BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC ĐÀ NẴNG

TRẦN VĂN MINH

NGHIÊN CỨU XÂY DỰNG CÔNG CỤ KIỂM THỬ ĐỘT BIẾN CÁC CÂU LỆNH SQL

Chuyên ngành: KHOA HỌC MÁY TÍNH

Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng - Năm 2011

Công trình được hoàn thành tại

ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: TS. NGUYỄN THANH BÌNH

Phản biện 1: PGS.TS. LÊ VĂN SƠN

Phản biện 2: TS. NGUYỄN MẬU HÂN

Luận văn được bảo vệ trước Hội đồng chấm Luận văn tốt nghiệp thạc sĩ kỹ thuật họp tại Đại học Đà Nẵng vào ngày 16 tháng 10 năm 2011

Có thể tìm hiểu luận văn tại:

- Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng
- Trung tâm Học liệu, Đại học Đà Nẵng.

MỞ ĐẦU

1. Lý do chọn đề tài

Kiểm thử là một trong những hoạt động quan trọng trong tiến trình phát triển phần mềm. Nó góp phần vào việc đánh giá chất lượng sản phẩm phần mềm. Hiện nay, hầu hết các sản phẩm phần mềm ứng dụng đều có sử dụng ngôn ngữ truy vấn dữ liệu để lưu trữ quản lý thông tin, do đó việc quản lý và kiểm thử chúng trong quá trình phát triển phần mềm tốn nhiều thời gian.Vì vậy, nhu cầu có được một hệ thống ứng dụng kiểm thử đột biến để đánh giá chất lượng các bộ dữ liệu kiểm thử khi thực hiện kiểm thử các câu lệnh truy vấn SQL thực sự là cần thiết. Mặt khác, hiện nay vấn đề bản quyền phần mềm đang là vấn đề nan giải đối với các tổ chức, người dùng và nhu cầu sử dụng mã nguồn mở đang phát triển rộng khắp nhằm giúp giảm chi phí.

Xuất phát từ những phân tích và nhận xét trên, tôi chọn đề tài "***Nghiên cứu xây dựng công cụ kiểm thử đột biến các câu lệnh SQL***" dưới sự hướng dẫn của TS. Nguyễn Thanh Bình, sẽ giúp giảm thời gian và chi phí trong việc giám sát và kiểm thử sản phẩm phần mềm.

2. Mục đích ý nghĩa

Mục đích của đề tài là nghiên cứu và ứng dụng kỹ thuật kiểm thử đột biến vào việc đánh giá chất lượng bộ dữ liệu kiểm thử khi kiểm thử các câu lệnh truy vấn SQL, từ đó phát hiện các

lỗi còn tồn tại để các lập trình viên hoàn thiện hơn sản phẩm của mình.

Ý nghĩa khoa học: Hiểu và đánh giá các kỹ thuật kiểm thử đột biến và phương pháp kiểm thử đột biến câu lệnh truy vấn SQL. Kết quả có thể làm tài liệu tham khảo cho các kiểm thử viên hoặc các đơn vị phát triển phần mềm.

Ý nghĩa thực tiễn: Cung cấp một công cụ ứng dụng kỹ thuật kiểm thử đột biến vào việc kiểm thử cho các câu lệnh truy vấn SQL.

3. Nhiệm vụ mục tiêu

Đề tài tập trung nghiên cứu về kỹ thuật kiểm thử đột biến và cấu trúc đặc điểm của ngôn ngữ truy vấn dữ liệu SQL để nhận biết các toán tử đột biến, từ đó đề xuất giải pháp xây dựng công cụ hỗ trợ kiểm thử đột biến câu lệnh SQL và triển khai kiểm thử thực nghiệm trên các câu lệnh truy vấn SQL làm cơ sở để phân tích và đánh giá kết quả.

4. Đối tượng và phạm vi nghiên cứu

Đề tài tập trung nghiên cứu trên các đối tượng như sau:

- Kỹ thuật kiểm thử đột biến.
- Ngôn ngữ truy vấn có cấu trúc.
- Mã nguồn SQL Parser (GuduSoft.gsqlparser.dll).
- Kỹ thuật lập trình ngôn ngữ VS.Net.

Đề tài thuộc phạm vi nghiên cứu và ứng dụng.

5. Phương pháp nghiên cứu

- Thu thập và phân tích các tài liệu và thông tin liên quan đến đề tài.
- Thảo luận, lựa chọn hướng giải quyết vấn đề.
- Phân tích thiết kế hệ thống chương trình ứng dụng.
- Triển khai xây dựng chương trình ứng dụng.
- Kiểm tra, thử nghiệm, nhận xét và đánh giá kết quả.

6. Dự kiến kết quả đạt được

- Về mặt lý thuyết: Nắm được kiến thức về kỹ thuật kiểm thử đột biến và các toán tử đột biến câu lệnh truy vấn SQL.
- Về mặt thực tiễn: Xây dựng và đánh giá công cụ kiểm thử áp dụng kỹ thuật kiểm thử đột biến cho các câu lệnh truy vấn SQL.

