Punkte

Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4
/ 55	/ 15	/ 25	/ 5

Gesamtpunktzahl:

Name: Shada Maayouf Matrikelnummer: q4311663

Allgemeine Hinweise zur Klausur 01882 **Data Engineering für Data Science**

> ***NOTE:*** *Bitte lesen Sie sich diese Hinweise vor der Bearbeitung der Klausur aufmerksam durch.*

- Die **Klausurdauer** beträgt **2 Stunden**.
- Bevor Sie mit der Bearbeitung der Klausuraufgaben beginnen, tragen Sie bitte Ihre(n) **Nachnamen**, Ihre(n) **Vornamen** und Ihre **Matrikelnummer** in die vorgesehenen Felder (oben) ein.
- Wie schon in den Übungsaufgaben erfolgt die Bearbeitung in zwei unterschiedliche Zell-Typen: In **Code-Zellen** sollen Sie (Python-)Code (bei Bedarf kommentiert) einfügen und ausführen. In **Raw-Zellen** erwarten wir eine Antwort in Textform.
- Sollten bestimmte Fakten aus Ihrer Sicht nicht eindeutig formuliert sein, so schreiben Sie Ihre Annahmen zur Lösung dazu.
- Denken Sie bei Plots an Titel, Achsenbeschriftung und bei Bedarf Legende.
- Es empfiehlt sich, das Notebook, bzw. das daraus generierte PDF in regelmäßigen Abständen zu sichern. So müssen Sie nicht bei Null anfangen, sollte es unerwartetere Weise zu Software-Problemen kommen.
- Diese Klausur ist eine sogenannte "Open-Book-Klausur", was bedeutet, dass Sie alle Vorlesungsunterlagen, etc. als Hilfsmittel verwenden dürfen. Es ist jedoch ausdrücklich **verboten**, dass Sie untereinander kommunizieren oder andere Personen zu Rate ziehen. Explizit untersagt ist außerdem die Nutzung von AI-Chatbots wie beispielsweise ChatGPT.
- Es werden nur Lösungsabgaben im PDF-Format akzeptiert. Diese müssen mit der vom Lehrgebiet bereitgestellten Software-Umgebung über `File -> Download as -> PDF via LaTeX (.pdf)` oder dem entsprechenden Konsolenaufruf erzeugt worden sein.

In [1]:
```
# Imports
import pandas as pd
import numpy as np
import math
import re
import matplotlib.pyplot as plt
import seaborn as sns
import iplantuml
```

Diese Zelle bitte auf keinen Fall löschen, da der Seitenumbruch für die Korrektur benötigt wird.

\newpage

Aufgabe 1 - Datenvorverarbeitung und Datenqualität: Pixelpizza Pixhagen (55 Punkte)

Ein fiktives Szenario: Pizza in Pixelform - Das gibt es nur bei der Pizzeria Pixel in Pixhagen! Die Idee kam Pippa De Luigi, der Gründerin der Pizzeria, am 5. Mai 1999, als die gelernte Programmiererin in ihrem alten Job mal wieder mit ihren Kolleg*innen ein Coding Camp veranstaltete. Während sie gerade die Grafik eines Computerspiels entwickelte, entbrannte eine Diskussion, wie viele und welche Pizzen bestellt werden sollten. Pippa, selbst große Pizzaliebhaberin, überlegte sich, dass es doch praktisch wäre, wenn man in Pixeln angeben könnte, wie groß die Pizza sein sollte und pro Pixel einen unterschiedlichen Belag wählen könnte. Die Idee für Pippas Pixeria war geboren! Ein halbes Jahr später gründete sie das Unternehmen, das inzwischen neben ihr 91 Angestellte hat. Diese arbeiten in den Abteilungen:

- Marketing (um die Pizzen auch außerhalb von Pixhagen zu verkaufen)
- Entwicklung (ein Team, das ständig an neuen Rezeptideen arbeitet)
- Küche (die Pizzabackenden)
- Verwaltung (die sich um Buchhaltung etc. kümmern)
- IT (für interne Systeme, Bestellwebseite etc.)

Nach einem sehr erfolgreichen Jahr 2023, hat Pippa Budget übrig, dass sie für eine Weiterbildung verwenden möchte. Nachdem sie verschiedene Angebote für die unterschiedliche Abteilungen eingeholt hatte, stellte sie fest, dass sie zum aktuellen Zeitpunkt nur einer Abteilung eine Weiterbildung finanzieren kann. Um rauszufinden, für welche Abteilung das Geld am besten eingesetzt ist, setzte sich Pippa selbst wieder ans Coden und erstellte eine Webseite, in der die Angestellten verschiedene Fragen beantworten sollten.

Die Daten dieser Befragung sollen nun in der Klausur dazu verwendet werden, herauszufinden, welcher Abteilung die Fortbildung finanziert werden soll und über wie viele

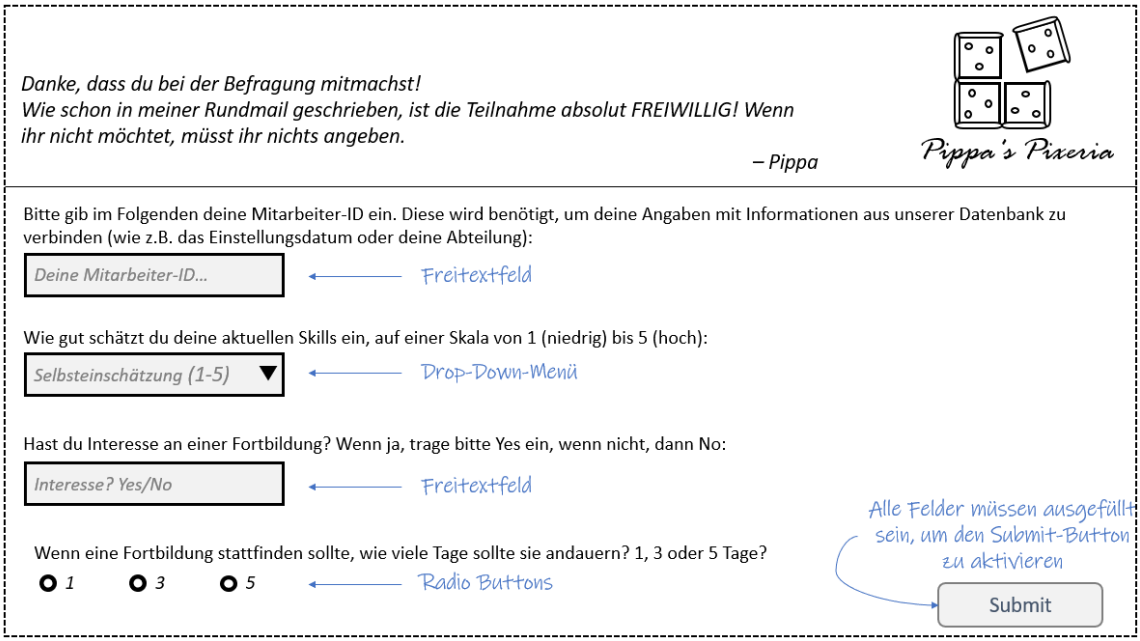
Tage. Die durchzuführende Analyse ist Inhalt dieser Klausur. Zeitpunkt der Analyse ist der heutige Tag.

Folgende Dokumente stehen Ihnen zur Verfügung:

- pixeria.csv - Die Daten der Befragung
- pixeria.pdf - Ein Auszug aus Pippas Notizbuch, in dem sie unterschiedliche Informationen zur Pizzeria und zur Befragung dokumentiert hat

Datenerhebung (6 Punkte)

Betrachten Sie den folgenden Screenshot der Webseite, die für die Datenerhebung genutzt wurde:



In Blau sind die einzelnen Felder genauer beschrieben (dies ist jedoch nicht beim Aufruf der Webseite zu sehen).

Data Integrity (4 Punkte)

*Nennen Sie einen Aspekt, der in Bezug auf Data Integrity** gut gelöst wurde und einen Punkt, den Sie verbessern würden, um eine bessere **Data Integrity** sicherzustellen. Begründen Sie Ihre Entscheidung und geben Sie an, auf welches Qualitätskriterium (die Sie im Buch von McGregor kennengelernt haben) sich der jeweilige Aspekt bezieht.*

Ihre Antwort: Durch die Submit button, die nur eine Submittion erlaubt, falls alle Felder ausgefüllt worden sind, wurde die Metrik **Complete** sichergestellt. Denn **Vollständige** Daten enthalten alle **notwendigen** Teile. Dies ist wichtig für eine umfassende Analyse, stellt jedoch nicht sicher, dass die Daten korrekt sind.

Data Fitness (2 Punkte)

Beurteilen Sie die Datenerhebung nun in Bezug auf Data Fitness: Nennen Sie hierfür einen Aspekt der Data Fitness (von denen, die Sie im Buch von McGregor kennengelernt haben), der

durch diese Art der Datenerhebung beeinträchtigt werden könnte? (2 Punkte)

Ihre Antwort: 1. **Gültigkeit (Validity)**: Die Gültigkeit bezieht sich darauf, ob die Daten tatsächlich das messen, was sie zu messen beabsichtigen. In diesem Fall scheinen die Daten gültig zu sein, da sie Informationen über die Einschätzung der Fähigkeiten der Mitarbeiter uind deren Interesse an einer Weiterbildung tatsächlich ein genaues Bild der Bedürfnisse und Fähigkeiten der Mitarbeiter liefern, was direkt relevant für die Fragestellung ist. Die Daten scheinen auch korrekt erfasst und codiert worden zu sein, wie aus den Anmerkungen hervorgeht. 2. Representativeness (Repräsentativität): die Daten scheinenn repräsentativ zu sein da alle Mitarbeiter an der Umfrage teilgenommen hatten. Da wir keine Information darüber haben, wie groß die Abteilungen sind, können wir nicht einschätzen ob einige Abteilungen über- oder unterrepräsentiert sind. Ein Aspekt der Data Fitness, der durch diese Art der Datenerhebung beeinträchtigt werden könnte, ist die **Zuverlässigkeit** der Daten. In der Befragung wurden Daten zu verschiedenen Aspekten wie der **Selbsteinschätzung** der Fähigkeiten, dem Interesse an einer Weiterbildung, der Dienstzeit und der bevorzugten Dauer der Weiterbildung erhoben. Allerdings basiert die Selbsteinschätzung der Fähigkeiten auf der subjektiven Einschätzung der Mitarbeiter, die möglicherweise nicht genau ist. Es könnte hilfreich sein, objektivere Maßnahmen für die Fähigkeiten der Mitarbeiter zu haben, um eine fundiertere Entscheidung treffen zu können.

Daten laden und Qualität einschätzen (13 Punkte)

Daten laden (1 Punkt)

Laden Sie die Daten (`pixeria.csv`) in ein DataFrame namens `df`, so dass der Index rein numerisch ist.

In [3]:
```
# Ihre Lösung
df = pd.read_csv('./pixeria.csv')
df
```

Out[3]:

	id	einschaetzung	interesse	tage	dienstzeit	einstellung	abteilung
0	KU3472	2	Y	3	0.0	2016-05-09	Kueche
1	IT4503	4	N	1	2.0	2000-11-19	IT
2	KU2731	4	Y	5	1.0	2004-07-11	Kueche
3	KU2470	3	Y	3	0.0	2015-12-05	Kueche
4	VT3574	3	N	1	0.0	2018-03-06	Verwaltung
...	...	...	...	...	...	...	...
88	VT4397	1	No	1	1.0	2004-10-29	Verwaltung
89	EW4654	2	No	5	1.0	2005-08-18	Entwicklung
90	KU4560	4	Yes	3	NaN	2012-08-15	Kueche
91	KU2138	3	Y	3	0.0	2015-05-07	Kueche
92	EW3661	1	Y	1	0.0	2016-11-07	Entwicklung

93 rows × 7 columns

Data Integrity (10 Punkte)

Machen Sie sich mit den Daten vertraut.

