

21st WORLD  
CONTINUOUS  
AUDITING &  
REPORTING  
SYMPOSIUM



**Date:** November 5-6, 2010

**Place:** Rutgers Business School  
1 Washington Park, Lecture Hall 220  
Newark, New Jersey 07102

[raw.rutgers.edu/21wcars](http://raw.rutgers.edu/21wcars)

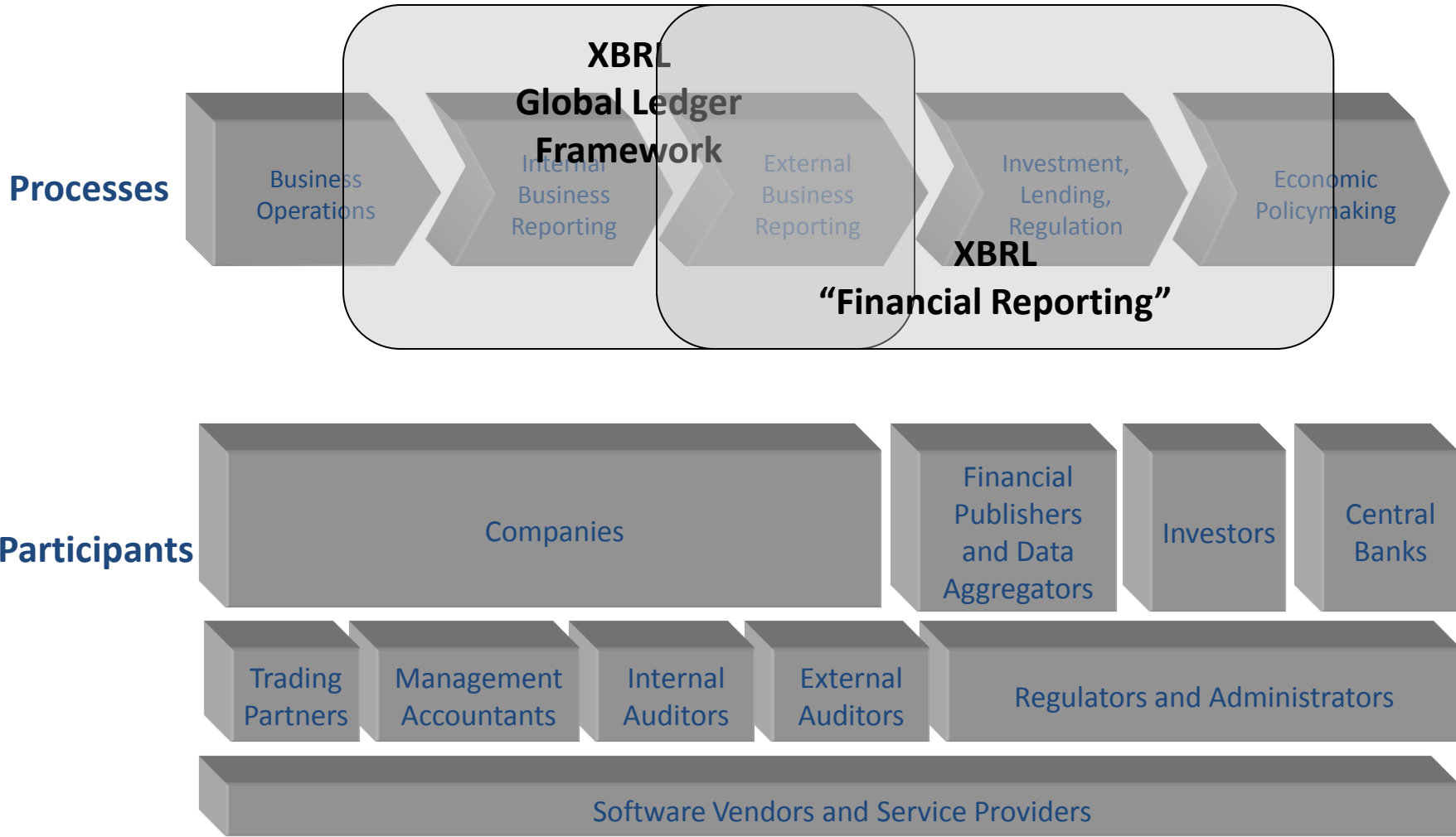
# SQL Access to XBRL Historical Data: A Continuous Benchmarking Story

Eric E. Cohen



SANKT ERIK

# Business Reporting Supply Chain



# XBRL Data Beginning to Proliferate

2011: 8000 more companies in  
Year 3

2010: 1200 more  
companies in Year 2

2009: 500  
companies in  
Year 1

## SEC

- Interactive Data Mandate 33-9002 already underway
- Mutual Funds Risk-Return begins 2011

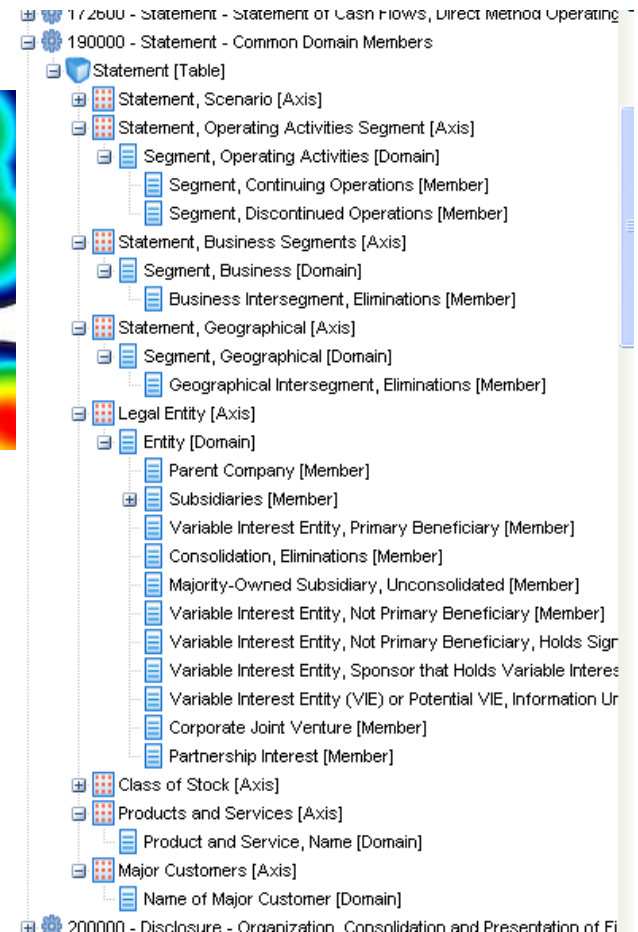
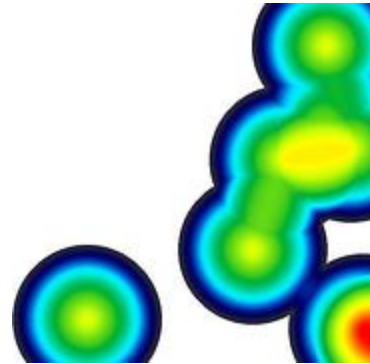
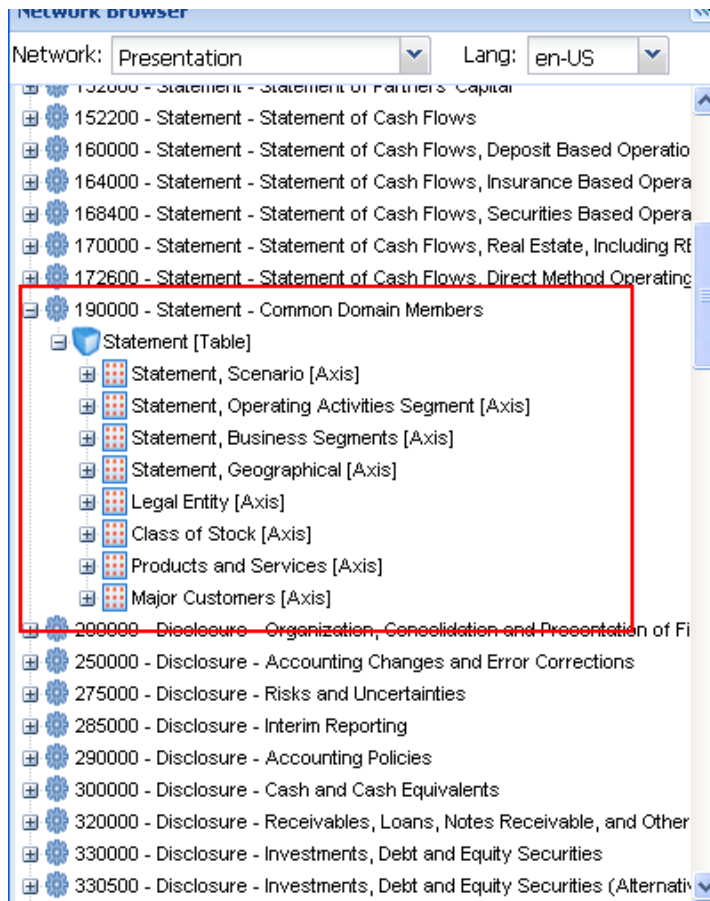
Other sources include:  
US FDIC  
Numerous global regulators  
(e.g., ISA)

