

Software Testing

Testing Database Application

Nhóm thực hiện : 4 - DevSoft

Trình bày : Trương Toàn Thịnh



Trường Đại học Khoa học Tự nhiên
Khoa Công nghệ Thông tin

- Overview
- Testing DB application
- Kỹ thuật kiểm chứng
- Công cụ kiểm chứng
- Q&A

- Giới thiệu về DB Application
- Các khái niệm cơ bản

- Ứng dụng CSDL đóng vai trò quan trọng
- Ứng dụng ngày càng phức tạp
- Dữ liệu lớn

- Giới thiệu về DB Application
- Các khái niệm cơ bản

- **Model**
- **Table**
- **Column**
- **Row**
- **Attribute**
- **Entities**
- **Primary Key**
- **Null Values**
- **Constraint**

- Relationship types

- ❖ one - one
- ❖ one - Many
- ❖ Many - Many

Tester phải có kiến thức về CSDL nếu muốn thực hiện Test DB Application

- **Overview**
- **Testing DB application**
- **Kỹ thuật kiểm chứng**
- **Công cụ kiểm chứng**
- **Q&A**

- **Testing DB là một thách thức**
- **Một vài khía cạnh kiểm chứng**
 - Ứng dụng hoạt động chính xác ?
 - Database ánh xạ chính xác lên thế giới thực qua mô hình ?
 - Dữ liệu có đúng đắn ?
 - Bảo mật ?
 - ...

Test những gì với DB application ?

- **Integrity Testing**
- **Control Testing**
- **Transaction Concurrency Testing**

Test những gì với DB application ?

- **Integrity Testing**
- **Control Testing**
- **Transaction Concurrency Testing**

- **Test tất cả các ràng buộc toàn vẹn trên CSDL**
 - ➔ **Entity Integrity**
 - ➔ **Primary Key Integrity**
 - ➔ **Column Key Integrity**
 - ➔ **Domain Integrity**
 - ➔ **User Defined Integrity**
 - ➔ **Referential Integrity**

- **Referential Integrity**

- Toàn vẹn tham chiếu
- Kiểm chứng các mối quan hệ
 - 1 - 1
 - 1 - N
 - N - N
- Example :
 - LOP (MALOP, TENLOP, SOHOCSINH)
 - HOCSINH(MAHS, TENHOCSINH, MALOP)

- Referential Integrity**

| Test case | Expected Result |
|---|--|
| 1. Thêm một lớp | Không liên quan tới thực thể học sinh |
| 2. Thêm một học sinh | Phải tồn tại một khóa ngoại tới thực thể lớp |
| 3. Xóa một lớp (lớp có quan hệ với thực thể học sinh) | Không được xóa. Nên thông báo lớp còn học sinh hay không |
| 4. Cập nhật một học sinh | Phải tồn tại một khóa ngoại tới thực thể lớp |

Test những gì với DB application ?

- Integrity Testing
- Control Testing
- Transaction ConcurrencyTesting
- ...

- **Security Testing**

- Kiểm chứng khả năng bảo vệ từ các truy cập bất hợp lệ

- **Backup Testing**

- Kiểm tra khả năng sao lưu dữ liệu

- **Recovery Testing**

- Kiểm tra khả năng phục hồi dữ liệu

Test những gì với DB application ?

