

HCMUS - Software Engineering

Date: Oct, 27. 2025

Lý Trọng Tín

Phan Thanh Tiến

Nguyễn Bùi Vương Tiến

Giang Đức Nhật



DATABASE TESTING

CONTENT

- 1 Giới thiệu về Kiểm thử Cơ sở dữ liệu ↗
- 2 Tổng quan về Database Testing ↗
- 3 Các loại Kiểm thử Cơ sở dữ liệu ↗
- 4 Quy trình Kiểm thử CSDL ↗
- 5 Kỹ thuật Kiểm thử CSDL ↗
- 6 Thách thức trong Kiểm thử CSDL ↗
- 7 Công cụ Kiểm thử CSDL ↗

Giới thiệu

Giới thiệu về Kiểm thử Cơ sở dữ liệu

- **Cơ sở dữ liệu (CSDL)**

- Là một tập hợp dữ liệu có cấu trúc được lưu trữ và quản lý bởi Hệ quản trị cơ sở dữ liệu (DBMS).
- Ví dụ: MySQL, PostgreSQL, SQL Server, Oracle.

- **Kiểm thử CSDL**

- Là quá trình xác thực và xác minh chất lượng, chức năng, hiệu suất và bảo mật của CSDL.
- Đảm bảo việc lưu trữ, truy xuất và quản lý dữ liệu hoạt động chính xác, hiệu quả và an toàn.

- **Vai trò trong quy trình phát triển:**

- Đảm bảo tính toàn vẹn và chính xác của dữ liệu.
- Ngăn chặn mất mát hoặc hỏng hóc dữ liệu.
- Tối ưu hóa hiệu suất và khả năng mở rộng của hệ thống.
- Tăng cường bảo mật bằng cách xác định các lỗ hổng.

Tổng quan

Mục đích của Database Testing

- **Database Testing** là quá trình xác thực và xác minh chất lượng, chức năng, hiệu suất và bảo mật của hệ thống CSDL.
- **So với UI Testing:**
 - UI Testing: tập trung vào những gì người dùng thấy (giao diện, bố cục, tương tác).
 - Database Testing: tập trung "bên dưới" giao diện, đảm bảo dữ liệu chính xác, toàn vẹn, an toàn.
- **Vai trò trong quy trình phát triển:**
 - Đảm bảo tính toàn vẹn và chính xác của dữ liệu.
 - Ngăn chặn mất mát hoặc hỏng hóc dữ liệu.
 - Tối ưu hiệu suất và khả năng mở rộng.
 - Tăng cường bảo mật qua việc phát hiện lỗ hổng.

Mục tiêu chính của Database Testing

- **Toàn vẹn & Ánh xạ Dữ liệu:**

- **Data Integrity:** Xác thực tính chính xác, nhất quán của dữ liệu qua các ràng buộc (khóa, check, not null).
- **Data Mapping:** Đảm bảo luồng dữ liệu từ UI → API → DB được ánh xạ chính xác, không sai lệch.

- **Tuân thủ Quy tắc Nghiệp vụ:**

- Xác minh logic trong **Stored Procedures, Triggers** và các hàm có hoạt động đúng theo yêu cầu nghiệp vụ hay không.

- **Đảm bảo thuộc tính ACID:**

- **Atomicity:** Một giao dịch phải được thực hiện hoàn toàn hoặc không gì cả (all or nothing).
- **Consistency:** Dữ liệu phải luôn ở trạng thái hợp lệ sau mỗi giao dịch.
- **Isolation:** Các giao dịch đồng thời không ảnh hưởng lẫn nhau.
- **Durability:** Dữ liệu đã commit phải được lưu trữ bền vững.

Các loại kiểm thử

Các loại Kiểm thử Cơ sở dữ liệu

- **Structural Testing:** Tập trung vào việc xác thực các thành phần cấu trúc của CSDL.
- **Functional Testing:** Kiểm tra các chức năng của CSDL từ góc độ người dùng cuối.
- **Non-functional Testing:** Đánh giá các khía cạnh như hiệu suất, bảo mật và khả năng sử dụng của CSDL.

Kiểm thử cấu trúc

Kiểm thử Cấu trúc (Structural)

- **Kiểm thử Schema & Consistency:**

- Xác thực **Schema** (bảng, cột, kiểu dữ liệu) khớp với đặc tả.
- Kiểm tra **Schema Consistency** giữa các môi trường (DEV, TEST, PROD).
- Đảm bảo không có đối tượng DB thừa/thiếu.

