



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0055802
(43) 공개일자 2013년05월29일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) G06F 11/08 (2006.01)
(21) 출원번호 10-2011-0121418
(22) 출원일자 2011년11월21일
심사청구일자 2011년11월21일

(71) 출원인
인하대학교 산학협력단
인천광역시 남구 인하로 100, 인하대학교 (용현동)
(72) 발명자
유상봉
경기도 부천시 원미구 상2동 하안마을 2608-404호
(74) 대리인
이원희

전체 청구항 수 : 총 9 항

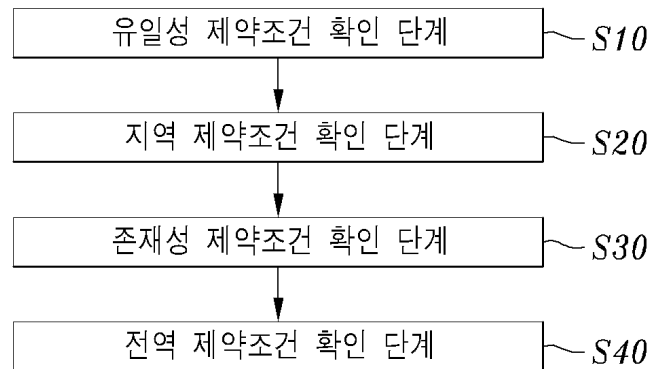
(54) 발명의 명칭 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법

(57) 요약

본 발명은 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법에 관한 것으로, 해결하고자 하는 과제는 익스프레스에서 선언된 데이터 무결성 제약 조건 중 관계형 데이터 베이스의 표준 언어인 SQL로는 확인하지 못하는 부분을 트리거를 이용하여 확인할 수 있는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법을 제공하는 것이다.

본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법은 관계형 데이터베이스에서 익스프레스(EXPRESS)가 정의한 무결성 제약조건을 확인하기 위해 SQL의 트리거(trigger)를 선언하여 상기 무결성 제약조건을 확인하는 것을 특징으로 한다.

대표도 - 도2



특허청구의 범위

청구항 1

관계형 데이터베이스에서 익스프레스(EXPRESS)가 정의한 무결성 제약조건을 확인하기 위해 SQL의 트리거(trigger)를 선언하여 상기 무결성 제약조건을 확인하되,

상기 무결성 제약조건 확인 방법은,

상기 무결성 제약조건 중 유일성 제약조건을 확인하는 유일성 제약조건 확인 단계;

상기 무결성 제약조건 중 지역 제약조건을 확인하는 지역 제약조건 확인 단계;

상기 무결성 제약조건 중 존재성 제약조건을 확인하는 존재성 제약조건 확인 단계; 및

상기 무결성 제약조건 중 전역 제약조건을 확인하는 전역 제약조건 확인 단계를 포함하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

청구항 2

제 1항에 있어서,

상기 유일성 제약조건 확인 단계에서,

다중 속성에 대한 유일성 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 카운트(count)문을 이용하여 확인하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

청구항 3

제 2항에 있어서,

상기 유일성 제약조건 확인 단계에서,

단일 속성에 대한 유일성 제약조건은 SQL의 UNIQUE 절로 확인하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

청구항 4

제 1항에 있어서,

상기 지역 제약조건 확인 단계에서,

다중 속성에 대한 지역 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 조건문을 이용하여 확인하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

청구항 5

제 4항에 있어서,

상기 지역 제약조건 확인 단계에서,

상기 다중 속성에 대한 지역 제약조건은 상기 트리거에서 조건문인 if 절을 이용하여 확인하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

청구항 6

제 4항에 있어서,

상기 지역 제약조건 확인 단계에서,

단일 속성에 대한 지역 제약조건은 SQL의 CHECK 절로 확인하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

청구항 7

제 1항에 있어서,

상기 존재성 제약조건 확인 단계에서,

상기 존재성 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 선택(select)문을 이용하여 확인하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

청구항 8

제 1항에 있어서,

상기 전역 제약조건 확인 단계에서,

상기 전역 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 조건문을 이용하여 확인하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

청구항 9

제 8항에 있어서,

상기 전역 제약조건 확인 단계에서,

상기 전역 제약조건은 상기 트리거에서 조건문인 if 절을 이용하여 확인하는 것을 특징으로 하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법.

명세서

기술분야

- [0001] 본 발명은 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법에 관한 것으로, 보다 상세하게는 익스프레스 언어로 선언된 무결성 제약조건을 관계형 데이터베이스에서 확인하기 위해 트리거를 사용하는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법에 관한 것이다.

