

아주대학교 권역응급외상센터 데이터베이스 설계 팀프로젝트

최종 설계 문서

학생 이름:

장해식(팀장),김두현,이관용,허민

교수:

허훈식

강의:

데이터베이스(02)

목차

서론	3
서비스 소개 및 선정 이유	3
본론	3
개념적설계	3
논리적설계	18
정규화	27
물리적설계	32
트렌젝션	37
결론	38

서론

서비스 소개 및 선정 이유

권역응급의료센터는 중증응급환자를 24시간 전문적이고 체계적으로 치료하기 위하여, 정부가 국가적 차원에서 전국의 도·시·군을 41개 권역으로 나누어, 권역내 응급의료 및 재해재난 등의 중추적인 역할을 할 수 있는 병원 1개소를 지정·운영된다.

권역응급의료센터는 중증응급환자는 물론 권역내 응급의료기관·보건소·소방서 등과의 유기적 네트워크운동을 주도하여 응급환자 치료에 필요한 정보 제공 및 교육 그리고 권역내 발생되어지는 재해 중심병원으로서의 역할을 수행하게 된다.

해당팀원 모두 다큐멘터리를 좋아하는 공통의 관심사가 있어 EBS 다큐멘터리 ‘명의’를 통해 권역응급외상센터를 알고 있었다. 이번 프로젝트를 통해 열악한 의료시스템에 관심을 가져보고 사회안전망을 구성하는 데 도움이 되고자하여 해당 주제를 선정하였다.

본론

개념적설계

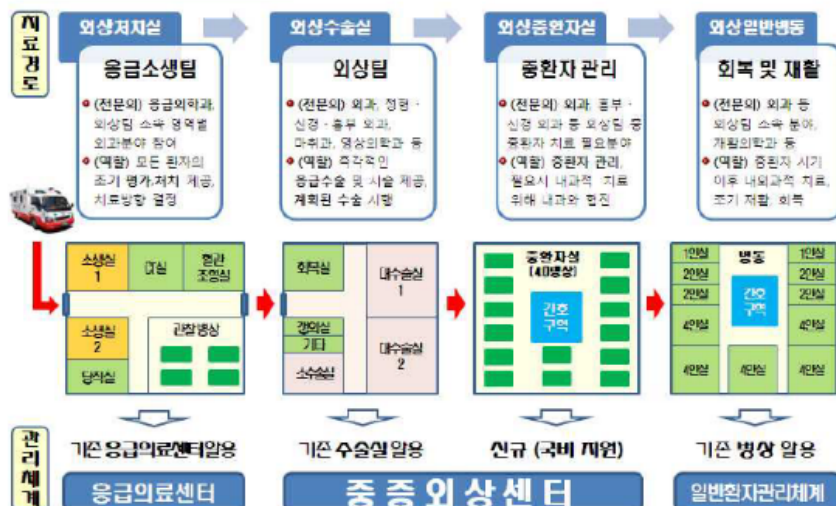
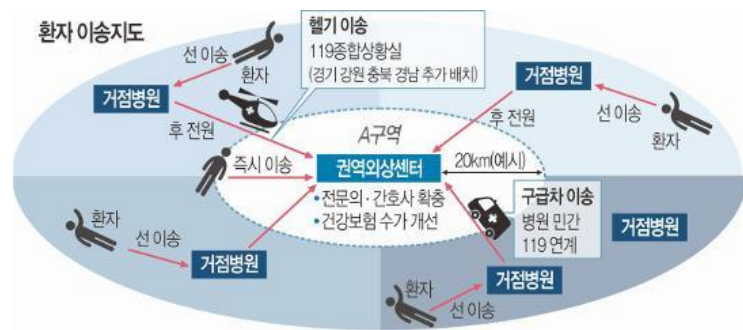
개념적설계는 서비스 소개 및 선정이유, 요구사항 수집 및 업무 분석, 데이터 요구사항 수집, 개념적 모델링의 과정에 따라 설계하였다.

사용자의 범위 및 업무분석은 다큐멘터리, 강연과 기존 문서를 참고하였다. 다큐멘터리의 경우, “EBS 명의 다큐멘터리 – 골든타임, 운명의 1시간” 편, 강연은 “세상을 바꾸는 시간 15분 797회, 외상외과 전문의 이국종 교수”편을 참고하였다.

기존문서는 아래와 같은 문서를 참고하였다.

- 보건복지부 -권역외상센터 표준운영체제
- 보건복지부 – 권역외상센터 설치지원 사업안내
- 보건복지부 – 중증응급환자 질환별 진료가능병원 실시간 정보제공 서비스
- 보건복지부 중앙응급의료센터 – 외상등록체계 구축 및 입력 지침서

권역외상센터의 업무흐름도를 이해하여 요구사항 수 및 업무 분석을 위해 중증 이송체계도, 응급환자 표준진료정보 및 외상환자 진료정보 전송 프로세스, 응급환자 표준진료정보 및 외상환자 진료정보 전송 프로세스를 참고하였다.



그렇게 도착한 환자는 내원과 진료를 통해 즉각적인 응급 수술 및 시술을 받게 된다. 그 후 중환자실에 입원하여 관리 및 필요시 내과적 치료와 내과와 중환자실이나 응급실과 외상센터의 차이점은 어떤 환자를 다루지만 아울러 권역병원(권역응급실)에서 치료하지 못하는 중증외상환자가 중심이 된다는 것이 차이점이다.

개체와 속성과 관련된 요구사항 명세

1. 모든 환자는 환자 ID(고유함)를 부여하여 식별한다.
2. 각 환자에 대해서 이메일, 주소, 전화번호, 성명, 주민번호, 성별, 전화번호, 이메일, 직업, 지병, 주민번호를 통해 나이 유추할 수 있도록 한다.
3. 이때 바이탈사인은 체온, 맥박, 혈압, 호흡으로 세분화한다.
4. 한 환자가 다양한 지병을 가지고 있을 수도 있다.
5. 환자의 보호자는 보호자 이름, 전화번호, 성별, 생년월일을 가진다.
6. 보호자는 자체적으로 기본 키를 갖고 있지 않으므로 약한 엔티티 타입이다. 보호자 이름이 부분키가 된다
7. 모든 의사는 의사 ID(고유함)를 부여하여 식별한다.
8. 담당과, 성명, 성별, 전화번호, 이메일, 직급을 가진다.
9. 차트는 차트ID(고유함)을 통해 구분이 가능해야 한다. 각 차트는 환자에 관한 의사소견을 기록할 수 있어야 한다.
10. 모든 간호사는 간호사 ID(고유함)를 부여하여 식별한다.
11. 각 간호사는 부서, 성명, 성별, 전화번호, 이메일, 직급을 가진다.
12. 모든 코디네이터는 코디네이터 ID(고유함)을 통해 식별한다.
13. 각 코디네이터는 부서, 이메일, 전화번호, 성별, 이름을 가진다.
14. 병실은 병실번호(고유함)을 통해 식별할 수 있고 총, 최대 수용인원, 현재 수용인원을 가진다.
15. 병실의 병상이 전부 차있는지 비어있는지에 대한 상태를 알 수 있어야 한다.
16. 수술실은 수술실 번호(고유함)을 통해 식별할 수 있어야 한다. 수술실에서 수술이 진행 중인지에 대한 상태를 알 수 있어야 한다.
17. 검사실은 검사실 이름(고유함)을 통해 식별할 수 있어야 한다. 검사실에서 검사가 진행되는 중 인지에 대한 상태를 알 수 있어야 한다.
18. 모든 Driver는 Driver_ID(고유함)를 부여하여 식별한다.
19. 각 Driver는 이름, 성별, 직급, 전화번호, 이메일을 가져야 한다.
20. 각 Driver는 닥터헬기를 운행하는지 앰불런스를 운전하는지 알 수 있도록 역할을 알 수 있어야 한다.
21. 의사의 전공을 전공번호(고유함)을 통해 식별할 수 있다. 각 전공은 전공 번호를 통해 전공명을 알 수 있다.
22. 이동수단은 이동수단 번호(고유함)을 통해 식별할 수 있어야 한다. 각 이동수단은 종류를 가지고 있다.
23. 부서는 부서번호(고유함)을 통해 식별할 수 있다. 각 부서는 부서명을 가지고 있다.

관계와 관련된 요구사항 명세

1. 환자는 한명 이상의 담당 간호사를 갖는다. 간호사는 한명 이상의 환자를 담당할 수 있다.
2. 병실에는 환자가 입원하지 않거나 여러명이 입원하고, 환자는 입원하지 않거나 1개의 병실에 입원한다.
3. 환자는 수술을 받지 않거나 수술실에서 수술을 받을 수 있다. 수술실은 비어있거나 한명의 환자를 수술할 수 있다.
4. 환자는 검사를 받지 않거나 1개 이상의 검사실에서 검사 받을 수 있고, 각 검사실엔 환자가 없을 수도 한명의 환자가 검사 받을 수 있다.
5. 환자는 보호자가 없거나 많을 수도 있다. 하지만 보호자는 1명 이상의 환자를 가질 수 있다.
6. 간호사는 한명 이상의 보호자에게 연락할 수 있고, 보호자는 한명의 간호사와 연락 되어야 한다.
7. 환자는 한 명의 담당 의사를 갖는다. 의사는 1명 이상의 환자를 담당한다.
8. 의사는 한명 이상의 환자를 진료볼 수 있다. 환자는 한명 이상의 의사에게 진료를 받는다.
9. 환자는 한 개의 차트에 기록되어야하고, 차트엔 한명의 환자만 기록되어야 한다
10. 모든 의사는 병실로 회진을 간다.
11. 코디네이터, 간호사, 의사는 서로에게 연락이 가능하고, 1명 이상에게 연락한다.
12. 각 의사는 수술을 하지 않거나 할 수 있다. 한 수술실에는 여러명의 의사가 수술 할 수 있어야 한다.
13. 의사는 무조건 1개의 전공을 가진다. 각 전공에는 1명 이상의 의사가 있다.
14. 간호사는 무조건 1개의 부서를 속한다. 각 부서에는 1명 이상의 간호사가 속해있다.
15. 모든 간호사는 한 개 이상의 병실을 담당한다.
16. 병실은 담당하는 간호사가 한명이상 있어야한다.
17. 각 코디네이터는 1개 이상의 병실,수술실,검사실을 조회가능하고,
18. 각 병실, 수술실, 검사실은 한명 이상의 코디네이터에 의해 조회된다.
19. 드라이버는 한명의 코디네이터에게 연락을 받는다. 코디네이터는 한명 이상의 드라이버에게 연락한다.
20. 한명 이상의 드라이버는 한 개 이상의 이동수당을 배정받는다.

이와 같은 요구사항 명세를 분석하여 엔터티 타입들을 식별하고 관계를 추출하였다.

엔터티 타입 및 애트리뷰트들을 식별

환자 엔터티 타입에는 주소,성명,주민번호,성별,전화번호,이메일,환자,직업,바이탈 사인,지병 등의 애트리뷰트가 필요하다. 이때 환자 ID가 기본키 이므로 ER 다이어그램에서 밑줄을 그어 표시한다. 지병은 다치-애트리뷰트이므로 이중선 타원으로 표시한다. 그림 4는 환자 엔터티 타입의 다이어그램을 보여준다.

