**ЛАБОРАТОРНА РОБОТА №3**

СТВОРЕННЯ ТА МОДИФІКАЦІЯ ТАБЛИЦЬ В T-SQL. ВИБІР ТИПІВ ДАНИХ

Мета: мета лабораторної роботи полягає в ознайомленні студентів з практичними аспектами створення, модифікації та управління структурами таблиць в базі даних за допомогою мови T-SQL, яка є ключовою складовою Microsoft SQL Server. Лабораторна робота спрямована на набуття студентами необхідних навичок для ефективної роботи з таблицями — від проектування до оптимізації та адаптації під конкретні потреби використання. Мета даної лабораторної роботи полягає у наданні студентам знань та практичних навичок в області вибору оптимальних типів даних при проектуванні таблиць у системі управління базами даних SQL Server, використовуючи мову T-SQL. Завдання лабораторної роботи спрямоване на глибоке розуміння різноманітності типів даних та їх застосування в реальних сценаріях баз даних.

Детальні цілі лабораторної роботи включають:

1. Освоєння команд DDL для створення таблиць: студенти навчаться використовувати команду `CREATE TABLE` для створення таблиць з різними типами даних та визначенням обмежень (constraints), таких як первинні ключі, унікальність даних та зовнішні ключі.

2. Застосування команди ALTER TABLE: розуміння та практичне застосування команди `ALTER TABLE` для зміни структури існуючих таблиць, що включає додавання нових стовпців, модифікацію існуючих стовпців та видалення стовпців, а також зміну обмежень.

3. Проектування ефективних таблиць: вивчення кращих практик проектування таблиць для оптимізації зберігання та доступу до даних. Особлива увага приділяється вибору правильних типів даних, встановленню індексів та розумінню впливу цих рішень на продуктивність бази даних.

4. Модифікація таблиць з врахуванням цілісності даних: обговорення та практичні завдання на тему управління цілісністю даних через використання обмежень, включаючи встановлення NOT NULL, DEFAULT значень та використання зовнішніх ключів для забезпечення відносин між таблицями.

1. Ознайомлення з типами даних у T-SQL: студенти дізнаються про основні категорії типів даних, доступних у T-SQL, включаючи цілі числа, дробові числа, дати та час, рядки та спеціалізовані типи, такі як XML та JSON. Лабораторна робота допоможе оцінити особливості кожного типу і визначити критерії їх вибору.

2. Практичне застосування типів даних: виконання завдань на створення таблиць з різними типами даних, де студенти можуть практично оцінити вплив вибору типу даних на ефективність зберігання і обробки інформації.

3. Вибір типів даних залежно від контексту застосування: аналіз типів даних з точки зору специфіки застосування, обсягу зберігаючихся даних, швидкості обробки запитів та оптимізації ресурсів. Студенти навчаться робити обґрунтований вибір, базуючись на технічних вимогах і бізнес-логіці проектів.

4. Оцінка впливу типів даних на продуктивність: дослідження, як неправильний вибір типів даних може вплинути на продуктивність системи. Лабораторна робота включає аналіз випадків, коли великий об'єм даних або неефективні запити можуть викликати зниження продуктивності.

5. Навчання принципам оптимізації баз даних: в рамках лабораторної роботи студенти розглянуть рекомендації щодо оптимального використання типів даних для забезпечення масштабованості та високої доступності даних у комерційних базах даних.

Ця лабораторна робота надає студентам можливість практично застосувати теоретичні знання в реальних ситуаціях, вчить адаптувати структуру бази даних до вимог додатків та забезпечувати високу продуктивність обробки запитів. Результатом виконання лабораторних робіт має стати здатність студентів самостійно проектувати, створювати та модифікувати таблиці в системах управління базами даних.

**ТЕОРЕТИЧНІ ОСНОВИ**

Вибір типів даних є ключовим аспектом при проектуванні схеми реляційних баз даних, особливо в T-SQL, який є розширенням SQL, використовуваним у Microsoft SQL Server. Оптимальний вибір типів даних дозволяє підвищити продуктивність, ефективність використання дискового простору та точність даних. Ось деякі теоретичні основи для вибору типів даних:

*1. Фізичне зберігання даних:*

- Кожен тип даних має фіксований або змінний розмір зберігання. Наприклад, тип `INT` займає 4 байти, а `VARCHAR(n)` може змінюватися і займати до `n` символів плюс два байти для зберігання довжини.

- Вибір меншого типу даних за розміром може зекономити місце і збільшити продуктивність за рахунок меншого використання дискового простору і пам'яті.

