**03 데이터베이스시스템**

2023.09.23 토 오후 4:31 ・ 31분 16초  
임소정

참석자 1 00:02  
안녕하세요 여러분 데이터베이스 시스템에 대해서 살펴보도록 하겠습니다.   
문혜영입니다.

참석자 1 00:14  
우리가 이번에 살펴볼 내용은 데이터베이스 시스템에 대한 구성 스키마 데이터베이스 사용자 그리고 데이터 언어에 대해서 살펴보도록 할게요.   
먼저 데이터베이스 시스템에 대한 구성이 되는데요.   
이제 일반 사용자가

참석자 1 00:38  
뭔가 이제 데이터베이스에 접속해서 뭔가 얻고자 하는 게 있겠죠 그러면 데이터베이스 언어를 통해서 dbms를 통해서 데이터베이스에 접속을 해서 그 결과를 이제 갖게 되는 그런 구조 전반적인 것을 우리는 데이터베이스

참석자 1 00:58  
시스템이다라고 하는 부분이에요. 그래서 이 데이터베이스 시스템은 약자로 dbs다라고 하는 부분입니다.   
데이터베이스의 데이터를 저장하고 관리하고 조직에 필요한 정보를

참석자 1 01:16  
생성하는 데 필요한 컴퓨터 중심의 시스템이다라고 정의를 내릴 수가 있고요.   
우리가 이제 데이터베이스다 그러면 데이터와 데이터 간의 관계의 집합인 거고 dbms 가 이제 db ms죠 dbms는 사용자가 데이터베이스에 접근할 수 있도록

참석자 1 01:41  
지원해주는 프로그램의 집합이었어요. 그러면 이 나머지 부분 스키마 데이터베이스의 사용자 데이터베이스 언어 이러한 것에 대해서 우리가 이제 살펴보도록 할게요.   
그래서 먼저 스키마부터 보도록 하겠습니다.

참석자 1 02:02  
스키마를 보겠습니다. 스키마는 이제 우리가 설계도다라고 생각해 주면 돼요.   
그래서 스키머를 만든다라는 것은 데이터베이스 설계를 한다 데이터 모델링을 한다 데이터베이스 디자인을 한다 이렇게 생각해 주면 될 것 같고요.   
스키마는 이제 데이터베이스에 대한 구조

참석자 1 02:27  
네 그래서 그다음에 제약 조건 이런 것들을 이제 기술을 하게 돼요.   
제약 조건은 예를 들어서 이름은 내가 문자 10자리로 해야지라는 것들 그다음에 주민번호는 반드시 13자리가 들어와야 돼.

참석자 1 02:48  
그죠? 그 크기를 맞추지 않으면 들어올 수 없음 이런 것들 특정한 필드는 널이 되면 안 돼.   
얘는 기본 키아 얘는 외래 키아 이런 것들에 대한 제약 조건을 정의를 합니다.   
현재 직원 테이블은 이름 주민번호 감독자 주민번호 부서 번호로 이루어져 있고

참석자 1 03:12  
부서 테이블은 부서명 부서 번호 관리자 주민번호로 이루어졌죠 직원 테이블의 부서 번호는 부서 테이블의 부서 번호를 이렇게 참조하고 있어요.   
이렇게 참조하는 거 이거를 이제 우리는 뭐라 그러죠? 외래키라고 하는 거죠.   
그리고 얘는 이제 밑줄이 쳐진 걸로 봐서 얘는 기본기가 되는군요.   
이러한 것의 용어들에 대해서는 이제 계속해서

참석자 1 03:40  
우리가 자세히 다루게 되겠고 여기 있는 감독자 주민번호는 본인 테이블에 있는 주민번호 를 참조를 하고 있어요.   
그래서 자기 테이블에 있는 것도 이렇게 참조할 수가 있어요라는 겁니다.   
그리고 이제 인스턴스는 무엇이냐 이 속에 들어가 있는 이제 들어가는 이제 내용물이다

참석자 1 04:04  
라고 이제 살펴보면 되겠습니다. 그래서 네 우리가 이제 특정 시점의 데이터베이스에 실제로 저장되어 있는 데이터다라는 거예요.   
그래서 보통 스키마 그러면 우리가 한 번 정해지면 거의 변경하지 않는 것이고요.

