Regression Testing Techniques for Relational Database Applications

Gregory M. Kapfhammer†

Department of Computer Science
Allegheny College
http://www.cs.allegheny.edu/~gkapfham/

University of Ulm – January 23, 2012

† Joint with Mary Lou Soffa (University of Virginia) and Jonathan Miller Kauffman (Allegheny College)
Presenter Introduction: Gregory M. Kapfhammer
Software and Data Challenges

Software and Data are Everywhere

Diagram:

- Program
- Computer Server
Software and Data Challenges

Software and Data are Everywhere

- Program
- Desktop Computer
- Program
- Computer Server
Software and Data Challenges

Software and Data are Everywhere

Program

Desktop Computer

Program

Computer Server

Program

Mobile Computer

Kapfhammer Allegheny College

Regression Testing Techniques for Relational Database Applications
Software and Data Challenges

Software and Data are Everywhere

Program

Desktop Computer

Program

Computer Server

Program

Mobile Computer

Program

Household Appliance
Software and Data Challenges

Software and Data are Everywhere

- Program
- Desktop Computer
- Scientific Device
- Program
- Computer Server
- Household Appliance
- Program
- Mobile Computer
Software and Data are Everywhere

- Program
- Desktop Computer
- Scientific Device
- Program
- Computer Server
- Household Appliance
- Program
- Mobile Computer
- Network Router
Software and Data Challenges

Software and Data are Everywhere

- Program
- Desktop Computer
- Scientific Device

- Program
- Computer Server
- Household Appliance

- Program
- Mobile Computer
- Network Router
Software and Data are Everywhere

- Program
  - Desktop Computer
  - Scientific Device

- Program
  - Computer Server
  - Household Appliance

- Program
  - Mobile Computer
  - Network Router
Software and Data Challenges

Software and Data are Everywhere

- Program
  - Desktop Computer
  - Scientific Device
- Program
  - Computer Server
  - Household Appliance
- Program
  - Mobile Computer
  - Network Router
Software and Data Challenges

Software and Data are Everywhere

Program

Desktop Computer

Computer Server

Mobile Computer

Program

Scientific Device

Household Appliance

Network Router

Kapfhammer Allegheny College

Regression Testing Techniques for Relational Database Applications
Software and Data Challenges

Software and Data are Everywhere

Program
Desktop Computer

Program
Computer Server

Program
Mobile Computer

Program
Scientific Device

Program
Household Appliance

Program
Network Router
Software and Data Challenges

Software and Data are Everywhere

- Program
  - Desktop Computer
- Program
  - Computer Server
- Program
  - Mobile Computer
- Program
  - Scientific Device
- Program
  - Household Appliance
- Program
  - Network Router
Software and Data Challenges

Software Complexity and Data Enormity

Computer Software
Software Complexity and Data Enormity
Software Complexity and Data Enormity

Software and Data Challenges

- computer software
- lines of code
- numerous features
Software Complexity and Data Enormity

- Computer Software
  - Lines of Code
  - Numerous Features
  - Feature Interactions
Software Complexity and Data Enormity

- Lines of Code
- Numerous Features
- Feature Interactions
- Execution Environments
Software Complexity and Data Enormity

Software entities are more complex for their size than perhaps any other human construct - Frederick P. Brooks, Jr.
Software and Data Challenges

Software Complexity and Data Enormity

Prediction: in 2011, 1.8 zettabytes (i.e., 1.8 trillion gigabytes) of data will be created - IDC Digital Universe Study
Software and Data Challenges

Software and Data are Evolving

Program

Execution Environment
Software and Data Challenges

Software and Data are Evolving

```
Program

Execution Environment

Program

Execution Environment
```
Software and Data Challenges

Software and Data are Evolving

Program Changed because of the addition of a new feature or the correction of a defect
Software and Data Challenges

Software and Data are Evolving

Program

Execution Environment
Software and Data Challenges

Software and Data are Evolving

Program

Execution Environment

Program

Execution Environment
Software and Data Challenges

Software and Data are Evolving

Execution Environment Changed due to modification of a kernel, device driver, or relational database
Relational Database Challenges

