Was gibt es aus Vermont, abseits neuer Ben & Jerrys Geschmacksrichtungen – Die Entwicklung von Data Vault 2 zu Data Vault 2.1

Was hat sich in den letzten drei Jahren getan?

- Vor Data Vault 2.1 wurden die Begriffe BKCC und Applied Date eingeführt,
- das Driving Key Pattern erweitert und
- mit Data Vault 2.1 rückt der Schwerpunkt mehr Richtung Logical Modeling und
- JSON Verarbeitung rückt konkreter in den Fokus.

Die bahnbrechende, alles in Frage stellende Neuigkeit:

tagesschau tagesschau24 live

Eismarken-Gründer verlässt Firma

Ben nun ohne Jerry

Stand: 17.09.2025 17:08 Uhr

Die US-Eishersteller Ben & Jerry's gehört seit vielen Jahren zum Großkonzern Unilever. Nun verlässt einer der Mitgründer das Unternehmen - im Streit über die politische Positionierung der Marke.

Die Eismarke Ben & Jerry's verliert einen ihrer Mitgründer: Jerry Greenfield hat seinen Rückzug aus dem Unternehmen angekündigt. Vorausgegangen war ein Streit mit dem Mutterkonzern Unilever. Greenfield wirft dem Unternehmen vor, politische Aktivitäten unterbunden zu haben. Die Eismarke ist bekannt für ihre Positionierung in gesellschaftspolitischen Debatten. Beim Verkauf ihrer Eismarke im Jahr 2000 hatten die beiden Gründer sich explizit das Recht gesichert, damit fortfahren zu können.

Über 20 Jahre lang habe man sich auch unter dem Dach von Unilever "für Frieden, Gerechtigkeit und Menschenrechte" eingesetzt, jetzt sei Ben &

Im Ernst, worüber sprechen wir:

- Das neue Training
  - Neue Struktur -> Recordings, Quizzies, Practice Exam
  - Ausweitung der (modernen) Grundlagenbegriffe
- DV 2.1 Inhaltliches
  - Schwerpunktverlagerung auf Logisches Modell
  - BKCC, Applied Date
  - Effectivity Sat
  - Exploration Link & PIT/BRIDGE Hybrid
  - JSON

Das neue Training
Neue Struktur -> Recordings, Quizzies, Practice Exam

Das neue Training
Ausweitung der (modernen) Grundlagenbegriffe

Bronze: Transient (formerly Landing),
Landing (Managed PSA. Lake, …)
Silber: Raw Vault
Gold: BDV, Info Mart

Data Lake
Delta Lake
Data Mesh
Data Hub
Fabric

DV 2.1 Inhaltliches
Schwerpunktverlagerung auf Logisches Modell

Wir haben eine wesentlich breitere
Palette an Datenplattformen
-> Flexibleres physisches Modell

DV 2.1 Inhaltliches
BKCC, Applied Date

BKCC

- Business Key Collision Code
- Element des Combined Business Key
- Never, never, never Record Source

Applied Date

- Metadaten Extraktion und Transport
- Wiederherstellung der Reihenfolge
- Umgang mit Late Arriving Data

DV 2.1 Inhaltliches Link Effectivity Sat (am Driving Key)

LHK	LDTS	START_TS	END_TS	RCRDSRC
123abc…	t_1	t_1	∞	sysa.tbla
123abc…	t_2	t_1	t_2	sysa.tbla
987fed…	t_2	t_2	∞	sysa.tbla

