

Oracle 19c, SQL Server 2019, Postgresql 12 and MySQL 8 database systems comparison

Porównanie systemów bazodanowych Oracle 19c, SQL Server 2019, PostgreSQL 12 oraz MySQL 8

Arkadiusz Solarz*, Tomasz Szymczyk

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This article presents a comparative analysis of four popular database technologies. Commercial Oracle Database and SQL Server systems have been compared with open source database management systems: PostgreSQL and MySQL. These systems have been available on the market for over a dozen years. Versions released in 2019 were selected for testing and comparison. For the purposes of the comparative analysis, a database schema was developed and instantiated. Then, test scenarios have been developed. They have been prepared on the basis of the most popular operations performed with the use of database systems.

Keywords: relational database; performance; SQL

Streszczenie

W artykule przedstawiona została analiza porównawcza czterech technologii bazodanowych. Komercyjne systemy Oracle Database i SQL Server porównane są z darmowymi systemami do zarządzania bazą danych: PostgreSQL i MySQL. Systemy te dostępne są na rynku od kilkunastu lat, do testów i porównania wybrane zostały wersje wydane w 2019 roku. Na potrzeby analizy porównawczej zaprojektowano oraz utworzono schemat bazy danych. Następnie opracowano scenariusze testowe. Przygotowane zostały one w oparciu o najpopularniejsze operacje wykonywane z wykorzystaniem systemów bazodanowych.

Słowa kluczowe: relacyjne bazy danych; wydajność; SQL

*Corresponding author

Email address: arkadiusz.solarz@pollub.edu.pl (A. Solarz)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

W obecnych czasach szeroko rozwinięte są aplikacje biznesowe, programy komputerowe oraz aplikacje internetowe. Rozwiązania te w dużym stopniu bazują na danych oraz zarządzaniu nimi. Istotnym elementem takich aplikacji jest zarówno przetwarzanie nowych danych jak i zapewnienie dostępu do danych zapisanych wcześniej [1,2].

Przechowywanie i odpowiednie zabezpieczenie danych stanowi dla twórców oprogramowania niemałe wyzwanie. Istnieje wiele sposobów przechowywania danych, a wybór odpowiedniej metody powinien być świadomy i przemyślany [1,3]. Zastosowanie arkusza kalkulacyjnego sprawdzi się w przypadku małej grupy użytkowników, która nie potrzebuje wielu skomplikowanych funkcji. Systemy bazodanowe wykorzystywane są do zarządzania znacznie większymi zbiorami danych, zapewniając jednocześnie dostęp wielu użytkownikom. Istnieje wiele różnych typów baz danych takich jak: relacyjne, obiektowe, relacyjno-obiektowe, NoSQL, grafowe czy też hurtownie danych [4-7]. Rozwój w dziedzinie sprzętu oraz programowania sprawił że powstają nowoczesne rozwiązania takie jak bazy danych chmurowe, wielomodelowe bazy danych, samoczynne bazy danych lub dokumentowe bazy danych [1,5,7,8].

Na rynku dostępnych jest wiele systemów zarządzania bazą danych, występuje zarówno oprogramowanie komercyjne jak i programy na licencji open source, różniące się pod względem obsługiwanego modelu danych, oraz stopnia zaawansowania i dostępnych funkcjonalnościach [9]. Na rysunku 1 przedstawiony jest ranking systemów bazodanowych bazujący na ich popularności [10]. W artykule przedstawiono porównanie czterech systemów: Oracle Database 19c, MySQL 8, SQL Server 2019 oraz PostgreSQL 12.

Rank Aug 2020	DBMS	Database Model	Score Aug 2020
1.	Oracle +	Relational, Multi-model	1355.16
2.	MySQL +	Relational, Multi-model	1261.57
3.	Microsoft SQL Server +	Relational, Multi-model	1075.87
4.	PostgreSQL +	Relational, Multi-model	536.77
5.	MongoDB +	Document, Multi-model	443.56
6.	IBM Db2 +	Relational, Multi-model	162.45
7.	Redis +	Key-value, Multi-model	152.87
8.	Elasticsearch +	Search engine, Multi-model	152.32
9.	SQLite +	Relational	126.82
10.	Microsoft Access	Relational	119.86

Rysunek 1: Ranking popularności systemów bazodanowych [10]

