COMP-421 Database Systems, Winter 2020
Project 2: Creating Your Database

CHEN, TSUNG-YU/ KUO, KUAN-TING/ LI Zhengdao/ XU Tianxiao

Q1 Relational Schema

In the file “Q1 Relational Schema.pdf”

Q2 Create Table

The SQL command is stored in “createTable.sql”
The output of “\d TABLE_NAME” is stored in “descripTable.log”

Q3 Insert

The SQL command is stored in “insert.sql”
The output of “\d TABLE_NAME” is stored in “insert.log”

Q4 Insert All

The output (truncated table) is stored in “printTable.log”

Q5 Write 5 Queries

Query 1

Description:
The first query will use the relationship table Infect to calculate the average amount of time that a person takes to recover from a virus.

Used Table:

```
CREATE TABLE Infect
(
   vName   VARCHAR(255),
   Nationality     VARCHAR(255),
   National_ID     VARCHAR(255),
   State        VARCHAR(255) NOT NULL,
   Begin_Time   DATE NOT NULL,
   End_Time     DATE,
       CHECK(End_Time IS NULL OR Begin_Time <= End_Time),
   PRIMARY KEY (vName, Nationality, National_ID, Begin_Time),
   FOREIGN KEY (vName) REFERENCES Virus (vName),
   FOREIGN KEY (Nationality, National_ID) REFERENCES Patient (Nationality,
National_ID)
);
```

SQL:

```
SELECT I.vName AS "Virus", AVG(I.End_Time - I.Begin_Time) AS "Avg_Recover_Time
(Days)"
FROM Infect I
GROUP BY I.vName
LIMIT 10;
```

The output of the query is stored in “query.log”

Query 2

Description:

In the second query, we want to find out the research institutes that have already produce vaccinations for MERS-CoV. We select research institutes name, country of the institute and the address of it.

Used Table:

```
CREATE TABLE Vaccination
(
   rName       VARCHAR(255),
   iName           VARCHAR(255),
   Code            VARCHAR(255),
   Currency        CHAR(3),
   Price           INT,
   PRIMARY KEY (rName, iName, Code),
   FOREIGN KEY (rName, iName) REFERENCES Institute(rName, iName)
);
CREATE TABLE Research_Institute
(
   rName VARCHAR(255),
   iName VARCHAR(255),
   Address VARCHAR(255) NOT NULL,
   PRIMARY KEY (rName, iName)
);
```

SQL:

```
SELECT V.iName AS "Research Institute", V.rName AS "Country", R.address AS
"Address"
FROM Vaccination AS V JOIN Research_Institute AS R ON V.rName = R.rName AND V.iName
= R.iName
WHERE V.vName = 'MERS-CoV'
LIMIT 10;
```

The output of the query is stored in “query.log”

Query 3

Description:

This query aims to calculate the possibility of a virus to really infect a person with an observed symptom. It may be useful to serve as a reference for hospitals to distinguish and inference whether a person is infected by a virus or not.

Used Table:

```
CREATE TABLE Infect
(
   vName   VARCHAR(255),
   Nationality     VARCHAR(255),
   National_ID     VARCHAR(255),
   State       VARCHAR(255) NOT NULL,
   Begin_Time  DATE NOT NULL,
   End_Time    DATE,
       CHECK(End_Time IS NULL OR Begin_Time <= End_Time),
   PRIMARY KEY (vName, Nationality, National_ID, Begin_Time),
   FOREIGN KEY (vName) REFERENCES Virus (vName),
   FOREIGN KEY (Nationality, National_ID) REFERENCES Patient (Nationality,
National_ID)
);
CREATE TABLE Exhibit
(
   sName       VARCHAR(255),
   Nationality     VARCHAR(255),
   National_ID     VARCHAR(255),
   PRIMARY KEY (sName, Nationality, National_ID),
   FOREIGN KEY (Nationality, National_ID) REFERENCES Patient(Nationality,
National_ID),
   FOREIGN KEY (sName) REFERENCES Symptom(sName)
);
```

SQL:

```
SELECT sub1.vName AS "Virus", sub1.sName AS "Symptom", (sub1.Count*1. /
sub2.Count*1.) AS "P (Virus | Symptom)"
FROM
(
       SELECT I.vName, E.sName, Count(*) AS Count
       FROM Infect I JOIN Exhibit E ON I.Nationality = E.Nationality AND
I.National_ID = E.National_ID
       GROUP BY I.vName, E.sName
) sub1 JOIN -- P (A and B)
(
       SELECT E.sName, Count(*) AS Count
       FROM Infect I JOIN Exhibit E ON I.Nationality = E.Nationality AND
I.National_ID = E.National_ID
       GROUP BY E.sName
) sub2 ON sub1.sName = sub2.sName -- P(B)
LIMIT 10;
```

The output of the query is stored in “query.log”

Query 4

Description:

In this query, we use relationship tables Virus and Infect to calculate the death rate for a certain virus within a certain country.

