

(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.⁷
G06F 17/40
G06F 12/00

(11) 공개번호 10-2005-0061597
(43) 공개일자 2005년06월22일

(21) 출원번호 10-2005-7007700
(22) 출원일자 2005년04월29일
 번역문 제출일자 2005년04월29일
(86) 국제출원번호 PCT/US2003/032187 (87) 국제공개번호 WO 2004/042581
 국제출원일자 2003년10월10일 국제공개일자 2004년05월21일

(30) 우선권주장 10/287,555 2002년11월01일 미국(US)

(71) 출원인 타임 인터스트리얼, 인크.
 캐나다 에이비 티5제이 1이3 에드몬톤 10259-105번 스트리트

(72) 발명자 클레인 랄프 빌헬름
 캐나다 알버타 티8에이 5케이7 셔우드 파크 7 코티네이 드라이브

(74) 대리인 김진환
 김두규

심사청구 : 없음

(54) 버저닝된 데이터베이스에 대한 리포트를 생성하기 위한시스템 및 방법

명세서

기술분야

본 발명은 일반적으로 데이터베이스에서의 버저닝된(versioned) 데이터 관리에 관한 것이다. 더 상세하게는, 본 발명은 데이터베이스에 데이터 객체의 복수의 버전들을 저장하고, 임의의 부여된 시간에 대응하는 데이터 객체의 하나의 버전을 디스플레이하는 리포트를 생성하기 위한 시스템 및 방법에 관한 것이다.

배경기술

인터넷 및 정보 기술(IT)의 출현은 비즈니스가 세계적으로 처리되는 방식을 혁신하였다. 비즈니스 조직은 이 기술들을 채택하여 그들의 구조, 업무 흐름, 비즈니스 관계를 조직하고 그들의 비즈니스 프로세스를 더 효율적으로 만들었다. 특히, 이 기술들은 비즈니스 조직들이 하루에 처리하는 방대한 양의 데이터를 관리할 수 있도록 도움을 주었다.

주어진 비즈니스 조직에 의해 생성되는 데이터는 비즈니스 조직의 정보 시스템의 다양한 데이터베이스에 통상적으로 저장된다. 데이터베이스는 콘텐츠가 쉽게 접근, 관리, 및 갱신될 수 있도록 조직된 데이터의 집합이다. 데이터베이스에 저장된 데이터 및 데이터를 조작하는데 사용된 알고리즘은 컴퓨터 메모리에서 큐, 연결 목록, 스택, 힙(heap), 트리, 또는 해쉬 테이블 등과 같은 데이터 구조에 의해 표현된다.

가장 유력한 데이터베이스의 형태는 관계형 데이터베이스로서, 형식적으로 기술된 테이블의 세트로서 조직되고, 이 테이블로부터 데이터는 테이블들을 인식하지 않고도 많은 상이한 방법들로 접근 또는 재구성된다. 또한, 관계로서 참조되는 각 테이블은 칼럼(column) 내에 하나 이상의 데이터 카테고리들을 포함한다. 테이블 내의 각 로우(row)는 칼럼에 의해 정의된 카테고리에 의하여 표현된 데이터 객체에 대한 유일한 데이터 레코드 또는 데이터의 인스턴스를 포함한다. 테이블 내의 각 로우는 일차 키(primary key)에 의해 유일하게 식별되고, 일차 키는 유일하게 로우를 식별하는 테이블 내의 어떤 칼럼 또는 칼럼들일 수 있다. 통상적으로, 테이블은 디폴트 일차 키로 행동하기 위하여 유일한 정수값을 갖는 칼럼을 포함할 것이다. 정수값은 데이터베이스에 의해 자동적으로 할당되고, 대개 증분하여 생성될 수 있다. 하나 이상의 칼럼으로 구성된 일차 키는 보통 복합 일차 키로서 참조된다.

예컨대, 캘리포니아 시애틀의 Amazon.com과 같은 전자 상거래 웹 사이트는 고객의 주문을 기술하는 관계형 데이터베이스를 가질 수 있다. 관계형 데이터베이스는 고객의 이름, 주소, 및 신용 카드를 칼럼으로 고객의 개인 정보를 저장하는 테이블, 및 구매 상품, 가격 및 수량을 칼럼으로 주문 자체를 기술하는 또 다른 테이블을 포함할 수 있다. 고객 및 고객의 주문

은 데이터 객체이고 테이블 내의 각 로우는 지정된 객체에 대한 데이터 레코드 또는 데이터의 인스턴스를 포함할 수 있다. 테이블은 각 데이터 레코드를 정수값으로 유일하게 식별하고 테이블의 일차 키로서 제공되는 ID 칼럼을 포함할 수 있다. 또는, 지정된 고객의 성과 이름은 고객의 개인 정보 테이블 내에 복합 일차 키로서 사용될 수 있다.

