SAP BusinessObjects USER CONFERENCE

October 5-7, 2010 | Orlando, Florida
Walt Disney World Swan and Dolphin

Better Universes by Design

Alan Mayer
Solid Ground Technologies

Session 802

ASUG Real Experience. Real Advantage.

Agenda

- Introduction
- Filed-based universes
- Relational universes
- Programmatic universes
- Relational warehouses
- Multi-dimensional warehouses
- Dashboard universes
- Universes on SAP BW
- Conclusion

Introduction

Introduction

Introduction

Introduction

Introduction

Introduction

Introduction

- Data sources are the foundation of any universes
- Several "Universe Best Practices" presentations are available
  - "Universe Best Practices" – Alan Mayer ASUG 2009
  - "BusinessObjects Designer Essentials" – Alan Mayer BOBJ 2004
- Very few of them consider the effects on the data source
- Taking the structure and nature of data into account can have dramatic impacts
  - Performance/speed
  - Simplicity
  - Longer DBA lifespans
  - Shorter universe development

Agenda

- Introduction
- Filed-based universes
- Relational universes
- Programmatic universes
- Relational warehouses
- Multidimensional warehouses
- Dashboard universes
- Universes on SAP BW
- Conclusion

Flat File Universes

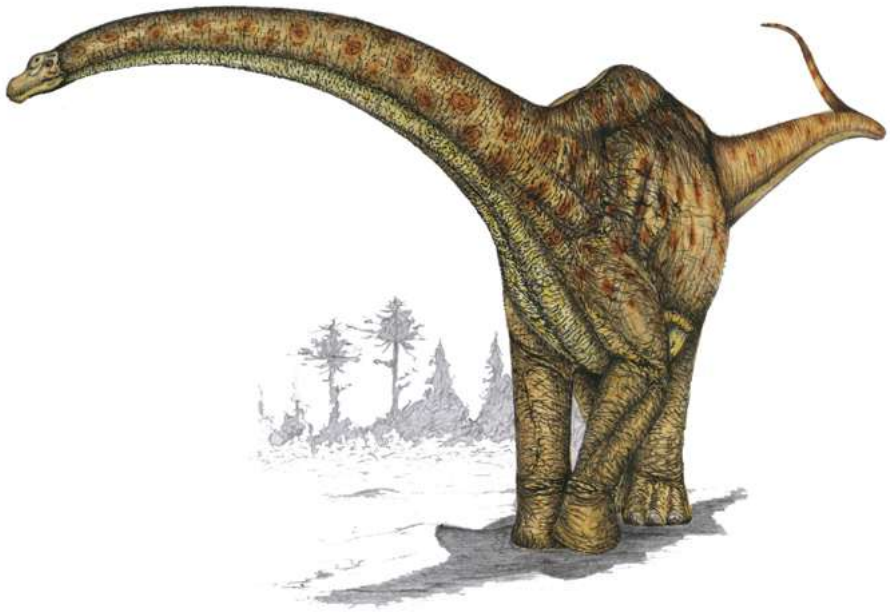
- Meant for modeling or previewing conceptual designs
- **NOT** designed for performance, scalability
- Restricted audience
- Universes can be built from
  - Text files
  - Excel workbooks
  - XML documents/schemas

Flat File Universes

- Demonstration

Agenda

- Introduction
- Filed-based universes
- Relational universes
- Programmatic universes
- Relational warehouses
- Multi-dimensional warehous
- Dashboard universes
- Universes on SAP BW
- Conclusion



Relational Universes

- Database structure is close to third normal form
- Translation:
  - Great for adding/modifying/deleting
  - Not optimized for reading or reporting
- This section will introduce some design principles that will make the most of this database type

Relational Universes

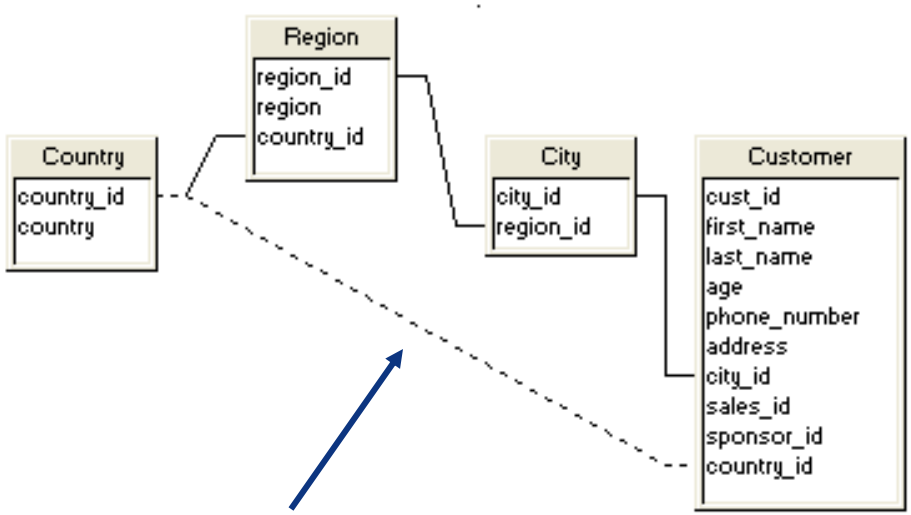
Principle 1: Minimize the number of joins in the final query