If we can't helium or curium, we barium

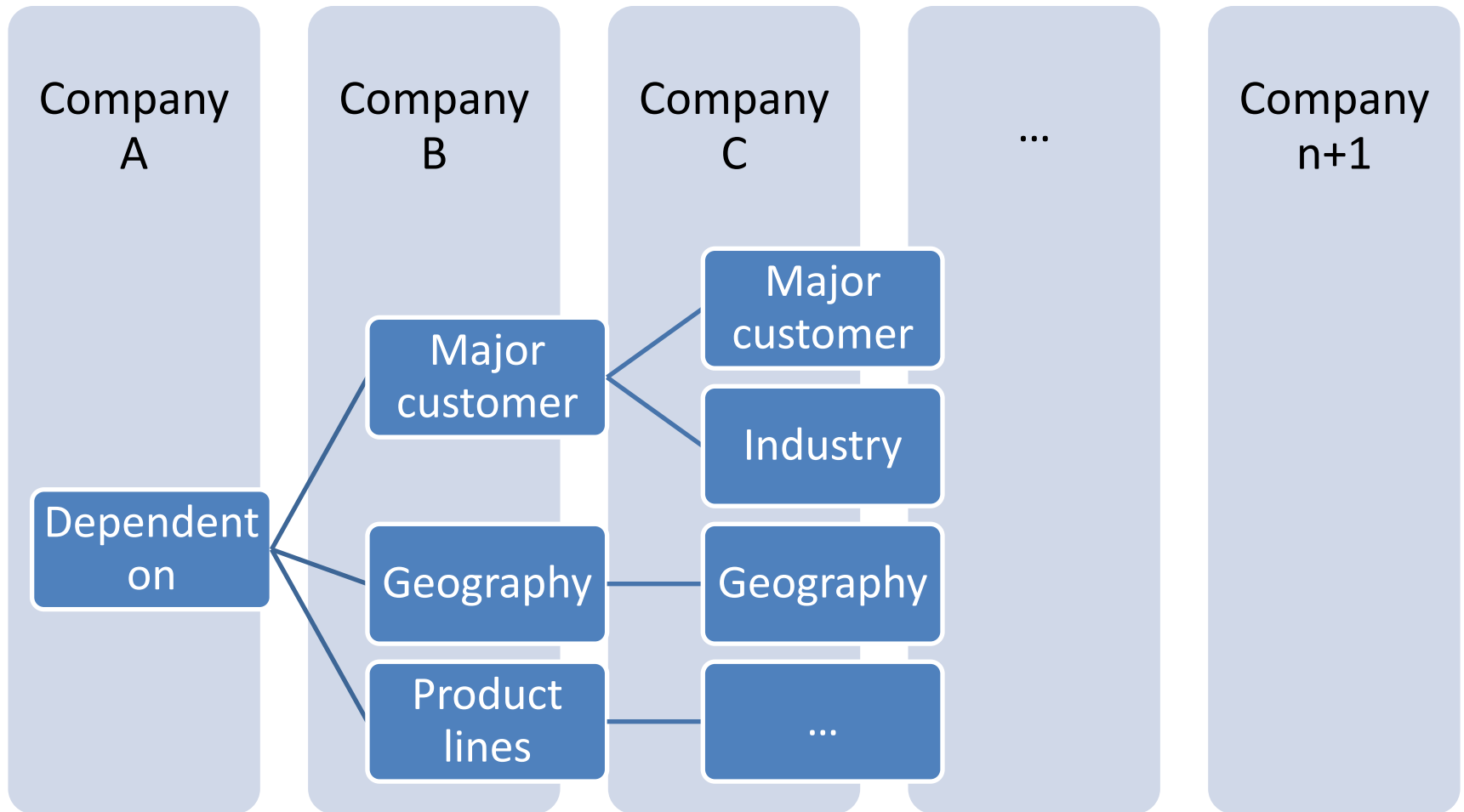


# Exposes Information as Data Points

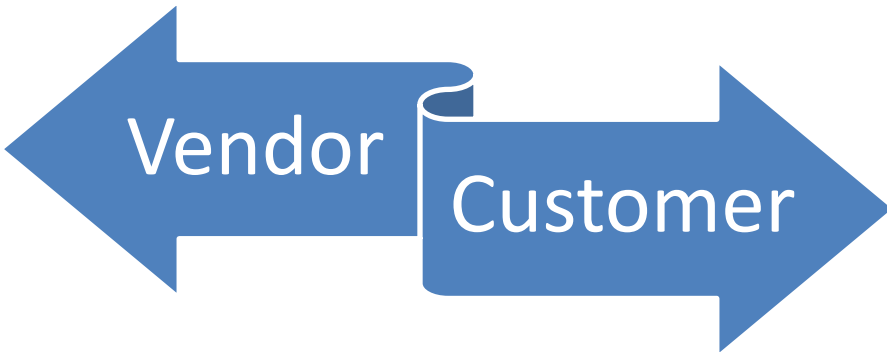
For example: Segmental Reporting



# Theory: Exploiting *Linked* Segmental Reporting

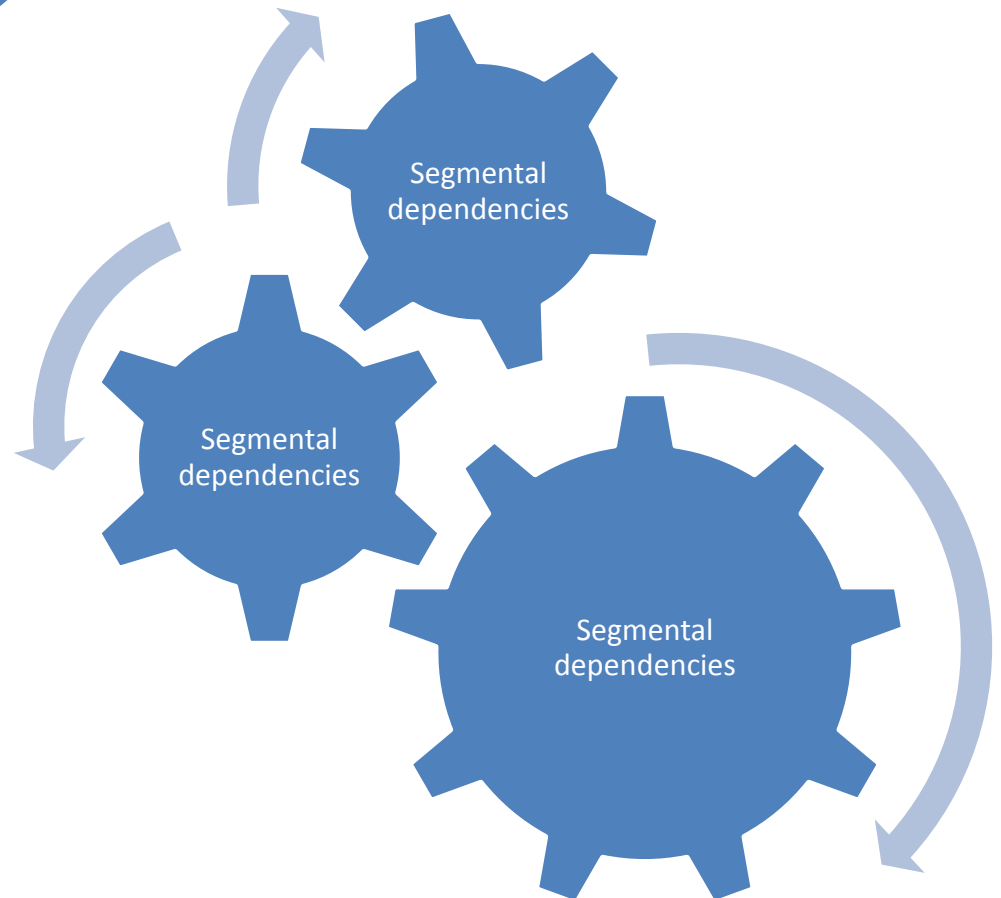


# Dependencies



Who is buying what from whom?  
(and, by extension, who is selling what to whom)?

Company Z goes bankrupt – what is the potential impact?





Reverse Lorenzian Effect  
Aka One Bad Apple Effect

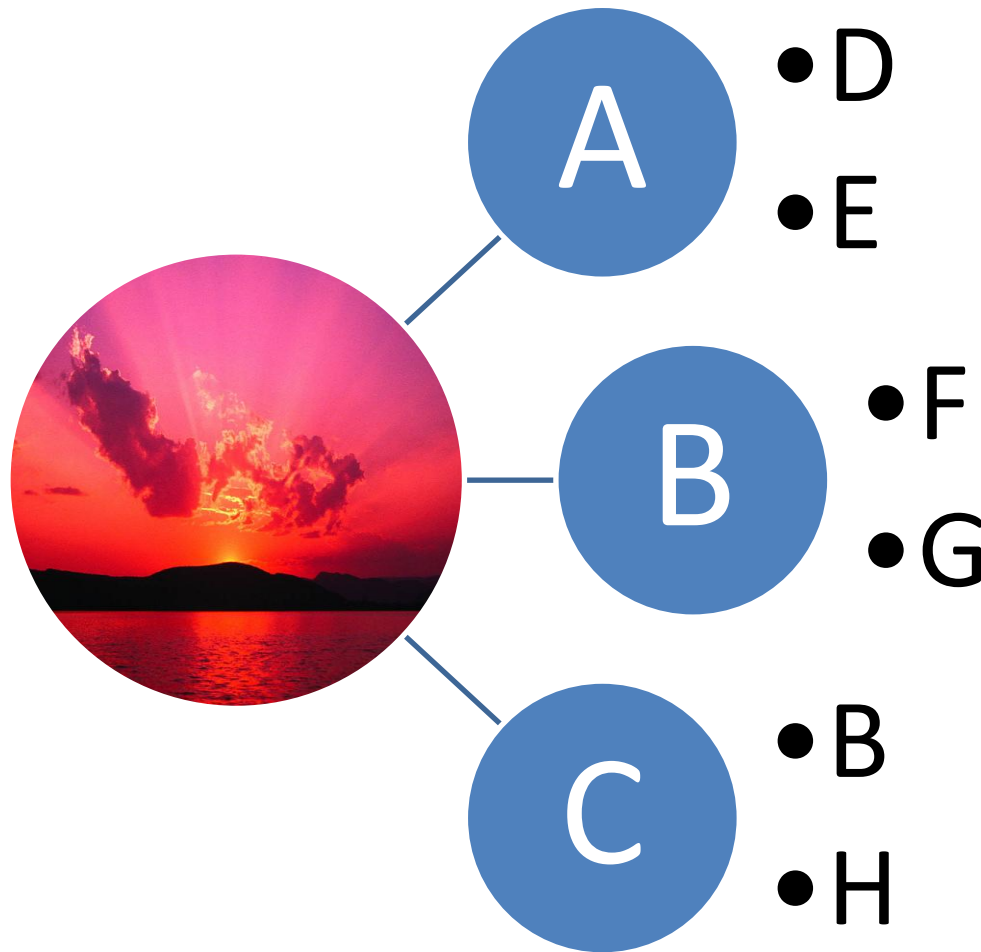


Edward Lorenz



# Enabling the Lorenzian Web

## Example: Customer/Vendor



The volume of a pizza with radius  $z$  and height  $a$  is  $\pi * z * z * a$

# XBRL Turns This Into Data



CONSISTENCY SUITE

Logged in as: ERIC COHEN [Sign Out](#)

**Consistency Checks**

**Company Filings Analysis**

**Element Analysis**

- Label Usage
- Summary of Concepts Reported
- Concepts Reported by Filing
- Values Reported by Filing
- Values Reported (Pivot by Period)

**Extension Analysis**

- Extension % By Filing (Industry)
- View Extension Details

**Basic Filter** **Advanced Filter**

Concept:

Axis:

Member:

CIK:

Filing:

Period:

**Taxonomy View & Compare**

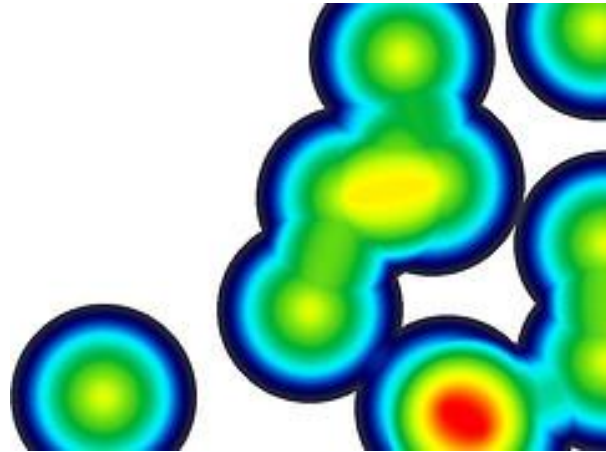
**Online Community and Support**

Count of Concepts Reported (Number of times a concept has been used in Filings)

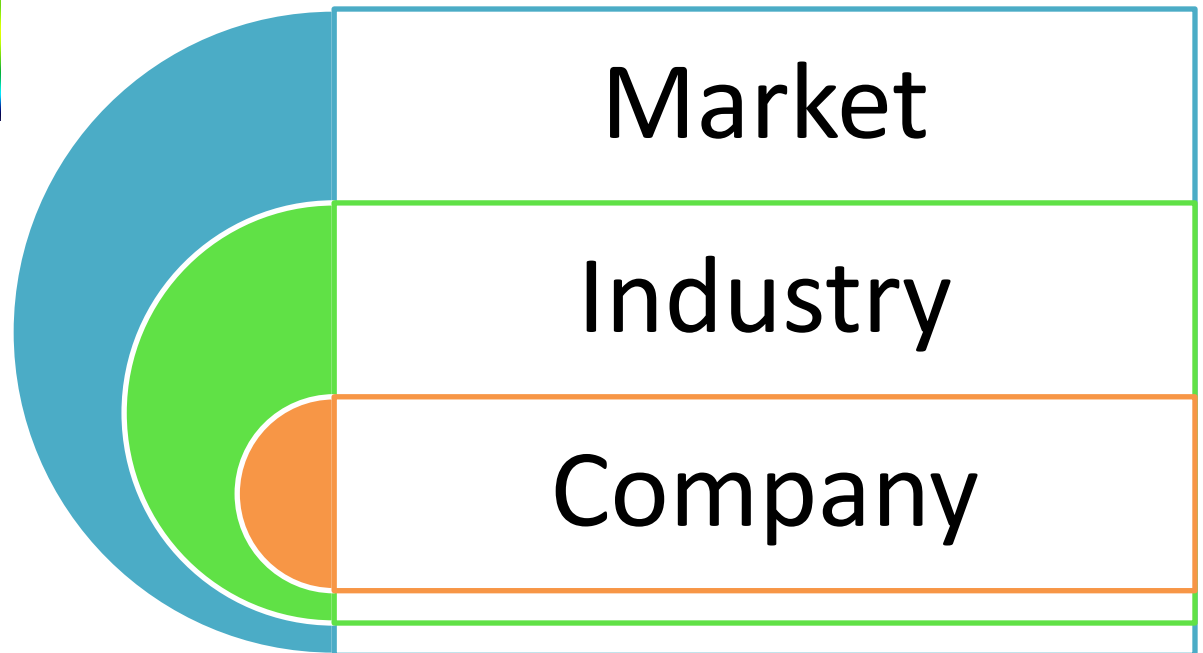
Number of Rows: 34

| Concept Name                                     | Axis               | Member                        | Fact Occurrences | Filings | Companies | Source    |
|--|--------------------|-------------------------------|------------------|---------|-----------|-----------|
| PercentageOfTotalRevenuesFromCustomer            | MajorCustomersAxis | AmerisourceBergenCorpMember   | 4                | 1       | 1         | Extension |
| AssetManagementFees                              | MajorCustomersAxis | BondAndMoneyMarketFundsMember | 8                | 2       | 1         | Taxonomy  |
| AssetsUnderManagementPeriodEndValue              | MajorCustomersAxis | BondAndMoneyMarketFundsMember | 4                | 2       | 1         | Extension |
| AssetsUnderManagementAverage                     | MajorCustomersAxis | BondAndMoneyMarketFundsMember | 8                | 2       | 1         | Extension |
| AccountsReceivableNetCurrent                     | MajorCustomersAxis | CaisoAndPxMember              | 1                | 1       | 1         | Taxonomy  |
| PercentageOfProductNetSalesByMajorCustomer       | MajorCustomersAxis | CardinalHealthIndMember       | 4                | 1       | 1         | Extension |
| PercentageOfTotalRevenuesFromCustomer            | MajorCustomersAxis | CardinalHealthIndMember       | 4                | 1       | 1         | Extension |
| EntityWideRevenueByMajorCustomerPercentOfRevenue | MajorCustomersAxis | GamestopCorpMember            | 2                | 1       | 1         | Extension |
| EntityWideRevenueMajorCustomerAmount             | MajorCustomersAxis | MajorCustomerOneMember        | 2                | 1       | 1         | Taxonomy  |

# What to Follow



What do leading and lagging indicators mean?  
How can this best be visualized/interpreted/leveraged?





