



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년11월19일
(11) 등록번호 10-1919816
(24) 등록일자 2018년11월13일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01)

(52) CPC특허분류
G06F 17/30581 (2013.01)

(21) 출원번호 10-2017-0101054

(22) 출원일자 2017년08월09일

심사청구일자 2017년08월09일

(56) 선행기술조사문헌

JP2015049542 A

KR1020130101349 A

(73) 특허권자

네이버 주식회사

경기도 성남시 분당구 불정로 6, 그린팩토리 (정자동)

(72) 발명자

강철규

경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)

강동완

경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)

(뒷면에 계속)

(74) 대리인

양성보

전체 청구항 수 : 총 16 항

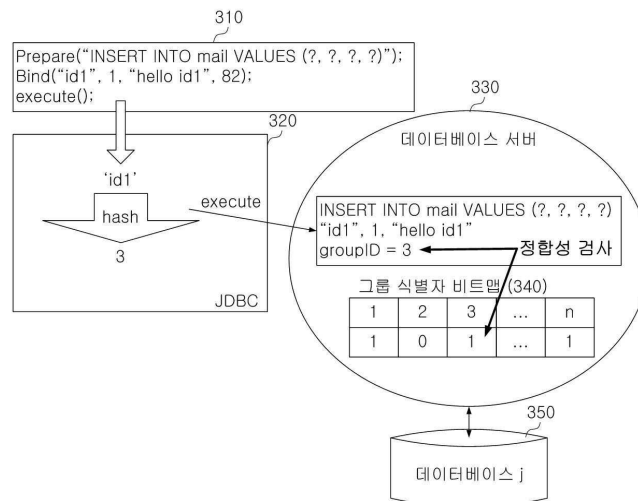
심사관 : 고재용

(54) 발명의 명칭 데이터베이스 샤딩 환경에서의 정합성 검사

(57) 요약

데이터베이스 샤딩 환경에서의 정합성 검사 기술을 제공한다. 일실시예에 따른 정합성 검사 방법은 샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들을 이용하여 다량의 데이터에 대한 정합성을 검사할 수 있다.

대표도 - 도3



(72) 발명자

김갑영

경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)

박부식

경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)

서경식

경기도 성남시 분당구 불정로 6(정자동, 그린팩토리)

명세서

청구범위

청구항 1

데이터베이스 서버의 정합성 검사 방법에 있어서,

샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 상기 데이터베이스 서버가 관리하는 샤드키가 분류된 그룹의 그룹 식별자들을 관리하는 단계; 및

요구된 연산과 관련된 샤드키에 대응하여 추출된 그룹 식별자를 이용하여, 상기 데이터베이스 서버에서 상기 요구된 연산과 연계된 데이터의 정합성을 검사하는 단계

를 포함하는 것을 특징으로 하는 정합성 검사 방법.

청구항 2

제1항에 있어서,

상기 그룹 식별자들을 관리하는 단계는,

상기 데이터베이스 서버가 관리하는 샤드키가 분류된 그룹 여부에 대한 정보가 상기 기설정된 수의 복수의 그룹들의 그룹 식별자들 각각에 대해 설정된 그룹 식별자 비트맵을 생성 및 관리하는 것을 특징으로 하는 정합성 검사 방법.

청구항 3

제1항에 있어서,

상기 그룹 식별자들을 관리하는 단계는,

데이터베이스 서버들간의 데이터의 이동에 따라 제1 그룹 식별자에 대응하는 데이터가 상기 데이터베이스 서버로 이동된 경우, 상기 제1 그룹 식별자가 상기 관리되는 그룹 식별자들에 포함되어 있는지 확인하는 단계; 및

상기 제1 그룹 식별자가 상기 관리되는 그룹 식별자들에 포함되어 있지 않은 경우, 상기 제1 그룹 식별자를 상기 관리되는 그룹 식별자들에 추가하여 더 관리하는 단계

를 포함하는 것을 특징으로 하는 정합성 검사 방법.

청구항 4

제1항에 있어서,

상기 그룹 식별자들을 관리하는 단계는,

데이터베이스 서버들간의 데이터의 이동에 따라 상기 데이터베이스 서버에서 관리하는 제1 그룹 식별자에 대응하는 데이터가 다른 데이터베이스 서버로 이동되는 경우, 상기 관리하는 그룹 식별자들에서 상기 제1 그룹 식별자를 제외하고 관리하는 것을 특징으로 하는 정합성 검사 방법.

청구항 5

제3항 또는 제4항에 있어서,

상기 데이터베이스 서버들간의 데이터의 이동은, 적어도 하나의 그룹 식별자에 대응하는 데이터의 단위로 이루어지는 것을 특징으로 하는 정합성 검사 방법.

청구항 6

제1항에 있어서,

상기 그룹 식별자는 샤드키들 각각을 기설정된 해시 함수의 파라미터로 입력하여 계산된 해시값을 포함하고, 동일한 해시값을 갖는 샤드키들이 동일한 그룹으로 분류되는 것을 특징으로 하는 정합성 검사 방법.

청구항 7

제1항에 있어서,

상기 정합성을 검사하는 단계는,

상기 추출된 그룹 식별자가 상기 관리되는 그룹 식별자들에 포함된 경우 정합성이 존재하는 것으로 결정하고, 상기 추출된 그룹 식별자가 상기 관리되는 그룹 식별자들에 포함되지 않는 경우 정합성이 결여된 것으로 결정하는 것을 특징으로 하는 정합성 검사 방법.

청구항 8

컴퓨터와 결합되어 제1항 내지 제4항, 제6항 또는 제7항 중 어느 한 항의 방법을 컴퓨터에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램.

청구항 9

제1항 내지 제4항, 제6항 또는 제7항 중 어느 한 항의 방법을 컴퓨터에 실행시키기 위한 프로그램이 기록되어 있는 것을 특징으로 하는 컴퓨터에서 판독 가능한 기록매체.

청구항 10

데이터베이스 서버를 위한 컴퓨터 장치에 있어서,

컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서
를 포함하고,

상기 적어도 하나의 프로세서는,

샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 상기 데이터베이스 서버가 관리하는 샤드키가 분류된 그룹의 그룹 식별자들을 관리하고,

요구된 연산과 관련된 샤드키에 대응하여 추출된 그룹 식별자를 이용하여, 상기 데이터베이스 서버에서 상기 요구된 연산과 연계된 데이터의 정합성을 검사하는 것

을 특징으로 하는 컴퓨터 장치.

