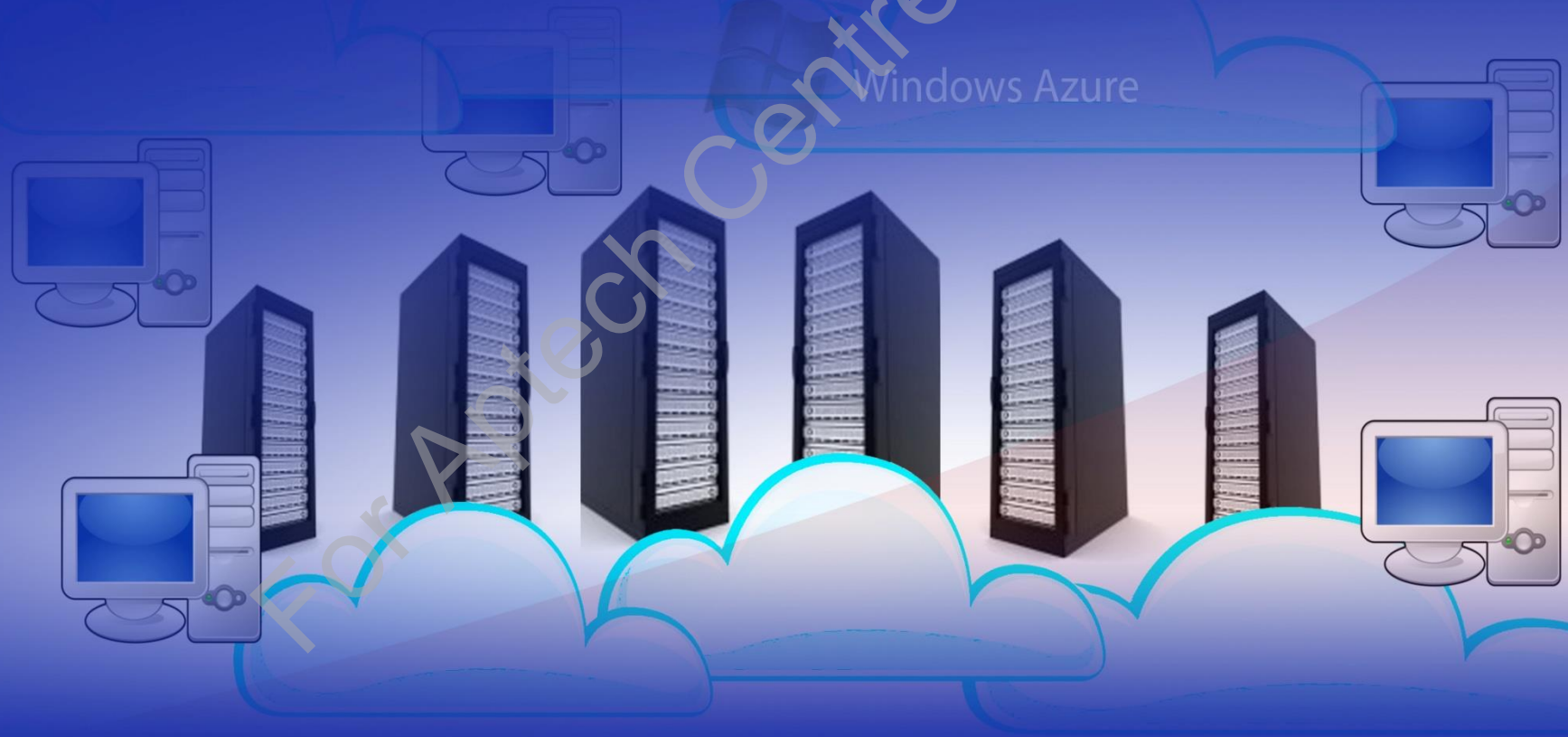


Enterprise Application Development Using Windows Azure and Web Services

Session 11

Windows Azure SQL Database



Learning Objectives 1-2



- Describe SQL Database
- List and describe the features of SQL Database
- Explain the working of SQL database
- Compare between SQL Azure and SQL Server
- Describe the procedure to create a Cloud database
- Explain the process to configure Firewall

