Validating granular XBRL-data

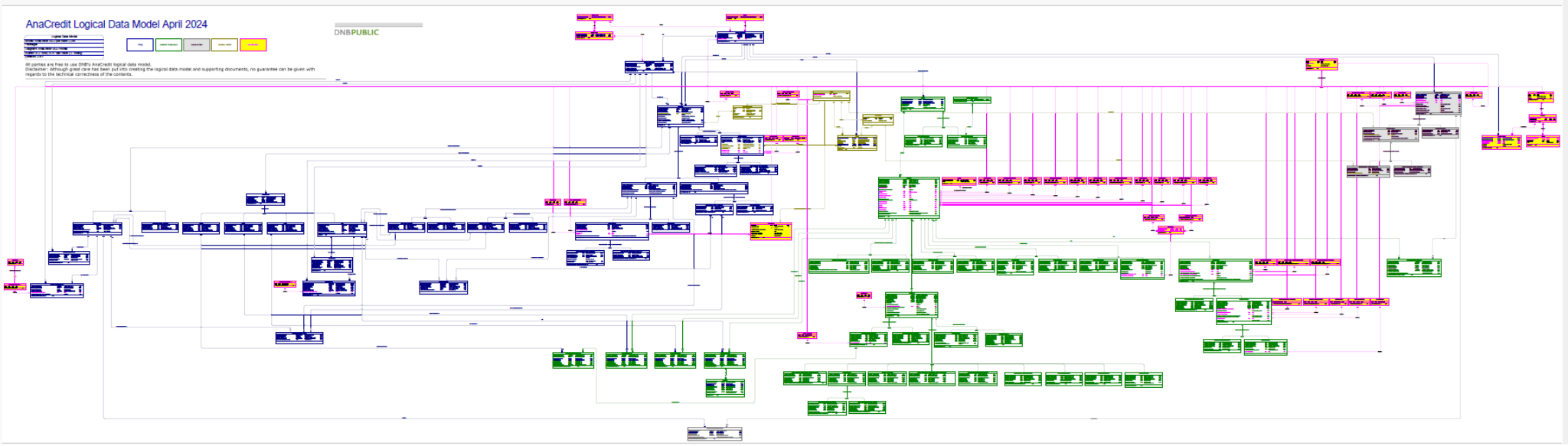
- Paul Hulst & Mark Goodhand
- De Nederlandsche Bank / CoreFiling

DeNederlandscheBank
EUROSYSTEEM

CoreFiling

Reporting becomes granular

- Regulators need more data to perform their tasks.
  They move towards granular, broad data sets that allow many usages replacing narrower, topic specific forms.
- Granular datasets consists of tables, each containing multiple facts about an item (e.g. a loan).
  Many of these tables are related to other tables (e.g. a loan to a party receiving the loan).



DNB logical model for AnaCredit consists of 50+ entities. All entities are open tables having primary keys and mandatory “columns”, most are related to other entities.

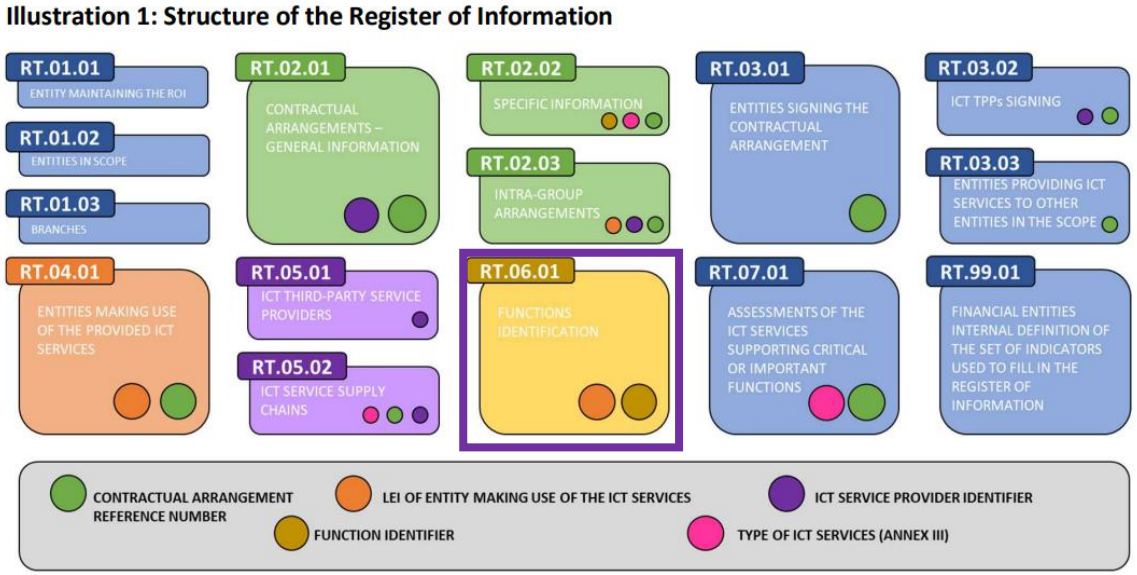
Reporting becomes granular and XBRL must keep up

