**데이터베이스 모니터링 및 최적화 개요**

**데이터베이스 모니터링**

• 데이터베이스 모니터링은 데이터베이스 시스템의 성능, 상태, 자원 사용량 등을 지속적으로 감시하고 분석하는 과정입니다.

• 데이터베이스의 실시간 운영 상태를 파악하기 위해 다양한 도구와 기술을 사용하여 주요 성능 지표(CPU 사용량, 메모리 사용량, 디스크 I/O 등)를 추적합니다.

**모니터링의 목적**

• 시스템 내에서 발생하는 이벤트와 로그를 수집하고 분석하여 잠재적 문제를 조기에 발견하고 대응합니다.

• 실시간 알림 설정을 통해 성능 저하나 장애 발생 시 즉시 인지하고 신속하게 대응할 수 있도록 합니다.

• 장기적인 성능 데이터를 기반으로 시스템의 트렌드를 분석하고, 성능 최적화 및 확장 계획을 수립합니다.

**데이터베이스 최적화**

• 최적화는 데이터베이스 시스템의 성능을 향상시키기 위해 구조, 쿼리, 인덱스 등을 개선하는 과정입니다.

• 최적화를 통해 자원 사용을 효율적으로 관리하고, 응답 속도를 단축시켜 사용자 경험을 향상시키며, 운영 비용을 절감합니다.

**최적화의 목적**

• 쿼리 성능을 향상시키기 위해 인덱스를 추가하거나 조정하고, 비효율적인 쿼리를 재작성합니다.

• 데이터베이스의 테이블 구조를 정규화 또는 비정규화하여 데이터 접근 속도를 개선합니다.

• 데이터 분할(파티셔닝) 및 아카이빙을 통해 큰 데이터를 효율적으로 관리합니다.

• 캐싱을 통해 반복적인 데이터 접근 속도를 높이고, 시스템 자원(CPU, 메모리, 디스크 I/O 등)의 사용을 최적화합니다.

**데이터베이스 모니터링과 최적화**

• 모니터링을 통해 수집된 데이터는 최적화의 기반이 됩니다. 성능 지표와 로그를 분석하여 최적화가 필요한 부분을 식별할 수 있습니다.

• 모니터링은 최적화 후의 효과를 검증하는 데 중요한 역할을 합니다. 최적화 전후의 성능 변화를 비교하여 개선 결과를 확인합니다.

• 지속적인 모니터링을 통해 최적화의 효과를 유지하고, 새로운 성능 문제를 사전에 발견하여 추가적인 최적화를 수행할 수 있습니다.

• 모니터링과 최적화는 함께 데이터베이스의 안정성과 성능을 극대화하는 데 필수적인 과정입니다.

• 모니터링으로 문제를 발견하고, 최적화로 문제를 해결하여 시스템의 효율성을 높입니다.

**왜 쿼리가 점점 느려질까?**

**• 데이터 양 증가**

• 테이블의 데이터가 늘어나면서 쿼리 처리 시간이 길어집니다.

• 인덱스가 커져서 검색 속도가 느려질 수 있습니다.

**• 자원 사용 경쟁**

• 동시에 실행되는 쿼리와 작업이 늘어나면서 CPU, 메모리, 디스크 I/O 등의 자원 경쟁이 발생합니다.

• 잠금 경합으로 인해 쿼리 처리 속도가 느려질 수 있습니다.

**• 애플리케이션 변경**

• 애플리케이션 코드가 변경되면서 쿼리의 호출 빈도나 방식이 바뀔 수 있습니다.

• 새로운 기능이 추가되면서 시스템 부하가 증가할 수 있습니다.

**-> 데이터베이스 모니터링을 통해 문제를 사전에 방지하는 것이 중요합니다.**

**데이터베이스 모니터링**

**모니터링할 지표**

**CPU 사용량**

• 데이터베이스 서버는 CPU를 이용하여 쿼리 처리와 데이터 연산을 수행합니다.

• 과도한 CPU 사용은 성능 저하를 초래할 수 있으며, 서버의 과부하 상태를 파악하고 조치를 취하는 것이 중요합니다.

