Assignment

|  |  |
| --- | --- |
| Course Code: | IT4A (MIS) |
| Course Name: | Managing Infrastructure Services |
| Assignment: | Final Project Part 1: SQL Server 2019 |

|  |  |
| --- | --- |
| Textbook: | Petkovic, D. (2020). Microsoft SQL Server 2019: A Beginner's Guide (7e). McGraw Hill. |
| Software: | Windows Server 2019, SQL Server 2019 |

Materials and Resources

Assignment Description

|  |
| --- |
| This project reviews the key topics related to database server administration in SQL Server 2019. Ensure that the following steps are performed on a **clean installation of Windows Server 2019**.  You are required to install a new SQL Server 2019 database server that will be used by developers within your organization. All server maintenance will be your responsibility and you must communicate/coordinate with the company developer in order to create the appropriate structures and objects on the database server. You will connect to the SQL Server 2019 computer as the Administrator Windows account (or sa account), but all other connections will be done using a separate account. |

Assignment Steps

|  |
| --- |
| Activity Steps |
| 1. Install a new virtual machine with Windows Server 2019. Ensure that this new virtual machine is called **DBSERVER** and is a domain controller for the domain ***yourname*.com** (where ***yourname*** is your last name**). 1 mark** |
| 1. Install a default instance of SQL Server 2019. **1 mark** Ensure that Windows and SQL logins can be used. Give the **sa** account a password of **Secret555**.**1 mark** |
| 1. Allow TCP connections on TCP port 1433 and ensure that all services (including the SQL Server Agent) are set to automatically start each time the server is booted. **2 marks** |
| 1. Set up a fake Database Mail profile and account that can be used to send email (the details and name are irrelevant).**1 mark** |
| 1. Configure an operator for yourself that lists your triOS email address and availability times (be creative here).**1 mark** |
| 1. Create a folder in your virtual machine called **C:\Project**. Next, open a new query in SSMS. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\1.rpt**.   SELECT @@SERVERNAME AS 'Server Name'  SELECT DEFAULT\_DOMAIN()[DomainName]  SELECT \* FROM sys.dm\_server\_services  SELECT \* FROM sys.tcp\_endpoints  SELECT \* FROM msdb.dbo.sysmail\_account  SELECT \* FROM msdb.dbo.sysoperators |
| 1. Create a new database called **Project**. Ensure that the location of database files optimizes database performance (do not use the default file location). **1 mark** |
| 1. Create a SQL login for the developer on the server called **woot** with a password of **Secret555**. **1 mark** |
| 1. Create a SQL user in the Project database called **woot** for the developer and ensure that the developer can perform any functions within the Project database. **2 marks** |
| 1. Open a new query in SSMS. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\2.rpt**.   SELECT db.name AS Name, type\_desc AS Type,Physical\_Name AS Location  FROM sys.master\_files mf  INNER JOIN sys.databases db  ON db.database\_id = mf.database\_id  SELECT \* FROM sys.sql\_logins WHERE name = 'woot'  USE Project  GO  SELECT \* FROM sys.sysusers WHERE hasdbaccess = 1  SELECT \* FROM sys.database\_role\_members  SELECT \* FROM sys.database\_principals WHERE name = 'db\_owner' |
| 1. Although the developers will be creating most tables in the database, they have asked you to create a table for them in the Project database called **Employee** in a new schema called **HR**. A sample of the type of information that this table will store is shown below – use this information to set up the appropriate fields and constraints (ensure that you use the best data types). Ensure that there is a primary key (you can add a field for one) and that there is a clustered index on the primary key field.**6 marks**  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Employee ID | Last Name | First Name | Title | Dept Name | Office # | Ext | Date Hired | Salary | | H101 | Benson | Max | General Manager | Housekeeping | 501 | 3410 | 26-Nov-87 | $42,000.00 | | H102 | Martinez | Sandra | North Wing Supervisor | Housekeeping | 503 | 3413 | 5-May-91 | $32,330.00 | | H103 | Erickson | Gregory | South Wing Supervisor | Housekeeping | 503 | 3411 | 17-Aug-91 | $31,800.00 | | H104 | Lew | Judy | East Wing Supervisor | Housekeeping | 503 | 3412 | 20-Jan-92 | $33,920.00 | | R221 | Valentine | Brian | Host | Restaurant | 101 | 7601 | 27-Jan-91 | $25,040.00 | | R222 | Poland | Carole | Hostess | Restaurant | 101 | 7602 | 10-Apr-89 | $29,000.00 | | R234 | Yukish | Gary | Asst. Chef | Restaurant | 167 | 7603 | 16-Sep-89 | $34,000.00 | | R344 | White | Suki | Head Chef | Restaurant | 167 | 7604 | 1-Jul-87 | $45,000.00 | | R455 | Jackson | Sue | Asst. Chef | Restaurant | 167 | 7605 | 11-Oct-88 | $36,450.00 | | S503 | Lee | Frank | Athletic Director | Sports & Rec | 314 | 6801 | 24-Jun-90 | $43,860.00 | | S504 | Harui | Roger | Asst. Athletic Director | Sports & Rec | 315 | 6803 | 10-Feb-91 | $38,390.00 | | S505 | Bourne | Stephanie | Asst. Athletic Director | Sports & Rec | 315 | 6804 | 13-May-92 | $36,800.00 | | S606 | Schatz | Bethany | Entertainment Director | Sports & Rec | 320 | 6805 | 15-Mar-92 | $32,500.00 | | S607 | Chen | John | Children's Director | Sports & Rec | 320 | 6806 | 19-Dec-92 | $32,500.00 | |
