Document-Centric OLAP in the Schema-Chaos World

Yannis Sismanis
Berthold Reinwald
Hamid Pirahesh

IBM Almaden Research Center

Introduction

- Business Insight Applications
  - Real-Time
  - Document-Centric
- Integrating plethora of data sources
  - In-house Applications (CRM, ERP,...)
  - Partner Databases (Retailers,…)
  - Syndicated Databases (Credit reports, UNSPSC,…)
- Different representation, same semantics
  - Schema-Chaos
  - Difficult to index the data and express queries
- ETL every source to the same schema
  - Bad Scalability
  - Very high cost of ownership

Schema-Chaos

```
<SAP46Order>
  <date> 23 Nov 2005 </date>
  <customer>
    <id>8334</id>
    <name>Sally Kwan</name>
    <address>
      S. Oak St.
      San Fransisco, CA 95100
    </address>
  </customer>

  <product>
    <id>KLE</id>
    <quantity>4</quantity>
    <price>56</price>
  </product>

  <product>
    <id>FGE</id>
    <quantity>6</quantity>
    <price>30</price>
  </product>
</SAP46Order>
```

```
<SAP46Product>
  <id>KLE</id>
  <category>Office</category>
  <name>Desk</name>
</SAP46Product>

<SAP46Product>
  <id>FGE</id>
  <category>Glass</category>
  <name>Window</name>
</SAP46Product>
```

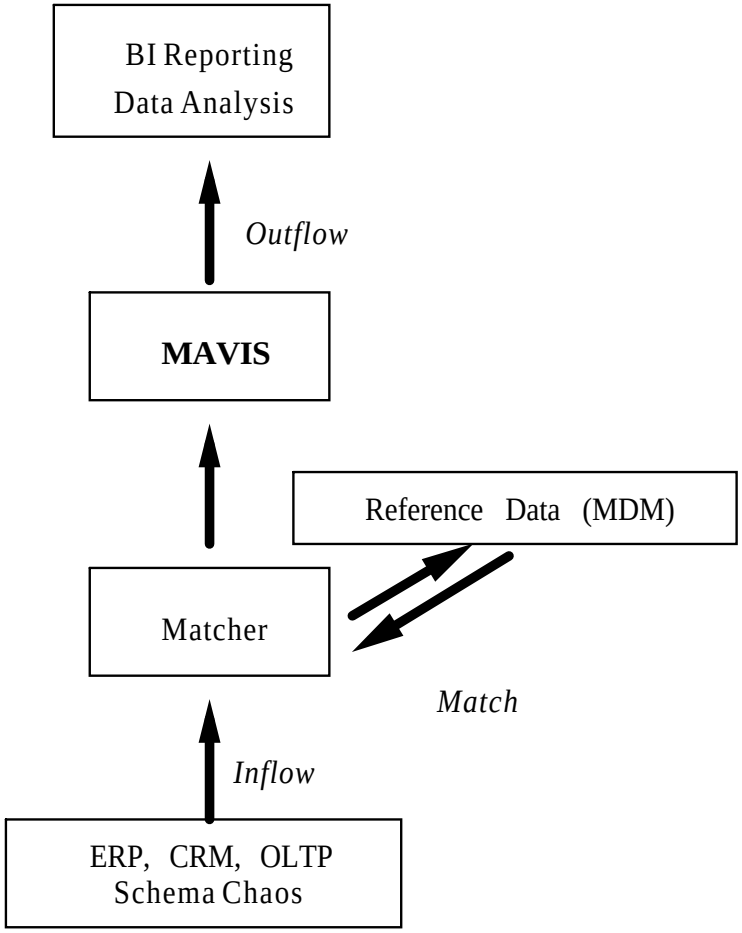
```
<PeopleSoft>
  <item>
    <date> 12 Sep 2005 </date>
    <customer>C345</customer>
    <sold>
      <id>P3445</id>
      <quantity>1</quantity>
      <cost>100</cost>
    </sold>
  </item>

  <item>
    <date> 10 Nov 2005 </date>
    <customer>C121</customer>
    <sold>
      <id>P4332</id>
      <quantity>2</quantity>
      <cost>50</cost>
    </sold>
  </item>
</PeopleSoft>
```

- Plethora of different schemas
  - New versions, applications, business components
  - Full mapping to a target schema does not scale (human cost)
- All hope is not lost!
- Typical Aggregation Query

