Assignment -3

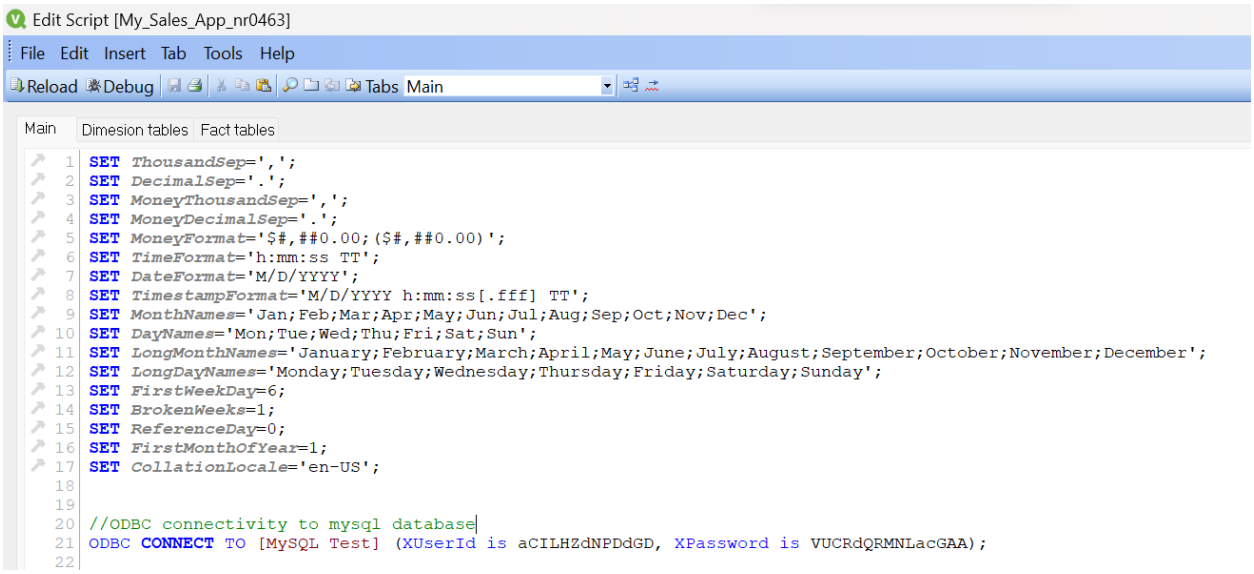
QlikView Application on top of my Dimensional model

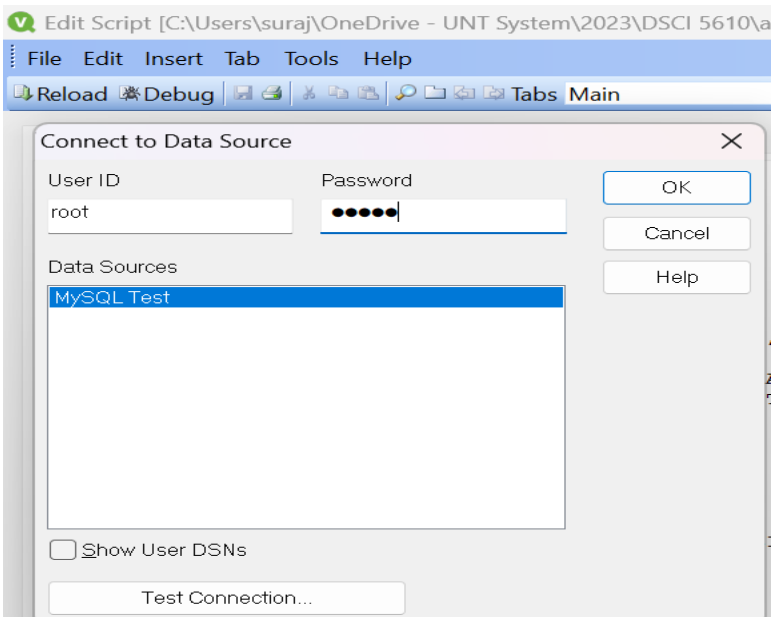
Created by-

Narsapure Suraj Reddy

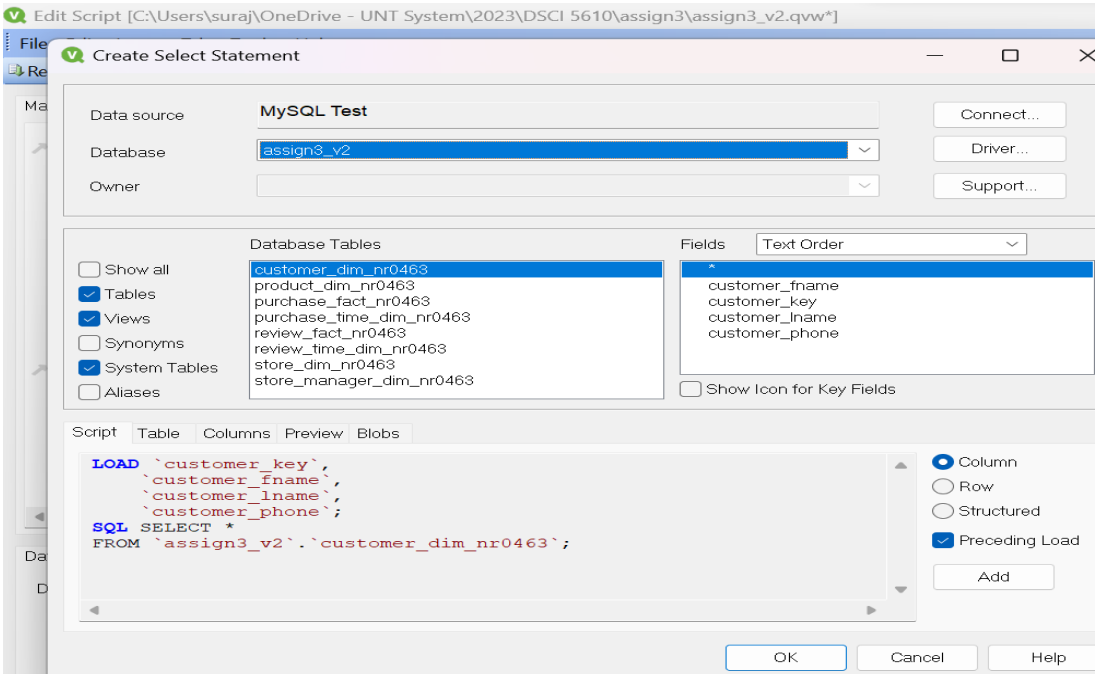
ID- 11612343

1. We started by establishing connection to my MySQL database using ODBC with the user credentials we use to login to the database.

