



# Урок 1

## План заняття:

1. СУБД MS SQL Server 2005/2008. Переваги та недоліки. Огляд версій та редакцій
2. Навчальний курс Microsoft по MS SQL Server
3. Інсталяція MS SQL Server 2005 Express Edition/2008
  - 3.1. Інсталяція MS SQL Server 2005 Express Edition
  - 3.2. Інсталяція MS SQL Server 2008 Enterprise
4. Утиліти SQL Server 2005/2008
  - 4.1. Загальна інформація про утиліти
  - 4.2. SQL Server Configuration Manager
  - 4.3. SQL Server Management Studio
  - 4.4. SQL Server Business Intelligence Development Studio
5. Робота з базою даних через утиліти та зберігаємі процедури
  - 5.1. Створення БД
  - 5.2. Зміна властивостей БД
  - 5.3. Переіменування БД
  - 5.4. Видалення БД
6. Введення в SQL. Що це таке? Категорії команд SQL: DDL, DML, DCL
7. Створення бази даних засобами SQL
8. Модифікація та видалення бази даних
9. Імпортування даних
10. Домашнє завдання

## 1. СУБД MS SQL Server 2005/2008. Переваги та недоліки. Огляд версій та редакцій

Сьогодні ми з вами розпочинаємо вивчення ще однієї планети з галактики «Бази Даних» – **MS SQL Server**. На даний час ви вже знаєте, що таке база даних та СУБД, які існують моделі баз даних, як здійснюється нормалізація, що являє собою первинний та зовнішній ключі тощо. Отже, в даному курсі вам достатньо буде сконцентрувати свою увагу на особливостях роботи SQL в середовищі СУБД MS SQL Server. Та перш, ніж перейти до вивчення її основ, розглянемо коротко історію її створення.

Первинно MS SQL Server розроблялась компанією **Microsoft** сумісно з **Sybase Corporation** для використання на платформах IBM OS/2. Весь код випущеного близько 1989 року SQL Server 1.0 (аж до версії 7.0), був оснований на коді Sybase SQL Server, що дозволило йому вийти на ринок баз даних. Коли Microsoft та IBM розділились на окремі компанії, Microsoft відмовилась від OS/2 та створила свою власну операційну систему - Windows NT Advanced Server 3.1. Її реліз майже співпав з офіційним релізом Microsoft SQL Server версії 4.21 для цієї ж ОС (1993 рік). З цього моменту було прийнято рішення розробляти SQL Server лише для середовища Windows NT і у 1995 р. Microsoft випускає **SQL Server 6.0**. Після цього Microsoft почали добиватись повних прав на всі версії SQL Server для Windows. Такі кроки були спричинені тим, що на цей час корпорація Sybase, щоб не стояти осторонь, випустила власну СУБД під Windows NT і назвала її **Sybase Adaptive Server Enterprise** (для уникнення путанини з Microsoft SQL Server).

Розвиток SQL Server не припинився і доволі скоро в світ виходять наступні версії:

- 1996 р. - SQL Server 6.5, кодова назва Hydra;
- 1999 р. - SQL Server 7.0, кодова назва Sphinx;
- 1999 р. - SQL Server 7.0 OLAP, кодова назва Plato;
- 2000 р. - SQL Server 2000 32-bit, кодова назва Shiloh (версія 8.0);
- 2003 р. - SQL Server 2000 64-bit, кодова назва Liberty;
- 2005 р. - SQL Server 2005, кодова назва Yukon (версія 9.0);
- 2008 р. - SQL Server 2008, кодова назва Katmai (версія 10.0).

Весь код SQL Server, починаючи з версії 7.0 був переписаний, щоб уникнути претензій з боку Sybase на авторські права. Сам MS SQL Server 7.0 став першим сервером баз даних з графічним інтерфейсом користувача для адміністрування.

Випуск версії SQL Server 2005 проходив паралельно випуску Visual Studio 2005 і тому його урізану версію також включили до інсталяції останньої. В новій версії, в порівнянні з попередніми, змінився інтерфейс користувача і адміністратора, частина утиліт об'єднані. Але суттєві зміни в основному стосувались реалізації технології ETL (витягування, перетворення і завантаження даних), які входять до складу компонента SQL Server Integration Services (SSIS), сервера оповіщення, засобів аналітичної обробки багатовимірних моделей даних (OLAP) і збору релевантної інформації



(обидві служби входять до складу Microsoft Analysis Services), а також кількох служб оповіщення, а саме Service Broker і Notification Services. Крім того, були зроблені суттєві покращення в продуктивності.

Остання версія SQL Server - SQL Server 2008, яка була випущена в серпні 2008 року, направлена на те, щоб зробити управління даними самоналаштуваним, самоорганізуючим і самообслуговуючим механізмом. Це дозволить зменшити до нуля час знаходження сервера в неробочому стані (простої). Для реалізації цих можливостей була створена технологія SQL Server Always On.

Для підвищення ефективності адміністрування в SQL Server була включені нові інструменти, які дозволяють контролювати кілька серверів баз даних, а також бібліотека Declarative Management Framework, яка дозволяє розподіляти повноваження для баз даних або окремих таблиць.

Крім того, були додані **наступні можливості**:

- Підтримка структурованих і частково структурованих даних, включаючи цифрові формати для зображень, звуків, відео та інших типів мультимедіа. Можливість використання мультимедіа форматів всередині СУБД дозволила спеціалізованим функціям взаємодіяти з цими типами даних.
- Включені ряд додаткових нових типів даних.
- Покращені методи компресії даних.
- Покращена робота з повнотекстовими індексами.
- Покращена багатопоточність деяких операцій, наприклад, створення великих індексів.
- Вбудовані процедури на мові .NET.
- Покращений інтерфейс управління і відлагодження (Management Studio).

Корпорація Microsoft випускає СУБД SQL Server в різних редакціях (реалізаціях). Це було зроблено з метою задоволення вимог широкого діапазону споживачів. Наприклад, для однієї організації необхідно, щоб база даних була дешевою або взагалі безкоштовною та задовольняла мінімальні вимоги по роботі. Для іншої ж організації потрібно, щоб аналітична частина бази даних виконувалась як на великих машинах, так і на персональних або карманих комп'ютерах. В зв'язку з цим основна відмінність різних реалізацій СУБД полягає в їх ефективності, продуктивності і звичайно ж ціні.

**SQL Server 2000** представлена наступними **редакціями**:

1. **Enterprise Edition** – це найбільш повне видання, яке підтримує до 32 процесорів і до 64 Гб ОЗУ. Основні характеристики:
  - масштабуємість до таких рівнів ефективності, які можуть знадобитись при підтримці великих Web-сайтів, промислових OLTP і Data Warehousing систем;
  - ефективність;
  - високий рівень доступності і готовності в критичних умовах (наприклад, при збоях);
  - включає кілька додаткових опцій аналізу, які не включені в SQL Server 2000 Standard Edition.
2. **Standard Edition** – використовується в основному на малих та середніх підприємствах, яким не потрібні високі показники масштабуємість, ефективності, аналітики чи доступності. Підтримує до 4-х процесорів і до 2-х Гб ОЗУ. Standard Edition включає в себе основні функціональні можливості, які необхідні для некритичних бізнес додатків та додатків електронної комерції, джерел даних тощо.
3. **Personal Edition** – урізана реалізація Standard Edition з точки зору апарату адміністрування. Фактично, являється ідеальним варіантом для мобільних користувачів, які проводять частину часу не в мережі, та для автономних прикладних програм, які вимагають локального збереження даних в SQL Server.
4. **Developer Edition** – аналог Enterprise Edition, але орієнтована на розробку (тобто на створення різного роду прикладних додатків) і заборонена для комерційного використання.
5. **Evaluation Edition** (або Enterprise Evaluation Edition) – аналог Developer Edition, але являється тріал-версією, тобто час використання обмежений до 120 днів.
6. **Desktop Engine** – найурізаниша реалізація, але являється ідеальним рішенням для використання в якості автономного джерела даних (розмір БД обмежений до 2 Гб), оскільки легко встановлюється і займає найменше ресурсів серед всіх реалізацій SQL Server. Характерна особливість даної реалізації – це те, що її можна включати в свої прикладні програми, які використовують SQL Server для збереження даних і поширювати їх.
7. **Windows CE Edition** (або SQL Server CE) – забезпечує обмежені можливості управління базами даних на Windows CE управляємих пристроях. Розмір бази даних обмежених до 2 Гб.



**SQL Server 2005** представлена в таких редакціях:

1. **Enterprise Edition** (32- та 64-розрядна версія). Аналогічна по своєму призначенню MS SQL Server 2000 Enterprise Edition;
2. **Standard Edition** (32- та 64-розрядна версія) – аналог аналогічної реалізації нижчої версії сервера;
3. **Developer Edition** (32- та 64-розрядна версія) – аналогічно MS SQL Server 2000 Developer Edition не призначена для комерційного використання, а лише ліцензована як середовище розробки і тестування;
4. **Workgroup Edition** (32-розрядна версія) – призначена для управління даними в невеликих організаціях з обмеженими по розміру та по кількості користувачів базами даних, які можуть обслуговувати невеликі веб-сервери, а також роботу підрозділів і філіалів. Фактично вона призначена для адміністраторів, які працюють на невеликих серверах з невеликими обсягами даних.
5. **Express Edition** (32-розрядна версія) – урізана та безкоштовна версія Microsoft SQL Server. Проста у використанні і управлінні, яка може використовуватись і в якості клієнтської, і в якості серверної бази даних. Працює з невеликими базами даних.

**SQL Server 2008** існує в редакціях:

1. **Enterprise;**
2. **Standard;**
3. **Workgroup;**
4. **Web** – спеціально презначена для створення веб-служб з високим рівнем доступності. Призначена для великих організацій, які в основному орієнтовані на роботу в Інтернет-середовищі.
5. **Compact** – безкоштовна та вбудована база даних SQL Server, яку можна використовувати для розробки автономних або малозв'язаних додатків для мобільних пристроїв (карманих ПК і смартфонів), ПК і веб-клієнтів.
6. **Express** – урізана безкоштовна версія, яка доступна в SQL Server 2008 в трьох варіантах:
  - 1) **SQL Server 2008 Express** – дозволяє створювати, зберігати, обновлювати та обробляти дані.
  - 2) **SQL Server 2008 Express with Tools** – крім основних дій з базою даних, включає в себе середовище SQL Server Management Studio Basic, - візуальний засіб для створення, редагування і управління базами даних.
  - 3) **SQL Server 2008 Express with Advanced Services** – аналог SQL Server 2008 Express with Tools і додатково включає в себе:
    - повнотекстовий пошук – потужний швидкісний засіб для пошуку даних в великих обсягах тексту;
    - службу звітів, яка має своє середовище конструювання.

Отже, **Microsoft SQL Server** – це система управління реляційними базами даних (СУБД), розроблена корпорацією Microsoft. Останні версії представлені розширеним набором реалізацій, які дозволяють кінцевому користувачу обрати необхідну, зекономивши при цьому кошти. Доречі, MS SQL Server для запитів використовує спеціальну різновидність SQL – **Transact-SQL (T-SQL)**, яка була створена сумісно Microsoft і Sybase. Transact-SQL являється реалізацією стандарту ANSI/ISO SQL з рядом розширень.

Оскільки версії MS SQL Server 2000 та 2005 являються на даний час самими поширеними, а найновішою MS SQL Server 2008, то розглянемо основні принципи роботи та інсталяції всіх трьох. Але більший акцент все ж будемо робити на нову версію.

## 2. Навчальний курс Microsoft по MS SQL Server

Система сертифікації Microsoft являється загальноприйнятим стандартом і являється ефективним способом досягнення кар'єрних цілей в середовищі ІТ. Програми сертифікації розроблені майже для кожного продукту Microsoft та для різних категорій користувачів: від ІТ професіоналів, розробників до офісних працівників і домашніх користувачів.

Перш, ніж перейти до системи сертифікації, необхідно ввести ряд понять предметної області. Зокрема, існують такі поняття як **екзамени** і **статуси**. Як правило, для отримання будь-якого статусу необхідно здати кілька екзаменів, але бувають і виключення, тобто для отримання статусу потрібно здати лише один екзамен.

Існує чітко побудована ієрархія статусів (<http://www.microsoft.com/learning/en/us/certification/view-by-name.aspx>) і тому доволі часто для отримання певного статусу необхідно мати набір інших. Базовий статус для ІТ професіонала зараз **MCTS (Microsoft Certified Technology Specialist)**. Даний статус можна отримати по різним областям знань, наприклад, Windows Forms, Web based development тощо. З повним списком MCTS можна ознайомитись по наступній адресі: <http://www.microsoft.com/learning/en/us/certification/mcts.aspx#tab2>.

Статус MCTS можна отримати також і по курсам MS SQL Server, здавши відповідний екзамен:

Статус MCTS	Екзамен
Microsoft SQL Server 2005	Exam 70-431
Microsoft SQL Server 2005 Business Intelligence	Exam 70-445
SQL Server 2008, Business Intelligence Development and Maintenance	Exam 70-448
SQL Server 2008, Database Development	Exam 70-433
SQL Server 2008, Implementation and Maintenance	Exam 70-432



Після отримання якого-небудь з цих статусів можна задуматись і про більш високий статус. В нашому випадку це **MCITP (Microsoft Certified IT Professional)**. Повний список варіантів MCITP можна отримати по адресі: <http://www.microsoft.com/learning/en/us/certification/mcftp.aspx#tab2>.

Для MS SQL Server це будуть наступні екзамени:

Статус MCITP	Екзамен
MCITP: Database Administrator 2008	Exam 70-432 Exam 70-450
MCITP: Database Developer 2008	Exam 70-433 Exam 70-451
MCITP: Business Intelligence Developer 2008	Exam 70-448 Exam 70-452
MCITP: Database Administrator	Exam 70-431 Exam 70-443 Exam 70-444
MCITP: Database Developer	Exam 70-431 Exam 70-441 Exam 70-442
MCITP: Business Intelligence Developer	Exam 70-445 Exam 70-446

Курс, який ми сьогодні з Вами розпочнемо підготує Вас до здачі екзаменів на отримання статусів для MCTS, MCITP: Database Developer, а також допоможе в отриманні початкових знань для сертифікаційного екзамену на отримання статусу MCITP: Database Administrator.

Після статусу MCITP можна отримати статус тренера **MCT (Microsoft Certified Trainer)**. Це людина, яка має право вести авторизовані курси Microsoft в центрах партнерів Microsoft. Для отримання даного статусу необхідно прослухати спеціальний тренінг і продемонструвати педагогічну майстерність перед аудиторією. Нажаль, на сьогоднішній день, даний статус можна отримати лише в Москві.

От і все. Насамкінець можна привести кілька **корисних посилань**:

- Базова сторінка з інформацією о сертифікаціях: <http://www.microsoft.com/learning/en/us/certification/cert-default.aspx>
- Інформація про сертифікації: <http://www.microsoft.com/education/MSITAcademy/curriculum/roadmap/default.aspx>
- Дерево видна діаграма з інформацією про статуси: [http://download.microsoft.com/download/a/0/6/a0675804-40a3-4f34-ac64-6de190da822d/ICT\\_Curriculum\\_Roadmap.pdf](http://download.microsoft.com/download/a/0/6/a0675804-40a3-4f34-ac64-6de190da822d/ICT_Curriculum_Roadmap.pdf)

### 3. Інсталяція MS SQL Server 2005 Express Edition/2008

#### 3.1. Інсталяція MS SQL Server 2005 Express Edition

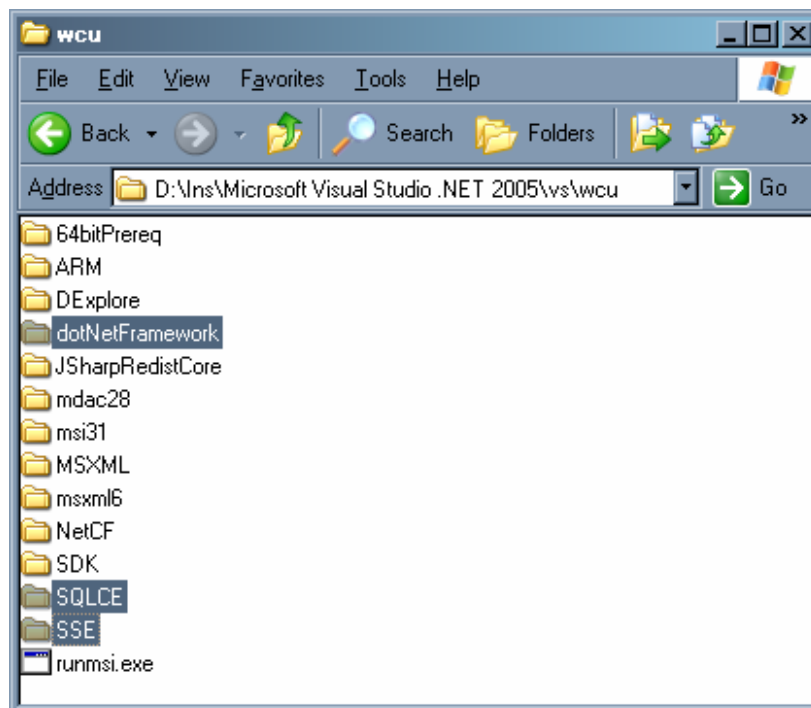
Спочатку розглянемо особливості інсталяції **MS SQL Server 2005 Express Edition**. На перший погляд в інсталяції нічого складного немає, але дана програма має багато налаштувань, які суттєво впливають на роботу сервера баз даних при його використанні.

Дана версія MS SQL Server 2005 поставляється разом з Visual Studio 2005, тому ви маєте можливість при її інсталяції встановити і сам сервер баз даних. Другим варіантом встановлення являється отримання інсталяційного файлу MS SQL Server 2005 Express Edition з офіційного сайту Microsoft, який ви і запускаєте:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=220549b5-0b07-4448-8848-dcc397514b41&displaylang=ru>.

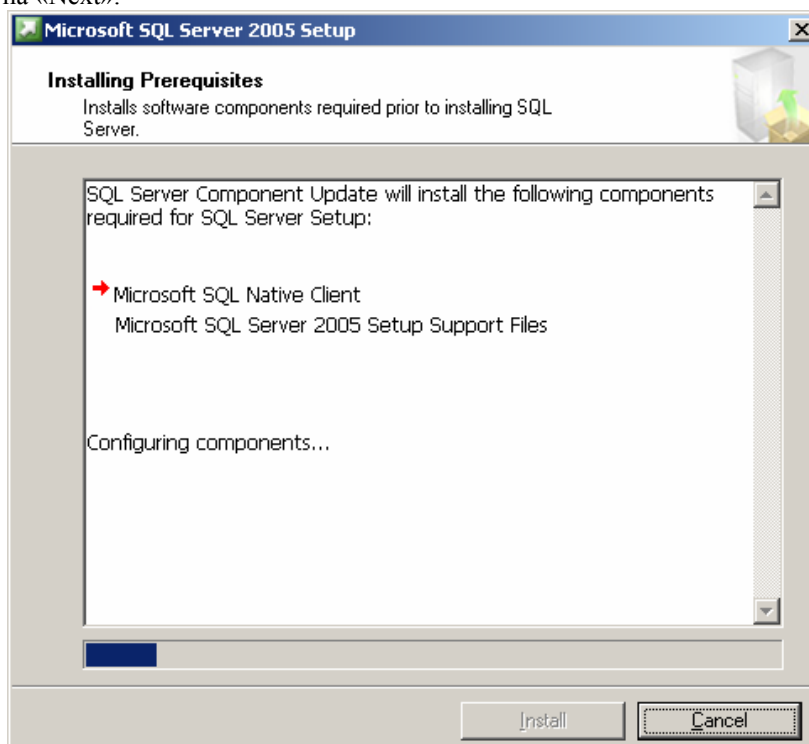
Останній варіант, - це інсталяція СУБД з компонентів, які поставляються разом з Visual Studio 2005. Цей варіант ми і оберемо, оскільки він має ряд особливостей. Але варто відмітити, що процес інсталяції самого сервера баз даних (в другому і третьому випадках), незалежно від вибору не відрізняється.

Отже, файли для встановлення MS SQL Server 2005 Express Edition знаходяться в папці **vs->wcu** на диску, де знаходиться інсталяція Microsoft Visual Studio .NET 2005. Для нормальної роботи сервера баз даних необхідно встановити наступні **компоненти**: dotNetFramework, SQLCE та SSE (власне сам сервер баз даних).

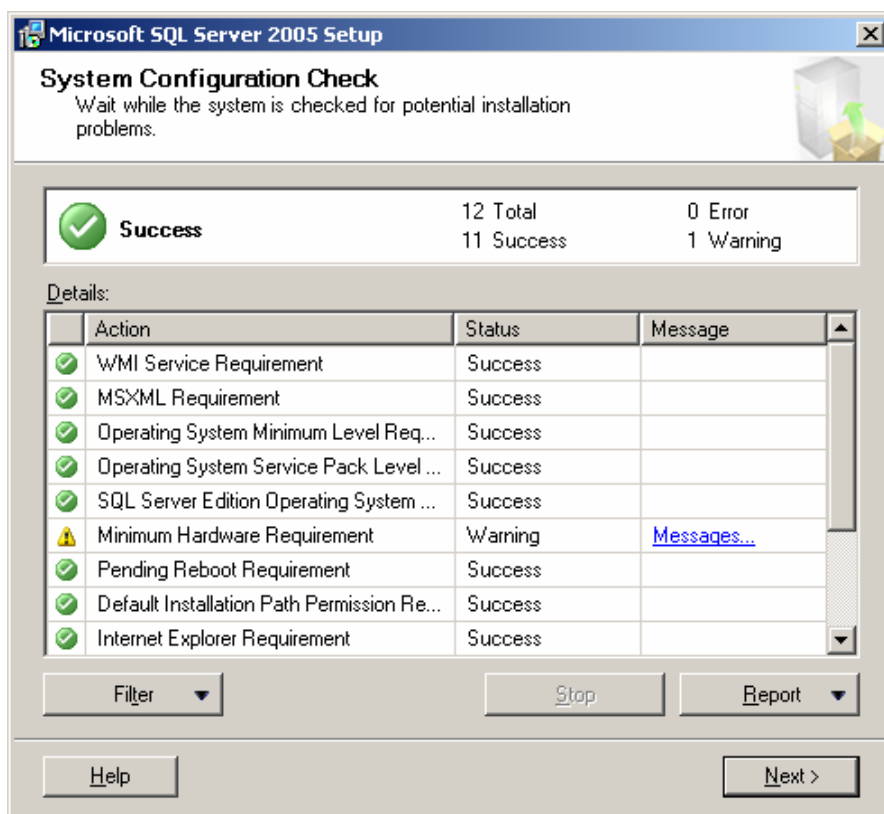


Спочатку встановлюємо необхідні компоненти з папок **dotNetFramework** та **SQLCE**. Доречі, якщо у Вас не встановлений парсер **MSXML6**, тоді зараз настала гарна нагода його становити. Після цього переходимо до безпосереднього встановлення SQL Server 2005 Express з папки **SSE**.

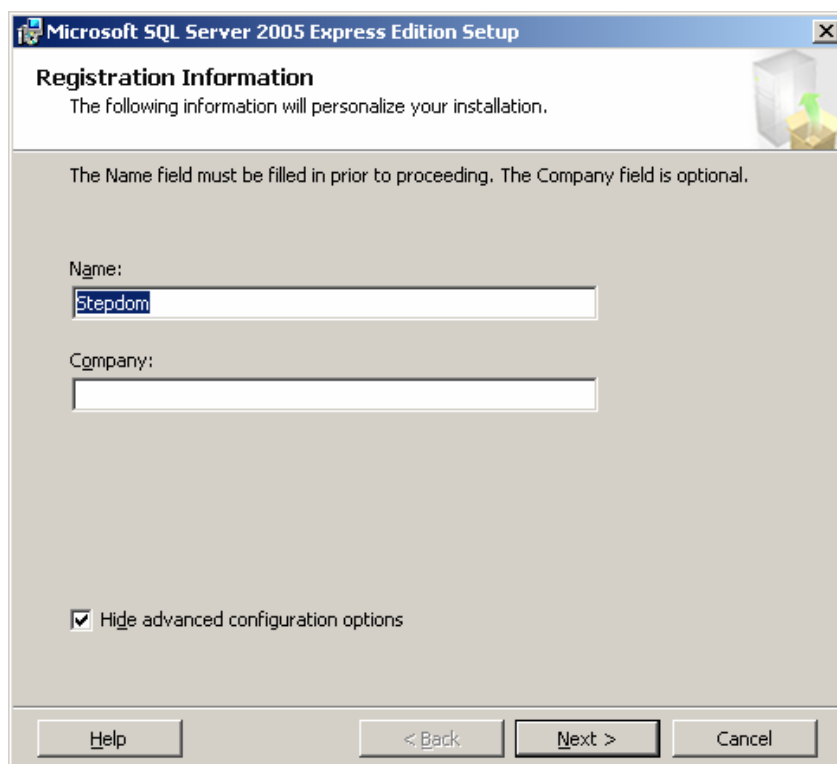
Починається процес інсталяції з знайомства з ліцензійними умовами користувача і, якщо ви з ними порогоджуєтесь, з'явиться вікно з встановленням компонентів, які необхідні для подальшої роботи інсталятора. Для цього натисніть на кнопку «Install», а в кінці на «Next».



Після встановлення цих компонентів запуститься власне сам інсталятор SQL Server. На першому екрані Ви побачите перелік конфігураційних параметрів разом з позначкою про те, який з них призведе до помилкової інсталяції. Наприклад, на нижчеприведеному рисунку можна побачити, що система отримала попередження, яке стосується мінімальних вимог до апаратної частини. Але дане попередження не являється критичним і тому інсталятор дозволяє нам продовжити подальшу роботу.



Наступним кроком є реєстрація Вас як користувача SQL Server, тобто необхідно ввести свої логін та назву компанії (по бажанню). При цьому, якщо флажок «**Hide advanced configuration options**» встановлений, то ці дані будуть приховані під час інсталяції.