- Joins can be reduced by correctly using:
  - Shortcut joins
  - Summary tables w/aggregate awareness

Relational Universes

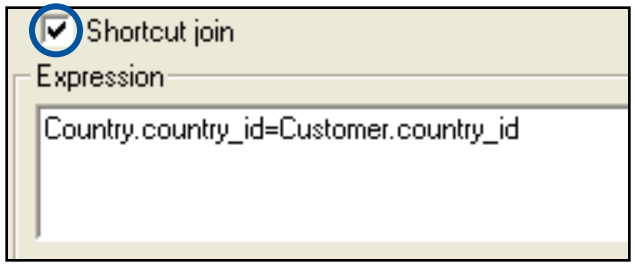
Shortcut Joins

- Provides a shortcut or alternative path between tables
  - The Customer table may contain an extra column that allows a direct join to Country



Shortcut Join

Join Editor



Relational Universes

Aggregate Awareness

- The only technique where a single object reacts to other objects in the same query
- Used to select the fastest/optimal table to retrieve data
- Originally meant for measures
- Steps involved in using Aggregate Awareness
  - Define the AggregateAware object
  - Define classes/objects incompatible with that object

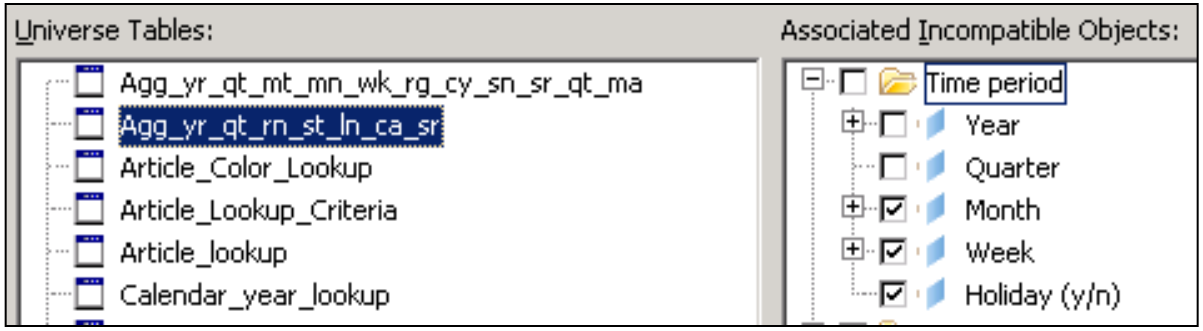
Relational Universes

Aggregate Awareness, cont'd

- Steps involved in using Aggregate Awareness:
  1. Define the AggregateAware object, fastest first

```
@Aggregate_Aware(
      sum(Agg_yr_qt_mt_mn_wk_rg_cy_sn_sr_qt_ma.Sales_revenue),
      sum(Agg_yr_qt_rn_st_ln_ca_sr.Sales_revenue),
      sum(Shop_facts.Amount_sold))
```

  2. Define incompatibilities



Relational Universes

Aggregate Awareness, cont'd

- Incompatibility is determined by the grain of the table

Class	Object	Incompatible ?
Time Period	Year	
	Quarter	
	Month	x
	Week	x
	Holiday (y/n)	x
Store	State	
	City	x
	Store Name	x
…		

Relational Universes

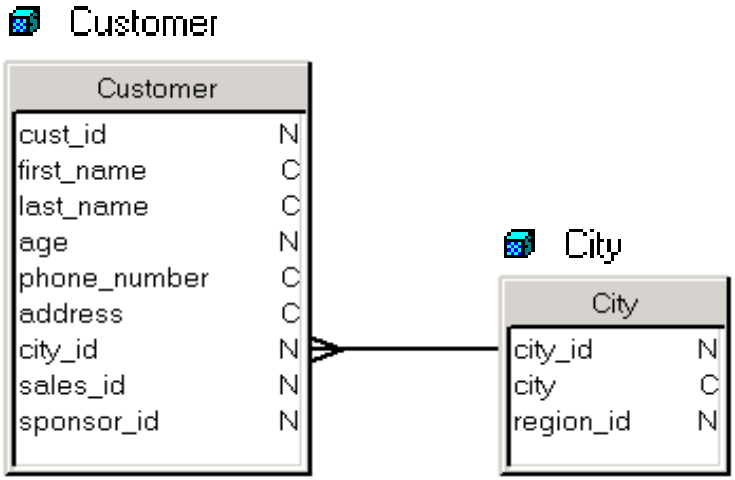
Principle 2: Take advantage of existing indexes

- Put existing indexes to work using the following techniques:
  - Index awareness
  - Predefined conditions
  - Object tagging

