**02 3 데이터베이스관리시스템 발전과정**

2023.09.23 토 오후 4:29 ・ 18분 37초  
임소정

참석자 1 00:01  
dbms의 발전 과정에 대해서 살펴보도록 할게요.   
dbms가 나오기 전에 파일 시스템에서는 각 응용 프로그램마다 별도의 데이터 파일을 관리를 해서 데이터에 대한 중복성의 문제, 무결성의 문제 그리고 이제 종속성의 문제가 있었던 거죠.

참석자 1 00:24  
그래서 이제 db ms가 나왔는데 다양한 dbms가 나와 있고 또 계속해서 연구 연구를 해서 발전된 dbms가 있고 또 현재 유행하고 있는 dbms가 있어요.   
이러한 것에 대해서 이제 살펴보겠습니다.   
먼저 계층형이 있는데요. 계층형은 이제 ibm에서 처음으로 사용을 했고요.

참석자 1 00:51  
데이터를 이제 계층 구조다 계층 구조를 이제 트리 구조다 부모 자식 간이다.   
이렇게 이제 다양하게 이야기를 하고 있어요.   
그래서 이제 이게 뻗어나가는 방식인데 이걸 이제 뒤집어서 보면 나무 모양이기 때문에 트리 구조다라고 하는 것이고

참석자 1 01:12  
나의 부모 밑에 자식이 있다. 이렇게 해서 부모 자식 간의 관계다라고 해요.   
부모 자식 간의 관계이기 때문에 우리가 이제 1 대 1 또는 1 대 n의 관계를 표현을 할 수가 있는 거죠.   
하나의 부모는 여러 자식을 가질 수가 있는 거죠.   
하지만 하나의 자식이

참석자 1 01:37  
여러 부모를 가질 수는 없죠. 그래서 이거를 이제 유연성이 부족하군.   
이렇게 이제 이야기를 하고 있습니다.   
그래서 이거는 이제 하나씩 뻗어나가는 형태 얘는 이렇게 이제 1 대 1의 관계 또는 1대 다의 관계를 이제 표현할 수가 있는 거죠.

참석자 1 01:59  
그래서 우리 눈에 보이기에는 이제 이렇게 계층 구조로 표현이 되어 있지만 이제 실질적으로는 각 데이터를 또는 각 객체를 포인터를 통해서 찾아가는 그런 방식이 되겠습니다.   
그러면 이제 여기서 1대 n의 관계니까 다대다의 관계가 지금 안 되는 거죠.   
그래서 연구자분들이 또 열심히 연구를 하겠죠.

참석자 1 02:27  
그래서 다의 관계가 될 수 있는 것을 이제 연구를 해놓은 거죠.   
그걸 우리는 이제 네트워크형이다라고 이야기를 해요.   
그래서 이제 코다실 모델을 기반으로 발전을 한 부분이고 얘는 이제 그래프 형태예요.   
사이클 형태다라고 이야기를 하는 거죠.

참석자 1 02:50  
그래서 앞에서 나온 네트워크는 이렇게 뻗어나가는 형태였는데 여기에 선이 하나 더 붙어요.   
그러면 이렇게 사이클이 형성이 되었다 그래프가 형성이 되었다라고 하는 거죠.   
그래서 그래프 형태로 표현되는 것을 우리는 이제 네트워크형이다라고 이야기를 해요.   
그래서 다대다 관계가 되는 것이고 다대다 관계다 보니까 좀 더 복잡성이 증가되었다

참석자 1 03:22  
라고 보는 거죠. 이런 식으로 사이클이 형성되는 것을 볼 수가 있는 거죠.   
이렇게요.

참석자 1 03:37  
그리고 이제 관계형 dbms가 있어요.   
이거는 이제 우리가 현재 가장 많이 사용되고 있는 부분이에요.   
그래서 우리가 이제 한 한 단어 많은 시간을 투자를 해가지고 우리가 별도로 공부를 할 거예요.   
데이터를 이제 테이블 형태 행과 열의 형식으로 저장을 한 거예요.

