

# Analiza możliwości zarządzania bazą danych przechowywaną w chmurze

Rafał Góźdz<sup>\*</sup>, Maria Skublewska-Paszkowska

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

**Streszczenie.** W artykule przedstawiono porównawczą analizę wydajnościową wybranych systemów baz danych. Porównano czasy wykonania zapytań SQL przez przygotowane bazy danych działające w usłudze chmurowej. Przygotowana baza danych Northwind działająca na SQL Server została przeniesiona na systemy Oracle i PostgreSQL, a następnie trzy wymienione bazy zostały umieszczone w chmurze Microsoft Azure. Proces badawczy został podzielony na trzy etapy. W pierwszej kolejności analizie podlegają zapytania typu Select. W drugim etapie analiza skupia się na kwestii wydajności zapytań SQL DDL, a końcowy etap opisuje analizę zapytań SQL DML.

**Słowa kluczowe:** usługi chmurowe; baza danych; analiza wydajnościowa

<sup>\*</sup>Autor do korespondencji.

Adres e-mail: rafal.gozdz@pollub.edu.pl

## Analysis of the possibility of managing the database stored in the cloud

Rafał Góźdz<sup>\*</sup>, Maria Skublewska-Paszkowska

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

**Abstract.** The article presents a comparative analysis of the performance of selected database systems. The time of execution of SQL statements has been compared by prepared database service which are running in the cloud. Prepared Northwind database running on SQL Server has been moved to systems Oracle and PostgreSQL, and then three of these bases have been placed in the cloud Microsoft Azure. The research process was divided into three stages. Firstly, queries with Select are analyzed. In the second stage, the analysis focuses on the performance of SQL DDL, and the final stage of the analysis describes the SQL DML.

**Keywords:** cloud computing; database; performance analysis

<sup>\*</sup>Corresponding author.

E-mail address: rafal.gozdz @pollub.edu.pl

### 1. Wstęp

Popularność usług chmurowych w ciągu ostatnich lat znacząco wzrosła [1]. Obecnie rynek informatyczny oferuje szereg zróżnicowanych usług chmurowych, które dają możliwość wykonywania podstawowych operacji takich jak: przechowywanie danych, tworzenie kopii zapasowych i innych o określonej specjalizacji. Pojęcie chmury w tematyce IT można rozumieć na kilka sposobów: jako chmura obliczeniowa (ang. Cloud computing) oraz jako dysk do przechowywania danych z możliwością przeglądania za pomocą urządzeń z dostępem do sieci. Funkcjonuje ona jako platforma, na której działają aplikacje, systemy informatyczne, maszyny wirtualne oraz systemy bazodanowe. Dostęp odbywa się za pomocą komputera lub urządzenia mobilnego z dowolnego miejsca, a ządania są przetwarzane po stronie dostawcy usługi.

Rosnąca popularność rozwiązań chmurowych wynika z możliwości dostępu do plików i usług z dowolnego komputera na świecie bez konieczności posiadania dodatkowych urządzeń. Kolejną zaletą jest możliwość wykupienia zasobu obliczeniowego tylko na okres, w którym będzie on potrzebny. Jest to atrakcyjne dla firm, które nie chcą inwestować w drogi sprzęt oraz infrastrukturę, a wolą skoncentrować się na rozwoju biznesu [3].

### 2. Charakterystyka chmury obliczeniowej

Chmura obliczeniowa stanowi pojęcie o wielu znaczeniach, dla jednych jest to dostarczenie usług obliczeniowych przez sieć, dla innych jest to inna definicja Internetu. Według NIST (National Institute of Standards and Technology) chmura obliczeniowa jest modelem umożliwiającym powszechny i wygodny dostęp do wspólnej puli konfiguracji zasobów obliczeniowych za pośrednictwem sieci. Przykładem takich zasobów są: sieci, serwery, pamięci masowe, aplikacje oraz usługi, które mogą być uruchomione w sposób zabezpieczony i z minimalnym obciążeniem po stronie usługodawcy [1].

Model chmury posiada pięć zasadniczych cech [2]:

- Samoobsługa na żądanie – klient ma możliwość jednostronnego zarządzania parametrami usług takimi jak: czas serwera, sieciowa pamięć masowa.
- Szeroki dostęp do sieci – możliwość dostępu do sieci poprzez standardowe mechanizmy (np. telefony komórkowe, tablety, laptopy i stacje robocze).
- Łączenie zasobów – zasoby usługodawcy są łączone, aby służyć wielu konsumentom. Istnieje możliwość dynamicznego przypisywania różnych zasobów fizycznych i wirtualnych w zależności od potrzeb klienta.

