

Verteidigung der Bachelorthesis

zum Thema:

Erarbeitung von Empfehlungen zur Optimierung der
Performance bei der Programmierung von
Datenbankanwendungen mittels .NET Framework und
objektrelationalem Mapping

Berufsakademie Sachsen
Staatliche Studienakademie Leipzig

Autor: Stefan Lerm
Matrikelnummer: 5000552
Seminargruppe: CS14-1
Erstgutachter: Dipl.-Ing. Dipl.-Wirtsch.-Ing. Gregor Wilhelmi
Zweitgutachter: Prof. Dr.-Ing. Christian Heller

18.09.2017

Gliederung

1. Motivation und Zielstellung
2. Vorgehensweise
3. Ergebnisse

1. Motivation und Zielstellung



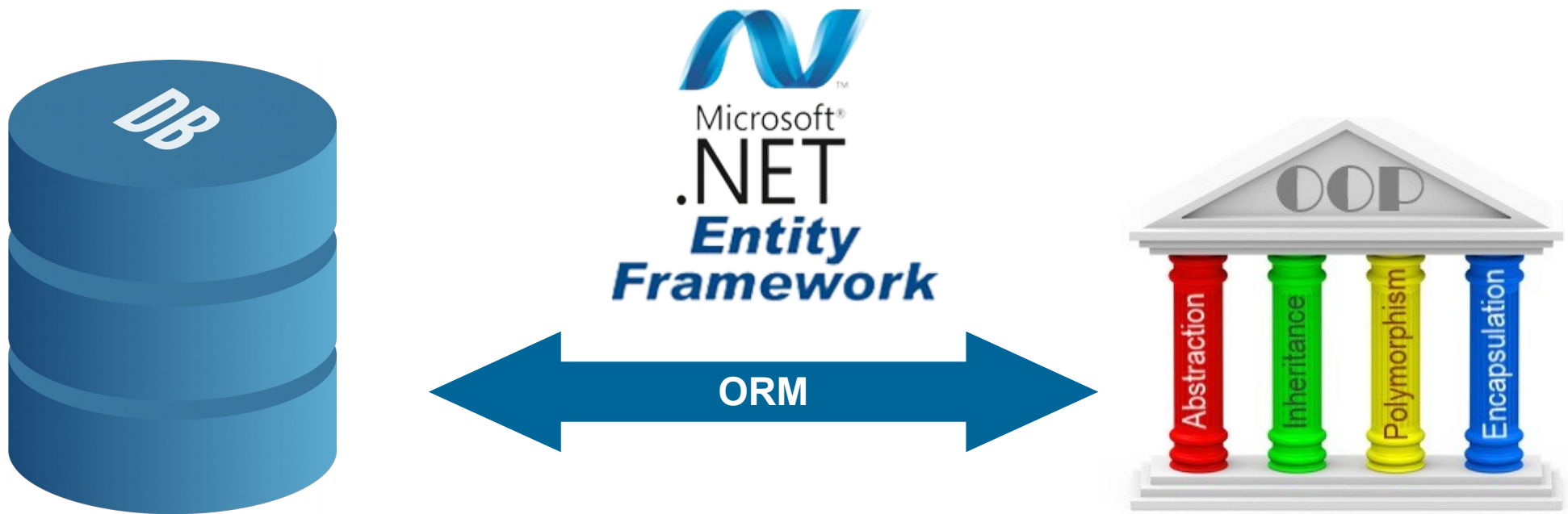
1.1 Motivation



1.1 Motivation



1.1 Motivation



1.2 Zielstellung



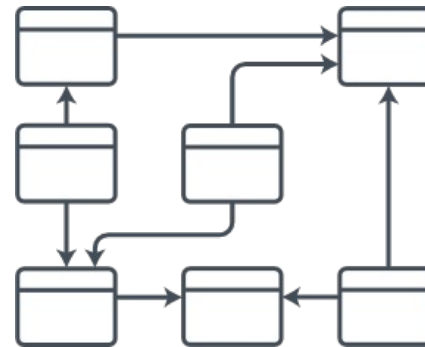
Bildquellen: <http://www.trading-portal.net/die-entscheidung-steht-aus/>

1.2 Zielstellung



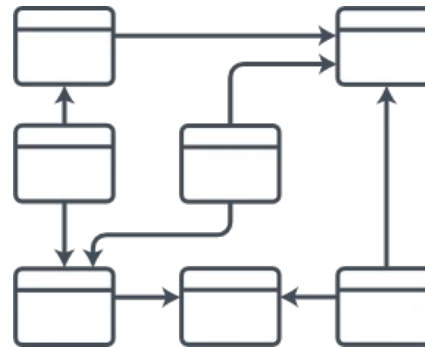
Bildquellen: <http://www.trading-portal.net/die-entscheidung-steht-aus/>
<https://www.lucidchart.com/pages/database-diagram/database-models>

1.2 Zielstellung



Bildquellen: <http://www.trading-portal.net/die-entscheidung-steht-aus/>
<https://www.lucidchart.com/pages/database-diagram/database-models>
<http://hostforlife.eu/European-Entity-Framework-7-Hosting>
<http://www.sedlak-partner.de/wp-content/uploads/2014/01/icon-zahnrad.png>

1.2 Zielstellung



Natives SQL

Stored Procedures

Bildquellen: <http://www.trading-portal.net/die-entscheidung-steht-aus/>
<https://www.lucidchart.com/pages/database-diagram/database-models>
<http://hostforlife.eu/European-Entity-Framework-7-Hosting>
<http://www.sedlak-partner.de/wp-content/uploads/2014/01/icon-zahnrad.png>

1.2 Zielstellung



Natives SQL

Stored Procedures



Bildquellen: <http://www.examedia.com.my/database/>
<http://www.trading-portal.net/die-entscheidung-steht-aus/>
<https://www.lucidchart.com/pages/database-diagram/database-models>
<http://hostforlife.eu/European-Entity-Framework-7-Hosting>
<http://www.sedlak-partner.de/wp-content/uploads/2014/01/icon-zahnrad.png>

2. Vorgehensweise

A - Theoretisch

Moderne Datenzugriffslösungen mit

Entity Framework 6

Manfred Steyer, Holger Schwichtenberg



Bildquellen: <http://www.sismarketresearch.com/>
<https://www.iasplus.com/en/images/responsive/photos/landing/standards>
<https://www.amazon.de/Moderne-Datenzugriffsl%C3%B6sungen-Entity-Framework-Datenbankprogrammierung/dp/3934279139>

2. Vorgehensweise

A - Theoretisch

Moderne Datenzugriffslösungen mit
Entity Framework 6

Manfred Steyer, Holger Schwichtenberg

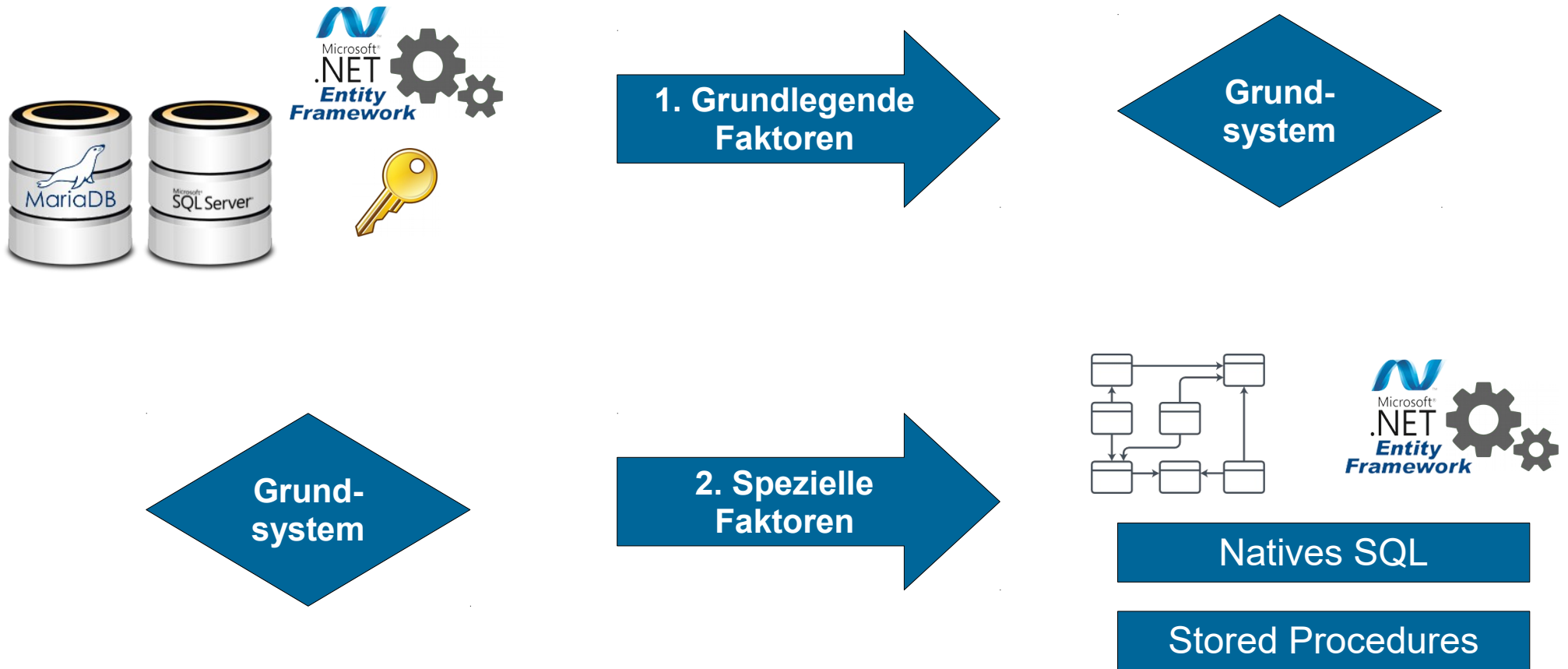


B - Experimentell



Bildquellen: <http://www.sismarketresearch.com/>
<https://www.iasplus.com/en/images/responsive/photos/landing/standards>
<https://www.amazon.de/Moderne-Datenzugriffsl%C3%B6sungen-Entity-Framework-Datenbankprogrammierung/dp/3934279139>