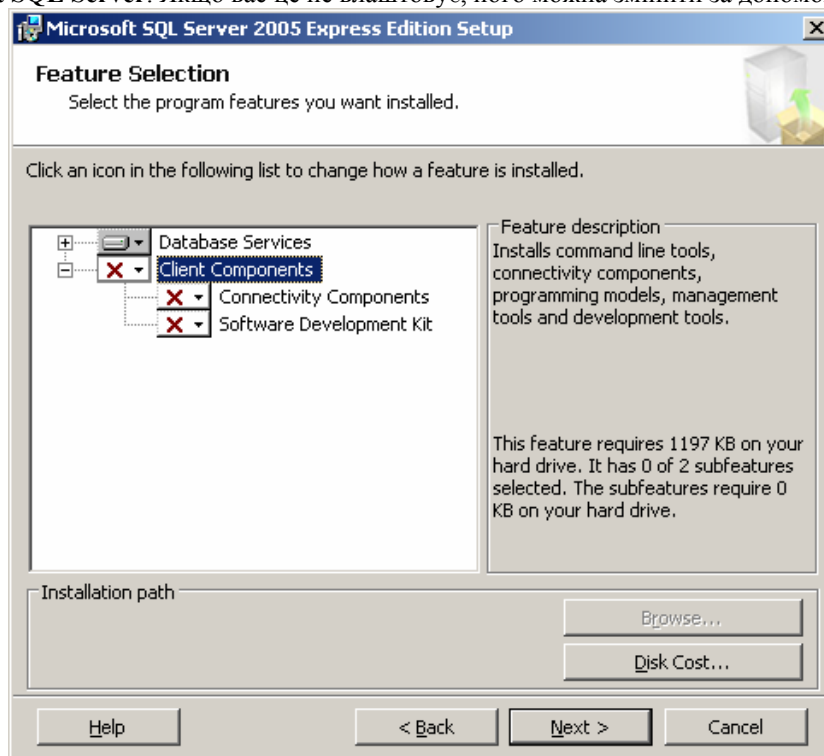


Далі потрібно вказати, що саме Ви збираєтесь ставити на комп'ютер, а саме чи потрібна Вам клієнтська частина чи ні. Якщо Ви клієнт, тоді ставимо клієнтські бібліотеки, інакше вони Вам не потрібні.

Далі, потрібно вказати, які компоненти ви хочете встановити. В нашому випадку вибір невеликий і передбачає лише підтвердження встановлення клієнтської частини (необхідних клієнтських бібліотек). Слід відзначити, що розглядаємий перелік компонентів може відрізнятись, в залежності від повноти версії MS SQL Server 2005 Express. Наприклад, серед компонентів може бути додана можливість встановлення Management Studio Express, - графічної оболонки адміністратора. Нажаль, до нашого файлу інсталяції вона не включена і доведеться її ставити окремо вручну.



На цьому корці також слід обрати майбутнє місцезнаходження сервера баз даних. По замовчуванню таким являється **C:\Program Files\Microsoft SQL Server**. Якщо вас це не влаштовує, його можна змінити за допомогою кнопки «Browse».



Після цього Вам потрібно обрати **варіант інсталяції** – екземпляр по замовчуванню (Default instance) чи іменованний екземпляр (Named instance). Якщо Вам нецікаво знати, що являє собою екземпляр SQL Server, тоді варто пропустити даний та наступний абзаци, обрати екземпляр по замовчуванню (Default instance) і перейти на наступний крок. Для тих, кому цікава більш детальніша інформація, пояснюємо. Отже, існує **два види екземплярів**:

1. **Стандартні екземпляри** – доступ до них ви отримуєте лише за допомогою клієнтського додатку, вказавши при цьому ім'я сервера. Тобто стандартний екземпляр або екземпляр по замовчуванню отримує ім'я сервера, на який його встановлюють. Таким чином, на сервері може бути запущений лише один стандартний екземпляр SQL Server незалежно від версії. Майже всі файли стандартного екземпляру встановлюються в папку **\MSSQL**, а служби мають наступні імена:

- ✓ Служба: SQL Server - назва служби: MSSQLServer
- ✓ Служба: SQL Server Agent - назва служби: MSSQLServerAgent

2. **Іменовані екземпляри** – це екземпляри, яким ви під час встановлення даєте ім'я. Для підключення до іменованого екземпляра Ви вказуєте ім'я сервера та ім'я екземпляру. Одночасно можна запускати кілька іменованих екземплярів SQL Server. Більшість програмних файлів іменованого екземпляру встановлюються в папку **\MSSQL\$InstanceName**, а служби мають наступні імена:

- ✓ Служба: SQL Server - назва служби: MSSQL\$InstanceName
- ✓ Служба: SQL Server Agent - назва служби: MSSQLAgent\$InstanceName

Однак деякі компоненти сумісно використовуються кількома екземплярами SQL Server. Ці компоненти не відносяться ні до стандартних, ні до іменованих. До цих компонентів відносяться ряд інструментів SQL Server і системні засоби.

В SQL Server існує можливість запускати на одному комп'ютері кілька окремих копій, тобто кілька окремих екземплярів сервера баз даних. SQL Server 2005 Express Edition підтримує до 16 іменованих екземплярів, тоді як Enterprise Edition - до 50. Але варто пам'ятати, що використання кількох екземплярів SQL Server збільшує витрати на адміністрування та призводить до дублювання компонентів. Додаткові екземпляри служб SQL Server та SQL Server Agent потребують додаткових ресурсів процесора і пам'яті.

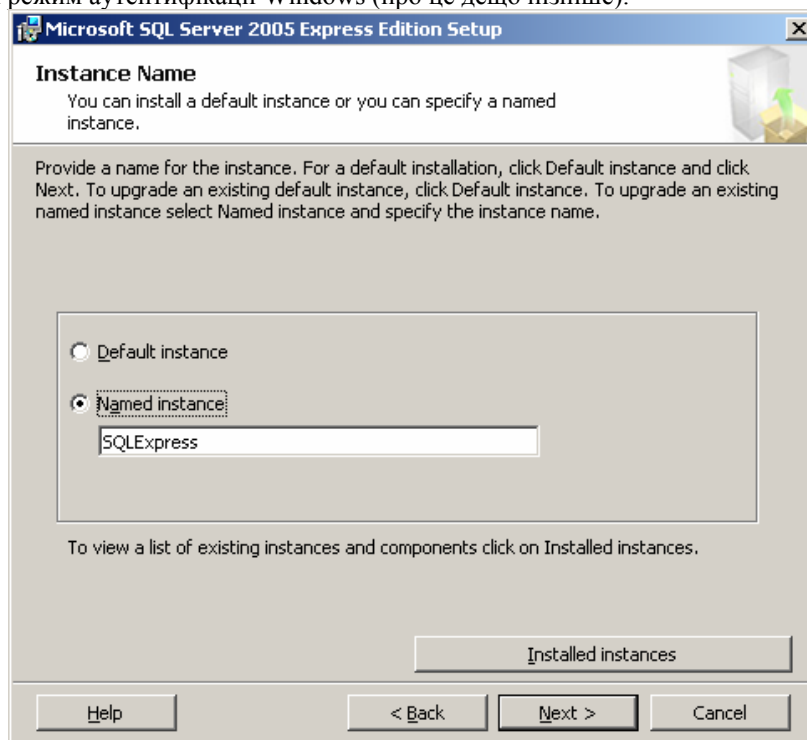
Тим не менше, **в наступних випадках слід віддати перевагу використанню саме кількох екземплярів сервера**:

- при тестуванні різних версій SQL Server на одному комп'ютері;
- при тестуванні пакетів оновлень і розробці баз даних та прикладних додатків;
- коли кілька замовників ПЗ ставлять різні вимоги до розроблюваних систем і користувацьким базам даних, і при цьому необхідно одночасно забезпечити повний набір можливостей адміністрування встановленого у них екземпляра SQL Server;
- коли розробляють прикладні додатки використовують вбудований компонент Desktop Engine, кожен додаток може встановлювати окремий екземпляр, незалежно від екземплярів, встановлених іншими додатками.





Ми з Вами оберемо опцію по замовчуванню, тобто будемо використовувати стандартний екземпляр. Якщо ви оберете іменований, тоді в відповідному полі потрібно вказати ім'я початкового іменованого екземпляру і в ході роботи ви не зможете використовувати режим аутентифікації Windows (про це дещо пізніше).

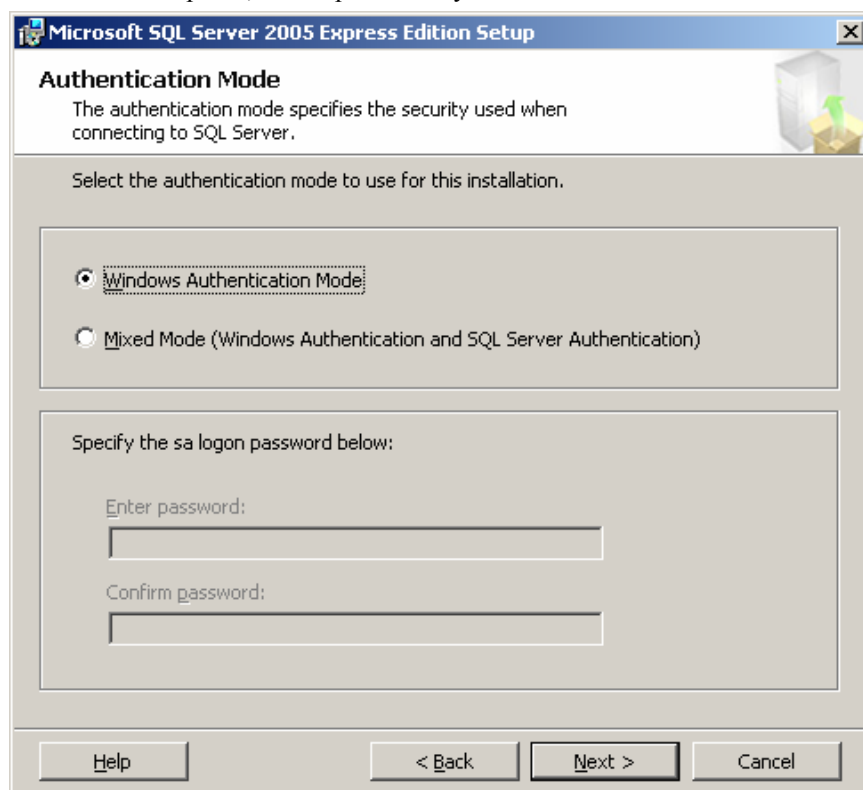


Для забезпечення безпеки вашого SQL Server використовуються облікові записи. Причому бажано кожному обліковому запису встановлювати окремі права.

SQL Server може авторизувати користувача двома способами:

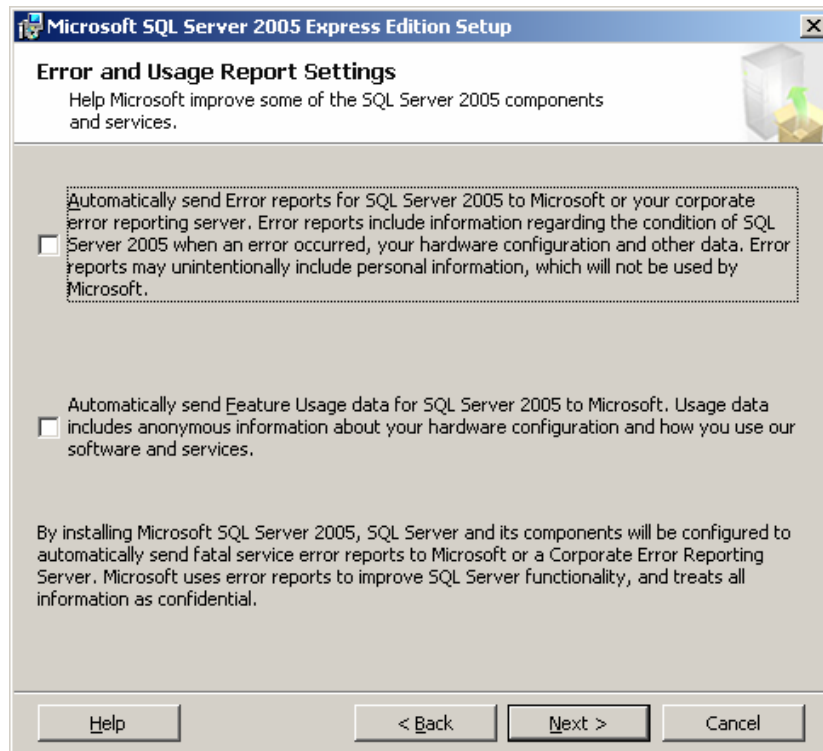
1. **режим аутентификации Windows (Windows Authentication Mode)** - безпосередньо за допомогою Active Directory;
2. **змішаний режим (Mixed Mode)** – за допомогою логінів та паролів, які зберігаються у власній базі даних.

З точки зору безпеки, режим Windows Authentication Mode більш прийнятний, оскільки ви маєте лише єдину базу даних користувачів - Active Directory, за якою потрібно дивитись, а не дві. Але, нажаль, він не завжди підходить. Наприклад, коли ви маєте справу з розробкою прикладних додатків до вашої бази даних, краще обрати змішаний режим доступу. В такому разі (при режимі Mixed Mode) вам обов'язково потрібно встановити складний пароль для користувача «sa» на SQL Server.

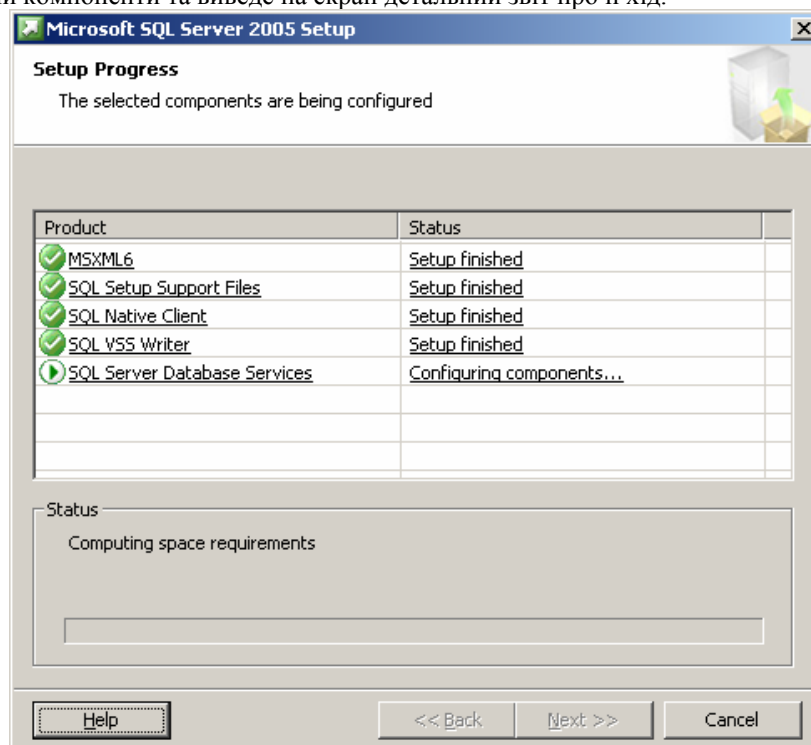


Microsoft доволі давно включила до складу своїх продуктів інструмент повідомлень про помилки і MS SQL Server не став виключенням. Для цього, при інсталяції вводиться окремий крок, в якому потрібно вказати спосіб відправки помилок: посилати дані анонімно і чи посилати взагалі. Жодна з опцій не обрана по замовчуванню і ваша справа чи їх обирати.

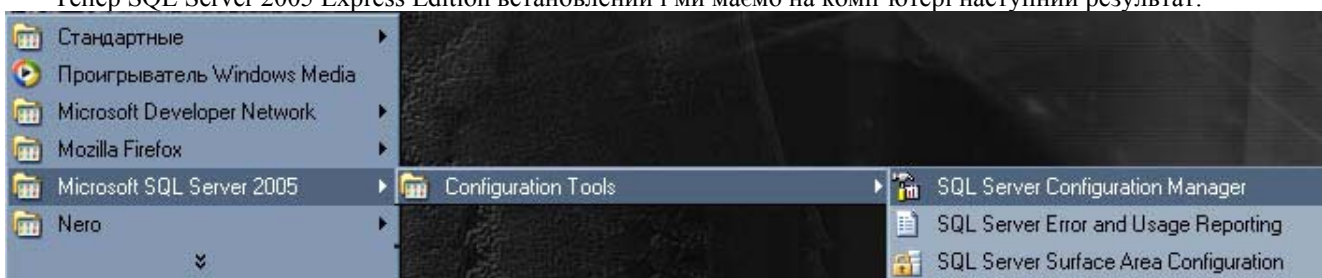




Це все, що ви можете обрати для SQL Server 2005 Express Edition. Після натиснення кнопки «Next» інсталятор встановить всі обрані вами компоненти та виведе на екран детальний звіт про її хід.



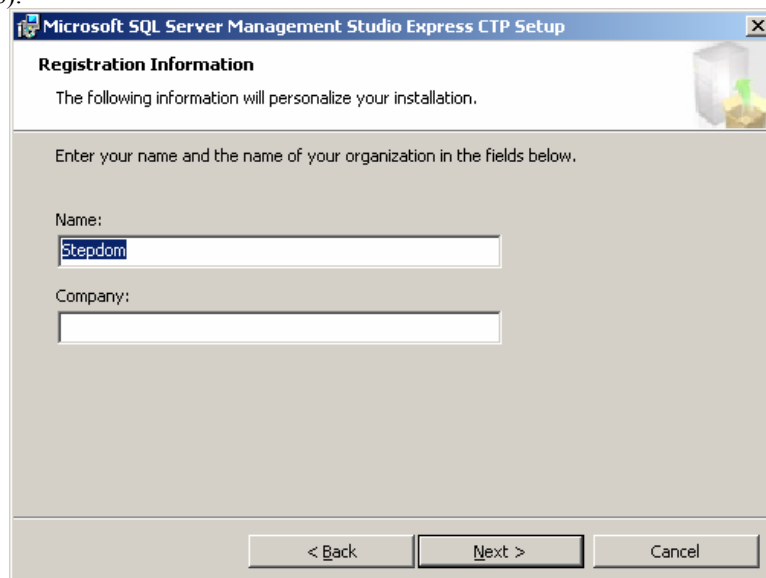
Тепер SQL Server 2005 Express Edition встановлений і ми маємо на комп'ютері наступний результат:



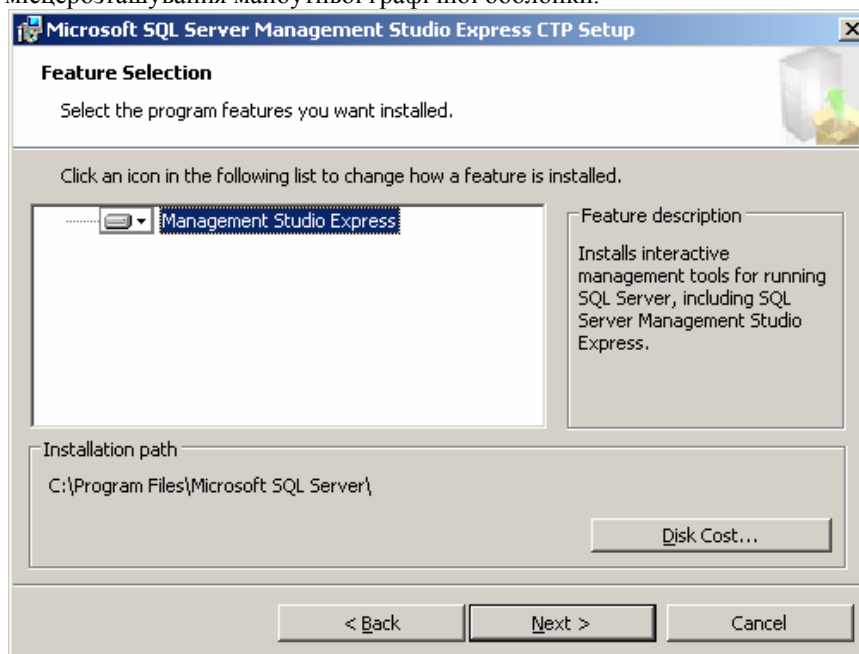


Але не дивлячись на пророблені дії, цього замало. Це лише сам сервер з певним набором утиліт для адміністрування. Для більш розширеного адміністрування та роботи з даними SQL Server існує інструмент **SQL Server Management Studio Express**. В нашу версію інсталяції він не був включений, тому ставимо його вручну. Для цього запускаємо файл **SQLServer2005\_SSMSEE.msi**.

Після запуску інсталятора і ознайомлення з ліцензійними умовами, вам буде запропоновано ввести свої дані: ім'я та назву компанії (по бажанню).



Після цього потрібно визначити, які компоненти ви хочете проінсталиувати. Нажаль, вибір буде невеликий ☹. Доречі, тут же можна змінити і місцезнаходження майбутньої графічної оболонки.



Після натиснення кнопки «Next» розпочнеться процес інсталяції обраних складових, в результаті успішного завершення якого ви будете мати зручну графічну оболонку SQL Server Management Studio Express. З нею ми познайомимось пізніше.

### 3.2. Інсталяція MS SQL Server 2008 Enterprise

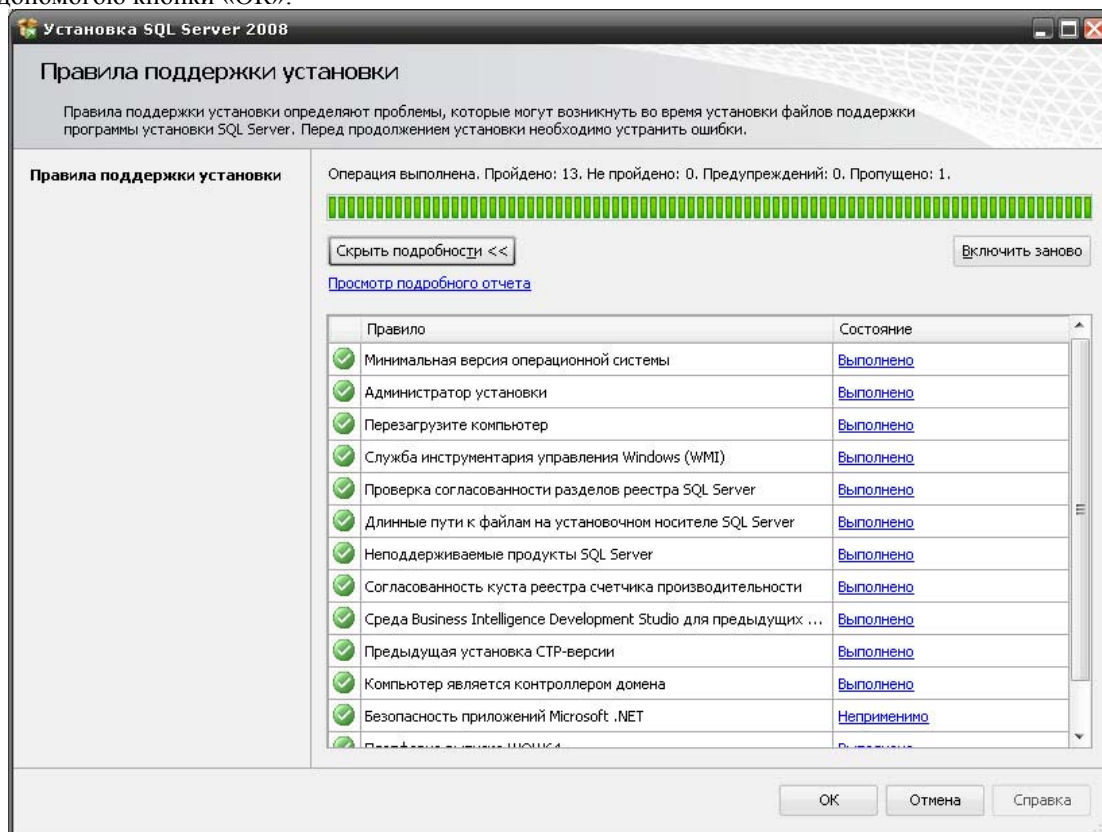
Для успішної інсталяції MS SQL Server 2008 Enterprise необхідна наявність встановлених пакетів: Microsoft .NET Framework 3.5 SP1, Windows Installer 4.5 та Windows PowerShell 1.0. Але, якщо у Вас немає необхідних компонентів - не слід розчаровуватись, програма-інсталятор їх встановить одразу після запуску.

