



(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。 Int. Cl. G06F 17/30 (2006.01)	(45) 공고일자 (11) 등록번호 (24) 등록일자	2007년08월03일 10-0745533 2007년07월27일
---	-------------------------------------	--

(21) 출원번호	10-2005-7005498	(65) 공개번호	10-2005-0059216
(22) 출원일자	2005년03월30일	(43) 공개일자	2005년06월17일
심사청구일자	2005년09월12일		
번역문 제출일자	2005년03월30일		
(86) 국제출원번호	PCT/US2003/033137	(87) 국제공개번호	WO 2004/042617
국제출원일자	2003년10월17일	국제공개일자	2004년05월21일

(30) 우선권주장 10/285,228 2002년10월31일 미국(US)

(73) 특허권자 인터내셔널 비지네스 머신즈 코포레이션
 미국 10504 뉴욕주 아몽크 뉴오차드 로드

(72) 발명자 데팅거 리차드 디
 미국 미네소타주 55901 로체스터 5305 켄싱턴 레인 엔.더블유.

 스티븐스 리차드 제이
 미국 미네소타주 55955 멘토빌 61432 252번 에버뉴

(74) 대리인 김태홍
 신정건

(56) 선행기술조사문헌
 US6,353,826

심사관 : 손영태

전체 청구항 수 : 총 10 항

(54) 전역 쿼리 상관 관계 속성 제공 방법 및 시스템

(57) 요약

데이터를 액세스하기 위한 시스템, 방법, 및 제조 물품이 개시된다. 일반적으로, 데이터 액세스는 상관 관계 표준 또는 표준들(203)로 구성된 쿼리(202)에 의해 용이해 진다. 상관 관계 표준(203)을 제공함으로써, 쿼리(202)의 조건들에 적용될 추가의 상관 관계 논리의 도입이 용이해 진다. 일반적으로, 예를 들어, 기간, 물리적 장소 및 연령에 기초하여 개체들의 상관 관계를 지원하는 상관 관계 표준(203)이 고려된다.

대표도

도 2c

특허청구의 범위

청구항 1.

쿼리 속성들(attributes)을 제공하기 위한 방법으로서,

하나 이상의 논리 필드들 중 특정 논리 필드에 대해 복수의 상관 관계 속성들 중 어느 것이 지원되는지를 지정하는 단계와,

각 특정 필드마다, 상기 특정 논리 필드에 대해 지원되는 상기 상관 관계 속성들의 값을 계산하기 위한 상관 관계 속성 정의(definition)를 지정하는 단계로서 상기 상관 관계 속성은 하나 이상의 논리 필드들에 의해 정의되는 추상적인 쿼리(abstract query)로부터 생성된 쿼리에 쿼리 논리를 추가하도록 구성되는 단계와,

범위(span) 상관 관계 표준이 상기 추상적인 쿼리의 2 개의 조건들의 논리 필드들에 적용되는 것으로 판정하는 단계와,

상기 2 개의 조건의 상기 논리 필드들에 대해 정의되고 상기 범위 상관 관계 표준에 대응하는 상기 상관 관계 속성 정의들에 의해 계산된 각각의 값 간의 차를 계산하는 표현식을 작성하는 단계

를 포함하는 속성 제공 방법.

청구항 2.

제1항에 있어서,

상기 복수의 상관 관계 속성들 중 어느 것이 지원되는지를 지정하는 단계는, 상기 특정 논리 필드에 대해 2 이상의 상관 관계 속성들을 지정하는 단계를 포함하는 것인 속성 제공 방법.

청구항 3.

제1항에 있어서,

상기 값은 시간값인 속성 제공 방법.

청구항 4.

제1항에 있어서,

상기 값은 색상값, 중량값, 체적값, 길이값, 및 너비값 중 하나일 수 있는 것인 속성 제공 방법.

청구항 5.

하나 이상의 논리 필드들에 의해 정의되는 추상적인 쿼리(abstract query)로부터 생성된 쿼리에 쿼리 논리(query logic)를 추가하도록 구성된 속성들(attributes)을 제공하기 위한 방법으로서,

복수의 논리 필드들 각각마다, 특정 논리 필드에 대해 복수의 상관 관계 속성들 중 어느 것이 지원되는지를 지정하는 단계와,

복수의 논리 필드들 각각마다, 상기 특정 논리 필드에 대해 지원된 상기 상관 관계 속성들의 값을 계산하기 위한 상관 관계 속성 정의를 지정하는 단계로서, 2 개의 상이한 논리 필드들에 대해 지정된, 특정 상관 관계 속성값을 계산하기 위한 상기 상관 관계 속성 정의들 중 적어도 2 개는 서로 다른 것인, 상기 정의 지정 단계와,

범위 상관 관계 표준이 상기 추상적인 쿼리의 2 개의 조건들의 논리 필드들에 적용되는 것으로 판정하는 단계와,

상기 논리 필드들에 대해 정의되고 상기 범위 상관 관계 표준에 대응하는 상기 상관 관계 속성 정의들에 의해 계산된 각각의 값 간의 차를 계산하는 표현식을 작성하는 단계

를 포함하는 속성 제공 방법.

청구항 6.

삭제

청구항 7.

삭제

청구항 8.

하나 이상의 논리 필드들에 의해 정의되는 추상적인 쿼리(abstract query)로부터 생성된 쿼리에 쿼리 논리(query logic)를 추가하기 위한 방법으로서,

상기 추상적인 쿼리의 각 조건마다,

실행가능한 쿼리 컨트리뷰션(executable query contribution)을 작성하는 단계로서, 상기 실행가능한 쿼리 컨트리뷰션을 작성하는 단계는, 상기 조건의 논리 필드에 대해 정의되는 액세스 메소드(access method)에 따라 쿼리 술어를 작성하고 물리적인 데이터에 액세스하는 메소드를 지정하는 단계를 포함하는, 상기 쿼리 컨트리뷰션 작성 단계와,

상기 추상적인 쿼리가 상기 조건의 논리 필드에 적용되는 복수의 상관 관계 표준들 중 적어도 하나를 포함하는지의 여부를 판정하는 단계로서, 상기 복수의 상관 관계 표준 각각은 소정의 값(a value)을 계산하기 위한 대응하는 상관 관계 속성 정의를 가지며, 주어진 상관 관계 표준에 대해 적어도 2 개의 상이한 대응하는 상관 관계 속성 정의가 정의되는 것인, 상기 판정 단계

를 포함하는 쿼리 논리 추가 방법.

청구항 9.

제8항에 있어서,

상기 추상적인 쿼리가 상기 조건의 상기 논리 필드에 적용되는 상기 복수의 상관 관계 표준들 중 적어도 하나를 포함한다면, 상기 상관 관계 표준 및 상기 상관 관계 속성 정의로부터 쿼리 술어(query predicate)를 작성하는 단계를 더 포함하는 쿼리 논리 추가 방법.

청구항 10.

삭제

청구항 11.

청구항 11은(는) 설정등록료 납부시 포기되었습니다.

제8항에 있어서,

상기 값은 연령값, 시간값, 및 장소값 중 하나일 수 있는 것인 속성 제공 방법.

청구항 12.

청구항 12은(는) 설정등록료 납부시 포기되었습니다.

제8항에 있어서,

범위(span) 상관 관계 표준이 상기 추상적인 쿼리의 2 개의 조건들의 논리 필드들에 적용되는 것으로 판정하는 단계와,

상기 논리 필드들에 대해 정의되고 상기 범위 상관 관계 표준에 대응하는 상기 상관 관계 속성 정의들에 의해 계산된 각각의 값 간의 차를 계산하는 표현식을 작성하는 단계를 더 포함하는 속성 제공 방법.

청구항 13.

실행될 때, 하나 이상의 논리 필드들에 의해 정의되는 추상적인 쿼리로부터 생성된 쿼리에 쿼리 논리(query logic)를 추가하는 동작을 수행하는 프로그램을 포함하는 컴퓨터 판독 가능 매체로서,

상기 동작은 제8항, 제9항, 제11항 및 제12항 중 어느 한 항에 따른 각각의 단계를 포함하는 것인,

컴퓨터 판독 가능 매체.

청구항 14.

삭제

청구항 15.

삭제

청구항 16.

삭제

청구항 17.

삭제

청구항 18.

컴퓨터 시스템으로서,

상기 정보는,

추상적인 쿼리들을 정의하는 복수의 논리 필드들을 포함하는 쿼리 명세(query specification)와,

상기 복수의 논리 필드들 중 적어도 두 개에 대해 지정된 적어도 하나의 상관 관계 속성 정의로서, 각 상관 관계 속성 정의는 상기 상관 관계 속성 정의의 값을 계산하기 위한 이름과 정의를 포함하는, 상기 적어도 하나의 상관 관계 속성 정의와,

데이터에 대해 발행된 추상적인 쿼리의 수신에 응답하여 소정의 동작을 수행하도록 실행가능한 런타임 컴포넌트(runtime component)로서, 상기 추상적인 쿼리는 상기 쿼리 명세에 따라 정의되고, 상기 적어도 하나의 상관 관계 속성 정의 각각이 지정된 상기 적어도 두 개의 논리 필드로 구성되는 것인, 상기 런타임 컴포넌트

를 포함하며,

상기 소정의 동작은

범위 상관 관계 표준이 상기 추상적인 쿼리의 2 개의 조건들의 논리 필드들에 적용되는 것으로 판정하는 단계로서, 상기 논리 필드들은 적어도 2 개의 논리 필드들을 포함하는, 상기 판정 단계와,

상기 논리 필드들에 대해 정의되고 상기 범위 상관 관계 표준에 대응하는 상기 상관 관계 속성 정의들에 의해 계산된 각각의 값 간의 차를 계산하는 표현식을 작성하는 단계와,

상기 적어도 2 개의 논리 필드들에 대한 상기 상관 관계 속성값을 계산하는 단계를 포함하는 것인, 컴퓨터 시스템.

청구항 19.

청구항 19은(는) 설정등록료 납부시 포기되었습니다.

제18항에 있어서,

상기 복수의 논리 필드들을 물리적인 데이터 개체들(data entities)에 매핑시키는 복수의 매핑 규칙들(mapping rules)을 더 포함하는 컴퓨터 시스템.

청구항 20.

청구항 20은(는) 설정등록료 납부시 포기되었습니다.

제18항에 있어서,

상기 값은 시간값인 컴퓨터 시스템.

청구항 21.

정보가 저장되어 있는 컴퓨터 판독가능 매체로서,

상기 정보는

추상적인 쿼리들을 정의하기 위한 복수의 논리 필드들을 포함하는 쿼리 명세(query specification)와,

상기 복수의 논리 필드들 중 적어도 2 개에 대해 지정된 복수의 상관 관계 속성 정의들로서, 상기 상관 관계 속성 정의 각각은 그 상관 관계 속성 정의의 값을 계산하기 위한 이름과 정의를 포함하는 것인, 상기 복수의 상관 관계 속성 정의들과,

데이터에 대해 발행된 추상적인 쿼리의 수신에 응답하여 소정의 동작을 수행하도록 실행가능한 런타임 컴포넌트(runtime component)로서, 상기 추상적인 쿼리는 상기 쿼리 명세에 따라 정의되고, 상기 상관 관계 속성 정의 각각이 지정된 상기 적어도 2 개의 논리 필드들로 구성되는 것인, 상기 런타임 컴포넌트

를 포함하며,

상기 소정의 동작은

범위 상관 관계 표준이 상기 추상적인 쿼리의 2 개의 조건들의 논리 필드들에 적용되는 것으로 판정하는 단계로서, 상기 논리 필드들은 적어도 2 개의 논리 필드들을 포함하는, 상기 판정 단계와,

상기 논리 필드들에 대해 정의되고 상기 범위 상관 관계 표준에 대응하는 상기 상관 관계 속성 정의들에 의해 계산된 각각의 값 간의 차를 계산하는 표현식을 작성하는 단계와,

상기 논리 필드들에 대한 상기 상관 관계 속성값을 계산하는 단계를 포함하는 것인, 컴퓨터 판독가능 매체.

청구항 22.

삭제

청구항 23.

삭제

명세서

기술분야

본 발명은 일반적으로 데이터 처리에 관한 것으로, 보다 상세하게는, 상관 관계 표준 또는 표준들을 이용하여 데이터를 액세스하는 것에 관한 것이다.

