데이터베이스 설계 과제

주제:병원

|  |  |  |
| --- | --- | --- |
| 조 원 | 학 번 | 학 과 |
| 김홍준 | 2014 | 컴퓨터공학과 |
| 윤기현 | 2014150023 | 컴퓨터공학과 |
| 이정찬 | 2014154031 | 컴퓨터공학과 |
| 김동진 | 2015 | 컴퓨터공학과 |



설계 첫 단계: 요구사항분석

어떤 한 병원에 대한 정보들과 요구를 분석하였다.

병원에는 직원이 근무한다. 병원에 총 직원은 12명이고 한 부서에 3명(의사, 간호사, 간호사)이 들어 간다. 병원은 4층 건물이다.

직원의 얘기를 들어보면 병원에서 주로 약품의 재고량을 검색하는 등 약품에 대해 검색을 자주하게 되고 약에 대해서 언제 공급이 되는지 묻기 위해 공급자에게 연락할 일이 있다. 병원 고객이 찾아 올 경우 대부분 환자의 보호자나 관계자이기 때문에 환자의 정보에대해서도 검색을 많이 하게 된다. 또한 보호자에게 동의, 환자에게 일이 발생 등 환자에 대해 연락할 일이 많다. 직원들도 서로간의 일이 생길 경우를 대비해서 직원 대한 정보도 검색 할 기능이 있어야한다고 한다. 특히 병실에 입실 한 환자에게 일이 생길 경우가 많아 병실 담당직원에게 연락 할 일이 많다고 한다. 또한 직원은 여러 가지를 검색 할 수 있다. 모든 정보를 검색 할 수 있고, 단 환자와 직원 검색 시 보안상 주민번호는 제외하고 검색된다. 의사는 환자의 수술명과 수술 날짜를 알아야한다. 병원 시스템 상 진단서 번호는 1번부터 시작한다. 약품 번호는 101번부터 시작한다. 부서 번호는 1, 2, 3, 4 가 있다. 직원 번호는 의사는 110부터 시작하며 간호사는 210부터 시작한다. 직원은 연봉이 병원 조건상 100000000을 넘을 수 없다. 공급자 번호는 4100번부터 시작한다. 환자 번호는 10번부터 시작한다.

직원에는 직원 ID, 나이, 이름, 층수, 급여, 직책, 주민번호, 전화번호이 있다.

직책은 의사와 간호사가 있다.

부서는 부서 이름, 부서 번호, 층으로 나뉜다. 층수는 건물이 1층부터 4층까지 있다고 한다. 그리고 각 부서는 한 층에 한 개씩 들어가 있다고 한다. 부서는 소아과, 신경외과, 신장내과, 정형외과가 있다.

환자는 환자ID, 혈액형, 병실, 나이, 담당의사ID, 질병 이름, 환자 이름, 주민번호, 주소( 시 , 구 ,동) ,전화번호로 이루어져있다. 환자들은 주로 경기도지역에서 온다고 한다. 주민번호 앞 두 자리로 나이를 대략 파악 할 수 있다.

환자는 같은 직원에게 진단서, 수술, 진료를 받는다. 진료는 여러 직원들에게 받을 수 있다.

환자는 여러 질병을 가질 수 있다. 수술과 진단은 의사만 내릴 수 있다고 한다.

환자보호자는 약한 엔티티타입으로 이름, 혈액형, 전화번호로 이루어져있다. 보호자는 보통 부(父)를 따른다.

즉 환자의 이름(소유엔티티의 기본키)과 환자보호자의 이름(부분키)로 사용한다.

병실은 병실 호수로 이루어져있다. 병실은 환자 3명을 수용한다고 한다. 병실은 1001호 1002호 1003호 총 3개호로 이루어져 있다.

약품은 약품 번호, 약품 이름, 제조회사로 이루어져있다. 공급자는 공급자 ID와 이름, 전화번호로 이루어져 있다. 약품은 의약 제품만 쓴다고 한다. 약품은 직원 중 의사, 간호사들이 관리를 함께 한다. 직원 모두가 약품은 공급자에게 받을 수 있다. 병원과 관계를 맺은 제조회사는 대웅제약, 광동제약, 코오롱제약, 메디카코리아, 종근당, 한국피엠지제약, 안국약품, 삼진제약 이 있다.

직원은 병실을 담당한다. 직원은 2개 이상의 병실을 담당 할 수 없고 병실은 직원 2명 이상이 담당될 수 없다.

직원은 부서 한 개에 속하고 부서는 직원이 여러 명 있다.

직원 중 직책(의사)는 한 명 이상의 환자를 **수술 ,진료** 할 수 있다.

환자는 한 의사에게 **수술을 받고** 여러 의사에게 **진료를 받는다.** 의사는 여러 환자들을 **진료한**다.

**수술관계**에는 수술명과 수술 날짜 에트리뷰트가 있다.

**진료관계**에는 진료 일자 에트리뷰트가 있다.

환자는 0명 이상의 보호자를 두고 보호자는 한 명의 환자를 둔다.

환자는 병실에 입실 할수도 있고 안 할 수도 있다.(부분참여) 입실 관계에는 입실날짜 에트리뷰트가 있다. 환자는 각자 한 개의 병실에 입실하고, 병실은 여러 명의 환자들을 수용한다.

의사는 환자 여러 명에게 진단서를 내린다. 진단 내리다 관계에는 진단서 에트리뷰트가 있다.

몇 명의 직원 중 직책(간호사)는 약품을 **관리한다.**

약품은 여려 명의 간호사를 수반한다.(M:N) 즉 간호사는 여러 약품을 관리하고, 약품은 여러 간호사들에게 관리 받는다.

병원과 관계를 맺은 공급자는 6명이 있다.

공급하다 관계에는 수량(QUANTITY) 에트리뷰트가 있다.

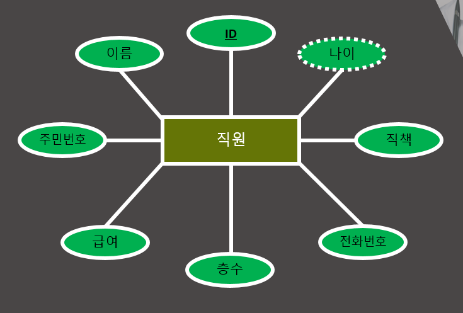
공급자는 어떤 약품을 직원들에게 공급한다. 이때 얼마나 공급 하는가를 나타낸다.(M:N)

공급자는 여러 개의 약품들을 공급한다. 약품들은 여러 공급자들에게 공급받는다.(M:L)

최종 시스템=> 병원에서 직원의 요구, 기능 분석하여 구현

설계 두 번째 단계: ER다이어그램

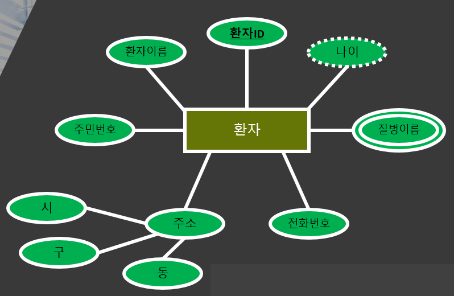
요구사항을 분석하여 ER다이어그램을 설계했다.

요구사항을 토대로 엔티티와, 관계, 애트리뷰트들을 뽑아 냈다.

<직원>엔티티

직원에는 직원 ID, 나이, 이름,

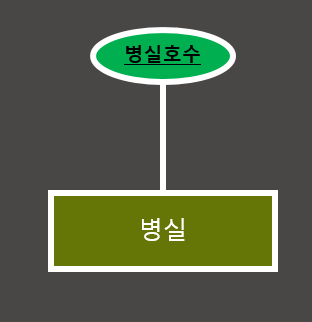
층수, 급여, 직책, 주민번호, 전화번호가 있다.



<환자>엔티티

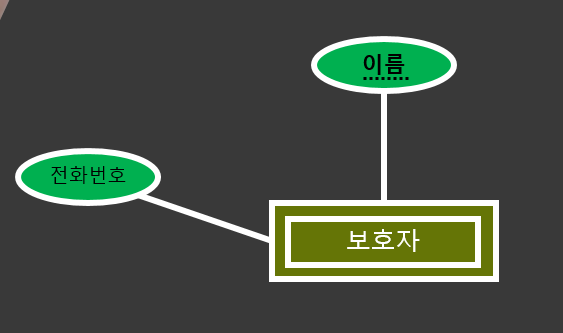
환자는 환자ID, 혈액형, 병실, 나이, 담당의사ID, 질병이름,

환자이름, 주민번호, 주소( 시 , 구 ,동) ,전화번호로 이루어져 있다

병실은

병실 호수로

이루어져 있다

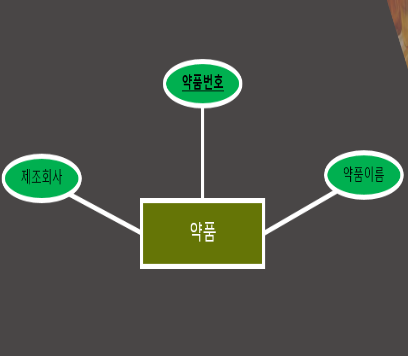
<보호자>

환자보호자는

약한 엔티티타입으로

**이름**, 전화번호로

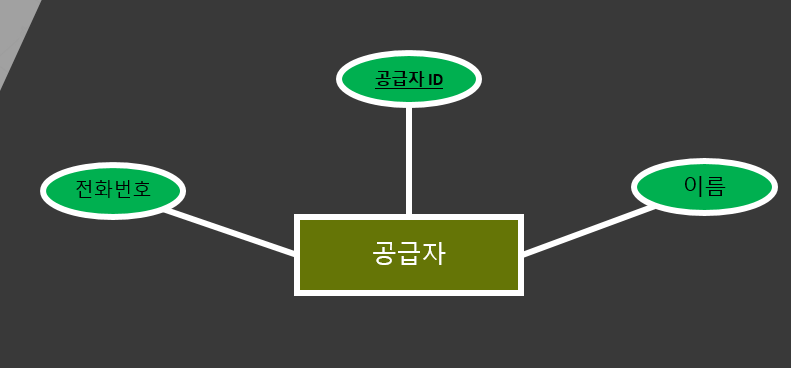
이루어져 있다.

