(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0080793
(43) 공개일자 2019년07월08일

(51) 국제특허분류(Int. Cl.)
G06F 11/14 (2006.01)
(52) CPC특허분류
G06F 11/1446 (2013.01)
(21) 출원번호 10-2018-0170741
(22) 출원일자 2018년12월27일
심사청구일자 2018년12월27일
(30) 우선권주장
1020170182811 2017년12월28일 대한민국(KR)

(71) 출원인
부경대학교 산학협력단
부산광역시 남구 신선로 365 (용당동, 부경대학교)
(72) 발명자
송하주
부산광역시 수영구 남천동로 41, 107동 2101호(남천동, 남천동코오롱하늘채골든비치아파트)
권오흠
부산광역시 남구 수영로 345, 111동 1801호(대연동, 대연힐스테이트푸르지오)
(뒷면에 계속)
(74) 대리인
특허법인명인

전체 청구항 수 : 총 5 항

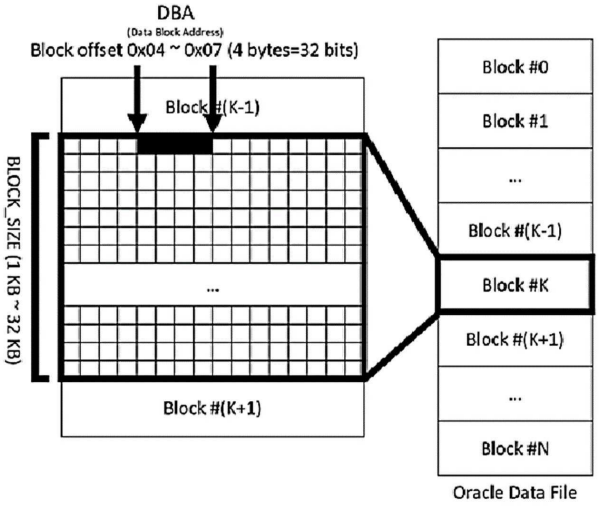
(54) 발명의 명칭 **디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법**

(57) 요 약

디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법이 개시된다.

본 발명의 실시예에 따른 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법은 입력부(10)를 통해, 데이터베이스의 데이터 파일을 입력받는 단계; 제어부(20)에 의해, 상기 데이터 파일의 데이터베이스 블록 어드레스(DBA: database block address)로부터 블록의 크기를 예측하는 단계; 및 상기 제어부(20)에 의해, 데이터베이스 블록 어드레스(DBA: database block address)로부터 파일 식별번호를 예측하는 단계;를 포함할 수 있다.

대 표 도 - 도2



(72) 발명자

김호준

부산광역시 부산진구 백양대로60번길 16 (당감동)

김종민

부산광역시 해운대구 재반로242번길 41, 5동 204호

명 세 서

청구범위

청구항 1

입력부(10)를 통해, 데이터베이스의 데이터 파일을 입력받는 단계;

제어부(20)에 의해, 상기 데이터 파일의 데이터베이스 블록 어드레스(DBA: database block address)로부터 블록의 크기를 예측하는 단계; 및

상기 제어부(20)에 의해, 데이터베이스 블록 어드레스(DBA: database block address)로부터 파일 식별번호를 예측하는 단계;

를 포함하는 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법.

청구항 2

제1항에 있어서,

상기 블록의 크기를 예측하는 단계는

상기 데이터 파일의 맨 앞에서 설정된 오프셋 지점으로부터 설정된 크기의 제1 데이터를 획득하는 단계;

상기 제1 데이터의 맨 뒤에서 설정된 크기의 제2 데이터를 획득하는 단계; 및

상기 데이터베이스에서 구성 가능한 최대 블록 크기를 상기 제2 데이터로 나누어 상기 블록의 크기를 계산하는 단계;

를 포함하는 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법.

청구항 3

제2항에 있어서,

상기 파일 식별 번호를 예측하는 단계는

상기 데이터 파일의 두 번째 블록의 DBA 오프셋 지점으로부터 설정된 크기의 제3 데이터를 획득하는 단계; 및

상기 제3 데이터의 맨 앞에서 설정된 크기의 제4 데이터를 상기 파일 식별 번호로 예측하는 단계;

를 포함하는 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법.

청구항 4

제1항에 있어서,

상기 데이터 파일은

상기 복수의 블록이 연속적으로 구성되고, 상기 데이터 파일의 크기는 단위 블록의 크기보다 작으며, 상기 블록의 크기는 2의 거듭제곱수로 이루어지고, 상기 데이터 파일의 첫 번째 블록은 메타데이터이며, 상기 블록의 시작점으로부터 설정된 위치에 상기 데이터베이스 블록 어드레스가 위치하며, 상기 블록의 번호는 0부터 순차적으로 1씩 증가하는 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법.

청구항 5

제1항에 있어서,

상기 데이터베이스 블록 어드레스는

상기 데이터 파일의 첫 번째 블록을 제외한 모든 블록이 동일한 오프셋 위치에 갖는 블록 위치의 식별자인 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법에 관한 것이다. 더욱 상세하게는, 데이터 파일만을 이용하여 데이터 파일의 구조 정보를 예측할 수 있는 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법에 관한 것이다.

