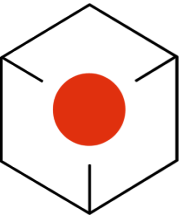


info.kennel

JSON oder relational

wie flexibel sind die Duality Views?

Dr. Andrea Kennel

InfoPunkt Kennel GmbH

Dübendorf-Schweiz

November 2024

Dr. Andrea Kennel

Consultant

Dozentin für Datenbanken

Coach für Project Management

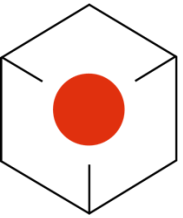
Fachhochschule Nordwestschweiz

Brugg/Windisch, Schweiz

andrea.kennel@fhnw.ch

andrea@infokennel.ch

www.infokennel.ch



Starting position

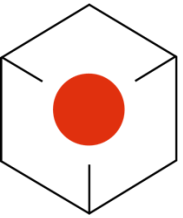
A new start-up wants to set up a **web store** for **network devices**. We are responsible for the database where the data is stored.

The web store is programmed with Java script and the data on the network devices is supplied to us as **JSON**.

The use of Duality Views is therefore obvious.

We analyze the 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"
      }
    ]
}
```

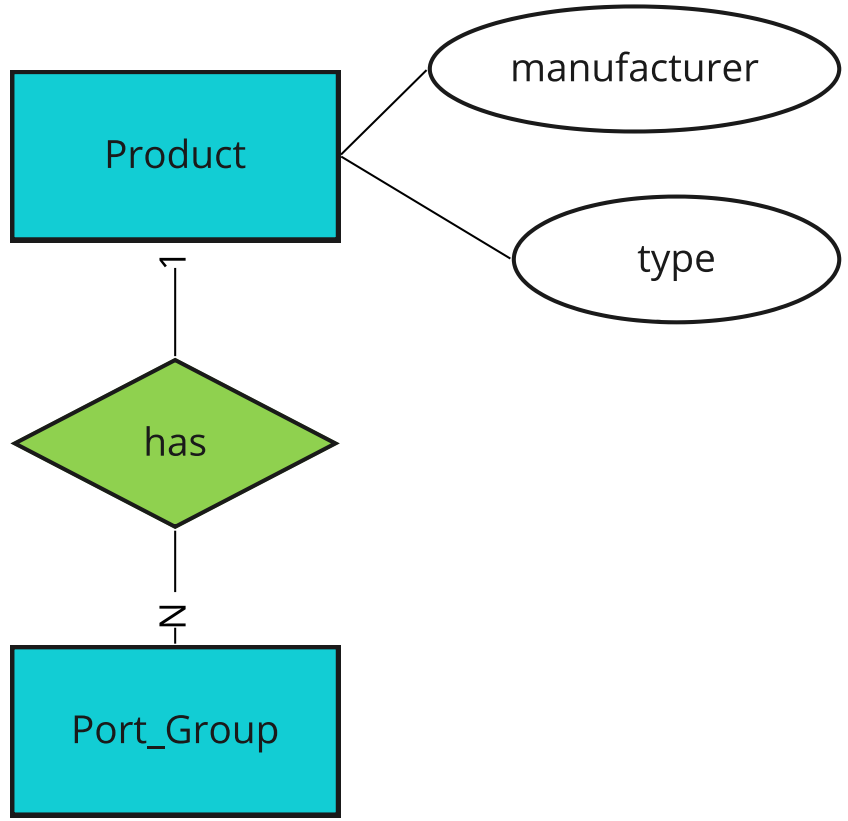
What do the devices look like?

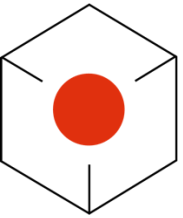
```
{    "type": "network switch",
     "name": "Fritz A16",
     "description": "unmanaged 16 port network switch",
     "manufacturer": "ABC",
     "price": 100,
     "port_group": [
        {
           "amount": 16,
           "type": "RJ45",
           "speeds": "10/100/1000"
        }
     ]
}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

The database model

For a web store that sells network devices, the data on the network devices must be stored in a database.

