EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

IBM

Explaining Missing Answers to SPJUA Queries

Melanie Herschel, Universität Tübingen, Germany
Mauricio A. Hernández, IBM Research - Almaden, USA

Conference on Very Large Databases, Singapore
September 14, 2010

VLDB
SINGAPORE 2010

Explaining Missing Data

U1_Email	Friend	U2_Email
john@univ.edu	Jane	jane@busy.com
jane@busy.com	John	john@univ.edu
jane@busy.com	Peter	peter@home.de
peter@home.de	Jane	jane@busy.com
john@univ.edu	$name	$email

U1_Email	Picture	PContributor
peter@home.de	pier39.jpg	John
john@univ.edu	$pic	$name

Query2
Pictures users are interested in

No user with that name?

No interests for user?

No pictures visible to user? ...

Query1
Pairs of connected users

Friend

UID1	UID2
U1	U2
U2	U3

User

UID	Email	Name
U1	john@univ.edu	John
U2	jane@busy.com	Jane
U3	peter@home.de	Peter

UserInterest

IID	UID
I1	U1
I4	U1
I2	U2
I3	U3
I4	U3

Picture

PID	UID	Picture	Visibility
P1	U1	goldengate.jpg	Friend
P2	U1	pier39.jpg	Public
P3	U2	market.jpg	Friend
P4	U3	winetasting.jpg	Public

PictureTag

PTID	PID	Category
PT1	P1	I3
PT2	P1	I1
PT3	P2	I4
PT4	P4	

Explaining Missing Answers

Why is some data not in the result of a query *Q*?

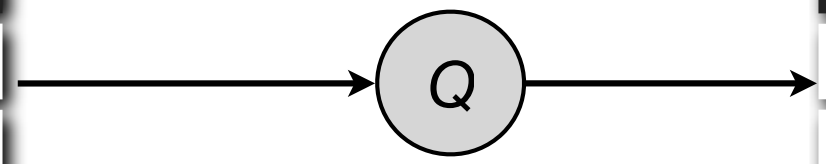
S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$$\Pi_{AC}(S \bowtie_B T)$$



A	C
a	c
a‘	c

Why is (a‘, c‘) not in the output?

Explaining Missing Answers

Why is some data not in the result of a query *Q*?

S

A	B
a	b
a‘	b
a‘	b‘
a‘	$x

T

B	C
b	c
b‘	c‘
b	c‘
$x	c‘

$$\Pi_{AC}(S \bowtie_B T)$$

Q

A	C
a	c
a‘	c
a‘	c‘

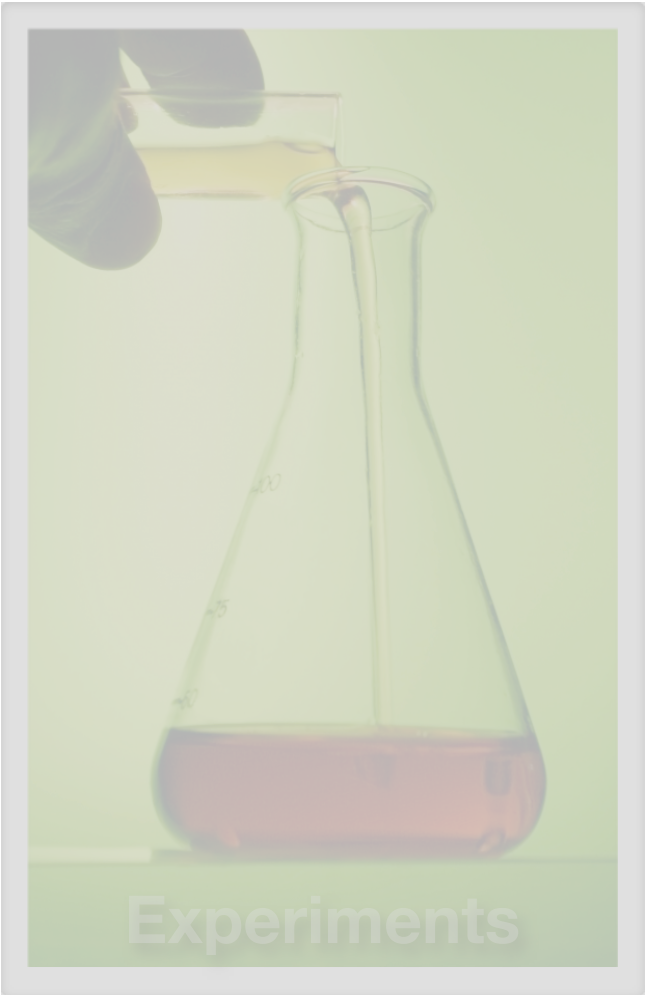
Instance-based explanations
Missing-Answers [Huang08],
Artemis

