

# Getting the Most out of Diver Platform

Dan Jablonski

# Getting The Most out of Diver Platform



Putting the pieces together to optimize Diver Platform performance

- ▶ Modular Architecture
- ▶ Data Extraction and Intake
- ▶ Measure Factory Scopes
- ▶ Invalid Data
- ▶ Quickviews
- ▶ Caching



# Getting The Most out of Diver Platform

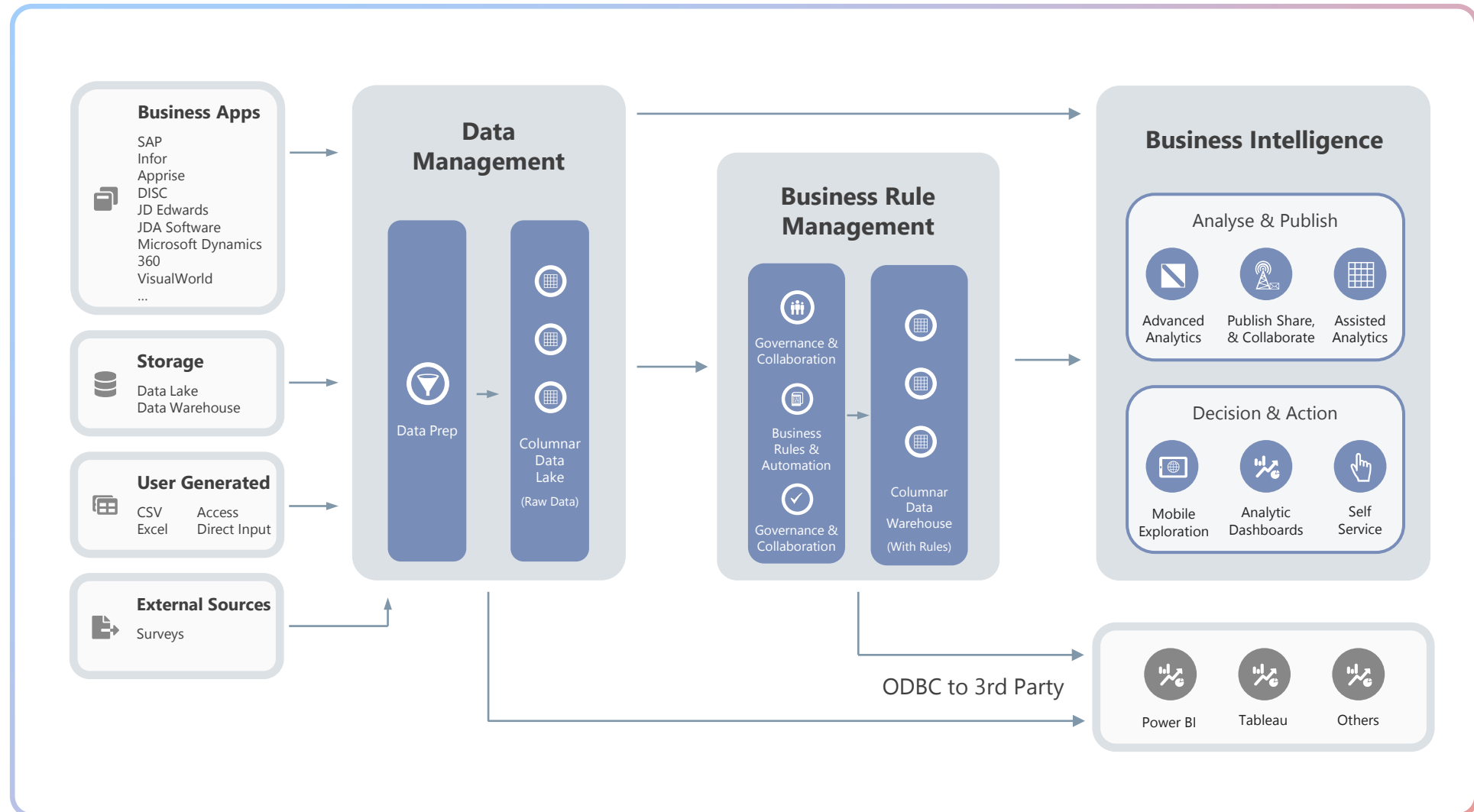


Putting the pieces together to optimize Diver Platform performance

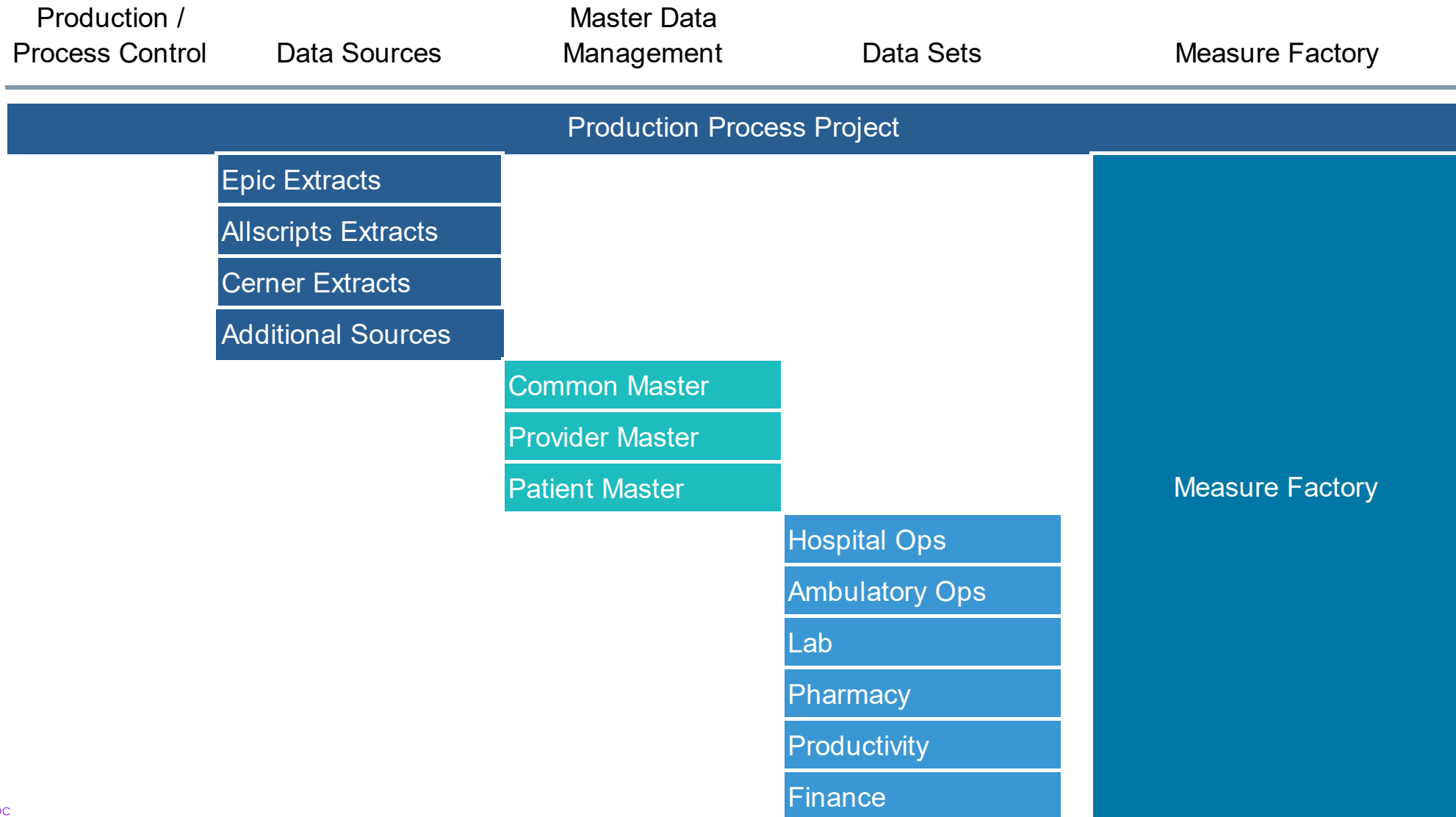
- ▶ **Modular Architecture**
- ▶ Data Extraction and Intake
- ▶ Measure Factory Scopes
- ▶ Invalid Data
- ▶ Quickviews
- ▶ Caching



# Diver Platform Architecture



# Projects Allow Logical, Modular Data Flow



# Getting The Most out of Diver Platform



Putting the pieces together to optimize Diver Platform performance

- ▶ Modular Architecture
- ▶ **Data Extraction and Intake**
- ▶ Measure Factory Scopes
- ▶ Invalid Data
- ▶ Quickviews
- ▶ Caching