참석자 1 04:27  
여기 있는 이제 인스턴스는 잦은 변동이 일어나게 되는 거죠.   
실제로 저장되는 값이니까 삽입 삭제 수정 이런 작업이 많이 일어나겠죠 그래서 여기까지는 우리가 이제 스키마다라고 하고 여기 하단은 인스턴스라고 해요.   
그래서 이제 여러분이 이제 우리가 집의 구조를 바꾸기는 그렇게 쉽지 않죠

참석자 1 04:55  
그래서 집의 구조다. 구조는 자주 바뀌지 않는데 구조는 이제 스키마고 이 속에 이제 들어와서 거주하는 사람들의 명단은 이제 자주 바뀌게 되는 거죠.   
그래서 여기 거주하는 사람들이다. 사람들 이 속에 사는 사람들은 계속해서 바뀌게 되는 거죠.   
네 그런 차이다라고 보면 되겠습니다.

참석자 1 05:21  
네 3단계의 데이터베이스 구조가 되는데요.   
내부 스키마 개념 스키마, 외부 스키마 이렇게 이루어져 있어요.   
우리가 데이터베이스를 사용을 할 때 이 데이터베이스의 그 복잡한 구조를 또 동작 방식을 모두 이해하고 있어야 한다

참석자 1 05:43  
라고 한다면 일반 사용자들은 매우 이제 힘들겠죠 그래서 일반 사용자가 데이터베이스를 쉽게 이해하고 이용할 수 있도록 이렇게 3단계 데이터베이스 구조를 제안을 했어요.   
그래서 모든 데이터에 저장에 대한 유지 그다음에 관련된 복잡한 내용을 숨기고

참석자 1 06:09  
이게 필요한 데이터만 이제 단순화한 외부 단계의 관점을 일반 사용자들에게 제공을 하게 된다.   
그래서 우리 이제 조직 전체에 대한 정보가 있지만 우리가 이제 예를 들어서 인사과 인사과에서는 인사부에서는 저희는 화면에 예를 들어서 이름과

참석자 1 06:37  
직급만 나오게 해주세요. 가장 필요한 거 그다음에 이제 회계부 회계부에서는 저희는 직원들 이름과 급여만 있으면 돼요.   
이런 식으로 또 어디는 뭐와 뭐만 나와주세요.   
그래서 다 다르죠 원하는 바가. 그리고

참석자 1 06:58  
조직 전체에 있는 정보는 얼마나 많습니까? 그중에서 굉장히 간략화해서 필요한 것만 이렇게 보여주는 거죠.   
그래서 인사과에 있는 분은 회계과에 있는 이것에 대해서 관심이 없는 거죠.   
나는 이것만 보면 되기 때문에 이러한 개념을 이제 사용자 관점에서

참석자 1 07:22  
한 외부 스키마다라고 그래요. 그래서 이 외부 스키마는 이제 여러 개 존재할 수가 있겠죠.   
그다음에 이제 조직 전체가 가지고 있는 우리 이제 모든 사원들에 대한 정보 모든 기업 내부에 있는 정보가 개념 스키마가 되고

참석자 1 07:44  
실제로 이제 어떤 식으로 저장할 것인지라는 것에서 나타내는 게 바로 이제 내부 스키마가 된다.   
일단 구분을 먼저 해볼게요. 그래서 아파트에 보면은 얘는 이제 301호에 사는 사람 302호에 사는 사람 그래서 이게 이제 아파트에 살고 있는 사용자의 관점에서 보는 것이고 이제 옆집에 대해서 관심이 없는 거죠.

참석자 1 08:12  
그러니까 옆집에 집에 tv가 몇 대가 있는지 그런 것에 대해서 굳이 관심을 가질 필요가 없는 거죠.   
그것에 비해서 이제 개념 스키마는 이제 조직 전체의 입장이니까 경비 아저씨라든지 통장님이라든지 이런 분들은 이제 각 호에 살고 있는 거주하는 사람들에 대해서

참석자 1 08:37  
이제 관심이 있죠. 그래서 지나가면 저 사람은 몇 동 몇 호군 이렇게 이제 알 수가 있는 그런 부분이고 물리적인 저장 장치에서는 이제 건설업자입니다.   
기둥에 철근을 몇 개 박았지 이런 것들입니다.   
그래서 이제 정리를 해서 보면은

참석자 1 09:01  
외부 스키마는 사용자에게 필요한 데이터베이스를 이제 정의를 하게 됩니다.   
그래서 쇼핑몰을 할 때 고객 관리 담당자는 고객과 관련된 데이터만 필요한 거고 상품 관리는 상품 관리에 필요한 데이터만 이제 필요한 거죠.   
또 배송 담당이다. 그러면 배송에 관련된 데이터만 필요하게 되는 거죠.