# Quantum Reporting: Particles and Waves

The screenshot shows Microsoft Excel with a PivotTable titled 'Count of company-name'. The PivotTable is structured with 'Count of company-name' as the Row Labels and months (Jan through Dec) as Column Labels. The data shows the count of filings for each company in each month. The PivotTable Field List on the right shows the following fields:

- xbrl-instance-url
- accession-number
- central-index-key
- company-name
- form-type
- report-period
- DocumentType
- filing-date
- standard-industrial-classification

The PivotTable data is as follows:

| Count of company-name | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0100                  |     |     |     |     |     |     |     | 1   |     |     |     |     |
| 0700                  |     |     |     |     |     |     |     | 1   |     |     |     |     |
| 1000                  |     | 3   |     | 1   | 4   | 2   | 2   | 6   | 1   | 4   | 1   |     |
| 1040                  |     | 2   | 2   | 2   | 3   |     | 2   | 6   | 2   | 1   | 2   | 1   |
| 1090                  |     |     |     |     |     |     | 1   |     |     | 1   |     |     |
| 1220                  |     | 1   |     |     |     |     |     | 3   |     |     | 1   |     |
| 1221                  |     | 2   | 3   | 2   | 4   |     |     | 12  | 2   | 2   | 4   |     |
| 1311                  |     | 17  | 5   | 5   | 17  |     | 12  | 51  | 3   | 16  | 19  |     |
| 1381                  |     | 7   | 2   | 3   | 6   | 1   | 6   | 16  | 4   | 5   | 10  |     |
| 1389                  |     | 3   |     | 2   |     |     | 7   | 4   |     | 7   | 1   |     |
| 1400                  |     | 2   |     |     | 2   |     | 1   | 7   |     | 2   | 2   |     |
| 1520                  |     |     |     |     |     |     |     |     |     | 1   |     |     |
| 1531                  |     |     |     |     |     |     | 1   | 4   | 1   | 2   |     |     |
| 1600                  |     | 1   | 3   |     | 2   | 3   | 1   | 5   | 4   | 2   | 3   |     |
| 1700                  |     |     |     |     |     |     |     | 1   |     | 1   |     |     |
| 1731                  |     |     |     | 1   | 1   | 1   | 1   | 2   |     |     | 2   |     |
| 2000                  |     | 1   | 2   | 1   | 1   | 2   | 1   | 1   | 4   | 1   | 2   | 3   |
| 2011                  |     |     |     |     |     |     |     |     | 2   |     |     |     |
| 2015                  |     |     |     |     |     |     |     | 1   |     |     |     |     |
| 2024                  |     |     |     |     |     |     |     | 1   |     |     |     |     |
| 2030                  |     | 1   |     |     |     | 1   |     | 2   | 1   |     | 1   |     |
| 2033                  |     |     | 1   |     |     | 1   |     |     | 2   |     |     |     |
| 2040                  |     | 1   | 1   |     | 1   |     | 1   | 3   | 3   |     | 1   | 1   |
| 2052                  |     |     |     |     |     |     | 1   |     |     |     |     |     |
| 2060                  |     | 1   |     |     | 1   |     |     | 3   |     |     | 1   |     |
| 2070                  |     | 1   | 1   |     | 2   |     |     | 4   |     |     | 2   |     |
| 2080                  |     | 4   |     | 2   | 1   |     | 2   | 2   | 1   | 8   |     |     |
| 2082                  |     | 1   |     | 1   | 1   |     |     | 2   | 3   |     | 1   |     |
| 2086                  |     | 1   |     | 1   |     |     | 2   | 1   |     | 2   |     |     |
| 2090                  |     |     |     |     |     |     | 1   |     |     | 2   |     |     |

# Some Industries Already Represented

The screenshot shows an Excel spreadsheet with the following data:

| Row Labels | Jan  | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Grand Total |
|------------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------------|
| 1          |      |     |     |     |     |     |     |     |     |     |     |     |             |
| 8          | 1221 |     | 2   | 3   | 2   | 4   |     |     | 12  | 2   | 2   | 4   | 31          |
| 9          | 1311 |     | 17  | 5   | 5   | 17  |     | 12  | 51  | 3   | 16  | 19  | 145         |
| 10         | 1381 |     | 7   | 2   | 3   | 6   | 1   | 6   | 16  | 4   | 5   | 10  | 60          |
| 61         | 2834 |     | 10  | 3   | 3   | 12  | 1   | 16  | 36  |     | 16  | 12  | 109         |
| 72         | 2911 |     | 8   |     |     | 10  |     | 1   | 18  | 1   |     | 8   | 46          |
| 135        | 3674 | 1   | 7   | 5   | 4   | 6   | 3   | 12  | 31  | 8   | 12  | 7   | 99          |
| 156        | 3841 | 2   | 7   |     | 5   | 5   |     | 9   | 12  | 1   | 7   | 4   | 52          |
| 167        | 4011 |     | 5   |     | 5   | 1   |     | 9   | 2   |     | 11  |     | 33          |
| 180        | 4813 |     | 5   |     | 1   | 3   | 1   | 5   | 16  |     | 4   | 3   | 38          |
| 184        | 4841 |     | 7   | 1   | 4   | 8   | 1   | 2   | 20  | 1   | 1   | 8   | 53          |
| 187        | 4911 |     | 15  | 4   | 4   | 15  |     | 7   | 48  | 4   | 27  | 14  | 138         |
| 188        | 4922 |     | 4   | 2   | 3   | 4   |     | 3   | 15  | 2   | 8   | 3   | 45          |
| 191        | 4931 |     | 8   |     | 1   | 8   |     | 6   | 22  | 1   | 16  | 5   | 67          |
| 239        | 6021 |     | 8   | 2   |     | 11  | 1   | 7   | 36  | 3   | 4   | 13  | 85          |
| 240        | 6022 |     | 6   |     | 1   | 5   | 2   | 2   | 26  | 3   | 3   | 7   | 55          |
| 251        | 6211 |     | 2   | 2   |     | 4   |     | 2   | 19  | 2   | 4   | 5   | 40          |
| 253        | 6282 |     | 4   |     |     | 4   |     | 4   | 13  | 1   | 5   | 4   | 35          |
| 256        | 6324 |     | 5   |     |     | 3   | 2   | 6   | 10  |     | 5   | 4   | 35          |
| 257        | 6331 |     | 7   | 2   | 3   | 8   |     | 6   | 42  | 2   | 9   | 9   | 88          |
| 268        | 6798 |     | 11  | 6   | 2   | 16  |     | 14  | 77  | 6   | 13  | 20  | 165         |
| 280        | 7372 | 4   | 5   | 5   | 2   | 12  | 4   | 7   | 30  | 8   | 15  | 7   | 102         |
| 284        | 7389 |     | 5   | 3   | 1   | 7   | 2   | 6   | 20  | 1   | 10  | 6   | 62          |
| 308        |      |     |     |     |     |     |     |     |     |     |     |     |             |

**306 SIC Codes**  
 96 > 10 filings  
 44 > 20 filings  
 22 > 30 filings  
 16 > 40 filings  
 14 > 50 filings  
 10 > 60 filings  
 8 > 70 filings

REITS 165  
 Petroleum &  
 Gas 145  
 Power 138

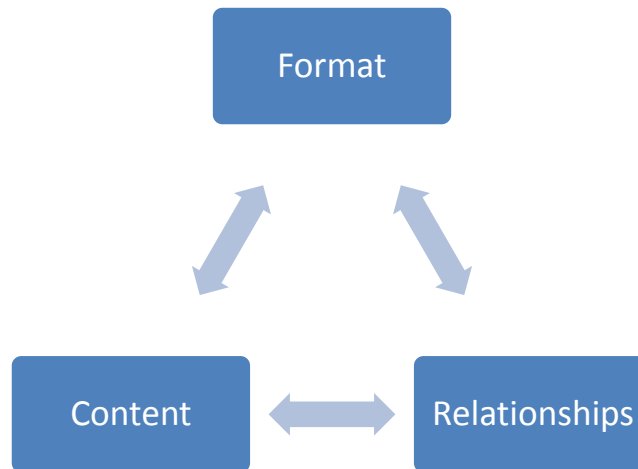
# Benefit of Waves

- Certain disclosures only required annually
- Offset of filings means data/trends can be refreshed more often - with sufficient population of data; psuedo-continuous
- Up to date picture, however grainy

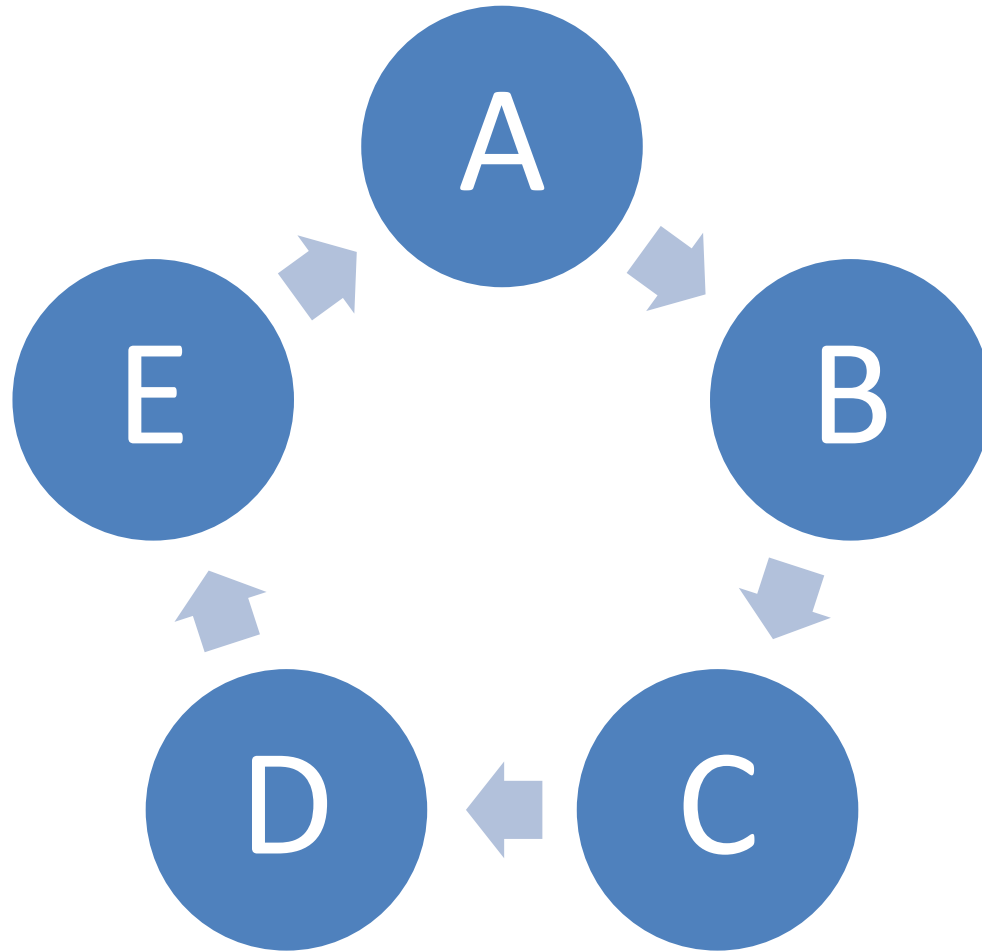


# “TCP/IP” for Business Information

Exploring the ever-changing “business reporting web”



# CIRC



What's a tachyon? A gluon that hasn't dried.

# Frictionless versus Aerodynamic





# Value to Benchmarking

- Performance indicators are relative
  - Sales are up 100% (yay!)
  - (but the competitor's sales are up 200%) (boo!)

The past, the future and the present walk into a bar. It was a tense day.

# Purpose of Benchmarking in the Lorenzian Web

XBRL-specific

- How do my filings compare with others in the industry?

Performance issues

- How does my performance compare with others in my industry?

Investing

- How should my investing strategy change?

Operations

- Where should I concentrate marketing efforts?
- What companies or industries are riskier than they were?

Then he ordered a drink. A tachyon walked into a bar.

# What Does This Require?

Agreement at an increasingly detailed, granular level.