Relational Universes

Index Awareness

- As conditions are added …



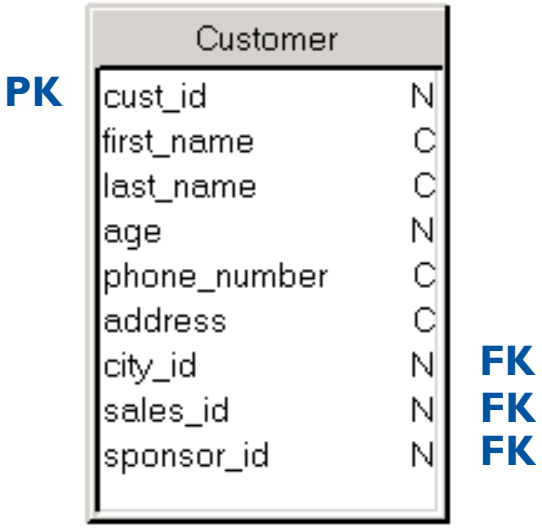
City in ('Dallas','Chicago')

- … query performance may suffer
  - Additional joins may be added
  - The columns(s) targeted by the condition may not be indexed

Relational Universes

Index Awareness, cont'd

- Primary and foreign keys are usually indexed

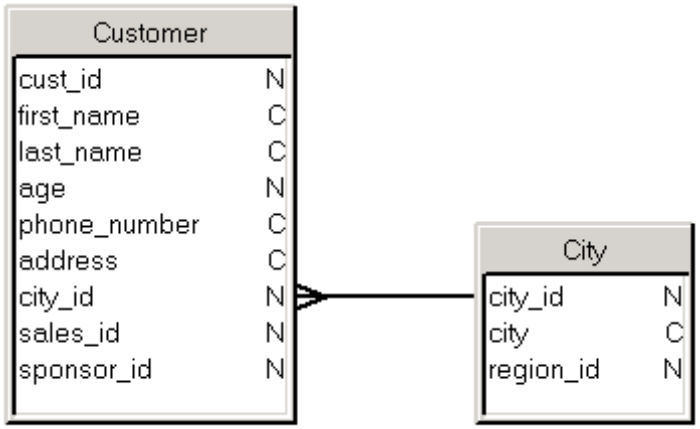


- Use the indexes to your advantage
  - Place the condition on the foreign key column
  - The index on that column will then be used and a join is no longer needed

Relational Universes

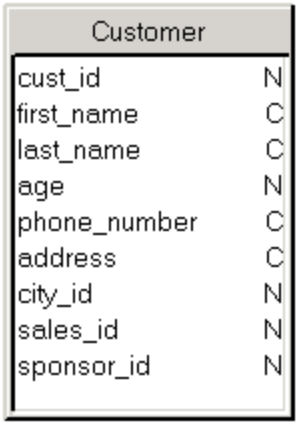
Index Awareness, cont'd

- Replace this …



Customer.city_id = City.city_id
and City.city in ('Dallas', 'Chicago')

- With this:



Customer.city_id in (11, 15)

Relational Universes

Predefined Conditions

- Condition objects can be created that will take advantage of existing indexes
- Conditions are placed only on indexed table columns
- These conditions can be set to prompt for values on those columns as well

Relational Universes

Object Tagging

- Label objects to inform the user that indexes have been applied
- Users would know that using that object in a condition would make their key faster

Relational Universes

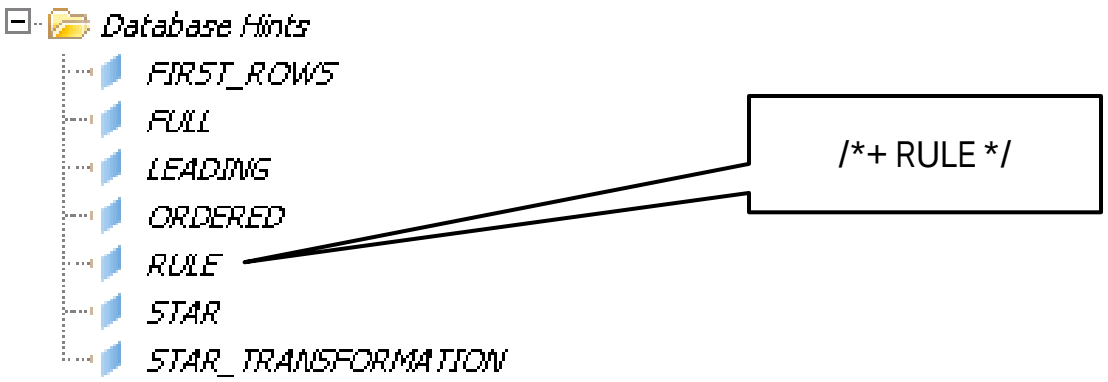
Principle 3: Selectively use database features

- Database vendors offer different techniques for accelerating queries
- Correct use of these technique can dramatically increase performance
  - Database hints
  - Dynamic parameters
  - Database-specific file parameters

