**Auswertung**

**Vergleich Insgesamt**

Hier der quantitative Vergleich der Datenbanksysteme im Allgemeinen:

Es wurde die verstrichene Zeit vom Bestätigen der Query durch den User bis hin zu Anzeige des Ergebnisses im GUI je System zehn Mal gemessen und daraus ein Durchschnitt gebildet worden.



Insgesamt stellt sich SAP Hana Express als schnellstes DB-System heraus. Cassandra ist am Langsamsten und Maria DB/ Memcache variiert in der Leistung. MSSQL stellt sich als zweitschnellstes System heraus.

Diese Abfragen haben wir extra für den Vergleich aller Systeme gewählt. Cassandra und Memcache konnten ab einer gewissen Größe der zu der analysierenden Datenmenge keine Ergebnisse mehr anzeigen.

Das Arbeiten in der Shell führte dann aber zu dem Problem, dass es keine Möglichkeit gab die Zeiten pro Abfrage zu sehen bzw. anzeigen zu lassen.

Folglich musste via CAPTURE-Befehl eine Textdatei angegeben werden, in der eine detaillierte Anzeige des Outputs samt Zeiten aufgelistet wird.

CAPTURE 'Logdatei.txt‘ ON

Leider führte das Anlegen der Logdateien zu leichten Verfälschungen bei den Abfragezeiten was den allgemeinen Vergleich jedoch nicht weiter beeinträchtigte, da Cassandra von den Zeiten her , sowieso um Längen geschlagen wird.

**SAP Hana Express im Quantitativen Vergleich zu MSSQL**

Hier sechs Vergleichs-Abfragen von SAP Hana und MSSQL. Es wurden sechs Querys getestet. Diese wurde falls möglich zehn Mal wiederholt:

1.

Select k."Name", k."Vorname", k."Kreditorennr", k."Land",   
b."Artbez", b."BetragGesamt", b."Menge", b."MwstGesamt", b."RabatMenge",   
l."Abholstationen", l."DauerDurchschnitt", l."Unternehmen",  
s."Leitername", s."Leitervorname"  
From "XSA\_ADMIN"."Kunde" k   
Inner Join "XSA\_ADMIN"."Bestellung" b On (k."Kunnr" = b."Kunnr")  
Inner Join "XSA\_ADMIN"."Lieferdienst" l On (b."Bestellnr" = l."Bestellnr")  
Inner Join "XSA\_ADMIN"."Standorte" s On (l."Liefernr" = s."Liefernr")  
Order by k."Name";

2.

Select k."Name", k."Vorname", k."Kreditorennr", k."Land",   
b."Artbez", b."BetragGesamt", b."Menge", b."MwstGesamt", b."RabatMenge",   
(b."Menge" \* b."BetragGesamt") As SumRand,  
(b."BetragGesamt" \* b."MwstGesamt" - ((b."Menge" - b."RabatMenge")/100) \* b."BetragGesamt") As test,  
l."Abholstationen", l."DauerDurchschnitt", l."Unternehmen",  
s."Leitername", s."Leitervorname"  
From "XSA\_ADMIN"."Kunde" k   
Inner Join "XSA\_ADMIN"."Bestellung" b On (k."Kunnr" = b."Kunnr")  
Inner Join "XSA\_ADMIN"."Lieferdienst" l On (b."Bestellnr" = l."Bestellnr")  
Inner Join "XSA\_ADMIN"."Standorte" s On (l."Liefernr" = s."Liefernr")  
Order by k."Name"

3.

SELECT AVG (cast (a."Preis" as bigint)) AS AVGLief, AVG (cast(b."Preis" as bigint)) AS AVGBestell, AVG (cast (c."Mitarbeiteranzahl" as bigint)) AS AVGMit   
 FROM "XSA\_ADMIN"."Lieferdienst" a, "XSA\_ADMIN"."Bestellung" b, "XSA\_ADMIN"."Standorte" c  
 WHERE c."Liefernr" = a."Liefernr" AND b."Bestellnr" = a."Bestellnr"  
GROUP BY a."Preis";

3.1.

