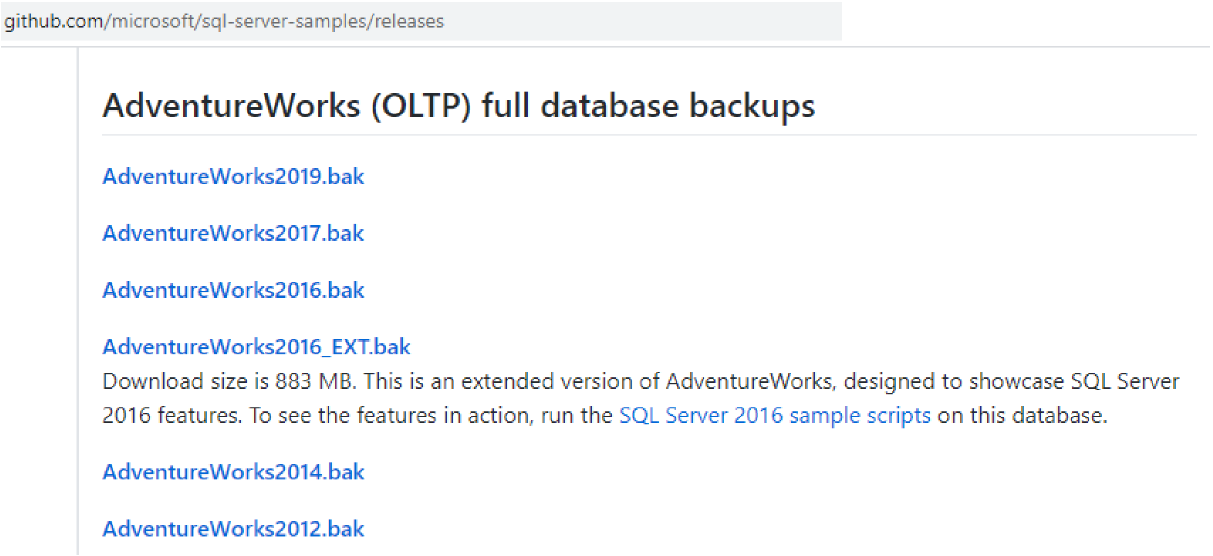
SE 308 Advanced Topics in Database Systems

TERM PROJECT 2

This is the second part of your term project. In this project, you are required to optimize some SQL queries.

PLEASE READ THE REST OF THE DOCUMENT CAREFULLY!

1. First of all, you will need to download **2012 version of AdventureWorks full database** from GitHub page of Microsoft (https://github.com/microsoft/sql-server-samples/releases). Please be careful not to download the LT version! If you cannot use 2012 version with the SQL Server version installed on your computer, you can try using a newer and compatible version. The files on this page are BACKUP files, therefore, you will need to RESTORE the database from the backup file.



github.com/microsoft/sql-server-samples/releases

AdventureWorks (OLTP) full database backups

AdventureWorks2019.bak

AdventureWorks2017.bak

AdventureWorks2016.bak

AdventureWorks2016_EXT.bak
Download size is 883 MB. This is an extended version of AdventureWorks, designed to showcase SQL Server 2016 features. To see the features in action, run the SQL Server 2016 sample scripts on this database.

AdventureWorks2014.bak

AdventureWorks2012.bak

2. Once you restore the database from the backup, you will make performance improvements on some SQL queries. You will measure the time it takes to complete a query before and after your optimizations (in seconds or milliseconds). In order to make the measurements, you have to run **each query at least 100 times before and after your optimizations**. You will provide all your measurements in your project report in very much detail. You can make incremental improvements for a query; therefore, you can have several measurements and comparisons like before-after-after-etc. I would like to see every incremental improvement in your report.
3. In order to ignore the effects of query caching and some other mechanisms of SQL Server, you have to execute "**DBCC FREEPROCCACHE**" and "**DBCC DROPCLEANBUFFERS**" before each and every run. If you do not do this, your measurements will be meaningless.
4. Beside your measurements, you need to explain why you do the specified improvement and how-and-why it makes the query faster than before. If you do not explain your improvement, or, if your explanation does not make sense, then you will not get any point for that part of your project. You cannot simply say "I tried randomly and it worked!".
5. Another important thing you have to be careful about is that you cannot change the SQL Query in any way. The queries have to stay exactly the same. If, however, you have a better idea to get

exactly the same result by altering a query, please put it into your report and explain your idea. You may get extra points for your extra work and improvement.

