Optimization of Preference Queries with Multiple Constraints

Markus Endres and Werner Kießling
Speaker: Timotheus Preisinger

University of Augsburg
Germany

2nd International Workshop on Personalized Access, Profile Management, and Context Awareness: Databases
Outline

1 Motivation

2 Preference Algebra

3 Optimization Techniques
   - Dominance Criterion
   - Selection Operators
   - Rewriting Technique

4 Experimental Results

5 Conclusion
Motivation

Diet sheet

Multiple Hard constraints

- nutritional requirements
- restriction on calories ($\leq 1100$ kcal)
- restriction on vitamin C ($\geq 38$ g)
- restriction on fat ($\leq 9$ g)

Soft constraints - Preferences

- Chicken soup as starter
- Main course should be beef
- Beef with lowest cholesterol
- Red wine for beverage, less important
Motivation
Diet Sheet - Database Query

Database Query

- Query on USDA Database to retrieve diet sheet www.nal.usda.gov/fnic
- More than 7000 types of food
- Relations for Soups, Meats, Beverages

- Must consider hard and soft constraints
- Using Preference SQL Kießling & Köstler 2002
- Combining hard and soft constraints
Motivation
Diet Sheet - Preference SQL

**Preference SQL**  Kießling & Köstler 2002

```
SELECT S.name, M.name, B.name
FROM Soups S, Meats M, Beverages B
WHERE
  S.cal + M.cal + B.cal <= 1100
  AND S.Vc + M.Vc + B.Vc >= 38
  AND S.fat + M.fat + B.fat <= 9
  multiple hard constraint
PREFERRING
  S.name IN ('Chicken soup') AND
  (M.name IN ('Beef')
    AND M.Cholesterol LOWEST)
  PRIOR TO B.name IN ('Red wine')
```

**Hard and Soft Constraints**

- SQL query with multiple hard constraints
- extended by soft constraints - PREFERING
Operator Tree
Transform Preference SQL into Operator Tree

Preference SQL

```
SELECT S.name, M.name, B.name
FROM Soups sS, Meats M, Beverages B
WHERE S.cal + M.cal + B.cal <= 1100
    AND S.Vc + M.Vc + B.Vc >= 38
    AND S.fat + M.fat + B.fat <= 9
PREFERRING
    S.name IN ('Chicken soup') AND
(M.name IN ('Beef')
    AND M.Cholesterol LOWEST)
PRIOR TO B.name IN ('Red wine')
```
## Problem Description

### High memory and computation costs

- **State-of-the-art:**
  - Join algorithms, but without preferences  
    - Ilyas, Liu
  - Optimization for one constraint with preferences  
    - Endres, Kießling
- **Full cartesian product necessary**
- **Pair-wise operators can’t remove intermediate results**
- **Hard constraints include different relations**

### Diagram

```
/ \               _______________Soft Constraints
|   |              |                  22.000.000
|   |  Hard Constraints                |                  87.000.000
|   |                                 |                    | 500
|   |                                 |                    Soups
|   |                                 |                      ▼
|   |                                 | 700                  Meats
|   |                                 |                      ▼
|   |                                 | 250                  Beverages
```

Optimization of Preference Queries under Hard Sum Constraints
Problem Description

Target

Reduce memory and computation costs for Preference Queries with Multiple Constraints
Outline

1 Motivation
2 Preference Algebra
3 Optimization Techniques
   - Dominance Criterion
   - Selection Operators
   - Rewriting Technique
4 Experimental Results
5 Conclusion

Formal description necessary
Preference Algebra  Kießling 2002

Preference Model Necessary

Preference is a

- strict partial order
- \( P = (A, <_P), <_P \subseteq \text{dom}(A) \times \text{dom}(A) \)
- \( x <_P y \) means "I like y more than x"

Preference constructors

- Numerical: Lowest, Highest, Around, ...
- Categorical: POS, NEG, ...
- Complex preferences
  - Pareto: \( P_1 \otimes P_2 \)
    \( P_1 \) and \( P_2 \) are equally important
  - Prioritized: \( P_1 \& P_2 \)
    \( P_1 \) is more important than \( P_2 \)
The BMO Query Model

Compute optimal outcome with respect to preference statements.

