

K_COVID 19 TEAM PROJECT

팀장 : 김재성 2018117736

팀원 : 방준환 2018117663

김한영 2015114563

실행 환경	Wampserver 3.2.3 / php 7.3.21 / Mysql 5.7.31 / Apach 2.4.46 / python 3.7.7 / Pycharm / VS Code / Chrome 87.0.4280.66	
1 주차		
조건	<div>1. 포트 번호 3306</div> <div>2. K_COVID19 Database가 있는 user, password로 파이썬코드 변경</div> <div>3. 같은 폴더 안에 K_COVID.csv파일이 있어야 함</div> <div>4. table 생성->python code로 parsing->foreign key추가(생략 가능)</div>	
각 구분된 데이터에 해당하는 축적 데이터를 만들기 위한 준비		
<pre># checking duplicate patient_id & checking patient_id == "NULL" if (((line[col_list['region_code']], line[col_list['wdate']]) in region_code_wdate) or (line[col_list['region_code']] == "NULL") or (line[col_list['wdate']] == "NULL")): continue else: region_code_wdate.append((line[col_list['region_code']], line[col_list['wdate']]))</pre>		
<div>weather table에 데이터를 Insert하는 python parsing 코드에서 Super key처리를 위해 region_code와 wdate의 튜플을 사용하여 두 개의 데이터가 동시에 중복이 되는 경우만 배제하도록 코드 작성</div> <div>※ 이외의 코드는 예시 코드와 비슷하므로 생략</div>		
2 주차		
조건	<div>1. 포트 번호 3306</div> <div>2. K_COVID19 Database 가 있는 user, password로 파이썬 코드 변경</div> <div>3. 같은 폴더에 K_COVID.csv, additional_Timeinfo 파일이 있어야 함</div> <div>4. table 생성->python code로 parsing->foreign key추가(생략 가능)</div>	
각 구분된 데이터에 해당하는 축적 데이터를 만들기 위한 준비		
<pre># 데이터프레임 사용 # province 의 중복되지 않은 값 생성 df = pd.DataFrame(data['province']) df = df.drop_duplicates().dropna(axis=1) province_unique = df['province'].tolist()</pre>		<div>해당하는 Column의 데이터로 구분하기 위해 중복되지 않은 데이터들의 정보가 필요하기에 데이터프레임을 만들어 사용</div> <div>※ NULL값의 데이터는 의미가 없다고 판단하여 생략</div>

각 구분된 데이터에 해당하는 축적 데이터를 만들기 위한 준비

```
total_confirmed = {} # 총 확인된 감염자 수
total_released = {} # 총 완치된
total_deceased = {} # 총 사망자

# 각 province 에 대한 확진자, 완치자, 사망자 수
for province in province_unique:
    total_confirmed[province] = 0
    total_released[province] = 0
    total_deceased[province] = 0
```

조건에 따라 데이터를 축적하기 위해서 앞서 구분된 unique값으로 구분된 dictionary 생성 및 값을 더해나가기 위해서 0으로 초기화

Composite key 처리

```
for i, line in enumerate(file_time_read):

    # Skip first line
    if not i:
        continue

    for province in province_unique:
        # composite key (date, province) 중복확인 및 null값 배제
        if (line[col_list['date']], province) in date_province \
            or (line[col_list['date']] == "NULL" or province == "NULL"):
            continue
        else:
            date_province.append((line[col_list['date']], province))
```

Composite key는 Super Key와 같이 두 가지 데이터가 모두 중복이 되어야 중복으로 정의한다. 따라서 두 가지 데이터를 따로 받아오기 때문에 단일 반복문으로 비교할 수 없었고 그래서 이중 반복문을 사용하여 두 개의 데이터에 대한 중복값이 있는지 확인하여 Composite key를 처리한다. 또한 뒤의 코드에서 첫 번째 데이터에 대해 모든 두 번째 데이터에 해당하는 축적값을 Insert 해야하므로 이중반복문을 사용하였다.

※ 이외의 코드는 예시 코드와 비슷하므로 생략

foreign key를 위한 composite key 설정하지 않기

```
create table TimeProvince(
    date DATE,
    province varchar(50),
    confirmed INT,
    released INT,
    deceased INT
);
```

foreign key는 null값으로 설정되어 있거나 primary key로 설정되어 있으면 오류가 발생할 가능성이 커짐으로 composite key이지만 할당하지 않는다.