배경기술

데이터베이스는 컴퓨터 처리된 정보의 저장 및 검색 시스템이다. 관계형 데이터베이스 관리 시스템은 데이터를 저장하고 검색하는 관계형 기법들을 이용하는 컴퓨터 데이터베이스 관리 시스템(DBMS)이다. 가장 일반적인 유형의 데이터베이스는 관계형 데이터베이스로서, 이는 데이터를 다수의 상이한 방식으로 재구성하고 액세스할 수 있도록 데이터를 정의한 테이블 방식(tabular)의 데이터베이스이다.

DBMS에서는, 특정 아키텍처에 관계없이, 요청하는 개체(requesting entity; 예를 들어, 응용 프로그램 또는 운영 체제)는 데이터베이스 액세스 요청을 발행하여 특정 데이터베이스로의 액세스를 요구한다. 이러한 요청들은, 예컨대, 단순한 카탈로그 참조 요청이나 트랜잭션, 및 데이터베이스의 특정 레코드를 판독하고, 변경하고, 추가하도록 동작하는 트랜잭션의 조합들을 포함할 수 있다. 이들 요청들은 구조화된 쿼리 언어(Structured Query Language; SQL) 등의 고급 쿼리 언어를 사용하여 행해진다. 예시적으로, SQL은 IBM사의 DB2, Microsoft사의 SQL 서버 등의 데이터베이스와, Oracle, Sybase, 및 Computer Associates사로부터 구매가능한 데이터베이스 제품들로부터 정보를 얻어 이를 갱신하는 대화형 쿼리들(interactive queries)을 작성하는 데 사용된다. 본원에서 사용된 "쿼리"란 용어는 저장된 데이터베이스로부터 데이터를 검색하는 명령들(commands)의 집합을 지칭하는 것이다. 쿼리들은 프로그래머들 및 프로그램들이 데이터의 위치를 선택하고, 삽입하고, 갱신하고, 찾아 내는 것 등을 허락하는 명령 언어의 형식을 취한다.

데이터베이스 또는 데이터 웨어하우스(warehouse)에 저장된 정보에 대해 복잡한 쿼리들을 작성할 경우에는, 공통 속성(common attribute)에 기반하여 쿼리인 데이터 개체들을 상관시키는 것이 바람직하다. 예컨대, 데이터 개체들은 모든 개체들 또는 사건들(events)-이들은 (i) 주어진 시점에서 또는 주어진 기간 내에 발생됨; (ii) 서로 동시에 또는 주어진 기간 내에 발생됨; (iii) 여러 가지의 상이한 방식으로 범위가 정해질 수 있는 동일 지역(예컨대, 동일한 거리 주소, 도시, 카운티, 주, 우편 번호, 또는 국가)에서 발생됨; (iv) 해당 당사자 개인 또는 항목이 주어진 연령 또는 주어진 연령 범위 내인 경우에 발생됨; (v) 당해 모든 데이터 개체들에 관련된 연령이 동일한 경우에 발생됨-에 기초하여 상관될 수 있다.

이러한 복잡한 쿼리들을 작성하는 전형적인 접근법은, 쿼리인 각 데이터 개체들에 대한 공통적 상관 관계 속성(들)의 논리(logic)를 계산에 넣으라는(factor in)라는 추가 술어(predicates)에 의해 쿼리를 확장(augment)하는 것을 포함한다. 그러나, 불행히도, 이런 접근법은 관련 데이터와 쿼리인 특정 데이터 개체들의 물리적인 관계에 대한 심도 깊은 이해(예컨대, 데이터 개체가 기간, 연령, 또는 지역의 개념을 지원하는 가에 대한 이해)를 필요로 한다. 일례로서, 소정 임계치 이상의 간

(liver) 효소 수치를 갖는 것으로 발견되었고 그 간 이형을 고치기 위해 외과적 치료를 받은 현재 60세 또는 그 이상의 남성들을 식별해 내는 것에 관심이 있는 사용자를 생각해 보기로 하자. SQL에서, 대응하는 쿼리는 Query 1의 형식을 취할 수 있다.

SQL QUERY 1

```
SELECT ASTValue, Diagnostics, SurgeryType
FROM Demographics, TestResults, SurgeryRecord
WHERE Gender ="M" AND YEAR (CURRENT_DATE - BirthDate) >= 60
AND ASTValue > 30 AND SurgeryType = "Liver"
```

이제, 사용자는 반환되는 데이터가, 검사를 받고 30일 이내에 외과적 치료를 받은 사람인 케이스들만을 포함하도록 한정되기를 바라는 것으로 가정한다. 이는 Query 2로 예시된 바와 같이, SQL Query 1을 동일 기간 내에 발생한 검사 및 외과적 치료 사건들을 선택하라는 추가 술어에 의해 확장시킴에 의해 달성될 수 있다.

SQL QUERY 2

```
SELECT ASTValue, Diagnostics, SurgeryType
FROM Demographics, TestResults, SurgeryRecord
WHERE Gender ="M" AND YEAR (CURRENT_DATE - BirthDate) >= 60
AND ASTValue > 30 AND SurgeryType = "Liver" AND
DAYS (SurgeryDate - TestDate) <= 30
```

다른 케이스에서는, 사용자는 동일한 기준 조합에 의해 개개인들을 식별하되, 그 결과들을 검사를 받고 외과적 치료를 받은 40대와 50대 사이의 사람들로만 한정시키길 바랄 수 있다. 이는 Query 3으로 예시된 바와 같이, 기본 쿼리에 다른 집합의 술어들을 필요로 할 것이다.

SQL QUERY 3

```
SELECT ASTValue, Diagnostics, SurgeryType
FROM Demographics, TestResults, SurgeryRecord
WHERE Gender ="M" AND YEAR (CURRENT_DATE - BirthDate) >= 60
AND ASTValue > 30 AND SurgeryType = "Liver" AND
YEAR (SurgeryDate - BirthDate) >= 40 AND YEAR (SurgeryDate -
BirthDate) <= 50 AND YEAR (TestDate - BirthDate) >= 40 AND YEAR
(TestDate - BirthDate) <= 50
```

상술한 예제에서는, 비록 각 케이스에서 기본 쿼리는 동일하게 남아 있지만, 쿼리에 의해 조사된 개체들 간의 원하는 상관 관계를 달성하기 위해서는 추가의 술어들이 부가되는 것을 보여 주고 있다. 결과적으로, 원하는 데이터를 액세스하려는 최종 사용자에게는 상당한 부담이 된다.

그러므로, 쿼리들 내에서 상관 관계 논리를 구현하는 방법이 필요로 된다.

발명의 상세한 설명

본 발명은 일반적으로 상관 관계 표준 또는 표준들을 이용하여 데이터를 액세스하기 위한 방법, 시스템, 및 제조 물품에 관한 것이다.

본 발명의 일 실시예에서는 하나 이상의 논리 필드(logical field)에 의해 정의되는 추상적인 쿼리(abstract query)로부터 생성된 쿼리에 쿼리 논리를 부가하도록 구성된 속성들을 제공하는 방법을 제공한다. 본 발명의 방법은 특정 논리 필드에 대해 복수의 상관 관계 속성들 중 어느 것이 지원되는가를 지정하는 단계와, 특정 논리 필드에 대해 지원된 상호 관계 속성들의 값을 계산하기 위한 정의(definition)를 지정하는 단계를 포함한다.

본 발명의 다른 실시예에서는 하나 이상의 논리 필드에 의해 정의되는 추상적인 쿼리로부터 생성된 쿼리에 쿼리 논리를 부가하도록 구성된 속성들을 제공하는 방법을 제공한다. 본 발명의 방법은 복수의 논리 필드 각각마다 특정 논리 필드에 대

해 복수의 상관 관계 속성들 중 어느 것이 지원되는 가를 지정하는 단계와, 복수의 논리 필드 각각마다 특정 논리 필드에 대해 지원된 상관 관계 속성들의 값을 계산하기 위한 정의(definition)를 지정하는 단계를 포함하며; 여기서, 상이한 두 논리 필드에 대해 지정된, 특정 상관 관계 속성의 값을 계산하기 위한 정의들 중 적어도 두 개는 상이하다.

본 발명의 다른 실시예에서는 하나 이상의 논리 필드에 의해 정의되는 추상적인 쿼리로부터 생성된 쿼리에 쿼리 논리를 추가하는 방법을 제공하며, 이 방법은 추상적인 쿼리의 각 조건(condition)에 대해; 실행가능한 쿼리 컨트리뷰션(executable query contribution)을 작성하는 단계와, 추상적인 쿼리가 그 조건의 논리 필드에 적용되는 복수의 상관 관계 표준들 중 적어도 하나를 포함하는 가를 판정하는 단계를 포함하며; 여기서, 복수의 상관 관계 표준 각각은 소정의 값을 계산하기 위한 대응하는 상관 관계 속성 정의를 가지며, 소정의 상관 관계 표준에 대해 적어도 두 개의 상이한 대응하는 상관 관계 속성 정의들이 정의된다.

본 발명의 또 다른 실시예에서는 실행될 때, 하나 이상의 논리 필드에 의해 정의되는 추상적인 쿼리로부터 생성된 쿼리에 쿼리 논리를 추가하는 동작(operation)을 수행하는 프로그램을 포함한 컴퓨터 판독가능 매체를 제공한다. 이 동작은 추상적인 쿼리의 각 조건에 대해; 실행가능한 쿼리 컨트리뷰션을 구축하고, 추상적인 쿼리가 그 조건의 논리 필드에 적용되는 복수의 상관 관계 표준들 중 적어도 하나를 포함하는 가를 판정하는 것을 포함하며; 여기서, 복수의 상관 관계 표준 각각은 소정의 값을 계산하기 위한 대응하는 상관 관계 속성 정의를 가지며, 소정의 상관 관계 표준에 대해 적어도 두 개의 상이한 대응하는 상관 관계 속성 정의들이 정의된다.

본 발명의 또 다른 실시예에서는 정보가 저장되어 있는 컴퓨터 판독가능 매체를 제공하며, 이 정보는 추상적인 쿼리들을 정의하는 복수의 논리 필드들을 포함하는 쿼리 명세(query specification), 복수의 논리 필드들 중 적어도 하나에 대해 지정된 적어도 하나의 상관 관계 속성 정의-이 상관 관계 속성 정의는 그 상관 관계 속성 정의의 값을 계산하기 위한 이름과 정의를 포함함-, 및 데이터에 대해 발행된 추상적인 쿼리-이 추상적인 쿼리는 쿼리 명세에 따라 정의되고 적어도 하나의 상관 관계 속성 정의가 지정된 적어도 하나의 논리 필드로 구성됨-의 수신에 응답하여 소정의 동작을 수행하도록 실행가능한 런타임 컴포넌트(runtime component)를 포함한다. 이 소정의 동작은 적어도 하나의 논리 필드에 대한 상관 관계 속성값을 계산하는 것을 포함한다.

본 발명의 또 다른 실시예에서는 정보가 저장되어 있는 컴퓨터 판독가능 매체를 제공하며, 이 정보는 추상적인 쿼리들을 정의하는 복수의 논리 필드들을 포함하는 쿼리 명세, 복수의 논리 필드들 중 적어도 두 개에 대해 지정된 복수의 상관 관계 속성 정의-각각의 상관 관계 속성 정의는 그 상관 관계 속성 정의의 값을 계산하기 위한 이름과 정의를 포함함-, 및 데이터에 대해 발행된 추상적인 쿼리-이 추상적인 쿼리는 쿼리 명세에 따라 정의되고 적어도 하나의 상관 관계 속성 정의가 지정된 적어도 하나의 논리 필드로 구성됨-의 수신에 응답하여 소정의 동작을 수행하도록 실행가능한 런타임 컴포넌트를 포함한다. 이 소정의 동작은 적어도 하나의 논리 필드에 대한 상관 관계 속성값을 계산하는 것을 포함한다.

실시예

도입

본 발명은 일반적으로 데이터를 액세스하기 위한 시스템, 방법, 및 제조 물품에 관한 것이다. 일반적으로, 데이터 액세스는 상호 관계 표준 또는 표준들로 구성된 쿼리에 의해 용이해진다. 상관 관계 표준의 존재는 쿼리의 조건들에 적용될 추가 상관 관계 논리의 도입을 용이하게 해준다. 일반적으로, 기간, 물리적 장소, 연령에 기초하여 개체들 간의 상관 관계를 지원하는 상관 관계 표준이 고려된다.

