<epam>

Datalogical Modelling Techniques and Tools

Relational Databases Basics

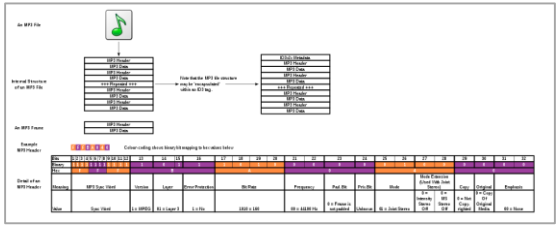
TRAINING
CENTER

<epam>

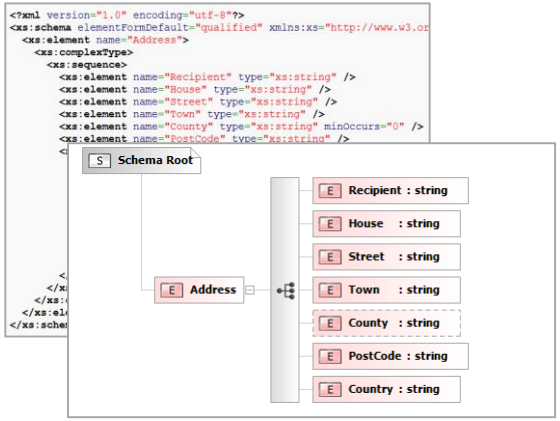
Datalogical level: representation types

Level	
Logical	Conceptual (infological)
	Logical (datalogical)
Physical	

Data specifications



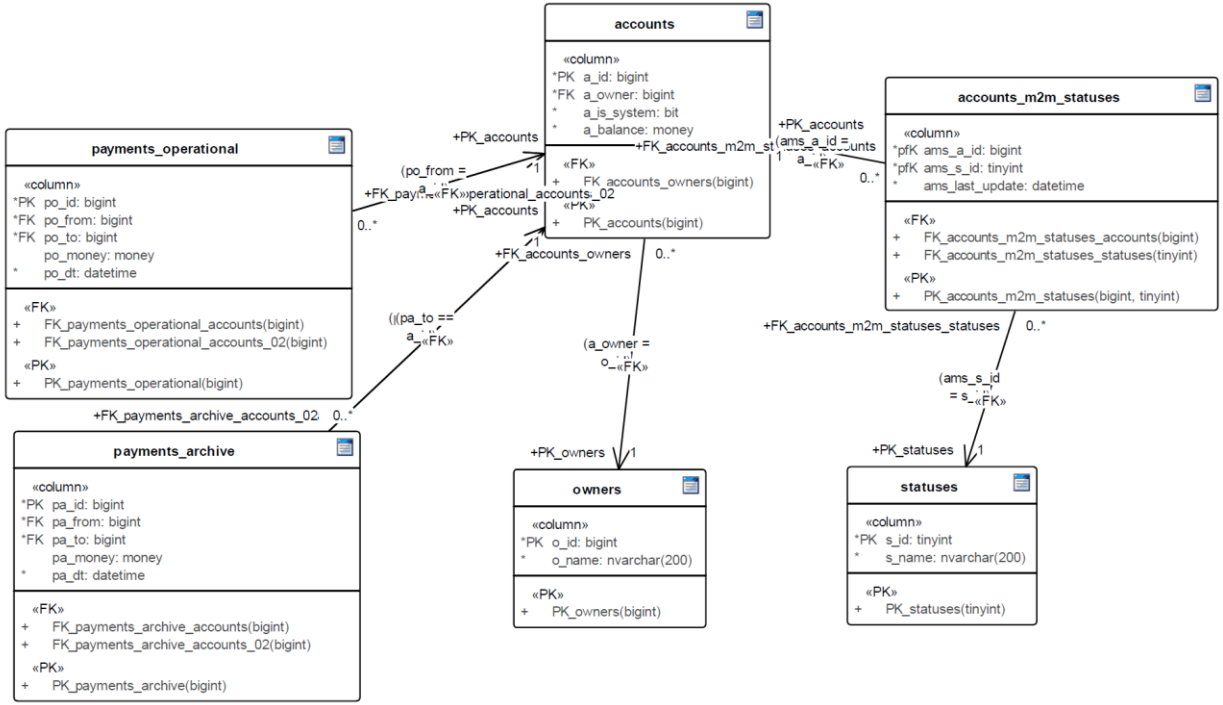
Special formats



Graphical representation (schemas)

Datalogical level: UML schemas

Level	
Logical	Conceptual (infological)
	Logical (datalogical)
	Physical



Necessary knowledge

In ideal situation…	At least…
Deep relational theory understanding	Basic relational theory understanding
Deep normalization understanding	Basic understanding of 1-3 normal forms
Deep SQL understanding	Basic SQL understanding
Deep knowledge of selected DBMS	Basic knowledge of selected DBMS
Deep modelling tool knowledge	Basic modelling tool knowledge

Before you begin modelling

DBMS convention	DB convention
DBMS type	Structures naming
Particular DBMS	SQL code formatting
Minimal DBMS version	Comments principles
DBMS infrastructure details	Other specifics (like API, and so on)

Specific tools for each DBMS

- MySQL Workbench
- SQL Server Management Studio
- SQL Developer Data Modeler
- ...

Universal tools

- Sparx Enterprise Architect
- DbSchema
- DbDiagram.io
- ...

Why universal tools?

No “direct DB manipulations” (zero chance to break DB)

Multiple DBMS support (easy to compare and convert)

Most “specific tools” weak spots are eliminated

User friendly interface, good support, and so on…

<epam>

Datalogical Modelling Techniques and Tools

Relational Databases Basics

TRAINING
CENTER

<epam>