



Урок 4

План заняття:

1. Підзапити в MS SQL Server
2. Об'єднання запитів
3. Об'єднання таблиць. Стандарт SQL2. Види об'єднань
4. Оператори модифікації даних INSERT, DELETE, UPDATE та підзапити
5. Домашнє завдання

1. Підзапити в MS SQL Server

Стандартом SQL підтримуються вкладені запити (підзапити), тобто запити можна вкладати один в один, в результаті чого одні запити можуть управляти іншими. Запит, який містить вкладений запит, називають **зовнішнім запитом** (outer query). Вкладений, тобто **внутрішній запит** являє собою повноцінний SELECT вираз, результат виконання якого використовується в зовнішньому запиті. Розміщуватись вони можуть майже в довільному місці SQL виразу, наприклад, замість одного з імен в списку SELECT. Підзапити часто використовуються для спрощення читабельності складних запитів при вказанні умови в операторі WHERE.

Розглянемо все на прикладі. Для початку напишемо запит без використання підзапитів, який виводить всю інформацію про книги лише однієї тематики.

```
-- без використання підзапитів
SELECT book.*
FROM book.Books book, book.Themes theme
WHERE book.ID_THEME=theme.ID_THEME AND theme.NameTheme='Учебники';

-- з підзапитом
SELECT *
FROM book.Books
WHERE ID_THEME =
    (SELECT ID_THEME
     FROM book.Themes
     WHERE NameTheme='Учебники');
```

Результат буде однаковий в обох випадках:

	ID_BOOK	NameBook	ID_THEME	ID_AUTHOR	Price	DrawingOfBook	DateOfPublish	Pages
1	6	Самоучитель работы на персональном компьютере: ...	2	10	49,6441	10000	2000-02-23 00:00:00.000	640
2	7	Основы работы на ПК	2	9	56,8055	3000	2000-01-11 00:00:00.000	440
3	8	Толковый словарь компьютерных технологий	2	2	82,7405	2000	2001-10-25 00:00:00.000	720
4	9	Первые шаги пользователя ПК с дискетой	2	10	47,6183	5000	2000-06-26 00:00:00.000	200

Що ж відбувається при використанні підзапиту? Спочатку цілісно виконується підзапит, який розміщується в інструкції WHERE. Результатом його виконання буде ідентифікатор, - значення поля NameTheme рівне «Учебники».

	ID_THEME	NameTheme
1	2	Учебники
2	3	Аппаратные средства ПК
3	8	Другие книги
4	9	Защита и безопасность ПК
5	10	Компьютерные журналы
6	11	ОС до Windows 98
7	12	Windows NT

	ID_THEME
1	2

Отриманий результат роботи внутрішнього підзапиту, тобто первинний ключ, повертається в основний (зовнішній) запит і використовується при його виконанні. При цьому зовнішні ключі таблиці Books, які використовуються для зв'язку з таблицею Themes співставляються з отриманим первинним ключем. В результаті будуть обрані лише підручники.

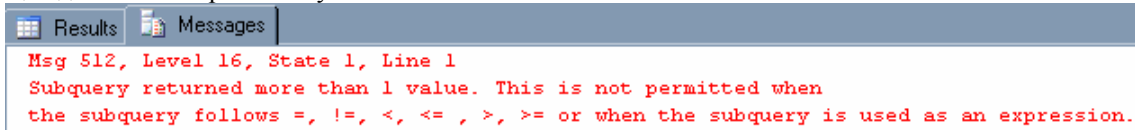
Обмеження при написанні підзапитів:

- 1) Вкладений запит завжди обмежується дужками.
- 2) Результатом підзапиту повинно бути лише одне значення і його тип даних повинен відповідати типу значення, яке використовується в зовнішньому запиті. Наприклад, вивести книги двох тематик: підручники та C/C++.



```
SELECT NameBook
FROM book.Books
WHERE ID_THEME =
      (SELECT ID_THEME
       FROM book.Themes
       WHERE NameTheme='Учебники' OR NameTheme='C & C++');
```

Вищеописаний запит може привести до помилки, якщо в базі даних існують обидві категорії або ж відпрацювати нормально, якщо однієї з категорій не існує.



Але цієї помилки можна уникнути шляхом заміни оператора порівняння на оператор IN, який застосовується для перебору значень результуючої множини підзапиту:

```
SELECT NameBook
FROM book.Books
WHERE ID_THEME IN
      (SELECT ID_THEME
       FROM book.Themes
       WHERE NameTheme='Учебники' OR NameTheme='C & C++');
```

3) Згідно стандарту ANSI SQL підзапити є **непереміщувані**, але в MS SQL Server, як і в MS Access дане правило не дійсне.

```
-- згідно ANSI SQL даний запит вірний
SELECT NameBook
FROM book.Books
WHERE ID_THEME =
      (SELECT ID_THEME
       FROM book.Themes
       WHERE NameTheme='Учебники' OR NameTheme='C & C++');
```

-- згідно ANSI SQL даний запит невірний, але в T-SQL він також являється
-- синтаксично коректним

```
SELECT NameBook
FROM book.Books
WHERE (SELECT ID_THEME
       FROM book.Themes
       WHERE NameTheme='Учебники' OR NameTheme='C & C++') = ID_THEME;
```

