Microsoft

Microsoft Azure Virtual Training Day: Data Fundamentals

We will be starting shortly

About this course

Course objectives:

- Describe core data concepts
- Identify services for relational data
- Identify services for non-relational data
- Identify services for data analytics

This course is supplemented by online training at https://aka.ms/AzureLearn_DataFundamentals

Course Agenda

Module 1: Explore fundamentals of data

- Core data concepts
- Data roles and services

Module 2: Explore fundamentals of relational data in Azure

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

Module 3: Explore fundamentals of non-relational data in Azure

- Fundamentals of Azure Storage
- Fundamentals of Azure Cosmos DB

Module 4: Explore fundamentals of large-scale data warehousing

- Large-scale data warehousing

Module 5: Explore fundamentals of real-time analytics

- Streaming and real-time analytics

Module 6: Explore fundamentals of data visualization

- Data visualization

Demos

- Demos in this course are based on exercises in Microsoft Learn

Module 1:

Explore Fundamentals of Data

- Lesson 1: Core data concepts
- Lesson 2: Data roles and services

Lesson 1: Core Data Concepts

What is data?

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

Structured

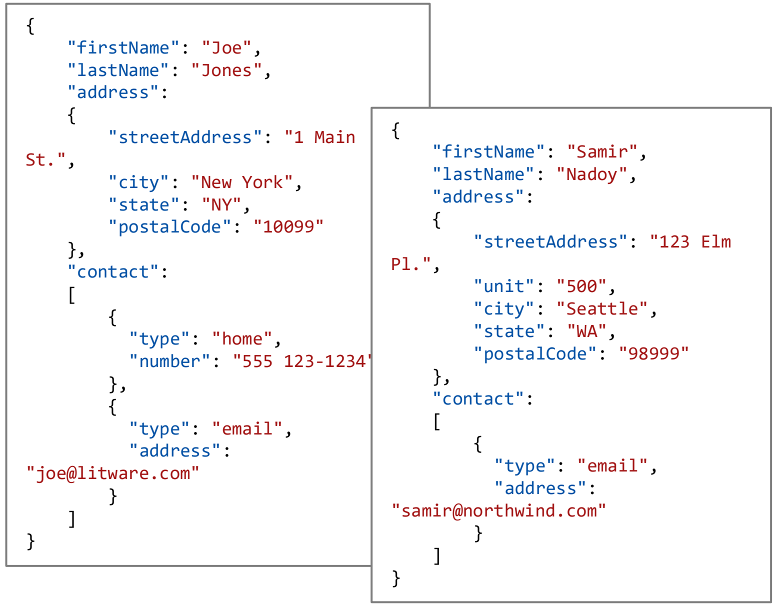
Customer

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

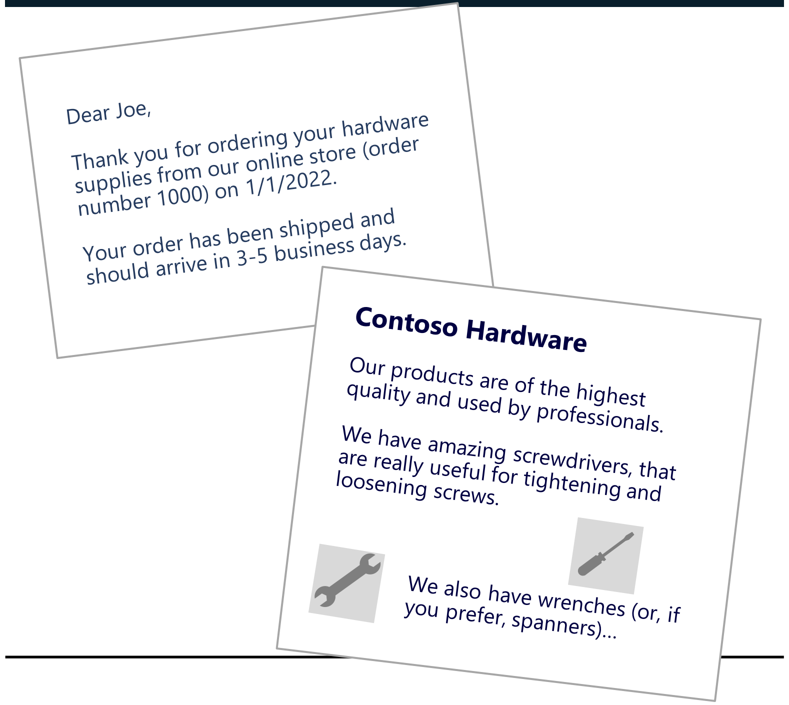
Semi-structured



```
{
    "firstName": "Joe",
    "lastName": "Jones",
    "address":
    {
        "streetAddress": "1 Main
St.",
        "city": "New York",
        "state": "NY",
        "postalCode": "10099"
    },
    "contact":
    [
        {
          "type": "home",
          "number": "555 123-1234"
        },
        {
          "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



Dear Joe,

Thank you for ordering your hardware supplies from our online store (order number 1000) on 1/1/2022.

Your order has been shipped and should arrive in 3-5 business days.

Contoso Hardware

Our products are of the highest quality and used by professionals.

We have amazing screwdrivers, that are really useful for tightening and loosening screws.

We also have wrenches (or, if you prefer, spanners)...

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)

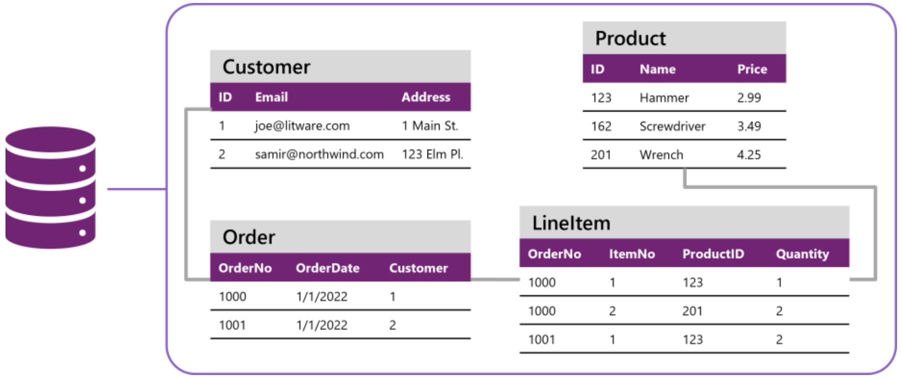
```
10110101101010110010...
```

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

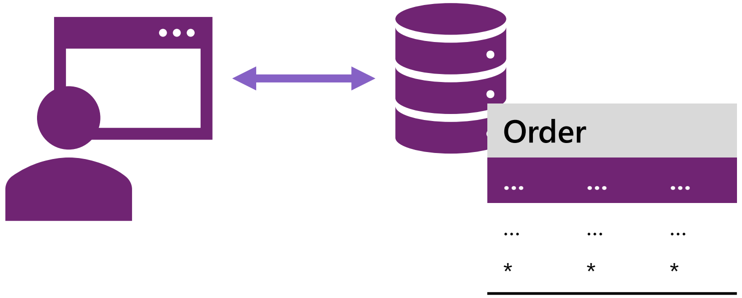
Transactional data workloads

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

