

Microsoft MSSQL Test

Database Test

Version 1



Contents

Contents	2
Revision History	3
1. Install MSSQL Express.....	4
2. Download Database Sample	4
4. Questions.....	6
Question 1 - Basic	6
Question 3 - Advanced	7
Question 4 - Intermediate.....	8
Question 11 - Basic	9
Question 12 - Basic	9
Question 13 - Basic	9
Question 14 - Intermediate.....	10
Question 15 - Intermediate.....	10
Question 21 - Intermediate.....	10
Question 22 - Intermediate.....	11
Question 61 - Basic	11
Question 62 - Advanced	12
Question 63 - Advanced	12
Question 64 - Intermediate.....	13
Question 91 - Advanced	14
Question 92 - Advanced	15
Question 93 - Advanced	16
Question 96 - Advanced	17
Question 98 - Advanced	18
Question 100 - Advanced	19
Question 101 - Advanced	20

Revision History

Author	Date	Version	Notes
Simon Abi Saad	15/7/2023	1	Creation of initial version

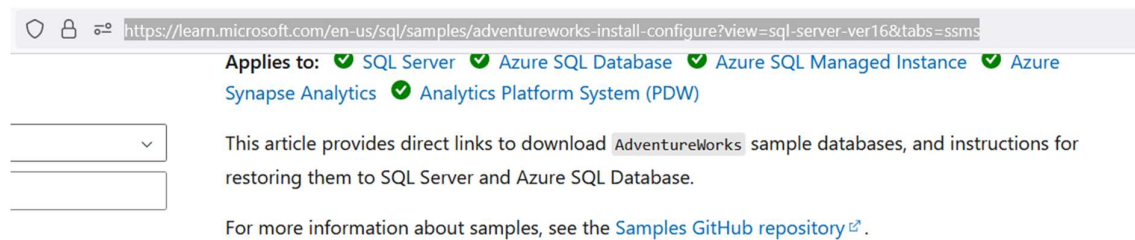
1. Install MSSQL Express

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Note: Please download and install Express version and install Microsoft SQL Server Management Studio (SSMS).

2. Download Database Sample

<https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms>



Prerequisites

- [SQL Server](#) or [Azure SQL Database](#)
- [SQL Server Management Studio \(SSMS\)](#) or [Azure Data Studio](#)

Download backup files

Use these links to download the appropriate sample database for your scenario.