2.1 Konzept für die experimentelle Ermittlung von Optimierungsansätzen



2.1 Grundlegende Faktoren

DBMS



2.1 Grundlegende Faktoren

DBMS



Schlüsseltyp



Identity



E40C1833-2EC8-4D5E-
BF35-00027A1904E4

GUID

2.1 Grundlegende Faktoren

DBMS



Schlüsseltyp



Identity



E40C1833-2EC8-4D5E-BF35-00027A1904E4

GUID

Änderungs-
verfolgung

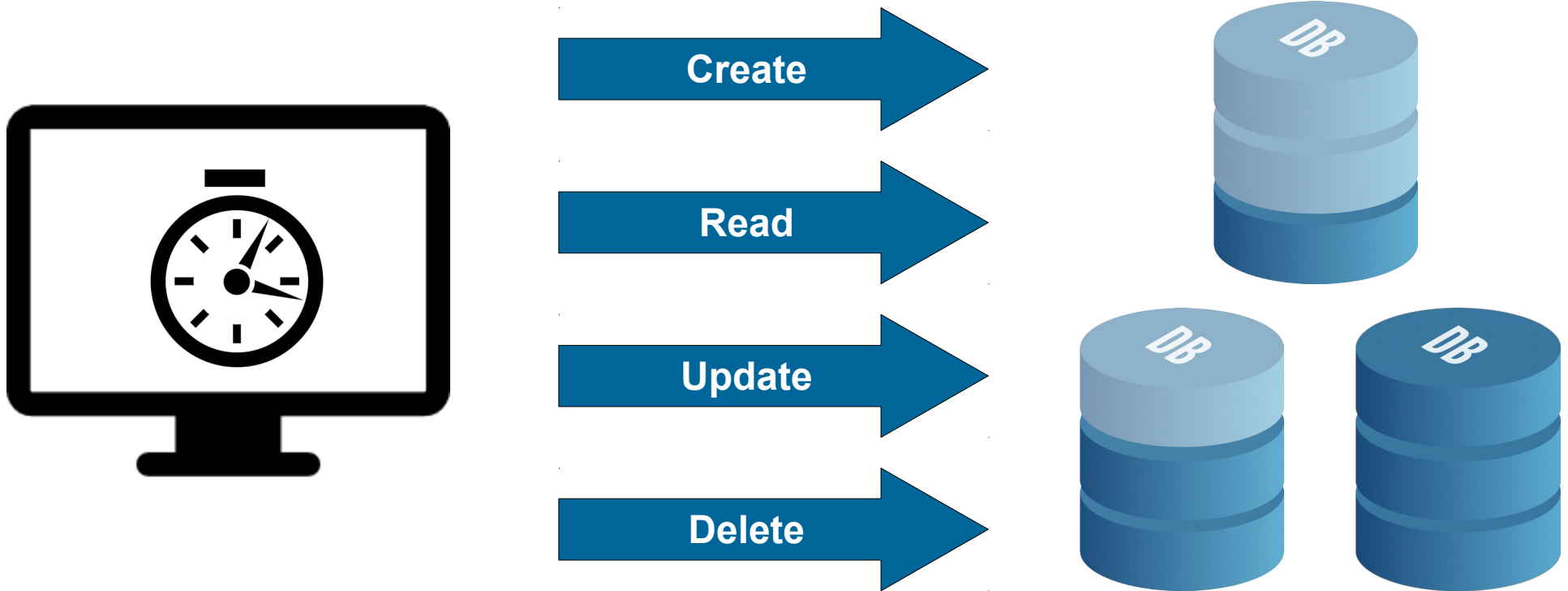


Snapshot Change
Tracking



Dynamic Change
Tracking

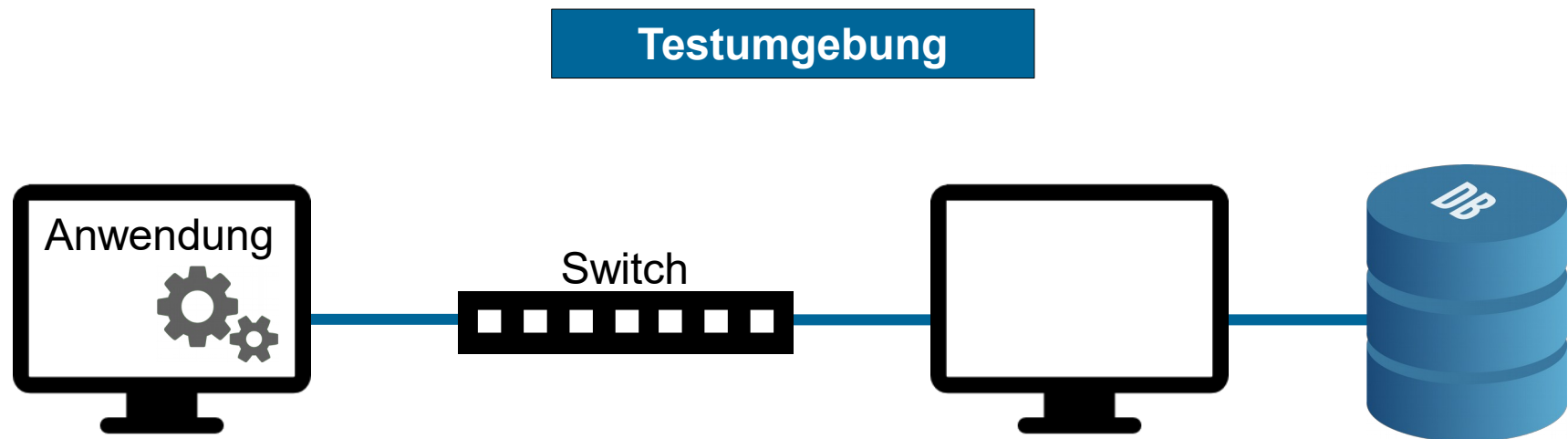
2.2 Feststellung der Performance



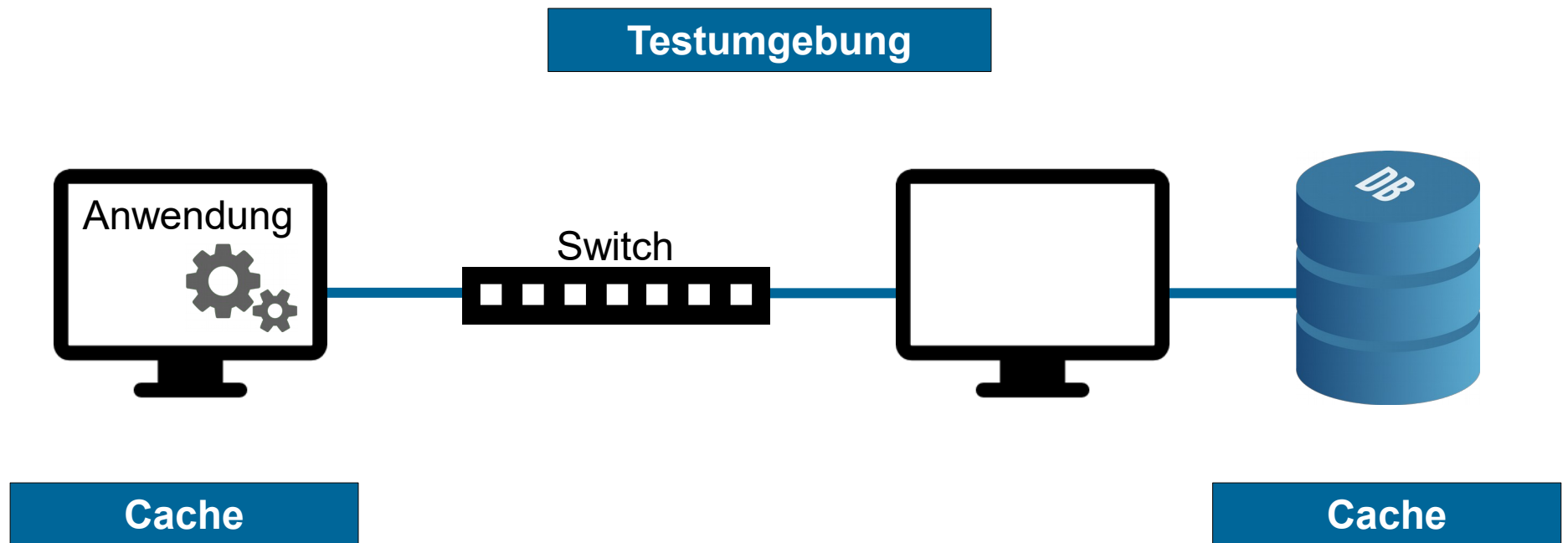
2.3 Experimente für die grundlegenden Datenbankoperationen

	Dynamic Change Tracking				Snapshot Change Tracking			
	Microsoft SQL Server		Maria DB		Microsoft SQL Server		Maria DB	
	Identity	GUID	Identity	GUID	Identity	GUID	Identity	GUID
Create								
Read								
Update								
Delete								

2.5 Fehlerquellen und Grenzen



2.5 Fehlerquellen und Grenzen



3. Ergebnisse



3.1 Empfehlungen für grundlegende Faktoren

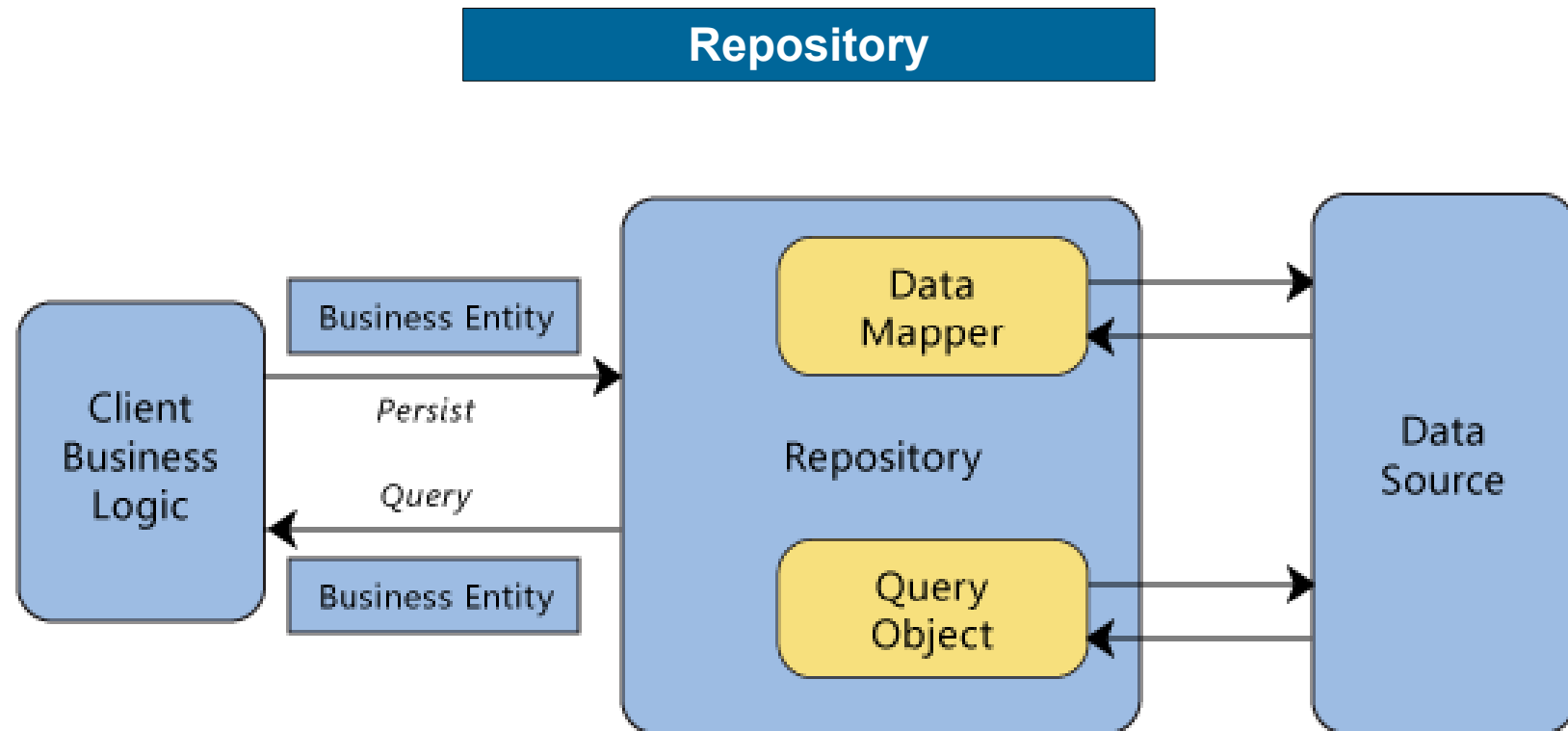
	Dynamic Change Tracking				Snapshot Change Tracking			
	Microsoft SQL Server		Maria DB		Microsoft SQL Server		Maria DB	
	Identity	GUID	Identity	GUID	Identity	GUID	Identity	GUID
	Identity	GUID	Identity	GUID	Identity	GUID	Identity	GUID
Create	●	✗	✗	✗	✓	✗	✗	✗
Read	●	✓	✗	✗	●	✓	✗	✗
Update	✓	✓	●	●	●	●	●	●
Delete	✓	✗	●	●	●	✗	●	●



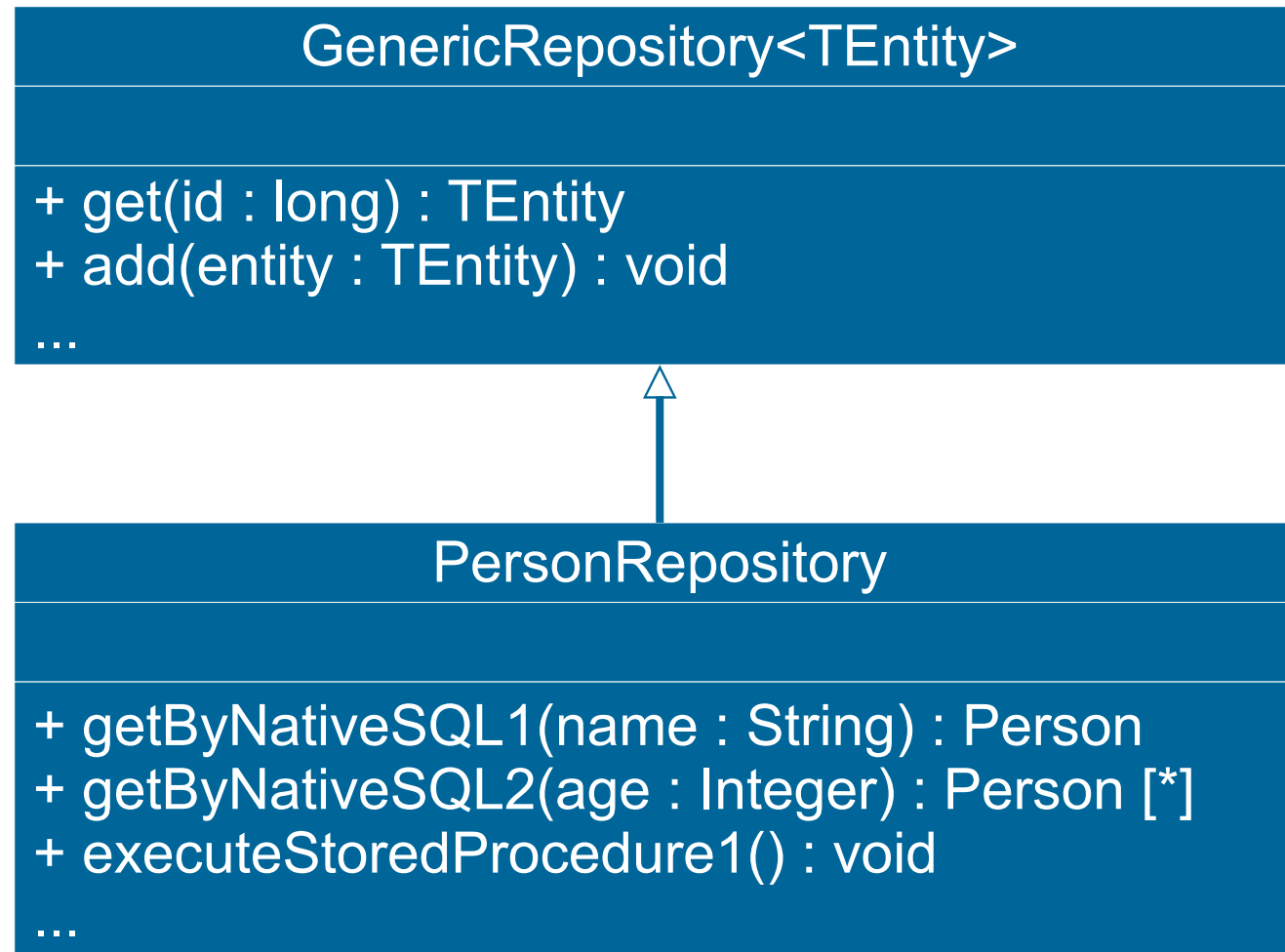
3.2 Empfehlungen für spezielle Faktoren

Kriterium	Empfehlung
Vererbungsstrategie	Table per concrete Type
Denormalisierung: Wiederholungsgruppe	wenn Anzahl der wiederholten Attribute begrenzt
Denormalisierung: redundante Datenspeicherung	Nur wenn die eingebundenen Tabellen mehr Daten aufweisen, als die Tabelle deren Daten abgefragt werden
Abfragen	<ul style="list-style-type: none"> - einfache Abfragen: LINQ to Entities - Anzahl Joins > 0: natives SQL
Massenvorgänge (Ändern, Einfügen, Löschen)	Natives SQL
Komplexe Vorgänge	<ul style="list-style-type: none"> - gegebenenfalls gespeicherte Prozeduren - Vergleich zum Vorgehen mit EF zwingend erforderlich

3.3 Mustervorschlag: Repository Wrapper



3.3 Mustervorschlag: Repository Wrapper



3.4 Fazit und Ausblick

- Hauptziel:
 - Empfehlungen für zukünftige Projekte ✓
- Nebenziel:
 - Optimierungsansätze für bestehende Anwendungen ✓
- Ausblick:
 - Experimente mit leerem Cache
 - weitere DBMS



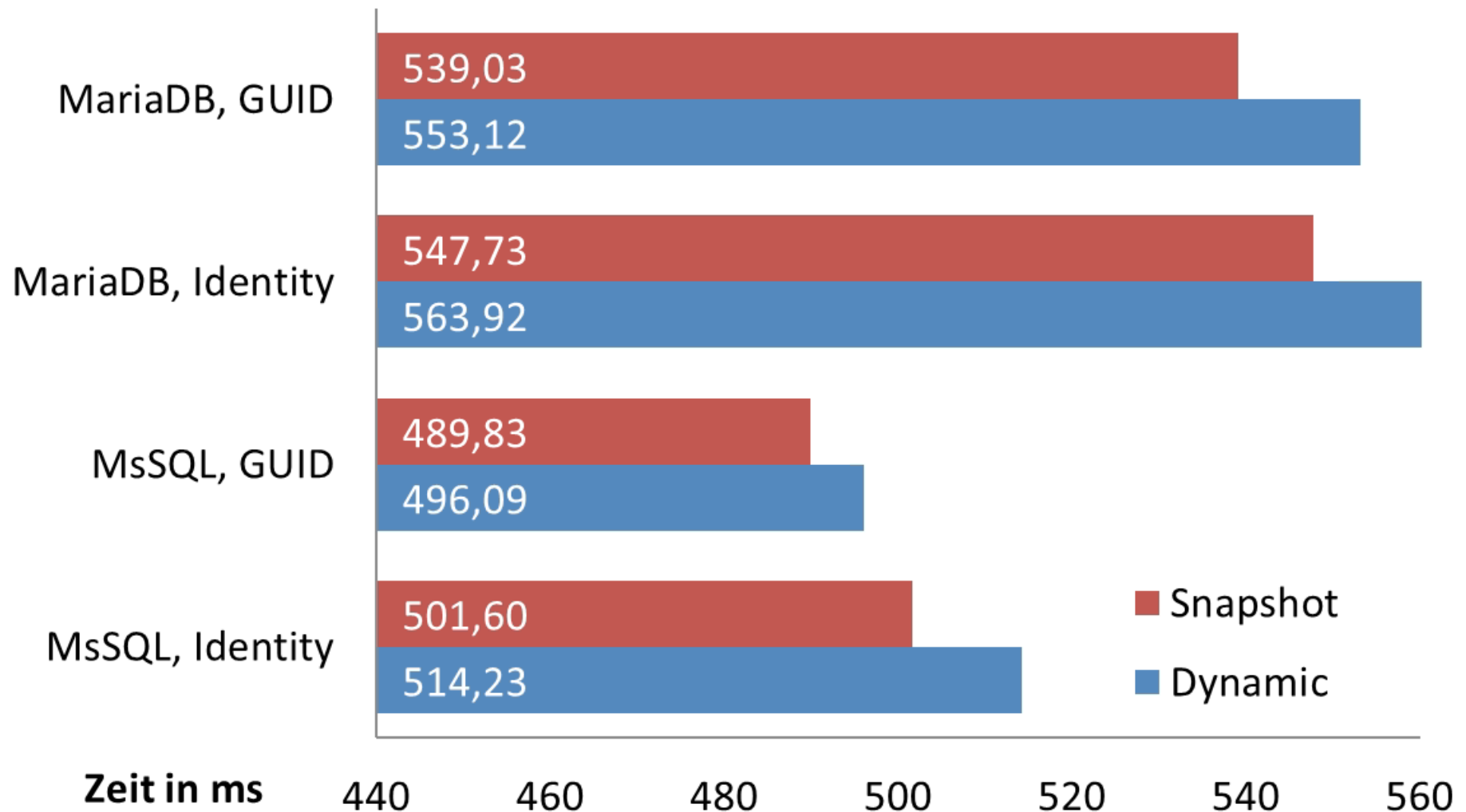
4. Reserve

- Empfehlungen auf Basis der theoretischen Betrachtung
- Tests der CRUD-Operationen
- Tests der Denormalisierung
 - Wiederholungsgruppe
 - Redundanz in einer Beziehungskette
- Tests der Vererbungsstrategien
- Tests mit nativem SQL
 - Massenvorgänge
 - Abfragen
- Test eines komplexen Vorgangs (Stored Procedures)
- Im EF verwendete Entwurfsmuster + ausführlicher Repository Wrapper

