

Temporal Data Warehouses: Logical Models and Querying

Waqas Ahmed, Esteban Zimányi, Robert Wrembel
waqas.ahmed@ulb.ac.be

Université Libre de Bruxelles
Poznan University of Technology

April 2, 2015



Outline

- 1 Introduction
- 2 Available Approaches
- 3 Running Example
- 4 Temporal Queries
- 5 Conclusion

Outline

- 1 Introduction
- 2 Available Approaches
- 3 Running Example
- 4 Temporal Queries
- 5 Conclusion

Type of Changes in External Data Sources

External data sources change in their:

① Content

- Daily operations

② Structure

- Changes in business requirements
- Changes in the modeled reality
- Adaptation of new technologies
- Simulation scenarios
- Changes in legislation

Propagate Changes in EDS into Data Warehouses

- As a result of changes in EDSs, DWs also change in their:
 - ① Content
 - products always change their prices
 - ② Structure
 - Splitting the hierarchy Product \rightarrow Category to Product \rightarrow Sub-Category \rightarrow Category
- The user may be interested in keeping the history of changes in the content and structure
 - to reconstruct the state of the business world in the past
 - to simulate the effect of future changes
 - for audit and accountability purposes
- In this paper, we consider the changes in the content

Outline

- 1 Introduction
- 2 Available Approaches**
- 3 Running Example
- 4 Temporal Queries
- 5 Conclusion

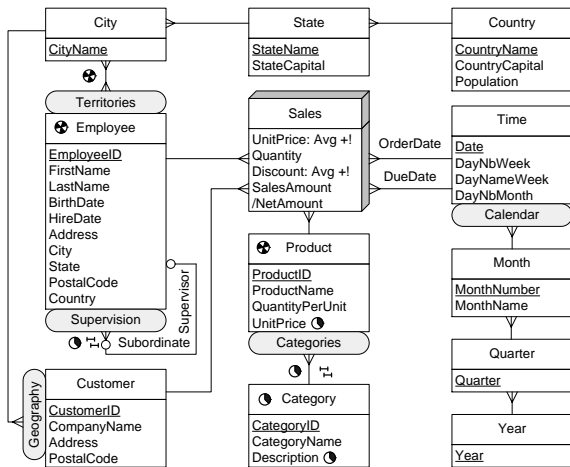
Available Approaches to Keep the Content History

- Two available approaches to keep the history of content changes in a DW
 - ① Slowly Changing Dimensions (SCDs) Type 2
 - ② Temporal Data Warehouses (TDWs)

Outline

- 1 Introduction
- 2 Available Approaches
- 3 Running Example**
- 4 Temporal Queries
- 5 Conclusion

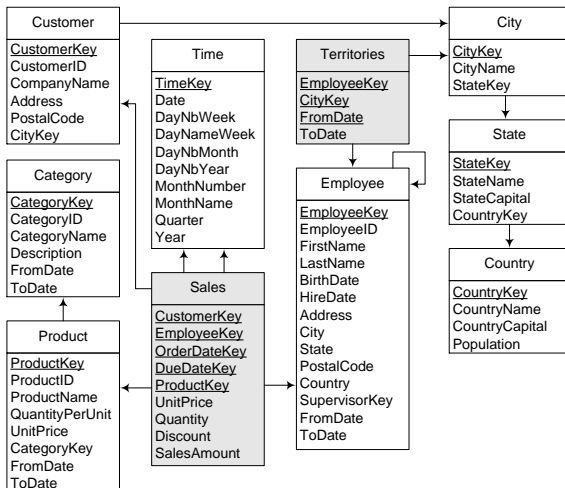
An Example Sales DW



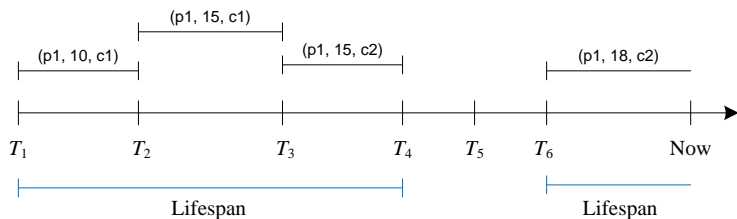
Temporal Requirements

- Product life-cycle
- Product price
- Product-category assignment
- Category description
- Employee life-cycle
- Employee-supervisor assignment
- Employee-city assignment

