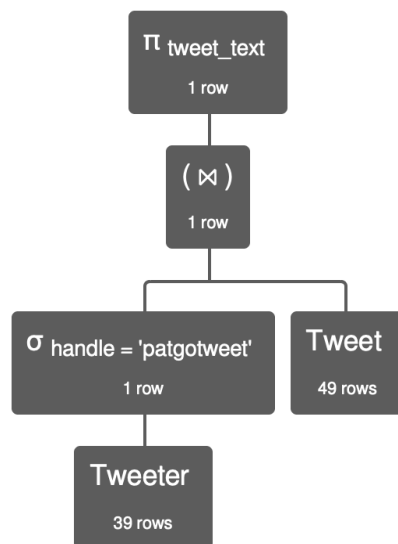


1. Find the text of all tweets that were posted by the tweeter with the handle 'patgottweet'.

a) Relational Algebra

$\pi_{\text{tweet\_text}} ((\sigma_{\text{handle} = \text{'patgottweet'}} \text{Tweeter}) \bowtie \text{Tweet})$

b) Parse Tree



c) Result

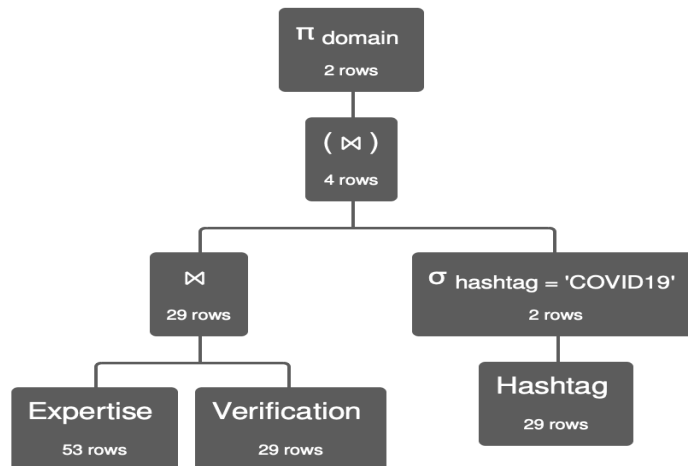
Tweet.tweet_text	
'@TheValuesVoter @MrsPerrin I'm in one of those states. It's a big Trump state so Covid is all a hoax here and masks don't work. It's a strange place to live right now.... and a bit scary and frustrating.'	

2. List the domains of expertise for checkers who have verified tweets that have the hashtag “COVID19”.  
 (Note: The hashtag value is all in capital letters.)

a) Relational Algebra

$\pi_{\text{domain}} (\text{Expertise} \bowtie \text{Verification} \bowtie (\sigma_{\text{hashtag} = \text{'COVID19'}} \text{Hashtag}))$

b) Parse Tree



$\pi_{\text{domain}} ( ( \text{Expertise} \bowtie \text{Verification} ) \bowtie ( \sigma_{\text{hashtag} = \text{'COVID19'}} \text{Hashtag} ) )$

c) Result

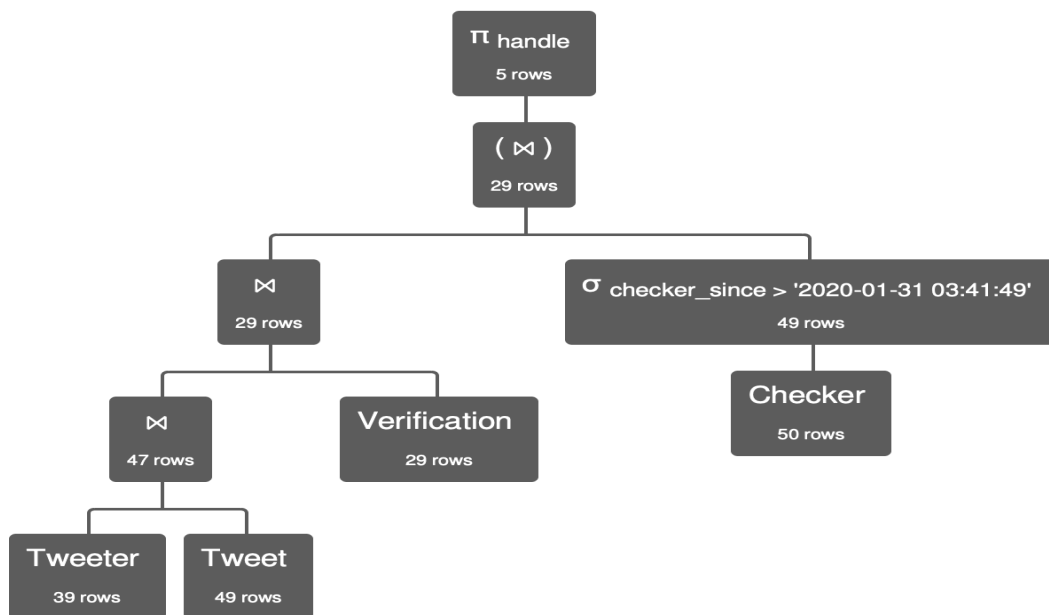
Expertise.domain
'Health Service Quality'
'Public Health'

