

## Лабораторна робота 1

### Ознайомлення з основними особливостями СУБД Microsoft SQL Server 2008 і можливостями інтегрованого середовища Microsoft SQL Server Management Studio. Створення бази даних і об'єктів бази даних

#### Опис завдання, що розглядається в лабораторному практикумі

Деяка фірма купує товари у різних постачальників (як юридичних, так і фізичних осіб). Придбання товарів здійснюється партіями і оформляється у вигляді договорів на поставку. Кожен договір на поставку товару має унікальний номер і може бути укладений тільки з одним постачальником. У документах за кожним договором для кожного товару зазначаються: найменування, розмір поставленої партії і ціна (в грн.).

Для зберігання і обробки подібної інформації засобами СУБД Microsoft SQL Server 2008 необхідно створити базу даних.

## ВИКОНАННЯ РОБОТИ

### I. Створення бази даних

#### I.1 Створення об'єктів бази даних

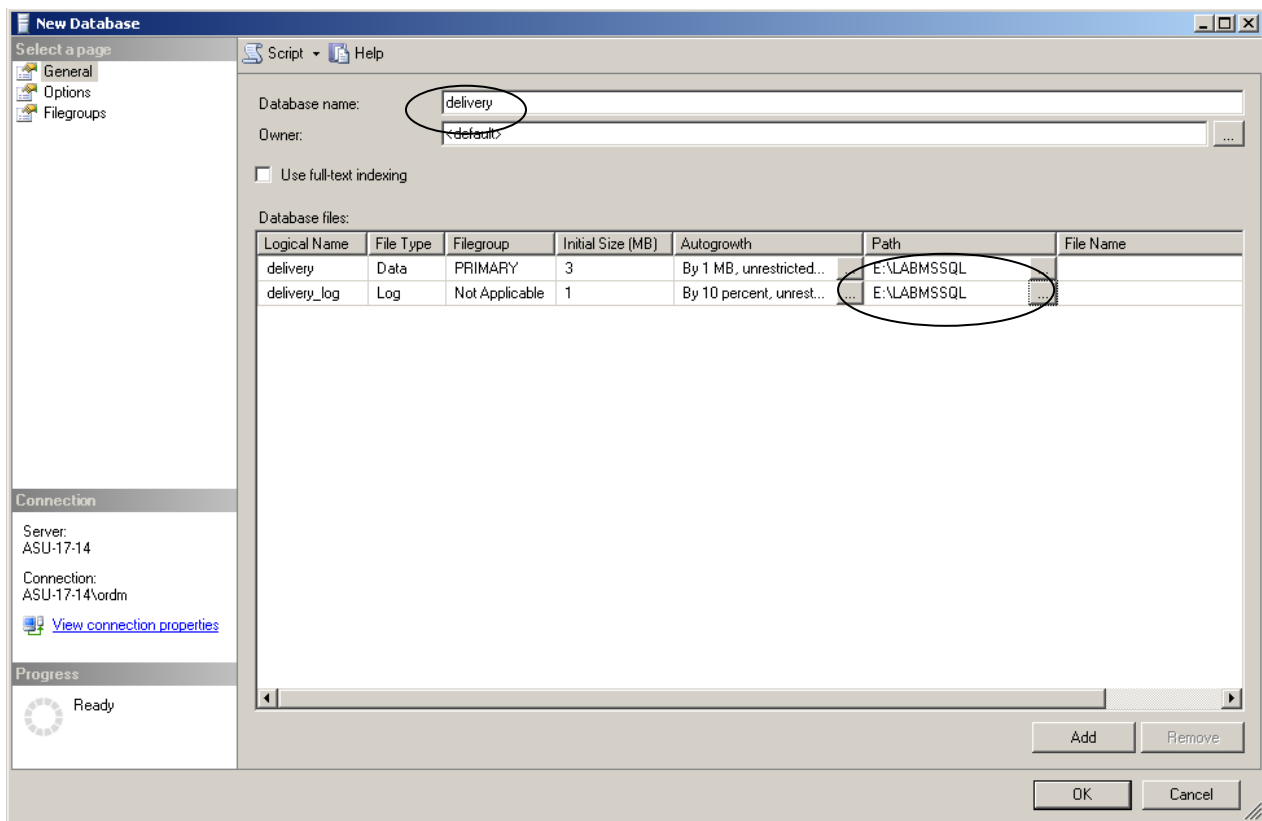
1. Створити на диску комп'ютера (D :, E: і т.п.) каталог з довільним ім'ям (наприклад, E: \ LABMSSQL)
2. Запустити Microsoft SQL Server Management Studio, для чого:
  - в панелі завдань вибрати пункт Microsoft SQL Server 2008;
  - вибрати підпункт SQL Server Management Studio;
  - у вікні підключення (малюнок 1.1) натиснути кнопку Connect;



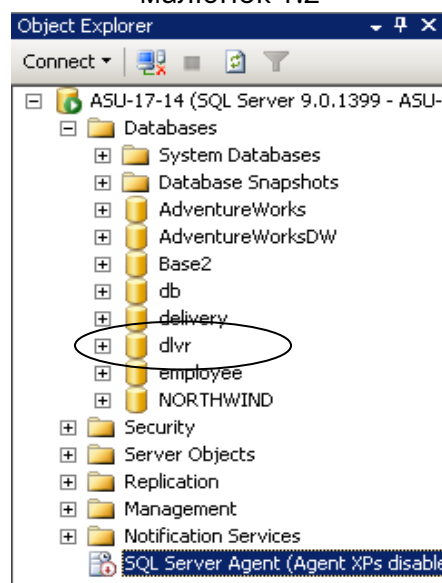
малюнок 1.1

\* (Примітка: ім'я сервера питати у адміністратора залу)

3. Після появи на екрані середовища Microsoft SQL Server Management Studio у вікні Object Explorer вибрати пункт Databases, натиснути праву кнопку миші і в меню вибрати пункт New Database .... В результаті на екрані з'явиться вікно, що дозволяє ввести основні параметри нової бази даних. Необхідно ввести ім'я нової бази даних - delivery і визначити місце розміщення файлів - E: \ LABMSSQL (рисунок 1.2). Після введення даних натиснути кнопку ОК. Нова база даних з'явиться в списку баз даних у вікні Object Explorer (рисунок 1.3).

