IMPLEMENTING LOGICAL DATA TYPES

During the database development process, as we move from data modeling to database engineering, we want to translate the logical data types used in the logical data model document into columns (simple logical data types) and sets of columns (complex logical data types) that can be implemented in the DBMS.

We know that the logical data types are not the data types used in the DBMS implementation of Codd's Relational Data Model since these DBMS data types are the physical data types, strings of bits that make up text and numbers. The only logical data type implemented in the DBMS is datetime which allows us to record and manipulate the clock and the calendar as defined by the BIPM.

The simple logical data types can be easily implemented as a single DBMS data type with straight forward operations. Simple data types are highlighted in blue.

The complex logical data types need to be implemented as structures (multiple DBMS columns and data types) with complex operations required when manipulating and comparing the structures. Complex data types are highlighted in yellow and are sequences of simple data types.

There is a CASE tool (Visible Analyst) that will allow you to define logical data types as domains (for simple logical data types) and structures (for complex logical data types) and use these domains and structures to define the logical data model. The output of the logical data model is an SQL create script that becomes the foundation of the physical data model.

When implementing the logical data types, the column naming convention becomes very important. The naming convention should be able to be used correctly in the DBMS's from various manufacturers so that changing DBMS manufacturers does not entail refactoring the database.

The lengths of the strings used in the implementations reflects the structure of the string in the DBMS (string length byte plus string data bytes) and whole fractions of the data sector on the disk (256 bytes). So, the longest sting is 255 bytes or one sector, then 127 bytes or half a sector, then 63 bytes or a quarter sector, etc.

The implementations of the logical data types are given in Table 1.

Logical Data Type	Component Description	Component Name	Physical Name	Default Value	Physical Data Type	Validation List Required
monetary amount	The value of the monetary amount.	amount	<attribute_name>_AMT	0	Decimal(21,2)	No
monetary amount	The currency of the monetary amount	currency	<attribute_name>_CURR_CD	'0'	Nvarchar(4)	Yes
monetary amount	the frequency at which the monetary amount is paid	frequency	<attribute_name>_FREQ_CD	'0'	Nvarchar(4)	Yes
measured quantity	The value of measurement that was made.	quantity	<attribute_name>_QTY	0	Decimal(16,5)	No
measured quantity	unit of measure of the measurement	unit of measure	<attribute_name>_UOM_CD	'0'	Nvarchar(4)	Yes
location (as an address)	suite or apartment	suite	<attribute_name>_APT	NULL	Nvarchar(15)	No
location	street address	street	<attribute_name>_STR	NULL	Nvarchar(63)	No
location	name of city	city name	<attribute_name>_CITY	NULL	Nvarchar(63)	No
location	name of state	state	<attribute_name>_ST	NULL	Nvarchar(63)	No
location	name of country	country	<attribute_name>_CTRY	NULL	Nvarchar(63)	No
calendar date	A date in the Gregorian calendar	calendar date	<attribute_name>_DT	Today()	Date	No
clock time	The time of day within a date	clock time	<attribute_name>_TM	Now()	Time	No
event timestamp	the date and time of a recorded event	event timestamp	<attribute_name>_TMSTP	Now()	Timestamp	No
time interval	start of time interval	start date	<attribute_name>_STRT_DT	'01-JAN-1900'	Date	No
time interval	end of time interval	end date	<attribute_name>_END_DT	'31-DEC-9999'	Date	No

Table 1a. Measurement Logical Data Type Physical Implementations

Logical Data Type	Component Description	Component Name	Physical Name	Default Value	Physical Data Type	Validation List Required
enumerated	list of values	enumerated	<attribute_name>_CD	'0'	Nvarchar(4)	Yes
indicator	Question answer (yes, no, etc.)	indicator	<attribute_name>_IND	'0'	Char(1)	Yes
address	addressee name	name	<attribute_name>_ADDRE_NM	NULL	Nvarchar(15)	No
address	suite or apartment	suite	<attribute_name>_APT	NULL	Nvarchar(15)	No
address	street address	street	<attribute_name>_STR	NULL	Nvarchar(63)	No
address	name of city	city name	<attribute_name>_CITY	NULL	Nvarchar(63)	No
address	name of state	state	<attribute_name>_ST	NULL	Nvarchar(63)	No
address	postal or zip code	post code	<attribute_name>_PSTL_CD	NULL	Nvarchar(15)	No
address	name of country	country	<attribute_name>_CTRY	NULL	Nvarchar(63)	No
document	the application that created the document	application id	<attribute_name>_APPL_CD	'0'	Nvarchar(4)	Yes
document	The document identification string	document id	<attribute_name>_ID	'0'	Nvarchar(63)	No
document	the line number in the document	line number	<attribute_name>_LN_NBR	0	Integer	No
identification	the type of identification, e.g. Key, drivers license, passport, etc.	identification type	<entity_name>_ID_CD	'0'	Nvarchar(4)	Yes
identification	the name of the authority that issued the identification	identification owner	<entity_name>_ID_OWNR_CD	'0'	Nvarchar(4)	Yes
identification	the identification string	identification number	<entity_name>_ID_NBR	'0'	Nvarchar(63)	No

