

Analyse, Design, Entwicklung und Evaluation eines  
skalierbaren, Echtzeit Entity Resolution Streaming  
Framework

---

Kevin Sapper

19.10.16  
Version: 0.1



Hochschule RheinMain



Hochschule **RheinMain**

DCSM - Design Informatik Medien  
Informatik (M.Sc.)

Masterarbeit

Analyse, Design, Entwicklung und Evaluation eines  
skalierbaren, Echtzeit Entity Resolution Streaming  
Framework

Kevin Sapper

Referent      Prof. Dr. Adrian Ulges  
Hochschule RheinMain  
DCSM - Design Informatik Medien

Koreferent    Prof. Dr. Reinhold Kröger  
Hochschule RheinMain  
DCSM - Design Informatik Medien

Betreuer      Thomas Strauß  
Universum Group

19.10.16

Kevin Sapper

Analyse, Design, Entwicklung und Evaluation eines skalierbaren, Echtzeit Entity Resolution  
Streaming Framework

Masterarbeit, 19.10.16

Referenten: Prof. Dr. Adrian Ulges und Prof. Dr. Reinhold Kröger

Betreuer: Thomas Strauß

Hochschule RheinMain

Informatik (M.Sc.)

DCSM - Design Informatik Medien

Kurt-Schumacher-Ring 18

65197 Wiesbaden

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Blocking . . . . .	5
2.1.1	Standard Blocking . . . . .	5
2.1.2	Q-gram Indexing . . . . .	6
2.1.3	Suffix Array Indexing . . . . .	7
2.1.4	Sorted Neighborhood . . . . .	7
2.1.5	Canopy Clustering . . . . .	8
2.2	Blocking-Paralellisierung . . . . .	9
	<b>Literaturverzeichnis</b>	<b>11</b>
	<b>Abbildungsverzeichnis</b>	<b>13</b>
	<b>Auflistungsverzeichnis</b>	<b>15</b>
	<b>Tabellenverzeichnis</b>	<b>17</b>
	<b>Erklärung</b>	<b>19</b>



# Einleitung

Die Methoden zur Duplikatserkennung stammen ursprünglich aus dem Gesundheitsbereich (Felegi & Sunter 1969). Je nach Fachgebiet gibt es unterschiedliche Fachbegriffe. Statistiker und Epidemiologen sprechen von record oder data linkage während Informatiker das Problem unter entity resolution, data oder field matching, duplicate detection, object identification oder merge/purge kennen. Dabei geht es nicht um die reine Personenidentifikation, sondern vielmehr um die Identifikation von Entitäten aller Art, beispielsweise Kunden, Patienten, Produkte oder Orte. Dabei können die Entitäten nicht durch ein einzigartiges Attribut identifiziert werden. Zudem sind die Datensätze oft fehlerhaft, beispielsweise durch Rechtschreibfehler oder unterschiedliche Konventionen. Die Methoden zur Entitätsauflösung arbeiten meist auf Datensatzpaaren und liefern als Ergebnis eine Menge von Übereinstimmungen. Eine Übereinstimmung verknüpft zwei Entitäten. Zusätzlich kann über einen optionalen Ähnlichkeitswert (engl. similarity score), normalerweise zwischen 0 und 1, die Intensität der Übereinstimmung angegeben [1].