An Interesting Defect Report

Database Server Crashes
An Interesting Defect Report

When you run a complex query against Microsoft SQL Server 2000, the SQL Server scheduler may stop responding. Additionally, you receive an error message that resembles the following: **Date Time server Error: 17883 Severity: 1, State: 0** Date Time server Process 52:0 (94c) ...
An Interesting Defect Report

Input-Dependent Defect
An Interesting Defect Report

This problem occurs when one or more of the following conditions are true: The query contains a `UNION` clause or a `UNIONALL` clause that affects many columns. The query contains several `JOIN` statements. The query has a large estimated cost. **BUG 473858 (SQL Server 8.0)**
The Risks Digest, Volume 22, Issue 64, 2003

**Jeppesen reports airspace boundary problems**

About 350 airspace boundaries contained in Jeppesen NavData are incorrect, the FAA has warned. The error occurred at Jeppesen after a software upgrade when information was pulled from a database containing 20,000 airspace boundaries worldwide for the March NavData update, which takes effect March 20.
Relational Database Challenges

Real-World Defective Database Application

The Risks Digest, Volume 22, Issue 64, 2003

Jeppesen reports airspace boundary problems

About 350 airspace boundaries contained in Jeppesen NavData are incorrect, the FAA has warned. The error occurred at Jeppesen after a software upgrade when information was pulled from a database containing 20,000 airspace boundaries worldwide for the March NavData update, which takes effect March 20.

Practically all use of databases occurs from within application programs [Silberschatz et al., 2006, pg. 311]
Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS).
The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS).

A *schema* is a collection of table definitions:

```
CREATE TABLE person (
    id INT,
    name VARCHAR(100) NOT NULL,
    age INT(3),
    PRIMARY KEY (id)
)
```
Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS).

The *data manipulation language* supports several operations:

```
SELECT name FROM person WHERE age >= 30 AND age <= 40
```
Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS).

The *data manipulation language* supports several operations:

```
UPDATE person SET name = Jan WHERE id = 2
```
Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS).

The *data manipulation language* supports several operations:

```
INSERT INTO person (id, name, age) VALUES (1, John, 38)
```
Structured Query Language

The *structured query language* (SQL) is an established standard and a query and manipulation language for *relational database management systems* (RDBMS).

The *data manipulation language* supports several operations:

```
DELETE FROM person WHERE id = 2
```
# Relational Database Tables

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chalker Conrad</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Abby Clulow</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>David Rogan</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Stacie Reckling</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Megan Hartnup</td>
<td>29</td>
</tr>
</tbody>
</table>
# Relational Database Tables

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chalker Conrad</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Abby Clulow</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>David Rogan</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Stacie Reckling</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Megan Hartnup</td>
<td>29</td>
</tr>
</tbody>
</table>
## Relational Database Tables

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chalker Conrad</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Abby Clulow</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>David Rogan</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Stacie Reckling</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Megan Hartnup</td>
<td>29</td>
</tr>
</tbody>
</table>
# Relational Database Tables

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chalker Conrad</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Abby Clulow</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>David Rogan</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Stacie Reckling</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Megan Hartnup</td>
<td>29</td>
</tr>
</tbody>
</table>
## Relational Database Table

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chalker Conrad</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Abby Clulow</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>David Rogan</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Stacie Reckling</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Megan Hartnup</td>
<td>29</td>
</tr>
</tbody>
</table>
### Relational Database Tables

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chalker Conrad</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Abby Clulow</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>David Rogan</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Stacie Reckling</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Megan Hartnup</td>
<td>29</td>
</tr>
</tbody>
</table>
Programs and Databases

Database Applications

Program
Programs and Databases

Database Applications

- Program
- Relational Database Management System
- Database State

Kapfhammer
Allegheny College

Regression Testing Techniques for Relational Database Applications
Database Applications

