Tableau at UNH: Published Tableau Data Source Best Practices

UNH’s EIM team has gathered best practice information regarding Tableau data sources published based on information from the Tableau vendor as well as our own experiences. They may not be absolute rules, but should all be considered during development as appropriate.

Have a Tableau best practice to share? Please send it to us using our webform at: https://www.unh.edu/it/enterprise-information-management/webform-tableau-at-unh.

1. **Remove personally identifiable fields like employee/student name, USNH ID, SSN, or Banner Pidm**
   Why: reduces the risk of exposing sensitive data. Data visualizations are mostly aggregate displays, so personal identifiers are rarely, if ever, needed.

2. **Remove fields that are unlikely to be used for visualization or analysis.**
   Why: fewer fields reduces data source size and makes visuals more responsive. Do this step before creating the extract so Tableau will retrieve just the requested columns from the data source.

3. **Use a consistent naming standard for your data sources.**
   Why: helps you stay organized, especially if you will have several versions. Names should be self-descriptive enough so you can easily identify the data’s source, purpose, and content.

4. **Minimize the use of custom SQL in data sources.**
   Custom SQL is harder to maintain and keep track of because it is not readily viewable without downloading locally and opening with Tableau Desktop.

5. **Use extracts for data that is refreshed on a daily or longer cycle (e.g., R30).**
   Why: Extracts can significantly improve visual performance compared to using live connections.

6. **Schedule extract refreshes during non-business hours.**
   Why: Resources used for refreshing extracts take from those needed for viewing visuals.

7. **Ensure your data source gets the right permissions when publishing.**

8. **Review data source fields to ensure they are correctly set as dimension or measure.**

9. **Give self-descriptive names to data source fields.**
10. Use extract filters when only a subset of the data source’s data is needed. For example, you may only need to use the latest 3 years of data within a 15-year set.

11. Minimize the number of joined tables you have if you’re using a live connection.

12. Use full extract refreshes rather than incremental refreshes whenever possible. Why: full refreshes avoid the inherent confusion and complications surrounding incremental refreshes, which are additive only.