Lab Homework 4: Performance, Security, NoSQL

# Learning Objectives

In this exercise, you will practice programming in Microsoft SQL Server:

* Create and manage indexes
* Use the security language features to control access to database objects
* Use JSON features to query and create semi-structured data.

# Setup

To complete this homework or follow along with the instructor’s class demos from your coursework, you must first complete the steps in this setup section. For this exercise we require the SQL Server instance in the Docker environment. For more details on the how’s and why’s of managing the database environments used in this course, consult Lab 1. Instructions for this assignment:

1. Open the PowerShell prompt on Windows or the Terminal on MacOS.
2. Type:  
   cd adv-db-labs  
   to change the working directory to the repository folder. If you are in the correct spot, your command prompt should have adv-db-labs in it, for instance: PS adv-db-labs>
3. Change into the mssql folder, type:  
   cd mssql  
   if you are in the correct folder, your command prompt should have mssql in it.
4. Bring up the MSSQL environment, type:  
   docker-compose up -d
5. Check to make sure the environment is running, type:  
   docker-compose ps

The state of the **mssql** should be **Up** on **port 1433**.

1. You are now ready to connect to the running instance. Open **SQL Server Management Studio** (on Windows) or **SQL Server Operations Studio** (if you’re on a Mac), a.k.a the SQL Client. Use the SQL Client to connect to SQL server with:
   1. Server name: **localhost**
   2. Authentication: **SQL Server Authentication**
   3. Login: **sa**
   4. Password: **SU2Orange!**
2. After you’ve connected, open a new query window by pressing CTRL+n. Run the following script in the query window to create the required data set:  
   select \* into demo.dbo.timesheets  
   from fudgemart\_employees join fudgemart\_employee\_timesheets

on employee\_id = timesheet\_employee\_id;

1. You are ready to begin the lab!

# Exercises

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;

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.
2. 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

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.
2. 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.

# Turning it in

Take your copy and paste each of the solutions to the exercises into the submission template file included with this assignment. Make sure your name and SU email are at the top and turn in your work through the course learning management system.

# 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