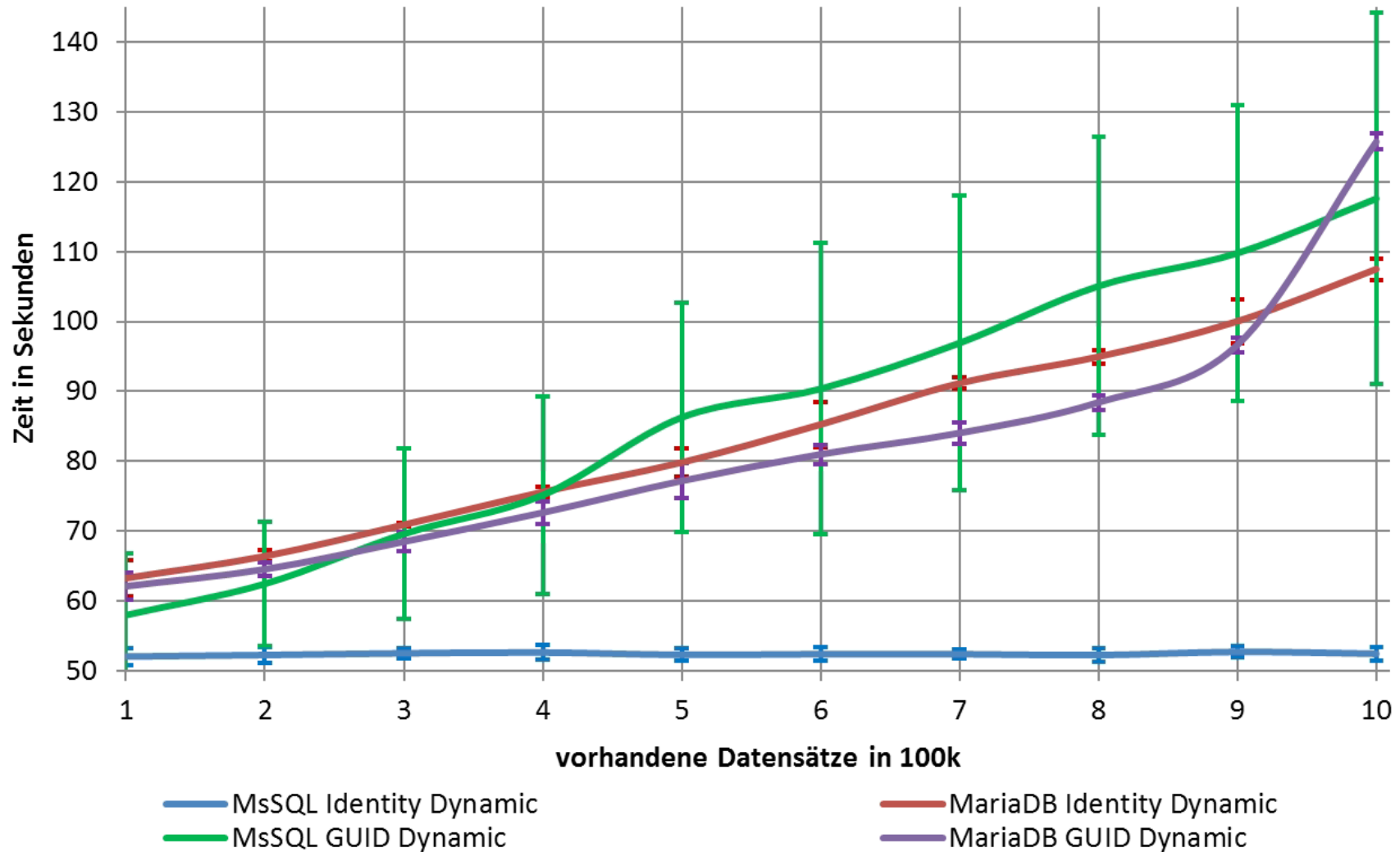
Empfehlungen auf Basis der theoretischen Betrachtung









Kriterium	Empfehlung
Ladestrategie	<ul style="list-style-type: none"> - mit Änderungsverfolgung <ul style="list-style-type: none"> - "Includes" < 3 : "Eager Loading" - "Includes" > 2 : "Pre-Loading" - ohne Änderungsverfolgung <ul style="list-style-type: none"> - "Includes" < 3 : "Eager Loading" - "Includes" > 2 : "Lazy Loading"
Query Plan Caching	Vermeidung unterschiedlicher Parameter für <i>take</i> und <i>skip</i>
Verwendung von Datenbankkontexten	Kurzlebig innerhalb eines "Using-Blocks"
View Generation	<ul style="list-style-type: none"> - zur Laufzeit beschleunigen durch zusätzliche Fremdschlüsseleigenschaften - "Pre-Generated Views" durch <i>EF Power Tools</i> einsetzen
Abfragen	- wenn Möglich Abfragen ohne Änderungsverfolgung ausführen (AsNoTracking)

Einfügen von 1.000 Datensätzen

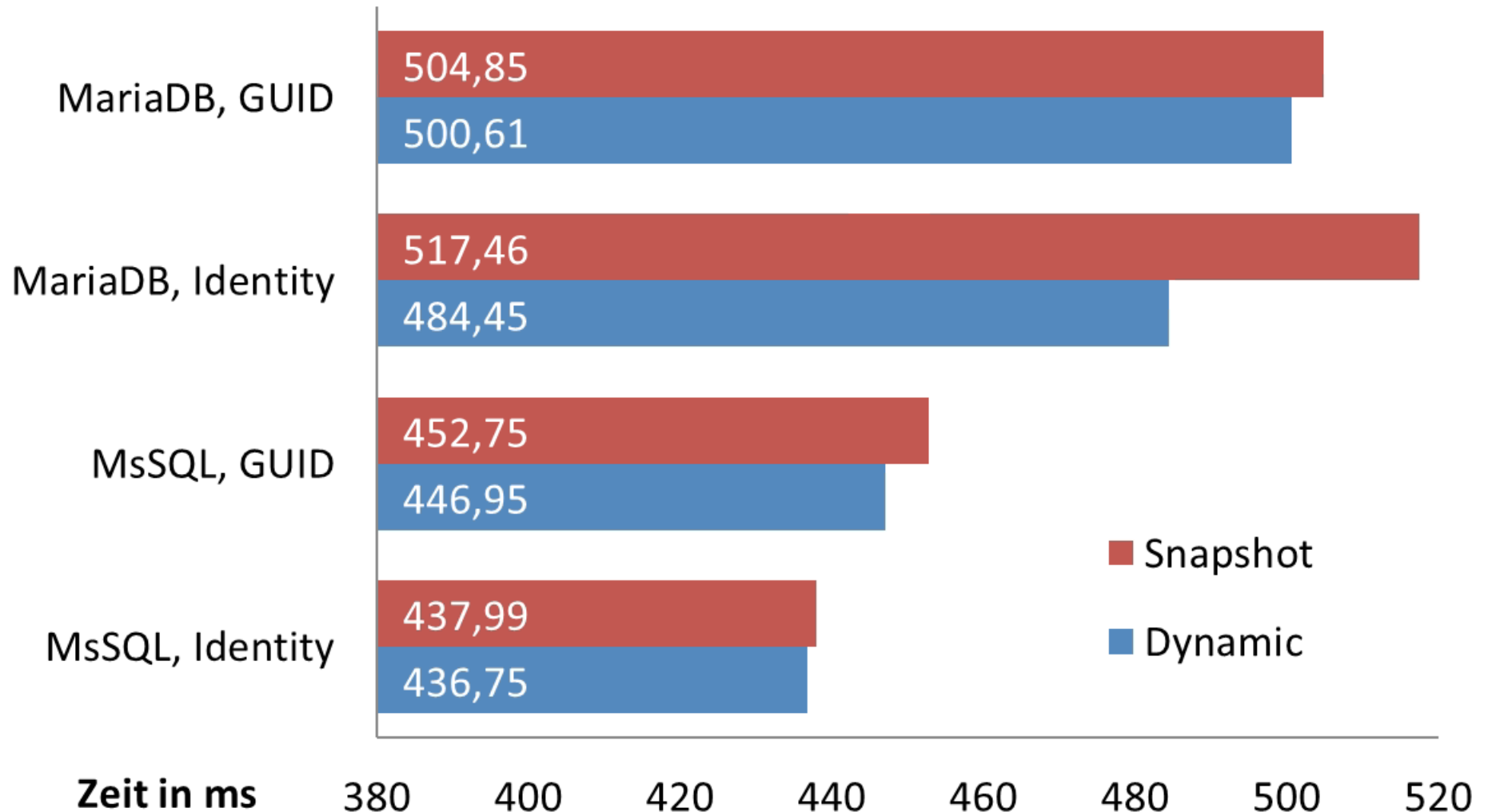


Einfügen von 100.000 Datensätzen bei variablem Datenbestand

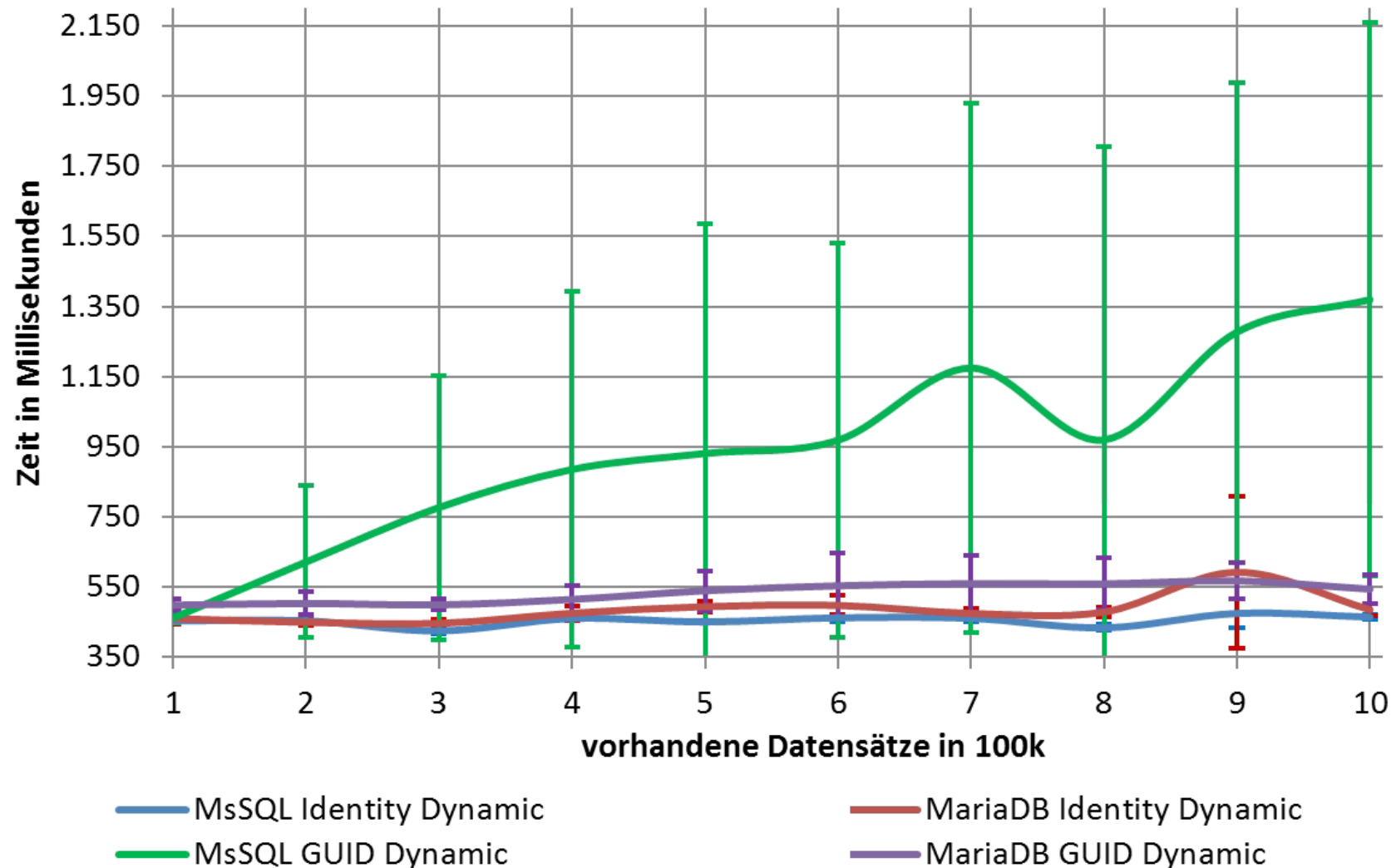


















	Dynamic Change Tracking				Snapshot Change Tracking			
	Microsoft SQL Server		Maria DB		Microsoft SQL Server		Maria DB	
	Identity	GUID	Identity	GUID	Identity	GUID	Identity	GUID
Create								
Delete								
Update								
Read								

