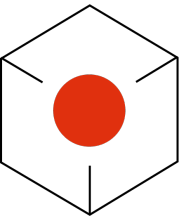


info.kennel

JSON oder relational

wie flexibel sind die Duality Views?

Dr. Andrea Kennel
InfoPunkt Kennel GmbH
Dübendorf-Schweiz
April 2024



Dr. Andrea Kennel

SYM^{L2}



Consultant

Dozentin für Datenbanken

Coach für Project Management

Fachhochschule Nordwestschweiz

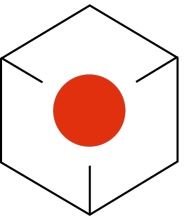
Brugg/Windisch, Schweiz



andrea.kennel@fhnw.ch

andrea@infokennel.ch

www.infokennel.ch

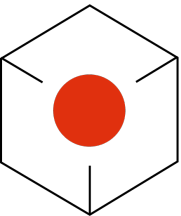


Ausgangslage

Ein neues Start-Up will einen **Web-Shop** für **Netzwerkgeräte** aufbauen. Wir sind für die Datenbank zuständig, wo die Daten abgelegt werden.

Der Web-Shop wird mit Java-Skript programmiert und die Daten zu den Netzwerkgeräten werden uns als **JSON** geliefert.

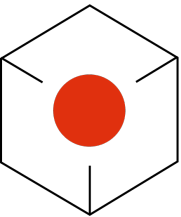
Da liegt der Einsatz von Duality Views wohl auf der Hand.



Wir analysieren das JSON

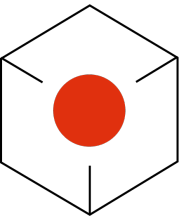
```
{  
  "type": "network switch",  
  "name": "Fritz A16",  
  "description": "unmanaged 16 port network switch",  
  "manufacturer": "ABC",  
  "price": 100,  
  "port_group": [  
    {  
      "amount": 16,  
      "type": "RJ45",  
      "speeds": "10/100/1000"  
    }  
  ]  
}
```





Wie sehen die Geräte aus?

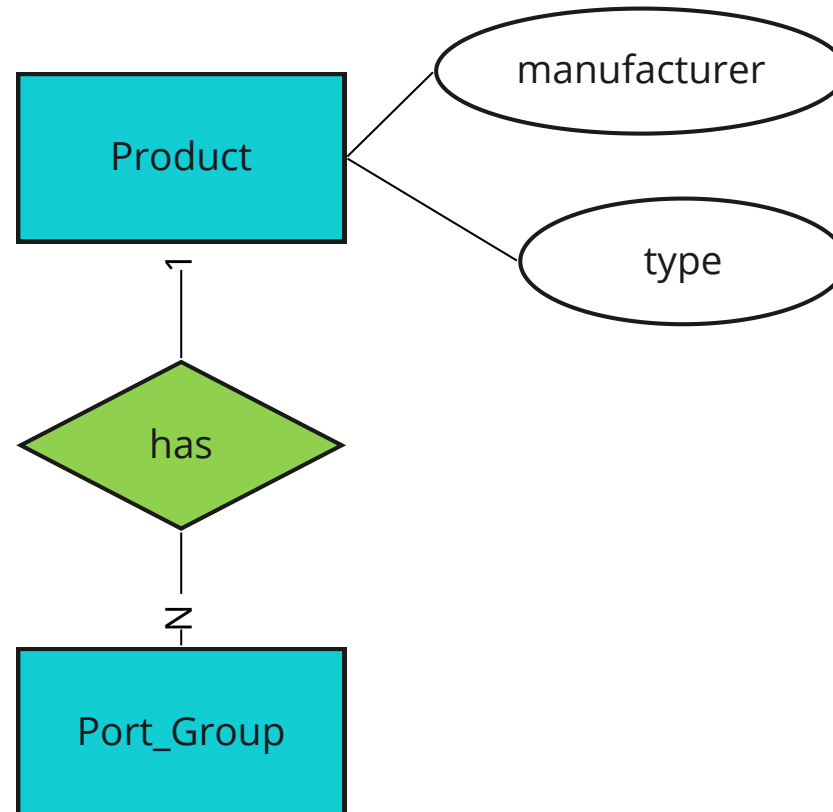
```
{  "type": "network switch",
    "name": "Fritz A16",
    "description": "unmanaged 16 port network switch",
    "manufacturer": "ABC",
    "price": 100,
    "port_group": [
      {
        "amount": 16,
        "type": "RJ45",
        "speeds": "10/100/1000"
      }
    ]
}
```

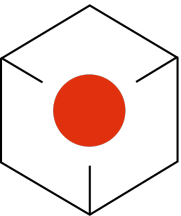


Das Datenbankmodell

Für einen Web-shop, der Netzwerkgeräte verkauft, müssen die Daten zu den Netzwerkgeräten in einer Datenbank gespeichert werden.