참석자 1 04:03  
그리고 이 sql을 사용해 가지고 데이터를 조회하고 추가하고 수정하고 삭제하는 명령어를 사용하는 거고 현재 가장 많이 사용되는 dbms에 대한 유형이 돼요.   
그래서 이 관계형 dbms 방식으로 표현된

참석자 1 04:25  
것들이 이제 sql이 뭐가 있냐면 오라클도 있고 마이 sql도 있고 마이크로 소프트 pl 서버도 있고 또 포스트 그래스 등이 있다는 거예요.   
그래서 행과 열로 표현된 거 이렇게 이제 반듯한 테이블 형태인 것을 이제 우리가 볼 수가 있죠.   
그래서 우리가 이제 직원에 대한 정보,

참석자 1 04:52  
부서에 대한 정보, 급여에 대한 정보 이러한 것들을 이제 테이블 형태로 저장을 하고 또 이런 것들의 관계를 표현을 해가지고 인사 관리 시스템이다.   
이런 식으로 이제 만들기도 하죠. 네 이런 방식이 이제 관계형입니다.   
관계형 하면 우리가 이제 테이블 형태로

참석자 1 05:18  
저장을 한다라고 기억해 주면 될 것 같고요.   
객체 지향 dbms에 대해서 보겠습니다.   
이거는 객체 지향 프로그램이 인기가 있으면서 이제 같이 이제 개발된 것이에요.   
그래서 데이터를 객체로 표현을 했다.   
상소 캡슐화, 다양성과 같은 객체 지향에 대한 개념을 지원을 했다라는 부분이에요.

참석자 1 05:46  
그래서 객체다라는 개념이 이제 조금 이제 어려운 부분인데 객체 지향 프로그램에서 이제 객체다라는 개념은 우리가 이제 레고 블록이다라고 생각해 주면 될 것 같아요.   
그래서 이제 프로그램 쪽에서는 이제 예를 들어서 여러분이 이제 레고 블록이다.   
그러면 이제 하나는 이제 이렇게

참석자 1 06:11  
아주 조그마한 거지만 그걸 이어붙이면 비행기도 만들어지고 자동차도 만들어지고 그런 개념이 되죠.   
그래서 내가 이제 엑셀이라는 프로그램을 개발을 하게 되는데 여기에 이제 글꼴에 관련된 거 글자세 글자 크기, 글꼴 이런 이제 명령어들이 이렇게 하나로 만들어져 있는 거죠.

참석자 1 06:36  
이게 이제 만들어진 레고 세트다라고 보면 되세요.   
그다음에 이제 단락에 관련된 것도 왼쪽 맞춤 오른쪽 맞춤 이런 명령어들을 이렇게 세트로 만들어 놓은 거죠.   
잘 오픈이 돼서 아무런 문제가 없었어요.

참석자 1 06:53  
그러다가 이제 파워포인트 프로그램을 개발하는데 글골이 필요하군 글골에 관련된 건 필요하군.   
그럼 이걸 통째로 가져와서 쓸 수 있는 거죠 문단이 필요하군.   
이걸 통째로 가져와서 쓸 수 있는 곳 이렇게 이제 끼워넣기를 할 수 있는 그런 개념이라고 봐주면 되고요.   
여기서 상속이다. 그러면 부모님에 있는 것을 내가 물려받는 거예요.   
재사용의 개념이 되는 거죠. 그럼 여러분이 그냥

참석자 1 07:28  
선언해서 쓰면 되지 왜 재사용하나요? 그럴 수가 있는데 아주 복잡한 것들에 대해서 타입 설정이라든지 글자 크기 설정, 제한 조건 설정 이런 것들을 이제 다시 설정하다 보면은 여러 가지 이제 또 복잡하니까 이미 선언되어 있으면 그걸 그냥 갖다 쓰는 거죠.   
완전히 갖다 쓸 수도 있고 그걸 조금 고쳐서 쓸 수도 있는 부분이고요.

참석자 1 07:57  
캡슐하다. 그러면 이제 우리가 또 항상 정보화 시대에 항상 뭘 조심해야 되느냐 해커를 좀 조심해야 되겠죠 그래서 이렇게 딱 놔두고 여기에 이제 접근할 수 없도록 해주고 꼭 필요한 것들만 이제 바깥에 보여주는 거죠.   
그래서 해커가 침입을 했을 때 이 안에를 못 들여다볼 수 있도록 이제 막아버리는 거 그게 이제 캡슐라라고 보면 되세요.

