Personalization of Queries in Database Systems

Georgia Koutrika, Yannis Ioannidis

Department of Informatics and Telecommunications
University of Athens, Hellas

Presented by: Linas Baltrunas and Tenzin Choesang
1 Introduction
   - Authors
   - Summary of the Approach
   - Related Approaches

2 Preference Selection
   - Preference Selection Algorithm
   - Preference Integration

3 Experimental Results
Yannis Ioannidis. University of Athens

- **Interests**
  - Query Optimization in Database and Information Systems
  - Heterogeneous Systems
  - Human-Computer Interaction Database User Interfaces

- **Known as one of the key researcher in Histograms**

- **CV**
  - Department of Informatics, Univ. of Athens, Hellas (Associate Professor)
  - Computer Sciences Department, Univ. of Wisconsin (Associate Professor)
  - 1986 Computer Science, Univ. of California, Berkeley
  - 1983 Applied Mathematics (Computer Science), Harvard University
Georgia Koutrika

- **Interests**
  - various aspects of personalization
  - user profiling
  - data mining

- Ph.D. student of Ioannidis
Motivation of Query Personalization

Goal: to personalize the query

Which movie is shown tonight?

select MV.title
from MOVIE MV, PLAY PL
where MV.mid=PL.mid and PL.date='2/7/2003'

Julie
select MV.title
from MOVIE MV, PLAY PL, GENRE GN
MV.mid=PL.mid and PL.date='2/7/2003' and
MV.mid=GN.mid and (GN.genre='comedy' or
GN.genre='thriller')

Rob
select MV.title
from MOVIE MV, PLAY PL, GENRE GN, CAST CA, ACTOR AC
MV.mid=PL.mid and PL.date='2/7/2003' and
MV.mid=GN.mid and MV.mid=CA.mid and
CA.mid=AC.mid and (GN.genre='sci-fi' or
AC.name='J. Roberts')
Summary through Architecture

User input -> Profile Creation

Profile Creation -> User Profiles

User input -> Query Personalization

Query Personalization -> Content Access

Content Access -> Data

Data -> Presentation Personalization

Presentation Personalization -> Results

Results
Related Approaches

Content Based Approaches

- **Query-based approaches**: Select content on the basis of query
- **Filter-based approaches**: Select content from stored user profile
- **Personalized approaches**: Select content based on user query and preferences stored in profile
**User Preference Model**

- Director.name = "W. Allen"

**Preference on one imply preference on the other**
- Movie.mid = Directed.mid and Directed.did=Director.did and Director.name="W. Allen"
Atomic User Preferences
Personalization Graph G(V,E)

**Nodes (G):**
- Relation Nodes
- Attribute Nodes
- Value Nodes

**Edges (E):**
- Selection Edges
- Join Edges
Atomic User Preferences

THEATER.tid=PLAY.tid, 1
THEATER.REGION="DOWNTOWN", 0.8
PLAY.tid=THEATER.tid, 1
PLAY.mid=MOVIE.mid, 1
MOVIE.mid=PLAY.mid, 0.8
MOVIE.mid=GENRE.mid, 0.9
MOVIE.mid=DIRECTED.mid, 1
MOVIE.mid=CAST.mid, 0.8
ACTOR.name="HOPKINS", 0.8
ACTOR.name="KIDMAN", 0.9
ACTOR.name="ROSSELLINI", 0.6
GENRE.genre="COMEDY", 0.9
GENRE.genre="THRILLER", 0.7
GENRE.genre="ADVENTURE", 0.5
DIRECTOR.name="ALLEN", 0.7
DIRECTOR.name="LYNCH", 0.9
DIRECTED.did=DIRECTOR.did, 1
DIRECTOR.did=DIRECTED.did, 1
CAST.AID=ACTOR.AID, 1
Implicit/Transitive User Preferences

- **Transitive Join**: Between two attribute nodes
- **Transitive Selection**: From an attribute node to a value node. It can be decomposed into a transitive join and an atomic selection.

Transitive query element is defined as the conjunction of the constituent atomic ones
Implicit/Transitive User Preferences

**Transitive Preference**

- $D_N = d_i | d_i$: degree of interest in $P_i \in P_N$, i=1 . . . N
- $f \otimes (D_N) \leq \min(D_N)$.
- $f \otimes (D_N) = d_1 d_2 . . . d_N$.

