

SAS Data Integration Studio in the workplace

(or "I wish I had this when...")

Andrew Howell
ANJ Group Pty Ltd
ahowell@anjgroup.net.au

Agenda

N.B: NOT a Data Integration Studio demonstration

Use actual scenarios (i.e, “bitter experience”)

Examine the relationship between:

- The traditional SAS programming environment
- The newer interfaces (Enterprise Guide, DI Studio)

Very important:

- This is not an interface war
 - Programming vs GUI
 - *Apple Mac vs Lisa, VHS vs BetaMax, etc*
- GUI & code can (and should!) co-exist

SAS Programming

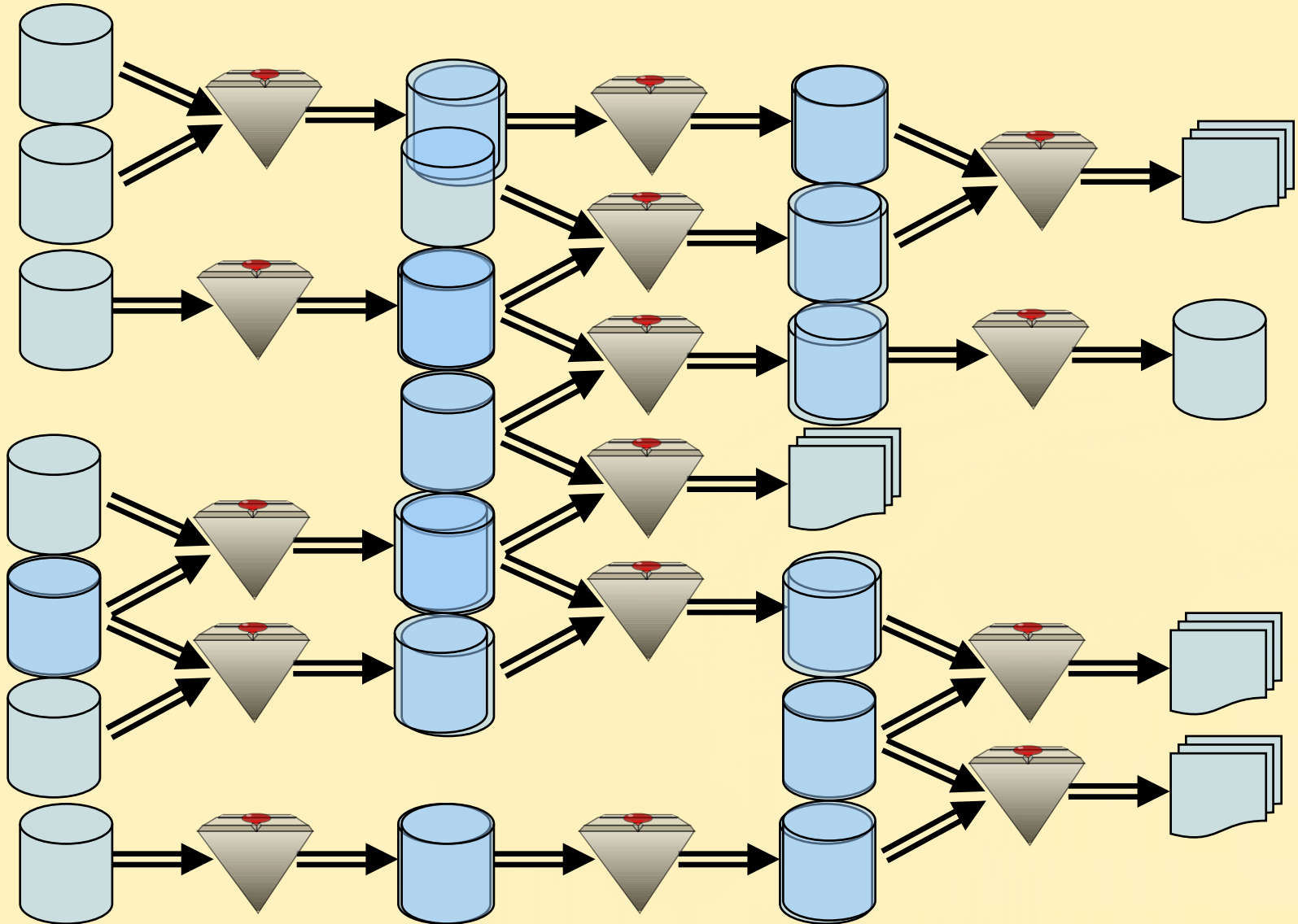
Script

- Flexible!
- Program how you want, to get the solution you want
- Completely customised solution

However...

