Final Project – Software Design and Engineering

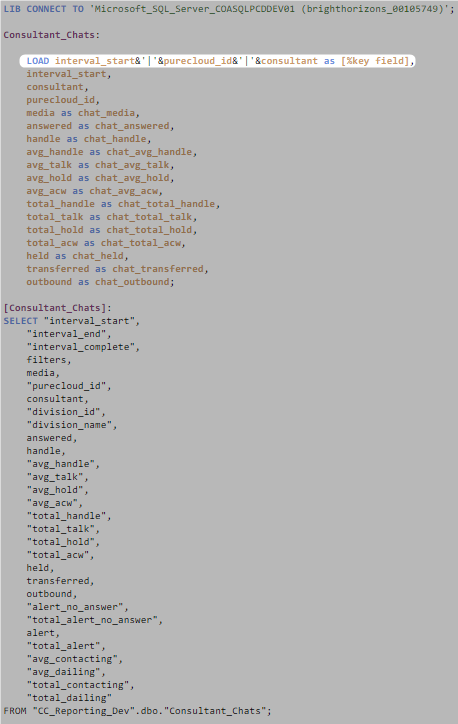
Nathaniel Carlson

CS-499

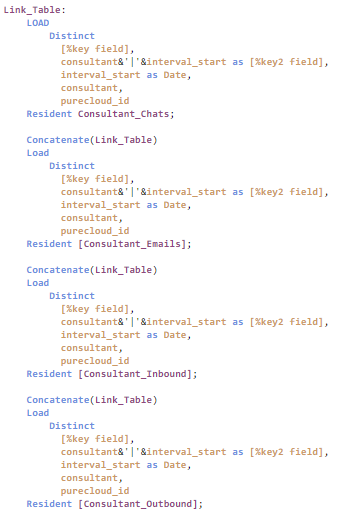
For this project, I have chosen a dashboard created in Microsoft Excel to modernize and implement in a Qlik dashboard. The dashboard was created for the contact center I work in within the last year and is used to report on consultant metrics for performance evaluations, bonus payouts, and consultant coaching and corrective action. These metrics are KPIs that come from source data in several different systems that are meant to give a good measure of a consultant’s performance. This update will provide additional functionality, ease of use, and flexibility due to the power and increased flexibility of Qlik Sense. Additionally, housing the data for the dashboard in a SQL database will make updates easier and the results more accurate by avoiding data transcription errors and other possible issues of human error.

The biggest design problem to overcome in my project is how to tie together the data needed for the data points and analysis I need to present. Typically when developing a schema for a data analysis application, a star schema is preferred. This work by using a central fact table that contains a list of all of the transactions being reported on or analyzed, and dimension tables splitting off from this central fact table that give additional details on different elements of each transaction. The problem with this layout with my implementation is that I have multiple fact tables. For each consultant, I need to report on inbound calls, outbound calls, case work, schedule adherence and more. These all essentially represent different fact tables, and within each I need both the consultant that the data represents, and the date the data represents. This becomes especially problematic because both of these values, and in some cases, a consultant ID value as well all appear on multiple tables. This causes problems when trying to link tables as Qlik is not able to correctly a primary key, and having multiple keys causes circular references and synthetic keys, which interfere with proper filtering in the final dashaboard.