Quality in selection and creation

| Row Labels  | Sum of Fact Occurrences |
|---|-------------------------|
| DomesticMember                                      | 156                     |
| WorldwideMember                                     | 101                     |
| InternationalMember                                 | 96                      |
| AsiaPacificMember                                   | 89                      |
| TotalReportableSegmentsMember                       | 75                      |
| NorthAmericaMember                                  | 67                      |
| LatinAmericaMember                                  | 57                      |
| WesternEuropeMiddleEastAfricaMember                 | 46                      |
| CentralEasternEuropeMember                          | 43                      |
| SwitzerlandMember                                   | 42                      |
| NonSwitzerlandMember                                | 42                      |
| UnitedStatesMember                                  | 39                      |
| ChinaMember   | 39                      |
| SoutheastMember                                     | 37                      |
| SouthwestMember                                     | 37                      |
| NortheastMember                                     | 37                      |
| NorthwestMember                                     | 37                      |
| SalesOfTradeAccountsReceivableNorthAmericaMember    | 34                      |
| BrazilMember  | 28                      |
| GeographicalIntersegmentEliminationsMember          | 28                      |
| CorporateMember                                     | 26                      |
| USMember  | 26                      |
| OtherMember   | 25                      |
| UnitedKingdomMember                                 | 24                      |
| NonUsMember   | 24                      |
| SalesOfTradeAccountsReceivableEuropeMember          | 21                      |
| EuropeMember  | 20                      |
| DampierGasPipelineMember                            | 20                      |
| EuropeMiddleEastAndAfricaMember                     | 19                      |
| CanadaMember  | 18                      |
| JapanMember   | 18                      |
| UnitedStatesAndCanadaMember                         | 18                      |
| SerraDoFacaoProjectMember                           | 18                      |
| BusinessIntersegmentEliminationsInternationalMember | 16                      |
| BusinessIntersegmentEliminationsMember              | 16                      |
| TotalFromOperationsMember                           | 16                      |
| OtherOperatingSegmentMember                         | 16                      |



# Geographical Analysis

## **AggregateGeographicalMember**

AmericaMember, AmericasMember,  
AmericasOtherThanUSMember

ArgentinaMember

AsiaMember, AsiaPacificMember,  
AsiaPacificTimeshareResortsMember,  
AustralasiaAndOtherMember, AustraliaMember,  
AustralianDollarsMember

BarraGrandeMember

BelgiumMember

BrazilMember

BronxNYMember

BulgariaMember

**BusinessIntersegmentEliminationsDomesticMember,  
BusinessIntersegmentEliminationsInternationalMember,  
BusinessIntersegmentEliminationsMember**

CanadaMember

CentralEasternEuropeMember

ChileMember

ChinaMember

ColombiaMember

ColumbiaRiverMember

**ConsolidatedMember, CorporateAndEliminationsMember,  
CorporateMember, CorporationMember**

CostaRicaMember

DampierGasPipelineMember

**DE**

DiabloCanyonMember

DomesticMember

EAMEMember

EastStLouisSiteMember

EMEAMember

**EmergingBusinessesMember, EmergingMarketsMember**

EsteirtoProjectMember

EulesTXMember

EuropeAndAsiaRMember

EuropeanMarketsMember

EuropeanTimeshareAndFractionalBusinessMember

EuropeMarketsMember

EuropeMember

EuropeMiddleEastAndAfricaMember

FinlandMember

ForeignCountriesMember, ForeignCountryMember

**ForeignCurrencyTranslationEurosMember**

# Geographical Analysis

ForeignMember

FranceMember

FusinaSiteMember

**GatewayEnergyAndCokeCompanyLLCMember**

**GeographicalIntersegmentEliminationsMember**

GlobalAndOtherMember

GrasseRiverMember

GreeceMember

HumboldtBayUnitMember

**ImpairmentOfGoodwillMember**

InternationalMember

IsraelMember

**ItalianGovernmentMember**

ItalyMember

JapanMember

KoreaMember

LatinAmericaAndCanadaMember, LatinAmericaMember,  
LatinAmericaMexicoAndCanadaMember

LosAngelesCAMember

**LuxuryResidentialProjectsMember**

MachadinhoAndBarraGrandeMember

MachadinhoMember

ManchesterNHMember

MassenaNyMember

MexicoMember

NetherlandsMember

NigeriaMember

NonSwitzerlandMember, NonUsMember,  
NonUSOperationsMember, NonUSSourcesMember

NorthAmericaMember

**NorthAmericanLuxuryFractionalProjectsMember**

**NorthAmericanTimeshareProjectMember**

NorthAmericaRMember

NortheastMember

NorthwestMember

OffshoreDevelopmentFacilityMember

**OtherCurrenciesMember**

OtherForeignCountriesExcludingGermanyMember

OtherGeographicalAreasMember

Pennies were once made of steel, but CU later

# Geographical Analysis

OtherMember

**OtherOperatingSegmentMember**

OutsideUnitedStatesMember

PittsburghPAMember

PortovesmeSiteMember

**ReconcilingItemsMember**

RefineryMember

RubyPipelineMember

**SalesOfTradeAccountsReceivableAsiaPacificMember**

**SalesOfTradeAccountsReceivableEuropeMember**

**SalesOfTradeAccountsReceivableNorthAmericaMember**

ScotlandMember

**SegmentGeographicalGroupsOfCountriesGroupOneMember,  
SegmentGeographicalGroupsOfCountriesGroupThreeMember,  
SegmentGeographicalGroupsOfCountriesGroupTwoMember**

SerraDoFacaoProjectMember

SoutheastMember

SouthwestMember

SpainMember

**StateFourMember, StateOneMember, StateThreeMember,  
StateTwoMember**

SwitzerlandMember

TaiwanMember

**TariffsGrantedBySpainMember**

ThePhilippinesMember

**TotalFromOperationsMember,  
TotalReportableSegmentsMember, TotalSalesMember**

TucsonAZMember

TucsonAZPreferredEquityMember

TurkeyMember

UkAndJapanOperationsMember

UnitedKingdomMember

UnitedKingdomMoneyMarketLineBorrowingMember

UnitedStatesAndCanadaMember

UnitedStatesDollarsMember

UnitedStatesMember

US

USAndPuertoRicoMember

USMember

USOperationsMember

USSmeltingMember

USSourcesMember

VancouverSmelterSiteMember

VariousSitesMember

VE

VenezuelaMember

WesternEuropeMiddleEastAfricaMember

WorldwideMember

# Frictionless Data

- Starting up from rest on a frictionless surface is not an intuitive task, due to Newton's 3<sup>rd</sup> Law
- Until you have a start, you don't know if you are going anywhere!



What works after it is fired? A neuron.



# Getting at Data

- Data query languages and techniques
- Commercial sources
- Collaborative sources

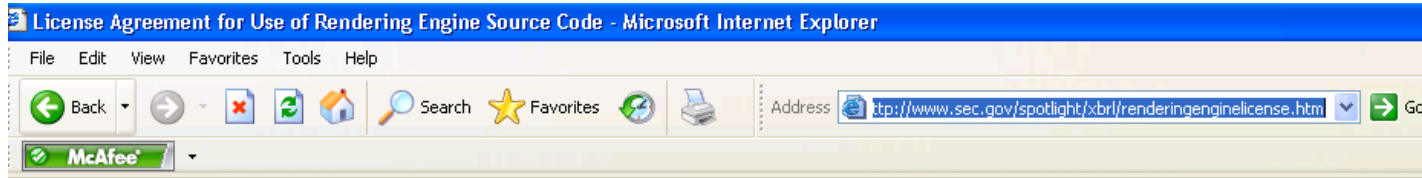
# Things to Do With XBRL

- View it
- Audit it
- Analyze it
- Transform it
- Move it <- I like to do this

# Readily Available Tooling

- View it
  - SEC Open Source Rendering Tool
- Query it
  - SQL
  - XQuery
  - Transform and then use other tools
    - SPARQ
    - SQL and other traditional database tools

# SEC Open Source Rendering: Freely Available Tool



days, Licensor may immediately terminate this License Agreement, and shall notify Licensee in writing of such termination within five (5) days of such termination. The rights of recipients of the Rendering Engine Source Code from Licensee who remain in compliance with the terms of this License Agreement shall be unaffected by any such termination.

## 10. Entire Agreement

This License Agreement represents the complete agreement of Licensor and Licensee concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

## 12. Choice of Law

This Agreement shall be interpreted in accordance with the federal common law as interpreted by the U.S. District Court for the District of Columbia and its reviewing courts.

## 13. Indemnity

Licensee agrees to indemnify and hold Licensor harmless from any and all claims, demands, causes of action, damages, judgments and attorney's fees and costs arising from Licensee's using, offering, selling, promoting, or distribution of the Rendering Engine Source Code.

- ▶ [Download the Rendering Engine configurable binary distribution](#) (Note: 17 MB)
- ▶ [Download the Rendering Engine source code](#) (Note: 55 MB)  
(We suggest working with the configurable binary distribution before the source code.)

<http://www.sec.gov/spotlight/xbrl/renderingenginelicense.htm>

Command-line tool  
XSLT transformation

Read and follow directions:

MS-DOS skills  
Text editing skills  
XSL handling skills

# Commercial Software

- Example:
  - Excel-based tooling
    - IMetrix
    - Xinba



# What is Xinba

- Xinba is an XBRL viewer.
  - Users can view XBRL within Excel.
  - Users can also create templates to show XBRL their own way.



## Standard Sheet View

A screenshot of a Microsoft Excel spreadsheet window. The spreadsheet has a table with the following structure:

|              | <u>A Corp.</u><br>2004-1Q | <u>B Corp.</u><br>2004-2Q | <u>C Corp.</u><br>2004-3Q | <u>D Corp.</u><br>2004-2Q |
|--------------|---------------------------|---------------------------|---------------------------|---------------------------|
| <u>Label</u> |                           |                           |                           |                           |
| Asset        |                           |                           |                           |                           |
| xxx          |                           |                           |                           |                           |
| xxx          |                           |                           |                           |                           |
| xxx          |                           |                           |                           |                           |

A yellow oval with the text "Standard Table Style" is overlaid on the table.

## Style Sheet view

A screenshot of a Microsoft Excel spreadsheet window. The spreadsheet has a table with the following structure:

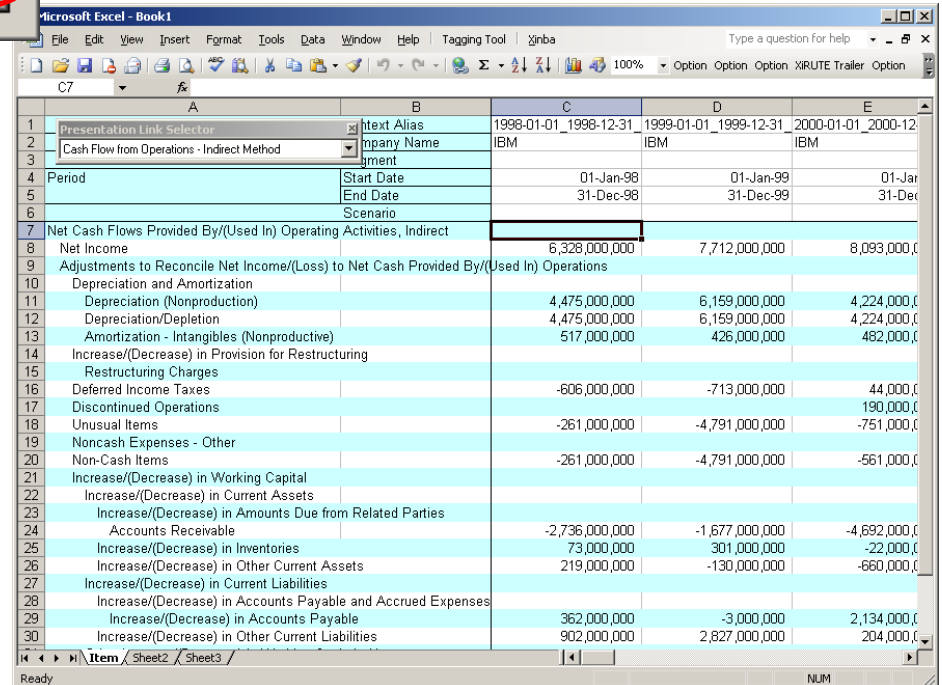
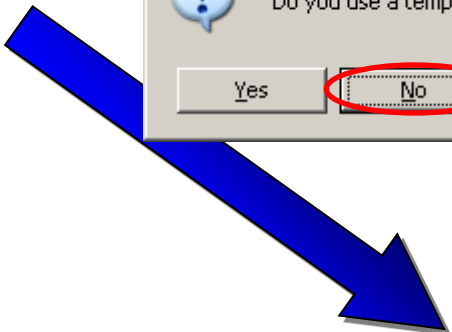
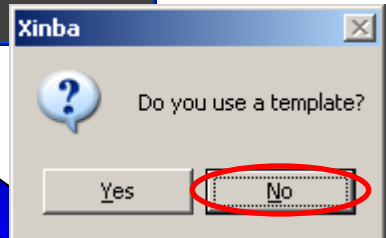
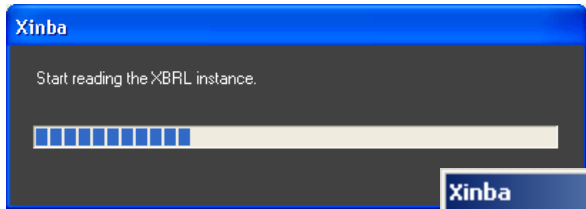
|                        |                             |
|------------------------|-----------------------------|
| <u>A Corp. 2004-1Q</u> |                             |
| <u>Label</u>           | =XBRL Value (xxx, yyy, zzz) |
| xxx                    |                             |
| xxx                    |                             |
| xxx                    |                             |

A yellow oval with the text "Free Style" is overlaid on the table.

Xinba provides Excel functions to get specific XBRL values into cells.

