

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Методичні вказівки
до виконання лабораторної роботи за темою «Реалізація простого
клієнтського застосунку для роботи з базою даних Microsoft SQL
Server»

для студентів, що навчаються за спеціальностями
«121 Інженерія програмного забезпечення»
«122 Комп'ютерні науки»
«126 Інформаційні системи та технології»

Затверджено
редакційно-видавничою
радою університету,
протокол № 2 від 16.06.2023 р.

Харків
НТУ ХПІ
2023

Методичні вказівки до виконання лабораторної роботи за темою «Реалізація простого клієнтського застосунку для роботи з базою даних Microsoft SQL Server» для студентів, що навчаються за спеціальностями 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 126 «Інформаційні системи та технології» / Уклад. Орловський Д.Л., Копп А.М. – Харків: НТУ «ХПІ», 2023. – 22 с.

Укладачі Д.Л. Орловський,
А.М. Копп Гринченко М.А.

Рецензент Гринченко М.А.

Кафедра програмної інженерії та інтелектуальних технологій управління

ЗМІСТ

Вступ.....	4
1 Теоретичні відомості.....	5
1.1 SQL і прикладне програмне забезпечення	5
1.2 Технологія ODBC та її особливості	8
2 Виконання роботи	13
2.1 Створення джерела даних ODBC	13
2.2 Використання СУБД Microsoft Access як клієнтського застосунку .	14
2.3 Збереження результатів роботи	18
3 Вимоги до звіту.....	19
4 Питання для самоперевірки.....	20
Список літератури	22

ВСТУП

Сучасні інформаційні системи ґрунтуються на використанні баз даних, в яких накопичується різна інформація. Тому зараз розробляються і значно поширюються методи і засоби роботи з базами даних з метою підвищення ефективності роботи людини у різних галузях діяльності. Ці засоби та методи пов'язані з узагальненням і різними додатковими способами обробки даних. Основні ідеї сучасної інформаційної технології базуються на концепції, згідно з якою дані повинні бути організовані в бази даних з метою адекватного відображення реального світу, що змінюється, і задоволення інформаційних потреб користувачів. Ці бази даних створюються і функціонують під управлінням спеціальних програмних комплексів, які називають системами управління базами даних (СУБД).

Методичні вказівки до лабораторної роботи за темою «Реалізація простого клієнтського застосунку для роботи з базою даних Microsoft SQL Server» призначені для студентів, що навчаються за спеціальностями «121 Інженерія програмного забезпечення», «122 Комп'ютерні науки», «126 Інформаційні системи та технології».

Виконання лабораторної роботи повинно забезпечити закріплення теоретичних знань і практичних навичок, отриманих при вивченні лекційної частини дисциплін, пов'язаних із проектуванням, розробкою та застосуванням баз даних.

В методичних вказівках розглянуті основні питання, пов'язані з теоретичним обґрунтуванням та безпосереднім виконанням лабораторної роботи.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 SQL і прикладне програмне забезпечення

Мову SQL можна використовувати як в інтерактивному режимі, так і шляхом впровадження його операторів в програми, написані на процедурних мовах високого рівня. Прикладом інтерактивного використання SQL-операторів є вікно Query Analyzer в середовищі MS SQL Server. Застосування ж мови SQL в прикладних програмах на практиці реалізоване двома різними способами:

1. Впроваджені SQL-оператори. Окремі SQL-оператори впроваджуються прямо в початковий текст програми і змішуються з операторами базової мови. Цей підхід дозволяє створювати програми, що звертаються безпосередньо до бази даних. Спеціальні програми-передкомпілятори перетворюють початковий текст з метою заміни SQL-операторів відповідними викликами підпрограм СУБД, потім він компілюється і збирається звичайним способом.

2. Використання прикладного інтерфейсу програмування (Application Programming Interface, API). Альтернативний варіант полягає в наданні програмісту стандартного набору функцій, до яких можна звертатися із створюваних ним програм. Конкретний варіант API може надавати той же набір функціональних можливостей, який існує при підключенні вбудованих операторів, проте при цьому усувається необхідність передкомпіляції початкового тексту. Крім того, деякі розробники вказують, що в цьому випадку використовується зрозуміліший інтерфейс і створений програмний текст зручніший з точки зору його супроводу.

Обидва способи припускають використання операторів як статичного SQL, так і динамічного SQL.

Що стосується операторів статичного SQL, то якого-небудь зміни після їх одноразового написання не передбачається. Вони можуть зберігатися як у файлах, призначених для подальшого використання, так і

у вигляді процедур бази даних, що зберігаються, проте програмісти не отримують усієї тієї гнучкості, яку пропонує їм динамічний SQL. Незважаючи на наявність великого числа запитів, доступних кінцевому користувачеві, може статися так, що жоден з цих "законсервованих" запитів не зможе задовольнити його поточним потребам.

