韓國 컴퓨터情報學會 論文誌
第14卷 第1號, 2009. 1.

2009-14-1-5-2

OLAP 큐브에서의 집계함수 AVG의 적용

이 승 현*, 이 덕 성*, 최 인 수**

Applying an Aggregate Function AVG to OLAP Cubes

Seung-Hyun Lee*, Duck-Sung Lee*, In-Soo Choi**

요 약

데이터에 내재되어있는 특이 패턴을 찾고자 데이터 분석을 할 때에 보통 다차원적인 데이터 집계를 하는데, 이때에 표준 SQL 쿼리를 사용해도 좋지만 쿼리가 아주 복잡해진다는 단점이 생기게 된다. 쿼리가 복잡해지면 표준 테이블을 여러 번 참조해야 되고 결과적으로 쿼리의 성능이 저하된다는 뜻이다. OLAP 쿼리는 복잡한 것이 대다수이기 때문에 SQL 쿼리를 대신할 새로운 집계용 연산자인 데이터 큐브를 간단히 불러 큐브를 만들 필요가 생기는 것이다. 집계를 하고, 부분 합을 구하는 것과 같은 OLAP 업무를 지원해 주는 것이 데이터 큐브이다. 이러한 데이터 큐브를 작성하는데 관련된 집계함수에는 여러 가지가 있는데, 이를 분배적 함수, 대수적 함수 그리고 전체관적 함수의 3가지로 분류할 수 있다. 이 중, SUM, COUNT, MAX, MIN과 같은 분배적 함수는 데이터 큐브를 작성하는 데에 직접 사용할 수 있고, AVG와 같은 대수적 함수는 매개함수를 활용하면 사용가능하다고 알려져 있다. 즉, AVG 자체는 분배적 함수가 아니지만, (SUM, COUNT)와 같은 매개함수로 분배적 함수가 되기 때문에 매개함수를 이용하여 구하면 된다는 뜻이다. 그러나 본 연구에서는 (SUM, COUNT)와 같은 매개함수를 통해 AVG를 구하는 것이 OLAP 큐브 작성에 적용시킬 수 없다는 사실을 확인했으며, 결과적으로 이 매개함수를 활용하면 잘못된 결론에 다다르고 그릇된 의사결정을 하게 된다는 사실을 확인하게 되었다. 따라서 본 연구에서는 집계함수 AVG를 OLAP 큐브에 적용시켰을 때의 여러 문제점을 밝혀내고 또한 이들 문제점을 해결할 방안을 찾고자 하는 데에 목적을 두고 있다.

Abstract

Data analysis applications typically aggregate data across many dimensions looking for unusual patterns in data. Even though such applications are usually possible with standard structured query language (SQL) queries, the queries may become very complex. A complex query may result in many scans of the base table, leading to poor performance. Because online analytical processing (OLAP) queries are usually complex, it is desired to define a new operator for aggregation, called the data cube or simply cube.

Data cube supports OLAP tasks like aggregation and sub-totals. Many aggregate functions can be used to construct a data cube. Those functions can be classified into three categories, the distributive, the algebraic, and the holistic. It has been thought that the distributive functions such as SUM, COUNT, MAX, and MIN can be used to construct a data cube, and also the algebraic function such as AVG can be used if the function is replaced to an intermediate function. It is believed that even though AVG is not distributive, but the intermediate function (SUM, COUNT) is distributive, and AVG can certainly be computed from (SUM, COUNT).

• 제1저자 : 이승현
• 투고일 : 2008. 12. 22, 심사일 : 2008. 12. 29, 게재확정일 : 2009. 1. 20.
* 숭실대학교 대학원 산업·정보시스템공학과 재학 ** 숭실대학교 산업·정보시스템공학과 교수
※ 본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌음.

In this paper, however, it is found that the intermediate function (SUM, COUNT) cannot be applied to OLAP cubes, and consequently the function leads to erroneous conclusions and decisions. The objective of this study is to identify some problems in applying aggregate function AVG to OLAP cubes, and to design a process for solving these problems.

▸ Keyword : Aggrigate Function, AVG, Average, OLAP Cubes, Algebraic, Distributive, Base table

Ⅰ. 서 론

기업은 서로 경쟁적 우위를 차지하기 위하여 BI(Business Intelligence) 시스템을 도입한다. BI 시스템이란 비즈니스 의사결정을 진작 시키는데 사용할 수 있는 정보를 산출하는 정보시스템을 말한다. BI 시스템의 유형에는 보고 시스템(Reporting Systems), 데이터마이닝 시스템(Data-Mining Systems), 지식경영 시스템(Knowledge-Management Systems), 전문가 시스템(Expert Systems)이 있는데, 이러한 BI 시스템을 활용함으로써 기업은 상대기업보다 경쟁 우위의 자리를 선점할 수 있게 되는 것이다. 이 중에서 보고 시스템은 다양한 원천 데이터로부터 자료를 분류, 구분, 합계, 평균, 비교처리를 함으로써 자료를 통합하는 시스템으로, 이 결과 정보를 적시적소에 필요한 사용자에게 제공함으로써 의사결정 능력을 향상시키는 시스템을 말한다. 이러한 보고 시스템에는 대표적으로 OLAP(On-Line Analytical Processing)이 사용되고 있다. 집계함수 기능을 활용하는 OLAP의 주목할 만한 특징은 동적인 구조를 갖는다는 것이다. 다시 말해, 최종사용자가 다차원 구조를 자유자재로 변경함으로써 정보를 분석하고 의사결정에 활용할 수 있는 과정이 OLAP이라고 정의할 수 있다는 뜻이다. 여기서 자유자재로 다차원 구조를 변경 시킬 수 있는 최종사용자의 능력을 나타내는 말이 OLAP에서의 『On-Line』 인 것이다[4].

OLAP 큐브는 사실 테이블(Fact Table)과 차원 테이블(Dimension Table)로 구성된다. 사실 테이블에는 측정값을 기입하는데, 여기서 측정값이라는 것은 관심 있는 데이터 항목을 말하며 OLAP에서 합계, 평균 및 기타 다른 처리가 되어지는 값을 말한다. 즉 총 판매량, 평균 판매량, 평균 비용 등이 이러한 측정값의 좋은 예이다. 차원 테이블에는 구매 날짜, 고객 유형, 고객 주소, 판매 지역 등 측정값의 특성을 나타내는 항목들이 기입된다[3].

이렇듯 OLAP에서는 최종사용자가 다차원 구조를 변경함으로써 정보를 분석하게 되고, 의사결정에 활용하는 과정에 있어서 집계함수를 사용하여 측정값들을 각 차원별로 표현하게 되는 것이다. 이처럼 OLAP에서는 데이터 분석을 하는 데에 집계함수와 같은 통계적 기능을 활용하기 때문에 만약 그 기능이 잘못 적용되게 된다면 심각한 결과와 결정을 초래한다[8].