데이터는 구조화 쿼리(query) 언어(SQL)로 불리는 표준 응용 프로그램 인터페이스를 사용하여 데이터베이스로부터 조회될 수 있다. SQL은 사용자가 다른 데이터 동작들 중에서 데이터의 위치를 선택, 삽입, 삭제, 갱신 및 검색할 수 있게 한다. 사용자는 관계형 데이터베이스를 생성, 갱신 및 관리하기 위한 프로그램인 관계형 데이터베이스 관리 시스템(RDBMS)의 일부로서 데이터베이스 내의 데이터를 조작하는 SQL문을 특정할 수 있다. 상용가능한 RDBMS의 예는 뉴욕 화이트 플레인(White Plains)의 IBM사가 판매하는 DB2, 캘리포니아 레드우드 쇼어즈(Redwood Shores)의 Oracle사가 판매하는 Oracle 9i 데이터베이스, 뉴욕 아이슬란드(Islandia)의 Computer Associates International사가 판매하는 Openings를 포함한다. 또는, 비즈니스 조직들은 RDBMS를 모방하는 워싱턴 레드몬드(Redmond)의 Microsoft사가 판매하는 Excel과 같은 스프레드시트 소프트웨어 애플리케이션을 사용할 수 있다.

비즈니스 조직들은 종종 상용가능한 RDBMS를 사용하여 고객, 공급자, 계정 및 회계 정보와 같은 내부 데이터, 종업원 레코드, 발명자, 및 법적 레코드 등을 저장하는 데이터베이스를 관리한다. 또는 건설업 및 제조업과 같은 산업의 비즈니스 조직들은 다양한 비즈니스 활동들을 포함하는 활동 기준 비용 데이터 및 각 활동들에 대한 비용의 판단 및 비용 추진을 관리하기 위해 더 전문화된 RDBMS를 필요로 할 수 있다. 활동 기준 비용 데이터의 예는 건축 유닛의 원료 및 노동 비용과 지정된 구매 주문 또는 기계의 사용에 의해 발생된 비용을 포함한다. 상용가능한 활동 기준 비용 RDBMS는 캘리포니아 폴섬(Folsom)의 Meridian Project Systems사가 판매하는 Prolog Manager 및 Prolog Scheduler와 콜로라도 덴버의 J.D. Edwards & Company사가 판매하는 OneWorld 소프트웨어 패키지를 포함한다.

이러한 활동 기준 비용 시스템의 주요 기능은 진행과정에 맞춰 실제 비용과 예산된 비용 사이의 편차(variance)를 강조하는 것이다. 실제 비용의 정확하고 일치된 계산을 보증하기 위하여, 이 시스템은 다수의 변수를 고려할 필요가 있는 데이터베이스 쿼리를 실행한다. 이 변수들은 자원 식별 및 분류, 프로젝트, 상대 및 업무 참조, 비용 코드 참조, 및 날짜 및 시간 등을 포함할 수 있다. 이 시스템에서 실제의 비용을 계산하는 트랜잭션은 정확한 결과를 생성하기 위하여 이 변수들 각각을 정확하게 기록해야 한다.

이것은 RDBMS가 비즈니스 조직들이 생성하는 모든 데이터 레코드 및 트랜잭션의 히스토리 레코드를 유지하도록 요구한다. 데이터베이스에서 히스토리를 유지하기 위한 보통의 방법은 데이터베이스 내의 그 객체에 대한 새로운 버전으로 데이터 객체 모두를 갱신함으로써 데이터 객체의 유효 날짜를 추적하는 것이다. 각 버전은 하나의 데이터 레코드 또는 데이터베이스의 로우에 대응하고 데이터 객체의 특정 표현이 제공된 시점을 표시하는 하나 이상의 날짜 필드를 포함한다. SQL 쿼리는 데이터베이스 내의 날짜 필드에 대하여 날짜 범위를 사용함으로써 어떤 시점에 제공된 로우 또는 버전의 세트를 리턴하는데 사용될 수 있다.

예컨대, 비즈니스 조직은 종업원의 이름 및 주소에 관한 종업원의 개인 정보를 칼럼으로 종업원의 개인 정보를 저장하는 개인 정보 테이블, 및 비즈니스 명칭, 기술(skill), 지정된 날짜에 일했던 시간 및 그들이 속하는 비즈니스 유닛을 칼럼으로 비즈니스 조직 내의 종업원의 역할을 기술하는 또 다른 워크 테이블을 가지고 종업원에 관한 정보를 저장하는 노동 데이터베이스를 가질 수 있다. 종업원의 업무경과를 추적하기 위하여, 비즈니스 조직은 종업원의 업무 히스토리를 비즈니스 조직으로 저장하는 것을 필요로 할 것이다. 각 버전, 또는 지정된 날짜 객체 또는 종업원에 대한 워크 테이블 내의 로우는 "시작 날짜" 및 "완료 날짜" 필드를 포함하여 종업원이 특정 비즈니스 유닛을 위하여 일하고 있었던 동안의 시간 간격을 표시할 수 있다. 또는 각 버전은 하나의 날짜 필드로 표현되어 그 버전이 날짜 필드에 표시된 날짜부터 유효함을 표시할 수 있다. 다른 경우에, 비즈니스 조직에서 종업원의 업무 생활의 히스토리를 레코드를 유지하는 것은 비즈니스 조직 내에서 종업원의 역할이 변경될 때마다 새로운 버전 또는 로우를 워크 테이블에 추가하는 것의 문제이다. SQL 쿼리는 종업원이 지정된 시간 간격 동안 어느 비즈니스 유닛에서 일하고 있었는지를 찾는 데 사용될 수도 있다.

