

USING ODATA PROTOCOL IN DISTRIBUTED SYSTEMS WITH MOBILE DEVICES AS A CLIENT LAYER

A.O. Polyakov, V.M. Fedorchenko, A.P. Ischenko

The modern approach is to use mobile devices in the enterprise infrastructure. But such client requires access to large amounts of data, but they have small size of the memory on the device. Also the topicality and confidentiality of data require centralized storage of it. Connection to different types of database with data you want to get requires a middleware data model mapping. Odata protocol provides mechanisms for data access allows remapping of data, and provide selective access to data.

Keywords: distributed data processing, REST, data remapping, OData protocol.

УДК 004.655

О.В. Тарасов

Харківський національний економічний університет, Харків

**ОСОБЛИВОСТІ ВИКОРИСТАННЯ МОВИ ВИЗНАЧЕННЯ ДАНИХ SQL
У СУЧАСНИХ СКБД**

У статті розглянуті особливості синтаксису мови визначення даних SQL у сучасних СКБД Access, Microsoft SQL Server, Oracle та DB2 з точки зору мобільності скрипта по створенню основних об'єктів бази даних при переході на іншу платформу. Визначені основні моменти на які слід звернути увагу при здійсненні подібної практичної задачі.

Ключові слова: мова SQL, тип даних, посилальна цілісність, сурогатний ключ

Вступ

Сучасний етап розвитку суспільства вже не уявляється без використання інформаційних технологій. У цьому контексті використання баз даних є майже повсюдним. Невід'ємна і найважливіша частина будь-якої системи, що застосовує бази даних – мовні засоби, що забезпечують можливість доступу та дій над даними, визначення їхніх структур, способів використання та інтерпретації. Стандартом такої мови для усіх реляційних систем керування базами даних (СКБД) стала мова SQL (Structured Query Language – мова структурованих запитів). Вона 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 [4].

Кожний запит у мові SQL подається у вигляді спеціальної команди, яка може складатися з декількох речень. Зазвичай, усі команди SQL підрозділяють на три основні групи [12]:

1. Data Definition Language (DDL) – мова визначення даних, призначена для створення, модифікації та видалення таблиць і усієї бази даних.

2. Data Manipulation Language (DML) – мова маніпулювання даними) призначена для виконання основних операцій під час роботи з даними.

3. Data Control Language (DCL) – мова керування даними, призначена для забезпечення захисту

бази даних і застосовується для здійснення адміністративних функцій.

Крім того, часто виділяють ще 2 групи команд:

4. Transaction Control Language (TCL) – спеціальні оператори, які застосовуються для керування модифікацією бази даних, що здійснюють оператори DML (команди керування транзакціями).

5. Cursor Control Language (CCL) – оператори CCL використовуються для визначення курсору, а також для деяких інших операторів [12].

Незважаючи на те, що мова SQL стандартизована і, як декларується, більшість сучасних СКБД підтримують стандарт SQL92 [8], тим не менш реалії показують, що досягти у повній мірі переносу запитів на SQL з однієї системи на іншу дуже складно, а іноді і неможливо [14, 15].

Мета статті – проаналізувати характерні особливості використання мови визначення даних SQL у середовищі Microsoft Access, Microsoft SQL Server, Oracle та DB2. Перша з них є найбільш простою системою, хоча й досить потужною. Вона часто використовується як в навчальних цілях, так і кінцевими користувачами. Три інші системи є найбільш уживаними. Вони займають до 90% ринку комерційних СКБД [9]. При тестуванні використовувалися: Microsoft Access 2010; Microsoft SQL Server 2008 R2 (Express); Oracle XE 10g; DB2-Express C 9.7.

Основна частина

Оскільки основним об'єктом будь-якої реляційної бази даних є таблиця, то при її створенні мовою SQL необхідно мати відповіді на низку запитань:

- Як називатиметься таблиця?

