Full Stack Application Development with Cloud Computing

Module 5 – Azure Data Services

Lab Practical Manual

Unit 1 - Azure Database Services (e.g., Azure SQL Database, Cosmos DB

Topic: Data Analytics – Solved Question

Ex 1: Data Analytics on Microsoft Azure Cloud

Learning Objective In

this you will:

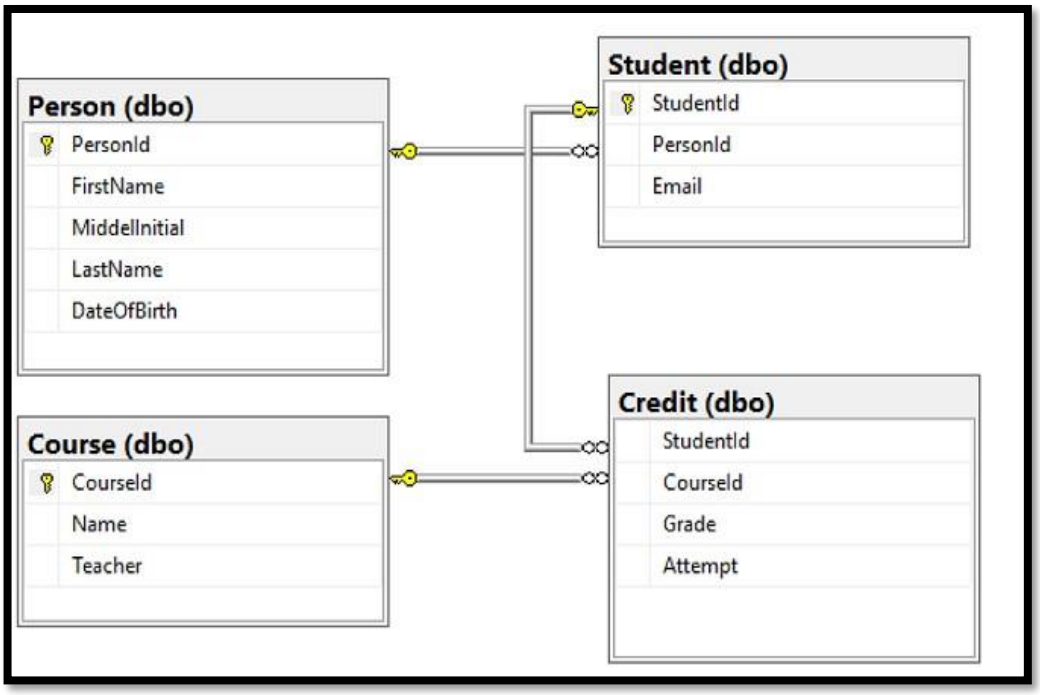
- Create a database schema
- Understand perform Data Analytics on Microsoft Azure Cloud

Create a database schema with four tables that model a student management system for universities using Transact-SQL:

- Person
- Course
- Student
- Credit

The following diagram shows how these tables are related to each other. Some of these tables reference columns in other tables. For example, the Student table references the PersonId column of the Person table. Study the diagram to understand how the tables in this tutorial are related to one another. For an in-depth look at how to create effective database tables, see Create effective database tables. For information about choosing data types, see Data types.

