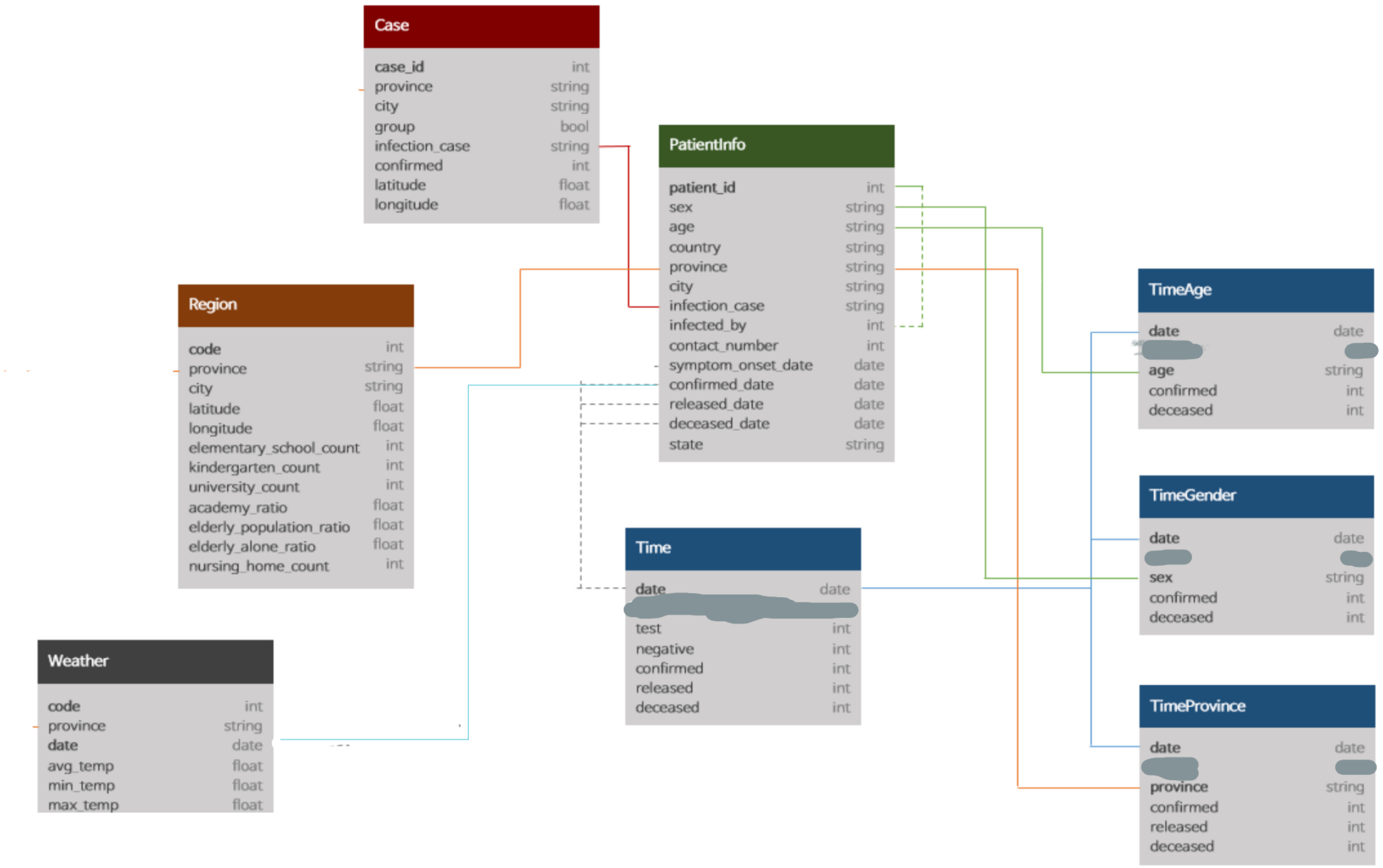
Team Project. K_COVID19

Donghyeon Kim, Donghee Kim, Inuk Jung
College of IT Engineering, School of Computer Science and Engineering
2020-2

Add “Time table”



Case	
case_id	int
province	string
city	string
group	bool
infection_case	string
confirmed	int
latitude	float
longitude	float

PatientInfo	
patient_id	int
sex	string
age	string
country	string
province	string
city	string
infection_case	string
infected_by	int
contact_number	int
symptom_onset_date	date
confirmed_date	date
released_date	date
deceased_date	date
state	string

Region	
code	int
province	string
city	string
latitude	float
longitude	float
elementary_school_count	int
kindergarten_count	int
university_count	int
academy_ratio	float
elderly_population_ratio	float
elderly_alone_ratio	float
nursing_home_count	int