Zur Bestimmung der Ähnlichkeit eines Datensatzpaares unterscheiden Elmagarmid et al. [??] zwischen Attributvergleichs- (engl. field matching) und Datensatzvergleichsmethoden (engl. record matching). Methoden zum Attributvergleich sind zeichenbasierend (edit distance, affine gap distance, Jaro distance metric oder Q-gram distance), tokenbasierend (atomic strings, Q-grams mit tf.idf), phonetisch (soundex) oder numerisch. Die Datensatzvergleichsmethoden sind probabilistisch (Naive Bayes), überwachtes bzw. semi-überwachtes Lernen (SVMlight, Markov Chain Monte Carlo), aktives Lernen (ALIAS), distanzbasierend (siehe Attributvergleich - Datensatz als konkatenierter String) oder regelbasierend (AJAX). Die Ausführung der Vergleichsmethoden ist enorm teuer, da diese das Kreuzprodukt zweier Mengen bilden müssen. Um die Ausführungszeit zu reduzieren wird versucht den Suchraum auf die wahrscheinlichsten Duplikatsvorkommen zu begrenzen. Diese Vorgehen werden als Blocking oder Indexing bezeichnet. Elmagarmid et al. nennen Standard Blocking, Sorted Neighborhood Approach, Clustering und Canopies, sowie Set Joins als Vorgehensweisen. (Referenzen zu den Methoden folgen noch!)

Da es keine Methode zur Entity Resolution gibt, welche allen anderen überlegen ist, wurden Ende der 90er Jahre begonnen Frameworks zu entwickeln, welche verschiedene Methoden miteinander kombinieren. Einen Vergleich dieser Frameworks wurde durch Köpcke & Rahm 2010 [??] durchgeführt. Ein Framework besteht aus verschie-

denen Matchern. Ein Matcher ist dabei ein Algorithmus, welcher die Ähnlichkeit zweier Datensätze ermittelt. Ähnlich wie Elmagarmid et al. unterscheiden Köpcke & Rahm zwischen attributs- und kontextbasierenden Matchern. Als Kontext bezeichnen Sie die semantische Beziehung bzw. Hierarchie zwischen den Attributen. Um die Matcher miteinander zu kombinieren nutzen die Frameworks min. eine Matching Strategie. Eine Strategie ist, die Ähnlichkeitswerte verschiedener Matcher numerisch zu kombinieren, beispielsweise durch eine gewichtete Summe oder einen gewichteten Durchschnitt. Ein anderer Ansatz ist regelbasierend. Eine einfache Regel besteht aus einer logischen Verbindung und einer Match-Kondition, beispielsweise einem Schwellenwert. Die dritte und komplexeste Strategie ist workflow-basierend. Hierbei kann beispielsweise eine Sequenz von Matchern die Ergebnisse iterativ einschränken. Grundsätzlich können Workflows beliebig komplex werden. Einen passenden Workflow zu finden kann selbst Domainexperten vor eine große Herausforderung stellen. Daher gibt es trainingbasierende Ansätze passende Parameter für Matcher oder Kombinationsfunktionen (z.B. Gewicht für Matcher) zu bestimmen. Solche Ansätze sind etwa, Naive Bayes, Logistic Regression, Support Vector Maschine oder Decision Trees. (Referenzen zu den Ansätzen folgen noch!)

Ein Großteil der Forschung in Entity Resolution konzentriert sich auf die Qualität der Vergleichsergebnisse. Die von Köpcke & Rahm verglichenen Frameworks konzentrieren sich alle auf zwei statische Mengen zu miteinander vergleichen. Bei großen Datenmengen kann dies durchaus mehrere Stunden dauern. Daher gibt es in den letzten Jahre einige Ansätze und Frameworks, welche MapReduce zum Skalieren nutzen [???][???]. Zudem gibt es immer mehr Bedarf, Vergleichsergebnisse in nahe Echtzeit zu liefern. Erste Ergebnisse Entity Resolution skalierbar und in nahe Echtzeit zu erreichen, präsentieren Christen & Gayler in [???] 2008, unter Verwendung von Inverted Indexing Techniken, welche normalerweise bei der Websuche anwendung finden. Dabei betrachten Sie vor allem die Anforderungen eines Anfragestroms (engl. query stream). Ihre Anforderungen sind einen Strom von Anfragedatesätzen, gegen potentielle riesige Datenmengen, im Subsekundenbereich pro Anfrage abzuarbeiten. Dabei sollen die Treffer der Anfrage mit einem Ähnlichkeitswert versehen sein. Zudem muss es möglich sein die Menge an Anfragen zu skalieren. Das Hauptproblem ist hierbei die Skalierung. Um skalieren zu können wird versucht die Abarbeitung des Suchraums zu parallelisieren. Eine Studie von Kwon, Balazinska, Howe, & Rolia [???] in MapReduce Anwendungen zeigt, dass selbst geringe Ungleichgewichte bei der Verteilung des Suchraum auf Mapper bzw. Reducer, aufgrund der Komplexität der Matching Algorithmen, zu deutlich längeren Laufzeiten und damit Gesamtlaufzeiten führt. In einem ihrer Beispiele sind bei einer Gesamtzeit von 5 Minuten die meisten Mapper innerhalb von 30 Sekunden fertig. Auch beim Streaming kann diese sog. Datenschiefe (engl. data skew) den Durchsatz eines Clusters signifikant mindern. Einen weiteren Ansatz die Laufzeit für nahe Echtzeit Anwendungen zu optimieren präsentieren Whang et al. [???]. An-



