

# 병원 정보 DB 구축

2013312343 이상헌

## 1. 설명

- 병원 정보 시스템에 대한 DB를 설계 및 MySQL을 이용하여 컴퓨터 상에 구축한다.

## 2. DB 요구사항 분석 ( **Entity**, **Attribute**, **Constraint**, **Relationship type** 로 표기. )

### 1) 부서 (Department)

- 병원은 여러 종류의 **부서**를 가진다.
- 각 부서는 고유한 부서명을 통해 식별된다.
- 병원은 각 **부서**에 대해 **부서명**, **위치 번호** 및 **전화번호**를 저장한다.
- 한 부서는 여러 명의 의사와 간호사를 포함한다.
- 부서에 속한 의사들 중 한 명의 의사가 그 부서의 책임자가 된다.

### 2) 의사 (Doctor)

- 의사는 고유한 사원 번호를 통해 식별된다.
- 병원은 **의사**의 **이름**, **사원 번호**, **재직 년 수** 및 **가능한 진료 요일**을 저장한다..
- 한 의사는 여러 환자의 진료를 보거나 수술을 집도할 수 있다.
- 한 의사는 자신이 속한 부서의 책임자가 되거나 되지 않을 수 있다.

### 3) 간호사 (Nurse)

- 간호사는 고유한 사원 번호를 통해 식별된다.
- 병원은 **간호사**의 **이름**, **사원 번호**, **재직 년 수**를 저장한다.
- 간호사는 여러 병실 및 환자들을 관리한다.

#### 4) 환자 (Patient)

- 환자는 고유한 주민등록번호를 통해 식별된다.
- 병원은 환자의 이름, 주민등록번호, 나이, 성별, 보험 코드 등을 저장한다.
- 환자는 여러 수술을 받거나, 의사에게 진료를 받을 수 있다.
- 환자는 한 병실에 입원할 수 있고, 간호사에 의해 관리된다.

#### 5) 수술 (Surgery)

- 수술은 고유한 수술 번호를 통해 식별된다.
- 병원은 특정 수술에 대해 수술명과 수술 번호를 저장한다.
- 특정 수술에 대한 수술 날짜, 수술 시간 및 환자의 상태가 주어진다.
- 수술은 한 환자에게 행해지며, 여러 의사에 의해 집도 된다.

#### 6) 병실 (Room)

- 병실은 고유한 병실 번호를 통해 식별된다.
- 병원은 병실의 병실 번호 및 병실의 전화번호를 저장한다.
- 병실은 한 간호사에 의해 관리되며, 여러 환자들을 수용한다.

3. 각 Entity에 대한 세부 Attributes ( **Entity**, Primary key 로 표기. )

1) 부서(DEPARTMENT) = 부서명(dept-name), 위치 번호(location-num), 전화번호(telephone)

2) 의사(DOCTOR) = 사원 번호(doctor-id), 이름(name), 재직 년 수(work-year), 진료 요일(treatment-day)

3) 간호사(NURSE) = 사원 번호(nurse-id), 이름(name), 재직 년 수(work-year)

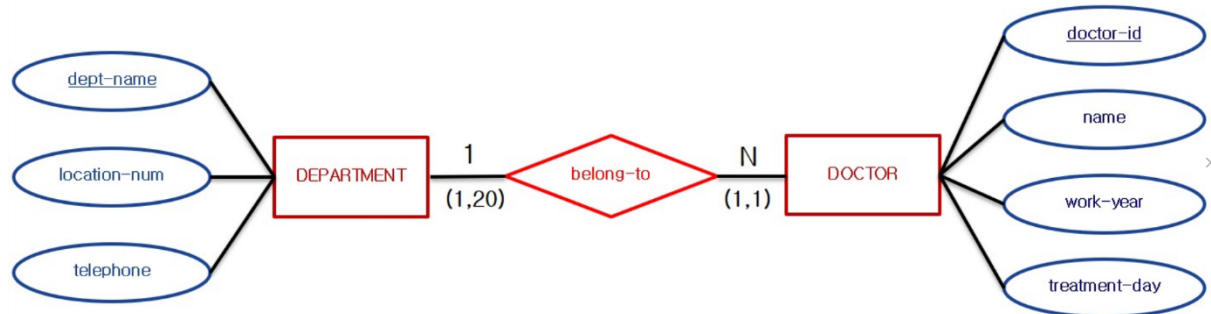
4) 환자(PATIENT) = 주민등록번호(RRN), 환자 이름(patient-name), 나이(age), 성별(sex), 보험 코드(insurance-code)

5) 수술(SURGERY) = 수술 번호(surgery-num), 수술명(surgery-name)

6) 병실(ROOM) = 병실 번호(room-num), 병실 전화번호(room-telephone)

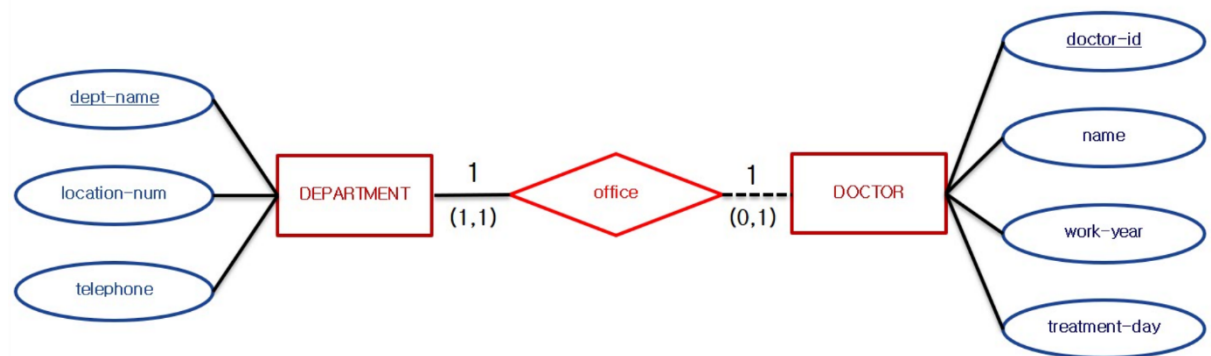
#### 4. 각 Entity에 대한 Relationship

##### 1) belong-to Relationship



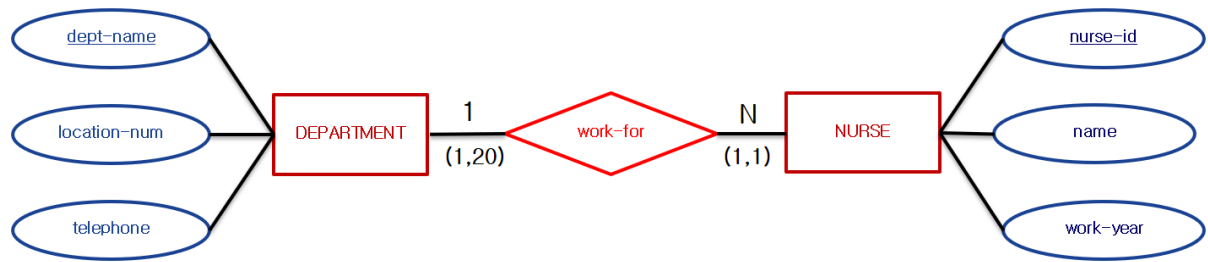
- 부서(DEPARTMENT)와 의사(DOCTOR) 사이 소속 관계를 나타낸다.
- 한 명의 의사는 반드시 단 하나의 부서에 소속되어 있어야 하고, 한 부서는 최대 20명의 의사를 보유할 수 있다.
- 부서 : 의사 = 1 : N의 관계.

##### 2) Office(부서의 책임자) Relationship



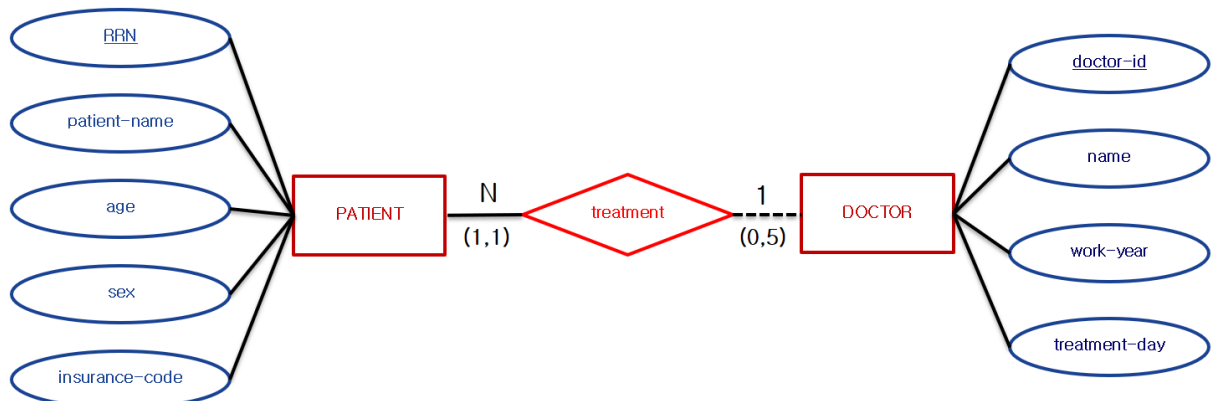
- 부서(DEPARTMENT)와 의사(DOCTOR) 사이 책임 관계를 나타낸다.
- 특정 부서에 속한 한 명의 의사는 그 부서의 책임자가 될 수 있으며, 각 부서는 반드시 단 한 명의 책임자가 되어야 한다.
- 부서 : 의사(책임자) = 1 : 1의 관계.