Learning Objectives 2-2



- Explain the process to manage the cloud database
- Describe connecting to SQL Database using SQL Server Management Studio
- Describe connecting to SQL Database using ADO.NET
- Explain the process to query SQL Database

Need for SQL Database

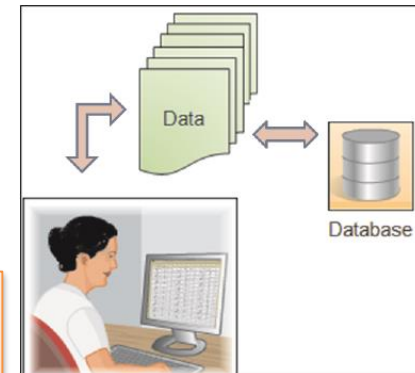
- ❑ SQL Server 2012, as an RDBMS, had a few merits and demerits:

Merits

- Full control of the databases and the server.
- Latency does not cause an issue.
- Hosting requirements can be customized.

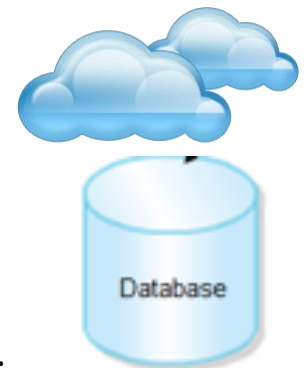
Demerits

- Scaling of databases running in local instances of SQL Server restricted with the hardware resources.
- Hardware management of the host also a tedious task.



- ❑ Solution to these issues:

- Adopt a cloud environment that can host SQL databases.
- Resources can be scaled up or down as and when required.
- Hardware is taken care of by Microsoft teams across different datacenters.



Introduction to SQL Database

❑ The fundamental goal of SQL Database is to:

- Provide database services to the users such as cloud computing services rather than local services.
- Enable developers to work on SQL Azure in the same manner as they would when working on the local instance.
- Provide deployment of databases is in the cloud, which is hosted in the Microsoft datacenters across the globe.

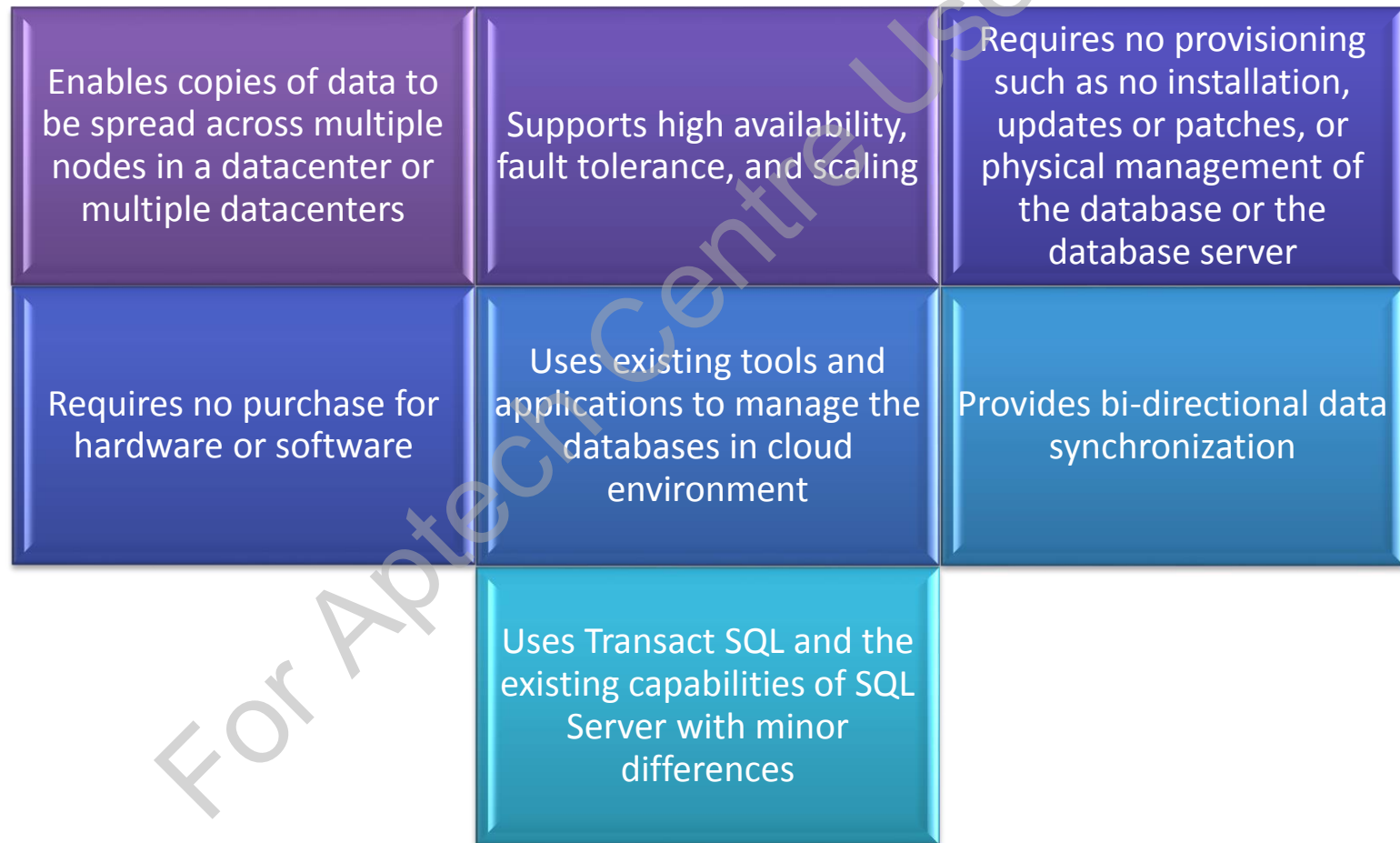
❑ Developer can perform the following tasks in the cloud environment:

