Unleash the power of your Universe

Pierpaolo Vezzosi
Sr. Product Manager, Information Services
2007 Partner Summit

Business Objects™

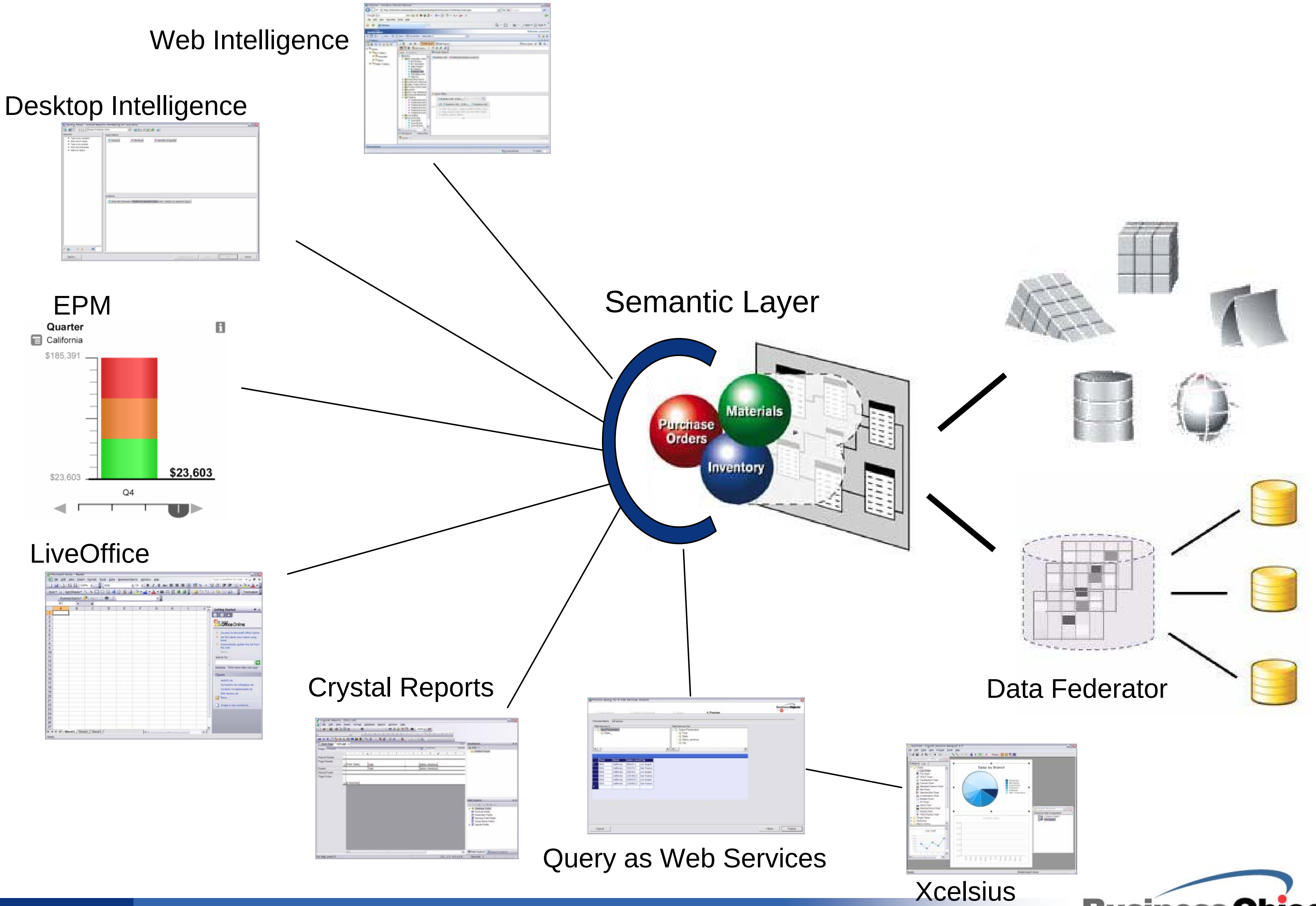
Agenda

- **Clients accessing the universe**
  - The Semantic Layer concept
  - Using data in your applications
- **Universe tips**
  - Measures on multiple fact tables
  - Index awareness
  - Multi-source Universes
- **Semantic Layer roadmap**

Agenda

- **Clients accessing the universe**
  - The Semantic Layer concept
  - Using data in your applications
- **Universe tips**
  - Measures on multiple fact tables
  - Index awareness
  - Multi-source Universes
- **Semantic Layer roadmap**

Semantic Layer: Where can you use it?



Web Intelligence

Desktop Intelligence

EPM

LiveOffice

Crystal Reports

Query as Web Services

Xcelsius

Semantic Layer

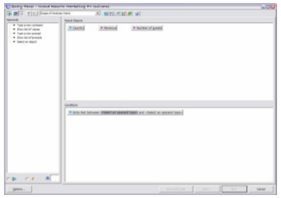
Data Federator

The “Semantic Layer” concept

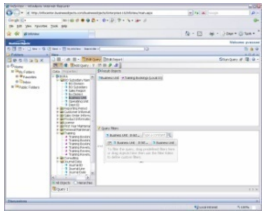
- The Semantic Layer is a set of data and metadata retrieved from the database and from the Universe.
- The Universe is the definition of your Semantic Layer
- The Semantic Layer allows manipulation of data based on the metadata and on the additional business logic in the Universe (e.g. aggregation functions, aggregate awareness)
- The Semantic Layer data and metadata is stored within the document

Tools with Semantic Layer

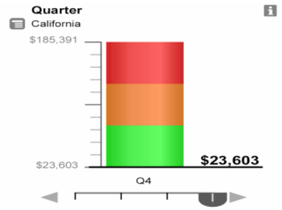
Desktop Intelligence



Web Intelligence



EPM

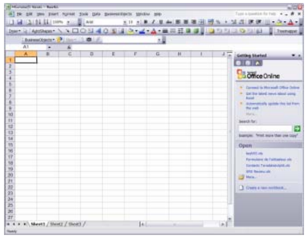


A Semantic Layer allows:

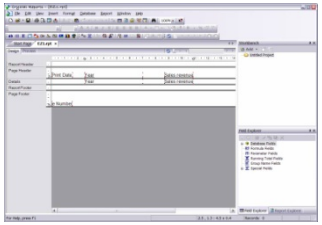
- Reuse of business logic added into the Universe
- On report calculations (when doing filtering, sections, breaks…)
- On report-drill with scope of analysis
- Offline interactive capabilities (Desktop Intelligence, Web Intelligence rich client
- Multiple data provider synchronization
- Optimal for simple, intuitive and not-guided data analysis

Tools without Semantic Layer

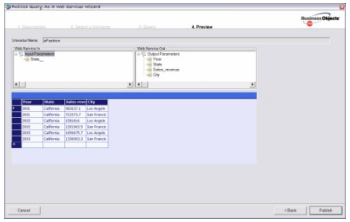
LiveOffice



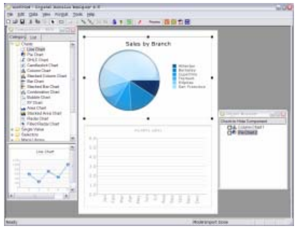
Crystal Reports



Query as a Web Service



Xcelsius



Without a Semantic Layer:

- The tools retrieve only the data you queried with the Query panel
- No metadata or Universe business logic information is retrieved
- A smaller amount of information is transferred
- Retrieved data can be easily reused by the client tool
- Personalized workflows can be executed (the client tool will not force the results to fit with business logic)
- Optimal for pre-canned, smaller reports and for IT fine tuning

Basic Semantic Layer demo

- **Semantic Layer**
  - Desktop Intelligence using Semantic Layer to calculate a Section revenue
  - Crystal Report using manual intervention to create a Summary revenue

Agenda

- **Clients accessing the universe**
  - The Semantic Layer concept
  - Using data in your applications
- **Universe tips**
  - Measures on multiple fact tables
  - Index awareness
  - Multi-source Universes
- **Semantic Layer roadmap**

Semantic Layer and Query as Web Services

- **Create innovative solutions for your customers**
- **Open BI to users not currently using BI**
- **Integrate your customers enterprise workflows with BI**
- **SQL is generated by Semantic Layer Query Technique**
  - SQL is regenerated dynamically each time the report is refreshed
  - Changes in Universe are taken into account immediately
- **Any Web Services client can access the data**
  - Try Xcelsius Workgroup Edition!