배 경 기 술

[0002] 일반적으로 조직은 많은 양의 데이터를 통합적으로 관리하기 위해 데이터베이스를 이용하므로, 데이터베이스에는 업무데이터, 개인정보와 같은 중요한 정보가 저장되어 있는 경우가 많다. 이는 의미 있는 정보가 남아있을 가능성이 크다는 것을 뜻하므로 디지털 포렌식 관점에서 데이터베이스가 중요한 조사대상이다.

[0003] 데이터베이스에서 정상적인 레코드를 추출하는 것도 중요하지만, 삭제된 레코드를 복구하는 것도 중요하다. 삭제된 레코드 복구를 통해 사용자의 행위를 재구성할 수 있고, 용의자의 고의적 증거 인멸 행위에 대응할 수 있으므로 수사에 큰 도움을 줄 수 있다.

[0004] 테이블 스페이스는 데이터베이스에서 실제 데이터를 물리적으로 저장하는 공간을 말한다. 이 테이블 스페이스를 분석하면 데이터베이스의 테이블 데이터와 테이블 스키마 정보를 얻을 수 있다. IDC에 따르면 오라클 데이터베이스는 전 세계적으로 40.7%의 점유율을 기록하고 있으며 국내 데이터베이스 시장에서 오라클 데이터베이스는 60% 점유율을 기록하고 있다.

[0005] 또한, 현재까지의 데이터베이스 포렌식에 대한 선행 연구는 트랜잭션 로그를 토대로 레코드를 복구하는 것에 집중되어 있다. P. Wright는 오라클 데이터베이스에서 리두로그(redolog)와 로그마이너(LogMiner)를 통해 데이터베이스의 변경사항을 추적하는 방안과 이전 시점의 데이터를 복원하는 방법에 대해서 설명하였다. 또한, Heloise Pieterse는 데이터베이스에서 암호화, 분할 저장 등을 이용하여 데이터 값이나 레코드, 테이블을 은닉하는 기법에 대해서 정리하였으며, 트랜잭션 로그의 중요성에 대해서 강조하였다. Oluwasola Mary는 데이터베이스에서 쿼리를 대수학으로 표현할 수 있음을 보였다. 그리고 쿼리를 역으로도 나타낼 수 있으며, 이 수식을 저장하는 Relational Algebra Log를 제안하였다. 이 로그를 통해 이전 시점의 데이터들을 복원하는 기법에 대해서 설명하였다. 일반적으로 오라클 데이터베이스의 삭제된 레코드를 복원할 때는 DBF파일과 트랜잭션 로그가 쌍으로 존재해야 한다. 그리고 운용중인 오라클 데이터베이스에 로그마이너를 이용해 삭제된 레코드를 복원한다. 하지만, 이 방법은 트랜잭션 로그가 기록된 시점만 복구가 가능하며, 기록되지 않은 시점은 복구하지 못한다. 또한, 오라클 데이터베이스에서 트랜잭션 로그를 남기는 방법 중 노아카이브 모드(No Archive Mode)의 경우 미리 지정한 용량을 초과하게 되면 기존의 트랜잭션 로그에 덮어쓰기 때문에 오라클 데이터베이스 운용 환경이나 회사 정책에 따라 복구 여부가 결정될 수도 있다.

[0006] 그러나 종래 기술에 의하면 데이터베이스에 데이터를 복원할 때, 트랜잭션 로그를 통해서만 복구가 가능하다는 문제가 존재하였다. 트랜잭션 로그에 의하지 않고 데이터를 복원할 수 있는 연구가 요구되고 있다.

[0007] 앞에서 언급한 바와 같이, 오라클 데이터베이스는 상용 DBMS 중 세계에서 가장 널리 사용되는 제품이며, 국내에서도 많은 기관에서 중요한 데이터를 운용 및 관리하는데 사용되고 있다. 만약, 오라클 데이터베이스에 문제가 발생하여 데이터에 접근할 수 없게 되면 해당 기관은 정상적인 업무를 수행하는데 어려움이 발생한다.

[0008] 일반적으로 오라클 데이터베이스에 문제가 발생할 경우 고장회복(recovery) 기능을 사용하여 거의 자동화된 복구가 가능하지만, 사용자의 부주의 혹은 시스템 오류로 인해 일부 시스템 파일에 오류가 발생하면 불가능한 경우가 발생한다.

[0009] 예를 들어, 오라클 데이터베이스의 구조 정보(데이터 파일의 블록 크기, 식별 번호, 및 경로 등)를 포함하는 제어 파일이 손상되는 경우, 데이터 파일의 블록 크기와 식별 번호 또한 손상된다. 이러한 경우, 데이터 베이스의 운용이나 기동은 물론이고 데이터를 복구 및 추출하는 것도 불가능한 문제가 발생한다.

