

정보처리기사 실기

데이터베이스 5

데이터베이스 5

- ▶ 가칭 미래한국정보기술진흥원은 우리나라의 미래를 결정하는 IT기술의 발굴 및 진흥을 위한 가상기관이다.
- ▶ 이 기관에서는 미래 한국의 IT기술과 관련한 정보를 온라인상에서 국민에게 제공하는 정보시스템 구축 사업을 발주하게 되었다.
- ▶ 귀하가 속한 가칭 (주)금란정보시스템이 해당 사업을 수주하여 구축 프로젝트가 진행되었고, 귀하는 정보처리기사로서 구축되는 정보시스템의 데이터베이스 부분의 (①)을(를) 담당하게 되었다.
- ▶ (①)(이)라 함은 “조율” 또는 “조정”이라는 의미를 가지고 있지만, 데이터베이스 세계에서는 ‘최적화’라는 개념으로 이해할 수 있다. (①)이(가) 이루어지게 되면 업무의 최적화, 하드웨어적인 병목 현상 해결, SQL의 최적화 등 여러 가지 개선을 도모할 수 있다. 이를 통해서 처리 속도의 향상 등 성능을 제고시키고 사용자가 필요한 때에 원하는 정보를 보다 원활하게 제공받을 수 있도록 할 수 있다.
- ▶ 데이터베이스 (①)(이)라는 것이 데이터베이스를 이용함에 있어서 가급적 현재보다 효율적인 시스템 자원을 사용하고, 최소한의 디스크 입/출력을 도모하는 등 최상의 성능과 최적의 환경을 유지하도록 조정해주는 것임을 감안해 볼 때 성능에 대한 부분을 생각하지 않을 수 없다.
- ▶ 귀하는 우선 시스템의 성격에 대해 검토한 결과 본 시스템이 삽입, 수정, 삭제 작업보다는 조회가 위주로 되는 시스템임을 알게 되었다. 삽입, 수정, 삭제 등의 작업이 빈번하게 발생하는 테이블에서는 인덱스가 많을 경우 매번 인덱스를 수정해야 하는 관계상 성능이 저하될 수 있지만, 본 시스템의 경우 조회가 대부분이므로 테이블에 인덱스가 설정되어 있는 편이 유리하다.
- ▶ 이처럼 테이블에 대한 조회 위주로 되어 있는지 삽입, 수정, 삭제 작업이 주로 발생하는지를 고려하여 적당한 인덱스 개수를 지정하는 것은 데이터베이스 (①)에서 의미가 있다.

데이터베이스 5

- ▶ 인덱스를 검토할 때는 키에 대한 고려가 필요하다.
- ▶ 외래키에 대해서는 가급적 인덱스를 생성하여 전체 스캔이 발생하는 경우와 불필요하게 발생하는 (②)를(을) 피하도록 한다. (②)(이)라는 것은 데이터베이스에서 동시성으로 인해 존재하지 않는 데이터를 읽게 되거나, 변경된 데이터가 손실되는 등의 부정적인 효과를 방지하기 위한 것이다. 즉, 데이터베이스 관리에서 하나의 트랜잭션이나 세션에 사용되는 데이터를 다른 트랜잭션이나 세션은 접근하지 못하게 하는 것이다.
- ▶ (②)의 단위가 크면 병행성 수준이 낮아지고, 단위가 작으면 병행성 수준이 높아진다.
- ▶ (②)의 종류에는 여러 가지가 있다. 행 수준의 (②)는(은) 페이지 수준이나 테이블 수준의 (②)보다 데이터베이스 공유도가 증가하고 시스템 오버헤드도 증가한다.
- ▶ 기본키는 일반적으로 지정된 순서대로 인덱스를 생성하므로 키를 구성하는 컬럼 순서를 복합 컬럼 인덱스를 지정하는 규칙에 따라 나열한다. 식별자가 너무 길거나 여러 개의 속성으로 구성되는 경우 작업의 수행 속도가 떨어질 수 있기 때문에 인공키를 추가하는 것도 필요하다.
- ▶ 또한, 인덱스를 설계할 때에는 (③)가(이) 좋도록 설계해야 한다.
- ▶ 인덱스의 (③)는(은) 인덱스의 키 값당 행의 개수를 기준으로 산정한다.
- ▶ 낮은 (③)를(을) 가지는 인덱스는 키 값당 많은 수의 행을 가지는 인덱스로서, 남자의 경우에는 M, 여자의 경우에는 F의 두 가지 값만 가지는 성별과 같은 컬럼에 있는 인덱스가 이에 해당한다.
- ▶ 좋은 (③)를(을) 가지는 인덱스는 고객번호 컬럼에 있는 인덱스와 같이, 인덱스 키당 하나 또는 소수의 값을 가진다. 고유 인덱스는 가장 좋은 (③)를(을) 가진다.

