QlikView

QlikView 11.2 SR7 DIRECT DISCOVERY

FAQ and What's New

Published: November, 2012

Version: 6.5

Last Updated: September, 2014

www.qlikview.com

QlikView

1 What's New in Direct Discovery 11.2 SR7?

Direct discovery in SR7 has a number of changes and additions which have been categorized below:

1.1 Multi-Table Support

Direct Discovery can now be used to load more than one table/view and supports ANSI SQL join functionality. A limitation exists in the fact that in a single chart all measures must be derived from the same logical table in QlikView, this could in fact be a combination of tables from source linked via join statements.

- Direct Discovery can be deployed in a single fact/multi-dimension in memory scenario with large datasets.
- Direct Discovery can be used with more than one table **only where the cardinality of the key field in the join is low.**
- Direct Discovery is not suitable for deployment in a third normal form scenario with all tables in Direct Discovery mode.

1.2 New Set Statements

Support has been introduced for Teradata Query banding for use with Direct Discovery statements (Note in memory not supported) a feature allowing information to be logged into a Teradata database, Teradata Query banding provides a mechanism for enterprise applications to collaborate with the underlying Teradata Database in order to provide for better accounting, prioritization, and workload management. Query banding provides the capability to wrap metadata around a query (i.e. band the query with metadata). More information can be found in the Teradata online help.

Feature	New Syntax
Teradata query banding for session	`SET SQLSessionPrefix = 'SET QUERY_BAND = ' & Chr(39) & 'Who=' & OSuser() & ';' & Chr(39) & ' FOR SESSION;';`
Teradata query banding for query	`SET SQLQueryPrefix = 'SET QUERY_BAND = ' & Chr(39) & 'Who=' & OSuser() & ';' & Chr(39) & ' FOR TRANSACTION;';`

1.3 Additional features

- Section access on QlikView server is now supported
- Single Calculated dimensions are now supported
- Direct Discovery table rename now supported

2 What's New in Direct Discovery 11.2 SR5?

Direct discovery in SR5 has a number of changes and additions which have been categorized below:

2.1 Syntax

Old Syntax	New Syntax
`DIRECT SELECT`	`DIRECT QUERY`
`EXPLICIT`	`DIMENSION`
`IMPLICIT`	`MEASURE`
`SQL(``)`	`NATIVE(``)`
None available	`DETAIL`
None available	`DETACH`

2.2 New and changed Set Statements

Old Syntax	New Syntax
STALE after 15 seconds	`SET DirectCacheSeconds= 15;`
None available	`SET DirectStringQuoteChar='"';`
None available	`SET DirectIdentifierQuoteChar='[]';`
`SET LinkedConnectionMax= 4;`	`SET DirectConnectionMax= 4;`
None available	`SET DirectTableBoxListThreshold= 100000;`
None available	`SET DirectDistinctSupport=false;`
None available	`SET DirectIdentifierQuoteStyle='ANSI';`

2.3 Additional features

- It is now possible to add a WHERE clause to the script for reload
- Global search on dimension fields now supported
- Added feature for drill to details with table boxes and extra syntax keyword - Detail
- Cancel query enhancement

3 FAQ

3.1 What is QlikView Direct Discovery?

QlikView Direct Discovery capability combines the associative capabilities of the QlikView in memory dataset with a query model where the source data is not directly loaded into the QlikView data model, instead the aggregated query result is passed back to QlikView user interface. The direct discovery data set is still part of the associative experience where the user can navigate both on the in-memory data and the direct discovery data associatively.

