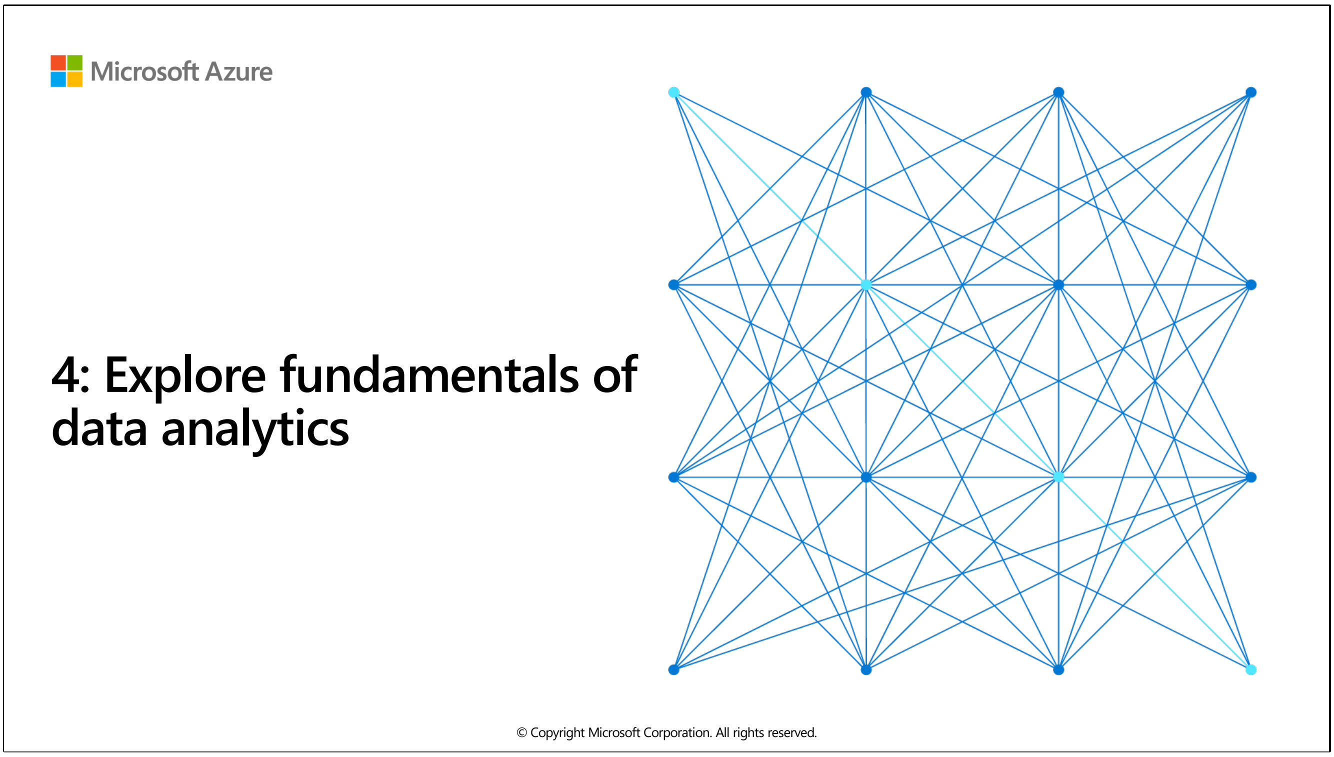
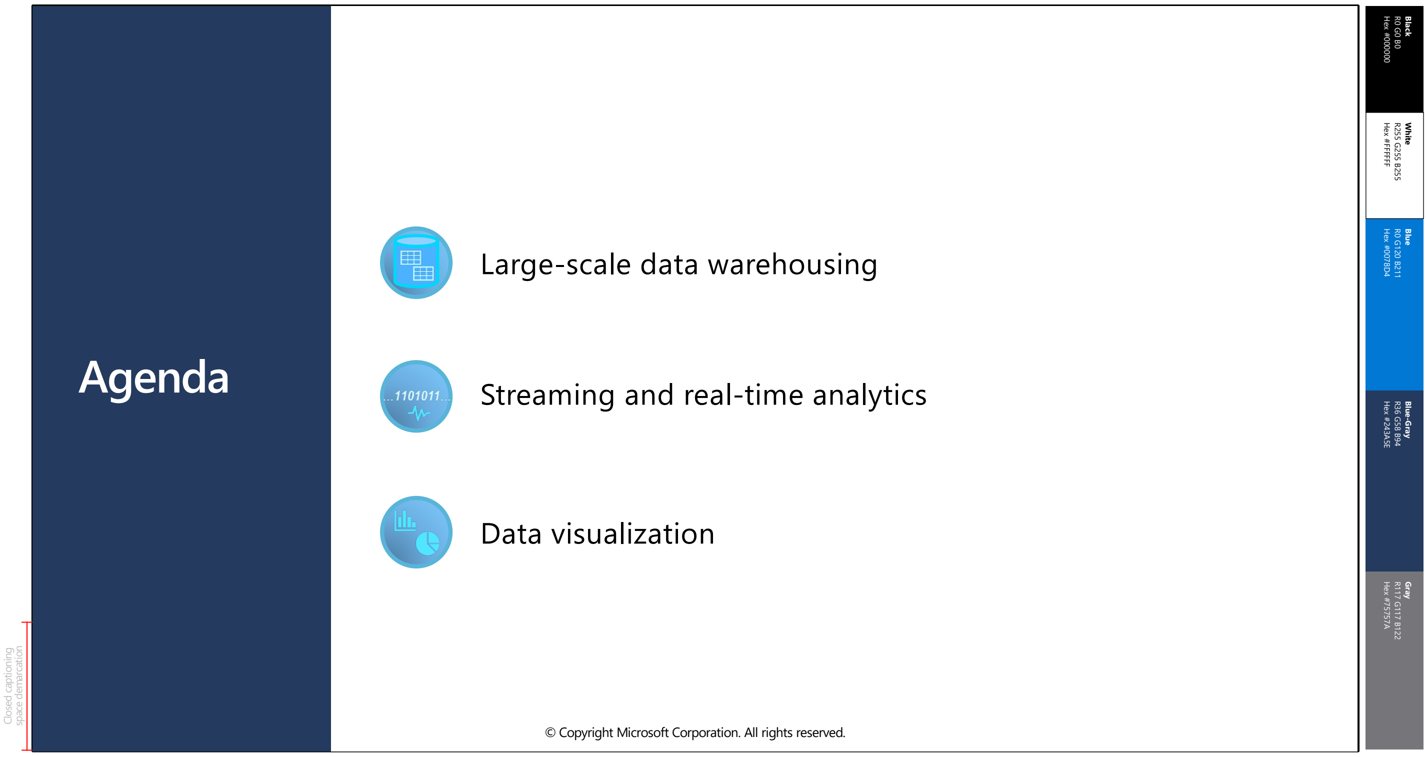


Microsoft Azure

4: Explore fundamentals of data analytics



Agenda

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

1101011
Closed captioning space demarcation
Black R0 G0 B0 Hex #000000
White R255 G255 B255 Hex #FFFFFF
Blue R0 G120 B211 Hex #0078D4
Blue-Gray R36 G58 B94 Hex #243A5E
Gray R117 G117 B122 Hex #75757A

This should take approximately 150 minutes (2.5 hours) to deliver, including the three lab exercises.

1: Large-scale data warehousing

Closed captioning space demarcation

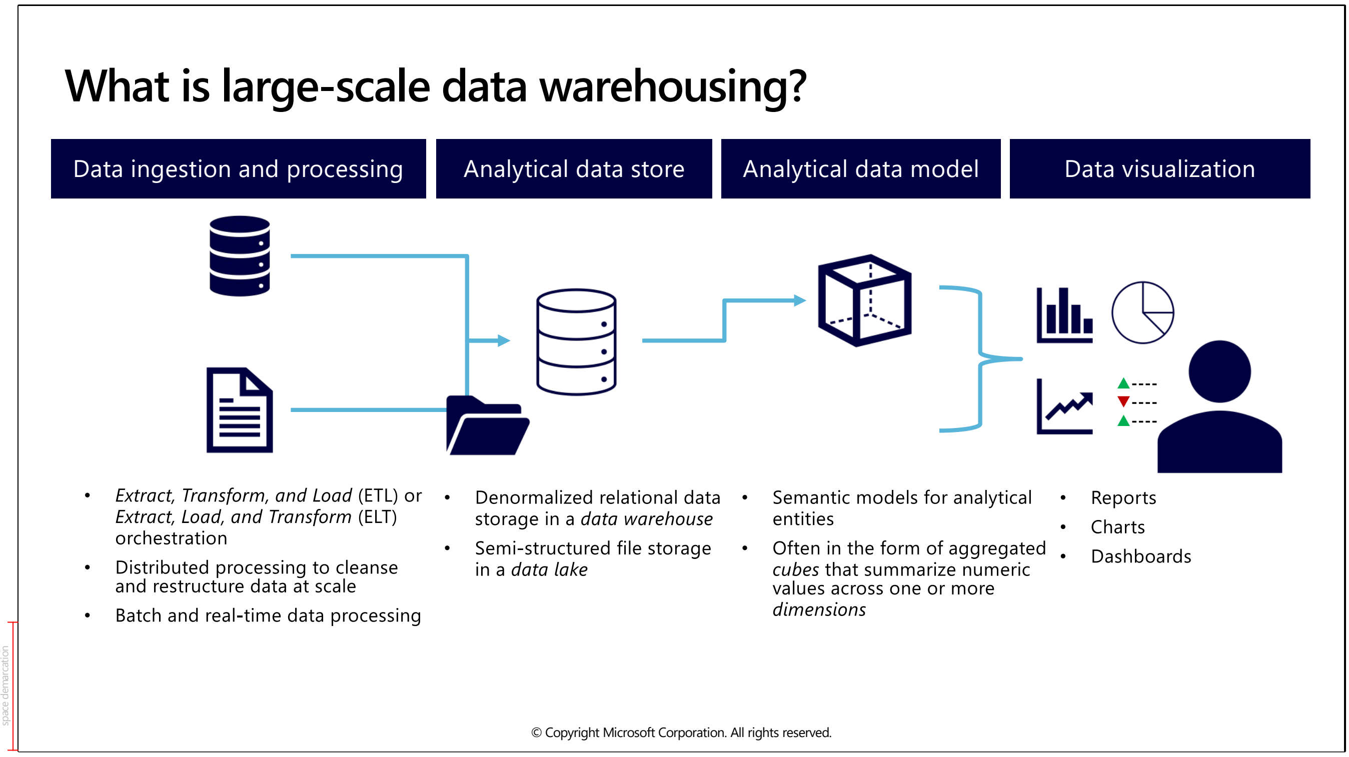
Black
R0 G0 B0
Hex #000000

White
R255 G255 B255
Hex #FFFFFF

Blue
R0 G120 B211
Hex #0078D4

Blue-Gray
R36 G58 B94
Hex #243A5E

Gray
R117 G117 B122
Hex #75757A



What is large-scale data warehousing?

