

Explanations of Provenance Challenge Queries

Rui Huang

James Michaelis

RPI / TWC

Overview

- Intended to get both RPI/TWC and UTEP on the same page regarding query semantics
- Queries covered in this draft:
 - Core Queries: 1, 2, 3
 - Optional Queries: 1, 2, 3, 5, 6, 7, 8, 10, 11, 12
- Missing queries (not enough information)
 - Optional Queries: 4,9, 13
 - Additional Queries: Core 5, Optional 14, 15
- Other teams' current status

Some important notations

- CSV file – a data file with the extension .csv
- Workflow execution – an execution of the workflow used in the Provenance Challenge
- Operation execution – execution of a process in the workflow
- Halt – denotes the end of a workflow execution

Core Query 1

Original Query

For a given **detection**, which CSV files **contributed** to it?

Meaning of Query

- **Detection** – (artifact) A table row created in the database
- **contributed** – Detections are defined in certain CSV files, and in turn loaded into the database

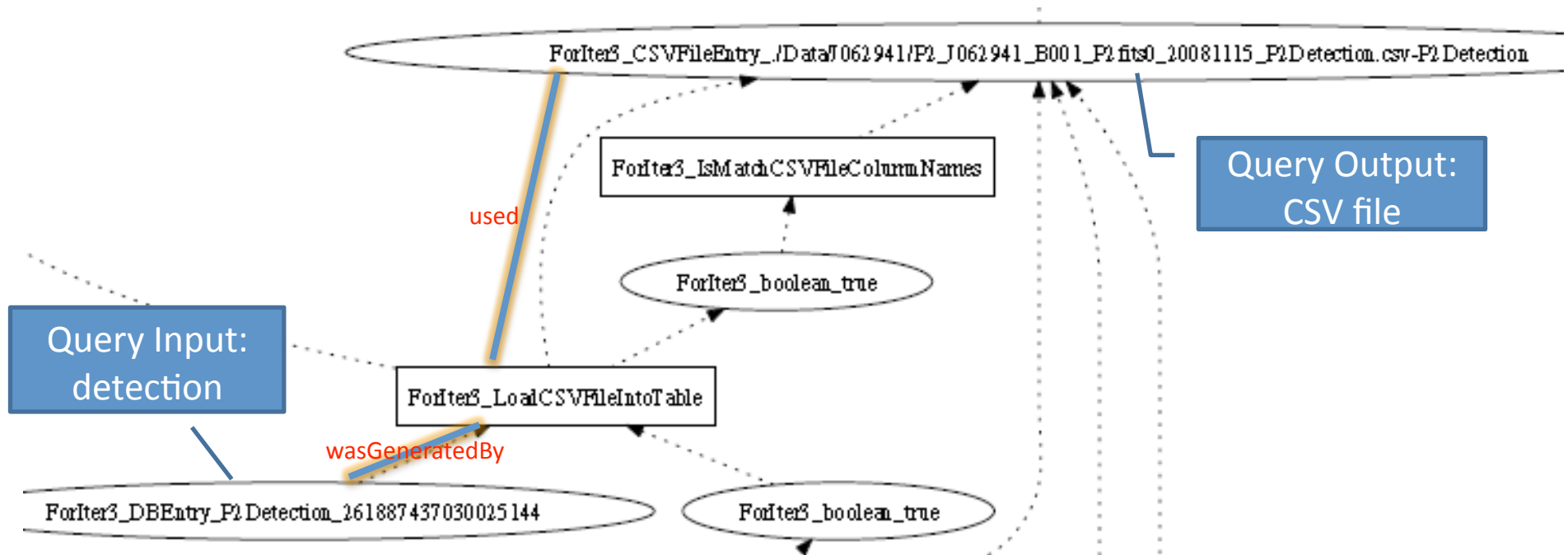
Query Input

- A detection

Expected Query output

- A list of one or more CSV files

Core Query 1 (answer)



More Details

Process “LoadCSVFileIntoTable” generates detection entry using contributing parameter “CSV File”

Core Query 2

Original Query

For a given CSV File, was the range check (process IsMatchTableColumnRanges) **performed for** its table in the SQL database?

Meaning of Query

- **performed for**– checks whether the process IsMatchTableColumnRanges was run on the database table corresponding to CSV File