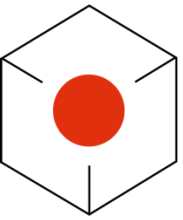
The data is available as JSON, the model Is relatively simple:





DEMO 1

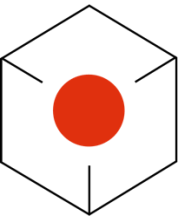
- Script 01



We have more data

Worksheet | Query Builder

```
INSERT INTO product_dv d (data)
VALUES ('
{
        "type": "network switch",
        "name": "Fritz B24",
        "description": "unmanaged 24 port 10gbit/s network switch",
        "manufacturer": "ABC",
        "price": 250,
        "port_group": [
            {
                "amount": 24,
                "type": "RJ45",
                "speeds": "10/100/1000/2500/5000/10000"
            }
        ]
    }');
```



We have more data

Worksheet | Query Builder

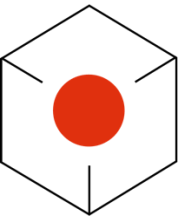
```
        "price": 1000,
        "port_group": [
            {   "amount": 4,
                "type": "RJ45",
                "speeds": "100/1000/10000"
            },
            {   "amount": 2,
                "type": "SFP+",
                "speeds": "1000/40000/100000"
            },
            {   "amount": 2,
                "type": "SFP",
                "speeds": "1000/10000"
            } ],
        "routing": {
            "protocols": "static, RIP, OSPF, BGP",
            "table_size": 10
        }
    }');
```

Script Output | Query Result

Task completed in 0.135 seconds

```
Error report -
ORA-40944: Cannot insert into JSON Relational Duality View 'PRODUCT_DV': The input JSON
JZN-00651: field 'routing' is unknown or undefined
```