Query-based explanations
Why-Not [Chapman09],
ConQuer [Tran10]

Contributions

- **Artemis** algorithm
  - Explains a **set** of missing tuples over a **set** of queries that involve selection, projection, join, union, aggregation, and grouping (SPJUA).
  - Considers **side-effects**.
  - Guarantees on completeness and **correctness using a constraint solver**.
- **Framework** for instance-based explanation generation
- Comparative experimental **evaluation**

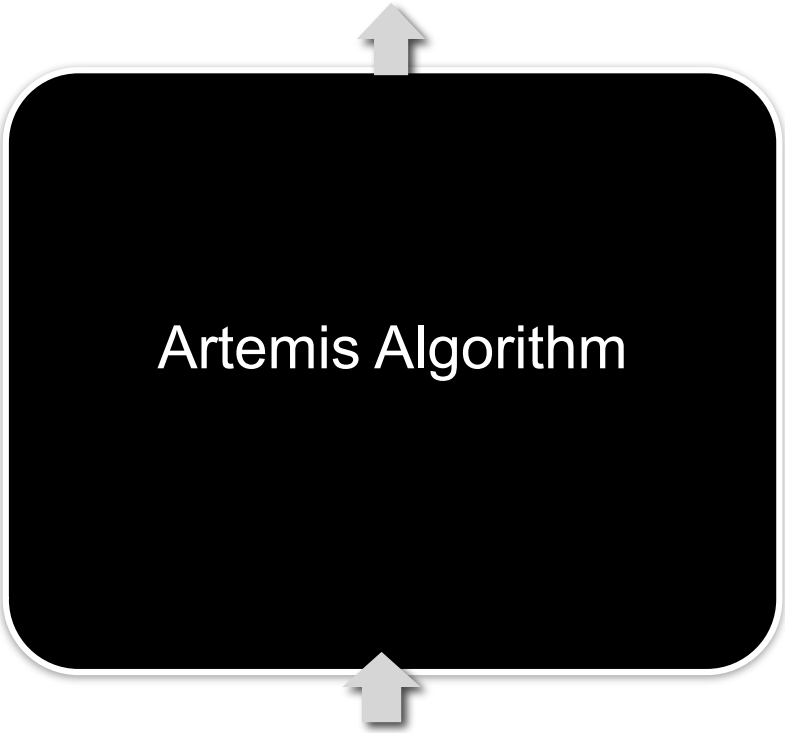
Agenda



The Artemis Algorithm

For SPJU Queries

Set of explanations ***X***

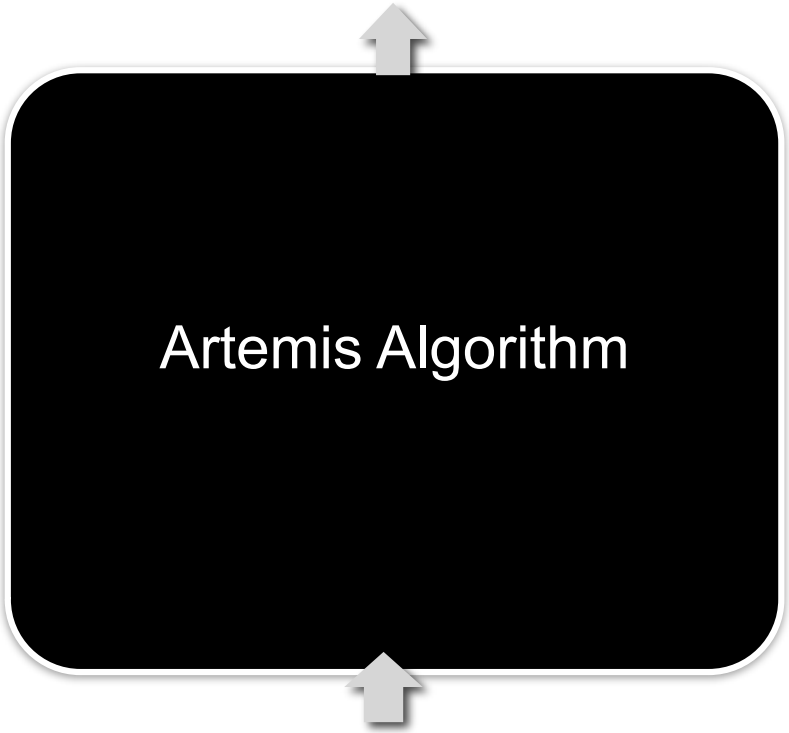


1) Source database ***D***
2) A set of SPJU queries ***Q***
3) A set of missing tuples ***E***
4) Further constraints