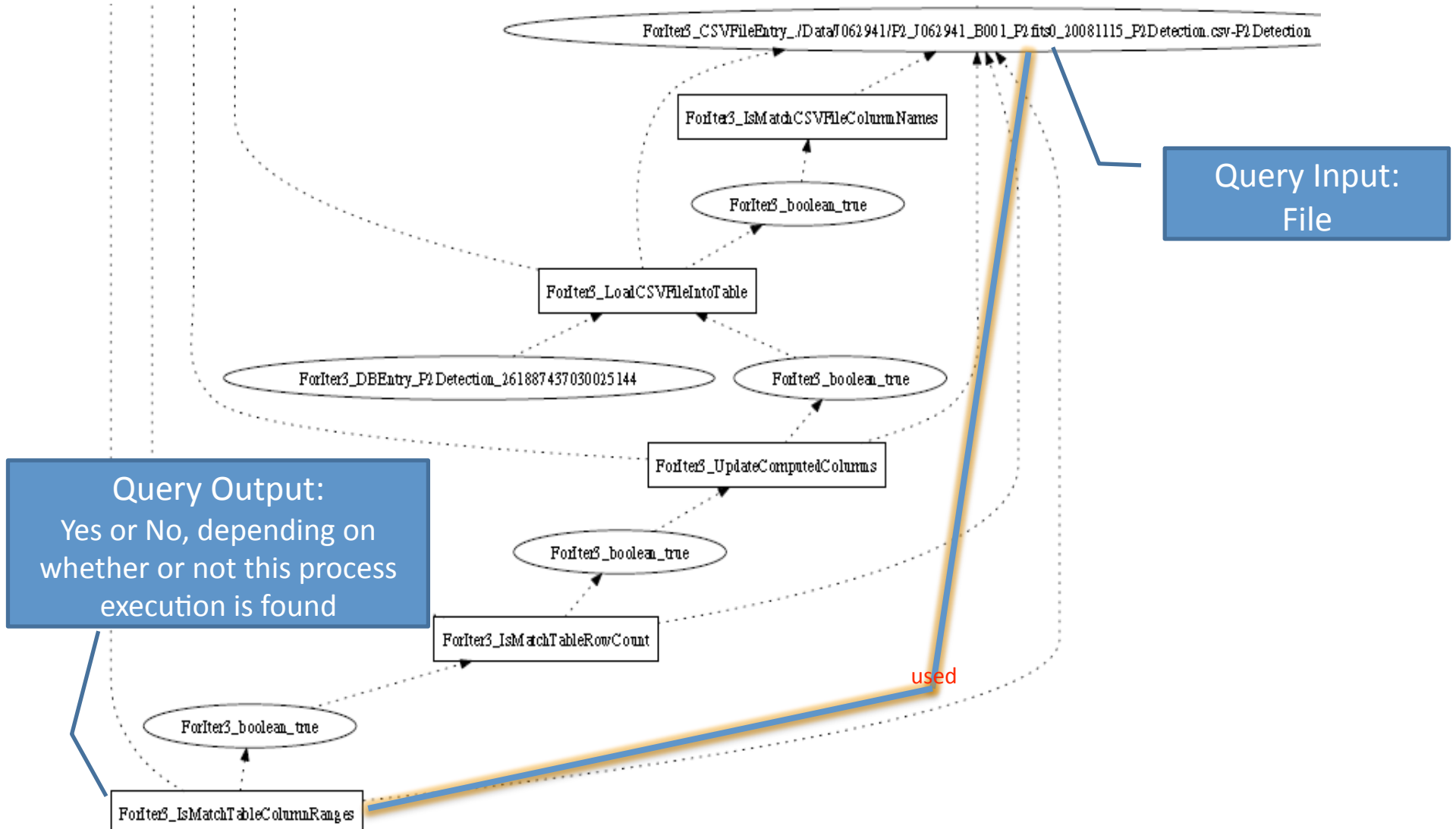
Query Input

- CSV File

Expected Query output

- Yes: If an execution of IsMatchTableColumnRanges(no matter what output it returns) is found which takes CSV File as input
- No: otherwise

Core Query 2 (answer)



Core Query 3

Original Query

Which **operation executions** were **necessary** for the **Image table** to contain a particular (non-computed) **value**?

Meaning of Query

- **Image table** – (artifact) an table in the database
- **value** – (artifact) a section(cell, row, etc) in the **Image table**
- **operation executions** – *see slide 3*
- **necessary** – processes that generates “real” data, rather than control flow data(true/false) or nothing