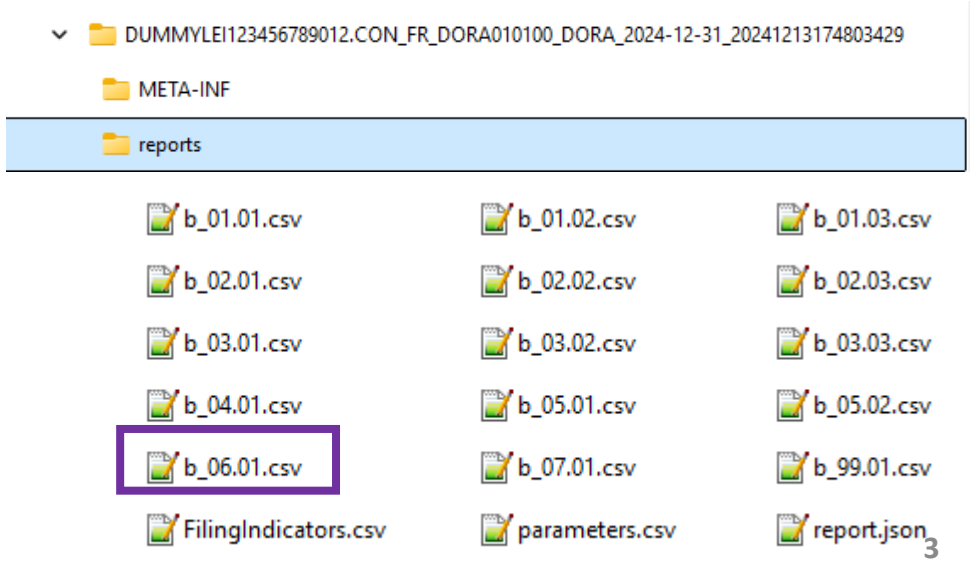
Typical requirements for granular datasets are mandatory facts, unique keys and referential integrity.

These kind of checks are nearly impossible to execute using XBRL Formula on large dataset.
Main reason: XBRL Formula works on finding individual facts in an unstructured collection.

Table Constraints builds on the grouping of facts in meaningful containers offered by xBRL-CSV.

Reading the data by “table” and “row-by-row” makes them “easy” to run.





TC Example – 1 – DORA B_06-01 “Functions”

B_06.01 - Functions identification

		Columns									
		Function Identifier	Licenced activity	Function name	LEI of the financial entity	Criticality or importance assessment	Reasons for criticality or importance	Date of the last assessment of criticality or importance	Recovery time objective of the function	Recovery point objective of the function	Impact of discontinuing the function
		0010	0020	0030	0040	0050	0060	0070	0080	0090	0100
Rows	Open Rows	480217 <Identifier of the operational/business function of an entity>	485465 [Type of licenced activity]	487452 text	480201 <Lei code of the entity making use>	485467 [Function is critical or important?]	485468 text	485466 yyyy-mm-dd	487474 #	487475 #	485464 [Impact of function discontinuing]
	Unique key	B_06_01			B_06_01						
	Mandatory column		yes	yes		yes		yes	yes	yes	yes
	Column constraint - datatype	string	string	string	string	boolean	string	date	integer	integer	string
	Column constraint - value	minLength, maxLength	enumerationValues	minLength, maxLength	patterns		enumerationValues		minExclusive	minExclusive	enumerationValues

If a value exists for c0030, .. Then c0020 must exist as well.

C0040 must follow the LEI pattern.

