



# ZEN and the ART of Universe Design

Dave Rathbun  
Integra Solutions





# Who Is Integra Solutions?

- **Founded in 1993**
- **Corporate headquarters in Dallas, TX**
- **Offices in Missouri and Connecticut**
- **Services offered include**
  - **Consulting**
  - **Training**
  - **Software sales**



# Who Is Dave Rathbun?

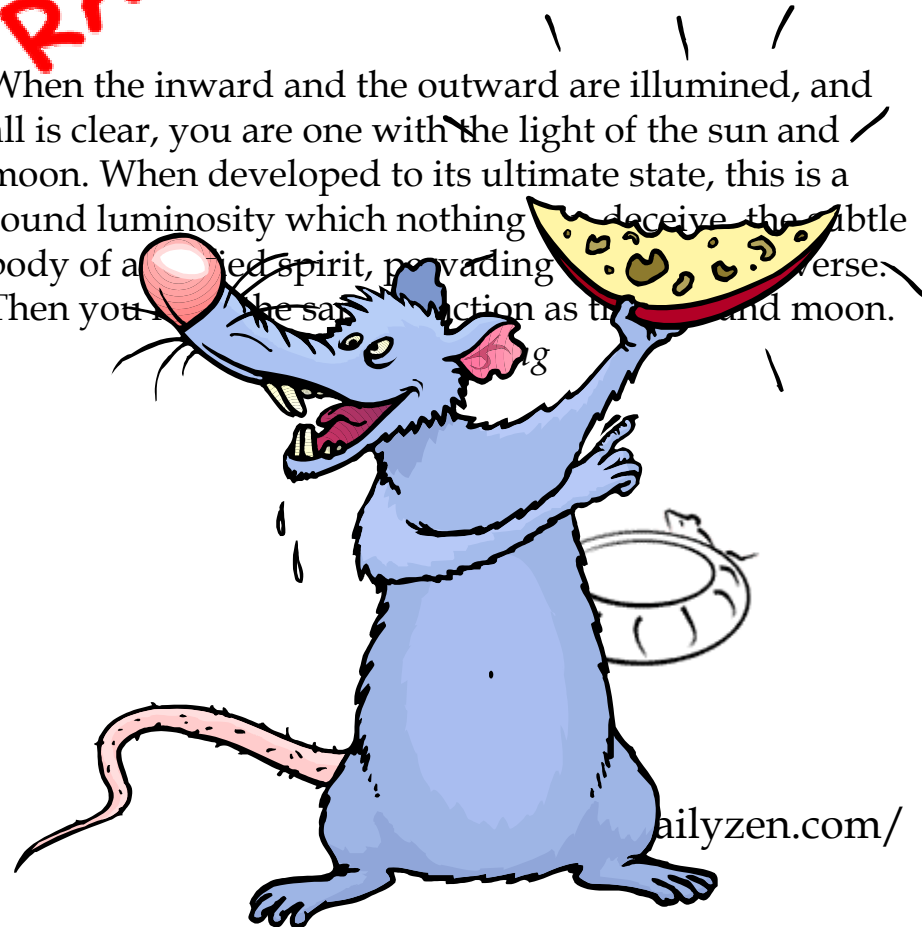
- **Providing BusinessObjects solutions since 1995**
  - Consulting / mentoring / trouble shooting
  - Certified BusinessObjects trainer
  - Develop and deliver customer specific training programs
- **Selected to present at 1996 - 2003 User Conference**
- **User group presentations**
  - NTBOUG, DENBOUG, BONYMAUG
- **Founding member of BOB**
  - <http://busobj.forumtopics.com>

# ZEN... and the Art of Universe Design

**RAT**



When the inward and the outward are illumined, and all is clear, you are one with the light of the sun and moon. When developed to its ultimate state, this is a round luminosity which nothing can deceive the subtle body of a purified spirit, pervading the universe. Then you make the same function as the sun and moon.



[dailyzen.com/](http://dailyzen.com/)



# A RAT By Any Other Name...

- **R**elationship Identification
- **A**tttribute Definition
- **T**est Process
  
- **Z**ero Defect Coding
- **E**nhance Performance
- **N**arrow Scope



# Identify Relationships

- **Relationships between tables are defined by Joins**
- **Adding joins can create relationship problems**
  - **Fan traps**
  - **Chasm traps**
- **Relationship problems must be solved**



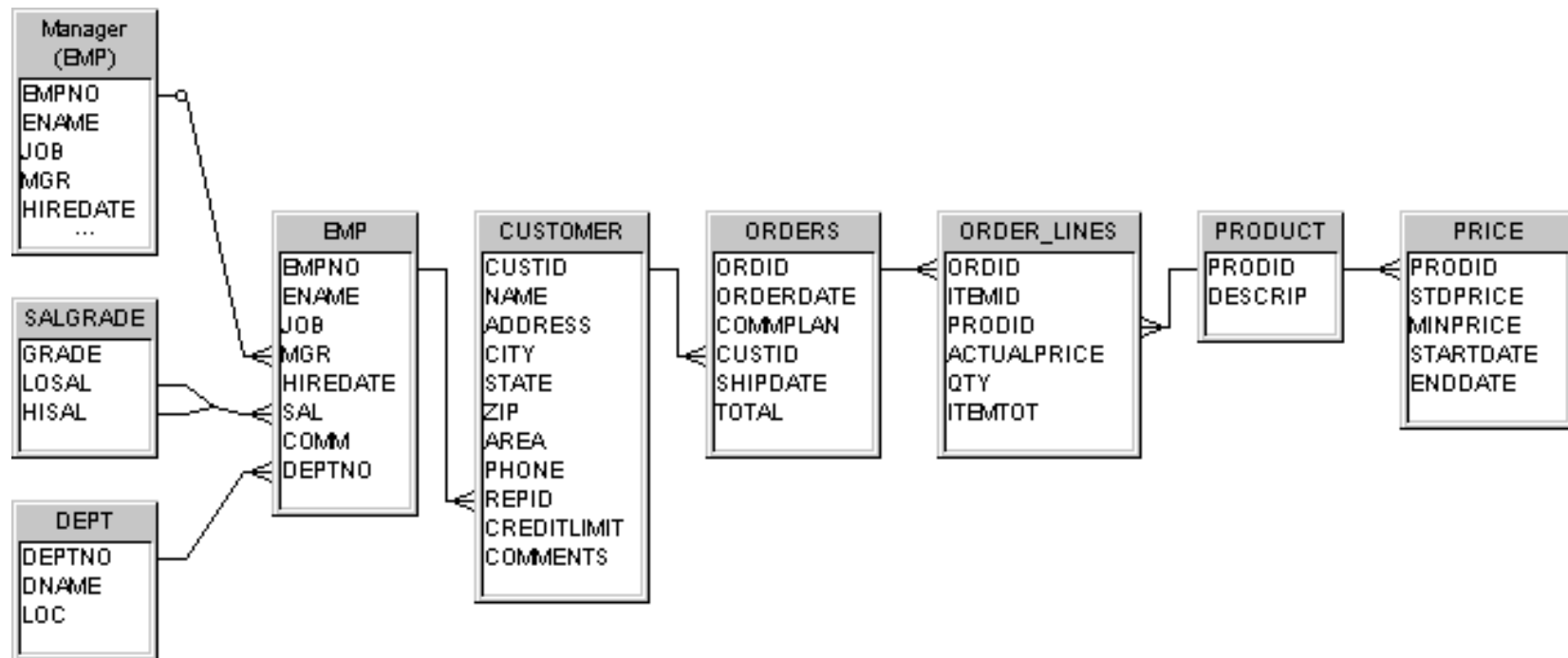
# Cardinalities

- **Cardinalities define the type of relationships**
  - One to Many
  - One to One
  - Many to Many
- **SQL does not require cardinalities to be defined**
- **We use cardinalities to identify relationship problems**



# Summit Sporting Goods Universe

- A total is stored for each order
- Each line item has an extended amount





# Measure Scenario I - No Aggregation

1/3



## *No Aggregation Functions Used*

- Initially none of the measure objects have aggregate functions
- What is the impact on reports?



# Measure Scenario I - No Aggregation

2/3

-  Create a query showing Customer, Quantity, and Line Amount
-  Make a note of:
  - Rows retrieved
  - Total quantity
  - Total line amount
  - Rows displayed

Name		Qty	Line Amount
JUNTA		2,896.00	\$28,643.20
RTS		6,512.00	\$21,123.60
IS		328.00	\$3,056.00
RTS		10,200.00	\$185,480.00
DODS		2,600.00	\$25,600.00
SHAPE UP		3,124.00	\$36,097.60
TKB SPORT SHC		20.00	\$405.60
VOLLYRITE		10,208.00	\$111,102.00
WOMENS SPOR		616.00	\$2,840.00
Sum:		36,504.00	\$414,348.00

See ZEN Demo 01.rep

# Measure Scenario I - No Aggregation

3/3

*Consider the Following...*

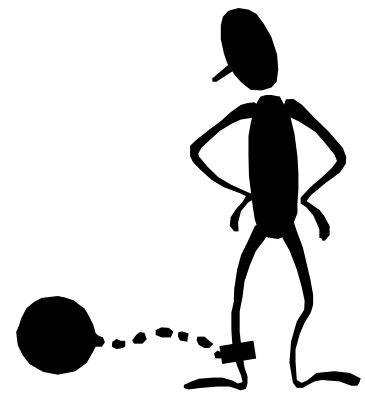
- Without aggregation detailed rows are required
- The details are **projected** on the client
- Disadvantages of a higher row count
  - Longer download times
  - More network traffic
  - Increased computation times
- **ZEN calls for efficient queries**
  - Use Aggregate Functions on Measures
  - Clue: If it won't aggregate then it's not a measure



# Understanding SQL

## *Important Concepts*

- **Sum() requires rows to be grouped**
  - Defines the scope for the sum
  - Provides “semantically dynamic” measure objects
- **Any item not included in an aggregate function must be in the GROUP BY clause**
- **This is a SQL rule!**