일 실시예에서, 상관 관계 속성들은 데이터의 논리적 모델의 일부로서 구현된다. 논리적 모델은 본원에서 데이터 저장소 추상화 계층(data repository abstraction layer)으로서 참조되며, 이는 기본(underlying) 데이터 저장소에 대한 논리적 뷰(logical view)를 제공한다. 이와 같이 하여, 데이터는 그 데이터가 물리적으로 표현되는 특정 방식과는 독립적으로 만들어진다. 쿼리 추상화 계층 또한 제공되며, 데이터 저장소 추상화 계층에 기반을 둔다. 런타임 컴포넌트는 추상적인 쿼리를 특정 물리적 데이터 표현에 대해 사용될 수 있는 형식(form)으로의 변환을 수행한다.

설명의 편의 상, 본원에서는 "상관 관계(correlation)"란 용어는 데이터 저장소 추상화 계층의 일부로서 정의된 쿼리 표준과 속성을 참조하는 것으로 사용된다. "상관 관계"란 용어는 쿼리 논리가 공통 개체에 대해 둘 이상의 쿼리 조건들에 적용될 수 있는 본 발명의 일 양상을 나타낸다. 그러나, 다른 양상에서는, 쿼리 논리는 또한 하나의 쿼리 조건에만 적용될 수 있다. 이하의 기술로부터 알 수 있는 바와 같이, 최종 사용자가 개체를 계산해야 하는 방법을 결정하는 것에 대한 부담을 갖지 않는다는 점에서, 최종 사용자측에서의 이점이 여전히 실현된다. 이처럼, (둘 이상의 쿼리 조건들 간의 단순한 상관 관

계에 대하여) 쿼리 조건과, 예컨대, 특정 시간, 장소 또는 연령 속성 간의 "상관 관계"가 발생한다. 중요하게는, 상관 관계 속성은 쿼리에 전체로서 적용되어, 특정 유형의 상관 관계 속성을 지원하는 모든 조건 필드들에 연관된 데이터 선택 논리에 영향을 미칠 것이다.

본 발명의 일 실시예는, 예컨대, 도 1에 도시되고 이하에서 기술될 컴퓨터 시스템 등의 컴퓨터 시스템에서 사용하기 위한 프로그램 제품으로서 구현된다. 프로그램 제품의 프로그램(들)은 실시예들의 함수(본원에서 기술된 메소드들(methods)을 포함)을 정의하며, 각종의 신호 전달 매체 상에 포함될 수 있다. 신호 전달 매체의 예로서, (i) 기록 불가 저장 매체(예컨대, CD-ROM 드라이브에 의해 판독가능한 CD-ROM 디스크 등의 컴퓨터 내의 판독 전용 메모리 장치) 상에 영구적으로 저장된 정보, (ii) 기록 가능 저장 매체(예컨대, 디스켓 드라이브 또는 하드 디스크 드라이브 내의 플로피 디스크) 상에 저장된 변경가능한 정보, 또는 (iii) 무선 통신을 비롯해, 컴퓨터 또는 전화 통신망 등을 통해 통신 매체에 의해 컴퓨터로 전달된 정보를 포함하지만, 이들에만 한정되는 것은 아니다. 나중의 실시예에서는 구체적으로 인터넷 및 기타 통신망으로부터 다운로드된 정보를 포함한다. 이런 신호 전달 매체는, 본 발명의 함수들을 전달하는 컴퓨터 판독가능 명령어들을 포함할 때, 본 발명의 실시예들을 나타낸다.

일반적으로, 본 발명의 실시예들을 구현하도록 실행된 루틴들은 운영 체제 또는 특정 응용 프로그램, 컴포넌트, 프로그램, 모듈, 객체(object), 명령어 시퀀스의 일부일 수 있다. 본 발명의 소프트웨어는 전형적으로 기계-판독가능 포맷으로 원시(native) 컴퓨터에 의해 변환될 수많은 명령어 및 실행가능 명령어들로 구성된다. 또한, 프로그램들은 프로그램에 지역적으로 상주하거나 메모리 내에서 또는 저장 장치 상에서 발견되는 변수(variables) 및 데이터 구조들로 구성된다. 또한, 이 후에 기술될 각종 프로그램들은 이들을 본 발명의 특정 실시예에서 구현하는 응용 프로그램에 기초하여 식별될 수 있다. 그러나, 이하에 기술될 임의의 특정 전문어는 단지 설명의 편의 상 사용된 것으로, 본 발명은 이런 전문어에 의해 식별 및 또는 암시되는 임의의 특정 응용 프로그램에서 독점으로 사용되는 것으로 제한되는 것은 아닌 것에 유의해야 한다.

환경의 물리적 뷰

도 1은 본 발명의 실시예들을 구현할 수 있는 네트워킹 시스템(100)의 블록도를 도시한 것이다. 일반적으로, 네트워킹 시스템(100)은 클라이언트(예컨대, 사용자) 컴퓨터(102)(세 대의 클라이언트 컴퓨터(102)가 도시됨) 및 적어도 하나의 서버 컴퓨터(104)(하나의 서버 컴퓨터(104)가 도시됨)를 포함한다. 클라이언트 컴퓨터(102) 및 서버 컴퓨터(104)는 네트워크(126)를 통해 연결되어 있다. 일반적으로, 네트워크(126)는 근거리 통신망(LAN) 및/또는 광역 통신망(WAN)일 수 있다. 특정 실시예에서, 네트워크(126)는 인터넷이다.

클라이언트 컴퓨터(102)는 버스(126)를 통해 메모리(112), 저장 장치(114), 입력 장치(116), 출력 장치(119), 및 네트워크 인터페이스 컴포넌트(118)에 연결된 중앙 처리 장치(CPU; 110)를 포함한다. 입력 장치(116)는 클라이언트 컴퓨터(102)에 입력을 제공하는 임의의 장치일 수 있다. 예를 들면, 키보드, 키패드, 라이트-펜, 터치-스크린, 트랙-볼, 또는 음성 인식 장치, 오디오/비디오 플레이어, 등이 사용될 수 있다. 출력 장치(119)는 사용자에게 출력을 제공하는 임의의 장치, 예컨대, 통상의 디스플레이 스크린일 수 있다. 비록 출력 장치(119)가 입력 장치(116)와 분리되어 도시되어 있지만, 출력 장치(119)와 입력 장치(116)는 결합될 수 있다. 예를 들어, 터치-스크린을 일체화시킨 디스플레이 스크린, 키보드를 일체화시킨 디스플레이, 또는 텍스트 음성 변환기와 결합된 음성 인식 장치를 사용할 수 있다.

네트워크 인터페이스 컴포넌트(118)는 클라이언트 컴퓨터(102)와 서버 컴퓨터(104) 간의 네트워크 통신을 네트워크(126)를 통해 허용하도록 구성된 임의의 입력/출구(entry/exit) 컴포넌트일 수 있다. 예컨대, 네트워크 인터페이스 컴포넌트(118)는 네트워크 어댑터 또는 다른 네트워크 인터페이스 카드(NIC)일 수 있다.

바람직하게는, 저장 장치(114)는 직접 액세스 저장 장치(Direct access storage device; DASD)이다. 비록 저장 장치가 단일 장치로서 도시되어 있지만, 고정형 디스크 드라이브, 플로피 디스크 드라이브, 테이프 드라이브, 가변형 메모리 카드, 또는 광 저장 장치 등의 고정형 및/또는 가변형 저장 장치의 조합일 수 있다. 메모리(112) 및 저장 장치(114)는 다수의 주 저장 장치와 보조 저장 장치에 걸쳐있는 하나의 가상 어드레스 공간의 일부일 수 있다.

메모리(112)는 바람직하기로는, 본 발명에 필요한 프로그래밍 및 데이터 구조들을 보유할 정도로 충분히 큰 랜덤 액세스 메모리이다. 비록 메모리(112)를 단일의 개체로서 도시하였지만, 메모리(112)는 실제로는 복수의 모듈들을 포함할 수 있으며, 고속 레지스터 및 캐시들로부터 저속이지만 대용량의 DRAM 칩들에 이르기까지 다양한 레벨로 존재할 수 있다.

예시적으로, 메모리(112)는 운영 체제(124)를 포함한다. 유리하게 이용될 수 있는 운영 체제의 예로서는, 리눅스(Linux) 및 마이크로소프트사의 Windows®이 있다. 보다 일반적으로는, 본원에서 개시된 함수들을 지원하는 어떠한 운영 체제도 사용될 수 있다.

메모리(112)는 또한, CPU(110) 상에서 실행될 때, 각종 서버 컴퓨터들(104) 사이를 탐색하고(navigating), 하나 이상의 서버 컴퓨터들(104)에서 네트워크 주소의 위치를 찾아 내기 위한 지원(support)을 제공하는 브라우저 프로그램(122)을 포함하는 것으로 도시되어 있다. 일 실시예에서, 브라우저 프로그램(122)은 웹-기반 그래픽 사용자 인터페이스(GUI)를 포함하는데, 이 GUI는 사용자가 하이퍼 텍스트 마크업 언어(HTML) 정보를 디스플레이하는 것을 가능케 한다. 그러나, 보다 일반적으로는, 브라우저 프로그램(122)은 서버 컴퓨터(104)로부터 전송된 정보를 렌더링할 수 있는 임의의 프로그램(바람직하게는, GUI-기반)일 수 있다.

서버 컴퓨터(104)는 클라이언트 컴퓨터(102)와 유사한 방식으로 물리적으로 배치될 수 있다. 따라서, 서버 컴퓨터(104)는 일반적으로 버스(136)에 의해 서로 연결되어진 CPU(130), 메모리(132), 및 저장 장치(134)를 포함하는 것으로 도시되어 있다. 메모리(132)는 서버 컴퓨터(104) 상에 위치해 있는 필요한 프로그래밍 및 데이터 구조들을 보유하기에 충분히 큰 랜덤 액세스 메모리일 수 있다.

서버 컴퓨터(104)는 일반적으로 메모리(132)에 상주하는 것으로 도시된 운영 체제(138)의 제어 하에 있다. 운영 체제(138)의 예로서는, IBM OS/400®, UNIX, Microsoft Window® 등 있다. 보다 일반적으로는, 본원에서 개시된 함수들을 지원할 수 있는 어떠한 운영 체제라도 사용될 수 있다.

메모리(132)는 또한 하나 이상의 응용 프로그램(140) 및 추상적인 쿼리 인터페이스(146)를 포함한다. 응용 프로그램(140) 및 추상적인 쿼리 인터페이스(146)는 컴퓨터 시스템(100)의 각종 메모리 및 저장 장치들에 여러 시점들에서 상주해 있는 복수의 명령어들을 포함하는 소프트웨어 제품들이다. 응용 프로그램(140) 및 추상적인 쿼리 인터페이스(146)가 서버 컴퓨터(104)의 하나 이상의 프로세서(130)에 의해 관독되어 실행될 때, 응용 프로그램(140) 및 추상적인 쿼리 인터페이스(146)는 컴퓨터 시스템(100)에게 본 발명의 여러 특징들을 구체화시키는 단계들(steps) 또는 엘리먼트들(elements)을 실행하는 데 필요한 단계들을 수행하도록 하게 한다. 응용 프로그램(140)(및, 보다 일반적으로는, 운영 체제(138) 및 최고 레벨에서의 사용자들을 비롯하여, 요청하는 임의 개체)은 데이터베이스(예를 들어, 데이터베이스(156)₁...156_N; 집합적으로 데이터베이스(156)이라 함)에 대해 쿼리들을 발행한다. 예시적으로, 데이터베이스(156)는 저장 장치(134)의 데이터베이스 관리 시스템(DBMS)의 일부로서 도시되어 있다. 데이터베이스(156)는 특정의 물리적 표현에 관계없이 임의 데이터 집합체(collection)를 나타낸 것이다. 일례로서, 데이터베이스(156)는 관계형 스키마(relational schema)(SQL 쿼리에 의해 액세스가능함)에 따라, 또는 XML 스키마(XML 쿼리에 의해 액세스가능함)에 따라 체계화될 수 있다. 그러나, 본 발명은 특정 스키마로만 제한되는 것이 아니라, 현재 알려지지 않은 스키마로의 확장도 고려된다. 본원에서 사용된 "스키마"란 용어는 일반적으로 특정의 데이터 배열을 일컫는 것이다.

