МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХАРКІВСЬКИЙ ДЕРЖАВНИЙ ПОЛІТЕХНІЧНИЙ КОЛЕДЖ



Для спеціальності:

151 Автоматизація та комп'ютерноінтегровані технології

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з дисципліни " Організація баз даних "



Харків 2019

Методичні вказівки до виконання лабораторних робіт з дисципліни "Організація баз даних"для студентів освітньо-професійної програми «Обслуговування інтелектуальних інтегрованих систем» спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»

Укладач: В.О.Величко - Харків: ХДПК, 2019, 60с.

Затверджено на засіданні циклової комісії інформаційних технологій Протокол від _____2019 р. №____

Голова циклової комісії _____ М.М. Бочарніков "____" ____ 2019 року

Схвалено методичною радою коледжу Протокол від _____2019р. №____

Голова методичної ради ______ 2019 р.

3MICT

ЛАБОРАТОРНА РОБОТА №1 КОМПОНЕНТИ MICROSOFT SQL	
SERVER	4
ЛАБОРАТОРНА РОБОТА №2 ВВЕДЕННЯ У TRANSACT-SQL	10
ЛАБОРАТОРНА РОБОТА №3 ВИБІРКА ДАНИХ	16
ЛАБОРАТОРНА РОБОТА №4 ЗБЕРЕЖЕНІ ПРОЦЕДУРИ	23
ЛАБОРАТОРНА РОБОТА №5 СИСТЕМА БЕЗПЕКИ MICROSOFT	
SQL SERVER	26
ЛАБОРАТОРНА РОБОТА №6 КЕРУВАННЯ БАЗАМИ ДАНИХ	33
ЛАБОРАТОРНА РОБОТА №7 ВИБІРКА ДАНИХ З ДЕКІЛЬКОХ	
ТАБЛИЦЬ	35
ЛАБОРАТОРНА РОБОТА №8 АНАЛІТИЧНА ВИБІРКА ДАНИХ	41
ЛАБОРАТОРНА РОБОТА №9 ВИКОРИСТАННЯ ПОДАНЬ	46
ЛАБОРАТОРНА РОБОТА №10 ПРОЄКТУВАННЯ БАЗИ ДАНИХ	49
ЛАБОРАТОРНА РОБОТА №11 КЕРУВАННЯ ДАНИМИ	54
Рекомендована література	57

ЛАБОРАТОРНА РОБОТА №1 КОМПОНЕНТИ MICROSOFT SQL SERVER

Ціль: навчити правильно конфігурувати Microsoft SQL Server 2008 як на стороні сервера, так і на стороні клієнта.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №1 «Введення в теорію баз даних», в якій розкриваються основні поняття баз даних, і лекцію №2 «Компоненти Microsoft SQL Server 2008», де детально розглянуті всі основні компоненти MS SQL Server 2008, які будуть конфігуруватися в ході даної лабораторної роботи.

Вимоги до звіту: у процесі виконання лабораторної роботи потрібно складати звіт. Інформація, необхідна для звіту, буде вказана в роботі словами «Запишіть у звіт ...».

Завдання 1. Вивчіть утиліту SQL Server Configuration.

Вказівки до виконання:

1. На сервері запустіть утиліту SQL Server Configuration Manager і з її допомогою визначте список запущених на сервері служб. Запишіть цей список до звіту.

Зауваження. Додаткові можливості з управління роботою служб надає утиліта адміністрування Windows XP Services, доступ до якої здійснюється за допомогою папки Панель керування – Administrative Tools.

2. На сервері зі встановленим MS SQL Server 2008 за допомогою утиліти Services визначте параметри запуску служб MS SQL Server і запишіть їх у звіт. (Якщо немає доступу до утиліти Services, то за допомогою SQL Server Configuration Manager).

3. Визначте, за допомогою яких мережевих бібліотек може бути встановлено з'єднання з MS SQL Server (див. рис. 1.1). Які бібліотеки є активними на даний час? Запишіть цю інформацію до звіту.

🚡 Sql Server Configuration Manager		
File Action View Help		
← → 🗈 🖀 🖗 😫		
🕵 SQL Server Configuration Manager (Local)	Protocol Name	Status
SQL Server Services	🕉 Shared Memory	Enabled
SQL Server Network Configuration	🐺 Named Pipes	Disabled
SOL Native Client 10.0 Configuration	TCP/IP	Enabled
	VIA 🏹	Disabled

Рис. 1.1. Протоколи на стороні серверу, через які до нього можна підключатися

4. За допомогою SQL Server Configuration Manager визначте, на основі яких

мережевих бібліотек клієнт може підключатися до MS SQL Server (див. рис. 1.2). Запишіть список бібліотек у звіт.

🚡 Sql Server Configuration Manager			
File Action View Help			
🕵 SQL Server Configuration Manager (Local)	Name	Order	Enabled
SQL Server Services	🕉 Shared Memory	1	Enabled
SQL Server Network Configuration	TCP/IP	2	Enabled
	👸 Named Pipes	3	Enabled
Aliases	¥TVIA		Disabled

Рис. 1.2. Протоколи на стороні клієнта, через які він може підключатися до серверу

Завдання 2. Створення псевдоніму серверу для клієнтських підключень.

Перед початком роботи з сервером необхідно налаштувати параметри підключення. Для цього потрібно в утиліті SQL Server Configuration Manager у вузлі SQL Native Client Configuration – Client Protocols додати конфігурацію мережевої бібліотеки, потім вказати логічне ім'я серверу, вибрати зі списку необхідну бібліотеку і визначити її параметри. Наприклад, при взаємодії на основі стека TCP / IP потрібно вказати ім'я вузла мережі, на якому встановлений MS SQL Server, і номер TCP-порту, використовуваного при підключенні (див. рис. 1.3).

TCP/IP Properties	? 🛛
Protocol	
General	
Default Port	1433
Enabled	Yes
Keep Alive	30000
Keep Alive Interval	1000
Default Port	
Default port on which connection will b	e made
OK Cancel	Apply Help

Рис. 1.3. Властивості протоколу ТСР/ІР

Вказівки до виконання:

1. Визначте і запишіть в звіт мережеве ім'я комп'ютера, на якому встановлений MS

SQL Server.

2. У контекстному меню (права кнопка миші) для вузла SQL Native Client Configuration – Aliases виконайте команду New alias... Встановіть Alias Name в MyServ, виберіть протокол TCP/IP, в поле Server вкажіть визначене раніше ім'я.

3. Який номер порту повинен бути зазначений при налаштуванні даного з'єднання і чому? Запишіть у звіт.

Завдання 3. Встановіть з'єднання з SQL сервером.

Вказівки до виконання:

1. На робочій станції – клієнтському комп'ютері¹ – запустіть *SQL Server Management Studio* і виберіть зі списку логічне ім'я серверу *MyServ*, визначене в попередньому завданні за допомогою утиліти *SQL Server Configuration Manager*. Якщо потрібного серверу немає в списку, то можна вибрати **«Browse for more...»** і знайти необхідний сервер в списку серверів, до яких може бути виконано підключення.

2. Встановіть з'єднання із сервером *MyServ* з використанням засобів автентифікації MS SQL Server: обліковий запис «sa», пароль відсутній.

3. Для того щоб написати новий запит необхідно виконати команду New Query розташовану на панелі інструментів *SQL Server Management Studio*. В результаті відкриється нова вкладка, яка надає такі можливості:

- заголовок, в якому вказується логічне ім'я серверу, поточна база даних і ім'я користувача, який встановив з'єднання;

- область запиту, яка використовується для введення запитів, переданих MS SQL Server;

– область результатів, в якій відображаються результати виконання запиту, а спосіб відображення задається кнопками Messages (у формі тексту) і Results (у формі таблиці) відповідно.

4. За допомогою команди **SELECT** @@version визначте і запишіть у звіт інформацію про використовувану версію MS SQL Server і операційної системи (результат запиту повинен бути відображений у текстовій формі) (рис. 1.4).

L (N							
1.04	lo column namej						
1 M	licrosoft SQL Server :	008 (RTM) - 10.0.16	500.22 (Intel X86)	Jul 9 2008 1-	4:43:34 C	opyright (c) 1988-2008 Micro	IS

Рис. 1.4. Сведения о версии MS SQL Server

Примітка: Для виконання запиту необхідно виконати команду **Query – Execute (F5)**, а для аналізу правильності його синтаксичного запису можна скористатися командою **Query – Parse (Ctrl+F5).**

SQL Server Management Studio дозволяє відкривати кілька вікон запитів і працювати

¹ Якщо серверна і клієнтська частини MS SQL Server 2008 встановлений на одному комп'ютері, то поняття «сервер» і «клієнтський комп'ютер» в подальшому збігаються.

з декількома базами даних одночасно. В кожному вікні встановлюється власне з'єднання з MS SQL Server на основі різних облікових записів користувачів і їх паролів. Для створення нового підключення використовується команда File – New – Database Engine Query.

Вміст області запиту поточного підключення може бути збережено в файлі на зовнішньому носії командою File – Save.

5. За допомогою панелі *Object Explorer* визначте імена підтримуваних баз даних і які бази даних серверу є системними (для цього потрібно розгорнути вузол **Databases** у панелі *Object Explorer*). Запишіть цю інформацію до звіту.

Завдання 4. Вивчіть параметри конфігурації MS SQL Server.

Конфігурування служб *MSSQLServer* може бути виконано або спеціальною збереженою процедурою, що виконується в утиліті *SQL Server Management Studio*, або графічним способом засобами цієї ж утиліти. Вибір способу не має значення, тому що графічний спосіб здійснює доступ до системних даних за допомогою цієї ж збереженої процедури, тільки в більш наочній формі.

Вказівки до виконання:

1. Для зміни параметрів служби за допомогою SQL Server Management Studio необхідно вибрати потрібний сервер у Object Explorer і в контекстному меню вибрати команду **Properties.** У діалоговому вікні, що з'явилося, можна виконати налаштування всіх необхідних параметрів.

🚦 Server Properties - TEST	SYS-414BCA6\SQL2008	
Select a page	🛒 Script 🔻 🛐 Help	
General Memory Processors Security		
 Connections Database Settings Advanced Permissions 	Name Product Operating System Platform Version Language Memory Processors Root Directory Server Collation Is Clustered	TESTSYS-414BCA6\SQL2008 Microsoft SQL Server Enterprise Edition Microsoft Windows NT 5.1 (2600) NT INTEL X86 10.0.1600.22 English (United States) 1023 (MB) 1 C:\Program Files\Microsoft SQL Server\MSSQL10. Cyrillic_Genera_CLAS False
Connection Server: TESTSYS-414BCA6\SQL2008 Connection: TESTSYS-414BCA6\User	Name Name of the server and instance.	
Progress Ready	Changes to server properties ar of this SQL Server instance. Be documentation.	nd settings may affect the performance, security, and availability fore making any such changes, consult the product
		OK Cancel

2. Відобразіть список параметрів серверу MyServ (див. рис. 1.5).