Динамічний SQL дає можливість програмісту або кінцевому користувачу створювати оператори під час виконання додатка і передавати їх бази даних, яка після виконання цих операторів поміщає вихідні дані в змінні програми. Динамічний SQL часто використовується інструментальними засобами, призначеними для побудови заздалегідь незапланованих запитів, що дозволяють оперативно формувати той або інший оператор SQL залежно від особливих вимог, що виникли в конкретній ситуації. Після налаштування оператора SQL відповідно до потреб користувача він прямує серверу баз даних для перевірки на наявність синтаксичних помилок і необхідних для його виконання привілеїв, після чого відбувається його компіляція і виконання.

Розглянемо застосування прикладного інтерфейсу програмування для виконання операторів SQL.

Прикладний API включає набір бібліотечних функцій, що надають програмістові різноманітні типи доступу до бази даних, а саме: підключення, виконання різних SQL-операторів, вибірка окремих рядків даних з результуючих наборів даних і т. д.

Щоб не розробляти окремі версії призначеного для користувача застосування для кожної з цільових СУБД, з якими це застосування планується використовувати, Microsoft розробила стандарт, що отримав назву Open Database Connectivity (ODBC). Технологія ODBC передбачає застосування єдиного інтерфейсу для доступу до різних баз даних SQL, причому мова SQL розглядається як основний стандартний засіб доступу. Цей інтерфейс забезпечує високу міру універсальності, в результаті одно і те ж застосування може діставати доступ до даних, що зберігаються у базах різних цільових СУБД, без необхідності внесення змін до його

програмного тексту. Таким чином, розробники отримали інструмент для створення і поширення додатків архітектури "клієнт-сервер", здатних працювати з широким спектром різних цільових СУБД, а зв'язати додатки з будь-якою вибраною цільовою СУБД можна за допомогою відповідного ODBC -драйвера.

Нині технологія ODBC фактично набула значення галузевого стандарту. Головною причиною її популярності є властива їй гнучкість, що надає розробникам наступні переваги:

- застосунки більше не пов'язані з прикладним API якоїсь однієї СУБД;
- SQL-оператори можуть явно включатися в початковий текст додатка або динамічно створюватися безпосередньо під час виконання програм;
- застосунок здатний ігнорувати особливості використовуваних протоколів передачі даних;
- дані посилаються і доставляються в тому форматі, який найбільш зручний для конкретного застосування;
- засоби підтримки ODBC розроблені з урахуванням вимог стандартів X/Open і CLI (Call Level Interface);
- нині існують драйвери ODBC для різних типів найпоширеніших СУБД.

Джерела даних комп'ютера зберігають відомості про підключення в реєстрі Windows на певному комп'ютері (<https://support.microsoft.com/uk-ua/office/%D0%B0%D0%B4%D0%BC%D1%96%D0%BD%D1%96%D1%81%D1%82%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%B4%D0%B6%D0%B5%D1%80%D0%B5%D0%BB-%D0%B4%D0%B0%D0%BD%D0%B8%D1%85-odbc-b19f856b-5b9b-48c9-8b93-07484bfab5a7>). Джерела даних комп'ютера можна використовувати лише на тому комп'ютері, на якому їх визначено. Є два типи джерел даних комп'ютера: користувацькі та системні. Користувацькі джерела даних може використовувати лише поточний користувач, і вони видимі лише для

цього користувача. Системні джерела даних можуть використовувати всі користувачі на комп'ютері, і вони видимі для всіх користувачів на комп'ютері та в системних службах. Джерело даних комп'ютера особливо корисне, коли потрібно посилити безпеку, оскільки його можуть переглядати лише користувачі, які ввійшли в систему, і віддалений користувач не зможе скопіювати це джерело даних на інший комп'ютер.

Файлові джерела даних (так звані файли DSN) зберігають відомості про підключення в текстовому файлі, а не в реєстрі Windows, і зазвичай гнучкіші у використанні ніж джерела даних комп'ютера. Наприклад, ви можете скопіювати файлове джерело даних на будь-який комп'ютер із правильним драйвером ODBC, щоб програма посилалася на узгоджені та точні відомості про підключення на всіх комп'ютерах, які вона використовує. Або можна розмістити файлове джерело даних на одному сервері, надати до нього спільний доступ на багатьох комп'ютерах у мережі та легко зберігати відомості про підключення в одному розташуванні.

Крім того, для файлового джерела даних можна заборонити спільний доступ. Файлове джерело даних із забороненим спільним доступом розташовується на одному комп'ютері та вказує на джерело даних комп'ютера. Ви можете використовувати файлові джерела даних із забороненим спільним доступом, щоб отримати доступ до наявних джерел даних комп'ютера із файлових джерел даних.