### 3) work-for Relationship



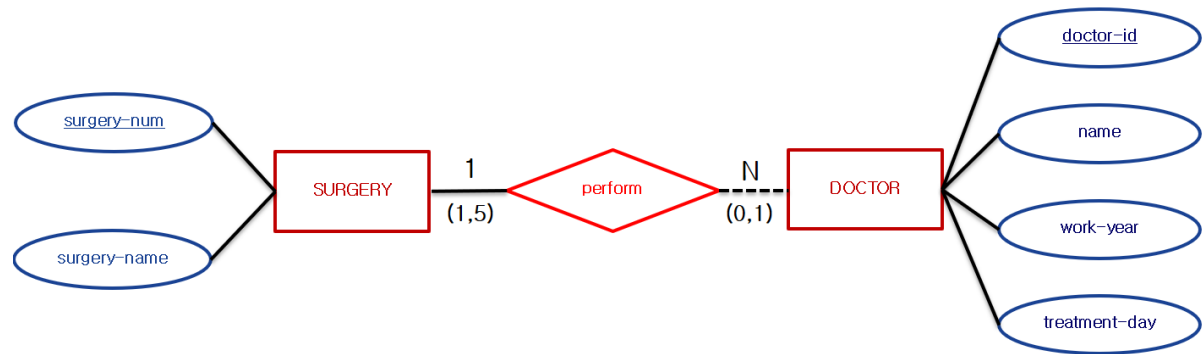
- 부서(DEPARTMENT)와 간호사(NURSE) 사이 소속 관계를 나타낸다.
- 한 명의 간호사는 반드시 단 하나의 부서에 소속되어 있어야 하고, 한 부서는 최대 20명의 간호사를 보유할 수 있다.
- 부서 : 간호사 = 1 : N의 관계.

### 4) treatment Relationship



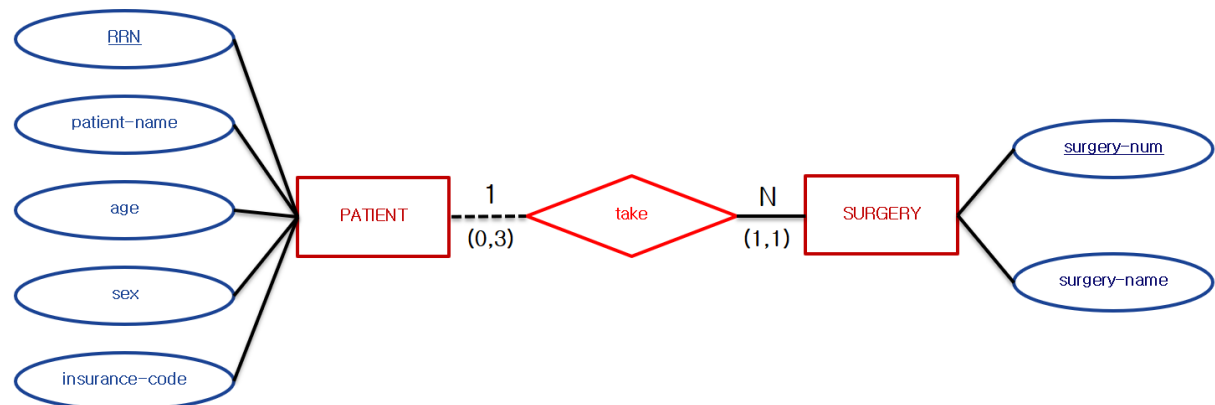
- 환자(PATIENT)와 의사(DOCTOR) 사이 진료 관계를 나타낸다.
- 한 명의 환자는 반드시 단 한 명의 의사에게 진료를 보아야 하고, 한 의사는 최대 5명의 환자의 진료를 볼 수 있다.
- 환자 : 의사 = N : 1의 관계.

## 5) perform Relationship



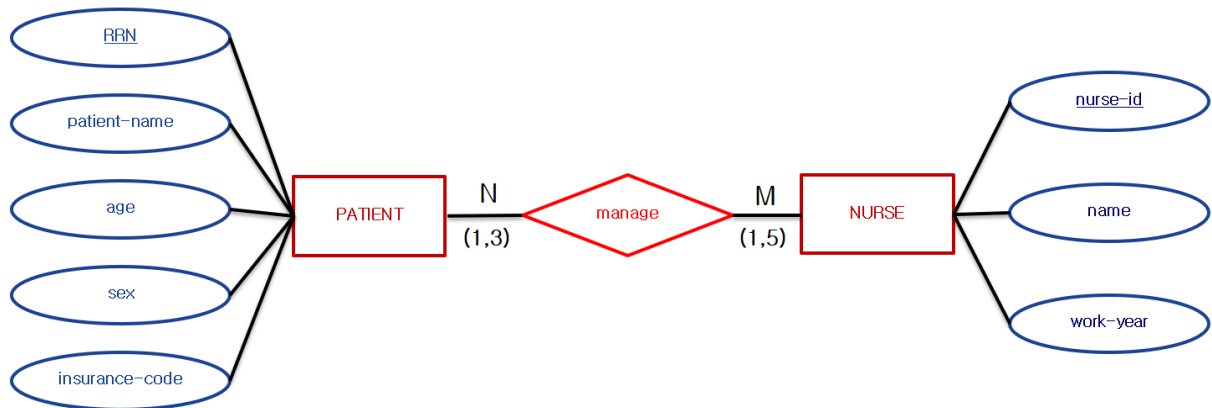
- 수술(SURGERY)과 의사(DOCTOR) 사이 집도 관계이다.
- 한 의사는 최대 1개의 수술을 집도할 수 있고, 한 수술은 최대 5명 이상의 의사에게 집도 되어야 한다.
- 수술 : 의사 = 1 : N의 관계.

## 6) take Relationship



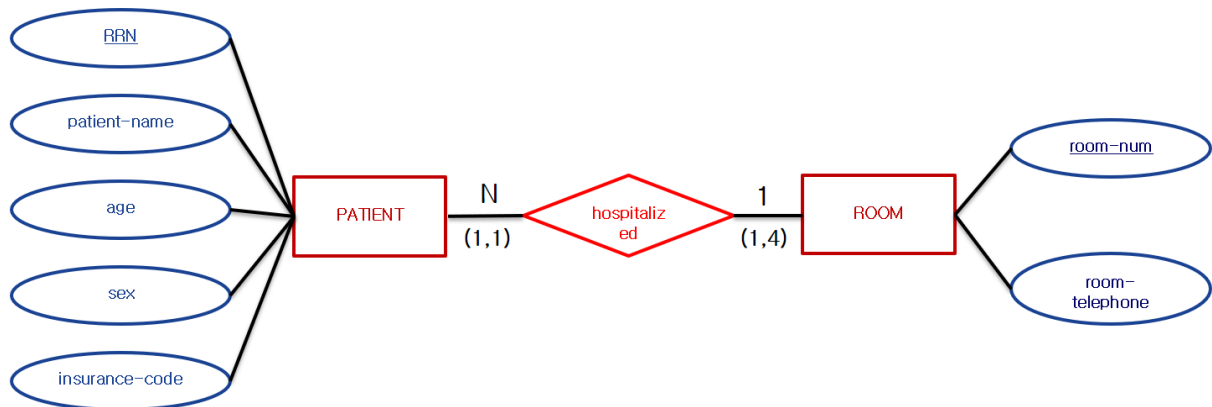
- 환자(PATIENT)와 수술(SURGERY) 사이 (수술을)받는 관계이다.
- 한 수술은 반드시 단 한 명의 환자를 대상으로 해야 하고, 한 환자는 최대 3번의 수술을 받을 수 있다.
- 환자 : 수술 = 1 : N의 관계.

### 7) manage Relationship



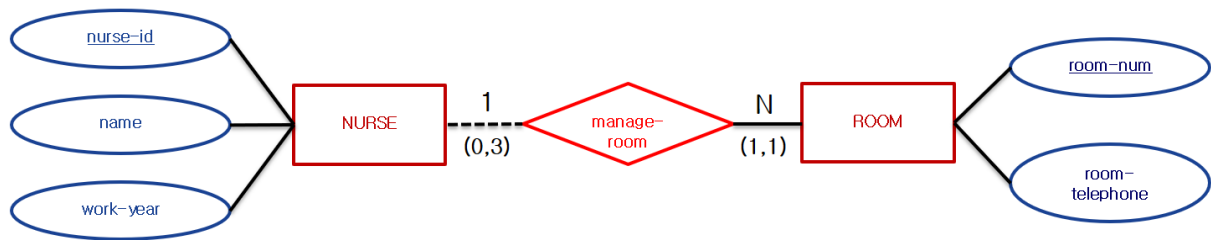
- 환자(PATIENT)와 간호사(NURSE) 사이 관리 관계이다.
- 한 명의 환자는 최대 5명의 간호사에게 관리 받아야 되며, 한 명의 간호사는 최대 3명의 환자를 관리해야 한다.
- 환자 : 간호사 = N : M의 관계.

