Arcsight Logger Query Cheat Sheet Latest Edition

arcsight logger query cheat sheet is an essential resource for security analysts, SIEM administrators, and anyone involved in managing and analyzing security logs using ArcSight Logger. Whether you are a beginner or an experienced user, having a comprehensive cheat sheet can significantly streamline your workflows, improve query efficiency, and enhance your overall understanding of the platform?s capabilities. This article provides a detailed, SEO-friendly overview of ArcSight Logger queries, including fundamental concepts, common query syntax, best practices, and advanced tips to optimize your security data analysis.

Understanding ArcSight Logger and Its Query Language

What is ArcSight Logger?

ArcSight Logger is a scalable, high-performance log management and SIEM solution designed to collect, store, and analyze security and operational log data from diverse sources. It helps organizations detect threats, comply with regulations, and conduct forensic investigations by providing real-time visibility into their security posture.

What Is the Query Language in ArcSight Logger?

The query language in ArcSight Logger is a powerful, flexible syntax used to search, filter, and analyze log data stored within the system. It allows users to construct complex queries that pinpoint specific events, patterns, or anomalies across vast datasets efficiently.

Basic Components of an ArcSight Logger Query

Field Names

Field names refer to specific data attributes within logs, such as `srcIp`, `destPort`, `eventName`, or `severity`. Understanding field names is fundamental to crafting effective queries.

Operators

Operators define the logic for combining or comparing fields and values. Common operators include:

- = (equals)
- != (not equals)
- ~ (contains)
- !~ (does not contain)
- > (greater than)
-
- AND, OR, NOT (logical operators)

Values

Values are specific data points or patterns you want to find within logs, such as IP addresses, port numbers, or keywords.

Basic Query Syntax

A typical query combines these components in a structured way:

```plaintext

```

For example:

```plaintext

srcIp = "192.168.1.10"

```

Commonly Used Query Patterns and Examples

Filtering by Time Range

To focus on specific periods, use the `startTime` and `endTime` parameters:

```plaintext

(startTime >= "2023-10-01 00:00:00" AND endTime
```

Searching for Specific Event Types

For example, to find all login failures:

```plaintext
eventName = "LoginFailure"
```

Filtering by Severity Levels

If you want to see high-priority security events:

```plaintext
severity >= 4
```

Using Wildcards and Contains Operators

To search for logs containing a particular substring:

```plaintext
message ~ "failed login"
```

Combining Multiple Conditions

Use logical operators to refine your search:

```plaintext
(srcIp = "10.0.0.5" AND eventName = "MalwareDetected")
```

Advanced Query Techniques

Regular Expressions

ArcSight Logger supports regex patterns to match complex string patterns:

```plaintext
message ~ /Error\s\d+/
```

Aggregation and Grouping

While Logger's query language primarily focuses on filtering, aggregation can be performed through the Logger interface or external tools, such as Elasticsearch or Kibana.

Creating Saved Queries

To reuse complex queries, save them within the Logger interface for quick access:

- Use the "Save Query" option.
- Assign descriptive names for easy identification.

Best Practices for Efficient Querying

Optimize Your Queries

- Limit the time range to reduce data processing.
- Use specific field filters rather than broad searches.
- Avoid unnecessary wildcards that can slow down performance.

Use Indexes When Available

Ensure your queries utilize indexed fields for faster search results.

Leverage Query Templates

Create common query templates to standardize searches across different investigations.

Regularly Review and Refine Queries

Continuously optimize your queries based on outcomes and system performance insights.

Commonly Used Query Cheat Sheet

- **Basic Equality:** `field = "value"`
- **Negation:** `field != "value"`
- **Contains:** `field ~ "substring"`
- **Does Not Contain:** `field !~ "substring"`
- **Greater Than / Less Than:** `field > 10`, `field
- **Logical AND:** `condition1 AND condition2`
- **Logical OR:** `condition1 OR condition2`
- **Negation with NOT:** `NOT condition`
- **Time Range:** `startTime >= "YYYY-MM-DD HH:MM:SS"` and `endTime
- **Combining Conditions:** `(field1 = "value1" AND field2 != "value2")`

Integrating Query Results with External Tools

Exporting Data

ArcSight Logger allows exporting query results in various formats such as CSV, JSON, or XML for further analysis.