[0010] 따라서, 이러한 제어 파일이 손상 또는 손실되는 경우, 데이터베이스의 운동이나 기동은 물론이고 데이터 파일을 복구 또는 추출하는 것도 불가능하게 된다.

[0011] 이러한 문제를 해결하기 위해, 제어 파일이 손상 또는 손실되었을 때, 데이터 파일만을 이용하여 각 파일의 구조 정보를 추측하는 방법에 대한 연구가 요구되고 있다.

[0012] 이 배경기술 부분에 기재된 사항은 발명의 배경에 대한 이해를 증진하기 위하여 작성된 것으로서, 이 기술이 속하는 분야에서 통상의 지식을 가진 자에게 이미 알려진 종래기술이 아닌 사항을 포함할 수 있다.

발명의 내용

해결하려는 과제

[0013] 본 발명은 상기한 바와 같은 문제점을 해결하기 위한 것으로, 데이터베이스의 제어 파일이 손상되는 경우, 데이터 파일만을 이용하여 데이터 파일의 구조 정보를 예측할 수 있는 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법을 제공하는 것을 목적으로 한다.

과제의 해결 수단

[0014] 상기한 바와 같은 목적을 달성하기 위한 본 발명의 실시예에 따른 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법은 입력부(10)를 통해, 데이터베이스의 데이터 파일을 입력받는 단계; 제어부(20)에 의해, 상기 데이터 파일의 데이터베이스 블록 어드레스(DBA: database block address)로부터 블록의 크기를 예측하는 단계; 및 상기 제어부(20)에 의해, 데이터베이스 블록 어드레스(DBA: database block address)로부터 파일 식별번호를 예측하는 단계;를 포함할 수 있다.

[0015] 상기 블록의 크기를 예측하는 단계는 상기 데이터 파일의 맨 앞에서 설정된 오프셋 지점으로부터 설정된 크기의 제1 데이터를 획득하는 단계; 상기 제1 데이터의 맨 뒤에서 설정된 크기의 제2 데이터를 획득하는 단계; 및 상기 데이터베이스에서 구성 가능한 최대 블록 크기를 상기 제2 데이터로 나누어 상기 블록의 크기를 계산하는 단계;를 포함할 수 있다.

[0016] 상기 파일 식별 번호를 예측하는 단계는 상기 데이터 파일의 두 번째 블록의 DBA 오프셋 지점으로부터 설정된 크기의 제3 데이터를 획득하는 단계; 및 상기 제3 데이터의 맨 앞에서 설정된 크기의 제4 데이터를 상기 파일 식별 번호로 예측하는 단계;를 포함할 수 있다.

[0017] 상기 데이터 파일은 상기 복수의 블록이 연속적으로 구성되고, 상기 데이터 파일의 크기는 단위 블록의 크기보다 작으며, 상기 블록의 크기는 2의 거듭제곱수로 이루어지고, 상기 데이터 파일의 첫 번째 블록은 메타데이터이며, 상기 블록의 시작점으로부터 설정된 위치에 상기 데이터베이스 블록 어드레스가 위치하며, 상기 블록의 번호는 0부터 순차적으로 1씩 증가할 수 있다.

[0018] 상기 데이터베이스 블록 어드레스는 상기 데이터 파일의 첫 번째 블록을 제외한 모든 블록이 동일한 오프셋 위치에 갖는 블록 위치의 식별자일 수 있다.

발명의 효과

[0019] 상기한 바와 같은 본 발명의 실시예에 의한 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법에 의하면, 데이터 파일만을 이용하여 데이터 파일의 구조 정보를 예측할 수 있는 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법을 예측할 수 있다.

도면의 간단한 설명

[0020] 이 도면들은 본 발명의 예시적인 실시예를 설명하는데 참조하기 위함이므로, 본 발명의 기술적 사상을 첨부한 도면에 한정해서 해석하여서는 아니된다.

도 1은 본 발명의 실시예에 따른 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 장치의 블록도이다.

도 2는 본 발명의 실시예에 따른 데이터베이스를 구성하는 데이터 파일의 구조를 설명하기 위한 도면이다.

도 3은 본 발명의 실시예에 따른 데이터베이스 블록 어드레스의 구조를 설명하기 위한 도면이다.

도 4는 본 발명의 실시예에 따른 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법을 설명하기 위한 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0021] 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자

가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다.

[0022] 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 동일 또는 유사한 구성요소에 대해서는 동일한 참조 부호를 붙이도록 한다.

[0023] 추가적으로, 몇몇 방법들은 적어도 하나의 제어기에 의하여 실행될 수 있다.

[0024] 제어기라는 용어는 메모리와, 알고리즘 구조로 해석되는 하나 이상의 단계들을 실행하도록 된 프로세서를 포함하는 하드웨어 장치를 언급한다.

[0025] 상기 메모리는 알고리즘 단계들을 저장하도록 되어 있고, 프로세서는 아래에서 기재하는 하나 이상의 프로세스들을 수행하기 위하여 상기 알고리즘 단계들을 특별히 실행하도록 되어 있다.