표준 SQL(Structured Query Language)에서 테이블의 값을 집계하는 데에는 기본적으로 COUNT, SUM, MIN, MAX, AVG의 5가지 집계함수(Aggregation Function)를 사용하고 있다. 집계함수에는 분배적 함수(Distributive Function), 대수적 함수(Algebraic Function), 전체관적 함수(Holistic Function)의 3종류가 있는데, COUNT, SUM, MIN, MAX는 분배적 함수이고, AVG는 대수적 함수이다. 또한 중앙값 함수인 MEDIAN은 전체관적 함수이다. SQL에서 이와 같이 3가지로 구분하여 다루는 집계함수가 OLAP에서는 분배적 함수와 비분배적 함수(Non-Distributive Function)의 2가지로 구분되어 다루어지고 있다[2].

자세히 설명하면, COUNT, SUM, MIN, MAX를 분배적 함수로 분류하고, MEDIAN은 비분배적 함수로 분류하고 있다. 분배적 함수가 아닌 AVG는 (SUM, COUNT)형식의 매개함수(Intermediate Function)로 해석하여 분배적 함수에 포함시키고 있다.

현재 사용되고 있는 OLAP의 경우 AVG를 분배적 함수 영역에 포함시켜 사용하고 있어 중요한 문제점을 발생시키고 있다. 따라서 본 논문에서는 이러한 문제점을 해결하기 위한 방안으로 OLAP 큐브에서 집계함수 AVG를 올 바르게 적용시키는 방안을 제시하고자 한다.

Ⅱ. 관련연구

본 장에서는 집계함수, 큐보이드, 데이터 타입 및 결측값에 대한 기존연구를 살펴보고, 이에 따른 문제점을 알아보고자 한다.

1. 집계함수

표준 SQL에서는 간단한 통계적 기능을 활용하여 자료를 분석한다. 이때 활용되는 여러 함수들을 집계함수라 말한다. 이러한 집계함수는 분석능력을 증가시키기 위하여 제공되는 함수로, 대표적인 것으로는 SUM, AVG, COUNT, MAX, MIN 등이 있다. 이러한 집계함수들을 다시 그 기능적인 측면에서 분류를 하면 분배적 함수, 대수적 함수, 전체관적 함수로 나눌 수 있다. 그림 1에서와 같은 두 개의 차원으로 구성

된 데이터 셀 $\{X_{ij}|i=1,...,I;\ j=1,...,J\}$을 대상으로 자세하게 살펴보면 다음과 같다.

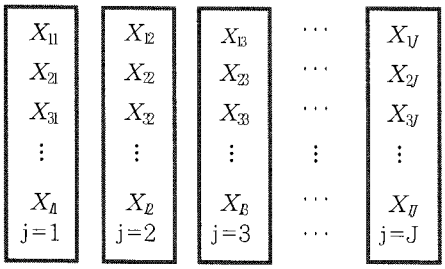

그림 1. $\{X_{ij}|i=1,...,I;\ j=1,...,J\}$ 데이터 셀
Fig. 1. A two dimensional set of values $\{X_{ij}|i=1,...,I;\ j=1,...,J\}$

첫 번째, 만약에 함수 G()가

$F(\{X_{i,j}\})=G(\{F(\{X_{ij}|i=1,...,I\})|j=1,...,J\})$를 만족시킨다면, 집계함수 F()는 분배적이 된다. COUNT, MIN, MAX, SUM 집계함수는 모두 분배적이 되는데 COUNT를 제외한 나머지 집계함수에 있어서는 F()=G()가 된다. SUM을 예로 설명하면,

$F(\{X_{ij}|i=1,...,I\})$는 그림 1에서 j=1인 경우 $\sum_{i=1}^{I} X_{i1}$가 되며, j=2인 경우 $\sum_{i=1}^{I} X_{i2}$, …, j=J인 경우 $\sum_{i=1}^{I} X_{iJ}$가 되고,

$$G(\{F(\{X_{ij}|i=1,...,I\})|j=1,...,J\})$$
$$= \sum_{i=1}^{I} X_{i1} + \sum_{i=1}^{I} X_{i2} + \cdots + \sum_{i=1}^{I} X_{iJ} = F(\{X_{i,j}\})$$

가 되어서 F()=G()가 되는 것이다. COUNT의 경우는 F()=COUNT가 되고, G()=SUM이 된다.

두 번째, 만약에

$F(\{X_{ij}\})=H(\{G(\{X_{ij}|i=1,...,I\})|j=1,...,J\})$를 만족시키는 M-터플(M-tuple) 함수 G()와 함수 H()가 존재하면 집계함수 F()는 대수적이 된다. 평균을 구하는 AVG가 대표적인 대수적 함수인데 이를 설명하면 다음과 같다. 함수 G()는 그림 1에서의 하부 셀 J개의 합(SUM)과 개수(COUNT)를 기록하며, 함수 H()는 이들 J개의 합을 다 더하여 개수 합의 더한 것으로 나누어 줌으로써 AVG를 구하게 된다.

마지막으로 중앙값을 구하는 MEDIAN, 순위를 구하는 RANK 같은 것이 전체관적 함수이다[7].

이상, SQL에서의 집계함수 분류에 대해서 알아보았으나 OLAP 큐브에서는 이와 같지 않다. OLAP에서는 AVG를 분배적함수로 취급하고 있다[2]. 본 연구에서는 이 같은 사용방법에 오류가 있음을 증명하고, 바르게 사용할 수 있는 방안을 제시하고자 한다.

2. 큐보이드(Cuboid)

OLAP은 다차원적인 구조를 갖는다. 이러한 다차원적인 구조를 사용자가 슬라이스(Slice), 다이스(Dice), 드릴다운(Drill-down), 롤업(Roll-up) 등의 스킬을 이용하여 변화시킴으로써 분석을 하게 되는 것이다. 이렇게 하여 만들어지는 각각의 보고서(Report Viewer)를 큐보이드라 한다. 즉, 큐보이드란 데이터 큐브를 격자구조로 표현할 때에 있어서의 격자(직평행 육면체)를 말한다. 따라서 하나의 다차원적 데이터 큐브에 존재하는 큐보이드의 수는 해당 큐브가 관계하고 있는 차원의 수와 각 차원의 계층(수준) 수가 밀접한 관계를 갖게 되는 것이다. 하나의 큐브에서 나타내어지는 큐보이드의 수를 알아보기 위하여 각각 2계층을 갖는 3개의 차원(EMPLOYEE, TIME, STORE)으로 구성된 데이터 큐브를 도식화하여야 한다. 도식화된 데이터 큐브는 그림 2와 같다.

