

# COVID-19 위기 대응 시스템

발표일

2020년 5월 18일

팀원

강제영, 김연희, 이민호, 이제현

# 프로젝트 주제 및 목적

## 주제

COVID-19 위기 대응 시스템

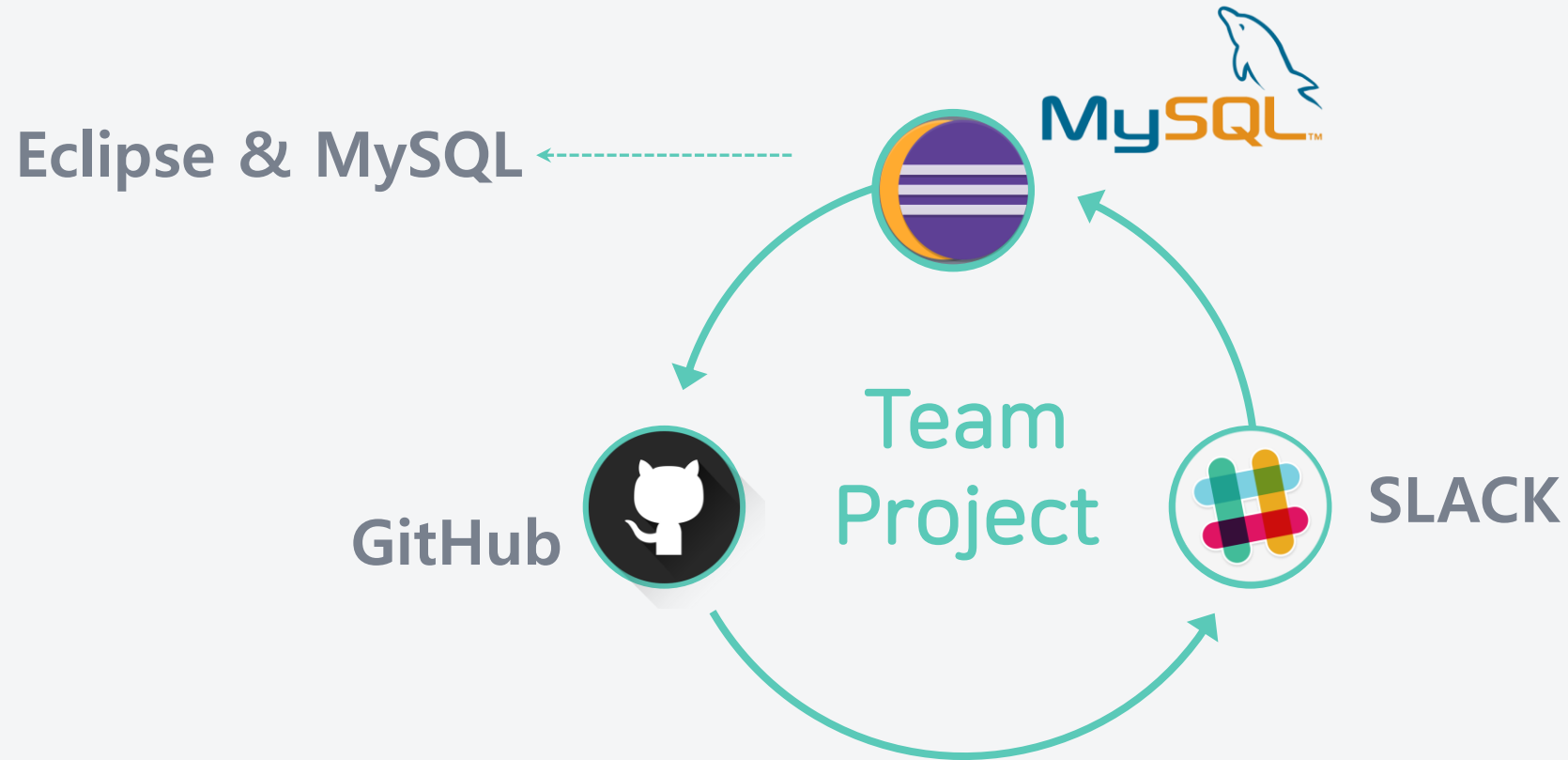
## 선정 이유

- 현실 세계 시의성 반영
- 기존 프로그램을 참고하여 직접 구현
- 메소드 구현의 높은 확장성

## 목적

- 대국민 안심 조회 서비스
- 확진자 관리 및 검사
- 방역 시행 여부 관리
- 코로나 관련 통계 알림

# 작업 과정



# Hierarchical Structure

```
Corona19 [2ndProject master ↑ 1]
├── JRE System Library [JavaSE-1.8]
└── src
    ├── config
    │   ├── QueryInfo.java
    │   └── ServerInfo.java
    ├── corona.dao
    │   ├── Database.java
    │   └── DatabaseTemplate.java
    ├── corona.exception
    │   ├── DuplicateSSNException.java
    │   └── RecordNotFoundException.java
    ├── corona.test
    │   └── Corona_test.java
    └── corona.vo
        ├── Hospital.java
        ├── Infectee.java
        ├── Person.java
        ├── Place.java
        └── Visit.java
```

