**ЛАБОРАТОРНА РОБОТА №15**

**ФУНКЦІЇ РАНЖУВАННЯ В T-SQL**

Метою лабораторної роботи є вивчення функцій ранжування в T-SQL (ROW\_NUMBER, RANK, DENSE\_RANK, NTILE) для аналізу та обробки даних. Студенти навчаться застосовувати PARTITION BY і ORDER BY для нумерації, класифікації та групування рядків, створювати запити для вибору топ-N записів, розподілу даних на групи та виконання статистичного аналізу. Очікуваним результатом є здатність використовувати функції ранжування для вирішення прикладних задач і оптимізації запитів.

**ТЕОРЕТИЧНІ ОСНОВИ**

Функції ранжування в Transact-SQL (T-SQL) використовуються для роботи з наборами даних, дозволяючи присвоювати кожному рядку унікальний або груповий номер у рамках певної логіки. Ці функції корисні для нумерації, класифікації та розподілу даних. До основних функцій ранжування належать:

1. **ROW\_NUMBER**
2. **RANK**
3. **DENSE\_RANK**
4. **NTILE**

#### 1. ROW\_NUMBER

Функція ROW\_NUMBER() присвоює кожному рядку унікальний номер у межах заданої логіки сортування. Якщо рядки мають однакове значення в колонках, які використовуються для сортування, вони отримують різні номери.

##### Синтаксис:

ROW\_NUMBER() OVER ([PARTITION BY <column\_list>] ORDER BY <order\_list>)

* **PARTITION BY**: необов'язковий параметр, який розбиває результуючий набір даних на частини.
* **ORDER BY**: обов'язковий параметр, що визначає порядок сортування.

##### Приклад 1: Нумерація всіх рядків

SELECT \*,

ROW\_NUMBER() OVER (ORDER BY AMOUNT ASC) AS RowNumber

FROM [dbo].[ORDERS];

У цьому прикладі кожен рядок у таблиці ORDERS отримує унікальний номер, упорядкований за зростанням AMOUNT.

##### Приклад 2: Нумерація в межах групи

SELECT \*,

ROW\_NUMBER() OVER (PARTITION BY CUST ORDER BY AMOUNT DESC) AS RowNumber

FROM [dbo].[ORDERS];

У цьому прикладі кожен клієнт (CUST) отримує свої номери замовлень, упорядкованих за спаданням AMOUNT.

#### 2. RANK

Функція RANK() призначає номер кожному рядку на основі порядку. Якщо рядки мають однакове значення, вони отримують однаковий ранг, але наступний номер пропускається.

##### Синтаксис:

RANK() OVER ([PARTITION BY <column\_list>] ORDER BY <order\_list>)

##### Приклад:

SELECT \*,

RANK() OVER (ORDER BY AMOUNT DESC) AS Rank

FROM [dbo].[ORDERS];

Якщо два замовлення мають однакову суму (AMOUNT), вони отримують однаковий ранг, а наступний ранг пропускається.

#### 3. DENSE\_RANK

DENSE\_RANK() працює аналогічно до RANK(), але без пропуску номерів. Ранги залишаються послідовними.

##### Синтаксис:

DENSE\_RANK() OVER ([PARTITION BY <column\_list>] ORDER BY <order\_list>)

##### Приклад:

SELECT \*,

DENSE\_RANK() OVER (ORDER BY AMOUNT DESC) AS DenseRank

FROM [dbo].[ORDERS];

Якщо два замовлення мають однакову суму, вони отримують однаковий ранг, але наступний номер буде без пропуску.

#### 4. NTILE

Функція NTILE() розподіляє рядки набору даних на вказану кількість груп (n-тайлів).

##### Синтаксис:

NTILE(<number\_of\_groups>) OVER ([PARTITION BY <column\_list>] ORDER BY <order\_list>)

* **<number\_of\_groups>**: кількість груп для розподілу.

##### Приклад:

SELECT \*,

NTILE(4) OVER (ORDER BY AMOUNT ASC) AS Quartile

FROM [dbo].[ORDERS];

У цьому прикладі функція розподіляє рядки на 4 групи на основі значень AMOUNT.