3 주차	
조건	<ol style="list-style-type: none"> 1. K_COVID19 Database 가 있는 user, password로 php 코드 변경 2. 출력하기 위한 데이터들이 데이터베이스에 Insert 되어진 상태이어야 한다. 3. php파일이 C:\wamp64\www에 있어야한다.
<p>※첫 번째 과제는 예시 파일과 비슷한 부분이 많으므로 생략</p> <p>※두 번째 과제는 하나의 파일이 다른 파일들과 비슷한 부분이 많으므로 코드 설명 생략</p>	
초기 상태 구분	
<pre>\$action = ''; if(isset(\$_POST['action']))\$action = \$_POST['action']; if(\$action == 'form_submit') <input type="hidden" name="action" value="form_submit" /></pre>	
<p>초기 상태인 경우 아직 attribute에 대한 값이 설정되지 않은 상태이므로 모든 정보를 출력하는 기본페이지로 나타내야한다. 따라서 action이라는 값이 form_submit인지 아닌지로 기본페이지와 선별되어진 페이지를 구분짓는 역할을 하기 위해 작성되었다.</p>	
선택지 생성	
<pre>\$infection_case_array = array(); while(\$row_list = mysqli_fetch_assoc(\$result)) { array_push(\$infection_case_array, \$row_list['infection_case']); } \$infection_case_unique_array = array_unique(\$infection_case_array);</pre>	
<p>선택지에 해당하는 값들을 하드코딩하는 방법도 있으나 유지보수 측면에서 비효율적이라 판단하여 데이터베이스에서 Column의 데이터를 불러와 중복된 값을 제거하는 코드</p>	
선택지 드롭다운 만들기	
<pre>foreach (\$infection_case_unique_array as \$infection_case){ if(\$infection_case != '') { echo '<option value="", \$infection_case, ">', \$infection_case, '</option>'; } else { echo '<option value="", \$infection_case, ">NULL</option>'; } }</pre>	
<p>선택지에 대한 사용자 인터페이스를 제공하기 위해 드롭다운을 선택하고 선택지를 입력하도록 하였다. 하지만 NULL값을 경우 비어있는 칸으로 출력되기 때문에 조건처리하여 NULL 문자열을 출력하도록 하였다.</p>	

선택지에 따른 데이터 출력

```
if($current_select == '-') {
    $sql = "select * from patientinfo";
} elseif($current_select == '') {
    $sql = "select * from patientinfo where infection_case is null";
} else {
    $sql = "select * from patientinfo where infection_case=\"".$current_select."\"";
}
```

sql의 구문의 경우 NULL값과 아닌 값을 출력하는 구문이 다르고 전체 데이터를 출력하는 것과 일정 부분을 출력하는 구문이 다르기 때문에 해당 구문들의 구분을 위해 current_select의 값에 따라 if문을 사용하여 구분짓게 하였다.

Table	Attribute	Content
Patientinfo	infection_case	infection_case별로 나이대, 성별, 사람의 수나 사망여부 및 전염 여부를 파악할 수 있음
caseinfo	province	province 별로 infection_case의 종류나 개수 및 확진자의 수를 파악할 수 있어 province별로 case가 어디가 가장 많은지 파악할 수 있음
region	province	province 별로 구분지음으로써 작은 city들을 모아볼 수 있어 그 province의 좀 더 세분화된 정보를 얻을 수 있고 그 정보들을 비교할 수 있음
weather	province	province별 날짜에 따른 기온의 변동을 파악할 수 있음

3 주차 (3)

기생성된 view, table에 대한 예외처리

```
<?php

$sql = "drop view table1";
mysqli_query($link,$sql);
$sql = "drop view result";
mysqli_query($link,$sql);
$sql = "drop table tempo";
mysqli_query($link,$sql);
```

앞으로 사용할 table과 view를 만들기 앞서 이미 생성되어 있을 수도 있는 테이블 및 view를 삭제하는 예외처리 작업 코드

view table1 생성

```
$query = "create view table1 as
select c.infection_case as c1 ,c.province as c2, c.city as c3, r.academy_ratio as c4, r.elderly_population_ratio as c5, c.confirmed as c6
from caseinfo c, region r, patientinfo p
where c.province = r.province and c.infection_group = 1 and p.infection_case=c.infection_case
group by c.infection_case
order by confirmed desc";
mysqli_query($link,$query);
```

infection_case가 발생한 경우(infection_group=1), patientinfo, caseinfo, region 테이블로부터 Infection_case, Province, city, academy_ratio, elderly_population_ratio, confirmed를 select 한 뒤 infection_case를 기준으로 group by로 묶어 view TABLE1을 만드는 코드이다. 이후 만들어질 table TEMPO와 JOIN 하여 결과 table인 RESULT를 작성하기 위해 만들었다.

