UNIVERSITY of GREENWICH
Alliance with FPT Education

BTEC

Higher Nationals in Computing

Unit 04: Database Design and Development
ASSIGNMENT 2

Learner's name: **TRINH THI DIEU HUYEN**

Assessor name: **PHAN MINH TAM**

Class: **GCS0801B.2**

ID: **GDD18606**

Subject code: **1622**

Assignment due: Assignment submitted:

ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	**Unit 04: Database Design & Development**		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name		**Student ID**	
Class		**Assessor name**	Phan Minh Tam
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	

Grading grid

P2	P3	P4	P5	M2	M3	M4	M5	D2	D3

☼ Summative Feedback:	☼ Resubmission Feedback:	
Grade:	**Assessor Signature:**	**Date:**
Signature & Date:		

ASSIGNMENT 2 BRIEF

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number	Unit 04: Database Design & Development		
Assignment title			
Academic Year	2019		
Unit Tutor	Phan Minh Tam		
Issue date		**Submission date**	
IV name and date			

Submission Format:

Format: This assignment is an Individual assignment and specifically including 1 document:

You must use font *Calibri size 12, set number of the pages and use multiple line spacing at 1.3. Margins must be: left: 1.25 cm; right: 1 cm; top: 1 cm and bottom: 1 cm.* The reference follows Harvard referencing system. The recommended word limit is *2.000-2.500 words*. You will not be penalized for exceeding the total word limit. The cover page of the report has to be the Assignment front sheet 1.

Submission Students are compulsory to submit the assignment in due date and in a way requested by the Tutors. The form of submission will be a soft copy posted on http://cms.greenwich.edu.vn/

Note: The Assignment *must* be your own work, and not copied by or from another student or from books etc. If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference your sources, using the Harvard style. Make sure that you know how to reference properly, and that understand the guidelines on plagiarism. *If you do not, you definitely get fail*

Unit Learning Outcomes:

LO1 Use an appropriate design tool to design a relational database system for a substantial problem.

LO2 Develop a fully functional relational database system, based on an existing system design.

LO3 Test the system against user and system requirements.

LO4 Produce technical and user documentation

Assignment Brief and Guidance:

You are employed as a Database Developer for a large IT consultancy company. The company has been approached by FPT university which is expanding due to the growth of the number of students. FPT is currently facing difficulties in dealing with managing the university. It decided to develop several academic systems to manage the university easier including: **Online Library system, Student Grading System, Attendance System, CMS System, Scheduling System, Enrolment Systems, and so on.**

You are tasked to select one of those systems to develop database for FPT university. Your tasks are to:

Work with FPT to find out about current requirements for each system

Analyze the requirements and produce clear statements of user and system requirements.

Design a relational database system using appropriate design tools and techniques

Develop a fully functional relational database system, based on an existing system design.

Test the system against user and system requirements.

Produce technical and user documentation

Part 1 (assignment 1)

Before you start the development process, your manager has asked you to produce a report for the CEO of FPT, containing:

1. Clear statements of user and system requirements.

2. The design of the relational database system using appropriate design tools and techniques. It should contain at least four interrelated tables.

You would prefer to produce a more detailed document, so you will produce a comprehensive design for a fully functional system which will include interface and output designs, data validations and cover data normalisation.

Your manager would lik on the report your assessment of the effectiveness of the design in relation to user and system requirements.

Part 2 (Assignment 2)

Once the designs have been accepted by your manager you have been asked to:

1. develop the database system using evidence of user interface, output and data validations and querying across multiple tables.

You want to include more than just the basics so you will implement a fully functional database system which will include system security and database maintenance features.

You have decided to implement a query language into the relational database system.

The developed system will be demonstrated to your manager.

Your manager has asked you to include in the report:

2. Assessing whether meaningful data has been extracted through the use of query tools to produce appropriate management information.

3. Evaluating the effectiveness of the database solution in relation to user and system requirements, and suggest improvements.

4. Once the system has been developed, you will test the system and your manager will complete a witness statement indicating how your tests are performing against user and system requirements.

You will produce a brief report assessing the effectiveness of the testing, including an explanation of the choice of test data used.

Lastly you will produce technical and user documentation which will be given to the company.

You want to provide some graphical representations for ease of reference in the technical guide, so you have decided to produce a technical and user documentation for a fully functional system, including diagrams showing movement of data through the system, and flowcharts describing how the system works.

<table>
<tr><th colspan="3">Learning Outcomes and Assessment Criteria</th></tr>
<tr><th>Pass</th><th>Merit</th><th>Distinction</th></tr>
<tr><td colspan="2">LO2 Develop a fully functional relational database system, based on an existing system design.</td><td rowspan="6">LO2 & 3
D2 Evaluate the effectiveness of the database solution in relation to user and system requirements, and suggest improvements.</td></tr>
<tr><td>P2 Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables.

P3 Implement a query language into the relational database system.</td><td>M2 Implement a fully functional database system which includes system security and database maintenance.

M3 Assess whether meaningful data has been extracted through the use of query tools to produce appropriate management information.</td></tr>
<tr><td colspan="2">LO3 Test the system against user and system requirements</td></tr>
<tr><td>P4 Test the system against user and system requirements.</td><td>M4 Assess the effectiveness of the testing, including an explanation of the choice of test data used.</td></tr>
<tr><td colspan="3">LO4 Produce technical and user documentation</td></tr>
<tr><td>P5 Produce technical and user documentation.</td><td>M5 Produce technical and user documentation for a fully functional system, including ER Diagram and normalization statements and describing how the system works.</td><td>D3 Assess any future improvements that may be required to ensure the continued effectiveness of the database system.</td></tr>
</table>

Table of Contents

ASSIGNMENT 2 ANSWERS

LO2 Develop a fully functional relational database system, based on an existing system design.

P2 Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables.

1. Physical Design.

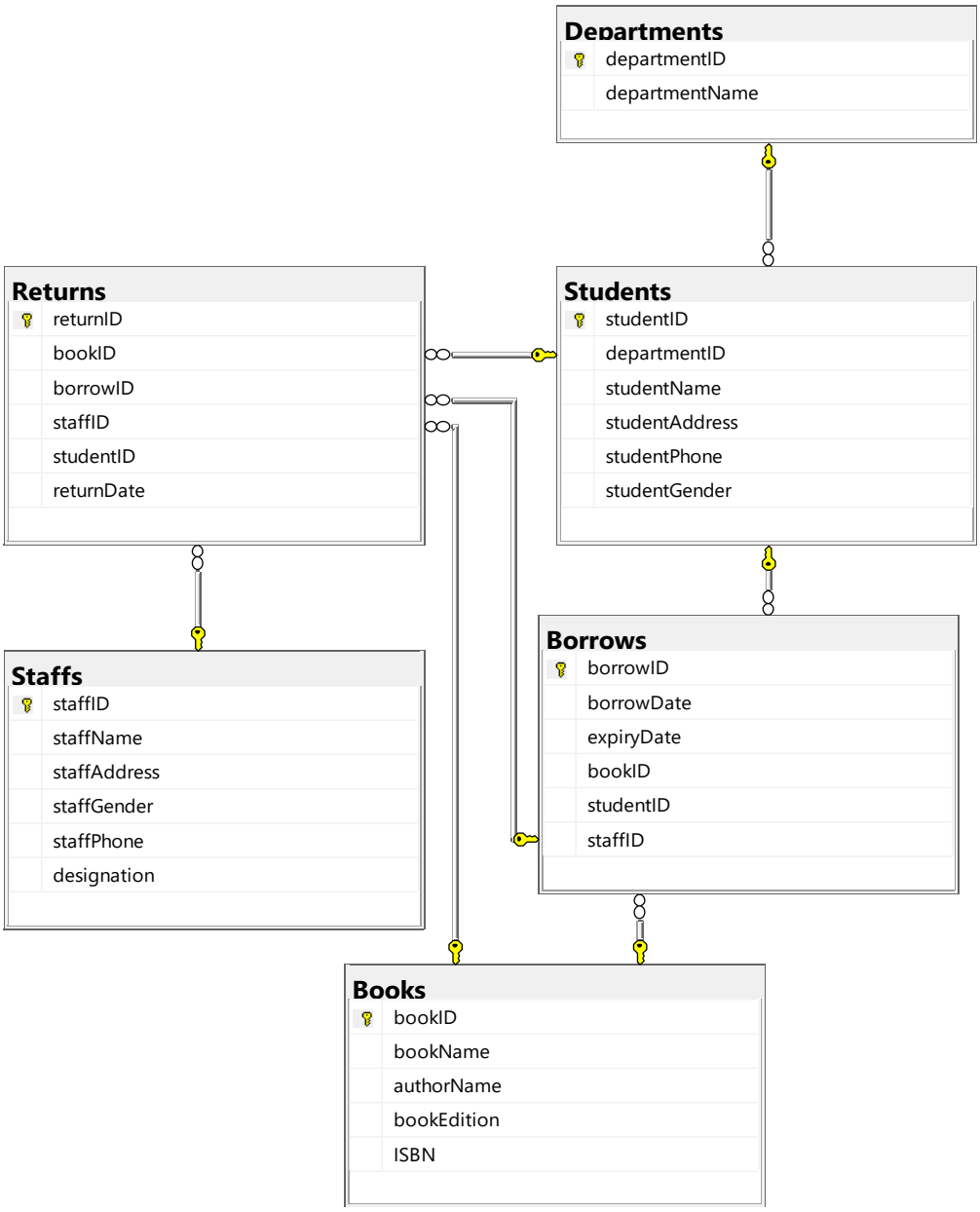
1.1. Code create database and table.

```
--Create new database.
CREATE DATABASE OnlineLibrarySystem
GO

--Use database.
USE OnlineLibrarySystem
GO

--Create table, primary key, foreign key
--Create books table.
CREATE TABLE Books(
    bookID      VARCHAR(10) PRIMARY KEY,
bookName    NVARCHAR(255),
authorName  NVARCHAR(255),
bookEdition NVARCHAR(255),

--Use Unique.
ISBN        SMALLINT UNIQUE
)
GO

--Create borrows table.
CREATE TABLE Borrows(
    borrowID   VARCHAR(10) PRIMARY KEY,
borrowDate DATETIME,
expiryDate DATETIME,
bookID     VARCHAR(10) NOT NULL,
```

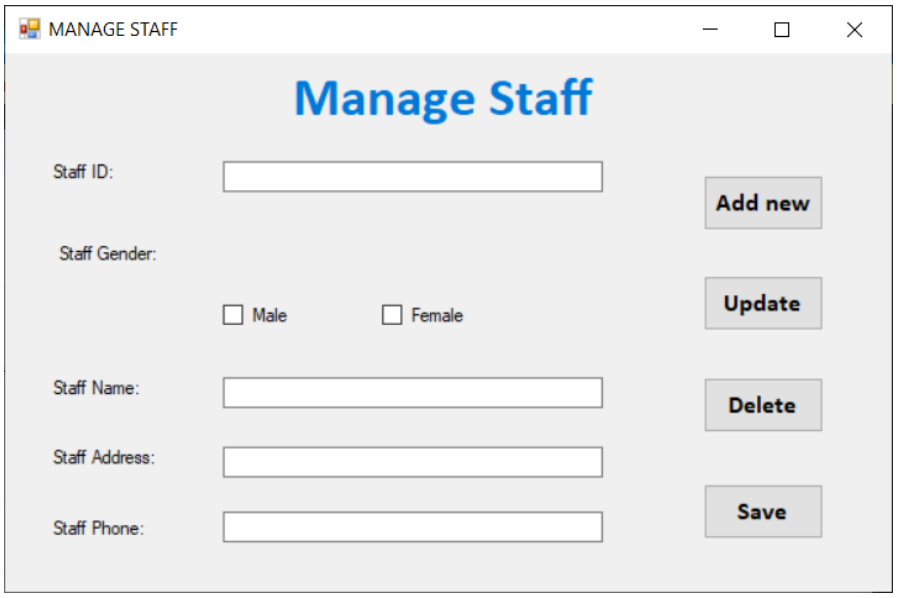
```
studentID  VARCHAR(10) NOT NULL

--Create Foreign key.
FOREIGN KEY (studentID) REFERENCES Students(studentID),
FOREIGN KEY (bookID)    REFERENCES Books(bookID),
)
GO

--Create students table.
CREATE TABLE Students(
    studentID VARCHAR(10) PRIMARY KEY,
departmentID VARCHAR(10) NOT NULL,
studentName NVARCHAR(255),
studentAddress NVARCHAR(255),
studentPhone INT,

--Use Check.
studentGender VARCHAR(10) CHECK (studentGender IN ('Male','Female'))

--Use Unique.
CONSTRAINT unique_studentPhone UNIQUE(studentPhone)

--Create Foreign key.
FOREIGN KEY (departmentID) REFERENCES Departments(departmentID),
)
GO

--Create departments table.
CREATE TABLE Departments(
    departmentID   VARCHAR(10) PRIMARY KEY,
departmentName NVARCHAR(50)
)
GO

--Create returns table.
CREATE TABLE [Returns](
    returnID   VARCHAR(10) PRIMARY KEY,
```

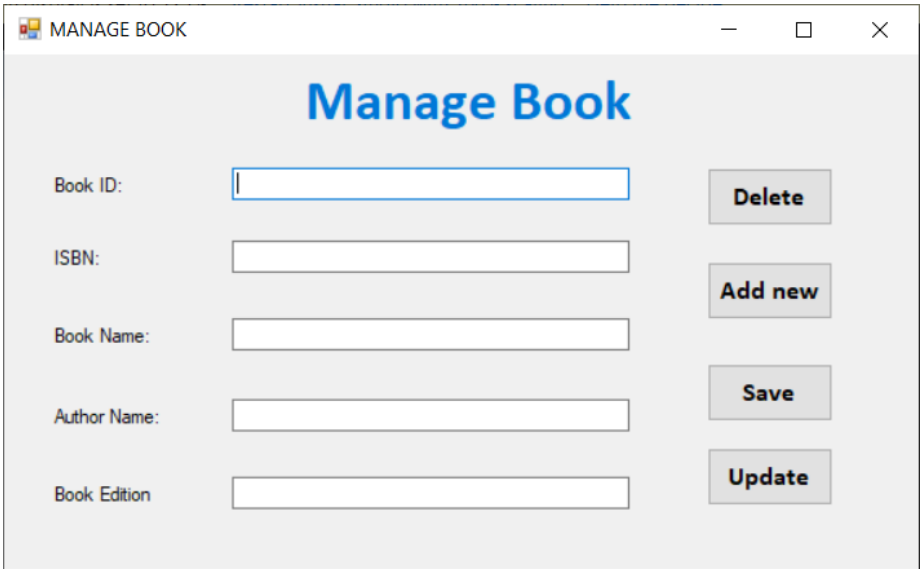
```
bookID     VARCHAR(10) NOT NULL,
borrowID   VARCHAR(10) NOT NULL,
staffID    VARCHAR(10) NOT NULL,
studentID  VARCHAR(10) NOT NULL,
returnDate DATETIME

--Create Foreign key.
FOREIGN KEY (bookID)    REFERENCES Books(bookID),
FOREIGN KEY (borrowID)  REFERENCES Borrows(borrowID),
FOREIGN KEY (staffID)  REFERENCES Staffs(staffID),
FOREIGN KEY (studentID) REFERENCES Students(studentID),
)
GO
--Create staffs table.
CREATE TABLE Staffs(
    staffID VARCHAR(10) PRIMARY KEY,
staffName NVARCHAR(50),
staffAddress NVARCHAR(50),
staffGender VARCHAR(10) CHECK (staffGender IN ('Male','Female')),
staffPhone INT,
designation NVARCHAR(50),

--Use Unique.
UNIQUE (staffPhone,designation)
)
GO
--Add CHECK in Borrows.
ALTER TABLE dbo.Borrows ADD CONSTRAINT CK_Date
CHECK (borrowDate < expiryDate AND expiryDate > borrowDate)

--Add UNIQUE in Departments.
ALTER TABLE dbo.Departments
ADD CONSTRAINT unique_department UNIQUE(departmentName);

--Add StaffID into table borrows.
ALTER TABLE dbo.Borrows
ADD staffID VARCHAR(10)
```

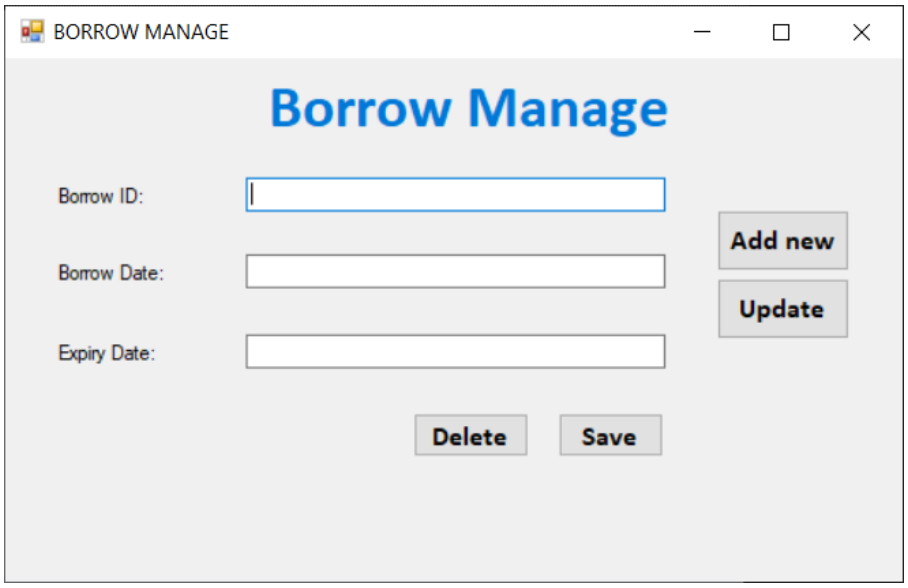
1.2. Database diagram.

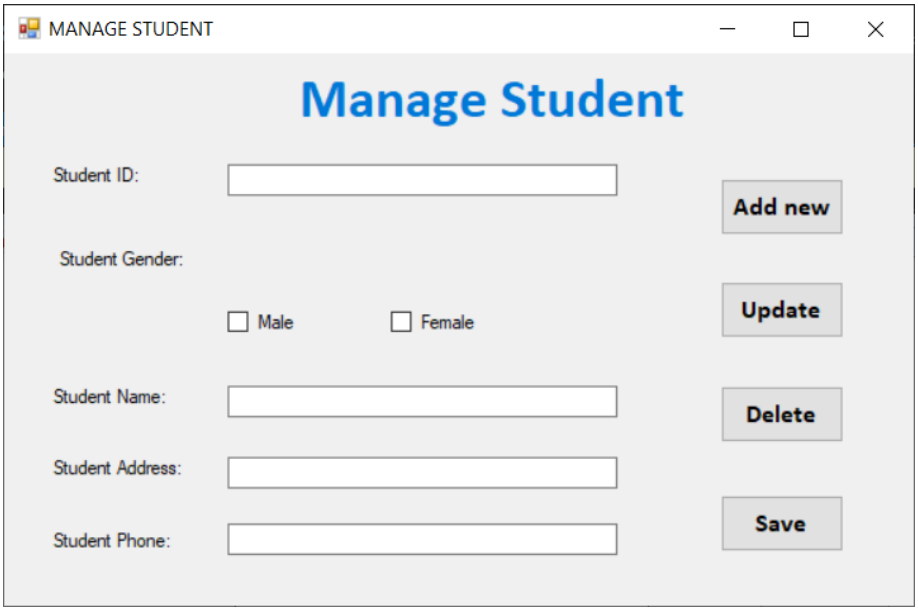


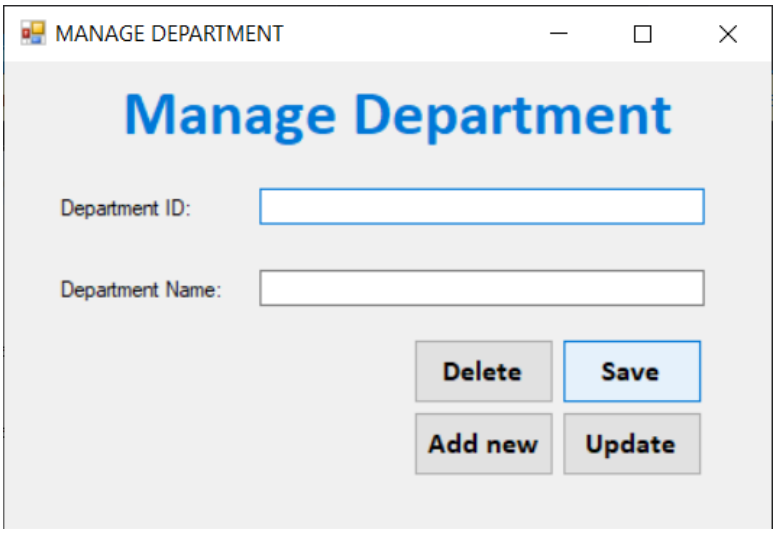
2. Graphic User Interface.

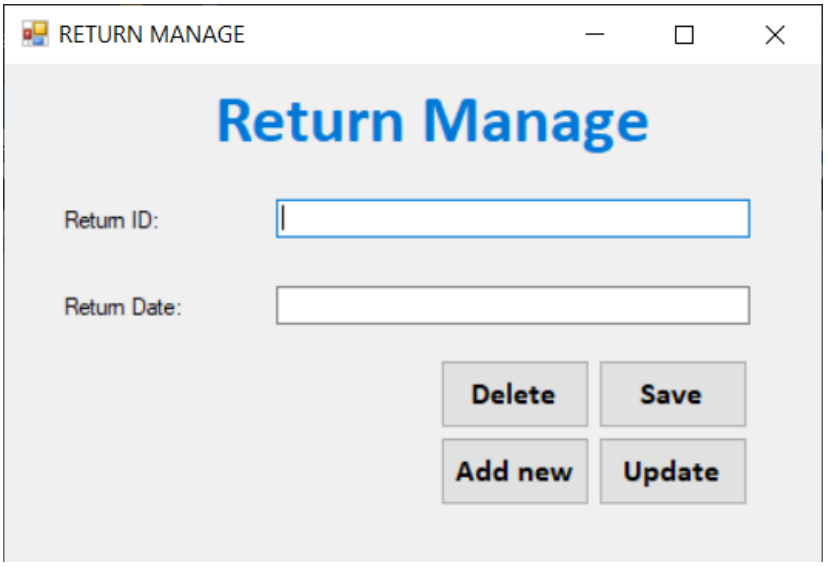





BORROW MANAGE
Borrow Manage
Borrow ID:
Borrow Date:
Expiry Date:
Add new
Update
Delete
Save


MANAGE STUDENT
Manage Student
Student ID:
Student Gender:
Male
Female
Student Name:
Student Address:
Student Phone:
Add new
Update
Delete
Save


MANAGE DEPARTMENT
Manage Department
Department ID:
Department Name:
Delete
Save
Add new
Update


RETURN MANAGE
Return Manage
Return ID:
Return Date:
Delete
Save
Add new
Update

P3 Implement a query language into the relational database system.

1. Execute the INSERT query to input data into all tables in the database.

```
--INSERT: input data
--Table books
INSERT INTO Books(bookID,bookName,authorName,bookEdition,ISBN)
VALUES ('K2002','Jane Eyre','Charlotte Bronte','First',2589)
GO
INSERT INTO Books(bookID,bookName,ISBN) VALUES
('K2012','.Net & C#',1579),
('K2013','Java & php',1679),
('K2014','AI-My life',1570)
GO
INSERT INTO Books VALUES
('K2015','Love or you','Mino Song','Second',3003),
('K2016','Fiance','Mino Song','Second',2412),
('K2017','Ah yeah','Winner','First',5790)
GO
INSERT INTO Books VALUES ('K2003','The Adventures of Huckleberry
Finn','Mark Twain','Second',2590)
GO
INSERT INTO Books VALUES ('K2004','Of Mice and Men','John
Steinbeck','Third',2365)
GO
INSERT INTO Books VALUES ('K2005','Paradise Lost','John
Milton','Fourth',8354)
GO
INSERT INTO Books VALUES ('K2006','The Communist Manifesto','Karl
Marx','First',2889)
GO
INSERT INTO Books VALUES ('K2007','Frankenstein','Mary Wollstonecraft
Shelley','Second',6754)
GO
INSERT INTO Books VALUES ('K2008','The Epic of
Gilgamesh','Anonymous','First',7908)
GO
```

```
INSERT INTO Books VALUES ('K2009','Lolita','Vladimir
Nabokov','Second',7872)
GO
INSERT INTO Books VALUES ('K2010','Tuesdays with Morrie','Mitch
Albom','First',2679)
GO
INSERT INTO Books VALUES ('K2011','The Story of a Seagull and the Cat Who
Taught Her to Fly','Luis Sepulveda','First',9964)
GO

--Table departments.
INSERT INTO Departments VALUES ('D1','Information Technology')
GO
INSERT INTO Departments VALUES ('D2','Math')
GO
INSERT INTO Departments VALUES ('D3','Art')
GO
INSERT INTO Departments VALUES ('D4','Commerce')
GO
INSERT INTO Departments VALUES ('D5','Islamic')
GO
INSERT INTO Departments VALUES ('D6','Urdu')
GO
INSERT INTO Departments VALUES ('D7','Education')
GO
INSERT INTO Departments VALUES ('D8','Economics')
GO
INSERT INTO Departments VALUES ('D9','Physics')
GO
INSERT INTO Departments VALUES ('D10','Food Science')
GO

--Table students.
INSERT INTO Students VALUES ('S100','D2','Smith
Caleb','Alabama',2051001010,'Male')
GO
```

```
INSERT INTO Students VALUES ('S101','D5','Blake
Johnson','California',2090101010,'Female')
GO
INSERT INTO Students VALUES ('S112','D1','Williams
Berel','Hawaii',2081010101,'Female')
GO
INSERT INTO Students VALUES ('S113','D8','Jones
Cayden','Alabama',2012020200,'Male')
GO
INSERT INTO Students VALUES ('S105','D9','Brown
Cedric','California',2130200202,'Female')
GO
INSERT INTO Students VALUES ('S107','D3','Davis Berwin','New
York',2101111111,'Male')
GO
INSERT INTO Students VALUES ('S116','D4','Miller
Chanan','Alabama',2101020100,'Female')
GO
INSERT INTO Students VALUES ('S117','D6','Wilson
Cameron','Alaska',1075746382,'Female')
GO
INSERT INTO Students VALUES ('S118','D10','Moore
Abraham','California',1187264585,'Male')
GO
INSERT INTO Students VALUES ('S119','D7','Taylor
Calev','California',2004856243,'Female')
GO

--Table borrows.
INSERT INTO Borrows VALUES
('B001','07/01/2019','07/02/2019','K2004','S113')
GO
INSERT INTO Borrows VALUES
('B002','10/03/2019','10/04/2019','K2007','S119')
GO
INSERT INTO Borrows VALUES
('B003','10/03/2019','10/04/2019','K2003','S101')
```

```
GO
INSERT INTO Borrows VALUES
('B004','07/01/2019','07/02/2019','K2007','S101')
GO
INSERT INTO Borrows VALUES
('B005','12/02/2019','03/01/2020','K2004','S112')
GO
INSERT INTO Borrows VALUES
('B006','01/01/2019','03/02/2019','K2005','S113')
GO
INSERT INTO Borrows VALUES
('B007','07/01/2019','07/02/2019','K2006','S119')
GO
INSERT INTO Borrows VALUES
('B008','12/02/2019','06/06/2020','K2008','S118')
GO
INSERT INTO Borrows VALUES
('B009','08/03/2019','12/06/2019','K2004','S105')
GO
INSERT INTO Borrows VALUES
('B010','09/01/2019','10/05/2019','K2004','S116')
GO

--Table staffs.
INSERT INTO Staffs VALUES ('TF01','Kamboj Zane','New
York','Male',2111572010,'Librairan')
GO
INSERT INTO Staffs VALUES ('TF02','Dua
Padraig','Hawaii','Male',1051044480,'Sweeper')
GO
INSERT INTO Staffs VALUES ('TF03','Smith
Roderick','California','Female',1151811010,'Manager')
GO
INSERT INTO Staffs VALUES ('TF04','Thind
Roderick','Alabama','Male',2022221010,'Bookeeper')
GO
```

```
INSERT INTO Staffs VALUES ('TF05','Bhatia
Phillip','Hawaii','Female',2051012220,'Naibqasid')
GO
INSERT INTO Staffs VALUES ('TF06','Kamboh
Oliver','Alaska','Male',1151011010,'Accountant')
GO
INSERT INTO Staffs VALUES ('TF07','Smith
Frederick','Alabama','Male',2057711010,'Manage')
GO
INSERT INTO Staffs VALUES ('TF08','Thind
Gideon','Hawaii','Female',2001011010,'Secretary')
GO
INSERT INTO Staffs VALUES ('TF09','Shetty
Caleb','Alaska','Female',2052221010,'Assistant')
GO
INSERT INTO Staffs VALUES ('TF010','Smith
Ezra','Alabama','Male',2051221010,'Guard')
GO

--Table returns.
INSERT INTO [Returns] VALUES
('RT001','K2004','B005','TF010','S112','01/01/2020')
GO
INSERT INTO [Returns] VALUES
('RT002','K2003','B003','TF010','S101','04/10/2019')
GO
INSERT INTO [Returns] VALUES
('RT003','K2004','B001','TF01','S113','02/07/2019')
GO
INSERT INTO [Returns] VALUES
('RT004','K2008','B008','TF02','S118','01/01/2020')
GO
INSERT INTO [Returns] VALUES
('RT005','K2007','B004','TF01','S101','02/07/2019')
GO
INSERT INTO [Returns] VALUES
('RT006','K2004','B009','TF02','S105','09/09/2019')
```

```
GO
INSERT INTO [Returns] VALUES
('RT007','K2004','B010','TF01','S116','01/10/2019')
GO
INSERT INTO [Returns] VALUES
('RT008','K2006','B007','TF03','S119','02/07/2019')
GO
INSERT INTO [Returns] VALUES
('RT009','K2007','B002','TF01','S119','04/10/2019')
GO
INSERT INTO [Returns] VALUES
('RT010','K2005','B006','TF01','S113','01/02/2019')
GO
```