Рис. 1.5. Властивості MS SQL Server 2008

На вкладці *General* відображаються основні відомості про систему: версія операційної системи, обсяг пам'яті, кількість процесорів і ін., а також параметри запуску служб серверу.

Вкладка *Memory* дозволяє керувати виділенням пам'яті для виконання дій MS SQL Server: або динамічне керування пам'яттю, або встановити фіксований розмір.

За допомогою вкладки Security визначається тип автентифікації користувачів, також

визначаються параметри аудиту доступу до серверу. Можна налаштувати сервер на використання певного облікового запису, під яким буде запускатися служба *MSSQLServer*.

Вкладка *Connections* дозволяє конфігурувати клієнтські підключення до сервера. Максимальну кількість користувачів, які можуть одночасно підключитися до серверу. Якщо вказано нульове значення, то їх кількість становить 32767.

Вкладка *Advanced* містить деякі загальні установки сервера. Наприклад, визначається мова за замовчуванням для повідомлень серверу або регулюється підтримка 2000 року, яка визначає, як будуть інтерпретуватися дві останні цифри року.

За допомогою вкладки *Database Settings* вказуються налаштування новостворюваних баз даних: параметри індексів і роботи з пристроями резервного копіювання, час відновлення бази даних.

3. Визначте і запишіть в звіт кореневий каталог серверу, кількість процесорів в системі, тип автентифікації користувачів і максимальну кількість користувачів, підтримуваних сервером.

4. Вивчіть властивості MS SQL Server, доступні в цьому діалозі.

ЛАБОРАТОРНА РОБОТА №2 ВВЕДЕННЯ У TRANSACT-SQL

Ціль: вивчити основні оператори і алгоритмічні конструкції Transact-SQL.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №3 «Загальні свідчення про Transact-SQL», оскільки в ній представлено загальний опис та призначення мови SQL, а також розглянуті всі алгоритмічні конструкції, доступні в SQL.

Вимоги до звіту: за результатами роботи потрібно представити набір SQL-скриптів, що вирішують завдання з розділу «Самостійна робота».

Завдання 1. Встановити з'єднання із сервером MyServ за допомогою утиліти Management Studio.

Вказівки до виконання:

1. Запустіть *SQL Server Management Studio* через меню Пуск – Програми – Microsoft SQL Server 2008.

2. Оберіть тип автентифікації: SQL Server Authentication. Задайте User name: sa, i Password: порожній і натисніть кнопку Connect.

Завдання 2. Визначити, чи має користувач право вибірки з таблиці Product.

Вказівки до виконання:

1. Встановіть поточною базою даних *AdventureWorks2008*. Це можна зробити одним із таких способів:

- виконайте команду USE AdventureWorks2008;
- оберіть у випадному списку *AdventureWorks2008* (див. рис. 2.1).



2. За допомогою функції object_id отримаймо ідентифікатор таблиці Product.

3. Функція **Permissions** повертає 32-бітову маску, в якій кожен біт означає право поточного користувача на певну дію. При цьому значення біта залежить від параметрів, з якими була викликана функція. Функція **Permissions** може бути викликана одним з трьох способів:

Без параметрів – повертає список дозволених дій користувача, які він може виконувати в поточній БД;

з одним параметром *obect_id* – повертає список допустимих дій над об'єктом, наприклад, над таблицею;

з двома параметрами: *object_id* і атрибут – повертає дозволені дії над атрибутом.

У таблиці наведено список дозволених дій відповідно до різних способів виклику функції **Permissions.** Для кожної дії зазначений номер і десяткове значення відповідного біту.

Номер	Біт (у		Дозволена дія	
біту	десятковій	Без параметрів	Object_id	Object_id
	формі)			і атрибут
0	1	CREATE	SELECT ALL	SELECT
		DATABASE		
1	2	CREATE TABLE	UPDATE ALL	UPDATE
2	4	CREATE	REFERENCES ALL	REFERENCES
		PROCEDURE		
3	8	CREATE VIEW	INSERT	
4	16	CREATE RULE	DELECT	
5	32	CREATE	EXECUTE	
		DEFAULT		
6	64	BACKUP		
		DATABASE		
7	128	BACKUP LOG		
8	256	Зарезервовано		
12	4096		SELECT ANY	
13	8192		UPDATE ANY	
14	16384		REFERENCES ANY	

В даному випадку нас цікавить молодший (нульовий) біт – право на SELECT ALL.

4. Приклад виклику функції **Permissions** (рис. 2.2):

SELECT Permissions (object_id('production.product'))

	Results	6	Messages
	(No co	olumn	name)
1	18811	0851	1

Рис. 2.2. Результат функції Permissions

5. Для визначення того, чи не заданий цікавий для нас біт, можна використати бітову операцію «І» (&), тоді кінцевий варіант буде виглядати наступним чином:

IF Permissions (object_id('production.product'))&1=1

SELECТ 'Есть право выборки'

ELSE

SELECT 'Права выборки нет'



Рис. 2.3. Результат функції Permissions для виводу текстового опису

6. Замість оператора **SELECT** можна скористатися оператором **PRINT**, який просто дозволяє виводити дані у вигляді тексту, а не у вигляді таблиці, як це робить **SELECT**. Тоді код буде виглядати так:

IF Permissions (object_id('production.product'))&1=1 PRINT 'Есть право выборки' ELSE PRINT 'Права выборки нет'

А результат буде виведений не на вкладку Results, а на вкладку Messages (рис. 2.4).



Рис. 2.4. Результат використання команди PRINT

Завдання 3. Вивести ім'я комп'ютера, на якому виконується команда. Якщо ім'я комп'ютера більше дев'яти літер, то вивести тільки перші шість букв.

Вказівки до виконання:

- 1. Для отримання імені комп'ютера використовується функція HOST_NAME().
- 2. Для визначення довжини рядка можна скористатися функцією DataLength.
- 3. Приклад виконання завдання:

```
IF (DATALENGTH(HOST_NAME()) / 2 > 9)
SELECT LEFT(HOST_Name(), 6) + '...'
ELSE
SELECT HOST_Name()
```

Зауваження. Зверніть увагу, що результат функції **DATALENGTH** ми розділили на два. Це обумовлено тим, що **HOST_NAME()** повертає тип **nvarchar**, в якому під кожен символ відводиться два байти.

Завдання 4. Порахувати кількість цифр в числі.

Вказівки до виконання: 1. Один з можливих варіантів вирішення даного завдання (див. рис.2.5):

```
DECLARE @num int, @cnt int
SET @num = 19
IF (@num = 0) SET @cnt = 1
ELSE BEGIN
SET @cnt = 0
WHILE (@num <> 0) BEGIN
SET @cnt = @cnt + 1
SET @num = @num / 10
END
END
SELECT @cnt AS 'Количество цифр'
```

	Results	🛅 Messages
	Колич	ество цифр
1	2	

Рис. 2.5. Результат роботи коду

Самостійна робота

1. Оптимізувати код рішення завдання 4: позбутися від умовного оператора, повинен залишитися тільки цикл WHILE, при цьому код повинен як і раніше працювати як для нуля і позитивних чисел, так і для негативних.

2. Визначити, чи має користувач право доступу до поля '*CardNumber*', таблиці *CreditCard* в базі даних *AdventureWorks2008*. Якщо доступ є, то вивести «Доступ є», інакше - «В доступі відмовлено».

3. Написати програму перерахунку ваги з фунтів в кілограми (1 фунт дорівнює 453,6 г). Результат повинен бути виведений, наприклад, у такий спосіб:

3.3 фунт(и/ів) – це 1 кг 496 г.

4. Вивести інформацію про поточні іменах сервера, облікового запису та користувача бази даних в наступному вигляді:

Ви ввійшли на сервер User400-01\SQL2008 як User400-01\User з правами доступу dbo.

5. Визначити кількість годин і хвилин, що минули з часу запуску служб MS SQL Server.

6. Визначити, чи є поточний рік високосним.

7. Знайти суму чисел в заданому рядку символів.

8. Визначити величину оплати за відправку телеграми. Ознакою завершення телеграми є точка. Вартість одного слова 33 коп, результат вивести із зазначенням кількості карбованців і копійок. Рядок може містити довільне число прогалин.

9. Використовуючи шифр Цезаря, зашифруйте заданий рядок тексту. Ідея даного методу шифрування — алфавіт розміщується наче за годинниковою стрілкою. Для шифровки букви тексту замінюються буквами, віддаленими на задане число букв (зрушення) за годинниковою стрілкою.

10. Відоме прізвище, ім'я та по батькові користувача. Знайти число його особистості. Правило отримання числа особистості: кожній букві прирівняти число – порядковий номер букви в алфавіті. Ці числа складаються, якщо отримана сума не є однозначним числом, то цифри числа знову складаються і так до тих пір, поки не буде отримано однозначне число.

ЛАБОРАТОРНА РОБОТА №3 ВИБІРКА ДАНИХ

Ціль: навчити використовувати оператор SELECT і його основні розділи для вибірки даних відповідно до заданих умов.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №4 «Вибірка даних», в якій вивчається один з основних операторів мови SQL - SELECT і пояснюється призначення кожного з його розділів.

Вимоги до звіту: за результатами роботи представити набір SQL-скриптів, що вирішують завдання з розділу «Самостійна робота».

Завдання 1. Встановити з'єднання із сервером MyServ за допомогою утиліти Management Studio.

Вказівки до виконання:

1. Запустіть *SQL Server Management Studio* через меню Пуск – Програми – Microsoft SQL Server 2008.

2. Оберіть тип автентифікації: SQL Server Authentication. Задайте User name: sa, i Password: порожній та натисніть кнопку Connect.

Завдання 2. Отримайте список країн із зазначенням їх коду та останньої дати зміни запису з таблиці *CountryRegion* БД *AdventureWorks2008*.

Вказівки до виконання:

1. Для виконання завдання нам будуть потрібні атрибути CountryRegionCode, [Name], ModifiedDate з таблиці Person.CountryRegion. Зверніть увагу, що перед назвою таблиці використовується ще назва схеми Person, призначеної для управління об'єктами, пов'язаними з працівниками та департаментами.

2. Для перейменування стовпців скористаємося ключовим словом АS.

3. Переконайтеся, що БД *AdventureWorks2008* є поточною, і виконайте наступний запит:

SELECT CountryRegionCode AS 'Код', [Name] AS 'Страна', ModifiedDate AS 'Дата изменения' FROM Person.CountryRegion

Результат виконання запиту показаний на рис. 3.1.