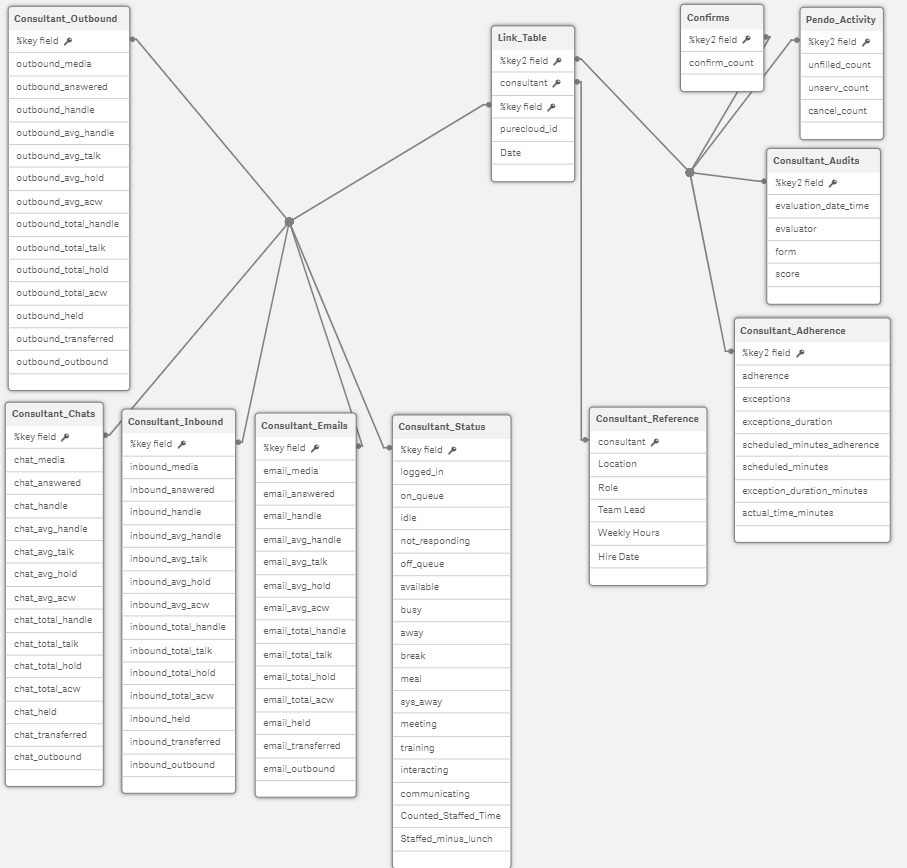
To address this, I had to combine the consultant name, date, and where applicable, ID number into a single key field in each table, and then iterate through these tables to create a link table to establish distinct combinations of name, date and identifier within each table, and separate out the correct consultant name and date to associate with them. This was all done within the data load editor in Qlik Sense. First, each table is loaded into Qlik Sense from the SQL database, with the consultants name, date and purecloud ID being concatenated into a single table value identified as the key field.