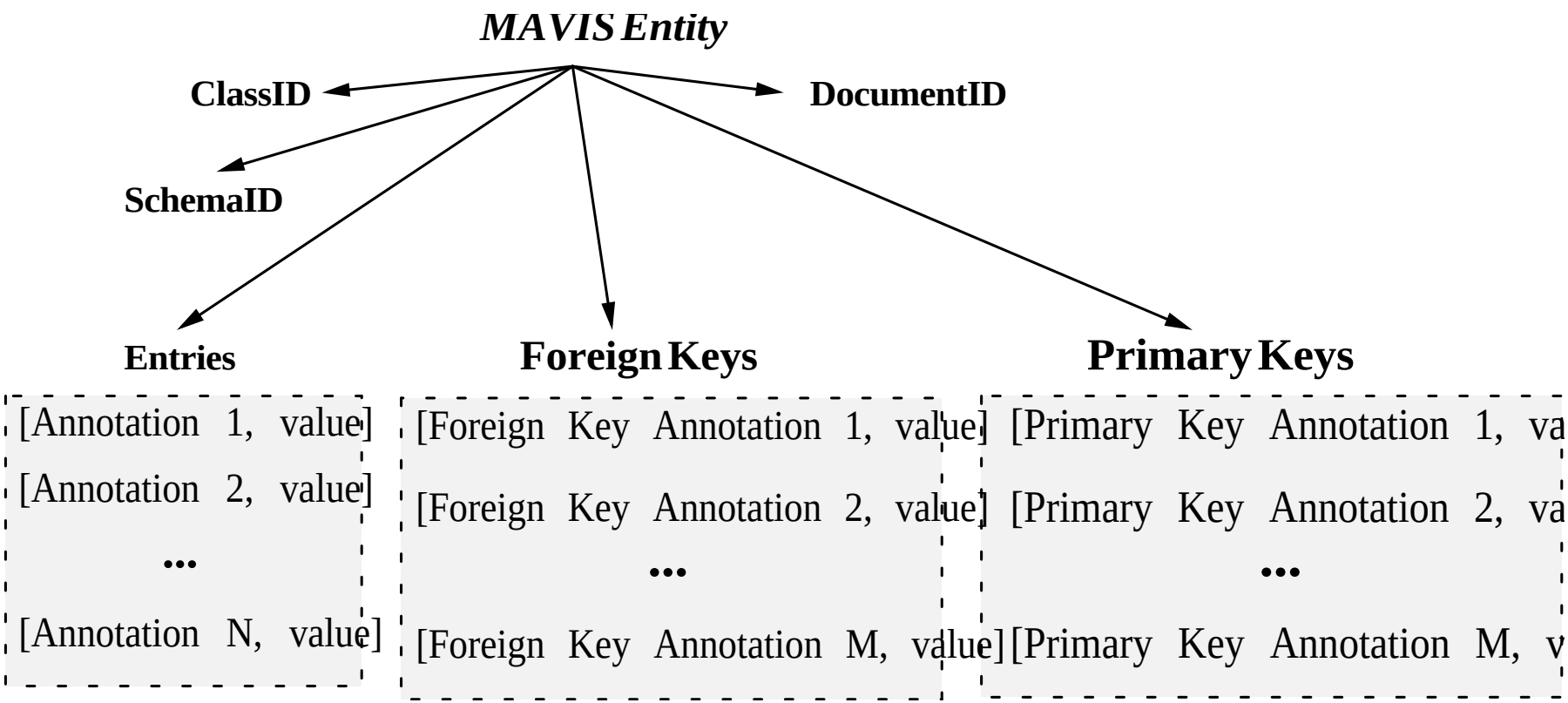
Reference Constraint:	**"Office Supplies" for Product**
Fact Constraint:	**"2005" for Date**
Grouping Attribute:	**State in Customer Location**
Aggregation:	**Sum of price in Order**

General Architecture



- Next-Generation Warehouse
  - ***Data-Driven***
  - See [2] for more details
- Automatic Metadata Discovery
  - Similarities (matches)
    - Operational data to Master Data
  - Primary/Foreign Key Discovery
    - See [8]
    - Join paths
- Unstructured Content
  - Annotated using approaches like [6]
- Storage/Indexing using MAVIS
  - MAterialized VIews for Schema-chaos
  - ***Flexibility (primary)***
  - Performance (secondary)