Weather	
code	int
province	string
date	date
avg_temp	float
min_temp	float
max_temp	float

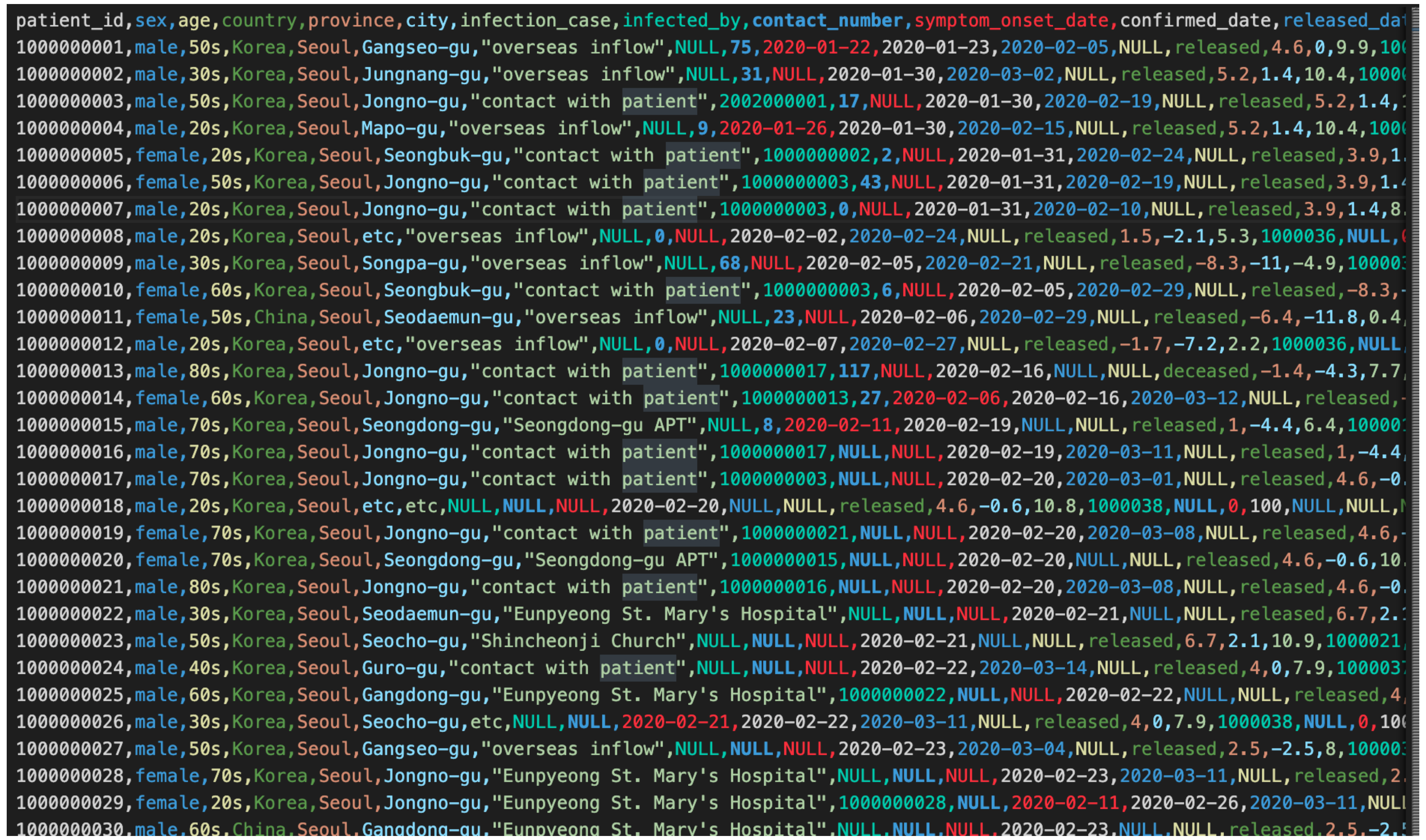
Time	
date	date
[redacted]	[redacted]
test	int
negative	int
confirmed	int
released	int
deceased	int

TimeAge	
date	date
[redacted]	[redacted]
age	string
confirmed	int
deceased	int

TimeGender	
date	date
[redacted]	[redacted]
sex	string
confirmed	int
deceased	int

TimeProvince	
date	date
[redacted]	[redacted]
province	string
confirmed	int
released	int
deceased	int