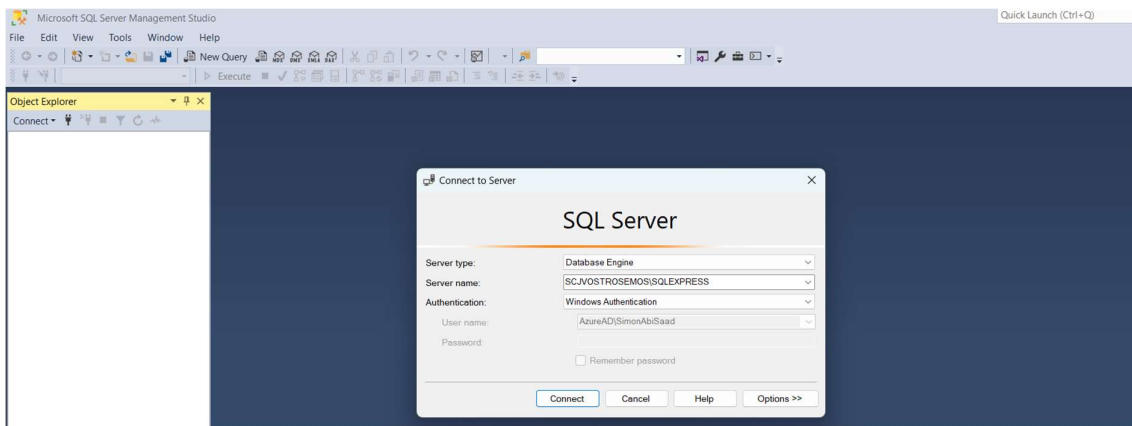
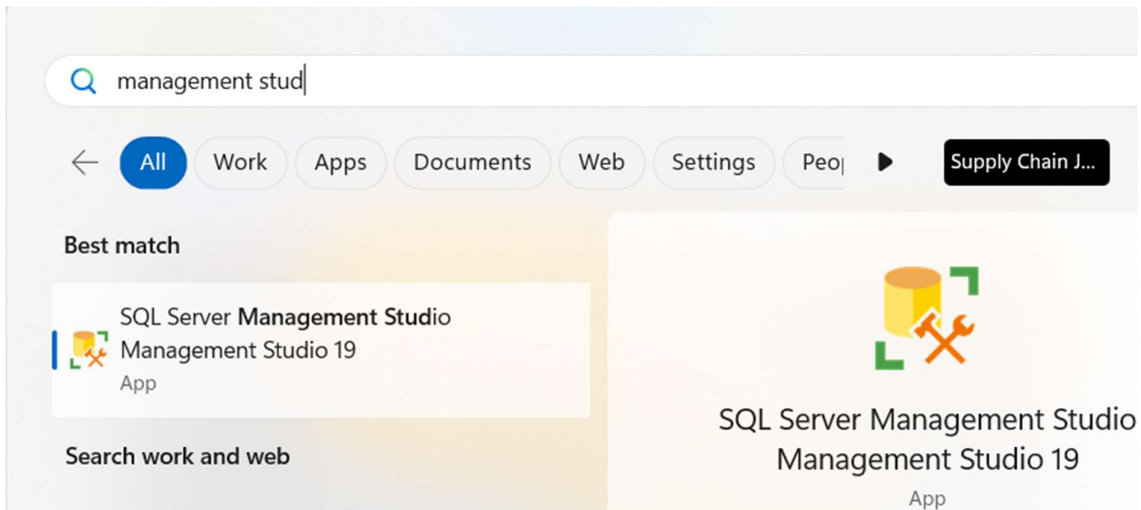
- **OLTP** data is for most typical online transaction processing workloads.
- **Data Warehouse (DW)** data is for data warehousing workloads.
- **Lightweight (LT)** data is a lightweight and pared down version of the OLTP sample.

If you're not sure what you need, start with the OLTP version that matches your SQL Server version.

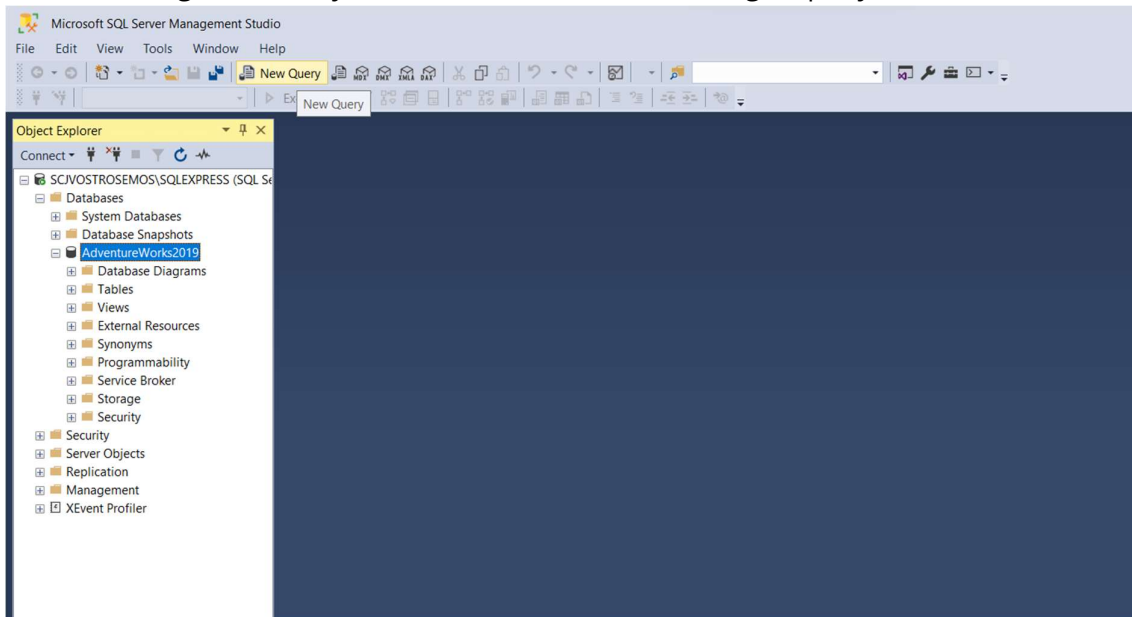
OLTP	Data Warehouse	Lightweight
AdventureWorks2022.bak	AdventureWorksDW2022.bak	AdventureWorksLT2022.bak
AdventureWorks2019.bak	AdventureWorksDW2019.bak	AdventureWorksLT2019.bak