Relational Universes

Database Hints

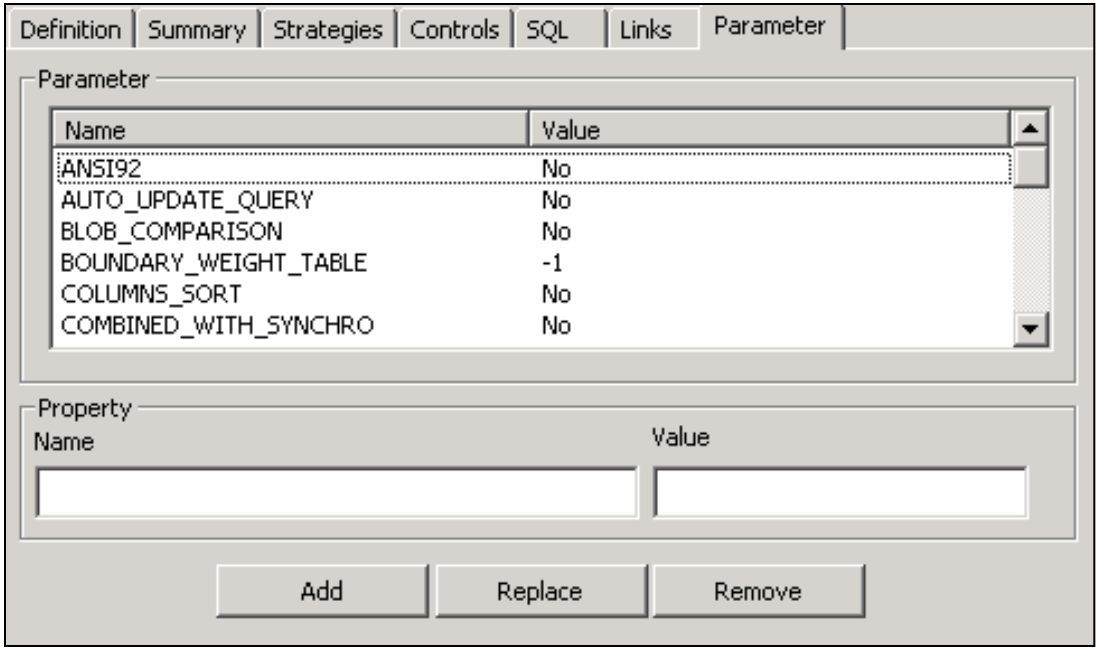
- NOT meant for ad-hoc universe in general
  - Objects *could* be hidden from public view
- Applicable for databases that use hints (Oracle)
- Objects are created that introduce the database hint
- Must be the first object added to a query



Relational Universes

Dynamic Parameters

- These parameters can expand or limit a universe's functionality



Relational Universes

Dynamic Parameters, cont'd

- Some of the more important candidates:

  - **DISTINCTVALUES**

    Controls whether DISTINCT or GROUPBY is used when retrieving unique rows. Used especially for List of Values queries
  - **END_SQL**

    Allows comments to be added at the end of every SELECT statement. DBAs can use to find the associated universe and user. Variations for different databases.

Relational Universes

Database Configuration Files

- Every database has an associated configuration file (.sbo)
- Parameters can be added which control how the database interprets and processes a request

Relational Universes

- Demonstration

Agenda

- Introduction
- Filed-based universes
- Relational universes
- **Programmatic universes**
- Relational warehouses
- Multi-dimensional warehouses
- Dashboard universes
- Universes on SAP BW
- Conclusion

Programmatic Universes

- Universes can also be based on top of existing programs:
  - Database stored procedures
  - Java Beans
  - …
- This design is **NOT** optimal but sometimes necessary
- The biggest drawback is the number of data sources per universe
  - Currently only one program/procedure per universe
- There is a way around this, however …

Programmatic Universes

- Dynamic Tables
  - Reference a stored procedure as a table
  - Not available through all database vendors
  - The following example will be Oracle-based.
- Implemented as a derived table within a universe
- Can be joined to other tables - carefully