보다 복잡한 상황에서, 비즈니스 조직은 동일한 테이블 또는 다른 테이블에 저장된 다른 데이터 객체를 참조하는 하나 이상의 외래 키(foreign key)를 각 데이터 객체에 대하여 포함해야 할 것이다. 외래 키는 일부 다른 테이블의 일차 키 칼럼에서 발견되는 값을 갖는 어떤 테이블 내의 하나 이상의 칼럼의 세트이다. 상기 예에서, 워크 테이블은 테이블의 일차 키로 제공되는 정수값을 가지는 ID 칼럼 및 개인 정보 테이블에 외래 키로 제공되는 이름 칼럼을 포함할 수 있다.

그러나, 데이터 객체의 복수의 버전들이 테이블에 저장되는 경우에, 외래 키가 동일한 데이터 객체의 하나 이상의 버전을 도출할 수 있기 때문에 하나의 외래 키를 사용하여 다른 객체를 참조하는 것은 어려울 수 있다. 예컨대, 이름 칼럼은 비즈니스 조직을 위하여 일하는 동안 종업원이 갖는 다른 주소에 대응하는 각 버전으로서 개인 정보 테이블에서 지정된 종업원의 복수의 버전을 참조할 수 있다.

현재, 버저닝된 데이터 객체의 하나의 버전을 도출하는 하나의 외래 키를 사용할 수 있는 RDBMS는 없다. 현재의 RDBMS를 사용하여 버저닝된 데이터 객체의 하나의 버전을 추출하는 것은 원하는 버전을 참조하는 날짜 필드를 포함하는 복합 외래 키, 또는 특정 데이터베이스 설계에 주문화된(customised) 비전통적인 외래 키의 사용을 필요로 할 것이다. 상기 예에서, 지정된 시점에서 종업원의 주소를 찾는 것은 원하는 시간에 대응하는 날짜 외래 키의 사용뿐만 아니라 종업원의 이름을 포함하는 복합 외래 키의 사용을 필요로 할 것이다. 그러나, 버저닝된 데이터 객체에 접근하기 위해 복합 외래 키를 사용하는 것은 유지가 어렵고 어렵다. 데이터 레코드가 자주 변하는 것은 외래 키 자체를 자주 변경하는 것을 필요로 한다.

또한, 팔로 알토(Palo Alto)의 Crystal Decisions사가 제공하는 Crystal Reports, 캘리포니아 레드우드 쇼어즈의 Oracle사가 제공하는 Oracle Report, 캐나다 오타와의 Cognos사가 제공하는 Cognos와 같은, 현재 사용가능한 데이터베이스 리포팅 툴(reporting tool)은 지정된 데이터 객체의 하나의 버전을 추출하기 위하여 비전통적인 외래 키를 필요로 하는 버저닝된 데이터 객체에 대한 데이터베이스 리포트를 생성할 수 없다. 데이터베이스 리포팅 툴은 데이터베이스의 데이터 객체의 문자식, 도표식, 멀티미디어식 리포트 또는 디스플레이를 만드는데 종종 사용된다. 이 리포팅 툴은 비즈니스 조직이 데이터 객체를 심리스(seamless)로 볼 수 있게 하지만, 데이터 객체가 비전통적인 방식으로 저장되고 하나의 외래 키를 사용하여 다른 데이터 객체를 참조하는 경우에, 리포팅 툴은 지정된 시간에 버저닝된 데이터 객체의 쿼리를 생성할 수 없다.

진술한 사항을 고려하여, 본 발명의 목적은 데이터베이스에 데이터 객체의 복수의 버전을 효율적으로 저장하기 위한 시스템 및 방법을 제공하는 것이다.

본 발명의 또 다른 목적은 버저닝된 데이터 객체의 하나의 버전을 도출하는 하나의 외래 키를 사용하기 위한 시스템 및 방법을 제공하는 것이다.

또한, 본 발명의 목적은 지정된 시간에 대응하는 데이터 객체의 하나의 버전을 디스플레이하는 데이터베이스 리포트를 생성하기 위한 시스템 및 방법을 제공하는 것이다.

발명의 상세한 설명

데이터베이스에서 데이터 객체의 복수의 버전들을 효율적으로 저장하고 지정된 시점에 대하여 데이터베이스 리포트를 생성하기 위하여 상기 데이터 객체의 하나의 버전을 검색함으로써 본 발명의 목적들은 본 발명의 원리에 따라 수행된다. 데이터 객체는 테이블에 저장되는 엔티티를 표현하고 서로 다른 필드들 또는 카테고리들과 관련된다. 예컨대, 데이터 객체는 이름 필드 및 종업원 번호 필드와 관련된 종업원을 표현할 수 있다.