*Beurteilen Sie die Daten anhand der 5 Kriterien für Data Integrity, die Sie im Buch von McGregor in Kapitel 3 als Important** kennengelernt haben.*

In [6]:
```
#timely
print(df['einstellung'].min(), " - ", df['einstellung'].max())
```

```
2000-05-23  -  2022-03-29
```

In [7]:
```
print(df['dienstzeit'].min(), " - ", df['dienstzeit'].max())
```

```
0.0  -  2.0
```

In [8]:
```
#complete
df.isnull().sum()
```

Out[8]:
```
id              0
einschaetzung   0
interesse       0
tage            0
dienstzeit      7
einstellung     0
abteilung       0
dtype: int64
```

In [11]:
```
df[df['dienstzeit'].isnull()]
```

Out[11]:

	id	einschaetzung	interesse	tage	dienstzeit	einstellung	abteilung
35	IT2481	2	N	5	NaN	2009-09-24	IT
49	KU1511	3	Y	5	NaN	2021-09-13	Kueche
56	EW1108	2	N	3	NaN	2015-09-04	Entwicklung
67	EW3233	2	Y	1	NaN	2021-09-15	Entwicklung
75	EW1895	3	Y	5	NaN	2005-01-18	Entwicklung
78	MK1914	5	Y	3	NaN	2004-06-29	Marketing
90	KU4560	4	Yes	3	NaN	2012-08-15	Kueche

In [28]:
```
df.describe(include='all')
```

Out[28]:

	id	einschaetzung	interesse	tage	dienstzeit	einstellung	abteilung
count	93	93.000000	93	93.000000	86.000000	93	93
unique	91	NaN	2	NaN	NaN	91	5
top	KU2138	NaN	Yes	NaN	NaN	2015-05-07	Kueche
freq	2	NaN	63	NaN	NaN	2	34
mean	NaN	3.096774	NaN	2.720430	0.790698	NaN	NaN
std	NaN	1.180130	NaN	1.401595	0.721376	NaN	NaN
min	NaN	1.000000	NaN	1.000000	0.000000	NaN	NaN
25%	NaN	2.000000	NaN	1.000000	0.000000	NaN	NaN
50%	NaN	3.000000	NaN	3.000000	1.000000	NaN	NaN
75%	NaN	4.000000	NaN	3.000000	1.000000	NaN	NaN
max	NaN	5.000000	NaN	5.000000	2.000000	NaN	NaN

In [4]:
```
# Platz für Analysen
len(df)
```

Out[4]:
```
93
```

In [29]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 93 entries, 0 to 92
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             93 non-null     object 
 1   einschaetzung  93 non-null     int64  
 2   interesse      93 non-null     object 
 3   tage           93 non-null     int64  
 4   dienstzeit     86 non-null     float64
 5   einstellung    93 non-null     object 
 6   abteilung      93 non-null     object 
dtypes: float64(1), int64(2), object(4)
memory usage: 5.2+ KB
```

In [12]:
```
df['abteilung'].value_counts()
#df.groupby('stadtteil')['entfernung'].mean()
```

Out[12]:
```
abteilung
Kueche         34
Entwicklung    25
IT             14
Marketing      11
Verwaltung      9
Name: count, dtype: int64
```

In [13]:
```
#multivariate
# Check the number of variables (columns) in the dataset
if df.shape[1] > 1:
    print("Der Datensatz ist multivariant mit ", df.shape[1], " Variablen.")
else:
    print("Der Datensatz ist nicht multivariant.")
```

```
Der Datensatz ist multivariant mit  7  Variablen.
```

In [14]:
```
#atomic
def is_atomic(df):
    for column in df.columns:
        if df[column].apply(lambda x: isinstance(x, (list, dict))).any():
            return False
    return True

#überprüft,ob ein Element in der Spalte eine Instanz einer Liste oder eines diction
print(is_atomic(df))
```

```
True
```

In [5]:
```
df['abteilung'].value_counts()
```

Out[5]:
```
abteilung
Kueche         34
Entwicklung    25
IT             14
Marketing      11
Verwaltung      9
Name: count, dtype: int64
```

Ihre Beurteilung: 1. **Timely**: Die Daten scheinen zeitgemäß zu sein, da die Einstellungsdatum der Mitarbeiter von 2000-05-23 bis 2022-03-29 reicht und die 'dienstzeit' diese Daten nicht überschreitet. In den Anmerkungen stand dass, die Webseite zur Datenabfrage vom 01.01.2024 bis 16.02.2024 verfügbar sein wird. Dies bedeutet, dass die Daten aktuell und zeitnah sind. 2. **Complete**: Es gibt einige fehlende Werte in der 'dienstzeit' Spalte, was darauf hinweist, dass die Daten nicht vollständig sind. Es könnte notwendig sein, diese fehlenden Werte zu behandeln, um genaue Analysen durchzuführen. Dies kann man mit der Funktion zur Berechnung des Attributs Dienstzeit aus Einstellungsdatum, die in der Anmerkungen zur Verf+gumg gestellt wurde. 3. **Multivariate**: Der Datensatz ist multivariant mit 7 Variablen bzw. 6 zur Beantwirtung der Fragestellung, was bedeutet, dass er mehrere Variablen für die Analyse enthält. 4. **Atomic**: Die Daten scheinen atomar zu sein, da keine Spalte Elemente enthält, die Instanzen einer Liste oder eines Wörterbuchs sind. Die Daten weisen auch keine zusammenfassenden Statistiken, kombinierte Daten, Aggregationen oder Kennzahlen auf. 5. **High Volume**: Die Einräge reichen aus, um die Hauptfrage zu beantworten. Insgesamt wurden 93 Personen befragt, verteilt auf die 5 abteilungen. diese Verteilung scheint nicht gleichmäßig zu sein, da die Abteilung 'Kueche' mehr Befragte hat als die anderen abteilungen, während die Abteilung 'Verwaltung' vielweniger befragten hat. Wenn die Größe edr Abteilungen jedoch stark variiert, könnte diese Verteilung repräsentativ sein. Wir haben leider keine Infos darüber wie groß die abteilungen sind. Insgesamt scheint der Datensatz eine gute Datenfitness zu haben, obwohl es einige potenzielle Probleme gibt, die berücksichtigt werden sollten.

Data Fitness: Representativeness (2 Punkte)

Machen Sie sich mit den Daten vertraut.

Beurteilen Sie die Daten in Bezug auf Representativeness.

Ihre Antwort: Folgende Beurteilungen habe ich über die Representativität dea Datensatzes: - Abdeckung aller Abteilungen: Die Daten enthalten Einträge aus allen Abteilungen (Küche, IT, Entwicklung, Marketing, Verwaltung), was darauf hindeutet, dass sie eine gute Repräsentation der gesamten Organisation darstellen könnten. - Ausgewogenheit der Abteilungen: Wie bereits erwähnt, variiert die Anzahl der Einträge aus jeder Abteilung stark, was die Repräsentativität der Daten beeinträchtigen könnte. Wenn eine Abteilung überrepräsentiert ist, könnten die Ergebnisse verzerrt sein. Wenn die Größe edr Abteilungen jedoch stark variiert, könnte

diese Verteilung repräsentativ sein. Wir haben leider keine Infos darüber wie groß die abteilungen sind. - Zeitliche Relevanz: Die ‘einstellung’ Daten reichen von 2000 bis 2022, was darauf hindeutet, dass die Daten eine breite Zeitspanne abdecken. Es wäre jedoch wichtig zu prüfen, ob die Daten gleichmäßig über diesen Zeitraum verteilt sind. - Vollständigkeit der Daten: Es gibt einige fehlende Werte in der ‘dienstzeit’ Spalte. Dies könnte die Repräsentativität der Daten beeinträchtigen, da nicht alle Mitarbeiter vollständige Daten haben. Insgesamt scheint der Datensatz eine gute Representativität zu haben.

Datenvorverarbeitung (16 Punkte)

Diese Aufgabe beschäftigt sich mit der Datenvorverarbeitung.

Plausibilitätschecks (4 Punkte)

Entsprechend der Informationen aus dem Bericht sind verschiedene Plausibilitätschecks der Daten denkbar. Bei diesen Checks wird geprüft, ob sich die vorliegenden Informationen in den Daten wiederspiegeln.

Führen Sie zwei Plausibilitätschecks Ihrer Wahl durch. Geben Sie bei jedem Check an, auf welche Information Sie sich beziehen und, ob der Check fehlgeschlagen ist oder nicht.

In [ ]:
```
# Ihre Lösung
#Bereichsprüfung: liegen die Werte in den Spalten innerhalb eines erwarteten Bereic

df['alter'].min()>=9 & df['alter'].max()<=19
```

In [25]:
```
#Vollständigkeitsprüfung: sind alle erforderlichen Daten vorhanden?

missing_values = df.isnull().sum()
print("Fehlende Werte in jeder Spalte:\n", missing_values)
print(f'sind alle erforderlichen Daten vorhanden?', missing_values.any()==False)
```

```
Fehlende Werte in jeder Spalte:
 id              0
einschaetzung    0
interesse        0
tage             0
dienstzeit       7
einstellung      0
abteilung        0
dtype: int64
sind alle erforderlichen Daten vorhanden? False
```

In [19]:
```
df.groupby('abteilung')['id'].first()
```

Out[19]:
```
abteilung
Entwicklung    EW2718
IT             IT4503
Kueche         KU3472
Marketing      MK3771
Verwaltung     VT3574
Name: id, dtype: object
```

In [20]:
```
#Konsistenzprüfung: Überprüfen, ob die Daten in verschiedenen Spalten konsistent si

id_dept_mapping = {
```

```
"KU":"Kueche",
"IT":"IT",
"EW":"Entwicklung",
"MK":"Marketing",
"VT":"Verwaltung"
}

import re

# Ihre Lösung
regular_expression = "[A-Z]+"
def check_id(id_, abteilung):
    dept_id = "".join(re.findall(regular_expression, id_))
    return id_dept_mapping.get(dept_id) == abteilung

# Ist der Check für alle Zeilen des Datensatzes erfolgreich?
df.apply(lambda x: check_id(x["id"], x["abteilung"]), axis=1).all()
```

Out[20]: True

In [21]:
```
#Gültigkeitsprüfung: Überprüfen Sie, ob die Werte in den Spalten gültig sind.

invalid_values_einschaetzung = ~df['einschaetzung'].isin([1,2,3,4,5])
print("Anzahl der ungültigen Werte in der Spalte 'einschaetzung':", invalid_values_

invalid_values_dienstzeit = ~df['dienstzeit'].isin([0,1,2])
print("Anzahl der ungültigen Werte in der Spalte 'dienstzeit':", invalid_values_die
```

```
Anzahl der ungültigen Werte in der Spalte 'einschaetzung': 0
Anzahl der ungültigen Werte in der Spalte 'dienstzeit': 7
```

Praktische Anwendung (12 Punkte)

Für eine sinnvolle Analyse sind diverse Vorverarbeitungsschritte notwendig.

Führen Sie Vorverarbeitungsmaßnahmen durch, die Ihrer Meinung nach nötig sind, um die Qualität der Daten zu verbessern und die nachfolgenden Analysen durchzuführen. Begründen Sie für jede Maßnahme Ihr Vorgehen.

> **Hinweise**:
>
> - *Eine Umbenennung** von Spalten ist nicht notwendig!*
> - *Es sollen keine neuen Spalten hinzugefügt** werden!*
> - *Nehmen Sie an, dass in den nachfolgenden Analysen sowie in denen, die Pippa zusätzlich vornimmt, alle Spalten benötigt werden. Aus der Spalte* `einstellung` *wird außerdem das Jahr benötigt.*

In [26]:
```
# Ihre Lösung
jein_recoding = {
    'Y': 'Yes',
    'N': 'No'
}
```

```
df['interesse'] = df['interesse'].map(jein_recoding).fillna(df['interesse'])
```

In [30]:
```
df['interesse'].value_counts()
```

Out[30]:
```
interesse
Yes    63
No     30
Name: count, dtype: int64
```

In [33]:
```
#einstellungs datum zur Datentzyp datetime tranformieren
df['einstellung'] = pd.to_datetime(df['einstellung'], format='%Y-%m-%d')
df['einstellung'].dtype
```

Out[33]:
```
dtype('<M8[ns]')
```

In [34]:
```
#identische Duplikate löschen
duplicates = df.duplicated(subset=['id'], keep=False)
print("Anzahl der Duplikate in der Spalte 'ID':", duplicates.sum())
df[duplicates]
```