### 8) hospitalized Relationship



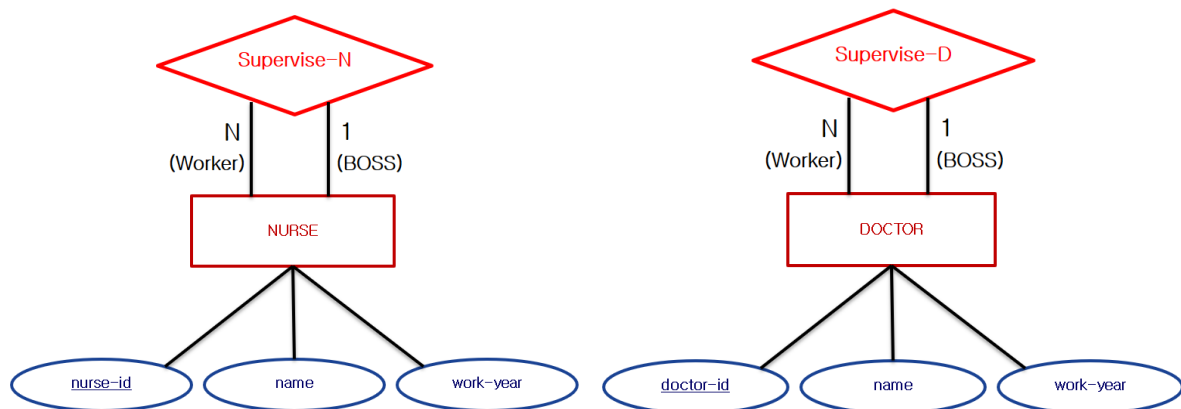
- 환자(PATIENT)와 병실(ROOM) 사이 입원 관계이다.
- 한 명의 환자는 반드시 단 하나의 병실에 입원해야 하고, 한 개의 병실은 최대 4명의 환자를 입원시키고 있어야 한다. 비어있는 병실은 존재하면 안된다.
- 환자 : 병실 = N : 1의 관계.

## 9) manage-room Relationship



- 간호사(NURSE)와 병실(ROOM) 사이 관리 관계이다.
- 한 간호사는 최대 3개의 병실을 관리할 수 있고, 한 병실은 반드시 단 한 명의 간호사에게 관리 되어야 한다.
- 간호사 : 병실 = 1 : N의 관계.

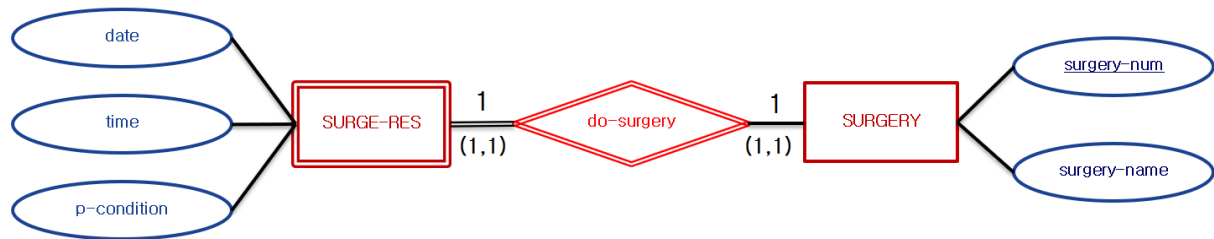
## 10) Recursive Relationship of NURSE and DOCTOR



- 한 명의 간호사는 반드시 단 한 명의 상사가 있고, 여러 명의 부하 직원을 가진다.
- 한 명의 의사는 반드시 단 한 명의 상사가 있고, 여러 명의 부하 직원을 가진다.
- 간호사 : 상사(간호사) = N : 1의 Recursive 관계.
- 의사 : 상사(의사) = N : 1의 Recursive 관계.

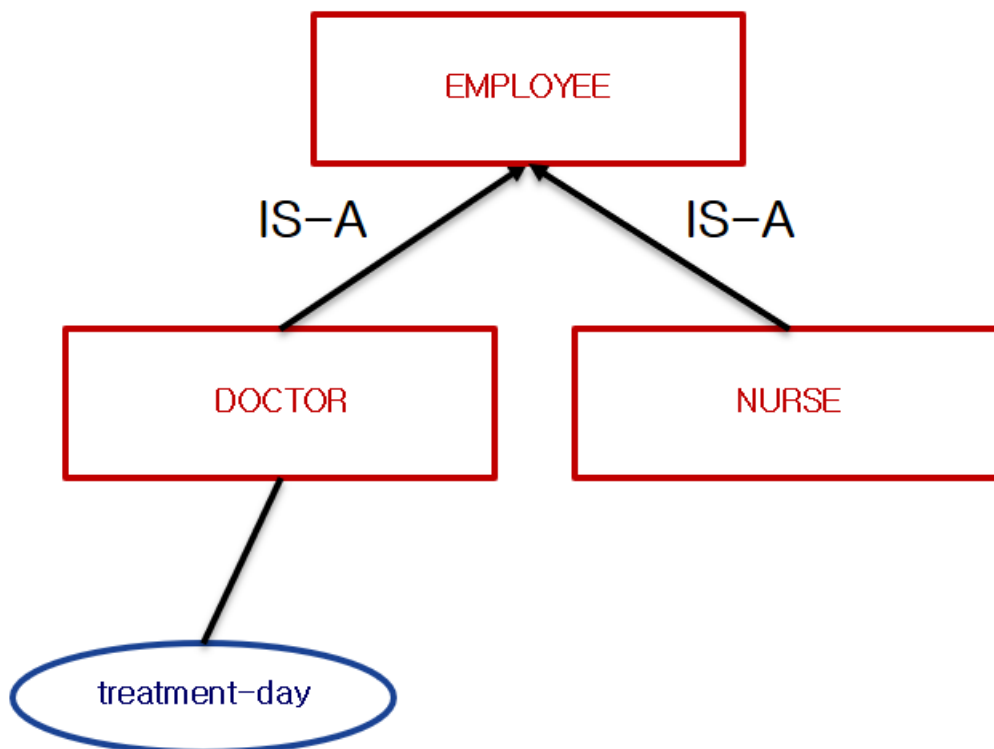


### 11) do-surgery Relationship



- 수술(SURGERY)과 수술결과(SURGE-RES) 사이 관계를 나타낸다.
- SURGE-RES는 수술의 결과에 대한 상세 정보를 담고 있는 entity로, Weak Entity Type이다.
- 한 수술에 대해 정확히 하나의 수술 결과 데이터를 가지고 있어야 한다.
- 수술 : 수술결과 = 1 : 1의 관계.

### 12) IS-A Relationship of NURSE and DOCTOR.



- 의사(DOCTOR)와 간호사(NURSE)의 IS-A 관계를 나타낸다.
- EMPLOYEE는 사원의 기본 정보를 담은 entity로, id, name, work-year를 attribute로 가지고 있다.
- DOCTOR와 NURSE는 EMPLOYEE를 IS-A Relationship으로 상속받는 entity이다.
- DOCTOR에는 그 entity만의 attribute인 treatment-day가 존재한다.

The diagram is a hand-drawn Entity-Relationship (ER) model for a hospital database. It includes the following components:

- Entities and Attributes:**
  - DOCTOR:** Attributes are doctor-id, name, work-year, and treatmentday.
  - NURSE:** Attributes are nurse-id, name, and work-year.
  - EMPLOYEE:** An abstract entity with no attributes.
  - DEPT:** Attributes are dept\_name, location-num, and telephone.
  - SURGERY:** Attributes are surgery-num and surgery-name.
  - PATIENT:** Attributes are RRN (Primary Key), patient-name, sex, age, and insurance-code.
  - ROOM:** Attributes are room-num and room-telephone.
  - SURGE-RES:** Attributes are date, time, and p-condition.
- Relationships and Cardinalities:**
  - supervise-D:** Connects DOCTOR and DEPT. Cardinalities: N (Worker) to 1 (Boss).
  - office:** Connects DOCTOR and DEPT. Cardinalities: 1 (DOCTOR) to 1 (DEPT).
  - belong-to:** Connects DOCTOR and DEPT. Cardinalities: N (DOCTOR) to 1 (DEPT).
  - work-for:** Connects NURSE and DEPT. Cardinalities: 1 (DEPT) to N (NURSE).
  - supervise-N:** Connects NURSE and DEPT. Cardinalities: N (Worker) to 1 (Boss).
  - IS-A:** Connects DOCTOR, NURSE, and EMPLOYEE, indicating inheritance.
  - treatment:** Connects DOCTOR and PATIENT. Cardinalities: 1 (DOCTOR) to N (PATIENT).
  - Manage:** Connects NURSE and PATIENT. Cardinalities: N (NURSE) to N (PATIENT).
  - manage-room:** Connects NURSE and ROOM. Cardinalities: 1 (NURSE) to N (ROOM).
  - perform:** Connects DOCTOR and SURGERY. Cardinalities: 1 (DOCTOR) to N (SURGERY).
  - take:** Connects PATIENT and SURGERY. Cardinalities: 1 (PATIENT) to N (SURGERY).
  - hospitalized:** Connects PATIENT and ROOM. Cardinalities: 1 (PATIENT) to N (ROOM).
  - do-surgery:** Connects SURGERY and SURGE-RES. Cardinalities: 1 (SURGERY) to N (SURGE-RES).

6. Relational Schema 작성 ( Primary key, Foreign key 로 표기. 겹치는 경우 PK를 따로 표기.)

- 모든 부서는 단 한 명의 책임자를 가지고 있다.

- FD : dept-name  $\rightarrow$  { location-num, telephone, officer-id }

(2) DOCTOR ( doctor-id, dept-name, super-d-id, surgery-num ) PK = { doctor-id }

- DOCTOR entity는 EMPLOYEE entity의 모든 attribute를 상속 받는다. 상속 관계는 DOCTOR entity의 PK인 doctor-id를 이용한다. 즉, doctor-id는 상속에서의 FK로 작용한다.