7. Bố cục luận văn

Luận văn được chia thành 3 chương như sau:

Chương 1: Kiểm thử đột biến.

Chương 2: Kiểm thử đột biến các câu lệnh truy vấn SQL.

Chương 3: Xây dựng công cụ hỗ trợ kiểm thử đột biến các câu lệnh truy vấn SQL.

CHƯƠNG 1. KIỂM THỬ ĐỘT BIẾN

1.1. GIỚI THIỆU

Trong chương này, chúng tôi trình bày chi tiết lý thuyết về kiểm thử đột biến, phân tích các ưu và nhược điểm của phương pháp. Tiếp theo, chúng tôi trình bày các kỹ thuật kiểm thử đột biến khác nhau cũng như các ứng dụng phổ biến của kiểm thử đột biến trong thực tế.

1.2. LÝ THUYẾT KIỂM THỬ ĐỘT BIẾN

Trước khi trình bày lý thuyết kiểm thử đột biến, chúng ta bắt đầu bởi một ý tưởng đơn giản sau: để ước lượng số lượng cá trong một cái hồ, một cách để thực hiện việc đó là đánh dấu một số cá và thả vào hồ (giả sử, 80 con cá), sau đó đánh bắt một số cá và đếm số cá bị đánh dấu. Nếu chúng ta bắt được 50 con cá và 5 trong số đó bị đánh dấu, như vậy 1/10 số cá trong hồ bị đánh dấu, khi đó toàn bộ số cá trong hồ có thể thể được ước lượng là 800 con. Nếu chúng ta bắt được tất cả các cá bị đánh dấu, chúng ta có thể cho rằng toàn bộ cá trong hồ đã bị đánh bắt.

Kỹ thuật kiểm thử đột biến được xây dựng dựa trên ý tưởng này. Chúng ta đưa vào mã nguồn một số lỗi "bị đánh dấu", sau đó tìm cách xác định chúng. Nếu chúng ta xác định được tất cả các lỗi này, "lưới" của chúng ta có thể cũng đã bắt được nhiều các loại "cá" khác, đó chính là các lỗi chưa biết.

1.2.1. Khái niệm kiểm thử đột biến

Kiểm thử đột biến được thiết kế nhằm tạo ra dữ liệu kiểm thử có hiệu quả, nghĩa là phát hiện các lỗi của chương trình. Trong khi thực hiện kiểm thử đột biến, chúng ta tạo ra các phiên bản lỗi của chương trình gốc bằng cách chèn lỗi vào mã nguồn của nó. Sau đó, thực thi kiểm thử với lần lượt các dữ liệu kiểm thử cho từng phiên bản lỗi. So sánh kết quả đầu ra của từng phiên bản lỗi với chương trình gốc, từ đó đánh giá được khả năng phát hiện lỗi của các dữ liệu kiểm thử [3].

Các phiên bản lỗi được tạo ra từ chương trình gốc gọi là các *đột biến* (mutants). Kiểm thử đột biến là một kỹ thuật kiểm thử hộp trắng, hay còn gọi kỹ thuật kiểm thử cấu trúc.

1.2.2. Hai giả thuyết cơ bản

Kiểm thử đột biến được xây dựng dựa trên hai giả thuyết cơ bản. Giả thuyết "*lập trình viên giỏi*" (competent programmer hypothesis) và giả thuyết "*hiệu ứng liên kết*" (coupling effect hypothesis) [3]. Giả thuyết lập trình viên giỏi cho rằng thông thường các lập trình viên đều rất giỏi và họ sẽ không bao giờ viết ra các chương trình một cách tuỳ tiện, cẩu thả.

Giả thuyết hiệu ứng liên kết cho rằng các lỗi phức tạp được liên kết từ các lỗi đơn giản, như vậy bộ dữ liệu kiểm thử đủ khả năng phát hiện tất cả các lỗi đơn giản thì cũng có khả năng phát hiện các lỗi phức tạp với tỉ lệ cao.

1.2.3. Một số khái niệm cơ bản

Toán tử đột biến

Toán tử đột biến (mutation operator) hay còn được gọi luật đột biến (mutation rule) là một luật được áp dụng vào chương trình gốc để tạo ra các phiên bản đột biến. Nó có thể là việc thay thế một toán tử này bằng một toán tử khác; thay đổi toán hạng của biểu thức; xoá toàn bộ các biểu thức; thay đổi câu lệnh… hay có thể được tạo ra bằng cách thay đổi nhỏ về cú pháp của chương trình theo hướng mà các lập trình viên thường phạm phải.