```
Anzahl der Duplikate in der Spalte 'ID': 4
```

Out[34]:

	id	einschaetzung	interesse	tage	dienstzeit	einstellung	abteilung
26	KU2138	3	Yes	3	0.0	2015-05-07	Kueche
72	EW3661	1	Yes	1	0.0	2016-11-07	Entwicklung
91	KU2138	3	Yes	3	0.0	2015-05-07	Kueche
92	EW3661	1	Yes	1	0.0	2016-11-07	Entwicklung

In [35]:
```
#to remove all, keep=False
df.drop_duplicates(subset=['id'], keep='first', inplace=True)
df.duplicated(subset=['id'], keep=False).sum()
```

Out[35]:
```
0
```

In [44]:
```
#fehlende werte in spalte dienszeit mit Mean von Spalte einstellung ersetzen
import math

df['einstellung'] = pd.to_datetime(df['einstellung'])

df['dienstzeit'] = ((pd.to_datetime("2024-02-26", format="%Y-%m-%d") - df['einstell

df['dienstzeit'] = df['dienstzeit'].apply(lambda x: 0 if x < 10 else (1 if x < 20 e
```

In [45]:
```
df['dienstzeit'].isnull().sum()
```

Out[45]:
```
0
```

Ihre Begründung: 1. **Recoding von 'Y' und 'N' zu 'Yes' und 'No' in der Spalte 'interesse'**: Diese Ändeerung stellt eine konsistente Codierung sicher und erleichtert die Interpretation der Daten. Es ist wichtig, dass alle Daten in einem einheitlichen Format vorliegen . 2. **Umformung des 'einstellung' Datums in den datetime Datentyp**: Dies ermöglicht eine korrekte Berechnung und Analyse von Datumsangaben. 3. **Entfernung von Duplikaten in der Spalte 'id'**: Duplikate können die Analyse

verzerren und zu falschen Schlussfolgerungen führen. Durch das Entfernen von Duplikaten stellen wir sicher, dass jede Zeile im Datensatz eindeutige Informationen enthält. Außerdem wird in dem beigefügten PDF erwähnt dass alle IDs eindeutig sein sollten.
4. **Ersetzen fehlender Werte in der Spalte 'dienstzeit' mit dem Durchschnitt der Spalte 'einstellung'**: Das Ersetzen fehlender Werte (imputation) ist wichtig zur Behandlung von fehlenden Daten. In diesem Fall verwenden ich die Formel in dem beigefügten Dokument, um die fehlenden Werte zu ersetzen. Dann habe ich die berechnete Dienstzeit auf die Zahlen 0,1 und 2 gemappt, wie im beuígefügten Dokument vorgeschrieben.

Datenanalyse (20 Punkte)

In dieser Aufgabe werden Analysen auf den Daten ausgeführt. Bei den zu erstellenden Diagrammen ist der Diagrammtyp immer von Pippa vorgegeben und soll deshalb auch so umgesetzt werden. Eine anschließende Beurteilung, ob der gewählte Typ für die Fragestellung geeignet ist, ist dabei Teil der Aufgabe.

Interesse nach Dienstzeit (4 Punkte)

Pippa interessiert sich zunächst dafür, ob es im Interesse einen Unterschied gibt, je nachdem seit wie vielen Jahren die Angestellten für sie arbeiten.

Geben Sie zunächst die absoluten Zahlen aus, wie viele der Angestellten, die seit 10 oder mehr Jahren dabei sind, Interesse/kein Interesse an einer Fortbildung haben. Geben Sie dies ebenfalls für die Angestellten aus, die seit weniger als 10 Jahren dabei sind. Berechnen Sie anschließend aus diesen Zahlen, wie viel Prozent der jeweils beiden Gruppen Interesse zeigen. Ist hier ein deutlicher Unterschied zu erkennen?

In [50]:
```
# Ihre Lösung
df['dienstzeit'].value_counts()
```

Out[50]:
```
dienstzeit
1    41
0    34
2    16
Name: count, dtype: int64
```

In [51]:
```
counts_10_or_more = df[df['dienstzeit'] >= 1]['interesse'].value_counts()
counts_less_than_10 = df[df['dienstzeit'] == 0]['interesse'].value_counts()

print("Absolute Zahlen (10 oder mehr Jahre):")
print(counts_10_or_more)
print("\nAbsolute Zahlen (weniger als 10 Jahre):")
print(counts_less_than_10)
```

```
Absolute Zahlen (10 oder mehr Jahre):
interesse
Yes    39
No     18
Name: count, dtype: int64

Absolute Zahlen (weniger als 10 Jahre):
interesse
Yes    22
No     12
Name: count, dtype: int64
```

In [52]:
```
percentages_10_or_more = df[df['dienstzeit'] >= 1]['interesse'].value_counts(normal
percentages_less_than_10 = df[df['dienstzeit'] == 0]['interesse'].value_counts(norm

print("\nProzentuale Verteilung (10 oder mehr Jahre):")
print(percentages_10_or_more)
print("\nProzentuale Verteilung (weniger als 10 Jahre):")
print(percentages_less_than_10)
```

```
Prozentuale Verteilung (10 oder mehr Jahre):
interesse
Yes    68.421053
No     31.578947
Name: proportion, dtype: float64

Prozentuale Verteilung (weniger als 10 Jahre):
interesse
Yes    64.705882
No     35.294118
Name: proportion, dtype: float64
```

Ihre Antwort: Nein kein großer unterschied ist zu erkennen.

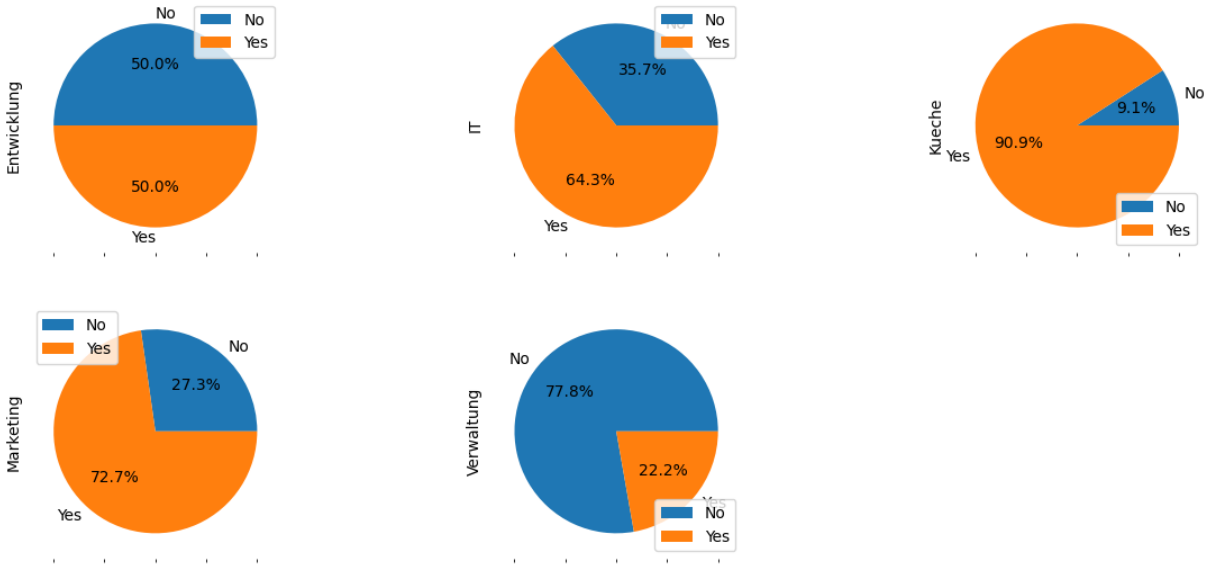
Interesse nach Abteilungen - Teil 1 (4 Punkte)

Um entscheiden zu können, welcher Abteilung die Fortbildung finanziert wird, möchte Pippa im nächsten Schritt das Interesse je nach Abteilung beurteilen können. Welche beiden Abteilungen haben am meisten Interesse?

- *Beantworten Sie die Frage mit einem Kreisdiagramm (eines pro Abteilung).*
- *Berücksichtigen Sie dabei die Kriterien, die Sie im Buch von McGregor auf den Seiten 327 ff. kennengelernt haben: Werden alle Regeln und Richtlinien erfüllt? Falls Nein: Welche sind verletzt?*
- *Welche beiden Abteilungen haben das größte Interesse?*

In [55]:
```
# Ihre Lösung

df.groupby(['abteilung', 'interesse'])\
.size()\
.unstack('abteilung')\
.plot(kind='pie',subplots=True, figsize=(15,10), layout=(3, 3), autopct='%1.1f%%')
```

```
plt.title('Interesse pro Abteilung')
plt.show()
```



Ihre Antwort und Einschätzung Regeln/Richtlinien: Die beiden abteilungen mit der grüßten interesse sind 'Kueche' und 'MArketing'.

Interesse nach Abteilungen - Teil 2 (4 Punkte)

Die Kreisdiagramme zeigen nur den Anteil und nicht die absoluten Zahlen. Diese sollen im nächsten Schritt genauer analysiert werden. Betrachten Sie deshalb im nächsten Schritt nur die beiden Abteilungen, in denen das Interesse an einer Fortbildung am höchsten ist. Sollten Sie in der vorherigen Aufgabe zu keiner Lösung gekommen sein, wählen Sie zwei Abteilungen Ihrer Wahl. In welcher Abteilung ist das Interesse größer?

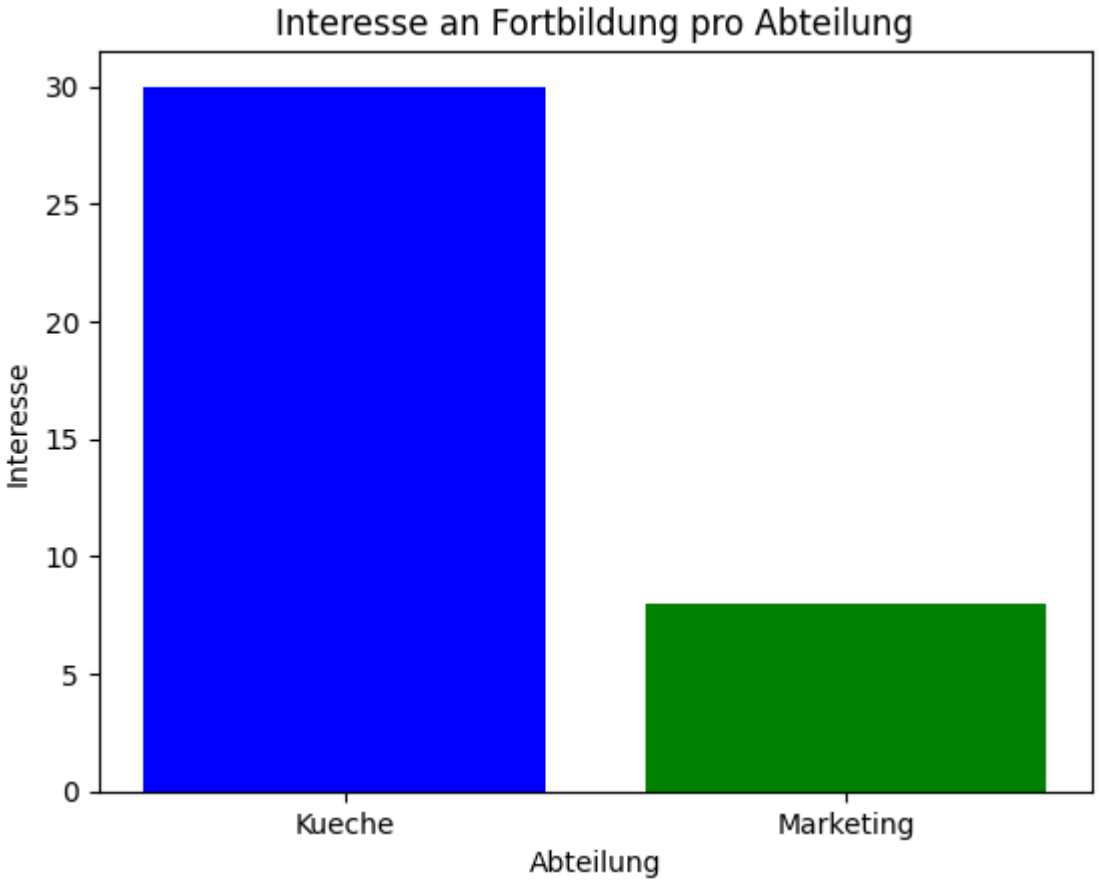
- *Beantworten Sie die Frage mit einem Säulendiagramm.*
- *Berücksichtigen Sie dabei die Kriterien, die Sie im Buch von McGregor auf den Seiten 327 ff. kennengelernt haben: Werden alle Regeln und Richtlinien erfüllt? Falls Nein: Welche sind verletzt?*
- *Für welche der beiden Abteilungen würden Sie Pippa die Fortbildung empfehlen und warum?*

In [71]:
```
df.groupby(['abteilung', 'interesse'])\
.size()[['Kueche','Marketing']]
```

Out[71]:
```
abteilung  interesse
Kueche     Yes          30
Marketing  Yes           8
dtype: int64
```

In [65]:
```
abteilungen = ['Kueche', 'Marketing']
interesse = [30, 8]