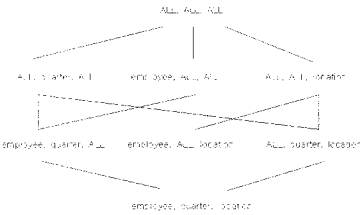

그림 2. 데이터 큐브의 격자구조
Fig. 2. The lattice structure of data cube

그림 2에서 나타낸 것과 같이 하나의 큐브에서 생성되는 큐보이드의 수는 각 차원이 표현할 수 있는 계층 수의 곱한 것과 같다.

$$_2C_1 \times _2C_1 \times _2C_1 = 8$$

이와 같이, 이 데이터 큐브는 8개의 큐보이드로 구성되는 것이다. 본 연구에서는 베이스 테이블(Base Table)에 관계없이 데이터 큐브의 격자구조를 논하는 기존[5][9]연구에 문제점이 있음을 발견하고 이를 해결하고자 하는 데에 목적을 두고 있다.

3. 데이터 타입(Data-type)

그림 3에서는 학과별, 연도별로 어떤 대학의 대학원에 재적하고 있는 학생 수를 나타내고 있다. 여기서 학과별 3년간 재적인원수의 합을 나타내는 제일 오른쪽 Totals 열은 보통 합계를 하는 그런 열은 아니다. 예로 Computer Science 학과의 3년간 재적생 수는 28명으로 되어 있고, 45명(15+17+13)으로는 되어 있지 않음을 알 수 있다. 이것은 이 대학원의 과정이 2년 석사과정이고, 또한 2년 만에 모든 학생이 이상없이 다 졸업한다는 대전제, 즉 SR(Semantic Rule ; 내부적인 대전제 혹은 기본 규칙)이 있음을 안다면 이런 계산을 할 수 있게 된다.

학과 \ 연도	1994	1995	1996	Totals
Computer Science	15	17	13	28
Statistics	10	15	11	21
Totals	25	32	24	49

그림 3. 학과별, 연도별 재적인원 현황
Fig. 3. Number of students by department and year

반면에 그림 3에서 대학원의 재적생 수를 구하는 제일 아래쪽 Totals 행은 보통 합계를 할 수 있는 그런 행이 된다. 예로, 1994년도에 대학원에 재적하고 있는 학생수는 25명(15+10)으로 계산되어 진다.

이상에서 알수 있듯이 합을 구하는데 있어서 SR을 알아야만 가능한 경우도 있고, SR을 몰라도 가능한 경우가 있다. 이 중 SR을 알아야만 올바른 합을 구할 수 있는 데이터를 저장형(Stock-type) 데이터라 하고, SR을 몰라도 올바르게 합을 구할 수 있는 데이터를 흐름형(Flow-type) 데이터라 한다. 전술한 예 중에서, 3년간 Computer Science 학과에 재적한 학생수를 구하는 경우에 사용되는 데이터는 저장형 데이터이고, 1994년도에 있어서의 대학원 재적생 수를 구하는데 사용되는 데이터는 흐름형 데이터인 것이다.

Lenz와 Shoshani는 저장형 데이터에 대해서는 집계함수 중 MIN, MAX, RANGE, AVG 분배함수를 다 적용시킬 수 있으나, SUM은 적용시킬 수 없다고 하였으며, 흐름형 데이터에 대해서는 앞의 5개 함수 모두 다 적용시킬 수 있다고 하였다[6]. 그러나, 본 연구에서 SUM의 경우는 전술한 바와 같이 이들의 주장이 옳으나, AVG의 경우는 흐름형 데이터에서 적용시킬 수 없음을 확인하고, 이의 해결책을 마련하고자 하고 있다.

4. 결측값

결측값(Missing values) 혹은 널(null)값이라고 부르는 값들은 이제까지 퍼뜨려진바 없는 값들을 말하는 것으로 공백값(Blank Values)은 아니다. 공백값에서는 아무 것도 없는 상태가 되는데 비해 결측값에서는 알려지지는 않았지만 뭔가 존재하게 되는 상태가 된다.

결측값이 생기는 여건은 다음의 3가지로 종합할 수 있다. 첫째, 값 자체가 부적합할 때 결측값이 되는 경우이다. 예로 마지막 출산일을 기록하는 경우 남성에게는 일어날 수 없는 부적합한 경우가 되기 때문에 도저히 기록할 수 없게 되는 것이다. 둘째, 값 자체는 적합하지만 알지 못할 때 결측값이 되는 경우이다. 마지막 출산일을 기록하는 앞의 예는 여성에게는 적합하게 되는데, 단 이 해당여성의 마지막 출산일을 모르는 경우는 기록할 수 없게 되는 것이다. 셋째로, 값 자체도 적합하고 알려져 있는 상태이지만 사정상 기록하지 못함으로써 결측값이 되는 경우이다. 예로 100개의 상품을 비치해서 판매하는 어느 상점에서 매일 팔리는 상품별로 판매개수와 판매액을 기록하는 경우를 생각해 보자. 어느 날 20개의 상품만 팔렸다면 이 20개의 상품에 대해서만 그 판매개수와 판매액을 기록하지 나머지 80개의 상품에 대해서는 각각 그 판매개수가 0이고 판매액이 0이라고 기록하지는 않는다. OLAP에서의 사실 테이블의 기입을 생각해보면 이를 쉽게 이해하리라 본다. 즉, 80개의 상품에 대해서 각각 그 판매개수가 0이고 판매액이 0이라는 사실을 알고 있지만 기입하지 않기 때문에 이들 모두가 결측값이 되는 것이다[3].

본 연구에서는 현실적으로 0을 기입할 수 없는 세 번째 종류의 결측값을 처리할 수 있는 방안을 찾고자 한다.

III. 현 OLAP 큐브에서의 집계함수의 적용

지금까지 OLAP 큐브에서의 집계함수 적용에 대한 기존 연구들을 살펴보았는데, 여기서는 기존 연구에서 발생된다고 하는 문제점을 자세히 검토하기 위하여 다음과 같은 하나의 가상 사례 데이터베이스를 구성하고자 한다.

1. 사례 OLAP 큐브

스타 스키마(Star Schema)를 채택하여 OLAP 큐브를 설계하는 것이 가장 보편화된 방법이다. 스타 스키마는 중앙의 사실 테이블(Fact Table)을 여러 개의 차원 테이블

(Dimension Table)이 둘러싼 형태를 취한다. 사실 테이블에서의 각 행은 여러 차원의 기본키들의 조합 즉, 외래키들의 조합으로 구성된 하나의 복합키와 이것과 관련된 여러 측정값으로 구성된다. 이상의 차원 테이블과 사실 테이블의 조합으로 구성되는 것을 큐브라 부른다. 본 연구에서는 1년간 각 상점별, 제품별 판매량에 대한 가상 데이터를 OLAP 큐브로 구성한다. OLAP 큐브를 구성하기 위한 세부 내용은 다음과 같다.