1.2.4. Đột biến bị diệt và đột biến sống

Khi tiến hành thực thi kiểm thử lần lượt chương trình gốc *P* và đột biến *P'* của *P* với một dữ liệu thử *T*, sẽ có hai kịch bản khác nhau có thể xảy ra:

- Một là, hoặc lỗi được chèn vào trong chương trình đột biến *P'* được nhận biết, nghĩa là chương trình *P* và đột biến *P'* cho ra các kết quả khác nhau. Trong trường hợp này, đột biến *P'* được gọi là *bị diệt* (killed) bởi dữ liệu thử *T*. Khi đó, *T* được gọi là dữ liệu thử thích hợp vì nó có khả năng phát hiện được sự khác nhau giữa chương trình gốc *P* và đột biến *P'*.
- Hai là, chương trình gốc *P* và đột biến *P'* cho ra kết quả hoàn toàn giống nhau. Trong trường hợp này, có thể có hai khả năng xảy ra. Khả năng thứ nhất là dữ liệu thử *T* không đủ tốt (hay được gọi là dữ liệu thử không thích hợp), chúng ta sẽ phải tiến hành thực hiện kiểm thử lại với các dữ liệu thử tốt hơn. Khả năng thứ hai là chương trình

P và đột biến *P'* là những chương trình tương tự nhau, mọi dữ liệu thử đều không thể phân biệt sự khác nhau giữa chúng. Trong cả hai trường hợp này, đột biến *P'* được cho là *còn sống* (alive).

1.2.5. Đột biến tương đương

Các đột biến tương đương (equivalent mutant) là các đột biến cho ra kết quả giống với chương trình gốc với mọi dữ liệu thử hoặc cú pháp của đột biến và chương trình gốc khác nhau nhưng hoạt động tương tự nhau. Một cách hình thức, chúng ta nói: đột biến tương đương là đột biến còn sống mà mọi dữ liệu thử $T \subset D$ (*D*, tập các dữ liệu thử cho *P*) đều xác định được *P* và *P'* tương đương nhau ($P \equiv P'$).

1.2.6. Tỷ lệ đột biến

Tỷ lệ đột biến (Mutation Score), được ký hiệu *MS*, của chương trình *P* và dữ liệu thử *T* là tỷ lệ các đột biến không tương đương (so với chương trình gốc) bị diệt bởi dữ liệu thử *T*, được mô tả bởi công thức sau:

$$MS(P,T) = \frac{D}{N - E}$$

trong đó,

- *D*: số đột biến đã bị diệt,
- *N*: tổng số các đột biến,
- *E*: số đột biến tương đương.

Như vậy, $0 \le MS \le 1$ hay $0 \le MS\% \le 100$.

1.2.7. Chi phí của kiểm thử đột biến

Chi phí trong kiểm thử đột biến tập trung ba bước tốn kém nhất là sản sinh đột biến, biên dịch các đột biến và kiểm thử từng phiên bản đột biến với các dữ liệu kiểm thử. Như vậy, số lượng lớn các đột biến sẽ làm cho chi phí kiểm thử đột biến rất lớn.

1.3. TIẾN TRÌNH KIỂM THỬ ĐỘT BIẾN

Gọi chương trình gốc là *P*, các đột biến là *P'* và tập dữ liệu kiểm thử là *T*. Chúng ta có thể giải thích tiến trình thực hiện kiểm thử đột biến như sau:

Bước 1: Tạo đột biến *P'* từ chương trình gốc *P*.

Bước 2: Sinh các dữ liệu kiểm thử *T*.

Bước 3: Thực hiện chương trình gốc *P* với mỗi dữ liệu kiểm thử. Kiểm tra kết quả nhận được:

- Nếu đầu ra không đúng, phải chỉnh sửa chương trình gốc *P* và kiểm thử lại.
- Nếu đầu ra đúng, thực hiện bước tiếp theo.

Bước 4: Thực hiện từng đột biến còn sống với mỗi dữ liệu kiểm thử. So sánh kết quả thực hiện đột biến với kết quả thực hiện chương trình gốc đối với mỗi dữ liệu thử.

- Nếu tất cả các đột biến đều bị diệt. Hoàn thành kiểm thử.
- Nếu còn đột biến chưa bị diệt, chuyển sang bước tiếp theo.

Bước 5: Phân tích và xác định các đột biến tương đương.

Nếu còn các đột biến không tương đương nhưng chưa bị diệt thì các dữ liệu kiểm thử không đủ khả năng diệt đột biến. Phải hiệu chỉnh tập dữ liệu kiểm thử. Quay lại bước 1.

1.4. HẠN CHẾ CỦA KIỂM THỬ ĐỘT BIẾN

Lý thuyết và kết quả thực nghiệm đã cho thấy rằng, kiểm thử đột biến là phương pháp hiệu quả để đánh giá chất lượng của các bộ kiểm thử. Tuy nhiên, có một số hạn chế khi thực hiện kiểm thử đột biến như sau:

- Việc nhận dạng các đột biến tương đương là rất quan trọng nhưng rất khó khăn.
- Một số các đột biến không tương đương nhưng vẫn còn tồn tại trong quá trình kiểm thử.
- Chi phí tính toán trong kiểm thử đột biến rất cao, do số lượng toán tử đột biến thường rất lớn.
- Kiểm thử đột biến cũng tốn nhiều nhân công trong quá trình kiểm thử.