• **메모리 사용량**

• 데이터베이스는 데이터 캐싱과 쿼리 실행 시 메모리를 사용합니다.

• 메모리 부족은 쿼리 성능 저하와 시스템 불안정을 유발할 수 있습니다.

**• 디스크 I/O**

• 데이터베이스의 읽기 및 쓰기 작업은 디스크 I/O에 의존합니다.

• 디스크 I/O 성능이 낮으면 데이터 접근 속도가 느려지고 전체 시스템 성능에 영향을 줄 수 있습니다.

**네트워크 트래픽**

• 데이터베이스 서버와 클라이언트 간의 데이터 전송이 네트워크를 통해 이루어집니다.   
• 네트워크 트래픽이 과도하면 데이터 전송 속도가 저하되고 응답 시간이 길어질 수 있습니다.

**• 쿼리 실행 시간**

• 쿼리 실행 시간이 긴 슬로우 쿼리는 성능 문제를 야기할 수 있습니다.

• 슬로우 쿼리를 모니터링하고 분석하여 비효율적인 쿼리와 인덱스 문제를 발견하고 최적화하는 것이 중요합니다.

**모니터링할 로그**

• **에러 로그**

• 목표: 시스템 장애, 예외 상황, 오류 발생 원인 파악

• 내용: 시스템 오류, 쿼리 실패, 연결 문제 등

• 도구: 데이터베이스 관리 도구에서 제공하는 에러 로그 뷰어

• **접속 로그**

• 목표: 데이터베이스 접근 패턴 분석, 비정상적 접근 시도 탐지

• 내용: 사용자 로그인, 접속 시간, 접속 IP 주소

• 도구: 데이터베이스의 접속 로그 기능, 보안 모니터링 도구

• **백업 로그**

• 목표: 백업 작업의 성공 여부 확인, 백업 주기 및 상태 모니터링

• 내용: 백업 시작 및 종료 시간, 백업 성공/실패 여부

• 도구: 데이터베이스 백업 관리 도구, 백업 로그 분석 도구

**모니터링 알람**

• 모니터링 알람은 시스템의 성능 지표나 상태가 설정된 임계값을 초과할 때 자동으로 경고를 보내는 기능입니다.

• 이를 통해 시스템 관리자나 개발자는 성능 문제를 조기에 감지하고 신속하게 대응할 수 있습니다.

• **알람 조정 및 최적화**

• 경고 레벨 설정: 경고의 심각도에 따라 다양한 레벨 설정 (정보, 경고, 심각)

• 알람 빈도 설정: 동일 알람의 빈도를 조정하여 과도한 경고 방지

• 알람 필터링: 중요도가 낮은 알람을 필터링하여 불필요한 경고 줄이기

**데이터베이스 최적화**

**다양한 쿼리**

• 데이터를 조회할 때 여러 가지 쿼리 방법이 존재합니다.

• 효율적인 쿼리를 사용하면 성능을 개선하고 자원 사용을 줄일 수 있습니다.

• 최적화된 쿼리는 데이터베이스 응답 속도를 빠르게 하고 시스템 자원을 절약합니다.

**효율적인 쿼리 사용의 중요성**

• **성능 향상**

• **쿼리 최적화:** 동일한 데이터를 조회하는 여러 쿼리 중에서 가장 효율적인 쿼리를 사용하면 데이터베이스의 성능을 향상시킬 수 있습니다. 효율적인 쿼리는 데이터 검색 속도를 빠르게 하고, 시스템 자원의 사용을 최소화합니다.