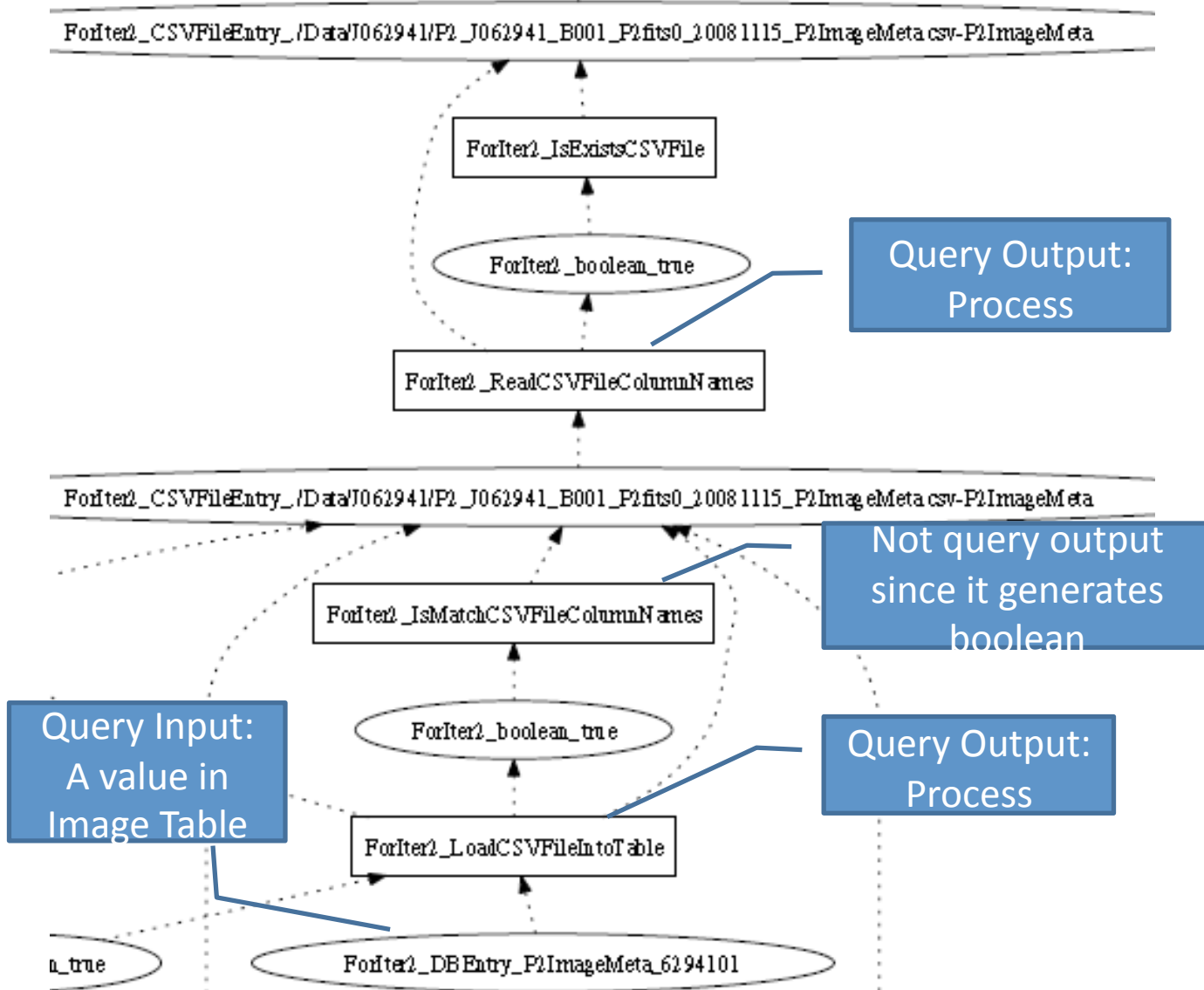
Query Input

- Particular **value** in the **Image Table**

Expected Query output

- **operation executions**

Core Query 3 (answer)



More Details

Currently, our team runs a recursive routine to obtain the list of necessary processes. Starting from the execution of `LoadCSVFileIntoTable`, each of the functions which generated one of its input parameters is added. In turn the functions which generated their input parameters are added, and so on. Functions that generates boolean value is not included here.

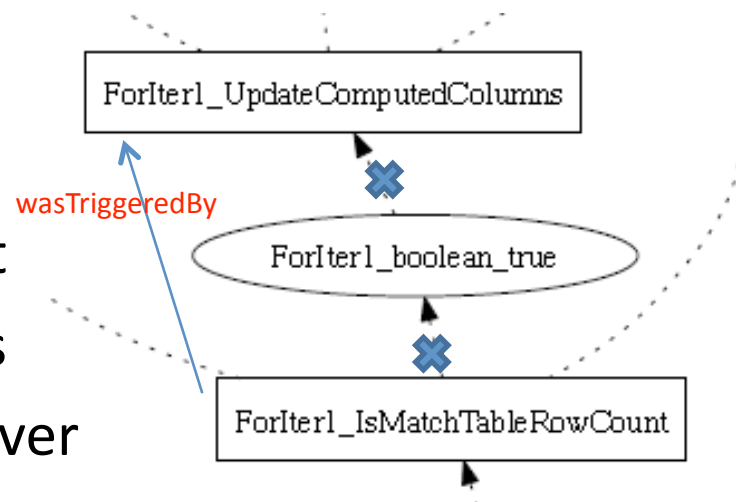
Core Query 3 (answer)

Currently, our team are also considering using “wasTriggeredBy” to represent the control flow .

For example, in the left graph,

We generates “wasTriggeredBy” between two processes.

Then in the above query, we don't need to filter control flow, process “UpdatedComputerdColumns” never generates data.



Note: This could also be used to represent step dependency.

Optional Query 1

Original Query

The workflow halts due to failing an `IsMatchTableColumnRanges` check. How many **tables successfully loaded** before the workflow halted due to a failed check?