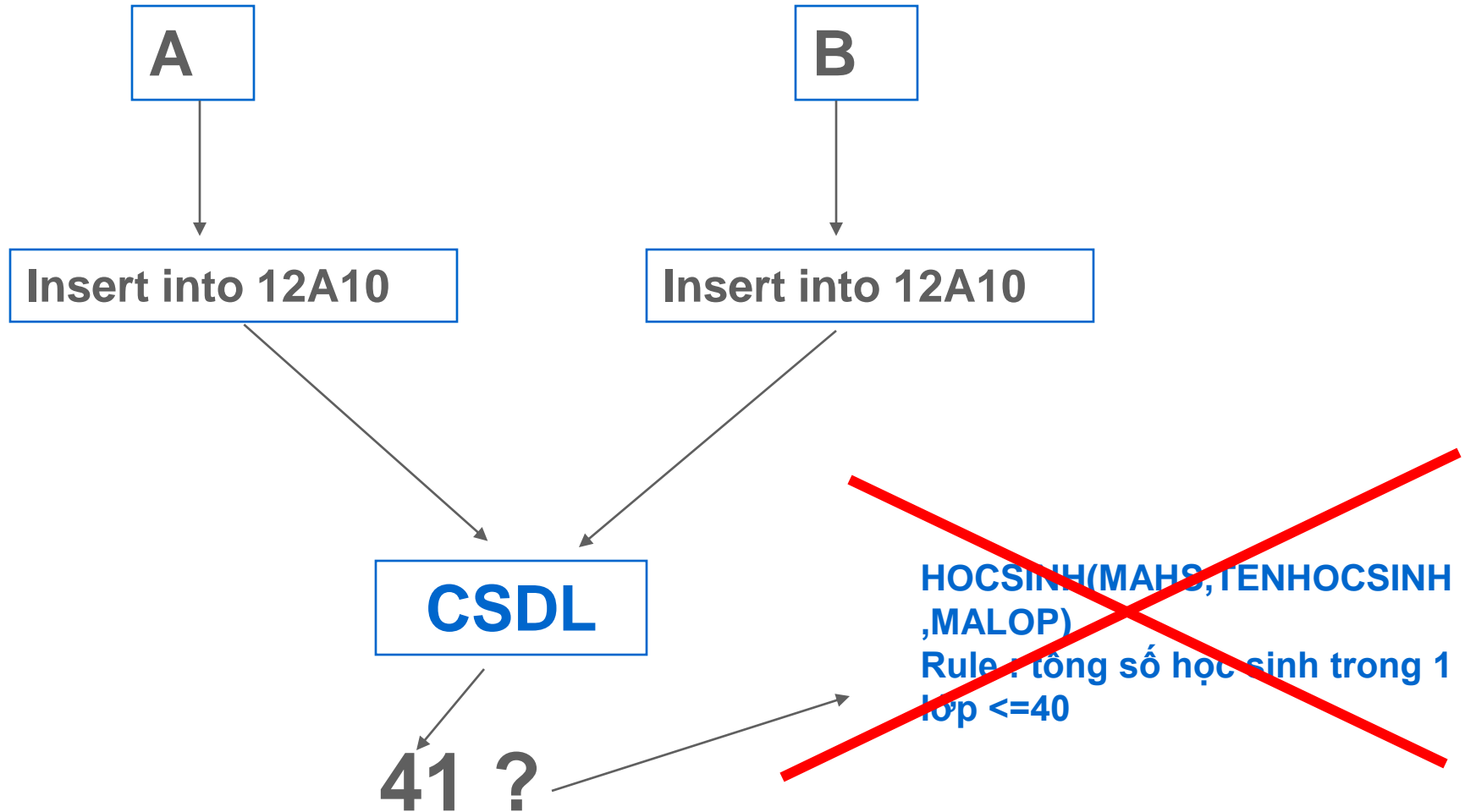
- **Integrity Testing**
- **Control Testing**
- **Transaction Concurrency Testing**

Transaction Concurrency Testing

- Kiểm chứng các thao tác trên dữ liệu vào cùng một thời điểm (truy xuất đồng thời)
- Khả năng xuất hiện nhiều lỗi nghiêm trọng
- Example :
 - LOP (MALOP, TENLOP, SOHOCSINH)
 - HOCSINH(MAHS, TENHOCSINH, MALOP)
 - Rule : tổng số học sinh trong 1 lớp ≤ 40