배경기술

- [0002] STEP(Standard for the Exchange of Product model data)은 국제표준화기구(ISO)가 산업분야별 정보교환 및 공유를 위해 개발한 객체 지향형 제품모델 데이터의 표준으로, 상기 STEP에서는 특히 공학 데이터의 풍부한 의미를 잘 표현할 수 있도록 하는 정보 모델링 언어인 EXPRESS를 정의하고 있다.
- [0003] 상기 EXPRESS는 전형적인 객체 지향 언어이기 때문에, 엔티티의 정의 내에 데이터와 제약조건들을 함께 요약하여 표현할 수 있고, 또한 엔티티들의 상속 계층구조를 지원한다. 상기 STEP의 정보모델은 공통 자원, 응용 자원

및 응용 프로토콜로 분류될 수 있다.

- [0004] 연관된 공통 자원과 응용 자원은 응용 프로토콜에 의해 참조될 수 있고, 다른 응용 프로그램들은 응용 프로토콜을 통하여 제품 데이터를 교환할 수 있다. 이때, 공통 자원과 응용 자원에 정의되어 있는 정보모델은 전역 규칙들과 엔티티의 하위 타입들을 기술함으로써 응용 프로토콜에서 수정될 수 있다.
- [0005] 도 1은 데이터의 무결성이 유지되는 통합 시스템의 예를 보여주는 도이다.
- [0006] 도 1과 같은 통합 시스템에서는 공유되는 데이터의 무결성이 적절히 유지되지 않는 한 전체 시스템의 행위를 예측할 수 없게 되는 문제점이 있다. 따라서, 데이터의 무결성 확인 및 유지는 필수적이다.
- [0007] 도 1에 도시된 SDAI는 EXPRESS로 선언된 데이터를 접근하기 위한 표준 프로토콜이며, 특히 PDM/CE 환경에서는 한 시스템에 의해 전역 데이터베이스로 전달된 설계 데이터가 이 환경에 포함된 모든 응용 시스템들에 의해 이용되므로, 적절한 무결성 확인이 필요하다.
- [0008] 데이터베이스 품질 관리와 관련하여 한국 공개특허 제2008-0001248호는 데이터베이스 품질관리 시스템에 관해 기술하고 있다.
- [0009] 보다 상세하게는, 상기 한국 공개특허 제2008-0001248호는 데이터베이스 품질관리가 전사적으로 이루어지는 데이터베이스 품질관리 프레임워크를 기반으로 데이터베이스의 품질을 관리함으로써, 짧은 시간과 저렴한 비용으로 고품질의 데이터베이스를 관리할 수 있고, 품질관리 과정에서 발생할 수 있는 실수와 오류를 단계적으로 차단하며, 프레임워크에 포함된 각각의 품질관리 주체가 상호 보완하여 데이터베이스 품질을 고품질로 유지토록 할 수 있다.
- [0010] 또한, 한국 공개특허 제2004-0047439호는 컴퓨터 시스템에 구축된 각종 데이터베이스에 의해 관리되는 데이터의 무결성을 관리하는 장치 및 그 방법에 관해 기술하고 있다.
- [0011] 보다 상세하게는, 상기 공개특허 제2004-0047439호는 업무 데이터베이스의 업무적 논리 구조를 테이블간의 관계, 비정규화 컬럼, 컬럼 형식/경계값 및 업무적인 연계성으로 구분하여 분석하고 논리에 위배되는 데이터를 추출하는 단계 및 추출된 오류 데이터를 업무적 논리 구조에 따라 보정하는 단계를 포함하여 업무 데이터베이스 전체의 무결성 오류 데이터를 일괄적이고 자동화된 방식으로 추출하고 보정할 수 있을 뿐만 아니라, 시스템간의 무결성 오류 데이터도 자동화된 방식으로 추출하고 보정할 수 있다.
- [0012] 더불어, 한국 공개특허 제2006-0100466호는 데이터베이스 또는 유사 정보 저장소 상에 저장된 데이터 기록 세트의 무결성을 보장하기 위한 방법, 시스템 및 컴퓨터 프로그램에 관해 기술하고 있다.
- [0013] 보다 상세하게는, 상기 공개특허 제2006-0100466호는 데이터베이스 상에 저장될 제 2 데이터 기록을 수신하는 단계와, 상기 저장될 제 2 데이터 기록 앞의 제 1 데이터 기록과 함께 저장된 제 1 무결성 체크섬(integrity checksum)을 검색하는 단계와, 저장소 키, 상기 검색된 제 1 무결성 체크섬 및 상기 저장될 제 2 데이터 기록에 기초하여, 암호화 방법으로 상기 저장될 제 2 데이터 기록에 대한 제 2 무결성 체크섬을 계산하는 단계 및 상기 데이터베이스 상에 상기 제 2 데이터 기록 및 상기 제 2 무결성 체크섬을 저장하는 단계를 포함하여 무결성 검사를 갖는 신뢰성있는 데이터베이스를 허용하여 단지 서명 당국만이 데이터베이스 콘텐츠를 변경할 수 있고, 데이터베이스 상에 저장된 데이터 기록들은 계산된 무결성 체크섬들의 체인(chain)을 차단함이 없이 어떠한 방식으로든 삭제 또는 변경될 수 없도록 하고 있다.
- [0014] 그러나, 상기 출원에서는 상기 익스프레스에서 선언된 데이터 무결성 제약 조건을 확인하는 방법에 대한 기재가 없으며, 익스프레스에서 선언된 데이터를 확인하지 못함에 따라 데이터베이스에 포함된 데이터가 오류를 포함할 가능성이 있다.