Саме встановлення MS SQL Server 2008 включає в себе наступні **кроки**:

1. Запустити програму інсталяції.
2. В розділі «Планирование» обрати пункт «Средство проверки конфигурации»:



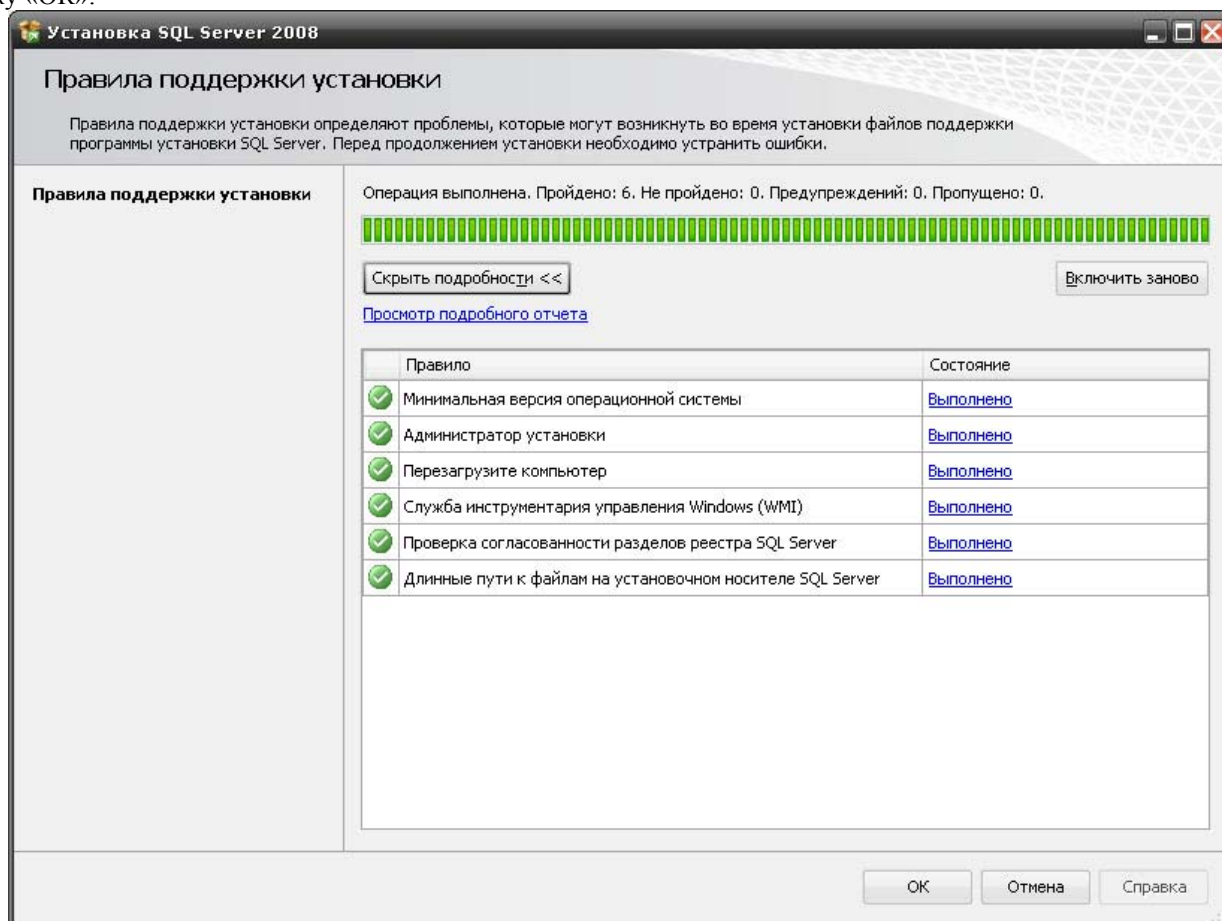
3. Натиснути кнопку «Показать подробности» та впевнитись, що всі перевірки успішно пройдені. Якщо будуть виявлені помилки, їх необхідно виправити і запустити повторну перевірку кнопкою «Включить заново». Після цього закрити вікно за допомогою кнопки «ОК»:



4. Після цього слід в вікні інстальатора обрати розділ «Установка» та пункт «Новая установка изолированного SQL Server или добавление компонентов к существующему экземпляру». Якщо у вас встановлена попередня версія, то ви можете її оновити до необхідної за допомогою пункту «Обновление с SQL Server 2000 или SQL Server 2005»:

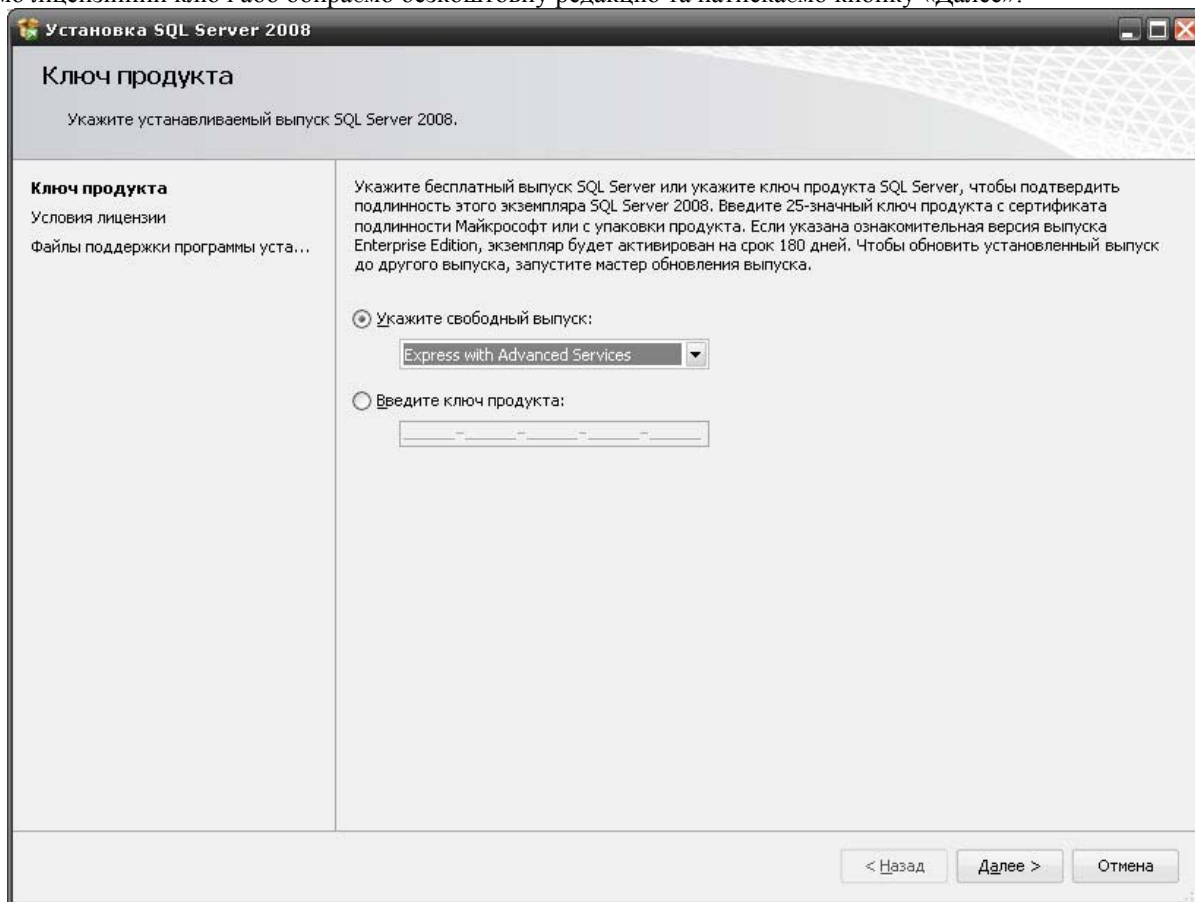


5. Після цього запуститься процес перевірки системи перед інсталяцією. Аналогічно до пункту 3, ви маєте можливість переглянути детальний звіт цієї перевірки за допомогою кнопки «Показать подробности», а для завершення натиснути кнопку «ОК»:

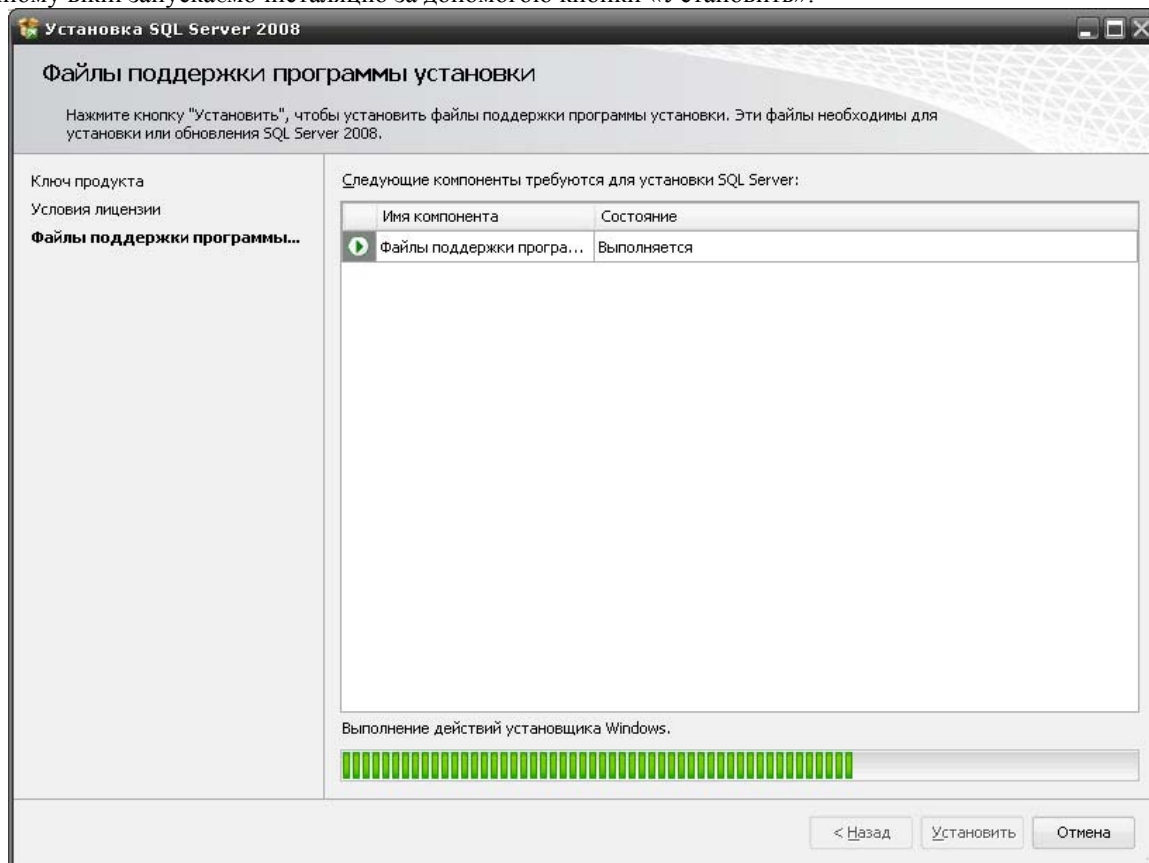




6. Вводимо ліцензійний ключ або обираємо безкоштовну редакцію та натискаємо кнопку «Далее»:



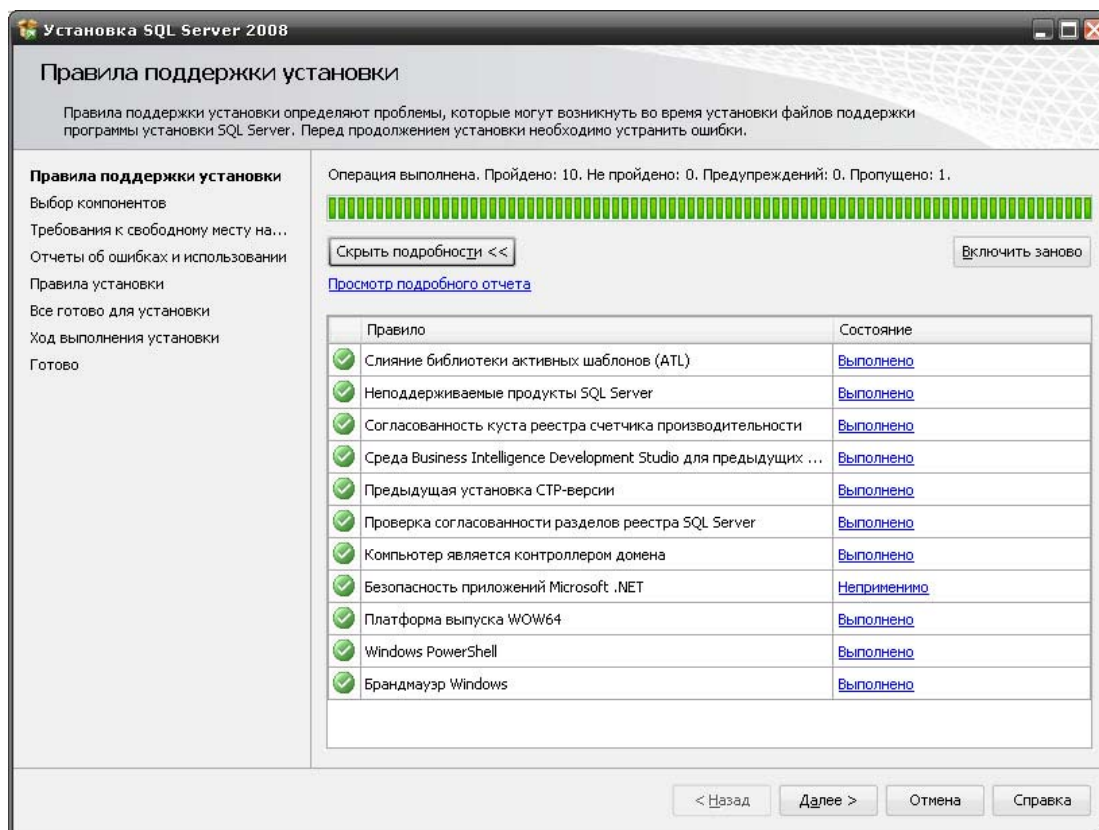
7. Далі необхідно прочитати ліцензію. Якщо після всього прочитаного, ви наважитесь з ними погодитись, тоді ставите відповідну галочку і переходите на наступний крок (кнопка «Далее»). Инакше процес інсталяції для вас завершився.
8. В наступному вікні запускаємо інсталяцію за допомогою кнопки «Установить»:





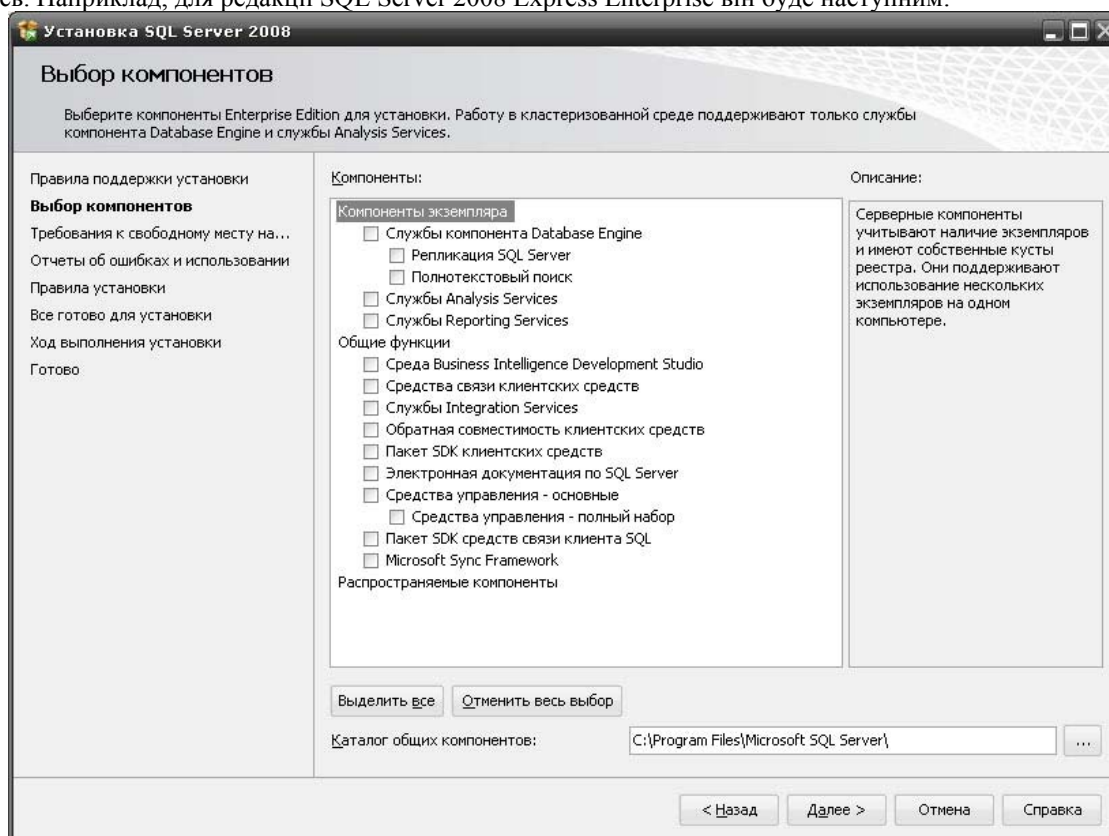


9. Перевірку для подальшої роботи здійснюємо аналогічно пункту 3 та 5 інсталяції. Після цього натискаємо кнопку «Далее»:



**Примітка!** Попередження в пункті «Брандмауер Windows», якщо воно виникло, можна проігнорувати – воно акцентує увагу на тому, що потрібне його (брандмауера) додаткове налаштування для доступу до SQL Server з інших комп'ютерів.

10. Обираємо необхідні компоненти для встановлення. Їх кількість і склад в залежності від редакції SQL Server 2008 буде відрізнятися. Наприклад, для редакції SQL Server 2008 Express Enterprise він буде наступним:





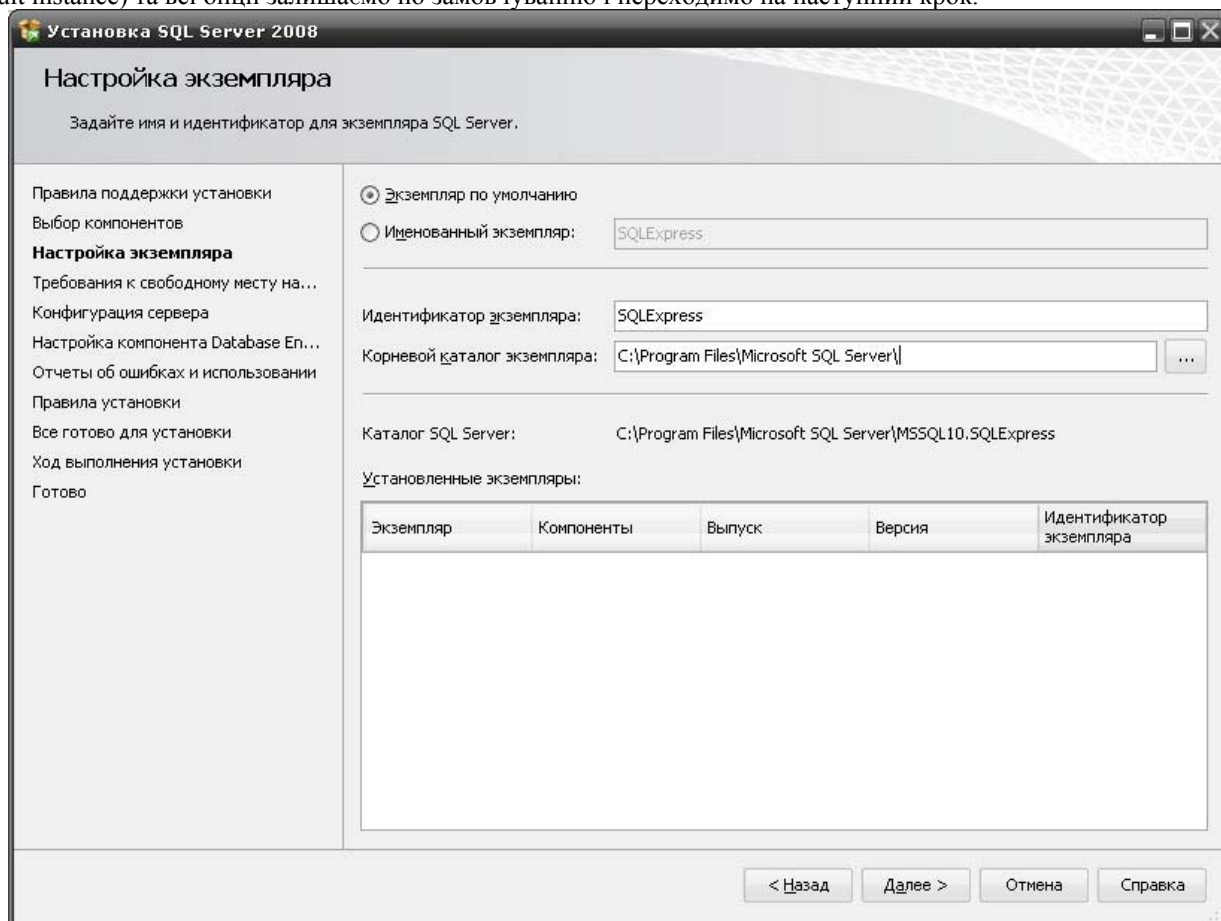
Коротко опишемо список **компонентів**:

1. **Компоненти екземпляра (Instance Features)** – це набір серверних компонентів для окремих екземплярів. Для кожного з цих компонентів існують окремі гілки реєстру. До них належать:
  - 1.1. **Служби компонента Database Engine (Database Engine Services)** – вмикають компонент Database Engine, який являє собою основну службу для збереження, захисту та обробки даних. По суті, це ядро SQL Server, яке використовує .NET. Ці служби забезпечують управляємий доступ, швидку обробку транзакцій та підтримку високого рівня доступності. Гадаємо, що ви і самі розумієте, що встановлення служби Database Engine являється обов'язковим.
    - **Реплікація SQL Server (SQL Server Replication)** – відповідає за можливість реплікації даних.
    - **Повнотекстовий пошук (Full-Text Search)** – відповідає за швидкий пошук тексту в існуючих даних, тобто за роботу з індексами, що зберігають значення даних та вказівники на рядки, де ці дані зустрічаються. Тобто відповідає за технологію сервера Microsoft Index Server.
  - 1.2. **Служби Analysis Services** – це набір засобів для підтримки аналітичної обробки (OLAP) та інтелектуального аналізу даних. Підтримується лише в редакції MS SQL Server 2008 Enterprise.
  - 1.3. **Служби Reporting Services** – це набір компонентів, засобів і програмних інтерфейсів для розробки та використання вбудованих звітів в управляємому середовищі. В набір засобів входять засоби розробки, налаштування і адміністрування, а також засоби перегляду звітів. До програмних інтерфейсів відносяться протокол доступу до об'єктів (Simple Object Access Protocol, SOAP), кінцеві точки URL-адреси і інструментарій управління Windows (WMI), який забезпечує просту інтеграцію з Web-додатками і порталами.
2. **Спільні функції (Shared Features)** – використовуються всіма екземплярами сервера:
  - 2.1. **Засоби Business Intelligence Development Studio** – нова версія середовища Microsoft Visual Studio 2008 з додатковими типами проектів, які характерні для роботи з бізнес-аналітикою в SQL Server. Це основне середовище для розробки бізнес-рішень, до складу яких входять проекти служб Analysis Services, Integration Services і Reporting Services. Детальніше дивіться розділ опису даної утиліти.
  - 2.2. **Засоби зв'язку клієнтських засобів (Connectivity client tools)** – засоби для зв'язку клієнтів та сервера.
  - 2.3. **Служби Integration Services** – ці служби використовуються для здійснення конвертування, інтеграції та переміщення даних між різного роду бізнес-додатками. Не написавши жодного рядка коду, можна створювати рішення служб Integration Services, які дозволяють вирішувати складні бізнес-проблеми, управляти базами даних SQL Server, а також копіювати об'єкти SQL Server з одного екземпляра в інший.
  - 2.4. **Зворотна сумісність клієнтських засобів (Backward compatibility client tools)** – передбачає включення клієнтських засобів, які будуть забезпечувати сумісність з молодшими версіями SQL Server, наприклад, консоль управління Microsoft, різного роду API-інтерфейси тощо.
  - 2.5. **Пакет SDK клієнтських засобів** – включає пакет програмного забезпечення з ресурсами для програмістів.
  - 2.6. **Електронна документація по SQL Server** – електронний довідник по SQL Server.
  - 2.7. **Засоби управління – основні (Management Tools – Basic)** – включає підтримку середовища Management Studio для компонента Database Engine, утиліту командного рядка SQLCMD та постачальник SQL Server PowerShell.
  - 2.8. **Пакет SDK засобів зв'язку клієнта SQL (SQL Client Connectivity SDK)**
  - 2.9. **Microsoft Sync Framework** – комплексна платформа синхронізації, яка підтримує сумісну роботу і автономний доступ для додатків, служб і пристроїв. В платформі застосовуються технології і засоби, які підтримують переміщення, сумісне використання даних і переведення їх в автономний режим. За допомогою платформи Sync Framework розробники можуть створювати екосистеми синхронізації, які дозволяють інтегрувати довільні додатки з даними будь-якого джерела даних, використовуючи будь-який мережний протокол.
3. **Розповсюджувані компоненти (Redistributable Features)** – набір компонентів, які встановлюються у випадку необхідності: звіти про помилки, власний клієнт SQL Server, MSXML версії 6.0, SQL Server Compact Workbench, середовище виконання SQL Server Compact, Sync Services для ADO.NET і оглядач SQL Server.

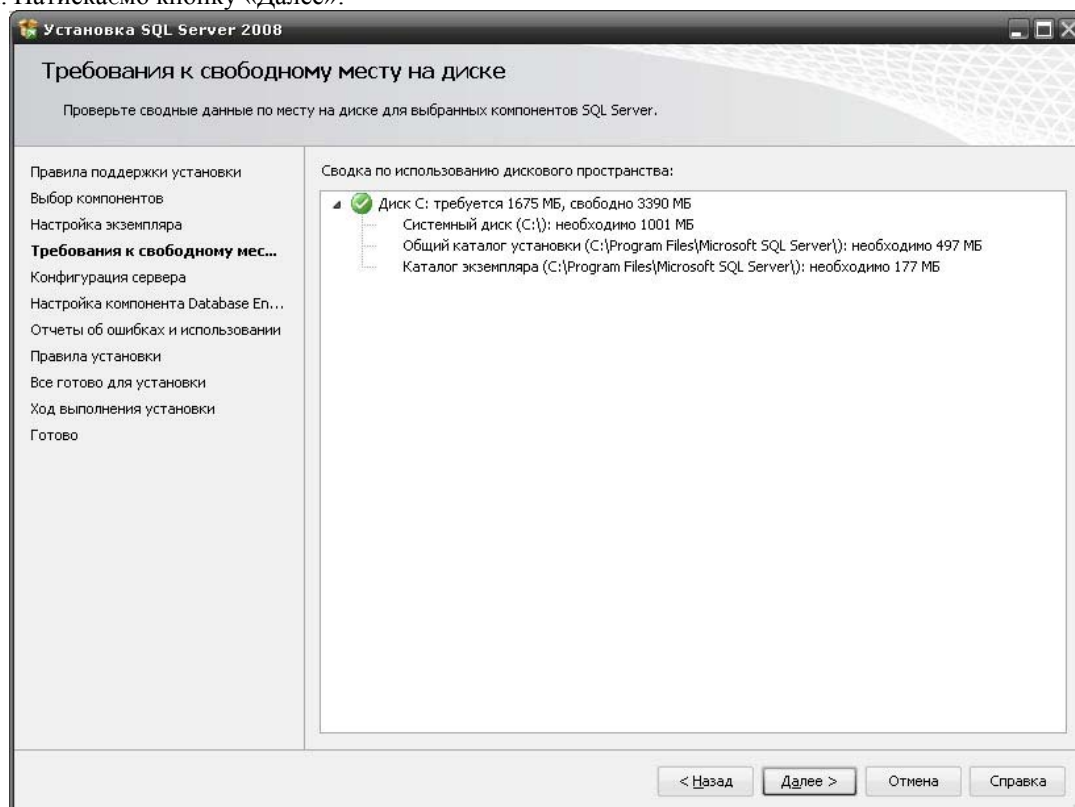




11. Обираємо **варіант інсталяції** – екземпляр по замовчуванню (Default instance) чи іменованний екземпляр (Named instance). Більш детально ця інформація міститься в попередньому [підрозділі](#). Ми обираємо стандартний екземпляр (Default instance) та всі опції залишаємо по замовчуванню і переходимо на наступний крок.

