

HI, THIS IS
YOUR SON'S SCHOOL.
WE'RE HAVING SOME
COMPUTER TROUBLE.



OH, DEAR - DID HE
BREAK SOMETHING?
IN A WAY -)



DID YOU REALLY
NAME YOUR SON
Robert'); DROP
TABLE Students;-- ?

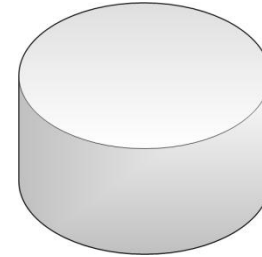


OH. YES. LITTLE
BOBBY TABLES,
WE CALL HIM.

WELL, WE'VE LOST THIS
YEAR'S STUDENT RECORDS.
I HOPE YOU'RE HAPPY.



AND I HOPE
YOU'VE LEARNED
TO SANITIZE YOUR
DATABASE INPUTS.



Thesis defense

Type Safe Integration of Query Languages into Scala

Christopher Vogt, 24.08.2011



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

RWTHAACHEN
UNIVERSITY

Motivation

Relational database trouble

1. SQL as Strings has problems

- no compile time guarantees
- runtime exceptions
- injection vulnerabilities
- composition
- ...

2. Data model mismatch

relational vs. programming language



Different existing solutions

- ORM: Object-Relational Mapping (JPA, etc.)
- LINQ: Language Integrated Query (on .NET)
- ...

My approach

SIQ: Scala Integrated Query

- type safe database query library
- compiles a subset of Scala to SQL
- similar to LINQ / LINQ-to-SQL

Feature Comparison

	SQL	Scala Query	SQueryl	Microsoft's LINQ-to-SQL on .NET	Scala Integrated Query
Type Safety		✓	✓	✓	✓
Nested Results				✓	✓
Avalanche Safe Compilation					✓
Backend Specific Type Safety					✓
...					

Agenda

1. Scala
2. Scala to SQL compilation
(Avalanche safe)
3. Implementation
4. Type safety of queries
(backend specific)
5. Summary