Meaning of Query

- **tables** – (artifact) list of tables created in the database
- `IsMatchTableColumnRanges` – (process)
- **successfully loaded** – indicates a table in the database was successfully checked with the “`IsMatchTableColumnRanges`” function

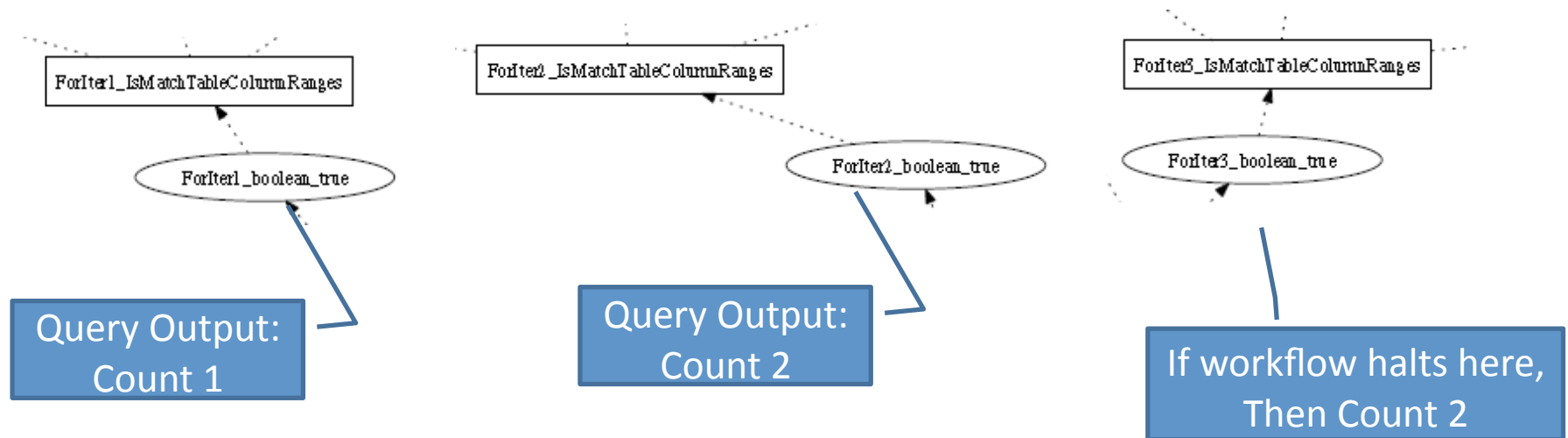
Query Input

- Process “`IsMatchTableColumnRanges`”

Expected Query output

- Number of **tables** loaded into database

Optional Query 1 (answer)



More details

Process “IsMatchTableColumnRanges” generates a boolean result to confirm that a table was checked. If this boolean value is true, then the table was successfully loaded. Could return the number of boolean values evaluating to true.

Another approach

Return the number of all the tables in the databases – 1.

Since tables will be loaded after process “LoadCSVFileIntoTable” and one of them will not pass “IsMatchTableColumnRanges”

Optional Query 2

Original Query

Which **pairs of procedures** in the workflow could be **swapped** and the same result still be obtained?

Meaning of Query

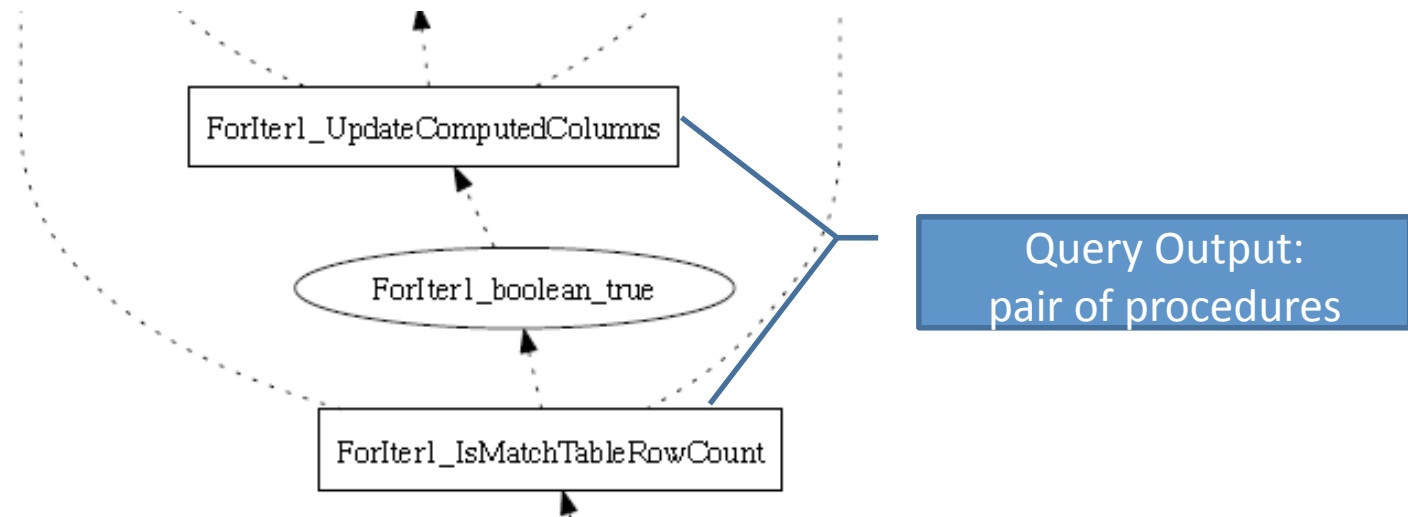
- **pairs of procedures** – (artifact) two processes
- **swapped** – change the order of two procedures and the output is the same