- Wysoka elastyczność – szybka skalowalność, dostosowywanie możliwości zasobów w dowolnym momencie może być dla klienta wręcz nieograniczone.
- Miarowość usług – systemy chmurowe automatycznie sterują i optymalizują wykorzystanie zasobów na pewnym poziomie abstrakcji w odpowiedni sposób dla rodzaju usługi (np. przechowywanie i przetwarzanie danych, przepustowość). Wykorzystanie zasobów jest monitorowane, aby zapewnić kontrolę i przejrzystość zarówno dla dostawcy jak i klientów usług.

Usługi chmurowe można podzielić ze względu na model świadczonych usług [3]:

- Software as Service (SaaS) – oprogramowanie jest świadczone jako usługa, której zarządzanie odbywa się po stronie dostawcy usługi. Jest to najbardziej popularna metoda świadczenia usług chmurowych po stronie klienta. Takie usługi przyczyniają się do zmniejszenia kosztów eksploatacji oprogramowania poprzez pominięcie procesu instalacji i potrzeby aktualizacji. Przykładami takich aplikacji jest: DropBox, Google Drive do przechowywania danych oraz aplikacje biurowe takie jak: Office online, czy Google Docs.
- Platform as a Service (PaaS) – platforma stanowi usługę, na której oprogramowanie jest tworzone i wdrażane. Dostawca platformy udostępnia serwer wraz z systemem operacyjnym oraz zapewnia wysoką jakość infrastruktury sieciowej. Przykładami usługi PaaS są: Microsoft Azure, Google Aps i Heroku.
- Infrastructure as a Service (IaaS) – ten rodzaj usługi składa się z zautomatyzowanych i wysoce skalowalnych jednostek obliczeniowych, przy użyciu których konsument jest w stanie wdrożyć i uruchomić dowolne oprogramowanie. Użytkownicy mają większą kontrolę nad serwerem niż w przypadku modelu PaaS. Poza kontrolą nad systemem operacyjnym użytkownik ma możliwość zarządzania całym serwerem i pamięcią masową. Jest to najbardziej elastyczny model chmury obliczeniowej ze względu na wysoką skalowalność, możliwość automatycznego wdrażania serwerów, dostęp do mocy obliczeniowej, danych i infrastruktury sieciowej. Przykładami modelu IaaS są dostawcy usług: HPCloud i CloudSigma.

### 3. Bazy danych

#### 3.1. PostgreSQL

PostgreSQL jest obiektowo-relacyjnym systemem zarządzania bazami danych opracowanym przez Uniwersytet Kalifornijski w Berkeley [4]. Jest jednym z najpopularniejszych systemów bazodanowych ze względu na wieloplatformowość oraz darmową licencję open source. System pozwala na tworzenie obszernych baz danych, przetwarzanie żądań nadesłanych przez inne aplikacje. System może obsługiwać obciążenia pochodzące od małych aplikacji, jak i od rozbudowanych aplikacji internetowych obsługujących równocześnie wielu użytkowników. PostgreSQL działa na wielu systemach operacyjnych: Linux, Windows oraz macOS oraz wspiera najpopularniejsze architektury sprzętowe: x86, x86-64 i ARM. Baza danych

Postgres implementuje standard SQL:2011, obsługuje procedury składowane w języku PL/pgSQL, indeksy, mechanizm wyzwalaczy, czy definiowanie reguł.

PostgreSQL jest oparty na architekturze klient-serwer. Sesja składa się z trzech współpracujących procesów [4]:

- procesu administratora – postmaster,
- aplikacji użytkownika – frontend,
- serweru bazy danych –backend.

Proces administratora zarządza klastrem baz danych na komputerze docelowym. Aplikacja kliencka, np. pgAdmin, otrzymuje dostęp do bazy danych w klastrze poprzez zapytania do biblioteki libpq. Biblioteka przekazuje żądania za pomocą sieci do postmastera, który tworzy nowy proces serwera i łączy go z procesem klienta. Po tym zdarzeniu komunikacja klient-serwer odbywa się bez udziału procesu administratora. Postmaster jest procesem stale działającym i oczekującym na żądania. Inaczej jest z procesami frontend i backend, które mają określone cykle życia. Pojedynczy klient ma możliwość nawiązania wielu połączeń z procesem serwera, który z postmasterem działa na tym samym serwerze, podczas gdy klient może nawiązać połączenie z innego miejsca. Dostęp do bazy danych odbywa się za pomocą logowania użytkowników. Każdy użytkownik posiada swoją unikalną nazwę, identyfikator (userid) oraz przypisane uprawnienia pozwalające na dostęp i modyfikację obiektów bazodanowych. W PostgreSQL zarządzanie uprawnieniami jest zrealizowane poprzez role. Użytkownik uprzywilejowany ze wszystkimi uprawnieniami to postgres.