# Open with standard sheet

- Open with standard sheet
  - Set company to the company selection window and click OK.



|    | A   | B            | C                     | D                     | E                     |
|----|---|--------------|-----------------------|-----------------------|-----------------------|
| 1  | Presentation Link Selector  | Text Alias   | 1998-01-01_1998-12-31 | 1999-01-01_1999-12-31 | 2000-01-01_2000-12-31 |
| 2  | Cash Flow from Operations - Indirect Method   | Company Name | IBM                   | IBM                   | IBM                   |
| 3  |   | Segment      |                       |                       |                       |
| 4  | Period  | Start Date   | 01-Jan-98             | 01-Jan-99             | 01-Jan-00             |
| 5  |   | End Date     | 31-Dec-98             | 31-Dec-99             | 31-Dec-00             |
| 6  |   | Scenario     |                       |                       |                       |
| 7  | Net Cash Flows Provided By/(Used In) Operating Activities, Indirect                     |              |                       |                       |                       |
| 8  | Net Income  |              | 6,328,000,000         | 7,712,000,000         | 8,093,000,000         |
| 9  | Adjustments to Reconcile Net Income/(Loss) to Net Cash Provided By/(Used In) Operations |              |                       |                       |                       |
| 10 | Depreciation and Amortization   |              |                       |                       |                       |
| 11 | Depreciation (Nonproduction)  |              | 4,475,000,000         | 6,159,000,000         | 4,224,000,000         |
| 12 | Depreciation/Depletion  |              | 4,475,000,000         | 6,159,000,000         | 4,224,000,000         |
| 13 | Amortization - Intangibles (Nonproductive)  |              | 517,000,000           | 426,000,000           | 482,000,000           |
| 14 | Increase/(Decrease) in Provision for Restructuring                                      |              |                       |                       |                       |
| 15 | Restructuring Charges   |              |                       |                       |                       |
| 16 | Deferred Income Taxes   |              | -606,000,000          | -713,000,000          | 44,000,000            |
| 17 | Discontinued Operations   |              |                       |                       | 190,000,000           |
| 18 | Unusual Items   |              | -261,000,000          | -4,791,000,000        | -751,000,000          |
| 19 | Noncash Expenses - Other  |              |                       |                       |                       |
| 20 | Non-Cash Items  |              | -261,000,000          | -4,791,000,000        | -561,000,000          |
| 21 | Increase/(Decrease) in Working Capital  |              |                       |                       |                       |
| 22 | Increase/(Decrease) in Current Assets   |              |                       |                       |                       |
| 23 | Increase/(Decrease) in Amounts Due from Related Parties                                 |              |                       |                       |                       |
| 24 | Accounts Receivable   |              | -2,736,000,000        | -1,677,000,000        | -4,692,000,000        |
| 25 | Increase/(Decrease) in Inventories  |              | 73,000,000            | 301,000,000           | -22,000,000           |
| 26 | Increase/(Decrease) in Other Current Assets   |              | 219,000,000           | -130,000,000          | -660,000,000          |
| 27 | Increase/(Decrease) in Current Liabilities  |              |                       |                       |                       |
| 28 | Increase/(Decrease) in Accounts Payable and Accrued Expenses                            |              |                       |                       |                       |
| 29 | Increase/(Decrease) in Accounts Payable   |              | 362,000,000           | -3,000,000            | 2,134,000,000         |
| 30 | Increase/(Decrease) in Other Current Liabilities  |              | 902,000,000           | 2,827,000,000         | 204,000,000           |

Microsoft Excel - Book1

File Edit View Insert Format Tools Data Window Help Xinba

Type a question for help

Xinba - iDP Toolbar

Presentation Link Selector

- Condensed Consolidated Balance Sheet
- Condensed Consolidated Cash Flows Statement
- Condensed Consolidated Balance Sheet
- Consolidated Statement of Income
- Tuple Content Models - Common Terms

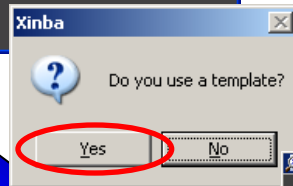
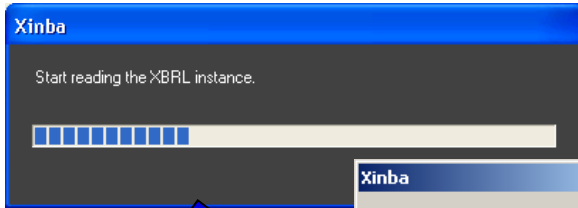
|    | A   | B             | C  | F   |
|----|---|---------------|--|---|
| 1  |   | Context Alias | 2008-01-01_2008-03-31                    |   |
| 2  |   | Company Name  | 0000037996                               | 2007-01-01_2007-03-31   |
| 3  |   | Segment       |  |   |
| 4  | Period  | Start Date    | 01-Jan-08                                | 01-Jan-07   |
| 5  |   | End Date      | 31-Mar-08                                | 31-Mar-07   |
| 6  |   | Scenario      | <fmc:Unaudited xmlns:fr                  | <fmc:Unaudited xmlns:fmc="http://www.ford.com/us/fr/fmc/2008-03-31" |
| 7  | Balance sheet   |               |  |   |
| 8  | Assets  |               |  |   |
| 9  | Cash and Cash Equivalents                                       |               | 33,751,000,000                           | 33,751,000,000  |
| 10 | Marketable Securities - Current                                 |               | 8,593,000,000                            | 26,859,000  |
| 11 | Loaned securities   |               | 6,746,000,000                            |   |
| 12 | Financing Receivables   |               | 108,858,000,000                          |   |
| 13 | Other Receivables, Net  |               | 8,089,000,000                            |   |
| 14 | Net investment in operating leases                              |               | 32,493,000,000                           |   |
| 15 | Retained interest in sold receivables                           |               | 474,000,000                              |   |
| 16 | Inventories, Net  |               | 11,721,000,000                           |   |
| 17 | Investment in Affiliates  |               | 3,120,000,000                            |   |
| 18 | Property, Plant and Equipment, Net                              |               | 37,007,000,000                           |   |
| 19 | Deferred income taxes   |               | 3,331,000,000                            |   |
| 20 | Goodwill and other intangible assets                            |               | 2,064,000,000                            |   |
| 21 | Assets of discontinued/held-for-sale operations                 |               | 10,002,000,000                           |   |
| 22 | Other Assets  |               | 16,664,000,000                           |   |
| 23 | Assets  |               | 282,913,000,000                          |   |
| 24 | Liabilities and Stockholders' Equity                            |               |  |   |
| 25 | Accounts Payable  |               | 23,964,000,000                           |   |
| 26 | Accrued liabilities and deferred revenue                        |               | 72,858,000,000                           |   |
| 27 | Long-Term Debt  |               | 169,205,000,000                          |   |
| 28 | Deferred income taxes   |               | 2,901,000,000                            |   |
| 29 | Liabilities of Discontinued Operations - Current and Noncurrent |               | 5,408,000,000                            |   |
| 30 | Liabilities   |               | 274,336,000,000                          |   |
| 31 | Minority Interest   |               | 1,466,000,000                            |   |
| 32 | Stockholder's Equity  |               |  |   |
| 33 | Capital stock   |               |  |   |
| 34 | Common Stock  |               |  |   |
| 35 | Common Stock - Description                                      |               | Common stock, par value \$0.01 per share |   |

Item Sheet2 Sheet3

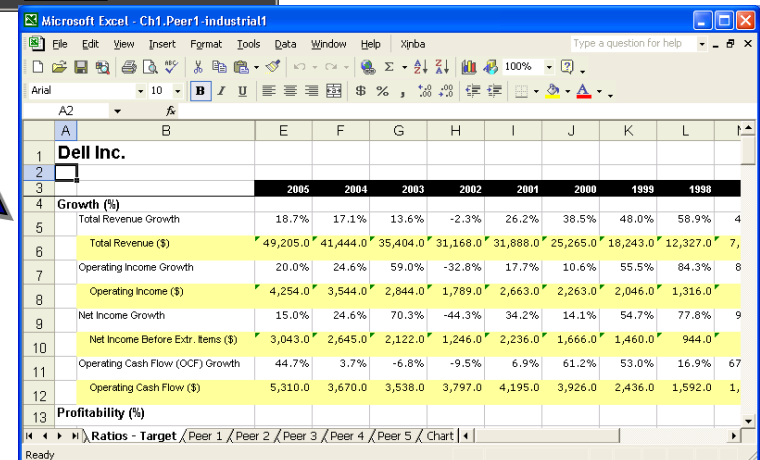
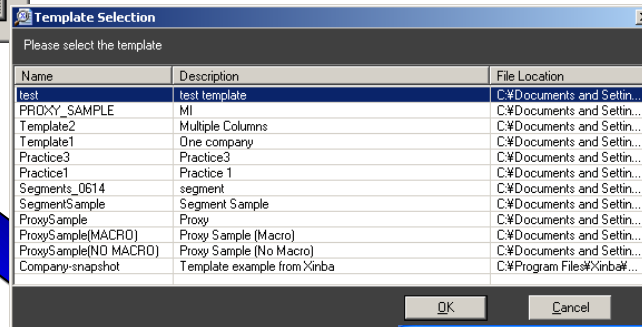
Ready

# Open with template

- Open with standard sheet
  - Set company to the company selection window and click OK.



– Select template



|                                    | 2005     | 2004     | 2003     | 2002     | 2001     | 2000     | 1999     | 1998     |
|------------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>Growth (%)</b>                  |          |          |          |          |          |          |          |          |
| Total Revenue Growth               | 18.7%    | 17.1%    | 13.6%    | -2.3%    | 26.2%    | 38.5%    | 48.0%    | 58.9%    |
| Total Revenue (\$)                 | 49,205.0 | 41,444.0 | 35,404.0 | 31,168.0 | 31,888.0 | 25,265.0 | 18,243.0 | 12,327.0 |
| Operating Income Growth            | 20.0%    | 24.6%    | 59.0%    | -32.8%   | 17.7%    | 10.6%    | 55.5%    | 84.3%    |
| Operating Income (\$)              | 4,254.0  | 3,544.0  | 2,844.0  | 1,789.0  | 2,663.0  | 2,263.0  | 2,046.0  | 1,316.0  |
| Net Income Growth                  | 15.0%    | 24.6%    | 70.3%    | -44.3%   | 34.2%    | 14.1%    | 54.7%    | 77.8%    |
| Net Income Before Extr. Items (\$) | 3,043.0  | 2,645.0  | 2,122.0  | 1,246.0  | 2,236.0  | 1,666.0  | 1,460.0  | 944.0    |
| Operating Cash Flow (OCF) Growth   | 44.7%    | 3.7%     | -6.8%    | -9.5%    | 6.9%     | 61.2%    | 53.0%    | 16.9%    |
| Operating Cash Flow (\$)           | 5,310.0  | 3,670.0  | 3,538.0  | 3,797.0  | 4,195.0  | 3,926.0  | 2,436.0  | 1,592.0  |
| <b>Profitability (%)</b>           |          |          |          |          |          |          |          |          |

# SQL

- A “standard”
  - ANSI/SQL (1986)
  - Variants
    - Oracle
    - Postgres
    - Many others



# Examples of SQL Tools

- General
  - Altova DatabaseSpy (30 day eval)
    - [www.altova.com](http://www.altova.com)
  - RazorSQL
    - [www.razorsql.com](http://www.razorsql.com)
- For Postgres (XBRL US database)
  - **pgAdmin III (open source)**
    - <http://www.pgadmin.org/>

# Tools with SQL

- Fujitsu XWand
  - <http://www.fujitsu.com/global/services/software/interstage/xbrltools/>
  - Documentation in text file
    - readme-SQLPlugin.txt

# Query Syntax

**select** [distinct] <expression>[, <expression>...]  
**from** <table\_name> [, <table\_name>...]  
[**where** <condition>]  
[**group by** <expression> [, <expression>...]]  
[**having** <condition>]  
[**order by** <expression> [asc | desc],  
[<expression> [asc | desc]]..]

# SELECT

- What do you want returned
- Fields, calculated and derived information
- Need to know your choices
  
- Can see just about everything by using the wild card “\*”, and then narrow down from there.

FROM

It's all about  
the data sources  
Actual or virtual



# WHERE

- Conditions driving which records are selected
  - Used for linking key fields between disparate tables
  - Providing other statements that provide Boolean responses (you are in, or you are out)

# ORDER BY

- Sort the information

# Tooling and Resources

- XBRL Data Sources
- Tooling

# Repositories: XBRL US CSuite

The screenshot displays the pgAdmin III interface. On the left, the Object browser shows a tree view of database objects, including Tables (32) and Views (6). The 'accession\_industry\_div' view is selected, and its columns are listed: accession\_id, filing\_date, entity\_id, and creation\_software.