**Best-Matches-Only**

- only the best matches
- w.r.t the strict partial order of a preference $P$
- all tuples in BMO are undominated by others regarding $P$

**preference selection** (Chomicki’s Winnow-Operator)

$$
\sigma[P](R) := \{ t \in R \mid \neg \exists t' \in R : t[A] <_P t'[A] \}
$$
Preference Model

Example

### Preference SQL

```sql
SELECT S.name, M.name, B.name
FROM Soups S, Meats M, Beverages B
WHERE S.cal + M.cal + B.cal <= 1100
  AND S.Vc + M.Vc + B.Vc >= 38
  AND S.fat + M.fat + B.fat <= 9
PREFERING
  S.name IN ('Chicken soup') AND
  (M.name IN ('Beef') AND
   AND M.Cholesterol LOWEST)
  PRIOR TO B.name IN ('Red wine')
```

### Preference Algebra

\[
\sigma[P_1 \otimes P_2 \& P_3]
\]

\[
\sigma S.cal + M.cal + B.cal \leq 1100 \land
S.Vc + M.Vc + B.Vc \geq 38 \land
S.fat + M.fat + B.fat \leq 9
\]

\((S \times M \times B)\)

with

- \(P_1 = \text{POS}(S.name, \{\text{Chicken soup}\})\)
- \(P_2 = \text{POS}(M.name, \{\text{Beef}\}) \otimes \text{LOWEST}(\text{cholesterol})\)
- \(P_3 = \text{POS}(B.name, \{\text{Red wine}\})\)
Outline

1. Motivation
2. Preference Algebra
3. Optimization Techniques
   - Dominance Criterion
   - Selection Operators
   - Rewriting Technique
4. Experimental Results
5. Conclusion

Extension of MPref 2008 Paper
A Dominance Criterion

Example

Preference SQL

```
SELECT S.name, M.name, B.name
FROM Soups S, Meats M, Beverages B
WHERE S.cal + M.cal + B.cal <= 1100
    AND S.Vc + M.Vc + B.Vc >= 38
    AND S.fat + M.fat + B.fat <= 9
PREFERRING
    S.name IN ('Chicken soup') AND ...
```

Dominance Criterion

```
S4 <_{P_1} S3 \land S4.cal \geq S3.cal \land S4.Vc \leq S3.Vc \land S4.fat \geq S3.fat
```

<table>
<thead>
<tr>
<th>Soups</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td>Vegetable</td>
<td>59</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td>Chicken</td>
<td>110</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td>Chicken</td>
<td>198</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td>Noodle</td>
<td>453</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Optimization of Preference Queries under Hard Sum Constraints
A Dominance Criterion

Definition

**Query**

\[
Q := \sigma[P_1 \Phi ... \Phi P_r] \sigma F_1 \land ... \land F_n (R_1 \times ... \times R_r)
\]

\(R_i(A_{i_1}, ..., A_{i_n}, B_i), P_i = (B_i, <_{P_i}), \Phi \in \{\otimes, \&\}, F_i \) hard constraints with \(\Theta \in \{<, \leq, >, \geq, =, \neq\}\).

If tuples \(t, t' \in R_i\) such that

\[
t[B_i] <_{P_i} t'[B_i] \land t[A_{i_1}] \Theta_1 t'[A_{i_1}] \land ... \land t[A_{i_r}] \Theta_r t'[A_{i_r}]
\]

and \(\hat{\Theta}_j\) defined as

\[
\hat{\Theta}_j := \begin{cases} 
  \geq & \text{iff } \Theta_j \in \{\leq, <\} \\
  \leq & \text{iff } \Theta_j \in \{\geq, >\} \\
  = & \text{iff } \Theta_j \in \{=, \neq\}
\end{cases}
\]

then an optimal solution exists **without** the tuple \(t \in R_i\).
A Dominance Criterion

Definition

Query

\[ Q := \sigma[\Phi_1 \ldots \Phi_r] \sigma_{F_1} \land \ldots \land F_n(R_1 \times \ldots \times R_r) \]

\( R_i(A_{i1}, \ldots, A_{in}, B_i), P_i = (B_i, <_{P_i}), \Phi \in \{\otimes, \&\}, F_i \) hard constraints with \( \Theta \in \{<, \leq, >, \geq, =, \neq\} \).

