# Aggregation and Ordering in Factorized Databases


http://www.cs.ox.ac.uk/projects/FDB/

**B**akibayev, **K**očiský, **O**lteanu, and **Z**ávodný
University of Oxford

VLDB Sept 2, 2014

# Outline


What are Factorized Databases?

Applications

A Glimpse at Aggregating Factorized Data

# Factorized Databases by Example

Orders

| customer | day | pizza |
|---|---|---|
| Mario | Monday | Capricciosa |
| Mario | Friday | Capricciosa |
| Pietro | Friday | Hawaii |
| Lucia | Friday | Hawaii |

Pizzas

| pizza | item |
|---|---|
| Capricciosa | base |
| Capricciosa | ham |
| Capricciosa | mushrooms |
| Hawaii | base |
| Hawaii | ham |
| Hawaii | pineapple |

Items

| item | price |
|---|---|
| base | 6 |
| ham | 1 |
| mushrooms | 1 |
| pineapple | 2 |

Consider the natural join of the three relations above:

Orders $\bowtie$ Pizzas $\bowtie$ Items

| customer | day | pizza | item | price |
|---|---|---|---|---|
| Mario | Monday | Capricciosa | base | 6 |
| Mario | Monday | Capricciosa | ham | 1 |
| Mario | Monday | Capricciosa | mushrooms | 1 |
| Mario | Friday | Capricciosa | base | 6 |
| Mario | Friday | Capricciosa | ham | 1 |
| Mario | Friday | Capricciosa | mushrooms | 1 |
| ... | ... | ... | ... | ... |

# Factorized Databases by Example

Orders ⋈ Pizzas ⋈ Items

| customer | day | pizza | item | price |
|---|---|---|---|---|
| Mario | Monday | Capricciosa | base | 6 |
| Mario | Monday | Capricciosa | ham | 1 |
| Mario | Monday | Capricciosa | mushrooms | 1 |
| Mario | Friday | Capricciosa | base | 6 |
| Mario | Friday | Capricciosa | ham | 1 |
| Mario | Friday | Capricciosa | mushrooms | 1 |
| ... | ... | ... | ... | ... |

A *flat* relational algebra expression encoding the above query result is:

$$\begin{array}{lllllllllll}
\langle \mathit{Mario} \rangle & \times & \langle \mathit{Monday} \rangle & \times & \langle \mathit{Capricciosa} \rangle & \times & \langle \text{base} \rangle & \times & \langle 6 \rangle & \cup \\
\langle \mathit{Mario} \rangle & \times & \langle \mathit{Monday} \rangle & \times & \langle \mathit{Capricciosa} \rangle & \times & \langle \text{ham} \rangle & \times & \langle 1 \rangle & \cup \\
\langle \mathit{Mario} \rangle & \times & \langle \mathit{Monday} \rangle & \times & \langle \mathit{Capricciosa} \rangle & \times & \langle \text{mushrooms} \rangle & \times & \langle 1 \rangle & \cup \\
\langle \mathit{Mario} \rangle & \times & \langle \mathit{Friday} \rangle & \times & \langle \mathit{Capricciosa} \rangle & \times & \langle \text{base} \rangle & \times & \langle 6 \rangle & \cup \\
\langle \mathit{Mario} \rangle & \times & \langle \mathit{Friday} \rangle & \times & \langle \mathit{Capricciosa} \rangle & \times & \langle \text{ham} \rangle & \times & \langle 1 \rangle & \cup \\
\langle \mathit{Mario} \rangle & \times & \langle \mathit{Friday} \rangle & \times & \langle \mathit{Capricciosa} \rangle & \times & \langle \text{mushrooms} \rangle & \times & \langle 1 \rangle & \cup \ldots
\end{array}$$

It uses relational product ($\times$), union ($\cup$), and singleton relations (e.g., $\langle 1 \rangle$).

- The attribute names are not shown to avoid clutter.

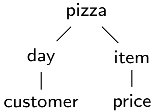
# Factorized Databases by Example

The previous relational expression entails lots of redundancy due to the joins:

$$
\begin{array}{lllllllllll}
\langle \textit{Mario} \rangle & \times & \langle \textit{Monday} \rangle & \times & \langle \textit{Capricciosa} \rangle & \times & \langle \text{base} \rangle & \times & \langle 6 \rangle & \cup \\
\langle \textit{Mario} \rangle & \times & \langle \textit{Monday} \rangle & \times & \langle \textit{Capricciosa} \rangle & \times & \langle \text{ham} \rangle & \times & \langle 1 \rangle & \cup \\
\langle \textit{Mario} \rangle & \times & \langle \textit{Monday} \rangle & \times & \langle \textit{Capricciosa} \rangle & \times & \langle \text{mushrooms} \rangle & \times & \langle 1 \rangle & \cup \\
\langle \textit{Mario} \rangle & \times & \langle \textit{Friday} \rangle & \times & \langle \textit{Capricciosa} \rangle & \times & \langle \text{base} \rangle & \times & \langle 6 \rangle & \cup \\
\langle \textit{Mario} \rangle & \times & \langle \textit{Friday} \rangle & \times & \langle \textit{Capricciosa} \rangle & \times & \langle \text{ham} \rangle & \times & \langle 1 \rangle & \cup \\
\langle \textit{Mario} \rangle & \times & \langle \textit{Friday} \rangle & \times & \langle \textit{Capricciosa} \rangle & \times & \langle \text{mushrooms} \rangle & \times & \langle 1 \rangle & \cup \ldots
\end{array}
$$

