# IST769 Homework Submission #4

## Basic Information

Your Name: Thulasi Ram Ruppa Krishnan   
Your SUID: 460746269  
Your Email: truppakr@syr.edu  
Date Due: 4-23-2020   
Homework #: HW4

## Instructions

For each answer, please include your answer as text, and any screenshot(s) which demonstrate your answer was executed. Most importantly, make sure to include evidence your answer is correct. This will most likely be a screenshot. If you had issues, problems, or had to make assumptions include them in your answer.

## Answers:

Complete each of the following exercises. If you are unsure how to accomplish the task, please consult the coursework videos where there are explanations and demos. Where appropriate use the live query statistics feature to verify your query uses the index as directed.

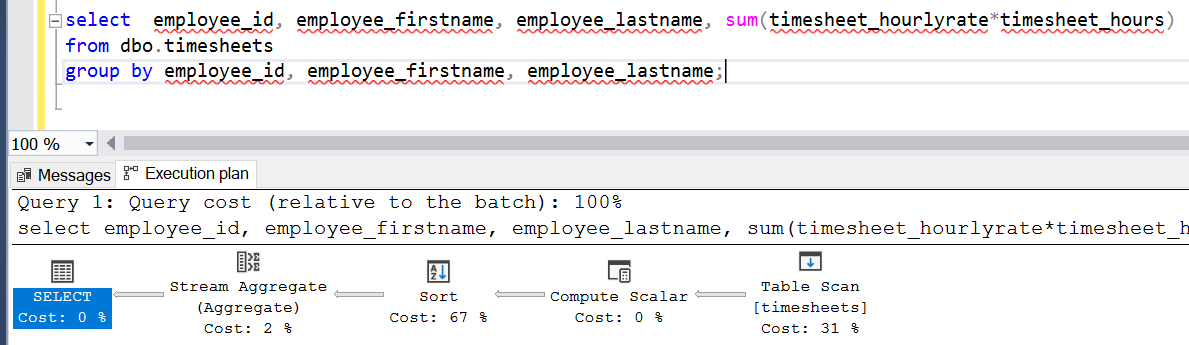
1. Create a non-clustered index on the **timesheets** table in the **demo** database. The index you create should be designed to improve the following query:

select employee\_id, employee\_firstname, employee\_lastname, sum(timesheet\_hourlyrate\*timesheet\_hours)

from timesheets

group by employee\_id, employee\_firstname, employee\_lastname;

Before



After

USE [demo]

GO

SET ANSI\_PADDING ON

GO

/\*\*\*\*\*\* Object: Index [IX\_employee] Script Date: 4/19/2020 4:34:42 PM \*\*\*\*\*\*/

CREATE NONCLUSTERED INDEX [IX\_employee] ON [dbo].[timesheets]

(

[employee\_id] ASC,

[employee\_lastname] ASC,

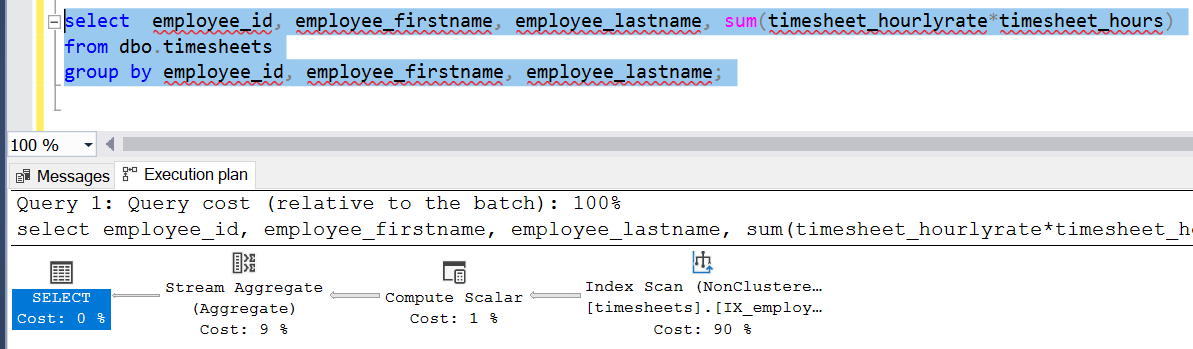
[employee\_firstname] ASC

)