#### 3.2. Oracle

Oracle to obiektowo-relacyjny system do zarządzania bazami danych opracowany przez firmę o takiej samej nazwie. Do obsługi bazy danych zastosowano standardowy język SQL oraz do tworzenia procedur składowanych język PL/SQL i Java (od wersji 8). Przechowywanie danych w systemie Oracle jest oparte na wykorzystaniu pamięci współdzielonej dostępnej dla wszystkich użytkowników bazy danych określanej jako Globalny Obszar Współdzielony – SGA. Instancja Oracle składa się z zablokowanego globalnego obszaru współdzielonego SGA i procesów tła, których celem jest minimalizacja ruchu informacji przy zachowaniu bezpieczeństwa utrzymania danych. Polecenia SQL trafiają do odpowiedniego bufora SGA, po przetworzeniu i przeanalizowaniu bloki danych są pobierane do obszaru SGA, a następnie zwracane użytkownikowi. Jeżeli wystosowane polecenie SQL odwoła się do danych znajdujących się już w obszarze SGA to zostaje pominięty etap pobierania danych i wynik zostanie zwrócony szybciej.

Architektura oparta na klient-serwer w systemie Oracle pozwala na równoczesny dostęp użytkowników do tej samej bazy danych, operacje odczytów danych przez wielu użytkowników nie powodują konfliktów. Podczas operacji modyfikacji mogą występować konflikty i niespójności, dlatego wprowadzono mechanizm transakcji. W przypadku konfliktu system bazodanowy szereguje operacje transakcji tak, aby nie powstawały konflikty. Użytkownik logujący się do systemu zarządzania bazy danych rozpoczyna tym samym

sesje i kończy w momencie zamknięcia systemu. W danej sesji może być realizowanych wiele transakcji, jedna po drugiej opisywanych za pomocą polecenia SQL. Użytkownik tworząc obiekty bazodanowe staje się ich właścicielem. Miejszem docelowym obiektów jest schemat użytkownika, który jest tworzony automatycznie podczas dodawania nowego użytkownika, posiadający unikalną nazwę oraz będącym logiczną przestrzenią bazy danych [5]. W systemie Oracle istnieje wiele obiektów służących do przechowywania danych oraz do wspomaganie zarządzania utworzonych danych. Tworzone obiekty przez użytkowników wymagają posiadania przez nich określonych uprawnień. System Oracle obsługuje obiekty takie jak: tabele, indeksy, klastry, widoki, sekwencje, wyzwalacze oraz pakiety, procedury i funkcje.

### 3.3. SQL Server

Microsoft SQL Server (MS SQL) to system zarządzania bazą danych opracowany i rozwijany przez firmę Microsoft. Jest to platforma typu klient-serwer, w której jedna instancja serwera przechowuje wiele baz danych. Bazy danych można podzielić na systemowe i użytkownika [6]. Wielu użytkowników systemu może posiadać uprawnienia do jednej bazy danych, referencja do takiego obiektu zapisana będzie w postaci: BazaDanych.uzytkownik.obiekt.

Domyślnym schematem i jednocześnie użytkownikiem posiadającym pełne uprawnienia do bazy danych jest dbo. Dane znajdujące się w bazie są przechowywane w postaci plików, które są odpowiednikiem fizycznych plików na dysku. Podczas tworzenia bazy danych tworzony jest plik główny dla danych będący miejscem startowym bazy danych o rozszerzeniu .mdf oraz pozostałe pliki nazywające się secondaryfile i mające rozszerzenie .ndf. Trzecim typem plików są pliki dziennika transakcji zawierające informacje niezbędne do odtworzenia stanu bazy danych po awarii, mające rozszerzeniem .ldf i nienależące do żadnej grupy plików. Baza może składać się z wielu plików danych i dziennika transakcji, które są zarządzane w inny sposób niż pliki danych. Utworzone pliki są wykorzystywane tylko przez jedną bazę danych. Takie rozwiązanie umożliwia umieszczenie plików na wielu dyskach fizycznych, poprawiając w ten sposób wydajność serwera. Referencje do wszystkich plików w bazie są składowane w pliku głównym (.mdf). System odczytuje plik główny bazy danych w sytuacji aktualizacji lub odtwarzania stanu bazy danych, a dostęp odbywa się za pomocą odwołania do nazwy logicznej, jak i fizycznej. Nazwę logiczną używa się w instrukcjach T-SQL, przy czym nazwa i identyfikator muszą być unikalne w całej bazie. Nazwa fizyczna pliku jest to nazwa pliku w systemie operacyjnym z rozszerzeniem mdf. Pliki bazy danych mdf oraz ndf składają się z ponumerowanych stron, gdzie numeracja rozpoczyna się od 0. Pierwsza strona plików zawiera nagłówki informujące o parametrach pliku. Każdy plik ma swój unikalny identyfikator. Odwołując się do konkretnej strony w pliku należy podać identyfikator pliku oraz numer strony. Rozmiary opisywanych plików rosną samoczynnie, istnieje możliwość określenie przyrostu rozmiaru pliku po jego wypełnieniu. Jeżeli plik należy do grupy plików, przyrost rozmiaru pliku może nastąpić jedynie po wypełnieniu wszystkich plików w grupie. Pliki bazy danych