### Практичні приклади використання функцій ранжування

1. **Вибір трьох останніх замовлень для кожного клієнта:**

SELECT \*

FROM (

SELECT \*,

DENSE\_RANK() OVER (PARTITION BY CUST ORDER BY ORDER\_DATE DESC) AS DenseRank

FROM [dbo].[ORDERS]

) AS Subquery

WHERE DenseRank <= 3;

1. **Розподіл замовлень на три рівні групи за розміром суми:**

SELECT \*,

NTILE(3) OVER (ORDER BY AMOUNT DESC) AS RankGroup

FROM [dbo].[ORDERS];

1. **Пронумерувати співробітників за датою найму у межах посад:**

SELECT \*,

ROW\_NUMBER() OVER (PARTITION BY TITLE ORDER BY HIRE\_DATE DESC) AS RowNumber

FROM [dbo].[SALESREPS];

### Порівняння функцій

| **Функція** | **Опис** | **Пропуск номерів** |
| --- | --- | --- |
| ROW\_NUMBER | Унікальна нумерація рядків | Так |
| RANK | Ранжування із пропуском номерів | Так |
| DENSE\_RANK | Ранжування без пропуску номерів | Ні |
| NTILE | Розподіл рядків на рівні групи | Не застосовується |

**ЗАВДАННЯ**

***Завдання 1***

Напишіть запит, який поверне список ідентифікаторів виробників та ідентифікаторів товарів без дублікатів.

Для відсікання дублікатів необхідно задіяти функцію ранжування ROW\_NUMBER.

- Використовується таблиця ORDERS

- Не використовуйте оператор DISTINCT

- Результуючий набір даних містить: ідентифікатори виробників та ідентифікатори товарів

***Завдання 2***

Напишіть запит, що повертає список замовлень на товари, які продавалися більше одного разу (за кількістю замовлень на <MFR>, <PRODUCT>).

- Використовується таблиця ORDERS

- Задійте функції ранжування для визначення товарів, які продавалися більше одного разу

***Завдання 3***

Напишіть запит, який для кожної окремої посади поверне найбільш досвідченого працівника (за кількістю відпрацьованих років).

Враховуйте ймовірність того, що одразу кілька працівників мають однаковий стаж.

Розрахуйте кількість відпрацьованих років за допомогою функції DATEDIFF.

Визначте найбільш досвідчених працівників для кожної окремої посади за допомогою функції ранжування DENSE\_RANK.

- Використовується таблиця [dbo].[SALESREPS]

**КОНТРОЛЬНІ ЗАПИТАННЯ**

1. **Що таке функції ранжування в T-SQL і які основні задачі вони вирішують?**
2. **Яка різниця між функціями** ROW\_NUMBER**,** RANK **і** DENSE\_RANK**?**
3. **Як використовується ключове слово** PARTITION BY **у функціях ранжування? Наведіть приклад.**
4. **Яка функція дозволяє розділити набір даних на рівні групи? Опишіть її принцип роботи.**
5. **Чим** ROW\_NUMBER **відрізняється від** RANK **при однакових значеннях в колонці сортування?**
6. **Який результат поверне функція** DENSE\_RANK**, якщо декілька рядків мають однакове значення?**
7. **Як функція** NTILE **визначає, до якої групи належить рядок?**
8. **Наведіть приклад запиту, який використовує** ROW\_NUMBER **для нумерації рядків у всьому наборі даних.**
9. **Як можна отримати перші три замовлення кожного клієнта, використовуючи функції ранжування?**
10. **У чому різниця між функціями** RANK **та** DENSE\_RANK **у випадку однакових значень?**
11. **Як за допомогою функції** NTILE **можна поділити дані на квартилі?**
12. **Чи можна використовувати функції ранжування разом із умовою** HAVING**? Чому?**
13. **Що відбудеться, якщо пропустити параметр** ORDER BY **у функції ранжування?**
14. **Як функції ранжування впливають на продуктивність запиту?**
15. **Чи можна застосовувати функції ранжування до кількох колонок одночасно? Наведіть приклад.**

**ДОДАТКОВІ МАТЕРІАЛИ:**

<https://learn.microsoft.com/en-us/sql/t-sql/functions/ranking-functions-transact-sql?view=sql-server-ver16>

ROW\_NUMBER

<https://learn.microsoft.com/en-us/sql/t-sql/functions/rank-transact-sql?view=sql-server-ver16>

DENSE\_RANK

<https://learn.microsoft.com/en-us/sql/t-sql/functions/dense-rank-transact-sql?view=sql-server-ver16>

RANK

<https://learn.microsoft.com/en-us/sql/t-sql/functions/rank-transact-sql?view=sql-server-ver16>

NTILE

<https://learn.microsoft.com/en-us/sql/t-sql/functions/ntile-transact-sql?view=sql-server-ver16>