# Measure Scenario II – Mixed Settings

1/5




- **Performance is an excellent reason to consider aggregated measures**
- **What about a mixed environment?**
  - Some measures are aggregated
  - Some others are not



# Measure Scenario II – Mixed Settings

2/5

## *Mixed Aggregate and Non-aggregate Measures*

-  In Designer add a sum() to Line Amount
-  Save the universe and run the query again
-  Make a note of the following
  - Rows retrieved
  - Total quantity
  - Total line amount
  - Rows displayed
- **What happened? Why?**

Name	Qty	Line Amount
NTA	323.00	\$28,643.20
S	1,616.00	\$21,123.60
	62.00	\$3,056.00
S	1,150.00	\$185,480.00
DS	450.00	\$25,600.00
SHAPE UP	671.00	\$36,097.60
TKB SPORT SHC	4.00	\$405.60
VOLLYRITE	1,552.00	\$111,102.00
WOMENS SPOR	154.00	\$2,840.00
Sum:	5,982.00	\$414,348.00

# Rows

44

256

See ZEN Demo 02.rep

## Measure Scenario II – Mixed Settings

3/5

*Consider The Following...*

- Remember the rules of the GROUP BY clause
- Any object that is not grouped with a function then it is grouped by the GROUP BY

```
SELECT CUSTOMER.NAME, ORDER_LINES.QTY,  
sum(ORDER_LINES.ITEMTOT)  
FROM ...
```

**GROUP BY**

CUSTOMER.NAME,  
**ORDER\_LINES.QTY**



## Measure Scenario II – Mixed Settings

4/5

- Examine the intermediate data
- TKB Sport Shop has several orders
- How many “unique” qty values appear?
- The correct total is 20
- The distinct or “grouped” total is 4

Name	Qty	Line Amount
TKB SPORT SHOP	1	\$35.00
TKB SPORT SHOP	3	\$8.40
TKB SPORT SHOP	1	\$58.00
TKB SPORT SHOP	1	\$35.00
TKB SPORT SHOP	3	\$8.40
TKB SPORT SHOP	1	\$58.00
TKB SPORT SHOP	1	\$35.00
TKB SPORT SHOP	3	\$8.40
TKB SPORT SHOP	1	\$58.00
TKB SPORT SHOP	1	\$35.00
TKB SPORT SHOP	3	\$8.40
TKB SPORT SHOP	1	\$58.00
Sum:	20	\$405.60

## Measure Scenario II – Mixed Settings

5/5

*Pick a Strategy and Stick With It*

- **Without aggregation correct results are returned**
- **But at the expense of:**
  - **Client efficiency**
  - **Network bandwidth**
- **When some measures are aggregated and others are not, incorrect results can occur**
- **The non-aggregated measures will be low because of the GROUP BY clause**
- **What if every measure has an aggregation function, does that work?**

# Measure Scenario III – Fan Traps

1/6

## *Summary Measure Objects*

- If measures are aggregated consistently the results are accurate
- What happens when measures are taken from summary and detail tables at the same time?

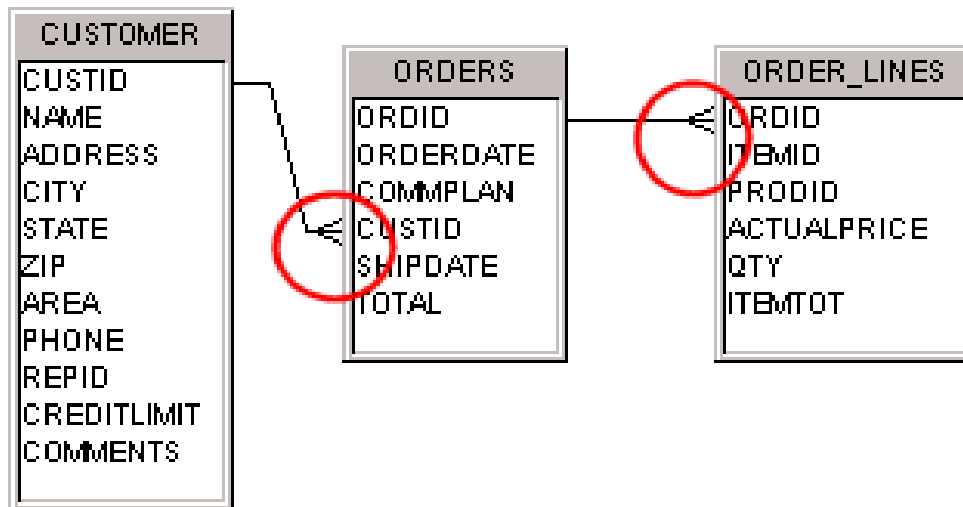


# Measure Scenario III – Fan Traps

2/6

## *Solving Cardinality Issues*

- The relationship between CUSTOMER, ORDERS and ORDER\_LINES is called a Fan Trap
- The rows “fan out” in a one - many - many relationship



# Measure Scenario III – Fan Traps

3/6

- **BusinessObjects resolves this transparently by using two queries**



```
SELECT
  datepart('yyyy', ORDERS.ORDERDATE),
  sum(ORDERS.TOTAL)
FROM
  ORDERS
GROUP BY
  datepart('yyyy', ORDERS.ORDERDATE)
```



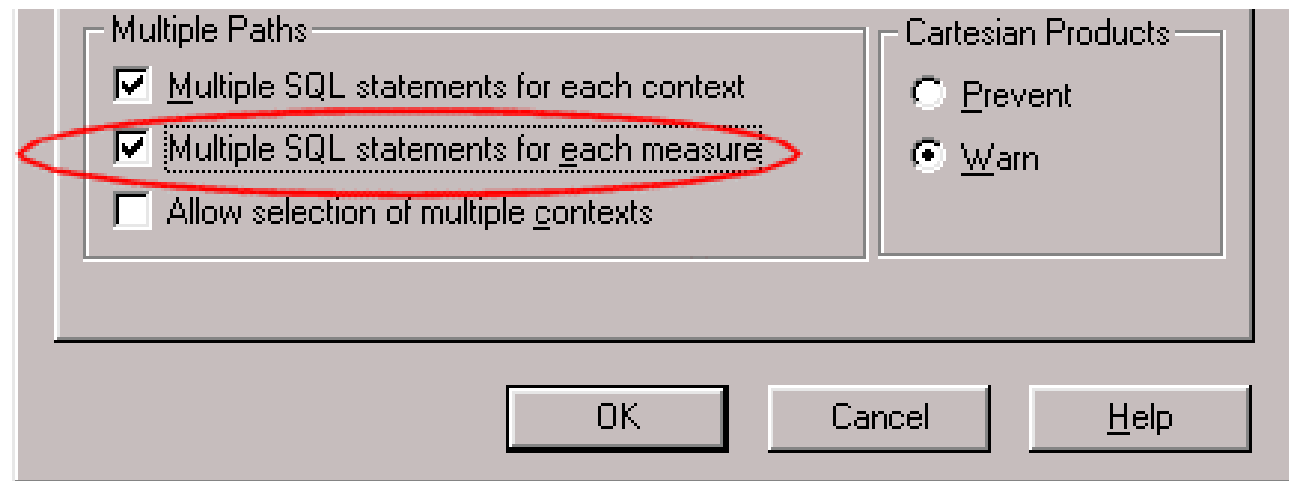
```
SELECT
  datepart('yyyy', ORDERS.ORDERDATE),
  sum(ORDER_LINES.QTY)
FROM
  ORDER_LINES,
  ORDERS
WHERE
  ( ORDERS.ORDID=ORDER_LINES.ORDID )
GROUP BY
  datepart('yyyy', ORDERS.ORDERDATE)
```

# Measure Scenario III – Fan Traps

4/6

## *Fan Trap Protection*

- **Fan Traps can be resolved automatically in some cases**
- **When all measures have aggregation functions BusinessObjects will split the query in two pieces**



# Measure Scenario III – Fan Traps

5/6

## *Multiple SQL Statements For Each Measure*

- The “fan trap protection” setting is on by default
- What happens if this setting is changed?
- With the setting on

Order Year	Total Order	Qty
2000	\$37,784.40	3,268
2001	\$376,563.60	33,236
<b>Sum:</b>	<b>\$414,348.00</b>	<b>36,504</b>

- After turning it off

Order Year	Total Order	Qty
2000	\$210,769.20	3,268
2001	\$2,484,962.80	33,236
<b>Sum:</b>	<b>\$2,695,732.00</b>	<b>36,504</b>

See ZEN Demo 03.rep



## Measure Scenario III – Fan Traps

6/6

*Consider the Following...*

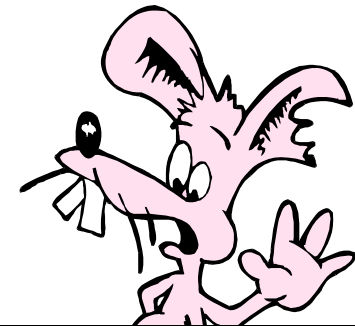
- **This special setting will only work with aggregated measures!**
- **Any measure object that does not have an aggregate function will not trigger this protection setting**
- **Before you change this setting be sure to understand its impact on query generation**




# Measure Scenario IV – Dimension Fans 1/5

## *Dimension Fan Traps*

- **Consider this example**
  - Order Line Number, Total Order, Quantity
- **Even with the Fan Trap Protection setting on incorrect results were returned**



Order Line Num	Total Order	Quantity
1	\$413,452.00	3,924.00
2	\$413,898.40	12,944.00
3	\$409,941.60	9,916.00
4	\$275,912.00	3,640.00
5	\$221,832.00	2,440.00
6	\$218,776.00	440.00
7	\$185,480.00	800.00
8	\$185,480.00	400.00
9	\$185,480.00	800.00
10	\$185,480.00	1,200.00
		<b>\$2,695,732.00</b>
		<b>36,504.00</b>

