



github.com/yashbhangale

yashbhangale.github.io

www.linkedin.com/in/yashbhangale

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": "10000"
  },
  "contact": [
    {
      "type": "home",
      "number": "555-123-1234"
    },
    {
      "type": "email",
      "address": "joe@litware.com"
    }
  ]
}

{
  "firstName": "Samir",
  "lastName": "Nadoy",
  "address": {
    "streetAddress": "123 Elm Pl.",
    "unit": "Apt 2",
    "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 the 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



Non-relational

Customer			Product		
ID	Email	Address	ID	Name	Price
1	joe@litware.com	1 Main St.	123	Hammer	2.99
2	samir@northwind.com	123 Elm Pl.	162	Screwdriver	3.49
			201	Wrench	4.25

Order			LineItem		
OrderRef	OrderDate	Customer	OrderRef	ItemRef	ProductID
1000	6/1/2002	1	1000	1	123
1000	6/1/2002	1	1000	2	201
1001	6/1/2002	2	1001	1	123

Products			Customer		
Key	Value	Customer	Key	Customer	Address
123	"Hammer (\$2.99)"		1000	Joe Jones	1 Main St.
142	"Screwdriver (\$3.49)"		1001	Samir Nadoy	123 Elm Pl.
201	"Wrench (\$4.25)"				

Orders			
Key	Customer	Address	Product
1000	Joe Jones	1 Main St.	Hammer
1001	Samir Nadoy	123 Elm Pl.	Wrench

Key-Value

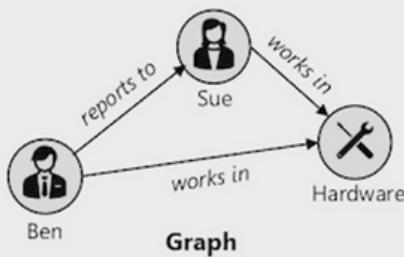
Column Family

Document

Graph



Non-relational



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

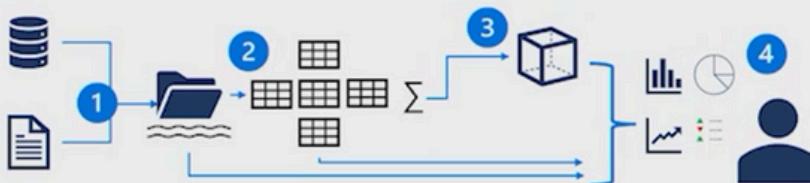


Data is stored using *transactions*

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*

Data professional roles



Database Administrator



Data Engineer



Data Analyst

Database provisioning,
configuration and management

Database security and user access

Database backups and resiliency

Database performance monitoring
and optimization

Data integration pipelines and ETL
processes

Data cleansing and transformation

Analytical data store schemas and
data loads

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 database system for relational and non-relational data



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

Platform-as-a-Service (PaaS)



Azure Synapse Analytics

- Integrated solution for data analytics in Azure
- Pipelines, Apache Spark, SQL, Data Explorer



Azure Databricks

- Lakehouse analytics platform



Azure HDInsight

- Apache open-source platform

others...

Relational tables

www.linkedin.com/in/yashbhangle

- 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	Middle	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)		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		Results		
ID	Name	Price	Name	Price
123	Hammer	2.99	Hammer	2.99
162	Screwdriver	3.49	Screwdriver	3.49
201	Wrench	4.25	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;
```



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';
```

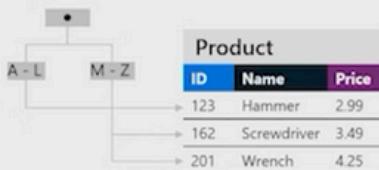
Product

ID	Name	Price
201	Wrench Spanner	4.25

Indexes

Tree-based structures that improve query performance

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



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

PaaS



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

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

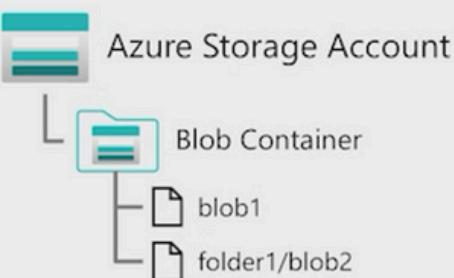
- Database service in the Microsoft cloud based on the PostgreSQL Community Edition database engine
- Hybrid relational and object storage