Logical Data Type	Component Description	Component Name	Physical Name	Default Value	Physical Data Type	Validation List Required
person name	the courtesy title for the person	title	<attribute_name>_TITL_CD	'0'	Nvarchar(4)	Yes
person name	first name	first name	<attribute_name>_FIRST_NM	NULL	Nvarchar(63)	No
person name	middle name	nickname	<attribute_name>_NK_NM	NULL	Nvarchar(63)	No
person name	nick name	middle name	<attribute_name>_MDL_NM	NULL	Nvarchar(63)	No
person name	surname	last name	<attribute_name>_LAST_NM	NULL	Nvarchar(63)	No
phone number	the country portion of the phone number	country code	<attribute_name>_CTRY_CD	'1'	Nvarchar(7)	No
phone number	the area portion of the phone number	area code	<attribute_name>_AREA_CD	NULL	Nvarchar(7)	No
phone number	the telephone exchange	exchange code	<attribute_name>_EXCH_CD	NULL	Nvarchar(7)	No
phone number	the station number	station code	<attribute_name>_STATN_CD	NULL	Nvarchar(7)	No
phone number	the extension of the phone	extension	<attribute_name>_EXT	NULL	Nvarchar(7)	No
ordinal	count or sequence number	ordinal	<attribute_name>_CNT	0	Integer	No
unstructured	use the component of the required length	unstructured code	<attribute_name>_UNSTR	'0'	Nvarchar(4)	No
unstructured		unstructured eighth	<attribute_name>_UNSTR	NULL	Nvarchar(31)	No
unstructured		unstructured full	<attribute_name>_UNSTR	NULL	Nvarchar(255)	No
unstructured		unstructured half	<attribute_name>_UNSTR	NULL	Nvarchar(127)	No
unstructured		unstructured quarter	<attribute_name>_UNSTR	NULL	Nvarchar(63)	No
unstructured		unstructured short	<attribute_name>_UNSTR	NULL	Nvarchar(15)	No

Logical Data Type	Component Description	Component Name	Physical Name	Default Value	Physical Data Type	Validation List Required
row meta data	row meta data is a required data type for each entity in the database	create date	RMD_CREAT_DT	Now()	Date	No
row meta data		create process	RMD_CREAT_PRCS	'UNSIGNED'	Nvarchar(63)	No
row meta data		data owner group	RMD_DATA_OWN_GRP	'0'	Nvarchar(4)	No
row meta data		restriction level	RMD_RSTR_LVL	0	Integer	No
row meta data		in use	RMD_IN_USE_IND	'Y'	Char(1)	Yes
row meta data		production date	RMD_PRODTN_DT	NULL	Date	No
row meta data		modified date	RMD_MODF_DT	Now()	Date	No
row meta data		modified process	RMD_MODF_PRCS	'UNSIGNED'	Nvarchar(63)	No

Table 1b. Descriptive Logical Data Type Definitions

The last column of the table is “Validation List Required”. For the simple data types “enumerated” and “indicator” this column contains a Yes and is highlighted. This means that a lookup/validation/reference table (ORACLE) or a domain (SQL SERVER) is require to ensure that the entered data is one value from the list of valid values for that column.

Indicator Lookup Table Description

The logical model for the indicator lookup table is shown below

Indicator Lookup Table
indicator
description [AK1] sort order row meta data

With the table key of indicator and an alternate key of description.

In the database, a column in the row meta data stores the answer to the question “Has this row of data been moved to production?” The recorded answer is one of four valid values:

1. System Default – no data was entered into the column
2. Unknown – the application does not know the answer to the question that was asked
3. Yes – the application gave an affirmative answer to the question
4. No – the application gave a negative answer to the question.

The indicator lookup table contains the list of these four valid values and referential integrity ensures that the answer stored in the field is valid.

Enumerated Data Type Lookup Table Description

To discuss the details of the enumerated data type lookup table, we will use currency as the example. The logical model for the currency code lookup table is shown below

Currency Code Lookup Table
currency
name [AK1] description sort order time interval row meta data

With the table key of currency and an alternate key of name. The time interval gives the start and end dates during which the currency code is valid.

In the database, a column in the monetary amount stores the answer to a question “What is the currency for this monetary amount?” The recorded answer is one of the values given at https://www.xe.com/iso4217.php and referential integrity ensures that the answer stored in the field is valid.