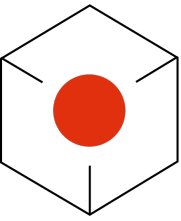
Die Daten liegen als JSON vor, das Modell ist relativ einfach:





DEMO 1

Skript 01

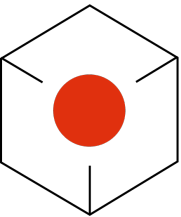


Wir haben weitere Daten

Worksheet

Query Builder

```
INSERT INTO product_dv d (data)
VALUES ( '
{
  "type": "network switch",
  "name": "Fritz C12",
  "description": "managed 12 port network switch",
  "manufacturer": "DEF",
  "price": 150,
  "port_group": [
    {
      "amount": 12,
      "type": "RJ45",
      "speeds": "10/100/1000/10000"
    },
    {
      "amount": 2,
      "type": "SFP",
      "speeds": "1000/10000"
    }
  ]
}' );
```

Wir haben weitere Daten

The screenshot shows a SQL Developer window with a 'Query Builder' tab. The main text area contains a SQL query that inserts a JSON document into a table. The JSON document is a list of two objects, each representing a network interface. The first object has 'amount': 2, 'type': 'SFP+', and 'speeds': '1000/40000/100000'. The second object has 'amount': 2, 'type': 'SFP', and 'speeds': '1000/10000'. The query also includes a 'routing' section with 'protocols': 'static, RIP, OSPF, BGP' and 'table_size': 10. The query is enclosed in a PL/SQL block and ends with a semicolon. Below the query, a status bar indicates 'Task completed in 0.336 seconds'. At the bottom, an 'Error report' section displays the following message: 'ORA-40944: Cannot insert into JSON Relational Duality View 'PRODUCT_DV': The JZN-00651: Message 651 not found; No message file for product=XDK, facility=J'.

```
},
{
  "amount": 2,
  "type": "SFP+",
  "speeds": "1000/40000/100000"
},
{
  "amount": 2,
  "type": "SFP",
  "speeds": "1000/10000"
} ],
"routing": {
  "protocols": "static, RIP, OSPF, BGP",
  "table_size": 10
}
}');

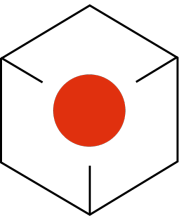
```

Script Output x Query Result x

Task completed in 0.336 seconds

Error report -
ORA-40944: Cannot insert into JSON Relational Duality View 'PRODUCT_DV': The
JZN-00651: Message 651 not found; No message file for product=XDK, facility=J

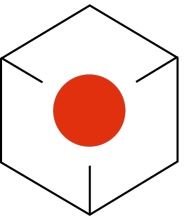
ORA-40944: Cannot insert into JSON Relational Duality View 'PRODUCT_DV': The input JSON document is invalid.



