



Best Practices *for* Universe Design

by Frank F. Lee

frank.f.lee@thej3corporation.com

Universe Design Best Practices

- Most of the time we modify existing universes having many reports depending on them.
- A universe enables business users to view raw data in a sensible way.
- We will explore some of the best practices that will enable ease of maintenance and good user experience.

Format of the Presentation

- Interactive and technical in nature.
- Use Designer and Desktop Intelligence to demonstrate the key points in a sample universe.
- Questions and comments are welcome.

Think for the Users

- When working with the end users, assume that they know nothing about the database structure.
- Don't use raw database column names.
- Define default format for objects.
- Let's do a quick health check on a sample universe in Designer....

Use Aggregate Function in Measure Objects

- If your measure objects don't have aggregate functions such as SUM and COUNT, the reporting side (WebI) most likely is getting too many rows then necessary.
- Be careful with the COUNT function when dealing with semi-additive data. Use "Database delegated" property (Works in WebI only).
- Let's look at an example...

How Reports are Affected when Objects are Modified

- Add, move, rename objects. (Small impacts)
- Change object type. (Small impacts)
- Delete objects. (Big impacts. Label them first and put them in a specially labeled class.)

WHERE Clauses in Dimension Objects

- They will be combined in the SQL Statement by the AND operator. You could end up with a query that returns no data.
- Consider using CASE WHEN in the SELECT clause instead.
- Let's look at an example...

Why Bother with Testing if the Object Parses?

- Add a new table, add a join and add a new object from the new table.
- If the new join and the new object parse without a problem, is there a need to test further?
- Check the data, isolated join, loops.
- Let's look at an example...

Flexibility and Maintainability

- Use standard functions and standard SQL whenever possible.
- Where to keep the logic? In reports, universe, function/views or in ETL?
- Examples: YTD Sales (table, case when, store the data), Account groupings in reports such as Income Statement.
- The key is to know the implications.

Dealing with Extremely Difficult Reports

- Pre-calculate data in table columns
- Build a customized table and map to a set of objects that contains rows that are ready to be displayed. This is ideal for calculating values that require complex logic and traversing multiple rows of data.
- This solution is expensive and should only be used in extreme cases.

How many Universes should You Have

- Benefits and limitations of Derived universe.
- Does it make sense to separate universes for ad hoc reports and canned reports?

Summary

- Keep future maintenance in mind when designing and modifying a universe.
- Understand the problem, the data and the SQL before jumping into the tool.
- Treat end users as customers.
- Do integrity check and testing in the reporting tool before deploying your universe.