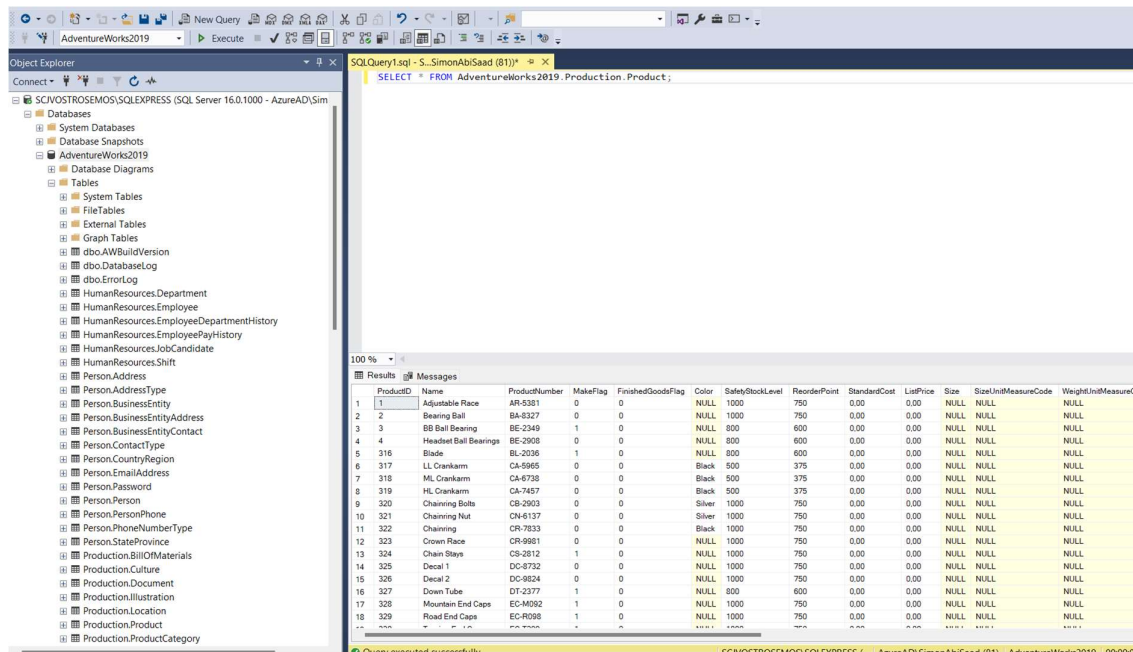
3. Restore Database

<https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms#restore-to-sql-server>



After restoring database, you can connect and test running a query





4. Questions

Question 1 - Basic

Write a SQL Statement that will give you a count of each object type in the AdventureWorks database. Order by count descending.

Hint:

Use the sys.objects table

Question 2 - Intermediate

a. Write a SQL Statement that will show a count of schemas, tables, and columns (do not include views) in the AdventureWorks database.

b. Write a similar statement as part a but list each schema, table, and column (do not include views). This table can be used later in the course.