E F	Results	🛅 Messages					
	Код	Страна	Дата изменения				
1	AD	Andorra	1998-06-01 00:00:00.000				
2	AE	United Arab Emirates	1998-06-01 00:00:00.000				
3	AF	Afghanistan	1998-06-01 00:00:00.000				
4	AG	Antigua and Barbuda	1998-06-01 00:00:00.000				
5	Al	Anguilla	1998-06-01 00:00:00.000				
6	AL	Albania	1998-06-01 00:00:00.000				
7	AM	Armenia	1998-06-01 00:00:00.000				
8	AN	Netherlands Antilles	1998-06-01 00:00:00.000				
9	AO	Angola	1998-06-01 00:00:00.000				
10	AQ	Antarctica	1998-06-01 00:00:00.000				
11	AR	Argentina	1998-06-01 00:00:00.000				
🕗 ણ	Jer	TESTSYS-414BCA6\SQL	2008 (10 TESTSYS-414	BCA6\User (52)	AdventureWorks2008	00:00:00	

Рис. 3.1. Список країн

Завдання 3. Отримайте список співробітників із зазначенням прізвища і першого символу імені, а також його ідентифікаційного номера. Список повинен бути впорядкований за прізвищем та ім'ям співробітника.

Вказівки до виконання:

1. Перший стовпець ми повинні будемо перейменувати за допомогою оператора AS, так як в ньому необхідно об'єднати дані з двох стовпців: *LastName* i *FirstName*.

- 2. Для отримання тільки першої літери імені скористаємося функцією Substring.
- 3. Для сортування кінцевої таблиці необхідно скористатися розділом ORDER BY.
- 4. Виконайте наступний код:

SELECT LastName+' '+Substring(FirstName,1,1)+'.' as [Имя сотрудника], BusinessEntityID FROM Person.Person

ORDER BY [Имя сотрудника]

Результат виконання запиту показаний на рис. 3.2.

iii F	lesults 🛅 Messag	ges												
	Имя сотрудника	BusinessEntityID												
1	Abbas S.	285												
2	Abel C.	293												
3	Abercrombie K.	295												
4	Abercrombie K.	2170												
5	Abercrombie K.	38												
6	Abolrous H.	211												
7	Abolrous S.	2357												
8	Acevedo H.	297												
9	Achong G.	291												
10	Ackerman P.	299												
11	Ackerman P.	121												
🕑 Qu	TESTSYS-414BO	TA6\SQL2008 (10	TESTSYS-414BCA6\User (52)	AdventureWorks2008	00:00:00	Ī	19	1997;	19972 ro	19972 row	19972 row	19972 row	19972 row:	19972 rows

Рис. 3.2. Список співробітників

Зауваження. За замовчуванням сортування здійснюється за зростанням, що відповідає зарезервованому слову ASC (може не вказуватися), для сортування в порядку спадання вказується – DESC.

Завдання 4. Отримайте список товарів, ціна яких знаходиться в діапазоні від \$ 12 до \$ 20, відсортувавши його за ціною.

Вказівки до виконання:

1. Для вибору записів по заданому критерію необхідно скористатися розділом **WHERE**.

2. Можна використати складену умову, тоді код буде виглядати так:

SELECT [Name], ListPrice From Production.Product WHERE (ListPrice>=12) and (ListPrice<=20) ORDER BY 2

3. Інший варіант: замінити два оператора порівняння одним логічним оператором **BETWEEN**, за допомогою якого можна отримати відповідь на питання, чи лежить величина в зазначеному діапазоні:

SELECT [Name], ListPrice From Production.Product WHERE ListPrice Between 12 and 20 ORDER BY ListPrice

Результат виконання запиту показаний на рис. 3.3.

📰 Results 📑 Messages	
Name	ListPrice
1 Taillights - Battery-Powered	13.99
2 Minipump	19.99

Рис. 3.3. Список товарів від \$12 до \$20

Завдання 5. Виведіть імена співробітників і їх посад, які проживають за межами USA.

Вказівки до виконання:

1. Для пошуку за шаблоном символьних рядків використовується логічний оператор LIKE, який найчастіше застосовується в ситуаціях, коли не відомий точний збіг.

2. У шаблоні нам буде потрібно вказати службовий символ %, який має на увазі будь-який рядок, що складається з 0 і більше символів.

3. Використовуючи таблицю HumanResources.vEmployee складемо запит:

Select LastName, JobTitle From HumanResources.vEmployee Where CountryRegionName Not Like '%United States%'

🔲 F	lesults 🛅 Message	es
	LastName	JobTitle
1	Pak	Sales Representative
2	Saraiva	Sales Representative
3	Tsoflias	Sales Representative
4	Valdez	Sales Representative
5	Vargas	Sales Representative
6	Varkey Chudukatil	Sales Representative

Результат виконання запиту показаний на рис. 3.4.

Рис. 3.4. Співробітники, які проживають за межами США

Завдання 6. Отримайте список співробітників з США і Канади.

Вказівки до виконання:

1. Для визначення відповідності вираження одному з перерахованих в заданому списку значень застосовується логічний оператор **IN**. Даний оператор завжди може бути записаний і у вигляді групи умов, об'єднаних операндом **OR**.

2. Запит з використанням оператора **IN** буде виглядати наступним чином:

Select LastName, JobTitle From HumanResources.vEmployee WHERE CountryRegionName IN ('United States', 'Canada')

Результат виконання запиту показаний на рис. 3.5.

B	esults 🛅 Mess	ages			
	LastName	JobTitle			
1	Abbas	Pacific Sales Manager			
2	Abercrombie	Production Technician - WC60			
3	Abolrous	Quality Assurance Manager			
4	Ackerman	Shipping and Receiving Supervisor			
5	Adams	Production Technician - WC60			
6	Ajenstat	Database Administrator			
7	Alberts	European Sales Manager			
8	Alderson	Production Technician - WC45			
9	Alexander	Quality Assurance Technician			
10	Altman	Facilities Manager			
11	Anderson	Production Technician - WC60			
🕗 Qu	er TESTSYS-4	14BCA6\SQL2008 (10 TESTSYS-4	14BCA6\User (52)	AdventureWorks2008	00:00:00

Рис. 3.5. Співробітники з США і Канади

Завдання 7. Отримайте список товарів, у яких не вказано колір.

Вказівки до виконання:

1. Для перевірки відсутності значення використовується функція вибірки **IS NULL**.

2. Складемо наступний запит по таблиці Product:

SELECT [Name] FROM Production.Product WHERE Color IS NULL

Результат виконання запиту показаний на рис. 3.6.

B	esults 🚹 Messages				
	Name	l.			
1	Adjustable Race]			
2	Bearing Ball				
3	BB Ball Bearing				
4	Headset Ball Bearings				
5	Blade				
6	Crown Race				
7	Chain Stays				
8	Decal 1				
9	Decal 2				
10	Down Tube				
11	Mountain End Caps				
🖉 Qu	er TESTSYS-414BCA	6\SQL2008 (10	TESTSYS-414BCA6\User (52)	AdventureWorks2008	00:00:00

Рис. 3.6. Товари, для яких не вказано колір

Самостійна робота

1. Відкрийте інформацію про способи доставки товару (*ShipMethod*), якщо активний користувач має право отримання цих даних.

2. Сконструюйте запит, який повертає список співробітників (таблиця *Employee*) із зазначенням їх *EmployeeID*, дати народження (в форматі *dd.mm.yyyy*) і віку на сьогоднішній день.

Вказівка: для конвертації значення одного типу в значення іншого типу використовується функція: CONVERT (тип, вираз [, стиль]).

3. Визначте список товарів, поставка яких припинена (використовуючи *SellEndDate*), із зазначенням їх кольору.

4. Отримати розташування складів з *Production.Location*, що містять '*Paint*' у своїй назві.

5. Відобразити список країн, в яких зареєстровані замовники. Вибірку здійснювати з подання *vIndividualCustomer*, а не з таблиці, тому що в таблиці недостатньо відомостей для такого запиту.

6. Вивести список співробітників (просто вказати їх *ID*), у яких завтра день народження, вказавши при цьому їх вік. Якщо таких співробітників немає, то вивести повідомлення про це. Рішення оформити у вигляді збереженої процедури.

Вказівка: використайте логічний оператор **EXISTS** (запит), який приймає істинне значення, якщо зазначений запит повертає хоча б один рядок.

7. Відкрийте список співробітників (просто вказавши їх *ID*), які в наступному місяці будуть відзначати ювілей, із зазначенням дати ювілею.

8. Визначте 10 найдорожчих товарів. Рішення оформити у вигляді збереженої процедури.

ЛАБОРАТОРНА РОБОТА №4 ЗБЕРЕЖЕНІ ПРОЦЕДУРИ

Ціль: навчити створювати і викликати користувальницькі збережені процедури, а також передавати в них параметри.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №3 «Загальні відомості про Transact-SQL», в якій дані основи синтаксису мови SQL, і лекцію №5 «Допоміжні об'єкти бази даних», де наведено базові відомості про збережені процедурах.

Вимоги до звіту: за результатами роботи представити набір SQL-скриптів, що вирішують завдання з розділу «Самостійна робота».

Завдання 1. Створіть збережену процедуру, яка виводить ім'я комп'ютера, на якому виконується команда. Якщо ім'я комп'ютера більше дев'яти букв, то вивести тільки перші шість букв.

Вказівки до виконання:

- 3. Для отримання імені комп'ютера скористайтеся функцією **HOST_NAME**().
- 4. Для створення процедури виконайте наступний код:

CREATE PROCEDURE Test AS IF (DATALENGTH(HOST_NAME()) / 2 > 9) SELECT LEFT(HOST_Name(), 6) + '...' ELSE SELECT HOST_Name()

Зауваження. Процедура буде створена у поточній базі даних.

5. Переконайтеся, що збережена процедура була створена (див. рис. 4.1).



Рис. 4.1. Список збережених процедур БД

6. Викличте створену збережену процедуру, виконавши наступний код:

TEST

Завдання 2. Створіть збережену процедуру, що бере число в якості подільника і повертає кількість цифр у ньому через параметр OUTPUT.

Вказівки до виконання:

1. Створіть збережену процедуру **DigitCount** з двома параметрами:

```
CREATE PROCEDURE DigitCount

@num int,

@cnt int OUTPUT

AS

IF (@num = 0) SET @cnt = 1

ELSE BEGIN

SET @cnt = 0

WHILE (@num > 0) BEGIN

SET @cnt = @cnt + 1

SET @num = @num / 10

END

END
```

2. Викличте процедуру **DigitCount** і переконайтеся, що вона працює правильно:

DECLARE @num int = 1249

DECLARE @cnt int EXEC DigitCount @num, @cnt OUTPUT SELECT @cnt

Зауваження. Зверніть увагу, що в даному випадку використання ключового слова **EXEC** обов'язково перед викликом збереженої процедури, тому що збережена процедури вказана в блоці інших SQL команд, а не виконується одна.

