Automated Semantic Extraction from Databases

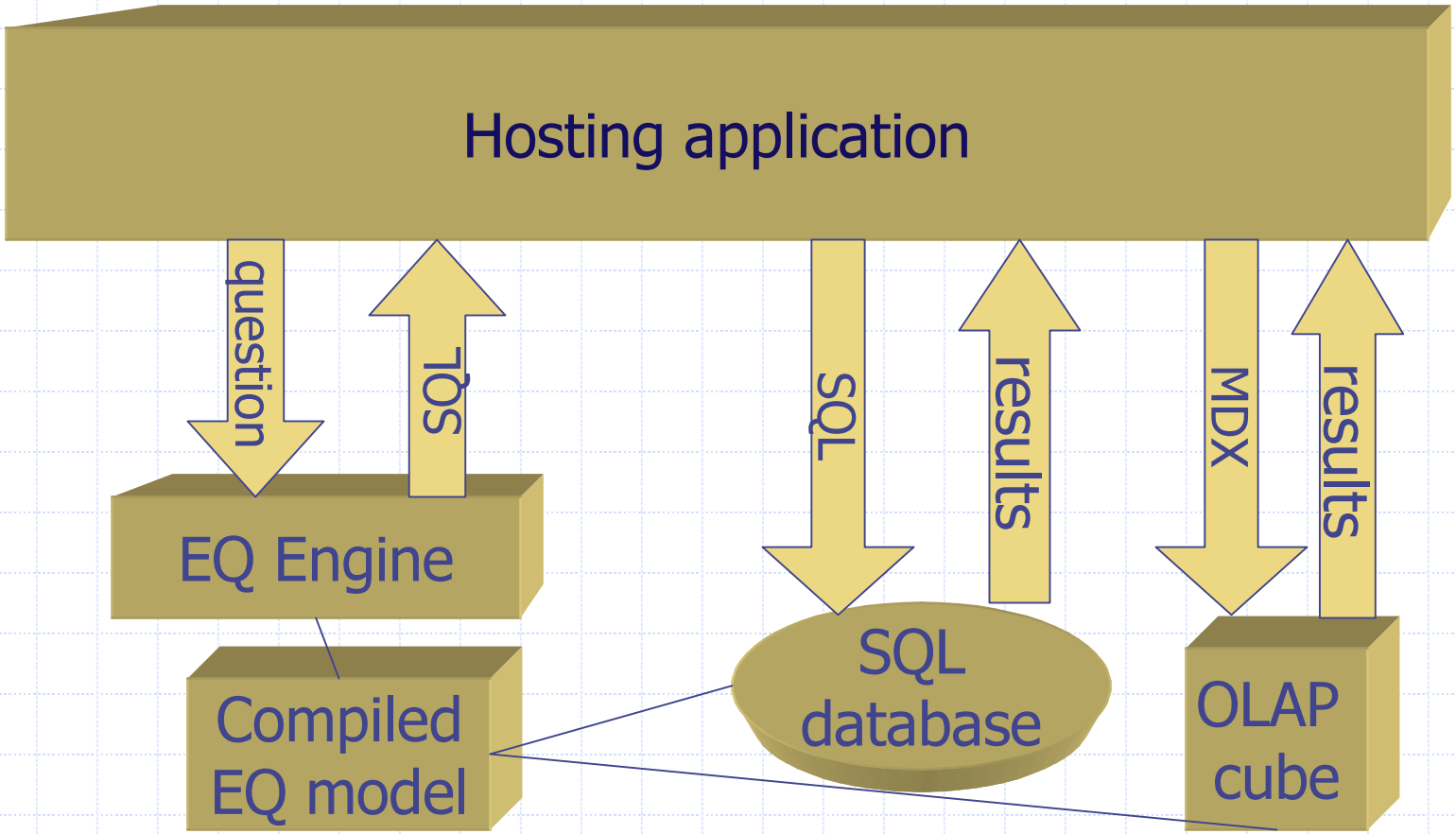
in Microsoft SQL Server's English Query

Adam Blum
team lead, English Query/SQL Server
www.microsoft.com/sql/eq
adamblum@microsoft.com

What Is English Query

- **Engine**
  - COM object model that converts English sentences (questions) into SQL statements
  - Can be used in any COM-supporting application
- **Modeling tool**
  - Used to build semantic model of database needed by EQ engine
  - Developer creates entities and relationships corresponding to objects in SQL or OLAP database

EQ Runtime Scenario



What are EQ Semantic Models?

- entities
  - map to **table**, **field**, OLAP **dimension**, **level**, **property**, or **fact**
- relationship
  - In SQL contain **join table**
  - Entities play **role**s in relationship
    - Roles have **join path**, and can be **quantified**
  - Phrasings may be attached to relationships: **name, trait**, **verb, adjective, subset, preposition**

English Query in SQL Server 7.0

- what's good
  - easy access to information for all users
  - powerful, flexible analysis of data
- what could be better
  - building semantic models for complex schemata is tedious and time-consuming
  - difficult to understand and manipulate the resulting large models
  - Use restricted to "fixed schemata" (vertical applications) that can be individually handmodeled
    - Generic database tool vendors (Excel, Access, OLAP clients, reporting tools) would like to allow English queryability

The Solution

(What We Just Built for English Query 7.5)

- Graphical authoring
  - Allows easier creation of relationships
  - Understandability of larger models
  - Oriented to showing subsets of model
- Programmatic authoring
  - Semantic Modeling Format (SMF) – XML grammar for representing database semantics
  - Allows applications to programmatically generate EQ semantic models via the XML Document Object Model
  - Authoring Object Model – allows building from DOM
- Model Wizards
  - Automate 90% of OLAP entity and relationship creation, 70% of SQL entities and relationship
  - Can be driven via AOM, allowing EQ use against unknown schemas