- **Kiểm thử Keys & Indexes:**

- **Khóa (Keys):** Xác thực khóa chính/ngoại (tính duy nhất, not-null, toàn vẹn tham chiếu).
- **Chỉ mục (Indexes):** Đảm bảo index được tạo đúng, sử dụng hiệu quả và không thừa.

- **Kiểm thử Stored Procedure:**

- Xác minh logic nghiệp vụ, xử lý lỗi.
- Kiểm tra kết quả trả về với các bộ dữ liệu đầu vào khác nhau (White-box).

- **Kiểm thử Trigger:**

- Đảm bảo trigger được kích hoạt đúng sự kiện (INSERT, UPDATE, DELETE).
- Xác thực logic thực thi có đúng và không gây side-effect.

Kiểm thử chức năng

Kiểm thử Chức năng (Functional)

- **Kiểm thử Hộp đen (Black Box):**

- Tập trung vào xác thực chức năng từ góc độ người dùng cuối, không cần biết logic nội tại.
- **Kiểm thử hoạt động CRUD:** Xác minh các thao tác `CREATE`, `READ`, `UPDATE`, `DELETE` từ giao diện/API được phản ánh chính xác trong CSDL.
- **Kiểm tra giá trị mặc định/ràng buộc:** Đảm bảo dữ liệu nhập vào tuân thủ đúng quy tắc.

- **Kiểm thử Hộp trắng (White Box):**

- Tập trung vào xác thực logic bên trong các đối tượng CSDL.
- **Xác thực logic nghiệp vụ:** Kiểm tra các trigger, stored procedure, và view đảm bảo chúng thực thi đúng yêu cầu.
- **Kiểm tra toàn vẹn dữ liệu:** Đảm bảo các ràng buộc (constraints) và quan hệ (relationships) hoạt động chính xác.

Kiểm thử phi chức năng

Kiểm thử Phi chức năng (Non-functional)

- **Kiểm thử Hiệu năng (Performance):**

- **Load Testing:** Đánh giá hiệu suất CSDL dưới tải trọng dự kiến, đo thời gian phản hồi truy vấn.
- **Stress Testing:** Xác định điểm gãy của CSDL bằng cách áp dụng tải trọng cực lớn.

- **Kiểm thử Bảo mật (Security):**

- Ngăn chặn các lỗ hổng phổ biến như **SQL Injection**.
- Xác thực quyền truy cập, đảm bảo người dùng chỉ thấy dữ liệu được phép.

- **Kiểm thử Phục hồi (Recovery):**

- Xác minh CSDL có thể được phục hồi thành công từ các bản sao lưu (backups) sau sự cố.

- **Kiểm thử Tương thích (Compatibility):**

- Đảm bảo CSDL hoạt động ổn định trên các phiên bản DBMS, hệ điều hành, và nền tảng khác nhau.

💡 Bảo mật CSDL cần ưu tiên: đảm bảo kiểm soát truy cập chặt chẽ, phòng chống SQL Injection, và mã hóa dữ liệu nhạy cảm ở trạng thái nghỉ và khi truyền tải.

Quy trình kiểm thử

Quy trình Kiểm thử CSDL

1. Phân tích và Thiết kế:

- Xác định yêu cầu nghiệp vụ, phạm vi kiểm thử.
- Thiết kế test case, kịch bản và bộ dữ liệu.

2. Chuẩn bị Môi trường & Dữ liệu:

- Thiết lập máy chủ thử nghiệm riêng biệt.
- Tạo hoặc sao chép bộ dữ liệu thử nghiệm (Test Data).

3. Thực thi Kiểm thử:

- Chạy các kịch bản kiểm thử (thủ công hoặc tự động).
- Ghi lại kết quả và các lỗi phát sinh.

4. Báo cáo và Theo dõi:

- Phân tích kết quả, so sánh với kết quả mong đợi.
- Ghi nhận lỗi (log bug) và báo cáo cho đội phát triển.

5. Kiểm thử Hồi quy và Hoàn tất:

- Thực hiện lại các bài kiểm thử sau khi lỗi được sửa.
- Hoàn tất báo cáo và ký duyệt (sign-off).