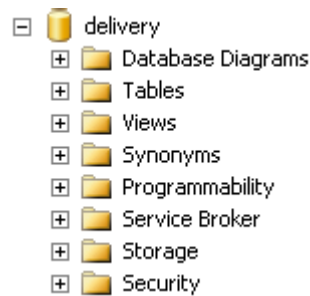


малюнок 1.2



малюнок 1.3

4. Вибрати створену базу даних і розкрити список її об'єктів (рисунок 1.4).

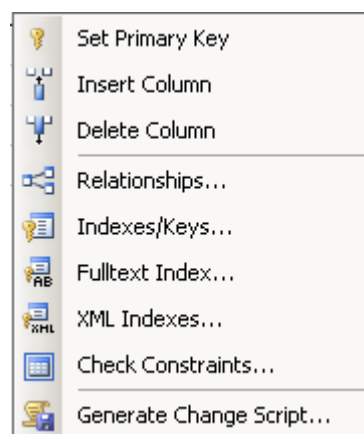


малюнок 1.4

5. У списку об'єктів бази даних натиснути правою кнопкою миші по пункту Tables і в меню вибрати пункт New Table .... Ввести поля нової таблиці (рисунок 1.5), визначивши при цьому типи даних і ключове поле (для цього потрібно клацнути по полю правою кнопкою миші і вибрати в меню відповідний пункт (рисунок 1.6)).

	Column Name	Data Type	Allow Nulls
	КодПоставщика	int	<input type="checkbox"/>
	Адрес	text	<input type="checkbox"/>
	Примечание	text	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

малюнок 1.5



малюнок 1.6

6. Закрити вкладку зі структурою нової таблиці. Зберегти нову таблицю з ім'ям «Постачальники» (без лапок).
7. Аналогічно створити таблиці «ФізическієЛіца» і «ЮрідическієЛіца». Їх структури приведені на малюнках 1.7 і 1.8 відповідно.

	Column Name	Data Type	Allow Nulls
	КодПоставщика	int	<input type="checkbox"/>
	Фамилия	char(20)	<input type="checkbox"/>
	Имя	char(20)	<input type="checkbox"/>
	Отчество	char(20)	<input type="checkbox"/>
	НомерСвидетельства	char(10)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

малюнок 1.7

	Column Name	Data Type	Allow Nulls
►?	КодПоставщика	int	<input type="checkbox"/>
	Название	char(20)	<input type="checkbox"/>
	НалоговыйНомер	char(20)	<input checked="" type="checkbox"/>
	НомерСвидетельстваНДС	char(10)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

малюнок 1.8

8. Створити таблицю «Договори». Структура таблиці приведена на малюнку 1.9. Особливістю цієї таблиці є те, що для поля «НомерДоговора» має бути встановлено властивість автоприращення (autoincrement) з початковим значенням 1 і кроком зміни 1. Для цього в Microsoft SQL Server використовується властивість Identity. Необхідно змінити значення властивості так, як показано на малюнку 1.10.

	Column Name	Data Type	Allow Nulls
►?	НомерДоговора	int	<input type="checkbox"/>
	ДатаДоговора	datetime	<input checked="" type="checkbox"/>
	КодПоставщика	int	<input type="checkbox"/>
	Комментарий	text	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

малюнок 1.9

Column Properties	
<div> <div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> </div> </div>	
Has Non-SQL Server Subscriber	No
<input checked="" type="checkbox"/> Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Merge-published	No

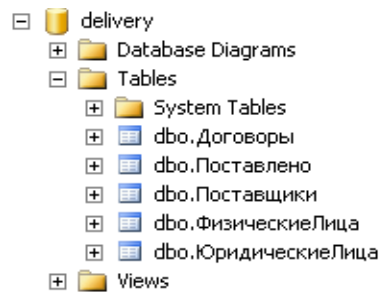
малюнок 1.10

9. Створити таблицю «Поставлено». Структура таблиці приведена на малюнку 1.11. Особливістю таблиці є складовою первинний ключ. Для його створення потрібно виділити ключові поля (мишею, при притиснутою клавіші Shift) і потім визначити їх як ключові

	Column Name	Data Type	Allow Nulls
►?	НомерДоговора	int	<input type="checkbox"/>
?	Товар	char(20)	<input type="checkbox"/>
	Количество	decimal(5, 0)	<input type="checkbox"/>
	Цена	decimal(8, 2)	<input type="checkbox"/>
			<input type="checkbox"/>