제공 되는 데이터 : additional_Timeinfo.csv, K_COVID19.csv



```
patient_id,sex,age,country,province,city,infection_case,infected_by,contact_number,symptom_onset_date,confirmed_date,released_da
1000000001,male,50s,Korea,Seoul,Gangseo-gu,"overseas inflow",NULL,75,2020-01-22,2020-01-23,2020-02-05,NULL,released,4.6,0,9.9,10
1000000002,male,30s,Korea,Seoul,Jungnang-gu,"overseas inflow",NULL,31,NULL,2020-01-30,2020-03-02,NULL,released,5.2,1.4,10.4,1000
1000000003,male,50s,Korea,Seoul,Jongno-gu,"contact with patient",2002000001,17,NULL,2020-01-30,2020-02-19,NULL,released,5.2,1.4,
1000000004,male,20s,Korea,Seoul,Mapo-gu,"overseas inflow",NULL,9,2020-01-26,2020-01-30,2020-02-15,NULL,released,5.2,1.4,10.4,100
1000000005,female,20s,Korea,Seoul,Seongbuk-gu,"contact with patient",1000000002,2,NULL,2020-01-31,2020-02-24,NULL,released,3.9,1.
1000000006,female,50s,Korea,Seoul,Jongno-gu,"contact with patient",1000000003,43,NULL,2020-01-31,2020-02-19,NULL,released,3.9,1.
1000000007,male,20s,Korea,Seoul,Jongno-gu,"contact with patient",1000000003,0,NULL,2020-01-31,2020-02-10,NULL,released,3.9,1.4,8.
1000000008,male,20s,Korea,Seoul,etc,"overseas inflow",NULL,0,NULL,2020-02-02,2020-02-24,NULL,released,1.5,-2.1,5.3,1000036,NULL,
1000000009,male,30s,Korea,Seoul,Songpa-gu,"overseas inflow",NULL,68,NULL,2020-02-05,2020-02-21,NULL,released,-8.3,-11,-4.9,10000
1000000010,female,60s,Korea,Seoul,Seongbuk-gu,"contact with patient",1000000003,6,NULL,2020-02-05,2020-02-29,NULL,released,-8.3,-
1000000011,female,50s,China,Seoul,Seodaemun-gu,"overseas inflow",NULL,23,NULL,2020-02-06,2020-02-29,NULL,released,-6.4,-11.8,0.4,
1000000012,male,20s,Korea,Seoul,etc,"overseas inflow",NULL,0,NULL,2020-02-07,2020-02-27,NULL,released,-1.7,-7.2,2.2,1000036,NULL
1000000013,male,80s,Korea,Seoul,Jongno-gu,"contact with patient",1000000017,117,NULL,2020-02-16,NULL,NULL,deceased,-1.4,-4.3,7.7,
1000000014,female,60s,Korea,Seoul,Jongno-gu,"contact with patient",1000000013,27,2020-02-06,2020-02-16,2020-03-12,NULL,released,-
1000000015,male,70s,Korea,Seoul,Seongdong-gu,"Seongdong-gu APT",NULL,8,2020-02-11,2020-02-19,NULL,NULL,released,1,-4.4,6.4,10000
1000000016,male,70s,Korea,Seoul,Jongno-gu,"contact with patient",1000000017,NULL,NULL,2020-02-19,2020-03-11,NULL,released,1,-4.4,
1000000017,male,70s,Korea,Seoul,Jongno-gu,"contact with patient",1000000003,NULL,NULL,2020-02-20,2020-03-01,NULL,released,4.6,-0.
1000000018,male,20s,Korea,Seoul,etc,etc,NULL,NULL,NULL,2020-02-20,NULL,NULL,released,4.6,-0.6,10.8,1000038,NULL,0,100,NULL,NULL,
1000000019,female,70s,Korea,Seoul,Jongno-gu,"contact with patient",1000000021,NULL,NULL,2020-02-20,2020-03-08,NULL,released,4.6,-
1000000020,female,70s,Korea,Seoul,Seongdong-gu,"Seongdong-gu APT",1000000015,NULL,NULL,2020-02-20,NULL,NULL,released,4.6,-0.6,10.
1000000021,male,80s,Korea,Seoul,Jongno-gu,"contact with patient",1000000016,NULL,NULL,2020-02-20,2020-03-08,NULL,released,4.6,-0.
1000000022,male,30s,Korea,Seoul,Seodaemun-gu,"Eunpyeong St. Mary's Hospital",NULL,NULL,NULL,2020-02-21,NULL,NULL,released,6.7,2.
1000000023,male,50s,Korea,Seoul,Seocho-gu,"Shincheonji Church",NULL,NULL,NULL,2020-02-21,NULL,NULL,released,6.7,2.1,10.9,1000021,
1000000024,male,40s,Korea,Seoul,Guro-gu,"contact with patient",NULL,NULL,NULL,2020-02-22,2020-03-14,NULL,released,4,0,7.9,100003
1000000025,male,60s,Korea,Seoul,Gangdong-gu,"Eunpyeong St. Mary's Hospital",1000000022,NULL,NULL,2020-02-22,NULL,NULL,released,4,
1000000026,male,30s,Korea,Seoul,Seocho-gu,etc,NULL,NULL,2020-02-21,2020-02-22,2020-03-11,NULL,released,4,0,7.9,1000038,NULL,0,10
1000000027,male,50s,Korea,Seoul,Gangseo-gu,"overseas inflow",NULL,NULL,NULL,2020-02-23,2020-03-04,NULL,released,2.5,-2.5,8,10000
1000000028,female,70s,Korea,Seoul,Jongno-gu,"Eunpyeong St. Mary's Hospital",NULL,NULL,NULL,2020-02-23,2020-03-11,NULL,released,2.
1000000029,female,20s,Korea,Seoul,Jongno-gu,"Eunpyeong St. Mary's Hospital",1000000028,NULL,2020-02-11,2020-02-26,2020-03-11,NULL
1000000030,male,60s,China,Seoul,Gangdong-gu,"Eunpyeong St. Mary's Hospital",NULL,NULL,NULL,2020-02-23,NULL,NULL,released,2.5,-2.
```

