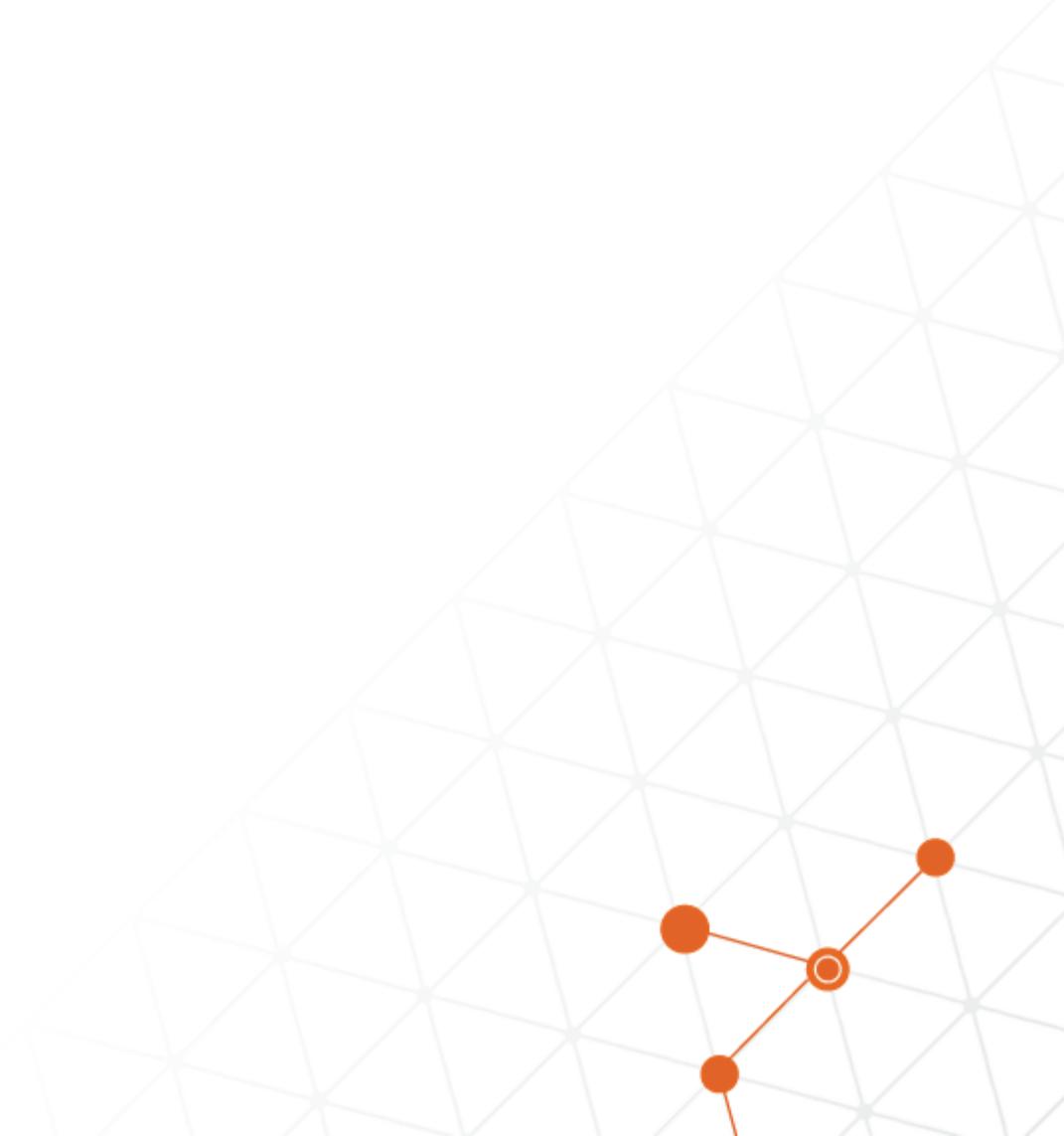


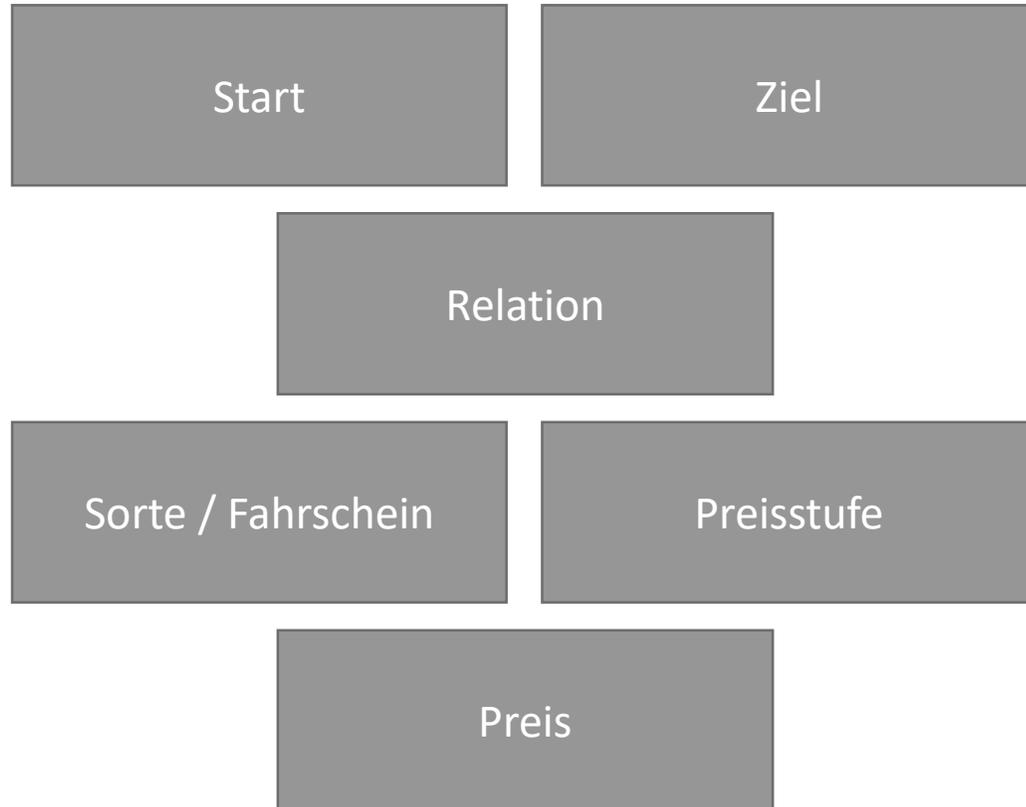
Tarifdaten

Beratertag 13.02.2020



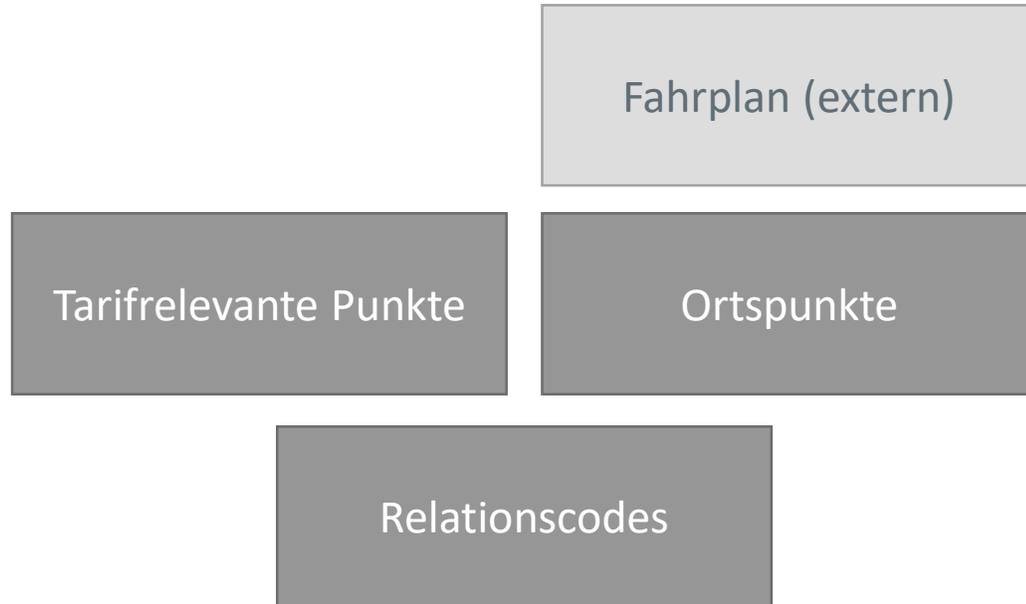
- XML Format
 - XSD / Schema Datei Verfügbar
- Datenbankkompatibel
 - Datenbankdefinition in der Schemadatei
 - SQLite Abbildung (ab Version 3.0)
 - Create SQL Statement (ab Version 3.0)
- Innerhalb einer Hauptversion abwärtskompatibel
- Dynamische Erweiterung von Entitäten

