**Politechnika Wrocławska**

**Wydział Elektroniki**

# LABORATORIUM Z ROZPROSZONYCH I OBIEKTOWYCH SYSTEMÓW BAZ DANYCH

System obsługi internetowego rozkładu jazdy, oparty o rozproszoną bazę danych

|  |  |
| --- | --- |
| Autorzy: Michał KowalczykIndeks: 163368Piotr StęplewskiIndeks: 164713 | Prowadzący zajęcia:  Dr inż. Robert Wójcik, W4/I-6 |
|  | Ocena pracy: |

### Wrocław 2011

Spis treści

[1. Wstęp 6](#_Toc312766402)

[2. Cel i zakres pracy 6](#_Toc312766403)

[2.1. Cel projektu 6](#_Toc312766404)

[2.2. Zakres projektu 6](#_Toc312766405)

[3. Replikacja w systemie baz danych MS SQL Server 2008 7](#_Toc312766406)

[3.1. Przegląd typów replikacji 7](#_Toc312766407)

[3.2. Przegląd programów (pośredników) wykorzystywanych przez MS SQL 10](#_Toc312766408)

[3.3. Zasada działania replikacji transakcyjnej 11](#_Toc312766409)

[3.4. Konfiguracja serwera poprzez SQL Server Management Studio 14](#_Toc312766410)

[4. Analiza wymagań 15](#_Toc312766411)

[4.1. Wymagania funkcjonalne 15](#_Toc312766412)

[4.2. Wymagania niefunkcjonalne 15](#_Toc312766413)

[4.3. Założenia projektowe 15](#_Toc312766414)

[5. Projekt systemu 16](#_Toc312766415)

[5.1. Schemat logiczny i koncepcja działania systemu 16](#_Toc312766416)

[5.2. Diagramy i scenariusz przypadków użycia 16](#_Toc312766417)

[5.3. Projekt rozproszonej bazy danych 16](#_Toc312766418)

[5.3.1. Struktura tabel i relacji 16](#_Toc312766419)

[5.3.2. Model konceptualny – diagram związków encji 16](#_Toc312766420)

[5.3.3. Model fizyczny bazy danych w środowisku MS SQL 16](#_Toc312766421)

[5.4. Projekt mechanizmu replikacji transakcyjnej 16](#_Toc312766422)

[5.5. Projekt aplikacji klienckich 16](#_Toc312766423)

[5.5.1. Interfejs operatora 16](#_Toc312766424)

[5.5.2. Interfejs aplikacji klienckiej 16](#_Toc312766425)

[6. Implementacja elementów systemu 17](#_Toc312766426)

[6.1. Realizacja konfiguracji bazy danych 17](#_Toc312766427)

[6.2. Realizacja mechanizmu replikacji 17](#_Toc312766428)

[6.3. Realizacja wybranych funkcjonalności 17](#_Toc312766429)

[6.3.1. Wyszukiwarka połączeń 17](#_Toc312766430)

[6.3.2. Aplikacja operatora kolejowego 17](#_Toc312766431)

[6.3.3. Aplikacja operatora autobusowego 17](#_Toc312766432)

[7. Testowanie systemu 18](#_Toc312766433)

[7.1. Testy funkcjonalne 18](#_Toc312766434)

[7.2. Testy systemu replikacji 18](#_Toc312766435)

[7.3. Wnioski z testów 18](#_Toc312766436)

[8. Podsumowanie 18](#_Toc312766437)

[Literatura 19](#_Toc312766438)

Spis rysunków

[Rysunek 1 MS SQL Server 2008 - porównanie edycji 7](#_Toc312772190)

[Rysunek 2 Replikacja transakcyjna 12](#_Toc312772191)

[Rysunek 3 Uruchomienie kreatora konfiguracji trybu rozproszonego 14](#_Toc312772192)

[Rysunek 4 Kreator konfiguracji trybu rozproszonego 14](#_Toc312772193)

[Rysunek 5 Tworzenie nowej publikacji 15](#_Toc312772194)

[Rysunek 6 Wybór trybu replikacji 15](#_Toc312772195)

[Rysunek 7 Tworzenie subskrypcji 16](#_Toc312772196)

[Rysunek 8 Wybór publikatora 16](#_Toc312772197)