- Як називатимуться стовпці (поля) таблиці?
- Які типи даних будуть закріплені за кожним стовпцем?
- Який розмір пам'яті потрібно виділити для зберігання кожного стовпця?
- У які стовпці таблиці обов'язково вводити дані?
- З яких стовпців складатиметься первинний ключ?
- Значення якого стовпця або стовпців є унікальними?
- Значення яких стовпців можуть мати значення "за замовчуванням"?

Іменування об'єктів у базі даних

Таким чином, перше з чим стикається розробник БД – це необхідність дати імена таблицям. А якщо подивитися на це більш загально, то взагалі виникає питання правил іменування будь-яких об'єктів у БД, оскільки у кожного об'єкта в базі даних має бути унікальне ім'я [8]. Основними іменованими об'єктами в реляційній базі є таблиці, стовпці і користувачі. Правила їхнього іменування були визначені ще в стандарті SQL1. У стандарті SQL2 цей перелік був значно розширений.

Відповідно до стандарту ANSI/ISO імена в SQL повинні містити від 1 до 18 символів, розпочинатися з букви і не містити пробілів або спеціальних символів пунктуації. У стандарті SQL2 максимальне число символів в імені збільшене до 128 [8]. Але на практиці підтримка імен у різних СКБД реалізована по різному. Вже на цьому першому кроці виникають проблеми. Допустима довжина імені таблиці істотно відрізняється: Access – 64 символи латиниці; Microsoft SQL Server – 128; Oracle – 30; DB2 – 128.

Подальший крок це назви стовпців таблиці, і тут ситуація аналогічна. Тому перший висновок – не давати таблицям та стовпцям надто довгі імена. Вони повинні бути довжиною до 30 символів, не містити спеціальних символів, але, звісно, відображати сутність тих даних, що зберігаються у таблиці.

Типи даних

Другий крок – визначення типу даних стовпця. Це ще більш важливий крок при створенні таблиці. Тип даних є фундаментальним поняттям у програмуванні і у мові SQL зокрема [8]. У стандарті SQL1 описано лише мінімальний набір типів даних, які можна використати для подання інформації в реляційній базі даних. Вони практично підтримуються в усіх комерційних СКБД. Стандарт SQL2 додав у цей перелік рядки змінної довжини, значення дати і часу та ін. До базових типів даних, що використовуються у SQL, відносяться такі [3; 8]:

CHARACTER(n), або CHAR(n) – символні рядки постійної довжини з n символів.

NUMERIC[(n, m)] – точні числа, де n – загальна кількість цифр в числі; m – кількість цифр зліва від коми.

DECIMAL[(n, m)] – точні числа (аналог NUMERIC), де n – загальна кількість цифр в числі; m – кількість цифр зліва від коми.

INTEGER або INT – цілі числа.

SMALLINT – цілі числа меншого діапазону.

FLOAT[(n)] – числа великої точності, що зберігаються у формі з плаваючою комою, де n – число байтів, що резервується для зберігання одного числа. Діапазон чисел визначається конкретною реалізацією.

REAL – дійсний тип чисел, який відповідає числам з плаваючою комою меншої точності, ніж FLOAT.

DOUBLE PRECISION – дійсний тип чисел з визначеною в реалізації точністю, яка більша, ніж для REAL.

Аналіз використання цих типів при створенні таблиць показує, що тільки Access не в повній мірі підтримує ці типи даних (табл.1).

У стандарті SQL92 були додані ще такі типи даних [1, 3, 8]:

VARCHAR(n) – рядки символів змінної довжини до n символів включно.

BIT(n) – рядок бітів постійної довжини.

BIT VARYING(n) – рядок бітів змінної довжини.

DATE – календарна дата.

TIMESTAMP(точність) – дата і час.

INTERVAL – часовий інтервал.

