



등록특허 10-2153806



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년09월10일
(11) 등록번호 10-2153806
(24) 등록일자 2020년09월02일

- (51) 국제특허분류(Int. Cl.)
G06F 16/00 (2019.01) *G06F 16/21* (2019.01)
G06F 16/22 (2019.01)
- (52) CPC특허분류
G06F 16/2228 (2019.01)
G06F 16/213 (2019.01)
- (21) 출원번호 10-2018-7028270
- (22) 출원일자(국제) 2017년02월20일
심사청구일자 2019년04월30일
- (85) 번역문제출일자 2018년10월01일
- (65) 공개번호 10-2018-0118736
- (43) 공개일자 2018년10월31일
- (86) 국제출원번호 PCT/CN2017/074171
- (87) 국제공개번호 WO 2017/148295
국제공개일자 2017년09월08일

(30) 우선권주장
201610116003.0 2016년03월01일 중국(CN)

(56) 선행기술조사문현
US20150169895 A1
(뒷면에 계속)

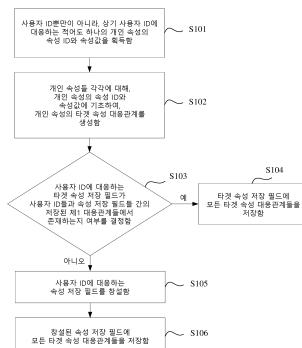
전체 청구항 수 : 총 11 항

심사관 : 박미정

(54) 발명의 명칭 데이터 처리 방법 및 디바이스

(57) 요 약

본 발명개시의 구현예들은 데이터 처리 방법 및 디바이스를 제공한다. 종래 기술에서의 저장 구조는 본 발명개시의 구현예에서 폐기된다. 첫번째로, 비어 있는 필드가 없으며, 이에 따라 비어 있는 필드들을 위해 저장 공간을 낭비하는 문제점을 회피하여, 저장 공간을 절약하는 목적을 달성한다. 또한, 종래의 기술에서는, 각 필드에 하나의 속성값만이 저장된다. 그러나, 본 발명개시의 구현예들에서는, 하나의 속성 저장 필드에 복수의 속성 대응관계들이 저장되고, 각 속성 대응관계는 속성값을 포함한다. 따라서, 본 발명개시의 구현예들에서는, 종래의 기술과 비교하여, 하나의 속성 저장 필드에 복수의 속성값들이 저장되고, 속성 저장 필드에서의 저장 공간이 최대한 활용되어서, 저장 공간을 절약한다.

대 표 도

(52) CPC특허분류

G06F 16/2291 (2019.01)

(56) 선행기술조사문현

KR1020070112219 A

KR1020050077681 A

KR100429142 B1

KR1020110013816 A

CN101046805 A

CN101504672 A

CN102436475 A

US20050223032 A1

명세서

청구범위

청구항 1

데이터 처리 방법에 있어서,

사용자 식별자, 상기 사용자 식별자에 대응하는 적어도 하나의 개인 속성의 속성 식별자와 속성값을 획득하는 단계(S101);

상기 적어도 하나의 개인 속성 각각에 대해, 상기 개인 속성의 상기 속성 식별자와 상기 속성값에 기초하여 상기 개인 속성의 타겟 속성 대응관계를 생성하는 단계(S102);

상기 사용자 식별자에 대응하는 타겟 속성 저장 필드가 사용자 식별자들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부를 결정하는 단계(S103); 및

a) 상기 사용자 식별자에 대응하는 상기 타겟 속성 저장 필드가 상기 제1 대응관계들에서 존재하는 경우, 생성된 모든 타겟 속성 대응관계들을 상기 타겟 속성 저장 필드에 저장하는 단계(S104)로서, 상기 생성된 모든 타겟 속성 대응관계들을 상기 타겟 속성 저장 필드에 저장하는 단계(S104)는,

상기 생성된 모든 타겟 속성 대응관계들에 의해 점유되는 저장 공간이 상기 타겟 속성 저장 필드의 유휴 저장 공간 이하인지 여부를 결정하는 단계(S201); 및

i) 상기 생성된 모든 타겟 속성 대응관계들에 의해 점유되는 저장 공간이 상기 타겟 속성 저장 필드의 유휴 저장 공간 이하인 경우, 상기 생성된 모든 타겟 속성 대응관계들을 상기 타겟 속성 저장 필드에 저장하는 단계(S202); 또는

ii) 상기 생성된 모든 타겟 속성 대응관계들에 의해 점유되는 저장 공간이 상기 타겟 속성 저장 필드의 유휴 저장 공간 보다 큰 경우, 상기 사용자 식별자에 대응하는 적어도 하나의 새로운 속성 저장 필드를 창설하고(S203), 상기 생성된 모든 타겟 속성 대응관계들을, 상기 새로운 속성 저장 필드에 저장하거나, 또는 상기 타겟 속성 저장 필드에 그리고 상기 새로운 속성 저장 필드에 저장하는 단계(S204)

를 포함하는 것인, 단계(S104); 또는

b) 상기 사용자 식별자에 대응하는 타겟 속성 저장 필드가 상기 제1 대응관계들에서 존재하지 않는 경우, 상기 사용자 식별자에 대응하는 속성 저장 필드를 창설하고(S105), 생성된 모든 타겟 속성 대응관계들을 상기 창설된 속성 저장 필드에 저장하는 단계(S106)

를 포함하는 데이터 처리 방법.

청구항 2

제1항에 있어서,

상기 개인 속성의 상기 속성 식별자와 상기 속성값에 기초하여 상기 개인 속성의 타겟 속성 대응관계를 생성하는 단계는,

상기 개인 속성의 상기 속성 식별자와 상기 개인 속성의 상기 속성값 간의 타겟 속성 대응관계를 구축하는 단계

를 포함한 것인 데이터 처리 방법.

청구항 3

제1항에 있어서,

상기 개인 속성의 상기 속성 식별자와 상기 속성값에 기초하여 상기 개인 속성의 타겟 속성 대응관계를 생성하는 단계는,

속성 식별자들과 시퀀스 인덱스들의 인덱스 식별자들 간의 저장된 제2 대응관계들을 획득하는 단계;
 상기 제2 대응관계들에서 상기 속성 식별자에 대응하는 인덱스 식별자를 검색하는 단계; 및
 상기 인덱스 식별자와 상기 개인 속성의 상기 속성값 간의 타겟 속성 대응관계를 구축하는 단계
 를 포함한 것인 데이터 처리 방법.

청구항 4

제2항에 있어서,
 상기 생성된 모든 타겟 속성 대응관계들을 상기 타겟 속성 저장 필드에 저장하는 단계는,

상기 타겟 속성 대응관계에서의 인덱스 식별자 및 상기 타겟 속성 저장 필드에 저장된 모든 속성 대응
 관계들에서의 인덱스 식별자들에 기초하여, 상기 타겟 속성 대응관계가 상기 타겟 속성 저장 필드에서 저장될
 위치를 결정하는 단계(S301); 및

상기 타겟 속성 대응관계를 상기 결정된 위치에 저장하는 단계(S302)
 를 포함한 것인 데이터 처리 방법.

청구항 5

제4항에 있어서,
 상기 타겟 속성 대응관계가 상기 타겟 속성 저장 필드에서 저장될 위치를 결정하는 단계는,

상기 타겟 속성 대응관계에서의 인덱스 식별자와 상기 저장된 속성 대응관계들에서의 인덱스 식별자들
 에 기초하여, 상기 저장된 속성 대응관계들을 정렬(sort)시키는 단계

를 포함한 것인 데이터 처리 방법.

청구항 6

제4항에 있어서,
 상기 방법은,

인덱스 식별자를 포함하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 상
 기 타겟 속성 저장 필드를 검색하는 단계
 를 더 포함하고,

상기 타겟 속성 저장 필드를 검색하는 단계는,

상기 인덱스 식별자를, 상기 타겟 속성 저장 필드에서의 하나 이상의 미리 결정된 위치에 있는 속성 대응관계들
 에서의 인덱스 식별자들과 비교하는 단계;

비교 결과에 기초하여, 상기 타겟 속성 저장 필드에서의 상기 인덱스 식별자를 포함하는 속성 대응관계의 대략
 적인 위치를 결정하는 단계; 및

상기 인덱스 식별자를 포함하는 속성 대응관계가 존재하는지 여부를 결정하기 위해, 상기 인덱스 식별자를, 상
 기 대략적인 위치에 있는 하나 이상의 속성 대응관계에서의 인덱스 식별자들과 비교하는 단계

를 포함한 것인 데이터 처리 방법.

청구항 7

제1항 내지 제6항 중 어느 한 항에 있어서,
 상기 방법은,

사용자에게 특정 서비스를 제공하기 위해 상기 사용자의 하나 이상의 개인 속성의 속성값을 질의하는
 단계

를 더 포함한 것인 데이터 처리 방법.

청구항 8

제1항 내지 제6항 중 어느 한 항에 있어서,

상기 방법은,

상기 사용자 식별자에 대응하는 상기 타겟 속성 저장 필드가 상기 제1 대응관계들에서 존재하는 경우, 상기 속성 식별자에 대응하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 상기 타겟 속성 저장 필드를 검색하는 단계; 및

상기 속성 식별자에 대응하는 속성 대응관계가 존재하는 경우, 상기 속성 대응관계에서의 속성값을 획득하는 단계

를 더 포함한 것인 데이터 처리 방법.

청구항 9

제8항에 있어서,

상기 속성 식별자에 대응하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 상기 타겟 속성 저장 필드를 검색하는 단계는,

상기 속성 식별자를 포함하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 상기 타겟 속성 저장 필드를 검색하는 단계; 및

상기 속성 식별자를 포함하는 속성 대응관계가 존재하는 경우, 상기 속성 식별자에 대응하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재한다고 결정하는 단계

를 포함한 것인 데이터 처리 방법.

청구항 10

제7항에 있어서,

상기 속성 식별자에 대응하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 상기 타겟 속성 저장 필드를 검색하는 단계는,

속성 식별자들과 인덱스 식별자들 간의 제2 대응관계들을 획득하는 단계;

상기 제2 대응관계들에서 상기 속성 식별자에 대응하는 인덱스 식별자를 검색하는 단계;

상기 인덱스 식별자를 포함하는 속성 대응관계가 상기 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 상기 타겟 속성 저장 필드를 검색하는 단계; 및

상기 인덱스 식별자를 포함하는 속성 대응관계가 존재하는 경우, 상기 속성 식별자에 대응하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재한다고 결정하는 단계

를 포함한 것인 데이터 처리 방법.

청구항 11

제1항 내지 제6항 중 어느 한 항의 방법에 따른 동작들을 수행하도록 구성된 복수의 모듈들을 포함하는 데이터 처리 디바이스.