Used Table:

```
CREATE TABLE Virus
(
   vName VARCHAR(255) PRIMARY KEY NOT NULL,
   Type VARCHAR(255),
   Danger_Degree INTEGER,
   Annotation VARCHAR(1000),
   CHECK (danger_degree > 0 AND danger_degree < 6)
);
CREATE TABLE Infect
(
   vName   VARCHAR(255),
   Nationality     VARCHAR(255),
   National_ID     VARCHAR(255),
   State       VARCHAR(255) NOT NULL,
   Begin_Time  DATE NOT NULL,
   End_Time    DATE,
       CHECK(End_Time IS NULL OR Begin_Time <= End_Time),
   PRIMARY KEY (vName, Nationality, National_ID, Begin_Time),
   FOREIGN KEY (vName) REFERENCES Virus (vName),
   FOREIGN KEY (Nationality, National_ID) REFERENCES Patient (Nationality,
National_ID)
);
```

<u>SQL:</u>

```
WITH Total (vName, Nationality, Count) AS
(
       SELECT V.vName, I.Nationality, COUNT(*) AS Count
       FROM Virus V JOIN Infect I ON V.vName = I.vName
       GROUP BY V.vName, I.Nationality
),
Dead (vName, Nationality, Count) AS
(
       SELECT V.vName, I.Nationality, COUNT(*) AS Count
       FROM Virus V JOIN Infect I ON V.vName = I.vName
       WHERE I.state = 'Dead'
       GROUP BY V.vName, I.Nationality
)
SELECT Total.vName AS "Virus", Total.Nationality AS "Nationality", Dead.Count*1. /
Total.Count*1. AS "Death Rate"
FROM Total JOIN Dead ON Dead.vName = Total.vName AND Dead.Nationality =
Total.Nationality
LIMIT 10;
```

The output of the query is stored in “query.log”

Query 5

<u>Description:</u>

The query is used to look for the first outbreak of a certain virus within a region. It will output the name of the virus, the name of the region, and the first outbreak that is found in the database.

<u>Used Table:</u>

```
CREATE TABLE Infect
(
   vName   VARCHAR(255),
   Nationality     VARCHAR(255),
   National_ID     VARCHAR(255),
   State       VARCHAR(255) NOT NULL,
   Begin_Time   DATE NOT NULL,
   End_Time     DATE,
       CHECK(End_Time IS NULL OR Begin_Time <= End_Time),
   PRIMARY KEY (vName, Nationality, National_ID, Begin_Time),
   FOREIGN KEY (vName) REFERENCES Virus (vName),
   FOREIGN KEY (Nationality, National_ID) REFERENCES Patient (Nationality,
National_ID)
);
CREATE TABLE Accommodation
(
   Nationality      VARCHAR(255),
   National_ID      VARCHAR(255),
   iName        VARCHAR(255),
   rName        VARCHAR(255),
   Begin_Time   DATE NOT NULL,
   End_Time     DATE,
       CHECK(End_Time IS NULL OR Begin_Time <= End_Time),
   PRIMARY KEY (Nationality, National_ID, iName, rName),
   FOREIGN KEY (Nationality, National_ID) REFERENCES Patient(Nationality,
National_ID),
   FOREIGN KEY (iName, rName) REFERENCES Hospital(iName, rName)
);
```

SQL:

```
SELECT I.vName AS "Virus", A.rName AS "Country", MIN(I.Begin_Time) AS "First
Outbreak Date"
FROM Infect I JOIN Accommodation A ON (I.Nationality = A.Nationality AND
I.National_ID = A.National_ID)
GROUP BY I.vName, A.rName
LIMIT 10;
```

The output of the query is stored in “query.log”

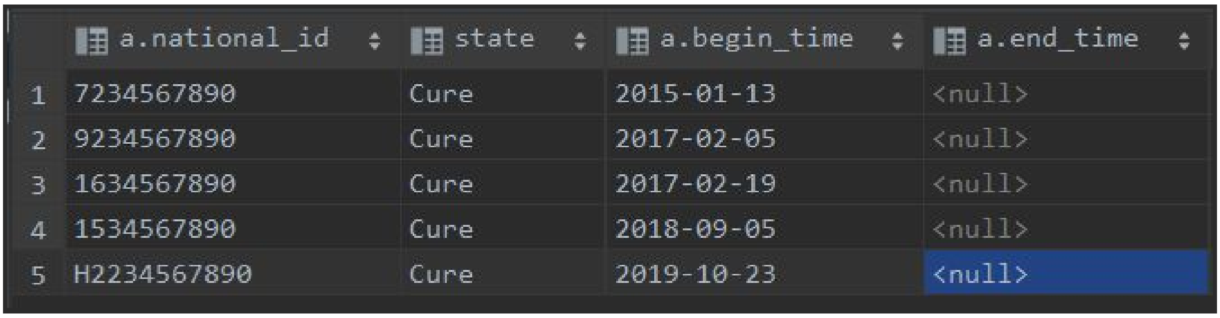
Q6 Data Modifications

1. Discharge patient that is cured on the same day
   Example: discharge patient that was cured on 2020-01-01

```
UPDATE accommodation A
SET end_time = '2020-01-01'
```

