

3.9 Змінні / Variables

- Часто результати запиту необхідно використовувати в наступних запитах
- Often the query results are required in subsequent queries
- Для цього отримані дані необхідно зберегти у тимчасових структурах
- Therefore, retrieved data should be stored in temporary structures
- Це завдання вирішується за допомогою змінних SQL і тимчасових таблиць
- This problem is solved by using SQL variables and temporary tables

3.9 Змінні / Variables

- Оголошення змінної починається з символу @, за яким слідує ім'я змінної
- Variable definition starts with the @ symbol followed by the name of a variable
- Значення змінним присвоюються за допомогою оператора SELECT з використанням оператора присвоювання: =
- Values are assigned to the variables by using the SELECT operator with the assignment operator: =
- Наприклад / For example
`SELECT @total := COUNT(*) FROM books;`

3.9 Змінні / Variables

```
SELECT @total := COUNT(*) FROM books;
```

```
SELECT @total;
```

- Оголошується змінна @total, якій присвоюється число записів в таблиці books
- The variable @total is declared, the number of records in the table books is assigned to this variable
- Потім дана змінна використовується в подальшому запиті в рамках поточного сеансу
- Then this variable is used in the sequential query within the same session
- Змінні діють тільки в рамках сеансу
- Variables are available only within a session

Result #1 (1x1)	Result #2 (1x1)
@total := COUNT(*)	
	30

Result #1 (1x1)	Result #2 (1x1)
@total	
	30

3.9 Змінні / Variables

```
SET @last = CURDATE() - INTERVAL 7 DAY;
```

```
SELECT CURDATE(), @last;
```

Result #1 (2×1)	
CURDATE()	@last
2019-03-19	2019-03-12

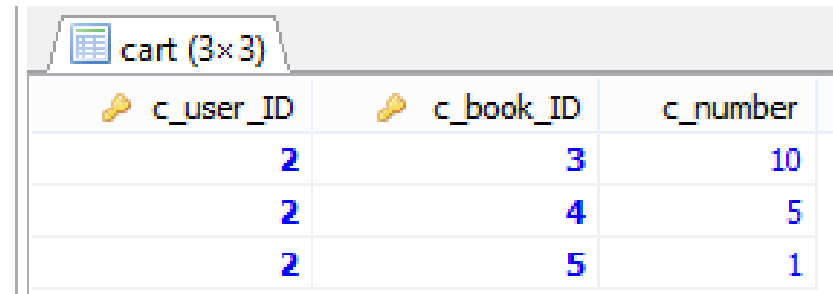
- Змінні також можуть оголошуватися за допомогою оператора SET
- Variables also might be declared using the SET operator
- При цьому може бути використаний звичайний знак рівності = в якості оператора присвоєння
- The usual equals sign = can be used as the assignment operator
- Оператор SET зручний тим, що він не повертає результат привласнення значення змінної
- The SET operator is convenient because it does not return the result of variable assignment



3.10 Тимчасові таблиці / Temporary tables

- Змінна SQL дозволяє зберегти тільки одне проміжне значення
- The SQL variable allows to store only a single intermediate value
- Коли необхідно зберегти результуючу таблицю, вдаються до тимчасових таблиць
- Temporary tables are used to store the result table
- CREATE TEMPORARY TABLE

3.10 Тимчасові таблиці / Temporary tables

```
CREATE TEMPORARY TABLE cart (  
    c_user_ID INT NOT NULL,  
    c_book_ID INT NOT NULL,  
    c_number INT NOT NULL,  
    PRIMARY KEY (c_user_ID, c_book_ID)  
) ENGINE=InnoDB;
```



 c_user_ID	 c_book_ID	c_number
2	3	10
2	4	5
2	5	1

```
INSERT INTO cart (c_user_ID, c_book_ID, c_number) VALUES  
(2, 3, 10), (2, 4, 5), (2, 5, 1);
```

```
SELECT * FROM cart;
```

Тимчасова таблиця автоматично видаляється після закінчення сеансу і може бути використана тільки протягом даного з'єднання

Temporary tables are automatically removed when the session is over, they might be used only within a current session

- Виведення списку покупців і числа здійснених ними покупок, покупців впорядкувати за спаданням числа замовлень
- Print the list of customers and numbers of their purchases, sort customers by a number of orders in a descending order

```
SELECT users.u_surname, users.u_name, users.u_patronymic,  
COUNT(orders.order_ID) AS total  
FROM users INNER JOIN orders ON users.user_ID = orders.o_user_ID  
GROUP BY users.user_ID  
ORDER BY total DESC;
```

users (4x4)			
u_surname	u_name	u_patronymic	total
Симонов	Игорь	Николаевич	2
Корнеев	Александр	Александрович	1
Кузнецов	Максим	Петрович	1
Иванов	Александр	Валерьевич	1

- Вивести повний список покупців, включаючи тих, які не зробили жодної покупки
- Print the complete list of customers, including those who did not make any orders

```
SELECT users.u_surname, users.u_name, users.u_patronymic,  
COUNT(orders.order_ID) AS total  
FROM users LEFT JOIN orders ON users.user_ID = orders.o_user_ID  
GROUP BY users.user_ID  
ORDER BY total DESC;
```

users (4×6)			
u_surname	u_name	u_patronymic	total
Симонов	Игорь	Николаевич	2
Кузнецов	Максим	Петрович	1
Корнеев	Александр	Александрович	1
Иванов	Александр	Валерьевич	1
Петров	Анатолий	Юрьевич	0
Лосев	Сергей	Иванович	0

- Вибрати рядки з таблиці catalogs, у яких первинний ключ збігається з одним зі значень, що повертаються вкладеним запитом
- Select records from the table catalog, which primary key matches one of the values returned by the nested query

```
SELECT cat_name FROM catalogs
WHERE cat_ID IN (SELECT b_cat_ID
FROM books GROUP BY b_cat_ID);
```

catalogs (1×5)
cat_name
Программирование
Интернет
Базы данных
Сети
Мультимедиа

- Вивести імена і прізвища покупців, які вчинили хоча б одну покупку
- Print names and surnames of customers that made at least one order

```
SELECT u_name, u_surname FROM users
WHERE user_ID = ANY(SELECT o_user_ID
FROM orders);
```

users (2×4)	
u_name	u_surname
Александр	Иванов
Игорь	Симонов
Максим	Кузнецов
Александр	Корнеев

- Вивести всі товарні позиції, ціна яких перевищує середню ціну кожного з каталогів
- Print all products which price is greater than average price of each category

```
SELECT b_name, b_price FROM books
WHERE b_price > ALL(SELECT AVG(b_price)
FROM books GROUP BY b_cat_ID);
```

b_name	b_price
Visual FoxPro 9.0	660.00
Delphi. Полное руководство	500.00
Совершенный код	771.00
Принципы маршрутизации в Internet	428.00
Компьютерные сети	630.00
Сети. Поиск неисправностей	434.00
Безопасность сетей	462.00

- Вивести імена і прізвища покупців, які вчинили хоча б одну покупку
- Print names and surnames of customers that made at least one order

```
SELECT u_name, u_surname FROM users
WHERE EXISTS (SELECT * FROM orders
WHERE orders.o_user_ID = users.user_ID);
```

u_name	u_surname
Александр	Иванов
Игорь	Симонов
Максим	Кузнецов
Александр	Корнеев