- 4) Результатом підзапиту може бути NULL-запис.
- 5) Вкладений запит не можна використовувати в інструкції ORDER BY.
- 6) Вкладений запит не може містити інструкції ORDER BY, COMPUTE, SELECT INTO.
- 7) При використанні підзапитів для перевірки результату не можна використовувати оператори BETWEEN, LIKE, IS NULL.
- 8) Якщо підзапит використовується з немодифікованим оператором порівняння, тобто звичаним знаком рівності (=), без використання ключових слів SOME, ANY або ALL, то він не може містити оператори групування GROUP BY та HAVING.
- 9) У списку вложеного оператора SELECT не можуть бути присутні поля типу text або image (крім форми з * в директиві EXISTS).
- 10) В підзапитах допускається використовувати функції агрегування, оскільки їх результатом є єдине значення.

Існують корельовані і некорельовані вкладені запити. **Некорельовані підзапити (noncorrelated subquery)** являються самими простими і означають, що сам підзапит не використовує стовпчики з таблиць зовнішнього запиту. Напишемо кілька прикладів некорельованих запитів:

1. Вивести список книг, які мають ціну вище середньої на 07/10/2006:

```
SELECT DISTINCT book.NameBook as 'Книга',
      convert(nchar(10), sale.Price)+' грн.' as 'Ціна'
FROM book.Books book, sale.Sales sale
```



```
WHERE book.ID_BOOK=sale.ID_BOOK
AND sale.Price > (SELECT avg(Price)
FROM sale.Sales
WHERE DateOfSale='2006.10.07');
```

Результат:

	Книга	DateOfSale	Цена
1	Windows NT 5 перспектива	1999-02-25 ...	110.00 грн.
2	Основы работы на ПК	2000-03-12 ...	25.00 грн.
3	Толковый словарь компьютерных технологий	2001-11-25 ...	40.00 грн.
4	Толковый словарь компьютерных технологий	2002-10-25 ...	45.00 грн.
5	Толковый словарь компьютерных технологий	2002-11-10 ...	45.00 грн.
6	Windows 2000 Professional. Руководство Питера Н...	2005-11-17 ...	90.00 грн.
7	Windows 2000 Professional. Руководство Питера Н...	2005-12-26 ...	92.00 грн.
8	Как программировать на С	2006-10-07 ...	160.00 грн.
9	С++ за 21 день	2006-10-07 ...	58.00 грн.
10	С++ за 21 день	2006-10-07 ...	156.00 грн.
11	JavaScript: сборник рецептов для профессионалов	2006-11-07 ...	110.00 грн.
12	JavaScript: сборник рецептов для профессионалов	2006-11-14 ...	110.00 грн.
13	Как программировать на С	2007-02-10 ...	165.00 грн.
14	JavaScript: сборник рецептов для профессионалов	2007-10-28 ...	150.00 грн.

	(No column name)
1	124.6666

підзапит

	Книга	Цена
1	JavaScript: сборник рецептов для профессионалов	150.00 грн.
2	Как программировать на С	160.00 грн.
3	Как программировать на С	165.00 грн.
4	С++ за 21 день	156.00 грн.

2. Визначити, які книги видавництва продаються в магазинах Росії:

```
SELECT NameBook as 'Книга', convert(nchar(10), Price)+' грн.' as 'Цена'
FROM book.Books
WHERE ID_BOOK IN ( SELECT DISTINCT ID_BOOK
FROM sale.Sales
WHERE ID_SHOP IN (SELECT sh.ID_SHOP
FROM sale.Shops sh, global.Country country
WHERE sh.ID_COUNTRY=country.ID_COUNTRY
AND country.NameCountry = 'Росія'))
ORDER BY 1;
```

3. Виведемо назву книги та її автора:

```
SELECT NameBook as 'Книга',
( SELECT FirstName+' '+LastName
FROM book.Authors a
WHERE a.ID_AUTHOR=b.ID_AUTHOR
) as 'Автор'
FROM book.Books b
```

Результат:

	Книга	Автор
1	Самучитель работы на персональном компьютере: 3...	Михаил Востриков
2	Основы работы на ПК	Андрей Ковязин
3	Толковый словарь компьютерных технологий	Johnson White
4	Первые шаги пользователя ПК с дискетой	Михаил Востриков
5	Реанимация, проверка, наладка современных ПК	NULL
6	Руководство для хакеров 2	NULL
7	Windows NT 5 перспектива	Михаил Миченный

автор невідомий

Корельований вкладений запит (correlated subquery) означає, що підзапит використовує один або кілька полів з зовнішнього запиту. При побудові цих запитів використовують один з 4-х наступних операторів: **EXISTS**, **ALL**, **ANY** та **SOME**. Найпоширенішим оператором є **EXISTS**.