Wir haben da ein neues Gerät

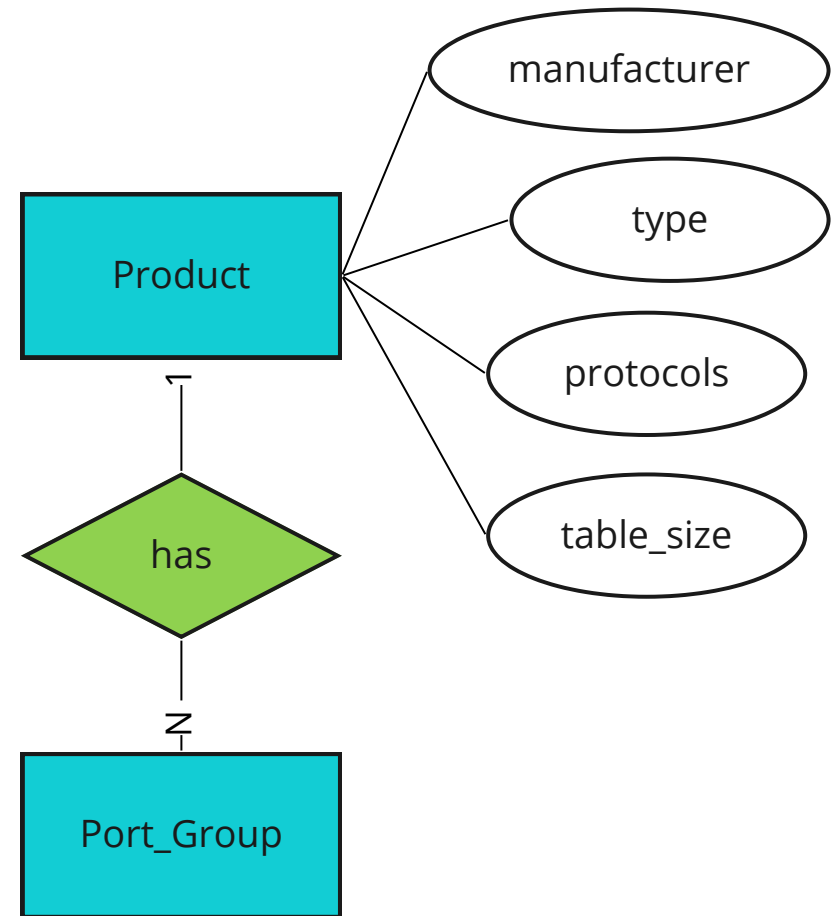
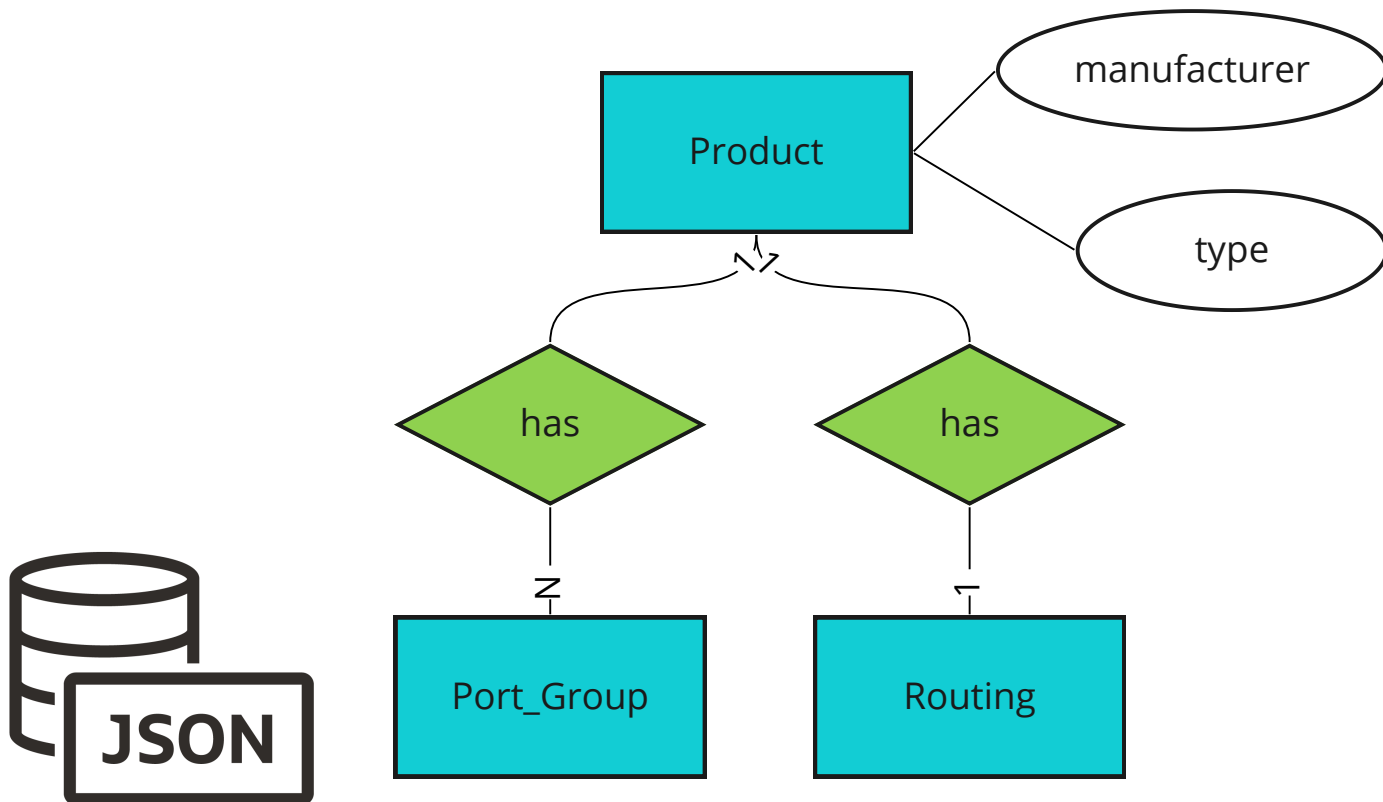
```
{  "type": "router",
    "name": "Hans B4",
    "description": "4 port router",
    "manufacturer": "XYZ",
    "price": 800,
    "port_group": [
      { "amount": 4,
        "type": "RJ45",
        "speeds": "100/1000"
      },
```

```
    { "amount": 2,
      "type": "SFP",
      "speeds": "1000/10000"
    },
    "routing": {
      "protocols": "static OSPF",
      "table_size": 5
    }
}
```