참석자 1 09:32  
그래서 개별 사용자마다 관심 있는 분야가 다른 거죠.   
그래서 여러 개의 외부 스키마가 존재할 수 있다라는 부분입니다.   
그래서 전체 데이터베이스 중에서 사용자가 가지고 있는 관심의 일부다라고 그래서 서브 스키마다 또는 사용자 뷰

참석자 1 09:56  
라고도 이야기를 합니다. 이제 개념 스키마는 조직 전체의 관점에서 우리 조직의 내부에 있는 전체 데이터가 어떻게 어떻게 존재하고 그 데이터들 간에는 또 어떤 관계가 존재하고 어떤 제약 조건이 있는지를 말하는 거예요.

참석자 1 10:20  
그래서 하나의 개념 스키마만 존재를 하게 되는 거고 일반적으로 스키마다 라고 하면은 개념 스키마를 의미합니다.   
데이터베이스에 대한 이제 접근 권한 왜냐면 우리 조직에 있는 모든 정보기 때문에 아무나 들어와서 이 정보를 보면 안 되겠죠

참석자 1 10:43  
접근 권한 그러기 때문에 보안 정책이 필요한 부분입니다.   
그리고 이제 권한을 가진 사용자가 데이터를 이제 수정할 수 있어야 된다라는 거죠.   
어느 날 갑자기 내 월급이 400만 원인데 앞에 일자가 더 붙여가지고 1400만 원이 돼버렸다.   
이렇게 변경이 되면 안 되는 그런 부분입니다.

참석자 1 11:06  
그래서 허가된 사용자가 데이터를 수정할 수가 있어야 된다.   
그게 이제 무결성 규칙에 대한 내용이 되고요.   
그리고 이제 내부 스키마는 물리적인 저장 장치의 관점에서 이제 표현이 되는 거고요.   
하나의 데이터베이스에는 하나의 내부 스키마가 존재하게 되는 부분입니다.

참석자 1 11:28  
그래서 이제 중요한 점을 이제 꼽자.   
그러면 외부 스키마는 이제 사용자 여러 개 개념 스키마는 조직 전체 접근 거나 보안 정책 무결성 내부 스키마는 이제 저장 장치 하나 하나 이렇게 기억해 주면 되겠군요.   
3단계 데이터베이스 구조에 대해서 한번 살펴보겠습니다.

참석자 1 11:56  
그래서 여기에 이제 환자가 있고요 의사가 있고 이제 총무팀이 있어요.   
그래서 환자 화면에는 이제 환자명, 나이, 담당 의사 아이디가 있고 의사에는 환자명, 진단명, 처방약이 있고요.   
총무팀에서는 이제 환자명, 진료비, 검사 내역 각각의 이제 화면이 다르게 되는 거죠.   
그래서 이런

참석자 1 12:20  
전체 데이터들 중에서 일부의 내용만 각 사용자가 이제 원하는 바에 따라서 관심 있는 분야에 따라서 나타내주는 내용이 다른 거죠.   
이렇게 함으로써 불필요한 접근을 사전에 막아가지고 보안 측면에서도 이제 효과적이다라고 하는 부분입니다.

참석자 1 12:43  
그래서 여기 있는 환자 테이블에 이제 내용이 있는데 한번 연결을 해보면 얘는 이제 여기에 있는 것을 참조를 하겠죠 얘는 이제 기본 키가 될 거고 얘도 얘가 이제 기본 키가 되겠고요.   
그다음에 이제 진료에서는 진료 아이디가 아마 기본 키가 될 거고

참석자 1 13:03  
환자 아이디는 이제 여기를 이렇게 참조를 하게 될 것이고 의사 아이디는 이제 여기를 참조를 하게 될 것이고 이런 식으로 이제 이루어지겠죠 그래서 이렇게 모든 정보가 있는 곳을 우리는 개념 스키마 조직 전체의 관점이다라고 보는 거고

참석자 1 13:29  
그다음에 이제 저장 실제로 저장되는 저장소에 있는 실제 파일에 대한 경로, 이름 또 인덱싱 방식, 압축 방식, 저장 형식 이러한 것들을 이제 지정하는 게 내부 스키마가 됩니다.   
그런데 여기서 이제 보면

