**Лабораторна робота 5**

**Вкладені та корельовані запити, іменовані набори та табличні вирази.**

**Мета роботи:** розібрати поняття псевдонімів та вкладених і корельованих запитів у різних розділах команди select.

**Задачі:**

* Вкладені запити
* Корельовані запити у реченні where.
* Корельовані запити у реченні from
* Корельовані запити у реченні select
* Робота із змінними і узагальненими табличними виразами
* Виконання індивідуального завдання згідно варіанту

1. Під час написання запиту для зручності (а інколи і для функціональності) по відношенню до таблиць або окремих полів можуть бути використані псевдоніми. Проілюструємо це на попередньому прикладі – вкажемо короткі псевдоніми g, s та f для таблиць, та зрозумілі facult\_name та special\_name для назв факультетів та спеціальностей (рис.1)

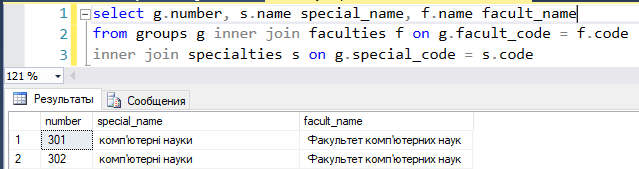


Рис.1. Використання псевдонімів

Поєднання таблиць може бути виконано не тільки за одним полем і не тільки на рівність полів табличок. Так, наприклад, отримаємо всі можливі варіанти пар студентів у межах однієї групи (рис.2)

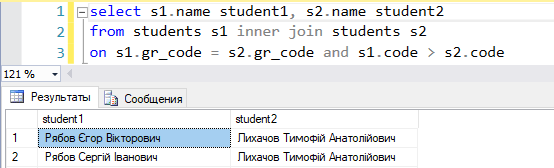


Рис.2. Всі можливі варіанти пар студентів у межах однієї групи

1. Далі розглянемо вкладені запити та роботу із ними. Вкладені запити можуть перебувати у реченнях where (having), from та select. Почнемо із випадку where, що зустрічається найбільш часто, а до інших повернемося трохи пізніше. Виберемо студентів, що мають таку ж дату народження, що і «Бірюкова Надія Миколаївна» (рис.3)

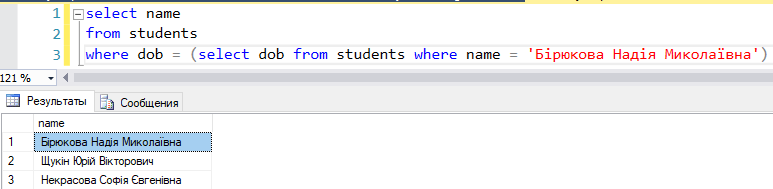


Рис.3. Студентів, що мають таку ж дату народження, що і «Бірюкова Надія Миколаївна»

Для виключення із результату самої Бірюкової, можемо додати ще один вкладений запит, але тепер уже із нерівністю (рис.4).

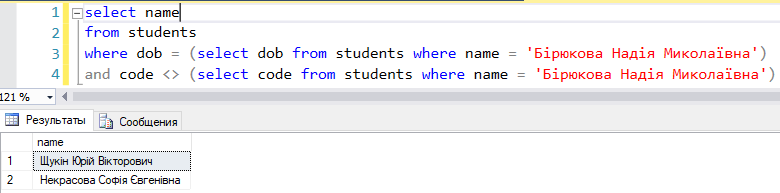


Рис.4. Вкладений запит із умовою на нерівність

Порівняння на рівність та нерівність доступні лише у випадку повернення вкладеним підзапитом 1 рядку з 1 полем, і тому не рекомендується використовувати. Для виберемо групи, у яких навчаються студенти із прізвищем «Іванов» із використанням оператора «in» (рис.5)