plt.bar(abteilungen, interesse, color=['blue', 'green'])

plt.title('Interesse an Fortbildung pro Abteilung')
plt.xlabel('Abteilung')
```

```
plt.ylabel('Interesse')

plt.show()
```



Ihre Antwort und Einschätzung Regeln/Richtlinien: Ich empfiehle Pippa die Fortbildung in der Abteilung Kueche vorzunehmen, da die Interesse dort am höchsten ist.

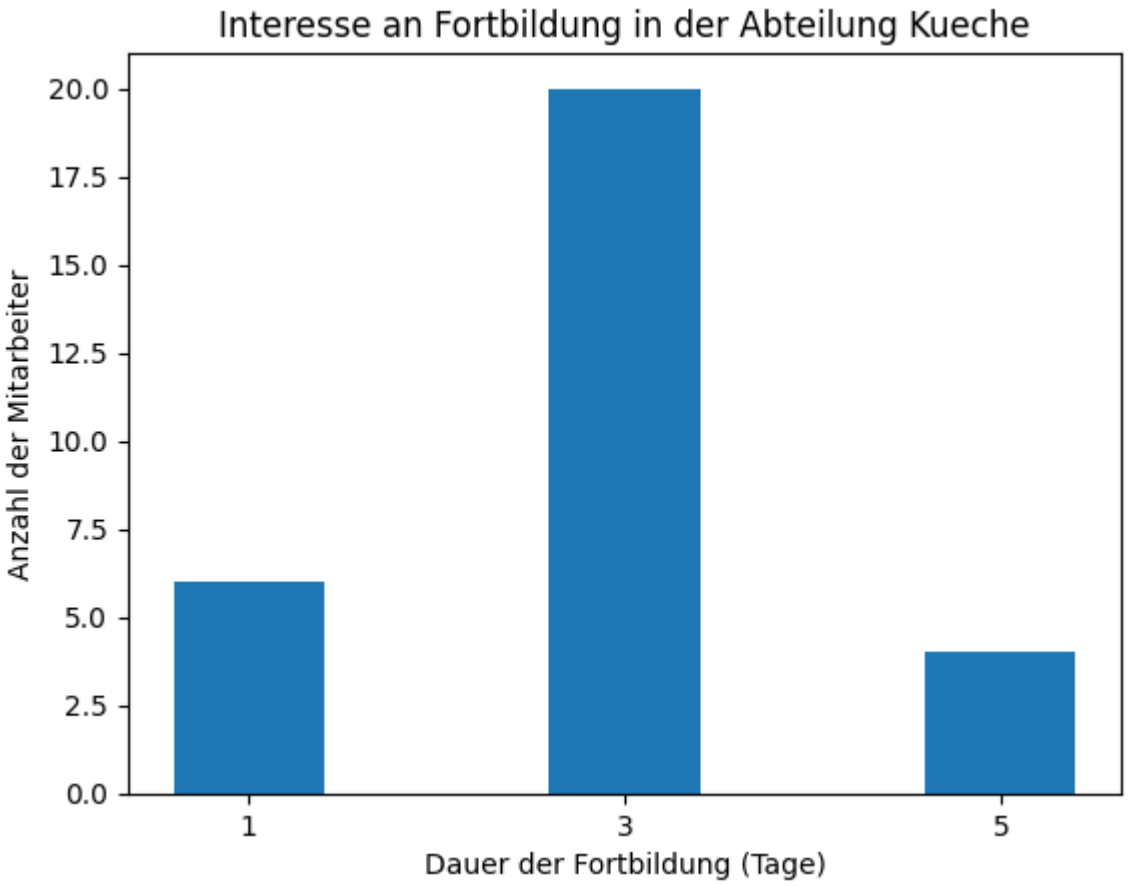
Anzahl Tage der Fortbildung (4 Punkte)

Für die weitere Planung möchte Pippa wissen, ob die Fortbildung einen, drei oder fünf Tage lang dauern soll. Wählen Sie die Abteilung, die Ihren Analysen zufolge das größte Interesse an einer Fortbildung hat. Sollten Sie in der vorherigen Aufgabe zu keinem Ergebnis gelangt sein, wählen Sie hier eine Abteilung Ihrer Wahl. Wie lange sollte die Fortbildung dauern - einen, drei oder fünf Tage?

- *Erstellen Sie für die gewählte Abteilung ein geeignetes Säulendiagramm, um die Fragestellung zu beantworten.*
- *Berücksichtigen Sie dabei die Kriterien, die Sie im Buch von McGregor auf den Seiten 327 ff. kennengelernt haben: Werden alle Regeln und Richtlinien erfüllt? Falls Nein: Welche sind verletzt?*
- *Welche Länge der Fortbildung raten Sie Pippa und für die gewählte Abteilung?*

In [68]:
```
# Ihre Lösung

# Nur die Zeilen auswählen, in denen 'interesse' Yes' ist
df_kueche = df[df['abteilung']=='Kueche']
```

```
df_kueche = df_kueche[df_kueche['interesse']=='Yes']

# Anzahl der Mitarbeiter zählen, die an einer ein-, drei- oder fünftägigen Fortbild
counts = df_kueche['tage'].value_counts().sort_index()

plt.bar(counts.index, counts.values)
plt.xlabel('Dauer der Fortbildung (Tage)')
plt.ylabel('Anzahl der Mitarbeiter')
plt.title('Interesse an Fortbildung in der Abteilung Kueche')
plt.xticks([1, 3, 5])
plt.show()
```



Ihre Antwort und Einschätzung Regeln/Richtlinien: Die Länge der Fortbildung soll 3 Tage beitragen.

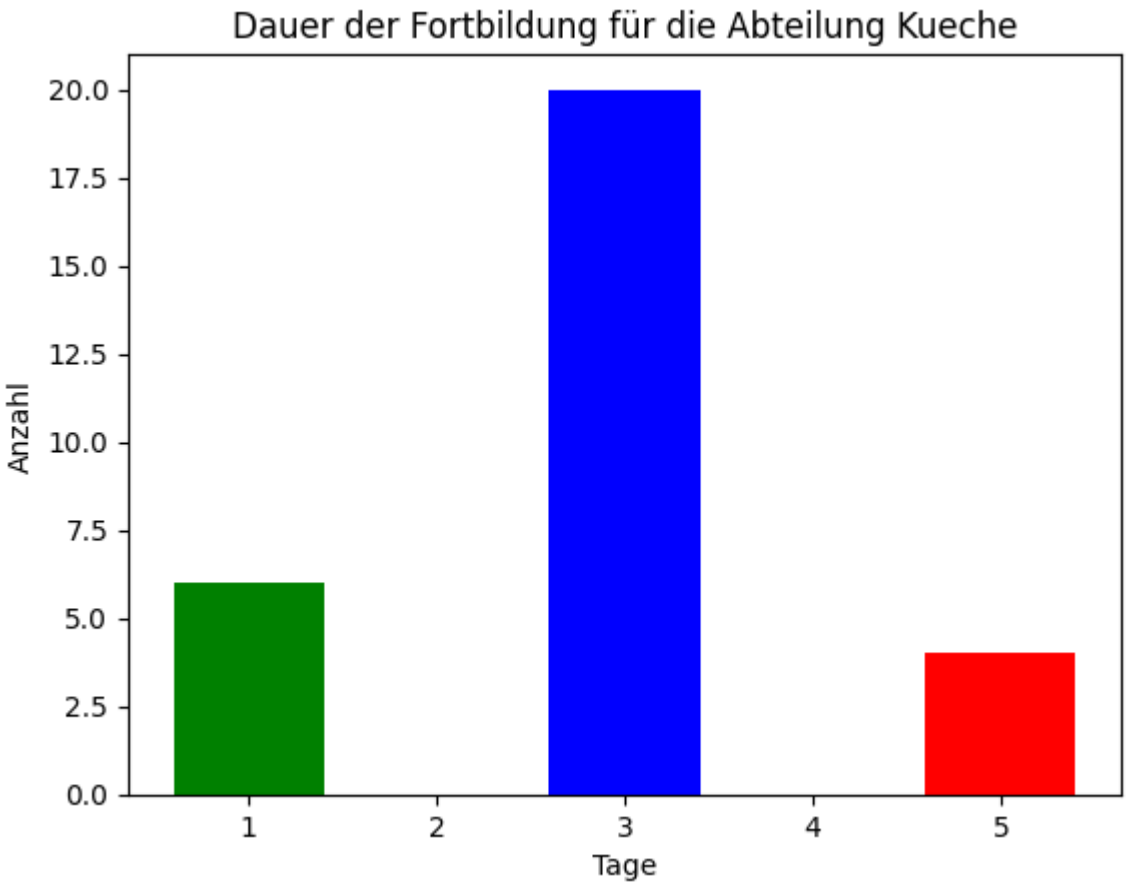
Selbsteinschätzung (4 Punkte)

Im letzten Schritt möchte Pippa wissen, ob die Fortbildung sich eher an Anfänger:innen oder an Fortgeschrittene richten soll. Wählen Sie wiederum die Abteilung, die Ihren Analysen zufolge das größte Interesse an einer Fortbildung hat. Sollten Sie in der entsprechenden Aufgabe zu keinem Ergebnis gelangt sein, wählen Sie hier eine Abteilung Ihrer Wahl.

- *Erstellen Sie für diese Abteilung ein Liniendiagramm, um die Fragestellung zu beantworten.*
- *Berücksichtigen Sie dabei die Kriterien, die Sie im Buch von McGregor auf den Seiten 327 ff. kennengelernt haben: Werden alle Regeln und Richtlinien erfüllt? Falls Nein: Welche sind verletzt?*

- *Welchen Schwierigkeitsgrad raten Sie Pippa füpr die gewählte Abteilung - Anfänger oder Fortgeschritten?*

In [69]:
```
# Ihre Lösung


#Anzahl der Tage zählen
tage_count = df_kueche['tage'].value_counts()

plt.bar(tage_count.index, tage_count.values, color=['blue', 'green', 'red'])

plt.title('Dauer der Fortbildung für die Abteilung Kueche')
plt.xlabel('Tage')
plt.ylabel('Anzahl')
plt.show()
```



Ihre Antwort und Einschätzung Regeln/Richtlinien:

Diese Zelle bitte auf keinen Fall löschen, da der Seitenumbruch für die Korrektur benötigt wird.

\newpage

Aufgabe 2 - Big Data Analysis (15 Punkte)

In diesem Aufgabenbereich geht es um Hadoop und Spark.

Eigenschaften von Spark (3 Punkte)

Iryna aus der IT-Abteilung möchte mit Spark das Kundenverhalten analysieren. Da sie sich noch nicht so gut mit Spark auskennt, macht sie zunächst ein paar Tests und hat dabei folgenden Spark-Code geschrieben:

```
In [6]: # RDD erstellen
        rdd = sc.parallelize(lists, 3)

In [7]: # Verteilung auf einzelne Partitionen anzeigen
        rdd.glom().collect()

Out[7]: [[2, 2, 4, 1, 2, 3, 4, 1, 3, 4, 4, 3, 1, 3, 2, 4, 2, 4, 3, 1, 1, 2, 4],
         [4, 2, 2, 3, 4, 1, 4, 4, 4, 3, 4, 2, 1, 3, 4, 3, 1, 3, 2, 2, 4, 3, 3],
         [3, 3, 1, 4, 3, 4, 1, 4, 1, 3, 2, 1, 4, 4, 2, 1, 2, 1, 1, 2, 1, 3, 2, 2, 1]]

In [8]: # Werte >=4 entfernen
        rdd_filtered = rdd.filter(lambda x: x < 4)

In [9]: # Verteilung auf einzelne Partitionen anzeigen
        rdd_filtered.glom().collect()

Out[9]: [[2, 2, 1, 2, 3, 1, 3, 3, 1, 3, 2, 2, 3, 1, 1, 2],
         [2, 2, 3, 1, 3, 2, 1, 3, 3, 1, 3, 2, 2, 3, 3],
         [3, 3, 1, 3, 1, 1, 3, 2, 1, 2, 1, 2, 1, 1, 2, 1, 3, 2, 2, 1]]