Hint:

Either use sys.tables to join sys.columns and sys.schemas.

Or use information_schema.columns, but you will need to remove the view tables (Use information_schema.views).

Question 3 - Advanced

We learned in question 1 that 89 check constraints exist in the AdventureWorks Database. In this question we are going to determine what the check constraints are doing while creating a new database and table.

a. Create a new database called "Edited_AdventureWorks" (we are creating another database so we don't overwrite or change the AdventureWorks database). Then write a USE statement to connect to the new database.

b. Using the following tables - sys.check_constraints, sys.tables, and sys.columns to write a query that will give you TableName, ColumnName, CheckConstraintName, and CheckConstraintDefinition

c. Create a table named "tbl_CheckConstraint" in the "Edited_AdventureWorks" database with the following columns and data types:

TableName varchar(100)

ColumnName varchar(100)

CheckConstraint varchar(250)

Definition varchar(500)

ConstraintLevel varchar(100)

d. Using the query in part b insert the data into "tbl_CheckConstraint"

e. Using a case statement write an update statement (update ConstraintLevel) that will specify whether the constraint is assigned to the column or the table.

f. Once you're done interpreting the results drop the tbl_CheckConstraint table

Hint:

a. Because we are now connected to Edited_AdventureWorks we must fully qualify AdventureWorks2019 in our From

i.e., Select * From AdventureWorks2019.sys.tables

b. -- tables.object_id = check_constraints.parent_object_id

-- Use a left join to join sys.columns to sys.check_constraints. (we want all 89 constraints/rows)

-- sys.columns and sys.check_constraints have to be joined on two columns:

columns.column_id = check_constraints.parent_column_id

columns.object_id = check_constraints.parent_object_id

e. If the ColumnName is null in our created table then the constraint is assigned to the table not the column.

Question 4 - Intermediate

This is a difficult question; however, the results will be used in majority of the questions moving forward. There is plenty of background to get you started; however, don't become overwhelmed by it. If you're struggling then move on to the Hint and the video explanation.

We learned in Question 1 that there are 71 tables in the AdventureWorks Database. We can also see these tables in our entity relationship diagram (ERD). These tables are connected via primary keys and foreign keys. For example, in the Sales.SalesOrderHeader table there is a foreign key on the CurrencyRateID. This Foreign key is connected to the primary key in the Sales.CurrencyRate table. Therefore, when we connect these two tables together we will use the CurrencyRateID from both tables. The name of this Foreign Key in the AdventureWorks database is "FK_SalesOrderHeader_CurrencyRate_CurrencyRateID". We also know which Schema, Table, and Column join to the referenced Schema, referenced Table, and referenced column. (See Below)

ForeignKeyName: FK_SalesOrderHeader_CurrencyRate_CurrencyRateID

SchemaName: Sales

TableName: SalesOrderHeader

ColumnName: CurrencyRateID

ReferencedSchema: Sales

ReferencedTable: CurrencyRate

ReferencedColumn: CurrencyRateID

In this question you will replicate the 7 columns above (ForeignKeyName, SchemaName, TableName, ColumnName, ReferencedSchema, ReferencedTable, Referenced Column) for every Foreign Key in the AdventureWorks database.

Hint:

Here are the Joins necessary to answer this question:

```
FROM sys.foreign_key_columns FKC
```

```
INNER JOIN sys.objects O ON O.object_id = FKC.constraint_object_id
```

```
INNER JOIN sys.tables T1 ON T1.object_id = FKC.parent_object_id
```

```
INNER JOIN sys.tables T2 ON T2.object_id = FKC.referenced_object_id
```



```
INNER JOIN sys.columns C1 ON C1.column_id = parent_column_id
        AND C1.object_id = T1.object_id
```

```
INNER JOIN sys.columns C2 ON C2.column_id = referenced_column_id
        AND C2.object_id = T2.object_id
```

```
INNER JOIN sys.schemas S1 ON T1.schema_id = S1.schema_id
```

```
INNER JOIN sys.schemas S2 ON T2.schema_id = S2.schema_id
```

Question 11 - Basic

- How many employees exist in the Database?
- How many of these employees are active employees?
- How many Job Titles equal the 'SP' Person type?
- How many of these employees are sales people?