Оператор **EXISTS** використовується, коли необхідно визначити наявність значень, які відповідають умові в підзапиті. Якщо дані на виході підзапиту існують, тоді даний оператор поверне значення true, інакше – false. Припустимо, необхідно знайти всіх авторів, які живуть в Україні. Зробити це можна як за допомогою корельованого, так і некорельованого запиту.

```
-- некорельований запит
SELECT *
FROM book.Authors
WHERE ID_COUNTRY IN ( SELECT ID_COUNTRY
FROM global.Country
WHERE NameCountry='Україна');
```



```
-- корельований запит
SELECT *
FROM book.Authors a
WHERE EXISTS ( SELECT *
               FROM global.Country c
               WHERE a.ID_COUNTRY=c.ID_COUNTRY AND NameCountry='Україна');
```

Результат:

Results			Messages	
	ID_COUNTRY	NameCountry		
1	1	Україна	→	
2	2	Росія		

Results					Messages	
	ID_AUTHOR	FirstName	LastName	ID_COUNTRY		
1	8	Тарас	Тимошок	1		
2	10	Михаил	Востриков	1		
3	16	Artur	Liliput	1		

У випадку некорельованого запису, зовнішній запит обирає дані з таблиці Authors, значення країни в яких відповідає ідентифікатору України (поверненого підзапитом). У випадку корельованого запиту, вкладений запит співставляє значення ID_COUNTRY (таблиці Country) з значенням аналогічного поля зовнішнього запиту (таблиці Authors). В результаті кожен запис таблиці Authors співставляється з результатом підзапиту і **У ВИПАДКУ ІСНУВАННЯ (WHERE EXISTS)** інформація про автора додається у результуючу множину.

Хоча ці два запити логічно еквівалентні, вони можуть виконуватись по-різному в залежності від даних і індексів. Якщо при побудові запиту ви не можете вирішити який тип запиту обрати (корельованій чи ні), тоді спробуйте обидва варіанти і порівняйте їх продуктивність.

Оператори **ALL**, **ANY** та **SOME** використовуються для порівняння одного значення з множиною даних, які повертаються підзапитом. Їх можна комбінувати з усіма операторами порівняння і вони можуть включати інструкції **GROUP BY** та **HAVING**. Оператори **ANY** (будь-який) та **SOME** (який-небудь) взагалі є ідентичними і в стандарті ISO вказується, що вони еквівалентні.

Для прикладу напишемо запит, який виводить інформацію про магазини, які реалізують книги видавництва:

```
SELECT NameShop
FROM sale.Shops
WHERE ID_SHOP = ANY ( SELECT ID_SHOP
                     FROM book.Books);
```

Оператор **ANY** співставляє всі значення поля ID_SHOP з таблиці Books та повертає результат true, якщо **БУДЬ_ЯКЕ** (ANY) значення співпадає з значенням поля ID_SHOP з таблиці Shops.

Оператор **ALL** повертає значення true, якщо кожне значення, яке буде отримане в результаті роботи підзапиту, відповідає умові зовнішнього запиту. Наприклад, виведемо лише тих авторів, чиї книги дорожчі, ніж книги авторів Англії.

```
SELECT FirstName+' '+LastName as 'Автор', b.Price as 'Ціна'
FROM book.Authors a, book.Books b
WHERE b.ID_AUTHOR=a.ID_AUTHOR AND
      b.Price > ALL ( SELECT b.Price
                    FROM book.Authors a, book.Books b, global.Country c
                    WHERE b.ID_AUTHOR=a.ID_AUTHOR AND a.ID_COUNTRY=c.ID_COUNTRY
                      AND c.NameCountry='Великобританія')
order by 2 desc;
```

Результат:

Diagram illustrating a SQL query workflow:

- Основні дані (Main Data):** A table with columns **Автор** (Author) and **Ціна** (Price). Rows 7-9 are highlighted.
- підзапит (Subquery):** A table with column **Price** containing values 251.0742, 253.9273, and 125.00. This is derived from the main data.
- результуюча вибірка (Resulting Selection):** A table with columns **Автор** and **Ціна** showing the final result of the query, including rows 1-6.

Таким чином, спочатку оператор **ALL** перевіряє значення цін на книги всіх авторів Англії. Після цього, він знаходить авторів з цінами на книги вищими, ніж у будь-якого з англійських авторів. Найбільша ціна книги англійського автора Річарда Веймаера – 253,93 грн. Отже, вибираються лише книги, які дорожчі 253,93 грн.



2. Об'єднання запитів

Об'єднання – це зв'язування даних, що містяться в двох таблицях, при цьому таблиці для об'єднання вказуються у виразі FROM разом з типом об'єднання.