- Can be used to build aggregated charts on **homogeneous** large data sets
- Can be used to look at detail records in a table box on large data sets
- Can reflect **updated** records without reloads (not new records)
- Can support more than one Direct Discovery table in certain scenarios

- Not as fast as in memory apps
- Will always be slower compared to SQL query run times on source due to associative model calculation times
- Not a solution for scalability/performance issues in the underlying source
- Not designed to convert all tables in apps into Direct Discovery mode
- Not a real time solution

3.2 How is Direct Discovery different from the traditional query or data visualization tools capabilities?

QlikView Direct Discovery feature is a hybrid capability where the in-memory and direct discovery data sets can be analyzed together, even on the same chart. The business users can associatively make selections on either of the data sets, and see what is associated and not associated with the same QlikView association colours; green, gray, and white. They can create charts that would help them

analyze data from both data sets together. This hybrid approach provides much greater power and flexibility than the data visualization tools or traditional query capabilities because with these tools the users can either create extracts to an in-memory engine or run queries on the database but cannot do both on the same application persistently. QlikView Direct Discovery enables users to perform business discovery and visual analysis against any amount of data, regardless of size. With the introduction of this unique hybrid approach, users can associate data stored within big data sources directly alongside additional data sources stored within the QlikView in memory model. QlikView can seamlessly connect to multiple data sources together within the same interface, e.g. Teradata, SAP, Google Big Query allowing the business user to associate data across the data silos.

3.3 How does QlikView manage the choice of going to the database vs. running in memory?

QlikView decides which data resides in-memory and which data is direct discovery data by using special script syntax. This allows certain data elements dictated by the script syntax not to be loaded into the QlikView data model during the script reload process, but still available for query purposes in QlikView objects in the user interface and to be combined for analysis with the QlikView in memory dataset. Once this structure is established, the direct discovery data can be joined with the in-memory data with the common field names. This allows the user associatively navigate both on the direct discovery and in memory data sets. QlikView's caching is also used to improve overall user experience. QlikView Server stores selection states of queries in memory. When the user makes the same set of selections, QlikView leverages the cache to improve the user experience. It is possible to set a time limit on caching. Please refer to question #3.14 for more information.

3.4 What are the benefits of using QlikView Direct Discovery feature?

With Direct Discovery, the business users can leverage the unique QlikView Business Discovery capabilities on larger data sets.

- **Drill down to details**: The business users can drill down to details from the aggregated in memory data set. They can leverage any data useful for analysis without scalability limitations of loading data in memory.
- **Associative data discovery on larger data sets**: Business users can analyze the larger data sets that they are not familiar with by making selections on the well-known data values that they use every day. QlikView will associatively show them the big data values for the selected in-memory data.
- **Extend QlikView's user experience beyond in-memory**: Self-service Big Data analysis by levering the unique QlikView experience with social, collaborative, and mobile capabilities. Business users can ask and answer questions on their own and in groups and teams to forge new paths to insight in the Data.

- **Easy to implement, rapid app development QlikView experience**: Business users can achieve Business Discovery on data sets that were previously used separately, or not used, because of their bulk and the development effort required. They can now analyze these data sets without waiting for the complicated and lengthy ETL developments.

3.5 How much does the QlikView Direct Discovery feature cost?

The QlikView Direct Discovery feature is offered free of charge with QlikView 11.2.

3.6 Are there any prerequisites to upgrade to QlikView 11.2?

No, there are not any prerequisites to upgrade to QlikView 11.2. However, it is advised to pursue the upgrade best practices.

3.7 Is this a developer enabled feature?

Yes. QlikView developer should setup the Direct Discovery table on the QlikView application load script to allow the business users to query the desired data source.

3.8 How does QlikView Direct Discovery work?

Within the script editor a new syntax is introduced to connect to data in direct discovery form. Traditionally the following syntax is required to load data from a database table:

```
ODBC CONNECT TO AdWorks;

LOAD CustomerID,
SalesPersonID,
SalesOrderID,
OrderDate,
month([OrderDate]) as OrderMonth,
year([OrderDate]) as OrderYear,
SubTotal,
TaxAmt,
TotalDue
DueDate,
ShipDate,
AccountNumber,
CreditCardApprovalCode,
rowguid,
ModifiedDate
SQL SELECT
CustomerID,
SalesPersonID,
SalesOrderID,
OrderDate,
```

```
SubTotal,
TaxAmt,
TotalDue
RevisionNumber,
DueDate,
ShipDate,
AccountNumber,
CreditCardApprovalCode,
rowguid,
ModifiedDate
FROM AdventureWorks.Sales.SalesOrderHeader;
```