참석자 1 13:50  
나이라는 필드가 있는데 이제 이거를 필드명을 ag로 좀 바꿔주시겠습니까라고 한 거예요.   
그럼 바꿔도 되죠 필드명 바꾸는 거 되죠? 필드명을 바꿨을 때 이 외부 스키마의 내용들은 달라지냐라는 거죠.   
환자분들에 대한

참석자 1 14:13  
나이에 대한 내용이 달라지냐 그렇지 않습니다.   
이 필드명 연결을 여기를 참조를 하는데 이 참조되는 이름이 ag다라고만 해주면 이 내용물은 전혀 변화가 없죠.   
그래서 개념 스키마가 변화하더라도 외부 스키마에는 영향을 미치지 않는다.   
그걸 우리는 이제 논리적 독립성

참석자 1 14:37  
이라고 하는 부분이 됩니다. 그리고 이제 필드의 순서를 바꾼다든지 내부에 대한 이제 변동이 되더라도 이런 이제 개념 스키마에는 이제 영향을 미치지 않는 거죠.   
그걸 우리는 물리적인 독립성이라고 하는 부분입니다.

참석자 1 15:05  
우리가 데이터의 독립성과 종속성을 한번 정리를 해볼게요.   
종속이다. 그러면 응용 프로그램에 대한 이제 데이터의 종속성입니다.   
그래서 파일 시스템에서는 응용 프로그램과 데이터가 상호 의존적이었죠 그래서 데이터를 저장한 파일 구조가 변경이 되면은 이에 대응되는 응용 프로그램도 변경이 돼야 되는 그런 부분입니다.

참석자 1 15:33  
그래서 필드의 순서를 변경한다든지 크기를 변경한다든지 이름을 변경한다든지 그러면 이제 응용 프로그램에 영향을 미친다라는 게 데이터의 종속성이었고요.   
데이터의 독립성이다. 그러면 데이터 구조가 변경이 되더라도 응용 프로그램에 변경할 필요가 없는 것을 말을 하죠.

참석자 1 15:59  
그래서 여기는 이제 논리적인 독립성 외부 스키마와 개념 스키마에 대한 내용이죠.   
그래서 개념 스키마에 대한 내용을 변경하더라도 외부 스키머에는 영향을 미치지 않는다가 논리적인 독립성이었고요.   
물리적인 독립성이죠.

참석자 1 16:18  
내부 스키마의 내용을 변경하더라도 개념 스키마에는 영향을 미치지 않는다.   
그게 이제 물리적 독립성이었습니다. 데이터 독립성이 유지되지가 않으면 어떤 일이 발생이 될까? 데이터에 대한 중복성 복잡도가 증가가 되는 부분이고 요구사항 대응 난이도가 증가가 됩니다.

참석자 1 16:44  
그래서 데이터 유지보수 비용이 증가가 되는 부분입니다.   
예를 들어서 나이라는 필드가 있는데 이거를 이제 에이지로 바꿔주세요라고 이제 한 거죠.   
그럼 사용자가 만약에 이걸 요구를 했다.   
그럼 개발자가 네 그거 별거 아니에요.   
바로 고쳐드릴게요라고 고쳤어요. 그래도 다른 곳에 영향을 미치지 않게 되는 거죠.

참석자 1 17:10  
그런데 나이를 ag로 바꿔주시겠어요? 그러면 그거 정말 힘든데요.   
그래도 열심히 바꿔보도록 할게요라고 한 거예요.   
그러면 독립성이 유지가 되지 않는다.   
그러면 이것도 바꿔야죠 저것도 바꿔야죠 저것도 바꿔야죠.   
그렇게 되는 부분이죠. 그렇게 되면 이제 일단은 기간도 많이 들게 되는 거고 그것에 관련된 거 응용 프로그램도 다시 개발해야 되고 이런 여러 가지 문제가

참석자 1 17:39  
발생될 수 있기 때문에 그것에 대한 이제 비용이 증가가 되게 되는 거고 굉장히 복잡해지는 거죠.   
난이도가 증가가 된다라는 부분입니다.   
데이터 독립성에 대해서 정리를 해볼게요.   
논리적인 독립성이다. 개념 스키마를 변동하더라도

참석자 1 18:01  
외부 스키마에는 영향을 받지 않는다.   
개념 스키마에 대한 변경을 하더라도 외부 스키마에 영향을 미치지 않는다라는 게 이제 논리적인 독립성이 되는 거고 물리적인 독립성이다.   
내부 스키마를 변동하더라도 개념 스키마에는 영향을 미치지 않는다.   
개념 스키마가 변동이 없기 때문에 그 위에 있는 외부 스키마에도 영향을 미치지 않게 되는 거죠.