Grobe Struktur



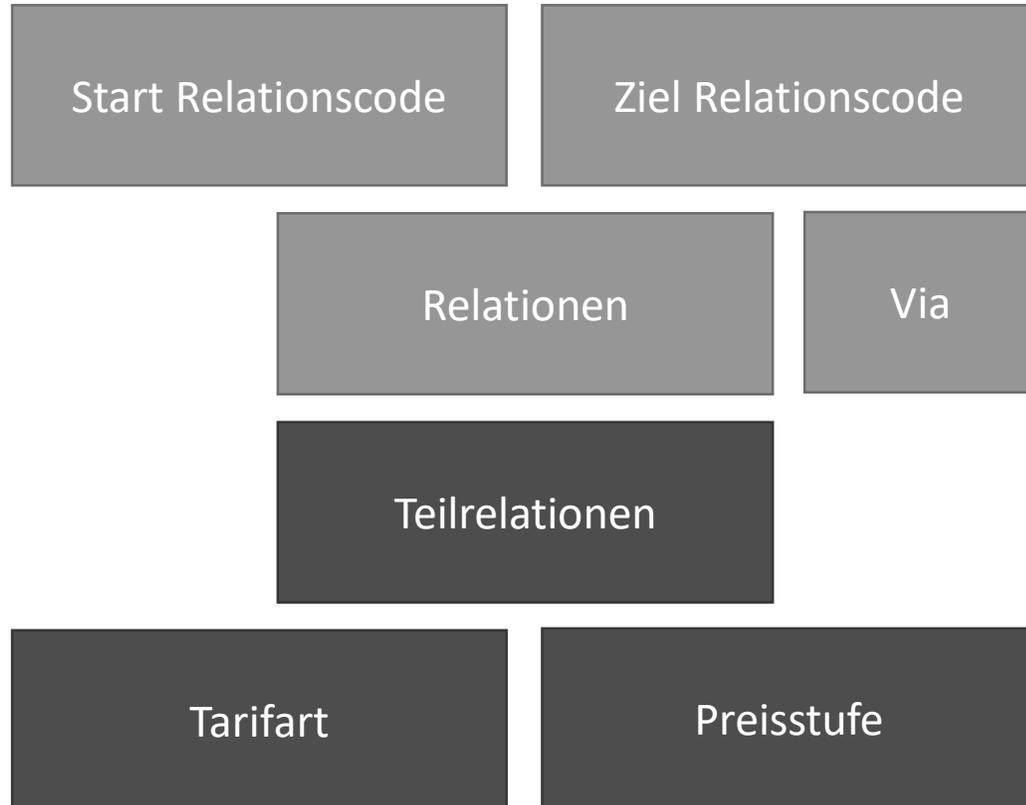
- Klassischer Relationsverkauf
- Festpreisverkauf ab Sorte / Fahrschein identisch