- Difficult to manage “free text”
- How to
 - gauge impact of change
 - scope modifications
 - trace results back to source systems
- Rarely commented sufficiently
- Difficult to remain consistent

SAS Programming



SAS Data Integration Studio



downloaded from SAS products web page

SAS Data Integration Studio - Features

- An easy-to-use, point-and-click GUI uses an intuitive set of configurable windows for managing data integration development processes.
- A visual end-to-end process designer lets developers quickly build and edit processes.
- Drag-and-drop functionality eliminates programming.
- Wizards make it easy to access source systems, create target structures, import and export metadata, and build and execute ETL process flows.
- Role-based permissions show users only what they are authorized to see.
- Customizable metadata tree views let users display, visualize and understand metadata.
- Check-in/check-out of jobs, related tables and objects; and job status viewing.
- Audit history lets designers see which jobs or tables were changed, when and by whom.

downloaded from SAS products web page

Actual Scenarios

*(Names & places have been changed
to protect the guilty..)*

Exception Reporting

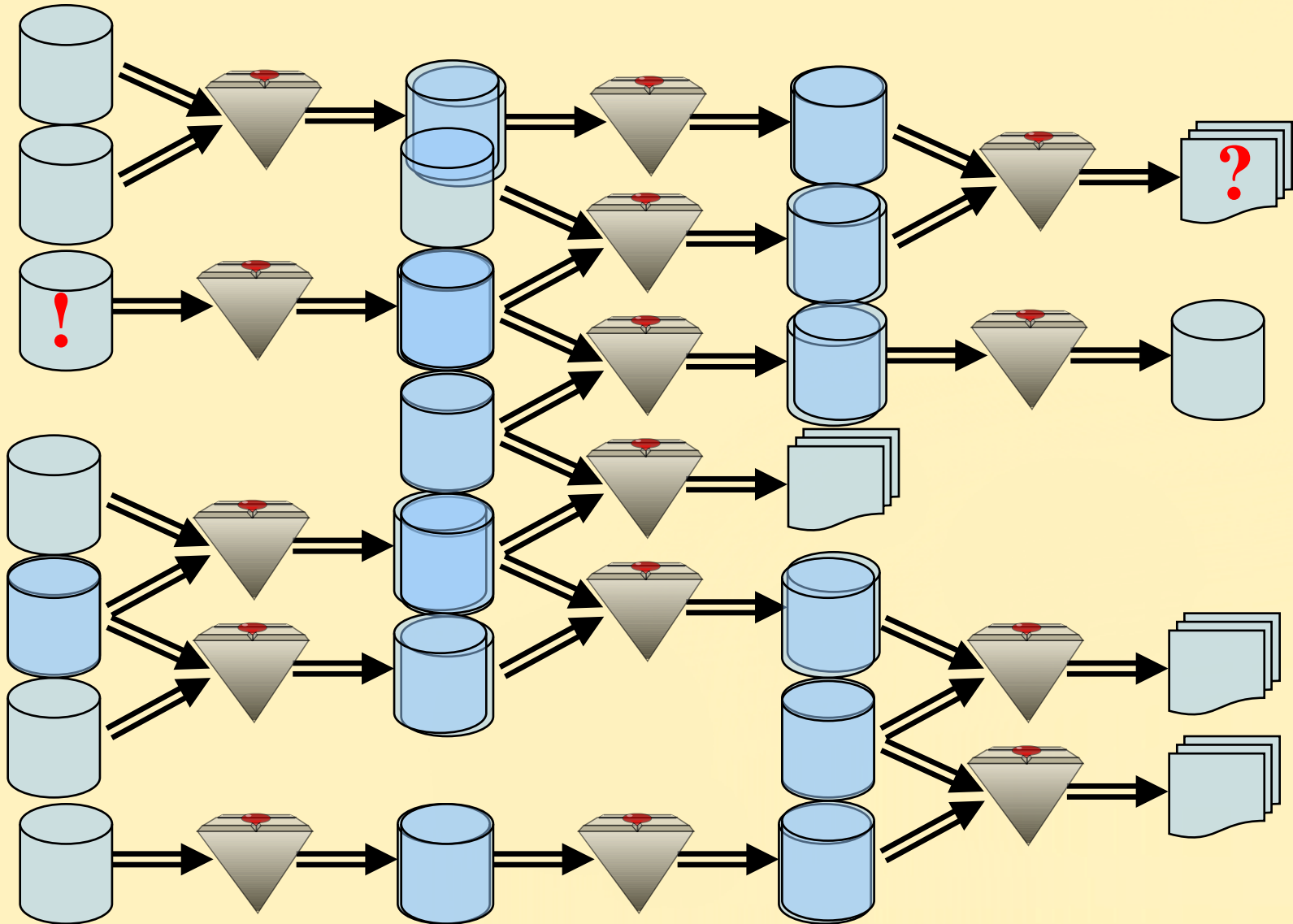
Background

- All operational stored in RDBMS data warehouse
- SAS used to consolidate operational data into reporting warehouse, then generate reports (HTML, PDF, Excel)

Problem

- Trace exception reporting back to source tables?

Exception Reporting



Fault finding

Background

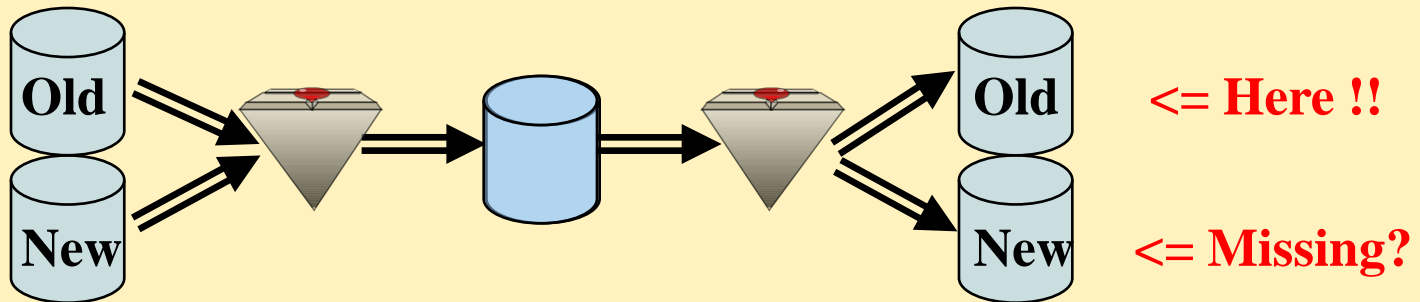
- Old & New data warehouses – 2 payment systems
- Each month, both warehouses are combined, updated, then re-split back to Old & New warehouses

Problem

- Some records in New data warehouse go missing after their first month's processing.

Fault finding

“Needle in a haystack”



```
dat a ol d_dwh new_dwh;  
  
  i f dat e_f i e l d < ' 01Jan2000' d t h e n  
    o u t p u t ol d_dwh;  
  e l s e  
    o u t p u t new_dwh;  
  
r u n;
```