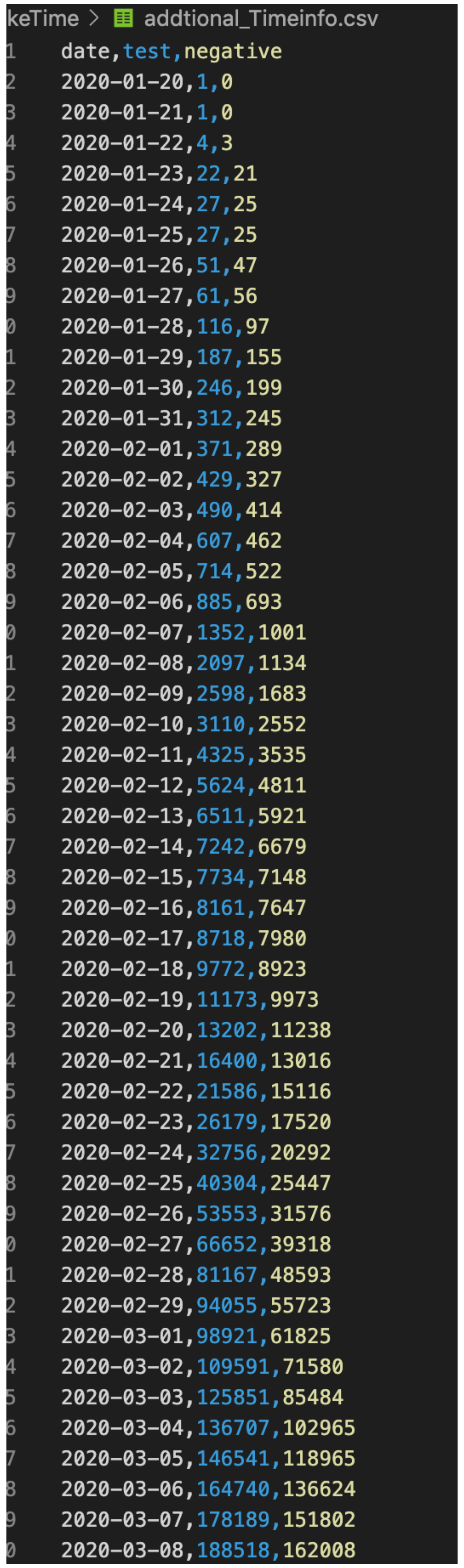
- 1주차 과제의 K_COVID19의 PatientInfo를 사용합니다.(parsing_patient.py를 참조합니다.)

- Additional_Time.csv는 총 3개의 column으로 구성되어 있습니다. 1행에 각각의 column 정보를 확인하실 수 있습니다.

- 각 row는 해당날짜의 누적 검사수, 누적 음성판정수로 이루어져 있습니다. (ex : 2020년 1월 20일에는 검사자 1명, 음성판정(확진)0명이다.)