Var1

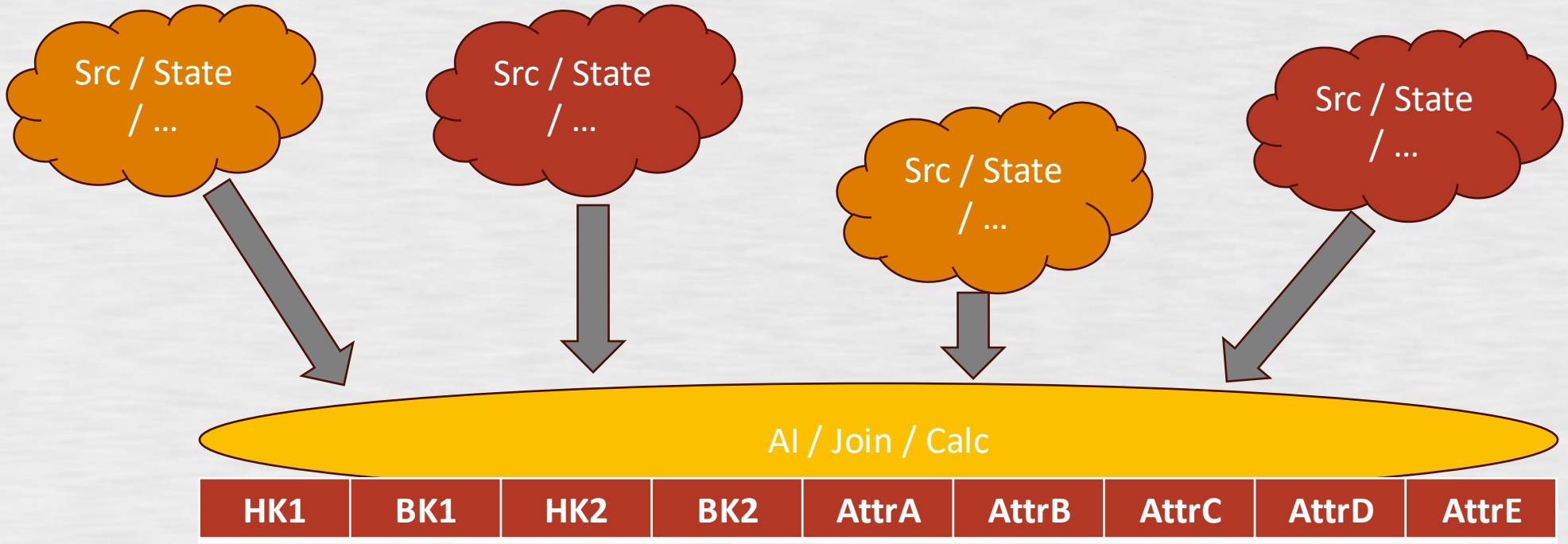
LHK	LDTS	OLD_LHK
123abc…	t_1	NULL
987fed…	t_2	123abc…

Var2

Linkeinträge hineinkopieren für mehr Performance
(Join in den Link sparen)

LHK	LDTS	BO1_HKD	BO2_HK	BO2_HK	OLD_LHK
123abc…	t_1	5d7ea2…	dd55a7…	c1d2e3…	NULL
987fed…	t_2	5d7ea2…	12cf37…	c1d2e3…	123abc…

DV 2.1 Inhaltliches
Exploration Link & PIT/BRIDGE Hybrid



DV 2.1 Inhaltliches
JSON

Wo kommen wir her? - Vergleich Strukturierter Tabellen mit JSON

- Was uns vertraus ist:
  - 1 Dimension (eine Reihe Spalten)
  - Sichere Datentypen
  - Festes Schema (on write)

VS:

- Was jetzt zu beachten ist :
  - Unbestimmte Anzahl von hierarchischen EbenenWeak data types
  - Flexibles Schema (on read)

Was haben wir in DV2 über strukturierte Daten gelernt, und warum ist die Verarbeitung von JSON gleich und doch anders?

- Auditierbarkeit
  - Die Fähigkeit, das geladene Objekt zu rekonstruieren
- Feld für Feld in H, L, S aufteilen
  - Einfach wieder zurück joinen
- Spätere Anwendung von Geschäftsregeln
  - Alles, was die (Roh)daten verändert

In JSON will auch die Struktur rekonstruiert werden!

In JSON sehr komplex -> eine Menge Regeln dynamisch anzuwenden.

Ist Umwandlung von Semi- in strukturiert eine Hard Rule??

So laden wir JSON in den Raw Vault

1. Konsistenzprüfung! (einschließlich Überprüfung von Erwartungen wie definierten Stammstrukturen)

- Verfügbare Frameworks nutzen
  - Um zu überprüfen, ob die Struktur gültig ist
- Überprüfen Sie, ob der richtige Geschäftsschlüssel (Kombination) vorhanden ist.
- Überprüfen Sie die Strukturen, um sie später in eine relationale Struktur zu extrahieren.