statt eine Ergebnismenge nach Abschluss eines Algorithmus zu liefern, zeigen Sie Möglichkeiten partielle Ergebnisse während der Laufzeit des Algorithmus zu erhalten.

Im Rahmen der Thesis soll ein Entity Resolution Framework für Datensatzströme entstehen. Als Basis soll ein (Event) Stream Processing Framework genutzt werden. Das Framework soll eine Reihe von Matchern, sowie Kombinationsfunktionen der Matcher unterstützen. Hauptaugenmerk ist jedoch die Skalierbarkeit. Gelöst werden soll das Data Skew Problem bei verschiedenen Blocking Strategien. Eine weitere Schwierigkeit ist, dass die Datenmenge nicht statisch ist, sondern neue Datensätze jederzeit hinzukommen können. Beim Erweitern des Suchraums soll beachtet werden, dass kein Data Skew auftritt. Dadurch soll vermieden werden, dass der Durchsatz innerhalb des Clusters signifikant sinkt. Idealerweise soll der Durchsatz, sowie die Qualität der Suchergebnisse, mit bereits bekannten Veröffentlichungen verglichen werden. Das Framework soll dabei kein Domainwissen eines bestimmten Entitätstypen berücksichtigen.



# Grundlagen

## Blocking

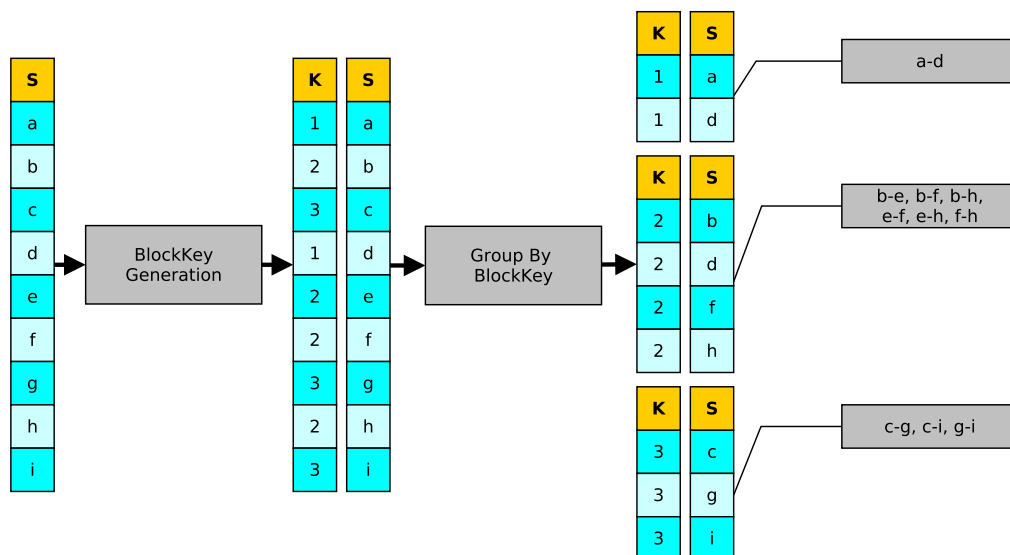
Für die Duplikatserkennung in zwei Datenquellen  $A$  und  $B$  sind  $|A| * |B|$  Paarvergleiche notwendig. Bei einer einzelnen Datenquelle  $A$  müssen  $\frac{1}{2} * |A| * (|A| - 1)$  durchgeführt Vergleiche werden. In beiden Fällen ist die Anzahl der Vergleiche quadratisch zur Eingabemenge [2]. In der Studie [3] zeigen Köpcke et al., dass das kartesische Produkt für große Datenmengen nicht skaliert. Aus diesem Grund reduzieren moderne Entity Resolution Frameworks den Suchraum auf die wahrscheinlichen Kandidaten, die sogenannten Match-Kandidaten. Diese Methoden zur Reduzierung des quadratischen Suchraum werden übergreifend als Blockingmethoden bezeichnet. Neben Blocking werden auch Windowing- und Indexing Verfahren eingesetzt. Während Blockingverfahren die Anzahl der notwendigen Vergleiche drastisch reduzieren, indem Non-Matches ausgeschlossen werden, besteht dennoch die Gefahr, dass fälschlicherweise tatsächliche Matches rausgefiltert werden. Daher ist es notwendig die Güte des Blockingverfahrens zu bestimmen. Dazu werden zwei Kennziffern erhoben. Zum einen die Reduction Ratio, welche die Reduzierung der Vergleiche im Gegensatz zum Kartesischen Produkt ausdrückt, sowie die Pairs Completeness, welche den Anteil der tatsächlich ausgewählten Duplikate, die sich nach dem Blocking in der Kandidatenmenge befinden, beschreibt.

