**Лабораторна робота 9.**

**Послідовності інструкцій, управляючі конструкції, пакети команд та спроби.**

**Мета роботи:** познайомиться із можливістю виконання послідовності команд у межах одного запиту та поняттям пакетів команд і спроб. Навчитись динамічно формувати та виконувати код T-SQL.

**Задачі:**

* Об’явлення та ініціалізація змінних;
* Використання управляючих конструкцій begin end, if else та while;
* Робота із пакетами команд та інструкцією go
* Використання try catch
* Використання execute

1. Код, реалізований на T-SQL може містити одну інструкцію, або команду, як це було у більшості попередньо розглянутих прикладів, але також може складатись із якоїсь послідовності команд. Коли ж робота програми включає у себе декілька етапів у більшості випадків ці етапи мають бути пов’язаними між собою. Зробити це дозволяють змінні. У T-SQL ім’я змінної обов’язково починається із символу @. Об’явлення змінної виконується інструкцією declare, а присвоїти значення можна за допомогою команд set або select. На рис.1 наведено декілька варіантів об’явлення та ініціалізації змінних із подальшим виведенням на екран за допомогою команди print.

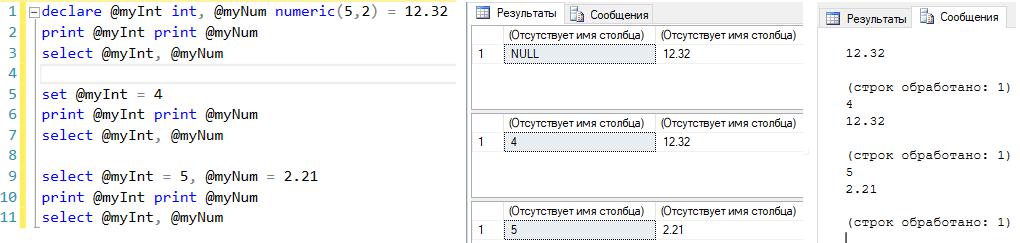


Рис.1. Об’явлення та запис значень у змінну

Даний приклад ілюструє декілька додаткових моментів. По-перше, змінна може бути проініціалізована відразу після об’явлення. Змінна, у яку ще не записано значення, містить NULL. Змінна може бути виведена в область повідомлень за допомогою команди print, або у область результату із використанням команди select. Запис значень у змінні може бути виконано за допомогою інструкцій set або select, при чому остання може виконувати запис одразу в декілька змінних.

Команда select також дозволяє записати до змінної результат роботи вибірки даних. Наприклад запишимо до змінної та виведемо у область повідомлень кількість студентів 308 групи (рис.2)

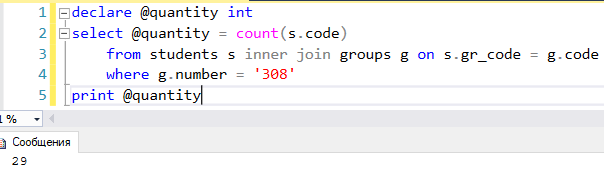


Рис.2. Запис до змінною результату роботи запиту

Якщо результат запиту містить більше одного рядку, у змінну буде записано останнє значення (рис.3)

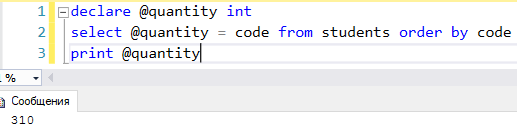


Рис.3. Запис у змінну останнього рядку результуючого набору

Також, можна записувати значення одразу у декілька змінних, якщо результуючий набір має декілька колонок. Так, запишимо у змінні мінімальний та максимальний бал із «» у групі 308 (рис.4)

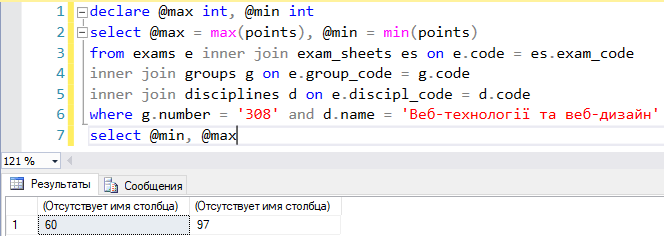


Рис.4. Запис декількох змінних із результуючого набору

Якщо ж запит не повертає даних, то змінна не зазнає жодних змін. Якщо змінна не була перед цим ініціалізована, її значення буде NULL.