[0015] 구체적으로, 데이터의 무결성 확인은 데이터베이스로부터 검색될 동일 타입 또는 다른 타입의 엔티티 인스턴스들에 대한 많은 교차 참조를 요구하기 때문에 비용이 증가되고, 특히, 관계형 데이터 베이스의 표준 언어인 SQL로는 익스프레스에서 선언된 데이터 무결성 제약 조건 중 확인하지 못하는 부분을 포함하고 있으므로, 제품 데이터의 무결성을 확인하여 데이터의 오류를 제거하지 못하는 문제점이 있다.

발명의 내용

해결하려는 과제

[0016] 본 발명은 상기한 바와 같은 문제를 해결하기 위해 발명된 것으로, 익스프레스에서 선언된 데이터 무결성 제약 조건 중 관계형 데이터 베이스의 표준 언어인 SQL로는 확인하지 못하는 부분을 트리거를 이용하여 확인할 수 있는 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법을 제공하는 것을 목적으로 한다.

과제의 해결 수단

[0017] 상기한 바와 같은 목적을 달성하기 위해 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법은 관계형 데이터베이스에서 익스프레스(EXPRESS)가 정의한 무결성 제약조건을 확인하기 위해 SQL의 트리거(trigger)를 선언하여 상기 무결성 제약조건을 확인하는 것을 특징으로 한다.

[0018] 또한, 상기 무결성 제약조건 확인 방법은 상기 무결성 제약조건 중 유일성 제약조건을 확인하는 유일성 제약조건 확인 단계와, 상기 무결성 제약조건 중 지역 제약조건을 확인하는 지역 제약조건 확인 단계와, 상기 무결성 제약조건 중 존재성 제약조건을 확인하는 존재성 제약조건 확인 단계 및 상기 무결성 제약조건 중 전역 제약조건을 확인하는 전역 제약조건 확인 단계를 포함할 수 있다.

[0019] 또한, 상기 유일성 제약조건 확인 단계에서, 다중 속성에 대한 유일성 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 카운트(count)문을 이용하여 확인할 수 있다.

[0020] 또한, 상기 유일성 제약조건 확인 단계에서, 단일 속성에 대한 유일성 제약조건은 SQL의 UNIQUE 절로 확인할 수 있다.

[0021] 또한, 상기 지역 제약조건 확인 단계에서, 다중 속성에 대한 지역 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 조건문을 이용하여 확인할 수 있다.

[0022] 또한, 상기 지역 제약조건 확인 단계에서, 상기 다중 속성에 대한 지역 제약조건은 상기 트리거에서 조건문인 if 절을 이용하여 확인할 수 있다.

[0023] 또한, 상기 지역 제약조건 확인 단계에서, 단일 속성에 대한 지역 제약조건은 SQL의 CHECK 절로 확인할 수 있다.

[0024] 또한, 상기 존재성 제약조건 확인 단계에서, 상기 존재성 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 선택(select)문을 이용하여 확인할 수 있다.

[0025] 또한, 상기 전역 제약조건 확인 단계에서, 상기 전역 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 조건문을 이용하여 확인할 수 있다.

[0026] 또한, 상기 전역 제약조건 확인 단계에서, 상기 전역 제약조건은 상기 트리거에서 조건문인 if 절을 이용하여 확인할 수 있다.

