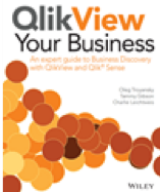


Master Class for QlikView and Qlik Sense Developers - Qlik Your Business

3 days trainings in Vilnius by Oleg Troyanski

March/31-Apr/2, 2020

AGENDA

Day 1: Advanced Data Modelling

Understanding fundamentals of Qlik Data Modeling is the most important skill that's necessary for developing high-performing scalable Qlik applications that can analyze truly large volumes of data. Qlik developers that don't possess this knowledge, may be developing clunky non-optimized, slow-performing applications even after several years of working with Qlik tools.

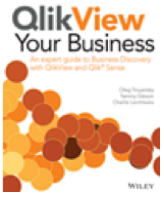
- **How Qlik stores data internally**
  In this section, students learn the most important and often misunderstood concept of internal data storage in Qlik. With this understanding under our belt, we are ready to absorb more advanced Data Modelling techniques.

- **Normalizing or Denormalizing?**
  This question comes up often in Data Modelling classes, and most students are confused about it. In our class, we will define practical rules, based on knowledge – when to normalize and when to denormalize.

- **Data Modelling techniques – Mapping, Joining, Concatenation**
  Most experienced developers already know these techniques, however very few are aware of the tricky and often dangerous behavior of some of them. We cover these techniques to level the field for everyone, and to review some of the unexpected issues related to these techniques.

- **Combining multiple fact tables in a single Data Model**
  This is by far the most challenging Data Model problem that most developers struggle with. Multiple Facts in a single data model cause various problems, including Synthetic Keys and Circular References, and solutions are not always obvious. We present the two commonly used approaches – Link Tables and Concatenated Facts, along with the comparison of advantages and disadvantages of each solution.

- **Link Tables**



Link Tables is the first of the two solutions. We review the methodology and then we build a working Link Table data model based on the provided electronic materials. We learn the process step by step, review any unexpected problems that may occur, and learn methods of overcoming these problems.

- **Concatenated Facts**
  Concatenated Fact is the second commonly used solution for combining multiple fact tables in a single dataset. While it looks quite simple from the first look, in reality this method hides many hidden issues that developers need to anticipate and know how to solve. In the class, we review the method and the anticipated problems, and we build a working Concatenated Facts model, while solving several problems.

- **As of Date tables**
  This is an advanced technique of managing a Calendar in such a way that allows creating analyses "as of" a certain date in the past. In addition to producing "Current Year-To-Date" report, we often need to create a Year-To-Date report as of April 1st, and compare it to the same as of June 1st. These advanced analyses become possible with the use of the "As of Date" table.

In just one day, we cover important fundamentals of advanced Data Modeling, review basic Data Modeling techniques, and then study and build two complex data models that represent common tools in the portfolio of the most advanced Qlik Developers.

Day 2: Advanced Scripting and Performance Tuning

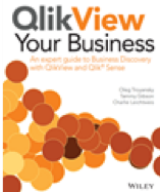
Excellent scripting habits is what differentiates professionals from amateurs. The best Qlik developers use advanced scripting techniques to organized their scripts and make them easier to read, maintain, and understand.

- **Variables**
  Most developers know and use Variables in their work, and yet some aspects related to variables still cause confusion even among experienced specialists. We will review in detail how to use variables effectively in the script in order to create better organized scripts.

- **Script control (loops, if, etc.)**
  Script control statements, such as loops and IF statements, are often overlooked by developers, because they are only needed for creating sophisticated scripting solutions. We will learn these techniques and review how they can be used to simplify developers' work.

- **Subroutines and reusable libraries**
  Using subroutines and reusable libraries can help developers, especially in large organizations, to reuse scripts among multiple applications and create segments of reusable logic that can save a lot of time, money, and effort in the long run.

- **CROSSTABLE load and data transformation for spreadsheets**
  These well known QlikView techniques are often overlooked by QlikView developers, and they are not available in Qlik Sense. Qlik Sense developers will benefit from learning these QlikView techniques and possibly implementing the same techniques in Qlik Sense.



- **Interval Match**
  Interval Match allows linking individual numbers to ranges of numbers with “from” and “to” boundaries. This advanced scripting technique remains confusing for many developers.

- **Storing expressions in variables and loading them in the script**
  This advanced scripting technique is used by many professionals and it offers many distinct advantages, especially for large companies – better reusability of expressions, easier maintenance, source control, and many other advantages. In the class, we will learn the full cycle – how to export formulas into variables, then how to export variables into external text files, and then how to load these text files back and generate variables in the script.

- **Developing a multi-tier data acquisition process**
  All large Qlik implementations face similar problems – database loads are too slow; the same business logic needs to be repeated in multiple applications, potentially creating multiple “versions of truth”. The multi-tier data acquisition process is the recommended solution for large scale implementations with large data volumes and with many Qlik applications. We will review the approach and the specific techniques of implementing it.