INCLUDE([timesheet\_hourlyrate],[timesheet\_hours]) WITH (

PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

GO



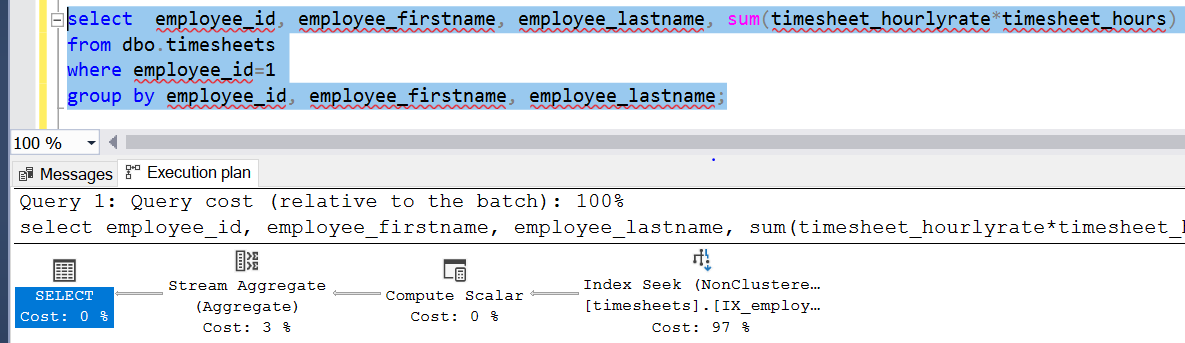
1. Write an SQL Select query which uses the index you created in the first question but does an index seek instead of an index scan.

select employee\_id, employee\_firstname, employee\_lastname, sum(timesheet\_hourlyrate\*timesheet\_hours)

from dbo.timesheets

where employee\_id=1

group by employee\_id, employee\_firstname, employee\_lastname;



1. Create a single columnstore index on the **timesheets** table in the **demo** database which will improve the following queries:

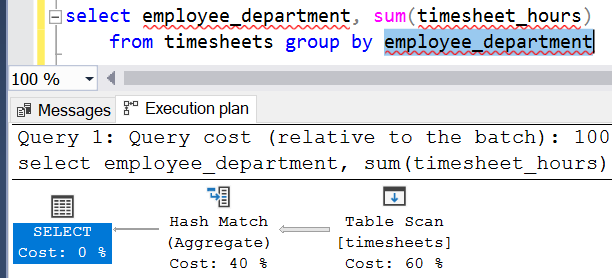
select employee\_department, sum(timesheet\_hours)

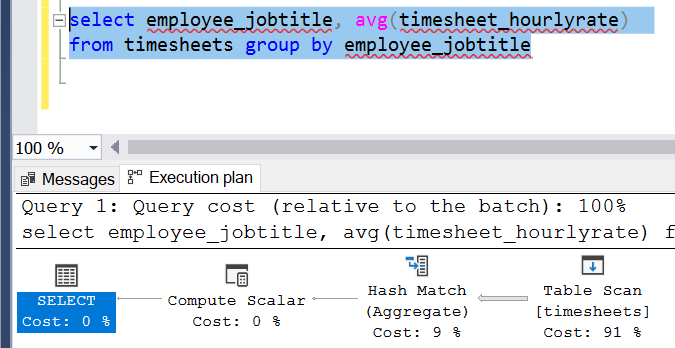
from timesheets group by employee\_department

select employee\_jobtitle, avg(timesheet\_hourlyrate)

from timesheets group by employee\_jobtitle

Before





After

USE [demo]

