

Analiza wydajności relacyjnych baz danych Oracle oraz MSSQL na podstawie aplikacji desktopowej

Grzegorz Dziewit*, Jakub Korczyński*, Maria Skublewska-Paszkowska

^a Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Porównywanie wydajności relacyjnych baz danych nie jest trywialnym zjawiskiem ze względu na różnice implementacji różnych systemów bazodanowych. Niniejszy artykuł przedstawia metodykę sposobu porównania relacyjnych systemów bazodanowych pod względem średniego czasu wykonania poszczególnych zapytań bazodanowych typu DML (ang. Data Manipulation Language) zawierających podzapytania oraz złączenia tabel. Metodyka może być dodatkowo dostosowana do badań wydajnościowych w zakresie samej bazy danych (badanie zapytań wywoływanych bezpośrednio w silniku bazodanowym). Zastosowana metodyka pozwala na stwierdzenie, który system bazodanowy jest lepszy w porównaniu do innych w zależności od funkcjonalności spełnianych przez aplikację zewnętrzną. W artykule przeprowadzono analizę średnich czasów wykonania poszczególnych zapytań bazodanowych typu DML. Zostały postawione dwie hipotezy badawcze: „System bazodanowy Microsoft SQL Server charakteryzuje się krótszym czasem wykonania zapytań typu INSERT oraz UPDATE, w porównaniu z bazą danych Oracle” oraz „Baza danych Oracle cechuje się szybszym wykonywaniem zapytań typu DML na danych binarnych, w porównaniu z Microsoft SQL Server'em”.

Słowa kluczowe: wydajność baz relacyjnych; operacje DML; aplikacja desktopowa; Oracle; MSSQL

* Autor do korespondencji.

Adresy e-mail: grzegorz.dziewit@pollub.edu.pl, jakub.korczynski@pollub.edu.pl

Performance analysis of relational databases Oracle and MS SQL based on desktop application

Grzegorz Dziewit*, Jakub Korczyński*, Maria Skublewska-Paszkowska

^a Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. Comparison of efficiency is not a trivial phenomenon because of disparities between different database systems. This paper presents a methodology of comparing relational database systems in respect of mean time of execution individual DML queries containing subqueries and conjunction of tables. The presented methodology can be additionally accommodated to studies of efficiency in a range of database system itself (study of queries executed directly in database engine). The described methodology allows to receive statement telling which database system is better in comparison to another in dependency of functionalities fulfilled by external application. In the article the analysis of mean time of execution individual DML queries was performed. Two research hypotheses have been put forward: "Microsoft SQL Server database system needs less time to execute INSERT and UPDATE queries than Oracle database" and "Oracle database system needs less time to execute DML queries with binary data than SQL Server"

Keywords: relational databases efficiency; DML operations; desktop application; Oracle; MSSQL

*Corresponding author.

E-mail addresses: grzegorz.dziewit@pollub.edu.pl, jakub.korczynski@pollub.edu.pl

1. Wstęp

W dzisiejszych czasach bazy danych są często stosowane w rozwiązaniach biznesowych. Na przestrzeni ostatnich lat powstało bardzo wiele różnorodnych systemów bazodanowych nierelacyjnych i relacyjnych (Oracle, MS SQL, My SQL, etc.). Ten ostatni typ systemów jest bardzo popularny pod względem użycia, jednakże w związku z narastającą liczbą zróżnicowanych bazodanowych systemów relacyjnych należy wprowadzić kryterium porównawcze, które pozwoli ocenić wydajność tych baz danych. Zanim zostanie wprowadzone to kryterium należy wspomnieć o tym, że niemalże każda aplikacja realizuje podobne funkcjonalności gromadzenia danych. Wprowadzone kryterium musi umożliwiać otrzymywanie wyników, które będą stanowić możliwość porównania wspomnianych relacyjnych baz danych, których działania koordynowane są z działaniem zewnętrznych aplikacji w tym przypadku desktopowych. Celem artykułu jest przeprowadzenie analizy średnich czasów

wykonania poszczególnych zapytań bazodanowych typu DML. To kryterium zostało dobrane, ponieważ operacje DML są prawdopodobnie najczęstszym typem wykonywanych działań na systemie bazodanowym. Na potrzeby pracy badawczej została opracowana aplikacja desktopowa wspomagająca zarządzanie warsztatem samochodowym zawierająca moduł testowy odnoszący się do przeprowadzanych badań.

Mając na uwadze badania literaturowe przeprowadzone w celu pogłębienia wiedzy z zakresu wydajności systemów bazodanowych (rozdział 2), sformułowane zostały następujące hipotezy:

- System bazodanowy Microsoft SQL Server charakteryzuje się krótszym czasem wykonania zapytań typu INSERT oraz UPDATE, w porównaniu z bazą danych Oracle

- Baza danych Oracle cechuje się szybszym wykonywaniem zapytań typu DML na danych binarnych, w porównaniu z Microsoft SQL Server'em.