2. Środowisko testowe

W celu przetestowania wydajności silników bazodanowych zostało utworzone środowisko testowe, na które

składa się zarówno hardware i software. Wykorzystane oprogramowanie uruchomiono w środowisku Windows 10. Zainstalowane oprogramowanie to: PostgreSQL 12, MySQL 8, SQL Server 2019, Oracle Database 19c

2.1. Baza danych

Omawiane systemy zarządzania bazą danych są narzędziami do obsługi baz danych wykorzystujących model relacyjny [6,10-12]. W tym celu do przeprowadzenia testów utworzona została relacyjna baza danych, której schemat przedstawiony jest na rysunku 2. Przedstawiona baza danych została zaprojektowana do obsługi systemu wypożyczalni samochodów. Celem zastosowania takiej bazy danych jest przetwarzanie transakcyjne (OLTP Online Transaction Processing). Tego typu baz używa się głównie w różnego rodzaju systemach ewidencyjnych takich jak: wypożyczalnie, systemy rezerwacji, sklepy internetowe, biblioteki, bankowość elektroniczna i inne [1,3,4].

Schemat bazy został uproszczony i zmodyfikowany głównie z myślą o przeprowadzeniu testów. Na stworzonym schemacie umieszczono jedynie niezbędne struktury, które pozwolą na traktowanie modelu bazy danych jako jedną spójną całość i wykonywanie zapytań SQL które będą zbliżone do rzeczywistych operacji [13,14].

Na podstawie przedstawionego poniżej schematu zostały przygotowane cztery skrypty SQL tworzące w bazie danych fizyczne tabele oraz powiązania między nimi.

2.2. Metoda testowania

Porównanie wydajności silników bazodanowych polegać będzie na przeprowadzeniu serii testów i zmierzeniu czasu ich wykonania. W celu zmniejszenia niepewności pomiarów, każdy test został wykonany pięciokrotnie, a jako wynik końcowy posłuży średnia arytmetyczna, obliczona z sumy wszystkich cząstkowych czasów wy-

konania zapytania - t_i podzielonej przez liczbę pomiarów - n .

$$t = \frac{\sum_{i=1}^n t_i}{n} \quad (1)$$

Rozpoczęcie testowania następuje od jednego rekordu, w każdej kolejnej próbie liczba rekordów w tabeli jest odpowiednio zwiększana poprzez wstawienie danych testowych.

Podczas testów sprawdzone zostaną najczęściej wykonywane operacje związane z relacyjnymi bazami danych: wyszukiwanie danych w bazie, grupowanie oraz wstawienie danych. Przetestowane zostanie również wykonanie funkcji typowo administracyjnych takich jak utworzenie kopii zapasowej oraz jej przywrócenie [15].