Grobe Struktur – Start / Ziel Auswahl



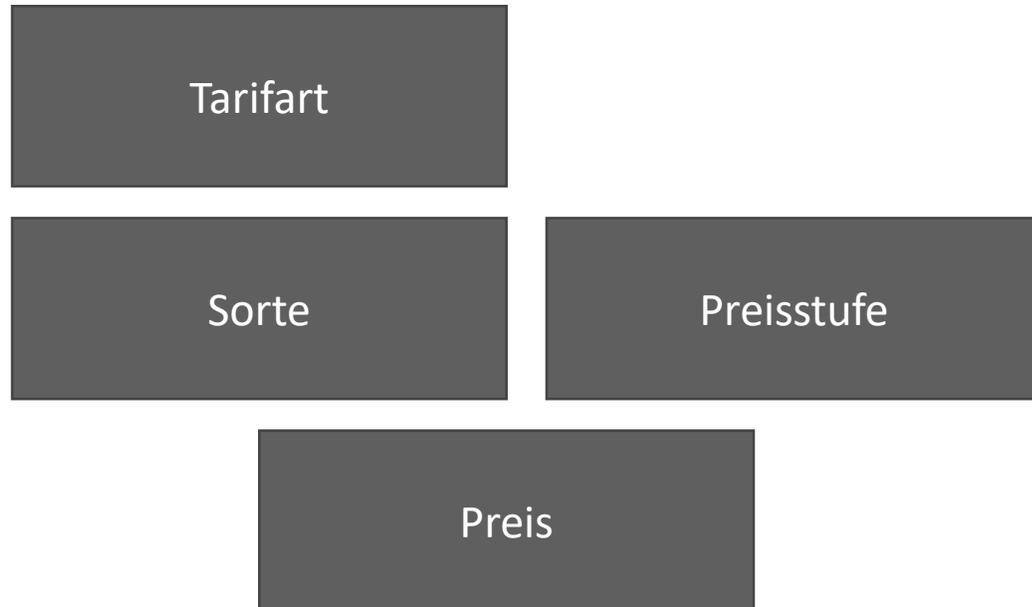
- Manuelle Auswahl
 - Beliebige Namen
 - Mehrere Namen für dieselbe Haltestelle
- Auswahl über Fahrplan
 - Direkte Referenz in Husst
- Befindet sich in der Angebotsebene

Grobe Struktur – Relation Ermitteln



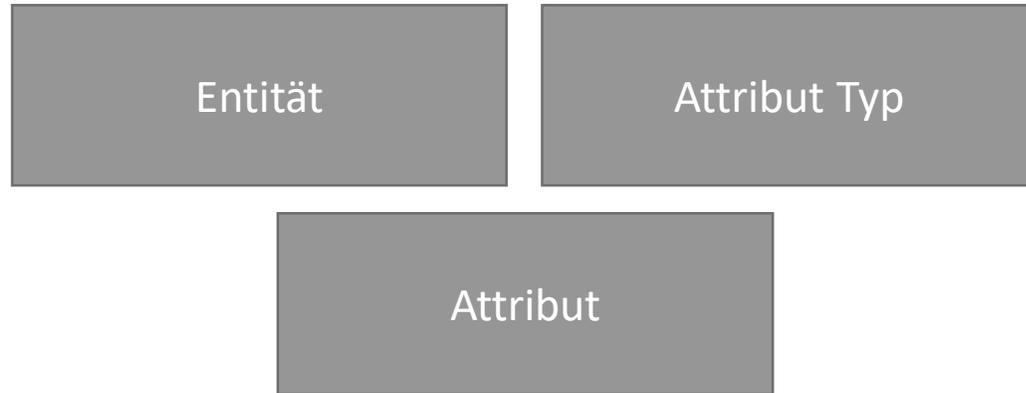
- Relationsfindung auf Angebotsebene
- Selbe Relation in verschiedenen Tarifen
- Teilrelation verweist auf Tarifpunkte (Waben / Zonen / etc.)
- Tarifart ist eine Zusammenfassung mehrere Sorten