2. Execute the UPDATE query to edit data in all tables in the database.

```
--UPDATE: edit data.
UPDATE Borrows
SET staffID = 'TF01'
WHERE bookID = 'K2004'
GO
--
UPDATE [Returns]
SET staffID = 'TF01'
WHERE bookID = 'K2004'
GO
--
UPDATE Students
SET studentPhone = studentPhone + 2
GO
--
UPDATE Staffs
SET staffGender = 'Female'
GO
--
UPDATE Staffs
SET staffGender = 'Male'
WHERE staffAddress = 'Alaska'
```

```
GO
--
UPDATE [Returns]
SET staffID = 'TF03'
WHERE returnDate = '02/07/2019'
GO
--
UPDATE Departments
SET departmentName = 'Marketing'
WHERE departmentID = 'D4'
GO
--
UPDATE Borrows
SET bookID = 'K2005'
WHERE studentID = 'S113'
GO
--
UPDATE Books
SET authorName = 'JUJU'
WHERE bookID = 'K2013'
GO
--
UPDATE Borrows
SET staffID = 'TF05'
WHERE studentID = 'S101'
GO
--
UPDATE [Returns]
SET staffID = 'TF05'
WHERE studentID = 'S101'
GO
```

3. Execute the DELETE query to remove data in all tables in the database.

```
--DELETE: remove data.
DELETE FROM [Returns]
WHERE staffID = 'TF03'
GO
```

```
--
DELETE FROM [Returns]
WHERE bookID = 'K2004'
GO
--
DELETE FROM Borrows
WHERE borrowID = 'B001'
GO
--
DELETE FROM Borrows
WHERE studentID = 'S118'
GO
--
DELETE FROM [Returns]
WHERE bookID = 'K2007'
GO
--
DELETE FROM Students
WHERE departmentID = 'D3'
GO
--
DELETE FROM Borrows
WHERE StaffID = 'TF05'
GO
--
DELETE FROM Students
WHERE studentID = 'S100'
GO
--
DELETE FROM Borrows
WHERE StaffID = 'TF010'
GO
--
DELETE FROM [Returns]
WHERE StaffID = 'TF02'
GO
```

4. Execute the SELECT query to show data in all tables in the database.

```
--SELECT: show data.
SELECT * FROM Students
GO
SELECT * FROM Staffs
GO
SELECT * FROM Books
GO
SELECT * FROM Borrows
GO
SELECT * FROM [Returns]
GO
SELECT * FROM Departments
GO

--List of year borrowed book less than or equal to 2020 and have
studentID = 'S113'
SELECT*
FROM dbo.Borrows
WHERE YEAR(borrowDate) <= 2020
AND Borrows.studentID='S113'

--List of other expiring year 2019
SELECT*
FROM dbo.Borrows
WHERE YEAR(expiryDate) <> 2019

--List of other expiring year 20120
SELECT*
FROM dbo.Borrows
WHERE YEAR(expiryDate) != 2020 --'!=' the same '<>'

--List of return book dates is greater than or equal to 7 and has
studentID = 'S119'
SELECT*
FROM dbo.[Returns]
```

```
WHERE DAY(returnDate) >= 7
AND [Returns].studentID='S119'

--List of returning book with month equal to 4
SELECT*
FROM dbo.[Returns]
WHERE MONTH(returnDate) = 4

--Show all Staff whose names start as Smith
SELECT * FROM Staffs
WHERE staffName LIKE 'Smith%'
GO
--
SELECT studentID, studentName, studentAddress FROM Students
GO
--
SELECT * FROM Staffs
WHERE staffAddress IN ('Alaska','Alabama')
AND staffName LIKE 'Smith%'
GO
--
SELECT studentID, COUNT(*) AS TotalBorrows
FROM Borrows
GROUP BY studentID
HAVING COUNT(*) >=2
GO
--
SELECT TOP(2) bookID, COUNT(*) AS TotalBorrows
FROM Borrows
GROUP BY bookID
ORDER BY COUNT(*) DESC
GO
--
SELECT bookID, COUNT(*) AS TotalBorrows
FROM Borrows
GROUP BY bookID
ORDER BY COUNT(*)
```

```
GO
--
SELECT * FROM Staffs
WHERE staffName LIKE '[^KS]%'
GO
--
SELECT * FROM Staffs
WHERE staffName LIKE '[DT]%G'
GO
--
SELECT * FROM Staffs
WHERE staffName LIKE '_H%K'
GO
--
SELECT * FROM Staffs
WHERE staffName LIKE '%Smith%'
GO
--
SELECT * FROM [Returns]
WHERE returnDate BETWEEN '01/01/2019' AND '02/03/2019'
GO
--
SELECT * FROM Students
WHERE studentAddress = 'Alaska' OR
      studentAddress = 'Hawaii' OR
      studentAddress = 'New York'
GO
```

5. Perform some advanced queries (advanced queries, queries across multiple tables, etc.)

```
--Queries across multiple tables--
--List of students who return books in detail
SELECT*FROM dbo.Students AS stu, dbo.Staffs AS sta, dbo.Books AS bok,
dbo.[Returns] AS rt
WHERE stu.studentID=rt.studentID
AND sta.staffID=rt.staffID
AND bok.bookID=rt.bookID
GO
```

```
--Combine two table are Borrows and Students according condition. If
which side without data will NULL
SELECT * FROM
dbo.Borrows FULL OUTER JOIN dbo.Students
ON Borrows.studentID=Students.studentID

--Combination each record of the Staffs table with all records of the
Borrows table
SELECT * FROM
dbo.Staffs CROSS JOIN dbo.Borrows
--List of students who return books in detail with condition is bookID =
'K2004'
SELECT*FROM dbo.Students AS stu, dbo.Staffs AS sta, dbo.Books AS bok,
dbo.[Returns] AS rt
WHERE stu.studentID=rt.studentID
AND sta.staffID=rt.staffID
AND bok.bookID=rt.bookID
AND bok.bookID='K2004'
GO
--
SELECT [Returns].* FROM Students, Staffs, Books, [Returns]
WHERE Students.studentID=[Returns].studentID
AND Staffs.staffID=[Returns].staffID
AND Books.bookID=[Returns].bookID
AND Students.studentID='S112'
AND Books.bookID='K2005'
GO
--
SELECT * FROM
dbo.Students INNER JOIN dbo.Borrows
ON Students.studentID=Borrows.studentID
--
SELECT * FROM
dbo.Books INNER JOIN dbo.[Returns]
ON Books.bookID=[Returns].bookID
AND [Returns].bookID='K2007'
```

```
--Advanced queries
SELECT * FROM Staffs
WHERE staffAddress = (SELECT staffAddress
                      FROM   Staffs
                      WHERE  staffID = 'TF04')
GO
--
SELECT * FROM Students
WHERE studentGender IN
      (SELECT studentGender
       FROM   Students
       WHERE  studentID = 'S113'
       OR     studentID = 'S118')
GO

--Function
--Create function no have parameter
CREATE FUNCTION UF_SelectAllStudent()
RETURNS TABLE
AS RETURN SELECT * FROM dbo.Students
GO
--Use function
SELECT * FROM UF_SelectAllStudent()

--Create function with parameter
CREATE FUNCTION UF_SelectStudentPhone (@studentID VARCHAR(10))
RETURNS INT
AS
BEGIN
     DECLARE @studentPhone INT
     SELECT @studentPhone = studentPhone FROM dbo.Students
     WHERE studentID = @studentID
     RETURN @studentPhone
END
GO
--Use function
```

```
SELECT dbo.UF_SelectStudentPhone('S113') AS PhoneNumber
--Delete function or store
DROP FUNCTION UF_SelectAllStudent

--Trigger
ALTER TABLE dbo.Students
ADD Date_of_birth DATETIME
GO

UPDATE Students
SET Date_of_birth = '05/10/1999'
WHERE studentID = 'S105'
GO
UPDATE Students
SET Date_of_birth = '08/09/1996'
WHERE studentID = 'S118'
GO
UPDATE Students
SET Date_of_birth = '07/05/2019'
WHERE studentID = 'S117'
GO
---Create trigger don't drop student bigger than 18 age.
CREATE TRIGGER UTG_AbortOlderThan18
ON dbo.Students
FOR DELETE
AS
BEGIN
     DECLARE @Count INT = 0

     SELECT @Count = COUNT(*) FROM deleted
     WHERE YEAR(GETDATE()) - YEAR(deleted.Date_of_birth) < 18

     IF (@Count > 0)
     BEGIN
     PRINT 'DO NOT DELETE PEOPLE BIGGER THAN 18 AGE'
     ROLLBACK TRAN
     END
```

```
END
---Test trigger new create
DELETE dbo.Students WHERE studentID = 'S117'
```

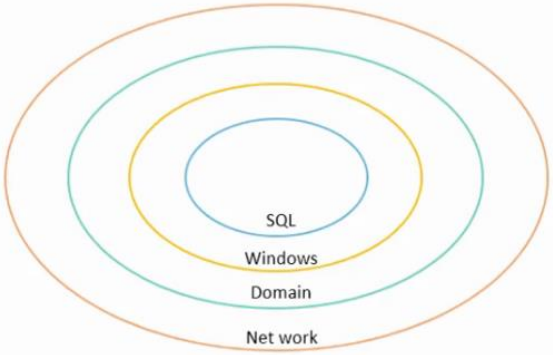
M2 Implement a fully functional database system which includes system security and database maintenance.

1. System security

Security Layers

SQL Server Security works using a concept called "defense-in-depth." This model:

- is a layer based security model;
- is similar to an onion, that is, peeling one layer reveals another;
- strengthens the system significantly;
- takes advantage of Windows security model; and
- leads to two types of accounts.

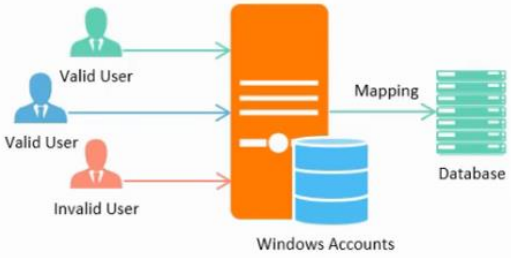


Windows Accounts

Windows accounts are the accounts that are created in Windows itself.

Windows accounts:

- can be Local accounts or Active Directory Domain accounts;
- can be used as accounts in SQL Server directly;
- must be mapped to a SQL Server "login" for access; and
- are the most secure form of security accounts.

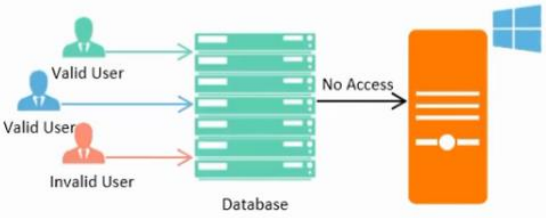


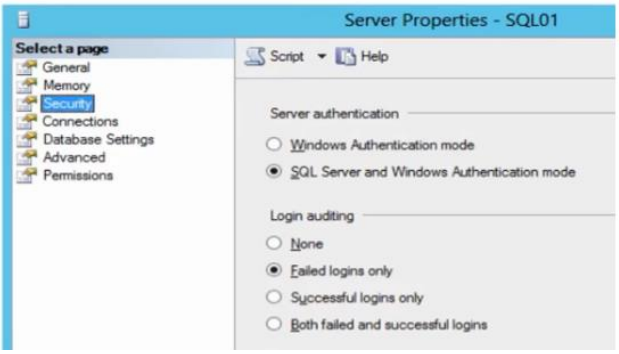
SQL Server Accounts

SQL Server accounts are the other type of accounts that can be allowed to access resources within SQL Server.

Features of Server accounts:

- Accounts are added within SQL Server.
- Each instance stores its own set of accounts.
- They cannot be used to log in to Windows.
- They are not as secure as Windows accounts.
- They are disabled by default.
- Enabling them creates one admin level user: "sa."



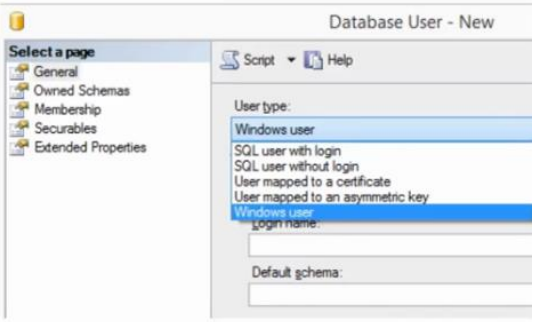


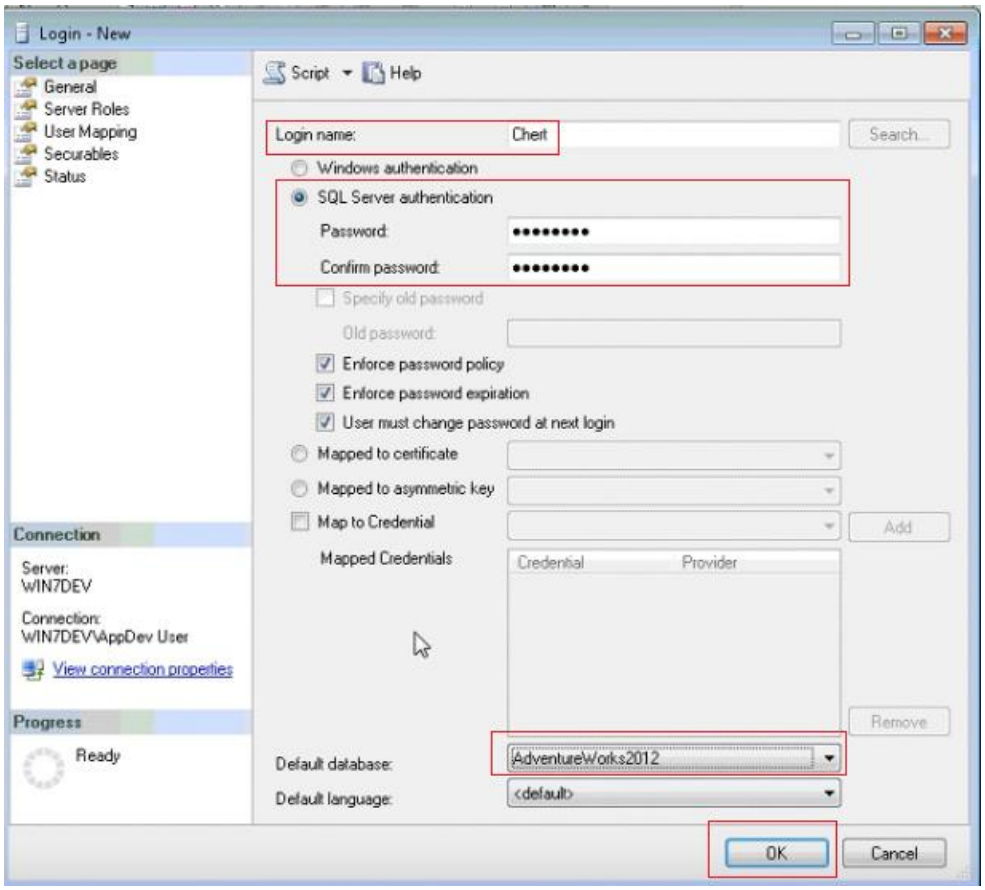
Login Accounts

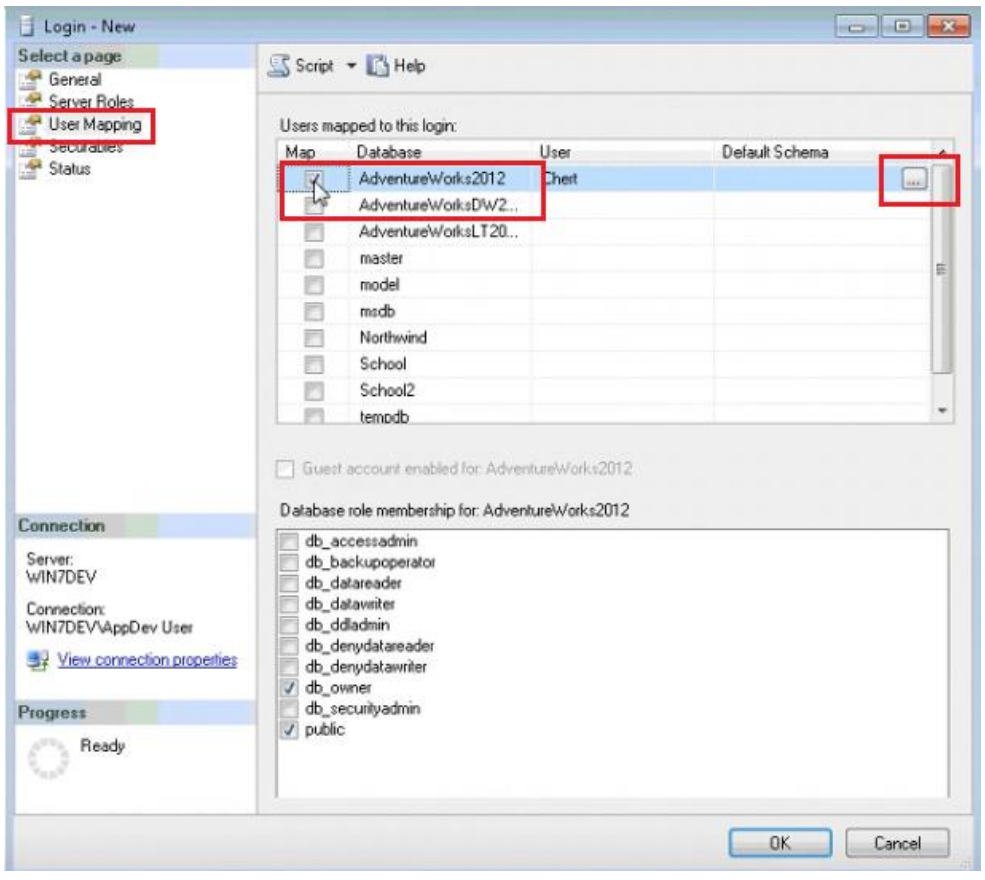
Both Windows accounts and SQL Server accounts need to be mapped with SQL Server to allow access to its resources. This is done using Login accounts in SQL Server.

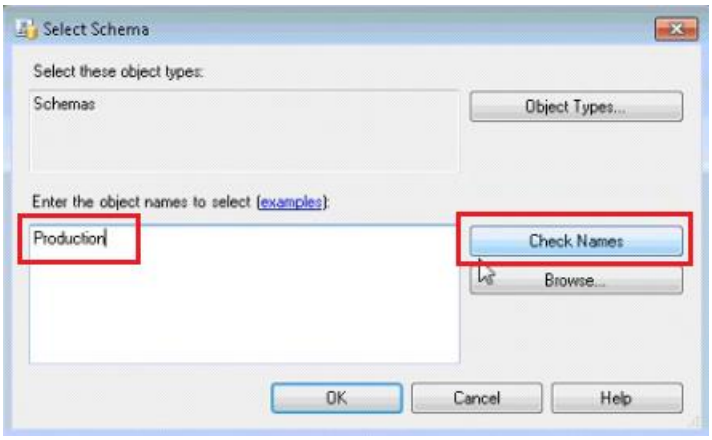
Login accounts:

- are specific to a database;
- are added within a particular database to provide access;
- provide mapping and grant authentication to access the databases; and
- do not provide rights, that is, the permissions on each object in the database.

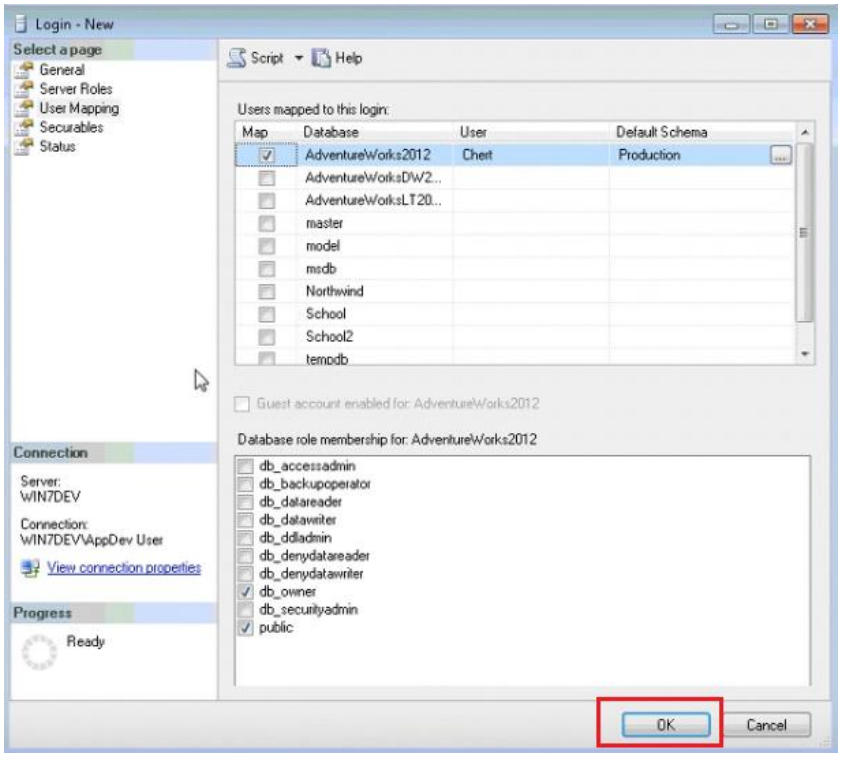


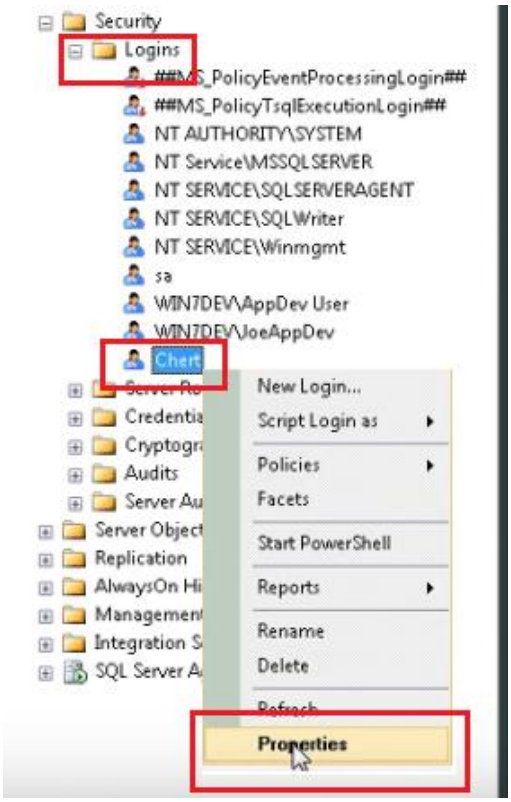

Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
Chert
Search...
Windows authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password at next login
Mapped to certificate
Mapped to asymmetric key
Map to Credential
Add
Mapped Credentials
Credential
Provider
Remove
Default database:
AdventureWorks2012
Default language:
<default>
Connection
Server:
WIN7DEV
Connection:
WIN7DEV\AppDev User
View connection properties
Progress
Ready
OK
Cancel