- DOCTOR : DEPT = N : 1 Relationship 이므로 1-side의 PK ( dept-name )를 N-side의 FK로써 attribute에 추가한다.

- DOCTOR의 supervise-D Relationship은 worker : boss = N : 1 relationship이므로 1-side의 PK ( doctor-id )를 N-side의 FK ( super-d-id )로써 attribute에 추가한다.

- DOCTOR : SURGERY = N : 1 Relationship 이므로 1-side의 PK ( surgery-num )를 N-side의 FK로써 attribute에 추가한다.

- FD : doctor-id -> { dept-name, super-d-id, surgery-num }

(3) NURSE ( nurse-id, dept-name, super-n-id ) PK = { nurse-id }

- NURSE : DEPT = N : 1 Relationship 이므로 1-side의 PK ( dept-name )를 N-side의 FK로써 attribute에 추가한다.

- NURSE의 supervise-N Relationship은 worker : boss = N : 1 relationship이므로 1-side의 PK ( nurse-id )를 N-side의 FK ( super-n-id )로써 attribute에 추가한다.

- FD : nurse-id -> { dept-name, super-n-id }

(4) EMP ( id, name, work-year )

- EMP entity는 DOCTOR와 NURSE entity가 상속받는 attribute가 존재하는 entity로, PK인 id를 통해 DOCTOR와 NURSE에게 모든 attribute를 상속한다.

- FD : id -> { name, work-year }

(5) MANAGE-N-P ( nurse-id, patient-RRN ) PK = { nurse-id, patient-RRN }

- NURSE와 PATIENT의 manage 관계는 NURSE : PATIENT = M : N Relationship이다. 따라서 Relational Schema로 바꿀 때에는 이 관계를 위한 새로운 entity를 생성해야 한다. 생성된 entity에는 각각 기존 entity에서의 PK를 FK로써 attribute에 추가하고, M : N 관계이므로 PK는 두 attribute 모두가 된다.

- FD : { nurse-id, patient-RRN } -> { nurse-id, patient-RRN }

(6) PATIENT ( RRN, patient-name, age, sex, insurance-code, doctor-id, room-num )

- PATIENT : DOCTOR = N : 1 Relationship 이므로 1-side의 PK ( doctor-id )를 N-side의 FK로써 attribute에 추가한다.

- PATIENT : ROOM = N : 1 Relationship 이므로 1-side의 PK ( room-num )를 N-side의 FK로써 attribute에 추가한다.

- FD : RRN -> { patient-name, age, sex, insurance-code, doctor-id, room-num }

(7) ROOM ( room-num, room-telephone, nurse-id )

- ROOM : NURSE = N : 1 Relationship 이므로 1-side의 PK ( nurse-id )를 N-side의 FK로써 attribute에 추가한다.

- FD : room-num -> { room-telephone, nurse-id }

(8) SUREGERY ( surgery-num, surgery-name, patient-RRN )

- SURGERY : PATIENT = N : 1 Relationship 이므로 1-side의 PK ( RRN )를 N-side의 FK ( patient-RRN )로써 attribute에 추가한다.

- FD : surgery-num -> { surgery-name, patient-RRN }

(9) SURGE-RES ( surgery-num, date, time, p-condition ) PK = { surgery-num }

- SURGE-RES 는 Weak entity 이다.

- SURGERY : SURGE-RES = 1 : 1 Relationship 이므로 1-side의 PK ( surgery-num ) 를 가지고 와서 SURGE-RES의 PK로 사용한다.

- FD : surgery-num -> { date, time, p-condition }

## 7. Normalization 정의

• 각 Relational Schema가 Normal Forms ( 1NF, 2NF, 3NF, BCNF )를 만족하는지를 확인하고, 만족하지 않는 경우에 대해서 정규화를 통해 만족하도록 하는 작업이다. 아래는 각 Normal Forms의 정의이다.

• First Normal Form (1NF) : A relation R is 1NF if (1) each tuple must be identified by primary key; (2) each attribute must have atomic (= only one) value.

• Second Normal Form (2NF) : A relation R is 2NF if any non-prime attribute is not partially dependent of any key.

• Third Normal Form (3NF) : A relation R is 3NF if any non-prime attribute is not transitive dependent of any key. 즉, 모든 FD :  $X \rightarrow A$ 에 대해서 X가 super key 이거나, A가 prime attribute (key를 구성하는 attribute) 이면 3NF를 만족한다.

• Boyce Codd Normal Form (BCNF) : A relation R is BCNF if for every FD :  $X \rightarrow A$ , X is a super key.

## 8. Normalization 확인

(1) DEPT Table

• key = { dept-name }

• 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.

• 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.

• 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.

• BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.

## (2) DOCTOR Table

- key = { doctor-id }
- 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.
- 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.
- 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.
- BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.

## (3) NURSE Table

- key = { dept-name }
- 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.
- 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.
- 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.
- BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.

## (4) EMP Table

- key = { id }
- 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.
- 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.
- 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.

- BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.

#### (5) MANAGE-N-P Table

- key = { nurse-id, patient-RRN }
- 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.
- 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.
- 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.
- BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.

#### (6) PATIENT Table

- key = { RRN }
- 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.
- 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.
- 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.
- BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.

#### (7) ROOM Table

- key = { room-num }
- 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.
- 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.

- 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.
- BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.

#### (8) SURGERY Table

- key = { surgery-num }
- 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.
- 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.
- 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.
- BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.

#### (9) SURGE-RES

- key = { surgery-num }
- 1NF : PK에 의해 각 tuple이 구분되어진다. 또한 각 attribute는 atomic value 이다. 따라서 만족한다.
- 2NF : 모든 non-prime attribute들이 기본 key에 함수 부분 종속이 되지 않는다. (non-partially dependent) 따라서 만족한다.
- 3NF : 모든 non-prime attribute들이 기본 key에 함수 이행 종속이 되지 않는다. (not transitive dependent) 따라서 만족한다.
- BCNF : 모든 FD의 결정자(X)가 super-key이다. 따라서 만족한다.



## 9. SQL DDL 명령문 작성 ; Table 생성 및 tuple 추가

- CREATE SCHEMA 를 이용하여 hospital DB를 생성하였다.
- CREATE TABLE을 이용하여 Base Table을 생성하고, INSERT 및 UPDATE를 이용하여 각 Table에 tuple을 추가하였다.

### (1) EMP, DEPT, DOCTOR Table 생성

```
1 • CREATE SCHEMA HOSPITAL;|
2 • USE HOSPITAL;
3
4 • CREATE TABLE EMP(
5     id INT NOT NULL,
6     name CHAR(20),
7     work_year INT,
8     PRIMARY KEY (id)
9 );
10
11 • CREATE TABLE DEPT(
12     dept_name CHAR(20) NOT NULL,
13     location_num INT,
14     telephone CHAR(20),
15     PRIMARY KEY (dept_name)
16 );
17
18 • CREATE TABLE DOCTOR(
19     doctor_id INT NOT NULL,
20     dept_name CHAR(20),
21     super_d_id INT,
22     PRIMARY KEY (doctor_id),
23     FOREIGN KEY (dept_name) REFERENCES DEPT(dept_name)
24         on update cascade,
25     FOREIGN KEY (doctor_id) REFERENCES EMP(id)
26         on update cascade
27         on delete cascade,
28     FOREIGN KEY (super_d_id) REFERENCES DOCTOR(doctor_id)
29         on update cascade
30         on delete set null
31 );
32
```

## (2) NURSE, MANAGE\_N\_P, PATIENT Table 생성

```
33 • CREATE TABLE NURSE(  
34     nurse_id INT NOT NULL,  
35     dept_name CHAR(20),  
36     super_n_id INT,  
37     PRIMARY KEY (nurse_id),  
38     FOREIGN KEY (nurse_id) REFERENCES EMP(id)  
39         on update cascade  
40         on delete cascade,  
41     FOREIGN KEY (dept_name) REFERENCES DEPT(dept_name)  
42         on update cascade,  
43     FOREIGN KEY (super_n_id) REFERENCES NURSE(nurse_id)  
44         on update cascade  
45         on delete set null  
46 );  
47  
48 • CREATE TABLE MANAGE_N_P(  
49     nurse_id INT NOT NULL,  
50     PRIMARY KEY (nurse_id),  
51     FOREIGN KEY (nurse_id) REFERENCES NURSE(nurse_id)  
52         on update cascade  
53 );  
54  
55 • CREATE TABLE PATIENT(  
56     RRN CHAR(20) NOT NULL,  
57     patient_name CHAR(20),  
58     age INT,  
59     sex CHAR(5),  
60     insurance_code CHAR(20),  
61     doctor_id INT,  
62     PRIMARY KEY (RRN),  
63     FOREIGN KEY (doctor_id) REFERENCES DOCTOR(doctor_id)  
64         on update cascade  
65 );  
66
```

## (3) ROOM, SURGERY, SUREG\_RES Table 생성

```
68 • CREATE TABLE ROOM(  
69     room_num INT NOT NULL,  
70     room_telephone CHAR(20),  
71     nurse_id INT,  
72     PRIMARY KEY (room_num),  
73     FOREIGN KEY (nurse_id) REFERENCES NURSE(nurse_id)  
74         on update cascade  
75 );  
76  
77 • CREATE TABLE SURGERY(  
78     surgery_num INT NOT NULL,  
79     surgery_name CHAR(20),  
80     patient_RRN CHAR(20),  
81     PRIMARY KEY (surgery_num),  
82     FOREIGN KEY (patient_RRN) REFERENCES PATIENT(RRN)  
83         on update cascade  
84 );  
85  
86  
87 • CREATE TABLE SURGE_RES(  
88     surgery_num INT NOT NULL,  
89     date CHAR(20),  
90     time INT,  
91     p_condition CHAR(10),  
92     PRIMARY KEY (surgery_num),  
93     FOREIGN KEY (surgery_num) REFERENCES SURGERY(surgery_num)  
94         on update cascade  
95 );  
96
```