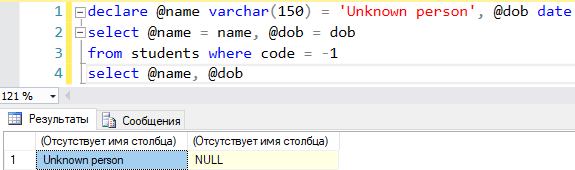


Рис.5. Запис до змінних полів пустого результуючого набору

1. Розглянемо приклади використання умовного оператора if else. Виведемо повідомлення «у групі є вільні місця», якщо у групі < 28 студентів, інакше виведемо «у групі немає вільних місць» (рис.6)

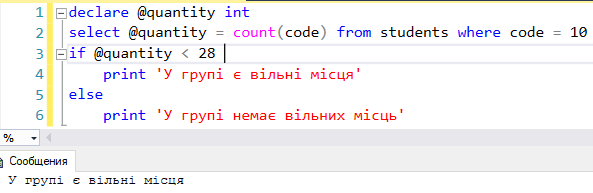


Рис.6. Використання умовного оператора

Якщо потрібно, щоб за умовою виконувалося декілька операторів, використовуємо конструкцію begin…end. Реалізуємо приклад, що груп із кількістю студентів > 20 виведе, що студентів більше 20 на додасть назву факультету, інакше виведе, що студентів менше 20 на додасть назву спеціальності (рис.7).

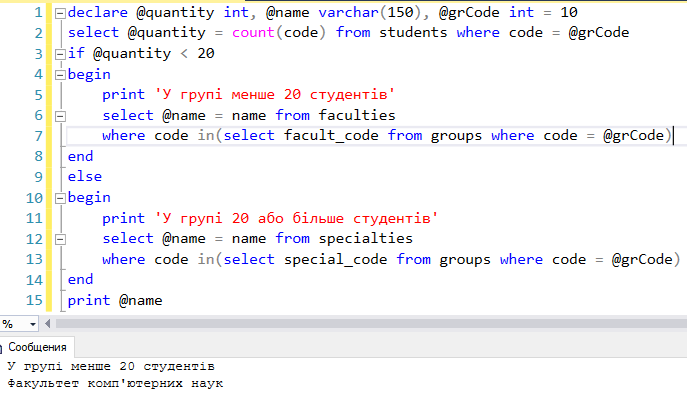


Рис.7. Використання begin…end

1. Розберемо роботу циклів. У наявності T-SQL є лише цикл з передумовою while (але це не є проблемою, оскільки будь який інший тип циклу може бути реалізований на його основі. Також доступні команди break та continue. Реалізуємо підрахунок суми додатніх чисел від 1 до 100 (рис.8)

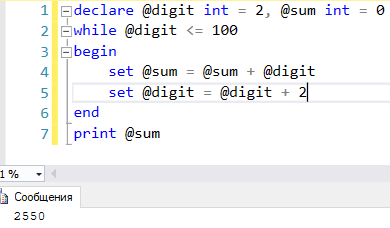


Рис.20. Приклад використання циклу

Реалізуємо той самий приклад із використанням оператора continue (рис.21)

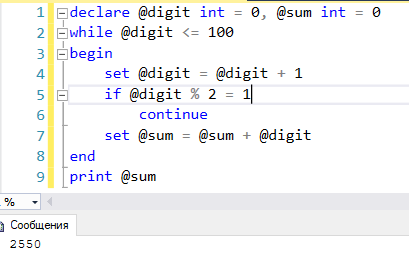


Рис.21. Приклад використання continue

Реалізуємо попередній приклад цикл с пост умовою, використавши while та break (рис.22)

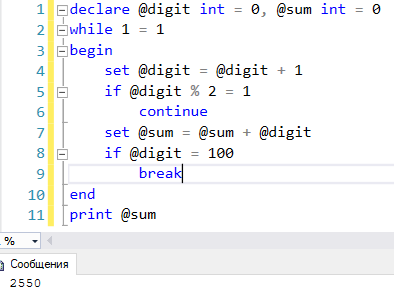


Рис.22. Приклад використання break

1. Розберемо поняття пакету команд в T-SQL. Коли в одному контексті використана послідовність команд, вона може бути розбита на окремі пакети. По перше, дані команди пакети будуть відправлятись не сервер порціями, відповідно до того, як були розбиті на пакети. Є деякий набір команд, що не можуть виконуватись сумісно із іншими, наприклад команда create batabase. Тому, якщо в одному коді ви хочете створити БД, а потім до неї звернутися, команда створення має бути винесена у окремий пакет. Також пакет визначає область видимості змінних. Виконаємо приклад, що вибирає кількість студенів та виводить її на екран (рис.23). Маємо помилку, оскільки змінна визначається у першому пакеті, а використовується у другому

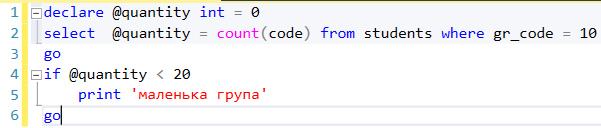


Рис.23. Приклад помилки при роботі із пакетами команд.