약품은

약품번호, 약품 이름,

제조회사로

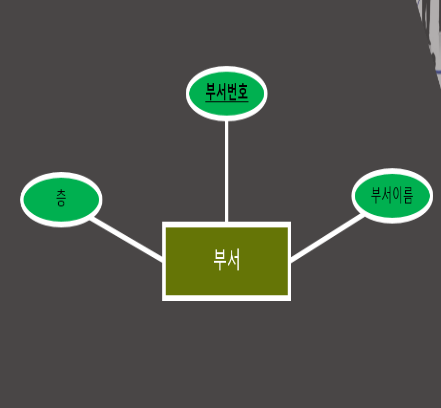
이루어져있다

공급자는

공급자 ID와 이름,

전화번호로

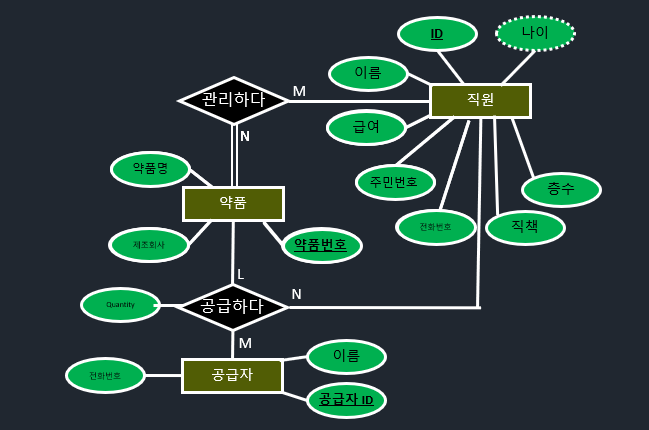
이루어져 있다

부서는

부서 이름, 부서 번호,

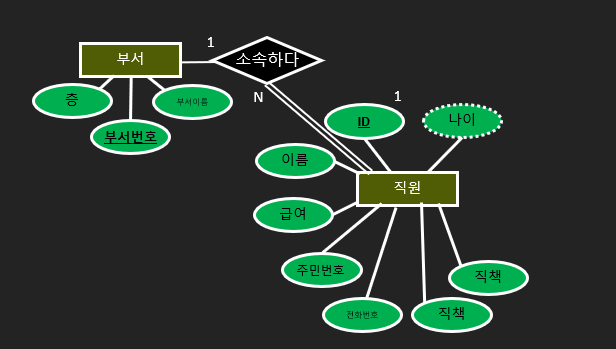
층으로 나뉜다

그 이후 관계를 형성하여 전체적인 ER다이어그램을 나타냈다.

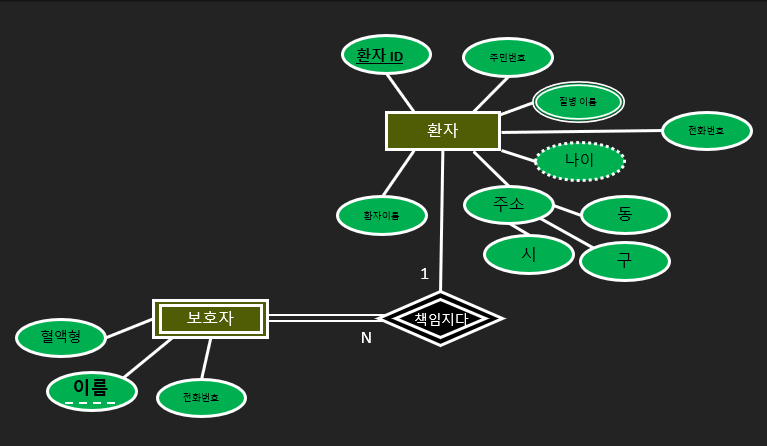


앤티티는 총 7개로 부서, 약품, 공급자, 직원, 병실, 환자, 보호자가 있다. 관계로는 총 9개로 공급자와 약품 , 직원 사이의 3진관계로 M:L:N 관계이다. 공급자는 약품 몇 개를 직원에게 공급한다.

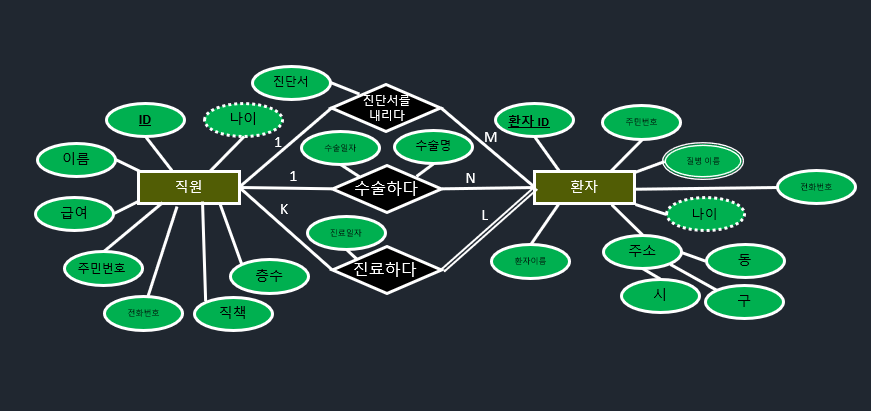
공급자는 약품 여러 개를 직원 여러 명에게 제공한다. 부서와 직원 사이 소속하다 관계가 있다.



부서는 직원 여러 명을 포함하고 직원은 한 부서에 소속되기 때문에 1:N이다.



보호자는 약한엔티티로 소유엔티티인 환자엔티티에서 기본 키인 환자id를 같이 써야한다. 이때 보호자엔티티에서 이름이 부분 키이다. 책임지다라는 관계를 통해 보호자는 환자 한 명을 책임지고 환자는 여러 보호자를 가질 수 있다. 1:N관계이다.

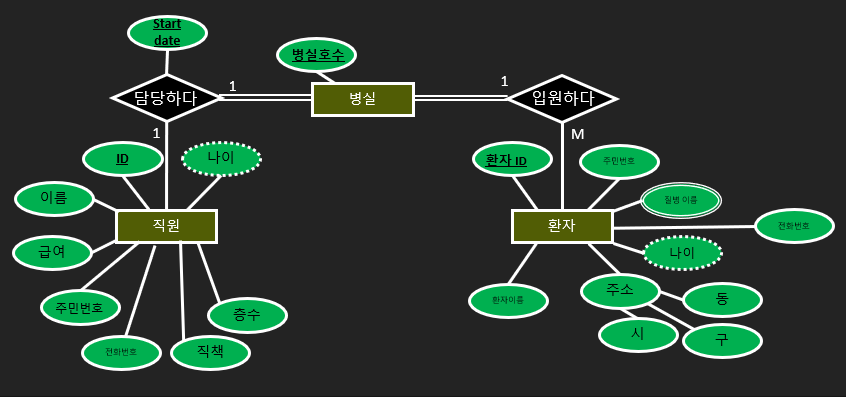


직원 중 직책(의사)는 한 명 이상의 환자를 수술 ,진료 할 수 있다. 환자는 한 의사에게 수술을 받고 한 의사는 환자 여러 명을 수술 할 수 있다. (1:N)

수술관계에는 수술명과 수술 날짜 에트리뷰트가 있다. 환자는 여러 의사에게 진료를 받는다. 의사는 여러 환자들을 진료한다. (K:L)

진료관계에는 진료 일자 에트리뷰트가 있다.

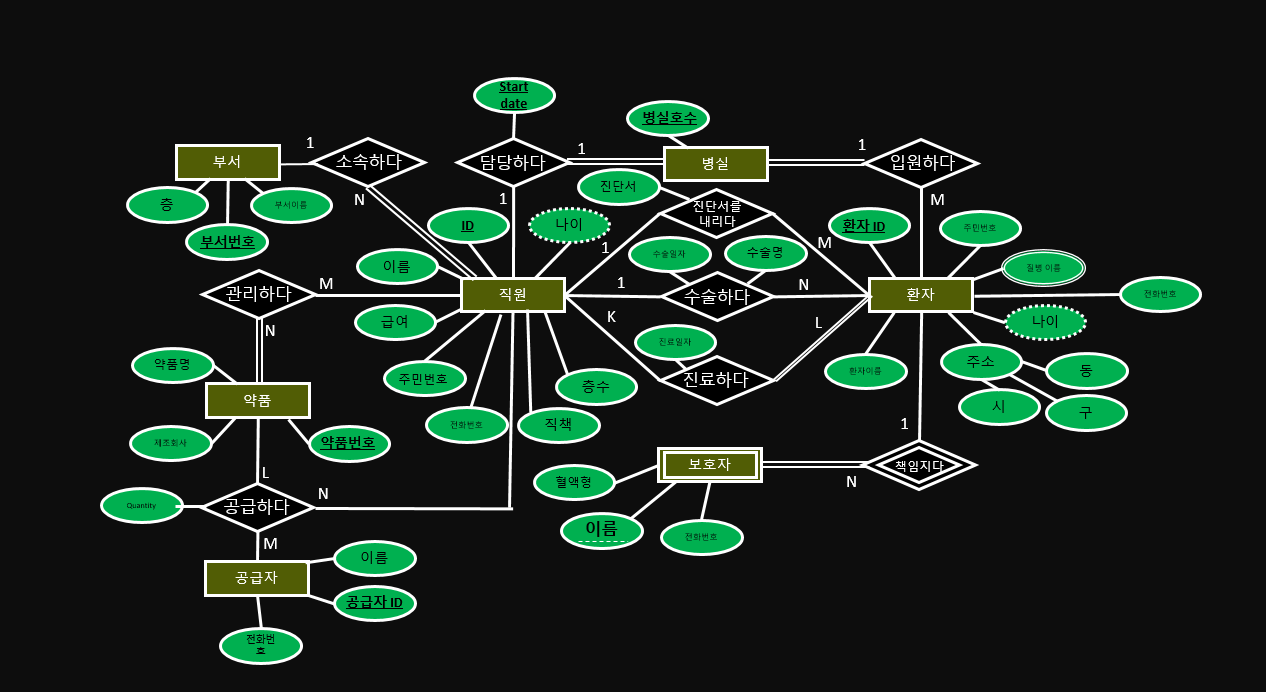
의사는 환자 여러 명에게 진단서를 내린다. 환자는 직원 한 명에게 진단서를 받는다.(1:M) 진단 내리다 관계에는 진단서 에트리뷰트가 있다.