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

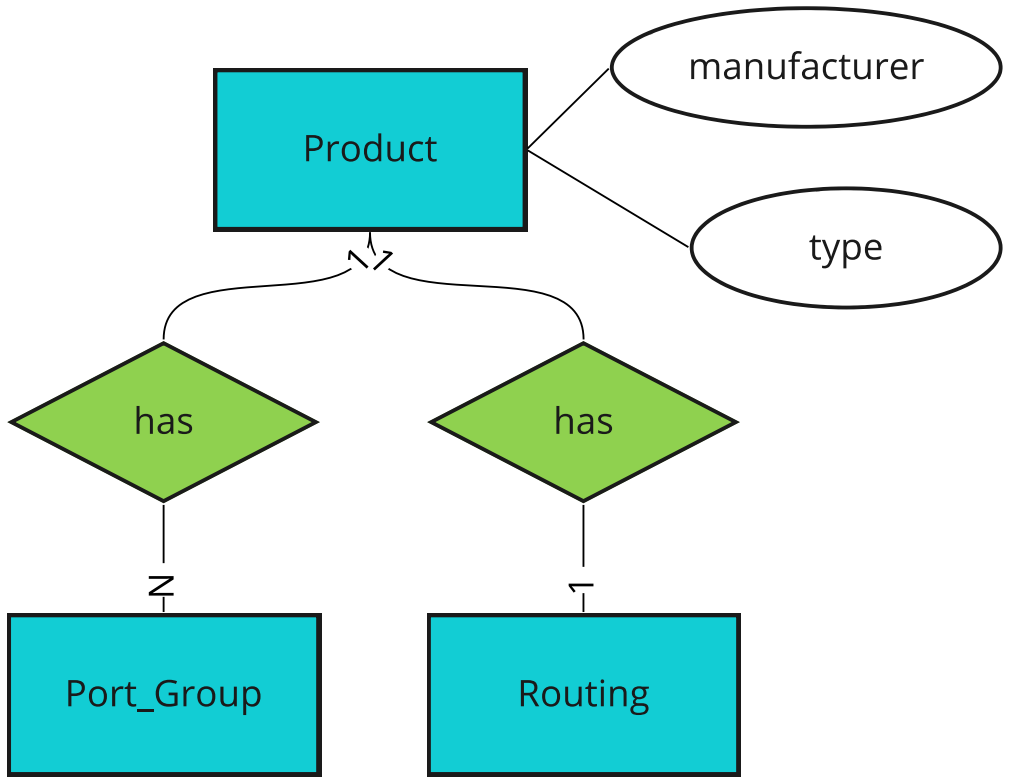


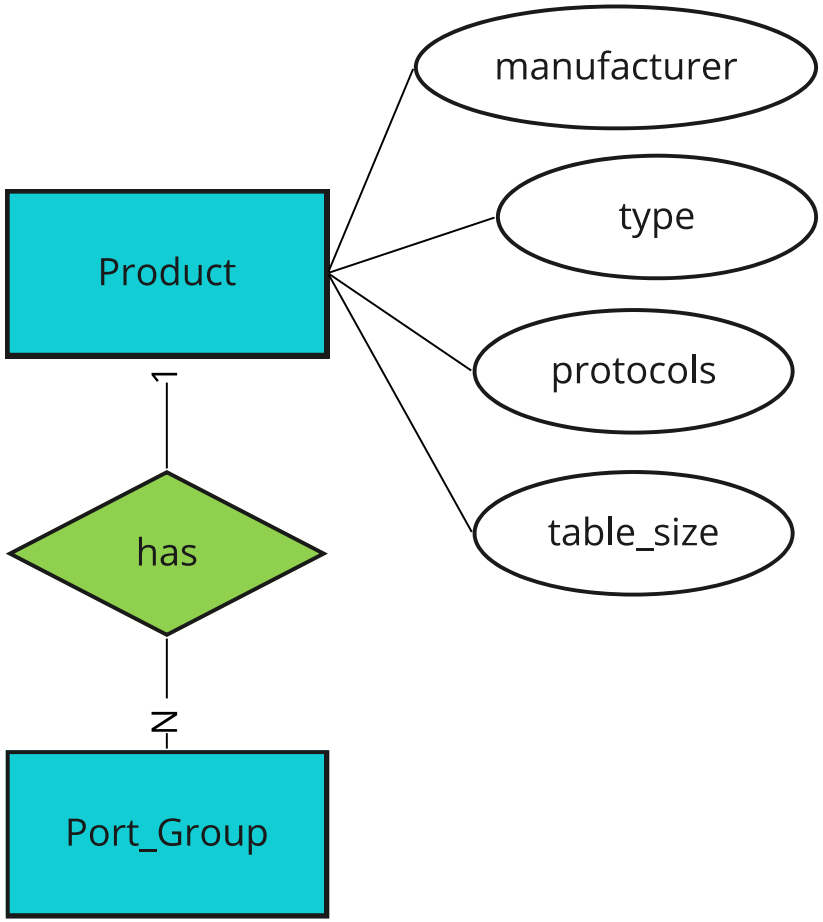
We have a new device


```
{    "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
     }
}
```

