**디지털 병 간호 서비스**

**엔지니어링 DB 11조**

**담당 교수님: 박기진 교수님**

201520202 이민혁

201520219 김현용

201720170 김동빈

201820168 이예지

201820217 류이슬

* **목 차 -**

1. **서론**

**1)프로젝트 문제 정의**

**2)프로젝트 목적 및 수행 내용 요약**

1. **본론**

**1) DB 목적 및 해결 알고리즘 설명**

**2) 프로그램 설계**

**(1) 요구 사항 분석: 요구사항 명세서**

**(2) 개념적 설계: 개체, 속성, 관계 도출 및 E-R 다이어그램**

**(3) 논리적 설계: 릴레이션 도출 및 데이터 설명**

**(4) 정규화 검증**

**3) 프로그램 구현**

**(1) MySQL Workbench를 통한 데이터 정의 및 조작**

**(2) Python(Jupyter)을 통한 알고리즘 구현**

**(3) C# Windows Forms을 이용한 시각화**

1. **결론**

**1. 서론**

**1) 프로그램 문제 정의**

중국에서 발생한 신종 코로나 바이러스 감염에 의한 폐렴이 급속도로 전 세계에 확산됨에 따라 우리나라를 포함한 여러 나라에서 코로나 19 바이러스에 감염된 환자를 찾아내고 치료하며 감염을 예방하느라 전력을 다하고 있다. 국가적으로 사회적 거리 두기를 장려하고 확진자가 다녀간 곳은 출입국 봉쇄와 이동 제한을 실시하는 등 추가적인 바이러스 감염을 막기 위해 모두가 노력하고 있다.

이러한 시국에 우리나라의 많은 병원들이 병원 내 감염 예방을 위해 병문안을 부분 통제하기 시작하였고 입원실과 중환자실, 응급의료센터, 로비 입구 등에 열 감지 카메라를 설치하여 출입하는 사람들을 관리하고 있다. 실제로 신종 코로나 바이러스로 인해 병원에서 병문안을 통제하여 가족, 친지의 병문안을 가지 못한다는 사람들이 많다는 소식을 접하였고 이러한 문제를 해결할 방법에 대해 생각해 보았다.

신종 코로나 바이러스의 문제뿐만 아니라 매일 출근시간에 쫒기고 일에 치이며 살아가는 바쁜 현대인들, 피치못할 사정으로 인해 시간을 낼 수 없는 사람들 등 각자의 사정으로 인해 병문안을 가고 싶어도 그러지 못하는 사람들이 많았다. 또한 장기적인 치료가 필요하거나 수술이 있는 환자가 오랜 시간 병원에 입원할 경우에는 간병인이 필요로 한데, 가족들이 환자 곁을 계속 지키기는 힘들 것이라 생각하였다.

코로나 바이러스로 인한 병문안 통제 문제와 사정이 있어 환자 곁을 지키지 못하는 간병인들을 모두 고려하여 병문안 및 병 간호를 집이나 직장 등 병원 밖에서 비대면으로 환자의 상태를 확인할 수 있도록 한다면 좋은 방안이 될 수 있을 것이라 판단하였다. 이에 따라 디지털 간호 데이터 베이스의 필요성은 충분하다고 생각하였고, 본 프로젝트에서는 병원 방문없이 환자의 현재 상태와 호전도, 다른 질병에 걸릴 가능성 등의 질병 및 건강 데이터를 제공하고 더 나아가, 환자 식단의 영양 성분과 처방받은 약 성분을 보호자가 확인하고 안심할 수 있도록 하는 디지털 병 간호 서비스 시스템을 설계해 보았다.

**2) 프로젝트 목적 및 수행 내용 요약**

프로젝트의 큰 목표는 아래와 같다.

1. 보호자가 병원 방문을 하지 않더라도 입원한 환자의 건강 상태를 확인할 수 있는 시스템
2. 누적된 건강 상태 데이터를 분석하여 앞으로의 건강 상태를 예측하는 시스템
3. 환자의 식단을 분석하여 섭취한 영양 성분 정보를 제공하는 시스템
4. 환자가 처방받은 약 성분의 정보를 제공하는 시스템

이 프로젝트를 수행하기 위해 공공 데이터 포털 사이트에서 건강 상태 및 질병 관련 데이터를 찾았다. 이를 토대로 건강 상태 데이터를 입력하면 특정 질병에 걸릴 가능성을 분석할 수 있는 알고리즘을 설계하였고, 그 과정에서 해당 질병에 큰 영향을 주는 요인들을 추출할 수 있었다. 그 결과, 이 시스템의 사용자인 보호자가 입원 환자를 입력하면 환자의 현재 건강 상태를 담은 데이터가 제공되고, 환자의 누적된 건강 상태 데이터를 바탕으로 예측한 앞으로의 건강 상태와 다른 질병에 걸릴 가능성 역시 확인할 수 있도록 하였다. 더하여, 식단이 입력되면 영양 성분을 분석할 수 있는 알고리즘을 설계하였고, 보호자가 입원기간 동안 환자에게 제공되는 식단과 그에 해당하는 영양 성분 데이터를 제공받을 수 있도록 하였다. 이 시스템을 통해 사용자인 보호자가 환자의 입원에 대해 안심할 수 있도록 식단뿐만 아니라 환자에게 처방된 약 성분 데이터도 제공받을 수 있도록 데이터베이스를 설계해 보았다.

데이터를 다루고 필요한 정보만 추출해내는 것에서 나아가, 입력하고 도출해내는 일련의 과정들을 통해 얻은 정보들을 한눈에 확인할 수 있도록 하기 위해 Jupyter 노트북을 이용하여 데이터 분석하였다. 또한 이를 사용자가 편리하게 전달받을 수 있도록 C# Windows Forms 프로그램을 이용해 시스템의 완성도를 높이 고자 하였다.

**2. 본론**

1. **DB 목적 및 해결 알고리즘 설명**

본 프로젝트는 크게 네 가지 목적을 두고 있으며 각 목적을 해결하는 알고리즘 내용은 아래와 같다.

1. **보호자가 병원 방문을 하지 않더라도 입원한 환자의 건강 상태를 확인할 수 있는 시스템**

우리는 본 프로그램을 통해 사용자인 보호자에게 입원 환자의 질병과 건강 상태 데이터를 보여줄 것이다. 그러기 위해 우리는 보호자의 입원 환자를 입력 받은 후, 그 환자의 성명, 나이, 성별, 질병명의 기본 정보와 아래와 같은 목록의 건강 상태 데이터를 제공한다.

1. 건강 상태 데이터 목록

|  |  |
| --- | --- |
| 번호 | 건강 상태 |
| 1 | 허리둘레 |
| 2 | 통증 불편 |
| 3 | 수면 시간 |
| 4 | 수축 혈압 |
| 5 | 불안, 우울 |
| 6 | 헤마토크리트 |
| 7 | 체질량 지수 |
| 8 | 일상활동 |
| 9 | 이완 혈압 |
| 10 | 헤모글러빈 |

1. 질병명 데이터 목록

|  |  |  |  |
| --- | --- | --- | --- |
| 번호 | 질병명 | 번호 | 질병명 |
| 1 | 고혈압 | 13 | 간암 |
| 2 | 이상 지질 혈증 | 14 | 유방암 |
| 3 | 뇌졸증 | 15 | 자궁경부암 |
| 4 | 심근경색증 | 16 | 갑상선암 |
| 5 | 협심증 | 17 | 폐암 |
| 6 | 관절염 | 18 | 우울증 |
| 7 | 골관절염 | 19 | 아토피 피부염 |
| 8 | 류마티즈 관절염 | 20 | 알레르기 비염 |
| 9 | 골다공증 | 21 | 백내장 |
| 10 | 폐결핵 | 22 | 녹내장 |
| 11 | 천식 | 23 | B형 간염 |
| 12 | 갑상선 질환 | 24 | C형 간염 |
| 13 | 위암 |  |  |

수많은 질병과 요인들이 존재하지만, 데이터베이스 구축을 위해 구할 수 있는 데이터 중 여러 질병들 중에 가장 영향력 있고 중요한 요인들만 선정하였다. 실제로 이 시스템이 상용화된다면, 모든 질병과 요인들을 다룰 수 있는 것을 목적으로 한다.

1. **누적된 건강 상태 데이터를 분석하여 앞으로의 건강 상태와 다른 질병에 걸릴 가능성을 예측하는 시스템**

본 시스템을 이용해 현재 환자의 앞으로의 건강 상태예측과 앞서 확인한 24가지 질병 중 해당 질병일 확률이 높은 5가지 질병에 대한 확률을 제공할 것이다.

- 해당 환자에 대한 정보 파악

정보를 확인하고자 하는 환자의 ID를 입력을 받게 되면, 해당 환자의 질병명과 환자가 검사 받은 항목들에 대한 결과 값들을 갖고 분석을 실시하게 된다.

- 과거 데이터 분석

과거 환자데이터는 24가지 질병의 유무와, 10가지의 건강상태 데이터 목록을 갖고 있다. 이를 파이썬을 이용한 로지스틱 회귀를 통해 각각의 질병과 건강상태 데이터 목록 간의 상관 관계식을 도출했다. 자세한 설명은 Python을 통한 알고리즘 구현 부분에서 다루기로 하겠다.

- 과거 데이터 분석한 결과 값을 이용한 분석

과거 데이터를 분석한 결과 값과, 분석하고자 하는 환자의 건강상태 데이터들을 이용해 24가지 질병 중 걸릴 확률이 높은 5가지 질병을 예측해 볼 것이다. 또한 해당 환자가 이전에도 여러 번 검사를 받았었다면, 그 데이터들을 비교해 상태의 호전 정도를 시각화 해서 보여준다.

이 시스템이 상용화된다면, 모든 질병들을 다룰 수 있는 것을 목적으로 한다.

**(3)환자의 식단을 분석하여 섭취한 영양 성분 정보를 제공하는 시스템**

보호자에게 환자의 식단과 영양 성분의 정보를 제공하도록 데이터베이스를 구축하기 위해서는 가상의 식단 데이터가 필요하였는데, 실제 병원에서 환자에게 제공하는 식단표를 참고하여 가상의 식단들을 데이터화 하여 사용하였다. 그 데이터화된 식단들 중에서 환자의 아침, 점심, 저녁에 해당하는 식단이 정해지도록 하였고, 각 식단은 밥, 국, 3가지의 반찬으로 이루어져 있다. 음식별 영양 성분 데이터를 활용하여 각 식단에 해당되는 음식들을 추출하였고, 이를 도합하여 보호자에게 식단과 함께 제공할 수 있도록 하였다. 실제로 이 시스템이 상용화된다면, 모든 식단과 음식들을 다룰 수 있는 것을 목적으로 한다.