청구항 11

제10항에 있어서,

상기 적어도 하나의 프로세서는,

상기 데이터베이스 서버가 관리하는 샤드키가 분류된 그룹 여부에 대한 정보가 상기 기설정된 수의 복수의 그룹들의 그룹 식별자들 각각에 대해 설정된 그룹 식별자 비트맵을 생성 및 관리하는 것

을 특징으로 하는 컴퓨터 장치.

청구항 12

제10항에 있어서,

상기 적어도 하나의 프로세서는,

데이터베이스 서버들간의 데이터의 이동에 따라 제1 그룹 식별자에 대응하는 데이터가 상기 데이터베이스 서버로 이동된 경우, 상기 제1 그룹 식별자가 상기 데이터베이스 서버에 의해 관리되고 있는지 여부를 확인하고,

상기 제1 그룹 식별자가 상기 관리되는 그룹 식별자들에 포함되어 있지 않은 경우, 상기 제1 그룹 식별자를 상기 관리되는 그룹 식별자들에 추가하여 더 관리하는 것

을 특징으로 하는 컴퓨터 장치.

청구항 13

제10항에 있어서,

상기 적어도 하나의 프로세서는,

데이터베이스 서버들간의 데이터의 이동에 따라 상기 데이터베이스 서버에서 관리하는 제1 그룹 식별자에 대응하는 데이터가 다른 데이터베이스 서버로 이동되는 경우, 상기 관리하는 그룹 식별자들에서 상기 제1 그룹 식별자를 제외하고 관리하는 것

을 특징으로 하는 컴퓨터 장치.

청구항 14

제12항 또는 제13항에 있어서,

상기 데이터베이스 서버들간의 데이터의 이동은, 적어도 하나의 그룹 식별자에 대응하는 데이터의 단위로 이루어지는 것을 특징으로 하는 컴퓨터 장치.

청구항 15

제10항에 있어서,

상기 그룹 식별자는 샤드키들 각각을 기설정된 해시 함수의 파라미터로 입력하여 계산된 해시값을 포함하고,

동일한 해시값을 갖는 샤드키들이 동일한 그룹으로 분류되는 것을 특징으로 하는 컴퓨터 장치.

청구항 16

제10항에 있어서,

상기 적어도 하나의 프로세서는,

상기 추출된 그룹 식별자가 상기 관리되는 그룹 식별자들에 포함된 경우 정합성이 존재하는 것으로 결정하고, 상기 추출된 그룹 식별자가 상기 관리되는 그룹 식별자들에 포함되지 않는 경우 정합성이 결여된 것으로 결정하는 것

을 특징으로 하는 컴퓨터 장치.

발명의 설명

기술 분야

[0001] 아래의 설명은 데이터베이스 샤딩 환경에서의 정합성 검사 기술에 관한 것이다.

배경 기술

[0002] 데이터베이스 샤딩(database sharding)은 대량의 데이터를 처리하기 위해 데이터베이스 테이블을 수평 분할(horizontal partitioning)하여 물리적으로 서로 다른 곳에 분산 저장 및 조회하는 기술에 관한 것이다. 예를 들어, 한국등록특허 제10-1527634호는 샤딩 서비스를 제공하는 방법 및 장치에 대해 개시하고 있다.

[0003] 이때, 대량의 데이터에 대한 정합성을 검증하는 것은 매우 어렵다. 예를 들어, 100만 명의 이용자들이 이용하는 서비스에서 이용자들 각각에 대한 데이터가, 다수의 데이터베이스들에 분산 저장된 경우를 고려할 수 있다. 이때, 다수의 데이터베이스들 각각은 자신이 저장하고 있는 데이터만을 관리하기 때문에, 100만 명의 이용자들 중 특정한 1명의 기존 이용자에 대한 데이터를 선택, 갱신, 삭제할 때, 이러한 데이터가 다수의 데이터베이스들 중 특정한 데이터베이스에 포함되어 있는지 확인하기 어렵다는 문제점이 있다.

[0004] 이를 명확히 파악하기 위해서는 다수의 데이터베이스들 각각이 100만 명의 이용자들 중 어느 이용자에 대한 정보를 포함하고 있는가에 대한 정합성 정보를 관리해야 하며, 1명의 새로운 이용자에 대한 데이터를 추가함에 있어서도 관리되는 정합성 정보가 다수의 데이터베이스 각각에 대해 갱신되어야 한다. 이러한 관리 및 갱신은

매우 큰 비용을 요구한다는 문제점이 있다.

발명의 내용

해결하려는 과제

- [0005] 샤드키를 그룹핑한 기설정된 수의 복수의 그룹들에 대한 그룹 식별자를 이용하여 대량의 데이터에 대한 정합성을 검사할 수 있는 정합성 검사 방법 및 정합성 검사 방법을 수행하는 컴퓨터 장치, 그리고 컴퓨터와 결합되어 본 발명의 실시예들에 따른 정합성 검사 방법을 컴퓨터에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램과 그 기록매체를 제공한다.
- [0006] 기설정된 수의 그룹 식별자들에 대해 샤드키들이 해당 그룹 식별자의 그룹에 포함되어 있는지 여부를 검사하는 것만으로 대량의 데이터에 대한 정합성을 검사할 수 있는 정합성 검사 방법 및 정합성 검사 방법을 수행하는 컴퓨터 장치, 그리고 컴퓨터와 결합되어 본 발명의 실시예들에 따른 정합성 검사 방법을 컴퓨터에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램과 그 기록매체를 제공한다.

과제의 해결 수단

- [0007] 데이터베이스 서버의 정합성 검사 방법에 있어서, 샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 상기 데이터베이스 서버가 관리하는 샤드키가 분류된 그룹의 그룹 식별자들을 관리하는 단계; 및 요구된 연산과 관련된 샤드키에 대응하여 추출된 그룹 식별자를 이용하여, 상기 데이터베이스 서버에서 상기 요구된 연산과 연계된 데이터의 정합성을 검사하는 단계를 포함하는 것을 특징으로 하는 정합성 검사 방법을 제공한다.
- [0008] 상기 정합성 검사 방법을 컴퓨터에 실행시키기 위한 컴퓨터 프로그램이 기록되어 있는 것을 특징으로 하는 컴퓨터 판독 가능한 기록매체를 제공한다.
- [0009] 컴퓨터와 결합하여 상기 정합성 검사 방법을 컴퓨터에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램을 제공한다.
- [0010] 데이터베이스 서버를 위한 컴퓨터 장치에 있어서, 컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는, 샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 상기 데이터베이스 서버가 관리하는 샤드키가 분류된 그룹의 그룹 식별자들을 관리하고, 요구된 연산과 관련된 샤드키에 대응하여 추출된 그룹 식별자를 이용하여, 상기 데이터베이스 서버에서 상기 요구된 연산과 연계된 데이터의 정합성을 검사하는 것을 특징으로 하는 컴퓨터 장치를 제공한다.

