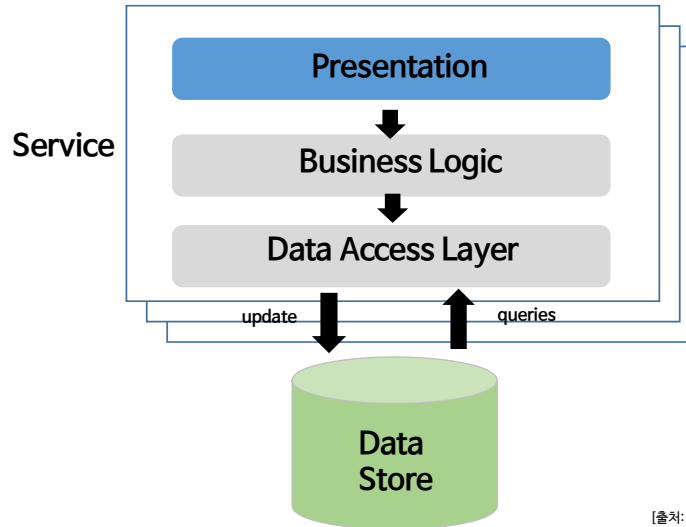


명령과 조회의 역할 분리 -CQRS 패턴

SK(주) C&C 한정현

CQRS (Command Query Responsibility Segregation : 명령 조회 책임 분리)

- 전통적 DB 트랜잭션 → 리소스 교환상대 발생 가능

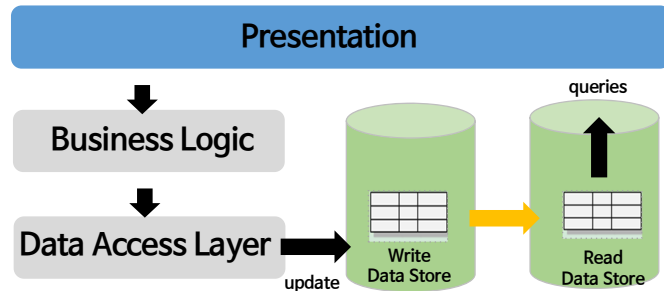
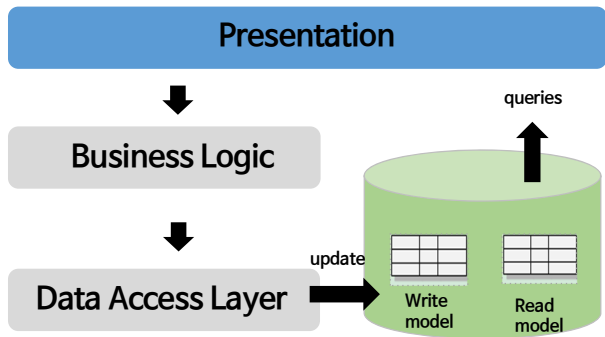


레이어를 넣고, 입력, 수정, 삭제, 조회하는 방식

CQRS (Command Query Responsibility Segregation : 명령 조회 책임 분리)

입력, 수정, 삭제 보다 조회가 훨씬 많이 사용됨

- 트랜잭션에 필요한 데이터 모델과 쿼리에 사용하는 데이터 모델을 분리
- 좀 더 나아가서 아예 트랜잭션용 저장소와 쿼리용 저장소를 따로 준비

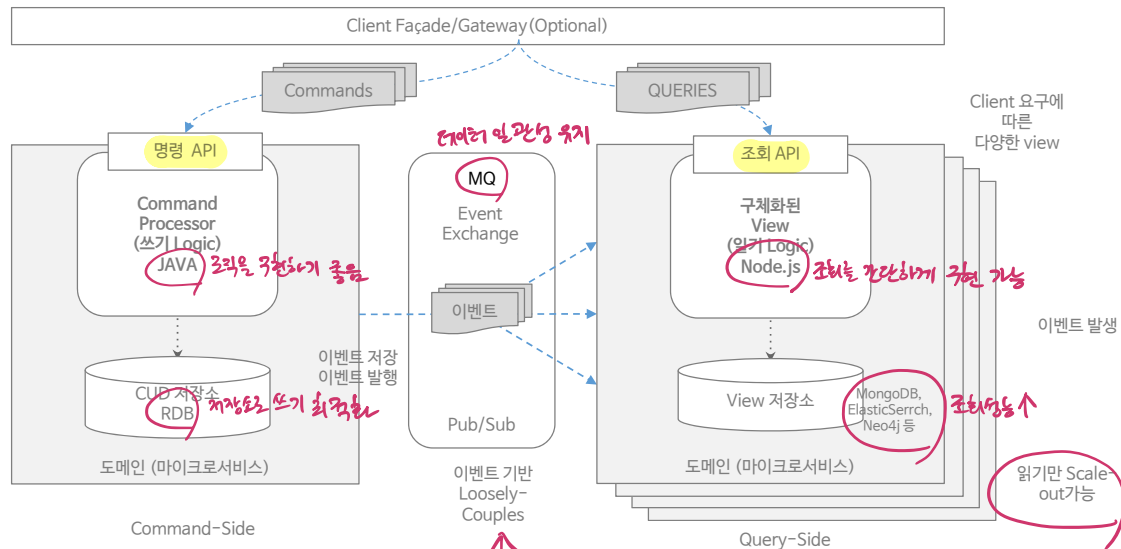


쓰기와 조회의 전략을 각각 분리하면

쓰기시스템의 부하를 줄이고 대기시간을 줄이는 등 엄청난 이점을 가져옴

[출처: <https://msdn.microsoft.com/en-us/library/dn568103.aspx>]