**(4) 환자가 처방받은 약 성분의 정보를 제공하는 시스템**

해당 질병들을 치료하기 위해 쓰이는 여러 약들 중 공통된 성분을 추출하여 데이터 값으로 선정하였다.

**2) 프로그램 설계**

**(1) 요구 사항 분석(요구사항 명세서)**

1. 디지털 간호 서비스에 현재 환자로 등록하려면, 환자ID, 성별, 나이, 질병 명 데이터를 입력해야 한다

2. 현재 환자는 현재 환자 ID로 식별한다.

3. 의약품은 약품명, 약성분1, 약성분2에 대한 정보를 유지해야 한다.

4. 의약품은 약품명으로 식별한다.

5. 질병별로 처방받는 약이 있다.

6. 환자는 한 번에 여러 약을 복용할 수 있으며, 하나의 약 또한 여러 환자에게 복용될 수 있다.

7. 환자가 복용중인 약은 의약품 중 하나여야 한다.

8. 현재 환자는 여러 날에 검사를 받을 수 있지만 하루에 한 번만 받을 수 있다.

9. 측정 데이터는 현재 환자의 환자ID, 검사날짜, 허리둘레, 통증, 평균 수면시간, 수축기 혈압, 이완기 혈압, 불안증상 여부, 헤마토크리트 수, 헤모글로빈 수, 체질량 지수, 일상활동 가능 여부, 신체정보 측정한다.

10. 검사 데이터는 환자ID와 검사날짜로 식별한다.

11. 과거 환자는 퇴원한 환자로서 ID, 성별, 나이, 질병 명, 허리둘레, 통증, 평균 수면시간, 수축기 혈압, 이완기 혈압, 불안증상, 헤마토크리트, 헤모글로빈 수, 체질량 지수, 일상활동 가능 여부 들을 유지하고 있다.

12. 과거 환자는 과거 환자 ID로 식별한다.

13. 현재환자는 복수의 날짜에 병원 식단표 메뉴 중 하나를 골라 아침, 점심, 저녁 식사를 할 수 있으며 이를 유지해야 한다.

14. 식단표는 식단표ID, 밥, 국, 반찬에 대한 정보를 담고 있다.

15. 식단표는 식단표ID로 식별한다.

16. 영양소 데이터는 식단표에 있는 각 음식에 대한 영양소 정보를 담고 있다.

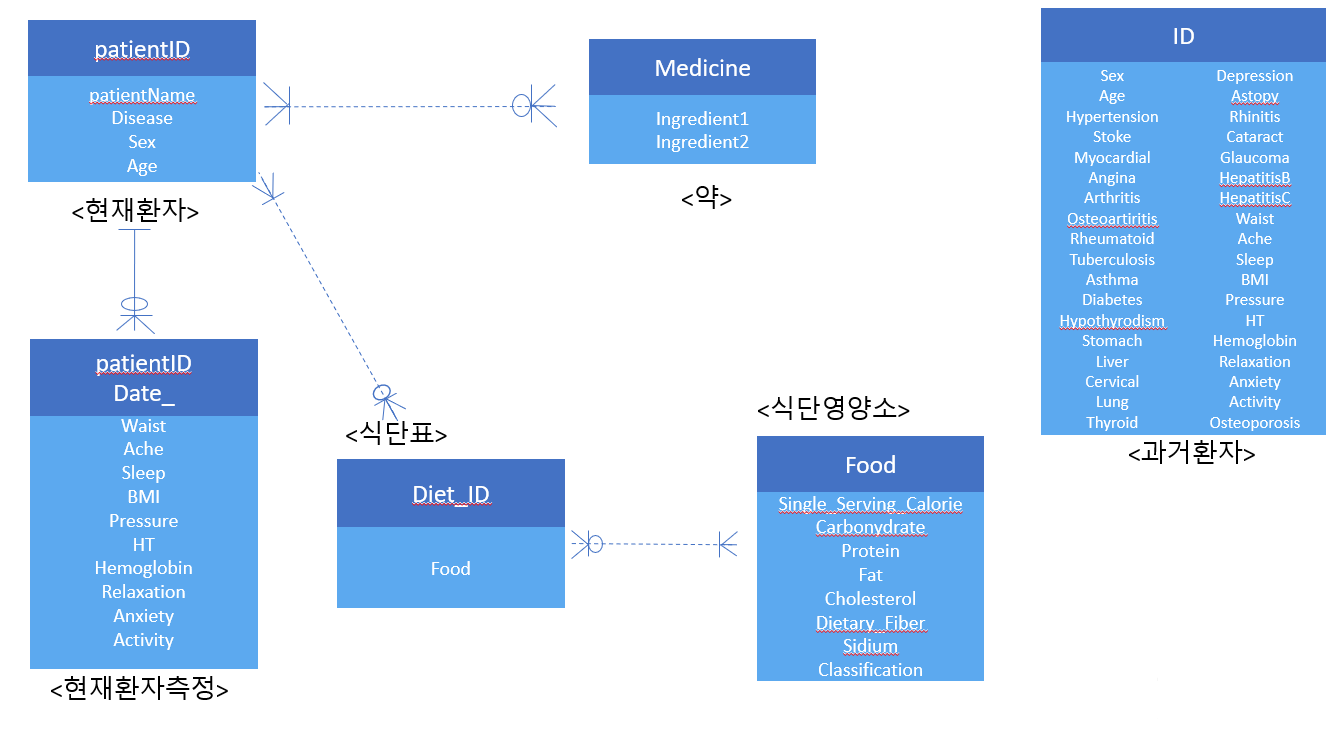
17. 영양소 데이터는 음식 명, 칼로리, 탄수화물, 단백질, 지방, 콜레스트롤, 식이섬유, 나트륨, 분류 정보를 유지한다.

18. 각 음식에 대한 정보는 영양소 데이터의 정보 중 하나와 일치해야 하며, 영양소 데이터에 있는 정보를 여러 번 사용할 수 있다.

19. 영양소 데이터는 음식명으로 식별한다.

1. **개념적 설계 및 논리적 설계: EER다이어그램과 관계 데이터 모델**

요구사항 명세서를 바탕으로 도출한 E-R 다이어그램은 다음과 같다.



현재환자 개체는 강한 개체로 ID, 성별, 나이, 질병 4가지의 속성을 가지고 있다. 의약품 개체 또한 강한 개체로 질병 명, 성분 2가지의 속성을 가지고 있다. 환자는 여러 약을, 한종류의 약은 여러 환자에게 복용될 수 있으므로 두 개체는 다대다 관계이다.

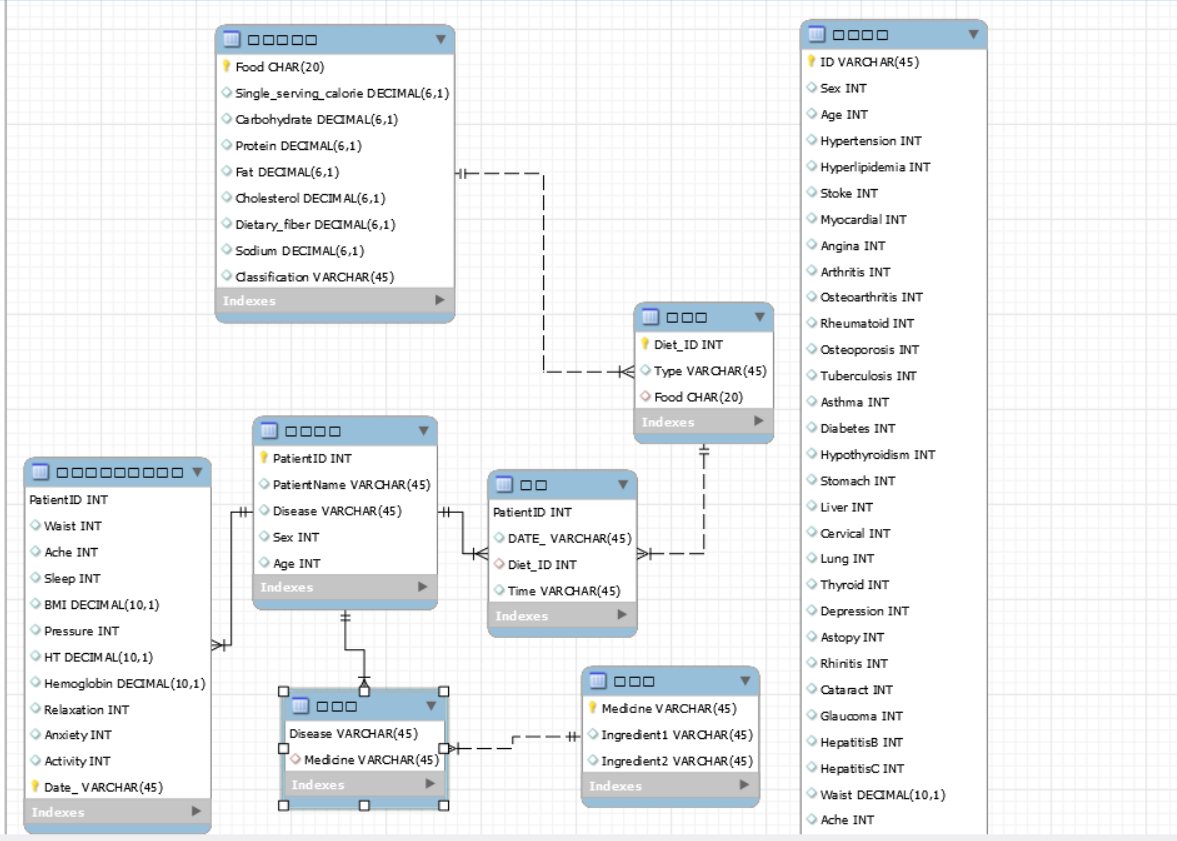
검사데이터 개체는 약한 개체이다. 현재 환자가 검진을 받지 않으면 생기지 않는 데이터이기 때문이다. 이 개체는 검사날짜, 허리둘레, 통증, 평균 수면시간, 수축기 혈압, 이완기 혈압, 불안증상 여부, 헤마토크리트 수, 헤모글로빈 수, 체질량 지수, 일상활동 가능 여부, 신체정보 12가지의 속성을 가지고 있다. 환자는 여러 번 측정을 할 수 있고 한번의 측정데이터는 한 환자의 데이터 이므로 현재환자 개체: 검사데이터 개체의 관계는 일대다 관계이다.

식단표 개체는 강한 개체로 식단표ID 속성을 가지고 있다. 현재환자 개체와 식단표 개체는 식사라는 관계를 가지며 식사는 날짜, 아침, 점심, 저녁 4가지의 속성을 가지고 있다.

음식 영양소 데이터 개체는 강한 개체로 음식 명, 칼로리, 탄수화물, 단백질, 지방, 콜레스테롤, 식이 섬유, 나트륨, 분류 9가지의 속성을 가지고 있다. 식단표 개체와 음식 영양소 데이터 개체는 구성이라는 관계를 가지며 하나의 식단표에는 여러 음식이, 하나의 음식은 여러 식단표에 포함될 수 있으므로 두 개체는 다대다 관계이다.