See ZEN Demo 04.rep

## Measure Scenario IV – Dimension Fans 2/5

- With a dimension fan trap it does not help to separate the SQL into two queries
- The summary line amount is multiplied by the number of rows in the detailed table
- The extra table is included because of the dimension



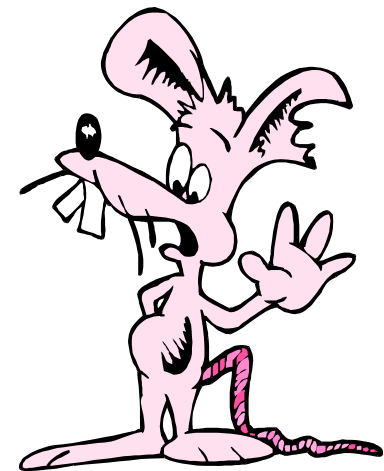
```
SELECT
ORDER_LINES.ITEMID,
sum(ORDERS.TOTAL)
FROM
ORDER_LINES,
ORDERS
WHERE
( ORDERS.ORDID=ORDER_LINES.ORDID )
GROUP BY
ORDER_LINES.ITEMID
```

## Measure Scenario IV – Dimension Fans 3/5

### *The Problem Gets Worse*

- Here is a query using Total Order, Quantity, and Product

Product Description	Total Order	Quantity
ACE TENNIS BALLS-3 PACK	\$338,413.60	9,152.00
ACE TENNIS BALLS-6 PACK	\$338,323.20	7,212.00
ACE TENNIS NET	\$225,419.60	2,048.00
ACE TENNIS RACKET I	\$340,615.60	2,568.00
ACE TENNIS RACKET II	\$268,436.00	1,192.00
RH: "GUIDE TO TENNIS"	\$244,685.60	2,884.00
SB ENERGY BAR-6 PACK	\$236,985.60	6,044.00
SB VITA SNACK-6 PACK	\$216,120.00	2,200.00
SP JUNIOR RACKET	\$242,216.00	1,800.00
SP TENNIS RACKET	\$244,516.80	1,404.00
Sum:	<b>\$2,695,732.00</b>	<b>36,504.00</b>

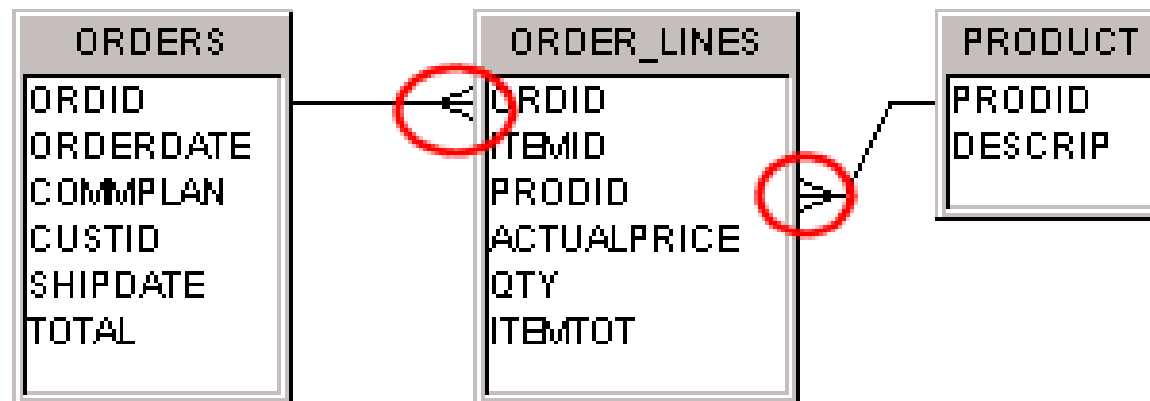


See ZEN Demo 05.rep

# Measure Scenario IV – Dimension Fans 4/5

## *One - Many - One*

- A fan trap is one - many - many
- The relationship from Products to Orders is one - many - one instead
- Multiple SQL statements will not solve this problem



## Measure Scenario IV – Dimension Fans 5/5

### *Pick Your Source*

- In the data model for Summit Sporting Goods the Total Order can be obtained from two places
  - Total Order from the ORDERS table
  - Quantity \* Price from the ORDER\_LINES table
- To solve our dimension fan trap we need to carefully consider where the Total Order value comes from
- Your mission: Make the selection process completely transparent to the user
  - No training issues
  - No wrong answers
  - ZEN !

# Introducing Aggregate Awareness

1/8

## *The @Aggregate\_Aware() Function*

- **There are two places to obtain the total order value**
  - **Sum(ORDERS.TOTAL)**
  - **Sum(ORDER\_LINES.QTY \* ACTUALPRICE)**
- **The first choice is faster because there are fewer rows to process**
- **The second choice avoids the dimension fan trap**
- **The @Aggregate\_Aware() function will solve this problem**



# Introducing Aggregate Awareness

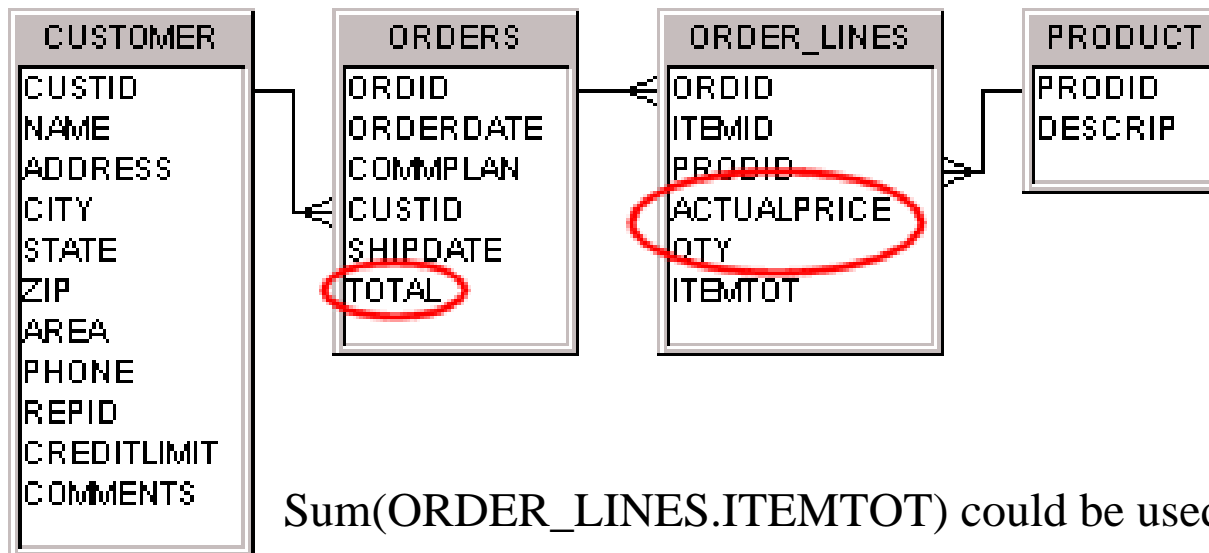
2/8

- **@Aggregate\_Aware()** works by allowing a designer to set up a hierarchy of choices
  - **@Aggregate\_Aware( best choice,  
next choice,  
...  
last choice)**
- **The first step is to define the choices in the proper order**

# Introducing Aggregate Awareness

3/8

- The SQL for Total Order becomes
  - @Aggregate\_Aware(  
Sum(ORDERS.TOTAL),  
Sum(ORDER\_LINES.QTY \* ACTUALPRICE) )



Sum(ORDER\_LINES.ITEMTOT) could be used for the second option

# Introducing Aggregate Awareness

4/8

## *Make the Choice*

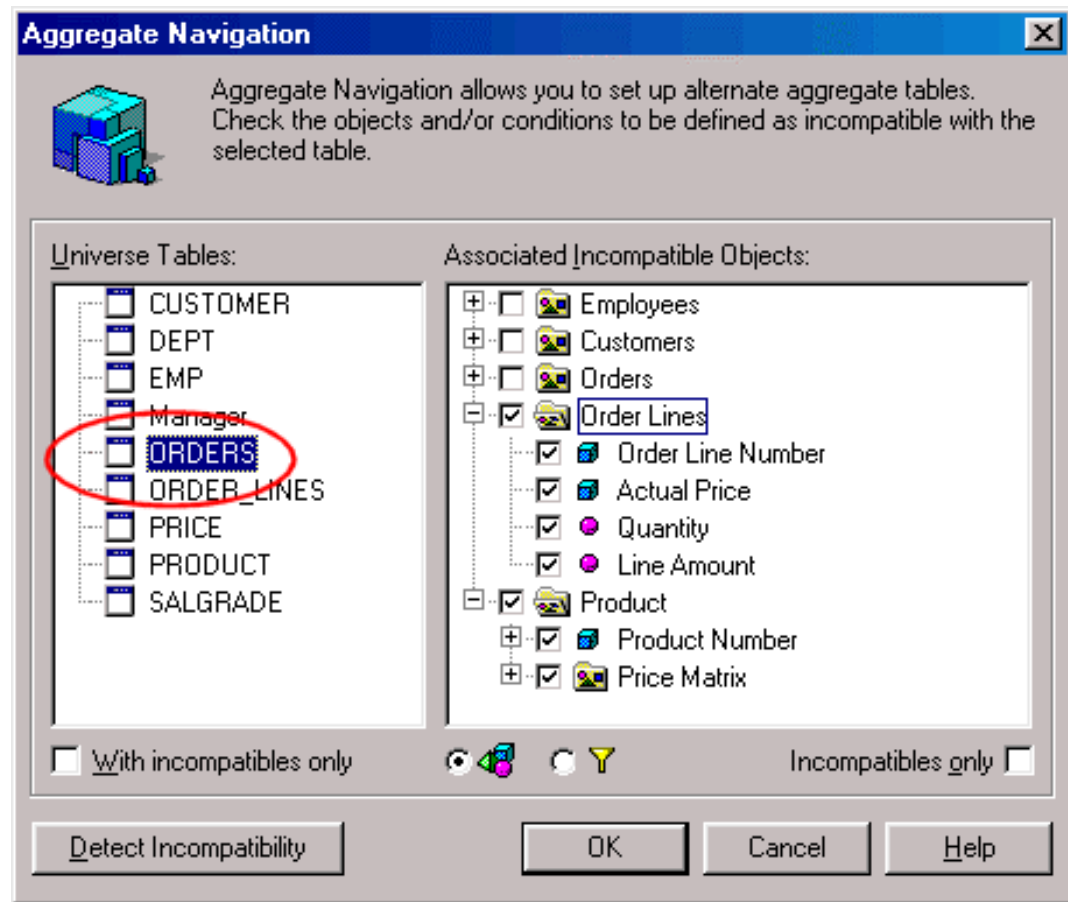
- The next step is to define when each choice may be used
- This process is called **Aggregate Navigation**
- There are some simple rules
  - Any object from a lower level table is incompatible
  - Any object from a summary table is compatible
  - Incompatibility is defined between objects and tables
  - Tables cannot be incompatible with other tables