The Artemis Algorithm

For SPJU Queries

Set of explanations ***X***



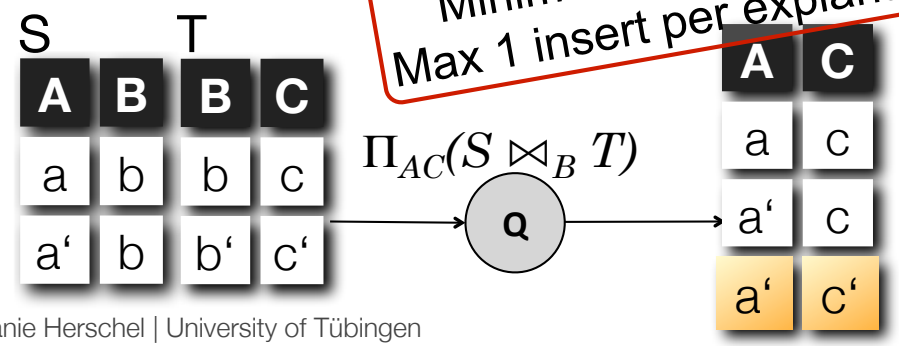
1) Source database ***D***
2) A set of SPJU queries ***Q***
3) A set of missing tuples ***E***
4) Further constraints

Explanation 1

S		T	
a‘	b‘	b’	c’

Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

Minimum #side-effects,
Max 1 insert per explanation



The Artemis Algorithm

For SPJU Queries

Set of explanations ***X***

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic wittness

1) Source database ***D***
2) A set of SPJU queries ***Q***
3) A set of missing tuples ***E***
4) Further constraints

Explanation 1

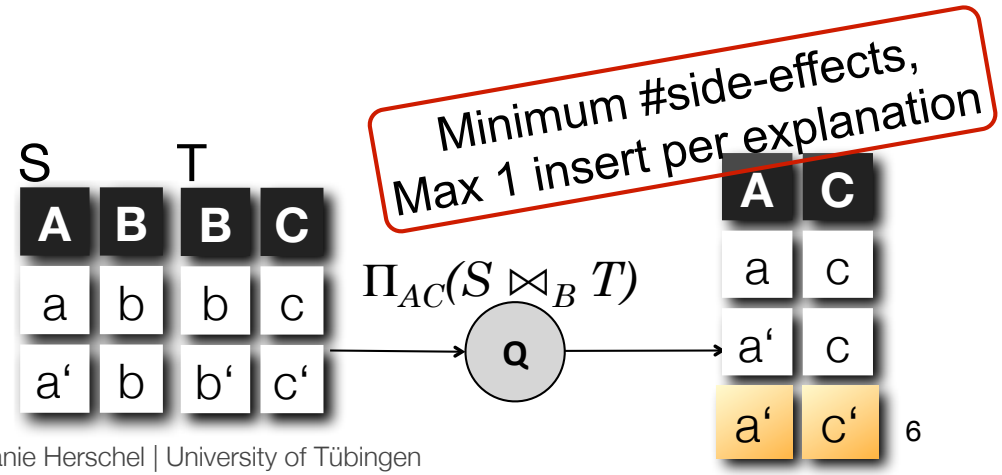
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

S

A	B
a‘	b‘

T

B	C
b’	c’



The Artemis Algorithm

For SPJU Queries

S: a‘ b‘ | T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic witness

Minimum #side-effects,
Max 1 insert per explanation

S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

$E = \{(a‘, c‘)\}$

Q: V(a’, c’) :- R(a’, v_b), S(v_b, c’)

The Artemis Algorithm

For SPJU Queries

S: a‘ b‘ | T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic witness

Minimum #side-effects,
Max 1 insert per explanation

S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

Generic Witness:
R(a’, $x), S($x, c’)

The Artemis Algorithm

For SPJU Queries

S: a‘ b‘ | T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for *D*

(1) Compute generic wittness

Minimum #side-effects,
Max 1 insert per explanation

S^C

A	B	con
a	b	TRUE
a‘	b	TRUE
a’	$x1	$x1 ≠ b

T^C

B	C	con
b	c	TRUE
b‘	c‘	TRUE
$x2	c’	$x2 ≠ b’

S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

Generic Witness:
R(a’, $x), S($x, c’)

The Artemis Algorithm

For SPJU Queries

S		T	
a‘	b‘	b’	c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of *Q*

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic wittness

Minimum #side-effects,
Max 1 insert per explanation

S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

A	C	con
a	c	TRUE
a‘	c	TRUE
a	c‘	$x2 = b ∧ $x2 ≠ b‘
a‘	c‘	$x2 = b ∧ $x2 ≠ b‘
a‘	c‘	$x1 = b‘∧ $x1 ≠ b
a‘	c‘	$x1 = $x2 ∧ $x2 ≠ b‘ ∧ $x1 ≠ b

