Technical Appendix
Catch the Pink Flamingo Analysis

Produced by: Pablo Langa Blanco

Acquiring, Exploring and Preparing the Data

Data Exploration

Data Set Overview

The table below lists each of the files available for analysis with a short description of what is found in each one.

File Name	**Description**	**Fields**
ad-clicks.csv	This table has accumulated all clicks of the users on an advertisement in the app	**timestamp**: the timestamp when the event occurs. **txId**: a unique id for the click, its a primary key **userSessionid**: the id of the user session, its a foreing key from the table User_sessions **teamid**: the current team id, its a foreing key from the table Team **userid**: the user id of the user who made the click, its a foreing key from the table User **adId**: the id of the ad clicked on. This id must be in other table wich have the id of the adds. **adCategory**: the category/type of ad clicked on. This attribute is an enumerated type.
buy-clicks.csv	This table have one row for each purchase in the app	**timestamp**: the timestamp when the event occurs **userSessionId**: the id of the user session for the user who made the purchase. its a foreing key from the

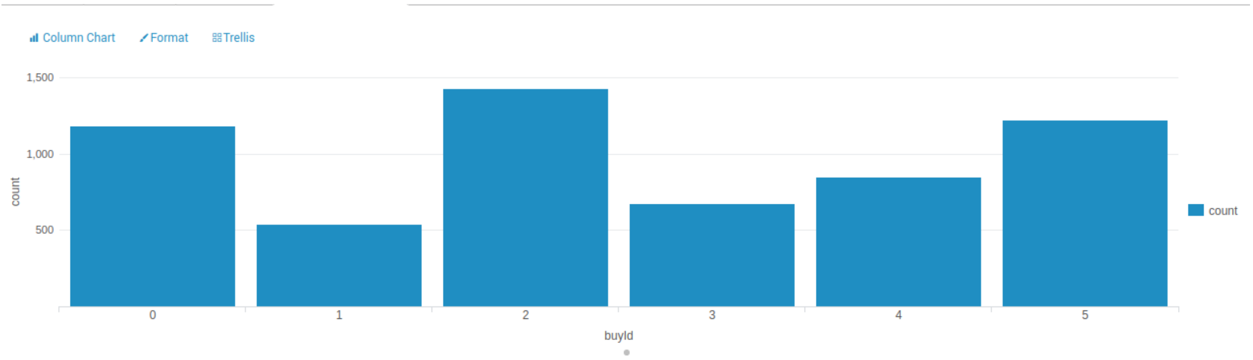
		table User_sessions **team**: the current team id of the user who made the purchase, its a foreing key from the table Team (teamId) **userId**: the user id of the user who made the purchase, its a foreing key from the table User **buyId**: the id of the item purchased, its a primary key **price**: the price of the item purchased
users.csv	This table contains the players of the game	**timestamp**: when user first played the game. **userId**: the user id assigned to the user. its a primary key. Numeric Id **nick**: the nickname chosen by the user. **twitter**: the twitter handle of the user. **dob**: the date of birth of the user. In this format AAAAMMDD **country**: the twoletter country code where the user lives.
team.csv	This table contains all the teams in the game.	**teamId**: the id of the team, its a primary key. Numeric Id **name**: the name of the team **teamCreationTime**: the timestamp when the team was created **teamEndTime**: the timestamp when the last member left the team **strength**: a measure of team strength, roughly corresponding to the success of a team **currentLevel**: the current level of the team
team-assignments.csv	Each row contains when a user join a team. When one user join a new team it indicates that leave the last team.	**timestamp**: when the user joined the team. **team**: the id of the team, foreing key **userId**: the id of the user, foreing key **assignmentId**: a unique id for this

		assignment, primary key.
level-events.csv	This table contains all the level events, when a team start an event on when finish it.	**timestamp**: when the event occurred. **eventId**: a unique id for the event, primary key **teamId**: the id of the team, foreing key **teamLevel**: the level started or completed. Its grater than 0 **eventType**: the type of event, either start or end. Enumerated value.
user-session.csv	Each line in this file describes a user session, which denotes when a user starts and stops playing the game. Additionally, when a team goes to the next level in the game, the session is ended for each user in the team and a new one started.	**timestamp**: a timestamp denoting when the event occurred. **userSessionId**: a unique id for the session. Primary key **userId**: the current user's ID. Foreing key **teamId**: the current user's team. Foreing key **assignmentId**: the team assignment id for the user to the team. Foreing key **sessionType**: whether the event is the start or end of a session. Enumerated type **teamLevel**: the level of the team during this session. Positive integer **platformType**: the type of platform of the user during this session.
game-clicks.csv	A line is added to this file each time a user performs a click in the game.	**timestamp**: when the click occurred. **clickId**: a unique id for the click. Primary key **userId**: the id of the user performing the click. Foreing key **userSessionId**: the id of the session of the user when the click is performed. Foreing key **isHit**: denotes if the click was on a flamingo (value is 1) or missed the flamingo (value is 0) **teamId**: the id of the team of the user, Foreing key **teamLevel**: the current level of the team of the user