PaaS

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

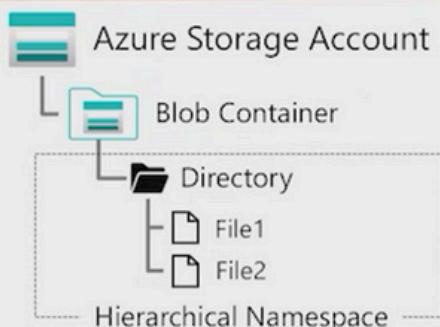
Azure Data Lake Store Gen 2

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

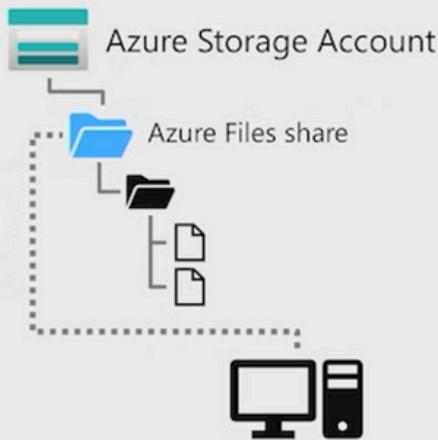


File system includes directories and files, and is compatible with large scale data analytics systems like Hadoop, Databricks, and Azure Synapse Analytics

Azure Files

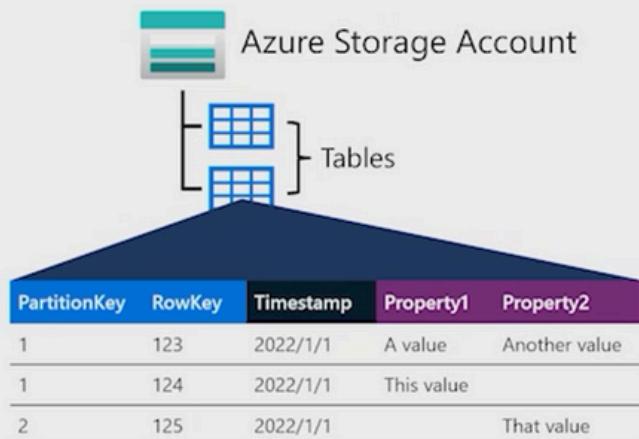
Files shares in the cloud that can be accessed from anywhere with an internet connection

- 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



Key-Value storage for application data

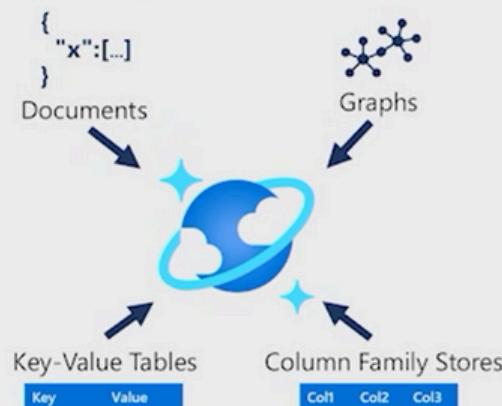
- Tables consist of *key* and *value* columns
 - Partition and row keys
 - Custom property columns for data values
 - A *Timestamp* column is added automatically to log data changes
- Rows are grouped into partitions to improve performance
- Property columns are assigned a data type, and can contain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



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

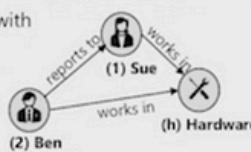
Azure Cosmos DB for Apache Cassandra

- 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*)