과거 환자 데이터 개체는 강한 개체로 검사날짜, 허리둘레, 통증, 평균 수면시간, 수축기 혈압, 이완기 혈압, 불안증상 여부, 헤마토크리트 수, 헤모글로빈 수, 체질량 지수, 일상활동 가능 여부, 신체정보 12가지의 속성을 가지고 있다.

이 E-R 다이어그램을 바탕으로 Workbench를 통해 다시 작성한 EE-R 다이어그램은 다음과 같다.



기본적으로 개체는 릴레이션으로 변환한다.

개체간 다대다 관계는 각 개체의 기본키를 외래키로 하는 새로운 릴레이션을 만들어 구현함을 알 수 있다.

개체간 일대다 관계는 참조되는 릴레이션의 기본키를 외래키로 하는 속성을 만들어 구현함을 알 수 있다.

이 EE-R 다이어그램의 개체는 관계데이터 모델의 릴레이션과 상통한다. 개체간 다대다관계로 만들어진 새로운 릴레이션도 EE-R 다이어그램의 새로운 개체(새로운 테이블)로 표현하였으며, 기본 키와 외래 키 관계도 전부 표현 되어있다. 따라서 개념적 모델링에 연속적으로 논리적 모델링이 되었다.

다음은 릴레이션에 관한 설명이다.

1. 현재환자 릴레이션

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PatientID** | name | Sex | Age | Disease |

현재환자 릴레이션의 기본키는 환자ID(patientID)이다. 속성으로는 이름(name), 성별(sex), age(나이), 질병(disease) 3가지를 갖는다. 환자는 동명이인이 있을 수 있으므로 각 투플은 이름으로 식별하지 않고 별도의 ID로 식별한다. 각 속성은 순서대로 int, varchar(45), varchar(45), int, int이다. 환자의 모든 정보는 누락될 수 없으므로 not null이며 sex는 두가지이므로 제약 조건으로 0또는 1만 입력할 수 있게 하였다. sex 0은 남자, 1은 여자이다. 나이는 음수일 수 없으므로 제약조건으로 0보다 크도록 하였다.

1. 현재환자 측정 데이터 릴레이션

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Patient ID** | **Date** | Waist | Ache | Sleep | BMI |
| Pressure | HT | hemoglobin | relaxation | anxiety | activity |

현재환자 측정 데이터 릴레이션의 기본키는 환자ID(patientID)와 날짜(Date)의 복합 키이다. 한 명의 환자가 여러 날 측정한 건강 데이터는 다를 수 있고 같은 날짜에 검자를 받은 환자들은 여러 명일 수 있으므로 기본키가 아닌 속성들은 기본키에 완전 함수 종속된다. BMI속성, HT속성, Hemoglobin속성은 Decimal(10,1)이고 date는 Date타입이다. 나머지는 int타입이다. 측정 데이터는 누락되거나 측정하지 않을 수도 있으므로 null값을 허용하며 patientID는 현재환자 릴레이션의 기본키를 참조하는 외래키이다.

1. 과거환자 릴레이션

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | Sex | Age | Hypertension | Hyperlipidemia | Stoke | Myocardial angina | Arthritis | Osteoporosis | Tuberculosis |
| Asthma | Diabetes | Hypothyroidism | Stomach | Liver | Cervical | Lung | Thyroid | Depression | Astropy |
| Rhinitis | Cataract | Glaucoma | hepatitis | hepatitis | Waist |  |  |  |  |

과거환자 릴레이션의 기본키는 과거환자ID(ID)이다. BMI, HT, Hemoglobin속성만 decimal(10,1) 이고 나머지 속성은 int이다. 기본키인 id와 속성 sex, age는 환자가 가지고 있어야만 하는 정보이므로 null값을 허용하지 않는다.

1. 음식 영양소 릴레이션

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Food** | Calorie | | Carbonydrate | | Protein | | Fat |
| Cholesterol | | Fiber | | Sodium | | Classification | |

음식 영양소 릴레이션의 기본키는 음식명이다. 속성으로는 음식명, 칼로리, 탄수화물, 단백질, 지방, 콜레스테롤, 식이섬유, 나트륨, 분류를 가지며 차례로 음식명은 varchar(20), 분류는 varchar(45)이며 not null, 나머지 속성은 decimal(6,1)이고 특정 영양소가 포함되지 않은 음식도 있을 수 있으므로 null 값을 허용한다.

1. 약이름 릴레이션

|  |  |
| --- | --- |
| **Disease** | Medicine |

약이름 릴레이션의 기본키는 질병명(disease)이다. 속성으로는 질병명(disease), 약이름(medicine)을 가지며 각각 varchar(45), varchar(10)이고 본 데이터베이스에서는 처방할 약이 있는 질병만을 다루기 때문에 null값을 허용하지 않는다. Medicine는 약성분 릴레이션의 기본키를 참조하는 외래키이다.

1. 약성분 릴레이션

|  |  |  |
| --- | --- | --- |
| **Medicine** | Ingredient1 | Ingredient2 |

약성분 릴레이션의 기본키는 약이름이다. 속성으로는 약이름(medichine), 성분1(ingredient1), 성분2(ingredient2) 3가지를 갖는다. 성분이 하나인 약이 있을 수도 있으므로 각 성분은 null 값을 허용한다.

1. 식단 릴레이션

|  |  |  |  |
| --- | --- | --- | --- |
| **PatientID** | **Date** | Diet\_ID | **Time** |

식단 릴레이션의 기본키는(환자ID(patientID), 날짜(Date), 식사시간(Time)이다. Time은 아침, 점심,저녁 중 하나의 값을 가진다. 일반적인 일대다 관계로 설정하면 아침, 점심, 저녁 총 3번을 초과하는 식단을 참조할 수 있으므로 time이라는 아침, 점심, 저녁 중 하나에 해당하는 값을 가지는 속성을 기본키에 포함하게 되었다. 식단 릴레이션의 외래키는 식단ID(diet\_id)로 식단표 릴레이션의 복합키인 기본키 중 하나의 속성인 diet\_id를 참조한다. 일반적으로 외래키는 기본키를 참조하지만 복합키인 기본키중 하나의 속성을 참조하기 위해서 인덱스를 설정해주었다.

1. 식단표 릴레이션

|  |  |  |
| --- | --- | --- |
| **Diet\_ID** | Type | **Food** |

식단표 릴레이션의 기본키는 식단ID(Diet\_ID)와 음식명(Food)이다. type속성은 밥, 국, 반찬 중 하나의 값을 갖는 속성이다.

모든 외래키 제약사항은 on delete no action 및 on update no action이다. 개체, 참조 무결성 제약조건에 위배되는 투플을 삭제, 업데이트 할 수 없다.

**(4) 정규화 검증**

설계한 8개의 릴레이션에 대하여 정규화 검증을 실시한 결과 내용은 다음과 같다.

**환자 릴레이션**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PatientID** | patientname | Sex | Age | Disease |

1NF 검증: 모든 속성이 더이상 분해되지 않는 원자 값을 가진다.

2NF 검증: 릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키인 PatientID에 완전 함수 종속이 된다.

3NF 검증: 릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키 인 PatientID에 이행적 함수 종속이 되지 않는다.

BCNF 검증: 릴레이션이 제3 정규형에 속하고, 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이다.

4NF 검증: 릴레이션이 BCNF에 속하고, 다치 종속이 제거가 되어 있다.

**식사 릴레이션**

|  |  |  |  |
| --- | --- | --- | --- |
| **PatientID** | **Date** | Diet\_ID | **Time** |

1NF 검증: 모든 속성이 더 이상 분해되지 않는 원자 값을 가진다.

2NF 검증: 릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키인 PatientID, Date, Time에 완전 함수 종속이 된다.

3NF 검증: 릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키 인 PatientID, Date, Time에 이행적 함수 종속이 되지 않는다.

BCNF 검증: 릴레이션이 제3 정규형에 속하고, 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이다.

4NF 검증: 릴레이션이 BCNF에 속하고, 다치 종속이 제거가 되어 있다.

**식단표 릴레이션**

|  |  |  |
| --- | --- | --- |
| **Diet\_ID** | Type | **Food** |

1NF 검증: 모든 속성이 더 이상 분해되지 않는 원자 값을 가진다.

2NF 검증: 릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키인 Diet\_ID, Food에 완전 함수 종속이 된다.

3NF 검증: 릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키 인 Diet\_ID, Food에 이행적 함수 종속이 되지 않는다.

BCNF 검증: 릴레이션이 제3 정규형에 속하고, 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이다

4NF 검증: 릴레이션이 BCNF에 속하고, 다치 종속이 제거가 되어 있다.

**영양소 데이터 릴레이션**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Food** | Calorie | | Carbonydrate | | Protein | | Fat |
| Cholesterol | | Fiber | | Sodium | | Classification | |

1NF 검증: 모든 속성이 더 이상 분해되지 않는 원자 값을 가진다.

2NF 검증: 릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키인 음식명에 완전 함수 종속이 된다.

3NF 검증: 릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키 인 음식명에 이행적 함수 종속이 되지 않는다.

BCNF 검증: 릴레이션이 제3 정규형에 속하고, 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이다

4NF 검증: 릴레이션이 BCNF에 속하고, 다치 종속이 제거가 되어 있다

**약 성분 릴레이션**

|  |  |  |
| --- | --- | --- |
| **Medicine** | Ingredient1 | Ingredient2 |

1NF 검증: 모든 속성이 더 이상 분해되지 않는 원자 값을 가진다.

2NF 검증: 릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키인 질병명에 완전 함수 종속이 된다.

3NF 검증: 릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키 인 질병명에 이행적 함수 종속이 되지 않는다.

BCNF 검증: 릴레이션이 제3 정규형에 속하고, 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이다

4NF 검증: 릴레이션이 BCNF에 속하고, 다치 종속이 제거가 되어 있다.

**현재 환자 측정 데이터 릴레이션**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Patient ID** | **Date** | Waist | Ache | Sleep | BMI |
| Pressure | HT | hemoglobin | relaxation | anxiety | activity |

1NF 검증: 모든 속성이 더 이상 분해되지 않는 원자 값을 가진다.

2NF 검증: 릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키인 Patient ID, Date에 완전 함수 종속이 된다.

3NF 검증: 릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키 인 Patient ID, Date에 이행적 함수 종속이 되지 않는다.

BCNF 검증: 릴레이션이 제3 정규형에 속하고, 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이다

4NF 검증: 릴레이션이 BCNF에 속하고, 다치 종속이 제거가 되어 있다.

**약이름 릴레이션**

|  |  |
| --- | --- |
| **Disease** | Medicine |

1NF 검증: 모든 속성이 더 이상 분해되지 않는 원자 값을 가진다.

