Dauer der Klausur: 120 Minuten

Punkte der Klausur: 100

Bepunktung

|  |  |  |
| --- | --- | --- |
| Aufgabe | Punkte | Erreichte Punkte |
| A1 | 15 |  |
| * A1.a | 4 |  |
| * A1.b | 6 |  |
| * A1.c | 4 |  |
| * A1.d | 1 |  |
|  |  |  |
| A2 – DBS, SQL, Datenbereinigung und Provenance | 35 |  |
| * A2.a | 12 |  |
| * A2.b | 10 |  |
| * A2.c | 13 |  |
|  |  |  |
| A3 – IT-Management | 12 |  |
| * A3.a | 2 |  |
| * A3.b | 2 |  |
| * A3.c | 8 |  |
|  |  |  |
| A4 – Hadoop und Streams | 8 |  |
| * A4.a | 6 |  |
| * A4.b | 2 |  |
|  |  |  |
| A5 – Cloud Computing | 12 |  |
| * A5.a | 8 |  |
| * A5.b | 4 |  |
|  |  |  |
| A6 – Python | 8 |  |
|  |  |  |

# A1: Einführung, Herausforderungen (LE1, LE2) – 15 Punkte

1. Nennen und beschreiben Sie zwei Gründe, warum Daten- und Prozessintegration in Unternehmen eine große Rolle bei der Erstellung betrieblicher Informationssysteme spielen (4 Punkte):

Grund 1: Kooperation neuer Informationssysteme mit Legacy-Systemen

Grund 2: Integration neuer Datenformate, z.B. JSON

1. Definieren Sie mit eigenen Worten den Begriff „Homegrown Integration“ und erläutern Sie stichpunktartig, welche Probleme dadurch entstehen (2 Probleme) (6 Punkte).

Definition

Problem 1:

Problem 2:

**Homegrown Integration**

* Backendsysteme rufen sich synchron gegenseitig auf, um Daten auszutauschen.
* Zwischenspeicher (Dateien oder Datenbanken) werden verwendet, um Daten asynchron auszutauschen. Der Speicherort wird vorab vereinbart.
* Wenn drei und mehr Systeme involviert sind, muss jedes Paar einen eigenen Speicherort für den Austausch von Daten definieren.
* Letztendlich müssen die Backendsysteme sich selbst um den Datenaustausch kümmern.

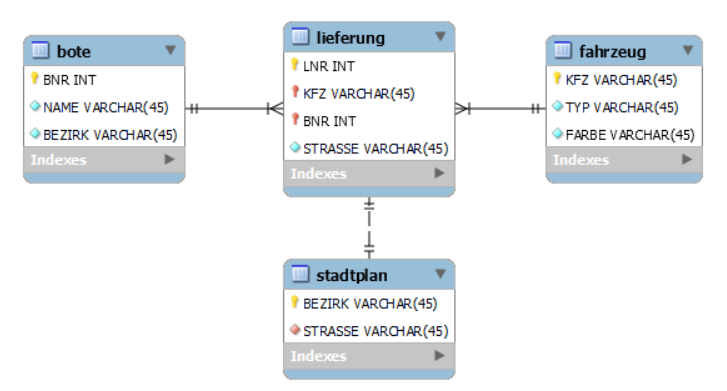
Die Probleme aus der "Homegrown Integration" führten zur Entwicklung der ersten Integrationssoftware-Systeme.

(Bussler 2003)

1. Erklären Sie den Unterschied zwischen einer „virtuellen Integration“ und einer „materialisierten Integration“ (DWH). (4 Punkte)
2. In welcher Integrationsform (virtuell, materialisiert) ist die Datenaktualität tendenziell höher. ( 1 Punkt)

In der virtuellen, da diese live Daten von den Quellen abruft.

# A2: Datenbanken und SQL, Datenbereinigung, Data Provenance (LE3) - 35 Punkte



|  |  |  |
| --- | --- | --- |
| bote | lieferung | fahrzeug |
|  |  |  |
| Hinweis: BEZIRK ist der Sitz des Pizzadienstes |  |  |
|  | stadtplan |  |
|  | Hinweis: Der Stadtteil (BEZIRK) enthält mehrere Straßen |  |

1. Das obige relationale Modell beschreibt die Datenbank eines Pizza-Lieferdienstes, der nur Pizzen ausliefert. Beschreiben Sie das Modell in folgender Form:

Die Relation „bote“ beschreibt einen Entity-Typen / Relationship-Typen und umfasst einen Primärschlüssel („BNR“). „bote“ wird von „lieferung“ referenziert („LNR“). Die Relation enthält keine Fremdschlüssel. Sie ist ein „Strong Entity“ der identifizierenden Beziehung zwischen „bote“ und „lieferung“. Die Relation umfasst zwei weitere Attribute namens „NAME“ und „BEZIRK“.