VR Code	Expression
e23682_e	with {tB_06.01, default: null, interval: false}: not (isnull ({{c0020, c0030, c0050, c0070, c0080, c0090, c0100}}))
v8877_m	with {tB_06.01, default: null, interval: false}: if (not (isnull ({{c0030}}))) or not ((isnull ({{c0050}}))) or not ((isnull ({{c0060}}))) or not ((isnull ({{c0070}}))) or not ((isnull ({{c0080}}))) or not ((isnull ({{c0090}}))) or not ((isnull ({{c0100}}))) then ((not (isnull ({{c0020}}))) endif
v8878_m	with {tB_06.01, default: null, interval: false}: if (not (isnull ({{c0020}}))) or not ((isnull ({{c0050}}))) or not ((isnull ({{c0060}}))) or not ((isnull ({{c0070}}))) or not ((isnull ({{c0080}}))) or not ((isnull ({{c0090}}))) or not ((isnull ({{c0100}}))) then ((not (isnull ({{c0030}}))) endif
v8879_m	with {tB_06.01, default: null, interval: false}: if (not (isnull ({{c0030}}))) or not ((isnull ({{c0020}}))) or not ((isnull ({{c0060}}))) or not ((isnull ({{c0070}}))) or not ((isnull ({{c0080}}))) or not ((isnull ({{c0090}}))) or not ((isnull ({{c0100}}))) then ((not (isnull ({{c0050}}))) endif
v8880_m	with {tB_06.01, default: null, interval: false}: if (not (isnull ({{c0030}}))) or not ((isnull ({{c0050}}))) or not ((isnull ({{c0060}}))) or not ((isnull ({{c0020}}))) or not ((isnull ({{c0080}}))) or not ((isnull ({{c0090}}))) or not ((isnull ({{c0100}}))) then ((not (isnull ({{c0070}}))) endif
v8881_m	with {tB_06.01, default: null, interval: false}: if (not (isnull ({{c0030}}))) or not ((isnull ({{c0050}}))) or not ((isnull ({{c0060}}))) or not ((isnull ({{c0070}}))) or not ((isnull ({{c0020}}))) or not ((isnull ({{c0090}}))) or not ((isnull ({{c0100}}))) then ((not (isnull ({{c0080}}))) endif
v8882_m	with {tB_06.01, default: null, interval: false}: if (not (isnull ({{c0030}}))) or not ((isnull ({{c0050}}))) or not ((isnull ({{c0060}}))) or not ((isnull ({{c0070}}))) or not ((isnull ({{c0080}}))) or not ((isnull ({{c0020}}))) or not ((isnull ({{c0100}}))) then ((not (isnull ({{c0090}}))) endif
v8883_m	with {tB_06.01, default: null, interval: false}: if (not (isnull ({{c0030}}))) or not ((isnull ({{c0050}}))) or not ((isnull ({{c0060}}))) or not ((isnull ({{c0070}}))) or not ((isnull ({{c0080}}))) or not ((isnull ({{c0020}}))) or not ((isnull ({{c0020}}))) then ((not (isnull ({{c0100}}))) endif
v8897_m	match({tB_06.01, c0020}[get LES], "^[A-Z0-9]{18}[0-9]{2}$")

These validation rules currently expressed in XBRL formula can be replaced by Table Constraints.

TC Example – 2 – DORA B_02-02 “Contractual arrangements – specific information”

B_02.01 - Contractual arrangements – General Information

		Columns				
		Contractual arrangement reference number	Type of contractual arrangement	Overarching contractual arrangement reference number	Currency of the amount reported in RT.02.01.0050	Annual expense or estimated cost of the contractual arrangement for the past year
		0010	0020	0030	0040	0050
Rows	Open Rows	480197 <Contractual arrangement reference number between the financial entity or, in case of a group, the group subsidiary and the direct ICT third-party service provider>	483167 [Type of contractual arrangement]	487321 text	487512 [Code of the currency]	483144 €£$

B_02.02 - Contractual arrangements – Specific information

		Columns								
		Contractual arrangement reference number	LEI of the entity making use of the ICT service(s)	Identification code of the ICT third-party service provider	Type of code to identify the ICT third-party service provider	Function identifier	Type of ICT services	Start date of the contractual arrangement	Sensitiveness of the data stored by the ICT third-party service provider	Level of reliance on the ICT service supporting the critical or important function.
		0010	0020	0030	0040	0050	0060	0070	0170	0180
Rows	Open Rows	480197 <Contractual arrangement reference number between the financial entity or, in case of a group, the group subsidiary and the direct ICT third-party service provider>	480201 <Lei code of the entity making use>	480199 [Identification code of the ICT third-party service provider>	3296069 [Type of code to identify the ICT third-party service provider]	480217 <Identifier of the operational/business function of an entity>	480190 <Type of activity>	3291598 yyyy-mm-dd	3291602 [Data sensitivness]	3291601 [Level of reliance on the service]