| 1. Create 2 separate non-clustered indexes on the last name and office fields. **2 marks** |
| 1. Open a new query in SSMS. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\3.rpt**.   USE Project  GO  SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS where TABLE\_NAME='Employee'  SELECT \* FROM HR.Employee  SELECT \* FROM sys.indexes  WHERE object\_id =  (SELECT object\_id FROM sys.objects WHERE name = 'Employee') |
| 1. Create a maintenance plan that rebuilds all indexes on this table and updates all statistics each Sunday at 5:00a.m. Any problems should be emailed to you. Execute your maintenance plan to ensure that it works. Finally, right-click your maintenance plan and choose **Modify** to view a flowchart of its configuration. Take a screenshot of that configuration and save it as **C:\Project\4.png**. **2 marks** |
| 1. Create a SQL login and user called **project-access** that has the password **Secret555**. Ensure that **project-access** only has SELECT permissions to the HR.Employee table. **2 marks** |
| 1. Create a SQL login and user called **project-update** that has the password **Secret555**. Ensure that **project-update** only has permissions to add, update and delete records in the HR.Employee table. **2 marks** |
| 1. Open a new query in SSMS. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\5.rpt**.   SELECT \* FROM sys.sql\_logins  WHERE name = 'project-access' OR name = 'project-update'  USE Project  GO  SELECT  (  dp.state\_desc + ' ' +  dp.permission\_name collate latin1\_general\_cs\_as +  ' ON ' + '[' + s.name + ']' + '.' + '[' + o.name + ']' +  ' TO ' + '[' + dpr.name + ']'  ) AS GRANT\_STMT  FROM sys.database\_permissions AS dp  INNER JOIN sys.objects AS o ON dp.major\_id=o.object\_id  INNER JOIN sys.schemas AS s ON o.schema\_id = s.schema\_id  INNER JOIN sys.database\_principals AS dpr  ON dp.grantee\_principal\_id=dpr.principal\_id  WHERE 1=1 AND o.name IN ('Employee')  ORDER BY dpr.name |
| 1. Create a trigger that prevents the database developer from deleting tables accidentally in the Project database. Save the query you used to create this trigger to **C:\Project\6.sql**.**1 mark** |
| 1. Manually perform a full backup of your Project database to **C:\Project\Project.bak**. **1 mark** |
| 1. Ensure that full backups of the Project database are performed every day at 10:00p.m. using a maintenance plan. Execute your maintenance plan to ensure that it works. Finally, right-click your maintenance plan and choose Modify to view a flowchart of its configuration. Take a screenshot of that configuration and save it as **C:\Project\7.png**. **2 marks** |
| 1. Create a SQL Query file that can quickly be executed to restore the previous night’s backup (without losing any transactions since the previous night’s backup). Perform a sample restore using this query to make sure it works. Finally, save the query to **C:\Project\8.sql**. **2 marks** |
| 1. Ensure that the Project database is automatically checked for errors and then shrunk every Monday and Thursday at 2:00am using a single maintenance plan. Execute your maintenance plan to ensure that it works. Finally, right-click your maintenance plan and choose **Modify** to view a flowchart of its configuration. Take a screenshot of that configuration and save it as **C:\Project\9.png**.**2 marks** |
| 1. Check your database for errors. Perform a sample repair in single user mode and return to multiuser mode when finished. Save the commands you performed in a file called **C:\Project\10.sql**.**2 marks** |
| 1. Configure the Performance Data Warehouse to monitor the key areas of your SQL server (this data should be stored in a database called **PDW**). After a period of time (such as 1 day), generate the three reports (Disk Usage, Server Activity, Query Statistics). Right-click each report and choose **Export > PDF** and save it to C:\Project\ using the default filename. You should have 3 reports in this folder:   **C:\Project\Disk Usage Summary<date>.pdf 1 mark**  **C:\Project\Query Statistics History<date>.pdf 1 mark**  **C:\Project\Server Activity History<date>.pdf 1 mark** |
| 1. Enable your system as a publisher and distributor (use the default database name of **distribution** for your distributor database). Next, configure snapshot replication to back up the Employee table in the Project database to a new database you create called **Project2**. **2 marks** Although Project2 would normally be on another SQL server, you will create Project2 on your own SQL server instance. |