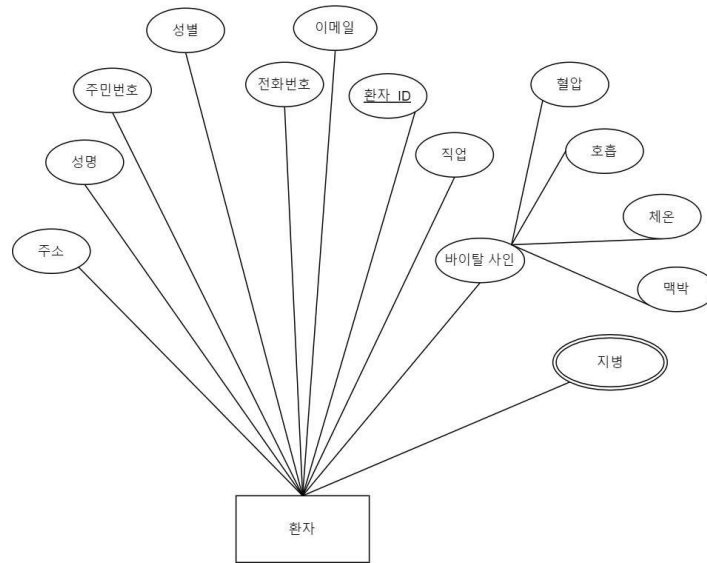


그림 4 환자 엔티티 타입

보호자 엔티티 타입은 약한 엔티티 타입이므로 이중선 직사각형으로 나타낸다. 이름, 성별, 생년월일, 전화번호 등의 애트리뷰트를 가지고 이때 보호자 이름은 부분키이므로 점선으로 표시한다.

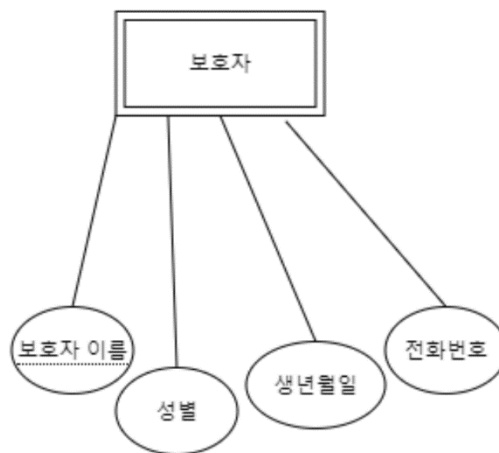


그림 5 보호자 엔티티 타입

의사 엔티티 타입에는 의사 ID, 담당과, 성별, 성명, 전화번호, 이메일, 직급 등의 애트리뷰트가 필요하다. 이때 의사_ID가 기본키이므로 ER 다이어그램에서 밑줄을 그어 표시한다.

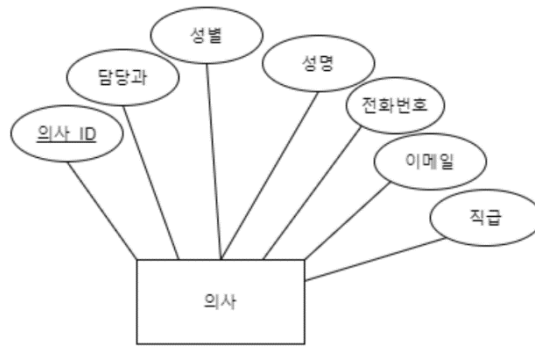


그림 6 의사 엔터티 타입

차트 엔터티 타입에는 차트 ID, 의사소견 등의 애트리뷰트가 필요하다. 이때 차트_ID가 기본키이므로 ER 다이어그램에서 밑줄을 그어 표시한다.

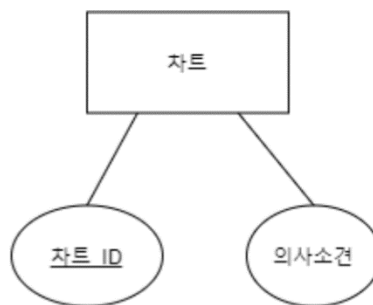


그림 7 차트 엔터티 타입

간호사엔터티 타입에는 간호사 ID, 부서, 성명, 전화번호, 성별, 이메일(EMAIL), 직급(TITLE) 등의 애트리뷰트가 필요하다. 이때 간호사_ID(NUR_ID)가 기본키이므로 ER 다이어그램에서 밑줄을 그어 표시한다.

그림 8 간호사 엔터티 타입

코디네이터 ID엔터티 타입에는 이름, 성별, 전화번호, 이메일, 부서 등의 애트리뷰트가 필요하다. 이때 코디네이터 ID가 기본키이므로 ER 다이어그램에서 밑줄을 그어 표시한다.

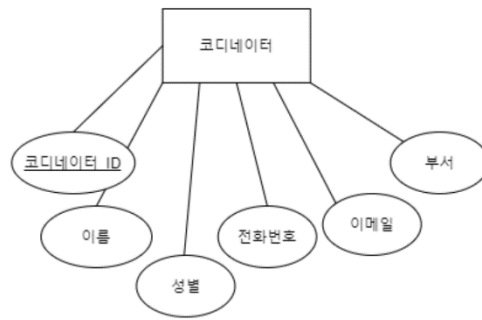


그림 9 코디네이터 엔티티 타입

병실 엔티티 타입에는 병실 번호, 층, 최대 수용 인원, 현재 수용 인원 등의 애트리뷰트가 필요하다. 이때 병실 번호가 기본키이므로 ER 다이어그램에서 밑줄을 그어 표시한다.

수술실 엔티티 타입에는 수술실 번호 하나의 애트리뷰트를 가지고 있다. 이때 수술실 번호를 기본키로 가진다.

검사실 엔티티 타입은 검사실 이름과 층을 애트리뷰트로 가진다. 이때 수술실 번호를 기본키로 가진다.

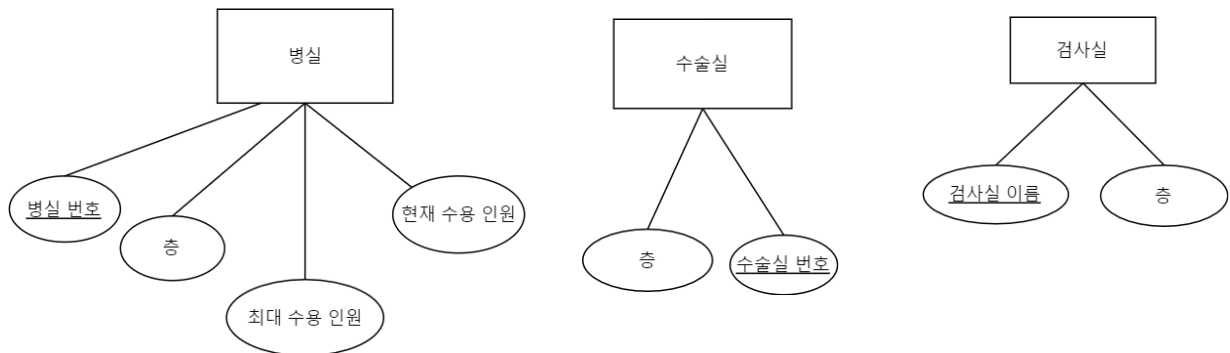


그림 10 병실, 수술실, 검사실 엔티티 타입

Driver 엔티티 타입에는 DRIVER ID, 이름, 성별, 전화번호, 이메일, 역할 등을 애트리뷰트로 가진다. 이때 DRIVER_ID를 기본키로 가진다.

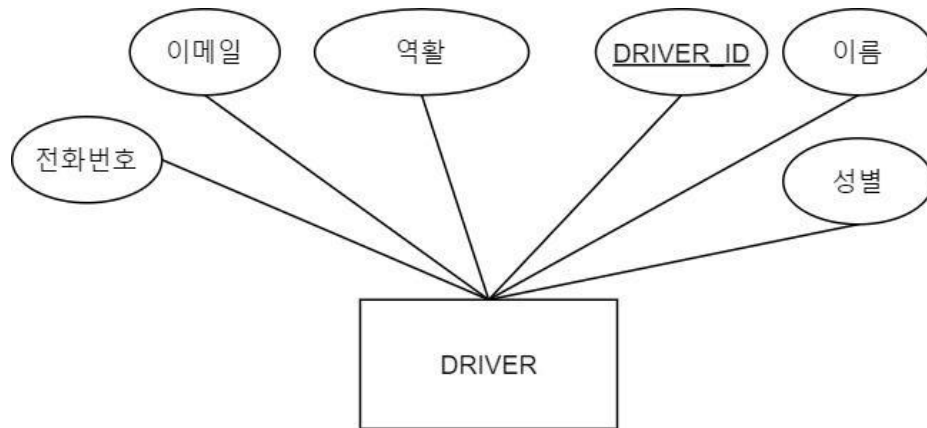


그림 11 Driver 엔티티 타입

전공 엔티티 타입에는 전공번호, 전공명을 애트리뷰트로 가진다. 이때 DRIVER_ID를 기본키로 가진다.

이동수단 엔티티 타입에는 이동수단번호와 종류 애트리뷰트를 가진다. 이때 이동수단번호를 기본키로 가진다.

부서 엔티티 타입에는 부서번호, 부서명을 애트리뷰트로 가진다. 이때 부서번호를 기본키로 가진다.

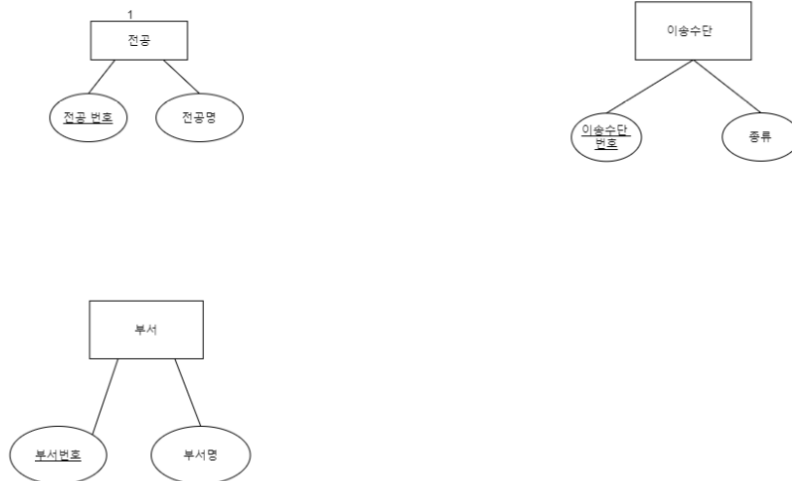


그림 12전공, 이동수단, 부서 엔티티 타입

모든 환자는 센터 내원시 담당 간호사가 꼭 존재해야하고, 모든 간호사는 한명 이상의 환자를 담당하여야 하므로 담당하다 관계에 대해 전체참여를 한다.

환자는 병실에 입원하거나 하지 않을 수 있다. 그리고 병실은 비어있거나 여러명의 환자가 입원할 수 있다. 입원하다 관계타입은 입원 날짜와 퇴원 날짜 애트리뷰트가 필요하다.

환자는 수술실에서 수술을 받거나 받지 않을 수 있다. 수술실 또한 비어있거나 사용할 수 있다. 수술받다 관계 타입은 수술날짜, 수술시간 관계 애트리뷰트가 필요하다.

