

DST2 COVID-19 mini-project

Roll number: 9057

Please note that the structure of the documentation is not 100% based on the guidance. Part 3 and 4 are integrated into part 1b. Part 3 is replaced by “details on the python script”, which briefly explains how the python code works. Part 4 is the instructions for building the database.

1. The design of the software

a. User interface description

After running the code, please make the user interface into full screen. The three buttons at the left-hand side are for query 1, while the two buttons at the right are for query 2. Once you click one of the buttons of query 1 or 2, you enter this query. The buttons are quite similar to the demo's. After you type in all the input, you need to press enter to get the result. If you want to change a query, you can simply click one of the buttons of that query and the results of the previous query will be cleared. If your input is wrong, the warning messages will be displayed on the screen.

b. The provided queries

Query 1: What is the number of confirmed cases in a certain country or province in the whole year or a certain month?

Users can choose to query either country or province. If you want to query country, you do not need to type in anything in the province field. However, if you want to query province, you should input both country and province, and they should be matched. Users can also choose to query either a specific month or the whole year of 2020. The table below shows how the outputs related to different inputs.

	Country only	Country + province
Year (2020)	A table which contains the number of confirmed cases in the last day of every month of the input country (e.g.: 1.31, 2.29...). A plot which shows the trend of the confirmed cases.	Similar to the Year + country query, but the output will be the confirmed cases of the input province.
Month (an integer between 1 and 12)	A table which contains the confirmed cases in every day of the input month and country. (e.g.: input month = 1, output: 1.22-1.31) A plot which shows the trend of the confirmed cases.	Similar to the Month + country query, but the output will be the confirmed cases of the input province.

Note that once the query occurs, the plot is saved in the same directory where the python script is. Tables cannot be saved.

Example input & output:

Input	Table	Plot
Country: China Province: Zhejiang Year/month : 2	(2020-02-01,599) (2020-02-02,661) (2020-02-03,724) (2020-02-04,829) (2020-02-05,895) (2020-02-06,954) (2020-02-07,1006) (2020-02-08,1048) (2020-02-09,1075) (2020-02-10,1092) (2020-02-11,1117) (2020-02-12,1131) (2020-02-13,1145) (2020-02-14,1155) (2020-02-15,1162) (2020-02-16,1167) (2020-02-17,1171) (2020-02-18,1172) (2020-02-19,1174) (2020-02-20,1175) (2020-02-21,1203) (2020-02-22,1205) (2020-02-23,1205) (2020-02-24,1205) (2020-02-25,1205) (2020-02-26,1205) (2020-02-27,1205) (2020-02-28,1205) (2020-02-29,1205)	
Country: US Province: Blank Year/month : 2020	(2020-01-31,9802) (2020-02-29,79356) (2020-03-31,82279) (2020-04-30,83956) (2020-05-31,84146) (2020-06-30,84785) (2020-07-31,87655) (2020-08-31,89914) (2020-09-30,90545) (2020-10-31,91366) (2020-11-30,92902) (2020-12-29,95797)	

Why do we design this query in this way? Since we have a large amount of data, our query needs to fetch the data that we want quickly and conveniently. If we want to know how many confirmed cases in China in 2020-3-12, we can input “China” and “3”. Then, we can easily find 2020-3-12 in the result. A large amount of data is also difficult to interpret, that’s why include a plot. We can get a rough idea of the trend from the plot easily.

Query 2: Which country has the fastest (or second or third ... to the tenth place) COVID-19 diagnosis in a certain month (users can input the month for themselves)? The rate of COVID-19 diagnosis is defined as (confirmed cases this month- last month)/last month. According to this definition, users can input month (2-12) and n th place (1-10). The output will be the country name and the rate. If a country’s last month confirmed cases is 0, then this country does not have the diagnosis rate in this month.

Example input & output:

Input	Output
Month: 12 N th place: 10	Cyprus 1.0175106483672502

2. Database

a. The database used

Two databases are used: countries-aggregated, time-series-19-covid-combined.