Löschen von 1.000 Datensätzen

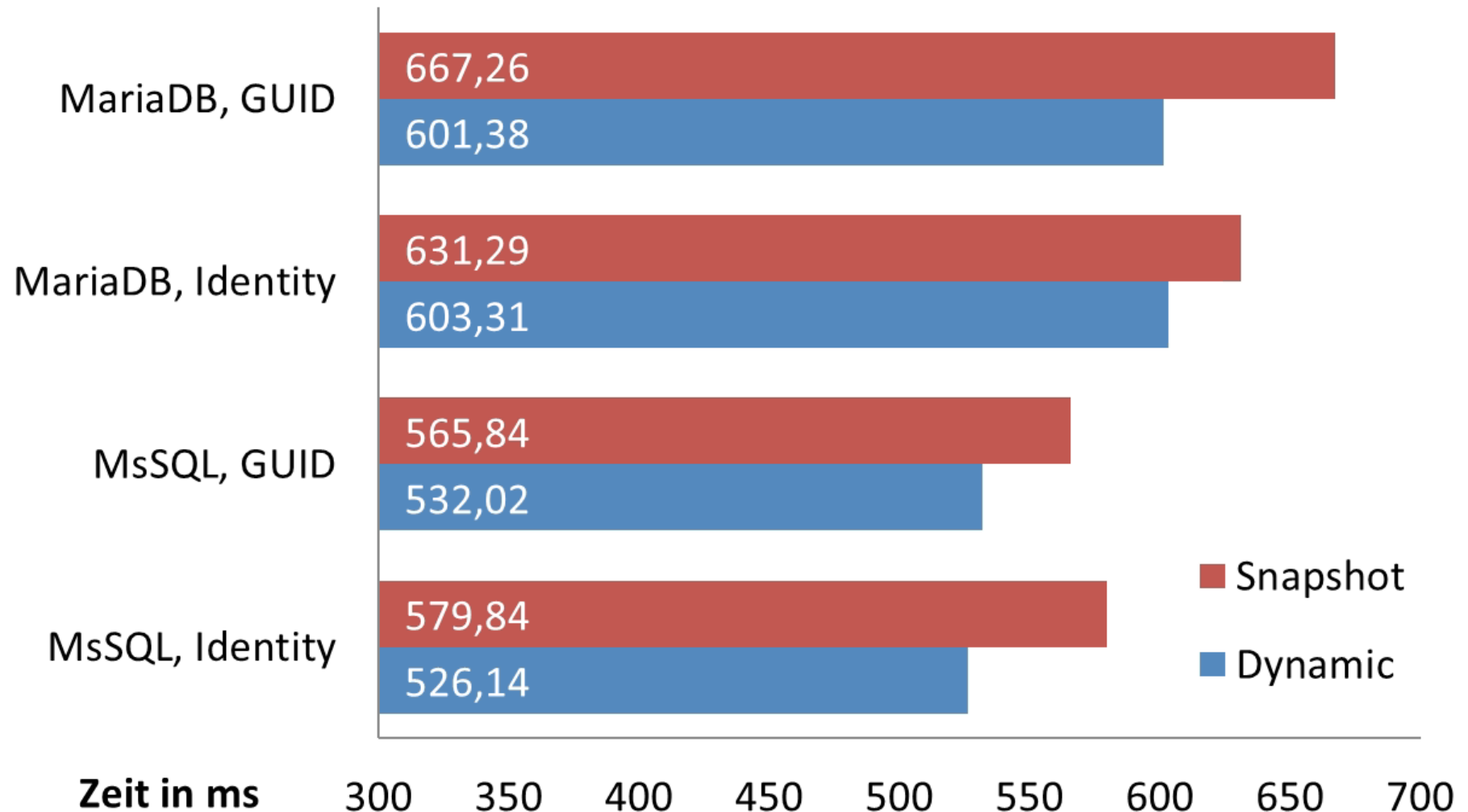


Löschen von 1.000 Datensätzen bei variablem Datenbestand

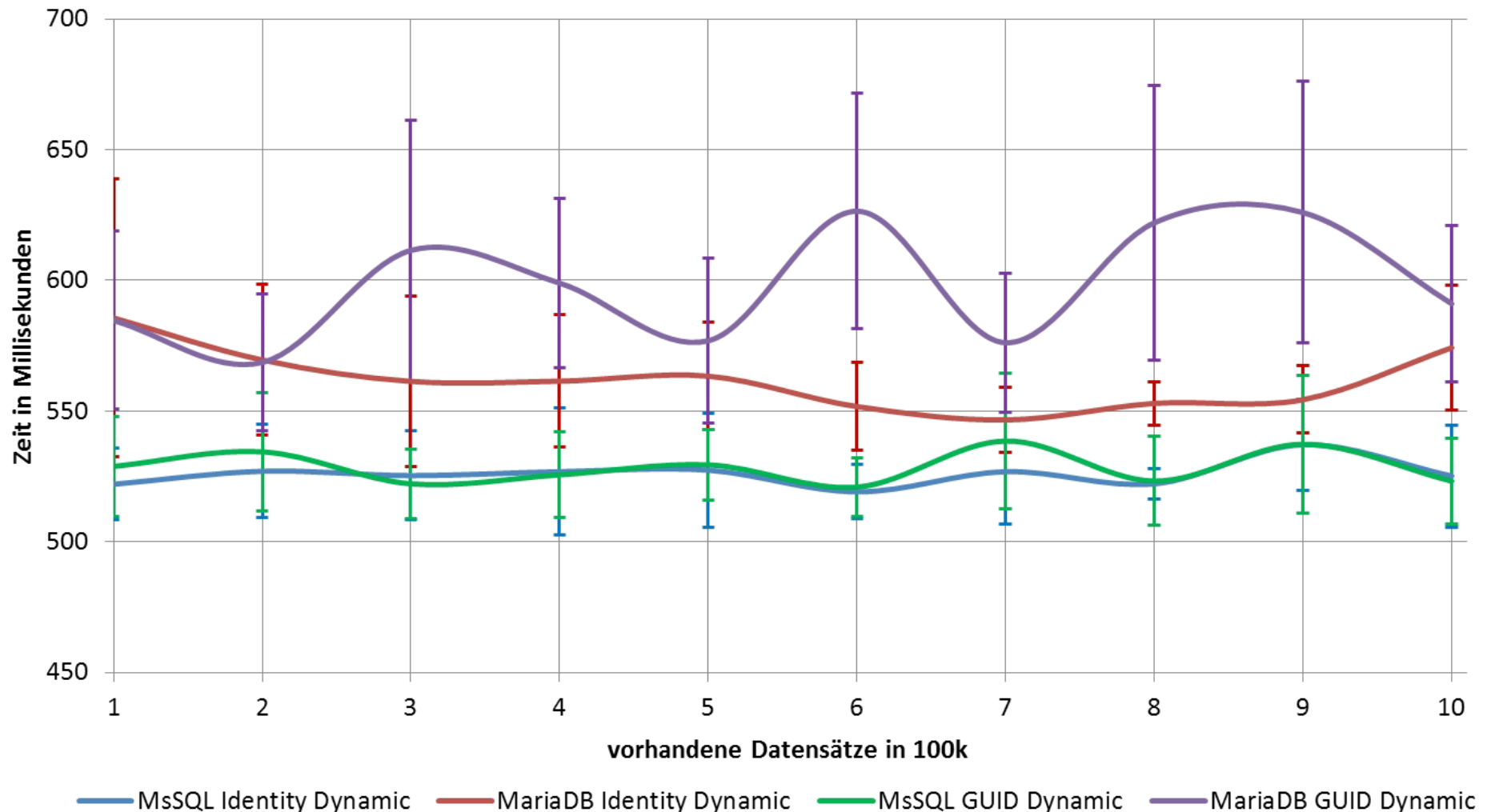


























	Dynamic Change Tracking				Snapshot Change Tracking			
	Microsoft SQL Server		Maria DB		Microsoft SQL Server		Maria DB	
	Identity	GUID	Identity	GUID	Identity	GUID	Identity	GUID
Create								
Delete								
Update								
Read								

Ändern von 1.000 Datensätzen

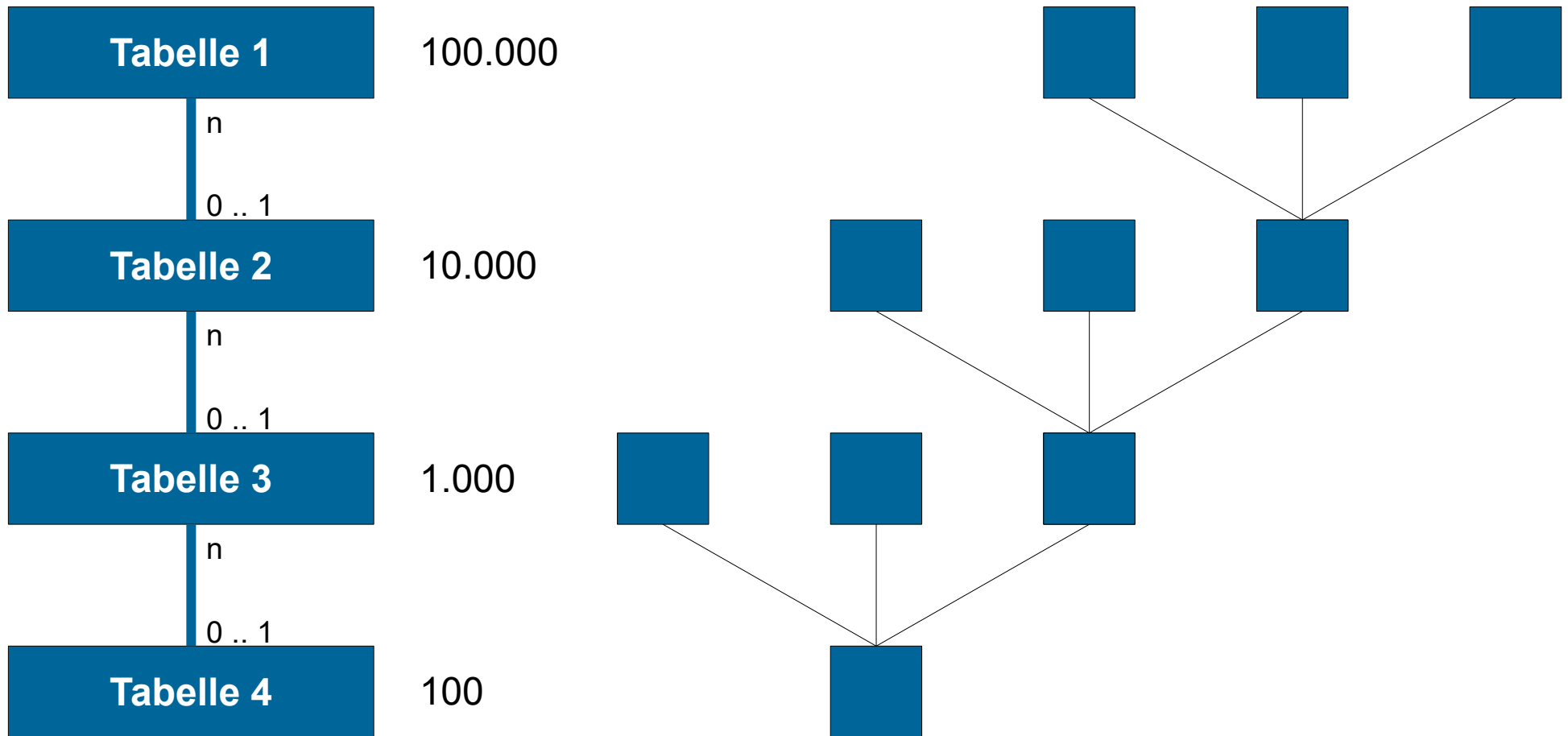


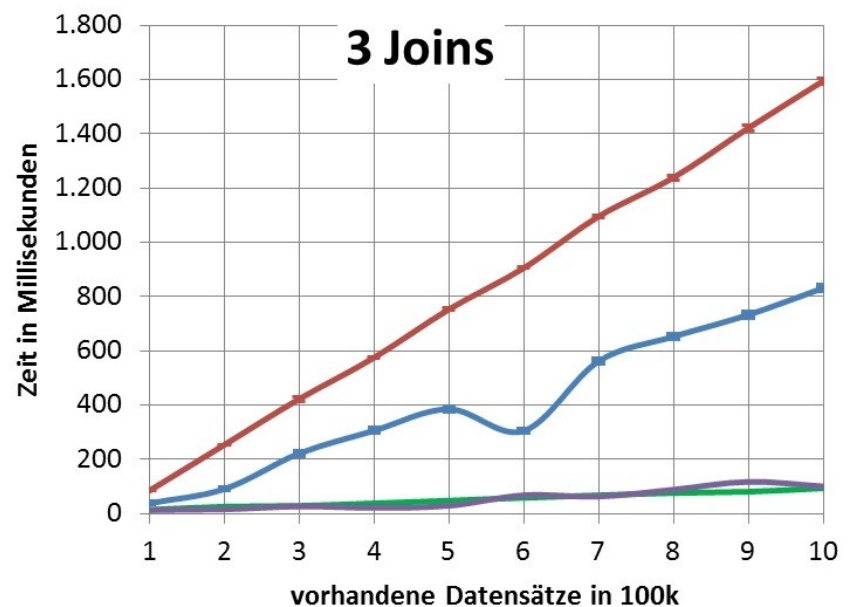
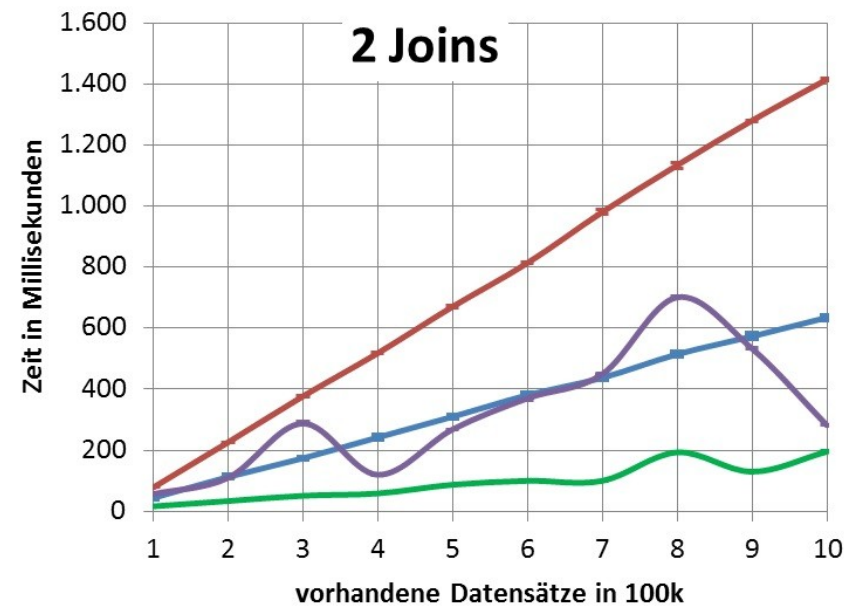
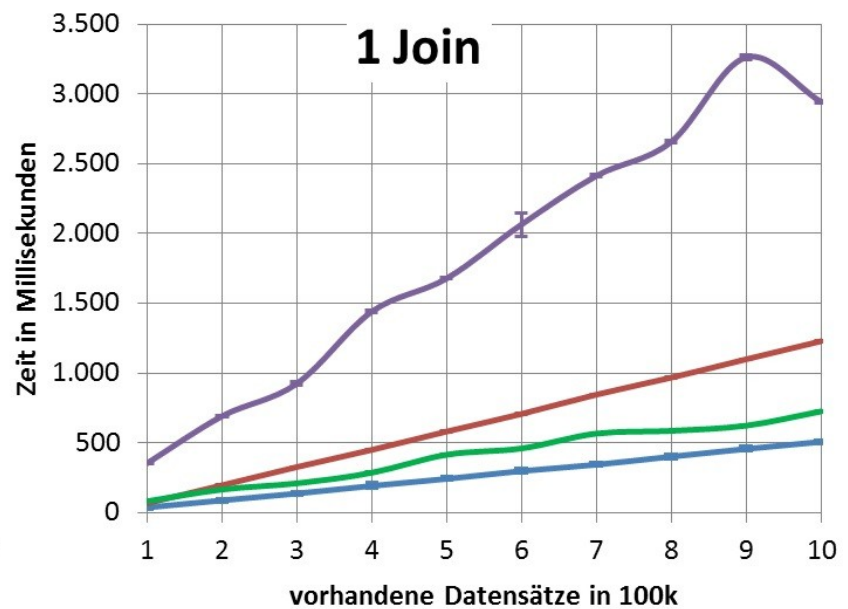
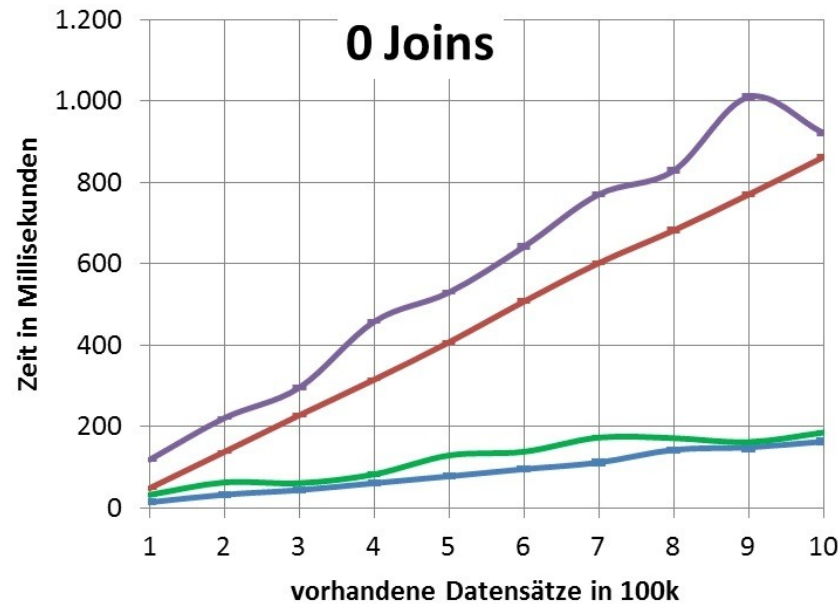
Ändern von 1.000 Datensätzen bei variablem Datenbestand