MAVIS


MAVIS Entity
ClassID
DocumentID
SchemaID
Entries
Foreign Keys
Primary Keys
[Annotation 1, value]
[Annotation 2, value]
...
[Annotation N, value]
[Foreign Key Annotation 1, value]
[Foreign Key Annotation 2, value]
...
[Foreign Key Annotation M, value]
[Primary Key Annotation 1, va
[Primary Key Annotation 2, va
...
[Primary Key Annotation M, v

MAVIS Example I

```
<SAP46Order>
  <date> 23 Nov 2005 </date>
  <customer>
    <id>8334</id>
    <name>Sally Kwan</name>
    <address>
      S. Oak St.
      San Fransisco, CA 95100
    </address>
  </customer>

  <product>
    <id>KLE</id>
    <quantity>4</quantity>
    <price>56</price>
  </product>

  <product>
    <id>FGE</id>
    <quantity>6</quantity>
    <price>30</price>
  </product>
</SAP46Order>
```

```
<SAP46Product>
  <id>KLE</id>
  <category>Office</category>
  <name>Desk</name>
</SAP46Product>

<SAP46Product>
  <id>FGE</id>
  <category>Glass</category>
  <name>Window</name>
</SAP46Product>
```

ClassID	Entries	Foreign Keys
Order	[Date, 23 Nov 2005]	[CustomerKey, 8334]
SchemaID: SAP	[Name, Sally Kwan]	[ProductKey, KLE]
DocumentID: D1	[Address, S. Oak St.]	
	[Quantity, 4]	
	[Price, $56]	

ClassID	Entries	Primary Keys
Product	[Name, Desk]	[ProductKey, KLE]
SchemaID: SAP	[Category, Office]	
DocumentID: D3		

Rental Column Implementation

			Entry Rental					Foreign Rental					Primary Rental		
Class	Schema	Doc	Map_1	$Value_1$	Map_2	$Value_2$	...	$Foreign_1$	$FValue_1$	$Foreign_2$	$FValue_2$	...	$Primary_1$	$PValue_1$	...
Order	SAP	D1	Date	23 Nov 2005	Name	Sally	...	CustKey	8334	ProdKey	KLE	...			...
Order	SAP	D1	Date	23 Nov 2005	Name	Sally	...	CustKey	8334	ProdKey	FGE	...			...
Order	People	D2	Date	12 Sep 2005			...	CustKey	C345	ProdKey	P3445	...			...
Order	People	D2	Date	10 Nov 2005			...	CustKey	C121	ProdKey	P4332	...			...
Product	SAP	D3			Category	Office	...					...	ProdKey	KLE	...
Product	SAP	D3			Category	Glass	...					...	ProdKey	FGE	...

ClassID
Order

SchemaID
SAP

DocumentID
D1

Entries
[Date, 23 Nov 2005]
[Name, Sally Kwan]
[Address, S. Oak St.]
[Quantity, 4]
[Price, $56]

Foreign Keys
[CustomerKey, 8334]
[ProductKey, KLE]