참석자 1 18:30  
그래서 결론은 저장 장치에 대한 구조 변경 이거는 응용 프로그램과 개념 스키마에 영향을 미치지 않는다 라는 게 이제 물리적인 독립성이 되겠습니다.   
그리고 이제 데이터베이스를 사용하는 사람들은 어떠한 사람들이 있을까라는 것을 보겠습니다.

참석자 1 18:55  
일반 사용자다. 그러면 응용 프로그래머가 작성한 프로그램을 이용하는 거죠.   
은행 창고라든지 관공서에 민원 접수처 등에서 데이터를 다루는 업무를 하는 거죠.   
그래서 특별한 지식이 없이 데이터베이스에 접근을 해서 검색하고 삽입하고 삭제하고 수정하는 작업을 할 수 있습니다.

참석자 1 19:22  
그다음에 이제 응용 프로그래머다. 일반 사용자가 응용 프로그램을 사용할 수 있도록 인터페이스를 이제 제공하는 거죠.   
그래서 프로그램을 이제 만드는 그런 부분이 되고요.

참석자 1 19:39  
sql 사용자는 이제 sql을 이용해가지고 업무 처리를 하는 it 부서의 담당자다.   
그래서 데이터에 대한 검색 데이터 구조 변경 데이터에 대한 이제 통계 처리를 하는 사람이다라는 것이고 데이터베이스 관리자 db a가 되고요.   
데이터베이스를 이제 총괄하는 거고

참석자 1 20:03  
데이터 전체를 파악을 하게 돼서 관리를 하게 되는 거고 데이터 이동 복사 이런 업무들 스키마에 대한 정의 또 데이터 설계 구현 유지보수 이런 이제 전 과정을 담당하는 게 이제 dba가 됩니다.   
그래서 이제 우리가 관심 있게 이제 살펴볼 사람은

참석자 1 20:26  
응용 프로그래머 일반 사용자가 응용 프로그램을 사용할 수 있도록 인터페이스를 제공하는 사람은 이제 응용 프로그래머다라고 하고요.   
그다음에 이제 데이터베이스에 관련된 전반적인 걸 책임지는 사람을 데이터베이스의 관리자 dba라고 한다라는 거죠.   
그러면 이제 언어에 대해서 살펴보겠습니다.   
언어는 크게

참석자 1 20:52  
세 가지로 구분을 하게 되는데요. 정의어 조작어 제어 이렇게 되는 부분이에요.   
그중에서 우리는 이제 조작어를 많이 쓰겠죠 정의어다라고 하면 이제 디파인하다 정의요 ddl이다라고 이야기를 해요.   
그래서 데이터베이스에 대한 구조 형식 접근 방식

참석자 1 21:15  
이렇게 데이터베이스를 이제 구축하거나 수정할 목적으로 사용하는 부분이에요.   
그래서 스키마를 기술하기 위해서 사용이 되고요.   
주로 데이터베이스 이제 설계자가 사용되는 부분입니다.   
그래서 예를 들어서 이제 지금부터 학생이라는 테이블을 만들겠습니다라고 이제 우리가 보는 거예요.

참석자 1 21:40  
그러면 학생이라는 테이블에서는 뭐가 필요하죠? 이름 그다음에 번호 일단 예를 들어서 주소 이런 게 이제 필요하다라고 했대.   
이걸 지금부터 만들어주세요라는 게 큐리에이트 생성이 되는 것이고요.   
만들고 났는데 여기다가 하나 좀 빼먹었어요.

참석자 1 22:05  
담임 교사 네 그럼 이제 새로 추가해야 되죠.   
또는 이제 여기 번호를 학번이라고 바꾼다든지 이렇게 이제 구조를 변경할 때 쓰는 게 알트 명령어가 있어요.   
그리고 이제 상황에 따라서

참석자 1 22:22  
학생 테이블 전체를 이제 제거를 하는 경우 우리는 이제 드랍한다라고 이야기를 하게 되는 부분입니다.   
이러한 것들이 이제 정의에 해당이 되는 것이고요.   
그다음에 이제 조작어가 dml이 있습니다.   
사용자와 이제 데이터베이스 간에 이제 인터페이스를 제공을 하는 부분이고요.