1.5. MỘT SỐ KỸ THUẬT NÂNG CAO HIỆU QUẢ KIỂM THỬ ĐỘT BIẾN

1.5.1. Giảm chi phí tính toán trong phân tích đột biến

Các kỹ thuật được nghiên cứu theo ba chiến lược: làm thông minh hơn, làm ít hơn và nhanh hơn. Chiến lược làm thông minh hơn, gồm các kỹ thuật: đột biến yếu (weak mutation), kiến trúc phân tán (distributed architectures). Chiến lược làm ít hơn hướng đến lựa chọn những đột biến sao cho hiệu quả nhất nhưng vẫn

đảm bảo chất lượng kiểm thử, gồm các kỹ thuật: đột biến lựa chọn (selective mutation), lấy mẫu đột biến (mutation sampling). Chiến lược làm nhanh hơn, hướng vào tự động hoá một số công đoạn và giảm tải ở các công đoạn chiếm nhiều chi phí tính toán, gồm các kỹ thuật: thực thi đột biến dựa vào giản đồ (schema-based), phương pháp tách rời biên dịch (separate compilation approach). Ngoài ra, còn nhiều kỹ thuật khác như kỹ thuật gom cụm đột biến (clustering of mutants).

1.5.2. Giảm bớt các công đoạn thủ công

Việc phát triển thủ công các dữ liệu kiểm thử đột biến một cách đầy đủ yêu cầu rất nhiều nỗ lực. Hơn nữa việc quyết định phiên bản đột biến nào tương đương với chương trình gốc là rất nhàm chán và hoạt động này thường dẫn đến nhiều sai sót. Việc tự động hóa các hoạt động này sẽ nâng cao hiệu quả và chất lượng của kiểm thử đột biến.

1.5.3. Cải tiến tiến trình kiểm thử đột biến

Tiến trình kiểm thử đột biến truyền thống còn tồn tại một số vấn đề, như thực thi lặp lại những ca kiểm thử, thực thi chương trình gốc, …

Tiến trình cải tiến được đề xuất nhằm khắc phục các hạn chế nêu trên. Trước hết, tự động tạo ra một tập các dữ liệu thử. Các dữ liệu thử đó sẽ được thực thi lần lượt với chương trình gốc và sau đó với các chương trình đột biến. Kiểm thử viên sẽ định nghĩa một giá trị ngưỡng, đó là giá trị nhỏ nhất có thể chấp nhận được của tỷ lệ đột biến. Nếu ngưỡng đó không đạt được, khi các dữ liệu thử không diệt được đột biến nào (giới hạn không hiệu

quả), sẽ bị loại trừ. Tiến trình này sẽ được lặp lại, với mỗi một thời điểm sản sinh các dữ liệu thử chỉ nhắm đến các đột biến còn sống, cho đến khi ngưỡng tỷ lệ đột biến đạt được.

1.6. ỨNG DỤNG CỦA KIỂM THỬ ĐỘT BIẾN

1.6.1. Đột biến mã nguồn

Đột biến mã nguồn chương trình đã được áp dụng cho cả hai mức kiểm thử đơn vị và kiểm thử tích hợp. Đối với kiểm thử mức đơn vị, đột biến được tạo ra để mô tả lỗi trong một đơn vị phần mềm mà lập trình viên thực hiện; trong khi đó đối với mức kiểm thử tích hợp, đột biến được tạo ra để mô tả lỗi tích hợp bởi lỗi kết nối hoặc tương tác giữa các đơn vị phần mềm. Đột biến này được áp dụng trên các ngôn ngữ lập trình như ngôn ngữ Fortran, Ada, C, Java, C#, AspectJ.

1.6.2. Đột biến đặc tả

Kiểm thử đột biến cũng đã được đề xuất áp dụng cho các đặc tả và thiết kế phần mềm. Kiểm thử đột biến áp dụng ở mức đặc tả và thiết kế thường được gọi là “đột biến đặc tả". Trong đột biến đặc tả, lỗi thường được phát sinh trong máy trạng thái hoặc các biểu thức lôgic để tạo ra các đột biến. Một đột biến bị diệt nếu điều kiện đầu ra là sai lệch. Kiểm thử đột biến thuộc loại này gồm đặc tả hình thức, môi trường thực thi, dịch vụ Web, hệ thống mạng.

1.7. TỔNG KẾT CHƯƠNG 1

CHƯƠNG 2. KIỂM THỬ ĐỘT BIẾN CÁC CÂU LỆNH TRUY VẤN SQL

2.1. GIỚI THIỆU

Trong chương này, trước hết chúng tôi trình bày sơ lược về cơ sở dữ liệu và ngôn ngữ truy vấn có cấu trúc. Một số công trình nghiên cứu về kiểm thử cơ sở dữ liệu cũng được đề cập. Đặc biệt, chúng tôi phân tích một số lỗi điển hình trong các câu lệnh truy vấn cơ sở dữ liệu. Trên cơ sở đó, chúng tôi trình bày ứng dụng kiểm thử đột biến cho các câu lệnh truy vấn.