Hint:

Use the HumanResources.Employee Table and the Person.Person Table

Question 12 - Basic

- What is the name of the CEO? Concatenate first and last name.
- When did this person start working for AdventureWorks
- Who reports to the CEO? Includes their names and title

Hint:

Join HumanResources.Employee to Person.Person

Question 13 - Basic

- What is the job title for John Evans
- What department does John Evans work in?

Hint

Use EmployeeDepartmentHistory to join Employee and Department

Question 14 - Intermediate

- a. Which Purchasing vendors have the highest credit rating?
- b. Using a case statement replace the 1 and 0 in Vendor.PreferredVendorStatus to "Preferred" vs "Not Preferred." How many vendors are considered Preferred?
- c. For Active Vendors only, do Preferred vendors have a High or lower average credit rating?
- d. How many vendors are active and Not Preferred?

Hint

- a. use the Vendor table
- c. Cast CreditRating as a decimal

Question 15 - Intermediate

Assume today is August 15, 2014.

- a. Calculate the age for every current employee. What is the age of the oldest employee?
- b. What is the average age by Organization level? Show answer with a single decimal
- c. Use the ceiling function to round up
- d. Use the floor function to round down

Hint:

use the DATEDIFF Function

Question 21 - Intermediate

AdventureWorks works with customers, employees and business partners all over the globe. The accounting department needs to be sure they are up-to-date on Country and State tax rates.

- a. Pull a list of every country and state in the database.
- b. Includes tax rates.
- c. There are 181 rows when looking at countries and states, but once you add tax rates the number of rows increases to 184. Why is this?
- d. Which location has the highest tax rate?

Hint:

- a. Start by using the StateProvince table
- b. Use a left join when joining SalesTaxRate to StateProvince
- c. Find the countries/states that have more than 1 tax rate

Question 22 - Intermediate

The Marketing Department has never ran ads in the United Kingdom and would like you pull a list of every individual customer (PersonType = IN) by country.

- a. How many individual (retail) customers exist in the person table?
- b. Show this breakdown by country
- c. What percent of total customers reside in each country. For Example, if there are 1000 total customers and 200 live in the United States then 20% of the customers live in the United States.

Hint

- b. Be sure the total retail customers found in part a doesn't change as you join tables.
- c. Multiple ways to do this. Try using an Inner Query

Question 61 - Basic

You've been asked my Brian Welcker, VP of Sales, to create a sales report for him. This scenario is true for questions 61, 62, 63, and 64.

Write a script that will show the following Columns

- BusinessEntityID
- Sales Person Name - Include Middle
- SalesTerritory Name
- SalesYTD from Sales.SalesPerson

Order by SalesYTD desc

Hint:

- Join Person.Person and SalesTerritory to SalesPerson
- Add Coalesce to Middle Name

Question 62 - Advanced

You've been asked my Brian Welcker, VP of Sales, to create a sales report for him. This scenario is true for questions 61, 62, 63, and 64.

Complete Question 61 before attempting this question.

In Question 61 we wrote the script below:

Select

```
sp.BusinessEntityID
,Concat(FirstName,COALESCE (' ' + MiddleName, ''),',',LastName) as FullName
,isnull(st.Name,'No Territory') as TerritoryName
,Format(sp.SalesYTD,'C0') as SalesYTD
```

From Sales.SalesPerson sp

```
Inner Join Person.Person p on p.BusinessEntityID = sp.BusinessEntityID
```

```
Left Join Sales.SalesTerritory st on st.TerritoryID = sp.TerritoryID
```

Order by sp.SalesYTD desc

Add three columns to this script.