테이블에서 하나의 데이터 레코드 또는 로우에 대응하는 데이터 객체의 각 버전은 데이터 레코드가 테이블에 입력된 날짜 및 시간, 즉 데이터 레코드가 그 때부터 유효하다고 신뢰되는 날짜 및 시간을 표시하는 신뢰 시간(belief time)을 포함한다. 또한, 각 버전은 다른 데이터 객체들을 참조하는 외래 키를 포함한다. 외래 키는 지정된 데이터 객체의 복수의 버전들을 도출할 수 있다. 지정된 데이터 객체의 하나의 버전은 외래 키 도출(resolution)을 신뢰 시간에 관련시킴으로써 조회될 수 있다.

바람직한 실시예에서, 본 발명의 시스템은 (1) 하나의 외래 키로 테이블에 버저닝된 데이터 객체들을 저장하기 위한 버저닝된 데이터 구조; (2) 버저닝된 데이터 객체들을 기술하기 위한 주문화 데이터 사전; (3) 버저닝된 데이터 객체에 대하여 하나의 외래 키를 강제하기 위한 주문형 트리거; 및 (4) 테이블로부터 데이터 객체의 버전을 검색하기 위한 쿼리 생성기를 4개의 주요 소프트웨어 구성으로 포함한다.

버저닝된 데이터 구조는 데이터 객체를 유일하게 식별하는 일차 키 필드, 데이터 객체의 기술적 필드들, 데이터 레코드가 그 때부터 유효한 것으로 신뢰되는 날짜와 시간을 표시하는 신뢰 시간 필드, 일차 키를 사용하여 테이블에서 또 다른 데이터 객체를 참조하는 외래 키 필드, 테이블에서 데이터의 각 버전을 표시하는 버전 번호 필드로 버저닝된 데이터 객체를 표현한다.

데이터 객체가 가질 수 있는 값, 데이터 객체의 문자적 기술, 및 데이터 객체를 완전히 기술하는데 사용될 수 있는 다른 정보 등과 같이, 테이블에 저장된 데이터 객체에 관련되는 모든 정보를 포함하는 주문화된 데이터 사전에 필드들이 특정된다.

데이터 객체 무결성(integrity)은 외래 키 필드에 할당된 값이 테이블에서 지원됨을 확인하는 주문형 트리거(custom trigger)에 의해 강제된다. 주문형 트리거는 지정된 데이터 객체의 새로운 버전이 테이블에 삽입되는 경우마다 호출된다.

지정된 데이터 객체에 대한 버전은 쿼리 생성기를 사용함에 의하여 테이블로부터 검색된다. 쿼리 생성기는 2개의 입력 파라미터, 즉 버전을 검색하는데 필요한 필드 및 쿼리에 의해 리턴되는 버전을 제한하는 필터로 기능을 실행한다. 쿼리 생성기는 파라미터를 파싱(parsing)하고 테이블 내의 지정된 데이터 객체에 대한 버전을 추출하는 SQL 코드를 생성한다. 신뢰 시간은 지정된 데이터 객체의 하나의 버전을 리턴하는 SQL 코드를 실행하기 위해 쿼리 생성기에 각 요청에 대하여 설정된다. 상용가능한 리포트 프리젠테이션 툴을 사용하여 원하는 버전을 디스플레이하는 형식화된(formatted) 데이터베이스 리포트가 생성될 수 있다.

본 발명의 시스템 및 방법은 비즈니스 조직들이 데이터 객체들의 복수의 버전을 효율적으로 저장하고 지정된 시점에 대하여 데이터베이스 리포트를 생성하기 위하여 데이터 객체의 하나의 버전을 검색할 수 있게 한다는 장점을 갖는다.

도면의 간단한 설명

본 발명의 기술한 목적 및 다른 목적들은 이하의 도면들과 함께 다음의 상세한 설명으로 명확해 질 것이고, 도면 전체에서 동일한 참조 부호는 동일한 부분을 참조한다.

도 1은 본 발명의 원리에 따라 사용되는 소프트웨어 구성의 개략도이다.

도 2는 버저닝된 데이터 구조의 개략도이다.

도 3은 버저닝된 데이터 구조를 사용하여 종업원 정보를 표현하기 위한 테이블의 예시도이다.

도 4는 쿼리 생성기를 사용하여 테이블로부터 데이터 객체의 하나의 버전을 검색하는 흐름도이다.

도 5는 쿼리 생성기 파라미터의 예시도, 쿼리 생성기에 의해 파라미터로부터 생성되는 관계형 SQL 코드, 신뢰 시간의 세트에 대하여 쿼리 생성기에 의해 생성되는 결과이다.

실시예

도 1을 참조하면, 본 발명의 원리에 따라 사용되는 소프트웨어 구성의 개략도가 설명된다. 버저닝된 데이터 구조(10)는 (1) 데이터 객체를 유일하게 식별하는 일차 키 필드; (2) 데이터 객체의 특별한 특성을 특정하는 기술적 필드(descriptive field); (3) 데이터 레코드가 그 때부터 유효한 것으로 신뢰되는 날짜 및 시간을 표시하는 신뢰 시간 필드; (4) 일차 키를 사용하여 테이블에서 또 다른 데이터 객체를 참조하는 외래 키 필드; 및 (5) 테이블에서 데이터 객체의 각 버전을 표시하는 버전 번호 필드를 포함하는 여러 필드로서 버저닝된 데이터 객체를 표현한다.