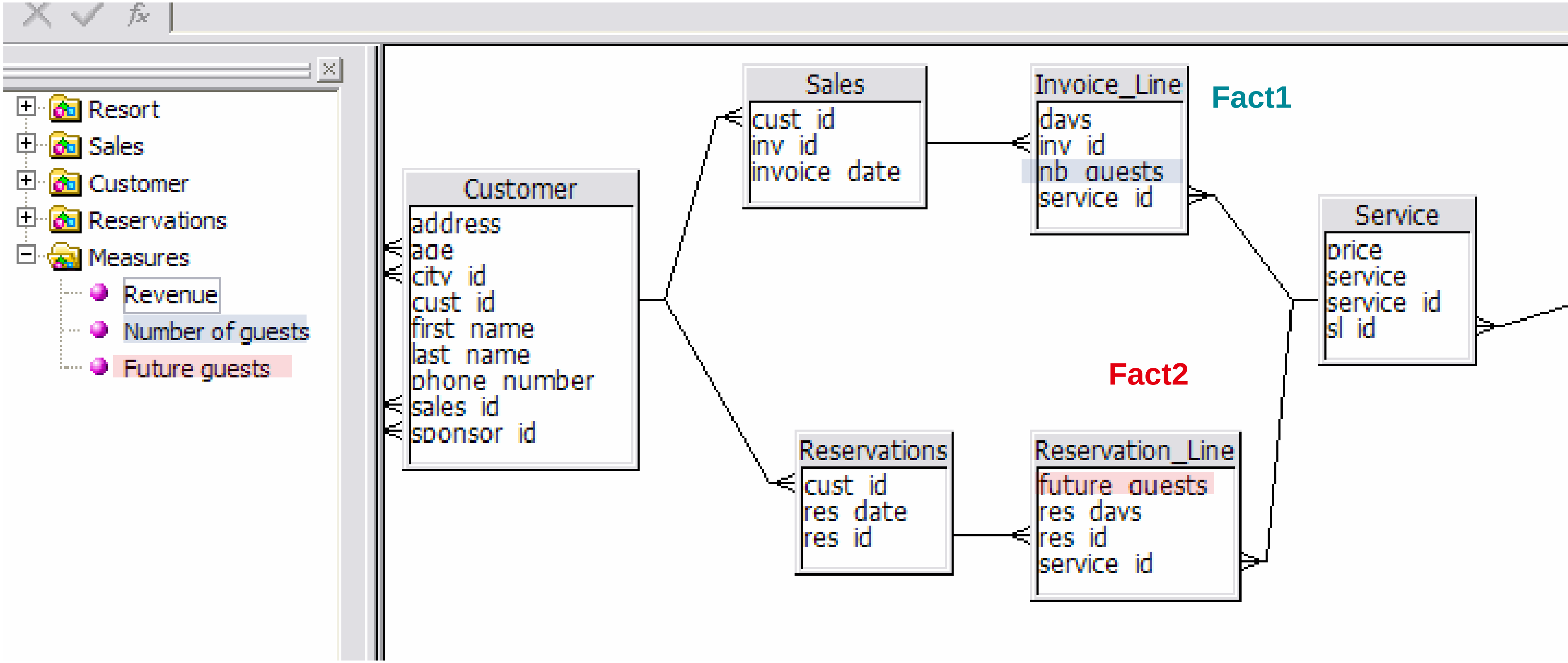
Query as a Web Services demo

- **Retrieving data in your applications**
  - Create a new Web Service
  - Use the Web Service in Microsoft Infopath

Agenda

- **Clients accessing the universe**
  - The Semantic Layer concept
  - Using data in your applications
- **Universe tips**
  - Measures on multiple fact tables
  - Index awareness
  - Multi-source Universes
- **Semantic Layer roadmap**

Measures over Multiple Fact Tables: the problem



It is not possible to create a Universe measure merging Information from Fact1 and Fact2
(e.g. NB_Guests+Future_Guests)

Measures over Multiple Fact Tables: a use case

Use case

- I have my sales fact table and my reservations fact table
- I want to create measures that combine values from both
- But measures from these fact tables are incompatible
- And I don’t want my users to create a report variable to combine them every time