Aggregation

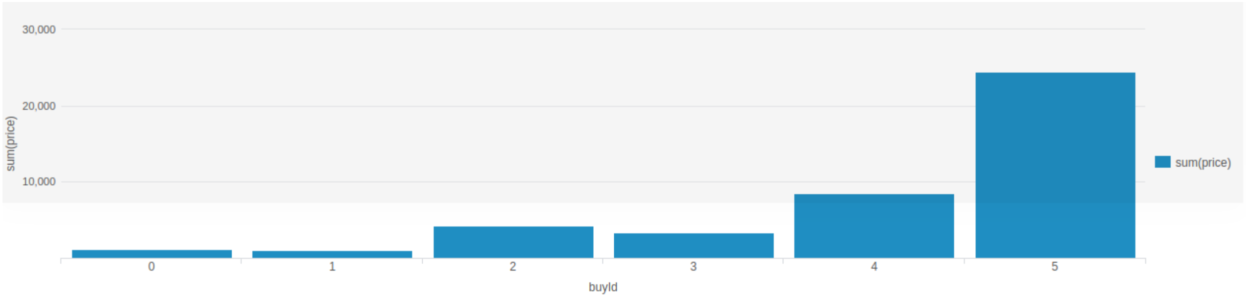
Amount spent buying items	21407
Number of unique items available to be purchased	6

- A histogram showing how many times each item is purchased:



the item 2 is the top shell item

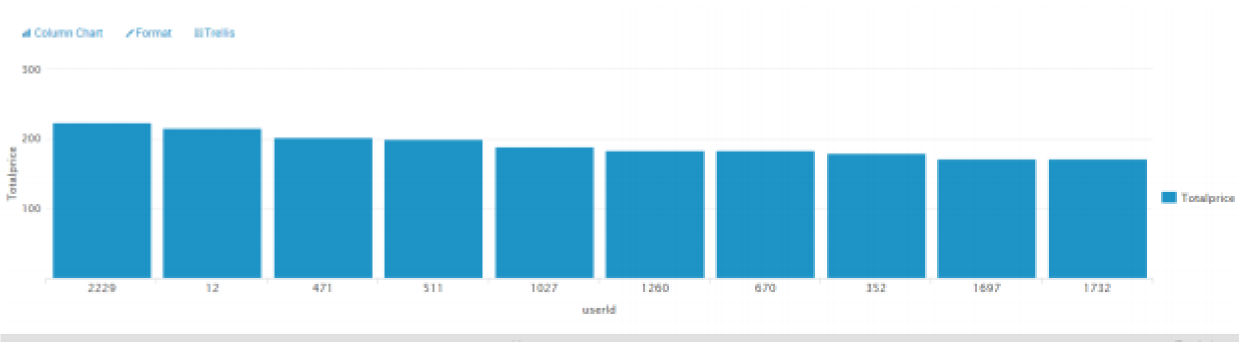
- A histogram showing how much money was made from each item:



The item 5 is the top of how much money spent the players

Filtering

A histogram showing total amount of money spent by the top ten users (ranked by how much money they spent).



The following table shows the user id, platform, and hit-ratio percentage for the top three buying users:

Rank	**User Id**	**Platform**	**Hit-Ratio (%)**
1	2229	iphone	11.59%
2	12	iphone	13.06%
3	471	iphone	14.50%

Data Classification Analysis

Data Preparation

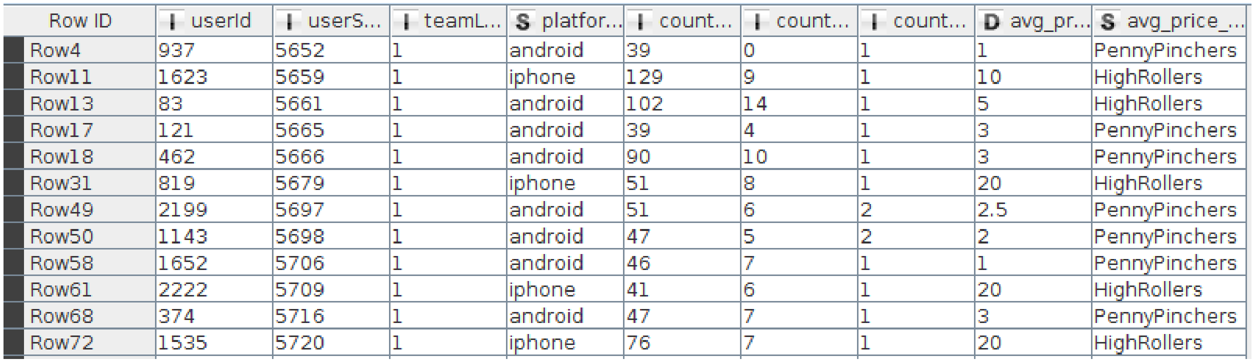
Analysis of combined_data.csv

Sample Selection

Item	Amount
# of Samples	4619
# of Samples with Purchases	1411

Attribute Creation

A new categorical attribute was created to enable analysis of players as broken into 2 categories (HighRollers and PennyPinchers). A screenshot of the attribute follows:



Row ID	I userId	I userS...	I teamL...	S platfor...	I count...	I count...	I count...	D avg_pr...	S avg_price_...
Row4	937	5652	1	android	39	0	1	1	PennyPinchers
Row11	1623	5659	1	iphone	129	9	1	10	HighRollers
Row13	83	5661	1	android	102	14	1	5	HighRollers
Row17	121	5665	1	android	39	4	1	3	PennyPinchers
Row18	462	5666	1	android	90	10	1	3	PennyPinchers
Row31	819	5679	1	iphone	51	8	1	20	HighRollers
Row49	2199	5697	1	android	51	6	2	2.5	PennyPinchers
Row50	1143	5698	1	android	47	5	2	2	PennyPinchers
Row58	1652	5706	1	android	46	7	1	1	PennyPinchers
Row61	2222	5709	1	iphone	41	6	1	20	HighRollers
Row68	374	5716	1	android	47	7	1	3	PennyPinchers
Row72	1535	5720	1	iphone	76	7	1	20	HighRollers

The new attribute avg_price_binned uses the avg_price attribute to classify the instances. When the value of avg_prive is less than 5 it classify the instance as "PennyPinchers". And when it is grater than 5 as "HighRollers"

The creation of this new categorical attribute was necessary because it will be the target attribute that we are going to use in the next steps to train the decision tree.