“countries-aggregated” contains date, country, confirmed, recovered and deaths.

“time-series-19-covid-combined” contains the information of provinces of some countries (but not all countries).

List of countries which have the province information: Australia, Canada, China, Denmark, France, Netherlands, United Kingdom. Although there is province information of US in us_confirmed.csv, the data is for the specific counties in a certain state. It is difficult to get the data of whole states. So I decided not to include province information of US.

I only focused on the confirmed cases in this project.

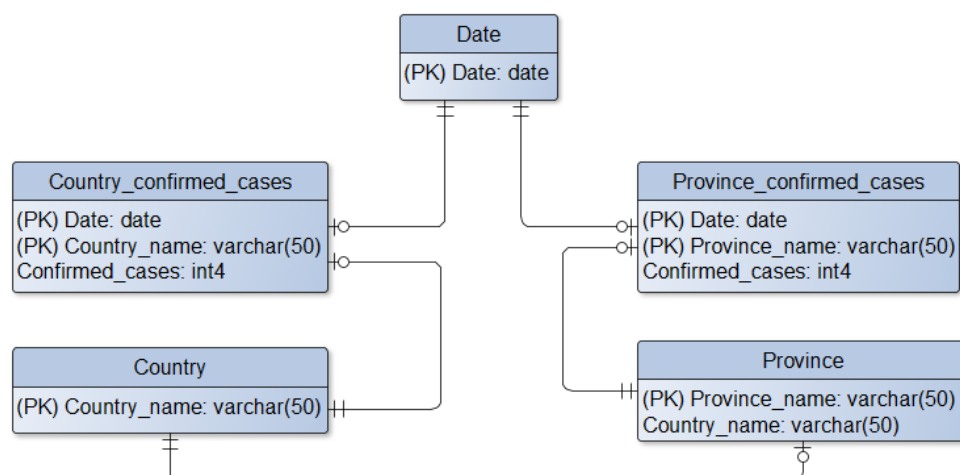
Note that all the databases have the date from 2020-1-22 to 2020-12-29.

b. The design of the database

Entities represent various real-world notion. The entities in the provided dataset can be country, province and date. The attribute of the country entity is just the country name (also serves as primary key). The attribute of date is a list of the date of 2020 (from 2020-1-1 to 2020-12-31) (also serves as primary key). Since not all countries have province information, it is appropriate to set the province name as the primary key in the province entity. The second attribute will be the country that the province belongs to. It links to the country entity (as a foreign key). There is no need to add an id to the province table since there are no two countries that have the provinces with the same name. Another possible way to build the province entity is to set the country name as the primary key. So the province name will be a multivalued attribute. However, I am not sure how to create a multivalued attribute in pgAdmin.

The confirmed cases will only be meaningful if both date and country/province name are provided. Therefore, it is necessary to create two tables that contain country/province name, date and confirmed cases. In these two tables, name and date will be the primary keys. It is a bit like the composite unique attributes. None of the three attributes is unique, but the combination of date and country/province name is unique. We link the tables with the entities and obtain the following ER diagram.

c. Database's ERD



Some explanation to the ERD:

Not all countries have province information, but a province must belong to a country. Thus, the cardinality constraint of country to province is optional one. The opposite cardinality constraint should be mandatory one.

The cardinality constraints of date to country_confirmed_cases and province_confirmed_cases are optional one, since we don't have records for every day in 2020. Similarly, the cardinality constraints of country/province to country_confirmed_cases and province_confirmed_cases are also optional one. It seems that we have data for all countries/province. However, data missing should be allowed. It is possible that we do not have data for a certain country/province in a future date. The date, country and province which appears in country_confirmed_cases and province_confirmed_cases should be found in the date, country and province entities. So the opposite cardinality constraints should be mandatory one.