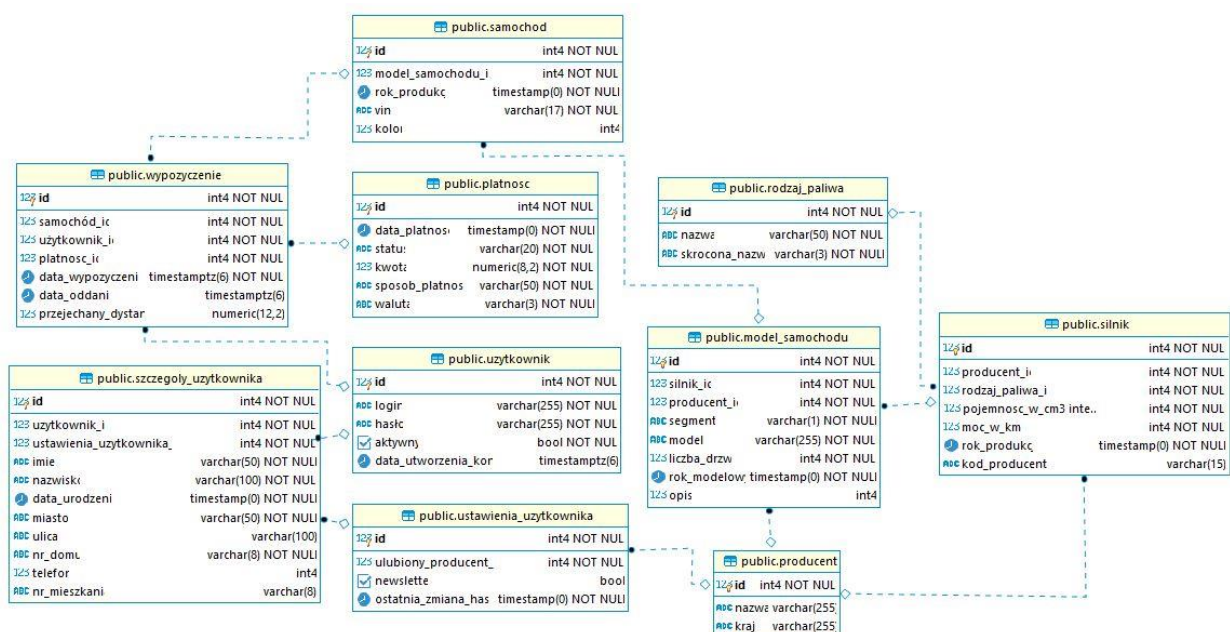
2.3. Technologie oraz narzędzia

Na czas wykonywania zapytania oraz wydajność systemu bazodanowego wpływ ma sprzęt (hardware), na którym jest on uruchomiony. Aby wyniki były niezależne od sprzętu, wszystkie testy zostały wykonane z wykorzystaniem laptopa o następującej specyfikacji:

- System operacyjny: Windows 10
- Procesor: Intel Core i3 6006U
- RAM: 8GB DDR4
- Dysk: SSD 120GB + HDD 500 GB

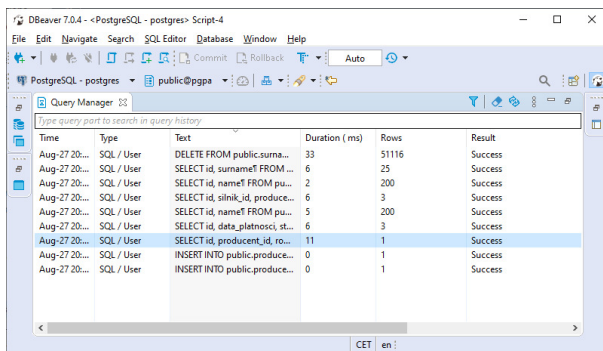
System operacyjny zainstalowany był na dysku SSD, systemy do zarządzania bazą danych na dysku HDD. Podczas testu urządzenie pracowało z włączonym zasilaniem i miało ustawiony tryb „Najwyższa wydajność”. W trakcie wykonywania testów programy, które mogłyby mieć wpływ na zużywanie zasobów laptopa zostały wyłączone.

Narzędziem wykorzystanym do połączenia z bazą danych i przeprowadzenia testów wydajnościowych był klient Dbeaver w wersji 7.0.4. Program ten jest darmo-



Rysunek 2: Schemat autorskiej bazy danych

wy i obsługuje połączenie z badanymi systemami zarządzania bazą danych PostgreSQL, SQL Server, Oracle Database, MySQL oraz wieloma innymi bazami w tym nierelacyjnymi (NoSQL). Narzędzie to zostało wybrane ze względu na dużą funkcjonalność, łatwość konfiguracji i połączenia z bazą danych. Zastosowanie takiego podejścia sprawia, że testy będą wykonywane w sposób jednolity. Do pomiaru czasu wykorzystany zostanie menedżer zapytań, okno programu przedstawione jest na rysunku 3. Widoczne są w nim najważniejsze informacje dotyczące zapytania przetworzonego przez bazę: czas jego wykonania w milisekundach, liczba wierszy, która została zwrócona lub zmodyfikowana oraz status informujący o poprawności wykonania [16].



Rysunek 3: Widok okna programu do pomiaru czasu wykonania zapytania

Dodatkowe narzędzia wykorzystane w testach wydajnościowych to programy dołączone lub zintegrowane z systemem bazodanowym. Wykorzystane zostały narzędzia do tworzenia i przywracania kopii zapasowej bazy danych. Dla systemu PostgreSQL są to `pg_dump` i `pg_restore`. W przypadku Oracle będą to narzędzia `IMP` oraz `EXP`. Do eksportu i importu danych w SQL Server wykorzystane zostanie składanie języka Transact SQL która umożliwi wykonanie tych operacji poprzez odpowiednie zapytanie [17,18].