1. Rank each Sales Person's SalesYTD to all the sales persons. The highest SalesYTD will be rank number 1
2. Rank each Sales Person's SalesYTD to the sales persons in their territory. The highest SalesYTD in the territory will be rank number 1
3. Create a Percentile for each sales person compared to all the sales people. The highest SalesYTD will be in the 100th percentile

Hint:

1. Rank() Over(Order by [order_by_clause])
2. Rank() Over(Partition by [partition_by_clause] Order by [order_by_clause])
3. Percent_Rank() Over(Order by [order_by_clause])

Question 63 - Advanced

You've been asked my Brian Welcker, VP of Sales, to create a sales report for him. This scenario is true for questions 61, 62, 63, and 64.

Complete Question 62 before attempting this question.

In Question 62 we wrote the script below:

Select

```
sp.BusinessEntityID
,Concat(FirstName,COALESCE (' ' + MiddleName, ''),',',LastName) as FullName
,isnull(st.Name,'No Territory') as TerritoryName
,Format(sp.SalesYTD,'C0') as SalesYTD
,RANK() Over(Order by sp.SalesYTD desc) as TotalRank
,RANK() Over(Partition by st.Name Order by sp.SalesYTD desc) as TerritoryRank
,Format(PERCENT_RANK() Over(Order by sp.SalesYTD asc),'P0') as TotalPercentRank
```

From Sales.SalesPerson sp

```
Inner Join Person.Person p on p.BusinessEntityID = sp.BusinessEntityID
```

```
Left Join Sales.SalesTerritory st on st.TerritoryID = sp.TerritoryID
```

Order by sp.SalesYTD desc

Add Four columns to this script.

1. Using the Lag function show the value one rank below. The highest SalesYTD will show the value of the second rank.
2. Using the Lag function show the BusinessEntityID one rank below. The highest SalesYTD will show the BusinessEntityID of the second rank
3. Using the Lead function show the value one rank above. The highest SalesYTD will show a null value (no rank is higher)
4. Using the Lead function show the value 1 rank above. The highest SalesYTD will show a null value (no rank is higher)

Hint:

1 & 2. Lag(column,[offset],[default]) Over(Order by [order_by_clause])

3 & 4. Lead(column,[offset],[default]) Over(Order by [order_by_clause])

Question 64 - Intermediate

You've been asked my Brian Welcker, VP of Sales, to create a sales report for him. This scenario is true for questions 61, 62, 63, and 64.

Complete Question 63 before attempting this question.

In Question 63 we wrote the script below:

Select

```
sp.BusinessEntityID
,Concat(FirstName,COALESCE (' ' + MiddleName, ''),',',LastName) as FullName
,isnull(st.Name,'No Territory') as TerritoryName
,Format(sp.SalesYTD,'C0') as SalesYTD
,RANK() Over(Order by sp.SalesYTD desc) as TotalRank
,RANK() Over(Partition by st.Name Order by sp.SalesYTD desc) as TerritoryRank
,Format(PERCENT_RANK() Over(Order by sp.SalesYTD asc),'P0') as TotalPercentRank
,Format(Lag(sp.SalesYTD,1,null) Over(Order by sp.SalesYTD),'C0') as LagValue
,Format(Lag(sp.BusinessEntityID,1,null) Over(Order by sp.SalesYTD),'N0') as LagID
,Format(Lead(sp.SalesYTD,1,null) Over(Order by sp.SalesYTD),'C0') as LeadValue
,Format(Lead(sp.BusinessEntityID,1,null) Over(Order by sp.SalesYTD),'N0') as LeadID
```

From Sales.SalesPerson sp

```
Inner Join Person.Person p on p.BusinessEntityID = sp.BusinessEntityID
```

```
Left Join Sales.SalesTerritory st on st.TerritoryID = sp.TerritoryID
```

Order by sp.SalesYTD desc

Rather than sending Brian a flat file you are going to put this script into a view called Sales.vSalesPersonSalesYTD. Be sure to delete the view when you are done with this question.