Semantic Modeling Format

- Grammar is defined by SMF.DTD (Document Type Definition) and SMFSchema.XML (XML-Data Schema) available at microsoft.com/sql/eq
- SMF is used to define semantics of a database and mappings of semantics to SQL or OLAP database objects
- <SEMANTICS> element contains <ENTITY> elements and <RELATIONSHIP> elements with links to <TABLE> or <FIELD> elements or OLAP database elements
- "Traditional XML advantages": human-readable, platform independent, Internet friendly, availability of XML tools and APIs (the DOM) gives us "free authoring API"

SMF Elements

- Top level elements
  - <MODEL ID="Northwind">
    - <MODULE>
      <SEMANTICS>
      - <ENTITY>…</ENTITY>
      - <RELATIONSHIP>…</RELATIONSHIP
    - </SEMANTICS>
    - <TABLES>…</TABLES>
    - </MODULE>
    - <PROJECT><DATABASE>
  - </MODEL>
- ENTITY
  - <ENTITY ID="ENTITY:customer"> <
    WORD>customer</WORD>
    <WORD>buyer</WORD>
    <WORD>client</WORD>
    <DBOBJECT TABLE="customers"/>
    <DISPLAY FIELD="CustomerName"/>
    </ENTITY>

The SMF Relationship Element

- RELATIONSHIP

```
<RELATIONSHIP ID="customers_order_products_from_employees">
<JOINTABLE TABLE="OrderDetails"/>
<ROLE ID="customer" HREF="customer"></ROLE>
<ROLE ID="employee" HREF="employee"></ROLE>
<ROLE ID="product" HREF="product"><AMOUNT TYPE="ENTITY“
HREF="quantity"/></ROLE>
<ROLE ID=“orderdate” HREF="ENTITY:order_date"></ROLE>
<PHRASINGS>
  <VERBPHRASING><SUBJECT ROLEREF="customer"/>
    <VERB>order</VERB>
    <VERB>buy</VERB>
    <VERB>purchase</VERB>
    <OBJECT ROLEREF="product"/>
    <PREPPHRASE>
      <PREP>from</PREP>
      <OBJECT ROLEREF="employee"/>
    </PREPPHRASE>
  </VERBPHRASING>
</PHRASINGS>
<WHEN ROLEREF="orderdate"/>
</RELATIONSHIP>
```

Model Wizards

- Automatically create entities and relationships based on structure of database
  - Default mode of starting a new English Query project
- Extremely effective against OLAP databases
  - Richer database model including hierarchy
  - Constrained, well-formed schemata
  - Friendly names
- Effective with SQL databases assuming good database design practices
  - Need friendly names, primary keys, foreign keys, …
- Invoked programmatically via AutoModel()
  - Enables generic database tool scenario: AutoModel(), create XML DOM object, add entities and relationships that tool knows about via XML

OLAP Model Wizard heuristics

- Entities created …
  - for every OLAP object: Dimensions, Levels, Properties, Measures, Facts
  - primary word from OLAP object name, two synonyms supplied
- Relationships created …
  - **Trait relationships –**
    - <dimension entity> has <level entity> (for each level entity)
    - <level entity> has <higher level entity>
    - <dimension entity> has <level entity> (for each level entity)
  - Hierarchy ("in") relationships
    - dimension entity becomes lowest level in semantic hierarchy
    - EQ engine single level of inference forces "extra relationships":
      - Direct relationships: "products are in brands", "brands are in subcategories", "subcategories are in categories", "categories are in departments"
      - "extra" relationships: "products are in subcategories", "subcategories are in departments"
  - Name relationship
    - bottom level entity becomes name entity of dimension entity, e.g. "product names are the names of products"
  - Dimension to dimension entity relationships
    - *can be created* between each pair dimension entities (modeler can enable), imputed from the dimension's common relationships to the OLAP fact table
    - Preferred method is for modeler to create verb relationship for fact table, e.g. "customers buy products in stores on dates"

OLAP Model Wizard Example

This cube

- Sales
  - Customer
    - Country
    - State
    - City
    - Country
  - Product
    - Department
    - Category
    - Subcategory
    - Brand
    - Product Name
  - Store
    - Country
    - Region
    - District
    - City
    - Store Name
  - Time
    - Year
    - Quarter
    - Month
    - Day

Creates these entities and relationships:

- customer
  - customer names are the names of customers
  - customers are in cities
  - cities are in states
  - states are in countries
- product
  - product names are the names of products
  - products are in brands
  - brands categorize products (checked)
  - brands are in subcategories
  - subcategories categorize products (checked)
  - subcategories are in categories
  - categories categorize products (checked)
  - categories are in departments
  - departments categorize products (checked)
  - products are in subcategories
  - subcategories are in departments
- store
  - store names are the names of stores
  - stores are in cities
  - cities are in districts
  - districts are in regions
  - regions are in countries
- sale time

SQL Model Wizard Heuristics

- Entities created …
  - for all tables and fields, except "join tables (tables w/only keys)
  - "When" entity type set if associated with date/time field
  - "Where" entity type set if entity is "where word"
  - Proper name type set by sampling data
- Relationships
  - Trait relationships to fields that are neither foreign keys, primary keys or binary
  - Name relationships based on patterns in field name
  - Trait relationships to "foreign key destination" entities
  - "in" relationships optionally (unchecked) created from table entity to field entity if field is String type and data is firstcaps