1.2 Технологія ODBC та її особливості

У інтерфейс ODBC включені наступні елементи:

- бібліотека функцій, виклик яких дозволяє застосунку підключатися до бази даних, виконувати SQL-оператори і витягати інформацію з результуючих наборів даних ;
- стандартний метод підключення і реєстрації в СУБД;
- стандартне представлення для цих різних типів;
- стандартний набір кодів помилок;

- типовий синтаксис SQL-операторів, побудований на використанні специфікації X/Open і ISO CGI.

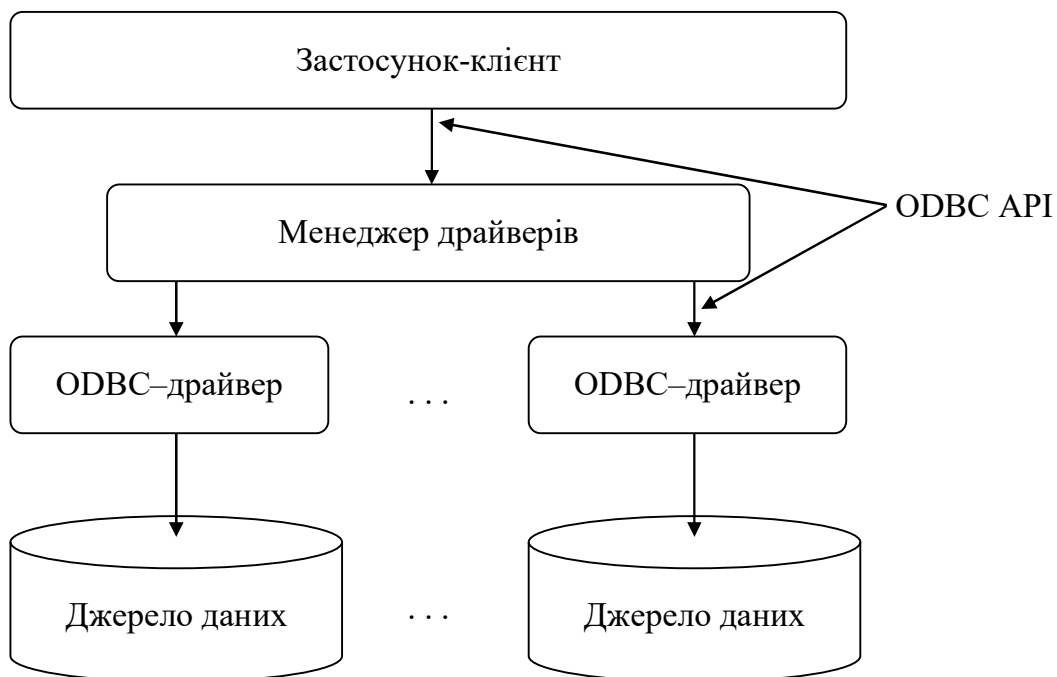


Рисунок 1.1

Загальна архітектура ODBC, яку наведено на рисунку 1.1, включає чотири елементи:

1. Застосунок. Цей компонент виконує обробку даних і виклик функцій бібліотеки ODBC для відправки SQL-операторів СУБД і вибірки поверненої СУБД інформації.

2. Менеджер драйверів. Він виконує завантаження драйверів на вимогу додатка.

3. Драйвери і агенти баз даних. Ці компоненти обробляють виклики функцій ODBC і направляють SQL-запити до конкретних джерел даних, а також повертають отримані результати застосунку. При необхідності драйвери виконують модифікацію початкового запиту застосунку з метою приведення його у відповідність синтаксичним вимогам цільової СУБД.

4. Джерела даних. Містять ті дані, доступ до яких потрібний користувачеві застосунка. Дані зберігаються у БД, контрольованій цільовою СУБД, операційною системою, а також мережевою операційною системою, якщо така використовується.

Виходячи з вищесказаного, можна відмітити, що технологія ODBC пропонує єдиний інтерфейс доступу до різноманітних баз даних SQL. Мова SQL використовується в ній як основний стандарт доступу до даних. Інтерфейс ODBC (вбудований в мову C та інші) забезпечує високу міру універсальності: одно застосування може звертатися до різних SQL-сумісних СУБД за допомогою загального коду. Це дозволяє розробникові створювати і поширювати застосунки "клієнт/сервер" без урахування особливостей конкретної СУБД, в результаті одно і те ж застосування дістає можливість доступу до баз цих різних СУБД, що підтримують мову SQL. Подібні функціональні можливості технології ODBC дозволяють розробляти застосунки для роботи із СУБД різного типу. Для зв'язку застосунків з різнотипними СУБД використовуються відповідні ODBC-драйвери.