To invoke the direct discovery method, the keywords “SQL SELECT” are replaced with “DIRECT QUERY” and to distinguish between data items to be loaded into memory and those which just have metadata loaded the DIMENSION/MEASURE/DETAIL keywords are used as shown below:

```
ODBC CONNECT TO AdWorks;

DIRECT QUERY
DIMENSION
CustomerID,
SalesPersonID,
SalesOrderID,
OrderDate,
NATIVE('month([OrderDate])') as OrderMonth,
NATIVE('Year([OrderDate])') as OrderYear
MEASURE
SubTotal,
TaxAmt,
TotalDue
DETAIL
DueDate,
ShipDate,
AccountNumber,
CreditCardApprovalCode,
rowguid,
ModifiedDate
FROM AdventureWorks.Sales.SalesOrderHeader;
```

In the example above, the source data table “SalesOrderHeader” has 16 fields, with the use of “DIRECT QUERY” and “DIMENSION” keywords, only columns CustomerID, SalesPersonID,SalesOrderID and OrderDate are loaded into memory as symbol tables. Other columns following the “MEASURE” and DETAIL keywords exist in the source data table within the database and they are not part of the in memory data model. Both measure and detail fields are fields that QlikView is aware of on a “meta level”. The actual data of measure/detail fields resides only in the database but the fields may be used in QlikView expressions. More information on the measure/detail fields are provided in the next section.

Please note that preceding load cannot be used with Direct Discovery as it only relates to the data that is loaded in memory.

3.9 What is a "DIMENSION" field?

A Dimension field can be thought of as a regular in memory field, all of the data and metadata is loaded into QlikView and the field is available to be used for associating the Direct table to other in memory tables. All dimension fields are defined after the DIMENSION keyword within the script and precedes the measures.

3.10 What is a "MEASURE" field?

A measure field is a field that QlikView is aware of on a "meta level". The actual data of a measure field resides only in the database during the reload process and is retrieved on an ad hoc basis driven by the QlikView front end chart expressions. The idea is that a measure field will be treated as any other field when the user works with expressions in QlikView. The reason to introduce the concept of measure fields is for QlikView to decide if an aggregation should be done on the database instead of being processed by QlikView itself. For example, if the expression Count(Address) is used in a QlikView chart and Address is a measure field QlikView will let the database do the count (obviously since QlikView doesn't have any data for the Address field). To put it in another way; measure fields will add support for aggregating in the database while using the same expression syntax QlikView already has. The result is that the usage of the direct discovery fields from the database will be transparent for the user.

3.11 How are the measure fields defined?

Using the keyword "MEASURE". Once this keyword is used, QlikView will only load the fields listed as MEASURE in addition to the fields listed after the DIMENSION keyword. This scenario can be used by the developer in the cases where there are fields on the database table that should be not used by the users on the QlikView application.

3.12 Is it possible to use source Database syntax in the QlikView script?

Yes. By using the new expression syntax, NATIVE(''), it is possible to run SQL syntax from a QlikView script however **NOT** through a QlikView chart. Please note that the SQL syntax should only be written against the source table defined as Direct Discovery table on the load script.

e.g. `NATIVE('cast(MktID as varchar(50))') as DetailID`

3.13 Where are the data aggregations done?

For the QlikView charts with Direct Discovery fields, aggregations are done on the database. If the chart also has in-memory dimensions, a second level aggregation is done on the chart level once the database level aggregations are done.