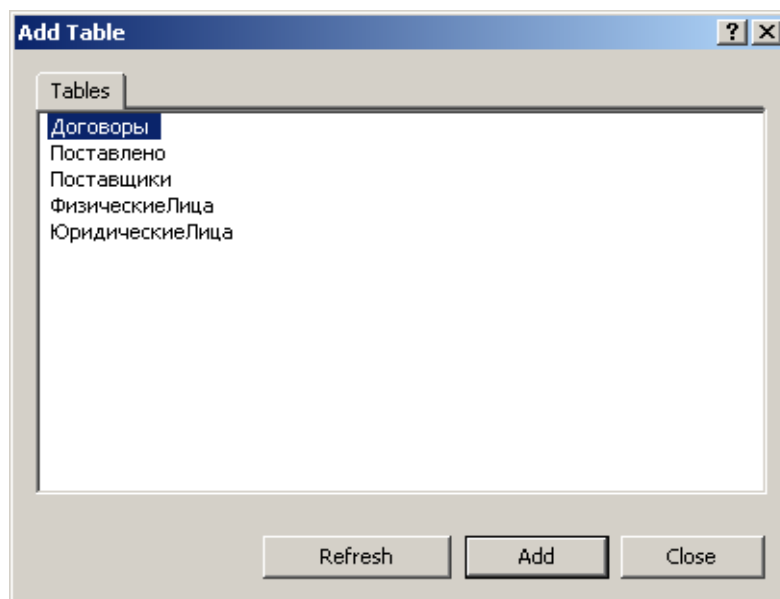
малюнок 1.11

10. В результаті створення таблиць структура створеної бази даних буде мати вигляд (рисунок 1.12). У тому випадку, якщо список таблиць не відображається, можна натиснути правою кнопкою миші на ім'я бази даних і в меню вибрати пункт Refresh.



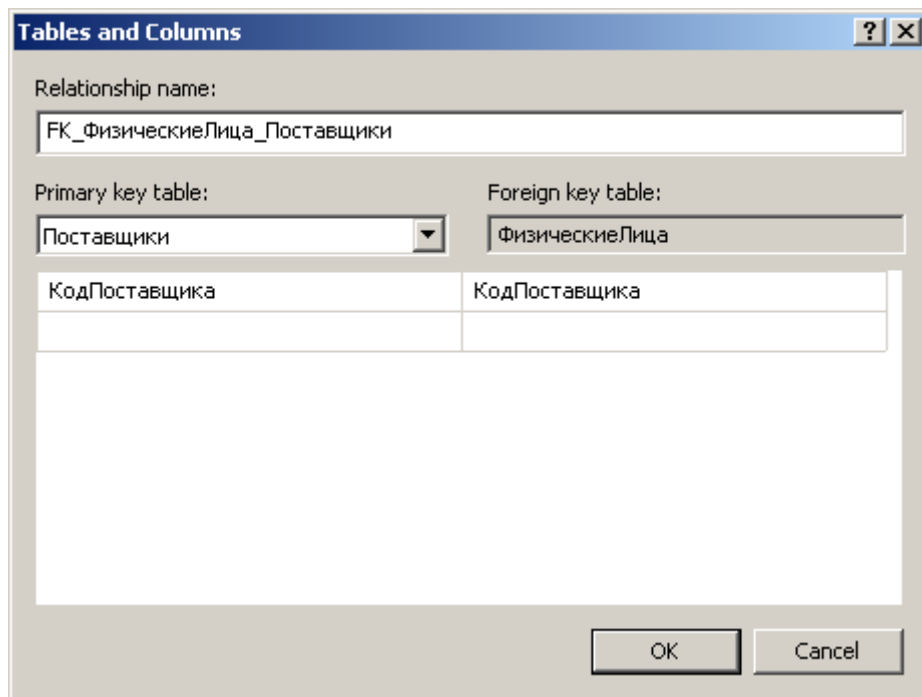
малюнок 1.12

11. Тепер між створеними таблицями потрібно налагодити зв'язки. Це, зокрема, можна зробити, використовуючи візуальні засоби. Для цього необхідно створити діаграму бази даних. Для створення діаграми потрібно натиснути правою кнопкою миші по пункту Database Diagrams (рисунок 1.12) і в меню вибрати пункт New Database Diagram. Потім потрібно послідовно додати до складу діаграми таблиці, вибираючи їх зі списку і натискаючи кнопку Add (рисунок 1.13)