[Rysunek 9 Przykładowa konfiguracja MS Management Studio 17](#_Toc312772198)

[Rysunek 10 Uruchamianie Monitora Replikacji 17](#_Toc312772199)

[Rysunek 11 Replication Monitor 18](#_Toc312772200)

Spis tabel

## Wstęp

Tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst

## Cel i zakres pracy

## Cel projektu

Tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst

## Zakres projektu

Tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst tekst

## Replikacja w systemie baz danych MS SQL Server 2008

MS SQL Server 2008 udostępnia mechanizmy replikacji (zbiór technologii umożliwiających kopiowanie i rozprowadzanie danych oraz obiektów bazodanowych pomiędzy różnymi bazami danych) na zróżnicowanym poziomie w zależności od rodzaju edycji oprogramowania. Pełną funkcjonalność (wraz z obsługą baz danych Oracle) zapewniają edycje Enterprise oraz Datacenter. Darmowa edycja Express zapewnia jedynie podstawowe mechanizmy replikacji w trybie subskrybenta. Szczegółowe różnice pomiędzy konkretnymi edycjami przedstawia rysunek:  
  


Rysunek 1 MS SQL Server 2008 - porównanie edycji

## Przegląd typów replikacji

Microsoft SQL Server zapewnia trzy typy replikacji:

* replikację transakcyjną (*Transactional replication*)
* replikację scalającą (*Merge replication*)
* replikację migawkową (*Snapshot replication*)

Wybór odpowiedniego typu replikacji jest zależny od wielu czynników, na przykład:

* Fizyczna struktura systemu: ilość oraz lokalizacja komputerów wykorzystywanych przy replikacji, ich rodzaj (czy są to komputery stacjonarne, czy np. urządzenia mobilne)
* Ilość i rodzaj danych do replikacji
* Tryb pracy subskrybenta: pasywny lub z możliwością aktualizacji danych

Ponadto, rodzaj replikacji zależy od zakładanej funkcjonalności, jaką chce osiągnąć projektant systemu. Przykładowe scenariusze zastosowań:

* Poprawa skalowalności i dostępności: utrzymywanie nieprzerwanie aktualizowanych kopii danych umożliwia przeniesienie operacji odczytu danych na wiele serwerów. Nadmiarowość wynikająca z takiego podejścia jest istotna podczas planowanych oraz awaryjnych operacji administracyjnych.
* Hurtownie danych: replikacja wykorzystywana jest do transferu danych pomiędzy serwerami OLTP (*online transaction processing*), a serwerami wspierającymi podejmowanie decyzji.
* **Integracja danych z wielu źródeł**: zbieranie danych z wielu serwisów i prezentowanie ich w ujednoliconej formie.
* Integracja danych z różnych systemów bazodanowych: umożliwienie aplikacjom pobieranie danych z systemów innych niż MS SQL np. z baz danych Oracle.
* Wymiana danych pomiędzy użytkownikami urządzeń mobilnych

Biorąc pod uwagę powyższe czynniki, bazę danych można skonfigurować do pracy z konkretnym typem replikacji:

**Replikacja migawkowa:**

Ten typ replikacji rozprowadza dane w ustalonych chwilach czasowych bez monitorowania późniejszych zmian tych danych. W trakcie wystąpienia zdarzenia synchronizacji, generowana jest cała migawka i wysłana do subskrybentów. Replikacja migawkowa może być wykorzystywana jako samodzielny mechanizm replikacji, jednak często jest częścią innych typów replikacji, zapewniając wstępną synchronizację danych.

Wykorzystanie samodzielnej replikacji migawkowej jest najbardziej korzystne, gdy system charakteryzuje się następującymi cechami:

* Dane są zmieniane rzadko
* Można zaakceptować, że zreplikowana kopia danych jest nieaktualna przez pewien okres czasu (do następnej migawki)
* Replikacji podlegają niewielkie ilości danych
* Duże zmiany występują w krótkim okresie czasu

Replikacja migawkowa charakteryzuje się mniejszym narzutem pod względem wykorzystania zasobów (po stronie serwera publikującego) niż replikacja transakcyjna, ze względu na brak śledzenia zmian w sposób przyrostowy. Jednak, gdy zbiór danych do replikacji jest bardzo duży, serwer wykorzystujący ten typ replikacji, musi zapewnić odpowiednie zasoby do generacji i zastosowania migawki.