Transaction Concurrency Testing

- Example** : tổng số học sinh 12A10 là 39



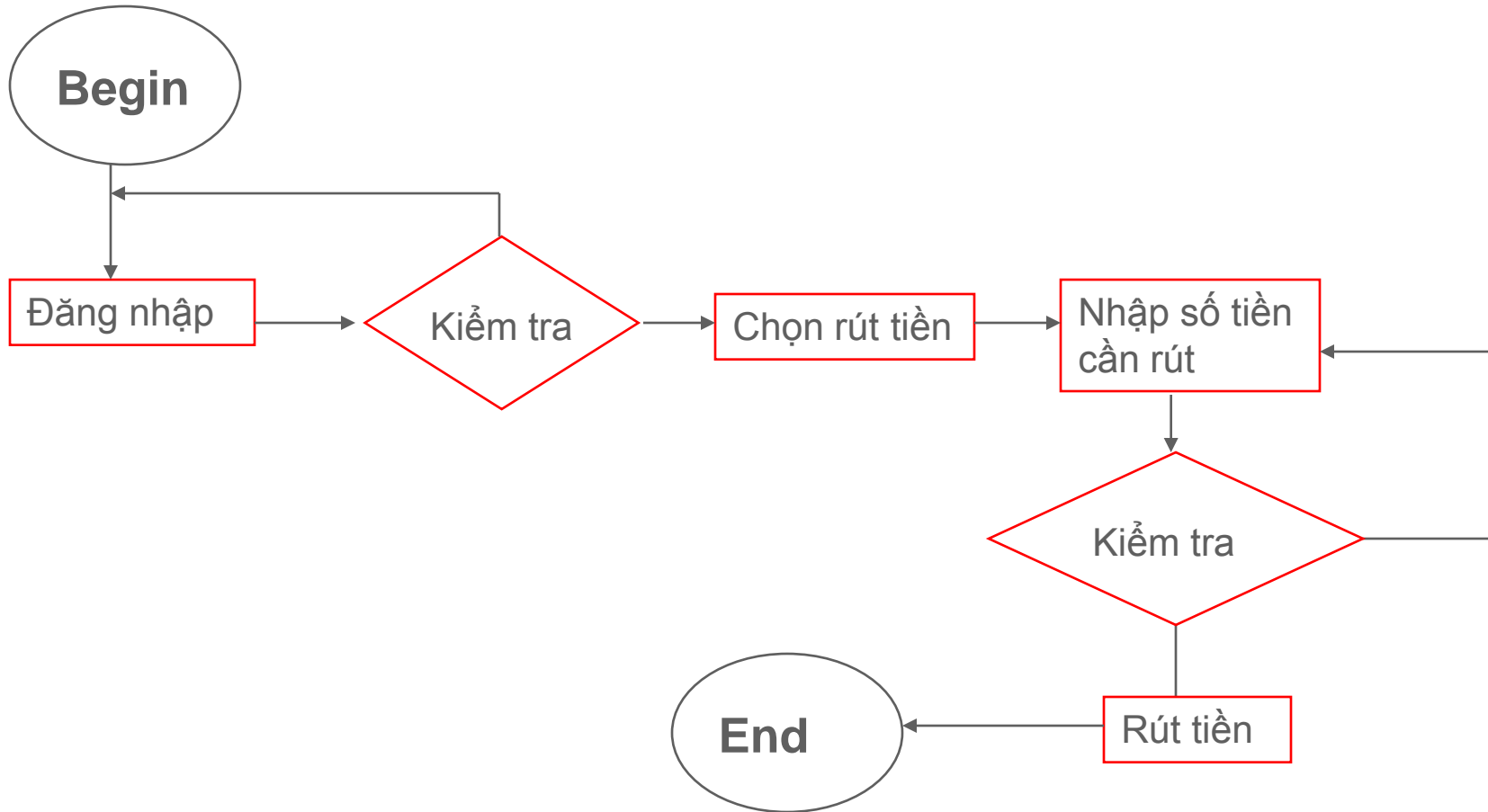
- **Overview**
- **Testing DB application**
- **Kỹ thuật kiểm chứng**
- **Công cụ kiểm chứng**
- **Q&A**

- **Transaction Flow Testing**
- **Data Flow Testing**

- **Black Box Testing**
- **Giúp tìm được những lỗi nghiêm trọng trong hệ thống**
- **Là nền tảng dùng để test các chức năng của hệ thống**

- **Transaction ?**
 - A Unit of work (từ quan điểm của người sử dụng)
 - Gồm một dãy các thao tác khác nhau
- **Example : Chức năng rút tiền bằng thẻ ATM**
 - Đăng nhập
 - Chọn chức năng rút tiền
 - Nhập số tiền cần rút
 - Kiểm Tra dữ liệu hợp lệ hay không ?
 - Rút tiền

- Transaction - Flow Graph ?