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

발명의 설명

기술 분야

[0001] 본 출원은 인터넷 기술 분야에 관한 것이며, 특히 데이터 처리 방법 및 디바이스에 관한 것이다.

배경 기술

[0002] 인터넷 기술의 급속한 발전에 따라, 사람들은 인터넷 상에서 다양하고 편리한 서비스들을 즐기려는 경향이 커지고 있다. 서비스 공급자에 있어서, 사용자에게 특정 분야의 정확한 서비스들을 제공하는 방법은 사용자의 인지도를 획득하는 중요한 요소이다. 사용자들의 개개별 요구를 충족시키고 사용자들에게 정확한 서비스들을 제공하기 위해서는, 사용자가 자신의 개인 속성에 따라 정확한 서비스들을 제공받을 수 있도록 사용자의 개인 속성을 수집하고 이러한 속성을 저장할 필요가 있다.

[0003] 개인 속성은 사용자의 행동 습관, 관심 선호사항, 맞춤화 요구사항 중 일부를 포함한다. 예를 들어, 개인 속성은 사용자의 선호 결제 방법, 사용자의 선호 결제 계좌 번호, 사용자가 관심을 둔 제품의 카테고리, 사용자가 자주 방문하는 사업, 및 사용자 자신의 디폴트 로그인 계좌 번호를 포함한다.

[0004] 사용자의 개인 속성은 개인 속성의 속성 명칭과 개인 속성의 속성값을 포함한다. 예를 들어, 사용자의 개인 속성의 속성 명칭은 "디폴트 결제 방법"이고, 속성값은 "온라인 결제"이다. 다른 예시로서, 사용자의 다른 개인 속성의 속성 명칭은 "관심 제품의 카테고리"이고, 속성값은 "디지털 가전제품"이다.

[0005] 현재, 기존 기술에 있어서, 사용자의 개인 속성은 일반적으로 아래 [표 1]에서 도시된 방법으로 저장된다. [표 1]의 가장 좌측 열에 있는 각 필드는 각 사용자의 ID(), 예컨대, 사용자 A에서 사용자 M까지를 각각 저장하며, [표 1]의 맨 윗 행의 각 필드는 기술자에 의해 미리 수집된 각 개인 속성의 속성 명칭, 예를 들어, 속성 명칭 1 내지 속성 명칭 N이 각각 저장되어 있다. [표 1]은 총 $(M+1)*(N+1)$ 개 필드들을 포함한다.

[0006] 특정 사용자의 개인 속성의 속성값을 [표 1]에 저장할 필요가 있는 경우, 사용자의 사용자 ID가 위치한 행과 개인 속성의 속성 명칭이 위치한 열을 [표 1]에서 검색한다. 그 후, 찾아낸 행과 찾아낸 열 모두에 위치해 있는 필드가 결정되고, 이 필드에 개인 속성의 속성값을 채운다.

[0007] [표 1]에서, 사용자 ID가 위치한 행과 속성 명칭이 위치한 열의 필드가 비어 있는 필드인 경우, 이것은 사용자 ID에 대응하는 사용자가 속성 명칭에 대응하는 개인 속성을 갖고 있지 않다는 것을 나타낸다.

표 1

	속성 명칭 1	속성 명칭 2	속성 명칭 3	...	속성 명칭 N
사용자 A	A1	A2			AN
사용자 B		B2	B3		
...					
사용자 M	M1		M3		

[0008]

[0009] 본 출원을 구현하는 과정에서, 본 발명자들은 적어도 다음과 같은 단점들이 관련 기술의 해결책에서 존재한다는 것을 알아냈다:

[0010] 기술자들이 미리 수집한 개인 속성들은 다양하게 있으며, 이에 따라, [표 1]의 첫번째 행에 많은 필드들이 포함될 것이지만, 때때로 상이한 사용자들의 개인 속성들은 차이가 크고 일치성은 낮으며, 각 사용자의 개인 속성들

은 기술자들에 의해 미리 수집된 모든 개인 속성들 중 소수 부분만을 차지한다. 결과적으로, 사용자 ID에 대응하는 행의 극소수 필드만이 속성값들로 채워질 수 있고, 다른 필드들은 모두 비어 있는 필드들이다. 따라서, [표 1]에는 비어 있는 필드들이 많을 것이며, 비어 있는 필드들은 일정량의 저장 공간을 차지하기 때문에, [표 1]의 많은 비어 있는 필드들은 많은 양의 저장 공간을 차지할 것이다.

발명의 내용

해결하려는 과제

과제의 해결 수단

- [0011] 관련 기술에서의 문제점을 완화시키기 위해, 본 발명개시는 데이터 처리 방법 및 디바이스를 제공한다.
- [0012] 본 발명개시의 구현예들의 제1 양태에 따르면, 데이터 처리 방법이 제공되며, 본 방법은, 사용자 식별자뿐만 아니라, 상기 사용자 식별자에 대응하는 적어도 하나의 개인 속성의 속성 식별자와 속성값을 획득하는 단계; 개인 속성들 각각에 대해, 개인 속성의 속성 식별자와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계를 생성하는 단계; 사용자 식별자에 대응하는 타겟 속성 저장 필드가 사용자 식별자들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부를 결정하는 단계; 사용자 식별자에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하는 경우, 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장하는 단계; 또는 사용자 식별자에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하지 않은 경우, 사용자 식별자에 대응하는 속성 저장 필드를 창설하고, 창설된 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장하는 단계를 포함한다.
- [0013] 개인 속성의 속성 식별자와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계를 생성하는 단계는, 개인 속성의 속성 식별자와 개인 속성의 속성값 간의 타겟 속성 대응관계를 구축하는 단계; 또는 속성 식별자들과 시퀀스 인덱스들의 인덱스 식별자들 간의 저장된 제2 대응관계들을 획득하는 단계; 제2 대응관계들에서 속성 식별자에 대응하는 인덱스 식별자를 검색하는 단계; 및 인덱스 식별자와 개인 속성의 속성값 간의 타겟 속성 대응관계를 구축하는 단계를 포함한다.
- [0014] 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장하는 단계는, 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은지 여부를 결정하는 단계; 및 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은 경우, 모든 타겟 속성 대응관계들을 타겟 속성 저장 필드에 저장하는 단계; 또는 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 큰 경우, 사용자 식별자에 대응하는 적어도 하나의 새로운 속성 저장 필드를 창설하는 단계; 모든 타겟 속성 대응관계들을, 새로운 속성 저장 필드에 또는 타겟 속성 저장 필드에 그리고 새로운 속성 저장 필드에 저장하는 단계를 포함한다.
- [0015] 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장하는 단계는, 타겟 속성 대응관계에서의 인덱스 식별자 및 타겟 속성 저장 필드에 저장된 모든 속성 대응관계들에서의 인덱스 식별자들에 기초하여 타겟 속성 대응관계가 타겟 속성 저장 필드에 저장될 위치를 결정하는 단계; 및 타겟 속성 대응관계를 상기 결정된 위치에 저장하는 단계를 포함한다.
- [0016] 본 발명개시의 구현예들의 제2 양태에 따르면, 데이터 처리 방법이 제공되며, 본 방법은, 사용자 식별자와 개인 속성의 속성 식별자를 포함하는 취득 요청을 획득한 경우, 사용자 식별자에 대응하는 속성 저장 필드가 제1 대응관계들에서 존재하는지 여부를 결정하기 위해 사용자 식별자들과 속성 저장 필드들 간의 제1 대응관계들을 검색하는 단계; 사용자 식별자에 대응하는 속성 저장 필드가 존재하는 경우, 속성 식별자에 대응하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하는 단계; 및 속성 식별자에 대응하는 속성 대응관계가 존재하는 경우, 속성 대응관계에서 속성값을 획득하는 단계를 포함한다.
- [0017] 속성 식별자에 대응하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하는 단계는, 속성 식별자를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하는 단계; 및 속성 식별자를 포함하는 속성 대응관계가 존재하는 경우, 속성 식별자에 대응하는 속성 대응관계가 속성 저장 필드에서 존재한다고 결정하는 단계를 포함한다.
- [0018] 속성 식별자에 대응하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드

를 검색하는 단계는, 속성 식별자들과 인덱스 식별자들 간의 제2 대응관계들을 획득하는 단계; 제2 대응관계들에서 속성 식별자에 대응하는 인덱스 식별자를 검색하는 단계; 인덱스 식별자를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하는 단계; 및 인덱스 식별자를 포함하는 속성 대응관계가 존재하는 경우, 속성 식별자에 대응하는 속성 대응관계가 속성 저장 필드에서 존재한다고 결정하는 단계를 포함한다.

[0019] 본 발명개시의 구현예들의 제3 양태에 따르면, 데이터 처리 디바이스가 제공되며, 본 디바이스는, 사용자 식별자뿐만이 아니라, 상기 사용자 식별자에 대응하는 적어도 하나의 개인 속성의 속성 식별자와 속성값을 획득하도록 구성된 제1 획득 모듈; 개인 속성을 각각에 대해, 개인 속성의 속성 식별자와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계를 생성하도록 구성된 생성 모듈; 사용자 식별자에 대응하는 타겟 속성 저장 필드가 사용자 식별자들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부를 결정하도록 구성된 결정 모듈; 사용자 식별자에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하는 경우, 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장하도록 구성된 제1 저장 모듈; 및 사용자 식별자에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하지 않는 경우, 사용자 식별자에 대응하는 속성 저장 필드를 창설하도록 구성된 창설 모듈; 및 창설된 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장하도록 구성된 제2 저장 모듈을 포함한다.

[0020] 생성 모듈은, 개인 속성의 속성 식별자와 개인 속성의 속성값 간의 타겟 속성 대응관계를 구축하도록 구성된 제1 구축 유닛; 또는 속성 식별자들과 시퀀스 인덱스들의 인덱스 식별자들 간의 저장된 제2 대응관계들을 획득하도록 구성된 제1 획득 유닛; 제2 대응관계들에서 속성 식별자에 대응하는 인덱스 식별자를 검색하도록 구성된 검색 유닛; 및 인덱스 식별자와 개인 속성의 속성값 간의 타겟 속성 대응관계를 구축하도록 구성된 제2 구축 유닛을 포함한다.

[0021] 제1 저장 모듈은, 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은지 여부를 결정하도록 구성된 결정 유닛; 및 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은 경우, 모든 타겟 속성 대응관계들을 타겟 속성 저장 필드에 저장하도록 구성된 제1 저장 유닛; 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 큰 경우, 사용자 식별자에 대응하는 적어도 하나의 새로운 속성 저장 필드를 창설하도록 구성된 창설 유닛; 및 모든 타겟 속성 대응관계들을, 새로운 속성 저장 필드에 또는 타겟 속성 저장 필드에 그리고 새로운 속성 저장 필드에 저장하도록 구성된 제2 저장 유닛을 포함한다.