One may ask why it is necessary to create an entity with just one attribute (country & date). This is because both country_confirmed_cases and province_confirmed_cases have two primary keys. Two primary keys should be linked to other entities (form foreign keys). Otherwise, pgAdmin will give you an error when you are creating the tables. The right-hand side is an example from the dvdrental database. Another reason is that the country and date are real-world notions. They agree with the definition of the entity. Moreover, the country name is the only information we can get from the provided data. We can add more information to the country entity, such as population, continent etc, if we have that information. We can also add holidays etc to date entity. Then, we can do other interesting queries such as which continent has the highest confirmed cases or whether holidays increase the spread of COVID-19 and so on.



3. Details on the python script

The python script can be divided into two parts. The first part defines all variables and functions. The second part builds up the user interface, possesses the data derived from SQL query and displays the data in the user interface.

a. The first part (focused on functions)

connect_postgres: to connect Python with PostgreSQL.

For query 1:

1. return_country: return the query result of the input country. Query from "Country_confirmed_cases" using the pre-defined SQL function "return_country" (code can be found in "return_country.sql" file). The SQL function basically contains if statement to judge the input month/year and find the country using "where" clause.
2. check_country_province: check whether the input country and province match. If not, the function returns []. Query from Country_province (using "SELECT Country_name from Country_province where country_name = country (input) and province_name = province (input)"). If country and province do not match, the return will be [].
3. return_province: return the query result of the input province. Query from "province_confirmed_cases" using pre-defined SQL function "return_province" (similar to

"return_country").

4. plot: gather necessary information for plotting the data. Since the query return is a list of string (date and confirmed cases are combined into one string, see figure), we need to extract date and confirmed case into two lists respectively (using regex expression).

For query 2:

1. return_rate: return the necessary data for the calculation of the rate of diagnosis. For example, if the input month is 2, the function will return a list of confirmed cases of Jan. 31st and Feb. 29th of each country using pre-defined SQL function "return_rate" (similar to "return_country"). The rate calculation is done in the pygame part.

```
In [2]: query1_return
Out[2]:
[(('2020-12-01,13791945'),),
 ('2020-12-02,13992765'),),
 ('2020-12-03,14212649'),),
 ('2020-12-04,14442788'),),
 ('2020-12-05,14658875'),),
 ('2020-12-06,14840269'),),
 ('2020-12-07,15030195'),),
 ('2020-12-08,15249204'),),
 ('2020-12-09,15469292'),),
 ('2020-12-10,15689380'),),
 ('2020-12-11,15909468'),),
 ('2020-12-12,16129556'),),
 ('2020-12-13,16349644'),),
 ('2020-12-14,16569732'),),
 ('2020-12-15,16789820'),),
 ('2020-12-16,17009908'),),
 ('2020-12-17,17230000'),),
 ('2020-12-18,17450092'),),
 ('2020-12-19,17670184'),),
 ('2020-12-20,17890276'),),
 ('2020-12-21,18110368'),),
 ('2020-12-22,18330460'),),
 ('2020-12-23,18550552'),),
 ('2020-12-24,18770644'),),
 ('2020-12-25,18990736'),),
 ('2020-12-26,19210828'),),
 ('2020-12-27,19430920'),),
 ('2020-12-28,19651012'),),
 ('2020-12-29,19871104'),),
 ('2020-12-30,20091196'),),
 ('2020-12-31,20311288'),),
 ('2021-01-01,20531380'),),
 ('2021-01-02,20751472'),),
 ('2021-01-03,20971564'),),
 ('2021-01-04,21191656'),),
 ('2021-01-05,21411748'),),
 ('2021-01-06,21631840'),),
 ('2021-01-07,21851932'),),
 ('2021-01-08,22072024'),),
 ('2021-01-09,22292116'),),
 ('2021-01-10,22512208'),),
 ('2021-01-11,22732300'),),
 ('2021-01-12,22952392'),),
 ('2021-01-13,23172484'),),
 ('2021-01-14,23392576'),),
 ('2021-01-15,23612668'),),
 ('2021-01-16,23832760'),),
 ('2021-01-17,24052852'),),
 ('2021-01-18,24272944'),),
 ('2021-01-19,24493036'),),
 ('2021-01-20,24713128'),),
 ('2021-01-21,24933220'),),
 ('2021-01-22,25153312'),),
 ('2021-01-23,25373404'),),
 ('2021-01-24,25593496'),),
 ('2021-01-25,25813588'),),
 ('2021-01-26,26033680'),),
 ('2021-01-27,26253772'),),
 ('2021-01-28,26473864'),),
 ('2021-01-29,26693956'),),
 ('2021-01-30,26914048'),),
 ('2021-01-31,27134140'),),
 ('2021-02-01,27354232'),),
 ('2021-02-02,27574324'),),
 ('2021-02-03,27794416'),),
 ('2021-02-04,28014508'),),
 ('2021-02-05,28234600'),),
 ('2021-02-06,28454692'),),
 ('2021-02-07,28674784'),),
 ('2021-02-08,28894876'),),
 ('2021-02-09,29114968'),),
 ('2021-02-10,29335060'),),
 ('2021-02-11,29555152'),),
 ('2021-02-12,29775244'),),
 ('2021-02-13,29995336'),),
 ('2021-02-14,30215428'),),
 ('2021-02-15,30435520'),),
 ('2021-02-16,30655612'),),
 ('2021-02-17,30875704'),),
 ('2021-02-18,31095796'),),
 ('2021-02-19,31315888'),),
 ('2021-02-20,31535980'),),
 ('2021-02-21,31756072'),),
 ('2021-02-22,31976164'),),
 ('2021-02-23,32196256'),),
 ('2021-02-24,32416348'),),
 ('2021-02-25,32636440'),),
 ('2021-02-26,32856532'),),
 ('2021-02-27,33076624'),),
 ('2021-02-28,33296716'),),
 ('2021-03-01,33516808'),),
 ('2021-03-02,33736900'),),
 ('2021-03-03,33956992'),),
 ('2021-03-04,34177084'),),
