

OPIS PRZEDMIOTU ZAMÓWIENIA

„Zakup licencji oprogramowania do posiadanego przez Zamawiającego systemu informatycznego opartego na oprogramowaniu Oracle”

1. Przedmiotem zamówienia jest zakup licencji oprogramowania w celu rozszerzenia funkcjonalności posiadanego przez Zamawiającego systemu informatycznego wykorzystującego Oracle Database Enterprise Edition 11g poprzez zakup dodatkowych licencji oraz zwiększenie licencji na posiadane przez Zamawiającego oprogramowanie umożliwiające zarządzanie i konfigurację baz danych, kompatybilne z posiadanym przez Zamawiającego systemem informatycznym wykorzystującym Oracle Database Enterprise Edition 11g.

Niezbędne licencje oprogramowania odpowiedniego rodzaju i odpowiedniej ilości konieczne dla rozszerzenia funkcjonalności posiadanego systemu informatycznego:

PN	Nazwa	obecnie posiadane	ilość do dokupienia
A90611	Oracle Database Enterprise Edition - Processor Perpetual	48	8
A90620	Oracle Partitioning - Processor Perpetual	32	8
L47225	Oracle Advanced Compression - Named User Plus Perpetual	0	150
A90619	Oracle Real Application Clusters - Processor Perpetual	48	8

2. Użyte określenia wskazujące znaki towarowe, nazwy własne, patent lub pochodzenie przedmiotu zamówienia należy odczytywać wraz z wyrazami „lub równoważne”. Zamawiający dopuszcza zastosowanie przez Wykonawcę rozwiązań równoważnych rozwiązaniom wskazanym w opisie przedmiotu zamówienia. Wykonawca, który w ofercie powoła się na stosowanie rozwiązań równoważnych obowiązany jest wykazać, że oferowane przez niego licencje oprogramowania i rozwiązania spełniają wymagania określone przez Zamawiającego.

W przypadku zaoferowania oprogramowania równoważnego do oprogramowania opisanego w pkt. 1, Wykonawca zobowiązany jest spełnić następujące wymagania:

3. Wykonawca oferując oprogramowanie równoważne do opisanego w pkt 1 jest zobowiązany wykazać równoważność w zakresie parametrów technicznych, użytkowych, funkcjonalnych i jakościowych, które muszą być na poziomie nie niższym od parametrów wskazanych przez Zamawiającego w pkt 4.
4. Minimalne wymagania dotyczące równoważności oferowanego oprogramowania:
- 1) Dostępność oprogramowania na współczesne 64-bitowe platformy Unix (HP-UX dla procesorów Itanium, Solaris dla procesorów SPARC i Intel/AMD, IBM AIX dla procesorów POWER, Intel/AMD Linux, MS Windows). Identyczna funkcjonalność serwera bazy danych na ww. platformach
 - 2) Dostępność narzędzi migracji baz danych pomiędzy platformami na poziomie fizycznym (kopiowanie / konwersja plików danych) oraz logicznym (narzędzia eksportu / importu).
 - 3) Oprogramowanie klienckie, za pomocą którego można łączyć się do bazy danych musi być dostępne na wielu platformach systemowo-sprzętowych (minimalny zakres platform taki jak dla oprogramowania serwera bazy danych)
 - 4) Wsparcie protokołu XA.
 - 5) Wsparcie standardu JDBC 3.0.