За допомогою оператора SQL **UNION** можна об'єднати результати від 2 до 255 таблиць або результатів запитів в один результуючий набір. В результаті такого об'єднання однакові записи по замовчуванню знищуються, але при наявності ключового слова ALL повертаються всі записи, в тому числі і однакові. Синтаксис оператора UNION наступний:

```
SELECT <список_полів>
[FROM <список_таблиць>]
[WHERE <умова>]
[GROUP BY <список_полів_для_групування>]
[HAVING <умова_на_групу>]
{ UNION | INTERSECT | EXCEPT } [ALL]
SELECT <список_полів>
[FROM <список_таблиць>]
[WHERE <умова>]
[GROUP BY <список_полів_для_групування>]
[HAVING <умова_на_групу>];
```

При використанні оператора UNION притримуються наступних правил:

- кількість, послідовність полів і їх типи даних в обох списках полів повинні співпадати;
- якщо в одному з запитів використовується оператор INTO, то він повинен задаватись в першому запиті;
- оператори GROUP BY та HAVING використовуються лише в одному запиті;
- жодна з таблиць не може бути відсортована окремо. Можна сортувати лише результуючий запит. Тому оператор ORDER BY можна використовувати лише в кінці оператора UNION;
- оператори COMPUTE або COMPUTE BY також повинні використовуватись лише в кінці оператора UNION для того, щоб обрахувати сумарні значення всього запиту;
- імена полів результату визначаються списком полів першого оператора SELECT;
- оператори UNION, EXCEPT і INTERSECT неможна використовувати разом з інструкцією INSERT.

Для початку виберемо всі книги, ціна яких більше 100 грн., **АБО** код тематики яких = 2.

```
SELECT NameBook
FROM book.Books
WHERE Price > 100
UNION
SELECT NameBook
FROM book.Books
WHERE id_theme = 2;
```

Результат:

	NameBook	Price	id_theme
1	Самоучитель работы на персональном компьютере: ...	17,40	2
2	Основы работы на ПК	19,91	2
3	Толковый словарь компьютерных технологий	29,00	2
4	Первые шаги пользователя ПК с дискетой	16,69	2
5	Реанимация, проверка, наладка современных ПК	17,69	3
6	Руководство для хакеров 2	11,37	8
7	Windows NT 5 перспектива	100...	12
8	Сетевые технологии Windows 2000 для профессиона...	210...	14
9	Windows 2000 Professional: Руководство пользователя	99,00	14

Results		Messages
NameBook		
1	Visual C++ и MFC. 2е издание	
2	Windows NT 5 перспектива	
3	Как программировать на С	
4	Основы работы на ПК	
5	Первые шаги пользователя ПК с дискетой	
6	Путь к LINUX. 2е изд.	

Выведем на экран все книги тиражом 5000 экземпляров **АБО** список американских авторов.

```
SELECT NameBook
FROM book.Books
WHERE DrawingOfBook=5000
UNION
SELECT FirstName+' '+LastName as 'Автор'
FROM book.Authors a, global.Country c
WHERE a.ID_COUNTRY=c.ID_COUNTRY AND c.NameCountry='США';
```



Результат:

The diagram illustrates a data transformation process. It shows two source tables on the left and a single target table on the right, connected by two red arrows indicating the flow of data.

Source Table 1 (Left):

	NameBook
1	Первые шаги пользователя ПК с дискетой
2	Windows NT 5 перспектива
3	Сетевые технологии Windows 2000 для профессионалов

Source Table 2 (Right):

	Автор
1	Johnson White
2	Marjorie Green
3	Livia Karsen
4	Джес Либерти

Target Table (Bottom):

	NameBook
1	Johnson White
2	Livia Karsen
3	Marjorie Green
4	Windows NT 5 перспектива
5	Джес Либерти
6	Первые шаги пользователя ПК с дискетой
7	Сетевые технологии Windows 2000 для профессионалов

The red arrows indicate that the data from the first source table is mapped to rows 6, 7, and 8 of the target table, and the data from the second source table is mapped to rows 1, 2, 3, and 4 of the target table.

А тепер виберемо всі книги з вказанням дати та ціни їх продажу кожним магазином, **АБО** назви книг, які продавались взагалі.

```
select 'Назва книги' = b.NameBook,
       'Назва магазину' = sh.NameShop,
       'Дата продажу' = s.DateOfSale,
       'Ціна книги' = s.Price
from book.Books b, sale.Sales s, sale.Shops sh
where b.id_book = s.id_book and sh.id_shop = s.id_shop
UNION
select 'Назва книги' = NameBook, 'Немає продаж', NULL, NULL
from book.Books
where id_book not in (select id_book
                     from sale.Sales)
order by s.Price;
```

Результат:

Результат:

Results	Messages			
	Назва книги	Назва магази...	Дата продажу	Ціна книги
12	Сетевые технологии Windows 2000 для профессионалов	Немає продаж	NULL	NULL
13	Язык программирования С	Немає продаж	NULL	NULL
14	Язык программирования С++	Немає продаж	NULL	NULL
15	Основы работы на ПК	Букинист	2000-03-12 00:00:00.000	25,00
16	Толковый словарь компьютерных технологий	Booksworld	2001-11-25 00:00:00.000	40,00
17	Толковый словарь компьютерных технологий	All about PC	2002-11-10 00:00:00.000	45,00
18	Толковый словарь компьютерных технологий	Букинист	2002-10-25 00:00:00.000	45,00

книги, які не продавались

книги, які продавались

книги, які не продавались

книги, які продавались

Напишемо ще один запит, який дозволить об'єднати дві умови: виводить список авторів Росії та США.

```
SELECT FirstName+' '+LastName as 'Автор'
FROM book.Authors a, global.Country c
WHERE a.ID_COUNTRY=c.ID_COUNTRY AND c.NameCountry='Росія'
UNION
SELECT FirstName+' '+LastName as 'Автор'
FROM book.Authors a, global.Country c
WHERE a.ID_COUNTRY=c.ID_COUNTRY AND c.NameCountry='США'
order by 1 desc;
```

