

Privacy and Security in Online Social Media

Course on NPTEL

NOC21-CS28

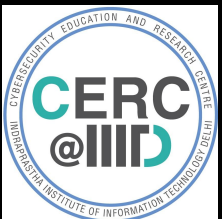
Week 2.1

Ponnurangam Kumaraguru (“PK”)

Full Professor

ACM Distinguished Speaker

fb/ponnurangam.kumaraguru, @ponguru



Online discussion

- <https://groups.google.com/a/nptel.iitm.ac.in/g/noc21-cs28-discuss>
- Please participate
- Read the posts before asking any questions

Assignment 1

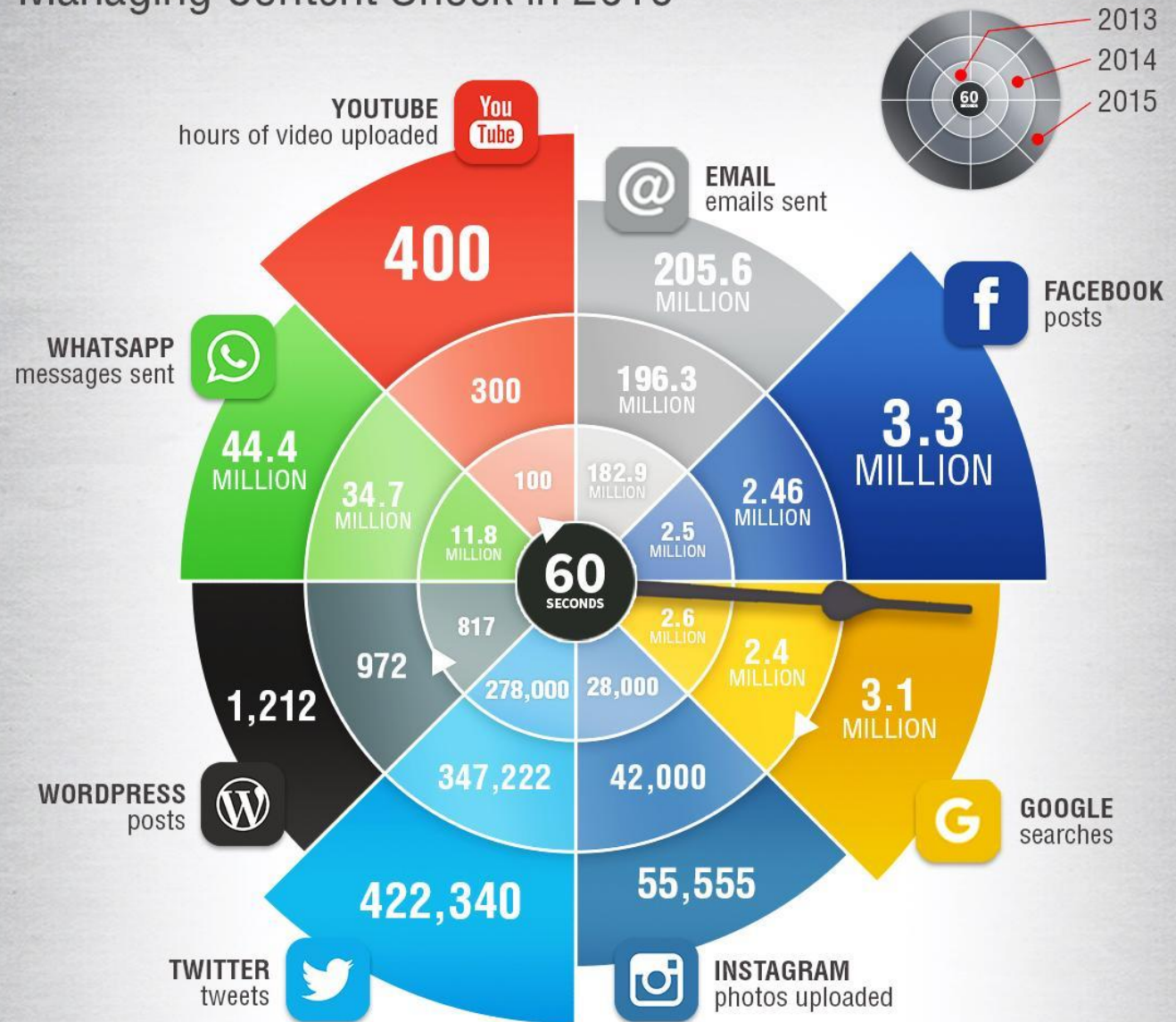
- Hope it was simple and all of you were able to do it satisfactorily