należące do grupy plików tworzą tzw. przestrzeń nazw znana z innych systemów baz danych np. Oracle. Obiekty bazodanowe rozmieszczane są w utworzonych grupach plików, które można podzielić na typy [6]:

- główny – obejmuje główny plik bazy danych i wszystkie inne pliki nieprzypisane do innej grupy plików,
- zdefiniowane przez użytkownika.

Istnieje również domyślna grupa plików, do której serwer przypisuje pliki, które nie mają określonej grupy plików.

Na serwerze SQL Server znajduje się baza danych tempdb, która jest buforową bazą, przechowywującą tymczasowo dane z aktualnie przetwarzanych transakcji, w której wyniki zapytań są obrabiane pod zadanymi kryteriami przed wysłaniem odpowiedzi. Zalecane jest umieszczenie bazy tempdb na innym dysku fizycznym celem zwiększenia wydajności serwera [6].

### 4. Metoda badawcza

Do przeprowadzenia badań użyto relacyjną bazę danych Northwind przeznaczoną na platformę SQL Server. Baza zawiera przykładowe dane na temat sprzedaży określonych produktów dostarczonych przez odpowiednich dostawców. W celu zaimportowania bazy danych należy przenieść plik northwind.mdf do katalogu \Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA. Następnie w środowisku SQL Server Management Studio z menu kontekstowego Databases należy wybrać Attach Databases. Importowanie bazy odbywa się za pomocą konsoli z użyciem pliku o rozszerzeniu sql. Po utworzeniu pustej bazy należy wpisać przykładowe polecenie: sqlcmd -S host -d Northwind -i sciezka\_do\_bazy. Wykonanie komendy spowoduje utworzenie kompletnego obiektu bazodanowego, a w sekcji Object Explorer pojawi się struktura zaimportowanej bazy Northwind w postaci drzewa katalogów [6]. Kolejnym etapem jest przeniesienie bazy danych Northwind na systemy Oracle i PostgreSQL. Na początku należy utworzyć pustą strukturę danych oddzielnie na każdym z systemów. Podczas tworzenia należy wziąć pod uwagę, że nie każdy typ danych występuje na wszystkich trzech systemach baz danych. W takim wypadku należy wybrać typ i jego rozmiar, który będzie kompatybilny z typem bazy SQL Server. Następnie należy przeprowadzić proces migracji danych, który określa sposób przeniesienia danych z miejsca źródłowego do miejsca docelowego. Dla realizacji celów pracy dane zostały przeniesione z tabel bazy źródłowej w systemie SQL Server do systemów Oracle i PostgreSQL.

Do migracji danych na gotowych strukturach baz danych wykorzystano Pentaho Data Integration w wersji próbnej [7]. W pierwszej fazie migracji należy utworzyć trzy połączenia do baz danych: jedno do bazy źródłowej oraz dwa do baz, do których dane będą przenoszone. W kolejnym etapie należy utworzyć indeksy, które w każdym systemie baz danych są tworzone w podobny sposób. Po utworzeniu gotowych skryptów i wykonaniu ich za pośrednictwem narzędzi PgAdmin i SQL developer połączonych z serwerami bazodanowymi, zostaną utworzone puste struktury bazy danych Northwind na systemach PostgreSQL i Oracle.