Using APIs for Automation

Leverage Logger?s REST API to automate queries, integrate with SIEM dashboards, or develop custom alerts.

Visualization and Dashboarding

Integrate query outputs with visualization tools like Kibana or Grafana for enhanced analysis.

Conclusion

Mastering the ArcSight Logger query language is crucial for efficient and effective security log analysis. This cheat sheet provides a foundational understanding, common patterns, and advanced techniques to help you craft precise queries, optimize performance, and extract valuable insights from your logs. Regular practice and continuous refinement of your queries will lead to better incident detection, faster investigations, and improved overall security posture.

Remember, the key to successful log analysis is not just knowing how to write queries but also understanding the underlying data, security context, and operational needs. Use this cheat sheet as a reference guide to enhance your skills and leverage the full potential of ArcSight Logger.

Arcsight Logger Query Cheat Sheet: Your Ultimate Guide to Efficient Log Analysis

In the realm of cybersecurity and enterprise security management, Arcsight Logger Query is an indispensable tool for security analysts, SIEM administrators, and threat hunters. It provides the ability to perform complex searches, generate reports, and analyze logs across vast data stores efficiently. Whether you are new to Arcsight Logger or looking to sharpen your skills, mastering the query language and its nuances can significantly enhance your operational effectiveness. This guide aims to serve as a comprehensive cheat sheet, breaking down the essentials of Arcsight Logger queries, best practices, and tips for maximizing your log analysis capabilities.

Understanding Arcsight Logger and Its Query Language

Before diving into query syntax and examples, it?s important to understand what Arcsight Logger is and how its query language functions.

Arcsight Logger is a scalable log management platform designed to collect, normalize, and analyze security event data from diverse sources. Its query language is designed to be expressive, allowing users to filter, aggregate, and manipulate large datasets efficiently.

The core components of the query language include:

- Filtering: Narrow down logs based on criteria.
- Aggregation: Summarize data to identify patterns.
- Functions: Perform calculations, extractions, and data transformations.
- Time Range Selection: Focus on specific periods.

Basic Structure of an Arcsight Logger Query

A typical query in Arcsight Logger follows a structured format:

```

[additional clauses]

```

For example:

```
```

sourceAddress = "192.168.1.100" AND eventName = "Login Failure"

```

Key elements:

- Fields (e.g., `sourceAddress`, `eventName`, `severity`)
- Operators (e.g., `=`, `!=`, `IN`, `LIKE`, `>`, `
- Values (e.g., IP addresses, strings, numbers)
- Logical operators (`AND`, `OR`, `NOT`)
- Time constraints (`startTime`, `endTime`, or relative periods)

Essential Query Syntax and Operators

Filtering Data

| Operator | Description | Example |

|---|---|---|

| `=` | Equals | `severity = "High"` |

| `!=` | Not equals | `eventName != "Login Success"` |

| `IN` | Within a list | `sourceAddress IN ("192.168.1.1", "10.0.0.5")` |

| `LIKE` | Pattern matching | `eventName LIKE "Login%"` |

| `>` / ` 5` |

| `IS NULL` / `IS NOT NULL` | Null checks | `userName IS NULL` |

Logical Combinations

| Keyword | Description | Example |

|---|---|---|

| `AND` | Both conditions true | `severity = "High" AND eventName = "Malware Detected"` |
```

| `OR` | Either condition true | `sourceAddress = "192.168.1.1" OR sourceAddress = "10.0.0.2"` |

| `NOT` | Negation | `NOT eventName = "Login Success"` |

Time Range Filtering

Time-based queries are fundamental in log analysis:

- Using `startTime` and `endTime` parameters:

```
startTime = "2023-10-01T00:00:00.000Z" AND endTime = "2023-10-02T00:00:00.000Z"
```

- Relative time ranges:

```
startTime = "LAST 24 HOURS"
```

- Combining with other filters:

```
startTime = "LAST 7 DAYS" AND severity = "High"
```

Advanced Query Techniques

- Wildcard and Pattern Matching

Using `LIKE` with wildcards:

- `%` (percent) matches zero or more characters
- `_` (underscore) matches a single character

Examples:

```

eventName LIKE "Login%"

```

Matches all events starting with "Login".

```

userName LIKE "admin_%"

```

Matches usernames like `admin_1`, `admin_A`.