config

QueryInfo / ServerInfo

corona.dao

Database / DatabaseTemplate

corona.exception

DuplicateSSNException

RecordNotFoundException

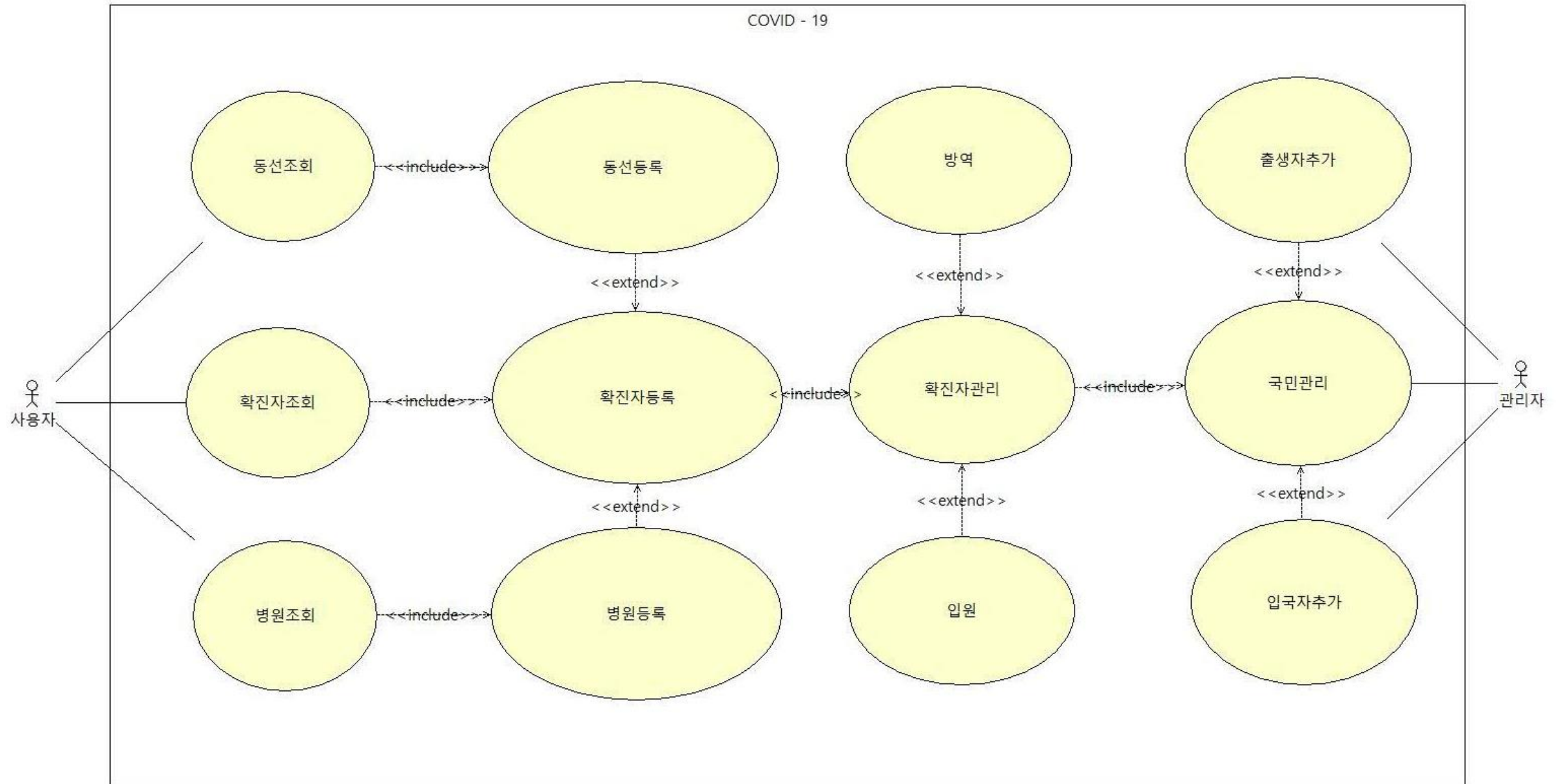
corona.test

Corona\_test

corona.vo

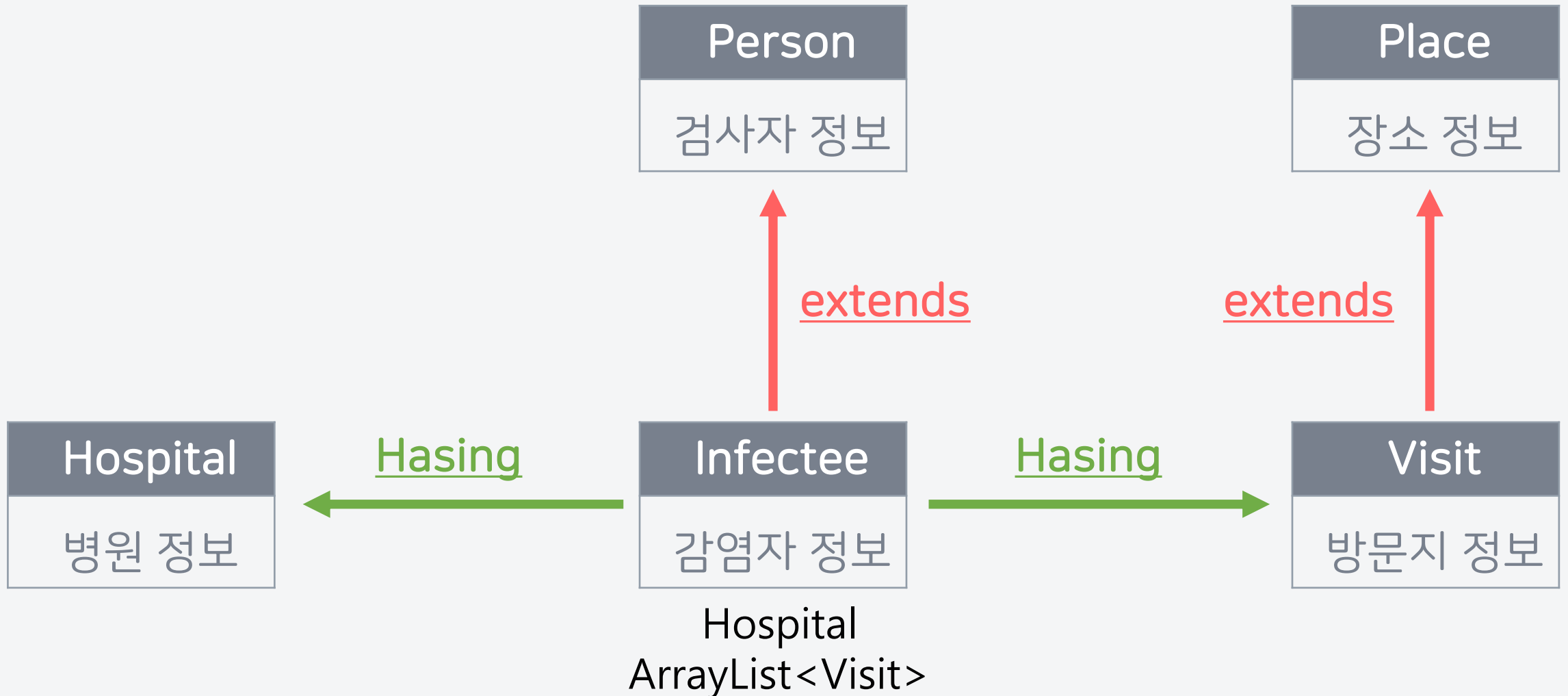
Hospital / Infectee / Person /  
Place / Visit