발명의 효과

[0027] 상기한 바와 같이 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법에 의하면, 익스프레스에서 선언된 데이터 무결성 제약 조건 중 관계형 데이터 베이스의 표준 언어인 SQL로는 확인하지 못하는 부분을 트리거를 이용하여 제품 데이터의 무결성을 확인함으로써 데이터의 오류를 제거하여 제품의 품질을 향상시키고 설계 원가를 감소시킬 수 있는 효과가 있다.

도면의 간단한 설명

- [0028] 도 1은 데이터의 무결성이 유지되는 통합 시스템의 예를 보여주는 도.
- 도 2는 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법의 일 블록도.
- 도 3은 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법 중 유일성 제약조건 확인 단계의 블록도.
- 도 4는 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법 중 지역 제약조건 확인 단계의 블록도.
- 도 5는 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법 중 존재성 제약조건 확인 단계의 블록도.
- 도 6은 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법 중 전역 제약조건 확인 단계의 블록도.
- 도 7은 본 발명의 이용에 적합한 컴퓨터 일 시스템의 블록도.

발명을 실시하기 위한 구체적인 내용

- [0029] 이하, 첨부된 도면을 참조하여 본 발명의 실시 예를 상세히 설명한다. 우선, 도면들 중 동일한 구성요소 또는 부품들은 가능한 한 동일한 참조부호를 나타내고 있음에 유의해야 한다. 본 발명을 설명함에 있어서 관련된 공지기능 혹은 구성에 대한 구체적인 설명은 본 발명의 요지를 모호하게 하지 않기 위해 생략한다.
- [0030] 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법은 관계형 데이터베이스에서 익스프레스(EXPRESS)가 정의한 무결성 제약조건을 확인하기 위해 SQL의 트리거(trigger)를 선언하여 상기 무결성 제약조건을 확인할 수 있다.
- [0031] 익스프레스(EXPRESS)는 STEP의 공식적인 정보 모델링 언어로서 객체 지향적이며 다양한 무결성 제약조건을 정의할 수 있는 강력한 표현력을 가지고 있으므로 일반적인 알고리즘을 포함할 수 있으나, 표현이 강해진 만큼 요구되는 계산도 많아진다.
- [0032] 상기 무결성 제약조건에 포함되는 알고리즘들은 개별적으로 실행 가능한 함수들로 변환되며, 이러한 함수들은 무결성 제약조건들이 확인될 동안에 필요할 경우 호출될 수 있다.
- [0033] 한편, 상기 익스프레스(EXPRESS)에서는 유일성 제약조건, 지역 제약조건, 존재성 제약조건 및 전역 제약조건과 같은 네 종류의 무결성 제약조건들이 있는데, 본 발명은 이러한 각 무결성 제약조건을 관계형 데이터베이스에서 확인하는 방법을 제시한다.
- [0034] 도 2는 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법의 일 블록도이다.
- [0035] 구체적으로, 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법은 도 2에 도시된 바와 같이, 유일성 제약조건 확인 단계(S10)와, 지역 제약조건 확인 단계(S20)와, 존재성 제약조건 확인 단계(S30) 및 전역 제약조건 확인 단계(S40)를 포함한다.
- [0036] 도 3은 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법 중 유일성 제약조건 확인 단계의 블록도이다.

[0037] 상기 유일성 제약조건 확인 단계(S10)는 상기 무결성 제약조건 중 유일성 제약조건을 확인하는 단계로, 도 3에 도시된 바와 같이, 단일 속성에 대한 유일성 제약조건은 SQL의 UNIQUE 절로 확인할 수 있고, 다중 속성에 대한 유일성 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 카운트(count)문을 이용하여 확인할 수 있다.

[0038] 구체적으로, UNIQUE 절은 엔티티 타입의 단 하나의 속성 또는 여러 속성들의 집합에 대하여 유일성 제약조건을 선언할 수 있다. 이때, 단일 UNIQUE 절이 단 하나의 속성에 대하여 정의된다면, 두 개 이상의 엔티티 인스턴스들은 그 속성에 대하여 동일한 값을 가질 수 없으며, 여러 속성의 집합에 대한 UNIQUE 절의 정의는 명명된 속성들에 대한 동일한 값들의 조합을 제한할 수 있다.

[0039] 여기서, 단일 속성에 대한 유일성 제약조건은 SQL의 UNIQUE 절로 확인할 수 있으나, 다중 속성에 대한 유일성 제약조건은 SQL의 UNIQUE 절에서 지원되지 않으며, 트리거(trigger)를 선언하여 카운트(count)문을 이용하여 확인할 수 있다.

[0040] 예를 들면, 하기의 예에서 ur1은 단일 속성에 대한 유일성 선언이고, ur2는 다중 속성에 대한 유일성 선언이다.

