

ОСОБЛИВОСТІ ВИКОРИСТАННЯ МОВИ МАНІПУЛЮВАННЯ ДАНИМИ SQL У СУЧАСНИХ СКБД

Розглянуті особливості синтаксису мови маніпулювання даними SQL у сучасних СКБД Access, Microsoft SQL Server, Oracle та DB2 з точки зору мобільності скрипта по здійсненню операцій пошуку та модифікації даних при переході на іншу платформу. Визначені основні моменти на які слід звернути увагу при здійсненні подібної практичної задачі.

Ключові слова: мова SQL, пошук даних, маніпулювання даними, NULL-значення, теоретико-множинні операції.

Вступ

Structured Query Language (SQL) – це досить потужна і одночасно не складна та непроцедурна мова для керування базами даних. Вона підтримується практично всіма сучасними системами керування базами даних (СКБД) і є універсальною комп'ютерною мовою для створення, модифікації та керування даними у реляційних базах даних. SQL використовує усі переваги реляційної моделі, зокрема її математичного апарату – реляційної алгебри та реляційного числення, використовуючи при цьому досить невелике число команд і відносно простий синтаксис [2]. Серед п'яти основних груп команд SQL [13] особливе місце займає Data Manipulation Language (DML) – мова маніпулювання даними) яка призначена для виконання основних операцій під час роботи з даними – пошуку та модифікації.

Незважаючи на те, що мова SQL стандартизована і, як декларується, більшість сучасних СКБД підтримують стандарт SQL92 [10], тим не менш практичне її використання показує, що досягти у повній мірі переносу запитів на SQL з однієї системи на іншу дуже складно, а іноді і неможливо [14,16].

Мета статті – не дати опис мови SQL, яких достатньо [1, 3, 6, 7, 9], а саме проаналізувати характерні особливості, на які слід звернути увагу, та відмінності у використанні мови маніпулювання даними SQL у середовищі Microsoft Access, Microsoft SQL Server, Oracle та DB2. Microsoft Access є найбільш простою системою, хоча й достатньо потужною. Вона часто використовується як у навчальних цілях, так і кінцевими користувачами, має багаті візуальні засоби проектування, потужні мовні засоби створення запитів до бази даних. Три інші системи є найбільш поширеними у світі і займають до 90% ринку комерційних СКБД [12]. При тестуванні використовувалися: Microsoft Access 2010; Microsoft SQL Server 2008 R2 (Express); Oracle XE 10g; DB2-Express C 9.7.

Основна частина

На відміну від мовних конструкцій визначення даних Data Definition Language (DDL), які зазвичай описують створення та модифікацію структури об'єктів у базі даних та виконуються досить рідко, а частіше за все – один раз, на початковому етапі формування схеми бази даних, запити на мові маніпулювання даними використовуються повсякденно при експлуатації бази даних і тому знання особливостей застосування таких мовних конструкцій дозволить з мінімальними витратами забезпечити мобільність програм, у яких використовується запити на мові SQL. Ілюстрація прикладів буде здійснюватися на основі трьох таблиць DEPT, EMP та SALGRADE тестової бази СКБД Oracle "SCOTT/TIGER", що супроводжує усі інсталяції цієї СКБД.

```
CREATE TABLE DEPT (
  DEPTNO NUMBER(2)
  CONSTRAINT PK_DEPT PRIMARY KEY,
  DNAME VARCHAR(14) ,
  LOC VARCHAR(13) ) ;
```

```
CREATE TABLE EMP (
  EMPNO NUMBER(4)
  CONSTRAINT PK_EMP PRIMARY KEY,
  ENAME VARCHAR(10),
  JOB VARCHAR(9),
  MGR NUMBER(4),
  HIREDATE DATE,
  SAL NUMBER(7,2),
  COMM NUMBER(7,2),
  DEPTNO NUMBER(2)
  CONSTRAINT FK_DEPTNO
  REFERENCES DEPT);
```

```
CREATE TABLE SALGRADE (
  GRADE NUMBER,
  LOSAL NUMBER,
  HISAL NUMBER ) ;
```

Початкове заповнення таблиць бази даних

При створенні таблиці у базі даних формується тільки її структура, але звісно ця таблиця є порожньою, бо ніяких даних до неї не додавалося. Для того щоб наповнити таблицю новими даними, необхідно виконати команду INSERT (вставити, додати), яка безпосередньо і виконує таку функцію.