Kolejną czynnością jest zdefiniowanie zbioru wejściowego dla określonej tabeli, poprzez dodanie z zakładki Input elementu Table Input. W ustawieniach należy wybrać połączenie z źródłową bazą danych MS SQL oraz wybrać dane z określonej tabeli za pomocą polecenia SQL. Alternatywnie można wybrać opcję Get SQL Statement, które automatycznie wprowadzi w pole tekstowe polecenia wybierające wszystkie rekordy z tabeli. Wybrane dane będą importowane do bazy docelowej, dlatego należy zwrócić uwagę, czy wszystkie typy i formaty danych są kompatybilne. Po ustawieniu Table Input dla określonej tabeli należy dołączyć do niej obiekt, do którego dane zostaną zaimportowane. W tym przypadku będzie to także tabela, ale innej bazy danych (Oracle albo PostgreSQL), dlatego zostaje dołączony Table output. W ustawieniach należy wybrać połączenie bazy, do której dane będą importowane oraz tabele. Istnieje także możliwość ustawienia mapowania metadanych kolumn. W tym przypadku mapowanie odbywa się automatycznie ze względu na spójność nazw. Po utworzeniu takiego przepływu należy powtórzyć operację dla każdej tabeli w bazie danych, a następnie uruchomić proces migracji danych. Należy pamiętać, że w czasie tego procesu tabele nie mogą być powiązane ze sobą, ponieważ może wystąpić wyjątek naruszenia unikalności klucza obcego. Po sfinalizowaniu procesu tabele obu baz danych powinny zawierać identyczną ilość danych. Następnie należy dokonać konfiguracji platformy Azure. Użyte konto jest w wersji próbnej 30 dniowej, w której użytkownik ma do wykorzystania 170 euro na zasoby obliczeniowe.

Do umieszczenia baz danych w systemach MS SQL, PostgreSQL i Oracle użyto odpowiednio:

- Azure SQL Server,
- Oracle Database 12.1.0.2 Standard Edition,
- PostgreSQL 9.5 działającym na Ubuntu 14.04.

Azure SQL Database jest wbudowaną usługą relacyjnych baz danych w chmurze Azure opartej na Microsoft SQL Server. Usługa SQL w chmurze współpracuje z narzędziami, bibliotekami i API SQL Server umożliwiając w ten sposób prostsze przeniesienie rozwiązań do chmury. Przykładem jest możliwość przeniesienia bazy danych za pomocą narzędzia Management Studio do SQL Azure przy użyciu opcji Deploy Database to SQL Azure [8].

Do utworzenia pozostałych systemów baz danych w usłudze Azure użyto maszyn wirtualnych będących rozwiązaniem typowo infrastrukturalnym opartym na modelu IaaS, w którym aplikacje, dane, środowisko jak i system są zarządzane przez użytkownika. Microsoft zapewnia różne warianty platformy sprzętowej takich jak. moc obliczeniowa CPU, ilość pamięci RAM oraz przestrzeń dyskowa, które są wybierane przez klienta podczas tworzenia maszyny wirtualnej. Płatność odbywa się za czas rezerwacji zasobów sprzętowych od momentu uruchomienia maszyn wirtualny z systemami Windows oraz Linux.

Dla potrzeb badań utworzono dwie maszyny wirtualne [9]:

- Standardowa DS12 v2 (4 rdzenie, 28 GB pamięci) z systemem operacyjnym Ubuntu oraz zainstalowanym PostgreSQL 9.5,
- Standardowa DS12 v2 (4 rdzenie, 28 GB pamięci) opartym na systemie Linux z zaimplementowanym Oracle Database 12.1.0.2 Standard Edition.

## 5. Badania

Badania zostały przeprowadzone dla trzech baz danych: SQL Server, Oracle i PostgreSQL, działających w ramach usługi Microsoft Azure. Dane zostały przedstawione w formie tabel oraz wykresów słupkowych, mając na celu utrzymanie przejrzystości rezultatów [9]. Kryterium badawczym jest czas wykonania zapytań przedstawionych na listingach 1, 2 i 3 mierzony w milisekundach [ms] dla poszczególnych systemów baz danych: SQL Server, PostgreSQL i Oracle. Wykonywanie zapytań zostało powtórzone 10-krotnie z wyłączoną funkcją cachowania, aby wyniki były miarodajne. Po wykonaniu zapytań, które dokonały modyfikacji struktury bazy danych, konieczne było przywrócenie jej w procesie backupu. Przykład 1 zawiera zapytania dla kategorii DML.