In [ ]: # Repartition?
```

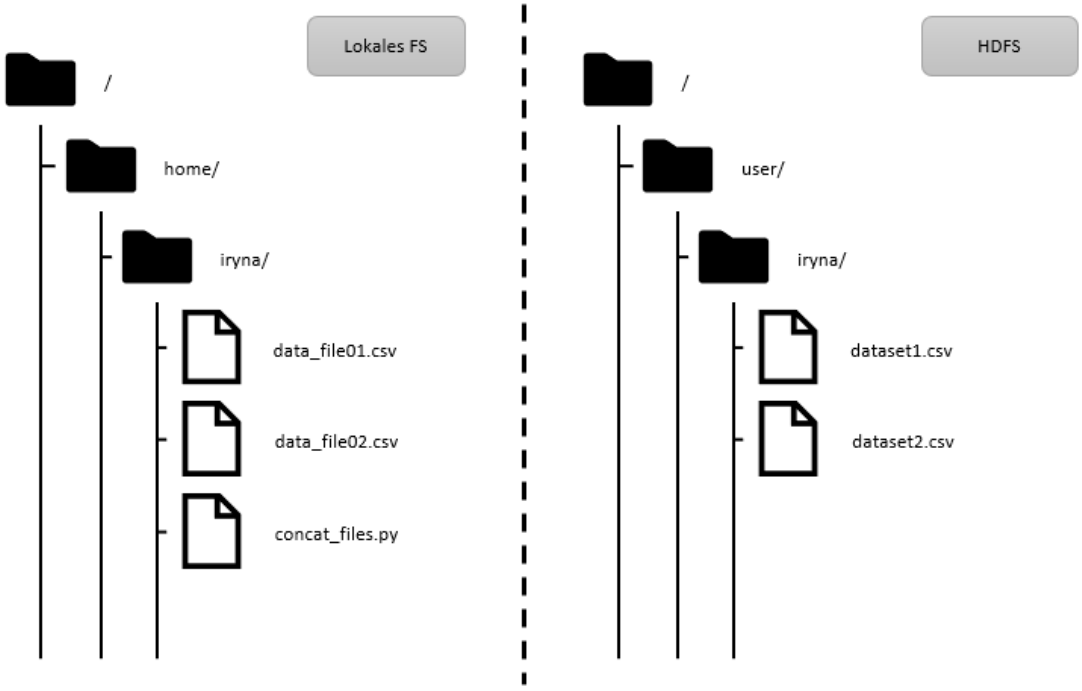
Nun überlegt sie, ob sie an dieser Stelle ein Repartitionieren ausführen soll oder nicht. Was würden Sie ihr raten:

Ist an dieser Stelle ein Repartitionieren sinnvoll oder nicht? Begründen Sie Ihre Entscheidung! (2 Punkte)

Ihre Antwort: ich würde Iryna raten, zuerst die Datenverteilung nach der Filteroperation zu überprüfen. Hier sieht man dass die Filteroperation dazu geführt hat, dass einige Partitionen deutlich weniger Daten enthalten als andere (was zu einer ungleichmäßigen Lastverteilung führen kann), also könnte ein Repartitionieren sinnvoll sein. Außerdem sind die Datenmenge nach der Filteroperation deutlich kleiner, ein Repartitionieren könnte dazu beitragen, die Anzahl der Partitionen zu reduzieren und so die Verarbeitungseffizienz zu verbessern.

Dateien einlesen (4 Punkte)

Betrachten Sie folgende Übersicht aus Irynas lokalem FS und dem HDFS:



Iryna hat wie folgt eine SparkSession und den SparkContext erstellt:

```
spark = SparkSession.builder \
    .master("yarn") \
    .appName("pippas-pixeria") \
    .getOrCreate()
sc = spark.sparkContext
```

Betrachten Sie nun den untenstehenden Spark-Code (`>>` zeigt die Ausgabe des Codes):

```
data_path = "/user/iryna/data_file01.csv"

# Data Loading 1
df = spark.read.csv(path=data_path)
>> AnalysisException: Path does not exist: hdfs://hadoop-
edge:9000/user/iryna/data_file01.csv
# Data Loading 2
rdd = sc.textFile(name=data_path)
>>
```

Warum kommt es beim Data Loading 1 zu einem Fehler, aber beim Data Loading 2 nicht? (4 Punkte)

Ihre Antwort: Der Unterschied im Verhalten zwischen den beiden Ladevorgängen liegt in der Art und Weise, wie Spark die Dateipfade behandelt. Beim Laden von Daten mit spark.read.csv, versucht Spark sofort, auf den angegebenen Pfad zuzugreifen und die Daten zu lesen. Wenn der Pfad nicht existiert oder nicht zugänglich ist (wie in diesem Fall), wird eine AnalysisException ausgelöst. Im Gegensatz dazu, wenn man sc.textFile(name=data_path) verwendet, wird die Datei nicht sofort gelesen. Stattdessen wird ein RDD erstellt, das auf den angegebenen Pfad verweist. Die Daten werden erst gelesen, wenn eine Aktion (wie count, collect usw.) auf dem RDD ausgeführt wird. Daher wird in diesem Fall kein Fehler ausgelöst, auch wenn der Pfad nicht existiert oder nicht zugänglich ist.

Ressourcenvebrauch (4 + 4 Punkte)

Der Pixeria steht ein Cluster mit 120GB RAM und 75 Kernen zur Verfügung.

Im Folgenden ist der Inhalt der Datei `spark-defaults.conf` zu sehen:

```
spark.executor.instances 6
spark.executor.memory 16GB
spark.executor.cores 8
spark.executor.memoryOverhead 2GB

spark.yarn.am.cores 4
spark.yarn.am.memory 8GB
spark.yarn.am.memoryOverhead 1GB

spark.dynamicAllocation.enabled false
```

Iryna startet eine pyspark-Shell mit folgendem Aufruf (zu diesem Zeitpunkt laufen keine weiteren Spark-Anwendungen):

```
pyspark --master yarn
```

Berechnen Sie den Ressourcen-Verbrauch (Memory und Cores) der Spark-Anwendung. Geben Sie dabei Ihren Rechenweg an. (4 Punkte)

Ihre Antwort: **Gesamter Ressourcenverbrauch:** - Gesamtspeicher: Executor-Speicher + AM-Speicher = 96Gb + 8gb = 104Gb - Gesamtkerne: Executor-Kerne + AM-Kerne = 48 Kerne + 4 Kerne = 52 Kerne - Gesamter Speicher-Overhead: 12gb (nur für Executor)

Während die eben gestartete pyspark-Shell noch läuft, startet Iryna eine zweite pyspark-Shell mit folgendem Aufruf:

```
pyspark --master yarn --num-executors 4 --executor-memory 8G --executor-cores 8
```

Wie viele Ressourcen werden benötigt? Kann die Shell gestartet werden? Kann Spark-Code ausgeführt werden? Begründen Sie Ihre Antworten! (4 Punkte)

Ihre Antwort: - Der Cluster hat 120GB RAM und 75 Kerne. - Die erste pyspark-Shell wurde bereits gestartet und beansprucht 104GB Speicher und 52 Kerne. -Ressourcenverbrauch der zweiten pyspark-shell: Gesamtspeicher: 4 * 8GB = 32GB Gesamtkerne: 4 * 8 = 32 Kerne -Verfügbare Ressourcen nach der ersten shell: - Verbleibender Speicher: 120GB - 104GB = 16GB - Verbleibende Kerne: 75 Kerne - 52 Kerne = 23 Kerne -Kann die shell gestartet werden? - Die neue Anwendung würde 32GB Speicher und 32 Kerne beanspruchen. - Es gibt genügend verfügbaren Speicherplatz (16GB) für die neue Anwendung. - Es gibt jedoch nicht genügend verfügbare Kerne (23 Kerne), um die neue Anwendung zu unterstützen. Die neue Anwendung beansprucht 32 Kerne, was mehr ist als die verfügbaren 23 Kerne. Da nicht genügend Kerne verfügbar sind, um die zweite pyspark-Shell zu unterstützen, kann sie nicht gestartet werden. Selbst wenn genügend Speicherplatz verfügbar wäre, würde die unzureichende Anzahl verfügbarer Kerne dazu führen, dass die Shell nicht gestartet werden kann.

Diese Zelle bitte auf keinen Fall löschen, da der Seitenumbruch für die Korrektur benötigt wird.

\newpage

Aufgabe 3 - Modellierung, NoSQL (25 Punkte)

Die IT-Abteilung von Pippas Pixeria hat den Vorschlag gemacht zu einem NoSQL-Datenbanksystem zu migrieren. Dafür wurden zunächst MongoDB und Cassandra in Betracht gezogen. Nachfolgende Abfragen wurden für die Entwicklung erster Schemaentwürfe herangezogen.

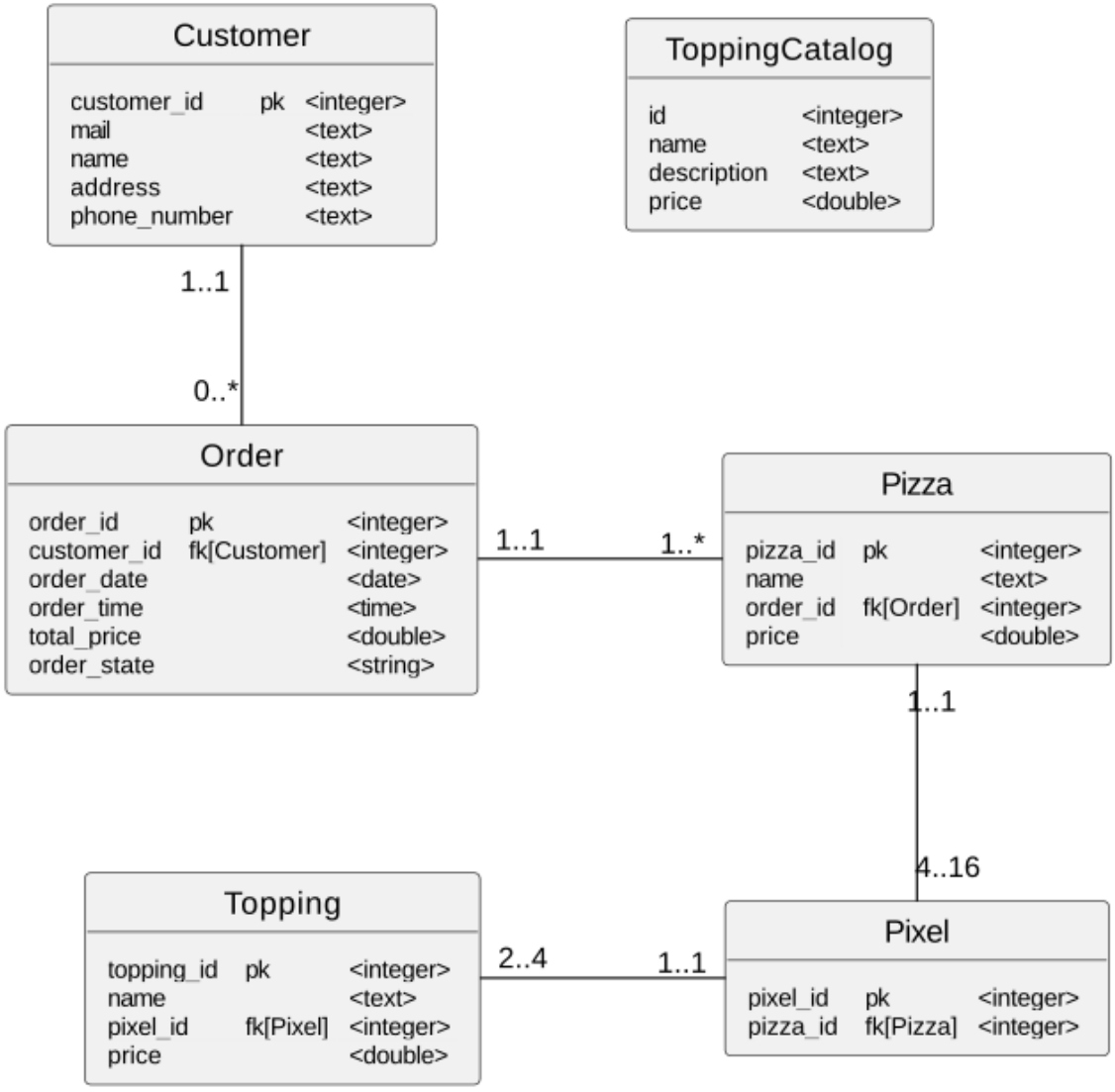
Leseoperationen:

- **R1 (Kundendaten)**
  - Gegeben ist eine E-Mail-Adresse, gebe alle Daten des Customer aus.
- **R2 (Topping-Liste)**
  - Gebe eine Liste aller angebotenen Toppings aus (*name*, *description*, *price*).
- **R3 (Eingegangene Bestellungen)**
  - Zeige eine List mit Bestellungen für den heutigen Tag (*order_date*, *order_time*, *order_id* sowie Pizzen und Pixel mit Toppings) nach Uhrzeit sortiert an, deren Status "Bestellung aufgenommen" aufweist.
- **R4 (Lieferbereite Bestellungen)**
  - Zeige eine Liste von Bestellungen (*order_date*, *order_time*, *order_id*, *customer_name*, *customer_address*, *customer_phone_number*, *total_price* sowie der dazugehörigen Pizzen inklusive Pixeln und Toppings) mit dem Status "Bereit zur Lieferung", sortiert nach dem Datum bzw. der Uhrzeit der Bestellung an.

Schreiboperationen

- **W1 (Neue Kundin)**
  - Gegeben ist eine E-Mail-Adresse, ein Name, eine Adresse und eine Telefonnummer, füge eine neue Kundin hinzu.
- **W2 (Neue Bestellung)**
  - Gegeben ist ein Datum, eine Uhrzeit, die *customer_id* (außerdem optional der *name*, die *address* und *phone_number* des Customer), eine Liste mit einer oder mehreren Pizzen, deren Pixel, die Toppings der Pixel sowie die entsprechenden Preise, füge eine neue Bestellung hinzu.