Hint:

Create View

As [Insert View Script]

Question 91 - Advanced

There are 290 employees that work for AdventureWorks. Leadership would like to ensure that wish their employees Happy Birthday on their actual Birthday.

- Using a Cross Join (Do not use any other join for practice purposes), write a statement that will give a list of all 290 employees FullName, Birthdate, and Age.
- Add a where clause that will use the GetDate() Function to limit employees that have a Birthday today.

*Note: My Getdate() date will most likely be different than yours.

Hint:

- a. Use a Cross Join and where BusinessEntityID are the same. Also, Use DateDiff for Age
- b. Use Month() and Day() to find Today's Birthdates.

Question 92 - Advanced

Batching using a while loop is an effective way to improve performance when working with large datasets. Although in this question we will not be working with a large dataset we will be creating a while loop. We are going to create a table with two columns - SalesOrderHeader.Orderdate and TotalDue... In other words this will give you Daily Gross Revenue. We are going to insert this data into a table in increments of 100. Below you will see a Create Table and Insert Into Statement:

Create Table Sales.DailyRevenue

```
(OrderDate Date
,TotalDue money)
```

Insert Into Sales.DailyRevenue

```
(OrderDate
,TotalDue)
```

Select

```
OrderDate
,Sum(Totaldue) as TotalDue
```

From Sales.SalesOrderHeader

Group by OrderDate

Order by 1 desc

- a. This insert statement is inserting 1,124 rows of data into Sales.DailyRevenue. Put this statement into a while loop and only insert 100 rows at a time.
- b. Print a message in the while loop that will give an update on what OrderDates were inserted into the table. For Example, "Data Between May 2011 through September 2011 inserted"

Hint:

- a. Use Local variables to define relevant dates

While condition

Begin

Insert Into TableName

(ColumnName1, ColumnName2)

Select Statement

From TableName

Where condition

Set Redefine Local variables

End

b. Before Redefining Local Variables put your print message

Question 93 - Advanced

You've been asked to create a stored procedure that can be used for an online interface. This interface will allow customers to search for their Purchase History at AdventureWorks. When the customer inputs their phone number and account number into the online interface they will be able to see the products they purchased, the date it was ordered/purchased on, and the line total for each product.

a. Write a statement that includes the following elements:

- Person FullName (First and Last Name)
- Account Number
- Phone Number
- Order Date (SalesOrderHeader)
- Product Name
- Line Total

b. Put this statement into a stored procedure. Add two parameters to the stored procedure - Account Number and Phone Number. These parameters will require input variables in order to return the correct information.

c. Exec the Stored Procedure for the following Account Numbers and Phone numbers.

- Phone Number: 245-555-0173 Account Number '10-4020-000001'
- Phone Number: 417-555-0131 Account Number '10-4030-021762'
- Phone Number 620-555-0117 Account Number '10-4020-000695'

d. Drop the Stored Procedure

Hint:

b.

Create Procedure ProcedureName

@Parameter1 datatype

as (Select Statement)

c.

Exec ProcedureName

@Parameter1 = 'value'

d.

Drop Procedure ProcedureName

Question 96 - Advanced

If you run the query below you will find that 293 Products have undergone a price change.

Select

Count(Distinct ProductID) CNT

From Production.ProductListPriceHistory

In this question we are going to determine how many of these 293 products have undergone more than one price change. We are going to figure this out using 3 different methods.

*Note: Results for a,b,c will all be the same.

a. Using an Inner Query determine how many products in ProductListPriceHistory have experienced more than one price change.

b. Using a CTE determine how many products in ProductListPriceHistory have experienced more than one price change.

c. Using a Self Join (Without using a CTE or Inner Query) determine how many products in ProductListPriceHistory have experienced more than one price change.

Hint:

a.

Select

Columns

From (Inner Query) TableAlias

b.