2NF 검증: 릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키인 Disease에 완전 함수 종속이 된다.

3NF 검증: 릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키 인 Disease에 이행적 함수 종속이 되지 않는다.

BCNF 검증: 릴레이션이 제3 정규형에 속하고, 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이다

4NF 검증: 릴레이션이 BCNF에 속하고, 다치 종속이 제거가 되어 있다.

**과거환자 릴레이션**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | Sex | Age | Hypertension | Hyperlipidemia | Stoke | Myocardial angina | Arthritis | Osteoporosis | Tuberculosis |
| Asthma | Diabetes | Hypothyroidism | Stomach | Liver | Cervical | Lung | Thyroid | Depression | Astropy |
| Rhinitis | Cataract | Glaucoma | hepatitis | hepatitis | Waist |  |  |  |  |

1NF 검증: 모든 속성이 더 이상 분해되지 않는 원자 값을 가진다.

2NF 검증: 릴레이션이 제1 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키인 ID에 완전 함수 종속이 된다.

3NF 검증: 릴레이션이 제2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키 인 ID에 이행적 함수 종속이 되지 않는다.

BCNF 검증: 릴레이션이 제3 정규형에 속하고, 릴레이션의 함수 종속 관계에서 모든 결정자가 후보키이다

4NF 검증: 릴레이션이 BCNF에 속하고, 다치 종속이 제거가 되어 있다.

**3) 프로그램 구현**

**(1) MySQL Workbench를 통한 데이터 정의 및 조작**

프로그램 구현에 있어 우리는 MySQL Workbench의 SQL 언어를 사용하여 데이터를 정의 및 조작하였다. 테이블 정의어인 DDL을 사용해 테이블을 생성하였고 테이블 조작어인 DML을 통해 테이블에 데이터를 삽입하거나, 테이블에 저장된 데이터를 수정, 삭제, 검색하였다. 이러한 기능들을 DB 설계 과정에서 활용한 내용은 아래와 같다.

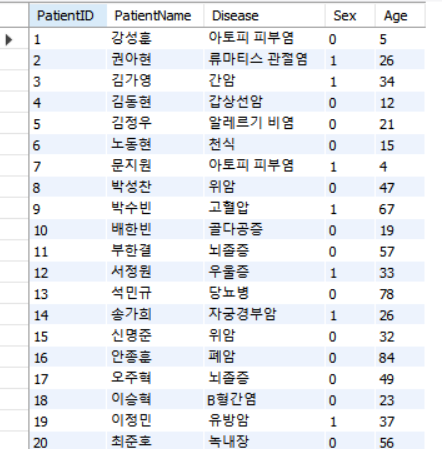
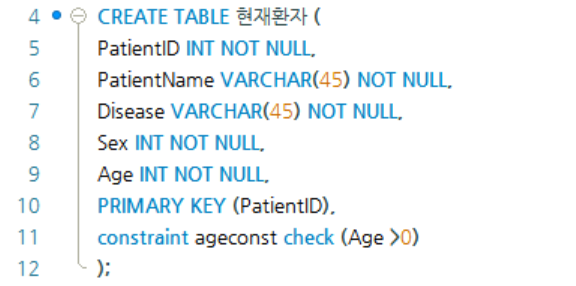
**<DDL을 활용한 데이터 정의>**

DDL언어를 사용하여 테이블을 생성하였으며 데이터 삽입은 Excel과의 연동을 통해 삽입하였다.



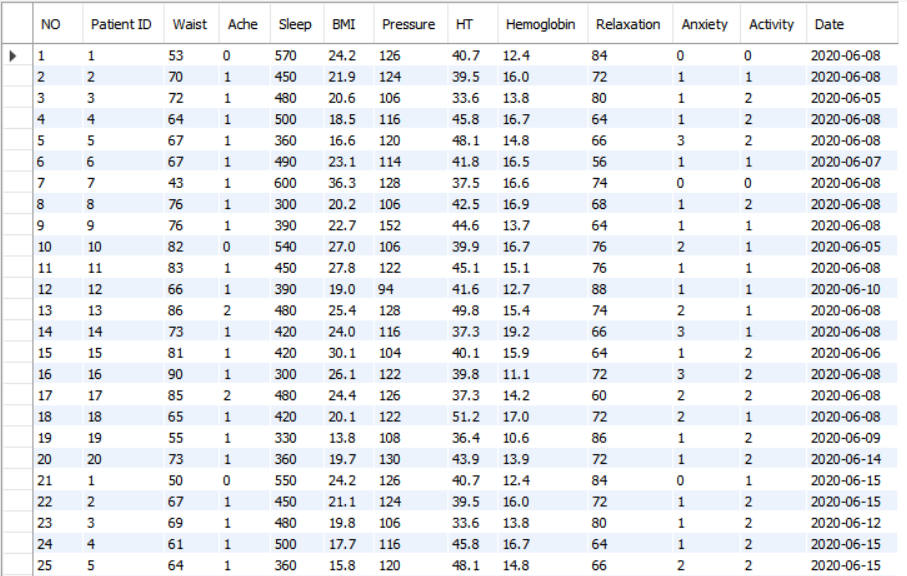
* 우선 사용할 데이터 베이스를 생성 및 사용에 대한 환경을 구축하였다.

1. ‘현재환자’ 테이블 생성 및 데이터 삽입

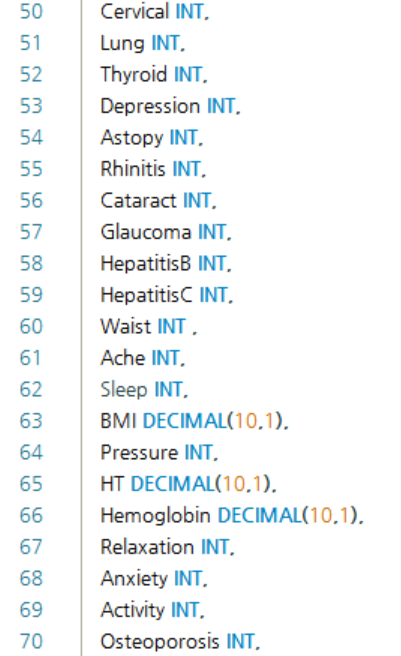
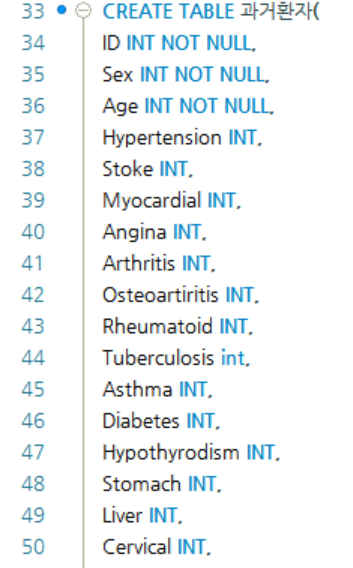


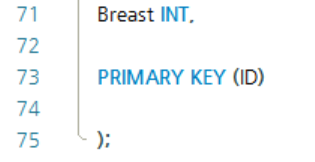
② ‘현재환자측정’ 테이블 생성 및 데이터 삽입

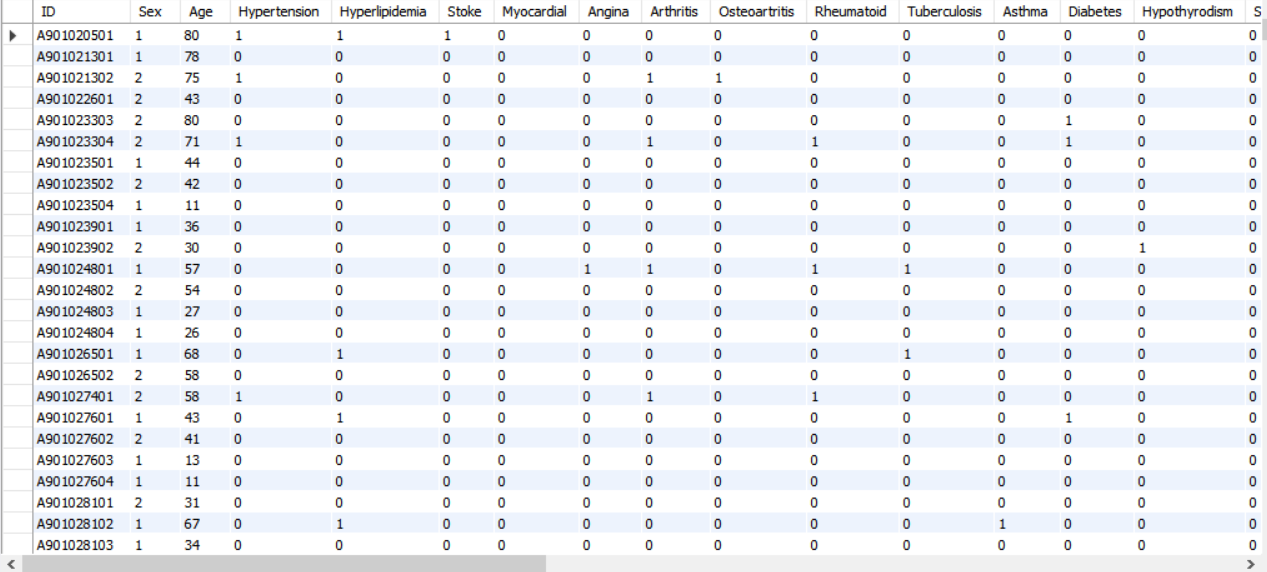




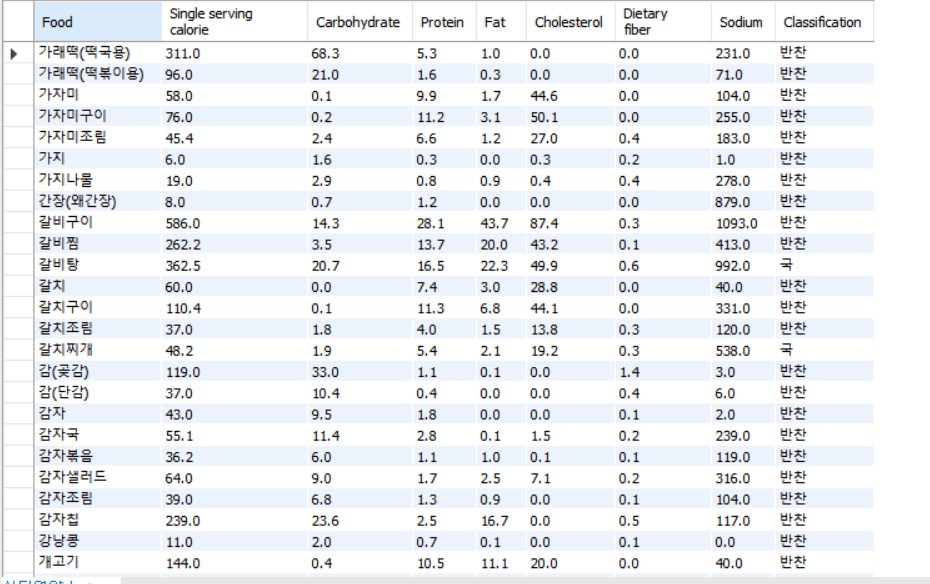
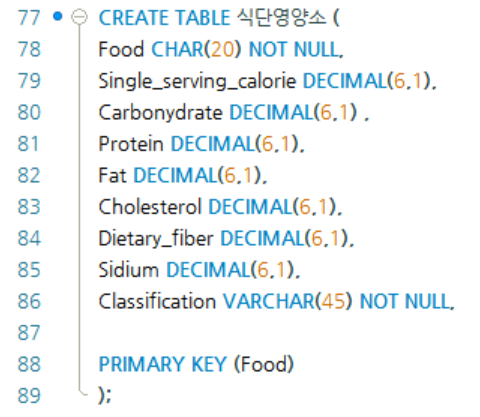
③ ‘과거환자’ 테이블 생성 및 데이터 삽입