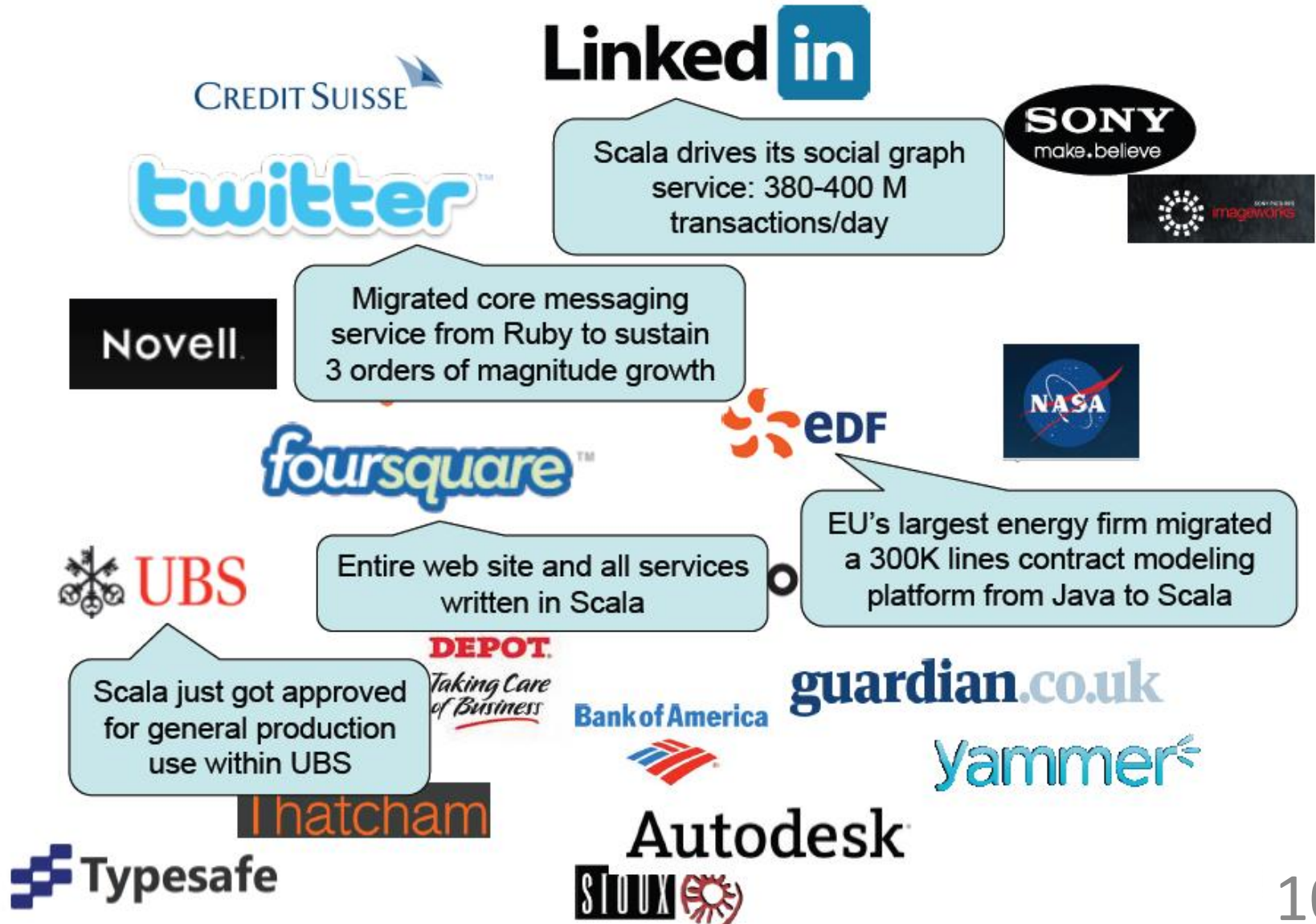
SCALA

**A MODERN, GENERAL PURPOSE
PROGRAMMING LANGUAGE**

What people say about Scala

- “If I were to pick a language to use today other than Java, it would be *Scala*”
James Gosling, creator of Java
- “If someone had shown me [Scala] in 2003 I'd probably have never created Groovy”
James Strachan, creator of Groovy

Who uses Scala?



Who created Scala?

- Martin Odersky and his team
 - Professor at EPFL Lausanne
 - Original author of javac
 - Co-author of Java generics

What is Scala?

Object-oriented

Performant,
Strong
compile time
guarantees



Agile,
Lightweight
Syntax

Functional

More on Scala

- Java-similar syntax
- runs on the JVM
- fully interoperable with Java
- features
 - powerful type system, type inference, mix-in composition, operator overloading, implicit conversions, pattern matching, immutability, singleton objects, ...
- please ask syntax questions!

Agenda

1. Scala
2. Scala to SQL compilation
(Avalanche safe)
3. Implementation
4. Type safety of queries
(backend specific)
5. Summary

SCALA TO SQL COMPILATION (AVALANCHE SAFE)

Data model mismatch

- SIQ data model \neq relational data model
- relational results always flat
- SIQ results can be nested