What Happens Online in 60 Seconds?

Managing Content Shock in 2016



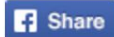
4 / 5 V's of Online Social Media



Home > India > India-News-India > Social media turns a blessing: Avani's missing photo goes viral, and she gets her mother back in 20 minutes

Social media turns a blessing: Avani's missing photo goes viral, and she gets her mother back in 20 minutes

The photo showed the girl holding a lady railway police official's hand.



559



28



1

Written by **Nisha Nambiar** | Pune | Updated: May 17, 2016 2:04 am



Irresistible fares
Book now

Take off now

Find tickets >

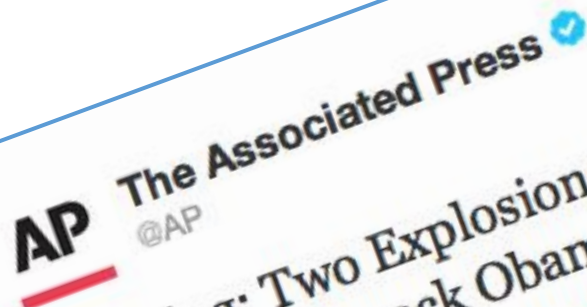
KLM
Journeys of inspiration

BEST OF EXPRESS



India

Pakistan covets
territory of others and
uses terrorism as state



Breaking: Two Explosions in the White House and Barack Obama is injured



3,063
RETWEETS

144
FAVORITES



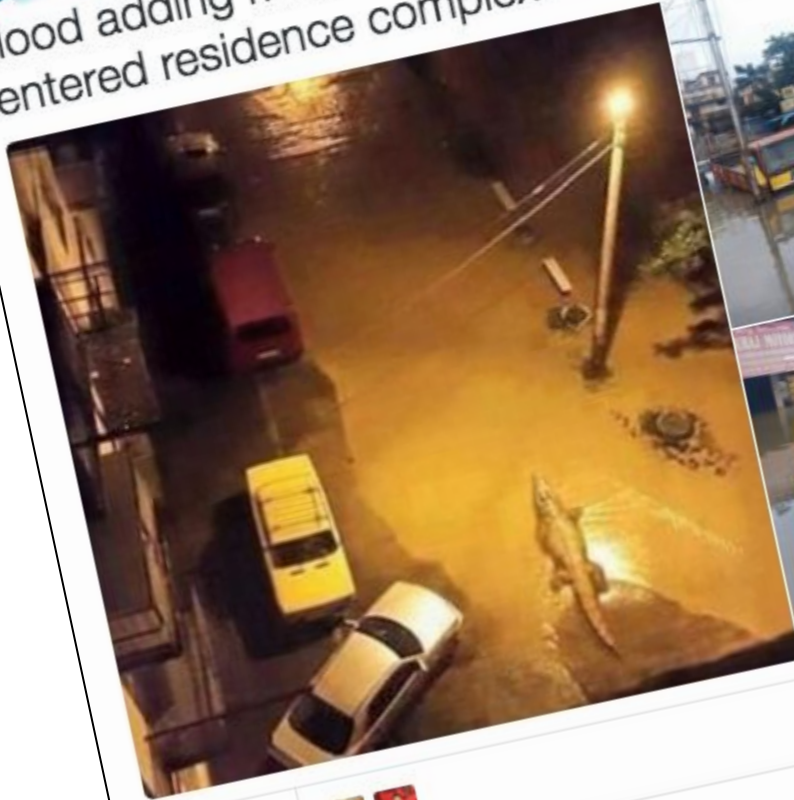
12:07 PM - 23 Apr 13



Follow



@ [redacted] : Chennai going down in horrible flood adding more to trouble crocodile has entered residence complex.