Технологія ODBC передбачає створення додаткового рівня між застосунком і використовуваною СУБД. Служби ODBC забезпечують отримання від застосунка запитів на вибірку інформації і переклад їх на мову ядра бази даних, що адресується, для доступу до інформації, що зберігається в ній.

Основне призначення ODBC полягає в абстрагуванні застосунка від особливостей ядра серверної бази даних, з якою воно здійснює взаємодію, тому серверна база даних стає як би прозорою для будь-якого клієнтського застосування.

Взаємодія застосунка з даними робиться за допомогою менеджера (диспетчера) драйверів, він підключає необхідний драйвер відповідно до формату цих СУБД. Драйвер СУБД, використовуючи мережеві засоби, як правило, комунікаційні модулі конкретної СУБД, передає SQL-оператори серверу СУБД. Результати виконання запитів на сервері пересилаються

назад до застосунку. Один із можливих варіантів підключення до бази даних клієнтського застосунку із використанням технології ODBC, наведений на рисунку 1.2 (джерело: <https://www.sqlshack.com/how-to-configure-a-linked-server-using-the-odbc-driver/>).

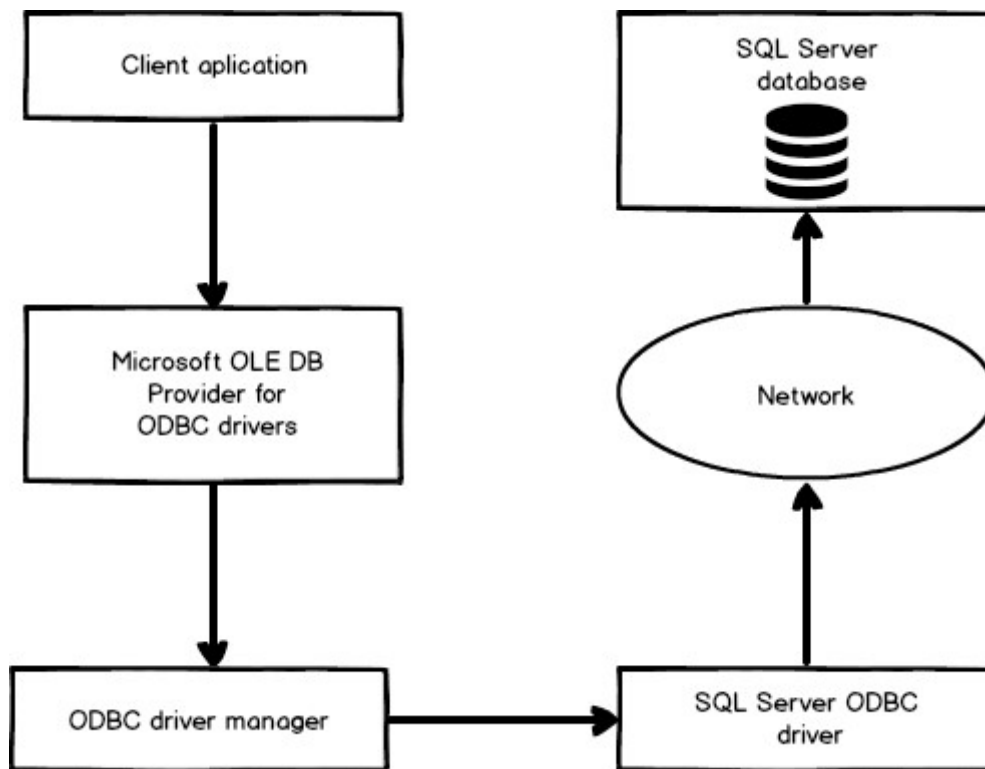


Рисунок 1.2

Перевага технології ODBC полягає в простоті розробки застосунків, що обумовлено високим рівнем абстрактності інтерфейсу доступу до даних практично будь-яких існуючих типів СУБД. При цьому можливе створення джерела даних, пов'язаного з будь-яким типом бази даних. За допомогою цієї технології можна створювати клієнт-серверні застосунки, причому засобами персональних СУБД доцільно розробляти клієнтську частину застосунка, а засобами SQL Server – серверну.

Основний недолік технології ODBC пов'язаний з необхідністю трансляції запитів, що знижує швидкість доступу до даних. У системах клієнт-сервер він усувається шляхом переміщення обробки запиту з комп'ютера-клієнта на комп'ютер-сервер.

При використанні в клієнтському застосунку засобів ODBC здійснюється звернення до певного джерела даних, а через нього – до СУБД, що надає це джерело. Крім того, встановлюється загальна підсистема ODBC і визначаються пари "драйвер - база даних", яким задаються імена, вживані при установці з'єднання з базою даних. Відповідні пари називаються іменами джерел даних, або поймаєнованими джерелами даних (Data Source Names, DSN).