=>

SIQ encodes nesting in flat results

Relational data model

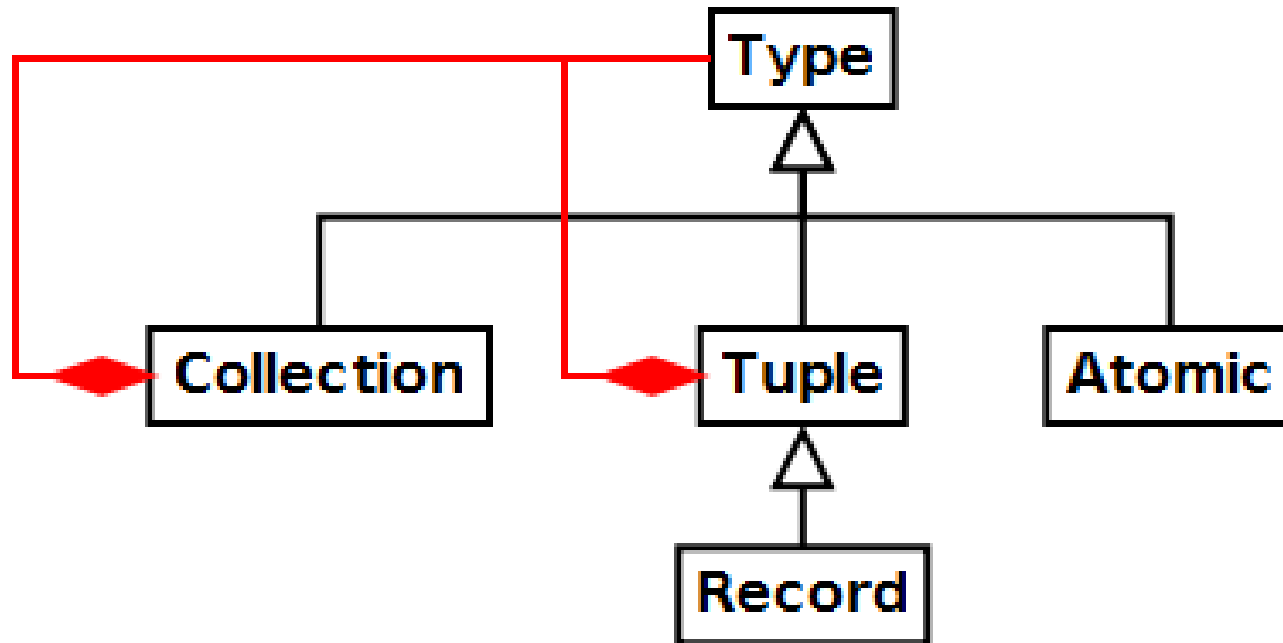


Example

SELECT * FROM nation

nationkey	name
1	ARGENTINA
2	BRAZIL
...	...
7	GERMANY
...	...

SIQ data model



Example (pseudo code)

for **nation** in nations

(**nation**, **customers** of this nation)

Result type:

Collection[(**Nation**, Collection[**Customer**])]

How many SQL Queries?

for **nation** in nations

(**nation**, **customers** of this nation)

Avalanche
Safe

	SQL	Scala Query	SQueryl	Microsoft's LINQ-to-SQL on .NET	Scala Integrated Query
written as loop	$n + 1$	$n + 1$	$n + 1$	$n + 1$	2
manually encoded	1-2	1-2	1-2	1-2	1-2

$n = \text{nations.length}$

Avalanche Safety

Number of SQL queries

- = number of Collections in type
- only depends on result type
- does not depend on database size

Result type:

Collection[(Nation, Collection[Customer])]