| 1. Open a new query in SSMS. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\11.rpt**.   USE distribution  GO  EXEC sp\_replmonitorhelppublication   1. Create a 2nd (named) instance of SQL on your computer called **ARFA** that allows network access. Ensure that the SQL Server Agent is started and set to start automatically at boot time. **2 marks** |
| 1. Create a new database on your ARFA instance called **HeadOffice** (use appropriate locations for the database and log files). **1 mark** |
| 1. Perform a bulk export of the data within that table that you created within your project database to a text file called **C:\Project\BulkRecords.txt**. Following this, bulk import the data into the same table within the HeadOffice database on your ARFA instance. **2 marks** |
| 1. Open a new query in SSMS that is connected to your ARFA instance. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\12.rpt**.   SELECT @@SERVERNAME AS 'Server Name'  SELECT \* FROM sys.dm\_server\_services  SELECT \* FROM sys.tcp\_endpoints  SELECT \* FROM HeadOffice.HR.Employee  SELECT db.name AS Name, type\_desc AS Type,Physical\_Name AS Location  FROM sys.master\_files mf  INNER JOIN sys.databases db  ON db.database\_id = mf.database\_id |
| 1. Set up database mirroring for your **Project** database between your default and 2nd instance that uses manual failover. Perform 2 manual failovers. **2 marks** |
| 1. Open a new query in SSMS that is connected to your default instance. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\13a.rpt**. Next, repeat this step while connected to your ARFA instance but save the results to **C:\Project\13b.rpt**.   SELECT \* FROM sys.database\_mirroring  SELECT \* FROM sys.database\_mirroring\_endpoints |
| 1. Remove the mirror to restore your **Project** database to its original state. **1 mark** |
| 1. Create a new database on your 2nd instance called **BranchOffice** (use appropriate locations for the database and log files). **1 mark** |
| 1. Open a new query in SSMS that is connected to your ARFA instance. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\14.rpt**.   SELECT db.name AS Name, type\_desc AS Type,Physical\_Name AS Location  FROM sys.master\_files mf  INNER JOIN sys.databases db  ON db.database\_id = mf.database\_id |
| 1. Set up replication so that the table that you created in the Project database on your default instance is replicated to the BranchOffice database on your ARFA instance using merge replication. **1 mark** |
| 1. Add a new record to your BranchOffice database and ensure that it is added to your Project database. When finished, open the Replication Monitor and take a screenshot of your replication status (expand your publication) and save it as **C:\Project\15.png**. |
| 1. Use the SQL Server Profiler to capture the existing performance of your Project database in a single trace file called **C:\Project\PerformanceTrace.trc** (no file rollover). Next, use the DTA to analyze the trace file and produce index recommendations (these will be poor recommendations because this is not a production SQL server). Take a screenshot of the recommendations and save it as **C:\Project\16.png**. **2 marks**      1. Add appropriate performance counters to the Performance Console in Windows and create a baseline for later use in HTML (web page) format. Save the baseline as **C:\Project\Baseline.htm**. **1 mark** |
| 1. Create alerts to ensure that you are notified by email when the following occurs on your default instance (research as necessary): **7 marks**  * Deadlocks are being created * The SQL memory usage is not performing well * There are too many connections to your SQL server * A database is offline * A database has failed * Log shipping has failed * Replication has failed |
| 1. Open a new query in SSMS that is connected to your default instance. Right-click the white background of the query window and click **Results To > Results to File**. Next, execute the following queries in this window. When prompted to save the results to a file, choose **C:\Project\17.rpt**.   SELECT \* FROM msdb.dbo.sysalerts |
| 1. Compress the C:\Project folder into a ZIP file called ***yourname*.zip** and upload this file into Brightspace for it to be graded by your instructor. Your C:\Project folder should have the following contents: **Total: 60 marks**   1.rpt 2.rpt 3.prt 4.png 5.rpt 6.sql 7.png  8.sql 9.png 10.sql 11.rpt 12.rpt 13a.rpt 13b.rpt  14.rpt 15.png 16.png 17.rpt  Project.bak PerformanceTrace.trc  Baseline.htm BulkRecords.txt  Disk Usage Summary<date>.pdf  Query Statistics History<date>.pdf  Server Activity History<date>.pdf |

Rubric

Marks are awarded as outlined in the Assignment Steps/Instructions.   
The total is out of 60 marks.