환자는 검사를 받지 않거나 여러 개의 검사실에서 검사를 받을 수 있다. 하지만 검사실에는 환자가 최소 0명, 최대 1명만 들어갈 수 있다. 검사받다 관계 타입에는 검사날짜, 검사시간 애트리뷰트를 필요로 한다.

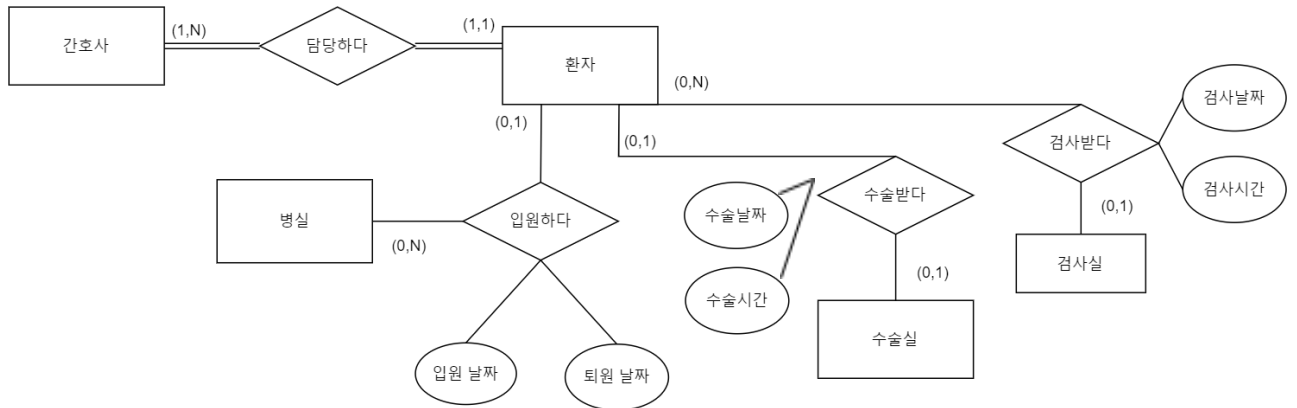


그림 13 환자와 관련된 관계 타입

환자 엔티티와 보호자 엔티티는 부양하다 관계 타입으로 연결된다. 환자는 0명 이상의 보호자를 가질 수 있고, 각 보호자는 한명의 보호자와 연결된다. 또한 보호자 엔티티는 그 엔티티와 연관된 사원 엔티티가 존재하지 않으면 존재할 수 없으므로 또는 보호자 엔티티는 자체적으로 기본키를 갖고 있지 않으므로 약한 엔티티 타입이다. 약한 엔티티 타입은 관계 타입에 전체 참여한다. 이때의 관계 타입은 이중선 다이아몬드로 표시한다.

간호사 엔티티와 보호자 엔티티는 연락하다 관계타입으로 연결된다. 간호사는 1명 이상의 보호자와 연락할 수 있고, 각 모든 보호자는 1명 이상의 간호사와 연락 가능하다.

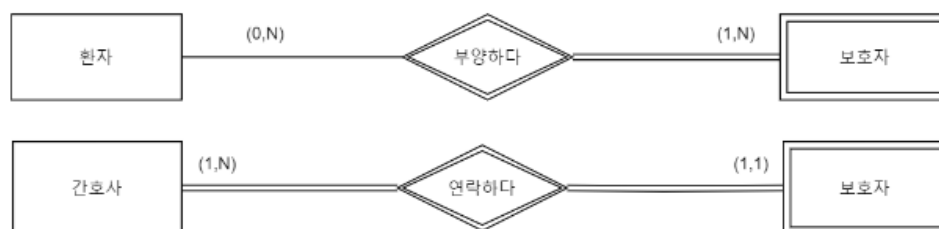


그림 14 보호자와 관련된 관계타입

차트는 차트 ID와 의사소견 애트리뷰트를 가지고 있고 차트 ID를 기본키로 가진다.

의사 엔티티 타입과 차트 엔티티 타입에는 기록하다라는 관계타입을 가진다. 한명 이상의 의사는 차트에 기록할 수 있고, 모든 각 차트는 의사에게 기록된다. 이때 차트와 의사는 전체 참여이다.

차트 엔티티와 환자 엔티티는 기록받다라는 관계타입을 가진다. 모든 차트와 환자는 1:1 관계로서 일대일 대응이 되어 전체참여가 일어난다.

의사 엔티티와 환자 엔티티는 진료하다 라는 관계 타입을 가진다. 한명 이상의 모든 의사는 환자에 대해 진료를 볼 수 있고, 모든 환자는 의사에게 진료를 받을 수 있어야 하므로 환자 엔티티 타입 진료하다 관계 타입에 대해 전체 참여한다. 모든 의사 또한 환자를 진료해야 되기 때문에 진료하다 관계 타입에 대하여 전체 참여한다.

의사 엔티티와 환자 엔티티는 담당하다라는 관계 타입을 가진다. 한명 이상의 모든 의사는 환자를 담당해야 하고, 모든 환자는 최소 한명의 담당 의사가 있어야 한다. 모든 의사는 1명 이상의 환자를 담당해야 되기 때문에 담당하다 관계 타입에 전체 참여하고, 모든 환자는 최소 1명의 담당의사가 있어야 되기 때문에 담당하다 관계 타입에 전체 참여한다.

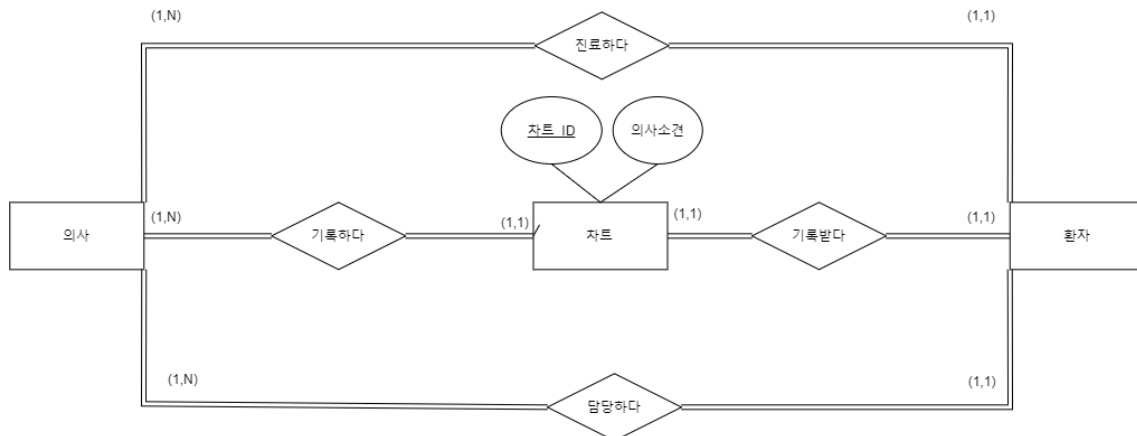


그림 15 의사와 차트, 차트와 환자, 의사와 환자 사이의 관계 타입

병실 엔티티 타입과 의사 엔티티 타입은 회진하다 관계타입을 가진다. 병실은 비어있을 수도 있기 때문에 최소 0, 최대 N번 의사가 회진할 수 있다. 모든 의사는 담당 환자가 존재 하기 때문에 1번 이상 회진을 돌아야 한다. 이때 의사 엔티티는 회진하다 관계 타입에 전체 참여한다.

의사 엔티티 타입과 전공 엔티티 타입은 가지다 라는 관계타입을 가진다. 모든 의사는 초최소 1개의 전공을 가져야 하고, 전공 최소 1명, 최대 N명의 의사가 포함되어야 한다. 이때 전공과 의사 엔티티 모두 가지다 관계 타입에 전체참여한다.

의사 엔티티 타입과 수술 엔티티 타입은 수술하다 관계타입을 가진다. 의사는 수술을 하지 않거나 하기 때문에 최소 0, 최대 1 관계를 가진다. 수술실엔 수술을 하지 않는 경우 의사가 없고, 수술시 최대 N명의 의사가 수술하기 때문에 최소 0, 최대 N의 관계를 가진다.

의사 엔티티 타입과 코디네이터 엔티티 타입 그리고 간호사 엔티티 타입은 연락하다 관계 타입을 가진다. 이 관계 타입에 참여하는 세 엔티티 타입은 1:M:N의 카디널리티를 갖는다. 모든 의사, 코디네이터, 간호사는 서로 연락이 되어야되기 때문에 연락하다 관계타입에 전체참여한다.

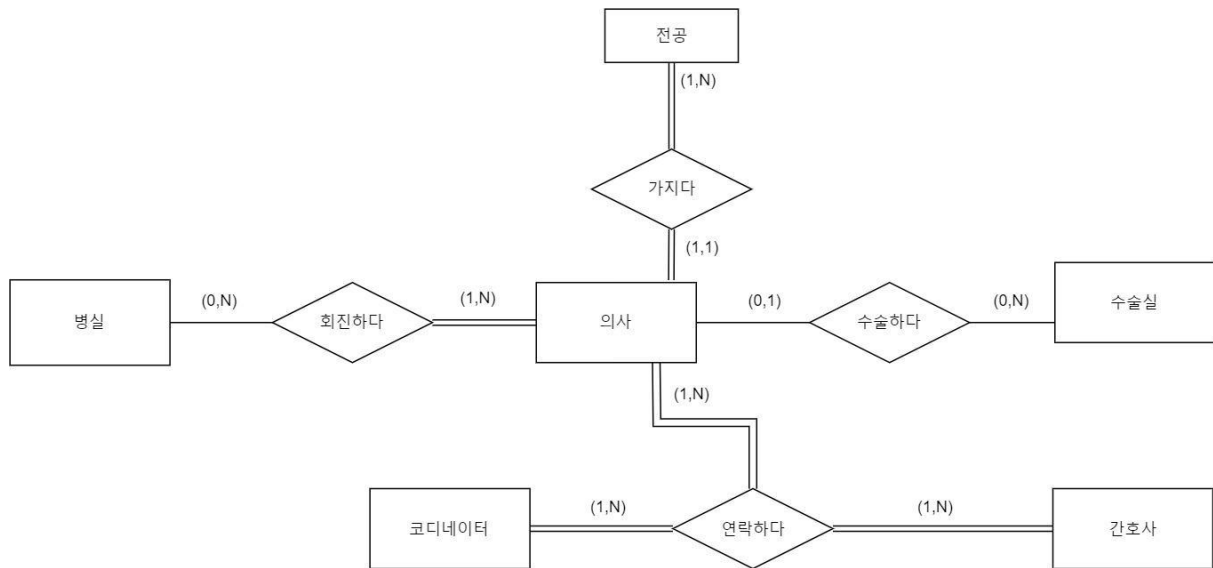


그림 16 의사와 관련된 관계 타입

간호사 엔티티와 환자 엔티티는 담당하다라는 관계 타입을 가진다. 한명 이상의 모든 간호사는 환자를 담당해야 하고, 모든 환자는 최소 한명의 담당 간호사가 있어야한다. 모든 간호사는 1명 이상의 환자를 담당해야 되기 때문에 담당하다 관계 타입에 전체 참여하고, 모든 환자는 최소 1명의 담당 간호사가 있어야 되기 때문에 담당하다 관계 타입에 전체참여 한다.