환자와 직원 병실 관계에서는 환자는 병실에 입실 할 수도 있고 안 할 수도 있다.(부분 참여) 입실 관계에는 입실 날짜 에트리뷰트가 있다. 환자는 각자 한 개의 병실에 입실하고, 병실은 여러 명의 환자들을 수용한다.(1:M)

병실은 환자에 의해서 모두 참여하므로 전체 참여이다.

직원은 병실을 담당한다. 직원은 2개 이상의 병실을 담당 할 수 없고 병실은 직원 2명 이상이 담당 될 수 없다.(1:1) 병실은 모두 참여하므로 전체 참여이다.

이렇게 ER다이어그램이 설계되었다.

설계 세 번째 단계: 논리적 설계(논리적 스키마)

ER다이어그램을 통해서 논리적 스키마를 형성했다.

위의 ER다이아그렘을 토대로 논리 스키마를 작성해 보겠다.

논리스키마는 총 7단계로 구분이 된다.

1단계에서는 정규 앤티티 타입과 단일 값 애트리뷰트에 대해서만 관계 모델의 릴레이션으로 사상을 한다. ER 스키마의 각 정규 앤티티 타입에 대해 하나의 릴레이션을 생성 함으로 여기에 포함되는 릴레이션들은 총 6개 이다. (직원, 환자, 부서, 병실, 약품, 공급자)

EMPLOYEE (EMPNO, EMPNAME, SALARY, EMPIDNO, EMPPHONENO)

PATIENT (PATNO, PATNAME, PATIDNO, CITY, GU, DONG, PATPHONENO)

DEPARTMENT (DEPTNO, DEPTNAME, FLOOR)

ROOM (ROOMNO)

MEDICINE (MEDNO, MEDNAME, MANUFACTURER)

SUPPLIER (SUPPNO, SUPPNAME, SUPPHONENO)

그 다음으로 2단계는 약한 엔티티 타입과 단일 값 애트리뷰트에 대해서 진행을 하면 된다. 우리가 진행한 ER다이아그렘상 약한 앤티티 타입은 보호자 뿐이다.

PROTECTOR (PATNO, PROTNAME, PROTPHONENO)

여기서 소유 엔티티인 환자 릴레이션의 기본 키 PATNO를 PROTECTOR 릴레이션에 외래 키로 포함시키고 PROTECTOR의 부분 키인 PROTNAME 와 환자 릴레이션의 기본키를 참조하는 외래 키 PATNO가 모여서 PROTECTOR의 기본 키가 된다.

다음은 2진 1:1 관계에 대해서 고려를 해야한다.

ROOM (ROOMNO, STARTDATE, MANAGER)

ROOM릴레이션이 전체참여를 하기 때문에 직원의 기본 키를 ROOM릴레이션에 외래키로 MANAGER로 포함시킨다. 이때 기본 키 구성은 아니다. 관계 에트리뷰트인 STARTDATE도 포함 시킨다. 기본 키는 ROOMNO가 된다.

전체 참여에 외래 키를 두는 이유는 부분 참여에 외래 키를 둘 경우 불필요한 경우가 생기기 때문이다.

다음은 정규 2진 1:N 관계 입니다.

아무래도 병원은 환자를 위한 시설이기에 환자가 제일 많이 관계가 연결되어 있다.

직원 - < 진단서를 받다 > - 환자 --> 환자 쪽이 전체 참여라 환자 릴레이션 + 관계에 대한 애트리뷰트 들과 직원의 기본 키가 외래 키로 쓰임으로써 밑에 와 같이 형성이 되었다.

PATIENT (PATNO, PATNAME,PATIDNO,CITY,GU,DONG,PATPHONENO,REPORT,ENO)

직원 - < 수술하다 > - 환자 --> 여기도 마찬가지로 환자 릴레이션 + 관계에 대한 애트리뷰트들과 직원의 주키가 외래키로 쓰임으로써 밑에와 같이 형성이 되었다.

PATIENT (PATNO, PATNAME, PATIDNO, CITY, GU, DONG, PATPHONENO, SURGEONADATE, SUGEONNAME, ENO)

환자 -< 입원하다 >- 병실 --> 여기도 마찬가지로 환자 릴레이션 + 관계에 대한 애트리뷰트들과 병실의 주키가 외래키로 쓰임으로써 밑에와 같이 형성이 되었다.

PATIENT (PATNO, PATNAME, PATIDNO, CITY, GU, DONG, PATPHONENO, RONO )

직원 -< 소속하다>- 부서 --> 직원 릴레이션 + 관계에 대한 애트리뷰트는 여기선 없고 부서에 대한 주키가 외래키로 쓰임으로써 밑에와 같이 형성이 되었다.

EMPLOYEE (EMPNO, EMPNAME, SALARY, EMPIDNO, TITLE, EMPPHONENO, DNO )

다음으로 M:N 관계 이다.

직원 -< 관리하다 > - 약품 --> 여기서는 직원과 약품의 관계가 M:N관계로 이루어져 있어서 릴레이션을 관계로 뽑아서, 릴레이션은 직원의 기본 키와 약품의 기본 키로 이루어져 있다. 여기서는 관계에 대한 애트리뷰트가 없어서 추가가되는 상황은 없다. 두 키의 조합이 기본 키다.

SUPERVISE (EMPNO, MEDNO)

환자 -< 진료하다 >- 직원 --> 여기서는 위와 다르게 관계에 대한 애트리뷰트인 “TREATEDATE” 가 있기 떄문에 릴레이션을 환자와 직원의 기본 키들과, 관계 애트리뷰트로 이루어 져있다. 두 키의 조합이 기본 키이다.

TREATE (PATNO, EMPNO, TREATDATE)

다음으로 3진 관계이다.

여기서는 3진관계가 공급하다 관계로 이루어져 있다.

공급자 -< 공급하다 >- 약품

공급자 -< 공급하다 > - 직원

먼저 릴레이션을 뽑고 설명을 드리자면,

SUPPLY (SUPPNO, MEDNO, EMPNO, QUANTITY) 이렇게 이루어져 있다.

3진 관계의 관계로써 릴레이션의 이름을 뽑고, 그 다음으로 3진 관계에 연루된 앤티티들의 기본키를 애트리뷰트들로 뽑는다. 물론 관계에 관한 애트리뷰트도 뽑아야 한다.

마지막으로 다치 관계에서 고려를 해보자.

각 다치 애트리뷰트에 대하여 릴레이션을 생성을하고, 이 릴레이션의 기본 키는 다치 애트리뷰트와 환자의 기본 키가 PAT\_DISEASE에서 외래키로 작용을 하므로 두 개의 조합이 기본 키가 된다.

PAT\_DISEASE (PATNO, DESEASENAME)

설계 네 번째 단계: 논리적 설계(정규화)

논리적 스키마를 가지고 갱신 오류, 삽입 이상, 삭제 이상, 수정 이상을 최소화하고 중복을 최소화 하기 위해서 릴레이션 분해를 통해 정규화를 진행하였다.

## 1정규화, 2정규화

분석 결과 모든 논리적 스키마가 원자 값들을 가지기 때문에 1정규화를 만족하고 후보 키에 속하지 않는 모든 에트리뷰트가 릴레이션의 기본 키에 완전하게 함수적으로 종속하기 때문에 2정규화를 만족한다.

## 3정규화

1,2정규화를 만족하지만 이행 적 종속성이 있는 경우가 있었다.

EMPLOYEE (EMPNO, EMPIDNO, EMPPHONENO, TITLE, DNO)

EMPLOYEEID (EMPIDNO1, EMPNAME, SALARY)

* EMPINDNO(주민번호)가 시스템상 직원 이름과 봉급을 알 수 있다.
* 봉급은 보안상 주민번호를 알 경우 알려지게 됨.

DEPARTMENT (DEPTNO, DEPTNAME)

DEPARTMENTNAME (DEPTNAME1, FLOOR)

* 부서이름으로 층 수도 알 수 있다.

MEDICINE (MEDNO, MEDNAME)

MEDICINENAME (MEDNAME1, MANUFATCTRER)

* 약품 이름으로 제조회사를 알 수 있다.

PATIENT (PATNO, PATIDNO, PATPHONENO, ENO, REPORT, SURGEONDATE, SURGEONNAME, RONO)

PATIENTIDNO (PATIDNO1, PATNAME, CITY, GU, DONG)

* 환자의 주민번호를 통해서 이름과 주소를 알 수 있다.