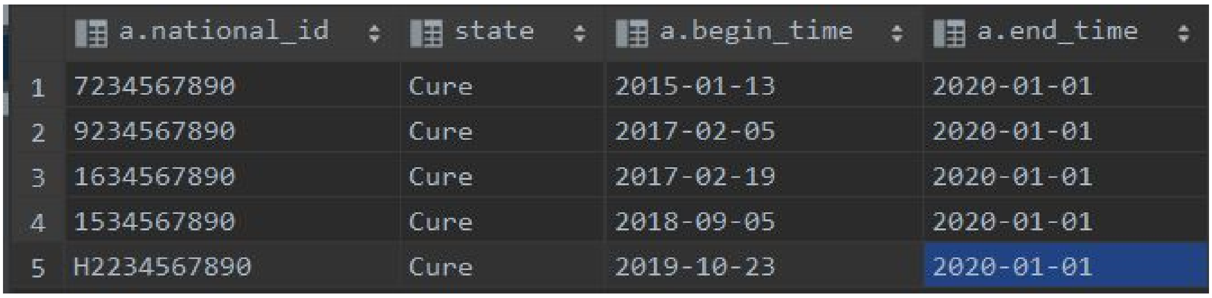
```
FROM infect I
Where I.national_id = A.national_id AND I.nationality = A.nationality AND I.state =
'Cure';
```

Before modification:



	a.national_id	state	a.begin_time	a.end_time
1	7234567890	Cure	2015-01-13	<null>
2	9234567890	Cure	2017-02-05	<null>
3	1634567890	Cure	2017-02-19	<null>
4	1534567890	Cure	2018-09-05	<null>
5	H2234567890	Cure	2019-10-23	<null>

After modification:

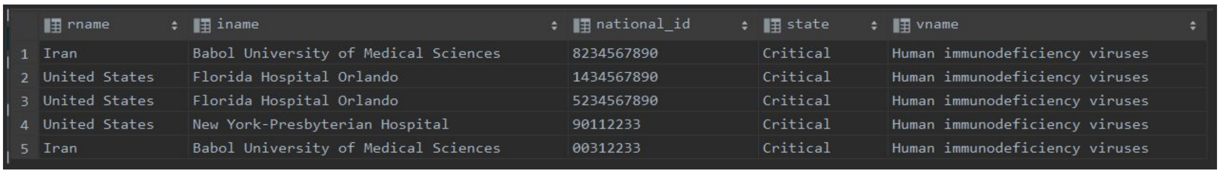


	a.national_id	state	a.begin_time	a.end_time
1	7234567890	Cure	2015-01-13	2020-01-01
2	9234567890	Cure	2017-02-05	2020-01-01
3	1634567890	Cure	2017-02-19	2020-01-01
4	1534567890	Cure	2018-09-05	2020-01-01
5	H2234567890	Cure	2019-10-23	2020-01-01

2. Transfer Patients base on medical state and virus
Example: Patients with Human immunodeficiency viruses that are in the critical state will be transferred to Ruijin Hospital in China.

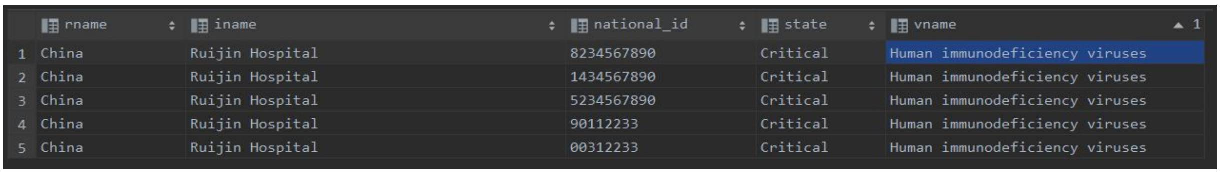
```
UPDATE accommodation A
SET iName = 'Ruijin Hospital', rName = 'China', begin_time = '2020-02-27'
FROM infect I
Where I.national_id = A.national_id AND I.nationality = A.nationality AND I.state =
'Serious'AND I.vname ='MERS-CoV';
```

Before modification: The 2 ~ 4 Records meets the condition for transfer.



	rname	iname	national_id	state	vname
1	Iran	Babol University of Medical Sciences	8234567890	Critical	Human immunodeficiency viruses
2	United States	Florida Hospital Orlando	1434567890	Critical	Human immunodeficiency viruses
3	United States	Florida Hospital Orlando	5234567890	Critical	Human immunodeficiency viruses
4	United States	New York-Presbyterian Hospital	90112233	Critical	Human immunodeficiency viruses
5	Iran	Babol University of Medical Sciences	00312233	Critical	Human immunodeficiency viruses

After modification: the results are shown as the 1~3 records.