- Nested Conditions

Use parentheses to group conditions:

```

(sourceAddress = "192.168.1.1" AND severity = "High") OR (eventName = "Malware Detected")

```

- Field Functions and Extraction

Arcsight Logger supports functions for extracting or transforming data:

- `lower(field)` / `upper(field)` ? change case
- `substring(field, start, length)` ? extract parts of a string
- `count()` ? aggregate count

- `distinct()` ? get unique values

Example:

```

| count() by eventName

```

Counts events grouped by event name.

Commonly Used Queries for Security Analysis

Detecting Failed Login Attempts

```

eventName = "Login Failure" AND severity >= 3 AND startTime > "LAST 24 HOURS"

```

Identifying Top Source IPs

```

| count() by sourceAddress | sort by count desc | limit 10

```

Unusual Activity ? Multiple Failed Logins Followed by Success

```

(sourceAddress = "X.X.X.X" AND eventName = "Login Failure") AND (time between last failure and success
```

(Note: Use of temporal functions or scripting may be necessary for complex sequences.)

Monitoring Specific Ports or Protocols

```

destinationPort IN (445, 3389) AND eventName LIKE "%Connection%"

```

Tips for Writing Efficient and Effective Queries

- Use specific filters to reduce dataset size early.
- Leverage indices: querying on indexed fields improves performance.
- Limit time ranges: narrow periods to focus analysis.
- Use aggregation functions to summarize large data.
- Test queries incrementally: start simple and add conditions.
- Document complex queries for future reference.
- Combine multiple filters logically to refine results.

Practical Examples and Use Cases

Example 1: Detecting Port Scanning Activity

```sql

destinationPort IN (22, 23, 80, 443) AND eventName LIKE "%Connection%" AND startTime > "LAST 1 HOUR"

```

Example 2: Tracking Data Exfiltration Attempts

```sql

eventName = "Large Data Transfer" AND sourceAddress = "X.X.X.X" AND bytesSent > 1000000

```

Example 3: Finding Suspicious User Account Changes

```sql

eventName = "User Account Modification" AND severity >= 4 AND userName != "admin"
```

```

Final Thoughts and Best Practices

Mastering Arcsight Logger query is crucial for proactive security monitoring and incident response. Focus on understanding your data sources, leveraging the powerful filtering and aggregation capabilities, and continually refining your queries based on evolving threats.

Always validate your queries with small time windows before scaling to broader periods. Document your most useful queries, and consider creating saved searches or alerts for ongoing monitoring.

By adhering to these guidelines, security teams can more effectively identify threats, reduce false positives, and respond swiftly to security incidents, ensuring the integrity and safety of their enterprise environments.

Remember: The key to effective log analysis with Arcsight Logger is not just knowing the syntax but understanding the story your logs tell. Use this cheat sheet as your reference, and continually evolve your skills to stay ahead of threats.

QuestionAnswer What is the basic syntax for creating a query in ArcSight Logger? The basic syntax involves specifying the search criteria within the 'Search' interface, using field names, operators, and values, for example: 'field_name = value'. You can also use advanced query language (AQL) for complex searches. How do I filter logs by date range in ArcSight Logger queries? Use the 'startTime' and 'endTime' parameters or the date filter options within the query interface. For example: 'timestamp between '2023-10-01 00:00:00' and '2023-10-07 23:59:59'. What are some common operators used in ArcSight Logger queries? Common operators include '=', '!=', 'LIKE', 'IN', 'AND', 'OR', 'NOT', '>', '=', '
**Related keywords:** ArcSight Logger, query syntax, command reference, log analysis, event filtering, search operators, report generation, log management, query examples, troubleshooting

## sap query sqvi

**sap query sqvi** is a powerful transaction in SAP that enables users to create, modify, and execute SAP queries efficiently. It is part of the SAP Query toolset, which allows for flexible data extraction and reporting without the need for extensive ABAP programming knowledge. Understanding how to utilize SQVI effectively can significantly enhance your reporting capabilities within SAP, providing quick insights and tailored reports to support business decision-making.
```

Introduction to SAP Query SQVI

SAP Query SQVI (Quick Viewer) is a simplified tool designed for end-users and functional consultants to create ad hoc reports and queries on SAP database tables. Unlike the more advanced SAP Query (SQ01, SQ02, SQ03), which require more detailed setup and authorization, SQVI offers a more straightforward interface suitable for quick and straightforward reporting needs.