# UseCase Diagram



# Class Diagram

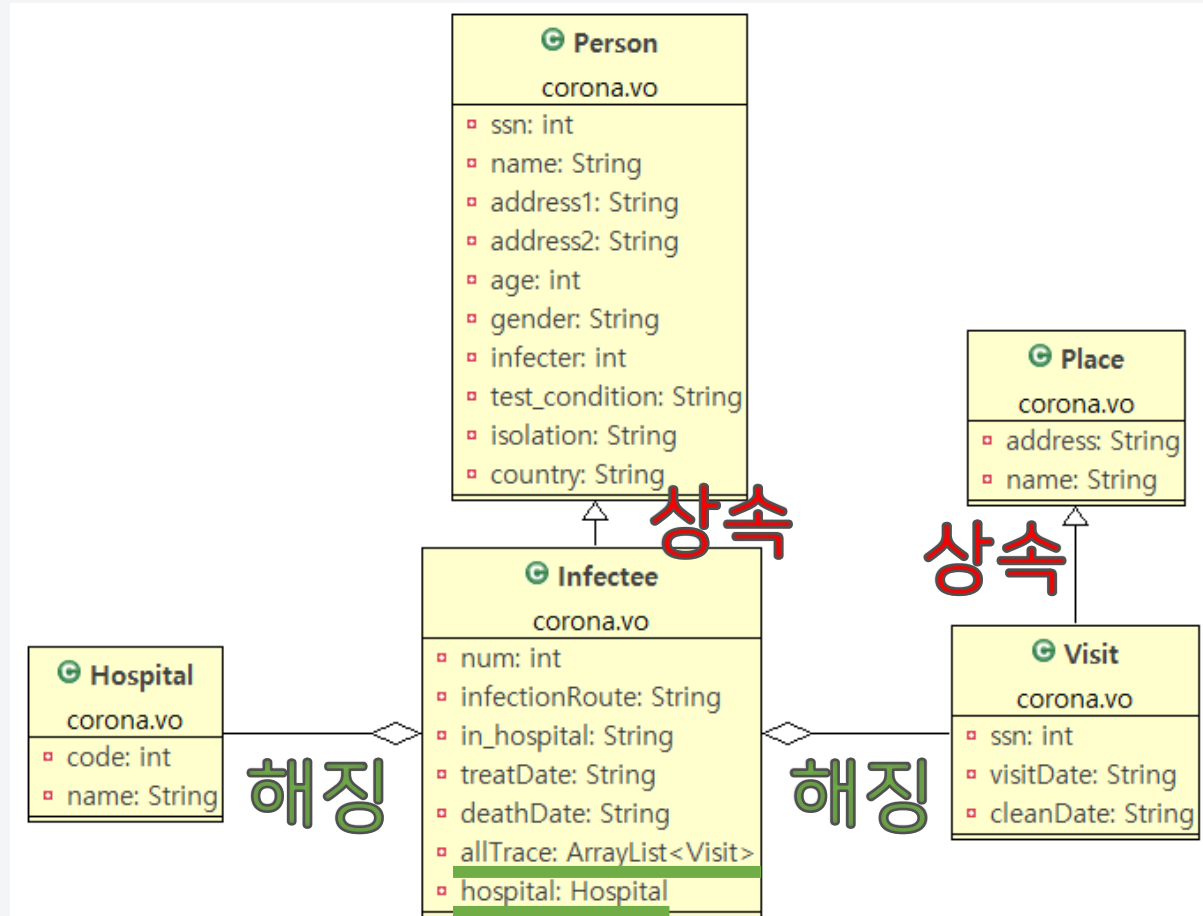
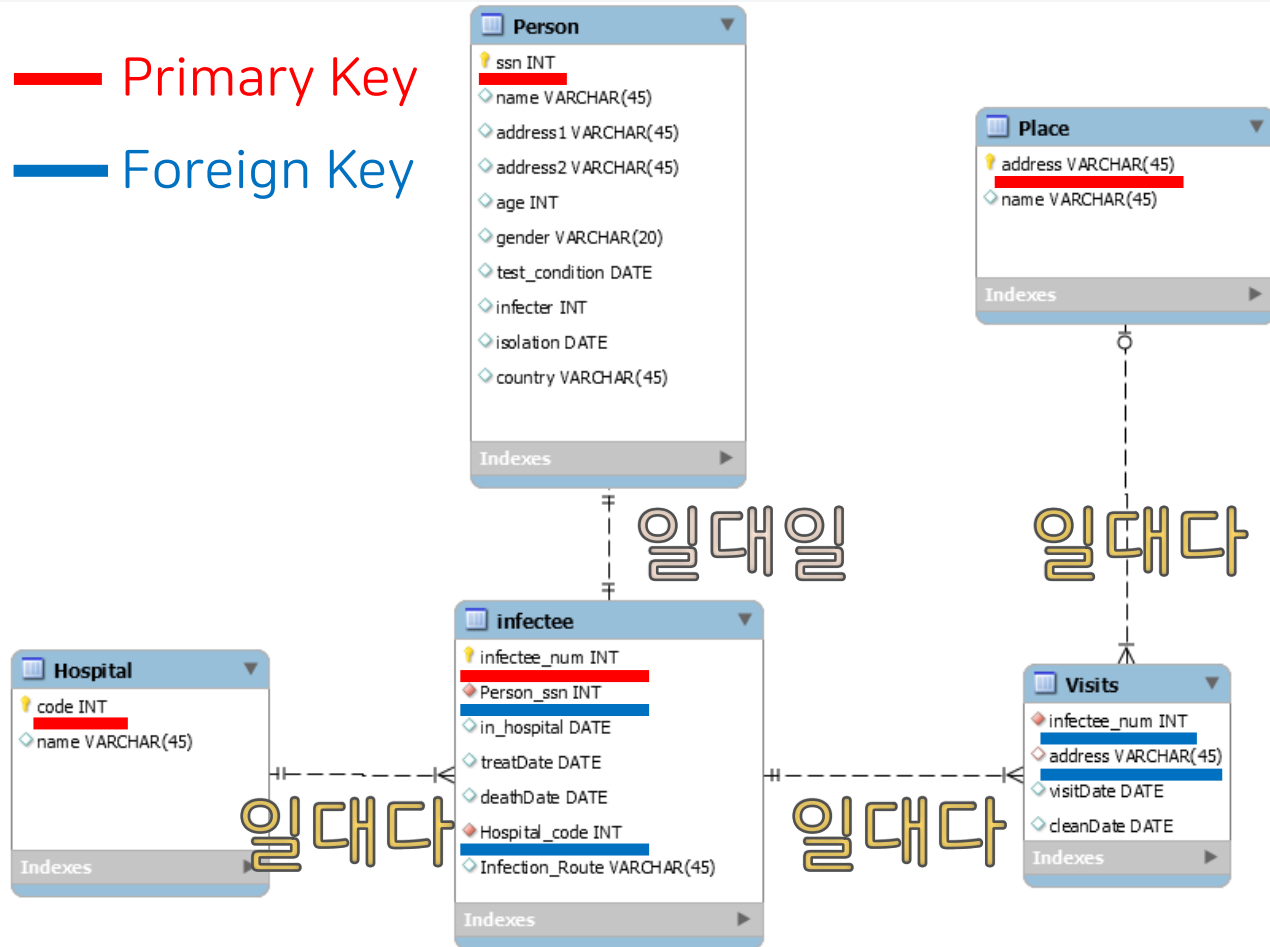
VO