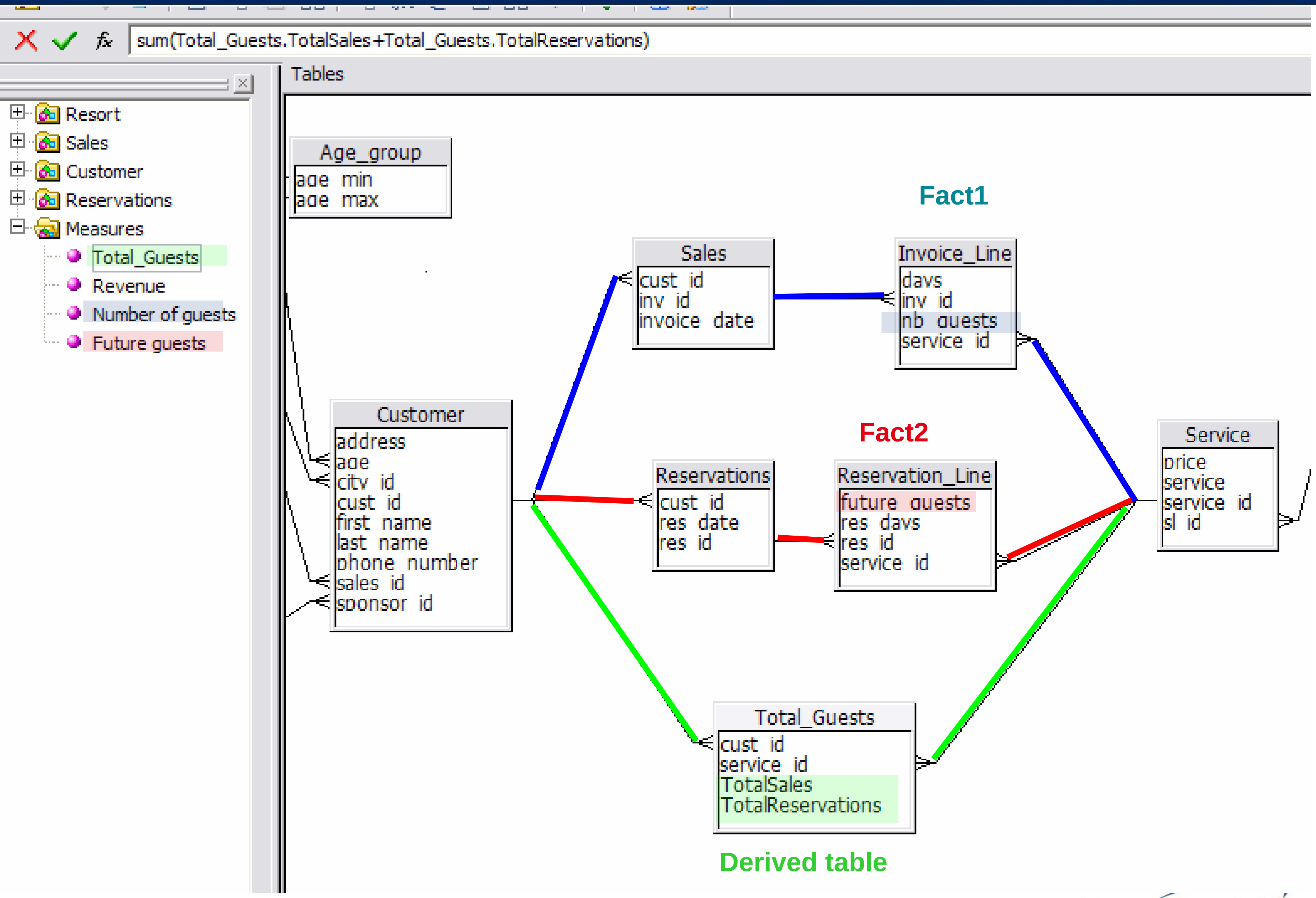
Steps

1. Create a derived table combining both fact tables and common dimension keys
2. Define the new measures in the derived table
3. Join back to dimension tables
4. Define contexts just as for any other fact table
5. Define measure objects on the derived table

Measures over Multiple Fact Tables: solution

- **Define a derived table containing**
  - Customer ID
  - Service ID
  - TotalSales (=sum(NB_Guests) grouped by customer and service)
  - TotalReservations (=sum(future_guests) grouped by customer and service)
- **Join the table to Customer and Service tables**
- **Create a new context by autodetecting it**
- **Create a new measure Total Guests as the sum of TotalSales and TotalReservations**

Measures over Multiple Fact Tables: solution



sum(Total_Guests.TotalSales+Total_Guests.TotalReservations)

Tables

- Resort
- Sales
- Customer
- Reservations
- Measures
  - Total_Guests
  - Revenue
  - Number of guests
  - Future guests

Age_group
- age min
- age max

Sales
- cust id
- inv id
- invoice date

Invoice_Line
- days
- inv id
- nb guests
- service id

Customer
- address
- age
- city id
- cust id
- first name
- last name
- phone number
- sales id
- sponsor id

Reservations
- cust id
- res date
- res id

Reservation_Line
- future guests
- res days
- res id
- service id

Service
- price
- service
- service id
- sl id

Total_Guests
- cust id
- service id
- TotalSales
- TotalReservations

Fact1

Fact2

Derived table

Agenda

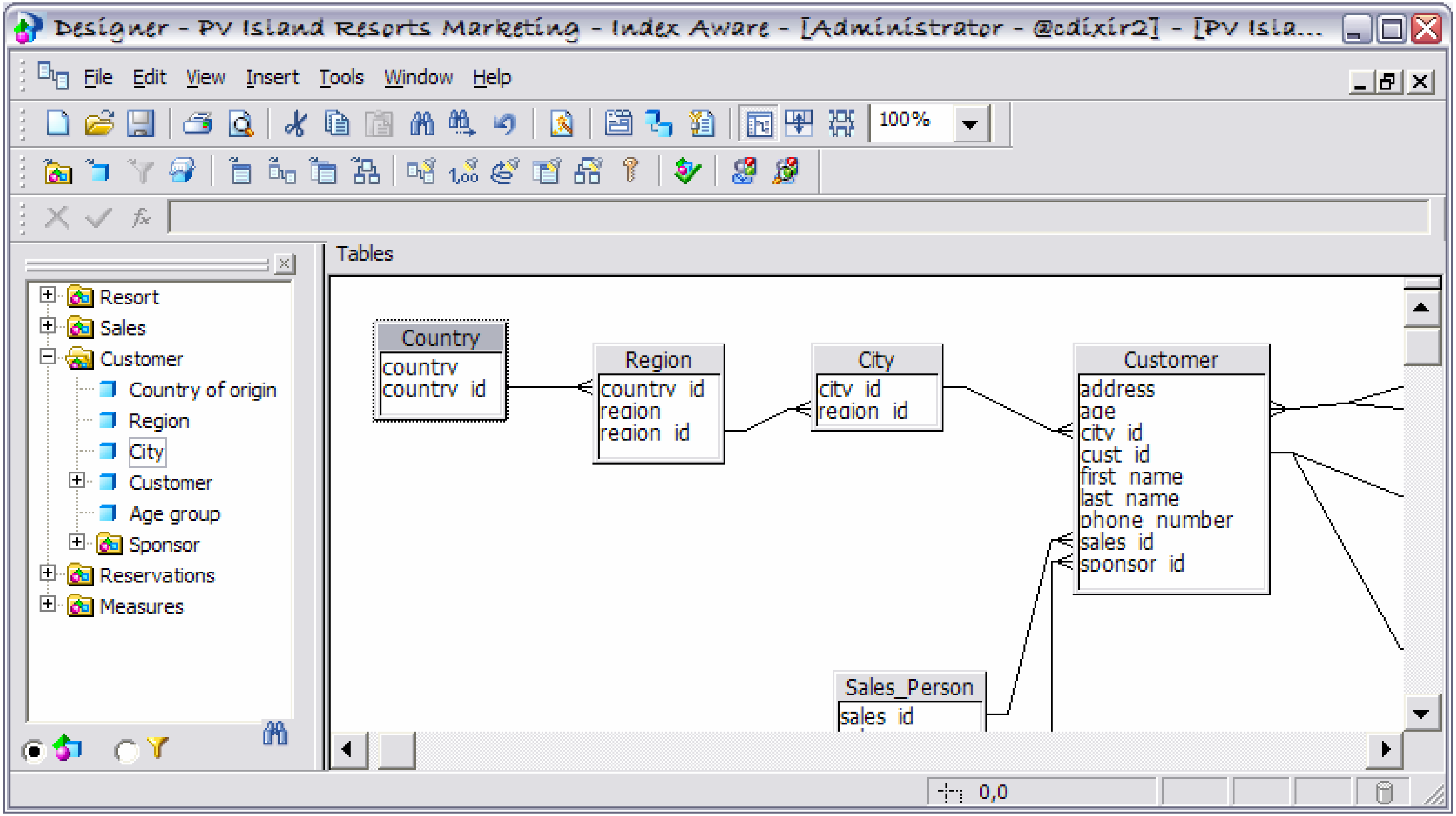
- **Clients accessing the universe**
  - The Semantic Layer concept
  - Using data in your applications
- **Universe tips**
  - Measures on multiple fact tables
  - Index awareness
  - Multi-source Universes
- **Semantic Layer roadmap**

Index Awareness: what is it?

