Metanome

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Data Profiling with Metanome

Exercise: INDs

Thorsten Papenbrock
PhD Candidate
Hasso-Plattner-Institute

Advanced Profiling
Three important metadata

Unique Column Combinations

Key candidates

Name	Type	Equatorial diameter	Mass
Mercury	Terrestrial	0.382	0.06
Venus	Terrestrial	0.949	0.82
Earth	Terrestrial	1.000	1.00
Mars	Terrestrial	0.532	0.11
Jupiter	Giant	11.209	317.8
Saturn	Giant	9.449	95.2
Uranus	Giant	4.007	14.6
...	...	...	...

|Name|

Inclusion Dependencies

Foreign key candidates

Sign	Domicile
Aries	Mars
Taurus	Venus
Gemini	Mercury
Cancer	Moon
Leo	Sun
Virgo	Mercury
Libra	Venus
Scorpio	Pluto
Sagittarius	Jupiter
Capricorn	Saturn
Aquarius	Uranus
...	...

Name	Type
Mercury	Terrestrial
Venus	Terrestrial
Earth	Terrestrial
Mars	Terrestrial
Jupiter	Giant
Saturn	Giant
Uranus	Giant
...	...

Domicile ⊆ Name

Functional Dependencies

Normalization criterion

Name	Atmosphere	Rings
Mercury	minimal	no
Venus	CO_2, N_2	no
Earth	N_2, O_2, Ar	no
Mars	CO_2, N_2, Ar	no
Jupiter	H_2, He	yes
Saturn	H_2, He	yes
Uranus	H_2, He	yes
...	...	...

Atmosphere → Rings

Exercise 1
Discovery of key candidates

- All teams have passed the exercise:
  - 36 submissions
  - No duplicate algorithm names!
  - Publication usually not wanted

- Tips:
  - Use the mailing-list for help!
  - Check your algorithm before submission
  - Use the Metanome helper projects carefully

Exercise 1
NULL values

Hasso Plattner Institut

NULL = NULL
can cause …

larger PLIs
(one additional large index-set for NULL values)

larger UCCs
(attributes with NULLs need other attributes to become unique)

NULL ≠ NULL
can cause …

smaller PLIs
(NULL values can directly be ignored during PLI creation)

smaller UCCs
(attributes with NULL values can be unique themselves)

Exercise 1
NULL values

NULL = NULL
can cause …

more minimal UCCs
(attributes are unique in different combinations)

A	B	C	D	E
-	-	-	1	1
-	-	-	2	2
1	1	1	-	-
2	2	2	-	-

NULL = NULL
{A,D}, {A,E}, {B,D}, {B,E}, {C,D}, {C,E}

NULL ≠ NULL
{A}, {B}, {C}, {D}, {E}

NULL ≠ NULL
can cause …

more minimal UCCs
(attributes with NULL values are unique itself)

A	B	C
-	1	1
-	2	2

NULL = NULL
{B}, {C}

NULL ≠ NULL
{A}, {B}, {C}

Exercise 1
Our evaluation

- DELL Optiplex 9010
  - CPU: Intel i5 3.2 GHz
  - RAM: 8 GB (2 GB for Metanome JVM)
  - OS: Debian 64-bit
  - JVM: Java 1.7

AlexoFredUniques
AWsome
Buelow_Flemming_UCC
CPUcc
dennis_marius.ucc
DJ_UCC
dpdc-cnms-ucc
DreamteamUcc
DreamteamUcc-par
EnteEnte
FrohnOttoUcc
HorLehUcc
klinger_marten_ucc
LucieKerstinUCC
Metanomnomnom
MgEsUcc
MMUniqueColumnCombinations
olorin
PracticeUccJoerke
RobertJPUniqueCol
RohloffUcc
SBMMUCC
Tsun12Ucc
ucc
ucc_bothe_reissaus
UCC_Grundke_Wiese
Ucc_Jung
Ucc_Kirsten_Zwerg
UCC_schaeffer_zoellner
UCC_SPIRO
UCCDraegerSchueler
UCCFMJR
UccHeikoMax
UccPerchykSchmidt
UniColCom
UniqueColumnCombinationDiscoverer
YUCC