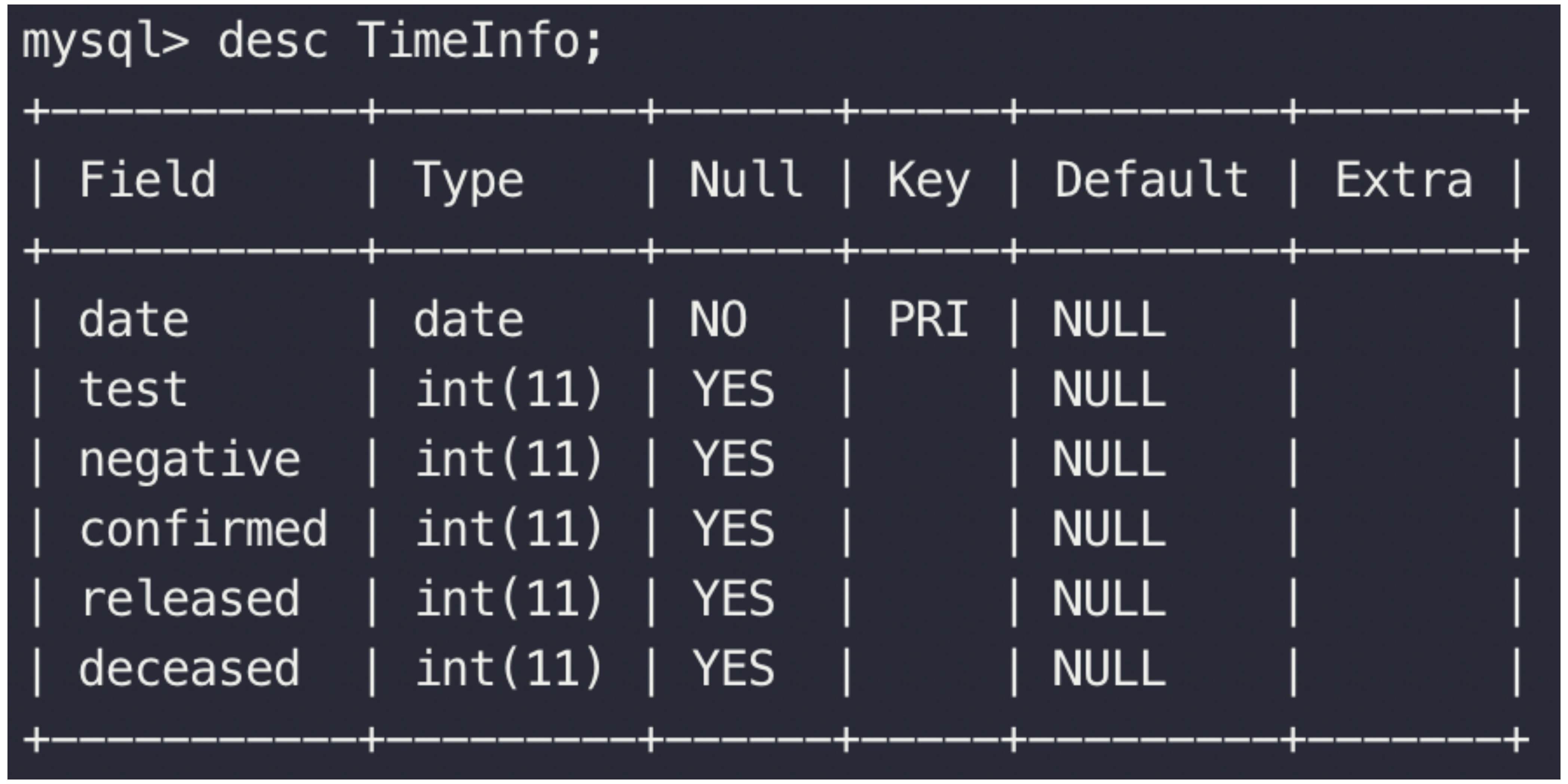
- 과제요약 : 주어진 additional_Timeintfo.csv와 K_COVID19.csv를 사용하여 앞장의 Relational model에 추가된 "Time, TimeAge, TimeGender, TimeProvince"를 만드는 것이 목적입니다.

  ➔ "Time"은 테이블 생성 및 파싱을 예시로 미리 제공해드립니다. 여러분들은 TimeGender, TimeProvince, TimeAge만 각각 팀원당 하나씩 구현해주시면 됩니다.



```
date,test,negative
2020-01-20,1,0
2020-01-21,1,0
2020-01-22,4,3
2020-01-23,22,21
2020-01-24,27,25
2020-01-25,27,25
2020-01-26,51,47
2020-01-27,61,56
2020-01-28,116,97
2020-01-29,187,155
2020-01-30,246,199
2020-01-31,312,245
2020-02-01,371,289
2020-02-02,429,327
2020-02-03,490,414
2020-02-04,607,462
2020-02-05,714,522
2020-02-06,885,693
2020-02-07,1352,1001
2020-02-08,2097,1134
2020-02-09,2598,1683
2020-02-10,3110,2552
2020-02-11,4325,3535
2020-02-12,5624,4811
2020-02-13,6511,5921
2020-02-14,7242,6679
2020-02-15,7734,7148
2020-02-16,8161,7647
2020-02-17,8718,7980
2020-02-18,9772,8923
2020-02-19,11173,9973
2020-02-20,13202,11238
2020-02-21,16400,13016
2020-02-22,21586,15116
2020-02-23,26179,17520
2020-02-24,32756,20292
2020-02-25,40304,25447
2020-02-26,53553,31576
2020-02-27,66652,39318
2020-02-28,81167,48593
2020-02-29,94055,55723
2020-03-01,98921,61825
2020-03-02,109591,71580
2020-03-03,125851,85484
2020-03-04,136707,102965
2020-03-05,146541,118965
2020-03-06,164740,136624
2020-03-07,178189,151802
2020-03-08,188518,162008
```