- 6) Zgodność ze standardem ANSI/ISO SQL 2008 lub nowszym.
- 7) Wbudowana obsługa wyrażeń regularnych zgodna ze standardem POSIX dostępna z poziomu języka SQL jak i procedur/funkcji składowanych w bazie danych.
- 8) RDBMS musi zapewniać niezależność platformy systemowej dla oprogramowania klienckiego od platformy systemowej bazy danych.
- 9) RDBMS musi zapewniać przetwarzanie transakcyjne wg reguł ACID z zachowaniem spójności i maksymalnego możliwego stopnia współbieżności. Mechanizm izolowania transakcji musi pozwalać na spójny odczyt modyfikowanego obszaru danych bez wprowadzania blokad, spójny odczyt nie może blokować możliwości wykonywania zmian.
- 10) RDBMS musi posiadać możliwość zagnieżdżenia transakcji – możliwość uruchomienia niezależnej transakcji wewnątrz transakcji nadrzędnej.
- 11) Dostępność nieblokującego poziomu izolowania transakcji „tylko do odczytu” (Read Only) pozwalający na uzyskanie w wielu kolejnych następujących po sobie zapytaniach rezultatów odzwierciedlających stan danych z chwili rozpoczęcia ww. transakcji.
- 12) Dostępność poziomu serializowanego poziomu izolowania transakcji (Serializable).
- 13) Możliwość zmiany domyślnego trybu izolowania transakcji (Read Committed) na inny (Read Only, Serializable) za pomocą komend serwera bazy danych.
- 14) Możliwość uruchamiania zapytań odwołujących się do stanu danych sprzed modyfikacji, których poprawność i kompletność jest gwarantowana przez serwer bazy danych i nie może podlegać możliwości manipulacji od strony użytkownika / administratora. Operowanie na ww. zasobie informacji powinno odbywać się za pomocą udokumentowanego języka zapytań serwera bazy danych lub jego wbudowanych rozszerzeń. Z technicznego punktu widzenia oczekuje się możliwości zapewnienia wykonalności ww. operacji sięgających do 24 godzin wstecz.
- 15) Wsparcie dla wielu ustawień narodowych i wielu zestawów znaków (włącznie z Unicode) zarówno po stronie serwera bazy danych jak i oprogramowania klienckiego. Wsparcie dla polskich stron kodowych – ISO-8859-2, MS Windows Code Page 1250 oraz PC 852. Automatyczna konwersja znaków pomiędzy różnymi ustawieniami stron kodowych po stronie klienta i serwera bazy danych.
- 16) Możliwość migracji bazy danych utrzymujących dane znakowe w 8-bitowej stronie kodowej do Unicode.
- 17) Możliwość definiowania w przestrzeni danych (plików) dla danych użytkownika obszarów o innym niż domyślny rozmiarze bloku.
- 18) Możliwość bez dodatkowych ograniczeń przechowywania wierszy, których rozmiar przekracza rozmiar bloku bazy danych.
- 19) Możliwość budowania indeksów o strukturze B-drzewa. Baza danych powinna umożliwiać założenie indeksu jednej lub większej liczbie kolumn tabeli, przy czym ograniczenie liczby kolumn na których założony jest 1 indeks nie powinno być mniejsze niż 16.
- 20) Możliwość budowania indeksów bitmapowych.
- 21) Możliwość tworzenia / odbudowy indeksów online, bez blokowania zapytań i transakcji operujących na tabelach, dla których są tworzone / odbudowywane indeksy.
- 22) Możliwość budowania widoków zmaterializowanych odzwierciedlających stan danych zdefiniowanych przez zapytanie SQL. Widok zmaterializowany przechowuje rezultat zapytania, którego aktualizacja odbywa się w jednej z dostępnych strategii – na żądanie, okresowo bądź po każdym zatwierdzeniu transakcji modyfikującej tabelę, na której oparty jest widok zmaterializowany.

