Expressing OLAP Preferences

Matteo Golfarelli and Stefano Rizzi

DEIS, University of Bologna, Italy
`matteo.golfarelli@unibo.it, stefano.rizzi@unibo.it`

Abstract. Multidimensional databases play a relevant role in statistical and scientific applications, as well as in business intelligence systems. Their users express complex OLAP queries, often returning huge volumes of facts, sometimes providing little or no information. Thus, expressing preferences could be highly valuable in this domain. The OLAP domain is representative of an unexplored class of preference queries, characterized by three peculiarities: preferences can be expressed on both numerical and categorical domains; they can also be expressed on the aggregation level of facts; the space on which preferences are expressed includes both elemental and aggregated facts. In this paper we propose a preference algebra for OLAP, that takes into account the three peculiarities above.

Keywords: OLAP, database preferences, multidimensional databases.

1 Introduction and Motivation

Personalizing e-services by allowing users to express preferences is becoming more and more common. When querying, expressing preferences is a natural way to avoid empty results on the one hand, information flooding on the other.

Though a lot of research has been carried out during the last few years on database preferences (e.g., [1,2]), the problem of developing a theory of preferences for multidimensional databases has been mostly neglected so far. We argue that expressing preferences could be valuable in this domain because:

- Preferences enable users to focus on the most interesting data. This is particularly beneficial in the OLAP context, since multidimensional databases typically store a huge amount of data. Besides, OLAP queries have a complex structure. An OLAP query may easily return huge volumes of data, or it may return little or no information as well. The data ranking entailed by preferences allows users to cope with both these problems.
- During an OLAP session, the user may not exactly know what she is looking for. The reasons behind a specific phenomenon or trend may be hidden, and finding those reasons by manually applying different combinations of OLAP operators may be very frustrating. Preferences enable users to specify the pattern she is searching for. Since preferences express soft constraints, the most similar data will be returned when no data exactly match that pattern. From this point of view, preference queries can be regarded as a basic OLAM (OnLine Analytical Mining) technique [3].

- Scientific data are often distributed across separate databases. In the business domain, federated data warehouse architectures are seen as an effective shortcut to integration. In these cases, schema heterogeneity may prevent from expressing distributed queries. Conversely, a schema preference can produce meaningful results even when a common schema is not defined.

It is well-known that aggregation plays an essential role in OLAP queries, since it enables decision-makers to get valuable, summary information out of the huge quantity of detailed data available in multidimensional databases. OLAP queries do not only formulate selections and projections on attributes and measures, they also specify on what hierarchy attributes data are to be aggregated (*group-by set*). The aggregation level has a strong impact on the size of the result returned to the user, and its inappropriate setting may end in either obtaining very coarse, useless information or being flooded by tons of detailed data, which is particularly critical when working with devices with small bandwidth and limited visualization capabilities. For this reason we argue that, in the OLAP domain, users may wish to express their preferences on group-by sets too, for instance by stating that monthly data are preferred to yearly and daily data.
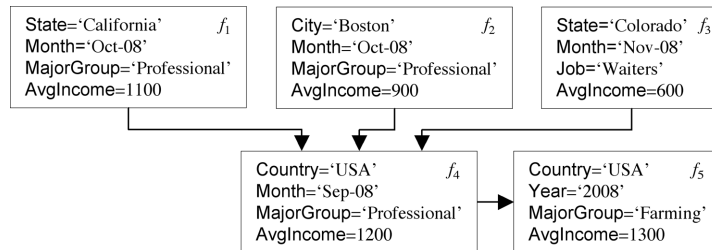

