

# 数据模型

数据库的Schema

```
email: string @index(exact) .  
name: string .  
work_on: [uid] .  
manage: [uid] .
```

```
notification: [uid] .  
err: [uid] .
```

```
title: string .  
description: string .  
assignee: [uid] @reverse .  
status: string .  
priority: string .  
reporter: [uid] @reverse .
```

```
message: string .  
n_task: [uid] @reverse .  
send_to: [uid] @reverse .
```

```
sync_action: string .  
try_times: int .  
err_task: [uid] @reverse .
```

```
create_date: int .
```

```
type User {  
    email  
    name  
}
```

```
type Task {  
    title  
    description
```

```
    assignee
    status
    priority
    reporter
}

type Notification {
    n_task
    message
    send_to
    create_date
}

type SynError {
    sync_action
    err_task
    try_times
    create_date
}
```

这里设计4个Type与20个Predicate

## User

记录用户信息

attribute	type	description
name	string	用户名
email	string	用户邮箱 （同时作为用户唯一标示）

## Task

记录事务信息及与用户的指派关系

attribute	type	description
title	string	事务名
description	string	事务详情
status	string	事务状态 (todo, processing, QA, done)
priority	string	事务优先级 (low, medium, high)
assignee	[User]	执行者
reporter	[User]	管理者

#### NOTE:

- 目前API仅支持事务拥有一个执行者与管理者

## Notification

记录事务事务状态变化消息

attribute	type	description
message	string	消息内容
send_to	[User]	收到消息的用户
n_task	[Task]	消息对应的事务
create_date	int	创建时间

#### NOTE:

- `send_to` 将同时指向Task的 `assignee` and `reporter`
- 目前仅支持create/update Task时，发送消息

## SyncError

记录DB/Elasticsearch同步过程中发生的错误，导致数据inconstant

attribute	type	description
sync_action	string	同步操作类型 （create, update, delete）
err_task	[Task]	所对应的事务
try_times	int	后台自动同步机制尝试同步的次数
create_date	int	创建时间

#### NOTE:

- sync\_action 用来记录create/update/delete Task的操作类型
- create\_date 用来排序
- 当后台尝试3次同步仍失败，将会发送notification给admin
- 当同步成功后，相应SyncError记录会从数据库中删除

## 设计意图

### 所希望实现的功能

#### User

- 查看消息（notification）
- 查看所分配的任务（work\_on）
- 查看所管理的任务（mange）

#### Task

- 查看事务的执行人（assignee）与管理者（reporter）
- 查看事务有关的消息（notification）
- 查看事务有关的同步错误记录（err）

## Other

- sort SyncError by create date
- sort Notification by create date

## 功能结果展示

用API进行如下操作:

- 创建两个User: yang & laura

```
$ curl http://localhost:8888/dgraph_sync/user -X POST -H 'Content-Type: application/json' -d '{"name": "laura", "email": "laura@ab.cd"}'
```

```
$ curl http://localhost:8888/dgraph_sync/user -X POST -H 'Content-Type: application/json' -d '{"name": "yang", "email": "yang@ab.cd"}'
```

- 创建一个Task
  - reporter: laura
  - assignee: yang

```
$ curl http://localhost:8888/dgraph_sync/task -X POST -H 'Content-Type: application/json' -d '{
  "title": "run 10000m",
  "description": "it is hard",
  "priority": "midum",
  "status": "todo",
  "reporter": {
    "email": "laura@ab.cd"
  },
  "assignee": {
    "email": "yang@ab.cd"
  }
}'
```

- Update刚刚创建的Task

```
$ curl http://localhost:8888/dgraph_sync/task -X PUT -H 'Content-Type: application/json' -d '{
  "uid": "0x1117d",
  "title": "run 100m",
  "description": "try you best",
  "priority": "midum",
  "status": "todo"
}'
```