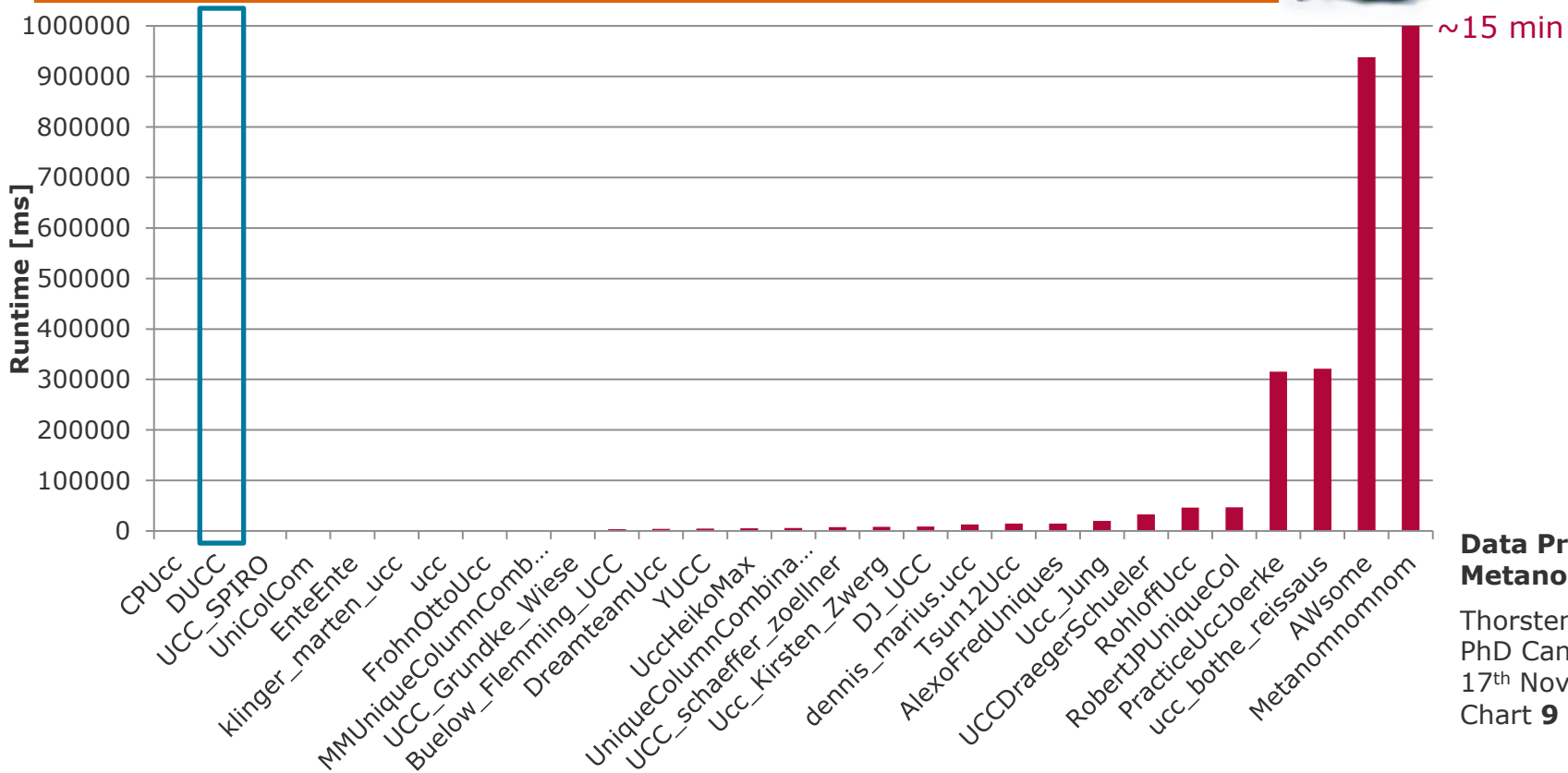
HPI Hasso Plattner Institut

AlexoFredUniques	
AWsome	
Buelow_Flemming_UCC	
CPUcc	
dennis_marius.ucc	
DJ_UCC	
dpdc-cnms-ucc	**Incorrect Result**
DreamteamUcc	
DreamteamUcc-par	
EnteEnte	
FrohnOttoUcc	
HorLehUcc	**NullPointerException**
klinger_marten_ucc	
LucieKerstinUCC	**NullPointerException**
Metanomnomnom	
MgEsUcc	**NullPointerException**
MMUniqueColumnCombinations	
olorin	**NoSuchMethodException**
PracticeUccJoerke	
RobertJPUniqueCol	
RohloffUcc	
SBMMUCC	**NoSuchMethodError**
Tsun12Ucc	
ucc	
ucc_bothe_reissaus	
UCC_Grundke_Wiese	
Ucc_Jung	
Ucc_Kirsten_Zwerg	
UCC_schaeffer_zoellner	
UCC_SPIRO	
UCCDraegerSchueler	
UCCFMJR	**NoSuchMethodError**
UccHeikoMax	
UccPerchykSchmidt	**NoSuchMethodError**
UniColCom	
UniqueColumnCombinationDiscoverer	
YUCC	

Reasons:

1. Java 1.8
2. Old Metanome version
3. Programming error

Exercise 1
Our evaluation – ncvoter-1k

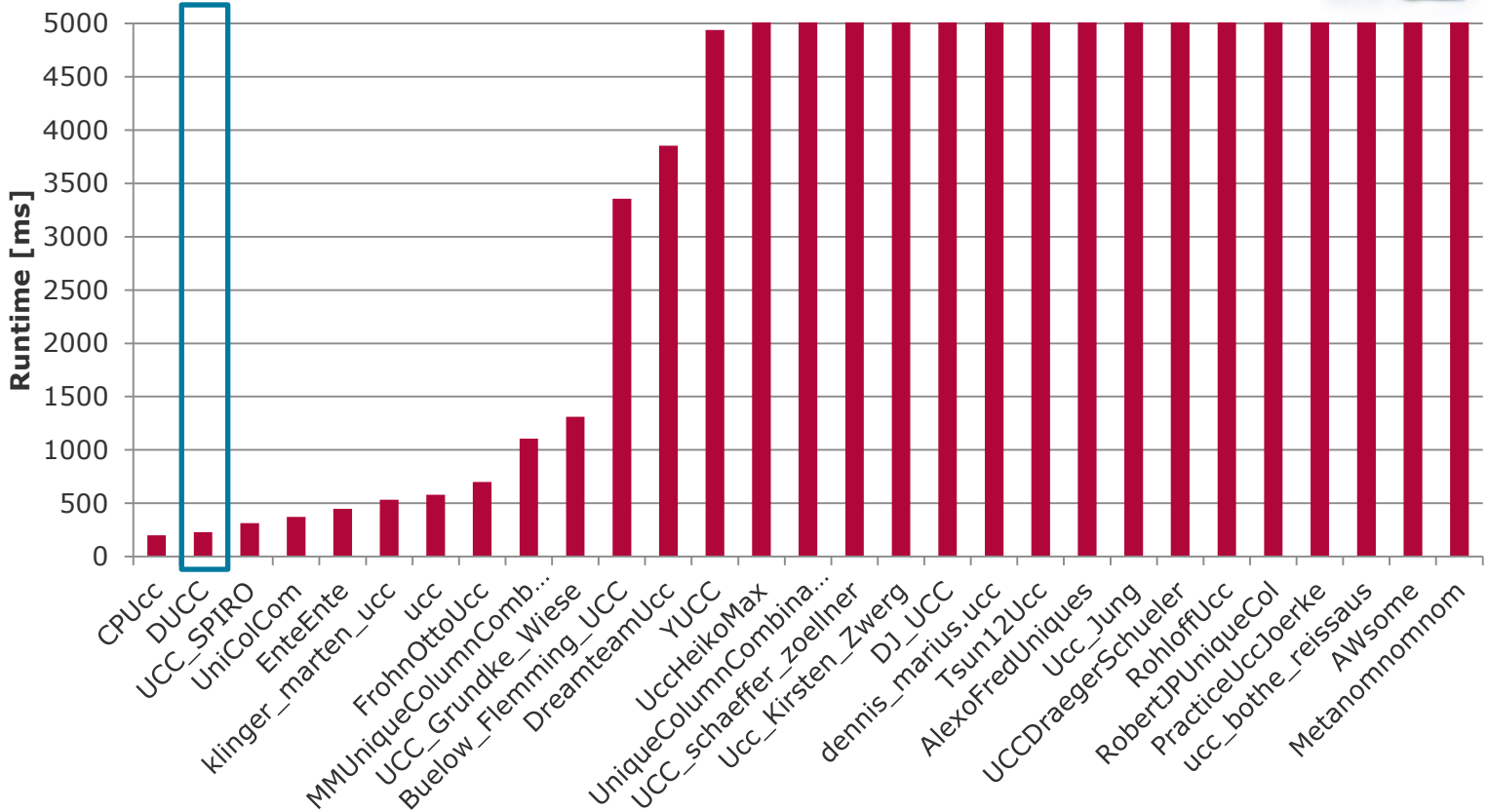


~15 min

Data Profiling with Metanome

ThorstenPapenbrock,
PhD Candidate,
17th November, 2014
Chart **9**

Exercise 1
Our evaluation – ncvoter-1k (<5s)