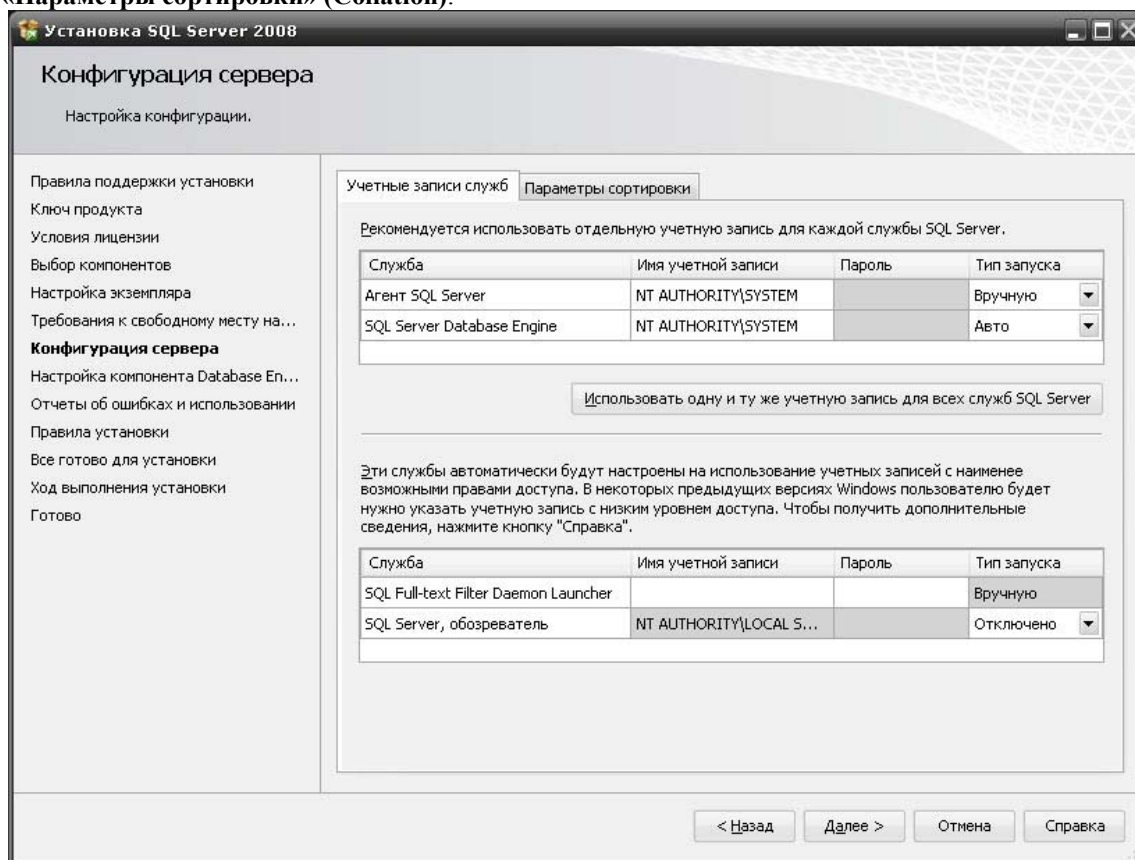


12. Наступний крок знайомить нас з тим, скільки місця буде займати наш сервер баз даних, та по якій адресі він буде розміщений. Натискаємо кнопку «Далее»:



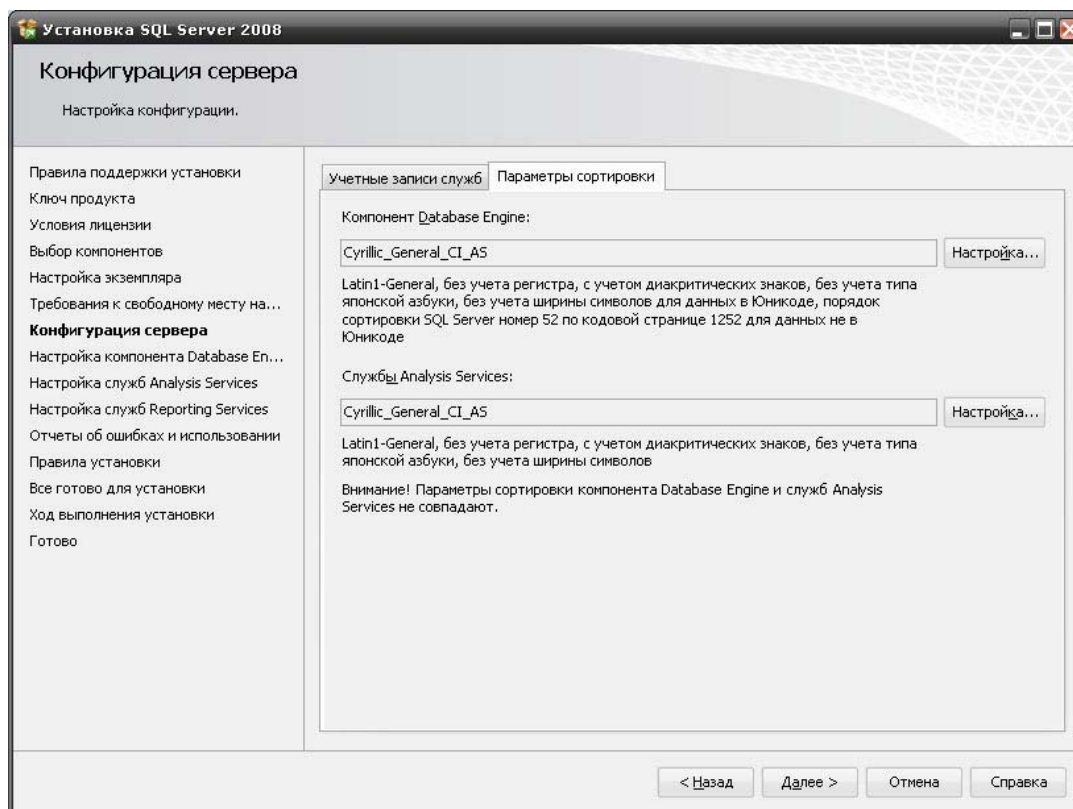


13. Налаштовуємо конфігурацію майбутнього сервера. Для цього обираємо опції як показано на рисунку і переходимо на закладку «Параметры сортировки» (Collation):



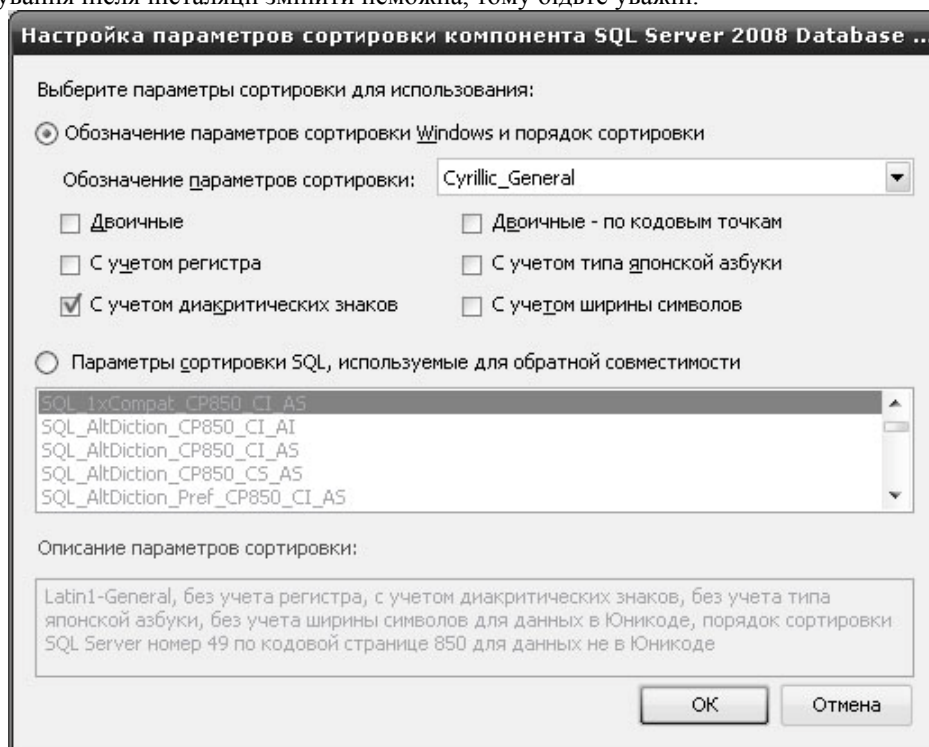
**Примітка!** Якщо Ви хочете використовувати SQL Server Agent для виконання регулярного резервного копіювання файлів БД на інший комп'ютер в своїй локальній мережі (рекомендується) і Ваша мережа побудована з використанням домена Windows NT, то необхідно завести в Вашому домені окремий обліковий запис для SQL Server Agent і надати йому права на відповідні ресурси. При такій конфігурації, в цьому ж вікні необхідно в поля «Имя учетной записи» (Account Name) і «Пароль» (Password) ввести ім'я створеного облікового запису (вигляду Домен\Ім'я) і її пароль.

14. В закладці «Параметры сортировки» вказати необхідні значення та натиснути кнопку «Далее». Ми залишаємо все без змін:

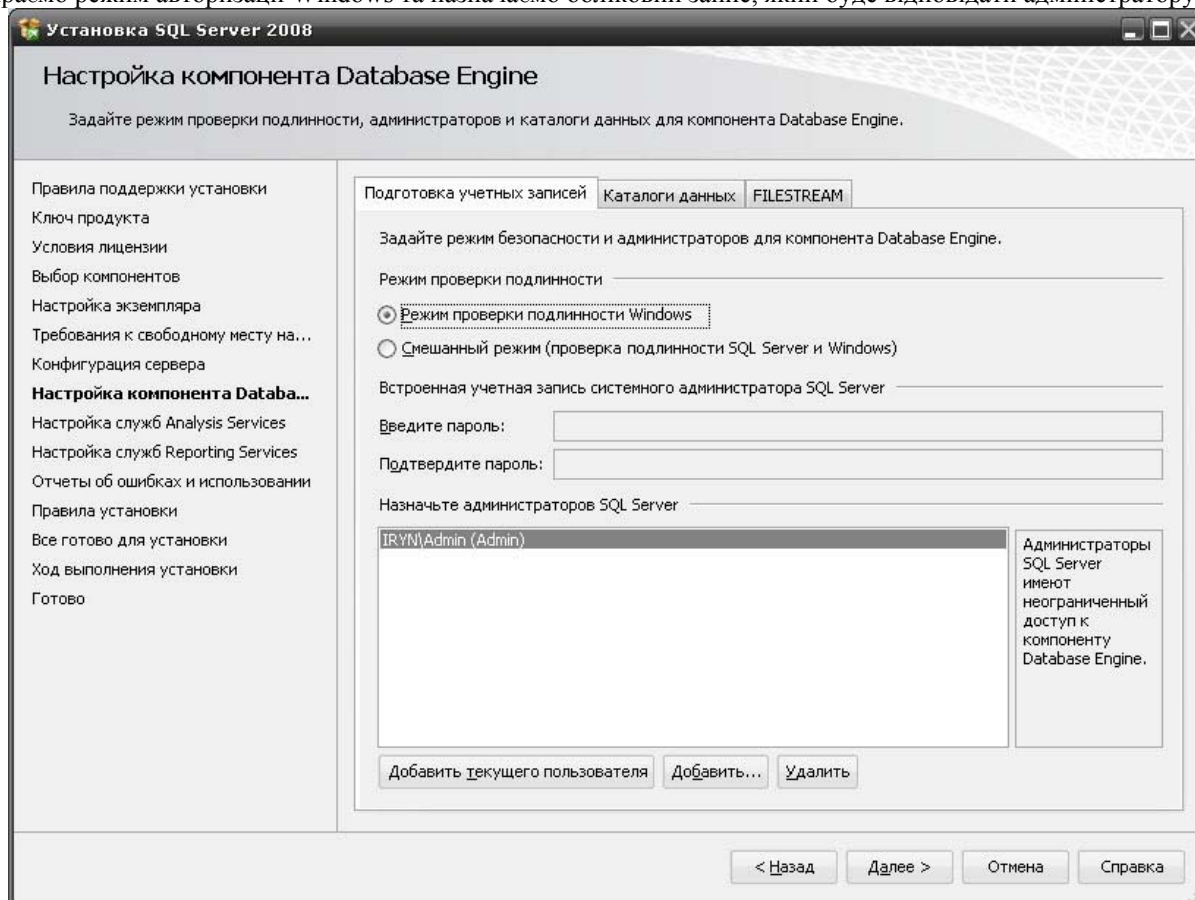




**Примітка!** Щоб змінити дані опції слід натиснути відповідну кнопку «Настройка» і встановити необхідні параметри. Доречі, дані налаштування після інсталяції змінити неможна, тому будьте уважні.

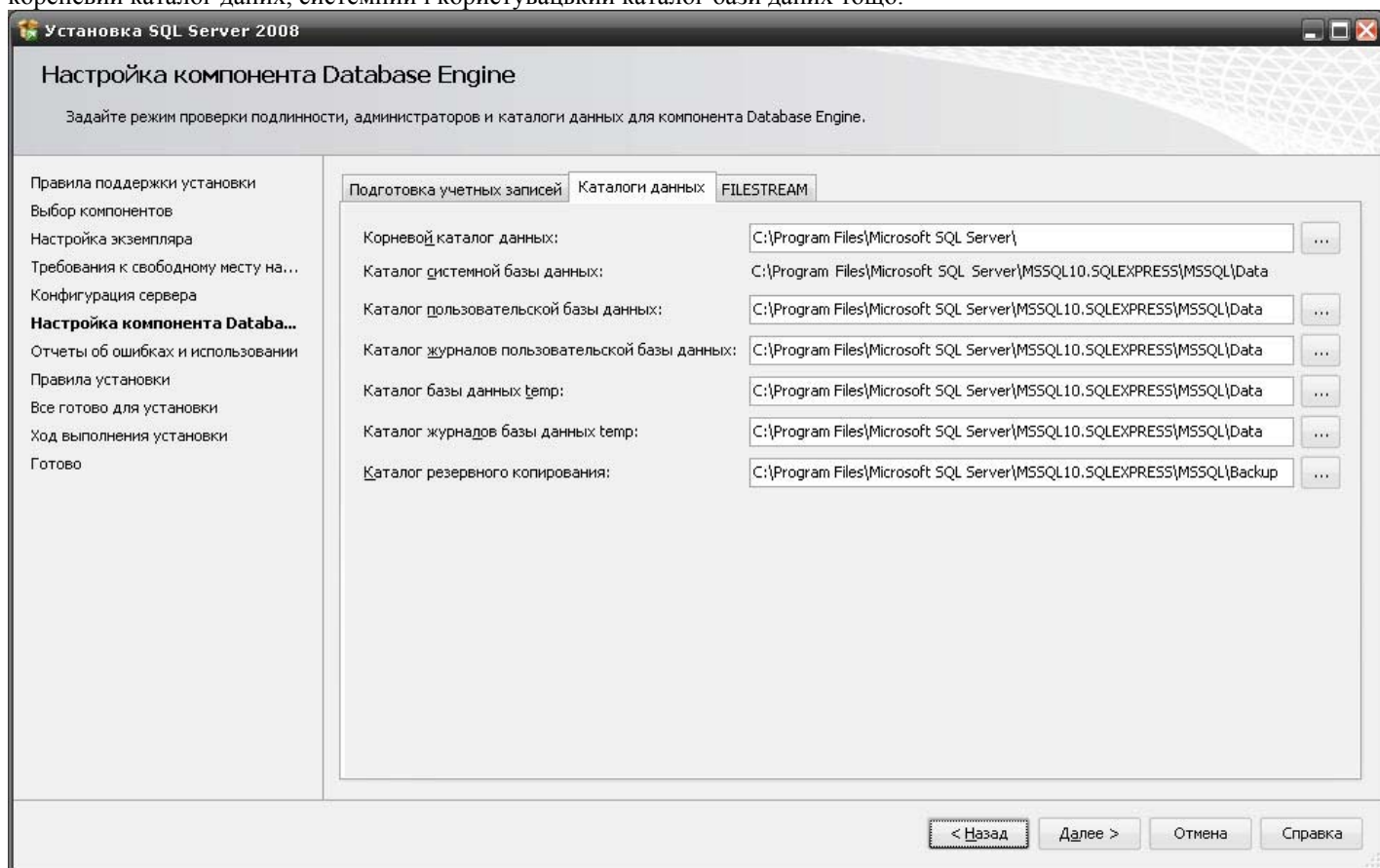


15. Оберіть спосіб авторизації користувача на сервері (детальніше можна прочитати в попередньому [підрозділі](#)). Ми обираємо режим авторизації Windows та назначаємо обліковий запис, який буде відповідати адміністратору.

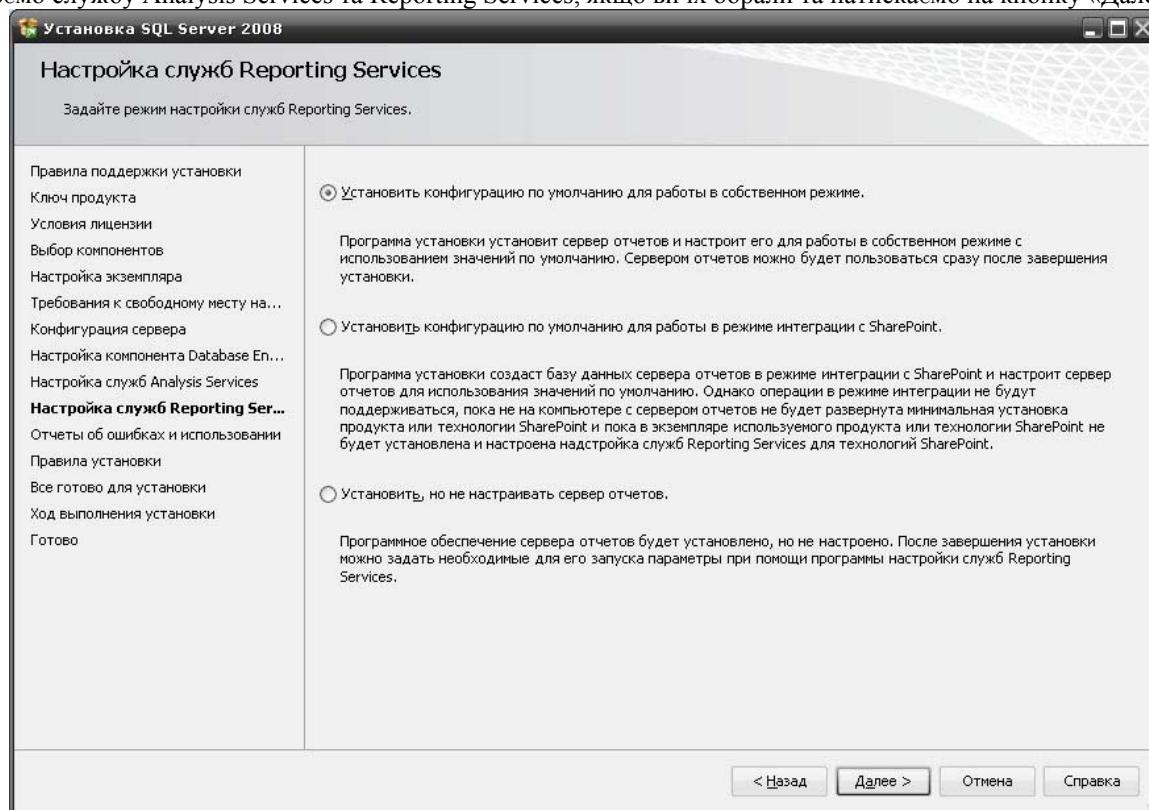




16. Після цього переходимо на закладку «Каталоги данных» (Data Directories) і налаштовуємо необхідні каталоги: кореневий каталог даних, системний і користувацький каталог бази даних тощо.



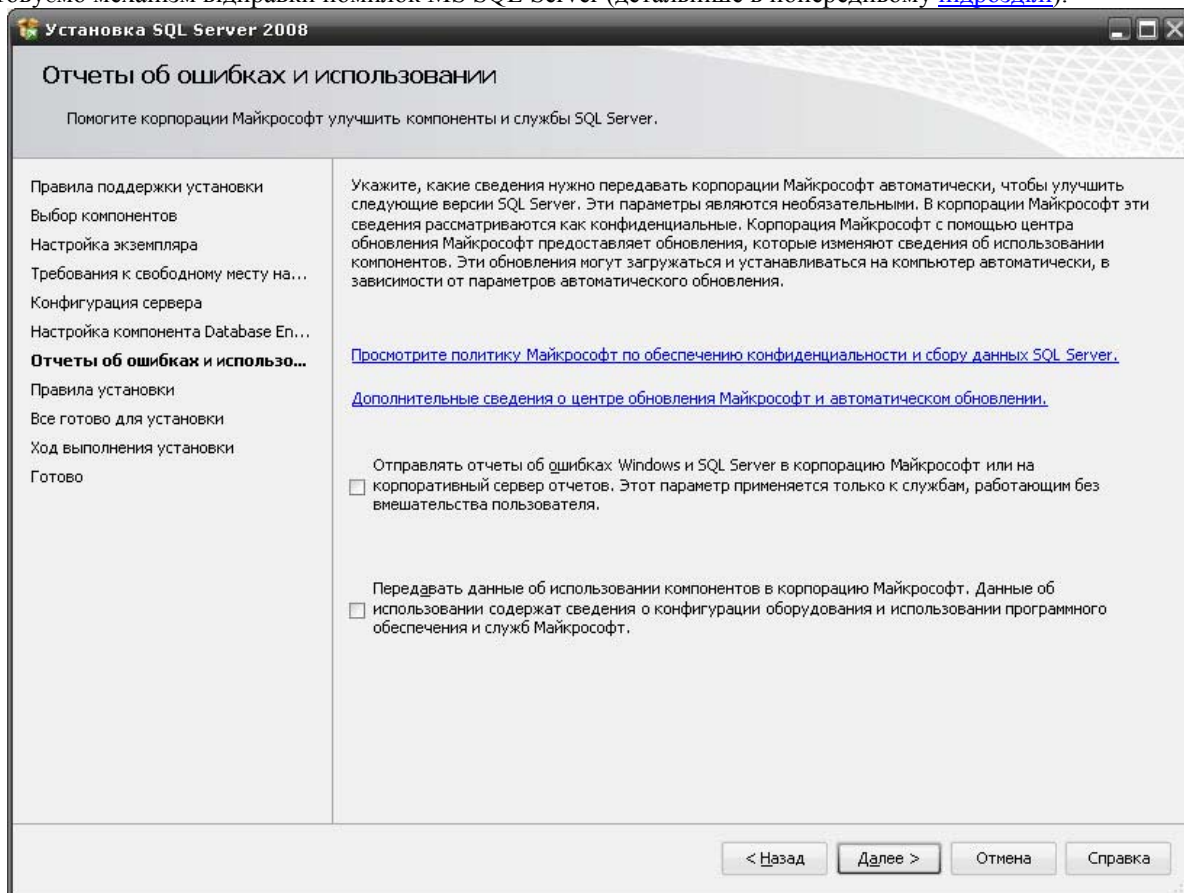
17. Встановлюємо службу Analysis Services та Reporting Services, якщо ви їх обрали та натискаємо на кнопку «Далее»:





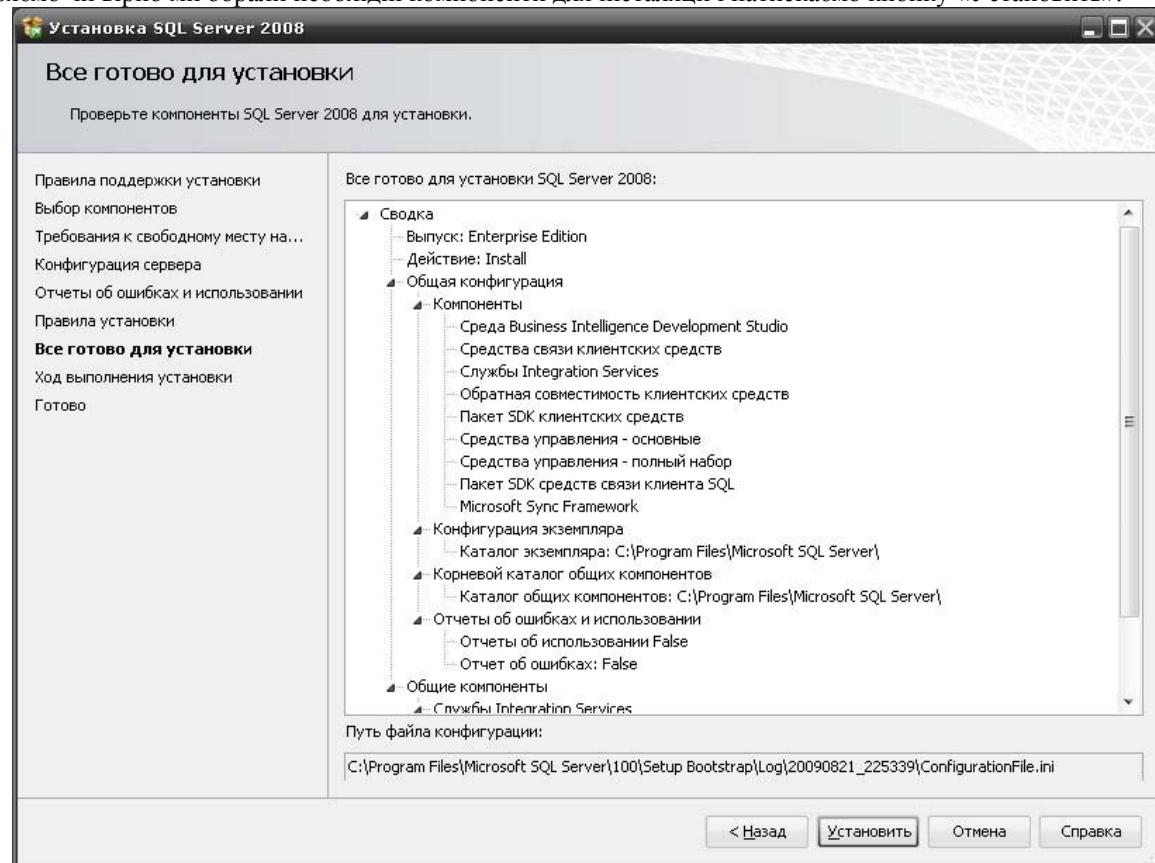


18. Налаштовуємо механізм відправки помилок MS SQL Server (детальніше в попередньому [підрозділі](#)).



