info.kennel

Dynamische Attribute zur Speicherung von flexiblen Daten

Dr. Andrea Kennel

InfoPunkt Kennel GmbH

Dübendorf-Schweiz

November 2023

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

Ausgangslage

Ein neues Start-Up will einen **Web-Shop** für **Netzwerkgeräte** aufbauen. Die Daten zu diesen Geräten sollen in einer Datenbank verwaltet werden.

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

User Job ist das Erstellen der Datenbank.

```
{    "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
     }
  }
```

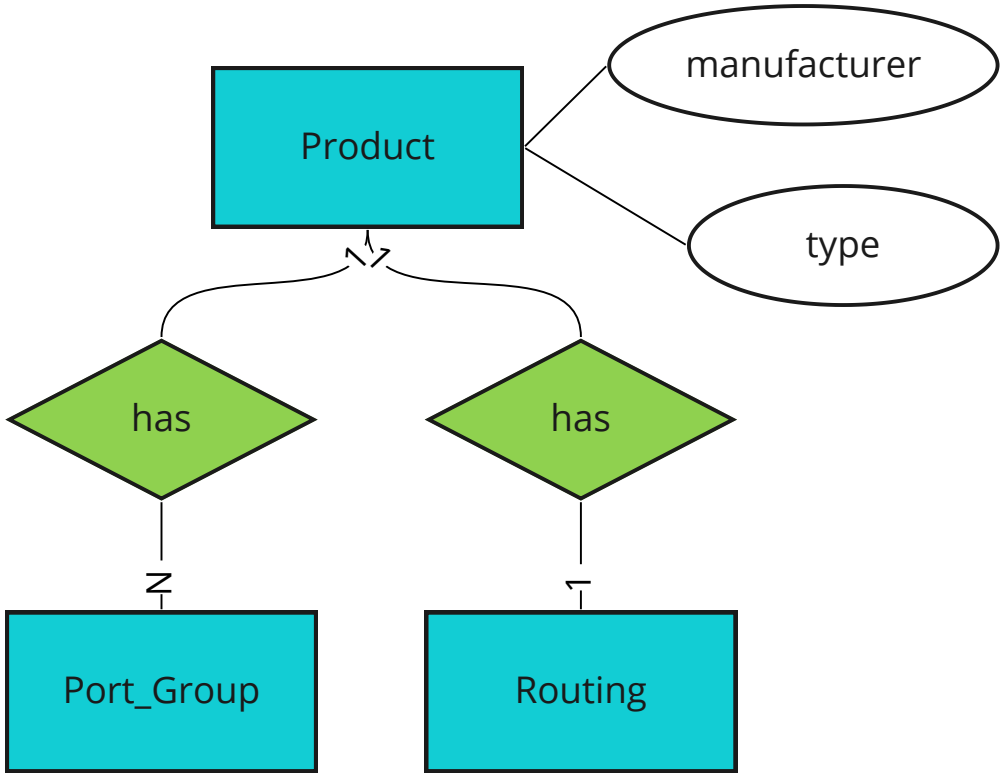

Wie sehen die Geräte aus?

```
{    "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
     }
}
```

Kannst Du mal schnell

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

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



Datenmodell umsetzten und Daten erfassen

Welcome Page | Docker_loacal_ANDREA | 1.972 seconds

Worksheet | Query Builder

```
CREATE TABLE product (
  prod_id       NUMBER(10),
  type          VARCHAR2(50),
  name          VARCHAR2(50),
  description   VARCHAR2(200),
  manufacturer  VARCHAR2(50),
  price         NUMBER (8,2)
);

CREATE TABLE port_group (
```

Query Result | Script Output | Task completed in 1.972 seconds

```
Table PRODUCT created.

Table PORT_GROUP created.

Table ROUTING created.
```

Welcome Page | Docker_loacal_ANDREA | 0.597 seconds | Docker_loacal_ANDREA

Worksheet | Query Builder

```
INSERT INTO product (prod_id, type, name, description, manufacturer
VALUES (1, 'network switch', 'Fritz A16', 'unmanaged 16 port networ
       (2, 'network switch', 'Fritz B24', 'unmanaged 24 port 10gbit
       (3, 'network switch', 'Fritz C12', 'managed 12 port network
       (4, 'router', 'Hans A8', '8 port router', 'ABC', 1000),
       (5, 'router', 'Hans B4', '4 port router', 'XYZ', 800);

INSERT INTO port_group (pogr_id, prod_id, amount, type, speeds)...

INSERT INTO routing (rout_id, prod_id, protocols, table_size)
```

Query Result | Script Output | Task completed in 0.597 seconds

```
5 rows inserted.

8 rows inserted.

2 rows inserted.
```

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}
    ]
}
```

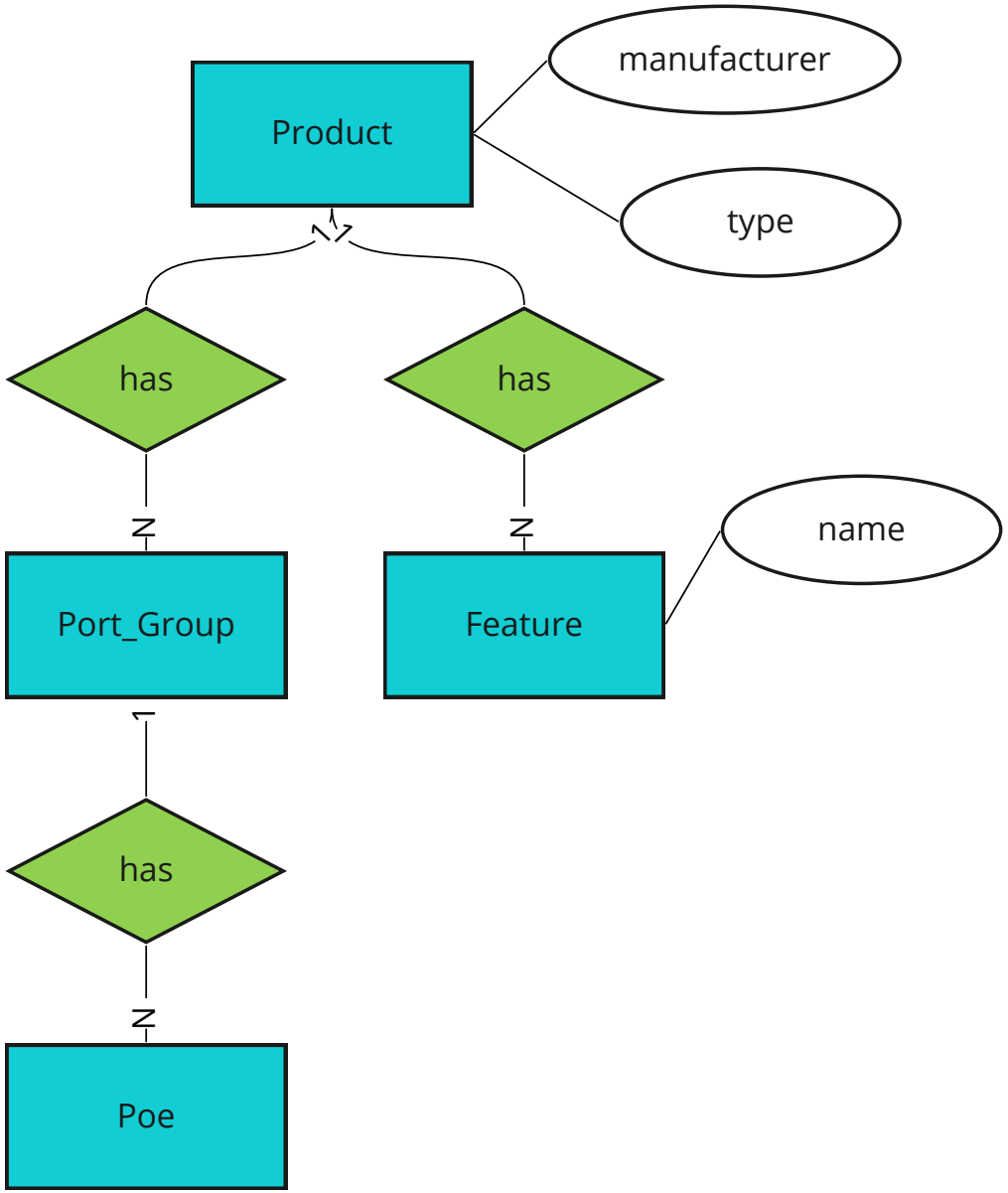
Dann müssen wir die Datenstruktur anpassen

Detail-Tabelle zu Port-Group und

Feature als Generalisierung mit mehreren Spezialisierungen

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

Neue Datenstruktur



Neue Datenstruktur

Welcome Page | Docker_loacal_ANDREA

0.55500001 seconds | Docker_loacal_ANDREA

Worksheet | Query Builder

