

XIII Konferencja PLOUG  
Kościelisko  
Październik 2007

# Archiwizacja danych historycznych

Krzysztof Mikołajczyk  
Bull Polska

*krzysztof.mikolajczyk@bull.com.pl*

**Abstrakt.** W artykule zostały zamieszczone podstawowe informacje o archiwizacji danych historycznych. Przedstawione zostały różne metody wykonywania archiwizacji oraz różne problemy, na które można się natknąć w trakcie wykonywania pracy.

## 1. Wprowadzenie

Bazy danych służą do przechowywania danych. Z czasem ilość informacji przechowywana wewnątrz bazy danych przyrasta do tego stopnia, że nie da się powiększyć plików bazy danych i składowanie nowych informacji staje się niemożliwe (mogą występować również problemy z aktualizacją danych). W tym momencie baza przestaje być użyteczna, przestaje spełniać swoją rolę. Dodatkowym problemem może być wydajność bazy, gdyż wraz ze wzrostem pojemności bazy będzie również wzrastał czas dostępu. Administrator musi wykonać jakąś akcję, która pozwoli korzystać z funkcjonalności bazy danych. Administrator może wykonać jedną z następujących czynności:

- Dołożyć nowe zasoby dyskowe (nowa macierz).
- Zamienić aktualne dyski na dyski o większej pojemności.
- Usunąć część danych.

Zarówno wymiana dysków, jak i całych macierzy mają swoje zalety, jakimi są przede wszystkim większa pojemność i lepszy czas dostępu. Obydwie metody dadzą większą pojemność na nowe dane, jak również lepszą wydajność. Jednak powyższe metody muszą mieć odpowiednie uzasadnienie – po pierwsze są to metody kosztowne (zakup nowych elementów sprzętowych, migracja danych), po drugie jest to tylko odsunięcie rzeczywistego problemu w czasie – prędzej czy później nowa pojemność również się skończy.

W związku z tym pozostaje metoda trzecia, jaką jest usuwanie części danych, które są mniej lub bardziej potrzebne. Oczywiście najłatwiej jest usunąć dane, które są zupełnie niepotrzebne. Jednak większość danych przechowywanych w bazie może być potrzebna, ale niekoniecznie w trybie natychmiastowym. Archiwizacja danych historycznych pozwala utrzymać wielkość bazy danych na mniej więcej stałym poziomie (a przynajmniej zmniejsza przyrost wielkości), utrzymywać wydajność na zbliżonym poziomie, zmniejszyć wymagania co do backupu i odtwarzania (czas i miejsce), a jednocześnie pozwala na dostęp do danych historycznych. Dlatego ważne jest prawidłowe opracowanie sposobu wykonywania archiwizacji (i ewentualnego odtworzenia) danych i właściwego ich składowania.

## 2. Charakterystyka danych

Cechą charakterystyczną wszystkich danych jest ich ciągły przyrost. Najlepszym przykładem składnicy danych jest pamięć ludzka. Wraz z wiekiem ilość zapamiętanych danych (faktów, zdarzeń) jest coraz większa, jednak czas dostępu do zapamiętanych danych może wzrastać. Podobnie jest w przypadku systemów bazodanowych – gdy ilość zgromadzonych danych rośnie, rośnie również czas dostępu do nich (nie mówiąc już o potrzebnej przestrzeni dyskowej/taśmowej, która jest wymagana do ich przechowywania). Dlatego ważne jest segregowanie zbieranych danych i umiejętność określenia dostępu do określonych danych.

### 2.1. Okresy

Generalnie „życie” danych można podzielić na 5 podstawowych okresów (okresy wyrażone w latach są orientacyjne, dla różnych baz i aplikacji będą różne):

- Dane bieżące – zwykle 1-2 lata.
- Dane historyczne (niezbędne do raportów i ewentualnie weryfikacji) – zwykle 2-5 lat.
- Dane historyczne – bieżące archiwum (dane potrzebne do okresowej weryfikacji bądź bardziej zaawansowanych raportów) – zwykle 4-8 lat.