Самостійна робота

1. Визначте кількість системних збережених процедур.

Вказівка. Оформіть розв'язок наступних завдань у вигляді збережених процедур.

2. Визначте час року для заданої дати: осінь, зима, весна чи літо.

Зауваження. Для керування форматом дати використовується наступна команда: **SET DATEFORMAT** *Формат*.

Опція $\Phi opmam$ вказує формат введення дати і може набувати таких значень – mdy, dmy, ymd, myd i dym (де m, d i y – день, місяць і рік відповідно).

3. Відома дата народження користувача. Перевірити, виповнилося йому чи ні повних 16 років.

4. Встановити, чи є рядок символів адресою електронної пошти. Зробіть так, щоб вихідний текст процедури не можна було переглянути. Доведіть правильність дій.

Вказівка: в рядку повинен бути присутнім символ @, після якого обов'язково наявність хоча б однієї точки.

5. Видаліть усі створені збережені процедури.

ЛАБОРАТОРНА РОБОТА №5 СИСТЕМА БЕЗПЕКИ MICROSOFT SQL SERVER

Ціль: навчити використовувати системні збережені процедури для управління іменами входу MS SQL Server і користувачами баз даних, а також дозволяти і забороняти виконання певних дій деякому користувачу.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №6 «Система безпеки в базах даних», в якій детально розглянуті питання управління обліковими записами і правами користувача.

Вимоги до звіту: за результатами роботи надати звіт зі скріншотами, що містять SQL-команди і результати їх виконання для кожного завдання з розділу «Самостійна робота».

Завдання 1. Встановити з'єднання із сервером MyServ за допомогою утиліти Management Studio.

Вказівки до виконання:

1. Запустіть *SQL Server Management Studio* через меню Пуск – Програми – Microsoft SQL Server 2008.

2. Оберіть тип автентифікації: SQL Server Authentication. Вкажіть User name: sa, i Password: порожній і натисніть кнопку Connect.

Завдання 2. Визначте список ролей серверу.

Вказівки до виконання:

1. Створіть новий запит або через команду меню File – New – Query with Current Connection або за допомогою кнопки New Query на панелі інструментів.

2. У вкладці SQLQuery1.sql виконайте команду sp_helpsrvrole (див. рис. 5.1).

🛄 F	lesults [) Mes	sages				
	ServerRo	le	Desci	ription			
1	sysadmin		System Administrators				
2	securitya	dmin	Security Administrators				
3	serverad	min	Server Administrators				
4	setupadn	nin	Setup Administrators				
5	processa	dmin	Process Administrators				
6	diskadmi	n	Disk Administrators				
7	dbcreato	r	Database Creators				
8	bulkadmi	n	Bulk Insert Administrators				

Рис. 5.1. Серверні ролі MS SQL Server

Зауваження. Для більш наочного подання даних використовуйте спосіб відображення інформації у вигляді таблиці (кнопки Results to Grid/Results to Text на

панелі інструментів або через команди меню Query – Results To).

Завдання 3. Створіть і налаштуйте новий обліковий запис *TempUser* для входу в SQL Server.

Вказівки до виконання:

1. Щоб додати обліковий запис використовуйте збережену процедуру **sp_addlogin:** sp_addlogin 'TempUser', 'Password!'

Зауваження. Для отримання довідки за командами Transact-SQL і збереженим процедурам можна скористатися утилітою SQL Server Management Studio. Для цього необхідно виділити ім'я оператору і натиснути клавішу **F1**.

2. Переконайтеся, що обліковий запис був доданий за допомогою збереженої процедури **sp_helplogins** (див. рис. 5.2).

H	esults 📑 Messages									
	LoginName	SI	D					D	^	^
3	##MS_PolicySigningCertificate##	0	x010	600000000000901000	00067D60BBB80C	50C84	46963875	m		
4	##MS_PolicyTsqlExecutionLogin##	0	x8F6	51FE8547A4644A0C0	6CA83723A876			m		
5	##MS_SQLAuthenticatorCertificate##	0	x010	600000000000901000	000088495742251	B68D	77D258B9	m		
6	##MS_SQLReplicationSigningCertificate	## 0	x010	600000000000901000	000060CB1FFC683	3DC2F	630DAA1	m		
7	##MS_SQLResourceSigningCertificate#	# 0	x010	600000000000901000	00068B4DC4C074	F8B9E	C20DC9A	m		
8	NT AUTHORITY\SYSTEM	0	x010	100000000000512000	0000			m		
9	sa	0	x01					m		
10	TempUser	0	x04E)7B2AAD9A43741AD0	B258AB8CDED04			Α		
11	TESTSYS-414BCA6\User	0	x010	500000000000515000	0009E407E146250	CBCOE	8AA7323	m	~	
<								>	-	
	LoginName	DBNar	me	UserName		Use	rOrAlias		^	≣
19	sa	maste	r	db_owner		Mer	nberOf			
20	sa	maste	r	dbo		Use	r			
21	sa	model		db_owner		Mer	nberOf			
22	sa	model		dbo		Use	r			
23	sa	msdb		db_owner		Mer	nberOf			
24	sa	msdb		dbo		Use	r			
25	sa	tempd	Ь	db_owner		Mer	nberOf			
26	sa	tempd	Ь	dbo		Use	r			
27	TempUser Adv		nt	TestUser		Use	r			
28	TESTSYS-414BCA6\User	Sales	s db_owner			MemberOf				
29	TESTSYS-414BCA6\User	Sales		dbo		Use	r			
30	TESTSYS-414BCA6\User	Unive	r	db_owner		Mer	nberOf			¥
🕗 Qu	ery TESTSYS-414BCA6\SQL2008 (10.	TES	STSY	S-414BCA6\User (52)	AdventureWorks2	2008	00:00:01	42 r	ow:	5

Рис. 5.2. Список імен користувачів MS SQL Server

3. Спробуйте увійти на сервер під створеним обліковим записом.

4. Зайдіть знову під обліковим записом **sa**, тому що для подальших дій знову будуть потрібні права адміністратора.

5. Для надання облікового запису для входу вбудованій серверній ролі використовується процедура:

sp_addsrvrolemember 'TempUser', 'securityadmin'

Завдання 4. Визначте список ролей бази даних AdventureWorks2008 і членів ролі db_owner.

Вказівки до виконання:

1. Виконайте збережену процедуру **sp_helprole** для отримання списку як вбудованих, так і визначених користувачем ролей бази даних.

2. За допомогою команди **sp_helprolemember** '**db_owner**' визначте членів ролі *db_owner* (див. рис. 5.3).

DbRole MemberName MemberSID	🔲 Results 📑	Messages	
	DbRole	MemberName	MemberSID
1 db_owner dbo 0x01010000000000512000000	1 db_owner	dbo	0x01010000000000512000000

Рис. 5.3. Список членів ролі *db_owner*

Завдання 5. Створіть нового користувача бази даних для логіну TempUser.

Вказівки до виконання:

1. За допомогою збереженої процедури додайте користувача:

sp_adduser 'TempUser', 'MyFirstUser'

2. За допомогою процедури **sp_helpuser** переконайтеся, що користувач був доданий. Яка роль йому була надана?

3. Додайте користувачу роль db_datareader:

sp_addrolemember 'db_datareader, 'MyFirstUser'

Завдання 6. Налаштуйте права доступу користувачу Andy: надайте явно право тільки для вибірки з таблиці Product і оновлення тільки полів Name і ListPrice цієї таблиці.

Вказівки до виконання:

1. С помощью следующей команды пользователю *TestUser* базы данных *AdventureWorks2008* предоставляются права выборки и изменения данных таблицы *Orders* этой базы данных:

GRANT select, update on AdventureWorks2008.Production.WorkOrder to TestUser

2. Наступна команда надає користувачеві *Andy* права тільки вибірки даних полів *Name* і *ListPrice* таблиці *Product* бази даних *AdventureWorks2008:*

GRANT select on AdventureWorks2008.Production.Product (Name, ListPrice) to Andy

Завдання 7. Вивчіть виконання вищезазначених функцій за допомогою графічного інтерфейсу утиліти *Management Studio*.

Вказівки до виконання:

1. Перегляд списку наявних облікових записів і їх параметрів здійснюється вибором групи *Logins* у папці **Security** серверу.

2. Для створення нового облікового запису для входу необхідно виконати команду **New Login...** контекстного меню **Logins,** і діалоговому вікні, що з'явилося, вказати:

• вкладка *General*: ім'я користувача, тип автентифікації (при автентифікації засобами MS SQL Server – задати пароль), базу даних, до якої користувач підключається автоматично, мова за замовчуванням;

• вкладка *Server Roles*: ролі серверу, в які буде входити створюваний обліковий запис;

• вкладка User Mapping: доступ до однієї із створених на сервері бази даних, в полі User ввести ім'я користувача бази даних і включити створюваного користувача в одну існуючих ролей.

Зауваження. Для зміни параметрів наявного облікового запису користувача для входу необхідно вибрати її зі списку і виконати команду контекстного меню **Properties**, для видалення – **Delete.**

3. Для відображення списку ролей сервера необхідно вибрати групу Server Roles у папці Security серверу. Перегляд користувачів, які входять в цю роль і дозволів, наданих їй, здійснюється виконанням команди контекстного меню Свойства.

Зауваження. Вбудовані ролі сервера не можуть бути вилучені із системи, і не можна змінити визначені для них дозволи. Також заборонено створювати і власні серверні ролі.

4. Для перегляду і керування параметрами користувачів деякої бази даних призначена група Security/Users цієї бази. Облікові записи відображаються в полі User Name, а в полі Login Name – відповідні їм облікові записи для входу.

Для створення нового користувача бази даних необхідно виконати команду New User..., потім у полі User name ввести ім'я користувача, а в списку Login Name вибрати відповідний обліковий запис для входу. Можна також включити користувача в ролі бази даних.

Зауваження. Для зміни параметрів облікового запису служить команда **Properties,** а для видалення – **Delete.**

5. Для відображення списку ролей бази даних використовується група *Roles*. Для перегляду користувачів, що входять в цю групу, необхідно виконати команду **Properties.**

6. Щоб призначити повноваження об'єкту безпеки необхідно вибрати його в групі *Users* (для зміни дозволу конкретного користувача бази даних) або в групі *Roles* (для дозволів певної ролі). Для цих цілей використовується вкладка **Securables**.

У вкладці перераховані всі об'єкти бази даних, з можливими правами доступу. Можна встановити один з трьох станів доступу: *надання* (галочка), *заборона* (хрестик) і *неявне відхилення* (порожнє поле) – у відповідному полі.

Завдання 8. Скасуйте надання ролі облікового запису і видаліть обліковий запис *TempUser*.

