**ЛАБОРАТОРНА РОБОТА №9**

МАТЕМАТИЧНІ ФУНКЦІЇ

*Мета:* основна мета лабораторної роботи полягає в ознайомленні студентів із широким спектром математичних функцій, доступних у сучасних системах управління базами даних (СУБД), та розвитку навичок їх застосування для аналізу та обробки даних. Лабораторна робота спрямована на практичне вивчення методів використання математичних функцій для розв'язання типових завдань, з якими студенти можуть зіткнутися в процесі роботи з базами даних в контексті аналітики, фінансових обчислень, обробки наукових даних та інших сфер діяльності.

Конкретні цілі лабораторної роботи включають:

1. Вивчення основних математичних функцій: Ознайомлення з базовими математичними операціями, такими як додавання, віднімання, множення, ділення, а також більш складними функціями, включаючи обчислення степенів, коренів, логарифмів, тригонометричних функцій і т.д.

2. Застосування математичних функцій для обробки даних: Розробка запитів до бази даних, які використовують математичні функції для обчислення агрегованих показників, аналізу трендів, визначення статистичних характеристик наборів даних.

3. Аналіз результатів застосування математичних функцій: Оцінка точності та ефективності використання різних математичних функцій в контексті обробки даних, виявлення потенційних помилок та неточностей у розрахунках.

4. Розуміння принципів вибору відповідних математичних функцій: Формування у студентів умінь обирати оптимальні математичні функції в залежності від специфіки завдань та доступних даних.

Лабораторна робота допоможе студентам розширити свої знання та практичні навички у використанні математичних функцій для ефективної роботи з базами даних. Вона сприятиме формуванню глибокого розуміння значення математичних обчислень у процесі аналізу та обробки даних, а також підготує студентів до вирішення складних аналітичних задач у майбутній професійній діяльності.

**ТЕОРЕТИЧНІ ОСНОВИ**

Математичні функції та оператори в SQL дозволяють виконувати різноманітні обчислення над даними, що зберігаються в базі даних. Вони є незамінними інструментами для аналізу, обробки та трансформації даних. Далі розглянемо ключові математичні функції та оператори, що часто застосовуються в SQL.

*Математичні функції*

1. ABS(x): Повертає абсолютне значення числа `x`.

- Приклад:

`SELECT ABS(-123.45) AS AbsoluteValue;` поверне `123.45`.

2. CEILING(x): Округляє число `x` до найближчого більшого або рівного цілого числа.

- Приклад:

`SELECT CEILING(123.01) AS CeilingValue;` поверне `124`.

3. FLOOR(x): Округляє число `x` до найближчого меншого або рівного цілого числа.

- Приклад:

`SELECT FLOOR(123.99) AS FloorValue;` поверне `123`.

4. PI(): Повертає значення числа π.

- Приклад:

`SELECT PI() AS PiValue;` поверне приблизно `3.141592653589793`.

5. POWER(x, y): Підносить число `x` до степеня `y`.

- Приклад:

`SELECT POWER(2, 3) AS PowerValue;` поверне `8`.

6. ROUND(x, d): Округляє число `x` до `d` десяткових знаків.

- Приклад:

`SELECT ROUND(123.4567, 2) AS RoundedValue;` поверне `123.46`.

*Математичні оператори*

1. + (Add): Додавання двох чисел.

- Приклад:

`SELECT 2 + 3 AS Addition;` поверне `5`.

2. - (Subtract): Віднімання одного числа від іншого.

- Приклад:

`SELECT 5 - 3 AS Subtraction;` поверне `2`.

3. \* (Multiply): Множення двох чисел.

- Приклад:

`SELECT 2 \* 3 AS Multiplication;` поверне `6`.

4. / (Divide): Ділення одного числа на інше.

- Приклад:

`SELECT 6 / 3 AS Division;` поверне `2`.

5. % (Modulo): Повертає залишок від ділення двох чисел.

- Приклад:

`SELECT 10 % 3 AS Modulo;` поверне `1`.

*Функції перетворення типів*

1. CAST(x AS type): Перетворює вираз `x` в тип `type`.

- Приклад:

`SELECT CAST('123.45' AS DECIMAL(5,2)) AS CastValue;` поверне `123.45`.

2. TRY\_CAST(x AS type): Схоже на `CAST`, але повертає `NULL`, якщо перетворення неможливе.

- Приклад:

`SELECT TRY\_CAST('abc' AS INT) AS TryCastValue;` поверне `NULL`.

3. CONVERT(type, x, style): Перетворює вираз `x` в тип `type`, можливе вказання стилю форматування `style`.

- Приклад:

`SELECT CONVERT(INT, '123') AS ConvertValue;` поверне `123`.