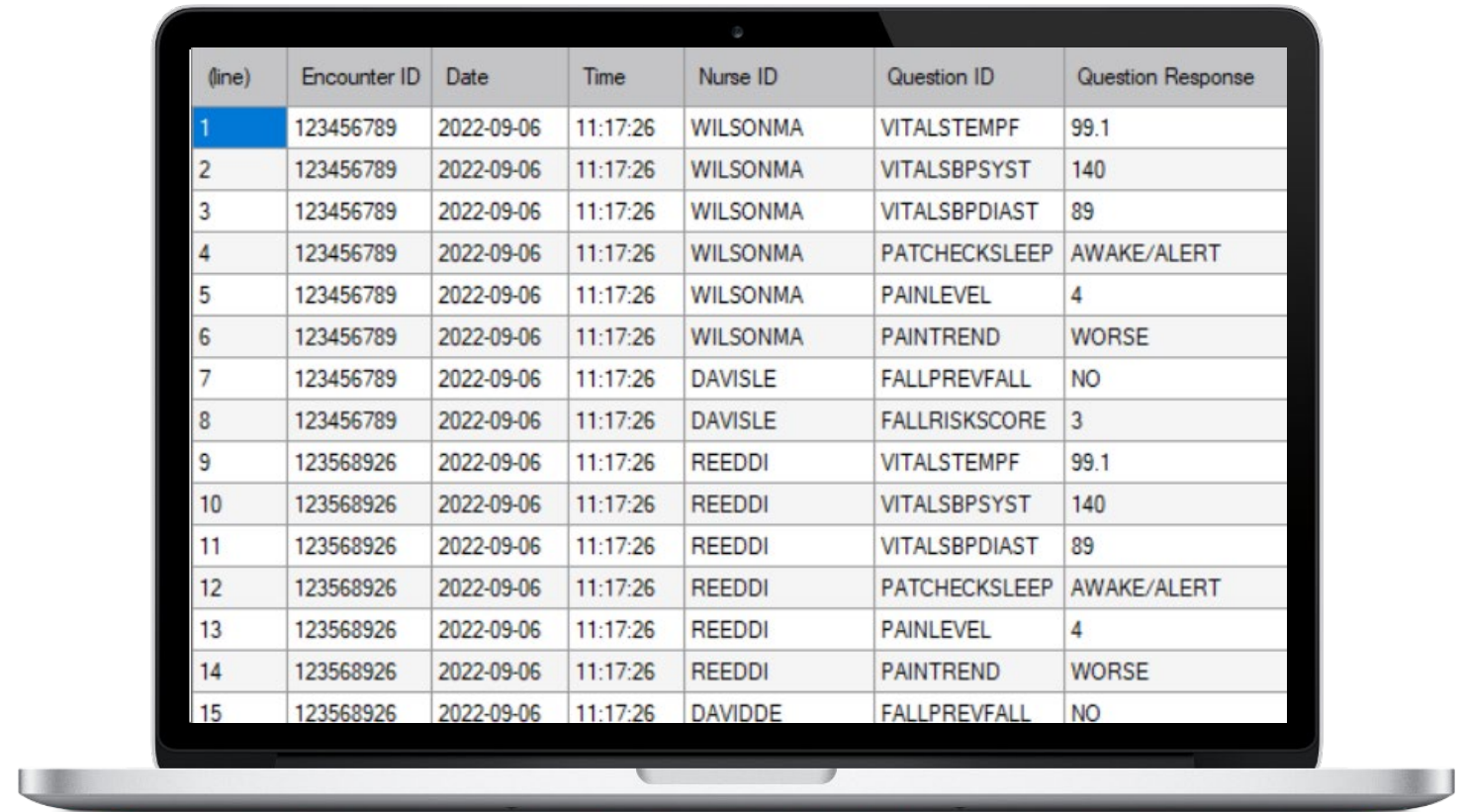
# Data Extraction and Intake

Divide and Conquer!

- Start from existing data set logic
- Counts and Amounts validation first
- Remove filters?
- Start adding additional measures
- Specialized queries get repetitive

# Data Extraction: Nursing Interventions

- Frequently used in queries
- Heavily Filtered
- Many small queries add up
- Pull Incrementally to cBase
- Updates take time – once.
- Reuse is easy.