The Properties tab on the right shows the following details for the 'accession\_industry\_div' view:

| Property     | Value  |
|--------------|--|
| Name         | accession_industry_div   |
| OID          | 103033   |
| Owner        | postgres   |
| ACL          | {postgres=arwdDxt/postgres,cpryde=arwdDxt/postgres,xbrl_view_only=r/postgres}        |
| Definition   | SELECT accession.accession_id, accession.filing_date, accession.entity_id, accession |
| System view? | No   |
| Comment      |  |

The SQL pane at the bottom shows the SQL definition for the view:

```
-- View: accession_industry_div
-- DROP VIEW accession_industry_div;
CREATE OR REPLACE VIEW accession_industry_div AS
SELECT accession.accession_id, accession.filing_date, accession.entity_id, a
FROM accession, ( SELECT accession.standard_industrial_classification, acc
CASE
WHEN foo.sic_11 < 10::numeric AND foo.sic_11 >= 1::numeri
WHEN foo.sic_11 < 15::numeric AND foo.sic_11 >= 10::numeri
WHEN foo.sic_11 < 18::numeric AND foo.sic_11 >= 15::numeri
WHEN foo.sic_11 < 22::numeric AND foo.sic_11 >= 20::numeri
WHEN foo.sic_11 < 40::numeric AND foo.sic_11 >= 22::numeri
```

Retrieving View details... Done. 0.38 secs

# Specialize for XBRL

- Actual approach – used in XBRL US historical analyses
  - Very normalized
  - QName information accessible for many purposes
- Virtual approach – used by Fujitsu for accessing information in individual filings

# XBRL US Historical Company Analysis

Object browser

- edgar\_db
  - Catalogs (2)
  - Schemas (1)
    - public
      - Domains (0)
      - Functions (26)
      - Sequences (23)
      - Tables (32)
        - accession
        - accession\_document\_association
        - accession\_timestamp
        - attribute\_value
        - context
        - context\_dimension
        - custom\_arcrole\_type
        - custom\_arcrole\_used\_on
        - custom\_role\_type
        - custom\_role\_used\_on
        - document
        - element
        - element\_attribute
        - element\_attribute\_value\_association
        - entity
        - enumeration\_arcrole\_cycles\_allowed
        - enumeration\_element\_balance
        - enumeration\_element\_period\_type
        - enumeration\_unit\_measure\_location
        - fact
        - label\_resource
        - network
        - qname
        - reference\_part
        - reference\_part\_type
        - reference\_resource
        - relationship
        - resource
        - sic\_code
        - unit
        - unit\_measure
        - uri
      - Trigger Functions (0)
      - Views (6)

Properties

| Property       | Value   |
|----------------|---|
| Name           | public  |
| OID            | 2200  |
| Owner          | postgres  |
| ACL            | {postgres=UC/postgres,xbrl_view_only=U/postgres,xbrl_admin=UC/postgres} |
| System schema? | No  |
| Comment        | standard public schema  |

SQL pane

```
-- Schema: "public"
-- DROP SCHEMA public;

CREATE SCHEMA public
  AUTHORIZATION postgres;
GRANT ALL ON SCHEMA public TO postgres;
GRANT USAGE ON SCHEMA public TO xbrl_view_only;
GRANT ALL ON SCHEMA public TO xbrl_admin;
COMMENT ON SCHEMA public IS 'standard public schema';
```



# Non-Trivial from Normalized Database

- SELECT entity.entity\_id, entity.entity\_name, accession.accession\_id, accession.filing\_accession\_number, context.context\_id, context.context\_xml\_id, context.period\_start, context.period\_end, context.period\_instant, context\_dimension.context\_dimension\_id, contextdimensionqname.local\_name AS context\_dimension\_qname, contextdimensionmemberqname.local\_name AS dimension\_member\_qname, element.element\_id, elementqname.local\_name AS element\_qname, elementbasedatatypeqname.local\_name AS element\_base\_datatype, elementdatatypeqname.local\_name AS element\_datatype, elementsubgroupqname.local\_name AS element\_substitution\_group, element.balance\_id AS balance\_code,
- CASE
- WHEN element.balance\_id = 1 THEN 'Debit'::text
- WHEN element.balance\_id = 2 THEN 'Credit'::text
- ELSE NULL::text
- END AS balance, element.period\_type\_id AS period\_type\_code,
- CASE
- WHEN element.period\_type\_id = 1 THEN 'Instant'::text
- WHEN element.period\_type\_id = 2 THEN 'Duration'::text
- WHEN element.period\_type\_id = 3 THEN 'Forever'::text
- ELSE NULL::text
- END AS period\_type, element.abstract, element.nullable, fact.fact\_id, fact.fact\_value, unit.unit\_id, unit.unit\_xml\_id, unit\_measure.unit\_measure\_id, unitmeasureqname.local\_name AS unit\_measure\_qname, unit\_measure.location\_id AS location\_code,
- CASE
- WHEN unit\_measure.location\_id = 1 THEN 'measure'::text
- WHEN unit\_measure.location\_id = 2 THEN 'numerator'::text
- WHEN unit\_measure.location\_id = 3 THEN 'denominator'::text
- ELSE NULL::text
- END AS location
- FROM fact
- JOIN accession ON fact.accession\_id = accession.accession\_id
- JOIN entity ON accession.entity\_id = entity.entity\_id
- JOIN element ON fact.element\_id = element.element\_id
- JOIN qname elementqname ON element.qname\_id = elementqname.qname\_id
- JOIN qname elementbasedatatypeqname ON element.xbrl\_base\_datatype\_qname\_id = elementbasedatatypeqname.qname\_id
- JOIN qname elementdatatypeqname ON element.datatype\_qname\_id = elementdatatypeqname.qname\_id
- JOIN qname elementsubgroupqname ON element.substitution\_group\_qname\_id = elementsubgroupqname.qname\_id
- JOIN context ON fact.context\_id = context.context\_id
- LEFT JOIN context\_dimension ON context\_dimension.context\_id = context.context\_id
- LEFT JOIN qname contextdimensionqname ON context\_dimension.dimension\_qname\_id = contextdimensionqname.qname\_id
- LEFT JOIN qname contextdimensionmemberqname ON context\_dimension.member\_qname\_id = contextdimensionmemberqname.qname\_id
- LEFT JOIN unit ON fact.unit\_id = unit.unit\_id
- LEFT JOIN unit\_measure ON unit\_measure.unit\_id = unit.unit\_id
- LEFT JOIN qname unitmeasureqname ON unit\_measure.qname\_id = unitmeasureqname.qname\_id
- ORDER BY entity.entity\_id, accession.accession\_id, context.context\_id, elementqname.local\_name;

# Using Fujitsu XWand's Data Query Tools

Use “simple” SQL commands to select and report data

Review on-screen, Cut-and-paste or export to CSV

Note: Freely licensed for academic, consortium use

<http://www.fujitsu.com/global/services/software/interstage/xbrltools/>

# Many Uses; Hidden Power

**Query Statement**

```
select element, elementTypename from element where elementTypename = /string/ and elementprefix = /met/i and element != /abstract/i and element != /Axis/i
```

Execute  
Register  
Create View  
Settings

|   | Element   | ElementTypeName |
|---|---|-----------------|
| 1 | met:NetIncomeAvailableToCommonShareholdersPerCommonShare                      | stringItem      |
| 2 | met:IncomeFromContinuingOperationsAvailableToCommonShareholdersPerCommonShare | stringItem      |
| 3 | met:IncorporationDate   | stringItem      |
| 4 | met:ProjectedTransactionCloseDateOfAcquisition                                | stringItem      |
| 5 | met:CommitmentLetterAmendedAndRestated  | stringItem      |
| 6 | met:TransfersBetweenLevelsOneAndTwo   | stringItem      |
| 7 | met:DerecognitionOfSecuritiesClassifiedAsEquitySecuritiesViaGuidance          | stringItem      |

./000119312510170695/slb-20100630.xml - SCHLUMBERGER LTD /NV/ [10-Q, 2010-06-30]

Tools Help

Query Table

Instance Table Query Table Dimension Table

**Query Statement**

```
select element, value from fact where unitcontent = /pure/ and element != /rate/i
```

Execute  
Register  
Create View  
Settings

|   | Element  | Value    |
|---|--|----------|
| 1 | slb:IntangibleAssetsEstimatedUsefulLife        | 19       |
| 2 | us-gAAP:FiniteLivedIntangibleAssetsWeighted... | 14000000 |

Details

Summary:

Element De

type

name

id

type

Label:

type

# Query Syntax

The query syntax is as follows:

```
select [distinct] <expression>[, <expression>...] from  
<table_name> [, <table_name>...]  
[where <condition>]  
[group by <expression> [, <expression>...]]  
[having <condition>]  
[order by <expression> [asc | desc], [<expression> [asc |  
desc]]..]
```

Note: If you specify "distinct", duplicated rows (if any) will be removed.

# Select

- Data fields necessary for each area of information

# Select Clause

- You can specify multiple items. In such a case, specify them by separating them with comma (",").
- When you specify "\*", it will include all potential columns in a given table.
- When you use "distinct" it eliminates duplicates
- Users are encouraged to specify "\*" to create an inventory of which columns are available in a given table.

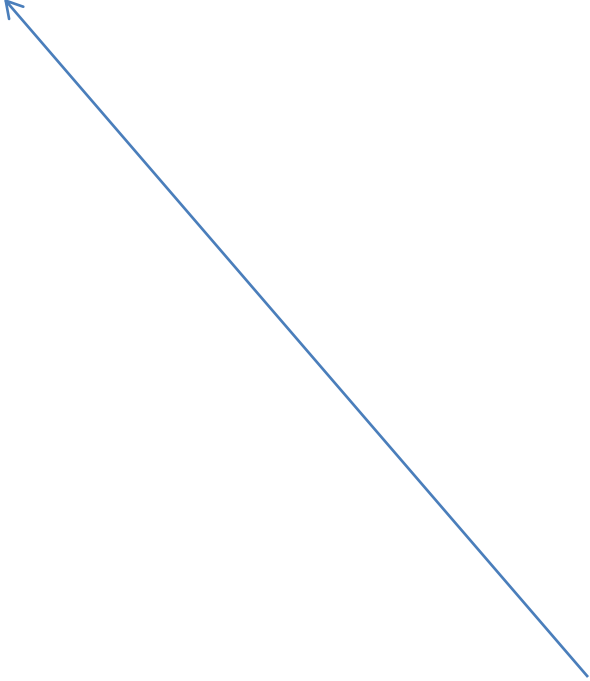
# SUPPORTED FUNCTIONS

## - String functions

- `concat(str1, str2 [, str3...])` ... Returns concatenated string
- `length(str)` ... Returns the length of a string
- `lower(str)` ... Returns the argument in lowercase
- `substr(str, startIndex, length)` ... Return the substring of a string
- `upper(str)` ... Returns the argument in uppercase
- `instr(str, searchstr, [,position [, occurrence]])` ... Returns the start position of searchstr within str.  
Returns 0 if searchstr is not found.

# Expression

- You can specify
  - Column names
  - Operators
    - (e.g. +, -, \*, /)
  - Functions,
  - Number/string literals



```
select [distinct] <expression>[,  
<expression>...] from <table_name> [,  
<table_name>...]  
[where <condition>]  
[group by <expression> [, <expression>...]]  
[having <condition>]  
[order by <expression> [asc | desc],  
[<expression> [asc | desc]]..]
```



# Aliases: Column Names

- You can specify an alias column name using the "as" keyword.
  - Example:
    - `select value as v from fact where v > 0`
- This is especially helpful when you wish to apply multiple conditions against the same column.
- You do lose the more specific identification of the field in the views

Query Table

Instance Table Query Table Dimension Table

Query Statement

```
select value as v from fact where v < 0
```

|   | v      |
|---|--------|
| 1 | -2000  |
| 2 | -1000  |
| 3 | -1000  |
| 4 | -84000 |
| 5 | -1000  |
| 6 | -49000 |
| 7 | -9000  |
| 8 | -25000 |

# From

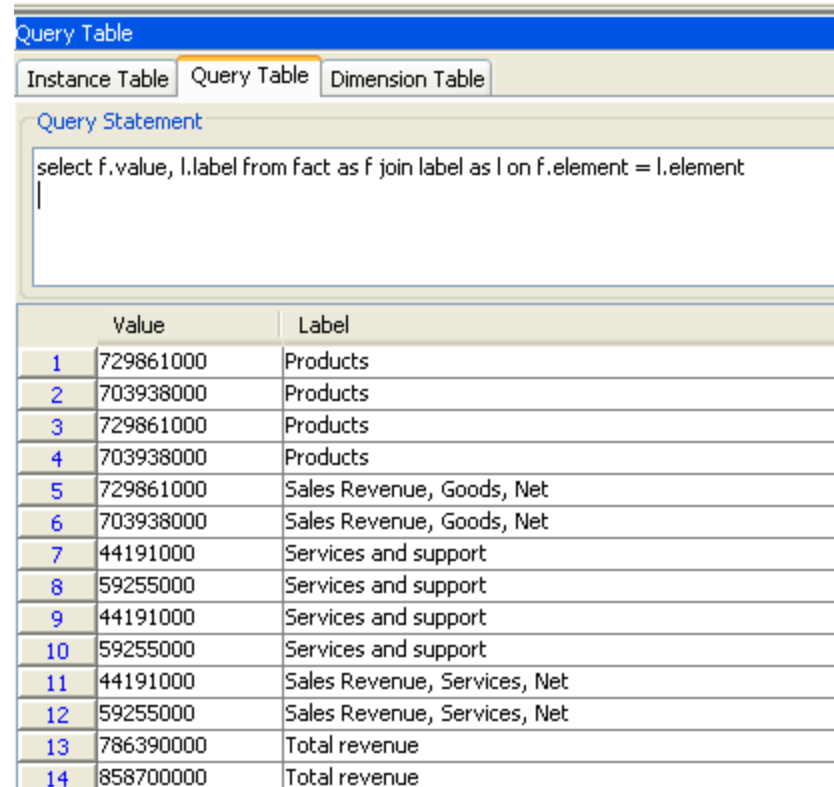
- Sources
  - Tables
  - Views/other queries