 ('2021-03-05,34397176'),),
 ('2021-03-06,34617268'),),
 ('2021-03-07,34837360'),),
 ('2021-03-08,35057452'),),
 ('2021-03-09,35277544'),),
 ('2021-03-10,35497636'),),
 ('2021-03-11,35717728'),),
 ('2021-03-12,35937820'),),
 ('2021-03-13,36157912'),),
 ('2021-03-14,36378004'),),
 ('2021-03-15,36598096'),),
 ('2021-03-16,36818188'),),
 ('2021-03-17,37038280'),),
 ('2021-03-18,37258372'),),
 ('2021-03-19,37478464'),),
 ('2021-03-20,37698556'),),
 ('2021-03-21,37918648'),),
 ('2021-03-22,38138740'),),
 ('2021-03-23,38358832'),),
 ('2021-03-24,38578924'),),
 ('2021-03-25,38799016'),),
 ('2021-03-26,39019108'),),
 ('2021-03-27,39239200'),),
 ('2021-03-28,39459292'),),
 ('2021-03-29,39679384'),),
 ('2021-03-30,39899476'),),
 ('2021-03-31,40119568'),),
 ('2021-04-01,40339660'),),
 ('2021-04-02,40559752'),),
 ('2021-04-03,40779844'),),
 ('2021-04-04,40999936'),),
 ('2021-04-05,41220028'),),
 ('2021-04-06,41440120'),),
 ('2021-04-07,41660212'),),
 ('2021-04-08,41880304'),),
 ('2021-04-09,42100396'),),
 ('2021-04-10,42320488'),),
 ('2021-04-11,42540580'),),
 ('2021-04-12,42760672'),),
 ('2021-04-13,42980764'),),
 ('2021-04-14,43200856'),),
 ('2021-04-15,43420948'),),
 ('2021-04-16,43641040'),),
 ('2021-04-17,43861132'),),
 ('2021-04-18,44081224'),),
 ('2021-04-19,44301316'),),
 ('2021-04-20,44521408'),),
 ('2021-04-21,44741500'),),
 ('2021-04-22,44961592'),),
 ('2021-04-23,45181684'),),
 ('2021-04-24,45401776'),),
 ('2021-04-25,45621868'),),
 ('2021-04-26,45841960'),),
 ('2021-04-27,46062052'),),
 ('2021-04-28,46282144'),),
 ('2021-04-29,46502236'),),
 ('2021-04-30,46722328'),),
 ('2021-05-01,46942420'),),
 ('2021-05-02,47162512'),),
 ('2021-05-03,47382604'),),
 ('2021-05-04,47602696'),),
 ('2021-05-05,47822788'),),
 ('2021-05-06,48042880'),),
 ('2021-05-07,48262972'),),
 ('2021-05-08,48483064'),),
 ('2021-05-09,48703156'),),
 ('2021-05-10,48923248'),),
 ('2021-05-11,49143340'),),
 ('2021-05-12,49363432'),),
 ('2021-05-13,49583524'),),
 ('2021-05-14,49803616'),),
 ('2021-05-15,50023708'),),
 ('2021-05-16,50243800'),),
 ('2021-05-17,50463892'),),
 ('2021-05-18,50683984'),),
 ('2021-05-19,50904076'),),
 ('2021-05-20,51124168'),),
 ('2021-05-21,51344260'),),
 ('2021-05-22,51564352'),),
 ('2021-05-23,51784444'),),
 ('2021-05-24,52004536'),),
 ('2021-05-25,52224628'),),
 ('2021-05-26,52444720'),),
 ('2021-05-27,52664812'),),
 ('2021-05-28,52884904'),),
 ('2021-05-29,53104996'),),
 ('2021-05-30,53325088'),),
 ('2021-05-31,53545180'),),
 ('2021-06-01,53765272'),),
 ('2021-06-02,53985364'),),
 ('2021-06-03,54205456'),),
 ('2021-06-04,54425548'),),
 ('2021-06-05,54645640'),),
 ('2021-06-06,54865732'),),
 ('2021-06-07,55085824'),),
 ('2021-06-08,55305916'),),
 ('2021-06-09,55526008'),),
 ('2021-06-10,55746100'),),
 ('2021-06-11,55966192'),),
 ('2021-06-12,56186284'),),
 ('2021-06-13,56406376'),),
 ('2021-06-14,56626468'),),
 ('2021-06-15,56846560'),),
 ('2021-06-16,57066652'),),
 ('2021-06-17,57286744'),),
 ('2021-06-18,57506836'),),
 ('2021-06-19,57726928'),),