SELECT AVG (cast (a."Preis" as bigint))/(SELECT SUM(cast("Preis" as bigint)) FROM "XSA\_ADMIN"."Bestellung") \* (SELECT AVG(cast("Preis" as bigint)) FROM "XSA\_ADMIN"."Bestellung") AS AVGLief, AVG (cast(b."Preis" as bigint)) AS AVGBestell, AVG (cast (c."Mitarbeiteranzahl" as bigint)) AS AVGMit   
 FROM "XSA\_ADMIN"."Lieferdienst" a, "XSA\_ADMIN"."Bestellung" b, "XSA\_ADMIN"."Standorte" c  
 WHERE c."Liefernr" = a."Liefernr" AND b."Bestellnr" = a."Bestellnr"  
GROUP BY a."Preis"

4.

SELECT sum(cast(s."Fuhrparkgröße" as bigint)) AS "SummeFuhrparkgröße", b."Datum"  
 FROM "XSA\_ADMIN"."Bestellung" b , "XSA\_ADMIN"."Standorte" s  
 Where b."Datum" = '2003-07-08'  
GROUP BY "Datum"

5.

Select AVG ("RabatKunde") AS DurchschnittlicherRabattKunde, k."Name", l."Liefernr"  
 FROM "XSA\_ADMIN"."Lieferdienst" l , "XSA\_ADMIN"."Kunde" k Join "XSA\_ADMIN"."Bestellung" b ON (k."Kunnr" = b."Kunnr")   
Group By k."Name", l."Liefernr"

6.

Select AVG (cast (l."Bestellnr" as bigint)) As AVGBestellProLief, l."Unternehmen"  
FROM "XSA\_ADMIN"."Bestellung" b   
JOIN "XSA\_ADMIN"."Lieferdienst" l ON(b."Bestellnr" = l."Bestellnr")  
Group BY l."Unternehmen"

Dabei konnte SAP Hana Express bei Abfrage 1, 2 und 5 kein vollständiges Ergebnis ausgeben.

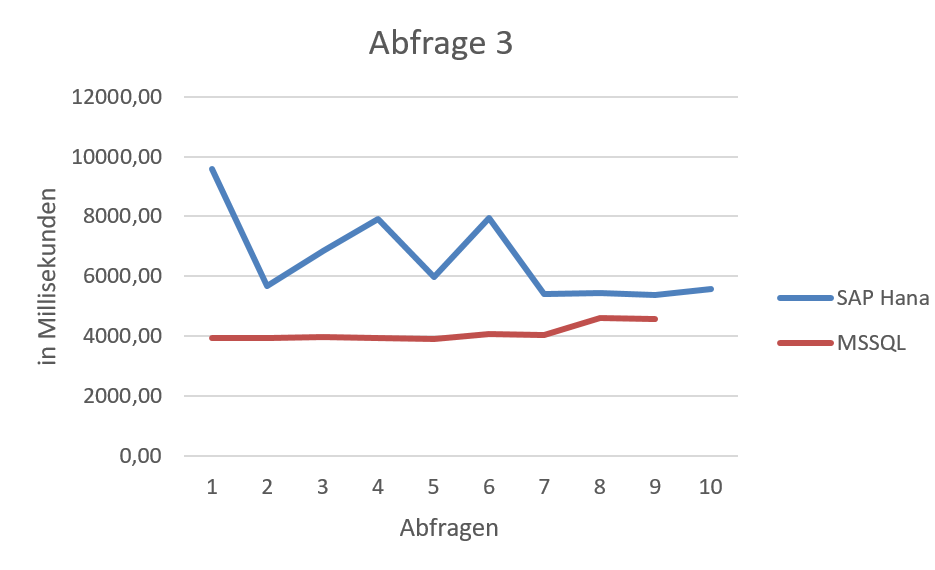
Wir vermuten, dass der Arbeitsspeicher voll ausgelastet war und keine Weiteren Ergebnisse mehr dort gespeichert werden konnten ehe alte Daten gelöscht wurden.

