

과제4 피드백 정리

손해배상청구서 롤플레이

상황 : '나'는 '갑'의 입장, 멘토님은 '을'(파트너사)의 입장

롤플레이 : 피드백

대부분 파트너사 : 시간 끌려고 함 + 영업부서에게 맡기려고 함 + 책임 회피하려 함

내가 '갑'인 경우

1. 처음부터 손해배상 청구서를 들이밀지 말기 – 반감 및 책임 회피 가능성 높음 : 날씨 얘기등 전혀 상관 없어 보이는 얘기도 하면서 손해배상건 관련 발뺌할 수 있는 부분을 먼저 물어서 명확하게 한다. 그 후, 손해배상 얘기를 꺼내면서 자연스럽게 말 꺼내기
2. 영업 부서에 맡기려고 하는 경우, 어영부영 시간만 끌 수도 있으므로, 기한을 명확히 하고, 나도 어쩔 수 없다는 제스처를 취한다.
3. 시간 끌려고 하는 부분이 보인다면, 많은 응답을 하지 말고, 시간이 없음을 어필하며 기한을 명확히 하고 빨리 빠져나오기

내가 '을'인 경우 : 파트너사의 행동대로 하면 됨

궁금한 점 : 피드백

1. mysql-error.log, mysql-error-running.log의 차이

Mysql-error.log : 정말 치명적인 문제 log – 거의 고장났다고 보면 됨

Mysql-error-running.log : 덜 치명적 – 그냥 서버의 에러

2. Aws의 backtrack으로 적절한 복구시점을 찾았다고 하더라도 그 시점 이후의 데이터 insert가 있었다라면 그러한 데이터에 대해서는 어떻게 복구를 하는지

1. 콜드 백업 + (mysql)bin log(백업 이후)

2. 핫 백업 : 사고 내용만 지운 bin log로 백업

궁금한 점 : 피드백

- 백업 관련 설명

- 1) Oracle : 안정성 짱짱맨 – log를 정말 세세하게 기록하고, 보여주며, 복구 당시에도 연결된 관같은게 있어서 안정성 있게 복구 가능함(복제 개념이 아니라 두개의 기계가 sync하게 동작하는 것)
- 2) Mysql : 많은 log를 지원하지 않음. 그나마 지원한 것이 general_log이지만, 이마저도 시스템에 큰 영향을 주어 보통의 경우는 꺼놓음. 또한, oracle처럼 sync한 두 시스템이 아니라, 하나의 master에 대해 여러 slave들이 복제된 형태 – not sync 하지만, 어느정도 연결하는게 bin-log. 본래 복제시스템 관리에 대한 log였으나, 복구에도 많이 쓰임



오라클



mysql

궁금한 점 : 피드백

3. show errors;에서 나타나는 오류 메시지는 어느 로그에서 확인 가능한지

4. AWS RDS에서 mysql-error.log를 보았을 때, 쿼리문 에러는 없었는데 mysql-error.log는 어떠한 에러를 저장하는 것인지

쿼리문 에러와는 조금 다름. 서버 구동시에 에러를 담는 것

5. DB 책임소재를 어떻게 명확하게 하는지 ; 파트너사가 한 것인지 우리쪽에서 한 것인지 책임소재 명확히 하는 방법

보통은 처음 계약할 때 부터 명확하게 누구의 소관인지 정함. 그게 모호할 때 파트너사와 싸우는 것.