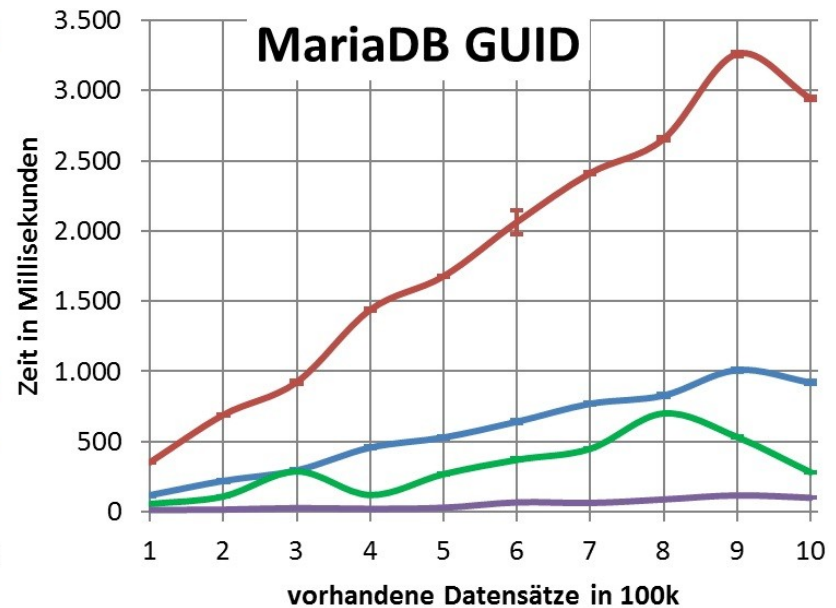
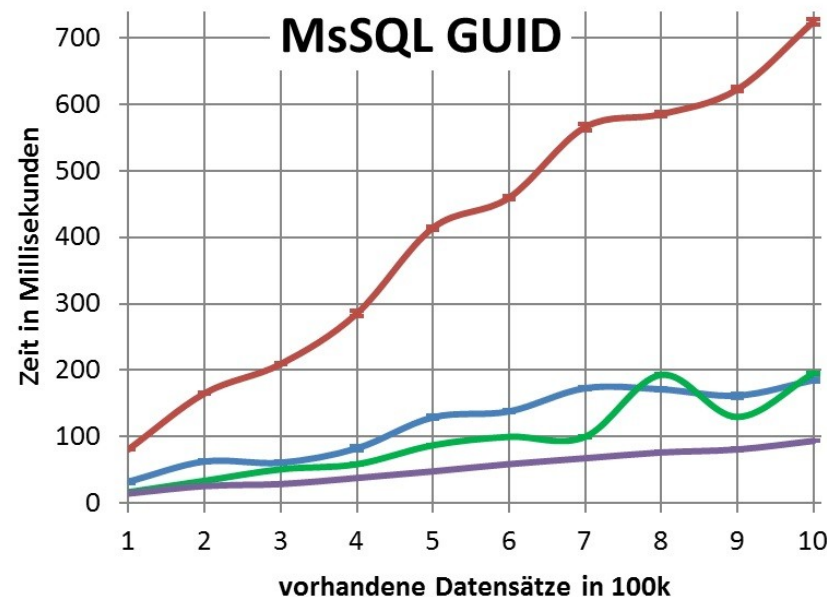
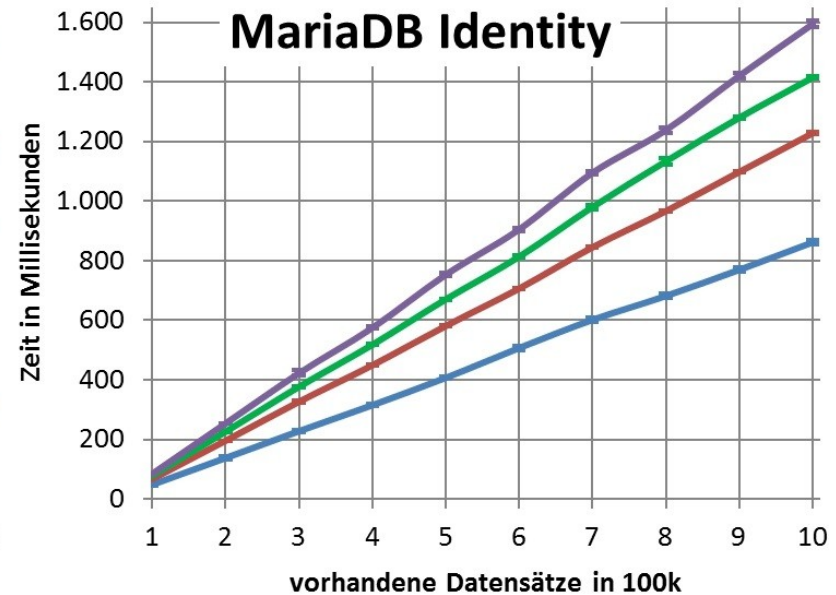
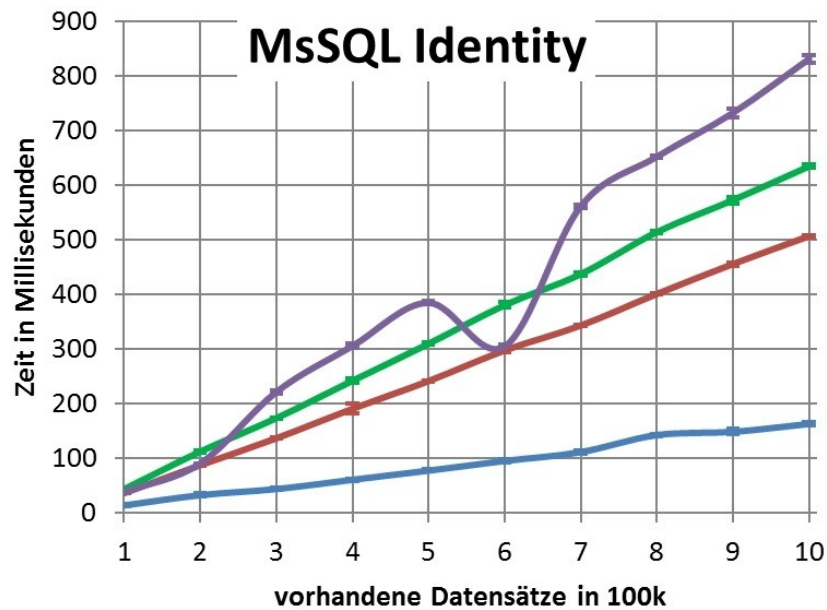
	Dynamic Change Tracking				Snapshot Change Tracking			
	Microsoft SQL Server		Maria DB		Microsoft SQL Server		Maria DB	
	Identity	GUID	Identity	GUID	Identity	GUID	Identity	GUID
Create								
Delete								
Update								
Read								

Abfragen von Datensätzen in einer Beziehungskette





— MySQL Identity Dynamic — MariaDB Identity Dynamic — MySQL GUID Dynamic — MariaDB GUID Dynamic



— 0 Joins — 1 Join — 2 Joins — 3 Joins

	Dynamic Change Tracking				Snapshot Change Tracking			
	Microsoft SQL Server		Maria DB		Microsoft SQL Server		Maria DB	
	Identity	GUID	Identity	GUID	Identity	GUID	Identity	GUID
Create	●	✗	✗	✗	✓	✗	✗	✗
Delete	✓	✗	●	●	●	✗	●	●
Update	✓	✓	●	●	●	●	●	●
Read	●	✓	✗	✗	●	✓	✗	✗

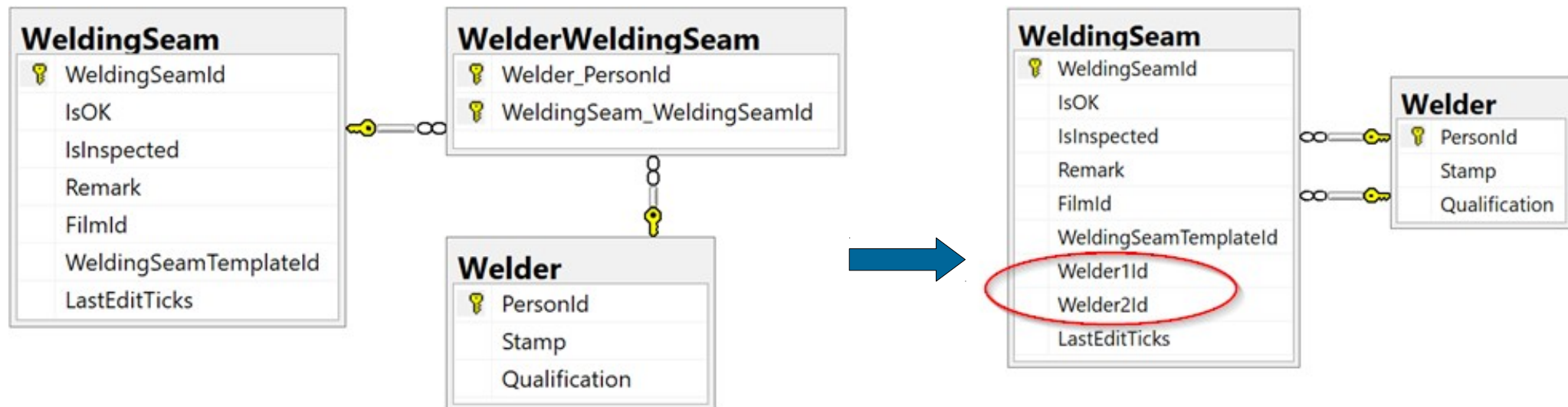


Die Testanwendung

Datenbank	Primärschlüssel	Datenbankschema	Change Tracking
<input checked="" type="radio"/> MS SQL Server 2016 <input type="radio"/> MariaDB	<input checked="" type="radio"/> Identity <input type="radio"/> GUID	<input checked="" type="radio"/> Normalisiert (TPT) <input type="radio"/> Denormalisiert TPH <input type="radio"/> Denormalisiert TPC <input type="radio"/> Denormalisiert TPT (Wiederholungsgruppe)	<input checked="" type="radio"/> Dynamic <input type="radio"/> Snapshot
Insert Einzufügende Elemente: <input type="text" value="0"/> Anzahl Durchläufe: <input type="text" value="0"/> Start mit leerer Tabelle: <input type="checkbox"/>	Update Zu bearbeitende Entitäten: <input type="text" value="0"/> Anzahl Durchläufe: <input type="text" value="0"/>	Delete Zu löschende Elemente: <input type="text" value="0"/> Anzahl Durchläufe: <input type="text" value="0"/>	
<div>Add (AutodetectChanges=On)</div> <div>Add (AutodetectChanges=Off)</div> <div>Add Range</div> <div>SQL Insert</div>	<div>Teste Update</div> <div>Teste Massenaktualisierung</div>	<div>Teste Delete</div>	
Select Anzahl Linien ($\text{Inspektionen} = \text{Linien} * 10^3$): <input type="text" value="0"/> Anzahl Durchläufe: <input type="text" value="0"/>	Vererbung Anzahl Durchläufe: <input type="text" value="0"/> Anzahl Personen pro konkretem Typ: <input type="text" value="0"/>	Wiederholungsgruppe Anzahl Durchläufe: <input type="text" value="0"/> Anzahl Inspektionen ($\text{Nächte} = \text{Inspekt.} * 100$): <input type="text" value="0"/>	TemplateTree Filme pro Prüfplan: <input type="text" value="0"/> Nächte prop Film: <input type="text" value="0"/> Anzahl Durchläufe: <input type="text" value="0"/> Anzahl konkrete Entitäten: <input type="text" value="0"/>
<div>Erstelle Testdaten</div> <div>Test LINQ</div> <div>Test SQL</div> <div>Test Cache</div>	<div>Teste Einfügen</div> <div>Teste polymorphe Suche (bool)</div> <div>Teste konkrete Suche (short)</div> <div>Teste polymorphe Suche (string)</div> <div>Teste konkrete Suche (string)</div>	<div>Teste Einfügen</div> <div>Teste Suchen</div>	<div>Erstelle Templates</div> <div>Test Erstelle konkrete Entitäten</div> <div>Test Lösche konkrete Entitäten</div>
Ergebnisse			