- Dane historyczne – wymagane archiwum (dane, które muszą być przetrzymywane głównie ze względów prawnych z okresowym, bardzo rzadkim dostępem) – zwykle 7-50 lat.
- Dane zbędne (dane, które z różnych względów się zdezaktualizowały, a są nadal przechowywane w bazie) – 1-∞ lat.

Przedstawione okresy przechowywania danych są orientacyjne i dla różnych aplikacji i baz danych mogą być różne. Podział na okresy najłatwiej jest wyrazić poprzez czas, jednak dla pewnych aplikacji można zastosować również inne kryteria – wtedy kryterium czasu jest jedynie pomocnicze.

## 2.2. Charakterystyka

Przedstawione powyżej okresy (podział ze względu na czas lub inne kryterium) są w rzeczywistości tylko pomocnicze. Rzeczywisty podział na okresy wynika z dopuszczalnego czasu dostępu do określonych danych (i np. wymogów prawnych). I tak dla danych bieżących jest to czas bardzo krótki (w zależności od aplikacji będzie to czas rzędu sekund lub dziesiątek sekund). Jeżeli dopuszczalny czas dostępu jest na poziomie minut lub dziesiątek minut (lub więcej) i dane nie są używane w sposób ciągły, to mamy do czynienia z danymi historycznymi, które można z podstawowej bazy usunąć, aby zwiększyć wydajność przy dostępie do danych aktualnych. Dane, do których dostęp jest bardzo rzadki i nie jest istotny czas dostępu, mogą być przechowywane w archiwum taśmowym (lub na innych nośnikach). Jednak w każdej sytuacji musi być opracowana metoda dostępu do danych archiwalnych.

## 3. Tworzenie planu archiwizacji

Utworzenie dobrego planu archiwizacji może być dość skomplikowaną pracą. Wymaga przede wszystkim dobrej znajomości struktury bazy danych, zarówno logicznej, jak i merytorycznej. Drugim niezwykle ważnym elementem jest opracowanie planu testów archiwizacji i zrealizowanie tego planu. Dopiero po zweryfikowaniu poprawności danych można przystąpić do rzeczywistej archiwizacji.

Przy tworzeniu planu archiwizacji należy wziąć pod uwagę kilka czynników:

- określenie kryterium wyboru danych do archiwizacji (kryterium czasowe wydaje się najbardziej oczywiste),
- zależności między danymi (tabelami),
- metoda dostępu do zarchiwizowanych danych,
- czy (i kiedy) dane mogą (bądź muszą) być trwale usunięte.

Sama archiwizacja danych historycznych będzie w najprostszym przypadku przeprowadzana w trzech krokach (plus czwarty związany z ewentualnym odtwarzaniem):

1. Przygotowanie zestawu danych do archiwizacji.
2. Pobranie danych i składowanie ich w miejscu docelowym.
3. Usunięcie danych z tabel źródłowych.
4. Dostęp do danych archiwalnych.

Przy określaniu zestawu danych do usunięcia (i archiwizacji) najważniejsze jest określenie późniejszego sposobu dostępu do tych danych. Należy również pamiętać o tym, aby usunięte dane nie powodowały problemów przy korzystaniu z pozostałych danych.

### 3.1. Przygotowanie zestawu danych do archiwizacji

Jest to najważniejszy i najtrudniejszy etap pracy. Najistotniejszą rzeczą jest poznanie i ewentualnie prawidłowe określenie zależności między danymi. Chodzi głównie o to, aby usunięcie danych, jak i ich ponowne załadowanie nie powodowało żadnych problemów – dane muszą być spójne. Należy zwrócić uwagę, czy wprowadzane modyfikacje nie spowodują naruszenia więzów integralności, jak również nie zostaną wprowadzone dodatkowe modyfikacje (np. na skutek działania pewnych elementów aplikacji, jak wyzwalacze). Archiwizacja pewnych danych z jednej tabeli może również spowodować, że pewne dane z innej tabeli też będą musiały być zarchiwizowane (nawet, jeśli nie byłyby usuwane – mogą być usuwane przy następnej iteracji archiwizacji). Chodzi o to, żeby po ewentualnym odtworzeniu obraz danych był spójny.

