# Лабораторна робота № 5. Підзапити

**УВАГА! Для роботи можна використовувати БД з лабораторної роботи 4.**

|  |
| --- |
| USE productsdb;  CREATE TABLE Products  (  Id INT IDENTITY PRIMARY KEY,  ProductName NVARCHAR (30) NOT NULL,  Manufacturer NVARCHAR (20) NOT NULL,  ProductCount INT DEFAULT 0,  Price MONEY NOT NULL  );  CREATE TABLE Customers  (  Id INT IDENTITY PRIMARY KEY,  FirstName NVARCHAR (30) NOT NULL  );  CREATE TABLE Orders  (Id INT IDENTITY PRIMARY KEY,  ProductId INT NOT NULL REFERENCES Products (Id),  CustomerId INT NOT NULL REFERENCES Customers (Id),  CreatedAt DATE NOT NULL,  ProductCount INT DEFAULT 1,  Price MONEY NOT NULL); |

Таблиця Orders містить посилання на дві інші таблиці через поля ProductId і CustomerId. Додамо в таблиці деякі дані:

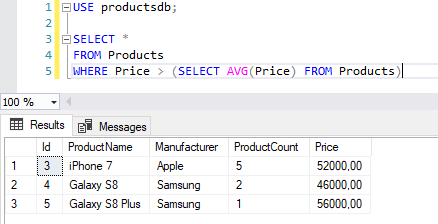
|  |
| --- |
| INSERT INTO Products  VALUES ( 'IPhone 11', 'Apple', 2, 21999),  ( 'IPhone 12 Pro', 'Apple', 2, 56000),  ( 'IPhone 12', 'Apple', 5, 28999),  ( 'Galaxy S21 Ultra', 'Samsung', 2, 39999),  ( 'Galaxy Z Fold2', 'Samsung', 1, 60000),  ( 'Xiaomi Mi 11', 'Xiaomi', 2, 26999),  ( 'OnePlus 8', 'OnePlus', 6, 17775)  INSERT INTO Customers VALUES ('Tom'), ('Bob'), ('Sam')  INSERT INTO Orders  VALUES  (4,2,'2021-02-11',2,39999),  (2,2,'2021-02-11',2,56000),  (4,1,'2021-02-13',1,39999),  (7,1,'2021-02-14',5,17775) |

Підзапит виконує команду SELECT і полягає в дужки. В даному ж випадку при додаванні одного товару виконується три підзапиту. Кожен підзапит повертає одного скалярний значення, наприклад, числовий ідентифікатор. В даному випадку підзапити виконувалися до іншої таблиці, але можуть виконуватися і до тієї ж, до якої викликається основний запит. Наприклад, знайдемо товари з таблиці Products, які мають мінімальну ціну:

|  |
| --- |
| SELECT \*  FROM Products  WHERE Price = (SELECT MIN (Price) FROM Products) |

Або знайдемо товари, ціна яких перевищує середню:

|  |
| --- |
| SELECT \*  FROM Products  WHERE Price> (SELECT AVG (Price) FROM Products) |



**Корелюючі підзапити**

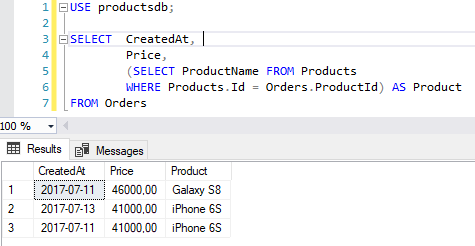
Підзапити бувають коррелирующими і некоррелірующімі. У прикладах вище команди SELECT виконували фактично один підзапит для всієї команди, наприклад, підзапит повертає мінімальну або середню ціну, яка не зміниться, скільки б ми рядків не вибирали в основному запиті. Тобто результат підзапиту не залежав від рядків, які вибираються в основному запиті. І такий підзапит виконується один раз для всього зовнішнього запиту.

Але також існують корелюють підзапити (correlated subquery), результати яких залежать від рядків, які вибираються в основному запиті.