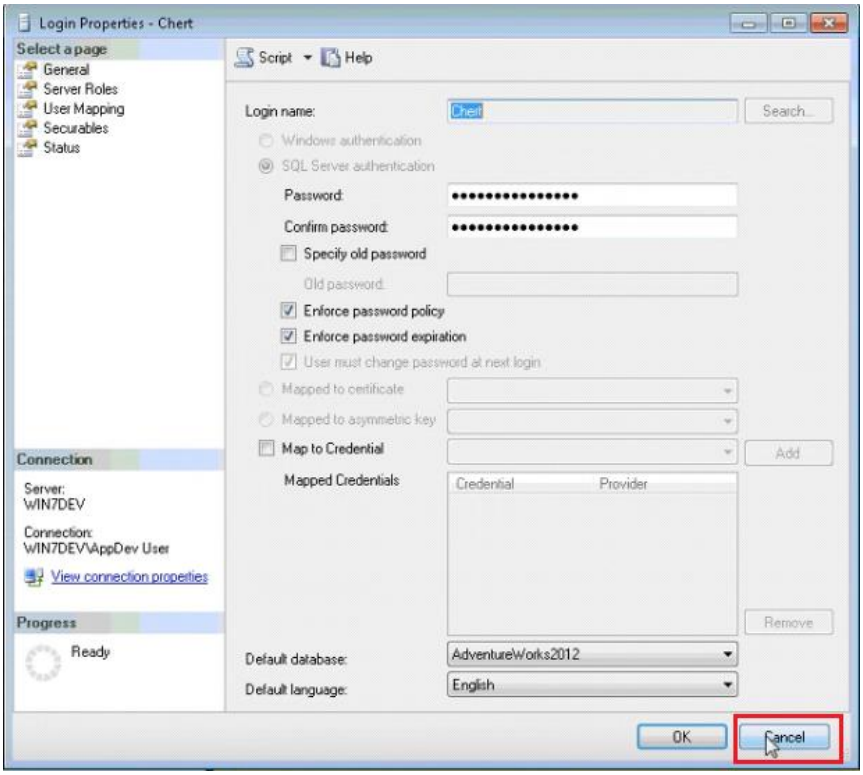

Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Users mapped to this login:
Map
Database
User
Default Schema
AdventureWorks2012
Chert
AdventureWorksDW2...
AdventureWorksLT20...
master
model
msdb
Northwind
School
School2
tempdb
Guest account enabled for: AdventureWorks2012
Database role membership for: AdventureWorks2012
db_accessadmin
db_backupoperator
db_datareader
db_datawriter
db_ddladmin
db_denydatareader
db_denydatawriter
db_owner
db_securityadmin
public
Connection
Server:
WIN7DEV
Connection:
WIN7DEV\AppDev User
View connection properties
Progress
Ready
OK
Cancel


Select Schema
Select these object types:
Schemas
Object Types...
Enter the object names to select (examples):
Production
Check Names
Browse...
OK
Cancel
Help


Select Schema
Select these object types:
Schemas
Object Types...
Enter the object names to select (examples):
[Production]
Check Names
Browse...
OK
Cancel
Help


Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Users mapped to this login:
Map
Database
User
Default Schema
AdventureWorks2012
Chert
Production
AdventureWorksDW2...
AdventureWorksLT20...
master
model
msdb
Northwind
School
School2
tempdb
Guest account enabled for: AdventureWorks2012
Database role membership for: AdventureWorks2012
db_accessadmin
db_backupoperator
db_datareader
db_datawriter
db_ddladmin
db_denydatareader
db_denydatawriter
db_owner
db_securityadmin
public
Connection
Server:
WIN7DEV
Connection:
WIN7DEV\AppDev User
View connection properties
Progress
Ready
OK
Cancel


Security
Logins
##MS_PolicyEventProcessingLogin##
##MS_PolicyTsqlExecutionLogin##
NT AUTHORITY\SYSTEM
NT Service\MSSQLSERVER
NT SERVICE\SQLSERVERAGENT
NT SERVICE\SQLWriter
NT SERVICE\Winmgmt
sa
WIN7DEV\AppDev User
WIN7DEV\JoeAppDev
Chert
Credentia
Cryptogr
Audits
Server Au
Server Object
Replication
AlwaysOn Hi
Managemen
Integration S
SQL Server A
New Login...
Script Login as
Policies
Facets
Start PowerShell
Reports
Rename
Delete
Refresh
Properties

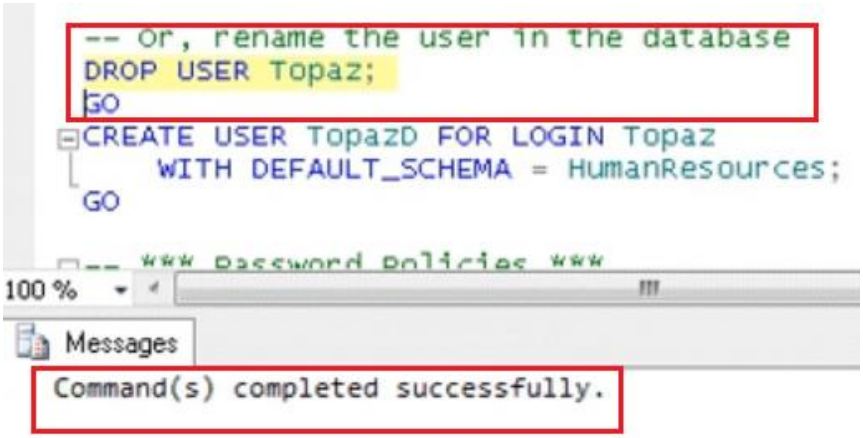

Login Properties - Chert
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
Chert
Search...
Windows authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password at next login
Mapped to certificate
Mapped to asymmetric key
Map to Credential
Add
Mapped Credentials
Credential
Provider
Remove
Connection
Server:
WIN7DEV
Connection:
WIN7DEV\AppDev User
View connection properties
Progress
Ready
Default database:
AdventureWorks2012
Default language:
English
OK
Cancel

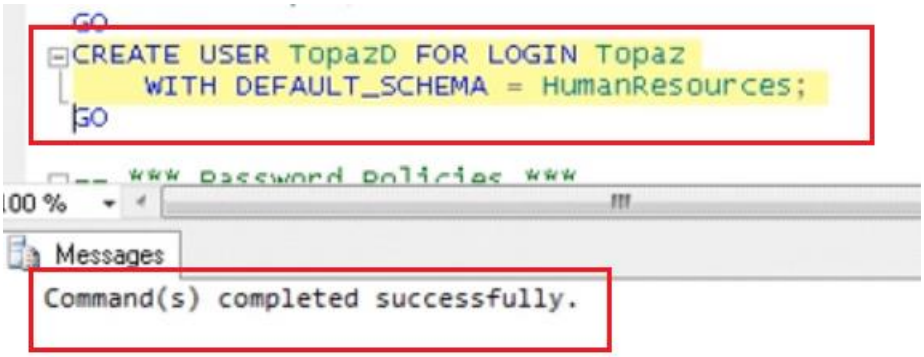
```
-- *** SQL Server Logins ***
-- ************************
CREATE LOGIN Topaz WITH PASSWORD = 'Bo3EXNWJ#N6ndg5';
GO

USE AdventureWorks2012;
GO

CREATE USER Topaz FOR LOGIN Topaz
    WITH DEFAULT_SCHEMA = HumanResources;
GO

-- Or, rename the user in the database
DROP USER Topaz;
GO
CREATE USER TopazD FOR LOGIN Topaz
    WITH DEFAULT_SCHEMA = HumanResources;
GO

-- *** Password Policies ***
```

```
Messages
Command(s) completed successfully.
```

```
GO

CREATE USER Topaz FOR LOGIN Topaz
    WITH DEFAULT_SCHEMA = HumanResources;
GO

-- Or, rename the user in the database
DROP USER Topaz;
GO
CREATE USER TopazD FOR LOGIN Topaz
    WITH DEFAULT_SCHEMA = HumanResources;
GO

-- *** Password Policies ***
```

```
Messages
Command(s) completed successfully.
```





Windows and SQL Server Logins

- SQL Server logins are not stored in Windows
  - Disabled if you select Windows authentication
- Mixed mode is much more flexible
  - But less secure

Password Policy and Enforcement

- Before SQL Server 2005, no enforcement of passwords for SQL Server logins
  - No minimum strength
  - No expiration policy
- SQL Server now hooks into Windows password policy
  - Windows Server 2003, Vista, and later versions
  - NetValidatePasswordPolicy API method

Beware of the sa Login

- System administrator login
- Mapped to sysadmin fixed server role
- Conveys full system administrator privileges
- Cannot modify or delete
- Must use a strong password!
- Use only as access of last resort
- NEVER use sa for database access through client applications

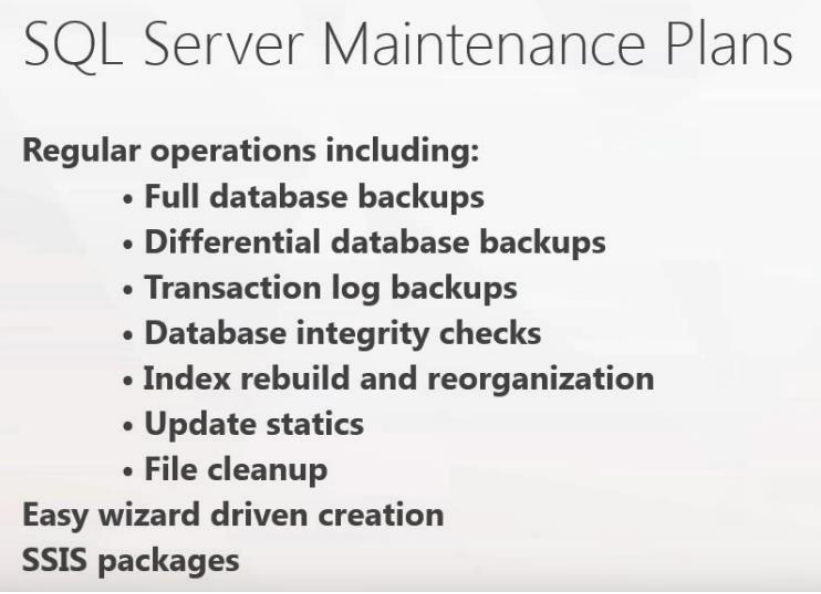
```
-- Requires that SQL Server be running on Windows Server 2003 or later
-- and that password policies be enabled in Local Security Settings.
USE master;
GO
CREATE LOGIN SIMPLEPWD WITH PASSWORD = 'SIMPLEPWD';
GO
ALTER LOGIN Topaz WITH PASSWORD = 'Bo3EXNWJ#N6ndg5',
    CHECK_EXPIRATION = OFF, CHECK_POLICY = OFF;

-- Unlock after too many unsuccessful login attempts
ALTER LOGIN Topaz WITH PASSWORD = 'Bo3EXNWJ#N6ndg5' UNLOCK

-- *** Authorization ***
-- **********************
-- *** Server Roles
```

100 %

Messages

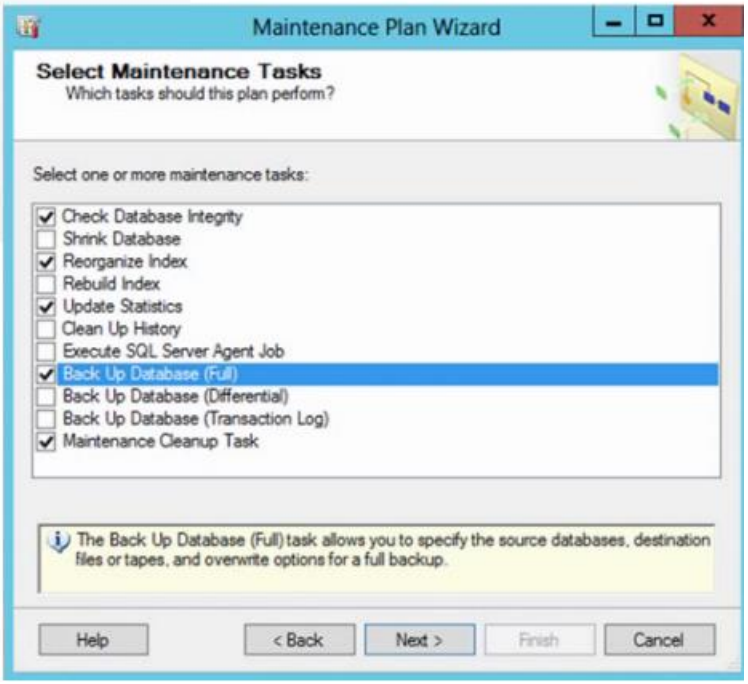
```
Command(s) completed successfully.
```

2. Database maintenance (Create Maintenance Plan and Optimise your database)



Maintenance Plan Wizard

Easy step-by-step Maintenance Plan creation
SSMS > Management > Maintenance Plans
Multiple tasks
Rearrange task order
Set schedule
Configure tasks
- **Select databases**
- **Other specific settings**

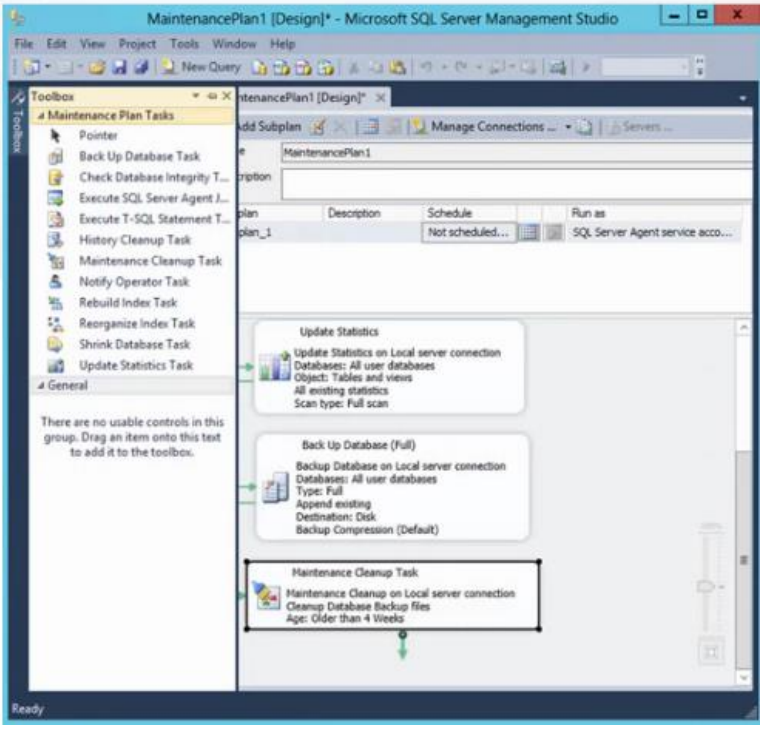


Updating Maintenance Plans

Create new Maintenance Plans
Update existing Maintenance Plans
SSMS > Management >Maintenance Plans
- **> New Maintenance Plan**

Or double click [Maintenance Plan]
Double click tasks
Toolbox
- **drag-and drop tasks**

Calendar icon to schedule



Viewing Maintenance Plan Run History

Viewing run history

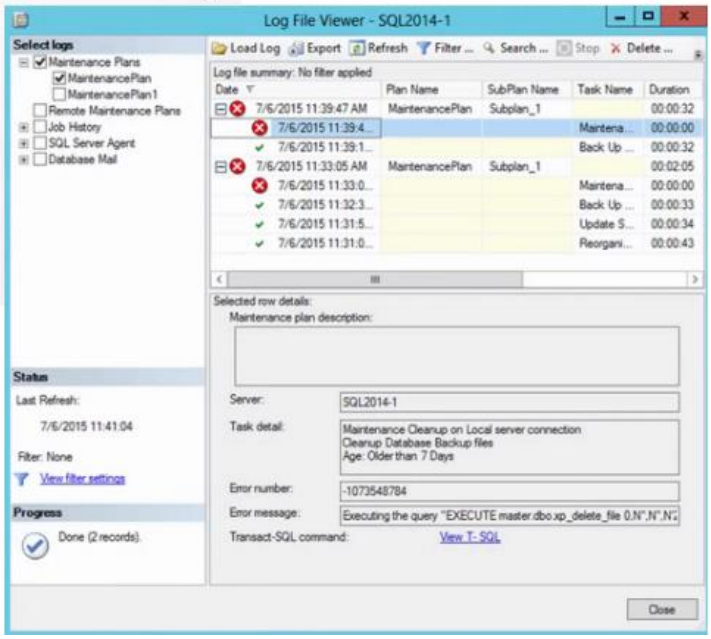
SSMS > Management >Maintenance Plans

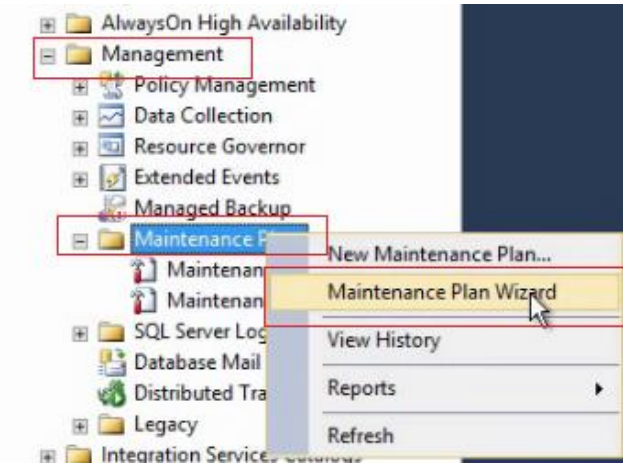
> [Maintenance Plan] > View History

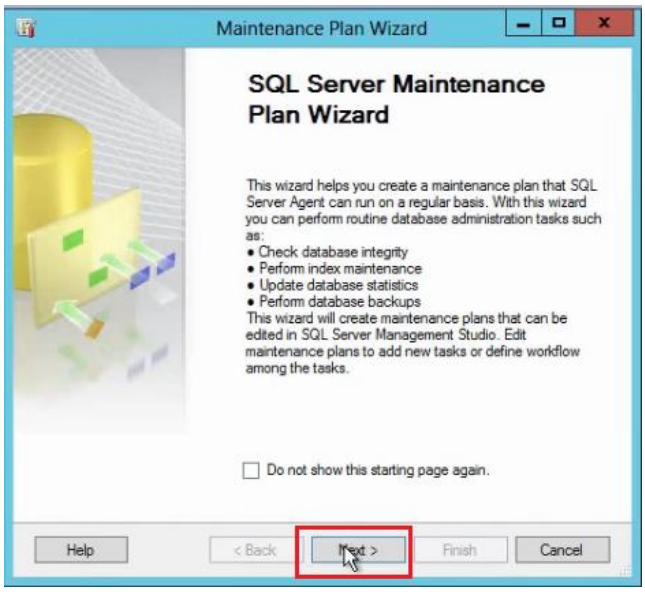
Lists plans and tasks

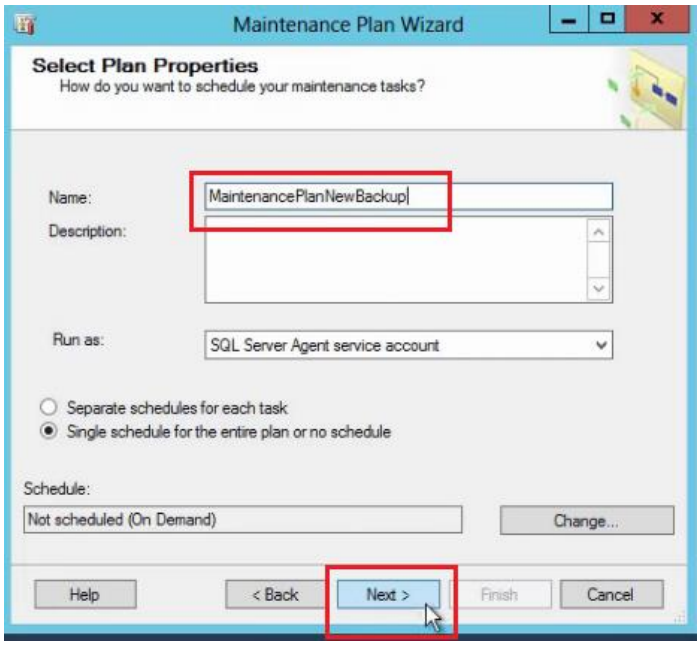
Green checks = success

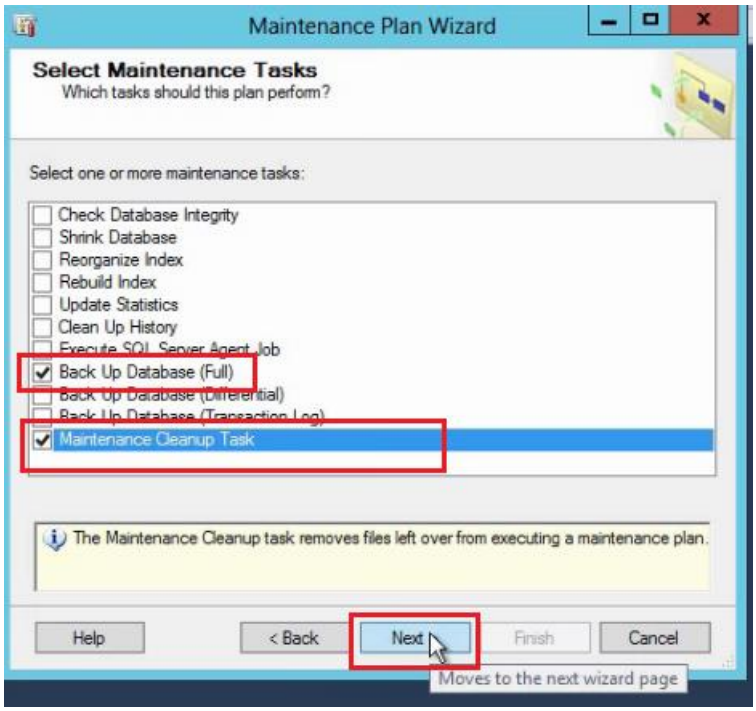
Red X = errors

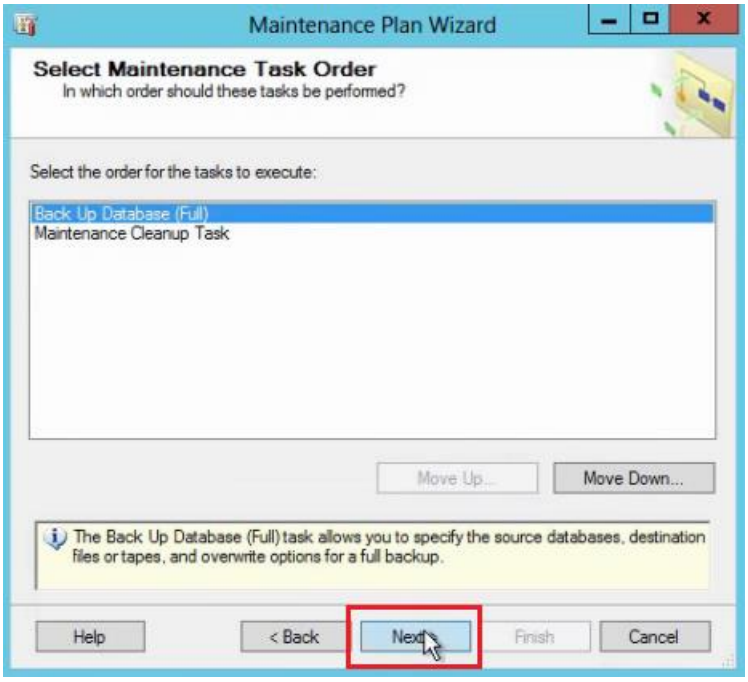


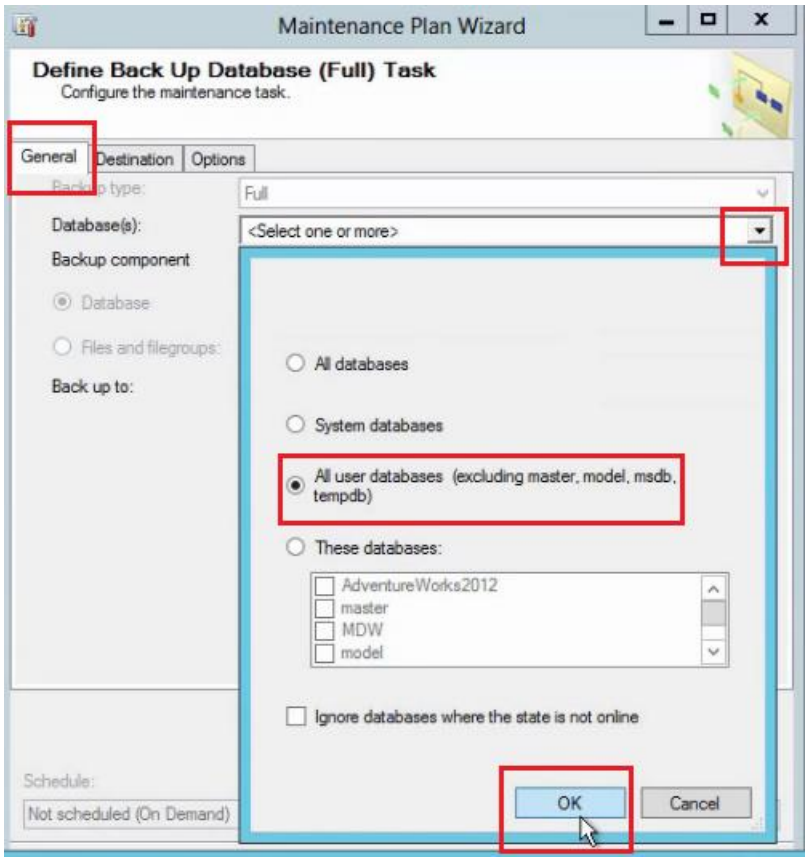


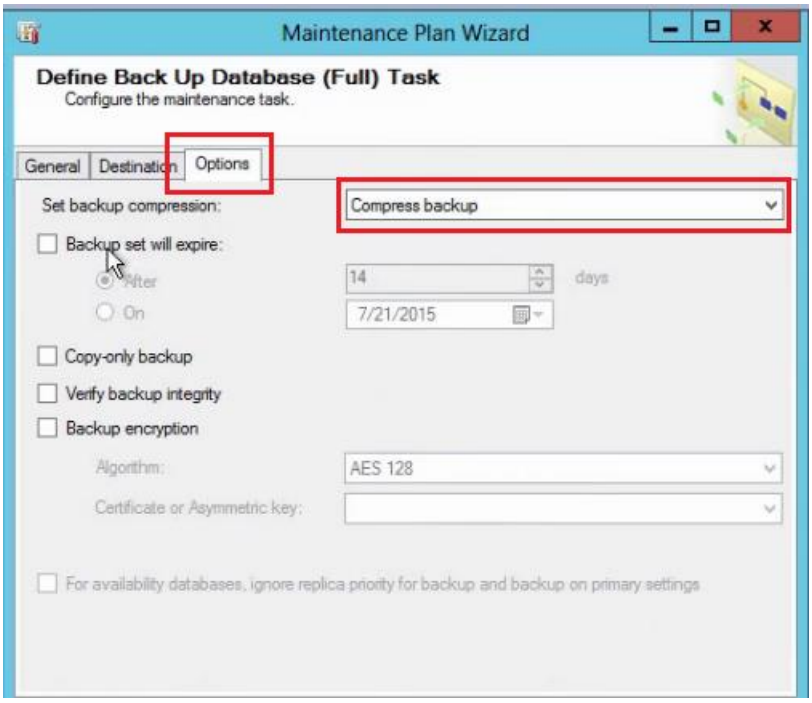

Maintenance Plan Wizard
SQL Server Maintenance Plan Wizard
This wizard helps you create a maintenance plan that SQL Server Agent can run on a regular basis. With this wizard you can perform routine database administration tasks such as:
• Check database integrity
• Perform index maintenance
• Update database statistics
• Perform database backups
This wizard will create maintenance plans that can be edited in SQL Server Management Studio. Edit maintenance plans to add new tasks or define workflow among the tasks.
Do not show this starting page again.
Help
< Back
Next >
Finish
Cancel

