OLAP Query personalisation and recommendation: an introduction

Patrick Marcel, Université François Rabelais Tours
Laboratoire d'Informatique

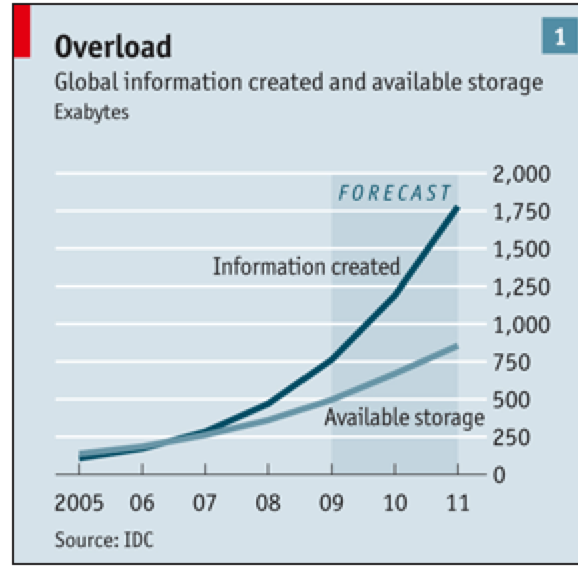
Outline

- Introduction
- Query personalisation
  - Basics on preferences
  - Overview of existing approaches in relational databases
  - Existing approaches in multidimensional databases
- Query recommendation
  - Basics on recommender systems
  - Overview of existing approaches in relational databases
  - Existing approaches in multidimensional databases
- Conclusion
- Bibliography

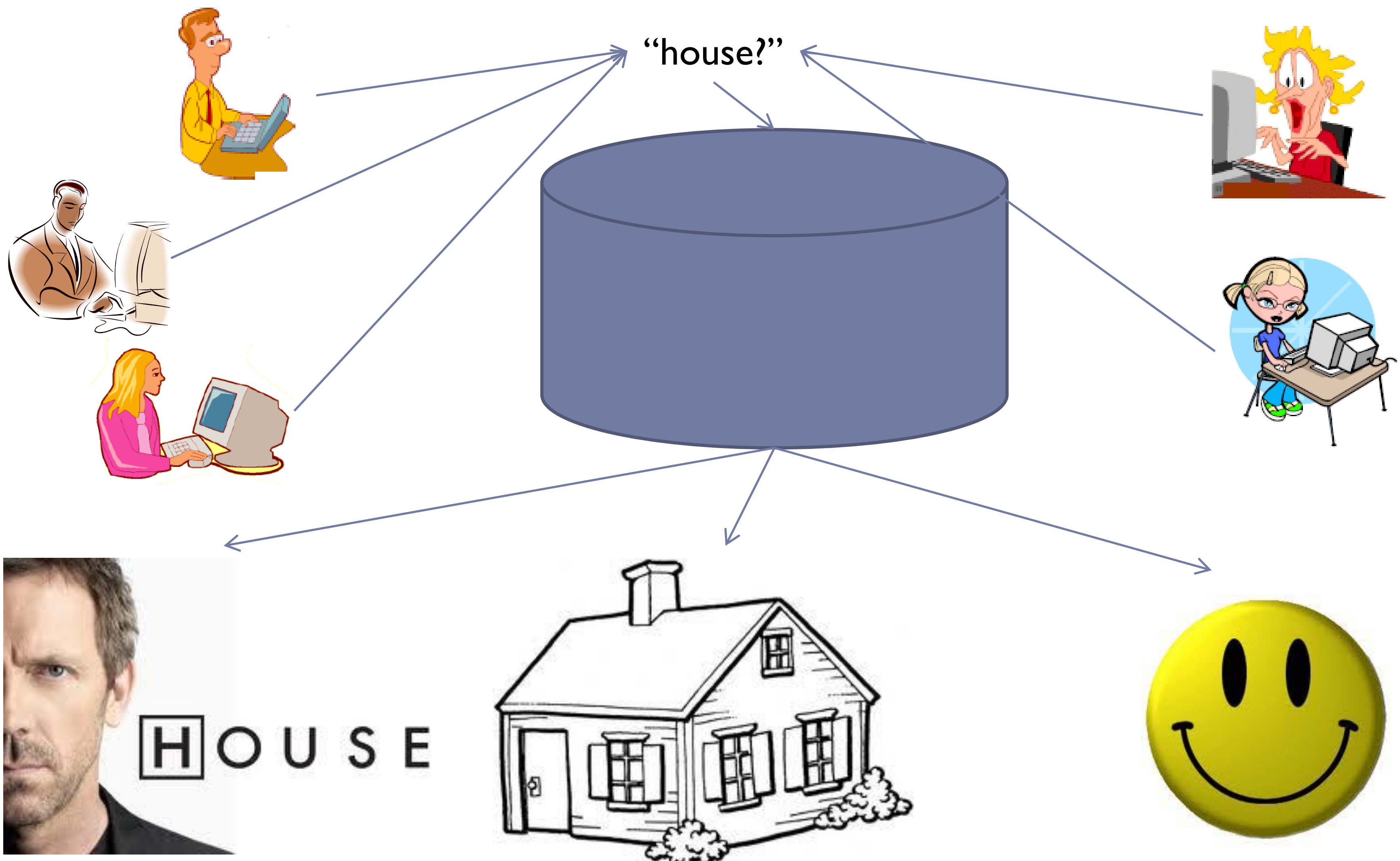
Introduction

Why personalisation or recommendation?

- Mankind created 150 exabytes (billion gigabytes) of data in 2005. In 2010, it will create 1,200 exabytes.
  - The Economist, The Data Deluge, Feb 25th 2010
- Databases should be more user-friendly [Jagadish & al., 2007]
  - Instances are huge, schemas are complex
  - The user may not know SQL, the schema, the values

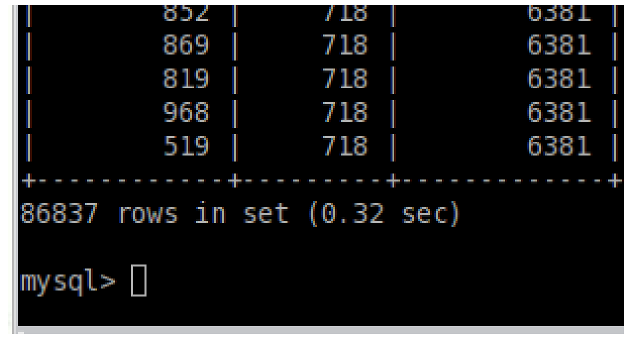


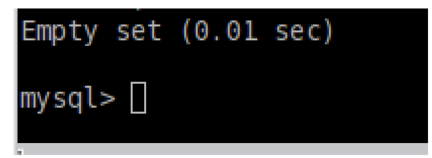
Why personalisation?



Why personalisation in database?

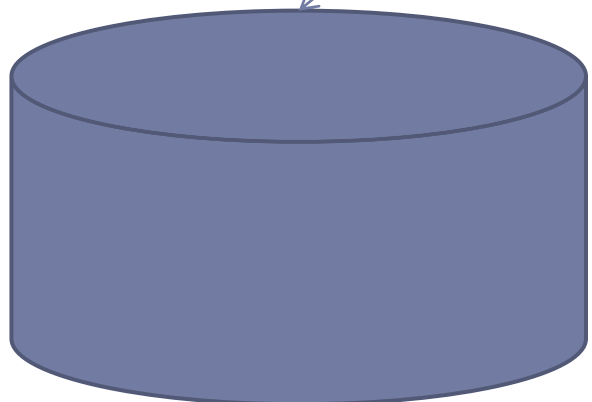
- Given a database query q
  - Am I always happy with the result?
    - Too many answers
      - How to focus on the most relevant?
    - Too few answers
      - How to soften hard constraints?
- Adding preferences to queries
  - If too many answers
    - Rank them to focus on the preferred ones
  - If too few answers
    - Consider selections as preferences, not constraints

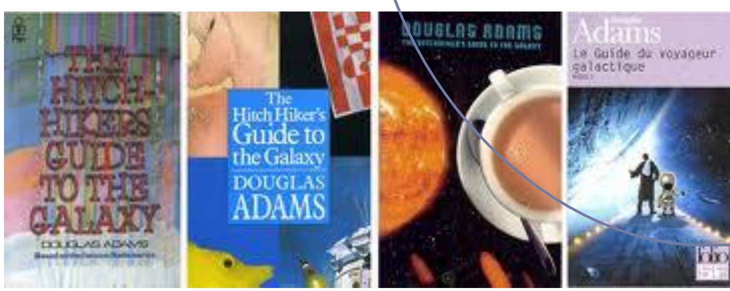




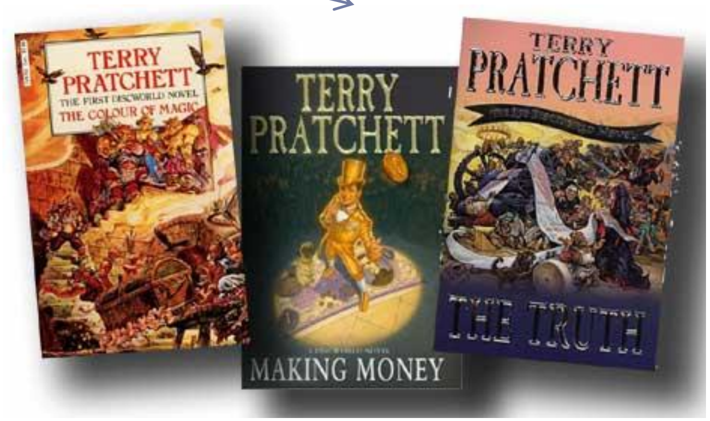
Why recommendation?

"Books by T. Pratchett?"

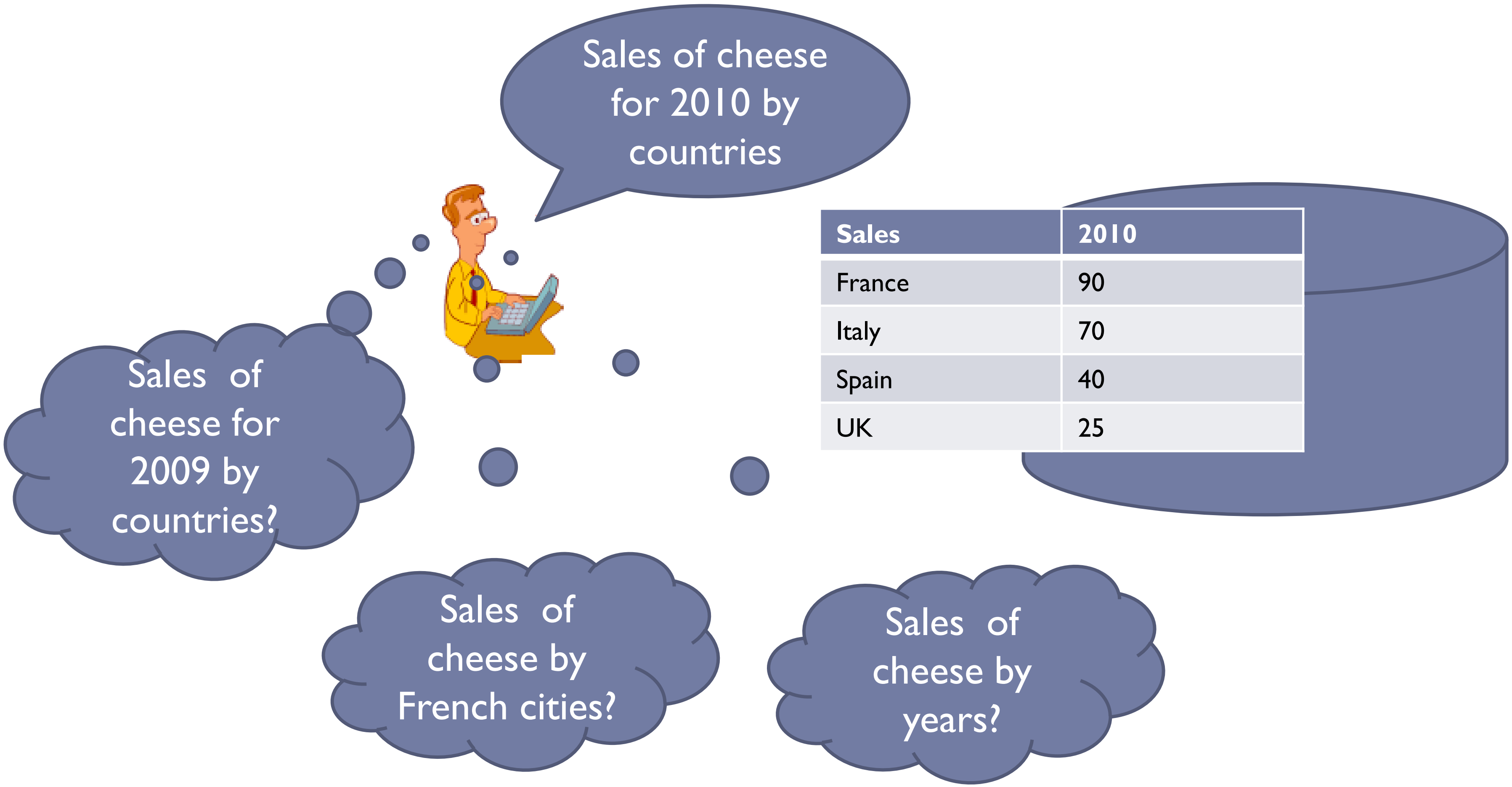




"Consider also books by D. Adams"
•Same style
•Same price
•Popular
•New edition
•…



Why recommendation in databases?