Grobe Struktur – Preis Ermitteln

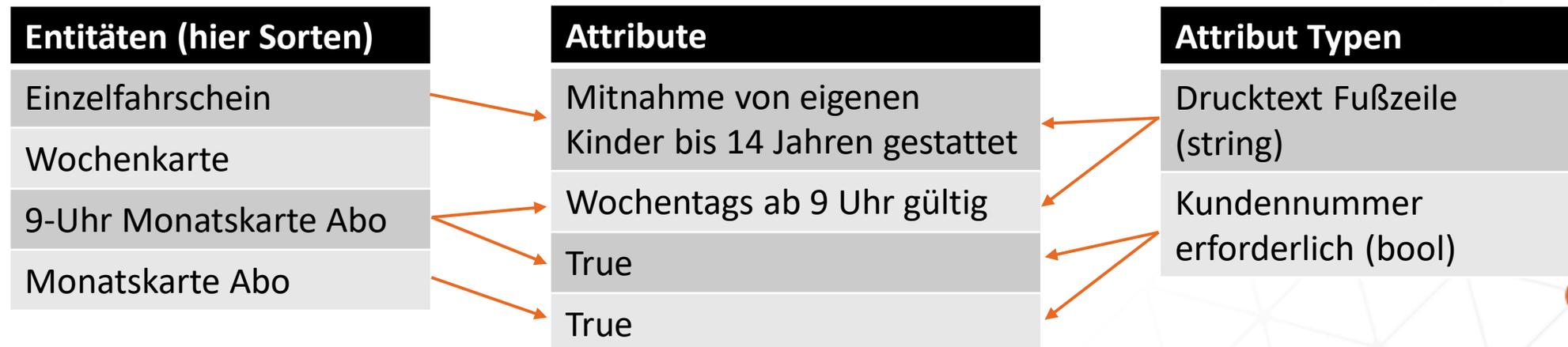


- Sorten können in Sortengruppen zusammengefasst werden
- Bsp.: Anwendungsfälle Sortengruppen
 - Sortimentszusammenfassung (Einzelfahrschein)
 - Gruppenfahrschein
 - Anschlussfahrschein Verkauf

Dynamische Attribute



- Entität kann pro Typ ein Attribute besitzen



Neuerungen Husst 3.0

- Dynamische Attribute in allen Tabellen
- Namensbereinigung
 - ID / Nummer / „Externe Referenz“
- Viele optionale Elemente entfernt
 - Nutzung der dynamischen Attribute
- Relationen -> Teilrelationen
 - N-zu-M Beziehung
- Enumerationen



Neuerungen Husst 3.0 – Bsp. Sorte

- Aus der Sorte wurden die meisten Attribute entfernt
- Viele Attribute wurden nur in einem Projekt genutzt
- Relevante Attribute davon werden als Husst Dynamische Attribut Typen aufgenommen

(links Husst 2, rechts Husst 3)

ID_Zeitraum
ID_Sorte
Deaktiviert
ID_Tarifgebiet
ID_Sortenklasse
SorteTarifgeber
BezeichnungKurz
BezeichnungLang
AnzeigeKurz
AnzeigeLang
KundendisplayKurz
KundendisplayLang
BelegdruckKurz
BelegdruckLang
Ausgabeoption
Druckoption
Layout
LayoutBeleg
BelegAnzahl
Sammelbeleg
Vertriebswege
ID_MwSt
ZahlungsArt
Komfortklasse
Rabattklasse
Fahrgasttyp
Sortentyp
Zusatzorte
Mindestpersonenanzahl
Hoechstpersonenanzahl
KdNummer
GueiligRegel
GueiligParam
GueiligText
GueiligRegelEfs
GueiligParamEfs
Vorverkauf
Umschalttag
PreisuebergangVorverkauf
Preisfindung
Preisabschlussberechnung
AnzahlBerechtigungen
AnzahlAbschnitte
HinUndRueckfahrt
TarifInfo
Drucktext1
Drucktext2
Drucktext3
KA_ReferenzEfs
KA_ProdOrgId
KA_ProdNr
KA_infotext
KA_Fahrgasttyp
KA_Mitnahme
KA_Verkehrsmittel
KA_Serviceklasse

Deaktiviert
ID_Zeitraum
ID_Sorte
ID_Tarifgebiet
ID_Sortenklasse
ReferenzExtern
BezeichnungKurz
Bezeichnung
ID_Ausgaberegulung
ID_Druckregelung
Layout
ID_Vertriebswege
ID_Mwst
ID_Zahlungsarten
ID_Komfortklasse
ID_Rabattklasse
ID_Fahrgasttyp
ID_Sortentyp
Mindestpersonenanzahl
Hoechstpersonenanzahl
ID_GueligkeitszeitRegel
ID_Preisfindung
ID_Reisotyp
BezeichnungDruck
KA_ProdOrgID
KA_ProdNr
KA_Infotext
KA_Fahrgasttyp
KA_Verkehrsmittel
KA_Serviceklasse