Fig. 1. Sample census facts and preference relationships between them

Example 1. IPUMS is a public database storing census microdata for social and economic research [4]. An analyst may wish to understand the reasons behind a decrease in the average income of US citizens. She suspects that this is a state-scale phenomenon mainly due to a decrease in the income of professionals. So she expresses a query with a preference on data of professionals, grouped by state, with low income (i.e. lower than $ 1000). To evaluate this preference, it is necessary to compare data characterized by different group-by sets and different values for attributes and measures. If the analyst is right, this query will return only the facts, aggregated by states, where professionals have an average income lower than $ 1000. In Figure 1, instead, we assume that the analyst's hypothesis is false and three relevant situations are pointed out. Fact f_1 shows that the lowest monthly income of professionals at the state-scale is in California, and it not that low. Fact f_2 shows significantly low income for professionals in the city of Boston. Fact f_3 shows that, in Colorado, waiters have very low income. Fact f_4 is worse than the previous three facts because it yields higher income, is not

aggregated by state, and is not related to professionals. Finally, f_5 is worse than f_4 because it yields an even higher income.

From Example 1, it is apparent that the OLAP domain is representative of an unexplored class of preference queries, characterized by three peculiarities:

- Preferences can be expressed on both attributes and measures, that respectively have categorical and numerical domains. This makes the existing approaches, that are mainly geared to handling either only categorical or numerical preferences, ineffective.
- Preferences can also be formulated on the aggregation level of data, which comes down to expressing preferences on schema rather than on instances. To the best of our knowledge, no approach includes this feature.
- The space on which preferences are declared includes both elemental and aggregated facts. In relational OLAP implementations, materializing all facts on *all* possible group-by sets is highly undesirable.

In this paper we present an approach for dealing with OLAP preferences. Specifically, we propose an algebra for expressing complex OLAP preferences including a set of base preference constructors on attributes, measures and hierarchies, composed by the Pareto operator. The most original of the domain-dependent aspects of our algebra is the possibility of declaring preferences on group-by sets, which is done by recognizing that preferences on the space of hierarchy attributes induce preferences on the space of facts.

2   Background Definitions and Working Example

In this section we introduce a basic formal setting to manipulate multidimensional data. To keep the formalism simpler, and without actually restricting the validity of our approach, we will consider hierarchies without branches, i.e., consisting of chains of attributes.

Definition 1 (Multidimensional Schema). *A* multidimensional schema *(or, briefly, a* schema*) is a triple* $\mathcal{M} = \langle A, H, M \rangle$ *where:*

- $A = \{a_1, \ldots a_p\}$ *is a finite set of* attributes*, each defined over a categorical domain* $Dom(a_k)$*;*
- $H = \{h_1, \ldots, h_n\}$ *is a finite set of* hierarchies*, each characterized by (1) a subset* $Attr(h_i) \subseteq A$ *of attributes (such that the* $Attr(h_i)$*'s for* $i = 1, \ldots, n$ *define a partition of A); (2) a* roll-up *total order* $\succeq_{h_i}$ *over* $Attr(h_i)$*; and (3) a family of roll-up functions including a function* $RollUp^{a_k}_{a_j} : Dom(a_k) \rightarrow Dom(a_j)$ *for each pair of attributes* a_k *and* a_j *such that* $a_k \succeq_{h_i} a_j$*;*
- *a finite set of* measures $M = \{m_1, \ldots, m_l\}$*, each defined over a numerical domain* $Dom(m_i)$*.*

For each hierarchy h_i, the top attribute of the order is denoted by DIM_i, and determines the finest aggregation level of the hierarchy. Conversely, the bottom attribute is denoted by ALL_i and determines the coarsest aggregation level.