참석자 1 08:26  
그래서 알약을 우리가 이제 아팠을 때 알약을 처방받은 경우도 있지만 제가 안약은 좀 삼키기가 힘들어요.   
어린아이 같은 경우. 그러면 이제 이렇게 조금 가루로 만들어서 주기도 하죠.   
그다음에 이제 여러 개의

참석자 1 08:46  
약을 이제 갈아가지고 그냥 하나의 캡슐에 넣어주세요.   
그렇게 하기도 하죠. 그래서 이렇게 이제 캡슐에다가 넣어서 이렇게 주기도 하죠.   
그러면 안락 하나하나가 있었을 때는 이걸 보면서 이건 두통약이고요 이건 소화제고요 이건 항생제예요.   
지칭할 수가 있지만 이렇게 캡슐에 딱 넣어놓으면은 이게

참석자 1 09:12  
무슨 약인지 뚜껑을 열어보지 않으면 캡슐을 열어보지 않으면 그걸 모르는 거잖아요.   
그런 개념이라고 보면 되세요. 이제 다양성이다가 마을 다 다양성이라는 것은 뭐냐면 객체들이 서로 다른 동작을 수행할 수 있는 것을 인정하는 거예요.   
무슨 말이냐 제가 여러분한테 지금부터 여러분 오픈해 주세요라고 한 거예요.

참석자 1 09:41  
뭐를 오픈하죠? 많아요. 책을 오픈한다 창문을 오픈한다 필통을 오픈한다 가방을 오픈한다.   
많죠 그래서 이렇게 다양한 형태로 나타날 수 있는 것을 지원한다라고 보면 되겠습니다.   
그리고 이런 이제 객체 지향 dbms는

참석자 1 10:06  
이제 복잡한 데이터 구조와 관계를 처리하는 데 적합하고 또 특정한 분야에 적합화되어 있다라고 이제 이야기를 해요.   
그래서 특정한 분야다. 그러면 이제 여러분이 우리 이제 인간들이 이제 보통 이제 일반인들이 많이 찾는 것도 있지만 위치 시장이다라고 해가지고

참석자 1 10:34  
아주 특정한 고객 그룹을 이제 관리하기 위한 특화된 제품 서비스를 말을 해요.   
그래서 예를 들어서 이제 특별한 애완동물에 대한 내용들 우리가 이제 보통 애완동물 하게 되면 개나 강아지나 또는 고양이 그것에 대한 물품은 많이 있죠 그런데 아주 특이하게

참석자 1 11:00  
이제 개구리를 키운다든지 햄스터를 키운다든지 쥐를 키운다든지 또 미니돼지를 키운다든지 악어도 키운 사람이 있고 도마뱀도 키운 사람도 있고 특이하게 바퀴벌레를 키운 사람도 있죠.   
그런 것들에 대한 물품도 필요하죠. 그리고 또 우리가 식단도 일반 식단도 있지만

참석자 1 11:24  
당뇨병 환자들을 위한 식단도 있죠. 이처럼 아주 특정한 고객 그룹의 관리에 특화된 제품 서비스가 이제 니치 시장이라고 있대요.   
이처럼 특정한 분야에 이제 적합한 dbms다라고 보면 되겠고요.   
그다음에 이제 노 sql이 있는데요.

참석자 1 11:47  
이거는 이제 뭐냐면 비관계형 데이터베이스예요.   
그래서 앞에서 관계형 데이터베이스는 뭐였어요? 반듯한 이렇게 이제 테이블 형태로 행을 중심으로 홍길동에 대한 정보 김길동에 대한 정보 이렇게 이제 행을 중심으로 된 데이터를 가지고 이제 연구를 했던 그런 부분이 됩니다.

참석자 1 12:14  
그런데 이제 여기 같은 경우는 노 sql은 구조적인 제한 없이 대용량의 분산 데이터 저장을 지원을 하는 거예요.   
그래서 이것을 한번 분류를 해보면 키 값 스토와 키와 값의 쌍으로 데이터를 저장하는 거예요.

참석자 1 12:36  
그래서 이제 파이썬의 딕셔너리 구조처럼 그렇게 되어 있는 거고 또 문서 기반 데이터베이스 제이슨 등의 문서 형식으로 데이터를 저장하는 거 그다음에 컬럼 지향 데이터베이스가 있어요.   
관계형은 행 기반이었는데 이제 얘는 열 기반이 되는 거죠.

참석자 1 12:59  
또 그래프 데이터베이스 그렇죠 데이터를 노드 엣지 이런 형태로 저장을 해가지고 연결에 중점을 둔 부분이에요.   
그래서 이런 걸 기본으로 해서 이제 만들어진 종류들이 이제 이러한 것들이 있다라는 거예요.