**Replikacja transakcyjna:**

Replikacja transakcyjna, w typowym przypadku, rozpoczyna się od wykonania migawki początkowej danych oraz obiektów bazodanowych. Po wykonaniu takiej inicjalizacji, wszystkie kolejne zmiany danych oraz modyfikacje struktury bazy zachodzące po stronie Publikatora przekazywane są natychmiastowo (zazwyczaj w czasie rzeczywistym) do Subskrybentów. Zmiany aplikowane Subskrybentom zachodzą w tej samej kolejności w jakiej wystąpiły po stronie Publikatora, co gwarantuje spójność wykonywanych transakcji.

Replikacja transakcyjna wykorzystywana jest głównie w następujących przypadkach:

* Wymagane jest przyrostowe śledzenie zmian w momencie ich występowania
* Wymagane jest małe opóźnienie pomiędzy wystąpieniem zmiany u Publikatora a aktualizacją Subskrybentów
* Aplikacja wymaga dostępu do pośrednich stanów w jakich znajdują się przesyłane dane. Na przykład, podczas kilkukrotnej zmiany danych w wierszu tabeli, replikacja transakcyjna pozwala aplikacji odpowiedzieć (np. poprzez uruchomienie odpowiedniego triggera) na każdą z tych zmian.
* Publikator wykonuję bardzo często operacje wstawiania i aktualizacji danych (*INSERT*, *UPDATE*)
* Wymagana jest komunikacja z serwerami innych producentów (np. Oracle)

W typowym zastosowaniu, replikacja transakcyjna działa w sposób jednokierunkowy (Subskrybent nie może zmienić danych u Publikatora). MS SQL Server udostępnia jednak specjalny tryb pracy replikacji transakcyjnej, umożliwiający komunikację dwukierunkową.

**Replikacja scalająca:**

Tak samo jak w przypadku replikacji transakcyjnej, replikacja scalająca zwykle rozpoczyna się od wykonania migawki inicjalizującej. Kolejne zmiany danych lub schematu bazy danych są śledzone za pomocą wyzwalaczy (triggerów). Subskrybent synchronizuje się z Publikatorem poprzez wymianę wszystkich danych, które zmieniły się od czasu ostatniej synchronizacji.

Typowe sytuacje w których stosowana jest replikacja scalająca:

* Wielu Subskrybentów może zaktualizować dane w różnym czasie i rozprowadzić je do Publikatora i innych Subskrybentów.
* Subskrybent otrzymuje dane, wykonuje zmiany w trybie offline, a następnie w późniejszym czasie synchronizuje się z Publikatorem i innymi Subskrybentami.
* Każdy z Subskrybentów wymaga różnej ilości danych
* Mogą wystąpić konflikty i wymagana jest możliwość ich detekcji i naprawy.
* Aplikacja wymaga zmian całej grupy danych, bez znajomości ich stanów pośrednich. Przykładowo, gdy Publikator zaktualizuje jeden wiersz tabeli kilkukrotnie przed wykonaniem synchronizacji, Subskrybent otrzyma tylko ostateczną wersję danych.

Replikacja scalająca pozwala różnym stronom na autonomiczną pracę, a następnie na późniejsze scalenie aktualizacji w jeden, spójny rezultat. Ponieważ zmiany są wykonywane w kilku „węzłach” systemu, ta sama zmiana musi być zaaplikowana zarówno przez Publikatora jak i u wielu Subskrybentów. Prowadzi to do występowania konfliktów i niejednoznaczności. Replikacja scalająca w MS SQL Server 2008 dostarcza wielu narzędzi do rozwiązywania tego typu problemów.

## Przegląd programów (pośredników) wykorzystywanych przez MS SQL

Mechanizmy replikacji wykorzystują podczas swojej pracy dodatkowe, samodzielne programy pomocnicze, nazywane również „pośrednikami”. Programy te związane są z zadaniami odpowiadającymi za śledzenie zmian i dystrybucję danych. Domyślnie, pośrednicy pracują jako zadania uruchomione przez „SQL Agent Server”. Pośrednicy mogą być uruchomieni również z poziomu linii poleceń lub przez aplikacje korzystające z RMO (Replication Management Objects), na przykład SQL Server Replication Monitor i SQL Server Management Studio.