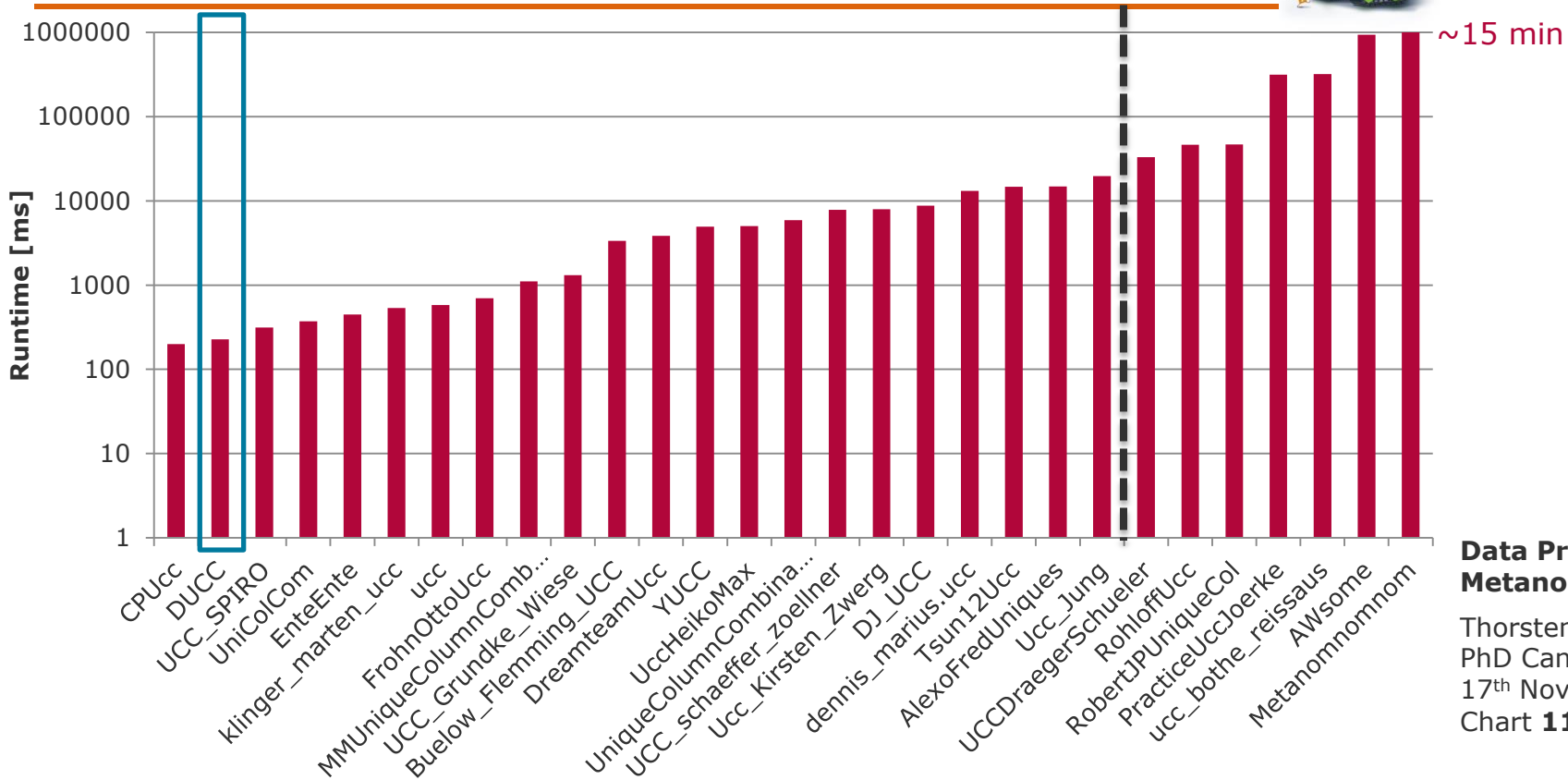


HPI Hasso Plattner Institut

Data Profiling with Metanome

ThorstenPapenbrock,
PhD Candidate,
17th November, 2014
Chart **10**

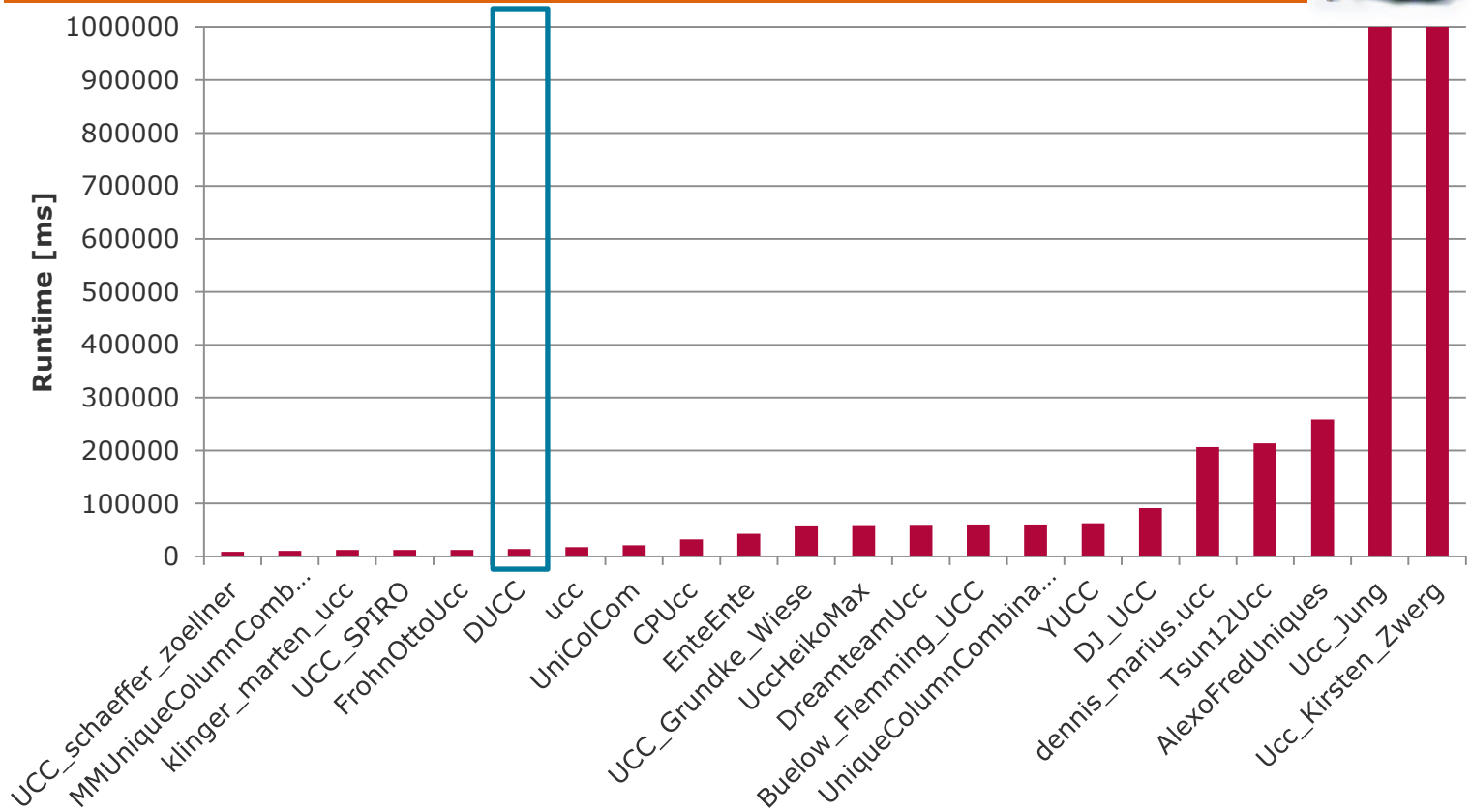
Exercise 1
Our evaluation – ncvoter-1k (log)



