Exposé eines Promotionsprojektes: Provenance Management für Data-Science-Anwendungen unter Berücksichtigung von Daten- und Schema-Evolution

Tanja Auge

Lehrstuhl für Datenbank- und Informationssysteme
Institut für Informatik
Universität Rostock

Im Promotionsvorhaben sollen Techniken des Provenance Managements im Bereich des Forschungsdatenmanagements angewendet und verallgemeinert werden. Die Forschungsdaten werden laufend erhoben, ständig erweitert und in größeren Abständen auch in ihrer Struktur verändert (Daten- und Schema-Evolution). Auf den Forschungsdaten werden verschiedene Auswertungen vorgenommen (Data-Science-Anwendungen), deren Ergebnisse publiziert werden. Mit Hilfe von ***why***- und ***how***-Provenance soll ermittelt werden, welcher Teil der Primärforschungsdaten zu Zwecken der Replizierbarkeit der Auswertungen langfristig aufgehoben werden muss. Dabei sollen Änderungen auf Daten und Schemata rückberechnet werden, so dass alte Zustände von Daten nicht komplett als Replikat archiviert werden müssen. Das Promotionsvorhaben ordnet sich dabei in zwei langfristige Forschungsschwerpunkte des Lehrstuhls Datenbank- und Informationssysteme der Universität Rostock ein.

1 Einordnung in die langfristigen Forschungsschwerpunkte des Lehrstuhls DBIS der Universität Rostock

Der Lehrstuhl Datenbank- und Informationssysteme konzentriert seine Forschung langfristig auf Grundlagenforschung und angewandte Projekte in den vier P-Bereichen Privacy, Performance (Parallelisierung von Datenbankauswertungen), Provenance und Preservation (Nachhaltigkeit von Daten und Datenbankauswertungen), die die vier V-Bereiche von Big-Data-Szenarien (Volume, Velocity, Variety, Veracity) ergänzen sollen.

PArADISE: In der Projektgruppe PArADISE (Privacy AwaRe Assistive Distributed Information System Environment) werden parallele und performante Techniken zur Auswertung von großen Mengen von Sensordaten entwickelt. Ein Schwerpunkt ist hierbei die Transformation von Verfahren maschinellen Lernens in (Folgen von) SQL-Anfragen, um sowohl die Erfüllung definierter Privatheitsansprüche der späteren Nutzer zu erreichen [24] als auch eine Performance-Verbesserung durch Parallelisierung [31, 32]. Als Nebeneffekt ist die SQL-ifizierung der Data-Science-Auswertungen ein Beitrag zur Nachhaltigkeit (und damit Replizierbarkeit) der Auswertungsprozeduren, da SQL-basierte Datenbanksysteme mit hoher Wahrscheinlichkeit noch über Jahrzehnte verfügbar sein werden [11].

METIS: In der Projektgruppe METIS (Management, Evolution, Transformation und Integration von Schemata) werden Verfahren zur Integration, Transformation und Evolution von Schemata und Daten behandelt. Bei vielen Fragestellungen stehen zwar die Schemata der Datenbanken im Vordergrund, es müssen aber immer auch bestehende Instanzen mit abgebildet werden. Wir sprechen daher hier auch von Schema-Instanz-Abbildungen. Stehen die Abbildungen der Instanzen im Vordergrund, so kann man eine klassische Datenbankanfrage oder eine Auswertung von Daten mittels eines Maschinellen-Lern-Verfahrens auch als Schema-Instanz-Abbildung in Form einer Datenbanktransformation auffassen. Auch die Frage der Datenherkunft bei solchen Datenbanktransformationen (Provenance) wird in diesem Projekt untersucht.

Unter dem Begriff *Provenance Management* versteht man die Rückverfolgbarkeit eines Anfrageergebnisses bis zu den relevanten Originaldaten [11]. Dies beinhaltet sowohl den Datensatz an sich (***where*-**Provenance) als auch den zurückgelegten Weg (***why*-** und ***how*-**Provenance).

METIS in PArADISE: Im Projekt METIS in PArADISE [26] werden die Techniken des Provenance Management auf den Bereich der Auswertung von Sensordatenmengen für die Entwicklung von Assistenzsystemen angewendet. Das Promotionsvorhaben soll diese Anwendung von Provenance Management auf das Forschungsdatenmanagement und die zugehörigen Auswertungen von Messdaten erweitern. Als Leitszenario dient eine Kooperation mit dem Leibniz-Institut für Ostseeforschung (IOW), die schon seit knapp einem Jahrzehnt besteht und neben der Konzeption einer langfristigen Forschungsdatenverwaltung auch Fragestellungen des Provenance Managements betrifft [10].

2 Der Anwendungsbereich: Forschungsdatenmanagement

Der Anwendungsbereich für die Erweiterung der Provenance-Management-Techniken unter Daten- und Schema-Evolution ist das *Forschungsdatenmanagement*. Das Ziel besteht hierbei in der Verfolgung und Archivierung von Daten, die bei wissenschaftlichen Projekten, Experimenten oder Beobachtungen erfasst worden sind. Der Weg von der Datenerhebung bis zur Publikation soll so nachvollziehbar, rekonstruierbar und plausibel gehalten werden [11]. Zwei Forschungsziele des Projektes bestehen daher in der Berechnung einer minimalen Teildatenbank der Original-Forschungsdatenbank sowie der Vereinheitlichung von Provenance und Evolution.

3 Problemstellung des Promotionsprojektes

Im Forschungsdatenmanagement soll neben der Auswertungsanfrage und dem -ergebnis selbst auch der verwendete Ausschnitt der Originaldatenbank archiviert werden. Falls sich die Daten oder das Schema der Forschungsdatenbank häufig verändern, müsste man nach jeder auf dem Datenbestand vorgenommenen Auswertung die Originaldatenbank „einfrieren“ und sichern. Um dies zu vermeiden, berechnen wir mit Hilfe von Provenance-Management-Techniken die minimale Teildatenbank, die gesichert werden muss, um das Anfrageergebnis wieder erzeugen zu können. Dazu wollen wir Techniken der ***why***- und ***how***-Provenance mit der Theorie der inversen Schema-Abbildungen für Datenintegration und Datenaustausch kombiniert. Basis für die Transformationen auf den Daten zur Evolution von Schema und Daten als auch für die Transformation, die die Auswertungsanfrage darstellt, ist der CHASE. Damit können wir drei verschiedene Teilgebiete der Theorie zusammenführen.

In Vorarbeiten haben wir bereits erste konkrete Teilergebnisse zur Klassifizierung von inversen Anfragen bei der Berechnung der ***why***-Provenance vorgestellt: Die Berechnung der Inversen nur mit

Hilfe von Ergebnis und Auswertungsanfrage, die Bestimmung der Inversen mit Hilfe zusätzlicher Provenance-Informationen (Provenance-Polynome der ***how***-Provenance) sowie die Definition eines Datenbankausschnitts, der homomorph auf die Originaldatenbank abgebildet werden kann.

Insgesamt sind im Promotionsvorhaben folgende Fragestellungen zu untersuchen bzw. bestehende Provenance-Techniken zu erweitern:

- Berechnung einer minimalen Teildatenbank der Originaldatenbank mit den Forschungsdaten, die eine Wiederholung der Auswertung mit demselben Ergebnis ermöglicht. Die minimale Teildatenbank soll dabei verschiedenen Struktureigenschaften (Randbedingungen) genügen.
- Vereinheitlichung von Provenance Management, Schema und Daten-Evolution: Die Techniken für ***why***- und ***how***-Provenance sollen mit Techniken zu Schemaabbildungen (Datenbanktransformationen oder Auswertungs-Anfragen im Sinne des Forschungsdatenmanagements) und inversen Schemaabbildungen (für die Darstellung der Provenance-Anfragen) kombiniert werden.
- Um die Vereinheitlichung der Theorie dahinter zu erreichen, stellen wir Auswertungs-Anfragen als s-t tgds (source-to-target tuple-generating dependencies) dar, damit der CHASE als Technik angewendet werden kann. Der CHASE ist somit eine Formalisierung der Auswertung des Forschungsdatenbestandes, d.h. der CHASE wird auf die Forschungsdatenbank angewendet und liefert das Auswertungsergebnis. In einem BACKCHASE-Prozess sollen dann die Provenance-Anfragen auf dem Ergebnis der Auswertung integriert werden. Wir können dann bis zu einem gewissen Grade exakte Inverse oder CHASE-Inverse berechnen, ohne weitere Provenance-Informationen speichern zu müssen.
- s-t tgds und CHASE sind bisher für Teile der Relationenalgebra (konjunktive Anfragen) eingehend untersucht und haben für diese Teile auch gute Eigenschaften (Komplexität, Terminierung, Konfluenz). Zu untersuchen ist, wie wir erweiterte Anfrageoperatoren, die bei der datenbanktechnischen Umsetzung von Machine-Learning-Operationen benötigt werden, als Erweiterung der s-t tgds und des CHASE-Prozesses integrieren können. Gruppierung, Aggregate, skalare Funktionen und Rekursion sind dabei die wichtigsten Operationen, die ergänzt werden müssen, um Algorithmen maschinellen Lernens darstellen zu können [32].

Wir werden jetzt die bereits bestehenden Überlegungen für Teile dieser Problemstellung näher erläutern.

Berechnung einer minimalen Teildatenbank

Die berechnete minimale Teildatenbank soll die Ergebnisse der Auswertungsanfrage unter den folgenden Randbedingungen rekonstruieren können:

- Die Tupelanzahl der Originalrelation bleibt erhalten.
- Die Teildatenbank kann homomorph auf das Original abgebildet werden.
- Die Teildatenbank bildet eine intensionale Beschreibung der Teildatenbank.

