

Data Vault - Catalog of field mappings in relations

(C) Matthias Wegner, cimt ag

Creative Commons License [CC BY-ND 4.0](#)

To create a toolset for loading data into a data vault model, we need to determine the completeness of the toolset. This article investigates the different possibilities how the source fields are mapped, when multiple relations to the same hub are delivered in one row.

Example of this requirement are:

- order, referring the customer in different roles (delivery and invoice).
- flight, referring the start and destination airport

Related Document

So also [Model topologies and basic field mapping variations](#) about possible variations of field distribution especially according one field to many columns.

Definitions

When describing the data we classify the elements as follows:

field: smallest element of source data. Will always be processed as unity. Will be stored in one or multiple columns in the data vault.

source row: the fixed structure of fields, containing the data of one or more business-objects and their relation in a single row/unit.

table/hub/sat/link: a table in the data vault model

column: a column in a table of the data vault model

business key (bk): data, used to identify business objects

dependent child key (dc): data, containing relation attributes

table key/hub key/link key: The join key of data vault model tables

content: Data that is not used for identification, and just stored in the data model

Property of the source data

Tabularized

Data can be complex in multiple ways, especially when it comes to hierarchical data or document formats. The following approach requires the source data representation to be tabularized (all data is organized in Rows,

every row contains all fields).

Information types of data

To define the variety of mappings, it is necessary to clarify the types of information, represented by a source field.

- (Part of) the identification of an object
- Attribution or Measure of an object
- Attribution or Measure in a relation between objects

Relations are expressed in the source by containing identifications of different objects in the same row.

Note: The data vault main stereotypes map to this classification as follows. hub=object / link=relation / satellite=attribution.

2nd Note: data that is stored in dependent child key columns of a link counts also as identification, since it is needed to address attributes, that are attached by the satellite

Properties of relation data

- Relation data always must contain the business key columns of all participants.
- Data sets with multiple relations to the same object must contain multiple instances of the business key fields. It might (but must not) contain multiple instances of content data fields.

There are two *flavours* for relations in the source data.

- **Business object relation**, is the obvious flavour, covering any relation between business objects or self relation in hierarchies
- **Unit of work relation**, expresses the circumstance that multiple objects are delivered in the same data row, but without any known business relation meaning

Unit of work relations might be misread as a lack of normalization in the source data. But as the words "without any known" indicate, it might just be a lack of knowledge about a hidden meaning.

Denormalized data

When source data contains multiple fields, which target the same satellite columns without any different business keys, this might look like denormalized data and trigger the desire to normalize it into a multiactive satellite.

Data vault highly recommends to keep the denormalized structure in the raw vault to allow full auditability. That's why DVPD core will not support any explicit syntax that allows denormalization in the load phase.

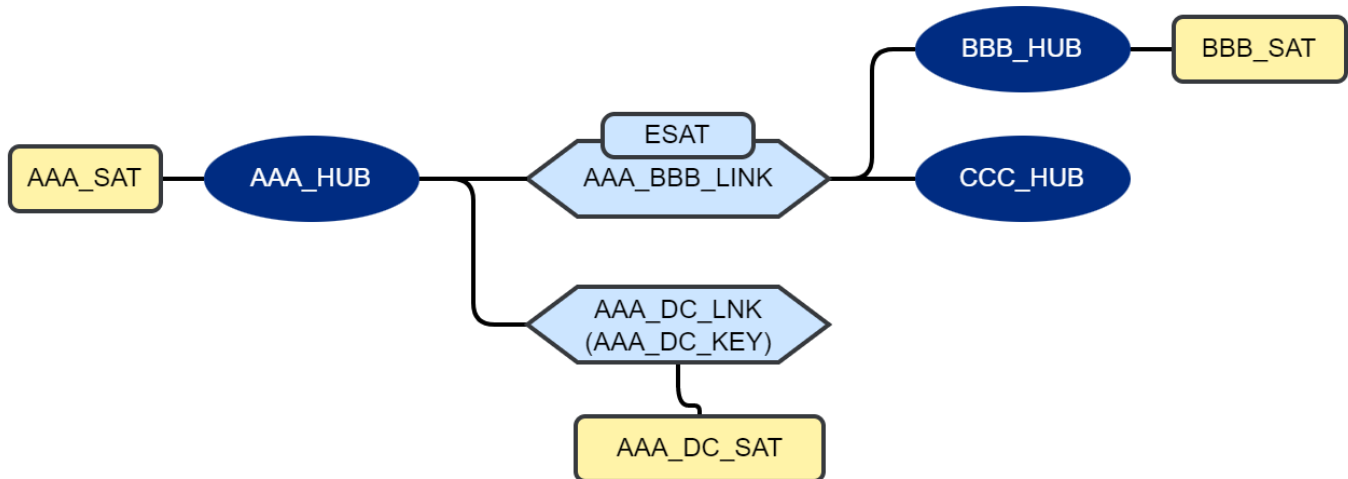
Model topologies

This section identifies the data vault model topologies, that have different properties regarding field mapping variations. The topologies will be used later to describe the different mapping scenarios.

Simple relations

As long, as there are no fields mapped to the same data vault column, the mapping stays simple, regardless of the complexity of the model topology. For every table there is only one set of fields, that has to be used for hashing and loading.

The following example will be used as a base to work out the possible mapping variations. It is designed to embed the most common simple models as a subset. By being capable to define this model with DVPD, the coverage of all subsets is also proven.



Multi relations to same hub

For sources that provide multiple relations to the same object in their data, the data vault model must provide a proper structure.

The following approaches are available to represent multiple relations

- a single link with multiple hub keys of the same hub. One for every kind of relation
- dedicated links to the hub for every kind of relation
- a single link but dedicated effectivity satellites for every kind of relation
- a single link with a dependent child key, declaring the relation type
- a single link with a multi active satellite, that contains a column to store the currently valid relation types (not recommended)

These approaches can also be mixed up, which might happen on purpose or due to legacy.

side note: When the relation type is declared in a field in the data (not by the field structure), this "just data" from the perspective of the data vault. This kind of relation type can simply be stored in a dependent child key or a satellite of the link.