발명의 효과

- [0011] 샤드키를 그룹핑한 기설정된 수의 복수의 그룹들에 대한 그룹 식별자를 이용하여 대량의 데이터에 대한 정합성을 검사할 수 있다.
- [0012] 기설정된 수의 그룹 식별자들에 대해 샤드키들이 해당 그룹 식별자의 그룹에 포함되어 있는지 여부를 검사하는 것만으로 대량의 데이터에 대한 정합성을 검사할 수 있다.

도면의 간단한 설명

- [0013] 도 1은 본 발명의 일실시예에 있어서, 데이터베이스에 저장된 데이터의 예를 도시한 도면이다.
- 도 2는 본 발명의 일실시예에 있어서, 데이터의 샤딩 예를 도시한 도면이다.
- 도 3은 본 발명의 일실시예에 있어서, 정합성 검사 과정의 예를 도시한 도면이다.
- 도 4는 본 발명의 일실시예에 있어서, 잘못된 데이터 조회를 검사하는 예를 도시한 도면이다.
- 도 5는 본 발명의 일실시예에 있어서, 컴퓨터 장치의 내부 구성을 설명하기 위한 블록도이다.
- 도 6은 본 발명의 일실시예에 있어서, 정합성 검사 방법의 예를 도시한 흐름도이다.
- 도 7은 본 발명의 일실시예에 있어서, 그룹 식별자들을 관리하는 제1 예를 도시한 흐름도이다.

도 8은 본 발명의 일실시예에 있어서, 그룹 식별자를 관리하는 제2 예를 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0014] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.
- [0015] 본 발명의 실시예들에 따른 정합성 검사 방법은 컴퓨터 장치에 의해 수행될 수 있다. 일례로, 컴퓨터 장치에는 컴퓨터 장치가 정합성 검사 방법을 수행하도록 제어하기 위한 컴퓨터 프로그램이 설치 및 구동될 수 있으며, 컴퓨터 프로그램은 컴퓨터 장치와 결합되어 정합성 검사 방법을 컴퓨터 장치에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장될 수 있다.
- [0016] 도 1은 본 발명의 일실시예에 있어서, 데이터베이스에 저장된 데이터의 예를 도시한 도면이고, 도 2는 본 발명의 일실시예에 있어서, 데이터의 샤딩 예를 도시한 도면이다. 도 1 및 도 2는 데이터베이스(110)에 저장된 데이터에 대한 데이터베이스 샤딩(database sharding)에 의해, 데이터들이 n 개의 샤드 노드들(210, 220 및 230)로 분산된 예를 나타내고 있다. 예를 들어, 도 1 및 도 2에서는 레코드들을 식별하기 위한 식별자인 'id1', 'id2', 'id3', 'id4'를 샤드키(shardkey)로 이용하여 동일한 샤드키의 레코드들을 모아서 n 개의 샤드들(210, 220 및 230)로 분산한 예를 나타내고 있다. 이때, 샤드키 단위의 데이터는 동일한 샤드키를 통해 식별되는 데이터를 의미할 수 있다. 예를 들어, 도 2에서는 샤드키 'id2'를 통해 식별되는 데이터를 굵은 실선(240)을 통해 표시하고 있다.
- [0017] 샤드키 단위의 데이터는 용량과 QPS(Query Per Second)는 상대적으로 작은 편이나, 샤드키의 개수가 매우 많고, 따라서 결과적으로 데이터베이스 전체 용량과 QPS는 매우 크다는 특징을 갖는다. 또한, 샤드키를 통해 식별되는 데이터간의 관계(relation)은 의미가 있지만, 서로 다른 샤드키를 통해 식별되는 데이터간의 관계는 의미가 없다는 특징을 갖는다. 다시 말해, 하나의 샤드키를 통해 식별되는 데이터는 다른 샤드키를 통해 식별되는 데이터에 대해 의존성을 갖지 않으며, 샤드키 단위로 독립적인 개별 연산(트랜잭션)의 처리가 가능하다.
- [0018] n 개의 샤드 노드들(210, 220 및 230) 각각은 제1 샤드 노드(210)을 통해 나타나는 바와 같이, 하나의 샤드키에 대응하는 데이터를 포함할 수도 있으나, 제2 샤드 노드(220)을 통해 나타나는 바와 같이, 다수의 샤드키에 대응하는 데이터를 포함할 수도 있다.
- [0019] 이때, 본 발명의 실시예들에서는 앞서 설명한 샤드키 단위의 데이터와 관련하여 샤드키들을 기설정된 수의 복수의 그룹으로 분류하고, 복수의 그룹들의 그룹 식별자들을 통해 데이터에 대한 정합성 검사를 처리할 수 있다. 이때, 각각의 데이터베이스 서버들은 자신이 관리하는 샤드키가 분류된 그룹의 그룹 식별자들을 관리할 수 있다.
- [0020] 도 3은 본 발명의 일실시예에 있어서, 정합성 검사 과정의 예를 도시한 도면이다. 본 실시예들에 따른 데이터베이스 서버들 각각은 삽입(INSERT), 갱신(UPDATE), 삭제(DELETE)와 같은 모든 DML(Data Manipulation Language) 질의 수행 시 샤딩 데이터에 대한 정합성 검사를 수행할 수 있다.
- [0021] 도 3의 실시예에서는 입력된 쿼리문(310)에서 JDBC(Java DataBase Connectivity, 320)가 샤드키 'id1'을 추출하고, 샤드키 'id1'을 해싱하여 해시값 '3'을 획득하는 예를 나타내고 있다. 이때, 해시값 '3'은 샤드키 'id1'이 포함된 그룹의 그룹 식별자가 될 수 있다. 예를 들어, 모든 샤드키들 각각이 기설정된 해시 함수의 파라미터로 이용하여 모든 샤드키들 각각을 위한 그룹 식별자들이 계산될 수 있다. 보다 구체적인 예로, 100만 개의 샤드키들이 해시 함수를 거쳐 1만 개의 그룹 식별자들이 계산될 수 있고, 그룹 식별자 각각에 대한 1만개의 그룹으로 분류될 수 있다.
- [0022] 이 경우, 새로운 샤드키가 추가된다 하더라도 해시 함수를 통해 해시되는 과정에서 1만개의 그룹 중 하나로 분류되기 때문에 그룹이나 그룹 식별자는 추가되지 않는다. 따라서, 각각의 데이터베이스 서버들은 100만개의 샤드키들 중 자신이 관리하는 샤드키들을 관리하는 것이 아니라, 1만개의 그룹 식별자들 중 자신이 관리하는 그룹 식별자만을 관리함으로써, 특정 샤드키가 자신이 관리하는 그룹에 포함되어 있는지 여부를 확인할 수 있다.
- [0023] 이때, 데이터베이스 서버들간의 데이터 이동은 그룹 식별자에 대응하는 데이터 단위로 이루어질 수 있다. 예를 들어, 데이터베이스 서버 a이 그룹 식별자 1에 대한 데이터를 저장할 때, 데이터베이스 서버 a에서 데이터베이스 서버 b로의 데이터의 이동은 그룹 식별자 1에 대응하는 데이터의 단위로 이루어질 수 있다.
- [0024] 한편, 도 3에 도시된 바와 같이 데이터베이스 서버(330)는 그룹 식별자들 각각에 대해 자신이 관리하는 데이터에 대응하는 그룹 식별자와 자신이 관리하지 않는 데이터에 대응하는 그룹 식별자를 나타내기 위한 그룹 식별자