Результат:

Результат:

Results		Messages		
	ID_AUTHOR	FirstName	LastName	ID_COUNTRY
1	8	Тарас	Тимошок	1
2	13	Стивен	Прата	NULL
3	6	Сергей	Парижский	2
4	17	Сергей	Михайлов	3
5	7	Сергей	Михайлов	2
6	1	Вильям	Реймаер	4



Results		Messages	
	Автор		
1	Сергей Парижский		
2	Сергей Михайлов		
3	Олег Лисовский		
4	Михаил Мочерный		
5	Джес Либерти		
6	Андрей Ковязин		



Як видно з вищеописаного прикладу, оператор UNION повертає нам лише унікальні записи. Це гарно видно з того, що запит повернув одного автора з іменем Сергій Михайлов, тоді як насправді їх два. Щоб перелічити всіх авторів, слід скористатись оператором UNION ALL.

```
SELECT FirstName+' '+LastName as 'Автор'
FROM book.Authors a, global.Country c
WHERE a.ID_COUNTRY=c.ID_COUNTRY AND c.NameCountry='Росія'
UNION ALL
SELECT FirstName+' '+LastName as 'Автор'
FROM book.Authors a, global.Country c
WHERE a.ID_COUNTRY=c.ID_COUNTRY AND c.NameCountry='США'
order by 1 desc;
```

Результат:

Results				
ID_AUTHOR	FirstName	LastName	ID_COUNTRY	
1	8	Тарас	Тимошок	1
2	13	Стивен	Прата	NULL
3	6	Сергей	Парижский	2
4	17	Сергей	Михайлов	3
5	7	Сергей	Михайлов	2
6	1	Ричард	Вейман	4

Results	
Автор	
1	Сергей Парижский
2	Сергей Михайлов
3	Сергей Михайлов
4	Олег Лисовский
5	Михаил Мочерный

Крім інструкції UNION для об'єднання таблиць як множини записів в SQL Server використовуються інструкції INTERSECT та EXCEPT, які здійснюють відбір значень при їх перетині (записи, які містяться в обох командах SELECT) або відніманні (записи, які містяться в першому операторі SELECT і не містяться в другому) даних відповідно. При цьому обидва оператори можуть використовуватись з інструкцією ALL (наприклад, INTERSECT ALL) і її наявність вказує на дублювання даних в результуючій вибірці.

ПРИМІТКА! В інших СУБД замість EXCEPT використовується інструкція MINUS або DIFFERENCE, синтаксис та суть яких аналогічна.

Наприклад, щоб отримати інформацію про те, які автори однофамільці проживають у Росії та США, слід виконати наступний запит:

```
SELECT FirstName+' '+LastName as 'Автор'
FROM book.Authors a, global.Country c
WHERE a.ID_COUNTRY=c.ID_COUNTRY AND c.NameCountry='Росія'
INTERSECT
SELECT FirstName+' '+LastName as 'Автор'
FROM book.Authors a, global.Country c
WHERE a.ID_COUNTRY=c.ID_COUNTRY AND c.NameCountry='США'
order by 1 desc;
```

Результат:

Results				
ID_AUTHOR	FirstName	LastName	ID_COUNTRY	
1	8	Тарас	Тимошок	1
2	13	Стивен	Прата	NULL
3	6	Сергей	Парижский	2
4	17	Сергей	Михайлов	3
5	7	Сергей	Михайлов	2
6	1	Ричард	Вейман	4

Results	
Автор	
1	Сергей Михайлов

3. Об'єднання таблиць. Стандарт SQL2. Види об'єднань

Базова структура виразу FROM містить ім'я однієї таблиці або представлення, але щоб скористатись всією потужністю реляційної моделі бази даних, потрібно мати можливість створювати багатотабличні запити, тобто отримувати дані з кількох зв'язаних таблиць або представлень. Існує можливість об'єднувати максимум 256 таблиць, але навряд чи Вам доведеться дійти до такої цифри.

Щоб здійснити об'єднання таблиць, потрібно співставити одне або більше полів таблиці з одним або більше полями іншої таблиці (таблиць). Результатом такого співставлення будуть нові записи, які складаються з полів об'єднаних таблиць, перерахованих в списку полів оператора SELECT і які задовольняють умові об'єднання. При об'єднанні таблиць можна скористатись одним з синтаксисів стандарту ANSI: старого стандарту SQL'89 або стандартів SQL2 та вище.



Вони виглядають наступним чином:

```
-- SQL'89
SELECT [таблиця.]поле [, ... n]
FROM список_таблиць
WHERE умова_об'єднання

-- SQL2
SELECT [таблиця.]поле [, ... n]
FROM таблиця [тип_об'єднання] JOIN таблиця
ON умова_об'єднання
```

Оператор об'єднання описує тип виконуваного зв'язування. Умова зв'язування являє собою вираз, аналогічний умові відбору, який використовується в виразі **WHERE**. Вона задає як будуть відноситись рядки в двох таблицях. Більшість операцій зв'язування виконуються на основі виразів еквівалентності, таких як **ПолеА = ПолеВ**. Однак умова об'єднання може бути і більша, при цьому всі вирази, які входять в умову об'єднуються за допомогою логічних операторів **AND** або **OR**.