Data Warehouse – data feeds

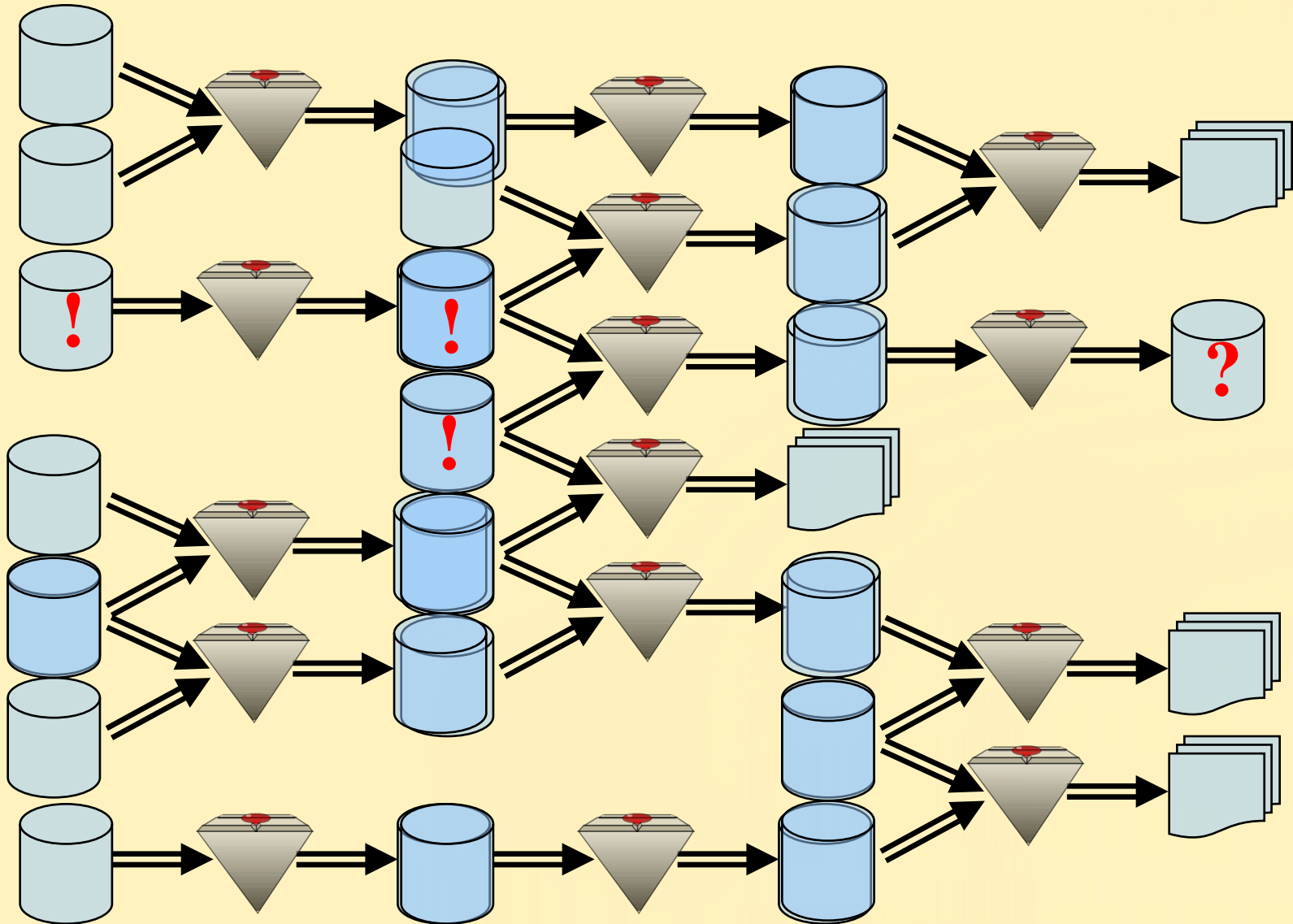
Background

- Data warehouse load – 100+ SAS programs
- Working fine for years
- Programmers, contractors come & gone
 - Different development practices