일 실시예에서, 응용 프로그램(140)에 의해 발행되는 쿼리들은 각 응용 프로그램(140)에 포함된 응용 프로그램 쿼리 명세(142)에 따라 정의된다. 응용 프로그램(140)에 의해 발행되는 쿼리들은 미리 정의될 수 있거나(즉, 응용 프로그램(140)의 일부로서 하드(hard) 코딩될 수 있음), 또는 입력(예를 들면, 사용자 입력)에 응답하여 생성될 수 있다. 어느 경우에도, 쿼리들(본원에서는 "추상적인 쿼리들"이라 일컬음)은 추상적인 쿼리 인터페이스(146)에 의해 정의된 논리 필드들을 이용하여 구성/실행된다. 특히, 추상적인 쿼리들에서 사용되는 논리 필드들은 추상적인 쿼리 인터페이스(146)의 데이터 저장소 추상화 컴포넌트(148)에 의해 정의된다. 추상적인 쿼리들은 런타임 컴포넌트(150)에 의해 실행되며, 이 런타임 컴포넌트(150)는 먼저 추상적인 쿼리들을 DBMS(154)에 포함된 데이터의 물리적 표현과 일치하는 형식(form)으로 변환시킨다.

일 실시예에서, 데이터 저장소 추상화 컴포넌트(148)는 상관 관계 속성(162)로 구성된다. 상관 관계 속성(162)은, 예를 들면, 하나의 쿼리의 조건들 간의 관계들을 정의하는 데 사용되는 논리 필드 상에 위치될 수 있는 공간 및 기간 제한 조건들(constraints)을 지정할 수 있다. 런타임 컴포넌트(150)의 상관 관계 알고리즘(151)은 상관 관계 속성(162)에 의해 지정된 규칙들(rules)을 시행하도록 동작한다. 이를 위해, 런타임 컴포넌트(150)는 범위(span) 표준 목록(170), 개인 표준 목록(172), 및 범위 필드 목록(174)을 포함하는 일시적 데이터 구조들을 유지할 수 있다.

일 실시예에서, 쿼리의 엘리먼트들은 그래픽 사용자 인터페이스(GUI)를 통해 사용자에게 의해 지정된다. GUI의 성분(content)은 응용 프로그램(140)에 의해 생성된다. 특정 실시예에서, GUI의 성분은 브라우저 프로그램(122)에 의해 클라이언트 컴퓨터(102) 상에서 렌더링될 수 있는 하이퍼텍스트 마크업 언어(HTML) 성분이다. 따라서, 메모리(132)는 클라이언트 컴퓨터(102)로부터의 요청들을 서비스하도록 적응된 하이퍼텍스트 전송 프로토콜(hypertext Transfer Protocol (http)) 서버 프로세스(152)(예컨대, 웹 서버)를 포함한다. 예를 들어, 서버 프로세스(152)는 예시적으로 서버(104) 상에 상주해 있는 데이터베이스(156)로의 액세스에 대한 요청들에 응답할 수 있다. 데이터베이스(156) 중의 데이터에 대해 입력되는 클라이언트 요청들은 응용 프로그램(140)을 호출하여 기동시킨다(invoked). 응용 프로그램(140)이 프로세서(130)에 의해 실행될 때, 응용 프로그램(140)은 서버 컴퓨터(104)에게 데이터베이스(156)로의 액세스를 비롯하여, 본 발명의

여러 양상들을 실현하는 단계들 또는 엘리먼트들을 수행하도록 하게 한다. 일 실시예에서, 응용 프로그램(140)은 GUI 엘리먼트들을 작성하도록 구성된 복수의 서브릿(servlets)을 포함하며, GUI 엘리먼트들은 이후에 브라우저 프로그램(122)에 의해 렌더링된다.

도 1은 네트워킹 클라이언트 컴퓨터(102)와 서버 컴퓨터(104)에 대한 단지 하나의 하드웨어/소프트웨어 구성을 도시한 것이다. 본 발명의 실시예들은 컴퓨터 시스템들이 복잡한 다중-사용자 컴퓨팅 장치, 단일-사용자 워크스테이션, 또는 자신들의 비휘발성 저장 장치를 갖지 않는 네트워크 장치들에 관계없이, 상당하는 어떠한 하드웨어 구성에도 적용시킬 수 있다. 또한, 비록 HTML을 포함한, 특정 마크업 언어에 대해 참조하였지만, 본 발명은 특정 언어, 표준, 또는 버전(version)으로만 제한되는 것은 아닌 것에 주목해야 할 것이다. 따라서, 당해 기술 분야의 숙련자들은 본 발명은 기타 마크업 언어 뿐 아니라 비(non)-마크업 언어에도 적용가능하며, 또한 본 발명은 특정 마크업 언어의 장래 변경 뿐 아니라 현재 알려지지 않은 기타 언어들에도 적용가능하다는 것을 인식할 수 있을 것이다. 마찬가지로, 도 1에 도시된 http 서버 프로세스(152) 역시 단지 예시에 불과한 것으로, 알려진 또는 알려지지 않은 임의의 프로토콜들을 지원하도록 적용된 다른 실시예들이 고려된다.

환경의 논리적/쿼리뷰

도 2A 및 도 2B에서는 본 발명의 컴포넌트들의 예시적인 관계형 뷰(200)를 보여 준다. 요청하는 개체(예컨대, 응용 프로그램(140) 중 하나)는 그 요청하는 개체의 각자의 응용 프로그램 쿼리 명세(142)에 의해 정의된 쿼리(202)를 발행한다. 최종 쿼리(202)를 본원에서는 일반적으로 "추상적인 쿼리"로 참조하는데, 이는 이 쿼리가 DBMS(154) 내의 기본 물리적 데이터 개체들에 대한 직접적인 참조에 의해서보다는 추상(즉, 논리) 필드에 따라 구성되기 때문이다. 결과적으로, 사용된 특정의 기본 데이터 표현과는 독립적인 추상적인 쿼리들을 정의할 수 있다. 일 실시예에서, 응용 프로그램 쿼리 명세(142)는 데이터 선택에 사용되는 표준(선택 표준(204)), 쿼리의 하나 이상(보다 일반적으로는, 둘 이상)의 조건들을 상호 관련시키는 데 사용되는 표준, 및 선택 표준(204)에 기초하여 반환(return)되어질 필드들의 명시적인 명세(반환 데이터 명세(206))를 포함할 수 있다.

도 2B에 보여진 추상적인 쿼리(202)에 대한 세부 사항들을 아래의 표 I에 나타내었다. 일례로서, 추상적인 쿼리(202)는 XML을 이용하여 정의된다. 그러나, 기타 임의다른 언어도 유리하게 사용될 수 있다.

표 I - 추상적인 쿼리 예제

TABLE I - ABSTRACT QUERY EXAMPLE

```

001 <QueryAbstraction>
002   <Selection>
003     <Condition field="Diagnosis" operator="EQ" value="Anemia"/>
004     <Condition field="Hemoglobin" operator="LT" value="20"
005       relOperator="AND"/>
006     <Correlation attribute="Age" operator="GT" value="40"/>
007     <Correlation attribute="Span(Time)" operator="LT" value="5"
008   relOperator="AND"/>
009   </Selection>
010   <Results>
011     <Field name="FirstName"/>
012     <Field name="LastName"/>
013   </Results>
014 </QueryAbstraction>
    
```

예시적으로, 표 1에 나타난 추상적인 쿼리는 선택 표준을 포함하는 선택 명세(라인 005-014) 및 결과 명세(라인 015-019)를 포함한다. 일 실시예에서, 선택 표준(본원에서는, "조건"으로도 참조됨)은 (논리 필드에 대한) 필드명, 비교 연산자(=, >, < 등), 및 값 표현식(비교되는 필드의 것)으로 구성된다. 일 실시예에서, 결과 명세는 쿼리 실행의 결과로서 반환되어질 추상 필드들의 목록이다. 추상적인 쿼리의 결과 명세는 필드명과 소팅(sort) 표준으로 구성될 수 있다. 예시적으로, 상관 관계 표준은 Age 상관 관계 표준 및 Span(Time) 상관 관계 표준이다. 상관 관계 표준 각각은 쿼리의 하나 이상의 조건들에 적용된다. 이 예제에서, Age 상관 관계 표준은 특정 진단 및 검사 결과 값들을 선택하는 데 사용되는 개개의 술어에 추가 논리를 부가한다. Span(Time) 상관 관계 표준은 쿼리에 추가 논리를 전체로서 부가하여, 이 경우에는, 각기 5일 이내의 진단 및 검사 결과를 갖는 환자들만을 선택한다.

비교예로서, 다음은 표 I의 추상적인 쿼리에 대응하는 SQL 쿼리를 나타낸 것이다.

SQL QUERY 4

```
SELECT f_name, l_name FROM patient WHERE
  (diag = 'Anemia' AND YEAR(diagdate-birthdate) > 40) AND
  (testval < 20 AND YEAR(testdate-birthdate) > 40) AND
  (ABS(DAYS(diagdate - testdate)) < 5)
```

여기에서, 사용자는 원하는 각 조건에 상관 관계 논리를 적용시킴에 따른 부담을 갖게 되는 것에 주목할 필요가 있다. 이에 대해, 본 발명에서는, 데이터 저장소 추상화 컴포넌트(148)에 미리 정의된 상관 관계 속성이 존재해 있는 각 조건에 상관 관계 표준을 적용시킨다. 이러한 점에서 데이터 저장소 추상화 컴포넌트(148)의 양상들에 대해 이하에서 보다 상세히 기술하기로 한다.

일 실시예에서, 상관 관계 표준(203), 선택 표준(204), 및 반환 데이터 명세(206) 각각은 사용자 인터페이스, 예컨대, 브라우저 프로그램(122)을 통해 사용자에게 의해 지정될 수 있다. 이를 위해, 사용자 인터페이스는 상관 관계 표준(203), 선택 표준(204), 및 반환 데이터 명세(206) 각각에 대해 독립된 입력 필드들을 포함할 수 있으며, 이에 의해 추상적인 쿼리(202)가 구성된다.

응용 프로그램 쿼리 명세(142)에 의해 지정되며 추상적인 쿼리(202)를 구성하는 데 사용되는 논리 필드들은 데이터 저장소 추상화 컴포넌트(148)에 의해 정의된다. 일반적으로, 데이터 저장소 추상화 컴포넌트(148)는 데이터 선택에 대한 표준, 쿼리 조건들의 상관 관계에 대한 표준을 지정하고 쿼리 동작으로부터 반환되는 데이터의 결과 형식을 지정하는 정보를, (사용자 입력 쿼리 조건들에 응답할 수 있는)응용 프로그램(140)에 의해 발행된 쿼리(예컨대, 추상적인 쿼리(202)) 내에서 사용될 수 있는 논리 필드들의 집합으로서 보여 준다. 논리 필드들은 DBMS(154)에서 사용되는 기본 데이터 표현과는 독립적으로 정의되므로, 기본 데이터 표현에 느슨하게(loosely) 결합되는 쿼리들을 구성할 수 있다.

일반적으로, 데이터 저장소 추상화 컴포넌트(148)는 복수개의 필드 명세(208₁, 208₂,...; 집합적으로 필드 명세(208)라 칭함)를 포함한다. 구체적으로는, 필드 명세는 추상적인 쿼리의 구성에 이용할 수 있는 각 논리 필드마다 제공된다. 일 실시예에서, 필드 명세(208)는 논리 필드명(210₁, 210₂,...; 집합적으로, 필드명(210)이라 칭함) 및 연관된 액세스 메소드(212₁, 212₂,...; 집합적으로, 액세스 메소드(212)라 칭함)를 포함한다. 예시적인 실시예에서, 필드 명세(208)는 또한 하나 이상의 카테고리명(216₁, 216₂, 및 216₃; 집합적으로, 카테고리명(216)이라 칭함)을 포함한다. 카테고리명은 논리 필드명 그룹에 연관된다. 예를 들어, 도 2B에서, 필드 명세(208₁, 및 208₂)는 Demographic 카테고리(216₁)의 일부이고, 필드 명세(208₃, 208₄, 및 208₅)는 Diagnostics 카테고리(216₂)의 일부이고, 필드 명세(208₆, 208₇, 및 208₈)는 Tests 카테고리(216₃)의 일부이다. 그러나, 카테고리의 사용은 단지 특정 실시예를 나타낸 것으로, 다른 실시예에서는 카테고리를 사용하지 않는다.