Створення джерела даних виконується за допомогою утиліти ODBC Data Source Administrator, що викликається з вікна панелі управління. До складу параметрів джерела даних входять: його ім'я і опис; сервер, з яким встановлюється з'єднання ; метод аутентифікації; ім'я бази даних.

Ім'я DSN дозволяє звернутися до джерела даних ODBC з прикладного застосунку.

Використання ODBC значно спрощується за допомогою Microsoft Foundation Classes Library (Бібліотека основних класів Microsoft). Прості застосування, що отримують доступ к таблицям через ODBC, можуть бути створені всього лише декількома натисненнями кнопки миші з використанням майстрів AppWizard та ClassWizard. Існує декілька класів MFC, які підтримують доступ до баз даних і наборам записів.

Докладні відомості про інтерфейс ODBC можна також подивитися, наприклад, в розділі MSDN «Довідкові матеріали для програміста ODBC» за посиланням:

<https://learn.microsoft.com/en-us/sql/odbc/reference/odbc-programmer-s-reference?view=sql-server-ver16>

2 ВИКОНАННЯ РОБОТИ

При виконанні лабораторної роботи передбачається використання утиліти ODBC Data Source Administrator та СУБД MS Access. У зв'язку із цим елементи інтерфейсу можуть мати відмінності порівняно із тими, які зазначено далі (особливо у випадку використання версії програмного забезпечення, локалізованої для використання певної національної мови). Ці відмінності не є принциповими та не впливають на виконання роботи та отримані результати.

Для початку виконання роботи треба підключити базу даних, яку було створено при виконанні лабораторної роботи за темою «Ознайомлення з основними особливостями СУБД Microsoft SQL Server. Створення бази даних та об'єктів бази даних».

2.1 Створення джерела даних ODBC

Для доступу до джерел даних ODBC необхідно виконати таку послідовність дій.

1. Відкрити Панель управління (Control Panel) Windows.
2. Відкрити теку Адміністрування (Administrative Tools).
3. Вибрати пункт Джерела даних (ODBC) (Data Sources (ODBC)).
4. Відкрити список ODBC-джерел подвійним клацанням миші.

Для створення нового джерела даних ODBC необхідно виконати таку послідовність дій.

1. Вибрати вкладку Користувацький DSN (User DSN) і натиснути кнопку Додати (Add).
2. Вибрати драйвер SQL Server і натиснути кнопку Готово (Finish).
3. Ввести ім'я джерела ODBC. Нехай це буде delivery і вибрати сервер, до якого потрібно підключитися. При виборі сервера потрібно вибрати сервер, який відповідає комп'ютеру, на якому виконується робота.

Це ім'я також можна побачити при підключенні до Microsoft SQL Server із використанням застосунку SQL Server Management Studio.

4. Натиснути кнопку Далі (Next).

5. У вікні автентифікації користувача залишити дані без змін і натиснути кнопку Далі (Next).

6. У наступному вікні вибрати зі списку баз даних базу даних delivery для використання за замовчуванням.

7. У наступному вікні можна усе залишити без змін і натиснути кнопку Готово (Finish).

8. Перевірити підключення до сервера та у разі успішного підключення натиснути кнопку ОК. У цьому випадку джерело ODBC буде збережено і з'явиться в списку джерел ODBC.

2.2 Використання СУБД Microsoft Access як клієнтського застосунку

У даному випадку СУБД MS Access буде використано не як СУБД, а як програмний засіб, за допомогою якого буде створено дуже простий клієнтський застосунок для роботи із базою даних Microsoft SQL Server. Для цього необхідно виконати таку послідовність дій.

1. Запустити СУБД Microsoft Access.

2. Створити нову базу даних з ім'ям client_mssql (бажано в тій же папці, де знаходиться база даних, створена під час виконання лабораторних робіт).

3. У підменю пункту File вибрати пункт Зовнішні дані (Get External Data) та підпункт Зв'язок із таблицями (Link Tables...) (рисунок 2.1).