**SQL Server Agent**:

SQL Server Agent hostuje i harmonogramuje pracę innych programów pośredniczących. Program ten również steruje i monitoruje operacjami nie związanymi z samym procesem replikacji. Po instalacji serwera, domyślnie SQL Server Agent jest wyłączony i wymaga jawnego określenia trybu jego uruchamiania.

**Snapshot Agent**:

Program ten wykorzystywany jest zazwyczaj przez wszystkie typy replikacji. Przygotowuje schematy, początkowe struktury danych oraz obiekty bazodanowe, a następnie zapisuje je w postaci migawek. Zarządza również informacjami o synchronizacji wewnątrz bazy danych odpowiedzialnej za dystrybucję.

**Log Reader Agent:**

Log Reader Agent ma zastosowanie przy replikacji transakcyjnej. Przenosi on (z logów Publikatora) transakcje oznaczone jako rozproszone, do bazy odpowiedzialnej za dystrybucję danych. Każda baza danych publikująca dane z wykorzystaniem replikacji transakcyjnej wykorzystuje własną instancję programu Log Reader Agent.

**Distribution Agent:**

Pośrednik ten wykorzystywany jest przez replikację migawkową oraz replikację transakcyjną. Aplikuje on początkową migawkę Subskrybentom, oraz przenosi transakcje znajdujące się w bazie dystrybucyjnej. Distribution Agent uruchomiony jest zarówno przez Dystrybutora do wysyłania subskrypcji, jak i przez Subskrybentów do ich pobierania.

**Merge Agent:**

Merge Agent jest używany przez replikację scalającą. Wykonuje on początkową migawkę do Subskrybenta, oraz przenosi uzgodnione wersje przyrostowych zmian danych. Każda subskrypcja wykorzystująca replikację transakcyjną korzysta z oddzielnej instancji tego programu. Łączy on zarówno Publikatora jak i Subskrybenta oraz aktualizuje obydwie strony replikacji. Merge Agent jest uruchomiony zarówno na Dystrybutorze do wysyłania subskrypcji, jak i na Subskrybencie do ich pobierania. Domyślnie Merge Agent wysyła zmiany od Subskrybenta do Publikanta, a następnie pobiera zmiany w odwrotnym kierunku.

**Queue Reader Agent:**

Program ten jest wykorzystywany przez replikację transakcyjną z włączoną opcją aktualizacji kolejkowej. Jest on uruchomiony na serwerze Dystrybutora i przenosi zmiany wykonane przez Subskrybenta z powrotem do Publikatora. W przeciwieństwie do programów Distribution Agent oraz Merge Agent, występuje tylko jedna instancja tego pośrednika dla danej rozporoszonej bazy danych.

## Zasada działania replikacji transakcyjnej

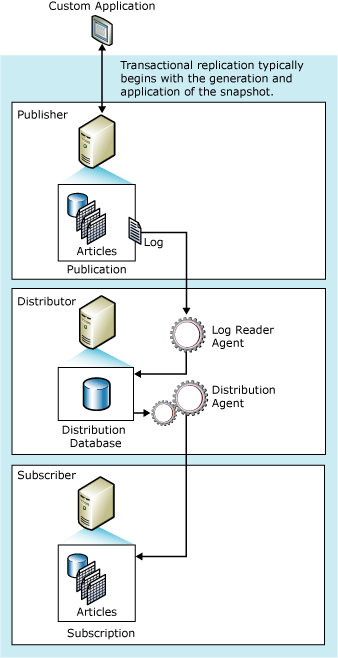
Replikacja transakcyjna jest zaimplementowana z wykorzystaniem programów pośredniczących: SQL Server Snapshot Agent, Log Reader Agent oraz Distribution Agent.

Snapshot Agent przygotowuje pliki z migawkami zwierające schemat oraz dane publikowanych tabel oraz obiektów bazodanowych. Zapisuje te informacje w specjalnym folderze przeznaczonym do przechowywania migawek i zapisuje zadania synchronizacyjne w bazie danych Dystrybutora.