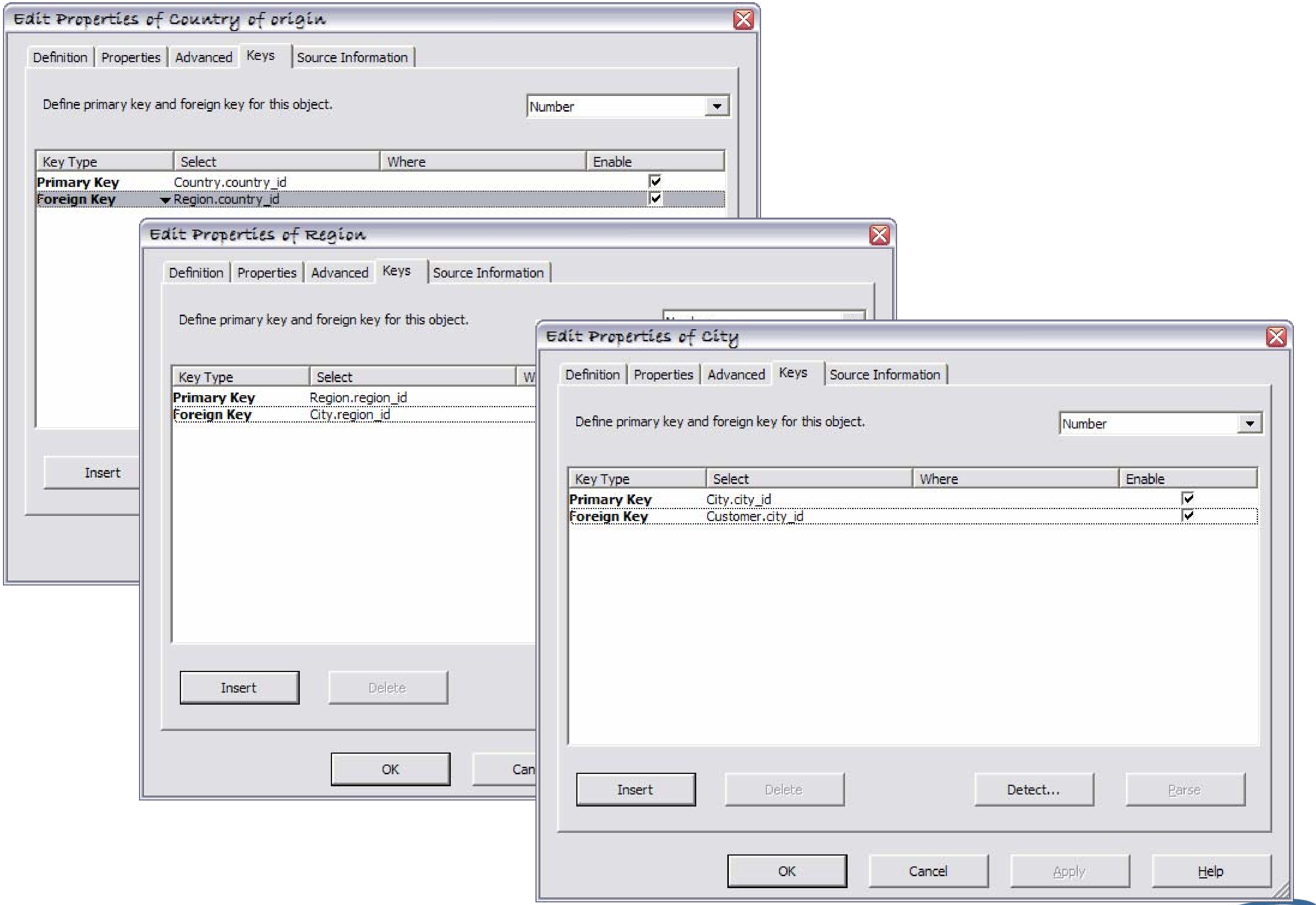
- **Use keys instead of labels whenever possible**
  - E.g. show Country Name but query with country_id

- **Benefits of Index Awareness: performance and uniqueness**
  - Reduces the number of Joins in the query
  - Reduces the number of tables joined
  - Parses keys instead of labels
  - Avoids issues with duplicate labels

Index Awareness: hierarchy sample



Index Awareness: enabling objects



Edit Properties of Country of origin

Define primary key and foreign key for this object.

Number

Key Type	Select	Where	Enable
Primary Key	Country.country_id		☑
Foreign Key	Region.country_id		☑

Edit Properties of Region

Define primary key and foreign key for this object.

Key Type	Select
Primary Key	Region.region_id
Foreign Key	City.region_id

Edit Properties of City

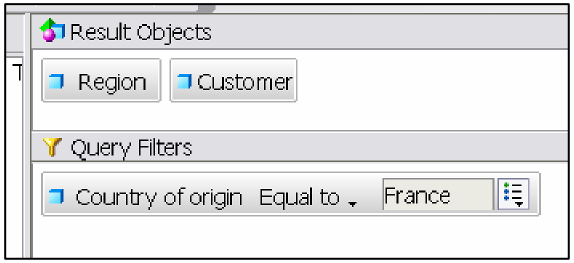
Define primary key and foreign key for this object.

Number

Key Type	Select	Where	Enable
Primary Key	City.city_id		☑
Foreign Key	Customer.city_id		☑

Index Awareness benefits: performance

Without



Result Objects

Region | Customer

Query Filters

Country of origin Equal to France

With

```
SELECT
  Region.region,
  Customer.last_name
FROM
  Country INNER JOIN Region ON
(Country.country_id=Region.country_id)
   INNER JOIN City ON
(City.region_id=Region.region_id)
   INNER JOIN Customer ON
(City.city_id=Customer.city_id)

WHERE
  Country.country  =  'France‘
```

```
SELECT
  Region.region,
  Customer.last_name
FROM
  Customer INNER JOIN City ON
(City.city_id=Customer.city_id)
   INNER JOIN Region ON
(City.region_id=Region.region_id)

WHERE
  Region.country_id  =  2
```