RETWEETS
2



1:28 PM - 4 Dec 2015



Other implications

Using too much Facebook and Twitter may cost you your job

Sun, Oct 10 10:50 AM

London, Oct 10 (ANI): Government and council employees in the UK using social networking sites at work or posting inappropriate comments from home are likely to face serious action, which includes being sacked.

Police censured eight employees last year after they found using police computers to chat with their friends online.

Morga

Morgan Stan
A.C.E. Fun

Tutorials online

- Linux
- Python



Frameworks / Platforms to know

- APIs of OSM (e.g. Facebook / Twitter API)
- A programming language to write code to extract data (e.g. Python / RoR)
- A database to store data (e.g. MySQL / MongoDB)
- A visualization tool to query and analyze data (e.g. PhpMyAdmin / RoboMongo)

Programming language:



- High level programming language to instruct commands and facilitate data collection
- Supports libraries for reading urls, parse data, interact with APIs, etc.

```
# Start with importing some useful libraries
import json                # Library for handling the JSON response
import urllib              # Library for handling URLs

# API base URL
endpoint = "https://graph.facebook.com/search?"

i=0
while i<10:
    print result_formatted["data"]
    i=i+1
```

Application Programming Interface (API)

- OSM API enables developers to interact with the OSM website programmatically
- We use APIs to extract data from Twitter, Facebook, etc.
- Rate limit: How much data requests can we make?
- Each OSM has its own API and API rate limits



Data format

- API returns data in the following two formats:
 - JSON
 - XML

JSON

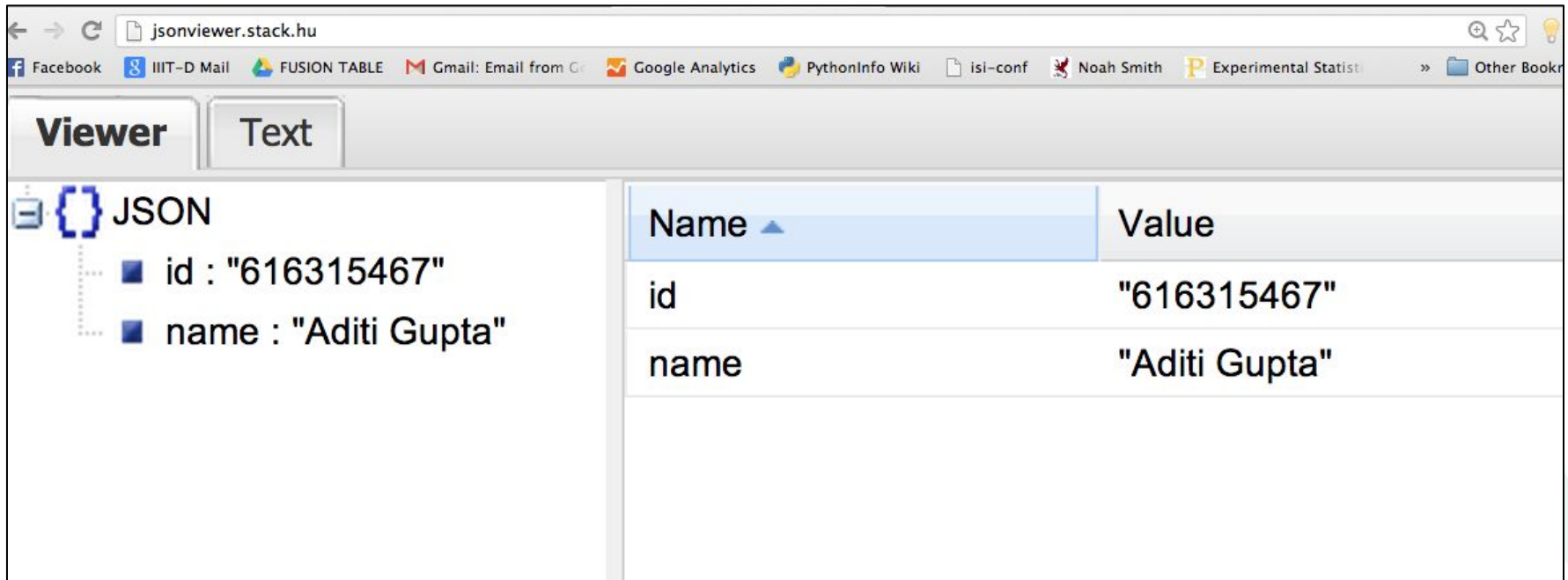
- JSON - JavaScript Object Notation
- Data structuring notation
- Sample:

The screenshot displays the Graph API Explorer interface. At the top, it says "Graph API Explorer" with a breadcrumb "Home > Tools > Graph API Explorer". On the right, there are dropdowns for "Application: [?] Graph API Explorer" and "Locale: [?] English (US)". Below this, an "Access Token" field contains a long alphanumeric string, followed by a "Debug" button and a "Get Access Token" button. The main section is titled "Graph API" and "FQL Query". It shows a "GET" method selected and a query path "/616315467?fields=id,name". A "Submit" button is to the right. Below the query, there is a link "Learn more about the Graph API syntax." On the left, under "Node: 616315467", there are checkboxes for "id" and "name", both of which are checked. On the right, the JSON response is displayed:

```
{  "id": "616315467",  "name": "Aditi Gupta"}
```

JSON

● Viewing



The screenshot shows a web browser window with the address bar displaying 'jsonviewer.stack.hu'. The browser's toolbar includes various icons and links such as Facebook, IIIT-D Mail, FUSION TABLE, Gmail, Google Analytics, PythonInfo Wiki, isi-conf, Noah Smith, Experimental Statistics, and Other Books. The page content is divided into two main sections: a 'Viewer' tab (active) and a 'Text' tab. The 'Viewer' tab displays a JSON object in a tree view on the left and a corresponding table on the right. The JSON object is:

```
{  "id": "616315467",  "name": "Aditi Gupta"}
```

The table on the right has two columns: 'Name' and 'Value'. It lists the key-value pairs from the JSON object:

| Name | Value |
|------|---------------|
| id | "616315467" |
| name | "Aditi Gupta" |

- Relational Database to store data
- Data is stored in rows and columns
- Retrieve using SQL queries
- Sample:

```
mysql> select user_id , user_screen_name from users_data_boston2 limit 2;
+-----+-----+
| user_id | user_screen_name |
+-----+-----+
| 605747286 | MacieBliss13 |
| 294673452 | I_like_HotCake |
+-----+-----+
2 rows in set (0.05 sec)

mysql> 
```

MongoDB

```
> db.tweets.find({}, {text: 1, 'user.screen_name': 1}).limit(5).pretty()
{
  "_id" : ObjectId("54d746bd582d7f6f22e0909a"),
  "text" : "@concroc @BlrCityPolice @CPBlr will attend to this",
  "user" : {
    "screen_name" : "dcptrwestbcp"
  }
}

{
  "_id" : ObjectId("54d746bd582d7f6f22e0909b"),
  "text" : "@dcptrwestbcp @CPBlr @addlcptrtraffic pls let me clarify to all.Rockline Venkatesh was our chief guest who wished the children with flowers.",
  "user" : {
    "screen_name" : "dcptrwestbcp"
  }
}

{
  "_id" : ObjectId("54d746bd582d7f6f22e0909c"),
  "text" : "Road safety week: Handbills on Traffic awareness given to the public at Yeshwanthpur by PI Swaminathan http://t.co/eSkG8kjeVS",
  "user" : {
    "screen_name" : "dcptrwestbcp"
  }
}

{
  "_id" : ObjectId("54d746bd582d7f6f22e0909d"),
  "text" : "Road safety week:Bike rally for Traffic awareness at Jayanagar today. http://t.co/oA3zhmcdmN",
  "user" : {
    "screen_name" : "dcptrwestbcp"
  }
}

{
  "_id" : ObjectId("54d746bd582d7f6f22e0909e"),
  "text" : "Road Safty Week:Addl commnr of Police Sri B Dayanand with children at a painting competition at Cubbon park on Sunday http://t.co/khMlhvzxc",
  "user" : {
    "screen_name" : "dcptrwestbcp"
  }
}
>
```