#### NOTE:

- 此时关闭ES服务
- 此时代码已启动

#### 此时数据库中

- 2个User
- 1个Task
- 2个Notification
- 2个SyncError

#### 在Dgraph中运行

#### 查看yang

```
{
  q(func: eq(email, "yang@ab.cd")) {
    uid
    name
    email
    work_on: ~assignee {
      uid
      expand(_all_) {
        expand(_all_)
      }
    }
  }
  manage: ~reporter {
    uid
  }
}
```

```

        expand(_all_) {
            expand(_all_)
        }
    }
    notification: ~send_to {
        uid
        expand(_all_) {
            expand(_all_)
        }
    }
}
}
}

```

结果:

```

{
  "data": {
    "q": [
      {
        "uid": "0x13881",
        "name": "yang",
        "email": "yang@ab.cd",
        "work_on": [
          {
            "uid": "0x13883",
            "reporter": [
              {
                "email": "laura@ab.cd",
                "name": "laura"
              }
            ],
            "title": "run 100m",
            "description": "try you best",
            "assignee": [
              {
                "email": "yang@ab.cd",
                "name": "yang"
              }
            ],
            "status": "todo",
            "priority": "midum"
          }
        ]
      }
    ]
  }
}

```

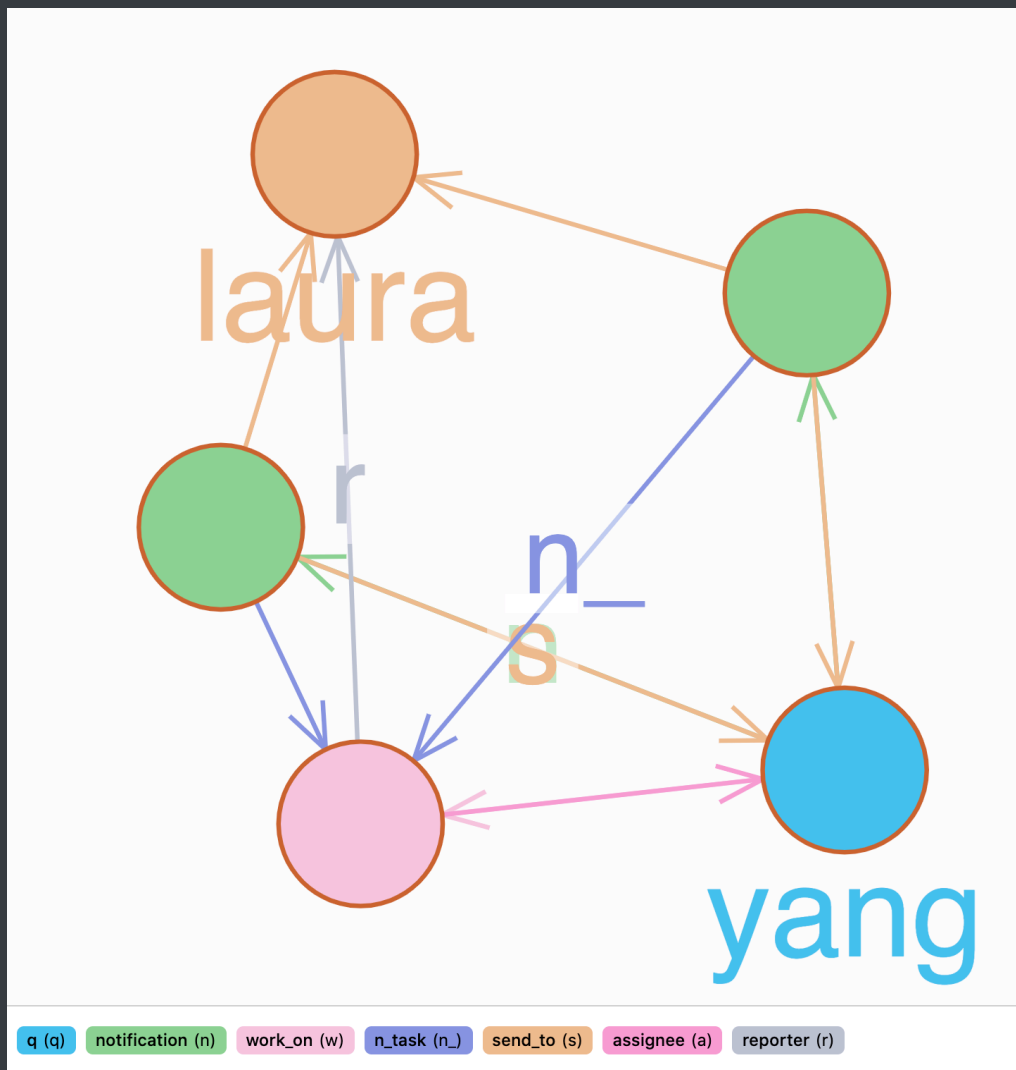