6. Furthermore, you cannot change database structure in any possible way. You cannot drop any existing index, but you can create your own indexes. Besides, you cannot add/update/delete record to/from the tables.
7. **For each query**, you will put the following information **into your report**:
    - Number of returned rows.
    - Top 10 records from the result set.
    - 100 before/after time measurements in a table. This table should have summary lines of total and average at the bottom.
    - Explanation of your optimization with every detail you find useful to put into the report.
8. It is very important that you do your experiments carefully and that you write a detailed report in a clear and organized way. Half of the points you may get will be from the presentation quality of your report. Therefore, please pay attention to your report very much. You need to use standard A4 paper size and 11 pt Calibri font for your report.
9. You can do this project individually or as a group of a maximum of four (4) students.
10. You will submit your project report file to the submission point on Blackboard with the following naming format:
    - "SE 308 Term Project 2 - Your First Name Your Last Name - Your Student ID.**pdf**"
    - If you are doing group work, then put only one name into the file name.
11. **Deadline**: **Sunday 18 May 2025 by 23:59.**

QUERY 1/3

```
-- 1. Select Order Date, Shipment Address State Name, Shipment Address City Name,
--     Total Order Quantity, Total Order Line Total
--     from Online orders between 1 Jan 2013 and 31 Dec 2013
SELECT SOH.OrderDate,
       PROV.Name AS StateProvinceName,
       ADDR.City,
       SUM(SOD.OrderQty) AS TotalOrderQty,
       SUM(SOD.LineTotal) AS TotalLineTotal
  FROM Sales.SalesOrderDetail SOD
 INNER JOIN Sales.SalesOrderHeader SOH
    ON SOH.SalesOrderID = SOD.SalesOrderID
 INNER JOIN Person.Address ADDR
    ON ADDR.AddressID = SOH.ShipToAddressID
 INNER JOIN Person.StateProvince PROV
    ON PROV.StateProvinceID = ADDR.StateProvinceID
 WHERE SOH.OrderDate BETWEEN '20130101' AND '20131231'
   AND SOH.OnlineOrderFlag = 1
 GROUP BY SOH.OrderDate, PROV.Name, ADDR.City
 ORDER BY SOH.OrderDate, PROV.Name, ADDR.City
```

QUERY 2/3

```
-- 2. Select Order Date, Product Category Name,
--     Total Order Quantity, Total Order Line Total
--     from Online orders between 1 Jan 2013 and 31 Dec 2013
--     of the products with MakeFlag = 1 or FinishedGoodsFlag = 1,
--     and color Black or Yellow.
SELECT SOH.OrderDate,
       CAT.Name as CategoryName,
       SUM(SOD.OrderQty) AS TotalOrderQty,
       SUM(SOD.LineTotal) AS TotalLineTotal
  FROM Sales.SalesOrderDetail SOD
 INNER JOIN Sales.SalesOrderHeader SOH
    ON SOH.SalesOrderID = SOD.SalesOrderID
 INNER JOIN Production.Product P
    ON P.ProductID = SOD.ProductID
 INNER JOIN Production.ProductSubcategory SUBCAT
    ON SUBCAT.ProductCategoryID = P.ProductSubcategoryID
 INNER JOIN Production.ProductCategory CAT
    ON CAT.ProductCategoryID = SUBCAT.ProductSubcategoryID
 WHERE SOH.OrderDate BETWEEN '20130101' AND '20131231'
   AND SOH.OnlineOrderFlag = 1
   AND (P.MakeFlag = 1 OR P.FinishedGoodsFlag = 1)
   AND P.Color IN ('Black', 'Yellow')
 GROUP BY SOH.OrderDate, CAT.Name
 ORDER BY SOH.OrderDate, CAT.Name
```

QUERY 3/3

```
-- 3. Select Store Name, Product Category Name,
--    Total Order Quantity, Total Order Line Total
--    from orders (not online but from physical stores)
--    between 1 Jan 2013 and 31 Dec 2013
--    of the products with MakeFlag = 1 or FinishedGoodsFlag = 1,
--    and color Black or Yellow.
SELECT STOR.Name as StoreName,
       CAT.Name as CategoryName,
       SUM(SOD.OrderQty) AS TotalOrderQty,
       SUM(SOD.LineTotal) AS TotalLineTotal
  FROM Sales.SalesOrderDetail SOD
 INNER JOIN Sales.SalesOrderHeader SOH
    ON SOH.SalesOrderID = SOD.SalesOrderID
 INNER JOIN Production.Product P
    ON P.ProductID = SOD.ProductID
 INNER JOIN Production.ProductSubcategory SUBCAT
    ON SUBCAT.ProductCategoryID = P.ProductSubcategoryID
 INNER JOIN Production.ProductCategory CAT
    ON CAT.ProductCategoryID = SUBCAT.ProductSubcategoryID
 INNER JOIN Sales.Customer CUST
    ON CUST.CustomerID = SOH.CustomerID
 INNER JOIN Sales.Store STOR
    ON STOR.BusinessEntityID = CUST.StoreID
 WHERE SOH.OrderDate BETWEEN '20130101' AND '20131231'
   AND SOH.OnlineOrderFlag = 0
   AND (P.MakeFlag = 1 OR P.FinishedGoodsFlag = 1)
   AND P.Color IN ('Black', 'Yellow')
 GROUP BY STOR.Name, CAT.Name
 ORDER BY STOR.Name, CAT.Name
```