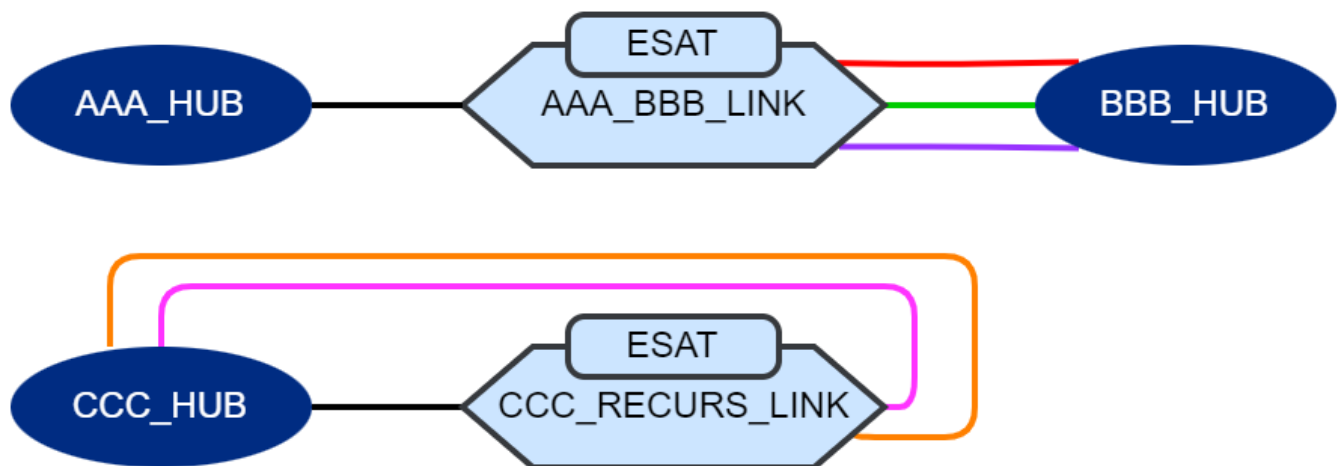
The example models in the investigation are created with 3 relations to the same hub, even though this is a rare constellation. This is necessary to prove completeness of the DVDP syntax. Only with 3 Elements or more, it is possible to create subsets with more than 1 element and less than all elements.

Satellites are omitted in the diagram for simplicity. There can be satellites on every hub and the effectivity satellites can be replaced by normal satellites.

Multiple hub keys in link to the same hub (Type-R)

This approach keeps the provided unit of work together but forces a refactoring when another relation type needs to be added (See also the discussion of the concept in chapter 4.4.4 of the Data Vault 2.0 Book of Dan Linstedt). It uses the minimal amount of tables

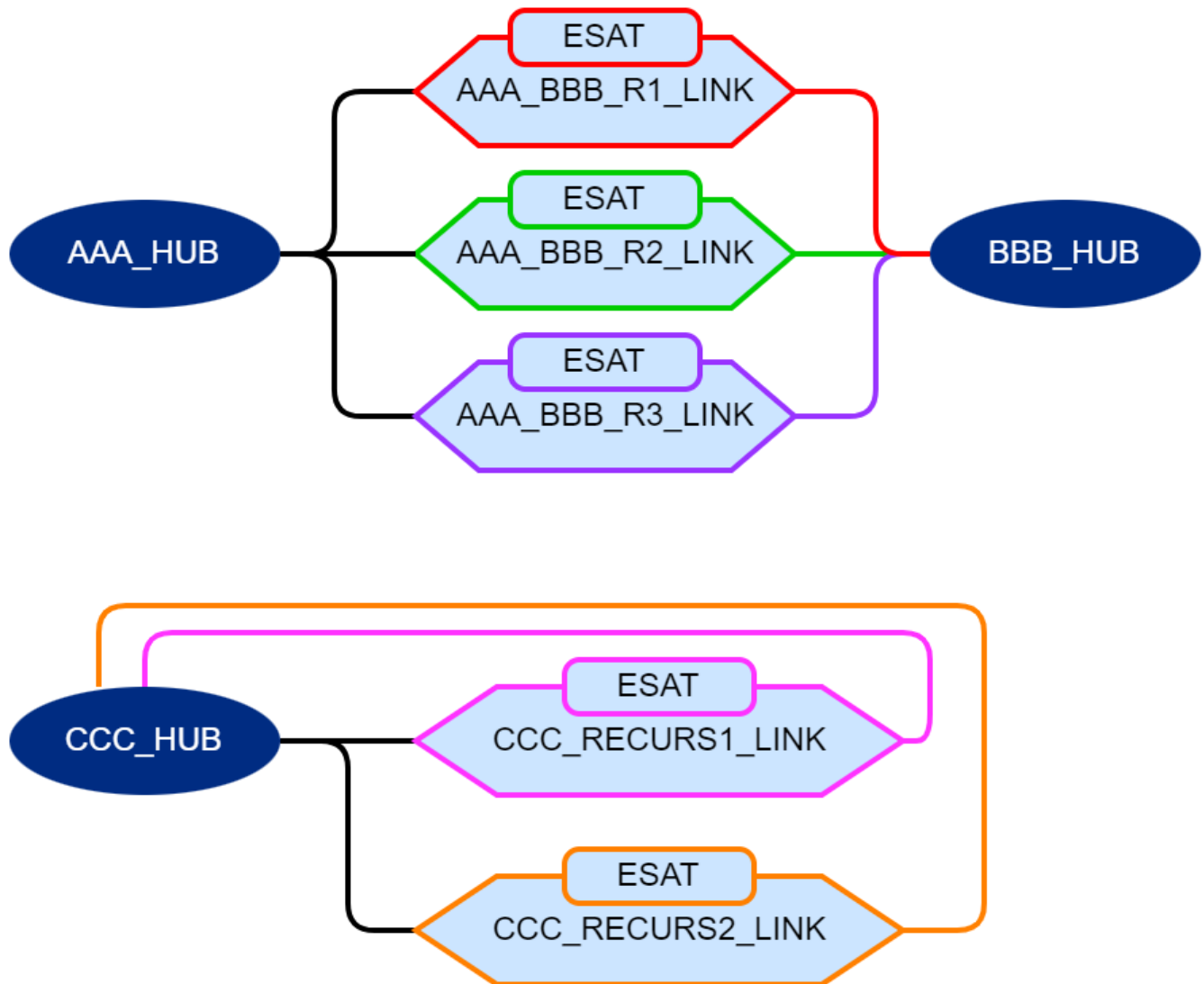
- The link contains a hub key for every relation type to the hub
- The hub keys must be named properly to explain the kind of relation, they represent
- Depending on the meaning, one relation might be the "main" object. This relation might use the original hub key column name from the hub. Hierarchical links are a common example for this constellation.
- To calculate the link key, all business keys for the multi referenced hub have to be assembled
- Link and Esat must be loaded once.
- The multi referenced hub needs a load operation for every reference



Dedicated link tables for every relation (Type-L)

This approach is extendable without any impact to existing structures but might suffer from breaking the unit of work, and creating "phantom relations". (See also the discussion of the concept in chapter 4.4.4 of the Data Vault 2.0 Book of Dan Linstedt)

