

정보처리기사 실기

데이터베이스 2

데이터베이스 2

- ▶ 가칭 이산교통문화진흥원은 보다 쾌적한 교통 환경을 구현하기 위해 설립된 가상기관으로 교통환경분석사 자격관리시스템을 구축하여 자격체제에 대한 자동화 관리를 수행하고 있는데 성능이 향상된 데이터베이스 하드웨어 서버를 도입하였음에도 불구하고 수험생으로부터 불만이 제기되어 (주)홍국영선컨설팅사에 데이터베이스 튜닝을 의뢰하게 되었고 (주)홍국영컨설팅사에서는 귀하를 교통환경분석사 자격관리시스템의 성능향상을 위한 컨설턴트로 파견하였다.
- ▶ 귀하는 DBMS에서 성능 확보의 방안으로 각종 데이터베이스 관련 매개변수 설정, 논리모델링에서 데이터의 무결성을 확보하기 위해 수행된 정규화에 대해 데이터의 중복을 허용하는 역정규화는 물론 검색빈도가 높은 (①)에 (②) 설정에 관한 검토가 중요한 점을 감안하여 기존 데이터베이스에 대해 분석해 본 결과 성능과 관련한 여러 가지 현상이 파악되었다.
- ▶ 우선 논리 데이터베이스 모델링에 대한 검토한 결과 데이터 중복이 어느 정도 적당하게 이루어져 있어 역정규화를 고려할 필요는 없다고 판단되었으며, PK에 대한 Unique (②)가(이) Default로 구현되고 있는 사실을 확인하였으나, 테이블별 데이터량 및 검색빈도 등의 분석을 통한 (②) 설계 및 구현이 적절하게 이루어지고 있지 않는 사실이 인지되었다.

데이터베이스 2

[그림1] 논리 데이터베이스 모델의 일부 예

교시별시험과목

시행년도: <u>varchar(4)</u> (Fk)
시행구분: <u>varchar(2)</u> (Fk)
회차: <u>varchar(2)</u>
응시자격코드: <u>varchar(1)</u> (Fk)
차수: <u>varchar(2)</u> (Fk)
1교시 과목: <u>varchar(2)</u>
2교시 과목: <u>varchar(2)</u>
비고: <u>varchar(50)</u>
등록자: <u>varchar(10)</u>
등록일자: <u>datetime</u>

시행계획_응시자격

시행년도: <u>varchar(4)</u>
시행구분: <u>varchar(2)</u>
차수: <u>varchar(2)</u>
응시자격코드: <u>varchar(1)</u>
비고: <u>varchar(50)</u>
등록자: <u>varchar(10)</u>
등록일자: <u>datetime</u>



데이터베이스 2

[그림2] 테이블 정의서(교시별시험과목)의 예

테이블 정의서							
시스템명	시험관리					작성일	2008. 00. 00
테이블ID	TB_TR_CLASS_ITEM					작성자	박달호
테이블명	교시별 시험과목						
NO	컬럼ID	컬럼명	형태	길이	KEY	NULL	비고
1	OPER_YEAR	시행년도	VARCHAR	4	PK	N	
2	OPER_SEC	시행구분	VARCHAR	2	PK	N	
3	OPER_SEQ	회차	VARCHAR	2	PK	N	
4	EXAM_CODE	응시 자격코드	VARCHAR	1	PK	N	
5	OPER_SEP	차수	VARCHAR	2	PK	N	
6	CLASS_1	1교시과목	VARCHAR	2		Y	
7	CLASS_2	2교시과목	VARCHAR	2		Y	
8	REMARK	비고	VARCHAR	2		Y	
9	REG_ID	등록자	VARCHAR	50		Y	
10	REG_DATE	등록일자	DATETIME	10		Y	

데이터베이스 2

[그림3] (㉔) 속성의 일부 예

(㉔) 속성					
테이블 이름	TB_TR_CLASS_ITEM				
(㉔) 이름	PK_TR_CLASS_ITEM				
(㉔) 유형	클러스터형				
고유	<input checked="" type="checkbox"/>				
(㉔) 키열					
이름	정렬순서	데이터형식	크기	ID	NULL 허용
OPER_YEAR	오름차순	varchar(4)	4	아니오	아니오
OPER_SEC	오름차순	varchar(2)	2	아니오	아니오
OPER_SEQ	오름차순	varchar(2)	2	아니오	아니오
OPER_CODE	오름차순	varchar(1)	1	아니오	아니오

데이터베이스 2

- ▶ 또한, 데이터 컬럼이 [그림4]와 같이 대부분 (③) (으)로 설정되고 있는바, 물리적인 한 블록 공간에 데이터가 모두 입력된 경우, 이 블록 공간에 입력된 (③) 형태의 데이터 수정시 다른 블록으로 (④) 현상이 발생하게 되며, (④) 현상이 발생한 데이터를 검색하면, 먼저 저장되었던 블록을 검색한 후 이주된 블록을 검색하게 됨으로써 검색속도를 떨어뜨리게 되는 문제가 발생하고 있었다.