ENTITY part;

id : STRING;

name: STRING;

alternate_name: STRING;

quantities _in_superitem: INTEGER;

UNIQUE

ur1: id;

ur2: name, alternate_name;

END_ENTITY;

[0041]

[0042] 이러한 익스프레스(EXPRESS) 엔티티는 SQL에서 하기와 같이 변환되며, 유일성 제약조건 ur1은 UNIQUE(id)로 변환된다.

```
CREATE TABLE part
( id VARCHAR2(8),
  name VARCHAR2(8),
  alternate_name VARCHAR2(8),
  quantities_in_superitem INTEGER,
  UNIQUE (id) );
```

[0043]

[0044] 한편, 유일성 제약조건 ur2를 확인하기 위해서는, 하기와 같은 트리거(trigger)의 선언이 필요하다.

```
CREATE OR REPLACE TRIGGER "part_ur2"
AFTER insert OR update ON "part" FOR EACH ROW
BEGIN
  if ( count (SELECT name, alternate_name FROM part) <>
      count (SELECT DISTINCT name, alternate_name FROM part) )
  then dbms_output.put_line('Warning: uniqueness constraint ur2 is
violated');
END;
```

[0045]

[0046] 여기서 SELECT DISTINCT는 중복된 값을 제외하기 때문에, SELECT 문과 SELECT DISTINCT 문의 결과의 개수를 count() 함수를 이용하여 계산하고 이를 비교하여 중복 여부를 판단할 수 있다.

[0047] 도 4는 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법 중 지역 제약조건 확인 단계의 블록도이다.

[0048] 상기 지역 제약조건 확인 단계(S20)는 상기 무결성 제약조건 중 지역 제약조건을 확인하는 단계로, 도 4에 도시된 바와 같이, 단일 속성에 대한 지역 제약조건은 SQL의 CHECK 절로 확인할 수 있고, 다중 속성에 대한 지역 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 조건문, 구체적으로 if 절을 이용하여 확인할 수 있다.

[0049] 구체적으로, WHERE 절은 엔티티 타입의 각 인스턴스에 적용되는 지역 제약조건들을 정의하며, 지역 제약조건들은 하나의 속성 또는 여러 속성들의 조합에 대한 유효한 값을 기술한다.

[0050] 여기서, 단일 속성값에 대한 지역 제약조건은 SQL의 CHECK 절에 의해 확인할 수 있으나, WHERE 절이 2개 이상의 다중 속성값에 대한 지역 제약조건을 포함하는 경우, 트리거(trigger)를 선언하여 조건문을 통해 확인할 수 있

다.

[0051] 예를 들면, 하기의 익스프레스(EXPRESS) 엔티티의 선언은 WHERE 절에 두 개의 지역 제약조건을 포함하고 있는데, 여기서, `positive_a`는 단일 속성에 대한 지역 제약조건이고, `length_1`은 다중 속성에 대한 지역 제약조건이다.

```
ENTITY unit_vector;
```

```
    a, b, c : REAL;
```

```
WHERE
```

```
    positive_a : a > 0;
```

```
    length_1 : a**2 + b**2 + c**2 = 1.0;
```

```
END_ENTITY;
```

[0052]

[0053] 여기서, 지역 제약조건 `positive_a`는 SQL에서 하기와 같이 CHECK 절로 확인할 수 있다.

```
CREATE TABLE unit_vector
```

```
    a decimal, b decimal, c decimal,
```

```
    CHECK ( a > 0 );
```

[0054]

[0055] 한편, 지역 제약조건 `length_1`은 하기와 같이 트리거(trigger)를 선언하여 조건문인 if 절로 확인할 수 있다.

```
CREATE OR REPLACE TRIGGER "unit_vector_length_1"
```

```
AFTER insert OR update ON "unit_vector" FOR EACH ROW
```

```
BEGIN
```

```
    if (SQUARE (a) + SQUARE (b) + SQUARE (c) <> 1.0)
```

```
        then dbms_output.put_line('Warning: local constraint length_1 is
violated');
```

```
END;
```

[0056]

[0057] 도 5는 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법 중 존재성 제약조건 확인 단계의 블록도이다.

[0058] 상기 존재성 제약조건 확인 단계(S30)는 상기 무결성 제약조건 중 존재성 제약조건을 확인하는 단계로, 도 5에 도시된 바와 같이, 상기 존재성 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 선택(select)문을 이용하여 확인할 수 있다.