*2. Логічне представлення даних:*

- Типи даних визначають, які операції можна виконувати з даними. Наприклад, арифметичні операції можуть бути виконані над числовими типами, але не над текстовими.

- Вони також визначають правила перетворення і порівняння. Наприклад, порівняння `DATE` і `DATETIME` враховує час у другому типі.

*3. Масштабування і точність:*

- Для числових типів даних, таких як `DECIMAL` і `NUMERIC`, важливо враховувати масштабування і точність, оскільки вони впливають на розмір зберігання і точність обчислень.

*4. Продуктивність:*

- Вибір типу даних впливає на продуктивність. Наприклад, оператори з більш малими числовими типами виконуються швидше, ніж з більшими.

- Індексація також залежить від типу даних; менші типи зазвичай означають швидшу індексацію.

*5. Нормалізація даних:*

- Правильний вибір типів даних сприяє нормалізації бази даних, зменшуючи дублювання і спрощуючи цілісність даних.

*7. Консистентність даних:*

- Вибір відповідного типу даних допомагає у забезпеченні консистентності даних.

*8. Основні принципи вибору типів даних:*

- *Мінімізація простору*: Вибирайте тип даних, який використовує найменше місця на диску, але при цьому здатен зберігати всі потрібні значення.

- *Точність*: Вибирайте тип даних, який найточніше представляє дані, особливо коли мова йде про фінансові або інші точні розрахунки.

- *Продуктивність*: Вибирайте тип даних, який оптимізує швидкість запитів, особливо при індексації та пошуку.

**Типи даних**

1. Скалярні типи даних: Це основні типи даних, які представляють одне значення з певним діапазоном і форматом. Вони включають числові, символьні, дати/часу та інші спеціалізовані типи.

2. Числові типи даних: Вибір числових типів даних (цілі, дробові, числа з плаваючою комою, тощо) базується на діапазоні даних та точності, які необхідні для конкретного поля. Для T-SQL числові типи даних включають:

- `INT`: Для цілих чисел.

- `BIGINT`: Для дуже великих цілих чисел.

- `SMALLINT`: Для менших цілих чисел.

- `TINYINT`: Для дуже малих цілих чисел.

- `DECIMAL` або `NUMERIC`: Для точних дробових чисел з фіксованою точністю та масштабом.

- `FLOAT` і `REAL`: Для чисел з плаваючою комою з великою точністю.

3. Символьні типи даних: Для текстових даних використовуються символьні типи:

- `CHAR`: Фіксована довжина.

- `VARCHAR`: Змінна довжина.

- `TEXT`: Для великих текстових даних, хоча у новіших версіях SQL Server рекомендується використовувати `VARCHAR(MAX)`.

4. Дата і час: Для зберігання значень дати і часу використовуються спеціалізовані типи даних:

- `DATE`: Тільки дата.

- `TIME`: Тільки час.

- `DATETIME` і `SMALLDATETIME`: Дата і час разом.

5. Бінарні типи даних: Для зберігання бінарних даних (наприклад, файлів або зображень) використовуються типи `BINARY` та `VARBINARY`.

6. Спеціальні типи даних: Існують також спеціальні типи даних, такі як `UNIQUEIDENTIFIER` для зберігання GUID значень, або `XML` для зберігання XML документів.

Важливо розуміти, що неправильний вибір типу даних може призвести до втрати точності, неефективного використання простору та зниження продуктивності системи. Крім того, це може вплинути на інтеграцію даних, міграцію, розробку та підтримку додатків.

**ЗАВДАННЯ 1**

1. Створити нову базу даних University

2. Створення таблиць:

- Таблиця Students: створіть таблицю для зберігання інформації про студентів. Таблиця повинна містити наступні стовпці:

- StudentID (ідентифікатор студента, ціле число)

- FirstName (ім'я студента, рядок)

- LastName (прізвище студента, рядок)

- EnrollmentDate (дата запису, дата)

- Таблиця Courses: створіть таблицю для зберігання інформації про курси. Таблиця повинна містити наступні стовпці:

- CourseID (ідентифікатор курсу, ціле число)

- CourseName (назва курсу, рядок)

- CourseDescription (опис курсу, текст)

- Credits (кредити, ціле число)

3. Заповнення таблиць даними:

- Вставте мінімум по 3 записи в кожну таблицю з довільними даними, що відображають студентів та курси.

4. Вибірка даних:

- Виконайте запити `SELECT \* FROM Students;` та `SELECT \* FROM Courses;` для перегляду даних, які ви вставили.