액세스 메소드(212)는 논리 필드명들을 어느 한 데이터베이스(예를 들어, 데이터베이스(156) 중 하나) 내의 특정 물리적 데이터 표현(214₁, 214₂,...214_N)에 연관(즉, 매핑)시킨다. 예로서, 도 2A에는 두 개의 데이터 표현, 즉 XML 데이터 표현(214₁)과 관계형 데이터 표현(214₂)이 도시되어 있다. 그러나, 물리적 데이터 표현(214_N)은 알려지거나 알려지지 않은 임의의 다른 데이터 표현이 고려되는 것을 나타낸다.

일 실시예에서, 단일의 데이터 저장소 추상화 컴포넌트(148)는 둘 이상의 물리적 데이터 표현(214)에 대한 필드 명세(연관된 액세스 메소드와 함께)를 포함한다. 다른 실시예에서, 각각의 독립된 물리적 데이터 표현(214)마다 단일의 다른 데이터 저장소 추상화 컴포넌트(148)가 제공된다. 또 다른 실시예에서는, 다수의 데이터 저장소 추상화 컴포넌트(148)가 제공되며, 각각의 데이터 저장소 추상화 컴포넌트(148)는 동일한 기본 물리적 데이터의 상이한 부분(하나 이상의 물리적 데이터 표현(214)을 포함할 수 있음)을 보여 준다. 이와 같이, 다수의 사용자가 동시에 단일 응용 프로그램(140)을 사용하여, 응용 프로그램에 보여진 기본 데이터의 특정 부분들을 각자의 데이터 저장소 추상화 컴포넌트(148)에 의해 결정하는 동일한 기본 데이터를 액세스할 수 있다.

지원되는 상이한 유형의 논리 필드들의 수에 따라 액세스 메소드의 수가 고려된다. 일 실시예에서는, 단순(simple) 필드들, 필터된(filtered) 필드들, 및 구성된(composed) 필드들에 대한 액세스 메소드가 제공된다. 필드 명세(208₁ 및 208₂)는 단순 필드 액세스 메소드(212₁ 및 212₂) 각각을 예시한 것이다. 단순 필드들은 기본 물리적 데이터 표현 내의 특정 개체에 직접적으로 매핑된다(예를 들어, 필드는 주어진 데이터베이스 테이블 및 컬럼에 매핑됨). 일례로서, 도 2C에 도시된 단순 필드 액세스 메소드(212₁)는 논리 필드명(210₁)("FirstName")을 "환자"로 명명된 테이블 내의 "f_name"로 명명된 컬럼에 매핑시킨다. 필터된 필드들(도 2A 내지 도 2C에는 예시되지 않음)은 연관된 물리 개체를 식별하고, 물리적 데이터 표현 내의 항목들의 특정 부분집합을 정의하는 데 사용되는 규칙들을 제공한다. 필터된 필드의 예로는, ZIP 코드들의 물리적 표현에 매핑되고, 데이터를 뉴욕주에 대해 정의된 이들 ZIP 코드로만 데이터를 제한시키는 뉴욕 ZIP 코드 필드가 있다. 구성된 액세스 메소드(예를 들어, 213₁ 및 213₂)는 액세스 메소드 정의의 일부로서 제공되어진 표현식을 이용하여 하나 이상의 물리 필드들로부터 논리 필드들을 계산한다. 이와 같이 하여, 기본 데이터 표현에 존재하지 않는 정보를 계산할 수 있다. 일례로서, 판매 가격 필드를 판매 세율로 곱하여 구성된 판매 세금 필드가 있다.

기본 데이터의 임의 주어진 데이터 타입(예를 들어, 날짜, 십진수 등)에 대한 포맷들이 변할 수 있는 것으로 고려된다. 이에 따라, 일 실시예에서, 필드 명세(208)는 기본 데이터의 포맷을 반영하는 타입 속성을 포함한다. 그러나, 다른 실시예에서는, 필드 명세(208)의 데이터 포맷은 연관된 기본 물리 데이터와는 다르며, 이 경우에, 액세스 메소드는 요청하는 개체에 의해 취해진 적절한 포맷으로 데이터를 반환하는 것을 담당한다. 따라서, 액세스 메소드는 무슨 데이터 포맷을 취해야 하는가(즉, 논리 필드에 따라) 뿐만 아니라 기본 물리 데이터의 실제 포맷도 알아야 한다. 그러면, 액세스 메소드는 기본 물리 데이터를 논리 필드의 포맷으로 변환시킬 수 있다.

일례로서, 도 2C에 도시된 데이터 저장소 추상화 컴포넌트(148)의 필드 명세(208)는 관계형 데이터 표현(214₂)에서 표현된 데이터에 매핑된 논리 필드를 나타낸 것이다. 그러나, 데이터 저장소 추상화 컴포넌트(148)의 다른 예들에서는 논리 필드들을 XML 등의 다른 물리 데이터 표현에 매핑시킨다.

하나 이상의 필드 명세(208)는 또한 상관 관계 속성(도 1에서 상관 관계 속성(162)으로 집합적으로 도시됨)으로 구성될 수 있다. 예를 들어, Diagnosis 필드 명세(208₃)는 Age 상관 관계 속성(218₁) 및 Time 상관 관계 속성(218₂)을 포함하고, Test Result 필드 명세(208₆) 또한 Age 상관 관계 속성(218₁) 및 Time 상관 관계 속성(218₂)을 포함한다. 모든 필드 명세가 상관 관계 속성을 갖는 것은 아니라는 것에 주목할 필요가 있다.

일반적으로, 상관 관계 속성은 논리 필드가 상관될 수 있는 개체들/사건들을 식별한다. 일 실시예에서, 논리 필드들은 기간(특정 시점 또는 시간 범위), 물리적 장소 또는 연령에 상관될 수 있다. 따라서, 일 실시예에서, 상관 관계 속성은 논리 필드에 연관된 데이터의 인스턴스들(instances)을 제한시키는 데 사용될 수 있는 공간 또는 시간 차원을 식별한다. 그러나, 보다 일반적으로는, 쿼리 동작에 관련된 여러 개체들의 다른 특성들-예를 들어, 색상, 중량, 체적, 길이, 너비 등-을 고려한 상관 관계 속성들을 정의할 수 있다. 도 2B에서 보여지는 바와 같이, 각 필드 정의에 연관된 상관 관계 속성들은 일반적으로 "Name" 및 "Fieldref"를 포함한다. Name은 (상관 관계 속성이 정의된) 대응하는 필드가 상관될 수 있는 것에 따라 상관 관계 속성을 식별하는 핸들(handle)을 제공한다. 예를 들어, 상관 관계 속성(218₁)에는 대응하는 논리 필드가 연령에 대해 상관될 수 있는 것을 의미하는 이름 "Age"가 주어진다. 본 예제에서, Age 상관 관계 속성(218₁) 및 Time 상관 관계 속성(218₂)은 "Diagnosis" 논리 필드(210₃) 및 "Test Result" 논리 필드(210₆)에 대한 필드 명세의 일부이다. "Fieldref"는 특정 논리 필드에 대한 상관 관계 속성의 정의를 제공한다. 예를 들어, "Diagnosis" 논리 필드(210₃)의 Age 상관 관계 속성(218₁)에 대한 "Fieldref"는 "AgeAtDiagnosis"로 참조되는 정의를 지정한다. 본 실시예에서, "AgeAtDiagnosis"는 또 다른 논리 필드(210₅)로서, 이는 또한 Diagnostics 카테고리(216₂)의 일부이다. 보다 상세히 기술하자면, "AgeAtDiagnosis" 논리 필드(210₅)는 위에서 정의된 구성된 필드이다. 본 예제에서, 구성된 논리 필드(210₅)는 입력 매개변수(parameters)로서 Diagnosis Date 및 Birth Date를 수신하여 매개변수 값들 간의 차를 취한다. 이와 같이 하여, Age 상관 관계 속성은 임의 여러 정의를 취할 수 있다. 예를 들면, Test Result 논리 필드(210₆)에 대해 지정된 Age 상관 관계 속성(218₁)은 검사 시의 연령(Fieldref="AgeAtTest")에 따라 정의된다. "AgeAtTest" 논리 필드(210₈)는 입력으로서 Test Date 및 Birth Date를 수신하여 그 차를 취하는 구성된 필드로, 이에 의해 검사 시의 연령이 계산된다.

동작 중에, 상관 관계 표준에 의해 구성된 추상적인 쿼리(예를 들어, 표 I에 도시되고 도 2B의 추상적인 쿼리(202)에 의한 것)가 실행을 위해 발행되면, 상관 관계 알고리즘(151)은 상관 관계 표준이 적용되는 각 조건이 상관 관계 표준을 지원하

는 가의 여부를 판정하는 단계들을 취한다. 이는 추상적인 쿼리(202)에 의해 참조된 논리 필드들에 대해 필요한 메타데이터 및 액세스 메소드 정보를 정의하는 대응하는 데이터 저장소 추상화 컴포넌트(148) 내의 정보를 이용하여 행해진다. 상세히 기술하자면, 추상적인 쿼리의 지정된 논리 필드에 대해 어느 상관 관계 표준이 지원되는 가를 알아 내기 위해 데이터 저장소 추상화 컴포넌트(148)의 상관 관계 속성(162)을 조사한다. 상관 관계 표준을 지원하는 각각의 논리 필드마다, 추상적인 쿼리를 표현하도록 생성된 실행가능 쿼리문(query statement)에 추가 술어가 부가된다.

도 2B에 도시된 것에 대응하는 예시적 데이터 저장소 추상화 컴포넌트(148)는 표 II를 참조하여 예시될 수 있다. 일례로서, 데이터 저장소 추상화 컴포넌트는 XML을 이용하여 정의된다. 그러나, 임의 다른 언어를 유리하게 사용할 수 있다.

표 II - 데이터 저장소 추상화 예제

TABLE II - DATA REPOSITORY ABSTRACTION EXAMPLE

```

001 <DataRepository>
002   <Category name="Demographic">
003     <Field queryable="Yes" displayable="Yes" name="FirstName">
004       <AccessMethod>
005         <Simple columnName="f_name" tableName="patient" />
006       </AccessMethod>
007       <Type baseType="char"/>
008     </Field>
009     <Field queryable="Yes" displayable="Yes" name="LastName">
010       <AccessMethod>
011         <Simple columnName="l_name" tableName="patient" />
012       </AccessMethod>
013       <Type baseType="char"/>
014     </Field>
015     <Field queryable="Yes" displayable="Yes" name="BirthDate">
016       <AccessMethod>
017         <Simple columnName="b_date" tableName="patient" />
018       </AccessMethod>
019       <Type baseType="date"/>
020     </Field>
021   </Category>
022   <Category name="Diagnostics">
023     <Field queryable="Yes" displayable="Yes" name="Diagnosis">
024       <AccessMethod>
025         <Simple columnName="diag" tableName="patient" />
026       </AccessMethod>

```

```

027     <Correlation name="Age" fieldRef="AgeAtDiagnosis"/>
028     <Correlation name="Time" fieldRef="DiagnosisDate"/>
029     <Type baseType="char"/>
030   </Field>
031   <Field queryable="Yes" displayable="Yes" name="DiagnosisDate">
032     <AccessMethod>
033       <Simple columnName="diagdate" tableName="patient" />
034     </AccessMethod>
035     <Type baseType="date"/>
036   </Field>
037   <Field queryable="Yes" displayable="Yes" name="AgeAtDiagnosis">
038     <AccessMethod>
039       <Composed>
040         <Composition>YEAR(Field(DiagnosisDate) – Field(BirthDate))
041       </Composition>
042     </Composed>
043   </AccessMethod>
044   <Type baseType="int"/>
045 </Field>
046 </Category>
047 <Category name="Tests">
048   <Field queryable="Yes" displayable="Yes" name="Hemoglobin">
049     <AccessMethod>
050       <Simple columnName="testval" tableName="patient" />
051     </AccessMethod>
052     <Correlation name="Age" fieldRef="AgeAtTest"/>
053     <Correlation name="Time" fieldRef="TestDate"/>
054     <Type baseType="int"/>
055   </Field>
056   <Field queryable="Yes" displayable="Yes" name="TestDate">
057     <AccessMethod>
058       <Simple columnName="testdate" tableName="patient" />
059     </AccessMethod>
060     <Type baseType="date"/>
061   </Field>
062   <Field queryable="Yes" displayable="Yes" name="AgeAtTest">
063     <AccessMethod>
064       <Composed>
065         <Composition>YEAR(Field(TestDate) – Field(BirthDate))
066       </Composition>
067     </Composed>
068   </AccessMethod>
069   <Type baseType="int"/>
071 </Field>
072 </Category>
073 </DataRepository>