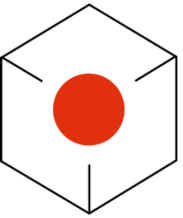
Extend data model

We also need a routing:
2 possible solutions





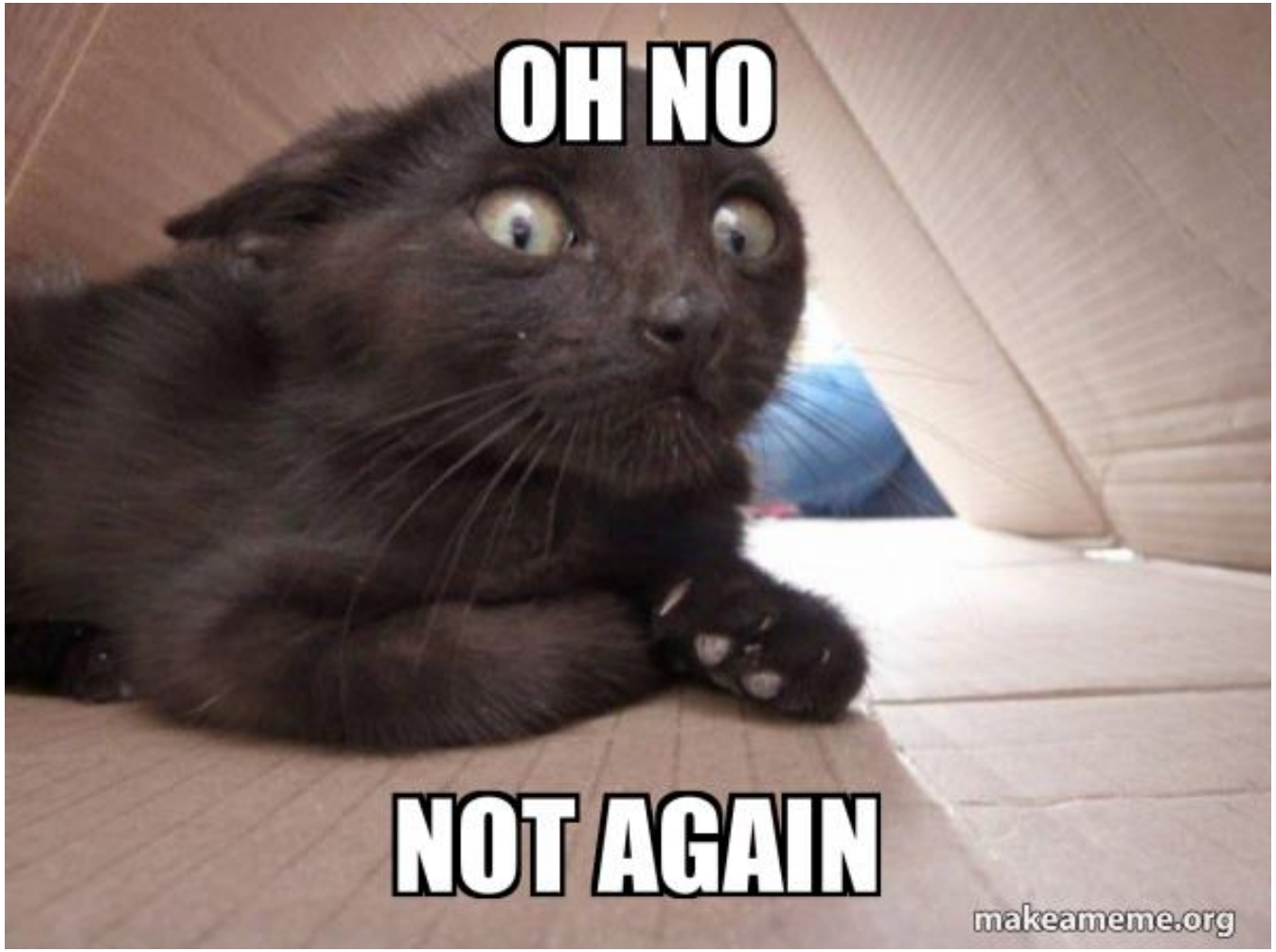
Additional attributes are easier



DEMO 2

- Script 02

We have another new device



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

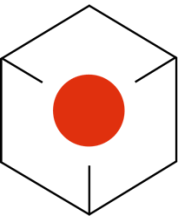
We have another new device


```
{   "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]}
        }
      ],
…
```