Query Input

Expected Query output

- **pairs of procedures**

Optional Query 2 (answer)



More details

Two processes that take same inputs(boolean not included) and only have step dependency between the two could be swapped.

For example, both “UpdateComputedColumns” and “IsmatchTableRowCount” takes three inputs “File Entry”, “Database Entry” and boolean check(ignore).

Note: If use “wasTriggeredBy” to represent control flow, boolean check will not become an input.

Optional Query 3

Original Query

How much time **expired** between a successful **IsMatchCSVFileTables** test and an unsuccessful **IsExistsCSVFiles** test?

Meaning of Query

- **IsMatchCSVFileTables** – (process)
- **IsExistsCSVFiles** – (process)
- **expired** – time interval between the execution of two processes

Query Input

- **IsMatchCSVFileTables**
- **IsExistsCSVFiles**

Expected Query output

- **time interval**

Optional Query 5

Original Query

A user executes the **workflow** many times over different **sets of data**. He wants to determine, which of the **execution halted**?

Meaning of Query

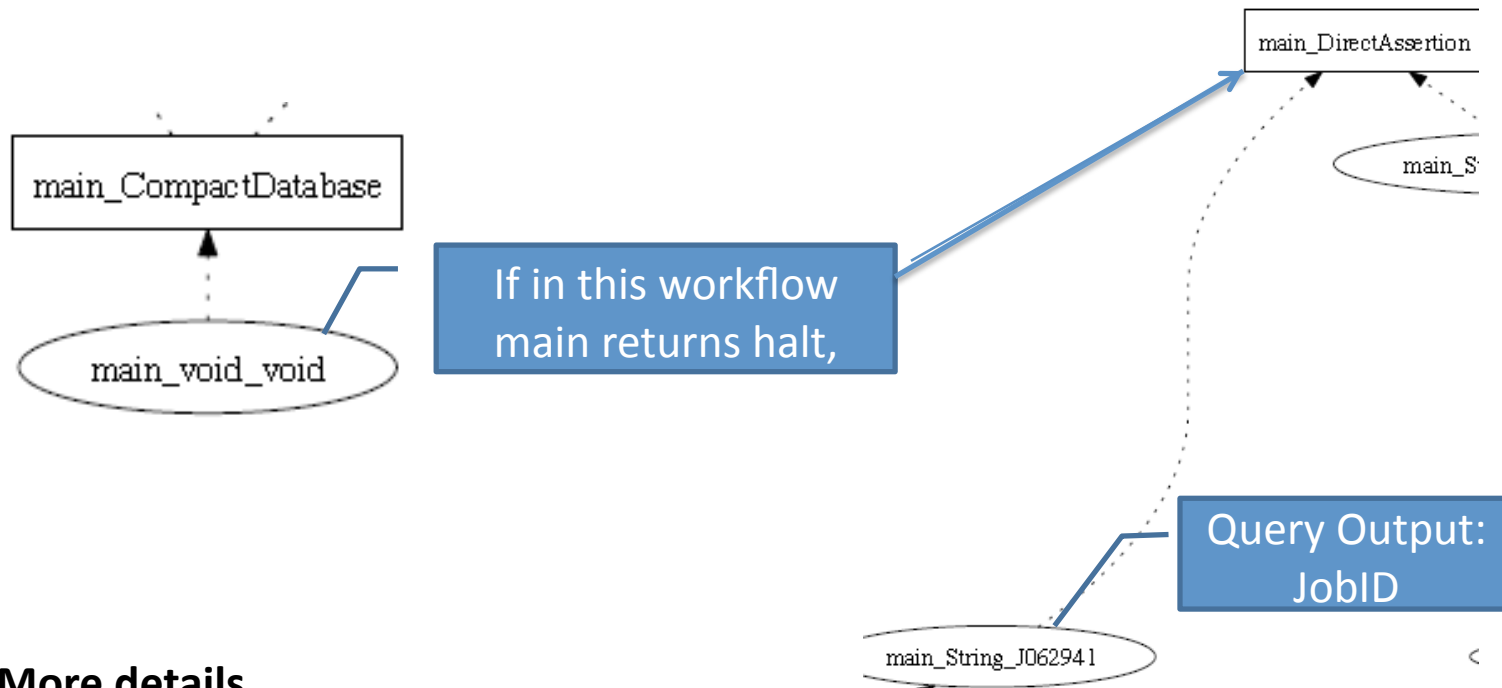
- **workflow execution**
- **set of data** – one input(**JobID**) provided by user
- **halt** – *see slide 3*