Time_ID	Week	Month	Quarter	Year
1	1	1	1	2008
2	2	1	1	2008
3	3	1	1	2008
4	4	1	1	2008
5	1	2	1	2008
⋮	⋮	⋮	⋮	⋮
48	4	12	4	2008

그림 4. TIME 차원 테이블
Fig. 4. Dimension Table. TIME

차원 테이블로는 시간(TIME) 테이블, 상점(STORE) 테이블, 제품(PRODUCT) 테이블의 3개가 있다. TIME 차원 테이블은 연(Year), 분기(Quarter), 월(Month), 주(Week)의 4개 수준으로 구성된다. 각 수준의 구성원은 주 48개, Month 12개, Quarter 4개, Year 1개의 구성원이 존재하며, 1개월은 4주로 구성된다. 또한 기본키를 구성하는 Time_ID 48개를 부여하여 각 행을 식별할 수 있도록 하였다. 이렇게 구성된 TIME 차원 테이블은 그림 4와 같다.

STORE 차원 테이블은 상점이 위치하는 지역을 대상으로 나라(Country), 도(Locate), 시(City)의 3개 수준으로 하였다. 나라 수준에는 Korea, 1개의 구성원이 있으며, 도 수준에는 Kyunggi, Chungcheong, Kyeongsang, Jeolla라는 4개의 구성원이 있다. 또한 도의 하위 수준으로는 시가 존재하며, Kyunggi의 하위 구성원으로 Ansan, Sihwa가 있으며, Chungcheong의 하위 구성원으로는 Jaechon, Gongju가, Kyeongsang의 하위 구성원으로는 Changwon이 있고, Jeolla의 하위 구성원으로는 Yusu, Iksan이 있어 도시의 총 구성원은 7개가 있으며, 기본키로 사용할 Store_ID 7개를 부여하여 각 행을 식별하도록 하였다. 이렇게 구성된 각 수준을 테이블로 표기하면 그림 5와 같다.

Store_ID	City	Locate	Country
1	Ansan	Kyunggi	Korea
2	Sihwa	Kyunggi	Korea
3	Jaechon	Chungcheong	Korea
4	Gongju	Chungcheong	Korea
5	Changwon	Kyeongsang	Korea
6	Yusu	Jeolla	Korea
7	Iksan	Jeolla	Korea

그림 5. STORE 차원 테이블
Fig. 5. Dimension Table. STORE

PRODUCT 차원 테이블은 각 상점에서 판매되는 제품을 나타내며, 3개의 수준으로 Class 2개, Category 5개, Name 20개의 구성원이 존재한다. 최상위 수준에는 제품의 종류인 Class 수준으로 Drink와 Food가 있다. 이러한 Class의 하위 수준에는 각 제품의 종류별 세부 부류인 Category가 존재하며, Class 구성원의 각각 하위 구성원으로 Drink에 Soda와 Juice가 있으며, Food는 FastFood, Noodle, PackingFood가 존재하여 Category의 구성원은 총 5개로 구성된다. 또한 최하위 수준으로 각 제품 명칭인 Name 수준이 존재하고, Category 수준의 각 구성원에 대하여 Soda에는 Coke, Diet Coke, Fanta, Cider가 있으며, Juice에는 Apple juice, Grape juice, Orange juice, Peach juice가 있고, FastFood에는 Burger, Corn Cheese, Fried Potato, Pizza, Salad가 있으며, Noodle에는 Bibim Myun, Cup Myun, Saeng Myun, Spaghetti, Tang Myun이 있고, PackingFood에는 Peach, Tuna가 존재하여 Name에 속하는 구성원은 총 20개가 된다. 여기에 Product_ID 20개를 부여하여 기본키로 사용한다. 이렇게 PODUCT 차원 테이블을 구성하여 표현하면 그림 6과 같다.

Product_ID	Name	Category	Class
1	Coke	Soda	Drink
2	Diet Coke	Soda	Drink
3	Cider	Soda	Drink
4	Fanta	Soda	Drink
5	Orange Juice	Juice	Drink
⋮	⋮	⋮	⋮

그림 6. PRODUCT 차원 테이블
Fig. 6. Dimension Table. PRODUCT

사실 테이블은 각 차원의 기본키로 구성된 복합키 부분과 측정값으로 나타나는데, 판매량을 나타내는 측정값으로는 Quantity가 있다. 이러한 사실 테이블 중 1월달 일부를 나타낸 것이 그림 7이다.

Time_ID	Store_ID	Product_ID	Quantity
1	2	2	123
1	3	3	32
1	4	9	209
1	4	11	197
1	5	11	219
1	6	14	18
1	6	17	210
1	6	18	163
2	2	5	193
2	2	8	255
2	3	9	107
2	3	13	222
2	5	17	162
2	7	19	85
3	1	3	209
3	1	6	131
3	2	9	126
3	3	11	3
3	4	18	236
3	7	19	283
4	2	6	88
4	3	8	228
4	3	9	179
4	3	11	297
4	6	14	51
4	6	15	132
4	7	19	242
5	1	1	157
⋮	⋮	⋮	⋮

그림 7. ORDER 사실 테이블
Fig. 7. Fact Table. ORDER

이렇게 구성한 데이터베이스를 기반으로 스타 스키마를 구성하면 그림 8과 같다.

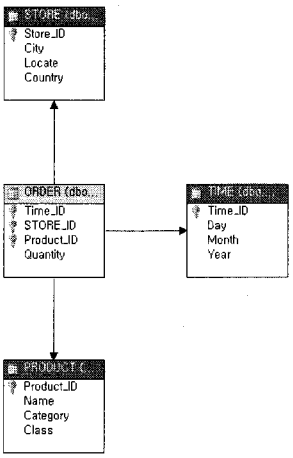

그림 8. ORDER 큐브의 스타스키마
Fig. 8. Star schema of ORDER cube

이렇게 스타스키마를 구성하면 바로 큐브를 생성할 수 있는데, 2008년 1년 동안 1개 상점 당 각 제품에 대한 주별 평균판매량을 보여주는 큐브를 생성하면 그림 9와 같이 된다. 이것은 3개의 차원을 2차원에 표기하기 위하여 좌측면에 PRODUCT.name 수준에서 STORE.Country를 배치시키고 상단에 TIME.Week를 배치시킨 것이다.

