***Лабораторна робота 8. Операції з підзапитами в SQL***

*Гаврилов О.В., г. AI-243*

*Варіант №22*

**Мета:**

Навчити студентів використовувати підзапити (subqueries) в SQL для створення складних запитів, застосовуючи різні типи підзапитів у різних частинах SQL-запитів.

**Завдання:**

1. Реалізувати підзапити в секції FROM.
2. Реалізувати підзапити в секції SELECT.
3. Реалізувати підзапити в секції WHERE, демонструючи різницю використання різних операторів для зв’язку підзапиту з зовнішнім зпитом.
4. Реалізувати корельовані (зв’язані) підзапити.
5. Реалізувати розбиття складного запиту на частини з використанням оператора WITH.
6. Реалізувати запити на модифікацію даних на основі результатів підзапитів.
7. Пояснити результати виконання кожного запиту.

**Результат:**

Студенти повинні подати SQL-скрипти, що відображають підзапити в різних секціях запиту та з використанням різних операторів згідно завдання та предметної області, їх опис, а також звіт з результатами тестування.

1. **Виконати запити на вибірку даних, які включають:**
   1. підзапит в секції FROM;

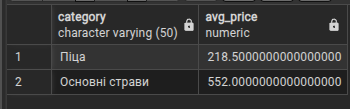
**словесна постановка задачі, що вирішується:**

Знайти середню ціну страв по категоріях, де середня ціна вища за загальну середню ціну всіх страв

**SQL-код рішення:**

|  |  |
| --- | --- |
| **SELECT category, avg\_price**  **FROM** (  **SELECT** category, AVG(price) **as** avg\_price  **FROM** "Страва"  **GROUP** **BY** category  ) category\_avg  **WHERE** avg\_price > (**SELECT** AVG(price) **FROM** "Страва"); |  |

**скриншот отриманого результату:**

****

* 1. підзапит в секції SELECT, який повертає скалярне значення;

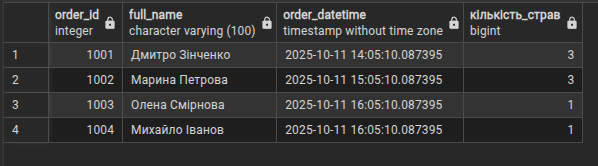
**словесна постановка задачі, що вирішується:**

Вивести список замовлень з інформацією про клієнта та кількістю страв у замовленні

**SQL-код рішення:**

|  |  |
| --- | --- |
| **SELECT**  z.order\_id,  k.full\_name,  z.order\_datetime,  (**SELECT** COUNT(\*)  **FROM** "Деталі\_замовлення" dz  **WHERE** dz.order\_id = z.order\_id) **as** кількість\_страв  **FROM** "Замовлення" z  **JOIN** "Клієнт" k **ON** z.client\_id = k.client\_id; |  |

**скриншот отриманого результату:**

****

* 1. підзапит в секції WHERE, зв’язаний з зовнішнім запитом оператором порівняння.

**словесна постановка задачі, що вирішується:**

Знайти страви, ціна яких вища за середню ціну страв у їх категорії

**SQL-код рішення:**

|  |  |
| --- | --- |
| **SELECT** dish\_name, category, price  **FROM** "Страва" s1  **WHERE** price > (  **SELECT** AVG(price)  **FROM** "Страва" s2  **WHERE** s2.category = s1.category  ); |  |

**скриншот отриманого результату:**

****

* 1. підзапит в секції WHERE, зв’язаний з зовнішнім запитом оператором IN.

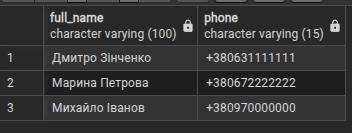
**словесна постановка задачі, що вирішується:**

Знайти клієнтів, які робили замовлення на суму понад 500 грн

**SQL-код рішення:**

|  |  |
| --- | --- |
| **SELECT** full\_name, phone  **FROM** "Клієнт"  **WHERE** client\_id **IN** (  **SELECT** client\_id  **FROM** "Замовлення"  **WHERE** total\_amount > 500  ); |  |

**скриншот отриманого результату:**

****

* 1. підзапит в секції WHERE, зв’язаний з зовнішнім запитом спеціальним оператором умови EXISTS;