Query Input

Expected Query output

- **JobID**

Optional Query 5 (answer)



More details

First find which main method return halts

Next, similar to optional query 10, find the user input - data set

Optional Query 6

Original Query

Determine the **step where halt occurred?**

Meaning of Query

- **step** – (process)
- **where halt occurred** – (artifact) point where workflow stopped, due to a control flow boolean evaluating to false

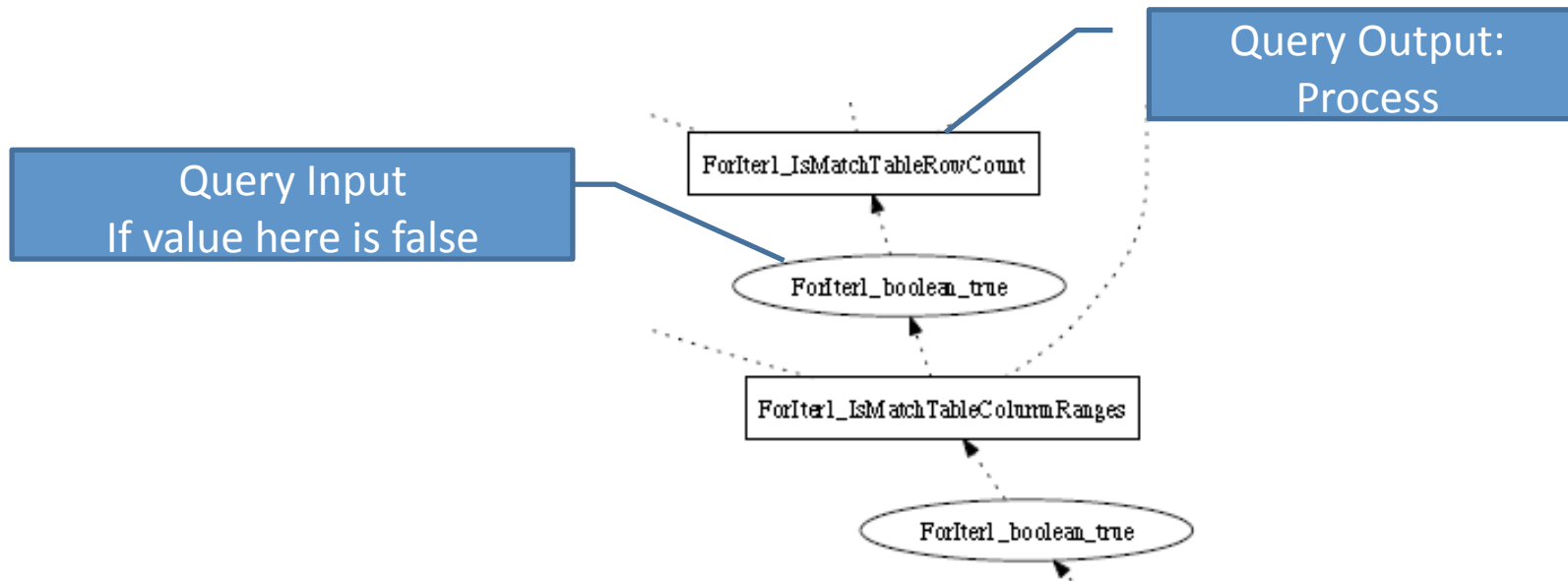
Query Input

- Control flow boolean (evaluating to false)

Expected Query output

- Process which generated the control check boolean

Optional Query 6 (answer)



More Details

Certain workflow processes will return boolean values indicating success or failure. In turn, these boolean values are used in workflow control flow checks, which halt the workflow when a value of false is encountered.

Optional Query 7

Original Query

Determine **data** and associated **granularities** of the data being **processed**, when **halt** occurred?

Meaning of Query

- **data** – (artifact) information being processed by the workflow
- **granularities** – indicates specific sections of a piece of data being processed.
- **processed** – indicates artifact use/manipulation by processes
- **halt** - *see slide 3*

Query Input

- Control flow boolean (evaluating to false)

Expected Query output

- A list of **data**, with associated **granularities**

Optional Query 8

Original Query

Which **steps** were **completed successfully** before the **halt** occurred?

Meaning of Query

- **steps** – (process)
- **halt**
- **completed successfully** – processes which finish operation execution prior to the **halt**

Query Input

- Control flow boolean (evaluating to false)

Expected Query output

- A list of processes

Optional Query 8 (answer)

- One approach: Similar to query 3, recursively call the query to the artifact which is used by the last query output, starting from the place where the halt occurs. In this case, the processes which generate boolean result are also kept.
- Another approach: find a list of processes which were recorded in the ProtoProv RDF data

Optional Query 10

Original Query

For a **workflow execution**, determine the **user inputs**?

