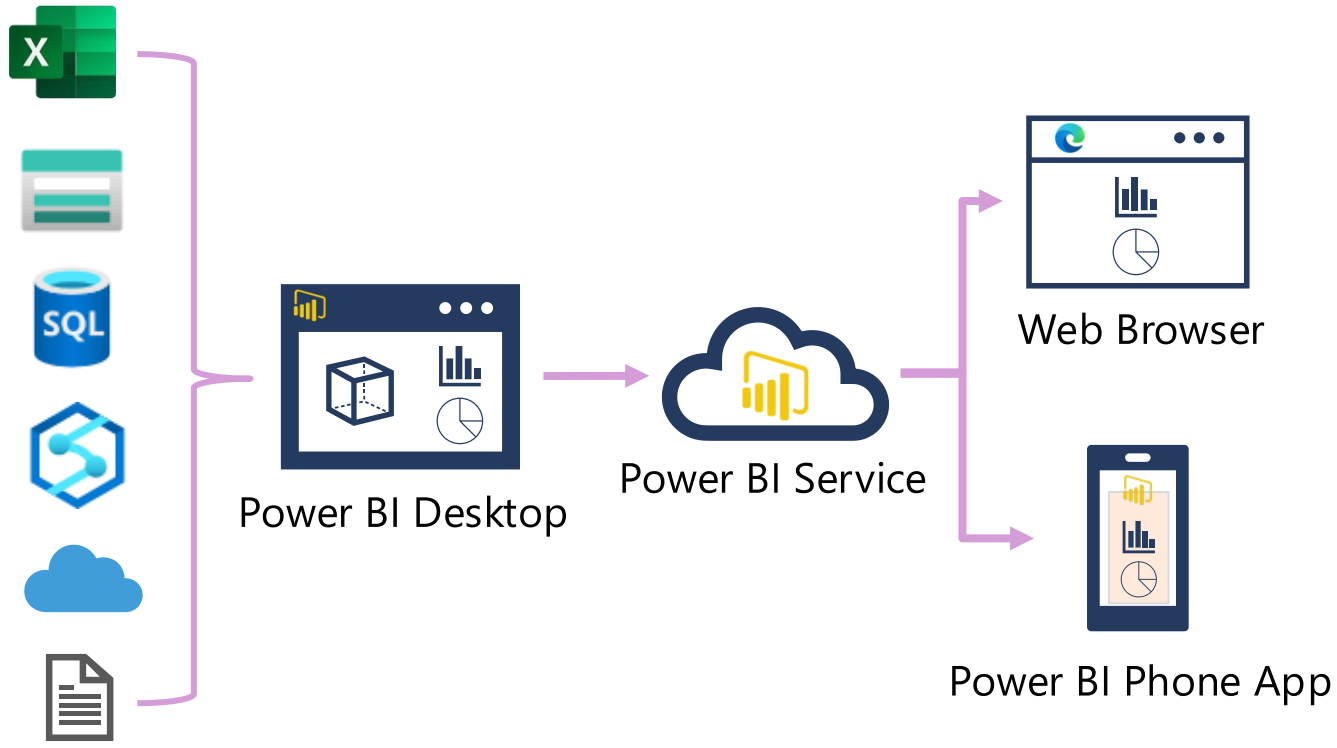
Delta Lake

Delta Lake can be used in Spark to define relational tables for both batch and stream processing.

3: Data visualization

Introduction to data visualization with Power BI

- Start with Power BI Desktop
  - Import data from one or more sources
  - Define a data model
  - Create visualizations in a report
- Publish to Power BI Service
  - Schedule data refresh
  - Create dashboards and apps
  - Share with other users
- Interact with published reports
  - Web browser
  - Power BI phone app



Analytical data modeling

Customer *(dimension)*

Key	Name	Address	City
1	Joe	1 Main St.	Seattle
2	Samir	123 Elm Pl.	New York
3	Alice	2 High St.	Seattle

Product *(dimension)*

Key	Name	Category
1	Hammer	Tools
2	Screwdriver	Tools
3	Wrench	Tools
4	Bolts	Hardware

Sales *(fact)*

Key	TimeKey	ProductKey	CustomerKey	Quantity	Revenue
1	01012022	1	1	1	2.99
2	01012022	2	1	2	6.98
3	02012022	1	2	2	5.98