2.2. CƠ SỞ DỮ LIỆU QUAN HỆ VÀ NGÔN NGỮ TRUY VẤN CÓ CẤU TRÚC

2.2.1. Cơ sở dữ liệu quan hệ

Cơ sở dữ liệu là một tập hợp có cấu trúc những dữ liệu có liên quan với nhau. Cơ sở dữ liệu được sử dụng phổ biến hiện nay là cơ sở dữ liệu quan hệ. Cơ sở dữ liệu quan hệ là các cơ sở dữ liệu dạng bảng có thể dễ dàng được tổ chức lại và được truy vấn.

2.2.2. Ngôn ngữ truy vấn có cấu trúc

Ngôn ngữ truy vấn có cấu trúc (SQL) là ngôn ngữ thường được sử dụng để định nghĩa lược đồ cơ sở dữ liệu và thực hiện việc cập nhật, xóa, chỉnh sửa và truy cập dữ liệu lưu trữ trong cơ sở dữ liệu.

Ngôn ngữ truy vấn có cấu trúc gồm các nhóm lệnh:

- Nhóm lệnh định nghĩa dữ liệu (Data Definition Language - DDL).
- Nhóm lệnh thao tác dữ liệu (Data Manipulation Language - DML).

Ngoài ra, ngôn ngữ truy vấn có cấu trúc còn có các lệnh dùng để quản lý quyền, các lệnh định nghĩa khung nhìn như CREATE VIEW, DROP VIEW; các lệnh điều khiển giao tác như COMMIT, ROLLBACK…

Lệnh truy vấn cơ bản SQL

Câu lệnh truy vấn cơ bản được mô tả bởi các mệnh đề SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY.

Các phép toán trong SQL

Trong SQL, có hai phép toán cơ bản: phép toán lô-gíc và phép toán quan hệ.

Các hàm kết hợp

Các hàm kết hợp như AVG, SUM, MAX, MIN, COUNT,…

2.3. KIỂM THỬ CƠ SỞ DỮ LIỆU

Phương pháp tạo ra các ca kiểm thử bằng cách xem xét các lược đồ cơ sở dữ liệu, các thuộc tính khác hoặc các ràng buộc quan hệ. Câu lệnh truy vấn SQL được chuyển thành dạng ngôn ngữ lập trình thủ tục, sau đó, kỹ thuật kiểm thử thông thường được áp dụng để kiểm thử và đánh giá câu lệnh SQL.

2.4. LỖI TRONG TRUY VẤN CƠ SỞ DỮ LIỆU

Các lỗi trong truy vấn SQL có thể chia làm hai loại: lỗi cú pháp và lỗi ngữ nghĩa. Các lỗi cú pháp nghĩa là chuỗi ký tự nhập vào không phải là truy vấn SQL hợp lệ. Các lỗi ngữ nghĩa là câu truy vấn SQL hợp lệ.

2.4.1. Mệnh đề SELECT

2.4.2. Mệnh đề FROM

2.4.3. Mệnh đề WHERE

2.4.4. Các hàm kết hợp

2.4.5. Mệnh đề GROUP BY

2.4.6. Mệnh đề ORDER BY

2.4.7. UNION/UNION ALL

2.4.8. Truy vấn vi phạm các mẫu chuẩn

2.4.9. Bộ trùng lặp

2.5. KIỂM THỬ ĐỘT BIẾN CÁC CÂU LỆNH TRUY VẤN

Ứng dụng kiểm thử đột biến cho các câu truy vấn cơ sở dữ liệu gồm các bước cơ bản:

- Xác định các lỗi thường phạm phải của lập trình viên khi viết của câu truy vấn.
- Xây dựng tập toán tử đột biến áp dụng cho câu truy vấn SQL.
- Xây dựng công cụ hỗ trợ thực hiện tiến trình kiểm thử đột biến các câu truy vấn.

- Thử nghiệm và đánh giá kỹ thuật kiểm thử đột biến trên các câu lệnh truy vấn trong các ứng dụng cơ sở dữ liệu cụ thể.

2.6. TOÁN TỬ ĐỘT BIẾN CHO CÁC CÂU LỆNH TRUY VẤN

Các toán tử đột biến được chia làm bốn nhóm:

- Toán tử đột biến các mệnh đề chính (SC - SQL Clause mutation operators);
- Toán tử đột biến cho các toán tử xuất hiện trong các điều kiện và biểu thức (OR - Operator Replacement mutation operators);
- Toán tử đột biến liên quan đến việc xử lý giá trị NULL (NL – NULL mutation operators);
- Toán tử đột biến thay thế các định danh: cột tham chiếu, hằng số và tham số (IR – Identifier Replacement mutation operators).

Mỗi nhóm đột biến được ký hiệu bởi các tên ngắn gọn chỉ gồm hai ký hiệu và mỗi toán tử đột biến trong các nhóm được ký hiệu bởi các tên ngắn gọn gồm ba ký hiệu.