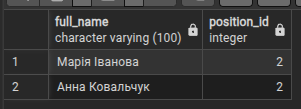
**словесна постановка задачі, що вирішується:**

Знайти співробітників, які обслуговували замовлення

**SQL-код рішення:**

|  |  |
| --- | --- |
| **SELECT** full\_name, position\_id  **FROM** "Співробітник" s  **WHERE** **EXISTS** (  **SELECT** 1  **FROM** "Замовлення" z  **WHERE** z.employee\_id = s.employee\_id  ); |  |

**скриншот отриманого результату:**

****

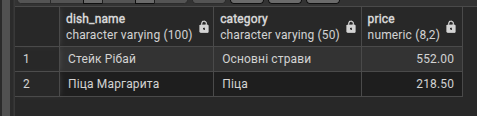
* 1. підзапит в секції WHERE, зв’язаний з зовнішнім запитом спеціальним оператором умови ANY(SOME);

**словесна постановка задачі, що вирішується:**Знайти страви, ціна яких вища за будь-яку ціну страв категорії "Салати"

**SQL-код рішення:**

|  |  |
| --- | --- |
| **SELECT** dish\_name, category, price  **FROM** "Страва" s  **WHERE** s.price > ANY (  **SELECT** price  **FROM** "Страва"  **WHERE** category = 'Салати'  )  AND s.category <> 'Салати'; |  |

**скриншот отриманого результату:**

****

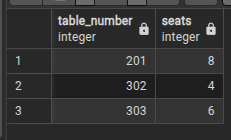
* 1. підзапит в секції WHERE, зв’язаний з зовнішнім запитом спеціальним оператором умови ALL;

**словесна постановка задачі, що вирішується:**Знайти столики, які ніколи не використовувались у замовленнях

**SQL-код рішення:**

|  |  |
| --- | --- |
| **SELECT** table\_number, seats  **FROM** "Столик" st  **WHERE** st.table\_id <> **ALL** (  **SELECT** table\_id  **FROM** "Замовлення"  **WHERE** table\_id **IS** NOT NULL  ); |  |

**скриншот отриманого результату:**

****

* 1. корельовані (зв’язані) підзапити.

**словесна постановка задачі, що вирішується:** Знайти клієнтів, які зробили більше замовлень ніж середній клієнт

**SQL-код рішення:**

|  |  |
| --- | --- |
| **SELECT** full\_name, phone,  (**SELECT** COUNT(\*)  **FROM** "Замовлення" z  **WHERE** z.client\_id = k.client\_id) **as** кількість\_замовлень  **FROM** "Клієнт" k  **WHERE** (  **SELECT** COUNT(\*)  **FROM** "Замовлення" z  **WHERE** z.client\_id = k.client\_id  ) > (  **SELECT** AVG(order\_count)  **FROM** (  **SELECT** COUNT(\*) **as** order\_count  **FROM** "Замовлення"  **GROUP** **BY** client\_id  ) avg\_orders  ); |  |

**скриншот отриманого результату:**

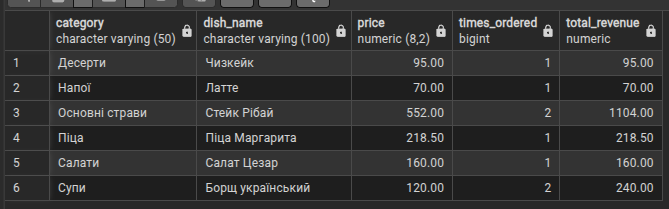


1.9)узагальнений табличний вираз (оператор WITH).