[0059] 구체적으로, 존재성 제약조건은 한 엔티티의 인스턴스 존재 여부가 다른 관련된 엔티티의 인스턴스 존재에 의존함을 선언한 것으로, 만일 다른 엔티티가 명시적인 속성에 의하여 현재 엔티티와의 관계를 설정하였다면, 역(inverse) 속성은 현재 엔티티의 문맥에 따라 그 관계를 기술하기 위하여 사용될 수 있다.

[0060] 상기 존재성 제약조건은 두 엔티티 인스턴스 간의 관계로서 SQL의 트리거(trigger)를 선언하여 확인할 수 있다.

[0061] 예를 들면, 하기의 예는 두 엔티티 bolt와 nut를 포함하고 있으며, 엔티티 bolt는 속성으로 대응하는 nut를 기술하기 위하여 fastener를 가지고 있고, 엔티티 nut는 대응하는 bolt를 기술하기 위하여 fasten을 속성으로 가지고 있다.

[0062] 여기서, 데이터베이스의 무결성이 적절히 유지되도록 하기 위해서는 서로 대응하는 bolt와 nut의 인스턴스 쌍이 데이터베이스 내에 존재해야 한다.

ENTITY bolt;

id: STRING;

size: STRING;

fastener: nut;

END_ENTITY;

ENTITY nut;

id: STRING;

size: STRING;

INVERSE

fasten: bolt FOR fastener;

END_ENTITY;

[0063]

[0064] 여기서, 상기 엔티티들은 하기와 같은 SQL 테이블로 변환된다.

```
CREATE TABLE bolt
    id VARCHAR2(8),
    size VARCHAR2(8),
    fastener VARCHAR2(8),
    FOREIGN KEY (fastener) REFERENCES nut;
```

```
CREATE TABLE nut
    id VARCHAR2(8),
    size VARCHAR2(8),
    fasten VARCHAR2(8),
    FOREIGN KEY (fasten) REFERENCES bolt;
```

[0065]

[0066] 한편, 존재성 제약조건을 확인하기 위해서는 하기와 같이 트리거(trigger)를 선언하여 선택(select)문을 통해 확인할 수 있다.

```
CREATE OR REPLACE TRIGGER "nut_inverse"
AFTER insert OR update on "NUT" for each row
DECLARE
    matfastner VARCHAR2(8);
begin
    select fastner into matfastner from bolt where bolt.id
= :new.fasten;
    if (:new.id <> matfastner)
    then dbms_output.put_line('Warning: an existence constraint is
violated');
    end if;
end;
```

[0067]

[0068] 도 6은 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법 중 전역 제약조건 확인 단계의 블록도이다.

[0069] 상기 전역 제약조건 확인 단계(S40)는 상기 무결성 제약조건 중 전역 제약조건을 확인하는 단계로, 도 6에 도시된 바와 같이, 상기 전역 제약조건은 SQL의 트리거(trigger)를 선언한 후 상기 트리거에서 조건문, 구체적으로 if 절을 이용하여 확인할 수 있다.

[0070] 구체적으로, 전역 제약조건, 즉, 전역 규칙은 EXPRESS의 RULE 절로 정의될 수 있으며, 하나의 엔티티 타입 또는 여러 엔티티 타입의 인스턴스 집합 간의 제약조건들을 기술할 수 있는데, 여기서, RULE 절은 실행 가능한 문장과 실행의 결과에 의한 데이터의 유효성을 결정하는 WHERE 절로 구성되며, 전역 규칙의 확인을 위해 SQL 트리거(trigger)를 선언할 수 있다.

[0071] 예를 들어, 하기의 전역 규칙은 3차원 좌표계의 첫번째 팔분의와 일곱번째 팔분수에 포함된 point의 개수가 같다고 규정하고 있다.

```

RULE point_match FOR (point);

LOCAL

    first_oct, seventh_oct : SET OF POINT := []; -- initialized as empty
    set of point

END_LOCAL;

    first_oct := QUERY (temp <* point | (temp.x>0) AND (temp.y>0) AND
(temp.z>0));

    seventh_oct := QUERY (temp <* point | (temp.x<0) AND (temp.y<0)
AND (temp.z<0));

WHERE

    SIZEOF(first_oct) = SIZEOF(seventh_oct);

END_RULE;

```

[0072]

[0073] 여기서, 이러한 전역 제약조건은 하기와 같이 트리거(trigger)를 선언하여 조건문인 if 절을 통해 확인할 수 있다.

```

CREATE OR REPLACE TRIGGER "point_match"

AFTER insert OR delete OR update ON "point" FOR EACH ROW

BEGIN

    if ( count (SELECT * FROM point WHERE x>0 AND y>0 AND z>0)
<>

        count (SELECT * FROM point WHERE x<0 AND y<0 AND
z<0) )

    then dbms_output.put_line('Warning: global rule point_match is
violated');

END;

```

[0074]

[0075] 하기의 [표 1]은 상술한 바와 같은 무결성 제약조건의 종류에 따라 무결성 제약조건을 확인하는 방법을 정리한 표이다.