| (line) | Encounter ID | Date       | Time     | Nurse ID | Question ID   | Question Response |
|--------|--------------|------------|----------|----------|---------------|-------------------|
| 1      | 123456789    | 2022-09-06 | 11:17:26 | WILSONMA | VITALSTEMPF   | 99.1              |
| 2      | 123456789    | 2022-09-06 | 11:17:26 | WILSONMA | VITALSBPSYST  | 140               |
| 3      | 123456789    | 2022-09-06 | 11:17:26 | WILSONMA | VITALSBPDIAS  | 89                |
| 4      | 123456789    | 2022-09-06 | 11:17:26 | WILSONMA | PATCHECKSLEEP | AWAKE/ALERT       |
| 5      | 123456789    | 2022-09-06 | 11:17:26 | WILSONMA | PAINLEVEL     | 4                 |
| 6      | 123456789    | 2022-09-06 | 11:17:26 | WILSONMA | PAINTREND     | WORSE             |
| 7      | 123456789    | 2022-09-06 | 11:17:26 | DAVISLE  | FALLPREVFALL  | NO                |
| 8      | 123456789    | 2022-09-06 | 11:17:26 | DAVISLE  | FALLRISKSCORE | 3                 |
| 9      | 123568926    | 2022-09-06 | 11:17:26 | REEDDI   | VITALSTEMPF   | 99.1              |
| 10     | 123568926    | 2022-09-06 | 11:17:26 | REEDDI   | VITALSBPSYST  | 140               |
| 11     | 123568926    | 2022-09-06 | 11:17:26 | REEDDI   | VITALSBPDIAS  | 89                |
| 12     | 123568926    | 2022-09-06 | 11:17:26 | REEDDI   | PATCHECKSLEEP | AWAKE/ALERT       |
| 13     | 123568926    | 2022-09-06 | 11:17:26 | REEDDI   | PAINLEVEL     | 4                 |
| 14     | 123568926    | 2022-09-06 | 11:17:26 | REEDDI   | PAINTREND     | WORSE             |
| 15     | 123568926    | 2022-09-06 | 11:17:26 | DAVIDDE  | FALLPREVFALL  | NO                |



# Data Extraction and Intake

Divide and Conquer!

- Specialized queries get repetitive
- Pull individual tables separately
- Use indexed dates for incremental pulls
- Build a data lake!
- Construct the data set using Spectre for ETL

# Getting The Most out of Diver Platform



Putting the pieces together to optimize Diver Platform performance

- ▶ Modular Architecture
- ▶ Data Extraction and Intake
- ▶ **Measure Factory Scopes**
- ▶ Invalid Data
- ▶ Quickviews
- ▶ Caching



# Measure Factory Data:

- Data Sets: Transactional Records at a level of detail:
- Scopes: Aligned view of a data set:
  - Alias dates and other dimensions to match the view
  - Filter for relevant transactions
  - Multiple scopes on a single data set are common



The image shows a laptop screen with a table titled 'MONTH-TO-DATE'. The table has four columns: 'Measures', 'Aug-2013', 'Jul-2013', and '% +/-'. Under the 'Measures' column, there is a section header 'Volumes' followed by several rows of data. The data shows a general decrease in most metrics from July to August 2013, with 'Patient Days' and 'ED Admissions' showing significant decreases.

| Measures          | Aug-2013 | Jul-2013 | % +/-   |
|-------------------|----------|----------|---------|
| <b>Volumes</b>    |          |          |         |
| Admissions        | 1,114    | 1,187    | -6.1    |
| Acute Admissions  | 789      | 831      | -5.1    |
| Discharges        | 1,176    | 1,169    | 0.6     |
| Surgical Patients | 180      | 167      | 7.8     |
| Patient Days      | 4,664    | 5,424    | -14.0 ▾ |
| Discharge Days    | 5,289    | 5,689    | -7.0    |
| Newborns          | 99       | 104      | -4.8    |
| ED Admissions     | 373      | 426      | -12.4 ▼ |

# Measure Factory Tips

## Avoiding Redundancy in Scopes

- Each scope becomes a separate cBase.
- Memory requirement: enough to hold all the cBases
- Where aliases are used to align the data, separate scopes cannot be avoided
- Where filters are used to segment the data, scopes may be redundant
- Avoid filtered scopes that add up to 2-3x the size of the data
- Filters can be done in measures

# Filters in Scopes

```
data-sets {
  data-set "Orders" {
    cbase-input "/sales_data/transactions.cbase"
  }
}
scopes {
  scope "Orders"
  scope "Shipments" data-set="Orders" {
    filter `value("Ship Date") != null`
  }
  scope "Open Orders" data-set="Orders" {
    filter `value("Ship Date") = null and value("Order Status") = "Active"`
  }
}
measures {
  category "Sales" {
    measure "Total Orders" `sum(value("Extended Amount"))` format="$#,##0.00" {
      scope "Orders"
    }
    measure "Total Sales" `sum(value("Extended Amount"))` format="$#,##0.00" {
      scope "Shipments"
    }
    measure "Total Open Orders" `sum(value("Extended Amount"))` format="$#,##0.00" {
      scope "Open Orders"
    }
  }
}
```