2

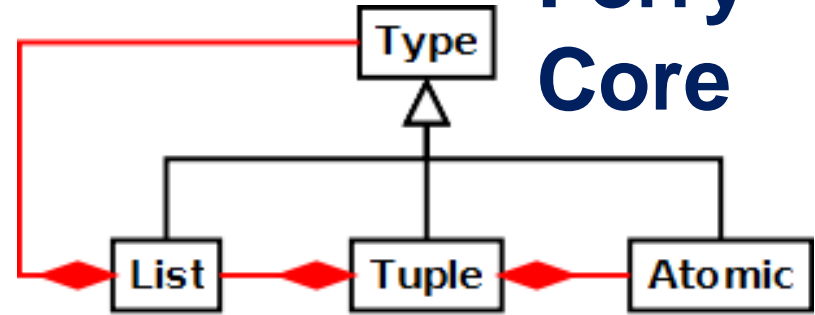
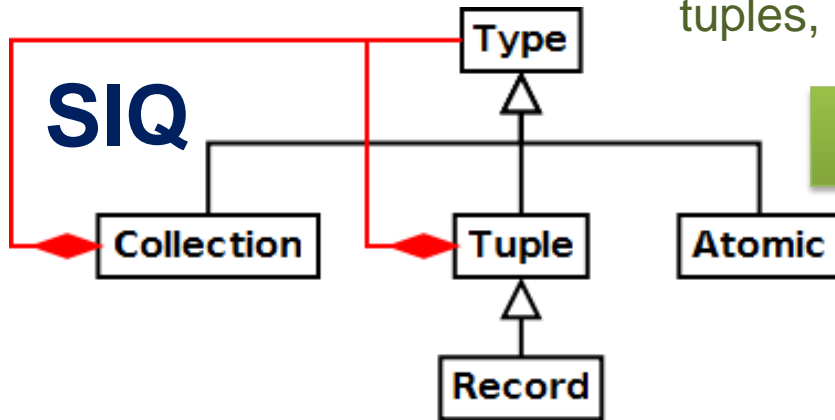
SIQ uses Ferry compilation

- Ferry Core \rightarrow relational algebra
- Avalanche safe
- by University of Tübingen

Compilation process

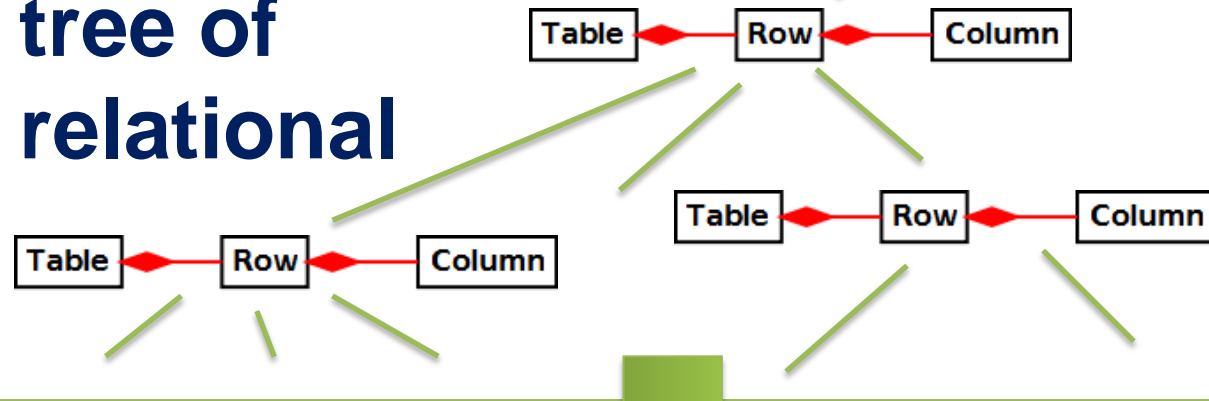
**Ferry
Core**

Flattening of
tuples, etc.