- **Transaction - Flow Testing Steps**
 - Xác định Transaction Flow và vẽ Transaction Flow Graph
 - Xác định tất các đường đi trong Graph (tương tự như phủ nhánh và phủ điều kiện)
 - Thiết kế Test Case

- **Transaction - Flow Testing Steps**
 - Xác định Transaction Flow và vẽ Transaction Flow Graph
 - Dựa trên yêu cầu của người dùng
 - Không nên xác định dựa trên thiết kế

- **Transaction Flow Testing**
- **Data Flow Testing**

- White Box Testing
- Phân tích lỗi dựa trên trạng thái của dữ liệu thông qua các hoạt động Read, Write
- **Test transaction Concurrency**

- Example:**

Function ThemHocSinh(hocsinh, malop)

{

int tongHS = (Select count(*) From HOCSINH where MALOP =malop) ;

if (tongHS <40)

{

//Thêm học sinh mới vào bảng Học Sinh

}

}

T1

Read

Write T2

- Thực hiện trên một thể hiện (instance) của ứng dụng : không có vấn đề.
- Hai thể hiện của ứng dụng cùng chạy : ?

Instance A

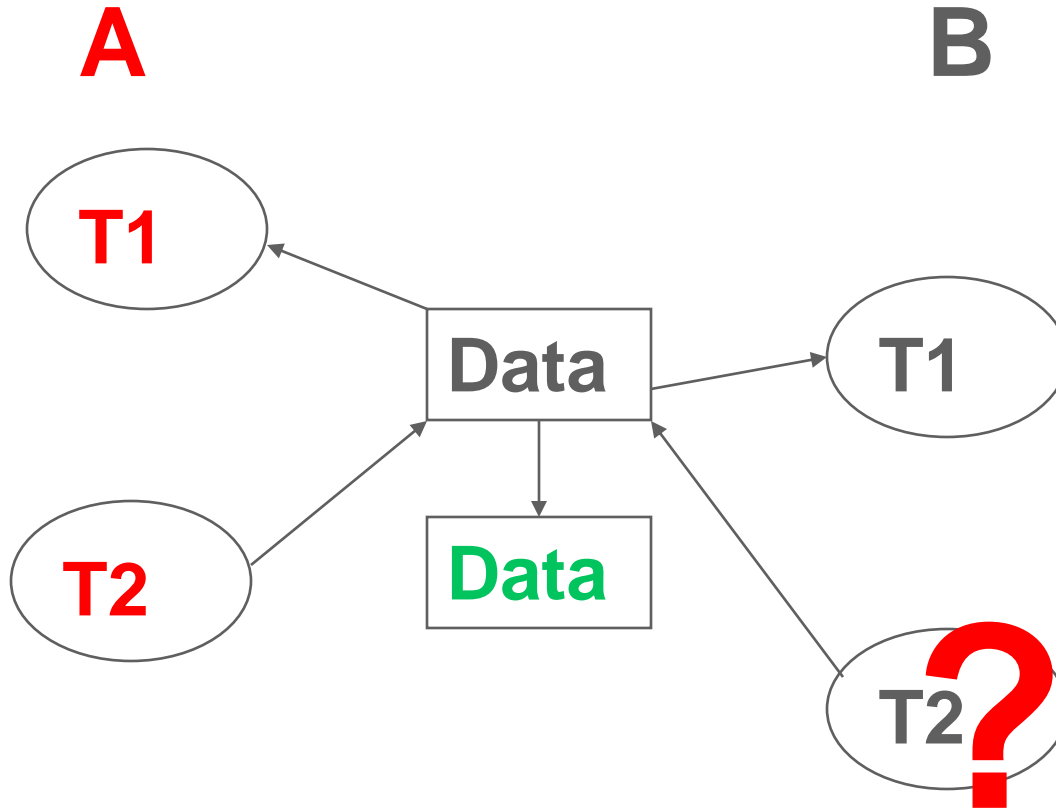
```
Function ThemHocSinh(hocsinh,
malop)
{
  int tongHS = (Select count(*) From
HOCSINH where MALOP =malop) ;
  if ( tongHS <40)
  {
    //Thêm học sinh mới
    vào bảng Học Sinh
  }
}
```