Konkret stellt sich die Frage, welche zusätzlichen Informationen bei archiviertem Ergebnis und gemerkter Anfrage für die Rekonstruktion der minimalen Teildatenbank benötigt werden. Reicht es, eine minimale Zeugenmenge aufzuheben (***why***-Provenance, [12]) und/oder die zugehörigen Provenance-Polynome zu berechnen (***how***-Provenance, [22])? Oder ist es notwendig, ganze Tupel bzw. Datenbankausschnitte direkt zu speichern?

Die Berechnung einer inversen Anfrage Q^{-1}, mit dessen Hilfe die gesuchte minimale Teildatenbank bestimmen soll, ist abhängig vom Typ der Originalanfrage Q sowie allen zusätzlich gemerkten Informationen. So kann für die Projektion etwa eine *relaxte CHASE-Inverse*, für die Projektion ohne Duplikateliminierung eine *tp-relaxte CHASE-Inverse* und für die Vereinigung eine *ergebnisäquivalente CHASE-Inverse* [4] angegeben werden. Das Kopieren liefert eine *exakte CHASE-Inverse* [19] (siehe Tabelle 7). Die in den Randbedingungen geforderte Homomorphie ist eine sehr starke Eigenschaft, welche im Verlauf weiterer Untersuchungen gezielt abgeschwächt werden soll. Ein erster Ansatz ist die Definition ergebnisäquivalenter CHASE-Inversen. Diese vernachlässigen die Homomorphie gänzlich, was wiederum zu einem starken Informationsverlust führen kann. So können ganze (vielleicht für andere Anfragen notwendige) Tupel verloren gehen, welche bei der Verwendung einer relaxten CHASE-Inversen zumindest als Nulltupel noch erhalten blieben. Andere Ansätze bestünden beispielsweise in der Einführung von Konzepthierarchien oder der Vergröberung der Datenbankbestände, wie es etwa für intensionale Provenance-Antworten notwendig ist [39].

Vereinheitlichung von Provenance und Evolution (nach [6])

Bisherige Provenance-Anfragen Q_{prov} (***why***-, ***where***- und ***how***-Provenance) erfolgen stets auf einer gegebenen festen Datenbank S sowie einer Auswertungs-Anfrage Q. Die Kombination von Provenance mit Schema- und Daten-Evolution soll die Auswertung von Provenance-Anfragen bei sich ändernden Daten und Schemata ermöglichen (siehe Abbildung 1). Mit Hilfe der Inversenbildung $\mathcal{E}^{-1}$ kann wiederum (falls möglich) der neue Datenbankbestand J in das alte Schema übertragen werden. Das mittels CHASE berechnete Auswertungsergebnis soll daher in einem zweiten Schritt, dem BACKCHASE, um eine inverse Provenance-Anfrage Q'_{prov} erweitert werden. Die minimale Teildatenbank I^* (rote gestrichelte Box) wird dann durch das CHASEN der Provenance-Anfrage Q_{prov} in das Anfrageergebnis $K^* \subseteq K$ (grüne Box) bei Hinzunahme notwendiger Provenance-Informationen (bspw. Provenance-Polynome) berechnet.

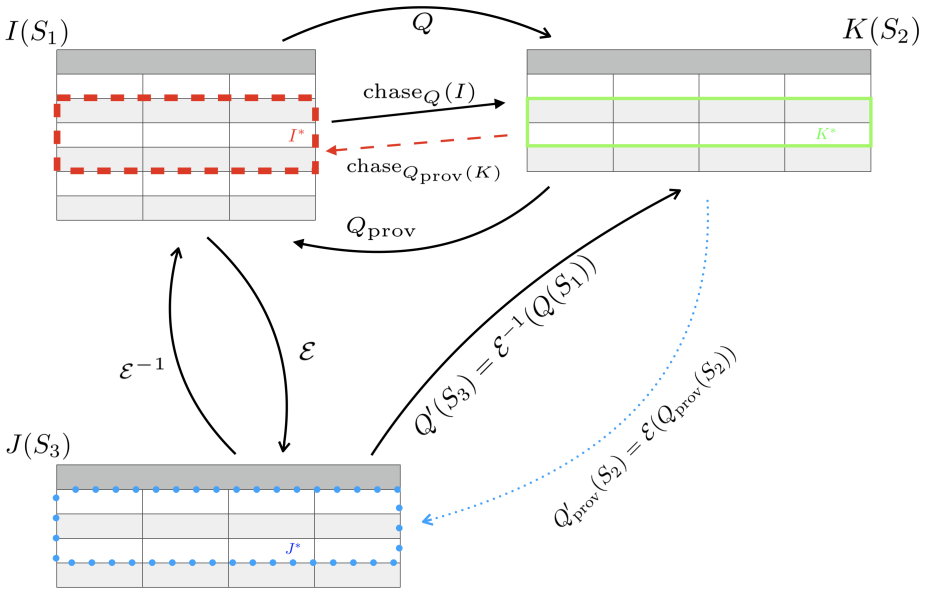


Abbildung 1 Provenance und Evolution mit

- minimalen Teildatenbanken $I^* \subseteq I$, $K^* \subseteq K$, $J^* \subseteq J$
- Q: Auswertungsanfrage
- $\mathcal{E}$: Schemaevolution
- Q_{prov}: Provenance-Anfrage

Formal: Nach der Schemaevolution $\mathcal{E} : S_1 \rightarrow S_3$ kann die Anfrage Q' als Komposition der ursprünglichen Anfrage Q sowie der inversen Evolution $\mathcal{E}^{-1}$ direkt berechnet werden:

$$Q'(J(S_3)) = (Q \circ \mathcal{E}^{-1})(I(S_1)) = \mathcal{E}^{-1}(Q(I(S_1))).$$

Die neue Provenanceanfrage Q'_{prov} ergibt sich analog als

$$Q'_{\text{prov}}(K^*(S_2)) = (\mathcal{E} \circ Q_{\text{prov}})(K^*(S_2)).$$

Es genügt daher, sich eine der beiden minimalen Teildatenbanken $I^*(S_1)$ (rote gestrichelte Box) oder $J^*(S_3)$ (blau gepunktete Box) zu merken. Die jeweils andere lässt sich mit Hilfe der Inversen leicht berechnen. Im Forschungsdatenmanagement entspricht K^* stets der gesamten Ergebnisdatenbank K, d.h. $K^* = K$, da das gesamte Ergebnis der wissenschaftlichen Auswertung reproduzierbar sein muss. Allgemeine Provenance-Anfragen können aber auch auf Teilmengen dieses Ergebnisses oder auf einzelne Tupel im Ergebnis angewendet werden.

Auswertungsanfrage Q: Die Repräsentation der Auswertungsanfrage Q in Form von extended s-t tgds (*source-to-target tuple-generating dependencies*) oder egds (equality-generating dependencies) erlaubt die Anwendung des CHASE-Algorithmus [16, 19]. Dieser arbeitet eine Menge von Abhängigkeiten in eine gegebene Datenbankinstanz ein (siehe Abschnitt 4). Eine Darstellung der Grundoperationen in Form von s-t tgds und egds findet sich im Abschnitt 5.

Evolution $\mathcal{E}$: Durch die Verwendung der Inversen $\mathcal{E}^{-1}$ kann die alte minimale Teildatenbank I^* aus der aktuellen minimalen Teildatenbank J^* berechnet werden. Hierfür werden sowohl die Schemaevolution $\mathcal{E}$ selbst, als auch seine (exakte) Inverse $\mathcal{E}^{-1}$ als s-t tgds und egds formuliert und mit Hilfe des CHASE verarbeitet.

Provenance-Anfrage Q_{prov}: Gegeben sei das mit Hilfe des CHASE berechnete Ergebnis K der Auswertungsanfrage Q. Der anschließende Aufbau der minimalen Teildatenbank I^* gelingt durch Invertierung der Anfrage Q. Die Inverse Q_{prov} muss dabei nicht zwangsläufig einer Inversen im klassischen Sinne

$$Q \circ Q_{\text{prov}} = \text{Id}$$

entsprechen. Grund hierfür ist, dass eine CHASE-Inverse nicht in allen Fällen angegeben werden kann [16, 19]. In den meisten Fällen kann jedoch eine quasi-inverse CHASE-Funktion [7, 4] wie die tp-relaxte oder ergebnisäquivalente CHASE-Inverse angegeben werden, welche nach Anwendung des CHASE auf die Originalinstanz I sowie die mittels BACKCHASE berechnete minimale Teildatenbank I^* das gleiche Ergebnis liefert, d.h.

$$\text{chase}_Q(I) = \text{chase}_Q(I^*).$$

4 Vorgehensweise und Methoden

Der CHASE-Algorithmus ist ein in der Datenbanktheorie universell einsetzbares Werkzeug. Er dient unter anderem der Darstellung von Abhängigkeiten in Datenbanken oder relationenalgebraischen Ausdrücken, der Implikation von Abhängigkeiten, dem Äquivalenznachweis von Datenbankschemata unter gegebenen Abhängigkeiten sowie der Behandlung von Nullwerten in Datenbanken.