Autenticzność przedstawionych hipotez zostanie zweryfikowana w dalszej części artykułu na podstawie przeprowadzonych badań.

2. Przegląd literaturowy

Przegląd literaturowy został podzielony ze względu na kryteria, które zostały wyszczególnione w sposób następujący:

- rodzaj zapytań wykorzystanych w badaniach,
- wykorzystane narzędzia pomocnicze,
- sposób podziału danych ze względu na ich typ oraz rozmiar,
- sposób przechowywania danych (internal storage, external storage),
- sprzęt wykorzystany do badań.

Pierwszym z kryteriów był rodzaj zapytań wykorzystywanych w badaniach wydajnościowych. Pierwszą pozycją niosącą przekaz merytoryczny z zakresu badania wydajności relacyjnych baz danych jest praca P.Singh'a i innych [1]. W danej pracy zostały wykorzystane operacje DML tj. SELECT, UPDATE, DELETE. Dodatkowo zostały użyte podzapytania oraz funkcje agregujące. Inną pracą o podobnym kontekście jest artykuł M. Shapiro i innych [2], których metodyka badań również zawierała użycie operacji DML z tym że nie obejmowała ona zapytań DELETE. Zamiast tej operacji obiektem badań zostało zapytanie typu INSERT. Ostatnią z wymienionych podobnych prac jest pozycja autorstwa I. Khawara oraz innych [3]. Metodyka badań jest porównywalna z pracą P.Singh'a [1], ponieważ skupiona jest ona na zapytaniach bazodanowych tego samego rodzaju.

W niniejszym artykule zostaną poddane testom operacje DML tj. SELECT, INSERT, UPDATE ze względu na najczęstsze ich występowanie podczas przeprowadzania operacji zarządzania relacyjnym systemem bazodanowym. Przeanalizowanie wyników badań odnośnie tych operacji pozwoli na porównanie badanych systemów bazodanowych w kontekście czasu wykonania zapytań poprzez zewnętrzną aplikację desktopową.

Kolejnym kryterium sklasyfikowania pozycji przeglądu były wykorzystane narzędzia pomocnicze. To kryterium jest istotną perspektywą, ponieważ w żadnej z wymienionych prac nie zostały podane szczegółowe informacje odnośnie sposobu pozyskiwania danych wydajnościowych. Wyjątkiem jest jedynie praca Singh'a [1], w której wykorzystane zostało narzędzie systemu Windows – Task Manager, który umożliwia pozyskiwanie danych odnośnie zużycia procesora, jednostek pamięci twardej jak i pamięci RAM. Najważniejszym jednak punktem dla zakresu tematycznego pracy był czas wykonania zbioru operacji przez dany proces bazodanowy. W podanym sposobie pracy Singh'a i innych [1] możliwe jest wystąpienie nieścisłości ze względu na to iż na wspomniany zbiór operacji mogły składać się dodatkowe działania procesu bazodanowego wykraczające poza czas działania należącego do egzekucji konkretnych zapytań bazodanowych.

Opierając się na tej tezie, niniejszy artykuł zawierać będzie rozwiązanie niwelujące potencjalne wady opisanego rozwiązania w postaci wykorzystania widoków bazodanowych dla obydwu baz relacyjnych Oracle oraz MS SQL. Wykorzystanie to będzie polegać na pobieraniu danych z widoków, które dostarczają danych dotyczących czasu wykonania jedynie poszczególnych zapytań bazodanowych w odseparowaniu od zewnętrznych działań, które mogłyby pojawiać się podczas uruchomienia/działania danej operacji DML.

We wszystkich pracach zawarta została informacja o rozmiarze danych w postaci liczby przechowywanych rekordów, rozmiarze poszczególnych danych – zajętości pamięci dyskowej oraz samym typie danych. Jednakże żadna z prac nie zawierała wszystkich tych informacji. Aby rzetelność badań była spełniona w jak najwyższym stopniu, należałoby dokładnie opisać, sklasyfikować zróżnicowanie danych pod względem rozmiaru oraz typu, dlatego też w pracy zostanie zawarta klasyfikacja podmiotów badań w obrębie której zostanie zaprojektowana metodyka przedmiotu badań, którym jest czas wykonania poszczególnych zapytań bazodanowych.