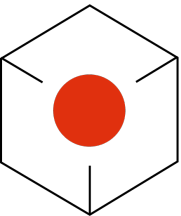
Datenmodell erweitern

*Wir benötigen auch ein Routing:
2 mögliche Lösungen*



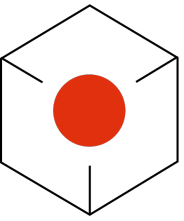
*Zusätzliche Attribute
sind einfacher*



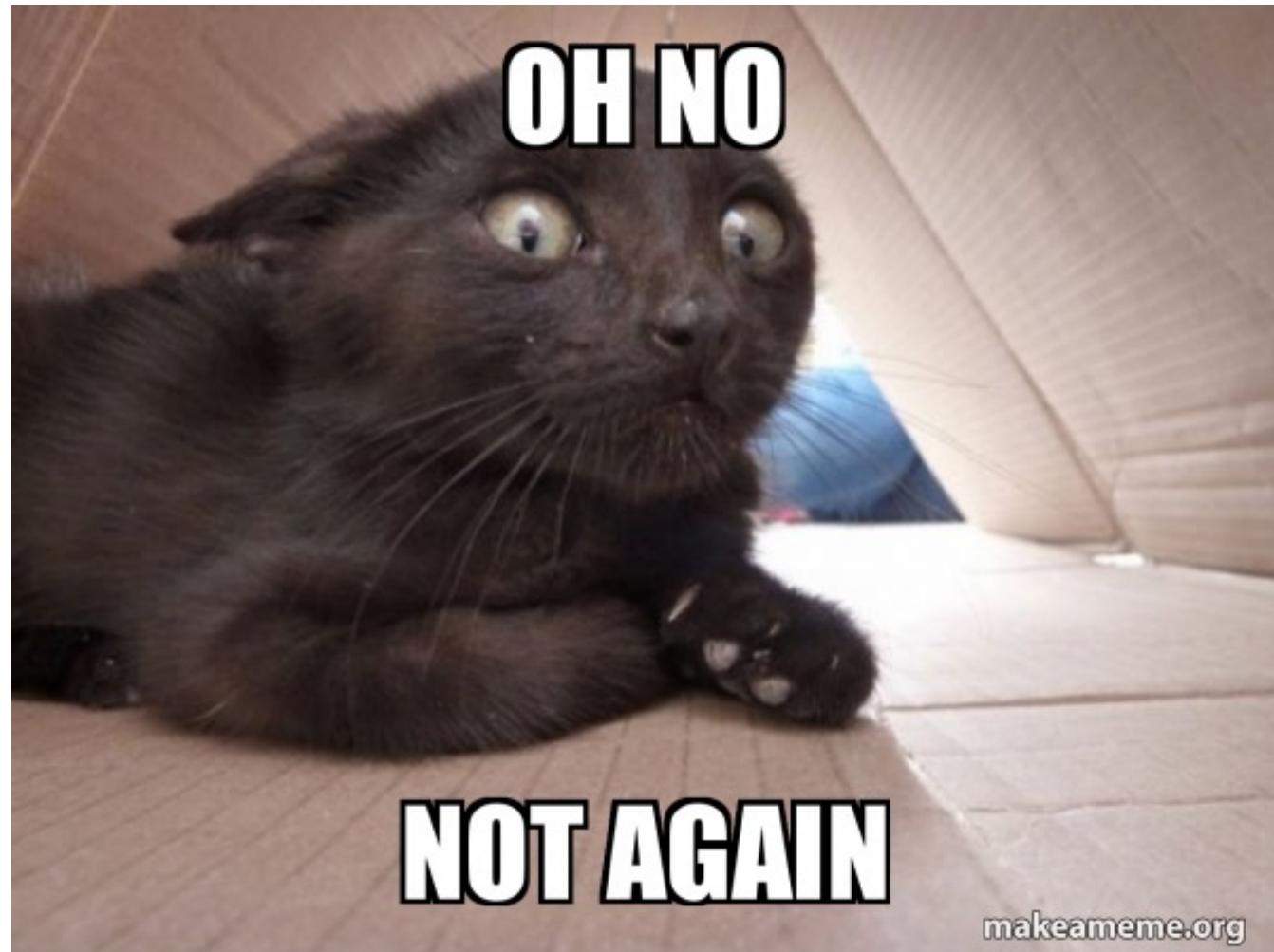


DEMO 2

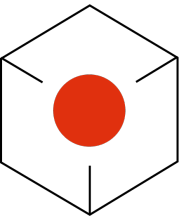
Skript 02



Wir haben wieder ein neues Gerät



<https://makeameme.org/meme/oh-no-not-6bc4c7>

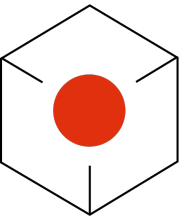


Wir haben da ein neues Gerät

```
{ "type": "network switch/layer3 switch",  
  "name": "Fritz C16",  
  "description": "16 port PoE layer 3 network switch",  
  "manufacturer": "ABC",  
  "price": 500,  
  "port_group": [  
    { "amount": 16,  
      "type": "RJ45",  
      "speeds": "10/100/1000",  