Виконуючи команду INSERT, необхідно мати на увазі такі особливості [3]:

- Під час введення значень у стовпці нового запису необхідно слідкувати, щоб типи значень відповідали типам стовпців таблиці. У разі необхідності можна використати функцію перетворення типів CAST() MS SQL Server або її подібну у конкретній СКБД.

- Дати зазвичай вводяться як символічні значення, що розташовані в апострофах (для Access – у #). Але оскільки формати подання дати у різних країнах відрізняються, то необхідно або використовувати функції перетворення, або встановлювати для сеансу роботи такий режим, який би "інструктував" СКБД про формат значень, що мають тип "дата". Наприклад, при виконанні команд у середовищі MS SQL Server необхідно перед виконанням команди INSERT ввести команду SET LANGUAGE RUSSIAN

яка і налаштує необхідним чином систему а, наприклад, у СКБД Oracle використати стандартну функцію to_date з відповідним форматом перетворення, чи на початку роботи виконати команду:

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD/MM/YYYY';
```

Ще одна складність виникає при спробі додати декілька рядків у таблицю у одній команді INSERT, згідно стандарту SQL/92.

```
INSERT INTO SALGRADE VALUES  
(6,10000,19999),  
(7,20000,49999);
```

Така конструкція дозволена у DB2 та SQL Server і видає помилку для Access та Oracle.

Пошук інформації у БД

Загальновідомий факт – понад 80% займають пошукові операції до бази даних які здійснюються за допомогою команди SELECT.

Перше, на що слід звернути увагу, це використання стандартних функцій. Хоча стандарт і визначає їх перелік [11], тим не менш кожна СКБД має свої відмінності [7, 8, 15]. Наприклад, для визначення співробітників, що були прийняті на роботу у лютому для Access, SQL Server та DB2 спрацює такий запит:

```
SELECT EMPNO, ENAME, HIREDATE  
FROM EMP  
WHERE (Month(HIREDATE)=2);
```

А для Oracle – умову пошуку слід записати таким чином: WHERE (to_char(HIREDATE,'MM')=2)

Пошук з використанням шаблону

У деяких запитів, які важко реалізувати з використанням операцій порівняння, діапазону чи належності до множини часто використовують спеціальну конструкцію LIKE (подібний, схожий), яка дозволяє виконувати порівняння виразу із

заданим шаблоном, в якому допускається використання символів-замінників: % (процент) – замість цього символу може бути підставлена будь-яка кількість (і навіть нуль) довільних символів та _ (підкреслення) – замінює один символ рядка. Зазначимо, що у Access для цього використовуються символи *(зірочка) та ?(знак питання). Однак СКБД SQL Server та Access мають розширений синтаксис шаблону, який дозволяє у квадратних дужках задати перелік або діапазон символів, яким повинен відповідати (чи не відповідати) шаблон умови пошуку. Наприклад:

```
SELECT EMPNO, ENAME FROM EMP  
WHERE (EMPNO LIKE '%[1-3]%');
```

вибирає усіх співробітників, у службовому номеру яких є цифри від 1 до 3.

У Oracle та DB2 результат подібного запиту – порожня таблиця. У цьому разі треба застосувати запит з більш складною конструкцією умови пошуку.

Упорядкування результатів вибірки даних

Упорядковані результати отримують використовуючи спеціальне речення ORDER BY (впорядкувати), яке дозволяє визначити стовпці, за значенням яких буде виконуватися впорядкування (сортування) рядків, а також напрям – за зростанням чи спаданням. Ознаки ASC та DESC вказують напрямом. ASC – за зростанням (за замовчуванням), DESC – за спаданням.

Особливо слід звернути увагу на впорядкування рядків, що мають значення NULL. Результат, отриманий у Access та MS SQL Server, відрізняється від результату в СКБД Oracle та DB2. Для продуктів Microsoft вважається, що NULL – це найменше значення, а для Oracle та DB2 – найбільше.

Зовнішні з'єднання

Поряд з командами LEFT JOIN та RIGHT JOIN, які виконуються у всіх СКБД, Oracle дозволяє задати зовнішнє з'єднання у конструкції WHERE за допомогою оператора зовнішнього з'єднання. Цей оператор представляється символом плюс, який береться в круглій дужці (+). Він знаходиться на тій стороні сполучної умови, що відповідає таблиці з відсутніми даними.

Агрегатні функції

При використанні у запитах SQL агрегатних функцій слід особливу увагу звернути на використання ключового слова DISTINCT, як аргументу агрегатної функції – ця особливість не підтримується Access, та обробки NULL-значень.

У стандарті ANSI/ISO визначені такі точні правила обробки значень NULL в агрегатних функціях [1]:

- якщо які-небудь значення, що містяться в стовпці, дорівнюють NULL, то під час обчислення результату функції вони виключаються;

- якщо усі значення в стовпці дорівнюють NULL, то функції SUM(), AVG(), MIN() та MAX() повертають значення NULL; функція COUNT() повертає нуль;

- якщо в стовпці немає значень (тобто стовпець порожній), то функції SUM(), AVG(), MIN() і MAX()

повертають значення NULL; функція COUNT() повертає нуль;

- функція COUNT(*) підраховує кількість рядків і не залежить від наявності або відсутності в стовпці значень NULL; якщо рядків у таблиці немає, ця функція повертає нуль.

```
SELECT COUNT(*) AS CCOUNT,
COUNT(COMM) AS CCOMM,
COUNT(SAL) AS CSAL,
COUNT(DISTINCT SAL) AS DSAL
FROM EMP
```

CCOUNT	CCOMM	CSAL	DSAL
12	4	12	11

СКБД, що аналізуються, підтримують ці вимоги стандарту, але ти не менш, слід завжди приділяти особливу увагу випадкам, коли усі значення у стовпці дорівнюють NULL, або таблиця порожня.

Функції обробки NULL-значень

Найбільш непередбачувані результати можна отримати у разі, коли атрибути, які використовуються у обчисленнях, мають значення NULL. При обробці значень NULL часто використовують спеціально призначені для цього функції: ISNULL, NZ – в Access; ISNULL, COALESCE – у SQL Server; NVL – в Oracle; COALESCE, NVL – у DB2. Наприклад, якщо необхідно отримати відомості про річний дохід співробітників, який визначається їх окладом та премією (поля SAL та COMM таблиці EMP), то запит у Oracle повинен бути таким:

```
SELECT EMPNO,ENAME,
SAL*12+NVL(COMM,0) AS YEAR_SAL
FROM EMP
```

Якщо ж вираз для обчислення записати як SAL*12+COMM, то отримаємо відсутні значення річного доходу для співробітників, які не отримували премію.

Підзапити

Підзапит (вкладений запит) – потужний засіб мови SQL, який дозволяє будувати складні ієрархії запитів. Підзапит (subquery) використовують як додатковий метод маніпуляцій з декількома таблицями. Зазвичай він формується за допомогою команди SELECT, що вкладена:

- у речення WHERE, HAVING або SELECT іншого оператора SELECT;
- у оператор INSERT, UPDATE або DELETE;
- в інший підзапит.

При використанні підзапитів найбільш неочікуваними можуть бути результати, які викликані наявністю значень NULL. Наприклад, визначити співробітників, заробітна плата яких вища за заробітну плату будь-якого співробітника відділу 30.

```
SELECT EMPNO, ENAME, DEPTNO, SAL
FROM EMP
WHERE (SAL > ALL ( SELECT SAL
FROM EMP WHERE DEPTNO=30))
```

EMPNO	ENAME	DEPTNO	SAL
7566	JONES	20	2975.00

7839	KING	10	5000.00
7902	FORD	20	3000.00

Визначити співробітників, премія яких вища за премію будь-якого співробітника відділу 30.