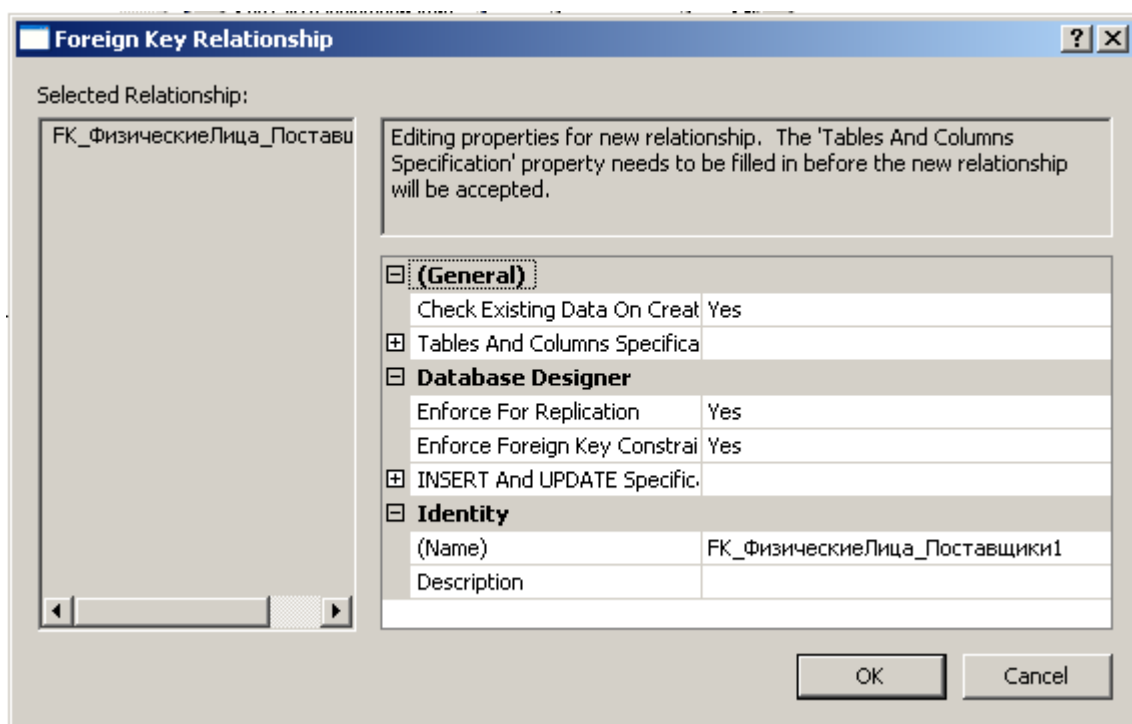


малюнок 1.13

12. Після включення таблиць до складу діаграми потрібно зв'язати їх ключові поля. Для цього потрібно вибрати за допомогою миші ключове поле в батьківській таблиці і, не відпускаючи кнопку миші, тягнути курсор миші до дочірньої таблиці. В результаті встановлення зв'язку на екран буде виведено вікно, яке відображає ім'я зв'язку і зв'язуються поля (малюнок 1.14). Цей приклад відображає встановлення зв'язку між таблицями «Постачальники» і «ЮридическиеЛица». Підтвердивши параметри зв'язку, користувач потім може підтвердити або змінити параметри зовнішнього ключа і тип відносин посилювальної цілісності (рисунок 1.15).