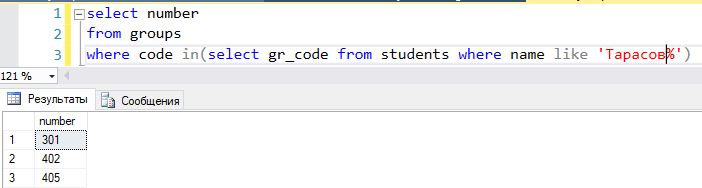


Рис. 5. Групи, у яких навчаються студенти із прізвищем «Тарасов»

Розглянемо використання операторів all та any при роботі із вкладеними запитами. Наприклад виберемо студентів групи 301, що старші за всіх студентів групи 302 (тобто старше найстаршого) (рис.6)

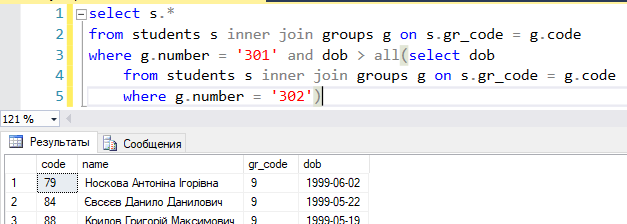


Рис.6. Використання all.

Можна звернути увагу, що даний приклад може бути модифіковано із використанням функції агрегації max замість оператора all (рис.7)

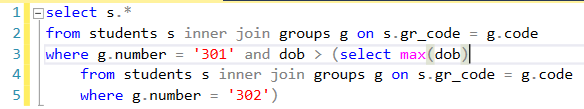


Рис.7. Використання max замість all.

Зрозуміло, що використання знаку рівності із all не має сенсу у випадку повернення вкладеним запитом більше, ніж одного рядку. Чого однак не можна сказати про оператор any. Реалізуємо приклад із використанням any, що вибирає студентів 301 групи, які народились в один і той день із студентами інших груп (рис.8)

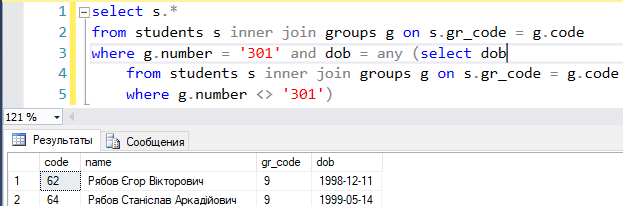


Рис.8. Використання оператора any

Даний приклад також може бути реалізовано із використанням оператора in замість any (рис.9)

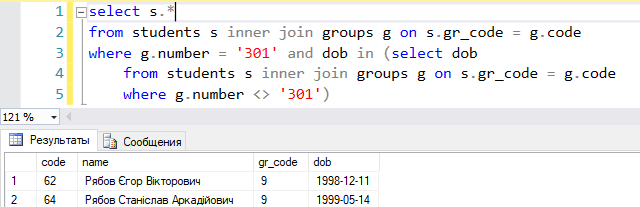


Рис.9. Використання in замість any

1. Далі розглянемо поняття корельованих запитів. Корельовані вкладені запити є такими, що посилаються на атрибути таблиць, вказаних у зовнішньому запиті. Розглянемо наступний приклад, що вибирає студентів, із віком вище середнього у своїй групі (рис.10). При реалізації даного прикладу довелось використати різницю у днях між датою народження та поточною датою (datediff(dd,dob,getdate()), оскільки T-SQL не дозволяє використовувати функцію агрегації avg для типу date.

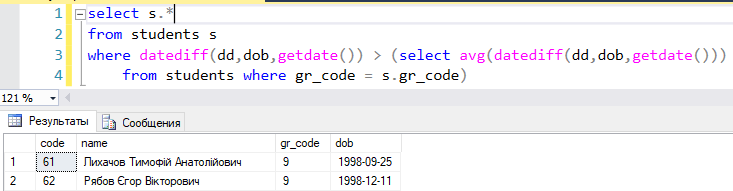


Рис.10. Студенти із віком вище середнього у своїй групі

Наведемо ще один приклад корельованого запиту у секції having, та повертає групи, у яких кількість студентів із датою народження навесні перевищує кількість студентів із датою народження восени (рис.11)