5. Модифікація таблиць:

- Додайте за допомогою команди ALTER по одному новому стовпцю до кожної таблиці:

- Для Students додайте стовпець Email (електронна адреса, рядок).

- Для Courses додайте стовпець Department (кафедра, рядок).

6. Вибірка даних:

- Повторно виконайте запити `SELECT \* FROM Students;` та `SELECT \* FROM Courses;` для перегляду оновлених таблиць.

7. Видалити базу даних University

**ЗАВДАННЯ 2**

1. Створити нову базу даних SHOP

2. Створити таблицю в БД SHOP для зберігання даних замовлень. Визначте найбільш оптимальні типи даних.

Найменування таблиці SALES.

Таблиця містить наступні поля:

SALES\_ORDER\_ID -- Унікальний ідентифікатор замовлення - число без дробової частини - може бути більше ніж 2147483647

REVISION\_NUMBER -- Інкремент для відстеження зміни за замовленням - число без дробової частини - не більше 255

ORDER\_DATE -- Дата і час створення замовлення - дата і час

DUE\_DATE -- Дата доставки замовлення - дата

SHIP\_DATE -- Фактична дата доставки замовлення - дата

ONLINE\_ORDER\_FLAG -- 0 = Замовлення було створено менеджером. 1 = Замовлення було створено онлайн - число без дробової частини - не більше 1

CUSTOMER\_ID -- Унікальний ідентифікатор клієнта, який здійснив замовлення - число без дробової частини - не більше 2147483647

SALES\_PERSON\_ID -- Унікальний ідентифікатор співробітника, який провів замовлення - число без дробової частини - не більше 2147483647

TERRITORY\_ID -- Унікальний ідентифікатор території - число без дробової частини - не більше 2147483647

SUB\_TOTAL -- Загальна сума замовлення - грошова величина - може бути більше ніж 214748.3647

TAX\_AMT -- Загальна сума податків - грошова величина - не більше 214748.3647

FREIGHT -- Сума доставки - грошова величина - не більше 214748.3647

MODIFIED\_DATE -- Дата і час останньої зміни рядка

3. Виконайте запит для внесення даних до створеної таблиці.

INSERT INTO SALES (SALES\_ORDER\_ID, REVISION\_NUMBER, ORDER\_DATE, DUE\_DATE, SHIP\_DATE, ONLINE\_ORDER\_FLAG,

CUSTOMER\_ID, SALES\_PERSON\_ID, TERRITORY\_ID, SUB\_TOTAL, TAX\_AMT, FREIGHT, MODIFIED\_DATE)

VALUES (43659, 11, '1753-01-01 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29825, 279, 5, 184590.1346, 1971.5149 ,616.0984 ,'2011-06-07 00:00:00.000'),

(43660, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29672, 279, 5, 1294.2529 , 124.2483 ,38.8276 ,'2011-06-07 00:00:00.000'),

(43661, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29734, 282, 6, 32726.4786 , 3153.7696 ,985.553 ,'2011-06-07 00:00:00.000'),

(43662, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29994, 282, 6, 28832.5289 , 2775.1646 ,867.2389 ,'2011-06-07 00:00:00.000'),

(43663, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29565, 276, 4, 419.4589 , 40.2681 ,12.5838 ,'2011-06-07 00:00:00.000'),

(43664, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29898, 280, 1, 24432.6088 , 2344.9921 ,732.81 ,'2011-06-07 00:00:00.000'),

(43665, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29580, 283, 1, 14352.7713 , 1375.9427 ,429.9821 ,'2011-06-07 00:00:00.000'),

(43666, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 30052, 276, 4, 5056.4896 , 486.3747 ,151.9921 ,'2011-06-07 00:00:00.000'),

(43667, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29974, 277, 3, 6107.082 , 586.1203 ,183.1626 ,'2011-06-07 00:00:00.000'),

(43668, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29614, 282, 6, 35944.1562 , 3461.7654 ,1081.8017 ,'2011-06-07 00:00:00.000'),

(43669, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29747, 283, 1, 714.7043 , 70.5175 ,22.0367 ,'2011-06-07 00:00:00.000'),

(43670, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29566, 275, 3, 6122.082 , 587.5603 ,183.6126 ,'2011-06-07 00:00:00.000'),

(43671, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29890, 283, 1, 8128.7876 , 780.8136 ,244.0042 ,'2011-06-07 00:00:00.000'),

(43672, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 30067, 282, 6, 6124.182 , 587.6023 ,183.6257 ,'2011-06-07 00:00:00.000'),