Data Manipulation Language (DML) Statements

Program \( \rightarrow \) select \( \rightarrow \) Relational Database Management System

select query

Database State
Database Applications

Data Manipulation Language (DML) Statements

Program \(\xrightarrow{\text{update}}\) Relational Database Management System \(\xrightarrow{\text{modify}}\) Database State
Programs and Databases

Database Applications

Data Manipulation Language (DML) Statements

Program \(\xrightarrow{\text{insert}}\) Relational Database Management System

\(\xrightarrow{\text{modify}}\) Database State
Programs and Databases

Database Applications

Data Manipulation Language (DML) Statements

Program \(\rightarrow\) delete \(\rightarrow\) Relational Database Management System

modify \(\downarrow\)

Database State

Kapfhammer Allegheny College

Regression Testing Techniques for Relational Database Applications
Database Applications

Data Definition Language (DDL) Statements

Program

create table

Relational Database Management System

modify

Database Structure
Can we categorize the different ways of implementing database applications?
Categorizing Database Applications
Categorizing Database Applications
Categorizing Database Applications

Programs and Databases

Kapfhammer

Allegheny College

Regression Testing Techniques for Relational Database Applications
Categorizing Database Applications

- Interaction Approach
- Program Location
- Interface
Categorizing Database Applications

- Interaction Approach
- Program Location
- Interface
- Embedded
Categorizing Database Applications

- Database Applications
  - Interaction Approach
    - Interface
    - Embedded
  - Program Location
    - Outside RDBMS
Categorizing Database Applications

- Interaction Approach
  - Interface
  - Embedded

- Program Location
  - Outside RDBMS
  - Inside RDBMS
Categorizing Database Applications

Database Applications

Interaction Approach

- Interface
- Embedded

Program Location

- Outside RDBMS
- Inside RDBMS
Categorizing Database Applications

**Database Applications**

- **Interaction Approach**
  - **Interface**
  - **Embedded**

- **Program Location**
  - **Outside RDBMS**
  - **Inside RDBMS**

Java application that submits SQL strings to HSQLDB using JDBC
Evolution of Database Applications

Program → DML Command → Relational Database Management System
Evolution of Database Applications

Program → DML Command → Relational Database Management System

- Database State
- Database Structure
Evolution of Database Applications

Only the database administrator can add new constraints to the schema!
Evolution of Database Applications

The programmers encode the constraints in the program’s source code!
Evolution of Database Applications

The programmers encode the constraints in the program’s source code!

Program $\rightarrow$ DML Command $\rightarrow$ Relational Database Management System $\rightarrow$ Database State $\rightarrow$ Database Structure

Constraint $C_i$

Constraint $C_j$
Evolution of Database Applications

The programmers encode the constraints in the program’s source code!
Programmers make other changes to the source code of the program.
Evolution of Database Applications

External programs can change the state of the relational database

Program -> DML Command -> Relational Database Management System

- Constraint $C_i$
- Constraint $C_j$
- Constraint $C_k$

Database State

Database Structure
Evolution of Database Applications

Database administrator can change the structure of the database

Program

DML Command

Relational Database Management System

Database State

Database Structure

Constraint $C_i$

Constraint $C_j$

Constraint $C_k$
Evolution of Database Applications

How can we test a rapidly changing database application?

Program → DML Command → Relational Database Management System

- Constraint $C_i$
- Constraint $C_j$
- Constraint $C_k$

Database State
Database Structure
Important Techniques

Regression Testing to the Rescue
Regression Testing to the Rescue
Regression Testing to the Rescue
Regression Testing to the Rescue
Regression Testing supports the efficient construction of database software that is complex and rapidly evolving.
What is a Test Case?

Method
Under Test
What is a Test Case?
Important Techniques

What is a Test Case?

Input → Method Under Test → Output

Kapfhammer
Regression Testing Techniques for Relational Database Applications
Allegheny College
What is a Test Case?

Diagram: Test Set Up → Method Under Test
What is a Test Case?

- **Input**
- **Method Under Test**
- **Test Set Up**

Kapfhammer Allegheny College

Regression Testing Techniques for Relational Database Applications
What is a Test Case?