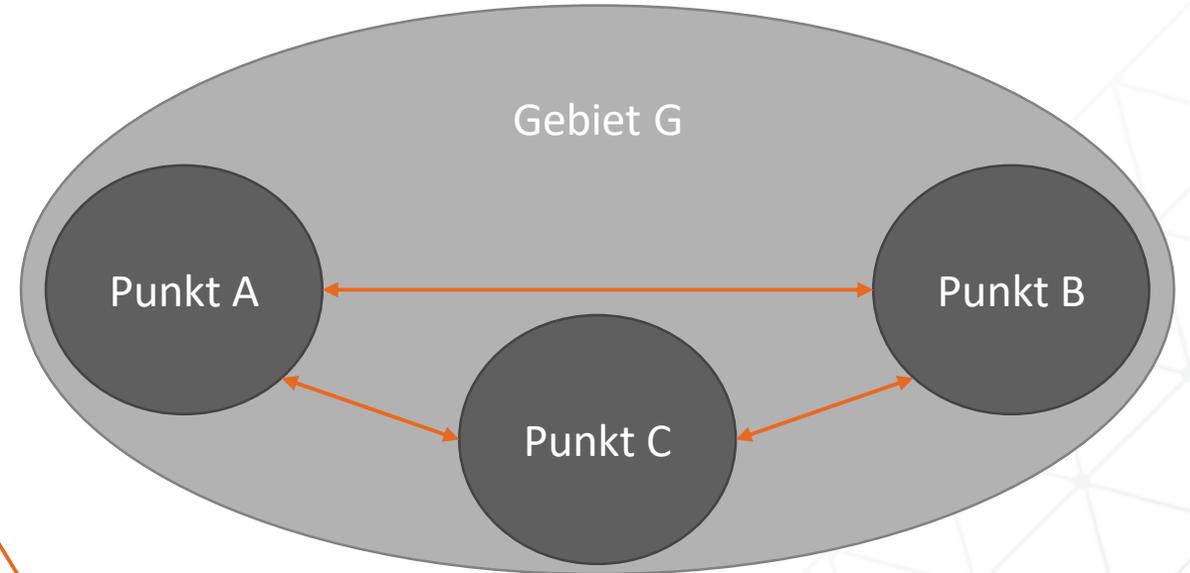
Neuerungen Husst 3.0 – Relationen M-zu-N

- Wiederverwendung von Teilrelationen möglich
- Zusätzliche Berechnungsregelungen an der Beziehungsentität

Teilrelationen				
1	A	B	PS1	Einzelt.
2	A	C	PS1	Einzelt.
3	B	C	PS1	Einzelt.
4	G	G	PS2	Gebietst.

Teilrel.	Rel.
1	1
4	1
2	2
4	2
3	3
4	3

Relationen		
1	A	B
2	A	C
3	B	C



Neuerungen Husst 3.0 - Enumerations

KundenTyp_CODE				
Der Kundentyp_CODE klassifiziert Kunden bzgl. typischer tarifrelevanter Kundengruppen des ÖPV.				
Attribute				
Attributname	Beschreibung	Kard.	Wertebereich	Format
Code	EN 1545: ProfileCodeOP	1..1	ENUMERATED (nicht spezifiziert/ unbestimmt 0 Erwachsener/ Vollzahler/jedermann 1 Kind 2 Student 3 Behinderter nicht weiter spezifiziert 5 Sehbehinderter 6 Hörgeschädigter 7 Arbeitsloser/ Sozialhilfeempfänger 8 Peronal 9 Militärangehöriger 10 Kostenpflichtiges Tier 17 Schüler 19 Azubi 20 Senior 25 RfuCE 26 ... 63 KA-spezifisch ermäßig 66 ...250 Fahrrad 64 Hund 65 Hund 66 ... frei für nicht interoperable Verwendung (251..255))	INT1

- KA-Enumerations (EN Standard) wenn sinnvoll übernommen
- Zusätzlich eigene Werte erlaubt
 - String-Enumerations mit „husst.“-Präfix
 - Int-Enumeration ab 1000 eigen
 - Flag-Enumerations ab 1024 eigen