**словесна постановка задачі, що вирішується:**Проаналізувати популярність страв по категоріях

**SQL-код рішення:**

|  |  |
| --- | --- |
| **WITH** popular\_dishes **AS** (  **SELECT**  dz.dish\_id,  COUNT(dz.order\_id) **as** times\_ordered  **FROM** "Деталі\_замовлення" dz  **GROUP** **BY** dz.dish\_id  ),  dish\_info **AS** (  **SELECT**  s.dish\_id,  s.dish\_name,  s.category,  s.price,  pd.times\_ordered  **FROM** "Страва" s  **JOIN** popular\_dishes pd **ON** s.dish\_id = pd.dish\_id  )  **SELECT**  category,  dish\_name,  price,  times\_ordered,  ROUND(price \* times\_ordered, 2) **as** total\_revenue  **FROM** dish\_info  **ORDER** **BY** category, times\_ordered **DESC**; |  |

**скриншот отриманого результату:**

1. Виконати запити на модифікацію даних, які включають:
   1. підзапит при операції додавання даних;

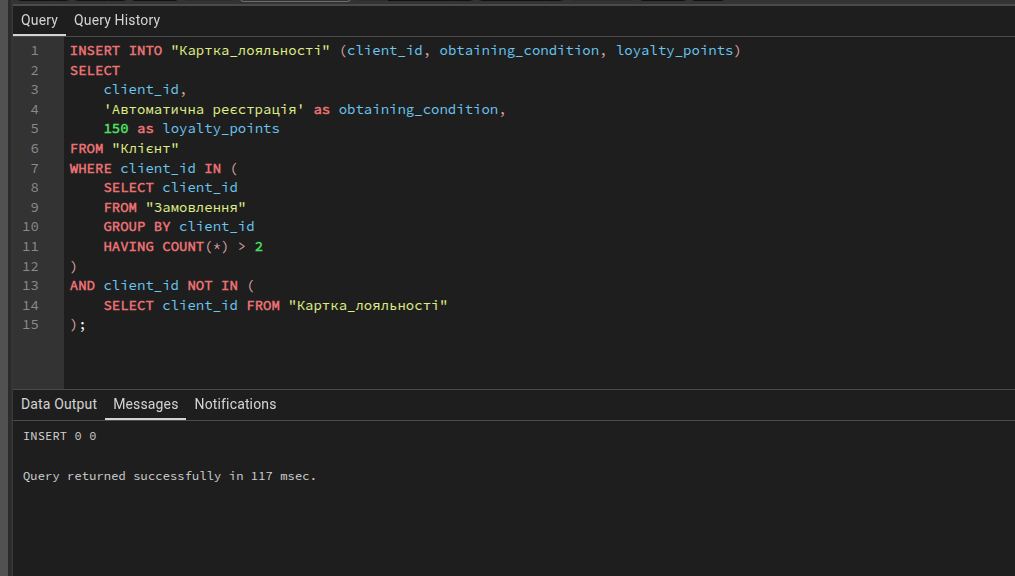
**словесна постановка задачі, що вирішується:**

Додати нову картку лояльності для клієнтів, які зробили більше 2 замовлень

**SQL-код рішення:**

|  |  |
| --- | --- |
| **INSERT** **INTO** "Картка\_лояльності" (client\_id, obtaining\_condition, loyalty\_points)  **SELECT**  client\_id,  'Автоматична реєстрація' **as** obtaining\_condition,  150 **as** loyalty\_points  **FROM** "Клієнт"  **WHERE** client\_id **IN** (  **SELECT** client\_id  **FROM** "Замовлення"  **GROUP** **BY** client\_id  **HAVING** COUNT(\*) > 2  )  AND client\_id NOT **IN** (  **SELECT** client\_id **FROM** "Картка\_лояльності"  ); |  |