표 1

	무결성 제약조건 확인 방법	
유일성 제약조건	단일속성	SQL의 UNIQUE 절로 확인
	다중속성	SQL의 트리거(trigger)를 선언한 후 트리거에서 카운트(count)문을 이용하여 확인
지역 제약조건	단일속성	SQL의 CHECK 절로 확인
	다중속성	SQL의 트리거(trigger)를 선언한 후 트리거에서 조건문을 이용하여 확인
존재성 제약조건	SQL의 트리거(trigger)를 선언한 후 트리거에서 셀렉트(select)문을 이용하여 확인	
전역 제약조건	SQL의 트리거(trigger)를 선언한 후 트리거에서 조건문을 이용하여 확인	

[0076]

[0077] 이하, 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법을 실행하기 위한 일 시스템을 설명한다.

[0078] 도 7은 본 발명의 이용에 적합한 컴퓨터 시스템의 일 블록도이다.

[0079] 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법을 실행하기 위한 시스템은 범용 마이크로 컴퓨터 상에서 구현될 수 있다.

[0080] 구체적으로, 상기 시스템은 도 7에 도시된 바와 같이, 사용자 인터페이스(100), 사용자 입력 장치(200), 프로세서(300), 메모리(400)를 포함할 수 있고, 이들은 모두 버스(700) 또는 정보 전달을 위한 다른 통신 수단에 연결될 수 있다.

[0081] 상기 프로세서(300)는 상술한 본 발명의 방법을 수행하는 프로그램을 실행한다. 여기서, 상기 프로그램은 상기 메모리로 로딩되거나, 적당한 저장 매체 인터페이스(500)를 통하여 상기 메모리에 로딩되도록 저장 매체(600)상에 있을 수 있다.

[0082] 상술한 바와 같이, 본 발명에 따르면, 익스프레스에서 선언된 데이터 무결성 제약 조건 중 관계형 데이터 베이스의 표준 언어인 SQL로는 확인하지 못하는 부분을 SQL의 트리거를 이용하여 제품 데이터의 무결성을 확인함으로써 데이터 오류의 제거를 통해 제품의 품질을 향상시킬 수 있고, 설계 원가를 감소시킬 수 있다.

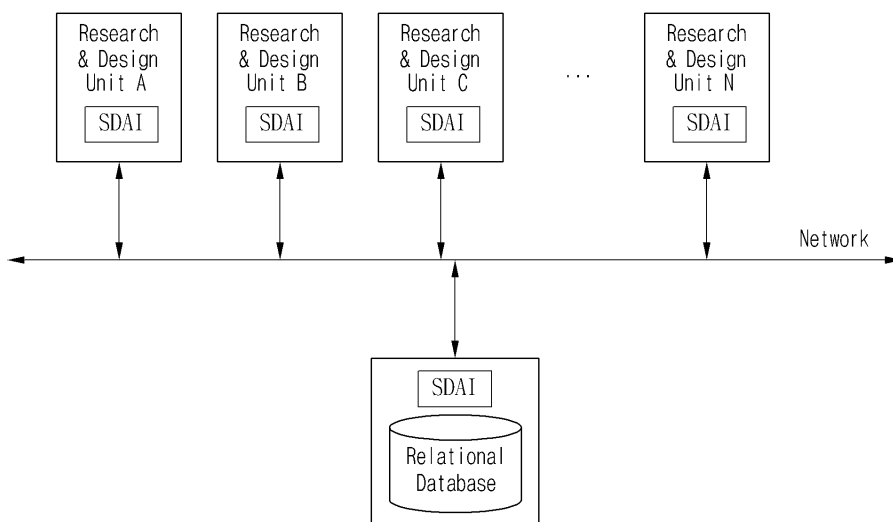
[0083] 이상과 같이 본 발명에 따른 관계형 데이터베이스에서 익스프레스 무결성 제약조건 확인 방법을 예시한 도면을 참조로 하여 설명하였으나, 본 명세서에 개시된 실시 예와 도면에 의해 본 발명이 한정되는 것은 아니며, 본 발명의 기술사상 범위 내에서 당업자에 의해 다양한 변형이 이루어질 수 있음은 물론이다.