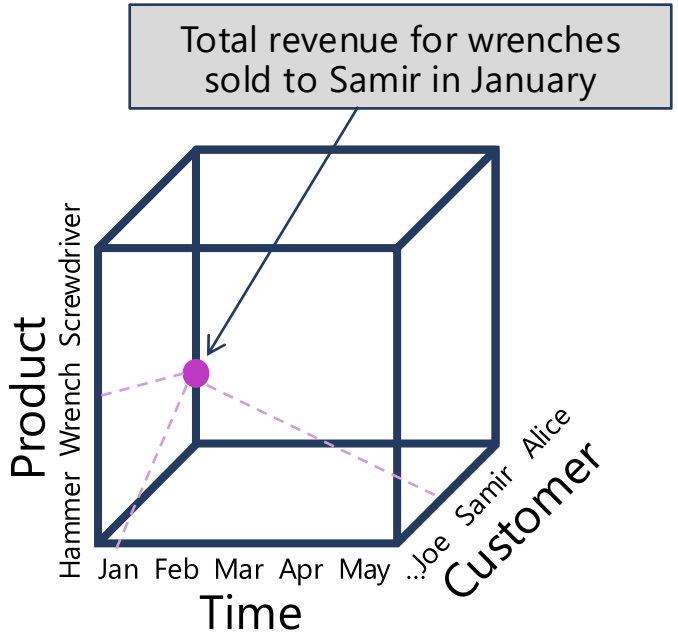
Measures

Time *(dimension)*

Key	Year	Month	Day	WeekDay
01012022	2022	Jan	1	Sat
02012022	2022	Jan	2	Sun

Hierarchy

Σ

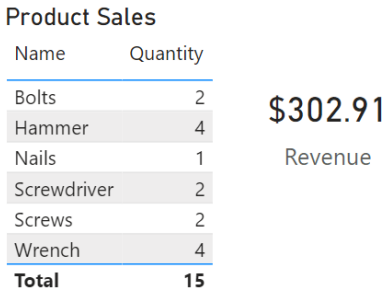


Model aggregates measures at each hierarchy level

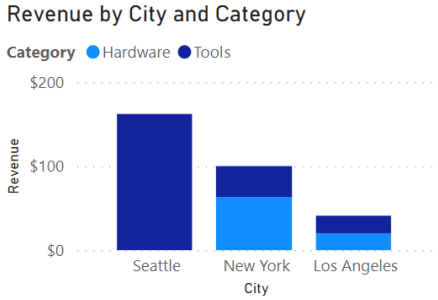
Year	Month	Day	Revenue
2022			8221.48
	Jan		574.86
		1	9.97
		2	5.98
		…	…

Common data visualizations in reports

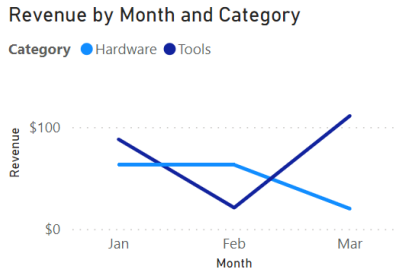
Tables and text



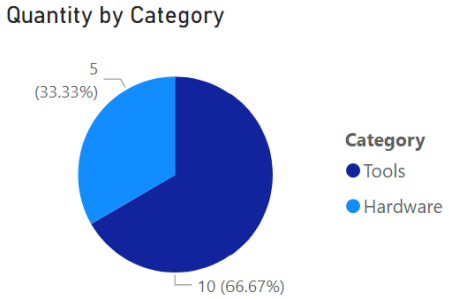
Bar or column chart



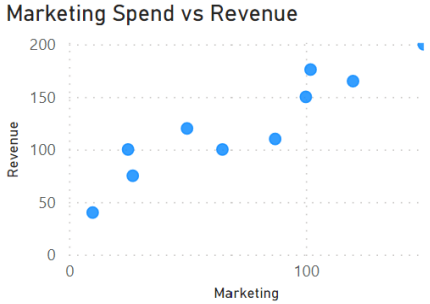
Line chart



Pie chart



Scatter plot



Map

Q&A Session