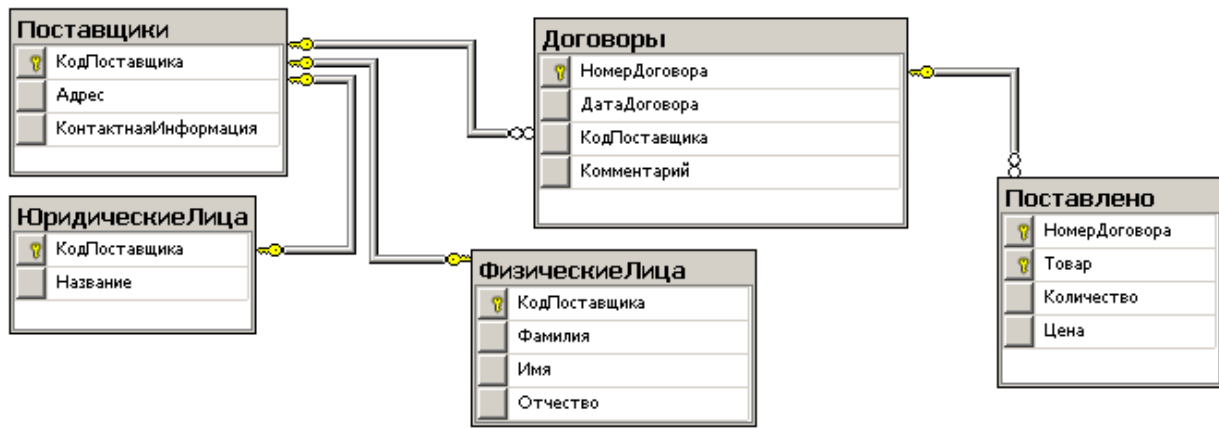


малюнок 1.14



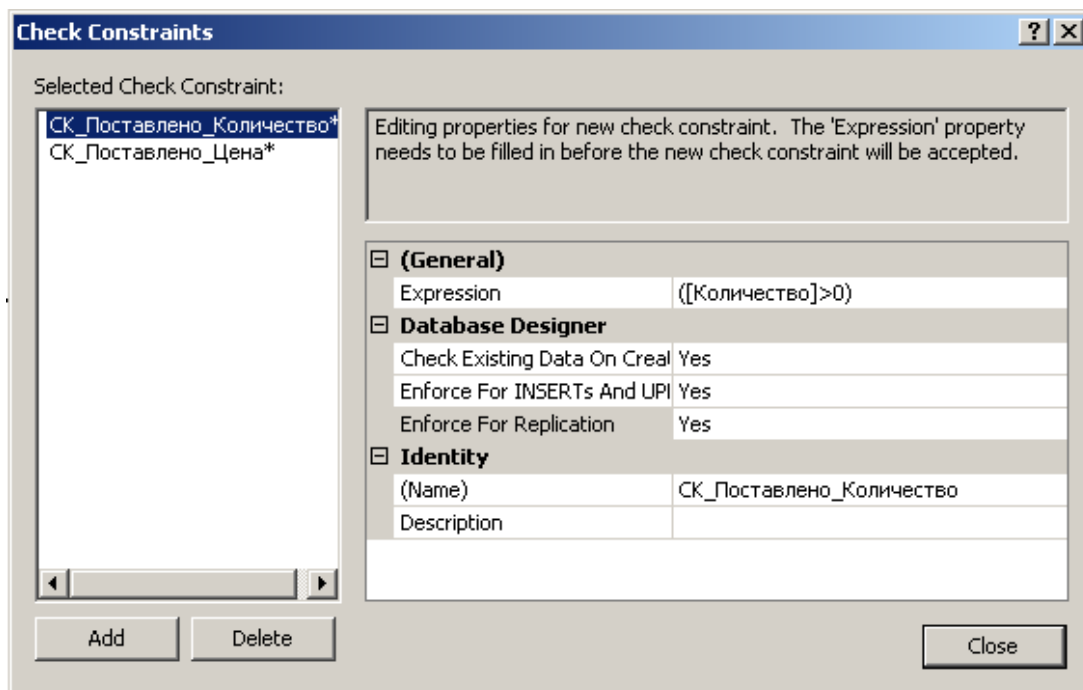
малюнок 1.15

13. В результаті встановлення зв'язків між таблицями діаграма може мати вигляд (рисунок 1.15). Сформовану діаграму можна закрити і зберегти при цьому з довільним ім'ям, наприклад Diagram\_0. Ця діаграма з'явиться в загальному списку діаграм бази даних.



малюнок 1.16

14.С допомогою діаграми бази даних можна змінювати структуру таблиць, встановлювати зв'язки, додаткові властивості полів і т.д. Припустимо, що для полі «Кількість» і «Ціна» таблиці «Поставлено» необхідно реалізувати вимоги, що складаються в тому, що дані, що зберігаються в цих полях, повинні бути позитивними. Для цього знову відкриємо діаграму, клацнемо правою кнопкою миші по таблиці «Поставлено» і в меню виберемо пункт Check Constraints ... У вікні потрібно натиснути кнопку Add і ввести вираз для контролю і назва (рисунок 1.17).



малюнок 1.17

- 15.Аналогічно можна сформулювати контрольний вираз для поля «Ціна». В цьому випадку вираз (Expression) матиме вигляд: ([Ціна]> 0), а ім'я (Name): СК\_Поставлено\_Цена. Після внесення цих змін діаграму можна закрити і зберегти.
- 16.Після закриття діаграми необхідно проаналізувати структурні зміни, зроблені в таблицях (поява нових ключів і т.д.). Для цього слід проаналізувати об'єкти кожної таблиць, послідовно відкриваючи таблиць в списку таблиць.

## I.2 Створення об'єктів бази даних на основі мовою SQL

### Використання коштів DDL для роботи з об'єктами бази даних

Дії, розглянуті вище, можуть бути виконані не тільки в інтерактивному режимі, але і на основі використання мовних засобів DDL і DML мови SQL. Для цього необхідно створити нову базу даних (наприклад, с ім'ям dlvr). Послідовність дій при створенні бази даних аналогічна діям вище. Для розміщення файлів бази даних можна вказати той же каталог.

У середовищі SQL Server Management Studio працювати з базою даних можна, використовуючи безпосередньо оператори мови SQL. Для цього необхідно створити один або кілька запитів. Кожен запит може містити довільну кількість операторів мови SQL. Розглянемо послідовність дій при створенні запиту, за допомогою якого будуть створені таблиці бази даних і зв'язку між ними.

1. На панелі інструментів натиснути кнопку New Query
2. Ввести текст запиту, наведений на малюнку 1.18.

```
USE dlvr
CREATE TABLE Поставщики (КодПоставщика int PRIMARY KEY,
                           Адрес text NOT NULL,
                           Примечание text)

CREATE TABLE ФизическиеЛица (КодПоставщика int PRIMARY KEY,
                              Фамилия char(20) NOT NULL,
                              Имя char(20) NOT NULL,
                              Отчество char(20) NOT NULL,
                              НомерСвидетельства char(10)
                              FOREIGN KEY (КодПоставщика) REFERENCES Поставщики(КодПоставщика))

CREATE TABLE ЮридическиеЛица (КодПоставщика int PRIMARY KEY,
                              Название char(20) NOT NULL,
                              НалоговыйНомер char(20),
                              НомерСвидетельстваНДС char(10)
                              FOREIGN KEY (КодПоставщика) REFERENCES Поставщики(КодПоставщика))

CREATE TABLE Договоры (НомерДоговора int IDENTITY (1,1) PRIMARY KEY,
                        ДатаДоговора datetime,
                        КодПоставщика int NOT NULL,
                        Комментарий text
                        FOREIGN KEY (КодПоставщика) REFERENCES Поставщики(КодПоставщика))

CREATE TABLE Поставлено (НомерДоговора int,
                          Товар char(20),
                          Количество decimal(4,0) NOT NULL CHECK (Количество>0),
                          Цена decimal(8,2) NOT NULL CHECK (Цена>0)
                          FOREIGN KEY (НомерДоговора) REFERENCES Договоры(НомерДоговора)
                          PRIMARY KEY (НомерДоговора, Товар))
```

малюнок 1.18

3. Виконати запит. Для цього на панелі інструментів потрібно натиснути кнопку Execute. У тому випадку, якщо текст запиту не містить помилок, на екрані з'явиться вікно Messages з повідомленням Command(s) completed successfully. В іншому випадку буде виведено інформацію про наявні в тексті запиту помилках.