Where, $P_N$ is a set of N composable atomic preferences $D_N$ is corresponding degree of interest

**Note:** The degree of interest in a transitive preference decreases as the length of the corresponding directed path increases.
Example

- Julie likes actress N. Kidman \( \text{Actor.name} = "N. Kidman" \) 0.9
- then, she also likes movies starring the same actress, which is implicitly expressed as

\[
\text{MOVIE.mid} = \text{CAST.mid} \quad \text{and} \\
\text{CAST.aid} = \text{ACTOR.aid} \quad \text{and} \\
\text{ACTOR.name} = "N.Kidman"
\]

Degree of interest on transitive preference is
\(0.8 \times 1 \times 0.9 = 0.72\)
Logical Combination of User Preferences

**Conjunctive Preferences:** \( f \wedge (D_N) \geq \max(D_N) \). \( f \wedge (D_N) = 1-(1-d_1)(1-d_2)\ldots(1-d_N) \) The degree of interest in multiple preferences satisfied together increases with the number of these preferences.

**Disjunctive Preferences:** \( \min(D_N) \leq f \vee (D_N) \leq \max(D_N) \) \( f \vee (D_N) = (d_1+d_2+\ldots d_N)/N \) The degree of interest in satisfying one of several preference is between the highest and the lowest degree of interest among the original preferences.
Example: transitive selection on Julies profile

MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did
AND DIRECTOR.name=”W. Allen”
MOVIE.mid=GENRE.mid and GENRE.genre=”comedy”

Comedies directed by W. Allen is
\[1-(1-1*1*0.7)(1-0.9*0.9) = 0.943\]

Either a comedy or a W. Allen movie is \((0.7 + 0.81)/2 = 0.755\)
Theorem

Let $\Omega$: Set of all conditions that represent logical combinations of preferences in $P_N$ according to the personalization model.

- For any $\omega_1, \omega_2$ in $\Omega$ with degrees of interest $d_1$ and $d_2$, if $\text{Result}(\omega_1) \subseteq \text{Result}(\omega_2)$, then $d_1 \geq d_2$.
- This theorem captures the intuition that strictly smaller query answers are of strictly higher interest to the user.
## Preference Selection

### Parameters for preference selection
- Top K preferences derived from user profile (Defined by CI)
- No. Of M preferences that are mandatory constrains \(0 \leq M \leq K\)
- No. Of L preferences that should at least be met by the results.

### Extracted preferences has following properties
- It is related to a query
- It is not conflicting with a query
SELECT * FROM MOVIE MV, PLAY PL WHERE MV.MID = PL.MID

Query graph is connected
Preference is syntactically related to a query, if it maps to a path that is attached to the query graph.
Preference is syntactically related to a query, if it maps to a path that is attached to the query graph.

MOVIE.MID=GENRE.MID AND GENRE.genre="COMEDY"
Syntactically Conflicting Preferences

- Preference is syntactically conflicting with a query if it is conflicting with a condition already there.
- Two conditions are conflicting if there is selection condition on the attribute and preference gives different condition on the same attribute.
- Example: THEATER.REGION='downtown' conflicts with THEATER.REGION='uptown'
Preference Selection Algorithm (Idea)

- Take personalization graph and query
- Gradually construct directed paths in decreasing order of their degree of interest
  - Paths starts from query graph (are syntactically related)
  - Expands outwards query subgraph
Preference Selection Algorithm (Example)

We take full user personalization graph
Preference Selection Algorithm (Example)

- Construct example query as subgraph of personalization graph
- SELECT mv.title FROM movie, play WHERE movie.mid=pl.mi AND play.date=”2006-12-11”;

Georgia Koutrika, Yannis Ioannidis ( ) Personalization of Queries UNIBZ, December 11, 2006 23 / 38
First step of algorithm

- For every atom property check if it is related and not conflicting with query
First step of algorithm

- For every atom property check if it is related and not conflicting with query
- \text{THEATER.tid=} \text{PLAY.tid}, 1
- Not related. Remember the order/directions matters!
Preference Selection Algorithm (Example)

First step of algorithm
- For every atom property check if it is related and not conflicting with query
  - MOVIE.mid=GENRE.mid, 0.9
  - Related. Adding To QP

QP={MOVIE.mid=GENRE.mid, 0.9}
First step of algorithm

- For every atom property check if it is related and not conflicting with query
  - `GENRE.genre="THRILER", 0.7`
  - Not related. Does not start from query graph

\[ QP=\{ \text{MOVIE.mid=} \text{GENRE.mid}, 0.9 \} \]
Preference Selection Algorithm (Example)

First step of algorithm

- For every atom property check if it is related and not conflicting with query
  - MOVIE.mid = DIRECTED.mid, 1
  - Related. Added to the QP
  - QP is always ordered by interest