Attribute Selection

The following attributes were filtered from the dataset for the following reasons:

Attribute	Rationale for Filtering
userSesionId	Its excluded because its not a significance value, its only the Id to identify the session

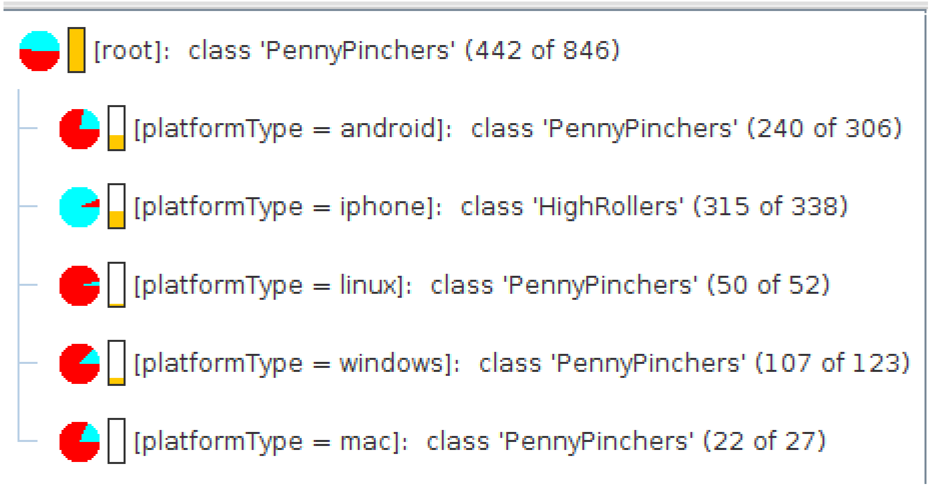
avg_price	Its excluded because the target attribute its created from this attribute

Data Partitioning and Modeling

The data was partitioned into train and test datasets.
The first partition data set was used to create the decision tree model.
The trained model was then applied to the second partition dataset.
This is important because we need to test our model in data set different from the training data set to see how is it behaving in different data with the same distribution.

When partitioning the data using sampling, it is important to set the random seed because we want to be able to reproduce the same results in each execution.

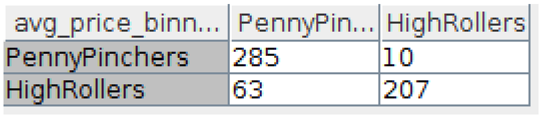
A screen-shot of the resulting decision tree can be seen below:



```
[root]:  class 'PennyPinchers' (442 of 846)
    [platformType = android]:  class 'PennyPinchers' (240 of 306)
    [platformType = iphone]:  class 'HighRollers' (315 of 338)
    [platformType = linux]:  class 'PennyPinchers' (50 of 52)
    [platformType = windows]:  class 'PennyPinchers' (107 of 123)
    [platformType = mac]:  class 'PennyPinchers' (22 of 27)
```

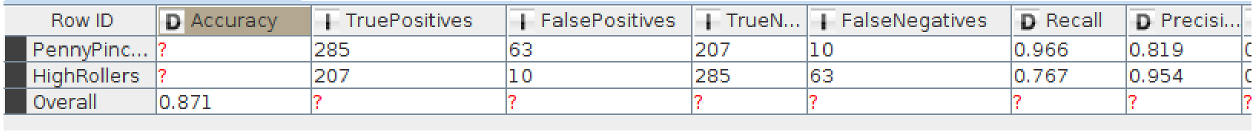
Evaluation

A screenshot of the confusion matrix can be seen below:



avg_price_binn...	PennyPin...	HighRollers
PennyPinchers	285	10
HighRollers	63	207