Log Read Agent monitoruje logi transakcji w każdej bazie danych skonfigurowanej do wykonywania replikacji transakcyjnej. Kopiuje transakcje zaznaczone jako rozproszone z logów transakcyjnych do bazy dystrybucyjnej w formie kolejki typu „store-and-forward”. Następnie, Distribution Agent aplikuje początkową migawkę, oraz przenosi transakcje znajdujące się w bazie dystrybucyjnej do Subskrybentów.

Przyrostowe zmiany wykonane po stornie Publikatora przepływają do Subskrybenta zgodnie z harmonogramem Pośrednika Dystrybucji (Distribution Agent). Może on być uruchomiony w trybie pracy ciągłej (aby uzyskać minimalne opóźnienia), lub w trybie okresowych synchronizacji. Ponieważ wszystkie zmiany danych występują jedynie po stronie Publikatora (w przypadku replikacji transakcyjnej jednokierunkowej) – nie występują konflikty. Ostatecznie, wszyscy Subskrybenci otrzymują te same wartości, które znajdują się u Publikatora.

Podstawowe komponenty replikacji transakcyjnej:



Rysunek 2 Replikacja transakcyjna

**Inicjalizacja:**

Zanim nowy Subskrybent replikacji transakcyjnej będzie mógł otrzymać przyrostowe zmiany od Publikatora, musi zostać utworzona w nim struktura bazy danych, odpowiadająca subskrybowanym obiektom. Wykonywane jest to automatycznie, zazwyczaj podczas początkowej migawki wykonanej przez pośrednika Snapshot Agent oraz rozprowadzonej i zastosowanej za pomocą pośrednika Distribution Agent. Początkowy zbiór danych i struktura bazy może zostać również uzyskana w inny sposób, na przykład z kopii zapasowej lub przez oprogramowanie SQL Server Integration Services.

Aplikowanie początkowej migawki obejmuje jedynie te subskrypcje, które są w stanie oczekiwania na migawkę inicjalizującą. Pozostałe subskrypcje (którym dostarczono migawki) są pomijane w procesie inicjalizacji.

W przeciwieństwie do replikacji migawkowej, w której podczas generowania migawek zakładane są blokady na wszystkie publikowane tabele, w replikacji transakcyjnej domyślnym systemem jest współbieżne przetwarzanie migawek. Oznacza to, że użytkownik może wykonywać pracę nieprzerwanie podczas tworzenia migawek inicjalizujących.

**Modyfikacja danych:**

Za modyfikację danych odpowiedzialny jest Log Reader Agent. Uruchomiony jest on na serwerze dystrybucyjnym i zazwyczaj pracuje w trybie pracy ciągłej. Pośrednik ten, najpierw przegląda (odczytuje) logi transakcji publikatora, oraz identyfikuje wszystkie operacje typu INSERT, UPDATE, DELETE lub inne modyfikacje danych wykonane w transakcjach zaznaczonych jako rozproszone. Następnie, pośrednik ten kopiuje każdy pakiet poleceń do Dystrybutora. Log Read Agent wykorzystuje wewnętrzną procedurę składowaną *sp\_replcmd*, aby przejść do kolejnego zbioru poleceń oznaczonych jako rozproszone. Baza dystrybutora wykorzystywana jest następnie jako kolejka typu „store-and-forward” z której zmiany wysyłane są do Subskrybentów. Tylko zatwierdzone transakcje są wysyłane do bazy dystrybucyjnej.

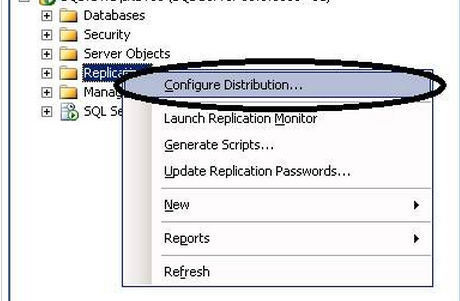
Po poprawnym zapisaniu całego pakietu transakcji rozproszonych do bazy dystrybucyjnej, są one zatwierdzane (commit). Po zatwierdzeniu każdego pakietu poleceń przez Dystrybutora, Log Reader Agent wywołuje procedurę *sp\_repldone* aby zaznaczyć w którym miejscu replikacja została zakończona. Ostatecznie, pośrednik zaznacza te wiersze w logu transakcji, które zostały pomyślnie przetworzone. Wiersze wciąż nie przetworzone nie zostaną oczyszczone.

