

Тема лекції 7:

Віртуальні таблиці (view – представлення)

- ☐ Базові та віртуальні таблиці
 - ☐ Використання віртуальних таблиць
 - ☐ Створення віртуальних таблиць
 - ☐ Типи віртуальних таблиць
 - ☐ Модифікація даних у віртуальних таблицях
 - ☐ Зміна схеми бази даних і віртуальні таблиці
 - ☐ Видалення віртуальної таблиці
-

Базові та віртуальні таблиці (TABLE and VIEW)

□ Базові таблиці (TABLE):

- фізичні об'єкти БД, які містять дані і зберігаються в пам'яті комп'ютера на жорсткому диску
 - запит на вибірку даних до базових таблиць є тимчасова результатна таблиця, доступна лише тому, хто виконав запит
-

Базові та віртуальні таблиці (TABLE and VIEW)

□ Віртуальні таблиці (VIEW):

- не є фізичними об'єктами зберігання даних
 - дозволяють повертати певні поля таблиць у вигляді зв'язаного набору даних, які можуть бути доступні багатьом користувачам і існують в базі даних до тих пір, поки не будуть спеціально видалені
-

Базові та віртуальні таблиці (TABLE and VIEW)

- ❑ Дані у віртуальних таблицях, подібно як і в результатних таблицях запиту, вибираються з базових таблиць, тобто представляються в тому чи іншому вигляді.
 - ❑ Параметри представлення даних у віртуальних таблицях зберігаються у розділі метаданих бази. Наприклад, в SQL Server вони зберігаються в системних представленні каталогу **sys.***.
 - ❑ Працювати з віртуальними таблицями можна як зі звичайними базовими таблицями.
 - ❑ Будь-який новий запит до віртуальної таблиці ініціює прихований запит до базових таблиць, який комбінується з цим новим запитом.
-

Використання віртуальних таблиць (представлень)

- надбудова для адаптації бази даних до різних категорій користувачів
 - потреба надати користувачу дані не в тому форматі, в якому вони зберігаються у базі даних:
 - представлення забезпечують «персоналізацію» даних, перетворюючи набір нормалізованих таблиць в одну або декілька віртуальних таблиць, зрозумілих користувачу
 - у представленнях можна переіменовувати поля таким чином, щоб користувачі з різним рівнем підготовки отримували дані у зрозумілих термінах
-

Використання віртуальних таблиць (представлень)

- вирішення проблеми захисту даних:
 - адміністратор бази даних надає користувачам доступ до певної інформації лише через віртуальні таблиці
 - адміністратор бази даних визначає для користувачів такі набори результатів, які не дозволять їм бачити дані, призначені для інших користувачів.
-

Використання віртуальних таблиць (представлень)

- основа для інших представлень або запитів:
 - представлення можуть вкладатись одне в інше, і при цьому кожен рівень буде виконувати свою конкретну функцію з перетворення базового набору даних
 - складний запит, що має ієрархічну структуру, легше сформулювати, використовуючи віртуальні таблиці як допоміжні
-

Створення віртуальних таблиць

- ❑ Інструкція створення віртуальної таблиці :
CREATE VIEW <ім'я віртТабл> AS <запитSELECT>;
 - ❑ Віртуальній таблиці присвоюють ім'я, яке не повинно співпадати ні з одним іменем базової таблиці.
 - ❑ За ключовим словом AS записується SQL-запит на вибірку даних.
 - ❑ Інструкція CREATE VIEW не вибирає дані з таблиць і не відображає їх, а лише дає вказівку СУБД запам'ятати команду SELECT як представлення з іменем <ім'я віртТабл>.
 - ❑ <ім'я віртТабл> використовується потім для формування запитів до цієї віртуальної таблиці.
-

Представлення (VIEW) в SQL Server

- ❑ Представлення може бути створене лише в поточній базі даних
 - ❑ Представлення може включати не більше 1024 стовпців
 - ❑ Відомості про представлення зберігаються в таких віртуальних таблицях каталогу:
 - ***sys.views***
 - ***sys.columns***
 - ***sys.sql_dependencies***
 - ❑ Текст інструкції CREATE VIEW зберігається в представленні каталогу ***sys.sql_modules***.
-