Sales	2010
France	90
Italy	70
Spain	40
UK	25

Scope

- Personalisation
  - A process that, given a **database query q** and some **profile**, computes **another query q' $\subset$ q** that has an added value for the user

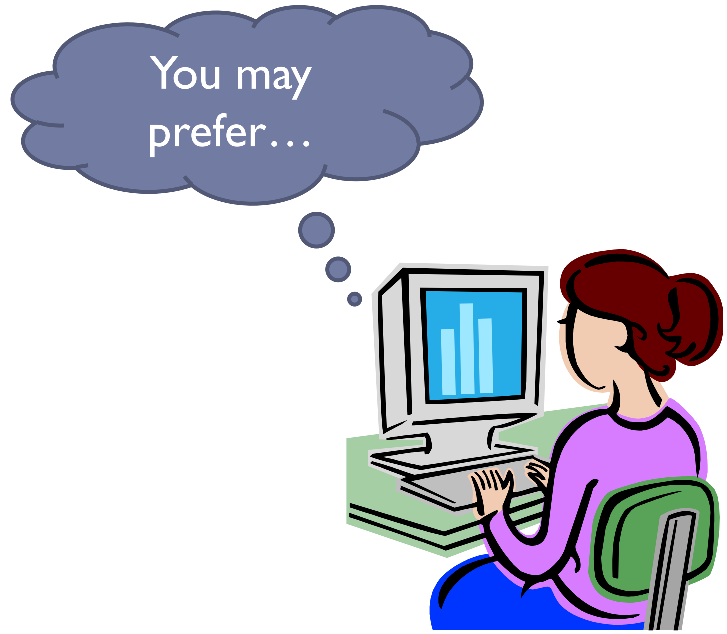
- Recommendation
  - A process that, given **a database query q** and some **profile**, computes **another query q' $\not\subset$ q, q $\not\subset$ q'** that has an added value for the user

- What is outside the scope
  - Other forms of query transformation (relaxation, completion, etc.)
  - Non relational data types (XML, etc.)
  - Implementation and evaluation issues

Categorisation: [Golfarelli & Rizzi, 2010]

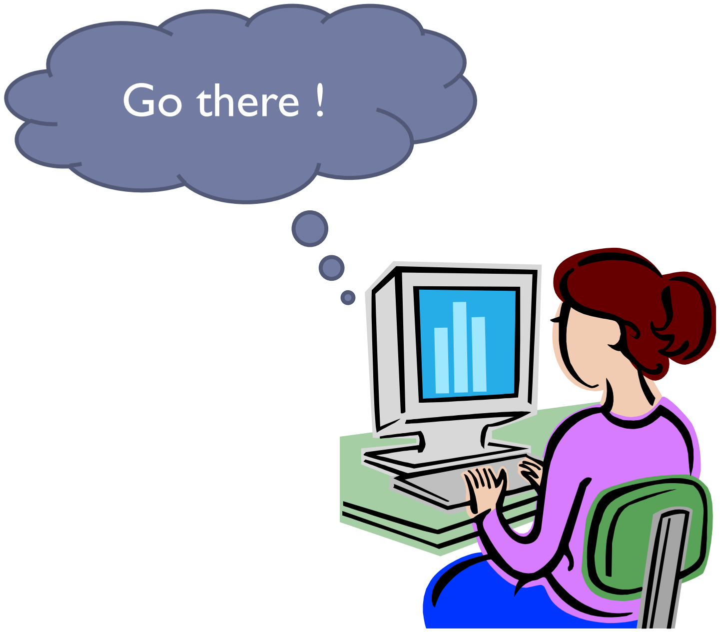
- Formulation effort:
  - How profile is specified
- Prescriptiveness:
  - How profile is incorporated to the query
- Proactiveness:
  - How profile affects query evaluation
- Expressiveness:
  - How complex profile is

Formulation effort



- Formulation effort:
  - Profile elements manually specified for each query, or
  - Profile inferred from the context and/or past actions.

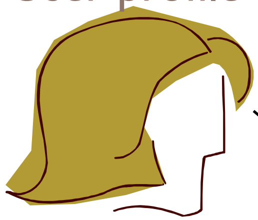
Prescriptiveness



- Prescriptiveness:
  - Profile elements added as hard constraints to a query, or
  - Tuples that satisfy as much profile as possible are returned even if no tuples satisfies all the profile.

Proactiveness (1)

User profile

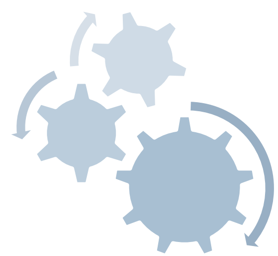


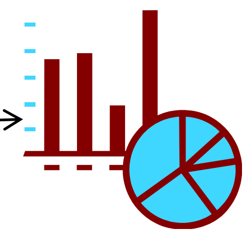
Personalize and execute or execute and personalize

Present the result

User query

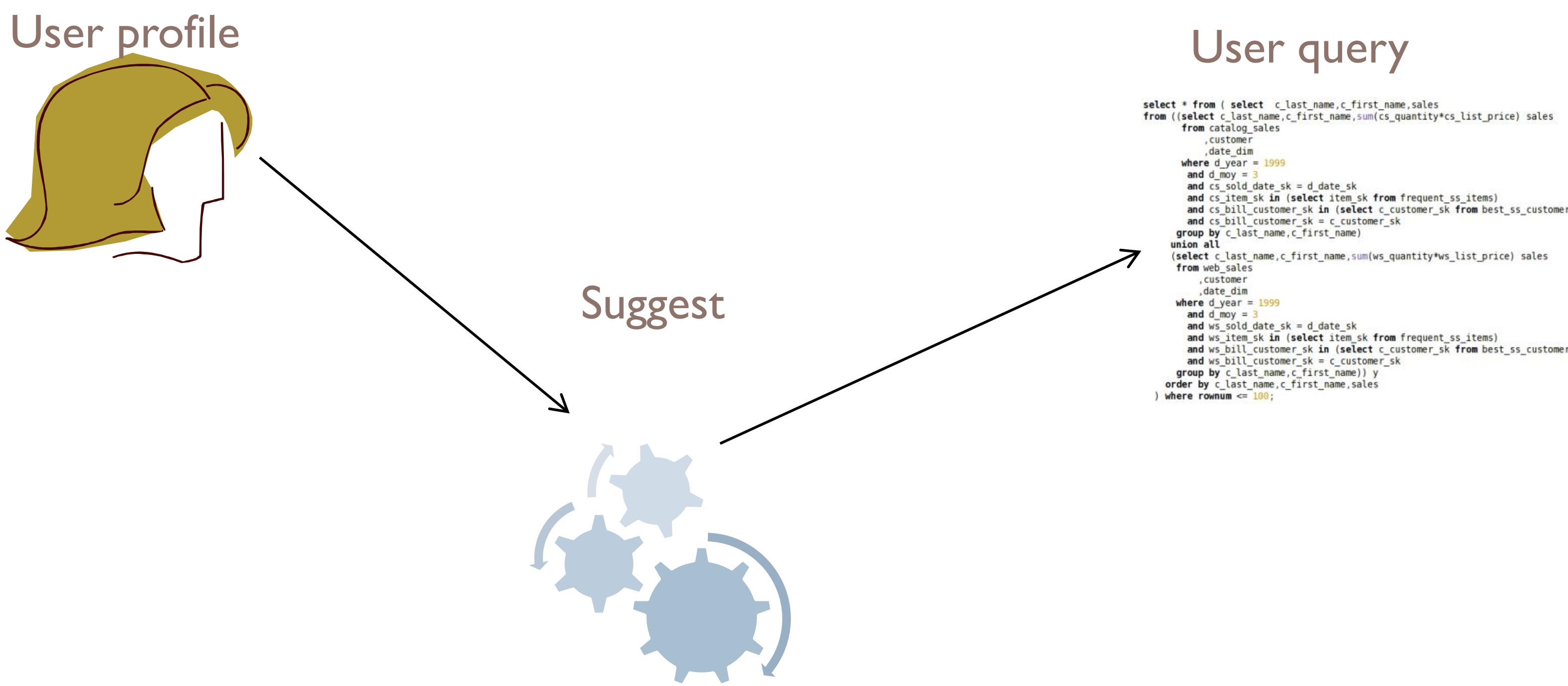
```sql
select * from ( select  c_last_name,c_first_name,sales
from ((select c_last_name,c_first_name,sum(cs_quantity*cs_list_price) sales
       from catalog_sales
           ,customer
           ,date_dim
       where d_year = 1999
        and d_moy = 3
        and cs_sold_date_sk = d_date_sk
        and cs_item_sk in (select item_sk from frequent_ss_items)
        and cs_bill_customer_sk in (select c_customer_sk from best_ss_customer)
        and cs_bill_customer_sk = c_customer_sk
      group by c_last_name,c_first_name)
     union all
     (select c_last_name,c_first_name,sum(ws_quantity*ws_list_price) sales
      from web_sales
          ,customer
          ,date_dim
      where d_year = 1999
        and d_moy = 3
        and ws_sold_date_sk = d_date_sk
        and ws_item_sk in (select item_sk from frequent_ss_items)
        and ws_bill_customer_sk in (select c_customer_sk from best_ss_customer)
        and ws_bill_customer_sk = c_customer_sk
      group by c_last_name,c_first_name)) y
    order by c_last_name,c_first_name,sales
  ) where rownum <= 100;
```





- Proactiveness:
  1. Change the current query before execution or post process its results, or
  2. Suggest new queries without executing them.

Proactiveness (2)



- **Proactiveness:**
  1. Change the current query before execution or post process its results, or
  2. Suggest new queries without executing them.

Expressiveness

- I prefer movies directed by David Lynch

- I prefer movies directed by David Lynch
- But I also prefer short movies
- I like Julia Roberts more than Nicole Kidman
- Well it depends if it is a drama or a comedy
- Length is more important than the director
- Except if it is a comedy
- …

Query personalisation

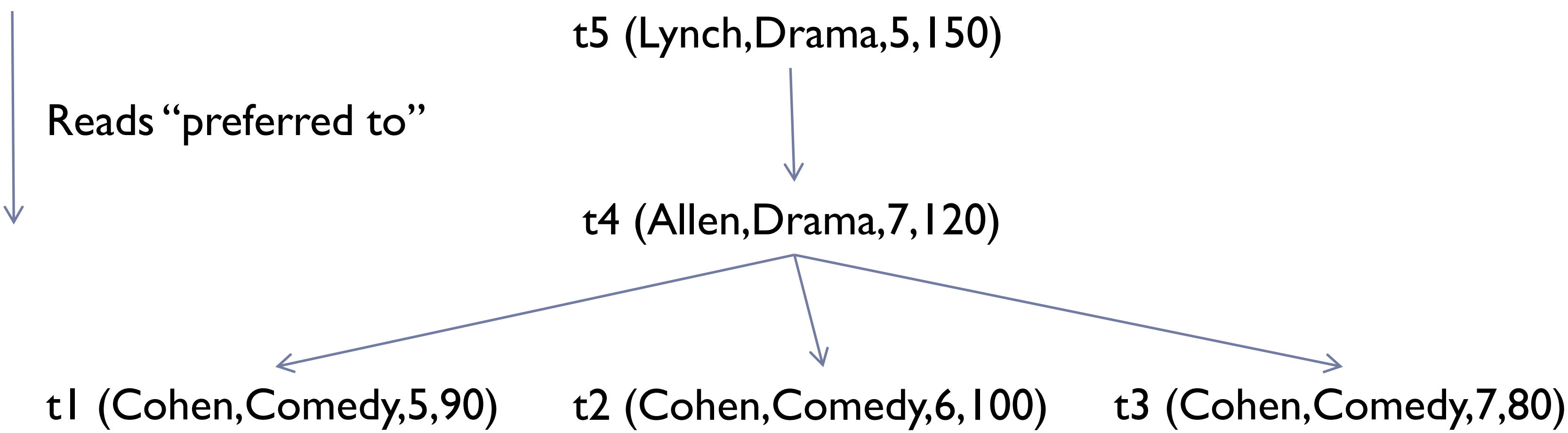
Basics on preferences

Example

Movies	Author	Genre	Price	Duration
t1	Cohen	Comedy	5	90
t2	Cohen	Comedy	6	100
t3	Cohen	Comedy	7	80
t4	Allen	Drama	7	120
t5	Lynch	Drama	5	150