GO

/\*\*\*\*\*\* Object: Index [IX\_CS\_employee] Script Date: 4/19/2020 5:24:50 PM \*\*\*\*\*\*/

CREATE NONCLUSTERED COLUMNSTORE INDEX [IX\_CS\_employee] ON [dbo].[timesheets]

(

[employee\_jobtitle],

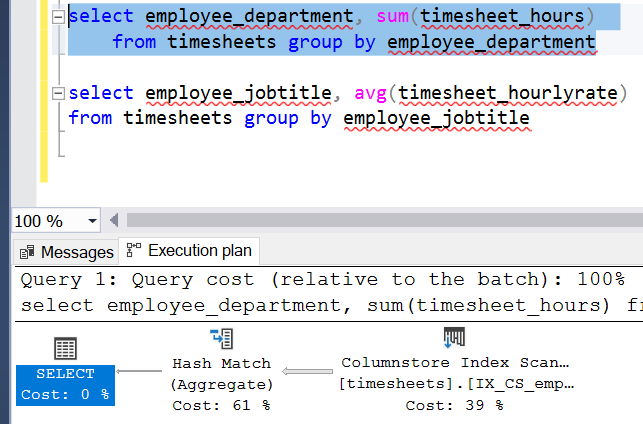
[employee\_department],

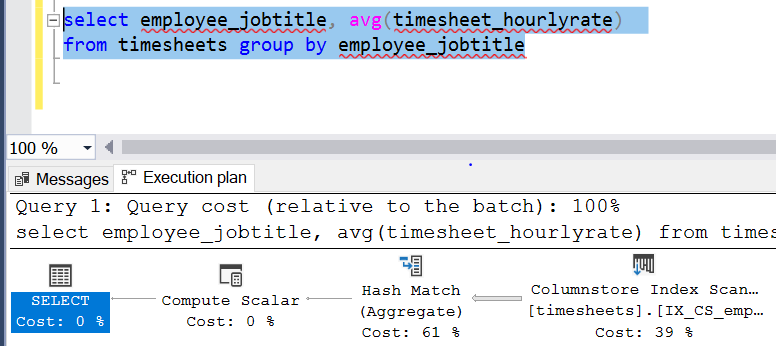
[timesheet\_hourlyrate],

[timesheet\_hours]

)WITH (DROP\_EXISTING = OFF, COMPRESSION\_DELAY = 0) ON [PRIMARY]

GO





1. Create an indexed view named **v\_employees** on the **timesheets** table in the **demo** database which lists the employee id, first name, last name, job title, and department columns values and one row per employee (essentially re-building the employee table). Then set a unique clustered index on the view and finish by writing an SQL Select query which uses the indexed view.

DROP VIEW IF EXISTS dbo.v\_employees

GO

CREATE VIEW dbo.v\_employees

WITH SCHEMABINDING

AS

SELECT

[employee\_id]

,[employee\_firstname] AS first\_name

,[employee\_lastname] AS last\_name

,[employee\_jobtitle] AS jobtitle

,[employee\_department] AS department

,COUNT\_BIG(\*) timesheets\_count

FROM dbo.timesheets

GROUP BY [employee\_id],[employee\_firstname],[employee\_lastname],[employee\_jobtitle],[employee\_department]

GO

/\*\*\*\*\*\* Object: Index [IX\_v\_employees] Script Date: 4/19/2020 5:51:20 PM \*\*\*\*\*\*/

CREATE UNIQUE CLUSTERED INDEX [IX\_v\_employees] ON [dbo].[v\_employees]

(

[employee\_id] ASC,

[first\_name] ASC,

[last\_name] ASC,

[jobtitle] ASC,

[department] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

GO

SELECT \* FROM dbo.v\_employees

ORDER BY [employee\_id]

GO

SELECT \* FROM dbo.v\_employees

WHERE department='Electronics'