ClassID
Product

SchemaID
SAP

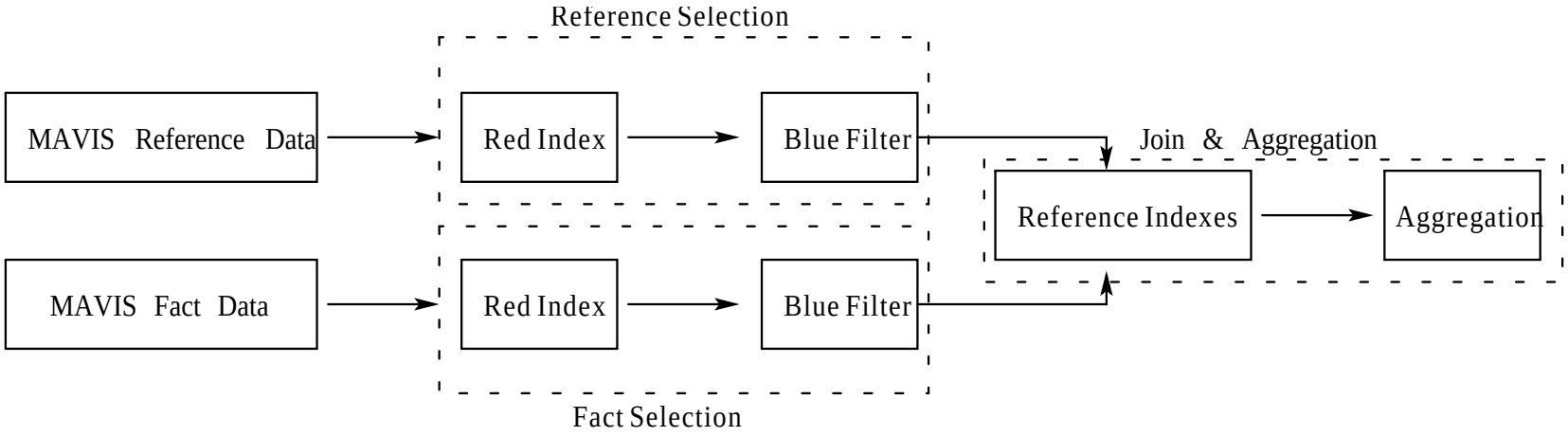
DocumentID
D3

Entries
[Name, Desk]
[Category, Office]

Primary Keys
[ProductKey, KLE]

- Focus on Flexibility
  - Remember Schema-Chaos
  - Performance is important (but secondary)
  - Moderns DBMS's optimize null storage
- Remotely similar to column-store oriented systems

Querying using MAVIS



- Builts upon automatic metadata discovery
  - Discovery of foreign/primary keys, similarities & annotations for unstructured content
- "Red" Index
  - Allows for fast keyword search
  - Low-precision high-recall
- "Blue" Filter
  - Increases accuracy
- Foreign/Primary Index
  - Composite indexes that allow for fast joins between facts and reference data

MAVIS Processing Implementation

			Entry Rental					Foreign Rental					Primary Rental		
Class	Schema	Doc	Map_1	$Value_1$	Map_2	$Value_2$	...	$Foreign_1$	$FValue_1$	$Foreign_2$	$FValue_2$	...	$Primary_1$	$PValue_1$	...
Order	SAP	D1	Date	23 Nov 2005	Name	Sally	...	CustKey	8334	ProdKey	KLE	...			...
Order	SAP	D1	Date	23 Nov 2005	Name	Sally	...	CustKey	8334	ProdKey	FGE	...			...
Order	People	D2	Date	12 Sep 2005			...	CustKey	C345	ProdKey	P3445	...			...
Order	People	D2	Date	10 Nov 2005			...	CustKey	C121	ProdKey	P4332	...			...
Product	SAP	D3			Category	Office	...					...	ProdKey	KLE	...
Product	SAP	D3			Category	Glass	...					...	ProdKey	FGE	...

Typical Aggregation Query

Reference Constraint: **"Office Supplies" for Product**
Fact Constraint: **"2005" for Date**
Grouping Attribute: **State in Customer Location**
Aggregation: **Sum of price in Order**

```
CREATE VIEW FilteredOrders AS
SELECT Class, Schema, Doc,
       CASE
            when Map1='Date' then Value1
            when Map2='Date' then Value2
            when Map3='Date' then Value3
            ...
       END as Date,...
FROM MAVIS
WHERE Date contains '2005' and Class=Order
```