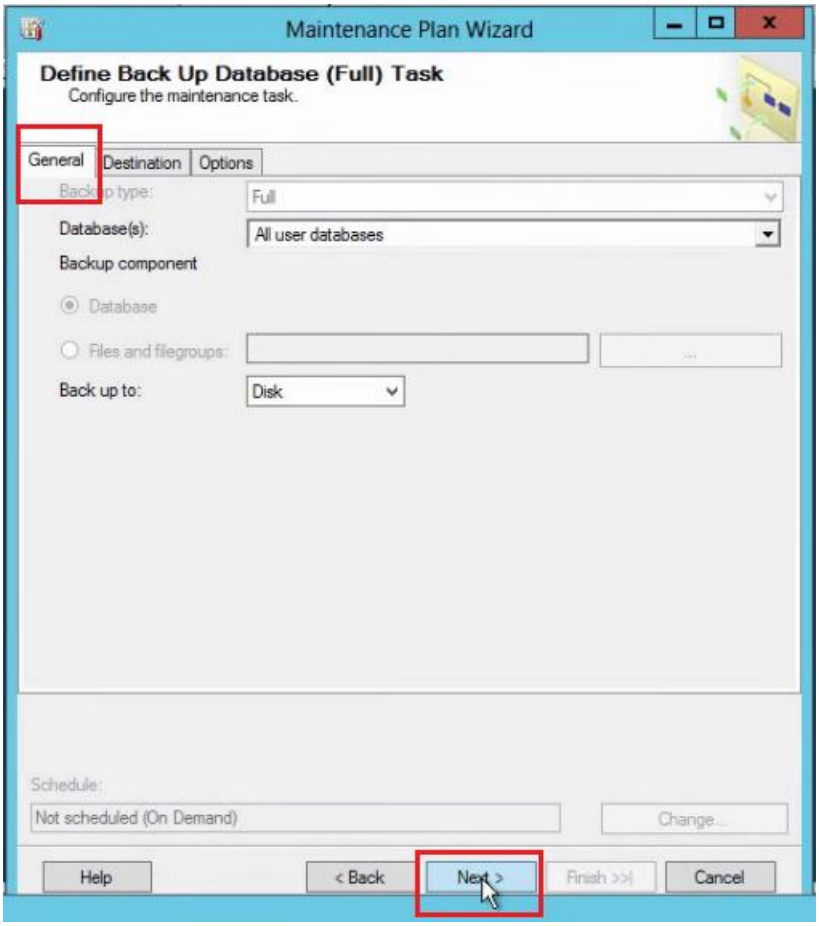

Maintenance Plan Wizard
Select Plan Properties
How do you want to schedule your maintenance tasks?
Name:
MaintenancePlanNewBackup
Description:
Run as:
SQL Server Agent service account
Separate schedules for each task
Single schedule for the entire plan or no schedule
Schedule:
Not scheduled (On Demand)
Change...
Help
< Back
Next >
Finish
Cancel

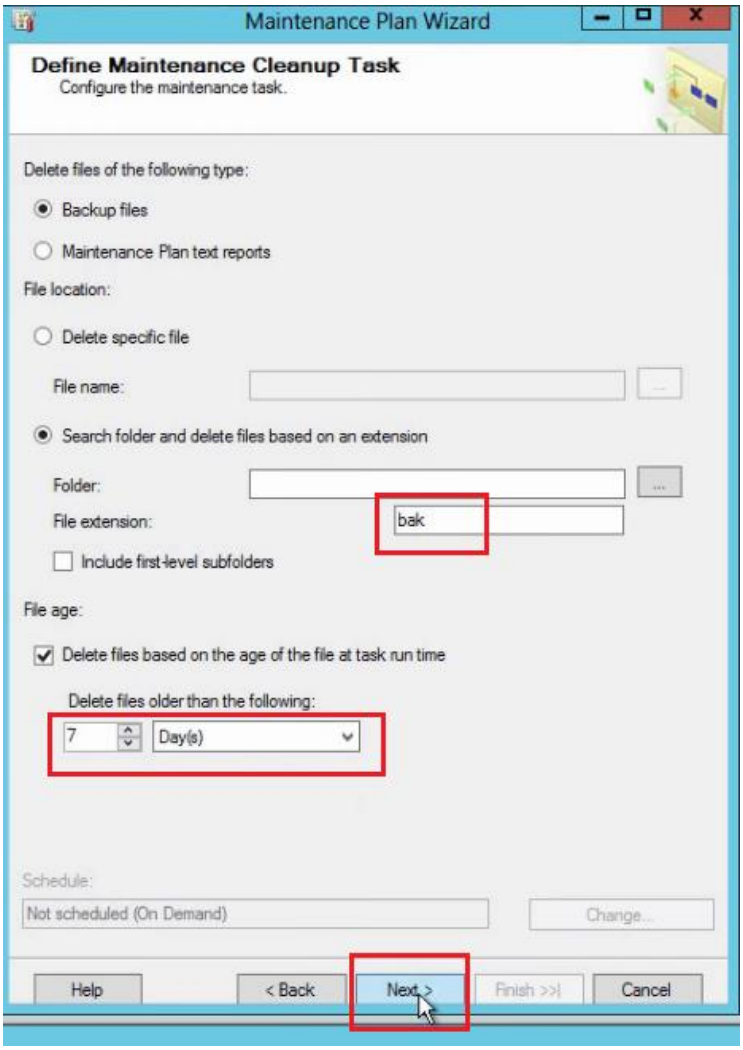

Maintenance Plan Wizard
Select Maintenance Tasks
Which tasks should this plan perform?
Select one or more maintenance tasks:
Check Database Integrity
Shrink Database
Reorganize Index
Rebuild Index
Update Statistics
Clean Up History
Execute SQL Server Agent Job
Back Up Database (Full)
Back Up Database (Differential)
Back Up Database (Transaction Log)
Maintenance Cleanup Task
The Maintenance Cleanup task removes files left over from executing a maintenance plan.
Help
< Back
Next
Finish
Cancel
Moves to the next wizard page

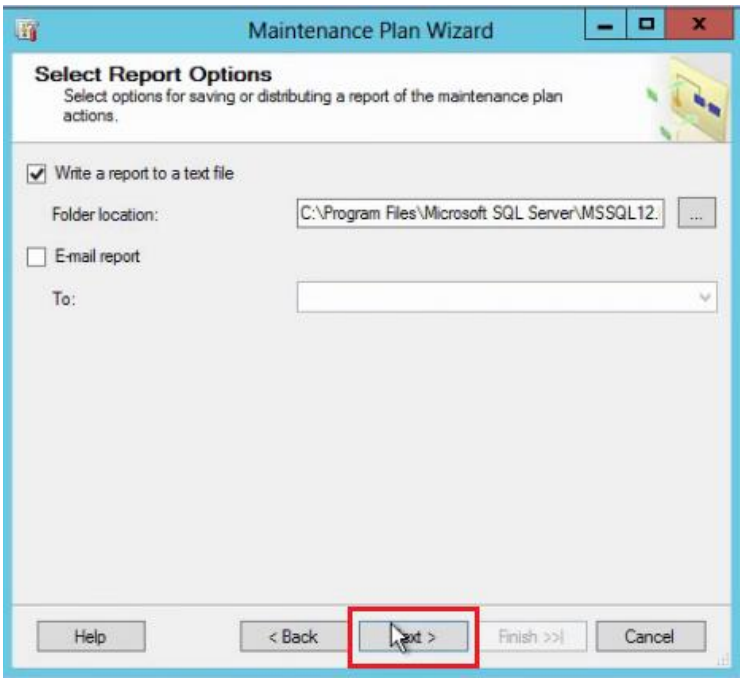

Maintenance Plan Wizard
Select Maintenance Task Order
In which order should these tasks be performed?
Select the order for the tasks to execute:
Back Up Database (Full)
Maintenance Cleanup Task
Move Up...
Move Down...
The Back Up Database (Full) task allows you to specify the source databases, destination files or tapes, and overwrite options for a full backup.
Help
< Back
Next
Finish
Cancel

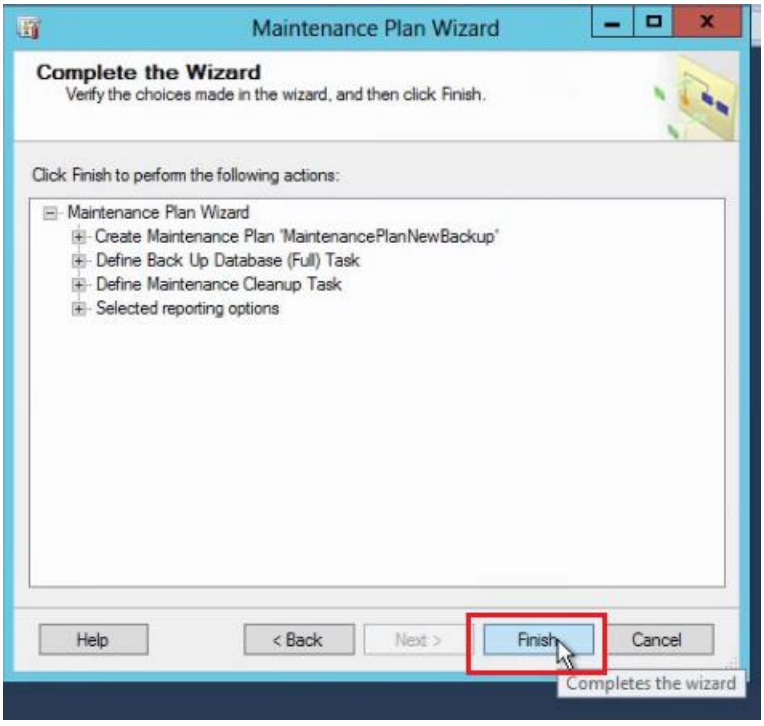

Maintenance Plan Wizard
Define Back Up Database (Full) Task
Configure the maintenance task.
General
Destination
Options
Backup type:
Full
Database(s):
<Select one or more>
Backup component
Database
Files and filegroups:
Back up to:
All databases
System databases
All user databases (excluding master, model, msdb, tempdb)
These databases:
AdventureWorks2012
master
MDW
model
Ignore databases where the state is not online
Schedule:
Not scheduled (On Demand)
OK
Cancel

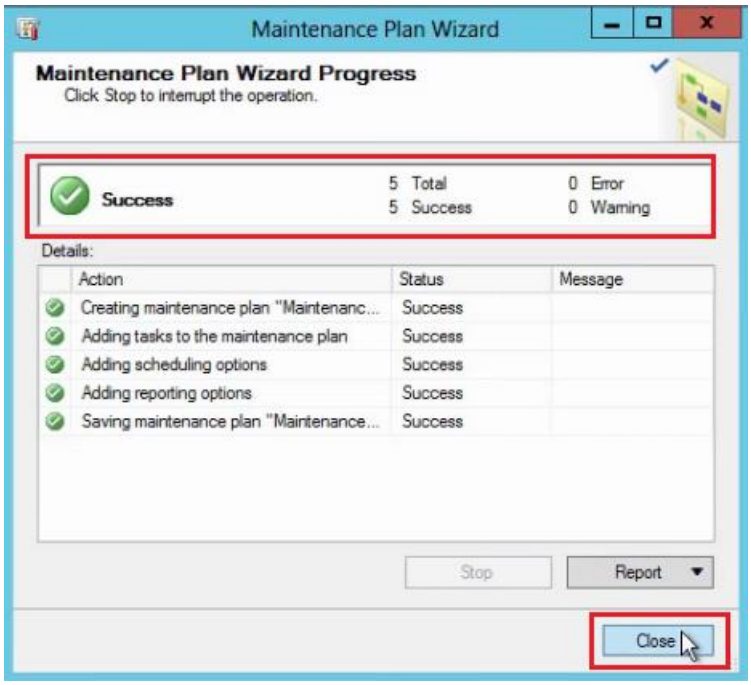

Maintenance Plan Wizard
Define Back Up Database (Full) Task
Configure the maintenance task.
General
Destination
Options
Set backup compression:
Compress backup
Backup set will expire:
After
14
days
On
7/21/2015
Copy-only backup
Verify backup integrity
Backup encryption
Algorithm:
AES 128
Certificate or Asymmetric key:
For availability databases, ignore replica priority for backup and backup on primary settings

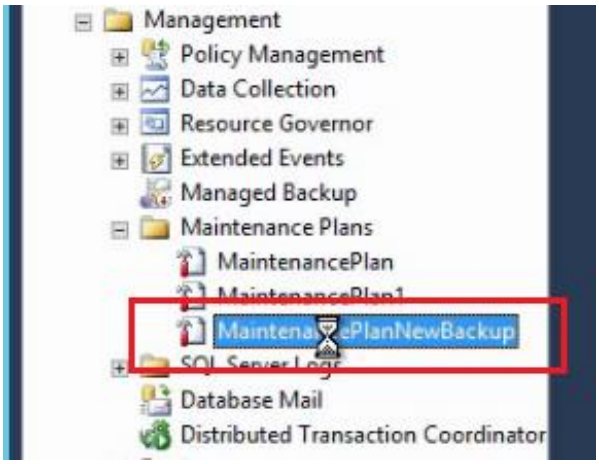

Maintenance Plan Wizard
Define Back Up Database (Full) Task
Configure the maintenance task.
General
Destination
Options
Backup type:
Full
Database(s):
All user databases
Backup component
Database
Files and filegroups:
Back up to:
Disk
Schedule:
Not scheduled (On Demand)
Change...
Help
< Back
Next >
Finish >>|
Cancel

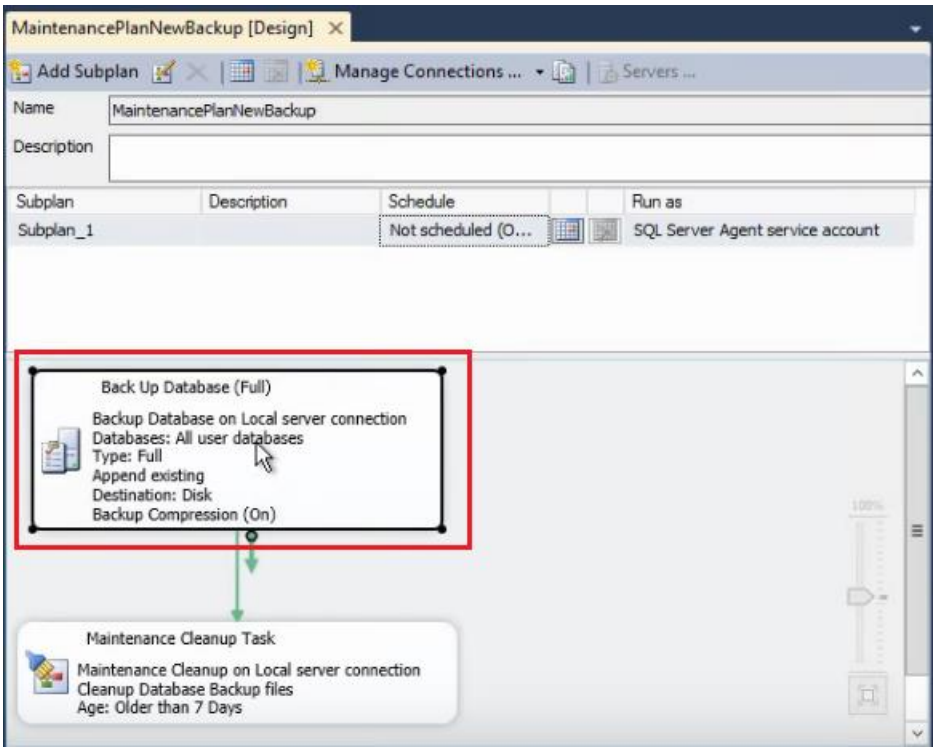

Maintenance Plan Wizard
Define Maintenance Cleanup Task
Configure the maintenance task.
Delete files of the following type:
Backup files
Maintenance Plan text reports
File location:
Delete specific file
File name:
Search folder and delete files based on an extension
Folder:
File extension:
bak
Include first-level subfolders
File age:
Delete files based on the age of the file at task run time
Delete files older than the following:
7
Day(s)
Schedule:
Not scheduled (On Demand)
Change...
Help
< Back
Next >
Finish >>|
Cancel

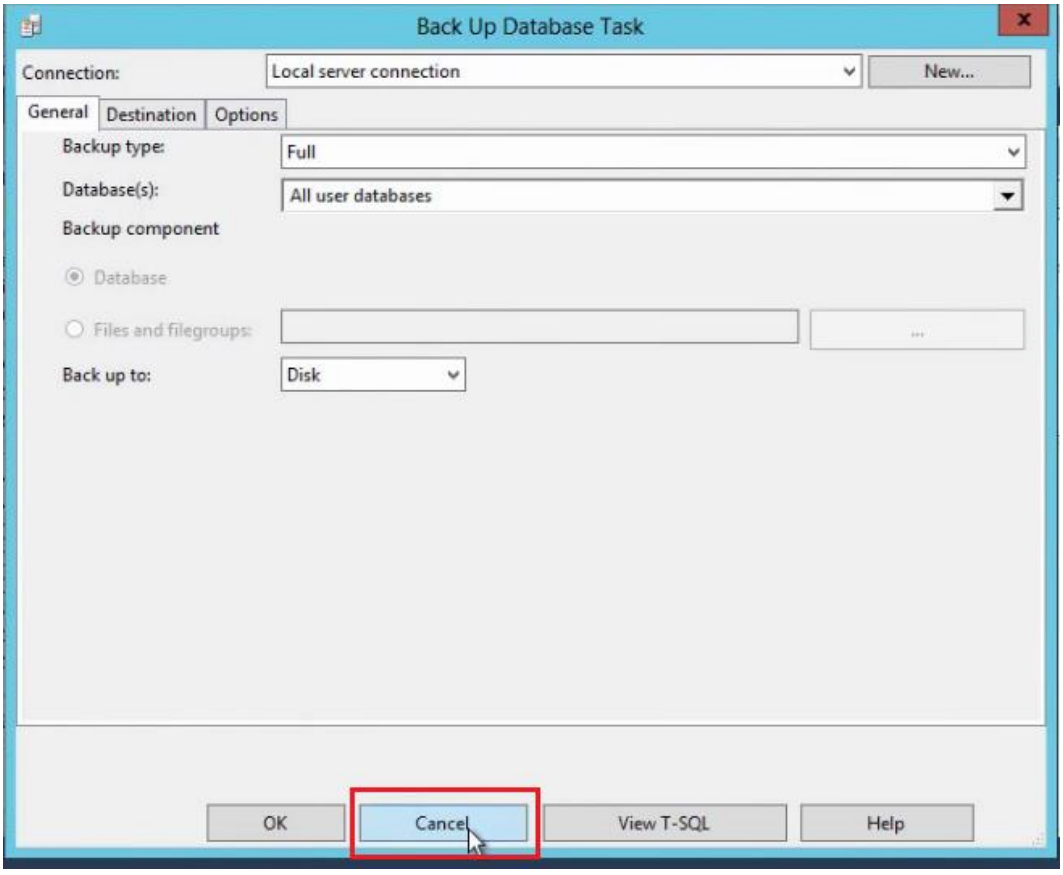

Maintenance Plan Wizard
Select Report Options
Select options for saving or distributing a report of the maintenance plan actions.
Write a report to a text file
Folder location:
C:\Program Files\Microsoft SQL Server\MSSQL12.
E-mail report
To:
Help
< Back
Next >
Finish >>|
Cancel

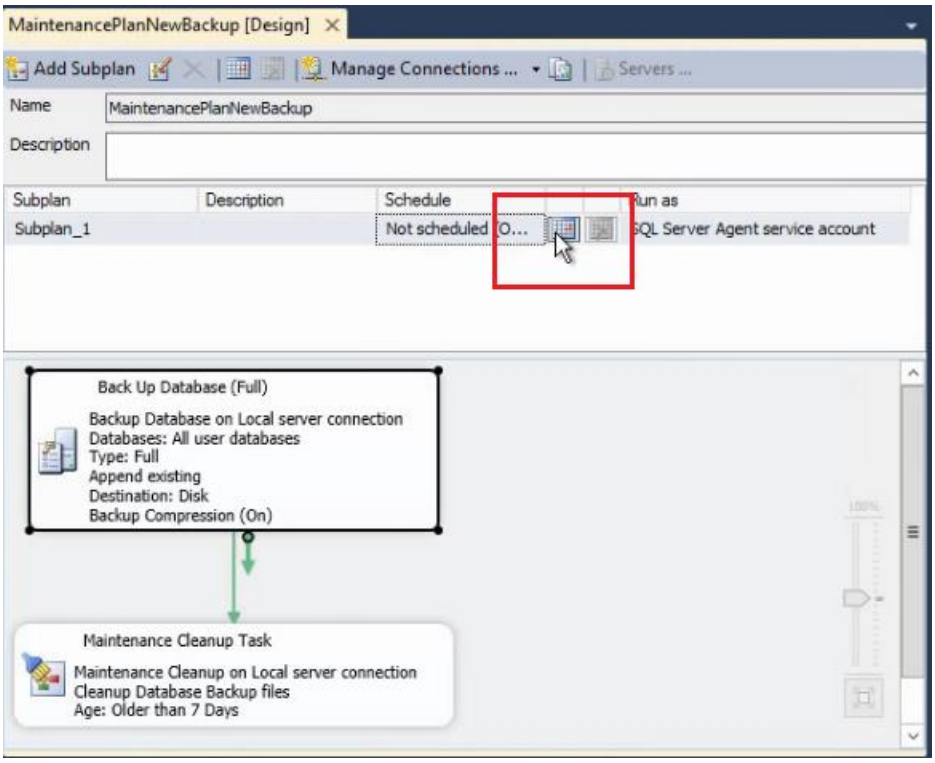

Maintenance Plan Wizard
Complete the Wizard
Verify the choices made in the wizard, and then click Finish.
Click Finish to perform the following actions:
Maintenance Plan Wizard
Create Maintenance Plan 'MaintenancePlanNewBackup'
Define Back Up Database (Full) Task
Define Maintenance Cleanup Task
Selected reporting options
Help
< Back
Next >
Finish
Cancel
Completes the wizard

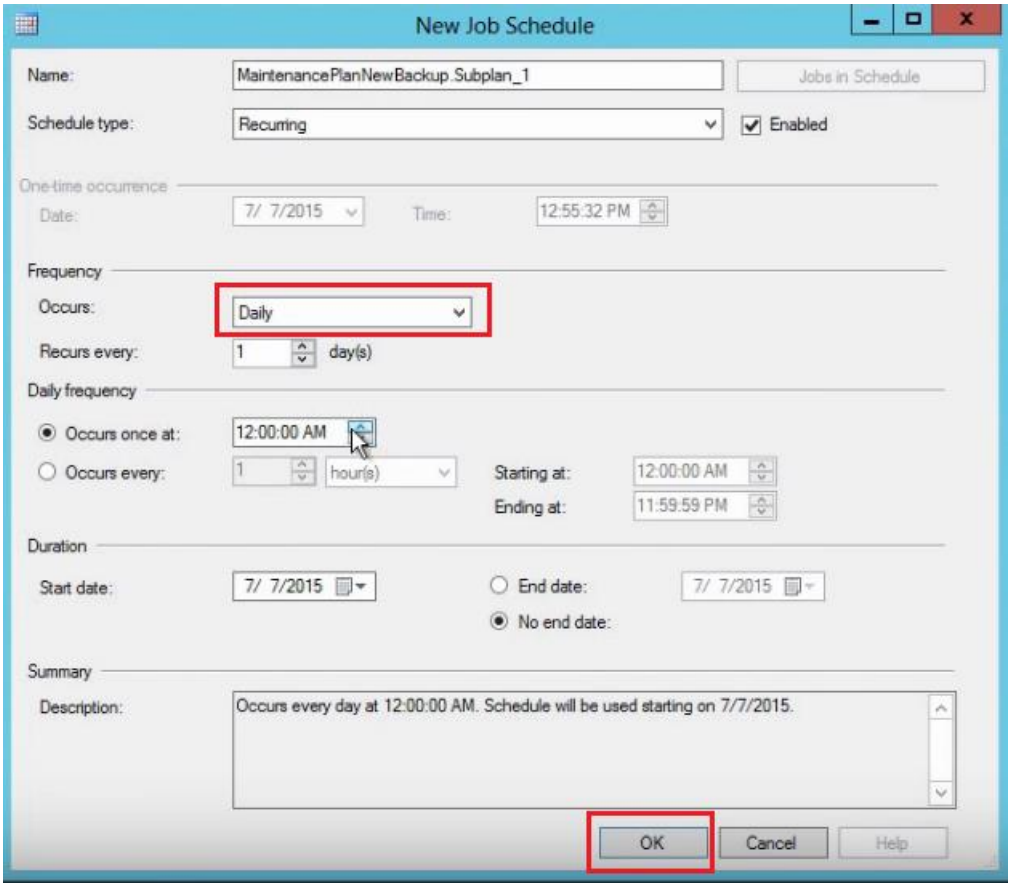

Maintenance Plan Wizard
Maintenance Plan Wizard Progress
Click Stop to interrupt the operation.
Success
5 Total
5 Success
0 Error
0 Warning
Details:
Action
Status
Message
Creating maintenance plan "Maintenanc...
Success
Adding tasks to the maintenance plan
Success
Adding scheduling options
Success
Adding reporting options
Success
Saving maintenance plan "Maintenance...
Success
Stop
Report
Close

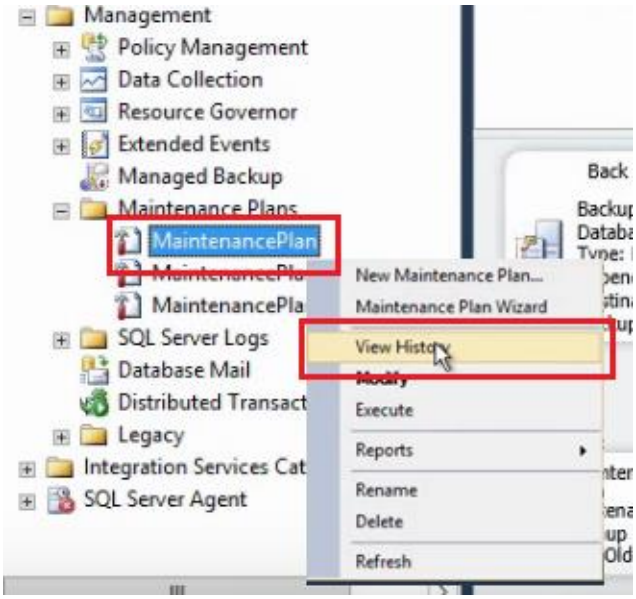

Management
Policy Management
Data Collection
Resource Governor
Extended Events
Managed Backup
Maintenance Plans
MaintenancePlan
MaintenancePlan1
MaintenancePlanNewBackup
SQL Server Logs
Database Mail
Distributed Transaction Coordinator