 ('2021-06-20,57947020'),),
 ('2021-06-21,58167112'),),
 ('2021-06-22,58387204'),),
 ('2021-06-23,58607296'),),
 ('2021-06-24,58827388'),),
 ('2021-06-25,59047480'),),
 ('2021-06-26,59267572'),),
 ('2021-06-27,59487664'),),
 ('2021-06-28,59707756'),),
 ('2021-06-29,59927848'),),
 ('2021-06-30,60147940'),),
 ('2021-07-01,60368032'),),
 ('2021-07-02,60588124'),),
 ('2021-07-03,60808216'),),
 ('2021-07-04,61028308'),),
 ('2021-07-05,61248400'),),
 ('2021-07-06,61468492'),),
 ('2021-07-07,61688584'),),
 ('2021-07-08,61908676'),),
 ('2021-07-09,62128768'),),
 ('2021-07-10,62348860'),),
 ('2021-07-11,62568952'),),
 ('2021-07-12,62789044'),),
 ('2021-07-13,63009136'),),
 ('2021-07-14,63229228'),),
 ('2021-07-15,63449320'),),
 ('2021-07-16,63669412'),),
 ('2021-07-17,63889504'),),
 ('2021-07-18,64109596'),),
 ('2021-07-19,64329688'),),
 ('2021-07-20,64549780'),),
 ('2021-07-21,64769872'),),
 ('2021-07-22,64989964'),),
 ('2021-07-23,65210056'),),
 ('2021-07-24,65430148'),),
 ('2021-07-25,65650240'),),
 ('2021-07-26,65870332'),),
 ('2021-07-27,66090424'),),
 ('2021-07-28,66310516'),),
 ('2021-07-29,66530608'),),
 ('2021-07-30,66750700'),),
 ('2021-07-31,66970792'),),
 ('2021-08-01,67190884'),),
 ('2021-08-02,67410976'),),
 ('2021-08-03,67631068'),),
 ('2021-08-04,67851160'),),
 ('2021-08-05,68071252'),),
 ('2021-08-06,68291344'),),
 ('2021-08-07,68511436'),),
 ('2021-08-08,68731528'),),
 ('2021-08-09,68951620'),),
 ('2021-08-10,69171712'),),
 ('2021-08-11,69391804'),),
 ('2021-08-12,69611896'),),
 ('2021-08-13,69831988'),),
 ('2021-08-14,70052080'),),
 ('2021-08-15,70272172'),),
 ('2021-08-16,70492264'),),
 ('2021-08-17,70712356'),),
 ('2021-08-18,70932448'),),
 ('2021-08-19,71152540'),),
 ('2021-08-20,71372632'),),
 ('2021-08-21,71592724'),),
 ('2021-08-22,71812816'),),
 ('2021-08-23,72032908'),),
 ('2021-08-24,72252996'),),
 ('2021-08-25,72473088'),),
 ('2021-08-26,72693180'),),
 ('2021-08-27,72913272'),),
 ('2021-08-28,73133364'),),
 ('2021-08-29,73353456'),),
 ('2021-08-30,73573548'),),
 ('2021-08-31,73793640'),),
 ('2021-09-01,74013732'),),
 ('2021-09-02,74233824'),),
 ('2021-09-03,74453916'),),
 ('2021-09-04,74674008'),),
 ('2021-09-05,74894100'),),
 ('2021-09-06,75114192'),),
 ('2021-09-07,75334284'),),
 ('2021-09-08,75554376'),),
 ('2021-09-09,75774468'),),
 ('2021-09-10,75994560'),),
 ('2021-09-11,76214652'),),
 ('2021-09-12,76434744'),),
 ('2021-09-13,76654836'),),
 ('2021-09-14,76874928'),),
 ('2021-09-15,77095020'),),
 ('2021-09-16,77315112'),),
 ('2021-09-17,77535204'),),
 ('2021-09-18,77755296'),),
 ('2021-09-19,77975388'),),
 ('2021-09-20,78195480'),),
 ('2021-09-21,78415572'),),
 ('2021-09-22,78635664'),),
 ('2021-09-23,78855756'),),
 ('2021-09-24,79075848'),),
 ('2021-09-25,79295940'),),
 ('2021-09-26,79516032'),),
 ('2021-09-27,79736124'),),
 ('2021-09-28,79956216'),),
 ('2021-09-29,80176308'),),
 ('2021-09-30,80396400'),),
 ('2021-10-01,80616492'),),
 ('2021-10-02,80836584'),),
 ('2021-10-03,81056676'),),
 ('2021-10-04,81276768'),),
 ('2021-10-05,81496860'),),