- 23) Możliwość szybkiego odświeżania danych w widoku zmaterializowanym na podstawie mechanizmu identyfikacji zmian w danych źródłowych.
- 24) Możliwość automatycznego skorzystania przez optymalizator SQL z danych zgromadzonych w widoku zmaterializowanym do celu optymalizacji bardziej złożonych zapytań. Możliwość kontroli wykorzystania widoków zmaterializowanych do ww. celu w przypadku, gdy nie zawierają informacji odzwierciedlających aktualnego stanu danych źródłowych.
- 25) Brak formalnych ograniczeń na liczbę tabel i indeksów w bazie danych oraz na ich rozmiar (liczbę wierszy).
- 26) Możliwość równoległego wykonania operacji SQL w bazie danych - dla potrzeb użytkowników (zapytania, ładowanie danych) oraz na potrzeby konserwacji systemu (tworzenie, przebudowa indeksów, backup, odtwarzanie, wyliczanie statystyk dla optymalizatora SQL).
- 27) RDBMS musi posiadać możliwość wycofania usunięcia tabeli w bazie danych do punktu w czasie bez konieczności przeprowadzania odtwarzania całej bazy danych z kopii zapasowej lub odtwarzania bazy danych w innej lokalizacji.
- 28) RDBMS musi posiadać możliwość wycofania przeprowadzonej transakcji na bazie danych przy czym przy wycofaniu transakcji musi być możliwość automatycznego wycofania transakcji zależnych od wycofywanej transakcji.
- 29) RDBMS musi posiadać możliwość buforowania wyników zapytań SQL (pobranie wyniku instrukcji SQL odbywa się bezpośrednio z bufora zamiast ponownego wykonania zapytania).
- 30) RDBMS musi posiadać możliwość redefinicji/przebudowy tabel w trybie online.
- 31) Kosztowy model optymalizacji instrukcji SQL.
- 32) Model statystyk optymalizatora kosztowego musi pozwalać na odwzorowanie nierównomierności rozkładu danych (składowanie informacji o rozkładzie wartości występujących w kolumnach za pomocą histogramu bądź porównywalnego funkcjonalnie modelu odwzorowania).
- 33) Możliwość uwzględnienia korelacji wartości występujących w niezależnych kolumnach tabeli w modelu statystyk optymalizatora kosztowego.
- 34) RDBMS powinien umożliwiać wskazywanie optymalizatorowi SQL preferowanych metod optymalizacji na poziomie konfiguracji parametrów pracy serwera bazy danych oraz dla wybranych zapytań. Powinna istnieć możliwość umieszczania wskazówek dla optymalizatora w wybranych instrukcjach SQL.
- 35) Silnik bazy danych musi zapewniać możliwość reaktywnej (podczas wykonywania instrukcji SQL) zmiany planu wykonania zapytania SQL w sytuacji kiedy optymalizator instrukcji SQL uzyska informację, że inna metoda dostępu/połączenia danych będzie efektywniejsza niż użyta początkowo. Zmiana planu wykonania zapytania musi mieć możliwość zajścia dla wykonywanego zapytania bez konieczności zatrzymywania tej instrukcji SQL.
- 36) Wsparcie dla procedur i funkcji składowanych w bazie danych. Język programowania powinien być językiem proceduralnym, blokowym (umożliwiającym deklarowanie zmiennych wewnątrz bloku), oraz wspierającym obsługę wyjątków. W przypadku, gdy wyjątek nie ma zadeklarowanej obsługi wewnątrz bloku, w razie jego wystąpienia wyjątek powinien być automatycznie propagowany do bloku nadrzędnego bądź wywołującej go jednostki programu.
- 37) Procedury i funkcje składowane powinny mieć możliwość parametryzowania za pomocą parametrów prostych jak i parametrów o typach złożonych, definiowanych przez

użytkownika. Funkcje powinny mieć możliwość zwracania rezultatów jako zbioru danych, możliwego do wykorzystania jako źródło danych w instrukcjach SQL (czyli występujących we frazie FROM). Ww. jednostki programowe powinny umożliwiać wywoływanie instrukcji SQL (zapytania, instrukcje DML, DDL), umożliwiać jednoczesne otwarcie wielu tzw. kursorów pobierających paczki danych (wiele wierszy za jednym pobraniem) oraz wspierać mechanizmy transakcyjne (np. zatwierdzanie bądź wycofanie transakcji wewnątrz procedury).