**• 자원 절약**

**• 자원 사용 감소:** 비효율적인 쿼리는 데이터베이스 서버의 CPU, 메모리, 디스크 I/O 자원을 과도하게 소모할 수 있습니다. 최적화된 쿼리는 자원 사용을 줄여 서버의 부하를 감소시킵니다.

**응답 시간 단축**

**• 빠른 응답:** 효율적인 쿼리는 데이터베이스가 요청을 처리하는 시간을 줄여 사용자에게 더 빠른 응답을 제공합니다. 이는 사용자 경험을 개선하고 시스템의 전반적인 반응 속도를 높입니다.

**• 비용 절감**

**• 운영 비용 절감:** 성능이 향상되면 하드웨어와 클라우드 서비스의 비용을 줄일 수 있습니다. 자원을 효율적으로 사용하는 쿼리는 운영 비용을 절감하는 데 기여합니다.

**인덱싱**

• 데이터베이스에서 테이블의 특정 컬럼에 대한 인덱스를 생성하여 데이터 검색 속도를 향상시키는 기법입니다.

• 일반적으로 B-트리 또는 해시 구조를 사용하여 데이터 검색 시 빠르게 접근할 수 있도록 합니다.

• **장점**

• 인덱스를 사용하면 테이블 전체를 스캔하지 않고 빠르게 검색할 수 있습니다.

• 조회 쿼리 성능을 크게 개선합니다.

• 정렬(ORDER BY) 및 집계(GROUP BY) 작업의 성능을 향상시킵니다.

**• 단점**

• 인덱스는 추가 디스크 공간을 사용합니다.

• 데이터 삽입, 삭제, 수정 시 인덱스도 갱신해야 하므로 쓰기 성능에 영향을 미칠 수 있습니다.

• 인덱스가 많아지면 관리 및 유지보수가 복잡해질 수 있습니다.

**인덱스 종류**

**• 단일 컬럼 인덱스**

• 하나의 컬럼에 인덱스를 생성합니다.

• 단일 컬럼에 대한 검색 성능을 개선합니다. • 복합 쿼리나 다중 컬럼 검색 시 성능이 제한적일 수 있습니다.

• **복합 인덱스**

• 여러 컬럼을 포함하는 인덱스입니다.

• 여러 컬럼을 동시에 검색할 때 성능을 개선합니다.

• 여러 컬럼을 포함하여 디스크 공간을 더 많이 사용합니다.

**• 풀텍스트 인덱스**

• 텍스트 검색을 최적화하기 위해 문서 내 텍스트를 인덱싱합니다.

• 인덱스 순서에 따라 데이터가 정렬되어 있어 범위 검색이 효율적입니다.

• 텍스트 인덱스는 상대적으로 큰 디스크 공간을 소모할 수 있습니다.

**비정규화를 통한 성능 개선**

**• 조회 성능 향상:** 자주 조회되는 데이터를 한 테이블에 저장하여 조인 연산을 줄입니다.

**• 쿼리 복잡성 감소:** 복잡한 조인 연산을 줄여 쿼리 작성과 실행을 단순화합니다.

**비정규화의 장점**

**• 성능 향상:** 데이터 조회와 집계 성능이 크게 개선됩니다.

• **쿼리 간소화:** 복잡한 쿼리를 간단하게 만들어 실행 속도를 증가시킵니다**.**

**• 자원 절약:** 서버 자원의 효율적인 사용으로 응답 시간을 단축시킵니다.

**비정규화의 단점**

**• 데이터 중복:** 데이터 중복 저장으로 인해 저장 공간이 증가할 수 있습니다.

**• 데이터 무결성:** 중복된 데이터로 인해 무결성 유지가 복잡해질 수 있습니다.

**• 데이터 업데이트:** 중복된 데이터의 업데이트가 복잡하고 오류가 발생할 가능성이 높습니다.

**쿼리 캐싱**

자주 실행되는 쿼리의 결과를 저장해두고, 동일한 쿼리가 다시 실행될 때 저장된 결과를 즉시 반환하여 성능을 개선하는 기법입니다.

**작동 원리**

**• 쿼리 실행 결과 저장:** 데이터베이스는 쿼리의 실행 결과를 메모리나 디스크에 저장합니다.

**• 쿼리 요청 시 캐시 사용:** 동일한 쿼리가 요청되면, 데이터베이스는 저장된 결과를 반환하고, 실제 데이터베이스 접근을 피합니다.

• **캐시 무효화:** 데이터베이스에 데이터 변경이 발생하면, 관련된 캐시를 무효화하거나 갱신합니다.