Багато сучасних СКБД також підтримують спеціальні поля для зберігання дуже великих за розміром об'єктів а також грошової одиниці:

BINARY LARGE OBJECT(n) (BLOB) – тип даних, що дозволяє зберігати великі неструктуровані бінарні об'єкти, наприклад звук і відео.

CHARACTER LARGE OBJECT(n) (CLOB) – тип даних, що дозволяє зберігати великі символні об'єкти.

MONEY – для зберігання даних про вартість товарів чи заробітну плату робітників.

Доступність вживання цих типів даних наведена у табл.1.

Відсутність позначки підтримки того чи іншого типу у табл. 1 та 2 зовсім не означає, що подібні типи, за своєю сутністю не підтримуються. Наприклад, замість DOUBLE PRECISION у Access та DB2 можна писати просто DOUBLE, замість BIT VARYING(n) у Microsoft SQL Server – VARBINARY, тощо.

З переліком типів даних окремих СКБД можна ознайомитись у [2,5,6,7,11,13]. Однак саме різний синтаксис щодо використання типів даних істотно перешкоджає мобільності програм, в яких використовується SQL.

Іншим підводним каменем може служити трактовка стандартних типів даних різними СКБД. Найбільш гостро ця проблема виникає з Oracle для числових типів (табл. 2), де NUMBER(n) – число з n значущими десятковими розрядами; FLOAT(n) – дійсне число з n значущими двійковими розрядами.

Таблиця 1
Підтримка базових типів даних СКБД

	Access	MS SQL	Oracle	DB2
CHAR	+	+	+	+
NUMERIC(n,m)		+	+	+
NUMERIC	+	+	+	+
DECIMAL(n,m)		+	+	+
INTEGER	+	+	+	+
SMALLINT	+	+	+	+
FLOAT	+	+	+	+
REAL	+	+	+	+
DOUBLE PRECISION		+	+	+
VARCHAR(n)	+	+	+	+
BIT		+		
BIT VARYING(n)				
DATE	+	+	+	+
TIMESTAMP	+	+	+	+
INTERVAL			+	
BLOB			+	+
CLOB			+	+
MONEY	+	+		

Таблиця 2
Трактовка Oracle стандартних числових типів даних

Тип у Create Table	Результат у Oracle
NUMERIC	NUMBER(38)
INTEGER	NUMBER(38)
SMALLINT	NUMBER(38)
FLOAT	FLOAT(126)
REAL	FLOAT(63)
DOUBLE PRECISION	FLOAT(126)

Обмеження на значення стовпців

Під час створення таблиці або модифікації її схеми є можливість задавати додаткові обмеження на ті значення, які можна вводити в окремі поля таблиці. Такі обмеження зазвичай описуються особливою конструкцією CHECK, яка дозволяє окреслити певні рамки, яким повинно відповідати значення стовпця. Якщо ж в обмеженні задіяні два чи більше стовпців, то конструкція CHECK задається при цьому в окремому обмеженні CONSTRAINT.

Таку конструкцію підтримують усі СКБД, за винятком Access.

Обмеження унікальності

При створенні таблиці може виникнути ситуація, коли декілька атрибутів чи груп атрибутів претендують на унікальність. Наприклад, такими атрибутами можуть бути "Ідентифікаційний код людини" та пара атрибутів "Серія паспорта"+"Номер паспорта". У цьому разі маємо два потенційні ключі, один з яких стає первинним, наприклад "Код", а інший – альтернативним ключем (серія + номер).

Для позначення унікальності атрибутів існує зарезервоване слово UNIQUE. Однак атрибути, позначені як UNIQUE, на відміну від PRIMARY KEY, можуть набувати значення NULL. Тому в разі необхідності треба вказати ознаку NOT NULL. Специфічно у цьому сенсі визначення первинного ключа у DB2. Для атрибута PRIMARY KEY автоматично не підтримується ознака NOT NULL, тому її треба вказати явно.

