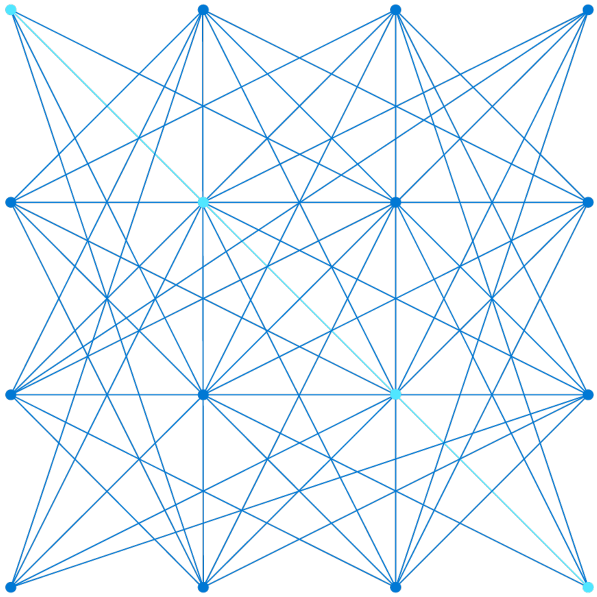
Microsoft Azure

2: Explore fundamentals of relational data in Azure



Agenda

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

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 60 minutes to deliver, including 20-25 minutes for the lab exercise

1: Explore relational data concepts

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

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

In a relational database schema, data is stored in tables; which consist of rows and columns. Relational tables are a format for *structured* data, and each row in a table has the same columns; though in some cases, not all columns need to have a value – for example, a customer table might include a **MiddleName** column; which can be empty (or *NULL*) for rows that represent customers with no middle name or whose middle name is unknown).

Each column stores data of a specific *datatype*. For example, An **Email** column in a **Customer** table would likely be defined to store character-based (text) data (which might be fixed or variable in length), a **Price** column in a **Product** table might be defined to store decimal numeric data, while a **Quantity** column in an **Order** table might be constrained to integer numeric values; and an **OrderDate** column in the same **Order** table would be defined to store date/time values. The available datatypes that you can use when defining a table depend on the database system you are using; though there are standard datatypes defined by the American National Standards Institute (ANSI) that are supported by most database systems.

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
...	...	...	...	...

Customer

ID	FirstName	LastName	Address	City
1	Joe	Jones	1 Main St.	Seattle
2	Samir	Nadoy	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

- 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

Don't get bogged down in details of 1st, 2nd, 3rd, 4th, etc, normal form for this audience. The essential learning point is that normalization is commonly used in relational databases to separate data for each entity into multiple related tables, minimizing duplication of data values and enforcing data integrity through specific data types for each piece of data and referential integrity (for example to ensure that orders only reference valid customers).

Normalization is a term used by database professionals for a schema design process that minimizes data duplication and enforces data integrity.

To understand the core principles of normalization, suppose the table on the left of the slide represents a spreadsheet that a company uses to track its sales. Notice that the customer and product details are duplicated for each individual item sold; and that the customer name and postal address, and the product name and price are combined in the same spreadsheet cells.

Now look at how normalization has changed the way the data is stored. Each *entity* that is represented in the data (customer, product, sales order, and line item) is stored in its own table, and each discrete attribute of those entities is in its own column. Instances of each entity are uniquely identified by an ID or other key value, and when one entity references another (for example, an order has an associated customer), the primary key of the related entity is stored as a foreign key – so we can look up the address of the customer (which is stored only once) for each record in the Order table by referencing the corresponding record in the Customer table. Typically, a relational database management system (RDBMS) can enforce *referential integrity* to ensure that a value entered into a foreign key field has an existing corresponding primary key in the related table – for example, preventing orders for non-existent customers.