3. List the handles of Tweeters who have posted a tweet that has been verified by a Checker who has been a checker since “2020-01-31 03:41:49”.

a) Relational Algebra

$\pi_{\text{handle}} (\text{Tweeter} \bowtie \text{Tweet} \bowtie \text{Verification} \bowtie (\sigma_{\text{checker\_since} > '2020-01-31\ 03:41:49'} \text{Checker}))$

b) Parse Tree



$\pi_{\text{handle}} ( ( ( \text{Tweeter} \bowtie \text{Tweet} ) \bowtie \text{Verification} ) \bowtie ( \sigma_{\text{checker\_since} > '2020-01-31\ 03:41:49'} \text{Checker} ) )$

c) Result

<b>Tweeter.handle</b>
'theblack_abyss'
'SandyInCalif'
'NecessaryPaper'
'mptrottier'
'oceanviewmom'

4. List the evidence-providing users' ids and the associated checkers' ids where the users live in the state of CA and the checkers used evidence submitted by users who are not checkers themselves.

Example:

Let's say that we have the following evidence and info about who submitted it:

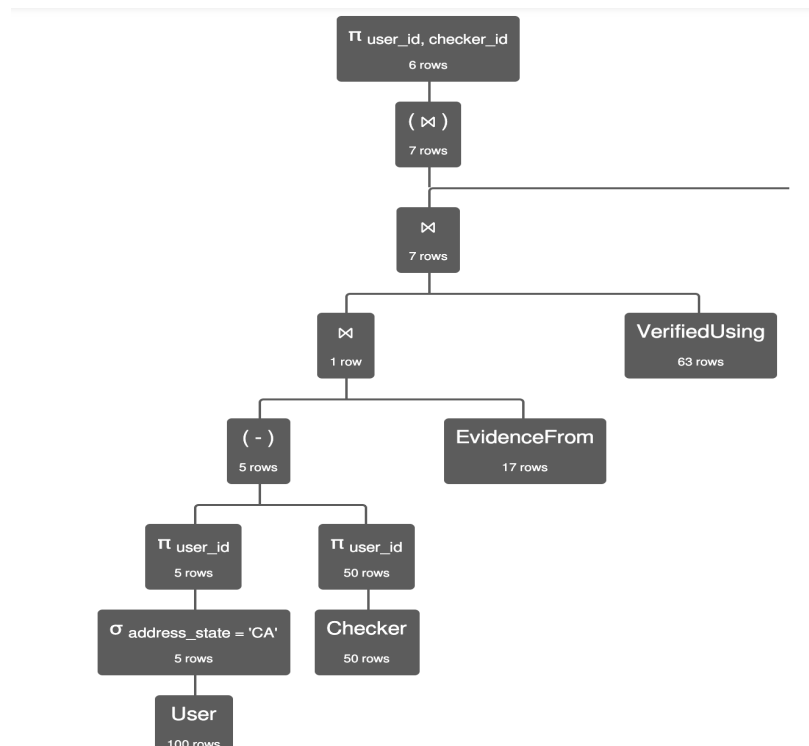
ev_id	url	Submitted by
0	"http://foo.com"	checker0
1	"http://baz.org"	user1
2	"http://baz.org"	user2