TABLE1과 JOIN 하기위해 table TEMPO생성

```
$query = "create table tempo(
infection_case char(70),
age char(10),
primary key(infection_case));
mysqli_query($link,$query);
```

결과적으로 생성되는 RESULT table에는 infection_case를 기준으로 Province, city, academy_ratio, elderly_population_ratio, confirmed, age (the greater part of, 가장 많은 감염세대)가 있어야 한다.

그래서 이전에 만들었던 table1과 JOIN 하기 위해 만드는 table tempo는 infection_case를 primary key로 가져 JOIN 할 수 있도록 하고 추가할 column인 age (the greater part of, 가장 많은 감염세대)를 가지도록 만들었다.

table tempo에 값을 넣기 위해 infection_case와 age(가장 많은 감염세대) 계산 및 INSERT

```
$sql = "select distinct c.infection_case
from patientinfo p, caseinfo c
where p.infection_case is not null and c.infection_group=1";
$result=mysqli_query($link,$sql);

while($row=mysqli_fetch_array($result)){
    $variable=$row['infection_case'];
    $query="insert into tempo (infection_case,age)
select t.ic,t.a1 from
(select p.infection_case as ic, p.age as a1
from patientinfo p
where p.infection_case='".$variable.'"
group by p.age
order by count(p.age) desc limit 1
)as t ";
    mysqli_query($link,$query);
}
```

이전에 만들어두었던 TEMPO table에 값을 넣기 위해 CASEINFO table에서 집단감염된 infection_case를 중복없이 추출하는 코드이다. 실행시킨 결과를 \$result에 넣고 INSERT query를 통해 \$result에 들어있는 각각의 infection_case의 가장 많은 감염비율의 세대와 infection_case를 SELECT하여 하나씩 TEMPO table에 넣는 코드이다.

TABLE1과 TEMPO를 JOIN 하여 RESULT table 생성

```
$query="create view result as
select t1.c1, t1.cz,t1.c2,t1.c3, t1.c4, t1.c5, tempo.age
from table1 t1
join tempo
on t1.c1=tempo.infection_case";
mysqli_query($link,$query);
```

앞서 구했던 TABLE1은 infection_case로 group by 하여 만든 view이고, TEMPO 역시 infection_case를 primary key로 하고 있기에 JOIN의 조건으로 infection_case를 연결하여 JOIN 하는 코드이다.

4 주차 - python

조건

1. 포트 번호 3306
2. K_COVID19 Database가 있는 user, password로 파이썬코드 변경
3. 같은 폴더 안에 Hospital.csv, Region.csv파일이 있어야 함
4. table 생성->python code로 parsing->foreign key추가(생략 가능)
5. patientinfo의 테이블에 데이터가 저장되어 있어야 함
6. Hospital.csv 컬럼 중 Hospital name -> Hospital_name으로 변경

euclidean 거리 값 함수

```
def get_euclidean_distance(x1, y1, x2, y2, round_decimal_digits=5):
    """
    유클리안 Formula 이용하여 (x1,y1)과 (x2,y2) 점의 거리를 반환
    """
    if x1 is None or y1 is None or x2 is None or y2 is None:
        return None
    assert isinstance(x1, numbers.Number) and -180 <= x1 and x1 <= 180
    assert isinstance(y1, numbers.Number) and -90 <= y1 and y1 <= 90
    assert isinstance(x2, numbers.Number) and -180 <= x2 and x2 <= 180
    assert isinstance(y2, numbers.Number) and -90 <= y2 and y2 <= 90

    dLon = abs(x2 - x1) # 경도 차이
    if dLon >= 180: # 반대편으로 갈 수 있는 경우
        dLon -= 360 # 반대편 각을 구한다
    dLat = y2 - y1 # 위도 차이
    return round(math.sqrt(pow(dLon, 2) + pow(dLat, 2)), round_decimal_digits)
```