2.6.1. Toán tử đột biến các mệnh đề chính (SC)

Mục đích của các toán tử đột biến các mệnh đề chính, được ký hiệu SC, là đột biến những tính năng khác nhau trong ngôn ngữ SQL tương tự như các ngôn ngữ khác (mệnh đề, hàm kết hợp, các câu truy vấn con…). Những toán tử SC góp phần phát hiện các lỗi như điều kiện kết nối không đúng, sử dụng không đúng từ khóa

DISTINCT, tính toán các hàm kết hợp không đúng hoặc không đúng thứ tự trong tập kết quả.

2.6.2. Toán tử đột biến cho các điều kiện và biểu thức (OR)

Các toán tử đột biến các điều kiện và các biểu thức được thiết kế nhằm phát hiện các lỗi lô-gíc trong các biểu thức trong các mệnh đề WHERE và HAVING.

2.6.3. Toán tử đột biến giá trị NULL (NL)

Trong ngôn ngữ truy vấn SQL, miền giá trị của mỗi thuộc tính được mở rộng thêm ký hiệu đặc biệt NULL để biểu thị cho mọi giá trị dữ liệu được hiểu là không được định nghĩa, hoặc không thích hợp, hoặc không xác định.

Việc xử lý các giá trị NULL không đúng có thể dẫn đến kết quả không lường trước được. Do đó, các đột biến có liên quan đến các giá trị NULL cần phải được xem xét để phát hiện các loại lỗi này.

2.6.4. Toán tử đột biến các định danh (IR)

Các toán tử đột biến IR thay thế các định danh cột, hằng và tham chiếu trong các tham số của truy vấn. Vì vậy, các toán tử đột biến này có khả năng phát hiện các lỗi như sử dụng không đúng các trường.

2.7. TỔNG KẾT CHƯƠNG 2

CHƯƠNG 3. XÂY DỰNG CÔNG CỤ KIỂM THỬ ĐỘT BIẾN CÁC CÂU LỆNH SQL

3.1. GIỚI THIỆU

Công cụ dùng để phân tích cấu trúc lệnh SQL, sinh các đột biến, thực thi đột biến, nhằm giúp đánh giá chất lượng của câu lệnh và các bộ dữ liệu kiểm thử.

Công cụ sử dụng bộ mã nguồn gsqlparser for .Net để phân tích cấu trúc lệnh SQL và sử dụng ngôn ngữ C# thực thi trong môi trường .NetFrameWork.

Công cụ tạo ra các đột biến bằng cách chèn lỗi vào câu lệnh gốc, thực thi lần lượt câu lệnh gốc và câu lệnh đột biến vào trên các bộ dữ liệu thử, từ đó đánh giá chất lượng của câu lệnh SQL và chất lượng của các bộ dữ liệu thử.

3.2. GIẢI PHÁP

Trên cơ sở phân tích các đột biến câu lệnh truy vấn cơ sở dữ liệu SQL ở chương 2, chúng tôi đề xuất thuật toán xây dựng công cụ kiểm thử đột biến cho câu lệnh SQL như sau.

Bước 1. Nhận câu lệnh SQL vào, kiểm tra, phân tích cú pháp câu lệnh và lưu dưới dạng cây cú pháp XML.

Bước 2. Thực hiện đột biến trên câu lệnh SQL tạo ra tập hợp các đột biến của câu lệnh SQL gốc.

Bước 3. Thực hiện vòng lặp gồm các bước sau:

a. Thực thi câu lệnh SQL đột biến.

b. So sánh kết quả thực thi câu lệnh đột biến và kết quả thực thi câu lệnh gốc.

c. Nếu kết quả khác nhau thì đột biến bị diệt, nếu ngược lại sẽ đánh dấu câu lệnh đột biến và xem xét tính tương đương câu câu lệnh đột biến với câu lệnh gốc.

d. *Bước 4*. Kết thúc quá trình, xuất ra báo cáo sơ bộ gồm các thông tin về số lượng đột biến sinh ra, số lượng đột biến bị diệt và tỷ lệ đột biến.

3.3. CÔNG NGHỆ

Công nghệ sử dụng:

- Hệ điều hành: Microsoft Windows Server, Windows XP, Windows 7, …
- Hệ quản trị cơ sở dữ liệu: MS SQL Server 2008
- Công nghệ .Net FrameWork 3.5
- Mã nguồn SQL Parser (*gudusoft.gsqlparser.dll*)

3.4. KIẾN TRÚC CÔNG CỤ

Cốt lõi của hệ thống này là bộ sản sinh đột biến, nghĩa là tạo ra các đột biến của câu lệnh truy vấn cơ sở dữ liệu. Để sinh các đột biến, bộ phân tích cú pháp phân tích câu lệnh truy vấn gốc và tạo ra cây cú pháp tương ứng. Từ cây cú pháp, bộ sinh đột biến thực hiện các toán tử đột biến để tạo ra các đột biến của câu lệnh truy vấn gốc. Sau đó, mỗi đột biến và câu lệnh truy vấn gốc sẽ được thực thi trên bộ dữ liệu thử để tính tỷ lệ đột biến.