데이터 객체가 가질 수 있는 값, 데이터 객체의 문자적 기술, 및 데이터 객체를 완전히 기술하는데 사용될 수 있는 다른 정보 등과 같이, 테이블에 저장된 데이터 객체에 관련되는 모든 정보를 포함하는 주문화된 데이터 사전(15)에 필드들이 특정된다.

데이터 객체 무결성(integrity)은 외래 키 필드에 할당된 값이 테이블에서 지원됨을 확인하는 주문형 트리거(custom trigger: 20)에 의해 강제된다. 주문형 트리거(20)은 지정된 데이터 객체의 새로운 버전이 테이블에 삽입되는 경우마다 호출된다.

쿼리 생성기(25)의 기능은 테이블로부터 지정된 데이터 객체의 하나 이상의 버전을 검색하는 것이다. 쿼리 생성기(25)는 2개의 입력 파라미터, 즉 버전을 검색하는데 필요한 필드 및 쿼리에 의해 리턴되는 버전을 제한하는 필터를 취득한다. 쿼리 생성기(25)는 파라미터를 파싱하고 테이블 내의 지정된 데이터 객체에 대한 버전을 추출하는 SQL 코드를 생성한다. 신뢰 시간은 지정된 데이터 객체의 하나의 버전을 리턴하는 SQL 코드를 실행하기 위해 쿼리 생성기(25)에 각 요청에 대하여 설정된다. 상용가능한 리포트 프리젠테이션 툴을 사용하여 원하는 버전을 디스플레이하는 형식화된(formatted) 데이터베이스 리포트가 생성될 수 있다.

도 2를 참조하면, 버저닝된 데이터 구조의 개략도가 설명된다. 버저닝된 데이터 구조(10)은 (1) 일차 키 필드(30); (2) 기술적 필드(35); (3) 신뢰 시간 필드(40); (4) 외래 키 필드(45); 및 (5) 버전 번호 필드(50)을 포함하는 여러 필드로 버저닝된 데이터 객체를 표현한다.

일차 키 필드(30)은 정수값, 텍스트, 아스키 코드 또는 데이터베이스 테이블에서 로우를 유일하게 식별하는데 사용될 수 있는 다른 표현으로 데이터 객체를 유일하게 식별하는데 사용된다.

기술적 필드(35)는 지정된 객체의 특성을 기술하는데 사용될 수 있는 하나 이상의 필드를 포함한다. 예컨대, 종업원 데이터 객체는 종업원을 기술하는 이름(the first name) 필드, 성(the second name) 필드, 및 주소 필드를 가질 수 있다.

신뢰 시간 필드(40)는 데이터 객체의 지정된 버전이 테이블에 삽입되는 날짜 및 시간을 표시하는데 사용된다. 또한, 신뢰 시간 필드(40)는 버전이 그 때부터 유효한 것으로 신뢰되는 날짜 및 시간을 표시한다. 신뢰 시간 필드(40)를 사용하여 데이터 객체의 복수의 버전을 저장하는 것은 사용자가 데이터 객체의 현재 버전과 같은 데이터 객체의 특정 시간 표현을 추출할 수 있게 한다. 또한, 신뢰 시간 필드(40)는 관심있는 시간 간격에 대응하는 신뢰 시간 필드(40)에 대한 값을 가지는 모든 버전들에 대하여 데이터베이스를 조회함으로써 지정된 시간 간격 동안에 데이터베이스의 완전한 감사(audit)를 사용자가 수행할 수 있게 한다.

버전 번호 필드(50)은 데이터베이스에 저장된 지정된 데이터 객체의 각 버전을 유일하게 식별하는 유일한 ID이다. 마지막으로, 지정된 데이터 객체에 대하여, 외래 키 필드(45)는 일차 키 필드(30)을 사용하여 또 다른 데이터 객체에 대한 참조를 표현한다.

데이터베이스의 버저닝된 데이터를 표현하는데 사용될 수 있는 추가적인 필드들은 당업자에 의하여 이해될 것이다.

도 3을 참조하면, 버저닝된 데이터 구조를 사용하여 종업원 정보를 표현하기 위한 테이블의 예시도가 설명된다. 테이블(60)은 비즈니스 조직에 대한 버저닝된 종업원 정보를 표현하는데, 즉, 테이블(60)은 종업원 레코드로 구성된 데이터 객체에 관련된 정보를 저장하는데 사용된다. 일차 키 필드(30)는 테이블(60)에서 종업원 번호를 지정하는 증분 정수값으로 표현되고, 각 값은 테이블(60)에서 데이터 객체를 참조한다. 기술적 필드(35)는 종업원이 비즈니스 조직에서 동료들에 의해 불려지기를 원하는 이름을 목록화한 하나의 종업원 이름 필드(65)에 의해 테이블(60)에서 표현된다. 신뢰 시간(40)은 수정된 필드(70)에 의해 테이블(60)에서 표현되는데, 수정된 필드는 일차 키 필드(30)에 의해 식별된 지정된 버전이 테이블(60)에 입력되었던 날짜, 즉 지정된 종업원의 이름이 테이블(60)에서 수정되었던 날짜를 표시한다. 마지막으로, 외래 키 필드(50)은 관리자 필드(75)에 의해 표현되는데, 관리자 필드는 테이블(60)에서 각 종업원의 관리자를 표시한다. 또한, 테이블(60)은 각 종업원의 버전을 유일하게 식별하는 버전 넘버 필드(55)를 가진다.