Die Idee dieses Algorithmus kann dabei in wenigen Worten zusammengefasst werden: Für ein Objekt $\bigcirc$ (beispielsweise Tableaus) und eine Menge von Abhängigkeiten $\star$ (beispielsweise eine Menge von JDs und FDs) arbeitet der CHASE $\star$ in $\bigcirc$ ein, sodass $\star$ implizit in $\bigcirc$ enthalten ist, $\bigcirc$ also $\star$ nicht verletzt. Bildhaft ergibt sich also:

$$\text{chase}_\star(\bigcirc) = \circledast.$$

Seinen Ursprung hat der CHASE im Jahre 1979. D. Maier et al. entwickeln in ihrem Artikel „Testing implications of data dependencies“ [30] einen Algorithmus, der auf den Tableaudefinitionen von A. V. Aho et al. [2] basiert. Der CHASE kann aber nicht nur auf Tableaus (Datenbankschablonen oder einfache konjunktive Anfragen) und mit einfachen Abhängigkeiten wie FDs und JDs, sondern auch auf Datenbanken und mit tgds/s-t tgds und egds definiert werden. Dies beschreiben unter Anderem Fagin et al. in ihren Artikeln [19] und [17]. Wir gehen auf diese Varianten weiter unten noch genauer ein.

CHASE auf Tableaus: Tableaus dienen unter Anderem der Darstellung von Datenbanken, der Untersuchung von Abhängigkeiten, zur Optimierung relationenalgebraischer Ausdrücke sowie der Äquivalenzuntersuchung von Datenbankschemata. Der CHASE-Algorithmus ermöglicht die Untersuchung dieser Zusammenhänge [29]. So zeigt er die Gültigkeit von Verbundabhängigkeiten und funktionalen Abhängigkeiten, die Tableau-Äquivalenzen unter Integritätsbedingungen, höhere Abhängigkeiten sowie vieles weitere. Der CHASE liefert stets ein neues äquivalentes Tableau

$$T' = \text{chase}_\star(T),$$

welches für die Untersuchung der obigen Zusammenhänge zu Rate gezogen werden kann.

CHASE auf Datenbanken: Der CHASE auf Datenbanken wird üblicherweise für die Datenintegration oder den Datenaustausch verwendet. Hierfür wird eine Menge von tgds und egds (CHASE-Parameter $\star$) — eine Erweiterung der FDs und JDs — in eine Datenbank d (CHASE-Objekt $\bigcirc$) gechaset [17]. Bei der durch

$$d' = \text{chase}_\star(d)$$

entstehenden Datenbank d' kann es sich um dieselbe Datenbank d wie zuvor handeln; die Datenbank würde „upgegradet“ werden. So werden durch egds Tupel angeglichen — Reduzierung von Nullwerten — und durch tgds neue Tupel erzeugt. Betrachten wir zwei Datenbanken, können mittels s-t tgds Tupel auch in der Zieldatenbank erzeugt werden.

CHASE — Formal: Fagin definiert den CHASE allgemein auf tgds und egds [16]. Eine *tuple-generating dependency (tgd)* ist eine Sequenz der Form

$$\forall \mathbf{x} : (\phi(\mathbf{x}) \rightarrow \exists \mathbf{y} : \psi(\mathbf{x}, \mathbf{y}))$$

mit Konjunktionen $\phi(\mathbf{x})$ und $\psi(\mathbf{x}, \mathbf{y})$ von Atomen aus $\mathbf{x}$ bzw. $\mathbf{x}$ und $\mathbf{y}$. Ist $\phi(\mathbf{x})$ Konjunktion von Atomen über einem Quellschema S und $\psi(\mathbf{x}, \mathbf{y})$ Konjunktion von Atomen über einem Zielschema T, so heißt eine solche Sequenz auch *source-to-target tuple-generating dependency (s-t tgd)*. Eine *equality-generating dependency (egd)* ist für zwei Variablen x_1, x_2 aus x definiert durch

$$\forall \mathbf{x} : (\phi(\mathbf{x}) \rightarrow (x_1 = x_2)).$$

Seien I_1 und I_2 zwei Instanzen über einem Datenbankschema $S = \{R_1, \dots, R_n\}$ mit Variablen aus $\underline{\text{Const}} \cup \underline{\text{Nulls}}$ für eine Konstantenmenge $\underline{\text{Const}}$ und $\underline{\text{Nulls}} = \{n_i \mid i \in \mathbb{N}\}$ (*(markierte) Nullwerte*). Dann heißt die Abbildung $h : I_1 \to I_2$ mit

- $\forall c \in \underline{\text{Const}} : h(c) = c$
- $\forall t \in I_1(\mathcal{R}_i(A_1, \dots, A_k)) : h(t) \in I_2(\mathcal{R}_i(h(A_1, \dots, A_k)))$

Homomorphismus von I_1 nach I_2. Existiert zudem ein zweiter Homomorphismus $h' : I_2 \to I_1$, heißt I_1 *äquivalent zu* I_2.

Der *CHASE* einer Instanz I über einer Abhängigkeitsmenge Σ ist eine (endliche oder unendliche) Sequenz

$$I_i \overset{\delta_i, h_i}{\to} I_{i+1}, \quad (i = 0, 1, \dots ; I_0 = I)$$

von Regeln, die diese Abhängigkeiten $\delta_i \in \Sigma$ verarbeiten. Die tgd-Regel erzeugt zusätzliche Tupel; sie entspricht somit der Ergänzung der Quelldatenbank gemäß der tgd durch entsprechende neue Tupel. Die egd-Regel hingegen ersetzt alle (markierten) Nullwerte durch entsprechende Konstanten oder (markierte) Nullwerte mit kleinerem Index. Existiert ein $n \in \mathbb{N}$ mit $I_n \models \Sigma$ oder $I_n = \ \perp$, so spricht man von einem *erfolgreichen* bzw. einem *scheiternden CHASE*.

- *tgd-Regel:* Seien I_i eine Instanz, $\delta : \phi(\mathbf{x}) \to \exists \mathbf{y} : \psi(\mathbf{x}, \mathbf{y})$ eine tgd und $h : \phi(x) \to I_i$ ein Homomorphismus. Kann h nicht zu einem Homomorphismus $h' : \phi(\mathbf{x}) \wedge \psi(\mathbf{x}, \mathbf{y}) \to I_i$ erweitert werden, ist δ mittels h anwendbar auf I_i und es gilt $I_{i+1} = I_i \cup h'(\psi(\mathbf{x}, \mathbf{y}))$. Dabei ist jede Variable in $\mathbf{y}$ ein (markierter) Nullwert.
- *egd-Regel:* Gegeben seien eine egd $\delta : \phi(\mathbf{x}) \to (x_1 = x_2)$, eine Instanz I und ein Homomorphismus $h : \phi(\mathbf{x}) \to I$ mit $h(x_1) \neq h(x_2)$. Dann gilt:
  - $h(x_1), h(x_2) \in \underline{\text{Const}}$: Die Anwendung von δ auf I liefert kein Ergebnis; es gilt $I_n = \ \perp$.
  - $h(x_i) \in \underline{\text{Const}} \wedge h(x_j) \in \underline{\text{Nulls}}$ mit $i \neq j$ und $i, j \in \{1, 2\}$: Ersetze die (markierte) Nullvariable $h(x_j)$ durch die Konstante $h(x_i)$.
  - $h(x_1)$, $h(x_2) \in \underline{\text{Nulls}}$: Setze $h(x_1)$ und $h(x_2)$ auf die selbe markierte Nullvariable $h(x_1)$ oder $h(x_2)$.

Beispiel nach [7]: Zur Veranschaulichung des CHASE sei exemplarisch die Datenbank einer Universität gegeben. Diese enthält unter Anderem eine Relation mit den Namen, Studiengängen und Matrikelnummern ihrer Studenten sowie eine weitere Relation, welche alle abgeschlossenen Abschlussarbeiten listet. Die zugehörigen Schemata sind definiert als

STUD(matrikelnummer, name, studiengang)

und

AA(id, titel, vertiefung, matrikelnummer, note).

Die Relation APS mit dem Schema

APS(id, titel, matrikelnummer, studiengang),

welche die Abschlussarbeiten pro Studiengang zusammenfasst, soll nun in die Relationen STUD und AA migriert werden. Hierzu seien folgende Abhängigkeiten gegeben (eine s-t tgd, eine tgd sowie eine egd):

$$
\begin{aligned}
\Sigma_{\mathrm{ST}} \;=\; & \{\delta_1 : \mathrm{APS}(\mathrm{id}, \mathrm{titel}, \mathrm{matrikelnummer}, \mathrm{studiengang}) \\
& \quad \rightarrow \exists\ \ni_{\mathrm{Name}}, \ni_{\mathrm{Vertiefung}}, \ni_{\mathrm{Note}} \colon \mathrm{STUD}(\mathrm{matrikelnummer}, \ni_{\mathrm{Name}}, \mathrm{studiengang}) \\
& \qquad \wedge\ \mathrm{AA}(\mathrm{id}, \mathrm{titel}, \ni_{\mathrm{vertiefung}}, \mathrm{matrikelnummer}, \ni_{\mathrm{note}})\} \\
\Sigma_{\mathrm{T}} \;=\; & \{\delta_2 : \mathrm{AA}(\mathrm{id}, \mathrm{titel}, \mathrm{vertiefung}, \mathrm{matrikelnummer}, \mathrm{note}) \\
& \quad \rightarrow \exists\ \ni_{\mathrm{name}}, \ni_{\mathrm{studiengang}} \colon \mathrm{STUD}(\mathrm{matrikelnummer}, \ni_{\mathrm{Name}}, \ni_{\mathrm{studiengang}}), \\
& \quad \delta_3 : \mathrm{STUD}(\mathrm{matrikelnummer}, \mathrm{name}_1, \mathrm{studiengang}) \\
& \qquad \wedge\ \mathrm{STUD}(\mathrm{matrikelnummer}, \mathrm{name}_2, \mathrm{studiengang}) \rightarrow \mathrm{name}_1 = \mathrm{name}_2\}
\end{aligned}
$$