- Connect to the SQL Database using SQL Server Management Studio (SSMS)
- Use libraries, such as ADO.NET to write code and deploy it
- Perform replication



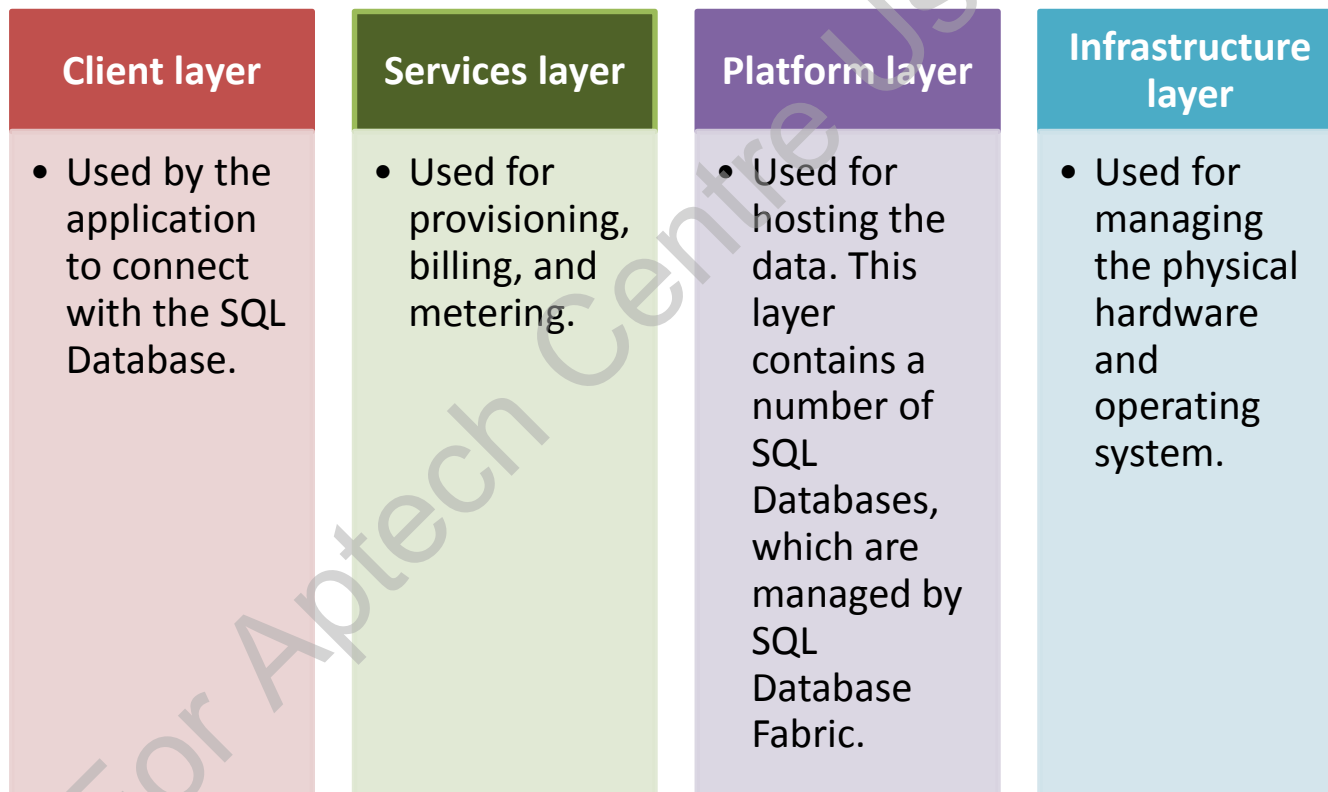
Features of SQL Database

❑ Following are the key features of SQL Database:



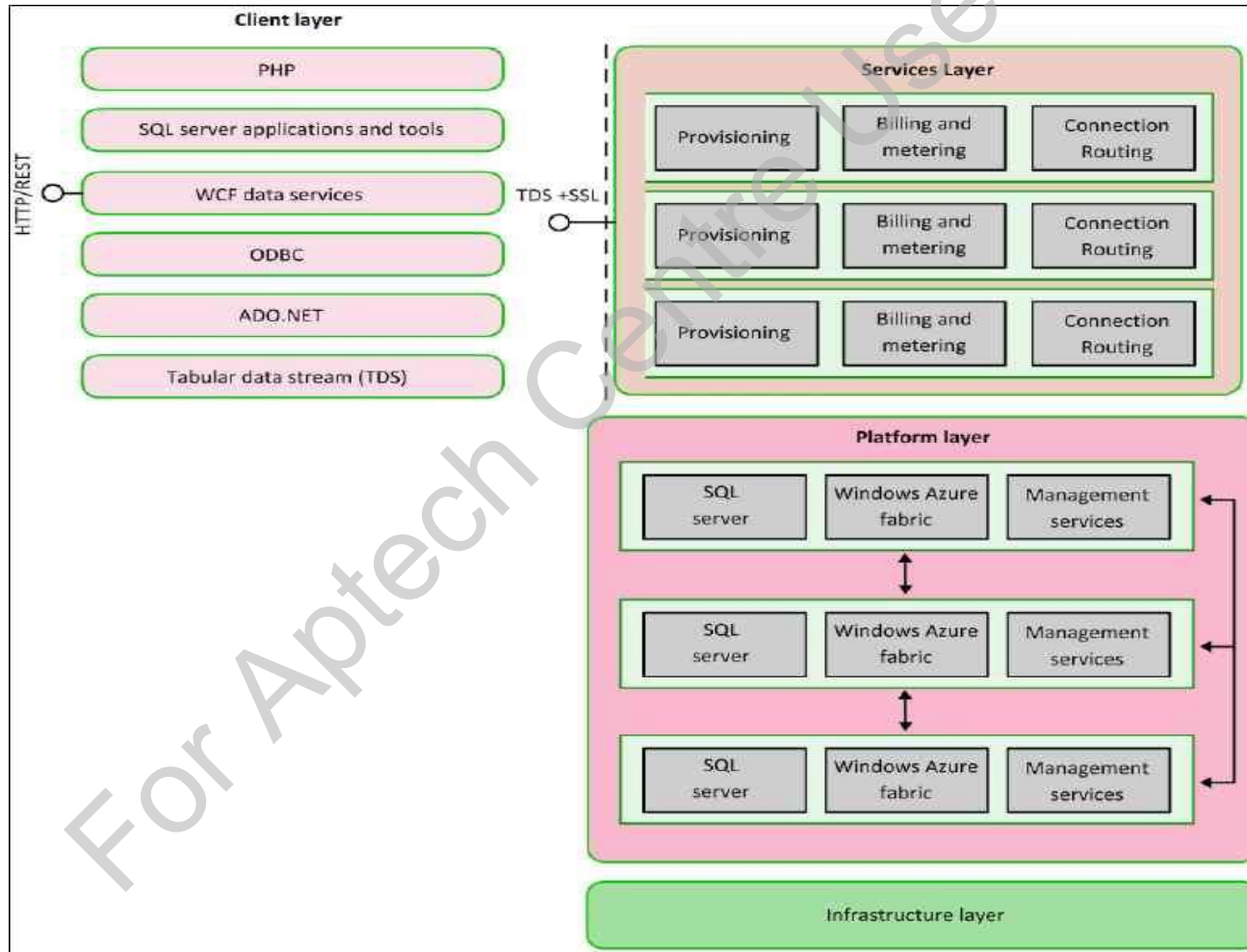
Working of SQL Database 1-3

- Following are the four layers on which the SQL Database is built:



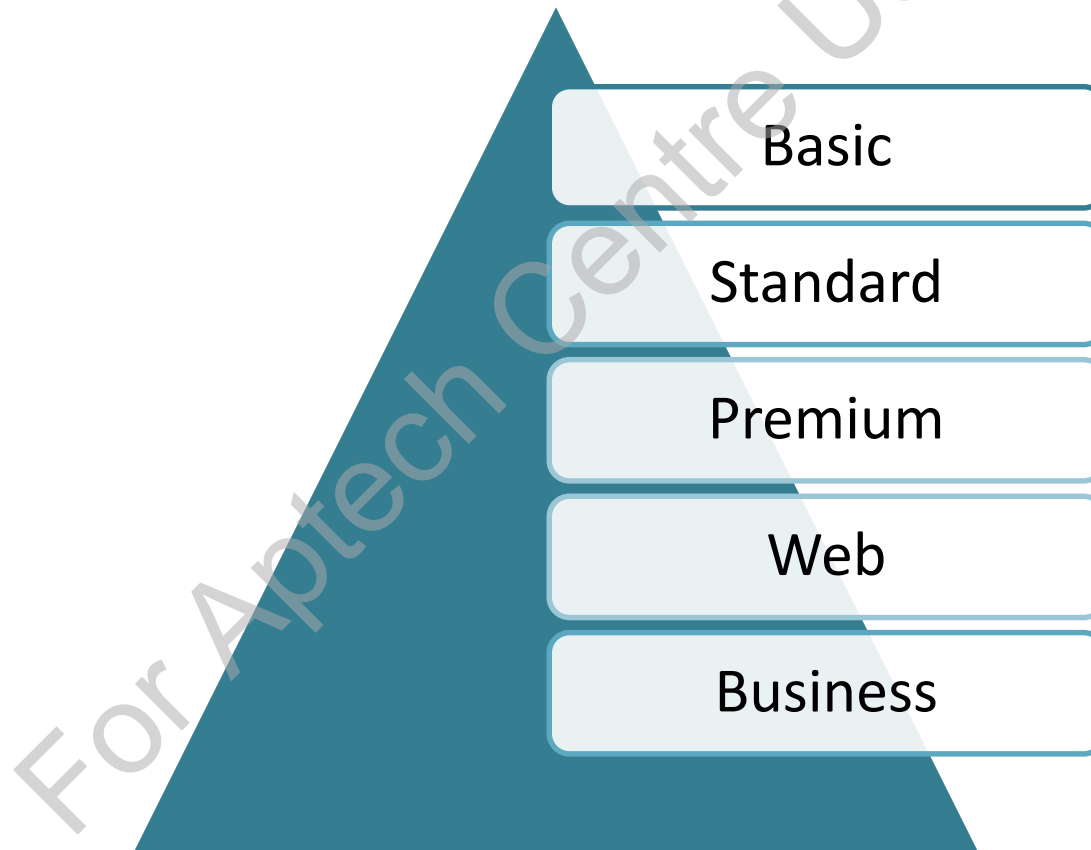
Working of SQL Database 2-3

- Following figure displays the SQL Database architecture:



