

Cơ sở dữ liệu nâng cao

Toàn vẹn và cơ sở dữ liệu active

Đỗ Thanh Nghị
dtnghi@cit.ctu.edu.vn

Cần Thơ
11-10-2016

Toàn vẹn và cơ sở dữ liệu active

■ Mục tiêu

- điều khiển toàn vẹn ngữ nghĩa
- đưa khái niệm trigger
- cài đặt trong mô hình quan hệ
- minh họa với SQL3



Toàn vẹn ngữ nghĩa

Định nghĩa

■ Toàn vẹn ngữ nghĩa

- dữ liệu phải hợp lệ và nhất quán
- sự hợp lệ và nhất quán được định nghĩa với sự trợ giúp ràng buộc toàn vẹn

■ Ràng buộc toàn vẹn

- luật được định nghĩa bởi người thiết kế CSDL
- cho phép định nghĩa tính hợp lệ của dữ liệu
- phải được hệ thống kiểm soát
- CSDL phải luôn thỏa tập các ràng buộc

Vấn đề

■ Định nghĩa ràng buộc

- ngôn ngữ định nghĩa ràng buộc
 - tích hợp vào SQL, mở rộng
- khi nào, làm sao định nghĩa ràng buộc?
 - nhiều cách tiếp cận khác

■ Kiểm chứng các ràng buộc

- khi nào?
- làm sao kiểm chứng?
- giải thuật nào?

Các thể thức ràng buộc toàn vẹn

Thể thức

- **Ràng buộc cấu trúc**
 - đặc thù cho một mô hình dữ liệu
- **Ràng buộc về xử lý**
 - sự tiến hóa dữ liệu
- **Ràng buộc phụ thuộc**
 - phụ thuộc hàm, bao hàm, đa trị
- **Khẳng định tổng quát**
 - luật quản lý và/hoặc biểu thức số học

Ràng buộc cấu trúc

■ Ràng buộc cấu trúc

- đặc thù cho mô hình dữ liệu
- trình bày các đặc tính cơ bản của mô hình

■ Mô hình quan hệ

- khóa duy nhất
- không rỗng
- phụ thuộc tham chiếu
- ràng buộc miền

Ràng buộc cấu trúc của mô hình quan hệ

■ Khóa duy nhất

- một hay nhóm thuộc tính được định nghĩa là khóa của quan hệ phải có tính duy nhất

thuộc tính NoAp của quan hệ Apartment xác định một và chỉ một mẫu tin của quan hệ

■ Không rỗng

- thuộc tính không nhận giá trị rỗng

thuộc tính NoAp của quan hệ Apartment không nhận giá trị rỗng

Ràng buộc cấu trúc của mô hình quan hệ

■ Phụ thuộc tham chiếu

- sự phụ thuộc giữa những giá trị của thuộc tính (nhóm thuộc tính) của một quan hệ với những giá trị của thuộc tính (nhóm thuộc tính) của một quan hệ khác

thuộc tính NoAp của quan hệ Room tham chiếu đến thuộc tính NoAp của quan hệ Apartment

■ Ràng buộc miền

- thuộc tính của một quan hệ chỉ nhận giá trị thuộc miền giá trị

thuộc tính Name của quan hệ Room có thể nhận giá trị nằm thuộc {phòng ngủ, phòng khách, phòng ăn, bếp, phòng tắm, văn phòng}

Ràng buộc về xử lý

■ Theo thời gian

- đặc tả sự tiến hóa dữ liệu khi cập nhật

lương của nhân viên không thể giảm, giá trị mới phải lớn hơn giá trị cũ

■ Tập hợp

- đặc tả luật để kiểm soát tập hợp dữ liệu

lương trung bình không thể nhỏ hơn 15 000\$

Ràng buộc phụ thuộc

■ Phụ thuộc hàm

- đặc tả mối liên hệ tồn tại giữa 2 hay những nhóm thuộc tính, tổng quát hóa một khóa

code postal xác định thành phố

■ Phụ thuộc bao hàm

- đặc tả những giá trị của một thuộc tính được bao hàm trong tập hợp giá trị của thuộc tính khác

■ Phụ thuộc đa trị

- hệ quả của sự giới hạn đơn trị
- đặc tả phụ thuộc của một thuộc tính đến thuộc tính thứ hai và độc lập với thuộc tính thứ 3

Định nghĩa ràng buộc

Định nghĩa ràng buộc

■ Khi nào định nghĩa ràng buộc?

