26일차

IoT 디바이스들을 통해 데이터 생성! + 해당 데이터를 허브로 송신

IoT 허브를 통하여 주기적으로 데이터를 수집

Stream Analytics job을 통해서 해당데이터들을 저장!

허브에서 모은 데이터를 Stream을 통해서 데이터의 흐름을 제어한다

해당 데이터들을 여러 곳으로 나눠서 보내거나, 혹은 일부분을 정제해서 보낸다거나 등

수집한 데이터를 저장해야하는데, 블롭 스토리지는 매우 싸니까

블롭 스토리지에는 전체 데이터를 때려 넣는다!

애저 SQL Database 에는 데이터 평균치와 같은 통계 데이터만 넣는다!

데이터?

1. 구조화된 데이터
   1. 데이터 베이스!
2. 반 구조화 데이터
   1. json, csv, yaml 등등
   2. 스스로 자신의 구조를 설명하는 데이터를 반 구조화 데이터
3. 비구조화 데이터
   1. 음성, 사진 등 구조가 없는 데이터

OLTP vs OLAP

1. OLTP – 온라인 거래 처리
   1. 주로 은행에 비유 – A 은행에서 B 은행으로 송금할 때, A 은행에서 출금하여 B 은행으로 입금할 것.
      1. 근데 A 은행에서 출금하자마자 B 은행이 정전되면??? 내 돈은???
      2. 이런 문제를 해결하기 위해, 두 동작(출금과 송금)을 묶어서 처리! -> 트랜잭션
   2. 혹은 여행사에 여행 상품을 사면 – 항공권, 호텔, 렌터카 등이 필요!
      1. 만약 항공권은 구매가 됐는데 나머지가 예약이 안됐다면? -> 그럼 상품 못 팔지
      2. 하나라도 실패하면 모두 취소하도록 하는 것
   3. 따라서 트랜잭션 방식은 도중에 해당 데이터가 바뀌지 않도록 하기위해 잠금 처리함 -> 많은 시스템 자원을 필요로 함
   4. 여러 트랜잭션이 동시에 이루어질 경우, 데드락 발생 가능성이 있다.
2. OLAP – 온라인 분석 처리
   1. 대기업 S 그룹의 전체 연간 매출 데이터를 집계해야한다면?
      1. 모오든 데이터를 다 집계해서 통계를 내려면 시간이 많이 걸릴 것
      2. 거기에 업무 시간에 다 할 수도 없을 것
   2. 따라서 주기적으로 ‘큐브’ 라는 곳에 분석할 데이터들을 미리 준비해 둔다
      1. 언젠가는 필요할 데이터들을 미리미리 준비해두는 것

트랜잭션 워크로드

1. 원자성
   1. 여러가지 작업이 하나로 묶여서 하나라도 실패하면 다 롤백!
2. 일관성
   1. 누군가가 이미 데이터를 수정 중이면, 해당 데이터는 잠금상태가 된다!
3. 격리
   1. 데이터베이스는 여러 사람들이 동시에 사용하지만, 데이터 간에 충돌하지 않게 구분되어 동일 상태로 유지된다
4. 영속성
   1. 예를 들어 데이터를 메모리로 가져와서 작업했다면, 작업이 다 끝났을 때는 그 데이터가 다시 데이터베이스에 적용되어야 한다

데이터 처리

1. 일괄 처리: batch processing
   1. 예를 들면 쇼핑몰에서 물건을 구매했는데, 우선 주문 접수만 하고 저녁이나 밤에 몰아서 주문 확정 처리하는 방식
   2. 예전에는 동시에 많은 데이터를 빠르게 처리하기 어려워서 이런 방식으로 처리했음
2. 스트림 처리: stream processing
   1. 각 데이터를 실시간으로 처리!

DB 관리 도구?

1. Azure Data Studio
2. SQL Server Management Studio
   1. 윈도우에서만 실행 가능하나, 기능도 가장 많고, 디테일하게 관리 가능
   2. 보통 SSMS라고 줄여부름
3. Azure Portal/CLI