4 tables
3 joins
Where on label

3 tables
2 joins
Where on key

Index Awareness benefits: uniqueness

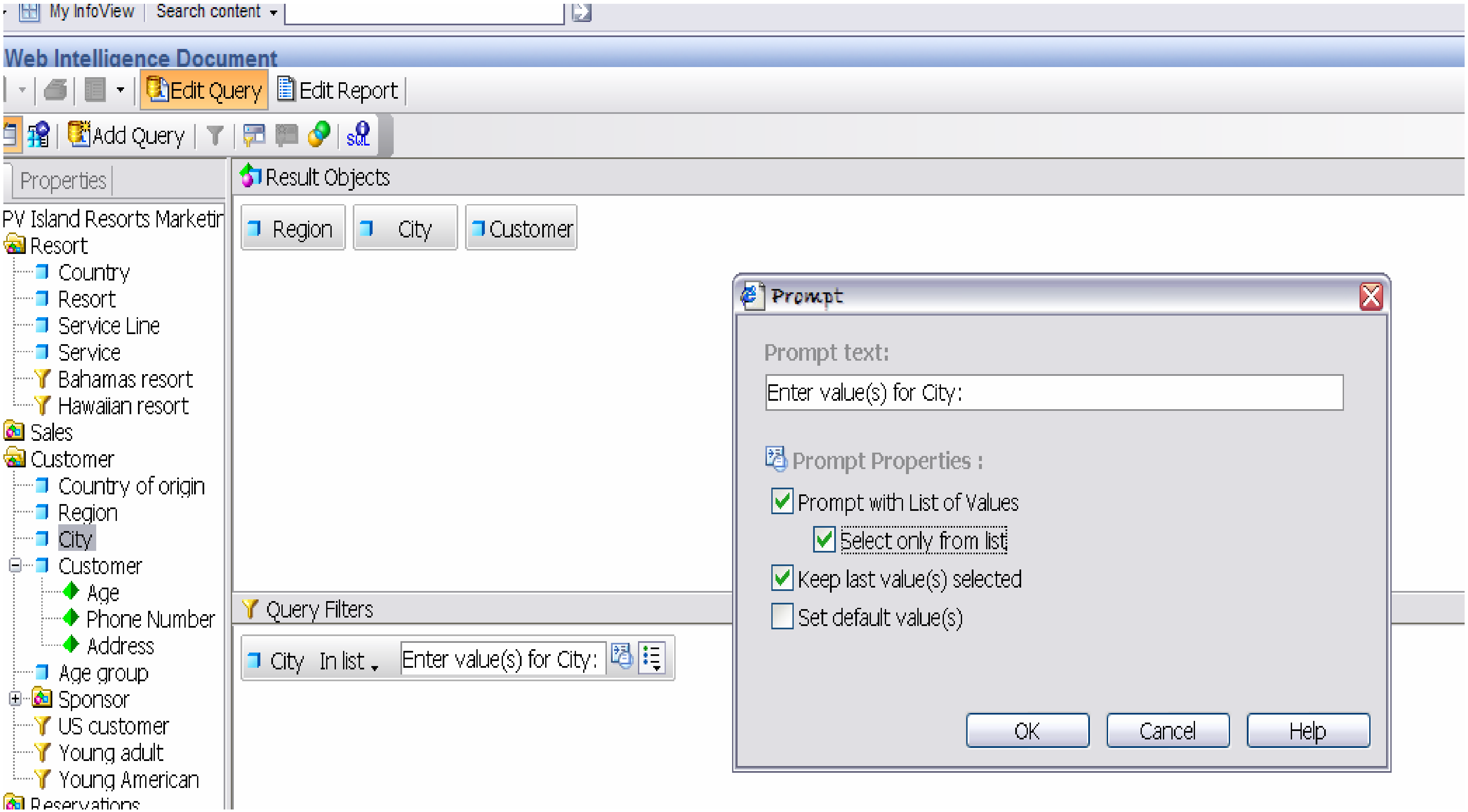
Use case

- I have a universe cascading prompt composed of country, region, and city
- London, Wisconsin and London, UK have the same city name
- How do I make sure my users pick just one?

Steps

1. Make sure that the universe dimensions (country, region, city) have their index defined
2. Create your cascading list of value using the wizard in Designer
3. Use in Web Intelligence and notice that the city name is replaced by City_Id in the query
4. Make sure that prompt values are constrained in your query

Index Awareness benefits: uniqueness



City In list Enter value(s) for City:

Prompt text:

Enter value(s) for City:

Prompt Properties :

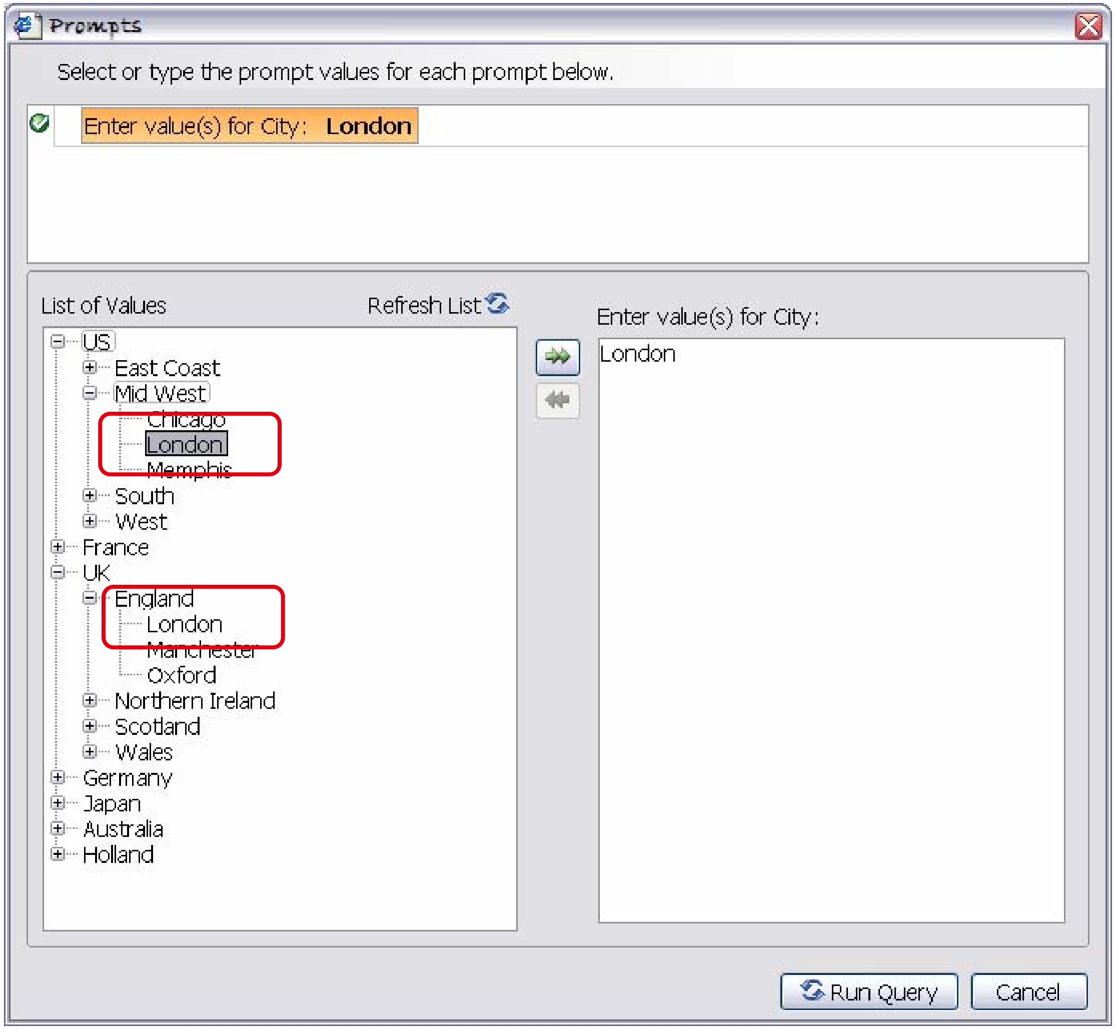
☑ Prompt with List of Values

☑ Select only from list

☑ Keep last value(s) selected

☐ Set default value(s)

Index Awareness benefits: uniqueness



Select or type the prompt values for each prompt below.

Enter value(s) for City: London

List of Values

- US
  - East Coast
  - Mid West
    - Chicago
    - London
    - Memphis
  - South
  - West
- France
- UK
  - England
    - London
    - Manchester
    - Oxford
  - Northern Ireland
  - Scotland
  - Wales
- Germany
- Japan
- Australia
- Holland

Enter value(s) for City:

London

Index Awareness benefits: uniqueness

Without Index awareness

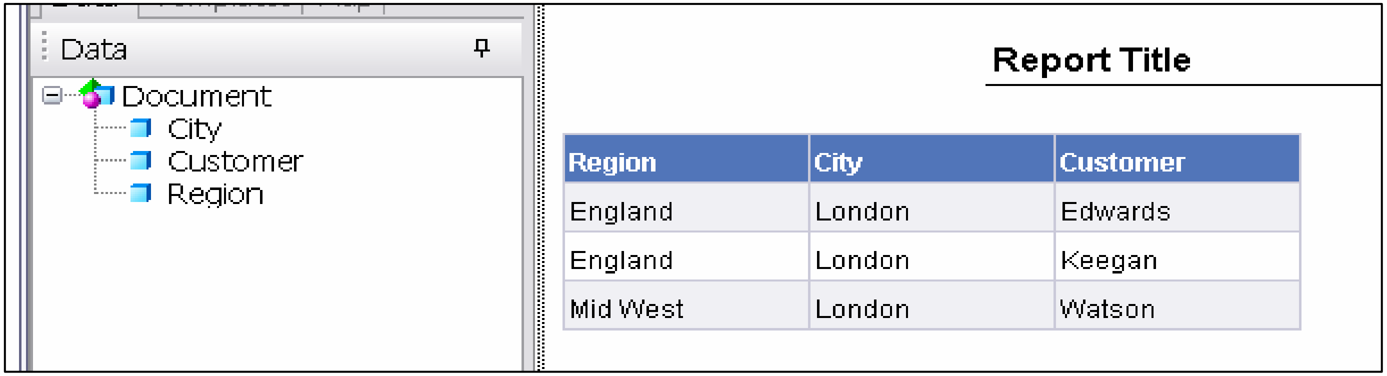
```
SELECT
  Region.region,
  City.city,
  Customer.last_name
FROM
  Region,
  City,
  Customer
WHERE
  ( City.city_id=Customer.city_id  )
  AND  ( City.region_id=Region.region_id  )
  AND
  City.city  In  @prompt('Enter value(s) for
City:','A','Customer\City',Multi,Constrained,
Persistent,,User:0)
```

