

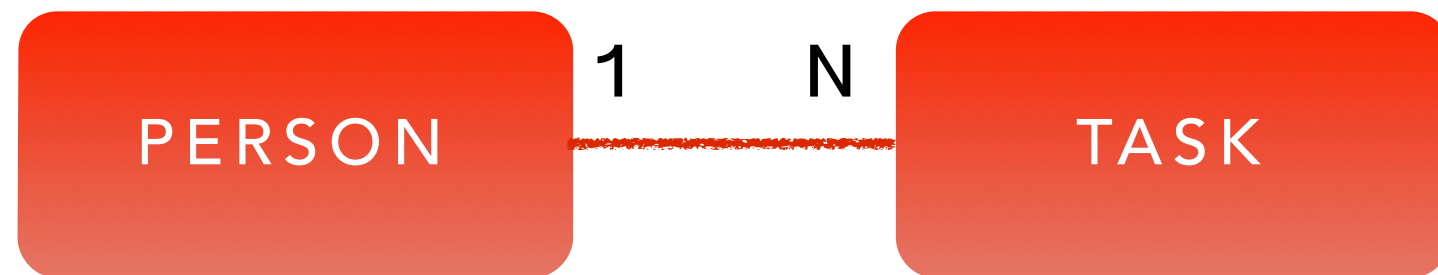
MongoDB

Data modeling part 2

Data modeling

- What is the cardinality of relationship ?
- Entity need standalone ?
- **Ratio Read vs. Write**

Entity



คำถาม

ถ้าต้องการค้นหา task ของแต่ละคน ?

Two-way referencing

person =

```
{  
  _id: 1,  
  name: "Somkiat Puisungnoen",  
  tasks: [1,2]  
}
```

task

```
{  
  _id: 1,  
  name: "task 1",  
  owner_id: 1  
}
```

task

```
{  
  _id: 2,  
  name: "task 2",  
  owner_id: 1  
}
```

คำถาม

ถ้าต้องการลดการ join ในระดับ app ละ ?

```
person =  
{  
  _id: 1,  
  name: "Somkiat Puisungnoen",  
  tasks: [1,2]  
}
```

Application ต้องการเพียงชื่อ task

```
person =  
{  
  _id: 1,  
  name: "Somkiat Puisungnoen",  
  tasks: [  
    { _id: 1, name: "task 1" },  
    { _id: 2, name: "task 2" },  
  ]  
}
```

Denormalize with Many -> One

- ลดการ join ในระดับ application
- แต่ขั้นตอนการ update จะเพิ่มขึ้น
- Good for Read > Write

ในการดึงข้อมูล Task ต้องการชื่อเจ้าของด้วย ?

task

```
{  
  _id: 1,  
  name: "task 1",  
  owner_id: 1  
}
```

Denormalize with One -> Many

task

```
{  
  _id: 1,  
  name: "task 1",  
  owner_id: 1,  
  owner_name: "Somkiat Puisungnoen"  
}
```

Data modeling

- What is the cardinality of relationship ?
- Entity need standalone ?
- Ratio Read vs. Write

Rules for design schema

- Embedded first
- Standalone entity ?
- Growth of size of array
- Don't fear application join
- Denormalize => Read vs Write
- Behavior of application

MongoDB

Design schema to match
the need of your application
not your need

Workshop

- สร้างระบบ contact