- *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

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

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

- Reports
- Charts
- Dashboards

Closed captioning space demarcation

Black R0 G0 B0 Hex #000000

White R255 G255 B255 Hex #FFFFFF

Blue R0 G120 B211 Hex #0078D4

Blue-Gray R36 G58 B94 Hex #243A5E

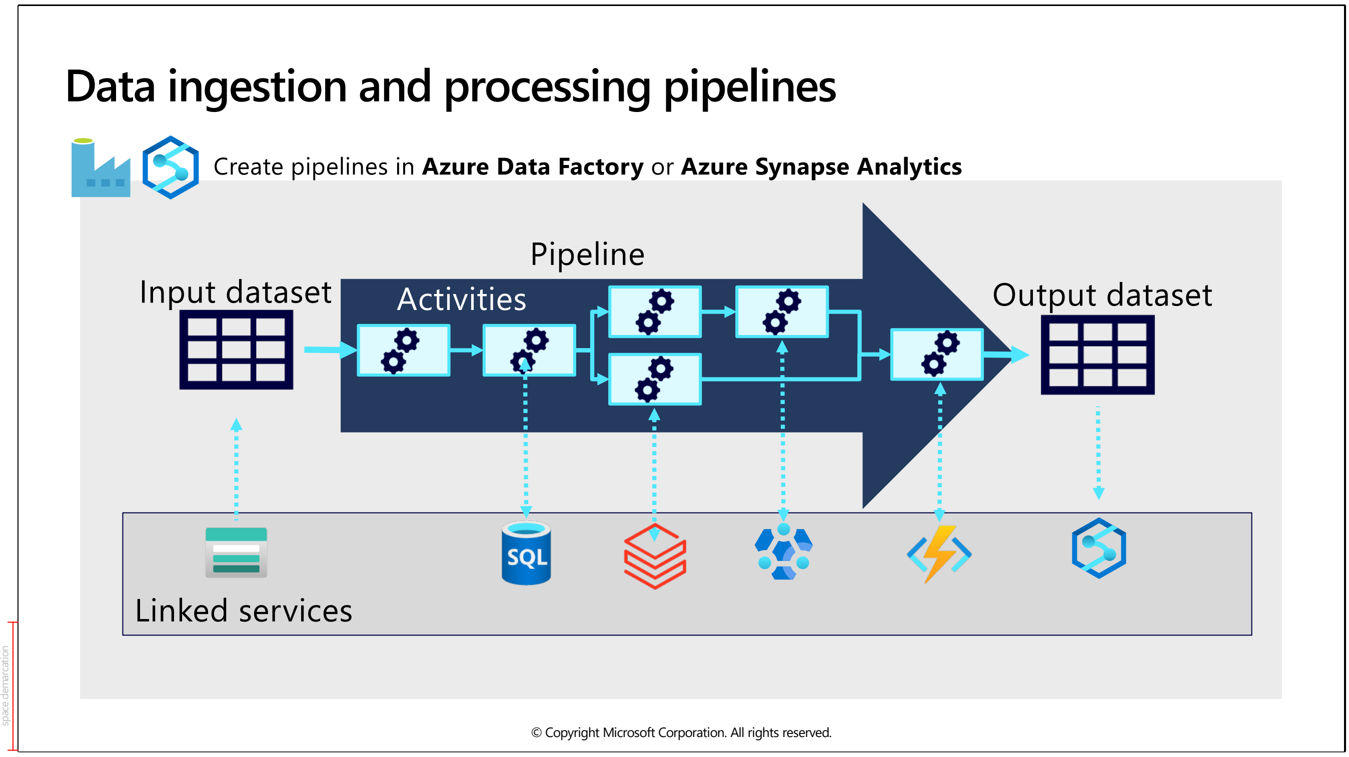
Gray R117 G117 B122 Hex #75757A

Large-scale data warehousing is a generic term that describes the infrastructure and processes used to support large-scale data analytics. Large-scale data warehousing solutions combine conventional data warehousing used to support business intelligence (BI) - which typically involves copying data from transactional data stores into a relational database with a schema that's optimized for querying and building multidimensional models; with techniques used for so-called "big data" analytics, in which large volumes of data in multiple formats are batch loaded or captured in real-time streams, and stored in a data lake from which distributed processing engines like Apache Spark are used to process the data at scale.

Large-scale data warehousing architecture can vary, as can the specific technologies used to implement it; but in general, the following components are included:

- Data ingestion and processing – data from one or more transactional data stores, files, real-time streams, or other sources is loaded into a data lake or a relational data warehouse. The load operation usually involves an extract, transform, and load (ETL) or extract load and transform process in which the data is cleaned, filtered, and restructured for analysis. In ETL processes, the data is transformed before being loaded into an analytical store, while in an ELT process the data is copied to the store and then transformed. Either way, the resulting data structure is optimized for analytical queries. The data processing is often performed by distributed systems that can process high volumes of data in parallel using multi-node clusters. Data ingestion includes both batch processing of static data and real-time processing of streaming data.
- Analytical data store – data stores for large scale analytics include relational *data warehouses*, file-system based *data lakes*, and hybrid architectures that combine features of data warehouses and data lakes (sometimes called *Data Lakehouses* or *Lake Databases*). We'll discuss these in more depth later.
- Analytical data model – while data analysts and data scientists can work with the data directly in the analytical data store, it’s common to create one or more data models that pre-aggregate the data to make it easier to produce reports, dashboards, and interactive visualizations. Often these data models are described as *cubes*, in which numeric data values are aggregated across one or more *dimensions* (for example, to determine total sales by product and region). The model encapsulates the relationships between data values and dimensional entities to support "drill-up/drill-down" analysis.
- Data visualization – data analysts consume data from analytical models, and directly from analytical stores to create reports, dashboards, and other visualizations. Additionally, users in an organization who may not be technology professionals might perform self-service data analysis and reporting. The visualizations from the

data show trends, comparisons, and key performance indicators (KPIs) for a business or other organization, and can take the form of printed reports, graphs and charts in documents or PowerPoint presentations, web-based dashboards, and interactive environments in which users can explore data visually.



Data ingestion and processing pipelines

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

Pipeline

Input dataset

Activities

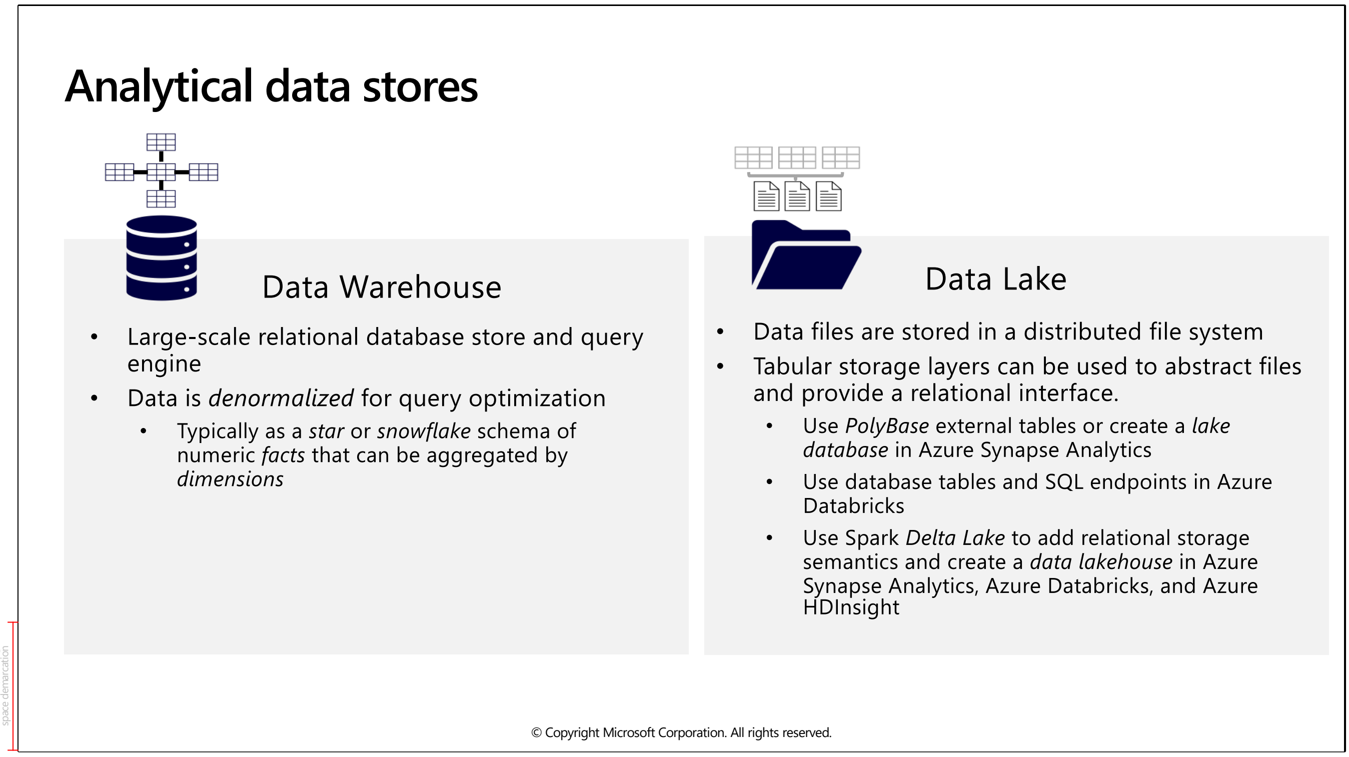
Output dataset

Linked services

Now that you understand a little about the architecture of a `large-scale` data warehousing solution, and some of the distributed processing technologies that can be used to handle large volumes of data, it's time to explore how data is ingested into an analytical data store from one or more sources.

On Azure, large-scale data ingestion is best implemented by creating *pipelines* that orchestrate ETL processes. You can create and run pipelines using Azure Data Factory, or you can use the same pipeline engine in Azure Synapse Analytics if you want to manage all of the components of your data warehousing solution in a unified workspace.

In either case, pipelines consist of one or more *activities* that operate on data. An input dataset provides the source data, and activities can be defined as a data flow that incrementally manipulates the data until an output dataset is produced. Pipelines use *linked services* to load and process data – enabling you to use the right technology for each step of the workflow. For example, you might use an Azure Blob Store linked service to ingest the input dataset, and then use services such as Azure SQL to run a stored procedure that looks up related data values, before running a data processing task on Azure Databricks or Azure HDInsight, or apply custom logic using an Azure Function. Finally, you can save the output dataset in a linked service such as Azure Synapse Analytics. Note that pipelines can support some built-in activities, which don't require a linked service.