Przykład 1. Zapytania Select

```
--1. Skrypt tworzący widok zamówień wszystkich klientów.  
Create View VCustomersOrders AS  
Select c.CustomerID , c.CompanyName, c.ContactName,  
c.Address, c.City, c.PostalCode, c.Country, o.OrderID,  
o.OrderDate, o.ShippedDate, o.ShipAddress,o.ShipCity,  
o.ShipCountry  
FROM [Customers] c Left JOIN [Orders] o ON  
c.CustomerID=o.CustomerID;
```

```
--2. Wybranie danych o kliencie oraz ilości jego zamówień  
Select c.CustomerID, c.CompanyName ,c.Address, c.City,  
c.Country, Count(o.OrderID) AS "Order quantity"FROM  
[Customers] c Left JOIN [Orders] o ON  
c.CustomerID=o.CustomerID GROUP BY c.CustomerID,  
c.CompanyName ,c.Address, c.City, c.Country;
```

```
--3. Wybranie dane o produkcie i ilości jego zamówień w  
poszczególnych transakcjach  
Select p.ProductID, p.ProductName, o.OrderID, o.Quantity  
FROM [Products] p LEFT JOIN [Order Details] o ON  
p.ProductID=o.ProductID Order by p.ProductID, Quantity;
```

```
--4. Wybranie danych o dostawcach i ich klientach  
Select c.CustomerID, c.CompanyName , s.SupplierID,  
s.CompanyName From Customers c Left Join Orders o ON  
c.CustomerID=o.CustomerID Left Join [Order Details] od ON  
o.OrderID=od.OrderID Left Join Products p ON od.ProductID  
=p.ProductID Left Join Suppliers s ON  
p.SupplierID=s.SupplierID ORDER BY 1,3;
```

```
--5. Zapytanie wybierające dane o ilości zamówień  
zrealizowanych w poszczególnych regionach przez  
pracowników.  
Create View vOrderEmployees AS  
Select e.EmployeeID,e.LastName, e.FirstName,  
Count(o.OrderID) AS 'Orders quantity',  
t.TerritoryDescription,r.RegionDescription FROM Orders o  
Left Join Employees e ON o.EmployeeID=e.EmployeeID  
Left Join EmployeeTerritories et on  
e.EmployeeID=et.EmployeeID  
Left Join Territories t On et.TerritoryID=t.TerritoryID  
Left Join Region r ON t.RegionID=r.RegionID
```

Group BY e.EmployeeID, e.LastName, e.FirstName,  
t.TerritoryDescription, r.RegionDescription;

W przykładzie 2 zebrano zapytania z kategorii SQL DDL.

Przykład 2. Zapytania SQL DDL

```
--6. Skrypt dodający do tabeli Employee kolumny idRating
jako powiązanie z tabelą EmployeeRating
Alter Table Employees add RatingID INT;
```

```
-- 7. Skrypt dodający tabelę Rating
Create Table Rating (
    RatingID INT PRIMARY KEY NOT
    NULL,
    RatingNumber samllint,
    ratingDescribtion Varchar(255),
    superEmployee boolean
);
```

```
Alter Table Employees add Constraint RatingIDFK FOREIGN
KEY(RatingID) REFERENCES Rating(RatingID);
```

```
--8. Skrypt usuwający kolumnę discount z tabeli OrderDetails
Alter table [Order_Details] DROP Constraint
DF_Order_Details_Discount
Alter table [Order_Details] DROP Constraint CK_Discount
Alter Table [Order_Details] DROP column Discount
```

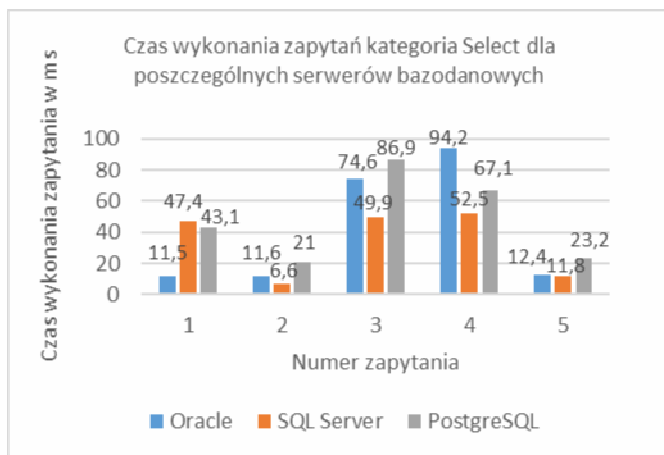
```
-- 9.Skrypt usuwający tabelę Contacts
Drop table [Contacts]
```

W Listingu 3. zebrano zapytania z zapytania łączone z poprzednich listingów, które przedstawiono w kolejnej części artykułu.