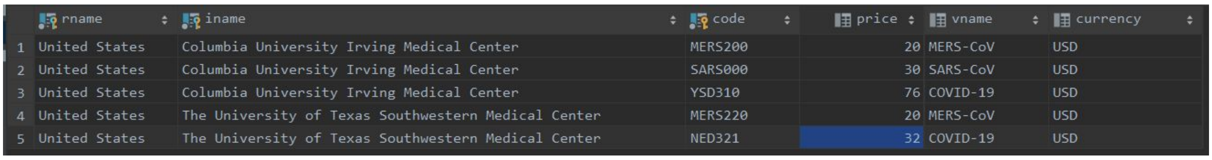


	rname	iname	national_id	state	vname
1	China	Ruijin Hospital	8234567890	Critical	Human immunodeficiency viruses
2	China	Ruijin Hospital	1434567890	Critical	Human immunodeficiency viruses
3	China	Ruijin Hospital	5234567890	Critical	Human immunodeficiency viruses
4	China	Ruijin Hospital	90112233	Critical	Human immunodeficiency viruses
5	China	Ruijin Hospital	00312233	Critical	Human immunodeficiency viruses

3. Decrease the price of vaccination in a certain country
   Example: vaccination in the United States dropped to 90%

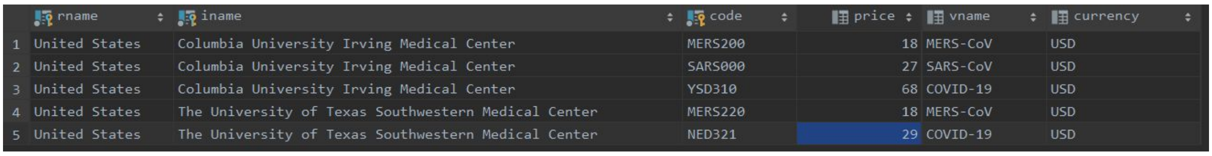
```
UPDATE vaccination
set price = price * 0.9
where rname = 'United States';
```

Before modification:



	rname	iname	code	price	vname	currency
1	United States	Columbia University Irving Medical Center	MERS200	20	MERS-CoV	USD
2	United States	Columbia University Irving Medical Center	SARS000	30	SARS-CoV	USD
3	United States	Columbia University Irving Medical Center	YSD310	76	COVID-19	USD
4	United States	The University of Texas Southwestern Medical Center	MERS220	20	MERS-CoV	USD
5	United States	The University of Texas Southwestern Medical Center	NED321	32	COVID-19	USD

After modification:



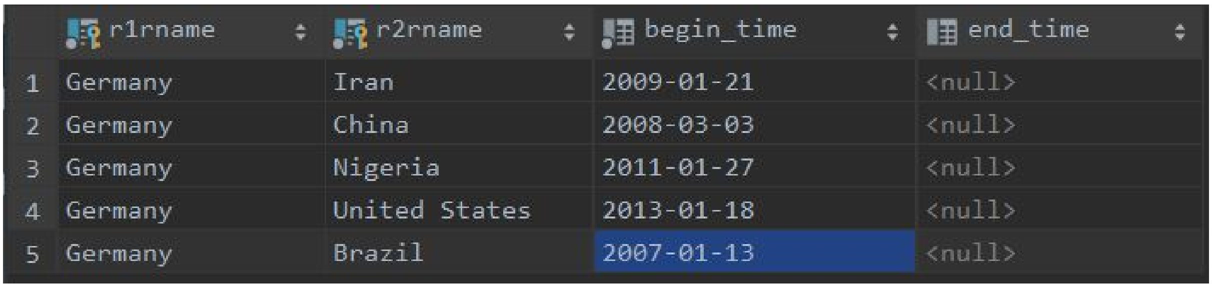
	rname	iname	code	price	vname	currency
1	United States	Columbia University Irving Medical Center	MERS200	18	MERS-CoV	USD
2	United States	Columbia University Irving Medical Center	SARS000	27	SARS-CoV	USD
3	United States	Columbia University Irving Medical Center	YSD310	68	COVID-19	USD
4	United States	The University of Texas Southwestern Medical Center	MERS220	18	MERS-CoV	USD
5	United States	The University of Texas Southwestern Medical Center	NED321	29	COVID-19	USD

4. Open Border between one country and other nations that have been unreachable.

   Example: Germany opened all borders that were closed on 2020-01-09

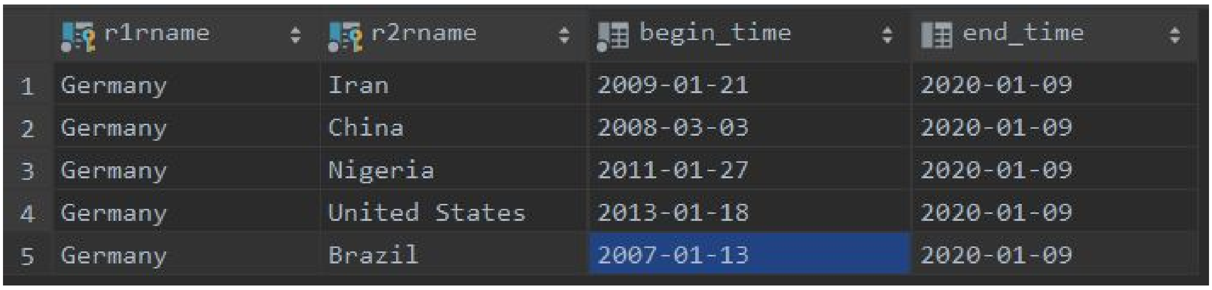
```
UPDATE closed_border
SET End_Time = ‘2020-01-09’
where r1rname = 'Germany';
```