SCD Type 2 Schema of Sales DW



SCD Type 2 Implementation



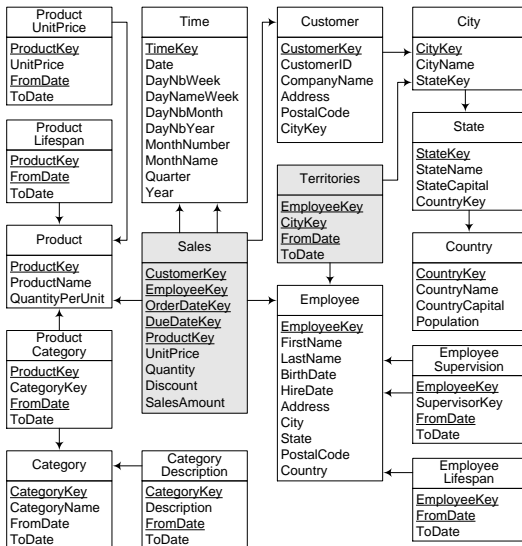
Time line of the evolution of unit price and category of product **p1**

Product

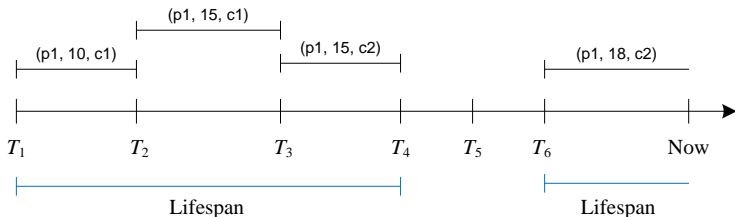
<u>ProductKey</u>	ProductID	UnitPrice	CategoryKey	FromDate	ToDate
1	p1	10	c1	T_1	T_2
2	p1	15	c1	T_2	T_3
3	p1	15	c2	T_3	T_4
4	p1	18	c2	T_5	Now

SCD Type 2 implementation

Temporal DW schema of Sales DW



Temporal Data Warehouse Implementation



ProductLifespan

ProductKey	FromDate	ToDate
p1	T_1	T_4
p1	T_6	Now

ProductUnitPrice

ProductKey	UnitPrice	FromDate	ToDate
p1	10	T_1	T_2
p1	15	T_2	T_4
p1	18	T_6	Now

ProductCategory

ProductKey	CategoryKey	FromDate	ToDate
p1	c1	T_1	T_3
p1	c2	T_3	T_4
p1	c2	T_6	Now

Temporal DW implementation

Outline

- 1 Introduction
- 2 Available Approaches
- 3 Running Example
- 4 Temporal Queries**
- 5 Conclusion