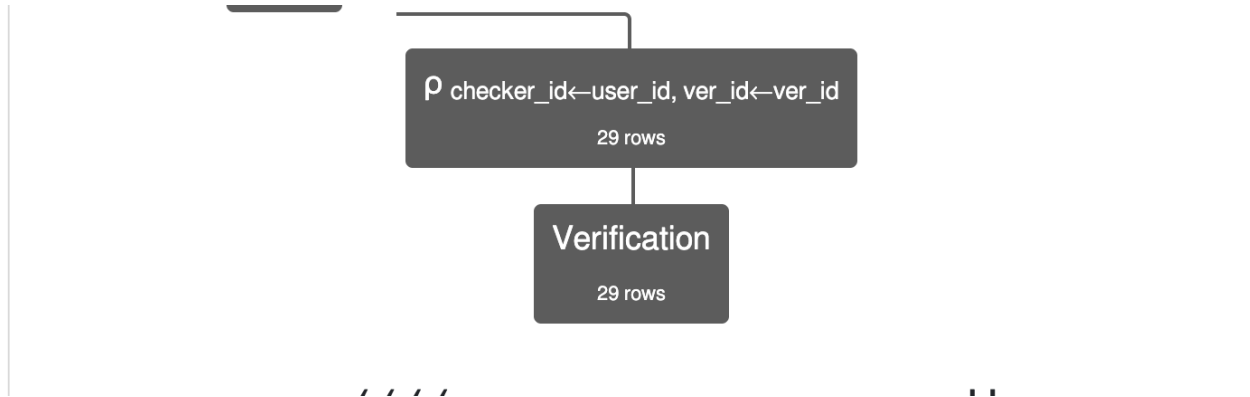
In your query (relational algebra), you need to consider only the evidence submitted by user1 and user2 (i.e., ev\_id =1 and ev\_id = 2, respectively) if they live in California. Your result should not include ev\_id = 0 as it was submitted by a checker (namely checker0).

a) Relational Algebra

$\pi_{\text{user\_id}, \text{checker\_id}} ( ((\pi_{\text{user\_id}} \sigma_{\text{address\_state}='CA'} \text{User} - \pi_{\text{user\_id}} \text{Checker})) \bowtie \text{EvidenceFrom} \bowtie \text{VerifiedUsing} \bowtie \rho_{\text{checker\_id} \leftarrow \text{user\_id}, \text{ver\_id} \leftarrow \text{ver\_id}} \text{Verification})$