 ('2021-10-06,81716952'),),
 ('2021-10-07,81937044'),),
 ('2021-10-08,82157136'),),
 ('2021-10-09,82377228'),),
 ('2021-10-10,82597320'),),
 ('2021-10-11,82817412'),),
 ('2021-10-12,83037504'),),
 ('2021-10-13,83257596'),),
 ('2021-10-14,83477688'),),
 ('2021-10-15,83697780'),),
 ('2021-10-16,83917872'),),
 ('2021-10-17,84137964'),),
 ('2021-10-18,84358056'),),
 ('2021-10-19,84578148'),),
 ('2021-10-20,84798240'),),
 ('2021-10-21,85018332'),),
 ('2021-10-22,85238424'),),
 ('2021-10-23,85458516'),),
 ('2021-10-24,85678608'),),
 ('2021-10-25,85898700'),),
 ('2021-10-26,86118792'),),
 ('2021-10-27,86338884'),),
 ('2021-10-28,86558976'),),
 ('2021-10-29,86779068'),),
 ('2021-10-30,86999160'),),
 ('2021-10-31,87219252'),),
 ('2021-11-01,87439344'),),
 ('2021-11-02,87659436'),),
 ('2021-11-03,87879528'),),
 ('2021-11-04,88099620'),),
 ('2021-11-05,88319712'),),
 ('2021-11-06,88539804'),),
 ('2021-11-07,88759896'),),
 ('2021-11-08,88979988'),),
 ('2021-11-09,89199980'),),
 ('2021-11-10,89420072'),),
 ('2021-11-11,89640164'),),
 ('2021-11-12,89860256'),),
 ('2021-11-13,90080348'),),
 ('2021-11-14,90300440'),),
 ('2021-11-15,90520532'),),
 ('2021-11-16,90740624'),),
 ('2021-11-17,90960716'),),
 ('2021-11-18,91180808'),),
 ('2021-11-19,91400900'),),
 ('2021-11-20,91620992'),),
 ('2021-11-21,91841084'),),
 ('2021-11-22,92061176'),),
 ('2021-11-23,92281268'),),
 ('2021-11-24,92501360'),),
 ('2021-11-25,92721452'),),
 ('2021-11-26,92941544'),),
 ('2021-11-27,93161636'),),
 ('2021-11-28,93381728'),),
 ('2021-11-29,93601820'),),
 ('2021-11-30,93821912'),),
 ('2021-12-01,94042004'),),
 ('2021-12-02,94262096'),),
 ('2021-12-03,94482188'),),
 ('2021-12-04,94702280'),),
 ('2021-12-05,94922372'),),
 ('2021-12-06,95142464'),),
 ('2021-12-07,95362556'),),
 ('2021-12-08,95582648'),),
 ('2021-12-09,95802740'),),
 ('2021-12-10,96022832'),),
 ('2021-12-11,96242924'),),
 ('2021-12-12,96463016'),),
 ('2021-12-13,96683108'),),
 ('2021-12-14,96903200'),),
 ('2021-12-15,97123292'),),
 ('2021-12-16,97343384'),),
 ('2021-12-17,97563476'),),
 ('2021-12-18,97783568'),),
 ('2021-12-19,98003660'),),
 ('2021-12-20,98223752'),),
 ('2021-12-21,98443844'),),
 ('2021-12-22,98663936'),),
 ('2021-12-23,98884028'),),
 ('2021-12-24,99104120'),),
 ('2021-12-25,99324212'),),
 ('2021-12-26,99544304'),),
 ('2021-12-27,99764396'),),
 ('2021-12-28,99984488'),),
 ('2021-12-29,100204500'),),
 ('2021-12-30,100424592'),),
 ('2021-12-31,100644684'),),
 ('2022-01-01,100864776'),),
 ('2022-01-02,101084868'),),
 ('2022-01-03,101304960'),),
 ('2022-01-04,101525052'),),
 ('2022-01-05,101745144'),),
 ('2022-01-06,101965236'),),
 ('2022-01-07,102185328'),),
 ('2022-01-08,102405420'),),
 ('2022-01-09,102625512'),),
 ('2022-01-10,102845604'),),
 ('2022-01-11,103065696'),),
 ('2022-01-12,103285788'),),
 ('2022-01-13,103505880'),),
 ('2022-01-14,103725972'),),
 ('2022-01-15,103946064'),),
 ('2022-01-16,104166156'),),
 ('2022-01-17,104386248'),),
 ('2022-01-18,104606340'),),
 ('2022-01-19,104826432'),),