2. Vollständiges Laden über Root-BK (Kombination)

- Reibungslose Absorption sicherstellen
- 100 % rückführbar in Bezug auf Inhalt und Struktur
- Laden des übergeordneten Hubs oder übergeordneten Links mit dem JSON-Auszug des Schlüssels
- Laden des Sat mit übergeordnetem HK und vollständiger JSON-Struktur

3. Verwendung von Error Mart zum Auffangen von unerwarteten Problemen in der JSON-Struktur

- Jede JSON-Struktur, die nicht absorbiert wird, wird in den Error-Vault verschoben.
  - Zur späteren Analyse der Ursache
  - Zur Rückführbarkeit

So laden wir JSON in den Raw Vault – Hub Load

- Der (kombinierte) Geschäftsschlüssel wird unverändert (Key-Value, geschweifte Klammern usw.) kopiert und in einer Variablen gespeichert.
- Der Haskey ist der Hash dieses Variant (Zeichen)..

Warum?

- Weniger BK Format Manipulation
- Flexibler wenn der BK die Struktur ände

```
CREATE TABLE part_hub (
    part_hkey binary(40), -- Hash key
    part_load_dts TIMESTAMP_NTZ,
    part_rec_src varchar(50),
    part_BK VARIANT, -- JSON
    PRIMARY KEY (part_hkey,
       part_sat_load_dts)
);
```

part_hkey = hash(part_bk)

Still in the labs!
Sharing experience appreciated!

So laden wir JSON in den Raw Vault – Link Load

Gleich / ähnlich wie Hub Load

- Die (kombinierte) Geschäftsschlüsselkombination wird unverändert kopiert (Schlüsselwert, geschweifte Klammern usw.) und in einem Variant gespeichert.
- Mit den Beziehungsdetails, was es zu einem Link/Sat Hybrid macht
- Der Haskey ist der Hash dieses Variant (Zeichen).
- Hashdiff einführen, um neue Beziehungen zu erkennen

Why?

- Weniger BK Format Manipulation
- Flexibler, wenn sich die Kombinationsstruktur d ändert

```
to LNK_VIN_OWNER (VINOWN_HKEY, VINOWN_LDTS, VINOWN_

8b135f9a07834313e3bf4780fd485905b',
2-18T00:14:00',
GISTRATION',
2eb2a2cfdd6169e12a79759827eebc55c1',
son( '{ "VIN":"1HGCM82633A004352",
ious_owners":[
date_of_sale":"2015-05-20", "drivers_license":"V12
date_of_sale":"2019-09-10", "drivers_license":"V87
```

So laden wir JSON in den Raw Vault – Link Load

Option 2

- Die Beziehung auf eine höhere Ebene bringen (Erstellen Sie zwei Verknüpfungen für die genaue Fahrgestellnummer und jeden Fahrer.)
- Vorteil: Umgang mit Effektivität pro Paar
- Nachteil: JSON muss „bearbeitet“ werden, was wir vermeiden wollten.

```
to LNK_VIN_OWNER (VINOWN_HKEY, VINOWN_LDTS, VINOWN_

8b135f9a07834313e3bf4780fd485905b',
2-18T00:14:00',
GISTRATION',
2eb2a2cfdd6169e12a79759827eebc55c1',
son( '{ "VIN":"1HGCM82633A004352",
ious_owners":[
date_of_sale":"2015-05-20", "drivers_license":"V12
date_of_sale":"2019-09-10", "drivers_license":"V87
```

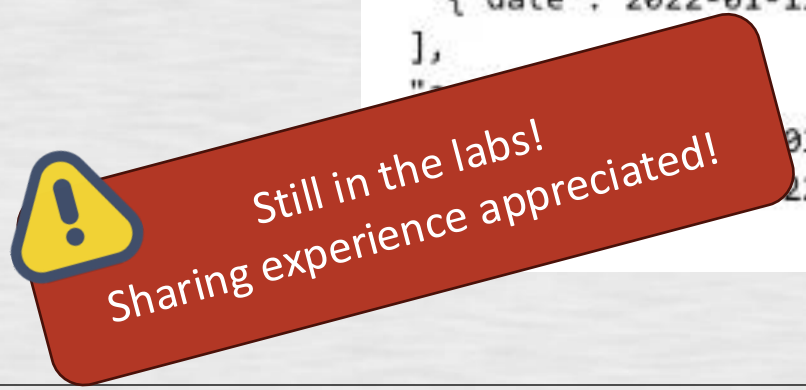
So laden wir JSON in den Raw Vault – Sat Load

- Der (kombinierte) Geschäftsschlüssel wird unverändert kopiert (Schlüsselwert, geschweifte Klammern usw.) und in HK gehasht.
- Die JSON-Datei wird unverändert in einem Variant im Sat gespeichert – ja, mit dieser „Kopie" der BK.