Przy planowaniu zestawu danych do archiwizacji należy również wziąć pod uwagę czas potrzebny na wykonanie zadania. Może się okazać, że na czas wykonania archiwizacji dane będą niedostępne dla użytkowników.

### 3.2. Pobranie danych i składowanie ich w miejscu docelowym

Przy pobieraniu danych należy pamiętać o tym, aby nie dopuścić do modyfikacji danych. Wprawdzie dane historyczne nie powinny być modyfikowane, ale dla poprawnego wykonania archiwizacji należy je zabezpieczyć. Ilość usuwanych danych (i wcześniej archiwizowanych) ma wpływ na czas wykonania operacji – w tym czasie może się zdarzyć, że zablokowane będą również dane, które w danym momencie nie są archiwizowane.

Do pobrania zestawu danych najwygodniej jest użyć procedur SQL (PL/SQL). W pewnych przypadkach można użyć programu RMAN.

### 3.3. Usunięcie danych z tabel źródłowych

Usuwanie danych historycznych z bazy ma dwa podstawowe cele:

- Zmniejszenie zajętości przestrzeni dyskowej przez bazę danych.
- Poprawienie wydajności przy dostępie do danych bieżących.

Jeżeli dane historyczne muszą być nadal dostępne, to osiągnięcie pierwszego celu jest dość trudne – dane te i tak muszą zajmować przestrzeń (mogą zająć nawet więcej miejsca, niż w podstawowej bazie). Jednak taki podział pozwala na fizyczne rozdzielenie danych na różne macierze i w efekcie można zmniejszyć koszt rozwiązania.

Drugi cel jest częściowo osiągnięty poprzez samo usunięcie danych. Jednak dobrze jest wykonać pewne dodatkowe operacje, które pozwolą jeszcze bardziej poprawić wydajność (samo usunięcie danych zmniejsza rozmiar danych, nie zmniejsza jednak obiektów, w których te dane są przechowywane).

Po pobraniu zarchiwizowanych danych mogą być one usunięte z bazy danych. Jest to ciąg dalszy transakcji pobierania danych (dane nie mogą być modyfikowane, nie mogą się pojawić nowe dane, które spełniałyby warunki usunięcia). Najbezpieczniejszą metodą jest używanie ROWID do identyfikacji usuwanych wierszy.

Usunięcie danych zmienia sposób dostępu do danych pozostałych w bazie. Spośród rzeczy, które warto w tym momencie wziąć pod uwagę, są:

- High Water Mark (HWM), który mimo usunięcia sporej porcji danych nie został zmieniony dla danej tabeli – powoduje to, że przy pełnym przeglądzie tabeli (full table scan) jest wykonywany odczyt takiej samej ilości bloków, jak przed usunięciem,
- Indeksy, które warto przebudować po operacji usuwania,

- Statystyki i histogramy dla poszczególnych tabel, które mogą być nieadekwatne do aktualnej sytuacji.

Jeżeli jest usuwana większa ilość danych (np. przy pierwszym podejściu do usuwania danych historycznych) problemy te mogą być ważniejsze, niż przy kolejnych, cyklicznie wykonywanych operacjach (z mniejszą ilością usuwanych danych).

W przypadku HWM, ponieważ nowe dane będą ładowane do tabel, to prawdopodobnie nie będzie wymagana modyfikacja (szczególnie w przypadku małych, cyklicznych modyfikacji). Jednak jeśli występuje często operacja pełnego przeglądu tabel (szczególnie przy większych zmianach ilości danych), to w celu poprawienia wydajności może być zalecane przesunięcie HWM.

Po operacji usunięcia danych statystyki powinny być zawsze przeliczone dla określonych tabel (raczej COMPUTE dla dużej zmiany, a nie ESTIMATE). Indeksy mogą również wymagać przebudowy – jest to również zależne od ilości usuwanych wierszy, a także sposobu dostępu do danych – po przebudowie indeks może być zdecydowanie mniejszy i może być lepsze jego wykorzystanie.