**쿼리 최적화 모범 사례**

**• 쿼리 리팩토링**

• 서브쿼리 대신 조인을 사용하거나, 불필요한 서브쿼리를 제거합니다.

• SELECT \* 대신 필요한 컬럼만 선택하여 성능을 개선합니다.

**• 쿼리 재사용**

• 자주 사용하는 쿼리를 최적화하여 재사용합니다.

• 이러한 재사용 쿼리를 캐싱하여 시스템 성능을 개선할 수 있습니다.

**• 인덱스 최적화**

• 쿼리에서 자주 검색되는 컬럼, 정렬, 조인 조건에 대해 인덱스를 생성합니다.

• 여러 컬럼을 동시에 검색하는 경우 복합 인덱스를 사용합니다.

**• 쿼리 분석**

• EXPLAIN 명령어를 사용하여 쿼리 실행 계획을 분석하고 최적화합니다.

**• 데이터베이스 설계 최적화**

• 데이터베이스를 필요에 따라 비정규화하여 성능을 개선합니다.

• 데이터 모델을 설계할 때 적절한 데이터 타입과 길이를 사용하여 저장 공간과 성능을 최적화합니다.

**데이터베이스 구조 최적화: 파티셔닝**

• 큰 테이블을 작은 부분으로 나누어 검색 및 쿼리 성능을 개선합니다.

**적용 사례**

• 로그 데이터: 시간 기반의 로그 데이터를 날짜별로 파티셔닝하여 성능을 최적화합니다.

• 대규모 사용자 데이터: 사용자 ID를 해시하여 분산 저장함으로써 검색 성능을 개선합니다.

**장점**

• 쿼리 성능을 향상시키고 대규모 데이터의 검색 속도를 높입니다.

• 파티션 단위로 데이터 백업 및 복구가 가능하여 관리가 용이합니다.

**단점**

• 파티션이 많아질수록 관리가 복잡해질 수 있습니다.

• 잘못된 파티셔닝 전략은 오히려 성능을 저하시킬 수 있습니다.

**아카이빙 (Archiving**)

• 아카이빙은 오래된 데이터나 사용하지 않는 데이터를 별도의 저장소에 보관하여 현재의 데이터베이스 시스템에서 성능을 유지하고 저장 공간을 절약하는 과정입니다.

• 아카이빙된 데이터는 일반적으로 조회 빈도가 낮지만 법적 요구 사항이나 향후 분석 필요성에 따라 보존됩니다.

**목표**

**• 성능 향상:** 데이터베이스의 크기를 줄여 쿼리 성능을 유지하고, 데이터베이스 관리의 복잡성을 줄입니다.

• **저장 공간 절약:** 자주 사용하지 않는 데이터를 별도로 보관하여 현재 운영 중인 데이터베이스의 저장 공간을 절약합니다.

**• 법적 요구 사항 준수:** 법적 요구 사항이나 규제에 따라 일정 기간 동안 데이터를 보존해야 할 경우 이를 충족하기 위해 아카이빙합니다.

**• 데이터 복구 및 분석:** 장기적으로 보존해야 할 데이터를 아카이빙하여 필요할 때 복구하거나 분석할 수 있도록 합니다.

**아카이빙 전략**

**• 정책 정의:** 아카이빙 정책을 수립하여 어떤 데이터를, 언제, 어떻게 아카이빙할지 결정합니다.

• 데이터 선택 기준, 아카이빙 주기, 아카이빙 저장소, 데이터 보존 기간

• 모니터링 및 유지보수, 백업 및 복구 계획, 보안 및 접근 제어

**• 테스트 및 검증:** 아카이빙 전략을 구현하기 전에 테스트하여 데이터의 정확성과 복구 가능성을 검증합니다.

• **모니터링 및 유지보수:** 아카이빙 작업의 성과를 모니터링하고 필요에 따라 조정합니다. 지속적으로 아카이빙 전략을 평가하고 개선합니다.