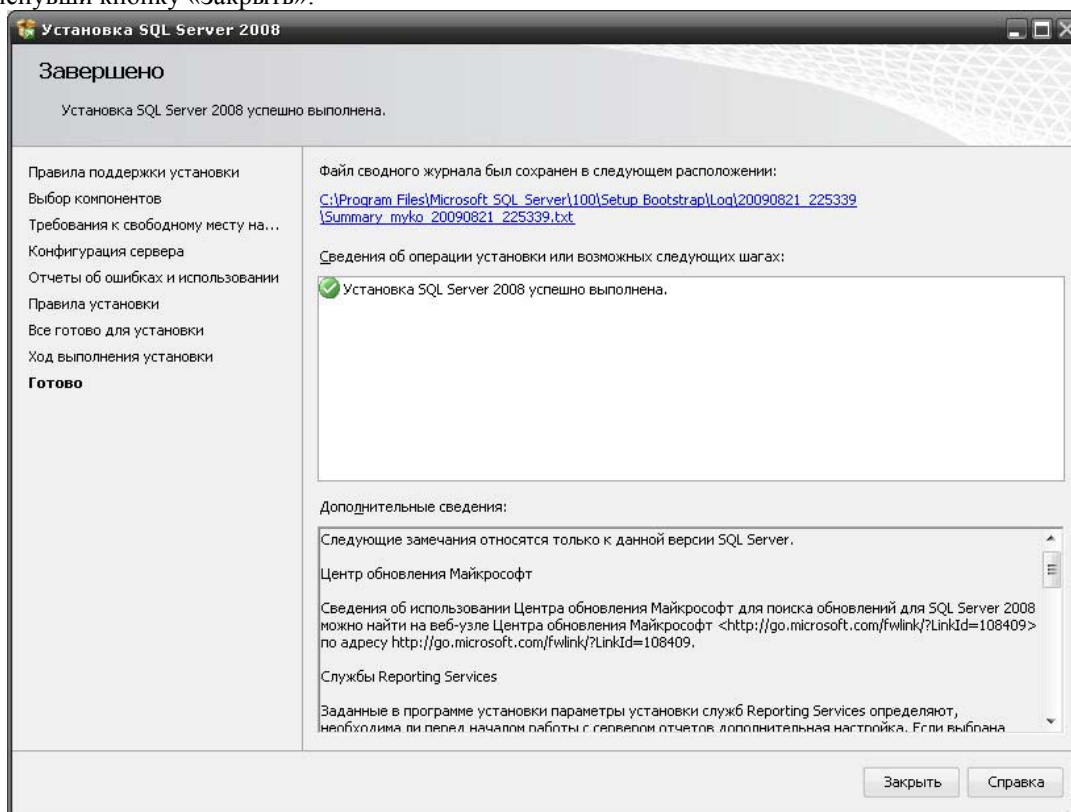
19. Проходимо чергову перевірку і якщо все гаразд, переходимо на наступний крок інсталяції.

20. Перевіряємо чи вірно ми обрали необхідні компоненти для інсталяції і натискаємо кнопку «Установить»:





21. Після завершення роботи інсталера і перегляду самого процесу інсталяції переходимо на наступний крок і завершуємо роботу, натиснувши кнопку «Закрити»:



От і все. Вітаємо Вас з завершенням процесу інсталяції MS SQL Server 2008 Enterprise.

## 4. Утиліти SQL Server 2000/2005

### 4.1. Загальна інформація про утиліти

В SQL Server 2000 вся робота відбувалась через набір утиліт, основними з яких є:

- ✓ **Enterprise Manager** – загальна утиліта для адміністрування, яка включає також засоби імпорту/експорту бази даних..
- ✓ **Query Analyzer** – утиліта для створення запитів.
- ✓ **Services Manager, Client Network Utility, Server Network Utility** – дозволяють контролювати SQL сервери, що працюють на машині. Точніше вони відповідають за управління службами SQL Server на комп'ютері, де встановлений сервер баз даних, конфігурувати мережеві протоколи серверної та клієнтської частин тощо.

SQL Server 2005 та 2008 надає нові засоби управління SQL Server, а вищеописані об'єднує:

- Інструмент управління базами даних - **SQL Server 2005 Management Studio**, що об'єднує в собі всі можливості колишніх утиліт SQL Server Enterprise Manager та SQL Query Analyzer та ряд можливостей програми Analysis Manager.
- Інструменти розробки баз даних - **SQL Server 2005 Business Intelligence Development Studio**, - це нова версія середовища Microsoft Visual Studio з додатковими типами проектів, які характерні для роботи з бізнес-аналітикою в SQL Server.
- Диспетчер конфігурації SQL Server – **SQL Server Configuration Manager**, який об'єднує в собі всі можливості, які містились порізно в утилітах Services Manager, Client Network Utility, Server Network Utility.
- **Sqlcmd** – нова утиліта командного рядка для заміни isql і osql.

Отже, познайомимось з ними.

### 4.2. SQL Server Configuration Manager

**SQL Server Configuration Manager** – це засіб, який призначений для:

- управління службами, зв'язаними з SQL Server;
- налаштування мережевих протоколів, які використовуються SQL Server;
- управління конфігурацією підключень з клієнтських комп'ютерів SQL Server.

SQL Server Configuration Manager являє собою оснастку консолі управління (mmc.exe) для відкриття якого використовується файл **SQLServerManager.msc** в папці **Windows->System32**. Як вже було сказано, дана утиліта об'єднує



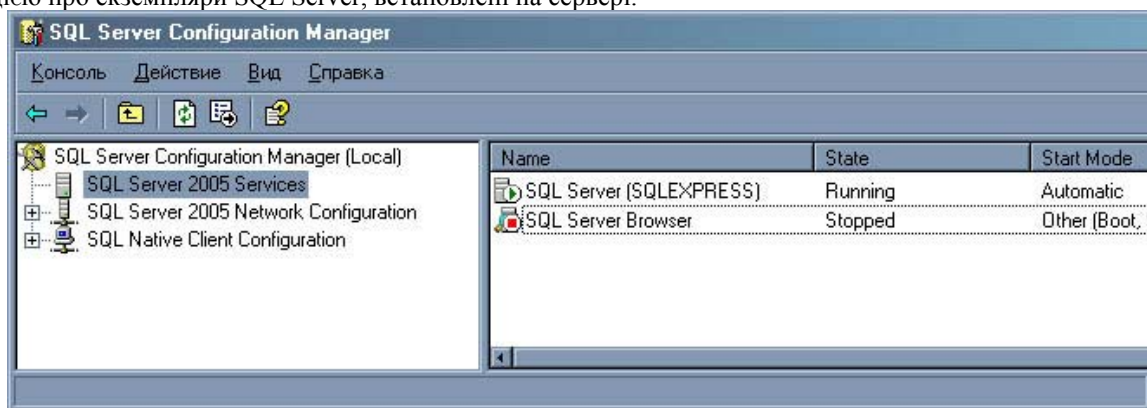
в собі функціональні можливості таких засобів SQL Server 2000 як Server Network Utility, Client Network Utility і диспетчер служб (Services Manager).

Вигляд і основні принципи роботи даної утиліти в версіях SQL Server 2005 та SQL Server 2008 не відрізняється.

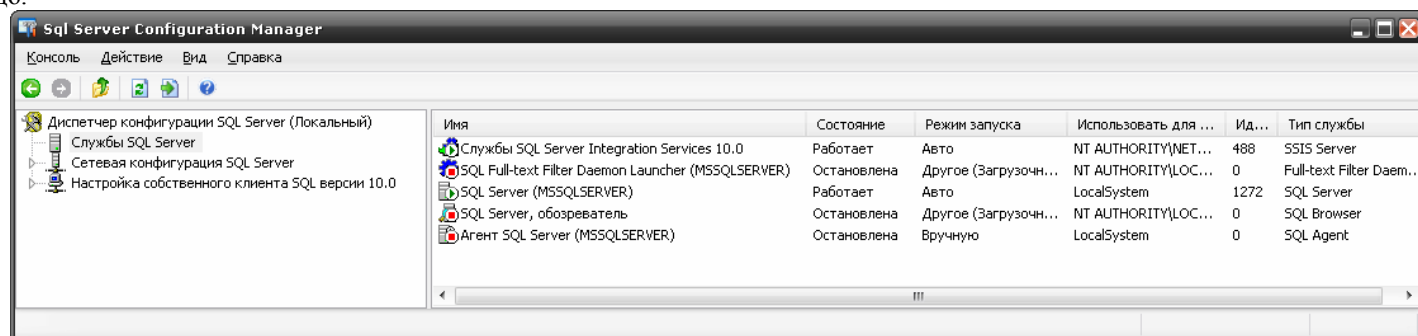
На закладці «SQL Server Services» («Службы SQL Server») ви можете переглянути список та поточний стан сервісів, якими Ви можете управляти.

В SQL Server 2005 Express Edition або SQL Server 2008 Express їх всього дві:

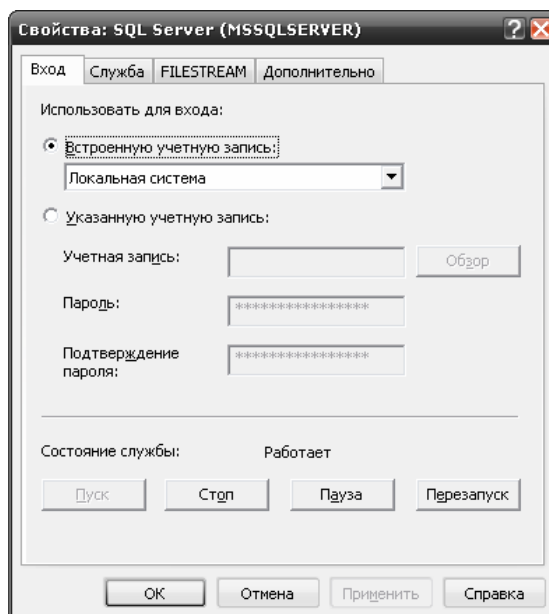
- **SQL Server** – це ядро SQL Server. Вона обслуговує такі ключові задачі як виконання запитів, управління доступом до даних і розподілення системних ресурсів (оперативна пам'ять та потужність процесора).
- **SQL Browser**. Ця служба прослуховує в мережі вхідні запити до ресурсів SQL Server, а також забезпечує користувача інформацією про екземпляри SQL Server, встановлені на сервері.



В SQL Server 2005/2008 Enterprise (або іншій розширеній редакції сервера) цей список може бути дещо ширшим і залежить від того, які компоненти служб Ви обрали при інсталяції сервера. Наприклад, на рисунку, який представлений нижче, включена додатково служба управління автоматизацією адміністрування SQL Agent, SQL Server Integration Services тощо.



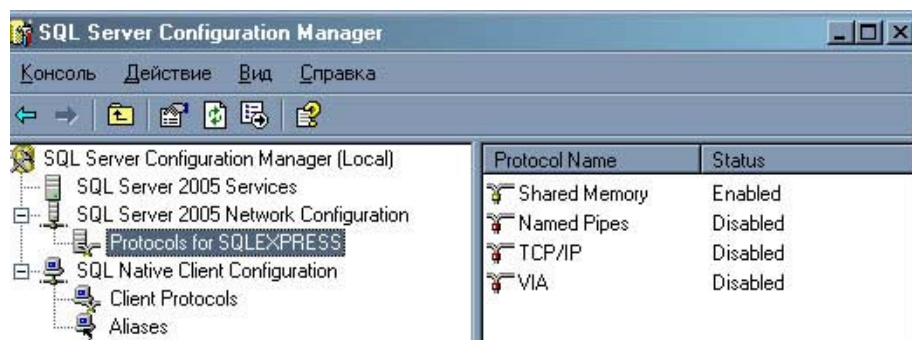
Щоб змінити конфігурацію певної служби, достатньо відкрити вікно її властивостей. Наприклад, ви можете запускати або зупиняти службу, вказувати режим її автозапуску, змінити ім'я користувача і пароль облікового запису даної служби тощо.



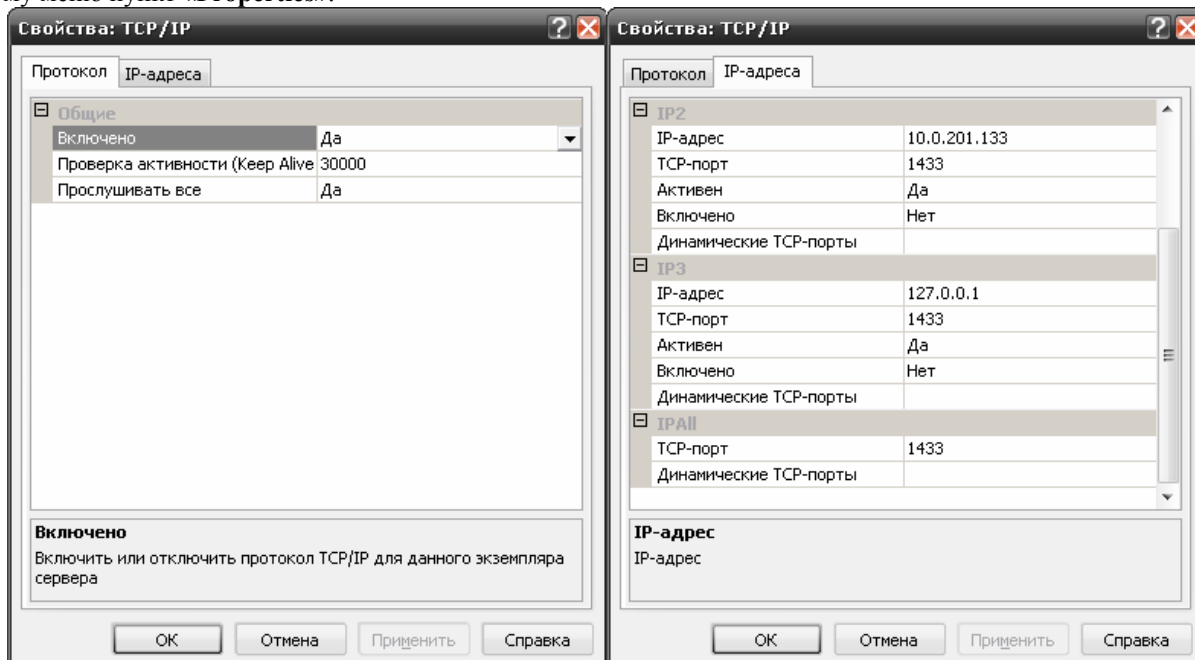




Щоб клієнт мав можливість підключатись до SQL Server по мережі, на обох кінцях з'єднання повинна бути встановлена спільна мережева бібліотека. В розділі «SQL Server Network Configuration» («Сетевая конфигурация SQL Server») ви зможете переглянути список мережевих бібліотек, встановлених на сервері і отримати можливість їх сконфігурувати.



Якщо розкрити групу «Protocols for SQLEXPRESS» або «Protocols for MSSQLSERVER», ви побачите список протоколів, які може «прослуховувати» сервер. Для налаштування властивостей будь-якої бібліотеки, оберіть в контекстному меню пункт «Properties».



На закладці «IP-Address» («IP-адреса») Ви можете налаштувати протокол на одних мережевих адаптерах і відключити на інших, сконфігурувати окремо кожен адаптер або ж використати спеціальну опцію «IP All» і сконфігурувати всі адаптери як єдиний пристрій.

Розділ «SQL Native Client Configuration» («Настройка собственного клиента SQL») являється партнером утиліти SQL Server 2000 Network Configuration. Його використовують для налаштування протоколів, які використовуються клієнтською стороною для взаємодії з сервером. В даному розділі існує лише два пункти:

- **Client Protocols** - список доступних протоколів. Відмінність даного списку від списку протоколів SQLSERVER (з розділу SQL Server Network Configuration) полягає лише в тому, що кожен протокол має свій порядковий номер. Ці порядкові номери вказують на порядок, в якому клієнт буде підключатись до сервера
- **Aliases**. Даний підрозділ використовується лише у виключних ситуаціях і тому розберемо його на прикладі. Багато компаній мають кілька екземплярів SQL Server, які працюють паралельно та мають різні налаштування. Наприклад, один сервер може працювати з включеною мережевою бібліотекою TCP/IP і прослуховувати порт 1433, а інший - порт TCP 37337 (зазвичай використовується системою безпеки). Інші сервери можуть мати інші налаштування, які дозволяють різним клієнтам підключатись до сервера. Якщо в Вашій компанії використовується такий підхід, тоді для кожного з серверів організації, який не використовує значення по замовчуванню для мережевої бібліотеки, можна створювати серверний псевдонім.

### 4.3. SQL Server Management Studio

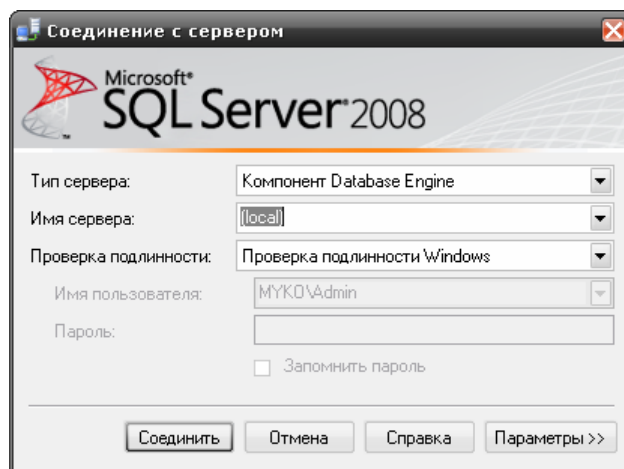
Середовище SQL Server Management Studio – це інтегроване середовище для доступу, налаштування, управління і адміністрування всіх компонентів SQL Server. Воно об'єднує в єдине середовище можливості програм Enterprise Manager і Query Analyzer, які входили до складу попередніх версій SQL Server. Тобто SQL Server Management Studio являється



гарним помічником в адмініструванні, створенні та розробці баз даних, таблиць, зв'язків тощо. Крім того вона надає середовище для управління службами Analysis Services, Integration Services, Reporting Services і запитам на XQuery.

Зовнішній вигляд і основні принципи роботи в SQL Server 2005 та SQL Server 2008 не відрізняються.

При вході в SQL Server Management Studio вам запропонують пройти аутентифікацію. На екрані з'явиться наступне вікно:



Для початку вам необхідно обрати тип сервера, а в полі «SQL Server» («Імя сервера») обрати власне сам сервер (при необхідності з вказанням його IP-адреси), з яким ви будете працювати. Після цього потрібно вказати режим авторизації: Windows authentication чи SQL Server authentication. При виборі останнього вам буде представлена можливість введення логіну та паролю.

Після введення даних для підключення перед вами з'явиться власне середовище Management Studio.

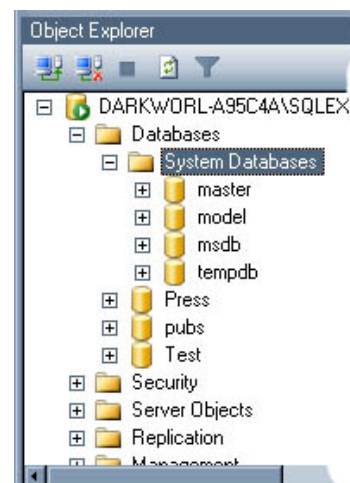


В лівій панелі цієї утиліти розміщується панель **Object Browser**, в випадаючому деревовидному списку якої відображаються всі активні з'єднання **SQL Server**. Для кожного зареєстрованого сервера відводиться своя гілка, назва якої рівна імені сервера, до якого здійснюється під'єднання, та логін, введений при реєстрації.

Кожен сервер має набір стандартних груп, які розміщуються в окремих папках:

1. В папці **Databases** вказані всі бази даних, які є на даному сервері. В СУБД існує група вже **наперед встановлених баз даних**, до яких належать:

- База **master** - одна з важливих баз даних SQL сервера. Вона містить інформацію про користувачів, зберігає процедури, повідомлення про помилки та іншу службову інформацію.
- База **model** - на її основі створюються всі нові бази, тобто вона являється шаблоном для інших (користувацьких) баз даних. При цьому в нову базу даних копіюються всі об'єкти model. Якщо ви маєте права на її модифікацію, то можете помістити в неї нові, наприклад, користувацькі типи даних (домени) або таблиці. Після цього при створенні користувацьких баз даних ці типи даних або таблиці завжди будуть в них існувати.

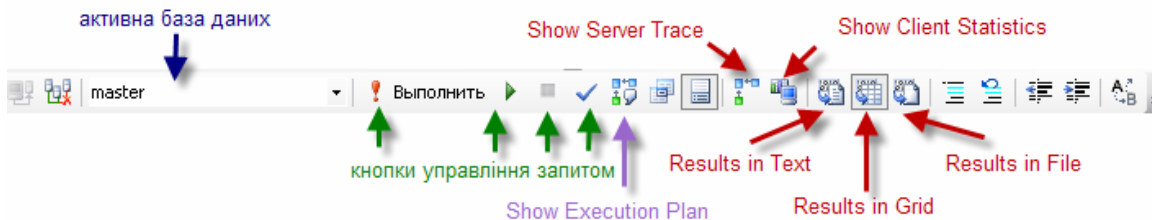




- База **msdb** – ця база даних підтримує службу SQL Server Agent, включаючи дані про завдання, оповіщення, події і реплікації. В ній також зберігаються відомості про всі резервні копії та процедури відновлення. Вона займає орієнтовно 12 Мб.
  - База **tempdb** – в цій базі виконується сортування, об'єднання, збереження тимчасових результатів і інші зміни з даними, які вимагають додаткового місця на жорсткому диску. Ця база даних після інсталяції займає приблизно 2Мб, але має здатність розростатись. При кожному перезапуску SQL Server база даних заново переініціалізовується.
2. Папка **Security** відповідає за налаштування системи безпеки даних, що зберігаються в SQL Server.
  3. Папка **Server Objects** містить засоби управління іншими службами комп'ютера, для створення зв'язаних серверів тощо.
  4. Папка **Replication** зберігає структуру системи реплікації даних. При необхідності можна вносити зміни в цю структуру.
  5. В папці **Management** знаходяться засоби адміністрування SQL Server.
  6. В SQL Server 2005/2008 служба **SQL Agent** має власну гілку.

По центру розміщується область, в якій за допомогою кнопки на панелі інструментів «New Query» («Создать запрос») або пункту меню «File->New->Blank Query Window» відкривається дочірнє вікно, в якому можна писати SQL запити. Таких вікон можна створювати безліч.

Панель інструментів, яка розміщена нижче, призначена для роботи з редактором SQL та має наступний вигляд.



Перші дві кнопки управляють з'єднанням з сервером баз даних («Connect» та «Disconnect»). Далі, у випадковому списку вказується, яка база даних являється активною. Всі запити та дії, які ви здійснюєте будуть відноситись саме до неї. По замовчуванню – це база даних master. Щоб змінити активну базу даних, її достатньо обрати в цьому списку.

Далі розміщується 4 кнопки для управління запитом. Перша кнопка (з **знаком оклику**) дозволяє перевірити Ваш запит на наявність помилок. Причому перевіряються також порядок ключових слів у запиті. При натисненні кнопки у вигляді **зеленої стрілочки**, написаний SQL запит запуститься на виконання. У випадку помилки, на екрані з'явиться відповідне повідомлення. Кнопка з **червоним квадратом** – це кнопка «СТОП», яка дозволяє критично завершити роботу запиту. Остання кнопка у вигляді **синьої галочки** призначена для перевірки запиту на наявність синтаксичних помилок.

До **важливих кнопок**, які ще розміщені на цій панелі інструментів можна віднести:

- **Show Execution Plan** (показати імовірний план виконання) – дозволяє перед виконанням запиту переглянути процедуру його виконання, включаючи індекси, які приймають участь в пошуку даних.
- **Show Server Trace** (показати дійсний план виконання) - показує статистичну інформацію про виконання запиту, тобто час його виконання, кількість отриманих рядків, кількість помилок та попереджень тощо.
- **Show Client Statistics** (показати статистику клієнта) - перегляд статистики проведених запитів.
- **Results in Text** (у вигляді тексту). Такий вивід результату потрібен, якщо необхідно виводити рядки, в яких більше 128 символів.
- **Results in Grid** (у вигляді сітки). Визначає, що результат запиту буде виведений у вигляді таблиці. Пункт по замовчуванню.
- **Results in File** (у файл). Дозволяє занести результат запиту автоматично у файл з розширенням \*.rpt.

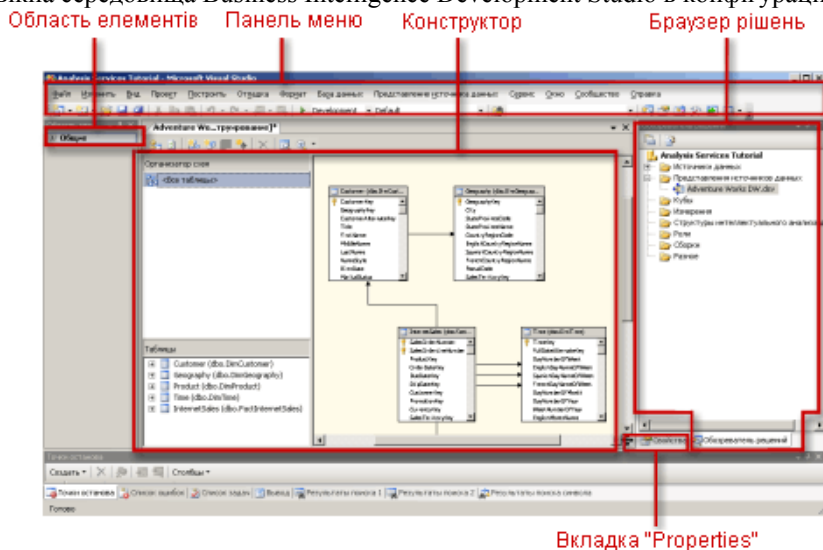
#### 4.4. SQL Server Business Intelligence Development Studio

**Business Intelligence Development Studio (BIDS)** – нова версія середовища Microsoft Visual Studio 2008 з додатковими типами проектів, які характерні для роботи з бізнес-аналітикою (business intelligence - BI) в SQL Server, наприклад: кубів, джерел даних, звітів і пакетів служб Integration Services. Це основне середовище для розробки бізнес-рішень, до складу яких входять проекти для служб Analysis Services, Integration Services і Reporting Services Services в програмах, написаних на C# і VB.NET. Кожен тип проектів містить шаблони для створення об'єктів, необхідних для рішень бізнес-аналітики, і надає різні типи конструкторів, засобів і майстрів для роботи з такими об'єктами.