Definition 2 (Group-by Set). *Given schema* $\mathcal{M} = \langle A, H, M \rangle$*, let* $Dom(H) = Attr(h_1) \times \ldots \times Attr(h_n)$*; each* $G \in Dom(H)$ *is called a* group-by set *of* $\mathcal{M}$*. Let* $G = \langle a_{k_1}, \ldots, a_{k_n} \rangle$ *and* $Dom(G) = Dom(a_{k_1}) \times \ldots \times Dom(a_{k_n})$*; each* $g \in Dom(G)$ *is called a* coordinate *of* G*. We denote with* $G.h_i = a_{k_i}$ *the attribute of* h_i *included in* G*.*

Let $\succeq_H$ denote the product order of the roll-up orders over the hierarchies in H. Then, $(Dom(H), \succeq_H)$ is a lattice, that we will call *group-by lattice*, whose top and bottom elements are $G^\top = \langle DIM_1, \ldots, DIM_n \rangle$ and $G^\perp = \langle ALL_1, \ldots, ALL_n \rangle$, respectively.

Example 2. Our working schema is CENSUS, that includes hierarchies RES (for "Residence"), OCC (for "Occupation"), and TIME, and measure AvgIncome. The roll-up orders are as follows:

$$\text{City} \succeq_{\text{RES}} \text{State} \succeq_{\text{RES}} \text{Country} \succeq_{\text{RES}} \text{AllCountries}$$

$$\text{Job} \succeq_{\text{OCC}} \text{MinorGroup} \succeq_{\text{OCC}} \text{MajorGroup} \succeq_{\text{OCC}} \text{AllGroups}$$

$$\text{Month} \succeq_{\text{TIME}} \text{Quarter} \succeq_{\text{TIME}} \text{Year} \succeq_{\text{TIME}} \text{AllYears}$$

For instance, it is $RollUp_{\text{State}}^{\text{City}}(\text{'Miami'}) = \text{'Florida'}$. Some examples of group-by sets are $G^\top = \langle \text{City}, \text{Job}, \text{Month} \rangle$, $G_1 = \langle \text{Country}, \text{Job}, \text{Month} \rangle$, $G_2 = \langle \text{State}, \text{AllGroups}, \text{Quarter} \rangle$, and $G^\perp = \langle \text{AllCountries}, \text{AllGroups}, \text{AllYears} \rangle$. It is $G^\top \succeq_H G_1 \succeq_H G^\perp$, $G^\top \succeq_H G_2 \succeq_H G^\perp$, while G_1 and G_2 are incomparable according to $\succeq_H$. A coordinate of G_1 is $\langle \text{'USA'}, \text{'Dentist'}, \text{'Oct-08'} \rangle$.

A schema is populated with facts. A fact is characterized by a group-by set G that defines its aggregation level, by a coordinate of G, and by a value for each measure. While the facts at the top group-by set of the lattice (*primary facts*) are those storing elemental data, those at the other group-by sets store summarized information.

Definition 3 (Fact). *Given schema* $\mathcal{M} = \langle A, H, M \rangle$ *and a group-by set* $G \in Dom(H)$*, a fact at* G *is a triple* $f = \langle G, g, v \rangle$*, where* $g \in Dom(G)$ *and* $v \in Dom(M) = Dom(m_1) \times \ldots \times Dom(m_l)$*. The space of all facts of* $\mathcal{M}$ *is* $\mathcal{F}_\mathcal{M} = \bigcup_{G \in Dom(H)}(\{G\} \times Dom(G) \times Dom(M))$*.*

Example 3. An example of fact of CENSUS is $f = \langle G_1, \langle \text{'USA'}, \text{'Dentist'}, \text{'Oct-08'} \rangle, 4000 \rangle$.

Finally, an instance of schema $\mathcal{M}$ is a *datacube* and is defined as a set of facts $C \subset \mathcal{F}_\mathcal{M}$. Intuitively, C includes a set of primary facts at the top group-by set of $Dom(H)$, united with all the other facts determined by aggregating primary facts at all the other group-by sets in $Dom(H)$.

3   Preferences on Facts

Before we start to deal with preferences, we briefly recall that a *strict partial order* (s.p.o.) on a given set S is an irreflexive and transitive (thus asymmetric)

binary relation on the elements of S. A negatively transitive s.p.o. is also called a *weak order* (w.o.). A w.o. on S partitions S into n (disjoint) levels such that the levels are totally ordered and each level determines an SV-relation on the w.o. itself.