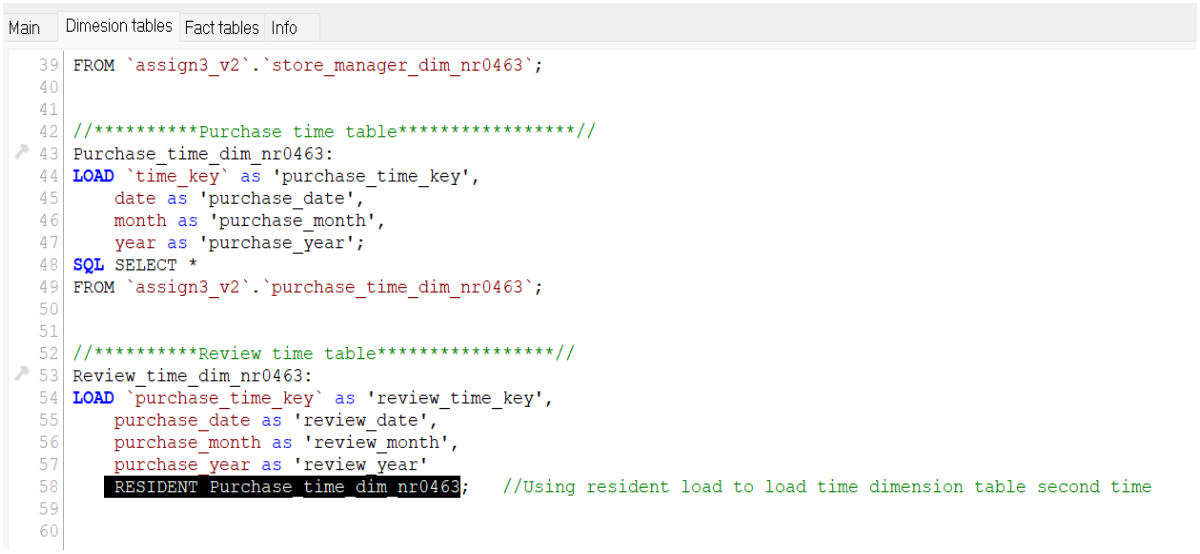


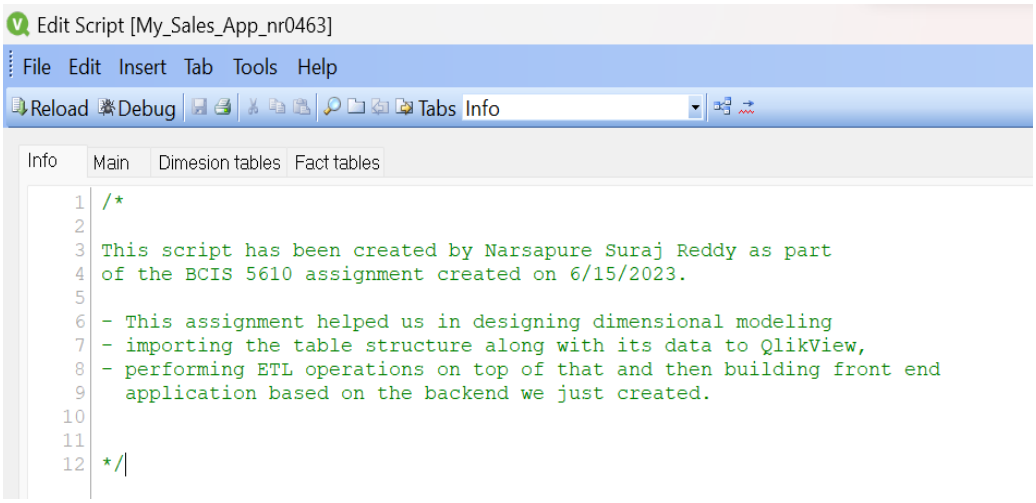


2. Once connected, choose the appropriate schema to list all the tables and select the preceding load option. In the script editor include the table name above each table along with a comment. Create separate tabs for dimension and fact table scripts. One info tab is included to give description about the script.

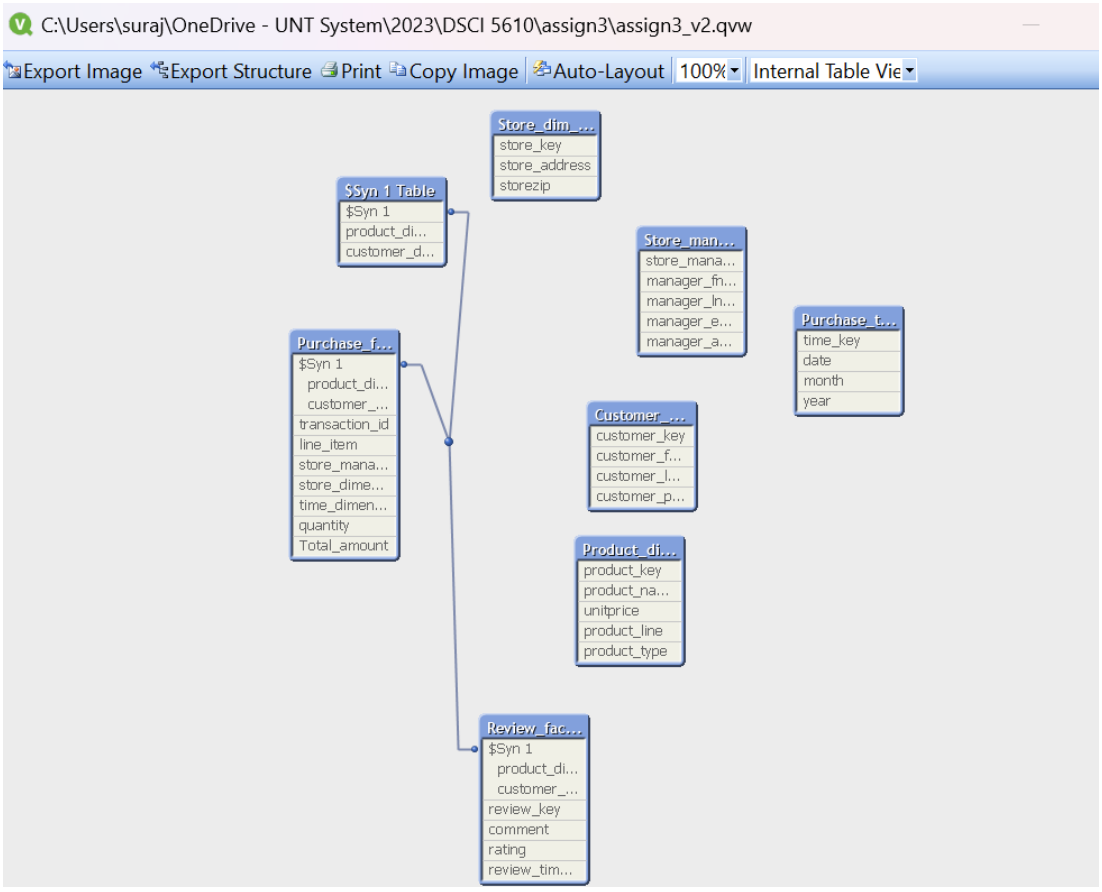


Following the best practices, I tried to include resident load to load similar data, used indentation, comments wherever required and Info tab to describe the purpose of the script.