- thời điểm tạo CSDL

bằng cách lệnh của ngôn ngữ định nghĩa dữ liệu

■ Làm sao định nghĩa ràng buộc?

- bằng các từ khóa cho từng ràng buộc

NOT NULL, PRIMARY KEY, UNIQUE, ...

- với những khẳng định

SQL92: CREATE ASSERTION

Bảo toàn những ràng buộc

■ Định nghĩa định dạng cho các ràng buộc

- trình bày thông tin đặc tả ràng buộc
quan hệ, kiểu ràng buộc...

■ Lưu trữ ràng buộc

- trong các metadata
CSDL mô tả CSDL khác và các lược đồ quan hệ
- tổ chức, chỉ mục ràng buộc
tăng tốc quá trình tìm kiếm ràng buộc

Giải pháp thương mại

■ Chuẩn SQL: SQL92

- đề xuất cho các kiểu ràng buộc khác nhau
- NOT NULL, UNIQUE, CHECK, CONSTRAINT, ASSERTION, PRIMARY, KEY, REFERENCES

■ Trong những hệ thống

- cài đặt gần như SQL92
- sử dụng những luật, thủ tục và triggers
- sử dụng các view

Ràng buộc toàn vẹn trong Oracle

■ Sử dụng các lệnh định nghĩa quan hệ

- mệnh đề CONSTRAINT trong lệnh CREATE TABLE và ALTER TABLE

■ Ràng buộc

- không rỗng : NOT NULL
- duy nhất : UNIQUE
- khóa : PRIMARY KEY
- ràng buộc miền : CONSTRAINT CHECK
- ràng buộc tham chiếu : REFERENCES

Ví dụ 1

```
CREATE TABLE    student (  
    no          integer NOT NULL PRIMARY KEY,  
    lname      char(20),  
    fname      char(20),  
    addr       char(40),  
    type       char(15)  
CONSTRAINT check_type  
CHECK (type in ('graduated', 'undergraduated')));
```

Chú ý

■ Diễn đạt ràng buộc

- không rỗng

NOT NULL

- khóa

PRIMARY KEY

- miền

CONSTRAINT và CHECK

■ check

- cho phép diễn đạt những điều kiện trên những thuộc tính của một quan hệ

Ví dụ 2

```
CREATE TABLE inscription (  
  (no          integer  
  CONSTRAINT fk_no REFERENCES student(no)  
  ON DELETE CASCADE,  
  nc          integer  
  CONSTRAINT fk_nc REFERENCES course(nc),  
  dateins     date );
```

Chú ý

■ Ràng buộc

- tham chiếu : **CONSTRAINT REFERENCES**
- tham chiếu **COURSE** và **INSCRIPTION**
- tham chiếu **STUDENT** và **INSCRIPTION**

■ Xóa cascade

- **ON DELETE CASCADE** : khi xóa mẫu tin của **STUDENT**, những đăng ký của sinh viên này cũng bị xóa theo

SQL và định nghĩa ràng buộc

■ Tạo các quan hệ

- có thể định nghĩa một số ràng buộc
ràng buộc cấu trúc

■ Thay đổi lược đồ quan hệ

- có thể thay đổi ràng buộc
thêm, xóa, thay đổi ràng buộc

■ Ràng buộc khác

- cần sử dụng cơ chế khác
Triggers



Toàn vẹn ngữ nghĩa
Kiểm soát ràng buộc

Vấn đề

■ Khi nào kiểm tra ràng buộc?

- trước, trong khi hay sau cập nhật?
- bắt đầu hay kết thúc giao dịch?

■ Làm thế nào để kiểm chứng ràng buộc?