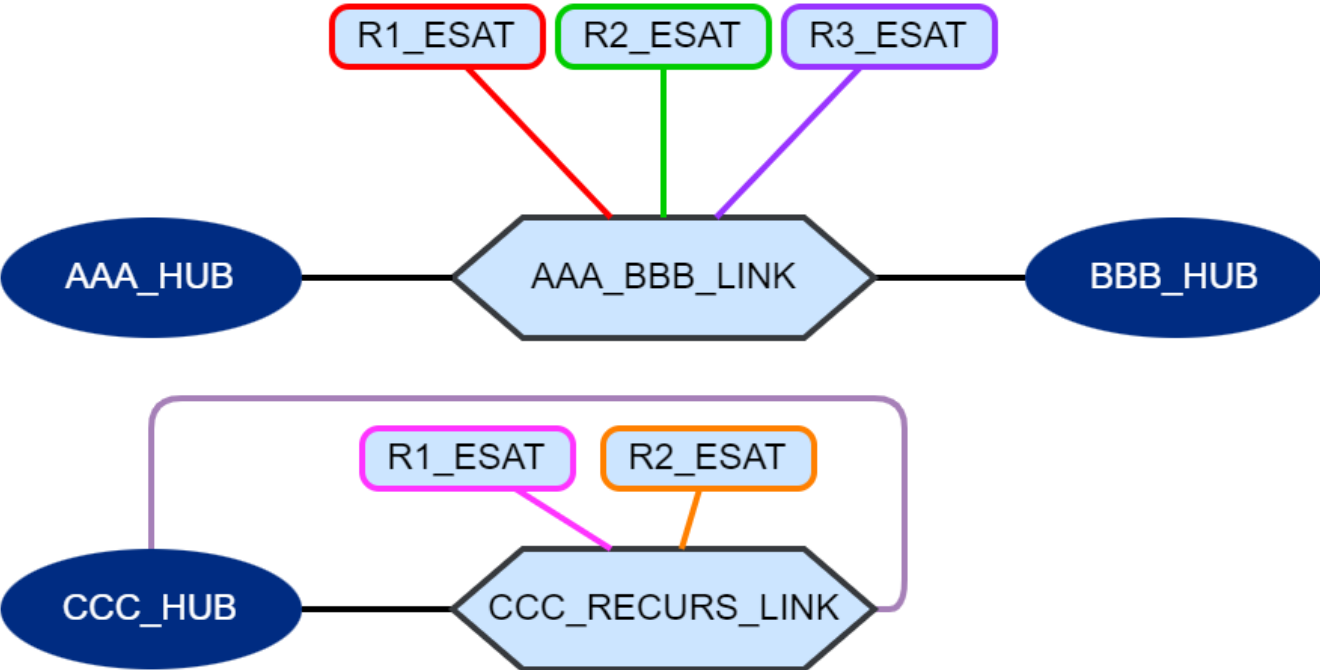
- the link key calculation only uses the business key fields for their specific relation
- every link contains one hub key for each linked hub
- every Link and Esat must be loaded once.
- The multi referenced hub needs a load operation for every reference



Single link table with relation specific effectivity satellites (Type-E)

This method reduced the number of link tables, without losing the flexibility of the multi link approach. It suffers from the same "phantom relation" issue as Type-L.

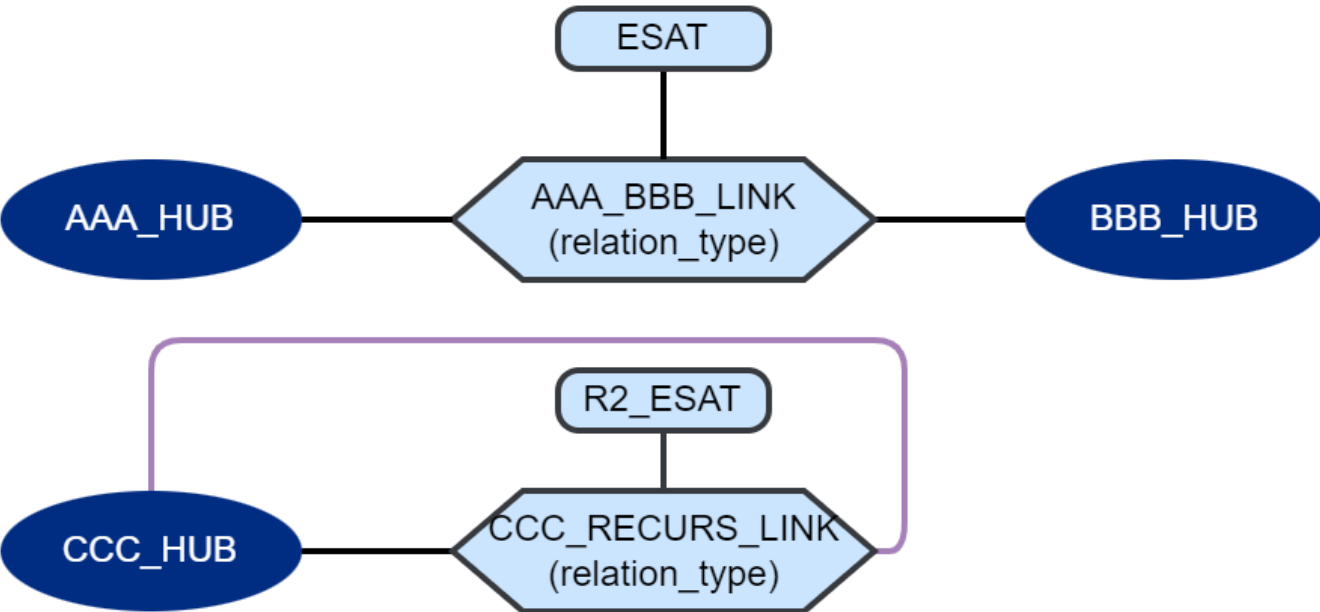
- the link only contains one reference to each hub but gathers data about all relations
- for every relation, there must be a separate the link key calculation only using the business key fields for the specific relation
- every Esat must be loaded once
- The link and the multi referenced hub needs a load operation for every relation



Single link table with a dependent child key declaring the relation type (Type-D)

This method uses the minimal amount of tables and allows extension of relations without any structure modification. It even can be extended without change of running pipelines, by just adding a new pipeline for the new relations. The downside being, that it hides the different kind of relations in the data, instead of communicating it through model elements.