(43673, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29844, 275, 2, 3746.2015 , 357.9614 ,111.8629 ,'2011-06-07 00:00:00.000'),

(43674, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29596, 282, 6, 2624.382 , 251.9407 ,78.7315 ,'2011-06-07 00:00:00.000'),

(43675, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29827, 277, 3, 5716.3102 , 546.8759 ,170.8987 ,'2011-06-07 00:00:00.000'),

(43676, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29811, 275, 5, 14203.458 , 1363.266 ,426.0206 ,'2011-06-07 00:00:00.000'),

(43677, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29824, 278, 6, 7793.1108 , 747.1011 ,233.4691 ,'2011-06-07 00:00:00.000'),

(43678, 8, '2011-05-31 00:00:00.000', '2011-06-12', '2011-06-07', 0, 29889, 281, 4, 9799.9243 , 942.074 ,294.3981 ,'2011-06-07 00:00:00.000');

4. Відібрати з таблиці (отримати) всі дані

5. Видалити таблицю SALES

6. Видалити БД SHOP

**ДОВІДКОВІ МАТЕРІАЛИ:**

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

**КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Які основні етапи створення таблиці в T-SQL? Опишіть послідовність кроків, які необхідно виконати для створення таблиці в SQL Server, звертаючи увагу на важливість вибору правильних типів даних та визначення первинного ключа.

2. Що таке DDL і які команди входять до цієї категорії? Наведіть приклади команд Data Definition Language, які використовуються для створення, модифікації та видалення об'єктів бази даних, і поясніть, як кожна з них застосовується при роботі з таблицями.

3. Як змінити структуру існуючої таблиці за допомогою команди ALTER TABLE? Дайте детальний опис того, як використовувати `ALTER TABLE` для додавання, видалення або зміни стовпців у таблиці. Включіть приклади команд, які демонструють кожну з цих операцій.

4. Які важливі обмеження (constraints) можуть бути застосовані під час створення таблиць? Обговоріть різні типи обмежень, такі як NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, і їхнє значення для забезпечення цілісності даних у базі.

5. Які наслідки можуть бути за неправильний вибір типу даних для стовпця таблиці? Проаналізуйте, як невідповідність типу даних вимогам додатка може вплинути на продуктивність, зберігання та обробку даних.

6. Які кращі практики слід дотримувати при проектуванні таблиць для бази даних? Запропонуйте рекомендації щодо оптимального визначення структури таблиць, включаючи кількість і типи стовпців, використання індексів та забезпечення швидкого доступу до даних.

7. Що таке типи даних у T-SQL та яке їхнє призначення у проєктуванні баз даних? Опишіть концепцію типів даних у контексті T-SQL і поясніть, чому правильний вибір типів даних є критичним для ефективності та надійності бази даних.

8. Які основні категорії типів даних у T-SQL і чим вони відрізняються? Надайте класифікацію типів даних, зазначте їх характеристики та основні випадки використання для числових, рядкових, дати і часу, а також спеціалізованих типів даних.

9. Які критерії слід враховувати при виборі типу даних для нового стовпця в таблиці? Обговоріть фактори, такі як обсяг зберігання, продуктивність, масштабованість та точність, які потрібно враховувати при виборі типів даних.

10. Чому важливо розуміти різницю між типами даних INT і BIGINT, VARCHAR і NVARCHAR? Проаналізуйте вплив вибору цих типів на розмір зберігання і швидкість обробки в базі даних.

11. Як можна визначити, що вибраний тип даних не є оптимальним для конкретної задачі? Розгляньте ознаки і наслідки неправильного вибору типів даних, такі як збільшення часу завантаження даних, проблеми з виконанням запитів, чи неправильне відображення даних.

12. Які типи даних у T-SQL можна використовувати для зберігання текстової інформації? Порівняйте і протиставте типи даних, такі як CHAR, VARCHAR, TEXT, і NVARCHAR, вказуючи їх плюси та мінуси та сценарії використання.

13. Які спеціалізовані типи даних пропонує T-SQL і в яких випадках їх застосування є доцільним? Детально описати типи даних, такі як XML та JSON, і обговорити, як ці типи можуть бути використані для роботи зі структурованими та напівструктурованими даними.

14. Які вправи та експерименти можна провести для підтвердження правильного вибору типів даних? Запропонуйте методи тестування та валідації вибору типів даних через практичні завдання та експерименти, які допоможуть студентам перевірити і зрозуміти наслідки своїх рішень.