As seen in the screenshot above, the overall accuracy of the model is 0.871

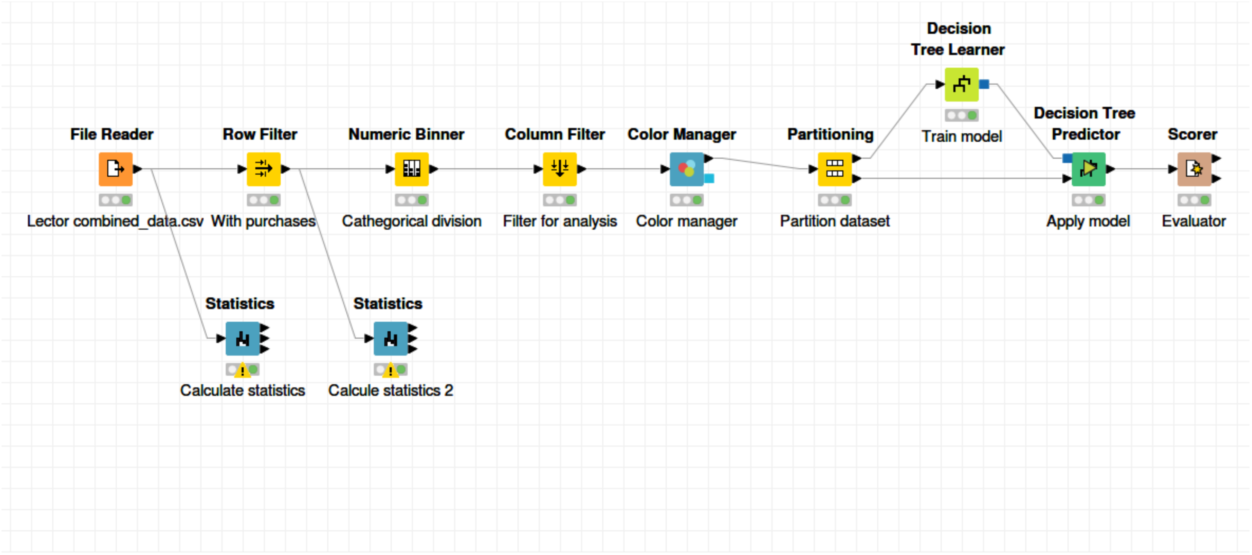


Row ID	D Accuracy	I TruePositives	I FalsePositives	I TrueN...	I FalseNegatives	D Recall	D Precisi...
PennyPinc...	?	285	63	207	10	0.966	0.819
HighRollers	?	207	10	285	63	0.767	0.954
Overall	0.871	?	?	?	?	?	?

- 285 “PennyPinchers” was correctly predicted
- 207 “HighRollers” was correctly predicted.
- 63 instances was predicted as “PennyPinchers” and they are “HighRollers”
- 10 instances was predicted as “HighRollers” and they are “PennyPinchers”

Analysis Conclusions

The final KNIME workflow is shown below:



What makes a HighRoller vs. a PennyPincher?
Based in the decision results the principal attribute is the platform type. The iphone user trend to be high rollers and the users of the oder platmorms trend to be Penny Pinchers.

Specific Recommendations to Increase Revenue
1. We need to focus our efforts to increase iphone users. We could make publicity of our game oriented in this platform.
2. We want to the no iphone platforms users spent more money in our game. We could personalize some characteristics of our game, like an andoid flamingo to increase the interests of this platform users.

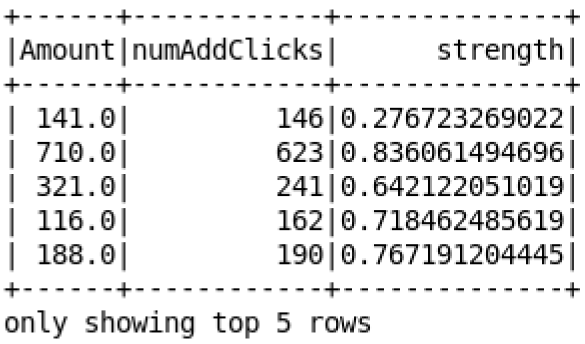
Clustering Analysis

Attribute Selection

Attribute	Rationale for Selection
Strength	The Strength of the team of team.csv. I want to know how this attribute is related with the purchases
Amount	This attribute is the sum of the price spent by a team. Its interesting because is the parameter we want to increase or we want to know how its related with other parameters of the team (from buy-clicks.csv)
NumAddClicks	This attribute is the count of the click in the adds by the teams (from ad-clicks.csv). Its interesting because it could be related with the amount spent

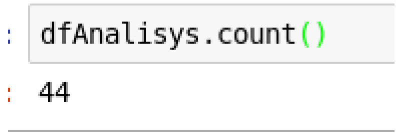
Training Data Set Creation

The training data set used for this analysis is shown below (first 5 lines):



```
+------+------------+--------------+
|Amount|numAddClicks|      strength|
+------+------------+--------------+
| 141.0|         146|0.276723269022|
| 710.0|         623|0.836061494696|
| 321.0|         241|0.642122051019|
| 116.0|         162|0.718462485619|
| 188.0|         190|0.767191204445|
+------+------------+--------------+
only showing top 5 rows
```

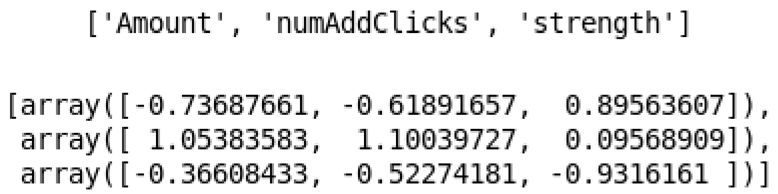
Dimensions of the training data set (rows x columns) : 44 x 3



```
: dfAnalisys.count()
: 44
```

# of clusters created: 3
I'm going to try with 3 clusters.

Cluster Centers



```
['Amount', 'numAddClicks', 'strength']

[array([-0.73687661, -0.61891657,  0.89563607]),
 array([ 1.05383583,  1.10039727,  0.09568909]),
 array([-0.36608433, -0.52274181, -0.9316161 ])]
```

Cluster #	Cluster Center
Cluster 1	[-0.73, -0.61, 0.89]
Cluster 2	[1.05, 1.10, 0.09]
Cluster 3	[-0.36, -0.52, -0.93]

These clusters can be differentiated from each other as follows:

Cluster 1 is different from the others in that… Has a high strength, very low number of addClicks an Amount spent

Cluster 2 is different from the others in that… Has a medium strength, very hihg number of addClicks an Amount spent

Cluster 3 is different from the others in that… Has a low strength, low number of addClicks an Amount spent

Recommended Actions

Action Recommended	Rationale for the action
Incentive team with medium strength	We have seen that the teams with medium strength are the team that have spent more money and clicked more adds, its important to incentive this teams to play
Put adds in strategic zone	We have seen than the amount spent is directly related with the addClicks. We could put the adds in strategic zones of the game to encourage the team to click it.