[0022] 제1 저장 모듈은, 타겟 속성 대응관계에서의 인덱스 식별자 및 타겟 속성 저장 필드에 저장된 모든 속성 대응관계들에서의 인덱스 식별자들에 기초하여 타겟 속성 대응관계가 타겟 속성 저장 필드에 저장될 위치를 결정하도록 구성된 제1 결정 유닛; 및 타겟 속성 대응관계를 상기 결정된 위치에 저장하도록 구성된 제3 저장 유닛을 포함한다.

[0023] 본 발명개시의 구현예들의 제4 양태에 따르면, 데이터 처리 디바이스가 제공되며, 사용자 식별자와 개인 속성의 속성 식별자를 포함하는 취득 요청이 획득된 경우, 사용자 식별자에 대응하는 속성 저장 필드가 제1 대응관계들에서 존재하는지 여부를 결정하기 위해 사용자 식별자들과 속성 저장 필드들 간의 제1 대응관계를 검색하도록 구성된 제1 검색 모듈; 사용자 식별자에 대응하는 속성 저장 필드가 존재하는 경우, 속성 식별자에 대응하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하도록 구성된 제2 검색 모듈; 및 속성 식별자에 대응하는 속성 대응관계가 존재하는 경우, 속성값을 속성 대응관계에서 획득하도록 구성된 제2 획득 모듈을 포함한다.

[0024] 제2 검색 모듈은, 속성 식별자를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하도록 구성된 제1 검색 유닛; 및 속성 식별자를 포함하는 속성 대응관계가 존재하는 경우, 속성 식별자에 대응하는 속성 대응관계가 속성 저장 필드에서 존재한다고 결정하도록 구성된 제2 결정 유닛을 포함한다.

[0025] 제2 검색 모듈은, 속성 식별자들과 인덱스 식별자들 간의 제2 대응관계들을 획득하도록 구성된 제2 획득 유닛; 제2 대응관계들에서 속성 식별자에 대응하는 인덱스 식별자를 검색하도록 구성된 제2 검색 유닛; 속성 식별자를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하도록 구성된 제3 검색 유닛; 및 인덱스 식별자를 포함하는 속성 대응관계가 존재하는 경우, 속성 식별자에 대응하는 속성 대응관계가 속성 저장 필드에서 존재한다고 결정하도록 구성된 제3 결정 유닛을 포함한다.

- [0026] 본 발명개시의 구현예들에서 제공되는 기술적 해결책들은 다음과 같은 유익한 효과를 갖는다:
- [0027] 본 발명개시의 구현예들에서, 사용자 ID뿐만이 아니라, 상기 사용자 ID에 대응하는 적어도 하나의 개인 속성의 속성 식별자 ID와 속성값이 획득되고; 개인 속성들 각각에 대해, 개인 속성의 속성 ID와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계가 생성되고; 사용자 ID에 대응하는 타겟 속성 저장 필드가 사용자 ID들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부가 결정되며; 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하는 경우, 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들이 저장되며; 대안적으로, 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하지 않는 경우, 사용자 ID에 대응하는 속성 저장 필드가 창설되고, 창설된 속성 저장 필드에 모든 타겟 속성 대응관계들이 저장된다.
- [0028] 종래의 기술에서의 저장 구조는 본 발명개시의 구현예에서 폐기된다. 첫번째로, 비어 있는 필드가 없으며, 이에 따라 비어 있는 필드들을 위해 저장 공간을 낭비하는 문제점을 회피하여, 저장 공간을 절약하는 목적을 달성한다. 또한, 종래의 기술에서는, 각 필드에 하나의 속성값만이 저장된다. 그러나, 본 발명개시의 구현예들에서는, 하나의 속성 저장 필드에 복수의 속성 대응관계들이 저장되고, 각 속성 대응관계는 속성값을 포함한다. 따라서, 본 발명개시의 구현예들에서는, 종래의 기술과 비교하여, 하나의 속성 저장 필드에 복수의 속성값들이 저장되고, 속성 저장 필드에서의 저장 공간이 최대한 활용되어서, 저장 공간을 절약한다.
- [0029] 또한, 서비스 사업의 급속한 발전에 따라, 사용자의 개인화된 요구는 무궁무진한 다양성으로 표출된다. 따라서, 종종 새로운 개인 속성들을 발굴해낼 필요가 있다. 이 경우, 종래의 기술에서는 [표 1]에서 첫번째 행의 맨 오른쪽에 필드가 추가될 필요가 있으며, 새로운 개인 속성의 속성 명칭이 새롭게 추가된 필드에 추가된다. 그 후, [표 1]에서의 각 사용자 ID의 행의 맨 오른쪽에 새로운 필드가 추가될 필요가 있어서, 사용자가 새로운 개인 속성을 갖는 경우, 사용자 ID가 위치해 있는 행의 맨 오른쪽에 새롭게 추가된 필드가 개인 속성의 속성값으로 채워지도록 한다.
- [0030] 그러나, 이 경우, [표 1]에서의 사용자 ID에 대응하는 사용자들 모두가 새로운 개인 속성을 갖는 것은 아니고, 이에 따라 모든 사용자 ID들이 위치해 있는 행의 맨 오른쪽에 새롭게 추가된 필드들이 속성값들로 채워지지는 않을 것인데, 이로 인해, 일부 비어 있는 필드들을 야기시키고, 저장 공간을 다소 낭비시킨다.
- [0031] 또한, [표 1]에 새로운 필드들을 추가하는 과정에서는, [표 1]을 잡그는 것이 필요하며, 잡긴 상태의 [표 1]은 사용할 수 없다. [표 1]에 매우 많은 수의 사용자 ID들이 저장되어 있는 경우, 즉, [표 1]에 많은 수의 행들이 있는 경우, 모든 사용자 ID들이 위치해 있는 행의 맨 오른쪽에 새로운 필드들을 추가하는 것은 많은 시간이 소요될 것이다. 결과적으로, [표 1]을 너무 오랜 시간동안 사용할 수 없게 되므로, 사용자에게 장기간에 걸쳐 정확한 서비스를 제공하는 것이 불가능해진다. 그러나, 본 발명개시의 구현예에서는, 새로운 개인 속성이 개발되더라도, 개발된 새로운 개인 속성을 위한 필드를 할당할 필요가 없고, 저장된 개인 속성을 잡을 필요가 없어서, 사용자에게 지속적으로 정확한 서비스를 제공할 수 있다. 또한, 개발된 새로운 개인 속성들을 위한 필드들을 할당할 필요가 없고, 이에 따라 비어 있는 필드가 없으므로, 저장 공간을 낭비하는 문제점을 완화시킬 수 있다.
- [0032] 전술한 일반적인 설명 및 아래의 상세한 설명은 단지 예시일 뿐이며, 본 발명개시에 제한을 두지 않는 것으로 이해되어야 한다.

도면의 간단한 설명

- [0033] 본 명세서에서 병합되어 있고 본 명세서의 일부를 구성하는 첨부 도면들은 본 발명개시와 일치하는 구현예들을 도시하며, 이러한 설명과 함께, 본 발명개시의 원리들을 설명하는 역할을 한다.
- 도 1은 구현예에 따른 데이터 처리 방법을 도시하는 흐름도이다.
- 도 2는 구현예에 따른 데이터 처리 방법을 도시하는 흐름도이다.
- 도 3은 구현예에 따른 데이터 처리 방법을 도시하는 흐름도이다.
- 도 4는 구현예에 따른 데이터 처리 방법을 도시하는 흐름도이다.
- 도 5는 구현예에 따른 데이터 처리 디바이스를 도시하는 블록도이다.
- 도 6은 구현예에 따른 데이터 처리 디바이스를 도시하는 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0034] 여기서는 구현예들을 상세히 설명하며, 이 구현예들의 예시들은 첨부 도면에서 제시된다. 아래의 설명이 첨부 도면을 참조할 때, 달리 명시되지 않는 한, 상이한 첨부 도면들에서 동일한 번호는 동일하거나 유사한 엘리먼트들을 나타낸다. 이하에서 설명되는 구현예들은 본 발명개시와 일치하는 모든 구현예들을 나타내는 것은 아니다. 반대로, 구현예들은 첨부된 청구범위에서 상세히 설명되고 본 발명개시의 일부 양태들와 일치하는 디바이스 및 방법의 단지 예시일 뿐이다.
- [0035] 도 1은 구현예에 따른 데이터 처리 방법을 도시하는 흐름도이다. 도 1에서 도시된 바와 같이, 본 방법은 다음의 단계들을 포함한다.
- [0036] 단계 S101: 사용자 식별자(ID)뿐만이 아니라, 상기 사용자 ID에 대응하는 적어도 하나의 개인 속성의 속성 식별자(ID)와 속성값을 획득한다.
- [0037] 사용자 ID는 사용자의 계좌 번호일 수 있다. 예를 들어, 사용자는 사용자 계좌를 미리 서버에 등록할 수 있으며, 사용자 ID는 사용자 계좌의 계좌 번호일 수 있다.
- [0038] 개인 속성의 속성 ID는 개인 속성의 명칭, 예를 들어 "디폴트 결제 방법" 및 "디폴트 결제 계좌 번호" 등일 수 있다.
- [0039] 개인 속성의 명칭이 "디폴트 결제 방법"인 경우, 개인 속성의 속성값은 "온라인 결제" 또는 "후불제(cash on delivery)"일 수 있다.
- [0040] 개인 속성의 명칭이 "디폴트 결제 계좌 번호"인 경우, 개인 속성의 속성값은 계좌 번호, 예를 들어 "273356214" 또는 "178549624"일 수 있다.
- [0041] 개인 속성의 명칭이 "관심 상품의 카테고리"인 경우, 개인 속성의 속성값은 "디지털 가전제품", "음식", "신발/가방" 등일 수 있다.
- [0042] 본 발명개시의 구현예에서, 서버가 특정 사용자의 개인 속성들을 처음으로 저장할 때, 서버는 사용자의 사용자 ID, 사용자의 각 개인 속성의 속성 ID와 속성값을 사용자의 사용자 계좌로부터 획득할 수 있다.
- [0043] 그 후, 사용자가 사용자 자신의 단말 디바이스 상에서 사용자 자신의 사용자 ID를 사용하여 서버에 로그인한 후, 사용자가 사용자 자신의 계좌에서 하나 이상의 기준의 개인 속성의 속성 ID에 대응하는 속성값을 업데이트한 경우, 단말 디바이스는 사용자의 사용자 ID, 하나 이상의 업데이트된 개인 속성의 속성 ID, 및 업데이트된 하나 이상의 개인 속성의 각 속성 ID에 대응하는 속성값을 획득하고, 그 후, 획득된 사용자 ID, 속성 ID, 및 속성값을 서버에 전송한다. 서버는 단말 디바이스에 의해 전송된 사용자 ID, 속성 ID, 및 속성값을 수신한다.
- [0044] 사용자가 사용자 자신의 단말 디바이스 상에서 자신의 사용자 ID를 사용하여 서버에 로그인한 후, 사용자가 하나 이상의 새로운 개인 속성의 속성 ID와 새로운 각 속성 ID에 대응하는 속성값을 사용자 자신의 계좌에 추가하는 경우, 단말 디바이스는 사용자의 사용자 ID, 새로운 각 개인 속성의 속성 ID, 및 새로운 각 개인 속성의 속성 ID에 대응하는 속성값을 획득하고, 그 후, 획득된 사용자 ID, 속성 ID, 및 속성값을 서버에 전송한다. 서버는 단말 디바이스에 의해 전송된 사용자 ID, 속성 ID, 및 속성값을 수신한다.
- [0045] 단계 S102. 개인 속성을 각각에 대해, 개인 속성의 속성 ID와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계를 생성한다.
- [0046] 본 발명개시의 구현예에서, 개인 속성을 각각에 대해, 개인 속성의 속성 ID와 개인 속성의 속성값 간의 타겟 속성 대응관계가 구축될 수 있다. 타겟 속성 대응관계는 개인 속성의 속성 ID 및 개인 속성의 속성값을 포함한다. 개인 속성의 속성 ID 및 개인 속성의 속성값은 키-값 쌍(key-value pair)을 형성할 수 있다. 각 개인 속성에 대해 작업이 실행된다.
- [0047] 키-값 쌍에서, "키"는 개인 속성의 속성 ID를 나타내기 위해 사용되고, "값"은 개인 속성의 속성값을 나타내기 위해 사용되며, 속성 ID와 속성값은 제1 미리 결정된 식별자를 사용하여 연결된다. 제1 미리 결정된 식별자는 "=" , "-" , 또는 "+" 등일 수 있으며, 이들은 본 발명개시에서 한정되지 않는다.
- [0048] 예를 들어, 개인 속성의 속성 ID가 "디폴트 결제 방법"이고, 개인 속성의 속성값이 "온라인 결제"인 것으로 가정하면, 타겟 속성 대응관계는 "디폴트 결제 방법=온라인 결제"일 수 있다.
- [0049] 전술한 구현예에서, 개인 속성의 속성 ID는 일반적으로, 개인 속성의 명칭, 예컨대 "디폴트 결제 방법", "디폴트 결제 계좌 번호", 및 "관심 제품의 카테고리"이다. 그러나, 개인 속성의 명칭에는 일반적으로 많은 문자들이