Microsoft SQL Server підтримує наступні **типи об'єднань**:

1. **Внутрішні**, які здійснюються за допомогою оператора **INNER JOIN**. Причому використання даного оператора без ключового слова **INNER** також допускається. В результаті такого об'єднання отримується нова таблиця, записи якої задовольняють відповідним умовам. Внутрішні об'єднання повертають дані, якщо знаходять спільну інформацію в обох таблицях.

Наприклад, напишемо запит, який виводить список книг тематик "Windows NT" та "ОС до Windows 98":

```
-- старий стандарт SQL
select b.NameBook, t.NameTheme
from book.Books b, book.Themes t
where b.id_theme = t.id_theme
and t.NameTheme in('Windows NT', 'ОС до Windows 98')

-- стандарт SQL2
select b.NameBook, t.NameTheme
from book.Books b
INNER JOIN book.Themes t
ON b.id_theme = t.id_theme
where t.NameTheme in('Windows NT', 'ОС до Windows 98')
```

У випадку об'єднання більше, ніж двох таблиць, потрібно бути уважним, щоб прослідкувати всі зв'язки. Для прикладу, додамо до нашого запиту ще поле з імен автора певної книги:

```
-- старий стандарт SQL
select b.NameBook, t.NameTheme, a.FirstName + ' ' + a.LastName
from book.Books b, book.Themes t, book.Authors a
where b.id_theme = t.id_theme
and b.id_author = a.id_author
and t.NameTheme in('Windows NT', 'ОС до Windows 98')

-- стандарт SQL2
select b.NameBook, t.NameTheme, a.FirstName + ' ' + a.LastName
from book.Themes t
INNER JOIN book.Books b ON b.id_theme = t.id_theme
INNER JOIN book.Authors a ON b.id_author = a.id_author
where t.NameTheme in('Windows NT', 'ОС до Windows 98')
```

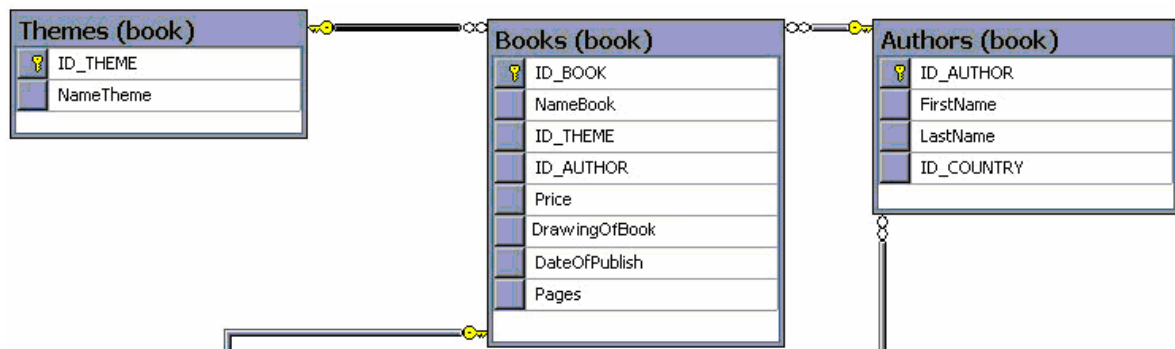
Результат:

Results			
	NameBook	NameTheme	(No column name)
1	Windows NT 5 перспектива	Windows NT	Михаил Мочерный

Results			
	NameBook	NameTheme	(No column name)
1	Windows NT 5 перспектива	Windows NT	Михаил Мочерный



Як бачите, результат однаковий. При використанні нового стандарту ANSI SQL2 слід лише просто запам'ятати один принцип: **при зв'язку таблиць, вони повинні утворювати суцільний послідовний ланцюг**. В нашому випадку він наступний:



2. **Зовнішні об'єднання** - здійснюються за допомогою оператора OUTER JOIN. Використовуються у випадку, коли потрібно, щоб запит повертав всі записи з однієї або більше таблиць, незалежно від того, чи мають вони відповідні записи в іншій таблиці. **Зовнішнє зв'язування може бути:**

- LEFT OUTER JOIN. Ліве об'єднання** – записи першої таблиці (зліва) включаються в результуючу таблицю повністю, а записи з другої таблиці (справа) лише ті, що мають пару в першій таблиці. В якості пари для записів першої таблиці, що не має пари використовують пусті (NULL) поля.
- RIGHT OUTER JOIN. Праве об'єднання** – навпаки.
- FULL OUTER JOIN. Повне об'єднання** – включає всі співставляємі і неспівставляємі записи обох таблиць.