Problem

- Suddenly generating empty data sets & unresolved macros

Data Warehouse – data feeds



Data Cleansing

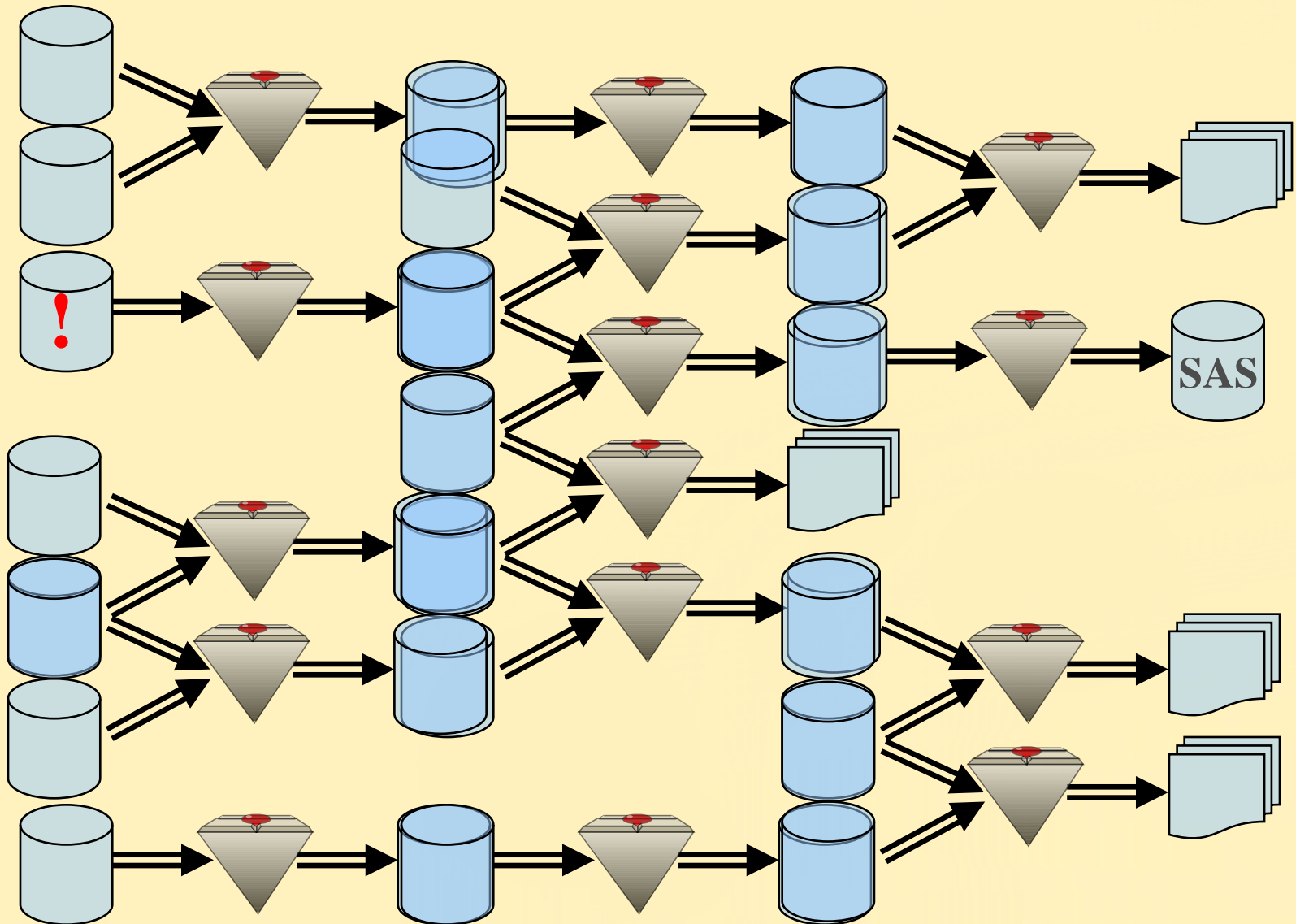
Background

- Data loaded from various operational sources prior to loading into a SAS solution
 - Cleansing
 - Validation
 - Etc