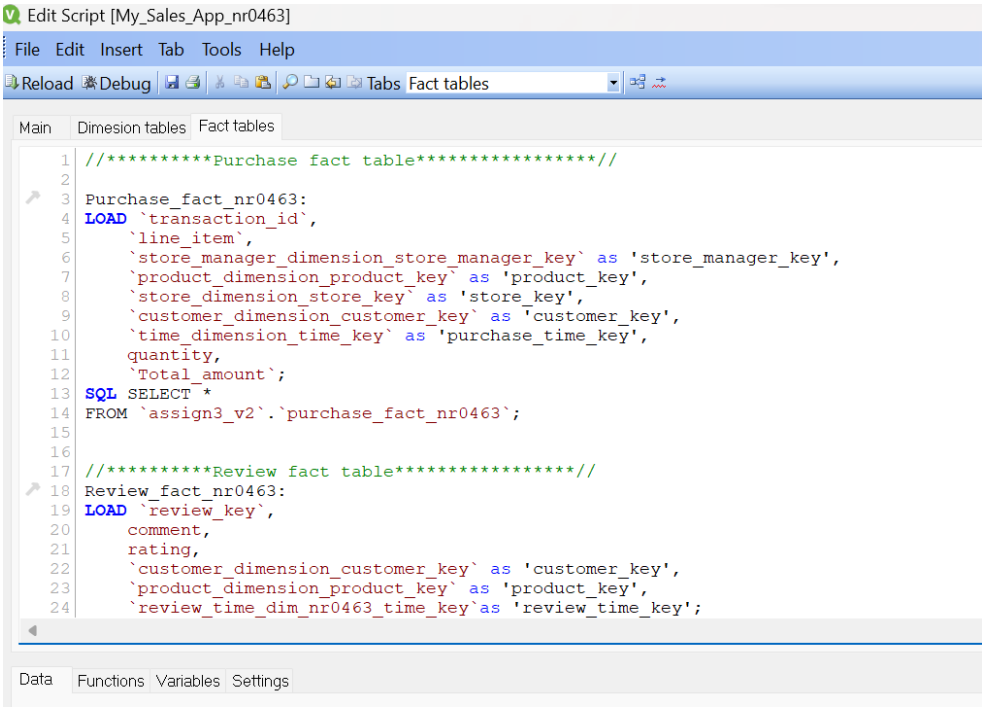




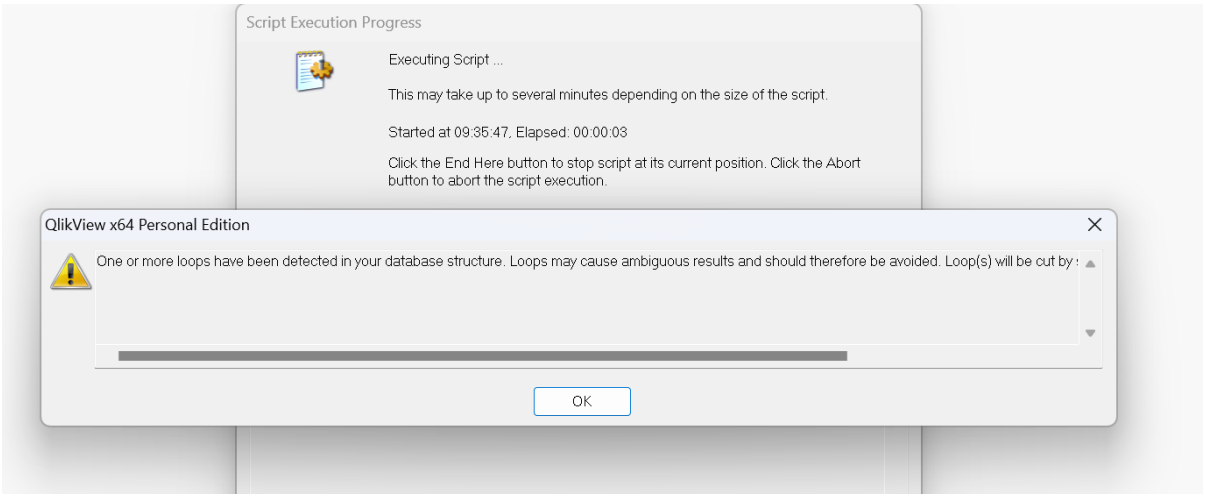
3. Once loading all the tables when we build the script we get the following table view. This is because the foreign keys in fact table are attributed along with their dimension table names.



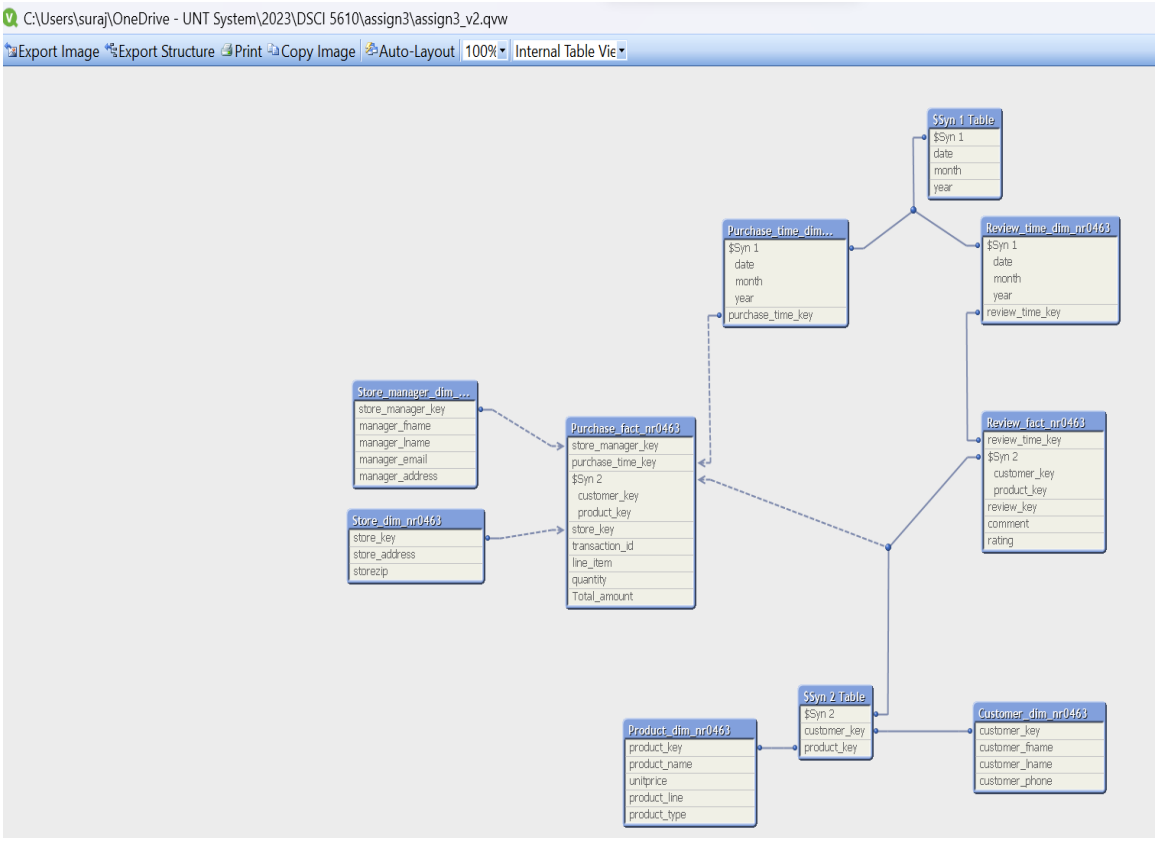
4. So, we need to alias them so that these foreign keys in fact tables have same names as they are in dimension tables, so that they can link with each other.



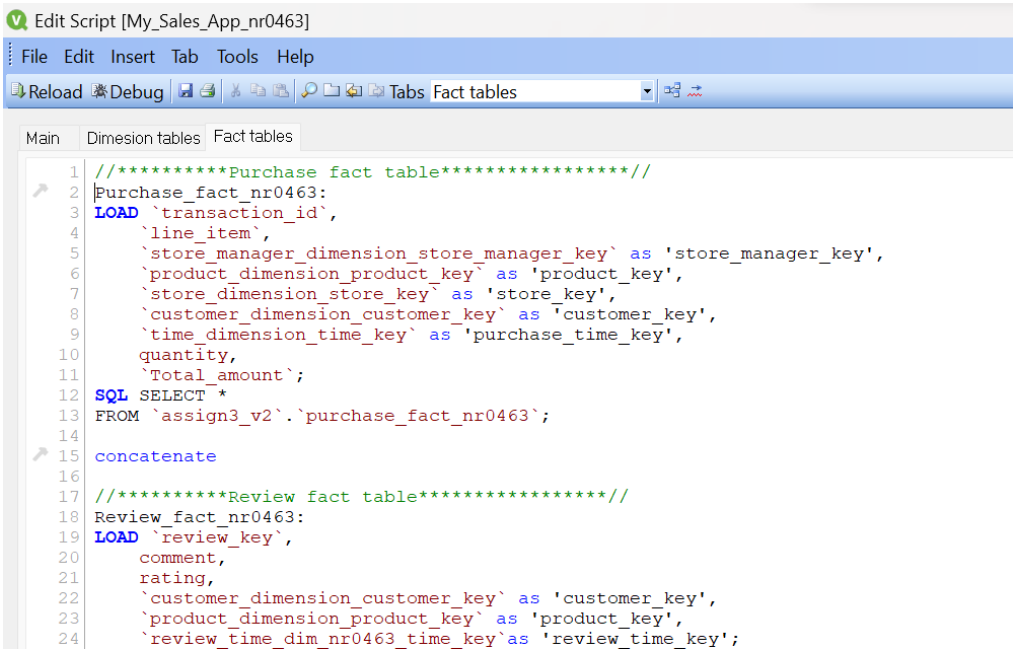
5. After these changes when we build we are prompted with errors saying there are loops which got created.