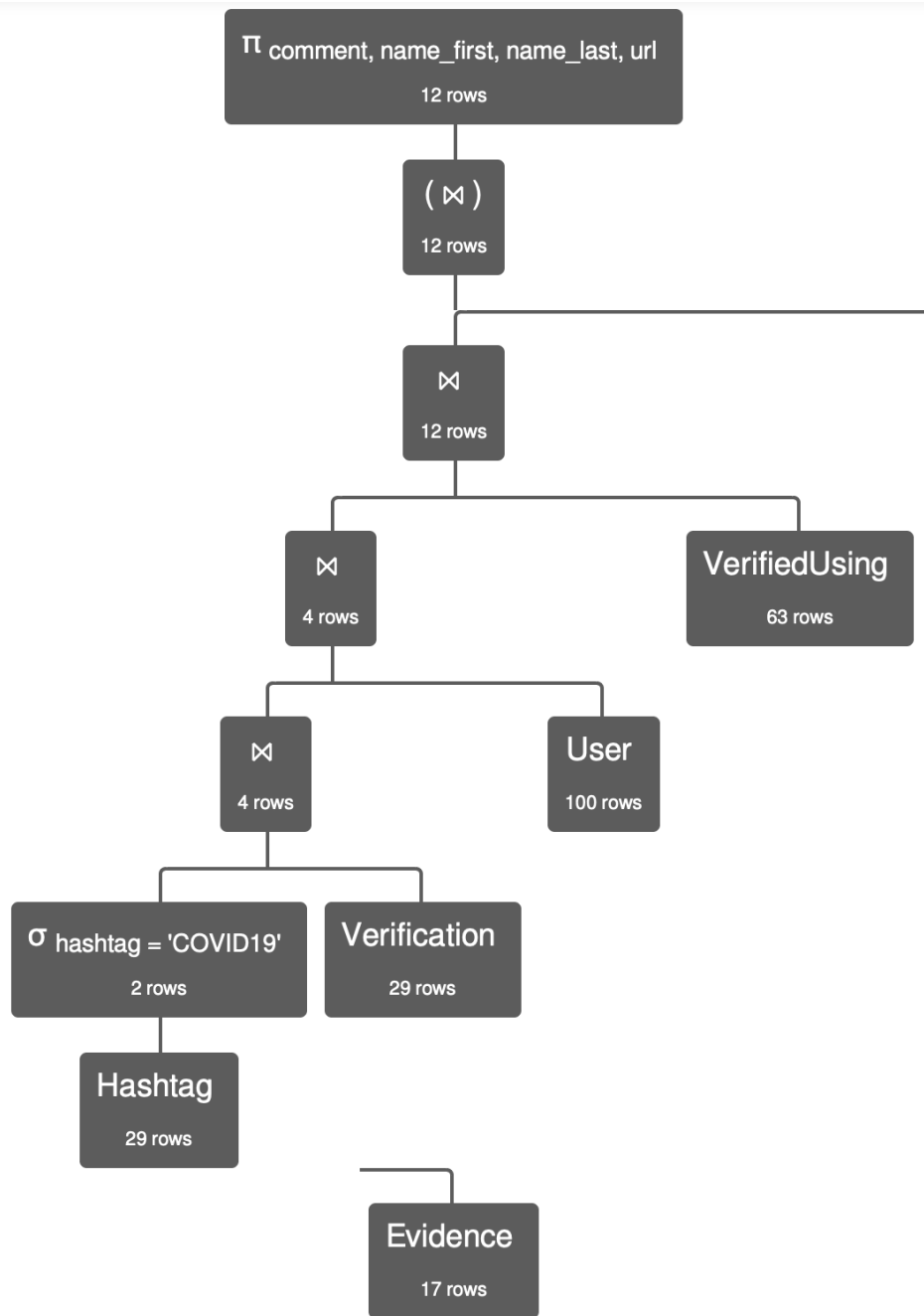
b) Parse Tree





c) Result

User.user_id	Verification.checker_id
44	0
44	3
44	6
44	15
44	22
44	21



c) Result

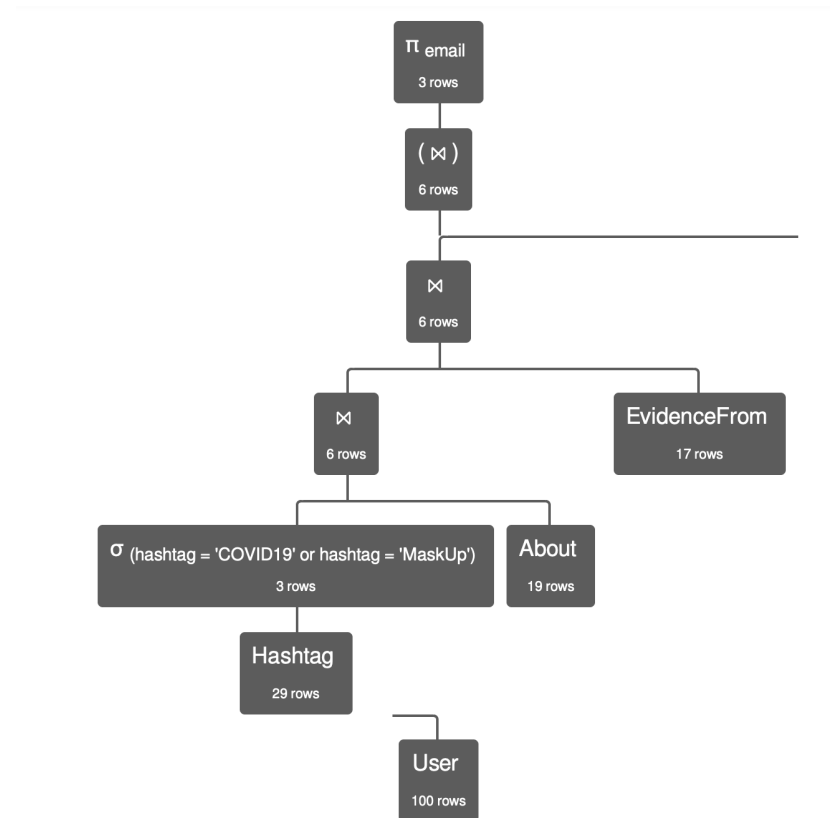
Verification.comment	User.name_first	User.name_last	Evidence.url
'Masks works! Check the CDC'	'Jonathan'	'Howard'	'http://states-covid-numbers.org'
'Masks works! Check the CDC'	'Jonathan'	'Howard'	'http://florida-covid19.gov'
'Masks works! Check the CDC'	'Jonathan'	'Howard'	'https://cdc.gov'
'Masks works! Check the CDC'	'Antonio'	'Olson'	'http://states-covid-numbers.org'
'Masks works! Check the CDC'	'Antonio'	'Olson'	'http://florida-covid19.gov'
'Masks works! Check the CDC'	'Antonio'	'Olson'	'https://cdc.gov'
'Masks works! Check the CDC'	'Gina'	'Miranda'	'http://states-covid-numbers.org'
'Masks works! Check the CDC'	'Gina'	'Miranda'	'http://florida-covid19.gov'
'Masks works! Check the CDC'	'Gina'	'Miranda'	'https://cdc.gov'
'Masks works! Check the CDC'	'Courtney'	'White'	'http://covid-is-not-hoax.net'
'Masks works! Check the CDC'	'Courtney'	'White'	'http://mask-works.info'
'Masks works! Check the CDC'	'Courtney'	'White'	'https://cdc.gov'