Elements of a large-scale data analytics solution

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



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

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 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 services



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

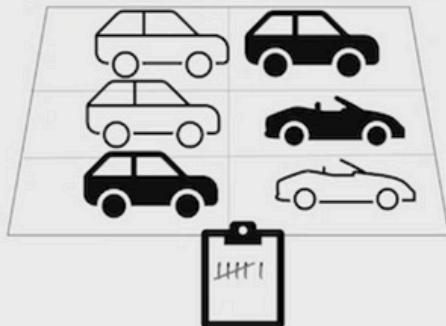
Use when you need to support multiple open-source platforms

SaaS data analytics with Microsoft Fabric



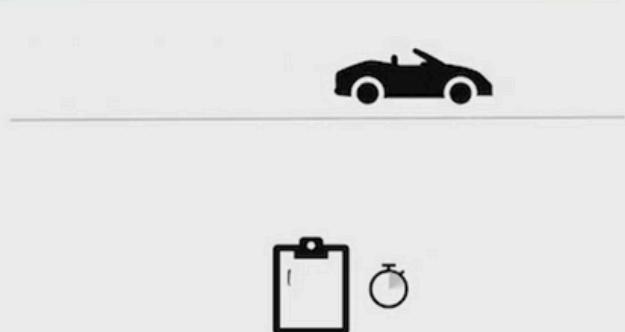
Batch vs stream processing

Batch processing



Data is collected and processed at regular intervals

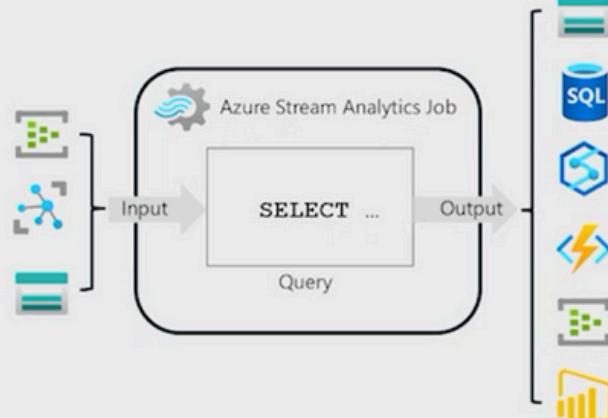
Stream processing



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

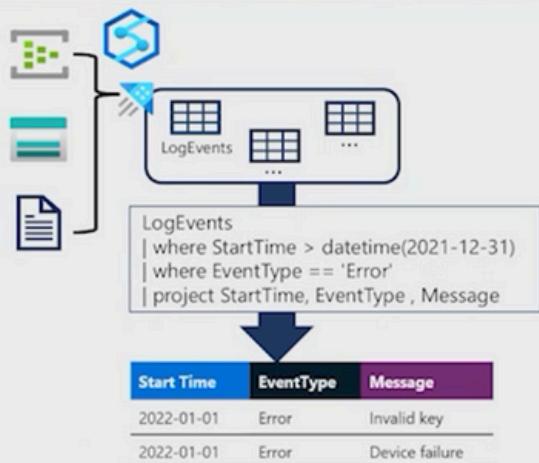
Real-time data processing with Azure Stream Analytics

- 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

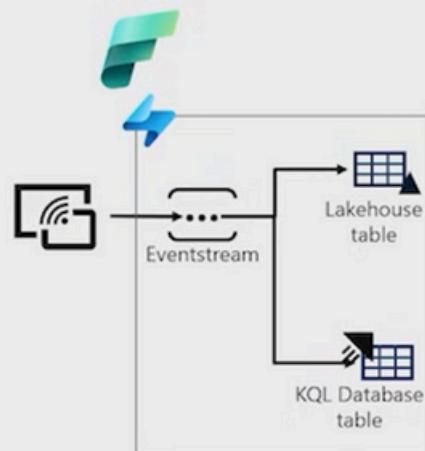
- High throughput, scalable service for batch and streaming data
 - **Azure Data Explorer** dedicated service
 - **Azure Synapse Data Explorer** runtime in Azure Synapse Analytics
- Data is ingested from streaming and batch sources into tables in a database
- Tables can be queried using *Kusto Query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