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

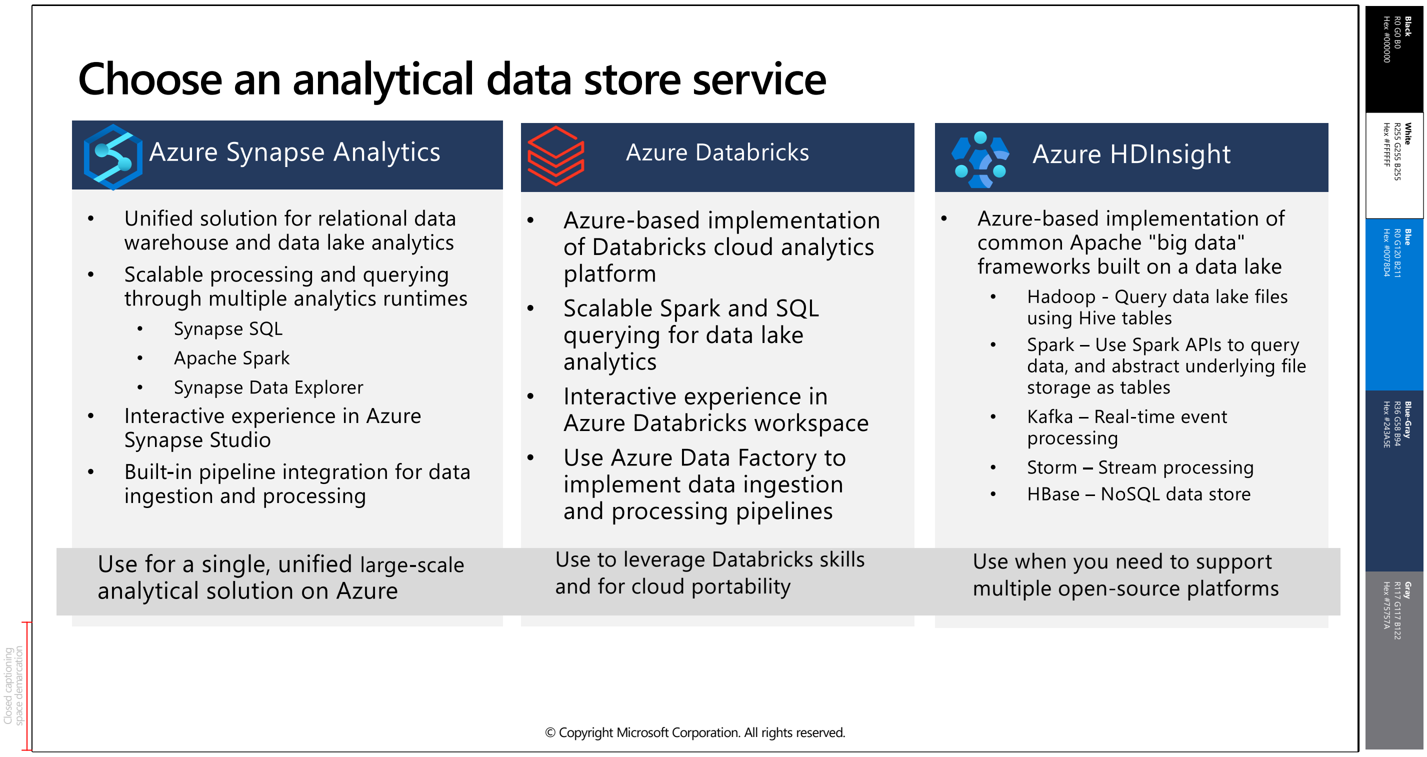
Closed captioning space demarcation

Black R0 G0 B0 Hex #000000
White R255 G255 B255 Hex #FFFFFF
Blue R0 G120 B211 Hex #0078D4
Blue-Gray R36 G58 B94 Hex #243A5E
Gray R117 G117 B122 Hex #75757A

As previously discussed, there are three common ways to implement an analytical data store.

- A data warehouse is a relational database in which the data is stored in a schema that is optimized for data analytics rather than transactional workloads. Commonly, the data from a transactional store is *denormalized* into a schema in which numeric values are stored in central *fact* tables, which are related to one or more *dimension* tables that represent entities by which the data can be aggregated. For example a fact table might contain sales order data, which can be aggregated by customer, product, store, and time dimensions (enabling you for example, to easily find monthly total sales revenue by product for each store). This kind of fact and dimension table schema is called a *star* schema; though it's often extended into a *snowflake* schema by adding additional tables related to the dimension tables to represent dimensional hierarchies (for example, product might be related to product categories. A data warehouse is a great choice when you have transactional data that can be organized into a structured schema of tables, and you want to use SQL to query them.
- A data lake is a file store, usually on a distributed file system for high performance data access. Technologies like Spark or Hadoop are often used to process queries on the stored files and return data for reporting and analytics. These systems often apply a *schema-on-read* approach to define tabular schemas on semi-structured data files at the point where the data is read for analysis, without applying constraints when it is stored. Data lakes are great for supporting a mix of structured, semi-structured, and even unstructured data that you want to analyze without the need for schema enforcement when the data is written to the store.
- You can use a hybrid approach that combines features of data lakes and data warehouses in a **Lake database** or **data lakehouse**. The raw data is stored as files in a data lake, and a relational storage layer abstracts the underlying files and expose them as tables, which can be queried using SQL. SQL pools in Azure Synapse Analytics include **PolyBase**, which enables you to define external tables based on files in a datalake (and other sources) and query them using SQL. Synapse Analytics also supports a Lake Database approach in which you can use database templates to define the relational schema of your data warehouse, while storing the underlying data in data lake storage – separating the storage and compute for your data warehousing solution. Data lakehouses are a relatively new approach in Spark-based systems, and are enabled through technologies like **Delta Lake**; which adds relational storage capabilities to Spark, so you can define tables that enforce schemas and transactional consistency, support batch-loaded

```
and streaming data sources, and provide a SQL API for querying.
```


Black R0 G0 B0 Hex #000000
White R255 G255 B255 Hex #FFFFFF
Blue R0 G120 B211 Hex #0078D4
Blue-Gray R36 G58 B94 Hex #243A5E
Gray R117 G117 B122 Hex #75757A
Closed captioning space demarcation

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

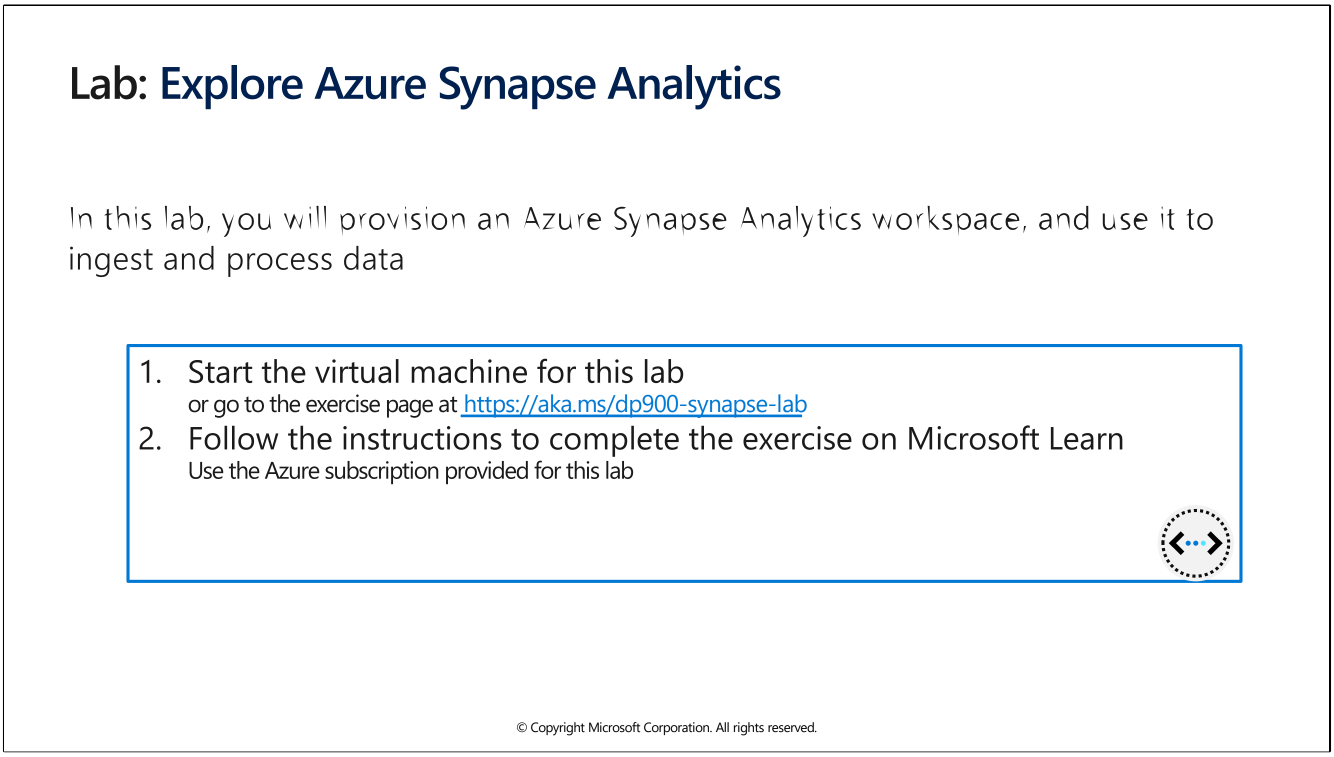
Azure HDInsight

- 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

On Azure, there are three main services that you can use to implement a large-scale analytical store.

- Azure Synapse Analytics is a unified, end-to-end solution for large scale data analytics. It brings together a number of technologies and capabilities, enabling you to combine the data integrity and reliability of a scalable, high-performance SQL Server based relational data warehouse with the flexibility of a data lake and open-source Apache Spark. It also includes native support for log and telemetry analytics with Azure Synapse Data Explorer pools, as well as built in data pipelines for data ingestion and transformation. All Azure Synapse Analytics services can be managed through a single, interactive user interface called Azure Synapse Studio, which includes the ability to create interactive *notebooks* in which Spark code and markdown content can be combined. Synapse Analytics is a great choice when you want to create a single, unified analytics solution on Azure.
- Azure Databricks is an Azure implementation of the popular Databricks platform. Databricks is a comprehensive data analytics solution built on Apache Spark, which offers native SQL capabilities as well as workload-optimized Spark runtimes for data analytics and data science, and an interactive user interface through which the system can be managed and data can be explored in interactive notebooks. Due to its common use on multiple cloud platforms, you might want to consider using Azure Databricks as your analytical store if you want to leverage existing expertise with the platform or if you need to operate in a multi-cloud environment or support a cloud-portable solution.
- Azure HDInsight is an Azure service that supports mulitple open-source data analytics cluster types. Although not as user-friendly as Azure Synapse Analytics and Azure Databricks, it can be a suitable option if your analytics solution relies on multiple open-source frameworks or if you need to migrate an existing on-premises Hadoop-based solution to the cloud.



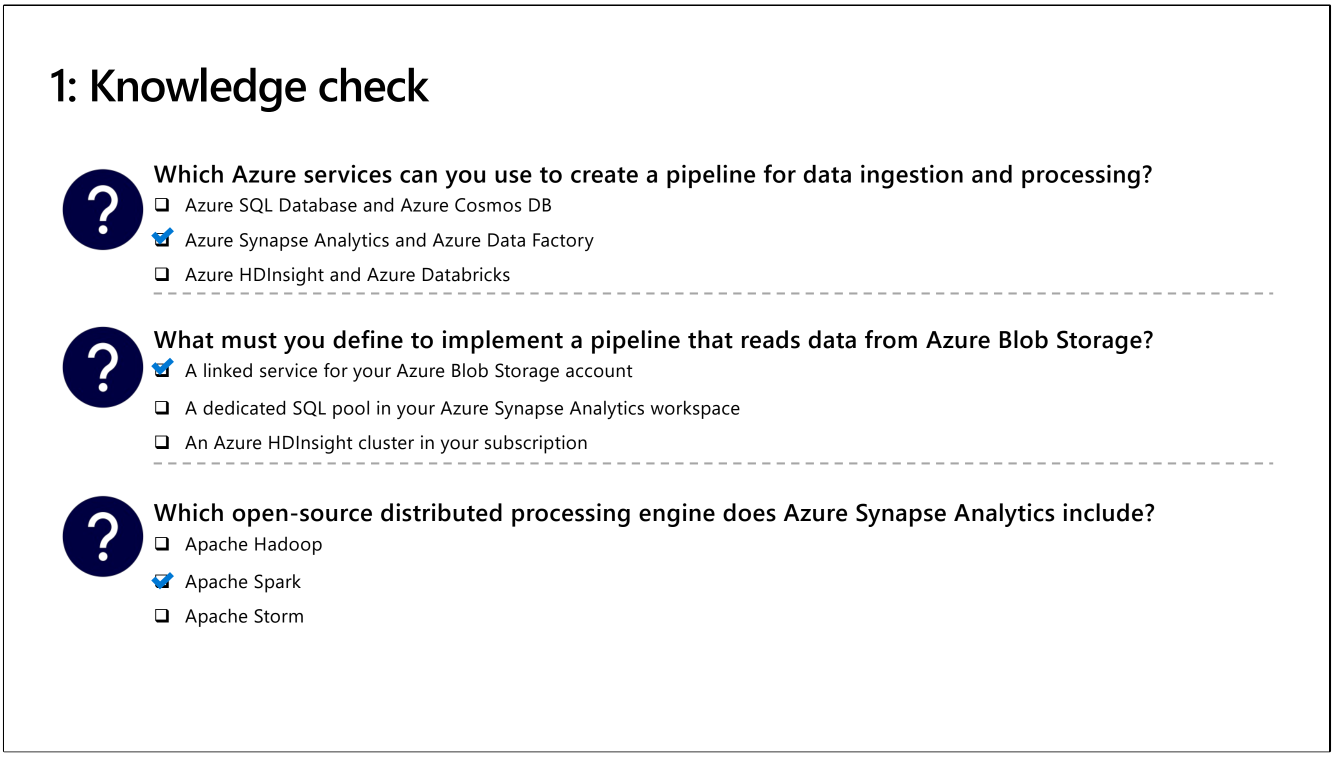
Lab: Explore Azure Synapse Analytics

In this lab, you will provision an Azure Synapse Analytics workspace, and use it to ingest and process data