Вказівки до виконання:

1. Скасування присвоєної користувачеві ролі може бути виконана за допомогою процедури:

sp_droprolemember 'db_datareader', 'MyFirstUser'

2. Для видалення користувача БД використовується процедура:

sp_dropuser 'MyFirstUser'

3. Скасування присвоєння облікового запису певної ролі виконується за допомогою збереженої процедури:

sp_dropsrvrolemember 'TempUser', 'securityadmin'

4. Для видалення облікового запису виконайте збережену процедуру:

sp_droplogin 'TempUser'

Самостійна робота

1. Визначте список всіх ролей серверу і дії, дозволені користувачам ролі *dbcreator*.

2. Встановіть, яка серверна роль надана системному обліковому запису sa.

3. Визначте, користувач якої ролі має можливість створення і видалення облікових записів для входу.

4. Зміна пароля облікового запису користувача для входу виконується за допомогою процедури *sp_password*.

5. Створіть власний обліковий запис для входу з підключенням до бази даних *AdventureWorks2008*, доведіть правильність виконаних дій. Створеному обліковому запису надайте права на створення і зміну баз даних, доведіть правильність виконаних дій.

Підключіться до MS SQL Server, використовуючи створений обліковий запис, і створіть ще один обліковий запис користувача для входу, результат поясніть.

6. Створіть користувача *manager* бази даних *AdventureWorks2008* на основі створеного раніше облікового запису для входу. Доведіть правильність виконаних дій.

7. Користувачу *manager* надайте роль, що володіє тільки можливістю перегляду вмісту бази даних *AdventureWorks2008*.

Зауваження. Для перевірки правильності виконаних дій можна виконати довільний запит до цієї бази даних, наприклад, такий, що відображає вміст таблиці *Production.WorkOrder*:

SELECT * FROM Production.WorkOrder.

8. Користувачеві *manager* забороніть перегляд даних БД *AdventureWorks2008*, надавши необхідну роль. Як довести правильність внесених змін?

9. Яка кількість користувачів бази даних може бути створена на основі одного облікового запису користувача для входу? Відповідь обгрунтуйте.

10. Засобами SQL Server Management Studio створіть обліковий запис користувача для входу на основі автентифікації SQL, що підключається за замовчуванням до бази даних AdventureWorks2008 і що має права серверної ролі diskadmin.

11. Визначте список користувачів, що входять в роль diskadmin і її дозволи.

12. У базі даних *AdventureWorks2008* створіть користувача на основі створеного раніше облікового запису для входу.

13. Для створеного раніше користувача бази даних *AdventureWorks2008* визначте, членом якої ролі він є і яке її призначення. Чи має даний користувач право вибірки даних з таблиці *Production.Product* цієї бази даних? Відповідь обґрунтуйте і перевірте, виконавши вилучення даних командою SELECT * *from Production.Product*.

14. В базі даних Adventure Works2008 створіть роль managers. Для цієї ролі визначте право вибірки даних з таблиці Production.Product бази даних Adventure Works2008. Надайте роль managers створеному раніше користувачеві. Чи має тепер цей користувач право вибірки даних? Перевірте зроблений висновок. До яких ще об'єктів бази даних Adventure Works2008 має право доступу цей користувач? Обґрунтуйте і перевірте висновок.

15. Створіть користувача, що має доступ до бази даних AdventureWorks2008 і що належить до групи clerks. Для цієї ролі визначте можливість вибірки даних з таблиці *Production.Location* тільки для полів *Name* і Availability. Для перевірки правильності виконаних дій виконайте команди:

• SELECT * from Production.Location – читання з усіх полів таблиці Authors;

• SELECT *Name, Availability from Production.Location* – читання даних за таблиці *Production.Location* тільки із зазначених полів.

16. Для ролі *clerks* заборонена вибірка даних з таблиці *Production.WorkOrder* бази даних *AdventureWorks2008*. Користувач *Andy* належить до користувальницької ролі *clerks* і до системної ролі *db_datareader*. Чи може цей користувач отримати дані з цієї таблиці?

ЛАБОРАТОРНА РОБОТА №6 КЕРУВАННЯ БАЗАМИ ДАНИХ

Ціль: навчити використовувати SQL команди для створення баз даних і зміни їх структури, а також створювати резервні копії та відновлювати бази даних з резервних копій.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №7 «Структура баз даних у MS SQL Server», в якій розглянуто організація бази даних всередині СКБД MS SQL Server 2008, а також основні структурні елементи бази даних.

Вимоги до звіту: за результатами роботи надати звіт зі скріншотами, що містять SQL-команди і результати їх виконання для кожного завдання з розділу «Самостійна робота».

Завдання 1. Створіть базу даних Sales з журналом.

Вказівки до виконання:

1. Скористаймося функцією СREATE DATABASE:

```
CREATE DATABASE Sales
ON
( NAME = Sales_dat,
FILENAME = 'e:\saledat.mdf',
SIZE = 10,
MAXSIZE = 50,
FILEGROWTH = 5 )
LOG ON
( NAME = Sales_log,
FILENAME = 'e:\salelog.ldf',
SIZE = 5MB,
MAXSIZE = 25MB,
FILEGROWTH = 5MB )
```

2. Виконайте наступний код, щоб переконатися, що БД була створена з потрібними параметрами (рис. 6.1):

sp_helpdb Sales

Результат виконання продемонстровано на рис. 6.1.

	Results	6	Messag	es							
	name	l dbj	_size	owner		dbid	created	status			
1	Sales		15.00 N	4B TESTSYS-4	414BCA6\User	12	Sep 30 2009	Status=0N	ILINE, Updat	eability=REA	.D_WRITE
	•										
<											>
	name		fileid	filename	filegroup	size	maxsize	growth	usage		
1	Sales_	dat	1	e:\saledat.mdf	PRIMARY	10240 k	(B 51200 KB	5120 KB	data only		
2	Sales_	log	2	e:\salelog.ldf	NULL	5120 KI	3 25600 KB	5120 KB	log only		
-					• • •					I	
େତ୍	uery	TES	STSYS-4	14BCA6\SQL2008	(10 TEST	5YS-414	BCA6\User (52)	Adventure	Works2008	00:00:00	3 rows

Рис. 6.1. Відомості про базу даних

Завдання 2. Створіть резервну копію бази даних Sales.

Вказівки до виконання:

1. Створімо пристрій для резервування:

sp_addumpdevice 'disk', 'backupdisk', 'e:\sales.bak'

2. Скористаймося функцією ВАСКИР для виконання резервного копіювання:

BACKUP DATABASE Sales TO backupdisk

Результат виконання показано на рис. 6.2.

Messages Processed 184 pages for database 'Sales', file 'Sales_dat' on file 1. Processed 3 pages for database 'Sales', file 'Sales_log' on file 1. BACKUP DATABASE successfully processed 187 pages in 0.334 seconds (4.352 MB/sec).

Рис. 6.2. Результат виконання оператора ВАСКИР

Самостійна робота

Вказівка. Для доведення правильності виконання дій можна використовувати процедури **sp_helpdb** до виконання деякої операції і після.

Завдання 1.

1. Створіть користувача, що має право створення і видалення баз даних.

2. Створіть власну базу даних, розмір основного файлу якої 5 Мб, розмір журналу – 2 Мб. Основний файл може збільшуватися до 10 Мб с кроком 20%. Обсяг файлу журналу збільшується до 5 Мб з кроком 1 Мб. Розташування файлів на диску – D: \ MSSQL \ DATA.

3. Переконайтеся, що база даних створена належним чином.

Завдання 2.

1. Перегляньте список параметрів бази даних, які можуть бути встановлені.

2. Подивіться список встановлених опцій створеної Вами бази даних.

3. Визначте використання бази даних тільки власником і в режимі підтримки одного користувача.

4. Переконайтеся в зміні параметрів бази даних.

Завдання 3.

1. Перейменуйте створену Вами базу даних.

Завдання 4.

1. Створіть власну файлову групу.

2. Для Вашої бази даних визначте ще один файл розміром 2 Мб і помістіть його в створену робочу групу.

3. Доведіть правильність виконаних Вами дій.

Завдання 5.

1. Для своєї бази даних встановіть можливість автоматичного стиснення даних.

Завдання 6.

1. Визначте відомості про дисковий простір, що займає база даних *AdventureWorks2008*.

2. Стисніть базу даних *AdventureWorks2008* так, щоб вона містила тільки 25% простору, доступного їй на поточний момент.

3. Доведіть правильність виконаної дії.

Завдання 7.

1. Для бази даних *Adventure Works2008* додайте ще один файл для групи PRIMARY.

2. Здійсніть стиснення бази даних *AdventureWorks2008*. Обґрунтуйте обрані Вами параметри.

3. Доведіть, що стиснення бази даних дійсно виконано.

Завдання 8.

1. Створіть резервну копію бази даних *AdventureWorks2008*. Фізичне ім'я пристрою резервного копіювання: *D:\MSSQL\BACKUP\AdventureWorks_backup.bak*.

2. Створіть резервну копію журналу транзакцій бази даних Adventure Works 2008.

Завдання 9.

- 1. Видаліть базу даних *AdventureWorks2008*.
- 2. Відновіть видалену базу з пристрою з фізичним ім'ям *AdventureWorks_backup.bak*.
- 3. Доведіть правильність виконаної дії.

ЛАБОРАТОРНА РОБОТА №7 ВИБІРКА ДАНИХ З ДЕКІЛЬКОХ ТАБЛИЦЬ

Ціль: навчити використовувати конструкцію JOIN для зовнішнього та внутрішнього сполучення двох або більше таблиць.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №4 «Вибірка даних», в якій вивчається один з основних операторів мови SQL – SELECT, і пояснюється призначення кожного з його розділів, а також розглядаються питання з'єднання декількох таблиць.

Вимоги до звіту: за результатами роботи представити набір SQL-скриптів, що вирішують завдання з розділу «Самостійна робота».

Завдання 1. Отримайте список товарів із зазначенням їх категорій.

Вказівки до виконання:

1. Оскільки в таблиці *Product* нормалізованою бази даних *AdventureWorks2008* зберігається тільки ідентифікатор категорії товару, то щоб отримати список товарів із зазначенням їх категорій, необхідно з'єднати таблиці *Product* і *ProductSubcategory*.

2. Перед зазначенням поля в операторі SELECT необхідно буде використати імена таблиць *Product* і *ProductSubcategory* для вирішення конфліктів, тому що в обох таблицях присутні поля *Name*.