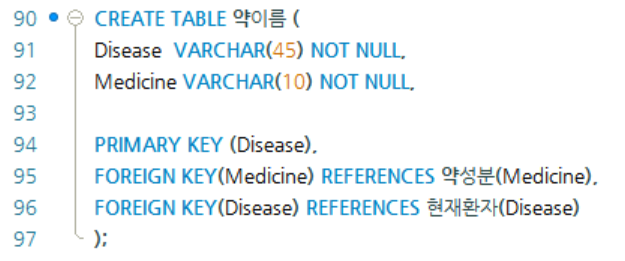


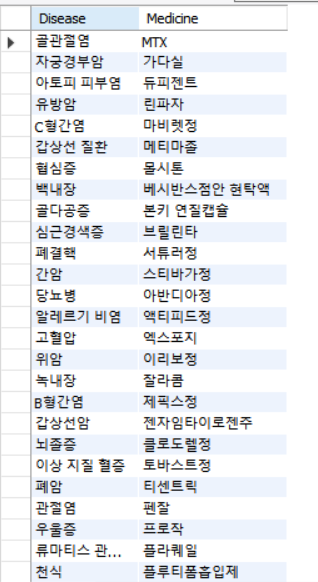


④ ‘식단영양소’ 테이블 생성 및 데이터 삽입

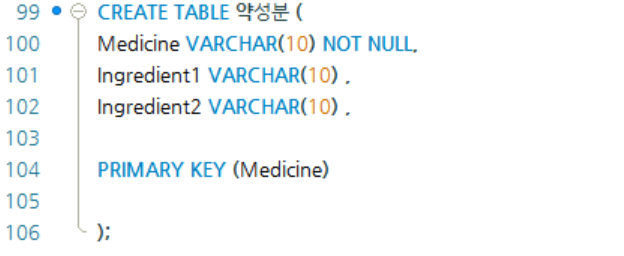


⑤ ‘약이름’ 테이블 생성 및 데이터 삽입



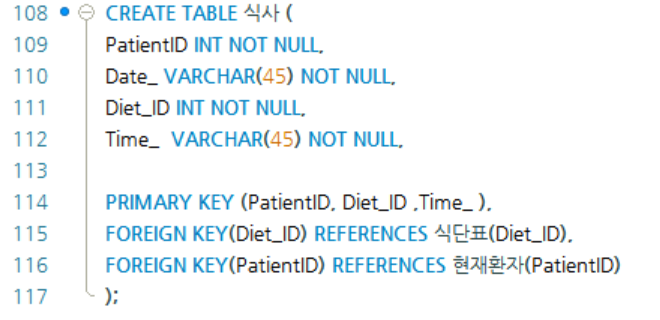


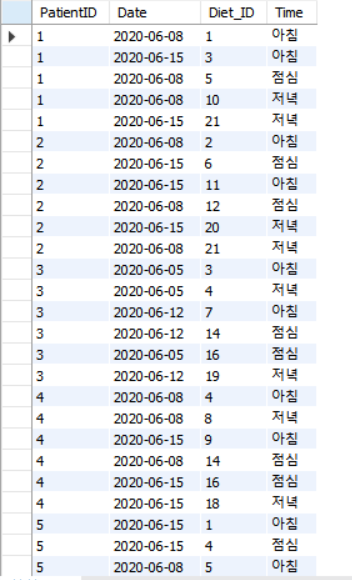
⑥ ‘약성분’ 테이블 생성 및 데이터 삽입



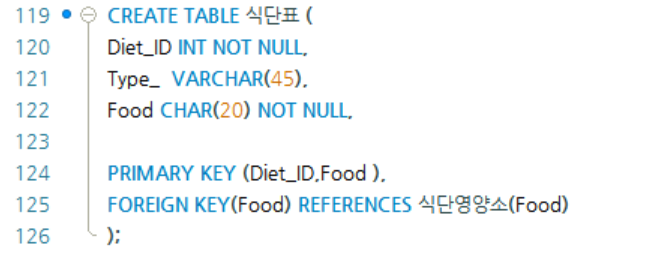


⑦ ‘식단’ 테이블 생성 및 데이터 삽입





⑧ ‘식단표’ 테이블 생성 및 데이터 삽입

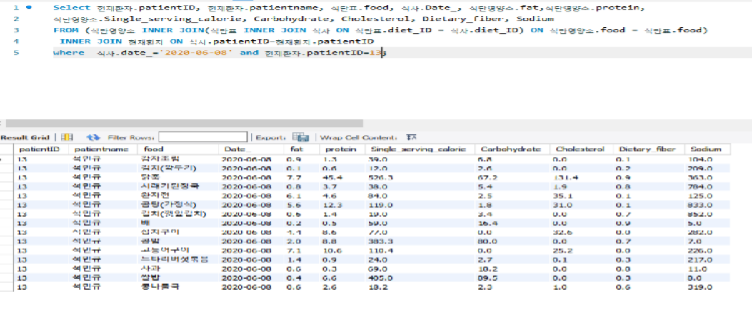


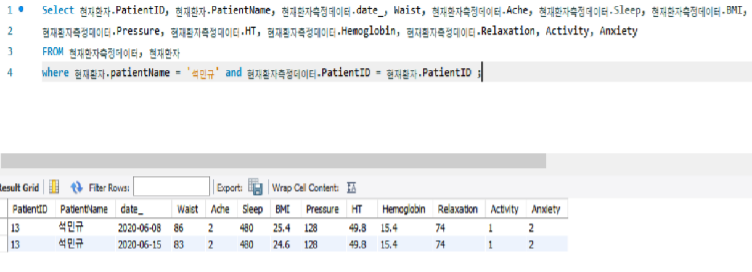
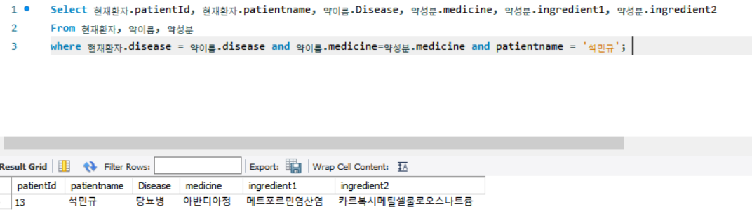


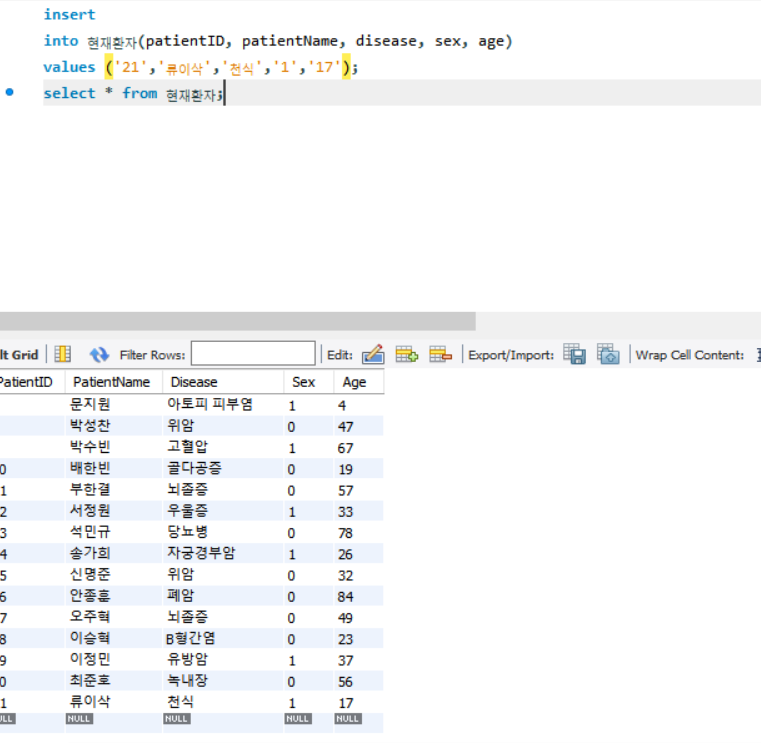
**<DML을 활용한 데이터 조작>**

‘디지털 병간호 서비스’ 시스템을 구현하기 위해서는 가지고 있는 테이블 데이터에서 필요한 정보를 추출해야 한다. DML언어를 활용하여 필요한 데이터를 조작해 보았다.

1. 석민규환자가 하루 동안 먹은 식단 데이터와 영양정보 데이터 추출

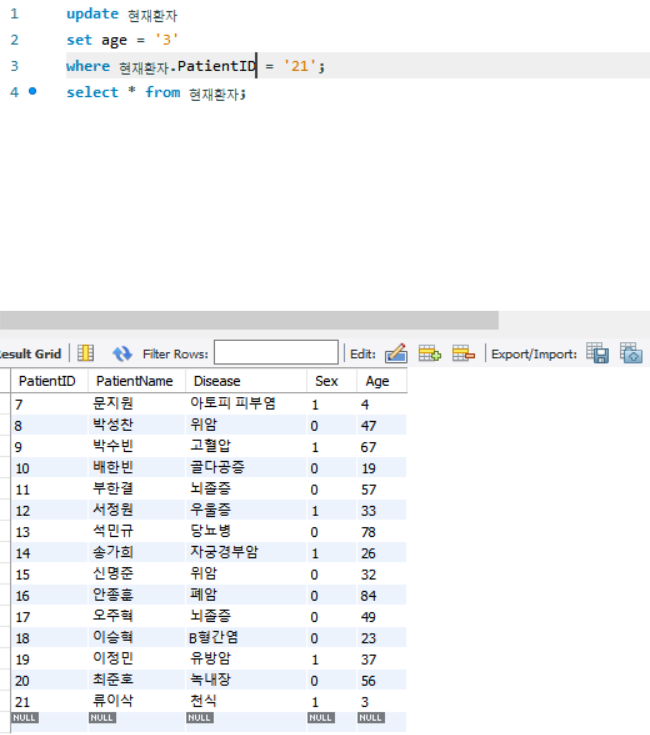


1. 석민규 환자의 일주일 전과 일주일 후의 건강상태 측정 데이터 추출
2. 석민규 환자의 질병에 따라 복용하는 약 품명 데이터와 약 성분 데이터 추출
3. 새로운 환자의 정보를 INSERT해 보았다.