```

이제부터, 런타임 컴포넌트(150)에 의해 추상적인 쿼리들을 처리하는 일 실시예에 대해 도 3을 참조하면서 기술하기로 한다. 쿼리 처리 방법(300)은 추상적인 쿼리가 처리를 위해 런타임 컴포넌트(150)에 의해 수신되는 단계(302)에서 시작된다. 또한, 런타임 컴포넌트(150)는 추상적인 쿼리의 인스턴스를 판독하여 구문 분석(parse)하고, 개개의 선택 표준, 상관 관계 표준, 및 원하는 결과 필드의 위치를 찾아 낸다(locate). 단계(304)에서, 추상적인 쿼리의 각 상관 관계 표준마다 루프가 시작되어 수행된다. 단계(306)에서, 런타임 컴포넌트(150)는 표준이 범위 타입(span type) 표준, 즉 각 데이터 개체 간의 기간에 기초하여 데이터 개체들을 상관시키는 표준인지를 판단한다. 예를 들어, 도 2B에 도시된 추상적인 쿼리는 기간 상관 관계 속성이 적용되는 추상적인 쿼리 데이터 개체들이 서로 5 단위의 기간(예를 들어, 날) 내에 있어야 함을 지정하는 상관 관계 표준 "Span(Time)<5"을 포함한다. 만일 단계(306)에서 표준이 범위 타입 표준이면, 이 표준은 범위 표준 목록(170; 도 1 참조)에 추가된다(단계 308). 이 후에, 방법(300)은 단계(304)로 복귀하여 그 다음 상관 관계 표준에 대한 처리를 시작한다. 만일 단계(306)에서 표준이 범위 타입 표준이 아니라면, 이 표준은 개개의 표준 목록(172; 도 1 참조)에 추가된다(단계 310). 이 후에, 방법(300)은 단계(304)로 복귀하여 그 다음 상관 관계 표준에 대한 처리를 시작한다.

추상적인 쿼리에서의 상관 관계 표준들 각각이 처리되었으면, 방법(300)은 선택 표준을 갖는 각 조건마다 루프가 시작되는 단계(312)로 진행된다. 일 실시예의 경우, 조건은 필드명(논리 필드용), 비교 연산자(=, >, < 등), 및 값 표현식(비교되는 필드의 것)으로 이루어질 수 있다. 단계(314)에서, 런타임 컴포넌트(150)는 추상적인 쿼리의 조건으로부터의 필드명을 이용하여 데이터 저장소 추상화 컴포넌트(148)의 필드 정의를 찾아 낸다. 이 후에, 런타임 컴포넌트(150)는 처리되는 논리 필드에 대한 실행가능 쿼리 컨트리뷰션(Executable Query Contribution)을 작성한다(단계 316). 본원에서 정의된 바와 같이, 실행가능 쿼리 컨트리뷰션은 현재 논리 필드에 기초하여 데이터 선택을 수행하는 데 사용되는 실행가능 쿼리의 일부이다. 실행가능 쿼리는 SQL 및 XML Query 등의 언어로 표현되는 쿼리로서, 주어진 물리적 데이터 저장소(예컨대, 관계형

데이터베이스 또는 XML 저장소)의 데이터와 일치한다. 따라서, 실행가능 쿼리는 도 1에서 데이터베이스(156)로 나타낸, 물리적 데이터 저장소로부터 데이터를 찾아 내어 검색하는 데 사용된다. 이 후에, 현재 필드에 대해 생성된 실행가능 쿼리 컨트리뷰션이 실행가능 쿼리문(Executable Query Statement)에 추가된다(단계 318).

단계(320)에서, 런타임 컴포넌트(150)는 처리되어지는 조건의 필드가 범위 표준 목록(170) 중의 상관 관계 표준에 대응하는 상관 관계 속성(즉, 데이터 저장소 추상화 컴포넌트(148) 내)을 갖는 지를 판단한다. 만일 그렇다면, 필드는 범위 필드 목록(174)에 추가된다(단계 322). 일 실시예에서, 범위 필드 목록(174)은 각 타입의 범위 상관 관계 속성마다 제공된다(즉, Span(Time)용 목록, Span(Location)용 목록, 등). 그러나, 다른 실시예에서는, 이 단계에서 하나의 범위 필드 목록이 생성되고, 개개의 범위 상관 관계 표준을 처리할 때 이 목록으로부터 필드의 부분집합이 선택된다(프로세스는 이하에서 기술됨). 이 후에, 방법(300)은 단계(312)로 복귀하여 그 다음 선택 표준 조건을 처리하기 시작한다. 그러나, 단계(320)에서 "아니오(NO)이면, 프로세스는 단계(324)로 진행하여, 런타임 컴포넌트(150)가 처리 중인 필드가 개개의 표준 목록(172) 중의 한 상관 관계 표준에 대응하는 상관 관계 속성(즉, 데이터 저장소 추상화 컴포넌트(148) 내)을 갖는 지를 판단한다. 만일 단계(324)에서 "아니오"이면, 프로세스는 단계(312)로 복귀한다. 단계(324)에서 "예"인 경우에는, 데이터 저장소 추상화 컴포넌트(148)로부터 처리 중인 필드에 대한 상관 관계 속성의 정의를 검색한다(단계 326). 단계(328)에서, 런타임 컴포넌트(150)는 상관 관계 표준 및 필드 상관 관계 속성 정의로부터 쿼리 술어를 작성한다. 단계(330)에서 이 술어가 실행가능 쿼리에 추가된다. 그 후, 프로세스는 단계(312)로 복귀한다.

선택 표준 조건들 각각이 단계(312)에서 시작된 루프에 의해 처리되었으면, 방법(300)은 범위 필드 목록(174) 내의 필드 각 쌍마다(즉, 동일한 상관 관계 속성이 정의되는 필드들의 비순서 종속형 필드 순열) 루프가 시작된다. 단계(334)에서, 런타임 컴포넌트(150)는 필드 쌍의 각 필드에 대한 상관 관계 속성의 필드 정의를 (데이터 저장소 추상화 컴포넌트(148)로부터) 검색한다. 단계(336)에서, 런타임 컴포넌트(150)는 각 필드 속성 정의 간의 절대 차(예를 들어, ABS(DAYS(testdate - diagdate)))를 계산하는 표현식을 작성한다. 이 표현식이 범위 표준과 조합하여 필드 쌍에 관련되는 범위 쿼리 술어(예컨대, ABS(DAYS(testdate - diagdate)) > 5)를 작성한다. 이 후, 이 술어가 실행가능 쿼리에 추가되고(단계 340), 프로세스는 단계(332)로 복귀하여 범위 필드 목록(174) 내의 그 다음 필드 쌍을 처리한다. 일단 각각의 필드 쌍이 처리되었으면, 방법(300)이 종료되어, 쿼리는 실행 준비를 한다.

단계(316) 및 단계(328)에 따라 논리 필드에 대한 실행가능 쿼리 컨트리뷰션을 작성하는 방법(400)의 일 실시예에 대해서도 4를 참조하여 기술하기로 한다. 단계(402)에서, 방법(400)은 현재 논리 필드에 연관된 액세스 메소드가 단순 액세스 메소드인지를 질의한다. 만일 그렇다면, 물리 데이터 위치 정보에 기초하여 실행가능 쿼리 컨트리뷰션이 작성되고(단계 404), 그 후에 프로세스는 상술된 방법(300)에 따라 계속된다. 그렇지 않은 경우에는, 프로세스는 단계(406)로 진행하여, 현재 논리 필드에 연관된 액세스 메소드가 필터된 액세스 메소드인지를 질의한다. 만일 그렇다면, 임의 물리적인 데이터 개체에 대한 물리 데이터 위치 정보에 기초하여 실행가능 쿼리 컨트리뷰션이 작성된다(단계 408). 단계(410)에서, 실행가능 쿼리 컨트리뷰션은 물리적 데이터 개체에 연관된 부분 집합 데이터를 사용하는 추가 논리(필터 선택)에 의해 확장된다. 그 후에, 프로세스는 상술된 방법(300)에 따라 계속된다.

액세스 메소드가 필터된 액세스 메소드가 아닌 경우, 프로세스는 단계(406)에서 단계(412)로 진행하여, 방법(400)이 액세스 메소드가 구성된 액세스 메소드인지를 질의한다. 액세스 메소드가 구성된 액세스 메소드이면, 구성된 필드 표현식의 각각의 서브 필드 참조에 대한 물리적 데이터 위치를 찾아 내어 검색한다(단계 414). 단계(416)에서, 구성된 필드 표현식의 논리적 필드 참조 대신, 구성된 필드 표현식의 물리적 필드 위치 정보로 대체되어, 이에 의해 실행가능 쿼리 컨트리뷰션이 생성된다. 그 후에, 프로세스는 상술된 방법(300)에 따라 계속된다.

액세스 메소드가 구성된 액세스 메소드가 아닌 경우, 프로세스는 단계(412)에서 단계(418)로 진행한다. 단계(418)는 본 발명의 실시예들로서 고려되는 임의 다른 액세스 메소드 타입을 나타낸다. 그러나, 이용가능한 모든 액세스 메소드가 구현되는 실시예들도 고려되는 것에 주목해야 한다. 예를 들어, 특정 실시예에서는, 단지 단순 액세스 메소드만이 사용된다. 다른 실시예에서는, 단지 단순 액세스 메소드 및 필터된 액세스 메소드만이 사용된다.

상술된 바와 같이, 쿼리 조건에서 참조되는 논리 필드는 상관 관계 속성과, 쿼리 조건 필드에 대해 주어진 상관 관계 속성을 해석하는 방법을 정의하는 다른 논리 필드를 식별하는 "FieldRef"를 정의할 것이다. 도 4의 방법(400)은 "참조된(referenced)" 논리 필드 및 상관 관계 속성에 연관된 논리를 이용하여 실행된다. 예를 들어, 상관 관계 표준이 Age > 50 이고 Test 필드가 AgeTest의 FieldRef에 의해 Age에 대한 정의를 포함하면, 방법(400)은 조건 AgeAtTest > 50에 기반한 쿼리 술어를 작성한다.

상술된 바와 같이, 논리 필드가 기본 물리 데이터와는 다른 데이터 포맷을 지정하면 데이터 변환을 행할 필요가 있을 수 있다. 일 실시예에서는, 방법(400)에 따라 논리 필드에 대한 실행가능 쿼리 컨트리뷰션을 작성할 때, 각각의 해당 액세스 메

소드에 대해 초기 변환이 수행된다. 예를 들어, 이 변환은 단계(404, 408, 및 416)의 일부로서, 또는 이들 단계에 바로 후속하여 수행될 수 있다. 단계(322)에서 쿼리가 실행된 후에, 물리 데이터 포맷에서 논리 필드 포맷으로의 후속 변환이 수행된다. 물론, 논리 필드 정의 포맷이 기본 물리 데이터와 동일하면, 변환은 필요치 않다.

상술된 예제들은 의료 분야에 관한 것으로, 상관 관계 속성들은 기간, 장소, 및 연령에 관해 예시하였다. 그러나, 본 발명은 이것으로만 제한되는 것이 아니라, 데이터에 대한 임의의 데이터베이스 환경 및 타입까지 확장된다. 예를 들어, 상술된 바와 같이, 쿼리 동작에 관련되는 여러 개체들의 다른 특성들, 예를 들어, 색상, 중량, 체적, 길이, 너비 등을 고려하는 상관 관계 속성들이 정의될 수 있다. 대응하는 데이터 추상화 컴포넌트가 정의될 수 있는 다음의 추상적인 쿼리(표 III)를 고려해 보기로 한다.