Der CHASE liefert nun, angewandt auf den Datenbestand aus Tabelle 1, nach Integration der s-t tgd δ_1 die Tupelmenge

$$
\begin{aligned}
I_1 \;=\; & \{\mathrm{STUD}(4711, \eta_{\mathrm{name}_1}, \mathrm{Informatik}), \\
& \mathrm{AA}(1234, \mathrm{Big\ Data}, \eta_{\mathrm{vertiefung}_1}, 4711, \eta_{\mathrm{note}_1}), \\
& \mathrm{STUD}(4711, \eta_{\mathrm{name}_2}, \mathrm{Informatik}), \\
& \mathrm{AA}(2467, \mathrm{Provenance}, \eta_{\mathrm{vertiefung}_2}, 4711, \eta_{\mathrm{note}_2})\}
\end{aligned}
$$

mit den (markierten) Nullwerten $\eta_{\mathrm{Name}_1}, \eta_{\mathrm{Name}_2}, \eta_{\mathrm{Vertiefung}_1}, \eta_{\mathrm{Vertiefung}_2}, \eta_{\mathrm{Note}_1}$ und η_{Note_2}. Bevor δ_2 sowie δ_3 eingearbeitet werden, muss geprüft werden, ob I_1 die Abhängigkeiten aus Σ_{T} bereits erfüllt, d.h. $I_1 \models \Sigma_{\mathrm{T}}$. In diesem Fall wäre ein weiterer *CHASE-Schritt*, also die Anwendung einer tgd- oder egd-Regel, nicht mehr notwendig und der CHASE selbst erfolgreich. Die tgd δ_2 ist bereits erfüllt, sodass die Eliminierung überflüssiger Nullwerte — das sogenannte *cleaning* — mittels δ_3 bereits die Ergebnisinstanz

$$
\begin{aligned}
I_2 \;=\; & \{\mathrm{STUD}(4711, \eta_{\mathrm{name}_1}, \mathrm{Informatik}), \\
& \mathrm{AA}(1234, \mathrm{Big\ Data}, \eta_{\mathrm{vertiefung}_1}, 4711, \eta_{\mathrm{note}_1}), \\
& \mathrm{AA}(2467, \mathrm{Provenance}, \eta_{\mathrm{vertiefung}_2}, 4711, \eta_{\mathrm{note}_2})\}
\end{aligned}
$$

liefert. Implementiert ist diese Art des CHASE etwa in der Software **Llunatic** [20].

<u>id</u>	titel	matrikelnummer	studiengang	
1234	Big Data	4711	Informatik	t_1
2467	Provenance	4711	Informatik	t_2

Tabelle 1 APS-Relation mit zugehörigen Tupelidentifikatoren t_1 und t_2

5 Stand der Forschung

Die beiden Hauptschwerpunkte dieser Dissertation bilden die Themen *Data Provenance* und *inverse Schemaabbildungen*. Der Stand der aktuellen Forschung in diesen Gebieten sei hier kurz angerissen.

Data Provenance

In der Provenance-Theorie werden typischerweise vier Provenance-Anfragen und vier Provenance-Antworten (siehe Tabelle 2) unterschieden [13]. Die Anfragen ***where*** (Woher kommen die Daten?), ***why*** (Warum dieses Ergebnis?) und ***how*** (Wie kommt das Ergebnis zustande?) können dabei nach ihrem Informationsgehalt geordnet werden:

$$\boldsymbol{where} \preceq \boldsymbol{why} \preceq \boldsymbol{how}.$$

Anfrage-Typ	Fragestellung
where	Woher kommen die Daten?
why	Warum dieses Ergebnis?
how	Wie kommt das Ergebnis zustande?
why not	Warum fehlt ein bestimmtes Element im Ergebnis?

Antwort-Typ	Ergebnis
extensional	Tupel aus den Originaldaten
intensional	Beschreibung der Daten
anfragebasiert	Selektionsprädikate
modifikationsbasiert	Vorschlag zur minimalen Änderung der Auswertungsanfrage

Tabelle 2 Provenance-Anfragen und -Antworten

Mit anderen Worten, aus dem Ergebnis der ***how***-Provenance können die ***why***- sowie die ***where***-Provenance abgeleitet werden. Gleiches gilt für die ***why***- und die ***where***-Provenance (siehe Abbildung 2). Der formale Beweis für die Reduktion der ***why***- auf die ***how***-Provenance kann in [13] nachgelesen werden. Die ***why not***-Anfrage kann in diese „Informationskette“ allerdings nicht eingeordnet werden, da sie, anders als die ***where***-, ***why***- oder ***how***-Anfragen, modifikationsbasierte Antworten liefern.

where-Provenance (Tabellenname)	***why***-Provenance (Zeugenbasis)	***how***-Provenance (Polynom)
APS	$\{\{t_1\},\{t_2\},\{t_1,t_2\}\}$	$(t_1 \cdot (t_1 + t_2)) + (t_2 \cdot (t_1 + t_2))$

Abbildung 2 Reduktion von ***why***, ***where*** und ***how***

Kann eine Art Formel — das sogenannte *Provenance-Polynom* [22] — zur Berechnung des Anfrageergebnisses angegeben werden, so ist es möglich, den Datensatz auf die „relevanten“ Datensätze einzuschränken und dennoch ein korrektes Anfrageergebnis wiederzugeben. Die Roh- bzw. Quelldatenmasse kann durch die Angabe einer (*minimalen*) *Zeugenbasis* [12] so erheblich reduziert werden. Der Antworttyp ist dabei extensional, d. h. die Antwort auf eine Provenance-Anfrage ist stets eine Menge von (auf bestimmte Attribute eingeschränkten) Tupeln der Originaldaten. Für intensionale, anfragebasierte und modifikationsbasierte Antworten sei etwa auf [28] oder [39] verwiesen.

Beispiel: Für die Relation APS (siehe Tabelle 1) sowie die Anfrage

$$\pi_{\text{matrikelnummer}}(\pi_{\text{id,studiengang}}(\text{APS}) \bowtie \pi_{\text{matrikelnummer,studiengang}}(\text{APS}))$$

ergeben sich nach Tannen [23] folgende Provenancen:

- ***where***-Provenance:
  - Name der am Ergebnis beteiligten Tabelle: APS
  - Menge der am Ergebnis beteiligten Tupel: $\{t_1, t_2\}$
- ***why***-Provenance:
  - Zeugenbasis: $\{\{t_1\},\{t_2\},\{t_1,t_2\}\}$
  - minimale Zeugenbasis: $\{t_1\}$ oder $\{t_2\}$

- ***how*-Provenance:**
  - Provenance-Polynom: $(t_1 \cdot (t_1 + t_2)) + (t_2 \cdot (t_1 + t_2))$
    mit $+$ für Alternativen (Projektion π und Vereinigung $\cup$) und $\times$ für Kombinationen wie der natürliche Verbund $\bowtie$

Zeugenbasis: Cui et al. [14] assoziieren jedes Ergebnistupel mit einer Menge von Tupeln der Ursprungstabelle, den sogenannten *Zeugen*. Die Menge aller Zeugenmengen entspricht der *Zeugenbasis* [12]. Die Frage nach dem „Warum?“ kann also mit der Angabe einer Zeugenbasis beantwortet werden. Die zur Beantwortung einer Anfrage minimal notwendigen Tupel werden in einer *minimalen Zeugenbasis* zusammengefasst. Diese ist nicht zwangsläufig eindeutig. Eine direkte Ableitung einer minimalen Zeugenbasis aus der Zeugenbasis ist uns bisher nicht bekannt.

Provenance-Polynome: Die in [22] definierten Provenance-Polynome entsprechen Polynomen über einem Halbring $(\mathbb{N}[\tau], +, \cdot, 0, 1)$, wobei τ einer Menge von Tupelidentifikatoren entspricht. Sie verarbeiten die Selektion und Projektion, den natürlichen Verbund sowie die Vereinigung. Außerdem wurden sie bereits um die Aggregation sowie Gruppierung erweitert [3]. Die Theorie der Provenance-Polynome basiert auf einer positiv relationalen Algebra. Die Selektion auf Ungleichheit sowie die Differenzbildung zählen aber nicht zu dieser Algebra. Die Aufstellung der zugehörigen Polynome ist daher mit einigen Schwierigkeiten verbunden. Erste Ansätze hierzu liefern etwa die Artikel *Provenance for Aggregate Queries* [3] und *The Semiring Framework for Database Provenance* [23].

Für die Angabe einer extensionalen Provenance-Antwort genügt somit die Berechnung der ***how*-Provenance**. Die Berechnung und Speicherung dieser Polynome erfolgt etwa über die schrittweise Verfolgung der algebraischen Grundoperationen der Anfrage oder über graphentheretische Ansätze. Im Bereich der Spieltheorie existieren für die Beantwortung der ***why*-**, ***why not*-** oder ***how*-Provenance** beispielsweise die sogenannten *Provenance Games* [27, 28].

Inverse Schemaabbildungen

Die für die Rekonstruktion einer (minimalen) Teildatenbank nötigen inversen Schemaabbildungen werden zwar bzgl. ihrer Eigenschaften untersucht und im Bereich der Schema-Evolution erfolgreich angewendet [16], können aber auch in der Datenherkunft von Nutzen sein. Ein offenes Problem im Bereich der Rekonstruierbarkeit besteht nun in der Berechnung dieser inversen Abbildungen durch „Rettung“ von Annotationen. Eine Schemaabbildung $\mathcal{M} = (S_1, S_2, \Sigma_1)$ ist eine Abbildung der Datenbankinstanzen [38]. Die zugehörige inverse Schemaabbildung $\mathcal{M}^{-1} = (S_2, S_1, \Sigma_2)$ mit $\mathcal{M} \circ \mathcal{M}^{-1} = \mathrm{Id}$ entspricht dann der Rückabbildung der Zielinstanz S_2 auf die Quellinstanz S_1 bzgl. einer gegebenen Menge von Abhängigkeiten Σ_1.