포함되므로, 개인 속성의 속성 ID는 많은 양의 저장 공간을 점유한다. 속성 대응관계는 개인 속성의 속성 ID를 포함하기 때문에, 속성 ID가 많은 양의 저장 공간을 점유하면, 속성 대응관계는 상대적으로 많은 양의 저장 공간을 점유한다.

[0050] 본 발명개시는 하나의 속성 저장 필드에 가능한 한 많은 속성 대응관계들을 저장하는 것을 목표로 한다. 그러나, 하나의 속성 저장 필드의 총 저장 공간은 제한되어 있다. 따라서, 속성 대응관계가 비교적 큰 저장 공간을 점유하는 경우, 하나의 속성 저장 필드에 저장될 수 있는 속성 대응관계의 갯수는 더 적어진다.

[0051] 하나의 속성 저장 필드에 더 많은 속성 대응관계들을 저장하기 위해, 본 발명개시의 다른 구현예에서는, 인덱스 식별자(ID)가 각각의 개인 속성에 대해 미리 할당될 수 있다. 상이한 인덱스 ID들이 상이한 개인 속성들에 대해 할당된다. 각 개인 속성의 속성 ID에 의해 점유된 저장 공간은 개인 속성의 속성 ID에 대해 할당된 인덱스 ID에 의해 점유된 저장 공간보다 더 작다.

[0052] 그 후, 각 개인 속성의 속성 ID와 개인 속성에 대해 할당된 인덱스 ID가 속성 ID들과 시퀀스 인덱스들의 인덱스 ID들 간의 제2 대응관계들에 저장된다.

[0053] 할당된 모든 인덱스 ID들 간에는 미리 결정된 인덱스 순서가 있다. 예를 들어, 인덱스 ID는 1, 2, 3, 4, 및 5 등과 같은 숫자이며, 이 숫자들은 오름차순으로 정렬된다.

[0054] 이와 같이, 본 발명개시의 다른 구현예에서, 속성 ID와 개인 속성의 속성값에 기초하여 개인 속성의 타겟 속성 대응관계를 생성하는 것은 다음과 같을 수 있다:

[0055] 임의의 개인 속성에 대해, 속성 ID들과 시퀀스 인덱스들의 인덱스 ID들 간의 저장된 제2 대응관계가 획득되고, 제2 대응관계에서 개인 속성의 속성 ID에 대응하는 인덱스 ID를 검색하고, 인덱스 ID와 개인 속성의 속성값 간의 타겟 속성 대응관계가 구축되어, 개인 속성의 속성 ID 및 속성값에 기초하여 개인 속성의 타겟 속성 대응관계가 생성될 수 있도록 한다. 타겟 속성 대응관계는 개인 속성의 속성 ID에 대응하는 인덱스 ID 및 개인 속성의 속성값을 포함한다. 개인 속성의 속성 ID에 대응하는 인덱스 ID 및 개인 속성의 속성값은 키-값 쌍을 형성할 수 있다. 각 개인 속성에 대해 작업이 실행된다.

[0056] 키-값 쌍에서, "키"는 개인 속성의 속성 ID에 대응하는 인덱스 ID를 나타내기 위해 사용되고, "값"은 개인 속성의 속성값을 나타내기 위해 사용되며, 인덱스 ID와 속성값은 제1 미리 결정된 식별자를 사용하여 연결된다. 제1 미리 결정된 식별자는 "=", "-", 또는 "+" 등일 수 있으며, 이들은 본 발명개시에서 한정되지 않는다.

[0057] 예를 들어, 개인 속성의 속성 ID가 "디폴트 결제 방법"이고, 개인 속성의 속성값이 "온라인 결제"이고, 개인 속성의 속성 ID에 대응하는 인덱스 ID가 1이라고 가정하면, 타겟 속성 대응관계는 "1=온라인 결제"일 수 있다.

[0058] 단계 S103: 사용자 ID에 대응하는 타겟 속성 저장 필드가 사용자 ID들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부를 결정한다.

[0059] 서버에 계좌를 등록한 모든 사용자에 대하여, 서버가 사용자의 속성 대응관계를 처음으로 저장할 때, 사용자 ID들과 속성 저장 필드들 간의 제1 대응관계에서 사용자 ID에 대응하는 속성 저장 필드가 구축된다. 그 후, 사용자의 속성 대응관계가 속성 저장 필드에 저장된다. 본 발명개시의 구현예에서, 사용자의 새로운 개인 속성이 나중에 추가되는 경우, 새로운 개인 속성의 속성 ID와 속성값에 기초하여 새로운 개인 속성의 새로운 속성 대응관계가 생성된 후, 새로운 속성 대응관계가 속성 저장 필드에 계속해서 저장된다. 이것은 서버에 계좌를 등록한 다른 모든 사용자들에게도 적용된다.

[0060] 타겟 속성 대응관계가 생성된 후, 사용자 ID에 대응하는 타겟 속성 저장 필드가 사용자 ID들과 속성 저장 필드 간의 저장된 제1 대응관계에서 존재하는지 여부를 결정할 필요가 있다.

[0061] 사용자 ID에 대응하는 타겟 속성 저장 필드가 존재하는 경우, 타겟 속성 대응관계가 타겟 속성 저장 필드에 직접 저장될 수 있다.

[0062] 사용자 ID에 대응하는 타겟 속성 저장 필드가 존재하지 않는 경우, 이것은 사용자의 속성 대응관계가 이전에 저장되어 있지 않다는 것을 나타내며, 즉 사용자 ID에 대응하는 속성 저장 필드가 저장되어 있지 않다는 것을 나타낸다. 이 경우, 사용자 ID에 대응하는 속성 저장 필드가 제1 대응관계들에서 창설될 필요가 있고, 그 후, 창설된 속성 저장 필드에 타겟 속성 대응관계가 저장될 수 있다.

[0063] 단계 S104: 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하는 경우, 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장한다.