OLAP 데이터 큐브에서는 기본적으로 SUM과 COUNT가 계산된다. 그림 9에서 1월달 4개 주와 Juice 부분만 발췌한 것이 그림 10인데, 이를 통해 SUM과 COUNT를 설명해 보면 다음과 같다. 열은 크게 5개(1, 2, 3, 4주 열과 한 개의 합계 열)로 되어 있고, 각 열 별로 다시 판매량 현황을 타나내는 3개(Quantity, ORDER 카운트, 평균)씩의 열이 존재하고 있다. 여기서 Quantity 열에는 판매량이, ORDER 카운트 열에는 판매 횟수가, 그리고 평균 열에는 1개 상점 당제품별로 본 주별 평균판매량이 기록된다. 그림 10에서 행을 살펴볼 것 같으면, Juice의 입 구성원별로 2개씩 행이 마련되어 있다. 여기서 첫 번째 행에는 첫 번째 잎 구성원인 Apple juice의 판매량 현황이 기록되고, 두 번째 행에는 첫 번째 잎 구성원인 Apple juice의 현황합계가 기록된다. Juice의 잎 구성원이 4개(Apple juice, Grape juice, Orange juice, Peach juice)이기 때문에 모두 8개의 행이 생기게 되고, 마지막 9번째 행에는 이들 4개 잎 구성원의 상위 구성원인 Juice의 판매량 현황 합계가 기록된다. 그림 10에서 볼것 같으면, Apple juice의 1월달 제 4주째 주별 판매량은 88개, 판매 횟수는 1회, 평균은 88이 되고, Juice의 판매량 합계는 895개, 판매 횟수는 5회, 평균은 179개가 된다.

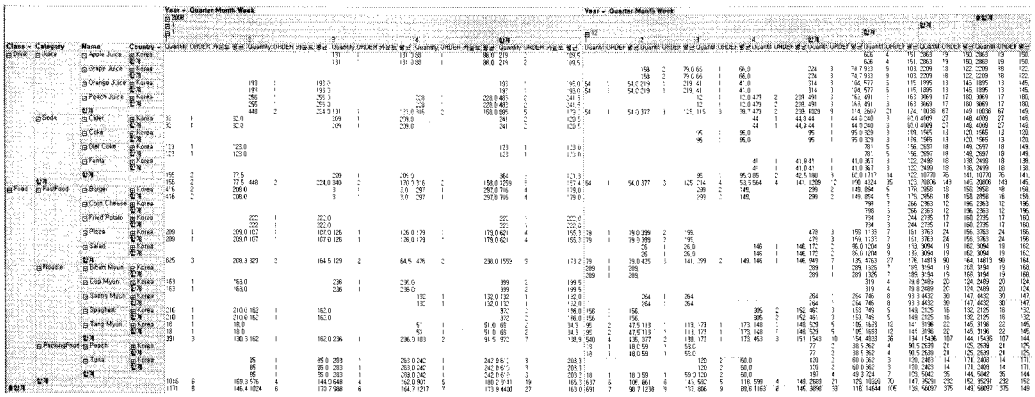

그림 9. ORDER 큐브
Fig. 9. ORDER cube

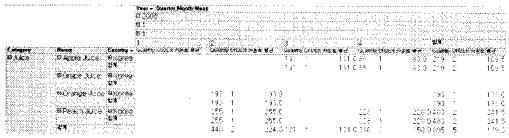

그림 10. 그림 9의 일부분
Fig. 10. Part of Figure 9

2. 집계함수 AVG 적용의 한계

이상 OLAP에서 기본적으로 제공되는 SUM과 COUNT 집계함수의 기능을 활용하여 1개 상점 당 각 제품에 대한 주별 평균 판매량을 그림 9에서와 같이 구해보았는데, 여기에는 다음과 같은 여러 가지 문제점이 있음을 인식하게 되었다.

(1) 제품이 판매되지 않았다면 판매량은 0이 되어야 되는데 0이 기록되지 않고 빈칸으로 되어 있다.
(2) 판매횟수에는 판매량이 0이 될 때마다 1회씩 기록되어야 하는데 그렇지 못하다.
(3) 정확하게 평균 판매량을 계산할 수 없게 되어 있다.

그림 10을 활용하여 이상의 문제점들을 설명하고자 한다. 첫째 제품이 판매되지 않았다면 해당 제품의 판매량을 0으로 기입해야 하는데 II의 4에서 언급한 바와 같이 현실적으로 0을 기입할 수 없는 세 번째 종류의 결측값이 되기 때문에 오류가 생기는 것이다. 그림 7의 사실 테이블에서 5개행(진하게 표시)으로만 기초하여 작성한 데이터 큐브가 그림 10인데, 정확한 평균값을 얻기 위해서는 그림 11과 같이 위의 5개 행이 포함된 112개의 행으로 구성된 사실 테이블을 작성해야만 한다.

Time_ID	Store_ID	Product_ID	Quantity
1	1	5	0
1	1	6	0
1	1	7	0
1	1	8	0
1	2	5	0
1	2	6	0
1	2	7	0
1	2	8	0
1	3	5	0
1	3	6	0
1	3	7	0
1	3	8	0
2	4	5	0
2	4	6	0
2	4	7	0
2	4	8	0
2	5	5	0
2	5	6	0
2	5	7	0
2	5	8	0
2	6	5	0
2	6	6	0
2	6	7	0
2	6	8	0
2	7	5	0
2	7	6	0
2	7	7	0
2	7	8	0

Time_ID	Store_ID	Product_ID	Quantity
1	1	5	0
1	1	6	191
1	1	7	0
1	1	8	0
1	2	5	0
1	2	6	0
1	2	7	0
1	2	8	0
1	3	5	0
1	3	6	0
1	3	7	0
1	3	8	0
2	4	5	0
2	4	6	0
2	4	7	0
2	4	8	0
2	5	5	0
2	5	6	193
2	5	7	0
2	5	8	255
2	6	5	0
2	6	6	0
2	6	7	0
2	6	8	0
2	7	5	0
2	7	6	0
2	7	7	0
2	7	8	0

그림 11. 재구성한 사실 테이블
Fig. 11. The modified Fact Table

그림 11의 사실 테이블을 사용하여 큐브를 재구성해 보면 그림 12와 같다. 본 연구에서는 전국 7개 상점(그림 5참조)을 대상으로 했기 때문에 각 PRODUCT 별 ORDER 카운트가 전부 7로 정확하게 표기되어 있음을 그림 12에서 확인할 수 있으리라 본다. 이는 해당 제품이 팔리지 않았을 경우, 즉 0개 팔렸을 경우도 정확히 ORDER 카운트에 반영되었기 때

문이다. 그림 10에서 Juice의 판매량 합계는 895개, 판매 횟수는 5회, 평균은 197개로 되어있는데 비해 그림 12에서 각각 895개, 112회, 8개로 되어 있다. 이 같은 차이가 그림 10에서는 0을 기입할 수 없는 세 번째 종류의 결측값을 채택하였고, 결과적으로 대수적 집계함수를 분배적 집계함수의 형태로 전향시키지 못했으며, 그림 12에서는 세 번째 종류의 결측값이 생기지 않게 하여 결과적으로 대수적 집계함수를 분배적 집계함수의 형태로 전환시킬 수 있도록 하였기 때문이라고 본다. 다시 말하면 그림 9에서의 결과는 다음과 같은 평균의 정확한 사전적 의미를 충족시키지 못한 채 구한 잘못된 결과이고, 그림 12에서의 결과는 이를 충족시켜서 구한 정확한 결과라는 뜻이다.