We can *factorize* the expression following the join structure, e.g.,:

$$
\begin{aligned}
&\langle \textit{Capricciosa} \rangle \times (\langle \textit{Monday} \rangle \times \langle \textit{Mario} \rangle \cup \langle \textit{Friday} \rangle \times \langle \textit{Mario} \rangle) \\
&\qquad \times (\langle \textit{base} \rangle \times \langle 6 \rangle \cup \langle \textit{ham} \rangle \times \langle 1 \rangle \cup \langle \textit{mushrooms} \rangle \times \langle 1 \rangle) \\
&\cup \langle \textit{Hawaii} \rangle \times \langle \textit{Friday} \rangle \times (\langle \textit{Lucia} \rangle \cup \langle \textit{Pietro} \rangle) \\
&\qquad \times (\langle \textit{base} \rangle \times \langle 6 \rangle \cup \langle \textit{ham} \rangle \times \langle 1 \rangle \cup \langle \textit{pineapple} \rangle \times \langle 2 \rangle)
\end{aligned}
$$




There are several *algebraically equivalent* factorized representations defined by distributivity of product over union and commutativity of product and union.

# Properties of Factorized Representations

Factorized representations of results of queries with select, project, join, aggregate, groupby, and orderby operators:

- **Very high compression rate**
  - Can be exponentially more succinct than the relations they encode.
  - Arbitrarily better than generic compression schemes, e.g., bzip2
  - Factorized representations of asymptotically-tight size bounds computable directly from input database and query

- **Querying in the compressed domain**
  - Factorizations are relational expressions
  - We developed the FDB in-memory query engine for this purpose

- **Constant-delay enumeration of represented tuples**
  - Tuple iteration as fast as listing them from equivalent flat relations

# Outline


What are Factorized Databases?

Applications

A Glimpse at Aggregating Factorized Data

# Spot the Factorized Database!

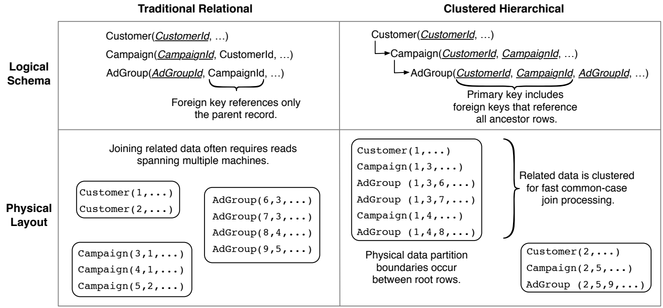



**Figure 2: The logical and physical properties of data storage in a traditional normalized relational schema compared with a clustered hierarchical schema used in an F1 database.**

Excerpt from *F1: A Distributed SQL Database That Scales*. PVLDB'13.

- Google's DB supporting their lucrative AdWords business
- Database factorization increases data locality for common access patterns
  - Tables pre-joined using a nesting structure defined by key-fkey constraints
- Data partitioned across servers into factorization fragments

# Spot the Factorized Database!

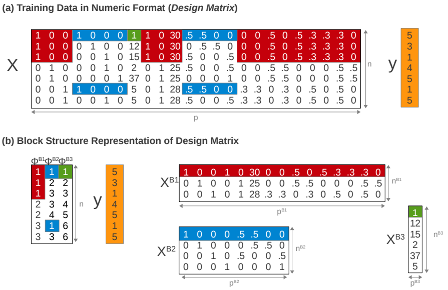
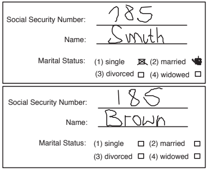



**Figure 3: (a) In relational domains, design matrices $X$ have large blocks of repeating patterns (example from Figure 2). (b) Repeating patterns in $X$ can be formalized by a block notation (see section 2.3) which stems directly from the relational structure of the original data. Machine learning methods have to make use of repeating patterns in $X$ to scale to large relational datasets.**

Excerpt from *Scaling Factorization Machines to Relational Data.* PVLDB'13.