참석자 1 22:45  
데이터를 이제 검색해 주세요. 찾아주세요라는 거죠.   
서울시에 살고 있는 40세 이상의 남자분들의 명단 좀 보여주세요.   
이런 식으로 검색을 할 때 쓰는 게 셀렉트가 있고요.   
새로운 신입생이 들어왔어요. 전 학생이 들어왔어요.

참석자 1 23:06  
네 그다음에 신규 사원이 입사를 했어요.   
그러면 이제 명단에 추가를 해야 되겠죠 그런 게 이제 그런 걸 할 때 인서트를 쓰게 되는 것이고요.   
이제 퇴직을 했거나 전학을 갔어요. 이제 삭제 처리가 이루어져야 되겠죠 그럴 때 쓰는 게 딜리트 명령어가 되고요.   
그다음에 제 핸드폰 번호 바뀌었거든요 수정해 주세요라고 할 때 갱신 업데이트가 된다.

참석자 1 23:35  
그다음에 이제 제어가 이제 dcl이다라고 하는 부분입니다.   
제어하는 거죠 네 무결성 보호하는 권한 제어 회복 등을 하기 위한 언어예요.   
그래서 아무나 이 테이블을 보면 안 되겠죠 그래서 홍길동은 테이블을 볼 수 있는 권한이 있어요라는 게 이제 그랜츠 명령어가 되고요.   
이제 이분이 퇴직을 했어요.

참석자 1 24:01  
그러면 이제 홍길동은 이 테이블을 볼 수 있는 권한이 없어집니다라는 게 리보크 명령어가 되고요.   
그다음에 이제 데이터베이스를 이제 수행을 하게 되는데 이렇게 이제 수행을 하다가 1번을 내가 완벽하게 끝냈어요.   
그러면 저희 완벽하게 끝냈어요라고 이제 컴이 명령어를 내려주게 됩니다.   
그러면 이제 그때 db를 저장을 하게 되는 부분이죠.

참석자 1 24:30  
이번 작업을 완벽하게 끝냈어요. 그러면 저희 완벽하게 끝냈어요라고 하고 이제 커윈 명령어를 내려주게 된다.   
그런데 이제 수행을 하다가 여기서 이제 에러가 났어요.   
저희 에러가 났어요. 되돌려야 될 것 같아요.

참석자 1 24:49  
그러면 이제 롤백이라는 명령어를 내려주게 되는데요.   
커밋 할 때마다 여기다가 이제 깃발을 세웠다라고 생각해 주면 돼요.   
롤백 해야 됩니다. 어디로 돌아갈까요? 깃발이 있는 곳으로 돌아갈 수 있어요.   
이렇게 하는 것을 우리는 이제 롤백이다라고 해요.

참석자 1 25:10  
그래서 또 tcl에서 tcl에서 별도로 tcl을 분리해서 나타내기도 하는 부분입니다.   
네 그래서 이제 데이터 언어를 구분을 해보면 독립 실행형이다 내장형이다 이렇게 이제 구분할 수가 있는 부분인데 독립 실행형이다.   
그러면 sql 인터페이스를 이용을 해가지고 sql 쿼리를 직접 dbms에 입력하는 방식이 되고요.

참석자 1 25:45  
내장형은 씨나 시볼불 자바 등의 프로그램 언어에 들어가서 기술되는 거예요.   
그래서 들어가서 된다 해서 내장형이다라고 하는 부분입니다.   
그다음에 절차형과 비절차형이다. 절차형은 프로그램 명령어들을 순차적으로 실행하는 부분이에요.

참석자 1 26:09  
그래서 이제 포트라는 아주 옛날에 나온 최초의 고급 언어 프로그램이 되죠.   
시어노 파스칼 자바 이러한 것들이 있어요.   
그래서 예를 들어서 이제 여러분이 시어노에서 intaa를 정수형 변수로 선언하겠습니다.   
그리고 a에다가 20을 주겠습니다. 이런 식으로 이제 진행을 하게 되는 부분이죠.   
a는 20이고요.

참석자 1 26:36  
그다음에 inta에요. 이렇게 하게 되면 어떠세요? 오류가 발생이 되죠 그래서 프로그램 명령어들을 이제 순차적으로 실행하는 걸 우리는 절차형이다라고 하고요.   
비절차형은 문제 해결의 절차를 명시하지 않고요.   
오직 원하는 결과만을 명세를 합니다.