Podobna zależność ukazuje się z perspektywy sposobu przechowywania danych. Sposób przechowywania dotyczy wewnętrznego przechowywania danych (surowych umieszczonych wewnątrz bazy) lub zewnętrznego (w bazie przechowywane są jedynie referencje do surowych danych) co może wpływać na jej wydajność. Jedynie praca M.Shapiro [2] zawiera informacje o sposobie przechowywania danych, ponieważ kryterium jego badań były obydwa wspomniane sposoby przechowywania danych. Praca ta zawiera informacje o podziale danych wielkości tj. 0.5 MB oraz 5MB.

Niniejsza praca będzie zawierać podobną charakterystykę jednakże o większych rzędach rozmiarów danych, z powodu funkcjonalności aplikacji desktopowej współpracującej z relacyjnymi bazami danych Oracle oraz MS SQL. Wspomnianą funkcjonalnością jest przechowywanie danych binarnych video o rozmiarach większych niż 100 MB.

W odniesieniu do sposobu składowania danych niniejsza praca ograniczy się do przetestowania sposobu wewnętrznego, ponieważ przy tym podejściu łatwiej jest zbadać wydajność bazy danych dlatego, iż referencje są o wiele mniejsze pod względem rozmiaru niż surowe dane.

Najważniejszym i ostatnim kryterium są wyniki zebrane podczas przeprowadzonych badań, a raczej wnioski wypływające z tych wyników. Z uwagi na brak możliwości krótkiego stwierdzenia, która z baz danych jest najbardziej wydajna wprowadzono podział wyników uzależniony od typu operacji DML. Poniżej zamieszczone zostały informacje wynikające z poszczególnych prac.

Według pracy Singh'a [1] wnioski istotne dla tej pracy niosą za sobą informację, iż baza Oracle okazała się mniej wydajna niż MS SQL w przeprowadzonych eksperymentach badawczych. Praca M.Shapiro i innych [2], która zajmowała się porównaniem samych operacji względem jednej bazy Oracle przedstawia następujące wnioski:

- dla operacji typu SELECT bardziej wydajne okazało się podejście z przechowywaniem plików binarnych bezpośrednio w bazie danych,
- dla operacji typu INSERT/UPDATE bardziej wydajne okazało się przechowywanie plików poza bazą danych, zaś samej bazie jedynie referencje do tych plików,
- wraz ze wzrostem wielkości bazy danych oba podejścia (internal storage, external storage) wykazywały spadek wydajności,
- wraz ze wzrostem wielkości plików obie bazy potrzebowały więcej czasu na ich przetworzenie.

Praca I. Khawary [3] przedstawia szereg wyników, które zostały przedstawione w sposób dzięki któremu można łatwo sformułować wnioski. Metodyka zakładała, że miało się odbyć siedem eksperymentów:

operacja typu select (324500 rekordów), z następującymi wynikami:

- MySQL,
- MS SQL,
- Oracle,

operacja typu select z klauzulą where (wyszukiwanie za pomocą ID, który jest primary key), z następującymi wynikami:

- MySQL,
- Oracle 3,
- Oracle,

operacja typu update na jednej kolumnie w obrębie całej tabeli (324500 rekordów), z następującymi wynikami:

- MS SQL,
- Oracle,
- MySQL,

przeniesienie danych z jednej tabeli do drugiej o takiej samej strukturze, z tą różnicą, że w tabeli docelowej nie ma primary key, z następującymi wynikami:

- Oracle,
- MS SQL,
- MySQL,

operacja typu select z klauzulą order by, z następującymi wynikami:

- MySQL,
- MS SQL,
- Oracle,

operacja typu select z klauzulą group by, z następującymi wynikami:

- MySQL,
- MS SQL,
- Oracle,

operacja typu select ze złączeniem dwóch tabel klauzulą join, z następującymi wynikami:

- MySQL,
- MS SQL,
- Oracle.

Porównanie wydajności sprowadziło się wówczas do zestawienia czasów w obrębie danego zapytania i stwierdzenia, które jest szybciej wykonywane w danym systemie bazodanowym. Wspomniane zestawienie zostało zastosowane w większości prac. Ten artykuł naukowy również będzie zawierać opis z podziałem na rodzaj zapytań ale też będzie zawierać wnioski odnośnie samego kontekstu kooperacji systemu bazodanowego z utworzoną aplikacją desktopową wspomagającą zarządzanie warsztatem samochodowym.