3. Testy wydajnościowe

Podczas testów wydajnościowych wykonane zostały popularne operacje związane z relacyjnymi bazami danych: wyszukiwanie, grupowanie oraz wstawienie danych. Zmierzone zostały także czasy utworzenia kopii zapasowej oraz jej przywrócenia.

3.1. Wyszukiwanie danych

Scenariusz testowy polegał na zmierzeniu czasu odpowiedzi bazy danych, który potrzebny jest na wyszukanie określonej informacji. W podanym przykładzie wybierani są użytkownicy, którzy na imię mają JAN. (Podczas wstawiania danych wszystkie imiona i nazwiska wpisane zostały w całości z dużych liter, również bazy

Listing 1: Zapytanie do wyszukiwania danych

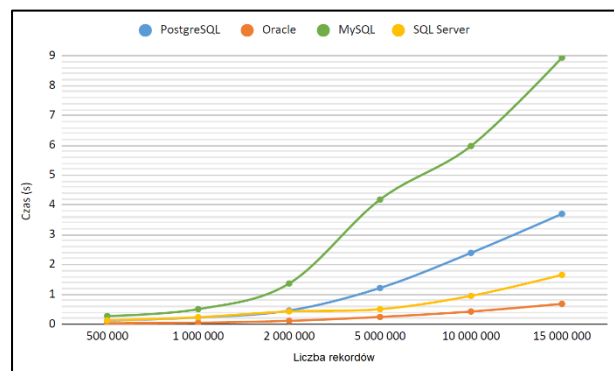
```
SELECT imie, nazwisko, miasto, ulica, nr_domu,
nr_mieszkania, FROM szczegoly_uzytownika
WHERE IMIE = 'JAN';
```

danych podczas tworzenia zostały skonfigurowane tak aby wielkość liter była rozróżniana.) Do wybierania danych posłużyło zapytanie przedstawione na listingu 1.

W tabeli 1 przedstawione są wyniki testów przeprowadzonych dla różnej liczby rekordów w tabeli z użytkownikami. Na rysunku 4 znajduje się wykres, na którym przedstawiono czasy wykonania zapytań, gdy liczba rekordów w tabeli wynosiła 500 000 lub więcej. Wraz ze wzrostem ilości danych różnice uwypuklają się. PostgreSQL potrzebował średnio dwa razy mniej czasu na odpowiedź, natomiast najwydajniejszy okazał się Oracle, który w ostatnim teście był o około sekundę szybszy od SQL Server 2019.

Tabela 1: Pomiary czasu wyszukiwania danych

Liczba rekordów	Czas wykonania (ms)			
	Postgres	Oracle	MySQL	SQL Server
1	3,0	2,4	3,0	3,0
1 000	3,2	3,8	6,4	6,0
5 000	3,0	3,4	6,2	9,0
10 000	6,4	3,0	13,8	9,4
20 000	9,4	3,0	13,4	12,4
50 000	18,6	6,2	32,2	22,0
100 000	28,0	9,4	66,0	28,2
200 000	45,2	12,6	124,8	53,0
500 000	118,6	31,2	274,2	128,2
1 000 000	225,8	49,8	503,0	234,2
2 000 000	456,6	112,2	1364,4	428,0
5 000 000	1215,6	243,2	4180,2	502,8
10 000 000	2391,8	424,2	5977,6	949,8
15 000 000	3700,0	681,8	8941,6	1654,6



Rysunek 4: Wykres czasu wyszukiwania w zależności od liczby rekordów

3.2. Grupowanie danych

Do grupowania danych wykorzystany został skrypt przedstawiony na listingu 2. Wykonywana operacja służy do wyświetlenia liczby klientów wypożyczalni z poszczególnych miast w zacinając od miast z naj-

Listing 2: Zapytanie wykorzystane do grupowania danych

```
SELECT miasto, count(*) AS liczba_klientow
FROM SZCZEGOLY_UZYTOWNIKA GROUP
BY miasto ORDER BY liczba_klientow desc;
```

większą liczbą klientów. Otrzymane rezultaty przedstawione są w tabeli 2. Podobnie jak podczas operacji wyszukiwania danych, najwolniejszy okazał się darmowy system zarządzania bazą danych MySQL 8. Czasy odpowiedzi są kilkanaście razy dłuższe niż w przypadku czołowych systemów: Oracle Database 19c i SQL Server 2019.