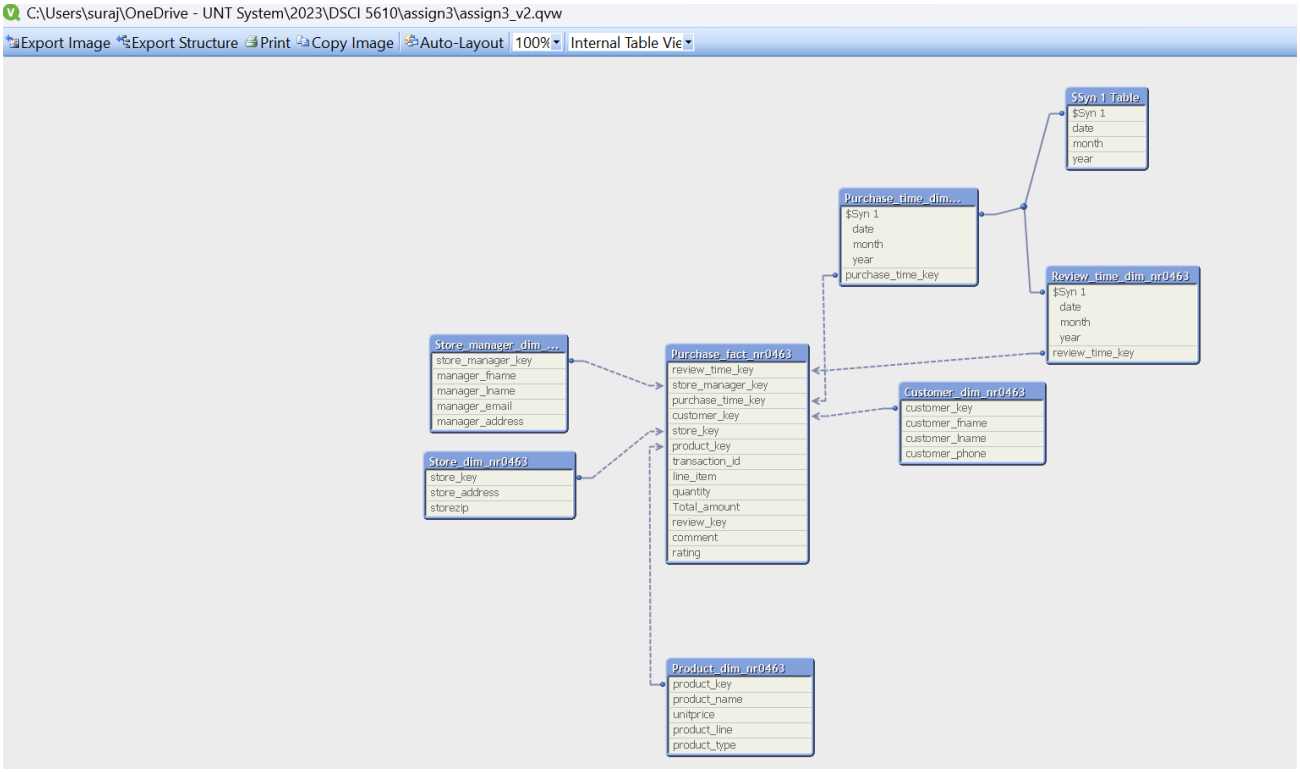
6. In the table viewer we see the below table structure.



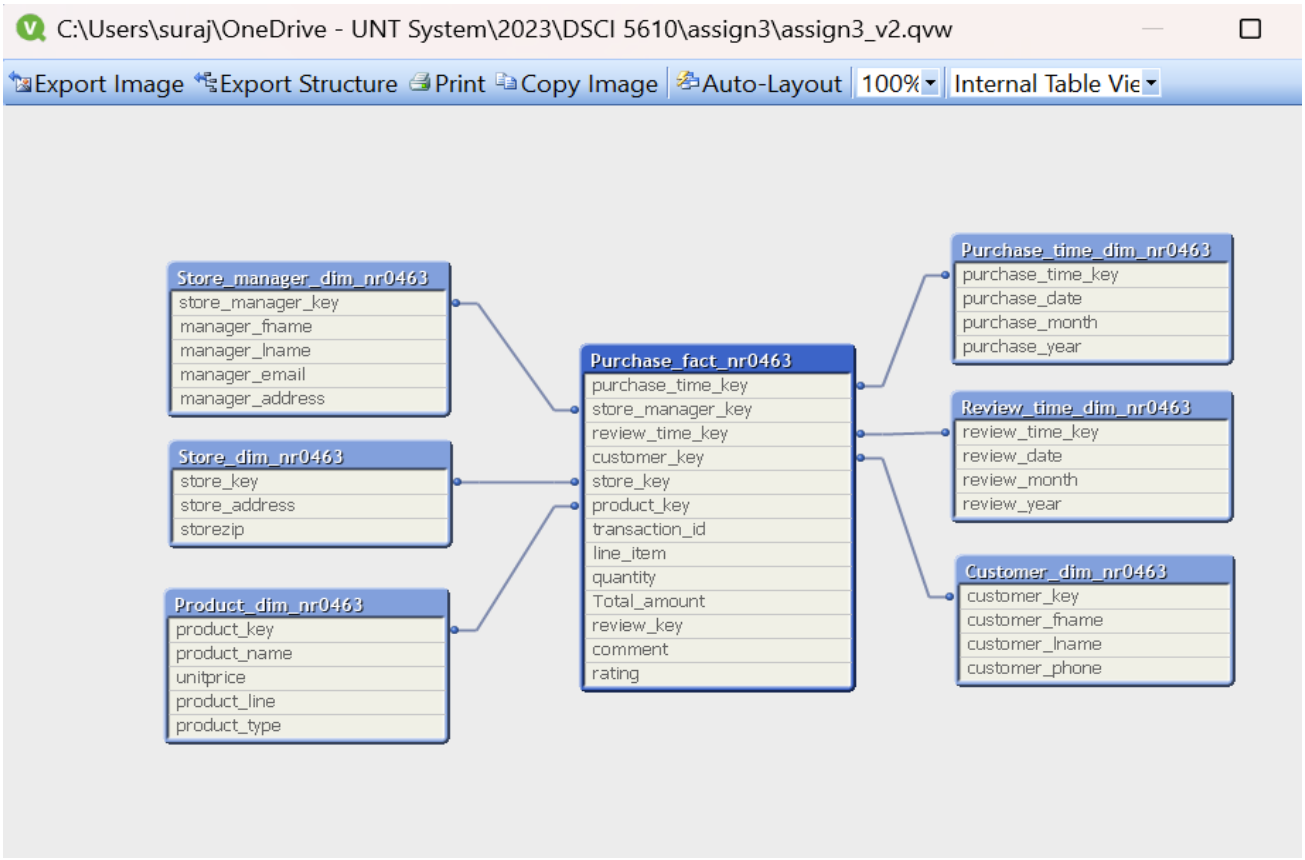
7. There is a loop created between two fact tables and product and customer dimension tables. To resolve this, we will use the' concatenate' command to concatenate both the fact tables. Since Purchase fact table was first table in concatenate command the final table will have the purchase fact table name.