비트맵(340)을 관리할 수 있다. 예를 들어, 도 3에서 데이터베이스 서버(330)는 그룹 식별자 1, 3, n에 대응하는 데이터를 데이터베이스 j(350)에 저장하여 관리하며(그룹 식별자 1, 3, n의 값이 1로 설정됨), 그룹 식별자 2에 대응하는 데이터는 관리하지 않음(그룹 식별자 2의 값이 0으로 설정됨)을 알 수 있다.

[0025] 이때, 앞서 설명한 바와 같이, JDBC(320)는 쿼리문(310)에서 샤프키 'id1'를 추출하고, 추출된 샤프키 'id1'를 해시 함수를 통해 해싱하여 그룹 식별자 '3'을 추출할 수 있다. 이때, 추출된 그룹 식별자 '3'은 데이터베이스 서버(330)로 전달될 수 있고, 데이터베이스 서버(330)은 추출된 그룹 식별자 '3'을 그룹 식별자 비트맵(340)에서 검색하여 그룹 식별자 '3'에 대응하는 데이터가 데이터베이스 j(350)에 저장되어 데이터베이스 서버(330)에 의해 관리됨을 확인할 수 있다. 다시 말해, 데이터베이스 서버(330)는 요구된 연산과 연계된 데이터의 정합성을 검사할 수 있게 된다. 만약, 추출되어 수신된 그룹 식별자가 '2'라면, 데이터베이스 서버(330)는 잘못된 데이터의 변경 요구가 입력되었음을 파악할 수 있게 된다.

[0026] 또한, 데이터베이스 서버(330)는 데이터의 변경 요구뿐만 아니라, 데이터의 조회에 대해서도 정합성 검사를 수행할 수 있다.

[0027] 도 4는 본 발명의 일실시예에 있어서, 잘못된 데이터 조회를 검사하는 예를 도시한 도면이다. 도 4는 데이터베이스 서버 1(410)과 데이터베이스 서버 2(420) 그리고, 데이터베이스 서버 1(410)이 관리하는 데이터베이스 1(411)과 데이터베이스 서버 2(420)가 관리하는 데이터베이스 2(421)를 각각 도시하고 있다.

[0028] 도 4에서는 데이터베이스 1(411)의 그룹 식별자 '3'에 대응하는 데이터가 데이터베이스 2(421)로 이전된 예를 가정한다. 또한, 데이터베이스 1(411)에는 아직 그룹 식별자 '3'에 대응하는 데이터가 삭제되지 않아서 데이터베이스 1(411)과 데이터베이스 2(412) 모두에 그룹 식별자 '3'에 대응하는 데이터가 존재하는 경우를 가정한다. 또한, 모든 데이터베이스 서버들에 대해 샤프키 '1'에 대한 메일을 검색하는 쿼리문(412, 422)이 입력된 경우를 가정한다. 이때, 요청된 연산에 대한 결과는 데이터베이스 1(411)로부터 조회된 결과와 데이터베이스 2(421)로부터 조회된 결과를 모두 포함하게 되기 때문에 잘못된 결과를 제공할 가능성이 존재한다. 예를 들어, 그룹 식별자 '3'에 대응하는 데이터가 데이터베이스 2(421)에서 변경된 경우, 변경전 조회 결과와 변경후 조회 결과가 모두 제공되게 되고, 이는 데이터의 정합성을 훼손시키게 된다.

[0029] 본 실시예에서는 이러한 문제를 방지하기 위해, 그룹 식별자 비트맵들(413, 423)이 활용될 수 있다. 샤프키 '1'에 대한 해시값이 '3'이라 가정할 때, 그룹 식별자 '3'이 추출될 수 있으며, 데이터베이스 서버 1(410)은 그룹 식별자 '3'을 그룹 식별자 비트맵(413)을 통해 검색함에 따라 그 값이 '0'이고 따라서 데이터베이스 서버 1(410)은 자신이 그룹 식별자 '3'에 대응하는 데이터를 관리하지 않음을 파악할 수 있다.

[0030] 예를 들어, 데이터베이스 서버 1(410)은 그룹 식별자 '3'에 대응하는 데이터를 데이터베이스 서버 2(420)로 이전할 때, 그룹 식별자 비트맵(413)에서 그룹 식별자 '3'에 대응하는 값을 '1'에서 '0'으로 변경할 수 있다. 따라서, 데이터베이스 서버 1(410)은 그룹 식별자 '3'에 대응하는 데이터가 데이터베이스 1(411)에 아직 남아있다 하더라도, 그룹 식별자 '3'에 대응하는 데이터가 자신이 관리하는 데이터가 아님을 파악할 수 있다.