Exakte, klassische und relaxte CHASE-Inverse: Inverse Abbildungen finden etwa bei der Schemaevolution Anwendung. Während eine *exakte CHASE-Inverse* stets die Originaldatenbank rekonstruiert, liefert die *klassische CHASE-Inverse* nur ein zur Originaldatenbank äquivalentes Ergebnis [19]. Kann eine Inverse nicht angegeben bzw. definiert werden, wird aber dennoch eine Art inverse Abbildung benötigt, kann auf die sogenannte Quasi-Inverse zurückgegriffen werden [18]. Sie liefert nicht mehr die Quellinstanz selbst, dafür aber eine hierzu äquivalente Instanz oder zumindest eine äquivalente Teilinstanz. Ein Beispiel hierfür wäre die in [19] eingeführte *relaxte CHASE-Inverse*.

CHASE-Szenarien: Als Werkzeug soll der CHASE-Algorithmus verwendet werden. Er ist ein Universalwerkzeug, welches eine Menge von Abhängigkeiten $\star$ in ein gegebenes Objekt $\bigcirc$ einarbeitet. In diesem Fall bedeutet dies die Einarbeitung von tgds und egds in eine gegebene Datenbank (Fall II. in Tabelle 3). Die Definition des klassischen CHASE (Fall I.) stammt von D. Maier aus dem Jahr 1979 [30], bei dem als Abhängigkeiten zunächst funktionale Abhängigkeiten (FDs) und Verbundabhängigkeiten (Join Dependencies, JDs) in eine Tableauanfrage eingearbeitet wurden. Mit *Answering Queries using Views* (AQuV) beschäftigt sich unter anderem [25].

	$\star$	$\bigcirc$	Ziel
I.	FDs und JDs	Anfrage in Form von Tableaus	Semantische Optimierung
II.	Sicht	Anfrage	AQuV
III.	s-t tgds, egds	Quelldatenbank	Datenaustausch, Datenintegration
VI.	s-t tgds, egds, tgds	Datenbank	Invertierte Anfrage

Tabelle 3 Überblick über einige CHASE-Varianten (Ausschnitt der Tabelle 6)

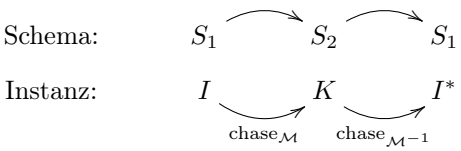
In unserem Anwendungsfall VI. soll der CHASE eine Datenbank (das CHASE-Objekt $\bigcirc$) durch eine Datenbanktransformation ($\star$, als st-tgds dargestellt) in ein Auswertungsergebnis überführen. Durch einen nachfolgenden BACKCHASE-Prozess (siehe unten) soll nun die Auswertungsanfrage invertiert werden, um Provenance-Informationen (etwa Zeugen) zu erhalten. Ist dies nicht ohne Zusatzinformationen (wie Provenance-Polynome) möglich, so sollen diese Zusatzinformationen statt der vollständigen Originaldatenbank gespeichert werden.

CHASE-Inverse auf Datenbanken: Für eine Anfrage Q und eine Quellinstanz I liefert der CHASE eine Ergebnisinstanz $K = \text{chase}_{\mathcal{M}}(I)$, wenn gilt:

- Q wird als Schemaabbildung $\mathcal{M} = (S_1, S_2, \Sigma_1)$ mit Quell- und Zielschema S_1 und S_2 sowie einer Menge von Abhängigkeiten Σ_1 aufgefasst;
- I ist Quellinstanz in S_1.
- K ist Zielinstanz in S_2.

Die *(wiederhergestellte) minimale Teildatenbank* $I^* = \text{chase}_{\mathcal{M}^{-1}}(K) = \text{chase}_{\mathcal{M}^{-1}}(\text{chase}_{\mathcal{M}}(I))$ ist somit das Ergebnis einer Rückanfrage Q' auf die Ergebnisinstanz K. Dabei enthält I^* ganze oder auf bestimmte Attribute des Quellschemas eingeschränkte (und mit (markierten) Nullwerten aufgefüllte) Tupel aus I.

CHASE&BACKCHASE:



Das CHASE&BACKCHASE-Verfahren zur Bestimmung einer CHASE-inversen Schemaabbildung $\mathcal{M}^{-1} = (S_2, S_1, \Sigma_2)$ zu $\mathcal{M} = (S_1, S_2, \Sigma_1)$ lässt sich daher wie folgt beschreiben:

- CHASE: Berechne den CHASE von I bzgl. $\mathcal{M}$ als Sequenz von s-t tgd- und egd-Regel
- BACKCHASE: Berechne den CHASE von I' bzgl. $\mathcal{M}^{-1}$ als Sequenz von s-t tgd- und egd-Regeln.

Enthält die wiederhergestellte Instanz $I^* \subseteq I$ Tupel mit (markierten) Nullwerten, deren restliche Attributwerte mit den Attributwerten eines Tupels der Quellinstanz I übereinstimmen, beispielsweise $(1, 3, \eta_1) \in I$ und $(1, 3, 5) \in I^*$, so heißt I^* *Ausschnitt* der Instanz I, kurz $I^* \preceq I$. Mit anderen Worten, es existiert ein Homomorphismus h, welcher die Tupel aus I^* auf die Tupel aus I abbildet. Für das obige Beispiel: $1 \mapsto 1, 3 \mapsto 3$ und $\eta_1 \mapsto 5$. Enthält I^* keine (markierten) Nullwerten und sind alle Tupel aus I^* auch Tupel in I. Schreibweise: $I^* \subseteq I$.

Tp-relaxte und ergebnisäquivalente CHASE-Inverse: Die *relaxte CHASE-Inverse* ist eine Abschwächung der klassischen CHASE-Inversen. Sie fordert keine Äquivalenzbeziehung zwischen Quellinstanz und minimale Teildatenbank I^*, dafür aber Ergebnisäquivalenz [19] sowie die Existenz eines Homomorphismus von der minimalen Teildatenbank I^* in die Quellinstanz I. Um die Tupelzahl der Originaldatenbank beizubehalten, erweitern wir die Definition der *relaxten CHASE-Inversen* zur sogenannten *tupel-erhaltenden CHASE-Inversen* (engl. tuple-preserving CHASE-inverse, [8]). Eine Abschwächung hingegen ist die *ergebnisäuqivalente CHASE-Inverse*. Eine Schemaabbildung $\mathcal{M} = (S_1, S_2, \Sigma_1)$ heißt *ergebnisäquivalent* bzgl. $\mathcal{M}$ (kurz: $I \leftrightarrow_{\mathcal{M}} J$), wenn für zwei Instanzen I, J über S gilt:

$$\text{chase}_{\mathcal{M}}(I) \equiv \text{chase}_{\mathcal{M}}(J).$$

Die ergebnisäquivalente CHASE-Inverse erfordert nur die Ergebnisäquivalenz und keinen zusätzlichen Homomorphismus von I^* nach I und ist damit die schwächste CHASE-Inversen (siehe Abschnitt 6).

6 Eigene Vorarbeiten

Die Definition der *tp-relaxten* und *ergebnisäquivalenten CHASE-Inversen* sowie die konkrete Angabe der CHASE-Inversen basiert auf den in [4] vorgestellten Ideen. Diese wurden im Laufe des ersten Promotionsjahre in einigen Veröffentlichungen [6, 8] stetig weiterentwickelt. Auch das Promotionsvorhaben an sich wurde in einer eigenen Veröffentlichung [7] vorgestellt.

Tp-relaxte und ergebnisäquivalente CHASE-Inverse: Die *tp-relaxte CHASE-Inverse* entspricht der relaxten CHASE-Inversen mit der zusätzlichen Forderung nach Erhalt der Tupelanzahl. Mit anderen Worten, es gilt Ergebnisäquivalenz und Erhalt der Tupelanzahl. Die *ergebnisäquivalente CHASE-Inverse* fordert nur noch Ergebnisäquivalenz und ist somit die schwächste CHASE-Inverse. Insgesamt ergibt sich die Reduktion:

$$\begin{aligned} \text{ergebnisäquivalente CHASE-Inverse} &\preceq \text{relaxte CHASE-Inverse} \\ &\preceq \text{tp-relaxte CHASE-Inverse} \\ &\preceq \text{klassische CHASE-Inverse} \\ &\preceq \text{exakte CHASE-Inverse.} \end{aligned}$$

Hinreichende und notwendige Bedingung: Diese Reduktion bildet die hinreichende Bedingung für die Existenz einer CHASE-Inversen. Die notwendige Bedingung folgt aus der jeweiligen Definition der CHASE-Inversen. Tabelle 4 fasst diese Bedingungen zusammen. Die Existenz einer exakten CHASE-Inversen ist somit hinreichende Bedingung für die Existenz einer CHASE-Inversen. Notwendige Bedingungen sind die Gleichheit von Ergebnisinstanz I und wiederhergestellter minimalen Teildatenbank I^* ($I^* = I$), Äquivalenz der beiden Instanzen ($I^* \leftrightarrow I$), der Test auf Existenz eines Homomorphismus (I^* ist Ausschnitt von I, $I^* \subseteq I$), der Erhalt der Tupelanzahl ($|\, I^* \,| = |\, I \,|$) sowie Ergebnisäquivalenz ($I^* \leftrightarrow_{\mathcal{M}} I$).