TimeInfo 테이블 : COVID-19 data by date



```
mysql> desc TimeInfo;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| date      | date    | NO   | PRI | NULL    |       |
| test      | int(11) | YES  |     | NULL    |       |
| negative  | int(11) | YES  |     | NULL    |       |
| confirmed | int(11) | YES  |     | NULL    |       |
| released  | int(11) | YES  |     | NULL    |       |
| deceased  | int(11) | YES  |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
```

- Date : 코로나 터진 이후 2020-06-30일까지의 날짜(primary key)
- Test : 그 날의 누적검사자 수
- Negative : 그 날의 누적 음성판정자 수
- Confirmed : 그날 누적 양성판정자 수
- Released : 확진받고 격리된 사람들 중 격리 해제된 사람들 수(누적)
- deceased : 누적사망자 수

주의1 : 모든 column은 “누적 " 됨.

Ex) 1월 20일에 확진자 1명 , 1월 21일에 확진자 1명 : 총 2명

=> 1월 21일의 confirmed은 “2” 여야함.

주의2 : Time의 date와 Patient의 Confirmed_date와 연결하셔야 합니다.

과제 : TimeAge, TimeGender, TimeProvince 테이블 생성 및 data parsing

➔ 본 과제의 목적은 주어진 data의 parsing을 한 뒤, 우리가 좀 더 쓰기 편하게 새로운 테이블을 만들어 보는 것이 목적 입니다. 따라서 TimeAge, TimeProvince, TimeGender에 대한 Table의 예시는 제시 하지 않겠습니다.

Hint1 : 각각 의 table column의 type(string,int …)은 patientInfo(1주차실습자료) 와 Time의 column type정보를 사용하여 만들어 주십시오.

Hint2 : Time에서는 date만 primary key였습니다. 하지만 TimeAge는 (날짜,나이대)를 기준으로 복합키를 설정하셔야 할 것입니다.