```
SELECT EMPNO, ENAME, DEPTNO, SAL
FROM EMP
WHERE (COMM > ALL (SELECT COMM
FROM EMP WHERE DEPTNO=30))
```

Аналогічний за записом запит однак видає пушту таблицю.

Причина криється у тому, що вкладений підзапит містить у якості результату рядки зі значенням NULL. Тому зовнішній запит не спрацьовує так, як очікувалось, бо для значення NULL не можна визначити, більше воно чи менше деякого фактичного значення. Для отримання вірної відповіді у вкладеному запиті слід додати одну додаткову умову, що виключає з переліку цін невизначене значення NULL, а саме:

```
... AND COMM IS NOT NULL
```

Порівняння за декількома значеннями для вкладеного запиту

Дуже часто виникає потреба у порівнянні значень декількох атрибутів зовнішнього запиту з відповідними атрибутами вкладеного. Наприклад, отримати відомості про співробітників, що отримують максимальну заробітну платню у кожному відділі.

```
SELECT EMPNO,ENAME, DEPTNO,SAL
FROM EMP
WHERE (DEPTNO,SAL) IN (
SELECT DEPTNO, MAX(SAL)
FROM EMP
GROUP BY DEPTNO)
ORDER BY 3
```

EMPNO	ENAME	DEPTNO	SAL
7839	KING	10	5000
7902	FORD	20	3000
7698	BLAKE	30	2850

На превеликий жаль, слід зазначити, що наведений приклад спрацьовує лише у СКБД Oracle та DB2. У інших системах з'являється повідомлення про помилку у тексті запиту.

Вкладені підзапити на створення нової таблиці у базі даних.

Досить цікаве використання підзапитів у команді CREATE TABLE. Фактично такий підзапит формує нову таблицю у базі даних, яка за своєю структурою, тобто переліком стовпців та їхніми типами даних, відповідає результату вкладеного запиту. Наприклад, створити нову таблицю у базі даних на ім'я "MinSalDept", яка повинна містити відомості про мінімальну заробітну платню у кожному відділі, можна таким чином.

```
CREATE TABLE MinSalDept AS
SELECT DEPTNO,MIN(SAL) AS MIN_SAL
FROM EMP
GROUP BY DEPTNO
```

Наведене рішення виконується тільки в СКБД Oracle. Аналогічна команда для Access та MS SQL Server записується так:

```
SELECT DEPTNO,MIN(SAL) AS MIN_SAL
INTO MinSalDept
FROM EMP
GROUP BY DEPTNO
```

тобто замість CREATE TABLE у запиті SELECT у додатковому реченні INTO вказується ім'я новостворюваної таблиці.

Реалізація теоретико-множинних операцій мовою SQL

Як відомо, Кодд у своїх роботах сформулював сім основних та одну додаткову операцію реляційної алгебри [2; 5]. До основних відносяться операції: об'єднання, перетину, віднімання (різниці), декартового добутку, вибірки, проекції та з'єднання. Додаткова операція – ділення. Результат кожної з цих операцій можна отримати використовуючи синтаксис мови SQL.

Операція об'єднання. У мові SQL є стандартна команда UNION. Вона об'єднує результати декількох команд SELECT, автоматично видаляючи рядки, що дублюються.

Операція вибірки формується за допомогою команди SELECT, яка містить речення WHERE. У ньому вказують умову пошуку (відбору) рядків результуючої таблиці, для яких значення стовпців дають результат "істина" (горизонтальний зріз).

Операція проекції за своєю суттю реалізується також командою SELECT, яка відбирає декілька стовпців із таблиці (вертикальний зріз).

Реалізація декартового добутку мовою SQL досить проста. Для її виконання у реченні FROM команди SELECT необхідно тільки вказати перелік таблиць через кому. СКБД MS SQL Server та DB2 мають додаткову мовну конструкцію для позначення декартового добутку – CROSS JOIN.

Віднімання (або різниця) двох відношень R1 та R2 – це нове відношення, що містить усі кортежі, які належать першому відношенню R1 і не належать другому R2. Наприклад, для визначення того, працівники яких посад є у 20 відділі, та відсутні у 30 відділі, треба виконати запит:

```
SELECT JOB FROM EMP WHERE DEPTNO=20
MINUS
SELECT JOB FROM EMP WHERE DEPTNO=30
```

Операція MINUS правильно виконується у СКБД Oracle та DB2, і дає помилку (тобто не реалізована) у Access та MS SQL Server. Замість використання ключового слова MINUS у MS SQL Server, а також DB2 підтримується інше – EXCEPT (за винятком). Воно виконує аналогічну операцію.

Операція перетину реалізується за допомогою ключового слова INTERSECT.

```
SELECT JOB FROM EMP WHERE DEPTNO=20
INTERSECT
SELECT JOB FROM EMP WHERE DEPTNO=30
```

Access також не підтримує безпосередньо операцію перетину.

Операція ділення є додатковою операцією реляційної алгебри, і як окрема операція не визначена у мові SQL, але вона може бути реалізована через інші операції [4].

Відбір ранжируваних значень

Дуже часто виникають завдання відбору перших рядків з результатів запиту, що впорядковані за тією чи іншою ознакою. Наприклад, отримати інформацію про трьох співробітників, що отримують найменшу заробітну платню. Реалізація подібних запитів відрізняється у всіх СКБД, що аналізувалися.

Для SQL Server запит виглядає таким чином:

```
SELECT TOP(3) EMPNO, ENAME, SAL
FROM EMP ORDER BY SAL
```

EMPNO	ENAME	SAL
7369	SMITH	800.00
7900	JAMES	950.00
7521	WARD	1250.00

Для Access

```
SELECT TOP 3 EMPNO, ENAME, SAL
FROM EMP ORDER BY SAL
```

Для DB2:

```
SELECT EMPNO, ENAME, SAL
FROM EMP ORDER BY SAL
LIMIT 0,3
```

Найбільш складна реалізація стосується СКБД Oracle, оскільки там не визначений предикат відбору перших n рядків за довільною умовою:

```
SELECT * FROM (
SELECT T.*, ROWNUM RNUM
FROM (SELECT ROWNUM, EMPNO,
ENAME, SAL FROM EMP ORDER BY SAL) T )
WHERE RNUM<=3
```

СКБД SQL Server та Access дозволяють поряд із ключовим словом TOP записати PERCENT, що дозволяє відбирати певну кількість рядків з таблиці, що складають конкретний відсоток, від усіх вибраних рядків.

Крім того SQL Server має додаткову фразу WITH TIES, яка вказує, що у результат виконання запиту може включати будь-які додаткові рядки, які відповідають значенням в стовпці ORDER BY або стовпцях в останньому поверненому рядку [17].

Великі можливості по обробці ранжируваних даних відкривають аналітичні функції SQL, однак аналіз і дослідження даного питання виходить далеко за рамки цієї статті.

Висновки

Аналіз застосування мови маніпулювання даними SQL для здійснення основних операцій у базах даних показав, що стандартизація SQL в реальних умовах застосування комерційних СКБД на превеликий жаль є не близькою перспективою. Домогтися мобільності скриптів по маніпулюванню даними дуже складно, а інколи й неможливо в повній мірі, враховуючи різноманітність іменування функцій, засоби обробки дат, використання шаблонів, специфіку обробки NULL-значень тощо. Тим не менше, якщо використовувати базові

элементы в языковых конструкциях SELECT, INSERT, UPDATE та DELETE можна в значній мірі досягти ідентичності виконання запитів і, як наслідок, знизити часові і трудовитрати на вирішення практичної задачі – забезпечення мобільності застосувань для різних СКБД.

Список литературы

1. Грофф Д. SQL: Полное руководство / Д. Грофф, П. Вайнберг. – К. : BHV, 2001. – 816 с.
2. Дейт Дж. Введение в системы баз данных / Дж. Дейт. – 8-е изд. – М. : Вильямс, 2005. – 1328 с)
3. Дунаев В. В. Базы данных. Язык SQL / В. В. Дунаев. – СПб. : БХВ-Петербург, 2006. – 288 с.
4. Пушкинов А. Ю. Введение в системы управления базами данных. Реляционная модель данных : учебн. пособ. / А. Ю. Пушкинов. – Уфа : Изд. Башкирского ун-та, 1999. – 108 с.
5. Тарасов О. В. Проектирование баз данных : навч. посіб. / О. В. Тарасов, Федько В. В., Лосев М. Ю. – Х. : Вид. ХНЕУ, 2011. – 200 с.
6. Форта Б. Освой самостоятельно SQL / Б. Форта. – М. : Вильямс, 2005. – 288 с.
7. Хендерсон К. Профессиональное руководство по Transact-SQL / К. Хендерсон. – СПб. : Питер, 2005. – 558 с.
8. Встроенные функции Oracle [Электронный ресурс]. – Режим доступа : <http://www.quizful.net/post/Oracle-inline-functions>.
9. Деревянко А. С. Язык SQL в диалектах Oracle и IBM DB2 [Электронный ресурс] / А. С. Деревянко – Режим доступа : <http://khpi-iip.mipk.kharkiv.edu/library/extent/dbms/sql/index.html>.
10. Кузнецов С. Д. Стандарты языка реляционных баз данных SQL: краткий обзор [Электронный

ресурс] / С. Д. Кузнецов – Режим доступа: http://citforum.ru/database/articles/art_2.shtml.

11. Карпова Т.С. Базы данных: модели, разработка, реализация. [Электронный ресурс] / Т.С. Карпова – Режим доступа : http://www.intuit.ru/department/database/dbmdi/5/dbmdi_5.html
12. Ривкин М. Тенденции развития универсальных коммерческих СУБД [Электронный ресурс] / М. Ривкин – Режим доступа : <http://citforum.univ.kiev.ua/database/articles/trends/>
13. Федоров А. Введение в базы данных. Часть 6. Введение в язык SQL [Электронный ресурс] / А. ЮЮФедоров, Н. Елманова – Режим доступа : <http://www.compress.ru/article.aspx?id=11944&iid=463#06/>.
14. Comparison of different SQL implementations [Электронный ресурс]. – Режим доступа : <http://troels.arvin.dk/db/rdbms/>
15. DB2. Справочник [Электронный ресурс]. – Режим доступа : <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/admin/c0006211.htm>.
16. SQL Dialects Reference [Электронный ресурс]. – Режим доступа : http://en.wikibooks.org/wiki/SQL_dialects_reference.
17. TOP (Transact-SQL) [Электронный ресурс]. – Режим доступа : <http://msdn.microsoft.com/ru-ru/library/ms189463.aspx>

Надійшла до редколегії 5.03.2013

Рецензент: д-р техн. наук, проф. В.О. Філатов, Харківський національний університет радіоелектроніки.

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ЯЗЫКА ОПРЕДЕЛЕНИЯ ДАННЫХ SQL В СОВРЕМЕННЫХ СУБД

А. В. Тарасов

В статье рассмотрены особенности синтаксиса языка манипулирования данными SQL в современных СУБД Access, Microsoft SQL Server, Oracle и DB2 с точки зрения мобильности скрипта по осуществлению операций поиска и модификации данных при переходе на другую платформу. Определены основные моменты, на которые следует обратить внимание при осуществлении подобной практической задачи.

Ключевые слова: язык SQL, поиск данных, манипулирование данными, NULL-значения, теоретико-множественные операции.

THE FEATURE OF USE SQL DATA DEFINITION LANGUAGE IN MODERN DBMS

Tarasov O.V.

The article describes the syntax features in SQL Data Manipulation Language of modern DBMS, such as Access, Microsoft SQL Server, Oracle and DB2 in terms of the mobility of the script to implement search and modification data operations when switching to another platform. Discussed the main points to be brought to account in the implementation of such a practical problem.

Keywords: SQL language, data search, data manipulation, NULL-value, set-theoretic operations