1. Start the virtual machine for this lab
   or go to the exercise page at https://aka.ms/dp900-synapse-lab
2. Follow the instructions to complete the exercise on Microsoft Learn
   Use the Azure subscription provided for this lab

If necessary, demonstrate how to sign into the lab virtual machine and follow the instructions there. If you're not using a lab VM, students can follow the instructions in the GitHub page for this lab.

Students should use the Azure subscription credentials provided to them. The lab is also available from the related module on Microsoft Learn, so students can complete it later if desired; but they will need to provide their own Azure subscription to do so.



1: Knowledge check

Which Azure services can you use to create a pipeline for data ingestion and processing?

- □ Azure SQL Database and Azure Cosmos DB
- ☑ Azure Synapse Analytics and Azure Data Factory
- □ Azure HDInsight and Azure Databricks

What must you define to implement a pipeline that reads data from Azure Blob Storage?

- ☑ A linked service for your Azure Blob Storage account
- □ A dedicated SQL pool in your Azure Synapse Analytics workspace
- □ An Azure HDInsight cluster in your subscription

Which open-source distributed processing engine does Azure Synapse Analytics include?

- □ Apache Hadoop
- ☑ Apache Spark
- □ Apache Storm

Allow students a few minutes to think about the questions, and then use the animated slide to reveal the correct answers.

2: Streaming and real-time analytics

Closed captioning space demarcation

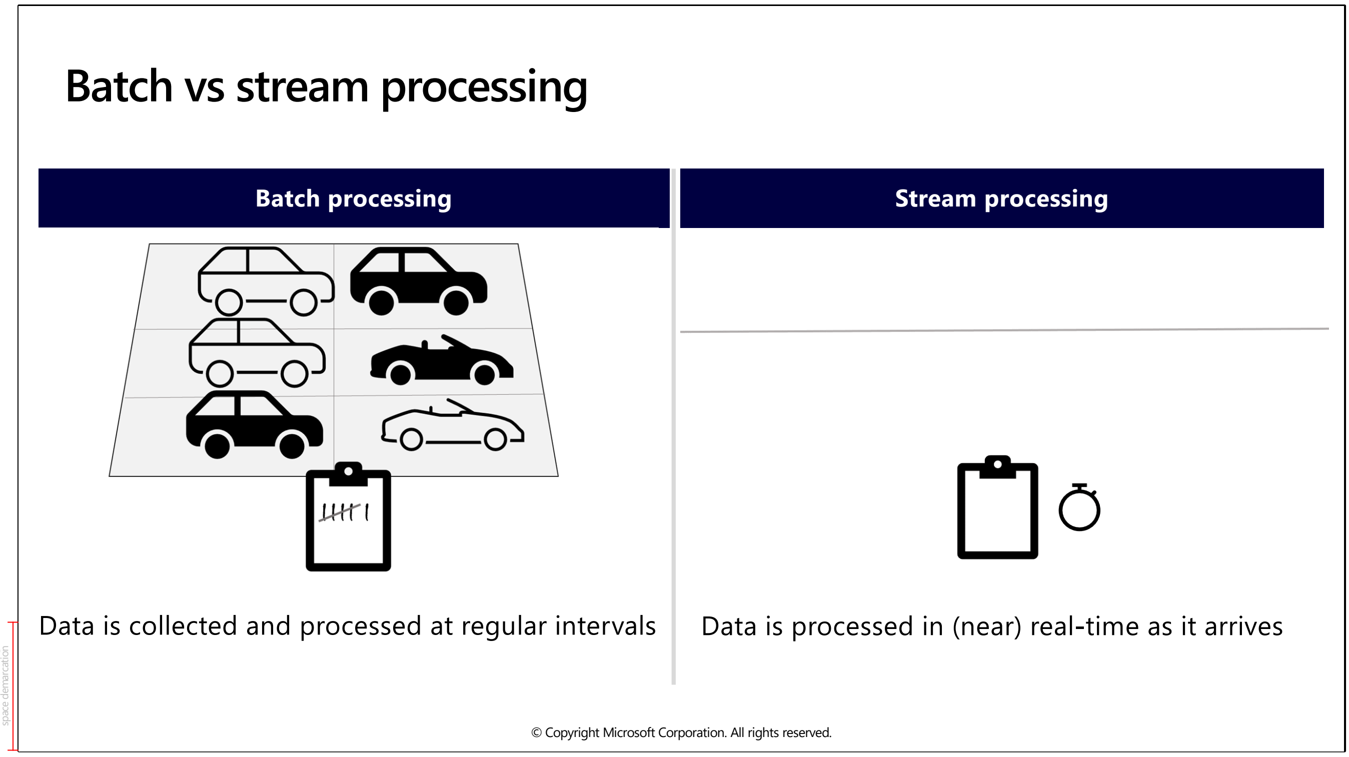
Black R0 G0 B0 Hex #000000

White R255 G255 B255 Hex #FFFFFF

Blue R0 G120 B211 Hex #0078D4

Blue-Gray R36 G58 B94 Hex #243A5E

Gray R117 G117 B122 Hex #75757A



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

Data processing is simply the conversion of raw data to meaningful information through a process. There are two general ways to process data:

- *Batch processing*, in which multiple data records are collected and stored before being processed together in a single operation.
- *Stream processing*, in which a source of data is constantly monitored and processed in real time as new data events occur.

In batch processing, newly arriving data elements are collected into a group. The whole group is then processed at a future time as a batch. Exactly when each group is processed can be determined in a number of ways. For example, you can process data based on a scheduled time interval (for example, every hour), or it could be triggered when a certain amount of data has arrived, or as the result of some other event.

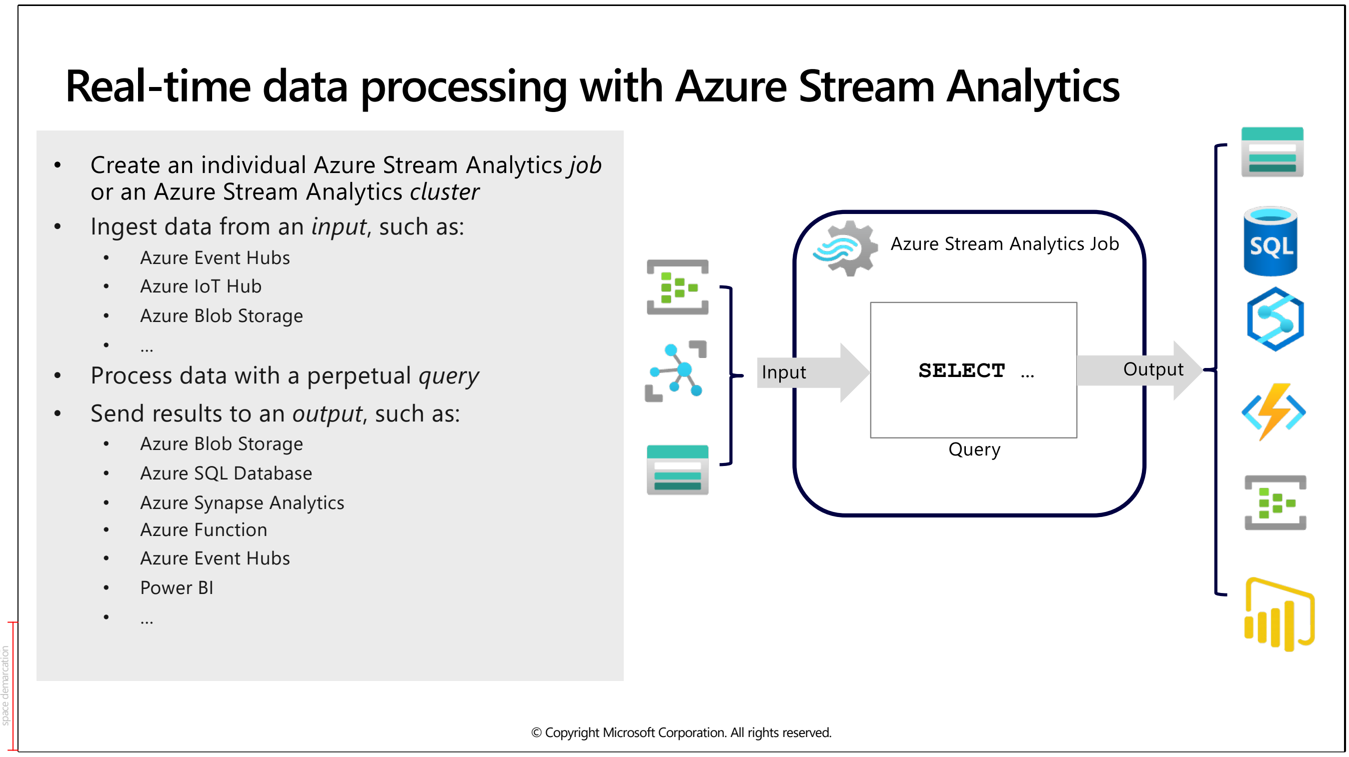
For example, suppose you want to analyze road traffic by counting the number of cars on a stretch of road. A batch processing approach to this would require that you collect the cars in a parking lot, and then count them in a single operation while they're at rest.

In stream processing, each new piece of data is processed when it arrives. Unlike batch processing, there's no waiting until the next batch processing interval - data is processed as individual units in real-time rather than being processed a batch at a time. Streaming data processing is beneficial in most scenarios where new, dynamic data is generated on a continual basis.

For example, a better approach to our hypothetical car counting problem might be to apply a *streaming* approach, by counting the cars in real-time as they pass:

In this approach, you don't need to wait until all of the cars have parked to start processing them, and you can aggregate the data over time intervals; for example, by counting the number of cars that pass each minute.