Programmatic Universes

- Demonstration

Agenda

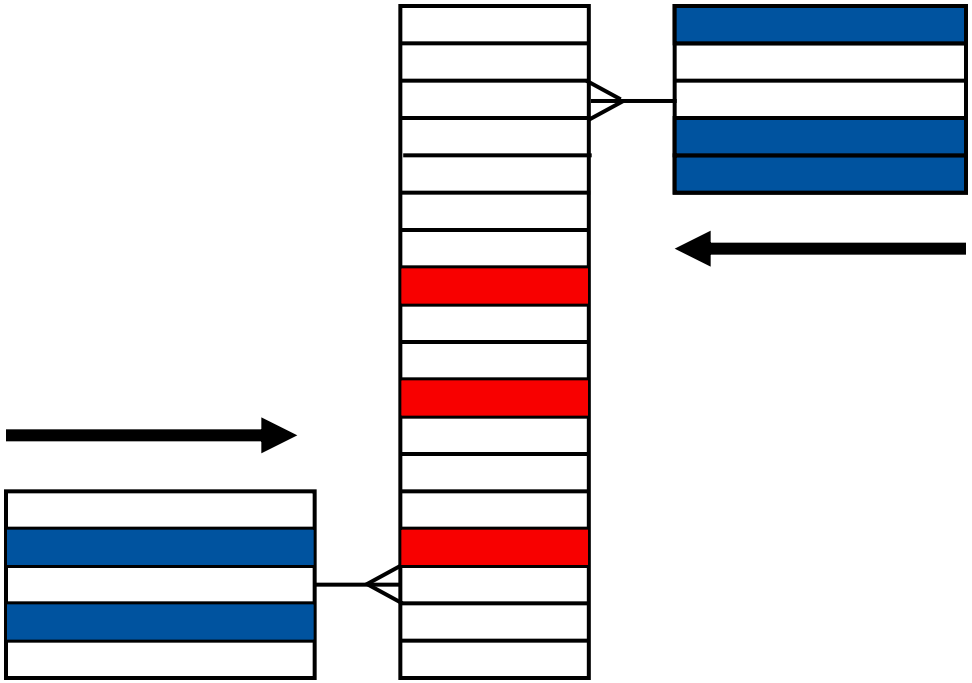
- Introduction
- Filed-based universes
- Relational universes
- Programmatic universes
- **Relational warehouses**
- Multidimensional warehouses
- Dashboard universes
- Universes on SAP BW
- Conclusion

Relational Warehouses

- Databases of this type are built as star/snowflake schemas
- The very nature of its design should shorten queries
- Several universe-related techniques and observations can build on those gains
  - Connection-based hints (where available)
  - Aggregate awareness
  - Optimization of joined / synchronized queries

Relational Warehouses

- Connection Hints
  - Add conditions on dimensions, collect surviving facts



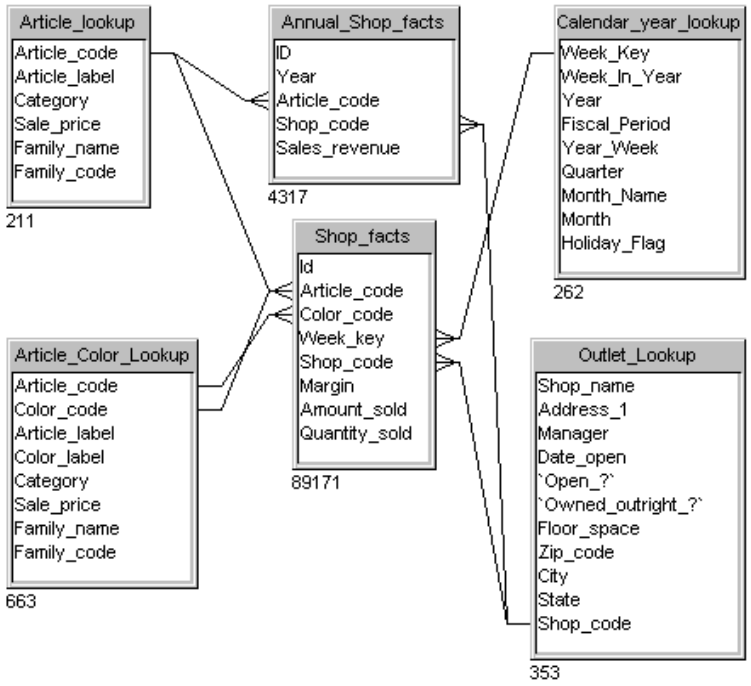
Relational Warehouses

- Connection Hints
  - Add hint that allows this behavior inside the connection
  - Oracle examples:

```
/*+ STAR */
/*+ STAR_TRANSFORMATION */
```

