**DATE: 31 July 2025**

**DAY: Thursday, Day: #9**

**AGENDA (If Any):**

* **MQL**

**CONCEPT 1:**

## **What is MQL?**

**MQL (Matrix Query Language)** is a proprietary command-line language used in the Matrix/ENOVIA platform for:

* Querying and managing schema
* Creating business objects
* Manipulating attributes, policies, states, relationships
* Controlling access
* Executing programs like JPOs

Think of it as **SQL + shell script + devops + god-mode**, all in one.

## **🧩 MQL Learning Agenda**

### **1️⃣ Basics: Syntax & CLI Operations**

* help
* list, print, add, delete, modify
* Chaining and scripting MQL commands
* Output formats and redirection
* Comments (#, //)

### **2️⃣ Schema Management (Foundation Layer)**

#### **📦 Types**

* add type MyType
* modify type MyType ...
* print type MyType select ...
* add attribute to type

#### **🧬 Attributes**

* add attribute Color type string default Red
* modify attribute Color range add Green
* print attribute Color

#### **🔗 Relationships**

* add relationship Connects type from TypeA to TypeB
* modify relationship
* print relationship

#### **🛡️ Policies**

* add policy MyPolicy type MyType
* add state, add trigger, add signature
* modify policy, delete state
* Lifecycle states & triggers

#### **🏰 Vaults, Formats, Interfaces**

* add vault MyVault
* add format
* add interface

### **3️⃣ Business Objects (Data Layer)**

* add bus <type> <name> <revision> [policy] [vault]
  + add bus Part ABC123 A policy PartPolicy vault eServiceProduction
* modify bus
* delete bus
* connect bus
* print bus Part ABC123 A select description current dump "|"

### **4️⃣ Access Control**

* add role, add group
* grant, revoke
* modify role, modify group
* print role Admin select person
* modify person add role, remove role
* add access in policies

### **5️⃣ Triggers and Programs**

* add program MyJPO java
* add trigger
  + add trigger PromoteCheck check program MyCheckJPO
  + add trigger CreateAction action program MyActionJPO
* print trigger
* modify program, enable/disable trigger
* write program MyProgram input "java code"

### **6️⃣ Queries and Selectables**

* print bus \* \* \* where "attribute[Part Number] == 123"
* select name, revision, attribute[Part Number]
* print connection, print table, print form
* expand bus, print structure

### **7️⃣ File & Route Operations**

* checkin, checkout, unlock
* add route
* connect document
* print route, print document

### **8️⃣ MQL Scripting (Batch or Embedded)**

* Scripting in .mql files
* exec program MyMQLScript
* Use inside JPOs (via MqlUtil.mqlCommand())

## **🧰 Tools You’ll Use**

* **MQL Console in Thick Client** (Tools > MQL)
* **TCL Scripts** using mql command
* **Program Objects** for storing JPO or TCL
* **Enovia Studio / Business Modeler** for admin + MQL combo

## **🎯 Deliverables You Should Practice**

1. Create a custom Type, Attribute, Relationship, and Policy
2. Add a sample Business Object and connect it with relationships
3. Simulate a Lifecycle with states and triggers
4. Query using where, select, and wildcards
5. Write and execute a basic JPO and attach it to a trigger

Shall I start you off with hands-on labs, one section at a time starting with **Schema creation**? Or want a full PDF/cheatsheet of all the above?

##### **You said:**

is there anything inline to practise these queries

###### **ChatGPT said:**

Absolutely, Mr. Stark. You’ve got two main battlegrounds to **practice MQL** — one inside ENOVIA itself, and one simulated.

## **🧪 PRACTICING MQL IN A LIVE ENVIRONMENT**

### **✅ Option 1: ENOVIA (Thick Client or Web)**

Most production and sandbox environments provide an **MQL Console**, usually accessible from:

**Tools → MQL Console** or matrix.exe CLI on Windows

You can:

* Run MQL interactively.
* Chain multiple commands.
* Get real-time schema feedback/errors.
* Load .mql scripts for batch execution.

💡 *Ideal if you already have access to your dev/test environment with admin creds.*

### **🖥️ Option 2: ENOVIA Business Modeler Studio (BMS)**

Business Modeler includes a built-in **MQL execution window** under:

**Tools → Execute MQL / TCL**

You can test:

* Schema commands
* Policy state flows
* Trigger assignments

## **🔧 Option 3: MatrixOne Simulation (Offline Practice)**

If you don’t have an ENOVIA instance handy:

* **Write commands in .mql files** like scripts.
* Organize into sections:
  + 01\_types.mql
  + 02\_attributes.mql
  + 03\_objects.mql

Example:

# Create a Type  
add type MyPart type "Part";  
  
# Add an attribute  
add attribute PartNumber type string multivalued false;  
  
# Attach attribute to type  
modify type MyPart add attribute PartNumber;

🗒 You won’t get real-time output, but this helps you **build muscle memory**.