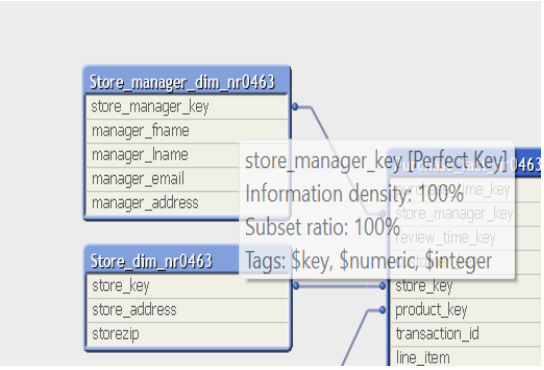

```
//**********Purchase fact table*****************//
Purchase_fact_nr0463:
LOAD `transaction_id`,
    `line_item`,
    `store_manager_dimension_store_manager_key` as 'store_manager_key',
    `product_dimension_product_key` as 'product_key',
    `store_dimension_store_key` as 'store_key',
    `customer_dimension_customer_key` as 'customer_key',
    `time_dimension_time_key` as 'purchase_time_key',
    quantity,
    `Total_amount`;
SQL SELECT *
FROM `assign3_v2`.`purchase_fact_nr0463`;

concatenate

//**********Review fact table*****************//
Review_fact_nr0463:
LOAD `review_key`,
    comment,
    rating,
    `customer_dimension_customer_key` as 'customer_key',
    `product_dimension_product_key` as 'product_key',
    `review_time_dim_nr0463_time_key`as 'review_time_key';
```

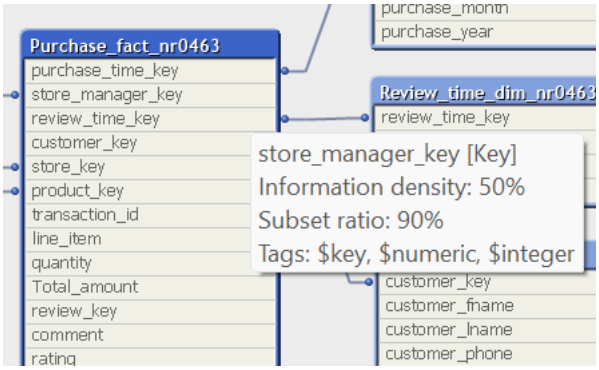


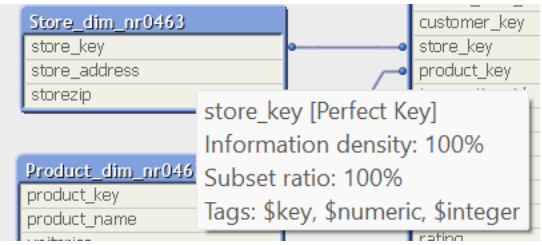
8. We will alias the attribute name of both the time dimension tables, so that they both don't form the synthetic key and tables. Final table view looks like below:

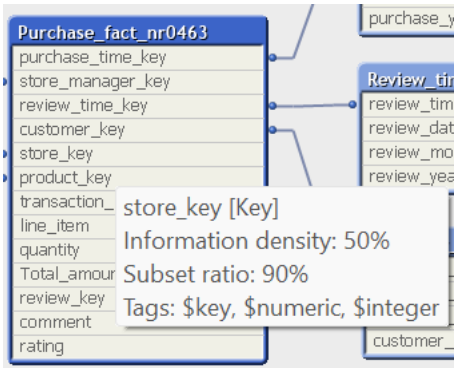


9. By hovering on the attributes of the dimension table and fact tables we can see the information density and subset ratio to understand how many values are present in those columns.