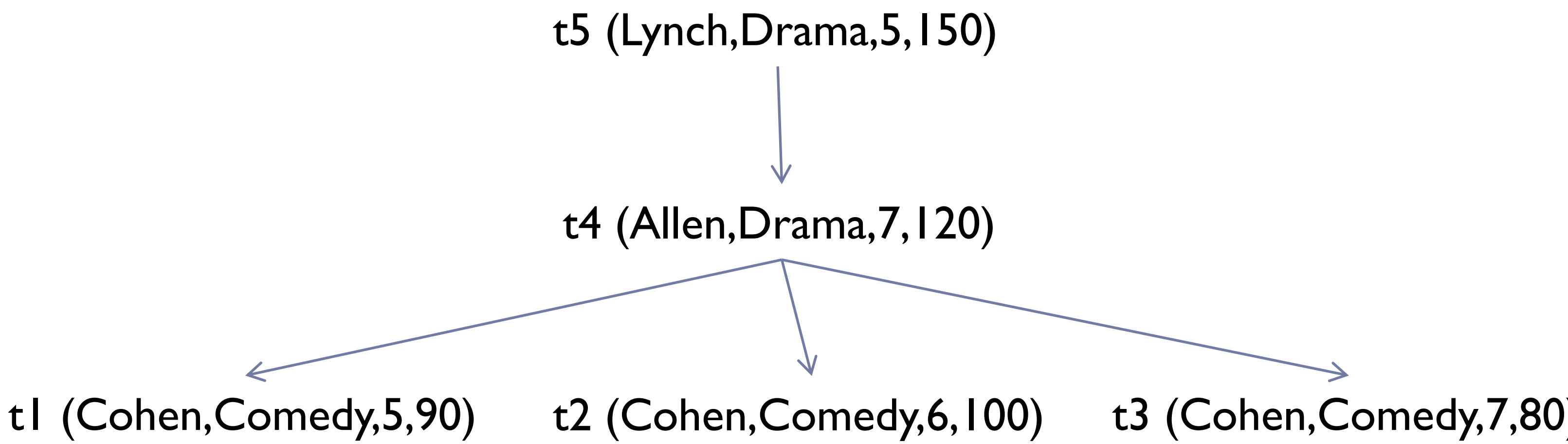
- "I prefer Lynch movies over Allen's and Allen movies over Cohen's"
  - Then t5 preferred to t4 and t4 preferred to t1, t2, t3
  - Nothing is said e.g., for t1 and t2, neither for t1 and t3

Example of representation



- “I prefer Lynch movies over Allen’s and Allen movies over Cohen’s”
  - t5 > t4
  - t4 > t1, t4 > t2, t4 > t3
  - Prefers(t5,t4)
  - Prefers(t4,t1), Prefers(t4,t2), Prefers(t4,t3)

Another formulation



- "I like Lynch: score=0.9"
- "I like Allen: score=0.8"
- "I like Cohen: score=0.5"

Qualitative versus quantitative

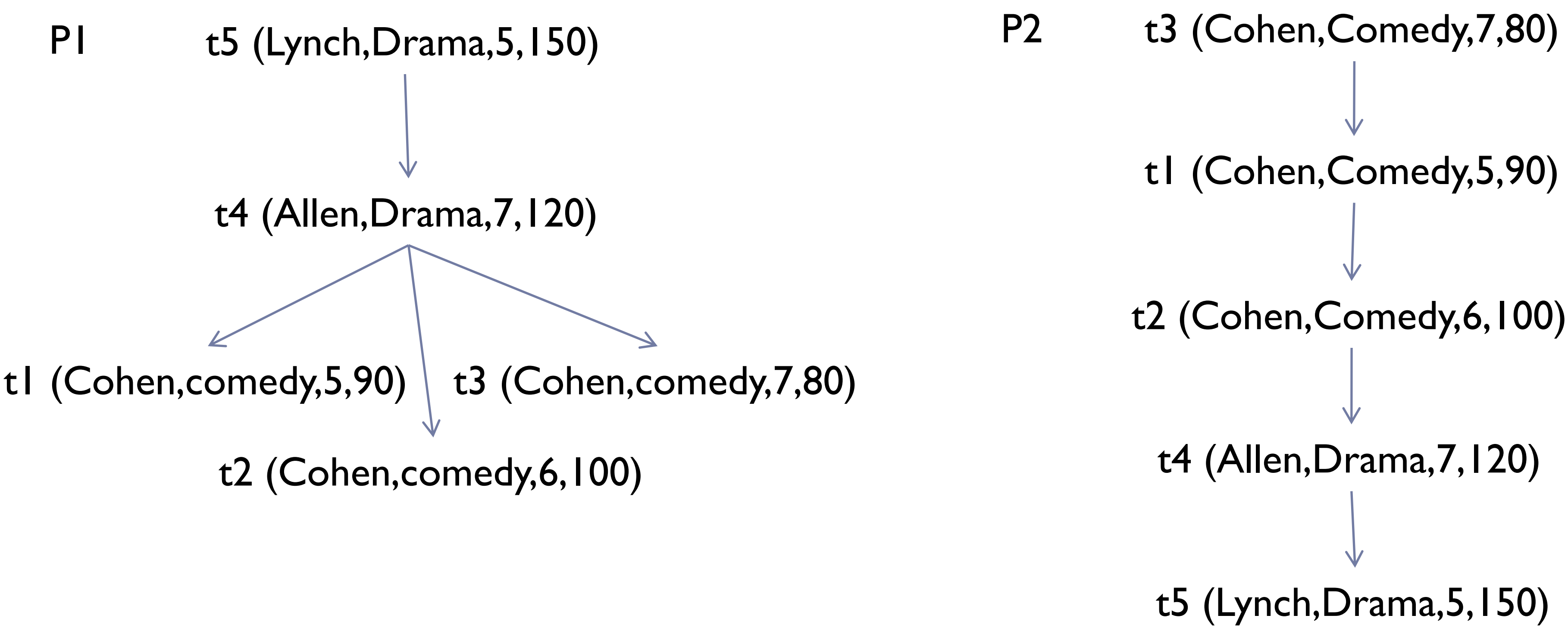
- Qualitative Approaches
  - Relative preferences of the form I like A better than B
  - Based on Partial ordering
    - I like A better than B iff (A > B) where ">" is a partial ordering
- Quantitative Approaches
  - Absolute preferences of the form I like A to a specific degree
  - Based on Scoring / Utility Functions
    - I like A better than B iff u(A) > u(B) where "u" is a scoring function
- However, not every intuitively plausible preference relation can be captured by scoring functions
  - But scoring functions can express the "intensity" of the preference

Preferences are usually SPO

- Strict Partial Order (SPO)
  - A binary relation “>” over a set O which is
    - Irreflexive: $\neg(a > a)$
    - Asymmetric: If $(a \neq b)$ and $(a > b)$ then $\neg(b > a)$
    - Transitive: If $(a > b)$ and $(b > c)$ then $(a > c)$
- Preferences are usually assumed to be SPO
  - I like “a” better than “b” if $(a > b)$
  - I consider a and b indifferent $(a \sim b)$ if $\neg(a > b)$ and $\neg(b > a)$

Preference composition

- P1: "I prefer Lynch's over Allen's and Allen's over Cohen's"
- P2: "I also prefer shorter movies"



Example of quantitative composition

- "I prefer Lynch's over Allen's and Allen's over Cohen's"
  - "I like Lynch with $score_{P1}$=0.9"
  - "I like Allen with $score_{P1}$=0.8"
  - "I like Cohen with $score_{P1}$=0.5"
- "I also prefer shorter movies"
  - "I like (duration=80) with $score_{P2}$=1", "I like (duration=90) with $score_{P2}$=0.9", …, "I like (duration=150) with $score_{P2}$=0.6"
- Combination can be with weighted summation
  - $Score_{f(P1,P2)}(t_i)=x\ score_{P1}(t_i) + (1-x)\ score_{P2}(t_i)$

Intersection P1 ∩ P2
($t >_{\cap} t'$) if ($t >_{P1} t'$) and ($t >_{P2} t'$)

- "I prefer Lynch's over Allen's and Allen's over Cohen's"
- "I also prefer shorter movies"

Would union achieve the same?

t3 (Cohen,Comedy,7,80)

t1 (Cohen,Comedy,5,90)

t2 (Cohen,Comedy,6,100)

t4 (Allen,Drama,7,120)

t5 (Lynch,Drama,5,150)

Prioritization P1 $\triangleright$ P2

$(t >_{\triangleright} t')$ if $(t >_{P1} t')$ or $(\neg(t' >_{P1} t)$ and $(t >_{P2} t'))$

- "I prefer Lynch's over Allen's and Allen's over Cohen's"
- "I also prefer shorter movies"

t5 (Lynch,Drama,5,150)

↓

t4 (Allen,Drama,7,120)

↓

t3 (Cohen,Comedy,7,80)

↓

t1 (Cohen,Comedy,5,90)

↓

t2 (Cohen,Comedy,6,100)

What about P2 $\triangleright$ P1?

Pareto P1 $\otimes$ P2

$(t >_{\otimes} t')$ if $((t >_{P1} t')$ and $(t >_{P2} t'$ or $t \sim_{P2} t'))$
or $((t >_{P2} t')$ and $(t >_{P1} t'$ or $t \sim_{P1} t'))$

- "I prefer Lynch's over Allen's and Allen's over Cohen's"
- "I also prefer shorter movies"

t5 (Lynch,Drama,5,150)

t4 (Allen,Drama,7,120)

t3 (Cohen,Comedy,7,80)

t1 (Cohen,Comedy,5,90)

t2 (Cohen,Comedy,6,100)

Existing approaches

In relational databases

Two approaches

- Preference operators
  - Use explicit preference operators in queries
    - Winnow [Chomicki, 2003]
    - *Preference SQL [Kießling, 2002]*
      - *High formulation effort , not prescriptive, not proactive, high expressiveness*
    - Skyline [Börzsönyi & al., 2001]
- Query expansion
  - Rewrite regular queries with elements of a profile
    - *[Koutrika & Ioannidis, 2004]*
      - *Low formulation effort, prescriptive, not proactive, low expressiveness*

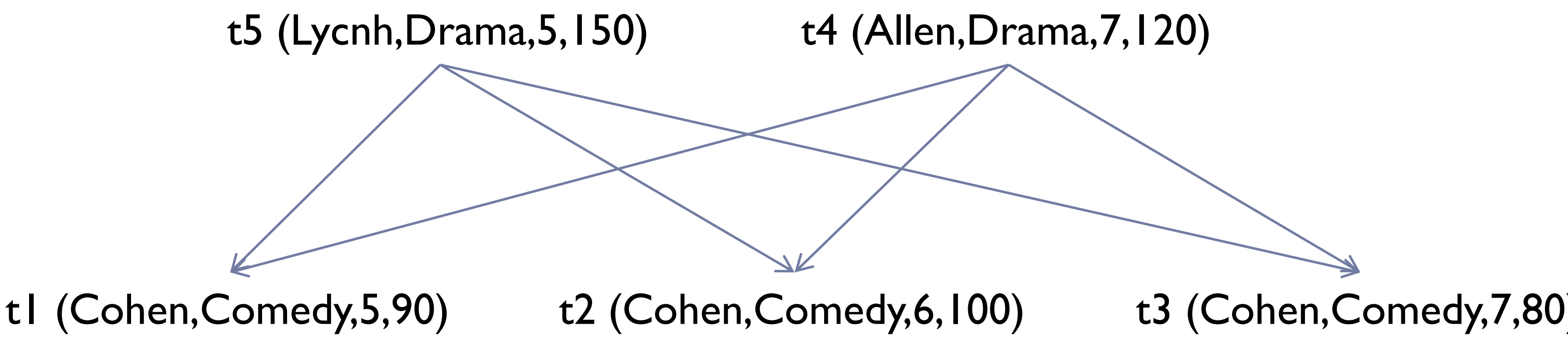
Winnow / BMO (Best-Matches-Only)