[0026] 더 나아가, 본 발명의 제어 로직은 프로세서, 제어기 또는 이와 유사한 것에 의하여 실행되는 실행 가능한 프로그램 명령들을 포함하는 컴퓨터가 읽을 수 있는 수단 상의 일시적이지 않고 컴퓨터가 읽을 수 있는 매체로 구현될 수 있다.

[0027] 컴퓨터가 읽을 수 있는 수단의 예들은, 이에 한정되지는 않지만, ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 플래쉬 드라이브, 스마트 카드 및 광학 데이터 저장 장치를 포함한다.

[0029] 먼저, 본 발명의 실시 예에 따른 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 장치에 관하여 첨부된 도면을 참조하여 상세하게 설명한다.

[0030] 도 1은 본 발명의 실시예에 따른 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 장치의 블록도이다.

[0031] 도 1에 도시된 바와 같이, 본 발명의 실시 예에 따른 파일 복구 장치는 데이터 파일을 입력받는 입력부(10), 및 입력부(10)를 통해 입력된 데이터 파일로부터 데이터 파일의 구조 정보를 예측하는 제어부(20)를 포함할 수 있다.오라클 데이터베이스는 제어 파일과 데이터 파일을 포함하여 구성된다.

[0032] 제어 파일은 작은 크기의 바이너리 파일(binary file)이며, 오라클 데이터베이스의 구조 정보를 포함한다. 예를 들어, 데이터 파일 및 로그 파일의 블록 크기, 식별 번호, 및 경로 등 데이터 파일을 열기 위한 필요한 구조 정보를 포함하며, 오라클 데이터 베이스가 기동하기 위해 가장 먼저 참조하는 파일이다.

[0033] 데이터 파일은 복수의 블록으로 구성되며, 오라클 데이터베이스가 데이터 파일을 생성할 때 한 블록의 크기는 최소 1024(1KB)부터 최대 262144(256KB)의 크기로 설정될 수 있다. 블록의 크기가 작을수록 파일 내에 빈 곳이 줄어들어 파일의 크기가 작아지며, 블록의 크기가 클수록 데이터베이스의 운용 속도가 빨라진다. 데이터 파일의 블록 크기는 제어 파일에 저장되는데, 만약 제어 파일이 손상되거나 손실되면 데이터 파일의 블록의 시작과 끝을 알 수 없기 때문에 의미 있는 데이터를 추출할 수 없다.

[0034] 오라클 데이터베이스의 데이터 파일의 블록은 다음과 같은 특징을 갖는다.

[0035] 도 2는 본 발명의 실시예에 따른 데이터베이스를 구성하는 데이터 파일의 구조를 설명하기 위한 도면이다.

[0036] 도 2에 도시된 바와 같이, 데이터 파일은 블록의 연속으로 이루어진다. 그리고 데이터 파일의 크기는 단위 블록의 크기보다 크다. 또한, 가능한 블록의 크기는 2의 거듭제곱수(1KB~256KB: 1024~262144)이다. 그리고 데이터 파일의 첫 번째 블록은 오라클 데이터베이스의 데이터가 아닌 데이터 파일의 메타데이터이다. 또한, 데이터베이스 블록 어드레스(DBA: database block address)는 어느 하나의 블록의 시작점에서 0x4 지점(DBA 오프셋)에 위치한다. 마지막으로, 블록 번호는 데이터 파일 내의 첫 번째 블록을 0으로 하여, 순서대로 1씩 증가한다.

[0037] 데이터 파일은 동일한 크기를 갖는 복수의 블록으로 구성되는데, 블록은 데이터 파일의 저장 공간을 관리하는 최소 단위로써, 데이터 파일을 생성하는 시점에 결정된다.

[0038] 위에서, 데이터베이스 블록 어드레스(DBA: database block address)는 데이터 파일의 첫 번째 블록을 제외한 모든 블록이 동일한 오프셋 위치(예를 들어, 0x4)에 가지는 블록 위치의 식별자를 의미한다.

[0039] 도 3은 본 발명의 실시예에 따른 데이터베이스 블록 어드레스의 구조를 설명하기 위한 도면이다.

[0040] 도 3에 도시된 바와 같이, 데이터베이스 블록 어드레스는 4바이트(32비트)로 이루어지며, 상위 10비트는 파일 번호를 의미하고, 하위 22비트는 블록 번호를 의미한다.

[0041] 이와 같이, 본 발명의 실시 예에 따른 파일 복구 장치는 위에서 설명한 오라클 데이터베이스의 데이터 파일과 데이터베이스 블록 어드레스(DBA)의 특성을 고려하여 제어 파일이 손상 또는 손실되었을 경우, 데이터 파일을 이용하여 데이터 파일의 구조 정보(예를 들어, 데이터 파일의 블록 크기 및 데이터 파일의 식별 번호)를 예측할 수 있다.

[0043] 이하에서는, 본 발명의 실시 예에 따른 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법에 대하여 첨부된 도면을 참조하여 상세하게 설명한다.