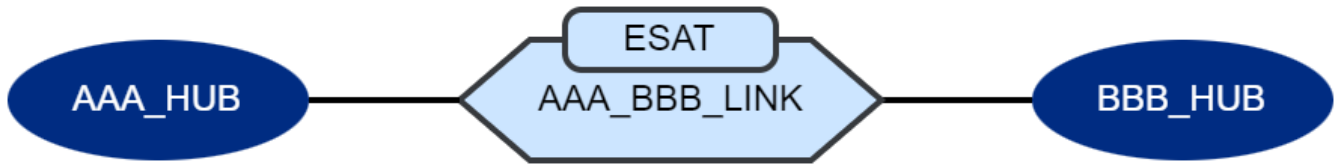
- the link only contains one reference to each hub
- for every relation, there will be a separate link key calculation only using the business key fields for a specific relation and the relation specific value in the dependent child key
- Link, esat and the multi referenced hub need a load pass for every reference



Normalize multiple relations (Type-N)

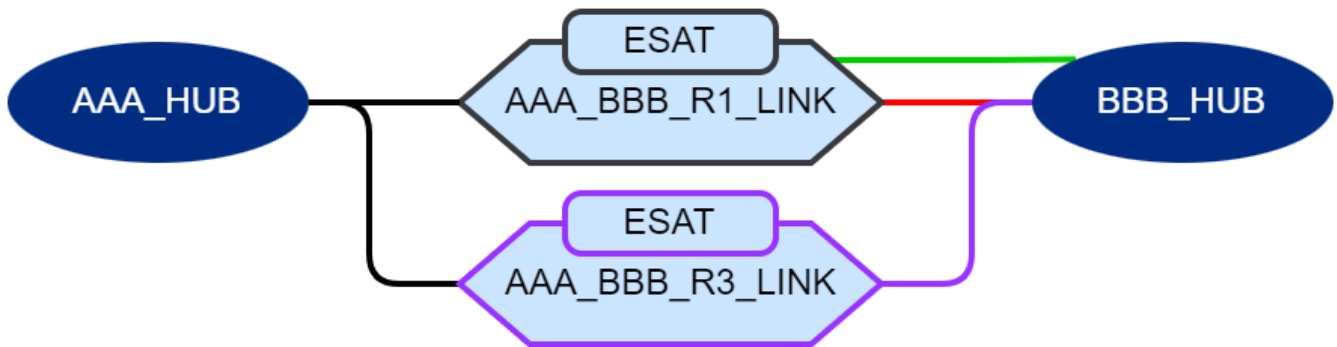
This approach omits any kind of structure to distinguish the different relations presented by the fields the same source row. This reduces the number of tables and connections to a simple model but will prevent a full reconstruction of the source dataset, since the precises field origin of the data gets lost. It is the less recommended approaches.

- the link only contains one reference to each hub
- Link, esat and the multi referenced hub need a load pass for every reference



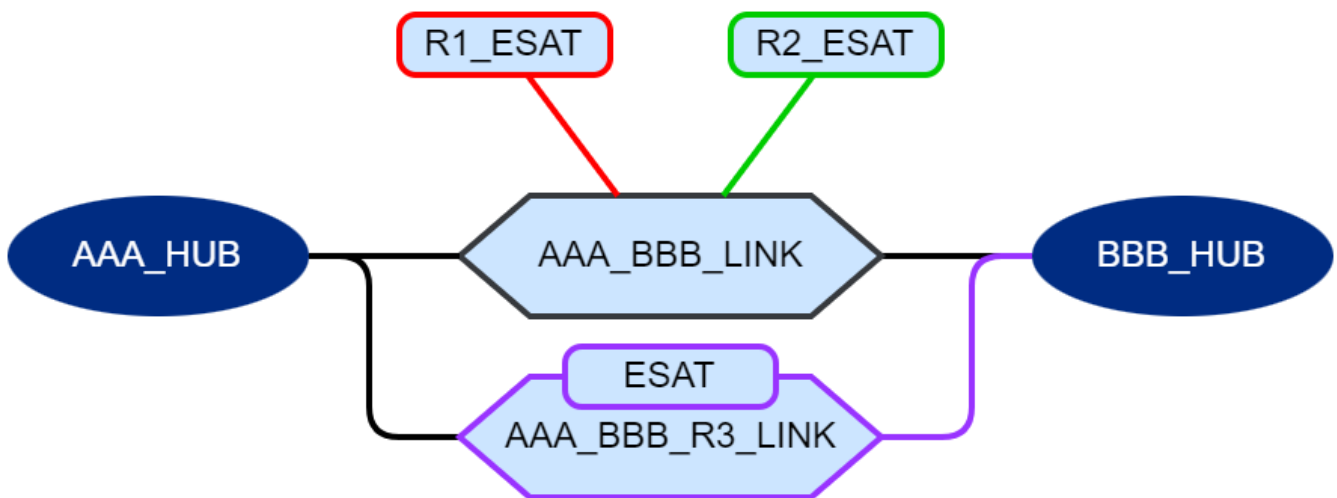
Mix of multi related link and normal link (R+L)

This model is a combinational edge case to challenge the abilities of the DVPD syntax and compiler. It might occur through model legacy.



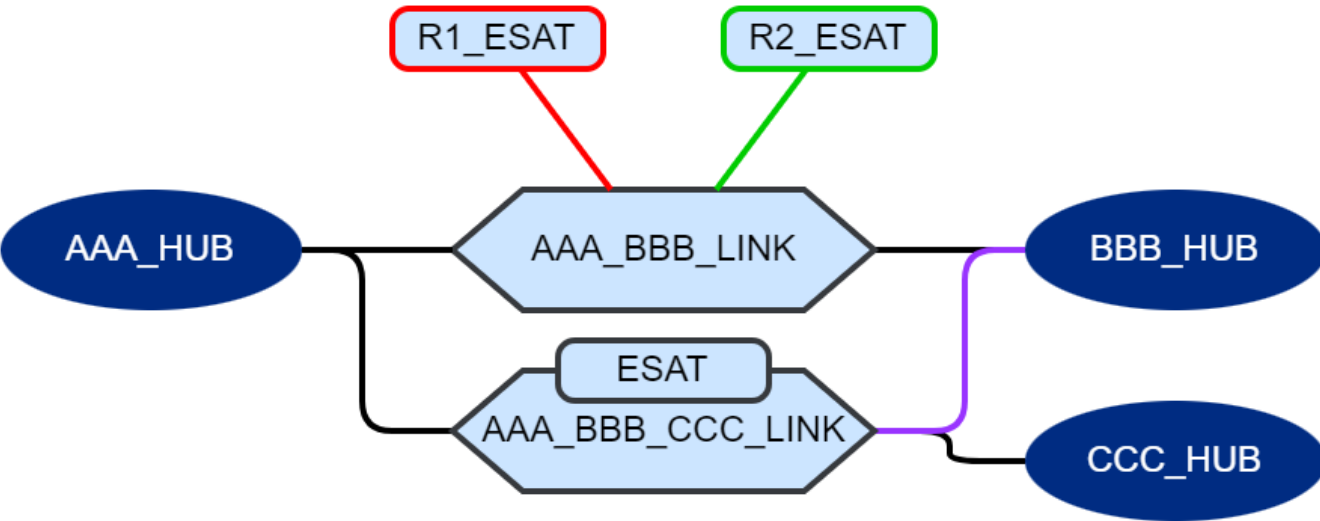
Relation specific effectivity satellites + parallel link (E+L)

This model is a combinational edge case to challenge the abilities of the DVPD syntax and compiler. It might occur through model legacy.