- Given
  - A relation r of schema sch(r)
  - A preference C over sch(r) defining a preference relation $>_C$

- The winnow operator, denoted w_C, is defined by:
  - $w_C(r) = \{ t \in r \mid (\nexists\, t' \in r)(t' >_C t) \}$

- Can be used to order query results
  - The answer to q can be partitioned according to C
    - $q = w_C(q) \cup w_C(q - w_C(q)) \cup \ldots$

Example



- Model C is
  - "I prefer drama"
- What are my most preferred affordable movies?
  - $w_C(\sigma_{Price<7}(Movies))$
- Answer is
  - First: t5
  - Then: t1,t2

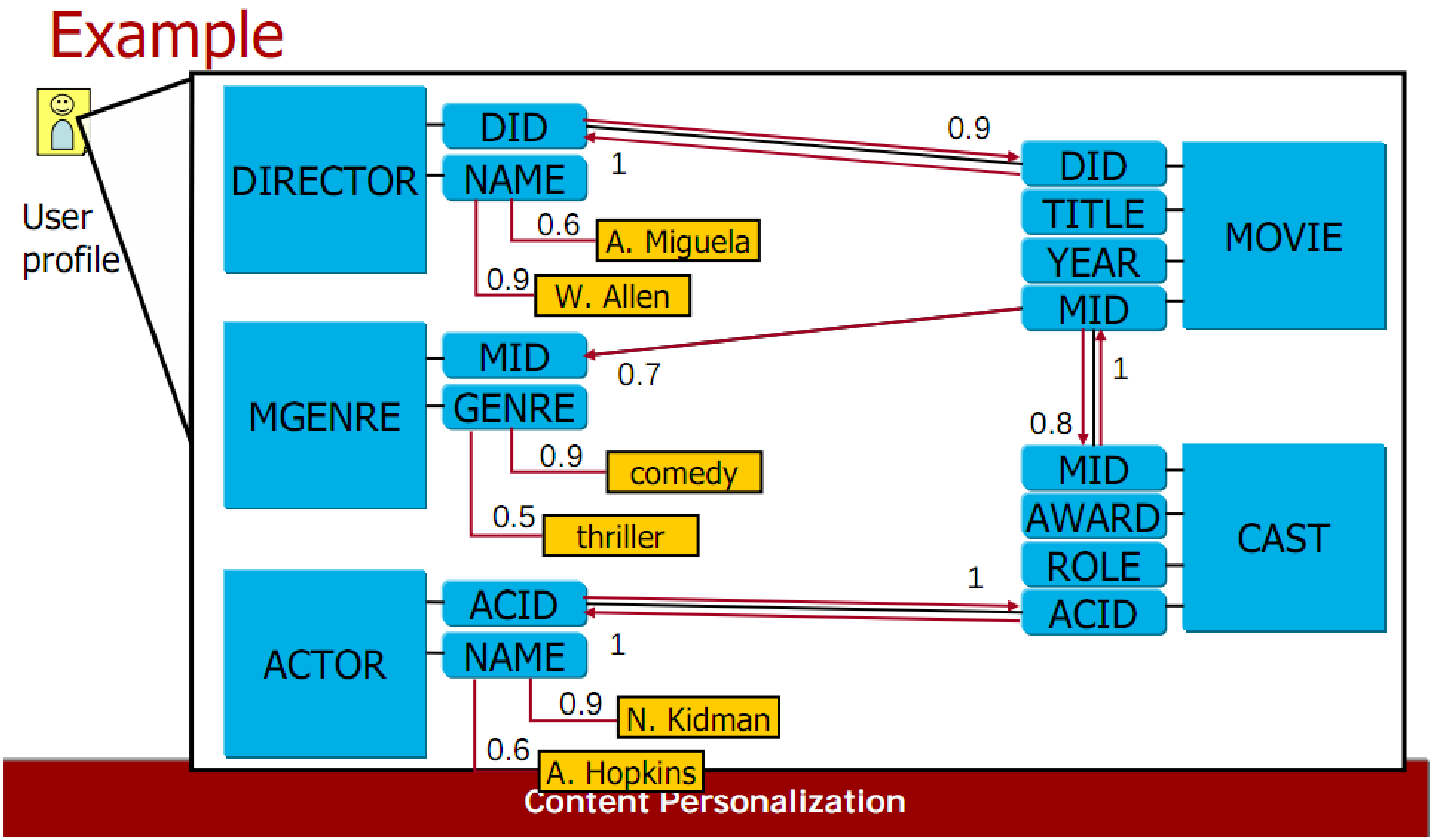
Preference SQL [Kießling, 2002]

- Built-in Preference Constructors
  - SELECT * FROM Movies
    PREFERING HIGHEST(Duration)
    - $(x >_{HIGHEST}\ y)$ if $x > y$
  - SELECT * FROM Movies
    PREFERING genre IN ('Drama','Thriller')
    - $(x >_{IN\ ('Drama','Thriller')}\ y)$ if $x \in \{$'Drama','Thriller'$\}$ and $y \notin \{$'Drama','Thriller'$\}$
  - SELECT * FROM Movies
    PREFERING Duration AROUND 90
    - $(x >_{AROUND(90)}\ y)$ if $|x - 90| < |y - 90|$

Preference SQL

- How to assemble Complex Preferences
  - With Pareto Composition
    - SELECT * FROM Movies
      PREFERING HIGHEST(Duration)
      AND Genre IN (‘Drama’,’Thriller’)
  - With Prioritized Composition
    - SELECT * FROM Movies
      PREFERING HIGHEST(Duration)
      CASCADE Genre IN (‘Drama’,’Thriller’)

Query expansion [Koutrika & Ioannidis, 2005]



User query

Example

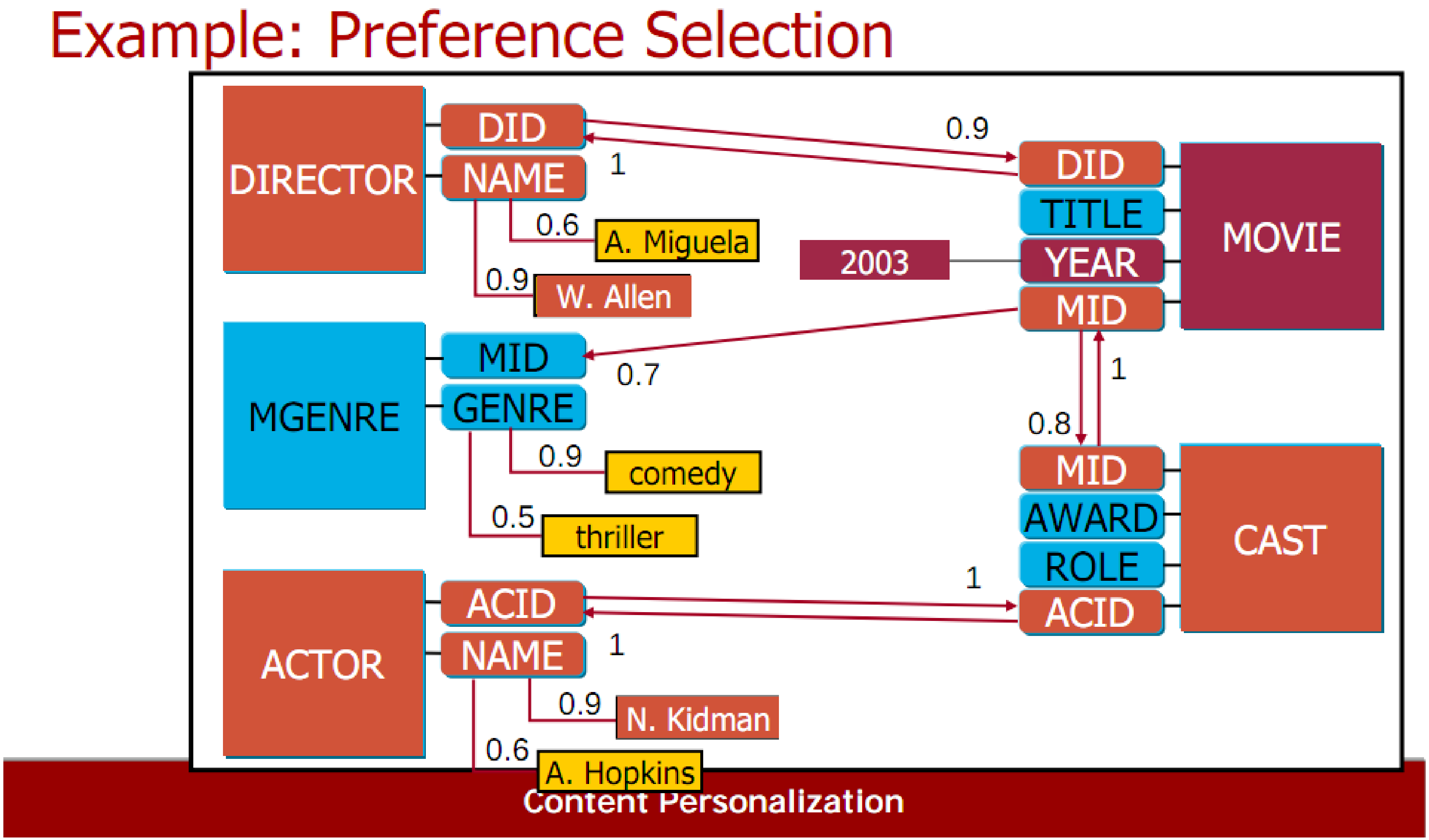
```
SELECT   MV.title
FROM     MOVIE MV
WHERE    MV.YEAR='2003'
```

Results should satisfy at least L of the K preferences

Parameters for personalization: K=2, L=1

Content Personalization

Using the profile



Expanding the query

Example: Personalized Query

- Query rewriting [70]

```
SELECT MV.title
FROM   MOVIE M, CAST C, ACTOR A, DIRECTOR D
WHERE  MV.YEAR='2003'
  and (M.DID=D.DID and D.NAME='W.Allen') or
      (M.MID=C.MID and C.ACID=A.ACID and
       A. NAME='N.Kidman')
```

Content Personalization

Existing approaches

In multidimensional databases

Peculiarities of data warehouses

- Data warehouses are particular databases
  - Read mostly instance, with an inflationist evolution
  - Schema inducing a particular topology (lattice of cuboids)
  - Shared in a multi-user environment
- OLAP queries over data warehouses
  - Expressed in a dedicated query language (MDX)
  - May produce large results, visualised as crosstabs
  - Are grouped into sessions having an analytical goal
  - Are written based on:
    - Past results of the session
    - User expectations

Two existing approaches

- [Bellatreche & al. 2005]
  - Inspired by Koutrika & Ioannidis
  - Query expansion for computing preferred visualisations
    - Low formulation effort, prescriptive, not proactive, low expressiveness

- [Golfarelli & Rizzi, 2009]
  - Inspired by Kießling
  - Preference operators adapted to the multidimensional context
    - High formulation effort, not prescriptive, not proactive, high expressiveness

[Bellatreche & al. 2005]

SELECT CROSSJOIN({City.Tours, City.Orleans},
{Category.Members}) ON ROWS
{2003, 2004, 2005, 2006} ON COLUMNS
FROM SalesCube
WHERE (Measures.quantity)

Visualization depends on the user's profile

		2003	2004	2005	2006
Tours	Drink	77	54	55	33
	Food	89	61	30	41
Orleans	Drink	25	50	49	32
	Food	33	44	59	27

		2003	2004	2005	2006
Tours	Drink	77	54	55	33
	Food	89	61	30	41
	Cloth	55	50	51	52
	Shoes	21	22	29	27

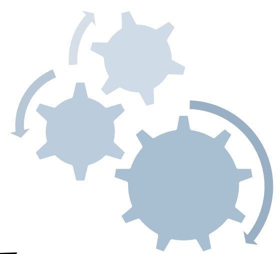
Problem formulation

User profile P

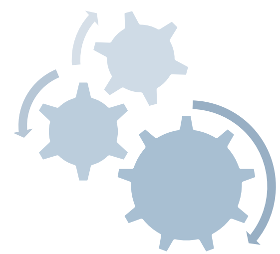
User query q

```
SELECT {AvgIncome} ON COLUMNS,
   CROSSJOIN(DESCENDANTS([RESIDENCE].[All],
      [RESIDENCE].[City], SELF_AND_BEFORE),
   CROSSJOIN(DESCENDANTS([RACE].[All],
      [RACE].[RaceGroup], SELF_AND_BEFORE),
      [OCCUPATION].[Occ].Members)) ON ROWS
FROM [CENSUS] WHERE [TIME].[Year].[2009]
```