3. Aplikacja desktopowa

Przedstawione w artykule badania są przeprowadzone przy użyciu aplikacji desktopowej. Służy ona do wspomagania zarządzania warsztatem samochodowym w postaci prowadzenia ewidencji napraw, pracowników a także poradników napraw w postaci video. Ta ostatnia cecha odróżnia tę aplikację od innych o podobnym przeznaczeniu, ponieważ ten sam proces naprawy może różnić się w zależności od modelu samochodu. Wówczas pracownik jest zwolniony od pamiętania wszystkich sposobów napraw i wystarczy aby był zalogowany w aplikacji i miał dostęp do modułu zarządzania poradnikami w celu obejrzenia danego sposobu naprawy. Aplikacja została napisana w języku programistycznym C# przy wykorzystaniu technologii MVVM (ang. Model-View-ViewModel)[4]. Wykorzystuje ona dwa silniki bazodanowe Oracle[5] oraz MS SQL[6]. Aplikacja zawiera moduł testujący, za pomocą którego realizowane są badania.

4. Systemy bazodanowe Oracle

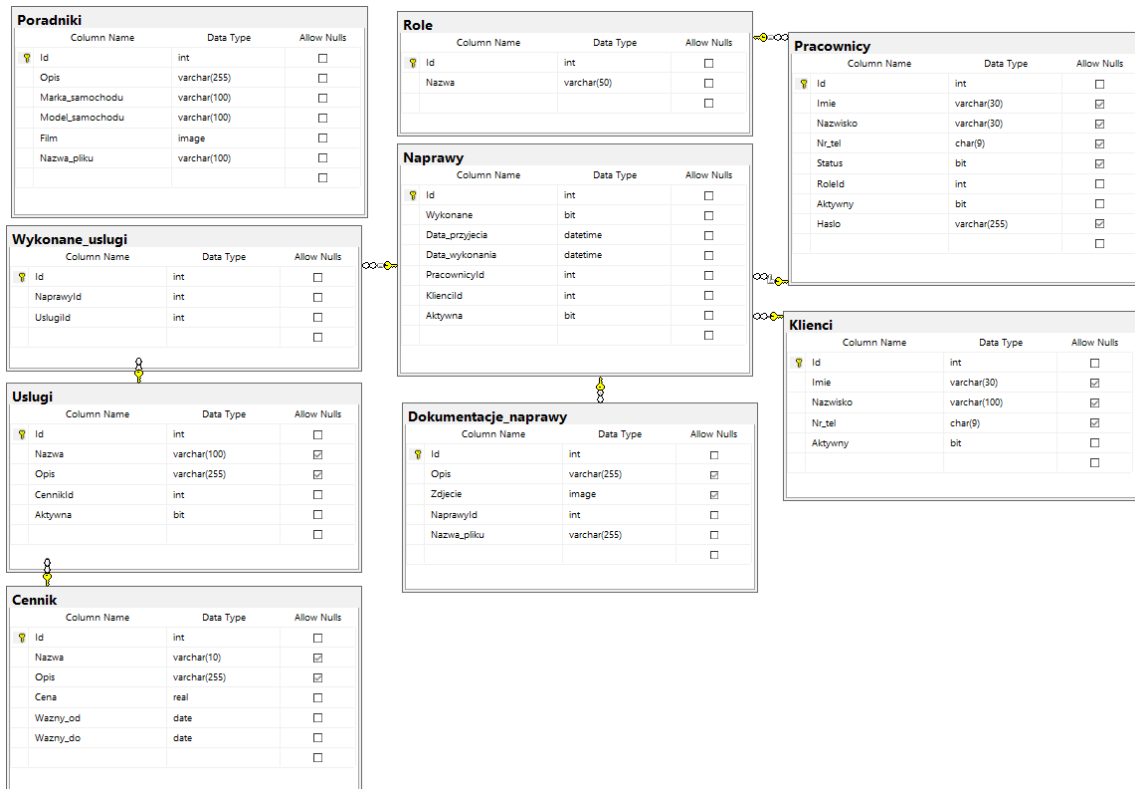
Oracle jest to system zarządzania bazą danych stworzony przez amerykańskie przedsiębiorstwo Oracle Corporation, posługujący się językiem zapytań SQL (ang. Structured Query Language). Technologia ta może być wykorzystywana zarówno lokalnie, jak również w globalnej sieci, ponadto twórcy udostępniają dedykowany komponent służący do komunikacji z bazą danych, co jest bardzo istotne dla programistów korzystających z tego rozwiązania.[5]