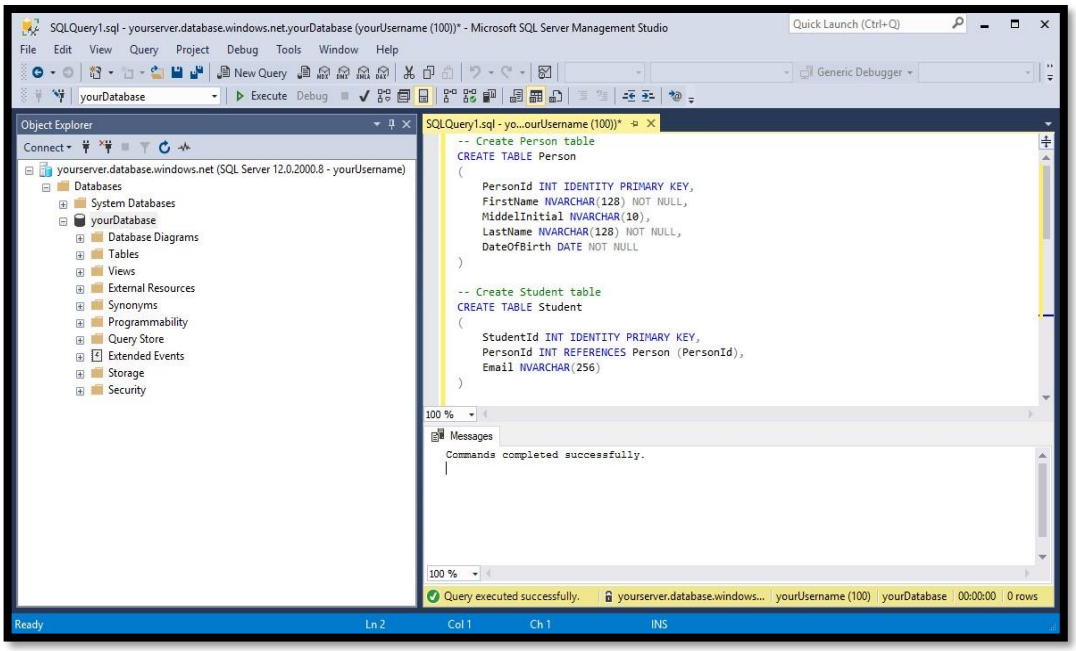
Note: You can use the table designer in SQL Server Management Studio to create and design your tables.



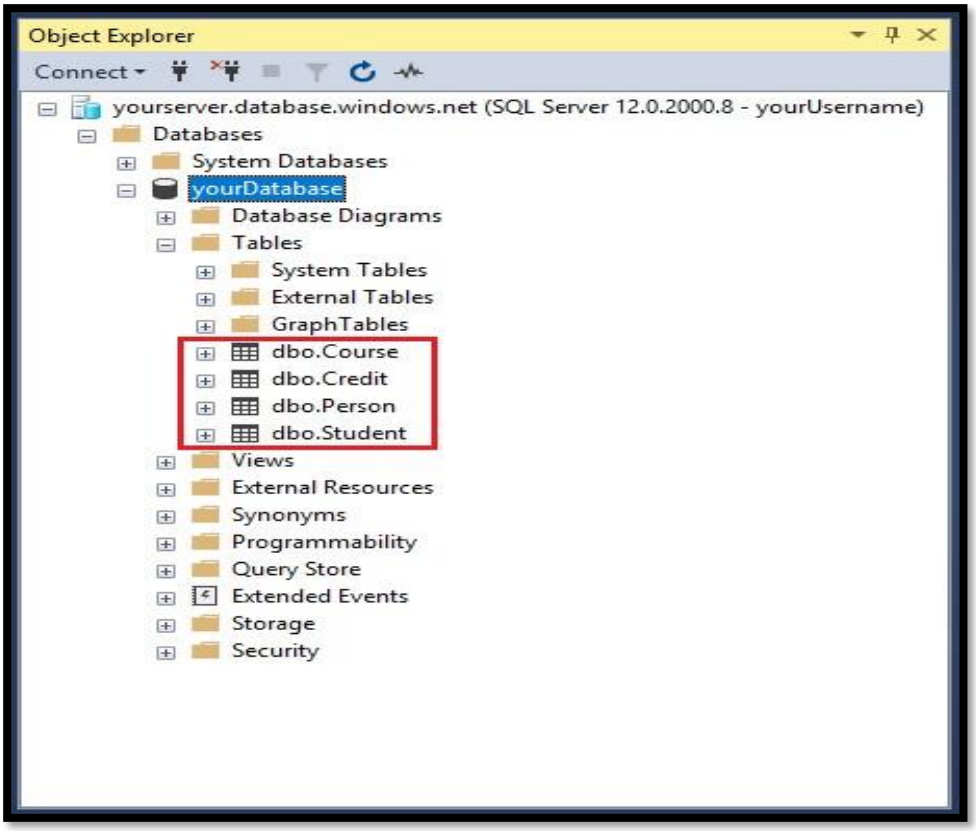
1. In Object Explorer, right-click yourDatabase and select New Query. A blank query window opens that is connected to your database.
2. In the query window, execute the following query to create four tables in your database:

```
SQLCopy

-- Create Person table

CREATETABLE Person

(

PersonIdINTIDENTITY PRIMARY KEY,

    FirstName NVARCHAR(128) NOTNULL,

MiddelInitialNVARCHAR(10),

LastNameNVARCHAR(128) NOTNULL,

DateOfBirthDATENOTNULL

)

-- Create Student table

CREATETABLE Student

(

StudentIdINTIDENTITY PRIMARY KEY,
```

```
PersonIdINTREFERENCES Person (PersonId),
    Email NVARCHAR(256)
)
-- Create Course table
CREATETABLE Course
(
CourseIdINTIDENTITY PRIMARY KEY,
NameNVARCHAR(50) NOTNULL,
    Teacher NVARCHAR(256) NOTNULL
)
-- Create Credit table
CREATETABLE Credit
(
StudentIdINTREFERENCES Student (StudentId),
CourseIdINTREFERENCES Course (CourseId),
    Grade DECIMAL(5,2) CHECK (Grade <= 100.00),
   Attempt TINYINT,
CONSTRAINT [UQ_studentgrades] UNIQUE CLUSTERED
    (
StudentId, CourseId, Grade, Attempt
    )
)
```