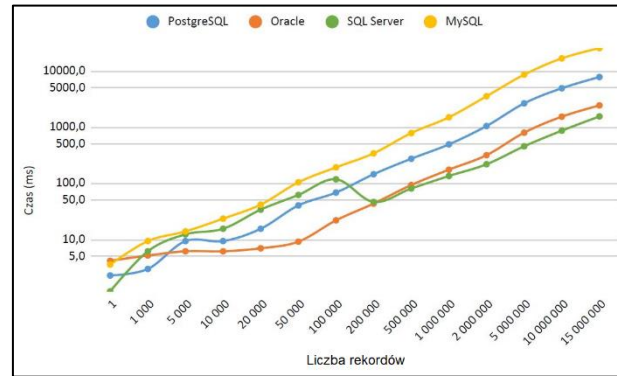
Tabela 2: Pomiary czasu grupowania danych

Liczba rekordów	Czas wykonania (ms)			
	Postgres	Oracle	MySQL	SQL Server
1	2,3	4,2	3,6	1,2
1 000	3,0	5,2	9,4	6,2
5 000	9,4	6,2	14,0	12,4
10 000	9,4	6,2	23,6	15,6
20 000	15,6	7,0	42,0	34,2
50 000	40,6	9,2	104,8	62,4
100 000	68,8	22,0	193,2	118,6
200 000	146,0	43,8	343,8	46,8
500 000	275,4	93,6	789,6	81,2
1 000 000	496,2	175,8	1505,8	135,6
2 000 000	1052,6	318,8	3573,8	219,6
5 000 000	2666,6	801,8	8654,2	458,4
10 000 000	4960,0	1544,8	16880,8	870,4
15 000 000	7861,0	2451,8	25716,2	1562,0

Trzy bazy danych osiągają czasy odpowiedzi poniżej 500 ms w sytuacji, gdy w tabeli znajduje się mniej niż milion rekordów. Najwolniejszy czas osiągnął MySQL, grupowanie danych gdy w tabeli znajduje się 1 000 000 rekordów zajmuje 1,5 sekundy. Czas ten jest dłuższy od czasów pozostałych systemów, lecz jest on akceptowalny w kontekście wykorzystania bazy w systemie OLTP. Powyżej tej liczby rekordów czasy wykonania zapytania wydłużają się proporcjonalnie do liczby rekordów w tabeli. Aby zobrazować różnicę pomiędzy czasami wykonania zapytania, warto spojrzeć na wykres (rysunek 5). Wynika z niego, że w czasie 1,5 sekundy, MySQL jest w stanie wykonać zapytanie dla 1 000 000 rekordów, Oracle potrafi wykonać to samo zapytanie w tym samym czasie dla 10 000 000 rekordów, natomiast SQL Server dla 15 000 000 rekordów.

3.3. Odtwarzanie bazy danych

Odtwarzanie bazy danych jest operacją wykonywaną zdecydowanie rzadziej niż tworzenie kopii zapasowej, jednak niezwykle ważne jest, aby w przypadku awarii lub problemów przywrócić dane i ponownie uruchomienie usługi odbyło się w jak najkrótszym czasie. W omawianym scenariuszu wykorzystane zostały pliki kopii zapasowej przygotowane według następującej metody: w każdym z plików znajduje się wszystkie tabele, w których początkowo znajduje się 71 300 rekordów. Wraz z kolejnymi testami do tabeli z wypożyczeniami wstawiane są kolejne rekordy. Dodawanie danych do tej tabeli ma symulować działanie rzeczywistego systemu, w którym istniejący użytkownicy dokonują wypożyczeń które zapisywane są właśnie w tej tabeli bazodanowej.

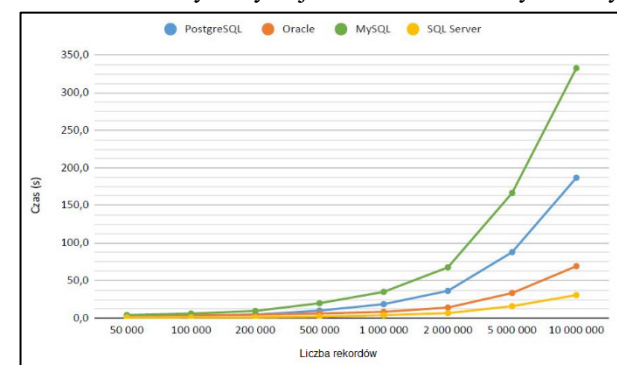


Rysunek 6: Porównanie czasu wykonania zapytania grupującego dane dla różnej liczby rekordów