### 3.4. Dostęp do danych archiwalnych

Dane historyczne są to dane, które od czasu do czasu mogą być wykorzystywane, a więc musi zostać opracowana metoda dostępu do nich. Może być zdefiniowanych kilka metod (np. w zależności od opóźnienia w czasie dostępu do nich), jednak należy pamiętać o dostosowaniu bazy danych i/lub aplikacji do wybranej metody (lub odwrotnie).

Najprostszym sposobem jest ponowne załadowanie danych historycznych do oryginalnych tabel. W tym wypadku jest konieczne sprawdzenie zależności (więzy, wyzwalacze), aby załadowanie danych historycznych nie spowodowało modyfikacji danych bieżących. Ważne może być również ponowne usunięcie tych danych po przeprowadzeniu operacji, które ich wymagały.

Drugą metodą jest łączenie zestawów danych bieżących i historycznych. Ta metoda wymaga, aby dane historyczne były dostępne w bazie (archiwizacja do plików płaskich w tym wypadku nie zadziała). Może być również konieczne zmodyfikowanie aplikacji i/lub struktury bazy danych – zamiana dostępu bezpośrednio do tabel na dostęp do synonimów i perspektyw. Jest to metoda podobna do tabel partycjonowanych – tabele partycjonowane są jakby z definicji przeznaczone do obsługi danych, które się „starzeją” i są po pewnym czasie usuwane.

## 4. Składowanie danych

Poniżej zostaną przedstawione wybrane trzy metody składowania danych, ich plusy, minusy i mechanizm wykonywania. Nie są to gotowe procedury do zastosowania w rzeczywistej sytuacji, ale przykłady, na bazie których można zbudować własną procedurę.

### 4.1. Metody składowania

Składowanie danych można przeprowadzić na następujący sposoby:

- Archiwizowane dane są przechowywane w plikach płaskich.
- Archiwizowane dane są przerzucane do innej bazy danych (schematu).
- Jest tworzona kopia bazy danych, która przechowuje wszystkie dane historyczne.

W celu oszczędzenia przestrzeni dyskowej dane mogą być po określonym czasie składowane również na taśmach lub innych nośnikach. Zawsze trzeba jednak pamiętać o opracowaniu sposobu dostępu do usuniętych danych (czy to poprzez ponowne załadowanie ich do aktualnej bazy, czy

też poprzez sięganie do innej bazy). Sposób dostępu musi być dobrze opracowany i najlepiej sprawdzony praktycznie przed rzeczywistym zastosowaniem (usunięciem danych).

## 4.2. Dane przechowywane w plikach płaskich.

Po określeniu zestawu danych do archiwizacji są one przeliczane do zestawu plików płaskich (lub pojedynczego pliku) za pomocą SQL, `exp` lub innych narzędzi. Po zakończeniu operacji eksportu zestawu danych są one usuwane z bazy danych. Pobieranie i usuwanie danych stanowi jedną operację. W tym czasie dostęp do danych może być ograniczony (dane muszą być zablokowane, aby nie zostały zmodyfikowane – również niektóre dane, które nie będą usuwane). Można (jest to zależne od zestawu danych) podzielić taką operację na kilka mniejszych. Pliki zawierające dane historyczne mogą być w razie potrzeby przeliczane na taśmy.

W razie konieczności dostępu do danych historycznych konieczne jest określenie zakresu tych danych. Jeśli są one już przeliczone na taśmy, to pierwszym krokiem będzie odtworzenie ich na dysk. Jeśli dane są już dostępne na dysku, to następuje ich załadowanie do oryginalnych tabel bazy danych (zachowując odpowiednią ostrożność co do ograniczeń aplikacji typu wyzwalacze czy też więzy integralności).