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*

Data Definition Language (DDL)

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

Product

ID	Name	Price

Data Control Language (DCL)

```
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)

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

Results

Name	Price
Hammer	2.99
Screwdriver	3.49
Wrench	4.25

The goal of this topic is not to teach students how to write SQL queries; but rather to help them understand that SQL is a standard language used to define and work with relational data structures in a database, and to differentiate between the three common kinds of SQL statements to manage database object definitions, control access, and manipulate data.

SQL is a standard language for working with relational databases, with syntax standards that are maintained by the American National Standards Institute (ANSI) and International Standards Organization (ISO). Most relational database systems (RDBMS) vendors extend the standard language with some proprietary syntax – for example Transact-SQL / T-SQL (Microsoft SQL Server based systems), PL/SQL (Oracle), and pgSQL (PostgreSQL).

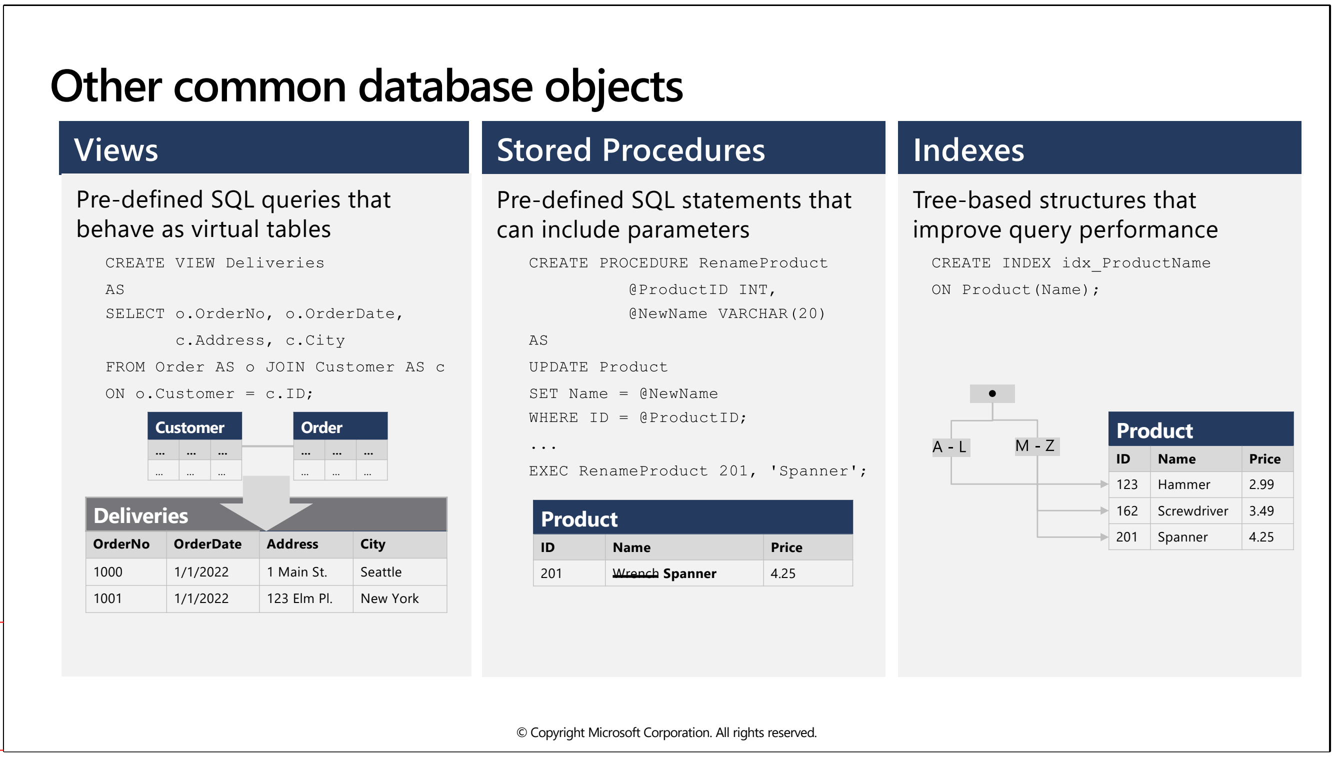
There are three broad types of SQL statements that can be used in a database system:

- Data Definition Language (DDL) is used to manage objects such as tables in the database. For example, you can CREATE new objects, and ALTER or DROP existing objects. The example on the slide shows a CREATE TABLE statement used to create a new, empty table named Product.
- Data Control Language (DCL) is used to manage access to objects in a database. You can GRANT, DENY, or REVOKE specific permissions for specific users (and groups of users). The example on the slide grants user1 permission to use SELECT, INSERT, and UPDATE statements on the Product table.
- Data Manipulation Language (DML) is the most commonly used type of SQL, and is generally used to INSERT, UPDATE, DELETE, or SELECT data in tables. The example on the slide assumes that some data has previously been inserted into the Product table, and shows the results returned by a SELECT query that retrieves the name and price of all products with a price greater than 2.50, sorted in order of price.

This slide shows a core set of SQL statements and examples. The SQL language is extensive, and there are other statements not shown here. Additionally, the syntax for the statements that are shown here can be much more complex than these simple examples.