BCNF정규화는 없었다.

이를 통해 나온 릴레이션들은 아래와 같다.

EMPLOYEE (EMPNO, EMPIDNO, EMPPHONENO, TITLE, DNO)

EMPLOYEEID (EMPIDNO1, EMPNAME, SALARY)

DEPARTMENT (DEPTNO, DEPTNAME)

DEPARTMENTNAME (DEPTNAME1, FLOOR)

MEDICINE (MEDNO, MEDNAME)

MEDICINENAME (MEDNAME1, manufacturer)

PATIENT (PATNO, PATIDNO, PATPHONENO, ENO, REPORT, SURGEONDATE, SURGEONNAME, RONO)

PATIENTIDNO (PATIDNO1, PATNAME, CITY, GU, DONG)

ROOM (ROOMNO, STARTDATE, MANAGER)

SUPPLIER (SUPPNO, SUPPNAME, SUPPHONENO)

PROTECTOR (PATNO1, PROTNAME, PROTPHONENO)

SUPERVISE (EMPNO1, MEDNO1)

TREAT (PATNO2, EMPNO2, TREATDATE)

SUPPLY (SUPPNO1, MEDNO2, EMPNO3, QUANTITY)

PAT\_DISEASE (DESPATNO, DESEASENAME)

설계 다섯 번 째 단계: 물리적 설계(인덱스 설정+질의)

요구자의 질의를 통해서 쿼리 문을 작성하면서 인덱스 선정을 통해 물리적 설계를 구현함

시나리오

* 해당 직원들의 급여를 통계내기위해 검색하는 일이 잦다.
* 직원들은 부서별로 평균 급여와 최대 급여를 통해 분석한다.
* 직원은 1003호실에 문제가 발생했다. 직원은 1003호실 관리 직원에게 전화를 해야하는 상황인데 전화번호를 모른다. 이때 해당 직원의 전화번호를 검색한다.
* 직원들은 모든 부서, 환자, 공급자 등등 에 관한 모든 정보들을 검색한다. 단 주민번호가 있을 경우 주민번호는 제외하고 검색한다.
* 직원은 약을 찾는 도중 타이레놀이 부족한 것을 알게되었다. 타이레놀의 약품 번호를 통해서 그것을 공급하는 공급자의 전화번호를 검색한다.
* “\_\_\_” 환자가 임종직전이다 즉 급한 일이 생겨 환자 보호자에게 연락 할 일이 생겼다. 그 환자의 보호자 연락처와 이름을 검색 한다.
* “신장암” 수술은 나이가 고령일수록 ( 고령 기준 70년생 밑으로 ) 수술 성공률이 희박하다.”\_\_\_\_” 수술이 잡힌 환자들에게 주의 할 부분을 알려주기 위해 환자 전화번호를 검색한다.
* 감기가 창궐해서 감기 관련해서 예방교육을 다시 직원들에게 교육을 시켜야 한다. 감기 질병을 지닌 환자의 담당 직원 이름을 검색 하시오.
* 타이레놀을 공급하는 약품회사가 부도났다. 당장 쓸 수 있는 타이레놀의 재고량을 검색한다.
* 지진 피해로 인해 외상을 입은 환자들이 많아 진지 한달 후, 현재 상황에 대해서 회의 통보를 문자로 전송하기 위해 각 외과 직원들의 전화번호를 검색한다.
* 스케줄을 조정하기 위해 소아과에서 근무하는 직원의 연락처가 궁금해졌다. 소아과에 근무하는 직원들의 전화번호를 검색한다. 각각 1, 2, 3, 4 부서에 속하는 직원의 정보를 검색( 자기 자신 말고 다른 사람(직원) 정보가 궁금할 때)
* 직원은 진료실에 찾아온 환자의 보호자나 관계인이 수술에 대해 물어 본다. 성함을 통해서 수술 날짜와 수술 명 입실 할 병실을 찾을 수 있다.
* 직원은 약품을 관리하는 도중 모 회사에서 만드는 약품이 궁금하여 모회사가 만든 약품을 검색한다. “ “회사에서 만든 약품을 검색한다.
* 직원이 환자 정보를 써놓는 도중 간단한 주소가 궁금해져서 환자의 정보 중 주소를 검색한다.
* 직원은 갑작스럽게 \_\_\_환자의 쇼크현상으로 인해 그 환자가 입원한 병실의 담당직원을 호출 하기 위해 직원전화번호가 궁금해졌다.
* 직원은 수술 일자가 \_\_\_인 환자의 보호자에게 동의여부를 묻기 위해 보호자 전화번호가 필요하다.
* 해당 약품이 거의 떨어져서 이 약품을 관리하는 직원의 전화번호가 궁금해졌다. 해당 직원의 전화번호를 검색한다.

요구분석글을 통해 사용자가 자주 하는 질의나 많은 튜플 수를 가질 수 있는 경우에 대해서 인덱스를 설정 하였다.

우선 쿼리 문을 작성을 하여 기본 키와 UNIQUE 설정을 통해서 기본 인덱스, 보조 인덱스를 설정하였다.

우선적으로 기본 인덱스와, 보조 인덱스를 통해 요구를 하는 경우가 많음을 요구사항을 통해 분석하였다. 또한 자주 쓰이는 제조회사 CREATE INDEX ManufactureIndex ON MEDICINENAME (MANUFACTURER); 를 통해 인덱스 선정하였고, 수술 명도 직원이 자주 검색하여 CREATE INDEX SurgeonNameindex ON PATIENT (SURGEONNAME);를 통해서 인덱스를 선정했다.

또한 복잡한 쿼리 문을 간단하게 하기 위해 뷰의 기능을 사용하여 간단하게 하였다.

예) SELECT EMPNAME

FROM EMPLOYEEID

WHERE EMPIDNO1 IN

(SELECT EMPIDNO

FROM EMPLOYEE

WHERE EMPNO IN

(SELECT PATIENT.ENO

FROM PATIENT,PAT\_DISEASE

WHERE PATIENT.PATNO=PAT\_DISEASE.DESPATNO

AND PAT\_DISEASE.DESEASENAME='감기'));

SELECT EMPNAME

FROM EMPLOYEEID

WHERE EMPIDNO1 IN (SELECT \* FROM EMPNO\_DISEASE);

그리고 요구 분석에 맞게 각 부서에 해당하는 직원들의 전화번호를 검색하는 경우가 많아서 각 부서 별로 뷰를 생성했다.

여러 개 중 예1)

SELECT \* FROM EMPPHONENO\_ORTHOPEDICS; //뷰를 보여주게함.

CREATE VIEW EMPPHONENO\_ORTHOPEDICS (직원번호,정형외과직원번호,부서이름)

AS SELECT E.EMPNO,E.EMPPHONENO, D.DEPTNAME

FROM EMPLOYEE E,DEPARTMENT D

WHERE E.DNO=D.DEPTNO AND D.DEPTNAME='정형외과';

부서별로 직원 번호와, 직원 전화번호, 부서 이름 까지 한번에 검색된다.

그리고 직원이 환자와 직원의 정보를 검색 할 경우 주민번호를 제외하고 검색되게 하기 위해서 뷰를 생성하였다.

예2)

CREATE VIEW EMPLOYEEINFORMATION

AS SELECT EMPNO, EMPPHONENO, TITLE, DNO

FROM EMPLOYEE;

SELECT \* FROM EMPLOYEEINFORMATION;

뷰의 보안기능을 이용하여 주민번호를 제외하고 검색하게끔 뷰를 형성했다.

직원의 요구사항과 부합하고 매핑 되도록 기능 구현을 하였다. 제일 많이 쓰이는 필수적 질의들을 뽑아 기능 구현을 했다.

1.해당 직원들의 급여를 통계내기위해 검색하는 일이 잦다.(뷰 생성)

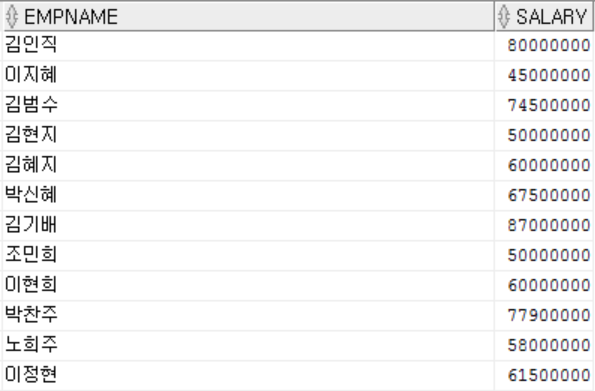
SELECT \* FROM EMPNAME\_SALARY\_INFO;

CREATE VIEW EMPNAME\_SALARY\_INFO

AS SELECT EMPNAME,SALARY

FROM EMPLOYEE,EMPLOYEEID

WHERE EMPLOYEE.EMPIDNO=EMPLOYEEID.EMPIDNO1;

 <= 해당 검색 결과

2. 직원들은 부서별로 평균 급여와 최대 급여가 궁금해졌다.(O)

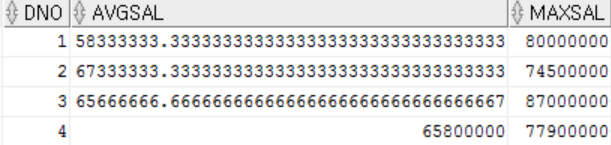
SELECT DNO, AVG(SALARY) AS AVGSAL, MAX(SALARY) AS MAXSAL

FROM EMPLOYEE, EMPLOYEEID

WHERE EMPLOYEE.EMPIDNO=EMPLOYEEID.EMPIDNO1

GROUP BY DNO

ORDER BY DNO ASC;

<= 해당 검색 결과

3. 직원은 1003호실에 문제가 발생했다. 직원은 1003호실 관리 직원에게 전화를 해야하는 상황인데 전화번호를 모른다. 이때 해당 직원의 전화번호를 검색한다.