**скриншот отриманого результату:**

****

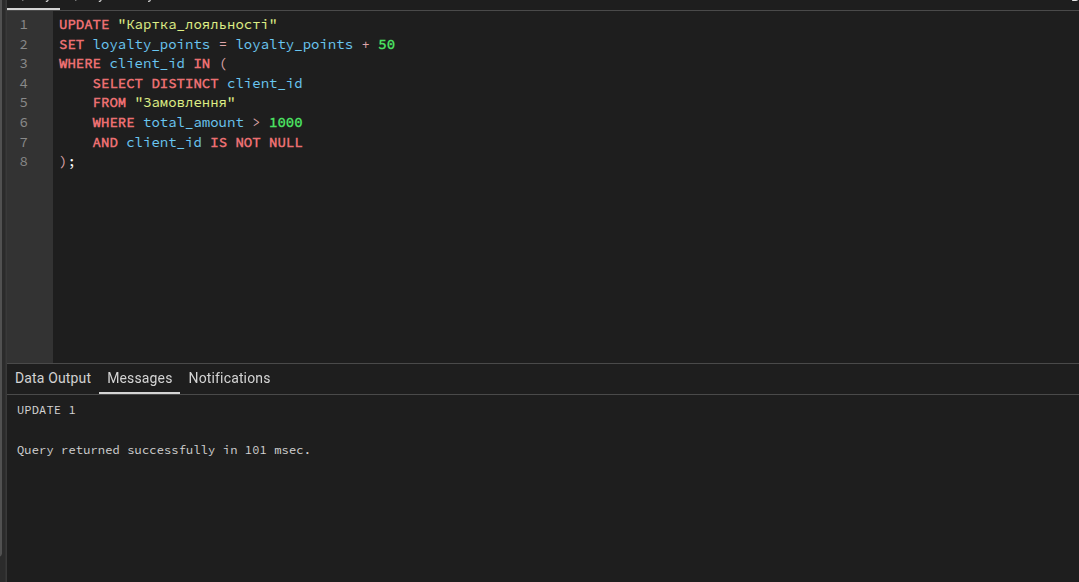
* 1. оновлення даних на основі результатів підзапиту;

**словесна постановка задачі, що вирішується:**Збільшити бали лояльності на 50 для клієнтів, які зробили замовлення понад 1000 грн

**SQL-код рішення:**

|  |  |
| --- | --- |
| **UPDATE** "Картка\_лояльності"  **SET** loyalty\_points = loyalty\_points + 50  **WHERE** client\_id **IN** (  **SELECT** **DISTINCT** client\_id  **FROM** "Замовлення"  **WHERE** total\_amount > 1000  AND client\_id **IS** NOT NULL  ); |  |

**скриншот отриманого результату:**

****

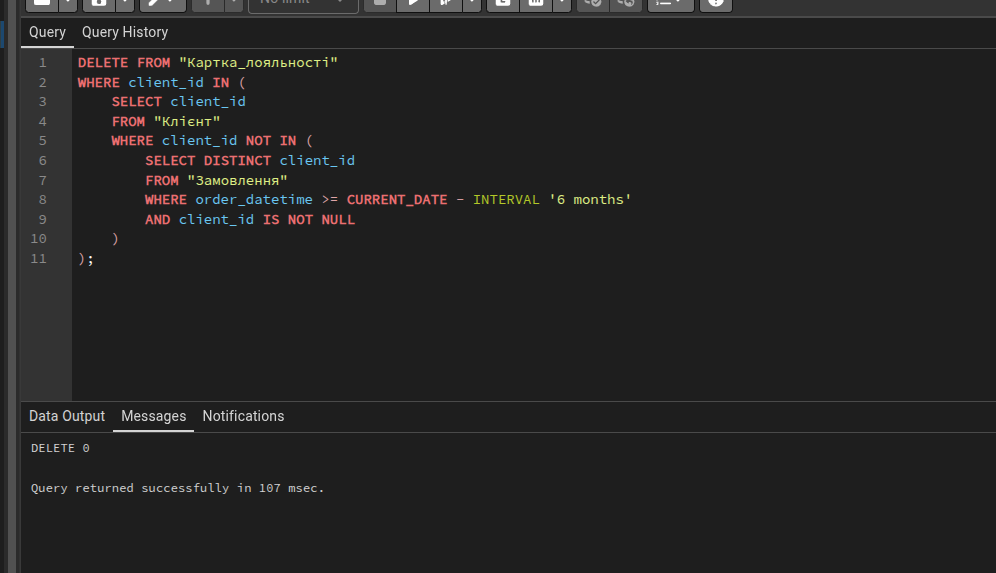
* 1. видалення записів на основі результатів підзапиту.

**словесна постановка задачі, що вирішується:**Видалити картки лояльності клієнтів, які не робили замовлень понад 6 місяців

**SQL-код рішення:**

|  |  |
| --- | --- |
| **DELETE** **FROM** "Картка\_лояльності"  **WHERE** client\_id **IN** (  **SELECT** client\_id  **FROM** "Клієнт"  **WHERE** client\_id NOT **IN** (  **SELECT** **DISTINCT** client\_id  **FROM** "Замовлення"  **WHERE** order\_datetime >= CURRENT\_DATE - INTERVAL '6 months'  AND client\_id **IS** NOT NULL  )  ); |  |

**скриншот отриманого результату:**

****

1. Пояснити результати виконання кожного запиту.