 ('2022-01-20,105046524'),),
 ('2022-01-21,105266616'),),
 ('2022-01-22,105486708'),),
 ('2022-01-23,105706800'),),
 ('2022-01-24,105926892'),),
 ('2022-01-25,106146984'),),
 ('2022-01-26,106367076'),),
 ('2022-01-27,106587168'),),
 ('2022-01-28,106807260'),),
 ('2022-01-29,107027352'),),
 ('2022-01-30,107247444'),),
 ('2022-01-31,107467536'),),
 ('2022-02-01,107687628'),),
 ('2022-02-02,107907720'),),
 ('2022-02-03,108127812'),),
 ('2022-02-04,108347904'),),
 ('2022-02-05,108568000'),),
 ('2022-02-06,108788092'),),
 ('2022-02-07,109008184'),),
 ('2022-02-08,109228276'),),
 ('2022-02-09,109448368'),),
 ('2022-02-10,109668460'),),
 ('2022-02-11,109888552'),),
 ('2022-02-12,110108644'),),
 ('2022-02-13,110328736'),),
 ('2022-02-14,110548828'),),
 ('2022-02-15,110768920'),),
 ('2022-02-16,110989012'),),
 ('2022-02-17,111209104'),),
 ('2022-02-18,111429196'),),
 ('2022-02-19,111649288'),),
 ('2022-02-20,111869380'),),
 ('2022-02-21,112089472'),),
 ('2022-02-22,112309564'),),
 ('2022-02-23,112529656'),),
 ('2022-02-24,112749748'),),
 ('2022-02-25,112969840'),),
 ('2022-02-26,113189932'),),
 ('2022-02-27,113410024'),),
 ('2022-02-28,113630116'),),
 ('2022-02-29,113850208'),),
 ('2022-03-01,114070300'),),
 ('2022-03-02,114290392'),),
 ('2022-03-03,114510484'),),
 ('2022-03-04,114730576'),),
 ('2022-03-05,114950668'),),
 ('2022-03-06,115170760'),),
 ('2022-03-07,115390852'),),
 ('2022-03-08,115610944'),),
 ('2022-03-09,115831036'),),
 ('2022-03-10,116051128'),),
 ('2022-03-11,116271220'),),
 ('2022-03-12,116491312'),),
 ('2022-03-13,116711404'),),
 ('2022-03-14,116931496'),),
 ('2022-03-15,117151588'),),
 ('2022-03-16,117371680'),),
 ('2022-03-17,117591772'),),
 ('2022-03-18,117811864'),),
 ('2022-03-19,118031956'),),
 ('2022-03-20,118252048'),),
 ('2022-03-21,118472140'),),
 ('2022-03-22,118692232'),),
 ('2022-03-23,118912324'),),
 ('2022-03-24,119132416'),),
 ('2022-03-25,119352508'),),
 ('2022-03-26,119572600'),),
 ('2022-03-27,119792692'),),
 ('2022-03-28,120012784'),),
 ('2022-03-29,120232876'),),
 ('2022-03-30,120452968'),),
 ('2022-03-31,120673060'),),
 ('2022-04-01,120893152'),),
 ('2022-04-02,121113244'),),
 ('2022-04-03,121333336'),),
 ('2022-04-04,121553428'),),
 ('2022-04-05,121773520'),),
 ('2022-04-06,121993612'),),
 ('2022-04-07,122213704'),),
 ('2022-04-08,122433796'),),
 ('2022-04-09,122653888'),),
 ('2022-04-10,122873980'),),
 ('2022-04-11,123094072'),),
 ('2022-04-12,123314164'),),
 ('2022-04-13,123534256'),),
 ('2022-04-14,123754348'),),
 ('2022-04-15,123974440'),),
 ('2022-04-16,124194532'),),
 ('2022-04-17,124414624'),),
 ('2022-04-18,124634716'),),
 ('2022-04-19,124854808'),),
 ('2022-04-20,125074900'),),
 ('2022-04-21,125294992'),),
 ('2022-04-22,125515084'),),
 ('2022-04-23,125735176'),),
 ('2022-04-24,125955268'),),
 ('2022-04-25,126175360'),),
 ('2022-04-26,126395452'),),
 ('2022-04-27,126615544'),),
 ('2022-04-28,126835636'),),
 ('2022-04-29,127055728'),),
 ('2022-04-30,127275820'),),
 ('2022-05-01,127495912'),),
 ('2022-05-02,12771600
```