SELECT EMPPHONENO

FROM EMPLOYEE,ROOM

WHERE EMPLOYEE.EMPNO=ROOM.MANAGER AND ROOM.ROOMNO=1003;

< 해당 검색 결과

4. 직원들은 모든 부서, 환자, 공급자 등등 에 관한 모든 정보들을 검색한다. (O) 단 주민번호가 있을 경우 주민번호는 제외하고 검색.

CREATE VIEW EMPLOYEEINFORMATION

AS SELECT EMPNO, EMPPHONENO, TITLE, DNO

FROM EMPLOYEE;

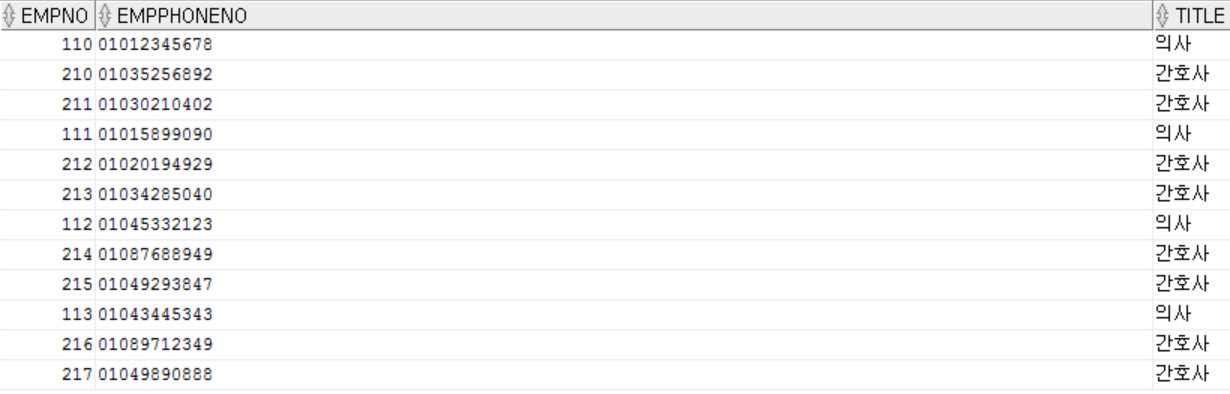
SELECT \* FROM EMPLOYEEINFORMATION;

CREATE VIEW PATIENTINFORMATION

AS SELECT PATNO,PATPHONENO,ENO,REPROT,SURGEONDATE,SURGEONNAME,RONO

FROM PATIENT; 주민번호 제외 검색을 위해 뷰 생성=> 보안기능!!

SELECT \* FROM PATIENTINFORMATION;



5. 타이레놀이 떨어졌다. 타이레놀의 약품 번호를 통해서 그것을 공급하는 공급자의 전화번호를 검색하시오.(O)

SELECT SUPPHONENO

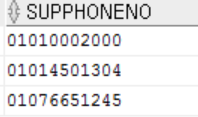
FROM SUPPLIER

WHERE SUPPNO IN

(SELECT SUPPNO1

FROM SUPPLY

WHERE MEDNO2=101);

<해당 공급자의 전화번호 검색결과

6. “\_\_\_” 환자가 임종직전이다 즉 급한 일이 생겨 환자 보호자에게 연락 할 일이 생겼다. 그 환자의 보호자 연락처와 이름을 검색 하시오.(O) <= 중요

SELECT PROPHONENO,PROTNAME

FROM PROTECTOR

WHERE PATNO1=10;



<=환자 보호자의 전화번호 검색결과

8. “신장암” 수술은 나이가 고령일수록 ( 고령 기준 70년생 밑으로 ) 수술 성공률이 희박하다.”\_\_\_\_” 수술이 잡힌 환자들에게 주의 할 부분을 알려주기 위해 환자 전화번호를 검색해라.

SELECT PATPHONENO

FROM PATIENT

WHERE SUBSTR(PATIDNO,1,2)<70 AND SURGEONNAME='신장암수술';

 <-환자 전화번호 검색결과(신장암수술)

9. 감기가 창궐해서 감기 관련해서 예방교육을 다시 직원들에게 교육을 시켜야 한다. 감기 질병을 지닌 환자의 담당 직원 이름을 검색 하시오.(O)

SELECT EMPNAME

FROM EMPLOYEEID

WHERE EMPIDNO1 IN

(SELECT EMPIDNO

FROM EMPLOYEE

WHERE EMPNO IN

(SELECT PATIENT.ENO

FROM PATIENT,PAT\_DISEASE

WHERE PATIENT.PATNO=PAT\_DISEASE.DESPATNO

AND PAT\_DISEASE.DESEASENAME='감기'));

뷰 생성

CREATE VIEW EMPNO\_DISEASE

AS SELECT EMPIDNO

FROM EMPLOYEE

WHERE EMPNO IN

(SELECT PATIENT.ENO

FROM PATIENT,PAT\_DISEASE

WHERE PATIENT.PATNO=PAT\_DISEASE.DESPATNO

AND PAT\_DISEASE.DESEASENAME='감기');

* 굉장히 복잡하기 때문에 뷰를 사용하여 간단하게 하였음.

SELECT EMPNAME

FROM EMPLOYEEID

WHERE EMPIDNO1 IN (SELECT \* FROM EMPNO\_DISEASE);



11. 타이레놀을 공급하는 약품회사가 부도났다. 당장 쓸 수 있는 타이레놀의 재고량을 검색하시오

SELECT SUM(QUANTITY)

FROM MEDICINE,SUPPLY

WHERE MEDICINE.MEDNO=SUPPLY.MEDNO2

AND MEDICINE.MEDNAME='타이레놀';



13.-1 지진 피해로 인해 외상을 입은 환자들이 많아진지 한달 후, 현재 상황에 대해서 회의 통보를 문자로 전송하기 위해 각 외과 직원들의 전화번호를 검색하시오.

SELECT \* FROM EMPPHONENO\_ORTHOPEDICS;

SELECT \* FROM EMPPHONENO\_NEUROSURGERY;

* 생성된 뷰 활용

OR

SELECT EMPPHONENO,EMPNO

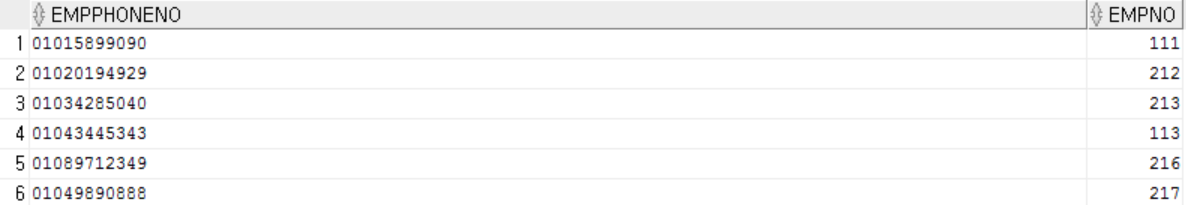
FROM EMPLOYEE

WHERE DNO IN

(SELECT DEPTNO

FROM DEPARTMENT

WHERE DEPTNAME='정형외과' OR DEPTNAME='신경외과');



13-2. 소아과에서 근무하는 직원의 연락처가 궁금해졌다. 소아과에 근무하는 직원들의 전화번호를 검색한다. 각각 1, 2, 3, 4 부서에 속하는 직원의 정보를 검색( 자기 자신 말고 다른 사람(직원) 정보가 궁금할 때) (O) = 각 부서별로 전화번호 뷰로 생성하여 간편 검색

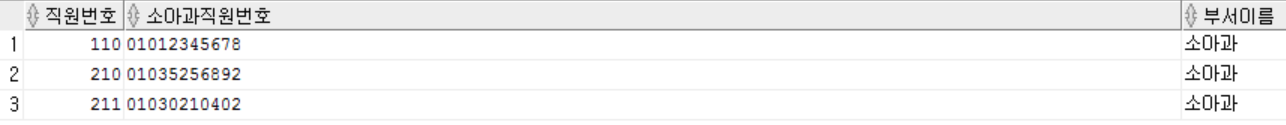
CREATE VIEW EMPPHONENO\_PEDIATRICS (직원번호,소아과직원번호,부서이름)

AS SELECT E.EMPNO,E.EMPPHONENO, D.DEPTNAME

FROM EMPLOYEE E,DEPARTMENT D

WHERE E.DNO=D.DEPTNO AND D.DEPTNAME='소아과';

* SELECT \* FROM EMPPHONENO\_PEDIATRICS;



14. 직원은 환자의 이름으로 수술 날짜와 수술 명 입실 할 병실이 궁금해졌다. (환자의 번호가 ‘ ‘ 인 환자의 수술에 대한 정보들을 검색한다.) (O) <= 보호자나 환자 관련 인원이 와서 환자 병실을 물어 볼 때 직원이 검색을 위해서 필요하다.

SELECT SURGEONDATE,SURGEONNAME,RONO

FROM PATIENT A, PATIENTIDNO B

WHERE A.PATIDNO=B.PATIDNO1 AND (PATNAME='김정훈');



16. 직원은 약품을 관리하는 도중 모 회사에서 만드는 약품이 궁금하여 모회사가 만든 약품을 검색한다. “ “회사에서 만든 약품을 검색한다.