If tuples \( t, t' \in R_i \) such that

\[ t[B_i] <_{P_i} t'[B_i] \land t[A_{i1}] \atop \hat{\Theta}_1 t'[A_{i1}] \land \ldots \land \]

\[ t[A_{ir}] \atop \hat{\Theta}_r t'[A_{ir}] \]

and \( \hat{\Theta}_j \) defined as

\[ \hat{\Theta}_j := \begin{cases} \geq & \text{iff } \Theta_j \in \{\leq, <\} \\ \leq & \text{iff } \Theta_j \in \{\geq, >\} \\ = & \text{iff } \Theta_j \in \{=, \neq\} \end{cases} \]

then an optimal solution exists without the tuple \( t \in R_i \).

**CUTOFF** Preference Constructor  Strict partial order

\( P_c := CUTOFF(P) \) : Evaluation of the Dominance Criterion
Outline

1. Motivation
2. Preference Algebra
3. Optimization Techniques
   - Dominance Criterion
   - Selection Operators
   - Rewriting Technique
4. Experimental Results
5. Conclusion

Additional hard selection operators
## Selection Operators

### Domain Knowledge

### Constraints:

- \( S\text{.cal} + M\text{.cal} + B\text{.cal} \leq 1100 \text{ kcal} \)
- \( S\text{.Vc} + M\text{.Vc} + B\text{.Vc} \geq 38 \text{ g} \)
- \( S\text{.fat} + M\text{.fat} + B\text{.fat} \leq 9 \text{ g} \)

### Introduce NEW Constraints

- \( S\text{.Cal} + M\text{.Cal} + B\text{.Cal} \leq 1100 \)
- \( S\text{.Cal} \leq 1100 - (M\text{.Cal} + B\text{.Cal}) \)
- A tuple with

  \[
  S\text{.Cal} \geq 1100 - (\min(M\text{.Cal}) + \min(B\text{.Cal}))
  \]

  can’t fulfill the hard constraint
Selection Operators

Example

**Constraint**

\[ S.Cal + M.Cal + B.Cal \leq 1100 \]

<table>
<thead>
<tr>
<th>Soups</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td>Vegetable</td>
<td>59</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td>Chicken</td>
<td>110</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td>Chicken</td>
<td>198</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td>Noodle</td>
<td>453</td>
<td>1</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meats</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
<th>Chol</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td></td>
<td>Turkey</td>
<td>818</td>
<td>13</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>M2</td>
<td></td>
<td>Beef</td>
<td>857</td>
<td>14</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>M3</td>
<td></td>
<td>Pork</td>
<td>911</td>
<td>12</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beverages</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td></td>
<td>Red Wine</td>
<td>85</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B2</td>
<td></td>
<td>Red Wine</td>
<td>181</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>B3</td>
<td></td>
<td>Coke</td>
<td>220</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>B4</td>
<td></td>
<td>Lemonade</td>
<td>281</td>
<td>17</td>
<td>2</td>
</tr>
<tr>
<td>B5</td>
<td></td>
<td>Red Wine</td>
<td>400</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
Selection Operators

Example

Constraint

\[ S.Cal + M.Cal + B.Cal \leq 1100 \]
\[ \Rightarrow \quad \min(M.Cal) + \min(B.Cal) = 903 \Rightarrow S.Cal \leq 197 \]

<table>
<thead>
<tr>
<th>Soups</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S1</td>
<td>Vegetable</td>
<td>59</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>Chicken</td>
<td>110</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>Chicken</td>
<td>198</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>Noodle</td>
<td>453</td>
<td>7</td>
<td>17</td>
</tr>
</tbody>
</table>

S.Cal > 197

<table>
<thead>
<tr>
<th>Meats</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
<th>Chol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M1</td>
<td>Turkey</td>
<td>818</td>
<td>13</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>Beef</td>
<td>857</td>
<td>14</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>Pork</td>
<td>911</td>
<td>12</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beverages</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B1</td>
<td>Red Wine</td>
<td>85</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>Red Wine</td>
<td>181</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>B3</td>
<td>Coke</td>
<td>220</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>B4</td>
<td>Lemonade</td>
<td>281</td>
<td>17</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>B5</td>
<td>Red Wine</td>
<td>400</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
Selection Operators