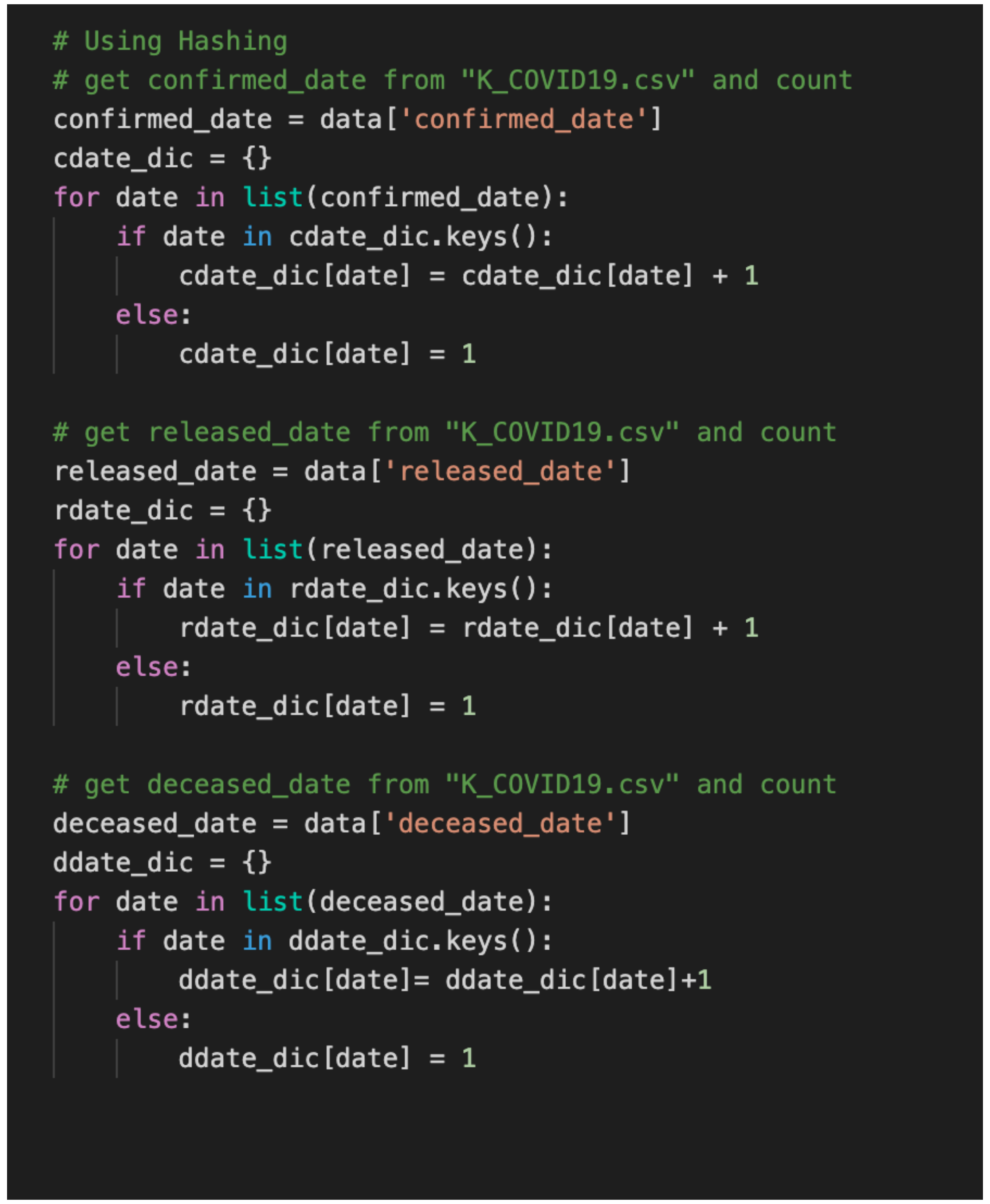
ex) TimeAge에 대해서, (1월 20일 10대 누적확진자 x명, 누적사망자 y명) , (1월 20일 20대 누적확진자 x2명, 누적 사망자 y2명) 처럼 한 날짜에 대한 10대, 20대 ,30대.. 모든 row가 생길 것 입니다.

Hint3 : TimeProvince의 장소는 Patientinfo의 patient province를 기준으로 합니다 case province(X).

➔ 내용 : 팀원당 각각 makeTime.py를 참조하여서 TimeAge, TimeProvince, TimeGender에 대한 sql쿼리문을 작성하고, 누적 확진자 수, 누적 사망자 수등을 계산하여서 parsing한 뒤 제출하십시오.

➔ 제출물 : 각각의 sql문3개 , parsing python파일 3개씩 총 6개 파일 및 각 table을 count(*)한 결과값 3개로 총 9개의 파일을 제출해주시면 됩니다.