[0031] 반면, 데이터베이스 서버 2(420)는 그룹 식별자 '3'에 대응하는 데이터를 데이터베이스 서버 1(410)로부터 이전(migration)받을 때, 그룹 식별자 비트맵(423)에서 그룹 식별자 '3'에 대응하는 값을 '0'에서 '1'으로 변경할 수 있다. 따라서, 데이터베이스 서버 2(420)는 데이터베이스 2(421)에 저장된 그룹 식별자 '3'에 대응하는 데이터가 자신이 관리하는 데이터임을 알 수 있게 된다.

[0032] 다시 말해, 데이터의 이전이 완료된 후에 입력된 쿼리문들(412, 422)에 대해 데이터베이스 서버 1(410)은 결과값을 제공하지 않고, 데이터베이스 서버 2(420)는 결과값을 제공하게 된다. 다시 말해, 잘못된 데이터 조회에 대한 검사가 가능해진다.

[0033] 앞서 도 3 및 도 4의 실시예들에서는 비트맵의 형태로 그룹 식별자들을 관리하는 예를 설명하였으나, 각각의 데이터베이스 서버들이 자신이 관리하는 그룹 식별자들을 파악할 수 있는 구조라면, 이에 제한되지 않을 수 있다. 또한, 앞선 실시예들에서는 샤프키들을 복수의 그룹으로 분류함에 있어서 해시 함수를 사용하는 예를 들었으나, 샤프키들을 복수의 그룹으로 분류할 수 있는 방법이라면, 이에 제한되지 않을 수 있다. 예를 들어, 샤프키들 값의 순서에 따라 순차적으로 복수의 그룹들에 할당할 수도 있다. 보다 자세한 예로, 샤프키들의 값이 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}이고, 복수의 그룹들의 그룹 식별자들이 {a, b, c}라면, 샤프키 1, 4, 7, 10을 그룹 a로, 샤프키 2, 5, 8를 그룹 b로, 샤프키 3, 6, 9를 그룹 c로, 각각 분류할 수 있다.

[0034] 도 5는 본 발명의 일실시예에 있어서, 컴퓨터 장치의 내부 구성을 설명하기 위한 블록도이다. 컴퓨터 장치(500)는 메모리(510), 프로세서(520), 통신 인터페이스(530) 그리고 입출력 인터페이스(540)를 포함할 수 있다.

메모리(510)는 컴퓨터에서 판독 가능한 기록매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 여기서 ROM과 디스크 드라이브와 같은 비소멸성 대용량 기록장치는 메모리(510)와는 구분되는 별도의 영구 저장 장치로서 컴퓨터 장치(500)에 포함될 수도 있다. 또한, 메모리(510)에는 운영체제와 적어도 하나의 프로그램 코드가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 메모리(510)와는 별도의 컴퓨터에서 판독 가능한 기록매체로부터 메모리(510)로 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록매체를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록매체가 아닌 통신 인터페이스(530)를 통해 메모리(510)에 로딩될 수도 있다. 예를 들어, 소프트웨어 구성요소들은 네트워크(560)를 통해 수신되는 파일들에 의해 설치되는 컴퓨터 프로그램에 기반하여 컴퓨터 장치(500)의 메모리(510)에 로딩될 수 있다.

[0035] 프로세서(520)는 기본적인 산술, 로직 및 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(510) 또는 통신 인터페이스(530)에 의해 프로세서(520)로 제공될 수 있다. 예를 들어 프로세서(520)는 메모리(510)와 같은 기록 장치에 저장된 프로그램 코드에 따라 수신되는 명령을 실행하도록 구성될 수 있다.

[0036] 통신 인터페이스(530)는 네트워크(560)를 통해 컴퓨터 장치(500)가 다른 장치(일례로, 앞서 설명한 저장 장치들)와 서로 통신하기 위한 기능을 제공할 수 있다. 일례로, 컴퓨터 장치(500)의 프로세서(520)가 메모리(510)와 같은 기록 장치에 저장된 프로그램 코드에 따라 생성한 요청이나 명령, 데이터, 파일 등이 통신 인터페이스(530)의 제어에 따라 네트워크(560)를 통해 다른 장치들로 전달될 수 있다. 역으로, 다른 장치로부터의 신호나 명령, 데이터, 파일 등이 네트워크(560)를 거쳐 컴퓨터 장치(500)의 통신 인터페이스(530)를 통해 컴퓨터 장치(500)로 수신될 수 있다. 통신 인터페이스(530)를 통해 수신된 신호나 명령, 데이터 등은 프로세서(520)나 메모리(510)로 전달될 수 있고, 파일 등은 컴퓨터 장치(500)가 더 포함할 수 있는 저장 매체(상술한 영구 저장 장치)로 저장될 수 있다.

[0037] 입출력 인터페이스(540)는 입출력 장치(550)와의 인터페이스를 위한 수단일 수 있다. 예를 들어, 입력 장치는 마이크, 키보드 또는 마우스 등의 장치를, 그리고 출력 장치는 디스플레이, 스피커와 같은 장치를 포함할 수 있다. 다른 예로 입출력 인터페이스(540)는 터치스크린과 같이 입력과 출력을 위한 기능이 하나로 통합된 장치와의 인터페이스를 위한 수단일 수도 있다. 입출력 장치(550)는 컴퓨터 장치(500)와 하나의 장치로 구성될 수도 있다.

[0038] 또한, 다른 실시예들에서 컴퓨터 장치(500)는 도 5의 구성요소들보다 더 적은 혹은 더 많은 구성요소들을 포함할 수도 있다. 그러나, 대부분의 종래기술적 구성요소들을 명확하게 도시할 필요성은 없다. 예를 들어, 컴퓨터 장치(500)는 상술한 입출력 장치(550) 중 적어도 일부를 포함하도록 구현되거나 또는 트랜시버(transceiver), GPS(Global Positioning System) 모듈, 카메라, 각종 센서, 데이터베이스 등과 같은 다른 구성요소들을 더 포함할 수도 있다.

[0039] 이러한 컴퓨터 장치(500)는 일례로, 도 3 및 도 4를 통해 설명한 데이터베이스 서버들(330, 410, 420)에 대응할 수 있다.

[0040] 도 6은 본 발명의 일실시예에 있어서, 정합성 검사 방법의 예를 도시한 흐름도이다. 본 실시예에 따른 정합성 검사 방법은 앞서 설명한 컴퓨터 장치(500)에 의해 수행될 수 있다. 예를 들어, 컴퓨터 장치(500)의 프로세서(520)는 메모리(510)에 로딩된 운영체제의 코드나 적어도 하나의 프로그램의 코드에 따른 제어 명령(instruction)을 실행하도록 구현될 수 있다. 여기서, 프로세서(520)는 이러한 제어 명령에 따라 컴퓨터 장치(500)가 도 6의 정합성 검사 방법이 포함하는 단계들(610 및 620)을 수행하도록 컴퓨터 장치(500)를 제어할 수 있다.