Kỹ thuật kiểm thử

Kỹ thuật Kiểm thử SQL Queries

- **Mục tiêu:**

- Đảm bảo các câu lệnh SQL trả về kết quả chính xác.
- Tối ưu hóa hiệu suất truy vấn.

- **Cách thực hiện:**

- **Kiểm tra tính đúng đắn:** So sánh kết quả của truy vấn với dữ liệu mong đợi.
- **Kiểm tra hiệu suất:** Phân tích **Execution Plan** để xác định các truy vấn chậm, thiếu index hoặc có full table scan.
- **Kiểm tra với dữ liệu lớn:** Đánh giá thời gian phản hồi khi CSDL có hàng triệu bản ghi.

Kỹ thuật Data-Driven

- **Phân vùng Tương đương (EP):**

- Chia dữ liệu đầu vào thành các nhóm (lớp) mà hệ thống xử lý tương tự nhau.
- **Ví dụ:** Với trường **tuổi**, các lớp có thể là:
 - **Âm** (không hợp lệ)
 - **0-17** (trẻ em)
 - **18-60** (người lớn)
 - **> 60** (người cao tuổi)
- Chỉ cần chọn một giá trị đại diện trong mỗi lớp để kiểm thử.

- **Phân tích Giá trị Biên (BVA):**

- Tập trung kiểm thử tại các giá trị biên của mỗi phân vùng.
- **Ví dụ:** Với lớp **18-60**, các giá trị biên cần kiểm thử là:
 - **17** (ngay dưới)
 - **18** (biên dưới)
 - **19** (ngay trên)
 - **59** (ngay dưới)
 - **60** (biên trên)
 - **61** (ngay trên)
- Giúp phát hiện lỗi logic tại các điểm chuyển tiếp.

Thách thức

Thách thức trong Kiểm thử CSDL

- **Dữ liệu lớn và phức tạp:** Việc kiểm thử với khối lượng dữ liệu lớn có thể rất khó khăn và tốn thời gian.
- **Kiến thức về SQL:** Người kiểm thử cần có hiểu biết tốt về SQL và các khái niệm CSDL.
- **Chi phí và Thời gian:** Kiểm thử CSDL có thể tốn kém và mất nhiều thời gian, đặc biệt với các hệ thống lớn.
- **Quản lý dữ liệu thử nghiệm:** Tạo và quản lý dữ liệu thử nghiệm phù hợp là rất quan trọng nhưng cũng đầy thách thức.
- **Cô lập môi trường thử nghiệm:** Đảm bảo môi trường thử nghiệm được tách biệt hoàn toàn với môi trường sản phẩm.

Công cụ kiểm thử

Phân loại Công cụ Kiểm thử CSDL

Để chọn đúng công cụ, chúng ta cần xem xét chúng qua các khía cạnh sau:

- **Phạm vi và Loại hình Kiểm thử:** Công cụ dùng để làm gì? (Hiệu năng, chức năng, bảo mật...).
- **Phương pháp tiếp cận:** Kiểm thử hộp đen hay hộp trắng? Mức độ cô lập ra sao?
- **Tự động hóa và Tích hợp:** Khả năng tích hợp vào CI/CD và quản lý dữ liệu.
- **Tính tương thích:** Hỗ trợ những hệ quản trị CSDL nào?
- **Thiết kế và Chức năng:** Giao diện, tính dễ sử dụng.

I. Phạm vi & Loại hình Kiểm thử (1/2)

- **Kiểm thử Hiệu năng (Performance):**

- Mô phỏng tải lớn, đo lường thời gian phản hồi và tìm điểm gãy.
- **Công cụ:** JMeter, HammerDB, Swingbench.

- **Kiểm thử Chức năng/Unit Test:**

- Xác minh logic nghiệp vụ và các thao tác CRUD.
- **Công cụ:** tSQLt (Unit Test), DbFit (Acceptance Test).

- **Kiểm thử Cấu trúc (Structural):**

- So sánh và xác thực schema, bảng, cột, khóa...
- **Công cụ:** Redgate SQL Compare.

- **Kiểm thử Dữ liệu và ETL:**

- So sánh dữ liệu giữa nguồn và đích, đảm bảo tính toàn vẹn.
- **Công cụ:** QuerySurge.