# Introducing Aggregate Awareness

5/8

## *Defining Incompatibilities*

- Our summary table is **ORDERS**
- Anything from **ORDER\_LINES** or beyond should force the switch to the detail level table




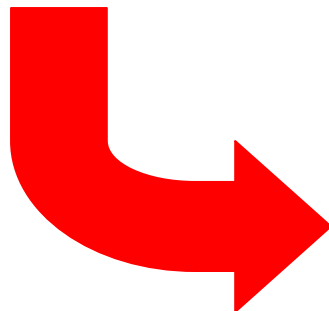
# Introducing Aggregate Awareness

6/8

## *Seeing The Results*

- **Once this has been configured, review the generated SQL**

 Product Number	 Product Description
 Total Order	 Quantity



```
SELECT
  PRODUCT.PRODID,
  PRODUCT.DESCRIP,
  sum(ORDER_LINES.QTY * ORDER_LINES.ACTUALPRICE),
  sum(ORDER_LINES.QTY)
FROM
  ORDER_LINES,
  PRODUCT
WHERE
  ( ORDER_LINES.PRODID=PRODUCT.PRODID )
GROUP BY
  PRODUCT.PRODID,
  PRODUCT.DESCRIP
```

# Introducing Aggregate Awareness

7/8

## *Seeing The Results*

- **And the resulting output with the correct total**

Product Description	Total Order	Quantity
ACE TENNIS BALLS-3 PACK	\$27,307.60	9,152.00
ACE TENNIS BALLS-6 PACK	\$40,227.20	7,212.00
ACE TENNIS NET	\$118,752.00	2,048.00
ACE TENNIS RACKET I	\$88,576.00	2,568.00
ACE TENNIS RACKET II	\$53,238.00	1,192.00
RH: "GUIDE TO TENNIS"	\$9,805.60	2,884.00
SB ENERGY BAR-6 PACK	\$14,345.60	6,044.00
SB VITA SNACK-6 PACK	\$8,800.00	2,200.00
SP JUNIOR RACKET	\$19,600.00	1,800.00
SP TENNIS RACKET	\$33,696.00	1,404.00
<b>Sum:</b>	<b>\$414,348.00</b>	<b>36,504.00</b>

See ZEN Demo 06.rep

# Introducing Aggregate Awareness

8/8

## *Another Test*

- **Even the first example of a dimension fan trap is fixed with this solution**

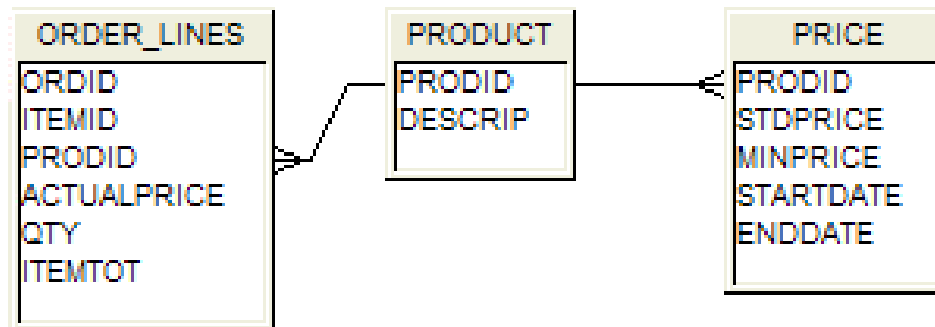
Order Line Num	Total Order	Quantity
1	\$113,461.60	3,924.00
2	\$80,982.40	12,944.00
3	\$51,176.00	9,916.00
4	\$21,216.00	3,640.00
5	\$119,696.00	2,440.00
6	\$9,736.00	440.00
7	\$10,000.00	800.00
8	\$1,360.00	400.00
9	\$1,920.00	800.00
10	\$4,800.00	1,200.00
	<b>\$414,348.00</b>	<b>36,504.00</b>

See ZEN Demo 06.rep



- **Cardinality can also be used to identify a chasm trap**
- **Chasm (ka - zum)**
  - A deep, steep-sided opening in the earth's surface; an abyss or gorge.
  - A sudden interruption of continuity; a gap.
  - A pronounced difference of opinion, interests, or loyalty.
- **The 2<sup>nd</sup> definition fits for database relationships**

- **Examine the following relationships**
  - Items may be ordered more than once from the product list
  - Each ordered item is assigned a price that is valid at that time
  - Each product has a product price history



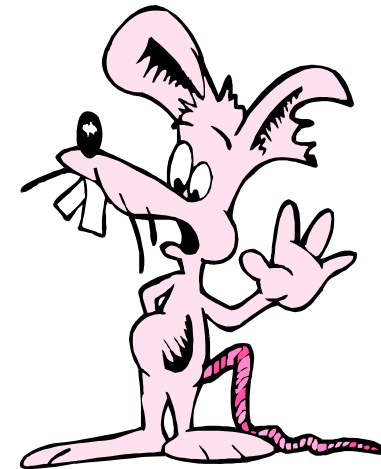
- **What is the relationship between Price and Order Lines?**

# Chasm Trap

3/4

-  Query Product and Order Line data together

Product Description	Standard Price	Actual Price	Quantity	Line Amount
ACE TENNIS BALLS-3 PACK	\$2	\$3	20	\$50
ACE TENNIS BALLS-3 PACK	\$2	\$3	9,092	\$25,458
ACE TENNIS BALLS-3 PACK	\$2	\$45	40	\$1,800
ACE TENNIS BALLS-3 PACK	\$3	\$3	20	\$50
ACE TENNIS BALLS-3 PACK	\$3	\$3	9,092	\$25,458
ACE TENNIS BALLS-3 PACK	\$3	\$45	40	\$1,800
ACE TENNIS BALLS-6 PACK	\$5	\$5	200	\$1,000
ACE TENNIS BALLS-6 PACK	\$5	\$6	400	\$2,200
ACE TENNIS BALLS-6 PACK	\$5	\$6	6,612	\$37,027
ACE TENNIS BALLS-6 PACK	\$6	\$5	200	\$1,000
ACE TENNIS BALLS-6 PACK	\$6	\$6	400	\$2,200
ACE TENNIS BALLS-6 PACK	\$6	\$6	6,612	\$37,027
ACE TENNIS NET	\$54	\$50	4	\$200
ACE TENNIS NET	\$54	\$50	4	\$200
SB ENERGY BAR-6 PACK	\$2	\$2	800	\$1,760
SB ENERGY BAR-6 PACK	\$2	\$2	5,244	\$12,586
SB VITA SNACK-6 PACK	\$4	\$4	2,200	\$8,800
SP JUNIOR RACKET	\$13	\$9	400	\$3,600
SP JUNIOR RACKET	\$13	\$10	600	\$6,000
SP JUNIOR RACKET	\$13	\$13	800	\$10,000
SP TENNIS RACKET	\$24	\$24	1,404	\$33,696
		Sum:	62,436	\$884,263

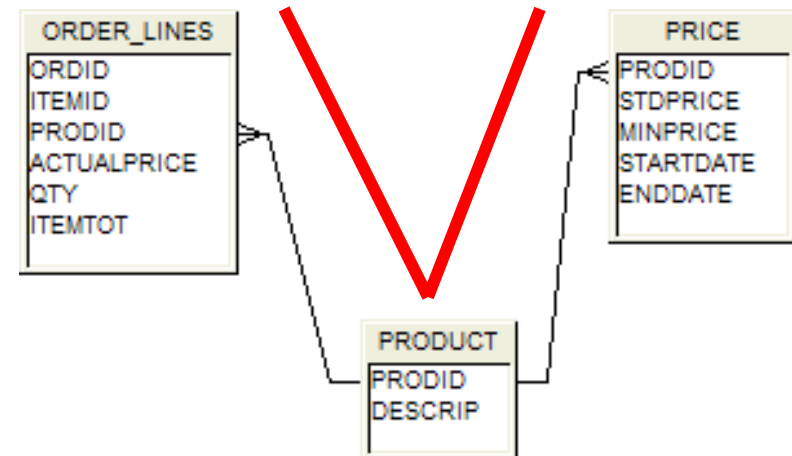
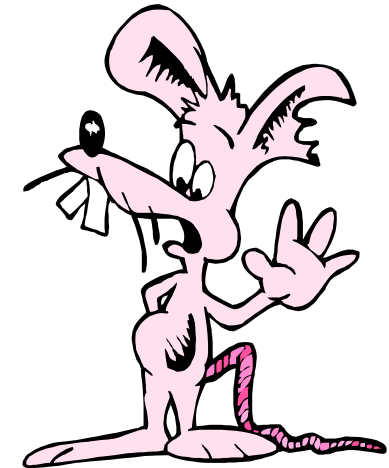