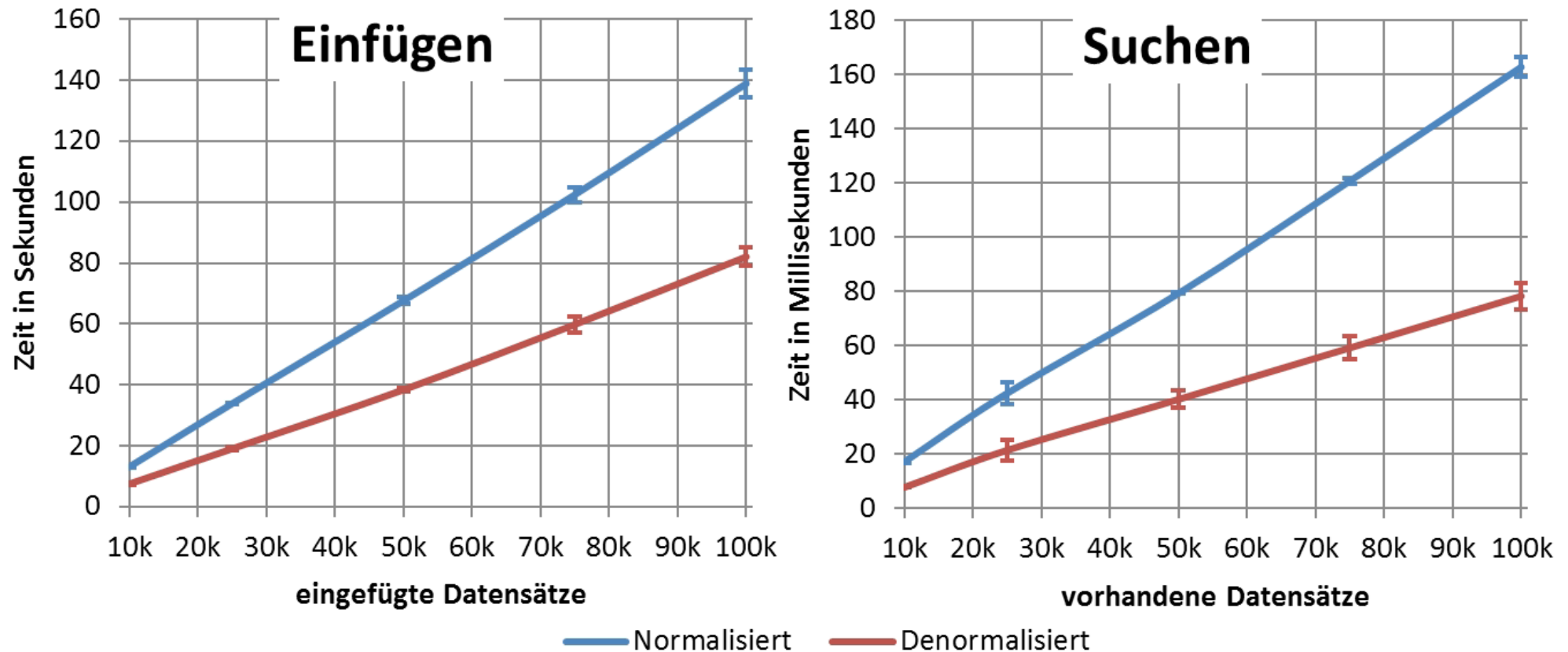
Denormalisierung

Wiederholungsgruppe



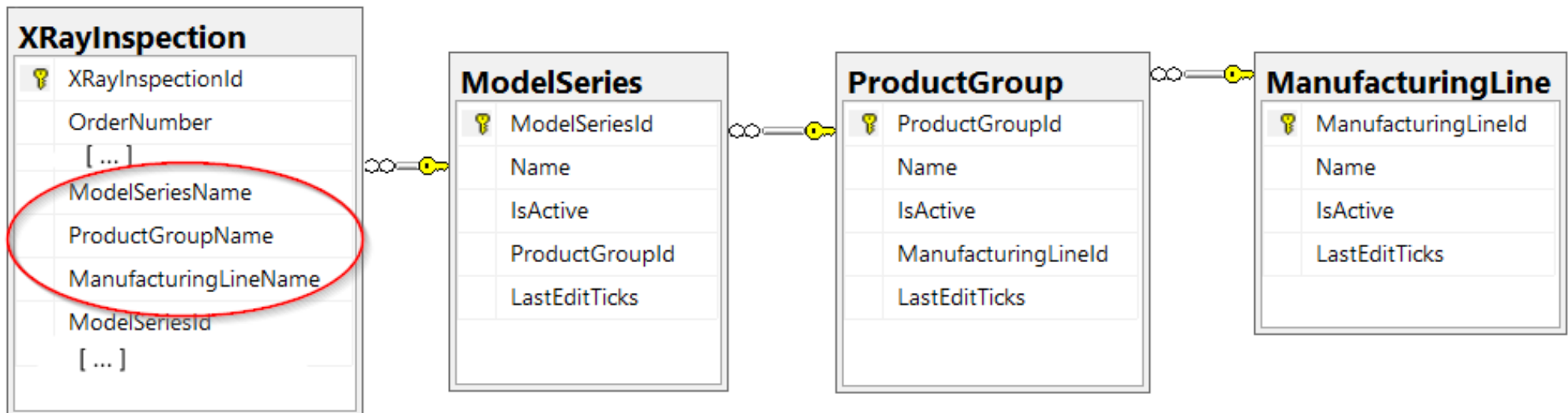
Denormalisierung

Wiederholungsgruppe



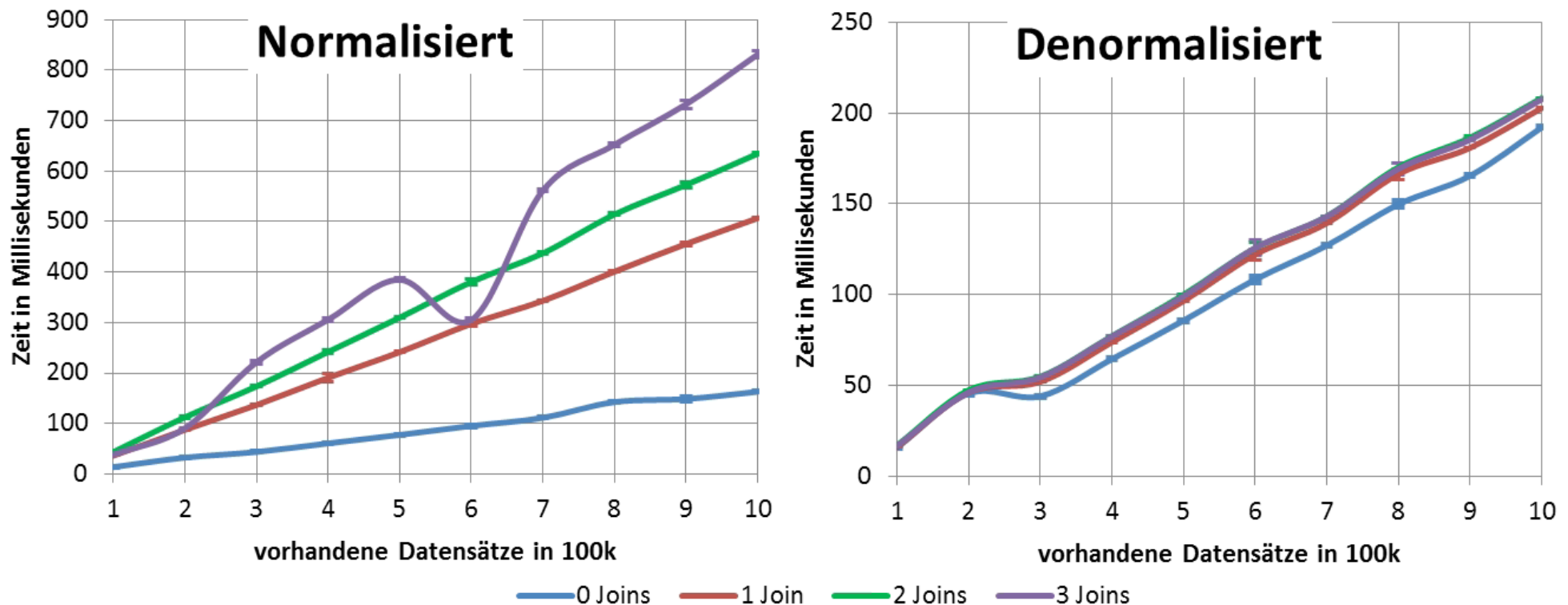
Denormalisierung

Redundanz in einer Beziehungskette



Denormalisierung

Redundanz in einer Beziehungskette



Vererbung

Table per Type

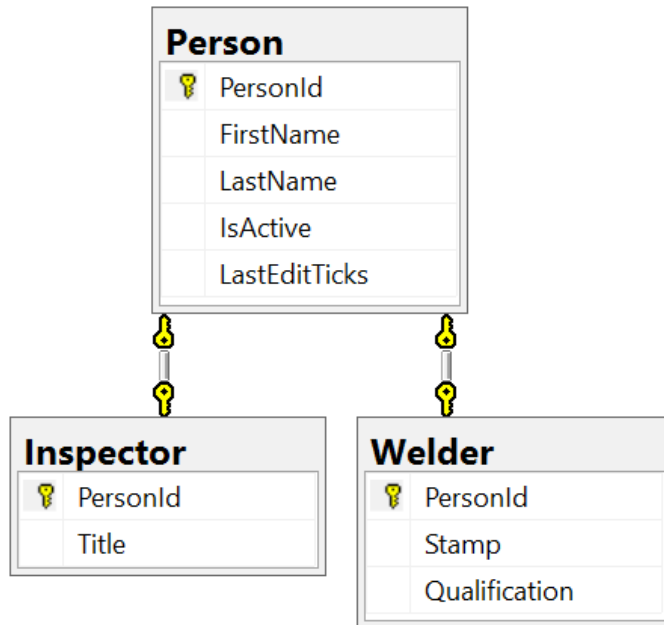


Table per concrete Type

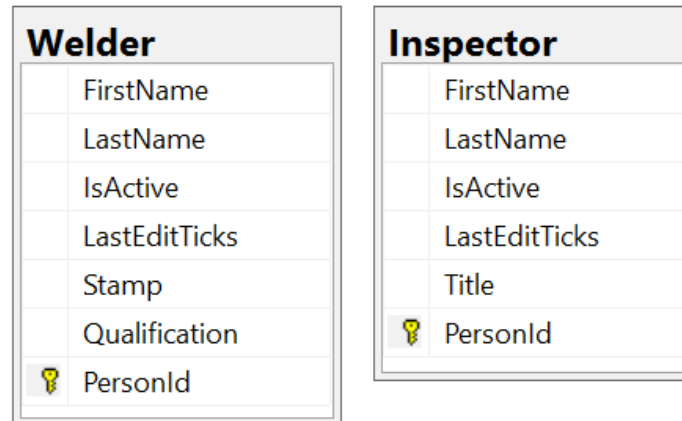
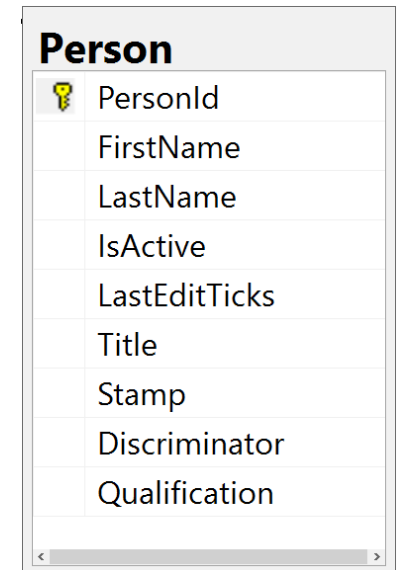
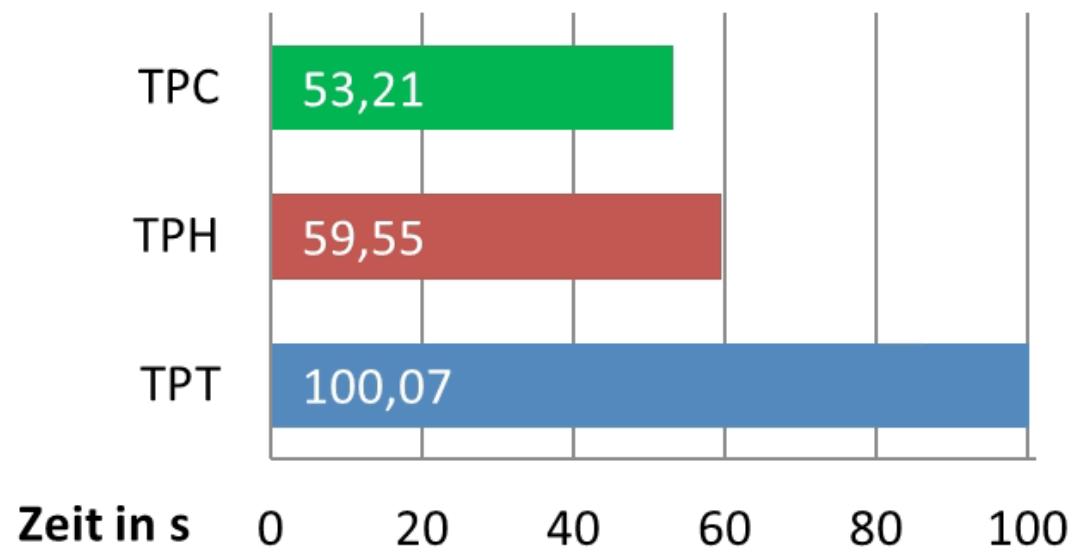