6. Find the email addresses of all users who have submitted evidence about tweets that have either the hashtag “MaskUp” or the hashtag “COVID19”. **(Note: Hashtags are case-sensitive.)**

a) Relational Algebra

$\pi_{\text{email}} (\sigma (\text{hashtag} = \text{'COVID19'} \vee \text{hashtag} = \text{'MaskUp'}) \text{Hashtag} \bowtie \text{About} \bowtie \text{EvidenceFrom} \bowtie \text{User})$

b) Parse Tree



$\pi_{\text{email}} ( ( ( \sigma (\text{hashtag} = \text{'COVID19'} \text{ or } \text{hashtag} = \text{'MaskUp'}) \text{Hashtag} \bowtie \text{About} ) \bowtie \text{EvidenceFrom} ) \bowtie \text{User} )$

c) Result

User.email
'davis_holly86@hotmail.com'
'pau.miller@yahoo.com'
'bradshaw73051@gmail.com'

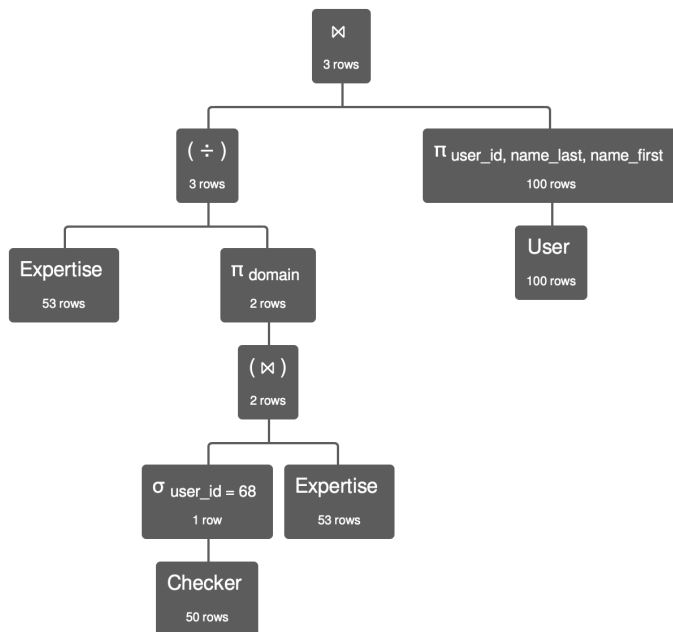


7. Find the user IDs, first names, and last names of checkers that have **all** the domains of expertise from the user with ID = 68. (Note: Your answer will include the “ID = 68” checker as well, of course.)

a) Relational Algebra

$(\text{Expertise} \div \pi_{\text{domain}} (\sigma_{\text{user\_id} = 68} \text{Checker} \bowtie \text{Expertise})) \bowtie \pi_{\text{user\_id, name\_last, name\_first}} \text{User}$

b) Parse Tree



$(\text{Expertise} \div \pi_{\text{domain}} (\sigma_{\text{user\_id} = 68} \text{Checker} \bowtie \text{Expertise})) \bowtie \pi_{\text{user\_id, name\_last, name\_first}} \text{User}$

c) Result

<b>Expertise.user_id</b>	<b>User.name_last</b>	<b>User.name_first</b>
40	'Rhodes'	'Lauren'
68	'Ortiz'	'Darren'
87	'Monroe'	'Anthony'

8. List the phone numbers of checkers who have either verified the tweet with the id “1321211561046933514” or who are experts in “Infectious Diseases”.