```
CREATE TABLE poe (
  poe_id        NUMBER(10),
  pogr_id       NUMBER(10),
  modes         VARCHAR2(20),
  volt          NUMBER (6,2)
);

ALTER TABLE poe ADD CONSTRAINT poe_pk PRIMARY KEY (poe_id);
ALTER TABLE poe ADD CONSTRAINT poe_pogr_fk FOREIGN KEY (pogr_id)
REFERENCES port_group (pogr_id);
```

Query Result | Script Output

Task completed in 0.555 seconds

```
Table POE created.

Table POE altered.

Table POE altered.
```

Datenstruktur

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

1.83200002 seconds

Worksheet | Query Builder

```sql
RENAME routing TO feature;
ALTER TABLE feature RENAME COLUMN  rout_id TO feat_id;
ALTER TABLE feature ADD name VARCHAR2(200);
ALTER TABLE feature ADD amount NUMBER(10);
ALTER TABLE feature ADD type VARCHAR2(200);
ALTER TABLE feature ADD vlan_support VARCHAR2(200);
-- all existing get name 'routing'
UPDATE feature
SET name = 'routing';
```

Query Result | Script Output

Task completed in 1.832 seconds

```
Table FEATURE altered.


Table FEATURE altered.


2 rows updated.
```

Neue Daten

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

0.52700001 seconds | Docker_loacal_ANDREA

Worksheet | Query Builder

```
INSERT INTO product (prod_id, type, name, description, manufacturer, price)
VALUES (21, 'network switch/layer3 switch', 'Fritz C16', '16 port PoE layer 3 netw

INSERT INTO port_group (pogr_id, prod_id, amount, type, speeds)
VALUES (21, 21, 16, 'RJ45', '10/100/1000');

INSERT INTO poe (poe_id, pogr_id, modes, volt)
VALUES (21, 21, 'active', 24),
       (22, 21, 'passive', 48);

INSERT INTO feature (feat_id, prod_id, name, amount)
VALUES (21, 21, 'VLAN', 4094),
       (22, 21, 'QoS', 8);

INSERT INTO feature (feat_id, prod_id, name, type, vlan_support)
VALUES (23, 21, 'network access control', 'MAC based authentication', 'TRUE');

INSERT INTO feature (feat_id, prod_id, name, protocols, table_size)
```

Was, wenn es mehr Attribute gibt?

Wir mussten bei FEATURES weitere Attribute zufügen.

Was passiert, wenn da noch weitere Attibute kommen?

Was wenn viele features dann nur wenige Attribute gefüllt haben?

Gibt es da eine flexiblere Lösung?