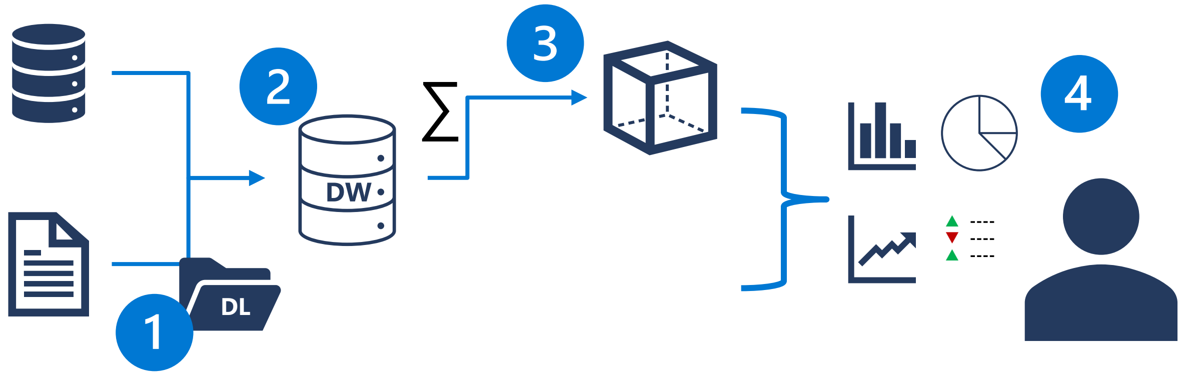


Data is stored using *transactions*

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. Data files may be stored in a central *data lake* for analysis
2. An extract, transform, and load (ETL) process copies data from files and OLTP databases into a *data warehouse* that is optimized for *read* activity
3. Data in the data warehouse may be aggregated and loaded into an online analytical processing (OLAP) model, or *cube*
4. The data in the data lake, data warehouse, and analytical model can be queried to produce reports and dashboards

Lesson 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

Data stores

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

Data engineering and analytics

Azure Data Factory
- Data pipelines

Azure Synapse Analytics
- Integrated, end-to-end analytics
- Pipelines, SQL, Apache Spark, Data Explorer …

Azure Databricks
- Apache Spark analytics and data processing

Azure HDInsight
- Apache open-source platform

Azure Stream Analytics
- Real-time data processing for IoT solutions

Azure Data Explorer
- Real-time data analysis for logs and telemetry

Microsoft Purview
- Enterprise data governance
- Data mapping and discoverability

Microsoft Power BI
- Analytical data modeling
- Interactive data visualization

others…

Module 2:

Explore Fundamentals of Relational Data in Azure

- Lesson 1: Explore relational data concepts
- Lesson 2: Explore Azure services for relational data

Lesson 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

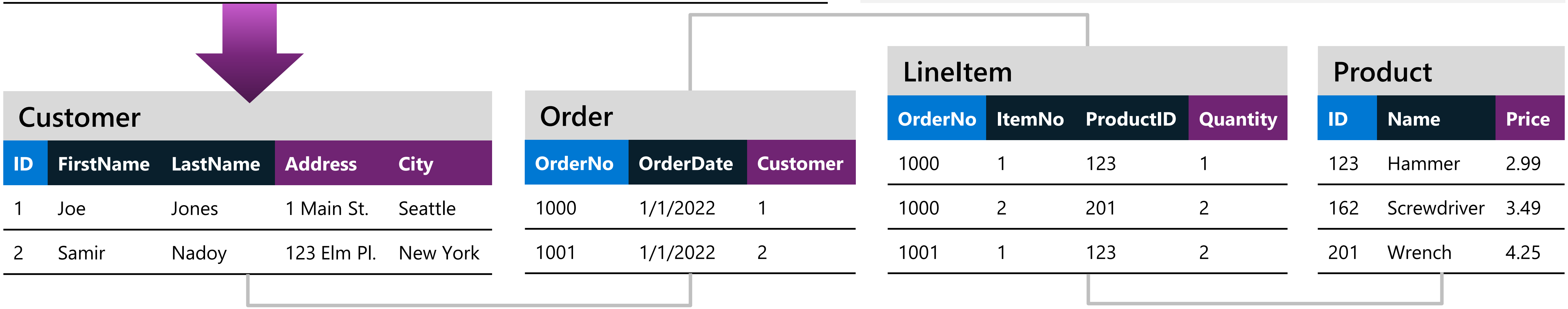
OrderNo	ItemNo	ProductID	Quantity
1000	1	123	1
1000	2	201	2
1001	1	123	2

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

LineItem

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

Product

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

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)

CREATE, ALTER, DROP, RENAME

```
CREATE TABLE Product
(
  ProductID INT PRIMARY KEY,
  Name VARCHAR(20) NOT NULL,
  Price DECIMAL NULL
);
```

Product

ID	Name	Price

Data Control Language (DCL)

GRANT, DENY, REVOKE

```
GRANT SELECT, INSERT, UPDATE
ON Product
TO user1;
```

Product

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

Data Manipulation Language (DML)

INSERT, UPDATE, DELETE, SELECT

```
SELECT Name, Price
FROM Product
WHERE Price > 2.50
ORDER BY Price;
```

Results

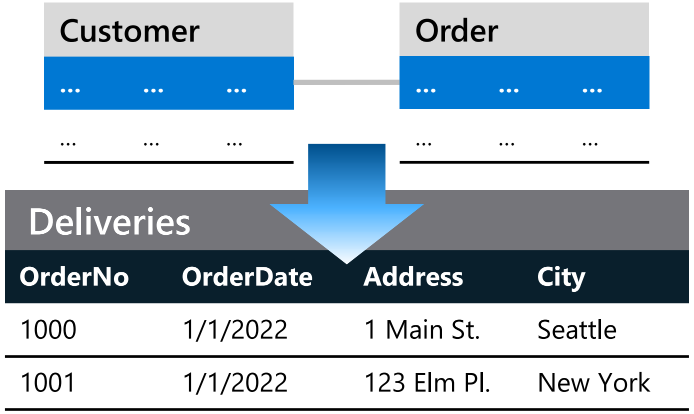
Name	Price
Hammer	2.99
Screwdriver	3.49
Wrench	4.25

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';
```