      "poe": {"modes": ["active", "passive"],  
              "volt": [24, 48]}  
    }  
  ],  
  ...  
}
```

```
"feature": [  
  { "name": "VLAN",  
    "amount": 4094},  
  { "name": "QoS",  
    "amount": 8},  
  { "name": "network access control",  
    "type": "MAC based authentication",  
    "vlan_support": true},  
  { "name": "routing",  
    "protocols": "static, RIP, OSPF, BGP",  
    "table_size": 10}  
]
```

-- *Poe = Power over Ethernet*



Dann müssen wir die Datenstruktur anpassen

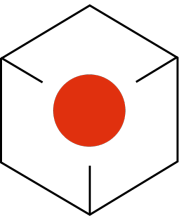
Detail-Tabelle zu Port-Group und

Feature als Generalisierung mit mehreren Spezialisierungen

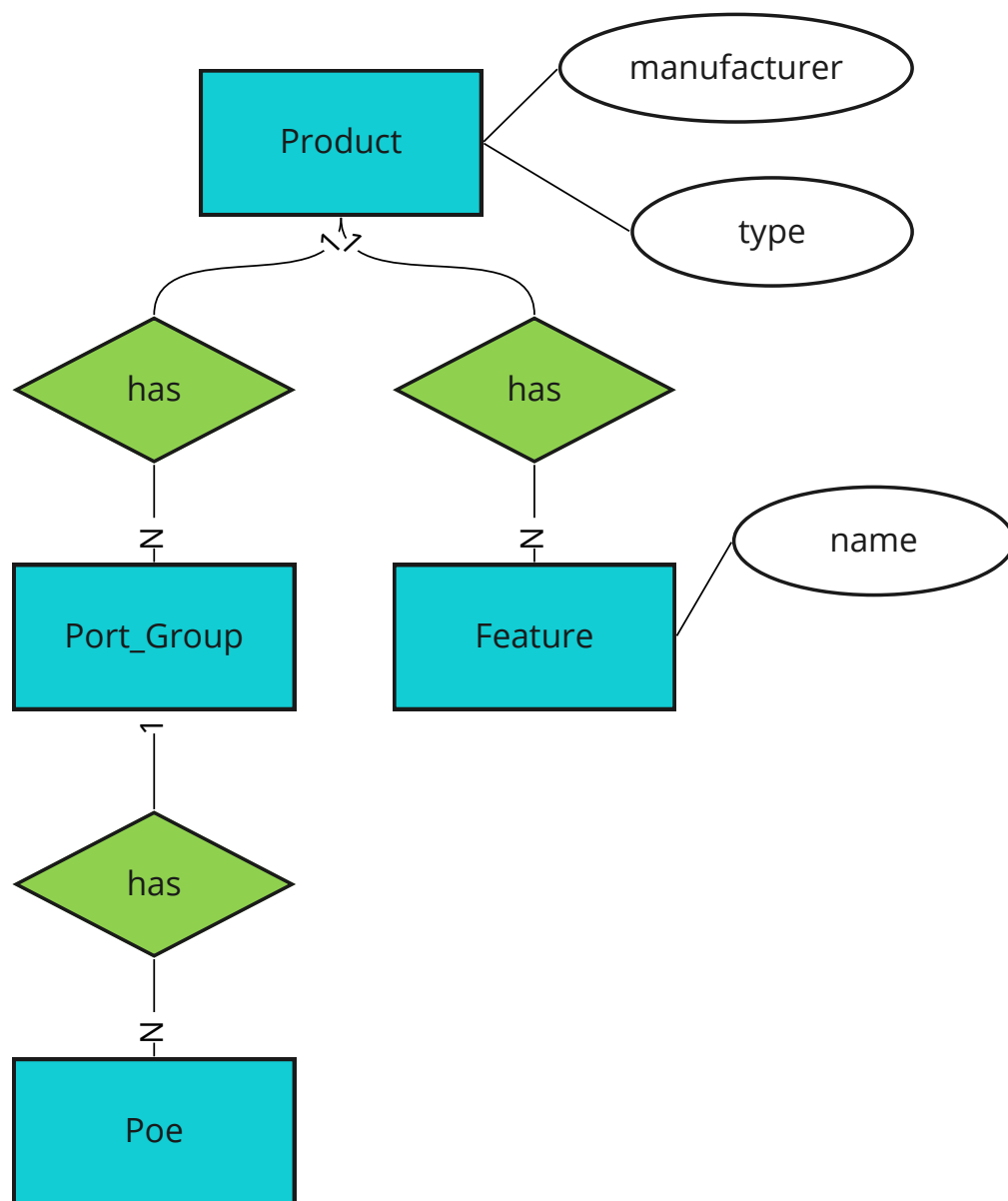
Wir wissen nicht, was noch kommt und fassen alle Attribute in der Generalisierung zusammen.