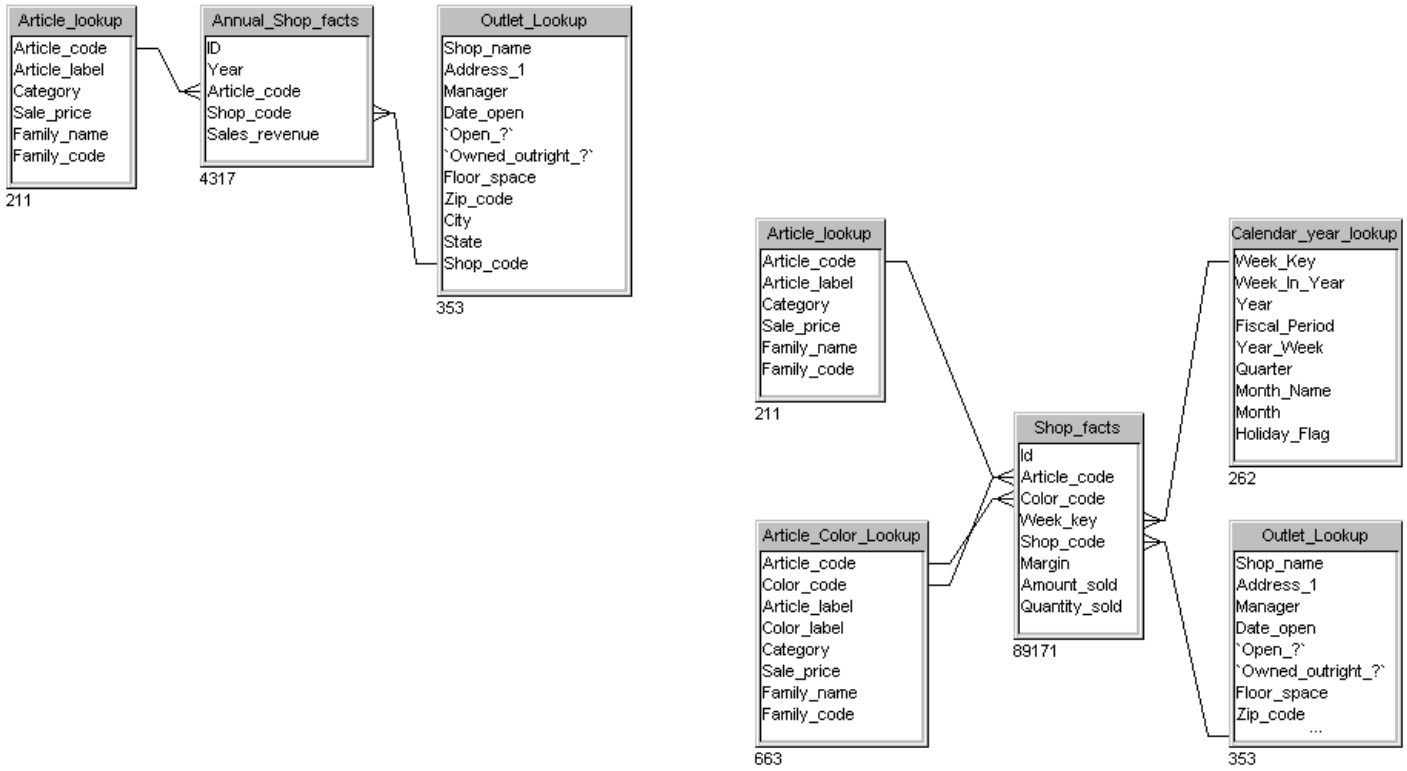
Relational Warehouses

- Aggregate Awareness
  - Each universe can be a collection of star schemas
  - Allows universe to switch between stars, retrieving the level of aggregation desired



Relational Warehouses

- Aggregate Awareness
  - Contexts are used to represent each star schema



Relational Warehouses

- Aggregate Awareness
  - @AggregateAware function is used to select between the stars
  - Incompatiblities are defined based on each context

Relational Warehouses

- Joined / Synchronized Queries
  - Occurs when measures are included from more than one star in a query
  - Users have no idea this is occurring in the background
  - This can turn a 5 query report into 50 to 100 SELECTs from a DBA perspective

Relational Warehouses

- Joined / Synchronized Queries, cont'd
  - If this occurs frequently, you may need to aggregate differently
    - Create a fact table at the grain that users are pulling information

Relational Warehouses

- Demonstration

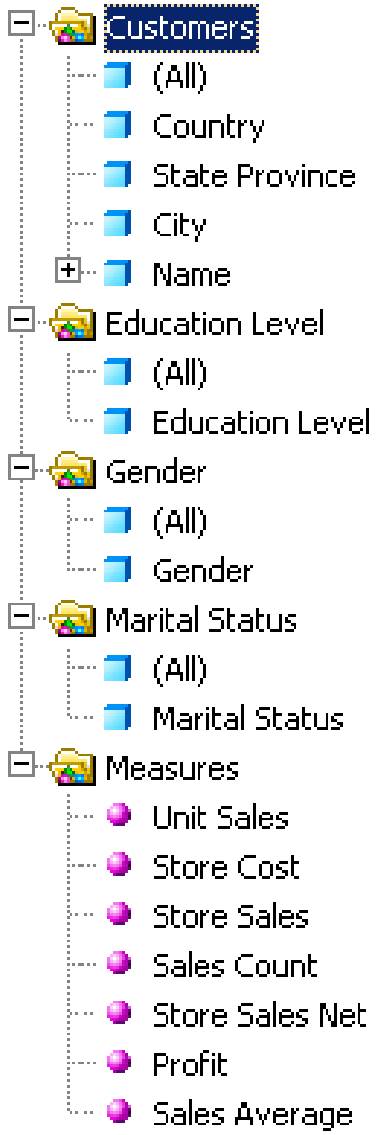
Agenda

- Introduction
- Filed-based universes
- Relational universes
- Programmatic universes
- Relational warehouses
- **Multi-dimensional warehouse**
- Dashboard universes
- Universes on SAP BW
- Conclusion

Multidimensional Warehouses

- No tables exist in multi-dimensional databases …
  - Just cubes
- There is no table structure to tune
- Even the language generated by the universe is different
  - MDX vs. SQL
- Past universes allowed rearranging / renaming of objects
- Recent software improvements allow a bit more control