- [0064] 단계 S105: 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하지 않는 경우, 사용자 ID에 대응하는 속성 저장 필드를 창설하고, 단계 S106를 실행한다.
- [0065] 단계 S106: 창설된 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장한다.
- [0066] 본 발명개시의 도 1에서 도시된 구현예에서, 사용자 ID뿐만이 아니라, 상기 사용자 ID에 대응하는 적어도 하나의 개인 속성의 속성 ID와 속성값이 획득되고; 개인 속성을 각각에 대해, 개인 속성의 속성 ID와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계가 생성되고; 사용자 ID에 대응하는 타겟 속성 저장 필드가 사용자 ID들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부가 결정되며; 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하는 경우, 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들이 저장되며; 대안적으로, 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하지 않는 경우, 사용자 ID에 대응하는 속성 저장 필드가 창설되고, 창설된 속성 저장 필드에 모든 타겟 속성 대응관계들이 저장된다.
- [0067] 종래의 기술에서의 저장 구조는 본 발명개시의 구현예에서 폐기된다. 첫번째로, 비어 있는 필드가 없으며, 이에 따라 비어 있는 필드들을 위해 저장 공간을 낭비하는 문제점을 회피하여, 저장 공간을 절약하는 목적을 달성한다. 또한, 종래의 기술에서는, 각 필드에 하나의 속성값만이 저장된다. 그러나, 본 발명개시의 구현예들에서는, 하나의 속성 저장 필드에 복수의 속성 대응관계들이 저장되고, 각 속성 대응관계는 속성값을 포함한다. 따라서, 본 발명개시의 구현예들에서는, 종래의 기술과 비교하여, 하나의 속성 저장 필드에 복수의 속성값들이 저장되고, 속성 저장 필드에서의 저장 공간이 최대한 활용되어서, 저장 공간을 절약한다.
- [0068] 또한, 서비스 사업의 급속한 발전에 따라, 사용자의 개인화된 요구는 무궁무진한 다양성으로 표출된다. 따라서, 종종 새로운 개인 속성을 발굴해낼 필요가 있다. 이 경우, 종래의 기술에서는 [표 1]에서 첫번째 행의 맨 오른쪽에 필드가 추가될 필요가 있으며, 새로운 개인 속성의 속성 명칭이 새롭게 추가된 필드에 추가된다. 그 후, [표 1]에서의 각 사용자 ID의 행의 맨 오른쪽에 새로운 필드가 추가될 필요가 있어서, 사용자가 새로운 개인 속성을 갖는 경우, 사용자 ID가 위치해 있는 행의 맨 오른쪽에 새롭게 추가된 필드가 개인 속성의 속성값으로 채워지도록 한다.
- [0069] 그러나, 이 경우, [표 1]에서의 사용자 ID에 대응하는 사용자들 모두가 새로운 개인 속성을 갖는 것은 아니고, 이에 따라 모든 사용자 ID들이 위치해 있는 행의 맨 오른쪽에 새롭게 추가된 필드들이 속성값들로 채워지지는 않을 것인데, 이로 인해, 일부 비어 있는 필드들을 야기시키고, 저장 공간을 다소 낭비시킨다.
- [0070] 또한, [표 1]에 새로운 필드들을 추가하는 과정에서는, [표 1]을 잠그는 것이 필요하며, 잠긴 상태의 [표 1]은 사용할 수 없다. [표 1]에 매우 많은 수의 사용자 ID들이 저장되어 있는 경우, 즉, [표 1]에 많은 수의 행들이 있는 경우, 모든 사용자 ID들이 위치해 있는 행의 맨 오른쪽에 새로운 필드들을 추가하는 것은 많은 시간이 소요될 것이다. 결과적으로, [표 1]을 너무 오랜 시간동안 사용할 수 없게 되므로, 사용자에게 장기간에 걸쳐 정확한 서비스를 제공하는 것은 불가능해진다. 그러나, 본 발명개시의 구현예에서는, 새로운 개인 속성이 개발되더라도, 개발된 새로운 개인 속성을 위한 필드를 할당할 필요가 없고, 저장된 개인 속성을 잡을 필요가 없어서, 사용자에게 지속적으로 정확한 서비스를 제공할 수 있다. 또한, 개발된 새로운 개인 속성들을 위한 필드들을 할당할 필요가 없고, 이에 따라 비어 있는 필드가 없으므로, 저장 공간을 낭비하는 문제점을 완화시킬 수 있다.
- [0071] 본 발명개시의 다른 구현예에서는, 도 2를 참조하면, 단계 S104는 다음을 포함한다.
- [0072] 단계 S201: 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은지 여부를 결정한다.
- [0073] 본 발명개시의 구현예에서, 각각의 문자는 한 개 단위의 저장 공간을 점유하고, 각각의 속성 대응관계는 복수의 문자들을 포함한다. 따라서, 각 속성 대응관계는 일정량의 저장 공간을 점유한다. 속성 저장 필드의 이용가능한 저장 공간은 제한되어 있기 때문에, 속성 대응관계가 속성 저장 필드에 저장될 필요가 있을 경우, 다음과 같은 객관적인 조건이 충족되어야 한다: 속성 대응관계에 의해 점유된 저장 공간이 속성 저장 필드의 이용가능한 저장 공간보다 작거나 또는 같다. 상기 조건이 충족된 경우에만, 속성 대응관계가 속성 저장 필드에 성공적으로 저장될 수 있다. 그렇지 않고, 속성 대응관계에 의해 점유된 저장 공간이 속성 저장 필드의 이용가능한 저장 공간보다 큰 경우, 속성 대응관계는 속성 저장 필드에 성공적으로 저장될 수 없다.
- [0074] 마찬가지로, 복수의 속성 대응관계들이 속성 저장 필드에 저장될 필요가 있는 경우, 복수의 속성 대응관계들에 의해 점유된 저장 공간이 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은지 여부가 또한 결정될 필요가

있다.

[0075] 단계 S202: 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은 경우, 모든 타겟 속성 대응관계들을 타겟 속성 저장 필드에 저장한다.

[0076] 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은 경우, 이것은 타겟 속성 저장 필드가 모든 타겟 속성 대응관계들을 수용할 수 있다는 것을 나타내므로, 모든 타겟 속성 대응관계들이 타겟 속성 저장 필드에 직접 저장될 수 있다.

[0077] 단계 S203: 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 큰 경우, 사용자 ID에 대응하는 적어도 하나의 새로운 속성 저장 필드를 창설하고; 그 후 단계 S204를 실행한다.

[0078] 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 큰 경우, 이것은 타겟 속성 저장 필드가 모든 타겟 속성 대응관계들을 수용할 수 없다는 것을 나타내므로, 사용자 ID에 대응하는 적어도 하나의 새로운 속성 저장 필드가 창설될 필요가 있다.

[0079] 창설될 새로운 속성 저장 필드들의 갯수는 하나의 속성 저장 필드의 이용가능한 저장 공간, 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간, 또는 타겟 속성 필드의 이용가능한 저장 공간에 기초하여 결정될 수 있다. 특정 결정 방법은 본 발명개시에서 제한되지 않는다.

[0080] 단계 S204: 모든 타겟 속성 대응관계들을, 새로운 속성 저장 필드에 또는 타겟 속성 저장 필드에 그리고 새로운 속성 저장 필드에 저장한다.

[0081] 타겟 속성 저장 필드가 완전히 채워진 경우, 타겟 속성 대응관계들은 새로운 속성 저장 필드에 저장된다. 타겟 속성 저장 필드가 완전히 채워지지 않은 경우, 일부 타겟 속성 대응관계들이 먼저 타겟 속성 저장 필드에 저장될 수 있고, 타겟 속성 저장 필드가 완전히 채워진 후, 나머지 타겟 속성 대응관계들은 새로운 속성 저장 공간에 저장될 수 있다.

[0082] 모든 타겟 속성 대응관계들이 저장되는 경우, 상이한 속성 대응관계들이 서로 혼동되는 것을 방지하기 위해 미리 결정된 제2 분리자를 사용하여 인접한 속성 대응관계들을 구분할 수 있다. 미리 결정된 제2 분리자는 "|", "*" 또는 "~" 등일 수 있으며, 이들은 본 발명개시에서 한정되지 않는다.

[0083] 예를 들어, 속성 대응관계가 "디폴트 결제 방법=온라인 결제"이고, 다른 속성 대응관계가 "관심 제품의 카테고리=디지털 가전제품"이고, 또 다른 속성 대응관계가 "디폴트 결제 계좌 번호=123456789"이라고 가정한다. 3개의 속성 대응관계들이 속성 저장 필드에 저장된 경우, 3개의 속성 대응관계들을 구분하기 위해 미리 결정된 제2 분리자 "|"가 사용될 수 있다: "디폴트 결제 방법=온라인 결제 | 관심 제품의 카테고리=디지털 가전제품 | 디폴트 결제 계좌 번호=123456789".

[0084] 이전의 구현예에서, 속성 대응관계는 개인 속성의 속성 ID에 대응하는 인덱스 ID 및 개인 속성의 속성값을 포함하고, 모든 개인 속성들의 인덱스 ID들 간에는 미리 결정된 인덱스 순서가 있다. 따라서, 본 발명개시의 구현예에서는, 이후에 개인 속성의 속성값을 질의하는 효율성을 향상시키기 위해, 저장된 속성 대응관계들에서의 인덱스 ID들 간의 미리 결정된 인덱스 순서와 타겟 속성 대응관계들에 포함된 인덱스 ID들에 기초하여 타겟 속성 대응관계들이 저장될 수 있다.

[0085] 따라서, 본 발명개시의 다른 구현예에서는, 도 3를 참조하면, 단계 S104는 다음을 포함한다:

[0086] 단계 S301: 타겟 속성 대응관계에서의 인덱스 ID 및 타겟 속성 저장 필드에 저장된 모든 속성 대응관계들에서의 인덱스 ID들에 기초하여, 타겟 속성 대응관계가 타겟 속성 저장 필드에 저장될 위치를 결정한다.

[0087] 본 발명개시의 구현예에서 속성 저장 필드에서의 속성 대응관계들의 위치는 속성 대응관계들에 포함된 인덱스 ID들 간의 미리 결정된 인덱스 순서에 기초하여 정렬된 후에 결정된다.

[0088] 예를 들어, 인덱스 ID는 일련 번호, 예컨대, 1, 2, 3, 4, 5 등인 것으로 가정한다. 임의의 속성 저장 필드에 대해, 해당 속성 저장 필드에 인접해 있는 두 개의 속성 저장 필드들에 있어서, 하나의 속성 저장 필드의 일련 번호는 해당 속성 저장 필드의 일련 번호보다 크고, 나머지 다른 하나의 속성 저장 필드의 일련 번호는 해당 속성 저장 필드의 일련 번호보다 작다. 다른 모든 속성 저장 필드에도 동일하게 적용된다.

[0089] 본 단계에서는, 타겟 속성 대응관계에서의 인덱스 ID와, 저장된 속성 대응관계들에서의 인덱스 ID들에 기초하여

모든 저장된 속성 대응관계들에서 타겟 속성 대응관계가 정렬될 수 있고, 타겟 속성 저장 필드에서 타겟 속성 대응관계를 사이에 끼고 있는 두 개의 속성 대응관계들이 결정된다. 두 개의 속성 대응관계들에서, 하나의 속성 대응관계의 일련 번호는 타겟 속성 대응관계의 일련 번호보다 크고, 나머지 다른 하나의 속성 대응관계의 일련 번호는 타겟 속성 대응관계의 일련 번호보다 작다. 두 개의 속성 대응관계들 사이의 위치는 타겟 속성 대응관계가 저장될 위치로서 결정될 수 있다.

[0090] 단계 S302: 타겟 속성 대응관계를 상기 결정된 위치에 저장한다.

[0091] 사용자에게 정확한 서비스들을 제공할 필요가 있는 경우, 서버는 사용자의 하나 이상의 개인 속성의 속성값을 질의할 필요가 있다. 도 4를 참조하면, 질의는 다음과 같은 방법에 따라 수행될 수 있다. 본 방법은 다음 단계들을 포함한다:

[0092] 단계 S401: 사용자 ID와 개인 속성의 속성 ID를 포함하는 취득 요청이 획득된 경우, 사용자 ID에 대응하는 속성 저장 필드가 제1 대응관계들에서 존재하는지 여부를 결정하기 위해 사용자 ID들과 속성 저장 필드들 간의 제1 대응관계들을 검색한다.

[0093] 본 발명개시의 구현예에서, 사용자에게 정확한 서비스를 제공할 필요가 있는 경우, 사용자의 하나 이상의 개인 속성의 속성값을 획득할 필요가 있다. 먼저, 사용자의 사용자 ID 및 획득될 필요가 있는 개인 속성의 속성 ID에 기초하여 취득 요청이 생성된다. 그 후, 사용자 ID들과 속성 저장 필드들 간의 제1 대응관계들이 획득될 수 있고, 제1 대응관계들을 검색하여 제1 대응관계들에서 사용자 ID에 대응하는 속성 저장 필드가 존재하는지 여부를 결정한다.