4.2 MS SQL

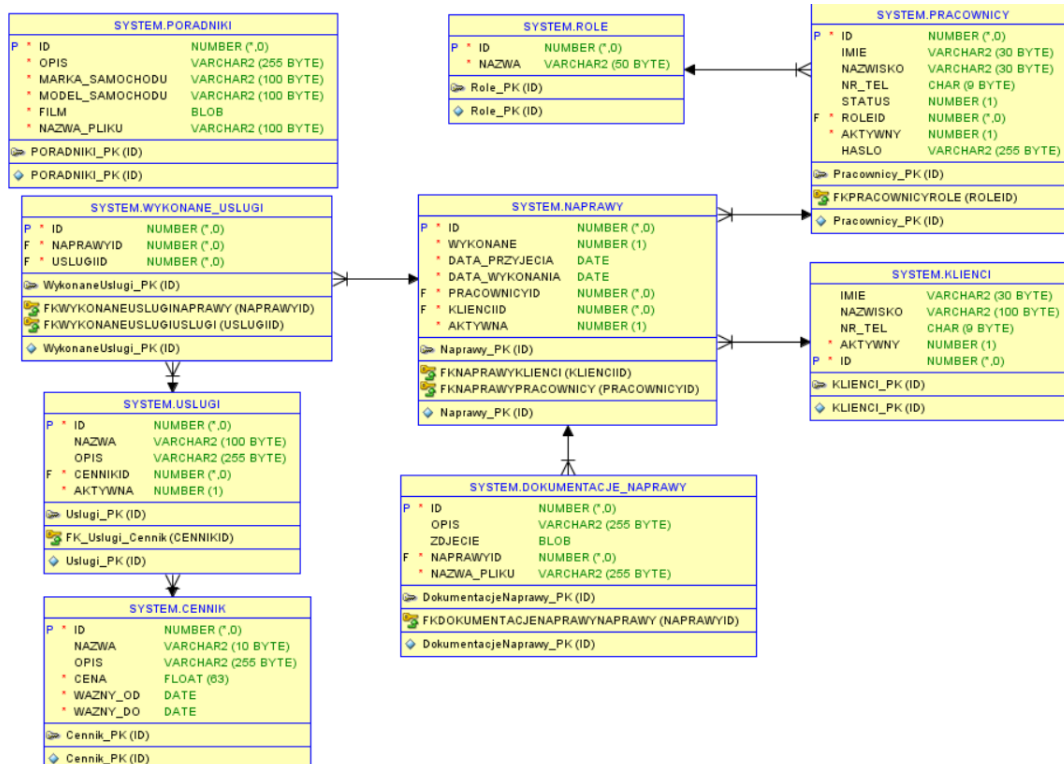
Microsoft SQL Server jest to system zarządzania bazą danych, którego twórcą jest dobrze znana firma Microsoft. Wykorzystuje on język zapytań T-SQL (ang. Transact-SQL), który jest implementacją języka SQL, dodającą zbiór nowych

rozszerzeń. Wspomniana baza danych w wersji 2016 posiada wiele dodatkowych możliwości takich jak analiza operacji w czasie rzeczywistym lub wizualizacja danych [6].

Na rys.1 oraz rys.2 przedstawione zostały schematy baz danych wykorzystanych podczas przeprowadzania badań w formie diagramów ERD (ang. Entity Relationship Diagram) [7].



Rys. 1. Diagram ERD dla bazy danych Ms SQL



Rys. 2. Diagram ERD dla bazy danych Oracle

5. Metodyka

W niniejszym podrozdziale zostanie opisana metodyka, która dotyczy badania czasu odpowiedzi systemów bazodanowych MS SQL oraz Oracle w zależności od wykonywanych zapytań wywoływanych przez zewnętrzną

(aplikacja pozasystemowa względem bazy danych) aplikację desktopową wspomagającą zarządzanie warsztatem samochodowym. Obiektem badań będzie czas realizacji zapytań bazodanowych. Zapytania budowane są przez zewnętrzną aplikację, która przekazuje treści zapytań do danego systemu bazodanowego, a następnie są one

wykonywane poprzez konkretny system. W momencie ukończenia wykonywania danego zapytania w systemie bazodanowym następuje zapisanie danych w widokach bazodanowych odnoszących się do statystyk wykonanego zapytania przekazanego przez aplikację. Wspomniane widoki to:

- **baza danych Oracle:** V\$SQL – widok wyświetlający statystyki wykonanych zapytań, gdzie jeden wiersz odnosi się do pojedynczego zapytania. Dane statystyczne aktualizowane są po wykonaniu polecenia, bądź w trakcie jego trwania (co 5 sekund), jeżeli trwa on dłuższy czas. W kontekście badania czasu wykonania danego zapytania, najważniejszą kolumną wspomnianego widoku jest „ELAPSED_TIME”, gdzie przechowywana jest informacja o czasie wykonania konkretnego polecenia w mikrosekundach [8],
- **baza danych MS SQL:** sys.dm_exec_query_stats – widok wyświetlający statystyki wykonanych zapytań. Jeden wiersz wynikowy odpowiada statystykom jednego wykonanego zapytania. Kolumną przechowującą dane odnośnie czasu wykonania zapytania jest „total_elapsed_time”, a jednostką czas jest mikrosekunda [9].