QP = \{ MOVIE.mid = DIRECTED.mid, 1 \}
\{ MOVIE.mid = GENRE.mid, 0.9 \}
Preference Selection Algorithm (Example)

Ordered list $QP$ after first step of algorithm

- $PLAY.tid = THEATRE.tid$, 1
- $MOVIE.mid = DIRECTED.mid$, 1
- $MOVIE.mid = GENRE.mid$, 0.9
- $MOVIE.mid = CAST.mid$, 0.8

Georgia Koutrika, Yannis Ioannidis

Personalization of Queries

UNIBZ, December 11, 2006
Next, algorithm proceeds as follows

- While $QP$ is not empty:
  - Take first preference (P) from ordered list $QP$
  - Try to expand it with one more atom
  - If not conflicting, store it into $QP$
Preference Selection Algorithm (Example)

Join preference expansion

- We take head of $QP$
- Identify that it is a join (not selection)
- Expand into longer paths which are placed (if satisfies conditions) into $QP$

$QP$:
- PLAY.tid=THEATRE.tid, 1
- MOVIE.mid=DIRECTED.mid, 1
- MOVIE.mid=GENRE.mid, 0.9
- MOVIE.mid=CAST.mid, 0.8
Preference Selection Algorithm (Example)

Join preference expansion

- Preference `PLAY.tid=THEATRE.tid,1`
- Expanded with two atoms `THEATER.tid=PLAY.tid, THEATER.REGION="DOWNTOWN"`
- Into `{PLAY.tid=THEATRE.tid AND THEATER.tid=PLAY.tid, 1*1=1}`
  AND `{PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 1*0.8=0.8}`
- `{PLAY.tid=THEATRE.tid AND THEATER.tid=PLAY.tid, 1}` Dropped, because expands to a relation belonging to `QP` (Also Cycles are formed)

`QP`:

- `PLAY.tid=THEATRE.tid, 1`
- `MOVIE.mid=DIRECTED.mid, 1`
- `MOVIE.mid=GENRE.mid, 0.9`
- `MOVIE.mid=CAST.mid, 0.8`

PK:
Preference Selection Algorithm (Example)

Join preference expansion

- Preference `PLAY.tid=THEATRE.tid, 1`
- Expanded with two atoms `THEATER.tid=PLAY.tid, THEATER.REGION="DOWNTOWN"`
  - Into `{PLAY.tid=THEATRE.tid AND THEATER.tid=PLAY.tid, 1*1=1}`
    - AND `{PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 1*0.8=0.8}`
  - `{PLAY.tid=THEATRE.tid AND THEATER.tid=PLAY.tid, 1}` Dropped, because expands to a relation belonging to $QP$ (Also Cycles are formed)
  - `{PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 0.8}` is stored to $QP$

$QP$:
- `PLAY.tid=THEATRE.tid, 1`
- `MOVIE.mid=DIRECTED.mid, 1`
- `MOVIE.mid=GENRE.mid, 0.9`
- `MOVIE.mid=CAST.mid, 0.8`
- `PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 0.8`
Preference Selection Algorithm (Example)

QP after first iteration we have QP:

MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did, 1
MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID, 0.8
PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 0.8
MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63
MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

\( P_k \):
### Preference Selection Algorithm (Example)

<table>
<thead>
<tr>
<th>THEATER</th>
<th>NAME</th>
<th>TID</th>
<th>PHONE</th>
<th>REGION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAY</td>
<td>MID</td>
<td>PHONE</td>
<td>REGION</td>
<td></td>
</tr>
<tr>
<td>MOVIE</td>
<td>MID</td>
<td>TITLE</td>
<td>YEAR</td>
<td></td>
</tr>
<tr>
<td>DIRECTED</td>
<td>MID</td>
<td>NAME</td>
<td>DIRECTOR</td>
<td></td>
</tr>
<tr>
<td>CAST</td>
<td>MID</td>
<td>AID</td>
<td>ROLE</td>
<td></td>
</tr>
<tr>
<td>ACTOR</td>
<td>AID</td>
<td>NAME</td>
<td>DIRECTOR</td>
<td></td>
</tr>
<tr>
<td>MOVIE</td>
<td>MID</td>
<td>GENRE</td>
<td>GENRE</td>
<td></td>
</tr>
<tr>
<td>DIRECTED</td>
<td>MID</td>
<td>DIRECTED</td>
<td>DIRECTED</td>
<td></td>
</tr>
</tbody>
</table>

#### Join preference expansion

- We go on. We similarly expand MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did
- Into MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9

#### QP:

- MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did, 1
- MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
- MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID, 0.8
- PLAY.tid=THEATRE.tid AND THEATER_REGION="DOWNTOWN", 0.8
- MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63
- MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