```
"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}
    ]
}
```


EAV

Was, wenn es mehr Attribute gibt?

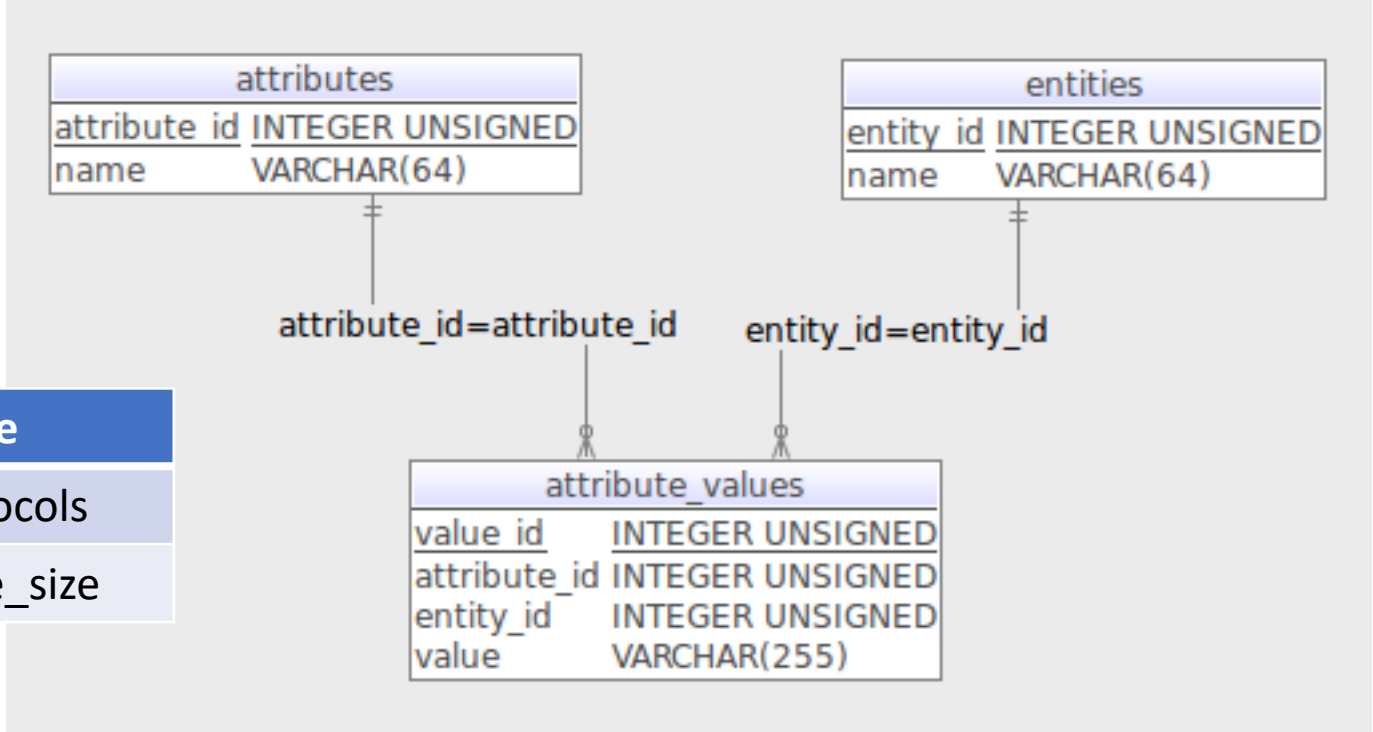
Die Antwort heisst EAV

Das Entity-Attribute-Value-Modell (EAV) ist eine Datenmodellierungstechnik, die in Datenbanken verwendet wird, um Daten auf flexible und skalierbare Weise zu speichern und abzurufen.

Details zum Modell: https://inviqa.com/blog/understanding-eav-data-model-and-when-use-it

EAV

EAV



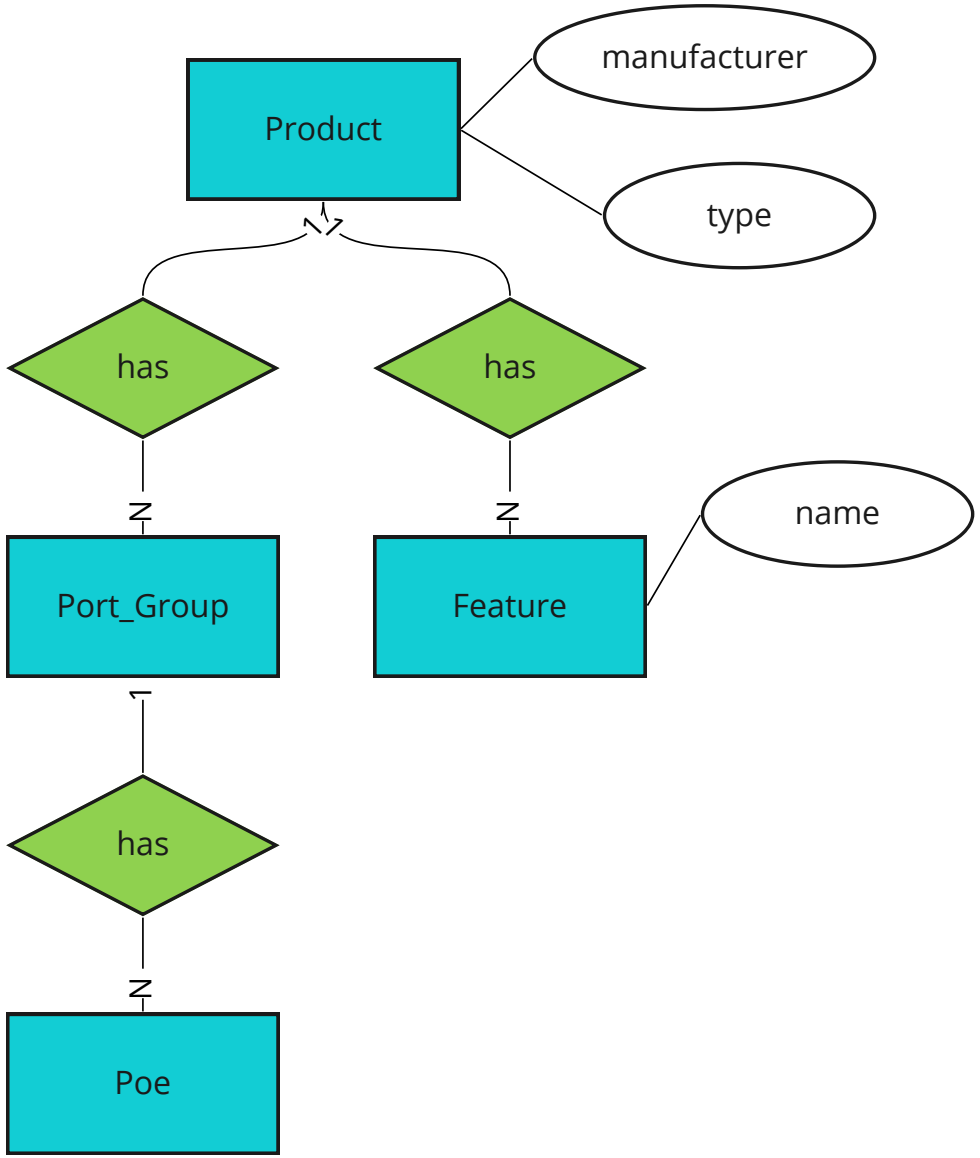
Attribute_id	name
84	protocols
85	table_size

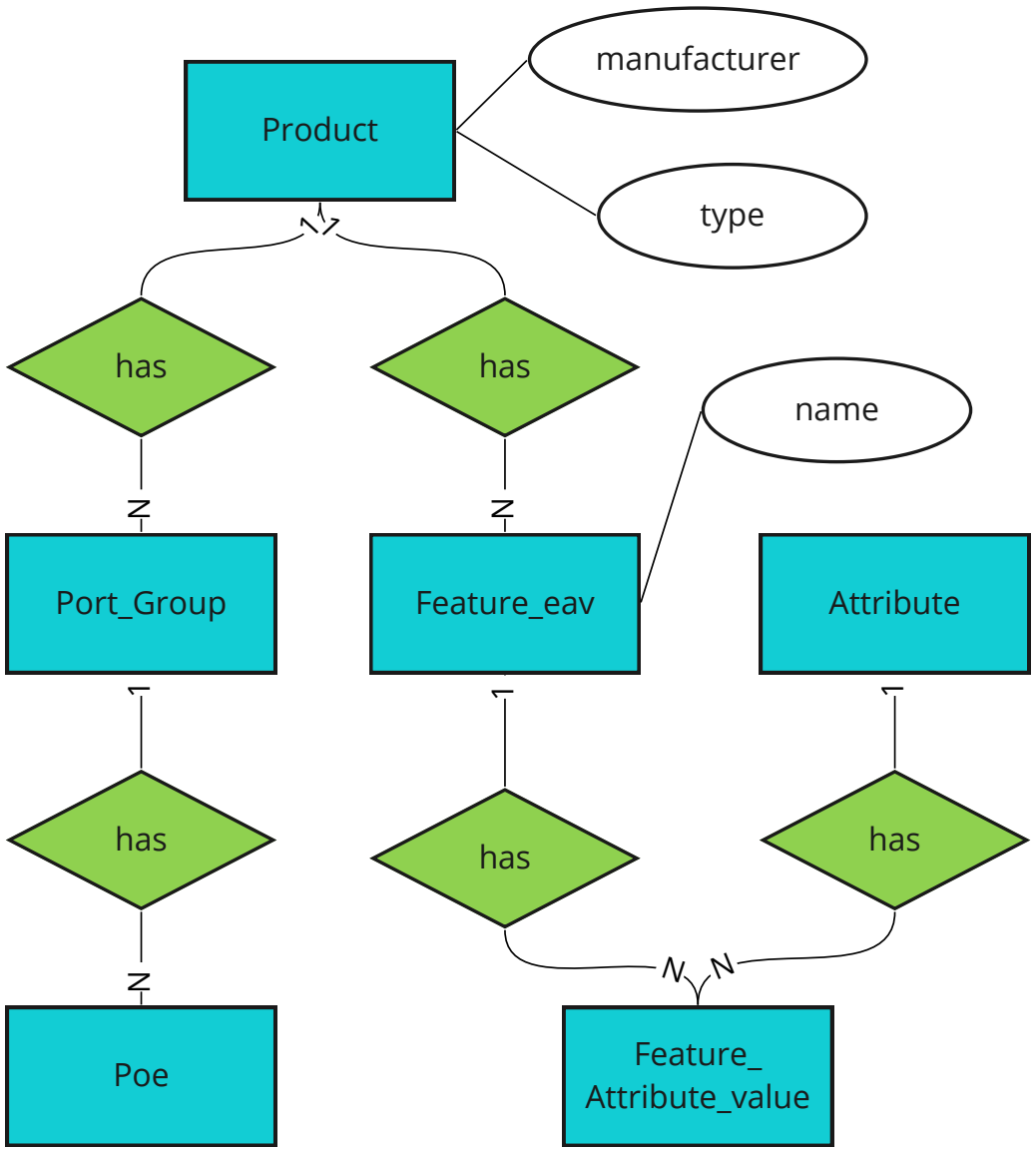
```
INSERT INTO feature (feat_id, prod_id, name, protocols, table_size)
VALUES (24, 21, 'routing', 'static, RIP, OSPF, BGP', 10);
```

Value_id	Attribute_id	Entity_id	values
1	84	24	static, RIP, OSPF, BGP
2	85	24	10

EAV

EAV


Product
manufacturer
type
1:1
has
has
N
N
Port_Group
Feature
name
1
has
N
Poe


Product
manufacturer
type
1:1
has
has
name
N
N
Port_Group
Feature_eav
Attribute
1
1
1
has
has
has
N
N:N
Poe
Feature_
Attribute_value

EAV

EAV

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_AN

Worksheet | Query Builder