If students are interested in exploring SQL beyond this data fundamentals course, recommend they attend course DP-080: Querying Data with Microsoft Transact-SQL (details at https://docs.microsoft.com/learn/certifications/courses/dp-080t00) or review the ***Get Started Querying with***

Transact-SQL *learning path on Microsoft Learn at* ***https://docs.microsoft.com/learn/paths/get-started-querying-with-transact-sql/****.*



>Animated slide, click to proceed

Don't go into great detail about the implementation of these objects. The key learning point is to be aware at a high-level of some of the common types of object found in a database other than tables.

In addition to tables, databases can contain other kinds of object that enable you to work with data.

- Views are pre-defined SQL SELECT queries that return a tabular dataset. Views behave as virtual tables, and can themselves be queried using SELECT statements, just like tables. They're often used to abstract the normalized schema of the database to encapsulate data from one or more tables.
- Stored Procedures are pre-defined SQL statements that can be run on-demand. They can be parameterized, and are often used to encapsulate data operations to insert, delete, or update records for data entitites.
- Indexes are tree-based structures that enable the database query engine to find individual records based on specific column values more quickly than if they just read the entire table.

These types of database object, and others, enable you to build a comprehensive relational database that applications can use to store, manage, and retrieve details of entities efficiently and securely.

1: Knowledge check

Which one of the following statements is a characteristic of a relational database?

- [ ] All columns in a table must be of the same data type
- [x] A row in a table represents a single instance of an entity
- [ ] Rows in the same table can contain different columns

Which SQL statement is used to query tables and return data?

- [ ] QUERY
- [ ] READ
- [x] SELECT

What is an index?

- [x] A structure that enables queries to locate rows in a table quickly
- [ ] A virtual table based on the results of a query
- [ ] A pre-defined SQL statement that modifies data

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

2: Explore Azure services for relational data

Closed captioning space demarcation

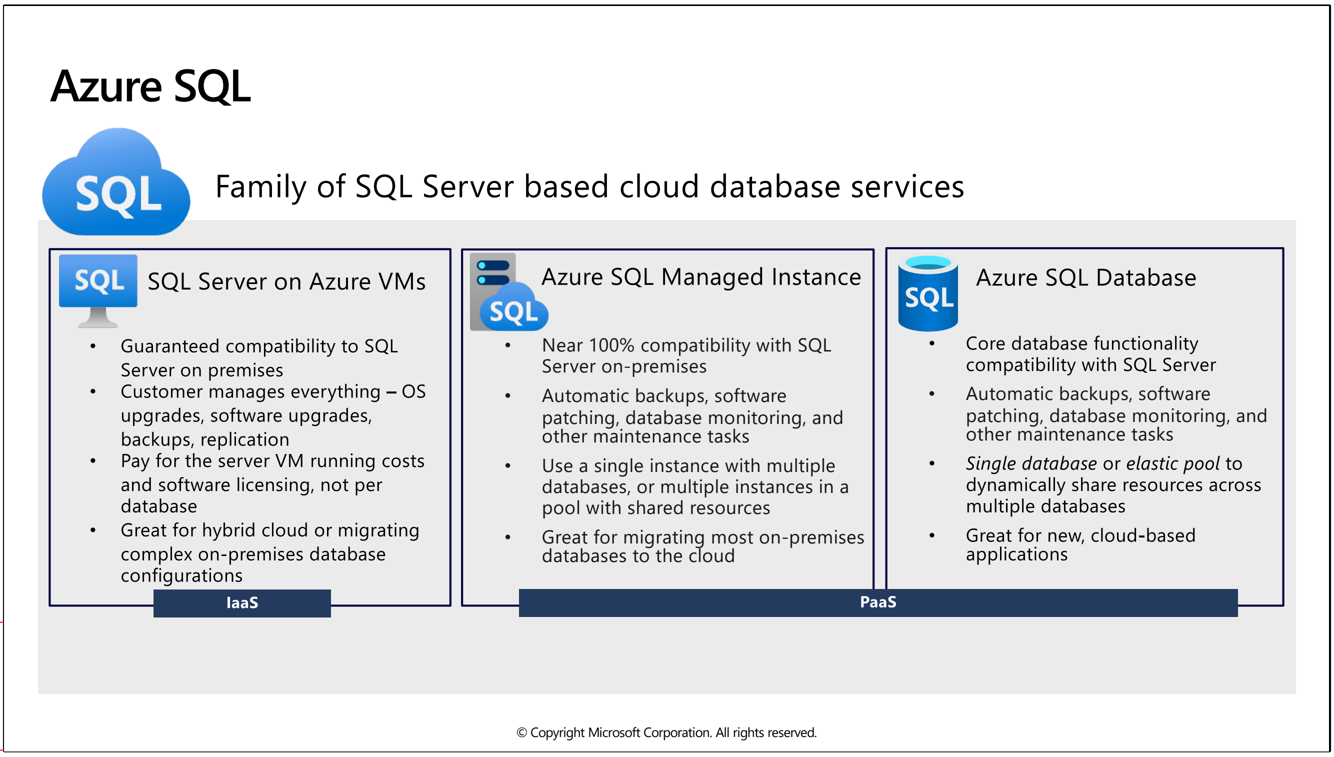
Black R0 G0 B0 Hex #000000

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



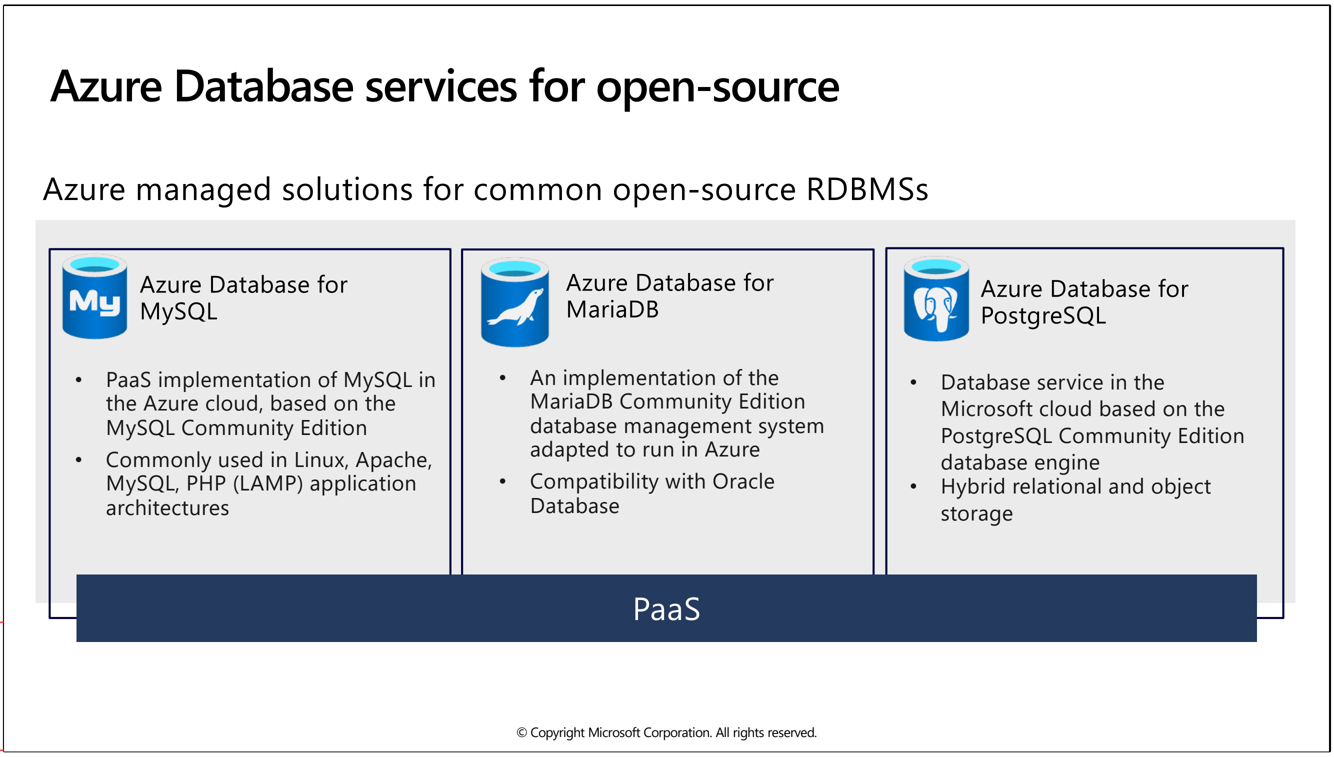