#### (4) ALTER Table

- 초기 table 생성에서 아직 존재하지 않았던 Foreign Key를 추가하고 각 Table에서 이를 Foreign Key로 설정 하였다.

```
98 • ALTER TABLE DEPT ADD COLUMN officer_id INT, ADD FOREIGN KEY (officer_id) REFERENCES DOCTOR(doctor_id)
99   on update cascade;
100
101 • ALTER TABLE DOCTOR ADD COLUMN surgery_num INT, ADD FOREIGN KEY (surgery_num) REFERENCES SURGERY(surgery_num)
102   on update cascade;
103
104 • ALTER TABLE MANAGE_N_P ADD COLUMN patient_RRN CHAR(20) NOT NULL, ADD FOREIGN KEY (patient_RRN) REFERENCES PATIENT(RRN)
105   on update cascade;
106
107 • ALTER TABLE MANAGE_N_P DROP PRIMARY KEY, ADD PRIMARY KEY (nurse_id, patient_RRN);
108
109 • ALTER TABLE PATIENT ADD room_num INT, ADD FOREIGN KEY (room_num) REFERENCES ROOM(room_num)
110   on update cascade;
111
```

#### (5) Insert tuples in EMP

```
130 • INSERT INTO EMP VALUES ('201323', '김이삼', 4);
131 • INSERT INTO EMP VALUES ('201324', '김이사', 10);
132 • INSERT INTO EMP VALUES ('201325', '김이오', 4);
133 • INSERT INTO EMP VALUES ('201326', '김이육', 2);
134 • INSERT INTO EMP VALUES ('201327', '김이칠', 12);
135 • INSERT INTO EMP VALUES ('201328', '김이팔', 1);
136 • INSERT INTO EMP VALUES ('201329', '김이구', 3);
137 • INSERT INTO EMP VALUES ('201330', '김삼영', 3);
138 • INSERT INTO EMP VALUES ('201331', '김삼일', 2);
139 • INSERT INTO EMP VALUES ('201332', '김삼이', 5);
140 • INSERT INTO EMP VALUES ('201333', '김삼삼', 8);
141 • INSERT INTO EMP VALUES ('201334', '김삼사', 16);
142 • INSERT INTO EMP VALUES ('201335', '김삼오', 21);
143 • INSERT INTO EMP VALUES ('201336', '김삼육', 22);
144 • INSERT INTO EMP VALUES ('201337', '김삼칠', 9);
145 • INSERT INTO EMP VALUES ('201338', '김삼팔', 13);
146 • INSERT INTO EMP VALUES ('201339', '김삼구', 12);
147
148
149 • INSERT INTO EMP VALUES ('201410', '이일영', 10);
150 • INSERT INTO EMP VALUES ('201411', '이일일', 7);
151 • INSERT INTO EMP VALUES ('201412', '이일이', 5);
152 • INSERT INTO EMP VALUES ('201413', '이일삼', 8);
153 • INSERT INTO EMP VALUES ('201414', '이일사', 2);
154 • INSERT INTO EMP VALUES ('201415', '이일오', 1);
155 • INSERT INTO EMP VALUES ('201416', '이일육', 13);
156 • INSERT INTO EMP VALUES ('201417', '이일칠', 16);
157 • INSERT INTO EMP VALUES ('201418', '이일팔', 12);
158 • INSERT INTO EMP VALUES ('201419', '이일구', 13);
159 • INSERT INTO EMP VALUES ('201420', '이이실', 8);
160 • INSERT INTO EMP VALUES ('201421', '이이일', 20);
161 • INSERT INTO EMP VALUES ('201422', '이이이', 19);
162 • INSERT INTO EMP VALUES ('201423', '이이삼', 4);
163 • INSERT INTO EMP VALUES ('201424', '이이사', 10);
164 • INSERT INTO EMP VALUES ('201425', '이이오', 4);
165 • INSERT INTO EMP VALUES ('201426', '이이육', 2);
166 • INSERT INTO EMP VALUES ('201427', '이이칠', 12);
167 • INSERT INTO EMP VALUES ('201428', '이이팔', 1);
168 • INSERT INTO EMP VALUES ('201429', '이이구', 3);
169
```

(중략)

(6) Insert tuples in DEPT

```
171 • INSERT INTO DEPT (dept_name, location_num, telephone)
172     VALUES ('내과', '1', '912-1111');
173 • INSERT INTO DEPT (dept_name, location_num, telephone)
174     VALUES ('흉부외과', '2', '912-2222');
175 • INSERT INTO DEPT (dept_name, location_num, telephone)
176     VALUES ('신경외과', '3', '912-3333');
177 • INSERT INTO DEPT (dept_name, location_num, telephone)
178     VALUES ('정형외과', '4', '912-4444');
179 • INSERT INTO DEPT (dept_name, location_num, telephone)
180     VALUES ('성형외과', '5', '912-5555');
181 • INSERT INTO DEPT (dept_name, location_num, telephone)
182     VALUES ('산부인과', '6', '912-6666');
183 • INSERT INTO DEPT (dept_name, location_num, telephone)
184     VALUES ('피부과', '7', '912-7777');
185 • INSERT INTO DEPT (dept_name, location_num, telephone)
186     VALUES ('비뇨기과', '8', '912-8888');
187 • INSERT INTO DEPT (dept_name, location_num, telephone)
188     VALUES ('안과', '9', '912-9999');
189 • INSERT INTO DEPT (dept_name, location_num, telephone)
190     VALUES ('신경과', '10', '912-1010');
191
```

(중략)

(7) Insert tuples in DOCTOR

```
193 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
194     VALUES ('201310', '내과', '201310');
195 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
196     VALUES ('201311', '내과', '201310');
197 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
198     VALUES ('201312', '내과', '201310');
199 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
200     VALUES ('201313', '흉부외과', '201313');
201 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
202     VALUES ('201314', '흉부외과', '201313');
203 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
204     VALUES ('201315', '흉부외과', '201313');
205 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
206     VALUES ('201316', '신경외과', '201316');
207 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
208     VALUES ('201317', '신경외과', '201316');
209 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
210     VALUES ('201318', '신경외과', '201316');
211 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
212     VALUES ('201319', '정형외과', '201319');
213 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
214     VALUES ('201320', '정형외과', '201319');
215 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
216     VALUES ('201321', '정형외과', '201319');
217 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
218     VALUES ('201322', '성형외과', '201322');
219 • INSERT INTO doctor (doctor_id, dept_name, super_d_id)
220     VALUES ('201323', '성형외과', '201322');
```

(중략)

(8) Insert tuples in NURSE

```
277 • INSERT INTO nurse
278     VALUES ('201410', '내과', '201410');
279 • INSERT INTO nurse
280     VALUES ('201411', '내과', '201410');
281 • INSERT INTO nurse
282     VALUES ('201412', '흉부외과', '201412');
283 • INSERT INTO nurse
284     VALUES ('201413', '흉부외과', '201412');
285 • INSERT INTO nurse
286     VALUES ('201414', '신경외과', '201414');
287 • INSERT INTO nurse
288     VALUES ('201415', '신경외과', '201414');
289 • INSERT INTO nurse
290     VALUES ('201416', '정형외과', '201416');
291 • INSERT INTO nurse
292     VALUES ('201417', '정형외과', '201416');
293 • INSERT INTO nurse
294     VALUES ('201418', '성형외과', '201418');
```

(중략)

(9) Insert tuples in ROOM

```
319 • INSERT ROOM VALUES ('101', '912-9101', '201411');
320 • INSERT ROOM VALUES ('102', '912-9102', '201412');
321 • INSERT ROOM VALUES ('103', '912-9103', '201413');
322 • INSERT ROOM VALUES ('104', '912-9104', '201414');
323 • INSERT ROOM VALUES ('105', '912-9105', '201415');
324 • INSERT ROOM VALUES ('106', '912-9106', '201416');
325 • INSERT ROOM VALUES ('107', '912-9107', '201417');
326 • INSERT ROOM VALUES ('108', '912-9108', '201418');
327 • INSERT ROOM VALUES ('109', '912-9109', '201419');
328 • INSERT ROOM VALUES ('110', '912-9110', '201420');
329 • INSERT ROOM VALUES ('111', '912-9111', '201421');
330 • INSERT ROOM VALUES ('112', '912-9112', '201422');
331 • INSERT ROOM VALUES ('113', '912-9113', '201423');
332 • INSERT ROOM VALUES ('114', '912-9114', '201424');
333 • INSERT ROOM VALUES ('115', '912-9115', '201425');
```

(중략)