Tabela 3: Pomiary czasu importu danych

Liczba rekordów	Czas wykonania (ms)			
	Postgres	Oracle	MySQL	SQL Server
1	1386,2	3227,8	2689,4	1018,4
500	1292,8	3417,0	2680,2	985,4
1 000	1240,0	3837,6	2721,0	1017,8
5 000	1238,0	3447,2	3036,4	1026,8
10 000	1464,6	3621,2	3065,6	1062,2
50 000	2125,6	3629,0	4245,4	899,0
100 000	2892,2	4245,4	6147,0	1064,2
200 000	4785,6	4656,2	9618,8	1120,8
500 000	10098,0	6255,4	19922,2	2128,0
1 000 000	18633,0	8470,4	35017,0	4028,2
2 000 000	36384,6	14118,4	67672,4	6760,2
5 000 000	87881,8	33458,4	166689,4	15987,4
10 000 000	186928,0	69167,4	332796,0	30759,8

Testy zostały przeprowadzone w zakresie od 1 do 10 000 000 rekordów, analizując otrzymane wyniki można jednoznacznie wskazać system, który najlepiej poradził sobie z odtwarzaniem bazy danych z kopii zapasowej. W każdym z przypadków czas importu bazy danych najkrótszy był dla SQL Server 2019. Warto zwrócić uwagę, że system ten nie korzysta z oddzielnego narzędzia do importu, ale wykonuje go poprzez składnię wbudowaną w język Transact-SQL rozwijany wraz z systemem bazodanowym. Drugi pod względem szybkości importowania jest system Oracle. Analizując rezultaty należy zwrócić uwagę na względne różnice pomiędzy systemami, w przypadku systemu Oracle czas odtworzenia bazy danych jest średnio dwa razy dłuższy



Rysunek 5: Porównanie czasu importu danych

niż czas analogicznej operacji wykonanej z wykorzystaniem SQL Server.

W tabeli 4 tabeli porównany został rozmiar wyeksportowanego pliku z daną liczbą rekordów. Na podstawie przedstawionych danych można stwierdzić, że najwyższy współczynnik kompresji danych występuje w przypadku systemu MySQL. Rozmiar pliku wygenerowanego z bazy Oracle miał dwukrotnie większy rozmiar niż analogiczny plik utworzony z wykorzystaniem systemu bazodanowego MySQL. Wysoki stopień kompresji może być też powodem wolnego odtwarzania bazy danych, w przypadku MySQL potwierdza to wykres na rysunku 6.

Tabela 4: Rozmiar odtwarzanej bazy danych

Liczba rekordów	Rozmiar (kB)			
	Postgres	Oracle	MySQL	SQL Server
1	1266	1335	542	7325
500	1294	1366	557	7325
1 000	1320	1398	571	7451
5 000	1540	1646	687	7581
10 000	1812	2273	831	7837
50 000	4000	4488	1987	11357
100 000	6734	7651	3431	14429
200 000	12202	14086	6328	19549
500 000	28604	33347	15138	35937
1 000 000	54958	65467	29498	62569
2 000 000	109636	130684	58559	116853
5 000 000	273668	326336	145741	278658
10 000 000	547056	652422	291035	547985

3.4. Wyszukiwanie danych na podstawie wzorca

Scenariusz testowy polegał na selekcji rekordów z tabeli na podstawie podanego wzorca. Wyszukiwanie odbywa się z wykorzystaniem trzyliterowej frazy, która może wystąpić na dowolnej pozycji. Test został przeprowadzony poprzez wyszukiwanie osób, które w nazwisku mają frazę PRO. Wykonanie takiego zapytania pozwala wyszukać osoby bez konieczności wprowadzenia pełnego nazwiska. Zapytanie wykorzystane do wyszukania osób z podaną frazą widoczne jest na listingu 3

Listing 3: Zapytanie wykorzystane do wyszukiwania danych na podstawie wzorca

```
SELECT imie, nazwisko, miasto, ulica, nr_domu,
nr_mieszkania, FROM szczegoly_uzytkownika su
WHERE nazwisko LIKE '%PRO%';
```