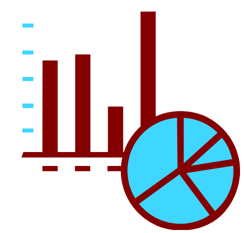
Personalize q



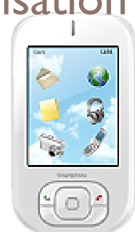
Execute the personalized query



Present the visualisation



Visualisation constraint v



- compute $q' = \max_{<P} \{q'' \subseteq q \mid v(q'') = true\}$

Example of personalization (1)

The query:

```
SELECT CROSSJOIN({City.Tours, City.Orleans},
                 {Category.Members}) ON ROWS
                 {2003, 2004, 2005, 2006} ON COLUMNS
FROM SalesCube
WHERE (Measures.quantity)
```

Preferences:

Time < Location and Product < Location
2002 < 2003 < 2004 < 2005 < 2006
Electronics < shoes < cloth < food < drink
Quantity < price

Constraint: 2 axes, no more than 4 positions on each axis

Example of personalization (2)

		2006
Drink	Orleans	
	Tours	

Step 1
The most preferred references

Example of personalization (3)

Step 2
The second most preferred references

		2006
Drink	Orleans	
	Tours	

		2006	2005
Drink	Orleans		
	Tours		
Food	Orleans		
	Tours		

Example of personalization (4)

		2006
Drink	Orleans	
	Tours	

		2006	2005
Drink	Orleans		
	Tours		
Food	Orleans		
	Tours		

		Drink	Food	Cloth
Tours	2005			
	2006			
Orleans	2005			
	2006			

Step 3: the next most preferred references

Example of personalization (5)

… finally, the constructed query is

SELECT CROSSJOIN({City.Tours, City.Orleans},
{Category.Food, Category.drink}) ON ROWS
{2003, 2004, 2005, 2006} ON COLUMNS
FROM SalesCube
WHERE (Measures.quantity)

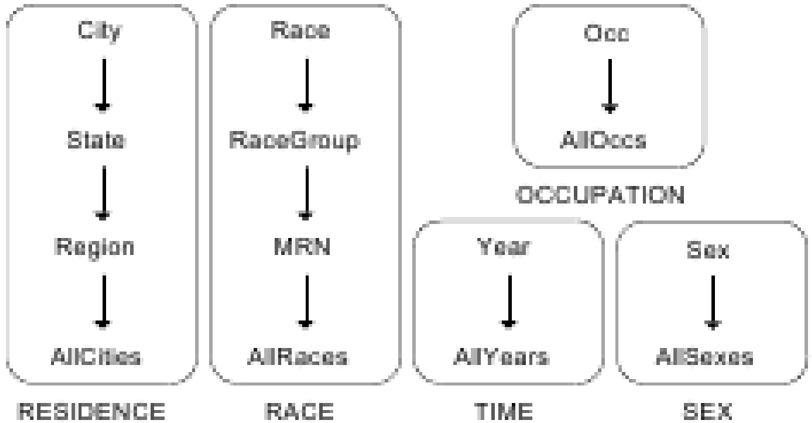
		2003	2004	2005	2006
Tours	Drink	77	54	55	33
	Food	89	61	30	41
Orleans	Drink	25	50	49	32
	Food	33	44	59	27

[Golfarelli & Rizzi 2009,2011]

- Adaptation of preference constructors to a multidimensional context
  - Taking into account hierarchies
  - Preferences can be expressed over levels and thus over cuboids
  - Preferences can be expressed over measures
- Composition: Prioritization and Pareto

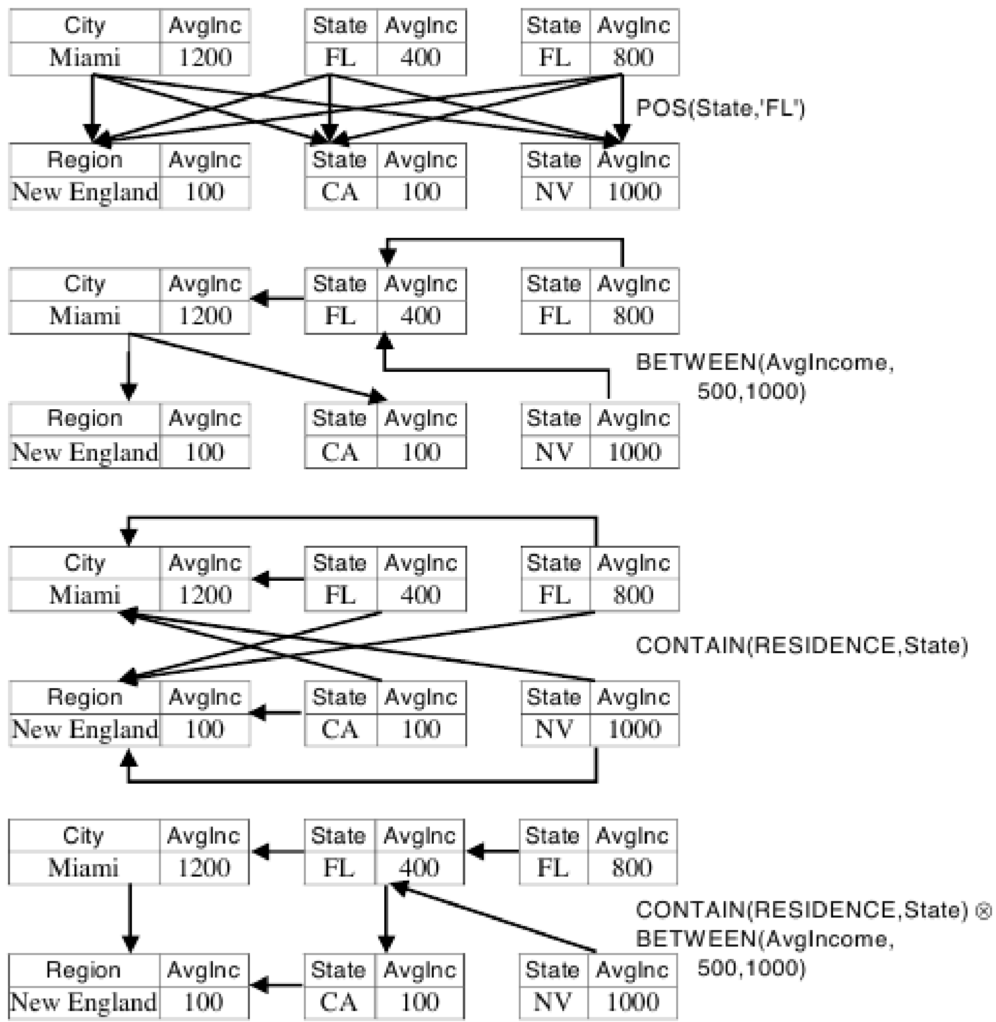
```
SELECT {AvgIncome} ON COLUMNS,
  CROSSJOIN(DESCENDANTS([RESIDENCE].[All],
    [RESIDENCE].[City], SELF_AND_BEFORE),
  CROSSJOIN(DESCENDANTS([RACE].[All],
    [RACE].[RaceGroup], SELF_AND_BEFORE),
    [OCCUPATION].[Occ].Members)) ON ROWS
FROM [CENSUS] WHERE [TIME].[Year].[2009]
PREFERRING AvgIncome BETWEEN 500 AND 1000
AND        RESIDENCE CONTAIN State
```

Example of constructors

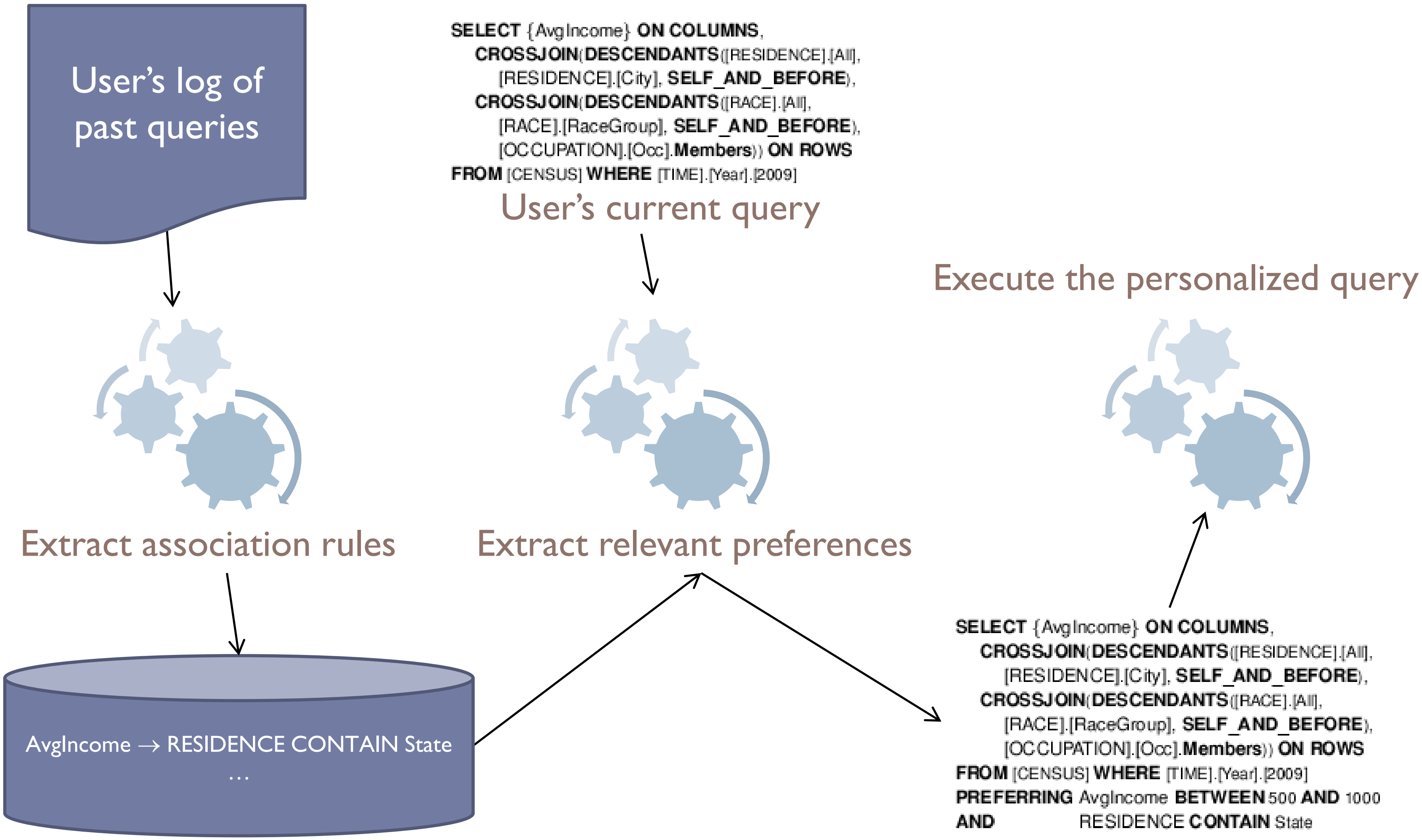


- POS(City,LA)
  - (LA,all,2010,F,all) > (NY,all,all,all,all)
    (California,all,2009,all,all) > (NY,all,2010,all,all)
- CONTAIN(RESIDENCE,City)
  - (LA,all,2010,F,all) > (California,all,2009,all,all)