Отже, напишемо запит, який виводить повний список книг та магазини, що продавали дані книги:

```

select b.NameBook, sh.NameShop
from book.Books b
     LEFT OUTER JOIN sale.Sales s   ON b.id_book = s.id_book
     LEFT OUTER JOIN sale.Shops sh  ON s.id_shop = sh.id_shop
order by 1;
  
```

Результат:

	NameBook	NameShop
1	Adobe Web-дизайн и публикация. Энциклопедия поль...	NULL
2	JavaScript: сборник рецептов для профессионалов	Книга
3	JavaScript: сборник рецептов для профессионалов	Booksworld
4	JavaScript: сборник рецептов для профессионалов	Світ книги
5	Macromedia Flash 4. Интерактивная веб-анимация	NULL
6	Visual C++ и MFC. 2е издание	NULL
7	Windows 2000 Professional. Руководство Питера Норт...	Слово
8	Windows 2000 Professional. Руководство Питера Норт...	Книга

книги, які
не продавались

В результаті запиту виводиться список всіх книг, які є в базі даних видавництва, та список лише тих магазинів, які продавали вказану книгу, інакше (якщо книга не продавалась) – NULL.

Якщо необхідно результат вивести навпаки, тобто відобразити повний список магазинів і книги, які вони продавали, слід скористатись правим об'єднанням:

```

select distinct sh.NameShop, b.NameBook
from book.Books b
     RIGHT OUTER JOIN sale.Sales s   ON b.id_book = s.id_book
     RIGHT OUTER JOIN sale.Shops sh  ON s.id_shop = sh.id_shop
order by 1;
  
```

Результат:

	NameShop	NameBook
1	All about PC	Толковый словарь компьютерных технологий
2	Book	Как программировать на С
3	Booksworld	Толковый словарь компьютерных технологий
4	Booksworld	JavaScript: сборник рецептов для профессионалов
5	Booksworld	Как программировать на С
6	Букинист	Основы работы на ПК
7	Букинист	Windows NT 5 перспектива
8	Букинист	Толковый словарь компьютерных технологий



Якщо переписати запит з використанням повного об'єднання, тоді отримаємо список всіх магазинів та список всіх книг, незалежно від того продавались вони чи ні:

```
select distinct sh.NameShop, b.NameBook
from book.Books b
    FULL OUTER JOIN sale.Sales s    ON b.id_book = s.id_book
    FULL OUTER JOIN sale.Shops sh  ON s.id_shop = sh.id_shop
order by 1;
```

Результат:

	NameShop	NameBook
10	NULL	Руководство для хакеров 2
11	NULL	Самоучитель работы на персональном компьютере: 3...
12	NULL	Сетевые технологии Windows 2000 для профессионал...
13	NULL	Язык программирования С
14	NULL	Язык программирования С++
15	All about PC	Толковый словарь компьютерных технологий
16	Book	Как программировать на С
17	Booksworld	JavaScript: сборник рецептов для профессионалов
18	Booksworld	Как программировать на С
19	Booksworld	Толковый словарь компьютерных технологий

3. **Самооб'єднання** дозволяють об'єднати записи однієї і тієї ж таблиці, тобто об'єднати таблицю саму з собою. Це може знадобитись, коли потрібні зв'язки між рядками однієї і тієї ж таблиці. Наприклад, наступний запит виведе інформацію про магазини однієї країни.

```
-- старий стандарт SQL
SELECT sh1.NameShop 'Назва магазину'
FROM sale.Shops sh1, sale.Shops sh2, global.Country c
WHERE sh1.id_shop = sh2.id_shop and sh1.id_country = c.id_country
    and c.NameCountry = 'Україна';

-- стандарт SQL2
SELECT sh1.NameShop 'Назва магазину'
FROM sale.Shops sh1
    INNER JOIN sale.Shops sh2    ON sh1.id_shop = sh2.id_shop
    INNER JOIN global.Country c  ON sh2.id_country = c.id_country
WHERE c.NameCountry = 'Україна';
```

Результат:

	NameShop	NameCountry
1	Букинист	Україна
2	Слово	Україна
3	Книга	Росія
4	Booksworld	США
5	All about PC	Великобританія
6	Світ книги	Україна
7	Book	США

	Назва магазину
1	Букинист
2	Слово
3	Світ книги

В такому запиті для таблиці **Shops** ми визначили два різних псевдоніми, якими повідомляємо сам SQL Server, що необхідно мати дві різні таблиці з однаковими даними. Після цього об'єднуємо таблиці так само, як у випадку довільного багатотабличного запиту. Після цього отримуємо записи, що задовільняють умову.

Спочатку це може здатись незвичним, але при роботі з кількома таблицями в запитах ідея самооб'єднання таблиць не повинна викликати великих труднощів.

4. **Перехресні або необмежені об'єднання** – це тип зв'язування таблиць, при якому в результуючому наборі повертаються сполучення всіх рядків з всіх таблиць, які приймають участь у запиті. Простішими словами, результатом такого об'єднання є декартова множина значень. Ці об'єднання створюються шляхом упущення використання оператора WHERE для визначення зв'язків між таблицями, а на рівні стандарту ANSI SQL2 для цього використовується оператор **CROSS JOIN**.