예컨대, 종업원 1은 2002년 2월 2일에 "John", 2002년 6월 2일에 "Johnny", 2002년 8월 2일에 "Jack"으로 불렸다. 유사하게, 종업원 3은 2002년 2월 2일에 "Robert", 2002년 4월 2일에 "Bobby", 2002년 6월 2일에 "Bob"으로 불렸다. 종업원 3은 종업원 3의 버전 1, 2, 3의 관리자 필드(75)의 값에 의해 표시된 것과 같이 종업원 1에 의해 관리된다. 종업원 3의 관리자 이름을 판단하기 위하여, 관리자 필드(75)의 값은 대응하는 신뢰 시간 필드(70)에 대한 값으로 테이블(60)에서 검색(lookup)되어야 한다. 어떠한 신뢰 시간도 관리자 필드(75)의 값과 관련되지 않는 경우에, 테이블(60)에서 외래 키로서 사용된 때의 관리자 필드(75)의 값은 종업원 1의 3개의 다른 버전을 도출한다. 관리자 필드(75)의 값은 주문형 트리거(20)에 의해 강제된다.

도 4를 참조하면, 쿼리 생성기를 사용하여 테이블로부터 데이터 객체의 하나의 버전을 검색하기 위한 흐름도가 설명된다. 단계 85에서, 데이터 객체의 버전을 검색하는데 필요한 필드들이 특정된다. 예컨대, 사용자는 종업원의 이름 및 관리자를 디스플레이함에 의해 테이블(60; 도 3)에 저장된 종업원 3의 현재 버전을 보는 것을 원할 수 있다.

단계 90에서, 지정된 쿼리에 대하여 리턴되는 버전을 제한하는데 사용되는 필터가 특정된다. 단계 95에서, 파라미터가 파싱되고 대응하는 SQL 코드가 생성된다. 테이블에서 데이터 객체의 하나의 버전을 리턴하기 위하여, 신뢰 시간이 단계

100에서 설정된다. 마지막으로, SQL 코드는 단계 105에서 실행되고 단계 110에서 리포트가 생성된다. SQL 쿼리는 필요한 필드를 주문화 데이터 사전(15)에서 분석함으로써 생성된다. 상기 예에서, 쿼리 생성기(25)는 종업원 3의 현재 관리자가 "Jack"으로 불린다는 것을 찾을 것이다.

추가적인 단계들이 데이터베이스에서 버저닝된 테이블로부터 데이터 객체의 하나의 버전을 검색하는데 사용된다는 것을 당업자들은 이해할 것이다.

도 5를 참조하면, 쿼리 생성기 파라미터의 예시도, 쿼리 생성기에 의해 파라미터로부터 생성되는 관계형 SQL 코드, 신뢰 시간의 세트에 대하여 쿼리 생성기에 의해 생성되는 결과가 설명된다. 쿼리 생성 파라미터(120)는 종업원의 이름과 관리자 이름을 목록화함으로써 테이블(60)로부터 종업원 데이터 객체를 검색하도록 특정된다. 필터(125)는 "Bob"으로 불리는 모든 종업원들을 리턴하도록 특정된다. 파라미터(120) 및 필터(125)로 쿼리 생성기(25)를 실행하면 SQL 코드(130)를 얻는다. SQL 코드(130)는 지정된 신뢰 시간에 종업원의 하나의 버전을 테이블(60)으로부터 검색하는 "where" 구문을 갖는다.

"now"의 신뢰 시간으로 SQL 코드(130)를 실행하면 "Bob"이라는 이름을 갖는 모든 종업원들의 현재 버전을 리턴하고, 이것은 테이블(60)에서 버전(135)을 도출한다. 유사하게, 2002년 5월 2일의 신뢰 시간으로 SQL 코드(130)를 실행하면, 그날 "Bob"이라는 이름의 종업원이 테이블(60)에 목록화되지 않으므로, 종업원의 이름 및 관리자 이름에 대한 널 값을 갖는 버전(140)을 리턴한다.

본 발명의 특별한 실시예가 상술되었지만, 이 설명은 단지 예시의 목적임을 이해할 것이다. 본 발명의 특별한 특징은 일부 도면에 도시되고, 편의상 일부 다른 도면에는 도시되지 않았으며, 일부 특징은 본 발명에 따라 다른 특징들과 조합될 수 있다. 설명된 프로세스의 단계들은 다시 순서가 매겨지거나 조합될 수 있고 다른 단계들이 포함될 수 있다. 또 다른 변형이 본 발명의 개시로 당업자에게 명확해 질 것이고 그러한 변형들은 첨부된 청구항의 범위 내에 포함될 것으로 의도된다.