In relational databases, a preference is commonly defined as a s.p.o. over the set of possible tuples. Here, we define a preference as a s.p.o. on the space of *all facts at all group-by sets*, which implies that a preference may involve two facts defined at different group-by sets.

Definition 4 (Preference). *Given schema $\mathcal{M}$, a preference P on $\mathcal{M}$ is a couple $(<_P, \cong_P)$ where $<_P \subseteq \mathcal{F}_\mathcal{M} \times \mathcal{F}_\mathcal{M}$ is a s.p.o. and $\cong_P \subseteq \mathcal{F}_\mathcal{M} \times \mathcal{F}_\mathcal{M}$ is an SV-relation on $<_P$.*

The semantics of $f_1 <_P f_2$ is that f_2 is preferred to f_1; the semantics of $f_1 \cong_P f_2$ is that f_2 is equivalent (or *substitutable*) to f_1 [5].

In our approach, complex preferences on the space of facts are inductively engineered by applying a set of base preference constructors and an operator for preference composition. In particular, a preference is defined by a *preference expression q* ruled by the following grammar:

$$< expr >::= < baseConstr > \mid < expr > \otimes < baseConstr >$$
$$< baseConstr >::= \text{POS}|\text{NEG}|\text{BETWEEN}|\text{LOWEST}|\text{HIGHEST}|$$
$$\text{CONTAIN}|\text{NEAR}|\text{COARSEST}|\text{FINEST}$$

where base preference constructors operate either on attributes, measures, or hierarchies. Adopting the SV-semantics allows for closing the set of composition operators on the set of preferences, thus obtaining an algebra [5].

In the next subsections we will introduce the set of base preference constructors and the composition operator we provide. For simplicity, in this work base preferences are defined over single attributes, measures and hierarchies; the extension to multiple attributes, measures and hierarchies is straightforward and smoothly supported by our approach. Besides, for space reasons, the formalization of some base constructors will be omitted.

3.1   Base Preferences on Attributes

While in the relational case each tuple is characterized by the same attributes, the attributes that characterize a fact depend on its group-by set. For instance, a fact reporting the average income for the California state does not explicitly provide values for City and Country. On the other hand, hierarchies allow for relating values of attributes belonging to the same hierarchy by means of roll-up functions. In order to avoid introducing an undesired relationship between preferences on attributes and preferences on hierarchies, in this work we use roll-up functions to propagate preferences expressed on attributes along the *whole* hierarchy, as explained in the following.

Given fact f and hierarchy h, let $\bar{a}$ be the attribute of h included in the group-by set of f. Then, let $\bar{c}$ be the (categorical) value assumed by $\bar{a}$ in the

coordinate of f. Given any attribute $a \in Attr(h)$, we denote with $f.a \in 2^{Dom(a)}$ the value(s) assumed in f by a, defined as follows:

$$f.a = \begin{cases} \{Rollup_a^{\bar{a}}(\bar{c})\}, & \text{if } \bar{a} \succeq_h a \\ \{c \in Dom(a) | Rollup_{\bar{a}}^a(c) = \bar{c}\}, & \text{otherwise} \end{cases}$$

For instance, if $\bar{a} = \mathsf{State}$ and $\bar{c} = $ 'California' for fact f, it is $f.\mathsf{City} = $ {'LosAngeles', 'S. Francisco', . . .} and $f.\mathsf{Country} = $ {'USA'}.

Let $c \in Dom(a)$; the base preference constructors we provide for declaring preferences on a are:

– $\mathsf{POS}(a, c)$. Facts whose coordinate on h maps to c are preferred to the others:

$$f_1 <_P f_2 \text{ iff } c \notin f_1.a \ \wedge \ c \in f_2.a$$
$$f_1 \cong_P f_2 \text{ iff } (c \notin f_1.a \ \wedge \ c \notin f_2.a) \vee (c \in f_1.a \ \wedge \ c \in f_2.a)$$

– $\mathsf{NEG}(a, c)$. Facts whose coordinate on h does not map to c are preferred to the others.