Data Profiling with Metanome

ThorstenPapenbrock,
PhD Candidate,
17th November, 2014
Chart **11**

Exercise 1
Our evaluation – fd_reduced_15

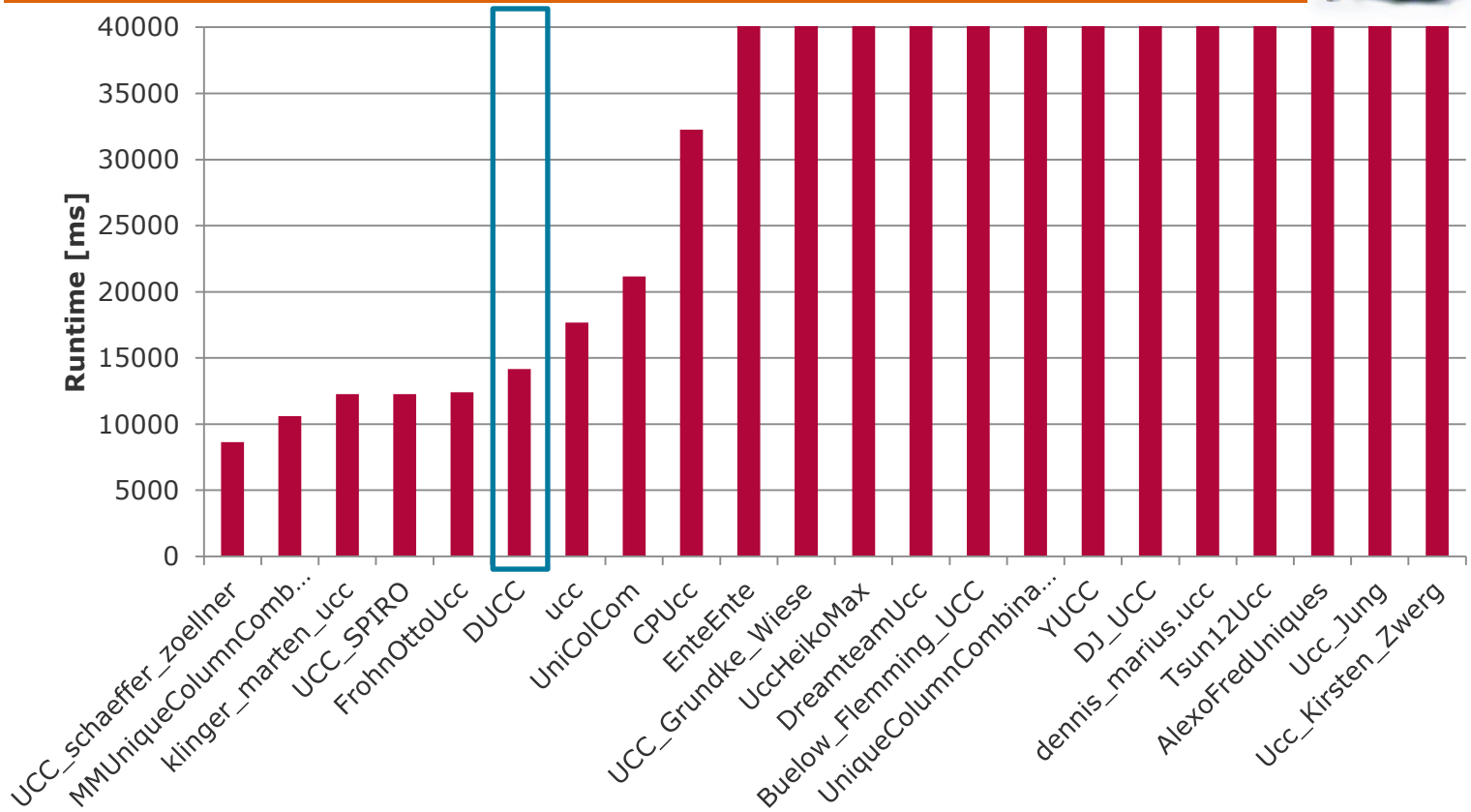


~15 min

Data Profiling with Metanome

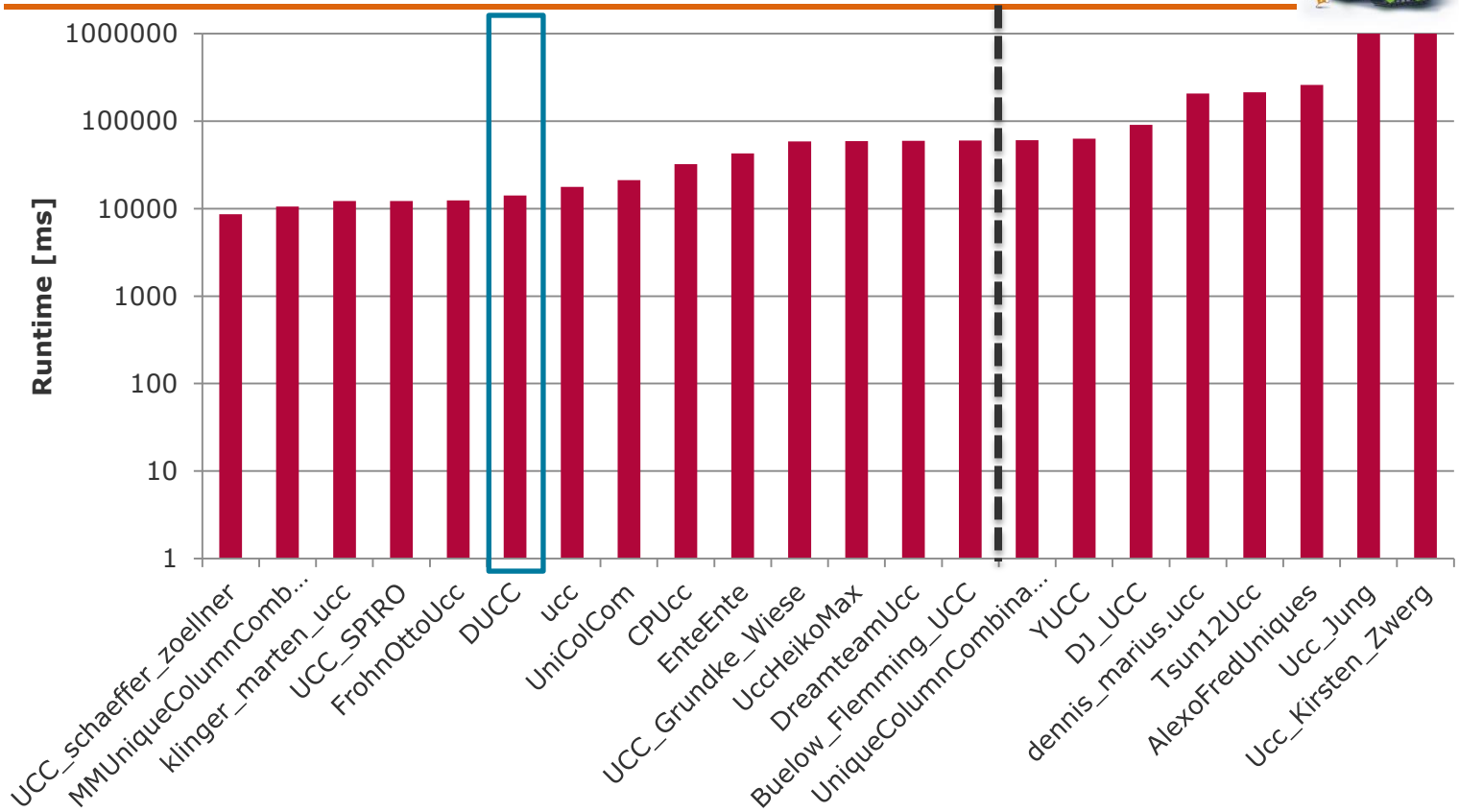
ThorstenPapenbrock,
PhD Candidate,
17th November, 2014
Chart **12**