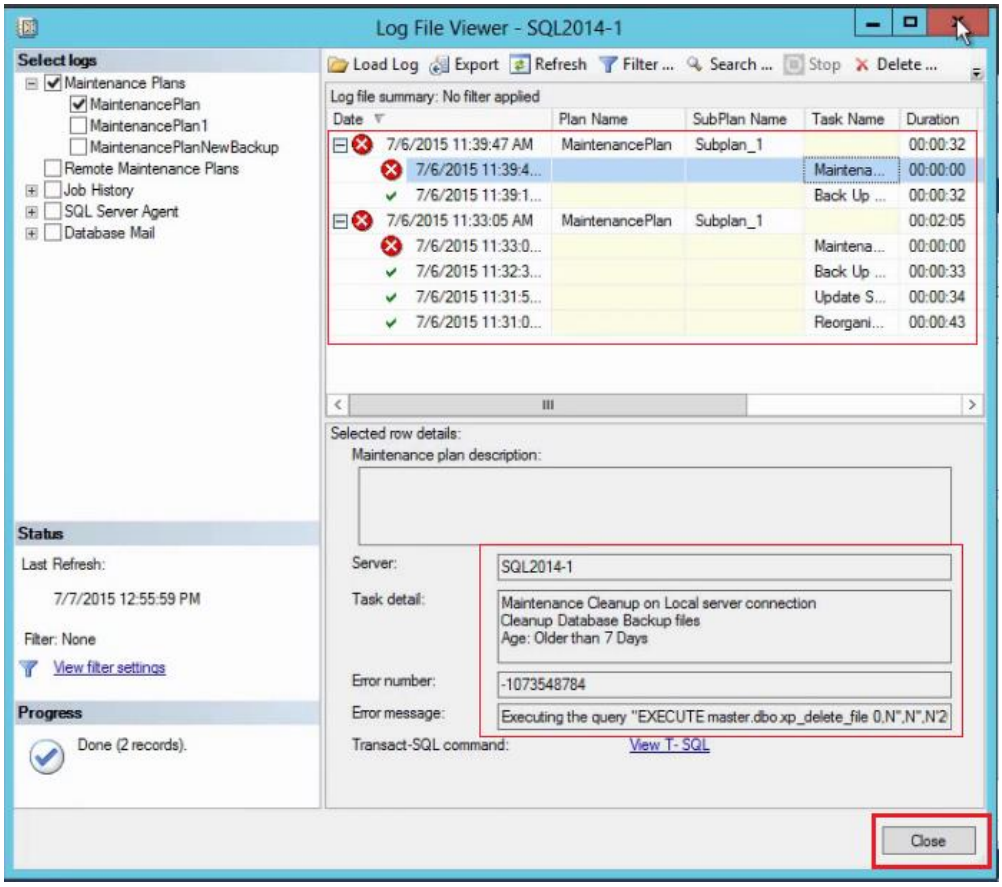

MaintenancePlanNewBackup [Design]
Add Subplan
Manage Connections ...
Servers ...
Name
MaintenancePlanNewBackup
Description
Subplan
Description
Schedule
Run as
Subplan_1
Not scheduled (O...
SQL Server Agent service account
Back Up Database (Full)
Backup Database on Local server connection
Databases: All user databases
Type: Full
Append existing
Destination: Disk
Backup Compression (On)
Maintenance Cleanup Task
Maintenance Cleanup on Local server connection
Cleanup Database Backup files
Age: Older than 7 Days


Back Up Database Task
Connection:
Local server connection
New...
General
Destination
Options
Backup type:
Full
Database(s):
All user databases
Backup component
Database
Files and filegroups:
Back up to:
Disk
OK
Cancel
View T-SQL
Help


MaintenancePlanNewBackup [Design]
Add Subplan
Manage Connections ...
Servers ...
Name
MaintenancePlanNewBackup
Description
Subplan
Description
Schedule
Run as
Subplan_1
Not scheduled (O...
SQL Server Agent service account
Back Up Database (Full)
Backup Database on Local server connection
Databases: All user databases
Type: Full
Append existing
Destination: Disk
Backup Compression (On)
Maintenance Cleanup Task
Maintenance Cleanup on Local server connection
Cleanup Database Backup files
Age: Older than 7 Days


New Job Schedule
Name:
MaintenancePlanNewBackup.Subplan_1
Jobs in Schedule
Schedule type:
Recurring
Enabled
One-time occurrence
Date:
7/ 7/2015
Time:
12:55:32 PM
Frequency
Occurs:
Daily
Recurs every:
1
day(s)
Daily frequency
Occurs once at:
12:00:00 AM
Occurs every:
1
hour(s)
Starting at:
12:00:00 AM
Ending at:
11:59:59 PM
Duration
Start date:
7/ 7/2015
End date:
7/ 7/2015
No end date:
Summary
Description:
Occurs every day at 12:00:00 AM. Schedule will be used starting on 7/7/2015.
OK
Cancel
Help


Management
Policy Management
Data Collection
Resource Governor
Extended Events
Managed Backup
Maintenance Plans
MaintenancePlan
SQL Server Logs
Database Mail
Distributed Transact
Legacy
Integration Services Cat
SQL Server Agent
New Maintenance Plan...
Maintenance Plan Wizard
View History
Execute
Reports
Rename
Delete
Refresh


Log File Viewer - SQL2014-1
Select logs
Maintenance Plans
MaintenancePlan
MaintenancePlan1
MaintenancePlanNewBackup
Remote Maintenance Plans
Job History
SQL Server Agent
Database Mail
Load Log
Export
Refresh
Filter ...
Search ...
Stop
Delete ...
Log file summary: No filter applied
Date
Plan Name
SubPlan Name
Task Name
Duration
7/6/2015 11:39:47 AM MaintenancePlan Subplan_1 00:00:32
7/6/2015 11:39:4... Maintena... 00:00:00
7/6/2015 11:39:1... Back Up ... 00:00:32
7/6/2015 11:33:05 AM MaintenancePlan Subplan_1 00:02:05
7/6/2015 11:33:0... Maintena... 00:00:00
7/6/2015 11:32:3... Back Up ... 00:00:33
7/6/2015 11:31:5... Update S... 00:00:34
7/6/2015 11:31:0... Reorgani... 00:00:43
Selected row details:
Maintenance plan description:
Server:
SQL2014-1
Task detail:
Maintenance Cleanup on Local server connection
Cleanup Database Backup files
Age: Older than 7 Days
Error number:
-1073548784
Error message:
Executing the query "EXECUTE master.dbo.xp_delete_file 0,N'',N'',N'2
Transact-SQL command:
View T-SQL
Status
Last Refresh:
7/7/2015 12:55:59 PM
Filter: None
View filter settings
Progress
Done (2 records).
Close

M3 Assess whether meaningful data has been extracted through the use of query tools to produce appropriate management information.

- The extracted data is clear, detailed, accurate and as expected. This helps us to manage information more easily. In addition, we can through the extracted data to edit the information as desired.

- The query commands used appropriate for their roles, meanings and its use. This helps the data to be extracted as expected. From there, the information will always be accurate.

LO3 Test the system against user and system requirements.

P4 Test the system against user and system requirements.

1. Test case.

Test	What is being tested	How	Test data used	Expected Results
Q1	Add new item	Input the new data following by the column of the table	{dbo.Staffs, staffID = 'TF22', staffName = 'Joy Park', designation = 'Architecture, staffAddress = 'Alaska', staffGender = 'Male'}	The new item was stored into the database
Q2	Validation of input	Input for add new item	{dbo.Students, studentID = 'S222', studentName = 'Shawn Mendes'}	Error. conflicted with the FOREIGN KEY constraint
Q3	Add columns to the table	Needs table name to add columns, a column name and a data type of columns. Use the ALTER TABLE ADD command.	{dbo.Staffs, column is Year_of_Birth, data type is INT}	New column has been added to the table successfully
Q4	Search the list of all primary and foreign keys in the database	Select the database and use the command SELECT DISTINCT FROM	{database is OnlineLibrarySystem}	Successful display of all primary and foreign keys in database
Q5	Delete a specific data	When deleting a specific data. It needs a condition	{dbo.Departments, departmentName = 'Art'}	The item or constraint was deleted from database
Q6	Delete constraint items	Deleting a constraint need the name of that foreign key or primary key	{dbo.Borrows, constraint = CK_Date}	The item or constraint was deleted from database

Q7	Change the name of the table	Use the global stored procedure named **sp_rename** to rename the table. To execute this procedure, use the **EXEC** command.	{'Departments' *change to* 'dp.Departments';}	The table has been renamed successfully
Q8	Update new item	For a column that holds Null value, we can update that column in specific rows	{dbo.Students, Date_of_birth = '02/02/1992', studentID = 'S119'}	The new item was updated database
Q9	Change name of existing item	When you already insert a row in a column but you want to change a specific row. You can use update.	{dbo.Students, Date_of_birth = '06/06/1996', studentID = 'S119'}	The new item was updated database
Q10	Show list of items of table being selected	Enter the name of the table to view. You can use select.	{dbo.Returns}	The list of items in being selected table was displayed

Queries for test case:

☞ Query 1

```
INSERT INTO
Staffs(staffID,staffName,designation,staffAddress,staffGender)
VALUES ('TF22','Joy Park','Architecture','Alaska','Male')
GO
```

☞ Query 2

```
INSERT INTO Students(studentID,studentName)
VALUES ('S222','Shawn Mendes')
GO
```

☞ Query 3

```
ALTER TABLE Staffs
ADD Year_of_Birth INT
GO
```

☞ Query 4

```
SELECT
DISTINCT
Constraint_Name AS [Constraint],
Table_Schema AS [Schema],
Table_Name AS [TableName] FROM INFORMATION_SCHEMA.[KEY_COLUMN_USAGE]
GO
```

☞ Query 5

```
DELETE FROM Departments
WHERE departmentName = 'Art'
GO
```

☞ Query 6

```
ALTER TABLE dbo.Borrows
DROP CONSTRAINT CK_Date
GO
```

☞ Query 7

```
EXEC sp_rename 'Departments', 'dp.Departments';
```

☞ Query 8

```
UPDATE Students
SET Date_of_birth = '02/02/1992'
WHERE studentID = 'S119'
GO
```

☞ Query 9

```
UPDATE Students
SET Date_of_birth = '06/06/1996'
WHERE studentID = 'S119'
GO
```

☞ Query 10

```
SELECT * FROM dbo.[Returns]
GO
```

2. Test logs.

Test	What is being tested	How	Test data used	Expected Results	Date	Actual results	Action taken
Q1	Lists the primary and foreign keys of a specific table	Select the table and use the command SELECT DISTINCT FROM	SELECT DISTINCT Constraint_Name AS [Constraint], Table_Schema AS [Schema], Table_Name AS [TableName] FROM INFORMATION_SCHEMA.[KEY_COLUMN_USAGE] WHERE INFORMATION_SCHEMA.[KEY_COLUMN_USAGE].[TABLE_NAME]='Returns' GO	Successful display of all primary and foreign keys in table	March 14, 2020	OK	None
Q2	List the size of each table	Select the database and use SUM()	SELECT sob.name AS Table_Name, SUM(sys.length) AS [Size_Table(Bytes)] FROM sysobjects sob, syscolumns sys WHERE sob.xtype='u' AND sys.id=sob.id GROUP BY sob.name	Show list the size of each table successfully	March 14, 2020	OK	None
Q3	Show list of items of table being selected	Enter the name of the table to view. You can use select.	SELECT * FROM dbo.Staffs GO	The list of items in being selected table was displayed	March 14, 2020	OK	None
Q4	Display details of one data in table selected	Enter data need view and What table does it come from (table name). You can use select.	SELECT * FROM dbo.Students WHERE studentID = 'S113' GO	Details of the data displayed successfully	March 14, 2020	OK	None

Q5	Validation of input	Input for add new item	`INSERT INTO Staffs(staffAddress,staffGender) VALUES ('Hawaii','Male') GO`	Error. conflicted with the FOREIGN KEY constraint	March 14, 2020	Column does not allow nulls. INSERT fails. The statement has been terminated. (Because StaffID is Primary key has not been entered)	Recode and re-test
Q6	Change name of existing item	When you already insert a row in a column but you want to change a specific row. You can use update.	`UPDATE Students SET Date_of_birth = '03/30/1993' WHERE studentID = 'S113' GO`	The new item was updated database	March 14, 2020	OK	None
Q7	Delete a specific data	When deleting a specific data. It needs a condition	`DELETE FROM Staffs WHERE designation = 'Guard' GO`	The item or constraint was deleted from database	March 14, 2020	OK	None
Q8	Change the name of the table	Use the global stored procedure named **sp_rename** to rename the table. To execute this procedure, use the **EXEC** command.	`EXEC sp_rename 'Departments', 'dp.Departments';`	The table has been renamed successfully	March 14, 2020	OK. This renaming will affect the foreign key at other tables that are pointing to this table.	None

Q9	Delete constraint items	Deleting a constraint need the name of that foreign key or primary key	`ALTER TABLE dbo.Borrows`
			`DROP CONSTRAINT CK_Date`
			`GO`
				The item or constraint was deleted from database	March 14, 2020	OK	None

Q10	Add new item	Input the new data following by the column of the table	`INSERT INTO Staffs(staffID,staffName,designation,staffAddress,staffGender,Year_of_Birth)`
			`VALUES ('TF888','Lalisa Mannoban','Doctor','New York','Male',1996)`
			`GO`
				The new item was stored into the database	March 14, 2020	OK	None

3. System requirements.

3.1. Hardware requirements.

- ✓ CPU AMD Ryzen5 3550H (3.7GHz)
- ✓ CPU: Other cores are okay (at least i3 or above) or Dual core
- ✓ HDD or SSD at least 20GB
- ✓ RAM at least 8GB

3.2. Software requirements.

- ✓ Windows 10
- ✓ Microsoft SQL Server 2012 above
- ✓ Microsoft .NET Framework 3.5 Server pack 1 (Full Package)
- ✓ Microsoft Server Management Studio

M4 Assess the effectiveness of the testing, including an explanation of the choice of test data used.

Tests were mostly successful and as expected. The selection of test data to use appropriate and diverse. The data are not duplicate and not errors. The combination of data with the syntax is relatively perfect. However, during the test it is possible to change the data. This means the data at the test may be different from the original data. Even so, the data must be similarly about meaning and have relationships with each other. Besides, the testing process went smoothly and quickly.

LO2 & 3.

D2 Evaluate the effectiveness of the database solution in relation to user and system requirements, and suggest improvements.

1. Evaluate

User and system requirements are mostly addressed with optimal solutions. Bring effective, performance and quality relative stable. Requirements about system are easily addressed. As for user requests, they still exist and have not been fully processed. Therefore, it is necessary to offer a more optimal solution for data to solve the remaining requirements of the user. In short, it is necessary to optimize more in all aspects of the system for the online library system to work well and last long.

2. Suggest improvements

- About the system: need to upgrade the memory larger than the original to avoid insufficient memory reduces the performance and quality of the system. In addition, the data processor needs upgrading.

- About users: it is necessary to regularly check user requirements and reviews to improve the user features of the system. In addition, it is necessary to remove unnecessary functions so that the system can run faster. Updated new the information and recent changes to refresh the system. Apply new commands, latest software to the system to suit the market and user tastes.

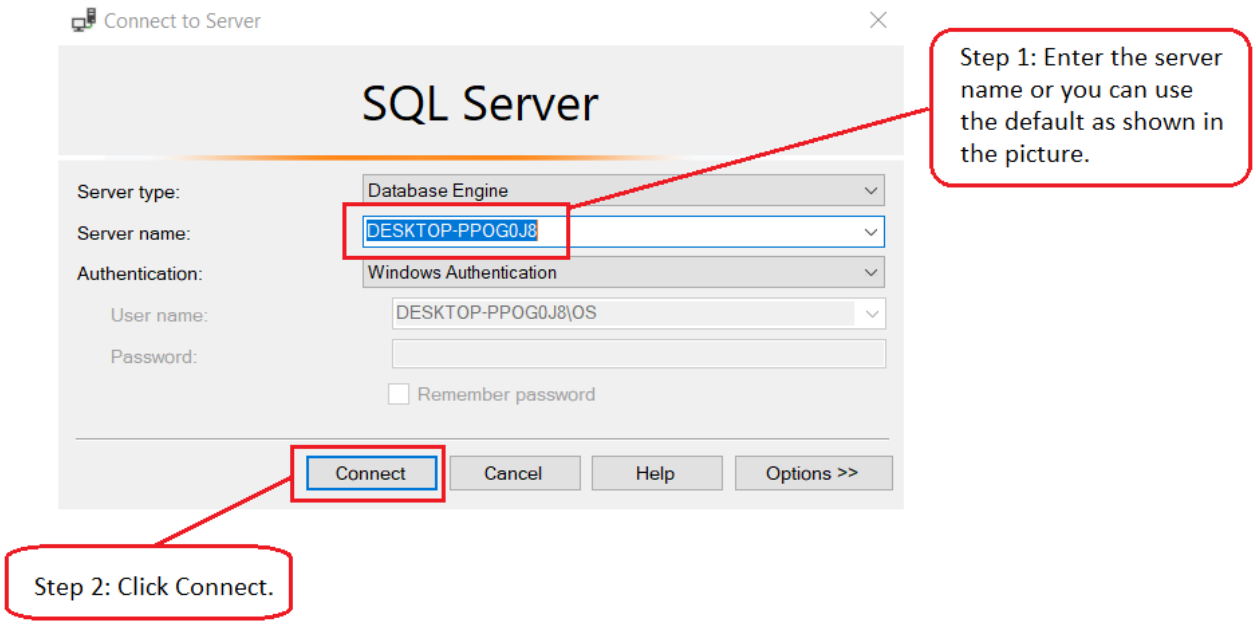
LO4 Produce technical and user documentation.

P5 Produce technical and user documentation.

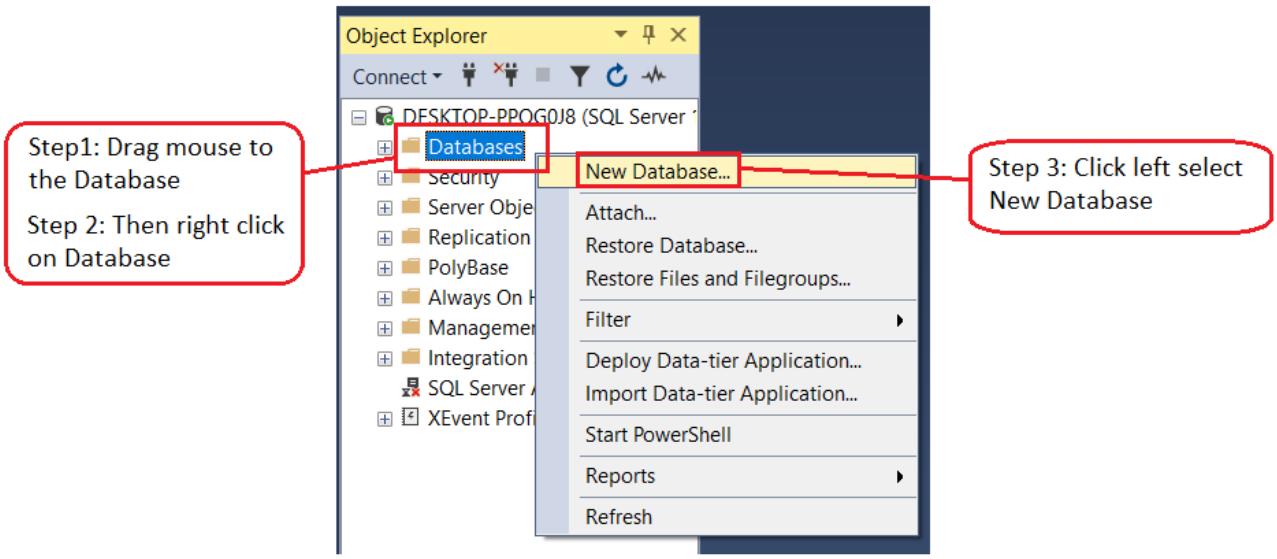
1. User document.

1.1. Create new database.

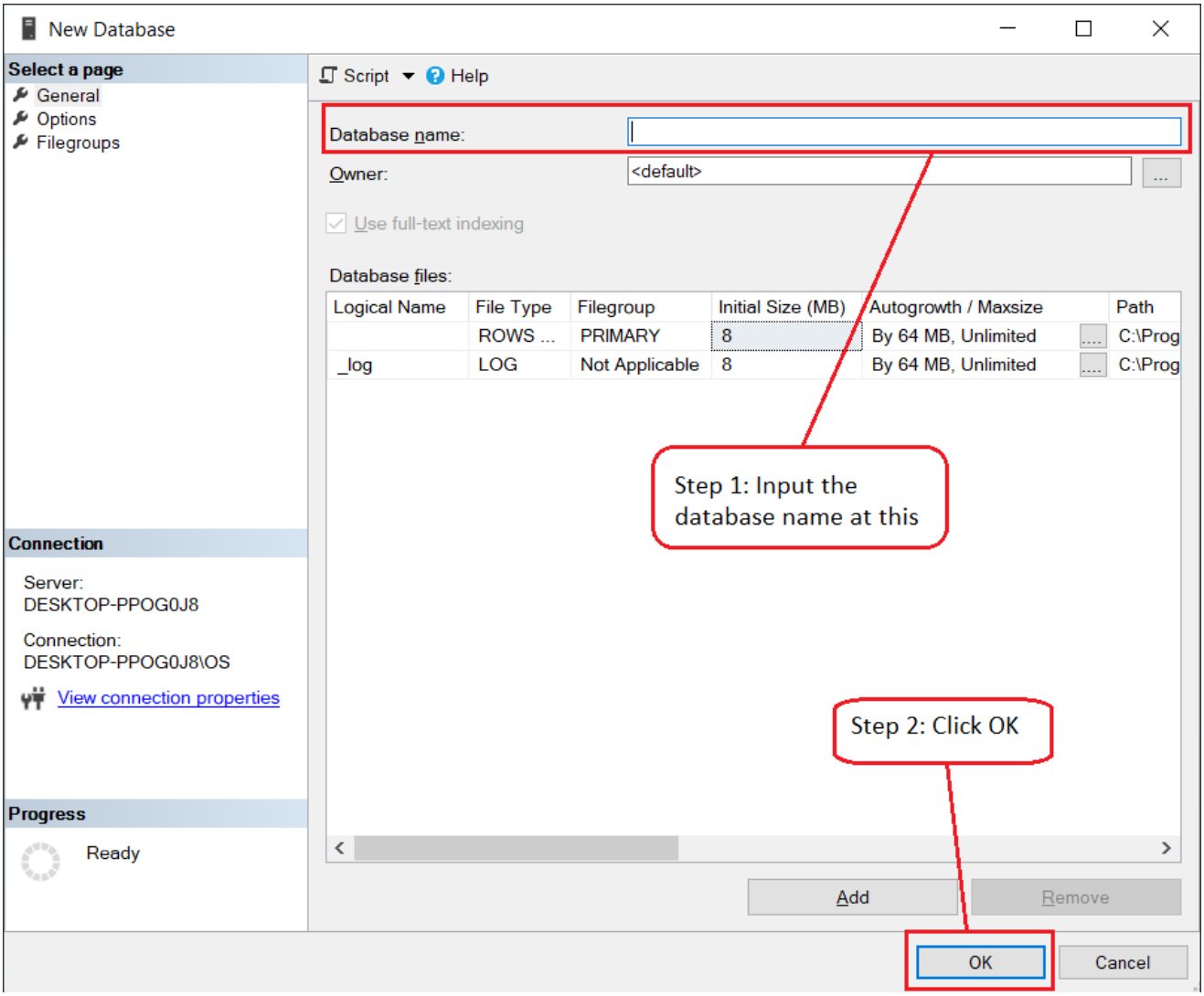
- ✓ Step 1: Open Microsoft SQL Server Management Studio
- ✓ Step 2: Connect to the server



- ✓ Step 3: Right click on Database in the Object Explorer and select New Database