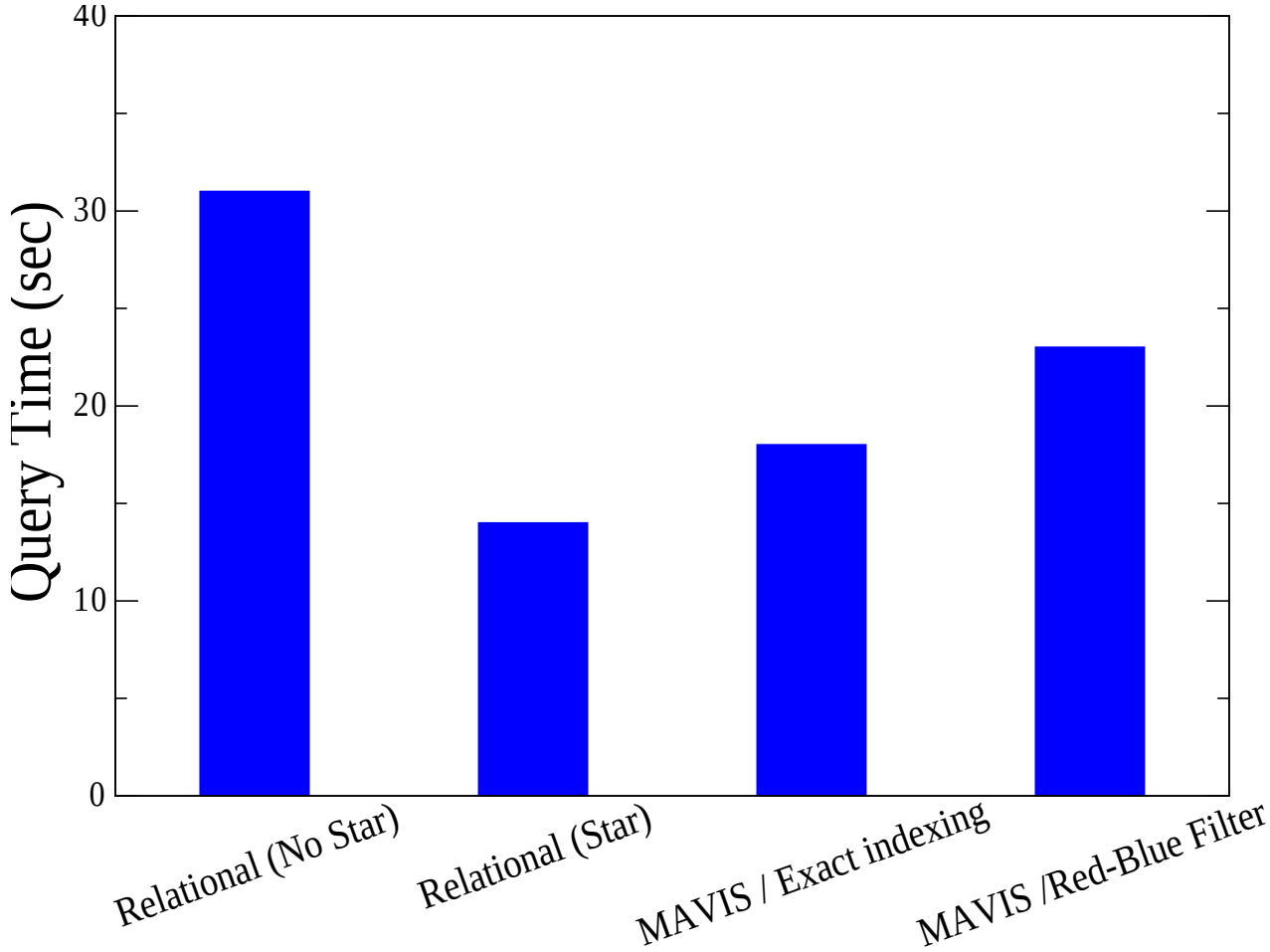
```
CREATE VIEW FilteredProducts AS
SELECT Class, Schema, Doc,
       CASE
            when Map1='Category' then Value1
            when Map2='Category' then Value2
            when Map3='Category' then Value3
            ...
       END as Category,...
FROM MAVIS
WHERE Class=Product
      and Category contains 'Office Supplies'
```

```
CREATE VIEW JoinResult AS
SELECT Fact.Price, Cust.State
FROM FilteredOrders as Fact, FilteredProduct as Prod, MAVIS as Cust
WHERE
       (*Join Orders and Products*)
      (Fact.Foreign1='ProdKey' and Prod.Primary1='ProdKey' and Fact.FValue1=Prod.PValue1) or
      (Fact.Foreign2='ProdKey' and Prod.Primary1='ProdKey' and Fact.FValue2=Prod.PValue1) or
      (Fact.Foreign1='ProdKey' and Prod.Primary2='ProdKey' and Fact.FValue1=Prod.PValue2) or
      (Fact.Foreign2='ProdKey' and Prod.Primary2='ProdKey' and Fact.FValue2=Prod.PValue2)
```

MAVIS Experiments

"Customers who buy Telephone Equipment in the states of CA,DC paying with Mastercard" (3-way join)

150k Orders with 300k Items, 3k Customers, 300k Products



Conclusions

- Next generation Business Intelligence applications
  - Require radically new approaches/technologies
- Scalable / Robust solution for OLAP queries
  - Document-centric, Schema-chaos
  - Data-driven
  - Automated metadata discovery
- Introduced MAVIS for storing/querying next gen warehouses
- Implemented on top of existing commercial-strength system
- Focus on flexibility
  - With performance comparable to an optimized warehouse

Questions