- **W3 (Bestellstatus)**
  - Gegeben ist eine *order_id* (außerdem optional das Datum und die Uhrzeit der Bestellung), update den Status der Bestellung.

Die nachfolgende Grafik zeigt einen vereinfachten Ausschnitt des aktuellen physischen Datenbankschemas (PostgreSQL) von Pippas Pixeria.



Schema der relationalen Datenbank

Aufgabe 3.1 (12 Punkte)

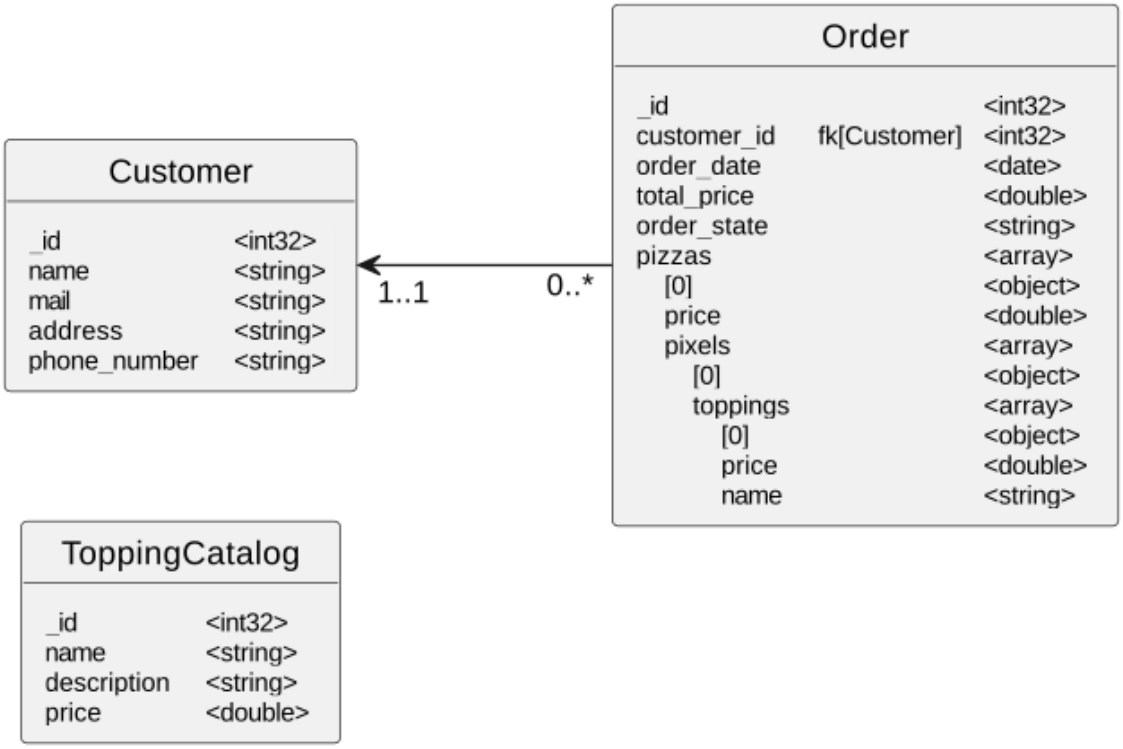
Da die IT-Abteilung noch nicht viel Erfahrung bei der Migration bzw. Modellierung von MongoDB-Schemata hat, wurden zunächst 4 verschiedene Vorschläge entwickelt.

> Beschreiben Sie zu jedem Schemavorschlag die Nachteile bezüglich der 7 Abfragen und wählen anschließend das am besten geeignetste Schema anhand Ihrer Begründungen aus.

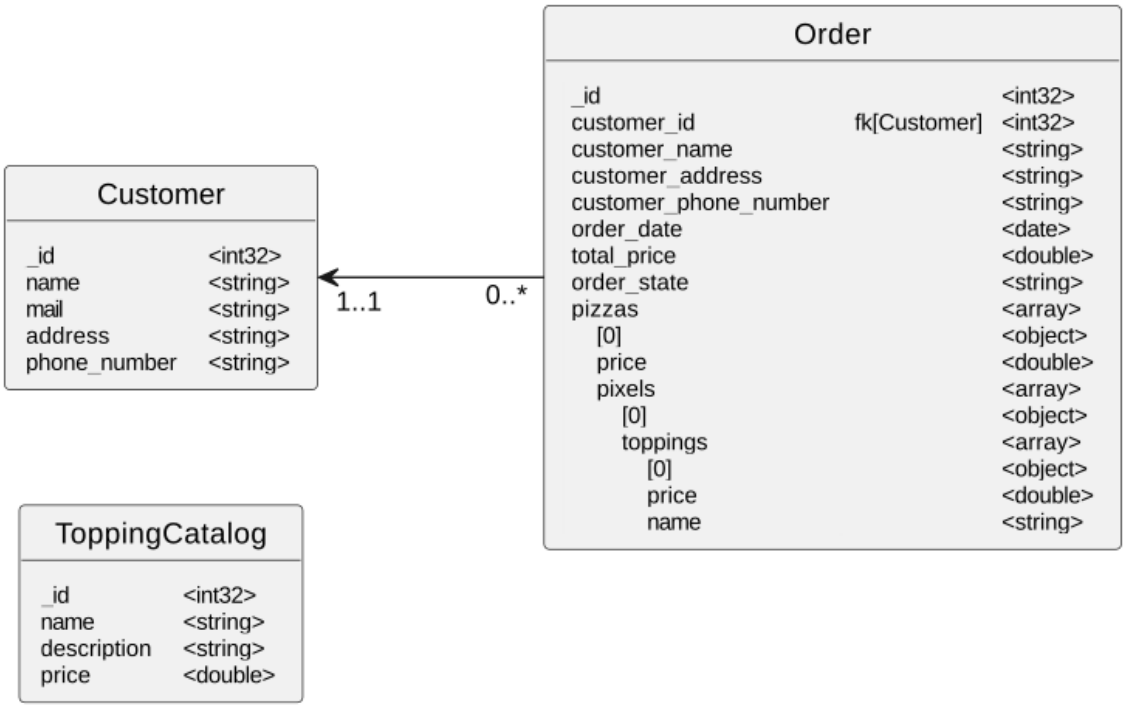
HINWEISE:

- Indexe können vernachlässigt werden.
- Die *order_time* ist hier in *order_date* enthalten, da MongoDB im Vergleich zu PostgreSQL und Cassandra keinen Datentyp "time" aufweist.

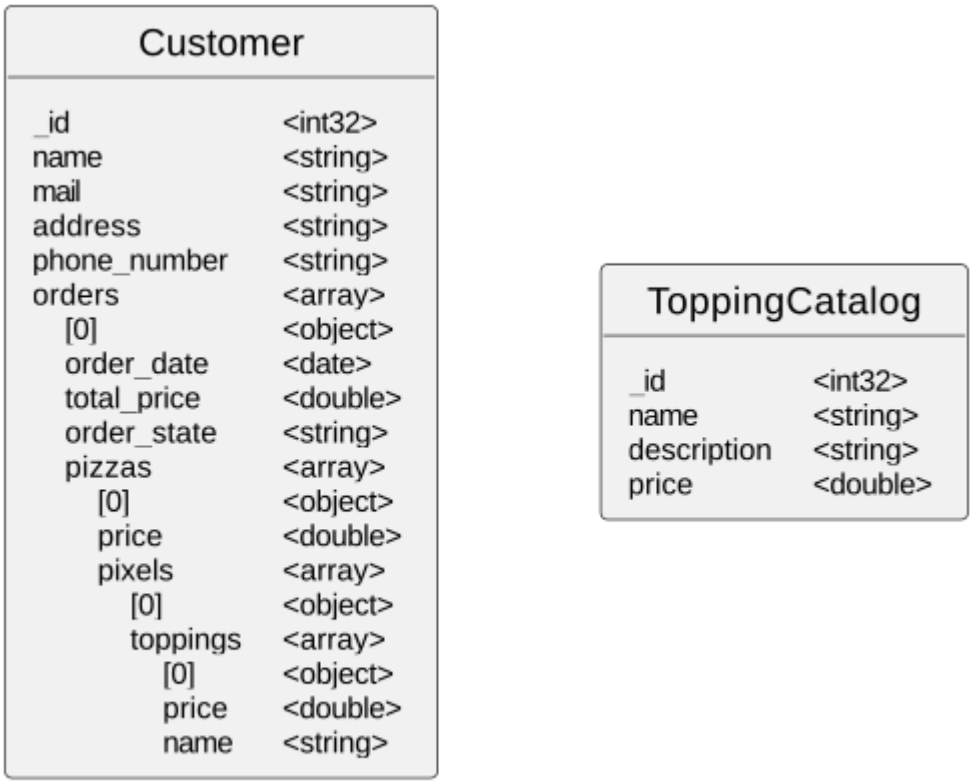
- Bei W1 und W2 kann das Inkrementieren der Primärschlüssel vernachlässigt werden.



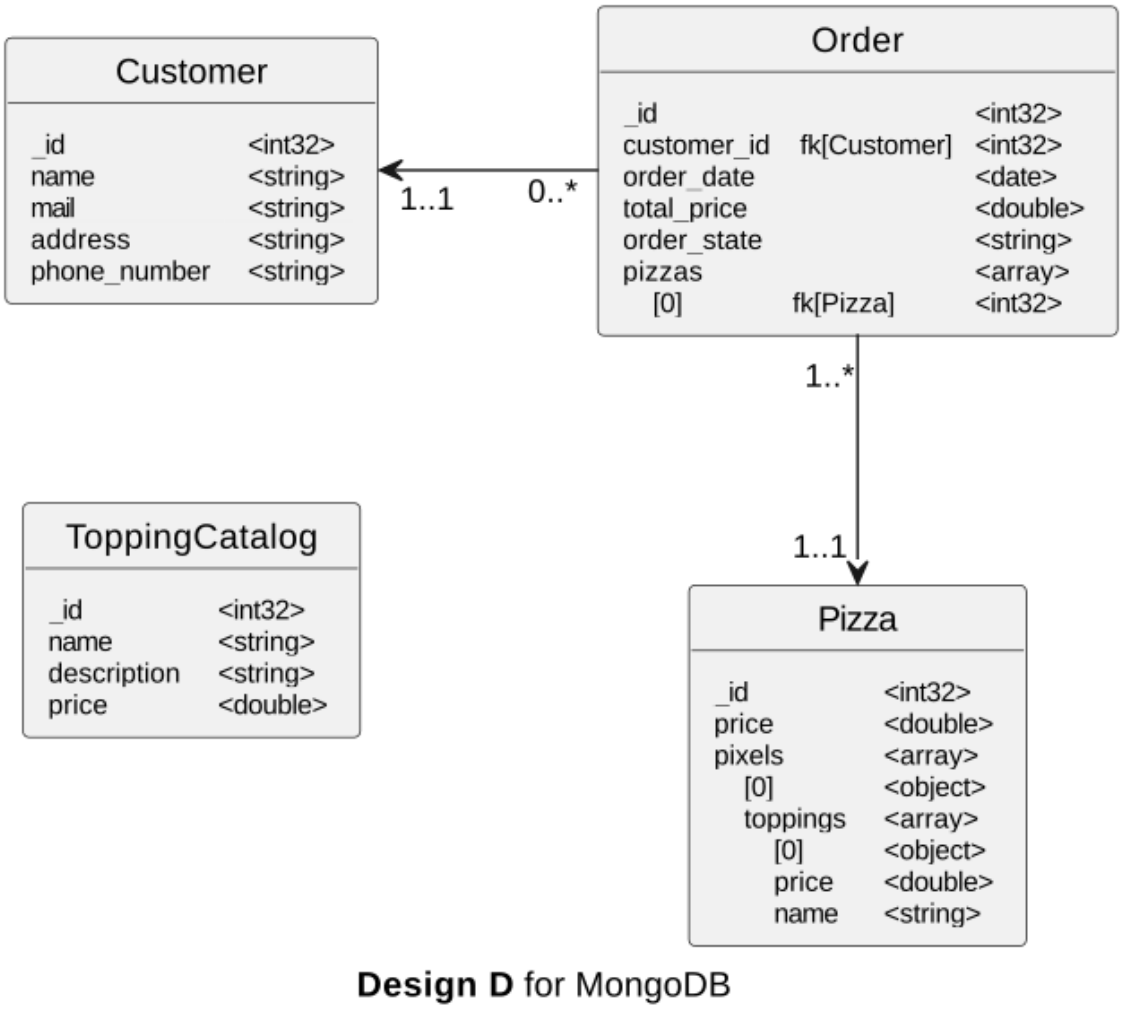
Design A for MongoDB



Design B for MongoDB



Design C for MongoDB



Design D for MongoDB

Ihre Antwort:

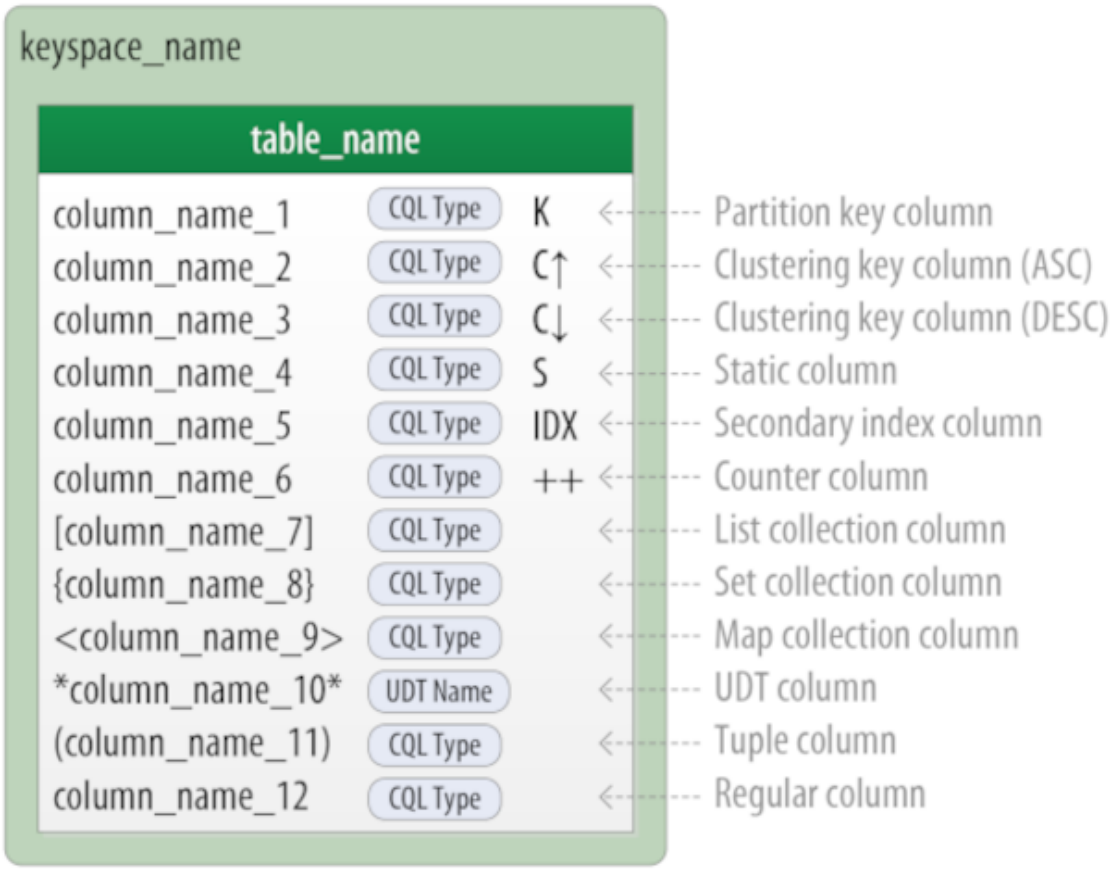
Aufgabe 3.2 (8 Punkte)

Die IT-Abteilung hat ebenfalls noch wenig Erfahrung bei der Migration bzw. Modellierung von Cassandra-Schemata und deshalb ebenfalls zunächst 3 verschiedene Vorschläge entwickelt.

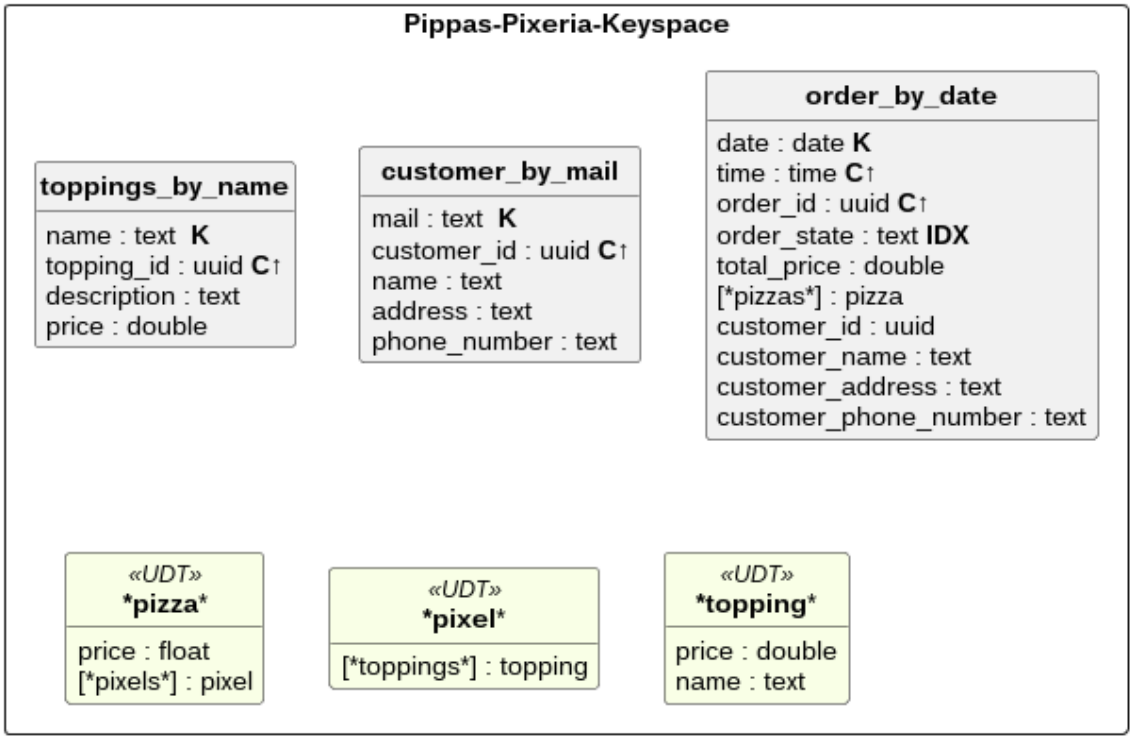
> Beschreiben Sie ebenfalls zu jedem Schemavorschlag die Nachteile bezüglich der 7 Abfragen und wählen anschließend das am besten geeignetste Schema anhand Ihrer Begründungen aus.

HINWEIS: Für eine bessere Lesbarkeit wurden "pizza", "pixel" und "topping" teilweise als User Defined Type (UDT) dargestellt.

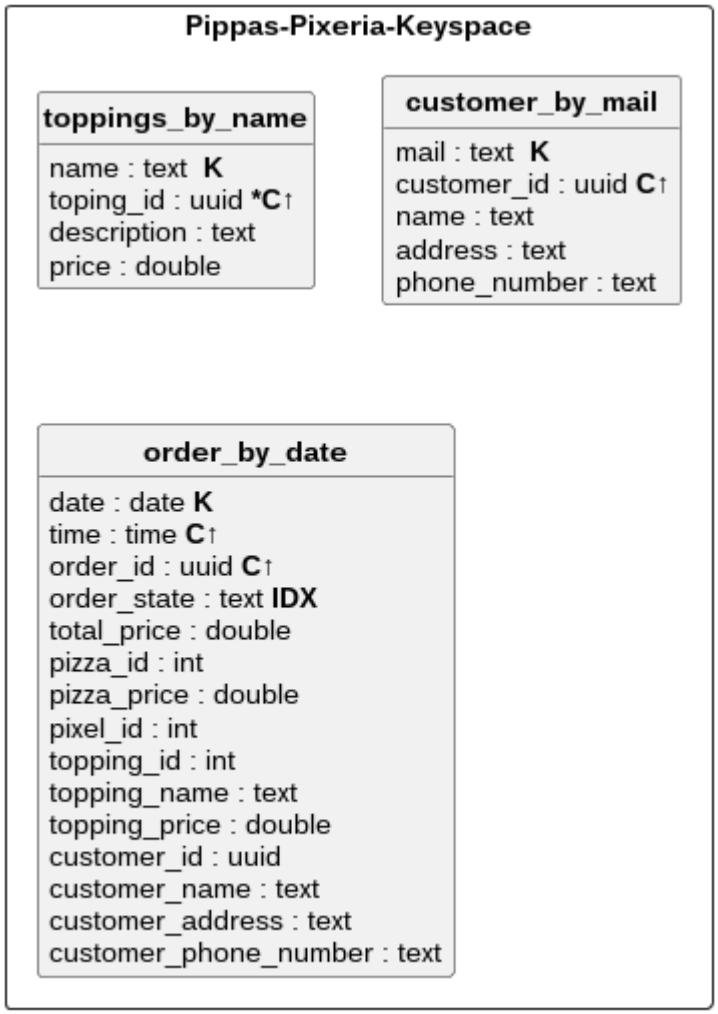
Im Folgenden finden Sie außerdem eine Übersicht der Chebotko Notation physikalischer Modelle in Cassandra.



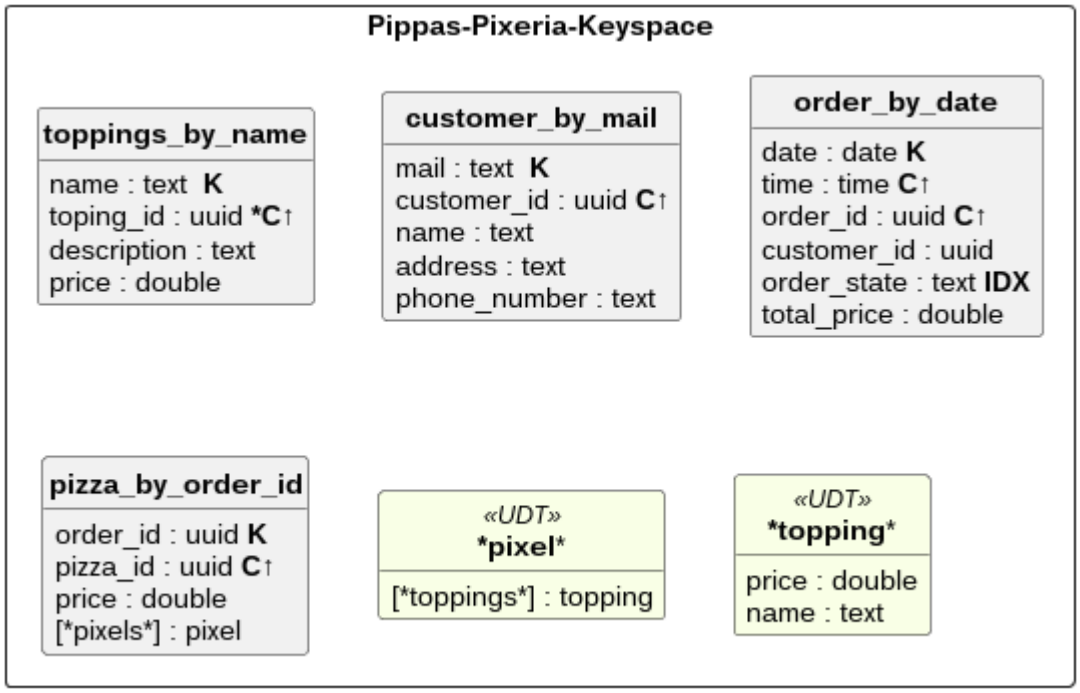
Source: https://cassandra.apache.org/doc/stable/cassandra/data_modeling



Design A for Cassandra



Design B for Cassandra



Design C for Cassandra

Ihre Antwort:

Aufgabe 3.3 (5 Punkte)

Das Bestellsystem soll erweitert werden, sodass den Kundinnen zunächst folgende Vorschläge angezeigt werden:

- Bei der Zusammenstellung der Pixel bzw. der Auswahl der Toppings soll auf Basis des eigenen Bestellverhaltens das am häufigsten ausgewählte bzw. zusammengestellte Pixel vorgeschlagen werden.
- Außerdem sollen Pixel vorgeschlagen werden, die am häufigsten gemeinsam mit dem gerade ausgewählten Pixel bestellt wurden (ebenfalls anhand des individuellen Verhaltens).

Dies soll nicht in Echtzeit berechnet werden, sondern die hierfür notwendigen Daten während des Bestellvorgangs gespeichert werden.

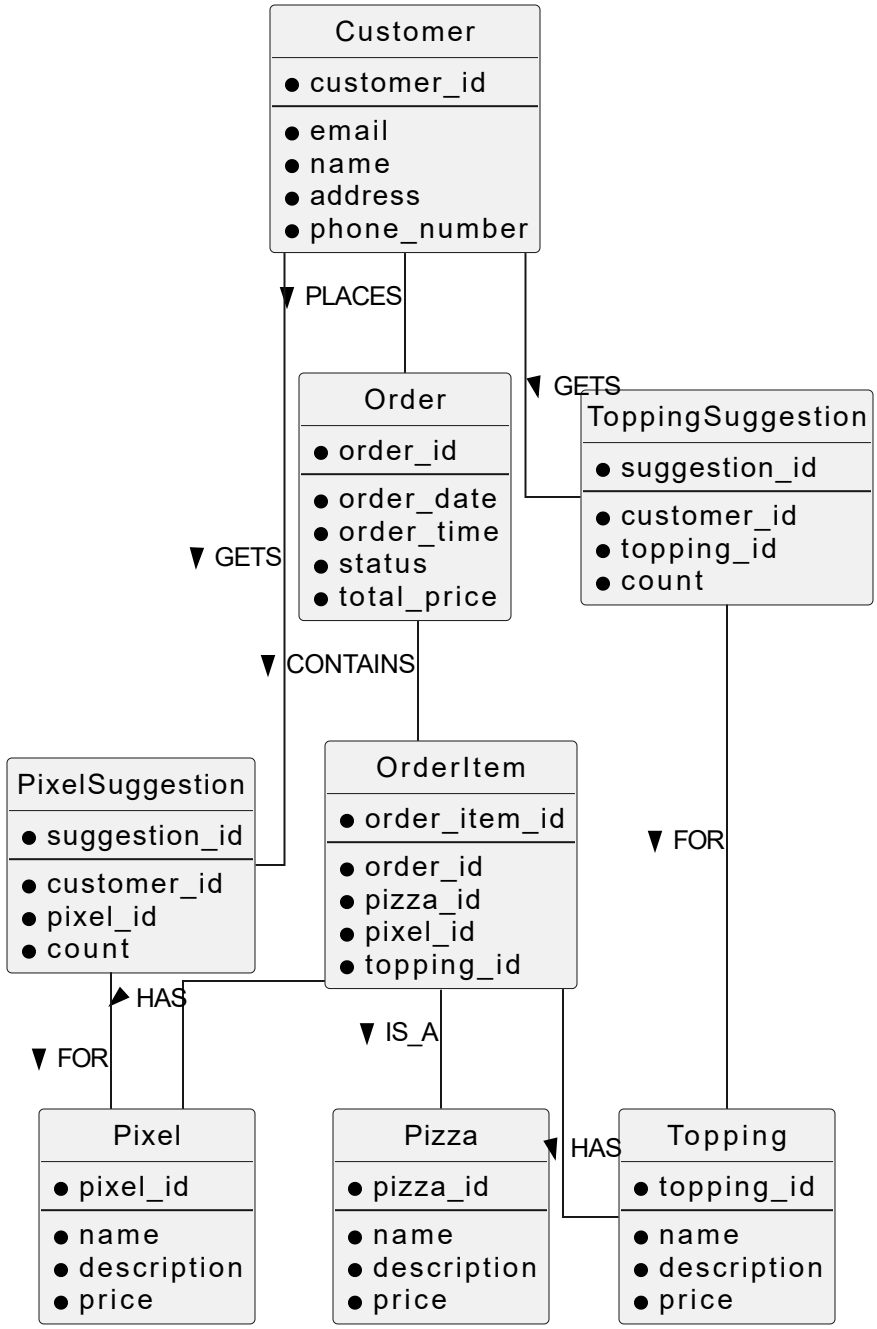
> Erweitern Sie hierzu auf einer abstrakten Ebene das folgende Modell um entsprechende Entitätstypen, Attribute und Beziehungen.

In [53]:
```
%%plantuml