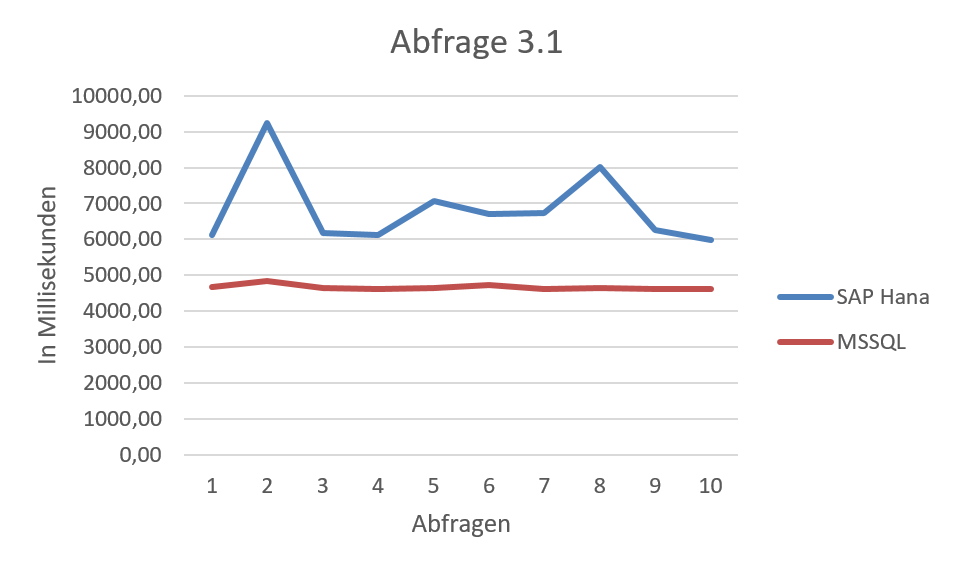
Als wir uns bei den Kritischen Abfragen über den Taskmanager die Resourcenauslastung des PCs haben anzeigen lassen, bestätigte dies die Theorie. Der Arbeitsspeicher war komplett ausgelastet.

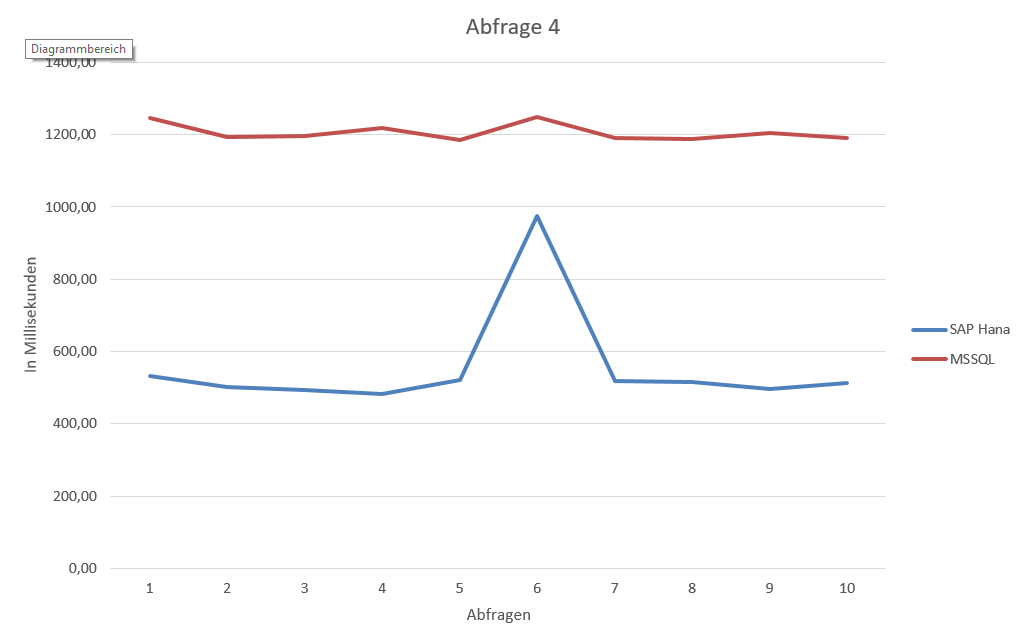
Da man SAP Hana Express nur maximal 30 GB Arbeitsspeicher zuweisen kann, haben wir also die Grenze von SAP Hana Express ausgelotet.