With Index awareness

```
SELECT
  Region.region,
  City.city,
  Customer.last_name
FROM
  Region,
  City,
  Customer
WHERE
  ( City.city_id=Customer.city_id  )
  AND  ( City.region_id=Region.region_id  )
  AND
  Customer.city_id  In  @prompt('Enter
value(s) for
City:','A','Customer\City',Multi,Constraine
d_key,Persistent,,User:0,KEY_'N')
```

Index Awareness benefits: uniqueness

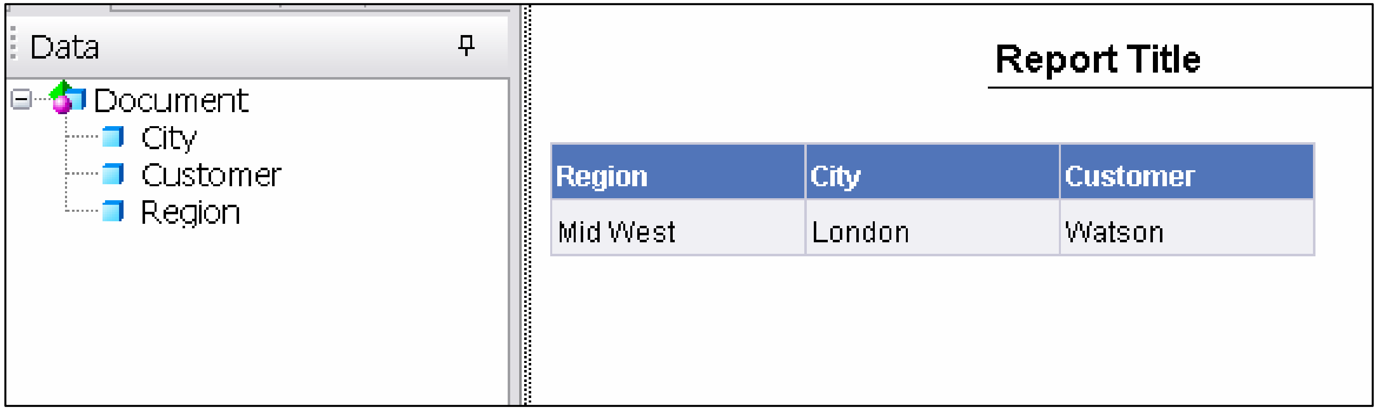
Without index awareness



Report Title

Region	City	Customer
England	London	Edwards
England	London	Keegan
Mid West	London	Watson

With index awareness



Report Title

Region	City	Customer
Mid West	London	Watson

Agenda

- **Clients accessing the universe**
  - The Semantic Layer concept
  - Using data in your applications
- **Universe tips**
  - Measures on multiple fact tables
  - Index awareness
  - Multi-source Universes
- **Semantic Layer roadmap**

Multi-source Universes using Data Federator

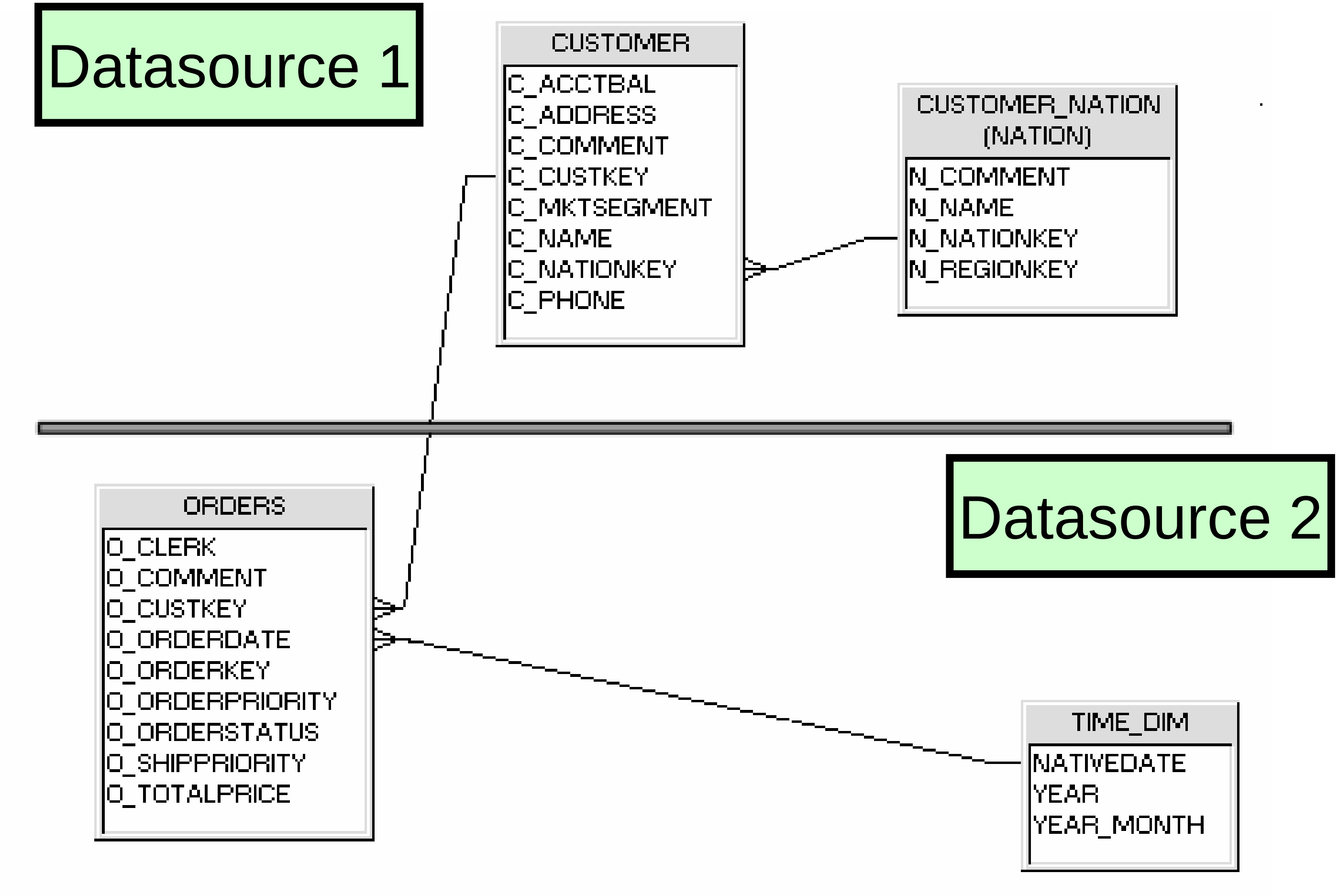
Use case

- My Business Intelligence deployment spans across multiple geographies
- Each geography has its own database instance
- I need to create ad-hoc queries across all of them