간호사 엔티티와 병실 엔티티는 담당하다라는 관계 타입을 가진다. 한명 이상의 모든 간호사는 병실을 담당해야 하고, 모든 병실은 한명 이상의 담당 간호사가 있어야한다. 모든 간호사는 1명 이상의 환자를 담당해야 되기 때문에 담당하다 관계 타입에 전체 참여하고, 모든 병실은 1명 이상의 담당 간호사가 있어야 되기 때문에 담당하다 관계 타입에 전체참여 한다.

간호사 엔티티 타입과 부서 엔티티 타입은 속하다 관계타입을 가진다. 모든 간호사는 최소 1개의 부서를 가져야 하고, 부서는 최소 1명, 최대 N명의 간호사가 포함되어야 한다. 이때 간호사와 부서 엔티티 모두 속하다 관계 타입에 전체참여한다.

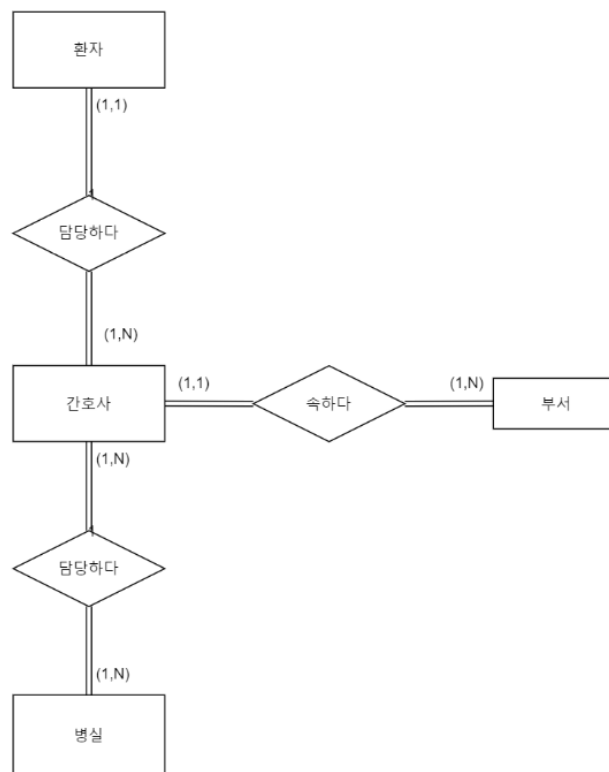


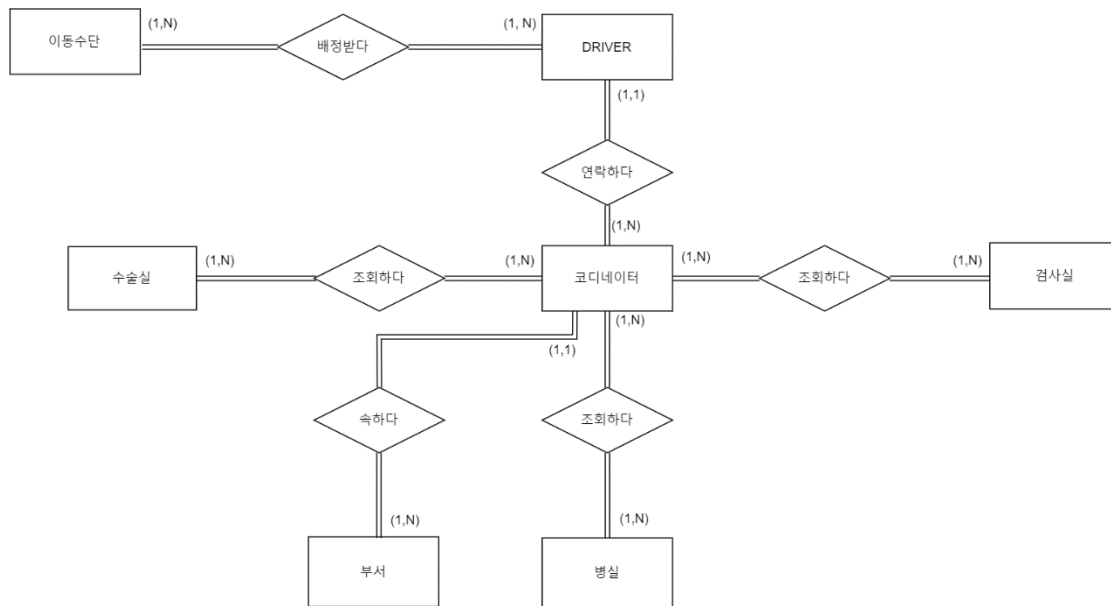
그림 17 간호사와 관련된 관계 타입

DRIVER 엔티타입과 이동수단 엔티타입은 배정받다 관계타입을 가진다. 이 관계 타입에서 **DERIVER** 엔티타입과 이동수단 엔티타입은 **M:N** 관계 타입이다. 모든 이동수단은 **DRIVER**가 배정되고, **DRIVER** 또한 배정된 이동수단이 존재해야 되기 때문에 두 엔티타입은 배정받다 관계타입에 전체참여한다.

코디네이터 엔티타입과 **DRIVER** 엔티타입은 연락하다 관계타입을 갖는다. 한명 이상의 코디네이터는 연락하다 관계에 전체참여하고, 최소 1명의 **DRIVER**는 코디네이터와 연락해야 한다. **DERIVER**는 연락하다 관계 타입에 전체참여한다.

수술실 엔티타입, 검사실 엔티타입, 병실 엔티타입은 코디네이터 엔티타입에 대해, 각각 조회하다 관계 타입을 가지고 있다. 코디네이터는 병원 내 모든 시설을 조회할 수 있어야하므로 각각의 조회하다 관계 타입에 전체참여한다. 수술실, 검사실, 병실도 관계 타입에 전체참여한다. 각각의 조회하다 코디네이터와 **M:N** 관계를 가진다.

코디네이터 엔티타입과 부서 엔티타입은 속하다 관계타입을 가진다. 모든 간호사는 최소 1개의 부서를 가져야 하고, 부서는 최소 1명, 최대 N명의 간호사가 포함되어야 한다. 이때 코디네이터와 부서 엔티타 모두 속하다 관계 타입에 전체참여한다.



1

그림 18 코디네이터와 관련된 관계타입

논리적설계

ER모델을 사용하여 조직체의 데이터베이스 응용을 위한 개념적 스키마를 생성하였다. 논리적 설계 단계에서는 ER 스키마를 관계 데이터 모델의 릴레이션들로 사상해야 한다. 이때 사상할 대상이 어떤 애트리뷰트이고 어떤 타입을 가지는냐에 따라 사상하는 방법이 달라지기 때문에 7단계 알고리즘을 따라가며 릴레이션으로 사상이었다.

단계 1 : 정규 엔티티 타입

환자 엔티티 타입은 기본키인 환자_ID(PAT_ID) 애트리뷰트, 기본키가 아닌 애트리뷰트

이메일(EMAIL),주소(ADDRESS),이름(PAT_NAME),주민등록번호(REGIST_NUM),성별(SEX),직업(JOB),(PHONE_NUM)과 복합 애트리뷰트인 바이탈사인과, 다이어트리뷰트인 지병을 가지고 있다. 복합 애트리뷰트인 바이탈 사인을 구성하는 단순 릴레이션들인 체온(BODY_HEAT), 맥박(PULSE), 혈압(BLOOD_PRESSURE), 호흡(BREATH)를 PATIENT(환자) 릴레이션에 포함시킨다.

다이어트리뷰트인 지병은 단계 1에서는 고려하지 않으므로 PATIENT(환자) 릴레이션에 PATIENT_DISEASE(지병) 릴레이션은 포함시키지 않는다.

PATIENT(환자) (PAT_ID, EMAIL, ADDRESS, PAT_NAME, REGIST_NUM, SEX, JOB, PHONE_NUM, BODY_HEAT, PULSE, BLOOD_PRESSURE, BREATH)

의사 엔티티 타입은 아래의 릴레이션으로 사상된다.

DOCTOR(의사) (DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL, TITLE)

차트 엔티티 타입은 아래의 릴레이션으로 사상된다. 이때 OPINION은 의사소견 애트리뷰트를 의미한다.

CHART(차트) (CHART_ID, OPINION)

간호사 엔티티 타입은 아래의 릴레이션으로 사상된다.

NURSE(간호사) (NUR_ID, NUR_NAME, PHONE_NUM, SEX, EMAIL, TITLE)

코디네이터 엔티티 타입은 아래의 릴레이션으로 사상된다.

COORDINATOR(코디네이터) (CODI_ID, CODI_NAME, SEX, PHONE_NUM, EMAIL)

병실 엔티티 타입은 아래의 릴레이션으로 사상된다. 이때

MAX_PATIENT_NUM은 최대 수용인원 애트리뷰트, CUR_PATIENT_NUM은 현재 수용인원 애트리뷰트를 의미한다.

PATIENTS_ROOM(병실)(ROOM_NUM, FLOOR, MAX_PATIENT_NUM, CUR_PATIENT_NUM)

수술실 엔티티 타입은 아래의 릴레이션으로 사상된다.

OPERATING_ROOM(수술실) (OPER_NUM, FLOOR)

검사실 릴레이션은 아래의 릴레이션으로 사상된다.

LABORATORY(검사실) (LABO_NAME, FLOOR)

DRIVER 릴레이션은 아래와 같은 릴레이션으로 사상된다. 이때 **ROLE**은 역할
에트리뷰트를 의미한다.

**DRIVER(DRIVER) (DRIVER_ID, DRIVER_NAME, SEX, PHONE_NUM,
EMAIL, ROLE)**

전공 에트리뷰트는 아래와 같은 릴레이션으로 사상된다.

SECTION(전공) (SEC_NUM, SEC_NAME)

부서 에트리뷰트는 아래와 같은 릴레이션으로 사상된다.

DEPARTMENT(부서) (DEPT_NUM, DEPT_NAME)

운송수단 에트리뷰트는 아래와 같은 릴레이션으로 사상된다.

TRANSPORT(운송수단) (TRANS_NUM, TYPE)

단계 2 : 약한 엔티티 타입

소유 엔티티 타입은 **PATIENT**이므로 **PATIENT**의 기본키를 **PROTECTOR** 릴레이션에 외래 키로 포함시킨다. 또한 **NURSE**와 연락하다라는 약한 관계를 가지고 있기 때문에 **NURSE**엔티티 타입의 기본키를 **NUR_NO** 외래 키로 참조 받는다. 기본 키를 참조하는 외래 키 **PAT_NOM**, **NUR_NO**와 **PROTECTOR**의 부분키 **PROTECT_NAME**이 모여 기본키가 된다.

PROTECTOR(보호자) (**PAT_NO**, **NUR_NO**, **PROTECT_NAME**, SEX, PHONE_NUM)

단계 3 : 정규 2진 1:1 엔티티 타입

차트 엔티티 타입은 기록받다라는 정규 2진 1:1 관계를 관계 타입을 가진다. 기록하다는 관계타입은 의사 엔티티 타입과 정규 2진 1:N 관계타입을 가지므로 다음 단계에서 다룬다.

차트 엔티티 타입과 환자 엔티티 타입은 기록받다 관계 타입에 전체참여한다. 이때 기록받다 관계 타입에서 의사가 환자에 대한 의사 소견을 적어야되기 때문에 차트 릴레이션에 환자 엔티티 타입의 기본키를 참조하는 외래키를 차트 릴레이션에 추가해준다.