# DB Modeling & Class Diagram

— Primary Key

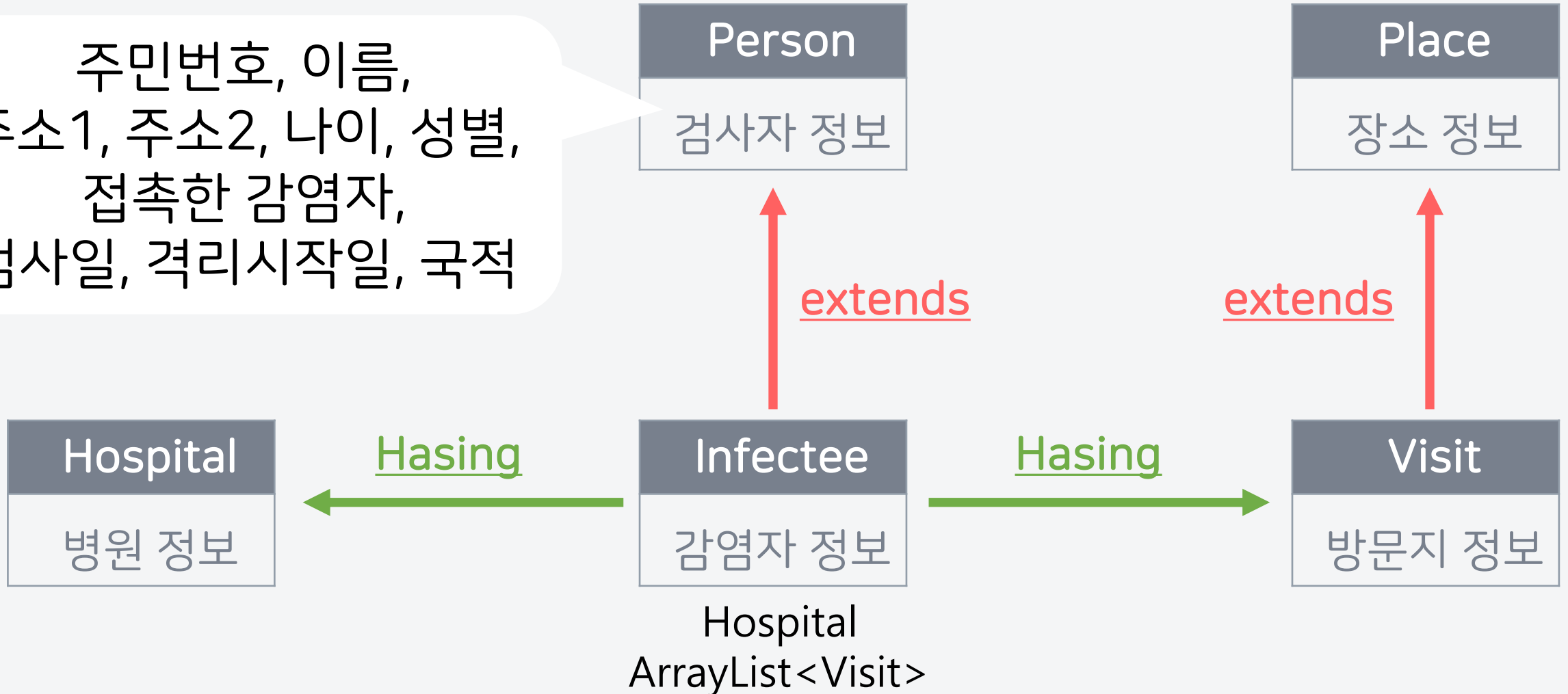
— Foreign Key



# Class Diagram

VO

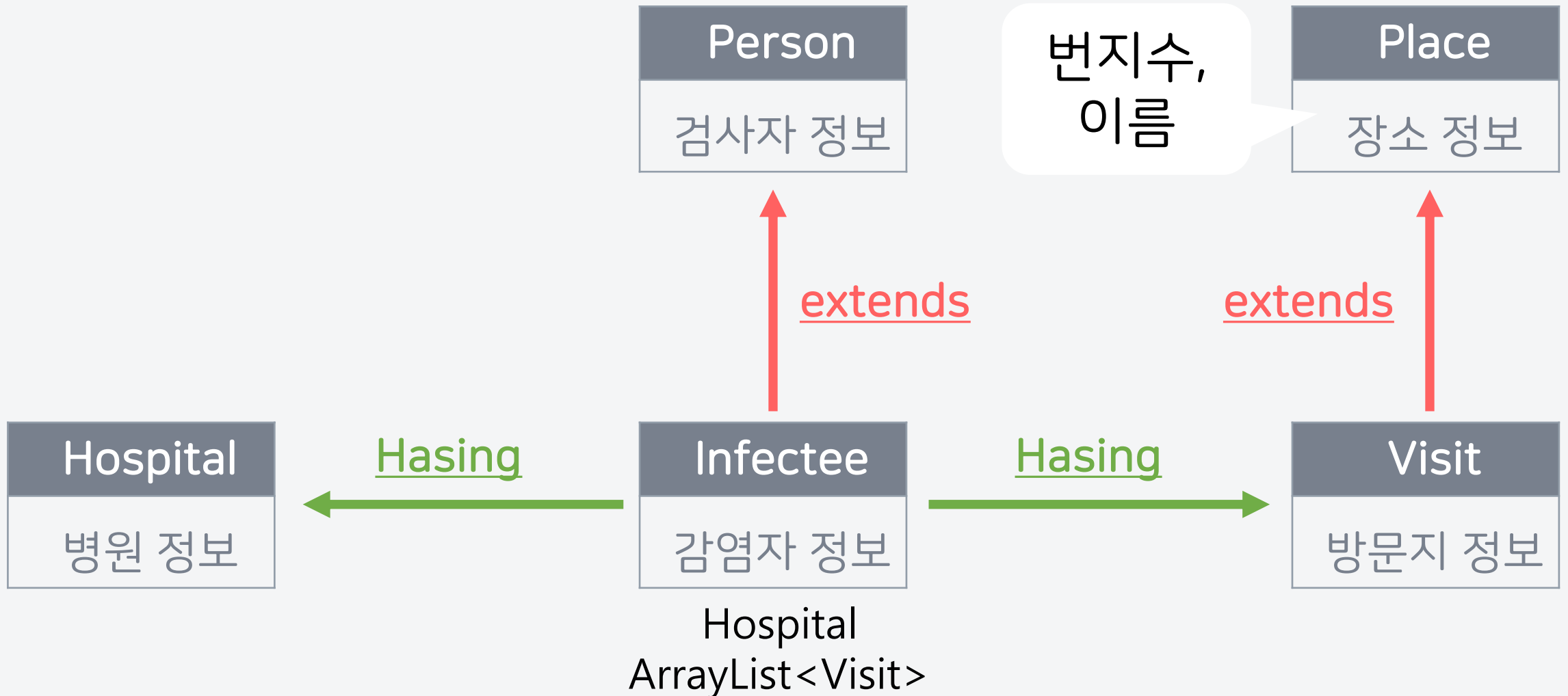
주민번호, 이름,  
주소1, 주소2, 나이, 성별,  
접촉한 감염자,  
검사일, 격리시작일, 국적