SELECT MEDNAME1

FROM MEDICINENAME

WHERE MANUFACTURER='대웅제약'; =>인덱스생성

CREATE INDEX ManufactureIndex ON MEDICINENAME (MANUFACTURER);



17. 직원이 환자 정보를 써놓는 도중 간단한 주소가 궁금해져서 환자의 정보 중 주소를 검색한다.

단, 그때 환자의 이름이 있어야한다. 이름이 중복일 경우 주민번호로 조건으로 한다.

SELECT CITY,GU,DONG

FROM PATIENTIDNO

WHERE PATNAME='김예지';



18 .직원은 갑작스럽게 \_\_\_환자의 쇼크현상으로 인해 그 환자가 입원한 병실의 담당직원을 호출 하기 위해 번호가 궁금해졌다.

SELECT EMPPHONENO

FROM EMPLOYEE,PATIENT,PATIENTIDNO

WHERE EMPLOYEE.EMPNO=PATIENT.ENO

AND PATIENT.PATIDNO=PATIENTIDNO.PATIDNO1

AND PATIENTIDNO.PATNAME='이정수';



19. 직원은 수술 일자가 \_\_\_인 환자의 보호자에게 동의여부를 묻기 위해 보호자 전화번호가 필요하다.

SELECT PROPHONENO

FROM PROTECTOR,PATIENT

WHERE PROTECTOR.PATNO1=PATIENT.PATNO AND SURGEONDATE=20171225;

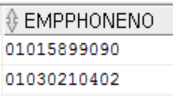


20. 해당 약품이 거의 떨어져서 이 약품을 관리하는 직원의 전화번호가 궁금해졌다. 해당 직원의 전화번호를 검색한다.

SELECT EMPPHONENO

FROM EMPLOYEE,SUPERVISE

WHERE EMPLOYEE.EMPNO=SUPERVISE.EMPNO1 AND MEDNO1=104;



이를 통해 요구사항과 비즈니스 기능이 매핑 되도록 구현함.

최종 질의

SELECT \* FROM DEPARTMENT;

SELECT \* FROM DEPARTMENTNAME;

SELECT \* FROM MEDICINE;

SELECT \* FROM MEDICINENAME;

SELECT \* FROM PAT\_DISEASE;

SELECT \* FROM PROTECTOR;

SELECT \* FROM ROOM;

SELECT \* FROM SUPERVISE;

SELECT \* FROM SUPPLIER;

SELECT \* FROM SUPPLY;

SELECT \* FROM TREAT;

SELECT \* FROM EMPNAME\_SALARY\_INFO;

COMMIT;

CREATE VIEW EMPNAME\_SALARY\_INFO

AS SELECT EMPNAME,SALARY

FROM EMPLOYEE,EMPLOYEEID

WHERE EMPLOYEE.EMPIDNO=EMPLOYEEID.EMPIDNO1;

SELECT DNO, AVG(SALARY) AS AVGSAL, MAX(SALARY) AS MAXSAL

FROM EMPLOYEE, EMPLOYEEID

WHERE EMPLOYEE.EMPIDNO=EMPLOYEEID.EMPIDNO1

GROUP BY DNO

ORDER BY DNO ASC;

SELECT EMPPHONENO

FROM EMPLOYEE,ROOM

WHERE EMPLOYEE.EMPNO=ROOM.MANAGER AND ROOM.ROOMNO=1003;

SELECT \* FROM EMPLOYEEINFORMATION;

CREATE VIEW EMPLOYEEINFORMATION

AS SELECT EMPNO, EMPPHONENO, TITLE, DNO

FROM EMPLOYEE;

SELECT \* FROM PATIENTINFORMATION;

CREATE VIEW PATIENTINFORMATION

AS SELECT PATNO,PATPHONENO,ENO,REPROT,SURGEONDATE,SURGEONNAME,RONO

FROM PATIENT;

SELECT SUPPHONENO

FROM SUPPLIER

WHERE SUPPNO IN

(SELECT SUPPNO1

FROM SUPPLY

WHERE MEDNO2=101);

SELECT PROPHONENO,PROTNAME

FROM PROTECTOR

WHERE PATNO1=10;

SELECT PATPHONENO

FROM PATIENT

WHERE SUBSTR(PATIDNO,1,2)<70 AND SURGEONNAME='신장암수술';

SELECT EMPNAME

FROM EMPLOYEEID

WHERE EMPIDNO1 IN

(SELECT EMPIDNO

FROM EMPLOYEE

WHERE EMPNO IN

(SELECT PATIENT.ENO

FROM PATIENT,PAT\_DISEASE

WHERE PATIENT.PATNO=PAT\_DISEASE.DESPATNO

AND PAT\_DISEASE.DESEASENAME='감기'));

CREATE VIEW EMPNO\_DISEASE

AS SELECT EMPIDNO

FROM EMPLOYEE

WHERE EMPNO IN

(SELECT PATIENT.ENO

FROM PATIENT,PAT\_DISEASE

WHERE PATIENT.PATNO=PAT\_DISEASE.DESPATNO

AND PAT\_DISEASE.DESEASENAME='감기');

SELECT EMPNAME

FROM EMPLOYEEID

WHERE EMPIDNO1 IN (SELECT \* FROM EMPNO\_DISEASE);

SELECT SUM(QUANTITY)

FROM MEDICINE,SUPPLY

WHERE MEDICINE.MEDNO=SUPPLY.MEDNO2

AND MEDICINE.MEDNAME='타이레놀';

SELECT \* FROM EMPPHONENO\_ORTHOPEDICS;

SELECT \* FROM EMPPHONENO\_NEUROSURGERY;

SELECT EMPPHONENO,EMPNO

FROM EMPLOYEE

WHERE DNO IN

(SELECT DEPTNO

FROM DEPARTMENT

WHERE DEPTNAME='정형외과' OR DEPTNAME='신경외과');

SELECT \* FROM EMPPHONENO\_PEDIATRICS;

CREATE VIEW EMPPHONENO\_PEDIATRICS (직원번호,소아과직원번호,부서이름)

AS SELECT E.EMPNO,E.EMPPHONENO, D.DEPTNAME

FROM EMPLOYEE E,DEPARTMENT D

WHERE E.DNO=D.DEPTNO AND D.DEPTNAME='소아과';

SELECT \* FROM EMPPHONENO\_ORTHOPEDICS;

CREATE VIEW EMPPHONENO\_ORTHOPEDICS (직원번호,정형외과직원번호,부서이름)

AS SELECT E.EMPNO,E.EMPPHONENO, D.DEPTNAME

FROM EMPLOYEE E,DEPARTMENT D

WHERE E.DNO=D.DEPTNO AND D.DEPTNAME='정형외과';

SELECT \* FROM EMPPHONENO\_NEUROSURGERY;

CREATE VIEW EMPPHONENO\_NEUROSURGERY (직원번호,신경외과직원번호,부서이름)

AS SELECT E.EMPNO,E.EMPPHONENO, D.DEPTNAME

FROM EMPLOYEE E,DEPARTMENT D

WHERE E.DNO=D.DEPTNO AND D.DEPTNAME='신경외과';

SELECT \* FROM EMPPHONENO\_NEPHROLOGY;

CREATE VIEW EMPPHONENO\_NEPHROLOGY (직원번호,신장내과직원번호,부서이름)

AS SELECT E.EMPNO,E.EMPPHONENO, D.DEPTNAME

FROM EMPLOYEE E,DEPARTMENT D

WHERE E.DNO=D.DEPTNO AND D.DEPTNAME='신장내과';

SELECT SURGEONDATE,SURGEONNAME,RONO

FROM PATIENT A, PATIENTIDNO B

WHERE A.PATIDNO=B.PATIDNO1 AND (PATNAME='김정훈');

CREATE INDEX ManufactureIndex ON MEDICINENAME (MANUFACTURER);

SELECT MEDNAME1

FROM MEDICINENAME

WHERE MANUFACTURER='대웅제약';

SELECT CITY,GU,DONG

FROM PATIENTIDNO

WHERE PATNAME='김예지';

SELECT EMPPHONENO

FROM EMPLOYEE,PATIENT,PATIENTIDNO

WHERE EMPLOYEE.EMPNO=PATIENT.ENO

AND PATIENT.PATIDNO=PATIENTIDNO.PATIDNO1

AND PATIENTIDNO.PATNAME='이정수';

SELECT PROPHONENO

FROM PROTECTOR,PATIENT

WHERE PROTECTOR.PATNO1=PATIENT.PATNO AND SURGEONDATE=20171225;

SELECT EMPPHONENO

FROM EMPLOYEE,SUPERVISE

WHERE EMPLOYEE.EMPNO=SUPERVISE.EMPNO1 AND MEDNO1=104;

CREATE TABLE EMPLOYEE

(

EMPNO NUMBER NOT NULL,

EMPIDNO NUMBER NOT NULL UNIQUE,

EMPPHONENO CHAR(100) ,

TITLE CHAR(50) DEFAULT '직원',

DNO NUMBER DEFAULT 1 CHECK (DNO IN (1,2,3,4)),

PRIMARY KEY ( EMPNO ),

FOREIGN KEY ( DNO ) REFERENCES DEPARTMENT(DEPTNO)

ON DELETE CASCADE,

FOREIGN KEY (EMPIDNO) REFERENCES EMPLOYEEID(EMPIDNO1)

ON DELETE CASCADE

);

INSERT INTO EMPLOYEE VALUES(110,9412121144321,'01012345678','의사',1);

INSERT INTO EMPLOYEE VALUES(210,8604052655432,'01035256892','간호사',1);

INSERT INTO EMPLOYEE VALUES(211,8507212448392,'01030210402','간호사',1);

INSERT INTO EMPLOYEE VALUES(111,7902121132456,'01015899090','의사',2);