Note that this slide does not cover Azure SQL Edge, which is a SQL Server-based service for edge computing – predominantly for Internet-of-things (IoT) scenarios.

Azure SQL is the generic term used to describe a family of Azure relational database services that are based on Microsoft SQL Server. SQL Server is an industry-leading relational database management system (RDBMS) that is used in on-premises solutions by some of the biggest organizations in the world. The Azure SQL services are based on the same database engine, making them a great solution for organizations that want to migrate existing on-premises databases to the cloud; as well as new applications that are designed as cloud-based from conception.

- **SQL Server on Azure Virtual Machines** is an infrastructure-as-a-service (IaaS) solution in which a full instance of SQL Server is installed in a virtual machine that is hosted in Azure. This makes it a good candidate for migration projects, where 1:1 compatibility with an existing on-premises SQL Server instance is required or for hybrid scenarios with a mix of cloud-based and on-premises databases that must maintain compatibility. Because it's an IaaS solution, you have full control of the configuration of the database; which also means you have responsibility to manage administrative tasks – just as you would for a SQL Server instance in your own data center. Costs for the service are based on SQL Server licensing and the cost of running the VM in Azure.
- **Azure SQL Managed Instance** is a platform-as-a-service (PaaS) service that enables you to pre-provision compute resources and deploy several individual SQL Server managed instances up to your pre-provisioned compute level. Core administrative tasks are automated while providing a high-degree of compatibility with on-premises SQL Server. You can choose to deploy a single managed instance that supports multiple databases, or you can create a pool of instances that share underlying infrastructure resources for cost-efficiency. SQL Managed Instance is a great choice for most migration scenarios, where you need to move an on-premises SQL Server database to the cloud with minimal changes.
- **Azure SQL Database** is another platform-as-a-service (PaaS) solution that offers the lowest-cost Azure SQL option. You have minimal administrative control over the service beyond creating the database schema, importing and exporting data, and configuring access controls. Azure SQL Database enables you to deploy a single database or an *elastic pool* that shares resources across multiple databases. Azure SQL Database is a great choice for new applications that require a low-cost relational data store