Steps

1. Use Data Federator
2. Set up source and target tables
3. Create a universe on top of the Data Federator target tables
4. Use Web Intelligence for ad-hoc reporting

Multi-source Universes example: schemas



Datasource 1

CUSTOMER
C_ACCTBAL
C_ADDRESS
C_COMMENT
C_CUSTKEY
C_MKTSEGMENT
C_NAME
C_NATIONKEY
C_PHONE

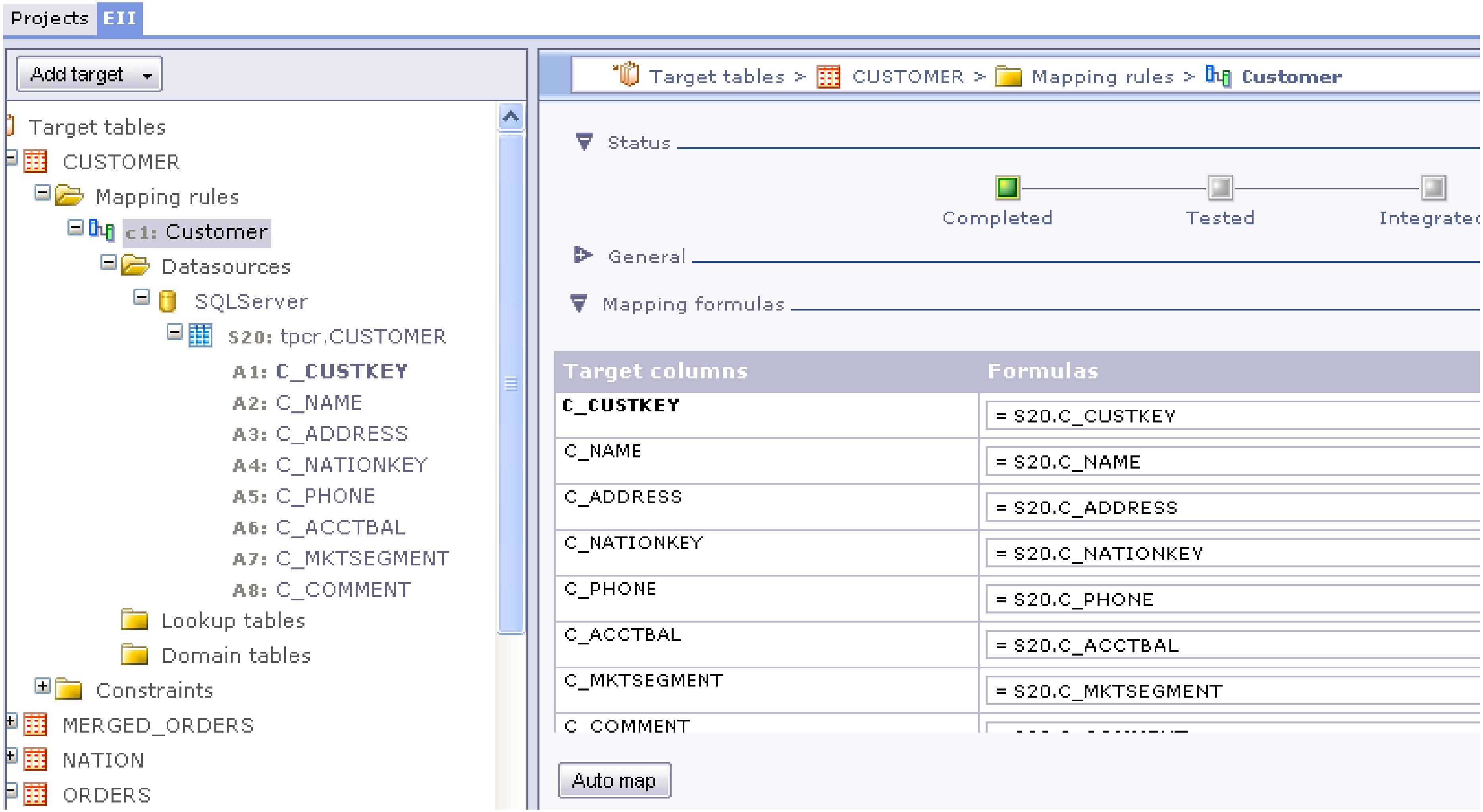
CUSTOMER_NATION (NATION)
N_COMMENT
N_NAME
N_NATIONKEY
N_REGIONKEY

Datasource 2

ORDERS
O_CLERK
O_COMMENT
O_CUSTKEY
O_ORDERDATE
O_ORDERKEY
O_ORDERPRIORITY
O_ORDERSTATUS
O_SHIPPRIORITY
O_TOTALPRICE

TIME_DIM
NATIVEDATE
YEAR
YEAR_MONTH

Multi-source Universes example: DF Designer



Target tables
- CUSTOMER
  - Mapping rules
    - c1: Customer
      - Datasources
        - SQLServer
          - S20: tpcr.CUSTOMER
            - A1: C_CUSTKEY
            - A2: C_NAME
            - A3: C_ADDRESS
            - A4: C_NATIONKEY
            - A5: C_PHONE
            - A6: C_ACCTBAL
            - A7: C_MKTSEGMENT
            - A8: C_COMMENT
      - Lookup tables
      - Domain tables
  - Constraints
- MERGED_ORDERS
- NATION
- ORDERS