CHART(차트) (**CHART_ID**, OPINION, **PATIENT_ID**)

수술실 엔티티 타입은 기록받다라는 정규 2진 1:1 관계를 관계 타입을 가진다. 수술하다는 관계타입은 의사 엔티티 타입과 정규 2진 1:N 관계타입을 가지므로 다음 단계에서 다룬다.

수술실 릴레이션은 수술실 번호를 기본키로 가지고 관계를 맺는 환자 엔티티 타입의 기본키에 대응되는 값을 참조하는 외래키 인 **OPER_PAT** 가져야 한다. **OPER_DATA**(수술날짜), **OPER_TIME**(수술일정)은 관계 타입이 갖고 있는 애트리뷰트이기 때문에 수술실 릴레이션에 추가해준다.

OPERATING_ROOM(수술실) (**OPER_NUM**, FLOOR, **OPER_PAT**, **OPER_DATE**, **OPER_TIME**)

단계 4 : 정규 2진 1:N 엔티티 타입

의사 엔티티 타입은 수술실과 수술하다 2진 1:N 관계 타입을 가진다. 1측에 대응되는 수술실 엔티티 타입의 기본키가 외래키(OPER_NO)로 의사 릴레이션에 추가된다.

전공 엔티티 타입과 의사 엔티티 타입은 가지다란 관계타입에 대해 모두 전체 참여를 하고 있는 1:N 관계를 가지고 있다. 이때 1 측의 전공 엔티티 타입의 기본키를 의사 릴레이션에 외래키 SEC_NO로 추가해준다. OPER_DATA와 OPER_DATE와 OPER_TIME 은 수술하다 관계타입에서 나온 관계 애트리뷰트이다.

DOCTOR(의사) (DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL, TITLE, OPER_NO, SEC_NO, OPER_DATE, OPER_TIME)

간호사 엔티티 타입은 부서 엔티티 타입과 가지다 라는 정규 2진 1:N 관계 타입을 가지고 있다. 이때 1측의 부서 엔티티 타입의 기본키를 외래키 DNO 라고 지정하여 NURSE 릴레이션에 추가해준다.

NURSE(간호사) (NUR_ID, NUR_NAME, PHONE_NUM, SEX, EMAIL, TITLE, DNO)

드라이버 엔티티 타입은 코디네이터 엔티티 타입과 가지다 라는 정규 2진 1:N 관계 타입을 가지고 있다. 이때 기본키를 외래키 CODI_NUM 라고 지정하여 드라이버 릴레이션에 추가해준다.

DRIVER(드라이버) (DRIVER_ID, DRIVER_NAME, SEX, PHONE_NUM, EMAIL, ROLE, CODI_NUM)

차트 엔티티 타입은 의사 엔티티 타입과 기록되다 정규 2진 1:N 관계 타입이 존재한다. 그렇기 때문에 1측의 의사 엔티티 타입의 기본키를 차트 릴레이션의 외래키RECORDER로 추가해준다.

CHART(차트)(CHART_ID, OPINION, PATIENT_ID, RECORDER)

환자 엔티티 타입은 의사 엔티티 타입, 간호사 엔티티 타입과 각각 담당하다 정규 2진 1:N 관계 타입이 존재한다. 그렇기 때문에 1측의 의사 엔티티 타입의 기본키를 차트 릴레이션의 외래키 CHARGE_DOC으로 추가해주고 1측의 간호사 엔티티 타입의 기본키를 차트 릴레이션의 외래키 CHARGE_NUR로 추가해준다.

PATIENT(환자) (PAT_ID, EMAIL, ADDRESS, PAT_NAME, REGIST_NUM, SEX, JOB, PHONE_NUM, BODY_HEAT, PULSE, BLOOD_PRESSURE, BREATH, CHARGE_DOC, CHARGE_NUR)

병실 엔티티 타입과 환자 엔티티 타입은 입원하다 정규 2진 1:N 관계 타입을 가진다. 이때 1 측의 환자 엔티티 타입의 기본키를 병실 릴레이션에 외래키 STAY_PAT로 추가해준다. IN_DATE(입원날짜)와 OUT_DATE(퇴원날짜)는 입원하다 관계타입에서 나온 관계 애트리뷰트이다.

PATIENTS_ROOM(병실) (ROOM_NUM, FLOOR, MAX_PATIENTS_NUM, CUR_PATIENTS_NUM, IN_DATE, OUT_DATE)

수술실 엔티티 타입과 환자 엔티티 타입은 수술받다 정규 2진 1:N 관계 타입을 가진다. 이때 1측의 환자 엔티티 타입의 기본키를 수술실 릴레이션에 외래키 CHECK_PAT로 추가해준다. CHECK_DATE(입원날짜)와 CHECK_TIME(퇴원날짜)는 입원하다 관계타입에서 나온 관계 애트리뷰트이다. **LABORATORY(수술실)** (LABO_NAME, FLOOR, CHECK_PAT, CHECK_DATE, CHECK_TIME)

코디네이터 엔티티 타입은 부서 엔티티 타입과 가지다 라는 정규 2진 1:N 관계 타입을 가지고 있다. 이때 1측의 부서 엔티티 타입의 기본키를 외래키 DEPT_NUM 라고 지정하여 코디네이터 릴레이션에 추가해준다. **COORDINATOR(코디네이터)**(CODI_ID, CODI_NAME, SEX, PHONE_NUM, DEPT_NAME, EMAIL, DEPT_NUM)

단계 5 : 2진 M:N 관계 타입

드라이버 엔티티 타입과 운송수단 엔티티 타입은 배정받다 2진 M:N 관계 타입을 가진다. 운송수단 엔티티 타입에 대응되는 기본키와 드라이브 엔티티 타입에 대응되는 기본키를 외래키로 포함시켜 **ASSIGN**배정받다 릴레이션에 추가시킨다. **ASSIGN** 릴레이션은 두 외래키를가 모여 기본키가 된다.

ASSIGN(배정받다) (TRANS_NUM, DRIVER_ID)

간호사 엔티티 타입과 환자 엔티티 타입은 배정받다 2진 M:N 관계 타입을 가진다. 간호사 엔티티 타입에 대응되는 기본키와 드라이브 엔티티 타입에 대응되는 기본키를 외래키로 포함시켜 **TAKE_CHARGE_OF(담당하다)** 릴레이션에 추가시킨다. **ASSIGN** 릴레이션은 두 외래키를가 모여 기본키가 된다.

TAKE_CHARGE_OF(담당하다) (NUR_ID, ROOM_NUM)

코디네이터 엔티티 타입과 수술실 엔티티 타입은 조회하다 2진 M:N 관계 타입을 가진다. 간호사 엔티티 타입에 대응되는 기본키와 드라이브 엔티티 타입에 대응되는 기본키를 외래키로 포함시켜 **TAKE_CHARGE_OF(담당하다)** 릴레이션에 추가시킨다. **ASSIGN** 릴레이션은 두 외래키를가 모여 기본키가 된다.

CHECK_OPER(조회하다) (CODI_ID, OPER_NUM)

코디네이터 엔티티 타입과 병실 엔티티 타입은 조회하다 2진 M:N 관계 타입을 가진다. 코디네이터 엔티티 타입에 대응되는 기본키와 병실 엔티티 타입에 대응되는 기본키를 외래키로 포함시켜 **CEHCK_PATIENT_ROOM(조회하다)** 릴레이션에 추가시킨다. **CHECK_PATIENT_ROOM** 릴레이션은 두 외래키를가 모여 기본키가 된다.

CHECK_PATIENT_ROOM(조회하다) (CODI_ID, ROOM_NUM)

코디네이터 엔티티 타입과 수술실 엔티티 타입은 조회하다 2진 M:N 관계 타입을 가진다. 코디네이터 엔티티 타입에 대응되는 기본키와 수술실 엔티티 타입에 대응되는 기본키를 외래키로 포함시켜 **CEHCK_OPER**(조회하다) 릴레이션에 추가시킨다. **CHECK_OPER** 릴레이션은 두 외래키를가 모여 기본키가 된다.

CHECK_OPER(조회하다) ([CODI_ID](#), [OPER_NUM](#))

코디네이터 엔티티 타입과 검사실 엔티티 타입은 조회하다 2진 M:N 관계 타입을 가진다. 코디네이터 엔티티 타입에 대응되는 기본키와 검사실 엔티티 타입에 대응되는 기본키를 외래키로 포함시켜 **CEHCK_LAB**(조회하다) 릴레이션에 추가시킨다. **CHECK_LAB** 릴레이션은 두 외래키를가 모여 기본키가 된다.

CHECK_LAB(조회하다) ([CODI_ID](#), [OPER_NUM](#))

의사 엔티티 타입과 병실 엔티티 타입은 조회하다 2진 M:N 관계 타입을 가진다. 의사 엔티티 타입에 대응되는 기본키와 병실 엔티티 타입에 대응되는 기본키를 외래키로 포함시켜 **VISIT**(회진하다) 릴레이션에 추가시킨다. **VISIT**(회진하다) 릴레이션은 두 외래키를가 모여 기본키가 된다.

VISIT(회진하다) ([ROOM_NUM](#), [DOC_ID](#))

단계 6 : 3진 관계 타입

ER 스키마에서 의사 엔티티 타입, 간호사 엔티티 타입, 코디네이터 엔티티 타입을 연결하는 **CONTACT**(연락하다) 엔티티 관계 타입이 유일한 3진 관계 타입이다. 3진 관계 타입에 대해서는 새로운 릴레이션을 생성한다. 이 릴레이션의 이름을 **CONTACT**라고 부르기로 하자. 의사 엔티티 타입에 대응되는 릴레이션의 기본 키, 간호사 엔티티 타입에 대응되는 릴레이션의 기본 키, 코디네이터 엔티티 타입에 대응되는 릴레이션의 기본키를 **CONTACT**(연락하다) 릴레이션의 외래 키로 포함시킨다. 세 외래 키의 조합이 **CONTACT**(연락하다) 릴레이션의 기본 키가 된다.

CONTACT(연락하다) ([NUR_ID](#), [DOC_ID](#), [CODI_ID](#))

단계 7 : 다치 애트리뷰트

ER 스키마에서 **PATIENT**(환자) 엔티티 타입의 지병 애트리뷰트가 다치 애트리뷰트이다. 다치 애트리뷰트에 대해서 새로운 릴레이션을 생성한다. 이 릴레이션의 이름을 **PATIENT_DISEASE**라고 부르기로 하자. 다치 애트리뷰트에 해당하는 애트리뷰트 **CHRONIC_DSEASE**를 **PATIENT_DISEASE** 릴레이션에 포함시키고, 다치 애트리뷰트를 갖고 있는 **PATIENT** 엔티티 타입에 대응되는 릴레이션의 기본키 **PAT_ID**를 외래 키로 포함시킨다. 두 애트리뷰트의 조합이 **PATIENT_DISEASE** 릴레이션의 기본키가 된다.

PATIENT_DISEASE(지병) ([PAT_ID](#), [CHRONIC_DISEASE](#))

ER 스키마에 대해 7단계의 알고리즘을 적용하면 최종적으로 아래와 같은 릴레이션들의 집합이 생성된다.