표 III - 추상적인 쿼리 예제

TABLE III - ABSTRACT QUERY EXAMPLE

```

001 <QueryAbstraction>
002   <Selection>
003     <Condition field="CarpetType" operator="EQ" value="berber"/>
004     <Condition field="WallpaperPattern" operator="EQ" value="floral"
005     relOperator="AND" />
006     <Condition field="PaintFinish" operator="EQ" value="satin"
007     relOperator="AND" />
008     <Correlation attribute="Span(Color)" operator="LT" value="10"/>
009   </Selection>
010   <Results>
011     <Field name="CarpetNumber"/>
012     <Field name="WallpaperNumber"/>
013     <Field name="PaintNumber"/>
014   </Results>
015 </QueryAbstraction>
    
```

이 추상적인 쿼리는 한 세트의 카펫, 벽지, 및 페인트 조합을 카펫 유형, 벽지 패턴, 및 페인트 마무리 유형에 대해 지정된 표준에 기초하여 선택하도록 설계된다. 여기서, 각각의 개체(카펫, 벽지, 페인트)가 "색상"의 상관 관계 속성을 정의한 것으로 하면, 상관 관계 조건인 Span(Color) < 10은 조합들을 동일 색상인 것으로만 선택하도록 제한시킬 것이다. Span(Color)에 대한 가능한 정의는 적색(Red), 녹색(Green), 및 청색(Blue) 성분에서 서로에 대해 "n" 단위 내에 있는 색상에 대한 RGB 값들일 것이다.

따라서, 데이터 저장소 추상화 컴포넌트(148)는 여러 이점들을 제공한다. 일 양상으로서, 응용 프로그램 쿼리 명세와 기본 데이터 표현 간에서의 느슨한 결합(loose coupling)을 정의함으로써 이점들이 얻어진다. SQL을 사용하는 경우에서와 같이, 응용 프로그램을 특정 테이블, 컬럼 및 관계 정보에 의해 부호화하기 보다는, 응용 프로그램은 나중에 실행 시에 특정의 물리 데이터 표현에 바인딩되는 데이터 쿼리 요건들을 보다 추상화된 방식으로 정의한다. 본 발명의 느슨한 쿼리-데이터 결합은 요청하는 개체들(예를 들어, 응용 프로그램들)이, 기본 데이터 표현이 수정되거나 또는 요청하는 개체가 그 요청하는 개체가 개발되었을 때 사용된 것보다 완전히 새로운 물리 데이터 표현에 대해 사용해야 하는 경우라도 기능을 행할 수 있도록 해 준다. 주어진 물리 데이터 표현이 수정되거나 재구성된 경우, 대응하는 데이터 저장소 추상화 컴포넌트는 기본 물리 데이터 모델에 대해 행해진 변경들을 반영하도록 갱신된다. 동일한 논리 필드 집합은 쿼리들에 의해 사용하도록 이용될 수 있으며, 단지 물리 데이터 모델 내의 상이한 개체들 또는 위치들에 바인딩되어 있다. 그 결과, 대응하는 물리 데이터 모델이 상당히 변경되었더라도, 추상적인 쿼리 인터페이스로 작성된 요청하는 개체들은 기능을 변경시키지 않고 속행된다. 요청하는 개체가 그 요청하는 개체가 개발되었을 때 사용된 것보다 완전히 새로운 물리 데이터 표현에 대해 사용해야 하는 경우에는, 새로운 물리 데이터 모델은 동일한 기술(예를 들어, 관계형 데이터베이스)을 사용지만 정보를 명명하고 조직화함에 있어 다른 전략(예를 들어, 다른 스키마)에 따라 구현될 수 있다. 새로운 스키마는 단순하고, 필터되고, 구성된 필드 액세스 메소드 기술들을 사용하여 응용 프로그램이 필요로 하는 논리 필드 집합에 매핑될 수 있는 정보를 포함할 것이다. 이와는 다르게, 새로운 물리 표현은 유사한 정보를 표현하기 위해 다른 기술을 이용(예를 들어, 관계형 데이터베이스 시스템에 대한 XML 기반 데이터 저장소의 이용)할 수 있다. 이 두 경우 모두, 추상적인 쿼리 인터페이스를 이용하도록 작성된 기존의 요청 개체들은, 새로운 물리 데이터 모델의 위치 및 물리 표현으로 쿼리에서 참조된 필드들을 매핑시키는 다른 데이터 저장소 추상화의 제공에 의해 새로운 물리 데이터 표현을 사용하도록 쉽사리 이행(migrate)될 수 있다.

또한, 추상적인 쿼리 및 기본 데이터 추상화 모델과 함께 전역(global) 상관 관계 속성들의 사용은, 고수준의 데이터 상관 관계 분석을 필요로 하는 복잡한 데이터 쿼리 논리를 기술하는 작업을 상당히 단순화시킨다. 더욱이, 상관 관계 논리와 기본 데이터 선택 논리를 분리시킴에 의해, 동일하지만 상관 관계 조건이 다른 기본 쿼리를 신속하게 재사용하는 것이 가능해진다.

최종 사용자에게 대해서는, 데이터 저장소 추상화는 상관 관계를 갖는 데이터 쿼리들의 표현에 연관된 많은 복잡성을 제거시킨다. 상세히 기술하자면, 최종 사용자는 쿼리 선택 조건들이 주어진 상관 관계 접근법을 지원하는 것에 대해 더 이상 걱정할 필요가 없으며, 또한 쿼리에 관련된 각각의 새로운 개체에 대한 사건-기반 및/또는 개체-기반(예컨대, 시간-기반, 위치-지향, 등) 논리를 표현하는 방법을 이해해야 부담을 더 이상 갖지 않는다.

비록 데이터 저장소 추상화의 사용으로 상기 및 다른 이점들을 제공하지만, 당 기술 분야의 숙련자들은 상관 관계 속성들의 목적 및 함수를 데이터 저장소 추상화 컴포넌트(148)와 독립적으로 구현될 수 있다는 것을 쉽사리 인식할 것이다. 예를 들면, 실시예들은 SQL-기반 쿼리 구현에 의해 사용되는 테이블 및 컬럼 상의 정보를 보충하도록 추가의 메타데이터(metadata)를 제공한다. 이 경우에, 주어진 컬럼에 대한 메타데이터는 Age 등의 상관 관계 속성, 및 컬럼에 나타난 개체에 대한 연령을 계산하는 방법을 식별하는 연관된 SQL 쿼리 표현식을 식별할 것이다. 이후, SQL 쿼리 프리-프로세서는 입력으로서 초기 SQL 쿼리 및 하나 이상의 상관 관계 조건들을 취할 수 있으며, SQL 쿼리를 제공된 상관 관계 조건들을 설명하는 추가 술어에 의해 확장시킬 것이다. 이는 상관 관계 조건 집합으로부터 상관 관계 속성(들)을 지원하는 컬럼에 대한 참조를 조사하고, 특정 상관 관계 속성의 주어진 컬럼의 정의에 대해 정의된 쿼리 표현식에 기반한 추가의 SQL WHERE 절(clause) 술어를 생성함으로써 행해질 수 있다.

본원에서 참조한 특정 값, 정의, 프로그래밍 언어, 및 예제들은 단지 예시를 목적으로 한 것임에 주목해야 한다. 따라서, 본 발명은 임의의 특정 예시 및 예제로만 한정되는 것은 아니다. 또한, 비록 본 발명의 양상을 SELECTION 동작에 대해 기술하였지만, ADD, MODIFY, INSERT, DELETE 등의 잘 알려진 동작들을 포함하여, 다른 입력/출력 동작이 고려된다. 물론, 소정의 액세스 메소드들은 특정 액세스 메소드를 이용하는 필드들을 사용하여 정의될 수 있는 추상적인 쿼리 함수 유형에만 제한될 수 있다. 예를 들면, 구성된 액세스 메소드를 필요로 하는 필드들은 MODIFY, INSERT, 및 DELETE를 실행할 수 있는 대상은 아니다.

비록 상기에서는 본 발명의 실시예에 대해 기술하였지만, 본 발명의 기본 범주 및 사상을 벗어나지 않는 한, 본 발명의 또 다른 실시예의 구현이 가능하며, 본 발명은 첨부된 청구범위에 의해서만 한정되는 것임에 주목할 필요가 있다.

산업상 이용 가능성

본 발명에 따르면, 데이터 액세스는 상호 관계 표준 또는 표준들로 구성된 쿼리에 의해 용이해지며, 상관 관계 표준을 제공함으로써 쿼리의 조건들에 적용될 추가 상관 관계 논리의 도입이 용이해져, 최종 사용자는 쉽사리 원하는 데이터를 액세스할 수 있게 된다.

도면의 간단한 설명

본 발명의 상술된 특징들이 얻어지는 방식을 상세히 이해할 수 있도록 하기 위해, 상기에서 간략히 요약된 본 발명을 첨부된 도면들에 예시된 실시예들을 참조하여 기술하기로 한다.

그러나, 첨부된 도면들은 단지 본 발명의 전형 실시예를 예시한 것으로, 본 발명의 범주나 다른 등가 실시예들을 한정하려는 의도는 아닌 것으로 이해해야 한다.

도 1은 컴퓨터 시스템의 일 실시예를 도시.

도 2A는 본 발명의 일 실시예의 소프트웨어 컴포넌트들의 논리적/물리적 뷰를 도시.

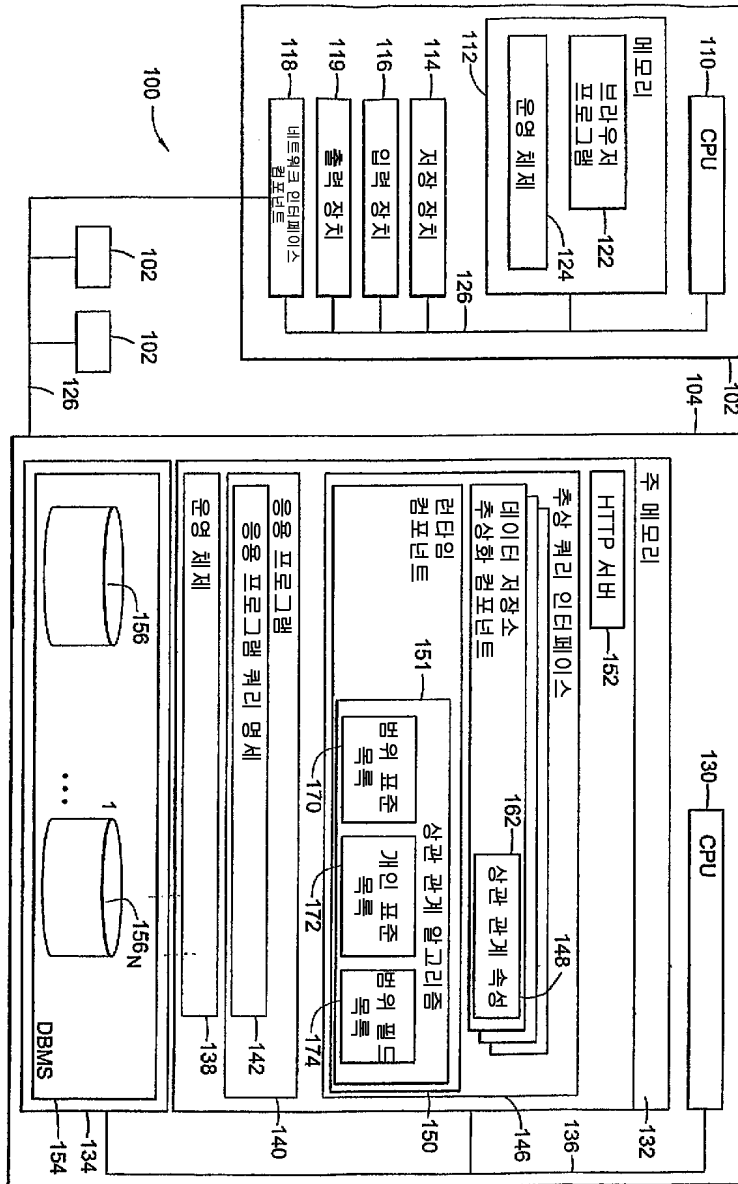
도 2B 및 도 2C는 추상적인 쿼리 및 데이터 저장소 추상화의 논리적 뷰를 도시.

도 3A 내지 도 3C는 런타임 컴포넌트의 동작을 예시하는 흐름도.