[0041] 단계(610)에서 컴퓨터 장치(500)는 샤딩(sharding)된 데이터의 샤드키(shardkey)들을 분류하여 생성된 기설정된 수의 복수의 그룹들 각각의 그룹 식별자들과 관련하여, 컴퓨터 장치(500)가 관리하는 샤드키가 분류된 그룹의 그룹 식별자들을 관리할 수 있다. 이때, 컴퓨터 장치(500)는 컴퓨터 장치(500)가 관리하는 샤드키가 분류된 그룹 여부에 대한 정보가 기설정된 수의 복수의 그룹들의 그룹 식별자들 각각에 대해 설정된 그룹 식별자 비트맵을 생성 및 관리할 수 있다.

[0042] 예를 들어, 그룹 식별자는 샤드키들 각각을 기설정된 해시 함수의 파라미터로 입력하여 계산된 해시값을 포함할 수 있으며, 동일한 해시값을 갖는 샤드키들이 동일한 그룹으로 분류될 수 있다. 앞선 도 3 및 도 4의 실시예에

서는 전체 그룹 식별자들에 대해 컴퓨터 장치(500)가 관리하는 그룹 식별자들에 대해서는 '1'이 설정되고, 컴퓨터 장치(500)가 관리하지 않는 그룹 식별자들에 대해서는 '0'이 설정된 그룹 식별자 비트맵을 설명하였다. 다른 실시예에서 컴퓨터 장치(500)는 자신이 관리하는 그룹 식별자들의 집합만을 포함하는 형태로 그룹 식별자들을 관리할 수도 있다.

- [0043] 단계(620)에서 컴퓨터 장치(500)는 요구된 연산과 관련된 샤프키에 대응하여 추출된 그룹 식별자를 이용하여, 요구된 연산과 연계된 데이터의 정합성을 검사할 수 있다. 도 3의 실시예에서는 JDBC(320)가 질의에 파라미터로 포함된 샤프키 'id1'을 해싱하여 그룹 식별자 '3'을 추출 및 제공하는 예를 설명한 바 있다. 컴퓨터 장치(500)는 이처럼 요구된 연산과 관련된 샤프키를 통해 추출되는 그룹 식별자를 이용하여 데이터에 대한 정합성을 검사할 수 있다.
- [0044] 예를 들어, 컴퓨터 장치(500)는 추출된 그룹 식별자가 컴퓨터 장치(500)에서 관리되는 그룹 식별자들에 포함된 경우 정합성이 존재하는 것으로 결정하고, 추출된 그룹 식별자가 컴퓨터 장치(500)에서 관리되는 그룹 식별자들에 포함되지 않는 경우 정합성이 결여된 것으로 결정할 수 있다. 또한, 컴퓨터 장치(500)는 정합성이 결여된 것으로 판단되는 경우 오류 메시지를 이용자나 관리자에게 제공할 수 있다.
- [0045] 도 7은 본 발명의 일실시예에 있어서, 그룹 식별자들을 관리하는 제1 예를 도시한 흐름도이다. 도 7의 단계들(710 및 720)은 도 6을 통해 설명한 단계(610)에 포함되어 수행될 수 있다.
- [0046] 단계(710)에서 컴퓨터 장치(500)는 데이터베이스 서버들간의 데이터의 이동에 따라 제1 그룹 식별자에 대응하는 데이터가 컴퓨터 장치(500)로 이동된 경우, 제1 그룹 식별자가 컴퓨터 장치에서 관리되는 그룹 식별자들에 포함되어 있는지 확인할 수 있다.
- [0047] 단계(720)에서 컴퓨터 장치(500)는 제1 그룹 식별자가 컴퓨터 장치(500)에서 관리되는 그룹 식별자들에 포함되어 있지 않은 경우, 제1 그룹 식별자를 컴퓨터 장치(500)에서 관리되는 그룹 식별자들에 추가하여 더 관리할 수 있다.
- [0048] 이미 설명한 바와 같이 데이터베이스 서버들간의 데이터의 이동은, 적어도 하나의 그룹 식별자에 대응하는 데이터의 단위로 이루어질 수 있다. 따라서 특정한 그룹 식별자에 대응하는 데이터가 컴퓨터 장치(500)로 이동하는 경우, 컴퓨터 장치(500)는 해당 그룹 식별자를 더 관리할 수 있다.
- [0049] 도 8은 본 발명의 일실시예에 있어서, 그룹 식별자를 관리하는 제2 예를 도시한 흐름도이다. 도 8의 단계(810)는 도 6을 통해 설명한 단계(610)에 포함되어 수행될 수 있다.
- [0050] 단계(810)에서 컴퓨터 장치(500)는 데이터베이스 서버들간의 데이터의 이동에 따라 컴퓨터 장치(500)에서 관리하는 제1 그룹 식별자에 대응하는 데이터가 다른 데이터베이스 서버로 이동되는 경우, 컴퓨터 장치(500)에서 관리하는 그룹 식별자들에서 제1 그룹 식별자를 제외하고 관리할 수 있다.
- [0051] 앞서 설명한 바와 같이 데이터베이스 서버들간의 데이터의 이동은, 적어도 하나의 그룹 식별자에 대응하는 데이터의 단위로 이루어질 수 있다. 따라서 특정한 그룹 식별자에 대응하는 데이터가 컴퓨터 장치(500)에서 다른 데이터베이스 서버로 이동하는 경우, 컴퓨터 장치(500)는 해당 그룹 식별자를 제외하고 나머지 그룹 식별자들을 관리할 수 있다.
- [0052] 도 7 및 도 8의 실시예들에서 그룹 식별자 비트맵을 이용하는 경우, 컴퓨터 장치(500)는 제1 그룹 식별자에 대응하는 값을 '1'이나 '0'으로 변경시킬 수 있으며, 이러한 그룹 식별자 비트맵을 통해 특정 그룹 식별자에 대응하는 데이터에 대한 관리 여부를 식별할 수 있다.
- [0053] 이처럼 본 발명의 실시예들에 따르면, 샤프키를 그룹핑한 기설정된 수의 복수의 그룹들에 대한 그룹 식별자를 이용하여 대량의 데이터에 대한 정합성을 검사할 수 있다. 또한, 기설정된 수의 그룹 식별자들에 대해 샤프키들이 해당 그룹 식별자의 그룹에 포함되어 있는지 여부를 검사하는 것만으로 대량의 데이터에 대한 정합성을 검사할 수 있다.
- [0054] 이상에서 설명된 시스템 또는 장치는 하드웨어 구성요소, 소프트웨어 구성요소 또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는

하나 이상의 소프트웨어 어플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소 (processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서 (parallel processor)와 같은, 다른 처리 구성 (processing configuration)도 가능하다.