With CTENAME as (

Query)

Select

Columns

From CTE

c.

Select Columns From Table1 a

Inner Join Table1 b on a.id = b.id

Question 98 - Advanced

You will notice that the Sales.SalesOrderDetail table has three indexes:

AK_SalesOrderDetail_rowguid

IX_SalesOrderDetail_ProductID

PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID

a. Drop the first index - AK_SalesOrderDetail_rowguid

b. Drop the second index - IX_SalesOrderDetail_ProductID

c. Drop the third index - PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID

d. Recreate the first index - AK_SalesOrderDetail_rowguid

Here are the details:

On the rowguid column

Unique

Non-Clustered

e. Recreate the second index - IX_SalesOrderDetail_ProductID

Here are the details:

On the ProductID column

Non-Unique

Non-Clustered

f. Recreate the third index - PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID

Here are the details:

On the SalesOrderID and SalesOrderDetailID columns

Clustered

Hint:

a&b.

Drop Index IndexName

on TableName

c. Drop the whole primary key:

Alter Table TableName

Drop Constraint ConstraintName

d&e.

Create Index IndexName

on TableName(ColumnName)

f. Recreate the Primary Key:

Alter Table TableName

Add Constraint ConstraintName

Primary Key (ColumnName)

Question 100 - Advanced

In Question 99 we practiced using a delete trigger. In this Question we are going to get familiar with the insert trigger.

a. Find the insert trigger on Production.WorkOrder. What is the name of this trigger?

b. What is this trigger doing?

c. Run the insert statement below:

```
INSERT INTO Production.WorkOrder
(ProductID,OrderQty,ScrappedQty
```

```
,StartDate,EndDate,DueDate
,ScrapReasonID,ModifiedDate)
VALUES
(680,1,0,'2014-06-03'
,'2014-06-13','2014-06-14'
,Null,'2014-06-15')
```

d. We have now inserted a row of data into the WorkOrder table, which also fired/triggered the trigger on Production.WorkOrder. Based on the trigger definition the newly insert data was also inserted into another table. Find the data that was insert into the other table.

e. Delete the rows of data insert into both tables.

**If you accidentally delete more rows than necessary then drop the AdventureWorks database and restore it again.

Hint:

a. Find the insert trigger via object explorer

b. Once you find the trigger in object explorer then right click on the table "Script table as" > "Create to" > "New Query Editor Window"

d. Find the table in the trigger script that also has data being insert into.

e.

Delete From TableName

Where ColumnName = condition

Question 101 - Advanced

In Question 99 we practiced using a delete trigger. And in Question 100 we practiced understanding an insert trigger. In this Question we are going to practice using an update trigger. However, in this question you will create a trigger that will update another column when fired.

a. Run this statement:

Select *

From HumanResources.Employee

Where CurrentFlag = '0'

You will notice that every employee in the database is considered a current employee. We are going to create a Trigger for a that will update a departure date when the employee current flag is updated to "1".

- b. Write a statement that will add a column to the HumanResources.Employee table. Call the column "DepartureDate." Make it a date datatype.
- c. Create a Trigger called HumanResources.uEmployee on the HumanResources.Employee table. Have the trigger set the DepartureDate = GetDate() when the current flag is updated to "0".
- d. Now set the current flag = 0 for BusinessEntityID 2. And check the DepartureDate in HumanResources.Employee
- e. Write a statement that will drop the trigger if it exists.
- f. Add another step in the trigger that will set the DepartureDate to NULL if the current flag is updated back to "1"
- g. Drop the trigger
- h. Delete the DepartureDate column

Hint:

b.

Alter Table TableName

Add ColumnName datatype

c.

Create Trigger TriggerName

On TableName

After Update As

Begin

Set NoCount ON --Optional

Update TableName

Set ColumnName = Update condition

From TableName

End

e.

Drop Trigger If Exists TriggerName

f. See hint c

g. Drop Trigger TriggerName

h.

Alter Table TableName

Drop Column ColumnName