[0044] 도 4는 본 발명의 실시예에 따른 디스크 블록 패턴 분석을 통한 데이터베이스 파일 복구 방법을 설명하기 위한 순서도이다.

[0045] 도 4에 도시된 바와 같이, 제어 파일이 손상 또는 손실되는 경우, 입력부(10)를 통해 오라클 데이터베이스의 데이터 파일을 입력 받는다(S10). 입력부(10)를 통해 입력된 데이터 파일은 제어부(20)로 전송된다.

[0046] 제어부(20)는 데이터 파일에 포함된 데이터베이스 블록 어드레스(DBA: database block address)로부터 블록의 크기를 예측한다(S20).

[0047] 구체적으로, 제어부(20)는 데이터파일의 맨 앞에서 설정된 오프셋 지점으로부터 설정된 크기의 제1 데이터를 획득한다(S110).

[0048] 여기서, 데이터 파일의 맨 앞에서 설정된 오프셋 지점은 0x40004지점을 의미하고, 설정된 크기의 제1 데이터는 0x40004 지점으로부터 독취한 4바이트(32비트) 데이터를 의미한다. 여기서, 설정된 오프셋 지점은 데이터베이스에서 구성 가능한 최대 블록의 크기에서 데이터베이스 블록 어드레스(DBA)가 위치하는 0x4(이하, 'DBA 오프셋'이라고 한다)를 더한 값이 된다.

[0049] 따라서, 0x40004 지점에서 독취한 4바이트 데이터는 데이터베이스 임의의 블록의 블록 어드레스(DBA: database block address)를 의미하게 된다.

[0050] 앞에서 언급한 바와 같이, 오라클 데이터베이스는 블록의 크기는 1KB에서 256KB의 크기를 갖는다. 즉, 데이터베이스에서 가능한 최대 블록의 크기는 256KB(256*1026=262144)이므로, 이를 16진수로 변환하면 0x400000이 된다. 따라서, 제1 데이터는 오라클 데이터베이스에서 설정 가능한 최대 블록의 다음 블록에서 추출된 DBA를 의미한다.

[0051] 그리고 앞에서 설명한 바와 같이, 데이터베이스 블록 어드레서(DBA)는 각 블록의 0x4 지점으로부터 4바이트 데이터이므로, 0x40004 지점으로부터 읽어들인 4바이트 데이터인 제2 데이터는 최대 크기의 블록의 다음 블록에서 읽어들인 데이터베이스 블록 어드레스(DBA)를 의미하게 된다.

[0052] 제어부(20)는 S110 단계에서 독취한 제1 데이터의 맨 뒤에서 설정된 크기의 제2 데이터를 획득한다(S120).

[0053] 여기서, 제1 데이터(DBA)의 맨 뒤에서 설정된 크기인 22비트의 제2 데이터를 독취하면, 제2 데이터가 최대 크기 블록의 다음 블록에서 읽어들인 데이터베이스 블록 어드레스(DBA)의 블록 번호를 의미하게 된다.

[0054] 제어부(20)는 S120 단계에서 최대 블록 크기를 S120 단계에서 획득한 제2 데이터로 나누어 블록의 크기를 계산한다(S130).

[0055] 예를 들면, 설정된 오프셋 지점인 0x40004 지점에서 읽어들인 4바이트 데이터(제1 데이터)의 맨 뒤에서 읽어들인 22비트 데이터(제2 데이터)가 4인 경우, 이때의 블록 번호가 4이다. 따라서, 이 블록은 앞에 4개(0번째 블록~3번째 블록)의 블록이 더 존재한다는 것을 의미하고, 최대 블록 크기인 256KB를 블록 번호 4로 나누면 각 블록의 크기는 64KB임을 예측할 수 있다.

[0056] 또 다른 예로써, 설정된 오프셋 지점인 0x40004 지점에서 읽어들인 4바이트 데이터(제1 데이터)의 맨 뒤에서 읽어들인 22비트 데이터(제2 데이터)가 1인 경우, 이때의 블록 번호가 1이다. 따라서, 이 블록은 앞에 1개(0번째 블록)의 블록이 더 존재한다는 것을 의미하고, 최대 블록 크기인 256KB를 블록 번호 1로 나누면 각 블록의 크기는 256KB임을 예측할 수 있다.

[0057] 제어부(20)는 데이터베이스 블록 어드레스(DBA: database block address)로부터 블록의 크기를 예측한 후, 데이터베이스 블록 어드레스(DBA: database block address)로부터 파일 식별번호를 예측한다(S30).

[0058] 파일 식별 번호를 예측하기 위해서는, 블록의 크기를 먼저 예측한 후, 파일 식별 번호를 예측하는 것이 바람직하다. 즉, 블록의 크기를 정확히 예측해야 데이터베이스 블록 어드레스를 데이터 파일로부터 정확히 추출할 수 있으므로, 블록 크기를 예측하는 방법이 식별 번호를 예측하는 방법에 선행되어야 한다.