PhpMyAdmin

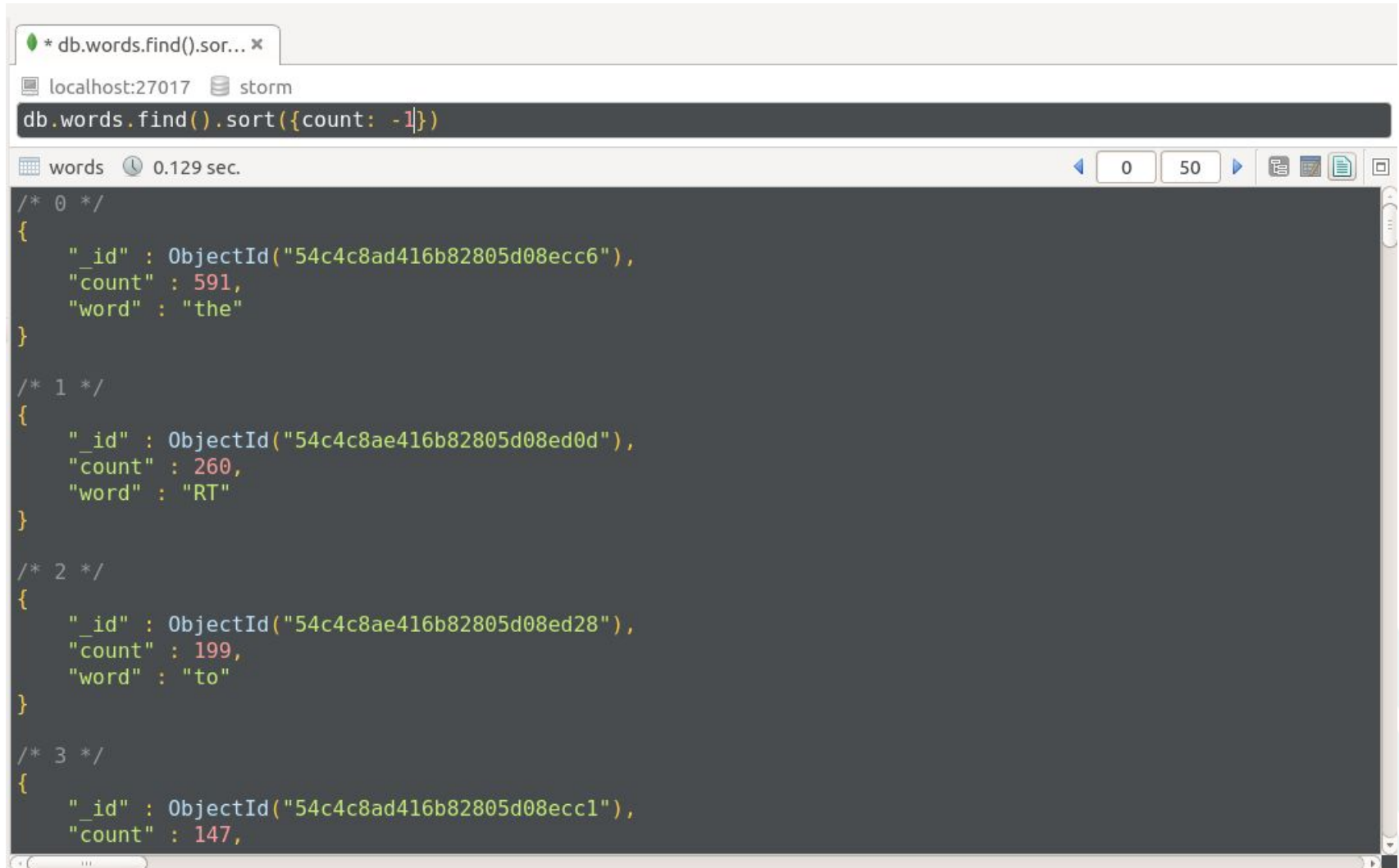
● Access MySQL databases and query using browser