Komendy transakcji są przechowywane w bazie dystrybutora do momentu, aż zostaną pomyślnie rozprowadzone do wszystkich subskrybentów, lub gdy minie maksymalny ustalony czas wykonania operacji. Subskrybenci otrzymują transakcje w tej samej kolejności, w jakiej były zastosowane po stronie Publikatora.

## Konfiguracja serwera poprzez SQL Server Management Studio

SQL Management Studio to zintegrowane środowisko umożliwiające zarządzanie, administrowanie oraz rozwijanie wszystkich komponentów serwera MS SQL. SQL Management Studio udostępnia te możliwości za pomocą graficznych narzędzi oraz licznych skryptów.

Aby skonfigurować serwer do pracy w trybie rozproszonym, należy wykonać kilka podstawowych operacji. Pierwszą czynnością, jest uruchomienie kreatora konfiguracji mechanizmów replikacji:



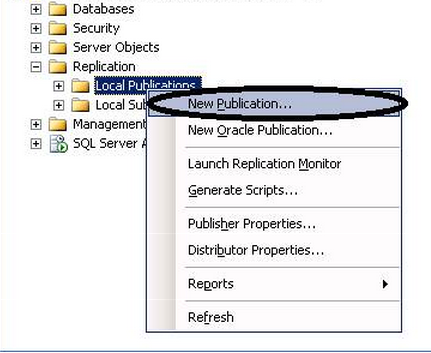
Rysunek 3 Uruchomienie kreatora konfiguracji trybu rozproszonego



Rysunek 4 Kreator konfiguracji trybu rozproszonego

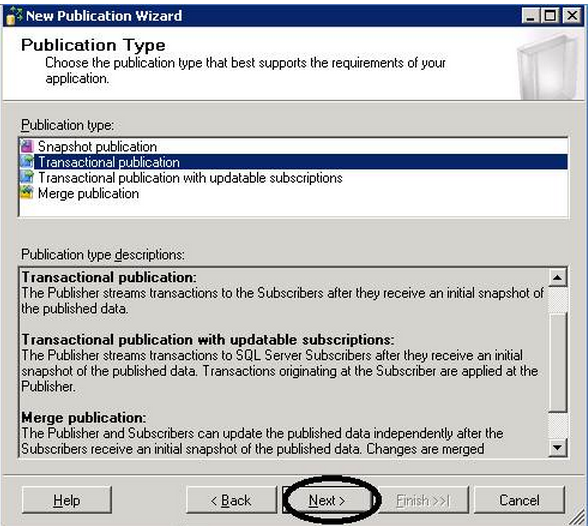
W trakcie pracy kreatora, możemy wskazać odpowiedni folder do przechowywania migawek, wskazać serwer Dystrybutora (w szczególności może być to ten sam serwer na którym działa Publikator) oraz wybrać tryb uruchamiania pośrednia SQL Server Agent.

Następnym krokiem, jest utworzenie nowej publikacji:



Rysunek 5 Tworzenie nowej publikacji

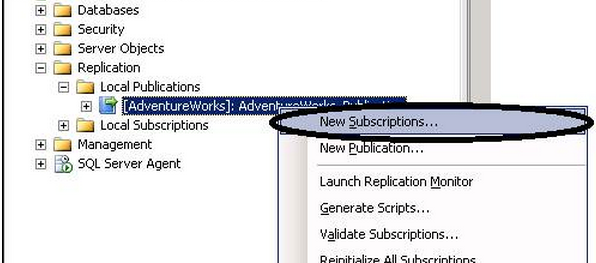
Po kliknięciu w zaznaczoną opcję, uruchomi się kolejny kreator. W trakcie jego działania, możemy m.in. określić rodzaj replikacji:



Rysunek 6 Wybór trybu replikacji

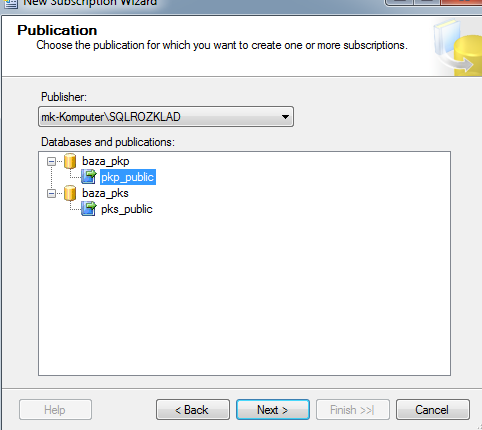
Możemy również wskazać, które dane chcemy publikować (wraz z definicją odpowiednich filtrów), określić tryb pracy pośrednika migawek, oraz wprowadzić dane potrzebne do autoryzacji użytkowników.