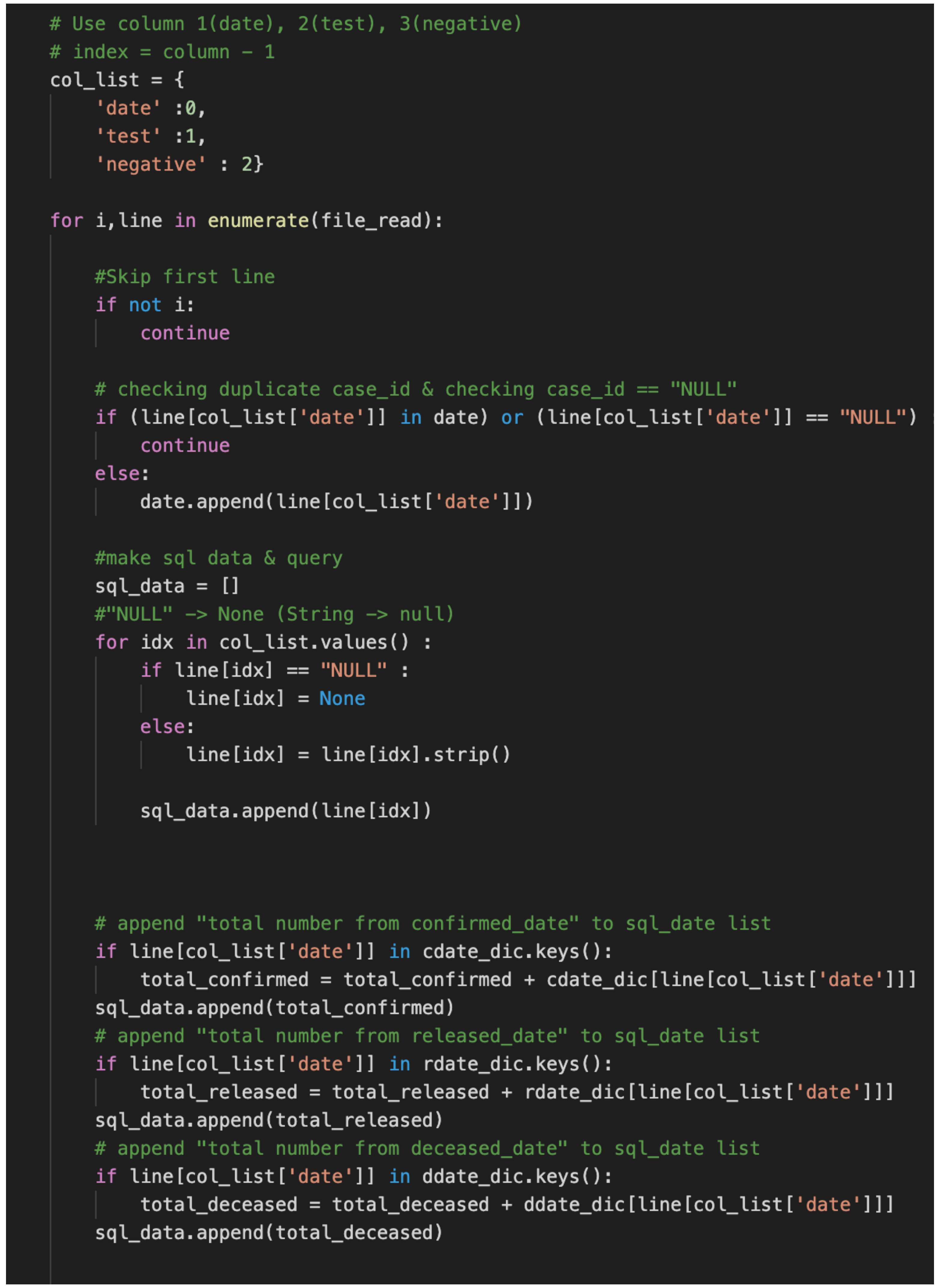
CSV파일 → 데이터 베이스 예시 : makeTime.py (lms 업로드)



```python
# Using Hashing
# get confirmed_date from "K_COVID19.csv" and count
confirmed_date = data['confirmed_date']
cdate_dic = {}
for date in list(confirmed_date):
    if date in cdate_dic.keys():
        cdate_dic[date] = cdate_dic[date] + 1
    else:
        cdate_dic[date] = 1

# get released_date from "K_COVID19.csv" and count
released_date = data['released_date']
rdate_dic = {}
for date in list(released_date):
    if date in rdate_dic.keys():
        rdate_dic[date] = rdate_dic[date] + 1
    else:
        rdate_dic[date] = 1

# get deceased_date from "K_COVID19.csv" and count
deceased_date = data['deceased_date']
ddate_dic = {}
for date in list(deceased_date):
    if date in ddate_dic.keys():
        ddate_dic[date]= ddate_dic[date]+1
    else:
        ddate_dic[date] = 1
```



```python
# Use column 1(date), 2(test), 3(negative)
# index = column - 1
col_list = {
    'date' :0,
    'test' :1,
    'negative' : 2}

for i,line in enumerate(file_read):

    #Skip first line
    if not i:
        continue

    # checking duplicate case_id & checking case_id == "NULL"
    if (line[col_list['date']] in date) or (line[col_list['date']] == "NULL")
        continue
    else:
        date.append(line[col_list['date']])

    #make sql data & query
    sql_data = []
    #"NULL" -> None (String -> null)
    for idx in col_list.values() :
        if line[idx] == "NULL" :
            line[idx] = None
        else:
            line[idx] = line[idx].strip()

        sql_data.append(line[idx])



    # append "total number from confirmed_date" to sql_date list
    if line[col_list['date']] in cdate_dic.keys():
        total_confirmed = total_confirmed + cdate_dic[line[col_list['date']]]
    sql_data.append(total_confirmed)
    # append "total number from released_date" to sql_date list
    if line[col_list['date']] in rdate_dic.keys():
        total_released = total_released + rdate_dic[line[col_list['date']]]
    sql_data.append(total_released)
    # append "total number from deceased_date" to sql_date list
    if line[col_list['date']] in ddate_dic.keys():
        total_deceased = total_deceased + ddate_dic[line[col_list['date']]]
    sql_data.append(total_deceased)
```

→ 왼쪽 코드 : dataframe으로 K_COVID19.csv의 확진 날짜별 누적적인 사람수를 세는 코드 입니다.

→ 오른쪽 코드 : additional_time.csv를 통하여서 TimeInfo table에 데이터를 insert하는 부분.

Do your BEST!