Exercise 1
Our evaluation – fd_reduced_15



~15 min

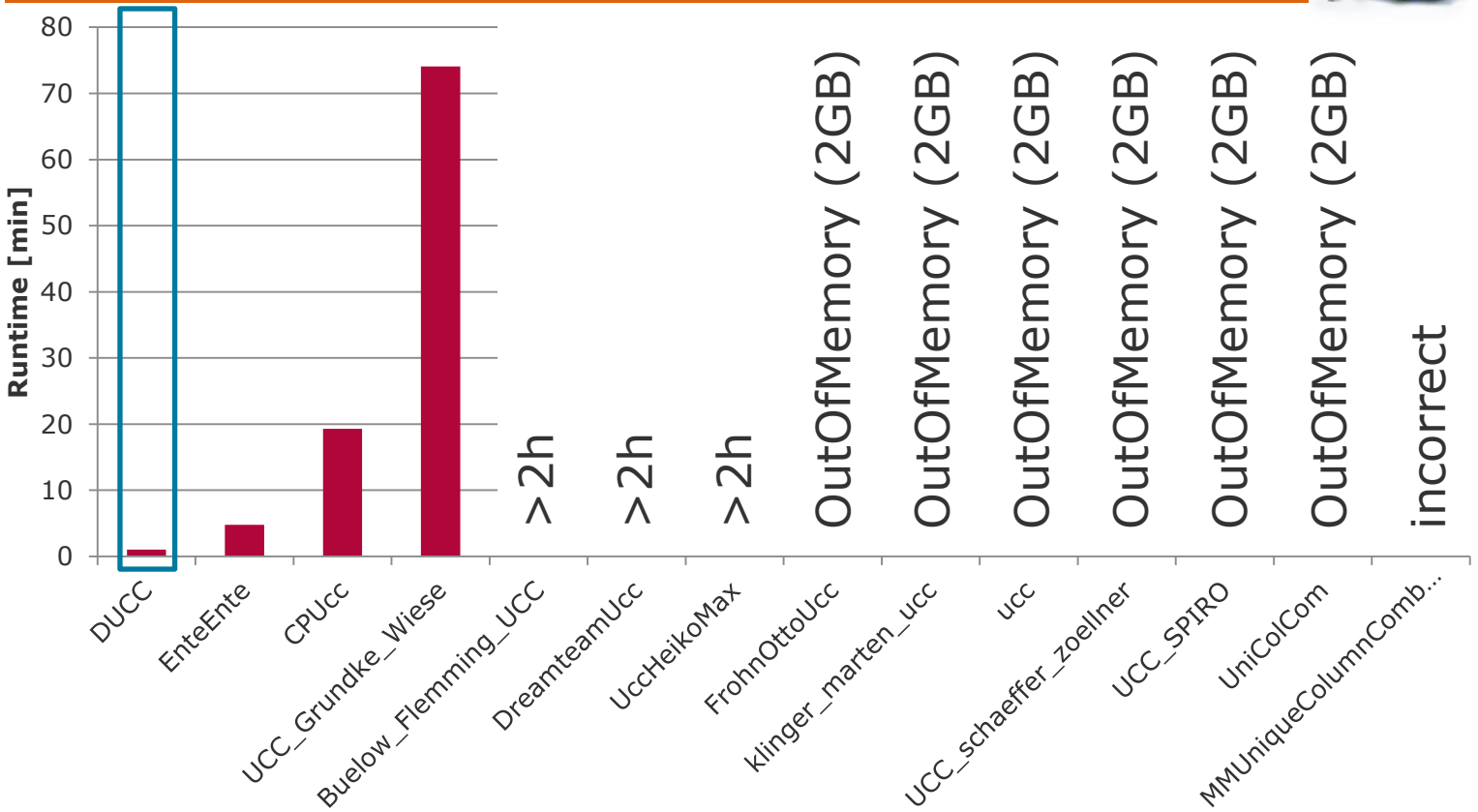
Exercise 1
Our evaluation – fd_reduced_15



Data Profiling with Metanome

ThorstenPapenbrock,
PhD Candidate,
17th November, 2014
Chart **14**

Exercise 1
Our evaluation – ncvoter-1m



Exercise 1
Short presentations

Advanced Profiling
Three important metadata

Hasso Plattner Institut

Unique Column Combinations

Key candidates

Name	Type	Equatorial diameter	Mass
Mercury	Terrestrial	0.382	0.06
Venus	Terrestrial	0.949	0.82
Earth	Terrestrial	1.000	1.00
Mars	Terrestrial	0.532	0.11
Jupiter	Giant	11.209	317.8
Saturn	Giant	9.449	95.2
Uranus	Giant	4.007	14.6
...	...	...	...

|Name|

Inclusion Dependencies

Foreign key candidates

Sign	Domicile
Aries	Mars
Taurus	Venus
Gemini	Mercury
Cancer	Moon
Leo	Sun
Virgo	Mercury
Libra	Venus
Scorpio	Pluto
Sagittarius	Jupiter
Capricorn	Saturn
Aquarius	Uranus
...	...

Name	Type
Mercury	Terrestrial
Venus	Terrestrial
Earth	Terrestrial
Mars	Terrestrial
Jupiter	Giant
Saturn	Giant
Uranus	Giant
...	...