Cechy charakterystyczne rozwiązania:

- + Nie są wymagane dodatkowe zasoby bazy danych (tylko przestrzeń dyskowa i/lub taśmowa na składowanie plików).
- + Jeśli pliki są tekstowe, to dostęp do danych historycznych może być również realizowany spoza bazy danych (przy pomocy edytora tekstu).
- + Łatwe utrzymanie (jedna baza + zestawy plików).
- + Łatwe modyfikacje struktury archiwizowanych danych (wymagają tylko modyfikacji skryptów – archiwizujących i odtwarzających).
- + Poza okresami archiwizacji i odtwarzania nie absorbują zasobów (procesor i pamięć).
- Wydłużony czas dostępu.
- Zarządzanie zestawami plików jest trudniejsze – łatwo „zgubić” plik.
- Konieczne zapamiętanie struktury i zależności między danymi dla kolejnych zestawów plików.
- Zarządzanie skryptami do ładowania danych jest dość skomplikowane przy zmianie struktury archiwizowanych danych.

## 4.3. Dane przechowywane w innym schemacie bazy danych (kopia częściowa)

Archiwizowane dane są przeliczane do innego schematu bazy danych, a następnie usuwane ze schematu podstawowego bazy. Przeliczenie danych można zrealizować bezpośrednio procedurami SQL lub poprzez `exp/imp`. Dane historyczne powinny być przechowywane w przestrzeniach tabel składowanych na innej macierzy (zmniejszenie kosztów).

Cechy charakterystyczne rozwiązania:

- + Dane historyczne są dostępne bezpośrednio w bazie historycznej.
- + Nie jest wymagana dodatkowa licencja na oprogramowanie Oracle.
- + Łatwiejsze utrzymanie danych archiwalnych (są zarządzane przez bazę danych).
- + Można zastosować transportowalne przestrzenie tabel.

- + Dodatkowa moc obliczeniowa i pamięć wymagana tylko w momencie aktywnego używania zestawu danych historycznych.
- + Łatwe przenoszenie danych.
- + Szybki dostęp do bazy historycznej (kwestia zdefiniowania odpowiednich synonimów i uprawnień w bazie – można wtedy zrealizować bez przełączania użytkowników).
- + Dane historyczne mogą być przechowywane na innej macierzy niż dane bieżące (macierz o większej pojemności i wolniejszym dostępie).
- Wymagane stworzenie i utrzymanie drugiego zestawu tabel – konieczna dodatkowa moc obliczeniowa, dodatkowa pamięć, dodatkowe zasoby dyskowe (z reguły wymagania większe, niż dla plików płaskich).
- Trudniejsze modyfikacje struktury archiwowanych danych (modyfikacje skryptów archiwizujących i odtwarzających, modyfikacje struktur bazy danych).
- Przyrost wielkości bazy danych i rywalizacja o zasoby wewnątrz bazy (np. bufory danych).

#### 4.4. Dane przechowywane w drugiej bazie danych (kopia częściowa)

Archiwizowane dane są przetrzucane do drugiej bazy danych, a następnie usuwane z bazy podstawowej. Przerzucanie danych można zrealizować bezpośrednio, jak również poprzez pliki płaskie (pobranie danych do pliku i następnie załadowanie do drugiej bazy). W pierwszym przypadku może być to wykonane procedurami SQL poprzez `dblink`. W drugim przypadku może być to zrealizowane poprzez SQL (z ewentualnym ładowaniem danych przez `SQL*Loader`), poprzez RMAN lub przez `exp/imp`. Ponieważ w tym przypadku stare dane historyczne mogą być przechowywane na taśmach, to druga metoda jest prostsza – na taśmach są składowane pliki płaskie (od razu), natomiast stare dane są również usuwane z historycznej bazy danych.

Baza danych archiwalnych może być utworzona na tym samym serwerze (partycji). Wybór rozwiązania zależy od zastosowanego sprzętu i systemu operacyjnego. W pewnych przypadkach można zdefiniować to rozwiązanie przy minimalnym nakładzie kosztów – tylko na składowanie danych z możliwością zastosowania prostszej macierzy (dającej dużą pojemność przy wolniejszym dostępie) i prawie zerowym wykorzystaniu mocy obliczeniowej i pamięci systemu (w momencie, gdy baza archiwalna nie jest aktywnie wykorzystywana).