- 38) Możliwość kompilacji procedur składowanych w bazie do postaci kodu binarnego (biblioteki dzielonej).
- 39) Możliwość deklarowania wyzwalaczy (triggerów) na poziomie instrukcji DML (INSERT, UPDATE, DELETE) wykonywanej na tabeli, poziomie każdego wiersza modyfikowanego przez instrukcję DML oraz na poziomie zdarzeń bazy danych (np. próba wykonania instrukcji DML, start serwera, stop serwera, próba zalogowania użytkownika, wystąpienie specyficznego błędu w serwerze). Ponadto mechanizm wyzwalaczy powinien umożliwiać oprogramowanie obsługi instrukcji DML (INSERT, UPDATE, DELETE) wykonywanych na tzw. niemodyfikowalnych widokach (views).
- 40) W przypadku, gdy w wyzwalaczu na poziomie instrukcji DML wystąpi błąd zgłoszony przez motor bazy danych bądź ustawiony wyjątek w kodzie wyzwalacza, wykonywana instrukcja DML musi być automatycznie wycofana przez serwer bazy danych, zaś stan transakcji po wycofaniu musi odzwierciedlać chwilę przed rozpoczęciem instrukcji w której wystąpił ww. błąd lub wyjątek.
- 41) Możliwość wykonania równoczesnych operacji DML (Insert/Update/Delete) na tej samej tabeli .
- 42) Musi istnieć możliwość buforowania wyników funkcji składowanych i wykonywanych w motorze bazy danych (pobranie wyniku funkcji z bufora zamiast ponownego wykonania kodu funkcji).
- 43) Powinna istnieć możliwość autoryzowania użytkowników bazy danych za pomocą rejestru użytkowników założonego w bazie danych bądź mechanizmu zewnętrznego w stosunku do bazy danych.
- 44) Przywileje użytkowników bazy danych powinny być określane za pomocą przywilejów systemowych (np. prawo do podłączenia się do bazy danych - czyli utworzenia sesji, prawo do tworzenia tabel itd.) oraz przywilejów dostępu do obiektów aplikacyjnych (np. odczytu / modyfikacji tabeli, wykonania procedury). Baza danych powinna umożliwiać nadawanie ww. przywilejów za pośrednictwem mechanizmu grup użytkowników / ról bazodanowych. W danej chwili użytkownik może mieć aktywny dowolny podzbiór nadanych ról bazodanowych.
- 45) Możliwość wykonywania i katalogowania kopii bezpieczeństwa bezpośrednio przez serwer bazy danych. Możliwość zautomatyzowanego usuwania zbędnych kopii bezpieczeństwa przy zachowaniu odpowiedniej liczby kopii nadmiarowych - stosownie do założonej polityki nadmiarowości backup'ów. Możliwość integracji z powszechnie stosowanymi systemami backupu (Legato, Veritas, Tivoli, itp.). Wykonywanie kopii bezpieczeństwa powinno być możliwe w trybie offline oraz w trybie online(hot backup)..
- 46) Odtwarzanie powinno umożliwiać odzyskanie stanu danych z chwili wystąpienia awarii bądź cofnąć stan bazy danych do punktu w czasie. W przypadku odtwarzania do stanu z chwili wystąpienia awarii odtwarzaniu może podlegać cała baza danych bądź pojedyncze pliki danych.
- 47) W zależności od rozmiaru uszkodzeń powinna być możliwość odtwarzania na poziomie całej bazy danych, pojedynczych wskazanych plików, pojedynczych wskazanych bloków