Instance B

```
Function ThemHocSinh(hocsinh, malop)
{
  int tongHS = (Select count(*) From
HOCSINH where MALOP =malop) ;
  if ( tongHS <40)
  {
    //Thêm học sinh mới
    vào bảng Học Sinh
  }
}
```

•Schedule : T1(A) → T1(B) → T2(A) → T2(B)

Error ?



- $T(i), T(j)$: các thao tác đọc ghi trên dữ liệu của một thể hiện A
- $T(k)$ là một thao tác nằm giữa $T(i)$ và $T(j)$ trên cùng một dữ liệu của thể hiện B

| | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|
| $T(i)$ | R | R | R | R | W | W | W | W |
| $T(k)$ | R | R | W | W | R | R | W | W |
| $T(j)$ | R | W | R | W | R | W | R | W |
| Error | N | N | Y | Y | N | Y | Y | Y |

- Xác định dãy các thao tác đọc ghi dữ liệu
- Xác định các Schedule dạng $\langle T^A(i), T^B(k), T^A(j) \rangle$
- Với mỗi Schedule : xác định trạng thái dữ liệu qua các bước thực hiện Schedule.

- Overview
- Testing DB application
- Kỹ thuật kiểm chứng
- Công cụ kiểm chứng
- Q&A

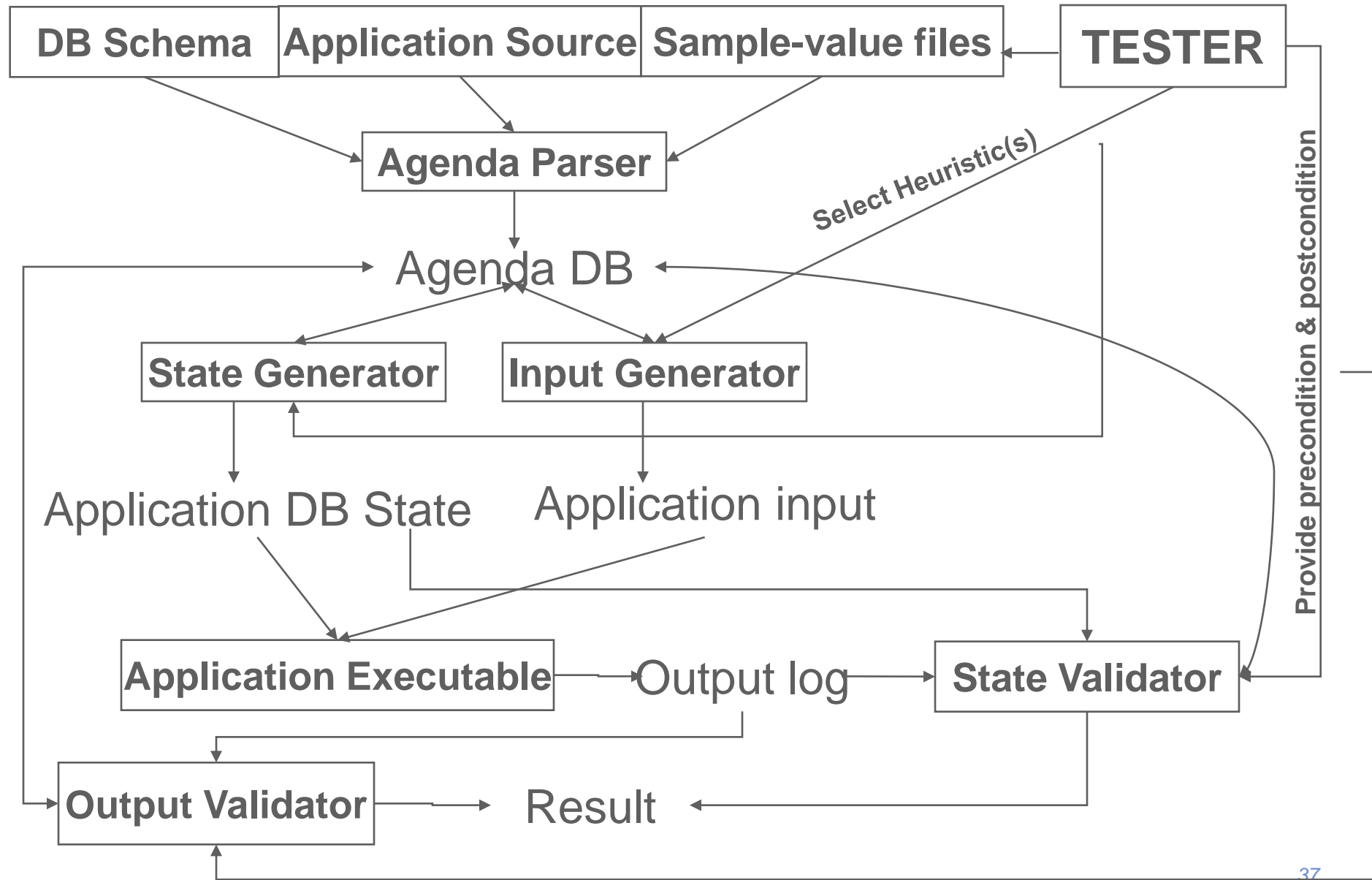
Tools : AGENDA - Introduce

- **Introduces a framework for testing database applications and describes a complete tool set, based on this framework, which has been prototyped.**