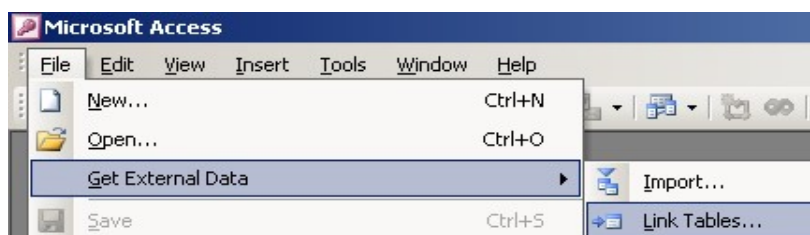


Рисунок 2.1

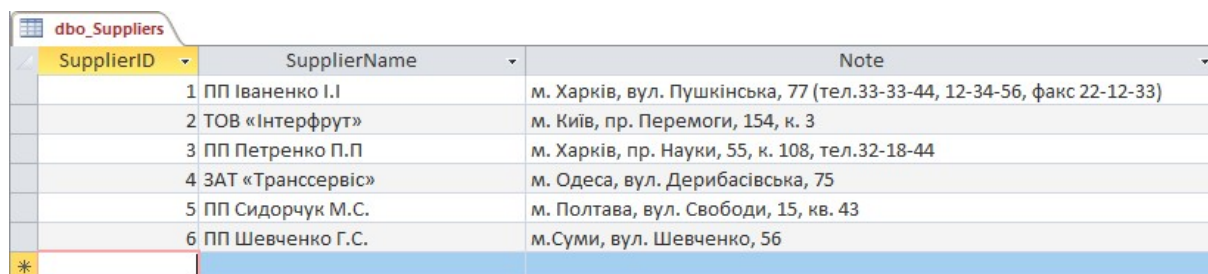
4. У вікні Зв'язок (Link), у полі зі списком Тип файлів (Files of type) вибрати пункт Бази даних ODBC (ODBC Databases).

5. У вікні Вибір джерела даних (Select Data Source) відкрити вкладку Джерело даних комп'ютера (Machine Data Source), вибрати джерело ODBC з ім'ям delivery.

6. У вікні Зв'язок із таблицями (Link Tables) вибрати таблицю dbo.Suppliers. В результаті буде створено таблицю dbo_Suppliers.

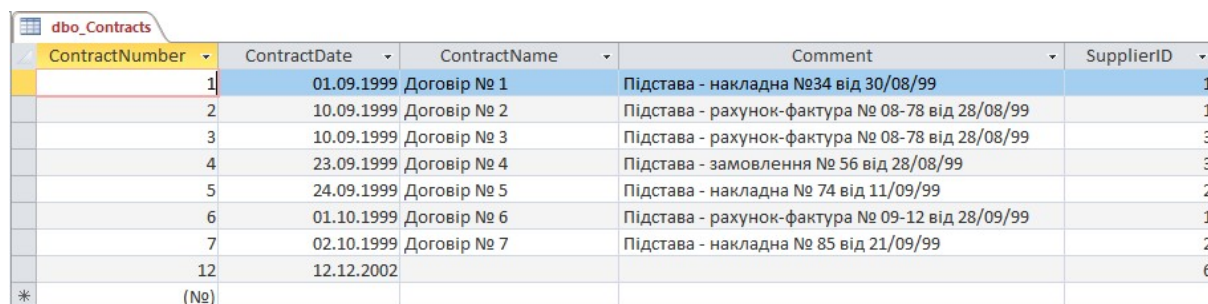
7. Аналогічно створити таблицю dbo_Contracts, яка буде пов'язана з таблицею Contracts.

8. Перевірити можливість роботи з базою даних, використовуючи як інтерфейс користувача СУБД Microsoft Access. Для цього відкрити створені таблиці dbo_Suppliers (рисунок 2.2) та dbo_Contracts (рисунок 2.3). Використовуючи інтерфейс Microsoft Access, виконати перевірки операцій маніпулювання даними (додавання, видалення, зміна даних). Перевірити збереження результатів маніпулювання за допомогою SQL Server Management Studio.



SupplierID	SupplierName	Note
1	ПП Іваненко І.І	м. Харків, вул. Пушкінська, 77 (тел.33-33-44, 12-34-56, факс 22-12-33)
2	ТОВ «Інтерфрут»	м. Київ, пр. Перемоги, 154, к. 3
3	ПП Петренко П.П	м. Харків, пр. Науки, 55, к. 108, тел.32-18-44
4	ЗАТ «Транссервіс»	м. Одеса, вул. Дерибасівська, 75
5	ПП Сидорчук М.С.	м. Полтава, вул. Свободи, 15, кв. 43
6	ПП Шевченко Г.С.	м.Суми, вул. Шевченко, 56

Рисунок 2.2



ContractNumber	ContractDate	ContractName	Comment	SupplierID
1	01.09.1999	Договір № 1	Підстава - накладна №34 від 30/08/99	1
2	10.09.1999	Договір № 2	Підстава - рахунок-фактура № 08-78 від 28/08/99	1
3	10.09.1999	Договір № 3	Підстава - рахунок-фактура № 08-78 від 28/08/99	3
4	23.09.1999	Договір № 4	Підстава - замовлення № 56 від 28/08/99	3
5	24.09.1999	Договір № 5	Підстава - накладна № 74 від 11/09/99	2
6	01.10.1999	Договір № 6	Підстава - рахунок-фактура № 09-12 від 28/09/99	1
7	02.10.1999	Договір № 7	Підстава - накладна № 85 від 21/09/99	2
12	12.12.2002			6

Рисунок 2.3

9. Розробити засобами Microsoft Access екранну форму, яка дозволить для кожного постачальника бачити список укладених із ним договорів. Можливий варіант реалізації такої форми наведено на рисунку 2.4. Як видно з рисунка, форма складається з головної та підлеглої форм, дані у яких зв'язані між собою.