### Query Preference

$P_K$:
Join preference expansion

- We go on. We similarly expand MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did
- Into MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9

**QP:**

- MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9
- MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
- MOVIE.mid=CAST,mid AND CAST.AID=ACTOR.AID, 0.8
- PLAY.tid=THEATRE.tid AND THEATER_REGION="DOWNTOWN", 0.8
- MOVIE.mid=DIRECTED,mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7
- MOVIE.mid=GENRE.mid AND GENRE.genre="THRILLER", 0.63
- MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45
Preference Selection Algorithm (Example)

Selection preference expansion

- And now first time we have selection preference
  
  \{MOVIE.mid=DIRECTED.mid AND
  DIRECTED.did=DIRECTOR.did AND
  DIRECTOR.name="LYNCH", 0.9\}

**QP**: MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND
DIRECTOR.name="LYNCH", 0.9
MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID, 0.8
PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 0.8
MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND
DIRECTOR.name="ALLEN", 0.7
MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63
MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

**PK**: 

Georgia Koutrika, Yiannis Ioannidis (UNIBZ, December 11, 2006)
Selection preference expansion

- And now first time we have selection preference

\{\text{MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH"}, 0.9\}

- It satisfies CI (preference with degree of interest > 0.43) criteria and we add it to $P_K$!

**$P_K$**:
MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7

MOVIE.mid=GENRE.mid AND GENRE.genre="THRILLER", 0.63

MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

**$QP$**:
MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID, 0.8
PLAY.tid=THEATRE.tid AND THEATERREGION="DOWNTOWN", 0.8
MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7
MOVIE.mid=GENRE.mid AND GENRE.genre="THRILLER", 0.63
MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

Georgia Koutrika, Yannis Ioannidis ()
Personalization of Queries
UNIBZ, December 11, 2006 30 / 38
Operation we do

**Move** MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81 $P_K$

**QP:**
- MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
- PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 0.8
- MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7
- MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63
- MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

**$P_K$:**
- MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9
Preference Selection Algorithm (Example)

**Operation we do**

**Expand** MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID, 0.8

\[ QP : \]
MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID, 0.8
PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 0.8
MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7
MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63
MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

\[ P_K : \]
MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9
MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
Preference Selection Algorithm (Example)

Operation we do

**Move** \( \text{PLAY} \cdot \text{tid} = \text{THEATRE} \cdot \text{tid} \text{ AND THEATER.REGION} = \text{"DOWNTOWN"}, 0.8 \) to \( P_K \)

**QP:**
- \( \text{PLAY} \cdot \text{tid} = \text{THEATRE} \cdot \text{tid} \text{ AND THEATER.REGION} = \text{"DOWNTOWN"}, 0.8 \)
- \( \text{MOVIE} \cdot \text{mid} = \text{CAST} \cdot \text{mid} \text{ AND CAST.AID} = \text{ACTOR.AID} \text{ AND CAST.AID} = \text{ACTOR.AID}, 0.8 \)
- \( \text{MOVIE} \cdot \text{mid} = \text{DIRECTED} \cdot \text{mid} \text{ AND DIRECTED.did} = \text{DIRECTOR.did} \text{ AND DIRECTOR.name} = \text{"ALLEN"}, 0.7 \)
- \( \text{MOVIE} \cdot \text{mid} = \text{GENRE} \cdot \text{mid} \text{ AND GENRE.genre} = \text{"THRILER"}, 0.63 \)
- \( \text{MOVIE} \cdot \text{mid} = \text{GENRE} \cdot \text{mid} \text{ AND GENRE.genre} = \text{"ADVENTURE"}, 0.45 \)

**\( P_K \):**
- \( \text{MOVIE} \cdot \text{mid} = \text{DIRECTED} \cdot \text{mid} \text{ AND DIRECTED.did} = \text{DIRECTOR.did} \text{ AND DIRECTOR.name} = \text{"LYNCH"}, 0.9 \)
- \( \text{MOVIE} \cdot \text{mid} = \text{GENRE} \cdot \text{mid} \text{ AND GENRE.genre} = \text{"COMEDY"}, 0.81 \)
Preference Selection Algorithm (Example)

Operation we do

Expand MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID, 0.8

\[ Q_P : \]
MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID, 0.8
MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7
MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63
MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

\[ P_K : \]
MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9
MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
PLAY.tid=THEATRE.tid AND THEATERREGION="DOWNTOWN", 0.8
Preference Selection Algorithm (Example)