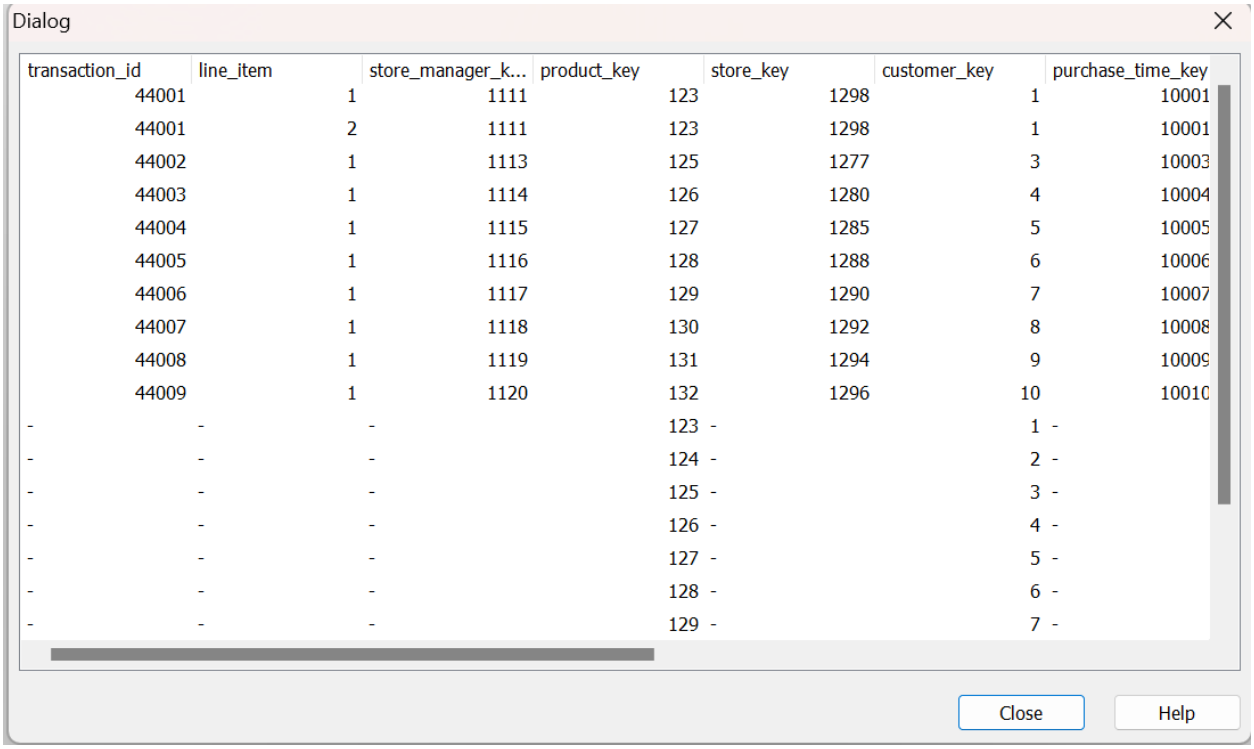




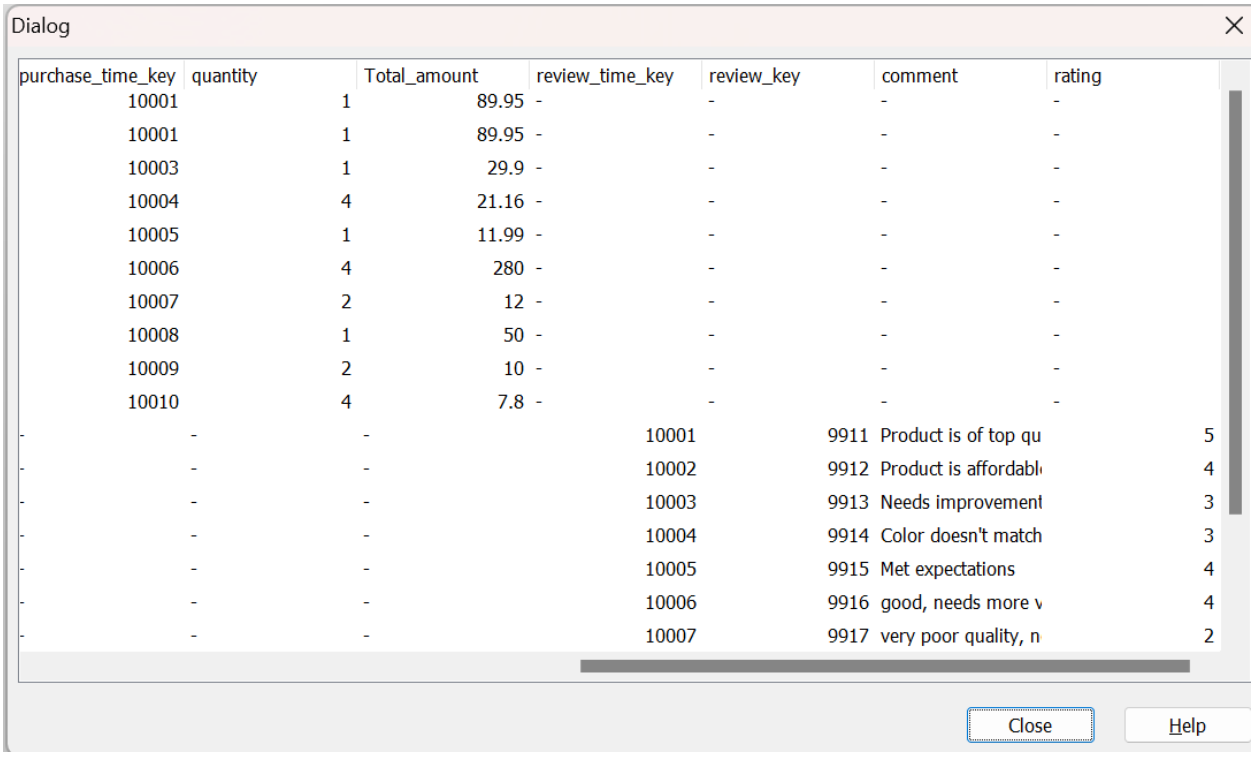




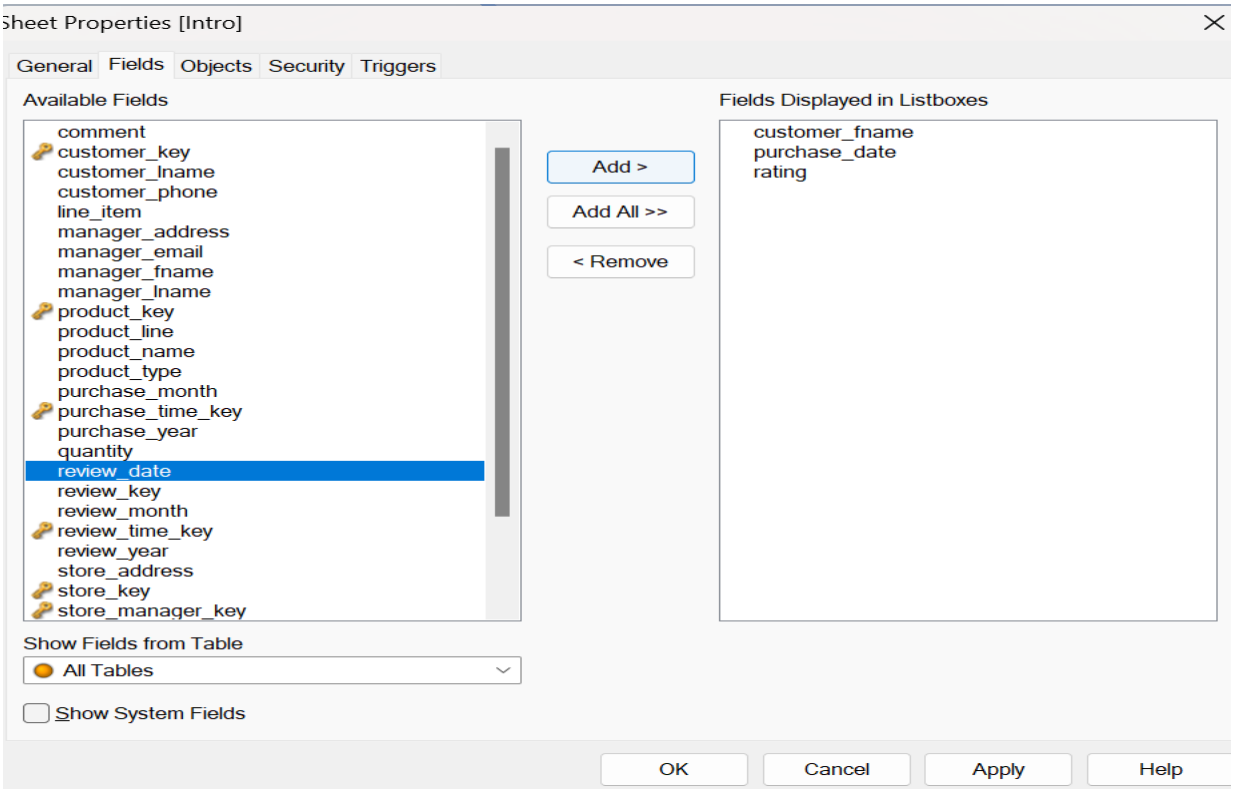
10. Below dialog boxes are result when we previewed the data in the concatenated fact table.



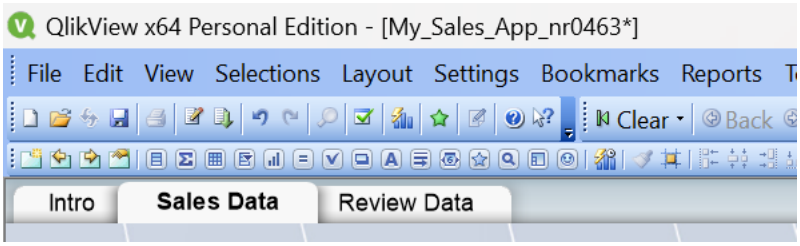
transaction_id	line_item	store_manager_k...	product_key	store_key	customer_key	purchase_time_key
44001	1	1111	123	1298	1	10001
44001	2	1111	123	1298	1	10001
44002	1	1113	125	1277	3	10003
44003	1	1114	126	1280	4	10004
44004	1	1115	127	1285	5	10005
44005	1	1116	128	1288	6	10006
44006	1	1117	129	1290	7	10007
44007	1	1118	130	1292	8	10008
44008	1	1119	131	1294	9	10009
44009	1	1120	132	1296	10	10010
-	-	-	123	-	1	-
-	-	-	124	-	2	-
-	-	-	125	-	3	-
-	-	-	126	-	4	-
-	-	-	127	-	5	-
-	-	-	128	-	6	-
-	-	-	129	-	7	-