3.14 What is the expected performance of the QlikView application that uses Direct Discovery?

- The performance will not be equal to that of an application with all in memory data
- Time is also taken to render the data into the associative model, comparison of SQL execution times between chart rendering in QlikView and executing the same SQL on source will differ

The performance of the Direct Discovery feature will also reflect the performance of the underlying source. It is possible to use standard database and query tuning best practices for this feature. Direct Discovery feature does not provide any support for query performance tuning from the QlikView application. To improve the overall user experience, QlikView's caching is used. QlikView will store selection states of queries in memory as the same types of selections are made. QlikView will leverage the query from the cache. These cached result sets are shared across users. Please note that, when the same types of selections are done, QlikView will not query the source data and will leverage the cached result set. A set statement can be used on the load script to set a caching limit on the Direct Discovery query results (in seconds). Once this time limit hits, QlikView Server will clear the cache for the Direct Discovery query results that were generated for the previous selections.

```
SET DirectCacheSeconds= 15;
```

When this happens QlikView will query the source data for the selections and will create the cache again for the designated time limit.

3.15 What is Drill to Details?

Drill to details is a new capability introduced in 11.2 SR5 and allows "detail" records to be displayed in a QlikView table box only. A detail record is one to which no aggregation is applied and effectively just generates a select statement from QlikView and renders the appropriate values. To cater for the potential large volume of data which could be rendered an additional Set variable is introduced:

```
SET DirectTableBoxListThreshold = 100000;
```

The default value is 1000.

3.16 What are DETAIL fields?

As part of the drill to details functionality additional syntax has been introduced (this follows the DIRECT QUERY statement):
DETAIL - These are the fields (like a "Comment" field) that the user may want to display in a drill-to-details table box but that should not be involved in any chart expressions

3.17 What type of database connection is required to use QlikView Direct Discovery?

QlikView Direct Discovery can only be used against SQL compliant data sources. The following data sources are supported:

- ODBC/OLEDB data sources - ODBC/OLEDB sources are supported, including SQL Server, Teradata and Oracle.
- Custom connectors which support SQL – SAP SQL Connector, Custom QVX connectors for SQL compliant data stores.
- SAP HANA,Cloudera/MapR/Horton Works Hadoop, HP Vertica and Parstream via ODBC.

3.18 Are both the 32-bit and 64-bit connections supported with this feature?

Yes.

3.19 Which QlikView aggregation functions can be used with this new feature?

As QlikView Direct Discovery generates SQL as its base code not all QlikView functionality will be compatible with the feature. The functions that are supported with this initial release are; Sum, Avg, Count, Min, Max.

3.20 How is the database connection security handled?

All of the users using the QlikView application with the Direct Discovery capability will be using the same connection. With this release, authentication pass through or credentials per user is not supported.

3.21 Is it possible to do asynchronous, parallel calls to the database?

Yes. By using the connection pooling capability, it is possible to execute multiple calls to the database. The load script syntax to setup the pooling capability is as follows:

```
SET DirectConnectionMax=10;
```

The default setting is 1.

3.22 Is it possible to optimize the Direct Discovery queries from the QlikView application?

All of the performance tuning should be done on the source database. QlikView Direct Discovery feature does not provide any support for query performance tuning from the QlikView application.

3.23 Are all of the QlikView features compatible with this feature?

Due to the interactive and SQL syntax specific nature of the Direct Discovery approaches, the following QlikView features are not supported;

- Set Analysis
- Direct Discovery Measure and Detail fields are not supported on Global Search
- Desktop Section access and data reduction
- Loop and Reduce
- Synthetic keys on the Direct Discovery table

3.24 Which data types are supported?

All data types are supported, however there maybe cases where specific data formats need to be defined to QlikView in order to send the correct literal to the source database. This can be done on the load script by using a series of SET statements examples are shown below:

Set Statement	Description
DirectDateFormat	Controls the format of the date literals sent to the database
DirectTimestampFormat	Controls the format of the datetime literals sent to the database
DirectTimeFormat	Controls the format of the time literals sent to the

	database
DirectUnicodeStrings	Controls whether string literals are prepended with the ANSI standard wide character marker N. Not all databases support this. Set to **TRUE** to enable; any other setting disables.
DirectMoneyFormat	Controls the format of the money literals sent to the database, This is not a display format, so it should not include currency symbols or thousands separators, default '#.0000'
DirectMoneyDecimalSep	Controls the decimal separator used in money literals sent to the database. Note that this character must match the character used in DirectMoneyFormat. Default is '.'
DirectStringQuoteChar	Controls the character to be used to quote the strings in a generated query. The default is a single quote.
DirectIdentifierQuoteChar	Controls the quoting of identifiers in a generated query. This can be set to either one character (such as a double quote) or two (such as the pair "[]"). The default is a double quote.
DirectCacheSeconds	Controls the amount of time a query result set from an application is stored in memory cache, default setting is 3600 seconds.
DirectTableBoxListThreshold	Controls the amount of drill to detail data available to be shown in a table box, the default is 1000 rows.
DirectDistinctSupport	Changes the default syntax of SELECT DISTINCT to use GROUP BY for sources which do not support DISTINCT
DirectIdentifierQuoteStyle	For non-standard ANSI quoting anything other than 'ANSI' in the statement will change the syntax

3.25 What are the logging capabilities with the Direct Discovery feature?

Behind the scenes a direct SQL statement is passed to the underlying data source as such the statement passed can be viewed through the trace files of the underlying connection. An example of this can be found in the ODBC Data Source Administrator in control panel "Start Tracing Now" button. The subsequent trace file details SQL statements generated through the user selections and interactions.

3.26 How do I join tables with Direct Discovery?

Direct Discovery tables can be linked via a where clause an example script which joins the Product/Product Sub Category table via ProductSubCategoryID is shown below:

```
Product_Join:
DIRECT QUERY
DIMENSION
      [ProductID],
      [AdventureWorks2012].[Production].[Product].[Name] as [Product Name],
      [AdventureWorks2012].[Production].[ProductSubCategory].[Name] as [Sub
Category Name],
      Color,
      [AdventureWorks2012].[Production].[Product].ProductSubcategoryID as
[SubcategoryID]
      MEASURE
      [ListPrice]
FROM
[AdventureWorks2012].[Production].[Product],[AdventureWorks2012].[Production].[Pr
oductSubcategory]
WHERE [AdventureWorks2012].[Production].[Product].ProductSubcategoryID =
[AdventureWorks2012].[Production].[ProductSubcategory].ProductSubcategoryID   ;
```

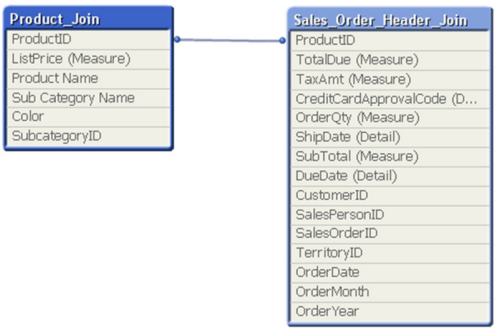
Multiple tables can be added into the Direct Discovery load statement using standard ANSI SQL, an example with a FROM/JOIN/ON statement is shown in Figure 13. This statement joins the SalesOrderHeader to the SalesOrderDetail table via SalesOrderID and also joins the Customer table to the SalesOrderHeader table via the CustomerID.

```
Sales_Order_Header_Join:
DIRECT QUERY
            DIMENSION
                  AdventureWorks2012.Sales.Customer.CustomerID as CustomerID,
                  AdventureWorks2012.Sales.SalesOrderHeader.SalesPersonID as
            SalesPersonID,
                  AdventureWorks2012.Sales.SalesOrderHeader.SalesOrderID as
            SalesOrderID,
                  ProductID,
                  AdventureWorks2012.Sales.Customer.TerritoryID as TerritoryID,
                  OrderDate,
                  NATIVE('month([OrderDate])') as OrderMonth,
                  NATIVE('year([OrderDate])') as OrderYear
```

```
MEASURE
     SubTotal,
     TaxAmt,
     TotalDue,
     OrderQty
DETAIL
    DueDate,
    ShipDate,
    CreditCardApprovalCode,
    PersonID,
    StoreID,
    AccountNumber,
    rowguid,
    ModifiedDate
FROM AdventureWorks2012.Sales.SalesOrderDetail
          JOIN  AdventureWorks2012.Sales.SalesOrderHeader
          ON (AdventureWorks2012.Sales.SalesOrderDetail.SalesOrderID =
          AdventureWorks2012.Sales.SalesOrderHeader.SalesOrderID)
          JOIN AdventureWorks2012.Sales.Customer
          ON(AdventureWorks2012.Sales.Customer.CustomerID =
          AdventureWorks2012.Sales.SalesOrderHeader.CustomerID);
```