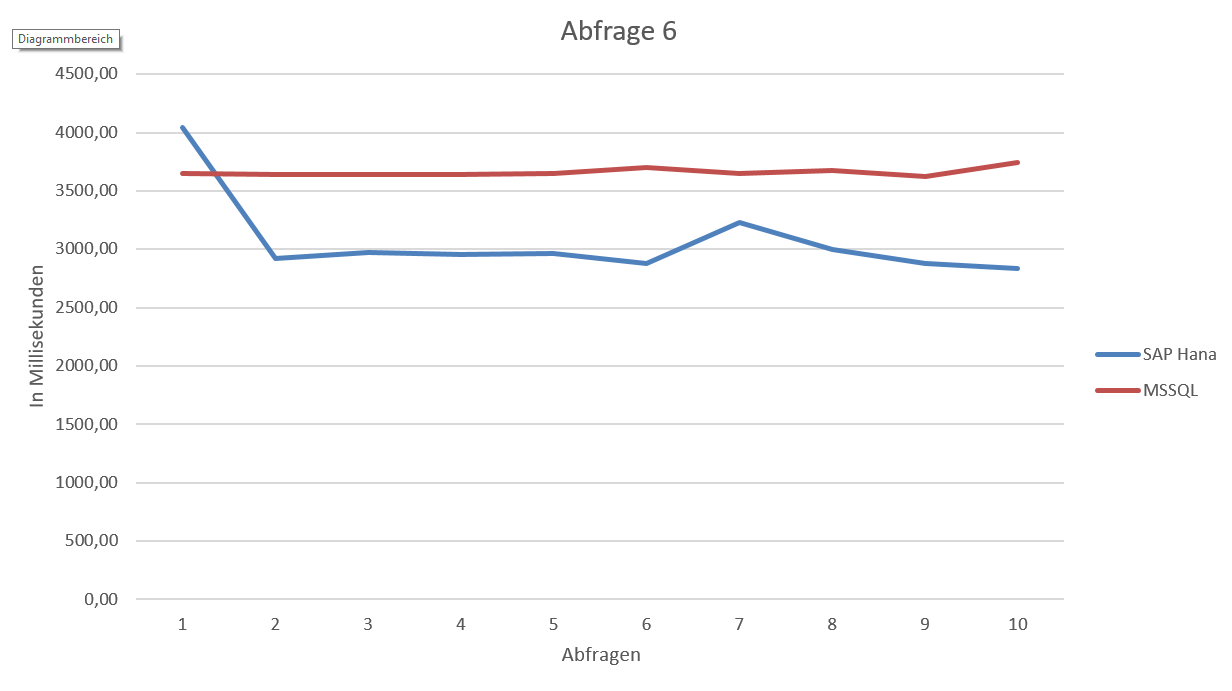
Sobald die Anzahl der Ergebnisdatensätze zu hoch wird kann SAP Hana Express die Abfrage nicht mehr zuverlässig und schnell bearbeiten.

Die Ergebnisse der durchführbaren Abfragen haben wir in Diagrammen zusammengetragen:









Bei sehr komplexen Abfragen kann nur noch MSSQL Server 2016 zuverlässig Ergebnisse liefern. Aber selbst bei solch komplexen Abfragen kann SAP Hana Express teilweise durch schnellere Abfragezeiten glänzen.

Grade bei Abfrage 6 und Abfrage 3 kann man die Vorteile von SAP Hana Express sehen. Es wird wie bei den anderen InMemory-DB-Systemen Logdateien der Abfragen erstellt, um die Verfügbarkeit nach einem Absturz des PCs sicherzustellen. SAP Hana Express kann diese Logdateien dazu nutzen seine Performance noch weiter zu optimieren in dem es auf diese Logdateien zugreift und damit die Abfrage löst. Somit werden bei wiederholter Durchführung der gleichen Abfrage die Antwortzeiten des Programmes noch weiter verringert.