- **PATIENT(환자)** (PAT_ID, EMAIL, ADDRESS, PAT_NAME, REGIST_NUM, SEX, JOB, PHONE_NUM, BODY_HEAT, PULSE, BLOOD_PRESSURE, BREATH)
- **DOCTOR(의사)** (DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL, TITLE)
- **CHART(차트)** (CHART_ID, OPINION)
- **NURSE(간호사)** (NUR_ID, NUR_NAME, PHONE_NUM, SEX, EMAIL, TITLE, DNO)
- **COORDINATOR(코디네이터)** (CODI_ID, CODI_NAME, SEX, PHONE_NUM, EMAIL)
- **PATIENTS_ROOM (병실)**(ROOM_NUM, FLOOR, MAX_PATIENT_NUM, CUR_PATIENT_NUM)
- **OPERATING_ROOM(수술실)** (OPER_NUM, FLOOR)
- **LABORATORY(검사실)** (LABO_NAME, FLOOR)
- **DRIVER(DRIVER)** (DRIVER_ID, DRIVER_NAME, SEX, PHONE_NUM, EMAIL, ROLE)
- **SECTION(전공)** (SEC_NUM, SEC_NAME)

- **DEPARTMENT(부서)** (DEPT_NUM, DEPT_NAME)
- **TRANSPORT(운송수단)** (TRANS_NUM, TYPE)
- **PROTECTOR(보호자)** (PAT_NO, NUR_NO, PROTECT_NAME, SEX, PHONE_NUM)
- **CHART(차트)** (CHART_ID, OPINION, PATIENT_ID)
- **OPERATING_ROOM(수술실)** (OPER_NUM, FLOOR, OPER_PAT, OPER_DATE, OPER_TIME)
- **DOCTOR(의사)** (DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL, TITLE, OPER_NO, SEC_NO, OPER_DATE, OPER_TIME)
- **NURSE(간호사)** (NUR_ID, NUR_NAME, PHONE_NUM, SEX, EMAIL, TITLE, DNO)
- **DRIVER(드라이버)** (DRIVER_ID, DRIVER_NAME, SEX, PHONE_NUM, EMAIL, ROLE, CODI_NUM)
- **CHART(차트)**((CHART_ID, OPINION, PATIENT_ID, RECORDER)
- **PATIENT(환자)** (PAT_ID, EMAIL, ADDRESS, PAT_NAME, REGIST_NUM, SEX, JOB, PHONE_NUM, BODY_HEAT, PULSE, BLOOD_PRESSURE, BREATH, CHARGE_DOC, CHARGE_NUR)
- **PATIENTS_ROOM(병실)** (ROOM_NUM, FLOOR, MAX_PATIENTS_NUM, CUR_PATIENTS_NUM, STAY_PAT, IN_DATE, OUT_DATE)
- **LABORATORY(수술실)** (LABO_NAME, FLOOR, CHECK_PAT, CHECK_DATE, CHECK_TIME)
- **COORDINATOR(코디네이터)**(CODI_ID, CODI_NAME, SEX, PHONE_NUM, DEPT_NAME, EMAIL, DEPT_NUM)
- **ASSIGN(배정받다)** (TRANS_NUM, DRIVER_ID)
- **TAKE_CHARGE_OF(담당하다)** (NUR_ID, ROOM_NUM)
- **CHECK_OPER(조회하다)** (CODI_ID, OPER_NUM)
- **CHECK_PATIENT_ROOM(조회하다)** (CODI_ID, ROOM_NUM)

- **CHECK_OPER**(조회하다) ([CODI_ID](#), [OPER_NUM](#))
- **CHECK_LAB**(조회하다) ([CODI_ID](#), [OPER_NUM](#))
- **VISIT**(회진하다) ([ROOM_NUM](#), [DOC_ID](#))
- **CONTACT**(연락하다) ([NUR_ID](#), [DOC_ID](#), [CODI_ID](#))
- **PATIENT_DISEASE**(지병) ([PAT_ID](#), [CHRONIC_DISEASE](#))

정규화

원래의 릴레이션들을 무손실 분해함으로써 중복과 갱신 이상을 최소화하여 일관성과 정확성을 유지하는 과정이다. 부주의한 데이터베이스 설계는 제어할 수 없는 데이터 중복을 야기하여 여러 가지 갱신 이상을 유발하는데, 이러한 문제들을 가지고 있는 릴레이션 스키마를 함수적 종속성과 기본 키를 기반으로 분석하여 좋은 관계 데이터베이스 스키마를 생성한다.

제1정규형

- **DOCTOR**(DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL, TITLE, OPER_NUM, SEC_NUM, OPER_DATE, OPER_TIME)

DOCTOR 릴레이션에서는 OPER_DATE와 OPER_TIME 애트리뷰트가 각 의사들이 진행하는 수술 시간들의 집합 값을 가지므로 제1정규형을 만족하지 못한다. 따라서 **DOCTOR**(DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL, TITLE, SEC_NUM) 릴레이션과 **OEPRATION**(DOC_ID, OPER_NUM, OPER_DATE, OPER_TIME) 릴레이션으로 나눈다.

DOCTOR(DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL, TITLE, SEC_NUM)

OEPRATION(DOC_ID, OPER_NUM, OPER_DATE, OPER_TIME)

- **LABORATORY**(LABO_NAME, FLOOR, CHECK_PAT, CHECK_DATE, CHECK_TIME)

LABORATORY 릴레이션에서는 CHECK_DATE와 CHECK_TIME 애트리뷰트가 각 검사실에서 진행하는 검사 시간들의 집합 값을 가지므로 제1정규형을 만족하지 못한다.

따라서 **LABORATORY**(LABO_NAME, CHECK_PAT, CHECK_DATE, CHECK_TIME, FLOOR) 릴레이션과 **CHECK_LAB**(LABO_NAME, CHECK_PAT, CHECK_DATE, CHECK_TIME) 릴레이션으로 나눈다.

LABORATORY(LABO_NAME, CHECK_PAT, CHECK_DATE, CHECK_TIME, FLOOR)

CHECK_LAB(LABO_NAME, CHECK_PAT, CHECK_DATE, CHECK_TIME)

- **OPERATING_ROOM**(OPER_NUM, FLOOR, OPER_PAT, OPER_DATE, OPER_TIME)

OPERATING_ROOM 릴레이션에서는 OPER_DATE와 OPER_TIME 애트리뷰트가 각 수술실에서 진행하는 수술 시간들의 집합 값을 가지므로 제1정규형을 만족하지 못한다. 따라서 **OPERATING_ROOM**(OPER_NUM, FLOOR) 릴레이션과 **GET_OPER**(OPER_NUM, OPER_PAT, OPER_DATE, OPER_TIME) 릴레이션으로 나눈다.

OPERATING_ROOM(OPER_NUM, FLOOR)

GET_OPER(OPER_NUM, OPER_PAT, OPER_DATE, OPER_TIME)

제2정규형

- **DOCTOR**(DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL, TITLE, SEC_NUM)

DOCTOR 릴레이션에서 SEX, DOC_NAME, PHONE_NUM, EMAIL 은 DOC_ID에 부분적으로 종속하며, TITLE은 DOC_ID와 SEC_NUM 키에 완전하게 함수적으로 종속된다. 따라서 TITLE 애트리뷰트를 새로운 릴레이션에 포함시키고, DOC_ID를 외래키로 이 릴레이션에 포함시키며, 기본키를 (DOC_ID, SEC_NUM)으로 복합 애트리뷰트로 만든다.

따라서 **DOCTOR**(DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL) 릴레이션과 **DOC_TITLE**(DOC_ID, SEC_NUM, TITLE) 릴레이션으로 나눈다.

DOCTOR(DOC_ID, SEX, DOC_NAME, PHONE_NUM, EMAIL)
DOC_TITLE(DOC_ID, SEC_NUM, TITLE)

■ **NURSE**(NUR_ID, NUR_NAME, PHONE_NUM, SEX, EMAIL, TITLE, DNO)

NURSE 릴레이션에서 NUR_NAME, PHONE_NUM, SEX, EMAIL은 NUR_ID에 부분적으로 종속하며, TITLE은 NUR_ID와 DNO키에 완전하게 함수적으로 종속된다. 따라서 TITLE 애트리뷰트를 새로운 릴레이션에 포함시키고, NUR_ID를 외래키로 이 릴레이션에 포함시키며, 기본키를 (NUR_ID, DNO)으로 복합 애트리뷰트로 만든다. 따라서 **NURSE**(NUR_ID, NUR_NAME, PHONE_NUM, SEX, EMAIL) 릴레이션과 **NUR_TITLE**(NUR_ID, DNO, TITLE) 릴레이션으로 나눈다.

NURSE(NUR_ID, NUR_NAME, PHONE_NUM, SEX, EMAIL)
NUR_TITLE(NUR_ID, DNO, TITLE)

제3정규형

어떠한 키가 아닌 모든 애트리뷰트도 기본 키에 대해서 이행적으로 종속되지 않을때 제3정규형을 만족한다고 하는데, 모든 릴레이션에 대해서 어떤 애트리뷰트도 기본 키에 대해서 이행적으로 종속하거나, 추론되지 않기 때문에 제3정규형을 사용하지 않았다.

각 릴레이션별 정규화가 진행되지 않은 이유는 아래 따로 서술하였다.

BCNF

BCNF의 조건인 제3정규형을 모든 릴레이션이 만족을 했고, 모든 결정자가 후보 키였기 때문에 BCNF를 사용하지 않았다.

각 릴레이션별 정규화가 진행되지 않은 이유는 아래 따로 서술하였다.

역정규화

데이터들이 이온화 되어있어서 검색하는데 복잡하다. 따라서 검색 복잡도를 줄이고, 검색 질의들의 수행 속도를 높이기 위해서 역정규화를 실행하였다.

PATIENT(PAT_ID, CHARGE_DOC, CHARGE_NUR, PAT_ROOM, IN_DATE, OUT_DATE, REG_NUM) 릴레이션과 **PAT_INFO**(REGIST_NUM, EMAIL, ADDRESS, PAT_NAME, SEX, PHONE_NUM, BODY HEAT, PULSE, BLOOD PRESSURE, BREATH) 릴레이션으로 나눈 PATIENT와 PAT_INFO를 PATIENT로 다시 통합한다. 따라서 **PATIENT**(PAT_ID, EMAIL, ADDRESS, PAT_NAME, REGIST_NUM, SEX, PHONE_NUM, BODY HEAT, PULSE, BLOOD PRESSURE, BERATH, CHARGE_DOC, CHARGE_NUR, PAT_ROOM, IN_DATE, OUT_DATE)

■ **PATIENT**(PAT_ID, EMAIL, ADDRESS, PAT_NAME, REGIST_NUM, SEX, PHONE_NUM, BODY HEAT, PULSE, BLOOD PRESSURE, BERATH, CHARGE_DOC, CHARGE_NUR, PAT_ROOM, IN_DATE, OUT_DATE)

정규화한 릴레이션을 제외한 다른 릴레이션들을 정규화 하지 않은 이유

■ **DRIVER**(DRIVER_ID, DRIVER_NAME, SEX, PHONE_NUM, EMAIL, ROLE, Codi_NUM)

DRIVER 릴레이션의 모든 애트리뷰트가 원자값만을 갖고, 어떤 후보 키에도 속하지 않는 모든 애트리뷰트들이 DRIVER의 기본 키에 완전하게 함수적으로 종속되며 된다.

키가 아닌 모든 애트리뷰트가 DRIVER 릴레이션의 기본 키에 이행적으로 종속되지 않기 때문이다.