It is easy to verify that, for both POS and NEG constructors, $<_P$ is a w.o. and $\cong_P$ is an SV-relation on $<_P$; thus, the result is a preference according to Def. 4.

Example 4. $\mathsf{POS}(\mathsf{Month}, '\mathsf{Oct}\text{-}08')$ states that the monthly data of October 2008, the daily data for all days of October 2008, and the yearly data for 2008 are preferred to all the other facts.

3.2   Base Preferences on Measures

Let $m \in M$ be a measure. Let $v, v_{low}, v_{high} \in Dom(m)$ ($v_{low} \leq v_{high}$); we define

$$\Delta(v, [v_{low}, v_{high}]) = \begin{cases} 0 & \text{if } v \in [v_{low}, v_{high}] \\ v_{low} - v & \text{if } v < v_{low} \\ v - v_{high} & \text{if } v > v_{high} \end{cases}$$

Also, given fact f, we denote with $f.m \in Dom(m)$ the (numerical) value assumed in f by m. Let $v \in Dom(m)$; the base preference constructors for declaring preferences on measure m are, like in [2]:

– $\mathsf{BETWEEN}(m, v_{low}, v_{high})$. Facts whose value on m is between v_{low} and v_{high} are preferred; the other facts are ranked according to their distance from the interval:

$$f_1 <_P f_2 \text{ iff } \Delta(f_1.m, [v_{low}, v_{high}]) > \Delta(f_2.m, [v_{low}, v_{high}])$$
$$f_1 \cong_P f_2 \text{ iff } \Delta(f_1.m, [v_{low}, v_{high}]) = \Delta(f_2.m, [v_{low}, v_{high}])$$

– $\mathsf{LOWEST}(m)$, $\mathsf{HIGHEST}(m)$. Facts whose value on m is as low (high) as possible are preferred.

All three constructors return w.o. preferences.

Example 5. $\mathsf{BETWEEN}(\mathsf{AvgIncome}, \mathsf{MININC}, 1000)$ states that the facts (aggregated at any group-by set) yielding average incomes lower than 1000 are preferred over the others, that are ranked according to increasing incomes.

3.3   Base Preferences on Hierarchies

As stated in the Introduction, one key feature of our approach is the possibility of declaring preferences on the aggregation level of facts, i.e., on their group-by sets. The basic idea is that of defining preferences on the space of hierarchy attributes, to let them induce preferences on the space of facts through a function that maps each fact into its group-by set. In particular, given fact f and hierarchy h, let $G(f)$ denote its group-by set and $G(f).h$ denote the attribute of h in $G(f)$.

Definition 5 (G-order and G-relation). *Let $<_{P'}$ and $\cong_{P'}$ be, respectively, an order and an equivalence relation on the attributes of h, $Attr(h)$. We call* G-order *the order $<_P$ induced on $\mathcal{F}_\mathcal{M}$ by $<_{P'}$ as follows: for each $f_1, f_2 \in \mathcal{F}_\mathcal{M}$, it is $f_1 <_P f_2$ iff $G(f_1).h <_{P'} G(f_2).h$. We call* G-relation *the equivalence relation $\cong_P$ induced on $\mathcal{F}_\mathcal{M}$ by $\cong_{P'}$ as follows: for each $f_1, f_2 \in \mathcal{F}_\mathcal{M}$, it is $f_1 \cong_P f_2$ iff $G(f_1).h \cong_{P'} G(f_2).h$.*

Theorem 1 shows that the properties of the relationships on hierarchy attributes are preserved in the relationships induced on facts through $G()$.