Szczegółowe wyniki testów przedstawione są w tabeli 5. Test wykonano dla czternastu różnych przypadków. W każdym z nich w tabeli z której wybierane są dane znajduje się określona liczba rekordów. Pierwszy scenariusz wykonano, gdy w tabeli znajdował się jeden rekord, w kolejnych testach ich liczba była odpowiednio zwiększana. W ostatnim przypadku test wykonany został dla 15 000 000 rekordów. Najkrótszy czas odpowiedzi uzyskany został gdy zapytanie było przetwarzane z wykorzystaniem systemu Oracle, drugi z komercyjnych systemów (SQL Server) osiągnął znacznie

gorsze wyniki przegrywając w niektórych próbach z MySQL.

Tabela 5: Wyszukiwanie na podstawie wzorca

Liczba rekordów	Czas wykonania (ms)			
	Postgres	Oracle	MySQL	SQL Server
1	6,0	1,8	3,0	6,2
1 000	1,0	4,0	5,4	9,4
5 000	3,0	4,2	6,4	12,4
10 000	6,2	9,4	13,4	18,8
20 000	9,4	6,2	20,2	43,8
50 000	21,8	9,4	46,4	96,8
100 000	37,4	24,8	73,4	165,6
200 000	75,4	43,6	144,6	284,4
500 000	161,6	112,4	320,8	690,6
1 000 000	306,4	202,6	639,2	1302,8
2 000 000	594,8	384,2	1614,6	1507,6
5 000 000	1496,0	914,8	5034,4	3215,2
10 000 000	3057,4	1880,0	7686,8	6710,0
15 000 000	4337,6	2856,4	10966,4	9490,8

Czas wykonania operacji wyszukiwania na podstawie wzorca jest dłuższy niż czas analogicznej operacji wykonanej z użyciem operatora równości. Procentowy przyrost czasu wykonania zapytania przedstawiony jest w tabeli 6. Obliczony został on dla każdego systemu zarządzania bazą danych ze wzoru

$$x_i = \left(\frac{t1_i}{t2_i} * 100\% \right) - 100\% \quad (2)$$

gdzie:

i – liczba rekordów w tabeli

x_i – procentowy przyrost czasu dla i -tej liczby rekordów w tabeli

$t1_i$ – czas wykonania zapytania wyszukującego dane na podstawie wzorca dla i -tej liczby rekordów

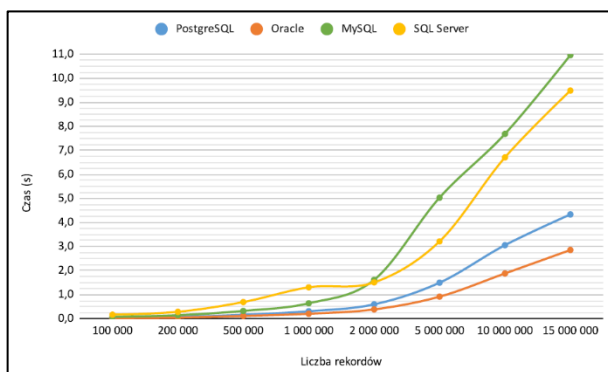
$t2_i$ – czas wykonania zapytania wyszukującego dane dla i -tej liczby rekordów

Tabela 6: Przyrost czasu wykonania zapytania wyszukiwania na podstawie wzorca w stosunku do wyszukiwania z wykorzystaniem operatora równości

Liczba rekordów	Procentowy przyrost czasu			
	Postgres	Oracle	MySQL	SQL Server
1	100,00%	-25,00%	0,00%	106,67%
1 000	-71,88%	5,26%	-15,63%	56,67%
5 000	0,00%	23,53%	3,23%	37,78%
10 000	-3,13%	213,33%	-2,90%	100,00%
20 000	0,00%	106,67%	50,75%	253,23%
50 000	17,20%	51,61%	44,10%	340,00%
100 000	33,57%	163,83%	11,21%	487,23%
200 000	66,81%	246,03%	15,87%	436,60%
500 000	36,26%	260,26%	16,99%	438,69%
1 000 000	35,70%	306,83%	27,08%	456,28%
2 000 000	30,27%	242,42%	18,34%	252,24%
5 000 000	23,07%	276,15%	20,43%	539,46%
10 000 000	27,83%	343,19%	28,59%	606,46%
15 000 000	17,23%	318,95%	22,64%	473,60%