Many large-scale analytics solutions include a mix of batch and stream processing, enabling both historical and real-time data analysis. It's common for stream processing solutions to capture real-time data, process it by filtering or aggregating it, and present it through real-time dashboards and visualizations (for example, showing the running total of cars that have passed along a road within the current hour), while persisting the processed results in a data store for historical analysis alongside batch processed data (for example, to enable analysis of traffic volumes over the past year).



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
  - ...

Closed captioning space demarcation

Black R0 G0 B0 Hex #000000
White R255 G255 B255 Hex #FFFFFF
Blue R0 G120 B211 Hex #0078D4
Blue-Gray R36 G58 B94 Hex #243A5E
Gray R117 G117 B122 Hex #75757A

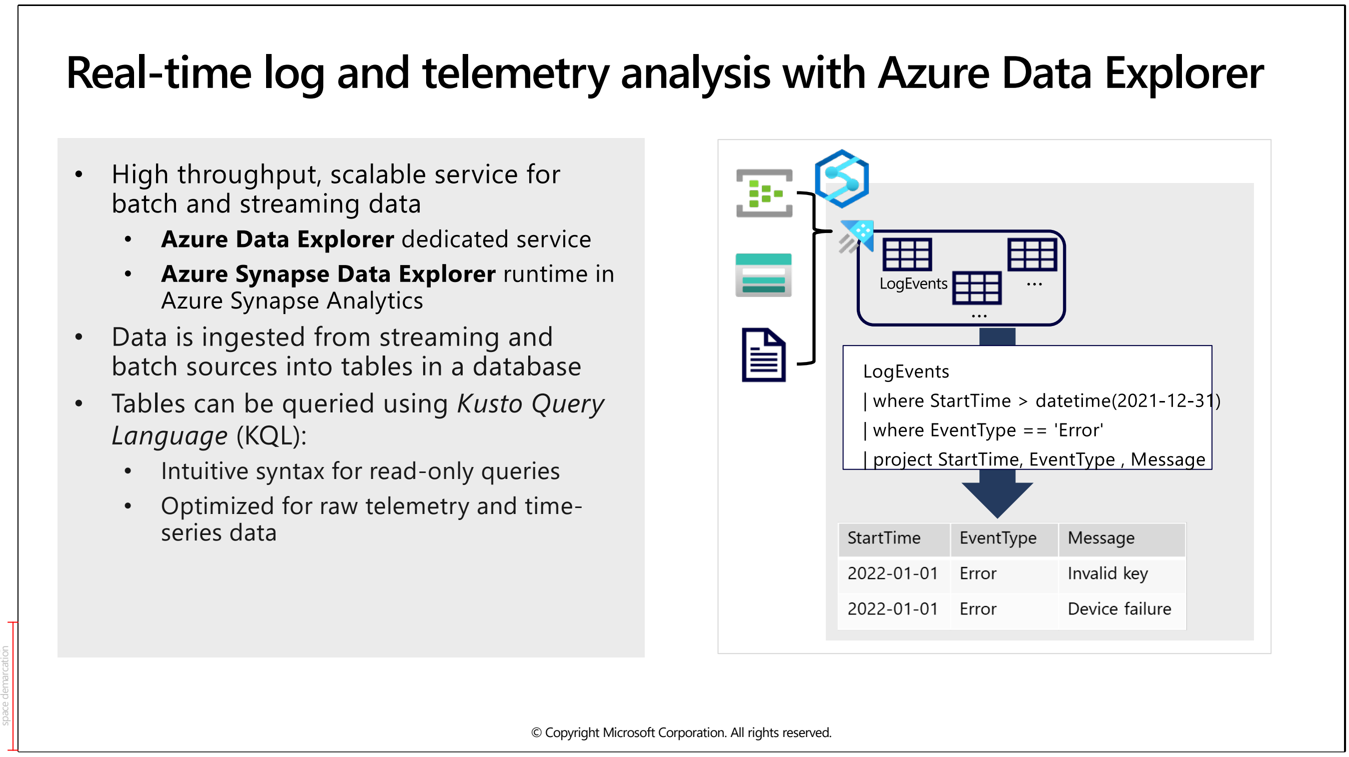
Azure Stream Analytics is a service for complex event processing and analysis of streaming data. Stream Analytics is used to:

- Ingest data from an *input*, such as an Azure event hub, Azure IoT Hub, or Azure Storage blob container.
- Process the data by using a *query* to select, project, and aggregate data values.
- Write the results to an *output*, such as Azure Data Lake Gen 2, Azure SQL Database, Azure Synapse Analytics, Azure Functions, Azure event hub, Microsoft Power BI, or others.

Once started, a Stream Analytics query will run perpetually, processing new data as it arrives in the input and storing results in the output.

The easiest way to use Azure Stream Analytics is to create a Stream Analytics *job* in an Azure subscription, configure its input(s) and output(s), and define the query that the job will use to process the data. The query is expressed using structured query language (SQL) syntax and can incorporate static reference data from multiple data sources to supply lookup values that can be combined with the streaming data ingested from an input.

If your stream process requirements are complex or resource-intensive, you can create a Stream Analysis *cluster*, which uses the same underlying processing engine as a Stream Analytics job, but in a dedicated tenant (so your processing is not affected by other customers) and with configurable scalability that enables you to define the right balance of throughput and cost for your specific scenario.