Example of dominations



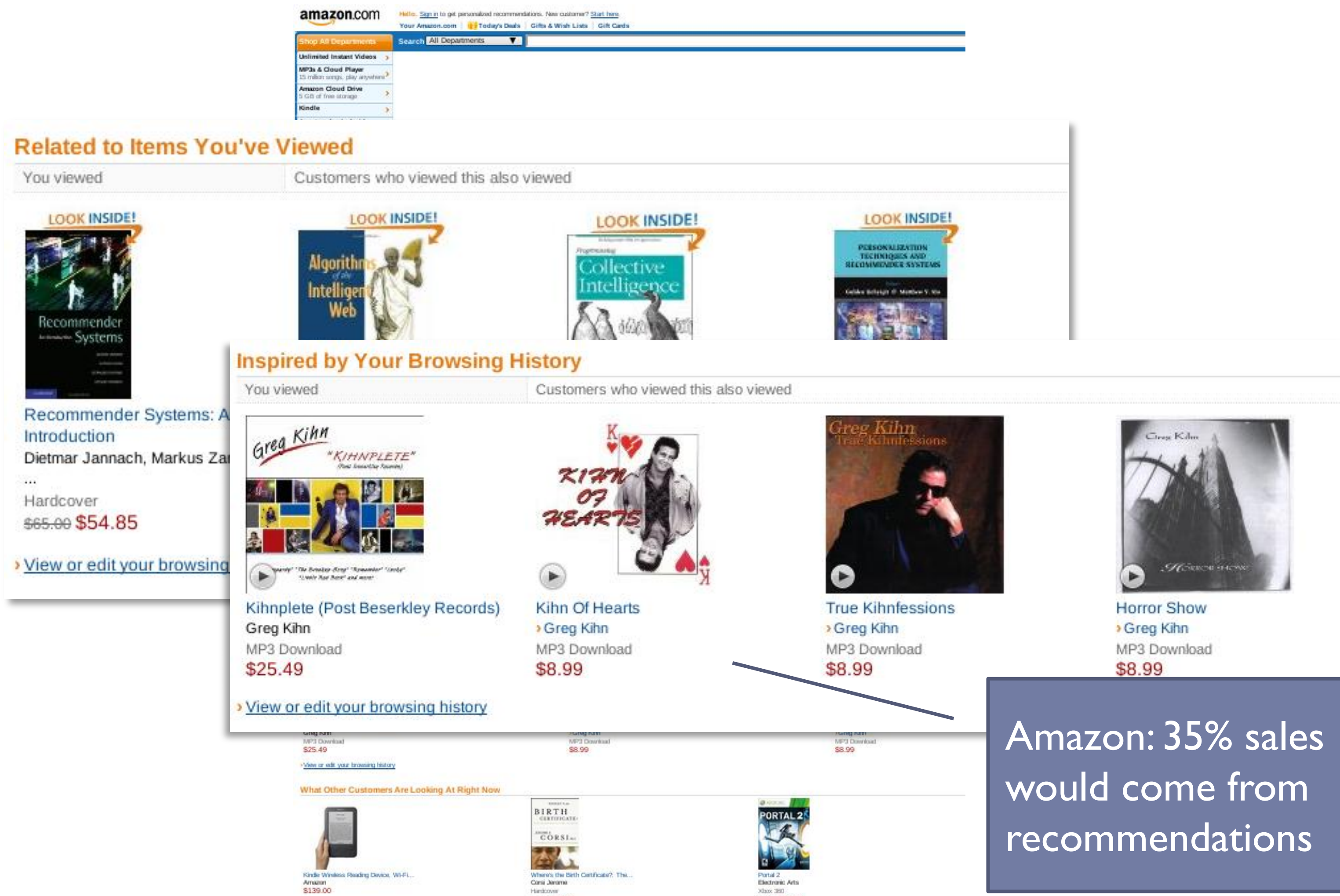
Improving proactiveness [Aligon & al, 2011]



Query recommendation

Basics of recommender systems

Recommender systems



Amazon: 35% sales would come from recommendations

The basic model

interest	Item 1	Item 2	Item 3	…	Item m
User 1	0.3		0.9	…	0.7
User 2		0.4	0.8	…	0.6
User 3					
…	…	…	…	…	…
User n	0.9	0.5		…	0.2

- A matrix customers * items recording the interests
- Recommend the items having highest ratings
- But
  - Ratings are hard to find
  - Matrix is huge and sparse
  - Everyone is a bit eccentric [WSDM 2010]

Three classical approaches

- Content-based
  - Recommend items similar to those highly rated
- Collaborative
  - Recommend items highly rated by similar users
- Hybrid
  - Combine content-based and collaborative

- A *lot* of works in the areas of e-commerce, Web, IR, …
  - See e.g., "Recommender systems handbook", Springer, 2011

Example of content-based recommendations
1. build item profiles

	Donuts	Duff	Apple	Tofu	Water	Bud	Ribs
Homer	0.9	0.8				0.7	
Marge			0.8		0.6		
Bart	0.7	0.6	0.1				0.8
Lisa	0.2			0.8	0.6		
Maggie	0.6			0.5	0.6		

- Features: contains sugar, ok for diet
- Profile of Donuts: (0.9,0)
- Profile of Duff: (0.6,0.1)
- Profile of Apple: (0.4,0.6)
- Profile of Tofu: (0,0.9)
- …

Example of content-based recommendations 2. build user profiles

	Donuts	Duff	Apple	Tofu	Water	Bud	Ribs
Homer	0.9	0.8				0.7	
Marge			0.8		0.6		
Bart	0.7	0.6	0.1				0.8
Lisa	0.2			0.8	0.6		
Maggie	0.6			0.5	0.6		

- Features: contains sugar, ok for diet
- Profile of Homer: (0.9*(0.9,0) + 0.8*(0.6,0.1) …)/3
  - = (0.8,0.1)
- Profile of Lisa: (0.3,0.8)
- …

Example of content-based recommendations 3. compare profiles to score

	Donuts	Duff	Apple	Tofu	Water	Bud	Ribs
Homer	0.9	0.8				0.7	
Marge			0.8		0.6		
Bart	0.7	0.6	0.1				0.8
Lisa	0.2			0.8	0.6		
Maggie	0.6			0.5	0.6		

- Compare Homer profile to Apple profile:
  - cosine((0.8,0.1),(0.4,0.6)) =0.33
- Compare Homer profile to Tofu profile
  - cosine((0.8,0.1),(0,0.9)) =0.1
- …
- In the end, recommend Ribs to Homer, Apple to Lisa

Example of collaborative recommendations 1. find similar users

	Donuts	Duff	Apple	Tofu	Water	Bud	Ribs
Homer	0.9	0.8				0.7	
Marge			0.8		0.6		
Bart	0.7	0.6	0.1				0.8
Lisa	0.2			0.8	0.6		
Maggie	0.6			0.5	0.6		

- Find similar users
  - Compare Homer and Marge
    - Cosine((0.9,0.8,0,…),(0,0,0.8,…))
  - Compare Homer and Bart
    - Cosine((0.9,0.8,0,…),(0.7,0.6,0.1,…))
  - …

Example of collaborative recommendations 2. compute scores

	Donuts	Duff	Apple	Tofu	Water	Bud	Ribs
Homer	0.9	0.8				0.7	*
Marge			0.8		0.6		
Bart	0.7	0.6	0.1				0.8
Lisa	0.2			0.8	0.6		
Maggie	0.6			0.5	0.6		

- Recommend items highly rated by similar users
  - Rating weighted with similarity score
    - Cosine(Homer,Bart)

Existing approaches

In relational databases

How to recommend? [Stefanidis & al., 2009]

- Use current state of the database
  - Find correlated attributes, most frequent values, etc.
- Use history (query log)
  - Compute similarities among users, similarities among queries
- Use external data
  - E.g., wikipedia, etc.

YMAL [Stefanidis & al., 2009] Example

- Local analysis
  - *Select title, genre from Movies where actor='C. Lee'*
  - The result has a lot of genre='fantastic'
  - Recommend:
    - *Select title, genre from Movies where genre='fantastic'*
- Global analysis
  - Value 'Allen' of attribute Director is correlated with value 'Comedy' of attribute Genre
  - *Select * from Movies where director='Allen'*
  - Recommend:
    - *Select * from Movies where genre='Comedy'*

QueRIE [Chatzopoulou & al., 2009]

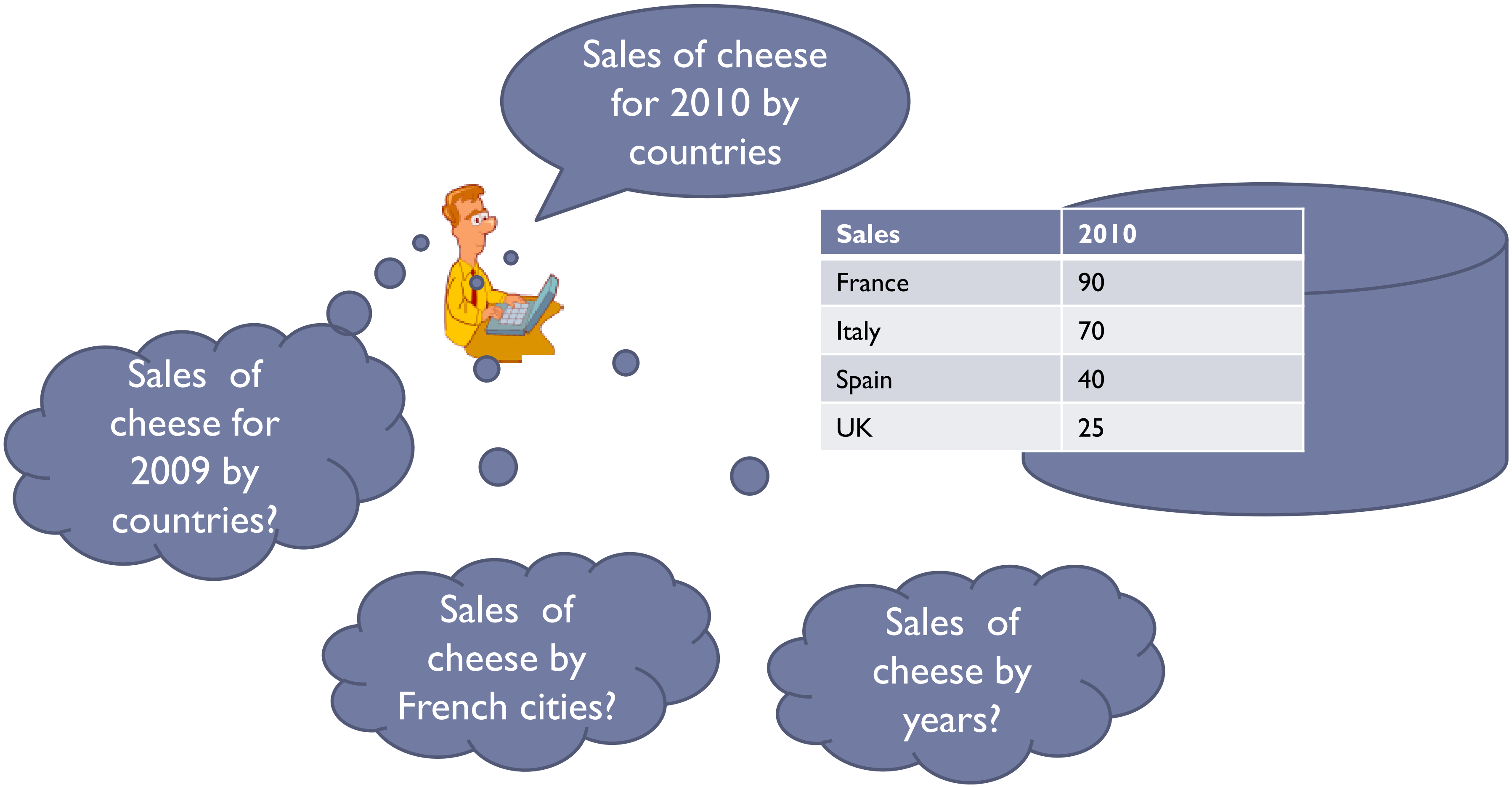
	Tuple 1	Tuple 2	Tuple 3	…	Tuple n
Session 1	1	0	0		0
Session 2	0	1	1		1
Session 3	0	0	0		1
…					
Session m	1	1	0		0

- Current session S_c=(1,…,0)
- Find session S the most similar to S_c using cosine
- Recommend the query of S that is the most similar to S_c

Existing approaches

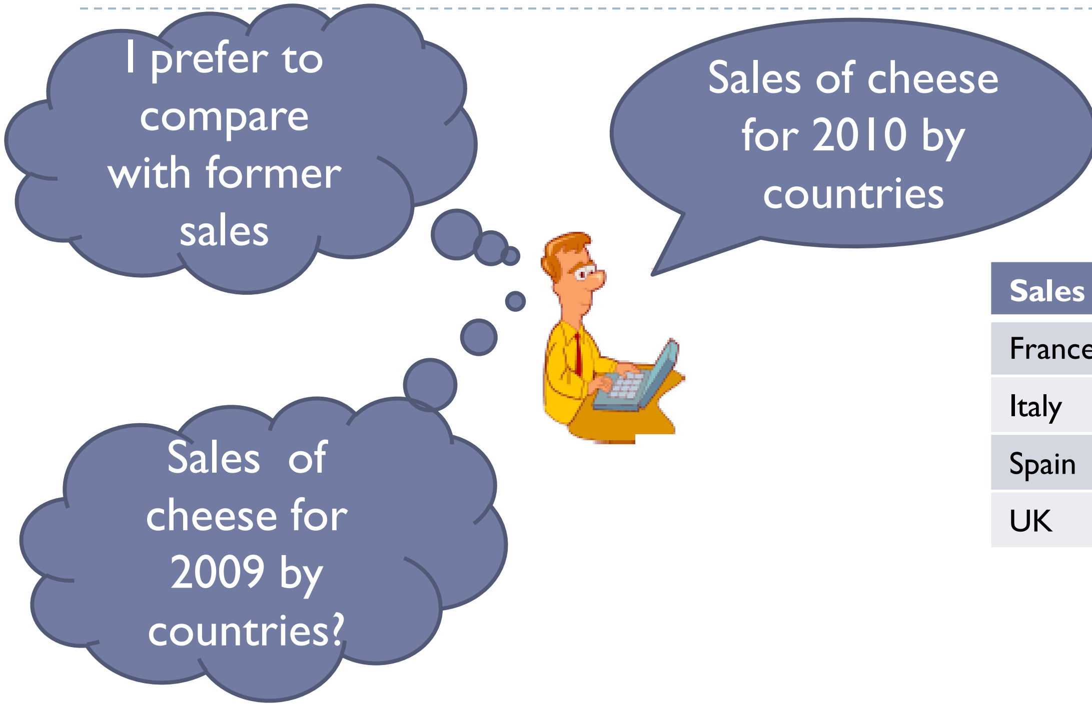
In multidimensional databases

Why recommendation?