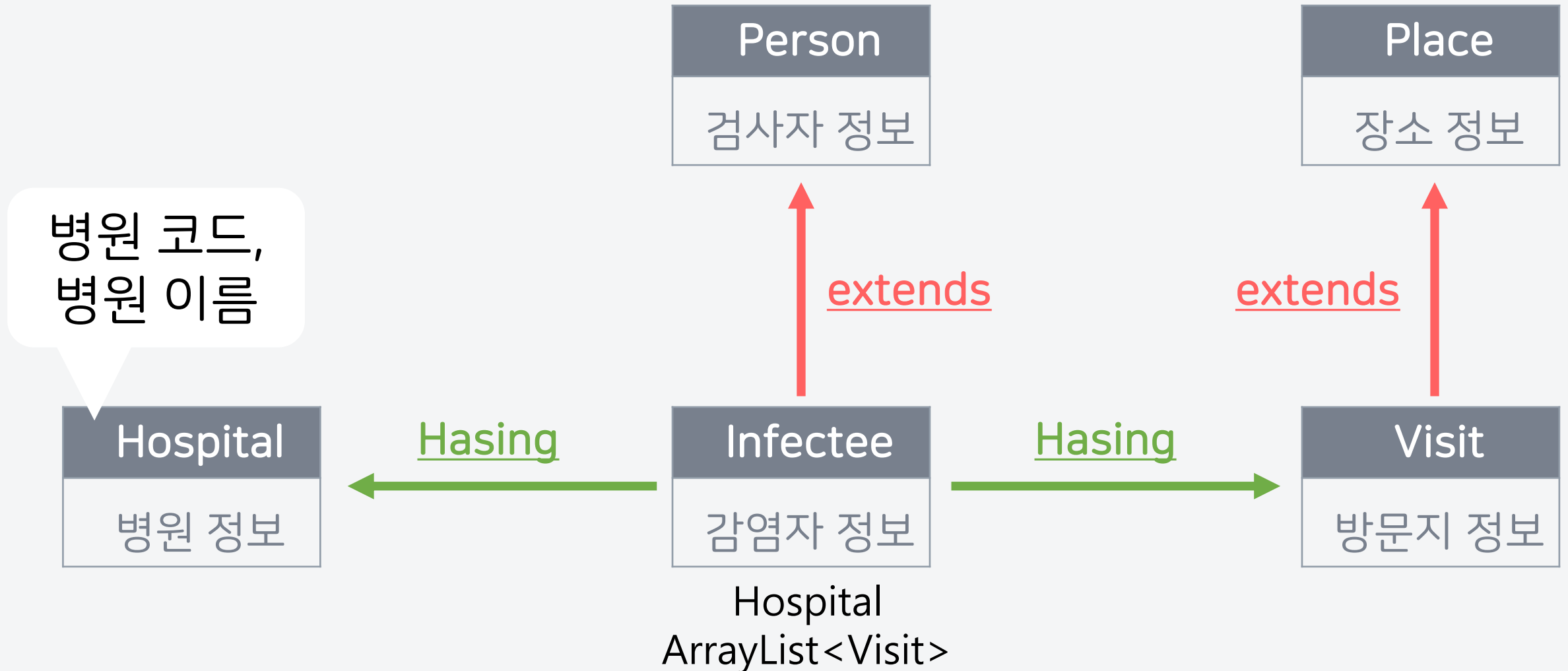
# Class Diagram

VO



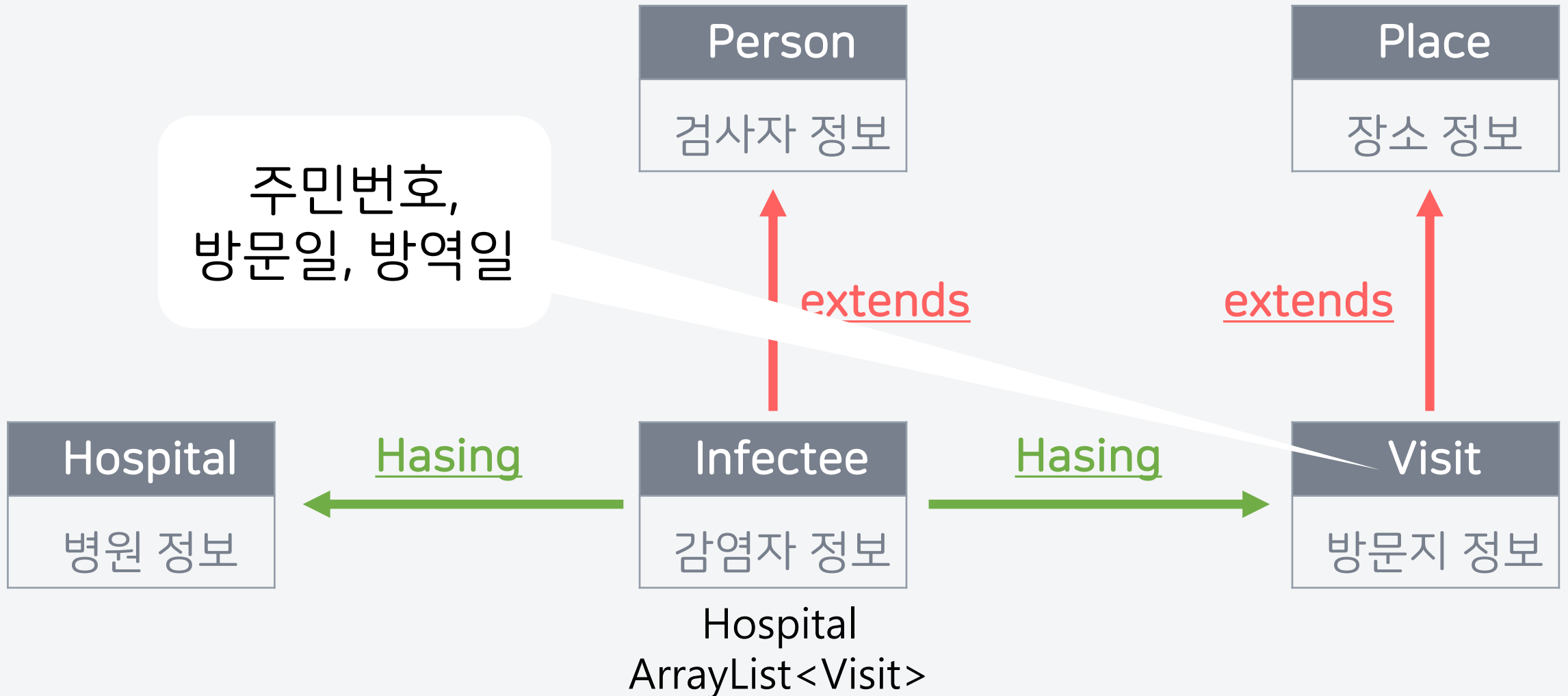
# Class Diagram

VO



# Class Diagram

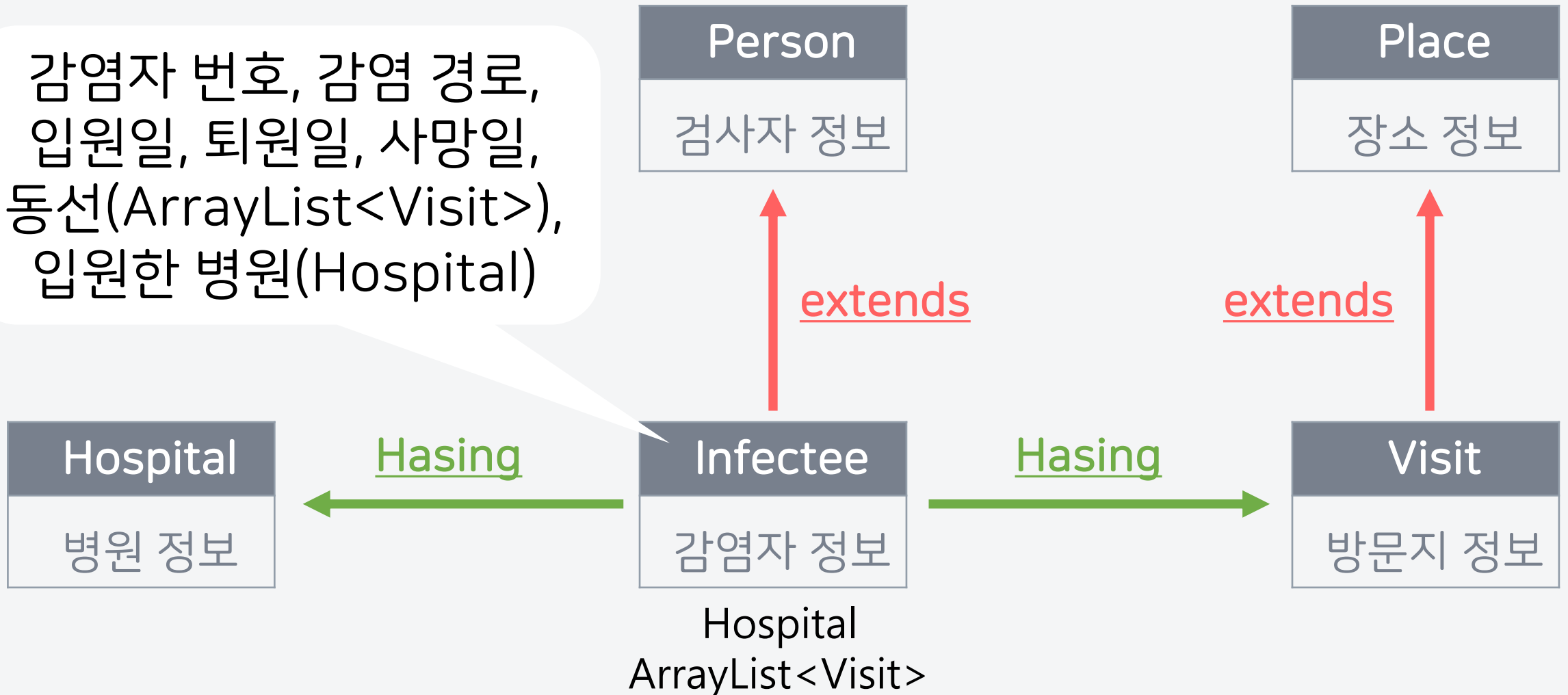
VO



# Class Diagram

VO

감염자 번호, 감염 경로,  
입원일, 퇴원일, 사망일,  
동선(ArrayList<Visit>),  
입원한 병원(Hospital)