Na podstawie danych z tabeli 6 można wywnioskować, że zmiana typu wyszukiwania wywołuje duży przyrost czasu wykonania zapytania w dwóch systemach komercyjnych: SQL Server i Oracle Database. Waha się on w zależności od liczby rekordów w tabeli. Jednak patrząc całościowo jest to wzrost o kilkaset procent. Odmienna sytuacja wystąpiła w przypadku systemów na licencji open source. Przyrost czasu wykonania zapytania wynosił w nich od kilkunastu do kilkudziesięciu procent.

Analizując bezpośrednie wyniki scenariusza testowego (rysunek 7), najlepsze czasy wyszukiwania na podstawie wzorca osiągnięte zostały przez system Oracle Database 19c. Duży przyrost czasu wykonania zapytania dla bazy danych SQL Server spowodował, że podczas tego scenariusza wydajnościowo uplasowała się ona na trzecim miejscu, znacząco odstając od Oracle i PostgreSQL.



Rysunek 7: Wyszukiwanie na podstawie wzorca - wykres

4. Wnioski

Przeprowadzone badania wykazały różnice w wydajności obsługi relacyjnych baz danych pomiędzy badanymi systemami. W testach najlepsze rezultaty osiągnęły systemy komercyjne czyli Oracle Database 19c który okazał się najlepszy w większości testów oraz SQL Server 2019. Dwa ostatnie miejsca przypadły dla PostgreSQL 12 oraz MySQL 8, który w zdecydowanej większości prób odnotował ostatni rezultat. Pomimo uzyskania niekonkurencyjnych rezultatów w odniesieniu do systemów komercyjnych, pojawiły się również pozytywne akcenty. MySQL uzyskał najlepszy wynik w teście dotyczącym utworzenia kopii zapasowej całej bazy danych, która zajmowała najmniej miejsca, natomiast PostgreSQL okazał się lepszy od SQL Server w teście wyszukiwania na podstawie wzorca.

Literatura

- [1] A. Pelikant, Bazy danych. Pierwsze starcie, Helion, 2012.
- [2] C.J. Date, Database Design and Relational Theory, O'Reilly, 2012.
- [3] B. Pękala, Bazy danych. Teoria i praktyka, Wydawnictwo Uniwersytetu Rzeszowskiego, 2015.
- [4] L. Rockoff, Język SQL. Przyjazny podręcznik, Helion 2017.
- [5] S. Feuerstein, B. Pribyl, Oracle PL/SQL Programming, O'Reilly, 2014.
- [6] Oficjalna dokumentacja SQL Server 2019 <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>, [05.2020].
- [7] B. Peasland, Oracle DBA Mentor: Succeeding as an Oracle Database Administrator, Apress, 2019.
- [8] L. Davidson, J. Moss, Pro SQL Server Relational Database Design and Implementation, Apress, 2016.
- [9] C. Mehta, A. Bhavsar, H. Oza, S. Shah, MySQL 8 Administrator's Guide: Effective guide to administering high-performance MySQL 8 solutions, Packt Publishing, 2018.
- [10] Ranking systemów zarządzania bazą danych <https://db-engines.com/en/ranking>, [08.2020].
- [11] Oficjalna dokumentacja bazy danych PostgreSQL <https://www.postgresql.org/files/documentation/pdf/12/postgresql-12-A4.pdf>, [02.2020].
- [12] R. West, SQL Server 2019 Administration Inside Out, 2020.
- [13] A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts, McGraw-Hill Higher Education, 2019.
- [14] W. Khadzhynov, P. Ratuszniak, Wprowadzenie do systemów baz danych (Wydanie VII), Helion, 2016.
- [15] M. Winand, SQL Performance Explained Everything Developers Need to Know about SQL Performance, 2012.
- [16] Dokumentacja narzędzia wykorzystanego do pomiarów <https://github.com/dbeaver/dbeaver/wiki>, [07.2020].
- [17] S. Riggs, G. Ciolli, S.K. Meesala, PostgreSQL 11 Administration Cookbook, Packt Publishing, 2019.
- [18] M. Leach, T. Lahdenmäki, Database Systems: Design, Implementation & Management 12th Edition, Cengage Learning, 2016.