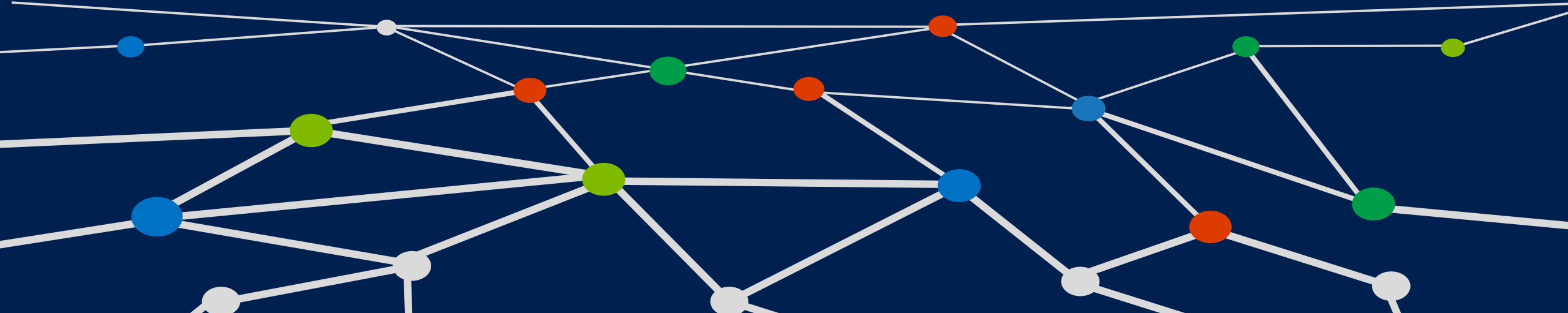


# Building A Conceptual Framework for Network Analytics

Minjian Liu ( u5506264 )  
Supervised by Dr. Qing Wang



# OUTLINE

## 1 Project Overview

High-level picture about Conceptual Framework



## 2 Project Schedule

Details about the work package timeline



## 3 Current Status

Demonstration of all current functions



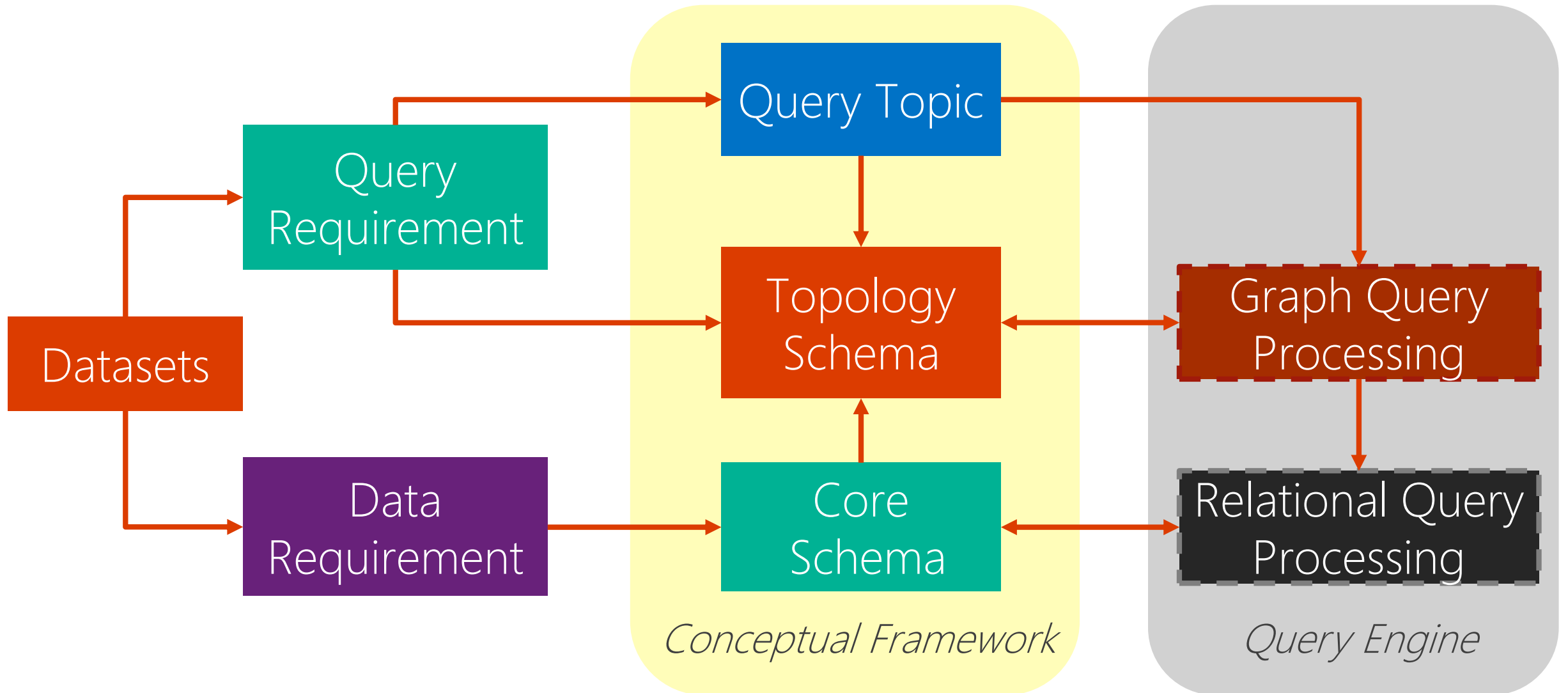