Przykład 3. Zapytania SQL DML

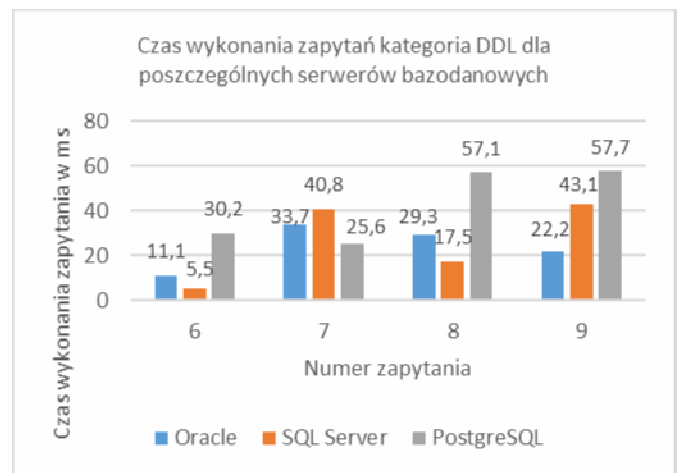
```
--10. Skrypt usuwający rekordy o szczegółach zamówień
wybranego produktu Delete FROM [northwind].[dbo].[Order
Details] od
inner join [northwind].[dbo].Products p ON
od.ProductID=p.ProductID --Order by ProductName
where p.ProductName='Gnocchi di nonna Alice'
```

```
--11. Skrypt aktualizujący dane o zniżkach produktów
Update Products set Discontinued=1 where UnitPrice>25;
Update [Order_Details] set Discount =0.50 where UnitPrice >
25;
```



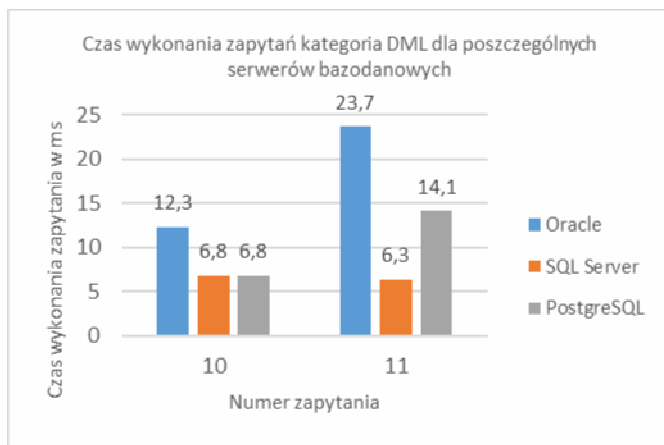
Rys. 1. Wykres czasów wykonania zapytań Select na wybranych serwerach baz danych

Rysunek 1 przedstawia wykres z rezultatami czasów wykonania zapytań Select. Zapytanie numer 1 zostało wykonane w podobnym czasie przez SQL Server i PostgreSQL (ponad 43 ms), jednak najkrótszy czas odnotowuje się dla bazy Oracle i wynosi 11,5 ms będący wynikiem 4 razy mniejszym od pozostałych baz danych. Drugie zapytanie z tej kategorii zostało najszybciej wykonane przez bazę Microsoftu w czasie 6,6 ms, prawie 2 krotnie szybciej niż Oracle i 3 krotnie szybciej niż baza PostgreSQL. Zapytanie 3 i 4 SQL Server ponownie wykonał najszybciej w czasie ok. 50 ms, przy czym w zapytaniu 3 PostgreSQL okazał się najwolniejszy (o 74 %), natomiast w zapytaniu 4 Oracle miał znacznie odbiegający czas wykonania wynoszący 94,2 ms stanowiący wynik o 81 % wolniejszy niż SQL Server i o 41 % wolniejszy niż baza PostgreSQL. Ostatnie zapytanie z tej kategorii zostało wykonane najszybciej przez SQL Server (11,8 ms) i Oracle (12,4 ms) co stanowi wynik o połowę mniejszy niż w przypadku PostgreSQL.



Rys. 2. Wykres czasów wykonania zapytań DDL na wybranych serwerach baz danych

Z wykresu na rysunku 2 można zaobserwować rozbieżność w czasach wykonaniu poszczególnych zapytań DDL dla różnych systemów baz danych. W przypadku zapytania 6 SQL Server wykonał je w czasie 5,5 ms, tj. ponad 5 krotnie szybciej niż PostgreSQL i 2 krotnie szybciej niż Oracle. Czasy wykonania zapytania numer 7 są już mniej rozbieżne od poprzedniego. Baza danych PostgreSQL wykonała zapytanie najszybciej, w czasie 25 ms, czyli o 40 % krótszym niż SQL Server i o 24% krótszym niż Oracle. W kolejnym zapytaniu można znowu zaobserwować dużą rozbieżność w czasie wykonania zapytania. Baza SQL Server wykonała je najszybciej, w czasie 17,5 ms – ponad 3 krotnie szybciej niż PostgreSQL i o 41 % szybciej niż Oracle. Zapytanie 9 zostało najszybciej zrealizowane przez bazę Oracle z czasem 22 ms – 2 krotnie szybciej niż SQL Server i 2,5 razy szybciej niż PostgreSQL.



