**ЛАБОРАТОРНА РОБОТА №12**

ФУНКЦІЇ ДЛЯ РОБОТИ З ДАТАМИ

*Мета:* основною метою цієї лабораторної роботи є надання студентам глибоких знань та практичних навичок у використанні функцій для роботи з датами, що пропонуються сучасними системами управління базами даних (СУБД). Вивчення цього аспекту є критично важливим, оскільки дати та час відіграють ключову роль у багатьох типах даних і бізнес-процесах, від фінансових транзакцій до логістики та управління персоналом.

Конкретні цілі лабораторної роботи включають:

1. Ознайомлення з функціями для роботи з датами та часом: Надати студентам знання про базові та розширені функції СУБД для обробки дат та часу, включаючи отримання поточної дати та часу, додавання та віднімання інтервалів часу, різницю між датами, форматування дати та часу тощо.

2. Застосування функцій для аналізу даних: Розробка запитів до бази даних, які використовують функції для роботи з датами для вирішення типових задач аналізу даних, таких як визначення віку, розрахунок термінів, аналіз сезонності продажів, відстеження строків дії контрактів тощо.

3. Визначення часових інтервалів: Навчити студентів визначати та працювати з часовими інтервалами, включаючи роботу з функціями, що дозволяють обчислювати періоди між датами, а також додавання чи віднімання періодів часу від заданої дати.

4. Практичне застосування набутих знань: Виконання практичних завдань з використанням функцій для роботи з датами на реальних або навчальних даних з метою закріплення теоретичних знань та розвитку практичних навичок.

Лабораторна робота дозволить студентам не лише освоїти широкий спектр функцій для роботи з датами в SQL, але й зрозуміти їх застосування в контексті реальних бізнес-задач. Володіння цими навичками є необхідним для будь-якого спеціаліста, який працює з базами даних, оскільки воно дозволяє глибше аналізувати та інтерпретувати дані, пов'язані з часом, що є невід'ємною частиною багатьох ділових, наукових та соціальних процесів.

**ТЕОРЕТИЧНІ ОСНОВИ**

Функції для роботи з датами в SQL забезпечують потужний інструментарій для виконання різноманітних операцій з датами та часом, від простого отримання поточної дати до складних обчислень різниці між датами. Розглянемо детальніше ключові функції для роботи з датами.

1. Отримання поточної дати та часу:

- `GETDATE()` та `CURRENT\_TIMESTAMP` повертають поточну дату і час сервера. Хоча `CURRENT\_TIMESTAMP` є аналогом `GETDATE()` і відповідає стандарту ANSI/ISO, обидві функції повертають тип даних `datetime`.

- `GETUTCDATE()` повертає поточну дату і час у форматі UTC-0, що дозволяє працювати з уніфікованим світовим часом.

- `SYSDATETIME()` та `SYSUTCDATETIME()` також повертають поточну дату і час, але з більш високою точністю (тип даних `datetime2`), що корисно для застосувань, де необхідна висока точність вимірювань часу.

- `SYSDATETIMEOFFSET()` повертає поточну дату і час з урахуванням часового пояса (тип даних `datetimeoffset`), що дуже важливо для додатків, що працюють в міжнародному середовищі.

2. Операції з датами:

- `DATEADD(datepart, number, date)` додає до вказаної дати певну кількість днів, місяців, років тощо.

- Приклад:

`SELECT DATEADD(year, 1, '2023-01-01')` поверне `2024-01-01`.

- `DATEDIFF(datepart, startdate, enddate)` повертає різницю між двома датами в роках, місяцях, днях залежно від вказаної частини дати.

- Приклад:

`SELECT DATEDIFF(day, '2023-01-01', '2023-01-31')` поверне `30`.

- `DATENAME(datepart, date)` і `DATEPART(datepart, date)` виокремлюють вказану частину дати і повертають її відповідно у символьному та числовому форматі.

- Приклад:

`SELECT DATENAME(month, '2023-01-01')` поверне `Січень`.

3. Функції для отримання компонентів дати:

- `DAY(date)`, `MONTH(date)`, `YEAR(date)` повертають відповідно день, місяць та рік з вказаної дати.