[0059] 구체적으로, 제어부(20)는 앞에서 설명한 S20 단계를 통해 블록의 크기를 예측한다.

[0060] 이후, 제어부(20)는 데이터 파일의 두 번째 블록의 맨 앞에서 DBA 오프셋 지점으로부터 설정된 크기의 제3 데이터를 획득한다(S210).

[0061] 여기서, DBA 오프셋 지점은 데이터 파일의 두 번째 블록의 맨 앞에서 0x4 지점을 의미하고, 설정된 크기의 제3 데이터는 두 번째 블록 맨 앞의 0x4 지점으로부터 독취한 4바이트(32비트) 데이터를 의미한다. 즉, 0x4 지점으로부터 독취한 4바이트의 제3 데이터는 데이터베이스 블록 어드레스(DBA)를 의미하게 된다.

[0062] 상기 S20 단계에서, 데이터 파일의 블록 크기를 계산하였으므로, 두 번째 블록의 시작 지점을 알 수 있다. 따라서, 두 번째 블록의 0x4 지점에서 읽어들인 4바이트 데이터(제3 데이터)가 해당 블록의 데이터베이스 블록 어드레스(DBA)가 된다.

[0063] 제어부(20)는 S210 단계에서 독취한 제3 데이터의 맨 앞에서 설정된 크기의 제4 데이터를 독취하여 제4 데이터를 파일 식별 번호로 예측한다(S220).

[0064] 여기서, 제3 데이터(DBA)의 맨 앞에서 설정된 크기인 10비트 데이터(제4 데이터)가 데이터 파일의 식별 번호를 의미하게 된다.

[0066] 이상에서 설명한 바와 같은 본 발명의 실시 예에 따른 파일 복구 방법에 의하면, 오라클 데이터베이스를 구성하는 제어 파일이 손상 또는 손실되더라도, 데이터 파일만을 이용하여 데이터 파일의 구조 정보(예를 들어, 블록 크기 및 식별 번호)를 예측할 수 있다.

[0067] 여기에 설명되는 다양한 실시 예는 예를 들어, 소프트웨어, 하드웨어 또는 이들의 조합된 것을 이용하여 컴퓨터 또는 이와 유사한 장치로 읽을 수 있는 기록매체 내에서 구현될 수 있다.

[0068] 하드웨어적인 구현에 의하면, 여기에 설명되는 실시 예는 ASICs (application specific integrated circuits), DSPs (digital signal processors), DSPDs (digital signal processing devices), PLDs (programmable logic devices), FPGAs (field programmable gate arrays, 프로세서(processors), 제어기(controllers), 마이크로 컨트롤러(micro-controllers), 마이크로 프로세서(microprocessors), 기타 기능 수행을 위한 전기적인 유닛 중 적어도 하나를 이용하여 구현될 수 있다.

[0069] 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능과 같은 실시 예들은 별도의 소프트웨어 모듈들로 구현될 수 있다. 상기 소프트웨어 모듈들 각각은 본 명세서에서 설명되는 하나 이상의 기능 및 작동을 수행할 수 있다. 적절한 프로그램 언어로 쓰인 소프트웨어 애플리케이션으로 소프트웨어 코드가 구현될 수 있다.

[0070] 한편, 본 발명은 일 실시예에 있어서, 상술한 데이터베이스 파일 복구 장치는 방법 또는 장치를 수행하도록 하는 일련의 지시를 포함하는 컴퓨터 프로그램으로 구현될 수 있다. 즉, 본 발명은, 소정의 운영체제, 예를 들어 모바일 운영체제 내에서 구동되는 컴퓨터 프로그램일 수 있으며, 해당 프로그램은 상기 일련의 지시에 따른 프로세스를 처리하여 본 발명에 따른 방법이 실시되도록 할 수 있다.

[0071] 또한, 본 발명의 다른 일 실시예에 있어서는, 상기 컴퓨터 프로그램이 저장되어 있는, 플로피디스크, CD, DVD, 하드디스크 메모리, 다양한 종류의 메모리 등 컴퓨터 또는 이와 유사한 전자 장치로 읽을 수 있는 기록 매체일 수 있으며, 상기 기록 매체 내의 상기 컴퓨터 프로그램을 읽어들인 컴퓨터 또는 이와 유사한 전자 장치, 예를 들어 스마트폰 등은 상기 컴퓨터 프로그램상의 상기 일련의 지시에 따라 본 발명에 따른 방법을 수행할 수 있다.

[0072] 이상을 통해 본 발명의 바람직한 실시 예에 대하여 설명하였지만, 본 발명은 이에 한정되는 것이 아니고 특허청구범위와 발명의 상세한 설명 및 첨부한 도면의 범위 안에서 여러 가지로 변형하여 실시하는 것이 가능하고 이 또한 본 발명의 범위에 속하는 것은 당연하다.