Key Features of SQVI:

- User-friendly interface for quick report creation
- Ability to join multiple tables
- Drag-and-drop functionality
- No requirement for deep ABAP knowledge
- Export options to Excel, CSV, or other formats
- Integration with SAP authorization roles

Understanding the SAP Query SQVI Interface

When you access SAP Query SQVI, you will encounter a screen designed for ease of use. Here are the main components:

1. Creation of a New Query

- You start by entering a name for your query.
- Choose whether the query is for a single table or multiple tables (joins).

2. Data Source Selection

- Select the database tables relevant to your report.
- Use the available data dictionary objects.

3. Table Join Configuration

- Define how tables relate to each other via joins.
- Choose between INNER JOIN, LEFT OUTER JOIN, etc.

4. Field Selection

- Select fields to include in the output.
- Arrange fields in the desired order.

5. Selection Criteria

- Specify filters to narrow down data.
- Create user-defined selection screens for input parameters.

6. Output Layout

- Customize the report layout.
- Set sorting and grouping options.

Creating a Query Using SQVI

Here is a step-by-step overview of creating a report with SAP Query SQVI:

Step 1: Access SQVI

- Use transaction code SQVI in the SAP GUI.

Step 2: Enter Query Name

- Provide a meaningful name, e.g., ZSALES_REPORT.

Step 3: Choose Data Source

- Decide whether to base the query on a single table or multiple tables.
- For multi-table queries, select the tables you need.

Step 4: Define Joins

- Establish relationships between tables.
- Ensure that joins are correctly set to avoid data inconsistencies.

Step 5: Select Fields

- Pick the fields you want to display in the report.
- Use the 'Available Fields' list to choose and move them to the 'Selected Fields' area.

Step 6: Set Selection Criteria

- Add filters for fields such as date ranges, customer IDs, etc.
- Create dynamic selection screens if needed.

Step 7: Customize Output Layout

- Arrange fields, define sorting order, and set grouping.
- Save the layout for future use.

Step 8: Execute and Save

- Run the query to view the results.
- Save the query for reuse or scheduling.

Advantages of Using SAP Query SQVI

Utilizing SQVI offers numerous benefits, especially for users who need quick and flexible reports:

- **Ease of Use:** Intuitive interface reduces the learning curve.
- **Flexibility:** Ability to join multiple tables and create complex reports without ABAP.
- **Speed:** Rapid report development for ad hoc analysis.
- **Customization:** Tailor layouts, filters, and grouping to meet specific needs.
- **Cost-effective:** Reduces reliance on ABAP programming for simple reporting tasks.

Limitations and Considerations

While SQVI is a versatile tool, it has some limitations that users should be aware of:

1. Performance Concerns

- Complex joins and large datasets can impact system performance.
- It's advisable to optimize query filters and avoid overly broad data retrieval.

2. Security Restrictions

- Users need appropriate authorization roles (e.g., S_QUERY, S_DATASET).
- Sensitive data access should be controlled.

3. Limited Advanced Features

- For highly complex reporting needs, SAP Query SQ01 (SAP Query), SQ02, and SQ03 might be more appropriate.
- It lacks advanced features like drill-down, calculated fields, or formulas.

4. Maintenance

- As queries proliferate, managing them can become cumbersome.
- Proper documentation and naming conventions are important.

Best Practices for Using SAP Query SQVI

To maximize the benefits of SQVI, consider the following best practices:

- **Plan Your Reports:** Clearly define the data requirements and report layout before starting.
- **Use Consistent Naming:** Adopt naming conventions for queries to facilitate easy retrieval and management.
- **Optimize Selection Criteria:** Apply filters early to reduce data volume and improve performance.
- **Test Thoroughly:** Validate query results with known datasets to ensure accuracy.
- **Document Your Queries:** Keep records of the purpose and configuration for future reference.
- **Leverage User Exits:** For more advanced customization, consider user exits or enhancements.

Integrating SAP Query SQVI into Business Processes

Proper integration of SQVI reports into business operations can streamline decision-making:

- **Report Distribution:** Export reports to Excel, PDF, or email for stakeholders.
- **Scheduling:** While SQVI itself doesn?t support scheduling, reports can be exported manually or automated via background jobs with custom scripting.
- **Role-Based Access:** Assign appropriate authorization roles to control who can create, modify, or execute queries.
- **Training:** Educate end-users on creating and maintaining reports to empower self-service analytics.