Multidimensional Warehouses



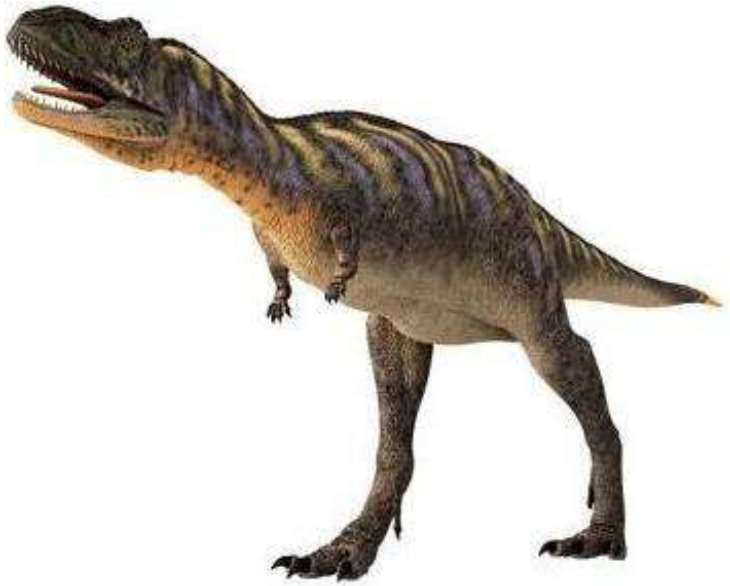
- Universe built against Microsoft SQL Server Analysis Services
  - Structure Panel is empty within Universe Designer
  - Universe creation is mostly automated
  - Clean-up tasks include:
    - Deleting extraneous classes and objects
    - Reorganizing classes
    - Renaming classes and objects

Multidimensional Warehouses

- Demonstration

Agenda

- Introduction
- Filed-based universes
- Relational universes
- Programmatic universes
- Relational warehouses
- Multi-dimensional warehouses
- Dashboard universes
- Universes on SAP BW
- Conclusion



Dashboard Universes

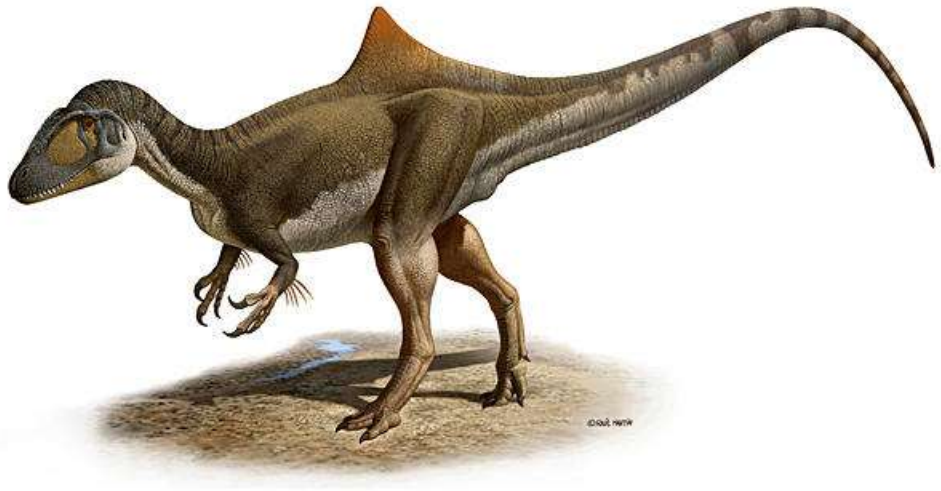
- Response time is critical to a dashboard user
  - Many SLAs are between 2 – 5 seconds per click
- Joins must be minimized or eliminated
  - Some joins to security tables may be necessary for personalization
- The resulting universe looks like pockets of tables that aren't connected in any way
  - No ad-hoc queries allowed!
- Tools like Xcelsius and Query as a Web Service (QaaWS) benefit from this type of structure