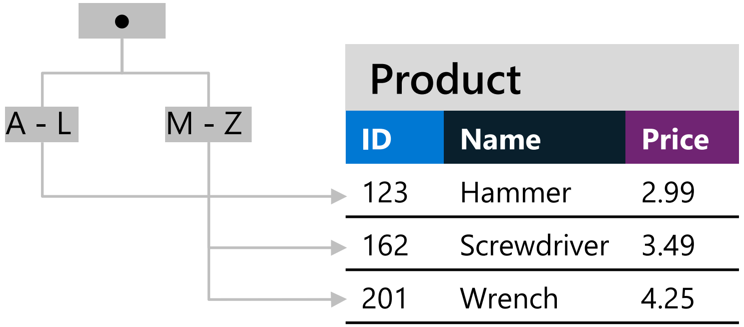
Product

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	Wrench	4.25

Lesson 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

IaaS

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

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

Demo

Provision Azure relational database services

Module 3:

Explore Fundamentals of Non-relational Data in Azure

- Lesson 1: Fundamentals of Azure Storage
- Lesson 2: Fundamentals of Azure Cosmos DB

Lesson 1: Fundamentals of Azure Storage

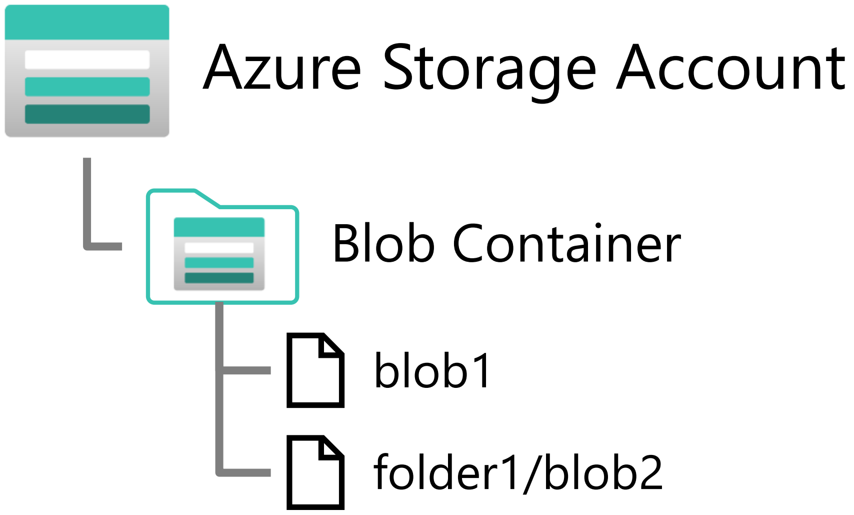
Azure Blob Storage

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

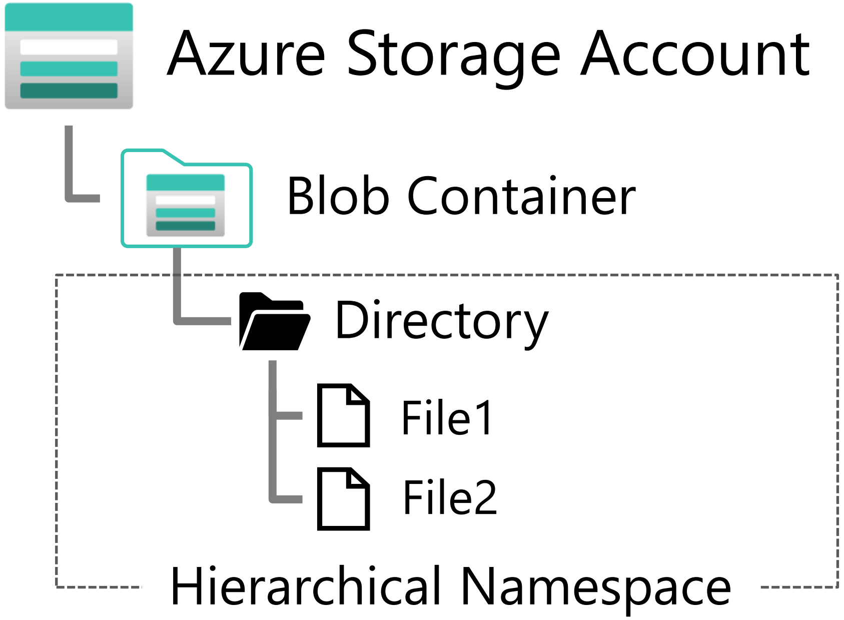
Azure Data Lake Store Gen 2

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