Relation specific effectivity satellites + parallel link with 3rd hub (E+3)

This model is used to create an edge case for mapping combinations of attributes needed for the BBB_HUB.

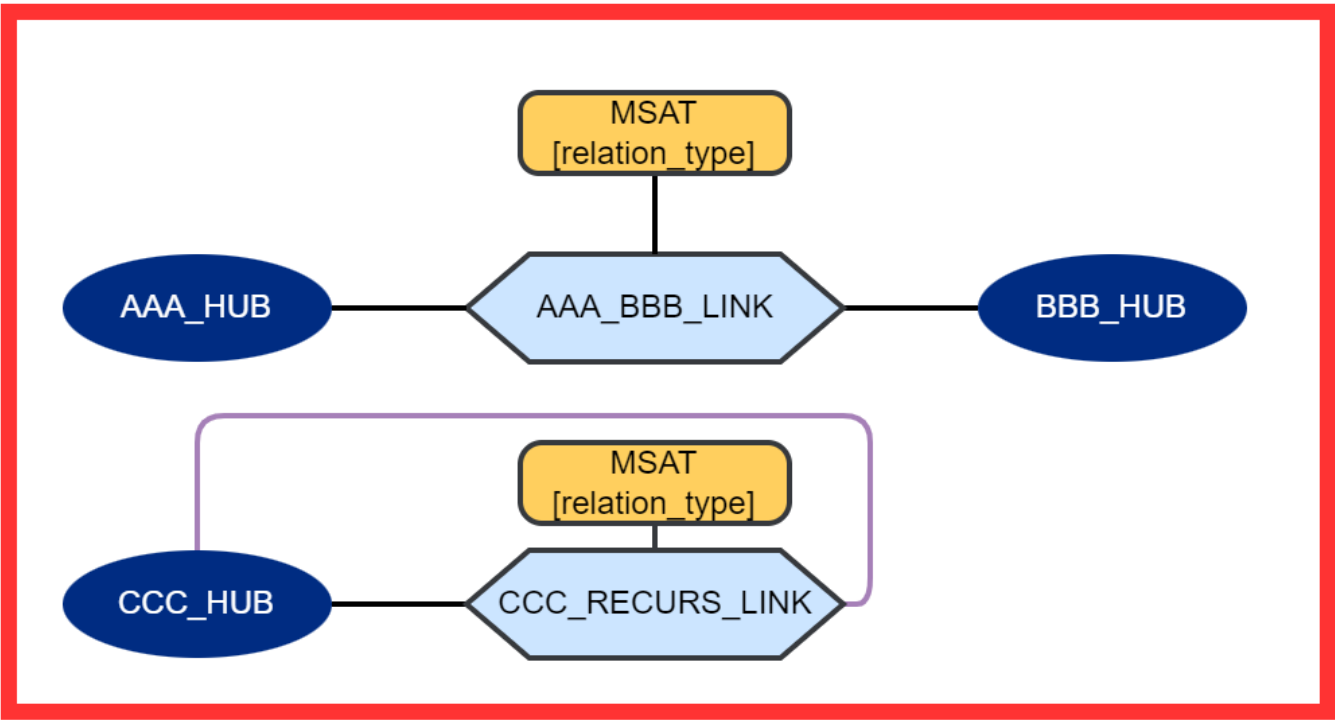


Out of scope edge cases

To keep the syntax of DVPD within a manageable complexity, some edge cases will be placed out of scope, since they are far from expected model requirements and can be solved by other designs.

Single link table with a multiactive satellite, that contains a column to store the relation type

Although completely valid for relation type information, that is contained in the data, it can't be used for relation declaraton inbetween objects of the same row, since it would need the creation of additional staging rows to feed the multiactive satellite. Creating new rows during staging has a high risk of duplicating data or generate "ghost" data, that was not in the source. Additionally the disadvantage of hiding relations in the data is the same as for the dependent child key approach.