참석자 1 13:19  
그래서 여러분이 내가 이제 정형 데이터를 가지고 데이터를 정확하게 처리를 하겠다.   
그러면 이제 관계형 dbms를 보는 거고 비정형 데이터를 내가 이제 처리를 하겠다.   
그러면 이제 no sql을 하는 거죠.   
그런데 이제 기업체 입장에서는 이것도 구입하고 이것도 구입하니까 또 비용적인 문제가 생기겠죠

참석자 1 13:49  
그래서 좀 불편하군 해서 또 연구자분들이 정형과 비정형 데이터를 그냥 다 안정적으로 빠르게 처리하는 방법은 없을까라고 해서 이제 또 연구에서 나온 것이 있어요.   
그게 이제 뭐냐면 우리가

참석자 1 14:13  
네 뉴sql이다라고 그래요. 그래서 관계형 dbms 또 sql 표준 쿼리 기능을 유지하면서 대규모 분산 환경에서 확장성을 제공하려는 시도예요.   
그래서 정형 데이터도 또 비정형 데이터도 이제 처리할 수 있는 부분입니다.

참석자 1 14:39  
그래서 구글 스페너가 있는데 클라우드 스페너라고도 하는 부분이고 구글 클라우드에서 제공하는 글로벌 분산 관계형 데이터베이스 서비스가 돼요.   
그래서

참석자 1 14:56  
아주 대규모 데이터를 다루면서 관계형 데이터베이스 또 노 sql의 장점을 결합한 것이 되고요.   
그다음에 이제 콩 로치라라고 있는데요.   
콩 로치가 바퀴벌레라는 뜻이에요. 그래서 이제 우리는 바퀴벌레가 정말 싫죠 그런데 인간이 있는 곳에는 항상 바퀴벌레가 있는 것 같아요.

참석자 1 15:24  
그런데 또 바퀴벌레에서 우리가 또 배울 점은 있는 것 같아요.   
그건 뭐냐 하면 끈질긴 생존력 그다음에 이제 잘 죽지 않죠 그래서 죽어가면서도 또 알을 깐다는 그런 게 있더라고요.   
그래서 데이터베이스에서도 지속적으로 긴 생존력처럼

참석자 1 15:51  
지속적으로 서비스를 제공할 수 있다.   
그런 의미고 또 데이터 복제를 통해가지고 장애 없이 서비스를 이제 제공할 수 있어요라는 의미에서 이름을 이렇게 이제 콩 노치다라고 지었다라고 그래요.   
그래서 이제 여기까지 보면 dbms의 발전을 보면

참석자 1 16:16  
파일 시스템에서 계층형과 네트워크형 그다음에 관계형 객체의 지향 노 에스ql 뉴 에스큐엘 이런 식으로 이제 발전이 된 거죠.   
그래서 이 뉴 sql은 아직 시장에 진입한 지 얼마 되지 않았지만 계속해서 관계형

참석자 1 16:37  
또 노 에스ql n에스ql 이러한 것들이 이제 상호 보완하면서 공존할 것이다라고 이제 우리는 보는 거죠.   
그러면 이제 여기까지 여러분이 파일 시스템에 대한 문제점 또 데이터베이스 관리 시스템에 대한 장점 에 대해서 설명할 수 있으면 좋겠습니다.

참석자 1 17:03  
문제 한번 풀어볼까요? 다음에서 설명하는 파일 시스템의 문제점은 무엇인가요라고 했어요.   
네 파일의 레코드 크기나 타입이 변경이 되면 해당 파일에 액세스하는 프로그램 코드도 수정이 돼야 된다라는 거예요.

참석자 1 17:21  
a가 변화가 되면 b도 변화가 돼야 된다.   
이게 이제 연결된 종속성이다라고 보는 거죠.   
네 a가 어떤 행동을 하든지 간에 b가 영향을 미치지 않는다.   
그건 우리는 독립성이라고 하죠. 그래서 이것에 대한 정답은 뭐겠습니까? 종속성이다라고 보는 거죠.   
다음 문제 한번 살펴볼게요.

참석자 1 17:49  
다음 주 데이터베이스 관리 시스템의 장점으로 보기 어려운 것은? 그랬어요.   
보안이 향상이 되었죠 중복에 대한 최소화 문제 적은 비용인가요? 아니죠.   
데이터베이스 관리 시스템을 구축하려면 굉장히 비용이 많이 드는 그런 부분입니다.   
투자비가 되는 거죠. 우리가 언제

참석자 1 18:14  
우리가 이제 어떤 사업이든지 간에 초기 투자 비용이 들어가듯이 데이터베이스 구축을 하는데 비용은 이제 들어가게 된다 라는 부분이에요.   
그래서 정답은 3번이 되겠습니다. 여기까지 우리가 데이터베이스 관리 시스템에 대한 내용이었어요.

clovanote.naver.com