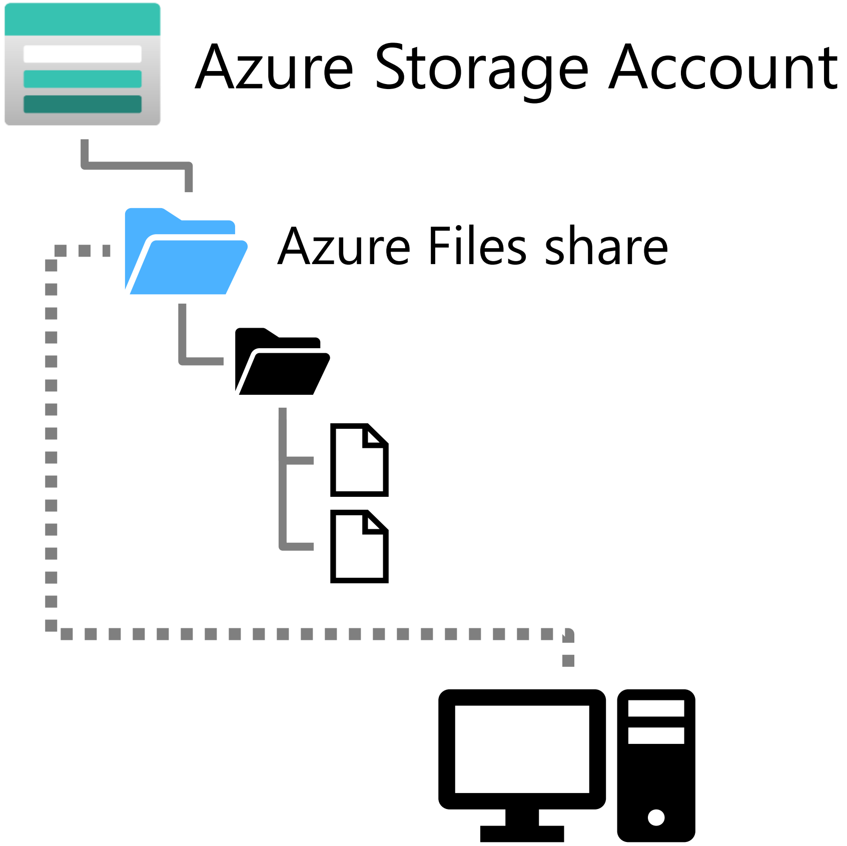


File system includes directories and files, and is compatible with large scale data analytics systems like Hadoop, Databricks, and Azure Synapse Analytics

Azure Files

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

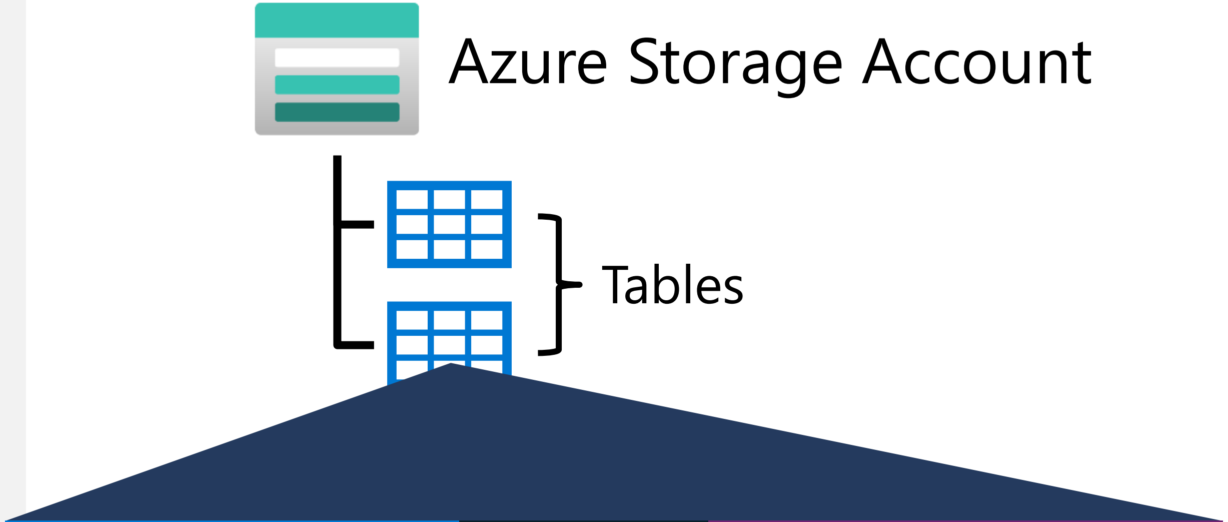
- Support for common file sharing protocols:
  - Server Message Block (SMB)
  - Network File System (NFS) – *requires premium tier*
- Data is replicated for redundancy and encrypted at rest



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	2022/1/1	A value	Another value
1	124	2022/1/1	This value	
2	125	2022/1/1		That value

Demo

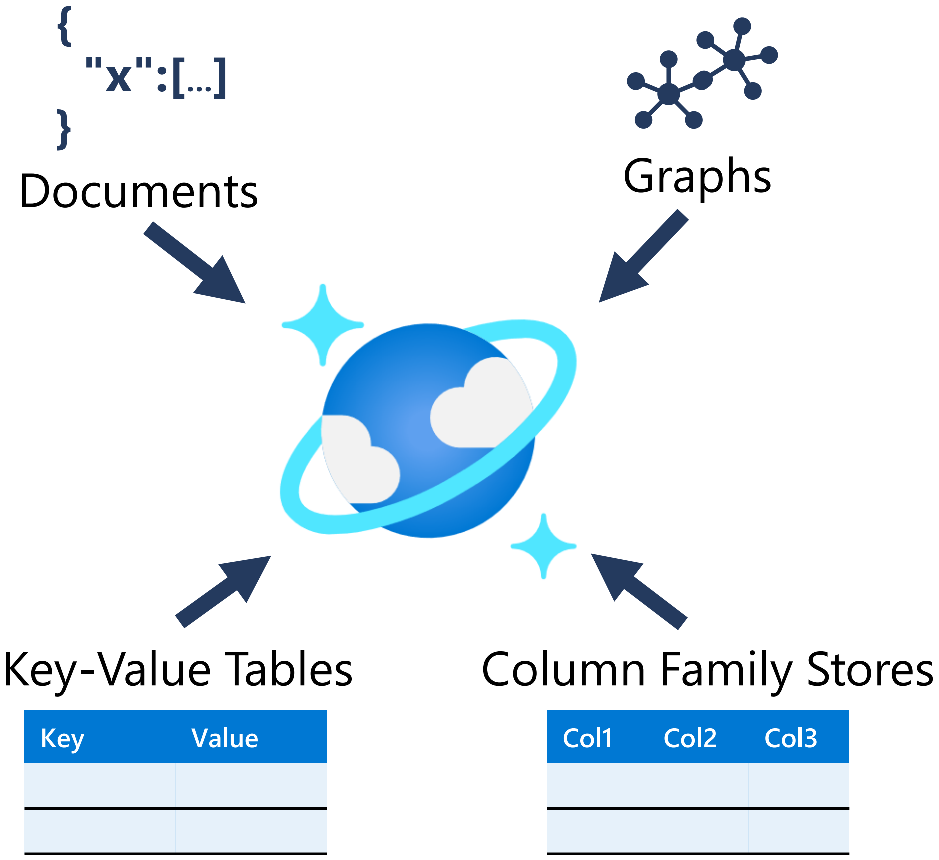
Explore Azure Storage

Lesson 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



```
{
  "x":[…]
}
```