Example

Constraint

\[ S.Cal + M.Cal + B.Cal \leq 1100 \]

\[ \Rightarrow \ min(S.Cal) + min(B.Cal) = 144 \Rightarrow M.Cal \leq 956 \]

<table>
<thead>
<tr>
<th>Soups</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S1</td>
<td>Vegetable</td>
<td>59</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>Chicken</td>
<td>110</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>Chicken</td>
<td>198</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>S4</td>
<td>Noodle</td>
<td>453</td>
<td>7</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meats</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
<th>Chol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M1</td>
<td>Turkey</td>
<td>818</td>
<td>13</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>Beef</td>
<td>857</td>
<td>14</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>Pork</td>
<td>911</td>
<td>12</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beverages</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B1</td>
<td>Red Wine</td>
<td>85</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>Red Wine</td>
<td>181</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>B3</td>
<td>Coke</td>
<td>220</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>B4</td>
<td>Lemonade</td>
<td>281</td>
<td>17</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>B5</td>
<td>Red Wine</td>
<td>400</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
Selection Operators

Example

Constraint

\[ S.\text{Cal} + M.\text{Cal} + B.\text{Cal} \leq 1100 \]

\[ \Rightarrow \ min(S.\text{Cal}) + min(M.\text{Cal}) = 877 \Rightarrow B.\text{Cal} \leq 223 \]

<table>
<thead>
<tr>
<th>Soups</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td>Vegetable</td>
<td>59</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td>Chicken</td>
<td>110</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td>Chicken</td>
<td>198</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td>Noodle</td>
<td>453</td>
<td>7</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meats</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
<th>Chol</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td></td>
<td>Turkey</td>
<td>818</td>
<td>13</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>M2</td>
<td></td>
<td>Beef</td>
<td>857</td>
<td>14</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>M3</td>
<td></td>
<td>Pork</td>
<td>911</td>
<td>12</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beverages</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td></td>
<td>Red Wine</td>
<td>85</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B2</td>
<td></td>
<td>Red Wine</td>
<td>181</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>B3</td>
<td></td>
<td>Coke</td>
<td>220</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>B4</td>
<td></td>
<td>Lemonade</td>
<td>281</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>B5</td>
<td></td>
<td>Red Wine</td>
<td>400</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

Optimization of Preference Queries under Hard Sum Constraints
### Selection Operators

#### Example

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Soups</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. Vc &lt; 3</td>
<td></td>
<td>S1</td>
<td>Vegetable</td>
<td>59</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2</td>
<td>Chicken</td>
<td>110</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>S. Cal &gt; 197</td>
<td></td>
<td>S3</td>
<td>Chicken</td>
<td>198</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>DC, S. Cal &gt; 197</td>
<td></td>
<td>S4</td>
<td>Noodle</td>
<td>453</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meats</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
<th>Chol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M1</td>
<td>Turkey</td>
<td>818</td>
<td>13</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>Beef</td>
<td>857</td>
<td>14</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>Pork</td>
<td>911</td>
<td>12</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beverages</th>
<th>ID</th>
<th>Name</th>
<th>Cal</th>
<th>Vc</th>
<th>Fat</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. Vc &lt; 12</td>
<td>B1</td>
<td>Red Wine</td>
<td>85</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>Red Wine</td>
<td>181</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>B3</td>
<td>Coke</td>
<td>220</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>B. Cal &gt; 223</td>
<td>B4</td>
<td>Lemonade</td>
<td>281</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>B5</td>
<td>Red Wine</td>
<td>400</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
Outline

1. Motivation
2. Preference Algebra
3. Optimization Techniques
   - Dominance Criterion
   - Selection Operators
   - Rewriting Technique
4. Experimental Results
5. Conclusion

Query optimizer integration
Rewriting Technique

Operator Tree

\[ \sigma \left[ P_1 \Phi \ldots \Phi P_r \right] \]

\[ \downarrow \]

\[ \sigma \Phi_1 \land \ldots \land \Phi_r \]

\[ \downarrow \]

\[ \times \]

\[ \times \]

\[ R_r \]

\[ R_1 \]

\[ R_2 \]

