

## Лабораторна робота №5

### Підзапити. Табличні вирази

#### 1. Мета роботи

1. Отримати навички написання запитів із використанням підзапитів мови SQL.
2. Pozнайомитися з табличними виразами, їх різновидами, вивчити сферу застосування табличних виразів.

#### 2. Порядок виконання роботи

1. Усі запити лабораторної роботи № 5 мають бути виконані до бази даних, розробленої у лабораторній роботі № 2.

2. Написати SQL-скрипт «Subqueries.sql», що складається з безлічі команд SELECT, що демонструють використання наступних операторів та конструкцій (написати мінімум по одній команді SELECT для кожної конструкції):

- 1.1. Self-contained subquery
  - 1.1.1. single-valued
  - 1.1.2. multivalued (використати логічні оператори IN, SOME, ANY, ALL)
- 1.2. Correlated subquery
  - 1.2.1. single-valued
  - 1.2.2. multivalued (використати логічні оператори IN, SOME, ANY, ALL)
- 1.3. EXISTS
- 1.4. Derived Tables
  - 1.4.1. Nested derived tables
  - 1.4.2. Multiple references to the derived table
- 1.5. Common Table Expressions (CTEs)
  - 1.5.1. Multiple CTEs
  - 1.5.2. Multiple References to the same CTE
  - 1.5.3. Recursive CTEs
- 1.6. Views (використати опції ENCRYPTION, SCHEMABINDING, CHECK OPTION)
- 1.7. Inline Table-Valued Functions (inline TVFs)
- 1.8. APPLY Operator
  - 1.8.1. CROSS APPLY
  - 1.8.2. OUTER APPLY

3. Запити зі скрипту «Subqueries.sql» мають бути осмисленими, мати цінність для користувача бази даних. Функціональність кожного запиту має бути описана у скрипті у формі коментаря, що передує відповідній команді.

4. Підготувати звіт. Підготуватися до захисту контрольних питань та вправ.

#### 3. Склад звіту

1. Діаграма бази даних із середовища SQL Server Management Studio.
2. Файл " Subqueries.sql".
3. Аналіз результатів та висновки.

#### 4. Контрольні питання

1. Яка різниця між Self-contained та Correlated підзапитами?
2. Що таке single-valued, multivalued та table-valued підзапити?
3. Як працюють логічні оператори IN, SOME, ANY, ALL?

4. Яка різниця між використанням у складних запитах оператора JOIN та підзапитів? Що рекомендується використовувати оператор JOIN або підзапит? Чому?
5. Яка логіка роботи предикату EXISTS? Якій формі предикату EXISTS віддати перевагу EXISTS(SELECT \* FROM ...) чи EXISTS(SELECT 1 FROM ...)? Чому?
6. Як реалізувати запит, який повертає «попереднє» або «наступне» значення обраного атрибуту таблиці у кожному рядку вихідного набору запиту?
7. Як реалізувати запит, який повертає значення обраного атрибуту таблиці, що наростає, у кожному рядку вихідного набору запиту?
8. Якщо підзапит може повертати значення NULL, а результати підзапиту оцінюються у зовнішньому запиті за допомогою операторів IN, SOME, ANY, ALL, які логічні помилки, викликані тризначною логікою предикатів, можуть виникнути при обробці запиту? Як виправити ці помилки?
9. Чому при складанні підзапитів рекомендується імена атрибутів передувати псевдонімом (alias) таблиці, що містить цей атрибут?
10. Що називається табличним виразом (table expression)? З якою метою застосовуються табличні вирази? Як використання табличних виразів впливає на продуктивність запитів, у яких вони використовуються? Назвіть три основні вимоги до запитів, що формують табличні вирази.
11. Що таке derived tables? Яка область видимості derived table? Які переваги та недоліки використання Inline та External форм визначення псевдонімів стовпців derived table?
12. Що таке Nested derived tables та які недоліки їх використання? Що таке Multiple references to the derived table та які складності виникають при їх використанні?
13. Що таке Common Table Expression (CTE)? Який синтаксис запиту, який використовує CTE? Які вимоги висуваються до запиту CTE? Яка область видимості CTE?
14. Які переваги та недоліки використання Inline та External форм визначення псевдонімів стовпців CTE? Що таке Multiple CTE? Які переваги та недоліки використання множинних посилань (multiple references) на той самий CTE?
15. Що таке Recursive CTE і навіщо він може бути використаний? Яка структура та логіка виконання запиту, який використовує Recursive CTE? Які умови завершення Recursive CTE Ви знаєте? Якою кількістю рекурсивних викликів обмежено виконання Recursive CTE за замовчуванням? Як задати нескінченну кількість викликів рекурсивної частини CTE та які наслідки такого виклику?
16. Що таке View і навіщо воно може бути використане? Які переваги використання View порівняно з derived tables та CTE? Яка область видимості View? Чому не рекомендується використовувати «...SELECT \*...» для визначення View? Для чого використовуються процедури sp\_refreshview та sp\_refreshsqlmodule?
17. Які вимоги висуваються до запиту, що формує View? Яке призначення опцій ENCRYPTION, SCHEMABINDING, CHECK OPTION, які застосовуються під час створення View?
18. Що таке Inline Table-Valued Function (TVF) і навіщо вона може бути використана? Які переваги використання TVF у порівнянні з derived tables, CTE та View? Яка область видимості TVF?
19. Навіщо застосовується оператор APPLY? Як працює CROSS APPLY та OUTER APPLY? У чому подібність і які відмінності у роботі операторів APPLY та JOIN? Як працює оператор LATERAL та які його відмінності від оператора APPLY?

## 5. Література

1. [Subqueries \(SQL Server\) - SQL Server | Microsoft Learn](#)