Die Relation „lieferung“ (4 Punkte) …

einfach

Die Relation „fahrzeug“ (4 Punkte) …

einfach

Die Relation „stadtplan“ (4 Punkte) …

einfach

1. Beschreiben Sie in eigenen Worten das Ergebnis der folgenden Abfragen (10 Punkte):
2. Abfrage 1 (2 Punkte)

select \*

from bote

order by `Name`;

Suche alle Einträge von Bote und ordne diese nach dem Namen

1. Abfrage 2 (4 Punkte)

select \*

from

bote join lieferung on bote.BNR = lieferung.BNR

join stadtplan on lieferung.STRASSE = stadtplan.STRASSE

where

bote.BEZIRK != stadtplan.BEZIRK;

Suche alle Lieferungen, bei denen der Bote nicht in seinen Heimatbezirk geliefert hat.

1. Abfrage 3 (2 Punkte)

select `Name`

from bote, lieferung, fahrzeug

where fahrzeug.FARBE ='rot' and

bote.BNR = lieferung.BNR and

fahrzeug.KFZ = lieferung.KFZ;

Suche den Namen aller Boten, die mit roten Fahrzeugen geliefert haben.

1. Abfrage 4 (2 Punkte)

insert into lieferung values

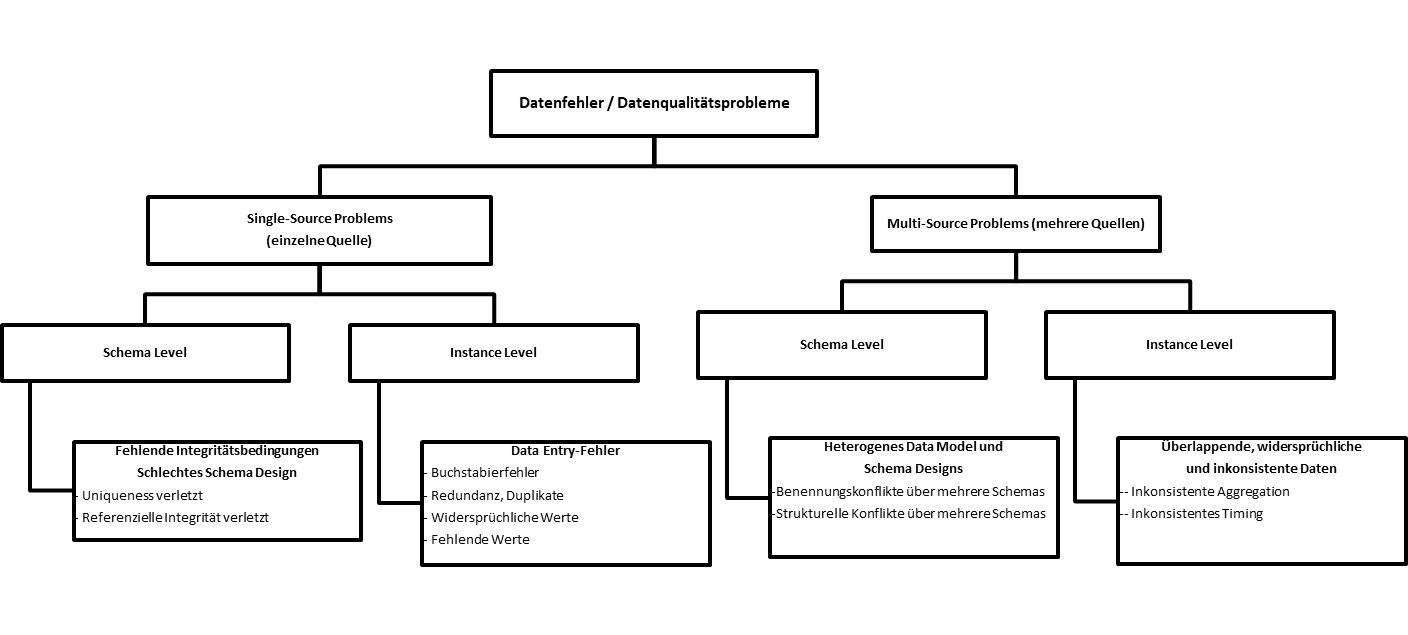
(5, 'WZ MG 44', 5, 'Oeder Weg');

Füge einen Eintrag in die Tabelle Lieferung ein, bei dem der Bote 5 mit dem Fahrzeug WZ MG 44 in den Oeder Weg liefert.