Domicile ⊆ Name

Functional Dependencies

Normalization criterion

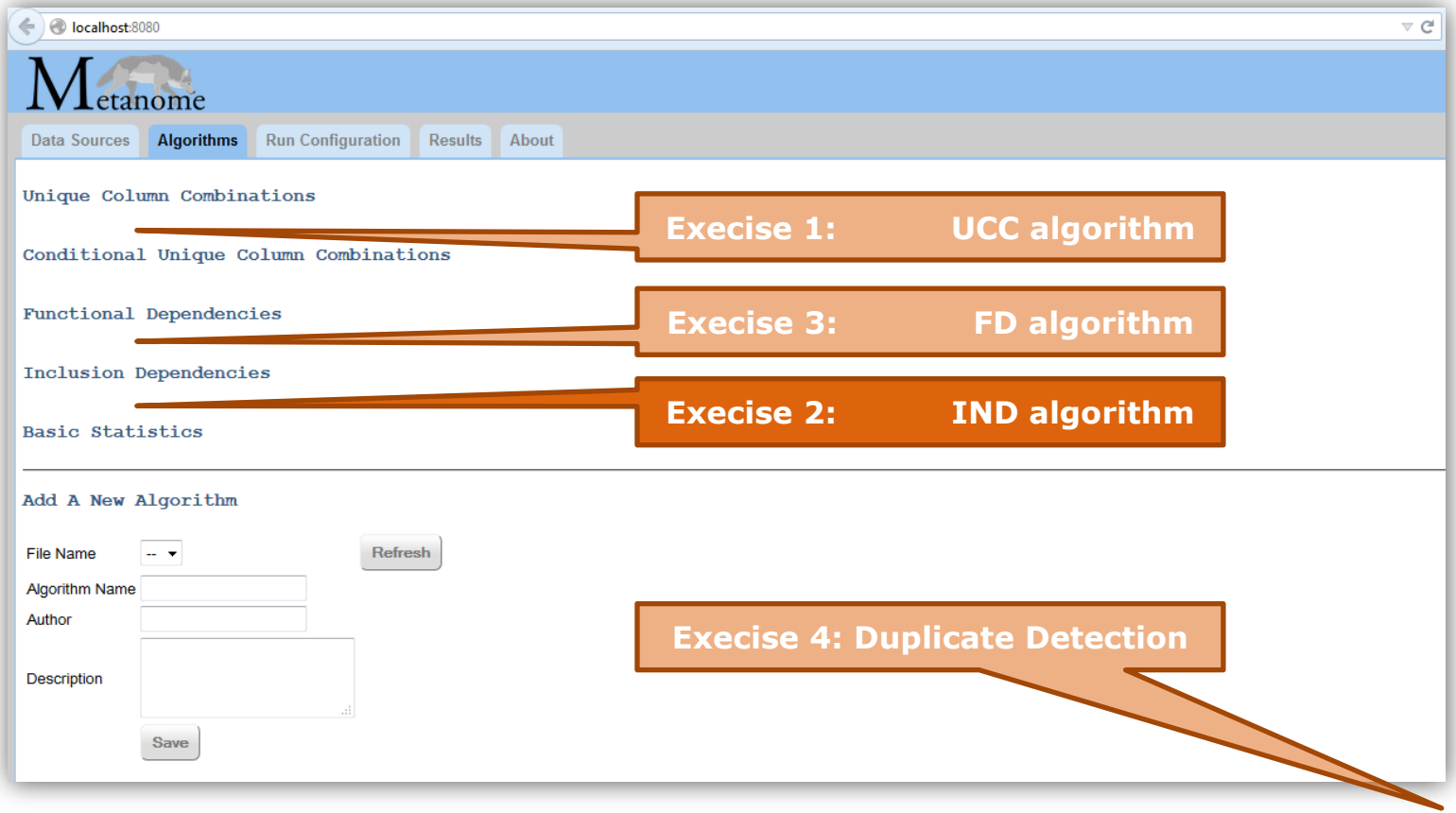
Name	Atmosphere	Rings
Mercury	minimal	no
Venus	CO_2, N_2	no
Earth	N_2, O_2, Ar	no
Mars	CO_2, N_2, Ar	no
Jupiter	H_2, He	yes
Saturn	H_2, He	yes
Uranus	H_2, He	yes
...	...	...

Atmosphere → Rings

Data Profiling with Metanome

ThorstenPapenbrock,
PhD Candidate,
17th November, 2014
Chart **17**

Metanome
The exercise: Build your own profiling tool



Exercise 2
Discovery of foreign-key candidates

Exercise 2
Inclusion Dependencies

- Deadline: **Monday, 01.12.14**
- The admission to the exam requires *all* exercises to be solved.
- The exercises should be solved in teams of two students.
- The Metanome project is available at GitHub:
  `https://github.com/HPI-Information-Systems/Metanome`
- The datasets and supplemental material can be found at network drive S:
  `\\fs3\bbs\DPDC`
- The submission system can be found at:
  `https://www.dcl.hpi.uni-potsdam.de/submit/`
- To solve an exercise, please submit a zip file containing the following items:
  - **<algorithm_name>.jar**: An executable Metanome algorithm.
  - **<algorithm_name>.zip**: The algorithm's source code (maven project).
  - **<algorithm_name>_docu.pdf**: Short documentation of the algorithm.
  - **<algorithm_name>_pres.pptx/ppt/pdf**: Two slides presentation of the algorithm.

Exercise 2
Discovery of foreign-key candidates

Task 1: Inclusion Dependencies - A discovery algorithm

Write an algorithm that discovers *all unary* inclusion dependencies on the given datasets. The rules for your implementation are as follows:

a) The algorithm discovers *exact* results, so no approximate or fuzzy results are allowed.

b) The algorithm is not allowed to use parallelization.

c) The algorithm implements the Metanome interface and is compatible with Metanome.

d) The algorithm takes an arbitrary number of tables as input.

e) The algorithm ignores NULL values.

Note that in contrast UCC and FD discovery, IND discovery uses *multiple tables*. INDs should therefore be discovered within and between tables. You can re-implement an existing algorithm from literature or find your own algorithm. To test and evaluate the algorithm, use the datasets provided on the network share. Your algorithm should at least be able to process the WDC dataset! Before submitting your algorithm, check that it correctly executes within Metanome!

BONUS TASK: If you like to dive deeper, you can try to find n-ary inclusion dependencies as well. Can you think of pruning rules for the n-ary discovery? You can also try parallelization or approximate strategies on your algorithm. If you made changes to your main algorithm, please submit them as a separate algorithm, e.g. <algorithm_name>_nary.jar

Exercise 2
Discovery of foreign-key candidates

```java
public class RelationalMyIndDetector
    extends MyIndDetector
    implements InclusionDependencyAlgorithm, RelationalInputParameterAlgorithm {

    public enum Identifier {INPUT_GENERATOR}

    @Override
    public ArrayList<ConfigurationRequirement> getConfigurationRequirements() {
        ArrayList<ConfigurationRequirement> configs = new ArrayList<ConfigurationRequirement>(1);
        configs.add(new ConfigurationRequirementRelationalInput(
            RelationalMyIndDetector.Identifier.INPUT_GENERATOR.name(), ConfigurationRequirement.ARBITRARY_NUMBER_OF_VALUES));
        return configs;
    }
    @Override
    public void setRelationalInputConfigurationValue(String identifier, RelationalInputGenerator... values)
        throws AlgorithmConfigurationException {
        if (RelationalMyIndDetector.Identifier.INPUT_GENERATOR.name().equals(identifier)) {
            this.relationalInputGenerators = values;
            // Useful for relation naming: this.relationalInputGenerators[0].generateNewCopy().relationName()
        }
        else
            this.handleUnknownConfiguration(identifier, CollectionUtils.concat(values, ","));
    }
    @Override
    public void setResultReceiver(InclusionDependencyResultReceiver resultReceiver) {
        this.resultReceiver = resultReceiver;
    }
    protected void handleUnknownConfiguration(String identifier, String value)
        throws AlgorithmConfigurationException {
        throw new AlgorithmConfigurationException("Unknown configuration: " + identifier + " -> " + value);
    }
    @Override
    public void execute() throws AlgorithmExecutionException {
        super.execute();
    }
}
```