```oracle_db_lang
CREATE TABLE feature_eav (
  feat_id       NUMBER(10),
  prod_id       NUMBER(10),
  name          VARCHAR2(200)
);

CREATE TABLE attribute (
  attr_id       NUMBER(10),
  name          VARCHAR2(30)
);

CREATE TABLE feature_attribute_value (
  valu_id       NUMBER(10),
  feat_id       NUMBER(10),
  attr_id       NUMBER(10),
  value         VARCHAR2(2000)
);
```

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

Docker_loacal_A

Worksheet | Query Builder

```oracle_db_lang
INSERT INTO feature_eav (feat_id, prod_id, name)
VALUES (24, 21, 'routing');

INSERT INTO attribute (attr_id, name)
VALUES (84, 'protocols'),
       (85, 'table_size');

INSERT INTO feature_attribute_value (valu_id, feat_id, attr_id, value)
VALUES (25, 24, 84, 'static, RIP, OSPF, BGP'),
       (26, 24, 85, '10');
```

EAV

Wäre da nicht eine Dokument DB besser?

- Ich verliere den Überblick
- Warum nicht einfach alle Daten als JSON speichern?

Wäre da nicht eine Dokument DB besser?

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

Worksheet | Query Builder

```
CREATE TABLE product_collection (
  prod_id       NUMBER(10),
  json_data     JSON,
  CONSTRAINT pro_ensure_json CHECK (json_data IS JSON)
);
```

Script Output | Query Result

Task completed in 0.698 seconds

```
Table PRODUCT_COLLECTION created.
```

Wäre da nicht eine Dokument DB besser?

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

Worksheet | Query Builder

```
INSERT INTO product_collection (prod_id, json_d
VALUES (1, json('{
        "type": "network switch",
        "name": "Fritz A16",
        "description": "unmanaged 16 port netwo
        "manufacturer": "ABC",
        "price": 100,
        "port_group": [
            {
                "amount": 16,
                "type": "RJ45",
                "speeds": "10/100/1000"
            }
        ]
    }'));
```

Welcome Page | Docker_loacal_ANDREA | edbsHS2

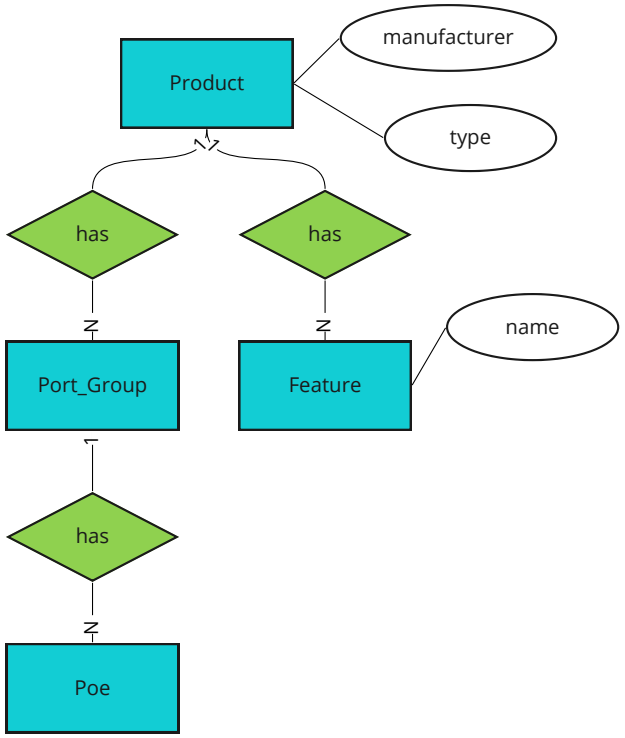
Worksheet | Query Builder

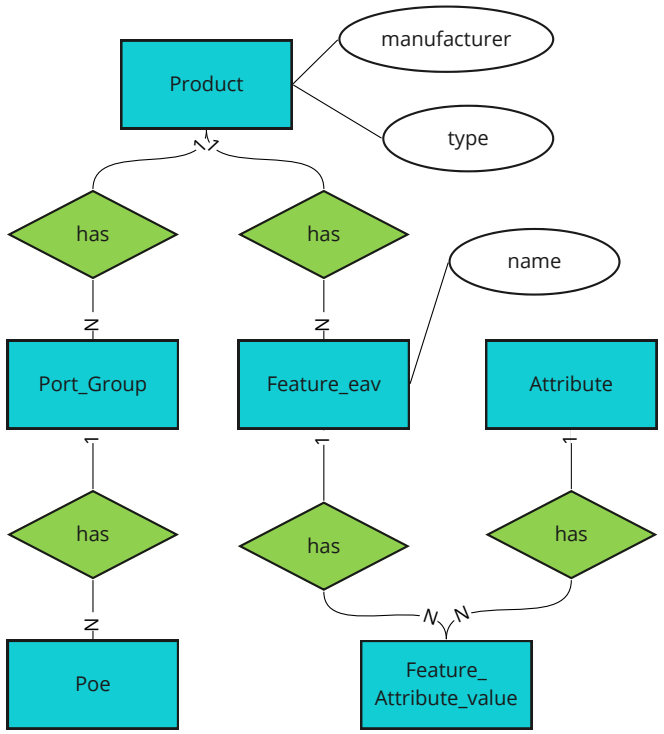
```
INSERT INTO product_collection (prod_id, json_data)
VALUES (21, json('{
        "type": "network switch/layer3 switch",
        "name": "Fritz C16",
        "description": "16 port PoE layer 3 network switch",
        "manufacturer": "ABC",
        "price": 500,
        "port_group": [
            {
                "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
            }
        ]
    }'));
```

Wäre da nicht eine Dokument DB besser?

Wie sieht das mit Abfragen aus?

Beipiel: Haben wir Geräte mit einer Port-Group vom Typ SFP?





Haben wir Geräte mit einer Port-Group vom Typ SFP? relational