병원과 환자를 연결하기 위해서 가장 가까운 거리의 병원을 찾아야 하는데 거기서 거리를 구하기 위한 공식을 함수로 작성

Patientinfo에 병원 ID Column 추가

```
try:
    sql = '''ALTER TABLE patientinfo ADD hospital_id bigint'''
    cursor.execute(sql)
except:
    print("이미 컬럼이 추가되었습니다.")
```

환자에게 할당된 병원을 기록하기 위해 병원 ID 컬럼을 sql구문을 활용해 추가한다.

현재 수용인원의 저장을 위한 DICTIONARY 생성

```
now = {} # 현재 수용인원 dictionary 만들기 & 중복된 case 제거를 위해 checking list & variable
for index, row in data_hospital.iterrows():
    # checking duplicate hospital_id & checking hospital_id == "NULL and calculating distance"
    if (row['Hospital_id'] in now.keys()) or (row['Hospital_id'] == "NULL"):
        continue
    else:
        now[row['Hospital_id']] = row['now']
```

현재의 수용인원을 저장하기 위해 Hospital.csv로부터 id와 now 값을 받아와 dictionary 형태로 저장

City의 값에 따른 지역 정보

```
region = '' # patient의 위도 경도를 위한 위치
if str(patient['city']) == "etc" or patient['city'] == None:
    region = data_region[data_region['province'] == patient['province']]
    region = region.head(1) # 대푯값 추출
else: # 다른 province더라도 같은 이름의 city가 존재할 수 있으므로 데이터 에러 가능성 배제
    region = data_region[
        (data_region['city'] == patient['city']) & (data_region['province'] == patient['province'])]
```

모든 데이터에 City값이 존재하는 것이 아니므로 City가 etc이거나 null값이면 대표 위치 값을 설정하고 City가 존재한다면 City의 위치값을 region 변수에 저장

환자별 모든 병원의 거리를 계산하여 오름차순으로 정렬

```
distance = {}
for index, row in data_hospital.iterrows():
    # checking duplicate hospital_id & checking hospital_id == "NULL and calculating distance"
    # 중복된 case 제거를 위해 checking list & variable
    if (row['Hospital_id'] in distance.keys()) or (row['Hospital_id'] == "NULL"):
        continue
    else: # hospital 과 patient의 거리
        try:
            distance[row['Hospital_id']] \
                = GeoUtil.get_euclidean_distance(float(region['longitude']),
                                                    float(region['latitude']),
                                                    float(row['Hospital_longitude']),
                                                    float(row['Hospital_latitude']))
        except: # region의 city와 patient의 city의 명칭이 다른 경우 예외처리
            continue

# 거리순으로 오름차순으로 정렬
sorted_distance = sorted(distance.items(), key=(lambda x: x[1]))
```

환자가 가까운 병원부터 선택될 수 있도록 모든 병원과의 거리를 계산하고 리스트로 저장한 다음 리스트를 거리를 기준으로 오름차순으로 정렬한다.

※ Region.csv파일과 저장되어있는 Patientinfo의 City의 명칭이 다른 경우가 있었으므로 예외처리하여 사용하지 않음

가장 가까우면서 정원미달인 병원 선택

```
# 수용인원이 가득차지 않으면서 가까운 병원 선택
for hospital_id, distance in sorted_distance:
    selected_hospital = data_hospital[data_hospital['Hospital_id'] == hospital_id]

    if int(selected_hospital['capacity']) > now[hospital_id]:
        now[hospital_id] += 1
        sql = '''UPDATE patientinfo set hospital_id=''' + str(hospital_id) + ''' where patient_id=''' \
            + str(patient['patient_id'])
        try:
            cursor.execute(sql)
            print("[OK] Inserting [%s] to patient : %s" % (hospital_id, patient['patient_id']))
        except (pymysql.Error, pymysql.Warning) as e:
            # print("[Error] %s"%(pymysql.IntegrityError))
            if e.args[0] == 1062: continue
            print('[Error] %s | %s' % (hospital_id, e))
            break
    break
```

거리가 짧은 병원부터 정원이 미달이면 그 병원에 입원하고 아니라면 다음으로 짧은 병원에 배정되도록 하기 위함이고 배정된 경우 patientinfo의 새로 생긴 병원 ID 컬럼에 배정 받은 병원의 ID가 저장되고 그 병원의 현재 수용인원의 숫자를 하나 증가시킨다. 이 수용인원들은 데이터베이스에 병원의 정보를 저장할 때 쓰이게 된다.