CHASE-Inverse	Hinreichende Bedingung	Notwendige Bedingung
Exakte CHASE-Inverse	-	$I^* = I$
Klassische CHASE-Inverse	Exakte CHASE-Inverse	$I^* \leftrightarrow I$
Tp-relatxte CHASE-Inverse	Exakte CHASE-Inverse	$I^* \preceq I, \mid I^* \mid = \mid I \mid$
Relaxte CHASE-Inverse	Klassische CHASE-Inverse	$I^* \preceq I$
Ergebnisäquivalente CHASE-Inverse	Relaxte CHASE-Inverse	$I^* \leftrightarrow_{\mathcal{M}} I$

Tabelle 4 Hinreichende und notwendige Bedingung für die Existenz von CHASE-Inversen

CHASE-Inverse der Relationenalgebra-Grundoperation: Die Untersuchung der Grundoperationen — Kopieren, Umbenennung, Projektion, natürlicher Verbund und Selektion, die Mengenoperationen (Vereinigung, Schnitt, Differenz), die klassischen Aggregatfunktionen (Minimum, Maximum, Anzahl, Summe, arithmetisches Mittel), Gruppierung mittels Group By sowie die arithmetischen Operationen (Multiplikation, Division, Addition und Subtraktion von Skalaren, also Ein-Variablen-Ausdrücke, sowie Modulorechnung) — liefert zudem eine Antwort auf die Frage, welche Informationen für die Rekonstruktion der Quelldatenbank notwendig sind. So können für diese Operationen bis auf wenige Ausnahmen (Selektion auf Ungleichheit, Differenzbildung) exakte, tp-relaxte, relaxte oder ergebnisäquivalente CHASE-Inverse angegeben werden (siehe Tabelle 5). Die Existenz einer exakten CHASE-Inversen kann etwa für die Umbennenung oder das Kopieren angegeben werden. Die Hinzunahme zusätzlicher Provenance-Informationen ermöglicht die Angabe stärkerer CHASE-Inversen. So kann im Falle der Vereinigung statt einer ergebnisäquivalenten CHASE-Inversen durch die Verwendung von Data Provenance sogar eine exakte CHASE-Inverse konstruiert werden. Eine Erweiterung ist allerdings nicht immer möglich: Für die Selektion auf Ungleichheit sowie die Differenzbildung können trotz Provenance keine CHASE-Inversen angegeben werden.

Angabe einer CHASE-Inversen: Eine konkrete Angabe der inversen Abbildung $\mathcal{M}^{-1}$ ist für die Grundoperationen ebenfalls möglich. Die ergebnisäquivalente CHASE-Inverse entspricht stets der Identitätsabbildung. Die tp-relaxte CHASE-Inverse kann neben den (markierten) Nullwerten sogar ganze Nulltupel (Aggregatfunktion COUNT) erzeugen. Der natürliche Verbund, die arithmetischen Ein-Variablen-Operationen $+, -, \cdot$ und $:$ sowie die Umbenennung lassen sich durch die Projektion, die arithmetischen Operationen $-, +, :$ und $\cdot$ sowie die Rückbenennung invertieren (siehe Tabelle 5).

Komposition von Inversen Auch Kompositionen dieser Grundoperationen sowie die Aggregation oder Gruppierung können auf diese Weise untersucht werden. Erste Ansätze hierzu finden sich in [4]. Für die Komposition $\mathcal{M} = \mathcal{M}_1 \circ \ldots \circ \mathcal{M}_n$ der obigen Operationen, ergibt sich die inverse Funktion

$$\mathcal{M}^{-1} = (\mathcal{M}_1 \circ \ldots \circ \mathcal{M}_n)^{-1} = \mathcal{M}_n^{-1} \circ \ldots \circ \mathcal{M}_1^{-1}$$

als Komposition der inversen Teiloperationen $\mathcal{M}_1^{-1}, \ldots, \mathcal{M}_n^{-1}$. Der Typ der Inversen $\mathcal{M}^*$ entspricht dabei dem Typ der schwächsten Teilinversen $\mathcal{M}_i^{-1}$ $(i = 1, \ldots, n)$. Das Wissen über die Provenance des Anfrageergebnisses reicht in einigen Fällen für eine vollständige Rekonstruktion jedoch nicht aus, wobei bei manchen Anfragetypen eine vollständige Rekonstruktion vielleicht gar nicht gefordert ist.

7 Weitere Vorarbeiten am Lehrstuhl

Als Vorarbeit für diese Dissertation dienen die Bachelorarbeit *Intensional Answers for Provenance Queries in Big Data Analytics* [39], die Masterarbeiten *Umsetzung von Provenance-Anfragen in Big-Data-Analytics-Umgebungen* [4] und *CHASE und BACKCHASE: Entwicklung eines Universal-*

Operation $\mathcal{M}$	ohne Provenance		mit Provenance
	Inversen-Typ	Inverse $\mathcal{M}^{-1}$	Inversen-Typ
$r(\mathcal{R})$	$=$	Id	$=$
$\beta_{A_j \leftarrow A_i}(r(\mathcal{R}))$	$=$	$\beta_{A_i \leftarrow A_j}$	$=$
$\pi_{A_i}(r(\mathcal{R}))$	$\preceq_{\text{tp}}$	Nullwerterweiterung in A_i	$\preceq_{\text{tp}}$
	$\preceq$	Nullwerterweiterung in A_i	$\preceq_{\text{tp}}$
$r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$	$=$	Projektion auf Attribute aus $\mathcal{R}_i$	$=$
	$\preceq$	Projektion auf Attribute aus $\mathcal{R}_i$	$\preceq$
$\sigma_{A_i \theta c}(r(\mathcal{R}))$	$\preceq$	Id	$\preceq$
$\sigma_{A_i \theta A_j}(r(\mathcal{R}))$ mit $\theta \in \{<, \leq, =, \geq, >\}$	$\preceq$	Id	$\preceq$
$\sigma_{A_i \neq c}(r(\mathcal{R}))$	xxx	xxx	xxx
$\sigma_{A_i \neq A_j}(r(\mathcal{R}))$	xxx	xxx	xxx
$r_1(\mathcal{R}_1) \cup r_2(\mathcal{R}_2)$	$\leftrightarrow$	Id	$=$
$r_1(\mathcal{R}_1) \cap r_2(\mathcal{R}_2)$	$\preceq$	Id	$\preceq$
$r_1(\mathcal{R}_1) - r_2(\mathcal{R}_2)$	xxx	xxx	xxx
$\mathbf{MAX}_{A_i}(r(\mathcal{R}))$ / $\mathbf{MIN}(r(\mathcal{R}))$	$\preceq$	Id	$\preceq$
$\mathbf{COUNT}_{A_i}(r(\mathcal{R}))$	$\preceq_{\text{tp}}$	Erzeugen von Nulltupeln	$\preceq_{\text{tp}}$
$\mathbf{SUM}_{A_i}(r(\mathcal{R}))$	$\leftrightarrow$	Id	$=$
$\mathbf{AVG}_{A_i}(r(\mathcal{R}))$	$\leftrightarrow$	Id	$=$

Tabelle 5 Basisoperationen und ihre exakten ($=$), tp-relaxten ($\preceq_{\text{tp}}$), relaxten ($\preceq$) und ergebnisäquivalenten ($\leftrightarrow$) CHASE-Inverse (Ausschnitt der Tabelle 7)

Werkzeugs für eine Basistechnik der Datenbankforschung, die Bachelorarbeit *Automatische Generierung von Domänengeneralisierungshierarchien* [15] sowie die Arbeiten der Studentengruppen *KSWS: Datenbanken* [37] des Wintersemesters 2017/2018 und *NEidI: Datenbanken* [5] des Sommersemesters 2016.

Intensional Answers for Provenance Queries in Big Data Analytics: Provenance-Antworten auf sehr große Datensätze können manchmal nur sehr schwer interpretiert werden. Der in dieser Arbeit entwickelte dreistufige Algorithmus liefert intensionale, d. h. beschreibende Antworten, sodass der Anwender sich an das gewünschte Ergebnis „herantasten“ kann.

KSWS — Datenbanken: Röger und Renn sichteten bereits existierende CHASE-Tools und testeten die Programme **Llunatic**[*] [20] und **PDQ**[†] [9] auf ihre Anwendbarkeit für den Datenaustausch (*data exchange*) sowie die Anfrageoptimierung (*query reformulation*).
Llunatic ist ein allgemeines Framework für die Datenreinigung und Schemaabbildung. Es bietet ein universell einsetzbares CHASE-Tool, welches tgds und egds ausführen kann. Es verarbeitet drei verschiedene Szenarien:

- Die traditionelle Datenreinigung mittels egds.
- Der traditionelle Datenaustausch mittels s-t tgds, target tgds und target egds.
- Eine neue Form von Mapping- und Bereinigungsszenarien, in denen Daten von den Quellen zum Ziel verschoben werden und das Ziel auf integrierte Weise nach gegebenen Reinigungsbedingungen repariert wird.

NEidI — Datenbanken: Das Projekt dient dem Verständnis der Konstruktion von Provenance-Polynomen für die Grundanfragen (Projektion, Selektion, Umbenennung, natürlicher Verbund und

[*] `http://db.unibas.it/projects/llunatic/`
[†] `http://www.cs.ox.ac.uk/projects/pdq/home.html`

Vereinigung) sowie aggregierten Anfragen [3]. Es folgen erste Implementierungen eines Tools, welches die ***where-***, ***why-*** und ***how-***Provenance einer gegebenen Grundanfrage bzw. aggregierten Anfrage automatisch berechnet.