```
SELECT pro.prod_id, pro.type product_type, pro.name, pog.amount, pog.type
FROM product pro INNER JOIN
    port_group pog ON (pro.prod_id = pog.prod_id)
WHERE pog.type = 'SFP’;
```

Haben wir Geräte mit einer Port-Group vom Typ SFP? JSON

```
SELECT pc.prod_id, pc.json_data.type product_type,
pc.json_data.name,
    pc.json_data.port_group[*].amount,
    pc.json_data.port_group[*].type
FROM product_collection pc
WHERE json_exists(pc.json_data,
            '$.port_group?(@.type == $v1)'
             PASSING 'SFP' AS "v1");
```

```
SELECT pc.prod_id, pc.json_data.type product_type,
pc.json_data.name,
    jt.*
FROM product_collection pc,
   JSON_TABLE(
      pc.json_data
      COLUMNS (
         NESTED port_group[*]
         COLUMNS (
            amount NUMBER(3) PATH '$.amount',
            type VARCHAR2(50) PATH '$.type'
         )
      )
   ) jt
WHERE jt.type = 'SFP';
```

Haben wir Geräte mit Feature, das amount = 8 hat? relational

```
SELECT pro.prod_id, pro.type product_type, pro.name,
      fe.name feature_name, fe.amount
FROM product pro INNER JOIN
    feature fe ON (pro.prod_id = fe.prod_id)
WHERE fe.amount = 8;
```

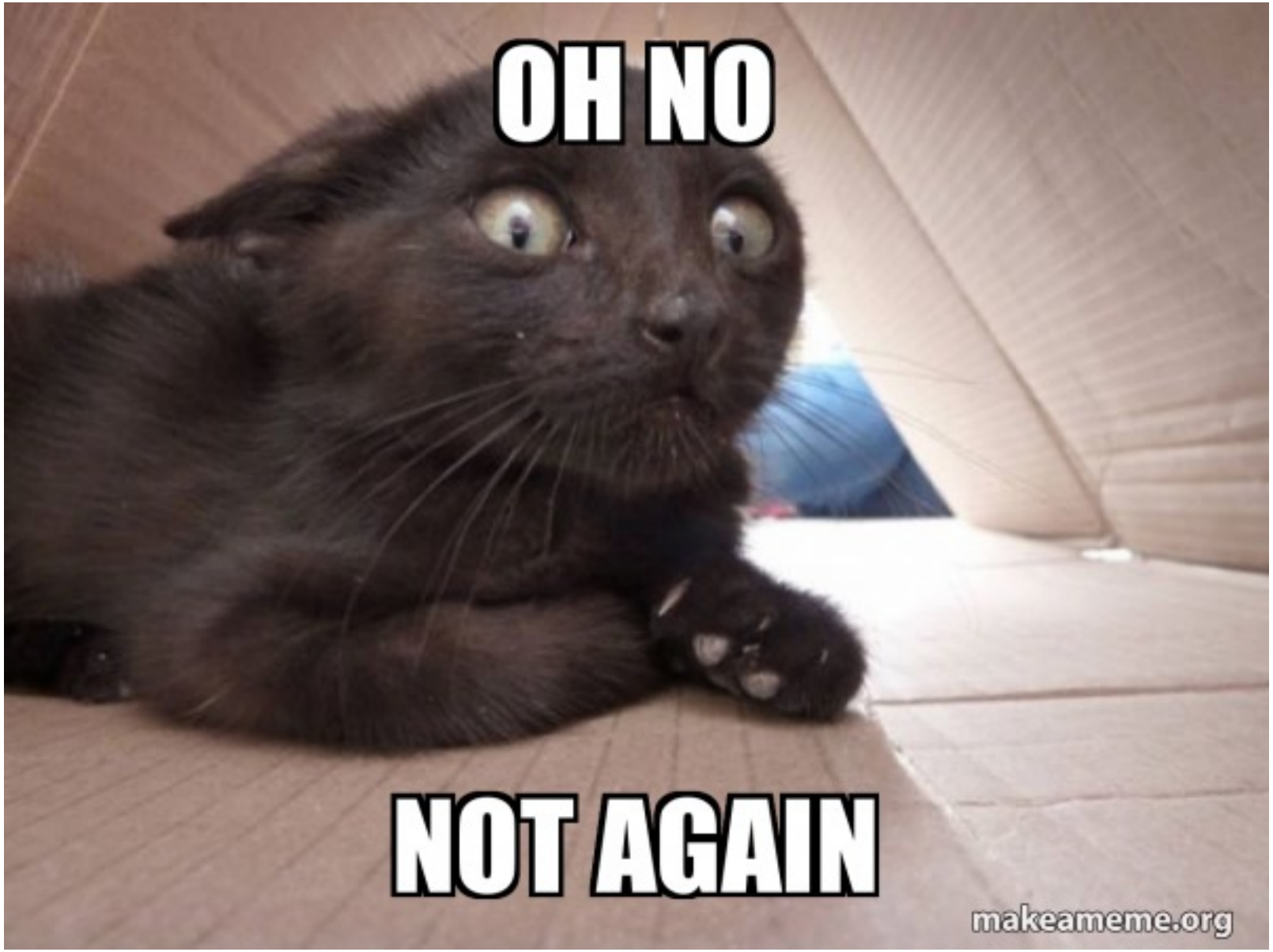
Haben wir Geräte mit Feature, das amount = 8 hat? JSON

```
SELECT pc.prod_id, pc.json_data.type product_type, pc.json_data.name,
      jt.*
FROM product_collection pc,
    JSON_TABLE(
        pc.json_data
        COLUMNS (
            NESTED feature[*]
            COLUMNS (
                feature_name VARCHAR2(50) PATH '$.name',
                amount NUMBER(10) PATH '$.amount'
            )
        )
    ) jt
WHERE  jt.amount = 8;
```

Haben wir Geräte mit Feature, das amount = 8 hat? EAV

```
SELECT pro.prod_id, pro.type product_type, pro.name,
      fe.name feature_name, feat.value amount
FROM product pro INNER JOIN
    feature_eav fe ON (pro.prod_id = fe.prod_id) INNER JOIN
    feature_attribute_value feat ON (fe.feat_id = feat.feat_id) INNER JOIN
    attribute at ON (feat.attr_id = at.attr_id)
WHERE at.name = 'amount'
  AND feat.value = '8';
```

Wir haben wie



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

Wir haben wieder ein neues Gerät