3.5. BỘ PHÂN TÍCH CÚ PHÁP

Bộ phân tích chuyển câu lệnh truy vấn SQL gốc sang định dạng tài liệu XML bằng cách thay thế các từ khóa của câu lệnh bằng các phần tử XML và tổ chức lại tài liệu XML theo thứ tự phù hợp với cấu trúc của câu truy vấn SQL. Các từ khóa được biểu diễn bởi các phần tử và các tên cột, tên bảng, tham số và các hằng số được biểu diễn bởi thuộc tính text của các phần tử XML.

Câu lệnh SQL sau khi được bộ phân tích phân tích sẽ tạo ra cây cú pháp truy vấn ứng với các kiểu mệnh đề như sau: TSelectSqlStatement, TDeleteSqlStatement, TUpdateStatement,… tuỳ theo đầu vào của kịch bản SQL là mệnh Select, Delete, Update hay Insert,...

3.6. BỘ SINH ĐỘT BIẾN

Bộ sinh đột biến nhận đầu vào là lược đồ cơ sở dữ liệu và câu lệnh truy vấn SQL, tập các nhóm toán tử đột biến để phân tích câu lệnh truy vấn và chuyển các tài liệu XML của câu lệnh truy vấn vào mô hình DOM. Bộ sinh đột biến sẽ duyệt qua các phần tử trong mô hình DOM và khi bắt gặp một phần tử hoặc thuộc tính text của một nút thì thực hiện các bước sau:

- Xác định phạm vi
- Chọn cột
- Áp dụng toán tử đột biến

Trong khi bộ sinh đột biến sản sinh ra mỗi đột biến thì nó sẽ gọi thực thi bộ ghi đột biến để lưu lại đột biến đó. Các đột biến được lưu trong một tệp tin tài liệu XML.

3.7. BỘ THỰC THI ĐỘT BIẾN

Bộ sản sinh và thực thi những đột biến hoàn toàn tự động trong một công cụ đột biến SQL. Thu thập thông tin về lược đồ cơ sở dữ liệu, tải vào các cơ sở dữ liệu kiểm thử, điều khiển sự thay đổi dữ liệu và cài đặt tham số ghi vào tài liệu XML. Cú pháp câu truy vấn được phân tích ghi vào tài liệu XML và mỗi phần tử là một tiến trình. Mỗi tiến trình, cả truy vấn và lược đồ cơ sở dữ liệu được xem xét và đột biến được sản sinh bằng cách áp dụng các luật đã trình bày ở trên. Cuối cùng, các đột biến được thực thi.

3.8. THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

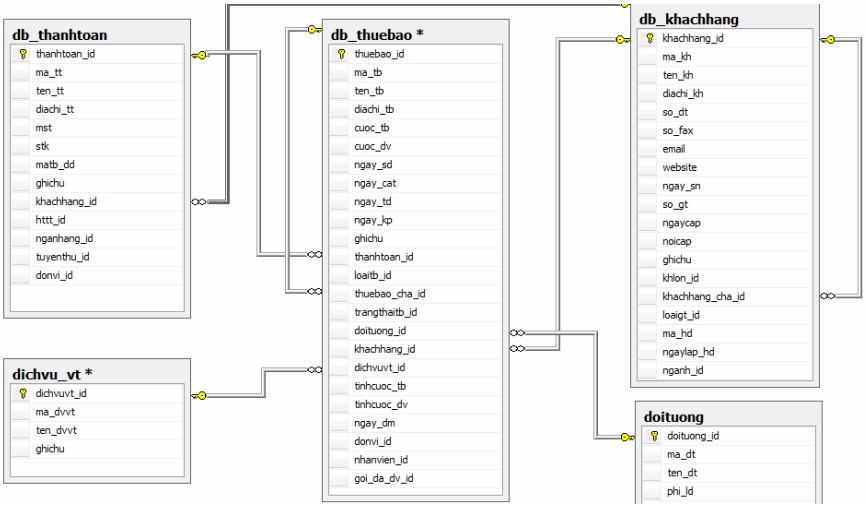
Trong phần thử nghiệm này, chúng tôi lựa chọn môi trường kiểm thử và bộ dữ liệu thử như sau:

Hệ quản trị cơ sở dữ liệu: MS Server SQL 2008

Bộ dữ liệu thử: Hệ cơ sở dữ liệu khách hàng sử dụng các dịch vụ viễn thông, gồm các thực thể như sau:

- **Danh bạ khách hàng**
- **Danh bạ thanh toán**
- **Danh bạ thuê bao**
- **Đối tượng thuê bao**
- **Dịch vụ viễn thông**

Lược đồ cơ sở dữ liệu



Hình 3.6. Lược đồ cơ sở dữ liệu

Toán tử đột biến

+ Toán tử đột biến các mệnh đề chính SC gồm: toán tử SEL, JOI, ARG,

+ Toán tử đột biến cho các điều kiện và biểu thức OR gồm các toán tử ROR, LCR, AOR, UOI, ABS, LKE,, BTW,

+ Toán tử đột biến các giá trị NULL gồm các toán tử NLI, NLO, NLF,

+ Toán tử đột biến định danh IRC.