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
```

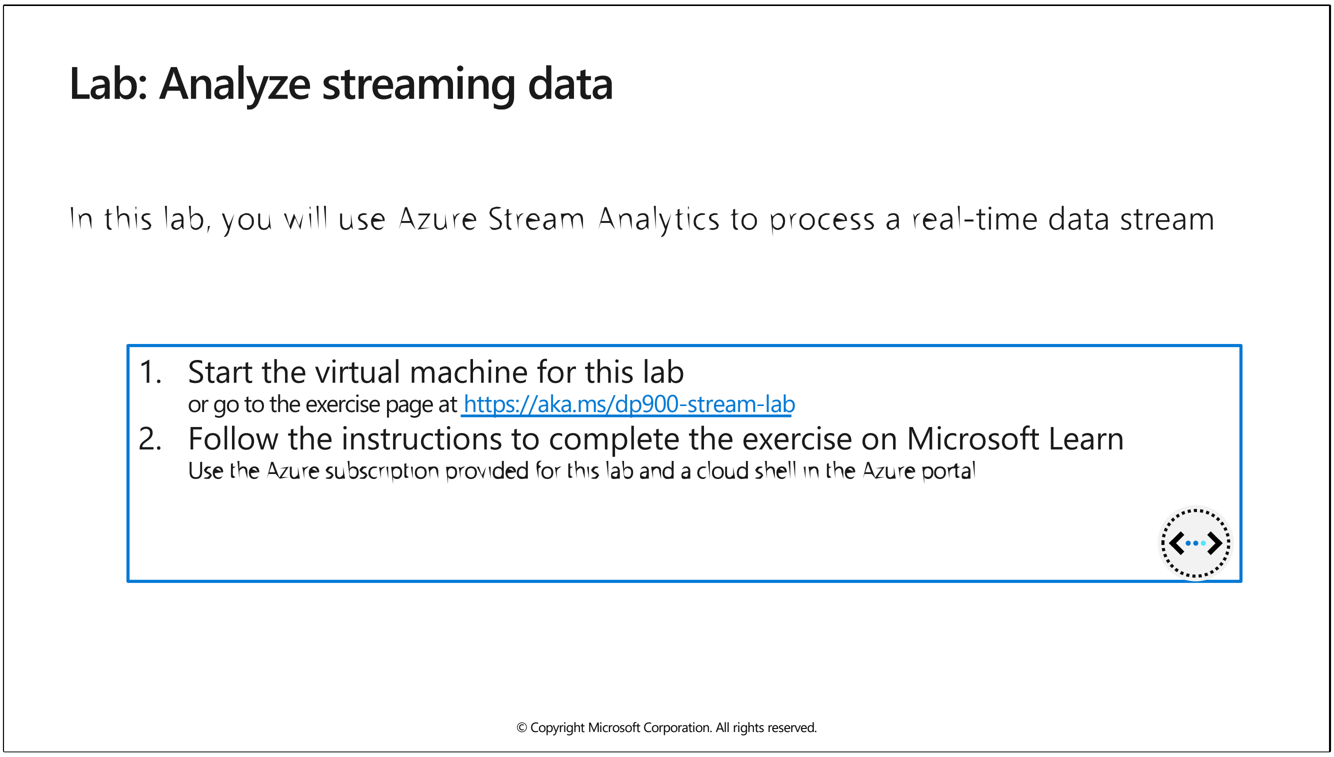
StartTime	EventType	Message
2022-01-01	Error	Invalid key
2022-01-01	Error	Device failure

Azure Data Explorer is a standalone Azure service for efficiently analyzing data. The service is also encapsulated as a runtime in Azure Synapse Analytics, where it is referred to as *Azure Synapse Data Explorer*; enabling you to build and manage analytical solutions that combine SQL, Spark and Data Explorer analytics in a single workspace.

Data is ingested into Data Explorer through one or more *connectors* or by writing a minimal amount of code. This enables you to quickly ingest data from a wide variety of data sources, including both static and streaming sources. The ingested data is stored in tables in a Data Explorer database, where automatic indexing enables high-performance queries.

To query the tables, you can use Kusto Query Language (KQL), a language that is specifically optimized for fast read performance – particularly with telemetry data that includes a timestamp attribute, such as log files or internet-of-things (IoT) device messages.

To learn more about Azure Data Explorer, see https://docs.microsoft.com/learn/modules/intro-to-azure-data-explorer/intro-to-azure-data-explorer/ (note that this Learn module is not part of the official course materials for this course – it supports further learning beyond the scope of the Data Fundamentals certification).



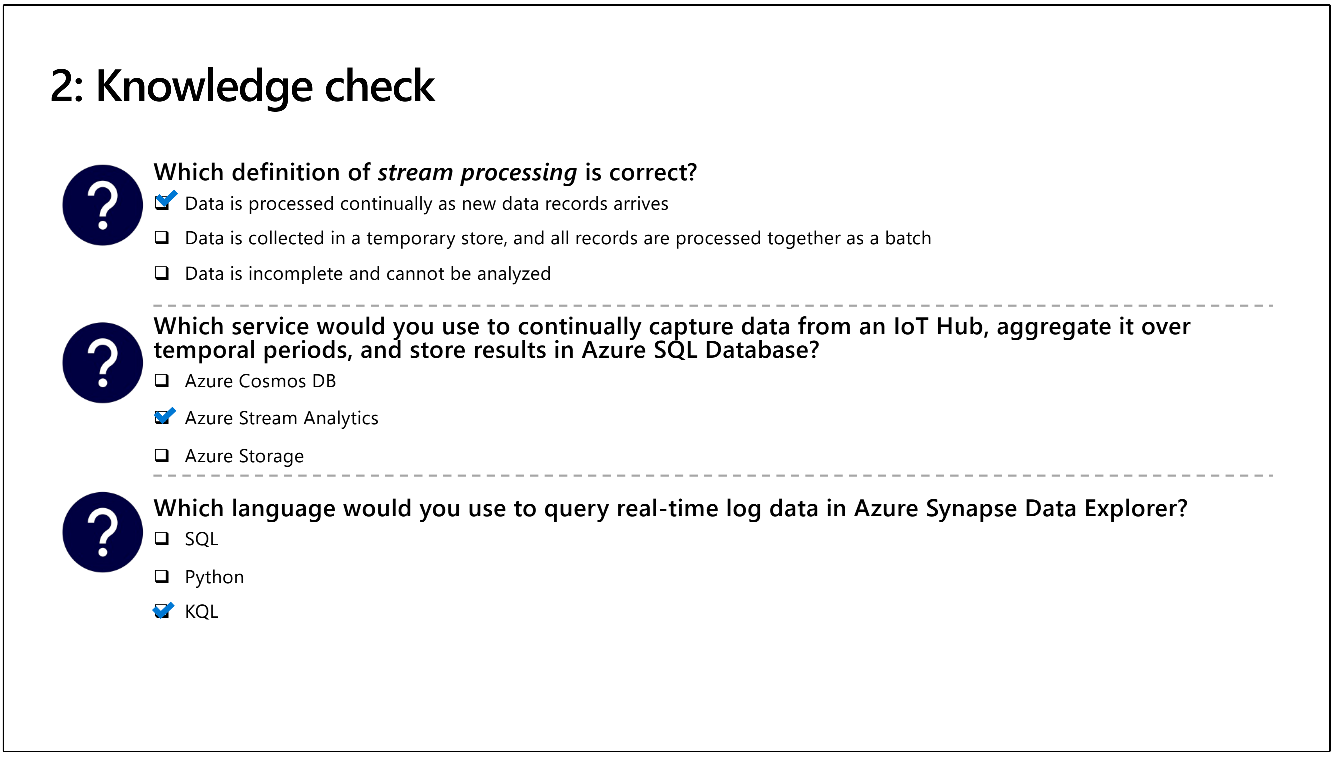
Lab: Analyze streaming data

In this lab, you will use Azure Stream Analytics to process a real-time data stream

1. Start the virtual machine for this lab
   or go to the exercise page at https://aka.ms/dp900-stream-lab
2. Follow the instructions to complete the exercise on Microsoft Learn
   Use the Azure subscription provided for this lab and a cloud shell in the Azure portal

If necessary, demonstrate how to sign into the lab virtual machine and follow the instructions there. If you're not using a lab VM, students can follow the instructions in the GitHub page for this lab.

Students should use the Azure subscription credentials provided to them. The lab (along with others) is also available from the related module on Microsoft Learn, so students can complete it later if desired; but they will need to provide their own Azure subscription to do so.



2: Knowledge check

Which definition of *stream processing* is correct?

☑ Data is processed continually as new data records arrives

□ Data is collected in a temporary store, and all records are processed together as a batch

□ Data is incomplete and cannot be analyzed

Which service would you use to continually capture data from an IoT Hub, aggregate it over temporal periods, and store results in Azure SQL Database?

□ Azure Cosmos DB

☑ Azure Stream Analytics

□ Azure Storage

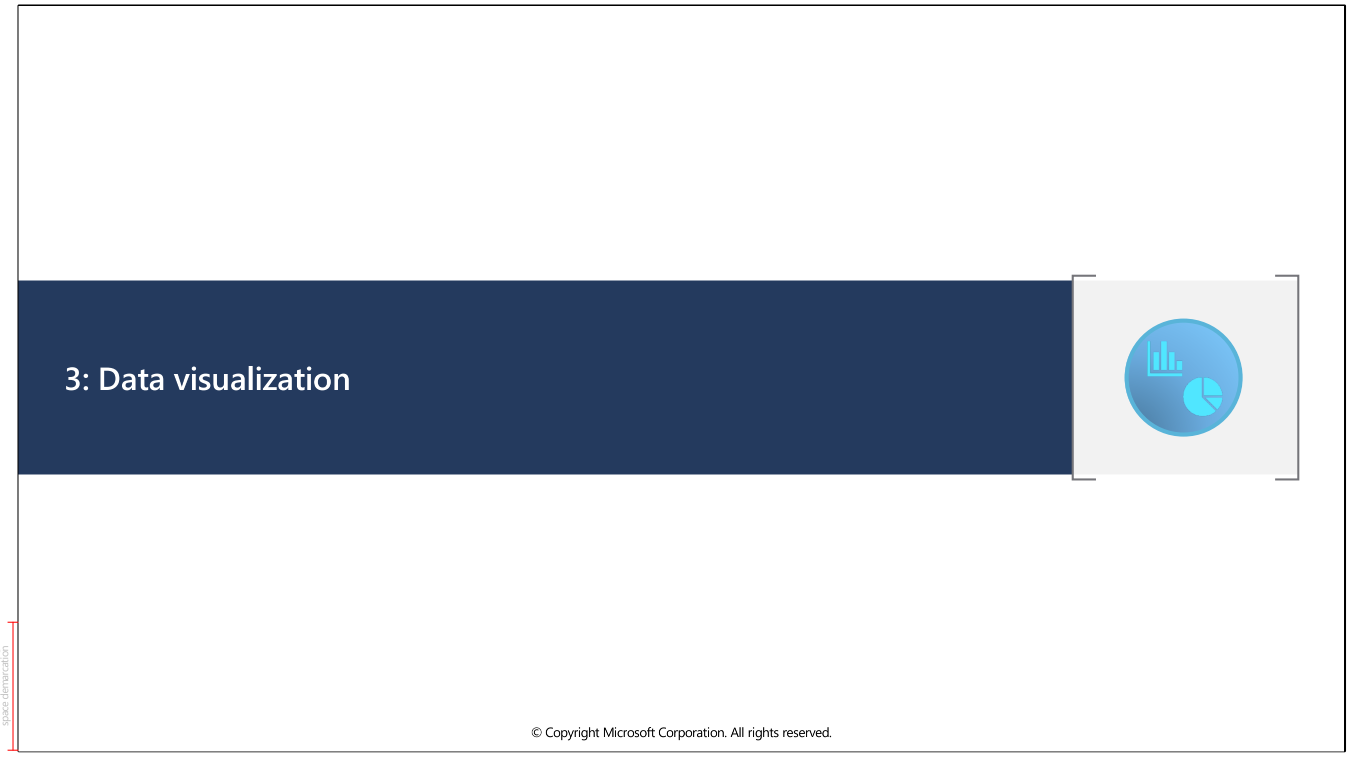
Which language would you use to query real-time log data in Azure Synapse Data Explorer?

□ SQL

□ Python

☑ KQL

Allow students a few minutes to think about the questions, and then use the animated slide to reveal the correct answers.



3: Data visualization

Closed captioning space demarcation

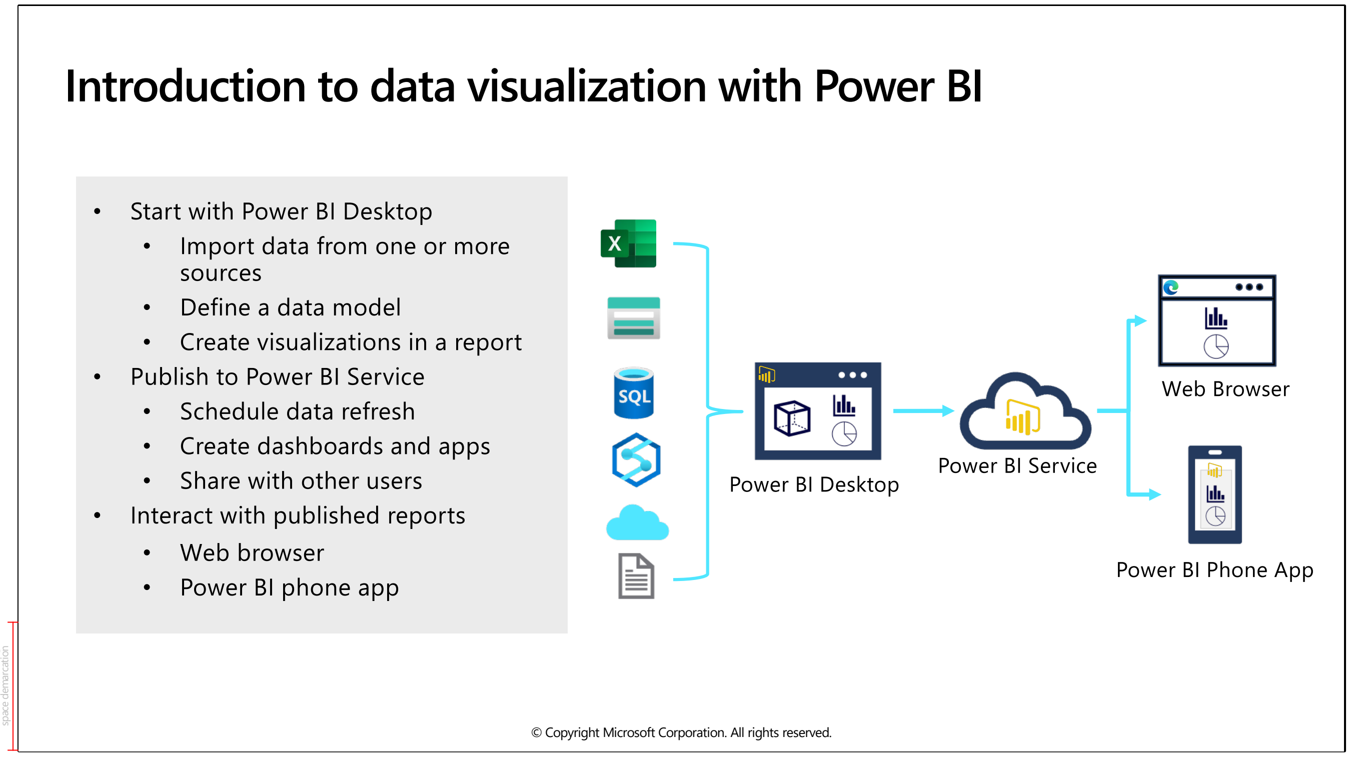
Black
R0 G0 B0
Hex #000000

White
R255 G255 B255
Hex #FFFFFF

Blue
R0 G120 B211
Hex #0078D4

Blue-Gray
R36 G58 B94
Hex #243A5E

Gray
R117 G117 B122
Hex #75757A



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

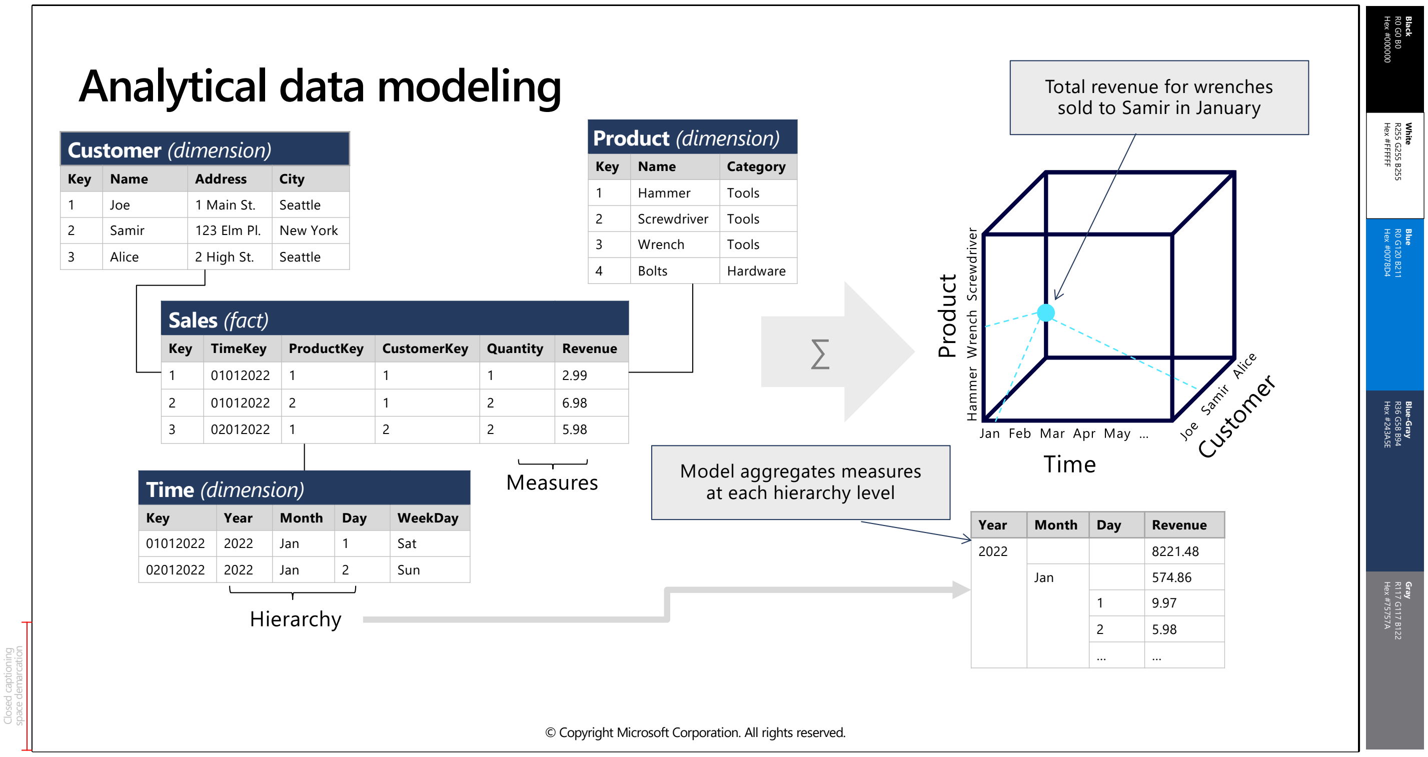
Data modelling and visualization is at the heart of "business intelligence" (BI) workloads that are supported by large-scale data analytics solutions.

Microsoft Power BI is a suite of tools and services that data analysts can use to build interactive data visualizations for business users to consume.

A typical workflow for creating a data visualization solution starts with **Power BI Desktop**, a Windows based application in which you can import data from a wide range of data sources, combine and organize the data from these sources in an analytics data model, and create reports that contain interactive visualizations of the data.

After you've created data models and reports, you can publish them to the **Power BI service**; a cloud service in which reports can be published and interacted with by business users. You can also do some basic data modeling and report editing directly in the service using a web browser, but the functionality for this is limited compared to the Power BI Desktop tool. You can use the service to schedule refreshes of the data sources on which your reports are based, and to share reports with other users. You can also define dashboards and apps that combine related reports in a single, easy to consume location.