**Qualitativer Vergleich der DB-Systeme**

**SAP Hana Express:**

**Vorteile:**

SAP Hana Express bietet gute und vielfältige Komprimierungsverfahren und bietet dem Nutzer die angewendeten Verfahren zu bestimmen oder zu verändern. Somit lässt es sich besser auf die direkte Aufgabe des DB-Systems im Anwendungsfall einstellen.

Durch die Nutzung von Eclipse war das GUI von SAP Hana Express relativ intuitiv und man konnte sich schnell zurechtfinden. Es ist nah an die GUI von ähnlichen Programmen angelegt.

Ein Pluspunkt ist natürlich auch, dass SAP Hana Express kostenlos zur Verfügung gestellt wir.

**Nachteile:**

Es gibt nur eine Linux Version des Programms und man ist als Windows-Nutzer dazu gezwungen eine VM zu nutzen.

Die Express-Version von SAP Hana deckt nur eine Nutzung von maximal 32 GB Arbeitsspeicher ab.

Die Syntax ist komplizierter und „unhandlicher“ gestaltet als die vergleichbarer Programme. So müssen Felder und Spaltennamen immer mit Anführungszeichen angegeben werden und die genutzte Datenbank muss jedes Mal aufgeführt werden.

**MSSQL Server 2016:**

**Vorteile:**

Die bekannte Oberfläche sowie die gewohnte einfache Syntax von MSSQL tragen zu einer komfortablen und intuitiven Bedienung bei.

Beim Erstellen der Abfragen kam uns MSSQL Server 2016 am meisten mit seiner Robusten Fehlerhilfe, den vielen vorgefertigten Funktionen und ähnlichen Hilfen sehr entgegen.

Insgesamt ist MSSQL ein guter Allrounder das eine breite Auswahl an Hilfsmitteln und Tools bereitstellt.

Nachteile:

Im Vergleich zu SAP Hana Express war keine Einflussnahme des Nutzers auf die Komprimierung möglich und auch sonst hielt sich Microsoft auch mit Hintergrundinformationen zurück.

Einschränkungen zu Aufrechterhaltung der referentiellen Integrität konnten nicht auf InMemory-Datenbanken angewendet werden, so wurden Check, Foreign Key und Unique nicht unterstützt.

**Cassandra:**

**Vorteile:**

Eine Horizontale Skalierbarkeit der Hardware Ressourcen ermöglicht einen Einsatz von Cassandra unabhängig von der Größe der Daten.

Cassandra setzt auf eine Vermeidung unnötiger Komplexität sowie die Vermeidung von relationalen Ansätzen des Datenmappings.

Eine Replikation der Datenbank ist sehr einfach möglich.

Es handelt sich um eine kostenlose Open Source Software.

**Nachteile:**

Ein Mangel an umfangreichen Dokumentationen machte die Arbeit mit Cassandra eher schwierig.

Es handelt sich nicht um eine universelle Sprache wie SQL.

Teilweise unerwartetes Verhalten und fehlender Support erschweren den Umgang mit der Software.

**Memcache:**

**Vorteile:**

Eine sehr einfache Installation sowie Implementation machen Memcache auch für Einsteiger gut Nutzbar.

Auch in der Bedienung ist Memcache sehr einfach und zeichnet sich durch simple Befehle wie:

Set(), add() und get() aus.

Es handelt sich um eine kostenlose Open Source Software.

**Nachteile:**

Bei Open Source können auch immer Fehler im Code ausgenutzt werden.

Überhaupt wird die Sicherheit komplett dem Nutzer überlassen und kann daher zu Fatalen Sicherheitslücken führen.

Die Kompression ist von den genutzten Bibliotheken und Programmiersprachen abhängig, weshalb ein Wechsel dieser innerhalb der Datenbank nicht unterstützt wird und Fehlfunktionen auslösen kann.