# From: Table

| table name             | data content                 | note                 |
|------------------------|------------------------------|----------------------|
| file                   | files in DTS/instance        |                      |
| element                | element list                 | -                    |
| presentation           | presentation link            | Resolved links only  |
| calculation            | calculation link             | Resolved links only. |
| definition             | definition link              | Resolved links only. |
| label                  | label link                   | Resolved links only. |
| reference              | reference link               | Resolved links only. |
| unresolvedpresentation | presentation link            | All links.           |
| unresolvedcalculation  | calculation link             | All links.           |
| unresolveddefinition   | definition link              | All links.           |
| unresolvedlabel        | label link                   | All links.           |
| unresolvedreference    | reference link               | All links.           |
| roletype               | role type                    |                      |
| arcroletype            | arcrole type                 |                      |
| context                | context                      | Instance only        |
| unit                   | unit                         | Instance only        |
| fact                   | item                         | Instance only        |
| footnote               | footnote link                | Instance only        |
| calculationerror       | facts with calculation error | Instance only        |

# Alias for Table Names

- You can specify an alias *table* name using the "as" keyword as follows:
  - `select f.value, l.label`  
`from fact as f join`  
`label as l on f.element`  
`= l.element`



The screenshot shows a software interface for a Query Table. At the top, there are three tabs: "Instance Table", "Query Table" (which is selected), and "Dimension Table". Below the tabs is a text area labeled "Query Statement" containing the SQL query: `select f.value, l.label from fact as f join label as l on f.element = l.element`. Below the query statement is a table with two columns: "Value" and "Label". The table contains 14 rows of data, with the first four rows highlighted in yellow.

|    | Value     | Label                        |
|----|-----------|------------------------------|
| 1  | 729861000 | Products                     |
| 2  | 703938000 | Products                     |
| 3  | 729861000 | Products                     |
| 4  | 703938000 | Products                     |
| 5  | 729861000 | Sales Revenue, Goods, Net    |
| 6  | 703938000 | Sales Revenue, Goods, Net    |
| 7  | 44191000  | Services and support         |
| 8  | 59255000  | Services and support         |
| 9  | 44191000  | Services and support         |
| 10 | 59255000  | Services and support         |
| 11 | 44191000  | Sales Revenue, Services, Net |
| 12 | 59255000  | Sales Revenue, Services, Net |
| 13 | 786390000 | Total revenue                |
| 14 | 858700000 | Total revenue                |

# Expression: Example

`select value / 1000000,  
length(elementname),  
100, "foo" from fact`

Returns as columns

- 1. The “value” divided by 1,000,000
- 2. The length in characters of the element name,
- 3. The number “100”
- 4. The word “foo”

Query Table

Instance Table Query Table Dimension Table

Query Statement

```
select value/1000000, length(element), 100, "foo" from fact
```

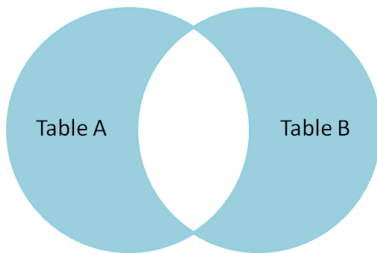
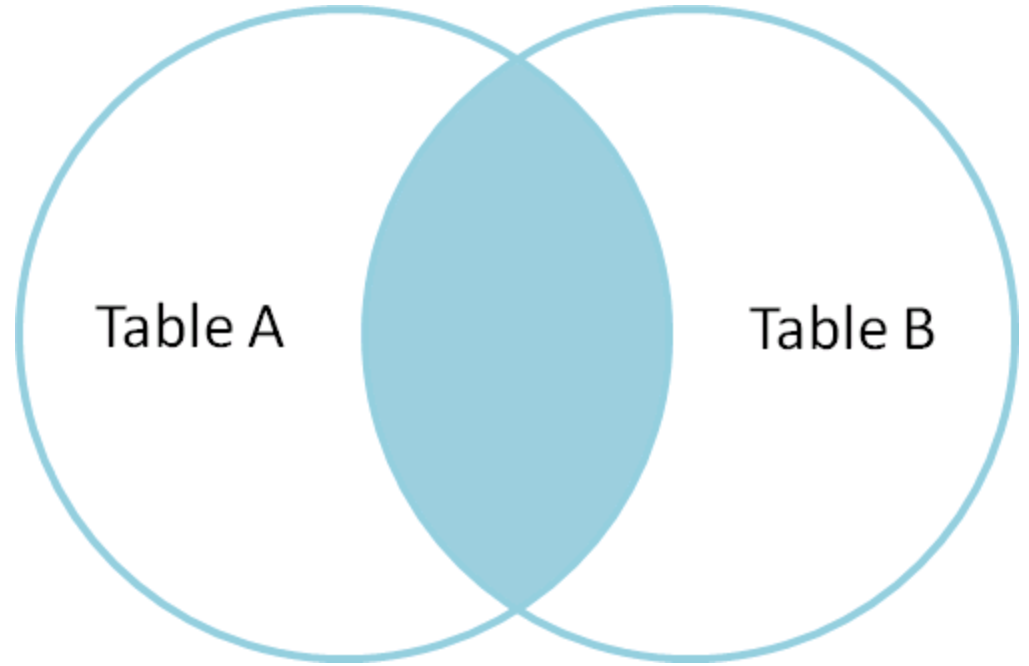
|    | value / 1000000 | length(element) ▾ | 100 | "foo" |
|----|-----------------|-------------------|-----|-------|
| 1  | 20.476          | 98                | 100 | foo   |
| 2  |                 | 97                | 100 | foo   |
| 3  | 18.8            | 97                | 100 | foo   |
| 4  | 254.7           | 97                | 100 | foo   |
| 5  |                 | 96                | 100 | foo   |
| 6  | 20.476          | 96                | 100 | foo   |
| 7  | 16.22           | 95                | 100 | foo   |
| 8  | 5.041           | 95                | 100 | foo   |
| 9  | 11.179          | 95                | 100 | foo   |
| 10 | 0               | 95                | 100 | foo   |
| 11 | 0               | 95                | 100 | foo   |
| 12 | 127.138         | 95                | 100 | foo   |
| 13 | 4.774           | 95                | 100 | foo   |
| 14 | 5.033           | 95                | 100 | foo   |
| 15 | 44.49           | 95                | 100 | foo   |
| 16 | 72.841          | 95                | 100 | foo   |
| 17 | 1.589           | 93                | 100 | foo   |

# From Clause

- Inner Join
  - You can use inner join by following syntax:
    - <table name1> join <table name2> [on <condition>]
- Joining multiple tables
  - If you want to join more than two tables, use parenthesis as follows:
    - (<table name1> join <table name2> [on <condition>])  
join <table name3> [on <condition>]

# Joins

Inner Join



Outer Join

# Subqueries in From Clause

- Example:
  - select element from (select \* from fact where value < 0)



# Where Clause

- In a "where" clause, you can specify
  - the formula of "<expression> <comparison\_operator> <expression>" and
  - "not", "and", and "or".
- The <comparison\_operator> is any one of
  - {"=", "!=", "<", ">", "<=", ">="}.
- You can use "in" conditional operator by following syntax:
  - <column name> "in" "(" <select statement> ")"
    - Return true when selected column value exists In the result of select statement following to "in".

# Where Clause

- "" for exact match
- You can use a regular expression (embraced with slash (/) or backslash(\)) with the operator "=" or "!=".
  - Used for “contains” rather than “equals”
  - // for contains match
- //i for case insensitive match
  - The lower() function (e.g., “select element as v,value from fact where lower(v) = /cash/”) can serve similar purpose
- /^/ for starts with match
- /\$/ for ends with match

# Where Clause

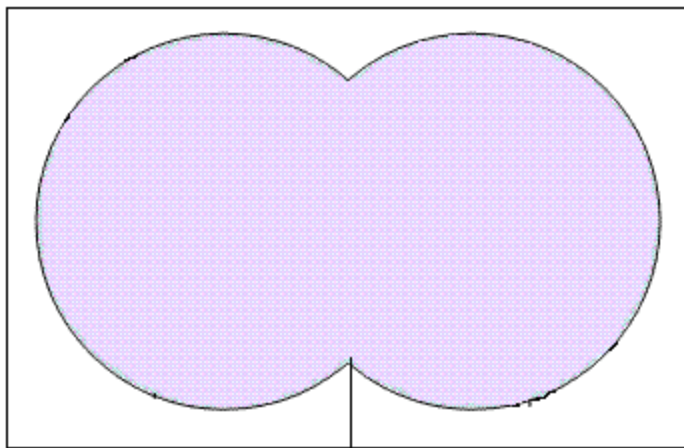
- **Use the decimal point "." as a wild card for any *single* character.**
  - /t.x/i will return anything with text or tax in it.
- **Use the decimal point/asterisk combo ".\*" as a wild card for any *grouping* of characters.**
  - select \* from element where element = /oil.\*gas/i will match anything with oil followed by gas (ignoring case)
- **Use [abc] to match a *single* character as either a, b or c**
  - /t[ae]x/i will likewise return text or tax
- **/\d/ will match any *single* digit**
  - select \* from element where element = /\d/ will return all elements with a numeric digit in them.

# Order By Clause

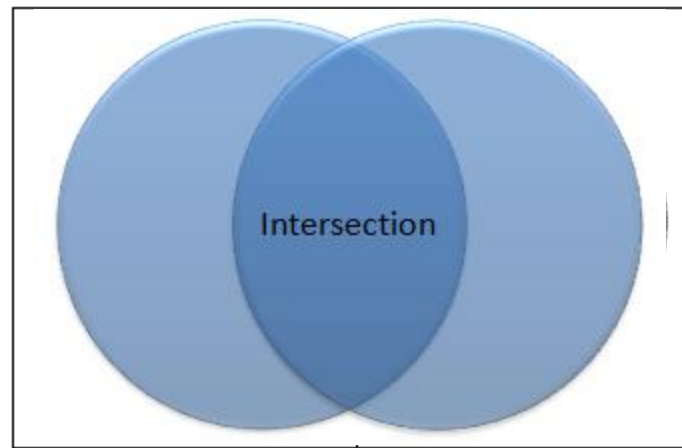
- In a "order by" clause, you should specify a expression to be sorted accompanied by either "asc" or "desc".
  - Ascending: When you want to sort by ascending order, specify a column name and "asc" (or leave it blank; it's the default value)
  - Descending: When you want to sort by descending order, specify a column name and "desc"

# Behavior of Items in Query Table

- The query results in Xwand are good tools for finding information elsewhere in XWand
  - When "element" is included in <table\_name> and when you double-click a row of the table, the corresponding element declaration will be selected in other panes in the application.
  - Similarly, when "Value" column is included, the corresponding item of that row will be selected in other panes.

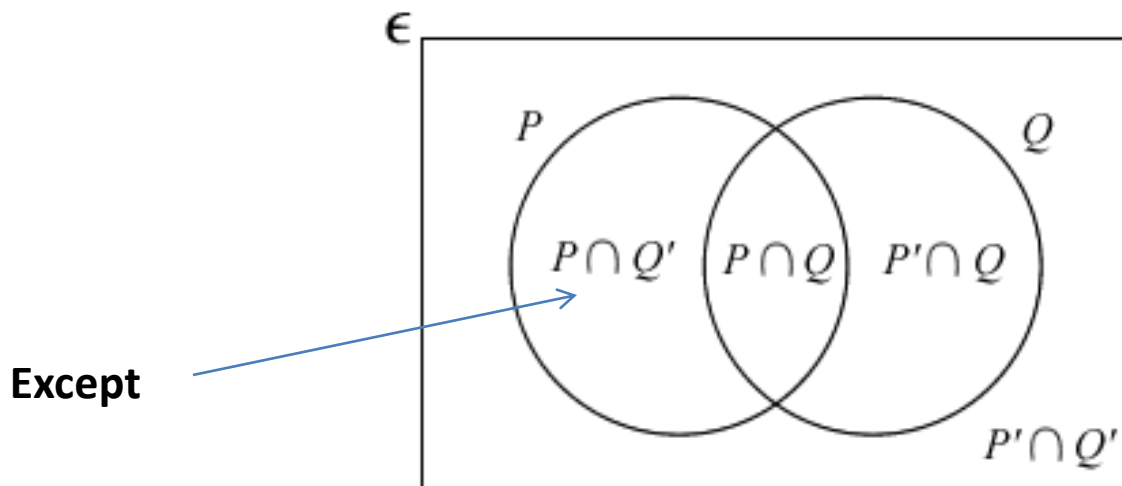


Union



Intersection

- **Union** does not retain duplicates
- **Union all** retains duplicates



# Union, Union All, Except, Intersect

- You can use
  - union
  - union all
  - except
  - intersect operators.
- In the syntax
  - `<select statement> <"union" ["all"] | "except" | "intersect"> <select statement>`
- Both select statements must specify the same columns

# Supported Functions: String Functions

- `concat(str1, str2 [, str3...])`
  - Returns concatenated string
- `length(str)`
  - Returns the length of a string
- `lower(str)`
  - Returns the argument in lowercase
- `upper(str)`
  - Returns the argument in uppercase

Query Statement

```
select length(element), substr(element, 3, 5) as v, lower(v), upper(v), concat(v,v) from fact
```

|    | length(element) | v     | lower(v) | upper(v) | concat(v, v) |
|----|-----------------|-------|----------|----------|--------------|
| 1  | 46              | -gaap | -gaap    | -GAAP    | -gaap-gaap   |
| 2  | 46              | -gaap | -gaap    | -GAAP    | -gaap-gaap   |
| 3  | 55              | be:Ca | be:ca    | BE:CA    | be:Cabe:Ca   |
| 4  | 37              | -gaap | -gaap    | -GAAP    | -gaap-gaap   |
| 5  | 68              | -gaap | -gaap    | -GAAP    | -gaap-gaap   |
| 6  | 54              | -gaap | -gaap    | -GAAP    | -gaap-gaap   |
| 7  | 35              | be:Ot | be:ot    | BE:OT    | be:Otbe:Ot   |
| 8  | 29              | be:Ac | be:ac    | BE:AC    | be:Acbe:Ac   |
| 9  | 36              | -gaap | -gaap    | -GAAP    | -gaap-gaap   |
| 10 | 71              | -gaap | -gaap    | -GAAP    | -gaap-gaap   |
| 11 | 34              | be:Re | be:re    | BE:RE    | be:Rebe:Re   |



# Eric's Favorite String Functions

- `substr(str, startIndex, length)`
  - Return the substring of a string
- `instr(st, search char, startIndex)`
  - Returns the location of a specific character/word in a string.

