

11. Релационен език SQL

Лекционен курс "Бази от данни"

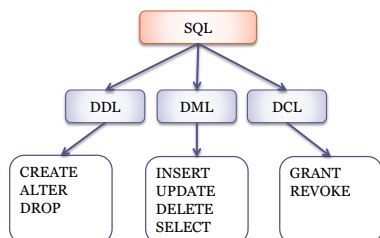
Въведение

SQL (Structured Query Language):

- конкретна реализация на релационния модел;
- стандарт за релационни езици;
- поддържа се от повечето развойни системи, които са днес на пазара.

- Ще разгледаме едно ядро на езика, което съдържа основните му аспекти, като имаме предвид, че ще разгледаме стандарт на езика, познат като SQL/92. Официалното му име - International Standard Database Language SQL (1992);
- никой продукт не поддържа пълния SQL/92;
- в SQL се използват понятията:
 - таблица (за релация);
 - ред (за запис);
 - колона (за атрибут).

Класификация на операторите в SQL



Дефиниране на данни

В този раздел ще се запознаем с DDL-операторите на езика SQL.

DDL (Data Definition Language) –компонент на SQL за дефиниране на базовите обекти от данни. От гледна точка на потребителите основните DDL-оператори са следните:

- CREATE TABLE
- CREATE DOMAIN
- ALTER DOMAIN
- ALTER TABLE
- DROP TABLE
- DROP DOMAIN

Домейни

SQL домейните се различават значително от тези на релационния модел - единственото предназначение на домейните в SQL е да разрешават една проста спецификация на типовете данни, така че веднъж дефинирани да могат да се използват за различни колони в различни таблици.

Домейни

Основни разлики между РМ-домейн и SQL-домейн:

- не се изисква непременно да се използват SQL-домейни. Колоните могат да се дефинират директно върху прости типове;
- в SQL не се поддържа домейн върху домейн - SQL-домейн съдържа само елементи от системните типове данни;
- SQL не поддържа възможност за дефиниране на операции, които се прилагат върху даден домейн;
- SQL домейните не поддържат концепцията за:
 - подтипове;
 - супертипове;
 - наследяване.

Домейни

Синтаксис на оператора:

```
CREATE DOMAIN domain-name data-type
    [ default-definition ]
    [ domain-constraint-definition-list ];
```

Базови таблици

За създаване на таблици общият формат на оператора е следният:

```
CREATE TABLE base-table (base-table-element-commalist);
```

base-table-element ::= column-definition | base-table-constraint-definition

column-definition ::= column representation [default-definition]

column representation ::= тип данни или домейн

default-definition ::= препокрива, ако има такива за домейна

Базови таблици

base-table-constraint-definition:

- дефиниция на ключ-кандидат
 - UNIQUE (column-commalist);
 - PRIMARY KEY (column-commalist)
- дефиниция на външен ключ;
 - FOREIGN KEY (column-commalist)
 - REFERENCES base-table [(column-commalist)]
 - [ON DELETE option]
 - [ON UPDATE option]

Базови таблици

option ::= NO ACTION | CASCADE | SET NULL | SET DEFAULT

кореспондиращи съответно с режимите:

NO ACTION – RESTRICTED

CASCADE – CASCADES

SET NULL – NULLIFIES

SET DEFAULT – стойности по подразбиране

- дефиниция на check constraints
 - CHECK (conditional-expression)

Базови таблици

```
CREATE TABLE FACULTY
(
  FACULTY_ID      int      not null,
  FACULTY_NAME    varchar(50) not null,
  YEAR_ESTABLISHED int      null,
  EMPLOYEE_NUMBER int      null,
  constraint PK_FACULTY primary key (FACULTY_ID),
  constraint CKC_EMPLOYEE_NUMBER check (EMPLOYEE_NUMBER is null
                                         or (EMPLOYEE_NUMBER between 1 and 2000))
)

create table LECTURER
(
  LECTURER_ID int      not null,
  FACULTY_ID   int      not null,
  FIRST_NAME   varchar(15) not null,
  LAST_NAME    varchar(15) not null,
  constraint PK_LECTURER primary key (LECTURER_ID)
```

Базови таблици

Променяме таблицата LECTURER като добавяме ограничение за външен ключ на полето FACULTY_ID със следните опции:

- при изтриване на реда със съответната стойност на първичен ключ операцията се разширява каскадно като се изтриват "свързаните" записи в таблицата LECTURER;
- при опит за промяна на стойността на първичния ключ в референцираната таблица ако той има кореспондиращи в LECTURER операцията се отхвърля.

```
ALTER TABLE LECTURER
ADD CONSTRAINT FK_LLECTURER_FACULTY foreign key (FACULTY_ID)
references FACULTY (FACULTY_ID)
on delete cascade
on update no action
```

Опериране с данните

Вторият компонент на SQL е Data Manipulation Language (DML). SQL поддържа 4 DML-оператора:

- SELECT
- UPDATE
- DELETE
- INSERT

Опериране с данните

Операциите за извличане на данни са релационни изрази с различна сложност. Общата форма на оператора е:

```
SELECT [DISTINCT] field(s)
[INTO new_table]
FROM table(s)
[WHERE predicate]
[GROUP BY field(s) [HAVING predicate]]
[ORDER BY field(s)]
```

Опериране с данните

INSERT INTO table [(field [, field] ...)]
VALUES (constant [, constant] ...);

или

INSERT INTO table [(field [, field] ..)]
SELECT ... FROM ... WHERE ... ;

Опериране с данните

UPDATE table SET field = expr [, field = expr] ...
[WHERE predicate]

DELETE FROM table
[WHERE predicate]

Опериране с данните

SQL поддържа едно множество от *агрегиращи*
функции:

- COUNT
- SUM
- AVG
- MAX
- MIN

Контрол на достъпа до данните

Третият компонент на SQL е Data Control Language (DCL).

- GRANT – оторизира потребител(и) да изпълняват операции с обектите от базата;
- REVOKE – премахва разрешението за операции.

Контрол на транзакциите

- START TRANSACTION (BEGIN WORK, BEGIN TRANSACTION) - маркира начало на транзакция;
- SAVE TRANSACTION (SAVEPOINT) – съхранява данните в базата в тази точка на транзакцията;
- COMMIT – край на транзакцията, всички промени стават постоянни;
- ROLLBACK – край на транзакцията, отхвърля всички направени промени.

Вграден SQL (Embedded SQL)

SQL - операторите могат да се използват:

- интерактивно;
- като част от приложна програма – SQL операторите физически се вграждат в първичния код на програмата.

До сега разгледахме интерактивното използване на езика.

Като език за програмиране SQL обикновено не съществува самостоятелно, а е вграден в друг език за програмиране (приемник). Напр., вграден SQL/PL1, SQL/FORTRAN, SQL/C, SQL/Java ...

Вграден SQL (Embedded SQL)

Основен принцип на SQL - изразните му средства в интерактивен режим могат да се използват без ограничения в програмния режим.

Интерпретация:

- вградените SQL оператори се маркират така, че те могат да бъдат различени от езика-приемник (host-language);
- изпълним SQL - оператор може да се запише там, където се допуска изпълним оператор на базовия език;
- SQL - операторите могат да включват обръщения към host-променливи;
- host-променливите трябва да имат съвместими типове със съответните им SQL-полета.

Стандарти

Година	Име	Псевдоним	Описание
1986	SQL-86	SQL-87	Първият формализиран от ANSI.
1989	SQL-89	FIPS 127-1	Кратка ревизия, одобрен като Federal Information Processing Standard 127-1
1992	SQL-92	SQL2, FIPS 127-2	Главна ревизия (ISO 9075).
1999	SQL:1999	SQL3	Добавени регулярни изрази, рекурсивни заявки, тригери, комплексни типове и др. като обектно-ориентирани характеристики.
2003	SQL:2003		XML-обработка, колони с автоматично генерирани стойности.
2006	SQL:2006		ISO/IEC 9075-14:2006 дефинира начините, чрез които се съхранява и манипулира XML чрез SQL (XQuery).
2008	SQL:2008		Легализира ORDER BY извън дефиницията на курсори; INSTEAD OF тригери и др.

Реализации на SQL стандарта

Източник	Име	Пълно име
ANSI/ISO Standard	SQL/PSM	SQL/Persistent Stored Modules
Interbase/Firebird	PSQL	Procedural SQL
IBM	SQL PL	SQL Procedural Language (реализира SQL/PSM)
Microsoft/Sybase	T-SQL	Transact-SQL
Mimer SQL	SQL/PSM	SQL/Persistent Stored Module (реализира SQL/PSM)
MySQL	SQL/PSM	SQL/Persistent Stored Module (реализира SQL/PSM)
Oracle	PL/SQL	Procedural Language/SQL (базиран на Ada)
PostgreSQL	PL/pgSQL	Procedural Language/PostgreSQL Structured Query Language (базиран на Oracle PL/SQL)