- danych. W sytuacji odtwarzania pojedynczych plików lub bloków danych, dla pozostałych nieuszkodzone obszarów danych powinna istnieć możliwość udostępnienia ich dla operacji użytkowników.
- 48) Możliwość zarządzania przydziałem zasobów obliczeniowych dla użytkowników bazy danych (Resource Manager).
 - 49) Możliwość zmiany przypisanych planów alokacji zasobów obliczeniowych w trakcie pracy systemu.
 - 50) Wbudowany w RDBMS mechanizm replikacji pomiędzy bazami danych źródłową, a docelową. Replikacji muszą podlegać wszystkie zmiany na bazie danych źródłowej. Rozwiązanie musi umożliwiać definiowanie wielu baz docelowych.
 - 51) Replikacja pomiędzy bazami danych musi mieć charakter logiczny. Zmiany replikowane muszą być zabezpieczone odpowiednimi mechanizmami zapewniającymi spójność replikowanych zmian oraz weryfikacja po stronie źródłowej jak i docelowej.
 - 52) Możliwość konfiguracji replikacji synchronicznej oraz asynchronicznej pomiędzy bazami danych.
 - 53) Motor serwera bazy danych powinien udostępniać charakterystyki wydajnościowe za pomocą otwartego, udokumentowanego zasobu metadanych.
 - 54) Funkcjonalność kompresji jest funkcjonalnością wbudowana niewymagająca rozbudowy przy pomocy zewnętrznych narzędzi programistycznych.
 - 55) Funkcjonalność kompresji musi być funkcjonalnością oprogramowania bazodanowego niezależna od systemu operacyjnego na którym pracuje oprogramowanie baz danych.
 - 56) Rozwiązanie dostarcza funkcjonalności kompresji danych znajdujących się w tabelach bazy danych. Użyty algorytm kompresji nie powinien degradować wydajności aplikacji o charakterze pracy OLTP.
 - 57) Użyty mechanizm kompresji nie powinien uniemożliwiać operacji DML na bazie danych. Operacje DML nie powinny wpływać na poziom kompresji danych w tabeli.
 - 58) Użyty mechanizm kompresji nie powinien wymuszać specjalnego trybu ładowania/uaktualniania danych do/w tabeli.
 - 59) Rozwiązanie dostarcza funkcjonalności kompresji eksportów logicznych bazy danych.
 - 60) Rozwiązanie dostarcza funkcjonalności kompresji danych niestrukturalnych przechowywanych w bazie danych. Dane takie przechowywane są zazwyczaj jako tzw. wielkie obiekty - LOB. Funkcjonalność kompresji danych niestrukturalnych powinna mieć możliwość konfiguracji wykorzystanego algorytmu kompresji. Dostępne powinny być przynajmniej trzy poziomy kompresji danych przechowywanych w bazie danych jako obiekty LOB.
 - 61) Rozwiązanie dostarcza funkcjonalności deduplikacji danych niestrukturalnych przechowywanych w bazie danych. Dane takie przechowywane są zazwyczaj jako tzw. wielkie obiekty - LOB. Funkcjonalność deduplikacji obiektów LOB pozwala na zmniejszenie przestrzeni potrzebnej na przechowywanie ich w bazie danych optymalizując powtarzające się fragmenty danych w wielkim obiekcie.
 - 62) Rozwiązanie dostarcza funkcjonalności kompresji transmisji pomiędzy baza danych podstawowa a baza danych standby znajdująca się w ośrodku zapasowym w celu obniżenia wymagań na pasmo pomiędzy ośrodkiem podstawowym a ośrodkiem zapasowym.
 - 63) Rozwiązanie dostarcza funkcjonalności kompresji kopii zapasowych (backup) bazy danych. Kompresja kopii zapasowych musi odbywać się od razu w czasie tworzenia kopii zapasowej. Funkcjonalność kompresji kopii zapasowych powinna mieć możliwość konfiguracji wykorzystanego algorytmu kompresji. W zależności od użytego algorytmu