```
}  
}
```

Graph:



查看laura

```
{  
  q(func: eq(email, "laura@ab.cd")) {  
    uid  
    name  
    email  
    dgraph.type  
    work_on: ~assignee {  
      uid  
    }  
  }  
}
```

```

        expand(_all_) {
            expand(_all_)
        }
    }
    manage: ~reporter {
        uid
        expand(_all_) {
            expand(_all_)
        }
    }
    notification: ~send_to {
        uid
        expand(_all_) {
            expand(_all_)
        }
    }
}
}
}

```

结果:

```

{
  "data": {
    "q": [
      {
        "uid": "0x13882",
        "name": "laura",
        "email": "laura@ab.cd",
        "dgraph.type": [
          "User"
        ],
        "manage": [
          {
            "uid": "0x13883",
            "title": "run 100m",
            "description": "try you best",
            "assignee": [
              {
                "email": "yang@ab.cd",
                "name": "yang"
              }
            ]
          }
        ]
      }
    ]
  }
}

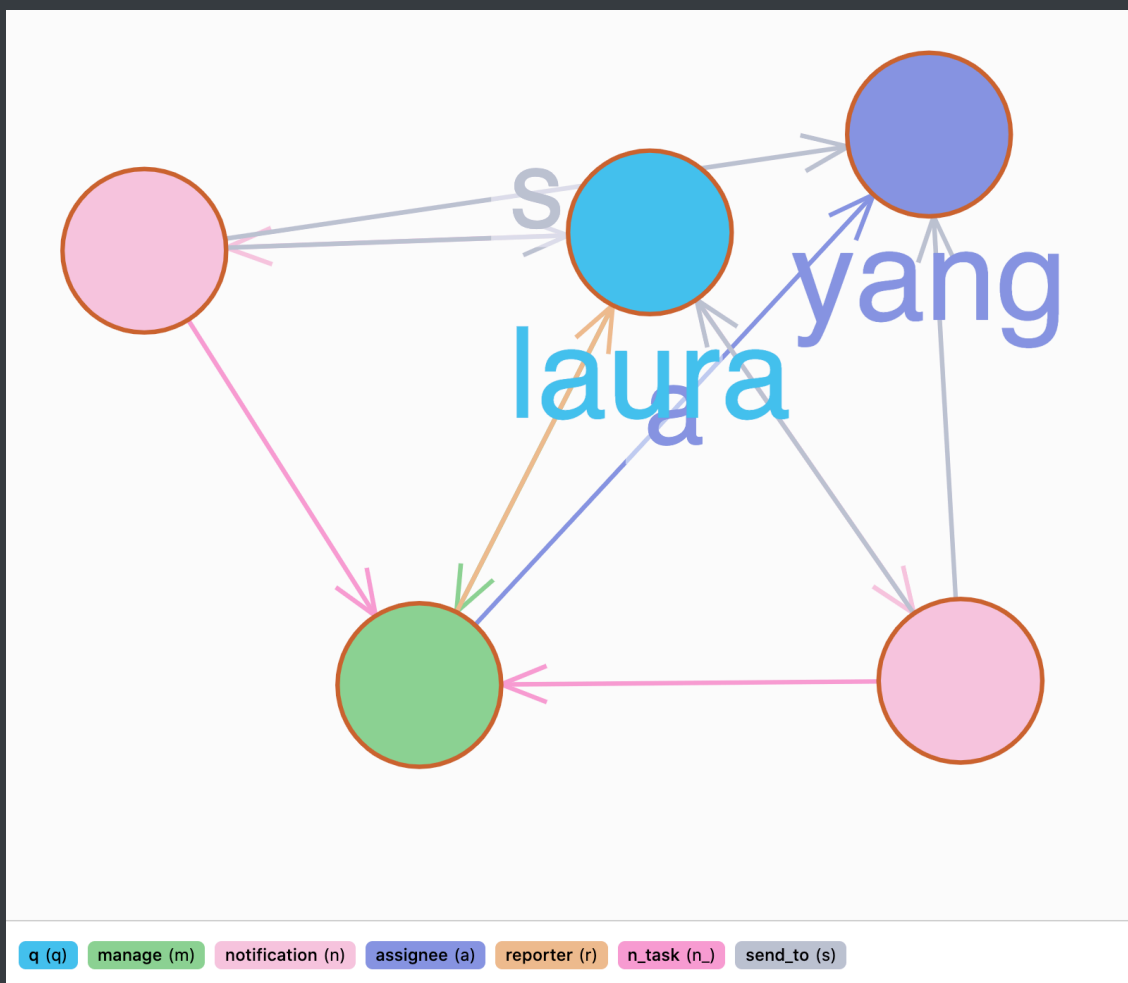
```





```
    "reporter": 2,  
    "send_to": 2,  
    "status": 2,  
    "title": 2,  
    "uid": 4,  
    "~assignee": 1,  
    "~reporter": 1,  
    "~send_to": 1  
  }  
}  
}  
}
```