Table per Hirarchy

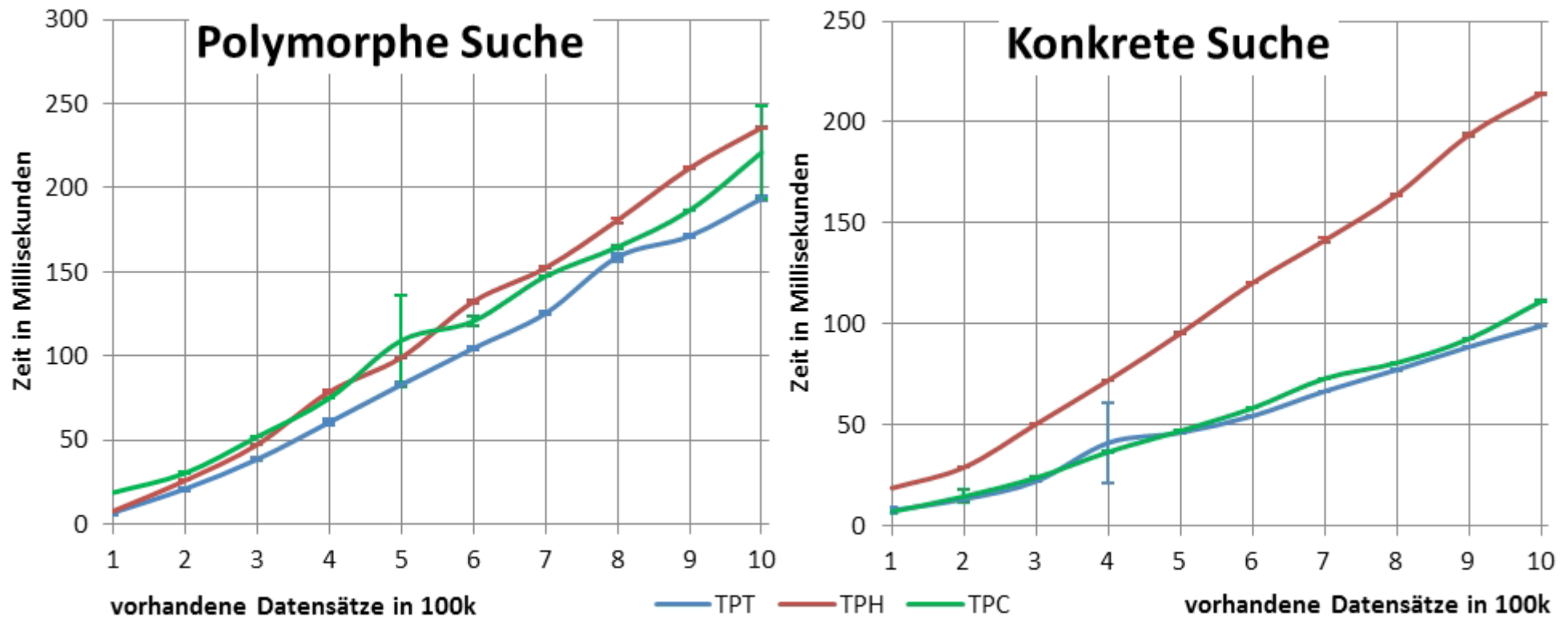


Vererbung

Einfügen von 100k Datensätzen

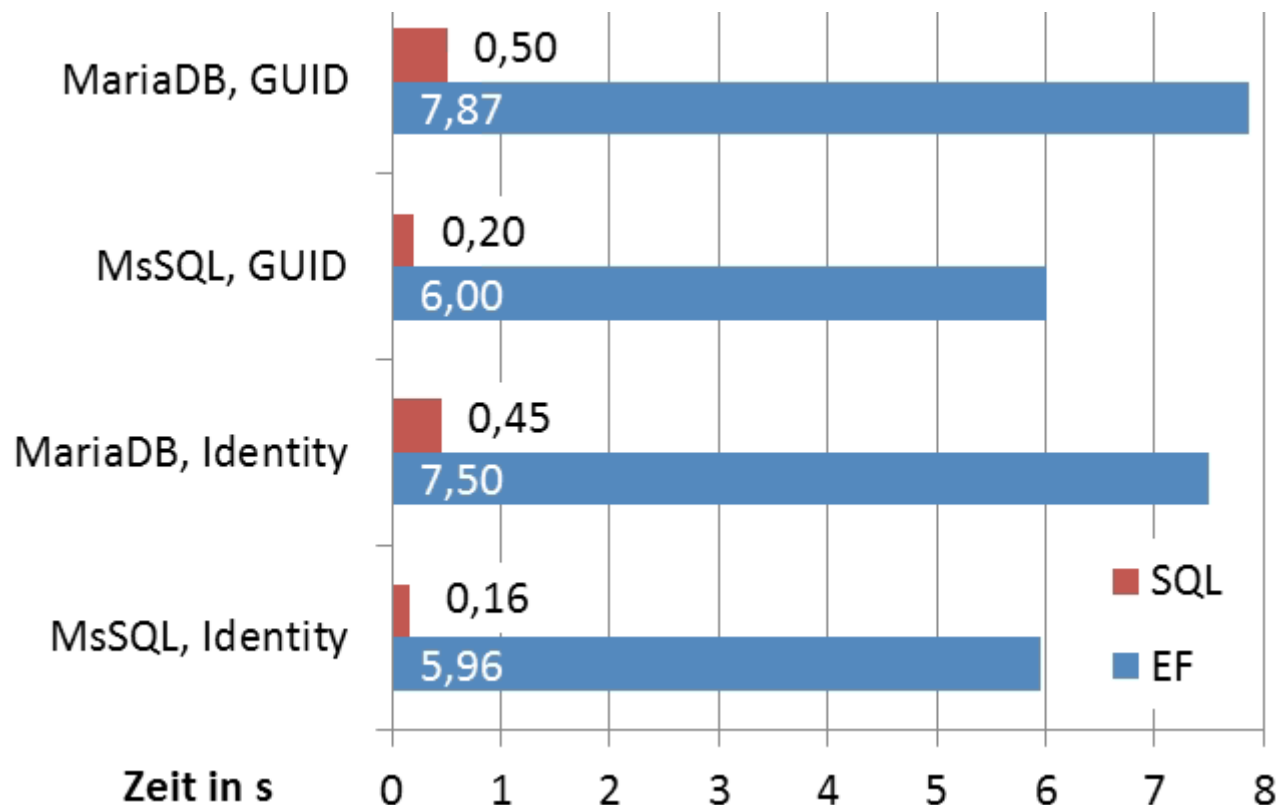


Vererbung



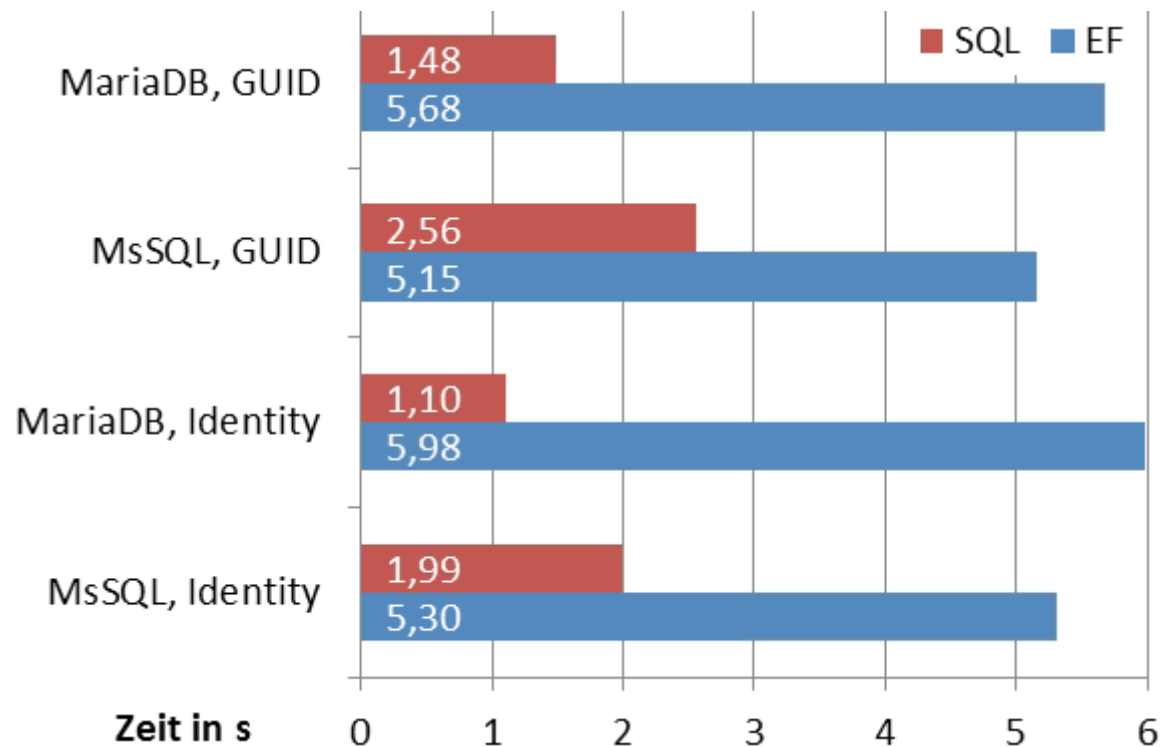
Natives SQL

Massenänderung von 10.000 Datensätzen



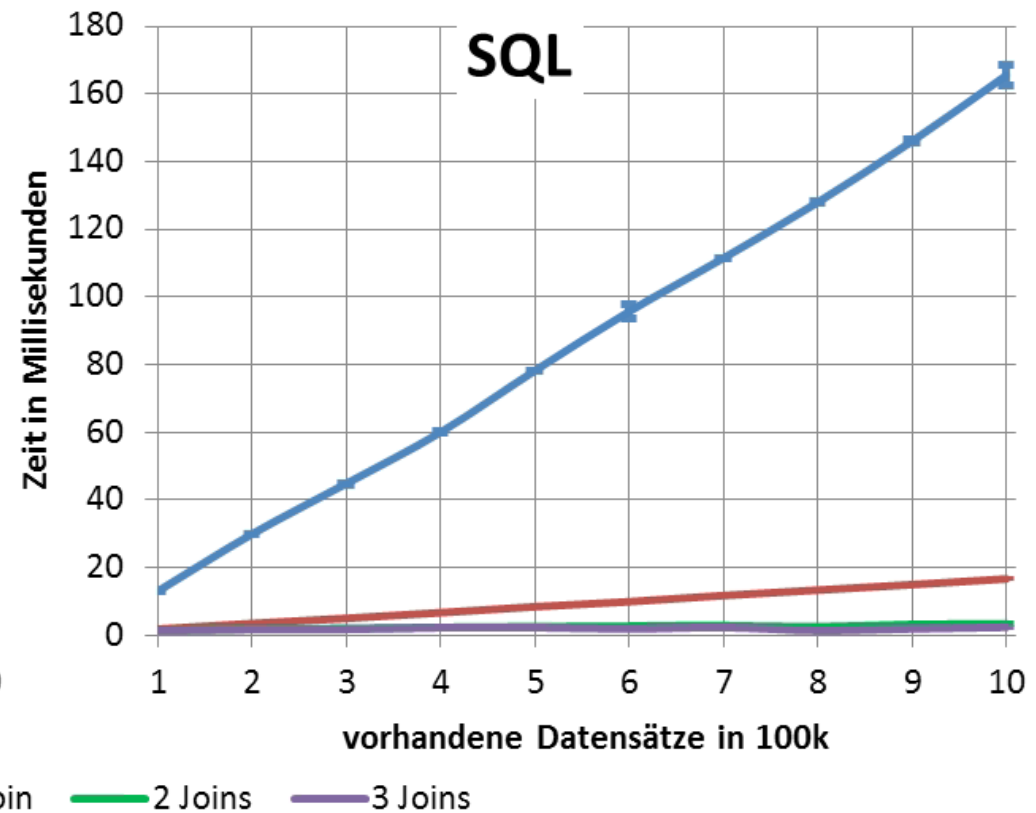
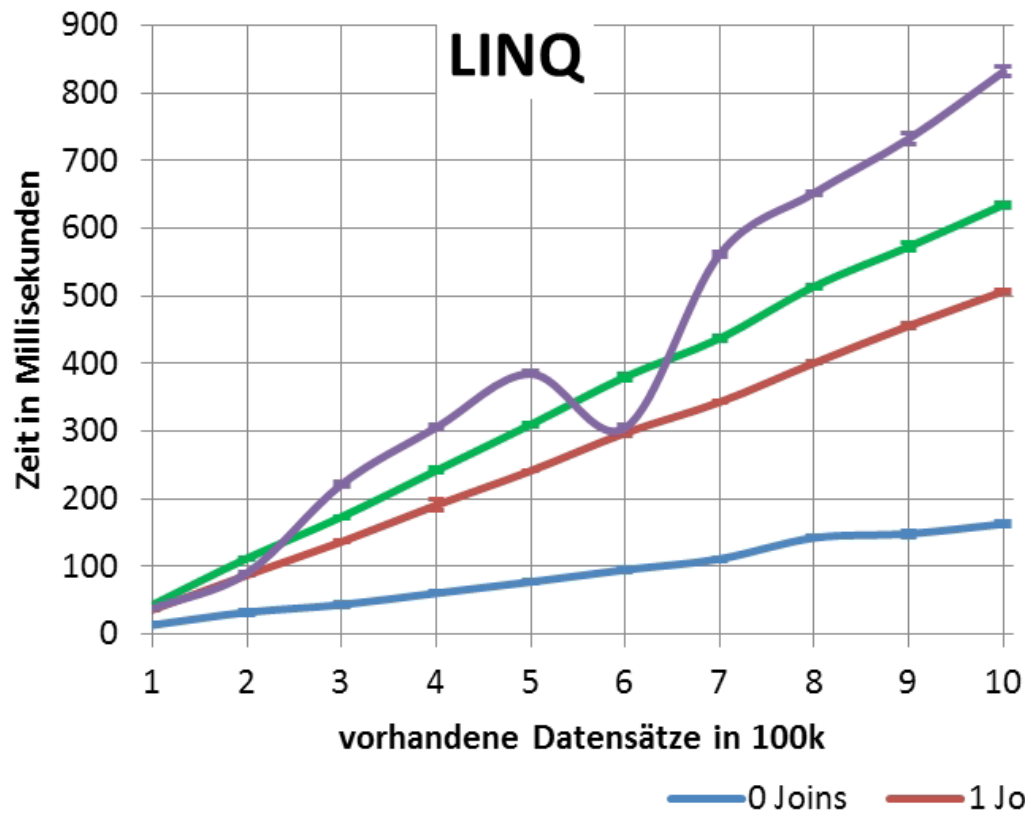
Natives SQL