![Diagram showing a test case flow]

- Input
- Method Under Test
- Test Set Up
- Output
What is a Test Case?

Input → Method Under Test → Output

Method Under Test → Test Clean Up → Test Set Up

Test Set Up
What is a Test Case?
What is a Test Case?

Input → Method Under Test → Output → Expected Output

Set Up → Test → Clean Up

Test Oracle
What is a Test Case?

- Input
- Method Under Test
- Output
- Expected Output
- Test Oracle
- Test Verdict
- Test Clean Up
- Test Set Up
What is a Test Case?

- **Input**
  - **Method Under Test**
    - **Test Set Up**
    - **Test Clean Up**
  - **Output**
  - **Test Oracle**
    - **Expected Output**
    - **Test Verdict**
What is a Test Case?

The test case passes and the code is correct!
What is a Test Case?

- **Input**
- **Method Under Test**
- **Output**
- **Test Oracle**
- **Expected Output**
- **Test Verdict**

- **Test Set Up**
- **Test Clean Up**
What is a Test Case?

The test case fails and a defect is found!
Important Techniques

What is a Test Suite?

$T_1$
What is a Test Suite?

\[ T_1 \rightarrow T_2 \]
What is a Test Suite?

$T_1 \rightarrow T_2 \rightarrow T_3$
What is a Test Suite?

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4$
What is a Test Suite?

- $T_1$
- $T_2$
- $T_3$
- $T_4$
- …
What is a Test Suite?
What is a Test Suite?

Organize the Test Cases into a Test Suite

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow \ldots \rightarrow T_n$
What is a Test Suite?

Organize the Test Cases into a Test Suite

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow \ldots \rightarrow T_n$

Tool Support for Software Testing?
What is a Test Suite?

Organize the Test Cases into a Test Suite

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow \ldots \rightarrow T_n$

Tool Support for Software Testing?

JUnit

Kapfhammer

Allegheny College

Regression Testing Techniques for Relational Database Applications
What is a Test Suite?

Organize the Test Cases into a Test Suite

Tool Support for Software Testing?

JUnit

Apache Ant
What is a Test Suite?

Organize the Test Cases into a Test Suite

Tool Support for Software Testing?

JUnit
Apache Ant
DBUnit
Test Suite Management

Organize the Test Cases into a Test Suite

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow \ldots \rightarrow T_n$
Test Suite Management

Regression Testing Technique
Important Techniques

Test Suite Management

What if Some Test Cases are More Effective?

Regression Testing Technique

Kapfhammer Allegheny College
Regression Testing Techniques for Relational Database Applications
Important Techniques

Test Suite Management

What if Some Test Cases are More Effective?

Regression Testing Technique

Prioritization

Kapfhammer
Allegheny College

Regression Testing Techniques for Relational Database Applications
Important Techniques

Test Suite Management

What if Some Test Cases are More Effective?

Regression Testing Technique

Prioritization

Kapfhammer Allegheny College

Regression Testing Techniques for Relational Database Applications
Test Suite Management

What if Some Test Cases are Redundant?

Regression Testing Technique

Prioritization
Test Suite Management

What if Some Test Cases are Redundant?

Regression Testing Technique

Prioritization  Reduction
Test Suite Management

What if Some Test Cases are Redundant?

Regression Testing Technique

Prioritization

Reduction
Test Suite Management

What if Only Certain Tests are Needed?

Regression Testing Technique

Prioritization

Reduction
Test Suite Management

What if Only Certain Tests are Needed?