> 평균(average)이란, 둘 이상의 수를 더할 때, 더한 합계를 더하는데 사용한 수의 개수로 나눈 결과를 말한다.

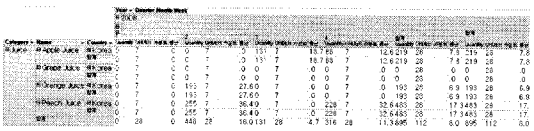

그림 12. 재구성한 큐브
Fig. 12. The modified Cube

Ⅳ. OLAP 큐브에서의 대수적 함수 AVG의 적용 프로세스

지금까지 OLAP 큐브를 사용함에 있어서 AVG를 적용할 때 발생되는 문제점을 확인 하였다. 지금부터는 이러한 문제점을 해결하기 위한 OLAP 큐브에서의 대수적 함수 AVG의 적용 프로세스를 살펴 보겠다.

1. 현방식의 사실 테이블 활용

Ⅲ에서 본 바와 같이 세 번째 종류의 결측값이 생기지 않도록 사실테이블을 작성하는 것은 현실적으로 불가능하고, 더구나 경제적이지 못하다. 따라서 본 연구에서는 기존 방식의 사실테이블 입력방법을 따르기로 한다. 우선 현행 방식대로 데이터 큐브를 작성한다.

2. 평균계산의 대상이 되는 차원의 잎구성원 수 책정

Ⅲ의 사례에서는 1개 상점 당 각 제품에 대한 주별 평균 판매량을 구하고자 하기 때문에 STORE 차원의 잎구성원 수가 7이 된다. Ⅳ의 1에서 구한 큐브에는 각 제품별로 우리나라 7개 상점에서 주별로 판매한 금액이 기록되어 있기 때문에, 1개 상점에서 평균적으로 얼마나 판매했는가를 알아보자면 7로 나누어야 한다는 뜻이다.

3. 새로운 베이스 테이블의 작성

Ⅳ의 1에서 구한 Quantity를 Ⅳ의 2에서 구한 7로 나누어 준 것을 측정값으로 한 새로운 베이스 테이블을 작성한다. Ⅲ의 사례에서 Quantity를 7로 나눈 것을 살펴보면 그림 13과 같다.

		Year = 2008									
		Quarter = 1									
		Month = 1									
		1		2		3		4		합계	
Class = Category	Name	Quantity	New AVG()	Quantity	New AVG()	Quantity	New AVG()	Quantity	New AVG()	Quantity	New AVG()
⊟Drink ⊟Juice	Apple Juice					131	18.7	88	12.6	219	31.3
	Grape Juice										
	Orange Juice			193	27.6					193	27.6
	Peach Juice			255	36.4			228	32.6	483	69.0
	합계			448	64.0	131	18.7	316	45.1	895	127.9
⊟Soda	Cider	32	4.6			209	29.9			241	34.4
	Coke										
	Diet Coke	123	17.6							123	17.6
	Fanta										
	합계	155	22.1			209	29.9			364	52.0
합계		155	22.1	448	64.0	340	48.6	316	45.1	1259	179.9
⊟Food ⊟FastFood	Burger	416	59.4			3	.4	297	42.4	716	102.3
	Corn Cheese										
	Fried Potato			222	31.7					222	31.7
	Pizza	209	29.9	107	15.3	126	18.0	179	25.6	621	88.7
	Salad										
	합계	625	89.3	329	47.0	129	18.4	476	68.0	1559	222.7
⊟Noodle	Bibim Myun										
	Cup Myun	163	23.3			236	33.7			399	57.0
	Saeng Myun							132	18.9	132	18.9
	Spaghetti	210	30.0	162	23.1					372	53.1
	Tang Myun	18	2.6					51	7.3	69	9.9
	합계	391	55.9	162	23.1	236	33.7	183	26.1	972	138.9
⊟PackingFood	Peach										
	Tuna			86	12.1	283	40.4	242	34.6	610	87.1
	합계			86	12.1	283	40.4	242	34.6	610	87.1
합계		1016	145.1	576	82.3	648	92.6	901	128.7	3141	448.7
총합계		1171	167.3	1024	146.3	988	141.1	1217	173.9	4400	628.6

그림 13. 새로운 평균의 적용
Fig. 13. A New AVG table

그리고 그림 13을 기반으로 새로운 베이스 테이블, 즉 새로운 사실테이블을 작성해 보면 그림 14와 같게 된다. 그림 14는 새로운 사실테이블 중 1월과 12월에 대한 현황을 보여준다.

Product_ID	Time_ID	New AVG()
1	1	0
1	2	0
1	3	0
1	4	0
1	5	22.4
1	6	4.9
1	7	26.3
1	8	0
1	9	15.4
1	10	71.9
1	11	0
1	12	0
1	13	0
1	14	0
1	15	0
1	16	0
1	17	0
1	18	14
1	19	0
1	20	0
1	21	0
1	22	0
1	23	0
1	24	0
1	25	0
1	26	0
1	27	0
1	28	0
1	29	0
1	30	0
1	31	20.6
1	32	0.6
1	33	0
1	34	0.6
1	35	0
1	36	0
1	37	0
1	38	0
1	39	33.4
1	40	0
1	41	0
1	42	0
1	43	0
1	44	0
1	45	0
1	46	0
1	47	13.6
1	48	0
2	1	17.6
19	48	0
20	1	0
20	2	0
20	3	0
20	4	0
20	5	0
20	6	0
20	7	0
20	8	0
20	9	0
20	10	0
20	11	0
20	12	13.6
20	13	31.3
20	14	1.9
20	15	26
20	16	66
20	17	6.9
20	18	0
20	19	0
20	20	0
20	21	0
20	22	0
20	23	11.1
20	24	30.1
20	25	15.9
20	26	0
20	27	0
20	28	0
20	29	17.1
20	30	18.7
20	31	0
20	32	9.6
20	33	0
20	34	36.9
20	35	20.3
20	36	20
20	37	0
20	38	0
20	39	0
20	40	0
20	41	0
20	42	0
20	43	40.7
20	44	0
20	45	2.6
20	46	8.4
20	47	0
20	48	0

그림 14. 새로운 베이스 테이블
Fig. 14. A new base table

여기서 주의 할 점은 새로운 사실 테이블을 생성할 때 기존 결측값으로 표기되는 곳에는 반드시 0을 기입해 주어야 한다.