$\Pi_{AC}(S \bowtie_B T)$

Q

S^C

A	B	con
a	b	TRUE
a‘	b	TRUE
a’	$x1	$x1 ≠ b

T^C

B	C	con
b	c	TRUE
b‘	c‘	TRUE
$x2	c’	$x2 ≠ b’

The Artemis Algorithm

For SPJU Queries

S: a‘ b‘ | T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic wittness

Minimum #side-effects,
Max 1 insert per explanation

S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

side-effect

matches of (a’,c’)

Constraint Satisfaction Problem

tuple (a’,c’) exists

AND

minimum number of side-effects

A	C	con
a	c	TRUE
a‘	c	TRUE
a	c‘	$x2 = b ∧ $x2 ≠ b‘
a‘	c‘	$x2 = b ∧ $x2 ≠ b‘
a‘	c‘	$x1 = b‘∧ $x1 ≠ b
a‘	c‘	$x1 = $x2 ∧ $x2 ≠ b‘ ∧ $x1 ≠ b

The Artemis Algorithm

For SPJU Queries

S		T	
a‘	b‘	b’	c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic wittness

Minimum #side-effects,
Max 1 insert per explanation

S		T	
A	B	B	C
a	b	b	c
a‘	b	b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

side-effect

matches of (a’,c’)

Constraint Satisfaction Problem

$x2 = b ∧ $x2 ≠ b‘

AND

$x2 = b ∧ $x2 ≠ b‘
$x1 = b‘∧ $x1 ≠ b
$x1 = $x2 ∧ $x2 ≠ b‘ ∧ $x1 ≠ b

A	C	con
a	c	TRUE
a‘	c	TRUE
a	c‘	$x2 = b ∧ $x2 ≠ b‘
a‘	c‘	$x2 = b ∧ $x2 ≠ b‘
a‘	c‘	$x1 = b‘∧ $x1 ≠ b
a‘	c‘	$x1 = $x2 ∧ $x2 ≠ b‘ ∧ $x1 ≠ b

The Artemis Algorithm

For SPJU Queries

S: a‘ b‘ T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic wittness

Minimum #side-effects,
Max 1 insert per explanation

S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

side-effect

matches of (a’,c’)

Output of constraint solver for...

1st match: $x2 = b,
1 side-effect

2nd match: $x1 = b’,
0 side-effects

3rd match: $x1 = $x2, $x1 ≠ b, $x2 ≠ b’,
0 side-effects

A	C	con
a	c	TRUE
a‘	c	TRUE
a	c‘	$x2 = b ∧ $x2 ≠ b‘
a‘	c‘	$x2 = b ∧ $x2 ≠ b‘
a‘	c‘	$x1 = b‘∧ $x1 ≠ b
a‘	c‘	$x1 = $x2 ∧ $x2 ≠ b‘ ∧ $x1 ≠ b

The Artemis Algorithm

For SPJU Queries

S: a‘ b‘ | T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic wittness

Minimum #side-effects,
Max 1 insert per explanation

S: a‘ b‘ | T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

S: a‘ $x1 | T: $x2 c’

Explanation 2
Insert (a’,$x1) into *S* and ($x2, c’) into T s.t. $x1 = $x2 and new tuples are no duplicates

S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

Output of constraint solver for...

~~**1st match**: $x2 = b, 1 side-effect~~

Too many side-effects

2nd match: $x1 = b’, 0 side-effects

3rd match: $x1 = $x2, $x1 ≠ b, $x2 ≠ b’, 0 side-effects

The Artemis Algorithm

For SPJU Queries

S: a‘ b‘ | T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

S: a‘ b‘ | T: b’ c’

Explanation 1
Insert (a‘, b‘) into *S* s.t. it joins with existing tuple (b‘, c‘) in *T*

(5) Filter and sort explanations

(4) Generate explanations

(3) Compute c-tables of ***Q***

(2) Create conditional tables (c-tables) for ***D***

(1) Compute generic wittness

Minimum #side-effects,
Max 1 insert per explanation

S

A	B
a	b
a‘	b

T

B	C
b	c
b‘	c‘

$\Pi_{AC}(S \bowtie_B T)$

Q

A	C
a	c
a‘	c
a‘	c‘

The Artemis Algorithm

Extension to Aggregation and Grouping

```
SELECT    A,
          COUNT(C) AS C_CNT
FROM      S, T
WHERE     S.B = T.B
GROUP BY  A
```

S

A	B
a	b
a‘	b
a’	b’

T

B	C
b	c
b‘	c‘

→ Q →

A	C_CNT
a	1
a‘	2

- Meet-in-the-middle approach
  - Given set of missing tuples ***E*** on an SPJA view, determine ***E’*** for SPJ view, s.t. applying aggregation and grouping on the SPJ view results in the original SPJA view.
  - Artemis algorithm for SPJ views for ***E’***.
  - Extend returned explanations to include aggregation and grouping.