Theorem 1. *Let $<_{P'}$ and $\cong_{P'}$ be, respectively, an order and an equivalence relation on $Attr(h)$, and let $<_P$ and $\cong_P$ be their G-order and G-relation, respectively. Then, $(<_P, \cong_P)$ is a preference iff $(<_{P'}, \cong_{P'})$ is a preference. Besides, $<_P$ is a w.o. iff $<_{P'}$ is a w.o.*

We now introduce the notion of distance between two attributes in a hierarchy h, that is necessary for declaring NEAR preferences:

Definition 6 (Distance). *Let $a_1, a_2 \in Attr(h)$. The* distance *between a_1 and a_2, $Dist(a_1, a_2)$, is the difference between the levels of a_1 and a_2 within the roll-up total order $\succeq_h$.*

For instance, with reference to the CENSUS schema, it is $Dist(\mathsf{Month}, \mathsf{Year}) = 2$.
   Given $a, a_{fine}, a_{coarse} \in Attr(h)$, $a_{fine} \succeq_h a_{coarse}$, let

$$\Delta(a, [a_{fine}, a_{coarse}]) = \begin{cases} 0, \text{ if } a_{fine} \succeq_h a \succeq_h a_{coarse} \\ min\{Dist(a, a_{fine}), Dist(a, a_{coarse})\}, \text{ otherwise} \end{cases}$$

We are now ready to define the following base preference constructors on hierarchies. Let $h \in H$ and $a, a_{fine}, a_{coarse} \in Attr(h)$ $(a_{fine} \succeq_h a_{coarse})$:

- CONTAIN(h, a). The facts whose group-by set includes a are preferred to the others:

   $f_1 <_P f_2$ iff $a \neq G(f_1).h \ \wedge \ a = G(f_2).h$

   $f_1 \cong_P f_2$ iff $(a \neq G(f_1).h \ \wedge \ a \neq G(f_2).h) \vee (a = G(f_1).h \ \wedge \ a = G(f_2).h)$

- NEAR$(h, a_{fine}, a_{coarse})$. The facts whose group-by set along h is between a_{fine} and a_{coarse} are preferred; the other facts are ranked according to the distance of their group-by set along h from the interval:

   $f_1 <_P f_2$ iff $\Delta(G(f_1).h, [a_{fine}, a_{coarse}]) > \Delta(G(f_2).h, [a_{fine}, a_{coarse}])$

   $f_1 \cong_P f_2$ iff $\Delta(G(f_1).h, [a_{fine}, a_{coarse}]) = \Delta(G(f_2).h, [a_{fine}, a_{coarse}])$

– COARSEST(h). Aggregated facts along h are preferred to detailed ones:

$$f_1 <_P f_2 \text{ iff } G(f_1).h \succeq_h G(f_2).h$$
$$f_1 \cong_P f_2 \text{ iff } G(f_1).h = G(f_2).h$$

– FINEST(h). Detailed facts along h are preferred to aggregated ones.

All four constructors return w.o. preferences over $Attr(h)$ and, for Theorem 1, w.o. preferences over $\mathcal{F_M}$. FINEST returns the roll-up order on h, $\succeq_h$.

Example 6. CONTAIN(RES,State) selects a set of preferred group-by sets (those including State combined with any attribute of OCC and TIME). In other terms, it states that the census data aggregated by residence state are preferred to the others, regardless of what their aggregation is on the occupation and time hierarchies. NEAR(TIME,Quarter,Year) states that data aggregated by either quarter or year are preferred to the others. Data aggregated by month and data completely aggregated along the time hierarchy are substitutable. FINEST(TIME) ranks the group-by sets (and their facts) according to the roll-up lattice of TIME.