Users can consume reports, dashboards, and apps in the Power BI service through a web browser, or on mobile devices by using the **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
		...	...

∑
Product
Hammer Wrench Screwdriver
Jan Feb Mar Apr May ...
Time
Joe Samir Alice
Customer
Closed captioning space demarcation
Black R0 G0 B0 Hex #000000
White R255 G255 B255 Hex #FFFFFF
Blue R0 G120 B211 Hex #0078D4
Blue-Gray R36 G58 B94 Hex #243A5E
Gray R117 G117 B122 Hex #75757A

The slide focuses on general concepts – students will see how to work with a model in Power BI in the lab.

Analytical models enable you to structure data to support analysis. Models are based on related tables of data and define the numeric values that you want to analyze or report (known as *measures*) and the entities by which you want to aggregate them (known as *dimensions*). For example, a model might include a table containing numeric measures for *sales* (such as *revenue* or *quantity*) and dimensions for *products*, *customers*, and *time*. This would enable you aggregate sale measures across one or more dimensions (for example, to identify total revenue by customer, or total items sold by product per month). Conceptually, the model forms a multidimensional structure, which is commonly referred to as a *cube*, in which any point where the dimensions intersect represents an aggregated measure for those dimensions.

Tables and schema

Dimension tables represent the entities by which you want to aggregate numeric measures – for example *product* or *customer*. Each entity is represented by a row with a unique *key* value. The remaining columns represent *attributes* of a entity – for example, products have names and categories, and customers have addresses and cities. It's common in most analytical models to include a *Time* dimension so that you can aggregate numeric measures associated with events over time.

The numeric measures that will be aggregated by the various dimensions in the model are stored in *Fact tables*. Each row in a fact table represents a recorded event that has numeric measures associated with it. For example, the Sales table on the slide represents sales transactions for individual items, and includes numeric values for quantity sold and revenue.

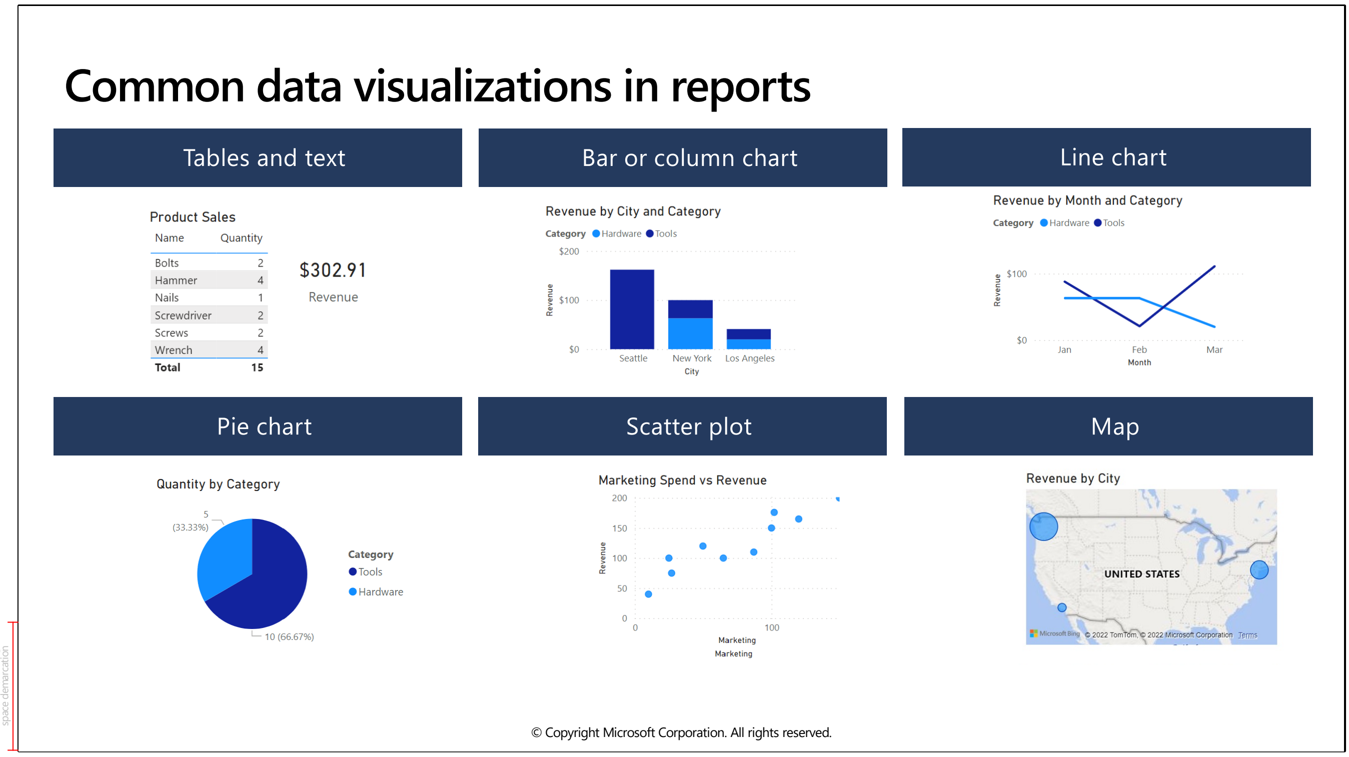
This type of schema, where a fact table is related to one or more dimension tables, is referred to as a *star* schema (imagine there are five dimensions related to a single fact table – the schema would form a five-pointed star!). You can also define more a complex schema in which dimension tables are related to additional tables containing more details (for example, you could represent attributes of product categories in a separate *Category* table that is related to the *Product* table – in which case the design is referred to as a *snowflake* schema. The schema of fact and dimension tables is used to create an analytical model, in which measure aggregations across all dimensions are pre-calculated; making performance of analysis and reporting activities much faster than calculating the aggregations each time.

Attribute hierarchies

One final thing worth considering about analytical models is the creation of attribute *hierarchies* that enable you to quickly *drill-up* or *drill-down* to find aggregated values at different levels in a hierarchical dimension. For example, consider the attributes in the dimension tables we've discussed so far. In the *Product* table, you can form a hierarchy in which each category might include multiple named products. Similarly, in the *Customer* table, a hierarchy could be formed to represent multiple named customers in each city. Finally, in the *Time* table, you can form a hierarchy of year, month, and day. The model can be built with pre-aggregated values for each level of a hierarchy, enabling you to quickly change the scope of your analysis – for example, by viewing total sales by year, and then *drilling down* to see a more detailed breakdown of total sales by month.

Analytical modeling in Microsoft Power BI