See ZEN Demo 07.rep

# Chasm Trap

4/4

- There is no valid relationship!
- Imagine that cardinalities go up or down a mountain
  - One – Many goes up
  - Many – One goes down
- Product is in a chasm
- You cannot cross this chasm



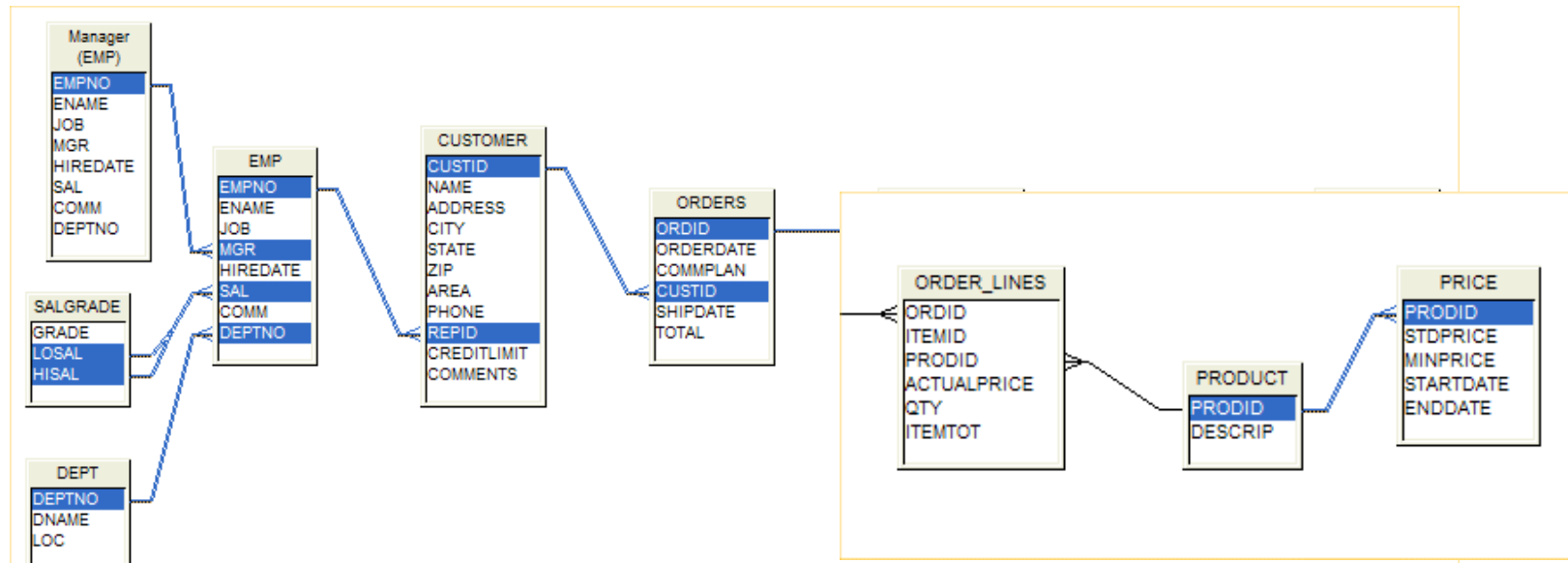
- **None of the “Measure Tricks” will fix this problem**
- **There is no valid relationship yet these items are in the same universe**
- **Context = Meaning**
  - **A context is a sub-set of joins in a universe**
  - **That sub-set of joins has a particular meaning**

# Introducing Contexts

2/3

## *Contexts Identify Joins With Purpose*

- Each context is a set of joins with a specific meaning
- A query cannot cross from one context to another
- This will solve the chasm trap



- **Summit Sporting Goods now has two contexts**
  - **Actual Sales**
  - **Price Matrix**
- **These two concepts are not related to each other**
  - **Each order is a price at a point in time**
  - **Price Matrix shows prices at all times**
- **By creating contexts these two areas with different meanings are kept in separate queries**
- **A Designer has the option to prevent two (or more) contexts from being used at the same time**



# Measure Objects – Wrap Up

## *Understanding Measure Objects*

- **The correct implementation of measure objects is crucial to the success of a BusinessObjects project**
- **Measures are designed to aggregate**
  - Without aggregate functions too much data can be returned
  - A mixture of aggregate and non-aggregate objects is very dangerous and should probably never be used
  - Adding aggregate functions can cause new problems
  - Understand the Fan Trap Protection setting
  - Implement @Aggregate\_Aware() if you can
  - Contexts may also be required





# A RAT By Any Other Name...

- **R**elationship Identification (Fan or Chasm Traps)
- **A**tttribute Definition
- **T**est Process
  
- **Z**ero Defect Coding (Proper Measure Definition)
- **E**nhance Performance (@Aggregate\_Aware)
- **N**arrow Scope (Contexts)



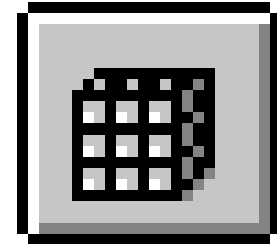
# What About Object Attributes? (RAT)

- Relationships (joins) provide structure to the universe
- Objects types are more visible to the report writer
- Do the attributes of an object really matter?
- What is a cube anyway?
- Why have two non-measure options?
  - Dimensions
  - Details

# Cube Theory

*What is a Cube?*

- **Results of a query**
  - Universe
  - Personal data file
- **Defined by dimension objects**
- **Detail objects offer supplemental data**
- **Measures are stored at lowest level**
  - Typically aggregated on the server
  - Additional projection on the client



# Cube Structure

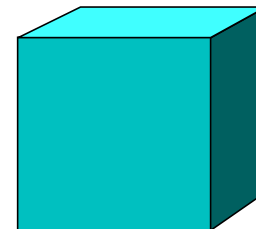
1/4

## *One Dimensional Cube*

- **A single dimensional cube is not drillable**
- **Measures are project to the highest level**

Resort	Revenue
Bahamas Beach	971,444.00
French Riviera	835,420.00
Hawaiian Club	1,479,660.00

- **This cube is “defined” by Resort**



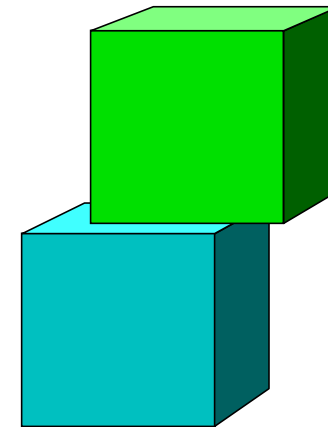
# Cube Structure

2/4

## *Two Dimensional Cube*

- **Measures stored only at lowest level**
- **Cube is “defined” by Resort and Year combinations**
  - **Measures are stored at the lowest level and projected**

Resort	Year	Revenue
Bahamas Beach	FY93	287,929.00
Bahamas Beach	FY94	307,400.00
Bahamas Beach	FY95	376,115.00
French Riviera	FY93	295,940.00
French Riviera	FY94	280,310.00
French Riviera	FY95	259,170.00
Hawaiian Club	FY93	479,685.00
Hawaiian Club	FY94	519,530.00
Hawaiian Club	FY95	480,445.00



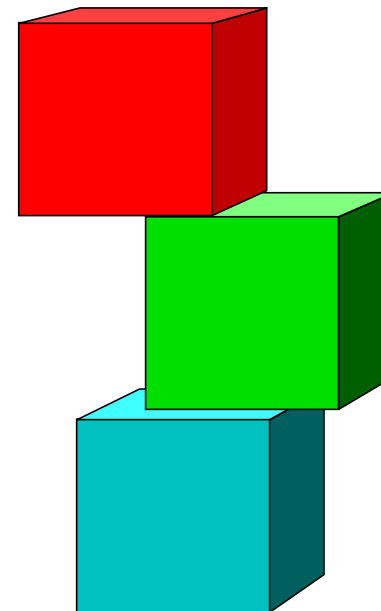
# Cube Structure

3/4

## *Three Dimensional Cube*

- Resort, Year, and Region**

Resort	Year	Region	Revenue
Bahamas Beach	FY93	Bavaria	61,424.00
Bahamas Beach	FY93	East Coast	19,124.00
Bahamas Beach	FY93	East Germany	30,032.00
Bahamas Beach	FY93	East Japan	76,724.00
Bahamas Beach	FY93	Mid West	32,112.00
Bahamas Beach	FY93	Ruhr	10,976.00
Bahamas Beach	FY93	South	23,445.00
Bahamas Beach	FY93	West	17,520.00
Bahamas Beach	FY93	West Japan	16,572.00
Bahamas Beach	FY94	Bavaria	84,944.00
Bahamas Beach	FY94	East Germany	30,032.00

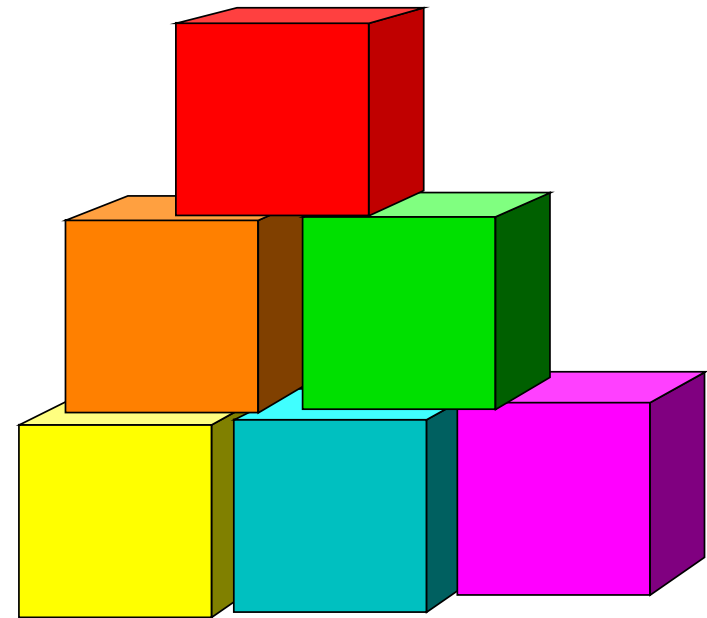




## 4/4

- **There is no limit to the number of dimensions**
- **Mathematicians define any four+ dimensional structure as a Hypercube**