# One Possible Solution

- Select
  - segment as a,
  - `instr(a, ">", 1) + 1` as start,
  - `instr(a, "</", start)` as end,
  - `substr(a, start, end - start)` as textvalue
- From fact
- Where segment  $\neq ''$

# Supported Functions: Numeric functions

- Acts on individual amounts
- `abs(num)`
  - Returns the absolute value
- `ceil(num)`
  - Returns the smallest integer value not less than the argument
- `floor(num)`
  - Returns the largest integer value not greater than the argument
- `round(num)`
  - Rounds the argument

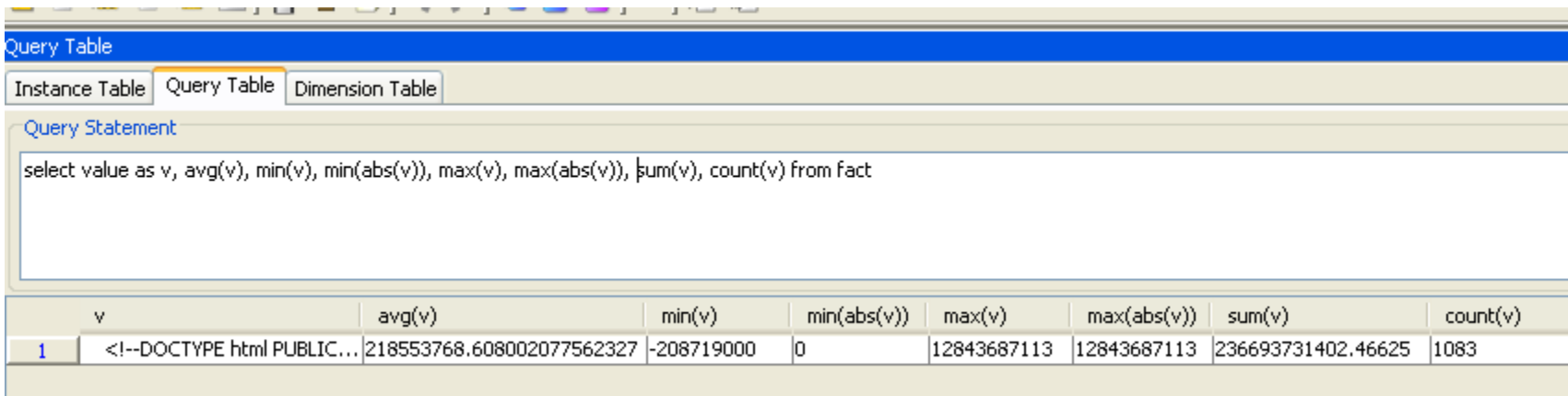
Query Statement

```
select value as v, abs(v), ceil(v), floor(v), round(v) from fact|
```

|    | v        | abs(v)   | ceil(v)  | floor(v) | round(v) |
|----|----------|----------|----------|----------|----------|
| 54 | 0.00475  | 0.00475  | 1        | 0        | 0        |
| 55 | 0.002    | 0.002    | 1        | 0        | 0        |
| 56 | 37793000 | 37793000 | 37793000 | 37793000 | 37793000 |
| 57 | 19773000 | 19773000 | 19773000 | 19773000 | 19773000 |
| 58 | 6921000  | 6921000  | 6921000  | 6921000  | 6921000  |
| 59 | 6104000  | 6104000  | 6104000  | 6104000  | 6104000  |
| 60 | 1022000  | 1022000  | 1022000  | 1022000  | 1022000  |
| 61 | 84313000 | 84313000 | 84313000 | 84313000 | 84313000 |
| 62 | 81299000 | 81299000 | 81299000 | 81299000 | 81299000 |
| 63 | 700000   | 700000   | 700000   | 700000   | 700000   |
| 64 | 95000000 | 95000000 | 95000000 | 95000000 | 95000000 |
| 65 | -2000    | 2000     | -2000    | -2000    | -2000    |
| 66 | -1000    | 1000     | -1000    | -1000    | -1000    |
| 67 | 0        | 0        | 0        | 0        | 0        |
| 68 | -1000    | 1000     | -1000    | -1000    | -1000    |
| 69 | 0        | 0        | 0        | 0        | 0        |
| 70 | -84000   | 84000    | -84000   | -84000   | -84000   |
| 71 | -1000    | 1000     | -1000    | -1000    | -1000    |
| 72 | -49000   | 49000    | -49000   | -49000   | -49000   |
| 73 | -9000    | 9000     | -9000    | -9000    | -9000    |
| 74 | -25000   | 25000    | -25000   | -25000   | -25000   |
| 75 | 20.98    | 20.98    | 21       | 20       | 21       |
| 76 | 35.13    | 35.13    | 36       | 35       | 35       |
| 77 | 18.1     | 18.1     | 19       | 18       | 18       |
| 78 | 20.2     | 20.2     | 21       | 20       | 20       |
| 79 | 24598000 | 24598000 | 24598000 | 24598000 | 24598000 |

# Supported Functions: Aggregate Functions

- Creates a single result from a group of amounts
  - avg(expr) Returns the average value
  - count("\*" | expr) Returns a count of values
  - max(expr) Returns the maximum value
  - min(expr) Returns the minimum value
  - sum(expr) Returns the sum



The screenshot shows a query tool interface with a 'Query Table' tab selected. The 'Query Statement' field contains the following SQL query:

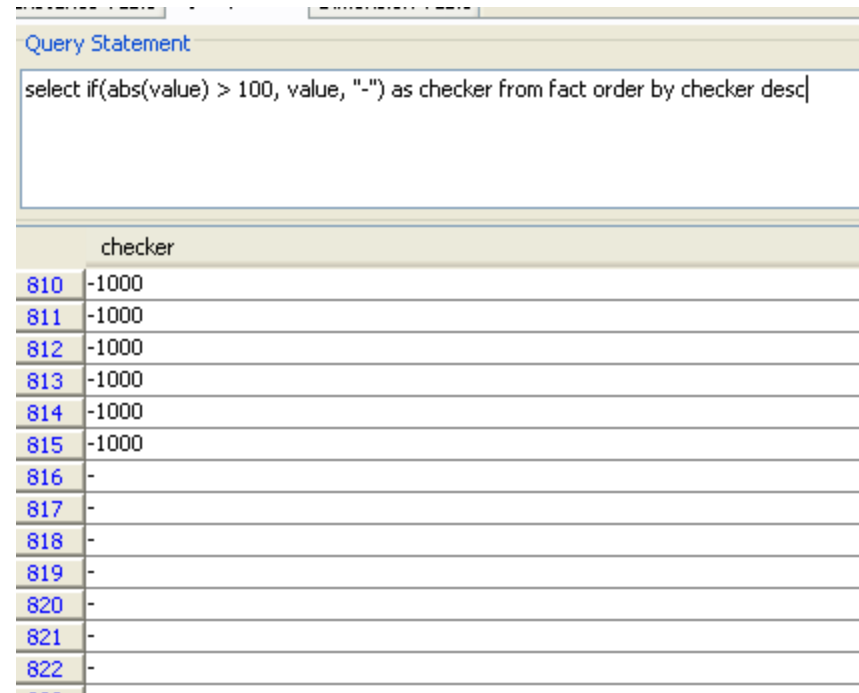
```
select value as v, avg(v), min(v), min(abs(v)), max(v), max(abs(v)), sum(v), count(v) from fact
```

The results are displayed in a table with the following columns and values:

|   | v                          | avg(v)                    | min(v)     | min(abs(v)) | max(v)      | max(abs(v)) | sum(v)             | count(v) |
|---|----------------------------|---------------------------|------------|-------------|-------------|-------------|--------------------|----------|
| 1 | <!--DOCTYPE html PUBLIC... | 218553768.608002077562327 | -208719000 | 0           | 12843687113 | 12843687113 | 236693731402.46625 | 1083     |

# XBRL functions

- Returning a specific label
  - `xbrl_label(element, [lang [, labelRole [, extendedLinkRole]]])`
    - Returns the label of the element. Note: The argument "element" must be the column "element" or "parentelement"
- Other functions
  - `if(condition, expr1, expr2)`
    - If condition is true, returns `expr1`. Otherwise returns `expr2`.



The screenshot shows a software interface with a 'Query Statement' field containing the following SQL query: `select if(abs(value) > 100, value, "-") as checker from fact order by checker desc`. Below the query field is a table with a single column header 'checker'. The table contains 13 rows of data, with the first five rows showing values '-1000' and the remaining eight rows showing empty cells. The rows are numbered 810 through 822 on the left side.

|     | checker |
|-----|---------|
| 810 | -1000   |
| 811 | -1000   |
| 812 | -1000   |
| 813 | -1000   |
| 814 | -1000   |
| 815 | -1000   |
| 816 |         |
| 817 |         |
| 818 |         |
| 819 |         |
| 820 |         |
| 821 |         |
| 822 |         |

# Sample Queries

- List all content from a taxonomy
  - select \* from element
    - For all the columns, gets all the element declarations from the table.
- List all unique start date and end date pairs
  - select distinct StartDate, EndDate from context
    - Gets all the start dates and end dates from the context table, duplication being removed.
- Select all concepts with a label of (exactly) “Sales”
  - select Value from fact where ElementLabel = 'Sales'
    - Returns value of an item whose element label matches 'Sales'.
    - (Label language etc. are in synchronization with the application display.)

# Sample Queries (Cont.)

- Select all concepts with a label of “Sales” and Value < 0
  - select ElementLabel, Value from fact where ElementLabel = /Sales/ and Value < 0
    - Returns labels and values, where 'Sales' is contained (whole or part) in the element label and the value is less than 0.
- Select all elements which are not included in the presentation
  - select element, elementlabel, value from fact where element in (select element from fact except select element from presentation)
    - Returns element names, element labels, and values of facts that will not be in the presentation link.
- Select elements and related labels from all concepts included in both instance and label linkbase
  - select element, label, value from fact join label on fact.element = label.element
  - This will join fact table and label link table with element column.

# Sample Queries (Cont.)

- Note that joins are faster than non-joins
  - `select element, label, value from fact, label where fact.element = label.element`
  - Same as above, but this query is slower than the above.
- Identify if labels are duplicated
  - `select label, count(*) from label group by label having count(*) > 1`
  - Returns the count of each label that appears more than once.



# Using the Viewer's Data Query tools (cont.)

| Query helpers                             |   |   |
|---|---|---|
| \$  | Search from end of the field (rightmost character match)      | select element, value from fact where elementName = /value\$/i  |
| ^   | Search from beginning of the field (leftmost character match) | select element, value from fact where elementName = /^cash/i  |
| .   | Wildcard a single character                                   | Where element = /t.x/i returns tax, tex, etc.   |
| .*  | Wildcare for any number of characters                         | select * from element where element = /oil.*gas/i will match anything with oil followed by gas (ignoring  |
| i   | Ignore case   | Where element = /member/i   |
| [abc]                                     | Match any of the enclosed characters                          | [abc] to match a single character as either a, b or c<br>/t[ae]x/i will likewise return text or tax   |
| /\d/                                      | \d replaces a single digit                                    | select * from element where element = /\d/ will return all elements with a numeric digit in them.<br>select * from element where element = /20\d\d/ will catch anything that look like a year in the 21st century (four digits, starting with 20; would match 2009, 2010, 2099, etc.) |
| Functions (with special .jar) – see EEC   |   |   |
| Instr(field,"search term", starting char) |   |   |
| Substr(field, starting char, length)      |   | select segment as a, instr(a, ">", 1) +1 as start, instr(a, "</", start) as end, substr(a, start, end - start) as textvalue from fact where segment != ""   |

# Random Thoughts

- What can you do with aerodynamic data?
- What can I uncover that will indicate someone else's future actions or state upon which I can capitalize?
- The enemy of my enemy is my friend.
- The vulture does not look to the healthy animal as its next meal.
- Efficiencies often come at the cost of compromise

# Appendix of Slides to be developed

Internal sources

External sources



Seamless audit trail  
Frictionless data  
Aerodynamic data

How often can you get

- Internal data?
- External data
  - Much larger population
  - Quarterly and annual
    - US
    - FPI