Warum?

- Weniger BK Format Manipulation
- Flexibler, wenn sich die Struktur des BK

```
select
 '9aaeb884bdbf7b85438e7ab26883b91853ac
  '2024-12-18T00:14:00',
  'DMV-REGISTRATION-SYSTEM',
  '0b662f399424bc848e27b931f8aac593d7c5
 parse_json('{ "VIN":"1HGCM82633A0043
   "color":"Blue", "registration_stat
 "previous_owners":[
   { "name":"John Doe","date_of_sale"
   { "name":"Jane Smith", "date_of_sa
 ],
 "maintenance_records":[
   {"date":"2021-04-10","service":"Oi
   {"date":"2022-01-15","service":"Br
 ],
                                  03","description"
                                  22","description"
```



So laden wir JSON in den Raw Vault – Sat Load

Über Delta Checks :

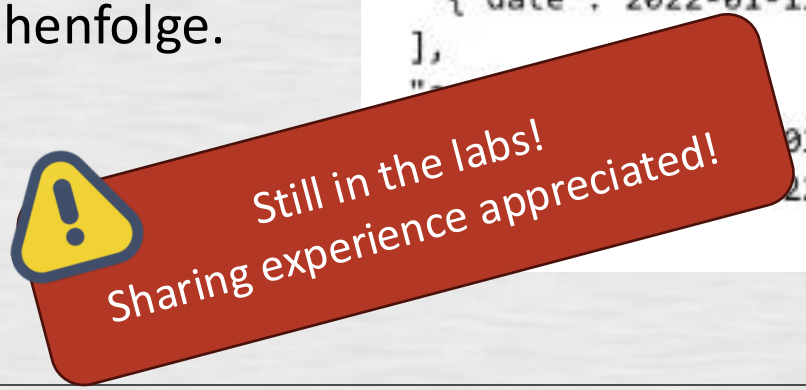
- Die JSON-Zeichenfolge in die Hashdiff-Spalte hashen

oder

- Verwenden Sie einen JSON-Delta-Parser zum Vergleichen und Überprüfen.
  - Benötigt mehr CPU-Ressourcen
  - Es ist robuster gegenüber Änderungen der Reihenfolge.

Beachten Sie die gelernten Hashing-Regeln!

```
select
 '9aaeb884bdbf7b85438e7ab26883b91853ac
  '2024-12-18T00:14:00',
  'DMV-REGISTRATION-SYSTEM',
  '0b662f399424bc848e27b931f8aac593d7c5
 parse_json('{ "VIN":"1HGCM82633A0043
   "color":"Blue", "registration_stat
 "previous_owners":[
   { "name":"John Doe","date_of_sale"
   { "name":"Jane Smith", "date_of_sa
 ],
 "maintenance_records":[
   {"date":"2021-04-10","service":"Oi
   {"date":"2022-01-15","service":"Br
 ],
                                  03","description"
                                  22","description"
```



Was macht man mit dem BV?

Nicht Teil des Training, einige Gedanken :

- Überlege Dir die Ansätze, diskutiere die Abwägung und treffe Deine Entscheidung.
- Ilch vermute, dass Geschäftsregeln strukturiert definiert sind, wodurch auch das Ergebnis strukturiert ist.
- Nutze erweitertes (Snowflake-)SQL so lange wie möglich vor der Materialisierung.

Wie ausliefern?

Auch nicht im Training, einige Gedanken:

- BI-Tools sind derzeit nicht in der Lage, JSON zu verarbeiten (wurde in der Schulung erwähnt)
- -> Konvertiere alle in ein BI-Tool zu liefernden Daten in relationalen Strukturen.
  Und wieder: Bleibe virtuell so lange wie …
- Aber …
  Eine REST-API ist ebenfalls ein Bereitstellungskonzept, für das eine JSON-Struktur gut geeignet sein kann.
  Jetzt gehst du den umgekehrten Weg.