[0055] 소프트웨어는 컴퓨터 프로그램 (computer program), 코드 (code), 명령 (instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로 (collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소 (component), 물리적 장치, 가상 장치 (virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화 (embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록매체에 저장될 수 있다.

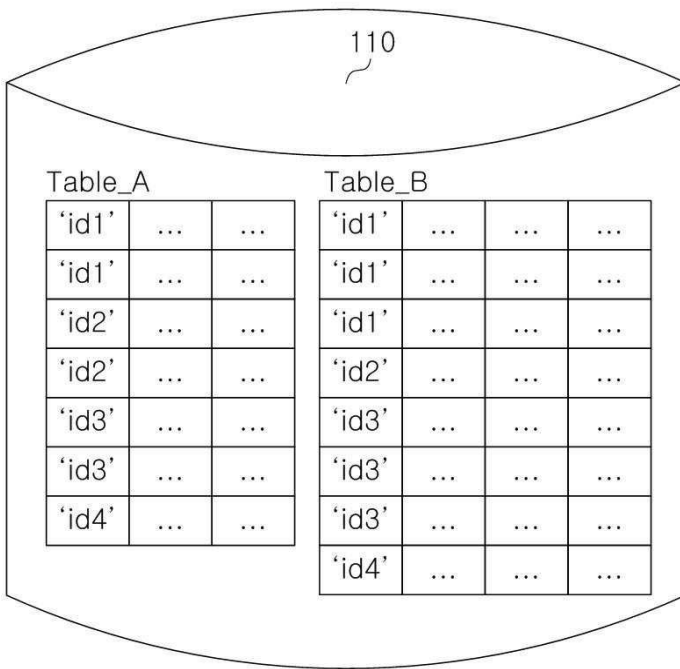
[0056] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 매체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수개 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크 (floptical disk)와 같은 자기-광 매체 (magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 어플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0057] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

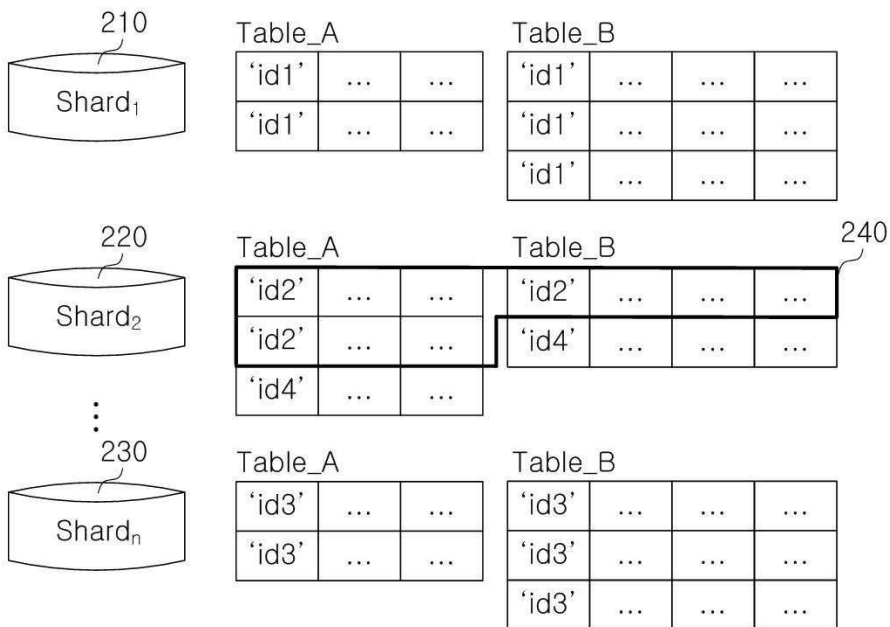
[0058] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

