

The smile factory case study report

by Richard Lamnea

FILE	11200_RICHARD_LAMNEA_THE_SMILE_FACTORY_CASE_STUDY_REPORT_31 5925_990486971.DOCX (570.92K)		
TIME SUBMITTED	22-JAN-2021 11:02PM (UTC+0000)	WORD COUNT	877
SUBMISSION ID	143138408	CHARACTER COUNT	4951

University of Wales Trinity St David

School of Applied Computing

The Smile Factory Case Study

Databases and Application Development

ACCA5026

Richard Lamnea

1802749

01/01/2021



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

ABSTRACT

This report outlines performance improvements to the “The Smile Factory” database queries and investigates their structure to provide a performance tuning solution to their database.

TABLE OF CONTENTS

I. INTRODUCTION	1
II. PERFORMANCE INVESTIGATION	1
III. METHODOLOGY	1
IV. CONCLUSION	2
V. REFERENCE LIST	2

TABLE OF FIGURES

Figure 1 - Start Date query investigation	1
Figure 2 - StartDate query investigation result.	1
Figure 3 - NHS Eligibility query investigation	2
Figure 4 - NHS Eligibility query index result	2
Figure 5 - NHS Eligibility query index improvement result	2

I. INTRODUCTION

A local dental clinic, named “The Smile Factory” has requested assistance to investigate their database performance issues and improve the query performance. This report discusses the investigation of the database transactions and provides a solution to the performance issues. It was determined adding two additional indexes would alleviate the performance issue.

II. PERFORMANCE INVESTIGATION

To investigate the database performance, commonly used queries need to be tested by their execution time and execution plan which highlights indexes which have been used and their structure. An index is an on-disk structure with a table that enhances the speed when retrieving rows from a table [1]. The index enables efficient values associated with the key index. Indexes have two types, Clustered and Non-clustered [2]. Clustered index sort and store the data rows within the table based on their key values. When a table has no clustered index its data rows are stored in a heap. Non-clustered indexes contain a structure which is separate from the data rows. This index contains an index key value and contains a pointer to each value entry of which contain a key value. The pointer value is called a row locator and can efficiently locate specific rows [2].

An investigation into the Date Table was conducted to improve the speed at which these values can be obtained and to improve performance when querying specific values contained within this table such as the StartDate of each week. The Patients table and values contained within were also investigated as these values are often used when booking an appointment for a patient.

III. METHODOLOGY

To investigate the performance issues of the database. The analysis of the database was undertaken using SQL Management Studio[3]. The investigation was undertaken upon the expected data queries

within the database, the first query inspected was the Date Table which can be seen in Fig 1. The query requests the database to return a specific start date of the week commencing from the Date table.

The SQL query can be seen below:

```
SET STATISTICS TIME ON
SET STATISTICS IO ON
GO

-- Test returning StartDate values
SELECT *
FROM TBL_DATE
WHERE TBL_DATE.StartDate = '2022-03-14';
```

The result was analysed which displayed an elapsed execution time for the query of 157ms which suggests inefficiencies within its index structure, and this can be seen in Fig 1.

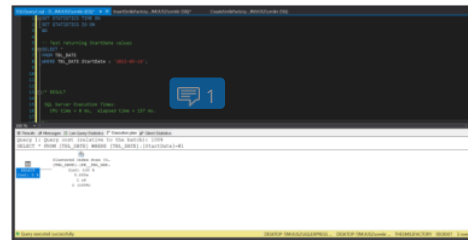


Figure 1 - Start Date query investigation

A Non clustered index was created for the Date Table on the StartDate column to improve the performance of the query using the command below: -

```
CREATE NONCLUSTERED INDEX IndexDateStartDate ON
TBL_DATE(StartDate);
```

This command returned an improved query execution time of 0ms which can be seen in Fig 2.

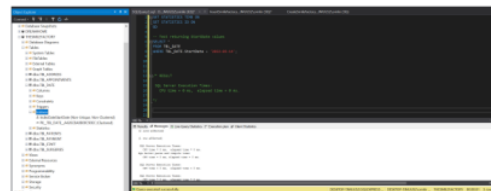


Figure 2 - StartDate query investigation result.

The second query identified to be commonly used within the database would be NHS eligibility within the Patients table. This attribute will be enquired upon with every time an appointment is booked so an investigation into its query performance was

conducted which can be seen in Fig 3. The result returned a 94ms query time which suggests performance inefficiencies.

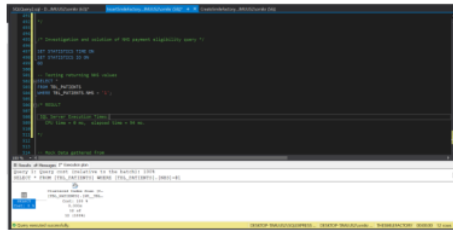


Figure 3 - NHS Eligibility query investigation

A Non-clustered index was created for the Patients table and NHS column to improve the performance of the query. This resulted in an 84ms query time which can be seen in Fig 4. This result improved performance slightly however when using the SELECT * command, SQL uses the primary key clustered index as default so efficiency from this query proved to still be inefficient in this case.

The query was adapted to return specific attributes such as FirstName, and LastName from the table which resulted with an improved execution time of 0ms. This suggests that specifying attributes when using a non-clustered index improves performance. This adaption to the SELECT field shows an improvement of the query performance which can be seen in Fig 5.

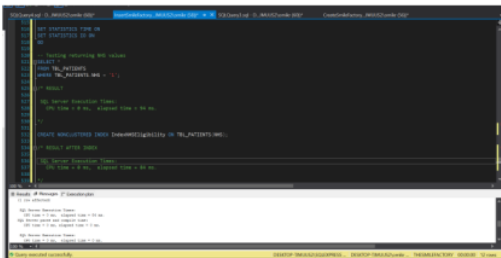


Figure 4 - NHS Eligibility query index result

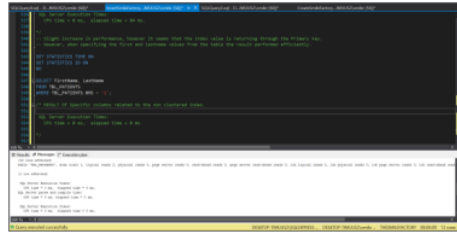


Figure 5 - NHS Eligibility query index improvement result

IV. CONCLUSION

In conclusion, the investigation into the query results yielded improved performance times using Non clustered indexes. To further improve the performance of the database additional secondary indexes could be created however using the non-clustered indexes implemented within this report then this should provide a significant performance enhancement to the database.

V. REFERENCE LIST

- [1] pmasl, 'CREATE INDEX (Transact-SQL) - SQL Server'. <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-index-transact-sql> (accessed Jan. 22, 2021).
- [2] MikeRayMSFT, 'Clustered and Nonclustered Indexes Described - SQL Server'. <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described> (accessed Jan. 22, 2021).
- [3] dzsquared, 'Download SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS)'. <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms> (accessed Jan. 22, 2021).

The smile factory case study report

GRADEMARK REPORT

FINAL GRADE

76/100

GENERAL COMMENTS

Instructor

A well written report with some excellent discussion on performance tuning. Maybe zoom in on the code screenshots as your figure were difficult to read. Perhaps place a copy of the schema and discuss your transaction examples within the report. Excellent knowledge of SQL and database testing demonstrated.

PAGE 1

PAGE 2



Comment 1

difficult to see maybe zoom in on relevant points

PAGE 3