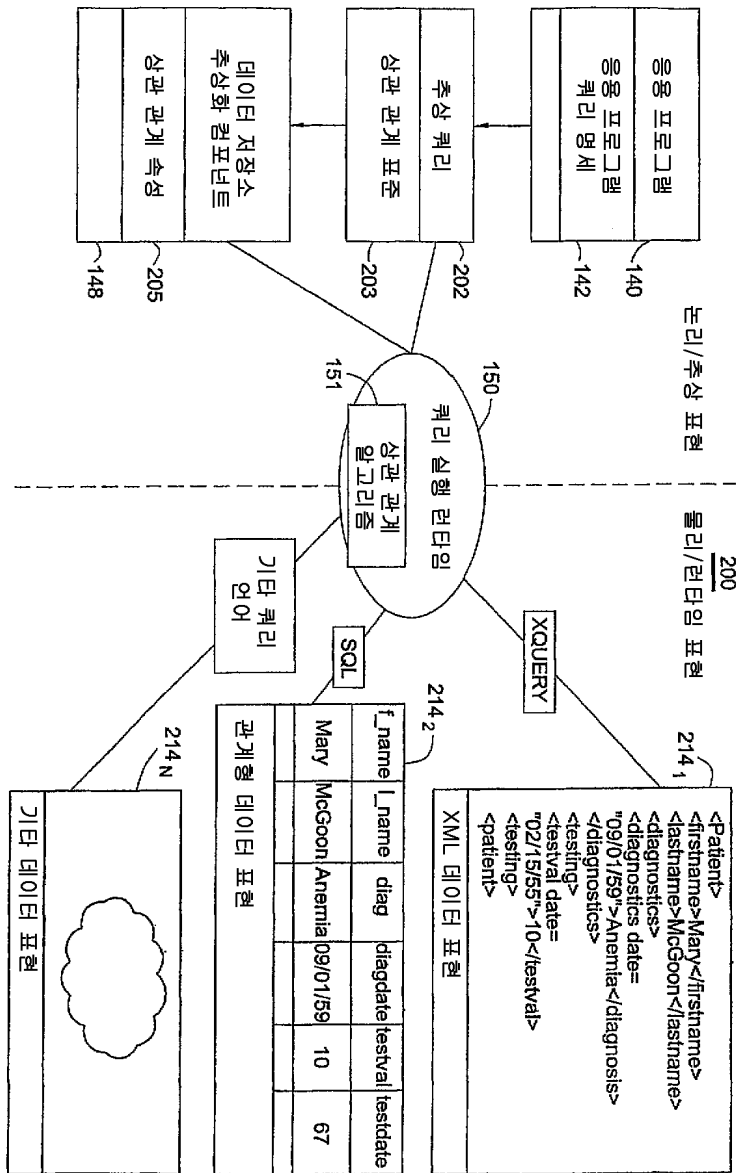
도 4는 런타임 컴포넌트의 동작을 예시하는 흐름도.

도면

도면1



도면2a



도면2b

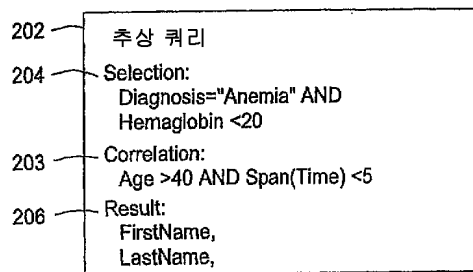
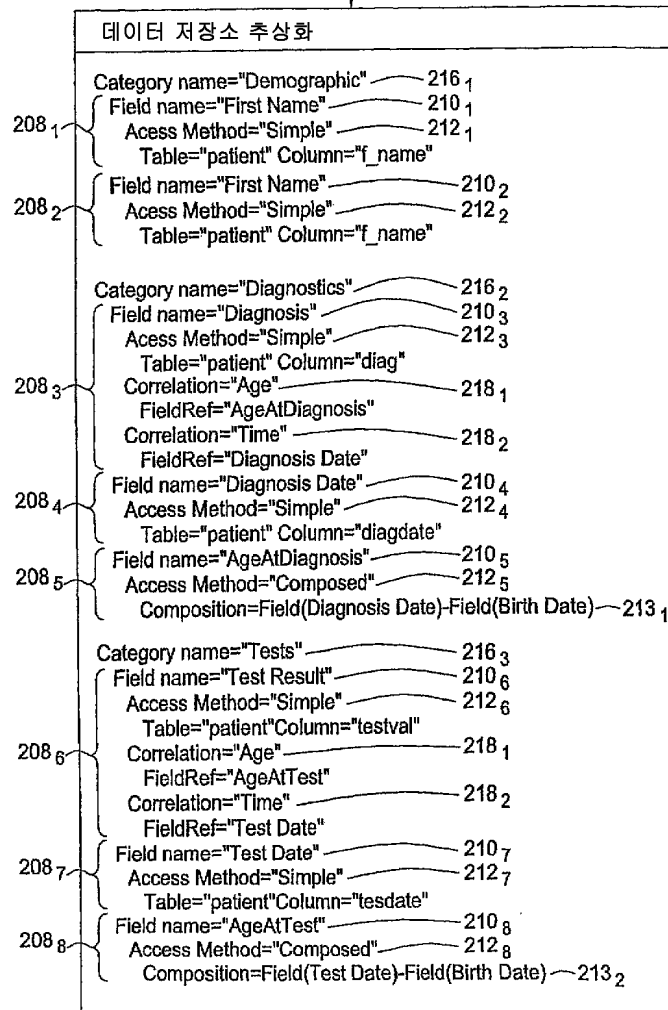


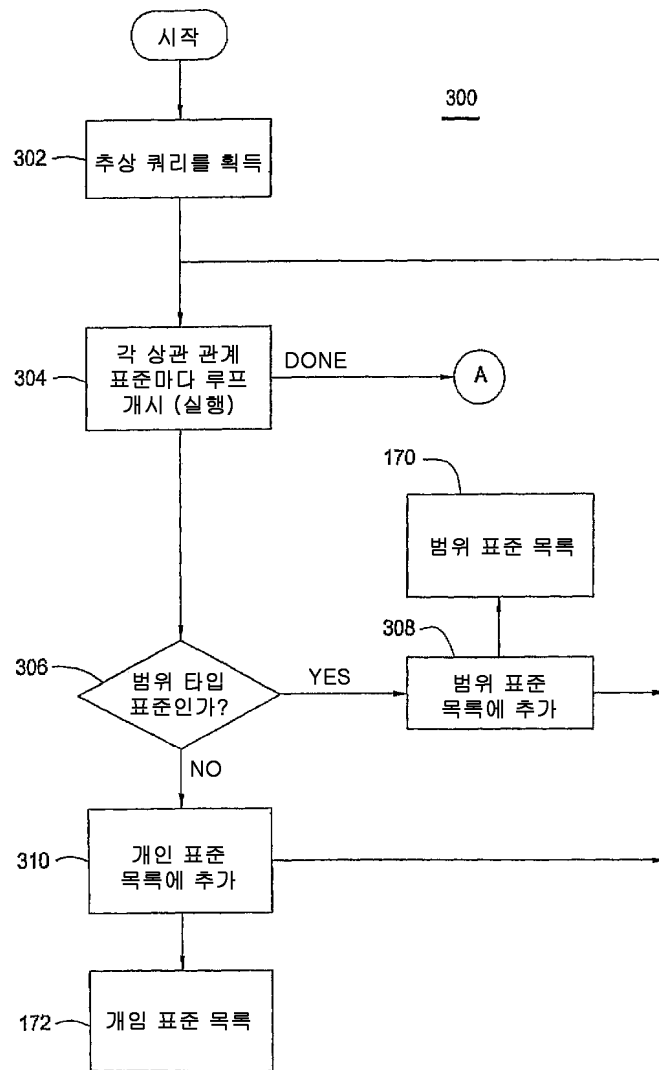
FIG. 2C

도면2c

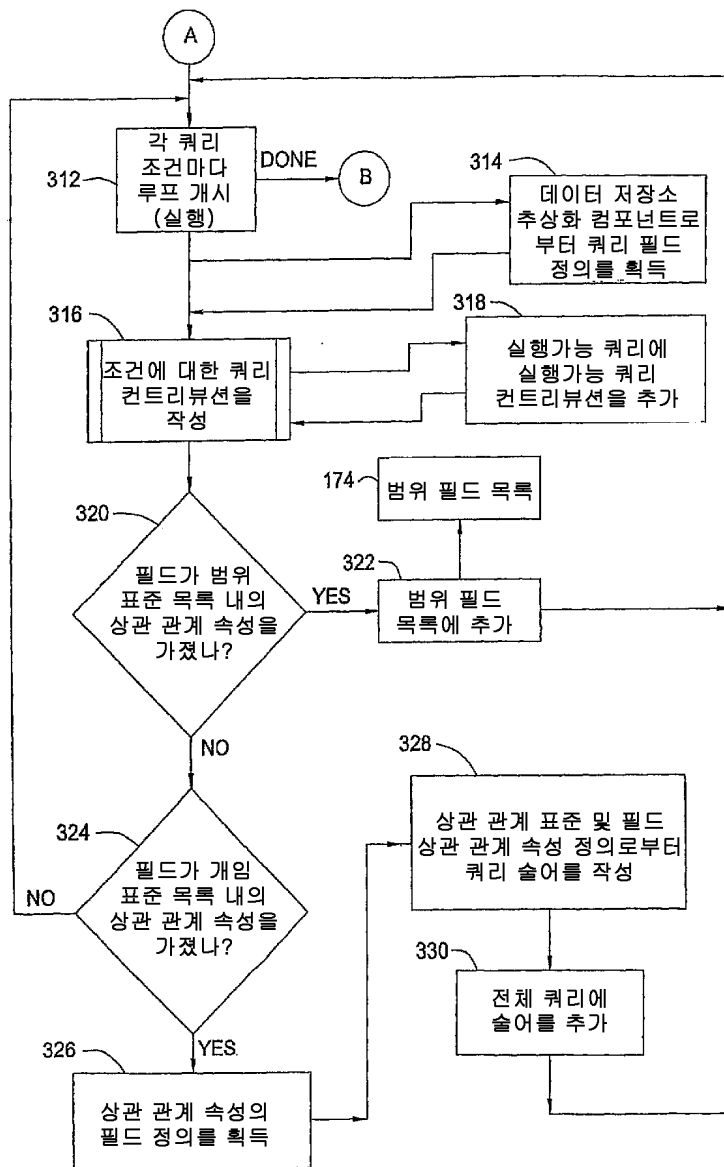
FIG. 2B



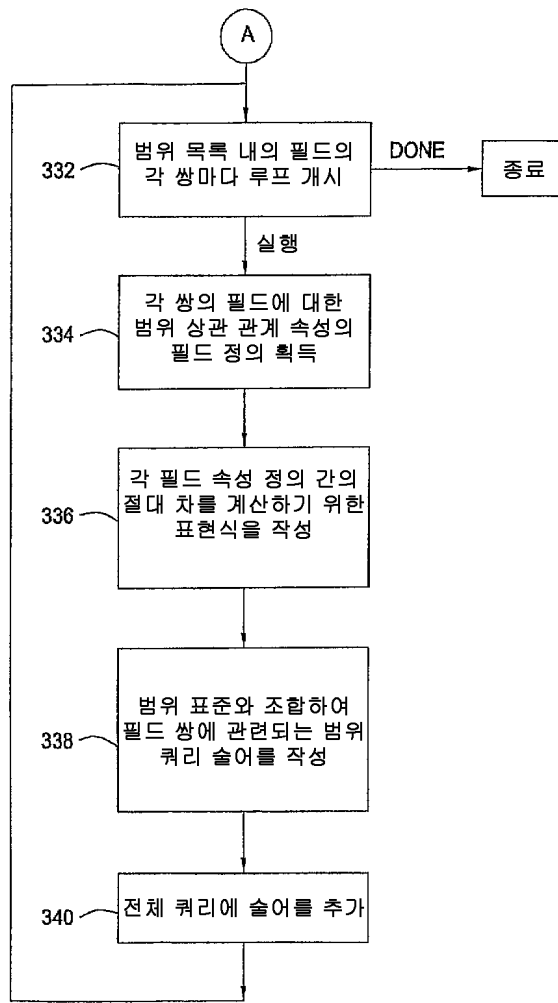
도면3a



도면3b



도면3c



도면4