- Приклад:

`SELECT YEAR('2023-01-01')` поверне `2023`.

4. Перевірка коректності формату дати:

- `ISDATE(expression)` перевіряє, чи є вираз допустимою датою.

- Приклад:

`SELECT ISDATE('2023-02-29')` поверне `0`, оскільки 2023 не є високосним роком.

5. Перетворення типів даних:

- `CAST` та `CONVERT` використовуються для перетворення даних між різними типами. `TRY\_CAST` і `TRY\_CONVERT` надають ту ж функціональність, але повертають `NULL` у разі помилки перетворення.

- `PARSE` і `TRY\_PARSE` спеціалізуються на перетворенні текстових рядків у числові та датові типи з урахуванням локалізації.

**ЗАВДАННЯ**

***Задача 1***

Напишіть запит, який поверне список ідентифікаторів виробників (без дублікатів).

Враховуйте тільки замовлення, що були проведені у 2008 році.

- Використовується таблиця [dbo].[ORDERS]

- Залучіть функцію Year

- Результативний набір даних містить: Ідентифікатор виробника

- Відсортуйте результативний набір даних за Ідентифікатор виробника (за зростанням)

***Задача 2***

Напишіть запит, який у розрізі кількості відпрацьованих років (математична різниця в роках) поверне кількість працівників.

- Використовується таблиця [dbo].[SALESREPS]

- Результативний набір даних містить: Кількість років, кількість працівників

- Відсортуйте результат за кількістю працівників за зменшенням

***Задача 3***

Напишіть запит, який поверне період (рік, місяць) найму з найбільшою кількістю працівників.

Враховуйте ймовірність того, що відразу кілька періодів можуть мати однакову кількість працівників.

- Використовується таблиця [dbo].[SALESREPS]

- Залучіть функції: Year, Month

- Результативний набір даних містить: Рік найму, місяць найму, кількість найнятих працівників

***Задача 4***

Напишіть запит, який у розрізі дня тижня (у строковому представленні) поверне кількість унікальних замовлень, загальну суму продаж, загальну кількість проданих од. товару (за весь час).

Враховуйте тільки замовлення, що були проведені в зимові місяці.

- Використовується таблиця [dbo].[ORDERS]

- Результативний набір даних містить: Номер дня тижня, назву дня тижня,

загальну кількість проведених замовлень, загальну суму продаж, загальну кількість проданих од. товару.

- Відсортуйте результат за днем тижня (за зростанням).

(Культура нумерації днів тижня значення не має)

**ДОДАТКОВІ МАТЕРІАЛИ:**

1. <https://learn.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-ver16>

**КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Яка основна відмінність між функціями `GETDATE()` та `SYSDATETIME()` та як це може вплинути на вибір функції для конкретної задачі? Оцініть різницю в точності цих функцій та їх застосування в залежності від потреб аналізу даних.

2. В яких ситуаціях важливо використовувати функцію `SYSDATETIMEOFFSET()` замість `SYSDATETIME()`? Обговоріть значення врахування часового пояса при роботі з датами та часом у міжнародних додатках.

3. Як можна використати функцію `DATEADD` для обчислення дати наступного дня після заданої дати? Надайте приклад запиту, що демонструє це використання.

4. Опишіть, як за допомогою функції `DATEDIFF` можна обчислити вік людини в роках. Поясніть кроки та надайте приклад SQL-запиту для цього розрахунку.

5. Як функція `DATENAME` може бути застосована для аналізу сезонності продажів? Продемонструйте, як отримати назву місяця з дати і як це може допомогти у групуванні продажів за місяцями.

6. Чому важливо використовувати функцію `ISDATE` перед конвертацією строкових значень у тип даних дата? Розгляньте сценарій, де необхідно забезпечити надійність даних при їх конвертації.

7. На які аспекти слід звернути увагу при використанні функції `MONTH` для аналізу розподілу подій протягом року? Вкажіть, як можна використати цю функцію для визначення місяця з найбільшою кількістю подій.

8. Як функція `YEAR` може допомогти у відстеженні динаміки змін за роками? Поясніть методику аналізу змін у показниках (наприклад, продажах) від року до року з використанням цієї функції.