ContractNumber	ContractDate	SupplierID	ContractName	Comment
1	01.09.1999	1	Договір № 1	Підстава - накладна №34 ві
2	10.09.1999	1	Договір № 2	Підстава - рахунок-фактура
6	01.10.1999	1	Договір № 6	Підстава - рахунок-фактура
*	(№)	1		

Рисунок 2.4

10. Після створення форми перевірити можливість роботи з базою даних, використовуючи як інтерфейс користувача створену форму.

11. Створити запит, за допомогою якого можна переглядати список договорів та основні відомості про постачальника, зокрема, назву. Текст такого запиту наведений на рисунку 2.5. Перевірити правильність роботи цього запиту та зберегти його із довільною назвою, наприклад, Запит1.

```
SELECT dbo_Contracts.ContractNumber, dbo_Contracts.ContractDate, dbo_Contracts.ContractName, dbo_Suppliers.SupplierName
FROM dbo_Contracts INNER JOIN dbo_Suppliers ON dbo_Contracts.SupplierID = dbo_Suppliers.SupplierID;
```

Рисунок 2.5

12. Припустимо, що список договорів, який формується за допомогою зазначеного вище запиту, треба мати можливість роздрукувати у певному вигляді. Для цього треба створити відповідний звіт та призначити для цього звіту джерелом даних Запит1. У режимі попереднього перегляду такий звіт може мати вигляд, наведений на рисунку 2.6. Перевірити правильність роботи цього звіту та зберегти його із довільною назвою, наприклад, Report1.



ContractNumber	ContractDate	ContractName	SupplierName
1	01.09.1999	Договір № 1	ПП Іваненко І.І
2	10.09.1999	Договір № 2	ПП Іваненко І.І
3	10.09.1999	Договір № 3	ПП Петренко П.П
4	23.09.1999	Договір № 4	ПП Петренко П.П
5	24.09.1999	Договір № 5	ТОВ «Інтерфрут»
6	01.10.1999	Договір № 6	ПП Іваненко І.І
7	02.10.1999	Договір № 7	ТОВ «Інтерфрут»

Рисунок 2.6

13. Підключити інші таблиці бази даних Microsoft SQL Server.

14. Використовуючи раніше отримані навички роботи з СУБД Microsoft Access створити інші форми, що дозволяють працювати з базою даних, а також засоби обробки даних (запити, звіти), що дозволяють обробляти інформацію, що зберігається в базі даних Microsoft SQL Server засобами Microsoft Access, виводити її на друк тощо.

Наведений вище приклад створення джерела даних ODBC та використання його у клієнтському застосунку є досить простим. Більш детальну інформацію щодо побудови та використання джерел даних ODBC при роботі з базами даних Microsoft SQL Server, можна, наприклад, отримати за посиланнями:

<https://learn.microsoft.com/en-us/sql/odbc/admin/odbc-data-source-administrator?view=sql-server-ver16>

<https://learn.microsoft.com/en-us/sql/odbc/admin/managing-data-sources?view=sql-server-ver16>

<https://learn.microsoft.com/en-us/sql/odbc/odbc-glossary?view=sql-server-ver16>

<https://docs.devart.com/odbc/sqlserver/access.htm>

2.3 Збереження результатів роботи

Для збереження результатів виконання лабораторної роботи треба зберегти файл client_mssql.accdb або файл client_mssql.mdb залежно від того, яку версію СУБД Microsoft Access було використано.

3 ВИМОГИ ДО ЗВІТУ

Відповідним чином оформлений та роздрукований звіт з лабораторної роботи є документом, що підтверджує виконання студентом лабораторної роботи.

У звіті з лабораторної роботи:

- 1) стисло описати основні етапи виконання завдання;
- 2) описати створений клієнтський застосунок та особливості роботи з ним;
- 3) зробити висновки за результатами виконання лабораторної роботи.

Звіт з лабораторної роботи роздруковується на аркушах формату А4, він повинен мати відповідній титульний аркуш. Роздрукованій звіт здається студентом викладачу у файлі.

Звіт має бути оформлень за такими вимогами:

- параметри сторінки: лівий відступ – 3 см; правий – 1,5 см; верхній та нижній відступи по 2 см;
- шрифт Times New Roman, 14;
- налаштування абзацу: вирівнювання – за шириною, відступи зліва та справа – 0 см, відступ першого рядка - 1,25 см, інтервал перед та після абзацу – 0 пт, міжрядковий інтервал – одинарний; на вкладці «Положення на сторінці» відключити функцію «Заборона висячих рядків».