## 4 Future Work

Work plan for next semester



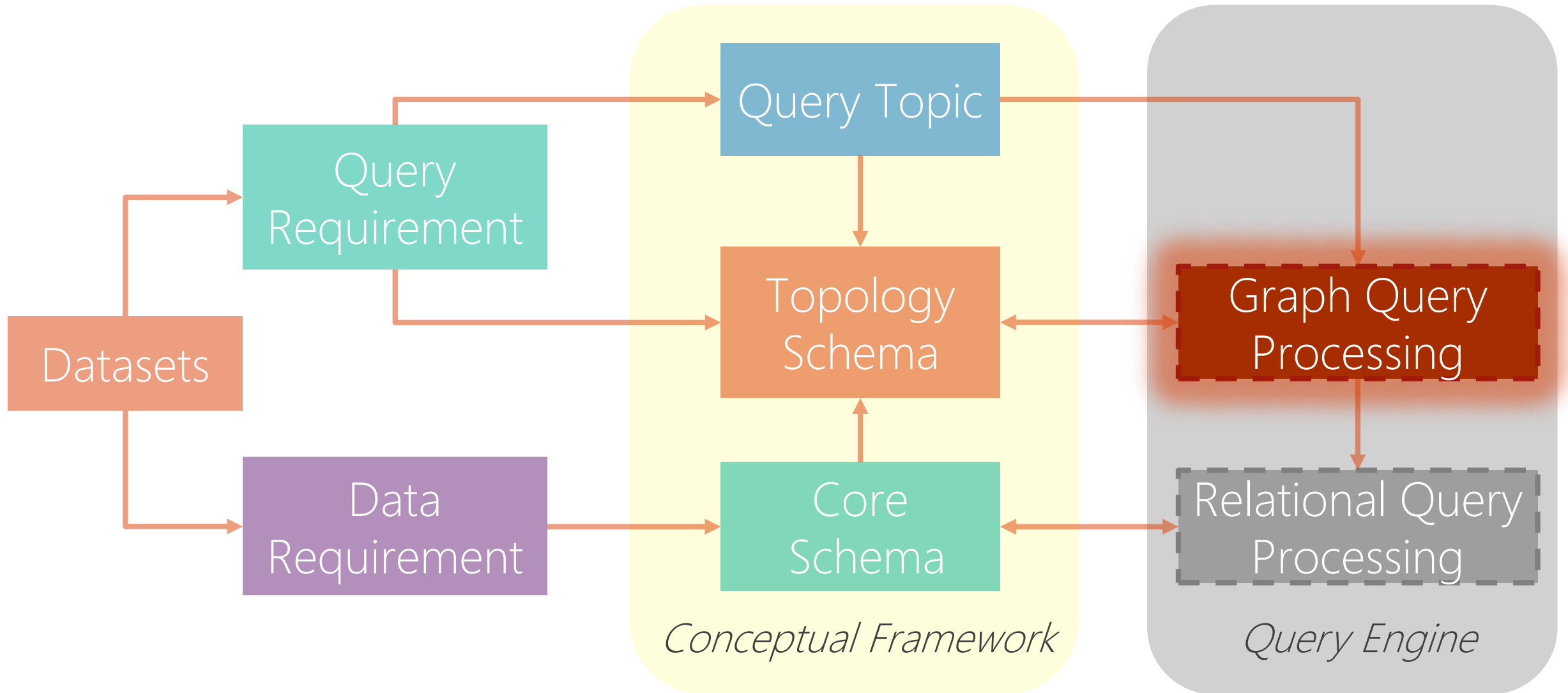


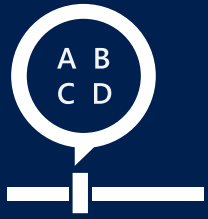
# Project Overview





# Project Overview





# Project Schedule

Gain experiment  
datasets and  
decide tools

Do conceptual  
modelling and  
design query  
language

Implement  
optimization  
strategy

Do performance  
evaluation

Submit thesis

Mar.