Documents

Graphs

Key-Value Tables

Key	Value

Column Family Stores

Col1	Col2	Col3

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

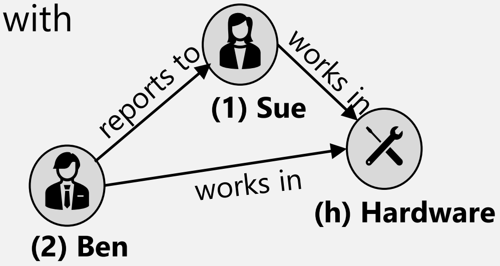
Azure Cosmos DB for Apache Cassandra

- 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*)



Demo

Explore Azure Cosmos DB

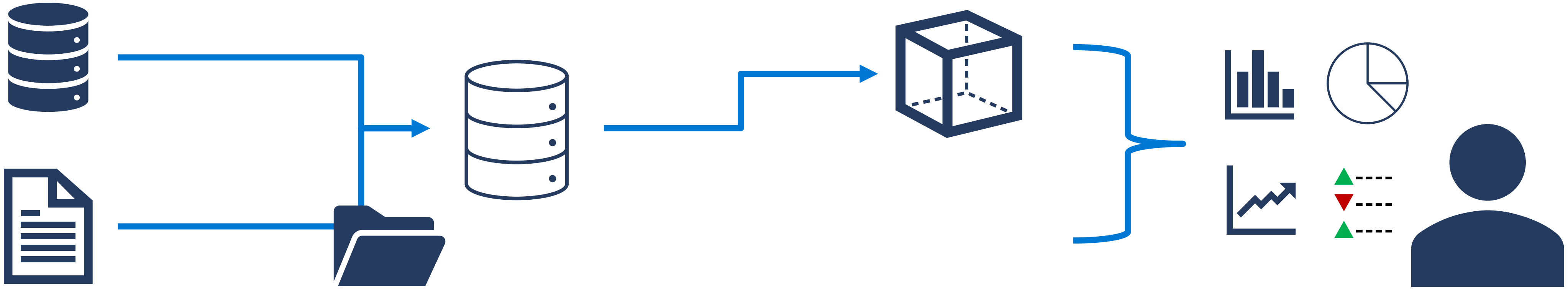
Module 4:

Explore Fundamentals of Large-scale data warehousing

- Lesson 1: Large-scale data warehousing

Lesson 1: Large-scale data warehousing

What is large-scale data warehousing?



Data ingestion and processing

- *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

Analytical data store

- Denormalized relational data storage in a *data warehouse*
- Semi-structured file storage in a *data lake*

Analytical data model

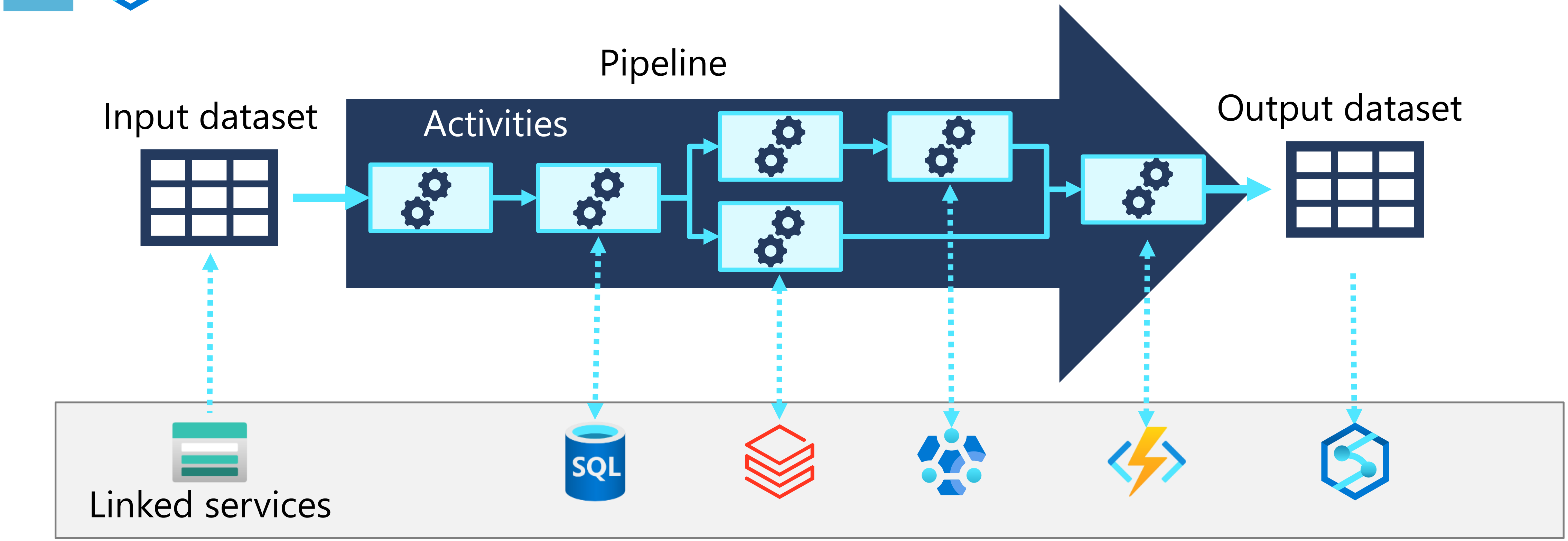
- Semantic models for analytical entities
- Often in the form of aggregated *cubes* that summarize numeric values across one or more *dimensions*

Data visualization

- Reports
- Charts
- Dashboards

Data ingestion and processing pipelines

Create pipelines in **Azure Data Factory** or **Azure Synapse Analytics**



Analytical data stores

Data Warehouse

- Large-scale relational database store and query engine
- Data is *denormalized* for query optimization
  - Typically as a *star* or *snowflake* schema of numeric *facts* that can be aggregated by *dimensions*

Data Lake

- Data files are stored in a distributed file system
- Tabular storage layers can be used to abstract files and provide a relational interface.
  - Use *PolyBase* external tables or create a *lake database* in Azure Synapse Analytics
  - Use database tables and SQL endpoints in Azure Databricks
  - Use Spark *Delta Lake* to add relational storage semantics and create a *data lakehouse* in Azure Synapse Analytics, Azure Databricks, and Azure HDInsight

Choose an analytical data store service

Azure Synapse Analytics

- Unified solution for relational data warehouse and data lake analytics
- Scalable processing and querying through multiple analytics runtimes
  - Synapse SQL
  - Apache Spark
  - Synapse Data Explorer
- Interactive experience in Azure Synapse Studio
- Built-in pipeline integration for data ingestion and processing