Середовище Business Intelligence Development Studio включає ряд вікон для всіх етапів розробки рішень і управління проектами. Наприклад, до складу середовища BIDS входять вікна, які дозволяють кількома проектами як єдиним цілим, а також переглядати властивості об'єктів в проектах і змінювати їх. Ці вікна доступні для всіх видів проектів.



На рисунку показані вікна середовища Business Intelligence Development Studio в конфігурації по замовчуванню.



Середовище BIDS складається з 4-х основних вікон:

1. **Браузер рішень** – відображає всі рішення з їх проектами, включаючи всі елементи, які з ними зв'язані. З його допомогою можна створювати пусті рішення, куди потім додаються нові або існуючі проекти.
2. **Вікно «Properties» (Властивості)** відображає всі властивості об'єкта. Його можна використовувати для перегляду і зміни властивостей файлів, проектів та рішень.
3. **Вікно конструктора** являє собою вікно засобів, в якому можна створювати або змінювати об'єкти бізнес-аналітики. Конструктор надає два представлення одного об'єкта: коду і схеми. Варто відмітити, що вікно конструктора недоступно до тих пір, поки в рішення не буде добавлений проект, а в цьому проекті не буде відкритий який-небудь об'єкт.
4. **Область елементів** містить елементи, які можуть використовуватись в проектах бізнес-аналітики. Вкладки і елементи на цій області змінюються в залежності від того, який конструктор і редактор використовується в даний момент часу.

Інші вікна середовища BIDS дозволяють переглядати результати пошуку, отримувати відомості про повідомлення помилок і відомості, які виводяться відлагоджувачами або конструкторами проектів. **Браузер серверів** показує список з'єднань з базами даних. **Браузер об'єктів** показує символи, які можуть бути використані в проекті.

## 5. Робота з базою даних через утиліти та зберігаємі процедури

### 5.1. Створення БД

Роботу з баз даних, тобто її створення, модифікацію та видалення можна здійснювати як засобами SQL Management Studio, так і викликаючи певний набір зберігаємих процедур або SQL команд. Розглянемо всі ці випадки. І почнемо з самого простого, тобто з роботи з середовищем SQL Management Studio.

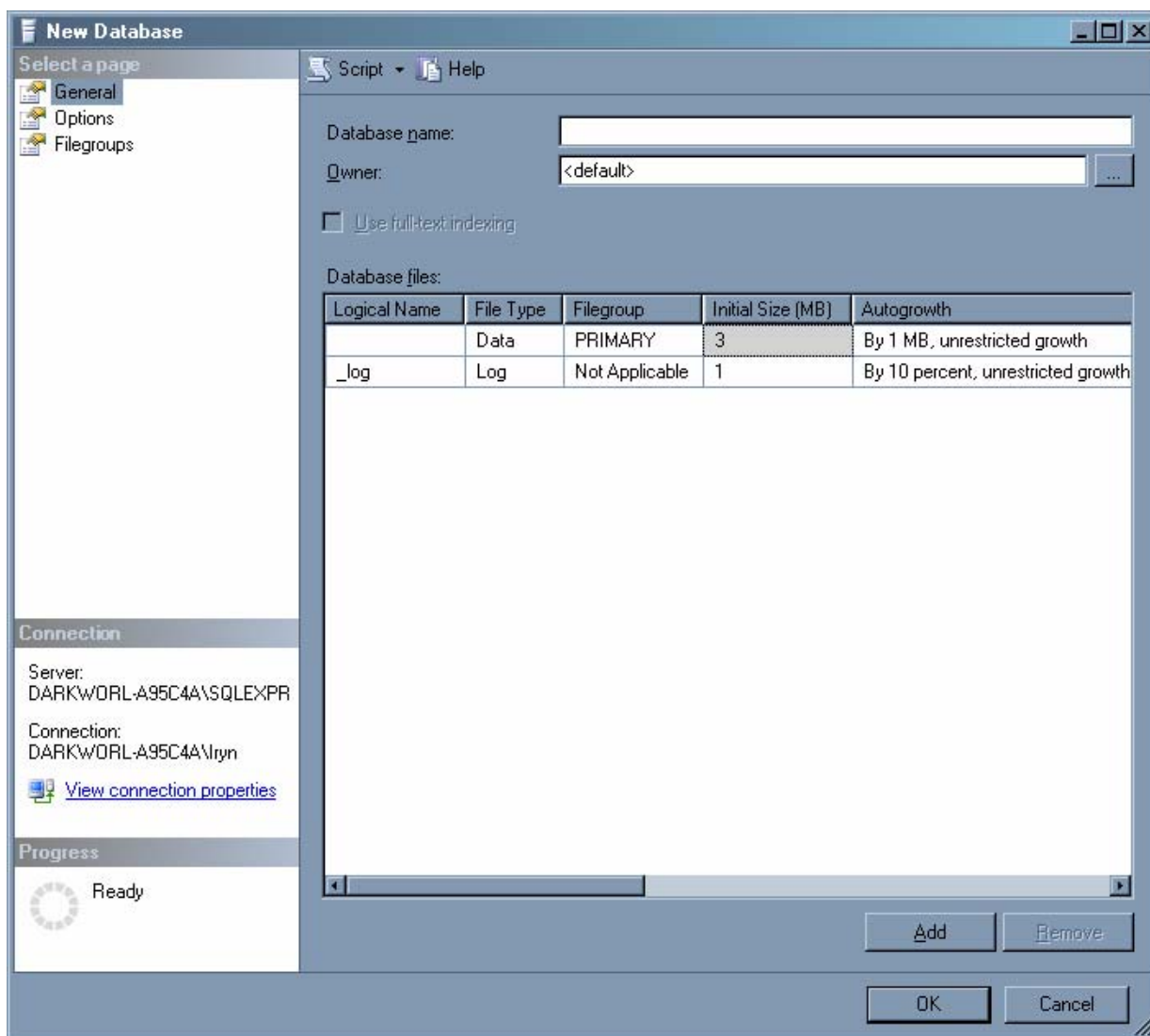
Отже, запускаємо SQL Server Management Studio, авторизуємось на сервері і викликаємо контекстне меню для папки Databases, в якому обираємо пункт «New Database» («Создать базу данных»). Діалогове вікно, яке з'явиться після цього, дозволить створити нову базу даних та налаштувати всі її параметри. Здійснюється це за допомогою **трьох закладок**: General, Options та Filegroups.

Зкладка **General** дозволяє вказати:

- ім'я майбутньої бази даних (поле **Database name**);
- власника нової бази даних (поле **Owner**). По замовчуванню – це **dbo (Data Base Owner)**, тобто поточний користувач. Щоб його змінити потрібно натиснути на кнопку **order** і обрати необхідного власника з зареєстрованого списку користувачів даного сервера;
- таблиця **Database files** дозволяє створити для нової бази даних кілька файлів, вказавши їх характеристики:
  - **Logical Name** визначає ім'я майбутнього файла бази даних;
  - **File Type** – тип файла: файл з даними (Data) чи файл журналу транзакцій (Log). Файл журналу транзакцій по замовчуванню носить ім'я у форматі «Ім'я\_створюваної\_бази\_Log.ldf». Його параметри налаштування такі ж, що і для файлів бази даних;
  - **Filegroup (група файлів)**. При створенні файла необхідно вказати, до якої групи файлів він належить. По замовчуванню обирається група файлів **PRIMARY**. Це говорить про те, що цей файл є первинним. **В первинному файлі зазвичай розміщуються всі системні таблиці**. Якщо база даних складається з кількох файлів, один з них все-рівно буде відноситись до групи файлів **PRIMARY** і має розширення **.mdf**. Іншим можна вказати іншу, власну групу файлів (створюється на закладці **Filegroups**).
  - **Initial Size** – розмір файла, який по замовчуванню рівний 3 Мб для файла бази даних і 1 Мб для журналу транзакцій.



- **Autogrowth (авторозширення)** - визначає як буде змінюватись розмір файлу, якщо початково встановленого розміру не вистачить. Налаштувати ці параметри можна за допомогою діалогового вікна, яке викликається за допомогою кнопки, розташованої в цьому полі. При цьому, прирощення можна заборонити, задати максимальний розмір файла (поле **Maximum file size**) або вказати темп росту файла:
  - **In Megabytes** - файл буде збільшуватись по вказаній праворуч кількості мегабайт;
  - **By Percent** – збільшення файла буде здійснюватись в процентному відношенні відносно початково вказаного розміру.
- **Location** відповідає за місцезнаходження файлів. По замовчуванню це папка **Data**, яка розміщується в %папка установки%\Microsoft SQL Server \MSSQL.



**New Database**

Select a page: General, Options, Filegroups

Script Help

Database name:

Owner:

☐ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
	Data	PRIMARY	3	By 1 MB, unrestricted growth
_log	Log	Not Applicable	1	By 10 percent, unrestricted growth

Connection

Server: DARKWORLD-A95C4A\SQLEXP

Connection: DARKWORLD-A95C4A\Iryn

[View connection properties](#)

Progress

Ready

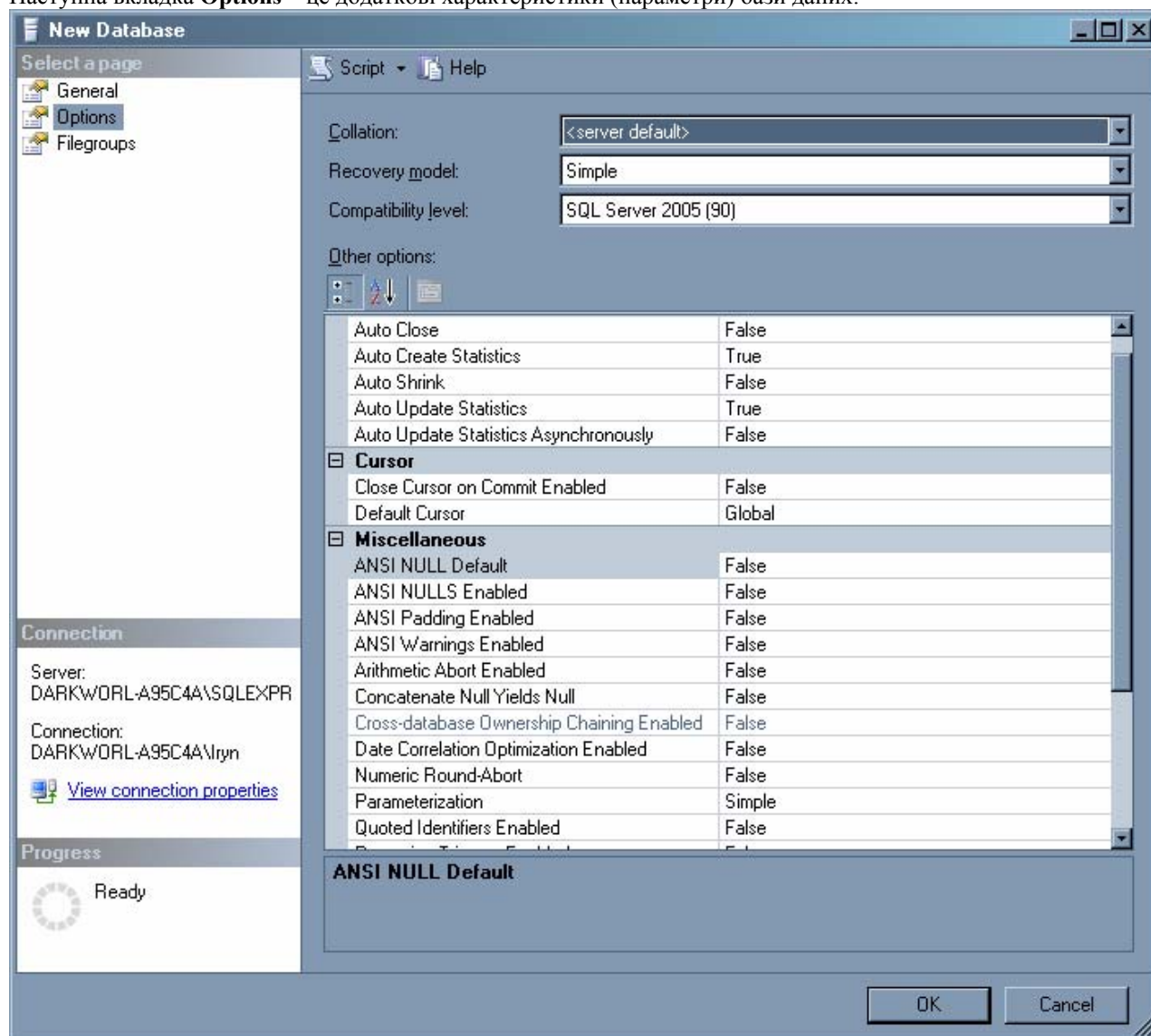
Add Remove

OK Cancel





Наступна вкладка **Options** – це додаткові характеристики (параметри) бази даних.



**New Database**

Select a page: General, **Options**, Filegroups

Script Help

Collation: <server default>

Recovery model: Simple

Compatibility level: SQL Server 2005 (90)

Other options:

Auto Close	False
Auto Create Statistics	True
Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False

☒ **Cursor**

Close Cursor on Commit Enabled	False
Default Cursor	Global

☒ **Miscellaneous**

ANSI NULL Default	False
ANSI NULLS Enabled	False
ANSI Padding Enabled	False
ANSI Warnings Enabled	False
Arithmetic Abort Enabled	False
Concatenate Null Yields Null	False
Cross-database Ownership Chaining Enabled	False
Date Correlation Optimization Enabled	False
Numeric Round-Abort	False
Parameterization	Simple
Quoted Identifiers Enabled	False

**ANSI NULL Default**

Connection:

Server: DARK\ORL-A95C4A\SQLEXP

Connection: DARK\ORL-A95C4A\Iryn

[View connection properties](#)

Progress: Ready

OK Cancel

Поле **Collation** (параметри сортування) визначає спосіб сортування даних в базі даних, а **Recovery model** (модель відновлення) визначає режим відновлення бази даних:

- Full – повне відновлення;
- Simple - просте (звичайне) відновлення;
- Bulk-logged - масове відновлення або з неповним протоколюванням.

Поле **Compatibility Level** вказує на рівень сумісності з іншими версіями SQL Server. Рівень 100 говорить про сумісність лише з версією SQL Server 2008, 90 - з версією SQL Server 2005; 80 - з версією SQL Server 2000. В БІД Букмук 2005 існував рівень сумісності 70, який використовувався для сумісності з SQL Server 7.0.

Далі йде список інших додаткових опцій, які умовно поділені на групи.

☐ **Automatic (автоматичні налаштування):**

- **Auto close** – автоматичне завершення роботи SQL Server, коли від'єднався останній користувач.
- **Auto create statistics** – статистичні відомості будуть створюватись автоматично.
- **Auto shrink** – відповідає за стиснення даних. Якщо опцію встановити, то як тільки невикористовувана частина загального об'єму бази даних буде менше 25%, буде здійснене автоматичне стиснення. Але воно ніяк не вплине на розмір бази даних, вказанної при її створенні.
- **Auto update statistics** – визначає автоматичне оновлення статистики.
- **Auto update statistics asynchronously** – вказує на те чи буде здійснюватись автоматичне асинхронне оновлення статистики.

☐ **Cursor (курсори):**

- **Close Cursor on Commit Enabled** – при завершенні транзакції всі відкриті курсори автоматично закриваються.
- **Default Cursor** – область бачення курсорів: глобальна (Global) чи локальна (Local).



❑ **Miscellaneous (пізнє):**

- **ANSI NULL default** – визначає, що значення полів нової таблиці будуть мати по замовчуванню значення NULL, якщо параметер NOT NULL не встановлений.
- **ANSI NULLS Enabled** – дозволяє або забороняє використання NULL значень.
- **ANSI Padding Enabled** – дозволити або заборонити ANSI заповнення.
- **ANSI Warnings Enabled** – дозволити або заборонити ANSI попередження.
- **Arithmetic Abort Enabled** - дозволити або заборонити переривання операції у випадку виникнення ситуації ділення на нуль.
- **Concatenate Null Yields Null (автоокруглення чисел)** – ця опція логічно співставна з операцією множення на нуль. Результат операції конкатенації будь-якої величини з значенням NULL завжди буде рівний NULL. Наприклад, 'Hello, World'+NULL = NULL.
- **Cross-database Ownership Chaining Enabled** – дозволити або заборонити довірену базу даних.
- **Date Correlation Optimization Enabled** – дозволити або заборонити оптимізацію кореляції дат.
- **Parametrization** – визначення параметрів: просте чи примусове.
- **Quoted Identifiers Enabled** – дає можливість вказати назву в подвійних лапках. Це необхідно, якщо назва поля співпадає з ключовим словом мови SQL або назва складається з кількох частин, через пропуск або в назві містяться лапки.
- **Recursive Triggers Enabled** – вказує на можливість створювати рекурсивні тригери.
- **Torn page detection** – можливість перевірки того чи здійснився запис розбитої на частини таблиці на жорсткий диск.

❑ **Recovery (відновлення),** в якому існує лише одна опція **Page Verify (перевірка сторінок)**, що визначає як буде здійснюватись перевірка сторінки бази даних:

- **TornPageDetection** – пошук частковозбереженої сторінки, тобто дозволяє визначити чи відбувся запис на диск розбитої на частини сторінки (форма пошкодження даних);
- **Checksum** – пошук пошкодженої сторінки по контрольній сумі;
- **None** – відсутня перевірка.

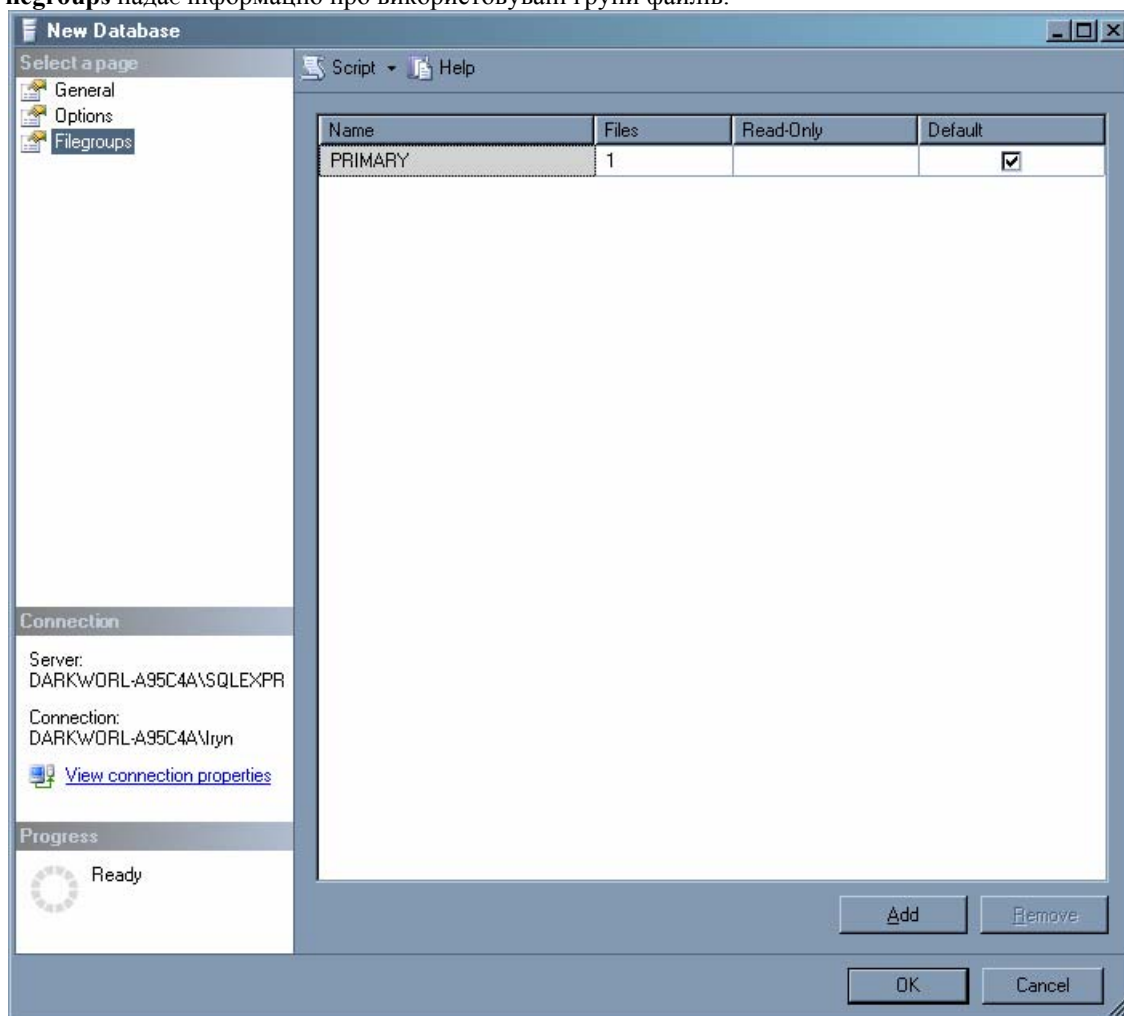
❑ **State (стан):**

- **Restrict Access** (обмеження доступу) - якщо дана опція встановлена, то до БД отримають доступ в певний момент часу всі користувачі (MultiUser), лише один (SingleUser) або ж доступ користувачів буде обмежений (RestrictedUser).
- **Database Read-only** вкаже на те, що БД буде доступна лише для читання.
- **Database State** – стан бази даних.
- **Encryption Enabled** - дозволити або заборонити шифрування даних.





Вкладка **Filegroups** надає інформацію про використовувані групи файлів.



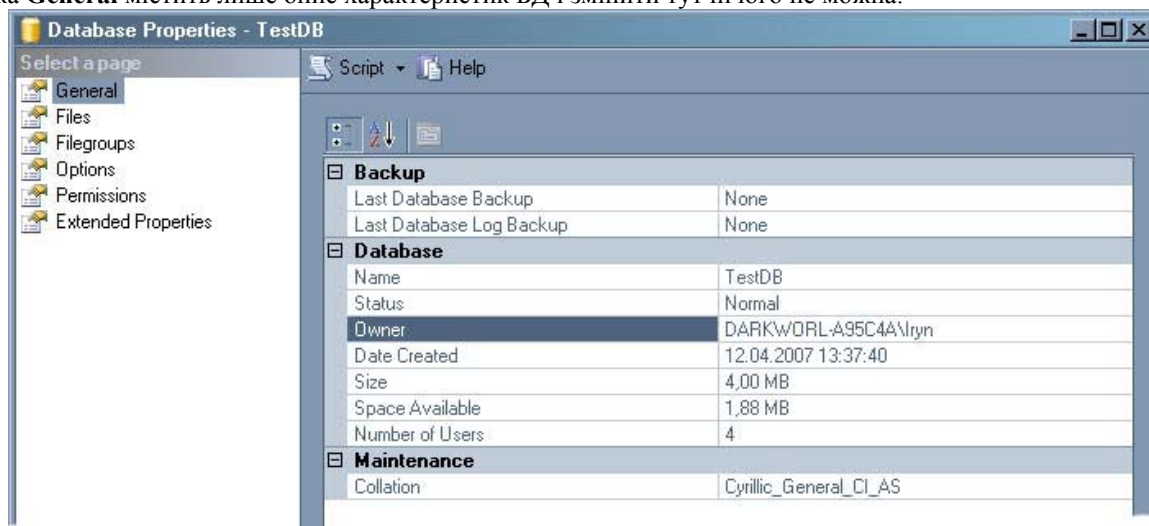
По замовчуванню, існує лише одна група файлів – PRIMARY, але ви можете додати свою групу, вказавши її назву, скільки максимально файлів в неї може входити, чи буде вона лише для читання і групою по замовчуванню для нових файлів баз даних. Якщо ж певна група файлів перестала відігравати функціональну роль, її можна видалити в цьому ж вікні.

## 5.2. Зміна властивостей БД

Якщо Вам необхідно змінити параметри бази даних після створення або змінити налаштування певного файла вже створеної бази даних, тоді слід в контекстному меню необхідної бази даних (в папці Databases) обрати пункт **«Properties» (Властивості)**.

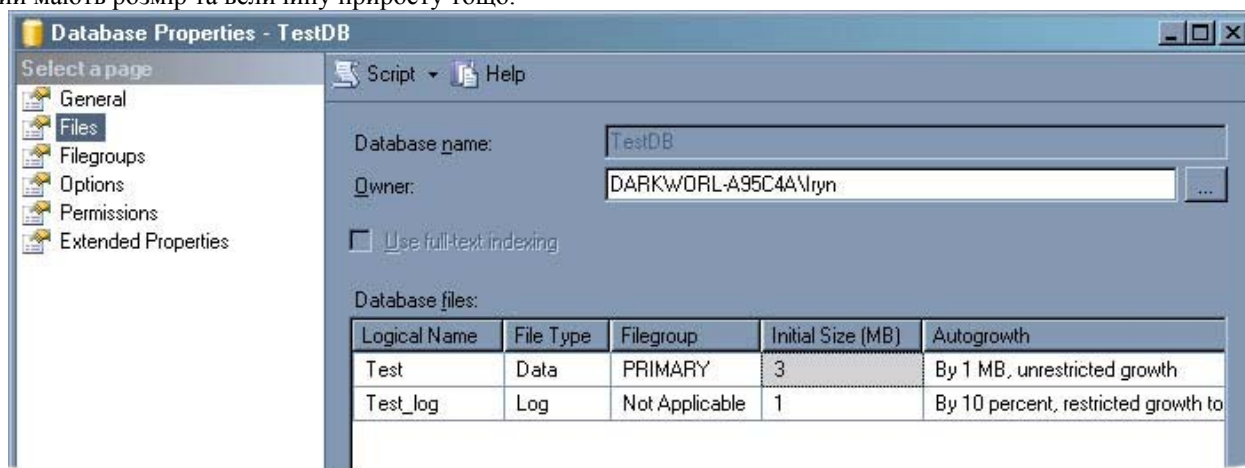
Це вікно властивостей дещо схоже з вікном створення бази даних, але має трішки більше вкладок:

а) Вкладка **General** містить лише опис характеристик БД і змінити тут нічого не можна.