4. У разі успішного виконання запиту далі слід перевірити наявність об'єктів бази даних. У тому випадку, якщо список таблиць відразу не відображається, можна натиснути правою кнопкою миші на ім'я бази даних і в меню вибрати пункт Refresh.
5. Створений запит можна закрити і зберегти з довільним ім'ям (наприклад, SQLQuery\_create\_tables.sql)

За допомогою операторів DDL мови SQL можна не тільки створювати об'єкти бази даних, але і змінювати структуру раніше створених об'єктів. Припустимо, що в таблиці «Поставлено» розмір поля «Кількість» може не відповідати реальним значенням збережених даних. У зв'язку з цим розмір поля необхідно збільшити. Це можна зробити за допомогою наступного запиту (рисунок 1.19)

```
use dlvr  
  
ALTER TABLE Поставлено ALTER COLUMN Количество decimal (5,0) NOT NULL
```

малюнок 1.19

Послідовність дій при створенні і виконанні запиту аналогічна послідовності дій, розглянутих вище. Створений запит можна закрити і зберегти з довільним ім'ям (наприклад, SQLQuery\_alter\_tables.sql)

Розглянемо ще один приклад використання оператора ALTER TABLE. Припустимо, що після того, як база даних була створена, з'ясувалося, що кожен договір поставки також характеризується формою оплати. Форма оплати визначає порядок взаєморозрахунків з постачальником за поставлену продукцію. Можуть, наприклад, використовуватися такі типи оплати як готівкова або безготівкова. Крім того, в будь-який момент може з'явитися якась нова форма оплати. Для того щоб відобразити в базі даних таке структурна зміна, необхідно створити таблицю ФормиОплати з полями КодФормиОплати і НазваніеФормиОплати і зв'язати її з таблицею Договори. Це можна зробити за допомогою наступного запиту (рисунок 1.20). Послідовність дій при створенні і виконанні запиту аналогічна послідовності дій, розглянутих вище. Створений запит можна закрити і зберегти з довільним ім'ям (наприклад, SQLQuery\_alter\_tables1.sql). Після виконання запиту обов'язково потрібно перевірити наявність змін в структурі бази даних.

```
use dlvr  
  
CREATE TABLE формыОплати (КодФормыОплаты int PRIMARY KEY,  
                             НазваниеФормыОплаты char(20) NOT NULL)  
  
ALTER TABLE Договори ADD КодФормыОплаты int NULL  
FOREIGN KEY REFERENCES формыОплати (КодФормыОплаты)
```

малюнок 1.20

Розглянутий запит дозволяє виконати необхідне структурна зміна, однак ім'я створеного зовнішнього ключа не вказано явно і буде визначено довільним чином. Це може створити певні незручності при роботі з цим зовнішнім ключем (наприклад, при його видаленні). Тому розглянемо змінений варіант попереднього запиту, що дозволяє явно вказати ім'я зовнішнього ключа. Текст запиту наведено на малюнку 1.21. Перед виконанням цього запиту необхідно видалити з бази даних зроблені за допомогою попереднього запиту структурні зміни - поле КодФормиОплати з таблиці Договори і таблицю ФормиОплати. Послідовність дій при створенні і виконанні запиту аналогічна послідовності дій, розглянутих вище. Створений запит можна закрити і зберегти з довільним ім'ям (наприклад, SQLQuery\_alter\_tables2.sql). Після виконання запиту обов'язково потрібно перевірити наявність змін в структурі бази даних, зокрема, ім'я створеного в таблиці Договори зовнішнього ключа.

```
use dlvr

CREATE TABLE формиОплати (КодФормиОплати int PRIMARY KEY,
                           НазваниеФормиОплати char(20) NOT NULL)

ALTER TABLE Договори ADD КодФормиОплати int NULL
CONSTRAINT FK_КодФормиОплати FOREIGN KEY REFERENCES формиОплати (КодФормиОплати)
```

малюнок 1.21

Тепер припустимо, що ці структурні зміни виявилися непотрібними і, отже, створену таблицю і зв'язок потрібно видалити. Це також можна зробити за допомогою оператора ALTER TABLE. Текст запиту, за допомогою якого проводиться така видалення, наведено на малюнку 1.22. Послідовність дій при створенні і виконанні запиту аналогічна послідовності дій, розглянутих вище. Створений запит можна закрити і зберегти з довільним ім'ям (наприклад, SQLQuery\_alter\_drop.sql). Після виконання запиту обов'язково потрібно перевірити наявність змін в структурі бази даних.

```
use dlvr

ALTER TABLE Договори DROP CONSTRAINT FK_КодФормиОплати
ALTER TABLE Договори DROP COLUMN КодФормиОплати

DROP TABLE формиОплати
```

малюнок 1.22

## II Введення даних в таблиці бази даних

### II.1 Введення даних в таблиці БД використовуючи інтерактивні засоби SQL Server Management Studio

Увага! Для введення даних за допомогою засобу SQL Server Management Studio, використовуємо раніше створену базу даних delivery.

1. Для введення інформації в таблицю потрібно вибрати таблицю в списку таблиць, клацнувши по ній правою кнопкою миші, і в меню вибрати пункт Open Table. В результаті таблиця буде виведена на екран у вигляді, що дозволяє вводити нові дані або коригувати введені раніше.
2. Використовуючи інтерактивні засоби SQL Server Management Studio необхідно ввести в таблиці бази даних інформацію, наведену на малюнках 1.23 - 1.27.

	КодПоставщика	Адрес	Примечание
►	1	г.Харьков, пр. Ленина, 55, к.108	тел. 32-18-44
	2	г. Киев, пр. Победы, 154, к. 3	
	3	г. Харьков, ул. Пушкинская, 77	тел.33-33-44, 12-34-56, факс 22-12-33
	4	г. Одесса, ул. Дерибасовская, 75	
	5	г. Полтава, ул. Ленина, 15, кв. 43	
*	NULL	NULL	NULL

Малюнок 1.23 - Дані, введені в таблицю «Постачальники»

	КодПоставщика	Фамилия	Имя	Отчество	НомерСвидетельства
►	1	Петров	Павел	Петрович	12345678
	3	Иванов	Илья	Ильич	00123987
	5	Сидоров	Сергей	Степанович	09876541
*	NULL	NULL	NULL	NULL	NULL

Малюнок 1.24 - Дані, введені в таблицю «ФізическієЛіца»

	КодПоставщика	Название	НалоговыйНомер	НомерСвидетельстваНДС
►	2	ООО "Интерфрут"	00123987	19848521
	4	ЗАО "Транссервис"	29345678	25912578
*	NULL	NULL	NULL	NULL

Малюнок 1.25 - Дані, введені в таблицю «ЮрідіческієЛіца»

Увага! При введенні даних в таблицю «Договори» слід враховувати, що номер кожного договору визначається автоматично.