Po utworzeniu publikacji, można zacząć konfigurować subskrypcje, które będą z nich korzystały:



Rysunek 7 Tworzenie subskrypcji

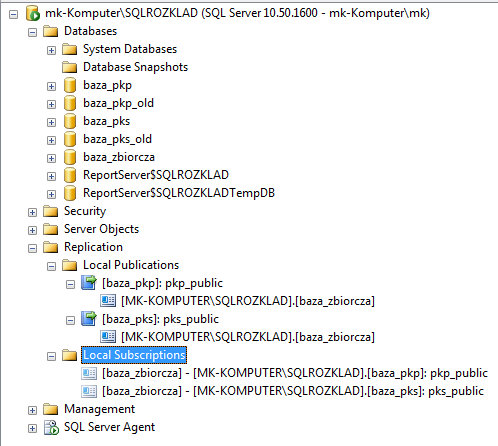
W tym celu ponownie korzystamy z kreatora. Możemy określić bazę danych z której będziemy subskrybować:



Rysunek 8 Wybór publikatora

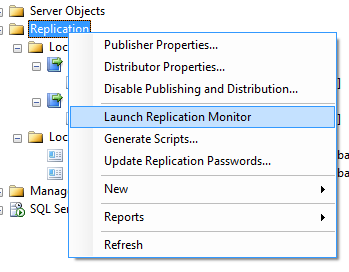
Kreator tworzenia subskrypcji umożliwia również podanie ścieżek dostępu do folderów przeznaczonych na logi, ustawienie odpowiedniego trybu pracy Dystrybutora oraz wybranie metod autoryzacji.

Po przykładowej konfiguracji, eksplorator obiektów MS Management Studio wygląda następująco:



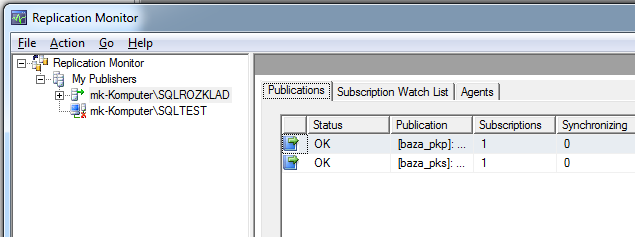
Rysunek 9 Przykładowa konfiguracja MS Management Studio

Ważnym narzędziem udostępnianym wraz z MS Management Studio jest Monitor Replikacji (Replication Monitor):



Rysunek 10 Uruchamianie Monitora Replikacji

Umożliwia on podgląd stanu działania wszystkich replikacji uruchomionych w systemie. Zawiera również informacje o stanie uruchomionych pośredników oraz przeglądarkę logów:



Rysunek 11 Replication Monitor

## Analiza wymagań

## Wymagania funkcjonalne

## Wymagania niefunkcjonalne

## Założenia projektowe

## Projekt systemu

## Schemat logiczny i koncepcja działania systemu

## Diagramy i scenariusz przypadków użycia

## Projekt rozproszonej bazy danych

## Struktura tabel i relacji

## Model konceptualny – diagram związków encji

## Model fizyczny bazy danych w środowisku MS SQL

## Projekt mechanizmu replikacji transakcyjnej

## Projekt aplikacji klienckich

## Interfejs operatora

## Interfejs aplikacji klienckiej

## Implementacja elementów systemu

## Realizacja konfiguracji bazy danych

## Realizacja mechanizmu replikacji

## Realizacja wybranych funkcjonalności

## Wyszukiwarka połączeń

## Aplikacja operatora kolejowego

## Aplikacja operatora autobusowego

## Testowanie systemu

## Testy funkcjonalne

## Testy systemu replikacji

## Wnioski z testów

## Podsumowanie

## Literatura