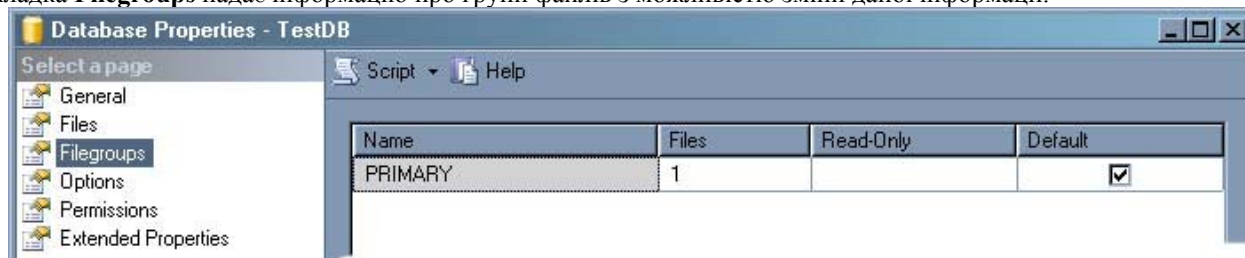




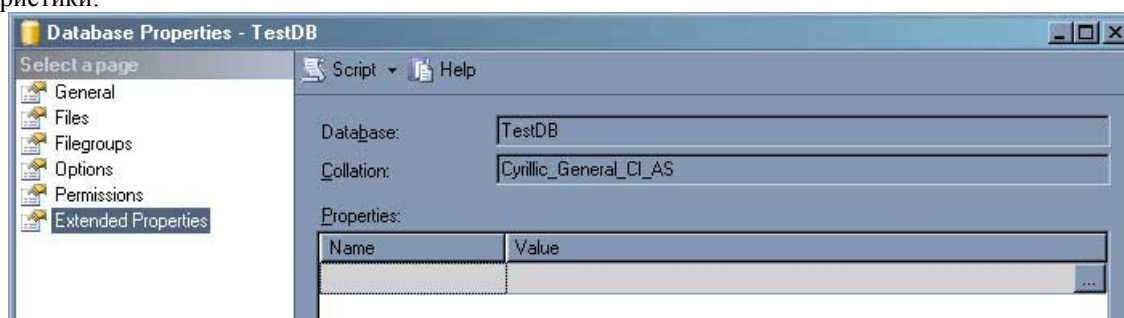
- b) Вкладка **Files** містить інформацію про власника поточної БД, з яких файлів вона складається, де вони розміщуються, який мають розмір та величину приросту тощо.



- c) Вкладка **Filegroups** надає інформацію про групи файлів з можливістю зміни даної інформації.

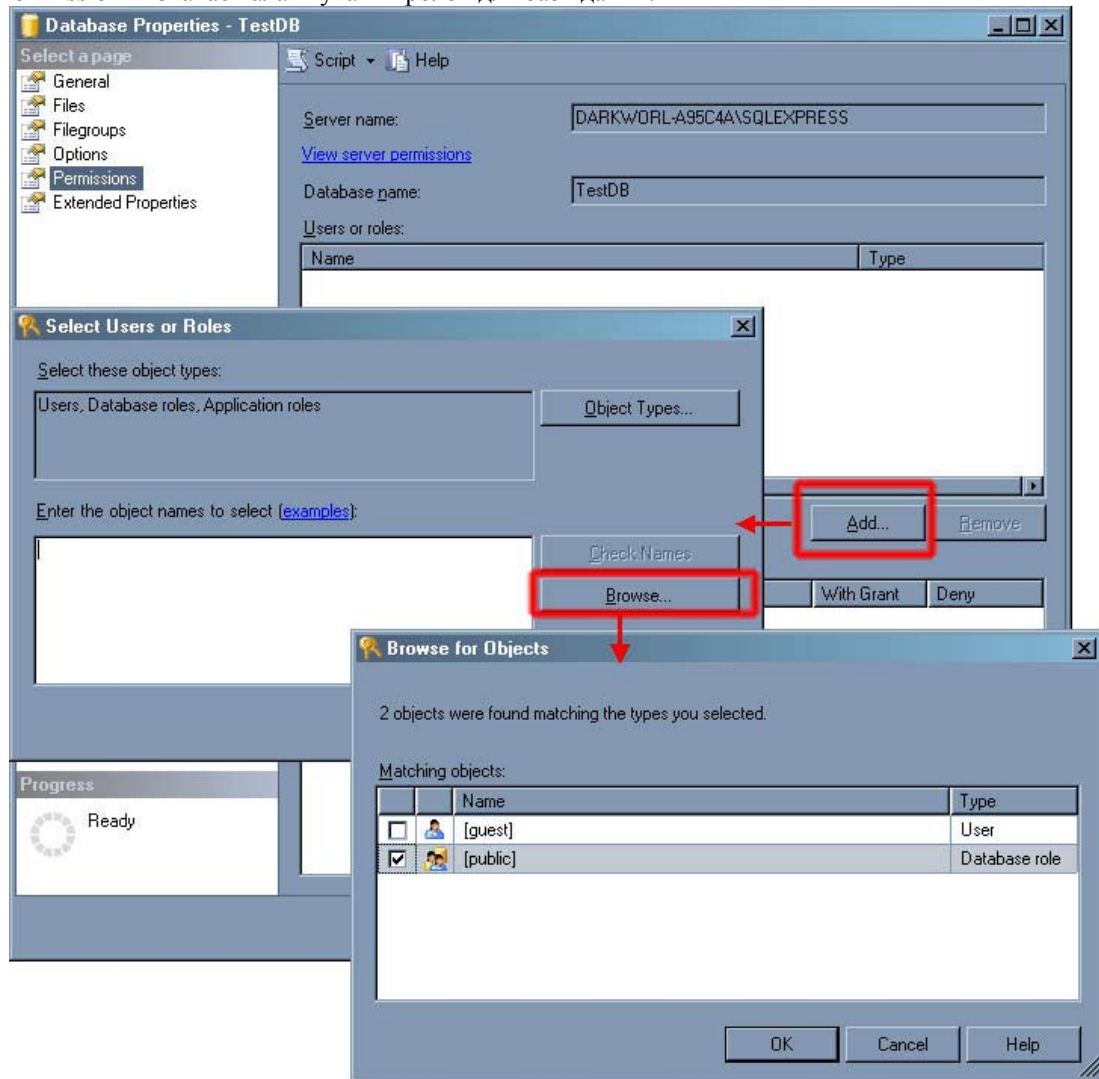


- d) Четверта вкладка **Options** відповідає за додаткові налаштування обраної бази даних (див. [Створення БД в SQL Server - Закладка Options](#)).
- e) Вкладка **Extended Properties** містить дані про назву бази даних, її параметри сортування та додаткові (користувачські) характеристики:

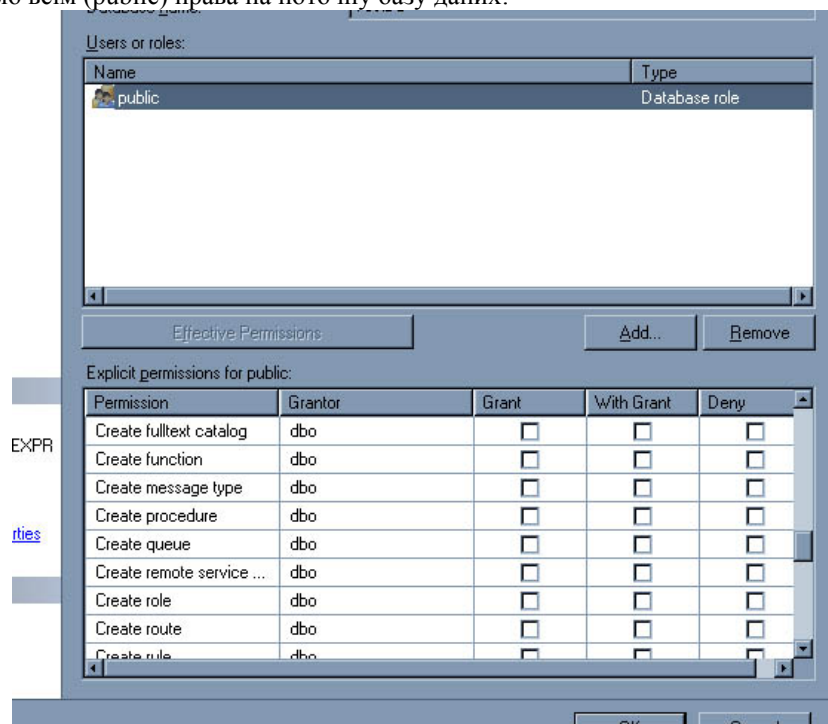




f) Вкладка **Permission** визначає налаштування ролей для бази даних.



В результаті ми надаємо всім (**public**) права на поточну базу даних:



Більш детально про права буде розглянуто наприкінці курсу MS SQL Server.



Переглянути та змінити властивості БД можна також за допомогою системних зберігаємих процедур (нагадаємо, що запуск зберігаємих процедур потрібно писати в редакторі SQL запитів). Для перегляду використовується системна зберігаєма процедура **sp\_helpdb**, а для зміни властивостей - **sp\_dboption**. Доречі, всі системні зберігаємі процедури в SQL Server мають префікс **sp\_xxx (stored procedure)**, а для запуску процедури використовується оператор **EXEC** (оскільки всі зберігаємі процедури вони є виконуваними).

```
/*EXEC sp_helpdb [імя_бази_даних]*/*
EXEC sp_helpdb          /*перегляд властивостей всіх баз даних*/
EXEC sp_helpdb TestDB   /*перегляд властивостей бази даних TestDB*/
```

Результат:

DARKWORL-A95C4...SQLQuery1.sql\* Summary

exec sp\_helpdb

Results Messages

	name	db_size	owner	dbid	created	status	compatibility_level
1	master	5.25 MB	sa	1	Apr 8 2003	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
2	model	3.19 MB	sa	3	Apr 8 2003	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
3	msdb	7.44 MB	sa	4	Oct 14 2005	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
4	MyTestDB	14.00 MB	DARKWORL-A95C4A\Iryn	8	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
5	Press	4.00 MB	DARKWORL-A95C4A\Iryn	7	Apr 4 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
6	pubs	5.50 MB	DARKWORL-A95C4A\Iryn	6	Apr 1 2007	NULL	80
7	tempdb	2.69 MB	sa	2	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90
8	TestDB	4.00 MB	DARKWORL-A95C4A\Iryn	5	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90

DARKWORL-A95C4...SQLQuery1.sql\* Summary

exec sp\_helpdb TestDB

Results Messages

	name	db_size	owner	dbid	created	status	compatibility_level
1	TestDB	4.00 MB	DARKWORL-A95C4A\Iryn	5	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	Test	1	c:\Program Files\Microsoft SQL Server\MSSQL.1\M...	PRIMARY	3072 KB	Unlimited	1024 KB	data only
2	Test_log	2	c:\Program Files\Microsoft SQL Server\MSSQL.1\M...	NULL	1024 KB	2147483648 KB	10%	log only

Синтаксис та приклад на зміну властивостей:

```
/*EXEC sp_dboption [імя_бази_даних] [, 'назва_властивості', ['TRUE' | 'FALSE']]*/
exec sp_dboption TestDB, 'Auto update statistics', 'TRUE'
```

В даному прикладі було змінено властивість бази даних «Auto update statistics» (автоматичне оновлення статистики) на значення «true», тобто ми його ввімкнули.

### 5.3. Переіменування БД

Щоб переіменувати БД засобами SQL Server Management Studio, необхідно в контекстному меню необхідної бази даних викликати пункт контекстного меню «**Rename**» (переіменувати).

Для переіменування бази даних також призначена системна зберігаєма процедура **sp\_renamedb**. Для виконання, запускаємо редактор SQL запитів та напишемо наступний запит на переіменування бази даних TestDB в NewTestDB:

```
/*EXEC sp_renamedb 'стапе_імя_бази_даних', 'нове_імя_бази_даних'*/
EXEC sp_renamedb 'TestDB', 'NewTestDB'
```

Але не слід забувати, що для того, щоб мати можливість переіменувати бази даних, ви повинні бути або членом групи системних адміністраторів SQL Server, або її власником (owner).

### 5.4. Видалення БД

Пункт «**Delete**» (видалити) знаходиться в контекстному меню кожної БД. Достатньо обрати його та підтвердити видалення. Слід відзначити, що відновленню видалена БД не підлягає, її можна відновити лише з резервної копії.

Щоб видалити певну базу даних, ви також повинні бути членом групи системних адміністраторів SQL Server або її власником (owner). Крім того, при видаленні потрібно знаходитись в базі даних master.



## 6. Введення в SQL. Що це таке? Категорії команд SQL: DDL, DML, DCL

SQL (Structured Query Language) – це універсальна комп'ютерна мова, яка використовується для створення, модифікації і управління даними в реляційних базах даних. Вато відміти, що SQL не являється мовою програмування. Унікальність даної мови полягає в тому, що вона є єдиною стандартною мовою баз даних. Мову SQL підтримують більше сотні СУБД. І саме головне: не дивлячись на те, що кожна СУБД по-своєму інтерпретує стандарт (вносить свої доповнення), набір стандартних інструкцій SQL підтримують всі бази даних.

Щодо історії, то її початок мови SQL був закладений у 1974 році, коли компанією IBM для експериментальної реляційної СУБД System R була розроблена спеціальна мова SEQUEL (Structured English QUery Language, **структурована англійська мова запитів**), яка дозволяла дуже просто управляти даними в базі даних. Розробкою займалися Дональд Чемберлін і Рей Бойс. Ціллю розробки SEQUEL було створення мови, якою міг би користуватись звичайний користувач, який не має досвіду програмування.

Пізніше мова SEQUEL була переіменована в SQL і в 1986 р. вийшов її перший стандарт ANSI (American National Standards Institute), а в 1987 р. стандарт ISO (International Organization for Standardization, Міжнародна організація по стандартизації). Але вона не була єдиною мовою для баз даних. В той же час Каліфорнійський університет Берклі розробив власну мову QUEL для некомерційної реляційної СУБД Ingres (прародич сьогоднішньої PostgreSQL). Та, на жаль, вона не витримала конкуренцію з SQL.

Таким чином, в 1987 р. SQL стала міжнародною мовою баз даних (SQL-86), в 1992 р. вийшла друга розширена версія даного стандарту (ANSI SQL-92 або SQL2), а в 1999 р. третя версія (SQL:1999, SQL-99 або SQL3). Сьогодні діє стандарт, який був прийнятий в 2003 р. (SQL:2003) з невеликими змінами, які були внесені в 2006 та 2008 роках. Доречі, в новий стандарт SQL була добавлена підтримка роботи і з XML даними.

Робота з SQL нескладна. У випадку, якщо користувачу необхідно отримати дані з бази даних, він звертається з запитом до СУБД на отримання цієї інформації. В результаті, СУБД обробляє даний запит і повертає користувачу результуючі дані. Фактично, **запит** – це команда, написана на мові SQL.

За допомогою SQL можна:

- створювати, модифікувати або видаляти об'єкти бази даних та саму базу даних;
- встановлювати зв'язки між об'єктами бази даних;
- здійснювати вибірку даних;
- вставляти, модифікувати та виділяти дані з бази даних;
- забезпечувати цілісність даних
- інше.

Вищеописані дії та ще багато інших, здійснюються за допомогою інструкцій (команд) SQL, які ми будемо з Вами вивчати на протязі наступних уроків.

Всі команди мови SQL поділяють на **чотири категорії**:

- 1) **DDL (Data Definition Language - Мова Визначення Даних)** – включає в себе інструкції, які призначені для створення, модифікації та виділення об'єктів бази даних (таблиці, представлення, індекси тощо).
- 2) **DML (Data Manipulation Language - Мова Маніпулювання Даними)** – це набір інструкцій, які призначені для роботи з даними в базі даних, тобто їх вставкою, видаленням, зміною та вибіркою.
- 3) **DCL (Data Control Language - Мова Управління Даними)** – містить набір інструкцій, які управляють правами доступу користувачів до об'єктів бази даних.
- 4) **TCL (Transaction Control Language – Мова Управління Транзакціями)**. В неї вносять набір інструкцій, що дозволяють управляти транзакціями.

Як вже було сьогодні сказано, MS SQL Server для запитів використовує спеціальну різновидність SQL – **Transact-SQL (T-SQL)**, яка була створена сумісно Microsoft і Sybase. Transact-SQL являється реалізацією стандарту ANSI/ISO SQL з рядом розширень, вивченню яких ми присвяtimo окрему пару.

## 7. Створення бази даних засобами SQL

Для створення бази даних в SQL існує оператор **CREATE DATABASE**, синтаксис якого має наступний вигляд:

```
CREATE DATABASE ім'я_БД
    [ON PRIMARY] /*група файлів*/
(
    NAME = логічне_ім'я_для_файла_на_диску,
    FILENAME = 'адреса_та_ім'я_файла_БД'
    [, SIZE = розмір_файла_БД ]
    [, MAXSIZE = максимальний_розмір_файлу | UNLIMITED ]
    [, FILEGROWTH = інкремент_автоматичного_збільшення_файла ]
)
[, ... n ] /*опис наступного файлу бази даних*/
```





```
LOG ON      /*файл журналу транзакцій*/
( NAME = логічне_ім'я_для_файла_журналу_транзакцій_на_диску,
  FILENAME = 'адреса_та_ім'я_файла_журналу_транзакцій'
[, SIZE = розмір_файла_журналу_транзакцій ]
[, MAXSIZE = максимальний_розмір_файлу | UNLIMITED ]
[, FILEGROWTH = інкремент_автоматичного_збільшення_файла ]
)
[COLLATE спосіб_сортування_даних_в_БД ]
[FOR LOAD ] /*використовується лише для сумісності з SQL Server 6.5*/
| [FOR ATTACH] /*дозволяє перекомпонувати набір вже створених файлів в єдину БД*/
```

Початковий, максимальний розмір файла бази даних та інкремент автоматичного збільшення бази даних можна задавати як в мегабайтах, так і в кілобайтах. Якщо початковий розмір не вказувати, тоді по замовчуванню він встановлюється рівний розміру БД **model**. Якщо не задати максимальний розмір, то по замовчуванню приймається параметр **UNLIMITED**. Величина автоматичного збільшення розміру файлу по замовчуванню рівна 1 Мб.

Наприклад, створимо базу даних з іменем MyTestDB, яка має аналогічну назву файла, в якому зберігаються її дані. База даних складається з одного файла, які входять до групи файлів PRIMARY та має початковий розмір файлу – 12 Мб, максимальний – 20 Мб, а величину прирощення при нестачі розміру 2 Мб. Початкова величина журналу транзакцій – 2 Мб, а максимальна – 12 Мб.

Створювана база даних використовує кодову сторінку Latin1 або 1251 та має алфавітний порядок сортування даних (General) – регістрозалежний (CI) та індексозалежний (AI).

```
create database MyTestDB on primary
(name = MyTestDB,
 filename = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\MyTestDB.mdf',
 size = 12MB,
 maxsize = 20MB,
 filegrowth = 2MB )
LOG ON
(name = MyTestDB_Log,
 filename = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\MyTestDB_Log.ldf',
 size = 2MB,
 maxsize = 12MB,
 filegrowth = 1MB)
COLLATE Latin1_General_CI_AI;
```

Після запуску даного запиту, створиться нова база даних з вищеописаними параметрами.

## 8. Модифікація та видалення бази даних

Для модифікації вже створеної бази даних (зміни розмірів, додавання нових файлів в БД тощо) використовується оператор SQL **ALTER DATABASE**:

```
ALTER DATABASE ім'я_БД
ADD FILE опис_файла [, ... n]          /*додати новий файл*/
[TO FILEGROUP ім'я_групи_файлів_в_яку_додати_новий_файл]
| ADD LOG FILE опис_файла [, ... n]    /*додає новий файл журналу транзакцій в БД*/
| REMOVE FILE логічне_ім'я_файла       /*видалити файл*/
| ADD FILEGROUP ім'я_групи_файлів       /*додати нову групу файлів*/
| REMOVE FILEGROUP ім'я_групи_файлів    /*видалити групу файлів і всі файли, що входять
                                         до її складу*/
| MODIFY FILE опис_файла                /*змінити властивості файлу БД*/
| MODIFY NAME = нове_ім'я               /*переіменувати БД*/
| MODIFY FILEGROUP ім'я_групи_файлів, властивість_групи_файлів /*змінити властивості
                                         групи файлів, включаючи параметри READONLY,
                                         READWRITE і DEFAULT*/
| SET доступ, ...n WITH завершення      /*задає параметри READONLY, READWRITE і DEFAULT
                                         для БД. Параметр завершення визначає момент переходу до
                                         початкового стану та може приймати два значення:
                                         * ROLLBACK AFTER n SECONDS
                                         * ROLLBACK IMMEDIATE */
| COLLATE спосіб_сортування_даних
;
```



```

/*=====*/
опис_файла:
( NAME = логічне_ім'я_для_файла_на_диску,
  FILENAME = 'адреса_та_ім'я_файла_БД'
[, SIZE = розмір_файла_БД ]
[, MAXSIZE = максимальний_розмір_файлу | UNLIMITED ]
[, FILEGROWTH = інкремент_автоматичного_збільшення_файла ]
)

```

Наприклад, додамо до існуючої бази даних MyTestDB ще один файл з іменем MyTestDB2, який має початковий розмір – 5 Мб, максимальний – 20 Мб, а величину прирощення при нестачі розміру 1 Мб.

```

alter database MyTestDB
add file
( name = MyTestDB2,
  filename = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\MyTestDB2.mdf',
  size = 5,
  maxsize = 20MB,
  filegrowth = 1MB
);

```

Щоб впевнитись, що до бази даних MyTestDB був добавлений ще один файл, переглянемо її властивості:

DARKWORL-A95C4...- Module1.sql\* Summary

exec sp\_helpdb MyTestDB

Results Messages

	name	db_size	owner	dbid	created	status	compatibility
1	MyTestDB	19.00 MB	DARKWORL-A95C4A\Iryn	8	Apr 12 2007	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	90

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	MyTestDB	1	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSS...	PRIMARY	12288 KB	20480 KB	1024 KB	data only
2	MyTestDBLog	2	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSS...	NULL	2048 KB	12288 KB	1024 KB	log only
3	MyTestDB2	3	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSS...	PRIMARY	5120 KB	20480 KB	1024 KB	data only

Як видно з рисунку, база даних MyTestDB дійсно має новий файл, в зв'язку з чим і розмір бази даних збільшився з 14 до 19 Мб.

Для видалення бази даних використовується SQL оператор **DROP DATABASE**:

```
DROP DATABASE ім'я_БД, ім'я_БД2 , ...
```

Для прикладу, видалимо піддослідну базу даних MyTestDB:

```
DROP DATABASE MyTestDB;
```

## 9. Імпортування даних

Для імпортування та експортування даних в MS SQL Server 2000 використовувалась утиліта **Enterprise Manager**, в головному меню якої потрібно обрати пункт **Tools/Data Transformation Services/Import Data**. При цьому повинен бути обраний той сервер, в який Ви збираєтесь імпортувати дані.

Починаючи з SQL Server 2005 за імпорт та експорт даних відповідає пакет служб **SQL Server Integration Services**, а всі налаштування цієї служби переміщення даних (extract, transformation and load - ETL) визначені в папці **Data Transformation Services**. Запустити цю службу можна одним з п'яти способів:

1. З меню «Пуск->Програми->Microsoft SQL Server 2008->Import and Export Data».
2. В вікні командного рядка запустити програму **DTSWizard.exe**, яка знаходиться в папці «C:\Program Files\Microsoft SQL Server\100\DTS\Binn».
3. В середовищі Business Intelligence Development Studio потрібно викликати контекстне меню папки «**SSIS Packages**» (пакети служб SSIS), а потім обрати пункт «**SSIS Import and Export Wizard**» (майстер імпорту і експорту служб SSIS).