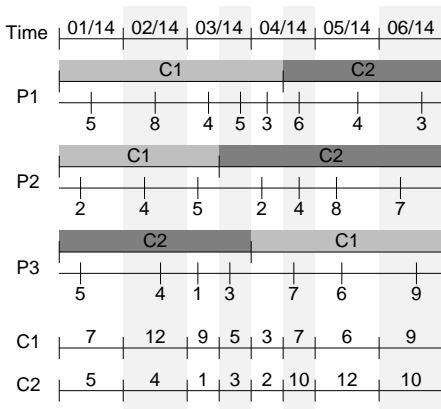
Classes of temporal queries

No.	Class	Example
Q1	Temporal roll-up	Total quantity per category and month.
Q2	Temporal-roll-up (with window)	Monthly year-to-date sales per category.
Q3	Temporal roll-up (recursive)	Total sales amount made by an employee and her subordinates during 1997.
Q4	Temporal selection	For each employee, total sales amounts of products she sold with unit price greater than \$30 at the time of the sale.
Q5	Temporal projection	Total sales amount for supervisors.
Q6	Temporal union	Total sales amounts for products assigned to categories beverages or dairy products.
Q7	Temporal join	Name, unit price, and total sales amount by month for products.
Q8	Temporal difference	Total sales amount for employees assigned to only one city.
Q9	Temporal aggregation	For each month, give the total quantity of products sold per category.
Q10	Temporal aggregation (for a many-to-many relationship)	For each employee, total sales amount, number of cities, and number of states to which she is assigned.
Q11	Temporal aggregation (duration of interval)	For each pair of employee and supervisor, total number of days when the supervision lasted.
Q12	Temporal universal quantifier	Total sales for categories in which all products have a price greater than \$15
Q13	Time slice	Average unit price by category as of January 1st, 1997.

Table: Classes of temporal business queries and example queries

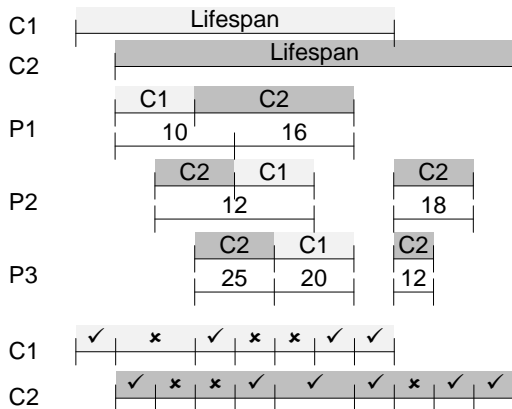
Temporal Roll-up

- Total quantity per category and month



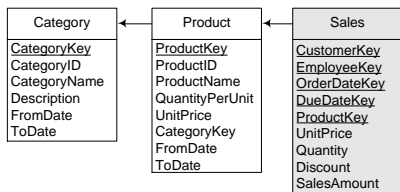
Temporal Universal Quantification

- Total sales per category in which all products have price greater than \$15



Comparing the Queries

- Name, unit price, and total sales amount by month for products



Product schema in SCD Type 2

In Query for SCD Type 2

- Month must be split according to unit price variations
- Temporal projection (with coalescing) is required to compute the Product's unit price with the period of its applicability
- A temporal join of the computed product unit price with the month from the sales table, and traditional join with the other tables

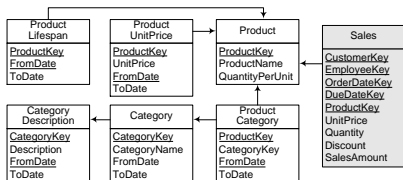
SQL Query for SCD Type 2

- Name, unit price, and total sales amount by month for products

```
WITH ProdUnitPrice(ProductID, UnitPrice, FromDate, ToDate) AS (  
SELECT ProductID, UnitPrice, FromDate, ToDate  
FROM Product ),  
ProdUnitPriceCoalesced AS (  
— Coalescing the table ProdUnitPrice above ... ),  
SELECT P.ProductName, U.UnitPrice, SUM(SalesAmount) AS SalesAmount,  
dbo.MaxDate(M.FromDate, U.FromDate) AS FromDate,  
dbo.MinDate(M.ToDate, U.ToDate) AS ToDate  
FROM Sales S, Time T, Product P, ProdUnitPriceCoalesced U, Month M  
WHERE S.OrderDateKey = T.TimeKey AND S.ProductKey = P.ProductKey AND  
P.ProductID = U.ProductID AND dbo.MaxDate(M.FromDate, U.FromDate) <  
dbo.MinDate(M.ToDate, U.ToDate) AND  
dbo.MaxDate(M.FromDate, U.FromDate) <= T.Date AND  
T.Date < dbo.MinDate(M.ToDate, U.ToDate)  
GROUP BY P.ProductName, U.UnitPrice, M.FromDate, U.FromDate,  
M.ToDate, U.ToDate  
ORDER BY P.ProductName, dbo.MaxDate(M.FromDate, U.FromDate)
```

Comparing the Queries (cont.)