Graph Analytics Analysis

Modeling Chat Data using a Graph Data Model

Our principal nodes are Users, Teams, Team Chat Sessions and Chat items. A User could create new chat. All the chats are part of a team chat session. A Chat item could respond to an other chat item or maybe Mention to a user. All users could join or leaves a chat session. Finally all chat sessions are owned by a Team.

Creation of the Graph Database for Chats

Describe the steps you took for creating the graph database. As part of these steps

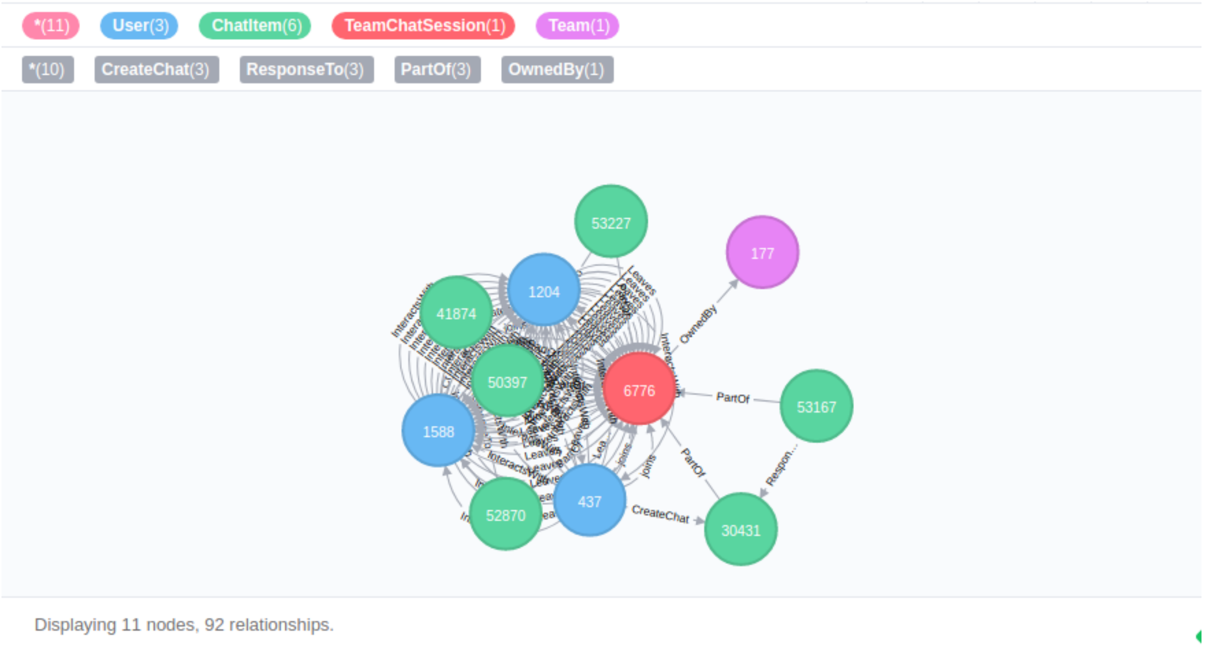
- Write the schema of the 6 CSV files

chat_create_team_chat.csv
userid,teamid,TeamChatSessionID,timestamp
chat_item_team_chat.csv
userid,teamchatsessionid,chatitemid,timestamp
chat_join_team_chat
userid,TeamChatSessionID,teamstamp
chat_leave_team_chat.csv
userid,teamchatsessionid,timestamp
chat_mention_team_chat.csv
ChatItem,userid,timeStamp
chat_respond_team_chat.csv
chatid1,chatid2,timestamp