Query Transformation

1)

2)
Rewriting Technique

Operator Tree

\[\sigma[P_1 \Phi ... \Phi P_r]\]
\[\sigma F_1 \land ... \land F_r\]
\[\sigma A_r \land C_r \land b_r\]

Query Transformation

1) Get lower and upper bounds and insert additional selection operators
Rewriting Technique

Operator Tree

\[ \sigma[P_1 \Phi \ldots \Phi P_r] \]
\[ \downarrow \]
\[ \sigma_{F_1} \land \ldots \land F_r \]
\[ \downarrow \]
\[ \times \]
\[ \sigma[P_{r_c}] \]
\[ \downarrow \]
\[ \sigma[A_r \Theta_r b_r] \]
\[ \downarrow \]
\[ R_r \]

Query Transformation

1) Get lower and upper bounds and insert additional selection operators

2) Apply dominance criterion
Outline

1. Motivation

2. Preference Algebra

3. Optimization Techniques
   - Dominance Criterion
   - Selection Operators
   - Rewriting Technique

4. Experimental Results

5. Conclusion
Experimental Results
USDA - National Nutrient Database for Standard Reference

USDA - A Real-World Database

- More than 7000 types of food
- 3 relations: 500 soups - 700 meats - 250 beverages
- about 87,500,000 possible combinations
- Preference SQL engine (Java)
- Integrated Preference Optimizer
- With and without rewriting
Experimental Results

Different Hard Constraints

Test query

```
SELECT S.name, M.name, B.name
FROM
Soups S, Meats M, Beverages B
WHERE
    S.cal + M.cal + B.cal <= max_cal
    AND S.Vc + M.Vc + B.Vc >= min_vc
    AND S.fat + M.fat + B.fat <= max_fat
PREFERING
    S.name IN ('Chicken soup') AND
    (M.name IN ('Beef')
        AND M.Cholesterol LOWEST)
    PRIOR TO B.name IN ('Red wine')
```
Experimental Results
Different Hard Constraints

Test query

SELECT S.name, M.name, B.name
FROM
Soups S, Meats M, Beverages B
WHERE
S.cal + M.cal + B.cal <= max_cal
AND S.Vc + M.Vc + B.Vc >= min_vc
AND S.fat + M.fat + B.fat <= max_fat
PREFERRING
S.name IN ('Chicken soup') AND
(M.name IN ('Beef')
AND M.Cholesterol LOWEST)
PRIOR TO B.name IN ('Red wine')
Experimental Results
Different Relation Sizes

Test query

```
SELECT S.name, M.name, B.name 
FROM 
Soups S, Meats M, Beverages B 
WHERE 
  S.cal + M.cal + B.cal <= 1100 
  AND S.Vc + M.Vc + B.Vc >= 38 
  AND S.fat + M.fat + B.fat <= 9 
PREFERING 
  S.name IN ('Chicken soup') AND 
  (M.name IN ('Beef') 
   AND M.Cholesterol LOWEST) 
  PRIOR TO B.name IN ('Red wine')
```
Experimental Results

Different Relation Sizes

Test query

SELECT S.name, M.name, B.name
FROM
Soups S, Meats M, Beverages B
WHERE
  S.cal + M.cal + B.cal <= 1100
  AND S.Vc + M.Vc + B.Vc >= 38
  AND S.fat + M.fat + B.fat <= 9
PREFERRING
  S.name IN ('Chicken soup') AND
  (M.name IN ('Beef')
    AND M.Cholesterol LOWEST)
  PRIOR TO B.name IN ('Red wine')
Outline

1 Motivation

2 Preference Algebra

3 Optimization Techniques
   - Dominance Criterion
   - Selection Operators
   - Rewriting Technique

4 Experimental Results

5 Conclusion
# Summary

- Preference queries with multiple hard constraints
- Diet sheet, tourism (overall price), planning tasks, ...
- Optimization based on dominance criterion
- Additional selection operators
- Experimental results on USDA real-world DB

---

# Outlook

- Consider further optimization techniques
- Develop special join algorithms to avoid complete cartesian product
Questions?

{endres, kiessling, preisinger}@informatik.uni-augsburg.de