Integration JSON / non-JSON (strukturierte Daten)

Hub Option 1:
“Beuge" die Regel, kein JSON zu generieren.

- Verwenden Sie den Spaltennamen, der für einen relationalen Hub als Schlüssel verwendet worden wäre.
- Wiederholen Sie, wenn Sie eine kombinierte BK haben.
- Optional „BKCC“ hinzufügen

Hub Option 2:
Generiere separate Hubs, integriert über einen Same-As-Link

- Erstellen Sie einen traditionellen relationalen Hub
- Laden Sie alle strukturierten Daten in Sats drumherum
- Einen neuen Standard-JSON-Hub erstellen
- Laden Sie alle semi-strukturierten Daten in Sats drumherum
- Erstellen Sie einen traditionellen Same-As-Link (relational, nur mit HKs und Metadaten)

Integration JSON / non-JSON (strukturierte Daten)

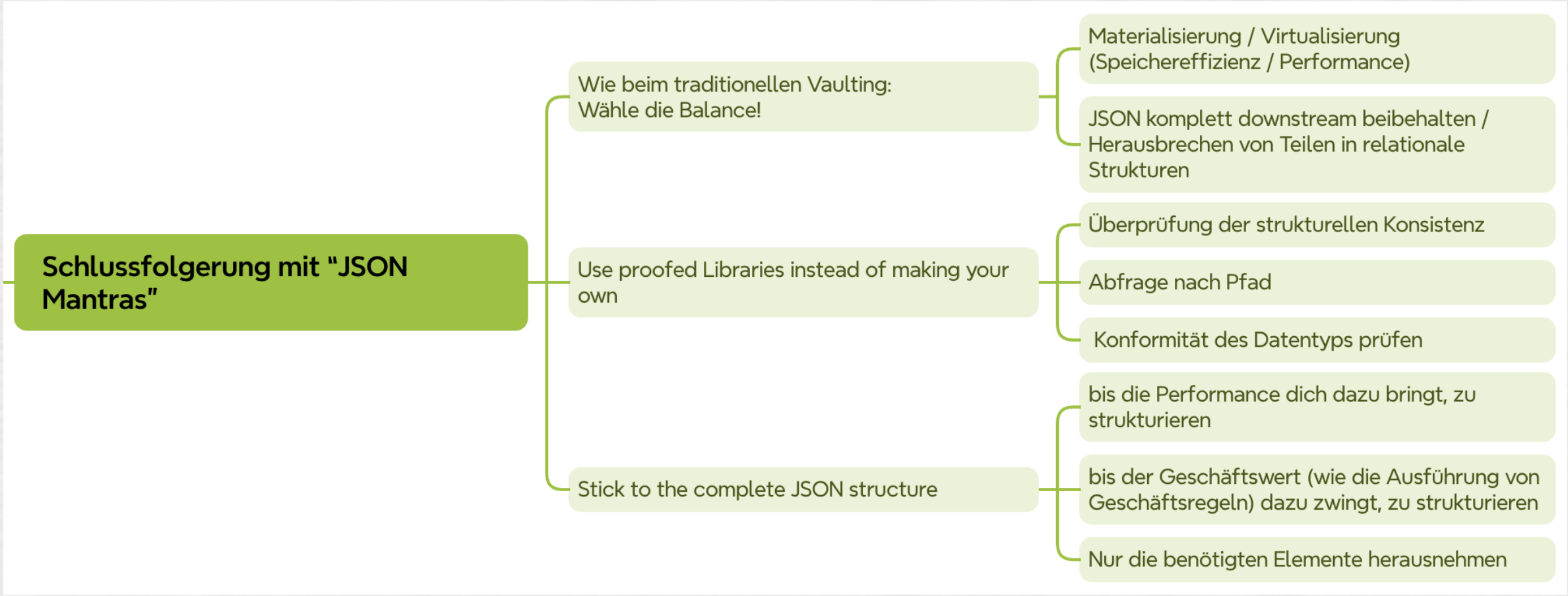
Lieferoptionion Snowflake:

Bereitstellen von Views mit der SQL-Erweiterung von Snowflake in Variant

Lieferoption Nicht-JSON-Abfrage unterstützende Plattformen :

In den sauren Apfel beißen und strukturierte Daten generieren (textuell)

Conclusion with “JSON Mantras”



Danke

SCAN ME