Die Duality View muss entsprechend erweitert werden und neue Views erstellt werden





Neue Datenstruktur



JSON



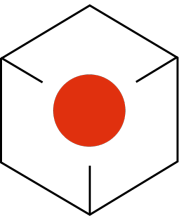


Passt das Werkzeug zum Problem?

Duality Views

Dieselben Daten als
JSON und als Tabellen

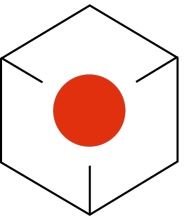
JSON als flexible
Datenstruktur



Wäre da nicht eine Dokument DB besser?

- Warum nicht einfach alle Daten als JSON speichern?
- Dann haben wir volle Flexibilität

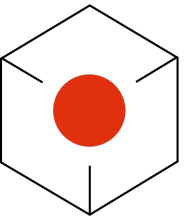




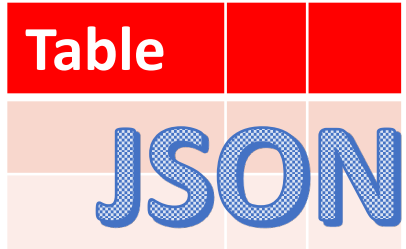
DEMO 3

Problem,
wenn JSON ein
Schreibfehler hat

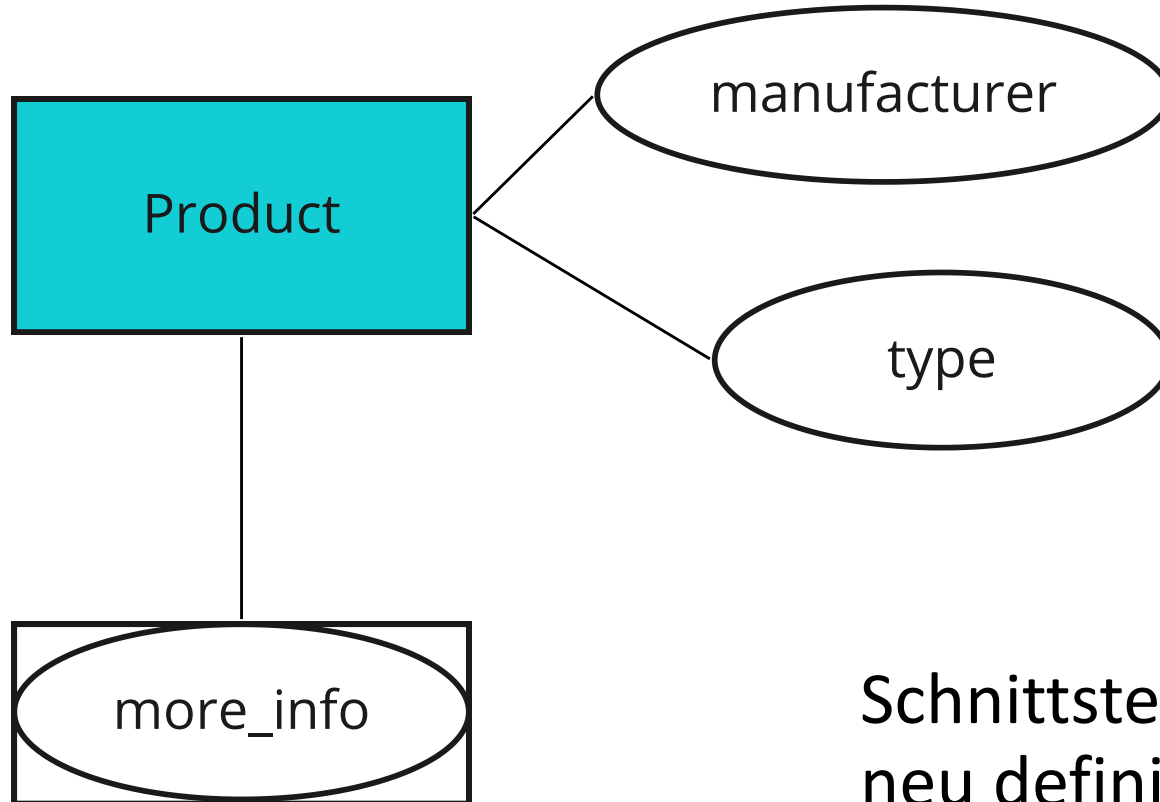
Skript 03, 04, 05



Gäbe es da noch andere Lösungen?



JSON in Attribut



Schnittstelle muss
neu definiert
werden



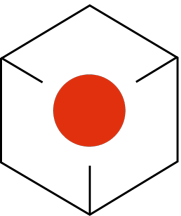
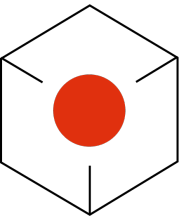


Table		
JSON		

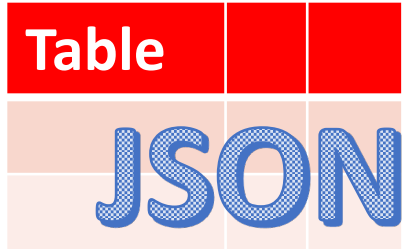