Apr.

May

Jun.

Jul.

Aug.

Sep.

Oct.

Nov.





# Current Status



## Stack Overflow Dataset

19,824,320 posts; 7,214,697 questions  
12,609,623 answers

### Normal Relational Query:

Q1: Answer Owner (For c# Questions)

### Rank Function:

Q2: Influence of answer owner  
(*Centrality* in the Graph)

Q3: Top 10 c# Experts

### Cluster Function:

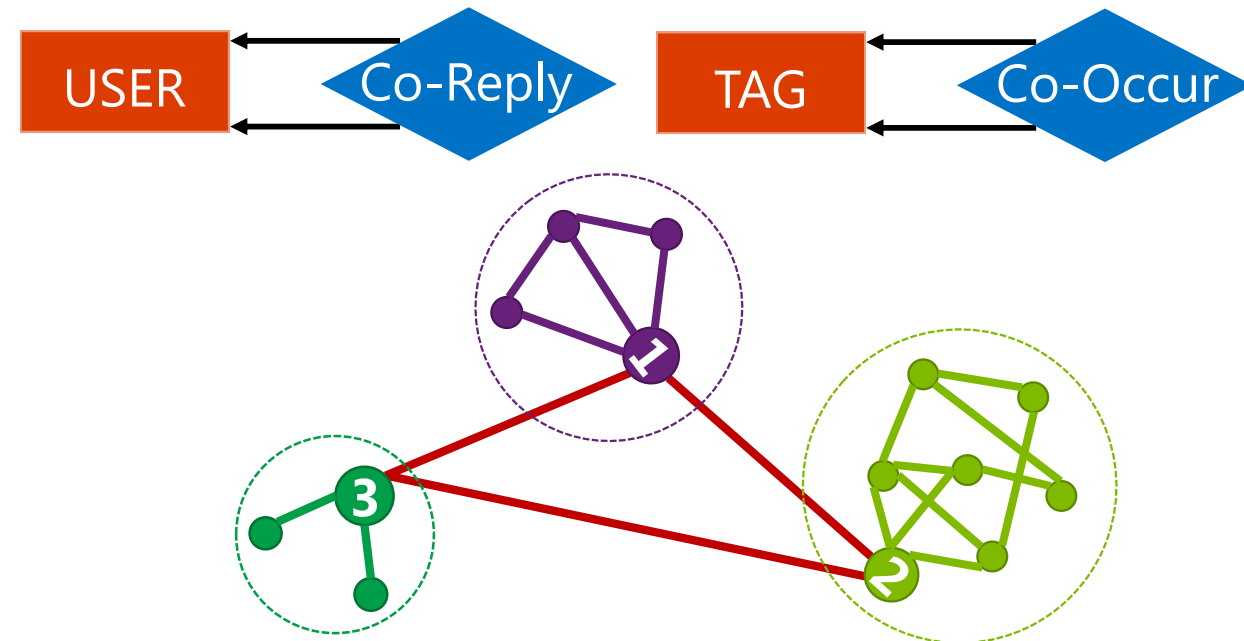
Q4: Set of Correlative Tags

Q5: Tags often used with  
programming language tags together

QuestionId	Accepted AnswerId	Question OwnerId	TagSet
8	162	9	c#,code-generation,j#
24	49	22	mysql,database,triggers

AnswerId	Parent QuestionId	Answer OwnerId
162	8	17
4061	8	526

TagId	TagLabel
3333	c#
72	mysql





# Current Status

## Path Function:

Q6: Find paths between user S and user D with the length is less than 3; User S & user D both have sent a tweet with a hashtag "iphone".

Q7: Is there any path between user S and user D (reachability).

Q8: Find paths between user S and user D, but user 17335 is in the middle of the path (with node constraints).