3.9. TỔNG KẾT CHƯƠNG 3

KẾT LUẬN

Kiểm thử phần mềm là một trong những khâu quan trọng của quy trình xây dựng phần mềm nhằm kiểm tra xem phần mềm làm ra có những lỗi gì cần khắc phục. Kiểm thử không thể chứng minh được phần mềm là hết lỗi mà nó chỉ giúp cho người viết mã tìm ra và có biện pháp khắc phục càng nhiều lỗi càng tốt, góp phần đánh giá chất lượng sản phẩm phần mềm.

Kiểm thử đột biến được xem như là một trong những kỹ thuật kiểm thử hấp dẫn đầy hứa hẹn, giải quyết các vấn đề tốt hay không tốt của các bộ dữ liệu kiểm thử đạt yêu cầu hiệu năng cao và những công cụ hỗ trợ để tạo ra những đột biến và thực thi chúng. Tuy nhiên, kiểm thử đột biến còn khó áp dụng vào trong thực tế. Kiểm thử viên thường thấy khó khăn để áp dụng vào và thay vào đó là dựa vào những kiến thức cũng như kinh nghiệm khi thiết kế những ca kiểm thử.

1. Kết quả đạt được

Trong khuôn khổ một luận văn thạc sĩ, sau khi tiến hành tìm hiểu nghiên cứu về kiểm thử phần mềm, kỹ thuật kiểm thử đột biến và kiểm thử đột biến câu lệnh truy vấn, học viên đã đạt được một số kết quả nhất định như sau:

- *Về mặt nghiên cứu lý thuyết*:

Nắm được cơ bản về kỹ thuật kiểm thử đột biến, là một kỹ thuật nhằm đánh giá chất lượng của các bộ dữ liệu kiểm thử. Qua đó, cho thấy có rất nhiều công trình đã và đang nghiên cứu về

kiểm thử đột biến nhằm cải tiến và nâng cao hiệu quả của kiểm thử đột biến, cũng như mở rộng khả năng ứng dụng của kiểm thử đột biến. Điều đó thể hiện rằng kiểm thử đột biến là một trong những kỹ thuật kiểm thử đóng vai trò rất quan trọng và được ứng dụng rất rộng rãi, từ mức mã nguồn đến mức đặc tả.

- *Về mặt ứng dụng thực tiễn*:

Dựa trên những nghiên cứu về kỹ thuật kiểm thử đột biến, các giải pháp cải tiến và nâng cao hiệu quả của kiểm thử đột biến đồng thời phân tích các toán tử đột biến trên câu lệnh SQL, học viên đã đề ra giải pháp xây dựng công cụ hỗ trợ kiểm thử đột biến các câu lệnh SQL nhằm đánh giá chất lượng các bộ dữ liệu kiểm thử.

Công cụ này có thể được sử dụng để nghiên cứu kiểm thử các ứng dụng cơ sở dữ liệu và cho đánh giá tính đầy đủ của những ca kiểm thử, so sánh với các kỹ thuật khác nhau. Đồng thời kết quả nghiên cứu có thể cung cấp một giải pháp sơ bộ về việc ứng dụng kiểm thử đột biến để làm tài liệu tham khảo và áp dụng thực tế cho các đơn vị phát triển phần mềm đang cần nâng cao chất lượng kiểm thử sản phần phần mềm.

2. Hạn chế

Do thời gian tìm hiểu nghiên cứu có hạn nên đề tài chỉ mới tập trung vào việc phân tích các toán tử đột biến cơ sở chung của ngôn ngữ truy vấn SQL chưa đi sâu vào các toán tử riêng biệt của từng hệ thống cơ sở dữ liệu hiện nay như Oralce, MS Server SQL, MySQL, …

Công cụ phát triển độc lập chưa tích hợp vào những ứng dụng quản lý có sử dụng ngôn ngữ truy vấn dữ liệu cũng như dò tìm các lệnh truy vấn tự động để phân tích và sản sinh đột biến một cách tự động nhằm giúp cho kiểm thử viên kiểm thử mã nguồn một sản phẩm tiết kiệm thời gian và công sức.

3. Hướng phát triển

Mặc dù đã thực hiện các nội dung cơ bản về kỹ thuật kiểm thử đột biến, phân tích đột biến toán tử trên các câu lệnh SQL và xây dựng công cụ hỗ trợ kiểm thử đột biến câu lệnh SQL vận hành thành công. Tuy nhiên, để có thể hoàn thiện tốt hơn, đề tài cần nghiên cứu bổ sung thêm các nội dung sau:

- Áp dụng kỹ thuật cải tiến tiến trình kiểm thử đột biến, tự động tạo ra một tập các dữ liệu thử. Các dữ liệu thử đó sẽ được thực thi lần lượt với chương trình gốc và sau đó với các chương trình đột biến.

- Mở rộng áp dụng kiểm thử đột biến trên từng hệ thống cơ sở dữ liệu như Oracle, MS Server SQL, MySQL, …

- Tích hợp công cụ vào những ứng dụng quản lý sử dụng ngôn ngữ truy vấn để kiểm thử.