Prinzipiell erfolgt Blocking entweder durch Gruppierung oder Sortierung. Dadurch sollen sich mögliche Duplikate in der "Nähe" von einander befinden. Zur Durchführung der Gruppierung oder Sortierung müssen sog. Block- bzw. Sortierschlüssel für jeden Datensatz erzeugt werden. Diese Schlüssel werden von den Attributwerten oder einem Teil der Attributwerte abgeleitet und stellen eine Signatur des Datensatzes dar.

## Standard Blocking

Standard Blocking ist eine der ersten und populärsten Blockingmethoden [4]. Die Idee ist eine Menge von Datensätzen in disjunkte Partitionen (gennant Blöcke) zu teilen. Anschließend werden nur die Datensätze des jeweiligen Blocks miteinander verglichen. Dazu wird jedem Datensatz ein Blockschlüssel zugeordnet. Die Qualität

des Blockingverfahrens hängt daher maßgeblich vom gewählten Blockschlüssel ab, da dieser die Anzahl und Größe der Partitionen bestimmt. In einer Menge von Personen ist ein schlechter Blockschlüssel etwa das Geschlecht. Da dieser die Menge lediglich in zwei große Partitionen teilt. Ein besserer Blockschlüssel ist beispielsweise die Postleitzahl oder die ersten Ziffern der Postleitzahl [5]. Abbildung Abb. 2.1 zeigt die Ausführung des Blockingverfahrens beispielhaft an einer Datenquelle  $S$ . Zunächst wird jedem Datensatz (a - i) ein Blockschlüssel (hier 1, 2, 3) zugeordnet. Anschließend wird anhand dieses Schlüssels gruppiert. Die Größe der einzelnen Blöcke bestimmt die Reduction Ratio. Diese hängt allerdings immer von der Datenquelle ab und kann daher nicht pauschalisiert werden. Bei der Generierung der Blockschlüssel können fehlerhafte Werte einzelner Attribute dazu führen, dass Duplikate in unterschiedlichen Blöcken landen. Damit diese Duplikate dennoch gefunden werden, kann für jeden Datensatz mehrere Blockschlüssel, anhand unterschiedlicher Attribute, generiert werden. Dieser Ansatz nennt sich Multi-pass Blocking.