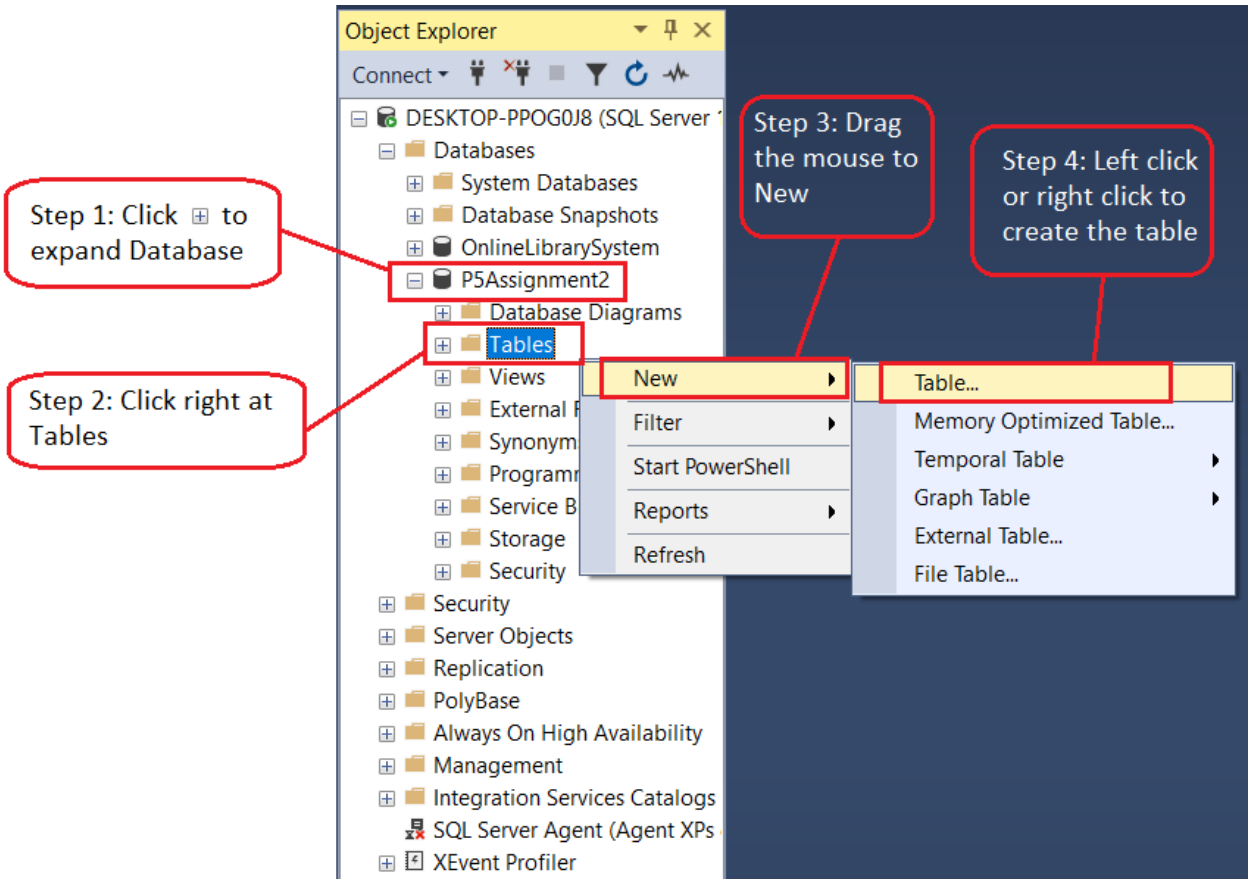


✓ Step 4: Input the database name and click OK button

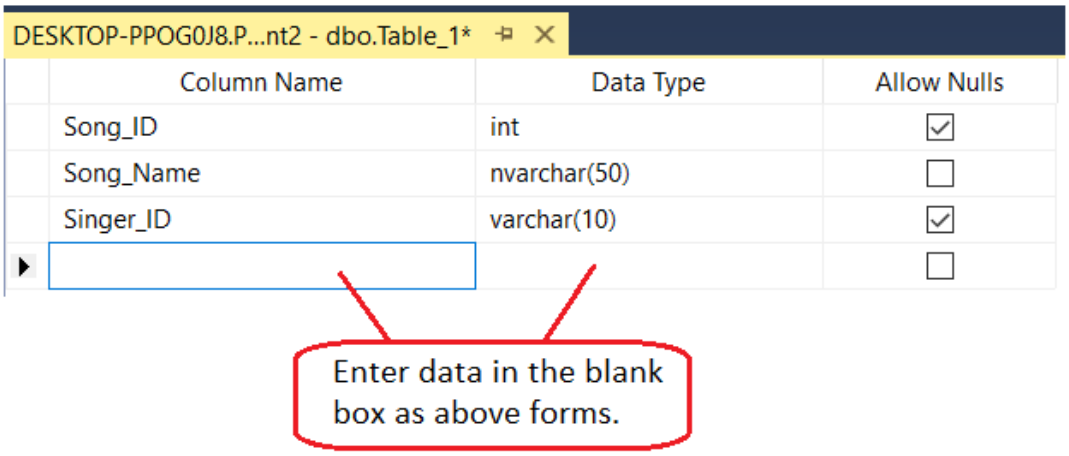


1.2. Create new table.

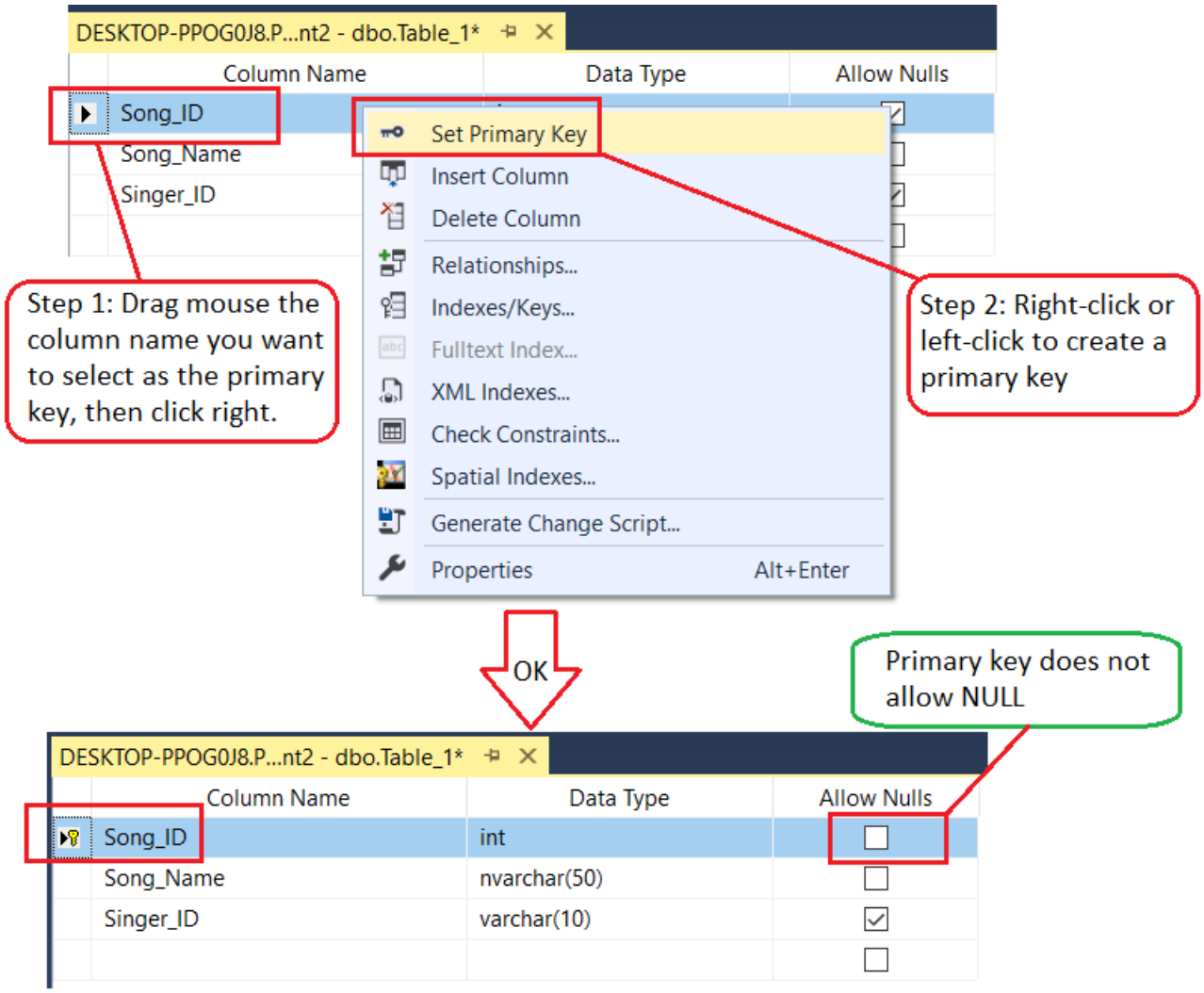
✓ Step1: Create new table



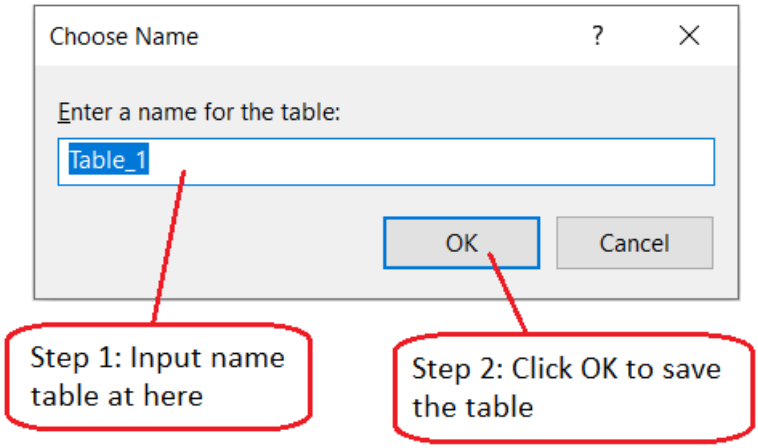
✓ Step 2: Input data and data type for column in table



✓ Step 3: Create Primary key and Foreign key (Foreign key may or may not) for table

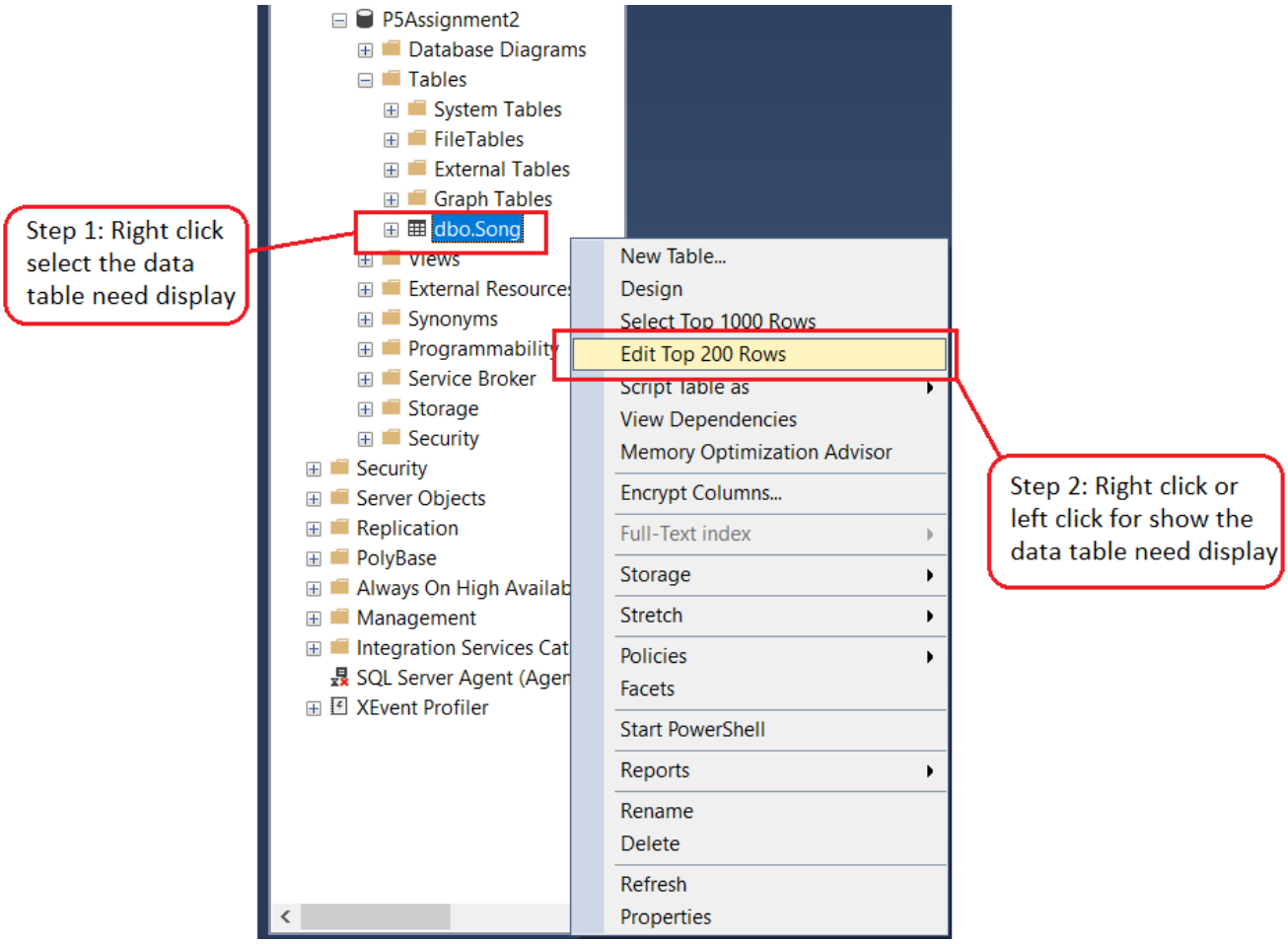


✓ Step 4: Press CTRL + S to save the table and Input name the table



1.3. Show data.

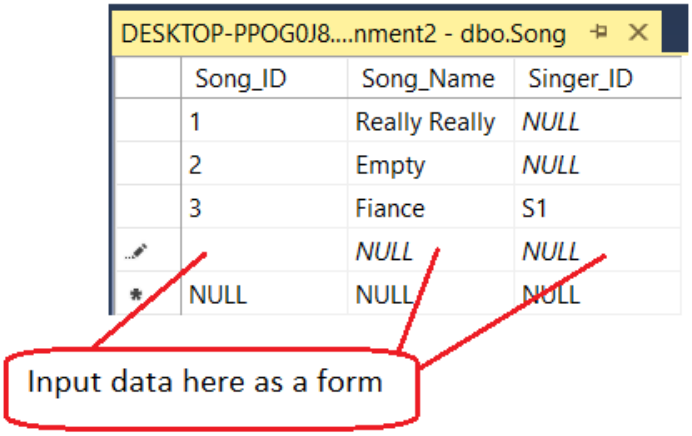
✓ Step 1: Show data



- ✓ Step 2: Close the designer and save your changes.

1.4. Insert data.

- ✓ Step 1: Open the table need insert data (This step similar as show data)
- ✓ Step 2: Input data into table

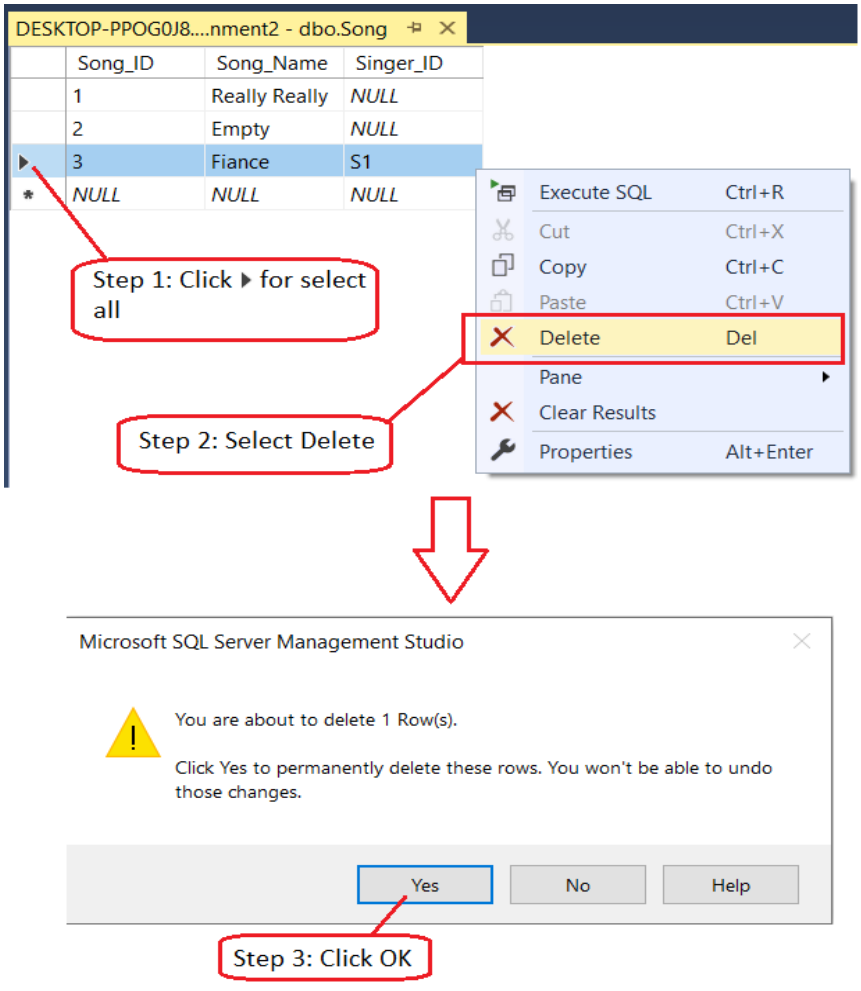


✓ Step 3: Close the designer and save your changes.

1.5. Remove data.

✓ Step 1: Open the table need Update data (This step similar as show data)

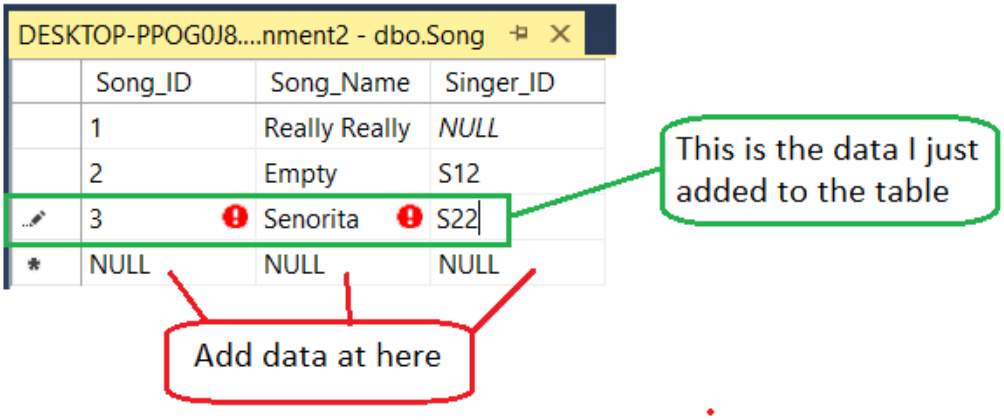
✓ Step 2: Remove data



- ✓ Step 3: Close the designer and save your changes.

1.6. Update data.

- ✓ Step 1: Open the table need Update data (This step similar as show data)
- ✓ Step 2: Update the desired data into table (You can add, delete, edit with Update. I will get example about add data)



2. Technical document.

- ✓ Step 1: Download and Install Microsoft SQL Server.

To install Microsoft SQL Server, we need to prepare:

(You must install .Net Framework 3.5 before opening the SQL Server installation file to avoid errors.)

- ➢ MS .Net Framework 3.5: (Home)

Link download: https://www.microsoft.com/en-us/download/details.aspx?id=25150

- ➢ SQL Server 2014:

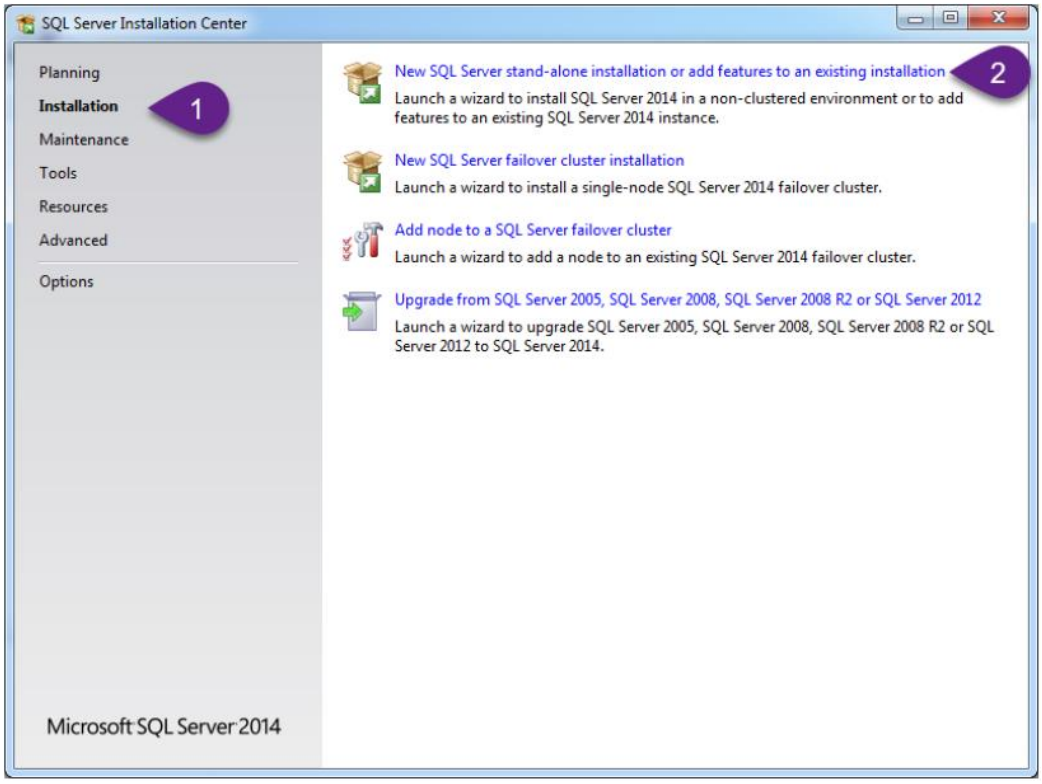
Link download: https://drive.google.com/file/d/0B64LZ7smfQ7qZjkxWVlUUmR6T2c/view

- ➢ If don't use MS .Net Framework, you must download SQL Server Management Studio (SSMS)

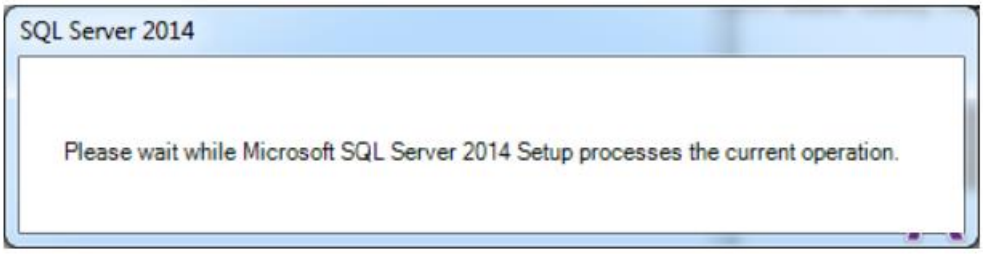
Link download: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15

To install Microsoft SQL Server, we need to follow these steps:

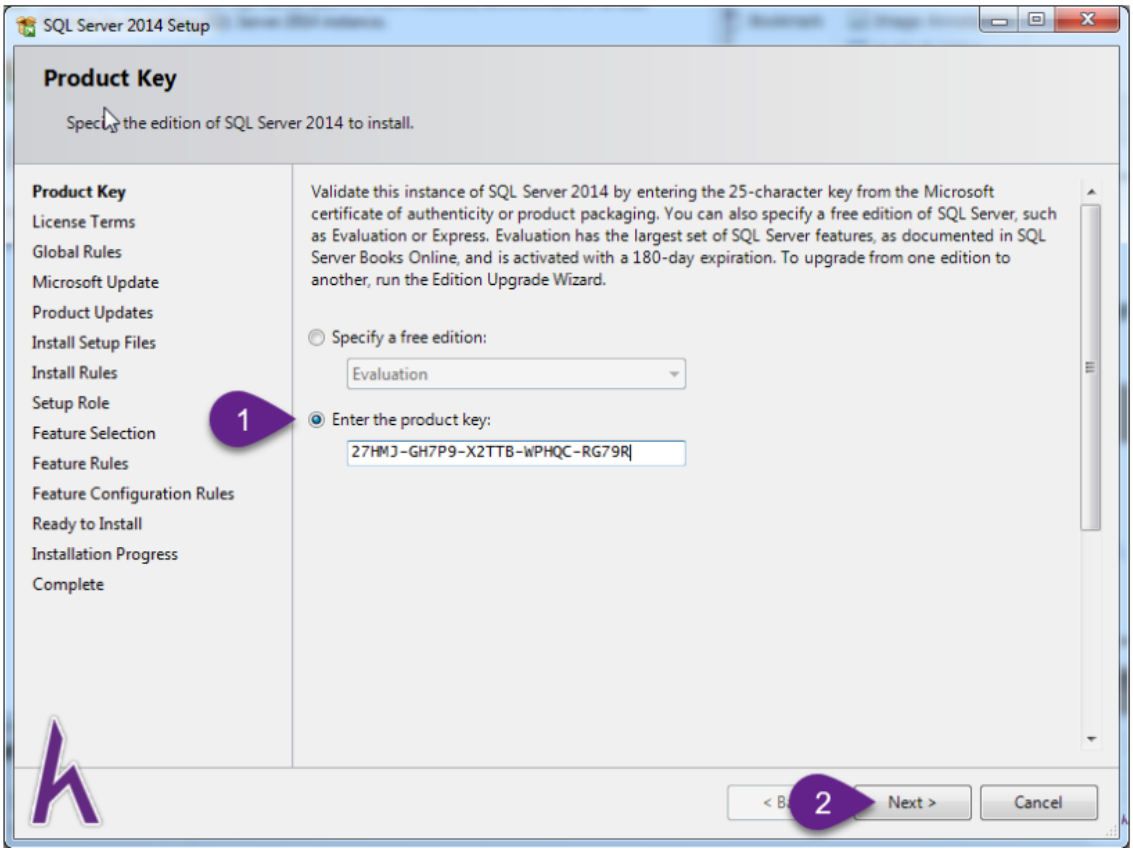
☞ After downloading you unzip the SqlServer2014.rar file then launch the setup.exe file. After the software runs up, choose as shown in the picture.



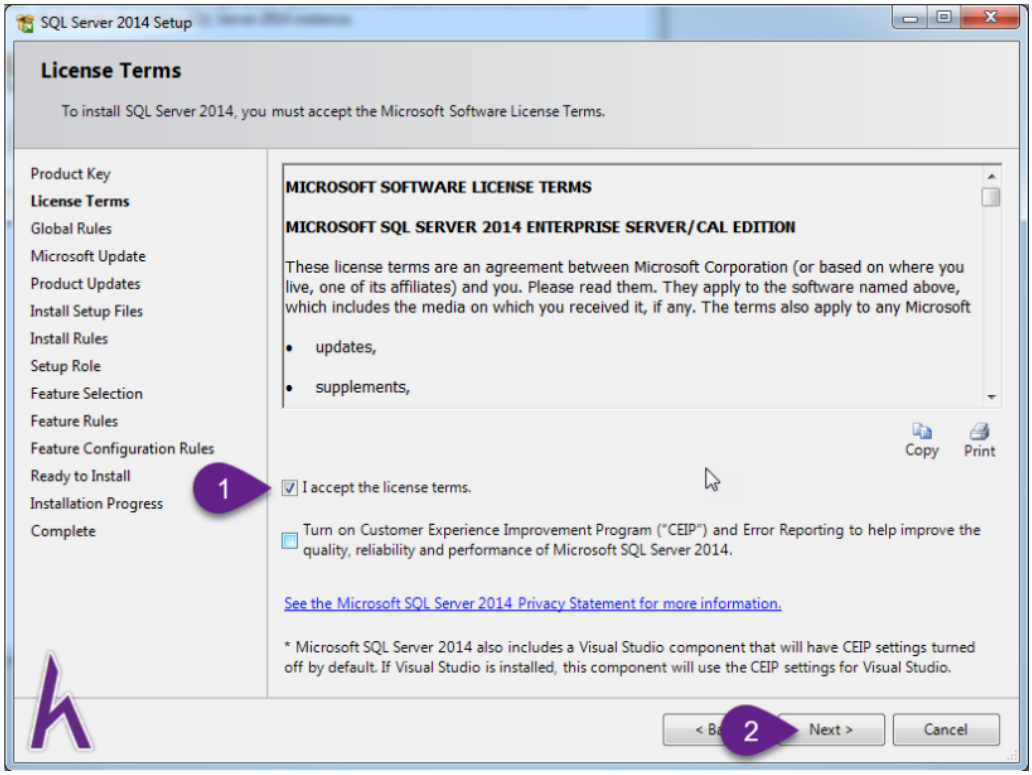
☞ The installation process begins.