Before modification:



	r1rname	r2rname	begin_time	end_time
1	Germany	Iran	2009-01-21	<null>
2	Germany	China	2008-03-03	<null>
3	Germany	Nigeria	2011-01-27	<null>
4	Germany	United States	2013-01-18	<null>
5	Germany	Brazil	2007-01-13	<null>

After modification:



	r1rname	r2rname	begin_time	end_time
1	Germany	Iran	2009-01-21	2020-01-09
2	Germany	China	2008-03-03	2020-01-09
3	Germany	Nigeria	2011-01-27	2020-01-09
4	Germany	United States	2013-01-18	2020-01-09
5	Germany	Brazil	2007-01-13	2020-01-09

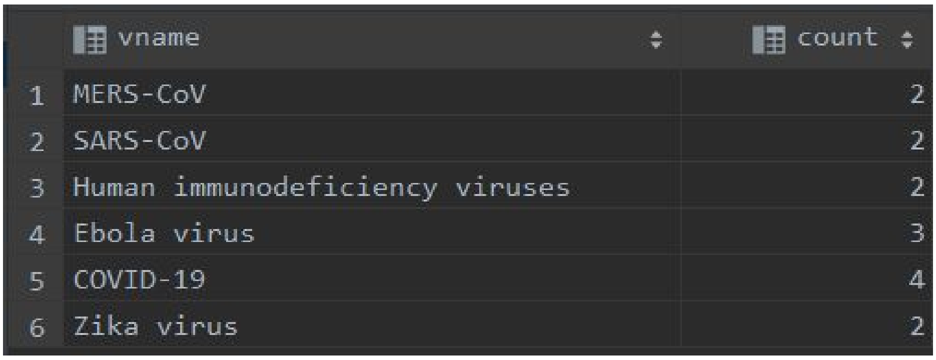
Q7 VIEW

VIEW 1:

CREATE VIEW NumOfDeath AS
SELECT Virus.vName, COUNT(*)
FROM Virus, Infect
WHERE Infect.vName = Virus.vName AND Infect.state = 'Dead'
GROUP BY Virus.vName

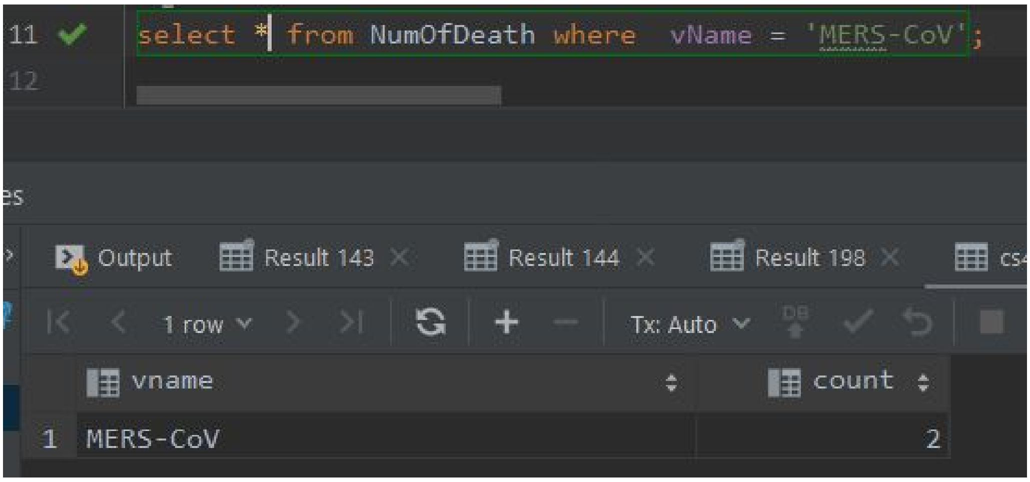
Description: return the fatality number caused by each virus.