Regression Testing Technique

- Prioritization
- Reduction
- Selection
Model of Regression Testing
Model of Regression Testing

![Diagram showing the model of regression testing with steps: Start, Coverage Report]
Model of Regression Testing

Start \rightarrow Coverage Report

- Selection
- Reduction
- Prioritization
Model of Regression Testing

Original Test Suite

Selection
Reduction
Prioritization

Start → Coverage Report
Model of Regression Testing

Original Test Suite

- Selection
- Reduction
- Prioritization

Modified Test Suite

Start

Coverage Report
Model of Regression Testing

- Start
- Coverage Report
- Original Test Suite
- Selection
- Reduction
- Prioritization
- Modified Test Suite
- Test Suite Execution
- Test Coverage Monitoring
Important Techniques

Model of Regression Testing

Start → Coverage Report → Original Test Suite

- Selection
- Reduction
- Prioritization

→ Modified Test Suite

→ Test Suite Execution

→ Test Coverage Monitoring

→ Program

→ Adequacy Criterion
Model of Regression Testing
Model of Regression Testing

Start -> Coverage Report -> Original Test Suite
          |                        | Selection
          |                        | Reduction
          |                        | Prioritization
          |                        | Modified Test Suite
          |                        | Test Suite Execution
          |                        | Test Coverage Monitoring
          |                        | Test Results
          |                        | End
Important Techniques

Model of Regression Testing

Use the Coverage Report During the Next Round of Regression Testing
Model of Regression Testing

Use the Same Test Suite for the Next Round of Regression Testing
Model of Regression Testing

Important Techniques

1. Start
2. Coverage Report
3. Selection
4. Reduction
5. Prioritization
6. Original Test Suite
7. Modified Test Suite
8. Test Suite Execution
9. Test Coverage Monitoring
10. Program
11. Adequacy Criterion
12. Test Results
13. End
14. Standard Repeat
15. Version Specific Repeat
16. Make a New Test Suite for the Next Round of Regression Testing
Test Suite Adequacy

\[ T_1 \quad T_2 \]
Important Techniques

Test Suite Adequacy

$T_1\quad T_2\quad T_3\quad T_4$
Important Techniques

Test Suite Adequacy

$T_1$  $T_2$  $T_3$  $T_4$  $T_5$  $T_6$
Test Suite Adequacy

$T_1$  $T_2$  $T_3$  $T_4$  $T_5$  $T_6$  $T_7$  $T_8$
Test Suite Adequacy

$T_1$ $T_2$ $T_3$ $T_4$ $T_5$ $T_6$ $T_7$ $T_8$ $T_9$ $T_{10}$
Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

$T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}$

$R_1, R_2, R_3, R_4$
Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

$T_1$  $T_2$  $T_3$  $T_4$  $T_5$  $T_6$  $T_7$  $T_8$  $T_9$  $T_{10}$

$R_1$  $R_2$  $R_3$  $R_4$  $R_5$  $R_6$
Test Suite Adequacy

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

\( T_1 \) \( T_2 \) \( T_3 \) \( T_4 \) \( T_5 \) \( T_6 \) \( T_7 \) \( T_8 \) \( T_9 \) \( T_{10} \)

\( R_1 \) \( R_2 \) \( R_3 \) \( R_4 \) \( R_5 \) \( R_6 \) \( R_7 \) \( R_8 \)
Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Important Techniques

**Test Suite Adequacy**

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Test Suite Adequacy

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \)
Important Techniques

**Test Suite Adequacy**

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Test Suite Adequacy

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Important Techniques

Test Suite Execution

$T_1$, $T_2$, $T_3$, $T_4$, $T_5$, $T_6$, $T_7$, $T_8$, $T_9$, $T_{10}$
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks

Run Test Case
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks

Passing Test Case: $O_A = O_E$
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks

Run Test Case
Test Suite Execution

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

JUnit and DBUnit Test Automation Frameworks

Failing Test Case: \( O_A \neq O_E \)
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks

Failing Test Case: $O_A \neq O_E$

Stop Running $T$
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks

Failing Test Case: $O_A \neq O_E$

Stop Running $T$
Test Suite Execution

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

JUnit and DBUnit Test Automation Frameworks

Failing Test Case: \( O_A \neq O_E \)

Stop Running \( T \)
**Test Suite Execution**

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks

Failing Test Case: $O_A \neq O_E$

Stop Running $T$  
Continue Running $T$
Test Suite Execution

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks

Failing Test Case: $O_A \neq O_E$

Stop Running $T$

Continue Running $T$
Test Coverage Monitoring

$T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}$
Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Test Coverage Monitoring