Наприклад, напишемо запит, який виводить список книг та їх тематик:

```
-- старий стандарт SQL
select b.NameBook, t.NameTheme
from book.Books b, book.Themes t;

-- стандарт SQL2
select b.NameBook, t.NameTheme
from book.Books b CROSS JOIN book.Themes t;
```



4. Оператори модифікації даних INSERT, DELETE, UPDATE та підзапити

Кілька пар назад ми розглядали оператори, які дозволяють маніпулювати даними об'єктів бази даних: INSERT, DELETE та UPDATE. Ці оператори призначені для вставки, видалення та зміни значень даних в базі даних і принцип їх роботи нескладний. Та бувають випадки, коли і при написанні такого роду запитів необхідно використовувати підзапити.

В команді INSERT використання підзапитів найпростіше і ви можете використовувати підзапити всередині запиту, який буде генерувати значення для команди INSERT. Наприклад, необхідно вставити дані в нову таблицю **BookAuthor**, яка містить два поля: назву книги та повне ім'я автора.

```
INSERT INTO book.BookAuthor (NameBook, FullNameAuthor)
SELECT b.NameBook, a.FirstName+' '+a.LastName
FROM book.Books b INNER JOIN book.Authors a
ON b.ID_AUTHOR = a.ID_AUTHOR;
```

Список полів при INTO та SELECT в даному випадку необов'язковий, оскільки відібрані дані про назви книг і авторів повністю беруться з таблиць Books та Authors.

Підзапит також можна використати у випадку додавання до таблиці BooksCh всіх книг чеських авторів:

```
INSERT INTO book.BooksCh
SELECT NameBook, Price, DrawingOfBook, DateOfPublish, Pages
FROM book.Books
WHERE ID_AUTHOR = ANY ( SELECT ID_AUTHOR
                        FROM book.Authors a, global.Country c
                        WHERE a.ID_COUNTRY=c.ID_COUNTRY AND c.NameCountry='Чехія');
```

Використання підзапитів з оператором DELETE дозволить накладати складні критерії видалення даних. Наприклад, необхідно видалити всі книги J.White:

```
DELETE
FROM book.Books
WHERE ID_AUTHOR =
      (SELECT ID_AUTHOR
       FROM book.Authors
       WHERE FirstName='Johnson' AND LastName='White');
```

Оператор оновлення даних UPDATE використовує підзапити з тією ж ціллю, що і DELETE. Наприклад, збільшити ціну на 10%, на всі книги, які реалізуються українськими магазинами:

```
UPDATE sale.Sales
SET Price = Price * 0.2
WHERE ID_SHOP IN
      (SELECT ID_SHOP
       FROM sale.Shops sh, global.Country c
       WHERE sh.ID_COUNTRY=c.ID_COUNTRY AND NameCountry='Україна');
```

5. Домашнє завдання

1. Вивести всі книги, які продаються більше, ніж одним магазином.
2. Вивести лише тих авторів, чиї книги продаються більше, ніж книги авторів США.
3. Вивести всіх авторів, які існують в базі даних з вказанням (при наявності) їх книг, які видаються видавництвом.
4. За допомогою підзапитів знайдіть всіх авторів, які живуть в країнах, де є магазин, що продає їх книги. Відсортувати вибірку по прізвищу автора.
5. Довести, що книги тематики, наприклад, «Підручники» випускаються найбільшим тиражем. **Примітка!** Доказом буде NULL значень, інакше книги найбільш реалізуємих тематик.
6. Сформуйте об'єднання з трьох запитів. Перше буде виводити список авторів, тематики книг яких, наприклад, «C & C++»; друге – список авторів, які видавали свої книги у 2000 році; третє виводить список найдорожчих авторів. В двох останніх запитах збережіть дублікати. Відсортувати вибірку по прізвищу автора в спадаючому порядку.
7. Скласти звіт про те, які магазини реалізували найбільшу та найменшу кількість книг видавництва (скористатись оператором UNION).
8. Вивести імена всіх авторів, книг яких не видавало ще видавництво, тобто які не присутні в таблиці Books. (написати два варіанти запиту: один з використанням підзапитів, другий з використанням операторів об'єднання запитів).
9. Створіть таблицю ShopAuthors, яка містить наступні поля: ім'я та прізвище автора, назва магазину та країна. Напишіть команду, яка б вставляла в неї всіх авторів, книги яких реалізуються більше, ніж одним магазином видавництва.
10. Видалити всі книги, в яких кількість сторінок більше, ніж середнє.
11. Видалити всі книги, в яких не вказаний автор, при цьому з результату виключити повторення.



-
12. В зв'язку з закриттям магазинів в Англії, написати запит, який знищує інформацію про всі магазини, які там розмішувались.
 13. Змінити інформацію про продаж книг, які здійснювались одним з магазинів в Чехії, наприклад, магазином «Буква» наступним чином: ціни на книги, які продавались 01/01/2010 року збільшити на 5%, а кількість збільшити на 10 од.
 14. Зменшити кількість книг в магазинах, які за останній рік продали найменшу кількість книг, на 15%.

Примітка! 75% запитів повинні бути написані згідно стандарту SQL2.

Copyright © 2010 Iryn Petrova.