



SHOPEE CODE LEAGUE 2021

Guidelines for Approaching the Multi-Channel Contacts Problem

Competition Details

This competition was launched on **6 March 2021**.

Please refer to the problem statements [here](#).

Task:

For each ticket, identify all contacts from each user if they have the same contact information. For the purpose of this question, assume that all contacts from the same Phone Number / Email are the same user.

Approach

There are many ways to solve this, the challenge is in selecting the most time efficient way to process the large amount of data. In this solution guideline, the recommended approach is to make use of classes to avoid having to run multiple loops.

Suggested Solution

- 1) Create 4 empty dictionaries
 - a) User
 - b) Emails
 - c) Phone
 - d) OrderID
- 2) Each dictionary from b) to d) will be in this format
 - a) {Email : userid}
 - i) Where userid is an unique value that is set by you
- 3) Create a class called User (technically any data structure that can hold multiple data types will do)
 - a) Class should contain 5 values -
 - i) List of Emails
 - ii) List of Phones
 - iii) List of Order Ids
 - iv) List of OrderId
 - v) No. of Contacts
 - vi) List of Ticket Ids
- 4) Loop through each ticket
 - a) Create an empty list of existing userid
 - b) Get the Email Value from the Email Dictionary - if there is an existing user id with a that same email, add it to a list of existing userid for this ticket
 - c) Get the Phone Value from the Phone Dictionary - if there is an existing user id with that same phone, add it to a list of existing userid.
 - d) Get the OrderID Value from the Phone Dictionary - if there is an existing user id with that same OrderID, add it to a list of existing userid.
 - e) If there is no existing userid matching this ticket
 - i) Initiate a new user object,
 - ii) Add the existing email. Phone and orderid to this user.
 - iii) Add the contact value to this user
 - iv) Add the value '<email>': userid to the email dictionary

- v) Add the value '<phone>': userid to the phone dictionary
 - vi) Add the value '<orderid>': userid to the orderid dictionary
 - vii) Add this user id to the user dictionary.
 - (1) {user_id: <User Object>}
 - f) If the list of existing userids is more than 0;
 - i) Choose one the first existing userid, add this current ticket's email phone and order id to the existing user object.
 - ii) Go through the rest of the existing userid and loop through each of the other existing userids
 - (1) Combine the emails list, phone list, orderid list between the users
 - (2) Sum up contacts between the users.
 - g) Use the user object that this ticket referenced to
 - i) Replace all Emails in the email dictionary with that user with the new userid
 - ii) Replace all Phone in the phone dictionary with that user with the new userid
 - iii) Replace all OrderId in the Order dictionary with that user with the new userid
- 5) Use the values from 3vi and 3v to get the final output by looping through each UserID and concatenating the list of ticket ids into a string and getting the sum of contacts.