Dashboard Universes

- Demonstration

Agenda

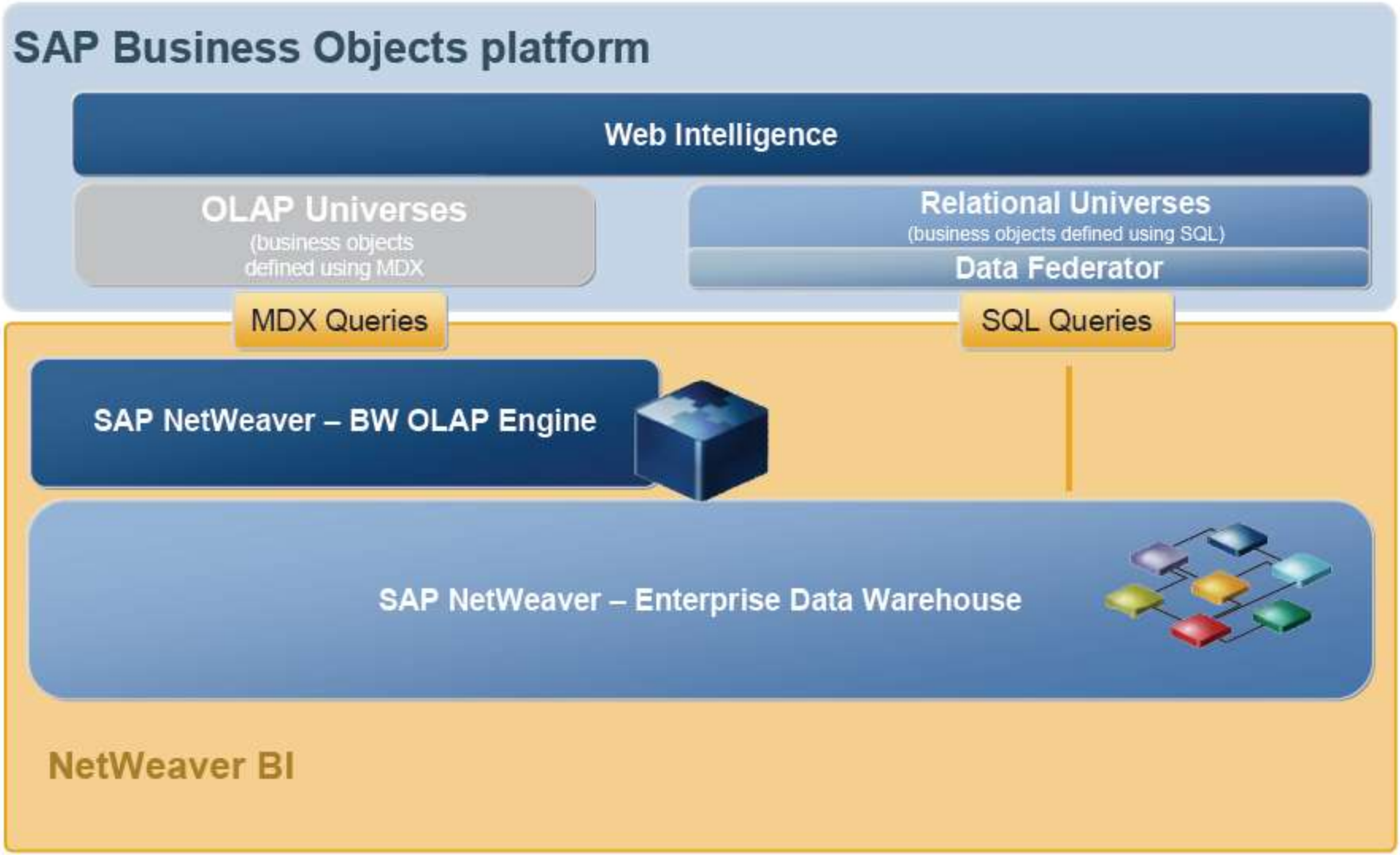
- Introduction
- Filed-based universes
- Relational universes
- Programmatic universes
- Relational warehouses
- Multi-dimensional warehouses
- **Universes on SAP BW**
- Conclusion



SAP BW

- Many customers want universes to access SAP BW
- SAP BW data is stored in a relational star schema but revealed to the universe as a Infocube.
- Universes can access this cube by using BEx queries or revealing the cube directly
- Recent software improvements accelerate the performance of those resultant queries
  - Query pruning
- Accessing the relational structure underneath can accelerate performance at a cost …
  - Data Federator views
  - Future semantic layer improvements

SAP BW – Universe Architecture



SAP BW – MDX vs. SQL

BW Server Feature	MDX	SQL
BW Hierarchies	■	
Restricted and Calculated Key Figures	■	
BEx Queries	■	
BW Variables	■	
Currency and Unit Conversion	■	
Exceptions, Conditions	■	
Security	■	■
AVG, COUNT, SUM, MIN, MAX Aggregations	■	■
Navigational Attributes	■	■
Mass Data Enabled	□	■
Ad-hoc Reporting		■
Federation (e.g. BW – RDBMS)		■
Non Cumulative Key Figures	■	

SAP BW – Evaluating Universe Access Methods

- OLAP universes support more SAP BW features
  - … but there is a performance cost (speed)
- Data Federator allows much faster retrieval
  - … with fewer SAP BW features supported
  - This is possible by accessing relational star behind the cube

Agenda

- Introduction
- Filed-based universes
- Relational universes
- Programmatic universes
- Relational warehouses
- Multi-dimensional warehouses
- Universes on SAP BW
- Conclusion

Conclusion

- Universes are a great way of making data accessible to the masses
- Taking into account the source and layout of that data matters
- This presentation has introduced several topology-specific techniques
- Using these techniques at home can accelerate current universe solutions

Questions ?

Alan Mayer

alan.mayer@solidgrounded.com
214-295-6250 (office)
214-755-5771 (mobile)
214-206-9003 (fax)

SolidGround
Technologies

Thank you for participating.

Please remember to complete and return your evaluation form following this session.

For ongoing education on this area of focus, visit the Year-Round Community page at www.asug.com/yrc

SESSION 802