3.4   Preference Composition

The most common operator for preference composition is the Pareto operator:

– $P_1 \otimes P_2$ (Pareto composition). A fact is better than another if it is better according to one preference and better or substitutable according to the other (the composed preferences are considered equally important):

$$f_1 <_{P_1 \otimes P_2} f_2 \text{ iff } (f_1 <_{P_1} f_2 \ \wedge \ (f_1 <_{P_2} f_2 \ \vee f_1 \cong_{P_2} f_2))$$
$$\vee (f_1 <_{P_2} f_2 \ \wedge \ (f_1 <_{P_1} f_2 \ \vee \ f_1 \cong_{P_1} f_2))$$
$$f_1 \cong_{P_1 \otimes P_2} f_2 \text{ iff } f_1 \cong_{P_1} f_2 \ \wedge \ f_1 \cong_{P_2} f_2$$

As reported in [5], Pareto composition with SV-semantics preserves s.p.o.'s. Thus, the result of applying this composition operator starting from the base preference constructors defined in this section is still a preference according to Def. 4. Note that Pareto composition is commutative and associative.

Example 7. The preference query introduced in Example 1 can be formulated as BETWEEN(AvgIncome,MININC,1000) $\otimes$ CONTAIN(RES,State) $\otimes$ POS(MajorGroup,'Professional').

4   Conclusions and Related Works

In this paper we have argued that preferences are a valuable technique for many OLAP applications. On the other hand, ad-hoc base preference constructors are needed to handle the required expressiveness. Thus, we have defined an algebra

that allows preferences to be formulated, besides attributes and measures, also on hierarchies, i.e., on the aggregation level of facts.

The literature on preference queries is huge, but only few works may be related to queries involving preferences on schema and aggregated data. An attempt to situate preferences in the context of multidimensional databases is [6], whose focus is to enable efficient computation of Boolean predicates and preference expressions on numerical domains. Preferences on categorical domains are not supported, and there is no mention to the possibility of expressing preferences on aggregation levels. Finally, in [7] preferences are expressed on a hierarchy of concepts, but information is always retrieved at the finest level of detail and preferences cannot be expressed on schema.

To close this section, we briefly discuss the effectiveness of our approach. We start by observing that OLAP preferences play a major role in reducing the effort of decision-makers to find the most interesting information. This effort can be quantitatively estimated by counting the number of OLAP queries necessary to "manually" retrieve the facts that best match the user preferences. Running an OLAP session entails formulating a sequence of queries, each specifying a group-by set, a list of required measures and an optional set of predicates on attributes and measures. To minimize her effort, a decision-maker should run an OLAP session by first formulating queries that may return best-matching facts. For instance, consider the preference in Example 7. To manually obtain the same results, the decision-maker should first formulate all possible queries including State (i.e., 16 queries) in the group-by set and select the facts related to professionals and yielding an average income lower than \$ 1000. However, if no facts exactly matching the preference are found, more queries will be required; in the worst case, 256 queries must be formulated, which means retrieving the whole datacube. In presence of complex preferences on measures (such as HIGH-EST(AvgIncome) $\otimes$ LOWEST(AvgMortgage)), the decision-maker would also have to analyze the results to check for numerical domination since this cannot be expressed by an OLAP query.

References

1. Chomicki, J.: Preference formulas in relational queries. ACM Trans. on Database Systems 28(4), 427–466 (2003)
2. Kießling, W.: Foundations of preferences in database systems. In: Proc. VLDB, Hong Kong, China, pp. 311–322 (2002)
3. Han, J.: Towards on-line analytical mining in large databases. ACM SIGMOD Record 27, 97–107 (1998)
4. Minnesota Population Center: Integrated public use microdata series (2008), http://www.ipums.org
5. Kiessling, W.: Preference queries with SV-semantics. In: Proc. COMAD, Goa, India, pp. 15–26 (2005)
6. Xin, D., Han, J.: P-cube: Answering preference queries in multi-dimensional space. In: Proc. ICDE, Cancún, México, pp. 1092–1100 (2008)
7. Koutrika, G., Ioannidis, Y.: Answering queries based on preference hierarchies. In: Proc. VLDB, Auckland, New Zealand (2008)