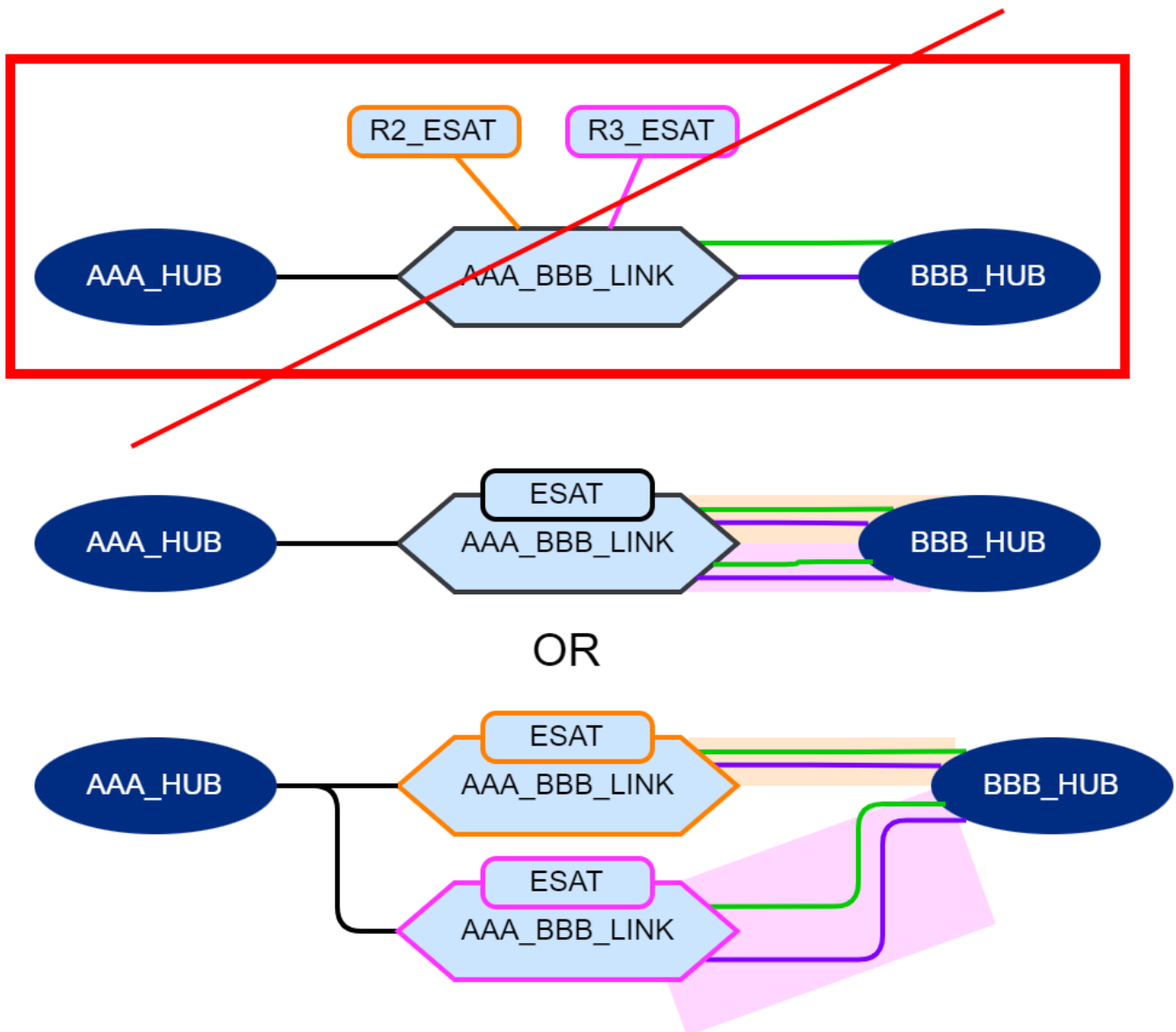
mixing esat and multi reference

Mixing different approaches at the same link is not supported. In the example two relations are expressed with two references to the hub and then are "multiplied" by two more relation types represented by esat. This makes no sense in the combination, at least not under the constraint, that we always only load a single tabularized source data structure.

It might be an expression of the following:

- all 4 relations are a unit of work
- pairs of 2 relations are a unit of work and are modelled seperatly

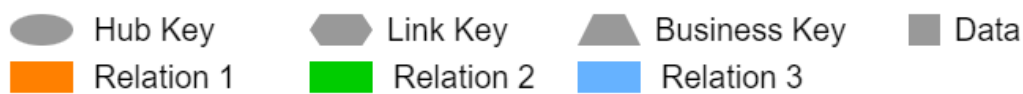
The same information can be modeled either with a link, having 4 relations to the same hub, or by modelling two links, keeping the pairs.



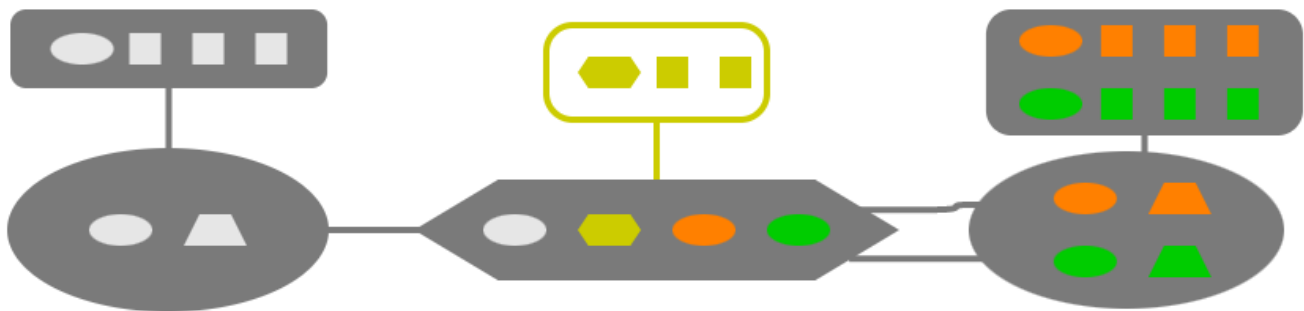
Relation participation

Observations and conclusions

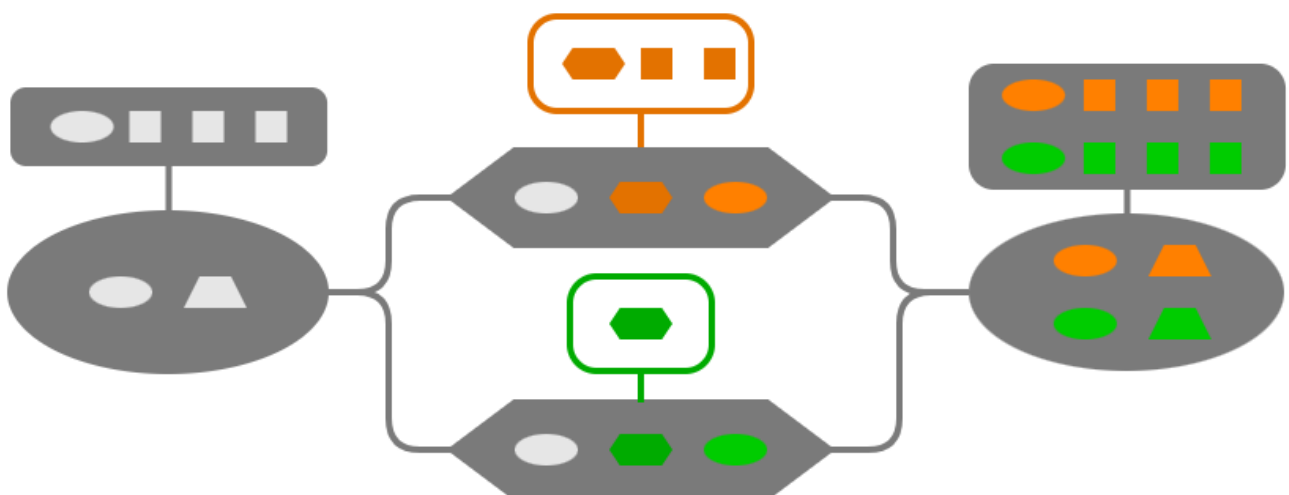
- 1-Multiple relations are only possible with multiple sets of business key fields for the hub, they refer.
- 2-From 1. -> every kind of multi relation to a hub needs at least one business key field mapping, that are restricted to a specific relation
- 3-hubs need to be loaded for every relation declared by their business key mappings
- 4-links can only contribute to relations, that are declared in the hubs they connect
- 5-a link with multiple references to the same hub, needs relation specific columns
- 6-a link with multiple references to the same hub, can only have satellites that contribute to the whole relation set, expressed by the link.
- 7-satellites on links contribute to every relation, they have a field mapping for
- 8-effectivity satellites need to declare the relation they track, due to the lack of a field, that would declare it
- 9-link satellites can only contribute to relations, the link can contribute according to its parents
- 10-Different dependent child keys for different relations can only be modeled with relation specific links or relation specific satellites.