- trên toàn bộ CSDL?
mất thời gian, chi phí cao
- trên tập con CSDL?
làm sao để xác định?

■ Chi phí?

- phức tạp của một số kiểm chứng
- thời gian thực thi kiểm chứng

Khi nào kiểm chứng?

■ Khi chèn, xóa, thay đổi mẫu tin

- tất cả ràng buộc của quan hệ liên quan đến sự cập nhật phải được kiểm chứng

cập nhật không thỏa ràng buộc bị bỏ

■ Sự cần thiết

- mỗi lần cập nhật

tìm tất cả các ràng buộc

kiểm chứng mỗi ràng buộc

Khi nào kiểm chứng?

■ Phụ thuộc vào kiểu ràng buộc

- ràng buộc miền

khi thao tác chèn, thay đổi, xóa

- ràng buộc tham chiếu

có thể kiểm tra vào cuối giao dịch

■ Phụ thuộc vào thứ tự thao tác

- kiểm chứng cuối giao dịch

ràng buộc tham chiếu

trình tự sau phép chèn

- quá khó, chi phí cao

thường không khả thi

Làm sao kiểm chứng?

- **Kiểm chứng tập các mẫu tin tương ứng với một ràng buộc**
 - toàn bộ CSDL
 - tập con của CSDL
- **Tập mẫu tin thao tác và kiểm chứng phải được tối thiểu hóa**
 - xác định tập con của CSDL liên quan đến việc cập nhật
 - có thể tùy thuộc vào kiểu ràng buộc khác nhau
- **Tiếp cận**
 - phát hiện
 - xét đoán

Phát hiện

■ Nguyên lý

- mọi cập nhật m được thực hiện;
- trạng thái của CSDL D được thay đổi thành D_m
- nếu D_m được phát hiện không nhất quán, ta có thể phục hồi lại trạng thái D

■ Khó khăn

- tập các mẫu tin cần kiểm chứng có thể là toàn bộ CSDL
- cần thực hiện lại giao dịch khi phát hiện sự thiếu nhất quán dữ liệu

Xét đoán

■ Nguyên lý

- một thay đổi m chỉ được thực hiện nếu trạng thái CSDL kết quả đảm bảo nhất quán

■ Khái niệm kiểm tra trước

- tìm một điều kiện để kiểm chứng trước
- tối thiểu số mẫu tin cần kiểm chứng

■ Giả thiết

- bắt đầu giao dịch, CSDL nhất quán
tất cả ràng buộc được kiểm chứng
- kết thúc giao dịch, CSDL phải nhất quán

Tiếp cận bằng quan hệ khác biệt

■ Một thay đổi quan hệ R

- mẫu tin chèn vào R^+
- mẫu tin bị xóa R^-

■ Kiểm chứng cập nhật

- kết hợp phép hội và trừ

$$R = (R \text{ UNION } R^+) \text{ DIFFERENCE } R^-$$

■ Ý tưởng

- định nghĩa những kiểm tra trên R^+ và R^-
- cực tiểu số lần kiểm chứng
- cần phân tích ràng buộc