The Artemis Algorithm

Extension to Aggregation and Grouping

S

A	B
a	b
a‘	b
a’	b’

T

B	C
b	c
b‘	c‘

```
SELECT   A, C
FROM     S, T
WHERE    S.B = T.B
```

Q1

A	C
a	c
a‘	c
a’	c’
b‘	$v

10000 X

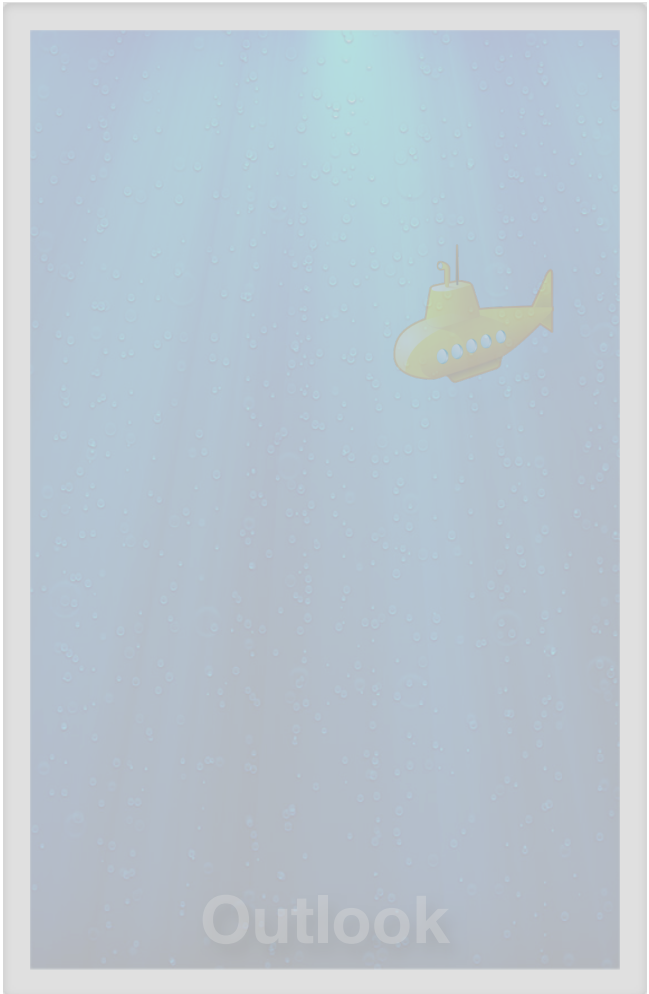
Q2

```
SELECT   A,
         COUNT(C) AS C_CNT
FROM     Q1
GROUP BY A
```

A	C_CNT
a	1
a‘	2
b‘	10000

- Meet-in-the-middle approach
  - Given set of missing tuples ***E*** on an SPJA view, determine ***E’*** for SPJ view, s.t. applying aggregation and grouping on the SPJ view results in the original SPJA view.
  - Artemis algorithm for SPJ views for ***E’***.
  - Extend returned explanations to include aggregation and grouping.

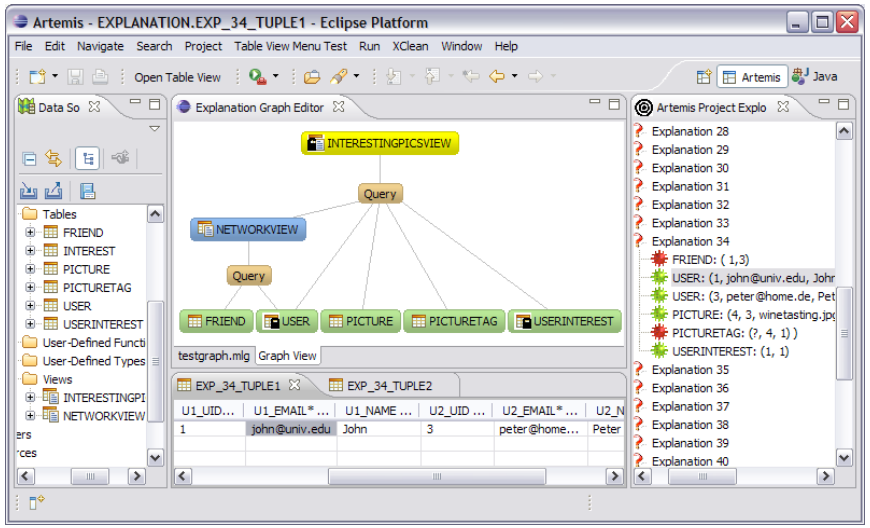
Agenda



Experimental Setup

Implementation

- Eclipse Plugin [VLDB09].
- Artemis and Missing-Answers [VLDB08]
- Minion used as constraint solver for Artemis.
- IBM DB2 9.5 used as RDBMS.