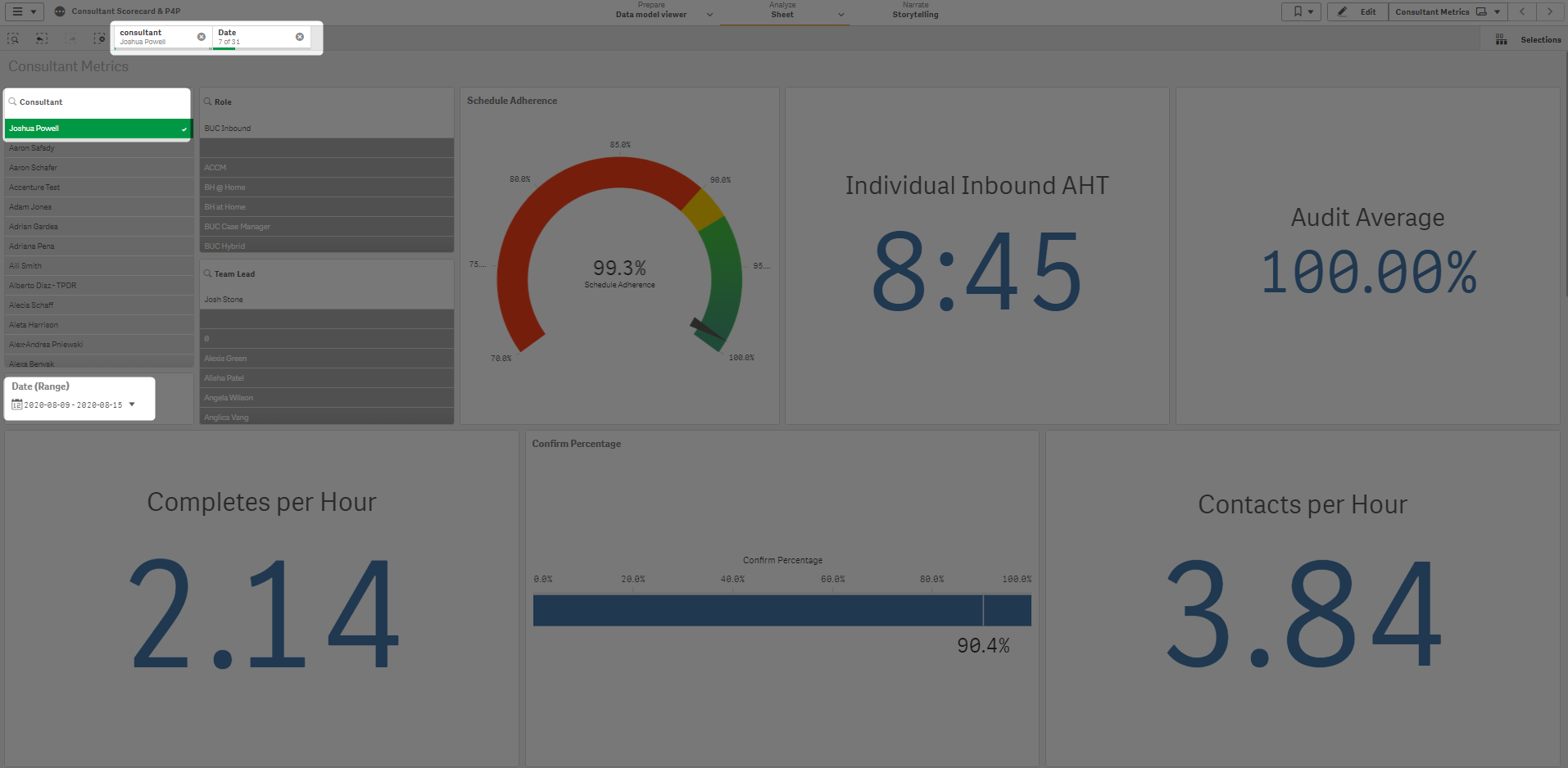
Once these tables are loaded, a single table needs to be created as a link between the four consultant interaction tables. To do this, a link table is created using the distinct key field values, as well as the separated corresponding name, date and ID from one of the initial tables by using a Resident Load. This is repeated with a concatenation function for the remaining tables to append additional distinct values to the existing table in order to capture every unique combination of name, date and ID in all four tables in the link table.



Once this is done, the consultant name, date and PureCloud ID are dropped from the original tables. The result is that each of the original tables has a distinct primary key, that can be linked back to the link table which in turn houses the consultant name and date for the corresponding key combination. By building the tables within Qlik Sense this way, the Link Table serves as the primary fact table to reference, and the star schema is preserved despite having multiple fact tables.



By designing the schema this way, filters for consultant, date and ID can be preserved and function correctly across the final dashboard to give the intended behavior.



This design solution perfectly addressed the needs of this particular application. Initially in trying to develop the dashboard I was running into problems getting date filters to apply correctly across all tables. Using the date field from one table would not update metrics being pulled from a different table, and trying to link all the tables to a master calendar resulted in circular references and synthetic keys. This solution ties all of the tables together correctly, and filters apply across all of them, giving correct and accurate filtered results with only a single filter needed for all tables.

Working out how to apply the linked table solution was a bit frustrating for me, as many of the available resources for Qlik don’t mention this type of application. In most applications, a single fact table such as a list of sales transactions or a budget statement make up all of the transactions to be analyzed, and additional dimensions take the shape of products sold, salesperson, client and so forth. Using multiple fact tables seems to be a format fairly specific to contact centers and consultant specific reporting. Learning how to do this however was a great experience for me, both in the context of this particular application, and more generally as a method of managing tables in a SQL database. The technique of combining multiple fields into a single unique key field can be used anywhere that there are tables that must have multiple matching values in them, and I’m sure as I continue to work with data professionally, I’ll find additional applications where this approach makes sense as a functional and elegant solution.