Operation we do

Expand \( \text{MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID, 0.8} \)
Note, that expansion of above lead to three terms. But, \( C_l \) is set to accept \( P \) with degrees >0.43

Discard \( \text{MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID AND ACTOR.name="ROSELLINI", 0.42} \)

\( Q_P : \)
\[
\text{MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID AND ACTOR.name="KIDMAN", 0.72}
\]
\[
\text{MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7}
\]
\[
\text{MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID AND ACTOR.name="HOPKINS", 0.64}
\]
\[
\text{MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63}
\]
\[
\text{MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45}
\]

\( P_K : \)
\[
\text{MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9}
\]
\[
\text{MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81}
\]
\[
\text{PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 0.8}
\]
Operation we do

What is left are selection preferences. Move all to $P_K$

$QP$:
- MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID AND ACTOR.name="KIDMAN", 0.72
- MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7
- MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID AND ACTOR.name="HOPKINS", 0.64
- MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63
- MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45

$P_K$:
- MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9
- MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
- PLAY.tid=THEATRE.tid AND THEATER_REGION="DOWNTOWN", 0.8
Preference Selection Algorithm (Example)

Operation we do

Return $P_K$

$QP :$

$P_K :$

MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="LYNCH", 0.9
MOVIE.mid=GENRE.mid AND GENRE.genre="COMEDY", 0.81
PLAY.tid=THEATRE.tid AND THEATER.REGION="DOWNTOWN", 0.8
MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID AND ACTOR.name="KIDMAN", 0.72
MOVIE.mid=DIRECTED.mid AND DIRECTED.did=DIRECTOR.did AND DIRECTOR.name="ALLEN", 0.7
MOVIE.mid=CAST.mid AND CAST.AID=ACTOR.AID AND CAST.AID=ACTOR.AID AND ACTOR.name="HOPKINS", 0.64
MOVIE.mid=GENRE.mid AND GENRE.genre="THRILER", 0.63
MOVIE.mid=GENRE.mid AND GENRE.genre="ADVENTURE", 0.45
Preference Integration

Single Query

```
SELECT DISTINCT MV.title
FROM MOVIE MV, PLAY PL, CAST CA, ACTOR AC,
GENRE GN, DIRECTED DD, DIRECTOR DI
WHERE MV.mid=PL.mid and PL.date="2/7/2003" and
  ((MV.mid=GN.mid and GN.genre="comedy"
    and MV.mid=CA.mid and CA.aid=AC.aid
    and AC.name="N. Kidman")(MV)
  or
  ((MV.mid=CA.mid and CA.aid=AC.aid
    and AC.name="N. Kidman"
    and MV.mid=DD.mid and DD.did=DI.did
    and DI.name="D. Lynch")
  or
  ((MV.mid=GN.mid and GN.genre="comedy"
    and MV.mid=DD.mid and DD.did=DI.did
    and DI.name="D. Lynch")))
```

- Conjunction of the mandatory conditions
- Disjunction of all possible conjunctions of L conditions from the remaining K-M ones
A set of K-M queries, each one containing a simpler qualification, which is a conjunction of the mandatory conditions, and one condition from the remaining K-M ones.
Implementation issues

- Implementation is done on top of Oracle 9i
- Data comes from Interned Movies Database (imdb.com). Information about 340000 movies
- Profiles are partially generated, partially extracted from data

Experimentation type

- Experiments show computational costs (Off-line evaluation)
- Do not provide On-line evaluation
Effect of Profile Size on Preference Selection Time

Execution time dependency on profile size

- Bigger profile, faster computations!
  - Top k atomic preferences are more likely to be integrated
  - Do not need to compute other preferences
- Smaller K, faster computations.
Size of the Results of Personalized Queries

Size of the results of personalized queries with $K$

- Unintuitive!
- Increasing number of preferences $K$, gives more data

![Graph showing size of results with $K$]

Size of the Results of Personalized Queries with $L$ ($K=60$)

- Bigger $L$, less data.
  - More preferences should be fulfilled. More constraints.

![Graph showing size of results with $L$]
Performance of Personalization

Execution time dependency on profile size

- Faster with personalization???
- Time of transferring data takes more than algorithm execution
- Raw query
  - Select * FROM MOVIE;
  - A LOT OF DATA to access/transfer
- Personalization
  - Select * FROM MOVIE WHERE . . . AND . . . AND . . . AND . . . ;
  - LITTLE OF DATA to access/transfer

Algorithm + Execution of complex query takes less time than transfer a lot of data
Our Opinion

- It is a nice theoretical approach which gives effortless integration of profile to the query
- Only off-line evaluation
- No serendipity. Only profile specific results will be given.