	НомерДоговора	ДатаДоговора	КодПоставщика	Комментарий
►	1	01.09.1999 0:00:00	1	Основание - накладная № 34 от 30/08/99
	2	10.09.1999 0:00:00	1	Основание – счет-фактура № 08-78 от 28/08/99
	3	10.09.1999 0:00:00	3	Основание – счет-фактура № 08-178 от 29/08/99
	4	23.09.1999 0:00:00	3	Основание – заказ № 56 от 28/08/99
	5	24.09.1999 0:00:00	2	Основание – накладная № 74 от 11/09/99
	6	01.10.1999 0:00:00	1	Основание – счет-фактура № 09-12 от 28/09/99
	7	02.10.1999 0:00:00	2	Основание – накладная № 85 от 21/09/99
*	NULL	NULL	NULL	NULL

Малюнок 1.26 - Дані, введені в таблицю «Договори»

	НомерДоговора	Товар	Количество	Цена
►	1	Видеомагнитофон	12	722,33
	1	Компьютер	24	1554,22
	1	Магнитофон	25	655,12
	1	Стереосистема	12	220,45
	1	Телевизор	10	1253,45
	2	Видеомагнитофон	8	450,67
	2	Компьютер	43	1453,18
	2	Магнитофон	5	455,14
	2	Стереосистема	11	511,43
	3	Магнитофон	11	544,00
	3	Монитор	85	545,32
	3	Телевизор	52	899,99
	4	Магнитофон	22	323,19
	4	Принтер	41	350,77
	4	Стереосистема	27	330,55
	4	Телевизор	56	990,56
	5	Видеомагнитофон	17	850,12
	5	Магнитофон	33	585,67
	5	Монитор	44	590,23
	5	Телевизор	14	860,33
	6	Компьютер	32	1850,24
	6	Монитор	51	520,95
	6	Телевизор	34	810,15
	7	Компьютер	15	1234,56
	7	Монитор	22	389,75
	7	Телевизор	62	900,58
*	NULL	NULL	NULL	NULL

Малюнок 1.27 - Дані, введені в таблицю «Поставлено»

## II.2 Введення даних в таблиці бази даних за допомогою мови SQL Використання коштів DML для введення інформації в таблиці бази даних

Увага! Для введення даних за допомогою мови SQL використовуємо раніше створену базу даних dlvr.

Запити можуть містити не тільки оператори DDL, а й оператори DML. Це дозволяє реалізувати основні операції маніпулювання даними. Розглянемо послідовність дій при створенні запиту, за допомогою якого в таблиці створеної бази даних буде введена інформація.

1. На панелі інструментів натиснути кнопку New Query
2. Ввести текст запиту, наведений на малюнках 1.28 - 1.30.

```
USE dlvr
```

```
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (1, 'г. Харьков, пр. Ленина, 55, к.108', 'тел. 32-18-44');
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (2, 'г. Киев, пр. Победы, 154, к. 3', '');
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (3, 'г. Харьков, ул. Пушкинская, 77', 'тел.33-33-44, 12-34-56, факс 22-12-33');
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (4, 'г. Одесса, ул. Дерибасовская, 75', '');
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (5, 'г. Полтава, ул. Ленина, 15, кв. 43', '');

INSERT INTO физическиеЛица
VALUES (3, 'Иванов', 'Илья', 'Ильич', '00123987');
INSERT INTO физическиеЛица
VALUES (1, 'Петров', 'Павел', 'Петрович', '12345678');
INSERT INTO физическиеЛица
VALUES (5, 'Сидоров', 'Сергей', 'Степанович', '09876541');

INSERT INTO ЮридическиеЛица
VALUES (2, 'ООО "Интерфрут"', '00123987', '19848521');
INSERT INTO ЮридическиеЛица
VALUES (4, 'ЗАО "Транссервис"', '29345678', '25912578');
```

малюнок 1.28

```
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19990901', 1, 'Основание - накладная № 34 от 30/08/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('1999/09/10', 1, 'Основание - счет-фактура № 08-78 от 28/08/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19990910', 3, 'Основание - счет-фактура № 08-178 от 29/08/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19990923', 3, 'Основание - заказ № 56 от 28/08/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19990924', 2, 'Основание - накладная № 74 от 11/09/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('1999/10/01', 1, 'Основание - счет-фактура № 09-12 от 28/09/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19991002', 2, 'Основание - накладная № 85 от 21/09/99');
```