Umsetzung von Provenance-Anfragen in Big-Data-Analytics-Umgebungen: Ziel der Arbeit war die Adaption von Techniken der Provenance-Anfragen ***why***, ***where*** und ***how*** in Umgebungen, die statt einfacher Anfragen wie Selektion, Projektion und Verbund auch OLAP-Operationen und weitere Machine-Learning-Algorithmen benutzen. Die ausschließlich extensionalen Provenance-Antworten werden dabei durch Provenance-Polynome sowie (minimalen) Zeugenbasen gegeben. Die Erweiterung des CHASE-Algorithmus für Datenbanken um eine BACKCHASE-Phase zur Provenance-Antwort-Bewertung ermöglicht so die Bestimmung des CHASE-Inversentyps (exakt / relaxt / ergebnisäquivalent) einer gegebenen Anfrage. Die Frage, welche Informationen für die Rekonstruktion der Quelldatenbank notwendig sind, kann so ein Stück weit beantwortet werden.

Automatische Generierung von Domänengeneralisierungshierarchien: Diese Arbeit beschäftigt sich mit der automatischen Erzeugung von Domänengeneralisierungshierarchien, um eine Grundlage für optimale Generalisierungen zu schaffen. Dabei werden bestehende Lösungsansätze zur Anonymisierung von Datensätzen untersucht, sowie Clustering-Verfahren und Ontologien angewendet, um optimale Generalisierungsbäume zu konstruieren. Als Quelle für Ontologien sollen dabei domänenspezifische Repositories herangezogen werden. Diese Arbeit diente primär dem Zweck der datensparsamen Verarbeitung personenbezogener Daten (also dem Ziel Privacy), kann aber auch für die Berechnung intensionaler Antworten von Provenance-Anfragen genutzt werden.

CHASE und BACKCHASE — Entwicklung eines Universal-Werkzeugs für eine Basistechnik der Datenbankforschung: Der Lehrstuhl DIBS unterscheidet vier verschiedene Anwendungsprobleme, welche mittels CHASE gelöst werden können. Hierfür sollte ein grundlegendes CHASE- und BACKCHASE-Werkzeug konzipiert und prototypisch implementiert werden. Dem Werkzeug sollten sowohl Datenbestände als auch Anfragen übergeben werden können. Die Schnittstellen sollen sich dabei an relationale Tabellen und ein Kern-SQL anlehnen. Für die Implementierung konnte sich der bearbeitende Student auf ein Teilproblem (einfache Daten, einfache Korrespondenzen, einfache Anfragen) beschränken. Die erste Version des Tools konnte zum Abschluss der Masterarbeit im Oktober 2018 vorgestellt werden. Das Tool soll durch weitere studentische Arbeiten in 2019 weiterentwickelt werden.

8 Teilziele der Promotion

Innerhalb der Bearbeitung der Dissertation müssen folgende Teilziele erreicht werden:

- Einheitliche Formalisierung von Forschungsdatenbank, Auswertungsanfragen, Provenance-Anfragen, Daten- und Schema-Evolution und weiteren Konzepten auf Basis des relationalen Datenbankmodells inklusive Abhängigkeiten (wie s-t tgds) und Anfrageformalismen (wie erweiterte Relationenalgebren oder Datalog).
- Falls für bestimmte Auswertungsoperationen nur schwächere Typen von Inversen angegeben werden können, Bestimmung der minimalen Zusatzinformationen, um eine Provenance-Anfrage ohne vollständig gespeicherte Originaldatenbank zu ermöglichen.
- Es ist zu ermitteln, wie diese Zusatzinformationen möglichst effizient gespeichert bzw. möglichst effizient für Provenance-Zwecke ausgewertet werden können.

- Das Ergebnis der Provenance-Anfrage soll unter verschiedenen Randbedingungen berechnet werden können. Die zu entwickelnde Technik soll dabei für das Provenance-Ergebnis die variierenden Randbedingungen
  - Erhalt der Tupelanzahl der Originalrelation im Provenance-Ergebnis
  - Homomorphes Abbild des Provenance-Ergebnisses auf die Originaldatenbank, sowie
  - intensionale Beschreibung der Originaldatenbank als Provenance-Ergebnis

  erfüllen können.
- In verschiedenen Stufen soll dabei sowohl die Mächtigkeit der Sprache für die Auswertungsanfragen gesteigert werden (konjunktive Anfragen, Relationenalgebra, Relationenalgebra mit Aggregatfunktionen, skalaren Funktionen und Gruppierung, schließlich Ergänzung von rekursiven Anfragen) als auch die Mächtigkeit der Evolutionsoperationen. Bei Provenance-Anfragen für semantisch reiche Anfragesprachen wie SQL sind andere Ansätze wie die von Grust et al ([34, 36, 35, 33]) zum Vergleich heranzuziehen.

Nicht Ziel der Promotion sind formale Betrachtungen der grundlegenden Techniken wie CHASE und inverse Schemaabbildungen, um etwa Komplexitätsklassen für die jeweiligen CHASE-Versionen und Sprachklassen für die Auswertungsanfragen zu bestimmen. Bestehende Ergebnisse in diesen Bereichen sollen genutzt werden. Effekte höherer Komplexitätsklassen wie NP-Vollständigkeit (NP-Vollständigkeit) und NP-Schwere (NP-hardness) sollen durch Tests an Beispieldatenbanken ermittelt werden.

9 Zeit- und Arbeitsplan

Die Bearbeitungszeit der oben genannten Zielsetzung ist auf 36 - 48 Monate ausgelegt. Der Beginn ist auf den Januar 2018 datiert. Die ersten 12 Monate sind dabei wie folgt strukturiert:

Monate	Ziel
1 - 9	Entwurf eines Exposés
1 - 6	Literaturrecherche: [21], [13], [23], [1], [19], [27], [11], [26], [28]
1 - 3	Betreuung eines Studentenprojekts zur Analyse bisher existierender CHASE-Tools
4 - 9	Betreuung eines Studentenprojekts zur Entwicklung bzw. Weiterentwicklung eines CHASE-Tools
5	Publikation und Vortrag auf dem GvD-Workshop der GI [7]
7	Publikation, Vortrag und Poster auf der Provenance Week [6]
8	Publikation, Vortrag und Poster auf dem LWDA-Workshop [8]
7 - 12	Erste Ansätze zur Lösung der Problemstellung sowie zur prototypischen Entwicklung
7 - 12	Dissertation: State of the Art (1.Fassung)

Die Monate 13 bis 30 der Dissertation dienen anschließend der Ausarbeitung und weiteren Entwicklung der im ersten Jahr entwickelten Ansätze. Das schriftliche Festhalten der neuen Erkenntnisse bildet schließlich den Schwerpunkt des letzten halben Jahres der geplanten 36 Monate.

Literatur

[1] Foto N. Afrati and Rada Chirkova. *Answering Queries Using Views*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2017.

[2] Alfred V. Aho, Catriel Beeri, and Jeffrey D. Ullman. “The Theory of Joins in Relational Databases”. In: *ACM Trans. Database Syst.* 4.3 (1979), pp. 297–314.

[3] Yael Amsterdamer, Daniel Deutch, and Val Tannen. "Provenance for aggregate queries". In: *PODS*. Ed. by Maurizio Lenzerini and Thomas Schwentick. ACM, 2011, pp. 153–164.

[4] Tanja Auge. *Umsetzung von Provenance-Anfragen in Big-Data-Analytics-Umgebungen*. Masterarbeit, Univeristät Rostock. 2017.

[5] Tanja Auge, Sabrina Brossmann, and Pia Wilsdorf. *Neueste Entwicklungen in der Informatik — ProSA*. Projektarbeit, Universität Rostock, `https://eprints.dbis.informatik.uni-rostock.de/826/`. Sommersemester 2016.

[6] Tanja Auge and Andreas Heuer. "Combining Provenance Management and Schema Evolution". In: *IPAW*. Ed. by Khalid Belhajjame, Ashish Gehani, and Pinar Alper. Vol. 11017. Lecture Notes in Computer Science. Springer, 2018, pp. 222–225.

[7] Tanja Auge and Andreas Heuer. "Inverse im Forschungsdatenmanagement". In: *30th Workshop Grundlagen von Datenbanken* (2018), pp. 108–113.

[8] Tanja Auge and Andreas Heuer. "The Theory behind Minimizing Research Data: Result equivalent CHASE-inverse Mappings". In: *LWDA*. Ed. by Rainer Gemulla et al. Vol. 2191. CEUR Workshop Proceedings. CEUR-WS.org, 2018, pp. 1–12.

[9] Michael Benedikt, Julien Leblay, and Efthymia Tsamoura. "PDQ: Proof-driven Query Answering over Web-based Data". In: *PVLDB* 7.13 (2014), pp. 1553–1556.

[10] Ilvio Bruder et al. "Daten wie Sand am Meer - Datenerhebung, -strukturierung, -management und Data Provenance für die Ostseeforschung". In: *Datenbank-Spektrum* 17.2 (2017), pp. 183–196.

[11] Ilvio Bruder et al. "Konzepte für das Forschungsdatenmanagement an der Universität Rostock(Concepts for the Management of Research Data at the University of Rostock)". In: *LWDA*. Ed. by Michael Leyer. Vol. 1917. CEUR Workshop Proceedings. CEUR-WS.org, 2017, p. 165.

[12] Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. "Why and Where: A Characterization of Data Provenance". In: *ICDT*. Ed. by Jan Van den Bussche and Victor Vianu. Vol. 1973. Lecture Notes in Computer Science. Springer, 2001, pp. 316–330.

[13] James Cheney, Laura Chiticariu, and Wang Chiew Tan. "Provenance in Databases: Why, How, and Where". In: *Foundations and Trends in Databases* 1.4 (2009), pp. 379–474.

[14] Yingwei Cui, Jennifer Widom, and Janet L. Wiener. "Tracing the lineage of view data in a warehousing environment". In: *ACM Trans. Database Syst.* 25.2 (2000), pp. 179–227.