4. 새로운 큐브의 작성

Ⅳ의 3에서 구한 새로운 사실 테이블을 기반으로 하여 새로운 큐브를 작성한다.

Ⅴ. 집계함수 AVG의 적용결과

여기서는 Ⅳ에서의 프로세스를 적용하여 집계함수 AVG를 사용하여 OLAP 큐브를 작성하고자 한다.

Ⅳ의 1~3은 이미 기술하였기 때문에 여기서는 Ⅳ의 4부터 기술하기로 한다.

1. 새로운 큐브 작성

새로운 큐브의 스타스키마는 그림 15, 큐브의 격자구조는 그림 16, 집계함수 AVG를 적용시켜서 작성한 새로운 큐브는 그림 17과 같다. 그림 15에서는 STORE 차원이 없다는 점, 그림 16은 그림 14의 베이스 테이블에 대한 격자구조라는 점에 주의할 필요가 있다고 본다.

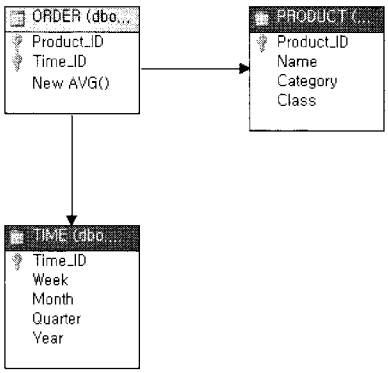

그림 15. 새로운 ORDER 큐브의 스타스키마
Fig. 15. Star schema of new ORDER cube

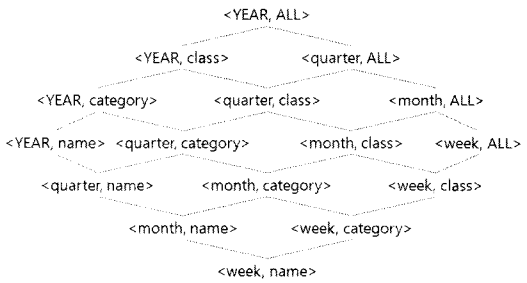

그림 16. 새로운 ORDER 큐브의 격자구조
Fig. 16. The lattice structure of new ORDER cube

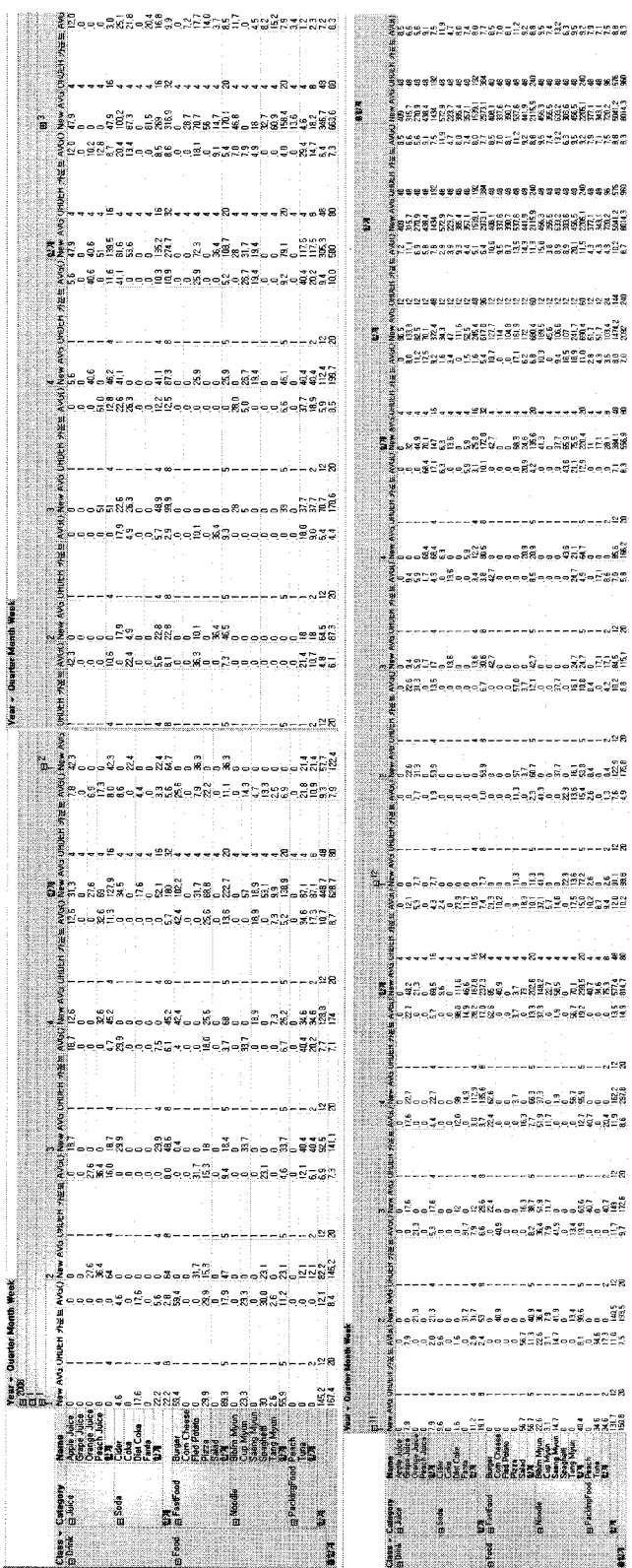

그림 17. 새로운 ORDER 큐브
Fig. 17. New ORDER cube

2. 결과 및 토론

우리나라 전체에 있는 7개 상점을 대상으로 하여 1개 상점당 각 제품에 대한 주별 판매량을 구해 본 결과가 그림 17의 큐브에서와 같은데, 이를 간결하게 정리해 보면 그림 18과 같다.