[0094] 단계 S402: 사용자 ID에 대응하는 속성 저장 필드가 존재하는 경우, 속성 ID에 대응하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색한다.

[0095] 본 발명개시의 구현예에서, 속성 대응관계가 속성 ID와 속성값 간의 대응관계인 경우, 속성 ID를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 본 단계에서 속성 저장 필드를 검색할 수 있으며; 속성 ID를 포함하는 속성 대응관계가 존재하는 경우, 속성 ID에 대응하는 속성 대응관계가 속성 저장 필드에서 존재한다고 결정할 수 있다.

[0096] 구체적으로, 속성 저장 필드에서 제1 속성 대응관계가 획득되고, 제1 속성 대응관계에서의 속성 ID가 획득되며, 제1 속성 대응관계에서의 속성 ID가 속성 ID와 동일한지의 여부가 결정된다. 양자가 동일한 경우, 제1 속성 대응관계가 속성 ID에 대응하는 속성 대응관계로서 결정된다. 이와 달리, 양자가 상이한 경우, 제2 속성 대응관계가 획득되고, 제2 속성 대응관계에서의 속성 ID가 속성 ID와 동일한지의 여부가 결정된다. 양자가 동일한 경우, 제2 속성 대응관계가 속성 ID에 대응하는 속성 대응관계로서 결정된다. 이와 달리, 양자가 상이한 경우, 다음번aze 속성 대응관계가 획득되고, 속성 대응관계에서의 속성 ID가 속성 ID와 동일할 때 까지 전술한 절차가 실행되며, 획득된 속성 대응관계가 속성 ID에 대응하는 속성 대응관계로서 결정된다. 속성 저장 필드 내의 모든 속성 대응관계들에서의 속성 ID들이 속성 ID와 상이한 경우, 속성 ID에 대응하는 속성 대응관계가 속성 저장 필드에서 없다고 결정될 수 있다.

[0097] 속성 ID에 대응하는 속성 저장 필드가 없는 경우, 본 절차는 종료된다.

[0098] 일반적으로, 속성 저장 필드는 복수의 속성 대응관계들을 저장한다. 속성 ID에 대응하는 속성 대응관계가 속성 저장 필드에서 뒤쪽 위치에 저장되면, 대부분의 속성 대응관계들을 전술한 방법에 따라 거쳐갈 것인데, 이것은 오랜 시간이 걸리고 검색 효율을 낮춘다.

[0099] 본 발명개시의 다른 구현예에서, 속성 대응관계가 속성 ID에 대응하는 인덱스 ID와 속성값 간의 대응관계인 경우, 이 단계에서는 속성 ID들과 인덱스 ID들 간의 제2 대응관계들이 획득될 수 있다. 제2 대응관계들에서 속성 ID에 대응하는 인덱스 ID를 검색하고; 인덱스 ID를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하며; 인덱스 ID를 포함하는 속성 대응관계가 존재하는 경우, 속성 ID에 대응하는 속성 대응관계가 속성 저장 필드에서 존재한다고 결정될 수 있다.

[0100] 본 발명개시의 구현예에서 속성 저장 필드에서의 속성 대응관계들의 위치는 속성 대응관계들에 포함된 인덱스 ID들 간의 미리 결정된 인덱스 순서에 기초하여 정렬된 후에 결정된다. 따라서, 검색 효율을 향상시키기 위해, 인덱스 ID를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하는 경우, 먼저, 인덱스 ID가, 속성 저장 필드에서의 하나 이상의 미리 결정된 위치들에 있는 속성 대응관계들에서의 인덱스 ID들과 비교될 수 있고; 비교 결과에 기초하여, 속성 대응관계의 속성 저장 필드에서, 인덱스

스 ID를 포함하는 대략적인 위치가 결정되며; 그 후, 인덱스 ID를 포함하는 속성 대응관계가 존재하는지 여부를 결정하기 위해, 인덱스 ID가 상기 대략적인 위치에서의 하나 이상의 속성 대응관계에서의 인덱스 ID와 비교된다.

[0101] 단계 S403: 속성 ID에 대응하는 속성 대응관계가 존재하는 경우, 속성값을 속성 대응관계에서 획득한다.

[0102] 속성 ID에 대응하는 속성 대응관계가 존재하지 않는 경우, 이것은 사용자 ID에 대응하는 사용자가 개인 속성을 갖지 않는다는 것을 나타낸다. 또한, 사용자 ID에 대응하는 사용자가 개인 속성을 갖지 않는다는 것이 사용자에게 프롬프트될 수 있다.

[0103] 도 5는 구현예에 따른 데이터 처리 디바이스를 도시하는 블록도이다. 도 5를 참조하면, 본 디바이스는, 사용자 ID뿐만이 아니라, 상기 사용자 ID에 대응하는 적어도 하나의 개인 속성의 속성 ID와 속성값을 획득하도록 구성된 제1 획득 모듈(11); 개인 속성을 각각에 대해, 개인 속성의 속성 ID와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계를 생성하도록 구성된 생성 모듈(12); 사용자 ID에 대응하는 타겟 속성 저장 필드가 사용자 ID들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부를 결정하도록 구성된 결정 모듈(13); 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하는 경우, 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장하도록 구성된 제1 저장 모듈(14); 및 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하지 않는 경우, 사용자 ID에 대응하는 속성 저장 필드를 창설하도록 구성된 창설 모듈(15); 및 창설된 속성 저장 필드에 모든 타겟 속성 대응관계들을 저장하도록 구성된 제2 저장 모듈(16)을 포함한다.

[0104] 생성 모듈(12)은, 개인 속성의 속성 ID와 개인 속성의 속성값 간의 타겟 속성 대응관계를 구축하도록 구성된 제1 구축 유닛; 또는 저장된 속성 ID들과 시퀀스 인덱스들의 인덱스 ID들 간의 제2 대응관계들을 획득하도록 구성된 제1 획득 유닛; 제2 대응관계들에서 속성 ID에 대응하는 인덱스 ID를 검색하도록 구성된 검색 유닛; 및 인덱스 ID와 개인 속성의 속성값 간의 타겟 속성 대응관계를 구축하도록 구성된 제2 구축 유닛을 포함한다.

[0105] 제1 저장 모듈(14)은, 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은지 여부를 결정하도록 구성된 결정 유닛; 및 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 작거나 또는 같은 경우, 모든 타겟 속성 대응관계들을 타겟 속성 저장 필드에 저장하도록 구성된 제1 저장 유닛; 모든 타겟 속성 대응관계들에 의해 점유된 저장 공간이 타겟 속성 저장 필드의 유휴 저장 공간보다 큰 경우, 사용자 ID에 대응하는 적어도 하나의 새로운 속성 저장 필드를 창설하도록 구성된 창설 유닛; 및 모든 타겟 속성 대응관계들을, 새로운 속성 저장 필드에 또는 타겟 속성 저장 필드에 그리고 새로운 속성 저장 필드에 저장하도록 구성된 제2 저장 유닛을 포함한다.

[0106] 제1 저장 모듈(14)은, 타겟 속성 대응관계에서의 인덱스 ID 및 타겟 속성 저장 필드에 저장된 모든 속성 대응관계들에서의 인덱스 ID들에 기초하여 타겟 속성 대응관계가 타겟 속성 저장 필드에 저장될 위치를 결정하도록 구성된 제1 결정 유닛; 및 타겟 속성 대응관계를 상기 결정된 위치에 저장하도록 구성된 제3 저장 유닛을 포함한다.

[0107] 본 발명개시의 도 5에서 도시된 구현예에서, 사용자 ID뿐만이 아니라, 상기 사용자 ID에 대응하는 적어도 하나의 개인 속성의 속성 ID와 속성값이 획득된다. 개인 속성을 각각에 대해, 개인 속성의 속성 ID와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계가 생성된다. 사용자 ID에 대응하는 타겟 속성 저장 필드가 사용자 ID들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부가 결정되며; 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하는 경우, 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들이 저장된다. 대안적으로, 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하지 않는 경우, 사용자 ID에 대응하는 속성 저장 필드가 창설되고, 창설된 속성 저장 필드에 모든 타겟 속성 대응관계들이 저장된다.

[0108] 종래의 기술에서의 저장 구조는 본 발명개시의 구현예에서 폐기된다. 첫번째로, 비어 있는 필드가 없으며, 이에 따라 비어 있는 필드들을 위해 저장 공간을 낭비하는 문제점을 회피하여, 저장 공간을 절약하는 목적을 달성한다. 또한, 종래의 기술에서는, 각 필드에 하나의 속성값만이 저장된다. 그러나, 본 발명개시의 구현예들에서는, 하나의 속성 저장 필드에 복수의 속성 대응관계들이 저장되고, 각 속성 대응관계는 속성값을 포함한다. 따라서, 본 발명개시의 구현예들에서는, 종래의 기술과 비교하여, 하나의 속성 저장 필드에 복수의 속성값들이 저장되고, 속성 저장 필드에서의 저장 공간이 최대한 활용되어서, 저장 공간을 절약한다.

[0109] 또한, 서비스 사업의 급속한 발전에 따라, 사용자의 개인화된 요구는 무궁무진한 다양성으로 표출된다. 따라서,

종종 새로운 개인 속성을 빨굴해낼 필요가 있다. 이 경우, 종래의 기술에서는 [표 1]에서 첫번째 행의 맨 오른쪽에 필드가 추가될 필요가 있으며, 새로운 개인 속성의 속성 명칭이 새롭게 추가된 필드에 추가된다. 그 후, [표 1]에서의 각 사용자 ID의 행의 맨 오른쪽에 새로운 필드가 추가될 필요가 있어서, 사용자가 새로운 개인 속성을 갖는 경우, 사용자 ID가 위치해 있는 행의 맨 오른쪽에 새롭게 추가된 필드가 개인 속성의 속성값으로 채워지도록 한다.

[0110] 그러나, [표 1]에서의 사용자 ID에 대응하는 사용자들 모두가 새로운 개인 속성을 갖는 것은 아니고, 이에 따라 모든 사용자 ID들이 위치해 있는 행의 맨 오른쪽에 새롭게 추가된 필드들이 속성값들로 채워지지는 않을 것인데, 이로 인해, 일부 비어 있는 필드들을 야기시키고, 저장 공간을 다소 낭비시킨다.