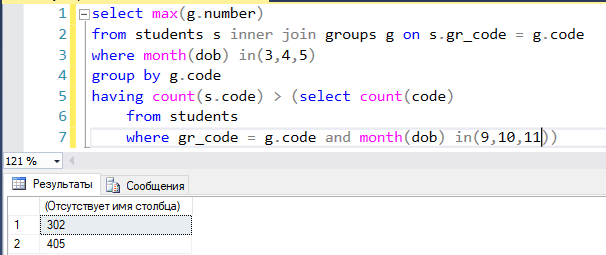


Рис.11. Корельований запит у реченні having

Розглянемо роботу оператора exists, що також часто використовується із корельованими запитами. Оператор приймає у якості параметра множину, та повертає істину, якщо вона не порожня. Інакше повертає хибність. Розглянемо приклад, який повертає назви спеціальностей, де є студенти, але їх менше 30 (рис.12).

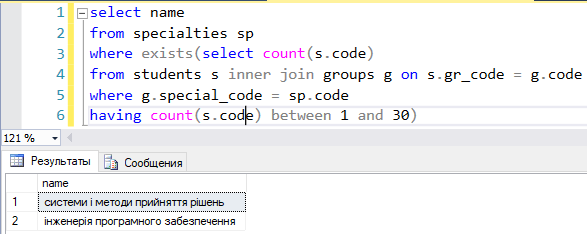


Рис.12. Приклад використання exists у корельованих запитах

1. Наступним кроком розберемо вкладені у реченні from, або як їх ще називають, табличні вираження. Почнемо із простого прикладу вибору студентів групи 301, що народилися у вересні (рис.13).

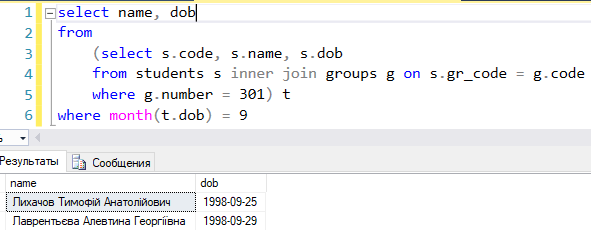


Рис.13. Приклад простого запиту із використанням табличного виразу

Але даний приклад міг бути легко реалізований із поєднанням 2-х умов у реченні where за допомогою and. Реалізуємо дещо цікавіший та складний приклад, що студентські групи, в яких середній бал перевищує середній бал у 2018 році був вищим за середній бал у 2017 році (рис.14)

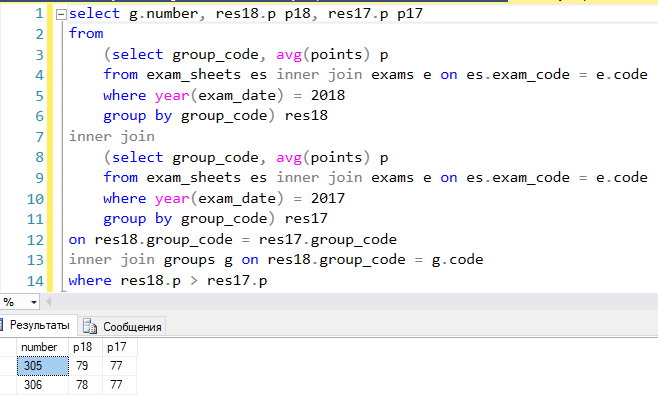


Рис.14. Приклад поєднання табличних виразів

1. І, нарешті розглянемо використання корельованих вкладених запитів у реченні select. Виведемо викладачів та кафедри, до яких вони належать, що прийняли більше 10 іспитів (рис.15)