B_06.01 - Functions identification

		Columns									
		Function Identifier	Licenced activity	Function name	LEI of the financial entity	Criticality or importance assessment	Reasons for criticality or importance	Date of the last assessment of criticality or importance	Recovery time objective of the function	Recovery point objective of the function	Impact of discontinuing the function
		0010	0020	0030	0040	0050	0060	0070	0080	0090	0100
Rows	Open Rows	480217 <Identifier of the operational/business function of an entity>	485465 [Type of licenced activity]	487452 text	480201 <Lei code of the entity making use>	485467 [Function is critical or important?]	485468 text	485466 yyyy-mm-dd	487474 #	487475 #	485464 [Impact of function discontinuing]

Specific information for contractual arrangements refers to (1) contractual arrangements general and to (2) functions.

Referential integrity can be enforced using Table Constraints.

Table Constraints 1.0

- Capabilities:
  - Checks on xBRL-CSV file level:
    - Unique key and reference key
    - Mandatory columns
  - Checks on xBRL-CSV column level:
    - Datatype
    - Value constraints: pattern, restricted list, min/max values, ..

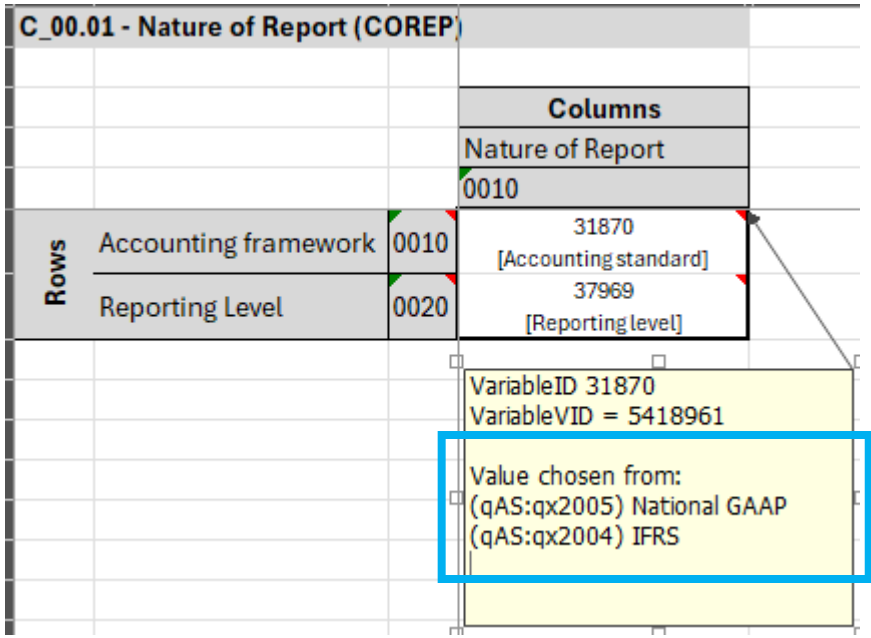
- Status at the end of May 2026 is Proposed Recommendation.
- Recommendation expected by 1 August 2026.

Table Constraints – next steps

- Extend to constrain data defined using property groups.
  EBA uses this approach extensively for their reporting.
- Cover simple validation rules as well

- Target dates for this specification
  - Candidate Recommendation at end of 2026.
  - Recommendation at end of 2027.

Table Constraints next – 1 – property group support




```
{
    "documentInfo": {
    "tableTemplates": {
        "C_00-01": {
            "columns": {
                "datapoint": {
                    "propertyGroups": {
                        "dp31870": {
                            "dimensions": {
                                "concept": "eba_met_4.2:qAST"
                            },
                            "tc:constraints": {
                                "type": "xs:string",
                                "enumerationValues": [ "eba_qAS:qx2004", "eba_qAS:qx2005" ]
                            },
                            "eba:documentation": {
                                "AllowedValue": {
                                    "eba_qAS:qx2004": 1012409106,
                                    "eba_qAS:qx2005": 1012409107
                                },
                                "...":"..."
                            }
                        },
                        "dp37969": {
                        }
                    },
                    "factValue": {
                        "dimensions": {},
                        "propertiesFrom": [
                            "datapoint"
                        ]
                    }
                }
            }
        }
    }
}
```

Please note, just to illustrate concept

Table Constraints next – 2 – simple validation rule support

- Most validation rules work on data in a single template.
  Being able to execute them on the only the data for that template instead of all data in the report will shorten the time required for processing considerably.
- Examples:
  - DNB MSR contains 273 validation rules of which 240 (87%) are single template based.
    DNB MESRAP contains 693 validation rules of which 583 (78%) are single template based.