Інструкція go має необов’язковий параметр, що визначає кількість разів, що буде виконано пакет. Проілюструємо це на прикладі, та декілька разів випадково обиремо місяць року і визначимо кількість іменинників у цьому місяці (рис.24)

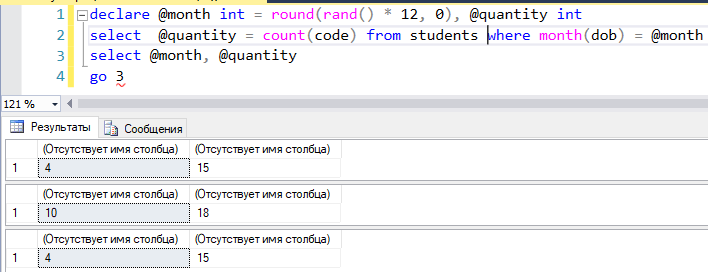


Рис.25. Багаторазове виконання пакету

1. Бувають ситуації, коли спочатку невідомо, який саме код на вибірку (або модифікацію) даних має бути виконано. В T-SQL на цей випадок є також механізми для виконання динамічного сформованого коду. Для цього ми попередньо об’являємо строкову змінну, записуємо у неї код, що має бути виконано код може бути різним в залежності від тих чи інших умов), та виконуємо його за допомогою оператора exec. Нижче розглянемо приклад, що вибирає список студентів факультету, якщо їх менше 100, а інакше вибирає список груп цього факультету із кількістю студентів у групі (рис.26)

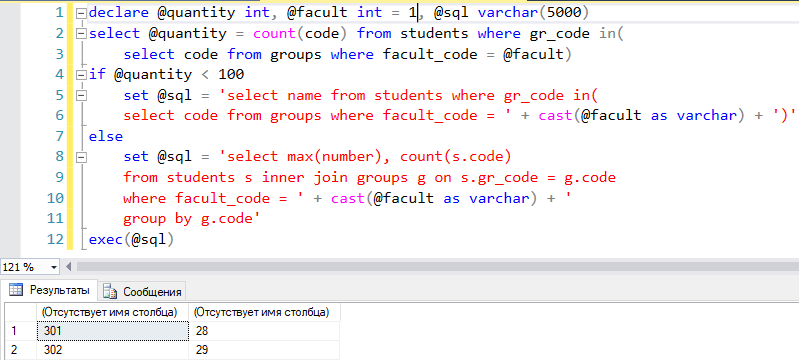


Рис. 26. Виконання динамічно сформованого коду за допомогою exec

1. T-SQL, як і більшість інших мов програмування, має у своєму розпорядженні оператор управління виконанням try … catch. У якості першого прикладу промоделюємо помилку ділення на 0 (рис.27). В області результату маємо повідомлення «division by zero», що повертається з блоку catch, а в області повідомлень бачимо реальну помилку, що повернулася від команди, що виконується за межами try.

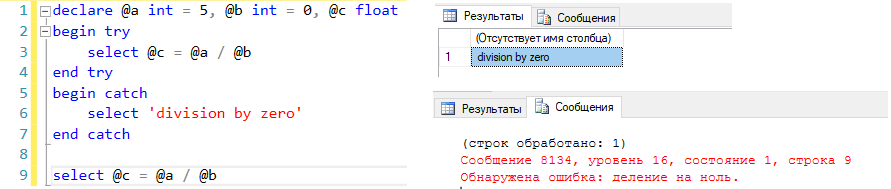


Рис.27. Помилка ділення на 0.

Розглянемо ще один більш цікавий приклад, що вибирає дані із таблиці, назва якої задається динамічно. У даному випадку є ймовірність помилки у імені таблиці. Виконаємо обробку даної ситуації (рис.28)

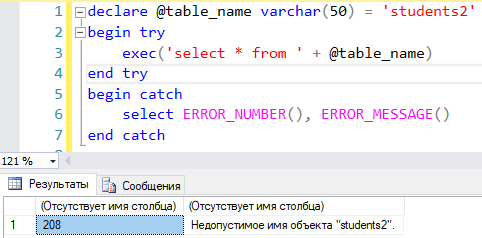


Рис.28. Обробка помилки у назві таблиці

1. Кінець роботи

**Завдання для індивідуального виконання.**

У власній БД виконати набори команд, що виконують наступні дії:

* Об’явлення змінної, її ініціалізація скалярним значенням за допомогою команд set та select та вивід на екран в область результату та в область повідомлень;
* Об’явлення змінної, її ініціалізація результатом роботи запиту та виведення на екран;
* Реалізувати використання умовного оператора (із використанням begin, end та обробкою else) та циклу
* Продемонструвати роботу оператора управління виконанням try … catch
* Реалізувати приклад із динамічним формуванням SQL-запиту із його подальшим виконанням