Gleichzeitiges Einfügen von 10.000 Datensätzen



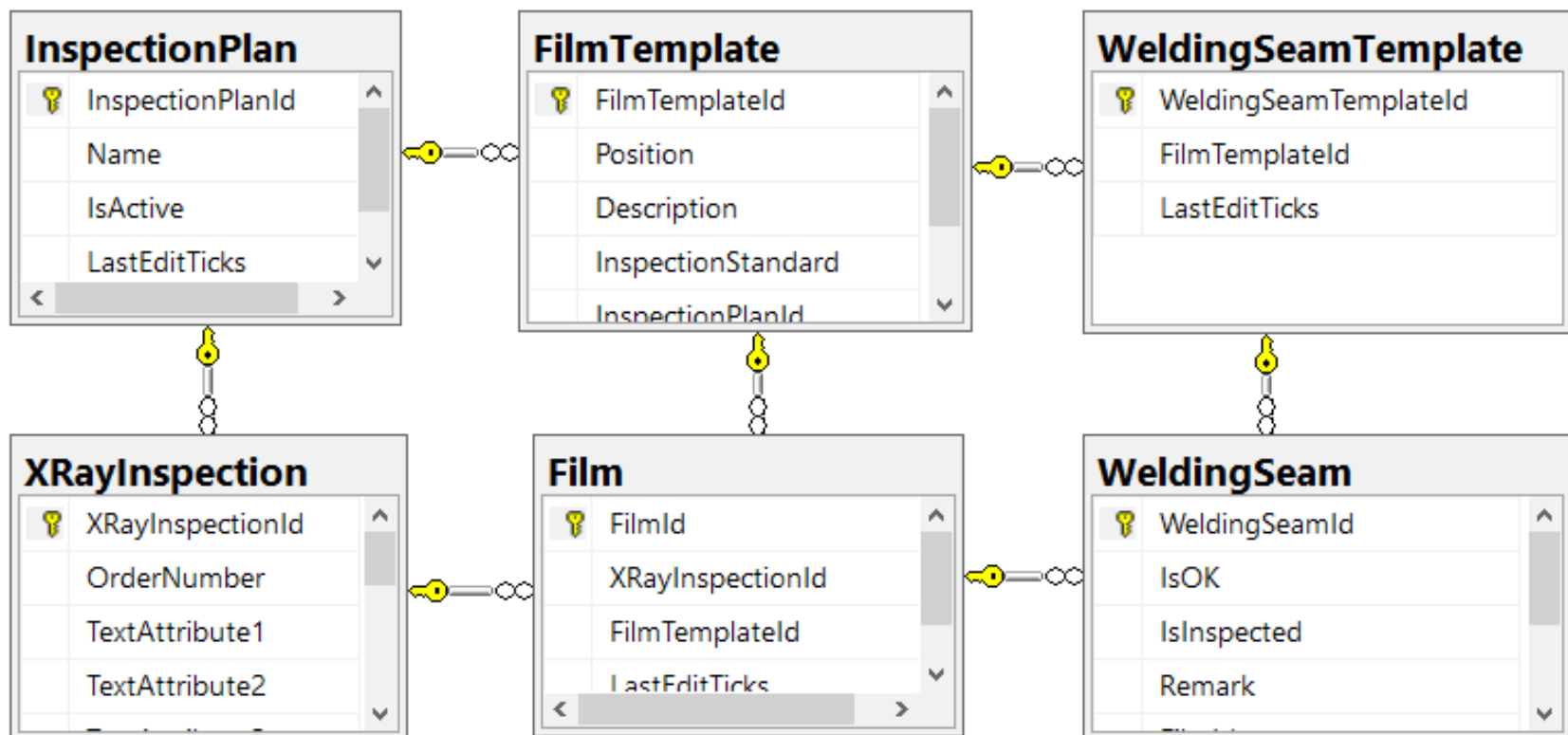
Massenvorgänge

Abfragen mit variablem Datenbestand



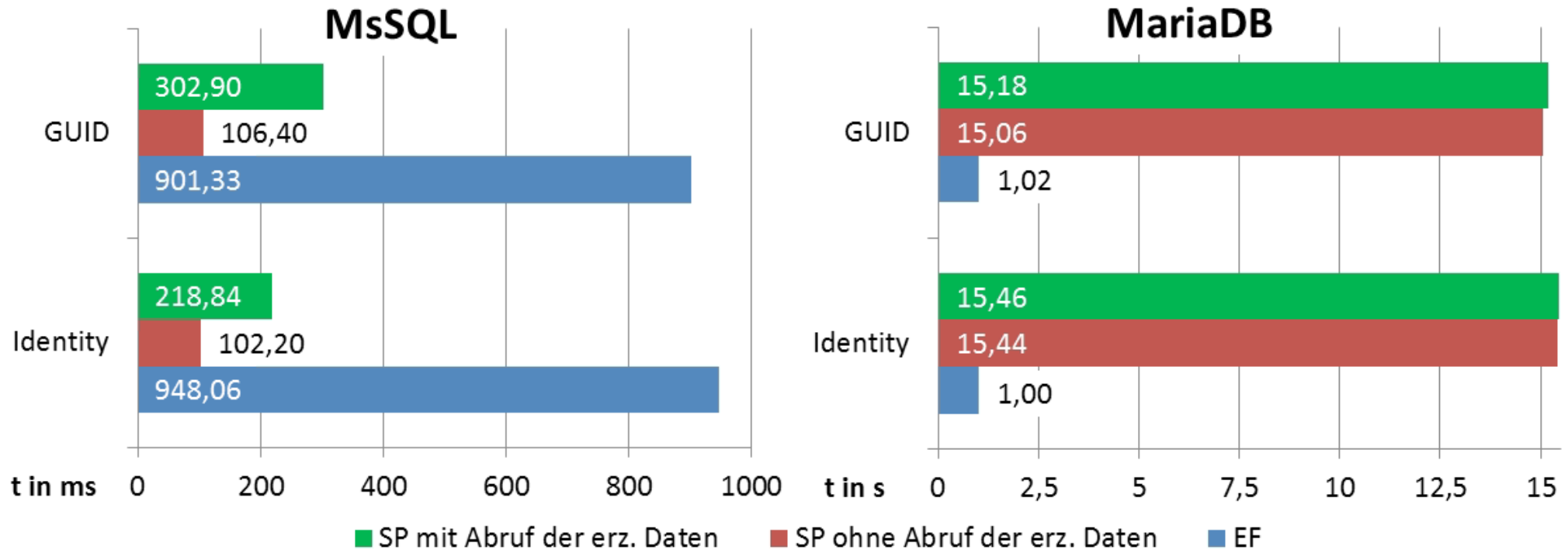
Komplexe Vorgänge

Template Tree



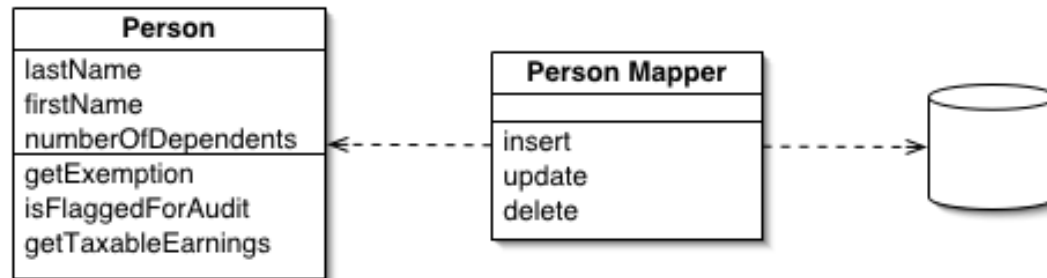
Komplexe Vorgänge

Template Tree

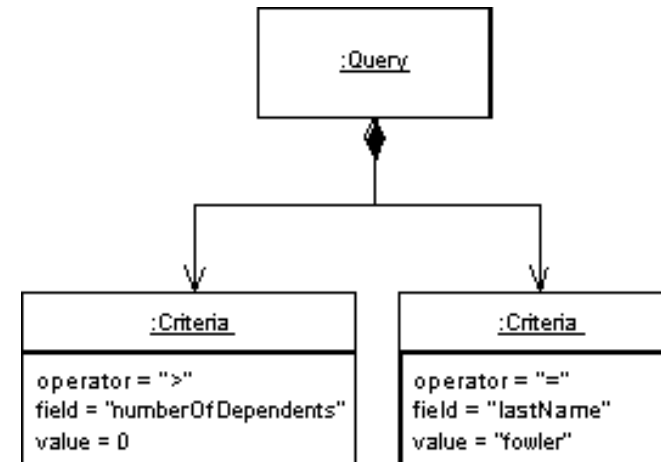


Im EF verwendete Entwurfsmuster

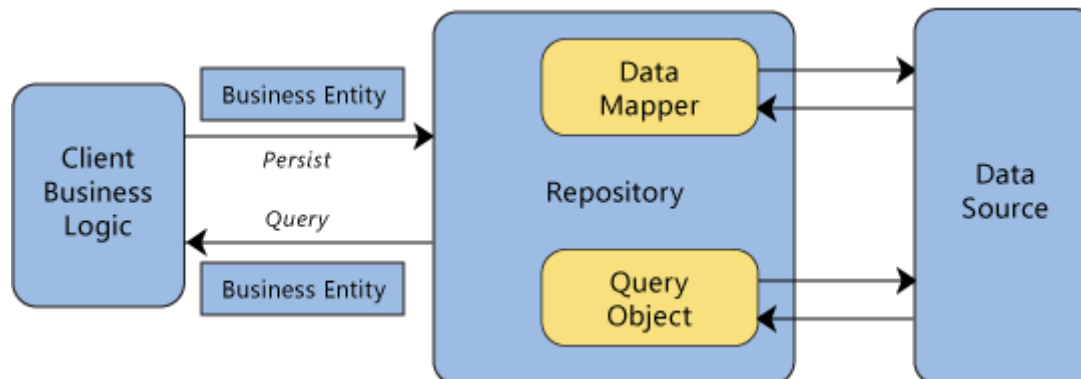
Data Mapper



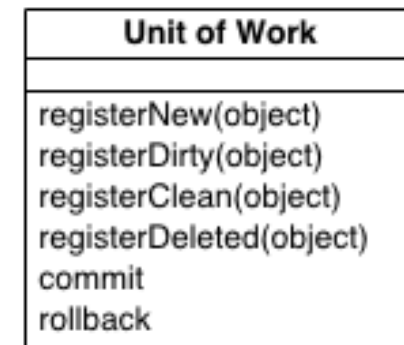
Query Object



Repository

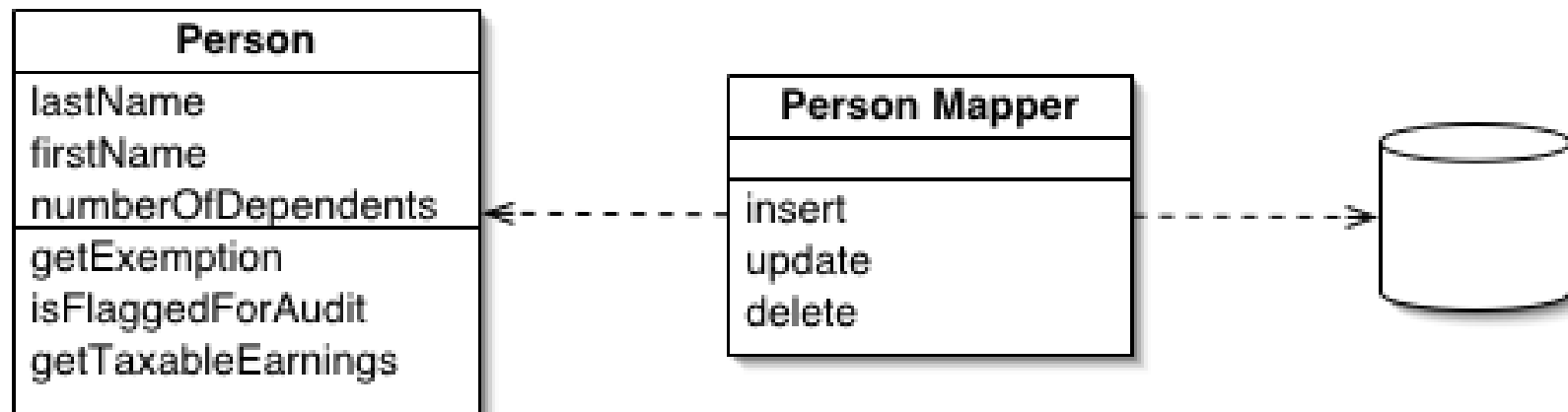


Unit of Work



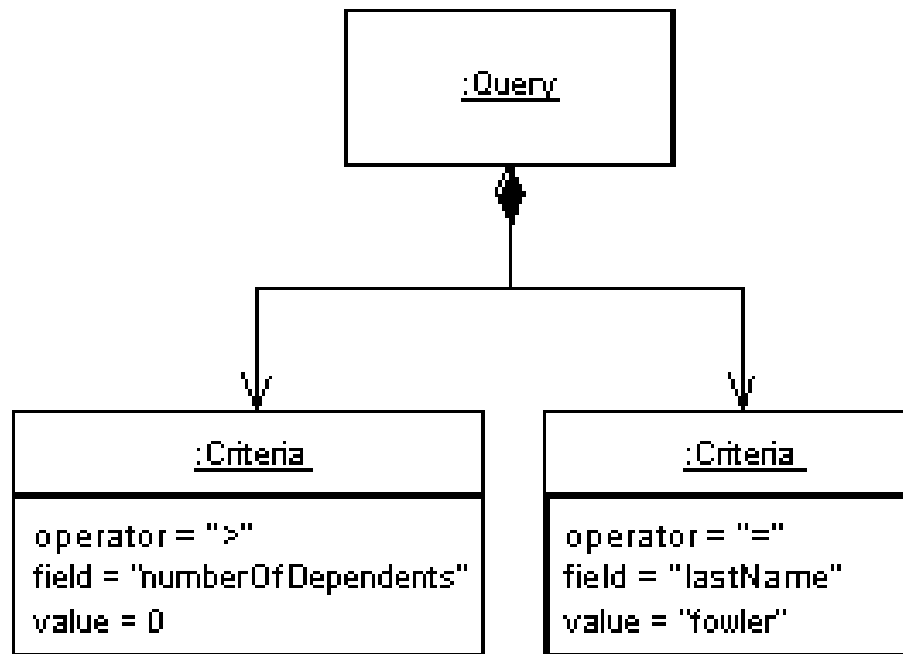
Im EF verwendete Entwurfsmuster

Data Mapper

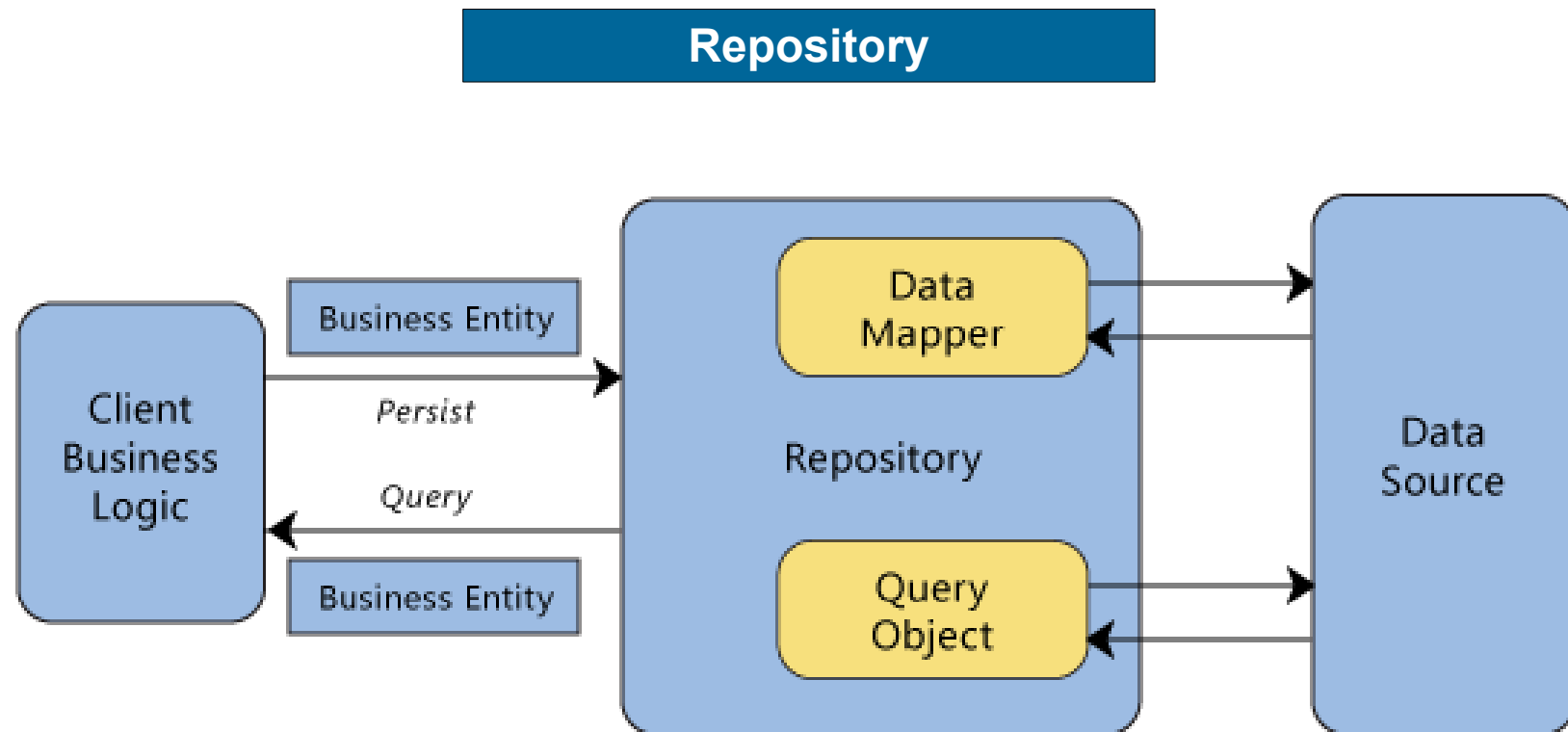


Im EF verwendete Entwurfsmuster

Query Object



Im EF verwendete Entwurfsmuster

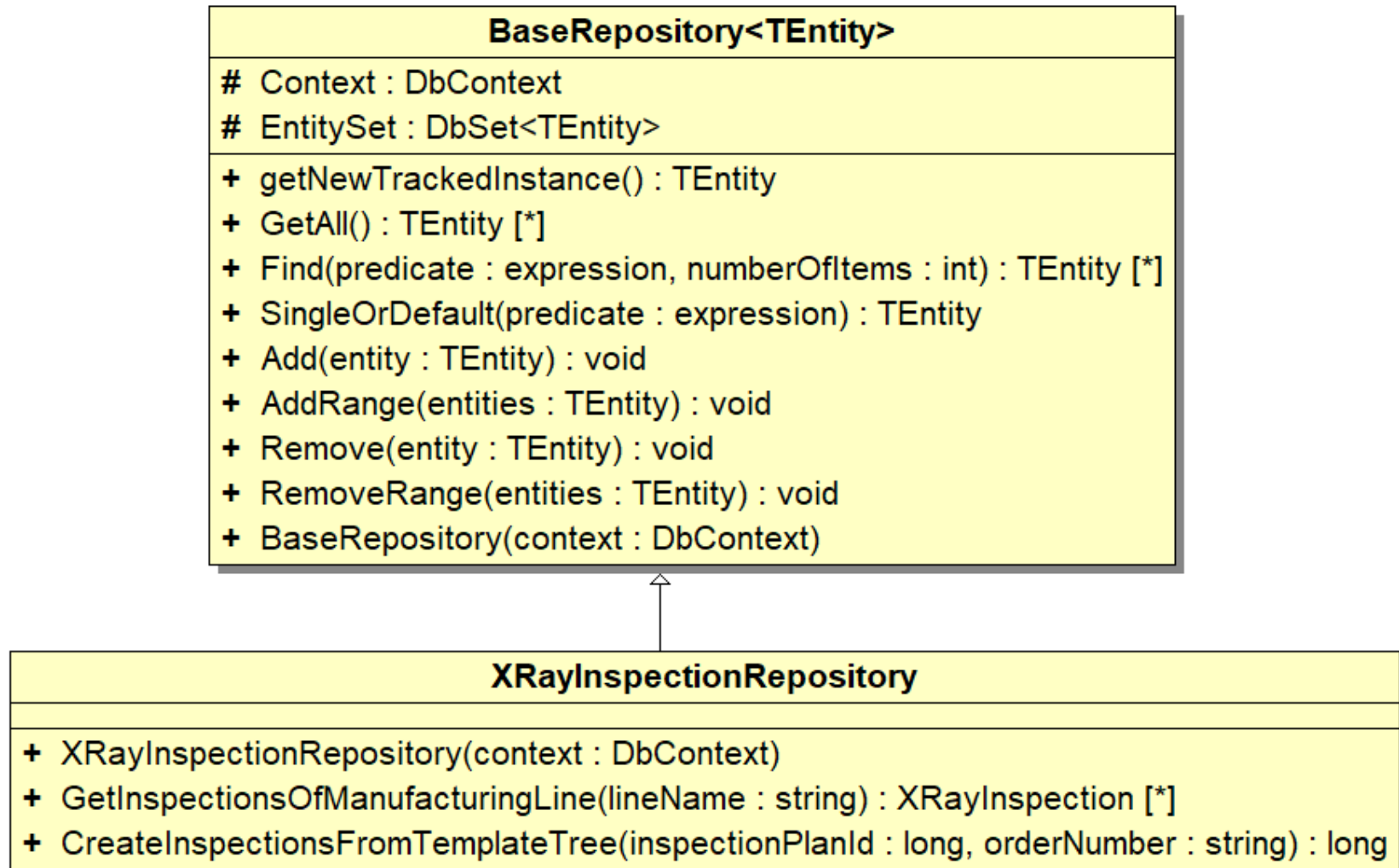


Im EF verwendete Entwurfsmuster

Unit of Work

Unit of Work
<code>registerNew(object)</code> <code>registerDirty(object)</code> <code>registerClean(object)</code> <code>registerDeleted(object)</code> <code>commit</code> <code>rollback</code>

Mustervorschlag: Repository Wrapper



Object-relational Impedance Mismatch

Realtionales Modell	Objektorientierte Programmierung
- Tupel enthält nur Daten	- Objekt enthält Verhalten und Daten
- Konzept der Vererbung nicht unterstützt	- Vererbung möglich
- Tupel hat eindeutige Identität, die sich anhand der Daten bestimmen lässt (Primärschlüssel)	- Objekt besitzt von Daten unabhängige Identität
- strikte Trennung von Daten und Verhalten	- Kapselung: Zugriff auf Daten nur über wohl definierte Wege + Verhalten bei Änderungen beeinflussbar
- alle Operationen erfolgen mengenbasiert	- Netz von individuell miteinander agierenden Objekten