Створення представлення (VIEW)

- Приклад 1 (віртуальна таблиця, побудована на декількох базових): створити віртуальну таблицю, в якій відображено імена продавців, які обслуговують кожного замовника:

```
CREATE VIEW Service AS
SELECT Customers.cname AS custname,
       Salers.sname AS salename
FROM Customers JOIN Salers
ON Salers.snum=Customers.snum;
```

- В результаті буде створена віртуальна таблиця Service до якої можна звертатись з запитами
-

Створення представлення (VIEW)

- Приклад 2 (віртуальна таблиця, побудована на декількох базових): створити віртуальну таблицю, в якій відображено операцію-купівлі-продажу, ім'я продавця, що обслуговує дану операцію, та суму комісійних продавця, яку він отримає за проведення операції:

```
CREATE VIEW Sale_Orders AS
SELECT Orders.onum AS onum,
       Salers.sname AS salename,
       Orders.amt* Salers.comm AS paym
FROM Orders JOIN Salers
ON Salers.snum=Orders.snum;
```

- В результаті буде створено віртуальну таблицю Sale_Orders, до якої можна звертатись з запитами.
-

Створення представлення (VIEW)

- Приклад 3 (віртуальна таблиця, побудована на одній базовій): створити віртуальну таблицю списку продавців:

```
CREATE VIEW Salerslist AS  
  SELECT sname,city,comm  
  FROM Salers;
```

Типи віртуальних таблиць

- ☐ Базові представлення
 - ☐ Представлення з'єднання таблиць
 - ☐ Представлення окремих записів
 - ☐ Представлення окремих полів
 - ☐ Підсумкові представлення
-

Базові представлення

- ❑ будуються через вибірку даних з базових таблиць (див. приклад 3)
 - ❑ користувачу не потрібні первинні і зовнішні ключі таблиць, тому у віртуальні таблиці вибираються лише поля з даними
 - ❑ у таких представленнях немає необхідності виконувати операції фільтрування або впорядкування базових даних
 - ❑ такий набір результатів є більш змістовним з точки зору кінцевих даних.
-

Базове представлення з'єднання таблиць

- ❑ Використовується для зв'язування усіх таблиць бази даних
 - ❑ Це базове представлення, в якому присутні усі дані бази без зовнішніх ключів.
 - ❑ Для формування таких базових представлень в SQL Server необхідно вказувати у фразі SELECT класифікатори даних, особливо, коли вибираються поля з однаковими іменами.
 - ❑ Базові представлення, побудовані як з окремих базових таблиць, так і з'єднаних базових таблиць формують основу для інших запитів, які використовують усі дані.
 - ❑ Базові представлення відображають базові запити до усіх даних.
-
- ❑ **Представлення з'єднання таблиць** будуються через вибірку конкретних полів, в якій присутні дані з однієї або декілька зв'язаних таблиць (можуть бути і небазовими).
-

Представлення окремих записів

- ❑ Використовують у запиті, який їх будує, фразу фільтрування рядків WHERE.
 - ❑ У деяких СУБД можна використовувати і фразу ORDER BY для сортування рядків за значеннями окремих полів, однак у цьому випадку накладаються обмеження на модифікацію даних через представлення.
 - ❑ Такі представлення можна побудувати на основі базових представлень.
-

Опція для представлень окремих записів

- ❑ Опція CHECK для перевірки умови фільтрування при операціях модифікації даних.
 - ❑ Інструкція побудови такого представлення:
CREATE VIEW <ім'я віртТабл> AS
SELECT {<стовпець> AS <псевдонім>, ...}
FROM <базова_таблиця>
WHERE <умова пошуку>
WITH CHECK OPTION;
 - ❑ СУБД зберігає <умову пошуку> разом з представленням.
 - ❑ Кожен раз, коли користувач виконує через цю віртуальну таблицю інструкції DML, СУБД перевіряє кожну дію на відповідність критеріям у фразі WHERE.
 - ❑ Будь-які дії, що не відповідають цим критеріям, виконуватись не будуть.
 - ❑ Операції модифікації для такого типу віртуальних таблиць повинні стосуватись лише тих рядків, які задовольняють умову фільтрування.
-