Graph



查看Task

```

{
  q(func: uid("0x1117d")){
    uid
    expand(_all_) {
      uid
      expand(_all_)
    }
    notification: ~n_task {
      uid
      n_task {
        expand(_all_)
      }
    }
    err: ~err_task {
      uid
      expand(_all_) {
        expand(_all_)
      }
    }
  }
}

```

结果:

```

{
  "data": {
    "q": [
      {
        "uid": "0x13883",
        "priority": "midum",
        "reporter": [
          {
            "uid": "0x13882",
            "email": "laura@ab.cd",
            "name": "laura"
          }
        ],
        "title": "run 100m",
        "description": "try you best",
        "assignee": [

```

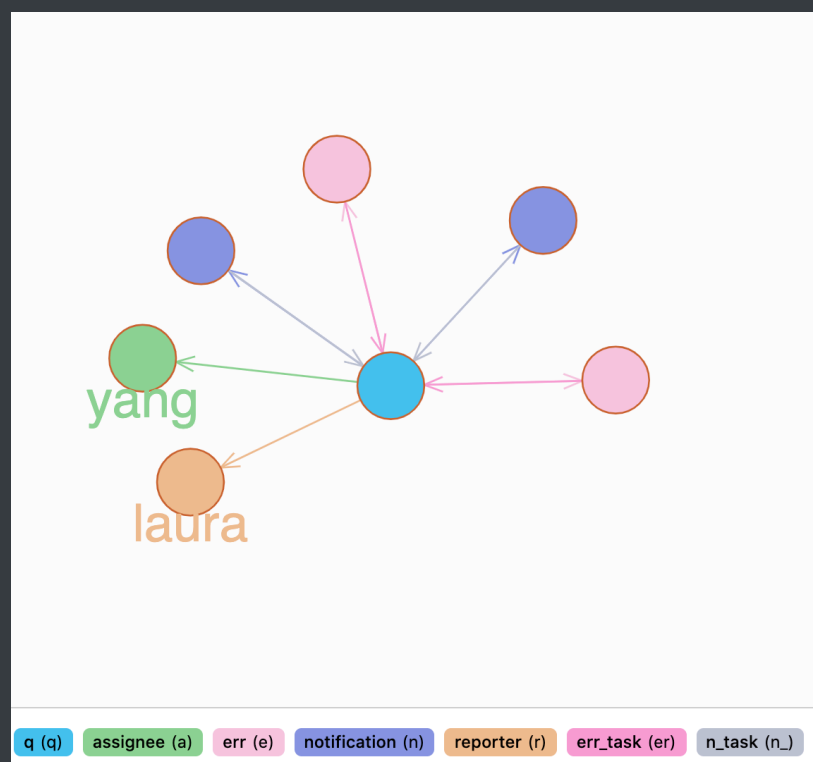






```
    "name": 2,  
    "priority": 3,  
    "reporter": 3,  
    "status": 3,  
    "sync_action": 2,  
    "title": 3,  
    "try_times": 2,  
    "uid": 7,  
    "~err_task": 1,  
    "~n_task": 1  
  }  
}  
}  
}
```