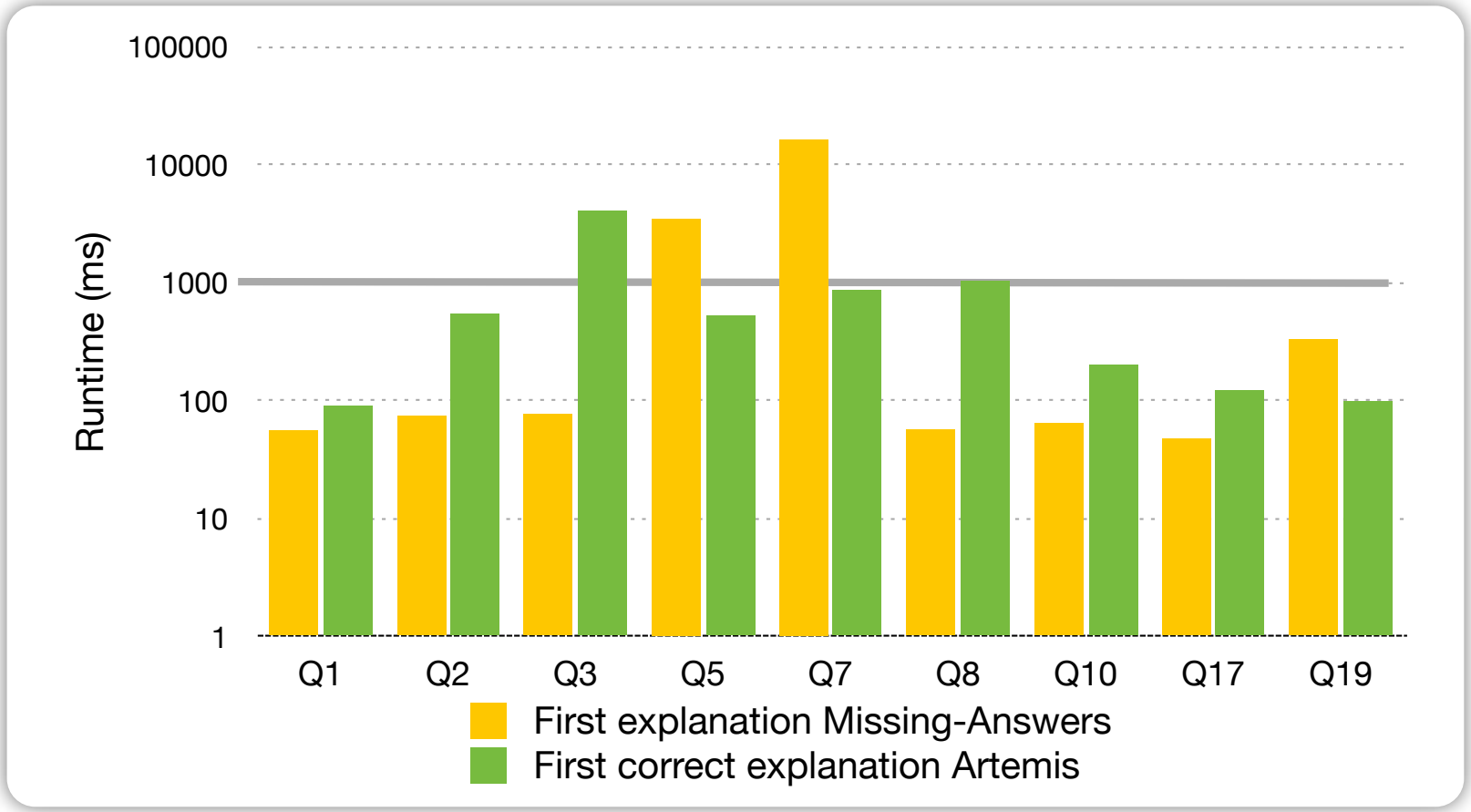


Datasets

- TPCH
  - 10 MB of data
  - 9 queries (adaptations of TPCH queries limited to supported types of queries)
  - No insertions on `Nation` and `Region`.