1. 현재 환자의 개인정보를 수정해 보았다.

현재 환자의 개인정보를 수정해 보았다.



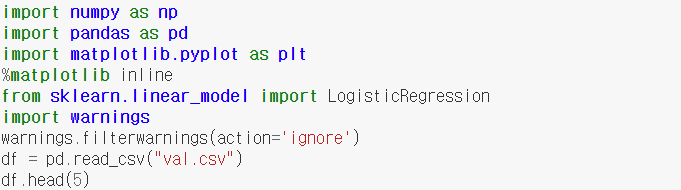
1. **Python(jupyter)을 통한 알고리즘 구현**

**로지스틱 회귀 분석을 이용한 질병 발생 예측**

\*\*로지스틱 회귀 - 선형 회귀 방식을 분류에 적용한 알고리즘으로, 주로 이진 분류에 사용됨.

1. **이전 환자 데이터를 이용한 질병 발생 예측**
2. **필요한 모듈을 불러온 후 분석에 필요한 데이터를 불러온다.**

**(분석할 때 용이하기 위해 속성 명을 한글로 사용했다.)**





* 고혈압 ~ C형 간염까지의 변수들은 환자가 질병에 걸렸는지 걸리지 않았는지를 나타낸다. 질병에 걸리면 1, 걸리지 않으면 0이 입력되어 있다.
* 신장 ~ 혈소판까지의 변수들은 환자 개인 특성에 대한 데이터이다.

1. **결측 치 제거**





* 데이터 분석하기에 앞서 결측 치를 찾아내서 제거하였다.

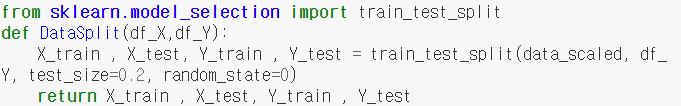
1. **필요한 작업들의 함수화**

* 로지스틱 회귀 분석에 필요한 작업들을 함수화 한다.

1. 데이터 전처리 함수 - 로지스틱 회귀분석에는 데이터 전처리가 필요하여 DataScaler이라는 전처리 함수를 생성하여 평균이 0, 분산이 1인 데이터 분포로 변환한다.

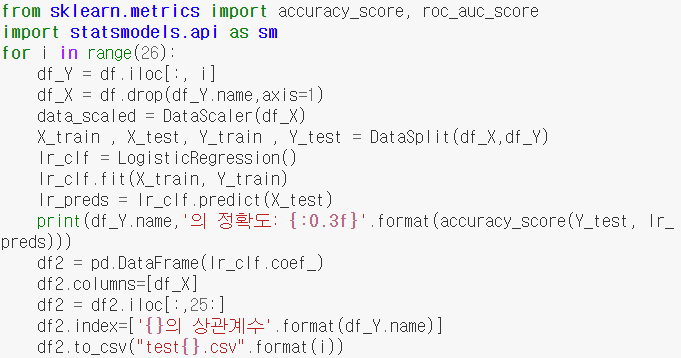


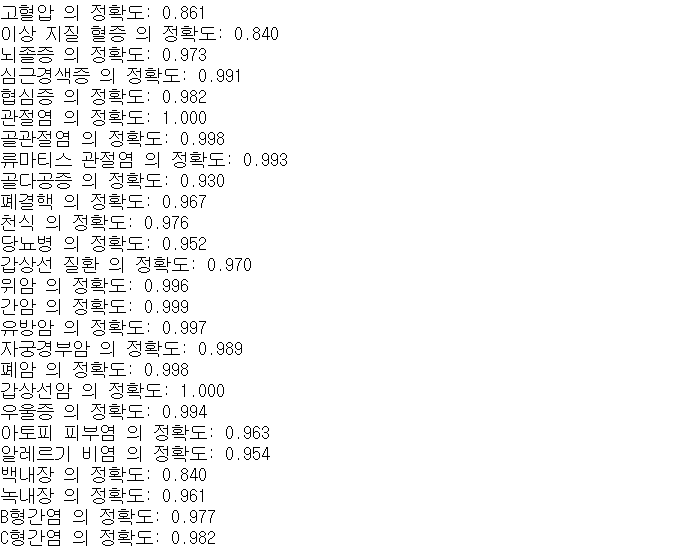
1. 데이터 스플릿 함수 – 학습 데이터와 테스트 데이터를 나누는 함수, 학습 데이터를 80%, 테스트 데이터를 20%로 설정하였다.



1. **로지스틱 회귀 분석**

* 각종 질병 발생 여부를 종속 변수로, 나머지 변수들은 독립변수로 설정한 후, 각종 질병에 대한 발생 가능성을 분석한다.
* 각각의 질병 발생 예측 분석의 정확도를 확인한다.
* 그 후, 환자 특성에 대한 변수들의 상관계수를 따로 추출하여 csv파일로 저장한다.





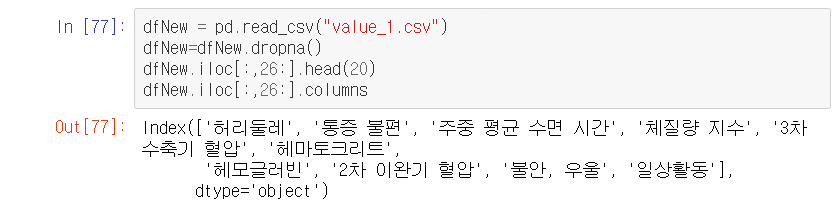
1. **상관계수 정리**

* 각 상관계수들의 상위 30%만 색깔을 다르게 하였다.





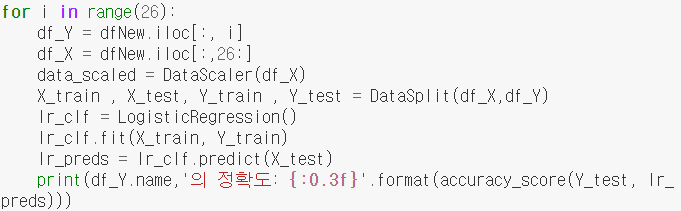
* 그 중 가장 많이 중복된 변수 중 상위 10개만 선택하여 새로운 환자 개인 특성에 대한 데이터를 만들었다.
* 선택한 변수들은 아래와 같다.

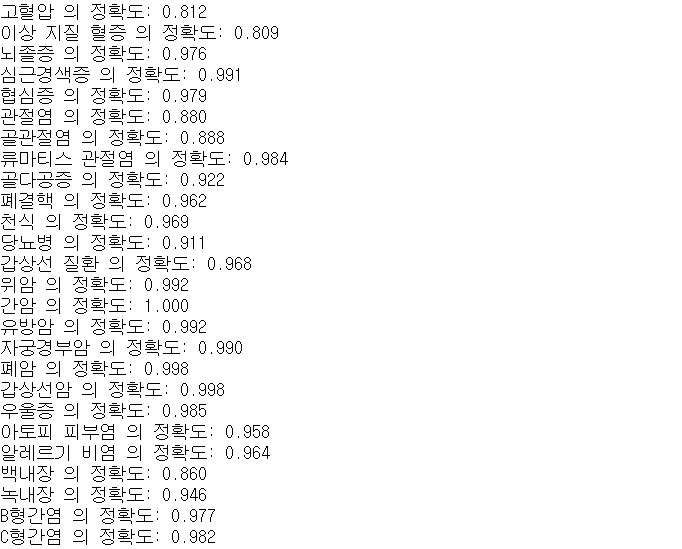




1. **정리된 상관계수를 이용한 질병 발생 예측**

* 선택한 변수들을 독립변수로 설정하여 다시 한번 질병 발생 가능성에 대한 로지스틱 회귀 분석을 진행하였다.





* 대부분 90% 이상의 정확도를 가지는 예측 모델이 생성되었다.