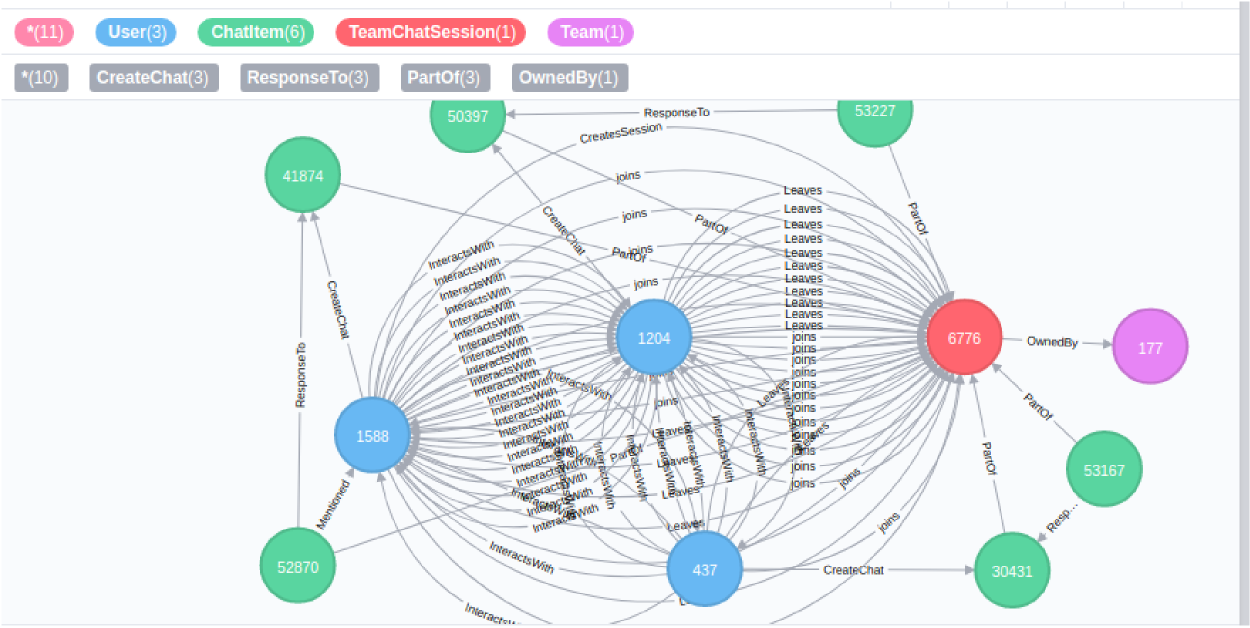
- Explain the loading process and include a sample LOAD command

```
    LOAD CSV FROM "file:///chat_create_team_chat.csv" as row
MERGE (u:User {id: toInteger(row[0])})
MERGE (t:Team {id: toInteger(row[1])})
MERGE (c:TeamChatSession {id: toInteger(row[2])})
MERGE (u)-[:CreatesSession{timeStamp: row[3]}]->(c)
MERGE (c)-[:OwnedBy{timeStamp: row[3]}]->(t)

LOAD CSV FROM "file:///chat_join_team_chat.csv" as row
MERGE (u:User {id: toInteger(row[0])})
MERGE (c:TeamChatSession {id: toInteger(row[1])})
MERGE (u)-[:joins{timeStamp: row[2]}]->(c)


LOAD CSV FROM "file:///chat_leave_team_chat.csv" as row
MERGE (u:User {id: toInteger(row[0])})
MERGE (c:TeamChatSession {id: toInteger(row[1])})
MERGE (u)-[:Leaves{timeStamp: row[2]}]->(c)

LOAD CSV FROM "file:///chat_mention_team_chat.csv" as row
MERGE (u:User {id: toInteger(row[1])})
MERGE (ch:ChatItem {id: toInteger(row[0])})
MERGE (ch)-[:Mentioned{timeStamp: row[2]}]->(u)
LOAD CSV FROM "file:///chat_respond_team_chat.csv" as row
MERGE (ch1:ChatItem {id: toInteger(row[0])})
MERGE (ch2:ChatItem {id: toInteger(row[1])})
    MERGE (ch1)-[:ResponseTo{timeStamp: row[2]}]→(ch2)
```

- Present a screenshot of some part of the graph you have generated. The graphs must include clearly visible examples of most node and edge types. Below are two acceptable examples. The first example is a rendered in the default Neo4j distribution, the second has had some nodes moved to expose the edges more clearly. Both include examples of most node and edge types.



Displaying 11 nodes, 92 relationships.

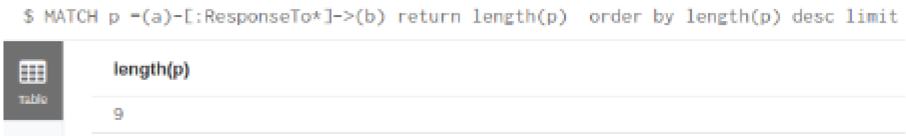


Finding the longest conversation chain and its participants

Report the results including the length of the conversation (path length) and how many unique users were part of the conversation chain. Describe your steps. Write the query that produces the correct answer.

Length of the conversation: 9

MATCH p =(a)-[:ResponseTo*]->(b)
return length(p)
order by length(p) desc
limit 1

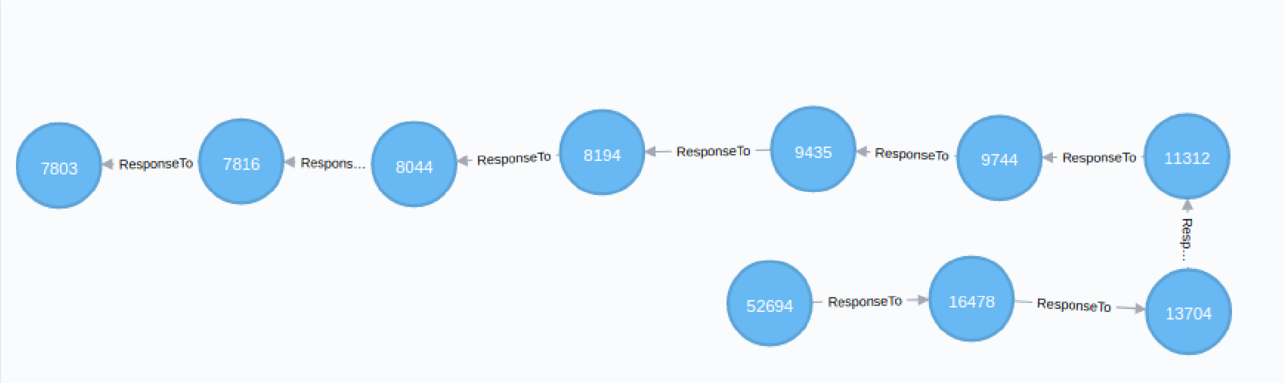


```
$ MATCH p =(a)-[:ResponseTo*]->(b) return length(p)  order by length(p) desc limit
```