TPC Transaction Processing Performance Council

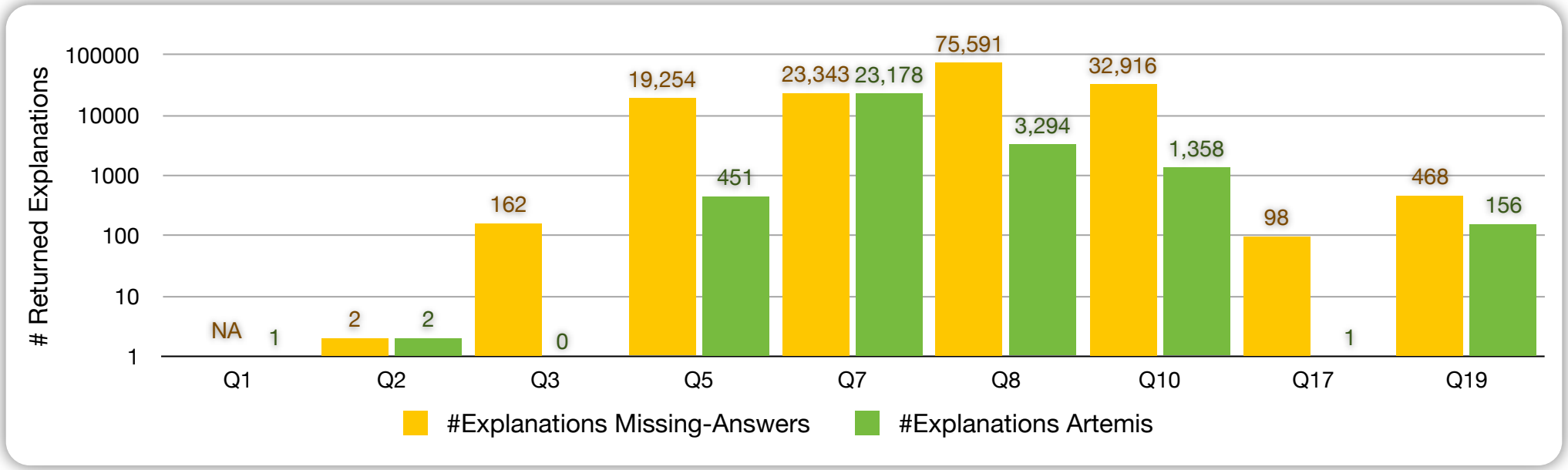
Runtime to First Correct Explanation

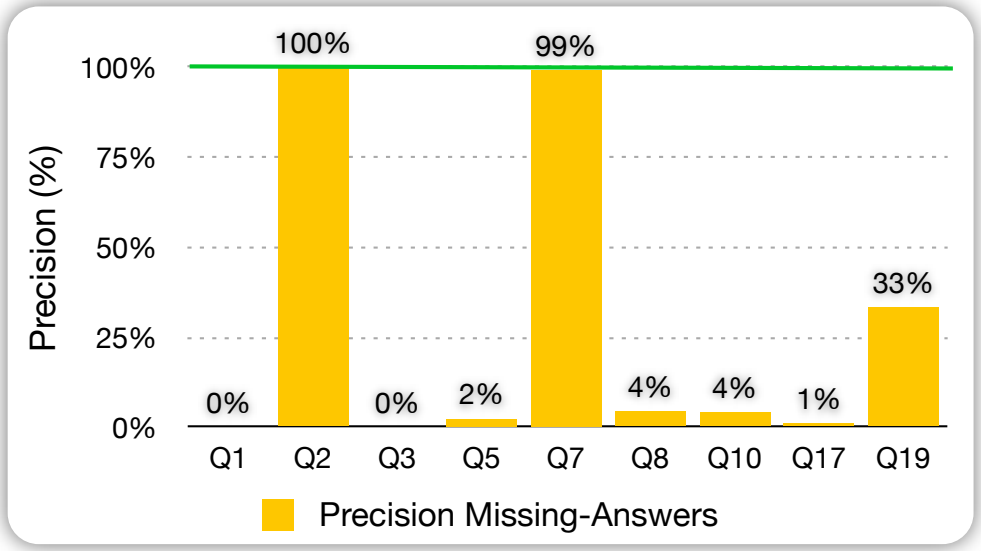


Artemis takes less than a second to find first correct explanation in most cases.

Missing-Answers usually faster, but returned explanation can be wrong.