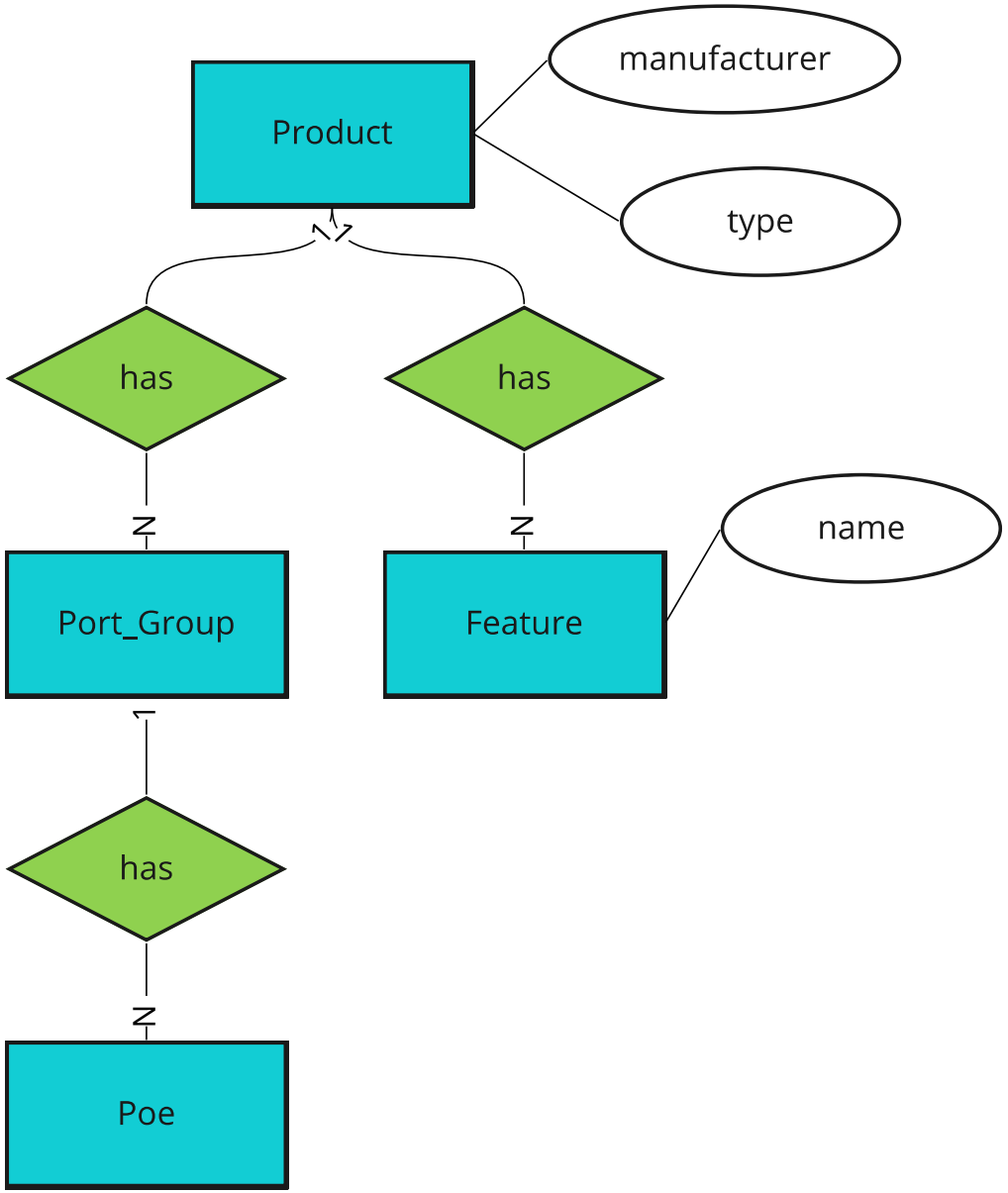
```
 "feature": [
        {   "name": "VLAN",
            "amount": 4094},
        {    "name": "QoS",
            "amount": 8},
        {   "name": "network access control",
            "type": "MAC based authentication",
            "vlan_support": true},
        {   "name": "routing",
           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

We then have to adapt the data structure

- Detail table for port group and
- Feature as generalization with several specializations
- We don't know what else is coming and summarize all attributes in the generalization.