SELECT Product.Name, ProductSubcategory.Name, Product.ListPrice FROM Production.Product INNER JOIN Production.ProductSubcategory ON Product.ProductSubcategoryID = ProductSubcategory.ProductSubcategoryID

F	Results	🛅 Messages							
	Name	9	Name	ListPrice					
1	HL R	load Frame - Black, 58	Road Frame	s 1431.50					
2	HL R	load Frame - Red, 58	Road Frame	s 1431.50					
3	Spor	t-100 Helmet, Red	Helmets	34.99					
4	Spor	t-100 Helmet, Black	Helmets	34.99					
5	Mour	ntain Bike Socks, M	Socks	9.50					
6	Mour	ntain Bike Socks, L	Socks	9.50					
7	Spor	t-100 Helmet, Blue	Helmets	34.99					
8	AWC	CLogo Cap	Caps	8.99					
9	Long	-Sleeve Logo Jersey, S	Jerseys	49.99					
10	Long	-Sleeve Logo Jersey, M	Jerseys	49.99					
11	Long	-Sleeve Logo Jersey, L	Jerseys	49.99					
🕗 Qu	Jer	TESTSYS-414BCA6\SQL2	008 (10 TI	ESTSYS-414BC	IA6\User (52)	AdventureWo	rks2008	00:00:00	

Результат виконання запиту показаний на рис. 7.1.

Рис. 7.1. Результат з'єднання двох таблиць – товари із зазначенням категорії

Зауваження. При такому з'єднанні товари, для яких не зазначено їх категорія, не включаються до набору результатів.

Завдання 2. Створіть запит, який повертає ім'я замовника і дату розміщення замовлення.

Вказівки до виконання: 1. Виконайте наступний код:

SELECT FirstName, LastName, OrderDate FROM Person.Person LEFT JOIN Sales.Customer ON Customer.PersonID = Person.BusinessEntityID LEFT JOIN Sales.SalesOrderHeader ON Customer.CustomerID = SalesOrderHeader.CustomerID

Результат виконання запиту показаний на рис. 7.2.

📰 F	Results 🛅	Messages						
	FirstName	LastName	OrderDate					
1	Syed	Abbas	NULL					
2	Catherine	Abel	2003-09-01-00	:00:00.000				
3	Catherine	Abel	2003-12-01-00	:00:00.000				
4	Catherine	Abel	2004-03-01-00	:00:00.000				
5	Catherine	Abel	2004-06-01-00	:00:00.000				
6	Kim	Abercrombie	2001-09-01-00	:00:00.000				
7	Kim	Abercrombie	2001-12-01-00	:00:00.000				
8	Kim	Abercrombie	2002-03-01-00	:00:00.000				
9	Kim	Abercrombie	2002-06-01-00	:00:00.000				
10	Kim	Abercrombie	2002-09-01-00	:00:00.000				
11	Kim	Abercrombie	2003-09-01-00	:00:00.000				
0 Qu	I TESTSYS	-414BCA6\SQL2	2008 (10 TE	STSYS-414BC	:A6\User (52)	AdventureWorks2008	00:00:00	32318

Рис. 7.2. Результат з'єднання трьох таблиць – клієнти і дати розміщення ними замовлень

Для осіб, які не розміщували замовлення, в полі OrderDate міститься значення NULL.

За *правого з'єднання* (ключове слово **RIGHT JOIN**) у результат включаються всі записи правої таблиці, незалежно від того, є для них відповідний рядок в лівій таблиці чи ні.

Змініть розглянутий запит так, щоб він видавав такі ж результати при використанні правого з'єднання.

Завдання 3. За допомогою перехресного з'єднання можна перерахувати всі можливі способи поставки товарів в базі даних *Northwind* (поставлялася з MS SQL Server до виникнення *AdventureWorks2008*).

Вказівки до виконання:

SELECT DISTINCT Suppliers.Country, Orders.ShipCountry FROM Suppliers CROSS JOIN Orders

Самостійна робота

1. Вивести прізвище та ім'я замовників з подання vIndividualCustomer, номер (SalesOrderNumber) і дату розміщення ними замовлень, починаючи з 1 січня 2003 року, з таблиці SalesOrderHeader. Якщо компанія розміщує кілька замовлень в один день, то повинен бути повернутий лише один запис.

Вказівка: для обмеження числа записів в підсумковому наборі даних, необхідно вказувати умову в розділі **WHERE**.

2. Отримати список товарів із зазначенням номеру замовлення і замовника, замовлених в певний день, наприклад, 1 січня 1998 року. Розв'язок оформіть у вигляді збереженої процедури.

3. Сконструюйте запит, який повертає номер замовлення, прізвище та ім'я замовника, а також прізвище співробітника, що обробив це замовлення.

4. Створіть запит про замовлення кожного товару із зазначенням його найменування, ціни та кількості за допомогою внутрішнього і зовнішнього з'єднань.

5. У чому полягає відмінність у використанні внутрішнього і зовнішнього з'єднань при вирішенні задачі №4?

6. Запропонуйте варіант перехресного з'єднання для БД *AdventureWorks2008* і реалізуйте його на SQL.

7. Отримати набір даних, що містить наступну інформацію: прізвища та імена замовників, їх адреси; назва компаній-виробників, із зазначенням їх адрес; прізвища, імена та адреси співробітників.

Вказівка: прізвище та ім'я повинні бути виведені одним полем. Вибірки даних здійснювати не з таблиць, а з наступних подань: *vEmployee*, *vIndividualCustomer*, *vVendor*.

ЛАБОРАТОРНА РОБОТА №8 АНАЛІТИЧНА ВИБІРКА ДАНИХ

Ціль: навчити використовувати агрегатні функції і розділ GROUP BY оператора SELECT для аналітичної вибірки даних.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №4 «Вибірка даних», в якій вивчається оператор SELECT і пояснюється призначення кожного з його розділів, а також розглядаються питання угруповання даних.

Вимоги до звіту: за результатами роботи представити набір SQL-скриптів, що вирішують завдання з розділу «Самостійна робота».

Завдання 1. Обчислити середню ціну товарів.

Вказівки до виконання:

1. Для обчислення середньої ціни необхідно скористатися агрегатної функцією AVG.

2. Агрегатна функція об'єднає значення поля *ListPrice*, так що наступний код поверне тільки одне число (рис. 8.1):

SELECT AVG(ListPrice) FROM Production.Product



Рис. 8.1. Середня ціна товарів

Завдання 2. Визначити імена замовників, які розмістили замовлення в останній обліковий день.

Вказівки до виконання:

1. Для простоти спочатку отримаємо список тільки CustomerID, а не імен замовників. Для цього можна скористатися наступним запитом:

Select CustomerID From Sales.SalesOrderHeader Where OrderDate=(SELECT Max(OrderDate) From Sales. SalesOrderHeader)

Результат виконання запиту показаний на рис. 8.2.

	Results 🛅 Messages
	CustomerID
1	11078
2	11927
3	28789
4	11794
5	14680
6	19585
7	27686
8	20601
9	26564
10	16170
11	16927

Рис. 8.2. ID клієнтів, які розмістили замовлення в останній обліковий день

Зауваження. Виконання цього запиту здійснюється в два етапи: на першому – виконується підзапит, як самостійний запит, який повертає значення, яке використовується на другому етапі при виконанні зовнішнього запиту.

2. Для повного вирішення задачі виберемо імена всіх замовників з таблиці *Person*, які відображаються у списку, отриманому в попередньому пункті. Однак таблиця *Person* не містить поля *CustomerID*, тому доведеться ще з'єднати її з таблицею *Customer*. Результуючий код буде виглядати наступним чином:

SELECT FirstName + ' ' + LastName AS 'CustomerName' FROM Person.Person INNER JOIN Sales.Customer ON Customer.PersonID = Person.BusinessEntityID WHERE CustomerID in (SELECT CustomerID FROM Sales.SalesOrderHeader WHERE OrderDate=(SELECT Max(OrderDate) FROM Sales. SalesOrderHeader))

Результат виконання запиту показаний на рис. 8.3.

🔲 F	lesults 🛅 Messages
	CustomerName
1	Alisha Raji
2	Chloe Harris
3	Kristi Fernandez
4	Tamara Lu
5	Devin Phillips
6	Rebekah Gonzalez
7	Caleb Lal
8	Marvin Munoz
9	Jacqueline Murphy
10	Chloe Martin
11	Christina Morris

Рис. 8.3. Імена клієнтів, які розмістили замовлення в останній обліковий день

Зауваження. Такий запит може бути оформлений і у вигляді з'єднання таблиць. Такі запити обробляються значно швидше, тому якщо виконання запиту не потрібно виконувати в кілька етапів, використовувати підзапитів не обов'язково.

Завдання 3. Визначити список замовників, які замовили за один раз більше 300 одиниць товару.

Вказівки до виконання:

1. Для кожного замовлення нам потрібно отримати загальну кількість товару, для цього буде потрібно скласти пов'язаний підзапит.

2. Для того щоб перед іменами полів не писати кожен раз повну назву таблиць, зручно скористатися псевдонімами. В даному випадку визначимо такі псевдоніми: для таблиці SalesOrderHeader псевдонім oh, а для SalesOrderDetail - od.

3. Результуючий код буде виглядати наступним чином:

SELECT SalesOrderID, CustomerID FROM Sales.SalesOrderHeader oh WHERE 300<(SELECT sum(OrderQty) FROM Sales.SalesOrderDetail od WHERE od.SalesOrderID=oh.SalesOrderID)

Завдання 4. Для кожного товару відобразити відомості про найбільше його замовлення: кількість замовленого товару і номер замовлення.

Вказівки до виконання:

1. Для виконання завдання необхідно використовувати внутрішнє з'єднання таблиць *Product* і *SalesOrderDetail*, а також пов'язаний підзапит, який повертає для кожного товару його максимальне замовлення.

2. У запиті нам буде потрібна робота з таблицею *SalesOrderDetail* як в зовнішньому запиті, так і у внутрішньому пов'язаному підзапиті. Тому для даної таблиці ми створимо два псевдоніми: od1 i od2.

3. Результуючий код буде виглядати наступним чином:

SELECT Name, OrderQty, SalesOrderID FROM Production.Product p INNER JOIN Sales.SalesOrderDetail od1 ON od1.ProductID=p.ProductID WHERE OrderQty = (SELECT Max(OrderQty) FROM Sales.SalesOrderDetail od2 WHERE od1.ProductID=od2.ProductID)

Завдання 5. Визначити кількість різних товарів в кожному замовленні.

Вказівки до виконання:

1. Необхідно згрупувати записи з однаковим ідентифікаційним номером замовлення *SalesOrderID* і підрахувати кількість записів в кожній групі по полю *ProductID*.

Для цього можна виконати такий запит:

SELECT SalesOrderID, count(ProductID) AS [Кількість] FROM Sales.SalesOrderDetail GROUP BY SalesOrderID

Зауваження. При використанні **GROUP BY** для кожної визначеної групи значень виводиться тільки один запис в підсумковому наборі даних.