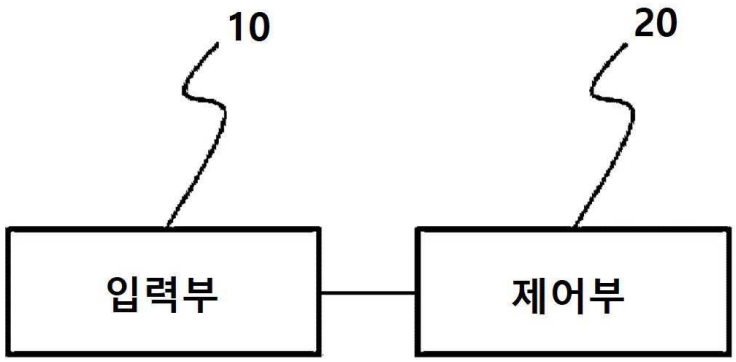
부호의 설명

[0073] 10: 입력부

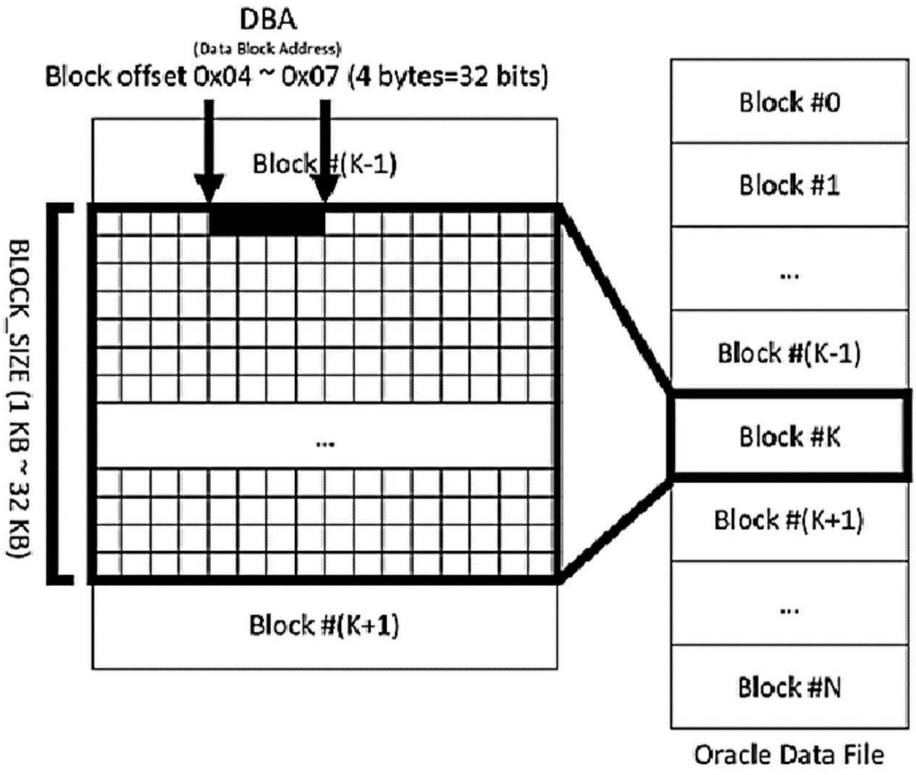
20: 제어부

도면

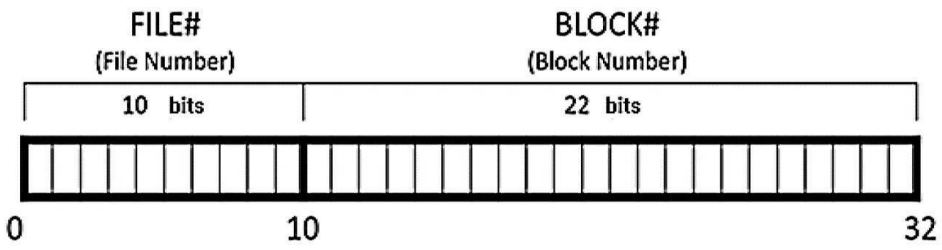
도면1



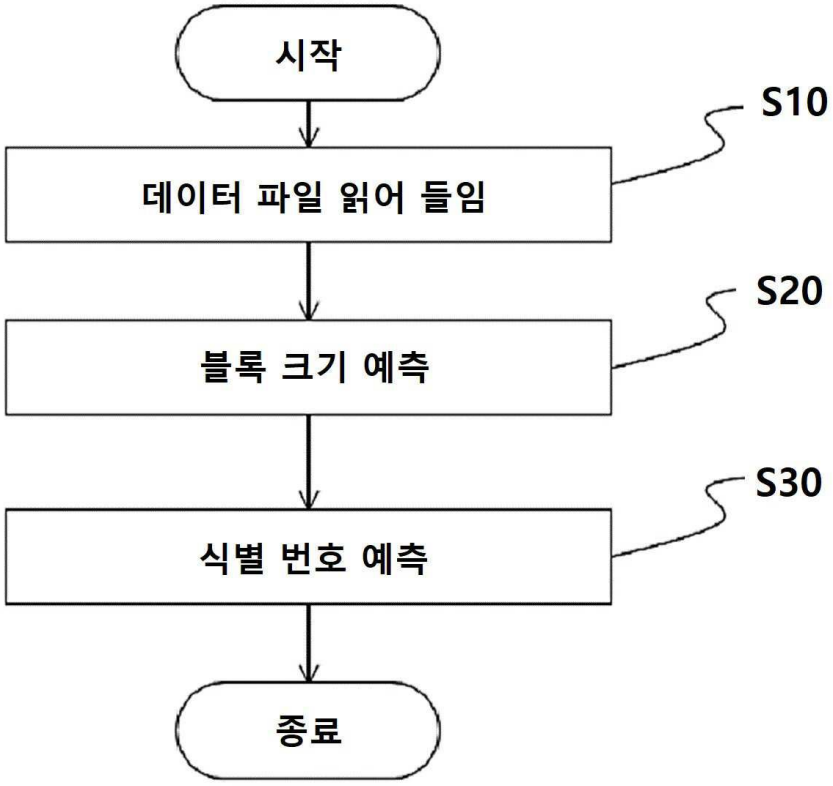
도면2



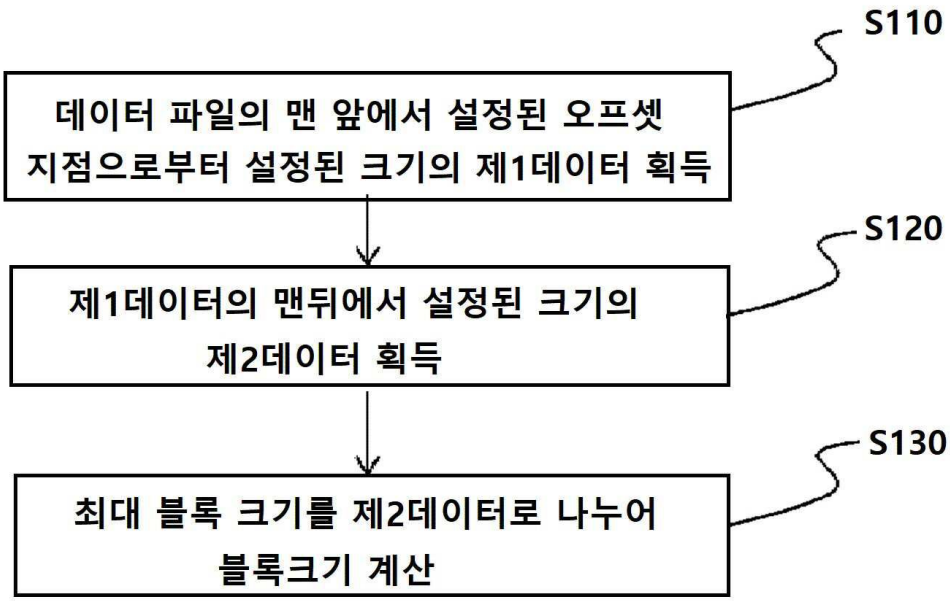
도면3


FILE#
(File Number)
BLOCK#
(Block Number)
10 bits
22 bits
0
10