Представлення окремих полів

- Використовують у запиті, який їх будує, фразу SELECT з переліком конкретних стовпців, які необхідно вибрати.
 - Вибираються усі записи (без фрази WHERE).
 - За потребою можна будувати комбіновані **представлення окремих полів з фільтрованими записами**. Такі представлення *рекомендується* будувати на основі базових представлень.
 - У деяких СУБД можна використовувати і фразу ORDER BY для сортування рядків за значеннями окремих полів, однак у цьому випадку накладаються обмеження на модифікацію даних через представлення.
 - Такі представлення можна побудувати на основі базових представлень.
-

Підсумкові представлення

- ❑ використовуються при аналізі даних
 - ❑ рекомендується будувати на основі базових представлень, щоб не турбуватись про операції з'єднання таблиць
 - ❑ при побудові використовуються обчислювані стовпці, агрегатні функції, групування та інші операції, необхідні для представлення підсумкових даних
 - ❑ не допускають модифікацію своїх даних
-

Обмеження модифікації даних у віртуальних таблицях

- ❑ будь-які зміни даних через інструкції UPDATE, INSERT та DELETE повинні стосуватись стовпців лише однієї базової таблиці
 - ❑ забороняється застосовувати інструкцію DELETE до віртуальних таблиць, визначених на декількох базових таблицях; в SQL Server це правило розповсюджується на інструкції UPDATE, INSERT та DELETE
 - ❑ забороняється модифікувати дані через віртуальну таблицю, визначену із використанням підзапитів
 - ❑ інструкція INSERT використовується лише в тому випадку, якщо віртуальна таблиця містить усі NOT NULL значення базової таблиці
-

Обмеження модифікації даних у віртуальних таблицях

Забороняється модифікувати дані через віртуальну таблицю, визначену:

- ❑ із застосуванням у фразі SELECT ключового слова DISTINCT
 - ❑ з використанням агрегатних функцій;
 - ❑ з використанням фраз GROUP BY та HAVING.
 - ❑ з використанням операцій над множинами
 - ❑ з використанням фрази ORDER BY;
-
- ❑ не дозволяється оновлювати обчислювані стовпці, тобто стовпці, значення яких є результатами обчислення виразів;
-

Зміна схеми бази даних і віртуальні таблиці

- ❑ Віртуальні таблиці повинні забезпечувати незалежність користувацьких програм від змін у логічній структурі БД.
 - ❑ В SQL Server якщо представлення створене без використання фрази `WITH SCHEMABINDING`, то при зміні об'єктів (базових або віртуальних таблиць), які впливають на визначення представлення, необхідно виконувати збережену процедуру ***sp_refreshview***. У протилежному випадку результат запиту, за яким будується представлення, може бути непередбаченим.
 - ❑ Якщо представлення залежить від видаленого об'єкта (базової або віртуальної таблиці), то воно не може використовуватись – його необхідно видалити.
 - ❑ Якщо замість видаленої базової таблиці створено нову базову таблицю, навіть з подібною структурою, то представлення і в цьому випадку необхідно видалити і побудувати заново.
-

Видалення віртуальної таблиці

- Інструкція:

`DROP VIEW <ім'я віртТабл>;`

- При видаленні віртуальної таблиці дані, які в ній відображались, залишаються без змін, оскільки віртуальна таблиця не є об'єктом зберігання даних
-

Видалення віртуальної таблиці, яка використовувалась для визначення інших віртуальних таблиць

- ❑ Використовуються ключові слова:
CASCADE і RESTRICT.
 - ❑ Якщо вказана інструкція :
 DROP VIEW <ім'я віртТабл> CASCADE;
то при видаленні вказаного представлення
видаляються усі залежні від нього представлення.
Таким чином видаляються цілі ланцюги
представлень.
 - ❑ Якщо вказана інструкція :
 DROP VIEW <ім'я віртТабл> RESTRICT;
то видалення вказаного представлення
відміняється, якщо існують залежні від нього
представлення.
-

Дякую за увагу