Meaning of Query

- **user input** – (artifact) input provided by user
- **for** – track down in the particular workflow
- **workflow execution**

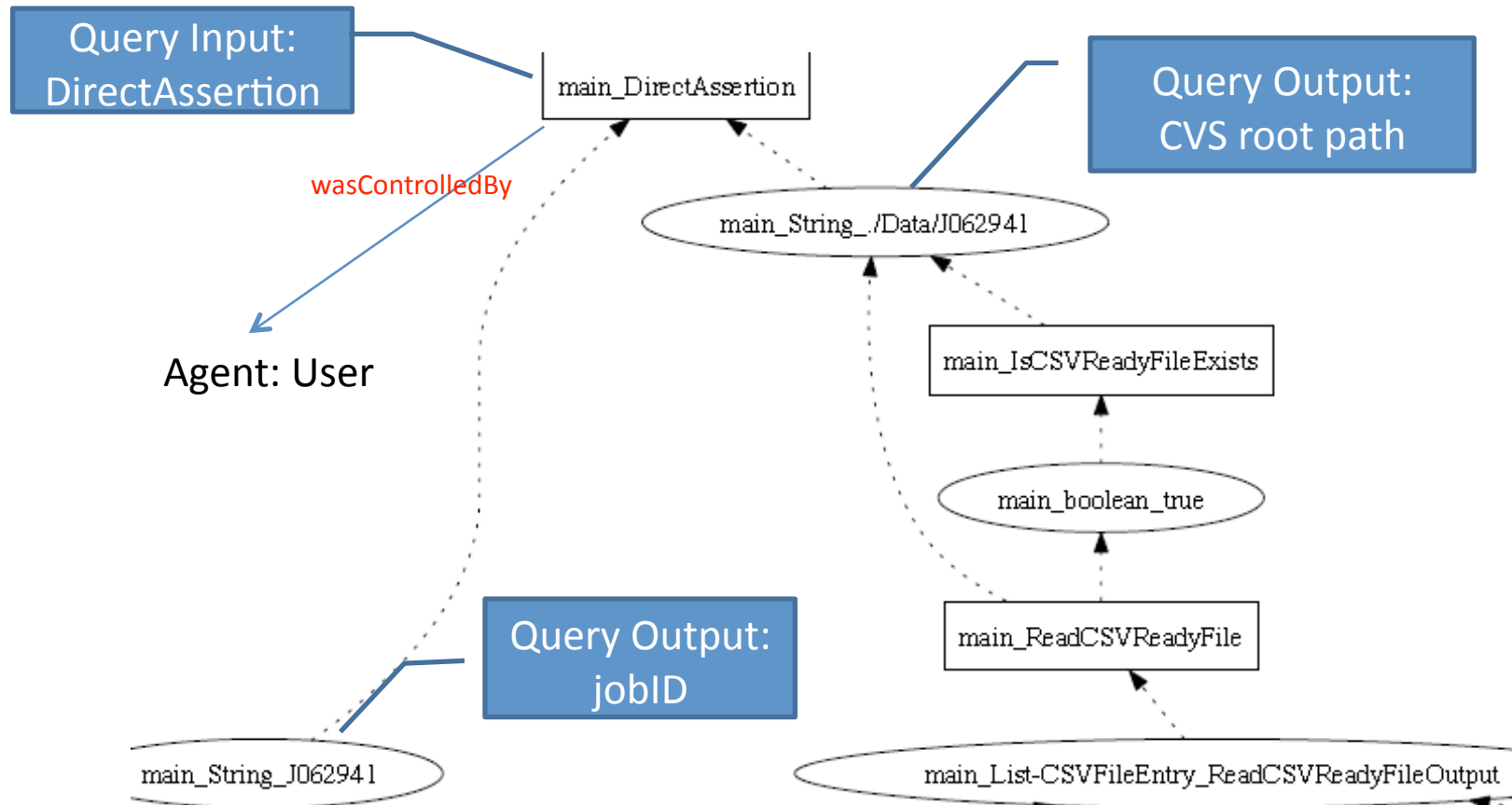
Query Input

- **workflow execution**

Expected Query output

- a list of artifacts, corresponding to the **user inputs**

Optional Query 10 (our OPM data)



More Details

Process “DirectAssertion” generates all the user inputs

Optional Query 11

Original Query

Determine **steps** that **require** the **user inputs** for a **workflow execution**.