I. Phạm vi & Loại hình Kiểm thử (2/2)

- **Kiểm thử Bảo mật (Security):**

- Phát hiện các lỗ hổng bảo mật, đặc biệt là SQL Injection.
- **Công cụ:** SQLMap là công cụ tấn công chuyên dụng để tự động phát hiện và khai thác lỗ hổng.

II. Phương pháp & Độ cô lập

- **Phương pháp tiếp cận:**

- **Hộp trắng (White Box):** Can thiệp và kiểm tra logic bên trong CSDL.
 - Ví dụ: tSQLt cho phép viết test bằng T-SQL.
- **Hộp đen (Black Box):** Kiểm tra chức năng từ góc độ người dùng (input/output).
 - Ví dụ: DbFit dùng bảng để mô tả logic.

- **Tính Cô lập (Isolation):**

- Đảm bảo mỗi test case chạy độc lập, không ảnh hưởng đến CSDL hay các test khác.
- **Ví dụ:**
 - tSQLt tự động ROLLBACK sau mỗi test.
 - DBUnit dùng CLEAN_INSERT để reset dữ liệu.

III. Tự động hóa & Tích hợp

- **Tích hợp CI/CD:**

- Khả năng chạy qua dòng lệnh (CLI) và xuất báo cáo (JUnit XML) để tích hợp vào Jenkins, GitLab CI.

- **Quản lý Dữ liệu Test:**

- Hỗ trợ thiết lập dữ liệu mẫu (dataset) trước khi test và dọn dẹp sau đó.
- **Công cụ:** DBUnit, NoSQLUnit sử dụng file XML/JSON để quản lý.

- **Khả năng Mô phỏng (Mocking):**

- Thay thế các đối tượng phụ thuộc (bảng, procedure) bằng đối tượng giả để cô lập test.
- **Công cụ:** tSQLt cung cấp FakeTable và SpyProcedure.

IV. Tương thích & V. Thiết kế

- **Hỗ trợ Hệ quản trị CSDL (DBMS):**

- **Chuyên dụng:** tSQLt (SQL Server), Swingbench (Oracle).
- **Đa nền tảng:** JMeter (qua JDBC), NoSQLUnit (MongoDB, Cassandra...).


- **Giao diện và Tính dễ đọc:**

- Kịch bản test có dễ đọc, dễ hiểu không?
- **Ví dụ:** DbFit dùng bảng, phù hợp cho cả BA, QA.

- **Hỗ trợ Kỹ thuật Thiết kế Test Case:**

- Hỗ trợ các kỹ thuật như **Data-Driven Testing (DDT)**.
- **Ví dụ:** JMeter có thể đọc dữ liệu từ file CSV để thực hiện nhiều kịch bản test.

Các phương pháp tốt nhất (Best Practices)

 Việc chọn công cụ kiểm thử CSDL giống như chọn dụng cụ cho thợ mộc: không thể dùng búa tạ (**JMeter** - stress test) cho việc chạm khắc tinh xảo (**tSQLt** - unit test). Hãy chọn công cụ phù hợp nhất với nhiệm vụ.

Công cụ Structural

- **Redgate SQL Compare**

- **Là gì:** Công cụ thương mại mạnh mẽ để so sánh schema CSDL.
- **Chức năng:** Tìm và đồng bộ hóa sự khác biệt về schema (bảng, cột, view...) giữa các môi trường (VD: DEV vs. PROD).
- **Chi phí:** Trả phí (Thương mại).

- **ApexSQL Diff**

- **Là gì:** Công cụ thương mại tương tự Redgate.
- **Chức năng:** So sánh, phát hiện sự khác biệt trong cả schema và dữ liệu.
- **Chi phí:** Trả phí (Thương mại).

Công cụ Performance & Security

- **Apache JMeter**

- **Là gì:** Công cụ mã nguồn mở chuyên về kiểm thử tải và hiệu năng.
- **Chức năng:** Tạo ra lượng lớn truy vấn đồng thời (qua JDBC) để đo thời gian phản hồi và khả năng chịu tải của CSDL.
- **Chi phí:** Miễn phí.

- **HammerDB**

- **Là gì:** Công cụ benchmark và load testing mã nguồn mở.
- **Chức năng:** Đo lường hiệu suất của nhiều loại CSDL (Oracle, SQL Server, PostgreSQL...) dưới các kịch bản tải khác nhau.
- **Chi phí:** Miễn phí.