Multiple hub keys in link to same hub

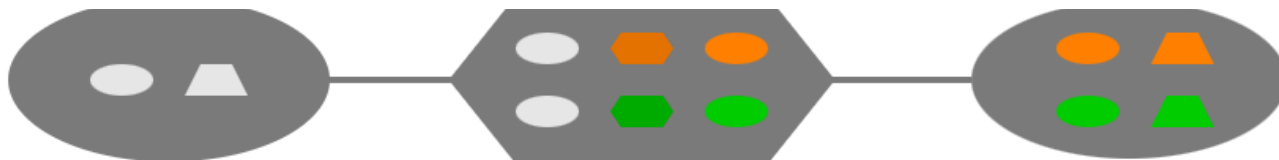


Dedicated link and (e)sat for every relation

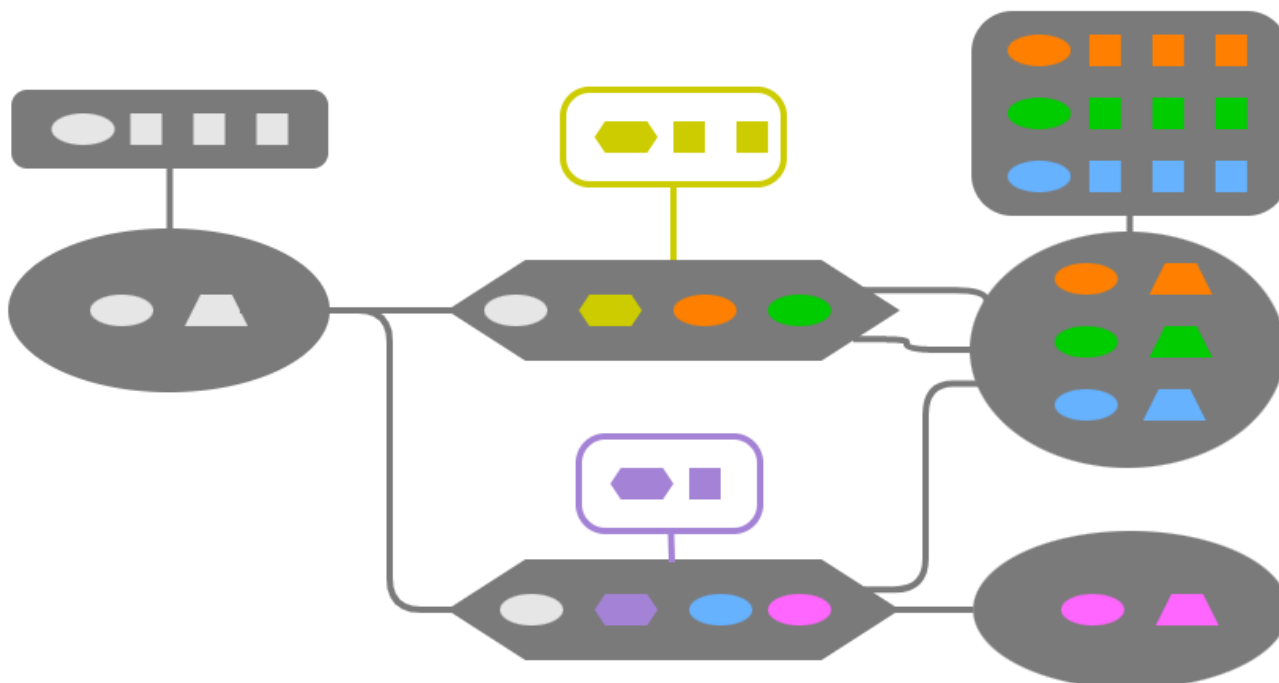


Dedicated sat for every relation





Edgecase



Participation of fields

When mapping fields to a multi related model, there are the following possibilities how a field will contribute or participate to the modelled relations:

- to one specific relation (+)
- a subset of specific relations (~)
- to all relations (*)

Depending on the function of the data, the field might be mapped to one or multiple model tables that are maybe of different stereotypes. Therefore the field can contain

- part of a business key (bk)
- a dependent child key (dc)
- part of a hubs satellite data (hs)
- part of a links satellite data (ls)

Participation to a relation is declared at every table mapping of the field. If not declared, a field is considered to participate in every relation of that table, when it is the only field mapped to the target, else it will be assigned only to the "main"(unnamed) relation. To declare the participation on a subset, that contains the "main" (unnamed) relation and another one, the syntax provides the reserved relation name "/" for the unnamed relation.

A target column must only have one field mapped in every specific relation.

Participation of hubs

Hubs participate in all relations that contain a full set of fields mapped to the business keys.

- The full set of business keys is determined by the relation with the most business key columns.
- relations with different or incomplete column outcome will fail the consistency check

These rules also include the "simple case" (only single relations), since fields without any relation declaration belong to the "main" (unnamed) relation .

If a hub only participates in the "main" unnamed relation, it is called a universal relation, since it can be used for all nonexplicitly declared relations of all connected links.

Participation of links with explicit relation mapping

These links have at least one explicit relation declaration in their parent mapping.

- they participate only in the relations that are represented by the connections
- the hubs targeted with an explicitly named relation must also have mappings for the same relation
- if a hub is referenced more then once, the hub key names in the link must be adapted
 - hub column names for the main (unnamed) reference should copy the hub key column name of the hub
 - hub column names names for the named references must be declared or will be generated by a hard rule (mostly the concatenation of targeted hub key name and relation name)
- Undeclared relations in the link belong to a "main" relation, so theses hubs must participate to the main (unnamed) or the universal relation
- Satellites on these kind of link are not allowed to declare a relation

Participation of simple links

These links have no explicit relation declaration to a hub (and therefore only one reference column for every hub).

- these links participate in all relations that are
 - declared at their satellites
 - declared at the mapping of their dependent child keys
- each relation the link contributes to, must be known by at least one connecting hub

This also covers the simple common model use case, since without declaration a satellite contributes to the main (unnamed) relation wich will place the link also in the main relation.