(10) Insert tuples in PATIENT

```
336 • INSERT PATIENT VALUES ('950712-1056821', '박환일', '22', 'M', '999011', '201310', '101');
337 • INSERT PATIENT VALUES ('950612-1056822', '박환이', '22', 'M', '999012', '201310', '106');
338 • INSERT PATIENT VALUES ('940512-1056823', '박환삼', '23', 'M', '999013', '201312', '106');
339 • INSERT PATIENT VALUES ('940412-1056824', '박환사', '23', 'M', '999014', '201313', '102');
340 • INSERT PATIENT VALUES ('930312-1056825', '박환오', '24', 'M', '999015', '201322', '102');
341 • INSERT PATIENT VALUES ('930212-1056826', '박환옥', '24', 'M', '999016', '201314', '103');
342 • INSERT PATIENT VALUES ('920112-1056827', '박환칠', '25', 'M', '999017', '201321', '103');
343 • INSERT PATIENT VALUES ('920713-1056828', '박환팔', '25', 'M', '999018', '201320', '104');
344 • INSERT PATIENT VALUES ('910714-1056829', '박환구', '26', 'M', '999019', '201331', '104');
345 • INSERT PATIENT VALUES ('910715-1056830', '박환십', '26', 'M', '999020', '201327', '109');
346 • INSERT PATIENT VALUES ('900716-1056831', '박환십일', '27', 'M', '999021', '201329', '109');
347 • INSERT PATIENT VALUES ('900714-1056832', '박환십이', '27', 'M', '999022', '201317', '112');
348 • INSERT PATIENT VALUES ('890812-1056833', '박환십삼', '28', 'M', '999023', '201315', '112');
349 • INSERT PATIENT VALUES ('891230-1056834', '박환십사', '28', 'M', '999024', '201336', '113');
350 • INSERT PATIENT VALUES ('870212-1056835', '박환십오', '30', 'M', '999025', '201338', '113');
351 • INSERT PATIENT VALUES ('830612-2056821', '여환일', '34', 'F', '999111', '201330', '101');
352 • INSERT PATIENT VALUES ('830612-2056822', '여환이', '34', 'F', '999112', '201331', '105');
353 • INSERT PATIENT VALUES ('840612-2056823', '여환삼', '33', 'F', '999113', '201332', '105');
354 • INSERT PATIENT VALUES ('840612-2056824', '여환사', '33', 'F', '999114', '201333', '106');
```

(중략)

(11) Insert tuples in MANAGE\_N\_P

```
368 • INSERT manage_n_p VALUES ('201410', '950712-1056821');
369 • INSERT manage_n_p VALUES ('201411', '950612-1056822');
370 • INSERT manage_n_p VALUES ('201412', '940512-1056823');
371 • INSERT manage_n_p VALUES ('201413', '940412-1056824');
372 • INSERT manage_n_p VALUES ('201414', '930312-1056825');
373 • INSERT manage_n_p VALUES ('201415', '930212-1056826');
374 • INSERT manage_n_p VALUES ('201416', '920112-1056827');
375 • INSERT manage_n_p VALUES ('201417', '920713-1056828');
376 • INSERT manage_n_p VALUES ('201418', '910714-1056829');
377 • INSERT manage_n_p VALUES ('201419', '910715-1056830');
378 • INSERT manage_n_p VALUES ('201420', '900716-1056831');
379 • INSERT manage_n_p VALUES ('201421', '900714-1056832');
380 • INSERT manage_n_p VALUES ('201422', '890812-1056833');
381 • INSERT manage_n_p VALUES ('201423', '891230-1056834');
382 • INSERT manage_n_p VALUES ('201424', '870212-1056835');
383 • INSERT manage_n_p VALUES ('201425', '830612-2056821');
384 • INSERT manage_n_p VALUES ('201426', '830612-2056822');
385 • INSERT manage_n_p VALUES ('201427', '840612-2056823');
386 • INSERT manage_n_p VALUES ('201428', '840612-2056824');
387 • INSERT manage_n_p VALUES ('201429', '850612-2056825');
388 • INSERT manage_n_p VALUES ('201410', '850612-2056826');
389 • INSERT manage_n_p VALUES ('201411', '860612-2056827');
390 • INSERT manage_n_p VALUES ('201412', '860612-2056828');
391 • INSERT manage_n_p VALUES ('201413', '930612-2056829');
392 • INSERT manage_n_p VALUES ('201414', '930612-2056830');
393 • INSERT manage_n_p VALUES ('201415', '920612-2056831');
394 • INSERT manage_n_p VALUES ('201420', '920612-2056832');
395 • INSERT manage_n_p VALUES ('201421', '910612-2056833');
396 • INSERT manage_n_p VALUES ('201422', '910612-2056834');
397 • INSERT manage_n_p VALUES ('201423', '900612-2056835');
```

(중략)

(12) Insert tuples in SURGERY

```
400 • INSERT SURGERY VALUES ('9901', '제1수술', '950712-1056821');
401 • INSERT SURGERY VALUES ('9902', '제2수술', '950612-1056822');
402 • INSERT SURGERY VALUES ('9903', '제3수술', '940512-1056823');
403 • INSERT SURGERY VALUES ('9904', '제4수술', '940412-1056824');
404 • INSERT SURGERY VALUES ('9905', '제5수술', '930312-1056825');
405 • INSERT SURGERY VALUES ('9906', '제6수술', '930212-1056826');
406 • INSERT SURGERY VALUES ('9907', '제7수술', '920112-1056827');
407 • INSERT SURGERY VALUES ('9908', '제8수술', '920713-1056828');
408 • INSERT SURGERY VALUES ('9909', '제9수술', '910714-1056829');
409 • INSERT SURGERY VALUES ('9910', '제10수술', '910715-1056830');
410 • INSERT SURGERY VALUES ('9911', '제11수술', '900716-1056831');
411 • INSERT SURGERY VALUES ('9912', '제12수술', '900714-1056832');
412 • INSERT SURGERY VALUES ('9913', '제13수술', '890812-1056833');
413 • INSERT SURGERY VALUES ('9914', '제14수술', '891230-1056834');
414 • INSERT SURGERY VALUES ('9915', '제15수술', '870212-1056835');
415 • INSERT SURGERY VALUES ('9916', '제16수술', '830612-2056821');
416 • INSERT SURGERY VALUES ('9917', '제17수술', '830612-2056822');
417 • INSERT SURGERY VALUES ('9918', '제18수술', '840612-2056823');
418 • INSERT SURGERY VALUES ('9919', '제19수술', '840612-2056824');
419 • INSERT SURGERY VALUES ('9920', '제20수술', '850612-2056825');
420 • INSERT SURGERY VALUES ('9921', '제21수술', '850612-2056826');
421 • INSERT SURGERY VALUES ('9922', '제22수술', '860612-2056827');
422 • INSERT SURGERY VALUES ('9923', '제23수술', '860612-2056828');
423 • INSERT SURGERY VALUES ('9924', '제24수술', '930612-2056829');
424 • INSERT SURGERY VALUES ('9925', '제25수술', '930612-2056830');
```

(중략)

(13) Insert tuples in SURGE\_RES

```
454 • INSERT SURGE_RES VALUES('9901', '16.06.20', '3', '4');
455 • INSERT SURGE_RES VALUES('9902', '16.06.19', '7', '6');
456 • INSERT SURGE_RES VALUES('9903', '16.06.18', '5', '7');
457 • INSERT SURGE_RES VALUES('9904', '16.06.17', '2', '8');
458 • INSERT SURGE_RES VALUES('9905', '16.05.20', '1', '8');
459 • INSERT SURGE_RES VALUES('9906', '16.04.20', '2', '7');
460 • INSERT SURGE_RES VALUES('9907', '16.03.20', '7', '8');
461 • INSERT SURGE_RES VALUES('9908', '16.02.20', '8', '9');
462 • INSERT SURGE_RES VALUES('9909', '16.01.20', '13', '9');
463 • INSERT SURGE_RES VALUES('9910', '15.09.21', '12', '8');
464 • INSERT SURGE_RES VALUES('9911', '15.09.12', '10', '7');
465 • INSERT SURGE_RES VALUES('9912', '15.06.13', '10', '5');
466 • INSERT SURGE_RES VALUES('9913', '15.07.07', '12', '5');
467 • INSERT SURGE_RES VALUES('9914', '15.08.23', '7', '1');
468 • INSERT SURGE_RES VALUES('9915', '15.09.21', '8', '2');
469 • INSERT SURGE_RES VALUES('9916', '15.10.18', '9', '1');
470 • INSERT SURGE_RES VALUES('9917', '15.11.05', '9', '6');
471 • INSERT SURGE_RES VALUES('9918', '14.12.14', '4', '4');
472 • INSERT SURGE_RES VALUES('9919', '14.04.23', '5', '3');
473 • INSERT SURGE_RES VALUES('9920', '14.05.12', '7', '3');
474 • INSERT SURGE_RES VALUES('9921', '14.06.17', '6', '4');
475 • INSERT SURGE_RES VALUES('9922', '14.07.09', '12', '9');
476 • INSERT SURGE_RES VALUES('9923', '14.08.14', '15', '4');
477 • INSERT SURGE_RES VALUES('9924', '13.06.12', '13', '8');
478 • INSERT SURGE_RES VALUES('9925', '08.06.02', '5', '6');
```

(중략)