**Abbildung 2.1:** Beispielhafte Standard Blocking Ausführung. Nach [2].

## Q-gram Indexing

Das Q-gram Indexing basiert auf der Idee Datensätze unterschiedlicher aber ähnlicher Blockschlüssel miteinander zu vergleichen. Ein Blockschlüssel wird dazu in eine Liste  $G$  von q-Grammen überführt. Ein q-Gram ist ein Substring der Länge  $q$  des ursprünglichen Blockschlüssels. Alle Permutationen der q-Gram Liste mit einer Mindestlänge  $l = \max(1, \lfloor \#G * t \rfloor)$  werden konkateniert und dienen als Schlüssel der Blöcke, wobei  $t$  ein Schwellwert zwischen 0 und 1 ist. Dabei werden Datensätze mehreren Blöcken zugewiesen. Ist  $t = 1$  entspricht Q-gram Indexing dem Standard Blocking. Dieses Verfahren kann als Alternative zum Multi-pass Verfahren beim Standard Blocking genutzt werden. Der große Nachteil ist der hohe Auf-

wand bei der Berechnung aller möglichen Sublisten. Ein Blockschlüssel mit  $n$  Zeichen muss in  $k = n - q + 1$   $q$ -Gramme zerlegt werden. Insgesamt müssen dadurch  $\sum_{i=\max\{1, [k*t]\}}^k \binom{k}{i}$  Sublisten berechnet werden [6].

## Suffix Array Indexing

Das Suffix Array Indexing [7] leitet, ähnlich wie Q-gram Indexing, mehrere Schlüssel aus einem Blockschlüssel ab. Grundidee ist es alle Suffixe mit einer Mindestlänge von  $l$  zu bestimmen. Ein Datensatz mit Blockschlüssellänge  $n$  wird in  $n - l + 1$  Blöcke eingeordnet. Ist  $n < l$  wird der Ausgangsschlüssel als einziger Schlüssel verwendet. Durch die größere Menge an Kandidatenpaaren ist i.Allg. die Pair Completeness höher (vgl. Multi-pass). Zudem ist der Aufwand der Berechnung der Schlüssel im Gegensatz zu Q-grammen deutlich geringer. Im Gegensatz zum Standard Blocking ist die Menge an Kandidatenpaaren jedoch deutlich höher. Dadurch ist auch die Wahrscheinlichkeit, dass zwei Datensätze unnötigerweise mehrfach miteinander verglichen werden hoch. Deshalb werden aus Blöcken, welche einen bestimmten Schwellwert überschreiten alle Datensätze entfernt, die min. einen weiteren längeren Blockschlüssel haben.

## Sorted Neighborhood

Das Sorted Neighborhood Verfahren, ist ein Sortiervorgang, welches 1995 von Hernández & Stolfo zur Erkennung von Duplikaten in Datenbanktabellen vorgestellt wurde [8]. Es besteht aus drei Phasen. Zunächst bekommt jeder Datensatz einen Sortierschlüssel zugewiesen. Dabei muss der Sortierschlüssel nicht einzigartig sein. Um die Berechnung des Schlüssels gering zu halten, soll dieser durch Verkettung von Attributen bzw. Teilen der Attribute bestimmt werden. Attribute die vorne im Schlüssel stehen haben dadurch eine höhere Priorität. In der zweiten Phase werden die Datensätze anhand des Schlüssels sortiert. In der dritten Phase wird ein Fenster (engl. Window) über die sortierten Datensätze geschoben und alle Datensätze innerhalb des Windows werden miteinander verglichen. Dieses Verfahren eignet sich besonders gut zur Erkennung von Duplikaten innerhalb einer Datenquelle. Sollen Duplikate in mehreren Datenquellen gefunden werden, müssen die Einträge beim Sortieren gemischt werden. Dadurch besteht allerdings die Gefahr, dass vorrangig Datensätze einer Datenquelle miteinander verglichen werden. Vorteil zum Standard Blocking ist, dass die Anzahl der Vergleiche lediglich von der Größe der Datenquelle und der gewählten Fenstergröße abhängen. Ein großer Nachteil ist, dass Datensätze die sich in der ersten Stelle des Schlüssels unterscheiden, weit von einander entfernt sind und dadurch nicht als Matches identifiziert werden. Um dennoch eine hohe Pair Completeness zu erreichen, werden mehrere Schlüssel pro Datensatz generiert