Ví dụ

■ Ràng buộc tham chiếu Student-Inscription

- kiểm tra trước Inscription+

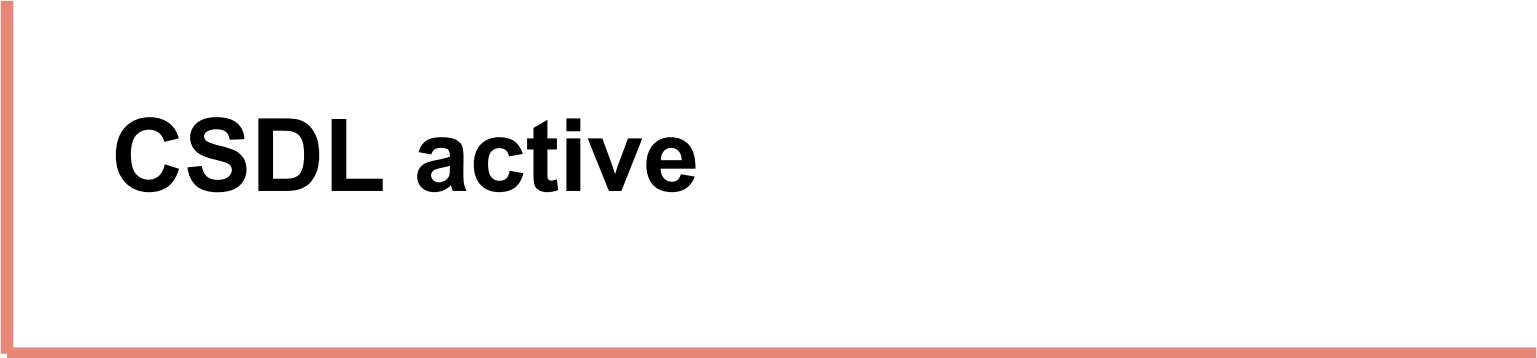
Inscription+.No=Student.No

- kiểm tra trước Inscription-

- kiểm tra trước Student+

- kiểm tra trước Student-

COUNT (Inscription.No WHERE Inscription.No=Student-.No)=0

A thick red line forming an L-shape, starting with a vertical segment on the left and a horizontal segment extending to the right.

CSDL active

Mục tiêu

■ Giới thiệu về sự tích cực trong CSDL

- bắt sự kiện (event)
- định nghĩa phép toán kích hoạt

■ Tích hợp cơ chế điều khiển

- cung cấp cơ chế sử dụng để giải quyết các vấn đề khác nhau
- cho phép tăng thao tác điều khiển và định nghĩa các thao tác điều khiển mới

Thực hiện

■ Định nghĩa

- trigger là luật dạng

nếu *<sự kiện (event)>* thì *<hành động (action)>*

- khi sự kiện xảy ra thì hệ thống kích hoạt hành động

■ Cơ chế trigger

- cho phép định nghĩa các « daemon »
- kích hoạt và điều khiển thực thi các « daemon »
- đảm nhận thực thi tốt tập các hành động

Hoạt động trong CSDL

■ Trong ứng dụng

- mô hình hóa sự tích cực
- cho phép tự động hóa một số hành động và phép toán
- *ví dụ* : quản lý giao dịch chứng khoán

nếu	quantity of stock < value
thì	generate a new command

Hoạt động trong CSDL

■ Trong hệ thống

- quản lý sự toàn vẹn
- quản lý view cụ thể
- cải tiến hiệu quả
- *ví dụ*

trường hợp cập nhật một quan hệ, kích hoạt cập nhật view tương ứng

Những hệ thống trigger

■ Vấn đề

- trình bày trigger
- mô hình thực thi tập các triggers

■ Nhiều trình bày có thể của một trigger

- mô hình EA : Event ==> Action
- mô hình ECA :
khi Event nếu <Condition> thì <Action>
- mô hình khác : tập luật

Event, condition, action

■ Sự kiện

- **sự kiện ngoài hay trong**
phép chèn trong một quan hệ
thời điểm cho trước
- **tham số của sự kiện**
quan hệ có liên quan, thời điểm
- **ngữ cảnh của sự kiện**
dữ liệu cần thiết cho đánh giá luật

■ Kiểu sự kiện

- đơn giản
- phức hợp

Event, condition, action

■ Điều kiện

- diễn đạt trên những dữ liệu của CSDL
- tùy chọn

■ Hành động

- mã thực thi khi điều kiện được thỏa mãn
- những thao tác trên CSDL (truy vấn SQL)
- thao tác giao dịch (kiểm chứng, từ bỏ)

Mô hình thực thi

■ Khi nào hệ thống kích hoạt trigger?

- trước sự kiện

BEFORE

- sau sự kiện

AFTER

- diễn ra sự kiện

INSTEAD

■ Kích hoạt hành động trigger như thế nào?