- The Duality View must be expanded accordingly



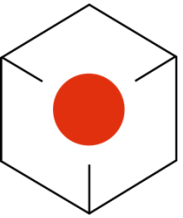
New data structure



- Does the tool fit the problem?

- Duality Views
- The same data in JSON and in tables

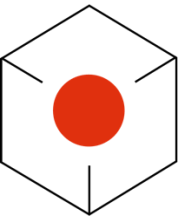
- JSON as a flexible data structure



Would a document DB not be better?

- Why not simply store all data as JSON?

- Then we have full flexibility

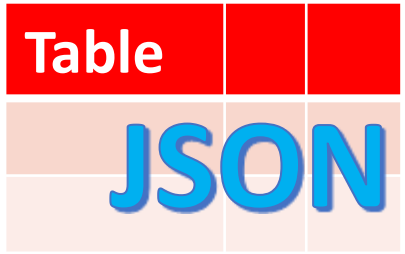


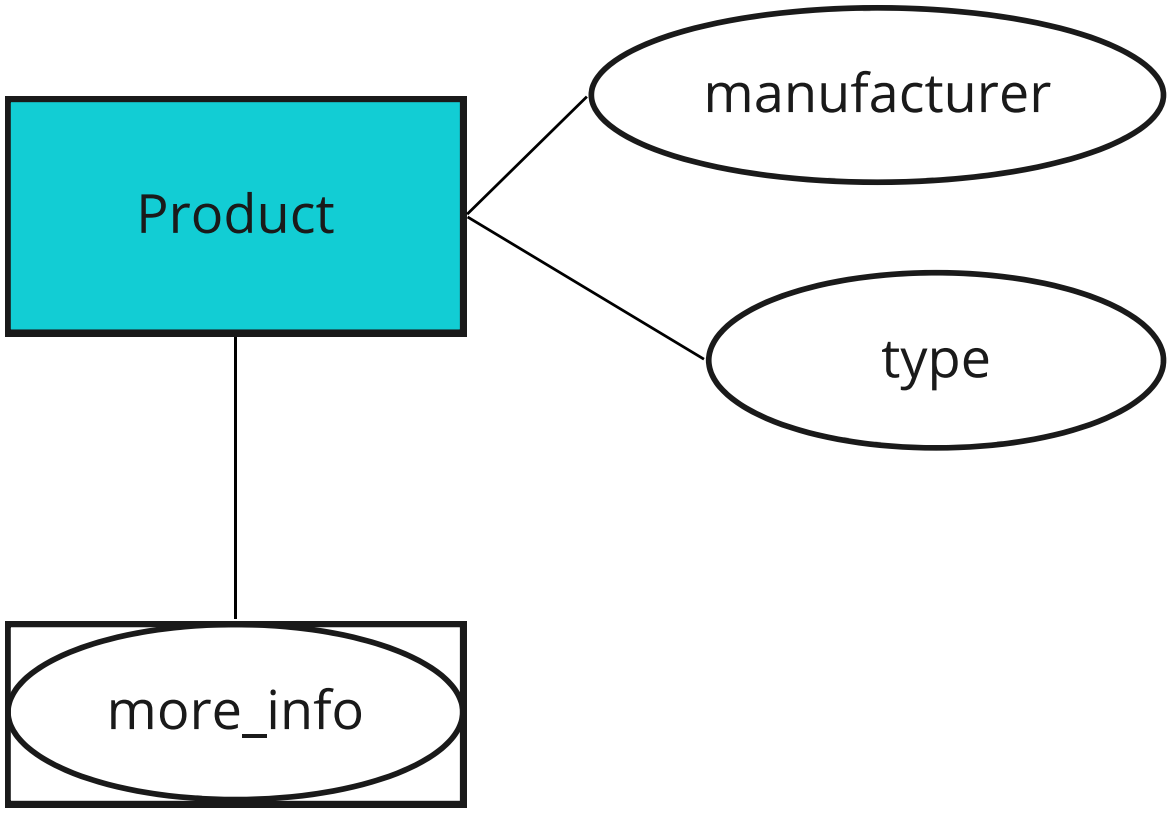
DEMO 3

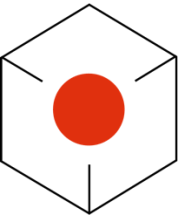
Problem,
if JSON has a