Use for a single, unified large-scale analytical solution on Azure

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

Azure HDInsight

- Azure-based implementation of common Apache "big data" frameworks built on a data lake
  - Hadoop - Query data lake files using Hive tables
  - Spark – Use Spark APIs to query data, and abstract underlying file storage as tables
  - Kafka – Real-time event processing
  - Storm – Stream processing
  - HBase – NoSQL data store

Use when you need to support multiple open-source platforms

Demo

Explore Azure Synapse Analytics

Module 5:

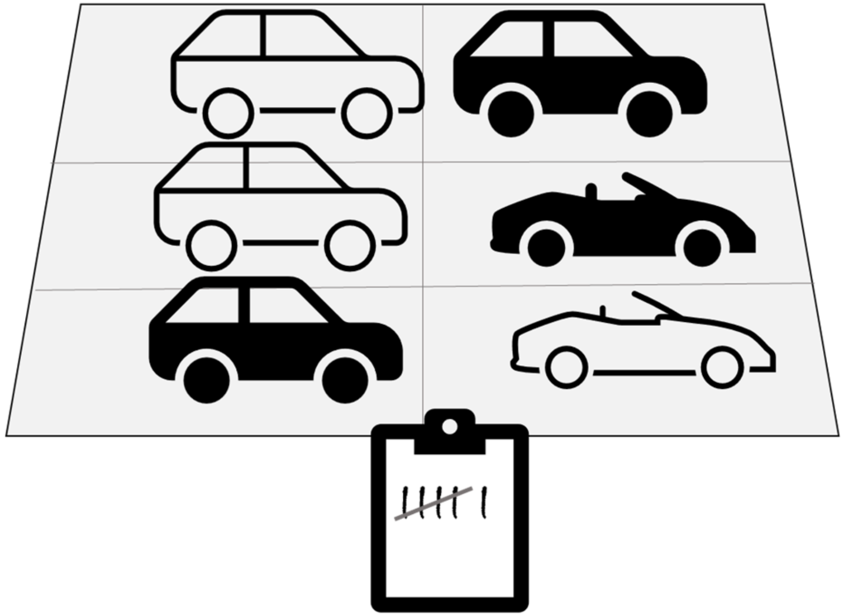
Explore Fundamentals of real-time analytics

- Lesson 1: Streaming and real-time analytics

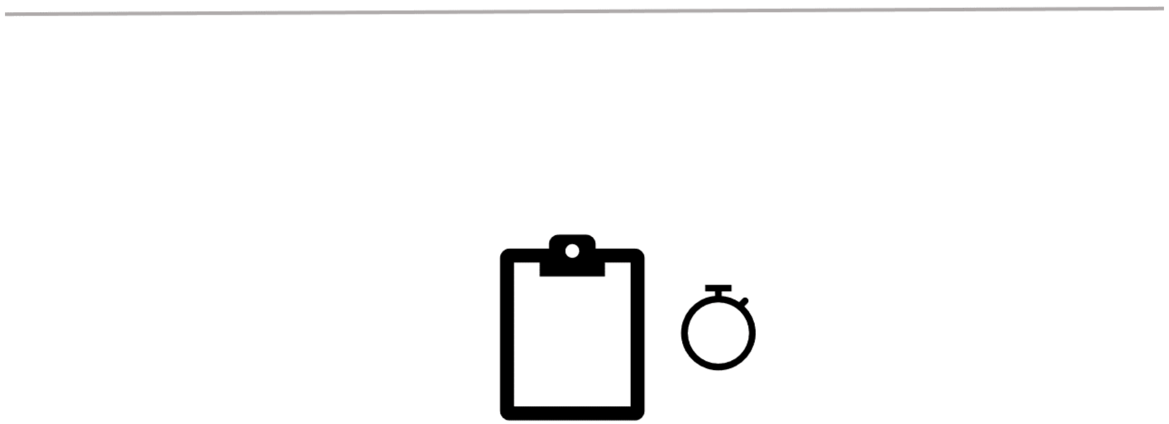
Lesson 1: Streaming and Real-time Analytics

Batch vs stream processing

Batch processing	Stream processing



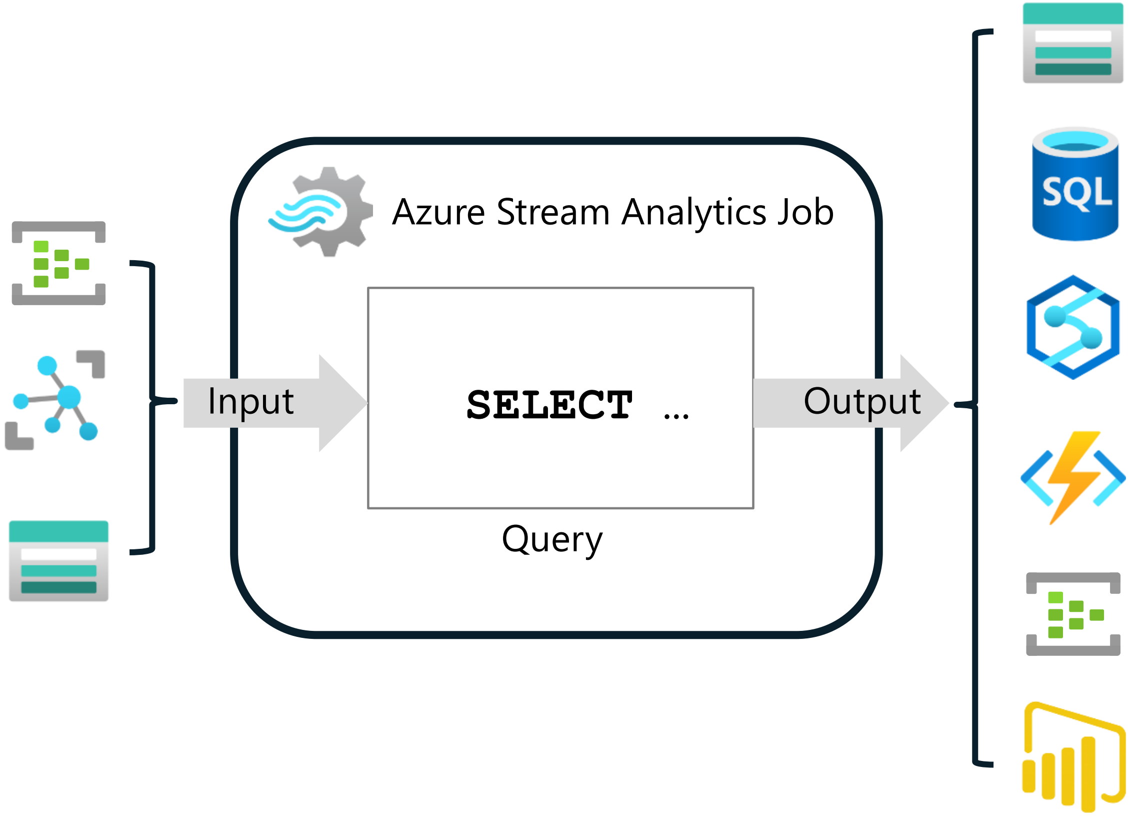
Data is collected and processed at regular intervals