Meaning of Query

- **user input** – (artifact) input provided by user
- **require** – process need some inputs to execute
- **step** – (processes)
- **workflow execution** – *see slide 3*

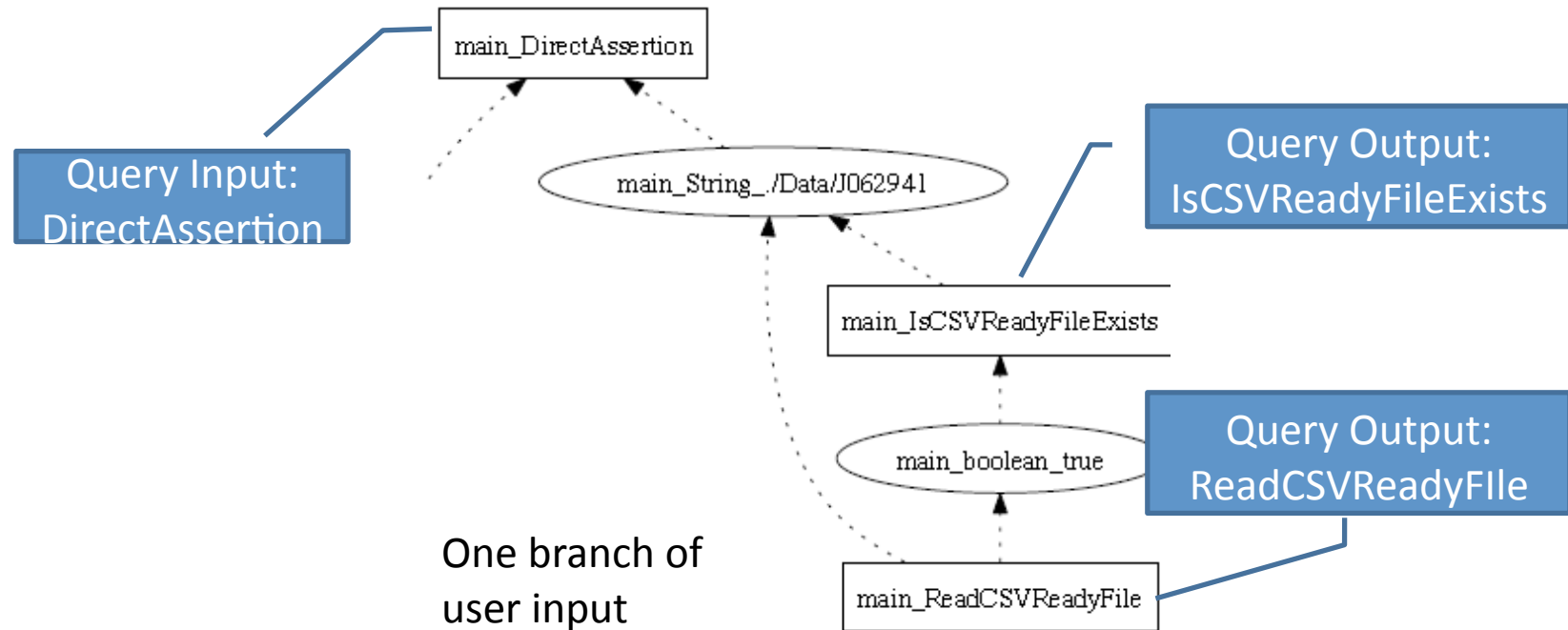
Query Input

- **workflow execution**

Expected Query output

- a list of processes

Optional Query 11 (our OPM data)



More Details

From optional query 10, we know how to find all the user inputs for a workflow execution. Next, all the processes which use one of those user inputs will be returned as output from this query.

Note: here only consider directly require. If problem asks for “Indirectly require”, need recursion.

Optional Query 12

Original Query

For a **workflow execution** that **halted**, which **files** were **processed successfully**?

Meaning of Query

- **file** – (artifact) .csv file
- **halt** – *see slide 3*
- **processed successfully** – indicates files which were passed process “IsMatchTableColumnRanges”, and in turn the corresponding control flow boolean evaluated to true
- **workflow execution** – *see slide 3*

Query Input

- **workflow execution**

Expected Output

- a list of .csv files

Optional Query 12 (our OPM data)

- Similar to optional query 1.
- Returns a list of data files (loaded from .csv files) which are passed to the process “IsMatchTableColumnRanges” and yield control flow booleans evaluating to true.

Optional Query 13

Original Query

For a **workflow execution**, display the following provenance views: data dependency views, step dependency views?

Meaning of Query

- data dependency – input(not control flow) of one process is the output of another process
- step dependency – input(control flow) of one process is the output of another process
- **workflow execution** – *see slide 3*

Query Input

- **workflow execution**

Expected Output

- processes that are data dependency
- processes that are step dependency

Optional Query 13 (our OPM data)

Note: if use “wasTriggeredBy” to represent the control flow, step dependency will be clearer and don’t need to differentiate types of data input (boolean or “real” data)

Other teams' current status

Team	Query answered
UC Davis	Core 1,2,3 Optional 1,3,4,5,6,7,8,10,11,13
University of Manchester	Core 1,2,3
San Diego	Core 1,2,3 Optional 1,3,6,8,10,11
King's College	Core 1,2,3
Harvard	Core 1,2,3 Optional 1,3,6,8,10,11