Working of SQL Database 3-3

- ❑ Following are the five editions of SQL Database where each of the editions has a unique set of capabilities:



Comparing SQL Azure and SQL Server

- ❑ Following table lists the primary differences between SQL Database and SQL Server:

Feature	SQL Database	SQL Server
Location	Databases can be spread across multiple nodes and even datacenters across geographical locations.	Databases are typically co-located within either the same node or the location. The location of the database is controlled by the developer.
Latency	The latency is in milliseconds and depends on the location of the databases that may be on different nodes, datacenters, or geographical locations.	The latency is in sub-milliseconds.
Hardware	Hardware is not a limitation. The databases are typically replicated across number of nodes. Hardware failure is hidden from the end users.	The hardware requirement may increase or decrease depending on the size and use of the database. Manual intervention for the hardware maintenance is required.
Resource Capacity	Individual application databases have their dedicated hardware. Upgrade or downgrade may be required from time to time depending on the usage.	A developer can create premium and secondary databases. Premium databases have dedicated hardware even if a single system hosts multiple databases.
Tables	Tables are designed and optimized for large data access but do not support referential integrity. The data is stored in non-relational format.	Tables support referential integrity. The data is stored in the relational format.

Using SQL Database

❑ Before a developer can use a SQL Database:

- It needs to be created.

- Developer must have an account on the <http://azure.microsoft.com> portal.
- After the account is created, then the developer can continue with creating the database.

Creating a Cloud Database 1-4

Following are the steps to create a cloud database:

Step 1

- Connect to the portal
<http://manage.windowsazure.com>.
- The figure displays the left pane of the **Management Portal** after successful login.