und ein Fenster mit kleiner Größe über die verschieden sortierten Listen geschoben. Dieses Verfahren entspricht im Grunde dem Multi-pass Verfahren beim Standard Blocking.

Ein großes Problem bei der klassischen und der Multi-pass Variante des Sorted Neighborhood Verfahrens ist, dass die zu wählende Fenstergröße  $w$  größer als die Anzahl der Datensätze mit dem häufig vorkommensten Sortierschlüssel sein muss, um eine gute Pair Completeness zu erreichen. Sei  $n$  die Menge an Datensätzen mit dem am häufig vorkommensten Schlüssel  $k$  und  $m$  die Menge der Datensätze des darauffolgenden Schlüssels  $k + 1$ , dann ist  $w = n + m$ . Nur dadurch kann sichergestellt werden, dass alle Datensätze aus  $n$  mit den “nahen” Datensätzen aus  $m$  verglichen werden. Da Sortierschlüssel für gewöhnlich nicht gleichverteilt sind, gibt es meist wenige große und viele kleine Mengen an Datensätzen mit dem gleichen Sortierschlüssel. Dadurch werden Datensätze mit seltenen Sortierschlüsseln unnötig oft mit “weit” entfernten Datensätzen verglichen. Zudem dominiert der am häufig vorkommenste Schlüssel, genauso wie beim Standard Blocking, die Ausführungszeit des Algorithmus.

In [5] schlagen Draisbach & Naumann eine optimierte Variante des Sorted Neighborhood Verfahrens vor. Dabei zeigen Sie, dass Standard Blocking und Sorted Neighborhood extreme von Overlapping bei Partitionen sind. Ihre Idee ist es diese Überlappung zu optimieren. Dabei soll die Überlappung groß genug sein, um tatsächliche Matches zu finden, aber gering genug, um die Menge der Vergleiche zu reduzieren. Zunächst wird wie beim klassischen Verfahren zu sortiert. Danach werden angrenzende Datensätze in disjunkte Partitionen zerlegt und schließlich wird ein Überlappungsfaktor (gennant Overlap)  $u$  gewählt. Innerhalb jedes Blockes wird analog zum Standard Blocking jeder Datensatz mit jedem anderen verglichen. Innerhalb des Overlap-Window  $w = u + 1$ , wird jeweils das erste Element mit allen anderen verglichen. Ist  $w = 0$  entspricht das Verfahren Standard Blocking und hat jede Partition nur ein Element entspricht es der Sorted Neighborhood Methode. Um zu vermeiden, dass eine Partition dominiert, können größere Partitionen in Subpartitionen geteilt werden.

Weitere Varianten verändern die Fenstergröße anhand identifizierte Duplikate [9].

## Canopy Clustering

Die Idee von Canopy Clustering ist es Datensätze anhand einer einfachen Vergleichsmetrik in überlappende Cluster (=Canopies) zu partitionieren. Zur Generierung wird eine Kandidatenliste gebildet, welche initial als allen Datensätzen besteht.