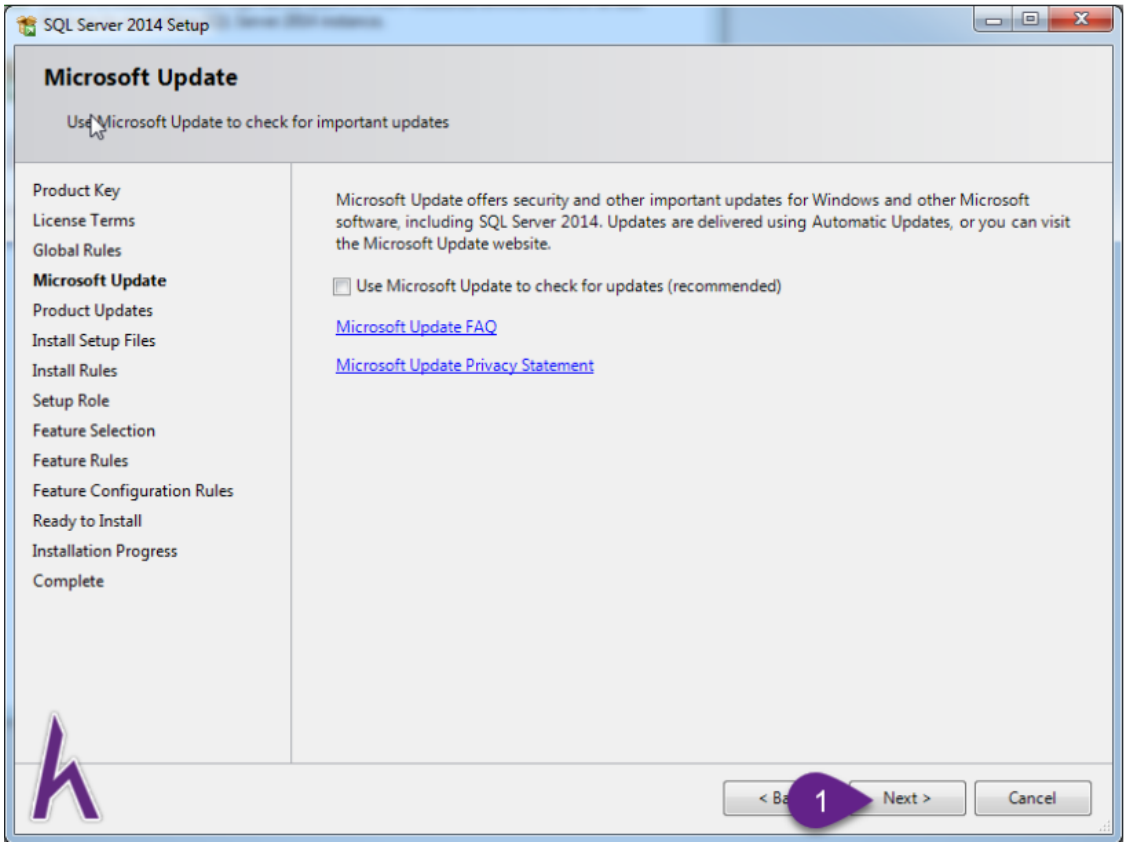
☞ The installation process starts the Enter the product key, if you download your copy the license key will automatically take into. You to click Next to continue.



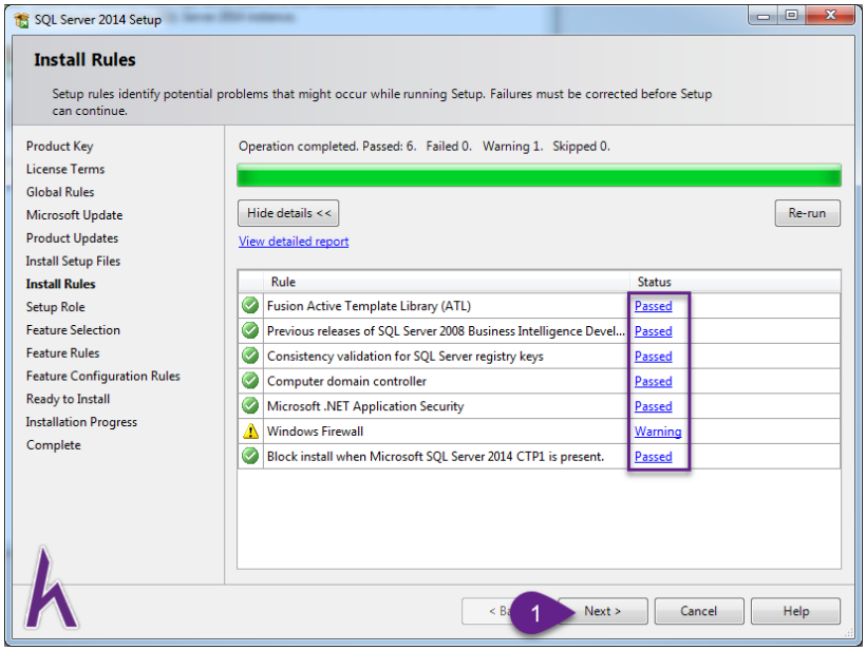
☞ You choose the I accept the license terms and deselect Turn on CEIP, click Next to continue.



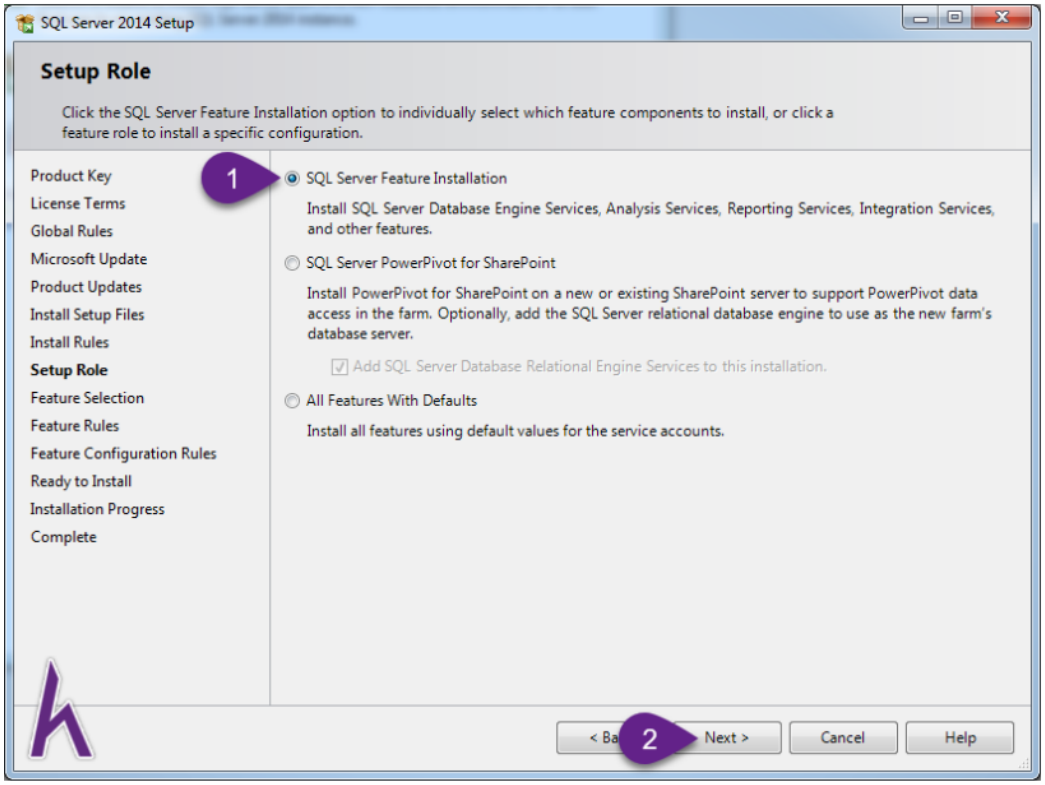
☞ In the Microsoft Update section, you should uncheck Use Microsoft Update to check for updates so that the installation process takes place faster than clicking Next to continue.



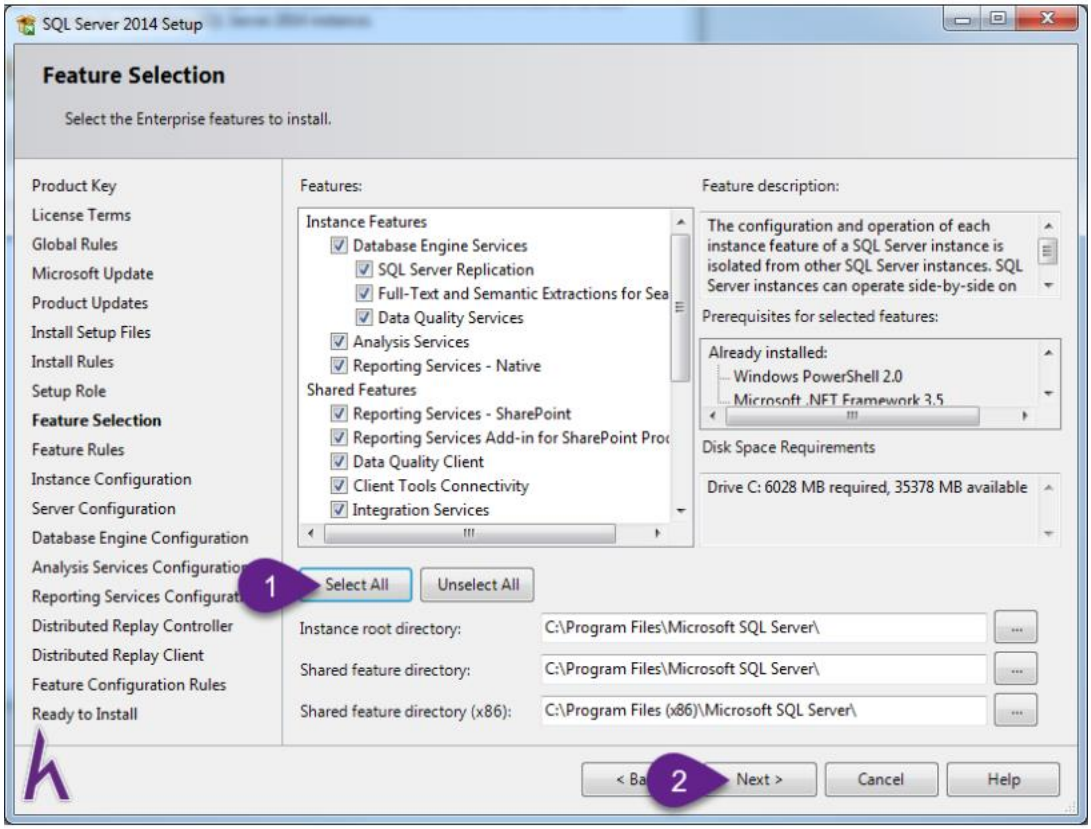
☞ The testing process starts if all Passed separately on Windows Firewall if Warning message, do not bother to click Next to continue.



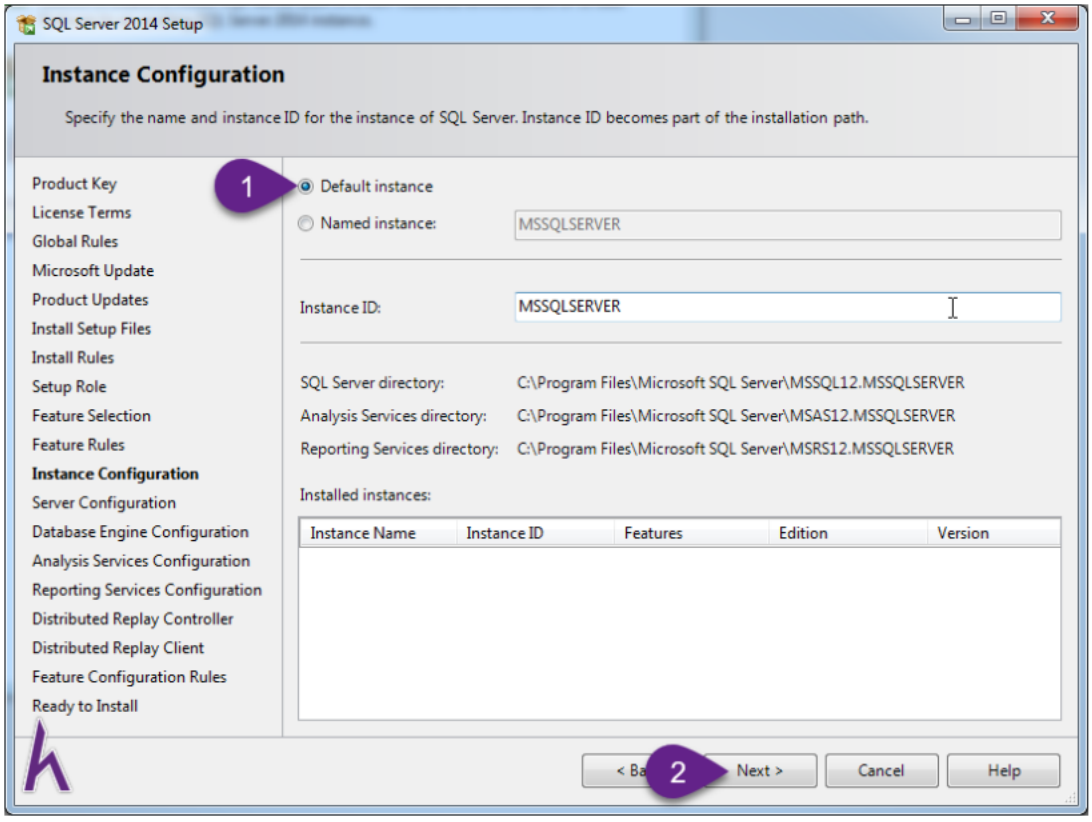
☞ You choose SQL Server Feature Installation and then click Next to continue.



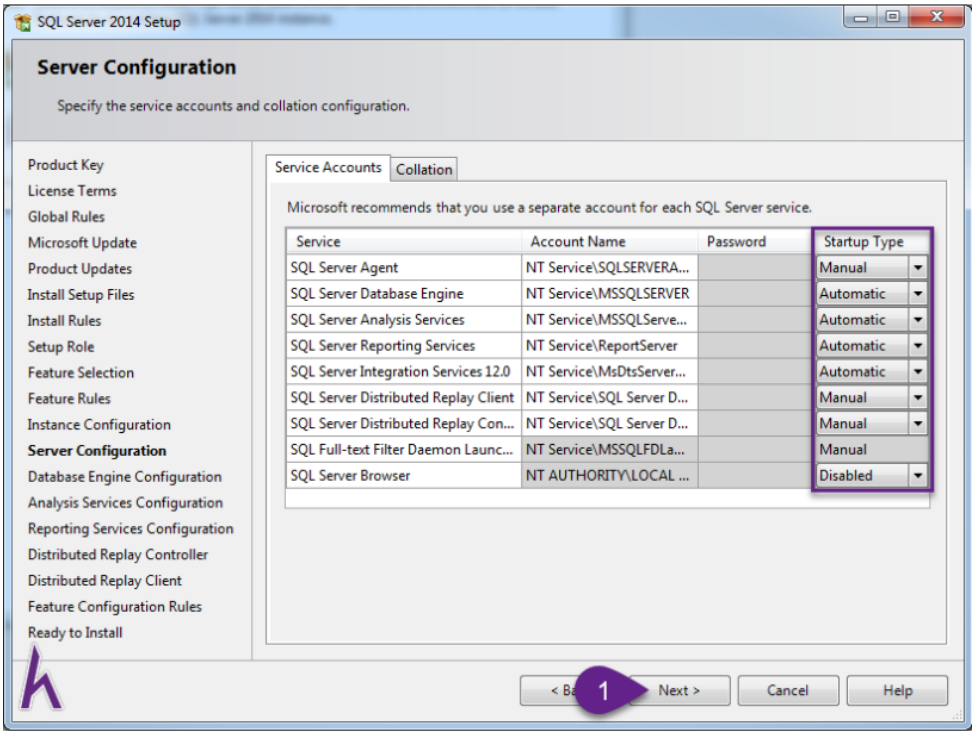
☞ You choose Select All to install the most complete and then click Next to continue.



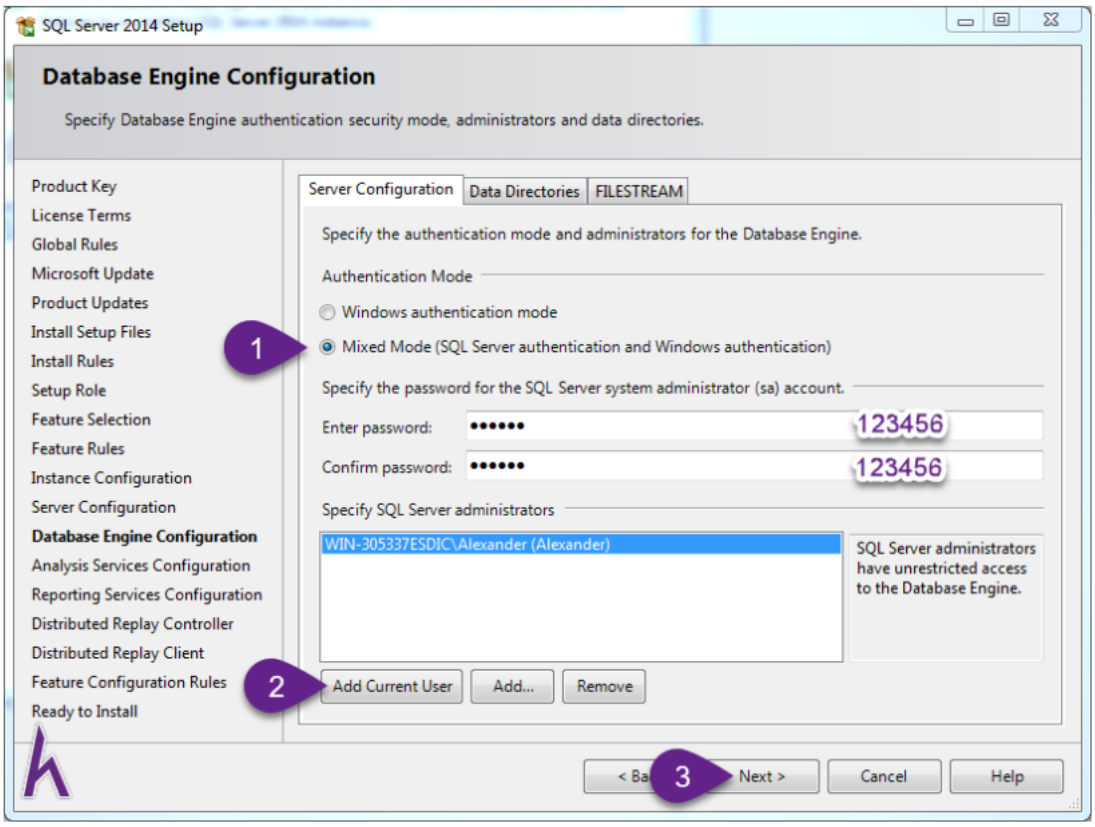
☞ You select the Default instance. In some cases, when you install SQL Server a second time, you must reset the Named instance field. Click Next to continue.



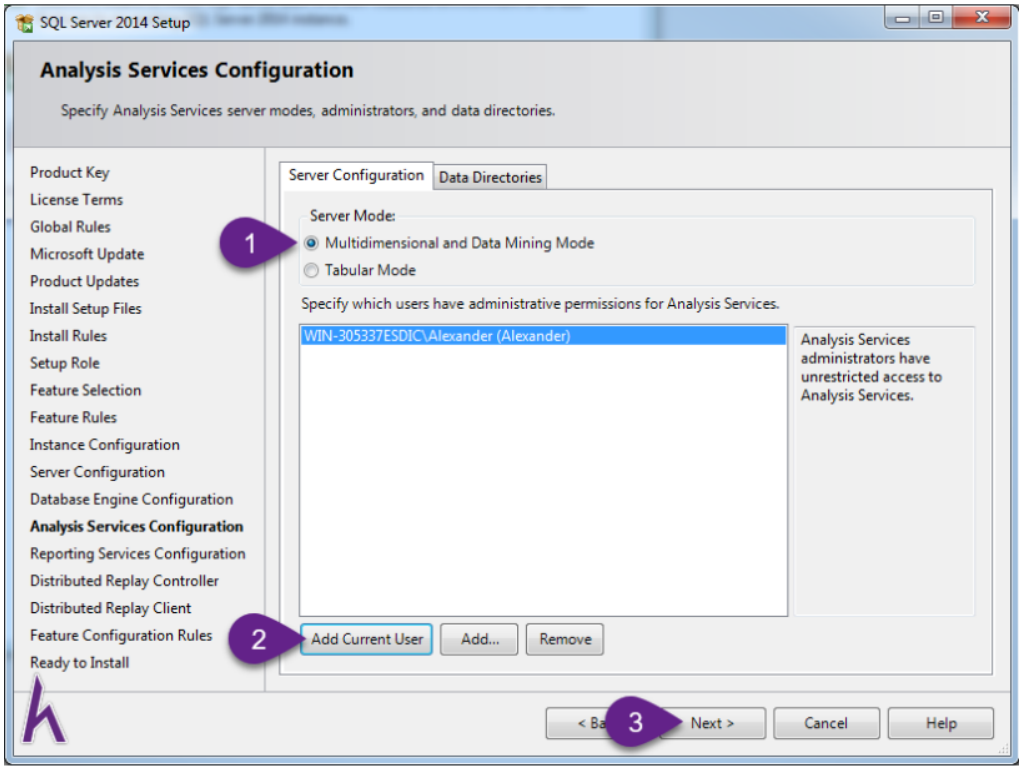
☞ Leave the default settings as they are (see picture for reference). Then click Next to continue.



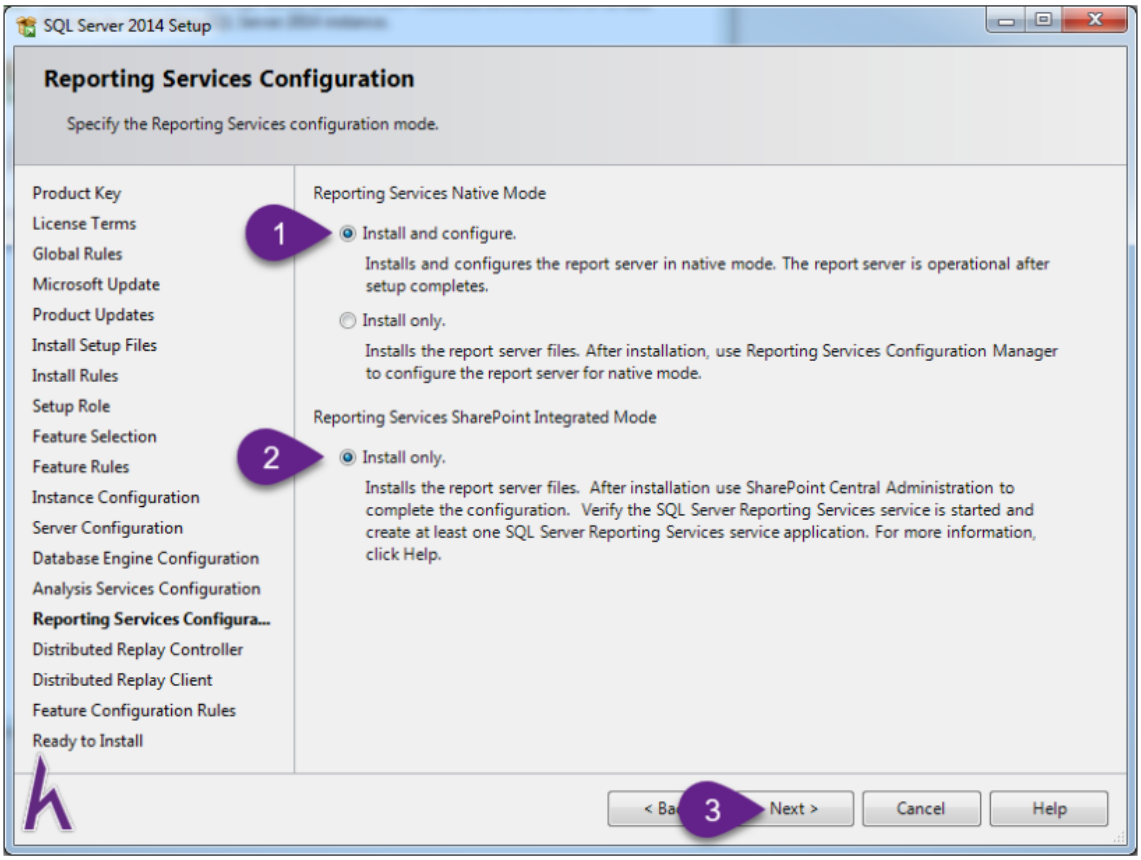
☞ On the Server Configuration tab, select Mixed Mode, then set the password for the two empty boxes below (It is recommended to set 123456 to make it easier to remember) then click Add Current User to let it add your computer user. Then click Next to continue.



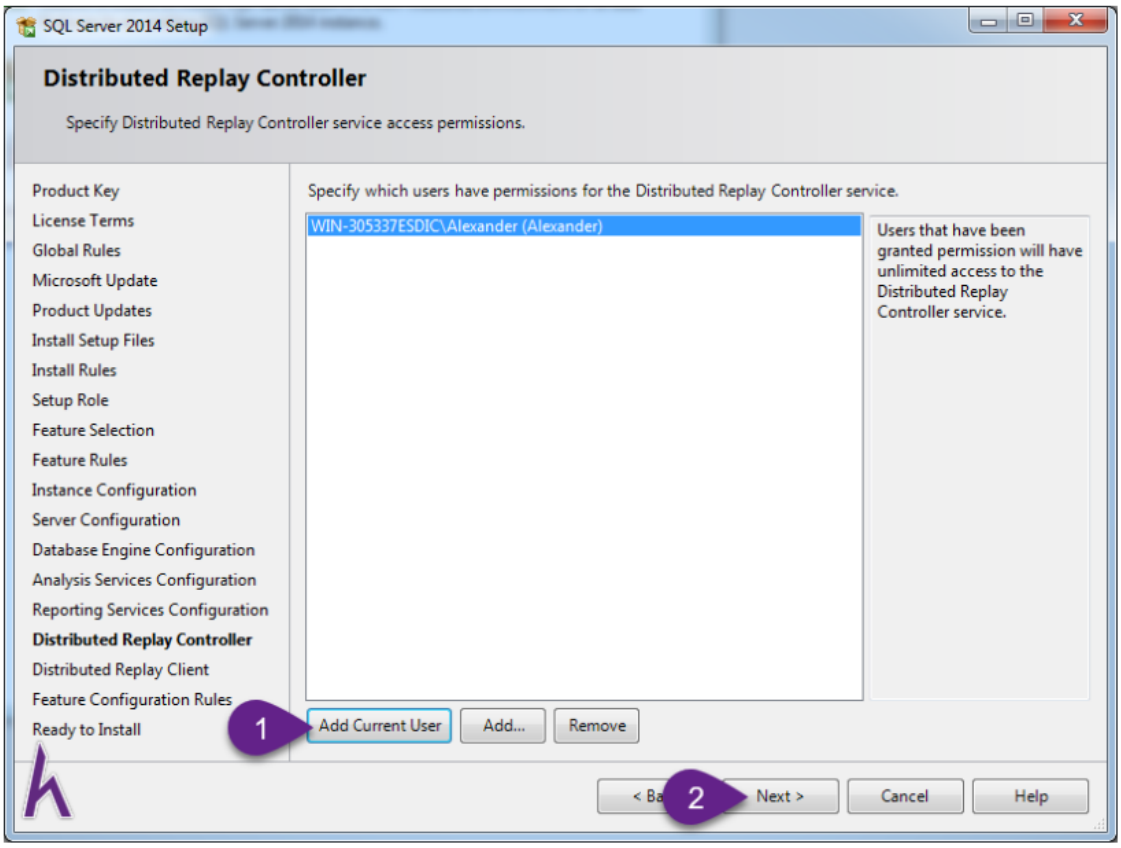
☞ In the Server Configuration tab select Multidimensional and Data Mining Mode then click Add Current User to add it to your computer user. Then click Next to continue.