@startuml

hide circle
hide members
show fields
```

```
skinparam linetype ortho

entity Customer {
  *customer_id
  --
  *email
  *name
  *address
  *phone_number
}
entity Order {
  *order_id
  --
  *order_date
  *order_time
  *status
  *total_price
}
entity Pizza {
  *pizza_id
  --
  *name
  *description
  *price
}
entity Pixel {
  *pixel_id
  --
  *name
  *description
  *price
}
entity Topping {
  *topping_id
  --
  *name
  *description
  *price
}
entity OrderItem {
  *order_item_id
  --
  *order_id
  *pizza_id
  *pixel_id
  *topping_id
}
entity PixelSuggestion {
  *suggestion_id
  --
  *customer_id
  *pixel_id
  *count
}
entity ToppingSuggestion {
  *suggestion_id
```

```
  --
  *customer_id
  *topping_id
  *count
}

Customer -- Order : PLACES >
Order -- OrderItem : CONTAINS >
OrderItem -- Pizza : IS_A >
OrderItem -- Pixel : HAS >
OrderItem -- Topping : HAS >
Customer -- PixelSuggestion : GETS >
PixelSuggestion -- Pixel : FOR >
Customer -- ToppingSuggestion : GETS >
ToppingSuggestion -- Topping : FOR >

@enduml
```

Out[53]:



Diese Zelle bitte auf keinen Fall löschen, da der Seitenumbruch für die Korrektur benötigt wird.

\newpage

Aufgabe 4 - Technologien für Big Data (5 Punkte)

Aufgabe 4.1 (3 Punkte)

Pippas Pixeria ist inzwischen so erfolgreich, dass sie nun auch außerhalb von Pixhagen ihre Pizzen vertreibt. Aus Performance- und Sicherheitsgründen werden die Bestellungen nun

über eine verteilte Datenbank mit drei Knoten A, B und C abgewickelt. Dabei kommen in dem verwendeten Datenbanksystem intern Vektor Clocks zum Einsatz.

Eine Read-Anfrage für einen Wert wird an Knoten B und C geschickt. Untenstehend sehen Sie zwei verschiedene Szenarien, welche Vektor Clocks von den beiden Knoten zurück gegeben wird.

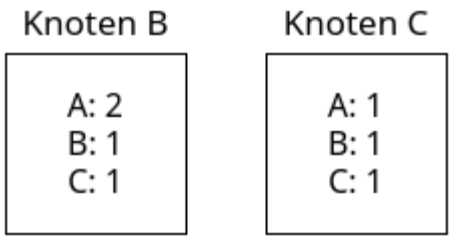
Beantworten Sie jeweils die Fragen:

Kann entschieden werden, welcher Knoten den aktuelleren Wert enthält?

Wenn ja, welcher Knoten enthält den aktuelleren Wert und warum?

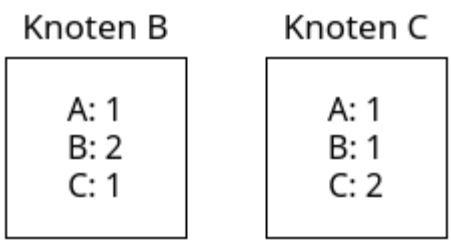
Wenn nein, warum nicht und was könnte passiert sein?

Szenario 1:



Ihre Antwort: #REgeln 1. Wenn für jeden Eintrag in einem Vektor Clock Knoten B gleich oder größer ist als Knoten C und mindestens ein Eintrag streng größer ist, dann enthält Knoten B den aktuelleren Wert. 2. Wenn für jeden Eintrag in einem Vektor Clock Knoten C gleich oder größer ist als Knoten B und mindestens ein Eintrag streng größer ist, dann enthält Knoten C den aktuelleren Wert. 3. Ansonsten können wir nicht entscheiden, welcher Knoten den aktuelleren Wert enthält. Angewendet auf Szenario 1: 1. Knoten B: A:2 > A:1 (Knoten C), B:1 = B:1, C:1 = C:1 -> Regel 3 2. Knoten C: A:1 < A:2 (Knoten B), B:1 = B:1, C:1 = C:1 -> Regel 3 Beide Knoten haben keinen streng größeren Eintrag im Vergleich zum anderen. Daher können wir nicht entscheiden, welcher Knoten den aktuelleren Wert enthält. Es ist möglich, dass in diesem Fall eine Kommunikationsverzögerung zwischen den Knoten aufgetreten ist, die dazu gefüührt hat, dass die Vektor Clocks nicht synchronisiert wurden. oder es könnte sein, dass die Knoten verschiedene Zustände basierend auf unterschiedlichen Aktualisierungszeiten haben, und es keine eindeutige Antwort darauf gibt, welcher Zustand der aktuellere ist.

Szenario 2:



Ihre Antwort: 1. Knoten B: A:1 = A:1, B:2 > B:1 (Knoten C), C:1 = C:2 (Knoten C) -> Regel 1 2. Knoten C: A:1 = A:1, B:1 < B:2 (Knoten B), C:2 > C:1 (Knoten B) -> Regel 2 In diesem Fall ist Knoten B in Bezug auf den Eintrag für Knoten B größer als Knoten C, und Knoten C ist in Bezug auf den Eintrag für Knoten C größer als Knoten B. Daher können wir entscheiden, dass Knoten C den aktuelleren Wert enthält. Dies deutet darauf hin, dass Knoten C einen neueren Zustand des Wertes hat als Knoten B.

Aufgabe 4.2 (2 Punkte)

Für die Erstellung von Analysen für den Vertrieb nutzt die Pixeria eine spalten-orientierte Datenbank. Dabei werden u.a. die Verkaufsdaten der einzelnen Toppings gespeichert. In der untenstehenden Abbildung sehen Sie einen Ausschnitt der Spalte, in der jeweils der Name der verkauften Toppings gespeichert wird. Welche Art der Komprimierung ist für diese Daten am besten geeignet und warum?

Ihre Antwort: Für die vorliegenden Verkaufsdaten der Toppings, die sich in einer Spalte befinden, scheint die **Run-Length-Encoding (RLE)** am besten geeignet zu sein. Run-Length-Encoding basiert auf der Idee, dass aufeinanderfolgende identische Werte in den Daten durch eine einzige Instanz dieses Werts und die Anzahl der aufeinanderfolgenden Wiederholungen dieses Werts ersetzt werden. Da die Verkaufsdaten der Toppings wahrscheinlich viele wiederkehrende Werte enthalten (z.B. "Salami" erscheint mehrmals), würde RLE dazu beitragen, die Anzahl der eindeutigen Werte zu reduzieren und somit den Speicherplatz zu optimieren.