**ЛАБОРАТОРНА РОБОТА №13**

АГРЕГАТНІ ФУНКЦІЇ

*Мета:* основною метою цієї лабораторної роботи є ознайомлення студентів з різними типами агрегатних функцій та розвиток навичок їх використання для ефективного аналізу даних.

Конкретні цілі лабораторної роботи:

1. Вивчення основних агрегатних функцій: Знайомство студентів з функціями, такими як COUNT, SUM, AVG, MIN, MAX, та іншими, які використовуються для агрегування даних.

2. Застосування агрегатних функцій у запитах SQL: Навчання студентів писати запити, що ефективно використовують агрегатні функції для вирішення практичних задач, таких як підрахунок кількості елементів, визначення загальних та середніх значень, та визначення екстремальних значень у датасетах.

3. Аналіз та інтерпретація результатів: Розвиток у студентів навичок інтерпретації результатів, отриманих за допомогою агрегатних функцій, для розуміння тенденцій та шаблонів в даних.

4. Розуміння впливу групування на результати агрегації: Вивчення того, як оператор GROUP BY може впливати на результати використання агрегатних функцій та як його використання може призводити до більш глибокого аналізу даних.

5. Практичне застосування знань: Виконання завдань на реальних або навчальних датасетах, які дозволяють студентам застосовувати теоретичні знання на практиці та розвивати аналітичні навички.

Ця лабораторна робота надасть студентам необхідний фундамент для розуміння та застосування агрегатних функцій у широкому діапазоні аналітичних завдань, сприятиме розвитку критичного мислення та аналітичного підходу до обробки та інтерпретації даних.

**ТЕОРЕТИЧНІ ОСНОВИ**

У контексті баз даних, агрегатні функції відіграють визначальну роль у сумації та аналізі великих наборів даних. Вони дозволяють виконувати операції, які зводять множину значень до одного показника, що може бути використане для статистичного аналізу, звітності та інших форм обробки даних.

1. Функція COUNT: Повертає кількість елементів у визначеному стовпці або виразі, ігноруючи `NULL` значення.

- Приклад:

`SELECT COUNT(ProductID) FROM Products;` підраховує кількість продуктів.

2. Функція SUM: Обчислює суму значень стовпця, що містить числові дані.

- Приклад:

`SELECT SUM(Quantity) FROM OrderDetails;` підраховує загальну кількість товарів у замовленнях.

3. Функція AVG: Визначає середнє значення по стовпцю, що містить числові дані.

- Приклад:

`SELECT AVG(Price) FROM Products;` обчислює середню ціну продукту.

4. Функції MIN та MAX: Повертають мінімальне та максимальне значення у визначеному стовпці відповідно.

- Приклад:

`SELECT MIN(CreatedDate), MAX(CreatedDate) FROM Orders;` знаходить найранішу та найпізнішу дату замовлення.

5. Функція GROUP BY: Використовується разом з агрегатними функціями для групування результатів за певними критеріями.

- Приклад:

`SELECT CustomerID, COUNT(OrderID) FROM Orders GROUP BY CustomerID;` показує кількість замовлень по кожному клієнту.

Розуміння та вміння правильно застосовувати ці функції відкривають широкі можливості для аналітичної роботи з даними. Студенти навчаться використовувати агрегатні функції для створення звідних таблиць, які можуть служити основою для рішення різних бізнес-задач, таких як підрахунок загального прибутку, визначення середнього часу доставки товарів, аналіз продуктивності продажів тощо. Завдяки глибокому розумінню принципів роботи з агрегатними функціями, студенти зможуть ефективно організовувати та оптимізувати запити до бази даних, що є важливим навиком для будь-якого фахівця у сфері обробки даних.

**ЗАВДАННЯ**

***Завдання 1: Аналіз продажів виробників***

Мета: Використовуючи таблицю `[dbo].[ORDERS]`, створіть SQL-запит, який агрегує загальну суму продажів по кожному виробнику за 2008 рік. Використайте функцію `SUM` для підрахунку загальної суми та `GROUP BY` для групування результатів за виробниками.

- Створіть запит, який виводить ідентифікатор виробника (`MFR`) та загальну суму продажів (`TOTAL\_SALES`) за 2008 рік.

- Використайте функцію `YEAR` для фільтрації дати замовлення.

- Відсортуйте результати за ідентифікатором виробника за зростанням.

***Завдання 2: Статистика робочого стажу працівників***

Мета: Застосуйте агрегатні функції на таблиці `[dbo].[SALESREPS]`, щоб визначити кількість працівників з різним стажем роботи.

- Створіть запит, який групує працівників (`EMP\_NUM`) за кількістю повних років роботи (`YEARS\_OF\_SERVICE`) з використанням `DATEDIFF`.

- Використайте функцію `COUNT` для обчислення кількості працівників у кожній групі.

- Відсортуйте результати за кількістю працівників за спаданням.

**Завдання 3: Аналіз найму працівників за періодами**

Мета: Вивчіть розподіл найму працівників по місяцях та роках, використовуючи таблицю `[dbo].[SALESREPS]` і функції `YEAR`, `MONTH`, `COUNT`, `GROUP BY`.

- Розробіть запит, який показує рік та місяць найму (`HIRE\_YEAR`, `HIRE\_MONTH`) разом з кількістю працівників, яких було найнято в кожен із цих періодів (`HIRED\_COUNT`).

- Використайте `GROUP BY` для групування даних за роком та місяцем найму.

- Відсортуйте результати спочатку за роком, а потім за місяцем за зростанням.

**ДОДАТКОВІ МАТЕРІАЛИ:**

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

**КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Які основні агрегатні функції в SQL ви знаєте та для чого вони використовуються? Охарактеризуйте кожну з них та приклади їх застосування в аналітичній роботі з базами даних.

2. Чим відрізняється використання функції `SUM` від `AVG` та в яких сценаріях кожна з них є найбільш доцільною? Надайте приклади, що ілюструють умови вибору між цими двома функціями.

3. Як агрегатні функції можуть бути використані разом з `GROUP BY` для отримання зведених даних? Продемонструйте на конкретному прикладі, як можна використати цю комбінацію для аналізу даних.

4. Чому необхідно знати різницю між `COUNT(\*)` та `COUNT(column\_name)`? Поясніть, як включення або виключення `NULL` значень може вплинути на результати підрахунку.

5. Як можна використовувати функції `MIN` та `MAX` для роботи з датами? Надайте приклади, коли ці функції можуть бути корисними для виявлення найраніших чи найпізніших дат у наборах даних.

6. Опишіть, як можна використати агрегатні функції для виявлення викидів чи аномалій в даних. Вкажіть, як обробка екстремальних значень може допомогти покращити якість даних.

7. Які проблеми можуть виникнути при використанні агрегатних функцій на великих обсягах даних? Обговоріть потенційні проблеми з продуктивністю та точністю і як їх можна вирішити.