Dann wird zufällig ein Zentroid eines neuen Clusters gewählt und alle Datensätze innerhalb des Mindestabstandes  $d_1$  zugewiesen. Zusätzlich werden alle Datensätze dieses Clusters mit einem weiteren Mindestabstandes  $d_2 < d_1$  aus der Kandidatenliste entfernt. Dieser Algorithmus wird wiederholt, bis die Kandidatenliste leer ist. Die Pair Completeness hängt hierbei stark der gewählten Abstandsfunktion ab. Anschließend werden alle Datensätze eines Cluster miteinander verglichen.

## Blocking-Paralellisierung



Hochschule RheinMain

**Abbildung 2.2:** My Logo

**Tabelle 2.1:** My Table

a	b	c
1	2	3
4	5	6

$$\text{math} \quad (2.1)$$

$$\text{math} \quad (2.2)$$

Auflistung 2.1 Listing caption

```
main :: IO ()
main = putStrLn "Hello World!"
```





# Literaturverzeichnis

- [1] Altwaijry, Hotham ; Mehrotra, Sharad ; Kalashnikov, Dmitri V.: QuERy: A Framework for Integrating Entity Resolution with Query Processing. In: Proc. VLDB Endow. Bd. 9 (2015), Nr. 3, S. 120–131
- [2] Kolb, Lars: Effiziente MapReduce-Parallelisierung von Entity Resolution-Workflows, University of Leipzig, Dissertation, 2014
- [3] Köpcke, Hanna ; Thor, Andreas ; Rahm, Erhard: Evaluation of Entity Resolution Approaches on Real-World Match Problems. In: Proceedings of the VLDB Endowment Bd. 3 (2010), Nr. 1-2, S. 484–493
- [4] Fellegi, Ivan P. ; Sunter, Alan B.: A Theory for Record Linkage. In: Journal of the American Statistical Association Bd. 64 (1969), Nr. 328, S. 1183–1210
- [5] Draisbach, Uwe ; Naumann, Felix: A Comparison and Generalization of Blocking and Windowing Algorithms for Duplicate Detection. In: Proceedings of the International Workshop on Quality in Databases (QDB), 2009, S. 51–56
- [6] Christen, P.: A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. In: IEEE Transactions on Knowledge and Data Engineering Bd. 24 (2012), Nr. 9, S. 1537–1555
- [7] Aizawa, A. ; Oyama, K.: A Fast Linkage Detection Scheme for Multi-Source Information Integration. In: International Workshop on Challenges in Web Information Retrieval and Integration, 2005, S. 30–39
- [8] Hernández, Mauricio A. ; Stolfo, Salvatore J.: The Merge/Purge Problem for Large Databases. In: Proceedings of the 1995 ACM SIGMOD International Confe-

rence on Management of Data, SIGMOD '95. New York, NY, USA : ACM, 1995  
— ISBN 978-0-89791-731-5, S. 127–138

[9] Draisbach, Uwe ; Naumann, Felix ; Szott, Sascha ; Wonneberg, Oliver: Adaptive Windows for Duplicate Detection. In: 2012 IEEE 28th International Conference on Data Engineering : IEEE, 2012, S. 1073–1083

# Abbildungsverzeichnis

2.1	Beispielhafte Standard Blocking Ausführung. Nach [2]. . . . .	6
2.2	My Logo . . . . .	9



# Auflistungsverzeichnis

2.1 Listing caption . . . . .	9
-------------------------------	---



# Tabellenverzeichnis

2.1	My Table . . . . .	9
-----	--------------------	---





# Erklärung

Erklärung gem. ABPO, Ziff. 6.4.3

Ich versichere, dass ich die Master-Thesis selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Wiesbaden, 19.10.16

---

Kevin Sapper

Hiermit erkläre ich mein Einverständnis mit den im Folgenden aufgeführten Verbreitungsformen dieser Master-Thesis:

Verbreitungsform	ja	nein
Einstellung der Arbeit in die Bibliothek der Hochschule RheinMain	✓	
Veröffentlichung des Titels der Arbeit im Internet	✓	
Veröffentlichung der Arbeit im Internet	✓	

Wiesbaden, 19.10.16

---

Kevin Sapper