```
{   "type": "router/firewall",
    "name": "Hans A48",
....
"feature": [
        {   … },
        {   … },
        {   … },
        {   "name": "firewall",
            "type": "stateful, stateless",
            "amount_of_rules": 1000,
            "amount_of_connections": 100000,
            "amount_of_nat_rules": 1000,
            "feature": [
                {   "name": "URL filtering",
                    "types": "whitelist, blacklist",
                    "amount_of_rules": 1000 },
....
                {   "name": "application filtering",
                    "types": "whitelist",
                    "amount_of_rules": 1000 },
                {   "name": "content filtering",
                    "types": "blacklist",
                    "amount_of_rules": 1000 },
                {   "name": "anti-virus",
                    "ssl_inspection": true }
            ]
        },
        {   "name": "VPN",
            "type": "IPsec, SSL, L2TP, PPTP",
            "amount_of_tunnels": 1000,
            "amount_of_users": 1000,
            "feature": [
                {   "name": "IPsec",
                    "type": "IKEv1, IKEv2",
                    "amount_of_tunnels": 1000,
                    "amount_of_users": 1000 },
                {   "name": "SSL",
…
```

Dann müssen wir die Datenstruktur anpassen

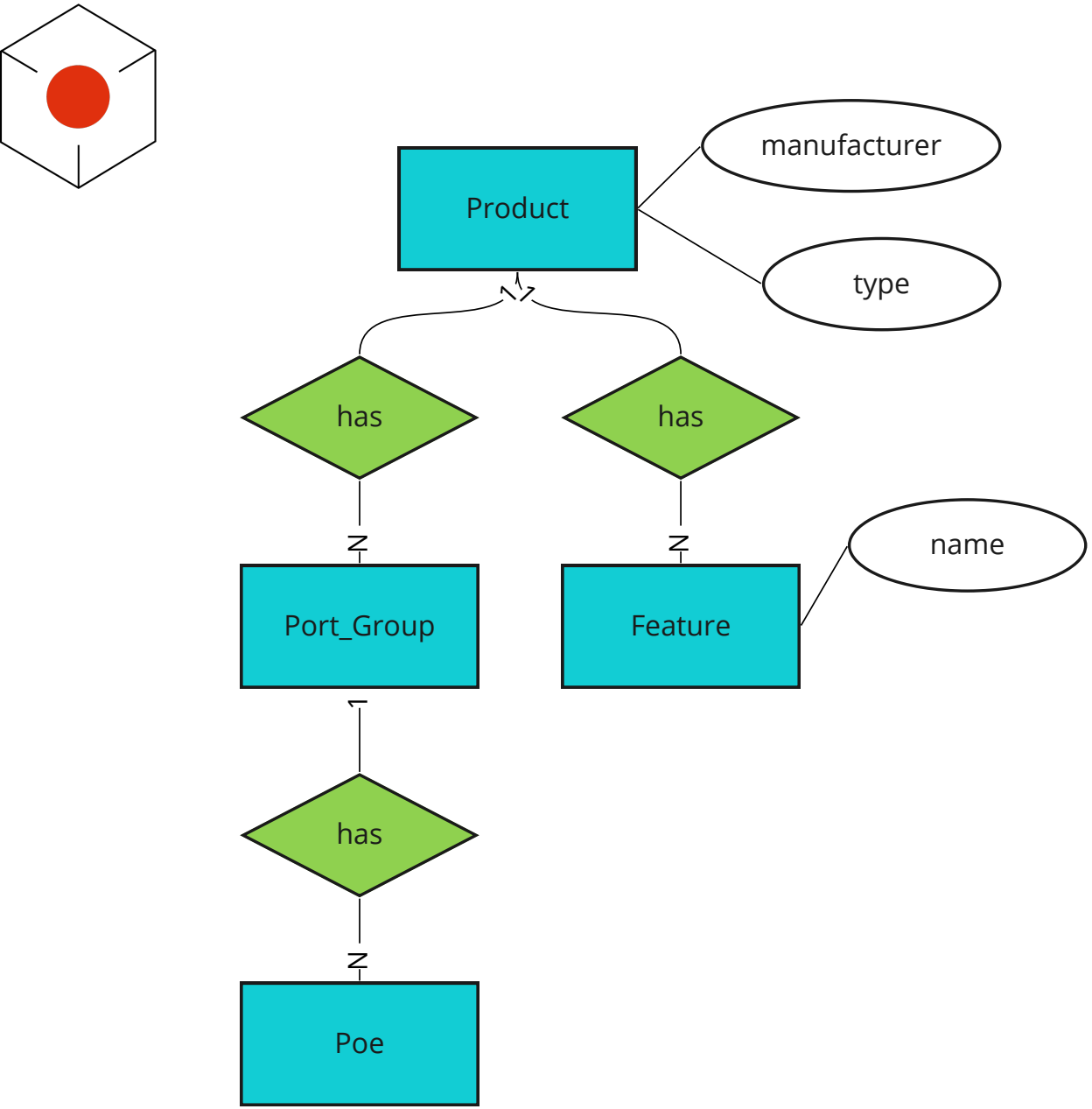
Weiter Typen von Features mit neuen Attributen, damit haben wir ja schon gerechnet

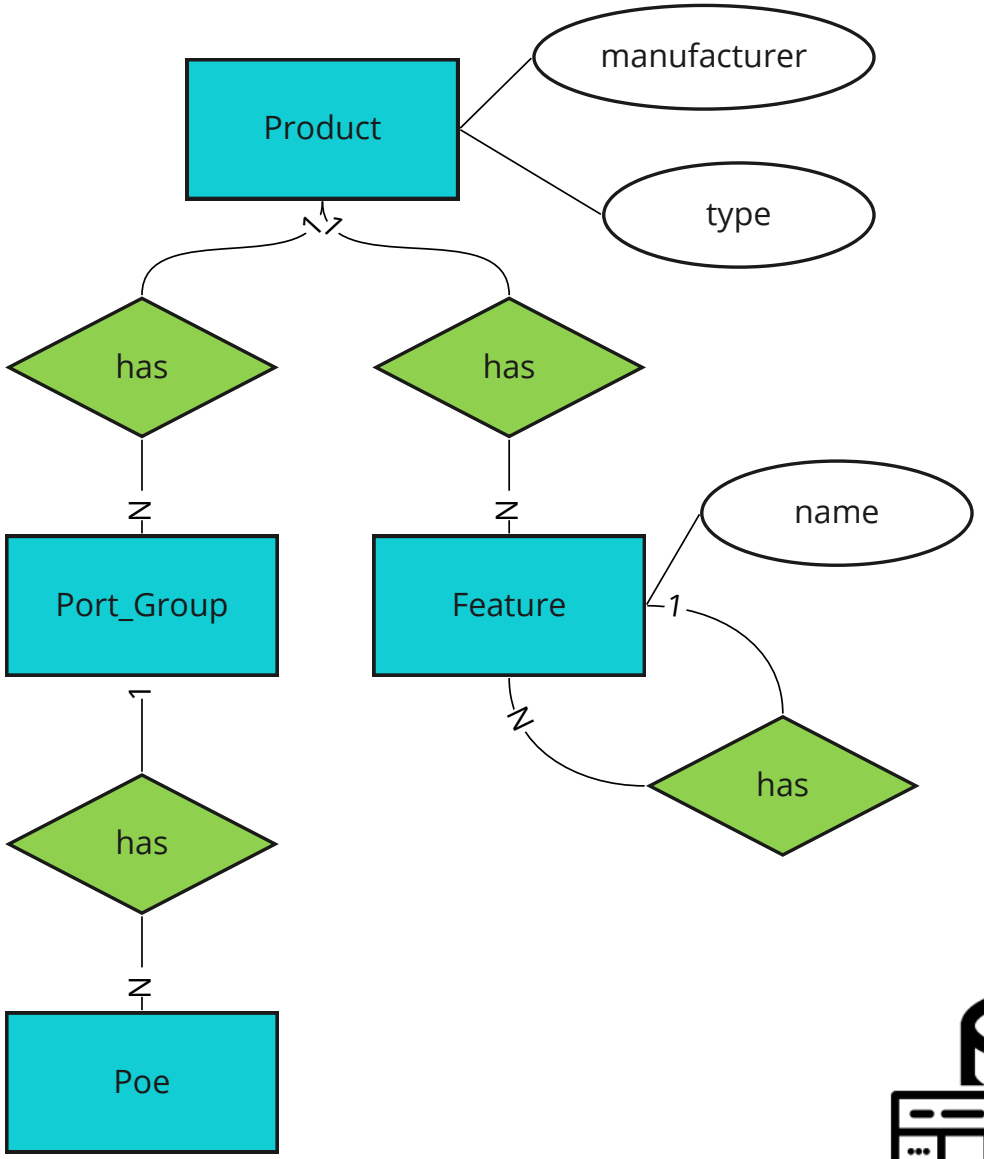
Features können Features enthalten, das ist neu, aber eigentlich auch nur ein weiteres Attribut in Features.

Datenmodell für spezifische Tabellen

Datenmodell für eav Ansatz und weitere Generalsierung indem Port-Group und PoE auch zu Feature wird.

EAV


Product
manufacturer
type
N
has
has
N
N
Port_Group
Feature
name
1
has
N
Poe


Product
manufacturer
type
N
has
has
N
N
Port_Group
Feature
name
1
N
has
1
has
N
Poe

Datenstruktur

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

8.47700024 seconds

Worksheet | Query Builder