- Feature vectors for predictive modelling represented as very large design matrices (= relations with high cardinality)
- Standard learning algorithms cannot scale on design matrix representation
- Use repeating patterns in the design matrix as key to scalability

# Spot the Factorized Database!




| $t_1.S$ | $t_1.N$ | $t_1.M$ | $t_2.S$ | $t_2.N$ | $t_2.M$ |
|---|---|---|---|---|---|
| 185 | Smith | 1 | 186 | Brown | 1 |
| 185 | Smith | 1 | 186 | Brown | 2 |
| 185 | Smith | 1 | 186 | Brown | 3 |
| 185 | Smith | 1 | 186 | Brown | 4 |
| 185 | Smith | 2 | 186 | Brown | 1 |
| 185 | Smith | 2 | 186 | Brown | 2 |
| 185 | Smith | 2 | 186 | Brown | 3 |
| 185 | Smith | 2 | 186 | Brown | 4 |
| | | ⋮ | | | |
| 785 | Smith | 2 | 186 | Brown | 4 |

**Fig. 1.** Two completed survey forms and a world-set relation representing the possible worlds with unique social security numbers.

| $t_1.S$ | $t_2.S$ |
|---|---|
| 185 | 186 |
| 785 | 185 |
| 785 | 186 |

$\times$

| $t_1.N$ |
|---|
| Smith |

$\times$

| $t_1.M$ |
|---|
| 1 |
| 2 |

$\times$

| $t_2.N$ |
|---|
| Brown |

$\times$

| $t_2.M$ |
|---|
| 1 |
| 2 |
| 3 |
| 4 |

Excerpt from $10^{10^6}$ *Worlds and Beyond: Efficient Representation and Processing of Incomplete Information.* ICDE'07.

Managing a large set of possibilities or choices:

- Configuration problems (space of valid solutions)
- Incomplete information (space of possible worlds)

# Spot the Factorized Database!



### 5.1.3 READ-ONCE FORMULAS

An important class of propositional formulas that play a special role in probabilistic databases are read-once formulas. We restrict our discussion to the case when all random variables $X$ are Boolean variables.

$\Phi$ is called *read-once* if there is a formula $\Phi'$ equivalent to $\Phi$ such that every variable occurs at most once in $\Phi'$. For example:

$$\Phi = X_1 Y_1 \vee X_1 Y_2 \vee X_2 Y_3 \vee X_2 Y_4 \vee X_2 Y_5$$

is read-once because it is equivalent to the following formula:

$$\Phi' = X_1(Y_1 \vee Y_2) \vee X_2(Y_3 \vee Y_4 \vee Y_5)$$

Excerpt from *Probabilistic Databases.* Morgan & Claypool. 2011.

Provenance and probabilistic data:

- Compact encoding for large provenance
- Factorization of provenance is used for efficient query evaluation in probabilistic databases

# Outline


What are Factorized Databases?

Applications

A Glimpse at Aggregating Factorized Data

# Aggregating Factorized Data

We only present here COUNT and SUM aggregation functions.

$\mathrm{COUNT}(F)$ is the number of tuples in a factorization $F$:

- $\mathrm{COUNT}(\langle a \rangle) = 1$.
- $\mathrm{COUNT}(F_1 \cup \dots \cup F_k) = \mathrm{COUNT}(F_1) + \ldots + \mathrm{COUNT}(F_k)$.
- $\mathrm{COUNT}(F_1 \times \dots \times F_k) = \mathrm{COUNT}(F_1) \cdot \ldots \cdot \mathrm{COUNT}(F_k)$.

$\mathrm{SUM}_A(F)$ is the sum of all values of attribute $A$ in a factorization $F$:

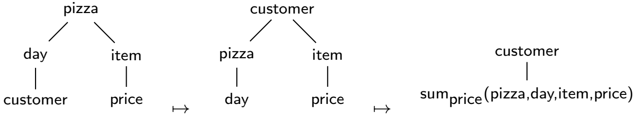
- $\mathrm{SUM}_A(\langle a \rangle) = a$, if the singleton $\langle a \rangle$ has attribute $A$.
- $\mathrm{SUM}_A(F_1 \cup \dots \cup F_k) = \mathrm{SUM}_A(F_1) + \ldots + \mathrm{SUM}_A(F_k)$.
- $\mathrm{SUM}_A(F_1 \times \dots \times F_k) = \mathrm{SUM}_A(F_1) \cdot \mathrm{COUNT}(F_2) \cdot \ldots \cdot \mathrm{COUNT}(F_k)$, where wlog values for attribute $A$ are in expression $F_1$.