■ **CHART(CHART_ID, OPINION, PATIENT_ID, RECORDER)**

CHART 릴레이션에서 모든 애트리뷰트가 원자값만을 갖고, 어떤 후보 키에 속하지 않는 모든 애트리뷰트들이 CHART의 기본 키에 완전하게 함수적으로 종속되며, 키를 제외한 모든 애트리뷰트가 CHART 릴레이션의 기본 키에 이행적으로 종속되지 않기 때문이다.

■ **ASSIGN**(TRANS_NUM, DRIVER_ID)

ASSIGN 관계 릴레이션에서는 TRANS_NUM 외래키를 TRANSPORT 릴레이션에서 가져오고, DRIVER_ID 외래키를 DRIVER 릴레이션에서 가져온다.
외래키로 가져온 TRANS_NUM 와 DRIVER_ID는 원자값만을 갖고, 복합키를 제외한 애트리뷰트가 없기때문에 제2정규형, 제3정규형을 만족하기 때문에 정규화를 하지 않았다.

■ **TAKE_CHARGE_OF**(NUR_ID, ROOM_NUM)

TAKE_CHARGE_OF 관계 릴레이션에서는 NUR_ID 외래키를 NURSE 릴레이션에서 가져오고, ROOM_NUM 외래키를 PATIENTS_ROOM 릴레이션에서 가져온다.
복합키인 NUR_ID와 ROOM_NUM 은 원자값만을 가지고, 복합키면서 키를 제외한 애트리뷰트가 없기때문에 정규화를 하지 않았다.

■ **CHECK**(CODI_ID, OPER_NUM)

CHECK 관계 릴레이션에서는 CODI_ID 외래키를 COORDINATOR 릴레이션에서 가져오고, OPER_NUM 외래키를 OPERATING_ROOM 릴레이션에서 가져온다. 복합키인 CODI_ID와 OPER_NUM은 원자값만을 가지고, 복합키면서 키를 제외한 애트리뷰트가 없기때문에 정규화를 하지 않았다.

■ **VISIT**(ROOM_NUM, DOC_ID)

VISIT 관계 릴레이션에서는 PATIENTS_ROOM 릴레이션에 포함된 ROOM_NUM 을 외래키로 가져오고, DOCTOR 릴레이션에 포함된 DOC_ID를 외래키로 가져온다. ROOM_NUM과 DOC_ID는 원자값만을 가지고, 복합키면서 키를 제외한 애트리뷰트가 없기때문에 정규화를 하지 않았다.

■ **CONTACT**(NUR_ID, DOC_ID, CODI_ID)

CONTACT 관계 릴레이션에서는 NURSE 릴레이션에 포함된 NUR_ID를 외래키로 가져오고, DOCTOR 릴레이션에 포함된 DOC_ID를 외래키로 가져오며, COORDINATOR 릴레이션에 포함된 CODI_ID를 외래키로 가져온다.
NUR_ID, DOC_ID와 CODI_ID 3개의 외래키가 복합키이며, 원자값만을 갖는다. 그리고 3개의 키들을 제외한 애트리뷰트가 없기때문에 정규화를 하지 않았다.

■ **PATIENT_DISEASE**(PAT_ID, CHRONIC_DISEASE)

PATIENT_DISEASE 릴레이션에서 PAT_ID는 PATIENT에 포함된 PAT_ID값을 가져온 외래키이고, PATIENT_DISEASE 릴레이션에서 기본키는 PAT_ID와 CRHONIC_DISEASE 애트리뷰트의 조합이다. 키를 제외한 애트리뷰트가 없기 때문에 정규화를 하지 않았다.

■ **PROTECTOR**(PAT_ID, NUR_ID, PROTECT_NAME, SEX, PHONE_NUM)

PATIENT 릴레이션에서 포함된 PAT_ID를 외래키로 가져왔고, NURSE 릴레이션에서 포함된 NUR_ID를 외래키로 가져왔다. PROTECTOR 릴레이션에 모든 애트리뷰트들은 원자값만을 가지며, 어떤 후보 키에 속하지 않는 모든 애트리뷰트들이 PROTECTOR의 기본 키에 완전하게 함수적으로 종속되며, 키를 제외한 모든 애트리뷰트가 PROTECTOR 릴레이션의 기본 키에 이행적으로 종속되지 않기 때문이다.

■ **TRANSPORT**(TRANS_NUM, TYPE)

TRANS_NUM과 TYPE은 원자값만을 갖는다. TRANS_NUM이 기본키이고 TYPE은 TRANS_NUM에 완전하게 함수적으로 종속된다. 키를 제외한 TYPE 애트리뷰트는 TRANSPORT 릴레이션의 기본 키에 이행적으로 종속되지 않기 때문에 정규화 하지 않았다.

■ **SECTION**(SEC_NAME, SEC_NUM)

SEC_NAME과 SEC_NUM은 원자값만을 갖는다. SEC_NAME은 기본키이고, SEC_NUM은 SEC_NAME에 완전하게 함수적으로 종속된다. 키를 제외한 SEC_NUM 애트리뷰트는 SECTION 릴레이션의 기본 키에 이행적으로 종속되지 않기 때문에 정규화 하지 않았다.

■ **DEPARTMENT**(DEPT_NUM, DEPT_NAME)

DEPARTMENT 릴레이션의 모든 애트리뷰트들은 원자값만을 가지고, DEPT_NAME은 기본키인 DEPT_NUM에 완전하게 함수적으로 종속된다. 키를 제외한 DEPT_NAME 애트리뷰트는 DEPARTMENT 릴레이션의 기본 키에 이행적으로 종속되지 않기 때문에 정규화 하지 않았다.

■ **COORDINATOR**(CODI_ID, CODI_NAME, SEX, PHONE_NUM, DEPT_NUM, EMAIL)

COORDINATOR 릴레이션의 모든 애트리뷰트는 원자값만을 가진다. DEPARTMENT 릴레이션에 포함된 DEPT_NUM을 외래키로 가져오고, 어떤 후보 키에도 속하지 않는 모든 애트리뷰트들은 COORDINATOR의 기본키인 CODI_ID에 완전하게 함수적으로 종속된다. 키가 아닌 모든 애트리뷰트들은 COORDINATOR 릴레이션의 기본키에 이행적으로 종속되지 않기 때문에 정규화를 하지 않았다.

■ **PATIENTS_ROOM**(ROOM_NUM, FLOOR, MAX_PATIENTS_NUM, CUR_PATIENTS_NUM, IN_DATE, OUT_DATE)

PATIENTS_ROOM 릴레이션의 모든 애트리뷰트들은 원자값만을 가진다. 어떤 후보 키에도 속하지 않는 모든 애트리뷰트들은 PATIENTS_ROOM 릴레이션의 기본키인 ROOM_NUM에 완전하게 함수적으로 종속된다. 키가 아닌 모든 애트리뷰트들은 PATIENTS_ROOM 릴레이션의 기본키에 이행적으로 종속되지 않기 때문에 정규화를 하지 않았다.

■ **BE_HOSPITALIZED**(ROOM_NUM, STAY_PAT, IN_DATE, OUT_DATE)

SPATEINT 릴레이션에서 포함된 PATIENT ID를 외래키로 가져오고, 어떤 후보 키에도 속하지 않은 모든 애트리뷰트들은 BE_HOSPITALIZED 릴레이션의 기본 키인 ROOM_NUM에 완전하게 함수적으로 종속된다. 키가 아닌 모든 애트리뷰트들은 BE_HOSPITALIZED 릴레이션의 기본 키에 이행적으로 종속되지 않기 때문에 정규화를 하지 않았다.

■ CHECK_PATIENT_ROOM(CODI_ID, ROOM_NUM)

CHECK_PATIENT_ROOM 릴레이션에 포함된 CODI_ID와 ROOM_NUM 애트리뷰트는 각각 COORDINATOR 릴레이션과, ROOM 릴레이션에 포함된 애트리뷰트값을 가져온 외래키 값이며, 두 애트리뷰트를 묶어 복합키로 사용한다. 두 애트리뷰트는 원장값만을 가지며, 복합키인 두 애트리뷰트밖에 없기 때문에 정규화를 하지 않았다.

물리적 설계

인덱스 설계

애트리뷰트와 튜플이 많이 존재하고 상이한 값들이 많은 PATIENT (환자) 테이블에 검색을 보다 빠르게 하기 위해서 REGIST_NUM(주민등록번호)를 인덱스로 생성하였다.

CHART(차트) 테이블 또한 환자마다 차트가 존재하기 때문에 상이한 값들이 많아 PATIENT_ID(환자 ID)로 인덱스를 생성하였다.

이 두 테이블을 제외한 다른 테이블들은 기본인덱스를 제외한 다른 애트리뷰트는 인덱스를 지정할 만 한 후보 애트리뷰트가 존재하지 않거나 데이터가 크지 않아 인덱스를 생성하지 않았다.

뷰 설계

PATIENTS_ROOM 릴레이션에 대해서 “각 병실의 병실번호, 최대 수용 환자 인원, 현재 수용 환자 인원”으로 정의된 뷰”를 정의해보았다. 아래의 뷰의 정의에는 뷰의 애트리뷰트들을 별도로 명시했기 때문에 뷰에는 **ROOM_NUM**, **MAX_PATIENT_NUM**, **CUR_PATIENT_NUM**의 세 애트리뷰트가 포함된다.

```
CREATE VIEW PATIENTS_ROOM (ROOM_NUM, MAX_PATIENT_NUM, CUR_PATIENT_NUM)
AS SELECT ROOM_NUM, MAX_PATIENT_NUM, CUR_PATIENT_NUM
FROM PATIENTS_ROOM;

UPDATE PATIENTS_ROOM
SET CUR_PATIENT_NUM = (SELECT COUNT(STAY PAT)
FROM BE_HOSPITALIZED B, PATIENTS_ROOM PR
WHERE B.ROOMNO = PR.ROOM_NUM AND ROOMNO=7
GROUP BY ROOM_NUM )
WHERE ROOM_NUM =7;
```

	ROOM_NUM	MAX_PATIENT_NUM	CUR_PATIENT_NUM
1	1	10	3
2	2	10	3
3	3	10	3
4	4	10	3
5	5	8	3
6	6	6	3
7	7	10	3

OPERATION 과 **GET_OPER** 릴레이션에 있는 “의사아이디, 수술실에 있는 환자 아이디, 수술 날짜, 수술 시간으로 이루어진 뷰”를 정의하였다. 뷰의 애트리뷰트들을 별도로 명시하지 않았기 때문에 뷰에 속하는 애트리뷰트들의 이름은 기본 릴레이션의 애트리뷰트들의 이름과 같고 결국 뷰에는 **DOCNO**, **OPER_PAT_ID**, **OPER_DATE**, **OPER_TIME**의 네가지의 애트리뷰트가 포함된다. **OPERATION** 릴레이션에 있는 **OPER_NUM**, **OPER_DATE**, **OPER_TIME**과 **GET_OPER** 릴레이션에 있는 **OPERNO**, **OPER_DATE**, **OPER_TIME**을 동등조인한다. 동등조인한 **OPERATION**과, **GET_OPER**에서 **DOCNO**, **OPER_PAT_ID**, **OPER_DATE**, **OPER_TIME**을 **SELECT**문을 사용하여 데이터를 검색한다. **VIEW**를 사용하여 간단하게 질의를 표현할 수 있다.

```
CREATE VIEW OPER_SCHEDULE
AS SELECT OP.DOCNO, OP.OPER_PAT_ID, OP.OPER_DATE, OP.OPER_TIME
FROM OPERATION OP, GET_OPER GOP
WHERE OP.OPER_NUM = GOP.OPERNO AND OP.OPER_DATE = GOP.OPER_DATE AND OP.OPER_TIME = GOP.OPER_TIME;
WHERE PP.PAT_ID = PT.PATNO;
```