length(p)
9

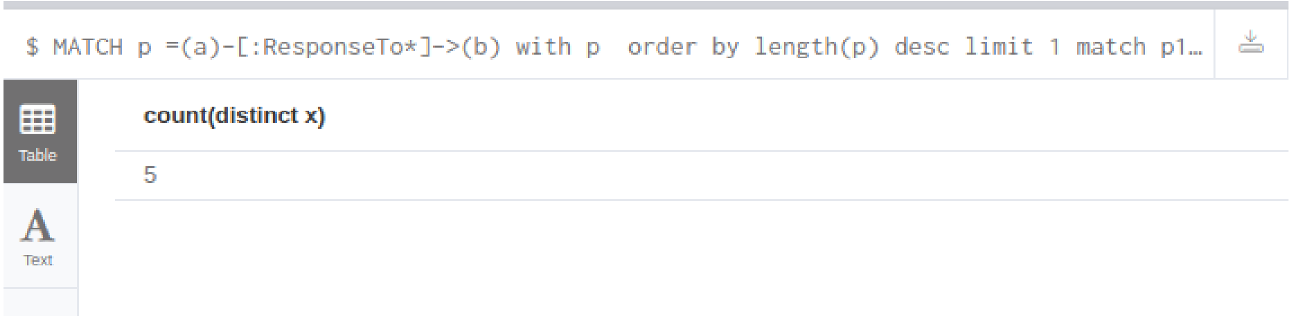
Conversation

MATCH p =(a)-[:ResponseTo*]->(b)
return p
order by length(p) desc
limit 1



Distinct Users: 5

```
MATCH p =(a)-[:ResponseTo*]->(b)
with p
order by length(p) desc limit 1
match p1=(x)-[:CreateChat]->(y)
where y in nodes(p)
return count(distinct x)
```



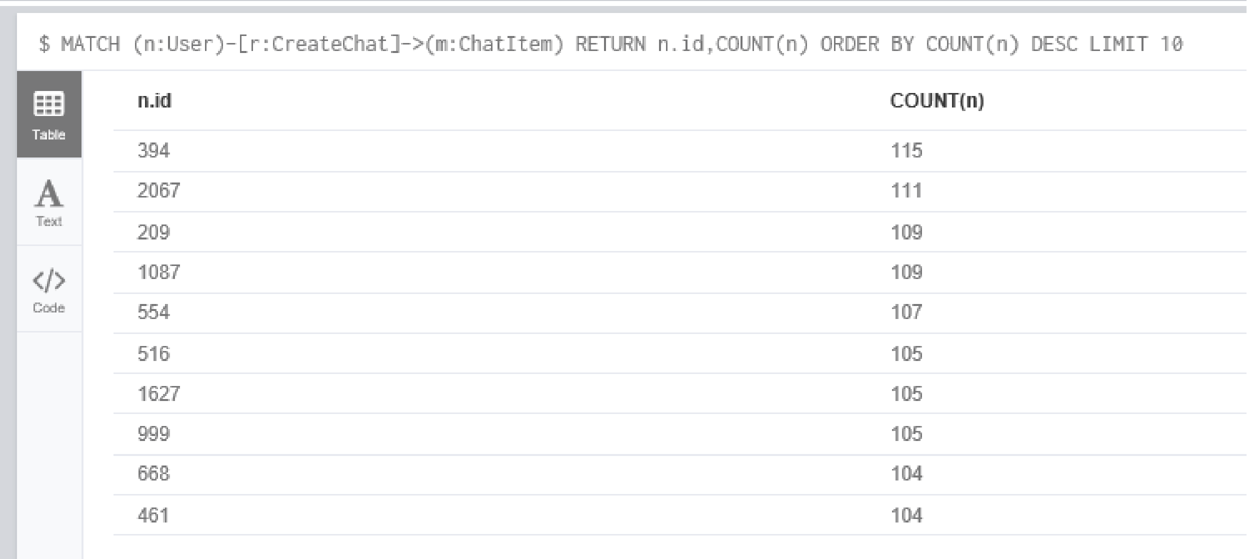
```
$ MATCH p =(a)-[:ResponseTo*]->(b) with p  order by length(p) desc limit 1 match p1…
```

count(distinct x)
5

Analyzing the relationship between top 10 chattiest users and top 10 chattiest teams

Describe your steps from Question 2. In the process, create the following two tables. You only need to include the top 3 for each table. Identify and report whether any of the chattiest users were part of any of the chattiest teams.

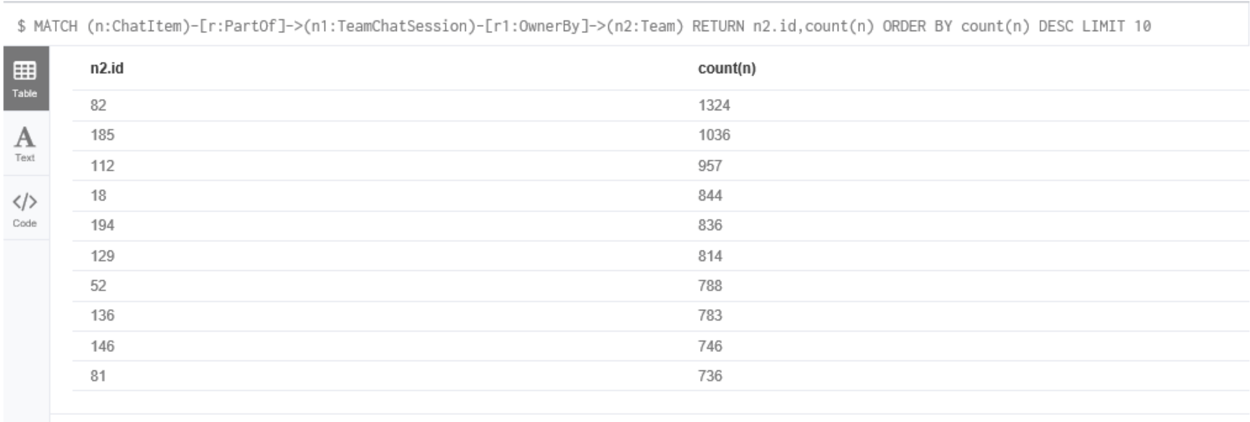
Chattiest User



```
$ MATCH (n:User)-[r:CreateChat]->(m:ChatItem) RETURN n.id,COUNT(n) ORDER BY COUNT(n) DESC LIMIT 10
```

n.id	COUNT(n)
394	115
2067	111
209	109
1087	109
554	107
516	105
1627	105
999	105
668	104
461	104