spelling mistake

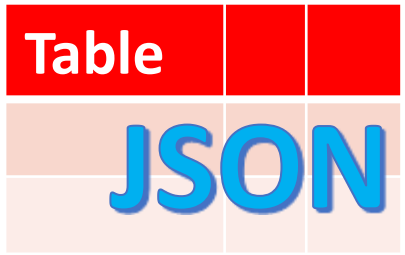
- Script 03, 04, 05

Are there any other solutions?









DEMO 4

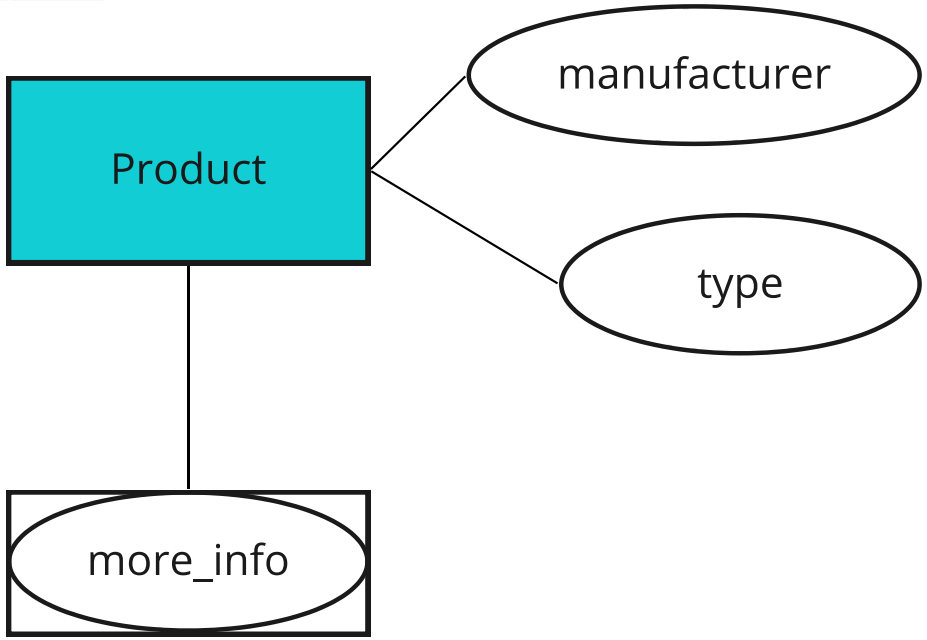
Problem: Interface must be redefined

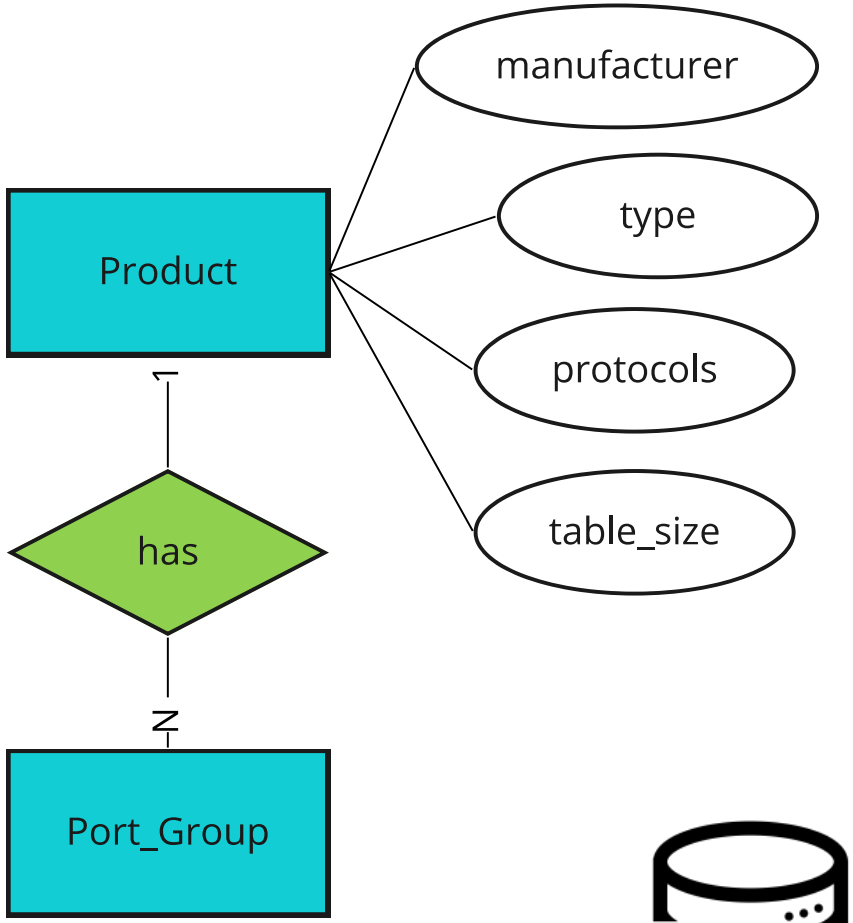
Script 06

I wish I could combine the two worlds

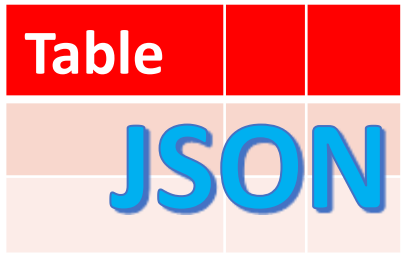
Table

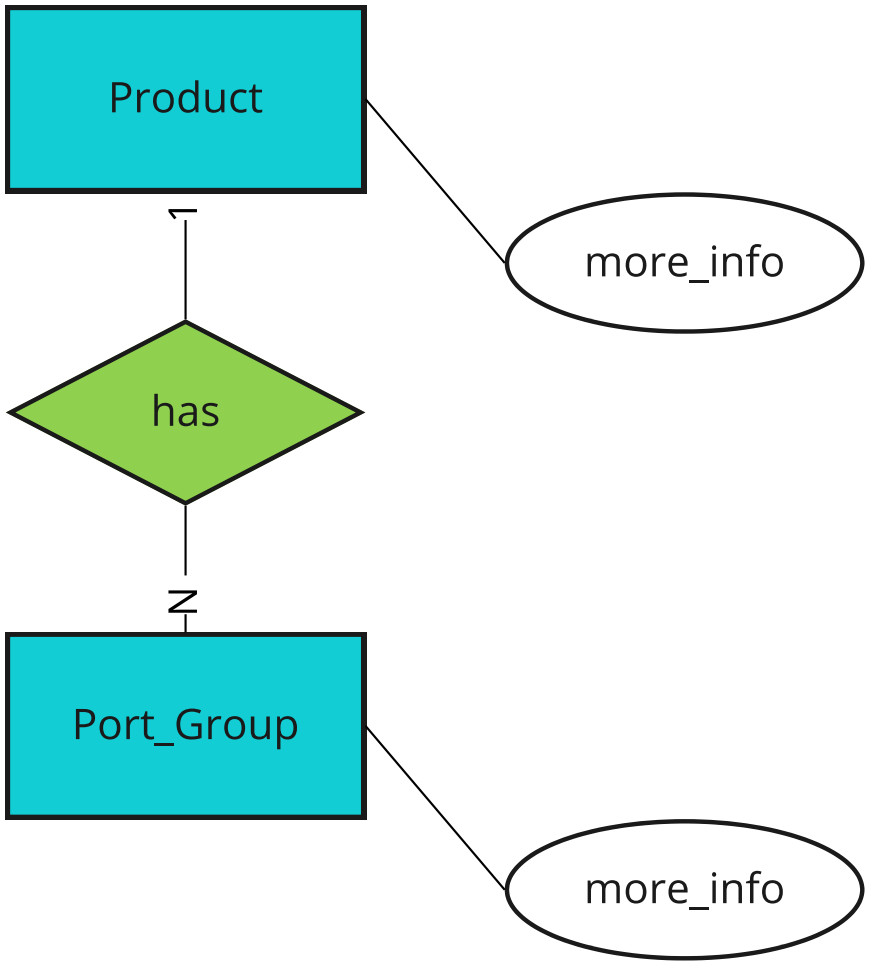
JSON



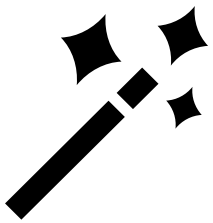


I wish I could combine the two worlds





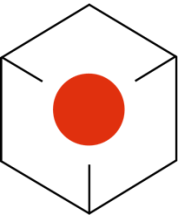
I wish that additional attributes were automatically saved in the correct table as a JSON attribute.



We have another new device


```
{   "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}
      ]
   }
```



Table

JSON

DEMO 5

Script 07

Best solution:
Duality View with @flex for “Overflow”

Ideal for any data structure that is written and read both as JSON
and as a table.
Thanks to a fixed and flexible part.

From version 23.3
And in OCI

Dr. Andrea Kennel

Consultant

Dozentin für Datenbanken

Coach für Project Management

Fachhochschule Nordwestschweiz

Brugg/Windisch, Schweiz

andrea.kennel@fhnw.ch

andrea@infokennel.ch

www.infokennel.ch