SQL to query databases

The screenshot shows the PhpMyAdmin interface in a web browser. The address bar displays the URL: `precog.iitd.edu.in/phpmyadmin/index.php?db=pj_whoswho&token=b88e8a8d39b49d9e44bc64389ed7c9ed`. The interface is for the `localhost` server, database `pj_whoswho`, and table `foursquare_user_info`. The `SQL` tab is selected, showing a query: `SELECT * FROM `foursquare_user_info` LIMIT 0, 30`. The query executed successfully, showing rows 0 - 29 (~6,288¹ total, Query took 0.0001 sec). The results are displayed in a table with 12 columns: `user_id`, `fname`, `lname`, `friends`, `gender`, `twitter`, `facebook`, `badges`, `following`, `photos`, and `checkin`. The table shows 4 rows of data.

| | <code>user_id</code> | <code>fname</code> | <code>lname</code> | <code>friends</code> | <code>gender</code> | <code>twitter</code> | <code>facebook</code> | <code>badges</code> | <code>following</code> | <code>photos</code> | <code>checkin</code> |
|--------------------------|----------------------|--------------------|--------------------|----------------------|---------------------|---------------------------|-----------------------|---------------------|------------------------|---------------------|----------------------|
| <input type="checkbox"/> | 1000388 | Toni | B. | 11 | female | <code>NULL</code> | 1321163198 | 21 | 8 | 2 | 41 |
| <input type="checkbox"/> | 10005512 | Tamara | O. | 188 | female | <code>tamarammm</code> | 541738782 | 230 | 576 | 73 | 504 |
| <input type="checkbox"/> | 1000980 | Marlyn Guzman | <code>NULL</code> | 522 | female | <code>marlynguzman</code> | 515328219 | 162 | 761 | 65 | 283 |
| <input type="checkbox"/> | 1001149 | Alejandro | M. | 236 | male | <code>imixe</code> | 517994607 | 18 | 4 | 2 | 31 |

RoboMongo



The screenshot shows the RoboMongo web interface. At the top, there's a tab labeled `* db.words.find().sor... *`. Below it, the connection details are `localhost:27017` and `storm`. The command entered in the query bar is `db.words.find().sort({count: -1})`. The results are displayed for the `words` collection, taking `0.129 sec.` to execute. The results are shown in a list with 4 items, each containing an `_id`, `count`, and `word` field. The counts are highlighted in red in the original image.

```
/* 0 */
{
  "_id" : ObjectId("54c4c8ad416b82805d08ecc6"),
  "count" : 591,
  "word" : "the"
}

/* 1 */
{
  "_id" : ObjectId("54c4c8ae416b82805d08ed0d"),
  "count" : 260,
  "word" : "RT"
}

/* 2 */
{
  "_id" : ObjectId("54c4c8ae416b82805d08ed28"),
  "count" : 199,
  "word" : "to"
}

/* 3 */
{
  "_id" : ObjectId("54c4c8ad416b82805d08ecc1"),
  "count" : 147,
```


RoboMongo

RoboMongo interface showing a query execution result for the `db.network_graph_analytics` collection.