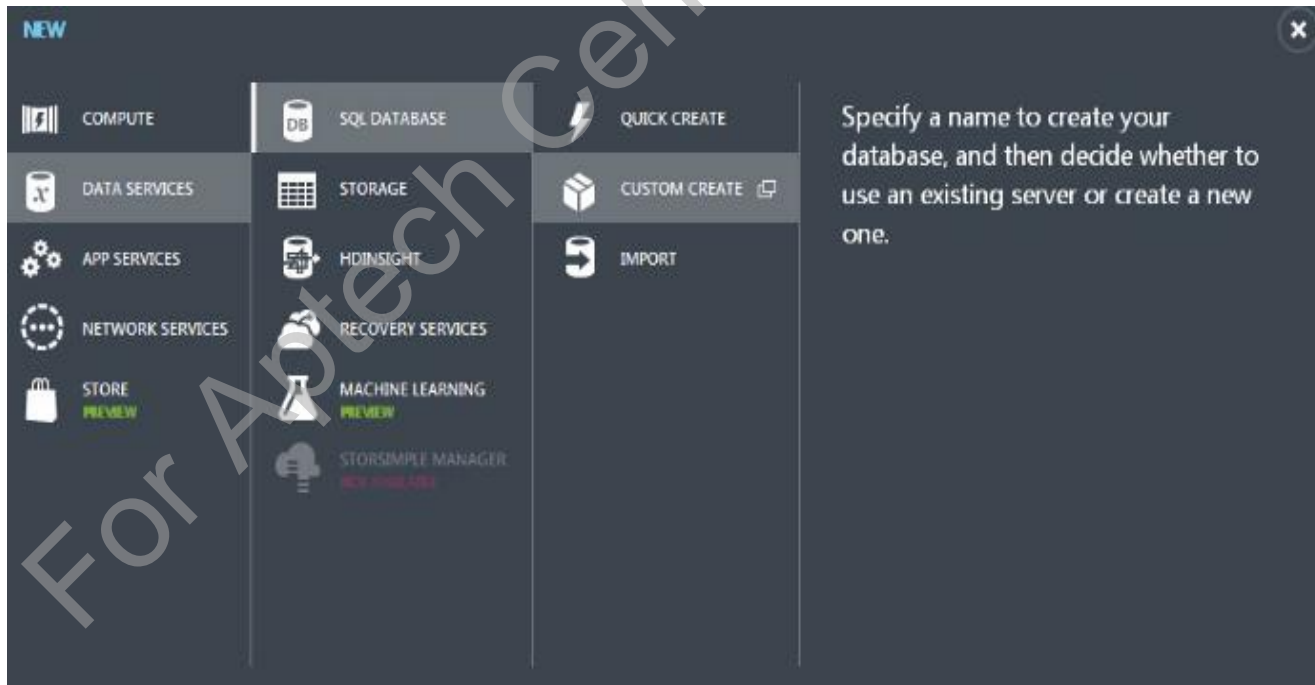
Creating a Cloud Database 2-4

Step 2

- Click **New** at the bottom of the browser screen. A list is displayed on the screen.

Step 3

- From the list, click **SQL DATABASE** and then, click **CUSTOM CREATE** as shown in the following figure:



Creating a Cloud Database 3-4

Step 4

- On the **NEW SQL DATABASE – CUSTOM CREATE** page, enter information, such as database name, size, and type. Click the arrow indicating next.

Step 5

- On the **CREATE SERVER** page, enter the login details.
- Ensure that the **Allow Windows Azure Services to access the server** check box is selected.
- The figure shows the **NEW SQL DATABASE – CUSTOM CREATE** page with login details:

NEW SQL DATABASE - CUSTOM CREATE

Server Settings

LOGIN NAME

LOGIN PASSWORD

LOGIN PASSWORD CONFIRMATION

REGION

West US

☒ Allow Windows Azure Services to access the server

1

Creating a Cloud Database 4-4

Step 6

- Click the check mark at the bottom to complete the database creation process, as shown in the following figure:

NEW SQL DATABASE - CUSTOM CREATE

Server Settings

LOGIN NAME
SQLAdmin

LOGIN PASSWORD
••••••••

LOGIN PASSWORD CONFIRMATION
••••••••

REGION
West US

☒ Allow Windows Azure Services to access the server

1

Navigation buttons: back and checkmark (highlighted in a red box)

Firewall Rules 1-3

- ❑ Following steps are used to configure the firewall through the portal:

Step
1

- From the left pane on the portal, click **SQL Databases** and click **Servers**.

Step
2

- Select the database server that was created in the previous task. The server specific information is displayed.

Step
3

- Click **Configure**. A new page is displayed with the **allowed ip addresses** settings.

Firewall Rules 2-3

Step 4

- Add the IP address in the **CURRENT CLIENT IP ADDRESS** text box and then click **ADD TO THE ALLOWED IP ADDRESS** link as shown in the following figure:

testdb

DASHBOARD DATABASES CONFIGURE HISTORY

allowed ip addresses

CURRENT CLIENT IP ADDRESS 115.252.136.49 ADD TO THE ALLOWED IP ADDRESSES.

RULE NAME	START IP ADDRESS	END IP ADDRESS
ClientIPAddress_2014-07-15_15:38:27	115.252.136.49	115.252.136.49

Firewall Rules 3-3

Step
5

- Click **SAVE**. Following figure shows the result of saving the configuration:

testdb

[DASHBOARD](#) [DATABASES](#) [CONFIGURE](#) [HISTORY](#)

allowed ip addresses ?

CURRENT CLIENT IP ADDRESS ADD TO THE ALLOWED IP ADDRESSES. +

ClientIpAddress_2014-07-15_15:30:27	115.252.136.49	115.252.136.49
<input type="text" value="RULE NAME"/>	<input type="text" value="START IP ADDRESS"/>	<input type="text" value="END IP ADDRESS"/>

allowed services ?

WINDOWS AZURE SERVICES

MANAGE **SAVE** DISCARD 1 1 1 ?