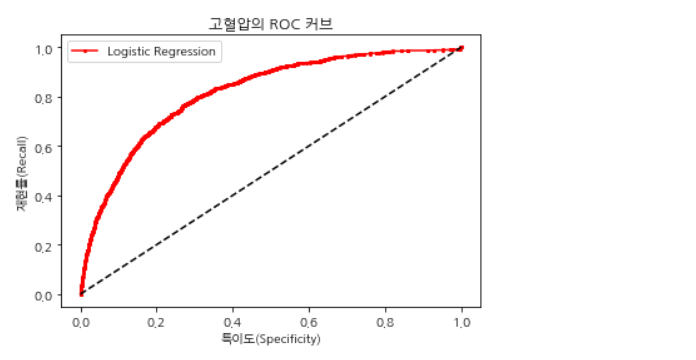
1. **모델의 신뢰도 평가**

* **모든 모델의 신뢰도를 알아보는 것은 많은 시간이 걸리므로 예시로 ‘고혈압’과 ‘위암’ 모델의 신뢰도를 알아보았다.**

1. **고혈압 모델의 정확도와 신뢰도**









1. **위암의 모델의 정확도와 신뢰도**

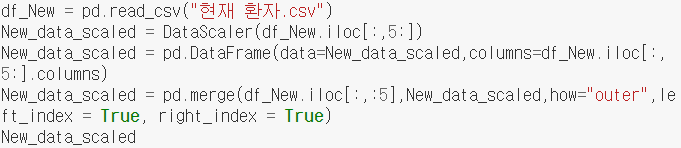


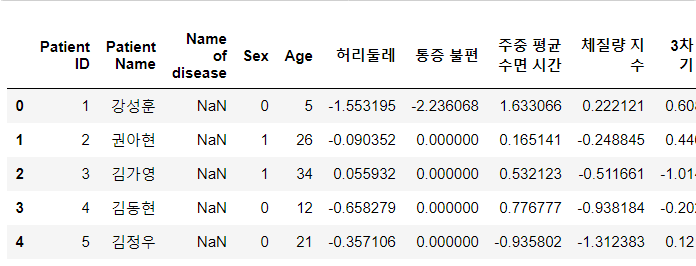


* **모든 모델이 70%이상의 신뢰도를 가지고 있다.**

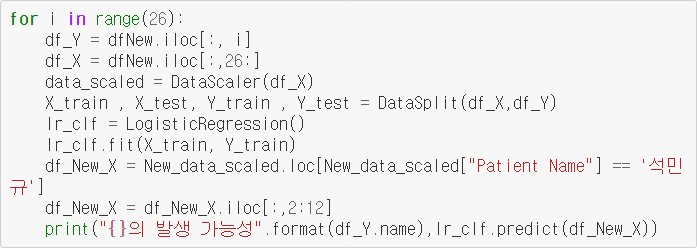
1. **현재 환자 데이터를 이용한 질병 발생 예측**

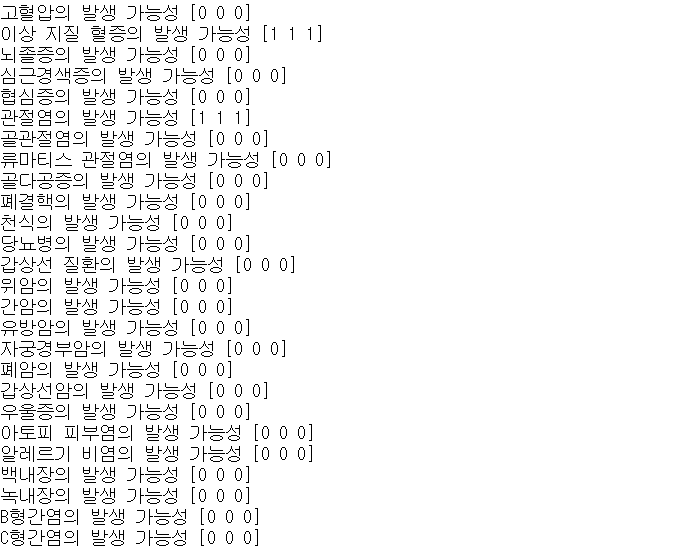
* 현재 환자 데이터를 위 예측 모델에 적용하였다.
* 예측을 위하여 환자 개인 특성에 대한 값 들은 전처리 하였다.



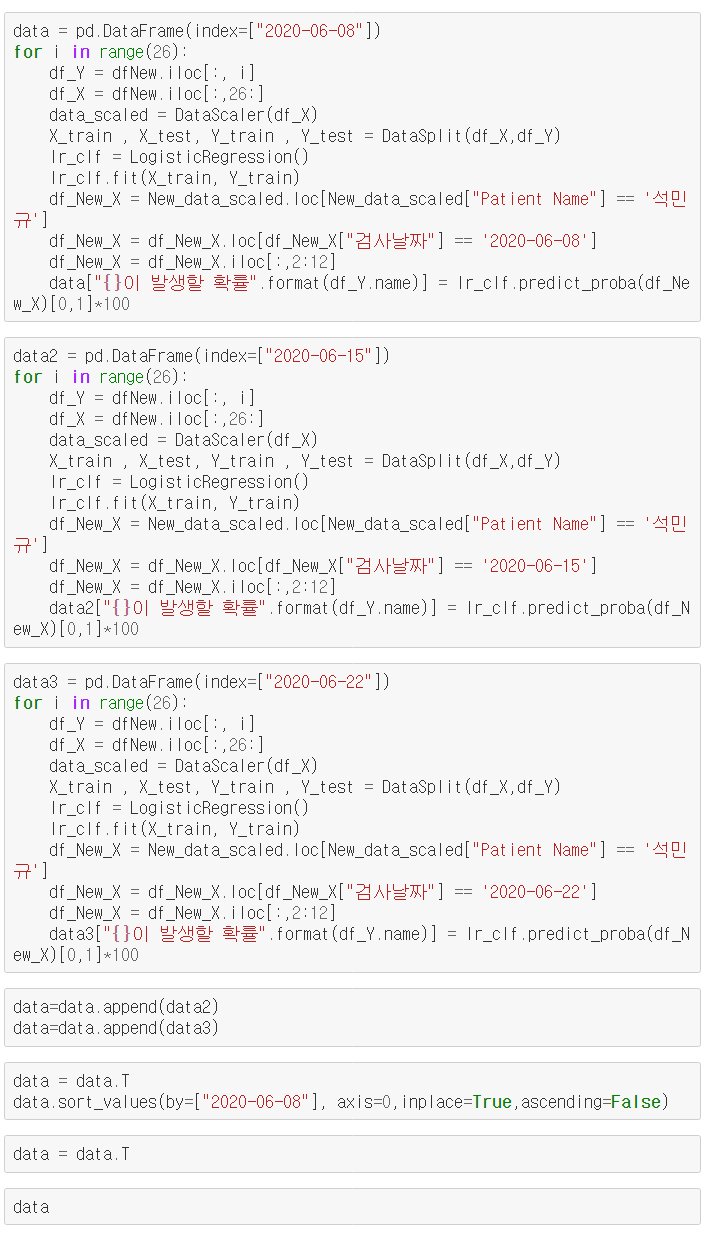


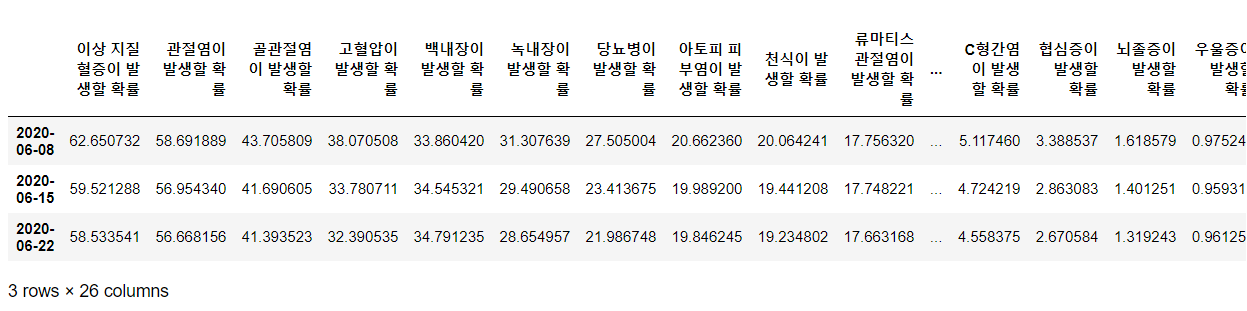
* 예시로 ‘석민규’ 환자의 질병 발생 가능성을 예측하였다.



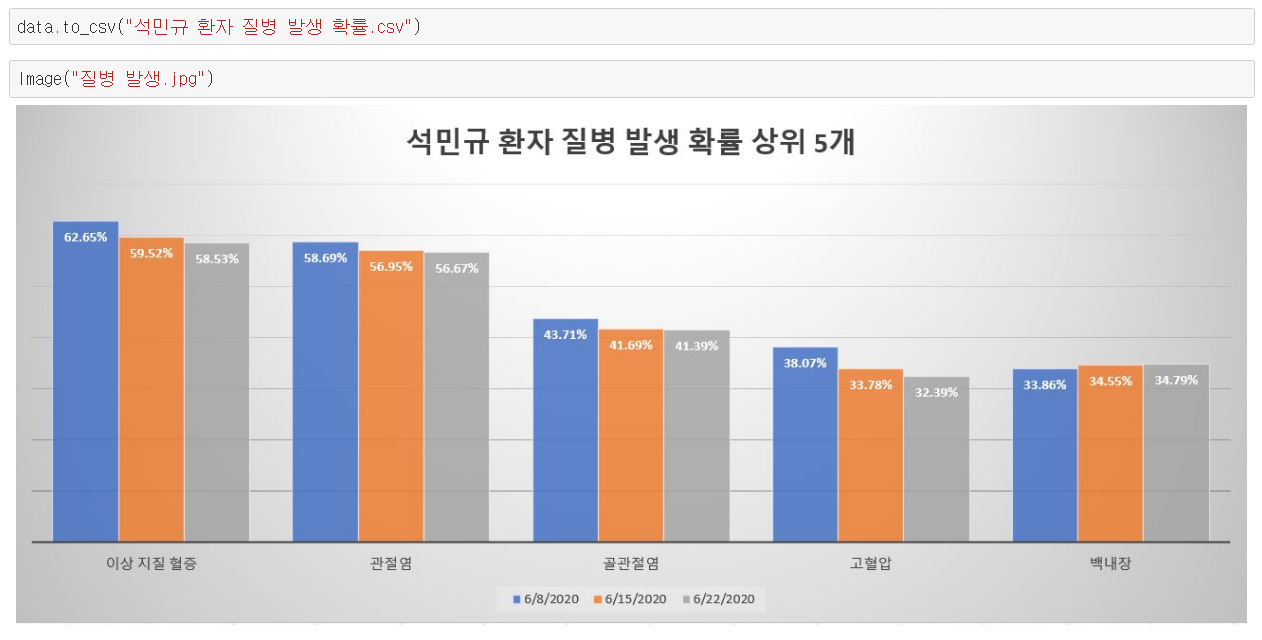


* 숫자가 0인 경우 질병이 발생하지 않을 확률이 더 큰 것이고, 숫자가 1인 경우 질병이 발생할 확률이 더 큰 것이다.
* 각종 질병의 발생확률을 표로 만들고, 확률이 큰 순서에서 낮은 순서로 내림차순 정렬하였다.



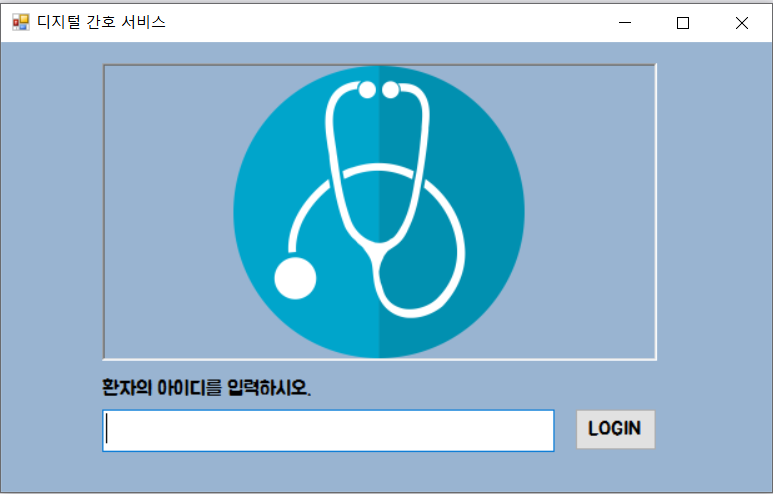


* 확률 상위 5개의 변화에 대해 알아보았다.



* 대부분의 발생 확률이 점차 감소하는 추세를 보인다.
* 따라서, ‘석민규’ 환자의 상태는 호전되고 있다고 할 수 있다.