- chỉ một lần trên các dữ liệu liên quan

STATEMENT

- từng mẫu tin một

ROW

Mô hình thực thi

■ Kích hoạt tập hợp triggers

- làm sao hệ thống quản lý tập hợp triggers và sự kích hoạt?
- nếu sự kiện kích hoạt nhiều triggers, thực thi tất cả các triggers? theo thứ tự ưu tiên?

xếp gỏi lên nhau của các ngữ cảnh thực thi

■ Kích hoạt một cách đệ quy

- kích hoạt một trigger có thể kích hoạt tập triggers khác
 - làm sao kết thúc?
 - giới hạn độ sâu

Triggers và HQTCSDDL quan hệ

■ Đề nghị trong SQL 3

- lệnh tạo trigger

CREATE TRIGGER

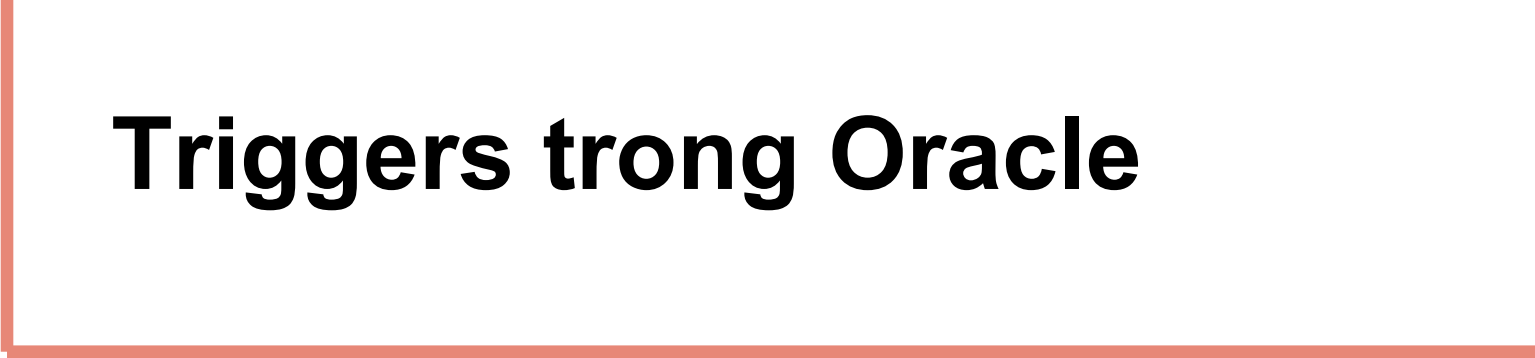
- định nghĩa biến về ngữ cảnh

OLD/NEW

- những hành động: truy vấn SQL trừ

lệnh thực hiện kết nối (connection)

lệnh định nghĩa lược đồ



Triggers trong Oracle

Triggers trong Oracle

- Tương thích chuẩn SQL3
- Định nghĩa bộ ba (E, C, A): event, condition, action
- Cung cấp khả năng tốt nhất để kiểm chứng ràng buộc
- Cho phép giới thiệu sự năng động
- Sử dụng ngôn ngữ PL/SQL

Cú pháp

```
CREATE [OR REPLACE] TRIGGER [schema.]trigger
    {BEFORE | AFTER}
    {DELETE | INSERT | UPDATE [OF column [, column] ...]}
[OR {DELETE | INSERT | UPDATE [OF column [, column] ...]}] ...
ON [schema.]table
[ [REFERENCING { OLD [AS] old [NEW [AS] new]
                | NEW [AS] new [OLD [AS] old] } ]
  FOR EACH ROW
  [WHEN (condition)] ]
pl/sql_block
```

Hành động

■ Ngôn ngữ PL/SQL

- ngôn ngữ thủ tục
- SQL và ngôn ngữ lập trình (giới hạn)
- cho phép sử dụng cấu trúc điều khiển
IF-THEN-ELSE, LOOP
- cho phép khai báo biến
- sử dụng để định nghĩa triggers và thủ tục

Hành động

■ Khởi lệnh PL/SQL

DECLARE

 khai báo biến

BEGIN

 lệnh ...

END.