데이터베이스 5

- ▶ 인덱스는 (④) 인덱스와 (⑤) 인덱스를 생각할 수 있다.
- ▶ (④) 인덱스는 인덱스가 가리키는 데이터가 이미 정렬되어 있어 최종 인덱싱 단계 없이 바로 데이터를 찾을 수 있는 특징을 가지고 있다.
- ▶ (④) 인덱스는 조회시 (⑤) 인덱스 보다 빠른 성능을 보인다.
- ▶ 그러나 삽입, 수정의 경우 변경된 내용을 인덱스에 반영하고 재정렬해야 하므로 (⑤) 인덱스보다 불리하며, 한 테이블에 하나의 인덱스만 만들 수 있다.
- ▶ (⑤) 인덱스 조회시 (④) 인덱스보다 성능은 떨어지나 한 테이블에 여러 개의 인덱스 설정이 가능한 특징이 있다.
- ▶ 인덱스의 정렬 순서를 오름차순이나 내림차순으로 조절하는 것은 여러 개의 컬럼들에 대하여 정렬을 하면서 그 컬럼들에 대하여 서로 다른 정렬 순서를 사용할 때 필요하다. 자주 정렬하는 컬럼에 대해서 (④) 인덱스를 만들어 주면 결과를 정렬해야 하는 오버헤드가 제거되어 성능에 도움이 된다.
- ▶ 인덱스는 각각의 값을 찾기 위하여 B-Tree를 탐색하기 때문에 많은 양의 데이터를 읽어 올 때에는 인덱스를 사용하는 것이 효율적이지 못하다.
- ▶ (④) 인덱스는 실제 데이터의 순서와 인덱스의 순서가 일치하기 때문에 일정한 범위를 주고 찾는 경우 속도 향상에 매우 도움이 된다. (⑤) 인덱스는 한 특정 값을 찾거나, 많은 양의 데이터 중에서 작은 범위를 찾을 때만 유용하다.
- ▶ 귀하는 결론적으로 본 시스템이 조회성이 강한 시스템으로서 특정 행 단위 위주의 액세스가 거의 없는 점을 감안하여 데이터베이스 튜닝에 있어서 (④) 인덱스의 사용을 고려하게 되었다.

데이터베이스 5 정답

1. 튜닝(Tuning)
2. 로킹(Locking)
3. 선택성(Selectivity)
4. 클러스터드(Clustered)
5. 언클러스터드(Non-Clustered)

답 항 보 기	1	Monitor	2	Degree	3	도메인 (Domain)	4	격리 (Isolation)	5	교착상태 (Deadlock)
	6	기본 (Primary)	7	언클러스터드 (Non-Clustered)	8	독립성 (Independence)	9	동시성 (Concurrency)	10	튜닝 (Tuning)
	11	보충 (Supplement)	12	보호 (Protection)	13	분산도 (Distributed Rate)	14	관계대수 (Relational Algebra)	15	선택성 (Selectivity)
	16	릴레이션 (Relation)	17	언차터드 (Unchartered)	18	언체인드 (Unchained)	19	오버라이드 (Override)	20	오버로드 (Overload)
	21	오버헤드 (Overhead)	22	외래 (Foreign)	23	인공 (Artificial)	24	로킹 (Locking)	25	직렬성 (Serializability)
	26	트랜잭션 (Transaction)	27	차터드 (Chartered)	28	체인드 (Chained)	29	추가 (Addition)	30	클러스터드 (Clustered)
	31	OLAP	32	확장 (Extension)	33	스키마 (Schema)	34	후보 (Candidate)	35	희소 (Sparse)
	36	개체 (Entity)	37	Tree	38	System catalog	39	View	40	Cardinality