[그림4] 테이블 정의서(시험장소코드)의 예

테이블 정의서							
시스템명		시험관리			작성일	2008. 00. 00	
테이블ID		TB_TR_EXAM_PLACE			작성자	박달호	
테이블명		시험장소코드					
NO	컬럼ID	컬럼명	형태	길이	KEY	NULL	비고
1	OPER_YEAR	시행년도	VARCHAR	4	PK	N	
2	OFFICE_CODE	지사	VARCHAR	2	PK	N	
3	EXAMROOM_CODE	시험장소코드	VARCHAR	4	PK	N	
4	POST_NO	우편번호	VARCHAR	6	PK	Y	
5	ADDRESS	주소	VARCHAR	150	PK	Y	
6	PHONE_NO	전화번호	VARCHAR	20		Y	
7	POSITION	위치	VARCHAR	50		Y	
8	CLASS_NUM	교실수	DECIMAL	5.0		Y	
9	PERSONS_NUM	인원	DECIMAL	5.0		Y	
10	USE_YN	사용여부	VARCHAR	1		Y	
11	REMARK	비고	VARCHAR	50		Y	
12	REG_ID	등록자	VARCHAR	10			
13	REG_DATE	등록일자	DATETIME				

데이터베이스 2

- ▶ [그림4]의 시험장소코드 테이블 정의서에서 알 수 있는 바와 같이 주소, 전화번호, 위치, 약도 URL 컬럼은 데이터의 크기가 비교적 큰 차이를 보이지 않을 경우, 검색속도의 향상을 위해 가변형의(③)형태 대신 고정형(⑤)형태의 설정을 고려해 볼 수 있을 것으로 판단되었다.
- ▶ 데이터베이스 매개변수 및 (②), 메모리 등은 물리적인 생성을 어떻게 하느냐에 따라 응용시스템의 처리성능에 직접적으로 영향을 주는 요소들이 된다. 따라서 설계 단계에서 설정된 데이터베이스 관련 매개변수(Parameter) 및 (②) 등에 대하여 성능을 향상시킬 수 있는 방향으로 설정하고 처리성능을 고려하여 일관성 있게 물리적으로 생성이 되도록 하며, 시험단계에서 안정적으로 성능 점검이 될 수 있도록 실행 가능한 수준으로 구체화하도록 하는 것이 좋은 것으로 판단되었다. 따라서, 데이터베이스의 성능향상을 위해서 제일 먼저 검색빈도가 높은(①)에 대해 (②)설정을 자문하게 되었고, 테이블별 데이터량 및 (①)별 검색빈도를 분석하여 (②)선정 및 구현을 권고하게 되었다.
- ▶ 귀하가 (②)선정과 관련하여 고려한 절차는 다음과 같다.
- ▶ - 모든 액세스 형태 수집- 대상(①)선정 및 분포도 조사
- ▶ - 반복 수행되는 액세스 경로의 해결
- ▶ - 클러스터링 검토(도입 데이터베이스 특성 고려)
- ▶ - 시험 생성 및 테스트
- ▶ 또한, 데이터 형태를 (③)(으)로 설정하게 되면 물리적인 블록공간에서 (④)현상의 발생으로 검색속도가 떨어지게 되는 점을 감안하여 현재 대부분 (③)형태로 설정된 (①)들을 면밀히 검토하여, (①)데이터 항목들의 크기가 비교적 큰 차이를 보이지 않을 경우 고정형인 (⑤)형태의 설정을 고려해 볼 필요가 있는 것으로 파악되었다.

정답

답 항 보 기	1	스택(stack)	2	트리(tree)	3	차수(degree)	4	관계(relation)	5	char
	6	매핑(mapping)	7	모델(model)	8	블록(block)	9	영역(domain)	10	클래스(class)
	11	특성(specialty)	12	컬럼(column)	13	행(row)	14	스큐잉(skewing)	15	실린더(cylinder)
	16	연관화 (association)	17	열 이주 (column migration)	18	열 전이 (column transition)	19	속성 전이 (attribute transition)	20	영역재배치 (domain relocation)
	21	외래키(FK)	22	이상(anomaly)	23	인덱스(index)	24	주키(PK)	25	체인(chain)
	26	캡슐화 (encapsulation)	27	섹터 전이 (sector transition)	28	클러스터링 (clustering)	29	클러스터 전이 (cluster transition)	30	섹터링 (sectoring)
	31	섹터시프트 (sector shift)	32	블록이주 (block migration)	33	행이주 (row migration)	34	행전이 (row transition)	35	레코딩 (recording)
	36	datetime	37	decimal	38	image	39	int	40	varchar

1. 컬럼(column)
2. 인덱스(index)
3. Varchar
4. 행이주(row migration)
5. char