Homework IST769 Unit A - Introduction

## Agenda

1. Your Questions
2. Go Over Homework

FRONT MATTER:

Start your Lab!

Azure Labs invite: <https://labs.azure.com/register/0zfx6kze>

Code Share: <https://codeshare.io/eVXbOJ>

## 1. Your Questions

* What is Indexing in sql specifically and how does it work?
  + Good overview of indexing  
    <https://medium.com/geekculture/indexing-in-postgres-db-4cf502ce1b4e>
  + I’ll do examples in class
* Can we go for more examples to understand the query plans?
* How can we evaluate the execution plan?
* When I use ‘JOIN’, Index search/seek is used, but when I use ‘LEFT JOIN’ It uses cluster index search/seek instead, can you explain why?

## 2. Homework

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

-- Q1

SELECT

ProductID,

CategoryID,

ProductName,

UnitPrice

FROM

Products

WHERE

Discontinued = 0

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

SELECT

p.ProductID, p.CategoryID, c.CategoryName, p.ProductName, p.UnitPrice

FROM Products p

JOIN Categories c ON p.CategoryID = c.CategoryID

WHERE p.Discontinued = '0';

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

* Question: Do we include discontinued products when calculating average product unit price?

SELECT Categories.CategoryID

,Categories.CategoryName

,AVG(Products.UnitPrice) as AverageUnitPrice

FROM Products

LEFT JOIN Categories ON Products.CategoryID = Categories.CategoryID

GROUP BY Categories.CategoryID, Categories.CategoryName

4. 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.

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

WITH table1 AS (

SELECT P.ProductID, P.CategoryID, C.CategoryName, P.ProductName, P.UnitPrice

FROM Products P

JOIN Categories C ON P.CategoryID = C.CategoryID

WHERE P.Discontinued = 0

),

table2 AS (

SELECT C.CategoryID, C.CategoryName, AVG(P.UnitPrice) AS Avg\_Product\_Unit\_Price

FROM Products P

JOIN Categories C ON P.CategoryID = C.CategoryID

WHERE P.Discontinued = 0

GROUP BY C.CategoryID, CategoryName

)

SELECT t1.ProductID, t1.CategoryID, t1.ProductName, t1.UnitPrice AS Product\_Unit\_Price,

t2.Avg\_Product\_Unit\_Price,

t1.UnitPrice - t2.Avg\_Product\_Unit\_Price AS Diff\_Betw\_Unit\_Price\_and\_Avg\_Product\_Unit\_Price

FROM table1 t1

JOIN table2 t2 ON t1.CategoryID = t2.CategoryID;

;With T1 as (

SELECT P.ProductID, P.CategoryID, C.CategoryName

, P.ProductName, P.UnitPrice

FROM Products P

JOIN Categories C ON P.CategoryID = C.CategoryID

WHERE P.Discontinued = 0

), T2 as (

SELECT CategoryID, CategoryName, AVG(UnitPrice) AS AverageUnitPrice

FROM T1

GROUP BY CategoryID, CategoryName

)

SELECT P.ProductID, P.CategoryID, P.CategoryName, P.ProductName, P.UnitPrice

, COALESCE(C.AverageUnitPrice,0) AS AverageUnitPrice

, P.UnitPrice - COALESCE(C.AverageUnitPrice,0) AS PriceDifference

FROM T1 P

JOIN T2 C ON P.CategoryID = C.CategoryID;

GO

5. Re-write the query in 4 to use a window function instead of 3 queries.

SELECT P.ProductID, P.CategoryID, C.CategoryName, P.ProductName, P.UnitPrice,

AVG(P.UnitPrice) OVER (PARTITION BY P.CategoryID) AS AverageUnitPrice,

P.UnitPrice - AVG(P.UnitPrice) OVER (PARTITION BY P.CategoryID) AS PriceDifference

FROM Products P

JOIN Categories C ON P.CategoryID = C.CategoryID

WHERE P.Discontinued = 0;

6. Compare the query plans of 4 and 5. Are they the same?

7. What is a good index candidate for query 4 or 5? How would it improve the performance of the query?