W przypadku konieczności skorzystania z danych historycznych występują dwa przypadki – jeśli są potrzebne tylko i wyłącznie dane historyczne (np. w celu weryfikacji), to wystarczy bezpośredni dostęp do bazy historycznej, jeśli potrzebne są zarówno dane bieżące, jak i historyczne, to dane historyczne trzeba przetrzucić do bazy podstawowej (lub odwrotnie), a po wykonaniu wymaganej operacji usunąć je.

Cechy charakterystyczne rozwiązania:

- + Dane historyczne są dostępne bezpośrednio w bazie historycznej.
- + Jeśli realizowane w ramach tej samej maszyny, to nie jest wymagana licencja na oprogramowanie Oracle.
- + Może wystarczyć licencja Standard (jeśli jest konieczna).
- + Łatwiejsze utrzymanie danych archiwalnych (są zarządzane przez bazę danych).
- + Można zastosować transportowalne przestrzenie tabel.
- + Dane historyczne mogą być przechowywane na innej macierzy niż dane bieżące (macierz o większej pojemności i wolniejszym dostępie).
- + rozdzielenie danych produkcyjnych od danych archiwalnych – w krytycznych sytuacjach można bazę archiwalną zamknąć, aby zwolnić zasoby dla krytycznego zadania.

- + brak rywalizacji o dostęp do zasobów macierzy (może być rywalizacja o dostęp do adaptera).
- Wymagane stworzenie i utrzymanie drugiej bazy danych – konieczna dodatkowa moc obliczeniowa, dodatkowa pamięć, dodatkowe zasoby dyskowe (z reguły wymagania większe, niż dla plików płaskich).
- Może być wymagana dodatkowa licencja na oprogramowanie Oracle.
- Zasoby systemu (procesor i pamięć) są zajęte przez instancję archiwalnej bazy danych (w szczególności gdy baza jest instalowana na oddzielnym serwerze (partycji)).
- Trudniejsze modyfikacje struktury archiwowanych danych (modyfikacje skryptów archiwizujących i odtwarzających, modyfikacje struktur bazy danych).
- rywalizacja o zasoby systemu operacyjnego (np. procesory, pamięć, połączenia do sieci – LAN, SAN).
- trudniejszy dostęp do bazy danych archiwalnych – konieczność przełączania.

#### **4.5. Dane przechowywane w drugiej bazie danych na innym systemie/partycji (kopia częściowa)**

Archiwizowane dane są przelicane do drugiej bazy danych, utworzonej na innym serwerze (może być serwer starszej generacji, np. po przeniesieniu aplikacji i bazy danych na nowy serwer), a następnie usuwane z bazy podstawowej. Przelicanie danych można zrealizować tymi samymi metodami, co w poprzednim przypadku – bezpośrednio lub przez pliki płaskie (pobranie danych do pliku i następnie załadowanie do drugiej bazy).

Baza danych archiwalnych jest utworzona na tym innym serwerze (partycji). Dzięki wykorzystaniu starego serwera i ewentualnie starej macierzy koszt tego rozwiązania jest minimalny (może być wymagana dodatkowa licencja Oracle).

W przypadku konieczności skorzystania z danych historycznych występują dwa przypadki – jeśli są potrzebne tylko i wyłącznie dane historyczne (np. w celu weryfikacji), to wystarczy bezpośredni dostęp do bazy historycznej, jeśli potrzebne są zarówno dane bieżące, jak i historyczne, to dane historyczne trzeba przeliczyć do bazy podstawowej (lub odwrotnie), a po wykonaniu wymaganej operacji usunąć je.

Cechy charakterystyczne rozwiązania:

- + Dane historyczne są dostępne bezpośrednio w bazie historycznej.
- + Może wystarczyć licencja Standard (jeśli jest konieczna).
- + Łatwiejsze utrzymanie danych archiwalnych (są zarządzane przez bazę danych).
- + Można zastosować transportowalne przestrzenie tabel.
- + Dane historyczne są przechowywane na innej macierzy niż dane bieżące (macierz o większej pojemności i wolniejszym dostępie).
- + rozdzielenie danych produkcyjnych od danych archiwalnych – dostęp do bazy archiwalnej nie ma wpływu (lub jest to wpływ minimalny) na wydajność bazy produkcyjnej.
- + brak rywalizacji o dostęp do zasobów macierzy.
- Wymagane stworzenie i utrzymanie drugiej bazy danych – konieczna dodatkowa moc obliczeniowa, dodatkowa pamięć, dodatkowe zasoby dyskowe związane z dodatkowym serwerem i macierzą.
- Może być wymagana dodatkowa licencja na oprogramowanie Oracle.