Усі скріншоти розміщені у звіті, є рисунками, отже повинні мати підписи та відповідну нумерацію.

Ознакою того, що студент не тільки виконав лабораторну роботу, але й здав її (тобто підтвердив наявність відповідних знань та навичок), є наявність на титульному аркуші підпису викладача та дати здачі.

4 ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Якими способами реалізоване на практиці застосування мови SQL в прикладних програмах?
2. Що таке впроваджені SQL-оператори?
3. Що таке API?
4. Що таке ODBC?
5. Які переваги надає використання технології ODBC?
6. Які недоліки є характерними для технології ODBC?
7. Які складові виділяють у інтерфейсі ODBC?
8. Які складові включено до загальної архітектури ODBC?
9. Загальна архітектура ODBC. Застосунок як архітектурна складова, його призначення.
10. Загальна архітектура ODBC. Менеджер драйверів як архітектурна складова, його призначення.
11. Загальна архітектура ODBC. Драйвери і агенти баз даних як архітектурні складові, їх призначення.
12. Загальна архітектура ODBC. Джерела даних як архітектурні складові, їх призначення.
13. Що таке DSN?
14. Які види DSN реалізовано у Windows?
15. Яким чином запустити утиліту ODBC Data Source Administrator?
16. Яким чином виконується створення джерела даних за допомогою утиліти ODBC Data Source Administrator?
17. Яким чином можна забезпечити автентифікацію користувача при створенні джерела даних ODBC?
18. Як забезпечити підключення із середовища СУБД Microsoft Access до бази даних Microsoft SQL Server?
19. Які особливості роботи із зовнішніми даними у середовищі СУБД Microsoft Access при використанні засобу зв'язку Link Tables?

20. Який засіб зв'язку із зовнішніми даними у середовищі СУБД Microsoft Access є альтернативним засобу зв'язку Link Tables?

21. Яким чином можна реалізувати роботу із зовнішніми даними у середовищі СУБД Microsoft Access для кінцевого користувача?

22. Які недоліки є характерними при використанні СУБД Microsoft Access як засобу розробки клієнтського застосунку?

СПИСОК ЛІТЕРАТУРИ

1. Програмування баз даних. Практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 152 «Метрологія та інформаційно-виміррювальна техніка» / М. В. Добролюбова, М. В. Філіппова, О. М. Маркіна. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 164 с.
2. Anthony DeBarros. Practical SQL. 2nd Edition. – No Starch Press, Inc., 2022. – 440 с.
3. Jeffrey A. Hoffer, V. Ramesh, Heikki Topi. Modern Database Management. Thirteenth Edition. – Pearson Education Limited, 2020. – 591 p.
4. Dušan Petković. Microsoft SQL Server 2019. A Beginner's Guide. Seventh Edition. – McGraw-Hill Education, 2020. – 865 p.
5. Mukesh Negi. Fundamentals of Database Management System: Learn essential concepts of database systems. – BPB Publications, 2019. – 175 p.
6. Edward Sciore. Database Design and Implementation: Second Edition. – Springer Nature, 2020. – 468 p.
7. Gavin Powell. Database Modeling Step by Step. – CRC Press, 2020. – 268 p.
8. Sanjiv Purba. Handbook of Data Management: 1999 Edition. – CRC Press, 2019. – 1101 p.
9. C. J. Date. Database Design and Relational Theory: Normal Forms and All That Jazz. – Apress, 2019. – 451 p.
10. Jonathan Eckstein, Bonnie R. Schultz. Introductory Relational Database Design for Business, with Microsoft Access. – John Wiley & Sons, 2018. – 328 p.
11. Alan Beaulieu. Learning SQL: Generate, Manipulate, and Retrieve Data. – O'Reilly Media, Inc., 2020. – 384 p.
12. M. Tamer Özsu, Patrick Valduriez. Principles of Distributed Database Systems. – Springer Nature, 2019. – 674 p.

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи за темою «Реалізація простого клієнтського застосунку для роботи з базою даних Microsoft SQL Server»
для студентів спеціальностей

121 «Інженерія програмного забезпечення»

122 «Комп'ютерні науки»

126 «Інформаційні системи та технології»

Укладачі:

ОРЛОВСЬКИЙ Дмитро Леонідович

КОПП Андрій Михайлович

Відповідальний за випуск Годлевський М.Д.
Роботу до видання рекомендував Гамаюн І.П.

План 2023 р., поз. 498

Підп. до друку __.__.__. Гарнітура Times New Roman.

Ум. друк. арк. 1,0.

Видавничий центр НТУ «ХП»,
вул. Кирпичова, 2, м. Харків, 61002
Свідоцтво про державну реєстрацію ДК № 3478 від 21.08.2017 р.

Самостійне електронне видання