- **The main purpose of this prototype is to populate a database, generate test inputs to an application, execute the application on those inputs, and check for some aspects of correctness in both the application output and the resulting database state.**

AGENDA : COMPONENTS

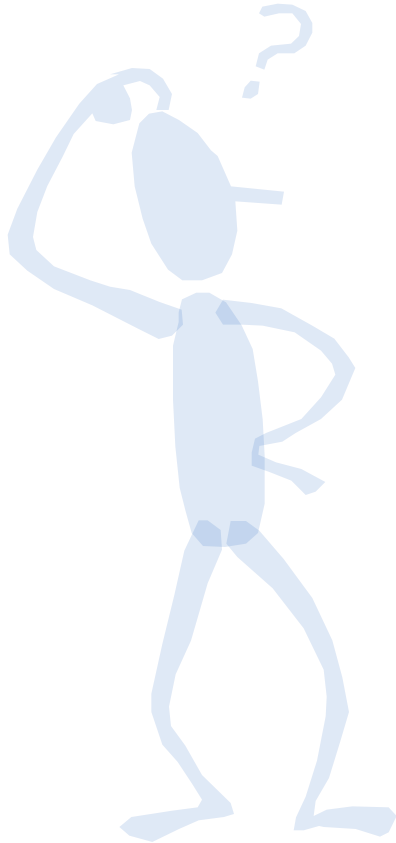
- A Parser that gathers relevant information from the database schema and application
- A State Generator that populates the database with meaningful data that satisfy database constraints
- An Input generation tool that generates test cases for the application
- A State Validator that checks the resulting database state after operations are performed by the application
- A Output Validator tool that assists the tester in checking the database application's output.

AGENDA : Architecture

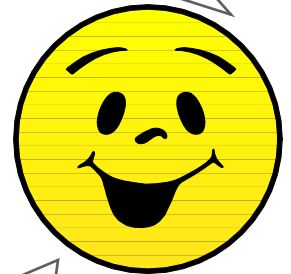


- Overview
- Testing DB application
- Kỹ thuật kiểm chứng
- Công cụ kiểm chứng
- **Q&A**

- **Software Testing Techniques, Second Edition** (Chapter 4, 5)
- **Testing Database Transaction Concurrency**
(*Department of Computer and Information Science,
Polytechnic University, Brooklyn, NY 11201
USA, {ytdeng, phyllis, zchen}@cis.poly.edu*)
- ***Database Application Testing*** (Presented by Shreyas Kale)



Understand us ?



**Anyway, thanks for
your attention**