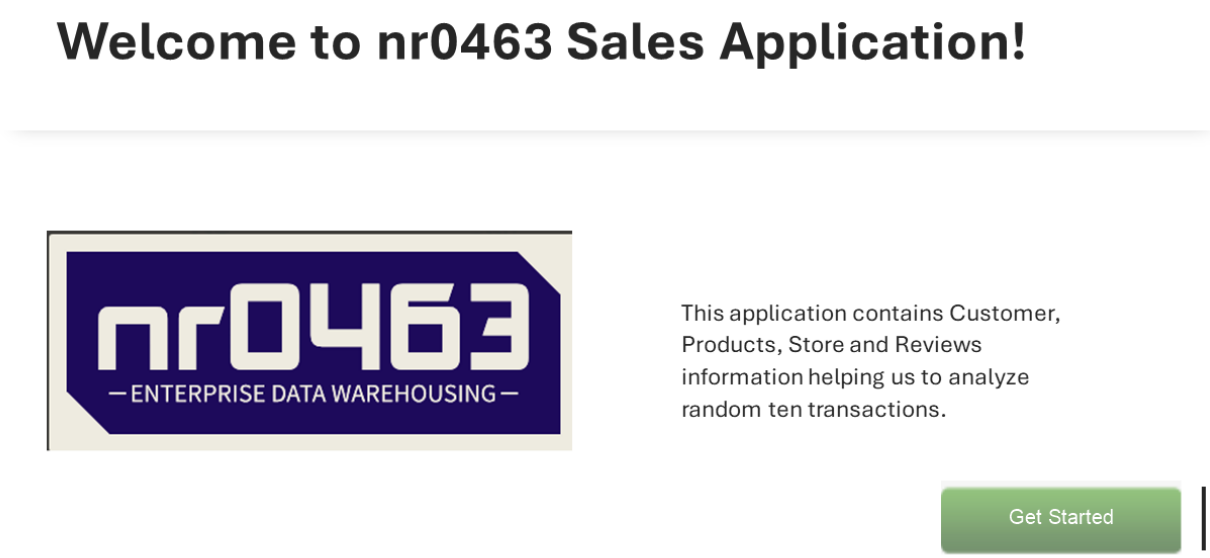
11. This completes the back-end part of our QlikView application, now we will work on the front-end part. Once we reload the script, we can choose the fields we want to insert in the sheet.



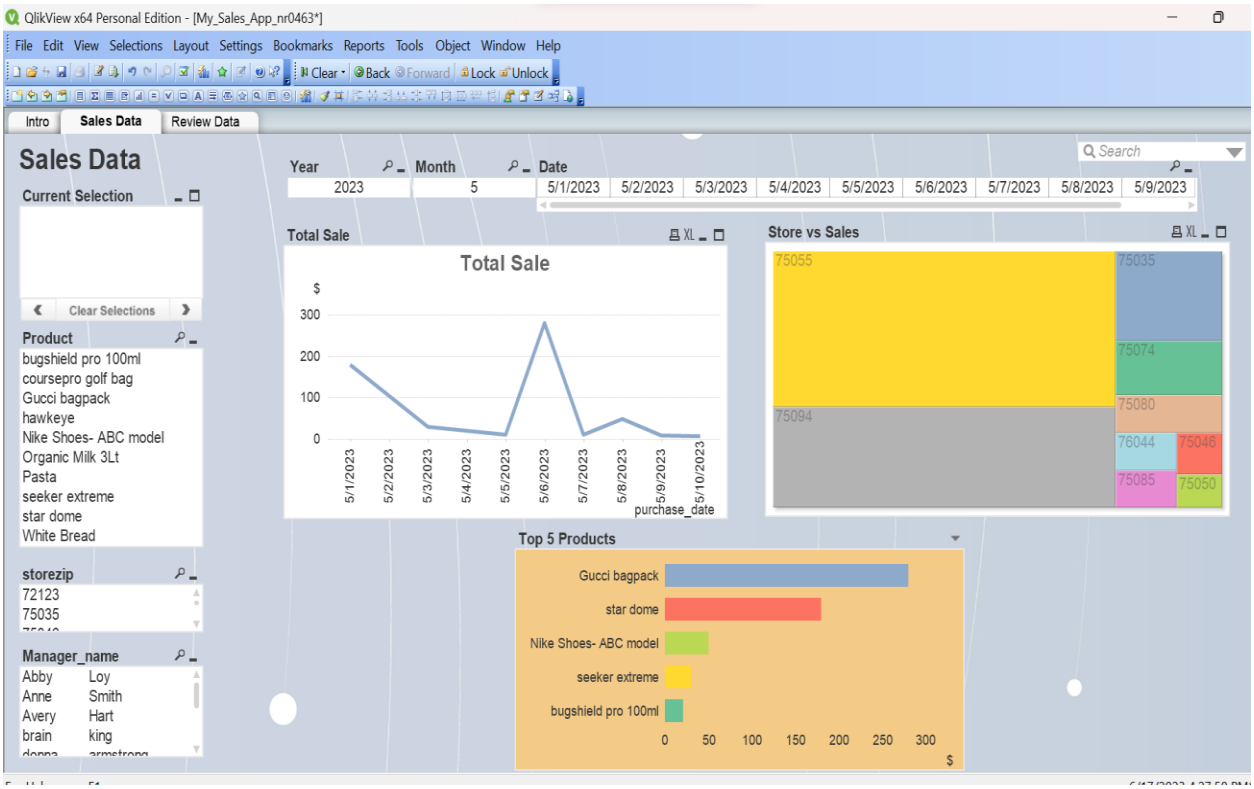
12. All the fields will appear as list boxes in the sheet. We can add more sheets and rename them as per our requirement.



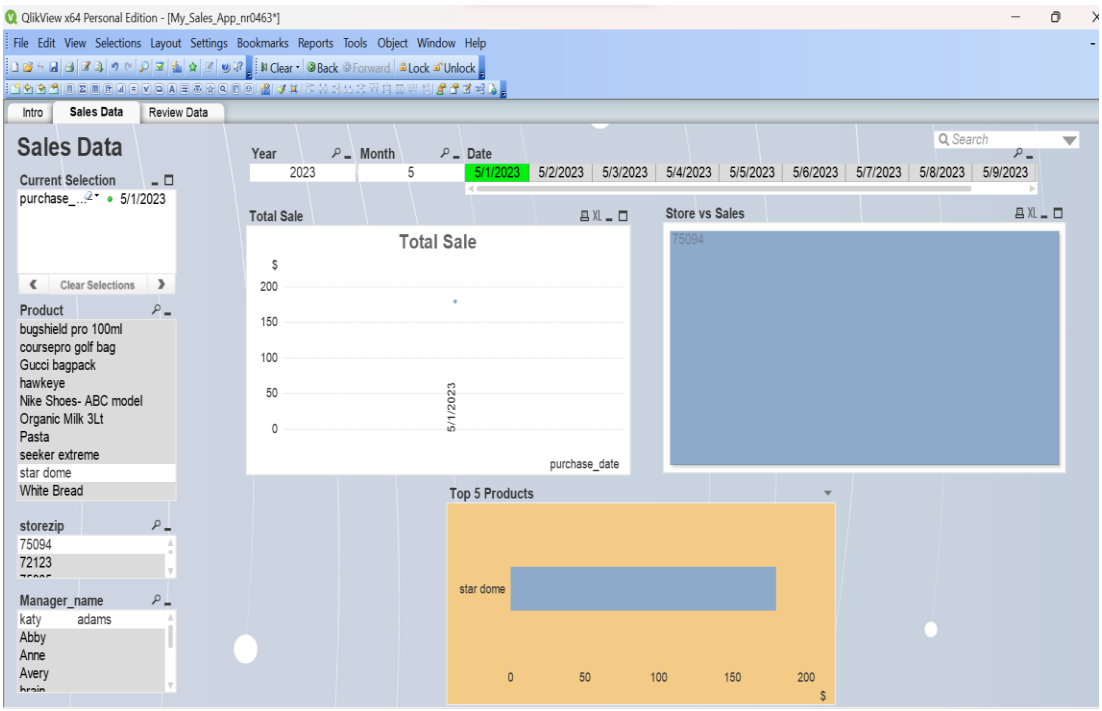
13. I created 3 tabs. The intro tab contains information about the application. Sales data contains information about the product sales along with store information whereas review data contains information about the product review ratings along with customer information.