INSERT INTO EMPLOYEE VALUES(212,8903012654432,'01020194929','간호사',2);

INSERT INTO EMPLOYEE VALUES(213,9010072441234,'01034285040','간호사',2);

INSERT INTO EMPLOYEE VALUES(112,7612111142324,'01045332123','의사',3);

INSERT INTO EMPLOYEE VALUES(214,9103202331478,'01087688949','간호사',3);

INSERT INTO EMPLOYEE VALUES(215,8610222441234,'01049293847','간호사',3);

INSERT INTO EMPLOYEE VALUES(113,7810201055849,'01043445343','의사',4);

INSERT INTO EMPLOYEE VALUES(216,8412232112342,'01089712349','간호사',4);

INSERT INTO EMPLOYEE VALUES(217,7707102443123,'01049890888','간호사',4);

CREATE TABLE EMPLOYEEID

(

EMPIDNO1 NUMBER NOT NULL,

EMPNAME CHAR(50) NOT NULL UNIQUE,

SALARY NUMBER CHECK ( SALARY < 100000000) ,

PRIMARY KEY (EMPIDNO1)

);

INSERT INTO EMPLOYEEID VALUES(9412121144321,'김인직',80000000);

INSERT INTO EMPLOYEEID VALUES(8604052655432,'김현지',50000000);

INSERT INTO EMPLOYEEID VALUES(8507212448392,'이지혜',45000000);

INSERT INTO EMPLOYEEID VALUES(7902121132456,'김범수',74500000);

INSERT INTO EMPLOYEEID VALUES(8903012654432,'김혜지',60000000);

INSERT INTO EMPLOYEEID VALUES(9010072441234,'박신혜',67500000);

INSERT INTO EMPLOYEEID VALUES(7612111142324,'김기배',87000000);

INSERT INTO EMPLOYEEID VALUES(9103202331478,'조민희',50000000);

INSERT INTO EMPLOYEEID VALUES(8610222441234,'이현희',60000000);

INSERT INTO EMPLOYEEID VALUES(7810201055849,'박찬주',77900000);

INSERT INTO EMPLOYEEID VALUES(8412232112342,'노희주',58000000);

INSERT INTO EMPLOYEEID VALUES(7707102443123,'이정현',61500000);

CREATE TABLE DEPARTMENT

(

DEPTNO NUMBER NOT NULL CHECK (DEPTNO IN(1,2,3,4)),

DEPTNAME CHAR(100) NOT NULL UNIQUE,

PRIMARY KEY(DEPTNO),

FOREIGN KEY (DEPTNAME) REFERENCES DEPARTMENTNAME(DEPTNAME1)

);

INSERT INTO DEPARTMENT VALUES(1,'소아과');

INSERT INTO DEPARTMENT VALUES(2,'정형외과');

INSERT INTO DEPARTMENT VALUES(3,'신장내과');

INSERT INTO DEPARTMENT VALUES(4,'신경외과');

CREATE TABLE DEPARTMENTNAME

(

DEPTNAME1 CHAR (100) NOT NULL ,

FLOOR NUMBER CHECK (FLOOR <5) NOT NULL,

PRIMARY KEY (DEPTNAME1)

);

INSERT INTO DEPARTMENTNAME VALUES('소아과',1);

INSERT INTO DEPARTMENTNAME VALUES('정형외과',2);

INSERT INTO DEPARTMENTNAME VALUES('신장내과',3);

INSERT INTO DEPARTMENTNAME VALUES('신경외과',4);

CREATE TABLE MEDICINE

(

MEDNO NUMBER NOT NULL ,

MEDNAME CHAR (50) NOT NULL UNIQUE,

PRIMARY KEY (MEDNO),

FOREIGN KEY (MEDNAME) REFERENCES MEDICINENAME(MEDNAME1)

ON DELETE CASCADE

);

INSERT INTO MEDICINE VALUES(101,'타이레놀');

INSERT INTO MEDICINE VALUES(102,'아스피린');

INSERT INTO MEDICINE VALUES(103,'코딜라트정');

INSERT INTO MEDICINE VALUES(104,'뮤코스텐캡슐');

INSERT INTO MEDICINE VALUES(105,'푸리노신정');

INSERT INTO MEDICINE VALUES(106,'레일라정');

INSERT INTO MEDICINE VALUES(107,'디스텍정');

INSERT INTO MEDICINE VALUES(108,'크리페낙정');

CREATE TABLE MEDICINENAME

(

MEDNAME1 CHAR(50) NOT NULL,

MANUFACTURER CHAR(50) NOT NULL ,

PRIMARY KEY(MEDNAME1)

);

INSERT INTO MEDICINENAME VALUES('타이레놀','대웅제약');

INSERT INTO MEDICINENAME VALUES('아스피린','광동제약');

INSERT INTO MEDICINENAME VALUES('코딜라트정','코오롱제약');

INSERT INTO MEDICINENAME VALUES('뮤코스텐캡슐','메디카코리아');

INSERT INTO MEDICINENAME VALUES('푸리노신정','종근당');

INSERT INTO MEDICINENAME VALUES('레일라정','한국피엠지제약');

INSERT INTO MEDICINENAME VALUES('디스텍정','안국약품');

INSERT INTO MEDICINENAME VALUES('크리페낙정','삼진제약');

CREATE INDEX Patijumino ON PATIENT (PATIDNO);

CREATE TABLE PATIENT

(

PATNO NUMBER NOT NULL,

PATIDNO NUMBER NOT NULL ,

PATPHONENO CHAR(100) ,

ENO NUMBER NOT NULL ,

REPROT NUMBER NOT NULL ,

SURGEONDATE NUMBER ,

SURGEONNAME CHAR(50) ,

RONO NUMBER ,

PRIMARY KEY ( PATNO ),

FOREIGN KEY ( ENO ) REFERENCES EMPLOYEE(EMPNO)

ON DELETE CASCADE,

FOREIGN KEY (RONO) REFERENCES ROOM(ROOMNO)

ON DELETE CASCADE,

FOREIGN KEY (PATIDNO) REFERENCES PATIENTIDNO(PATIDNO1)

ON DELETE CASCADE

);

INSERT INTO PATIENT VALUES(10,9810101055555,'01043244444',110,2,20170902,'맹장수술',1001);

INSERT INTO PATIENT VALUES(11,9408181042342,'01014678382',111,3,20171225,'골이식',1001);

INSERT INTO PATIENT VALUES(12,9104042312242,'01032556313',110,1,'','','');

INSERT INTO PATIENT VALUES(13,8910031158222,'01065331020',111,4,20180104,'골이식',1001);

INSERT INTO PATIENT VALUES(14,8710202141232,'01039389404',110,5,'','','');

INSERT INTO PATIENT VALUES(15,9307071143242,'01023231402',112,6,20171230,'신장이식',1002);

INSERT INTO PATIENT VALUES(16,9205052213454,'01068909884',112,7,'','','');

INSERT INTO PATIENT VALUES(17,6811101050032,'01059489569',112,8,20171221,'신장암수술',1002);

INSERT INTO PATIENT VALUES(18,8710201109842,'01020001020',111,9,20180108,'척추골절수술',1002);

INSERT INTO PATIENT VALUES(19,8804212003020,'01023088482',113,10,20180204,'허리디스크수술',1003);

INSERT INTO PATIENT VALUES(20,8904011043012,'01098858383',113,11,'','','');

INSERT INTO PATIENT VALUES(21,6602042030201,'01048990103',113,12,20180110,'목디스크수술',1003);

INSERT INTO PATIENT VALUES(22,6711101044332,'01049280898',112,13,20180112,'신장결석수술',1003);

DELETE FROM PATIENT

WHERE PATNO=17;

UPDATE PATIENT

SET PATIDNO=6811101050032

WHERE PATNO=17;

CREATE INDEX SurgeonNameindex ON PATIENT (SURGEONNAME);

CREATE TABLE PATIENTIDNO

(

PATIDNO1 NUMBER NOT NULL,

PATNAME CHAR(50) NOT NULL UNIQUE,

CITY CHAR(50) NOT NULL ,

GU CHAR(50) NOT NULL ,

DONG CHAR(50) NOT NULL ,

PRIMARY KEY(PATIDNO1)

);

INSERT INTO PATIENTIDNO VALUES(9810101055555,'김정훈','부천시','원미구','상동');

INSERT INTO PATIENTIDNO VALUES(9408181042342,'이정수','성남시','분당구','정자동');

INSERT INTO PATIENTIDNO VALUES(9104042312242,'김예지','수원시','영통구','매탄동');

INSERT INTO PATIENTIDNO VALUES(8910031158222,'박현수','부천시','소사구','범박동');

INSERT INTO PATIENTIDNO VALUES(8710202141232,'조현지','안양시','동안구','비산동');

INSERT INTO PATIENTIDNO VALUES(9307071143242,'이정준','안양시','동안구','평촌동');

INSERT INTO PATIENTIDNO VALUES(9205052213454,'손예지','용인시','기흥구','산갈동');

INSERT INTO PATIENTIDNO VALUES(6811101050032,'소순원','안산시','상록구','일동');

INSERT INTO PATIENTIDNO VALUES(8710201109842,'서진원','성남시','중원구','금광동');

INSERT INTO PATIENTIDNO VALUES(8804212003020,'서현진','성남시','중원구','상대원동');

INSERT INTO PATIENTIDNO VALUES(8904011043012,'김형오','수원시','팔달구','북수동');

INSERT INTO PATIENTIDNO VALUES(6602042030201,'김복자','수원시','팔달구','인계동');

INSERT INTO PATIENTIDNO VALUES(6711101044332,'김윤홍','고양시','일산서구','일산동');