TOC | T02.02 - Holdings in investment fund shares or units (NON ISIN)

Type of security identifier	Security identifier	Type of instrument	Type of investment fund	LEI	Issuer country	Issuer sector	Issuer name	Positions and transactions								Income
								Position at the beginning of the month (market value)	Changes during the month						Position at the end of the month (market value)	Dividend amount received during the month
									Transactions		Revaluation		Other changes (market value)	Rectifications (market value)		
									Purchases (transaction value)	Sales (transaction value)	Exchange rate changes	Price changes				
c001	c002	c010	c020	c030	c040	c050	c060	c070	c080	c090	c100	c110	c120	c130	c140	c150

if ($c010 = 'Money market fund shares/units' then ($c020 = 'Other') else (true())	T02.02: When the intrument type is 'Money market fund shares/units', 'Type of investment fund' must be 'Other'.
if (matches($c030, '^[A-Z0-9]$18[0-9]2')) then (lei-fn:validate-checksum($ c030)) else (false())	T02.02: LEI code is invalid.
if (empty($c030)) then (not(empty($c040)) and not(empty($c050))) else (true())	T02.02: When no LEI code is entered, the columns Issuer country and Issuer sector are mandatory.
if ($c010 = 'Money market fund shares/units' and empty($c030)) then ($c050 = 'Money market funds' else (true())	T02.02: When Type of instrument is 'Money market fund shares/units' and no LEI is entered, 'Issuer sector' must be 'Money market funds'.
if ($c010 = 'Non-MMF investment fund shares/units' and empty($c030)) then ($c050 = 'Non MMF investment funds' else (true())	T02.02: When Type of instrument is 'Non-MMF investment fund shares/units' and no LEI is entered, 'Issuer sector' must be 'Non MMF investment funds'.
if (empty($c030) and $c040 = 'Netherlands') then (not(empty($c060))) else (true())	T02.02: When no LEI code is entered and Issuer country is 'Netherlands' then Issuer name is mandatory.
$c080 >=0	T02.02: Purchases (transaction value) must be greater than or equal to zero.
$c090 >=0	T02.02: Sales (transaction value) must be greater than or equal to zero.
$c150 >=0	T02.02: Dividend amount received during the month can only be negative in exceptional cases.
iaf:numeric-equal(iaf:numeric-subtract(iaf:sum(($c070,$c080,$c100,$c110,$c120,$c130)),$c090),$c140)	T02.02: Sum of(beginning balance, purchases, exchange rate changes, price changes, other changes, rectifications) - sales should be equal to ending balance.
if($c040='Netherlands' then($c001='RIAD') else(true())	T02.02: When the issuer country is 'Netherlands', 'Type of security identifier' must be 'RIAD'

Table Constraints next – 2 – simple validation rule support - example

TOC T02.02 - Holdings in investment fund shares or units (NON ISIN)

Type of security identifier	Security identifier	Type of instrument	Type of investment fund	LEI	Issuer country	Issuer sector	Issuer name	Positions and transactions								Income
								Position at the beginning of the month (market value)	Changes during the month						Position at the end of the month (market value)	Dividend amount received during the month
									Transactions		Revaluation		Other changes (market value)	Rectifications (market value)		
									Purchases (transaction value)	Sales (transaction value)	Exchange rate changes	Price changes				
c001	c002	c010	c020	c030	c040	c050	c060	c070	c080	c090	c100	c110	c120	c130	c140	c150

if (empty($c030)) then (not(empty($c040)) and not(empty($c050))) else (true()) — T02.02: When no LEI code is entered, the columns Issuer country and Issuer sector are mandatory.

```
{
    "...": "...",
    "tableTemplates": {
        "t02.02": {
            "tc:rowAssertions": [
                {
                    "test": "if( empty( $c030 ) ) then (not( empty( $c040 ) ) and not( empty( $c050 ) ) else ( true() )",
                    "severity": "error",
                    "message": [
                        {
                            "language": "en",
                            "text": "T02.02: When no LEI code is entered, the columns Issuer country and Issuer sector are mandatory."
                        }
                    ]
                }
            ]
        }
    }
}
```

Please note, just to illustrate concept

Questions on Table Constraints?