DEMO 4



Skript 06



Fazit



- Tabelle mit JSON

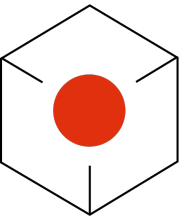
- Dokument Datenbank



- Relationale Datenbank

- Duality View





Fazit: Duality View

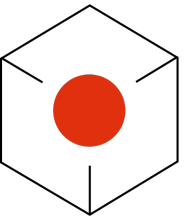
Ideal bei fixer Datenstruktur, die sowohl als JSON als auch als Tabelle geschrieben und gelesen wird.

- Relationale Datenbank

- Duality View



Quelle <https://thenounproject.com/browse/icons/term/database-table/>

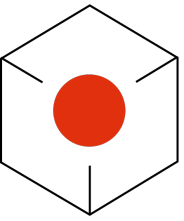


Fazit: Dokument Datenbank

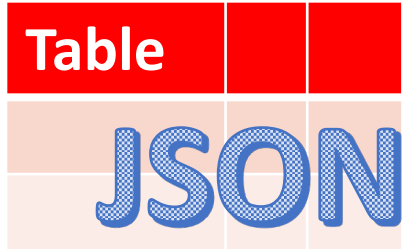
Ideal, wenn volle Flexibilität benötigt wird

- Dokument Datenbank





Fazit: Duality View mit Flexibilität



- Tabelle mit JSON

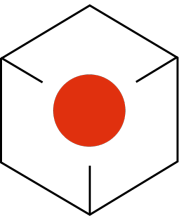
Fixe und flexible Teile können im JSON kombiniert werden.

So kann die Anforderung nach fixen und flexiblen Strukturen gelöst werden.