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

JUnit and DBUnit Test Automation Frameworks
Database-Aware Test Coverage Monitor
Proteja Test Suite Manager
Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

JUnit and DBUnit Test Automation Frameworks
Database-Aware Test Coverage Monitor
Proteja Test Suite Manager

Run Test Case
Collect Per-Test Case Coverage
Important Techniques

Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Test Coverage Monitoring

Test Suite: \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set: \( R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \)
Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Test Coverage Monitoring

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R \) for ... Statement Coverage
Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R$ for ... Database Interaction Coverage
Test Coverage Monitoring

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R \) for ... Database Table Coverage
Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R$ for ... Database Record Coverage
Important Techniques

Test Coverage Monitoring

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R \) for ... Database Attribute Coverage
Test Coverage Monitoring

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R \) for ... Database Attribute-Value Coverage
Important Techniques

Test Coverage Monitoring

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

See [Kapfhammer and Soffa, ISEC 2008] for more details
Important Techniques

Greedy Algorithms

$T_1 \quad T_2 \quad T_3 \quad T_4 \quad T_5 \quad T_6 \quad T_7 \quad T_8 \quad T_9 \quad T_{10}$
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$
Greedy Algorithms

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \)
Important Techniques

Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Greedy Algorithms

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \)
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10}\rangle$

$R_1$ $R_2$ $R_3$ $R_4$ $R_5$ $R_6$ $R_7$ $R_8$ $R_9$ $R_{10}$ $R_{11}$ $R_{12}$

$T_1$ $T_2$ $T_3$ $T_4$ $T_5$ $T_6$ $T_7$ $T_8$ $T_9$ $T_{10}$

$\text{ratio}(T_1) = \frac{\text{cost}(T_1)}{\text{coverage}(T_1)}$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12}\}$
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$

ratio$(T_1) = \frac{5}{3}$
Greedy Algorithms

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \)

\( \text{ratio}(T_1) = 1.66 \)
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
**Greedy Algorithms**

**Test Suite** 

\[ T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \]

**Requirements Set** 

\[ R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \]
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$

$\text{ratio}(T_5) = \frac{\text{cost}(T_5)}{\text{coverage}(T_5)}$
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Greedy Algorithms

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

Requirements Set \( R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \)

ratio\((T_5) = 2.66\)
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Greedy Algorithms

Test Suite $T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle$

$1.66 < 2.66$

$\text{ratio}(T_1) < \text{ratio}(T_5)$

Prefer $T_1$ over $T_5$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Greedy Algorithms

Test Suite \( T = \langle T_1, T_2, \ldots, T_9, T_{10} \rangle \)

\( R \) is a set of requirements: \( R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \)

Proceed incrementally, picking the test case with the lowest ratio value for the uncovered requirements.
Greedy Algorithms

Test Suite $T = \langle T_8, T_4, T_9, T_1, T_{10}, T_3, T_7, T_2, T_6, T_5 \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
**Greedy Algorithms**

Imported Techniques

\[ R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \} \]
Important Techniques

Greedy Algorithms

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Important Techniques

Greedy Algorithms

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$
Greedy Algorithms

Important Techniques

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$
Greedy Algorithms

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$
Greedy Algorithms

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$
Greedy Algorithms

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Greedy Algorithms

Requirements Set $R = \{R_1, R_2, \ldots, R_{11}, R_{12}\}$
Greedy Algorithms

Important Techniques

Requirements Set

$$R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$$
Greedy Algorithms

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Greedy Algorithms

Test Suite $T = \langle T_8, T_4, T_9, T_1, T_{10}, T_3, T_7 \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
Greedy Algorithms

Test Suite $T = \langle T_8, T_4, T_9, T_1, T_{10}, T_3, T_7 \rangle$

Requirements Set $R = \{ R_1, R_2, \ldots, R_{11}, R_{12} \}$
### Empirical Results – Test Suite Reduction