# Aggregation by Example

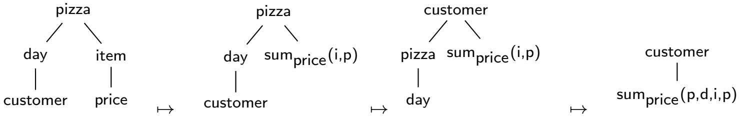
- Recall the natural join of Orders, Pizzas, and Items
- We would like to find the overall sales per customer
- Assume the factorization structure discussed before (leftmost below)

Examplea of possible evaluation plans:

1. First restructure for GROUP-BY, then aggregate

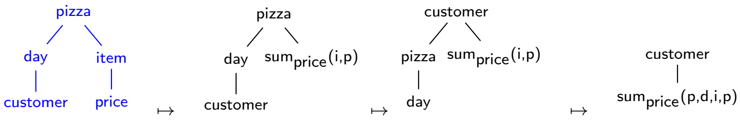



2. Intertwine restructuring for GROUP-BY and partial aggregation



# Query Evaluation Step by Step

Let us consider the second evaluation plan:

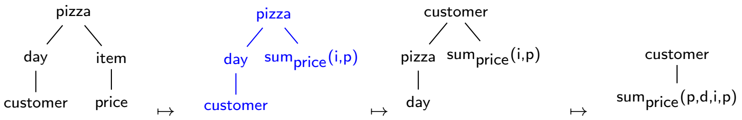



The initial factorization with the structure highlighted above:

$$
\begin{aligned}
&\langle \mathit{Capricciosa} \rangle \times (\langle \mathit{Monday} \rangle \times \langle \mathit{Mario} \rangle \;\cup\; \langle \mathit{Friday} \rangle \times \langle \mathit{Mario} \rangle) \\
&\qquad \times (\langle \mathit{base} \rangle \times \langle 6 \rangle \;\cup\; \langle \mathit{ham} \rangle \times \langle 1 \rangle \;\cup\; \langle \mathit{mushrooms} \rangle \times \langle 1 \rangle) \\
\cup\; &\langle \mathit{Hawaii} \rangle \times \langle \mathit{Friday} \rangle \times (\langle \mathit{Lucia} \rangle \cup \langle \mathit{Pietro} \rangle) \\
&\qquad \times (\langle \mathit{base} \rangle \times \langle 6 \rangle \;\cup\; \langle \mathit{ham} \rangle \times \langle 1 \rangle \;\cup\; \langle \mathit{pineapple} \rangle \times \langle 2 \rangle)
\end{aligned}
$$

# Query Evaluation Step by Step

Let us consider the second evaluation plan:

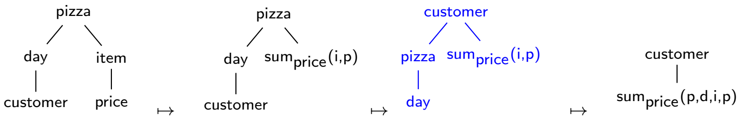



The factorization after partial aggregation with the structure highlighted above:

$$
\begin{aligned}
&\langle \mathit{Capricciosa} \rangle \times (\langle \mathit{Monday} \rangle \times \langle \mathit{Mario} \rangle \cup \langle \mathit{Friday} \rangle \times \langle \mathit{Mario} \rangle) \\
&\qquad \times \langle 8 \rangle \\
\cup\ &\langle \mathit{Hawaii} \rangle \times \langle \mathit{Friday} \rangle \times (\langle \mathit{Lucia} \rangle \cup \langle \mathit{Pietro} \rangle) \\
&\qquad \times \langle 9 \rangle
\end{aligned}
$$

# Query Evaluation Step by Step

Let us consider the second evaluation plan:

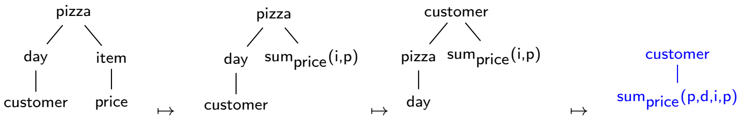



The factorization after restructuring with the structure highlighted above:

$$\langle \textit{Lucia} \rangle \times \langle \textit{Hawaii} \rangle \times \langle \textit{Friday} \rangle \times \langle 9 \rangle \cup$$
$$\langle \textit{Mario} \rangle \times \langle \textit{Capricciosa} \rangle \times (\langle \textit{Monday} \rangle \cup \langle \textit{Friday} \rangle) \times \langle 8 \rangle \cup$$
$$\langle \textit{Pietro} \rangle \times \langle \textit{Hawaii} \rangle \times \langle \textit{Friday} \rangle \times \langle 9 \rangle$$

# Query Evaluation Step by Step

Let us consider the second evaluation plan:




The factorization after final aggregation with the structure highlighted above:

$$\langle \mathit{Lucia} \rangle \times \langle 9 \rangle \cup$$
$$\langle \mathit{Mario} \rangle \times \langle 16 \rangle \cup$$
$$\langle \mathit{Pietro} \rangle \times \langle 9 \rangle$$

# Thank you!