Exercise 2
Discovery of foreign-key candidates

Task 2: Documentation

Write a short (max one A4 page) documentation for your algorithm describing the algorithm that you implemented:

a) Describe the algorithm's basic idea. How does the algorithm cope with the complexity of the given task?

b) If you used an algorithm from literature, provide a reference to the according publication.

c) If you came up with an own approach, provide one or two arguments why it is or could be better than related algorithms.

d) If your algorithm implements an adaption or optimization of existing approaches, describe these briefly.

e) State if your algorithm (jar) should be published as Metanome algorithm providing you as the authors.

f) *If you solved a bonus task, please discuss your findings here as well.*

In the same document, answer the following questions:

a) How many inclusion dependencies did your algorithm find on the provided datasets?

b) How long did the discovery take on each dataset and what machine did you use?

c) Did you discover any limitations of your approach (e.g. runtime or memory consumption) that made computing a certain dataset impossible?

d) Why is it a good idea to ignore NULL values for IND discovery?

Exercise 2
Discovery of foreign-key candidates

Task 3: Presentation

Prepare two slides for a short, 5 min presentation of your algorithm in the lecture. One slide about the algorithm and one slide about its performance. Here are some ideas for these slides:

a) Explain the idea of your approach and why you think it works well in comparison to others.

b) How did your algorithm perform on the given datasets?

c) Did you make any interesting observations?

d) What lets your algorithm crash?

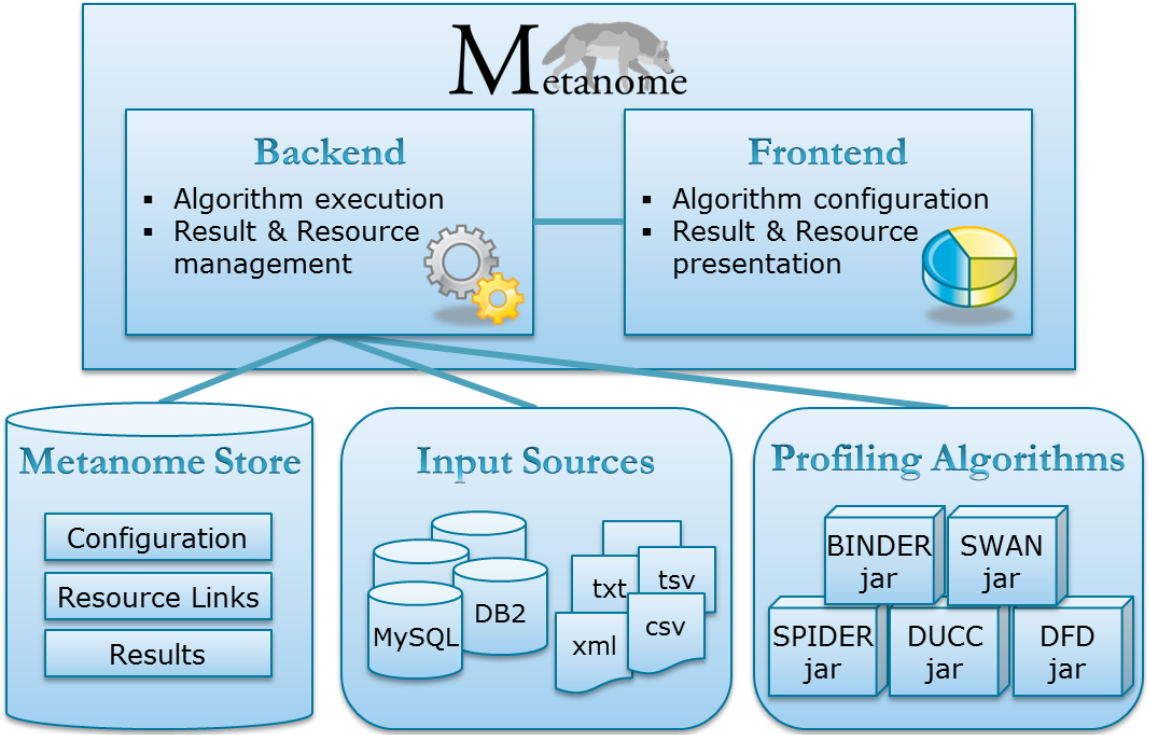
e) How hard was it to implement your algorithm?

f) *This is also a good time to introduce an optimization or adaption of your algorithm.*

Note that each team will present its work once!

Exercise 2
Hinweise zum Referenzalgorithmus BINDER

- **INPUT_GENERATPOR: WDC_planets, WDC_astronomical, …**
  - <die Tabellen, die durchsucht werden sollen; nur angegebene Tabellen werden genutzt>
- **INPUT_RO_LIMIT: -1**
  - <der Parameter erlaubt die künstliche Verkürzung aller Tabellen auf X Zeilen; -1 nimmt alle Tupel>
- **TEMP_FOLDER_PATH: temp**
  - <Verzeichnis für temporäre Daten; relativ zum Verzeichnis der Metanome-Instanz>
- **CLEAN_TEMP: true**
  - <sollen die temporären Daten gelöscht werden nach der Ausführung?>
- **DETECT_NARY: false**
  - <BINDER kann auch n-ary INDs finden; für eure Übung braucht ihr aber nur unary INDs>



HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Data Profiling with Metanome

www.metanome.de

Thorsten Papenbrock
PhD Candidate
Hasso-Plattner-Institute