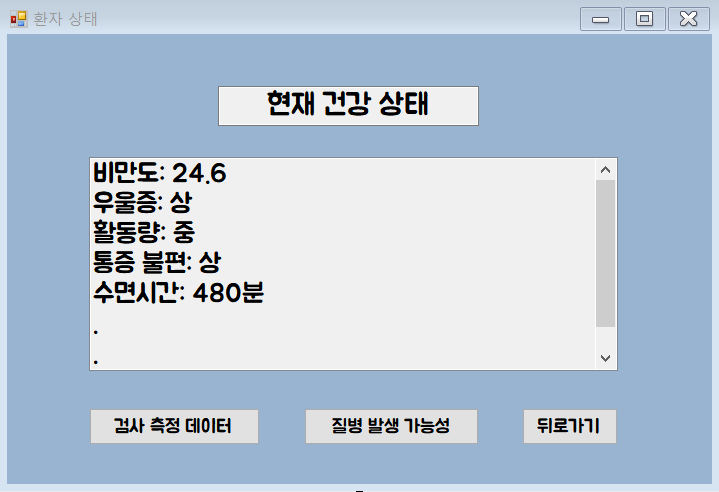
**(3) C# Windows Forms을 이용한 시각화**



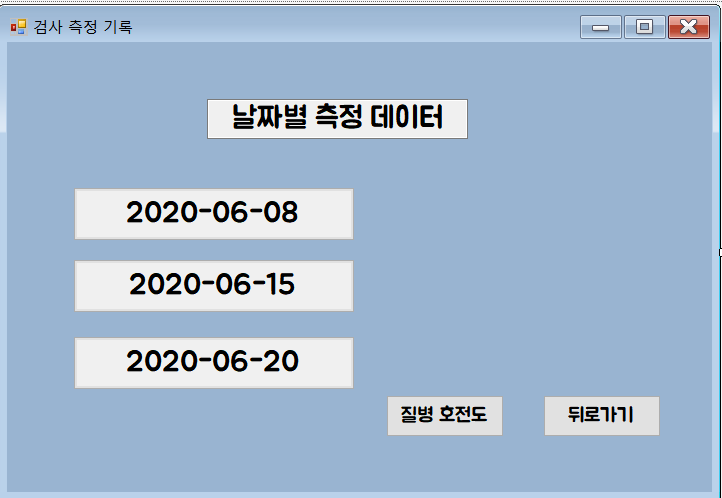
* 프로그램의 시작 화면이다.



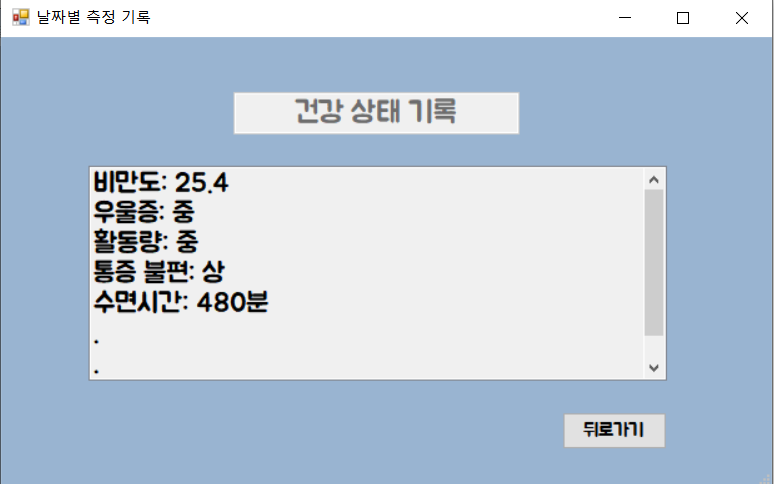
* 석민규 환자의 나이, 성별, 질병명과 같은 기본 정보를 표시해 준다.

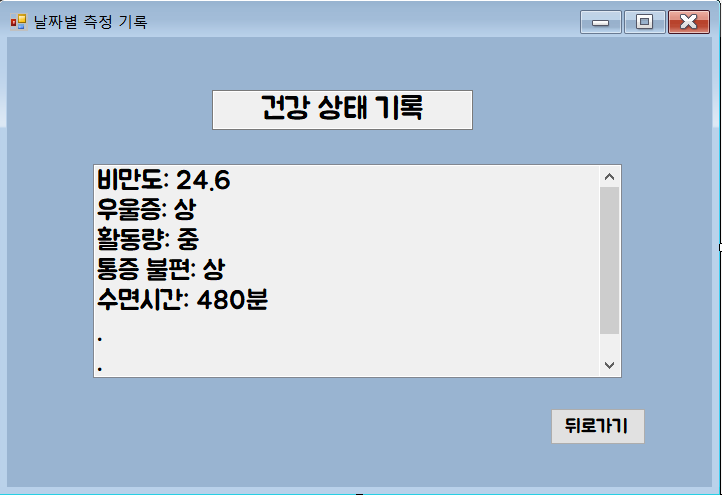


* 질병 간호에 들어오면 환자의 가장 최근에 검사한 정보들이 표시되어진다.



* 검사 측정 데이터 버튼에 들어오게 되면, 검사 날짜마다의 정보를 확인할 수 있다.



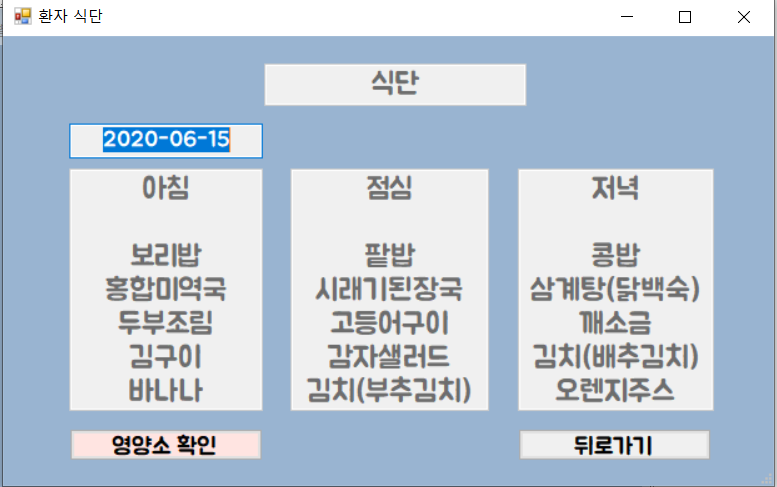


* 각각의 날짜에 대한 환자 건강상태 정보이다.

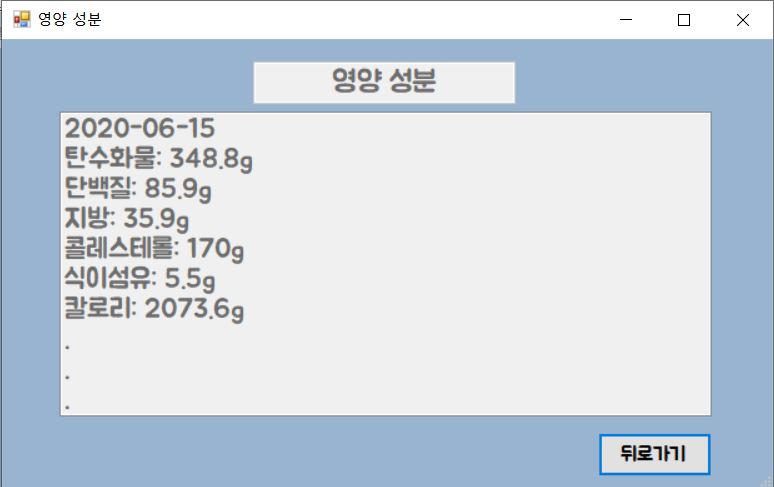




* 날짜별 환자의 건강상태 정보를 보고, 환자의 질병 확률을 표현해 주는 페이지이다.



* 식단 영양소에 들어오게 되면 날짜별로 환자가 먹은 음식을 확인할 수 있다.



* 영양소 확인으로 들어가게 되면, 해당 날짜에 환자가 섭취한 영양소 정보를 확인할 수 있다.



* 홈 화면에서 처방약으로 들어가게 되면 환자가 복용하고 있는 약과 그 약에 대한 성분을 확인할 수 있다.

**3.결론**

서류를 직접 작성하고 관리해야 했던 이전과는 달리, 많은 데이터를 효과적으로 저장하고 관리할 필요가 있는 현재에는 빅데이터를 관리하는 법에 대한 관심이 급증되었으며, 이는 곧 경쟁력이 되고 있다. 이러한 환경 속에서 데이터베이스는 많은 데이터를 저장할 수 있으며, 검색과 갱신의 효율은 높이고, 여러 사람에 의해 공유되어 사용할 수 있다는 점에서 매우 중요한 역할을 하고 있다. 이러한 데이터베이스는 산업공학도로서 반드시 배워야 할 분야이며, 본 과목을 통해 이론을 배우고, 실습을 통해 학습할 수 있었다.

이를 적용하여 우리는 질병 및 건강에 대한 데이터를 가지고 “디지털 간호 서비스”라는 주제를 선정해 “병원 방문없이 입원 환자의 건강 상태를 확인하고 간호할 수 있는 시스템”에 대해 고민해보았고, 이를 토대로 요구사항 명세서를 작성하였다. 요구 사항들을 모두 만족시키면서 데이터를 효과적으로 사용할 수 있도록 해결 알고리즘을 구상하였고, 이를 바탕으로 ER 다이어그램을 도출해냈다. 이어서 MySQL과 워크벤치를 이용하여 관계형 데이터 모델과 DB를 설계하고 구축하고자 하였고, SQL 명령어를 통해 사용자가 원하는 정보만을 질의할 수 있도록 하였다. 구축된 DB를 통해 원하는 정보를 쉽게 확인할 수 있도록 시각화를 하였고, 편리하게 정보를 제공받을 수 있도록 하였다. 이러한 과정을 통해 많고 많은 데이터 중에서 사용자가 원하는 정보만을 편리하고 효율적으로 제공받을 수 있도록 DB를 설계하고자 하였고, 개념적, 논리적 설계 및 정규화를 통해 데이터 관리 및 추출을 효과적으로 할 수 있는 시스템을 구축하였다.

본 프로젝트를 통해 문제를 정의하고 해결 알고리즘을 구상한 뒤에, 이를 빅데이터를 활용하여 데이터베이스를 구축함으로써 문제를 해결하였다는 점에서 데이터베이스의 중요성과 활용도를 느낄 수 있었다. 데이터베이스를 설계하는 관점에서 유용한 데이터를 추출하고, 원하는 데이터만 질의할 수 있도록 모델링을 하는 것은 상당히 중요하였으며, 이를 시각화를 통해 효과적으로 제공하고자 하였다. 그 과정속에서 수많은 정보들 사이의 관계를 파악하고 정의할 수 있었으며, 이는 사용자가 데이터를 조작하는 데에 효과적인 시스템을 구축하는 데에 도움이 되었다.