- Name, unit price, and total sales amount by month for products



Product schema in TDW

In Query for TDW

- Temporal projection with coalescing is not needed as product unit price is directly available
- A temporal join of the product unit price with the month from the sales table, and traditional join with the other tables
- No projection and coalescing needed, hence simplified query

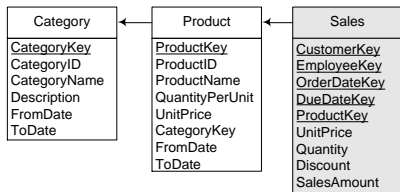
SQL Query for TDW

- Name, unit price, and total sales amount by month for products

```
SELECT P.ProductName, U.UnitPrice, SUM(SalesAmount) AS SalesAmount,
dbo.MaxDate(M.FromDate, U.FromDate) AS FromDate,
dbo.MinDate(M.ToDate, U.ToDate) AS ToDate
FROM Sales S, Time T, Product P, ProductUnitPrice U, Month M
WHERE S.OrderDateKey = T.TimeKey AND S.ProductKey = P.ProductKey AND
P.ProductKey = U.ProductKey AND dbo.MaxDate(M.FromDate, U.FromDate) <
dbo.MinDate(M.ToDate, U.ToDate) AND
dbo.MaxDate(M.FromDate, U.FromDate) <= T.Date AND
T.Date < dbo.MinDate(M.ToDate, U.ToDate)
GROUP BY P.ProductName, U.UnitPrice, M.FromDate, U.FromDate, M.ToDate, U.ToDate
ORDER BY P.ProductName, dbo.MaxDate(M.FromDate, U.FromDate)
```

Comparing the Queries (cont.)

- Average unit price by category as of January 1st, 1997.



Product schema in SCD Type 2

In Query for SCD Type 2

- Product evolution exists in the single table
- Query would need join of two table i.e. Product and Category

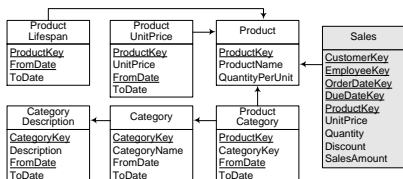
SQL Query for SCD Type 2

- Average unit price by category as of Januray 1st, 1997.

```
SELECT C.CategoryName, AVG(UnitPrice) AS AvgUnitPrice
FROM Product P, Category C
WHERE P.CategoryKey = C.CategoryKey AND
P.FromDate <= '1997-01-01' AND '1997-01-01' < P.ToDate AND
C.FromDate <= '1997-01-01' AND '1997-01-01' < C.ToDate
GROUP BY C.CategoryName
```

Comparing the Queries (cont.)

- Average unit price by category as of January 1st, 1997.



Product schema in TDW

In Query for TDW

- Product evolution is stored in multiple tables
- Query would need join of four table i.e. Product, ProductUnitPrice, ProductCategory, and Category

SQL Query for TDW

- Average unit price by category as of Januray 1st, 1997.

```
SELECT C.CategoryName, AVG(UnitPrice) AS AvgUnitPrice
FROM Product P, ProductUnitPrice PU, ProductCategory PC, Category C
WHERE P.ProductKey = PC.ProductKey AND
PC.CategoryKey = C.CategoryKey AND
P.ProductKey = PU.ProductKey AND
PC.FromDate <= '1997-01-01' AND '1997-01-01' <= PC.ToDate AND
PU.FromDate <= '1997-01-01' AND '1997-01-01' <= PU.ToDate AND
C.FromDate <= '1997-01-01' AND '1997-01-01' <= C.ToDate
GROUP BY C.CategoryName
```

Outline

- 1 Introduction
- 2 Available Approaches
- 3 Running Example
- 4 Temporal Queries
- 5 Conclusion**

Conclusion

- Keeping the history of changes in content enables users to query various states of the business world
- Slowly Changing Dimensions Type 2 and Temporal Data Warehouses are two approaches to keep content history
- It is important to know which of these two approaches is most suitable for time-related business questions

Future Work

- Generate temporal data set based on statistics from the real world data
- Run the representative queries from identified classes on the benchmark data
- Determine which approach is suitable for each class of queries
- Consider changes in the structure
- Combine the approaches to manage changes in the content and structure in a single solution