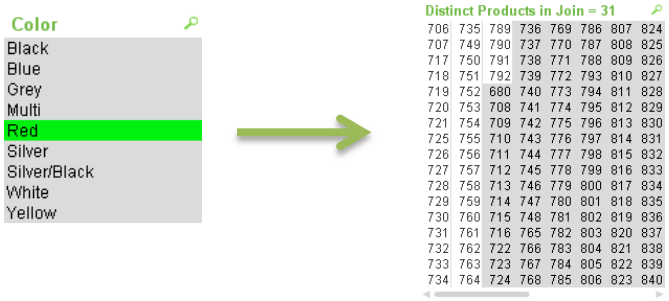
3.27 What are the supported multi-table scenario's?

In the following model the Product and Sales Order tables are joined via ProductID



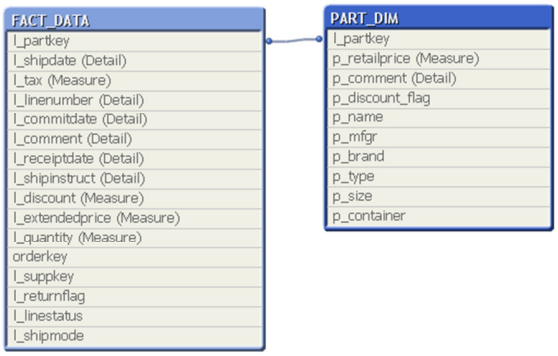
As the key field has a low distinct value count in this case 266 small SQL statements would be passed to the underlying source.

Building a chart of OrderMonth vs sum(Subtotal) and selecting a Color from the Product table will traverse the table and produce SQL as follows:



SELECT ProductID, month([OrderDate]), SUM(OrderQty), SUM(SubTotal)
FROM SalesTable
WHERE ProductID IN (739, 741, 742, 748, 771, 772, 773, 774, 779, 780, 781, 880, 894, 904, 905, 906, 907, 917, 918, 919, 920, 942, 944, 945, 948, 952, 980, 981, 982, 983, 984, 985, 986, 987, 988, 740)
GROUP BY ProductID, month([OrderDate])

In the following model the Fact and Part tables are joined via l_partkey, the join field has high cardinality with over 20,000 distinct rows. Selecting any dimension filter from the Part table could produce large SQL statements which may not be supported on the underlying source



For example the p_discount_flag has a distribution of 19,996 l_part_keys with a flag of “Y”. Building a chart of l_part_key vs sum(Quantity) and selecting Y in this dimension field from the Product table will traverse the table to find the corresponding l_part_keys and produce SQL as follows:

SELECT "l_partkey", SUM("l_quantity") FROM "lineitem"
WHERE "l_partkey" IN (1,2,3,4,.....+ another 19,996 partkeys)
GROUP BY "l_partkey"

This would result in slow running charts and may invoke an error from the underlying source on the length of the generated string containing the SQL statements (SQL Server limits http://msdn.microsoft.com/en-us/library/ms143432.aspx).

3.28 Is Direct Discovery going to be in Qlik Sense

Yes