참석자 1 27:00  
그래서 대표적으로 비절차형의 sql이 있어요.   
네 sql 명령어는 예를 들어서 네 21세 이상의 사용자의 이름 나이를 모두 검색해 주세요.   
이런 식으로 이제 우리가 사용할 수 있는 부분이죠.

참석자 1 27:20  
네 그리고 이제 용어적인 면을 살펴보면 이제 데이터 사전이다 라는 말이 있습니다.   
또 메타 데이터다라고 있는데요. 우리가 이제 많이 혼돈돼서 같은 의미로도 이제 사용되는 부분인데 여기에 이제 메타 데이터다.   
메타가 이제 모아놓다라는 뜻이죠. 종합적인

참석자 1 27:41  
여기에 이제 데이터 사전이 있어요라고 이제 생각해 주면 되겠습니다.   
그래서 메타데이터다. 그러면 데이터에 대한 데이터다.   
데이터의 유형 관계 수정 날짜 이런 이제 다양한 정보들을 이제 포함하고 있는 부분이 됩니다.

참석자 1 28:02  
그리고 이제 데이터 사전이다. 그러면 데이터베이스 내에 테이블에 대한 이름 칼럼 이름 데이터에 대한 유형 기본 키 외래 키 이런 정보들을 이제 포함하고 있는 부분이에요.   
그래서 스키마 사상 정보 다양한 제약 조건 등을 이제 저장을 하게 되어서 다른 말로는

참석자 1 28:27  
시스템 카탈로그다라고도 이제 하는 부분이 됩니다.   
그래서 스키마가 있고 거기에 대한 이제 매핑이 있으면 그 매핑 정보도 어딘가에는 저장해 둬야 되는 그런 부분입니다.   
그런 것들을 이제 저장하는 곳이다라고 하는 부분이에요.   
그래서

참석자 1 28:47  
데이터 사전은 데이터에 대한 정보를 저장을 하면서 메타 데이터에 대한 한 부분인데 굉장히 많은 부분을 차지하고 있기 때문에 데이터 사전 메타 데이터 이런 것들은 이제 혼용해서 사용되고 있는 부분인데 엄밀히 말한다면 메타데이터가 가장 넓은 의미가 되겠습니다.

참석자 1 29:14  
그래서 이제 여기까지 한 것을 이제 여러분이 이렇게 개념적으로 정리를 하고 있으면 좋을 것 같아요.   
데이터베이스가 있고 dbms가 있고 스키마가 있고 스키마에는 이렇게 세 가지가 있었습니다.   
외부 개념 내부 데이터베이스를 사용하는 사람들 여기에 이제 응용 특히 응용 프로그래머 데이터베이스 관리자가 있고

참석자 1 29:39  
데이터베이스 언어 ddl dml dcl이 있었다라고 이제 전반적으로 정리하면 되겠고요.   
여기까지 이제 여러분이 학습을 했으면 여러분이 외부 스키마 개념 스키마 내부 스키마에 대해서 설명할 수 있으면 좋겠어요.

참석자 1 29:59  
그리고 문제 한번 풀어볼게요. 사용자가 데이터에 대한 삽입 삭제 수정 검색에 대한 처리를 db ms에게 요구하기 위해서 사용되는 언어는 무엇입니까? 이러한 기능 등을 하는 것을 우리 뭐라 그러죠? 바로 이제 dml이다라고 하죠.   
개념 스키마에 대한 설명으로 옳지 않은 것은 무엇입니까? 그럼 개념 스키마는 뭘까? 우리 이제 조직의 전체 내용을

참석자 1 30:29  
저장하는 걸 나타내는 것을 말을 하죠.   
그래서 데이터베이스 파일에 저장되는 데이터의 형태를 나타낸 것이다라는 거고 일반적으로 스키마다 그러면 개념 스키마를 말하는 거죠.   
범 기관적인 입장이다. 또 통합한 조직 전체의 데이터베이스로 하나만 존재한다.

참석자 1 30:52  
사용자나 응용 프로그래머가 접근하는 데이터베이스 이거는 뭐라 그러죠? 바로 이제 외부 스키마다라고 하는 부분이죠.   
그래서 틀린 것은 4번이 되겠습니다.   
여기까지 데이터베이스 시스템에 대한 내용이었어요.   
여러분 수고하셨습니다.

clovanote.naver.com