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

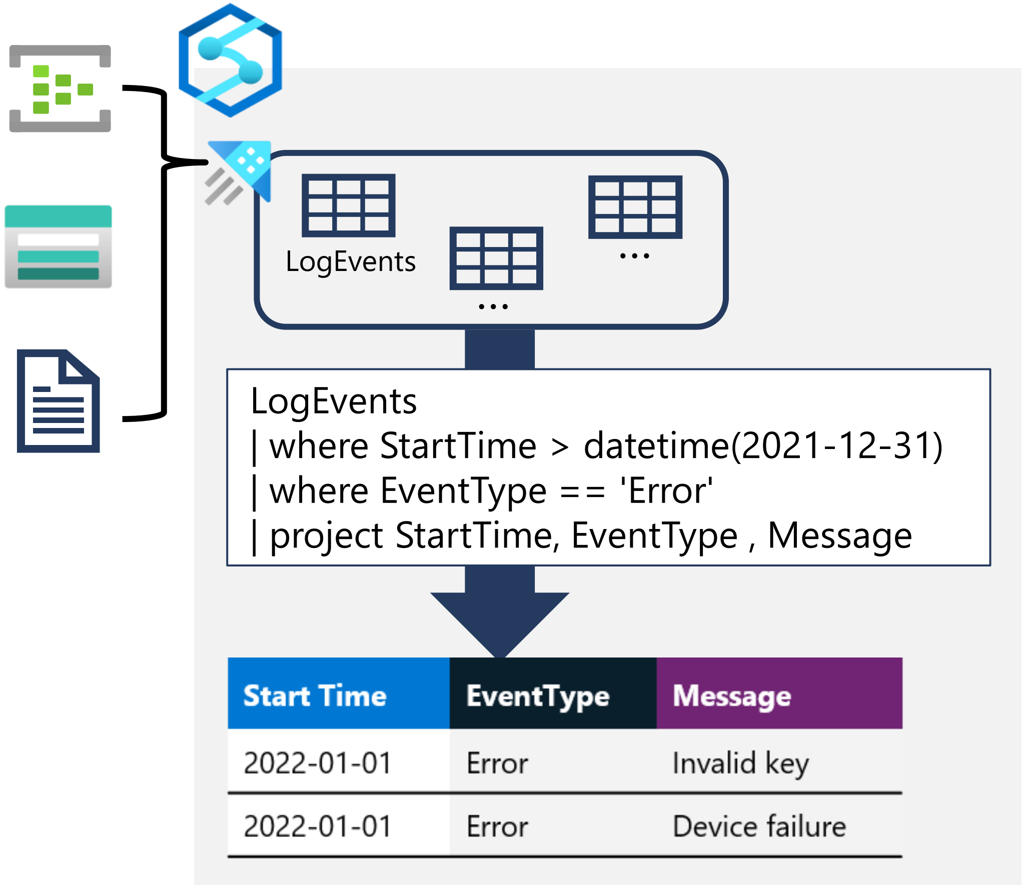
Real-time data processing with Azure Stream Analytics

- Create an individual Azure Stream Analytics *job* or an Azure Stream Analytics *cluster*
- Ingest data from an *input*, such as:
  - Azure Event Hubs
  - Azure IoT Hub
  - Azure Blob Storage
  - …
- Process data with a perpetual *query*
- Send results to an *output*, such as:
  - Azure Blob Storage
  - Azure SQL Database
  - Azure Synapse Analytics
  - Azure Function
  - Azure Event Hubs
  - Power BI
  - …



Real-time log and telemetry analysis with Azure Data Explorer

- High throughput, scalable service for batch and streaming data
  - **Azure Data Explorer** dedicated service
  - **Azure Synapse Data Explorer** runtime in Azure Synapse Analytics
- Data is ingested from streaming and batch sources into tables in a database
- Tables can be queried using *Kusto Query Language* (KQL):
  - Intuitive syntax for read-only queries
  - Optimized for raw telemetry and time-series data



```
LogEvents
| where StartTime > datetime(2021-12-31)
| where EventType == 'Error'
| project StartTime, EventType , Message
```

Start Time	EventType	Message
2022-01-01	Error	Invalid key
2022-01-01	Error	Device failure

Demo

Explore Azure Stream Analytics

Module 6:

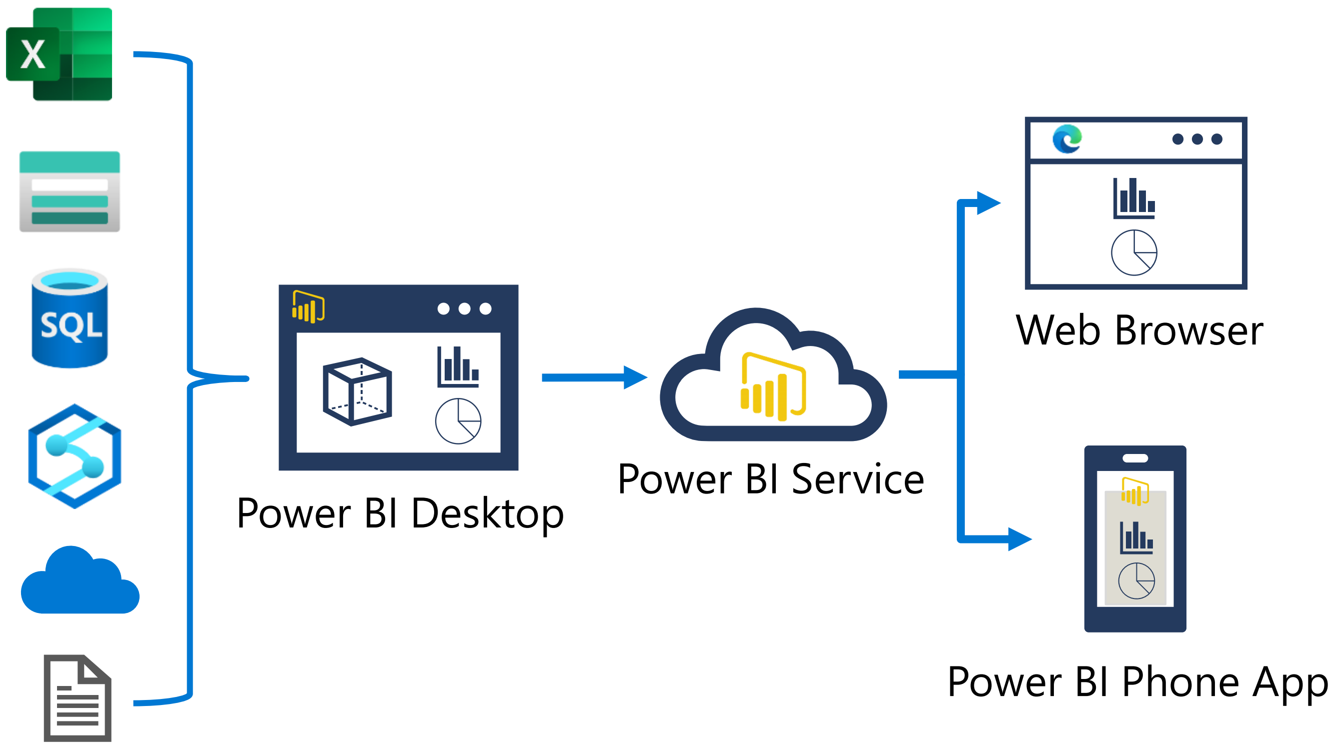
Explore Fundamentals of Data Visualization

- Lesson 1: Data visualization

Lesson 1: 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

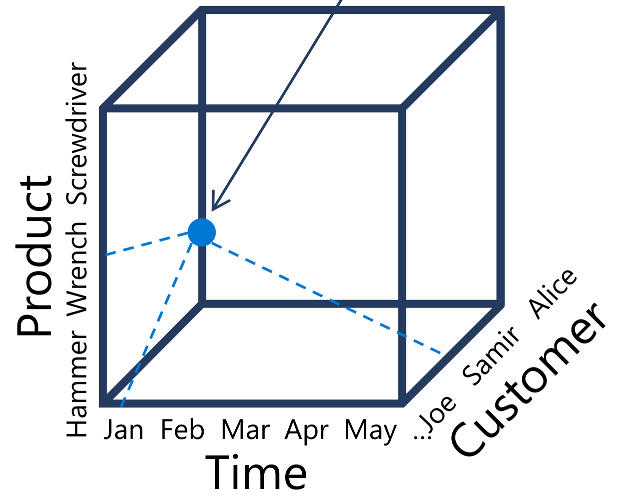
Time *(dimension)*

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

Hierarchy

∑

Total revenue for wrenches sold to Samir in January

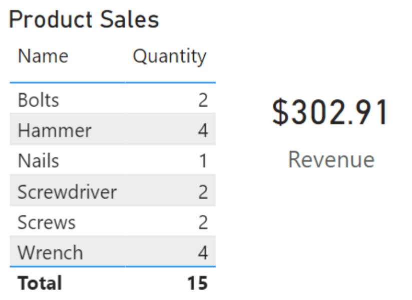


Model aggregates measures at each hierarchy level

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

Common data visualizations in reports

Tables and text



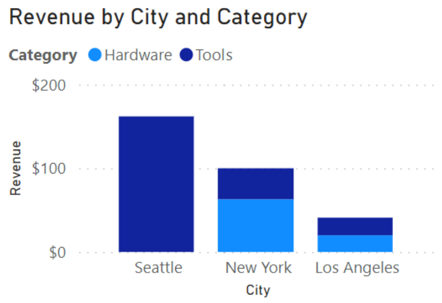
Product Sales

Name	Quantity
Bolts	2
Hammer	4
Nails	1
Screwdriver	2
Screws	2
Wrench	4
Total	**15**

$302.91

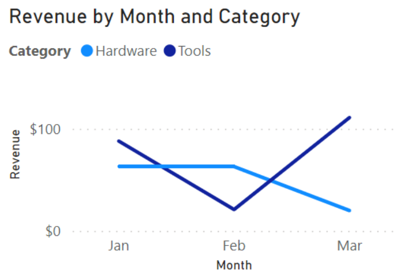
Revenue

Bar or column chart



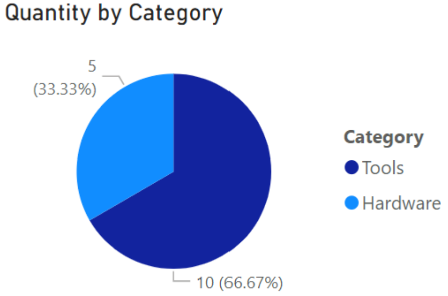
Revenue by City and Category

Line chart



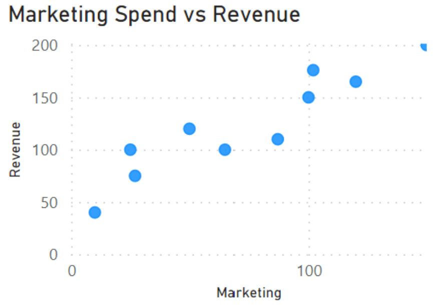
Revenue by Month and Category

Pie chart



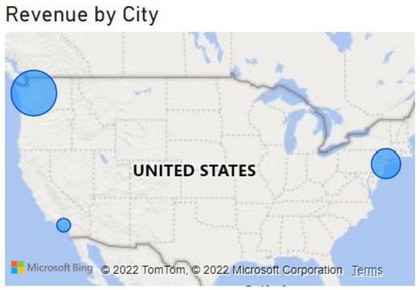
Quantity by Category

Scatter plot



Marketing Spend vs Revenue

Map



Revenue by City

Demo

Visualize data with Power BI

Further learning

To review what you've learned and do additional labs, review the Microsoft Learn modules for this course:

- Explore core data concepts https://aka.ms/ExploreDataConcepts
- Explore relational data in Azure https://aka.ms/ExploreRelationalData
- Explore non-relational data in Azure https://aka.ms/ExploreNonRelationalData
- Explore data analytics in Azure https://aka.ms/ExploreDataAnalytics

Microsoft

Thank you