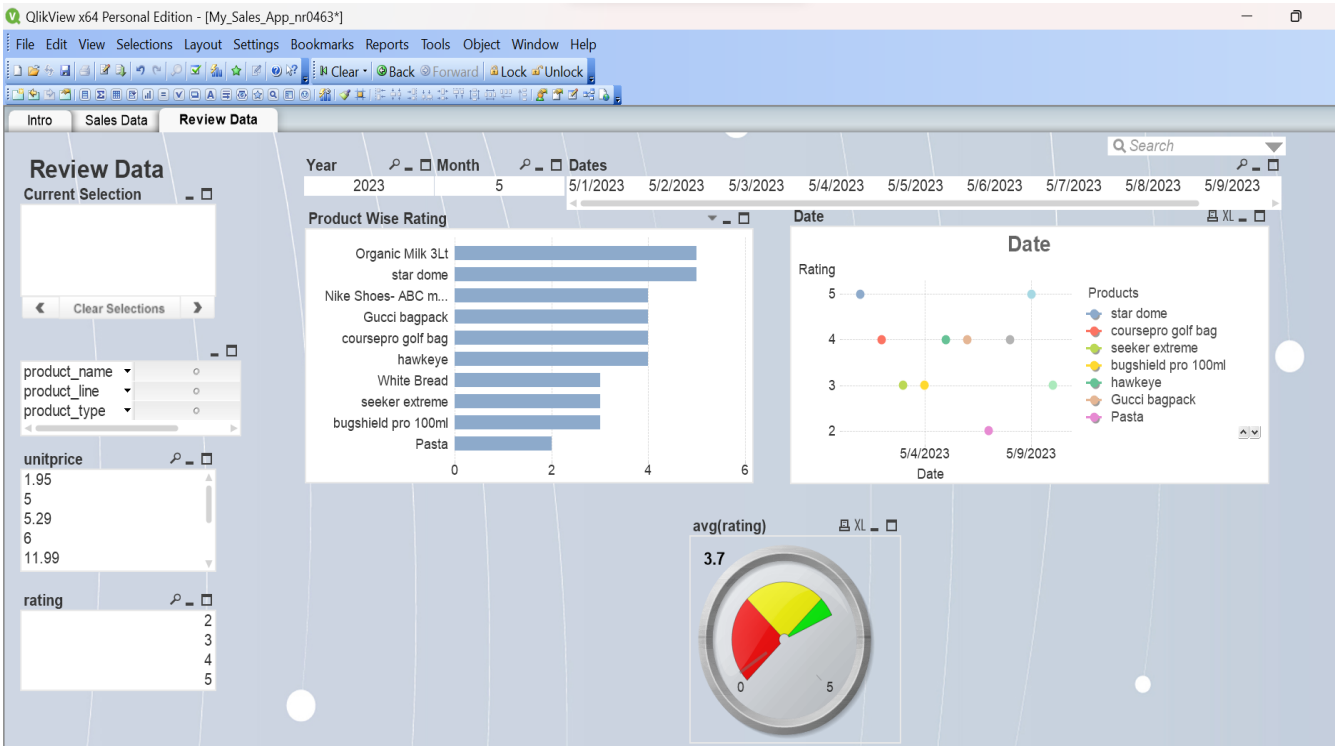
14. The Intro tab is the first landing page of the application giving brief introduction about the application purpose contains a get started button which directs to the sales data tab.
15. Sales tab contains Visualizations of sales with respect to dates, store zip and product in the form of bar chart, line chart and tree maps. The end user can choose the product/store zip/date to see the sales stats.



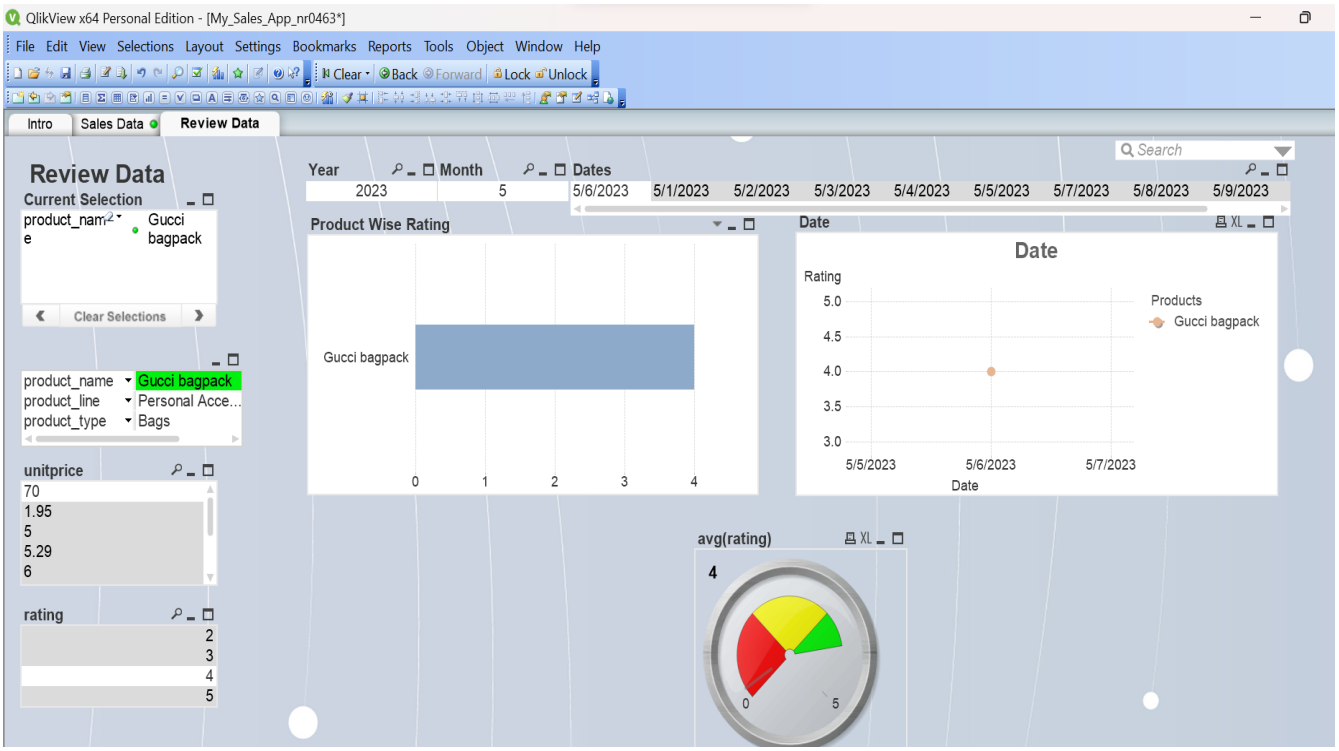
When we select any particular date or any other field, data relevant to that corresponding field will be reflected all through the sheet. For example here we have selected purchase date to be 5/1/2023 and we see data corresponding to that. We can click on clear section on the top left side to clear these selections and we will go back to previous view. The undo and redo buttons are present next to clear section.



16. In the reviews tab, we can see visualizations corresponding to review data. We have average rating for each product so far in the form of horizontal bar chart, a scatter plot to see the date and frequency of these ratings with respect to different products and a gauge chart to show the ratings.
17. For the data privacy issue, we have hidden the customer information here and we will not show their names here.



18. For example, when I choose ‘Gucci bagpack’ in the product name category on the left list box, it automatically fetches the values for its corresponding product line and product type. The product wise rating shows the product name and its rating whereas the scatter plot shows the date when that review was given, and gauge chart shows the ratings. The gauge chart becomes important when there are multiple reviews to be analyzed.



19. Clear selection and the undo and redo buttons work like the previous sheet.
20. The QlikView application is a perfect BI tool as it allows us to do in-memory processing which in the case of large data can improve performance immensely. Also, the inbuilt ELT capability allows us to resolve and transform loops thus helps us to work on clean data. This assignment helped us in designing dimensional modeling, importing the table structure along with its data to QlikView, performing ETL operations on top of that and then building front end application based on the backend we just created.