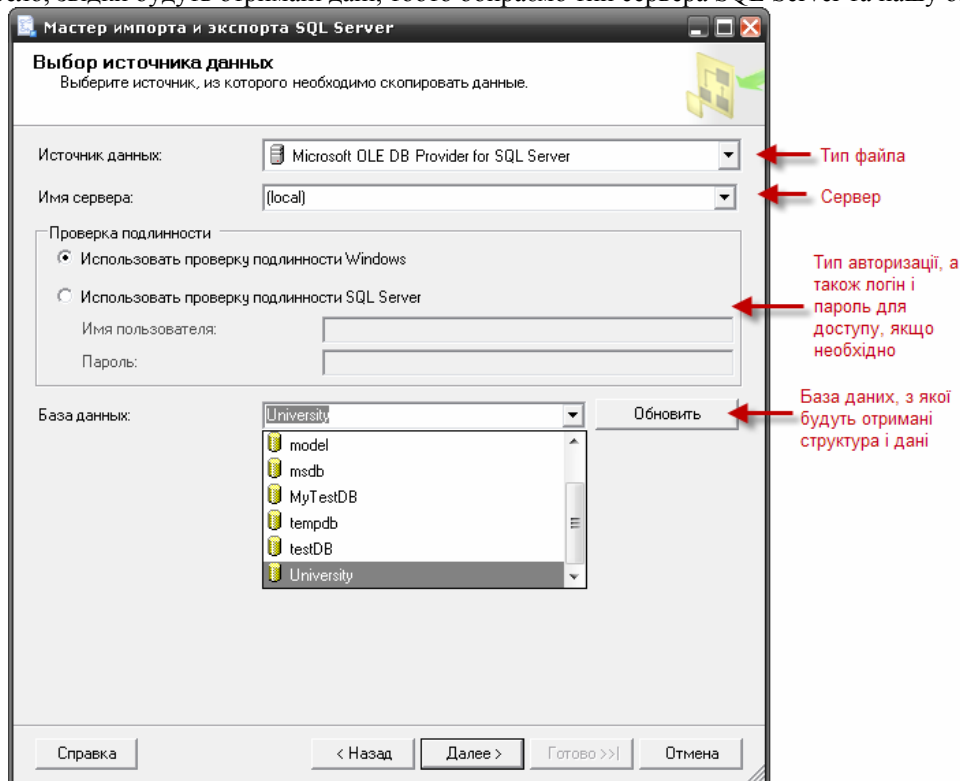




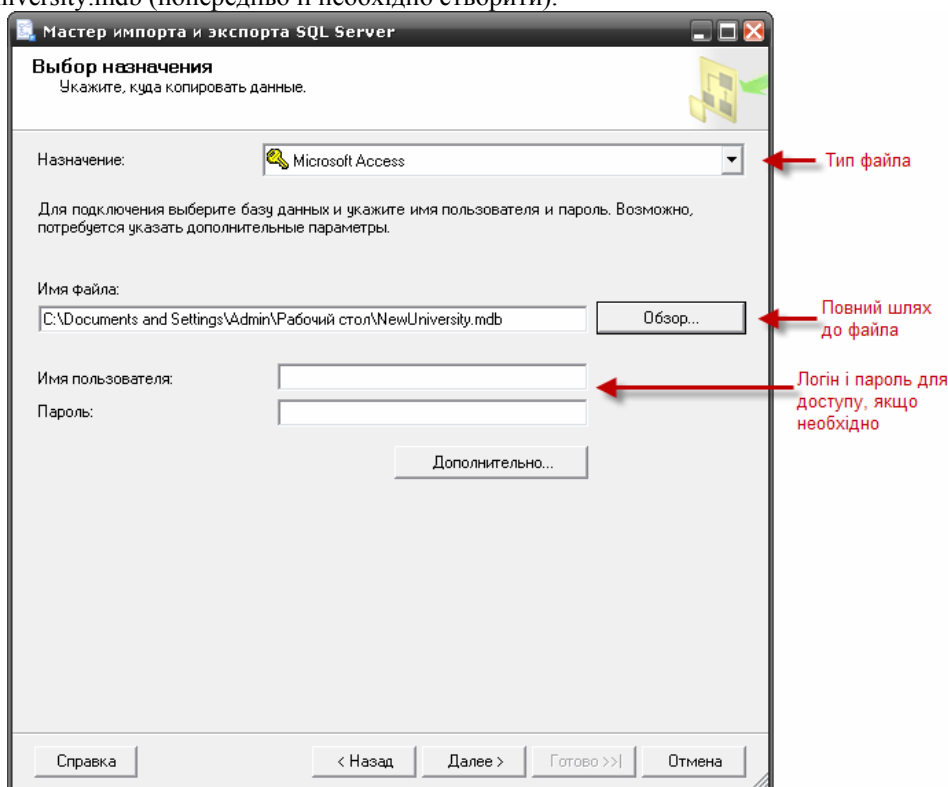
4. В среде Business Intelligence Development Studio в меню **«Project»** (проекты) оберіть пункт **«SSIS Import and Export Wizard»** (майстер імпорту і експорту служб SSIS).
5. В среде SQL Server Management Studio підключіться до сервера типа Database Engine, розгорніть бази даних, викличіть пункт контекстного меню необхідної бази даних **«Tasks»** (задачі), потім оберіть пункт **«Import Data»** (імпорт даних) або **«Export data»** (експорт даних).

Отже, спробуємо експортувати дані деякої бази даних University:

1. Запускаємо одним з вищезазначених способів майстер імпорту та експорту даних.
2. Обираємо джерело, звідки будуть отримані дані, тобто обираємо тип сервера SQL Server та нашу базу даних.

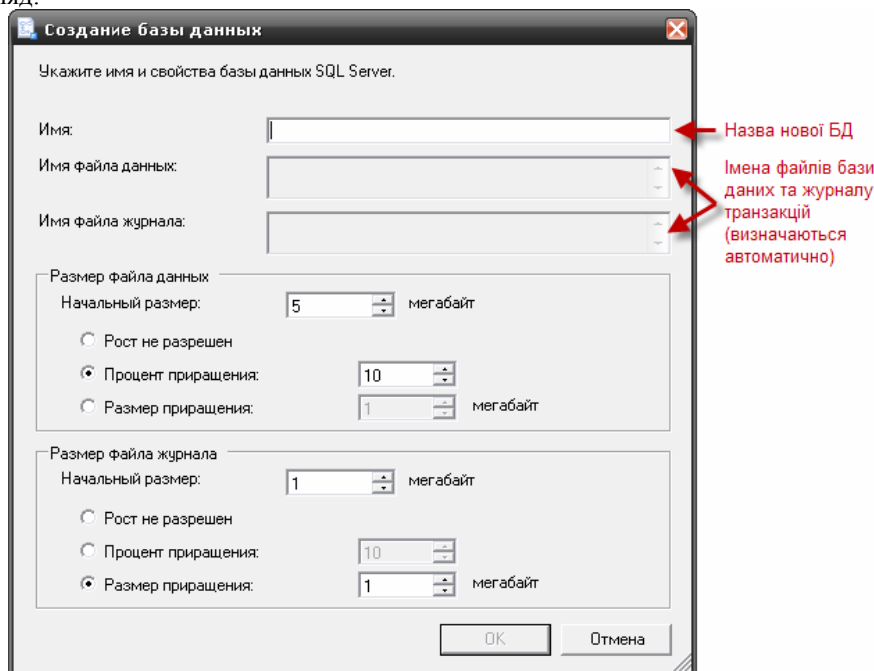


3. Вказуємо, куди будуть експортовані дані. Для прикладу, екпортуємо нашу базу даних в формат MS Access, а саме в базу даних NewUniversity.mdb (попередньо її необхідно створити).



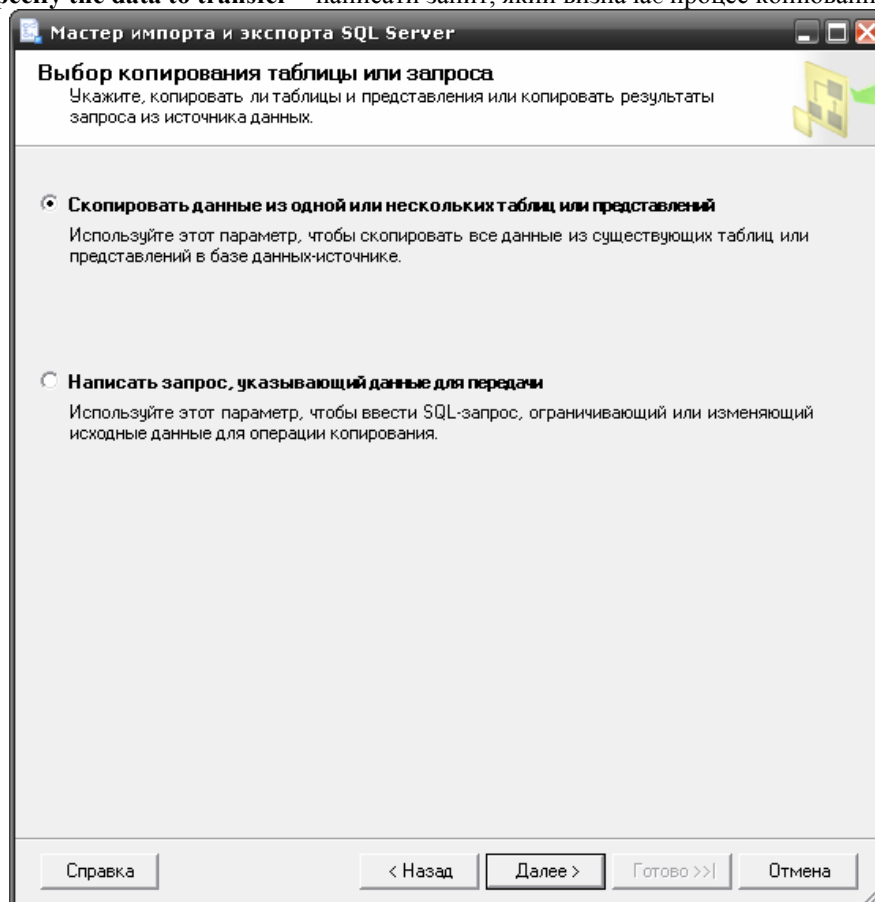


При імпорті даних в базу даних MS SQL Server вікно дещо зміниться: з'явиться додаткова кнопка «Create» (створити), яка дозволить створити нову базу даних, в яку будуть поміщені дані. Діалогове вікно створення нової бази даних буде мати наступний вигляд:



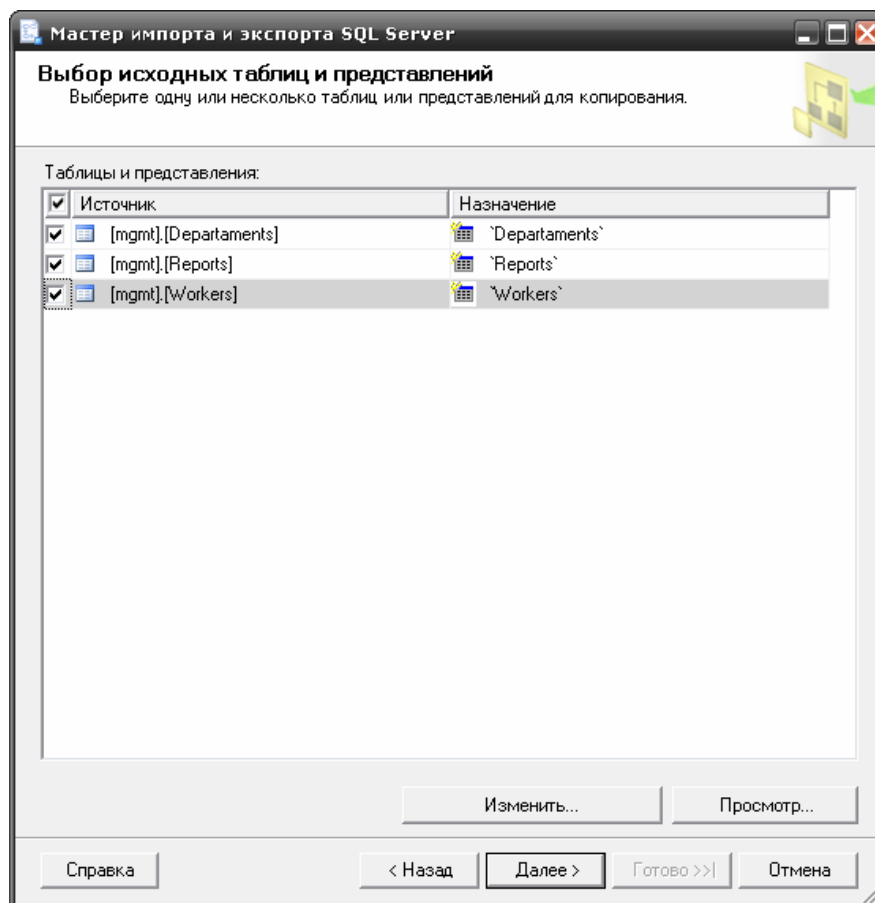
4. На наступному кроці Ви повинні визначити, що саме повинно бути скопійовано:

- **Copy table(s) and view(s) from the source database** – скопіювати таблиці і представлення з БД, вказаної в ресурсах;
- **Use a query to specify the data to transfer** – написати запит, який визначає процес копіювання.

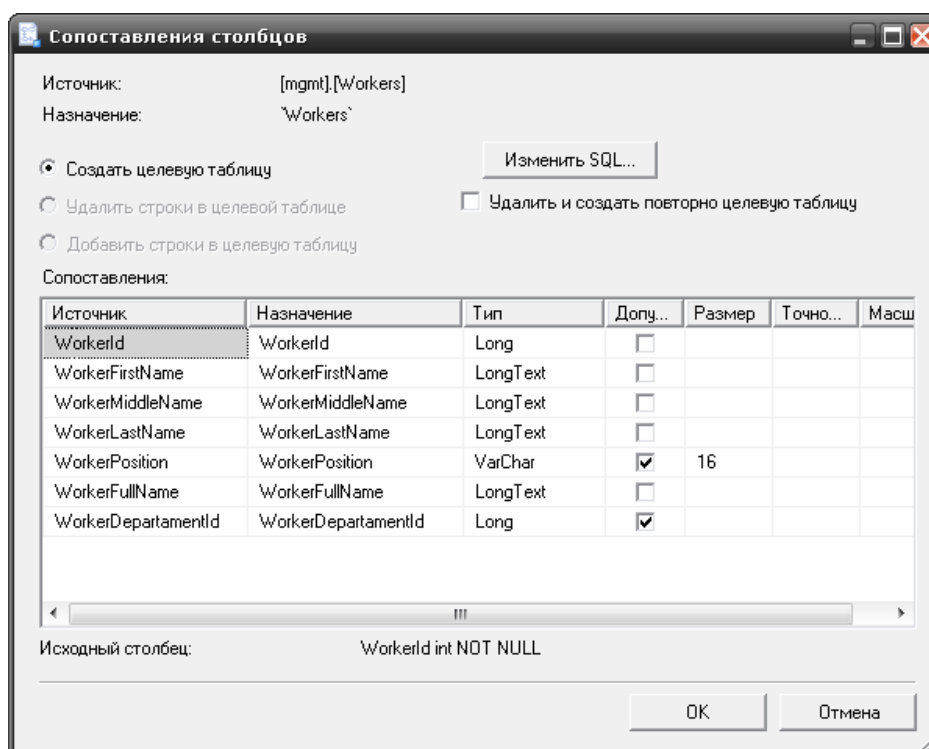




5. Якщо ви обрали перший пункт (а саме так ми і робимо), тоді на наступному кроці вам запропонують список експортуємих таблиць і представлень. Непотрібні таблиці ви можете відключити, знявши відповідний флажок в списку.



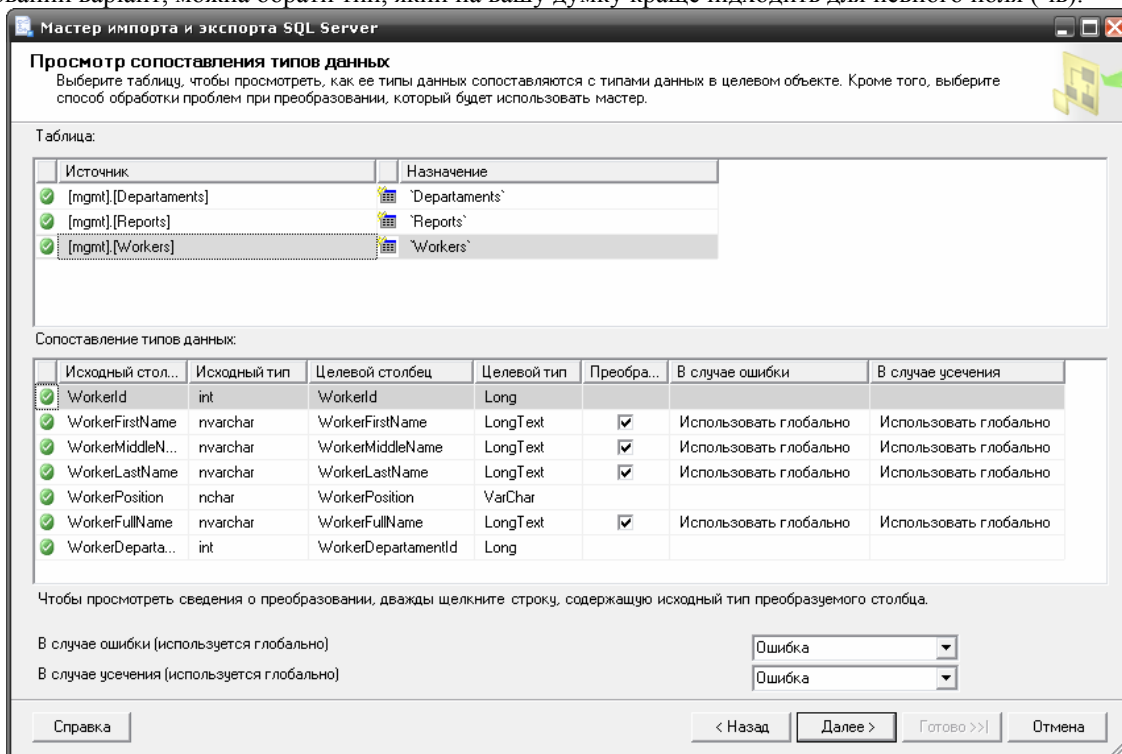
При цьому, якщо вас не влаштовує ім'я таблиці, яке було в експортуємій базі даних, ви можете його змінити разом з назвою полів або пропонуємими іншими характеристиками. Для цього необхідно натиснути на кнопку «**Transform...**» (змінити).



Кнопка «**Preview...**» (перегляд) дозволить переглянути вміст експортуємої таблиці.



6. Переглянути як будуть конвертовані типи даних з таблиць експортованої бази даних. Якщо Вас не влаштовує запропонований варіант, можна обрати тип, який на вашу думку краще підходить для певного поля (-ів).



**Мастер импорта и экспорта SQL Server**

**Просмотр сопоставления типов данных**  
Выберите таблицу, чтобы просмотреть, как ее типы данных сопоставляются с типами данных в целевом объекте. Кроме того, выберите способ обработки проблем при преобразовании, который будет использовать мастер.

Таблица:

	Источник	Назначение
<input checked="" type="checkbox"/>	[mgmt].[Departments]	'Departments'
<input checked="" type="checkbox"/>	[mgmt].[Reports]	'Reports'
<input checked="" type="checkbox"/>	[mgmt].[Workers]	'Workers'

Сопоставление типов данных:

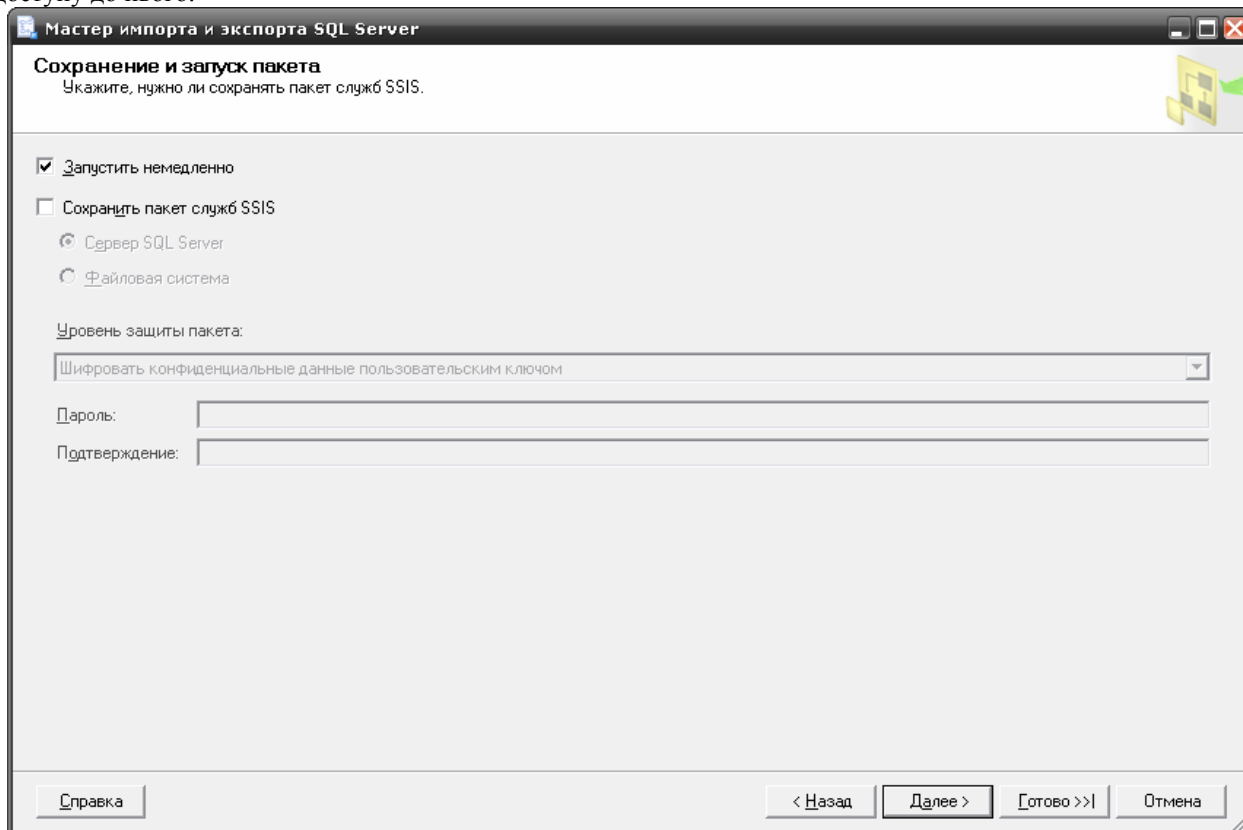
	Исходный столбец	Исходный тип	Целевой столбец	Целевой тип	Преобра...	В случае ошибки	В случае усечения
<input checked="" type="checkbox"/>	WorkerId	int	WorkerId	Long			
<input checked="" type="checkbox"/>	WorkerFirstName	nvarchar	WorkerFirstName	LongText	<input checked="" type="checkbox"/>	Использовать глобально	Использовать глобально
<input checked="" type="checkbox"/>	WorkerMiddleN...	nvarchar	WorkerMiddleName	LongText	<input checked="" type="checkbox"/>	Использовать глобально	Использовать глобально
<input checked="" type="checkbox"/>	WorkerLastName	nvarchar	WorkerLastName	LongText	<input checked="" type="checkbox"/>	Использовать глобально	Использовать глобально
<input checked="" type="checkbox"/>	WorkerPosition	nchar	WorkerPosition	VarChar			
<input checked="" type="checkbox"/>	WorkerFullName	nvarchar	WorkerFullName	LongText	<input checked="" type="checkbox"/>	Использовать глобально	Использовать глобально
<input checked="" type="checkbox"/>	WorkerDeparta...	int	WorkerDepartmentId	Long			

Чтобы просмотреть сведения о преобразовании, дважды щелкните строку, содержащую исходный тип преобразуемого столбца.

В случае ошибки (используется глобально)

В случае усечения (используется глобально)

7. Нарешті останній суттєвий крок, в якому необхідно сказати, чи ви хочете здійснити експортування негайно («**Run immediately**») і чи необхідно зберігати пакет служб SQL Server Intergration Services («**Save SSIS Package**») в SQL Server або ж у файловій системі. Тут ви також можете вказати на рівень захисту зберігаємого пакета, вказавши пароль для доступу до нього.



**Мастер импорта и экспорта SQL Server**

**Сохранение и запуск пакета**  
Укажите, нужно ли сохранять пакет служб SSIS.

☒ Запустить немедленно

☐ Сохранить пакет служб SSIS

☒ Сервер SQL Server

☐ Файловая система

Уровень защиты пакета:

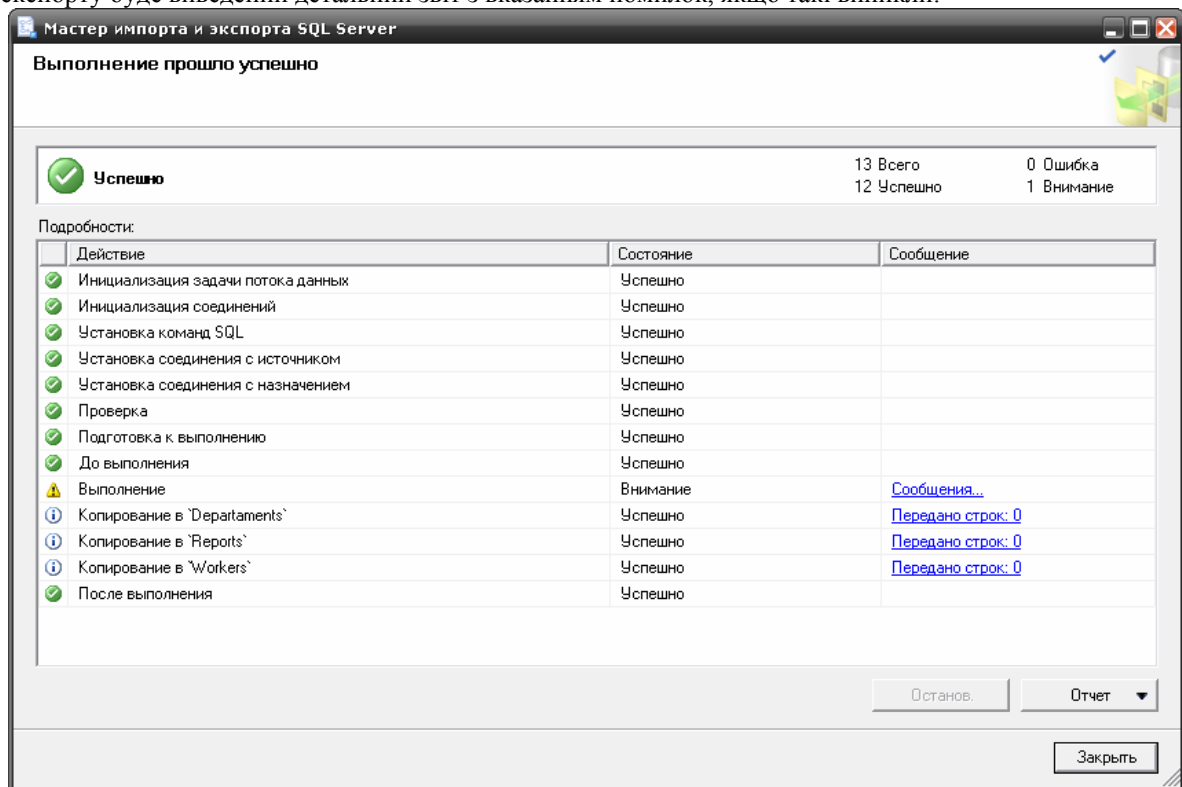
Пароль:

Подтверждение:

8. Далі вам запропонують переглянути значення, обрані за допомогою візарда і запуститься процес експорту.



9. В кінці експорту буде виведений детальний звіт з вказанням помилок, якщо такі виникли:



От і все. Якщо після імпорту бази даних ви її не бачите на сервері, тоді слід просто оновити список баз даних на сервері шляхом виклику контекстного меню папки Databases і вибору пункту «Refresh» (Оновити).

## 10. Домашнє завдання

1. Встановити SQL Server 2008 довільної редакції.
2. Створити базу даних TestDatabase, яка складається з двох файлів, розміром 2 та 3 Mb відповідно.
3. Спробувати імпортувати та експортувати довільні бази даних.