Chattiest Team



```
$ MATCH (n:ChatItem)-[r:PartOf]->(n1:TeamChatSession)-[r1:OwnerBy]->(n2:Team) RETURN n2.id,count(n) ORDER BY count(n) DESC LIMIT 10
```

n2.id	count(n)
82	1324
185	1036
112	957
18	844
194	836
129	814
52	788
136	783
146	746
81	736

Chattiest Users

Users	**Number of Chats**
394	115
2067	111
209	109
1087	109
554	107
516	105
1627	105

999	105
668	104
461	104

Chattiest Teams

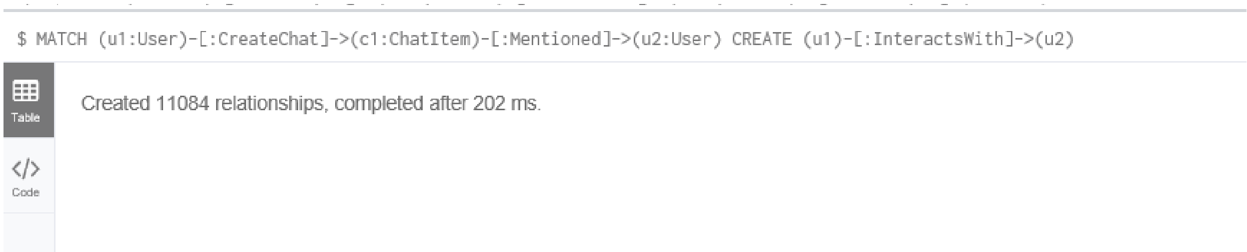
Teams	Number of Chats
82	1324
185	1036
112	957
18	844
194	836
129	814
52	788
136	783
146	746
81	736

Finally, present your answer, i.e. whether or not any of the chattiest users are part of any of the chattiest teams.
The Chattiest User id 999 is part of Team id 52 which is also among top 10 Chattiest Teams.

How Active Are Groups of Users?

Describe your steps for performing this analysis. Be as clear, concise, and as brief as possible. Finally, report the top 3 most active users in the table below.

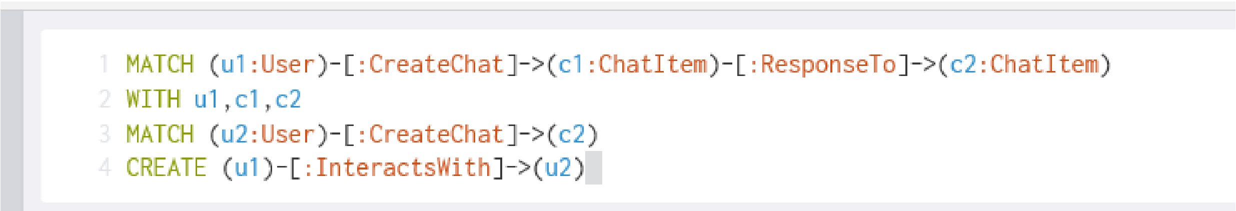
1. Created InteractWith Edge between the users based on Mentioned Edge



```
$ MATCH (u1:User)-[:CreateChat]->(c1:ChatItem)-[:Mentioned]->(u2:User) CREATE (u1)-[:InteractsWith]->(u2)
```

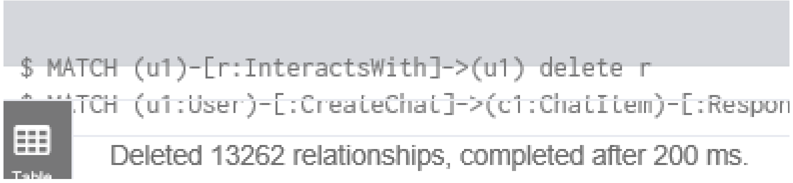
Created 11084 relationships, completed after 202 ms.

2.Created InteractWith Edge between the users based on ResponseTo Edge



```
MATCH (u1:User)-[:CreateChat]->(c1:ChatItem)-[:ResponseTo]->(c2:ChatItem)
WITH u1,c1,c2
MATCH (u2:User)-[:CreateChat]->(c2)
CREATE (u1)-[:InteractsWith]->(u2)
```

3. Delete the Self Loops Edges of InteractWith



```
$ MATCH (u1)-[r:InteractsWith]->(u1) delete r
$ MATCH (u1:User)-[:CreateChat]->(c1:ChatItem)-[:Respon
```

Deleted 13262 relationships, completed after 200 ms.

Most Active Users (based on Cluster Coefficients)

User ID	Coefficient
209	0.95
554	0.90
1087	0.80

Recommended Actions

Finally, make recommendations to Eglence, Inc. and include examples of how your findings support them. Include this information in Slide 6 of your final presentation.