[0111] 또한, [표 1]에 새로운 필드들을 추가하는 과정에서는, [표 1]을 잡그는 것이 필요하며, 잡긴 상태의 [표 1]은 사용할 수 없다. [표 1]에 매우 많은 수의 사용자 ID들이 저장되어 있는 경우, 즉, [표 1]에 많은 수의 행들이 있는 경우, 모든 사용자 ID들이 위치해 있는 행의 맨 오른쪽에 새로운 필드들을 추가하는 것은 많은 시간이 소요될 것이다. 결과적으로, [표 1]을 너무 오랜 시간동안 사용할 수 없게 되므로, 사용자에게 장기간에 걸쳐 정확한 서비스를 제공하는 것은 불가능해진다. 그러나, 본 발명개시의 구현예에서는, 새로운 개인 속성이 개발되더라도, 개발된 새로운 개인 속성을 위한 필드를 할당할 필요가 없고, 저장된 개인 속성을 잡글 필요가 없어서, 사용자에게 지속적으로 정확한 서비스를 제공할 수 있다. 또한, 개발된 새로운 개인 속성들을 위한 필드들을 할당할 필요가 없고, 이에 따라 비어 있는 필드가 없으므로, 저장 공간을 낭비하는 문제점을 완화시킬 수 있다.

[0112] 도 6은 구현예에 따른 데이터 처리 디바이스를 도시하는 블록도이다. 도 6를 참조하면, 본 디바이스는, 사용자 ID와 개인 속성의 속성 ID를 포함하는 취득 요청이 획득된 경우, 사용자 ID에 대응하는 속성 저장 필드가 제1 대응관계들에서 존재하는지 여부를 결정하기 위해 사용자 ID들과 속성 저장 필드들 간의 제1 대응관계를 검색하도록 구성된 제1 검색 모듈(21); 사용자 ID에 대응하는 속성 저장 필드가 존재하는 경우, 속성 ID에 대응하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하도록 구성된 제2 검색 모듈(22); 및 속성 ID에 대응하는 속성 대응관계가 존재하는 경우, 속성값을 속성 대응관계에서 획득하도록 구성된 제2 획득 모듈(23)을 포함한다.

[0113] 제2 검색 모듈(22)은, 속성 ID를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하도록 구성된 제1 검색 유닛; 및 속성 ID를 포함하는 속성 대응관계가 존재하는 경우, 속성 ID에 대응하는 속성 대응관계가 속성 저장 필드에서 존재한다고 결정하도록 구성된 제2 결정 유닛을 포함한다.

[0114] 제2 검색 모듈(22)은, 속성 ID들과 인덱스 ID들 간의 제2 대응관계들을 획득하도록 구성된 제2 획득 유닛; 제2 대응관계들에서 속성 ID에 대응하는 인덱스 ID를 검색하도록 구성된 제2 검색 유닛; 속성 ID를 포함하는 속성 대응관계가 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 속성 저장 필드를 검색하도록 구성된 제3 검색 유닛; 및 인덱스 ID를 포함하는 속성 대응관계가 존재하는 경우, 속성 ID에 대응하는 속성 대응관계가 속성 저장 필드에서 존재한다고 결정하도록 구성된 제3 결정 유닛을 포함한다.

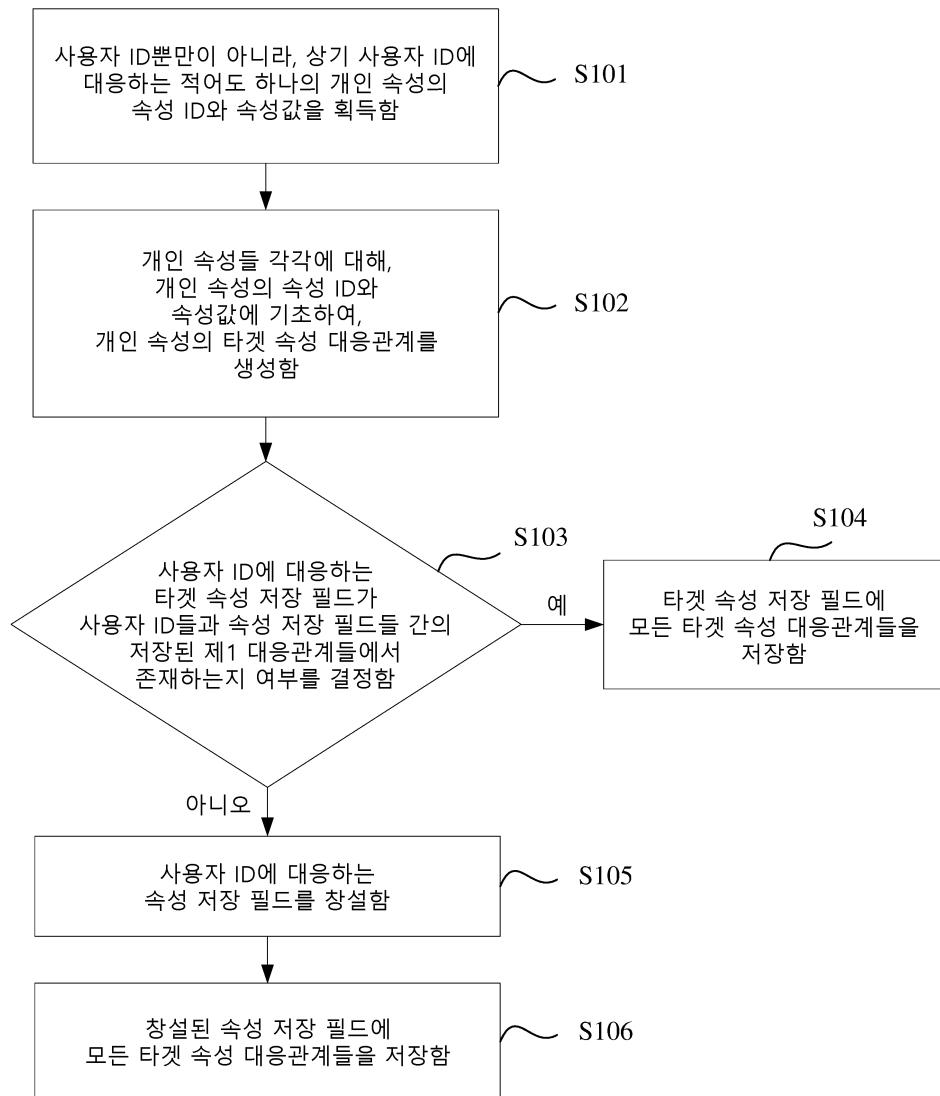
[0115] 본 발명개시의 도 6에서 도시된 구현예에서, 사용자 ID뿐만이 아니라, 상기 사용자 ID에 대응하는 적어도 하나의 개인 속성의 속성 ID와 속성값이 획득된다. 개인 속성을 각각에 대해, 개인 속성의 속성 ID와 속성값에 기초하여 개인 속성의 타겟 속성 대응관계가 생성된다. 사용자 ID에 대응하는 타겟 속성 저장 필드가 사용자 ID들과 속성 저장 필드들 간의 저장된 제1 대응관계들에서 존재하는지 여부가 결정되며; 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하는 경우, 타겟 속성 저장 필드에 모든 타겟 속성 대응관계들이 저장된다. 대안적으로, 사용자 ID에 대응하는 타겟 속성 저장 필드가 제1 대응관계들에서 존재하지 않는 경우, 사용자 ID에 대응하는 속성 저장 필드가 창설되고, 창설된 속성 저장 필드에 모든 타겟 속성 대응관계들이 저장된다.

[0116] 종래의 기술에서의 저장 구조는 본 발명개시의 도 1에서 도시된 구현예에서 폐기된다. 첫번째로, 비어 있는 필드가 없으며, 이에 따라 비어 있는 필드들을 위해 저장 공간을 낭비하는 문제점을 회피하여, 저장 공간을 절약하는 목적을 달성한다. 또한, 종래의 기술에서는, 각 필드에 하나의 속성값만이 저장된다. 그러나, 본 발명개시의 도 6에서 도시된 구현예들에서는, 하나의 속성 저장 필드에 복수의 속성 대응관계들이 저장되고, 각 속성 대응관계는 속성값을 포함한다. 따라서, 본 발명개시의 도 6에서 도시된 구현예들에서는, 종래의 기술과 비교하여, 하나의 속성 저장 필드에 복수의 속성값들이 저장되고, 속성 저장 필드에서의 저장 공간이 최대한 활용되어서, 저장 공간을 절약한다.