(57) 청구의 범위

청구항 1.

데이터베이스에서 데이터 객체의 히스토리를 유지하고 상기 데이터 객체의 버전에 대하여 상기 데이터베이스를 조회(query)하기 위한 방법으로서,

복수의 버전들로 상기 데이터 객체를 표현하기 위한 버저닝된(versioned) 데이터 구조를 제공하는 단계;

상기 데이터 객체를 참조하기 위한 하나의 외래 키(foreign key)를 제공하는 단계;

상기 데이터베이스로부터 상기 데이터 객체의 상기 버전을 검색하기 위한 쿼리 루틴을 생성하는 단계; 및

상기 데이터베이스로부터 특정 시간에 대응하는 상기 데이터 객체의 상기 버전을 검색하기 위하여 상기 특정 시간에 대하여 상기 쿼리 루틴을 실행하는 단계를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 2.

제1항에 있어서, 복수의 버전들로 상기 데이터 객체를 표현하기 위하여 버저닝된 데이터 구조를 제공하는 단계는 하나 이상의 일차 키(primary key) 필드; 기술적(desciptive) 필드; 신뢰 시간(belief time) 필드; 외래 키 필드; 및 버전 번호 필드를 포함하는 복수의 필드들을 포함하는 데이터 구조를 제공하는 단계를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 3.

제2항에 있어서, 상기 기술적 필드를 포함하는 데이터 구조를 제공하는 단계는 상기 데이터 객체의 특징을 특정하기 위한 필드를 제공하는 단계를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 4.

제2항에 있어서, 상기 신뢰 시간 필드를 포함하는 데이터 구조를 제공하는 단계는 상기 데이터 객체의 상기 버전이 상기 데이터베이스에 입력된 상기 시간을 특정하기 위한 필드를 제공하는 것을 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 5.

제2항에 있어서, 상기 외래 키 필드를 포함하는 데이터 구조를 제공하는 단계는 상기 일차 키 필드를 사용하여 상기 데이터베이스에서 데이터 객체를 참조하기 위한 필드를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 6.

제2항에 있어서, 상기 버전 번호 필드를 포함하는 데이터 구조를 제공하는 단계는 상기 데이터 객체의 상기 버전을 유일하게 식별하기 위한 필드를 제공하는 단계를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 7.

제1항에 있어서, 상기 데이터 객체를 참조하기 위한 하나의 외래 키를 제공하는 단계는 상기 외래 키의 상기 값에 의해 식별되는 상기 데이터 객체의 상기 복수의 버전들을 도출하기 위하여 상기 하나의 외래 키를 사용하는 단계를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 8.

제1항에 있어서, 상기 데이터베이스로부터 상기 데이터 객체의 상기 버전을 검색하기 위한 쿼리 루틴을 생성하는 단계는 상기 데이터 객체의 상기 버전을 검색하기 위하여 상기 복수의 필드들로부터 하나 이상의 필드들을 특정하는 단계를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 9.

제1항에 있어서, 상기 데이터베이스로부터 상기 데이터 객체의 상기 버전을 검색하기 위한 쿼리 루틴을 생성하는 단계는 필터를 특정하는 단계를 더 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 10.

제1항에 있어서, 상기 쿼리 루틴을 실행하는 단계는 SQL 루틴을 생성하기 위하여 상기 복수의 필드들로부터 하나 이상의 필드들을 상기 필터로 파싱하는 단계를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 11.

제1항에 있어서, 상기 데이터베이스로부터 특정 시간에 대응하는 상기 데이터 객체의 상기 버전을 검색하기 위하여 상기 특정 시간에 대하여 상기 쿼리 루틴을 실행하는 단계는 특정 신뢰 시간에 대하여 유효한 상기 데이터 객체의 상기 버전을 생성하기 위하여 상기 특정 신뢰 시간으로 상기 SQL 루틴을 실행하는 단계를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 방법.

청구항 12.

데이터베이스에서 데이터 객체의 히스토리를 유지하고 상기 데이터 객체의 버전에 대하여 상기 데이터베이스를 조회하기 위한 시스템으로서,

상기 데이터 객체를 복수의 버전들로 저장하기 위한 버저닝된 데이터베이스;

상기 복수의 버전에 대하여 하나의 외래 키를 강제하기 위한 주문형 트리거 루틴; 및

특정 시간에 상기 데이터 객체의 버전을 검색하기 위한 쿼리 루틴을 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 시스템.

청구항 13.

제12항에 있어서, 상기 버저닝된 데이터베이스는 하나 이상의 일차 키 필드; 기술적 필드; 신뢰 시간 필드; 외래 키 필드; 및 버전 번호 필드를 포함하는 복수의 필드들을 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 시스템.

청구항 14.

제13항에 있어서, 상기 기술적 필드는 상기 데이터의 버전의 특징을 특정하기 위한 필드를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 시스템.

청구항 15.

제13항에 있어서, 상기 신뢰 시간 필드는 상기 데이터 객체의 상기 버전이 상기 데이터베이스에 입력된 상기 시간을 특정하기 위한 필드를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 시스템.