Pobieranie danych statystycznych z widoków jest realizowane poprzez zapytania napisane w języku programistycznym SQL. Wspomniane zapytania z podziałem na bazę danych przedstawiają się następująco:

- Oracle: zapytanie pobierające statystyki wykonanych poleceń oraz czyszczące pamięć podręczną (rys. 3);
- SQL Server: zapytanie pobierające statystyki wykonanych poleceń oraz czyszczące pamięć podręczną (rys. 4).

```
----- zapytanie wydajnościowe
select
LAST_LOAD_TIME,
ELAPSED_TIME,
MODULE,
SQL_TEXT_elapsed,
EXECUTIONS,
from v$sql
where Module like '%Warsztat%'
order by LAST_LOAD_TIME desc;

----- czyszczenie pamięci podręcznej
alter system flush shared_pool;
```

Rys. 3. Zapytanie pobierające statystyki wykonanych poleceń oraz czyszczące pamięć podręczną w bazie danych Oracle

Dane otrzymywane w wyniku przeprowadzanych badań to dane numeryczne odpowiadające krotności wykonania danego zapytania oraz czas odpowiedzi zapytania wyrażony w mikrosekundach. Dane wynikowe zostały podzielone ze względu na liczbę wykonań danego zapytania rzędu 500, 1000, 5000 dla danych małego rozmiaru, oraz 10, 50, 100 dla danych o większym rozmiarze. Przy każdorazowym wykonywaniu badania liczba rekordów przechowywanych w tabelach odpowiadała liczbie rekordów, na której przeprowadzano poszczególne operacje DML. Rozmiar danych jest dodatkowym kryterium podziału danych ze względu na ich typ. Dane tekstowe odpowiadają rozmiarowi mniejszemu niż 1MB, zdjęcia w postaci danych binarnych

odpowiadają rozmiarowi z zakresu 1-50 MB, natomiast największym rozmiarem danych charakteryzują się filmy również w postaci binarnej, których zajętość pamięciowa wynosi więcej niż 50 MB.

```
----- zapytanie wydajnościowe
SELECT SUBSTRING(qt.TEXT, (qs.statement_start_offset/2)+1,
((CASE qs.statement_end_offset
WHEN -1 THEN DATALENGTH(qt.TEXT)
ELSE qs.statement_end_offset
END - qs.statement_start_offset)/2)+1) as command,
sum(qs.execution_count),
sum(qs.total_elapsed_time)
FROM sys.dm_exec_query_stats qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) qt
CROSS APPLY sys.dm_exec_query_plan(qs.plan_handle) qp
group by SUBSTRING(qt.TEXT, (qs.statement_start_offset/2)+1,
((CASE qs.statement_end_offset
WHEN -1 THEN DATALENGTH(qt.TEXT)
ELSE qs.statement_end_offset
END - qs.statement_start_offset)/2)+1)

----- czyszczenie pamięci podręcznej
DBCC FREEPROCCACHE
DBCC DROPCLEANBUFFERS
```

Rys. 4. Zapytanie pobierające statystyki wykonanych poleceń oraz czyszczące pamięć podręczną w bazie danych MS SQL

Poniżej przedstawiony został schemat postępowania podczas przeprowadzania badania:

- 1) Uruchomienie wybranego serwera bazy danych.
- 2) Uruchomienie aplikacji desktopowej i zalogowanie się jako konkretny użytkownik.
- 3) Wybranie opcji modułu testującego aplikacji.
- 4) Wykonanie zapytania przeznaczonego do badania wydajnościowego.
- 5) Uruchomienie zapytania statystycznego przy pomocy SQL Developer lub SQL Management Studio.
- 6) Pobranie wyników oraz wyczyszczenie pamięci podręcznej.

Czyszczenie pamięci podręcznej wiąże się także z czyszczeniem planu zapytań w obydwu bazach danych i ta operacja wykonywana jest każdorazowo po przeprowadzonym badaniu. W bazie Oracle zapytaniem czyszczącym zarówno pamięć podręczną jak i plan zapytań było zapytanie przedstawione na Rys. 3 [10][11]. Natomiast zapytania wymienione kolejno na Rys. 4 odpowiadały za wyczyszczenie planu zapytania oraz wyczyszczenie pamięci podręcznej dla bazy MS SQL [12]. Wszystkie badania wydajnościowe są przeprowadzone za pomocą zapytań DML, wyłączając z nich operacje DELETE. Dodatkowo zapytania zawierają złączenia (INNER JOIN), a także podzapytania.

6. Wyniki

Wszystkie wyniki stanowią średnią arytmetyczną czasów wykonań poszczególnych zapytań.