```sql
ALTER TABLE feature ADD amount_of_rules NUMBER(10);
ALTER TABLE feature ADD amount_of_connections NUMBER(10);
ALTER TABLE feature ADD amount_of_nat_rules NUMBER(10);
ALTER TABLE feature ADD ssl_inspection VARCHAR2(10);
ALTER TABLE feature ADD amount_of_tunnels NUMBER(10);
ALTER TABLE feature ADD amount_of_users NUMBER(10);
ALTER TABLE feature ADD parent_feat_id NUMBER(10);

ALTER TABLE feature ADD CONSTRAINT feat_feat_fk
  FOREIGN KEY (parent_feat_id) REFERENCES feature (feat_id);
```

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

8.47700024 seconds

Worksheet | Query Builder

```sql
ALTER TABLE feature_eav ADD parent_feat_id NUMBER(10);
ALTER TABLE feature_eav ADD CONSTRAINT feae_feae_fk
  FOREIGN KEY (parent_feat_id) REFERENCES feature_eav (feat_id);
```

EAV

Neue Daten

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

Worksheet | Query Builder

```
INSERT INTO feature_eav (feat_id, prod_id, name)...

INSERT INTO feature_eav (feat_id, parent_feat_id, name)...

INSERT INTO attribute (attr_id, name)
VALUES (86, 'amount_of_rules'),
       (87, 'amount
       (88, 'amount
       (89, 'ssl_in
       (90, 'amount
       (91, 'amount
```

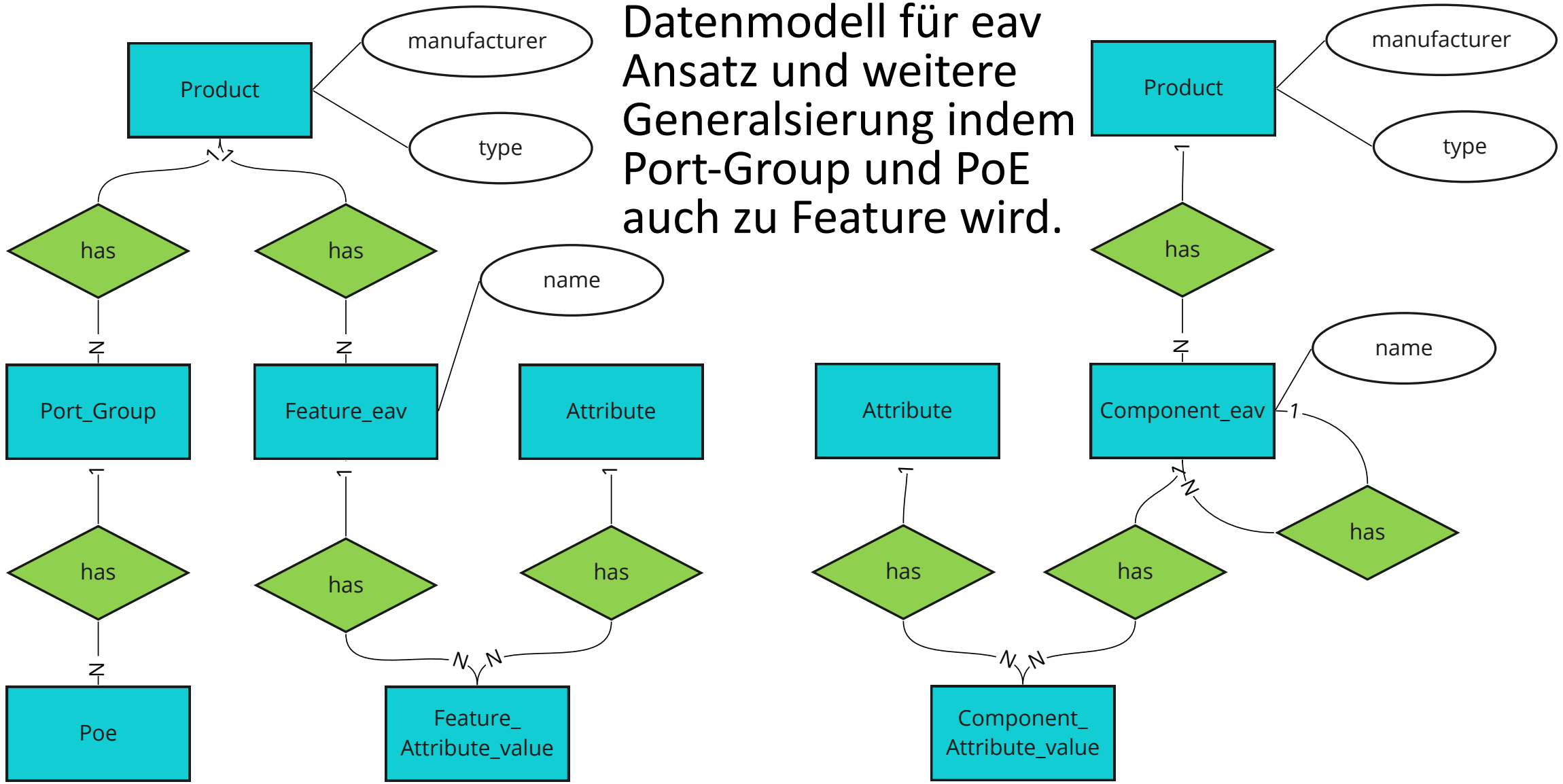
Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA | Docker_loaca

Worksheet | Query Builder