Sales	2010
France	90
Italy	70
Spain	40
UK	25

Profile?



Sales	2010
France	90
Italy	70
Spain	40
UK	25

Expectations?

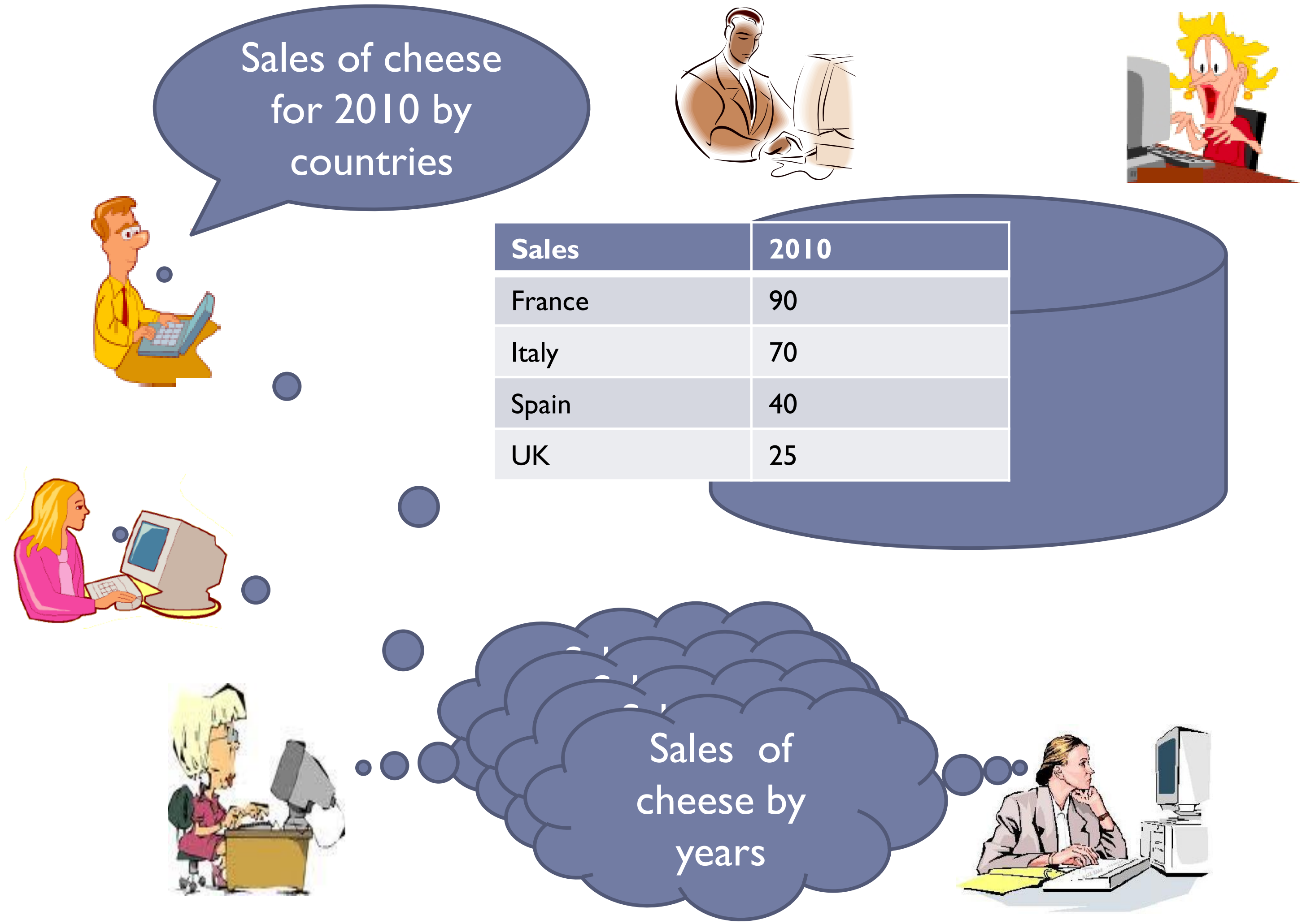
I expect sales to be uniformly distributed

Sales of cheese for 2010 by countries

Sales	2010
France	90
Italy	70
Spain	40
UK	25

Sales of cheese by French cities

Others?



Sales	2010
France	90
Italy	70
Spain	40
UK	25

Four different approaches

1. Content-based methods based on user preferences
   - Current state, with external data
2. Content-based methods based on expectations
   - Current state
3. Collaborative methods based on a query log
   - History-based
4. Collaborative methods based on log and expectations
   - Current state and history-based

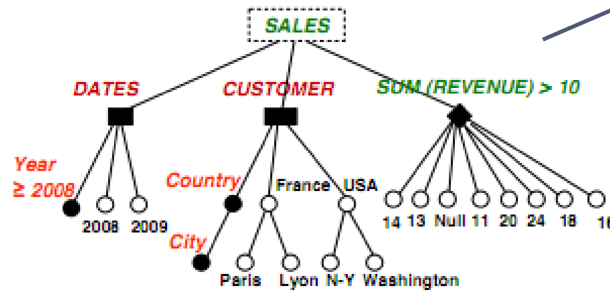
- All approaches:
  - Low formulation effort, prescriptive, proactive, low expressiveness

1. Preference-based recommendations [Jerbi & al., 2009]

If query concerns 2009, score of Barcelona is 0.9
If query concerns N-Y, score of SUM(REVENUE)>5 is 0.8
If query concerns 2009, score of Madrid is 0.4
If query concerns 2010, score of Paris is 0.3
…

The preferences

The query



Recommend:
Add Barcelona to the list of cities
Change SUM(REVENUE)>10 by SUM(REVENUE)>5