- Trudniejsze modyfikacje struktury archiwowanych danych (modyfikacje skryptów archiwizujących i odtwarzających, modyfikacje struktur bazy danych).
- trudniejszy dostęp do bazy danych archiwalnych – konieczność przełączania.

#### 4.6. Dane przechowywane w drugiej bazie danych (pełna kopia)

Archiwizowane dane są przelicane do drugiej bazy danych, a następnie usuwane z bazy podstawowej. Dane mogą być przelicane metodami wskazanymi wcześniej - bezpośrednio, poprzez pliki płaskie, `exp/imp` lub innymi narzędziami. Baza historyczna przechowuje wszystkie dane archiwalne (ewentualnie z dokładnością do danych zdezaktualizowanych, które są trwale usuwane z bazy).

W przypadku konieczności skorzystania z danych historycznych występują dwa przypadki – jeśli są potrzebne tylko dane historyczne, to jest bezpośredni dostęp do bazy historycznej, jeśli potrzebne są zarówno dane bieżące, jak i historyczne, to są one przelicane do bazy podstawowej, a po wykonaniu operacji usuwane z niej.

Cechy charakterystyczne rozwiązania:

- + Pełna kopia danych historycznych (bez konieczności korzystania z zasobów taśmowych).
- + Część operacji może być wykonana wyłącznie przy użyciu bazy historycznej.
- + Jeśli realizowane w ramach tej samej maszyny, to nie jest wymagana licencja na oprogramowanie Oracle.
- + Może wystarczyć licencja Standard (jeśli konieczna).
- + Być może da się zastosować transportowalne przestrzenie tabel.
- Wymagane stworzenie drugiej bazy danych – dodatkowa moc obliczeniowa, dodatkowa pamięć, dodatkowe zasoby dyskowe.
- Może być wymagana dodatkowa licencja na oprogramowanie Oracle.
- Duże zasoby dyskowe (ciągły przyrost).
- Spadek wydajności z upływem czasu (coraz więcej danych).
- Zasoby systemu (procesor i pamięć) są zajęte przez instancję archiwalnej bazy danych (w szczególności gdy baza jest instalowana na oddzielnym serwerze (partycji)).
- Trudniejsze modyfikacje struktury archiwowanych danych (modyfikacje skryptów archiwizujących i odtwarzających, modyfikacje struktur bazy danych).
- Rozwiązanie trudne do zastosowania w przypadku dużych przyrostów ilości danych.

### 5. Podsumowanie

Przy podejmowaniu decyzji o usunięciu danych historycznych z bazy należy przede wszystkim zastanowić się, czy będzie potrzebny dostęp do tych danych i jak on będzie realizowany. Najprostszą metodą jest dostęp oddzielny (dane bieżące i historyczne są odseparowane) – wtedy nie ma problemu z ponownym ładowaniem danych historycznych do bazy danych bieżących ani z modyfikacją aplikacji w celu łączenia tych danych. Dobrze opracowany plan archiwizacji i dostępu do danych (i przede wszystkim sprawdzony w praktyce na bazie testowej) jest podstawą do wykonania tej operacji na bazach produkcyjnych.

Drugą rzeczą, o której warto pamiętać, jest zadbanie o poprawę wydajności bazy produkcyjnej. Usunięcie części danych jest dopiero początkiem pracy. Należy wykonać zestaw dodatkowych czynności, aby osiągnąć pełny sukces w optymalizacji wydajności bazy produkcyjnej.

**Bibliografia**

- [Jar04] Dokumentacja produktowa Oracle  
Jarlstrom E.: Information Lifecycle Management for Oracle Apps Data, Materiały konferen-  
cyjne COUG, 2004 ([www.nocoug.org/presentations.html](http://www.nocoug.org/presentations.html))