3. Expand the Tables node under yourDatabase in the Object Explorer to see the tables you created.



Load data into the tables

4. Create a folder called sampleData in your Downloads folder to store sample data for your database.
5. Right-click the following links and save them into the sampleData folder.
6. SampleCourseData
7. SamplePersonData
8. SampleStudentData
9. SampleCreditData
10. Open a command prompt window and navigate to the sampleData folder.
11. Execute the following commands to insert sample data into the tables replacing the values for server, database, user, and password with the values for your environment.

cmdCopy

```
bcp Course inSampleCourseData -S <server>.database.windows.net -d
<database> -U <user> -P <password> -q -c -t ","

bcp Person inSamplePersonData -S <server>.database.windows.net -d
<database> -U <user> -P <password> -q -c -t ","

bcp Student inSampleStudentData -S <server>.database.windows.net -d
<database> -U <user> -P <password> -q -c -t ","

bcp Credit inSampleCreditData -S <server>.database.windows.net -d
<database> -U <user> -P <password> -q -c -t ","
```

You have now loaded sample data into the tables you created earlier.

Query data

Execute the following queries to retrieve information from the database tables. See Write SQL queries to learn more about writing SQL queries. The first query joins all four tables to find the students taught by 'Dominick Pope' who have a grade higher than 75%. The second query joins all four tables and finds the courses in which 'Noe Coleman' has ever enrolled.

12. In a SQL Server Management Studio query window, execute the following query:

SQLCopy

```
-- Find the students taught by Dominick Pope who have a grade higher than 75%

SELECTperson.FirstName, person.LastName, course.Name, credit.Grade


FROM  PersonAS person

INNERJOIN Student AS student ONperson.PersonId = student.PersonId
```

```
INNERJOIN Credit AS credit ONstudent.StudentId = credit.StudentId
INNERJOIN Course AS course ONcredit.CourseId = course.courseId
WHEREcourse.Teacher = 'Dominick Pope'
AND Grade > 75
```

13. In a query window, execute the following query:

```
SQLCopy
-- Find all the courses in which Noe Coleman has ever enrolled
SELECTcourse.Name, course.Teacher, credit.Grade
FROM  CourseAS course
INNERJOIN Credit AS credit ONcredit.CourseId = course.CourseId
INNERJOIN Student AS student ONstudent.StudentId = credit.StudentId
INNERJOIN Person AS person ONperson.PersonId = student.PersonId
WHEREperson.FirstName = 'Noe'
ANDperson.LastName = 'Coleman'
```