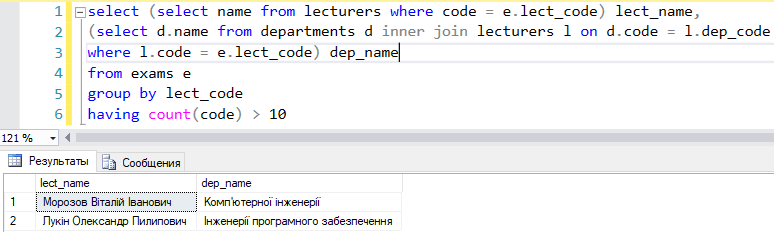


Рис.15. Корельований вкладений запитів у реченні select

Слід пам’ятати, що практично будь-яка задача може бути вирішена більш, ніж в один спосіб, і попередній приклад також може бути реалізований по іншому (рис. 16)

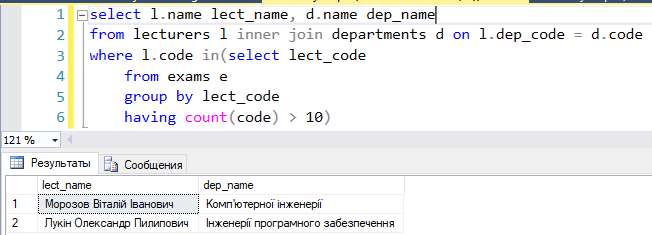


Рис.16. Альтернативна реалізація прикладу рис.46

1. В T-SQL всередині запиту також можуть бути використані змінні, що були попередньо об’явлені та проініціалізовані. Розглянемо приклад вибірки студентів деякої групи із середнім балом, вище вказаного значення (рис.17)

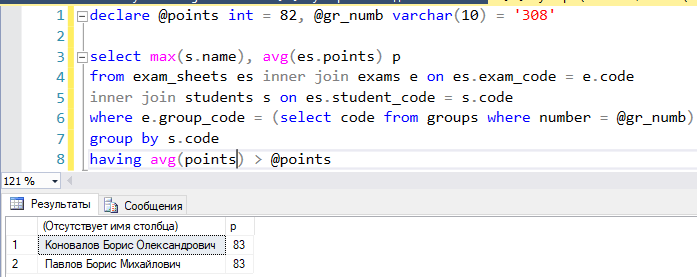


Рис.17. Використання змінних у запитах

Також змінні типу «таблиця» можуть бути використані для збереження проміжних результатів табличних виразів. Розглянемо приклад, що виведе середній бал за викладачем (викладач, дисципліна та середній бал) з дисциплін, що викладаються двома та більше викладачами (ри.18)

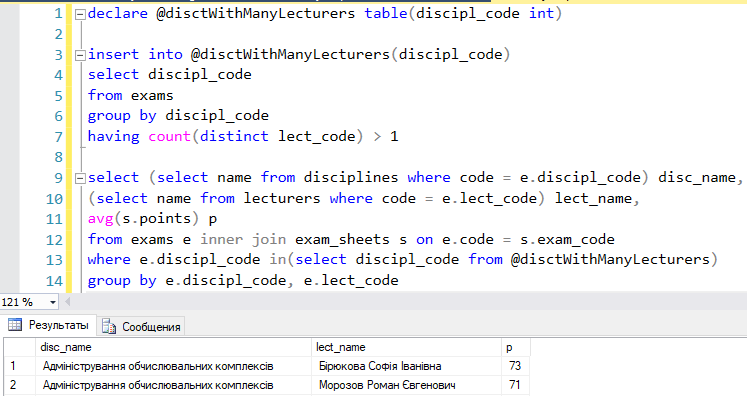


Рис.18. Використання змінних типу «таблиця»

Даний приклад може бути також реалізований із використанням тимчасової таблиці (рис.19).