Encoding as several
SQL queries

**tree of
relational**



execution

SQL generation

SIQ results

decoding
of results

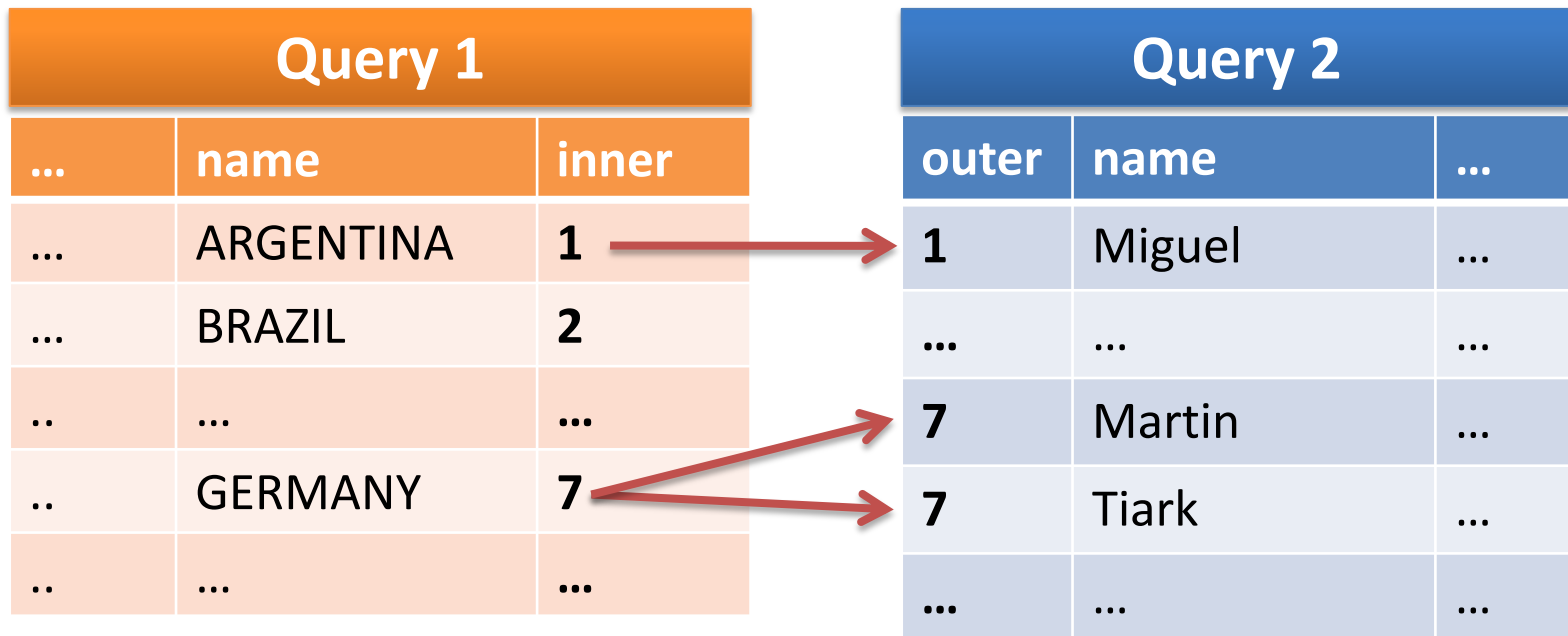
tree of relational results

tree of SQL₂₂

Relational encoding at a glance

for **nation** in nations
(**nation**, **customers** of this nation)

Relational encoding:



More in my thesis

- formal description of all involved steps
- prototype implementation

Agenda

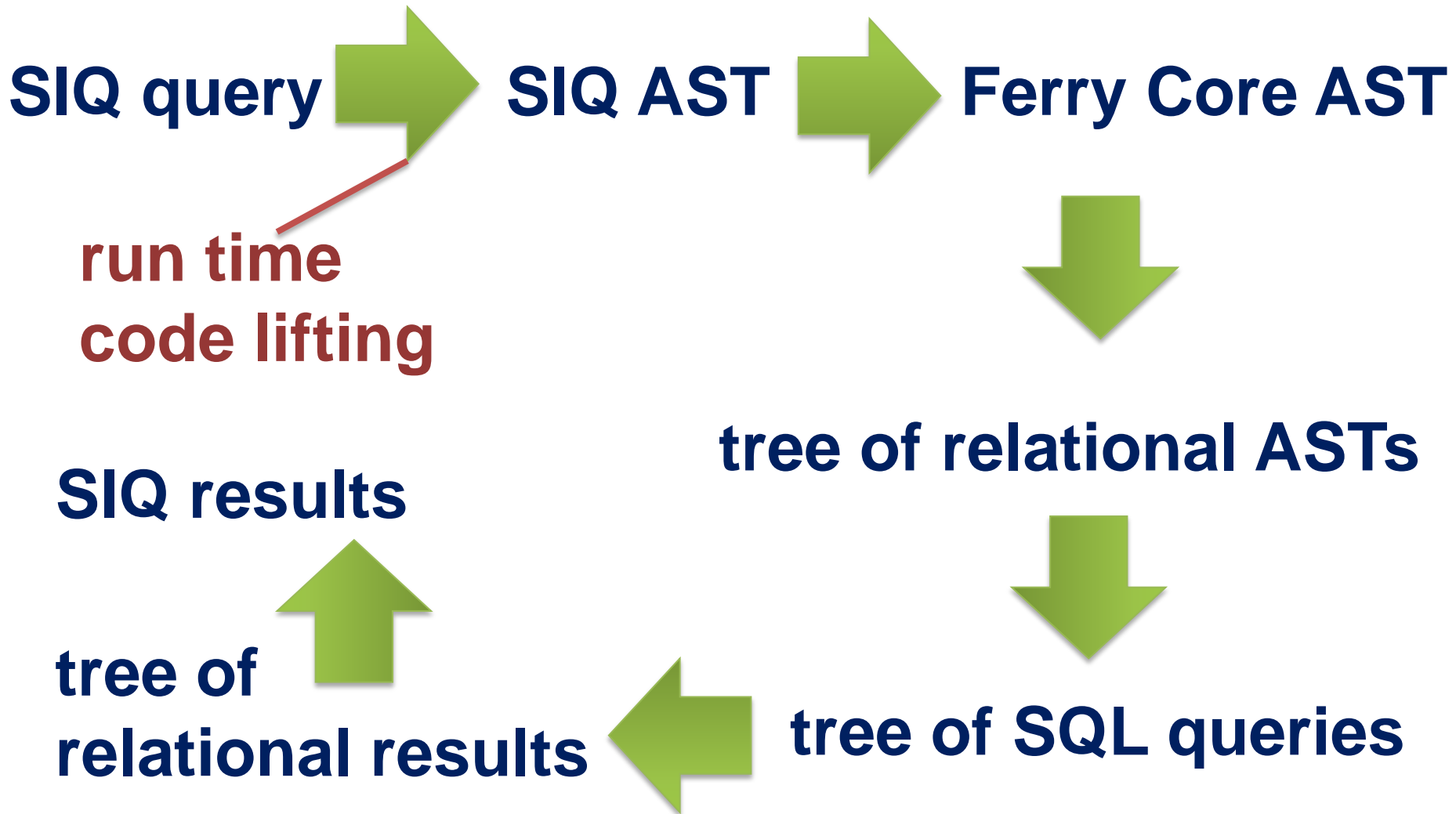
1. Scala
2. Scala to SQL compilation
(Avalanche safe)
3. Implementation
4. Type safety of queries
(backend specific)
5. Summary

IMPLEMENTATION

SIQ Implementation

- library
- mimics Scala collections library
- run time code lifting: query -> AST