Participation of satellites

Satellites contribute to all relations that contain a full set of fields, mapped with relation declaration to the satellite.

- The full set of satellite columns is determined from the relation with the most columns.
- relations with different column outcome will fail the consistency check
- all relations a satellite contributes to, must be supported by the parent

The simple common model use case is covered by participating on the main (unnamed) relation when without any declaration.

Participation of effectivity satellites

Effectivity satellites contribute to the main relation unless a tracked relation is declared.

- the relation an effectivity satellite participates in must be known by the parent link

The simple common model use case is covered by participating in the relation of the link.

Catalog of field mappings

To setup a test suite for all combinations of topologies and field mappings, the following will provide a notation description method and a table of combinations

- Model:** Short notation of the model topology by specifying the links as list of the connected (single letter A-D) hub name. Every hub is preceded by the number of connections from the link to the hub, when the number is greater than 1. The name of the ends with the relation model approach `<multiplicity> <Hub> [<multiplicity> <hub>] ... <approach (indicated by [R,L,E,D,S])> +` (multiplicity is only provided when greater than 1)
 - A2BE = Link from A to B with 2 references to B, modeled as effectivity satellites
 - 3AR = Link with 3 connections to A (one origin and two recursive relations)
 - AB3CR+ABC = Link from A to B and C with 3 references to C + Link to A,B,C
 - A3BL = 3 separate Links with esats from A to B
- overlap scenario:** Describes how wide a specific kind of data (key, sat data) is spread over the model and relations.
`<content type (bk,dc,hs,ls)> <number of tables it is mapped to> <relation participation (+,~,*,-,%)> [&]`
 with BK relation participation + = one, ~ more than one but not all, * all
 with HS relation participation + = one for every BK relation, - only for one bk relation, ~ multiple but not all relations, for all bk relations, % more than one but not for all bk relations, * all relations
 - BK1+ = Business key in one table used for one relation
 - BK2* & BK1+ & HS1+ = Business key in 2 tables for all relations & business key in one table single relation & hub satellite content in 1 table and different in every bk relation
 - BK1~ & LS1~ = Business key in 1 table for a subset of relations and link satellite content in more than one but not all relations
 - BK1+ & hs1- = Business key in 1 table and hub satellite processed only in 1 relation
 - BK1+ & hs1% = Business key in 1 table and hub satellite processed only in subset of relations
- Test:** Number of the test case, that will cover this setting (set *italic* when not implemented yet, set to "-" when this combination is not possible)

Generic case matrix

The generic cases all use a 1:1 field distribution. Every field is mapped to only one target column but might participate in multiple relations.

Model	A3BR	A3BL	A3BE	A3BD	3AR	A3BN
overlap scenario						
BK1+	201	301	401	501	601	101
BK1~	202	302	402	502	-	102
BK1*	203	303	403	503	603	103
BK2+	211	311	411	511	-	111
BK2~	212	312	412	512	-	112
BK2*	213	313	413	513	-	113
BK2* & BK1+	215	315	415	515	-	115
BK2* & BK1~	216	316	416	516	-	116
BK2* & BK1*	217	317	417	517	-	117
BK1+ & HS++	220	320	420	520	620	120*
BK1+ & HS~+	221	321	421	521	-	121
BK1+ & HS~~	222	322	422	522	622	122*
BK1+ & HS*+	223	323	423	523	623	123
BK1+ & HS*~	224	324	424	524	-	124
BK1+ & HS**	225	325	425	525	625	125
BK1~ & HS++	226	326	426	526	626	126
BK1~ & HS~+	227	327	427	527	-	127*
BK1~ & HS~~	228	328	428	528	-	128
BK1~ & HS*+	229	329	429	529	629	129
BK1~ & HS*~	230	330	430	530	-	130
BK1~ & HS**	231	331	431	531	631	131
BK1* & HS++	232	332	432	532	632	132
BK1* & HS~+	233	333	433	533	-	133
BK1* & HS~~	234	334	434	534	-	134
BK1* & HS*+	235	335	435	535	635	135
BK1* & HS*~	236	336	436	536	-	136
BK1* & HS**	237	337	437	537	637	137
BK1+ & LS1+	-	341	441	541	641	-

Model	A3BR	A3BL	A3BE	A3BD	3AR	A3BN
BK1+ & LS1~	-	342	442	542	-	-
BK1+ & LS1*	243	343	443	543	643	143
BK1~ & LS1+	-	344	444	544	-	-
BK1~ & LS1~	-	345	445	545	-	-
BK1~ & LS1*	246	346	446	546	-	146
BK1* & LS1+	-	347	447	547	647	-
BK1* & LS1~	-	348	448	548	-	-
BK1* & LS1*	249	349	449	549	649	149
BK1+ & HS--	281	381	481	581	-	181
BK1+ & HS%%	282	382	482	582	-	182*
BK1+ & HS*%	283	383	483	583	-	183
BK1~ & HS--	284	384	484	584	-	184
BK1~ & HS%%	285	385	485	585	-	185*
BK1~ & HS*%	286	386	486	586	-	186
BK1* & HS--	284	384	484	584	-	184
BK1* & HS%%	285	385	485	585	-	185
BK1* & HS*%	286	386	486	586	-	186

Designed cases (700-999)

Designed cases are created to test common and edge cases. They use the previous notation to describe the modell, but a more explicit way to express the field distribution.

- **field distribution:** Comma separated list of the incoming fields, indicating their distribution. <letters of target tables>[:<letters of relations>]
 - Business key in a hub: capital Letters from A to D
 - Data column in a satellite of a hub: lower case letter of the hub
 - Dependent child key: Capital Letters of the link in square brackets
 - Data column in a satellite of a link: lower case letter in square brackets
 - Relations declaration is uses small letters from T to W separated from the table after a colon
 - A,a,B,b = 4 Fields, each mapped to one of the tables (Hub A, Sat of Hub A, Hub B, Sat of Hub B)
 - AB,A,B,a,b,[ab] = 1 Feld used in Hub A and B, all other are separated. "[ab]" = Satellite on the Link of A and B.
 - ABC,A,B:t,B:u,C,a = 1 Field used in all hubs (ABC), 1 field exclusively in A hub, 1 field for B hub in relation t, one field for b hub in relation u, one field for C hub, one field in Sattelitte of A hub.

Model	field distribution	test
-------	--------------------	------

Model	field distribution	test
ABC	ABC,A,B,C,a,b,c	701
A2BCR	ABC,A,B:t,B:u,C,a,b,c	702
ABCL+ABL	ABC,A,B:t,B:u,C,a,b,c	703
A2BR+ACL	ABC,A,B:t,B:u,C,a,b,c	704