

# DB 관련 고민






## 목표

- 대용량 데이터 처리
  - 2~3백만개의 데이터를 효과적으로 처리할 수 있는 데이터베이스 시스템 설계
  - 2~3백만개의 데이터 속에서 찾고자하는 데이터를 어떻게하면 최대한 빠르게 조회할 수 있을까
- 데이터베이스 성능 최적화 : 데이터베이스 성능 최적화를 위한 다양한 기법 적용해보기 ex) 인덱싱, 모델링, 쿼리문 최적화 등
- 실시간 데이터 처리 : 실시간으로 수많은 데이터가 들어오고 나가는 것 구현

고민 주제	방향	추가사항
mongoDB vs MariaDB	<ul style="list-style-type: none"><li>• 몽고디비의 경우 json 형태라서 iot의 데이터 입출력이 편안하고 string 형태의 데이터 입출력이 좋다</li><li>복잡하고 다양한 형태의 데이터를 다룬다면 몽고디비</li><li>• mysql의 경우 장기간의 데이터 보관과 보안성, 설계에 따른 성능 향상이 더 크다. 익숙하고 일반적인 sql 형태이다.</li><li>구조화된 데이터와 복잡한 관계를 가진 데이터를 다룬다면 마리아</li></ul>	<p>(조언) 게임 로그 등과 같은 빠른 데이터 저장 및 일정 기간만의 저장이 있을 경우 몽고 디비가 효과적이다.</p> <p>필요한 로그의 경우 RDB에 따로 저장하여 사용하였다.</p> <p>개인정보의 경우 법적으로 일정기간 이상 저장해야 하는것이 정해져 있다.</p>
DB 구조 설계	<ul style="list-style-type: none"><li>• 고객 정보(전화번호, 주소)</li><li>• 관리자 정보</li><li>• 상품정보(선택사항)</li><li>• 주문번호에 따른 상품 정보 및 주소, 전화번호 등의 개인정보</li><li>• 실시간 설비 정보</li><li>• 데이터베이스에 주문번호에 따른 상품 목록을 저장할 때, 어떠한 형태로 저장해야할지</li></ul>	<p>주문 번호와 연계된 개인정보, 상품 정보 등으로 민감한 정보가 분류되어 있다.</p> <p>두 데이터 베이스를 같이 사용할 경우 설계가 복잡해진다.(로그 등은 몽고, 개인정보는 마리아)</p>
성능 최적화 방법 과 학습 곡선	<ul style="list-style-type: none"><li>• 몽고디비 : 샤딩, 복제 등을 사용하며 스키마리스 데이터베이스 사용과 JSON형식의 데이터에 익숙해 질 수 있음</li><li>• 마리아디비 : 인덱싱, 파티셔닝등을 사용하며 SQL문과 ORM에 대한 이해를 높일 수 있음, 데이터베이스 모델링 상세한 설계 필요</li><li>• 공통 : 인덱싱을 통한 조회성능 향상 , 캐싱(부적합할수도) 등</li></ul>	<p>스케일 업, 스케일 아웃을 해보는 것 까지 고려해야 하는지( <a href="#">🔗 Scale-up과 Scale-out에 대해 알아보자!</a> )</p>
장점만 추출해서 사용할 방법	<ul style="list-style-type: none"><li>• 보안적으로 덜 중요하고 임시 db적인 역할만 하면 되는 주문번호에 따른 상품정보나 로그의 경우 몽고에 저장(iot와 연계에 필요한 데이터만 몽고에 저장 고려 )</li><li>• 전반적인 데이터는 RDBMS에 저장</li></ul>	<p>IoT와 연결된 부분 중 상품 출하 부분 및 실시간 기기 상황에 대한 것은 빨라지지만</p> <p>결론적으로 자동 분류 및 조회 부분에서 RDBMS를 사용해야 하는 부분은 동일하다.</p>
최적화 방법	<ul style="list-style-type: none"><li>• 파티셔닝 : 상품주문정보에 있는 날짜 년/월/일을 기준으로 한 테이블에서 파티셔닝을 통해 데이터를 분산하여 저장할 수 있다.</li><li>• 인덱싱 : 인덱스를 추가하여 쿼리 실행속도 향상</li><li>• 조인, 서브쿼리 최적화</li></ul>	<ul style="list-style-type: none"><li>• <a href="#">🔴 RDBMS 성능 최적화 전략</a></li><li>• <a href="#">🔍 SQL 쿼리 최적화는 어떻게 해야할까?</a> · <a href="#">Issue #1 · jiyeon12345/notion_backup</a></li></ul>

	<ul style="list-style-type: none"> <li>• 페이지네이션</li> </ul>	
임시결론	<ul style="list-style-type: none"> <li>• RDBMS(MariaDB)의 경우 장기간 저장해야 할 전반적인 데이터를 저장하되 주문에 대한 정보의 경우 주문번호를 기준으로 연도별, 월별로 분리해서 저장</li> <li>• 사용자 및 관리자 정보 등 민감한 개인정보 등이 있는 경우 최대한 보안성을 유지한 채로 저장(이 경우에만 조인 혹은 select 사용 시 쿼리문 다각화)</li> <li>• 사용자 정보를 기반으로 저장할 경우 검색할 수 있는 기간을 1년 정도로 한정해서 조회할 테이블의 수를 최소화</li> <li>• NoSQL(MongoDB)의 경우 기기에 대한 로그를 저장하며 실시간 상태를 백업하여, 프론트와 실시간 통신에 필요한 자료를 저장</li> <li>• 주문 후 배송 완료까지 남은 상품에 대한 정보 및 배송 과정에 있는 상품 정보를 조회하여 배송중인 상품에 대한 빠른 조회 및 상태 업데이트를 지원.</li> </ul>	

#### 참고 사이트

-  [MySQL vs MongoDB 성능 분석](#)
-  [MongoDB vs. MySQL](#)
-  [DB에서 개인정보 처리 시 준수해야 하는 보호조치](#)
-  [\[DB\] DB 파티셔닝\(Partitioning\)이란 - Heee's Development Blog](#)
-  [00. MongoDB\(몽고디비\) Study - 목차](#)