2. Expectation-based recommendations Discovery driven analysis [Sarawagi, 2000]

Sales	Quarter 1
Europe	100

The current query result

Not surprising, do not recommend it

Sales	Quarter 1
France	25
Italy	25
Spain	25
UK	25

Surprising, recommend it

Sales	Jan	Feb	Mar
Europe	80	10	10

2. Expectation-based recommendations Discovery driven analysis [Cariou & al., 2008]

Sales	All, All
France	10
UK	20

The current query result

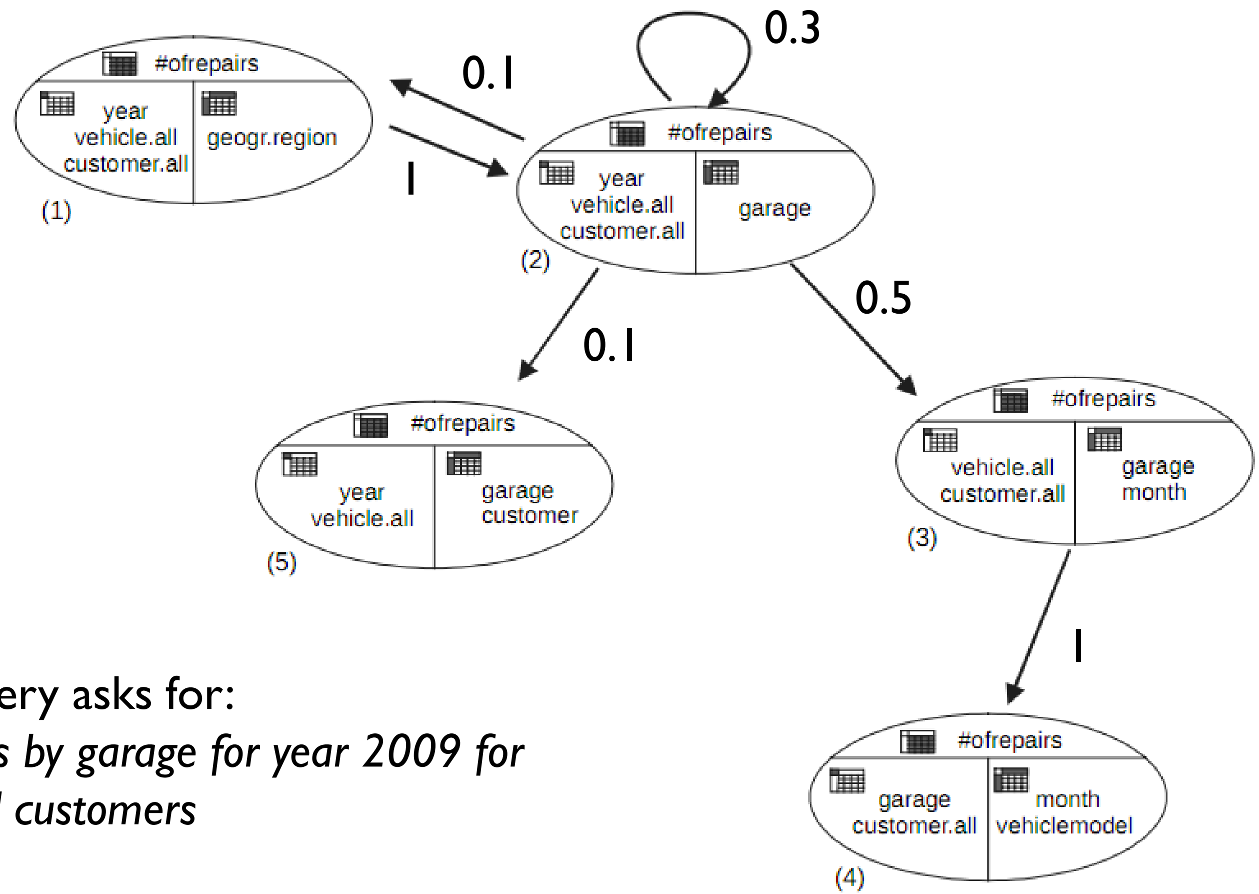
Not surprising, do not recommend it

Sales	Drink	Food
France	7	3
UK	4	16

Surprising, recommend it

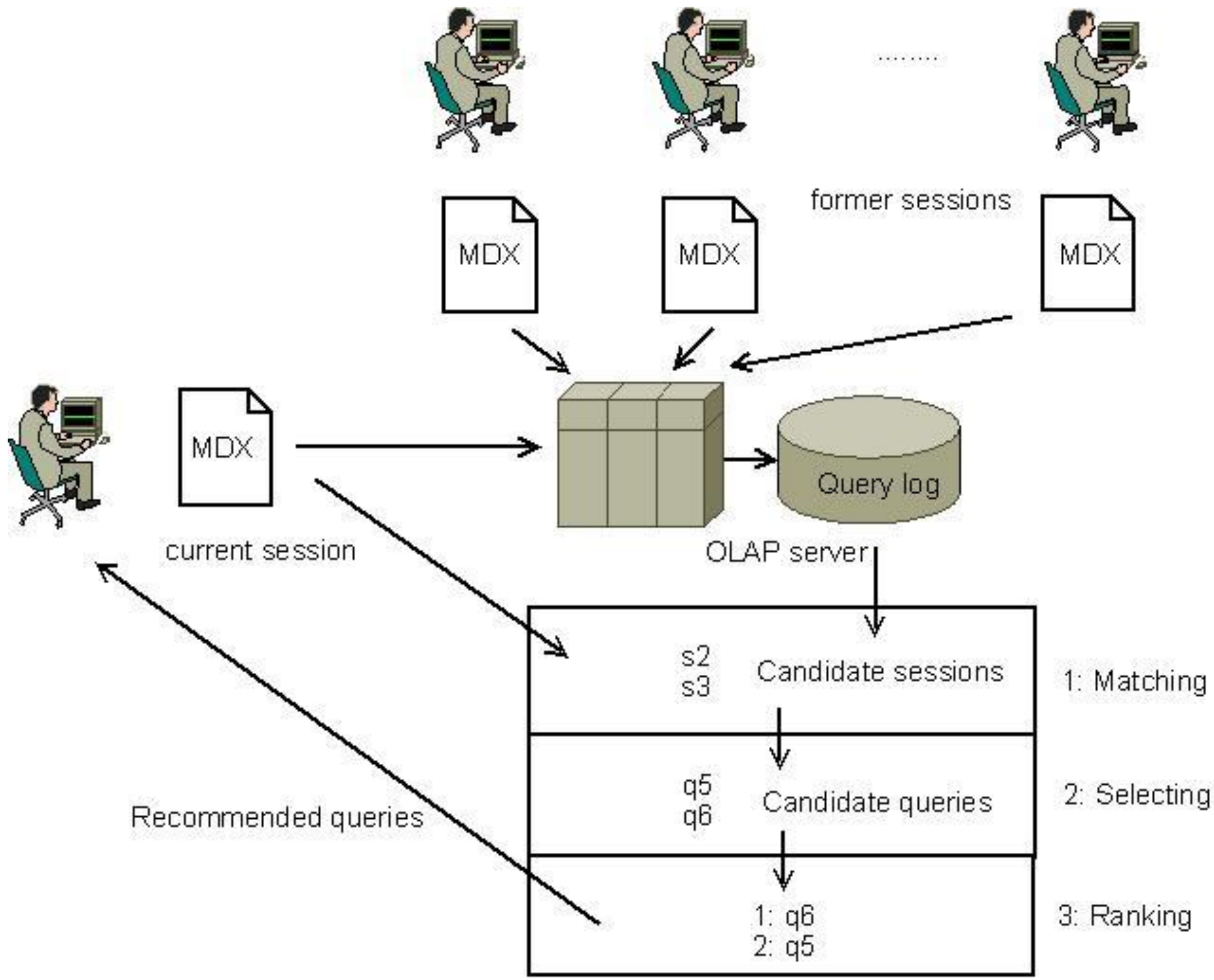
Sales	2009	2010
France	8	2
UK	16	4

3. Log-based recommendations Promise [Sapia, 2000]



If the current query asks for:
Number of repairs by garage for year 2009 for all vehicles and all customers
Recommend:
drilldown to month

3. Log-based recommendations [Giacometti & al., 2009]



3. Log-based recommendations [Giacometti & al., 2009]

- Distances proposed
  - Between positions in a cube
    - Hamming
    - Based on the shortest path in the dimension
  - Between queries
    - Based on dimension-wise differences
    - Hausdorff
  - Between sessions
    - Based on the subsequence
    - Edit distance

More distances? See "Similarity measures for multidimensional data" by Baikousi, Rogkakos, Vassiliadis at ICDE 2011

4. Log and expectation-based recommendations [Giacometti & al., 2009]

Session 1

Query 1

france	cheese	milk	butter
2007 sem 1	25	5	10
2007 sem 2	25	10	20
2008 sem 1	1	10	30
2008 sem 2	5	5	40

Query 2

france	cheese	milk
2007 sem 1	25	5
2007 sem 2	25	10
2008 q1	0.5	5
2008 q2	0.5	5
2008 q3	2	3
2008 q4	3	2

Session 2

Query 1

cheese	all
2006	100
2007	200

Query 2

cheese	all
2007	200
2008	20

Query 3

cheese	France	Italy	Spain
2007	50	1	1
2008	6	2	1

Query 4

Normandie	cheese
2007	0
2008	1

Query 5

Loire Valley	cheese
2007	40
2008	4

Session 3

Query 1

all	goat cheese
2005	10
2006	11
2007	10
2008	11

Query 2

all	cheese
2005	50
2006	100
2007	200
2008	20

Query 3

all	dairy
2005	100
2006	200
2007	300
2008	300

OLAP server query log

Interesting…

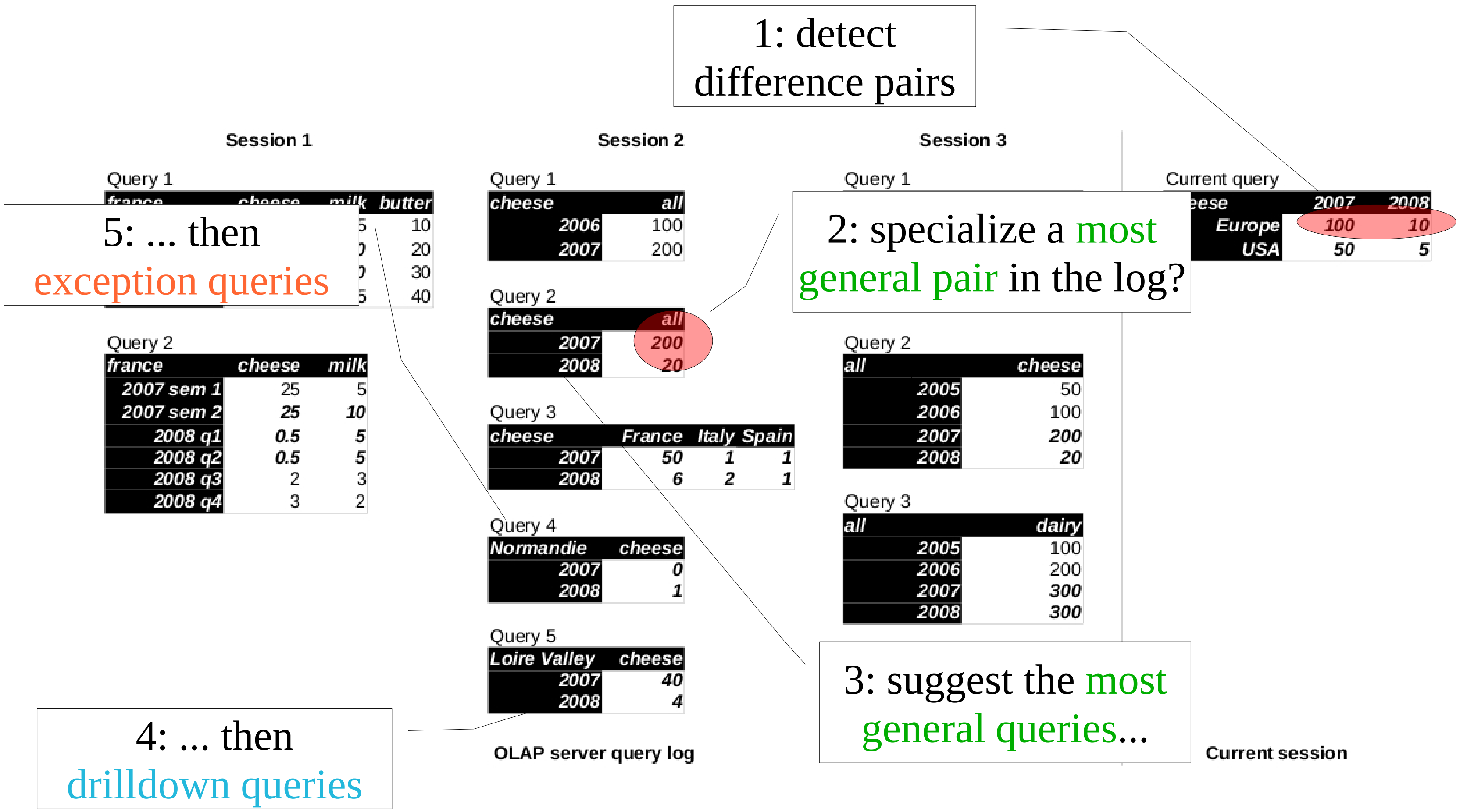
Hm this looks strange to me…

Current query

cheese	2007	2008
Europe	100	10
USA	50	5

Current session

4. Log and expectation-based recommendations [Giacometti & al., 2009]



Conclusion

Conclusion

- So far…
  - Given q, compute q' such that q' $\subset$ q or q $\not\subset$ q', q $\not\subset$ q'
- The best approach?
  - Low formulation effort, proactive, not prescriptive, high expressiveness… yet to be proposed!
  - Collaborative for naïve user, content-based for advanced user
- How about effectiveness?
  - Need to categorize database user's navigational behavior
    - A taxonomy exists in the web but not in databases…

Some open issues

- Some open issues
  - How to learn preferences? Navigational habits?
  - Can preferences be revised? What if I don't know what I prefer?
  - What about privacy?
  - How to handle preferences on data distribution?
  - How to assess the quality of a recommendation?
  - What recommendation in what context?
  - When are two sessions similar?
  - How to guess the intent of a query?
  - …

Bibliography

Bibliography

- Motivation
  - “The data deluge”, The economist (2010)
  - H.V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, C. Yu: “Making database systems usable”, SIGMOD (2007)
  - N. Khoussainova, M. Balazinska, W. Gatterbauer, Y. Kwon, D. Suciu: “A Case for A Collaborative Query Management System”, CIDR (2009)
- Surveys
  - G. Koutrika, Y. Ioannidis, “Personalized systems, from an IR and DB perspective”, tutorial at ICDE (2005)
  - K. Stefanidis, G. Koutrika, E. Pitoura, “A Survey on Representation, Composition and Application of Preferences in Database Systems”, ACM Transactions on Database Systems, to appear
  - G. Adomavicius, A. Tuzhilin: “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, IEEE Trans. Knowl. Data Eng. (2005)

Bibliography on preferences

- In relational databases
  - Preference Formulas
    - J. Chomicki, "Preference Formulas in Relational Queries", ACM Transactions on Database Systems, 28(4) (2003)
  - Skyline Operator
    - S. Börzsönyi, D. Kossmann & K. Stocker, "The Skyline Operator", ICDE (2001)
  - Preference SQL
    - W. Kießling, G. Köstler, "Preference SQL - Design, Implementation, Experiences", VLDB (2002)
  - Query personalisation
    - G. Koutrika, Y. Ioannidis. "Personalization of Queries in Database systems", ICDE (2004)

Bibliography on preferences

- In multidimensional databases

  - S. Rizzi. "OLAP Preferences: a research agenda". DOLAP (2007)
  - P. Biondi, M. Golfarelli, S. Rizzi. "myOLAP: An Approach to Express and Evaluate OLAP Preferences". IEEE TKDE, to appear
  - L. Bellatreche, A. Giacometti, D. Laurent, P. Marcel, H. Mouloudi "A Personalization Framework for OLAP Queries", DOLAP (2005)
  - J. Aligon, M. Golfarelli, P. Marcel, S. Rizzi, E. Turricchia. "Mining Preferences from OLAP Query Logs for Proactive Personalization", ADBIS (2011)

Bibliography on recommendation

- Existing approaches in relational databases
  - YMAL
    - Kostas Stefanidis, Marina Drosou, Evaggelia Pitoura, "You May Also Like Results in Relational Databases", PersDB (2009)
  - QueRIE
    - Gloria C., M. Eirinaki, N. Polyzotis, "Query Recommendations for Interactive Database Exploration", SSDBM (2009)
    - J. Akbarnejad, G. Chatzopoulou, M. Eirinaki, S. Koshy, S. Mittal, D. On, N. Polyzotis, J. Swarubini Vindhiya Varman, "SQL QueRIE Recommendations", PVLDB (2010)
  - Recommending join queries
    - X. Yang, C. M. Procopiuc, D. Srivastava, "Recommending Join Queries via Query Log Analysis", ICDE (2009)
  - SnipSuggest
    - N. Khoussainova, Y. Kwon, M. Balazinska, D. Suciu, "SnipSuggest: Context-Aware Autocompletion for SQL", PVLDB (2010)

Bibliography on recommendation

- Existing approaches in multidimensional databases
  - Expectation-based

    - S. Sarawagi, "Explaining Differences in Multidimensional Aggregates", VLDB (1999)
    - S. Sarawagi, "User-Adaptive Exploration of Multidimensional Data", VLDB (2000)
    - G. Sathe, S. Sarawagi, "Intelligent Rollups in Multidimensional OLAP Data", VLDB (2001)
    - V. Cariou, J. Cubillé, C. Derquenne, S. Goutier, F. Guisnel, H. Klajnmic, "Built-In Indicators to Discover Interesting Drill Paths in a Cube", DaWaK (2008)

  - Preference-based

    - H. Jerbi, F. Ravat, O. Teste, G. Zurfluh, "Preference-Based Recommendations for OLAP Analysis", DaWaK (2009)

  - Log-based

    - C. Sapia, "PROMISE: Predicting Query Behavior to Enable Predictive Caching Strategies for OLAP Systems", DaWaK (2000)
    - A. Giacometti, P. Marcel, E. Negre, "Recommending Multidimensional Queries", DaWaK (2009)

  - Log and expectation-based

    - A. Giacometti, P. Marcel, E. Negre, A. Soulet, "Query recommendations for OLAP discovery driven analysis", IJDWM (2011)