Code lifting during compilation



Ordinary Scala collections

List constructor

Anonymous function
Inferred type: `Int => Int`

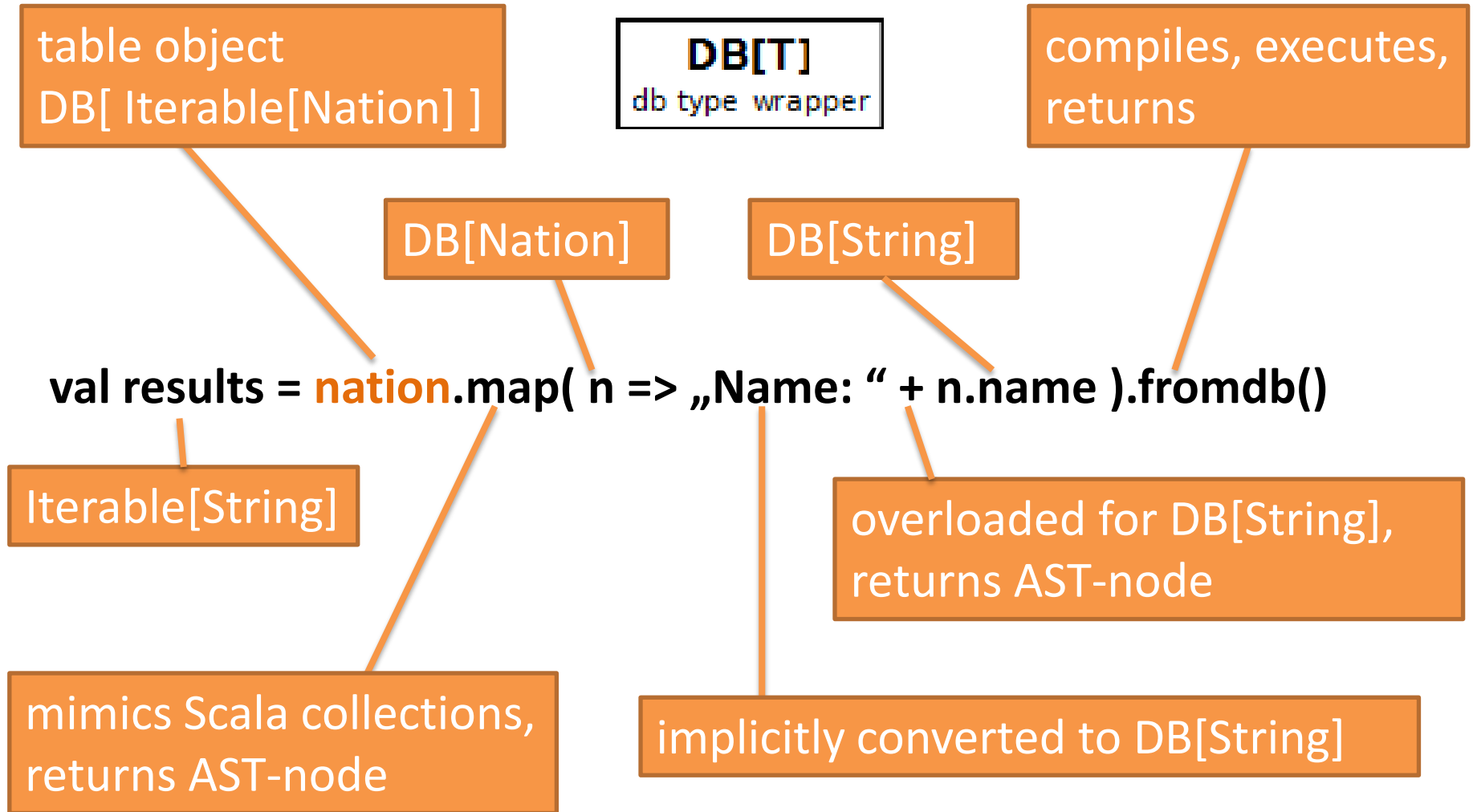
`List(1,2,3).map(x => x*x)`

higher-order
method

applied on every argument

Result: `List(1,4,9)`

Code lifting



Agenda

1. Scala
2. Scala to SQL compilation
(Avalanche safe)
3. Implementation
4. Type safety of queries
(backend specific)
5. Summary

TYPE SAFETY

(BACKEND SPECIFIC)