Query: `db.network_graph_analytics.find()`

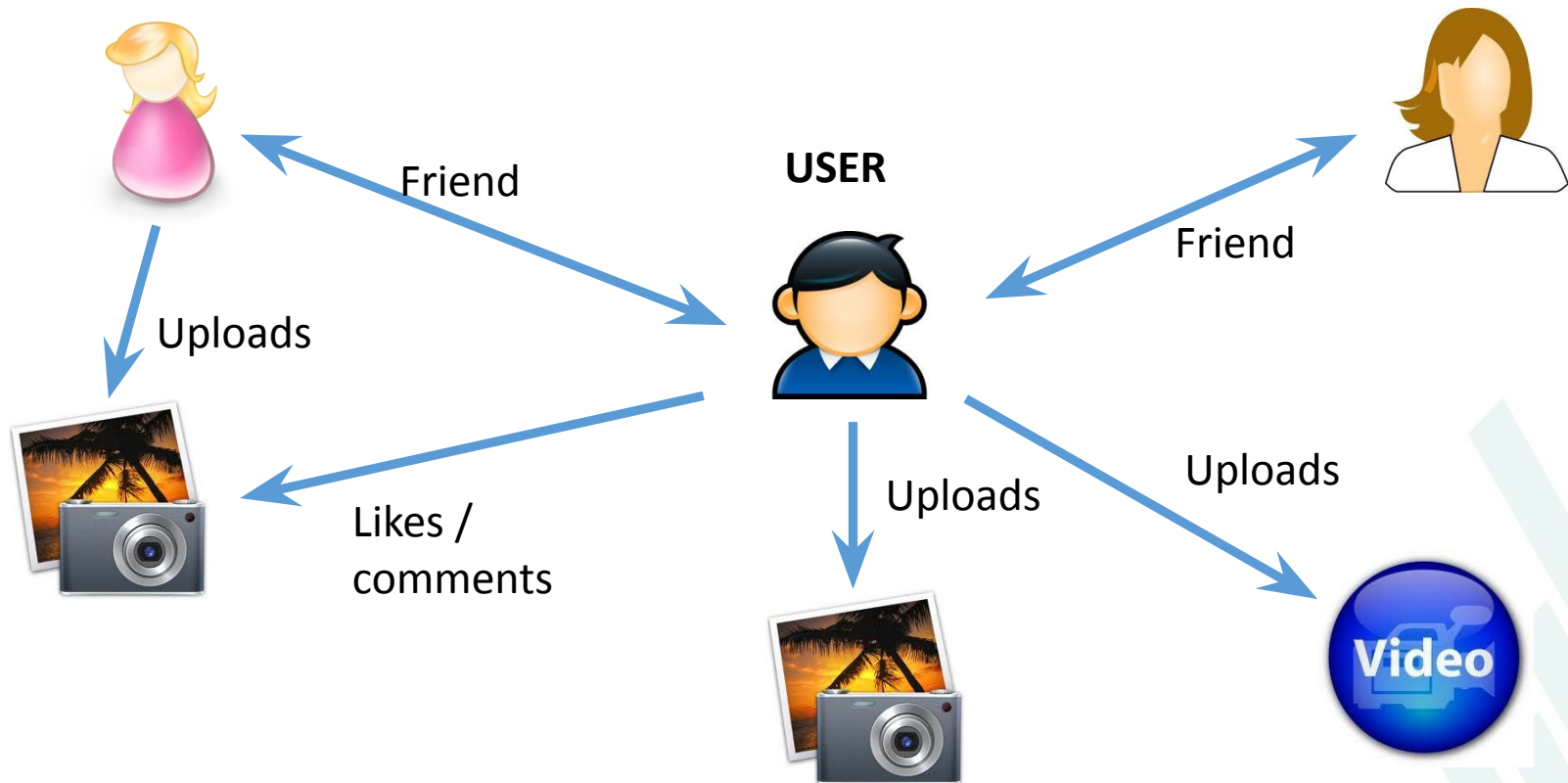
Execution time: 0.067 sec.