- **Note: BusinessObjects does not require a “top-down” dimensional hierarchy**

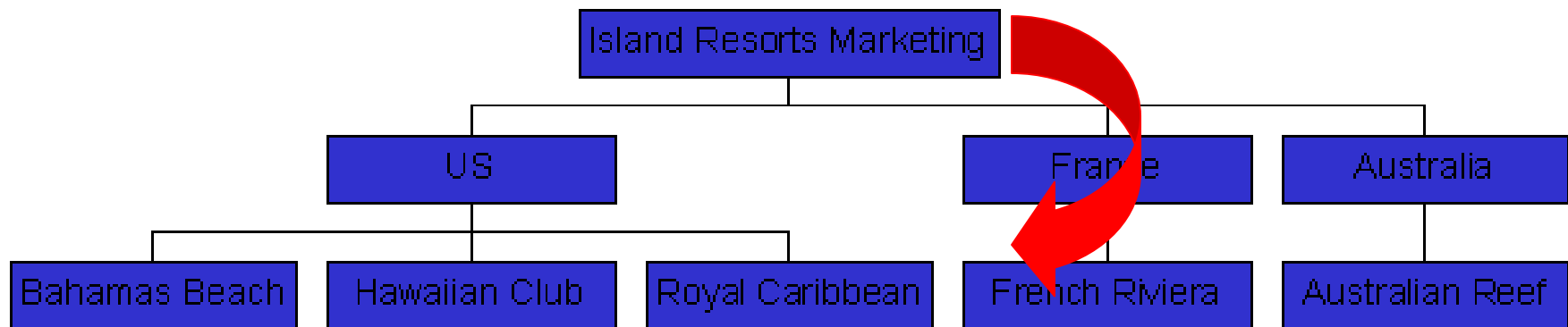


# Dimensional Hierarchies

1/3

*One - Many Relationships Provide a Top-Down Dimension Path*

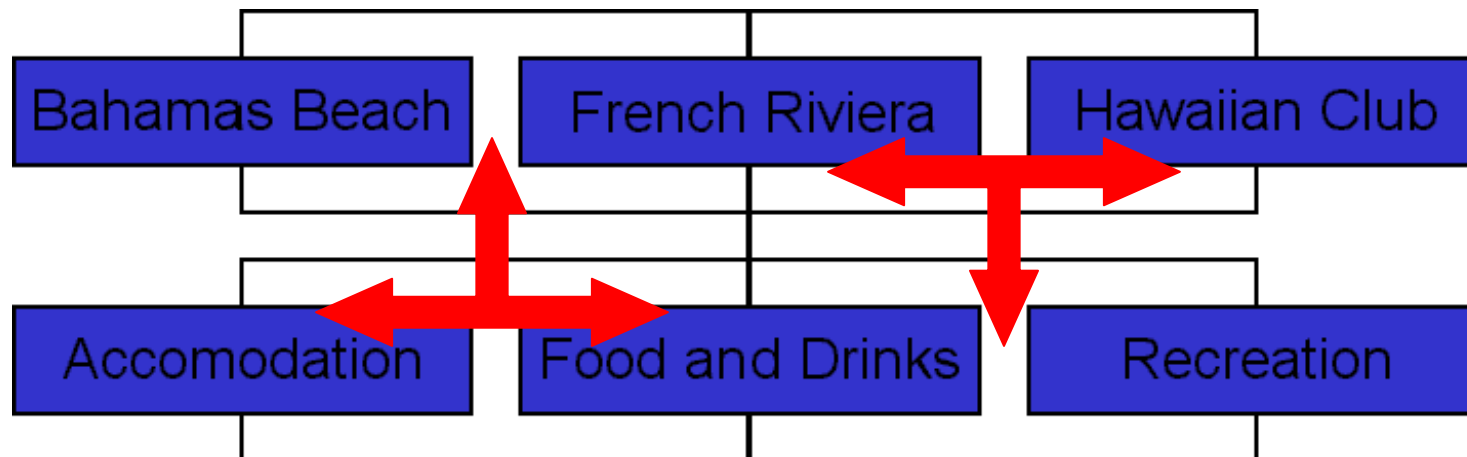
- **These objects progress from top (Country) down (Resort)**



- **What if we add in Service Line?**

## *Many - Many Relationships Are Not Top-Down*

- **Many - Many**
  - Each resort has many service lines
  - Each service line is offered at many resorts



# Dimensional Hierarchies

3/3

## *Dimensions Define the Cube*

- **The combination of dimensions defines the projection of the measures**

Service Line	Resort	Revenue
Accommodation	Bahamas Beach	673,664.00
Accommodation	French Riviera	563,250.00
Accommodation	Hawaiian Club	981,210.00
Food & Drinks	Bahamas Beach	169,680.00
Food & Drinks	French Riviera	107,400.00
Food & Drinks	Hawaiian Club	277,750.00
Recreation	Bahamas Beach	128,100.00
Recreation	French Riviera	164,770.00
Recreation	Hawaiian Club	220,700.00

Resort	Revenue
Bahamas Beach	971,444.00
French Riviera	835,420.00
Hawaiian Club	1,479,660.00

Service Line	Revenue
Accommodation	2,218,124.00
Food & Drinks	554,830.00
Recreation	513,570.00



# Cube Theory

## *What is Projection?*

- **Projection is the process used to aggregate measure objects on the client**
- **Data is stored at the most detailed level**
- **Measures are projected onto the dimensions used**
- **This affects both cube size and performance**

- **When a cube is created measures are stored only once**
- **As the report is manipulated the measure values are aggregated on the fly**
- **This reduces the size of the cube since only the lowest level of detail is stored**

- **Assume a cube with revenue data for**
  - 3 years with 52 weeks per year
  - 5 resorts with 15 salespeople
  - 200 customers purchasing up to 9 services
- **This cube would require a matrix with  $3 * 52 * 5 * 15 * 200 * 9$  or 21,060,000 cells**
- **BusinessObjects stores only the values returned from the database**





# Cube Theory Synopsis

- **Dimensions define the cube (key values)**
  - No limit to number of dimensions in a cube
  - Stored in tokenized form for compression
- **Measures project**
  - Avoid storing huge matrix of result values
  - Provide for efficient slice and dice or drilling
- **Details**
  - Must be good for something although we haven't talked about them yet...

# Cube Theory

- Enough of the theory, how does it all work?
- Let's look at the implementation of these ideas...

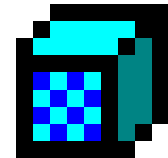


# Object Types – Dimensions

1/2

## *Dimensions are Key Fields*

- Define the scope of the cube
- Provide a logical link across data providers
- Furnish a natural path for drilling
- Often used for query criteria
- Do not aggregate
- Key component of any universe design



# Object Types – Dimensions

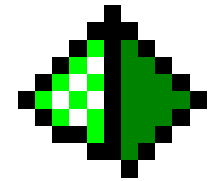
2/2

## *Dimension Objects Come With Baggage*

- **Multiple data providers**
  - Shared dimensions must be linked
  - Unlinked dimensions are not valid in a common block
- **Drilling**
  - Extra dimensions can confuse users
- **Report Generation**
  - #MULTIVALUE errors - dimensions in the wrong place
  - #COMPUTATION errors - dimensions missing



## *Details are Supporting Attributes*



- **Additional information about a dimension**
- **Generally will have a one - one relationship**
  - Often found in the same table as parent dimension
  - May come from lookup tables with many - one relationship (one detail value associated with multiple dimensions)
  - Never in a one - many relationship (one dimension value associated with multiple detail values)
- **Seldom used as query criteria**
- **Do not aggregate**
- **Optional component in any universe design**

# Object Types – Details

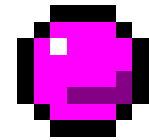
2/2

*Detail Objects are More Flexible... and Limited*

- **Multiple data providers**
  - Detail objects follow their owner dimension
  - Once a dimension is linked associated details are also valid
- **Drilling**
  - Detail objects are not drillable
- **Report Generation**
  - Available for use in a sort, filter, or break
  - Displayed in all block types
  - May not be ranked

## *Measure Objects are Indicators*

- Numerical values
- Often used as query criteria
- Should have a valid aggregation function
- Optional projection function
- Key component of any universe design





*Measure Objects Project Onto the Dimensions in the Cube*

- **Multiple data providers**
  - Any measure from any data provider is valid
  - The values project to the appropriate dimension values
- **Drilling**
  - Measure objects are projected to the level of the hierarchy
  - Measure objects by themselves are not drillable
- **Report Generation**
  - Understanding report context is critical to effective reporting

# The Point of All This...

*Each Object Type has its Own Characteristics*



- **Dimension?**
  - **Detail?**
  - **Measure?**
- 
- **The remaining slides will focus on whether to assign an object attribute of Dimension or Detail**



# Design Scenarios

- **Scenario I - Dimension or Detail, does it matter?**
- **Scenario II - Using multiple data providers**

# Design Scenario I

## *Dimension or Detail: Does It Matter?*

- **How do you determine whether to assign an object type as dimension or detail?**



*Database Attributes are Dimensions or Details... but Which?*

- **Dimensions**

- Are independent “stand alone” attributes of a database entity
- Can be drilled, linked, or used as query conditions
- Generally have associated LOV queries

- **Details**

- Have no distinct purpose as separate objects
- Depend on their dimension value for semantic clarity
- Cannot be drilled or linked
- May be conditions
- Often do not have associated LOV queries