Tabela 1. Wyniki pomiarów dla operacji select dla danych binarnych

Liczba rekordów	MS SQL		Oracle	
	Średni czas [µs]	Std	Średni czas [µs]	Std
10	3300906	148136	5844	520
30	10487079	66586	6244	918
50	218454820	119136	5486	1242

Tabela 2. Wyniki pomiarów dla operacji update dla danych binarnych

Liczba rekordów	MS SQL		Oracle	
	Średni czas [µs]	Std	Średni czas [µs]	Std
1	5400762	2332554	8198944	1529275
3	19262326	6387159	26779215	5212174
5	20176755	1813704	61494904	15772001

Tabela 3. Wyniki pomiarów dla operacji select z pojedynczej tabeli na danych tekstowych/numerycznych

Liczba rekordów	MS SQL		Oracle	
	Średni czas [µs]	Std	Średni czas [µs]	Std
10000	16401	2441	6183	545
50000	101555	3828	15937	965
100000	199985	4749	27222	1376

Tabela 4. Wyniki pomiarów dla operacji insert na danych tekstowych/numerycznych

Liczba rekordów	MS SQL		Oracle	
	Średni czas [µs]	Std	Średni czas [µs]	Std
500	22974	511	95579	3370
1000	44404	723	189186	5002
5000	213822	4015	915339	15786

7. Analiza wyników

Celem niniejszej pracy było porównanie relacyjnych baz danych Oracle oraz SQL pod względem czasów wykonania poszczególnych zapytań bazodanowych. Badania zakończyły się sukcesem poprzez zastosowanie opisanej metodyki. Wyniki badań są dosyć zróżnicowane, baza Oracle zdecydowanie lepiej wykonuje operacje SELECT zarówno na danych binarnych (tabela 6.1) oraz tekstowych (tabela 6.3) w porównaniu do bazy MS SQL, natomiast ta druga o wiele lepiej realizuje operacje UPDATE (tabela 6.2) oraz INSERT (tabela 6.4) również w odniesieniu do obydwu typu danych. We wstępie odniesiono się do potrzeby wprowadzenia kryterium porównawczego, które pozwoli porównać relacyjne systemy bazodanowe współpracujące z zewnętrznymi w stosunku do nich aplikacjami. Kryterium to jest uzależnione od częstotliwości występowania danego zapytania bazodanowego przy wykorzystaniu zewnętrznej aplikacji. Zatem najlepszą bazą danych jest ta, która najszybciej wykonuje najczęściej wykonywane zapytania a wolniej te, które w rzadszym stopniu pojawiają się podczas zarządzania bazą danych. Mianowicie, jeśli wiadomo, iż działanie aplikacji najczęściej sprowadza się do wywoływania operacji SELECT wówczas lepszą bazą danych jest baza Oracle. W innym przypadku, gdy aplikacja zleca najczęściej wykonanie zapytań INSERT oraz UPDATE systemowi bazodanowemu, wówczas lepszą bazą danych okaże się baza danych MS SQL. Odnosząc się do głównego kontekstu czyli aplikacji wspomagającej zarządzanie warsztatem samochodowym, można stwierdzić iż lepszą bazą danych do tego typu aplikacji jest baza Oracle. Mimo iż dane statystyczne pokazują, że MS SQL jest lepszy od bazy Oracle w operacjach UPDATE (tabela 6.2) oraz INSERT (tabela 6.4), to operacje UPDATE są znacznie rzadziej wykonywane niż operacje SELECT. Dodatkowo można zauważyć, że rząd wielkości czasu wstawiania 5000 wierszy do tabeli zamyka się poniżej 1s (tabela 6.4). W związku z tym te niedoskonałości bazy Oracle są o wiele mniej znaczące w odniesieniu do różnicy w pobieraniu danych

binarnych z tabeli. Ta różnica pomiędzy bazami jest ogromna, gdyż czas pobierania danych binarnych przez bazę Oracle zamyka się w czasie mniejszym niż 0.01s przy o wiele dłuższym czasie (10s) realizacji tego zapytania przez bazę MS SQL (tabela 6.1). Różnice te mogą wynikać ze sposobu przechowywania danych, ponieważ w bazie Oracle wykorzystanym typem danych binarnych był typ BLOB[13], natomiast w bazie danych MS SQL typem danych był image[14]. Ostatnim zagadnieniem, godnym uwagi jest fakt, że w badaniu średniego czasu wykonania operacji update na danych binarnych (Tabela 2) wykonywano operacje na maksymalnie 5 rekordach. Tak mała liczba wynika z czasu przetwarzania danych binarnych w tej operacji, ponieważ rozmiar tych rekordów przekraczał łącznie nawet 500 MB (duży rozmiar wiąże się z długim czasem przetwarzania) zajętości danych, które były przechowywane bezpośrednio w bazie danych. Mimo małej liczby rekordów przechowywanej w bazie danych wystarczyła ona na przeprowadzenie badań oraz analizę wyników, których dość długi czas wykonania został przedstawiony w Tabeli 2.