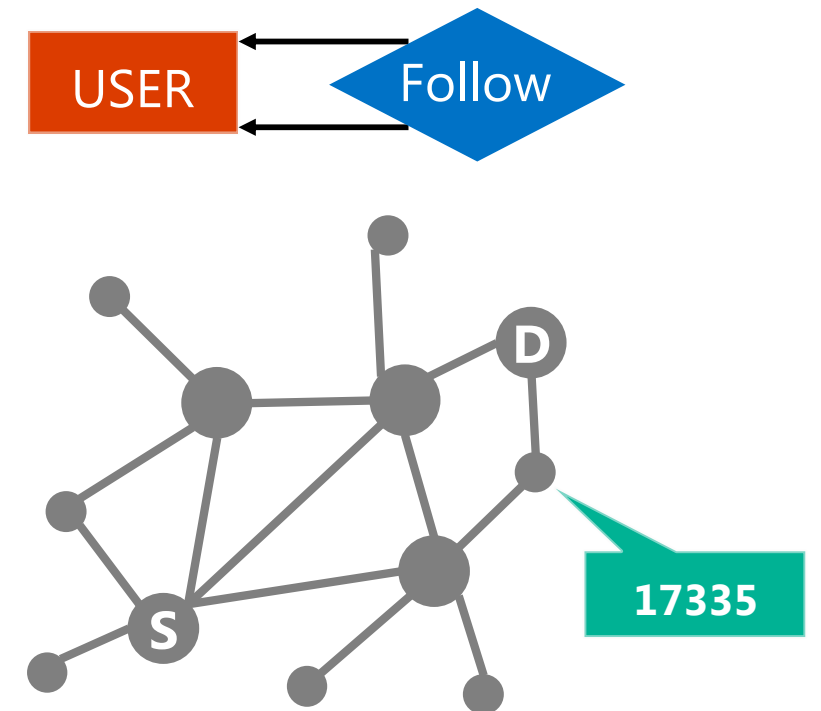


## Twitter Dataset

17,069,982 users; 476,553,560 tweets  
All users' follow relationship

TweetId	TweetOwner	HashTag	MentionedUser
2	cyberplumber	iphone	none
17	alphaexe	none	lessalla

UserId	UserName	TagId	Label	UserId	FollowerId
17095	cyberplumber	6278	iphone	17335	1850110
17335	alphaexe	127	WWDC	17335	9003541



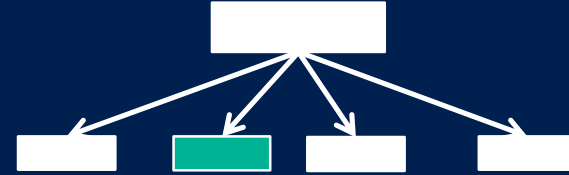


# Current Status

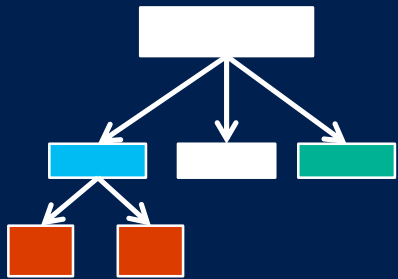
Query Tree



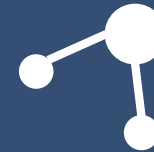
Query Tree



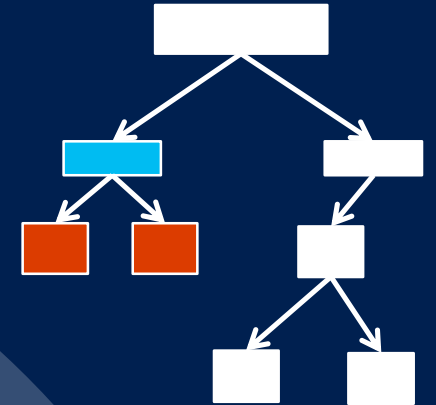
Query Tree



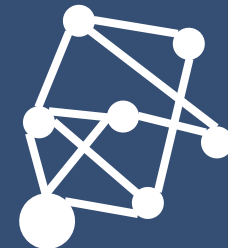
Query Topic



Query Tree



Topology Schema



Table



Core Schema



Table



Table





# Current Status

## Rank Function



*Evaluate node centrality:*

Degree -- Undirected Graph  
Indegree -- Directed Graph  
Outdegree -- Directed Graph  
Page Rank -- Undirected & directed Graph  
Betweenness -- Undirected Graph  
Closeness -- Undirected Graph

## Cluster Function



*Cluster detection (Undirected Graph):*

GN : Girvan-Newman Algorithm  
CNM : Clauset-Newman-Moore Algorithm

## Paths Function



*Find paths between nodes:*

$S \setminus \setminus \setminus D$  (with path length condition)  
 $S \setminus \setminus \setminus D$  (any path length, reachability)  
 $S \setminus \setminus N \setminus \setminus D$  (length and node condition)  
 $S \setminus \setminus \setminus N \setminus \setminus D$  (node condition)