Managing the Cloud Database

❑ Microsoft SQL Server Management Studio (SSMS) is:

- Is an integrated environment that can be used for managing local and cloud databases.
- Is installed with the SQL Server on a local system.
- Can be downloaded in Express edition for free (SQL Server 2012 Management Studio Express).

Connecting through SSMS

- ❑ Following steps are used to connect to SQL Database through SSMS:

Step 1

- Click **Start**, select **All Programs**, select **Microsoft SQL Server 2012**, and then select **Server Management Studio**. The **Connect to Server** dialog box is displayed.

Step 2

- Enter the fully-qualified server name such as **servername.database.windows.net**.

Step 3

- From the **Authentication** drop-down list, select **SQL Server Authentication**.

Step 4

- In the **Login** text box, enter the login name using the syntax **login@servername**.

Step 5

- In the **Password** text box, enter the password that you specified in the portal when creating your server.

Step 6

- Click **Connect**.

Connecting to SQL Database through ADO.NET

- ❑ ADO.NET has a limitation to connect to a single database. Switching between databases is not supported.
- ❑ Following code allows you to connect to the SQL Database named, aptechdb using ADO.NET within an application:

```
SqlConnectionStringBuilder connectionBuilder= new  
    SqlConnectionStringBuilder();  
.  
.  
.  
connectionBuilder.DataSource =  
    "dbserver.database.windows.net";  
connectionBuilder.InitialCatalog = "master";  
connectionBuilder.Encrypt = true;  
connectionBuilder.UserID = "alex";  
connectionBuilder.Password= "aptech123";
```

Creating an SQL Database Instance

- ❑ A database named aptechdb can be created using the following code:

```
using (SqlConnection objConn = new
SqlConnection(connectionBuilder.ToString()))
{
    using (SqlCommand objCommand = objConn.CreateCommand())
    {
        objConn.Open();
        // Create the aptechdb database
        string cmdText = String.Format("CREATE DATABASE {0}",
        aptechdb);
        objCommand.CommandText = cmdText;
        objCommand.ExecuteNonQuery();
        objConn.Close();
    }
}
```

- ❑ After creating the aptechdb database, a developer can connect to it and perform various operations such as create, modify, or delete tables and records.

Querying SQL Database 1-2

- ❑ Once the connection to the SQL Database is successfully established, the developer can perform various query operations.
- ❑ Following steps show how to insert and retrieve data:

Step 1

- Go to the account home page on the portal.

Step 2

- Click **aptechdb** to select it and then, click **Manage** at the bottom of the page. This displays the Management Portal for SQL Database, which is different from the Azure Management Portal.

Step 3

- Enter the login name and password to login to the **aptechdb** database.

Step 4

- Click **New Query** in the ribbon. An empty query window is displayed.

Querying SQL Database 2-2

Step 5

- Copy the following code and paste it into the query window.

```
-- Create the Course table.  
IF NOT EXISTS (SELECT * FROM sys.objects  
WHERE object_id = OBJECT_ID(N'[dbo].[Course]')  
AND type in (N'U'))  
BEGIN  
CREATE TABLE [dbo].[Course] (  
[CourseID] [int] NOT NULL,  
[Name] [nvarchar](50) NOT NULL,  
[Fees] [money] NOT NULL,  
[StartDate] [datetime] NOT NULL,  
CONSTRAINT [PK_Course] PRIMARY KEY CLUSTERED  
(  
[CourseID] ASC)WITH (IGNORE_DUP_KEY = OFF)  
) WITH (IGNORE_DUP_KEY = OFF)  
)  
END;  
GO
```

Summary 1-2

- ❑ SQL Database is Microsoft's implementation of SQL Server as a cloud service.
- ❑ SQL Database offers high availability and scalability.
- ❑ An SQL Database instance must be created before it can be used by a developer.
- ❑ An SQL Database instance is created through the Management Portal for SQL Azure.

Summary 2-2

- ❑ Microsoft SQL Server Management Studio (SSMS) is an integrated environment that can be used for managing local and cloud databases.
- ❑ A developer can also use ADO.NET to connect to the SQL Database similar to the connection through SSMS.
- ❑ After creating an SQL Database instance, a developer can perform various operations such as create, modify, or delete tables and records.