CQRS (Command & Query Responsibility Segregation)



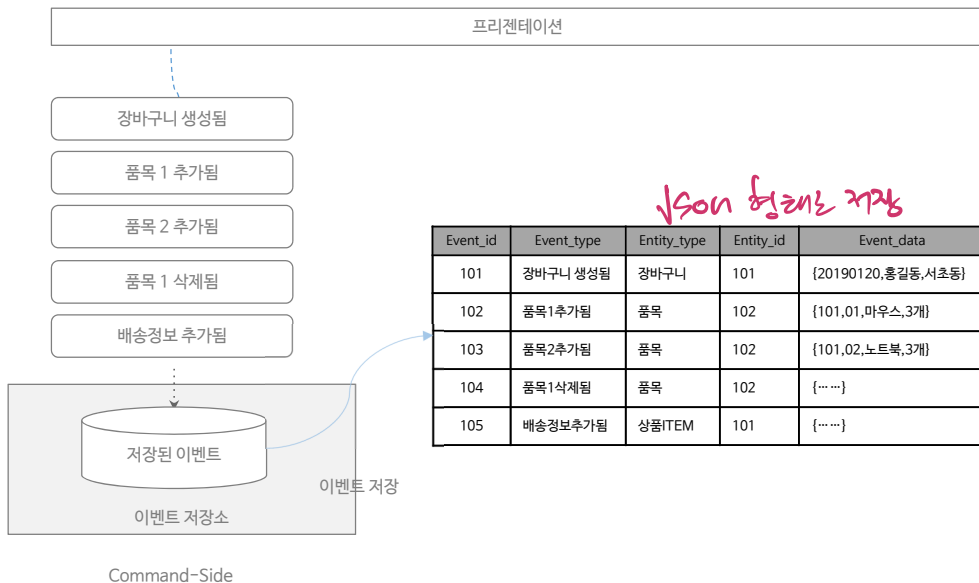
Eventual Consistency

사용량이 많기 때문

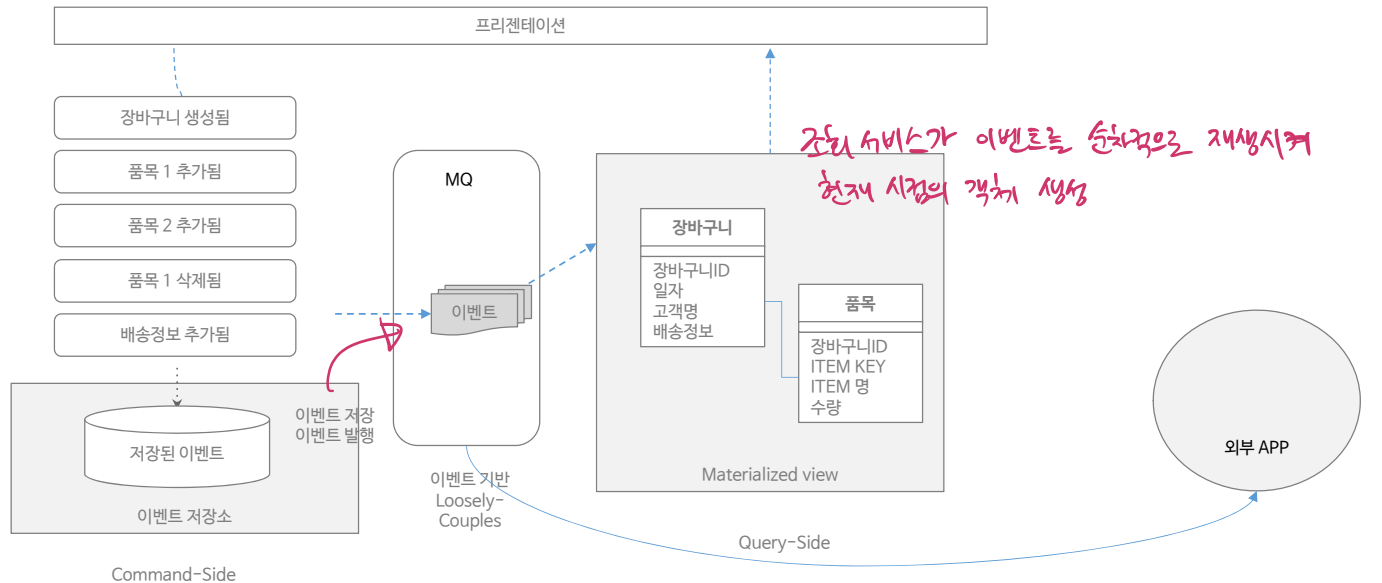
CQRS 패턴과 자주 같이 사용

Event sourcing

관계형 모델로 변환하지 않고 이벤트 자체를 순차적으로 포함하여 저장



Event sourcing



[출처: <https://msdn.microsoft.com/en-us/library/dn568103.aspx>]

이벤트 스트림 : 모든 쓰기를 이벤트로 전환해서 별도의 이벤트 스트림으로 스트림 저장소에 저장하는 방식

이벤트 스트림 저장소 : 추가만 가능하며 계속 이벤트들이 쌓이게 만들고,
필요한 레이어를 구현해서키는 시점에서 그때까지 축적된 레이어를 바탕으로 작성