## Example & Result

**rank** (graph constructor, table-cross condition, measurement)

Node ID	Measurement Value
32	56.00
15	43.00

**cluster** (graph constructor, cluster relation, measurement)

Cluster ID	Members
1	{13, 64, 147, 926, 1032, 1051}
2	{15, 78}

**paths** ( graph constructor, path notation, node conditions)

Path ID	Length	Path
1	5	{12, 18, 93, 116, 257, 589}
2	3	{7, 9, 178, 276}

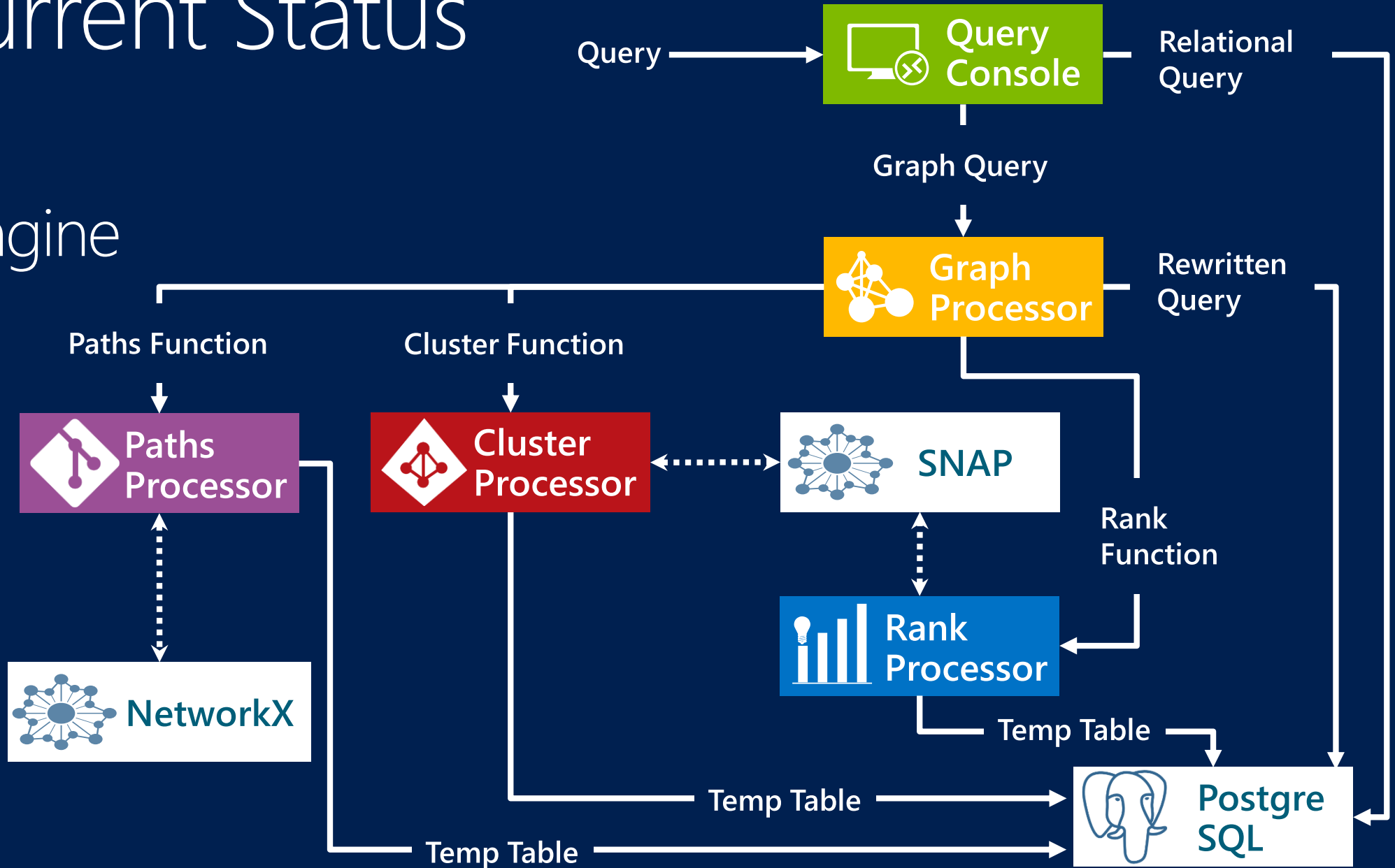


Demo



# Current Status

## Details of Query Engine





# Future Work

Conceptual Layer



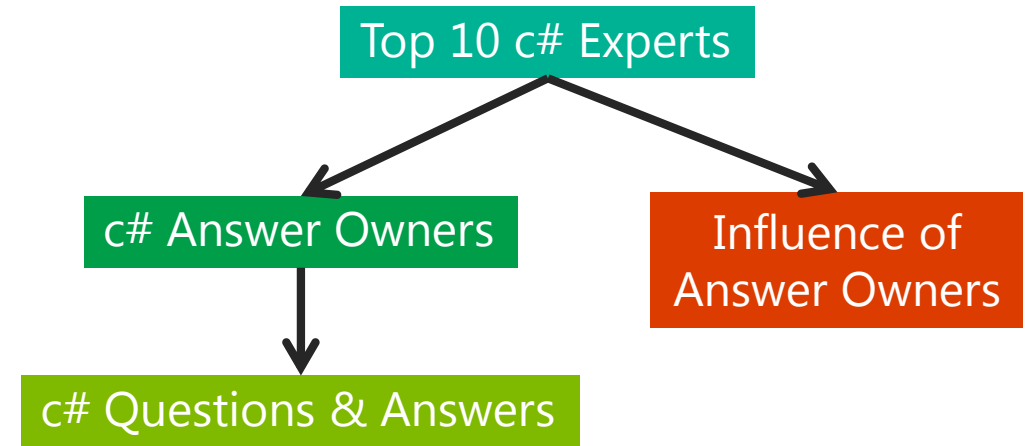
Logical Layer



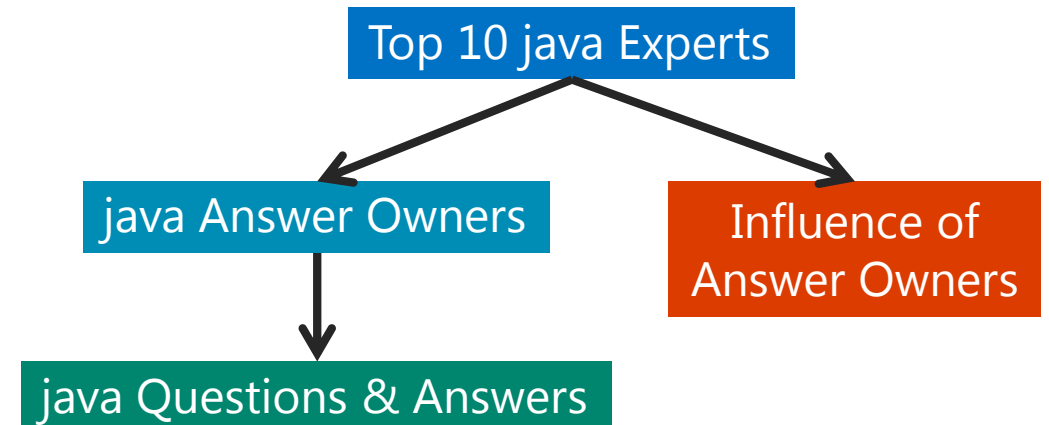
Physical Layer



## Q3: Top 10 c# Experts:



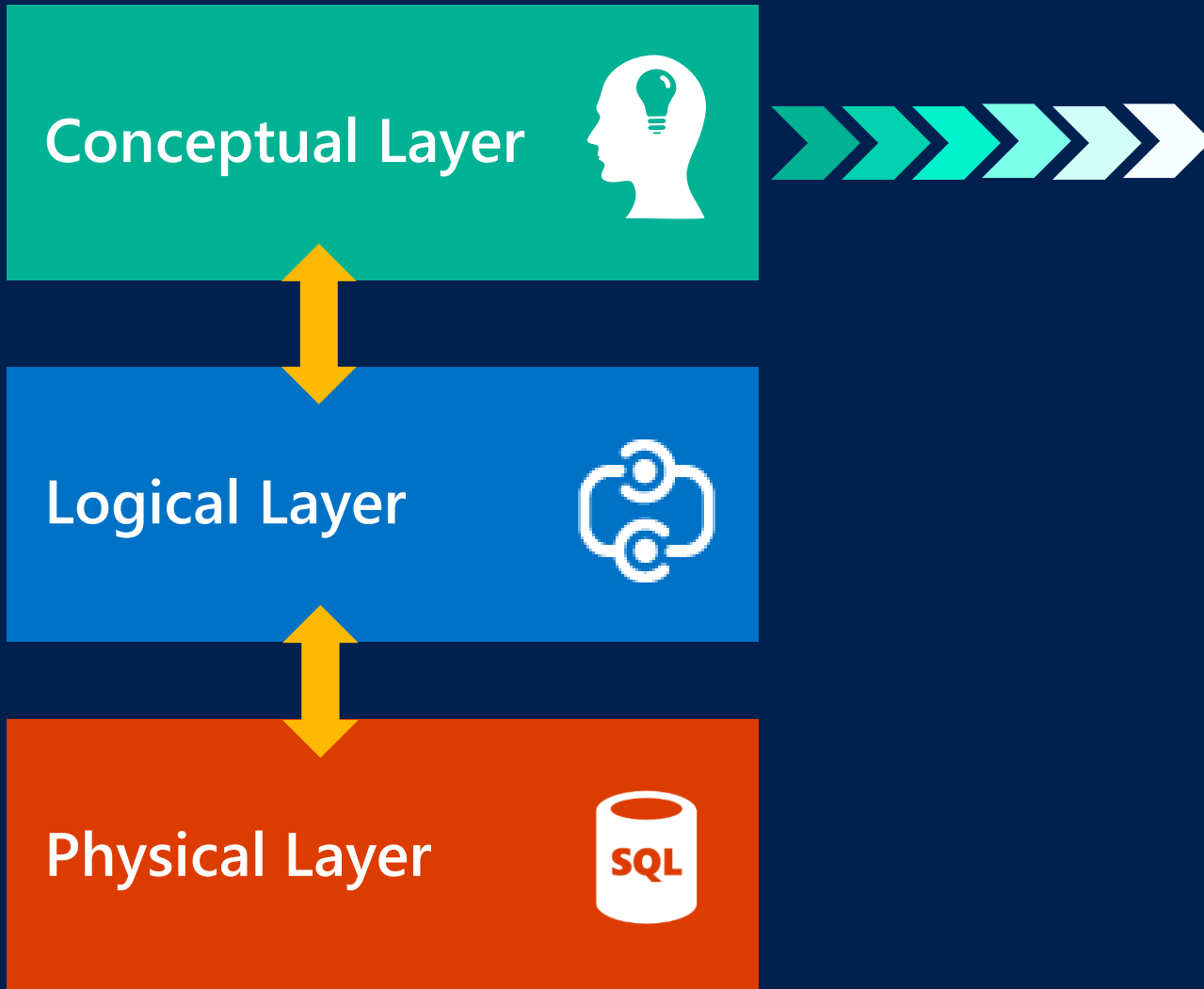
## Q9: Top 10 java Experts:



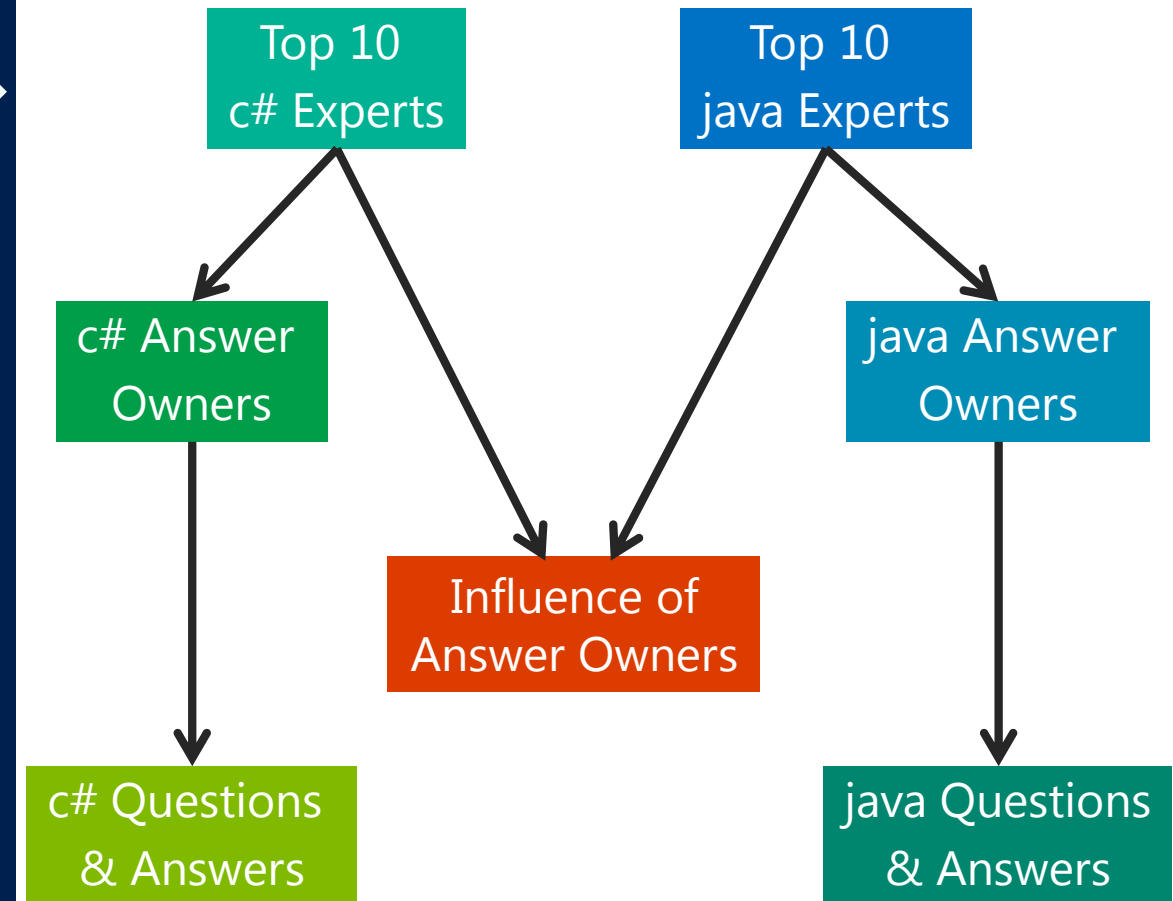




# Future Work



## Query Matching:





# Future Work

Conceptual Layer



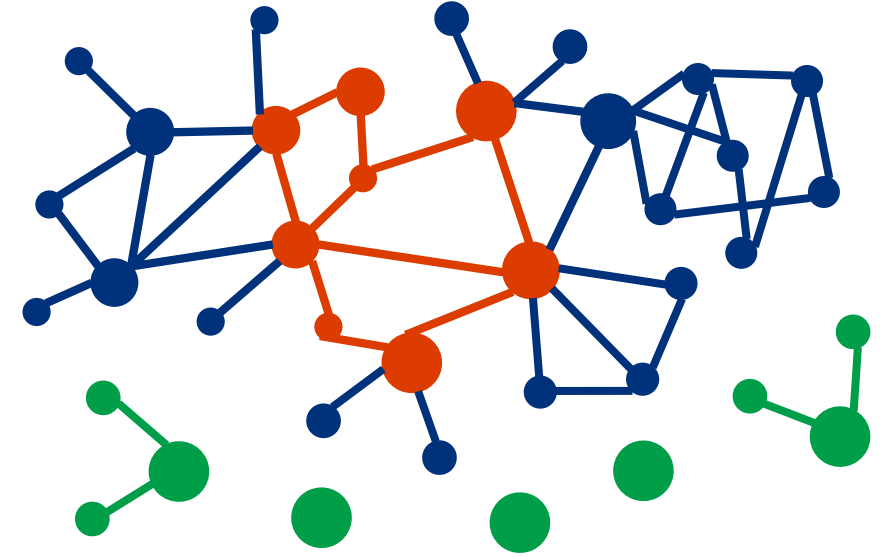
Logical Layer



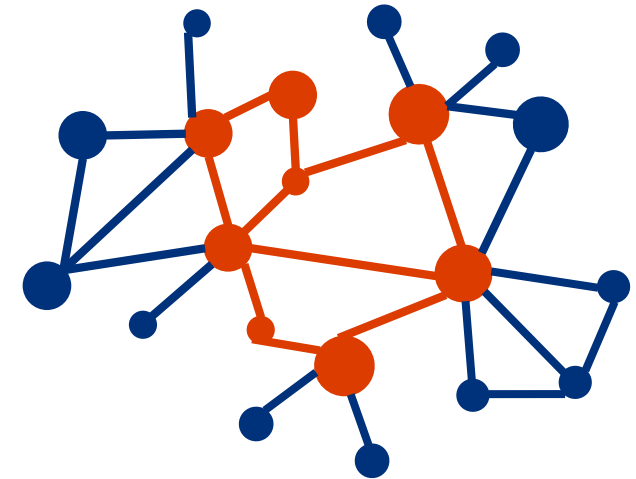
Physical Layer



Complete Graph:



Sub-Graph:





# Future Work

Conceptual Layer



Logical Layer



Physical Layer



## Materialized Views

- A database object that contains the results of a query
- As a form of pre-computation, materialized views are typically created for performance reasons



## Indexing

An index allows DB to find and retrieve specific rows much faster.

### Index Types:

- B-tree
- Hash
- GiST
- GIN



## Caching

Avoid repeated computation

How to store:

- Data caching
- Pointer caching

What to store:

- LRU: least recently used
- LFU: least frequently used
- LCS: largest cache space required

Thank You !  
Q & A



**Contact:**

**Minjian Liu**



**0426839321**



**378610682**



**u5506264@anu.edu.au**

Some of pictures are downloaded from [www.rapidbbs.cn](http://www.rapidbbs.cn)

The source code and query samples are available at  
[https://gitlab.com/Minken/COMP8800\\_Computing\\_Research\\_Project.git](https://gitlab.com/Minken/COMP8800_Computing_Research_Project.git)

The content of this file is based on the following articles:  
• Network Analytics ER Model – Towards a Conceptual View  
of Network Analytics, Qing Wang, ER 2014.