- użytkownik może sterować zależnością pomiędzy współczynnikiem kompresji, wykorzystaniem CPU, czasem wykonania kopii zapasowej. Mechanizm kompresji kopii zapasowych musi dostarczać przynajmniej trzech poziomów kompresji.
- 64) Rozwiązanie dostarcza funkcjonalności kompresji transmisji pomiędzy baza danych a aplikacja.
 - 65) Rozwiązanie dostarcza funkcjonalności automatycznej implementacji zarządzania cyklem życia danych (ILM Information Lifecycle Management). Dla danych często wykorzystywanych stosowany jest algorytm kompresji nie pogarszający wydajności pracy bazy danych w trybie OLTP. Dane archiwalne są automatycznie kompresowane silnymi algorytmami kompresji, mogą być w sposób online'owy (bez wpływu na prace użytkowników) przenoszone na inną warstwę storage (Automated Tiered Storage).
 - 66) Rozwiązanie dostarcza funkcjonalności śledzenia wykonywanych operacji SQL i DML na tabelach i partycjach tabel.
 - 67) Rozwiązanie dostarcza funkcjonalności wsparcia wykonywania kopii zapasowych z udziałem mechanizmów dostępnych na macierzach dyskowych np. tworzenia tzw. snapshot lub clone.
 - 68) Rozwiązanie dostarcza funkcjonalności online'owego przeniesienia partycji tabel przy czym przeniesienie partycji może być połączone z kompresją danych.
5. Zaoferowane oprogramowanie równoważne powinno zapewnić warunki licencji w każdym aspekcie licencjonowania nie gorsze niż licencje oprogramowania wyszczególnione w pkt 1.
 6. Nabycie licencji oprogramowania równoważnego ma pozwalać na legalne używanie posiadanych przez Zamawiającego licencji oprogramowania.
 7. Zaoferowane oprogramowanie równoważne powinno umożliwić zwiększenie licencji na każdy procesor w klastrze bez konieczności jego instalacji na posiadanych przez Zamawiającego serwerach.
 8. Dodatkowo, równoważne oprogramowanie zarządzające powinno umożliwić zarządzanie i konfigurowanie działających baz danych Zamawiającego.
 9. Oprogramowanie równoważne nie może zakłócać pracy środowiska systemowo-programowego Zamawiającego i współdziałać ze sprzętem i serwerem funkcjonującym u Zamawiającego.
 10. Jeżeli potrzeby oferowanego oprogramowania równoważnego będą przekraczały wymagania środowiska opartego na serwerze posiadany przez Zamawiającego, lub w jakikolwiek sposób będą wymagały stosowania dodatkowych licencji lub innych technologii niż na serwerze posiadany przez Zamawiającego, Wykonawca dostarczy na własny koszt niezbędną infrastrukturę sprzętową wraz ze wszystkimi licencjami wymaganymi w celu prawidłowego wdrożenia i dalszego funkcjonowania oferowanego oprogramowania równoważnego.
 11. W przypadku gdy zaoferowane oprogramowanie równoważne nie będzie właściwie działać ze sprzętem i oprogramowaniem funkcjonującym u Zamawiającego lub spowoduje zakłócenia w funkcjonowaniu pracy środowiska sprzętowo-programowego u Zamawiającego, Wykonawca pokryje koszty związane z przywróceniem i sprawnym działaniem infrastruktury sprzętowo programowej Zamawiającego oraz na własny koszt dokona niezbędnych modyfikacji przywracających właściwe działanie środowiska sprzętowo - programowego Zamawiającego również po odinstalowaniu oprogramowania równoważnego.

12. Oprogramowanie równoważne musi być kompatybilne i w sposób niezakłócony współdziałać ze sprzętem i systemem informatycznym funkcjonującym u Zamawiającego.
13. W przypadku zaoferowania przez Wykonawcę oprogramowania równoważnego Wykonawca przeszkoli na swój koszt 3 pracowników Zamawiającego w zakresie obsługi oprogramowania i implementacji funkcjonalności produktu równoważnego.