# Class Diagram

DAO

Exception

Test

DatabaseTemplate

템플릿

RecordNotFound

사용자를 찾을 수 없을 때

Corona\_test

실행

DuplicateSSN

기존의 사용자를 추가할 때

Database

감염자 정보

implements



# Interface

일반사용자

현황 통계 조회

동선 조회

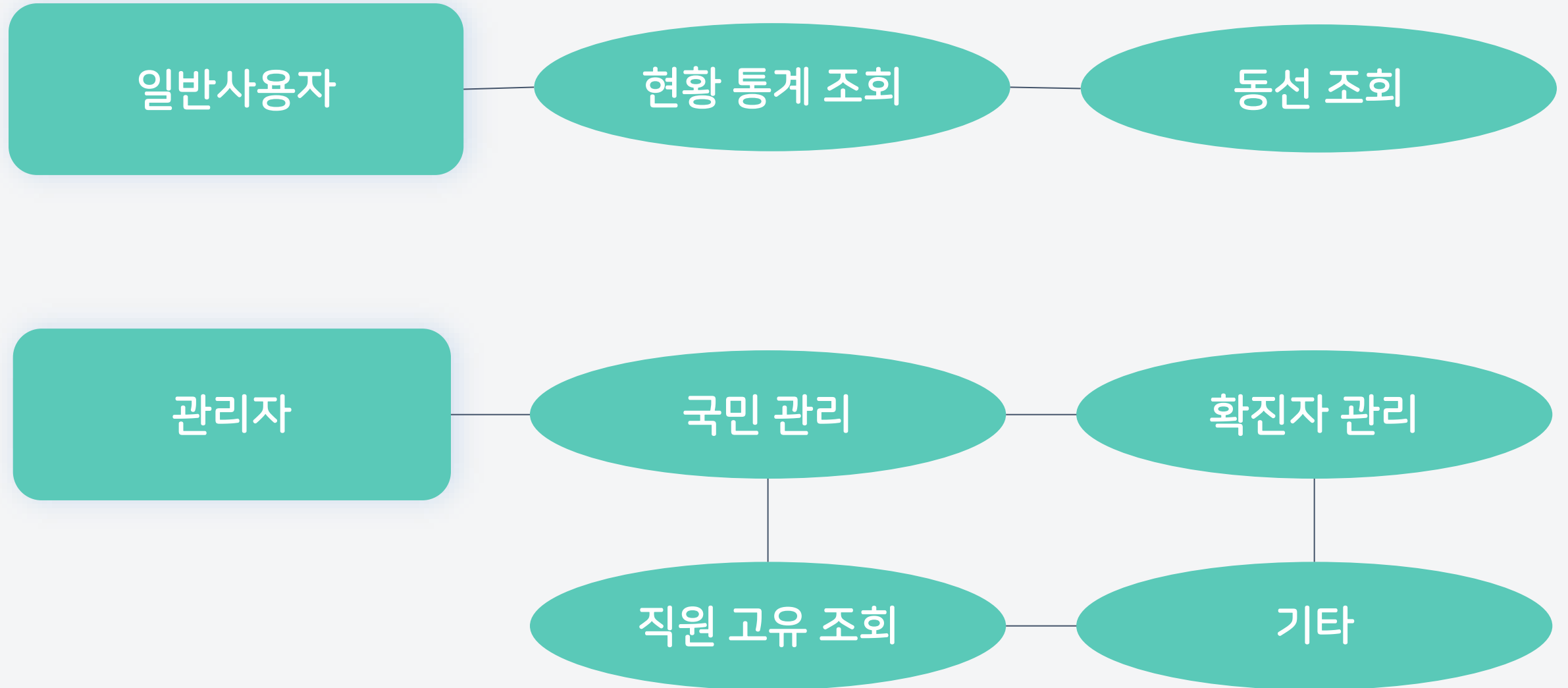
관리자

국민 관리

확진자 관리

직원 고유 조회

기타



## 1. 일반유저 조회

```
28  /* ----- (1)일반 유저 조회 ----- */
29  /* -- (1-1)코로나 현황 관련 조회 -- */
30  // B2 : 일일확진자수
31  int todayInfectee() throws SQLException;
32  // B3 : 일일완치자수
33  int todayTreated() throws SQLException;
34  // B4 : 일일사망자
35  int todayDeath() throws SQLException;
36  // B5 : 누적확진자
37  int totalInfectee() throws SQLException;
38  // B6 : 누적완치자
39  int totalTreated() throws SQLException;
40  // B7 : 누적사망자
41  int totalDeath() throws SQLException;
42  /* // B8 : 누적사망률
43  // B13 : 지역발생확진자
44  int localInfectee() throws SQLException;
45  // B14 : 해외발생확진자수
46  int foreignInfectee() throws SQLException;
47  // B15 : 광역시도별확진자
48  int infecteeInCertainCity(String address1) throws SQLException;
49  // B16 : 시구군별 확진자
50  int infecteeInCertainCity(String address1, String address2) throws SQLException;
51
52  /* -- (1-2)동선 및 방역 관련 조회 -- */
53  // B17 : 특정환자 동선 검색(지역 몇번 확진자)
54  ArrayList<Visit> searchAllTrace(String address1, String address2, int num) throws SQLException;
55  // B18 : 특정환자 동선 검색
56  ArrayList<Visit> searchAllTrace(int num) throws SQLException;
57  // B19 : 모든 방역 내역
58  ArrayList<Visit> isCleaned() throws SQLException; // 모든 방역 내역
59  // B20 : 특정 확진자가 간 곳의 방역 여부
60  ArrayList<Visit> isCleaned(int ssn) throws SQLException; // 특정 확진자가 간 곳
```

## 2. 직원 고유 조회

```
73  /* ----- (2)직원 고유 조회 ----- */
74  /* -- (2-1)Person 관련 조회 -- */
75  // B21 : Person 조회
76  Person searchPeople(int ssn) throws Exception;
77  // B22 : 국적조회(Person 테이블의 Country)
78  String searchForeigner(int ssn) throws SQLException, RecordNotFoundException;
79
80  /* -- (2-2)Infected 관련 조회 -- */
81  // B25 : 전체확진자
82      ArrayList<Infected> searchAllInfected() throws SQLException, RecordNotFoundException;
83  // B23 : 특정확진자 조회(주소1, 주소2, 확진번호)
84  Person searchAInfected(String address1, String address2, int num) throws SQLException, RecordNotFoundException;
85  // B24 : 특정 확진자 조회(확진번호)
86  Person searchAInfected(int num) throws SQLException, RecordNotFoundException;
87  // B26 : 해외발생확진자
88  ArrayList<Infected> searchAllForeignInfected() throws SQLException, RecordNotFoundException;
89  // B30 : 특정 확진자가 감염시킨 확진자 검색!
90  ArrayList<Infected> searchInfectedPeopleByCertainPerson(int num) throws SQLException, RecordNotFoundException;
91
92  /* -- (2-3)Hospital--Infected 관련 조회 -- */
93  // B27 : 병원이름으로 병원 코드 조회
94  int searchAHospitalCode(String name) throws SQLException, RecordNotFoundException;
95  // B28 : 확진자가 입원한 병원 조회
96  Hospital whereIsAInfected(int num) throws SQLException, RecordNotFoundException;
97  // B29 : 병원 내 모든 확진자 조회
98  ArrayList<Infected> searchAllInfectedInHospital(int code) throws SQLException, RecordNotFoundException;
99
100 /* -- (2-4)Visits--Infected 관련 조회 -- */
101 // B31 : 특정 장소 방역 조회
102 ArrayList<Visit> isCleaned(String address) throws SQLException; //
103 // B40 : ssn을 입력하고 동선 목록을 리턴 --> %%B18과 B40 겹침?
104 ArrayList<Visit> getTrace(int ssn) throws SQLException;
```



### 3. 확진자 관리 정보수정 & 방역

```
106  /* ===== 3.정보 수정 ===== */
107  /* -- (3-1)Person 관련 수정 -- */
108  // B32 : Person 테이블 수정
109  void updatePerson(Person p) throws SQLException;
110
111  /* -- (3-2)Infectee 관련 수정 -- */
112  // B33 : 확진자 추가 + 입원 처리 + 방역도 : 입원날짜 넣기
113  void addInfectee(int ssn, int code, String infectionroute, String date) throws SQLException, DuplicateSSNException;
114  // B34 : 완치 처리
115  void treatedInfectee (int ssn, String date) throws SQLException;
116  // B35 : 격리해제 : null = 격리된 적 없음. 2020년 이후 날짜 = 격리 중. '1999-12-31' = 격리해제
117  void NoMoreIsolation(int ssn) throws SQLException;
118  // B36 : 사망 처리
119  void setDeath(int ssn, String date) throws SQLException;
120
121  /* -- (3-3)Visits 관련 수정 -- */
122  // B37 : 동선 수정
123  void updateTrace(int num, String address, String visitDate,String oldaddress) throws SQLException;
124
125  /* ===== 4.방역 조치 ===== */
126  // B38 : 방역하기 : visit, place clean
127  void doClean(int num, String address, String visitDate, String cleanDate) throws SQLException ;
```