| Key | Value | Type |
|--|--------------------------------------|----------|
| (1) ObjectId("549e8adc786f72665701000...") | { 8 fields } | Object |
| _id | ObjectId("549e8adc786f726657010000") | ObjectId |
| first_node_seen_at | 2014-11-22 17:36:12.000Z | Date |
| job_id | 9d2e6c63681aa8fc00a09d71 | String |
| keyword_id | ObjectId("549e8aca786f7265c6000000") | ObjectId |
| last_index | 735 | Int32 |
| last_node_seen_at | 2014-12-11 17:00:39.000Z | Date |
| network_graph_links | Array [820] | Array |
| network_graph_nodes | Array [736] | Array |
| 0 | { 8 fields } | Object |
| _id | ObjectId("549e8b18786f726657020000") | ObjectId |
| first_seen_at | 2014-11-22 17:36:12.000Z | Date |
| in_degree | 194 | Int32 |
| index | 0 | Int32 |
| last_seen_at | 2014-12-11 17:00:39.000Z | Date |
| out_degree | 2233 | Int32 |
| screen_name | ShamiWitness | String |
| user_id | 57914577 | Int32 |
| 1 | { 8 fields } | Object |
| 2 | { 8 fields } | Object |
| 3 | { 8 fields } | Object |
| 4 | { 8 fields } | Object |
| 5 | { 8 fields } | Object |
| 6 | { 8 fields } | Object |
| 7 | { 8 fields } | Object |
| 8 | { 8 fields } | Object |

All content in graph form

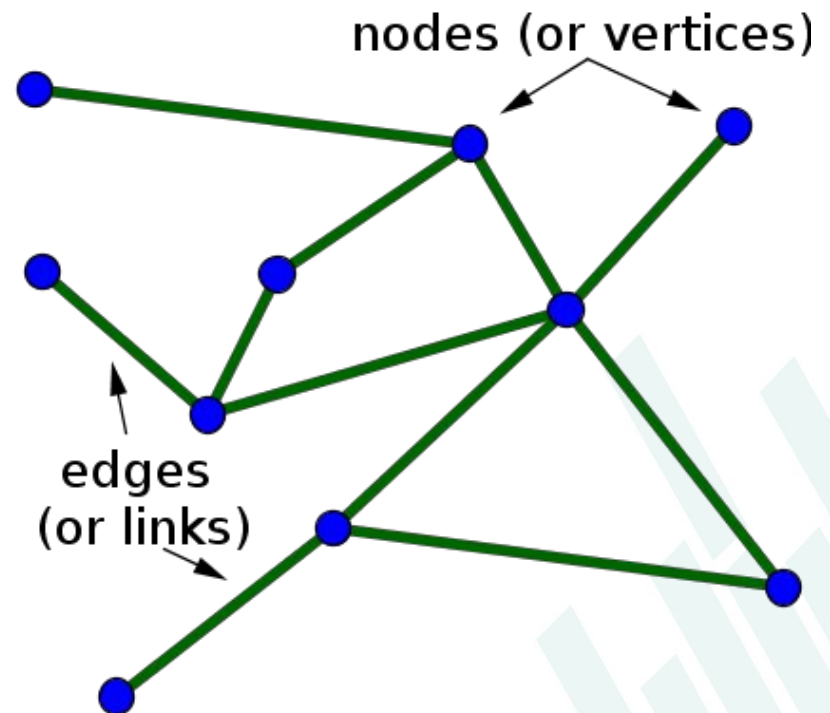
● Graph API

- Interface to extract data related to User profiles, activities, photos, pages, applications, etc.



Why is it called the Graph API

- All objects are stored as nodes of a “graph”
- Connections (likes, friendship etc.) are edges
- All nodes have a unique numeric ID
 - Users
 - Pages
 - Posts
 - ...



Tutorials for this week

● ~~Facebook API~~ Reddit API

Thank you

pk@iiitd.ac.in

precog.iiitd.edu.in

fb/ponnurangam.kumaraguru