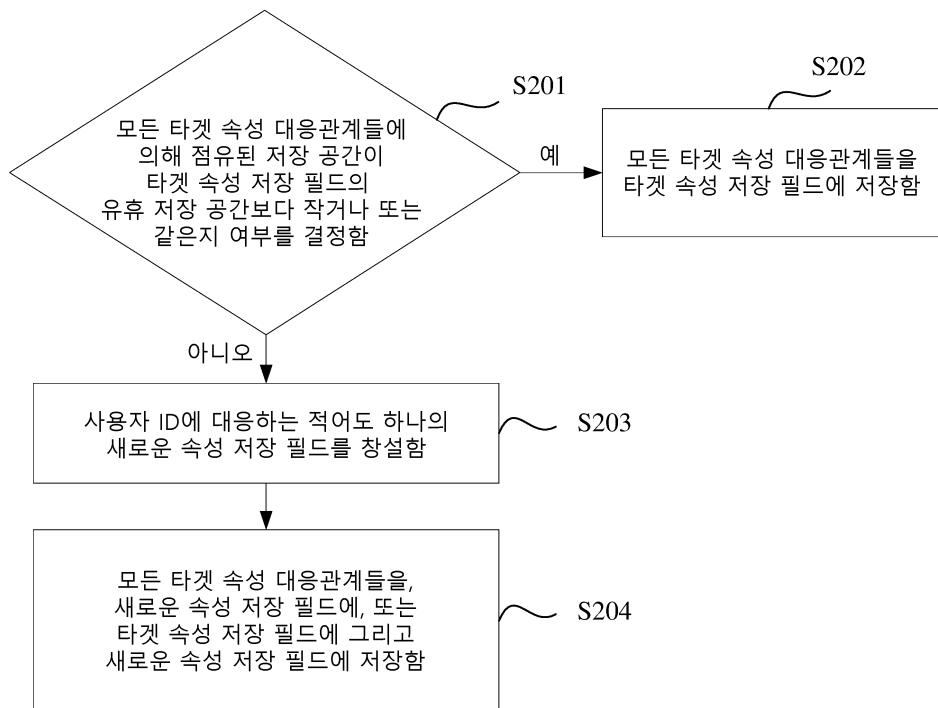
- [0117] 또한, 서비스 사업의 급속한 발전에 따라, 사용자의 개인화된 요구는 무궁무진한 다양성으로 표출된다. 따라서, 종종 새로운 개인 속성을 빨గ해낼 필요가 있다. 이 경우, 종래의 기술에서는 [표 1]에서 첫번째 행의 맨 오른쪽에 필드가 추가될 필요가 있으며, 새로운 개인 속성의 속성 명칭이 새롭게 추가된 필드에 추가된다. 그 후, [표 1]에서의 각 사용자 ID의 행의 맨 오른쪽에 새로운 필드가 추가될 필요가 있어서, 사용자가 새로운 개인 속성을 갖는 경우, 사용자 ID가 위치해 있는 행의 맨 오른쪽에 새롭게 추가된 필드가 개인 속성값으로 채워지도록 한다.
- [0118] 그러나, 이 경우, [표 1]에서의 사용자 ID에 대응하는 사용자들 모두가 새로운 개인 속성을 갖는 것은 아니고, 이에 따라 모든 사용자 ID들이 위치해 있는 행의 맨 오른쪽에 새롭게 추가된 필드들이 속성값들로 채워지지는 않을 것인데, 이로 인해, 일부 비어 있는 필드들을 야기시키고, 저장 공간을 다소 낭비시킨다.
- [0119] 또한, [표 1]에 새로운 필드들을 추가하는 과정에서는, [표 1]을 잠그는 것이 필요하며, 잠긴 상태의 [표 1]은 사용할 수 없다. [표 1]에 매우 많은 수의 사용자 ID들이 저장되어 있는 경우, 즉, [표 1]에 많은 수의 행들이 있는 경우, 모든 사용자 ID들이 위치해 있는 행의 맨 오른쪽에 새로운 필드들을 추가하는 것은 많은 시간이 소요될 것이다. 결과적으로, [표 1]을 너무 오랜 시간동안 사용할 수 없게 되므로, 사용자에게 장기간에 걸쳐 정확한 서비스를 제공하는 것은 불가능해진다. 그러나, 본 발명개시의 도 6에서 도시된 구현예에서는, 새로운 개인 속성이 개발되더라도, 개발된 새로운 개인 속성을 위한 필드를 할당할 필요가 없고, 저장된 개인 속성을 잡글 필요가 없어서, 사용자에게 지속적으로 정확한 서비스를 제공할 수 있다. 또한, 개발된 새로운 개인 속성들을 위한 필드들을 할당할 필요가 없고, 이에 따라 비어 있는 필드가 없으므로, 저장 공간을 낭비하는 문제점을 완화시킬 수 있다.
- [0120] 상기 구현예들에서의 디바이스와 관련하여, 모듈들에 의한 동작들을 수행하기 위한 특정 방법을 본 방법과 관련된 구현예들에서 상세히 설명하였으며, 여기서는 간략화를 위해 상세한 설명을 생략한다.
- [0121] 당업자는 명세서를 고려하고 본 발명개시를 실시한 후에 본 발명개시의 다른 구현예를 쉽게 파악할 수 있다. 본 출원은 본 발명개시의 임의의 변형, 사용, 또는 개조를 커버하도록 의도된 것이며, 이러한 변형, 사용, 또는 개조는 본 발명개시의 일반적인 원리를 따르며, 본 발명개시의 기술분야에 개시되지 않은 통상의 지식 또는 종래 기술을 포함한다. 명세서 및 구현예에는 단지 예시로서 간주되며, 본 발명개시의 진정한 범위와 사상은 첨부된 청구범위에 의해 표시된다.
- [0122] 본 발명개시는 위에서 설명되고 첨부 도면에서 도시된 정확한 구조물들로 한정되지 않고, 본 발명개시의 범위를 벗어나지 않고서 다양한 수정과 변경이 행해질 수 있다는 것을 이해해야 한다. 본 발명개시의 범위는 첨부된 청구범위에 의해서만 제한된다.

도면

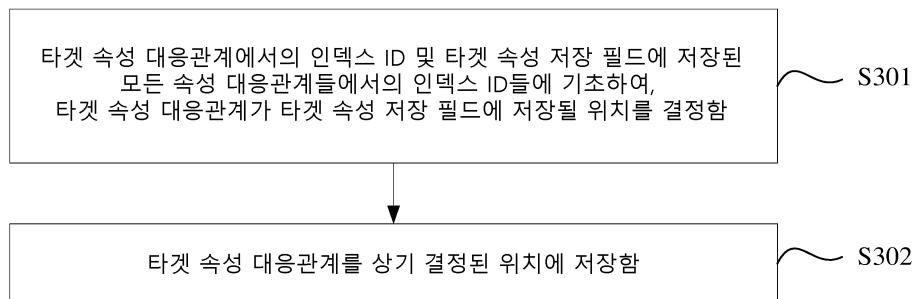
도면1



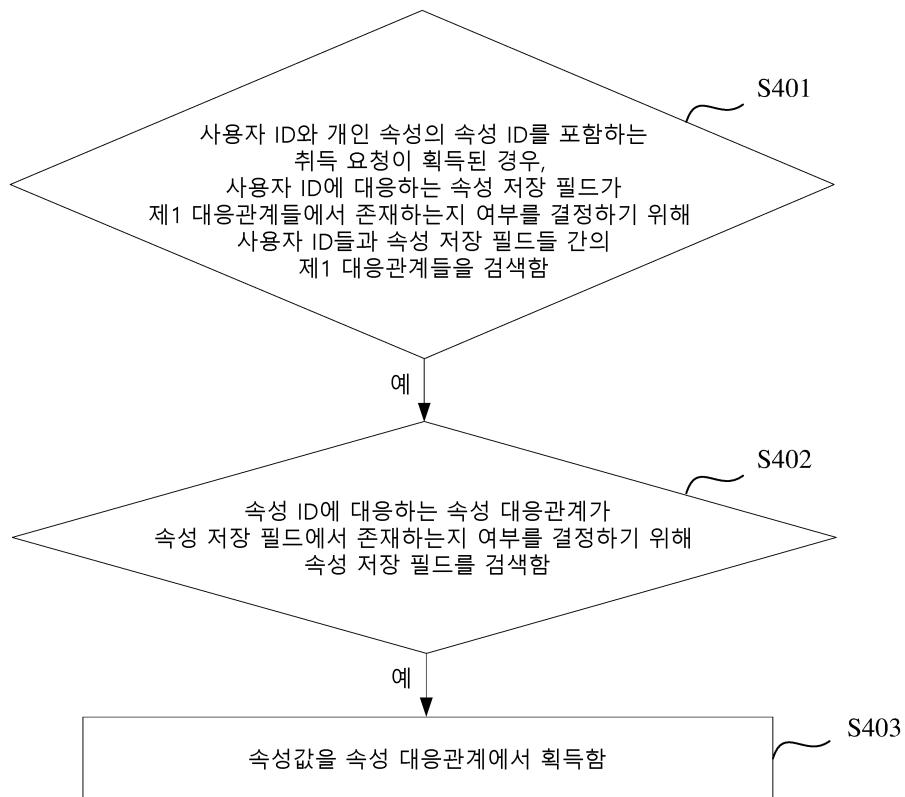
도면2



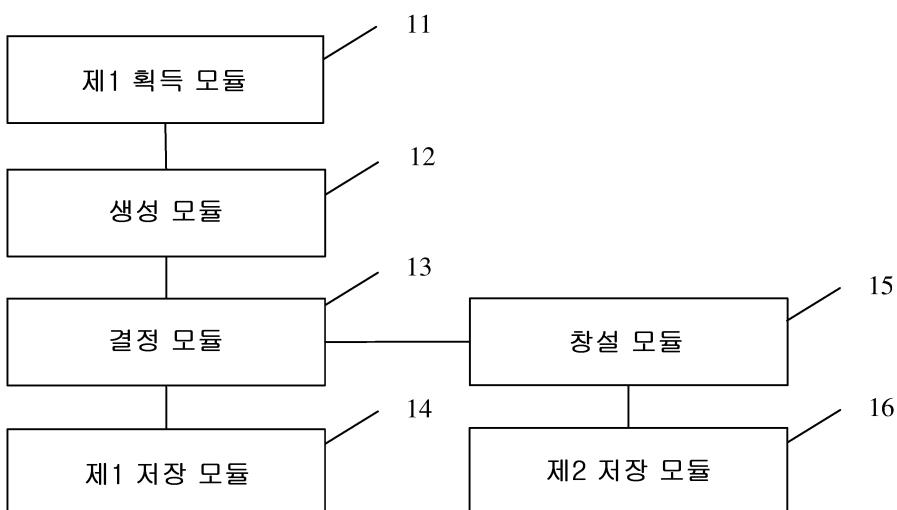
도면3



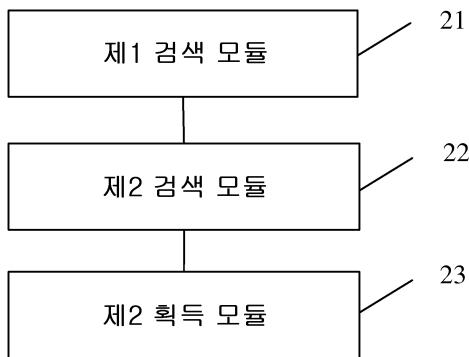
도면4



도면5



도면6



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 6

【변경전】

제4항에 있어서,

상기 방법은,

인덱스 식별자를 포함하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 상기 타겟 속성 저장 필드를 검색하는 단계

를 더 포함하고,

상기 타겟 속성 저장 필드를 검색하는 단계는,

상기 인덱스 식별자를, 상기 타겟 속성 저장 필드에서의 하나 이상의 미리 결정된 위치에 있는 속성 대응관계들에서의 인덱스 식별자들과 비교하는 단계;

상기 비교 결과에 기초하여, 상기 타겟 속성 저장 필드에서의 상기 인덱스 식별자를 포함하는 속성 대응관계의 대략적인 위치를 결정하는 단계; 및

상기 인덱스 식별자를 포함하는 속성 대응관계가 존재하는지 여부를 결정하기 위해, 상기 인덱스 식별자를, 상기 상기 대략적인 위치에 있는 하나 이상의 속성 대응관계에서의 인덱스 식별자들과 비교하는 단계

를 포함한 것인 데이터 처리 방법.

【변경후】

제4항에 있어서,

상기 방법은,

인덱스 식별자를 포함하는 속성 대응관계가 상기 타겟 속성 저장 필드에서 존재하는지 여부를 결정하기 위해 상기 타겟 속성 저장 필드를 검색하는 단계

를 더 포함하고,

상기 타겟 속성 저장 필드를 검색하는 단계는,

상기 인덱스 식별자를, 상기 타겟 속성 저장 필드에서의 하나 이상의 미리 결정된 위치에 있는 속성 대응관계들에서의 인덱스 식별자들과 비교하는 단계;

비교 결과에 기초하여, 상기 타겟 속성 저장 필드에서의 상기 인덱스 식별자를 포함하는 속성 대응관계의 대략적인 위치를 결정하는 단계; 및

상기 인덱스 식별자를 포함하는 속성 대응관계가 존재하는지 여부를 결정하기 위해, 상기 인덱스 식별자를, 상기 대략적인 위치에 있는 하나 이상의 속성 대응관계에서의 인덱스 식별자들과 비교하는 단계

를 포함한 것인 데이터 처리 방법.