2. Якщо для кожного замовлення ще необхідно підрахувати і загальна кількість товарів у замовленні, то запит повинен бути доповнений:

SELECT SalesOrderID, count(ProductID) AS [Кількість], sum(OrderQty) as [Загальна кількість] FROM Sales.SalesOrderDetail GROUP BY SalesOrderID

Зауваження. При групуванні записів допускається також використання розділу **WHERE**, в цьому випадку групуються записи, що задовольняють цій умові.

Самостійна робота

1. Визначити кількість співробітників компанії в базі даних Adventure Works 2008.

2. Підрахувати максимальну вартість доставки замовлень.

3. Вивести список товарів, ціна яких вища за середню ціну.

4. Вирішити це завдання з використанням об'єднання таблиць.

5. Для кожного товару відобразити відомості про найбільше його замовлення: найменування, кількість і номер замовлення.

Вказівка: використайте групування за найменуванням товару та номером замовлення.

6. Вивести список товарів, на які надійшло понад 50 замовлень.

7. Отримати список тільки тих товарів, чия сумарна кількість на замовлення перевищує 500 штук.

8. Визначити 10 товарів, найбільш популярних у замовників. Розв'язок завдання оформити у вигляді збереженої процедури.

9. Визначити загальну кількість і сумарну вартість всіх замовлень для кожного замовника.

10. Обчислити сумарну вартість кожного замовлення з урахуванням знижки, вказавши при цьому найменування компанії, яка виконала його.

11. З'ясувати, товари якого типу користуються у замовників найбільшим попитом.

ЛАБОРАТОРНА РОБОТА №9 ВИКОРИСТАННЯ ПОДАНЬ

Ціль: навчити створювати і використовувати подання при вибірці даних.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №4 «Вибірка даних», в якій вивчається один з основних операторів SELECT, пояснюється призначення кожного з його розділів, а також розглядаються питання групування даних, і лекцію №5 «Допоміжні об'єкти бази даних», що дає основні відомості про подання.

Вимоги до звіту: за результатами роботи представити набір SQL-скриптів, що вирішують завдання з розділу «Самостійна робота».

Завдання 1. Створити подання, що містить лише таку інформацію про співробітників компанії AdventureWorks2008, як: ID співробітника, його посаду та логін (одним полем), дату народження.

Вказівки до виконання:

1. Створіть подання:

CREATE VIEW InfoEmployees ([Номер], [Фамилия], [Дата рождения]) AS

SELECT BusinessEntityID, JobTitle + '(' + LoginID + ')', CONVERT (char(10), BirthDate, 104)

FROM HumanResources.Employee

2. Для перегляду вмісту проекції виконується наступний запит (результат показаний на рис. 9.1):

	Results	🛅 Messages				
	Номе	р Фамилия	🛛 Дата рождения			^
1	1	Chief Executive Officer(adventure-works\ken0)	02.03.1959			-
2	2	Vice President of Engineering(adventure-works\terri0)	01.09.1961			
3	3	Engineering Manager(adventure-works\roberto0)	13.12.1964			
4	4	Senior Tool Designer(adventure-works\rob0)	23.01.1965			
5	5	Design Engineer(adventure-works\gail0)	29.10.1942			
6	6	Design Engineer(adventure-works\jossef0)	11.04.1949			
7	7	Research and Development Manager(adventure-works)	27.03.1977			
8	8	Research and Development Engineer(adventure-works\	06.07.1976			
9	9	Research and Development Engineer(adventure-works\	21.02.1969			
10	10	Research and Development Manager(adventure-works\	01.01.1975			
11	11	Senior Tool Designer(adventure-works\ovidiu0)	18.02.1968			~
0 Q	uer	TESTSYS-414BCA6\SQL2008 (10 TESTSYS-414BCA6\User	(52) AdventureW	/orks2008	00:00:00	290 rows

SELECT * FROM InfoEmployees

Рис. 9.1. Результат вибірки з подання InfoEmployees

Зауваження. За допомогою даного подання обмежений доступ до деяких полів вихідної таблиці *Employee*, в цьому випадку говорять, що на таблицю накладений вертикальний фільтр, тобто обмежений доступ до частини полів таблиці без захисту на рівні стовпців.

Завдання 2. Создайте представление, которое обеспечивает доступ к информации только о производителях, имеющих онлайновые службы для заказа товара.

1. За аналогією з попереднім завданням необхідно створити вертикальний фільтр, тобто вибрати тільки одне поле *Name* з таблиці *Vendor*.

2. Для відсікання «зайвих» виробників, тих, для яких невідома адреса онлайнової служби замовлення, скористаємося розділом **WHERE**. В цьому випадку ми створимо ще і горизонтальний фільтр.

3. Код для створення подання буде наступним:

CREATE VIEW OnlineVendors AS SELECT [Name] FROM Purchasing.Vendor WHERE PurchasingWebServiceURL IS NOT NULL

Завдання 3. Створити подання, що відображає сумарну вартість кожного замовлення із зазначенням замовника і його номера.

Вказівки до виконання:

1. Для отримання всієї необхідної інформації нам буде потрібно з'єднати таблиці *Customer*, *Person*, *SalesOrderHeader* і *SalesOrderDetail*.

2. Для того щоб отримати сумарну вартість замовлення, необхідно згрупувати дані по полю *SalesOrderID*. Однак нам потрібно додати в розділ **GROUP BY** ще й поля, що формують ім'я замовника, тому що ми їх повинні будемо вказати в **SELECT**.

3. Для визначення вартості одного замовлення складемо наступну формулу: UnitPrice * OrderQty * (1 – UnitPriceDiscount).

4. У результаті код створення подання буде наступним:

CREATE VIEW InfoOrders AS

SELECT FirstName + ' ' + LastName as [Назва компанії], SalesOrderHeader.SalesOrderID as [Номер замовлення], Convert (money, SUM(UnitPrice*OrderQty*(1-UnitPriceDiscount)),0) AS [Підсумок] FROM (Person.Person INNER JOIN Sales.Customer ON Person.BusinessEntityID = Customer.PersonID INNER JOIN Sales.SalesOrderHeader ON Customer.CustomerID=SalesOrderHeader.CustomerID) INNER JOIN Sales.SalesOrderDetail ON SalesOrderHeader.SalesOrderID=SalesOrderDetail.SalesOrderID GROUP BY SalesOrderHeader.SalesOrderID, FirstName + ' ' + LastName

Самостійна робота

1. Визначте 5 замовників, які є лідерами закупівель (оформіть результат у вигляді збереженої процедури).

Вказівка: Використайте дані подання InfoOrders.

2. Виведіть статистику роботи компанії *AdventureWorks2008*, що відбиває обсяги продажів і кількість замовлень за місяцями.

Вказівка: Створіть подання, яке включає номер замовлення, рік і місяць його розміщення, підсумкову вартість кожного замовлення з урахуванням знижки. Щоб отримати рік і місяць розміщення замовлення, скористайтеся функцією перетворення **convert (char(7),OrderDate,102)**, яка перетворює дату до рядка типу уууу.mm.dd. Так як при перетворенні визначена рядок довжиною 7 символів, тому результатом функції буде рядок типу уууу.mm.

3. Створити подання *InfoSalesPerson*, ілюструє статистику роботи продавців компанії *AdventureWorks2008*: у подання потрібно включити підсумкову суму отриманих замовлень та їх кількість.

4. Визначити кількість розміщених замовлень за порами року.

ЛАБОРАТОРНА РОБОТА №10 ПРОЄКТУВАННЯ БАЗИ ДАНИХ

Ціль: продемонструвати процес проєктування бази даних і навчити використовувати SQL-команди для створення таблиць і управління обмеженнями.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №10 «Перші нормальні форми» і лекцію №11 «Четверта і п'ята нормальні форми», в яких розглядається необхідність і сам процес нормалізації, що дозволяє підтримувати цілісність бази даних.

Вимоги до звіту: за результатами роботи подати скріншот з фізичною моделлю даних, а також скріншоти з SQL-командами і результатами їх виконання для завдань №2 і №3 з розділу «Самостійна робота».

Етапи проектування бази даних

З точки зору корпорації Microsoft проєктування бази даних складається з двох фаз: логічного і фізичного моделювання. Під час логічного проєктування розробляється модель бази даних, яка не залежить від конкретної СУБД. У процесі ж фізичного проєктування створюється модель бази даних, оптимізована для реальної СУБД.

Процес проєктування бази даних складається з наступних етапів:

- 1. Збір інформації.
- 2. Ідентифікація об'єктів і їх моделювання.
- 3. Визначення типів інформації для кожного об'єкта.
- 4. Ідентифікація відношень.
- 5. Нормалізація.
- 6. Створення бази даних.

Перші п'ять етапів відносяться до фази логічного моделювання, а шостий етап – до фізичного моделювання.

Логічна модель

Розробимо логічну структуру бази даних, в якій зберігається інформація про викладачів і дисциплінах, що ними читаються, відомості про студентів і відвідуваних заняттях.

Встановлено, що:

- студент може відвідувати будь-яку кількість дисциплін;
- викладачі можуть вести кілька дисциплін;
- дисципліна читається в деякій аудиторії;

- в одній аудиторії читається тільки одна дисципліна.

Таким чином, виділяються наступні об'єкти: Викладачі, Студенти, Дисципліни й Аудиторії. Між об'єктами існують такі зв'язки: Студенти-Дисципліни (множина-домножини, реалізується через додатковий об'єкт Реєстрація), Викладачі-Дисципліни (одиндо-множини), Аудиторії-Дисципліни (один-до-одного).

Кожен об'єкт повинен містити наступну інформацію.

Викладачі:

- прізвище, ім'я, по батькові – обов'язковий атрибут;

– дата народження – обов'язковий атрибут, не може бути більше дати сьогоднішньої дати і менше 1900 року;

– домашня адреса – обов'язковий атрибут; якщо не вказана, то за замовчуванням встановлюється значення *'unknown'*;

- телефон – обов'язковий атрибут, або задається в форматі 00-00-00, або при його відсутності за замовчуванням вказується '*no*'.

– дата найму – обов'язковий атрибут, не може бути більше дати сьогоднішнього дня і менше дати народження, за замовчуванням встановлюється сьогоднішня дата;

- стаж – обчислюване поле, визначається різницею між сьогоднішньою датою і датою найму.

Студенти:

- прізвище, ім'я, по батькові – обов'язковий атрибут;

– дата народження – обов'язковий атрибут, не може бути більше дати сьогоднішньої дати і менше 1900 року;

Дисципліни:

- найменування – обов'язковий атрибут;

– прізвище викладача, який читає дисципліну, список студентів, що відвідують дисципліну, аудиторія – зовнішній ключ.

Аудиторії:

- номер обов'язковий атрибут;
- тип аудиторії обов'язковий атрибут.

Фізична модель

При переході від логічної моделі до фізичної необхідно забезпечити *цілісність даних*, під якою розуміється узгодженість і коректність інформації в базі даних.