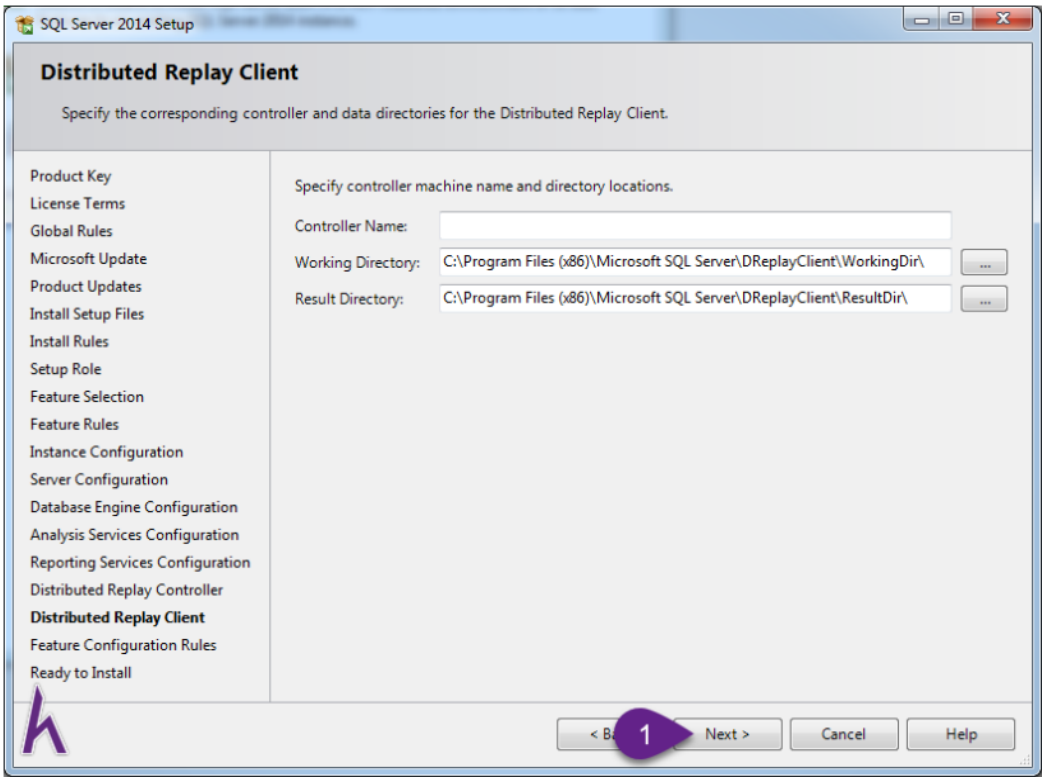
☞ You choose 'Install and configure' and Install only then click Next to continue.



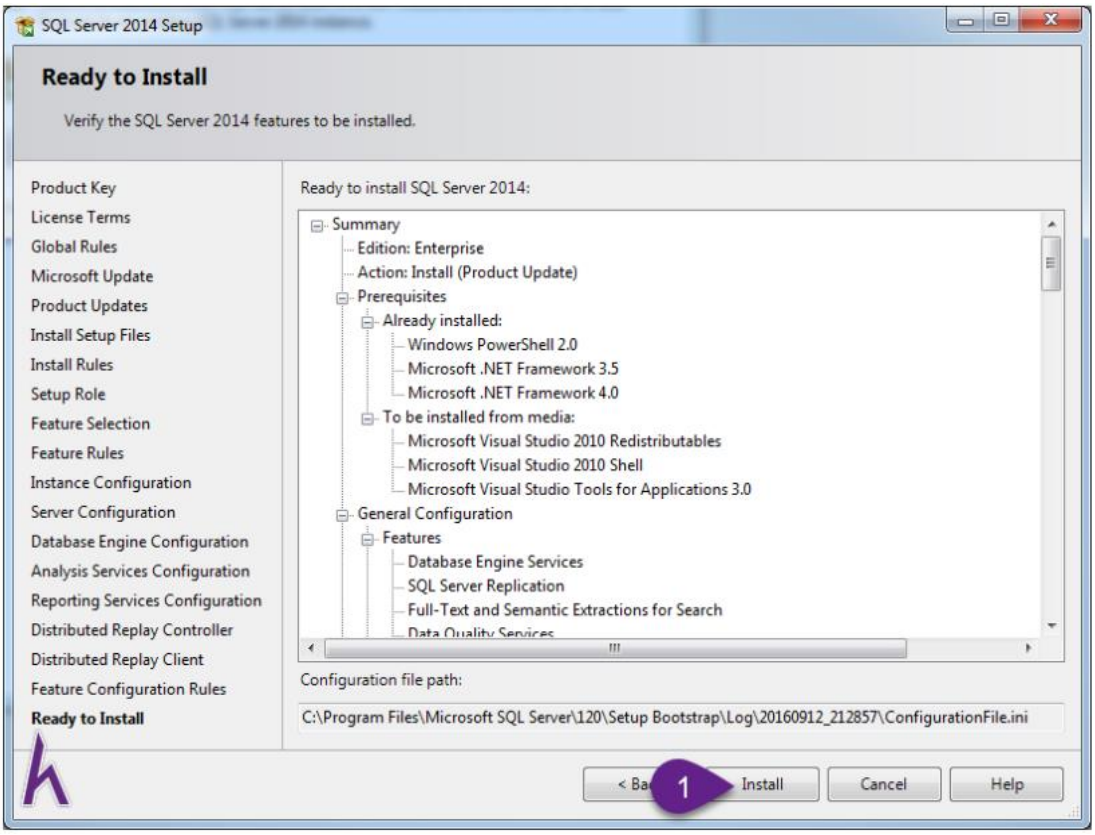
☞ You select Add Current User to add your user. Click Next to continue.



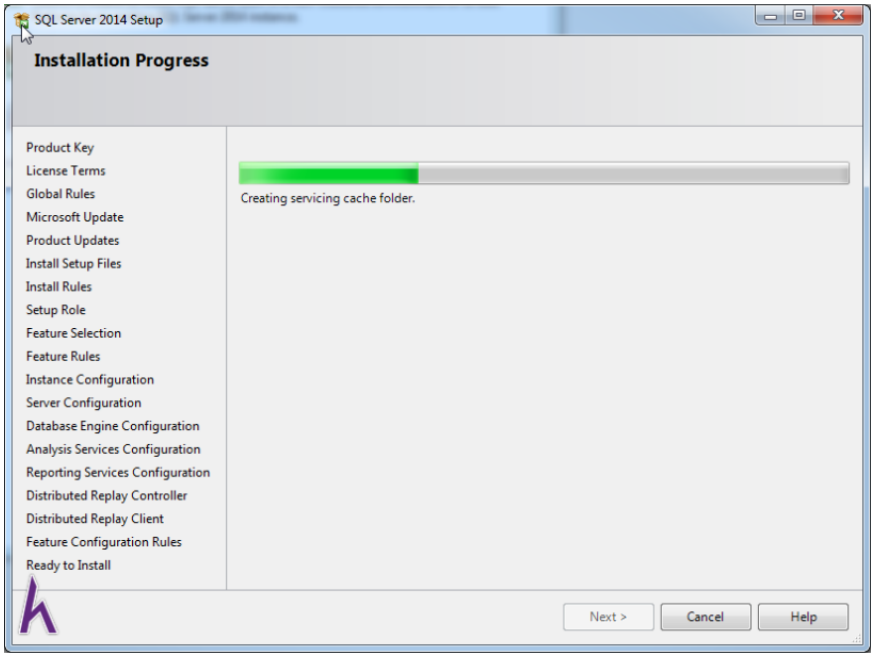
☞ Leave the default to click Next to continue.



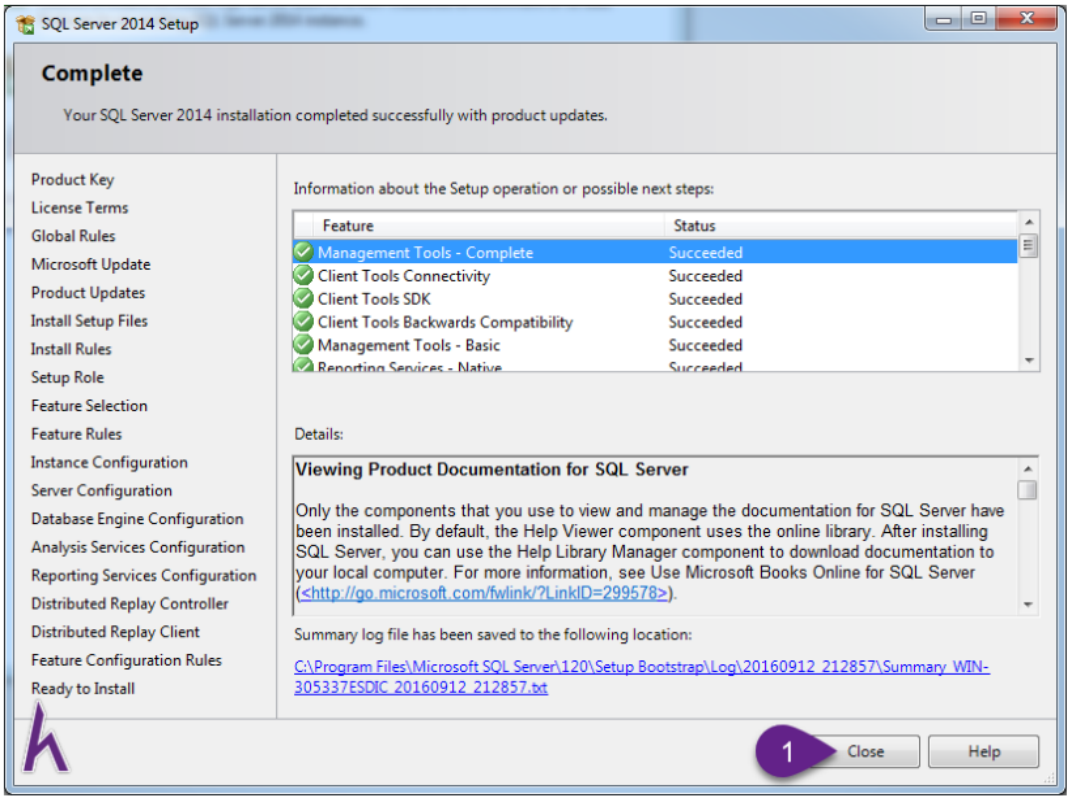
☞ Click Install to install the process.



☞ You wait for the installation to take place normally from 15 minutes - 30 minutes depending on your configuration.



☞ The installation process is successful, click Close to finish.



✓ Step 2: Open Microsoft SQL Server Management Studio and connect to server.

'You start Microsoft SQL Server 2014 Management Studio to continue'

- Set up your account under SQL Server Authentication (Login with your database account).

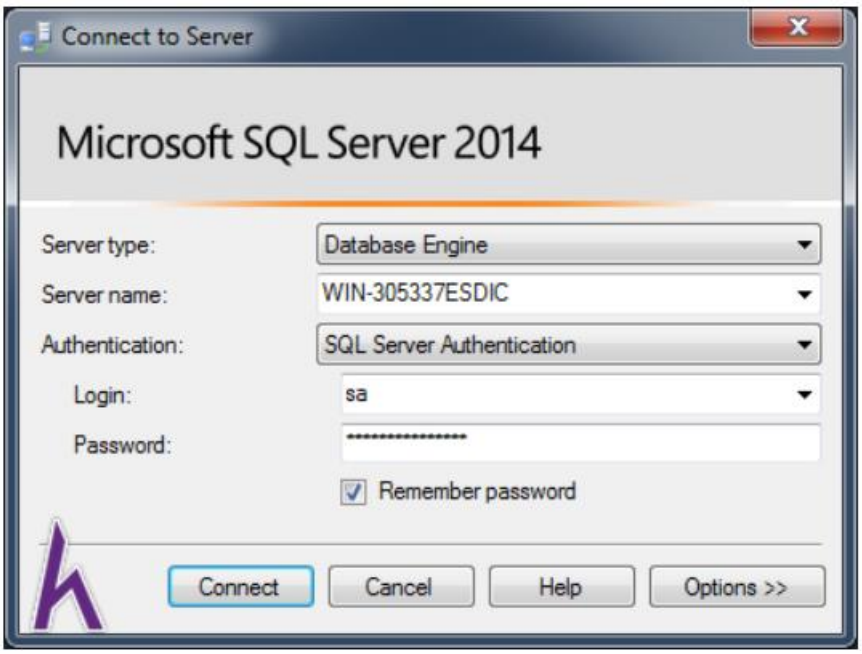
  - For Server type: select Database Engine.

  - Part Server name: you choose your hostname.

  - Part Authentication: you select SQL Server Authentication.

  - Part Login: you enter 'sa' is the default when you install.

  - Part Password: you enter the password when you set the default is '123456'



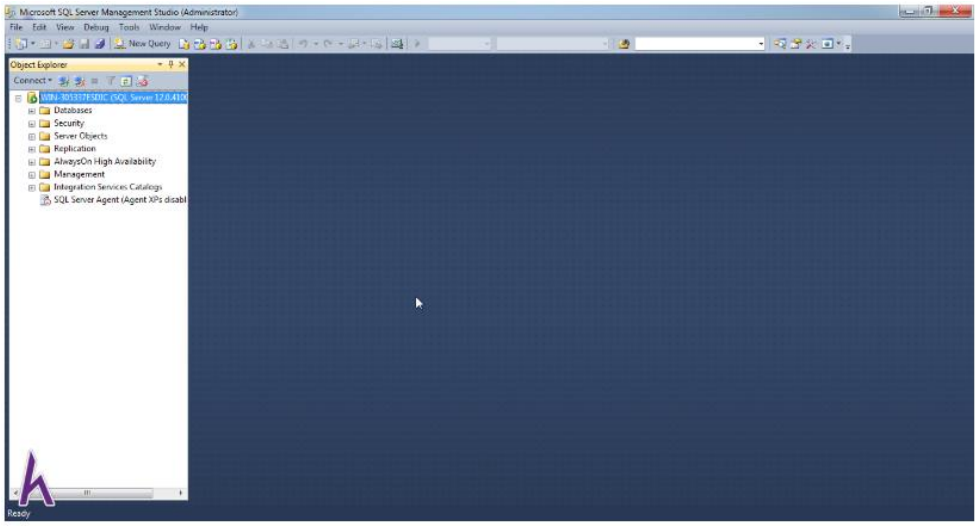
- Set up your account under Windows Authentication (Login with your machine account).

  - For Server type: select Database Engine.

  - Part Server name: you choose your hostname.

  - Part Authentication: you choose Windows Authentication.

- After successful connection, we will be presented with the following image.



✓ Step 3: Writing tutorial how to create new database by SQL code

Syntax:

```
CREATE DATABASE database
name;
```

Example:

```
CREATE DATABASE testDB;
```

✓ Step 4: Writing tutorial how to create new table by SQL code

Syntax:

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

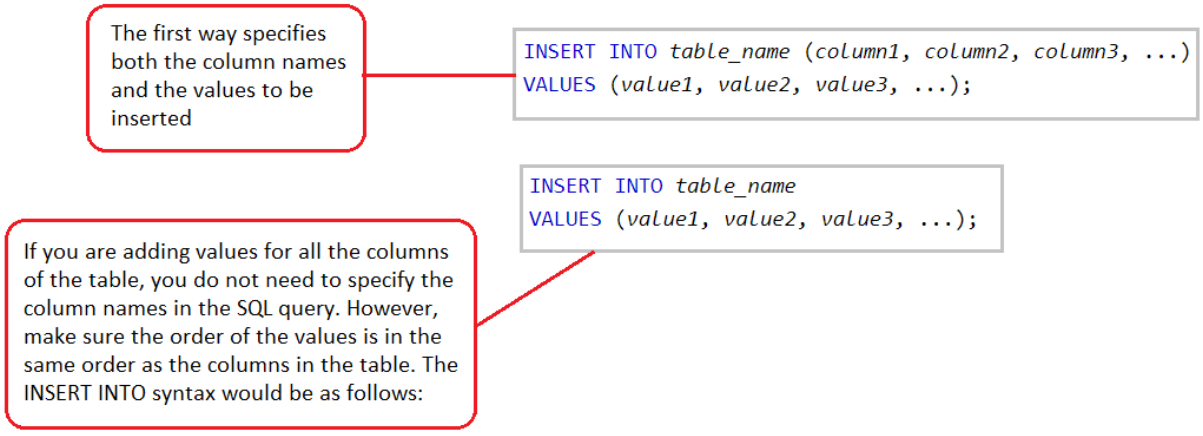
Example:

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
```

```
    [Address] varchar(255),
    City varchar(255)
);
```

✓ Step 5: Writing tutorial how to insert data into the table by SQL code

Syntax: There are two ways main.

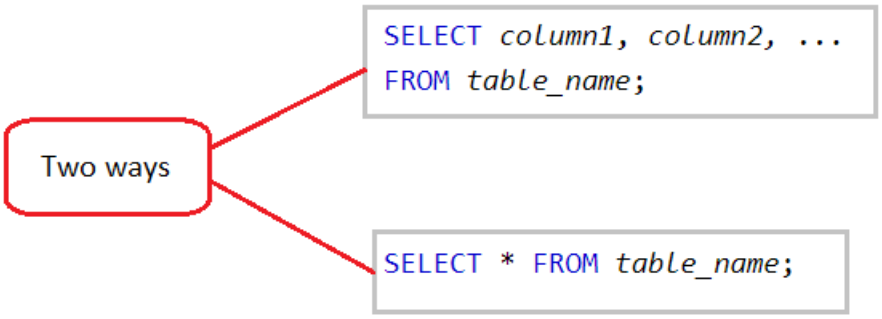


Example:

```
--Ways 1
INSERT INTO Persons(LastName, FirstName, [Address], City)
VALUES ('Kim', 'Jinu', 'Seoul', 'Korea');
--Ways 2
INSERT INTO Persons
VALUES ('1', 'Lee', 'Seunghoon', 'Busan', 'Korea');
```

✓ Step 6: Writing tutorial how to show data from tables by SQL code

Syntax: Use SELECT (Show data simple)



Example:

```
--Ways 1
SELECT City, LastName FROM Persons;
--Ways 2
SELECT * FROM Persons;
```

✓ Step 7: Writing tutorial how to update existed data in the table by SQL code

Syntax:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Example:

```
UPDATE Persons
SET LastName = 'Kang', FirstName= 'SeungYoon'
WHERE PersonID = 1;
```

✓ Step 8: Writing tutorial how to remove data from tables by SQL code

Syntax:

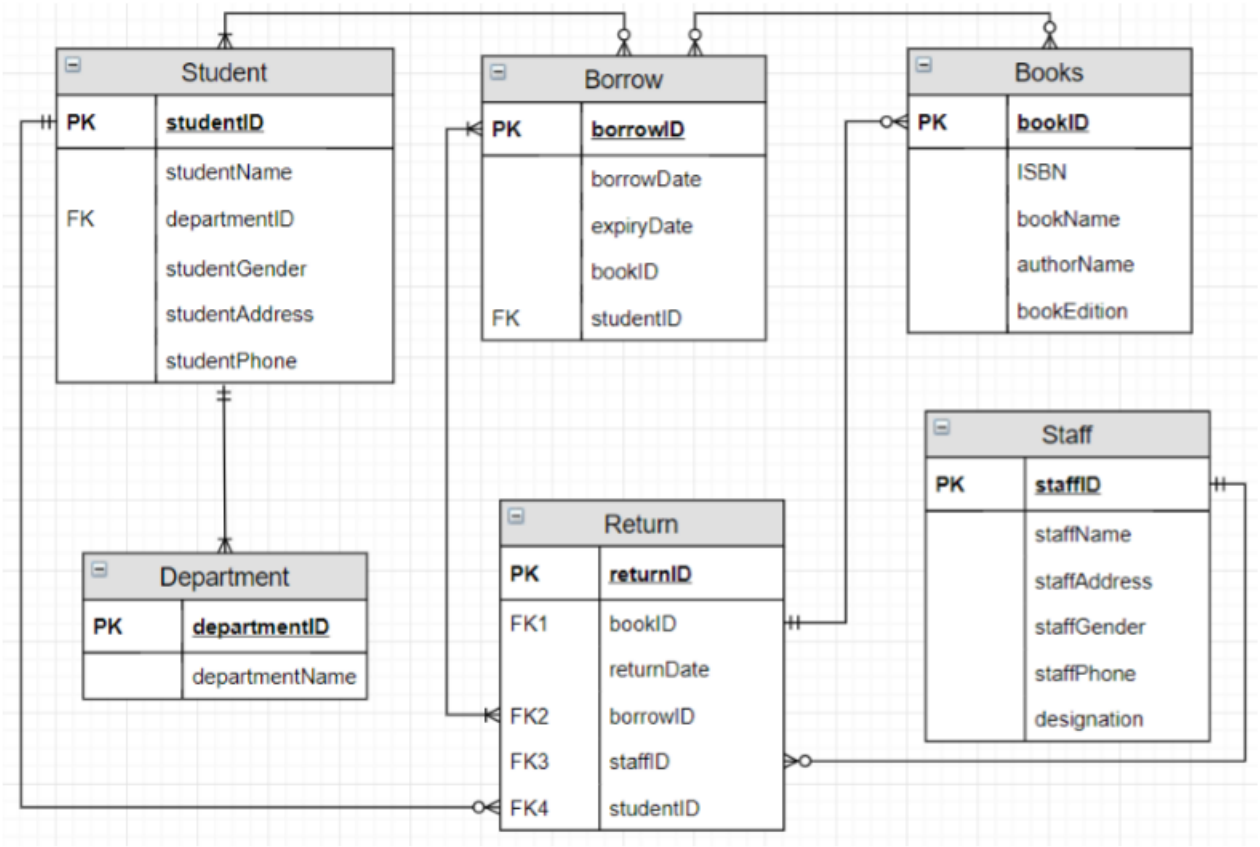
```
DELETE FROM table_name WHERE condition;
```

Example:

```
DELETE FROM Persons WHERE PersonID='1';
```

M5 Produce technical and user documentation for a fully functional system, including ER Diagram and normalization statements and describing how the system works.

1. ER Diagram



2. Normalization statements

Database normalization is the process of eliminating redundant data in tables to improve storage efficiency, data integrity, and scalability. It usually involves splitting the existing tables into multiple table the table rejoin or link every time a query is given.

• First normal form: the given table is converted into 1NF as follows.

- ✓ Step 1: Remove the duplicate columns from table 1.
- ✓ Step 2: Create separate tables for each group of related data and identify each row group with a unique column (primary key).

• Second normal form: A table is in first normal form and each non-key field is functionally dependent upon primary key.

• Third normal form (3NF) requires that there are no functional dependencies of non-key attributes on something other than a candidate key. A table is in 3NF if all of the non-primarykey attributes-mutually independent. There should not be transitive depeinces.

Normalization of database tables in library system: (Some examples)

Before 1st data normalization:

borrowID	bookID	studentID
1122	110, 120, 320, 303	bitE183

=> In the Borrowed Table there is repeating books. A student has borrowed 4 books.

After 1st data normalization:

borrowID	bookID	studentID
1122	110	bitE183
1122	120	bitE183
1122	320	bitE183
1122	303	bitE183

Before 2nd data normalized:

In the following Student relation, all attributes are dependent on the primary key StudentID.

StudentID	Name	DepartmentID	borrowDate	exparyDate
BITf13E18	Azhar	20	17-6-15	1-7-15

After 2nd data normalized:

Can create two other relations from Student Table.

Table1

departmentID	departmentName
11	CS & IT Department
22	Education Department

Table2

StudentID	Name	borrowDate	exparyDate
BITf13E18	Azhar	17-6-15	1-7-15

Before 3rd data normalized:

studentID	Name	Gender	departmentID	Address	Phone	City

After 3rd data normalized:

Table1

studentID	departmentID	departmant
IT-113	C-26	Cs & IT
Lm-456	L-11	Law

Table2: Student contact table.

studentID	City	Phone	Address
IT-111	Sargodha	0331236547	Statlite Twon
Cs-786	Sargodha	0327162226	Statlite Twon

Table3: Student

sdtID	Name	Gender	sdtDepartment
IT-111	Mino	Female	CS & IT
Cs-786	Yoon	Female	Education

3. Describing how the system works

- This is a library management system at school to handle all transactions at the library. It must suitable for use by small to medium sized libraries. It is used by library administrators and librarians to manage libraries using computer systems. Problems like file loss will not occur.

- Staff will be the person who directly uses and manages the online library system. Staff is responsible for importing all student data and information, books, etc.

- Example 1, when a student borrowed a book. Staff will have to enter book information, student information, loan date and return date into the archive system.

- Example 2, when a new batch of books is imported, Staff must enter all the information of those books (such as book ID, title, author, etc.) into the storage system.

D3 Assess any future improvements that may be required to ensure the continued effectiveness of the database system.

- First need to make a system maintenance plan. Second, regularly create surveys to evaluate the system. Finally, from the survey results, offer the optimal plan and solution for the system to meet the needs of the market and tastes.

- Predictive improvement and evaluation: Functional improvement needs to meet the needs of users. Improvements in the system need to survey the market for new products that can support the system. Help the system always to be updated and refreshed. From here, the system can work for a long time and not useless compared to this era of innovation.

REFERENCES

- www.howkteam.com. 2020. *How Kteam - Free Education | Website Hướng Dẫn Lập Trình Miễn Phí | How Kteam | Free Education*. [online] Available at: <https://www.howkteam.vn/> [Accessed 13 March 2020].
- YouTube. 2020. *K Team*. [online] Available at: <https://www.youtube.com/channel/UCBw4b26KZrBvHRPBjOCw6UQ> [Accessed 13 March 2020].
- freetuts. 2020. *Thủ Thuật PC / Mobile / Lập Trình / Review - Freetuts.Net*. [online] Available at: <https://freetuts.net/> [Accessed 13 March 2020].
- chủ, T., 2020. *Database*. [online] freetuts. Available at: <https://freetuts.net/database> [Accessed 13 March 2020].
- www.howkteam.com. 2020. *Kết Quả Tìm Kiếm | Website Hướng Dẫn Lập Trình Miễn Phí | How Kteam | Free Education*. [online] Available at: <https://www.howkteam.vn/search?tab=&q=sql> [Accessed 13 March 2020].
- Drive.google.com. 2020. *Slides – Google Drive*. [online] Available at: <https://drive.google.com/drive/folders/1yeu3NCIDU5DzEbonOF4EQbz2GWA6OuIn> [Accessed 13 March 2020].
- @Copyright Laptrinhvb.net. 2020. *[SQL SERVER] TOP 50 CÂU LỆNH SQLSERVER QUAN TRỌNG NÊN BIẾT (PHẦN 1)*. [online] Available at: <https://laptrinhvb.net/bai-viet/co-so-du-lieu/--SQL-SERVER---TOP-50-CAU-LENH-SQLSERVER-QUAN-TR0NG-NEN-BIET-(PHAN-1)/33d2d15d8a3cd32f.html> [Accessed 14 March 2020].
- www.howkteam.com. 2020. *Hướng Dẫn Cài Đặt SQL Server | Website Hướng Dẫn Lập Trình Miễn Phí | How Kteam | Free Education*. [online] Available at: <https://www.howkteam.vn/course/huong-dan-cai-dat/huong-dan-cai-dat-sql-server-102> [Accessed 15 March 2020].
- W3schools.com. 2020. *SQL Tutorial*. [online] Available at: <https://www.w3schools.com/sql/> [Accessed 15 March 2020].
- 2020. [online] Available at: <https://youtu.be/zPbMKmdFwnY> [Accessed 15 March 2020].
- 2020. [online] Available at: <https://youtu.be/q56I-xBCaiQ> [Accessed 15 March 2020].
- 2020. [online] Available at: <https://youtu.be/LWyyI4TMFII> [Accessed 15 March 2020].