Effectiveness

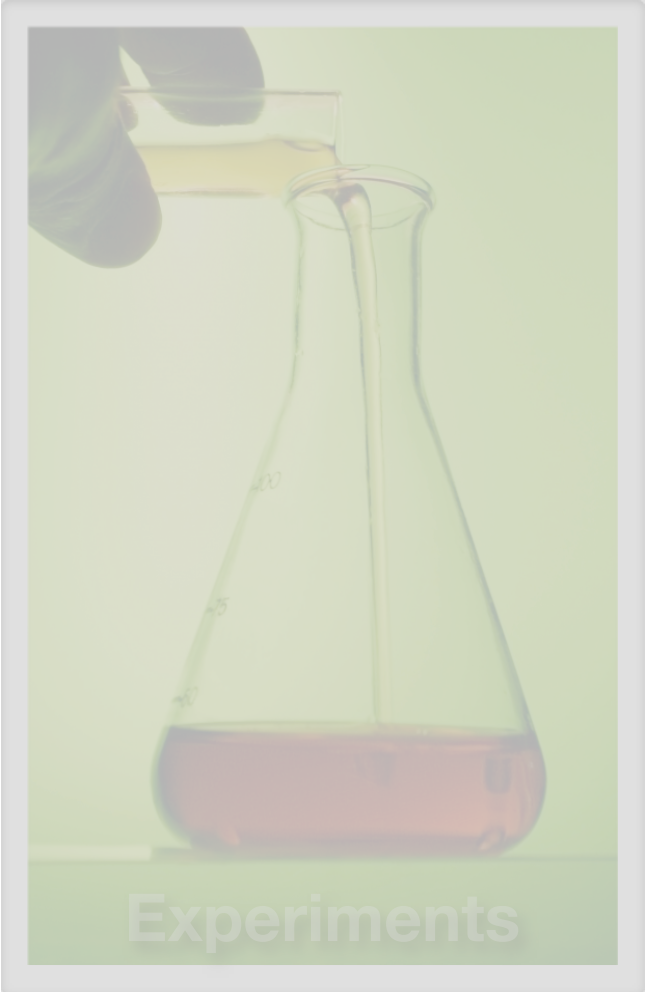


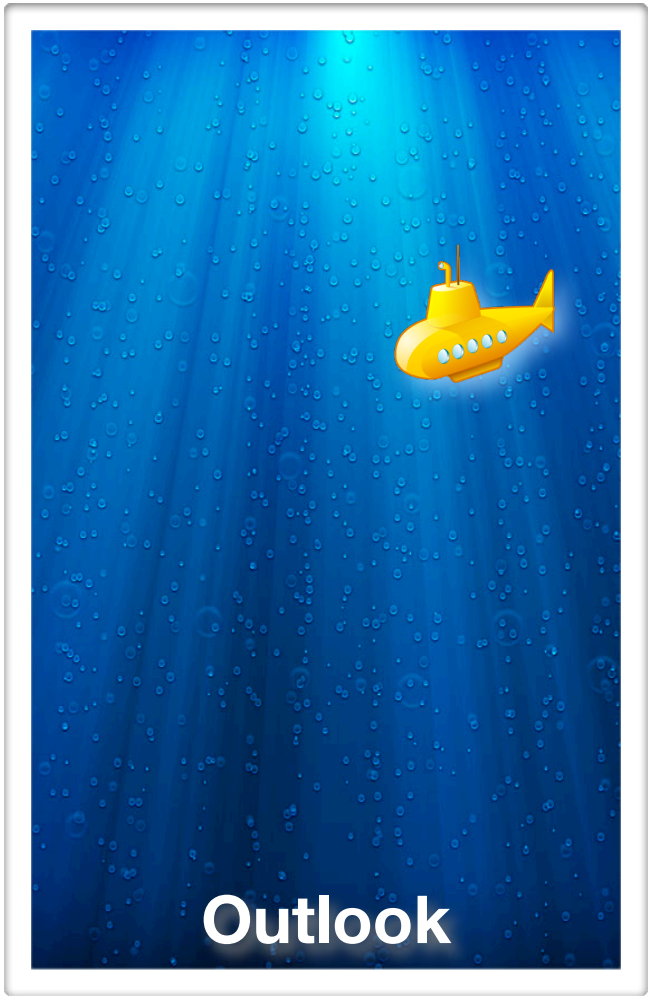


Number of unsatisfiable explanations can be substantial when using Missing-Answers.

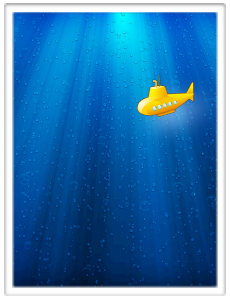
Constraint solver makes Artemis run slower, but effectiveness significantly improved.

Agenda





Outlook



So far...

- **Framework** for instance-based explanation generation.
- **Artemis** algorithm (SPJUA queries, side-effects, correctness)
- Comparative experimental **evaluation**

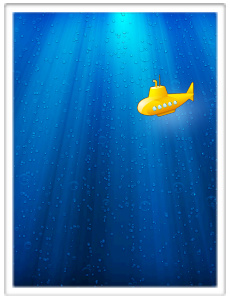
In the future..

- Efficiency improvement
- Visualization to improve usability

The “Big Picture”

- Build a system to analyze, fix, and test data transformations

Outlook



So far...

- **Framework** for instance-based explanation generation.
- **Artemis** algorithm (SPJUA queries, side-effects, correctness)
- Comparative experimental **evaluation**

In the future..

- Efficiency improvement
- Visualization to improve usability

The “Big Picture”

- Build a system to analyze, fix, and test data transformations