## *Dimension Examples*

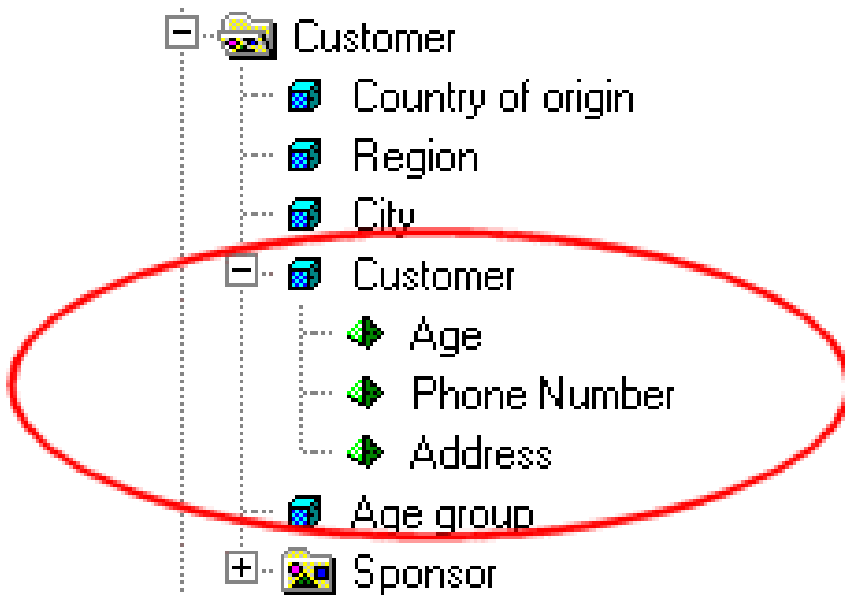
- In theory any non-key attribute can become a detail
- In practice this is not possible
- The following are typically defined as dimensions
  - Year, Quarter, Month, and Week derived from a date values
  - Users would rather drill on names than unique system ID's
- Dimensions are more obvious on the query panel

# Dimension or Detail?

3/4

## *Detail Examples*

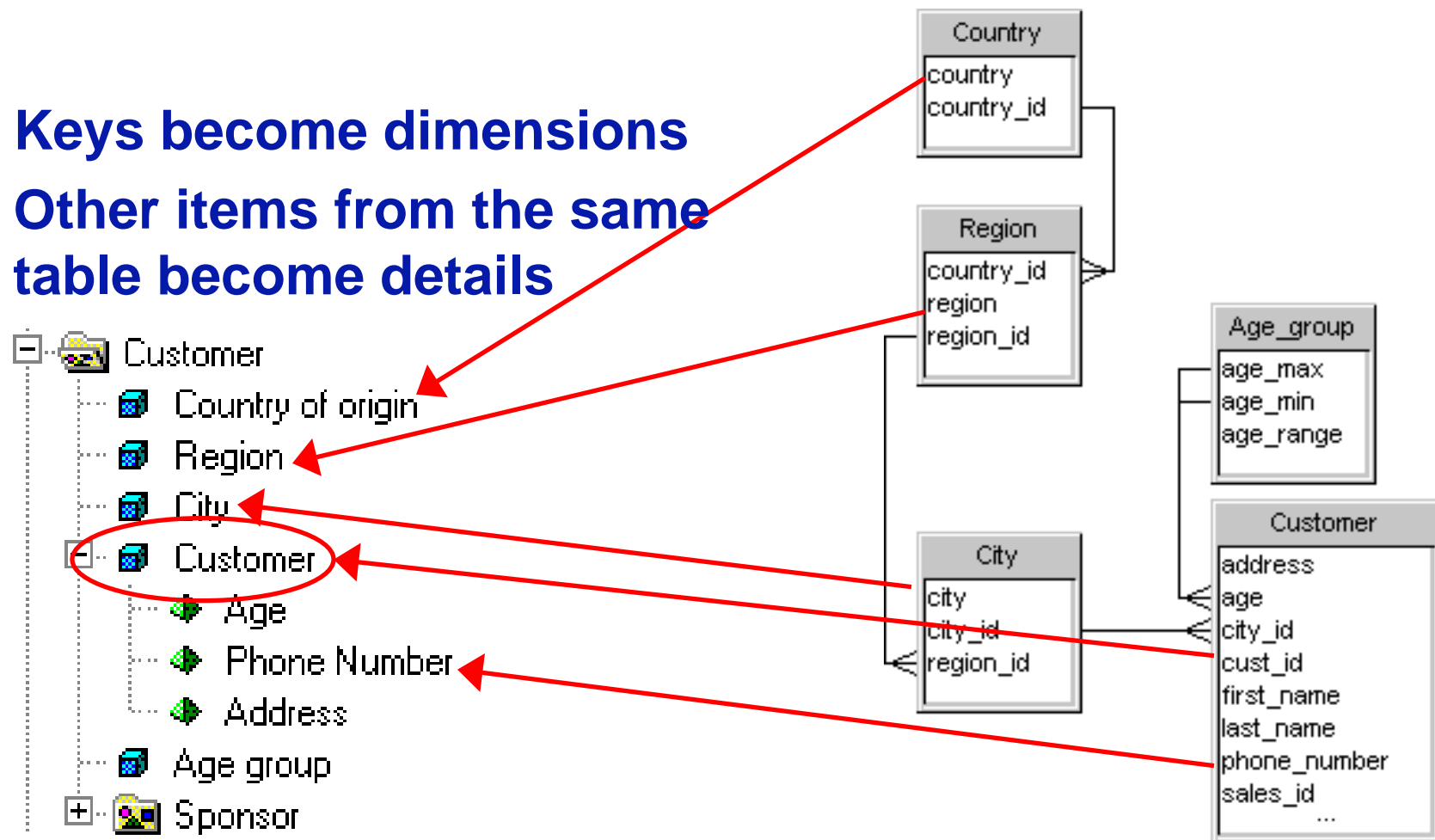
- Any attribute (non-measure) that is not a key, not drillable, and not used to link
- Requires user training on the query panel



# Dimension or Detail?

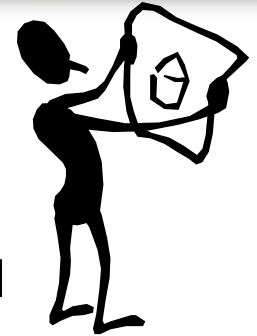
4/4

- Keys become dimensions
- Other items from the same table become details





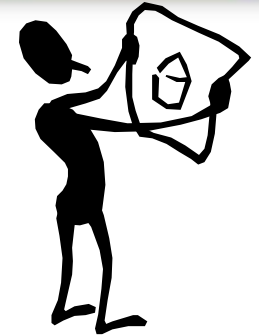
# eFashion Universe Review



- **eFashion**
  - Dimension Week and detail Year/week do not have 1-1
  - Class Store Details contains dimensions that could be details
  - Class Product, object SKU desc should be a detail of SKU Number
  - Color / Color number could be reversed... usability comes into play here
  - There may be other issues to discover
- **eFashion has some problems, but how will they impact reports?**

# eFashion Problem I

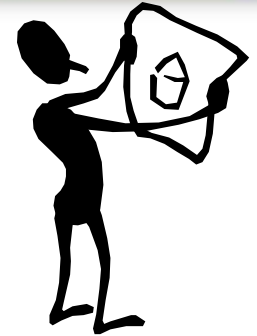
*Year/Week Should be a Key (Dimension)*



-  **Combine eFashion data with spreadsheet**
-  **Details**
  - Create an eFashion query with Week, Year/week, Sales Revenue, and Quantity Sold
  - Use the eFashionForecast.XLS file as a Personal Data source
  - Can the data providers be combined properly? Why or why not?
  - What needs to be adjusted before this will work?

See ZEN eDemo 01.rep

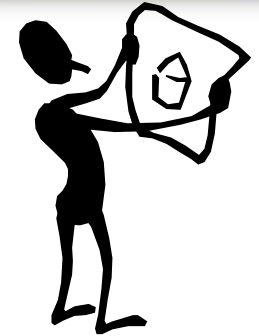
# eFashion Problem I Review





- Since the eFashion Year/Week object is incorrectly classified as a detail, it cannot be used as a link
- However, it is the “key” to identifying a unique weekly value
- Consider the following...
  - If the Year/Week object were a dimension, could the Week object then be a detail?

# eFashion Problem II

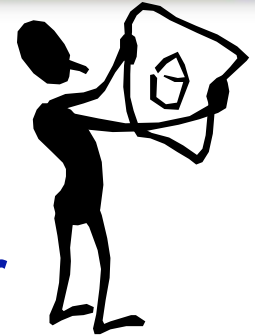
*SKU Description Should be a Detail*



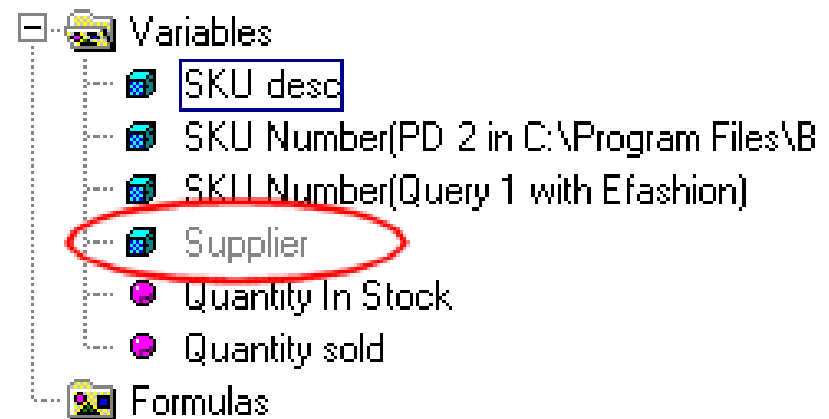
-  **Combining eFashion data with spreadsheet**
-  **Details**
  - Create an eFashion query with SKU Number, SKU Desc, and Quantity Sold
  - Use the eFashionSupplierInventory.XLS file as a Personal Data source
  - Can the data providers be combined properly? Why or why not?
  - What needs to be adjusted before this will work?