부호의 설명

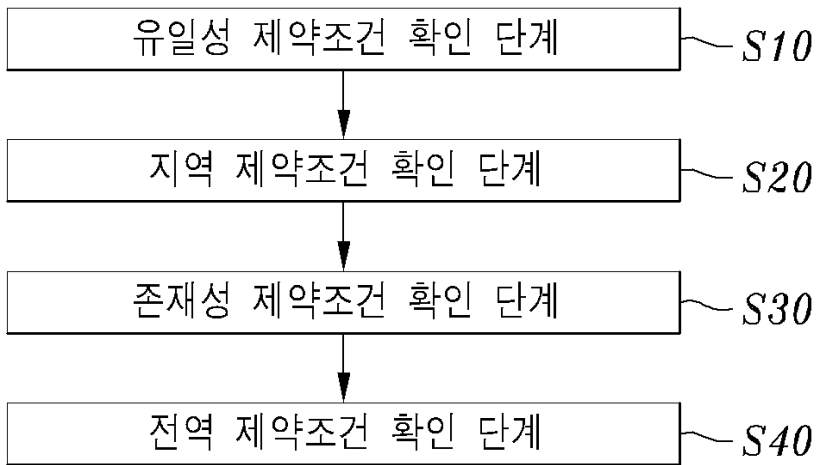
[0084] 100:사용자 인터페이스 200:사용자 입력 장치
300:프로세서 400:메모리
500:저장 매체 인터페이스 600:저장 매체
700:버스
S10:유일성 제약조건 확인 단계
S20:지역 제약조건 확인 단계
S30:존재성 제약조건 확인 단계
S40:전역 제약조건 확인 단계

도면

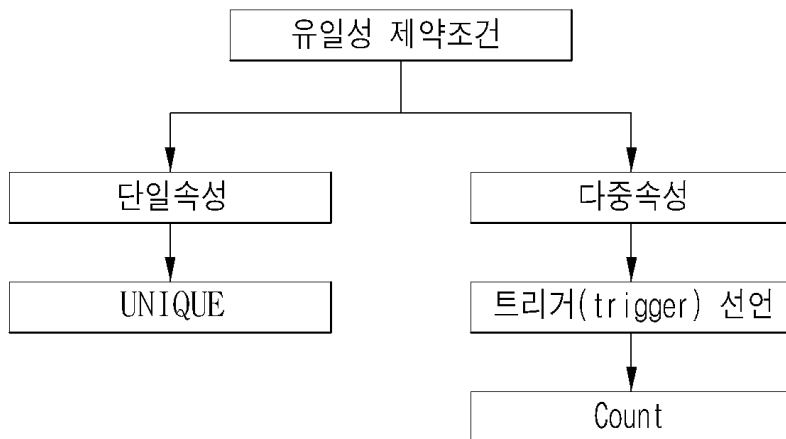
도면1



도면2



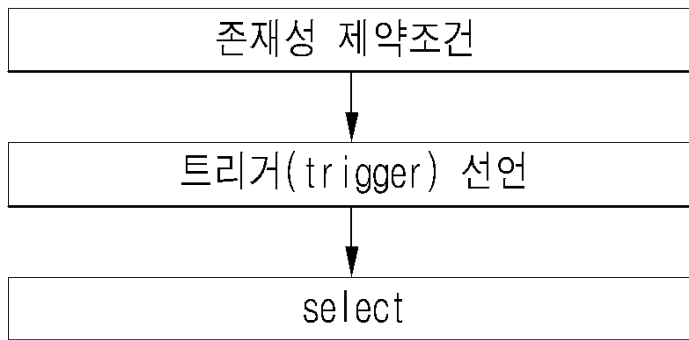
도면3



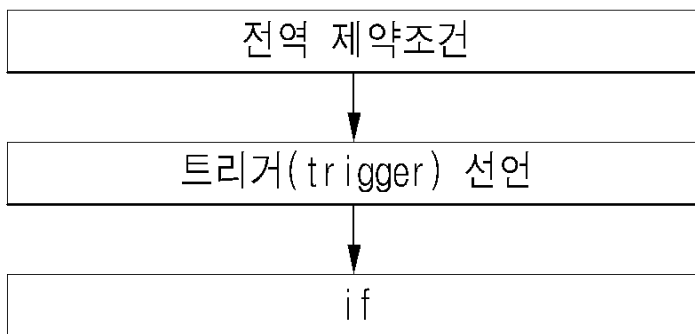
도면4



도면5



도면6



도면7