DELETE FROM PATIENTIDNO

WHERE PATNAME='소순원';

CREATE TABLE ROOM

(

ROOMNO NUMBER NOT NULL,

STARTDATE NUMBER NOT NULL,

MANAGER NUMBER NOT NULL UNIQUE,

PRIMARY KEY(ROOMNO),

FOREIGN KEY (MANAGER) REFERENCES EMPLOYEE(EMPNO)

ON DELETE CASCADE

);

INSERT INTO ROOM VALUES(1001,20170101,110);

INSERT INTO ROOM VALUES(1002,20170304,210);

INSERT INTO ROOM VALUES(1003,20170214,211);

INSERT INTO ROOM VALUES(1004,20170404,212);

CREATE TABLE SUPPLIER

(

SUPPNO NUMBER NOT NULL,

SUPPNAME CHAR(30) NOT NULL UNIQUE,

SUPPHONENO CHAR(100) ,

PRIMARY KEY (SUPPNO)

);

INSERT INTO SUPPLIER VALUES(4100,'김호준','01010002000');

INSERT INTO SUPPLIER VALUES(4101,'이재진','01014501304');

INSERT INTO SUPPLIER VALUES(4102,'방선호','01042315040');

INSERT INTO SUPPLIER VALUES(4103,'김범석','01076651245');

INSERT INTO SUPPLIER VALUES(4104,'강호진','01048289019');

INSERT INTO SUPPLIER VALUES(4105,'김복주','01048283192');

CREATE TABLE PROTECTOR

(

PATNO1 NUMBER NOT NULL,

PROTNAME CHAR(30) NOT NULL UNIQUE,

PROPHONENO CHAR(100) ,

PRIMARY KEY(PATNO1,PROTNAME),

FOREIGN KEY (PATNO1) REFERENCES PATIENT(PATNO)

ON DELETE CASCADE

);

INSERT INTO PROTECTOR VALUES(10,'김희승','01021304000');

INSERT INTO PROTECTOR VALUES(11,'이복남','01041233020');

INSERT INTO PROTECTOR VALUES(12,'박대선','01032310102');

INSERT INTO PROTECTOR VALUES(13,'박기춘','01020202010');

INSERT INTO PROTECTOR VALUES(14,'조민귀','01049772839');

INSERT INTO PROTECTOR VALUES(15,'이범진','01054021023');

INSERT INTO PROTECTOR VALUES(16,'손성락','01030202010');

INSERT INTO PROTECTOR VALUES(17,'소진권','01027372616');

INSERT INTO PROTECTOR VALUES(18,'서민석','01099615496');

INSERT INTO PROTECTOR VALUES(19,'서원혁','01021737771');

INSERT INTO PROTECTOR VALUES(20,'김관진','01048273910');

INSERT INTO PROTECTOR VALUES(21,'김부진','01010284938');

INSERT INTO PROTECTOR VALUES(22,'김해진','01030499929');

CREATE TABLE SUPERVISE

(

EMPNO1 NUMBER NOT NULL,

MEDNO1 NUMBER NOT NULL ,

PRIMARY KEY(EMPNO1, MEDNO1),

FOREIGN KEY (EMPNO1) REFERENCES EMPLOYEE(EMPNO)

ON DELETE CASCADE,

FOREIGN KEY (MEDNO1) REFERENCES MEDICINE(MEDNO)

ON DELETE CASCADE

);

DROP TABLE SUPERVISE;

DELETE FROM SUPERVISE

WHERE MEDNO1=1;

SELECT \* FROM SUPERVISE;

INSERT INTO SUPERVISE VALUES(110,101);

INSERT INTO SUPERVISE VALUES(110,102);

INSERT INTO SUPERVISE VALUES(110,103);

INSERT INTO SUPERVISE VALUES(210,101);

INSERT INTO SUPERVISE VALUES(210,102);

INSERT INTO SUPERVISE VALUES(210,103);

INSERT INTO SUPERVISE VALUES(211,104);

INSERT INTO SUPERVISE VALUES(111,104);

INSERT INTO SUPERVISE VALUES(110,105);

INSERT INTO SUPERVISE VALUES(212,105);

INSERT INTO SUPERVISE VALUES(213,106);

INSERT INTO SUPERVISE VALUES(112,107);

INSERT INTO SUPERVISE VALUES(210,108);

INSERT INTO SUPERVISE VALUES(211,108);

CREATE TABLE TREAT

(

PATNO2 NUMBER NOT NULL,

EMPNO2 NUMBER NOT NULL,

TREATEDATE NUMBER NOT NULL,

PRIMARY KEY (PATNO2,EMPNO2),

FOREIGN KEY (PATNO2) REFERENCES PATIENT(PATNO)

ON DELETE CASCADE,

FOREIGN KEY (EMPNO2) REFERENCES EMPLOYEE(EMPNO)

ON DELETE CASCADE

);

INSERT INTO TREAT VALUES(10,110,20170403);

INSERT INTO TREAT VALUES(11,111,20170404);

INSERT INTO TREAT VALUES(11,112,20170405);

INSERT INTO TREAT VALUES(12,110,20170401);

INSERT INTO TREAT VALUES(12,113,20170402);

INSERT INTO TREAT VALUES(13,111,20170602);

INSERT INTO TREAT VALUES(13,112,20170605);

INSERT INTO TREAT VALUES(14,110,20170102);

INSERT INTO TREAT VALUES(15,112,20141004);

INSERT INTO TREAT VALUES(15,110,20110302);

INSERT INTO TREAT VALUES(16,112,20130608);

INSERT INTO TREAT VALUES(16,113,20100402);

INSERT INTO TREAT VALUES(17,112,20140909);

INSERT INTO TREAT VALUES(18,111,20150421);

INSERT INTO TREAT VALUES(18,110,20160321);

INSERT INTO TREAT VALUES(19,113,20170201);

INSERT INTO TREAT VALUES(20,113,20170314);

INSERT INTO TREAT VALUES(21,113,20170704);

INSERT INTO TREAT VALUES(22,112,20151008);

CREATE TABLE SUPPLY

(

SUPPNO1 NUMBER NOT NULL,

MEDNO2 NUMBER NOT NULL ,

EMPNO3 NUMBER NOT NULL,

QUANTITY NUMBER ,

PRIMARY KEY (SUPPNO1, MEDNO2, EMPNO3),

FOREIGN KEY (SUPPNO1) REFERENCES SUPPLIER(SUPPNO)

ON DELETE CASCADE,

FOREIGN KEY (MEDNO2) REFERENCES MEDICINE(MEDNO)

ON DELETE CASCADE,

FOREIGN KEY (EMPNO3) REFERENCES EMPLOYEE(EMPNO)

ON DELETE CASCADE

);

INSERT INTO SUPPLY VALUES(4100,101,110,145);

INSERT INTO SUPPLY VALUES(4100,102,111,100);

INSERT INTO SUPPLY VALUES(4100,103,210,50);

INSERT INTO SUPPLY VALUES(4100,104,211,30);

INSERT INTO SUPPLY VALUES(4101,101,110,30);

INSERT INTO SUPPLY VALUES(4101,102,111,100);

INSERT INTO SUPPLY VALUES(4101,103,210,100);

INSERT INTO SUPPLY VALUES(4101,104,211,50);

INSERT INTO SUPPLY VALUES(4103,101,110,145);

INSERT INTO SUPPLY VALUES(4103,102,111,100);

INSERT INTO SUPPLY VALUES(4103,103,210,30);

INSERT INTO SUPPLY VALUES(4103,104,211,25);

INSERT INTO SUPPLY VALUES(4104,105,113,300);

INSERT INTO SUPPLY VALUES(4104,106,214,400);

INSERT INTO SUPPLY VALUES(4105,106,216,200);

INSERT INTO SUPPLY VALUES(4105,107,112,400);

INSERT INTO SUPPLY VALUES(4105,108,215,500);

CREATE TABLE PAT\_DISEASE

(

DESPATNO NUMBER NOT NULL,

DESEASENAME CHAR(100) ,

PRIMARY KEY(DESPATNO, DESEASENAME),

FOREIGN KEY(DESPATNO) REFERENCES PATIENT(PATNO)

ON DELETE CASCADE

);

INSERT INTO PAT\_DISEASE VALUES(10,'감기');

INSERT INTO PAT\_DISEASE VALUES(10,'맹장');

INSERT INTO PAT\_DISEASE VALUES(11,'골절');

INSERT INTO PAT\_DISEASE VALUES(12,'감기');

INSERT INTO PAT\_DISEASE VALUES(13,'골절');

INSERT INTO PAT\_DISEASE VALUES(13,'고혈압');

INSERT INTO PAT\_DISEASE VALUES(14,'감기');

INSERT INTO PAT\_DISEASE VALUES(14,'장염');

INSERT INTO PAT\_DISEASE VALUES(15,'신장암');

INSERT INTO PAT\_DISEASE VALUES(15,'신장염');

INSERT INTO PAT\_DISEASE VALUES(15,'고혈압');

INSERT INTO PAT\_DISEASE VALUES(16,'고혈압');

INSERT INTO PAT\_DISEASE VALUES(17,'신장암');

INSERT INTO PAT\_DISEASE VALUES(17,'신장결석');

INSERT INTO PAT\_DISEASE VALUES(18,'골절');

INSERT INTO PAT\_DISEASE VALUES(18,'당뇨병');

INSERT INTO PAT\_DISEASE VALUES(19,'허리디시크');

INSERT INTO PAT\_DISEASE VALUES(20,'당뇨병');

INSERT INTO PAT\_DISEASE VALUES(21,'목디스크');

INSERT INTO PAT\_DISEASE VALUES(22,'신장결석');