System response by selecting all column in the view:



	vname	count
1	MERS-CoV	2
2	SARS-CoV	2
3	Human immunodeficiency viruses	2
4	Ebola virus	3
5	COVID-19	4
6	Zika virus	2

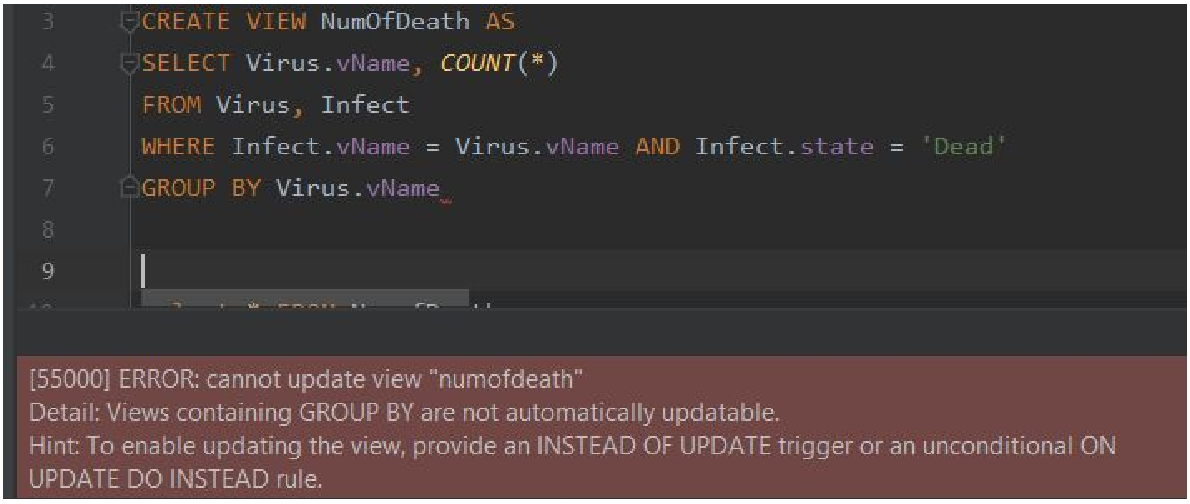
Query result concerning the virus name of MERS-CoV



```
select * from NumOfDeath where  vName = 'MERS-CoV';
```

	vname	count
1	MERS-CoV	2

Update vname attempt unsuccessful due to grouping



```
CREATE VIEW NumOfDeath AS
SELECT Virus.vName, COUNT(*)
FROM Virus, Infect
WHERE Infect.vName = Virus.vName AND Infect.state = 'Dead'
GROUP BY Virus.vName
```

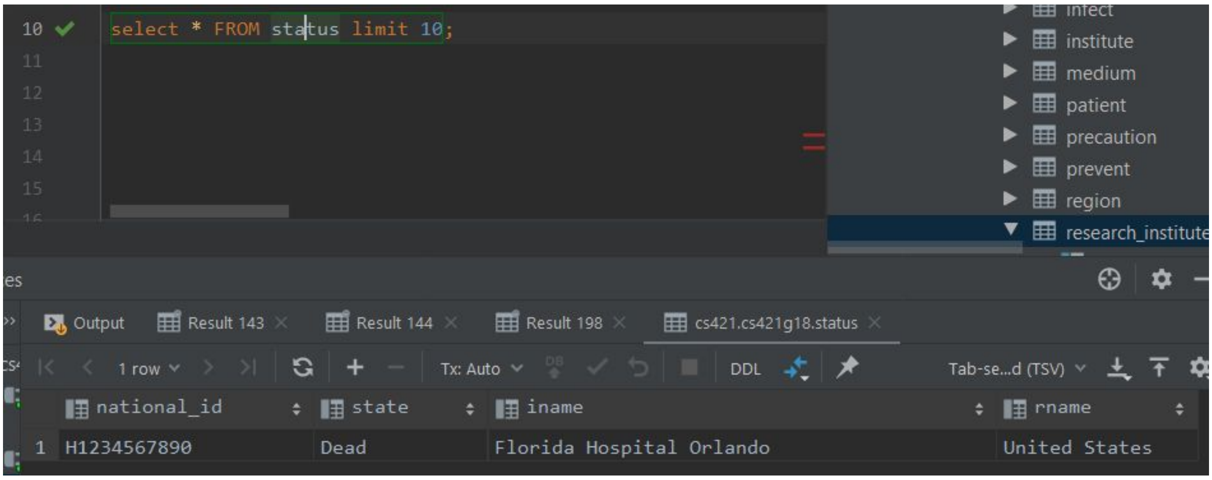
```
[55000] ERROR: cannot update view "numofdeath"
Detail: Views containing GROUP BY are not automatically updatable.
Hint: To enable updating the view, provide an INSTEAD OF UPDATE trigger or an unconditional ON UPDATE DO INSTEAD rule.
```

VIEW 2:

```
CREATE VIEW status( national_ID,state, begin_time, rName)
  AS SELECT I.national_id, I.state, A.begin_time, A.rName
  FROM infect I,  accommodation A
  where I.national_id = A.national_id
AND I.state = ‘Dead’ AND A.rName = 'United States';
```

Description: Find the record of patients that have died, and are in united states