1. Data Provenance
2. Ordnen Sie zu: (3 Punkte)

|  |  |  |
| --- | --- | --- |
| Why-Provenance |  | Zusammenhang zwischen dem Quell- und Zielort an dem die Daten „residieren“ |
| How-Provenance |  | Wie wurden die Tupel kombiniert, um das Ziel zu produzieren? |
| Where-Provenance |  | Zusammenhang zwischen Quell- und Zieltupeln. Welche Tupel wurden verwendet? |

1. Datenqualitätsprobleme. Nennen Sie jeweils 2 Beispiele für folgende Datenqualitätsprobleme (8 Punkte):



Datenqualitätsprobleme bei einer einzelnen Datenquelle (Single Source):

Problem 1:

Problem 2:

Datenqualitätsprobleme bei mehreren Datenquellen (Multi Source):

Problem 1:

Problem 2:

1. Definieren Sie in eigenen Worten den folgenden Begriff (2 Punkte):

Scrubbing:

**Bereinigung der Quelle**

Unzulässige Attributwerte: z.B. Geburtsdatum 24.24.24

Fehlende Attributwerte/ Attributwertteile: z.B. unvollständige Telefonnummern

Wechselnde Eintragsformate bei Freiformfeldern: z.B. „Daimler“, „DB Stuttgart“, „Mercedes Benz AG, Stuttgart“ 🡪 häufige Fehlerquelle für Duplikate

Rechtschreibfehler

Kryptische Datenwerte: z.B. „Cert X-903“ für Qualifikation

Falsch erfasste Attribute: z.B. Straße = „Frankfurt“

Inkonsistenzen auf Datensatzebene: z.B. Alter 50, aber Geburtsdatum 30.12.2010

Inkonsistenzen des Zustands von Tabellen: z.B. doppelte Verwendung desselben Keys

Fehler in Fremdschlüsselbeziehungen: Verweise auf fehlende Datensätze

# A3: IT-Management (LE4) – 10 Punkte

1. Informationsmanagement (IM) ist ein Teilbereich des IT-Managements. Welche zentrale Aufgabe erfüllt das IM? (2 Punkte)

Planung, Steuerung und Kontrolle von Information, von Informationssystemen (IS) und von Informations- und Kommunikationstechnik (IKT)

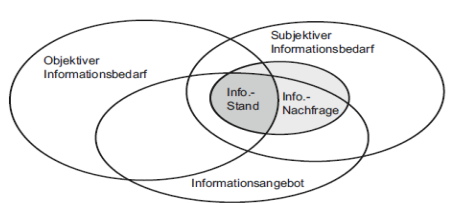
zentrale Aufgabe der Unternehmenskoordination: über die Koordination der Informationsströme die Koordination der Leistungserstellung erreichen.

1. Die Informationslogistik ist ein Teilbereich des IM. Was verstehen Sie unter diesem Begriff? (2 Punkte)

Die Informationslogistik ist als Teilbereich des Informationsmanagement anzusehen, welcher sich mit der Logistik des Objektes Informationen befasst.

Die Informationslogistik erleichtert die konzeptionelle Einbeziehung externer Informationen und Informationsquellen und liefert somit die Grundlagen zur Gestaltung der innerbetrieblichen wie auch zwischenbetrieblichen und branchenspezifischen Prozesse

1. Informationsbedarfsermittlung: Beschreiben Sie die folgende Grafik. Gehen Sie dazu auf die Begriffe "Informationsangebot", "Informationsbedarf", "Informationsnachfrage" und "Informationsstand" ein. (8 Punkte)



Objektiver I-Bedarf: für die Aufgabenerfüllung notwendig

Subjektiver I-Bedarf: vom Individuum für notwendig erachtet

Geäußerter Bedarf = Nachfrage

Informationsangebot = alle verfügbaren internen und externen Informationen

Informationsstand = die Informationen, die wirklich benötigt werden und verfügbar sind