# Using Query

## 조인

광역시도별확진자 / 시구군별 확진자 / 특정환자 동선 검색(지역 몇번 확진자) /  
모든 방역 내역 / 특정 확진자가 간 곳의 방역 여부 / 특정 확진자가 간 곳 / 국적조회 /  
특정확진자 조회(주소1,주소2,확진번호) / 특정 확진자 조회(확진번호) / 전체확진자 /  
확진자가 입원한 병원 조회 / 병원 내 모든 확진자 조회 / 특정 지역 n번 확진자의 ssn을 리턴 /  
동선 목록(ssn) / 특정 도시환자의 감염번호를 받고 ssn을 리턴  
/ 감염번호를 받고특정도시 감염번호로 리턴 / 특정도시 감염번호를 받고 총감염번호로 리턴

# Using Query

## 서브쿼리

특정환자 동선 검색 / 특정 확진자가 간 곳의 방역 여부 /

특정 지역 n번 확진자의 ssn을 리턴 / 특정 도시화자의 감염번호를 받고 ssn으로 리턴

## 분석함수

특정 지역 n번 확진자의 ssn을 리턴 / 특정 도시화자의 감염번호를 받고 ssn으로 리턴 /

감염번호를 받고 특정도시 감염번호를 리턴 / 특정도시 감염번호를 총감연번호를 리턴

# Using Query

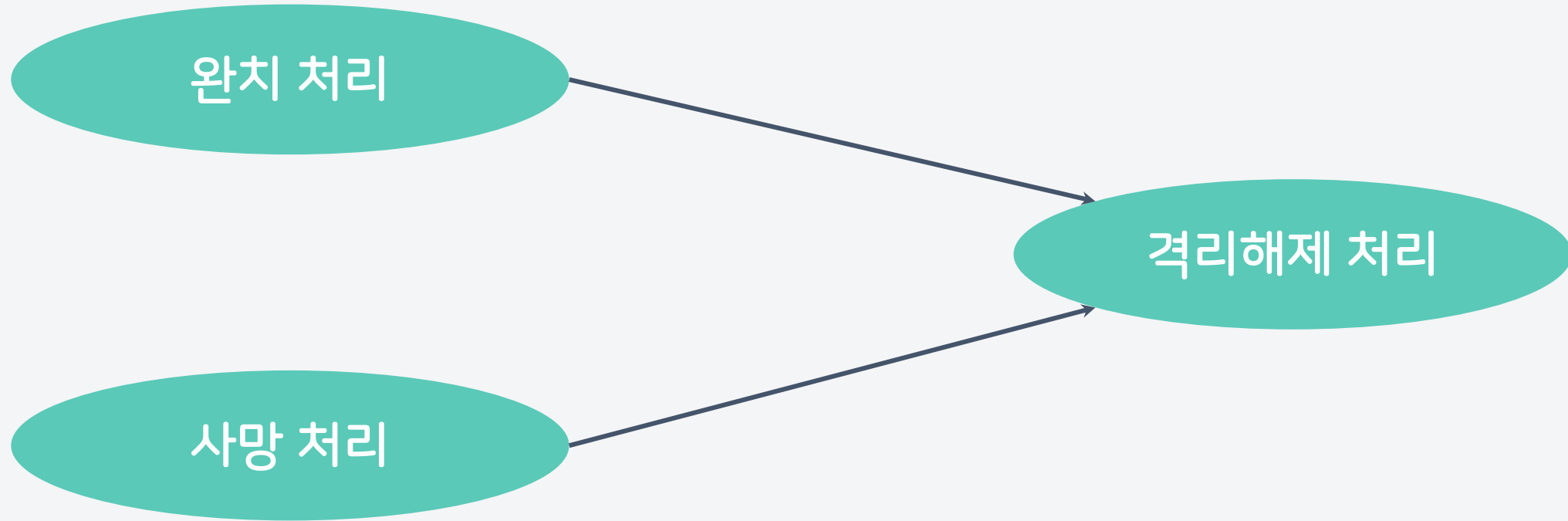
## 조인 & 서브쿼리 & 분석함수 예시

```
@Override
// B39 : 특정 지역 n번 확진자의 ssn을 리턴
public int returnSsn(String address1, String address2, int num) throws SQLException {
    int ssn = 0;
    Connection conn = getConnect();
    StringBuffer sb = new StringBuffer();
    sb.append("SELECT ssn FROM (SELECT address1, address2, ");
    sb.append("row_number() over(order by in_hospital ASC) no, ssn, in_hospital ");
    sb.append("FROM (SELECT infectee num, Person ssn, in hospital FROM infectee) i, ");
    sb.append("(SELECT ssn, address1, address2 FROM person) p ");
    sb.append("WHERE Person_ssn = ssn AND address1 = ? AND address2 = ?) temp ");
    sb.append("WHERE no = ?");
}
```

# Using Transaction

## 목적

정밀한 통계 및 정보 관리를 위해 Transaction 사용



```
public void treatedInfectee(int ssn, String date) throws SQLException{
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
        conn = getConnect();

        conn.setAutoCommit(false);

        String query = "UPDATE infectee SET treatDate = ? WHERE Person_ssn =?";
        ps = conn.prepareStatement(query);

        ps.setString(1, date);
        ps.setInt(2, ssn);

        System.out.println(ps.executeUpdate()+" 명 완치");

        String query2 = "UPDATE Person SET isolation = '1999-12-31' WHERE ssn =?";
        ps = conn.prepareStatement(query2);

        ps.setInt(1, ssn);

        System.out.println(ps.executeUpdate()+" 명 격리해제");
        conn.commit();
    }
}
```

```
}catch(Exception e){
    System.out.println("=====완치처리는 했으나 격리해제 처리가 안 됐습니다.=====");
    conn.rollback();
}finally {
    System.out.println("=====완치처리 후 정보 확인=====");

    String query3 = "SELECT p.ssn, i.treatDate, p.isolation FROM person p, infectee i WHERE ssn = ? AND  p.
ps = conn.prepareStatement(query3);
ps.setInt(1, ssn);

rs = ps.executeQuery();
while(rs.next()) {
    System.out.println(rs.getInt("ssn")+", "+rs.getString("treatDate")+", "+rs.getString("isolation"));
}
conn.setAutoCommit(true);
closeAll(ps, conn);
}
```

# UI Prototype



# 코로나바이러스감염증-19(COVID-19)

중앙사고수습본부

중앙방역대책본부

f

📺

📺

📺

English

中文

바로알기

발생동향

뉴스&이슈

홍보자료 & FAQ

피해지원정책

공지사항

생활 속 거리 두기

검색어를 입력하세요.



환자 현황 (5.17. 00시 기준, 1.3 이후 누계)

자세히 >

시도별 확진환자 현황 (05.17. 00시 기준, 1.3 이후 누계)

일일 확진자 13		일일 완치자 37	
확진환자	완치 (격리해제)	치료 중 (?)	사망
(누적) 11,050 전일대비 (+ 13)	= 9,888 (+ 37)	+ 900	+ 262 (+ 0)

검사현황		지역발생확진자: 56명
누적 검사수	747,653 건	해외유입확진자: 40명
누적 검사완료수	729,993 건	누적 사망률: 1.5%
누적 확진률	1.5 %	
(결과양성 / 총 검사완료수 * 100%)		