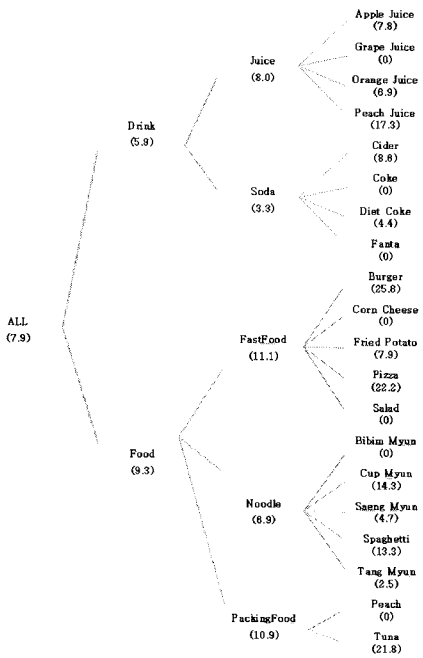

그림 18. 주별 평균 판매량
Fig. 18. Average Weekly Sales

이상은 주별로 알아본 결과이나 만약에 한 달 동안 보았을 때 1개 상점이 평균적으로 몇 개 판매되었는가를 알아보자면 그림 18에서의 값에 4를 곱하면 된다. 일례로 Juice의 월간 평균 판매량은 32(= 4 × 주간 Juice 판매량 8)가 된다는 뜻이다. 왜냐하면 본 연구에서는 4주를 한 달로 보기 때문이다. 분기별 Juice의 평균 판매량을 구할 때에는 그림 18에서의 값에 12를 곱하면 된다.

VI. 결론

측정값에는 단조(monotone)적인 것이 있고, 그렇지 않은 것도 있다. 측정값이 모두 양으로 되어 있거나, 모두 음으로 되어 있다면 단조 측정값이라 부르고, 양음이 혼재되어 있으면 비 단조 측정값이라 부른다.[5] 본 연구에서는 OLAP 큐브의 셀(cell)이 공백으로 되어 있으면 평균을 구하는 집계함수 AVG를 적용시킬 수 없음을 지적하고 있다. 왜냐하면 (SUM, COUNT)의 형식으로 AVG를 구하는데 있어서 세 번째 종류의 결측값이 되는 0 값의 체크를 제대로 할 수 없기 때문이다. 본 연구에서는 AVG 만 적용시킬 수 없음을 지적하고 있으나 사실은 COUNT, MAX, MIN의 3종류 분배적 집계함수도 OLAP 큐브의 셀이 공백으로 되어 있으면 적용시킬 수 없게 된다. COUNT의 경우는 이미 언급하였고, 측정값이 모두 음으로 되어 있는 단조 측정값의 경우는 MAX를, 측정값이 모두 양으로 되어 있는 단조 측정값의 경우는 MIN을 적용시킬 수 없게 되기 때문이다. OLAP 큐브에서는 분배적 집계함수 SUM 만을 큐브의 셀이 공백으로 되어 있건 말건 그리고 측정값이 단조이건 아니건 관계없이 아무런 문제없이 적용시킬 수 있게 된다. 따라서 본 연구에서는 그림 14와 같은 잎 멤버의 평균값 자체를 측정값으로 갖는 베이스 테이블을 갖고 큐브를 구성함으로써 SUM 만을 적용시켜서 AVG를 구할 수 있는 체제를 확립한 것이다. 이 만큼 그림 14와 같은 베이스 테이블의 작성이 아주 중요함을 다시 한 번 강조하고자 한다.

또한 이상의 베이스 테이블을 사실 테이블로 삼아 큐브를 구성하는 방법은 분석의 유용성이 아주 높음을 강조하고 싶다. V의 2에서 밝힌바 같이 시간 차원의 모든 수준별로 원하는 평균을 모두 구할 수 있기 때문이다.

본 연구는 경영자에게 정확한 분석정보를 제공함으로써 경영자들이 올바른 의사결정을 하게 하는 데에 큰 기여를 하리라 생각한다.

참고문헌

[1] Abello, Alberto et. al., "On Relationships Offering New Drill-across Possibliities," ACM, pp. 7-11, 2002.

[2] Gray, Jim et. al., "Data Cube : A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," Data Mining and Knowledge Discovery, pp. 29-53, 1997.

[3] Kroenke, D.M., *"Database Processing 10th Edition,"* Prentice Hall, pp. 126, pp. 547, 2006.

[4] Kroenke, D.M., *"Experiencing MIS,"* Pearson Education, pp. 195-461, 2007.

[5] Lakshmanan, Laks V. S., et. al., "Quotient cube: How to Summarize The Semantics of A Data Cube," Proceedings of The 28th VLDB Conference, pp. 778-789, 2002.
[6] Lenz, Hans-J., et. al., "Summarizability in OLAP and Statistical Data Bases," IEEE, pp. 132-143, 1997.
[7] Rizzi, Stefano et. al., *"What Time Is It in The Data Warehouse,"* Springer Verlag, pp. 134-144, 2006.
[8] Spofford, George et. al., *"MDX Solutions Second Edition,"* Wiley Publishing, pp. 48-51, 2005.
[9] Wang, Lingyu et. al., *"Preserving Privacy in On-Line Analytical Processing,"* pp. 130-131, 2007.
[10] 권오주, *"OLAP Solutions +a,"* 대림출판사, 2002.
[11] 김세지, *"더미멤버를 활용한 비균일 계층의 OLAP 해법,"* 숭실대학교, 2007년
[12] 유한주, *"트랜잭션기반 분석용 OLAP 큐브구조 설계,"* 숭실대학교, 2007년
[13] 유한주, 최인수, "장바구니 분석용 OLAP 큐브 구조의 설계," 한국컴퓨터정보학회논문지, 제12권, 제4호, 180-189쪽, 2007년 9월.
[14] 이덕근 외, "DW에서의 질의어처리 성능향상을 위한 데이터 구조화 방법," 한국컴퓨터정보학회논문지, 제10권, 제1호, 8-13쪽, 2005년 3월.
[15] 조재희, *"OLAP 테크놀로지,"* Sigma Insight, 1999.
[16] The OLAP Report, *http://www.olapreport.com/fasmi.htm*

저 자 소 개

이 승 현

2001년 숭실대학교 산업공학과 졸업 (학사)
2005년 ~ 현재 숭실대학교 대학원 산업·정보시스템공학과 재학
〈관심분야〉 MIS, DW, OLAP, MDX, CRM

이 덕 성

1990년 전남대학교 산업공학과 졸업 (학사)
1995년 전남대학교 산업공학 졸업 (석사)
2005년 ~ 현재 숭실대학교 대학원 산업·정보시스템공학과 재학
〈관심분야〉 MIS, DW, OLAP, MDX, CRM

최 인 수

1985년 서울대학교 산업공학 졸업(공학박사)
1980년 ~ 현재 숭실대학교 산업·정보시스템공학과 교수
〈관심분야〉 MIS, DW, OLAP, MDX, CRM