# A4: Hadoop, Datenströme und Complex Event Processing (LE5) – 8 Punkte

1. Ordnen Sie zu (6 Punkte)

|  |  |  |
| --- | --- | --- |
| Hadoop Common |  | wird für das technische / organisatorische Management von Hadoop benötigt; Koordination der Bestandteile eines Hadoop-Systems; verwaltet "Konfigurationen", Namen(sräume) und Gruppen; koordiniert Systembestandteile |
|  |  |  |
| Hadoop YARN |  | Ein Dateisystem mit hohen Datenverarbeitungsraten; kann auf Server-Hardware und einfacher Hardware (Laptop) ausgeführt werden, z.B. auch in einer VM; Daten können über tausende Server verteilt werden |
|  |  |  |
| Hadoop Map Reduce |  | Verteilt Ressourcen (CPU, Speicher, …) zwischen mehreren Prozessen, z.B. MapReduce-Prozessen und/oder Frameworks wie MapReduce, Impala und Spark |
|  |  |  |
| HDFS |  | Algorithmus für parallele Datenverarbeitung und Verdichtung der Daten in "managebare" Portitionen, die zu Analysezwecken benötigt werden; Programmierkomponente von Hadoop |
|  |  |  |
| Hadoop HIVE |  | Data Warehouse auf Basis von Hadoop; Abfragesprache stark an SQL angelehnt |
|  |  |  |
| Hadoop Zookeeper |  | Basis für weitere Produkte; grundlegende Dienste und Prozesse; Abstraktionsschicht über dem Betriebssystem und Dateisystem |