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

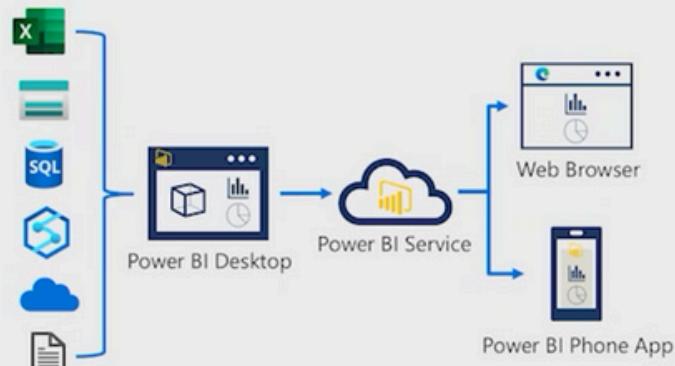
Real-time analytics in Microsoft Fabric

- 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



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

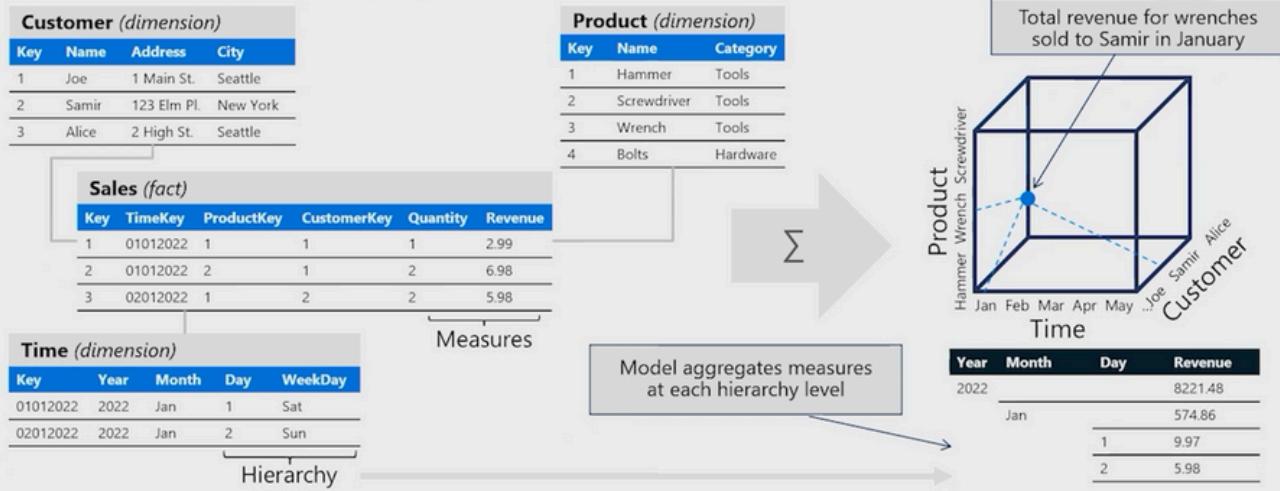
Sales (fact)					
Key	TimeKey	ProductKey	CustomerKey	Quantity	Revenue
1	01012022	1	1	1	2.99
2	01012022	2	1	2	6.98
3	02012022	1	2	2	5.98

Measures

Time (dimension)				
Key	Year	Month	Day	WeekDay
01012022	2022	Jan	1	Sat
02012022	2022	Jan	2	Sun

Hierarchy

Analytical data modeling



Common data visualizations in reports