32

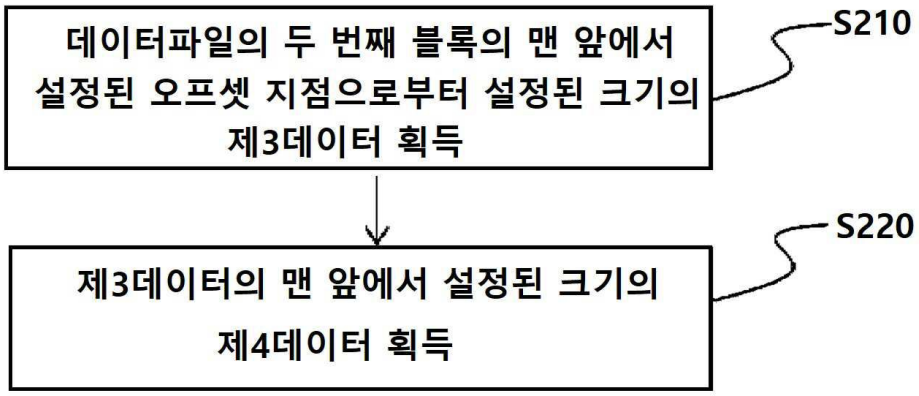
도면4


시작
S10
데이터 파일 읽어 들임
S20
블록 크기 예측
S30
식별 번호 예측
종료

도면5


S110
데이터 파일의 맨 앞에서 설정된 오프셋 지점으로부터 설정된 크기의 제1데이터 획득
S120
제1데이터의 맨뒤에서 설정된 크기의 제2데이터 획득
S130
최대 블록 크기를 제2데이터로 나누어 블록크기 계산

도면6


데이터파일의 두 번째 블록의 맨 앞에서 설정된 오프셋 지점으로부터 설정된 크기의 제3데이터 획득
S210
제3데이터의 맨 앞에서 설정된 크기의 제4데이터 획득
S220