Problem

- Quickly identify faulty data and/or processes

Data Cleansing



Data Warehouse Upgrade

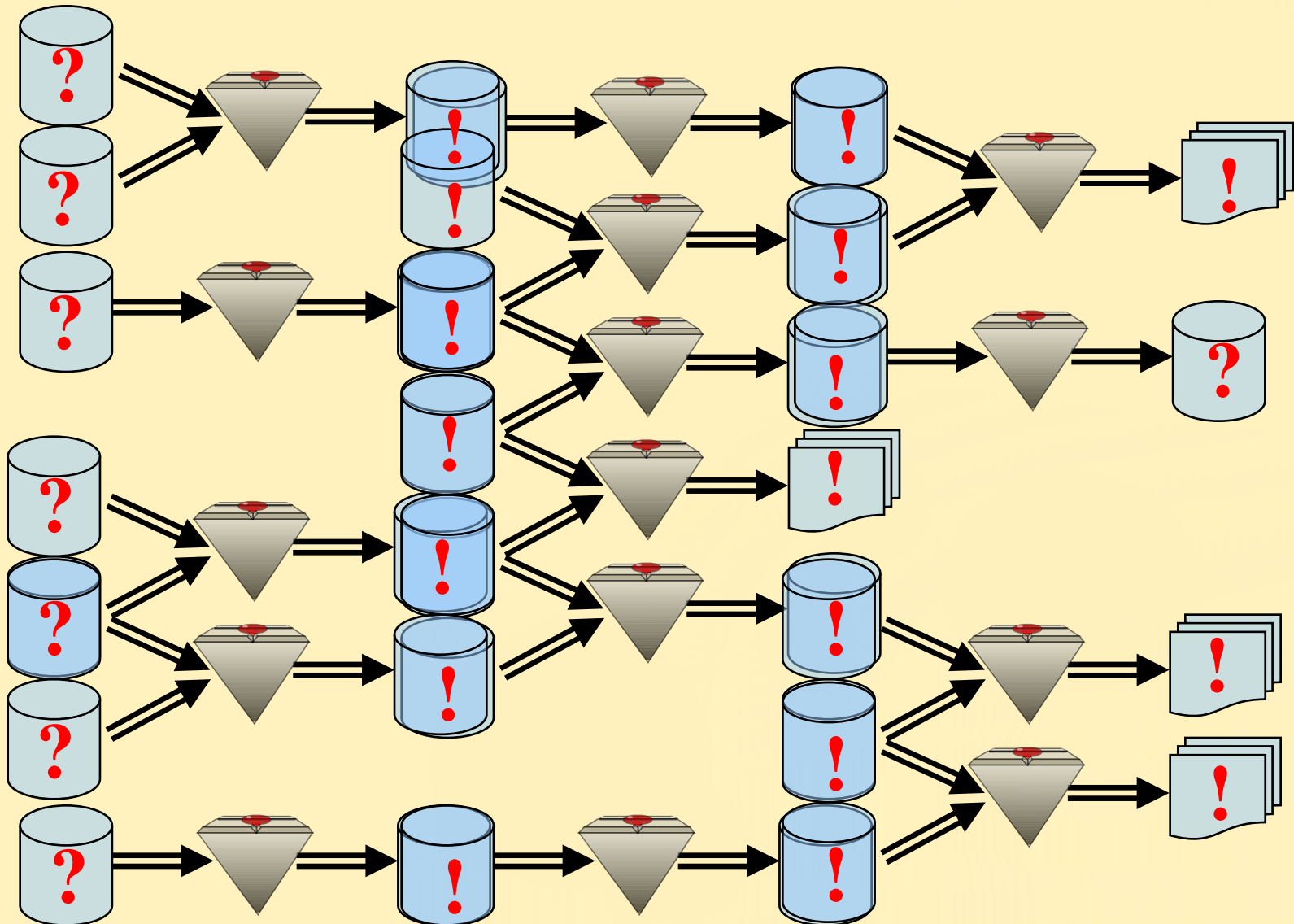
Background

- Operational data warehouse
- Analytical & reporting data marts
- SAS used for ETL.
- Initially small number of products, single character key field (A-Z). length ID \$ 1 ;
- As more products, also lower case & numeric
- After ($26 + 26 + 10 = 62$) current & retired products?

Problem

- Migrate all SAS processes to new 8 character key field
- How to scope work ?

Data Warehouse Upgrade



Data Migration Project

- SAS customer for 15+ years
 - Lots (100s) of embedded SAS programs
- Major IT upgrade
 - 2+ year project to consolidate & retire many legacy systems into a new highly-customised object-oriented system
- SAS Data Integration Studio chosen as the migration path.
- Migration team
 - Internal staff
 - Consulting partner

Data Migration Project - benefits

- Requirements change over time
 - Able to quickly identify impact of change / rework.
- Able to quickly trace issues back to data and/or processes.
- Use DI processes where appropriate & use customised code where appropriate (“Best of Both Worlds”)
- Create customised (re-usable) processes
- Effectively prioritise development
- Change controlled environment – check-out/check-in.
- Easily deploy Dev to Test to Production.

Data Migration Project - issues

Early adoption

- Learn as we go
- Too early to know “Best Practices”

Repository-based Check-In

Too many repositories

Scalability of metadata server

Most of been resolved by

- Product evolution
- Wisdom of usage – SAS support, other sites

SAS code & Data Integration Studio

SAS Code

- how long to identify problem
- how long to scope solution
- how to manually maintain & document relationship between all processes.

D.I. Studio

- centrally manage all processing
- rapidly determine impact of change
- trace results back to source data / processes

Don't Panic

You can still code!

Data Integration Studio

- SAS code, SAS Stored Processes, Java

Enterprise Guide

- SAS code, SAS Stored Processes, .Net

Web Report Studio

- SAS Stored Processes

Further reading

support.sas.com

- Whitepapers

SAS Global Forum

- Whitepapers

SasCommunity.org

- Online community