Performance Tuning

True professionalism is being tested when a developer is required to create a high-performance application and analyze hundreds of millions of rows with acceptable response times. Most developers are challenged by performance issues because these issues are not understood very well and are not covered well in the available Qlik education materials. We will cover:

- Common performance pitfalls in the layout:
  - What functions are too slow for large data sets and why?
  - What is the impact of data islands?
  - What better alternatives to slow functions can we use?
- Common performance pitfalls in data models
  - What data structures perform better than others?
  - Comparing performance of Link Tables and Concatenated Facts
  - Impact of normalization and denormalization on performance
  - Impact of high cardinality fields on performance and how to reduce it
- Data load performance
  - What’s causing slower data load performance
  - What can be done to speed things up
- Performance Optimization tools
  - Overview of available performance optimization tools
  - How to use Document Analyzer in QlikView and in Qlik Sense for performance tuning.

Day 3: Advanced Expressions – Set Analysis, AGGR, Alternate States

Qlik values simplicity in most things. Simple solutions are extremely easy to grasp and can be created in minutes, even by inexperienced users. However, complex problems require sophisticated solutions, that in turn require advanced analytical techniques that people master over months and years. Two advanced techniques usually cause a lot of confusion and misunderstanding – Advanced Aggregation function AGGR() and Set Analysis, which provides the

ability to alter the scope of aggregated data for each aggregation functions. Most developers remain confused about these techniques even after many years of working with the tools. This detailed hands-on session demonstrates the most advanced analytical solutions with the use of AGGR() and Set Analysis:

- **Advanced Aggregation (AGGR)**

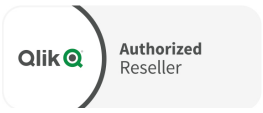
  Using Advanced Aggregation, developers can alter the level of aggregation for a particular calculation – for example, calculating the difference between Sales and Forecast at the Item level prior to aggregating it to the Company level. This function is one of the least understood and the most confusing for many developers. It is not being taught in detail in any of the existing Qlik education classes.

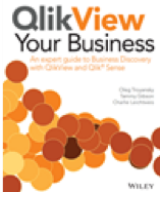
  - Common uses of AGGR()
    - When do we use Advanced Aggregation and for what purpose?
    - Introducing Bucket Analysis and creating Histograms
    - Comparing AGGR() to using the TOTAL qualifier
  - The 3 laws of AGGR()
    - There are 3 rules, or "laws" that should be satisfied in order to use AGGR() safely. Few developers are familiar with all 3 rules, which causes a lot of confusion around AGGR().
    - Counting colors, or aggregating conditional calculations
  - Advanced analytical uses of AGGR()
    - Advanced lookup – looking for top performers in several ways
    - Aggregated filters
  - Recent additions to the syntax of AGGR():
    - Sorting data within AGGR()
    - Sorting by expression

- **Advanced Set Analysis**

  With Set Analysis, developers can alter the set of data that should be aggregated in a certain calculation. Most developers master and use the basics of Set Analysis, however advanced Set Analysis techniques, such as Advanced Search, or using functions P() and E(), remain confusing for many developers. This is the most comprehensive and the most advanced class on Set Analysis across all available education sources.

  - Basic syntax of Set Analysis – a review for less advanced developers
  - Using Boolean operators in Set Analysis
  - Using $-sign expansions in Set Analysis
  - Simple and Advanced Search and what's the difference
  - Using Dates in Set Analysis
  - Using functions P() and E()
  - Set Analysis and AGGR() working together
  - Advanced analytical uses of Set Analysis and AGGR()
    - Limiting chart dimensions to top N without using Dimension Limits
    - Restoring missing links in Concatenated Facts
    - Using AGGR() with Synthetic Dimensions
    - Using Set Analysis for point in time comparisons
    - Using the function Above() and improving it with AGGR()
    - Pareto calculation, using AGGR() and Set Analysis





- **Alternate States**
  Alternate states existed in QlikView since version 11.2, and they were recently re-introduced into Qlik Sense. With Alternate States, developers can compare multiple sets of selections with each other, allowing users to select different values in the same database fields, for the purpose of comparative analysis.
    - What are alternate states and how they can be used?
    - Alternate States and Set Analysis
    - Alternate States and Actions/Variables
    - Creative uses of Alternate States

Conclusion

During the 3 days of intense advanced training, students receive the most vital advanced development techniques for QlikView and Qlik Sense, in four key areas – Data Modeling, Scripting, Performance, and Advanced Calculations. They take home proven developer methodologies and all class materials for future reference.

With the acquired skills, your developers will be developing better high-performing applications, make them faster, more reliable, and more insightful for your business.

Contact:

Lina Kriukeliene
Day Q, marketing and trainings manager
l.kriukeliene@dayq.eu
+370 688 83777
www.dayq.eu