청구항 16.

제13항에 있어서, 상기 외래 키 필드는 상기 일차 키 필드를 사용하여 상기 데이터베이스에서 데이터 객체를 참조하기 위한 필드를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 시스템.

청구항 17.

제13항에 있어서, 상기 버전 번호 필드는 상기 데이터 객체의 상기 버전을 유일하게 식별하기 위한 필드를 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 시스템.

청구항 18.

제12항에 있어서, 상기 쿼리 루틴은,

상기 데이터 객체의 상기 버전을 검색하기 위한 복수의 필드들을 특정하기 위한 선택 파라미터;

필터; 및

신뢰 시간 파라미터를 포함하는 SQL 루틴을 포함하는 것을 특징으로 하는 히스토리 유지 및 데이터베이스 조회 시스템.

요약

데이터베이스에서 데이터 객체의 복수의 버전을 효율적으로 저장하고 지정된 시간에 대응하는 데이터 객체의 하나의 버전을 디스플레이하는 리포트를 생성하기 위한 시스템 및 방법을 제공한다. 데이터 객체는 데이터 객체의 버전에 대응하는 신뢰 시간을 저장하기 위한 신뢰 시간 필드(40)를 포함하는 버저닝된 데이터 구조(10)에 의해 표현된다. 데이터 객체의 하나의 버전은 필드를 특정하기 위한 SELECT 구문 및 검색되는 데이터 객체의 버전을 특정하기 위한 WHERE 구문을 포함하는 SQL(95)을 생성함에 의해 검색된다. 하나의 외래 키는 동일한 데이터 객체의 복수의 버전들에 접근하는데 사용된다.

대표도

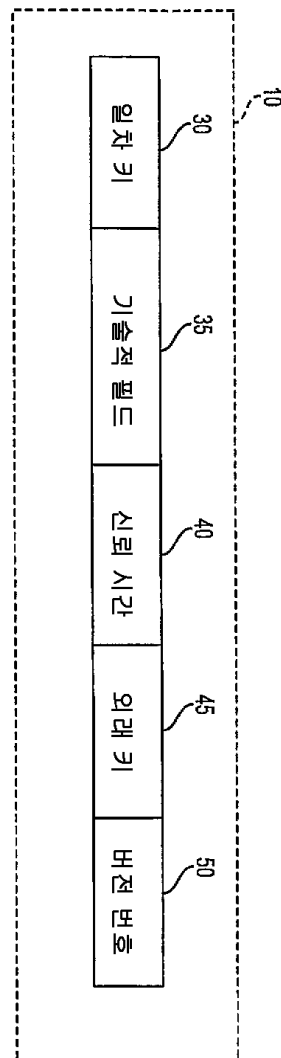
도 4

도면

도면1



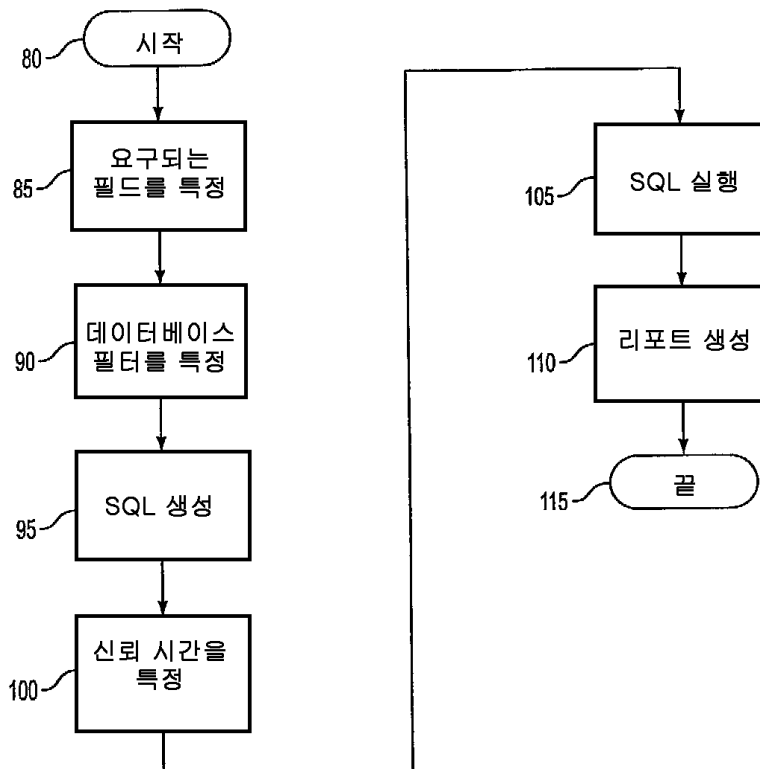
도면2



도면3

1	JOHN	02/02/2002		1
1	JOHNNY	02/06/2002		2
1	JACK	02/08/2002		3
3	ROBERT	02/02/2002	1	1
3	BOBBY	02/04/2002	1	2
3	BOB	02/06/2002	1	3

도면4



도면5