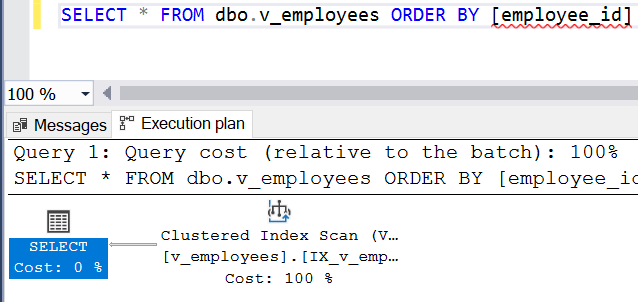
ORDER BY [employee\_id]

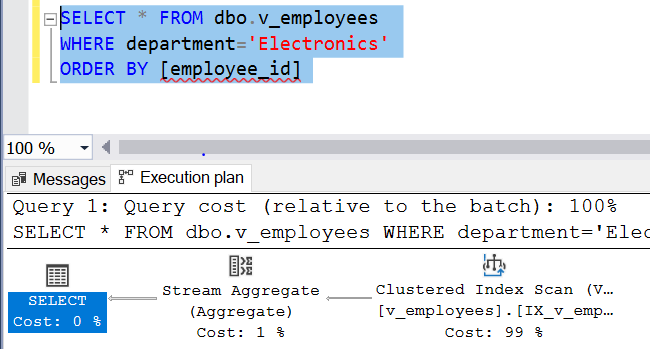
GO

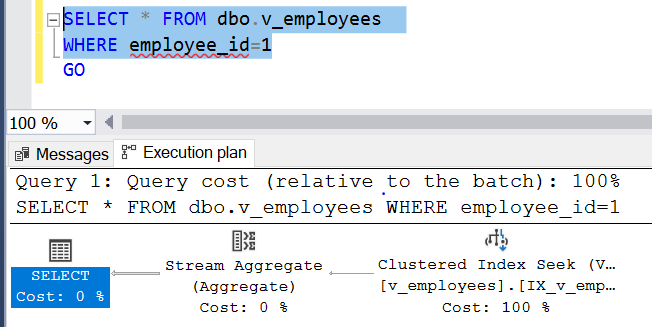
SELECT \* FROM dbo.v\_employees

WHERE employee\_id=1

GO







1. Output the following query in JSON format: Display the employee id, first name, last name, count of timesheets, total hours worked, and average timesheet hourly rate.

SELECT

employee\_id

, employee\_firstname AS first\_name

, employee\_lastname AS last\_name

, COUNT(\*) timesheets\_count

, SUM(timesheet\_hours) total\_hours\_worked

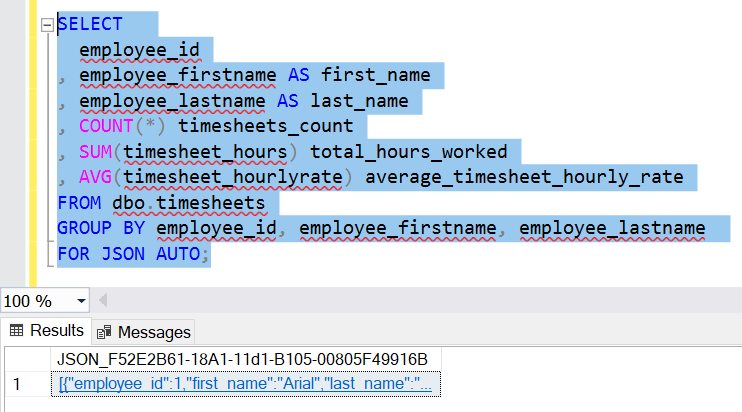
, AVG(timesheet\_hourlyrate) average\_timesheet\_hourly\_rate

FROM dbo.timesheets

GROUP BY employee\_id, employee\_firstname, employee\_lastname

FOR JSON AUTO

<result_json_form.txt>



# Tear-Down

When you are finished with the homework you should stop the environment:

1. From the terminal window where you typed docker-compose up -d type in the following:  
   docker-compose stop

