Problem Set B Submission Form

# Overview

|  |  |
| --- | --- |
| Your Name | Jaya Varshini Prabakar |
| Your SU Email | jprabaka@syr.edu |

# Instructions

Put your name and SU email at the top. Answer these questions all from the lab. When asked to include screenshots, please follow the screen shot guidelines from the first homework.

Remember as you complete the homework it is not only about getting it right / correct. We will discuss the answers in class so it’s important to articulate anything you would like to contribute to the discussion in your answer:

* If you feel the question is vague, include any assumptions you've made.
* If you feel the answer requires interpretation or justification provide it.
* If you do not know the answer to the question, articulate what you tried and how you are stuck.
* Highlight any doubts or questions you would like me to review.

This how you receive credit for answering questions which might not be correct. In addition, you must complete the reflection portion of the homework assignment for full credit. Since most answers will be similar this is an important part of your individual submission.

Complete Part II of this document first, then go back and complete the Reflection in Part I.

# Part I - Reflection

Use this section to reflect on your learning. To achieve the highest grade on the assignment you must be as descriptive and personal as possible with your reflection.

1. As you completed this assignment, identify what have you learned?  
   From this assignment concepts like creating and significance of Clustered and UnClustered Indexing are clear. The assignment also helped in understanding the concepts of Window Functions, Common table expressions etc..,
2. What barriers or challenges did you encounter while completing this assignment?  
   Indexing was bit tough to catch in the 1st go, but the Video lecture helped me better understand the concepts.
3. How prepared were you to complete this assignment? What can you do to be better prepared?  
   I was well prepared
4. Rate your comfort level with this week’s material. Use the rubric provided. Highlight the number representing your ability.

**4**

4 ==> I understand this material and can explain it to others.  
3 ==> I understand this material.  
2 ==> I somewhat understand the material but sometimes need guidance from others.  
1 ==> I understand very little of this material and need extra help.

# Part II – Questions

Paste your answers to the Exercises found in the lab document. Make sure to include your netid in any screenshots you provide (no screenshots needed for this first lab). If the question asks for commands, only include those commands which are necessary to complete the tasks. Number each answer.

1. List the Product ID, Category ID, product name and product unit price for products that are not discontinued.

Code:

Select ProductID, CategoryID, ProductName, UnitPrice from northwind.dbo.Products

where Discontinued = 0

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Join the categories table so that you display the category name in the first query.

Code: Select ProductID, northwind.dbo.Products.CategoryID, CategoryName ,ProductName, UnitPrice from northwind.dbo.Products join northwind.dbo.Categories

 on northwind.dbo.Products.CategoryID = northwind.dbo.Categories.CategoryID

where Discontinued = 0

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Produce a query displaying the category ID, category name, and average product unit price.

Code: SELECT p.ProductName,c.CategoryID, c.CategoryName, AVG(p.UnitPrice) as Average\_Unit\_Price from Categories c join Products p on c.CategoryID = p.CategoryID

group by c.CategoryID,c.CategoryName

A screenshot of a computer

Description automatically generated

1. Next, combine queries 2 and 3 with a join. Join on category ID so you can display product id, category id, product name, product unit price, and average product unit price, then subtract the unit price from the average.

HINT: Use a common table expression WITH clause to create two named queries then use them as tables to join for the last query.

Code: With table1 as(

    Select ProductID, p.CategoryID as CategoryID, CategoryName ,ProductName, UnitPrice from Products as p join Categories as c on p.CategoryID = c.CategoryID

where Discontinued = 0

),

table2 as(

   SELECT c.CategoryID, c.CategoryName, AVG(p.UnitPrice) as Average\_Unit\_Price from Categories c join Products p on c.CategoryID = p.CategoryID

group by c.CategoryID,c.CategoryName

)

SELECT t1.ProductID,t1.CategoryID,t1.ProductName,t1.UnitPrice, t2.Average\_Unit\_Price,( t2.Average\_Unit\_Price- t1.UnitPrice) as diff from table1 as t1  join table2 as t2 on t1.CategoryID = t2.CategoryID

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Re-write the query in 4 to use a window function instead of 3 queries.  
   Code: select ProductID, CategoryID,ProductName, UnitPrice,

avg(UnitPrice) over (partition by CategoryID), avg(UnitPrice) over (partition by CategoryID) - UnitPrice from Products

1. Compare the query plans of 4 and 5. Are they the same?  
   No, on comparing the query plans of 4 and 5 are different.
2. What is a good index candidate for query 4 or 5? How would it improve the performance of the query?

A screenshot of a computer

Description automatically generated  
 For query 5 and 4 the good index candidate Is CategoryID, since we are partitioning by CategoryID and we can see the sorting is also by CategoryID. This makes it an ideal candidate for Index.

After creating an Index for CategoryID, The plan of query 5 looks like this:

A computer screen shot of a computer

Description automatically generated

This eliminates sort and also introduces Index Scan. Removing the sort makes the query less time consuming since we are not spending time sorting and this improves the query performance.

After creating the Index for CategoryID, the query 4 had Index Seek which also had improved the performance since seek is more faster than a scan.