1. Beschreiben Sie in eigenen Worten eine betriebliche Aufgabenstellung, die Sie mit Hadoop lösen können. (2 Punkte)

Mehrere Lösungen; grundsätzliches Ziel: Verarbeitung von großen Datenmengen

# A5: Cloud Computing (LE7) – 12 Punkte

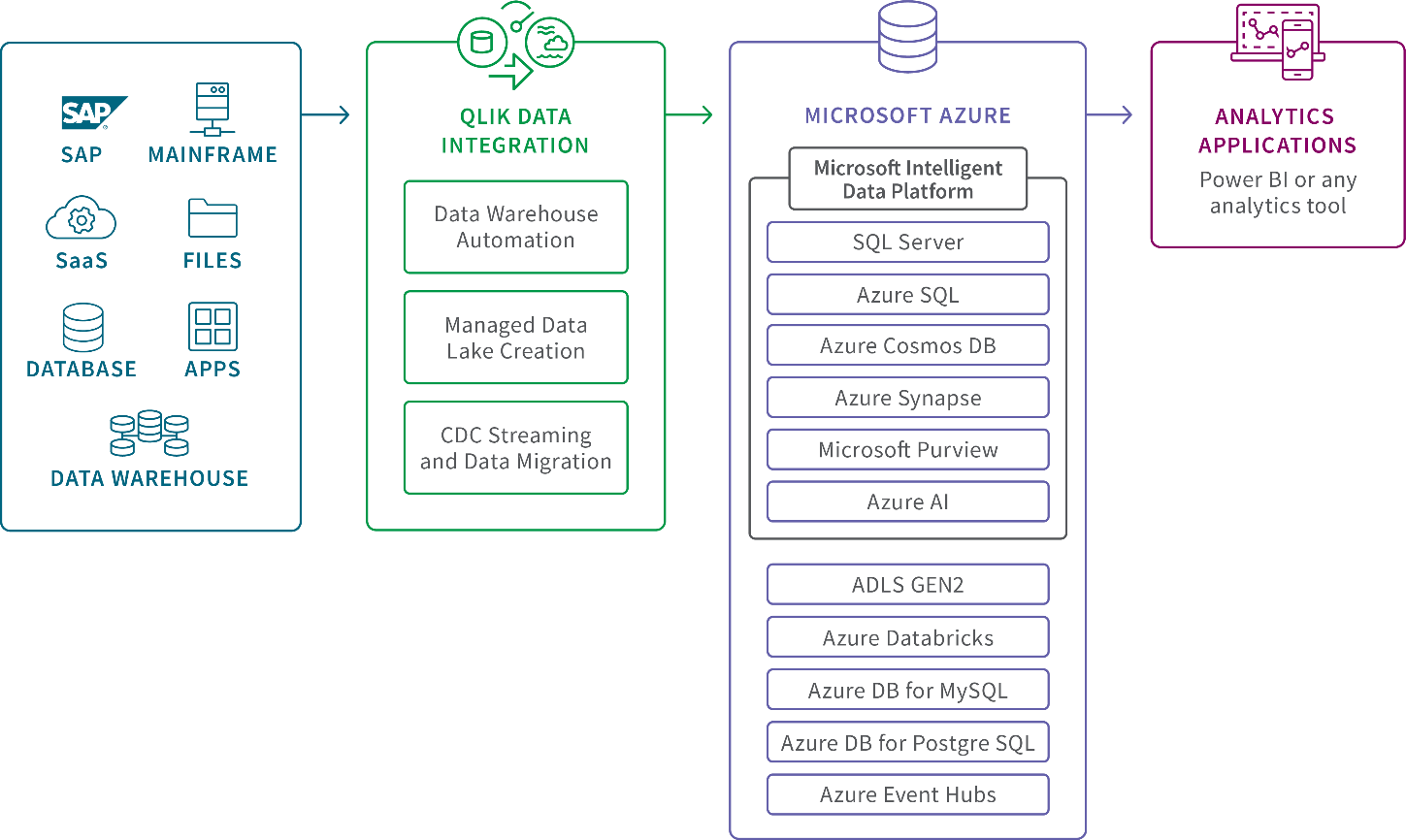


Abbildung 1: Qlik Data Integration - Quelle: https://www.qlik.com/us/products/technology/qlik-microsoft-azure-migration?msclkid=c1232af966721a127aaa4e6140514df9&utm\_team=DIG&utm\_subtype=cpc\_nb&ppc\_id=M9qnI658&kw=cloud%20data%20integration&utm\_content=sM9qnI658\_pcri

1. In der obigen Abbildung finden Sie eine Datenintegrations-Architektur eines Cloud-Providers. Beschreiben Sie in eigenen Worten, inwieweit sich diese Darstellung von einem klassischen Data Warehouse unterscheidet und welche Ähnlichkeiten bestehen. Gehen Sie dazu auf 4 Aspekte ein. (8 Punkte)

Erster Aspekt (Unterschied):

ein zentraler Datenspeicher in Form des Cores

Zweiter Aspekt (Unterschied):

nur ein "Stream" – hier mehrere (DWH Automation, Managed Data Lake Creation …)

Dritter Aspekt (Ähnlichkeit):

Quellsysteme werden mittels Integrationsprozessen geladen

Vierter Aspekt (Ähnlichkeit):

Reporting hier in Form von Analytics

1. Utility Computing

Beschreiben Sie, inwiefern Utility Computing und Cloud Computing miteinander zusammenhängen. Was bedeutet dies für Unternehmen, die Cloud-Dienste nutzen? (4 Punkte)

Utility Computing – Abrechnung der Verbräuche / Konsum von Utilities. Hier Dienste als Utilities

# A6: Datenverarbeitung mit Python (LE8) – 8 Punkte

# Bubble sort

def bubbleSort(array):

# loop to access each array element

for i in range(len(array)):

for j in range(0, len(array) - i - 1):

if array[j] > array[j + 1]:

temp = array[j]

array[j] = array[j+1]

array[j+1] = temp

data = [9, 10, 4, 5]

bubbleSort(data)

print('Sortiertes Array:')

print(data)

Der obige Code beschreibt den Sortieralgorithmus "Bubble Sort" Füllen Sie die unten stehende Tabelle aus.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | j = 0 ..  len(array) – i - 1 | | array[j] | array[j+1] | temp | array[j] - after | array[j+1] – after | arr[0] | [1] | [2] | [3] |
| 0 | 0 | 4-0-1 = 3 | 9 | 10 |  | 9 | 10 | 9 | 10 | 4 | 5 |
|  | 1 |  | 10 | 4 |  | 4 | 10 | 9 | 4 | 10 | 5 |
|  | 2 |  | 10 | 5 |  | 5 | 10 | 9 | 4 | 5 | 10 |
| 1 | 0 | 4-1-1 = 2 | 9 | 4 |  | 4 | 9 | 4 | 9 | 5 | 10 |
|  | 1 |  | 9 | 5 |  | 5 | 9 | 4 | 5 | 9 | 10 |
| 2 | 0 | 4-2-1 = 1 | 4 | 5 |  | 4 | 5 | 4 | 5 | 9 | 10 |
|  |  |  |  |  |  |  |  |  |  |  |  |

Bitte zur Probe in Python ausführen!

data[]= [4,5,9,10]