RUN;

Ví dụ

■ Toàn vẹn tham chiếu

- giữa Inscription và Student
- tạo trigger thực hiện kiểm chứng toàn vẹn tham chiếu khi chèn vào quan hệ Inscription

■ Sự kiện

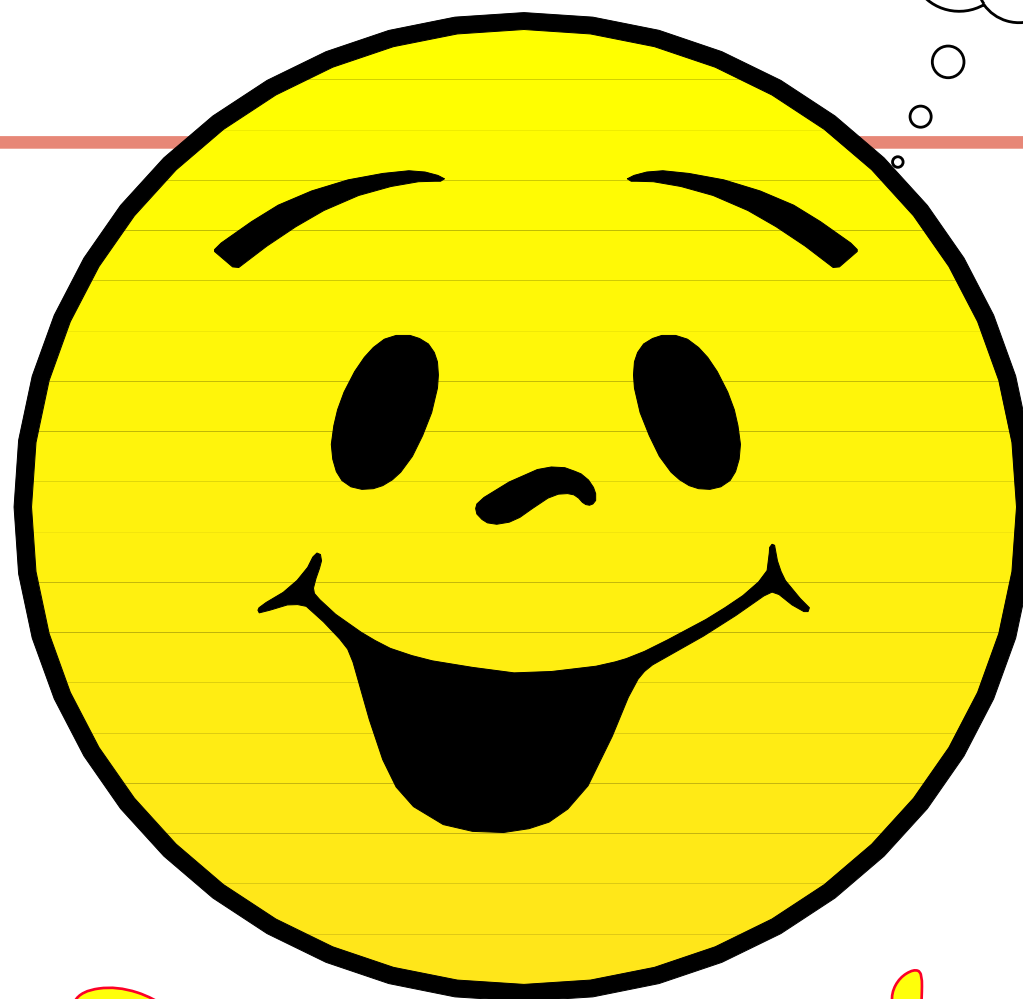
- chèn vào Inscription

■ Hành động

- kiểm tra xem bảng Student, tham khảo có tồn tại trong bảng Student

Ví dụ

```
CREATE TRIGGER verifstudent
  BEFORE INSERT ON inscription
  FOR EACH ROW
  DECLARE
    present integer;
  BEGIN
    select count(no) into present
    from student    where no=:new.no;
    if present < 0
    then    raise_application_error(-1, 'no student');
    end if;
  END.
RUN;
```



Cám ơn !