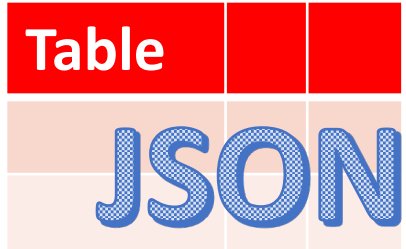
- Duality View



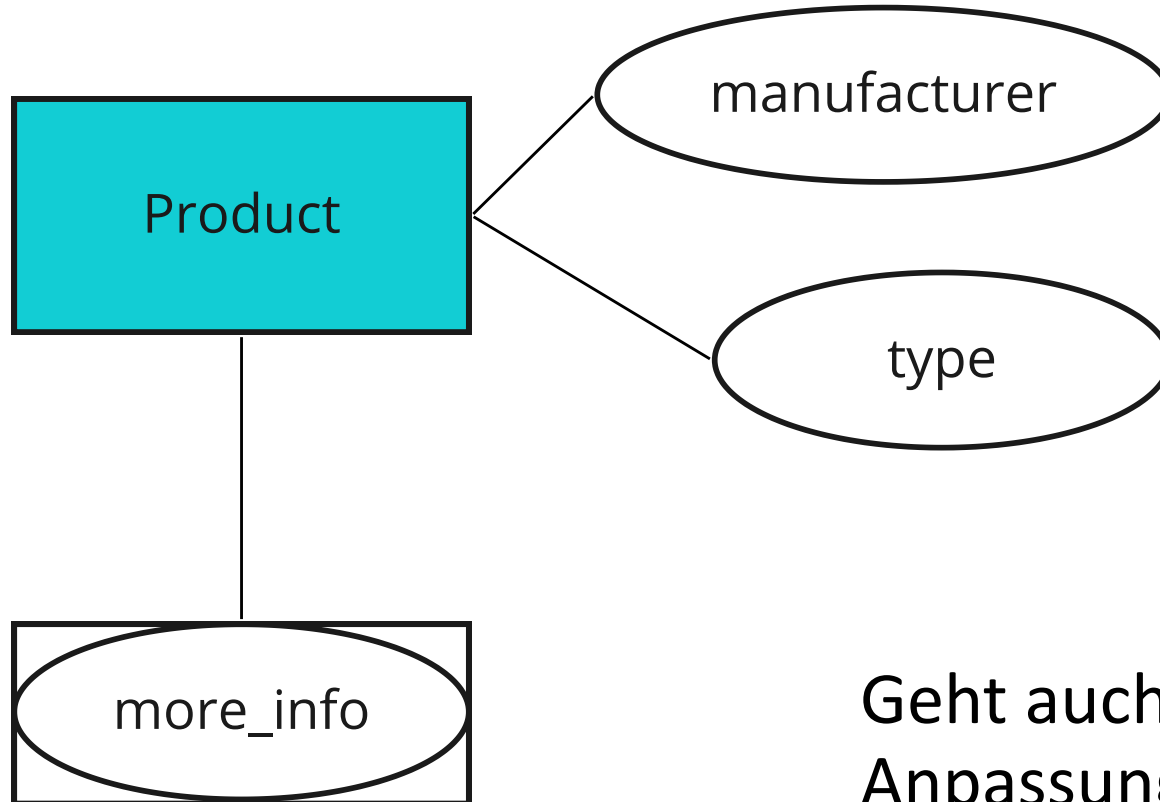
Schnittstelle muss jedoch “fixiert” werden



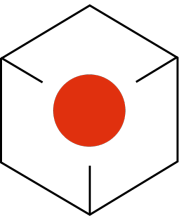
Gäbe es da noch andere Lösungen?



JSON in Attribut
und
@flex



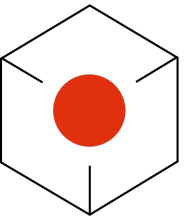
Geht auch ohne
Anpassung der
Schnittstelle



DEMO 5



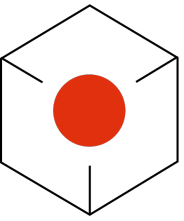
Skript 07



Wir haben da ein neues Gerät

```
{ "type": "network switch/layer3 switch",  
  "name": "Fritz C16",  
  "description": "16 port PoE layer 3 network switch",  
  "manufacturer": "ABC",  
  "price": 500,  
  "port_group": [  
    { "amount": 16,  
      "type": "RJ45",  
      "speeds": "10/100/1000",  
      "poe": {"modes": ["active", "passive"],  
              "volt": [24, 48]}  
    }  
  ],  
  ...
```

```
"feature": [  
  { "name": "VLAN",  
    "amount": 4094},  
  { "name": "QoS",  
    "amount": 8},  
  { "name": "network access control",  
    "type": "MAC based authentication",  
    "vlan_support": true},  
  { "name": "routing",  
    "protocols": "static, RIP, OSPF, BGP",  
    "table_size": 10}  
]  
}
```



Fazit: Duality View mit @flex für “Overflow”

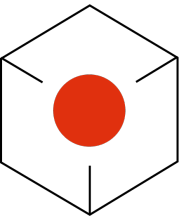
Ideal bei **beliebiger** Datenstruktur, die sowohl als JSON als auch als Tabelle geschrieben und gelesen wird.
Danke fixem und flexiblem Teil.

Ab Version 23.3

- Relationale Datenbank

- Duality View





Dr. Andrea Kennel

SYM^{L2}



Consultant

Dozentin für Datenbanken

Coach für Project Management

Fachhochschule Nordwestschweiz

Brugg/Windisch, Schweiz



andrea.kennel@fhnw.ch

andrea@infokennel.ch

www.infokennel.ch