## Graph



查看消息列表：

```

{
  q(func: has(message),orderasc: create_date) {
    uid
    expand(_all_) {
      expand(_all_)
    }
  }
}

```

## 结果

```

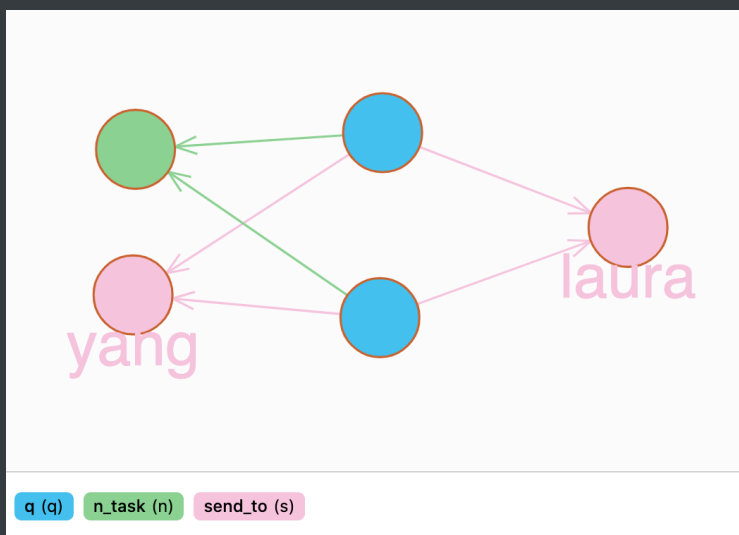
{
  "data": {
    "q": [
      {
        "uid": "0x13884",
        "create_date": 1594008010,
        "n_task": [
          {
            "priority": "midum",
            "title": "run 100m",
            "description": "try you best",
            "status": "todo"
          }
        ],
        "message": "task created",
        "send_to": [
          {
            "email": "yang@ab.cd",
            "name": "yang"
          },
          {
            "email": "laura@ab.cd",
            "name": "laura"
          }
        ]
      },
      {
        "uid": "0x13886",
        "create_date": 1594008043,

```



```
    "priority": 1,  
    "reporter": 1,  
    "send_to": 2,  
    "status": 1,  
    "title": 1,  
    "uid": 2  
  }  
}  
}
```

## Graph



## 查看SyncError

```
{  
  q(func: has(sync_action),orderasc: create_date) {  
    uid  
    expand(_all_) {  
      uid  
      expand(_all_)  
    }  
  }  
}
```

## 结果



```
    "total_ns": 14067900
  },
  "txn": {
    "start_ts": 250279
  },
  "metrics": {
    "num_uids": {
      "assignee": 1,
      "create_date": 2,
      "description": 1,
      "err_task": 2,
      "priority": 1,
      "reporter": 1,
      "status": 1,
      "sync_action": 2,
      "title": 1,
      "try_times": 2,
      "uid": 3
    }
  }
}
```

## Graph