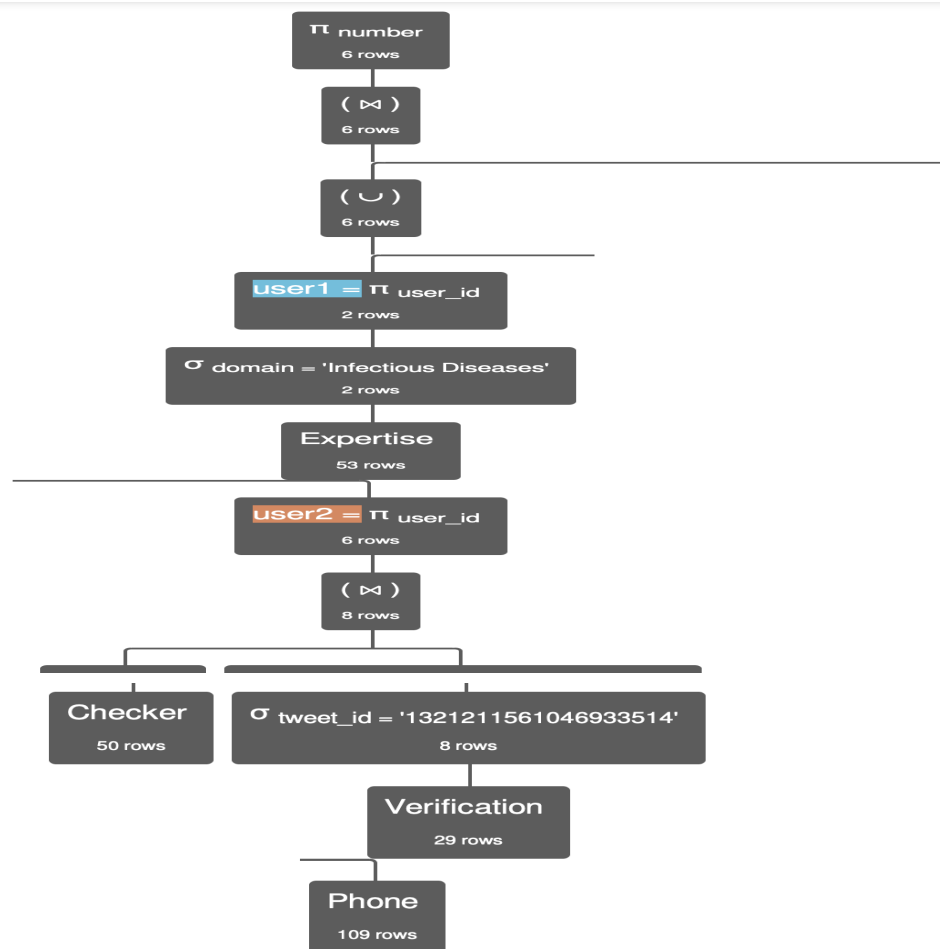
a) Relational Algebra

$user1 = \pi_{user\_id} \sigma_{domain = 'Infectious Diseases'} Expertise$

$user2 = \pi_{user\_id} ((Checker \bowtie \sigma_{tweet\_id = '1321211561046933514'} Verification))$

$\pi_{number} ((user1 \cup user2) \bowtie Phone)$

b) Parse Tree



$\pi_{number} ( ( \pi_{user\_id} \sigma_{domain = 'Infectious Diseases'} Expertise \cup \pi_{user\_id} ( Checker \bowtie \sigma_{tweet\_id = '1321211561046933514'} Verification ) ) \bowtie Phone )$

c) Result

Phone.number
'001-337-445-5627x321'
'193-407-5790x179'
'001-070-249-0204'
'524.899.8641'
'164.768.4712x8904'
'403-156-1446'