Наприклад, виберемо всі замовлення з таблиці Orders, додавши до них інформацію про товар:

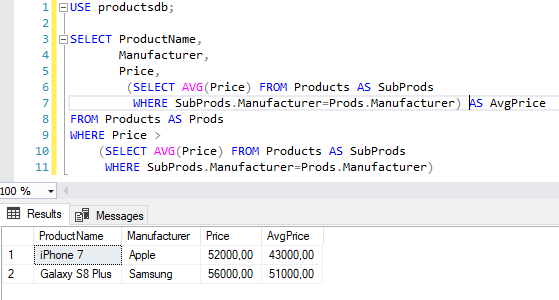
|  |
| --- |
| SELECT CreatedAt,  Price,  (SELECT ProductName FROM Products  WHERE Products.Id = Orders.ProductId) AS Product  FROM Orders |

Тут для кожного рядка з таблиці Orders буде виконуватися підзапит, результат якого залежить від стовпчика ProductId. І кожен підзапит може повертати різні дані.



Корелює підзапит може виконуватися і для тієї ж таблиці, до якої виконується основний запит. Наприклад, виберемо з таблиці Products ті товари, вартість яких вища за середню ціну товарів для даного виробника:

|  |
| --- |
| SELECT ProductName,  Manufacturer,  Price,  (SELECT AVG (Price) FROM Products AS SubProds  WHERE SubProds.Manufacturer = Prods.Manufacturer) AS AvgPrice  FROM Products AS Prods  WHERE Price>  (SELECT AVG (Price) FROM Products AS SubProds  WHERE SubProds.Manufacturer = Prods.Manufacturer) |



В даному випадку визначено два корелюють підзапиту. Перший підзапит визначає специфікацію стовпця AvgPrice. Він буде виконуватися для кожного рядка, що витягується з таблиці Products. У підзапит передається виробник товару і на його основі вибирається середня ціна для товарів саме цього виробника. І так як виробник у товарів може відрізнятися, то і результат підзапиту в кожному випадку також може відрізнятися.

Другий підзапит аналогічний, тільки він використовується для фільтрації витягають із таблиці Products. І також він буде виконуватися для кожного рядка.

Щоб уникнути подвійності при фільтрації в підзапиті при порівнянні виробників (SubProds.Manufacturer = Prods.Manufacturer) Для зовнішньої вибірки встановлений псевдонім Prods, а для вибірки з підзапитів визначено псевдонім SubProds.

Слід враховувати, що корелюють підзапити виконуються для кожної окремої рядки вибірки, то виконання таких підзапитів може уповільнювати виконання всього запиту в цілому.

**Завдання на підзапити**

* 1. Додати при виведенні таблиці Orders поле з загальною кількістю замовлень в базі.
  2. Додати при виведенні таблиці Products поле з середньою вартістю товарів
  3. Вивести товари, довжина назв яких довше середнього значення довжини.
  4. Вивести фірми продукції для кожного замовлення
  5. Вивести телефон і ім’я замовника у кожному замовленні (для цього в таблицю Cudtomers ввести поле Phone, якщо користувач не буде мати телефону (NULL або ‘’), то виводити в запиті повідомлення ‘Телефон відсутній’)
  6. Додати при виведенні користувачів поле з кількістю проведених замовлень
  7. Додати при виведенні таблиці Orders поле з загальною кількістю замовлень на день замовлення поточного продукту (НЕ плутати з 1)
  8. Додати при виведенні таблиці Products поле з середньою вартістю товарів даної фірми (НЕ плутати з 2)
  9. Додати при виведенні користувачів поле з сумою за всіма замовленнями за поточний місяць.
  10. Вивести таблицю, яка підрахує кількість замовлень, що містять товари фірми Apple, та в окреммому полі вкажуть текст «Залишилось мало», якщо кількість продукції даної фірми менше 5
  11. \* Додати при виведенні таблиці Products поле з ім’ям користувача, який купив цей товар останнім.
  12. \*На основі запиту п. 2 отримати різницю між середньою вартістю товарів та вартістю поточного товара
  13. \*\* Показати товари, кількість залишків яких менше сумарної кількості продажів за цим товаром за лютий місяць.
  14. \* Додати при виведенні користувачів (див. п.9) поле з сумою за замовленнями користувача за поточний місяць.