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RM (13)</td>
<td>(7, .46)</td>
<td>(7, .46)</td>
<td>(10, .30)</td>
<td>(9, .31)</td>
<td>(8.25, .37)</td>
</tr>
<tr>
<td>FF (16)</td>
<td>(7, .56)</td>
<td>(7, .56)</td>
<td>(11, .31)</td>
<td>(11, .31)</td>
<td>(9, .44)</td>
</tr>
<tr>
<td>PI (15)</td>
<td>(6, .60)</td>
<td>(6, .60)</td>
<td>(8, .70)</td>
<td>(7, .53)</td>
<td>(6.75, .55)</td>
</tr>
<tr>
<td>ST (25)</td>
<td>(5, .80)</td>
<td>(5, .76)</td>
<td>(11, .56)</td>
<td>(10, .60)</td>
<td>(7.75, .690)</td>
</tr>
<tr>
<td>TM (27)</td>
<td>(14, .48)</td>
<td>(14, .48)</td>
<td>(15, .45)</td>
<td>(14, .48)</td>
<td>(14.25, .47)</td>
</tr>
<tr>
<td>GB (51)</td>
<td>(33, .35)</td>
<td>(33, .35)</td>
<td>(33, .35)</td>
<td>(32, .37)</td>
<td>(32.75, .36)</td>
</tr>
<tr>
<td><strong>All (24.5)</strong></td>
<td>(12, .51)</td>
<td>(12.17, .50)</td>
<td>(14.67, .40)</td>
<td>(13.83, .44)</td>
<td></td>
</tr>
</tbody>
</table>

- Reduction values range from .30 to .80
- Reduction level varies depending on interaction granularity
- How will the reduction of a test suite impact defect detection?
Empirical Results – Test Suite Reduction

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RM (13)</td>
<td>(7, .46)</td>
<td>(7, .46)</td>
<td>(10, .30)</td>
<td>(9, .31)</td>
<td>(8.25, .37)</td>
</tr>
<tr>
<td>FF (16)</td>
<td>(7, .56)</td>
<td>(7, .56)</td>
<td>(11, .31)</td>
<td>(11, .31)</td>
<td>(9, .44)</td>
</tr>
<tr>
<td>PI (15)</td>
<td>(6, .60)</td>
<td>(6, .60)</td>
<td>(8, .70)</td>
<td>(7, .53)</td>
<td>(6.75, .55)</td>
</tr>
<tr>
<td>ST (25)</td>
<td>(5, .80)</td>
<td>(5, .76)</td>
<td>(11, .56)</td>
<td>(10, .60)</td>
<td>(7.75, .690)</td>
</tr>
<tr>
<td>TM (27)</td>
<td>(14, .48)</td>
<td>(14, .48)</td>
<td>(15, .45)</td>
<td>(14, .48)</td>
<td>(14.25, .47)</td>
</tr>
<tr>
<td>GB (51)</td>
<td>(33, .35)</td>
<td>(33, .35)</td>
<td>(33, .35)</td>
<td>(32, .37)</td>
<td>(32.75, .36)</td>
</tr>
</tbody>
</table>

- Reduction values range from .30 to .80
- Reduction level varies depending on interaction granularity
- How will the reduction of a test suite impact defect detection?
Conclusion

- Databases are widely used in real-world applications
- Database applications have complex state and structure
- Source code, database state, and relational schema evolve
- Prioritization techniques can increase effectiveness
- Reduction methods can improve the efficiency of testing

Future Work

- New empirical studies of database-aware regression testing
- Implement and release free and open source testing tools
Conclusion

- Databases are widely used in real-world applications
- Database applications have complex state and structure
- Source code, database state, and relational schema evolve
- Prioritization techniques can increase effectiveness
- Reduction methods can improve the efficiency of testing

Future Work

- New empirical studies of database-aware regression testing
- Implement and release free and open source testing tools
Regression Testing Techniques for Relational Database Applications

Gregory M. Kapfhammer

Department of Computer Science
Allegheny College
http://www.cs.allegheny.edu/~gkapfham/

Thank you for your attention!
I welcome your questions and comments.