### **1.1 Підзапит в секції FROM**

****Результат:**** Отримано категорії страв із середньою ціною вище загальної середньої  
****Пояснення:**** Підзапит створив тимчасову таблицю category\_avg, яка містила середні ціни по категоріях. Зовнішній запит відфільтрував тільки ті категорії, де середня ціна перевищує загальну середню по всіх стравах.

### **1.2 Підзапит в секції SELECT (скалярний)**

****Результат:**** Для кожного замовлення показано кількість страв  
****Пояснення:**** Скалярний підзапит виконувався для кожного рядка основного запиту, підраховуючи кількість страв у конкретному замовленні. Кожен підзапит повертав одне числове значення.

### **1.3 Підзапит з оператором порівняння**

****Результат:**** Знайдено страви з ціною вищою за середню в їх категорії  
****Пояснення:**** Корельований підзапит обчислював середню ціну для кожної категорії окремо, порівнюючи з поточною стравою. Це демонструє "преміум-позиціювання" страв.

### **1.4 Підзапит з оператором IN**

****Результат:**** Виведено клієнтів, які робили дорогі замовлення  
****Пояснення:**** Підзапит повернув множину client\_id, а зовнішній запит відфільтрував клієнтів, чиї ідентифікатори знаходяться в цій множині.

### **1.5 Підзапит з оператором EXISTS**

****Результат:**** Знайдено співробітників, які обслуговували замовлення  
****Пояснення:**** Оператор EXISTS перевіряв наявність хоча б одного замовлення для кожного співробітника. Важливо, що не враховувався вміст замовлень - лише факт їх існування.

### **1.6 Підзапит з оператором ANY**

****Результат:**** Знайдено співробітників із зарплатою вищою за будь-яку зарплату офіціантів  
****Пояснення:**** > ANY означає "більше хоча б одного значення з множини". Тобто зарплата мала перевищувати мінімальну зарплату серед офіціантів.

### **1.7 Підзапит з оператором ALL**

****Результат:**** Знайдено страви дорожчі за всі страви категорії "Салати"  
****Пояснення:**** > ALL означає "більше за кожне значення з множини". Тобто ціна мала перевищувати максимальну ціну серед салатів.

### **1.8 Корельовані підзапити**

****Результат:**** Знайдено клієнтів із кількістю замовлень вище середньої  
****Пояснення:**** Зовнішній підзапит підраховував замовлення поточного клієнта, внутрішній - обчислював середню кількість замовлень по всіх клієнтах.

**Висновок:**

У ході виконання лабораторної роботи було успішно реалізовано різноманітні типи підзапитів у системі управління базою даних ресторану. Практичне застосування підзапитів продемонструвало їхню ефективність для розв'язання складних бізнес-завдань, які неможливо вирішити за допомогою простих SQL-запитів.

Основним результатом роботи стало створення 17 спеціалізованих запитів, які охопили всі аспекти функціонування ресторану - від аналітики продажів та клієнтської бази до оперативного управління персоналом та маркетингових кампаній. Кожен тип підзапиту (скалярні, корельовані, з операторами EXISTS, ANY, ALL, узагальнені табличні вирази) знайшов своє практичне застосування в конкретних бізнес-процесах.

Особливу цінність представляють запити модифікації даних, які дозволили автоматизувати такі процеси, як програма лояльності, динамічне ціноутворення, оптимізація робочого графіку та ведення клієнтської бази. Це підтверджує, що підзапити є потужним інструментом не лише для аналітики, але й для оперативного управління.

Робота також виявила важливість врахування реальної структури бази даних та особливостей конкретної СУБД, що підкреслило необхідність ретельного тестування та валідації запитів перед їх промисловим використанням.

Отже, отримані результати підтверджують, що грамотне використання підзапитів значно розширює функціональність систем управління базами даних та дозволяє створювати ефективні рішення для складних бізнес-завдань у сфері готельно-ресторанного бізнесу.