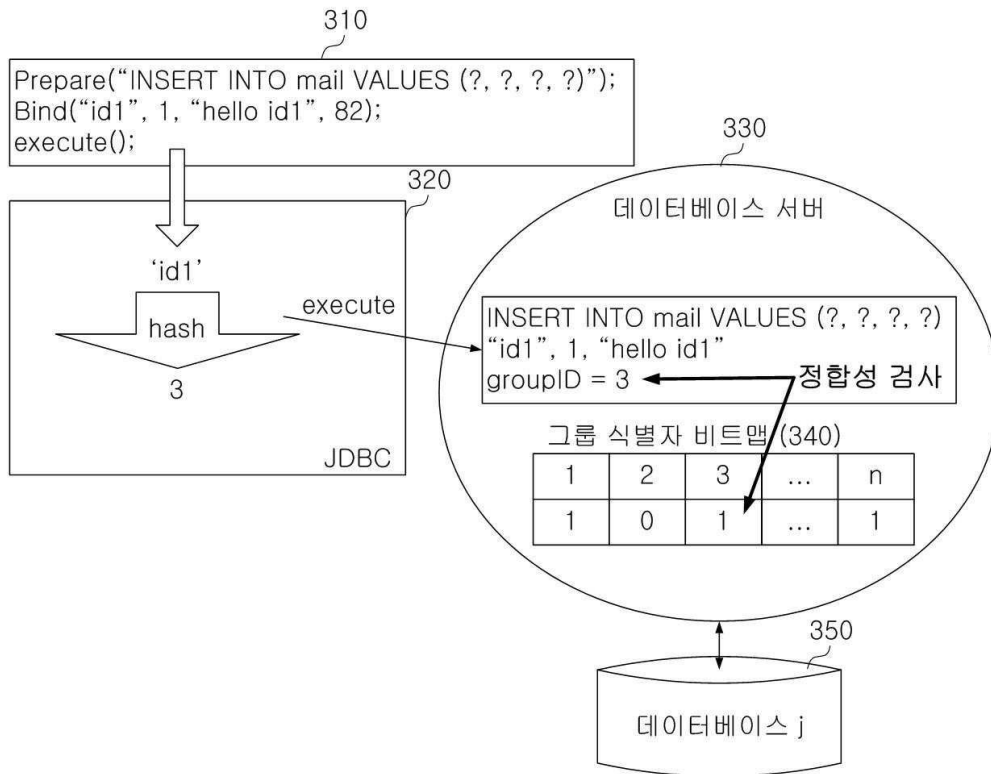
도면1



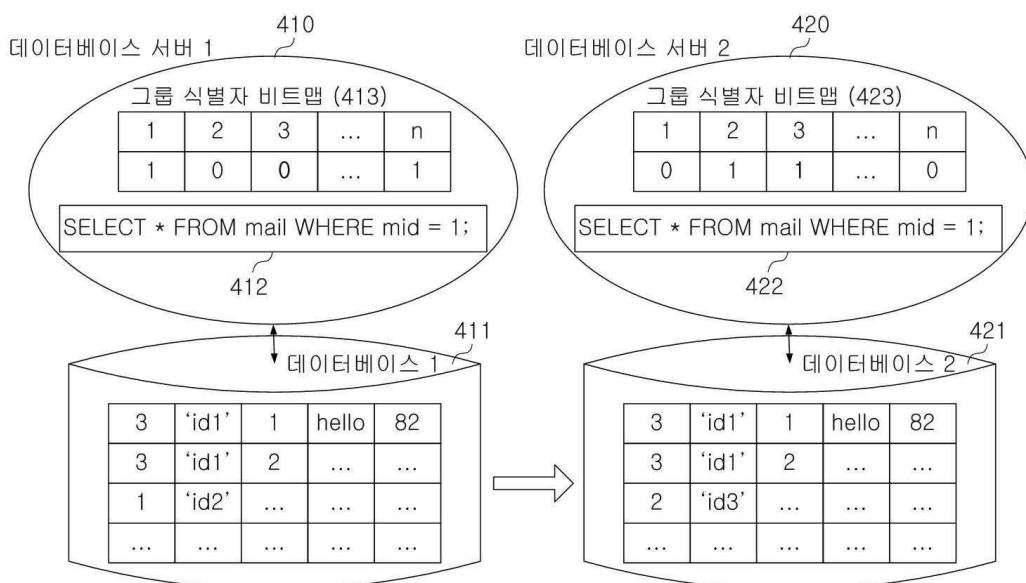
도면2



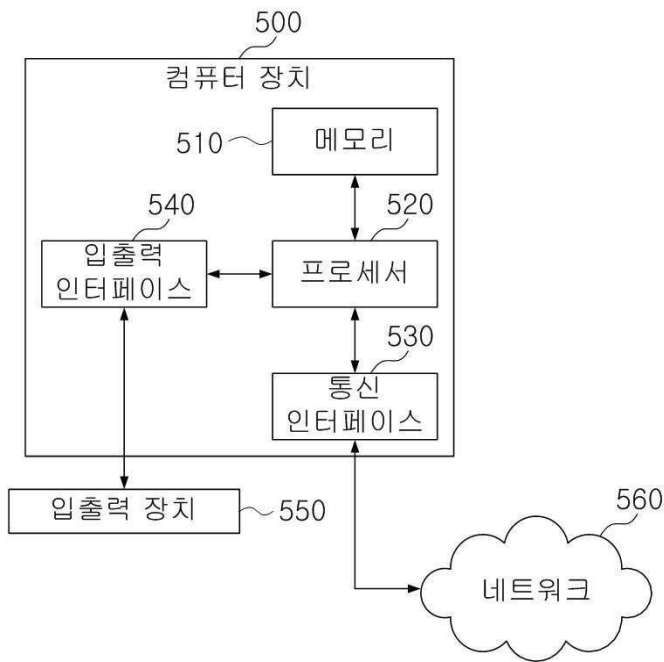
도면3



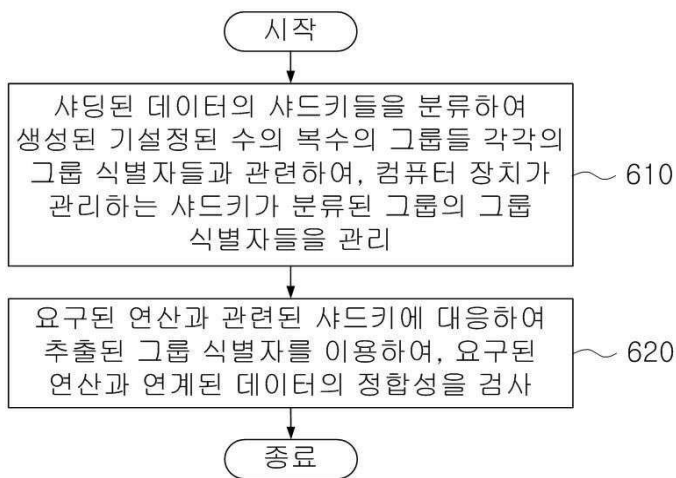
도면4



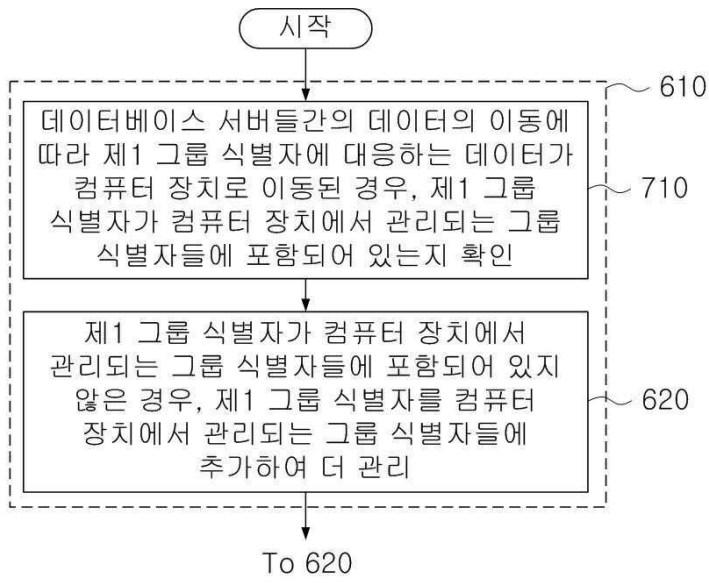
도면5



도면6



도면7



도면8