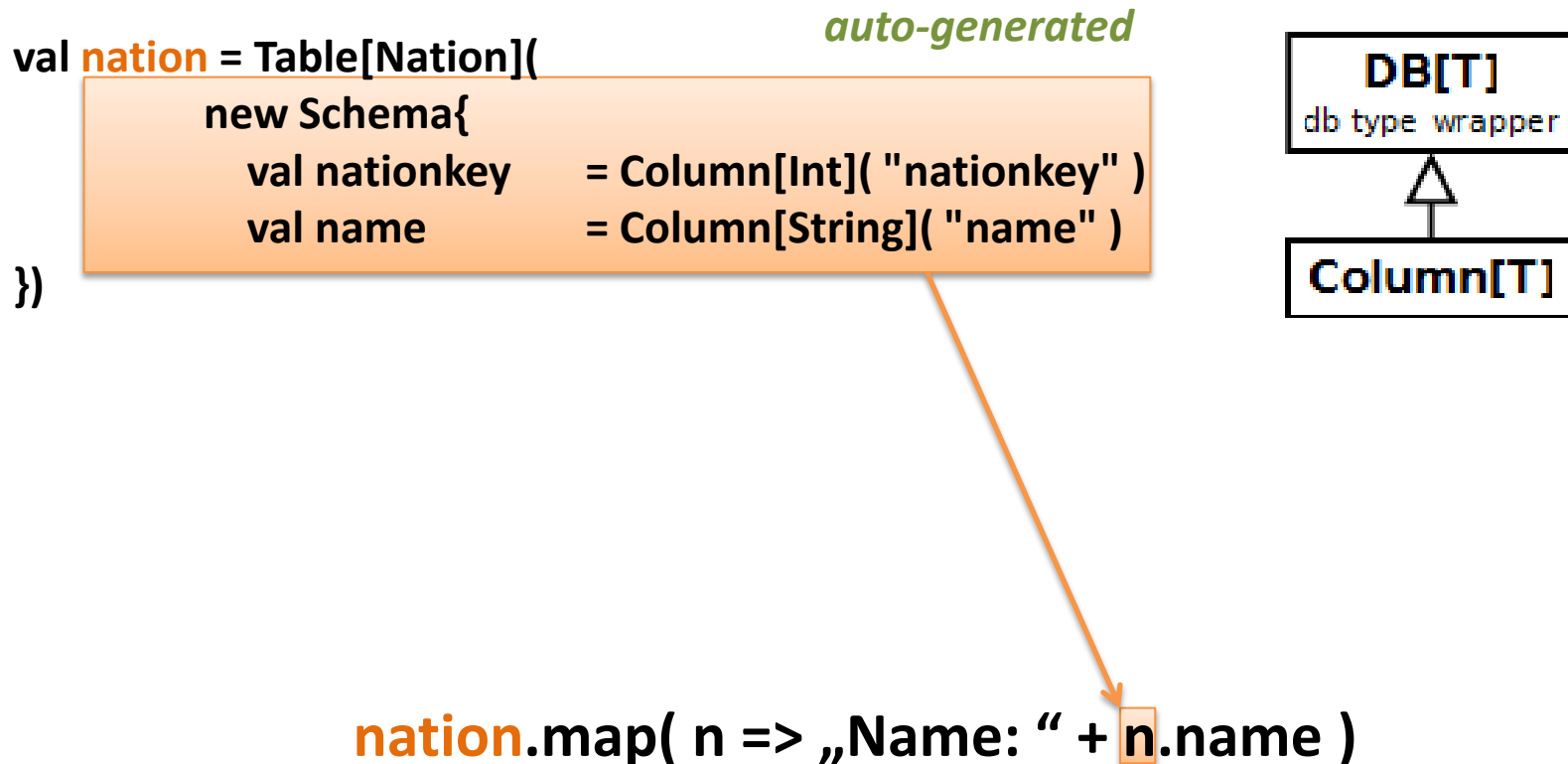
Allowed types and methods



DB[T]
db type wrapper

- type wrapper DB[T]
- SIQ methods defined for wrapped types
- restrictions using type constraints

Type safe schema integration



Type safety example ==

```
type class ValidType = {Int, String, Boolean}
```

pseudo code
real code not shown here

```
trait StrongEqual {  
  def infix_==[T <% ValidType]( a: DB[T], b: DB[T] ): DB[Boolean]  
}
```

“View bound”
type constraint
T in ValidType

Backend specific type safety

- SIQ divided into modules for ==, arithmetic, ...
- combined using mix-in composition
- alternative modules for different SQL dialects
- based on Lightweight Modular Staging framework

SIQ Summary

- Scala-like queries -> SQL
- Avalanche safe
- Type safe

THANK YOU