Target tables > CUSTOMER > Mapping rules > Customer

Status

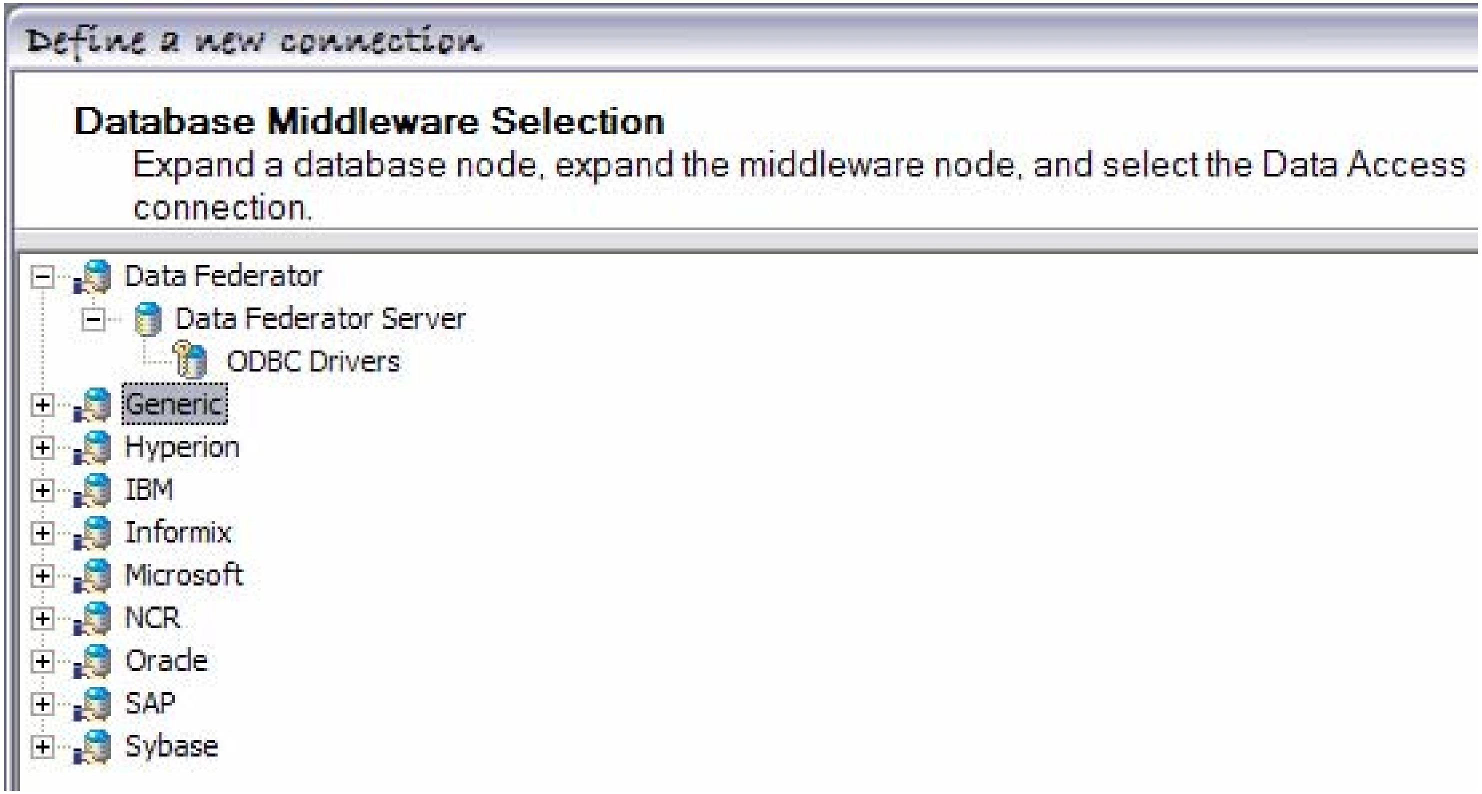
🟩 Completed — ▢ Tested — ▢ Integrated

General

Mapping formulas

Target columns	Formulas
C_CUSTKEY	= S20.C_CUSTKEY
C_NAME	= S20.C_NAME
C_ADDRESS	= S20.C_ADDRESS
C_NATIONKEY	= S20.C_NATIONKEY
C_PHONE	= S20.C_PHONE
C_ACCTBAL	= S20.C_ACCTBAL
C_MKTSEGMENT	= S20.C_MKTSEGMENT
C_COMMENT	= S20.C_COMMENT

Multi-source Universes example: Connectivity



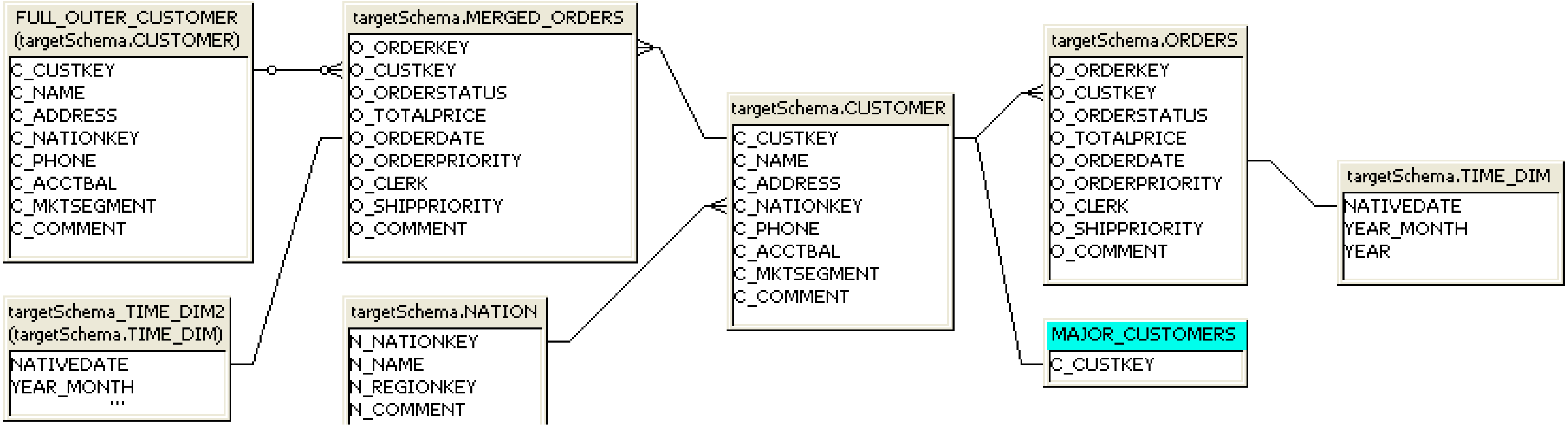
Define a new connection

Database Middleware Selection

Expand a database node, expand the middleware node, and select the Data Access connection.

- Data Federator
  - Data Federator Server
    - ODBC Drivers
- Generic
- Hyperion
- IBM
- Informix
- Microsoft
- NCR
- Oracle
- SAP
- Sybase

Multi-source Universes example: Universe



FULL_OUTER_CUSTOMER
(targetSchema.CUSTOMER)
C_CUSTKEY
C_NAME
C_ADDRESS
C_NATIONKEY
C_PHONE
C_ACCTBAL
C_MKTSEGMENT
C_COMMENT

targetSchema.MERGED_ORDERS
O_ORDERKEY
O_CUSTKEY
O_ORDERSTATUS
O_TOTALPRICE
O_ORDERDATE
O_ORDERPRIORITY
O_CLERK
O_SHIPPRIORITY
O_COMMENT

targetSchema.CUSTOMER
C_CUSTKEY
C_NAME
C_ADDRESS
C_NATIONKEY
C_PHONE
C_ACCTBAL
C_MKTSEGMENT
C_COMMENT

targetSchema.ORDERS
O_ORDERKEY
O_CUSTKEY
O_ORDERSTATUS
O_TOTALPRICE
O_ORDERDATE
O_ORDERPRIORITY
O_CLERK
O_SHIPPRIORITY
O_COMMENT

targetSchema.TIME_DIM
NATIVEDATE
YEAR_MONTH
YEAR

targetSchema_TIME_DIM2
(targetSchema.TIME_DIM)
NATIVEDATE
YEAR_MONTH
...

targetSchema.NATION
N_NATIONKEY
N_NAME
N_REGIONKEY
N_COMMENT

MAJOR_CUSTOMERS
C_CUSTKEY

Agenda

- **Clients accessing the universe**
  - The Semantic Layer concept
  - Using data in your applications
- **Universe tips**
  - Measures on multiple fact tables
  - Index awareness
  - Multi-source Universes
- **Semantic Layer roadmap**

Semantic Layer Roadmap

Semantic layer becomes pervasive

- **Universe support in Live Office** ✓
- **Query As a web service** ✓
- **EPM and dashboard manager universe access** ✓
- **Personal data provider in Web Intelligence**
- **Universes continue to extend to new sources**
  - Stored procedures call in Universes
  - Javabean access in Universes
  - XML and web services access available through Data Federator
  - JDBC drivers for all major sources

Semantic Layer Roadmap

Semantic layer becomes pervasive

- **Continued investment in OLAP universes**
  - Support of MSAS2005
  - Improved support for SAP BW
  - Improved OLAP universe maintenance and update
- **Closer integration between Data Federator and universes**
- **Business views continue with no investment**

Thank you!

- **Questions?**

- **Contact info:**

  Pierpaolo Vezzosi

  pierpaolo.vezzosi@businessobjects.com

  +33(0)1.4125.3840

Business Objects™