	DOCNO	OPER_PAT_ID	OPER_DATE	OPER_TIME
1	3399	1	200322	1320
2	2316	2	201216	1800
3	2589	9	200123	1800
4	2458	11	200318	1320

PATIENT와 **PROTECTOR** 릴레이션에 대해서 “환자 이름, 환자의 보호자의 이름, 환자의 보호자의 핸드폰 번호로 이루어진 뷰”에 대해서 정의하였다. 아래의 뷰의 정의에는 뷰의 애트리뷰트들을 별도로 명시하였기 때문에 뷰에는 **PATIENT**, **PROTECTOR**, **PHONE_NUM**의 세 애트리뷰트가 포함된다.

```
CREATE VIEW PROTECTOR_INFO (PATIENT, PROTECTOR, PHONE_NUM)
AS SELECT PP.PAT_NAME, PT.PROTECT_NAME, PT.PHONE_NUM
FROM PATIENT PP, PROTECTOR PT
WHERE PP.PAT_ID = PT.PATNO;
```


	PAT_ID	PATIENT	PROTECTOR	PHONE_NUM
1	1	국유경	금설주	1032468879
2	2	공언문	정희련	1021316878
3	3	원나실	민마술	1003116979
4	4	최세무	권덕관	1064758597
5	5	강똥희	신요취	1032134628
6	6	오현삼	미초구	1075346545
7	7	명앞형	양두식	1042124355
8	8	강사직	홍시인	1094447585
9	9	김열성	기호공	1066859888
10	10	주완타	전철초	1046658545
11	11	권람혼	탁부화	1056785242
12	12	전원근	이래균	1052947754
13	13	한해온	허손윤	1012101101
14	14	선호훈	남직미	1022311131
15	15	임우송	김민직	1046252525
16	16	곽늑문	김철구	1099564646
17	17	유취서	선영옥	1021665498
18	18	용상연	현우규	1032465489
19	19	양공오	신초백	1024201456
20	20	문환세	석남소	1075345698

DOCTOR, DOC TITLE SECTION 릴레이션에 대하여 “의사이름, 전공명, 직급으로 이루어진 뷰”에 대해서 정의하였다. 아래의 뷰의 정의에는 뷰의 애틀리뷰트들을 별도로 명시하였기 때문에 뷰에는 DOC_NAME, SEC_NAME, TITLE의 세 애틀리뷰트가 포함된다.

```
CREATE VIEW DOC_INFO (DOC_NAME, SEC_NAME, TITLE, PHONE)
AS SELECT DOC_NAME, SEC_NAME, TITLE, PHONE_NUM
FROM DOCTOR DD, DOC TITLE DT, SECTION S
WHERE DD.DOC ID=DT.DOCNO AND DT.SEC NO = S.SEC NUM;
```

	DOC_NAME	SEC_NAME	TITLE	PHONE
1	김태연	특수치료시설	인턴	1021515448
2	명곤직	외과	부교수	1075576526
3	배월상	내과	교수	1064896548
4	서변나	외과	인턴	1035637549
5	서오을	외과	의대생	1095454541
6	머민숨	수술실	부교수	1085497654
7	에일리	특수치료시설	조교수	1063326497
8	머곤석	외과	교수	1031688798
9	옥래신	외과	레지던트	1021048765
10	용타제	수술실	의대생	1022349934
11	위신을	내과	부교수	1000124687
12	정성하	외과	의대생	1073243456
13	채후관	내과	조교수	1075718965
14	최배태	외과	부교수	1008730154
15	최취리	수술실	레지던트	1031887954
16	탁광유	특수치료시설	레지던트	1022487654
17	폴킴	수술실	펠로우	1028856997
18	하민을	특수치료시설	의대생	1042649564

PATIENT, DOCTOR, NURSE 릴레이션에 있는 환자 이름, 의사 이름, 간호사 이름으로 이루어진 뷰를 정의한다. 이 때 두 개이상의 릴레이션에서 가져온 애틀리뷰트들의 이름이 같아서 뷰에서 두 개이상의 애틀리뷰트들의 이름이 같게 될 수 있기 때문에 모든 애틀리뷰트들의 이름을 지정했다.

뷰의 정의에서 뷰의 애트리뷰트들을 별도로 명시했기 때문에 뷰에는 PATIENT_NAME, DOCTOR, NURSE의 세 애트리뷰트가 포함된다. PATIENT 릴레이션에 있는 CHARGE_DOC과 DOCTOR 릴레이션에 있는 DOC_ID를 동등조인 하고, PATIENT 릴레이션에 있는 CHARGE_NUR과 NURSE 릴레이션에 있는 NUR_ID를 동등조인한다. 동등조인한 PATIENT, DOCTOR, NURSE 릴레이션에 있는 PAT_NAME, DOC_NAME, NUR_NAME 값을 SELECT 문을 사용하여 데이터를 검색한다. 이러한 SELECT 문은 VIEW를 사용하여 간단하게 질의를 표현할 수 있다.

```
CREATE VIEW CHARGE(PATIENT_NAME, DOCTOR, NURSE)
AS SELECT P.PAT_NAME, D.DOC_NAME, N.NUR_NAME
FROM PATIENT P, DOCTOR D, NURSE N
WHERE P.CHARGE_DOC = D.DOC_ID AND P.CHARGE_NUR = N.NUR_ID;
```

	PATIENT_NAME	DOCTOR	NURSE
1	강동희	옥래신	최취리
2	강사직	명곤직	장운미
3	공언문	여곤석	천곤운
4	곽낙문	채후관	공유
5	국유겸	배철상	인곤요
6	권람훈	최배태	강주영
7	김열성	어민숨	나노일
8	명윤희	채후관	소요철
9	문환세	명곤직	홍개직
10	선효훈	정성하	제시카
11	양공오	최배태	인곤요
12	오현삼	위신율	윤창익
13	용상연	폴킴	나동현
14	원나실	여곤석	진박장
15	유휘서	에일리	심찬섭
16	임우송	김태연	조보마
17	전원근	하인율	남평천
18	주완타	위신율	정정훈
19	최세무	탁광유	홍개직
20	한해온	용타제	티파니

BE_HOSPITALIZED, PATIENT 릴레이션에 있는 환자 아이디, 환자 이름, 입원방 번호, 입원 날짜, 입원시간으로 이루어진 뷰를 정의한다.

뷰의 정의에서 뷰의 애트리뷰트들을 별도로 명시했기 때문에 뷰에는 PAT_ID, PAT_NAME, ROOM, IN_DATE, OUT_DATE의 다섯개의 애트리뷰트가 포함된다. BE_HOSPITALIZED 릴레이션에 있는 STAY_PAT과 PATIENT 릴레이션에 있는 PAT_ID를 동등조인했다.

동등조인한 BE_HOSPITALIZED, PATIENT 릴레이션에 있는 PAT_ID, PAT_NAME, ROOMNO, IN_DATE, OUT_DATE 값을 SELECT 문으로 검색한다. 이것을 VIEW를 사용하여 간단하게 질의를 표현할 수 있다.

```
CREATE VIEW LOCATION PAT (PAT_ID, PAT_NAME, ROOM, IN_DATE, OUT_DATE)
AS SELECT P.PAT_ID, P.PAT_NAME, B.ROOMNO, B.IN_DATE, B.OUT_DATE
FROM BE_HOSPITALIZED B, PATIENT P
WHERE B.STAY_PAT = P.PAT_ID;
```

트렌젝션

질의 예시

- 20년 3월 환자 수술 일정.

```
SELECT *  
FROM OPER_SCHEDULE  
WHERE OPER_DATE LIKE '2003%';
```

	DOCNOM	OPER_PAT_ID	OPER_DATE	OPER_TIME
1	3399	1	200322	1320
2	2458	11	200318	1320

- 현재 수용인원이 최대 수용 인원보다 적은 병실확인

```
SELECT *  
FROM PATIENTS_COUNT  
WHERE CUR_PATIENT_NUM < MAX_PATIENT_NUM;
```

	ROOM_NUM	MAX_PATIENT_NUM	CUR_PATIENT_NUM
1	1	10	3
2	2	10	3
3	3	10	3
4	4	10	3
5	5	8	3
6	6	6	3
7	7	10	3

- 2번 병실 환자 정보

	PAT_NAME	IN_DATE	OUT_DATE
1	최세무	200714	200801
2	강똥희	200506	200530
3	오현삼	200531	200711

```
SELECT PAT_NAME, IN_DATE, OUT_DATE  
FROM LOCATION_PAT  
WHERE ROOM = 2;
```

결론

중증외상환자를 치료 가능한 병원으로 빠르게 이송하는 것은 병원 전 단계를 중증외상환자 치료 체계에 있어서 가장 중요한 부분이다. 심혈관계 질환, 뇌혈관계 질환, 산과적 질환 등의 응급 질환 뿐 아니라 특히 중증외상환자들의 이송에 있어서는 헬리콥터 등을 이용한 항공 이송은 환자가 발생한 현장에서 바로 치료 가능한 병원으로 이송하거나 하급응급의료기관으로부터 응급의료기관까지의 전원 이송 시간을 단축하고 의료진을 빨리 접촉할 수 있다는 점에서 환자의 예후에 중요한 역할을 하고 있다고 알려져 있다.

또한 2019년 기준 OECD 보건지표를 보게 되면 한 명이 일 년간 병원을 찾은 횟수는 OECD 회원국 중 가장 많으며, 병상의 수는 일본에 이어 2위로 많았다. 하지만 병원을 찾는 환자 수는 많은 데 비해 OECD 대비 우리나라의 의료진 수는 부족한 편이다. 그 중 외과의는 2018년 기준 외과 전공의 정원은 180명인데 148명이 지원해 충원율은 82%에 그쳤다. 이국종 교수가 소속된 아주대병원도 지난해에 이어 올해도 외과 전공의 모집에 실패해 외과에 3년차 전공의가 유일한 병원이 됐다.

결론적으로 중증외상환자를 현장에서부터 최대한 빨리 처치하면서 이송 중 적절한 초기 치료를 제공하고 최단 시간 내 수술적 치료까지 이어지는 것 것은 국가적으로 중증외상시스템을 구축하여 중증외상환자들의 생존률을 향상시키는데 가장 중요한 요소들 중 하나인 것이다.

우리는 위와 같은 인력난 문제를 더욱 최적화된 시스템으로 환자에게 신속한 치료를 제공하고자 기존의 병원 시스템이 아닌 권역응급외상센터에 맞춰 최적화되어있는 데이터베이스를 설계하여 빠른 시간 내 시스템 사용자들이 원하고자 하는 정보를 실시간으로 확인하고 수정할 수 있도록 하였다. 이를 통해 치료시간을 더욱 단축시켜 환자의 생존률을 높이는데 기여하고 대한민국의 의료사각지대를 해소하고자 하였다.