```
INSERT INTO feature_attribute_value (valu_id, feat_id, attr_id, value)
VALUES (31, 41, 81, '4094'),
       (32, 42, 81, '8'),
       (33, 43, 84, 'static, RIP, OSPF, BGP, EGP'),
       (34, 43, 85, '1000'),
       (35, 44, 82, 'stateful, stateless'),
       (36, 44, 86, '1000'),
       (37, 44, 87, '100000'),
       (38, 44, 88, '1000'),
       (39, 45, 82, 'IPsec, SSL, L2TP, PPTP'),
       (40, 45, 90, '1000'),
       (41, 45, 91, '1000'),
```

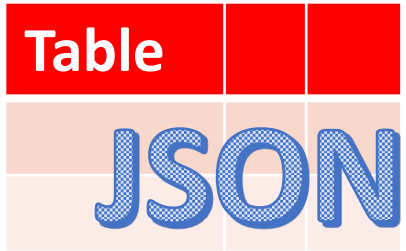
EAV

Datenmodell für eav Ansatz und weitere Generalsierung indem Port-Group und PoE auch zu Feature wird.



EAV

Gäbe es da noch andere Lösungen?

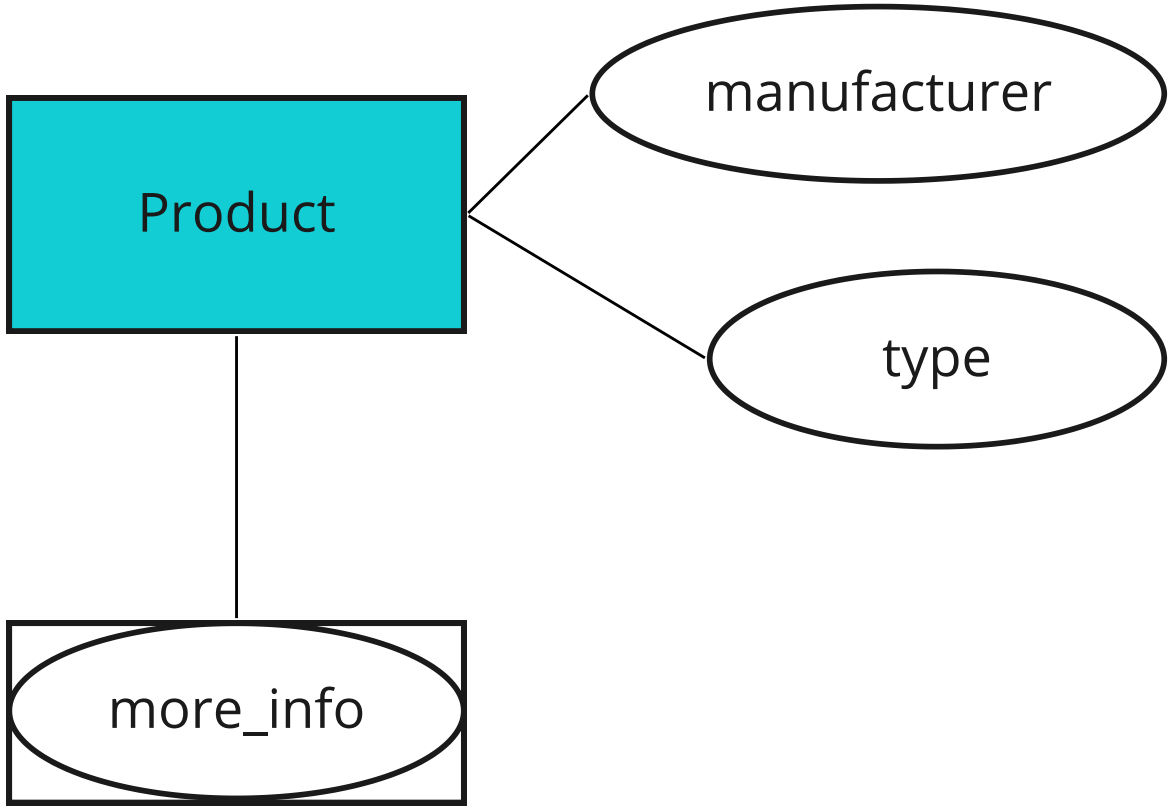


JSON in Attribut

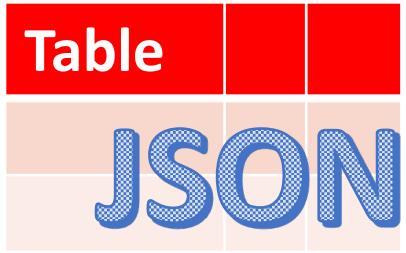
Prinzip erklären

Datenmodell

Demo



Gäbe es da noch andere Lösungen?



Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

Worksheet | Query Builder

```
CREATE TABLE product_flex (
   prod_id       NUMBER(10),
   type          VARCHAR2(50),
   name          VARCHAR2(50),
   description   VARCHAR2(200),
   manufacturer  VARCHAR2(50),
   price         NUMBER (8,2),
   more_info     JSON
```

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

Docker_loacal_ANDREA

Worksheet | Query Builder

```
INSERT INTO product_flex (prod_id, type, name, description, manufacturer, price, more_info)
VALUES (1, 'network switch', 'Fritz A16', 'unmanaged 16 port network switch', 'ABC', 100,
 json('{
        "port_group": [
            {
                "amount": 16,
                "type": "RJ45",
                "speeds": "10/100/1000"
            }
        ]
    }'));
```

Haben wir Geräte mit einer Port-Group vom Typ SFP?

```
SELECT pf.prod_id, pf.type product_type, pf.name,
      jt.*
FROM product_flex pf,
    JSON_TABLE(
        pf.more_info
        COLUMNS (
            NESTED port_group[*]
            COLUMNS (
                amount NUMBER(3) PATH '$.amount',
                type VARCHAR2(50) PATH '$.type'
            )
        )
    ) jt
WHERE jt.type = 'SFP';
```

Gib mir alle Angaben zum network switch “Fritz A16”

Welcome Page | Docker_loacal_ANDREA | edbsHS2023_ANDREA

Worksheet | Query Builder

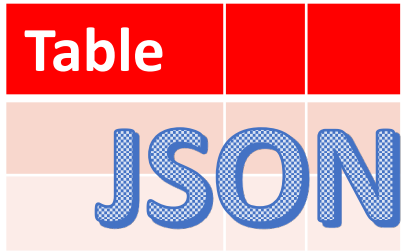
```
SELECT p.prod_id, p.type product_type, p.name, p.price
FROM product p INNER JOIN ...

SELECT pc.*
FROM product_collection pc
WHERE pc.json_data.type = 'network switch'
  AND pc.json_data.name = 'Fritz A16';

SELECT pf.*
FROM product_flex pf
WHERE pf.type = 'network switch'
  AND pf.name = 'Fritz A16';
```

Problem, wenn JSON ein typo hat

Zusammenfassung und Abstimmung



- Dokument Datenbank

- Tabelle mit JSON
- Relationale Datenbank
- EAV

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

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