малюнок 1.29

```

INSERT INTO Поставлено VALUES (1, 'Телевизор', 10, 1253.45);
INSERT INTO Поставлено VALUES (1, 'Магнітофон', 25, 655.12);
INSERT INTO Поставлено VALUES (1, 'Відеомагнітофон', 12, 722.33);
INSERT INTO Поставлено VALUES (2, 'Стереосистема', 11, 511.43);
INSERT INTO Поставлено VALUES (2, 'Магнітофон', 5, 455.14);
INSERT INTO Поставлено VALUES (2, 'Відеомагнітофон', 8, 450.67);
INSERT INTO Поставлено VALUES (1, 'Стереосистема', 12, 220.45);
INSERT INTO Поставлено VALUES (1, 'Комп'ютер', 24, 1554.22);
INSERT INTO Поставлено VALUES (2, 'Комп'ютер', 43, 1453.18);
INSERT INTO Поставлено VALUES (3, 'Телевизор', 52, 899.99);
INSERT INTO Поставлено VALUES (3, 'Магнітофон', 11, 544.00);
INSERT INTO Поставлено VALUES (3, 'Монитор', 85, 545.32);
INSERT INTO Поставлено VALUES (4, 'Телевизор', 56, 990.56);
INSERT INTO Поставлено VALUES (4, 'Магнітофон', 22, 323.19);
INSERT INTO Поставлено VALUES (4, 'Принтер', 41, 350.77);
INSERT INTO Поставлено VALUES (5, 'Телевизор', 14, 860.33);
INSERT INTO Поставлено VALUES (5, 'Магнітофон', 33, 585.67);
INSERT INTO Поставлено VALUES (5, 'Відеомагнітофон', 17, 850.12);
INSERT INTO Поставлено VALUES (4, 'Стереосистема', 27, 330.55);
INSERT INTO Поставлено VALUES (5, 'Монитор', 44, 590.23);
INSERT INTO Поставлено VALUES (6, 'Телевизор', 34, 810.15);
INSERT INTO Поставлено VALUES (6, 'Комп'ютер', 32, 1850.24);
INSERT INTO Поставлено VALUES (6, 'Монитор', 51, 520.95);
INSERT INTO Поставлено VALUES (7, 'Телевизор', 62, 900.58);
INSERT INTO Поставлено VALUES (7, 'Комп'ютер', 15, 1234.56);
INSERT INTO Поставлено VALUES (7, 'Монитор', 22, 389.75);

```

малюнок 1.30

3. Виконати запит. Для цього на панелі інструментів потрібно натиснути кнопку Execute. У тому випадку, якщо текст запиту не містить помилок, на екрані з'явиться вікно Messages з повідомленнями типу (1 row (s) affected). В іншому випадку буде виведено інформацію про наявні в тексті запиту помилки.
4. У разі успішного виконання запиту далі слід перевірити наявність інформації в таблицях бази даних. Для цього потрібно вибрати таблицю, клацнувши по ній правою кнопкою миші, і в меню вибрати пункт Open Table.
5. Створений запит можна закрити і зберегти з довільним ім'ям (наприклад, SQLQuery\_insert.sql)

### III. Зіставлення створених баз даних

В результаті виконання лабораторних робіт 1 були створені практично однакові бази даних. Проте, в цих базах даних можуть бути певні відмінності. Необхідно проаналізувати об'єкти баз даних, виявити відмінності (якщо такі є) і встановити причину їх появи. Також необхідно створити в новій базі даних dlvр діаграму. При створенні діаграми слід звернути увагу на те, що зв'язки між таблицями в діаграмі з'являються автоматично при їх додаванні в діаграму. Зіставити діаграму dlvр з діаграмою delivery.

### IV. Відключення і підключення бази даних

У процесі роботи з базою даних може виникнути необхідність копіювання файлів бази даних з метою створення резервної копії і т.д. В СУБД Microsoft SQL Server 2008 існує кілька способів створення копій баз даних. Одним з найпростіших способів є відключення і підключення бази даних. Для відключення і підключення бази даних потрібно виконати наступну послідовність дій

1. Вибрати відключається базу даних (в даному випадку - раніше створену базу даних delivery (dlvr).
2. Клацнути по базі даних правою кнопкою миші і в меню вибрати пункт Tasks. Цьому пункту відповідає підміню, в якому потрібно вибрати пункт Detach .... Потім у вікні Detach Database потрібно натиснути кнопку ОК. В результаті відключена база даних зникне зі списку баз даних, а файли бази даних стануть доступними для виконання файлових маніпуляцій.
3. Відключення базу даних можна знову підключити. Для цього у вікні Object Explorer потрібно натиснути правою кнопкою миші по пункту Databases і в меню вибрати пункт Attach .... Потім у вікні Attach Databases потрібно натиснути кнопку Add і вибрати підключається базу даних, вказавши місце розташування її файлів. Після цього потрібно натиснути кнопку ОК. В результаті база даних з'явиться в списку баз даних
4. Перевірити можливість роботи з базою даних (тобто наявність об'єктів знову підключеної бази даних, наявність даних в таблицях і т.д.).

### V. Збереження результатів роботи

Відключити базу даних.

Зберегти файли бази даних                      delivery.mdf, delivery\_log.ldf (dlvr.mdf, dlvр\_log.ldf)

Зберегти файли, що містять тексти запитів - SQLQuery\_create\_tables.sql, SQLQuery\_alter\_tables.sql, SQLQuery\_alter\_tables1.sql, SQLQuery\_alter\_tables2.sql, SQLQuery\_alter\_drop.sql, SQLQuery\_insert.sql

**Вимоги до звіту:**

- 1) коротко описати основні етапи виконання завдання
- 2) зобразити структуру створеної бази даних і відносини між таблицями
- 3) описати введену в базу даних інформацію.
- 4) привести текст запитів, реалізованих при виконанні лабораторної роботи, описати призначення операторів, особливості їх структури та використання;
- 5) описати розбіжності між базами даних, створеними внаслідок Вправи для самостійної роботи 1 і проаналізувати причини появи цих розбіжностей.