Значення за замовчуванням

Якщо під час додавання рядка у таблицю для якогось стовпця не вказано значення, то в цьому разі воно визначиться за замовчуванням. Найбільш поширеним значенням за замовчуванням є NULL. Для визначення іншого значення використовується конструкція DEFAULT, яка не підтримується мовою SQL у Access.

Посилальна цілісність

У визначенні посилальної цілісності беруть участь два відношення – батьківське та дочірнє. У кожному з них можливі три операції – вставка, оновлення, видалення в результаті яких посилальна цілісність може порушитися. З аналізу цих шести можливих варіантів видно, що в чотирьох із них в принципі може бути порушена посилальна цілісність, а саме під час виконання операцій: оновлення та видалення кортежу у батьківському відношенні; вставка та оновлення кортежу в дочірньому відношенні.

Стратегії підтримки посилальної цілісності визначають дії СКБД у разі, коли під час виконання однієї з чотирьох перерахованих операцій може статися порушення посилальної цілісності. Вони включають такі дії: RESTRICT (обмежити) – не дозволити виконання операції, що призводить до порушення посилальної цілісності; CASCADE (каскадувати) – дозволити виконання операції, але внести при цьому необхідні поправки в інші відношення так, щоб не допустити порушення посилальної цілісності і зберегти усі наявні зв'язки; SET NULL (встановити в NULL) – дозволити виконання необхідної операції, але усі некоректні значення зовнішніх ключів, що з'являються, змінювати на NULL-значення; SET DEFAULT (встановити за замовчуванням) – дозволити виконання необхідної операції, але усі некоректні значення зовнішніх ключів, що з'являються, змінювати на значення, яке прийняте за замовчуванням; IGNORE (ігнорувати) – виконувати операції, не звертаючи уваги на порушення посилальної цілісності.

Залежно від СКБД ті чи інші конструкції підтримки посилальної цілісності у мові SQL можуть відрізнятися, а окремі конструкції – не працювати.

Наприклад у Access підтримки посилальної цілісності задається тільки через будівник схеми БД і не підтримується у мові SQL. Для MS SQL SERVER стратегія ON UPDATE RESTRICT не вказується (

виникає помилка), можна тільки зазначити ON UPDATE NO ACTION. Аналогічним чином для цієї ж БД не вказується ON DELETE RESTRICT, тощо. Таким чином питання підтримки посилальної цілісності потребують особливої уваги у разі переходу з однієї СКБД на іншу.

Сурогатні ключі

При створенні таблиць у базі даних часто трапляються ситуації, коли у структуру таблиці до вже наявних інформаційних стовпців додають спеціальне службове поле, єдине призначення якого – служити первинним ключем. Значення цього поля не утворюється на основі яких-небудь інших даних з БД, а генерується штучно. Такий штучно утворений первинний ключ таблиці називають сурогатним ключем [10]. Для створення сурогатних ключів у таблиці СКБД Access використовує тип даних Counter (лічильник), а в MS SQL SERVER найчастіше пишуть integer identity (1,1). В той же час у СКБД Oracle треба створити об'єкт SEQUENCE (послідовність), яку використовувати при додаванні нових рядків у таблицю, або застосовувати тригер.

Домен

Домен – це набір допустимих значень для одного або декількох атрибутів. Домен визначає усі потенційні значення, які можуть бути присвоєні атрибуту [3]. Домен створюється за допомогою команди CREATE DOMAIN, що з'явилася в SQL2. Незважаючи на те що команда створення доменів обумовлена стандартом SQL, спроба виконати подібну команду у MS SQL Server, Oracle, Access та DB2 генерує помилку. Тому слід визнати, що хоч команда CREATE DOMAIN і є досить корисною, використання її у сучасних СКБД, з точки зору перенесення застосувань на різні платформи поки не є виправданим.