Query: Print out all patients died in U.S. hospital



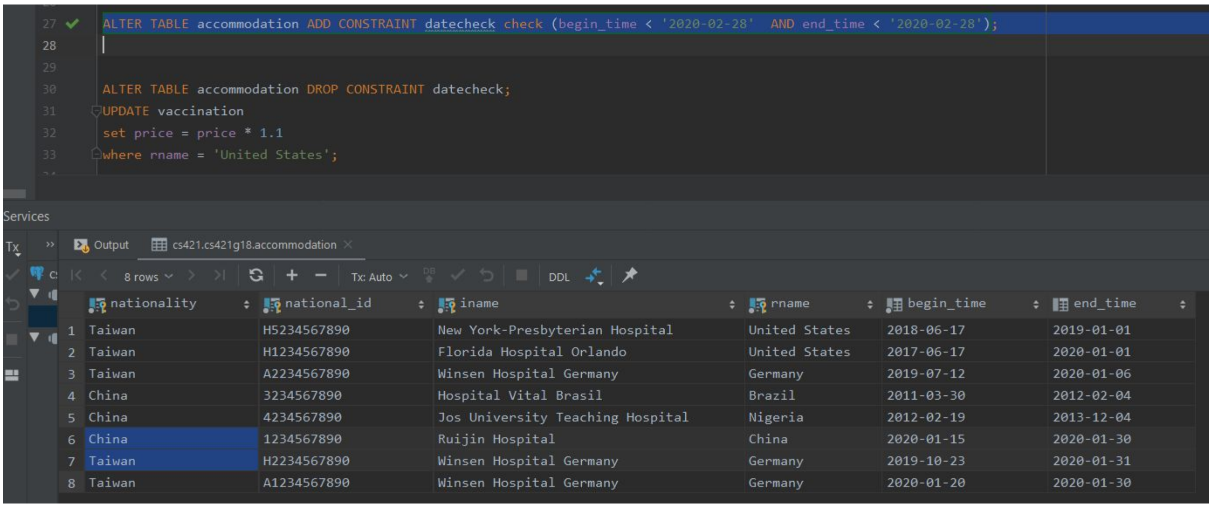
```
select * FROM status limit 10;
```

	national_id	state	iname	rname
1	H1234567890	Dead	Florida Hospital Orlando	United States

Update failure: can not update since the view was generated from 2 table.

Q8 Check

1. Check If the input data is valid (before today) for accommodation relationship
   Adding Constraint:

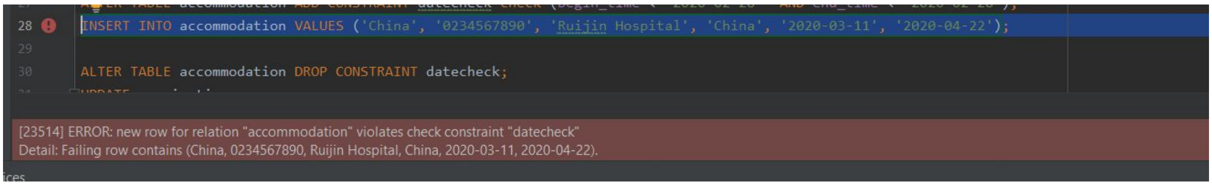


```
ALTER TABLE accommodation ADD CONSTRAINT datecheck check (begin_time < '2020-02-28'  AND end_time < '2020-02-28');


ALTER TABLE accommodation DROP CONSTRAINT datecheck;
UPDATE vaccination
set price = price * 1.1
where rname = 'United States';
```

	nationality	national_id	iname	rname	begin_time	end_time
1	Taiwan	H5234567890	New York-Presbyterian Hospital	United States	2018-06-17	2019-01-01
2	Taiwan	H1234567890	Florida Hospital Orlando	United States	2017-06-17	2020-01-01
3	Taiwan	A2234567890	Winsen Hospital Germany	Germany	2019-07-12	2020-01-06
4	China	3234567890	Hospital Vital Brasil	Brazil	2011-03-30	2012-02-04
5	China	4234567890	Jos University Teaching Hospital	Nigeria	2012-02-19	2013-12-04
6	China	1234567890	Ruijin Hospital	China	2020-01-15	2020-01-30
7	Taiwan	H2234567890	Winsen Hospital Germany	Germany	2019-10-23	2020-01-31
8	Taiwan	A1234567890	Winsen Hospital Germany	Germany	2020-01-20	2020-01-30

   Example of insertion that violates the constraint:

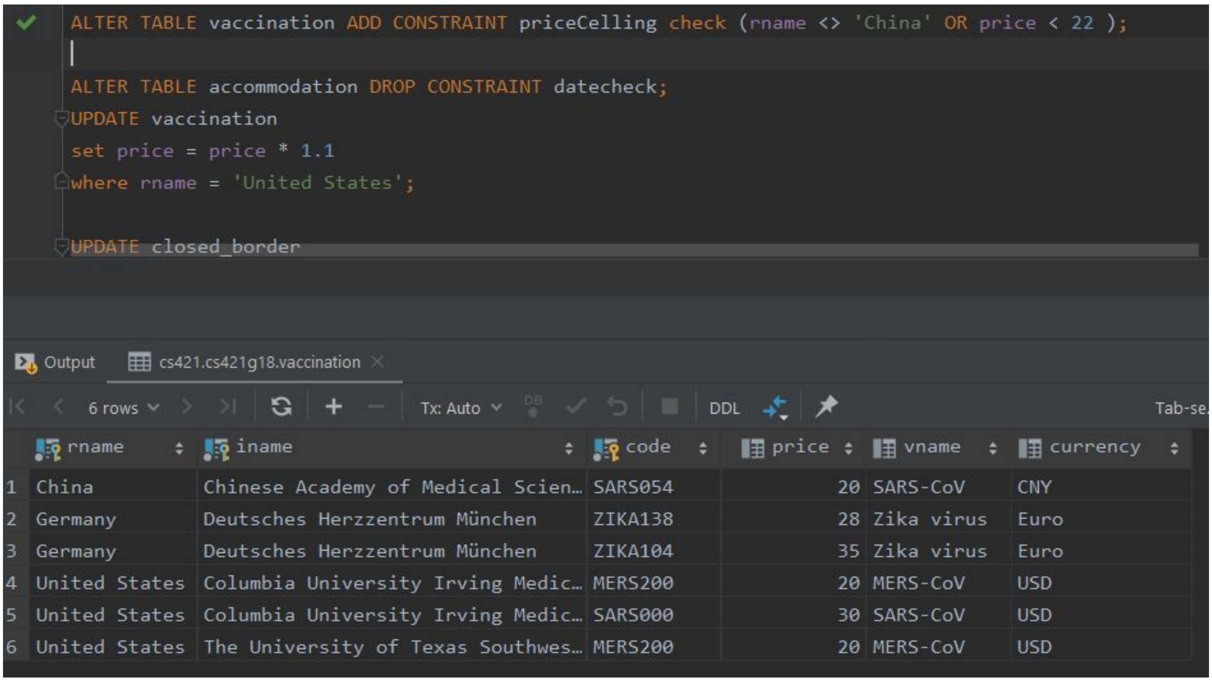


```
INSERT INTO accommodation VALUES ('China', '0234567890', 'Ruijin Hospital', 'China', '2020-03-11', '2020-04-22');

ALTER TABLE accommodation DROP CONSTRAINT datecheck;
```

```
[23514] ERROR: new row for relation "accommodation" violates check constraint "datecheck"
Detail: Failing row contains (China, 0234567890, Ruijin Hospital, China, 2020-03-11, 2020-04-22).
```

2. Check if the price of vaccination in a certain country exceeds the price ceiling set by governments.
   Adding Constraint:



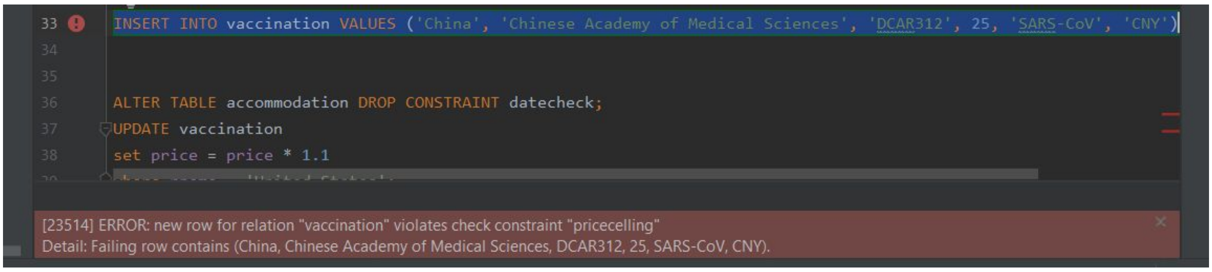
```sql
ALTER TABLE vaccination ADD CONSTRAINT priceCelling check (rname <> 'China' OR price < 22 );

ALTER TABLE accommodation DROP CONSTRAINT datecheck;
UPDATE vaccination
set price = price * 1.1
where rname = 'United States';

UPDATE closed_border
```

	rname	iname	code	price	vname	currency
1	China	Chinese Academy of Medical Scien…	SARS054	20	SARS-CoV	CNY
2	Germany	Deutsches Herzzentrum München	ZIKA138	28	Zika virus	Euro
3	Germany	Deutsches Herzzentrum München	ZIKA104	35	Zika virus	Euro
4	United States	Columbia University Irving Medic…	MERS200	20	MERS-CoV	USD
5	United States	Columbia University Irving Medic…	SARS000	30	SARS-CoV	USD
6	United States	The University of Texas Southwes…	MERS200	20	MERS-CoV	USD

Example of insertion that violates the constraint:



```sql
INSERT INTO vaccination VALUES ('China', 'Chinese Academy of Medical Sciences', 'DCAR312', 25, 'SARS-CoV', 'CNY')


ALTER TABLE accommodation DROP CONSTRAINT datecheck;
UPDATE vaccination
set price = price * 1.1
```

```
[23514] ERROR: new row for relation "vaccination" violates check constraint "pricecelling"
Detail: Failing row contains (China, Chinese Academy of Medical Sciences, DCAR312, 25, SARS-CoV, CNY).
```