You can use Power BI to define an analytical model from tables of data, which can be imported from one or more data source. You can then use the data modeling interface on the **Model** tab of Power BI Desktop to define your analytical model by creating relationships between fact and dimension tables, defining hierarchies, setting data types and display formats for fields in the tables, and managing other properties of your data that help define a rich model for analysis.



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

Line chart

Pie chart

Scatter plot

Map

© Copyright Microsoft Corporation. All rights reserved.

Closed captioning space demarcation

Black
R0 G0 B0
Hex #000000

White
R255 G255 B255
Hex #FFFFFF

Blue
R0 G120 B211
Hex #0078D4

Blue-Gray
R36 G58 B94
Hex #243A5E

Gray
R117 G117 B122
Hex #75757A

The slide focuses on general concepts – students will see how to use Power BI to create a report with interactive data visualizations in the lab.

After you've created a model, you can use it to generate data visualizations that can be included in a report.

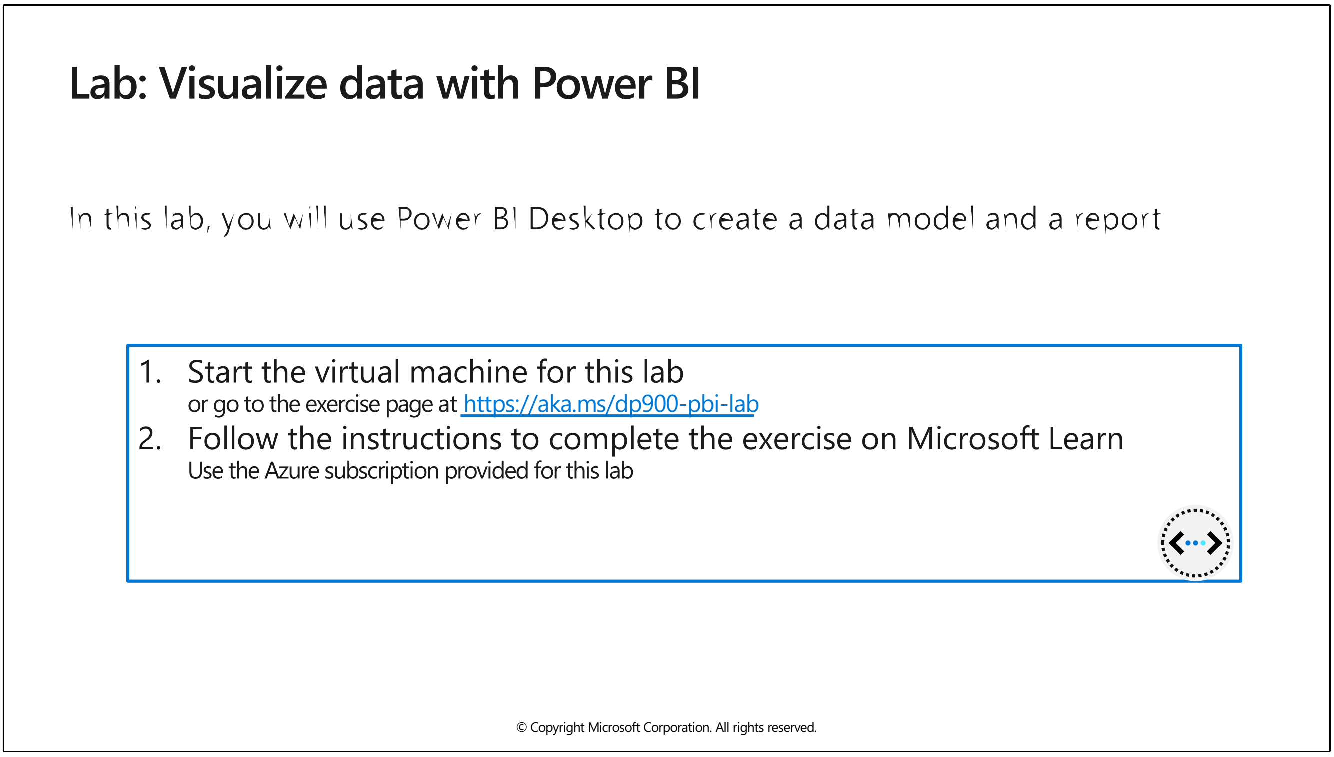
There are many kinds of data visualization, some commonly used and some more specialized. Power BI includes an extensive set of built-in visualizations, which can be extended with custom and third-party visualizations. The slide shows some common data visualizations but is by no means a complete list.

- **Tables and text** are often the simplest way to communicate data. Tables are useful when a lot of related values must be displayed, and individual text values in *cards* can be a useful way to show important figures or metrics.
- **Bar and column** charts are a good way to visually compare numeric values for discrete categories.
- **Line charts** can also be used to compare categorized values and are particularly useful when you need to examine trends, often over time.
- **Pie charts** are often used in business reports to visually compare categorized values as proportions of a total.
- **Scatter plots** are useful when you want to compare two numeric measures

and identify a relationship or correlation between them.

- **Maps** are a great way to visually compare values for different geographic areas or locations.

In Power BI, the visual elements in a report are automatically related to one another and provide interactivity. For example, selecting an individual category in one visual element will automatically filter and highlight that category in other related visualizations in the report.



Lab: Visualize data with Power BI

In this lab, you will use Power BI Desktop to create a data model and a report

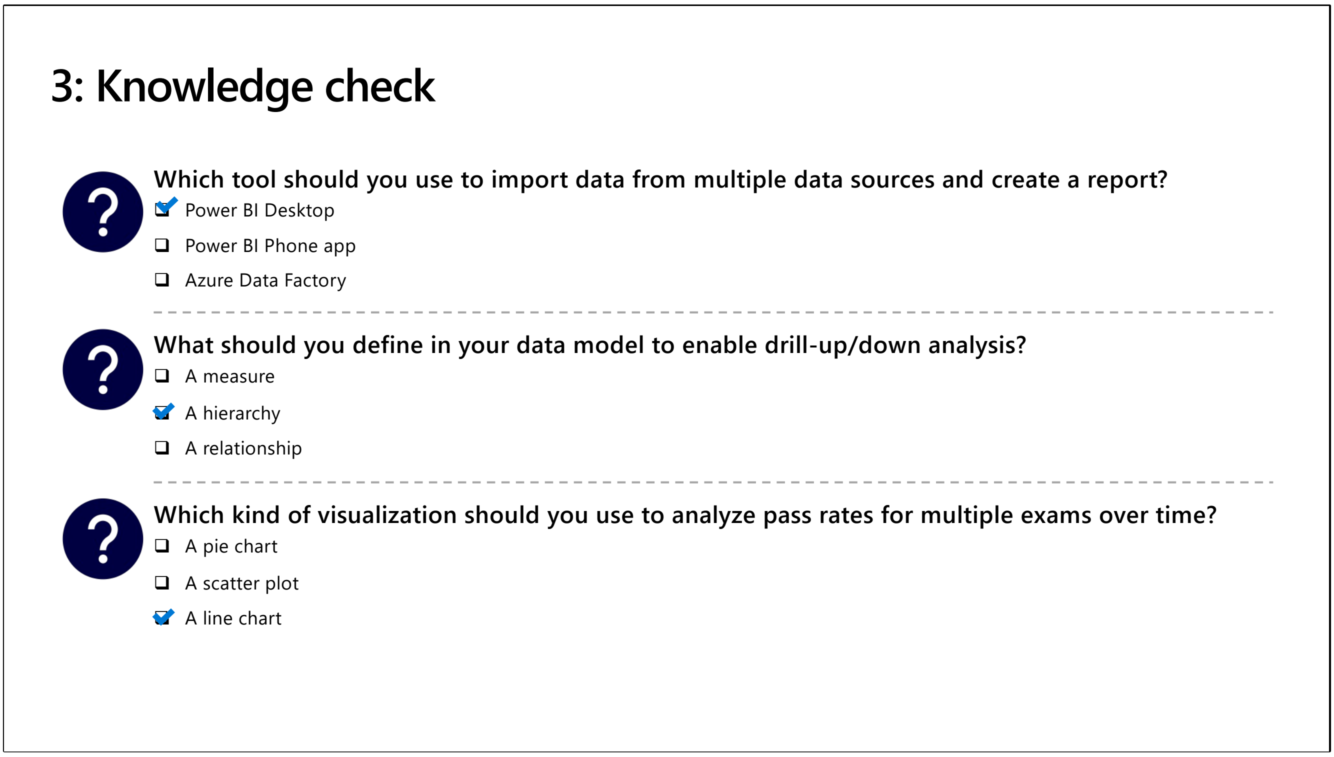
1. Start the virtual machine for this lab
   or go to the exercise page at https://aka.ms/dp900-pbi-lab
2. Follow the instructions to complete the exercise on Microsoft Learn
   Use the Azure subscription provided for this lab

If necessary, demonstrate how to sign into the lab virtual machine and follow the instructions there. If you’re not using a lab VM, students can follow the instructions in the GitHub page for this lab.

The lab is also available from the related module on Microsoft Learn, so students can complete it later if desired.

This lab does not require an Azure subscription, but it does require that students are using a Windows-based computer on which they can install Power BI Desktop. If the student's own computer is not running Windows, they should complete the lab using the virtual machine.

The download link for Power BI Desktop might need to be copied from the instructions and pasted into the VM browser.



3: Knowledge check

Which tool should you use to import data from multiple data sources and create a report?

☑ Power BI Desktop

□ Power BI Phone app

□ Azure Data Factory

What should you define in your data model to enable drill-up/down analysis?

□ A measure

☑ A hierarchy

□ A relationship

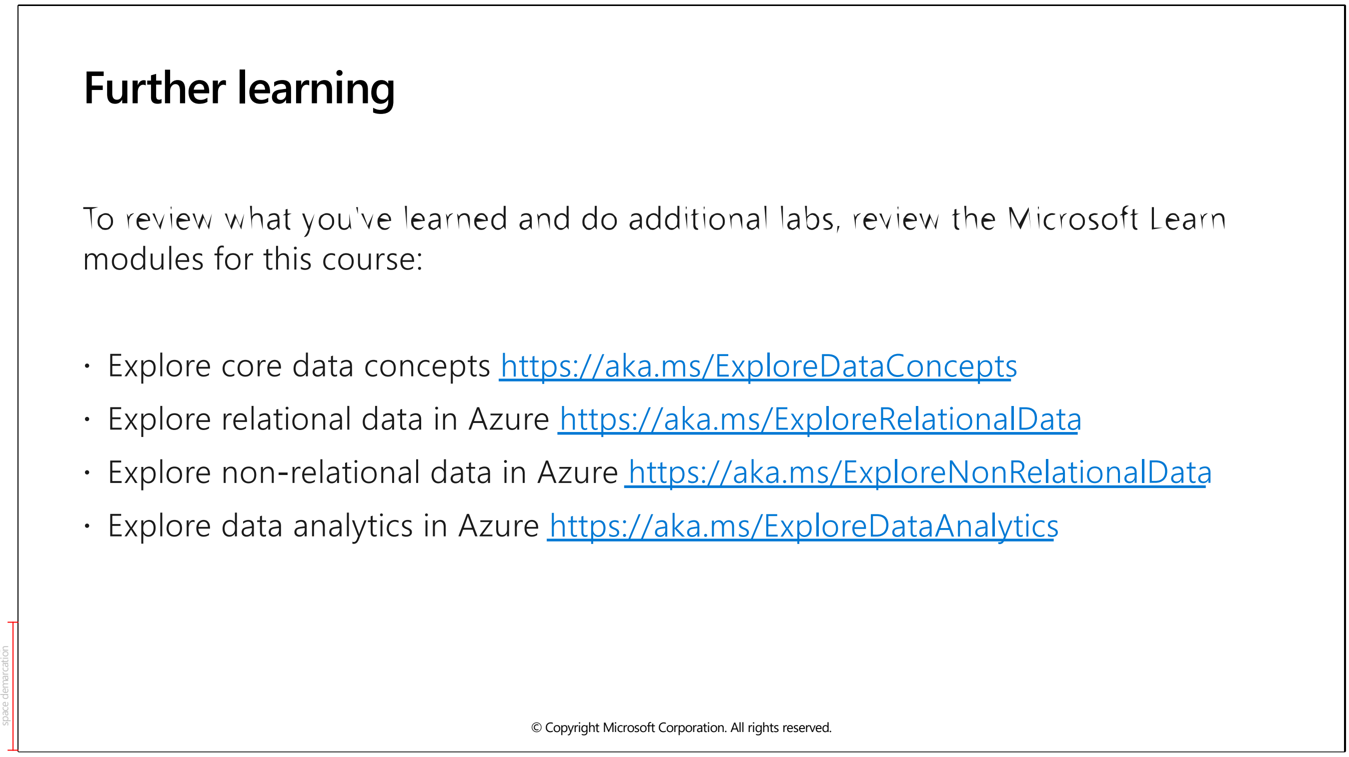
Which kind of visualization should you use to analyze pass rates for multiple exams over time?

□ A pie chart

□ A scatter plot

☑ A line chart

Allow students a few minutes to think about the questions, and then use the animated slide to reveal the correct answers.



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

Closed captioning space demarcation

Black
R0 G0 B0
Hex #000000

White
R255 G255 B255
Hex #FFFFFF

Blue
R0 G120 B211
Hex #0078D4

Blue-Gray
R36 G58 B94
Hex #243A5E

Gray
R117 G117 B122
Hex #75757A

© Copyright Microsoft Corporation. All rights reserved.

Black
R0 G0 B0
Hex #000000

White
R255 G255 B255
Hex #FFFFFF

Blue
R0 G120 B211
Hex #0078D4

Blue-Gray
R36 G58 B94
Hex #243A5E

Gray
R117 G117 B122
Hex #75757A