See ZEN eDemo 02.rep

# eFashion Problem II Review



- Since the SKU Desc is improperly defined as a dimension, it cannot be combined with Supplier
- Supplier is definitely a dimension
- In this case, a report variable can be created to solve the problem
- Careful consideration of object types at design time would have eliminated this problem



# Design Scenario I

## *Dimension or Detail: Does It Matter?*

- **Yes!**
- **It is worth the time to consider object types carefully when developing a universe**
- **Do not assume that attributes are automatically dimensions**
- **Do not assume that numeric values are always measures**





# Design Scenarios

- **Scenario I - Dimension or Detail, does it matter?**
- **Scenario II - Using multiple data providers**

# Design Scenario II

## *Using Multiple Data Providers*

- **What steps need to be taken to facilitate using multiple data providers in a document?**







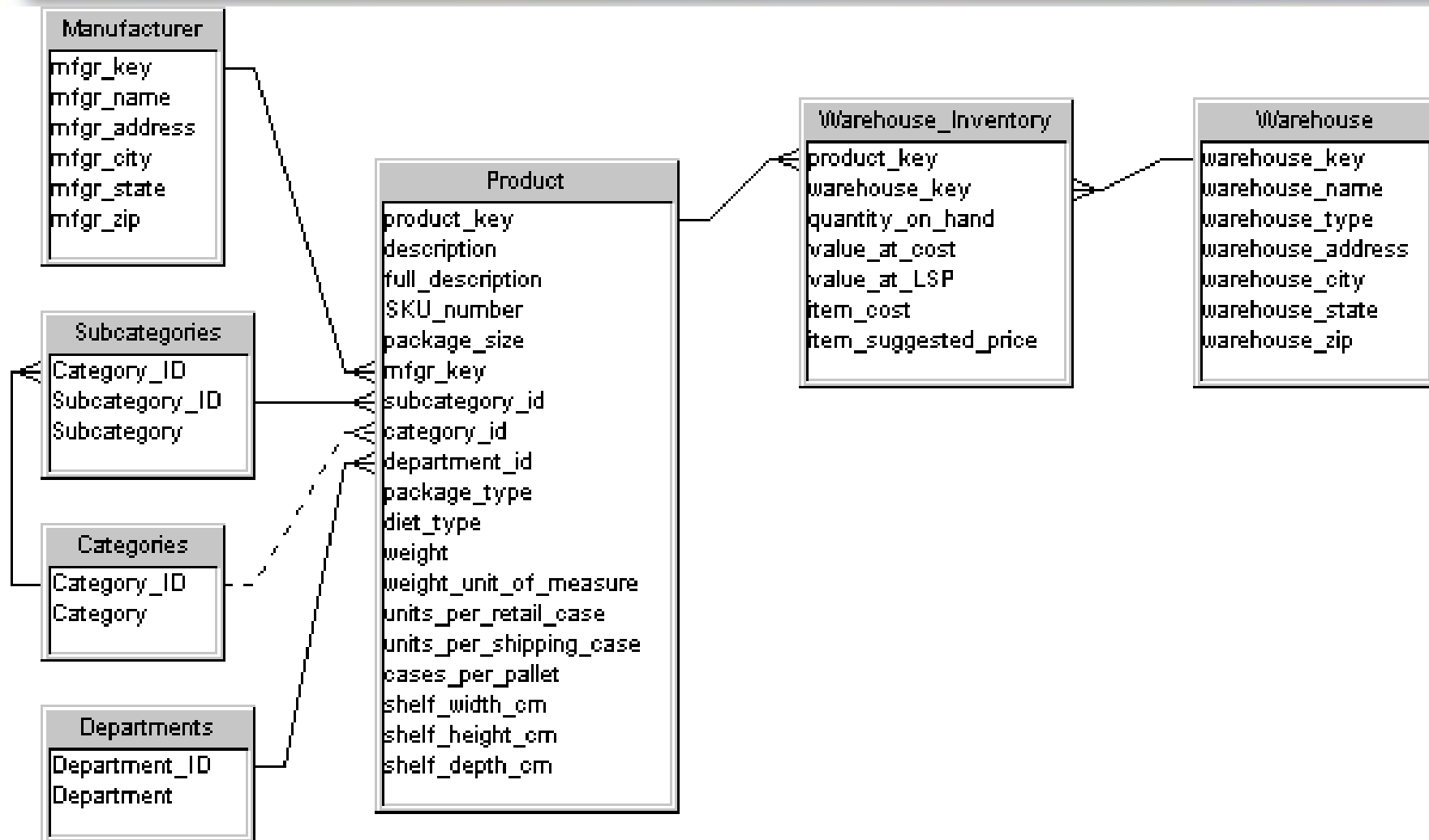
# Using Multiple Data Providers

- **Grocery store sales data**
- **EDI is used to collect inventory data from warehouses**
- **Two universes have been created**
  - **Grocery store sales**
  - **Supplier warehouse stock data**
- **Your mission**
  - **Report sales, cost, profit**
  - **Report available quantity on hand**
  - **Report showing items that will backorder**

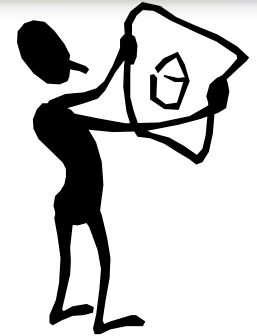
# Grocery Universe Structure



# Warehouse Universe Structure



# Combined Reports



- **Sales by Manufacturer**
  - The Multicube() function is required for this report
- **Gross Margin (Sales Vs. Cost) by SKU / Warehouse**
  - Note: Each warehouse has its own cost structure
- **Days Stock on Hand**
  - Use 30 days sales (12/1/1995 through 12/31/1995) to calculate the average run rate for sales
  - Divide the stock on hand by run rate to determine days stock on hand
  - Use an Alerter to highlight any value with less than one week of stock remaining

See ZEN gDemo 01, 02, and 03.rep



# Design Scenario II

## *Multiple Data Provider Wrap-up*

- **Objects used for linking must be dimensions**
- **Objects used in combined blocks must be**
  - Link-able (common) dimension objects
  - Details associated with key dimension objects
  - Measures
- **Objects in hierarchies must be dimension objects**
- **Some scenarios may require a duplicated object**
  - **Order date** is defined by **order number** and can be a **detail**
  - **Order date** participates in a **hierarchy** and must be a **dimension**

# Design Scenario II

## *Using Multiple Data Providers*

- **When designing universes be aware if this feature may be used in your reporting environment**
- **If so be sure to set up objects for**
  - **Linking**
  - **Hierarchies (Drilling)**
  - **Supporting detail**



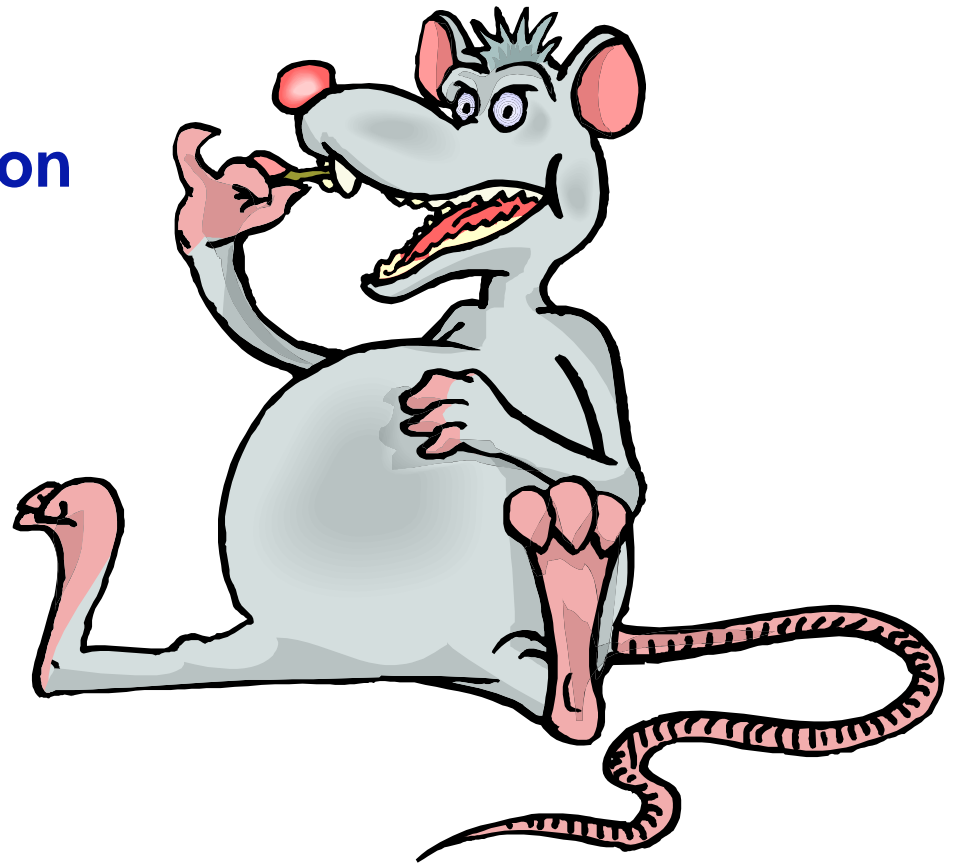


## A Brief Recap

- **R**elationship Identification (Chasm or Fan Traps)
- **A**tttribute Definition (Dimension versus Detail)
- **T**est Process (Write lots of queries!)
  
- **Z**ero Defect Coding (Proper Measure Definition)
- **E**nhance Performance (@Aggregate\_Aware)
- **N**arrow Scope (Contexts)

# Q&A

- **Questions and discussion**



Demonstration support material will be posted on the Integra Solutions web site <http://www.integrasolutions.net> after the conference.





# Thank You!

- Thank you for your time and attention today

*David G. Rathbun*  
*Integra Solutions Inc*  
*[www.IntegraSolutions.net](http://www.IntegraSolutions.net)*