# Filters in Measures

```
data-sets {
  data-set "Orders" {
    cbase-input "/sales_data/transactions.cbase"
  }
}
scopes {
  scope "Orders"
}
measures {
  category "Sales" {
    measure "Total Orders" `sum(value("Extended Amount"))` format="$#,##0.00" {
      scope "Orders"
    }
    measure "Total Sales" `sum(value("Extended Amount"))` filter=`value("Ship Date") != null` format="$#,##0.00" {
      scope "Orders"
    }
    measure "Total Open Orders" `sum(value("Extended Amount"))` filter=`value("Ship Date") = null and value("Order Status") = "Active"` format="$#,##0.00" {
      scope "Orders"
    }
  }
}
```

# Getting The Most out of Diver Platform



Putting the pieces together to optimize Diver Platform performance

- ▶ Modular Architecture
- ▶ Data Extraction and Intake
- ▶ Measure Factory Scopes
- ▶ **Invalid Data**
- ▶ Quickviews
- ▶ Caching





# What is <UNKNOWN>

- We can't produce a value of the specified type that accurately reflects the input data
  - Invalid input: "N/A" in a numeric column
  - Invalid input: 2021/02/30 in a date column
  - Invalid math operations: Divide by zero
- Stored as the special value <UNKNOWN>
- "Breaks" math:  $1 + \text{<UNKNOWN>} = \text{<UNKNOWN>}$
- "Breaks" date rollup: can't calculate YTD if <UNKNOWN> dates exist





# Managing <UNKNOWN>

- In inputs, read as a string, and do error checking
- Decide on remedy:
  - Null/zero
  - Filter
  - Additional processing/reporting
- In measure definitions: `catch()`
- Do you trust your data sources?

# Getting The Most out of Diver Platform



Putting the pieces together to optimize Diver Platform performance

- ▶ Modular Architecture
- ▶ Data Extraction and Intake
- ▶ Measure Factory Scopes
- ▶ Invalid Data
- ▶ **Quickviews**
- ▶ Caching



# Quickview Performance

Divide and Conquer!

- Quickviews specify filters to be applied to page content
- Searchable quickviews allow quick access to one of many values
- But.. Too many values will slow down your page
- Use mandatory higher level selections to minimize the number of values in very detailed quickviews



# Quickviews in Popup Detail Pages

- Quickviews get populated in two ways:
  - User selection
  - Value passed via click action
- Hidden Quickviews are good when just “receiving a message”
- ProDiver free entry quickviews can be used with marker-based portlets to streamline value passing of high-dimcount dimension values

# Getting The Most out of Diver Platform



Putting the pieces together to optimize Diver Platform performance

- ▶ Modular Architecture
- ▶ Data Extraction and Intake
- ▶ Measure Factory Scopes
- ▶ Invalid Data
- ▶ Quickviews
- ▶ **Caching**



# Spectre Cache

Refresh slower dives as part of the build process

- Spectre caches the results of dives, so they don't need to be repeated
- After cBase builds, cache entries become “stale”
- Cache Refresh replays stale entries to make the cached data “fresh” again
- Clean up after yourself – delete entries that are outdated

# Cache Refresh with Predicate

- Cache Refresh can be optimized to run the most “important” entries – those that actually get used, and recently.
- Check in on your cache refresh periodically!

```
--refresh-with-predicate arg
```

Take an expression and evaluate it; if it is true, refresh the **cache** entries. This option is useful if the criteria are more complex.

The expression runs over a virtual table with the following columns:

- "age": The number of hours since this **cache** entry was last updated
- "idle": The number of hours since this **cache** entry was last read
- "num\_hits": The number of queries that were satisfied by reading this **cache** entry
- "num\_updated": The number of times this **cache** entry has been updated
- "user": The current user name
- "size": The size of this **cache** entry in bytes
- "time\_to\_generate": The time to generate this **cache** in seconds

# Getting The Most out of Diver Platform



Putting the pieces together to optimize Diver Platform performance

- ▶ Modular Architecture
- ▶ Data Extraction and Intake
- ▶ Measure Factory Scopes
- ▶ Invalid Data
- ▶ Quickviews
- ▶ Caching