(14) Insert more attributes in some Table

- 초기에 존재하지 않은 Foreign key에 의해 입력하지 못했던 몇몇 attributes를 UPDATE를 이용하여 후에 추가하였다.

```
255 • UPDATE DEPT SET officer_id = '201310'
256     WHERE dept_name = '내과';
257 • UPDATE DEPT SET officer_id = '201313'
258     WHERE dept_name = '흉부외과';
259 • UPDATE DEPT SET officer_id = '201316'
260     WHERE dept_name = '신경외과';
261 • UPDATE DEPT SET officer_id = '201319'
262     WHERE dept_name = '정형외과';
263 • UPDATE DEPT SET officer_id = '201322'
264     WHERE dept_name = '정형외과';
265 • UPDATE DEPT SET officer_id = '201325'
266     WHERE dept_name = '산부인과';
267 • UPDATE DEPT SET officer_id = '201328'
268     WHERE dept_name = '과부과';
269 • UPDATE DEPT SET officer_id = '201331'
270     WHERE dept_name = '비뇨기과';
271 • UPDATE DEPT SET officer_id = '201334'
272     WHERE dept_name = '안과';
273 • UPDATE DEPT SET officer_id = '201337'
274     WHERE dept_name = '신경과';
275

427 • UPDATE DOCTOR SET surgery_num = '9901' WHERE doctor_id = '201339';
428 • UPDATE DOCTOR SET surgery_num = '9902' WHERE doctor_id = '201338';
429 • UPDATE DOCTOR SET surgery_num = '9903' WHERE doctor_id = '201337';
430 • UPDATE DOCTOR SET surgery_num = '9904' WHERE doctor_id = '201336';
431 • UPDATE DOCTOR SET surgery_num = '9905' WHERE doctor_id = '201335';
432 • UPDATE DOCTOR SET surgery_num = '9906' WHERE doctor_id = '201334';
433 • UPDATE DOCTOR SET surgery_num = '9907' WHERE doctor_id = '201333';
434 • UPDATE DOCTOR SET surgery_num = '9908' WHERE doctor_id = '201332';
435 • UPDATE DOCTOR SET surgery_num = '9909' WHERE doctor_id = '201331';
436 • UPDATE DOCTOR SET surgery_num = '9910' WHERE doctor_id = '201330';
437 • UPDATE DOCTOR SET surgery_num = '9911' WHERE doctor_id = '201329';
438 • UPDATE DOCTOR SET surgery_num = '9912' WHERE doctor_id = '201328';
439 • UPDATE DOCTOR SET surgery_num = '9913' WHERE doctor_id = '201327';
440 • UPDATE DOCTOR SET surgery_num = '9914' WHERE doctor_id = '201326';
441 • UPDATE DOCTOR SET surgery_num = '9915' WHERE doctor_id = '201325';
442 • UPDATE DOCTOR SET surgery_num = '9916' WHERE doctor_id = '201324';
443 • UPDATE DOCTOR SET surgery_num = '9917' WHERE doctor_id = '201323';
444 • UPDATE DOCTOR SET surgery_num = '9918' WHERE doctor_id = '201322';
445 • UPDATE DOCTOR SET surgery_num = '9919' WHERE doctor_id = '201321';
446 • UPDATE DOCTOR SET surgery_num = '9920' WHERE doctor_id = '201320';
447 • UPDATE DOCTOR SET surgery_num = '9921' WHERE doctor_id = '201310';
448 • UPDATE DOCTOR SET surgery_num = '9922' WHERE doctor_id = '201311';
449 • UPDATE DOCTOR SET surgery_num = '9923' WHERE doctor_id = '201312';
450 • UPDATE DOCTOR SET surgery_num = '9924' WHERE doctor_id = '201313';
451 • UPDATE DOCTOR SET surgery_num = '9925' WHERE doctor_id = '201314';
```

(종락)



## 10. Table 결과

### (1) EMP

	id	name	work_year
▶	201310	김일영	10
	201311	김일일	7
	201312	김일이	5
	201313	김일삼	8
	201314	김일사	2
	201315	김일오	1
	201316	김일육	13
	201317	김일칠	16
	201318	김일팔	12
	201319	김일구	13
	201320	김이십	8
	201321	김이일	20
	201322	김이이	19
	201323	김이삼	4
	201324	김이사	10
	201325	김이오	4

(중략)

### (2) DEPT

	dept_name	location_num	telephone	officer_id
▶	내과	1	912-1111	201310
	비뇨기과	8	912-8888	201331
	산부인과	6	912-6666	201325
	성형외과	5	912-5555	201322
	신경과	10	912-1010	201337
	신경외과	3	912-3333	201316
	안과	9	912-9999	201334
	정형외과	4	912-4444	201319
	피부과	7	912-7777	201328
	흉부외과	2	912-2222	201313
*	NULL	NULL	NULL	NULL

(중략)

### (3) DOCTOR

	doctor_id	dept_name	super_d_id	surgery_num
▶	201310	내과	201310	9921
	201311	내과	201310	9922
	201312	내과	201310	9923
	201313	흉부외과	201313	9924
	201314	흉부외과	201313	9925
	201315	흉부외과	201313	NULL
	201316	신경외과	201316	NULL
	201317	신경외과	201316	NULL
	201318	신경외과	201316	NULL
	201319	정형외과	201319	NULL
	201320	정형외과	201319	9920
	201321	정형외과	201319	9919
	201322	성형외과	201322	9918
	201323	성형외과	201322	9917
	201324	성형외과	201322	9916
	201325	산부인과	201325	9915

(중략)

### (4) NURSE

	nurse_id	dept_name	super_n_id
▶	201410	내과	201410
	201411	내과	201410
	201412	흉부외과	201412
	201413	흉부외과	201412
	201414	신경외과	201414
	201415	신경외과	201414
	201416	정형외과	201416
	201417	정형외과	201416
	201418	성형외과	201418
	201419	성형외과	201418
	201420	산부인과	201420
	201421	산부인과	201420
	201422	피부과	201422
	201423	피부과	201422
	201424	비뇨기과	201424
	201425	비뇨기과	201424
	201426	안과	201426

(중략)

(5) MANAGE\_N\_P

	nurse_id	patient_RRN
▶	201425	830612-2056821
	201426	830612-2056822
	201427	840612-2056823
	201428	840612-2056824
	201429	850612-2056825
	201410	850612-2056826
	201411	860612-2056827
	201412	860612-2056828
	201424	870212-1056835
	201422	890812-1056833
	201423	891230-1056834
	201423	900612-2056835
	201421	900714-1056832
	201420	900716-1056831
	201421	910612-2056833
	201422	910612-2056834
	201418	910714-1056829

(중략)

(6) PATIENT

	RRN	patient_name	age	sex	insurance_code	doctor_id	room_num
▶	830612-2056821	여환일	34	F	999111	201330	101
	830612-2056822	여환미	34	F	999112	201331	105
	840612-2056823	여환삼	33	F	999113	201332	105
	840612-2056824	여환사	33	F	999114	201333	106
	850612-2056825	여환오	32	F	999115	201334	107
	850612-2056826	여환육	32	F	999116	201335	107
	860612-2056827	여환칠	31	F	999117	201336	108
	860612-2056828	여환팔	31	F	999118	201337	108
	870212-1056835	박환십오	30	M	999025	201338	113
	890812-1056833	박환십삼	28	M	999023	201315	112
	891230-1056834	박환십사	28	M	999024	201336	113
	900612-2056835	여환십오	27	F	999125	201324	115
	900714-1056832	박환십미	27	M	999022	201317	112
	900716-1056831	박환십일	27	M	999021	201329	109
	910612-2056833	여환십삼	26	F	999123	201322	111
	910612-2056834	여환십사	26	F	999124	201323	114
	910714-1056829	박환구	26	M	999019	201331	104
	910715-1056830	박환십	26	M	999020	201327	109
	920112-1056827	박환칠	25	M	999017	201321	103
	920612-2056831	여환십일	25	F	999121	201320	110

(중략)

(7) ROOM

	room_num	room_telephone	nurse_id
▶	101	912-9101	201411
	102	912-9102	201412
	103	912-9103	201413
	104	912-9104	201414
	105	912-9105	201415
	106	912-9106	201416
	107	912-9107	201417
	108	912-9108	201418
	109	912-9109	201419
	110	912-9110	201420
	111	912-9111	201421
	112	912-9112	201422
	113	912-9113	201423
	114	912-9114	201424
	115	912-9115	201425

(중략)

(8) SURGERY

	surgery_num	surgery_name	patient_RRN
▶	9901	제1수술	950712-1056821
	9902	제2수술	950612-1056822
	9903	제3수술	940512-1056823
	9904	제4수술	940412-1056824
	9905	제5수술	930312-1056825
	9906	제6수술	930212-1056826
	9907	제7수술	920112-1056827
	9908	제8수술	920713-1056828
	9909	제9수술	910714-1056829
	9910	제10수술	910715-1056830
	9911	제11수술	900716-1056831
	9912	제12수술	900714-1056832
	9913	제13수술	890812-1056833
	9914	제14수술	891230-1056834
	9915	제15수술	870212-1056835
	9916	제16수술	830612-2056821

(중략)

(9) SURGE\_RES

	surgery_num	date	time	p_condition
▶	9901	16.06.20	3	4
	9902	16.06.19	7	6
	9903	16.06.18	5	7
	9904	16.06.17	2	8
	9905	16.05.20	1	8
	9906	16.04.20	2	7
	9907	16.03.20	7	8
	9908	16.02.20	8	9
	9909	16.01.20	13	9
	9910	15.09.21	12	8
	9911	15.09.12	10	7
	9912	15.06.13	10	5
	9913	15.07.07	12	5
	9914	15.08.23	7	1
	9915	15.09.21	8	2
	9916	15.10.18	9	1
	9917	15.11.05	9	6

(중략)