※ 데이터베이스에 저장하는 코드는 2주차에서 사용된 코드임으로 생략한다.

4 주차 - PHP + MAP 추가코드

조건

1. K_COVID19 Database가 있는 user, password로 PHP 변경
2. patientinfo 또는 hospital 테이블에 데이터가 저장되어 있어야 함

검색 입력 칸 생성 및 입력을 변수에 저장

```
if($action == 'form_submit') {
    $current_search = $_POST['patient_id'];
    echo '<xmp>';

    if($current_search == NULL){}
    else {
        echo 'Searching patient information by '.$current_search;
    }
    echo '</xmp>';
}
Put Patient_id : <input name="patient_id">
</input>
```

patient_id를 받아와 검색하기 위해 입력할 수 있는 <input>을 작성하였고 input에 해당하는 결과를 current_search에 저장하였다.

patient_id 에 맞는 row 가져오기

```
<?php

$sql = "select * from patientinfo where patient_id='".$current_search.'"";

$result = mysqli_query($link,$sql);
while( $row = mysqli_fetch_assoc($result) )
{
    print "<tr>";

    foreach($row as $key => $val)
    {
        if($key!="hospital_id")
        {
            print "<td>" . $val . "</td>";
        }
        else {
            print "<td><a href='search_patient_id_with_map.php?hospital_id=".$val."'>".$val."</a></td>";
        }
    }
    print "</tr>";
}
?>
```

선택한 patient_id에 맞는 row를 하나 가져오는 코드입니다.

hospital_id 부분에 링크를 걸기위해 반복문에서 hospital_id가 아니면 출력하게 해놓았고 마지막 hospital_id는 출력할 때 <a href>를 사용해 링크를 걸어주었습니다.

hospital_id 이용해서 위도와 경도 구하기

```
<?php
$lat=0;
$lng=0;

$hospital_id = NULL;
if(isset($_GET['hospital_id'])) {
    $hospital_id = $_GET['hospital_id'];

    $sql = "select Hospital_latitude, Hospital_longitude from hospital where hospital_id=$hospital_id";
    $result = mysqli_query($link,$sql);
    while( $row = mysqli_fetch_assoc($result) )
    {
        print "<tr>";

        foreach($row as $key => $val)
        {
            if($lat==0){$lat=$val;}
            else{$lng=$val;}
        }
        print "</tr>";
    }
}
?>
```

링크를 누르게 되면 \$hospital_id값이 넘어오면서 다시 같은 창으로 이동하게 되는데 그때 넘어온 값을 이용해서 다시 데이터를 뽑아 옵니다. 이번에는 위도와 경도를 뽑아오기 위해서 where 조건에 hospital_id를 넣고 select로 위도와 경도를 가져왔습니다.

MAP 추가코드

```
<div id="map" style="width:100%; height: 100vh;"></div>
<script async defer src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCgfsIMFbCsHb7D8fCE5wrWGJMgpOQBIEE&callback=initMap"></script>
<script>
    var lat1 =0;
    var lat1 = <?= $lat ?>;
    var lng1 = <?= $lng ?>;
    if(lat1!=0){
        function initMap() {
            var hospital = { lat: lat1 ,lng: lng1 };
            var map = new google.maps.Map(
                document.getElementById('map'), {
                    zoom: 17,
                    center: hospital
                });
            var marker = new google.maps.Marker({position: hospital, map: map});
        }
    }
</script>
</body>
```

위에서 저장한 위도와 경도 변수인 \$lat 와 \$lng는 php에서 사용한 변수라 자바스크립트에서 사용할 수 없었습니다. 자바스크립트에서 사용하기 위해서 변수 선언할 때 <?=\$변수이름 ?>를 사용해서 가져왔습니다.

initMap 함수를 이용해 지도를 생성하고 marker를 사용해 지도에 표시 마크를 생성했습니다. 그리고 if로 hospital_id가 넘어오지 않았을 때는 지도가 나오지 않도록 설정하였습니다.