Podsumowując, wykonane badania wykazują, iż baza danych Oracle jest lepsza do tego typu aplikacji.

8. Podsumowanie

Celem niniejszego artykułu była analiza wyników przy użyciu zaprojektowanej metodyki, który udało się osiągnąć. Metodyka okazała się skuteczna oraz cechuje się możliwością dostosowania do innych typów badań jakimi mogą być np. badania wydajnościowe zapytań uruchamianych przez sam system bazodanowy. W związku z przeprowadzonymi badaniami postawiono hipotezy:

- System bazodanowy Microsoft SQL Server charakteryzuje się krótszym czasem wykonania zapytań typu INSERT oraz UPDATE, w porównaniu z bazą danych Oracle,
- Baza danych Oracle cechuje się szybszym wykonywaniem zapytań typu DML na danych binarnych, w porównaniu z Microsoft SQL Server.

Pierwsza z wymienionych hipotez została potwierdzona, gdyż badania wykazały, iż baza danych MS SQL lepiej realizuje operacje INSERT i UPDATE niż baza Oracle. Druga z hipotez została obalona, mimo że baza Oracle znacznie lepiej realizuje operacje SELECT dla danych binarnych to baza MS SQL szybciej wykonuje operacje UPDATE. W wyniku badań okazało się, że baza Oracle lepiej nadaje się do współpracy z aplikacjami zewnętrznymi.

Podsumowując całość artykułu, może on służyć jako źródło informacji odnośnie projektowanej metodyki w celu analizy wydajności innych relacyjnych systemów bazodanowych.

Literatura

- [1] Singh P., Sharma S., Kaur S.: Performance Analysis of Different DBMS Systems. International Journal of Advance Foundation And Research In Science & Engineering, 2015,1. 1-9.
- [2] Shapiro M., Miller E.: Managing databases with binary large objects. 16th IEEE Symposium on Mass Storage Systems in cooperation with the 7th NASA Goddard Conference on Mass Storage Systems and Technologies, San Diego 1999

- [3] Khawar I., Kamran A., Syed B., Syed A.: Huge and Real-Time Database Systems: A Comparative Study and Review for SQL Server 2016, Oracle 12c & MySQL 5.7 for Personal Computer. *Journal of Basic and Applied Sciences*, 2017, 13.
- [4] Mihailescu M., Sorensen E.: Model-View-ViewModel (MVVM) Design Pattern using Windows Presentation Foundation (WPF) Technology. *Megabyte Journal*, 2010
- [5] Castel J.: Oracle 12c: SQL. Cengage Learning, 2015
- [6] Sanka, D., Durkin, W., Radivojevic, M.: SQL Server 2016 Developer's Guide. Packt Publishing, 2017
- [7] Specyfikacja diagramu ERD, <https://msdn.microsoft.com/pl-pl/library/projektowanie-baz-danych--diagramy-erd-relacje-miedzy-tabelami-zwiazki-rekordy.aspx>, [2018-06-20]
- [8] Opis widoku V\$SQL (Oracle), https://docs.oracle.com/cd/B19306_01/server.102/b14237/dynviews_2113.htm#REFRN30246, [2018-05-15]
- [9] Opis widoku sys.dm_exec_query_stats (Ms SQL Server), <https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-exec-query-stats-transact-sql?view=sql-server-2017>, [2018-05-15]
- [10] Opis shared_pool dla bazy Oracle, http://www.dba-oracle.com/concepts/shared_pool_concepts.htm, [2018-06-20]
- [11] Opis zapytania czyszczącego pamięć podręczną oraz plan zapytań dla bazy Oracle. http://www.dba-oracle.com/tp_Oracle_shared_pool_routine_flush.htm, [2018-06-20]
- [12] Opis zapytań czyszczących pamięć oraz plan zapytań dla bazy MS SQL, <http://www.sqlpedia.pl/aspekty-wydajnosciowe-zapytan-sql/>, [2018-06-20]
- [13] Opis bazodanowego typu BLOB w systemie Oracle, <https://docs.oracle.com/javadb/10.8.3.0/ref/rrefblob.html>, [2018-05-15]
- [14] Opis bazodanowego typu Image w systemie SQL Server, <https://docs.microsoft.com/en-us/sql/t-sql/data-types/text-text-and-image-transact-sql?view=sql-server-2017&viewFallbackFrom=sql-server-2017>, [2018-05-15]