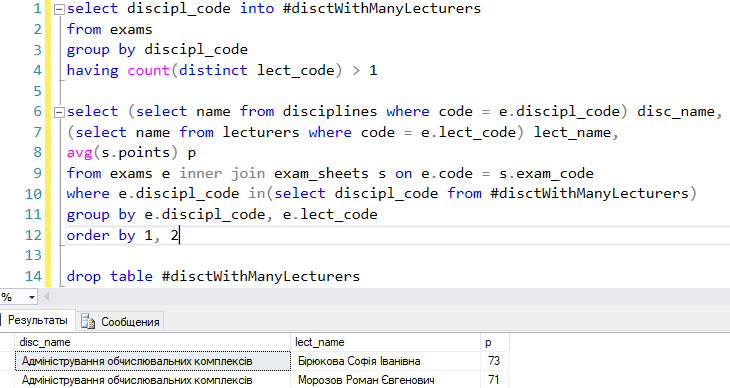


Рис.19. Тимчасові таблиці

Ще один варіант – використання узагальнених табличних виражень із оператором with (рис.20).

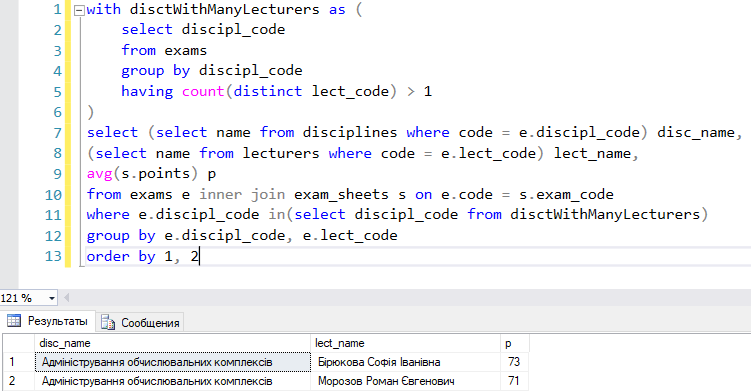


Рис.20. Узагальнені табличні вираження

1. Також запит може бути збережений на постійній основі у вигляді представлень, до яких у подальшому можна буде звертатись для вибіки даних (а іноді і для зміни), як до звичайних таблиць. Розглянемо створення та використання представлення, що зберігає відомості про те, хто не склав іспит. Використаємо його для отримання кількості боржників в аналітиці викладачів (рис.21-22)

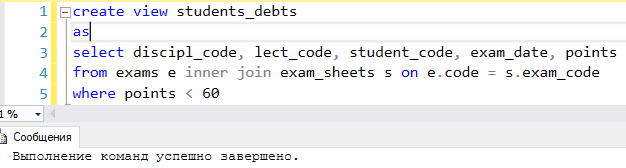


Рис. 21. Створення представлення

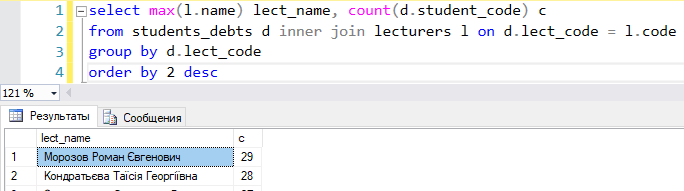


Рис.22. Використання створеного представлення

1. Кінець роботи

**Завдання для індивідуального виконання.**

1. Виконати додаткові запити до власної БД (придумати самостійно), використовуючи наступні засоби (при здачі має бути наведено текст запиту, результат та словесний опис, які саме дані вибирає запит):

* Вкладені запити у реченні where або having
* Вкладені запити у реченні from (табличні вираження)
* Вкладені запити у реченні select
* Змінні;
* Тимчасові таблиці;
* Узагальнені табличні вираження.

1. Створити представлення, що звертається мінімум до 2-х таблиць та використати його у запиті, пов’язавши (join) із іншою таблицею, або із використанням вкладеного запиту. При створенні або використанні представлення використати групування та/або функцію агрегації (при здачі має бути наведено текст запиту, результат та словесний опис, які саме дані вибирає запит).

! УВАГА. Запит не обов’язково має повертати дані. Оцінюється правильність тексту запиту, а не наповненість БД даними (хоча мінімальні вимоги щодо наповненості БД із попередньої роботи все ж мають бути виконані)