Rys. 3. Wykres czasów wykonania zapytań DML na wybranych serwerach baz danych

Rysunek 3 przedstawia wykres z czasami wykonania zapytań dla obu kategorii DDL i DML. Zapytanie numer 10 zostało wykonane najszybciej przez dwie bazy danych: SQL Server i PostgreSQL w jednakowym czasie wynoszącym 6,8 ms, który jest o 44 % krótszy niż wynik pochodzący z bazy Oracle. W drugim zapytaniu z kategorii DML odnotowuje się kolejne rozbieżności w wynikach wykonania zapytania. Baza Microsoftu ponownie wykonała zapytanie najszybciej z czasem 6,3 ms, baza PostgreSQL wykonała je w czasie ponad 2 krotnie dłuższym, a baza Oracle w czasie ponad 3 krotnie dłuższym.

Analizując wyniki zgromadzone na wykresach można zaobserwować otrzymywanie mało-odbiegających wartości czasów dla bazy danych PostgreSQL. W przypadku bazy Oracle różnice pomiędzy czasami są nieznacznie większe podczas, gdy w bazie SQL Server te wahania są dużo większe.

## 6. Wnioski

Usługi chmurowe pozwalają w łatwy i szybki sposób konfigurować i uruchamiać serwery baz danych. Dostępny wybór zasobów obliczeniowych jest dostosowany do indywidualnych klientów jak i dużych firm informatycznych, które coraz częściej potrzebują ogromnej mocy obliczeniowej. Duże zainteresowanie wykorzystaniem chmury obliczeniowej powoduje ciągły jej rozwój i zwiększenie zapotrzebowania ze strony firm informatycznych, które coraz częściej inwestują w tę technologię. Na podstawie przeprowadzonych badań można stwierdzić, że baza danych SQL Server wspierana technologią Azure, pochodząca od tego samego producenta, osiągnęła najlepsze rezultaty. Czasy wykonywania wszystkich zapytań SQL DML i części DDL na bazie danych Microsoftu były nawet kilkukrotnie krótsze niż konkurencyjne bazy danych. We wszystkich badanych kategoriach zapytań występują rozbieżności w czasach ich wykonania. Największą rozbieżność odnotowano w zapytaniu nr 6, w którym najkrótszy czas wynosi 5,5 ms dla SQL Server, a najdłuższy 30,2 ms dla PostgreSQL. Najmniejszą rozbieżność wyników odnotowano w zapytaniu nr 7 i waha się ona w granicach od 25,6 ms dla PostgreSQL do 40,8 ms dla SQL Server. Powodem wyżej przedstawionych wyników jest wsparcie technologii Microsoftu oraz lepsza optymalizacja usługi niż

w przypadku tradycyjnej konfiguracji serwera bazodanowego na systemie Linux. Jednakże w zapytaniu dotyczącym tworzenia nowej tabeli oraz zapytaniu dotyczącym usuwania wybranych rekordów baza danych PostgreSQL uzyskuje najlepsze rezultaty w stosunku do pozostałych opisywanych powyżej baz danych. Bazy danych Oracle i PostgreSQL mimo, że zostały skonfigurowane na maszynach wirtualnych o identycznych parametrach osiągały różne czasy wykonania tych samych zapytań. Uzyskane wyniki mogą ułatwić wybór systemu bazodanowego w zależności od struktury bazy oraz od tego jakie operacje będą na niej wykonywane.

## Literatura

- [1] Peter Mell, Timothy Grance, The NIST Definition of Cloud Computing, <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublicatio n800-145.pdf>, [11.09.2016]
- [2] Daniel J. Abadi, Data Management in the Cloud: Limitations and Opportunity, strony 2-5, Yale University 2009\_
- [3] Ben Kepes, Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS, <https://support.rackspace.com/white-paper/understanding-the-cloud-computing-stack-saas-paas-iaas/>, [11.09.2016]
- [4] Scott Zhang, Install and configure PostgreSQL on Azure, <https://azure.microsoft.com/pl-pl/documentation/articles/virtual-machines-linux-postgresql-install/>, [11.09.2016]
- [5] Andrzej Klusiewicz, Kurs Oracle, <http://andrzejklusiewicz.blogspot.com/2010/11/kurs-oracle-sql-podstawowe-definicje.html>, [11.09.2016]
- [6] Lech Banchowski, Realizacja SZBD w SQL Server, <http://edu.pjwstk.edu.pl/wyklady/szb/scb/wyklad15/w15.htm>, [11.09.2016]
- [7] Pentaho Data Integration manual, <http://www.pentaho.pl/kettle-etl.html>, [11.09.2016]
- [8] T. Redkar, T. Guidici, Platforma Windows Azure, Helion 2013.
- [9] Rob Boucher, Introducing Microsoft Azure, <https://azure.microsoft.com/en-us/documentation/articles/fundamentals-introduction-to-azure/>, [11.09.2016]