4. TRY\_CONVERT(type, x, style): Схоже на `CONVERT`, але повертає `NULL` при невдачі.

- Приклад:

`SELECT TRY\_CONVERT(INT, 'abc') AS TryConvertValue;` поверне `NULL`.

5. PARSE(x AS type USING culture): Аналізує рядкове представлення `x` і перетворює його на тип `type`, використовуючи культурні налаштування `culture`.

- Приклад:

`SELECT PARSE('1,234.56' AS DECIMAL USING 'en-US') AS ParseValue;` поверне `1234.56`.

6. TRY\_PARSE(x AS type USING culture): Схоже на `PARSE`, але повертає `NULL`, якщо аналіз неможливий.

- Приклад:

`SELECT TRY\_PARSE('not a number' AS INT USING 'en-US') AS TryParseValue;` поверне `NULL`.

Ці функції та оператори формують основу для виконання розрахунків та обробки даних у SQL, дозволяючи розробникам і аналітикам ефективно маніпулювати даними та перетворювати їх для задоволення конкретних бізнес-вимог.

**ЗАВДАННЯ**

***Задача 1***

Напишіть запит, який поверне список унікальних ідентифікаторів виробників.

Враховуйте тільки замовлення з парною сумою замовлення і непарною кількістю замовлених одиниць товару.

- Використовується таблиця [dbo].[ORDERS]

- Результативний набір даних містить: ідентифікатор виробника

- Відсортувати рез. набір даних за ідентифікатором виробника

***Задача 2***

Напишіть запит, який у розрізі ідентифікатора виробника товару поверне загальну вартість од. товару на складі та середню вартість од. товарів. Щоб розрахувати вартість товару на складі, необхідно помножити кількість од. товару на ціну за одиницю.

Округлити результати агрегування до 2 знаків після коми.

- Використовується таблиця [dbo].[PRODUCTS]

- Результуючий набір даних містить: ідентифікатор виробника, загальну суму, середню суму.

- Відсортувати результуючий набір даних за ідентифікатором виробника

***Задача 3***

Напишіть запит, який поверне працівника з найбільшою мат. різницею між ціллю за продажами і сумою поточних продаж, у абсолютному вимірі, тобто необхідно взяти число за модулем. Округлити результати обчислення до двох знаків після коми. Розрахувати кількість років працівника на момент найму.

Враховуйте ймовірність того, що відразу кілька працівників можуть мати одну і ту ж різницю.

- Використовується таблиця [dbo].[SALESREPS]

**ДОДАТКОВІ МАТЕРІАЛИ:**

1. <https://learn.microsoft.com/en-us/sql/t-sql/functions/mathematical-functions-transact-sql?view=sql-server-ver16>
2. <https://learn.microsoft.com/en-us/sql/t-sql/language-elements/arithmetic-operators-transact-sql?view=sql-server-ver16>
3. <https://learn.microsoft.com/ru-ru/sql/t-sql/data-types/data-type-precedence-transact-sql?view=sql-server-ver16>

**КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Які основні математичні функції доступні в SQL, і для яких задач вони зазвичай застосовуються? Обговоріть різноманіття математичних функцій в SQL та їх призначення для обробки та аналізу даних.

2. Наведіть приклад використання функції ABS і поясніть, чому важливо знати абсолютне значення числа при аналізі даних. Продемонструйте ситуації, в яких абсолютне значення може відігравати ключову роль.

3. Як функції CEILING і FLOOR можуть бути застосовані при роботі з фінансовими даними? Проаналізуйте приклади, де округлення чисел вгору або вниз має важливе значення для точності фінансових розрахунків.

4. Опишіть, як можна використовувати функцію PI разом із тригонометричними функціями для вирішення практичних задач. Розгляньте приклади, де знання та застосування тригонометрії є необхідним для обчислення параметрів в реальних проєктах.

5. Для чого використовується функція POWER, і які типові сценарії її застосування? Наведіть приклади, де потрібно піднімати число до степеня для обчислення певних величин або параметрів.

6. Яким чином функція ROUND впливає на точність даних, і коли її використання є критично важливим? Продемонструйте, як неправильне або відсутнє округлення може призвести до помилок у результатах.

7. Як оператори для базових математичних дій (додавання, віднімання, множення, ділення) впливають на обробку та аналіз даних? Розкрийте їх важливість для створення виразів і формул в запитах до бази даних.

8. Поясніть різницю між функціями CAST і CONVERT, наведіть приклади, коли краще використовувати одну замість іншої. Оцініть переваги кожної функції для перетворення типів даних і форматування виводу.

9. Як функції PARSE і TRY\_PARSE можуть бути використані для роботи з текстовими даними? Обговоріть їх застосування для аналізу та перетворення рядкових значень у більш специфічні типи даних.