Висновки

Аналіз застосування мови визначення даних SQL для створення основних об'єктів БД – таблиць показав, що стандартизація SQL в реальних умовах застосування комерційних СУБД є радше "міфом" або дуже не близькою перспективою. Домогтися мобільності скриптів по створенню об'єктів БД деколи дуже складно, а найчастіше і неможливо в повній мірі, враховуючи широкий діапазон застосовуваних типів даних і особливості синтаксису окремих конструкцій. Тим не менше, якщо використовувати базові типи даних і спрощений синтаксис конструкцій підтримки посилальної цілісності можна в значній мірі знизити часові і трудовитрати на здійснення подібної практичної задачі.

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ЯЗЫКА ОПРЕДЕЛЕНИЯ ДАННЫХ SQL В СОВРЕМЕННЫХ СУБД

А.В. Тарасов

В статье рассмотрены особенности синтаксиса языка определения данных SQL в современных СУБД Access, Microsoft SQL Server, Oracle и DB2 с точки зрения мобильности скрипта по созданию основных объектов базы данных при переходе на другую платформу. Определены основные моменты, на которые следует обратить внимание при осуществлении подобной практической задачи.

Список літератури

1. Боуман Д. *Практическое руководство по SQL* / Д. Боуман, С. Эмерсон, М. Дарновски. – М. : Вильямс, 2002. – 352 с.
2. Виейра Р. *Программирование баз данных. Microsoft SQL Server 2005. Базовый курс* / Р. Виейра. – М. : Вильямс, 2007. – 832 с.
3. Грофф Д. *SQL: Полное руководство* / Д. Грофф, П. Вайнберг. – К. : BHV, 2001. – 816 с.
4. Дейт Дж. *Введение в системы баз данных* / Дж. Дейт. – 8-е изд. – М. : Вильямс, 2005. – 1328 с.
5. Кайт Т. *Oracle для профессионалов* / Т. Кайт. – СПб. : ДиасофтЮП, 2003. – 672 с.
6. Хендерсон К. *Профессиональное руководство по Transact-SQL* / К. Хендерсон. – СПб. : Питер, 2005. – 558 с.
7. 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>.
8. 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.
9. Ривкин М. *Тенденции развития универсальных коммерческих СУБД* [Электронный ресурс] / М. Ривкин – Режим доступа: <http://citforum.univ.kiev.ua/database/articles/trends/>
10. *Суррогатный ключ* [Электронный ресурс]. – Режим доступа: http://access.avorut.ru/publ/bazy_dannykh_osnovnye_ponjatija/surrogatnyj_kljuch/46-1-0-35.
11. *Типы данных DB2* [Электронный ресурс]. – Режим доступа: http://www.sd-company.su/article/sql_data_type_db2
12. Федоров А. *Введение в базы данных. Часть 6. Введение в язык SQL* [Электронный ресурс] / А. Федоров, Н.Елманова – Режим доступа: <http://www.compress.ru/article.aspx?id=11944&iid=463#06>.
13. *Эквивалентные типы данных ANSI SQL* [Электронный ресурс]. – Режим доступа: <http://office.microsoft.com/ru-ru/access-help/HP001032229.aspx>.
14. *Comparison of different SQL implementations* [Электронный ресурс]. – Режим доступа: <http://troels.arvin.dk/db/rdbms/>
15. *SQL Dialects Reference* [Электронный ресурс]. – Режим доступа: http://en.wikibooks.org/wiki/SQL_dialects_reference.

Надійшла до редакції 1.10.2012

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

Ключевые слова: язык SQL, тип данных, ссылочная целостность, суррогатный ключ.

THE FEATURE OF USE SQL DATA DEFINITION LANGUAGE IN MODERN DBMS

O.V. Tarasov

The article describes the syntax features in SQL Data definition language of modern DBMS, such as Access, Microsoft SQL Server, Oracle and DB2 in terms of the mobility of the script to create a main database objects during the transition to another platform. Discussed the main points to be brought to account in the implementation of such a practical problem.

Keywords: SQL language, data type, referential integrity, a surrogate key.