[15] Richard Dabels. *Automatische Generierung von Domänengeneralisierungshierarchien*. Bachelorarbeit, Univeristät Rostock. 2017.

[16] Ronald Fagin. "Inverting schema mappings". In: *ACM Trans. Database Syst.* 32.4 (2007), p. 25.

[17] Ronald Fagin et al. "Data exchange: semantics and query answering". In: *Theor. Comput. Sci.* 336.1 (2005), pp. 89–124.

[18] Ronald Fagin et al. "Quasi-inverses of schema mappings". In: *ACM Trans. Database Syst.* 33.2 (2008), 11:1–11:52.

[19] Ronald Fagin et al. "Schema Mapping Evolution Through Composition and Inversion". In: *Schema Matching and Mapping*. Ed. by Zohra Bellahsene, Angela Bonifati, and Erhard Rahm. Data-Centric Systems and Applications. Springer, 2011, pp. 191–222.

[20] Floris Geerts et al. "That's All Folks! LLUNATIC Goes Open Source". In: *PVLDB* 7.13 (2014), pp. 1565–1568.

[21] Sergio Greco, Cristian Molinaro, and Francesca Spezzano. *Incomplete Data and Data Dependencies in Relational Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2012.

[22] Todd J. Green, Gregory Karvounarakis, and Val Tannen. "Provenance semirings". In: *PODS*. Ed. by Leonid Libkin. ACM, 2007, pp. 31–40.

[23] Todd J. Green and Val Tannen. "The Semiring Framework for Database Provenance". In: *PODS*. Ed. by Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts. ACM, 2017, pp. 93–99.

[24] Hannes Grunert and Andreas Heuer. "Datenschutz im PArADISE". In: *Datenbank-Spektrum* 16.2 (2016), pp. 107–117.

[25] Alon Y. Halevy. "Answering queries using views: A survey". In: *VLDB J.* 10.4 (2001), pp. 270–294.

[26] Andreas Heuer. "METIS in PArADISE Provenance Management bei der Auswertung von Sensordatenmengen für die Entwicklung von Assistenzsystemen". In: *BTW*. Ed. by Norbert Ritter et al. Vol. 242. LNI. GI, 2015, pp. 131–136.

[27] Sven Köhler, Bertram Ludäscher, and Daniel Zinn. "First-Order Provenance Games". In: *In Search of Elegance in the Theory and Practice of Computation - Essays Dedicated to Peter Buneman*. Ed. by Val Tannen et al. Vol. 8000. Lecture Notes in Computer Science. Springer, 2013, pp. 382–399.

[28] Seokki Lee et al. "A SQL-Middleware Unifying Why and Why-Not Provenance for First-Order Queries". In: *IEEE*. IEEE Computer Society, 2017, pp. 485–496.

[29] David Maier. *The Theory of Relational Databases*. Computer Science Press, 1983.

[30] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. "Testing Implications of Data Dependencies". In: *ACM Trans. Database Syst.* 4.4 (1979), pp. 455–469.

[31] Dennis Marten and Andreas Heuer. "A framework for self-managing database support and parallel computing for assistive systems". In: *PETRA*. Ed. by Fillia Makedon. ACM, 2015, 25:1–25:4.

[32] Dennis Marten and Andreas Heuer. "Machine Learning on Large Databases: Transforming Hidden Markov Models to SQL Statements". In: *OJDB* 4.1 (2017), pp. 22–42.

[33] Tobias Müller. "Where- und Why-Provenance für syntaktisch reiches SQL durch Kombination von Programmanalysetechniken". In: *GvD*. Vol. 1366. CEUR Workshop Proceedings. CEUR-WS.org, 2015, pp. 84–89.

[34] Tobias Müller, Benjamin Dietrich, and Torsten Grust. "You Say 'What', I Hear 'Where' and 'Why'? (Mis-)Interpreting SQL to Derive Fine-Grained Provenance". In: *PVLDB* 11.11 (2018), pp. 1536–1549.

[35] Tobias Müller and Torsten Grust. "Provenance for SQL through Abstract Interpretation: Value-less, but Worthwhile". In: *PVLDB* 8.12 (2015), pp. 1872–1875.

[36] Daniel O'Grady, Tobias Müller, and Torsten Grust. "How 'How' Explains What 'What' Computes - How-Provenance for SQL and Query Compilers". In: *TaPP*. USENIX Association, 2018.

[37] Fabian Renn and Frank Röger. *KSWS — ProSA*. Projekt, Univeristät Rostock. Wintersemester 2017/2018.

[38] Stuart J. Russell and Peter Norvig. *Künstliche Intelligenz - ein moderner Ansatz (2. Aufl.)* Pearson Studium, 2004.

[39] Jan Svacina. *Intensional Answers for Provenance Queries in Big Data Analytics*. Bachelor thesis, Universität Rostock. 2016.

A Überblick über verschiedene CHASE-Vairanten

	$\star$	$\bigcirc$	Result	Goal	Tool
0.	dependency	database schema	database schema with integrity condition	optimized database design	
I.	dependency	query	query	semantic optimization	PDQ[1]
II.	view	query	query using views	AQuV	ProvCB[2]
II'.	operator	query	query using operator	AQuO	
III.	s-t tgds, egds	source database	target database	data exchange, data integration	Llunatic[3], ChaseFUN[4]
IV.	tgds, egds	DB	modified database	cleaning	Llunatic, ChaseFUN
V.	tgds, egds	incomplete database	query result	save answers	
VI.	s-t tgds, egds, tgds	database	query result	invertible query evaluation	

[1] `http://www.cs.ox.ac.uk/projects/pdq/home.html`
[2] `https://github.com/IoanaIleana/ProvCB`
[3] `http://db.unibas.it/projects/llunatic/`
[4] `https://openproceedings.org/2017/conf/edbt/paper-396.pdf`

Tabelle 6 Überblick über einige CHASE-Varianten

B Grundoperationen und ihre zughörigen CHASE-Inversen

Operation $\mathcal{M}$	Without Provenance		With Provenance	
	Inverse type	Inverse $\mathcal{M}^{-1}$	Inverse type	Inverse $\mathcal{M}^{-1}$
$r(\mathcal{R})$	$=$	identity	$=$	identity
$\beta_{A_j \leftarrow A_i}(r(\mathcal{R}))$	$=$	renaming $\beta_{A_i \leftarrow A_j}$	$=$	renaming $\beta_{A_i \leftarrow A_j}$
$\pi_{A_i}(r(\mathcal{R}))$	$\preceq_{tp}$	null value extension in A_i	$\preceq_{tp}$	null value extension in A_i
	$\preceq$	null value extension in A_i	$\preceq_{tp}$	null value extension in A_i + reconstruction of lost tuples
$r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$	$=$	projection on attributes from $\mathcal{R}_i$	$=$	projection on attributes from $\mathcal{R}_i$
	$\preceq$	projection on attributes from $\mathcal{R}_i$	$\preceq$	projection on attributes from $\mathcal{R}_i$
$\sigma_{A_i \theta c}(r(\mathcal{R}))$	$\preceq$	identity	$\preceq$	identity
$\sigma_{A_i \theta A_j}(r(\mathcal{R}))$	$\preceq$	identity	$\preceq$	identity
with $\theta \in \{<, \leq, =, \geq, >\}$				
$\sigma_{A_i \neq c}(r(\mathcal{R}))$	xxx	xxx	xxx	xxx
$\sigma_{A_i \neq A_j}(r(\mathcal{R}))$	xxx	xxx	xxx	xxx
$r_1(\mathcal{R}_1) \cup r_2(\mathcal{R}_2)$	$\leftrightarrow$	identity	$=$	selection
$r_1(\mathcal{R}_1) \cap r_2(\mathcal{R}_2)$	$\preceq$	identity	$\preceq$	identity
$r_1(\mathcal{R}_1) - r_2(\mathcal{R}_2)$	xxx	xxx	xxx	xxx
$\mathbf{MAX}_{A_i}(r(\mathcal{R}))$ / $\mathbf{MIN}(r(\mathcal{R}))$	$\preceq$	identity	$\preceq$	null value extension in A_j, $i \neq j$
$\mathbf{COUNT}_{A_i}(r(\mathcal{R}))$	$\preceq_{tp}$	null tuple generation	$\preceq_{tp}$	null tuple generation
$\mathbf{SUM}_{A_i}(r(\mathcal{R}))$	$\leftrightarrow$	identity	$=$	reconstruction of lost attribute values in A_i
				+ null value extension in A_j, $i \neq j$
$\mathbf{AVG}_{A_i}(r(\mathcal{R}))$	$\leftrightarrow$	identity	$=$	reconstruction of lost attribute values in A_i
				+ null value extension in A_j, $i \neq j$
$\gamma_{G_i;F_j(A_j)}(r(\mathcal{R}))$	$\preceq$	identity	$\preceq$	identity
	$\preceq_{tp}$	null tuple generation	$\preceq_{tp}$	null value extension in A_i + reconstruction of lost tuples
	$\leftrightarrow$	identity	$=$	reconstruction of lost tuples and attribute values in A_i
$r(\mathcal{R})\ \theta\ \alpha$ with $\theta \in \{+, -, \cdot, /\}$	$=$	$- \ / \ + \ / \ : \ / \ \cdot$	$=$	$- \ / \ + \ / \ : \ / \ \cdot$
(one variable scalar functions)				

Tabelle 7 Basic operations and their exact ($=$), tp-relaxed ($\preceq_{tp}$), relaxed ($\preceq$) or result equivalent ($\leftrightarrow$) CHASE-inverse with restriction to the attribute A_i for **COUNT**, **SUM** and **AVG**