Conclusion

SAP Query SQVI is a valuable tool for users seeking quick and customizable reporting solutions within SAP. Its user-friendly interface, flexibility in joining tables, and minimal requirement for ABAP knowledge make it ideal for ad hoc reporting needs. Understanding how to create, optimize, and manage queries effectively can significantly improve data visibility and support strategic business decisions.

By adhering to best practices and being mindful of its limitations, organizations can leverage SAP Query SQVI to enhance their reporting landscape, reduce dependencies on technical teams, and foster a culture of data-driven decision-making.

If you wish to explore more advanced report building within SAP, consider transitioning to SAP Query (SQ01) or SAP Business Warehouse (SAP BW) for enterprise-level analytics. However, for day-to-day quick reports, SQVI remains an accessible and efficient tool for SAP users.

sap query sqvi: Unlocking Data Efficiency with SAP's Quick Viewer Interface

In the complex landscape of enterprise resource planning (ERP) systems, SAP stands out as a robust and versatile platform used by organizations worldwide. Central to SAP's data management capabilities is the ability to access, analyze, and report on vast amounts of information stored within its modules. Among the numerous tools designed to streamline these processes, SAP Query SQVI (QuickViewer) emerges as a user-friendly yet powerful solution, enabling both technical and non-technical users to create ad hoc reports efficiently. This article delves into the intricacies of SAP Query SQVI, exploring its features, functionalities, and best practices to harness its full potential.

Understanding SAP Query SQVI: An Overview

SAP Query SQVI is a simplified reporting tool within the SAP environment that allows users to quickly develop custom reports without the need for extensive ABAP programming knowledge. It is designed for ease of use, enabling users to combine tables, fields, and data sources seamlessly to generate meaningful insights.

What Makes SQVI Stand Out?

- User-Friendly Interface: Intuitive and accessible, even for users with minimal technical background.
- Rapid Report Development: Allows quick creation of reports through a guided process.
- Flexible Data Selection: Supports joining multiple tables and views to tailor reports.
- No ABAP Coding Required: Eliminates the need for programming, making report creation straightforward.
- Integration with SAP Data: Taps directly into SAP's database tables and views for real-time data access.

In essence, SAP Query SQVI bridges the gap between complex SAP reports and the everyday user requirements, democratizing data access across the organization.

The Architecture and Components of SAP Query SQVI

To effectively utilize SAP Query SQVI, understanding its core components and architecture is essential. Here's a breakdown:

- Data Sources

- Tables: Fundamental units where SAP stores data.
- Views: Predefined or custom views that aggregate or simplify access to multiple tables.
- Logical Databases: Predefined data models that organize related data.

- Query Creation

- Users define the data source for their report.
- Select relevant fields that need to be included.
- Specify selection criteria to filter data.
- Define output layout and sorting options.

- Output and Export

- Generated reports can be viewed directly within SAP.
- Data can be exported to Excel, CSV, or other formats for further analysis.

Step-by-Step Guide to Creating Reports with SQVI

Creating reports with SAP Query SQVI involves a series of straightforward steps. This section offers a comprehensive walkthrough.

Step 1: Accessing SQVI

- Use transaction code SQVI in the SAP GUI.
- You will land on the QuickViewer initial screen.

Step 2: Creating a New Query

- Click on Create.
- Assign a meaningful name and description to your query.
- Choose a data source type (e.g., Table Join, Table, or View).

Step 3: Selecting Data Sources

- For most scenarios, you'll select Table Join.
- Specify the primary table and join additional tables as needed.
- Use the SAP Data Browser to locate appropriate tables and views.

Step 4: Defining Fields and Layout

- Select fields to include in your report.
- Arrange fields in the desired order.
- Apply filters or selection criteria to narrow down data.

Step 5: Saving and Executing

- Save your query.
- Click Execute to generate the report.
- Review the output, and refine your query if necessary.

Step 6: Exporting Data

- Use options to export to Excel or other formats for detailed analysis or sharing.

Advanced Features and Best Practices

While SQVI is designed for simplicity, mastering some advanced features enhances its utility.

- Joining Multiple Tables

- Use Table Join queries to combine data from multiple tables.
- Define join conditions carefully to ensure data integrity.
- Utilize SAP documentation and data dictionaries for accurate joins.