with minimal administrative overhead.

The list is in decreasing order of administrative control/responsibility and cost. SQL Server on a VM is the most expensive option; but allows you greater control over server and database configuration. However, you also have full responsible for server maintenance and management. Azure SQL Database is the lowest cost option, but supports fewer configuration options, Most database maintenance other than access controls is automated for you. SQL Managed Instance offers a balance of cost, administrative control, and maintenance automation.



MySQL started life as a simple-to-use open-source database management system. It is commonly used in *Linux*, *Apache*, *MySQL, and PHP* (LAMP) stack apps.

MariaDB is a newer database management system, created by the original developers of MySQL. The database engine has since been rewritten and optimized to improve performance. MariaDB offers compatibility with Oracle Database (another popular commercial database management system).

PostgreSQL is a hybrid relational-object database. You can store data in relational tables, but a PostgreSQL database also enables you to store custom data types, with their own non-relational properties.

Lab: Provision Azure relational database services

In this lab, you will provision, configure and query an Azure SQL Database.

1. Start the virtual machine for this lab
   or go to the exercise page at https://aka.ms/dp900-sql-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.

Note that the exercise on Microsoft Learn includes options to provision MySQL and PostgreSQL databases. Encourage students to go back to this module and try these options later, but all students should provision and query Azure SQL Database in this lab.

2: Knowledge check

Which deployment option offers the best compatibility when migrating an existing SQL Server on-premises solution?

- ❑ Azure SQL Database (single database)
- ❑ Azure SQL Database (elastic pool)
- ✔ Azure SQL Managed Instance

Which of the following statements is true about Azure SQL Database?

- ✔ Most database maintenance tasks are automated
- ❑ You must purchase a SQL Server license
- ❑ It can only support one database

Which database service is the simplest option for migrating a LAMP application to Azure?

- ❑ Azure SQL Managed Instance
- ✔ Azure Database for MySQL
- ❑ Azure Database for PostgreSQL

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

Microsoft Azure

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