


QuickBooks - Data Extraction Queries

QuickBooks Queries

 Knowledge of QuickBooks and QuickBooksXML is required for mapping and creating a process for integrating with QuickBooks.

QuickBooks Desktop" adapter/connector provides interfaces for:

- Reader: reading from QuickBooks with an SQL like query.
- Writer: writing into QuickBooks.
- Metadata query: navigating through QuickBooks objects.

You can build the queries of QuickBooks or QuickBooks Desktop Adapter from the Reader Section of the Process Definition. You can build queries similar to SQL, which includes the straightforward query.

Reader Query Examples:

Customer Query

The following is a query to extract all Customers in batches of 100 that have been changed since the last successful run of the process.

```
Select * from Customer where Customer.MaxReturned=100 and  
Customer.FromModifiedDate=$last_success_run
```


- MaxReturned suggests how many records to be retrieved from QuickBooks.
- FromModifiedDate suggests the filter for selection Customers that are edited or created after the previous sync run.
- \$last_success_run is a process specific variable that gets setup whenever the ETL process is successfully completed.

Note: Refer the filters from the QuickBooksXML. This is available in DBSync On-Premise version under this DBSync2\src\www\WEB-INF\conf folder. The XML Structures of all QuickBooks Objects are specified in this file.

ItemService Query

The following is a query to extract all ItemService in batches of 100 that have been changed since the last successful run of the process.

```
Select * from ItemService where ItemService.MaxReturned=100 and  
ItemService.FromModifiedDate=$last_success_run
```

 Other items - such as ItemNonInventory, ItemInventory, ItemDiscount - are not supported in QuickBooks Online Edition.

Complex query with a join

```
select * from Invoice, Customer where Invoice.MaxReturned=100 and  
Invoice.ModifiedDateRangeFilter.FromModifiedDate=$last_success_run  
and Invoice.IncludeLineItems=true and Invoice.OwnerID=0 join on  
Invoice.CustomerRef.FullName=Customer.FullName
```

The above query joins Invoice and Customer QuickBooksXML responses into one XML. As a result, the transformation engine can avoid multiple processes. This is done to optimize data flow to and from QuickBooks.