- Using Selection Criteria Effectively

- Apply filters to reduce dataset size and improve performance.
- Use dynamic selections for flexible reports.

- Creating Variants

- Save specific parameter settings as variants.
- Reuse variants for consistent reporting.

- Managing Query Performance

- Limit the number of fields retrieved.
- Use appropriate join conditions.
- Avoid unnecessary data loads.

Limitations of SAP Query SQVI and How to Overcome Them

Despite its strengths, SAP Query SQVI has certain limitations that users should be aware of.

- Limited to Basic Reports: Not suitable for complex or highly formatted reports.
- Performance Constraints: Large datasets may impact response times.
- No Advanced Formatting: Limited options for customizing report appearance.
- Dependence on SAP Table Knowledge: Requires understanding of SAP data structures.

Overcoming Limitations:

- For complex reporting needs, consider using SAP Query (SQ01) or SAP BW.
- Use SAP's ABAP development tools for advanced customization.
- Optimize data filters and joins to improve performance.

Practical Use Cases of SAP Query SQVI

Organizations leverage SAP Query SQVI across various domains:

- Finance: Generating accounts payable/receivable reports.
- Material Management: Tracking inventory levels and movements.
- Human Resources: Listing employee details or payroll summaries.
- Sales and Distribution: Analyzing sales order data.

Its versatility makes SQVI an indispensable tool for operational reporting and decision-making.

Training and Resources for SAP Query SQVI Users

To maximize the benefits of SAP Query SQVI, users should pursue ongoing training and utilize available resources:

- SAP Official Documentation: Comprehensive guides and tutorials.
- Online Forums and Communities: SAP Community Network (SCN) offers practical tips.
- Internal Training: Organizations often have specialized training programs.
- Practice: Hands-on experience is the best way to learn its capabilities.

Final Thoughts: Empowering Users with SAP Query SQVI

SAP Query SQVI exemplifies how enterprise systems can be made accessible and adaptable to

user needs without sacrificing power or efficiency. By enabling quick report creation, reducing dependency on technical teams, and fostering data-driven decision-making, SQVI stands as a vital tool within the SAP ecosystem.

As organizations continue to seek agility and responsiveness in their operations, mastering SAP Query SQVI can provide a significant competitive advantage. Whether you are a seasoned SAP user or a beginner, understanding its functionalities will empower you to extract actionable insights swiftly and accurately, ultimately supporting informed business strategies.

In summary, SAP Query SQVI simplifies the complex task of reporting within SAP, offering an intuitive platform for creating tailored reports. Its ease of use, combined with its flexibility, makes it a cornerstone for operational reporting across industries. As part of a broader SAP reporting strategy, SQVI serves as a valuable tool to unlock the potential of enterprise data efficiently and effectively.

QuestionAnswer What is SAP Query SQVI and how is it used? SAP Query SQVI is a quick viewer tool that allows users to create simple ad-hoc reports without extensive programming knowledge. It provides an easy-to-use interface for designing queries by selecting data sources and fields. How do I create a new query using SQVI in SAP? To create a new query with SQVI, access transaction code SQVI, enter a name for your query, choose the data source (tables or join conditions), and then select the fields you want to include in your report. Save and execute the query to view results. Can I join multiple tables in an SAP Query using SQVI? Yes, SQVI allows you to join multiple tables by defining join conditions during the query setup. This enables complex data retrieval across related tables within a single report. What are some common limitations of SAP Query SQVI? SQVI is designed for simple to moderate reporting needs. It has limitations such as lack of advanced formatting, restricted complex joins, and performance issues with very large datasets. For complex reporting, SAP BI or SAP Query (SQ01) may be more suitable. How can I troubleshoot errors encountered while running an SQVI query? Troubleshoot by checking the join conditions, ensuring field selections are correct, verifying data source availability, and analyzing any error messages. Also, review user authorizations and consider testing with smaller datasets. Is it possible to export data from an SQVI query to Excel or other formats? Yes, after executing the query, you can export the results to Excel, CSV, or other formats directly from the SAP GUI by using the standard export options available in the output list.

Related keywords: SAP query, SQVI, SAP QuickViewer, SAP reporting, SAP query tool, SAP report development, SAP query design, SAP data extraction, SAP query customization, SAP report generation

Read the full article online: sap query sqvi