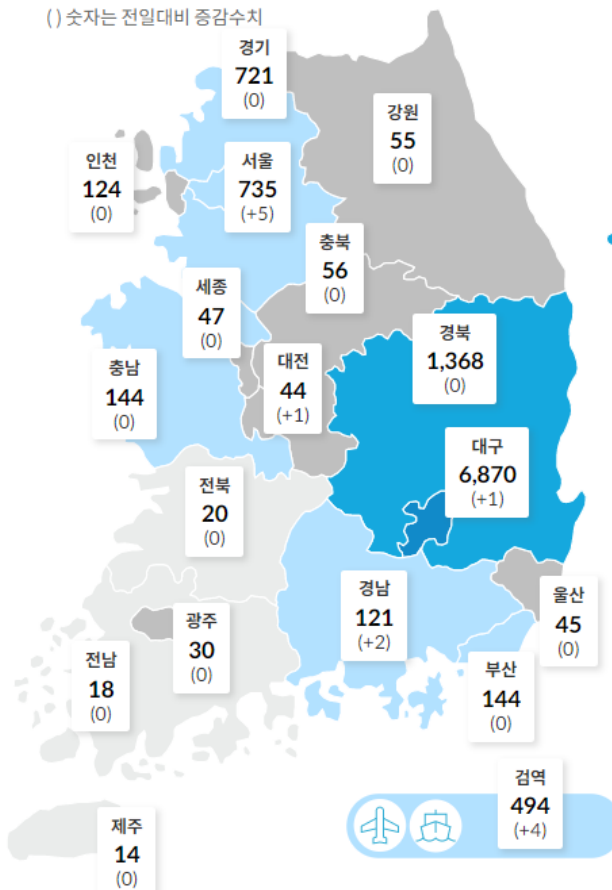
우리 동네 확진자 동선 검색:

지역

몇 번째 환자

검색

? 시도를 클릭하시면 상세 현황을 확인할 수 있습니다.  
( ) 숫자는 전일대비 증감수치



코로나바이러스감염증-19(COVID-19)

검사 추가	SSN 검사받은 날짜	이름 접촉자	주소1 접촉자	주소2 격리일	나이 격리일	성별 격리일	접촉자 국적	실행
확진자 추가	SSN	입원 병원 코드	입원날짜	감염 경로(국내/해외)				실행
치료 완료	SSN	치료 완료 날짜						실행

코로나바이러스감염증-19(COVID-19)

바로알기 발생동향 뉴스&이슈 홍보자료&FAQ 피해지원정책 공지사항 생활 속 거리 두기

검색어를 입력하세요.

격리  
해제

SSN

격리해제 날짜

실행

방역  
하기

지역 감염자 번호

주소1

주소2

지역 감염자 방문일

방역 날짜

실행

사망  
처리

SSN

사망 날짜

실행

# 코로나바이러스감염증-19(COVID-19)

🇰🇷 중앙사고수습본부 □ 🇰🇷 중앙방역대책본부 □ f 📺 📺 • English • 中文

바로알기 발생동향 뉴스&이슈 홍보자료&FAQ 피해지원정책 공지사항 생활 속 거리 두기

검색어를 입력하세요.



검사 받은  
사람 조회

SSN

실행

특정 확진자  
정보조회

SSN

실행

전체 확진자  
정보조회

실행

국적 조회

SSN

실행

코로나바이러스감염증-19(COVID-19)

중앙사고수습본부 중앙방역대책본부 f yt g+ t • English • 中文

바로알기 발생동향 뉴스&이슈 홍보자료&FAQ 피해지원정책 공지사항 생활 속 거리 두기

검색어를 입력하세요.



병원코드조회

병원 이름 실행

병원 내  
확진자 정보

병원코드 실행

입원 환자  
병원 조회

SSN 실행

확진자 방문지  
방역내역 조회

SSN 실행

[바로알기](#)

[발생동향](#)

[뉴스&이슈](#)

[홍보자료 & FAQ](#)

[피해지원정책](#)

[공지사항](#)

[생활 속 거리 두기](#)

검색어를 입력하세요.



확진자  
동선 조회

SSN

실행

모든 방역  
내역 조회

실행

특정확진자가  
감염시킨  
확진자 정보

SSN

실행

해외국적  
확진자

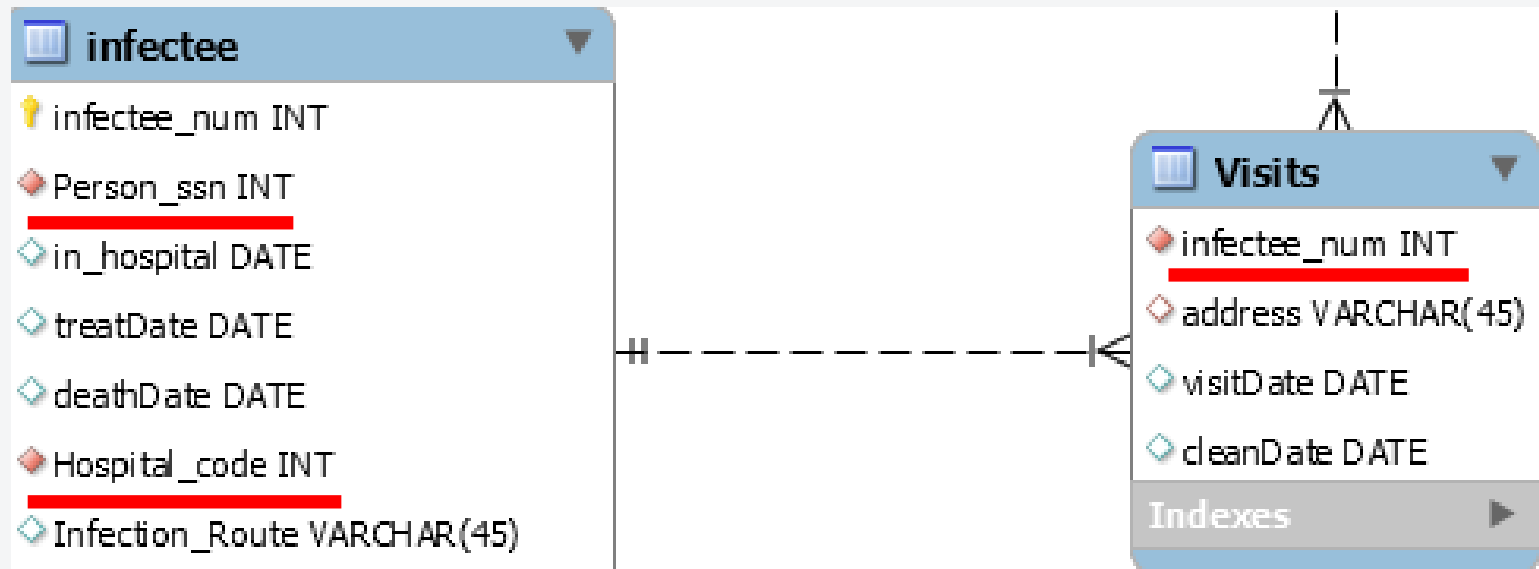
SSN

실행

# 시행착오

## Modeling

- 필요한 컬럼이 계속적으로 추가됨
- 해당 컬럼이 양측 엔터티에 중복적으로 요구될 때 어디에 넣어야 할지
- 외래키 설정 없이 관계 지으려고 해서 오류가 남
- Identifier 불일치 : DB 테이블 엔터티 속성명, 클래스 필드명



# 시행착오

GitHub

- Git push시 계속적인 에러

```
$ git push origin master
remote: You must verify your email address.
remote: See https://github.com/settings/emails.
fatal: unable to access 'https://github.com/windyagain/p4.git/': The requested URL returned error: 403
```

→ 에러 메시지를 그대로 복사-붙여넣기 후 검색



# 평가

## 잘 되었던 점

- 의견 수렴 과정에서 팀원들 간에 목표 설정이 잘 이루어졌다.
- DB 설계에 신경을 많이 써서 코드 작성 시간이 단축되었다.
- 레코드를 미리 추가해서 테스트 작업이 용이했다.
- 역할을 나누되 먼저 끝내고 여유가 있는 사람이 작업을 도왔다.

## 향후 보완할 점

- Git에 대한 추가적인 공부와 연습이 필요하다.
- 여러 메소드에서 쓰이는 공통 메소드는 먼저 구현해야 한다.
- 비슷한 메소드끼리 분류한 후 역할을 나눠야 한다.
- toString()은 최대한 늦게 하는 게 좋겠다.(필드 및 상속 관계 변경)