- **SQLMap**

- **Là gì:** Công cụ mã nguồn mở chuyên dò tìm và khai thác lỗ hổng SQL Injection.
- **Chức năng:** Tự động hóa việc phát hiện các điểm yếu bảo mật liên quan đến SQL Injection trong ứng dụng.
- **Chi phí:** Miễn phí.

Công cụ Unit/Functional & Structural

- **Kiểm thử Unit/Functional:**

- **tSQLt:** Framework mã nguồn mở cho SQL Server để thực hiện Unit Test.
- **DbFit:** Hỗ trợ kiểm thử CSDL qua các bảng quyết định (decision tables), dễ dàng cho cả tester và BA.
- **DbUnit:** Mở rộng của JUnit, chuyên dùng để quản lý trạng thái CSDL giữa các lần chạy test.

- **Kiểm thử Cấu trúc (Structural):**

- **Redgate SQL Compare:** So sánh và đồng bộ hóa schema giữa các CSDL, lý tưởng để kiểm tra tính nhất quán.
- **ApexSQL Diff:** Công cụ mạnh mẽ để so sánh, phát hiện sự khác biệt trong schema và dữ liệu.

Công cụ Performance & Security

- **Kiểm thử Hiệu năng (Performance):**

- **Apache JMeter:** Dù không chuyên về CSDL, JMeter có thể tạo tải (load) lên các API và đo lường thời gian phản hồi từ CSDL.
- **HammerDB:** Công cụ benchmark và load testing mã nguồn mở, hỗ trợ nhiều loại CSDL (Oracle, SQL Server, PostgreSQL).

- **Kiểm thử Bảo mật (Security):**

- **SQLMap:** Công cụ mã nguồn mở tự động phát hiện và khai thác các lỗ hổng SQL Injection, giúp xác định điểm yếu bảo mật.

Các phương pháp tốt nhất (Best Practices)

- **Sử dụng dữ liệu thực tế:**
 - Kiểm thử với dữ liệu gần giống với môi trường sản phẩm nhất có thể.
- **Tự động hóa khi có thể:**
 - Tự động hóa các bài kiểm thử lặp đi lặp lại để tiết kiệm thời gian và giảm lỗi do con người.
- **Cô lập môi trường thử nghiệm:**
 - Giữ môi trường thử nghiệm tách biệt khỏi môi trường sản phẩm để tránh xung đột.
- **Kết hợp với kiểm thử giao diện:**
 - Kiểm thử CSDL cùng với giao diện người dùng của ứng dụng.
- **Theo dõi hiệu suất:**
 - Liên tục theo dõi hiệu suất CSDL để xác định và giải quyết các tắc nghẽn.

Kết luận

- **Kiểm thử CSDL là nền tảng** để đảm bảo độ tin cậy, tính toàn vẹn và hiệu suất của ứng dụng.
- **Tự động hóa và AI đang định hình lại tương lai** của kiểm thử CSDL, giúp tạo test case thông minh, phát hiện sớm các dị thường và tối ưu hóa hiệu suất.
- Việc kết hợp các kỹ thuật truyền thống với công cụ hiện đại và phương pháp tiếp cận dựa trên AI sẽ giúp xây dựng các hệ thống CSDL không chỉ mạnh mẽ mà còn linh hoạt và an toàn trong môi trường phát triển nhanh.

Tài liệu tham khảo

- Divesh Mehta — Database testing: What is it? Why & best practices [↗](#) [truy cập: Oct. 22, 2025]
- Indaacademy — Database Testing Tutorial – Hướng dẫn kiểm thử Cơ sở dữ liệu (P2) [↗](#) [truy cập: Oct. 22, 2025]
- Thomas Hamilton — Database Testing Tutorial (Guru99) [↗](#) [truy cập: Oct. 22, 2025]
- Gunashree RS — Database Tests: Guide to Ensuring Data Integrity and Performance [↗](#) [truy cập: Oct. 22, 2025]
- David Eketé — Advanced Test Data Management: Techniques and Best Practices [↗](#) [truy cập: Oct. 23, 2025]
- HammerDB Documentation [↗](#) [truy cập: Oct. 23, 2025]

QnA