## 11. CREATE VIEW

• CREATE VIEW와 생성된 Table들을 이용하여 총 5가지의 각각 다른 view를 생성하였다. 아래는 5가지의 view를 생성하는 code이다.

```
482 • CREATE VIEW PATIENT_IN_DEPT (dept_name, num_patient) AS
483     SELECT DEPT.dept_name, COUNT(*)
484     FROM PATIENT, DOCTOR, DEPT
485     WHERE (PATIENT.doctor_id = DOCTOR.doctor_id)
486           AND (DOCTOR.dept_name = DEPT.dept_name)
487     GROUP BY DEPT.dept_name;
488
489 • CREATE VIEW DEPT_ROOM (dept_name, num_room) AS
490     SELECT dept_name, COUNT(*)
491     FROM ROOM, NURSE
492     WHERE ROOM.nurse_id = NURSE.nurse_id
493     GROUP BY dept_name;
494
495 • CREATE VIEW PATIENT_CONDITION AS
496     SELECT patient_name, p_condition
497     FROM SURGERY, surge_res, PATIENT
498     WHERE (SURGERY.surgery_num = SURGE_res.surgery_num)
499           AND (SURGERY.patient_RRN = PATIENT.RRN)
500     ORDER BY patient_name;
501
502 • CREATE VIEW SURGERY_DEPT (surgery_name, dept_name, doctor_name) AS
503     SELECT surgery_name, DEPT.dept_name, EMP.name
504     FROM SURGERY, DOCTOR, DEPT, EMP
505     WHERE (SURGERY.surgery_num = DOCTOR.surgery_num)
506           AND (DOCTOR.dept_name = DEPT.dept_name)
507           AND (DOCTOR.doctor_id = EMP.id)
508     ORDER BY surgery_name;
509
510 • CREATE VIEW DOCTOR_SURGERY_CONDITION (doctor_name, avg_condition) AS
511     SELECT e.name, AVG(sr.p_condition)
512     FROM EMP AS e, DOCTOR AS d, SURGERY AS s, SURGE_RES AS sr
513     WHERE (e.id = d.doctor_id)
514           AND (s.surgery_num = d.surgery_num)
515           AND (s.surgery_num = sr.surgery_num)
516     GROUP BY e.name
517     ORDER BY sr.p_condition;
```

#### (1) PATIENT\_IN\_DEPT

- 각 부서에서 진료를 받는 환자의 수를 표시한 view이다.

	dept_name	num_patient
▶	내과	3
	비뇨기과	4
	산부인과	1
	성형외과	4
	신경과	4
	신경외과	1
	안과	4
	정형외과	4
	피부과	2
	흉부외과	3

#### (2) DEPT\_ROOM

- 각 부서에 속한 간호사가 관리하는 병실의 수를 표시하는 view이다.

	dept_name	num_room
▶	내과	1
	비뇨기과	2
	산부인과	2
	성형외과	2
	신경외과	2
	정형외과	2
	피부과	2
	흉부외과	2

### (3) PATIENT\_CONDITION

- 수술을 받은 환자들의 수술 후의 condition을 표시한 view이다.

	patient_name	p_condition
▶	박환구	9
	박환사	8
	박환삼	7
	박환십	8
	박환십사	1
	박환십삼	5
	박환십오	2
	박환십이	5
	박환십일	7
	박환오	8
	박환육	7
	박환미	6
	박환일	4
	박환칠	8
	박환팔	9
	여환구	8
	여환사	3
	여환삼	4

(중략)



#### (4) SURGERY\_DEPT

- 각 수술명에 대해 수술의 담당 부서 및 수술을 집도한 의사의 이름을 표시한 view이다.

	surgery_name	dept_name	doctor_name
▶	제 10수술	피부과	김삼영
	제 11수술	피부과	김이구
	제 12수술	피부과	김이팔
	제 13수술	산부인과	김이칠
	제 14수술	산부인과	김이육
	제 15수술	산부인과	김미오
	제 16수술	성형외과	김이사
	제 17수술	성형외과	김이삼
	제 18수술	성형외과	김이이
	제 19수술	정형외과	김이일
	제 1수술	신경과	김삼구
	제 20수술	정형외과	김이십
	제 21수술	내과	김일영
	제 22수술	내과	김일일
	제 23수술	내과	김일이
	제 24수술	흉부외과	김일삼
	제 25수술	흉부외과	김일사
	제 2수술	신경과	김삼팔
	제 3수술	신경과	김삼칠
	제 4수술	안과	김삼육

(중략)

(5) DOCTOR\_SURGERY\_CONDITION

- 수술을 집도한 의사들에 대해, 수술 후 환자의 condition을 표시한 view이다.

	doctor_name	avg_condition
▶	김이사	1
	김이육	1
	김이오	2
	김이십	3
	김이일	3
	김이미	4
	김일이	4
	김일영	4
	김삼구	4
	김이팔	5
	김이철	5
	김삼팔	6
	김일사	6
	김이삼	6
	김이구	7
	김삼철	7
	김삼사	7
	김삼육	8
	김삼삼	8
	김삼오	8
	김삼영	8
	김일삼	8
	김삼일	9
	김삼미	9

(중략)

## 12. QUERY문 생성 및 확인

- 총 5가지의 QUERY문을 SELECT FROM WHERE 및 GROUP BY, ORDER BY, VIEW를 이용하여 명령문으로 작성하고, 결과를 확인하였다.

(1) 수술을 받은 환자들 중, 수술 후의 condition이 5 이상인 환자들의 이름을 출력하라.

```
1 • USE hospital;
2 |
3 • SELECT p_condition, patient_name
4 FROM surge_res, PATIENT, surgery
5 WHERE (surge_res.surgery_num = surgery.surgery_num)
6       AND (surgery.patient_RRN = patient.RRN)
7       AND (surge_res.p_condition > 5);
```

Result Grid

	p_condition	patient_name
▶	6	박환미
	7	박환삼
	8	박환사
	8	박환오
	7	박환육
	8	박환칠
	9	박환팔
	9	박환구
	8	박환십
	7	박환십일
	6	여환미
	9	여환칠
	8	여환구
	6	여환십

(2) 병원 경력이 10년 이상인 의사들의 이름을 출력하라.

```
1 • use hospital;
2
3 • SELECT emp.work_year, emp.name
4 from emp
5 where (emp.work_year >= 10) AND (emp.id < 201400)
6 order by emp.work_year;
```

work_year	name
10	김일영
10	김이사
12	김일팔
12	김이철
12	김삼구
13	김일육
13	김일구
13	김삼팔
16	김일철
16	김삼사
19	김이미
20	김이일
21	김삼오
22	김삼육

(3) 환자의 나이가 27세 이상인 환자의 수를 출력하라.

```
1 • use hospital;
2
3 • select count(*)
4   from patient
5  where patient.age >= 27;
```

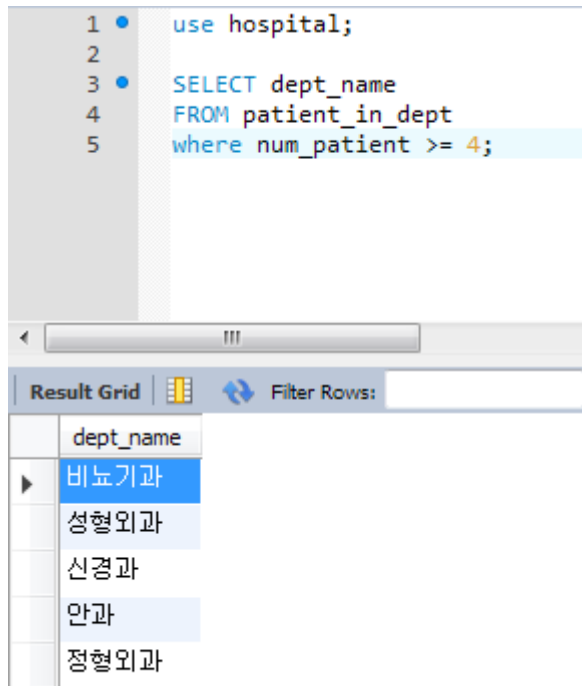
Result Grid	Filter Rows:
count(*)	
14	

(4) 각 부서에 따라 책임자의 이름 및 병원 경력을 출력하라.

```
1 • use hospital;
2
3 • select dept.dept_name, emp.name, emp.work_year
4   from dept, doctor, emp
5  where dept.officer_id = doctor.doctor_id
6        AND doctor.doctor_id = emp.id
7  order by dept.dept_name;
```

Result Grid	Filter Rows:	Export:	Wrap Cell C
dept_name	name	work_year	
내과	김일영	10	
비뇨기과	김삼일	2	
산부인과	김미오	4	
성형외과	김이미	19	
신경과	김삼철	9	
신경외과	김일욱	13	
안과	김삼사	16	
정형외과	김일구	13	
피부과	김미팔	1	
흉부외과	김일삼	8	

(5) 4명 이상의 환자들이 진료를 받은 의사의 해당하는 부서의 이름을 출력하라.



### 13. 고찰

- 본 프로젝트를 통해 병원 내부의 데이터베이스를 직접 설계하고, MySQL을 이용하여 컴퓨터 상에 구현 및 여러 QUERY문을 실행하였다. 실제 병원에서 데이터 저장에 사용되는 데이터베이스에는 훨씬 많은 table과 attribute가 존재할 것이다. 이 중에서 가장 중요하다고 생각되는 부분만을 추출하여 직접 구현하였다. 수업에서 이론적으로 배운 ER Diagram, Relational Schema, SQL을 직접 실제로 구현하고, 적용 및 구축하는 과정이 매우 흥미로웠고, 한 학기 동안 배운 지식을 사용할 수 있다는 점에서 매우 뜻 깊은 프로젝트였다.

### 14. 참고 문헌

- DB프로젝트 (2016-병원정보).ppt, 김응모 교수님, 2016년.
- 2016년 1학기 데이터베이스 수업자료, 김응모 교수님, 2016년.
- Online+Banking+DB.pdf, 2016.