Існує 4 групи правил цілісності:

- *Цілісність області значень (або полів):* визначає набір допустимих для поля значень, в тому числі і допустимість значень NULL.

– *Цілісність сутностей:* вимагає наявності у кожного запису таблиці унікального ідентифікатора – значення первинного ключа.

– Посилальна цілісність: гарантує підтримку постійного зв'язку між первинним ключем і зовнішнім ключем, тобто забороняє видалення запису і зміну значення її первинного ключа, якщо на нього посилається зовнішній ключ, а також забороняє додавання запису зі значенням зовнішнього ключа, що не відповідає жодному значенню первинного ключа.

– *Цілісність, що визначається користувачем,* вказує специфічні правила користувача, які не можна віднести до якої-небудь групи.

Цілісність даних може бути забезпечена двома способами.

1. Декларативний спосіб. Критерії, яким повинні задовольняти дані, задаються при визначенні об'єкта і є частиною визначення бази даних.

Переваги: контроль цілісності виконується автоматично MS SQL Server, такий спосіб повністю сумісний зі стандартом ISO SQL:2008².

2. **Процедурний спосіб**. Критерії описуються в пакетах операторів, виконання яких і визначає цілісність даних.

Переваги: реалізується як на клієнті, так і на сервері за допомогою різних програмних засобів.

Декларативна цілісність

Реалізується за допомогою обмежень, які є найбільш рекомендованими для забезпечення цілісності даних. Кожен тип цілісності забезпечується відповідним обмеженням:

• *PRIMARY KEY*. Визначення поля або групи полів в якості первинного ключа дозволяє унікально ідентифікувати кожен запис таблиці, тому що в цьому випадку неприпустимі повторювані і невизначені (NULL) значення.

• FOREIGN KEY. Дозволяє встановлювати зв'язок між полями, що містять ідентичні дані. Дані в цьому полі можуть набувати значень, визначені у відповідному первинному ключі, або значення NULL.

• UNIQUE. Визначає унікальність даних в деякому полі таблиці, причому це обмеження допускає значення NULL. В одній таблиці дозволено застосування кілька таких обмежень, поля з таким обмеженням можуть бути використані в якості зовнішніх ключів.

• *DEFAULT*. Вказує значення поля за замовчуванням, якщо значення явно не вказано при вставці даних. Для поля можна застосувати тільки одне таке обмеження.

² Стандарт ISO SQL:2008 є шостою версією мови запитів SQL, яка включає в себе розширення попереднього стандарту SQL:2003.

• *CHECK*. Накладає обмеження на значення поля у вигляді логічного виразу, що дозволяє визначити діапазон допустимих значень поля. Причому це обмеження дозволяє посилатися на значення інших полів.

Для отримання інформації про використовувані обмеження зазначеної таблиці призначена спеціальна збережена процедура:

sp_helpconstraint Ім'яТаблиці

Створення таблиці

Наприклад, для створення таблиці, що зберігає відомості про студентів, необхідно виконати команду:

CREATE TABLE Students (StudentID int IDENTITY(1,1) CONSTRAINT prkStudentID PRIMARY KEY, LastName char(20) NULL, FirstName char(20) NULL, BirthDate datetime NULL CONSTRAINT bdChek CHECK (BirthDate < GetDate()))

Щоб отримати інформацію про властивості таблиці, необхідно виконати спеціальну збережену процедуру: **sp_help** *Iм'яТаблиці*.

Проекції інформаційної схеми

Системні відомості про базу даних та всі її об'єкти називаються метаданими, для отримання яких MS SQL Server надає безліч різних способів: виклик системних збережених процедур та подання інформаційної схеми, а також використання утиліти адміністрування SQL Management Studio.

Подання інформаційної схеми забезпечують зовнішнє подання метаданих, незалежне від системних таблиць. Ці подання належать заздалегідь певному користувачеві *information_schema*. В кожному поданні містяться метадані об'єктів деякої бази даних.

Наприклад, такі подання повертають метадані про діючі обмеження бази даних:

- *table_constraints* список всіх обмежень бази даних.
- *check_constraints* список обмежень типу СНЕСК.
- referential_constraints список обмежень посилальної цілісності.

Самостійна робота

1. На основі описаної логічної моделі бази даних скласти її фізичну модель.

2. Визначити метадані обмежень PRIMARY KEY в створеній базі даних, вказавши імена обмежень і таблиць, до яких вони застосовані.

3. Вивести логічні вирази обмежень СНЕСК.

ЛАБОРАТОРНА РОБОТА №11 КЕРУВАННЯ ДАНИМИ

Ціль: навчити використовувати оператори SQL для вставки, зміни та видалення даних з таблиць.

Теоретичний матеріал: перед виконанням лабораторної роботи рекомендується вивчити лекцію №4 «Вибірка даних», з якої потрібні знання оператора SELECT, необхідного для вставки даних в одну таблицю з іншої, а також лекцію №9 «Оператори реляційної алгебри», в якій розглянуто теоретичні аспекти основних команд з управління даними.

Вимоги до звіту: за результатами роботи представити набір SQL-скриптів, що вирішують завдання з розділу «Самостійна робота», а також текстовий файл, необхідний для завдання №2 з того ж розділу.

Завдання 1. У таблицю Lect додайте запис із заданими значеннями полів, значеннями за замовчуванням.

Вказівки до виконання:

1. Задаймо значення трьох полів: Brdate, FirstName, LastName.

2. Поле лічильника ми не будемо задавати, тому що воно буде згенеровано автоматично.

3. Використовуйте наступний код для вставки нового запису:

INSERT INTO Lect (Brdate, FirstName, LastName) VALUES (Convert(datetime, '01.01.1970',104), '...', '...')

Зауваження. Виконання команди **INSERT INTO Lect DEFAULT VALUES** призведе до помилки, тому що не для всіх полів вказані значення за замовчуванням.

Завдання 2. Додайте в таблицю *Lect* дані з таблиці *Employee* бази даних *AdventureWorks2008*:

Вказівки до виконання:

- 1. Заповнюватимемо 4 поля в таблиці Lect: FirstName, LastName, Brdate, HireDate.
- 2. Для вставки відразу декількох записів скористаємося оператором SELECT:

INSERT INTO Lect (FirstName, LastName, Brdate, HireDate) SELECT FirstName, LastName, BirthDate, HireDate FROM AdventureWorks2008.HumanResources.Employee INNER JOIN AdventureWorks2008.Person.Contact ON Contact.ContactID = Employee.ContactID Зауваження. При використанні команди **SELECT** в таблицю вставляється кілька записів, що є результатом вибірки даних інших таблиць. Однак якщо хоча б однин із записів, що вставляється, порушує обмеження цілісності таблиці, то вся команда **INSERT** скасовується.

Завдання 3. Додайте в таблицю Students дані з файлу.

Вказівки до виконання:

1. Створіть в корені диску С: текстовий файл students.txt наступного змісту:

1	Ivan	Petrov	1.1.1990
2	Petro	Ivanov	12.10.1992
3	Sergiy	Kozakov	5.3.1991

Зауваження. Як роздільник в рядках використовується ТАВ, а не пробіли.

2. Виконайте наступний код:

BULK INSERT Students FROM 'c:\students.txt'

Зауваження. Зверніть увагу: незважаючи на те, що ми вказали поля StudentID в файлі students.txt, вони були проігноровані при вставці. Це пояснюється тим, що ми не використали опцію **KEEPIDENTITY**.

Завдання 4. Змініть номер телефону у викладача на ім'я King.

Вказівки до виконання:

- 1. Для пошуку потрібного імені скористаймося оператором LIKE.
- 2. Встановимо для всіх викладачів 'King' телефон рівний 35-35-35:

UPDATE Lect SET Phone = '35-35-35' WHERE FirstName like 'King'

Завдання 5. Вказати, що курси з інформатики повинен читати викладач на ім'я *King*.

Вказівки до виконання:

1. Для поновлення записів в таблиці *Course* нам буде потрібно використання підзапиту, що повертає потрібного викладача: SELECT LectID From Lect Where (FirstName like 'King').

2. При цьому необхідно враховувати, що підзапит повинен повертати єдине значення для кожного запису.

3. Код для оновлення викладача з інформатики буде виглядати наступним чином:

UPDATE Course SET LectID = (SELECT LectID From Lect Where (FirstName like 'King')) WHERE CourseName like '%[Ii]nform%'

Завдання 6. Видаліть усіх студентів із прізвищем Ivanov.

Вказівки до виконання:

- 1. Для видалення записів по заданій умові необхідно використовувати розділ **WHERE** в операторі **DELETE.**
- 2. Виконайте наступний код для видалення студентів:

DELETE Students WHERE LastName = 'Ivanov'

Зауваження. Видалення даних з таблиці здійснюється порядково, при цьому можна виконати видалення як одного запису, так і декількох, які відповідають деякій умові. Однак для видалення всіх записів деякої таблиці рекомендується команда: **TRUNCATE TABLE Ім'яТаблиці**. При цьому видаляються всі записи, однак структура таблиці зберігається, як і пов'язані з нею об'єкти. Дана команда виконується швидше команди **DELETE**, тому що сервером реєструється тільки звільнення сторінок пам'яті, а не видалення кожного запису зі сторінки.

Самостійна робота

- 1. Заповніть таблиці Lect і Class створеної бази даних.
- 2. Заповніть таблицю Students створеної бази даних з текстового файлу.
- 3. Заповніть всі інші таблиці створеної бази даних.
- 4. Видаліть всіх студентів, які народилися до 1990 року.

Рекомендована література

1. Дейт К. Дж. Введение в системы баз данных. – М.: Вильямс, 2008.

2. Дибетта П. Знакомство с Microsoft SQL Server 2005. – М.: Русская редакция, 2005.

3. Lobel L., Brust A. J., Forte S. Programming Microsoft SQL Server 2008. – Microsoft Press, 2008.

4. Уолтерс Р. Э., Коулс М., Рей Р., Феррачати Ф., Фармер Д. SQL Server 2008: ускоренный курс для профессионалов. – М.: Издательский дом «Вильямс», 2008.

5. Волоха А. В. Microsoft SQL Server 2005. Новые возможности. – СПб.: Питер, 2006.

6. Каленик А. И. Использование новых возможностей Microsoft SQL Server 2005. – М.: Русская редакция; СПб.: Питер, 2006.

7. Вьейра Р. SQL Server 2000 Программирование. Ч. 1. – М.: Изд-во «БИНОМ. Лаборатория знаний», 2004.

8. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – М.: Издательский дом «Вильямс», 2003.

9. Дэвидсон Л. Проектирование баз данных на SQL Server 2000. – М.: Издво «БИНОМ. Лаборатория знаний», 2003.

10. Жилинский А. Самоучитель Microsoft SQL Server 2008. – СПб.: БХВ-Петербург, 2009.