

KM-Project Presentation

Juyee Myeong

Juyeon Nam

Content

1. Idea
2. KB-Schema
3. Implemented instances
4. Queries – simple, Intermediate, Advanced

Vikini City



SpongeBob



Gary



Patrick Star



Squidward



Sandy Cheeks



Plankton



Mr. Krab



Pearl



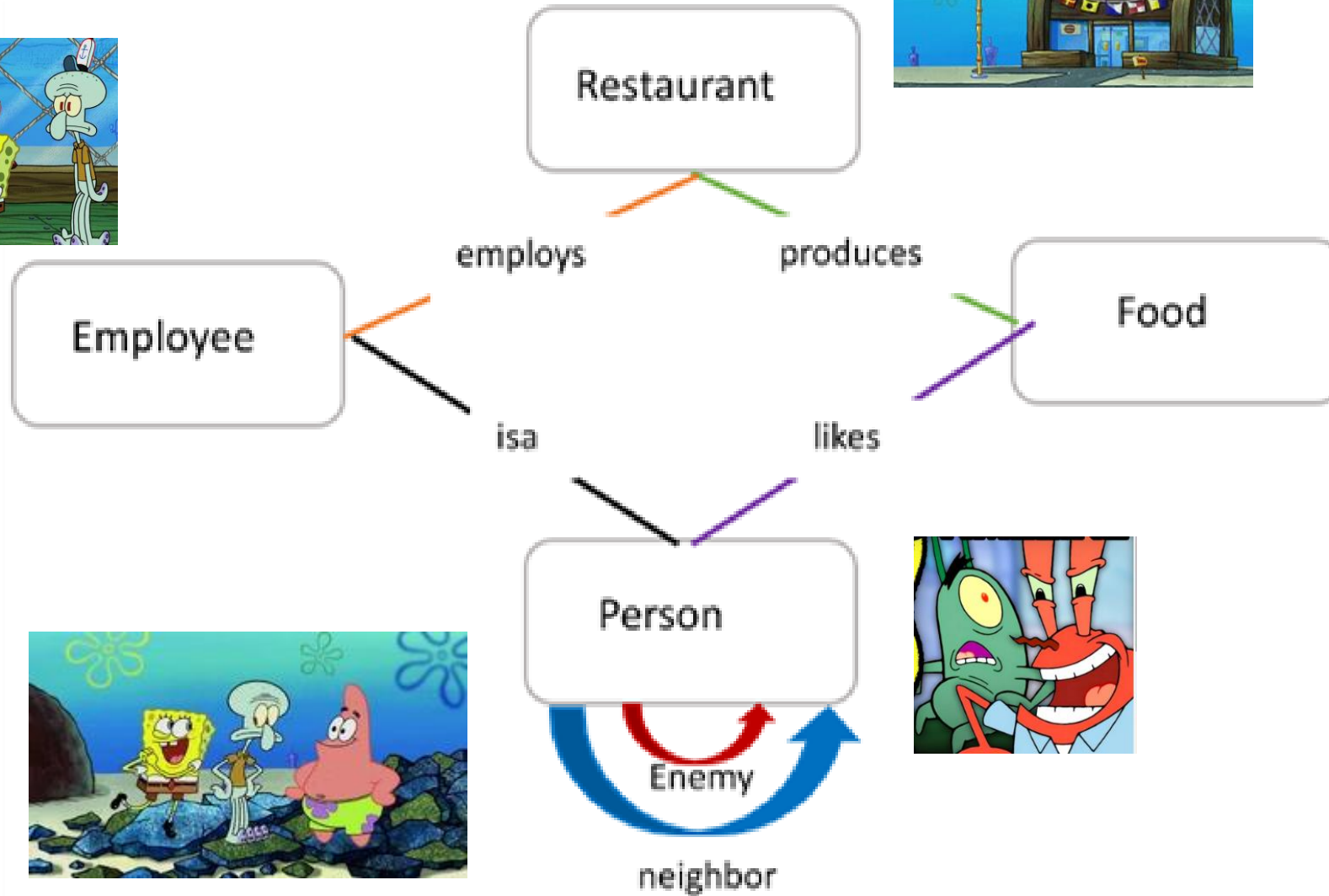
1. Idea

Information & Relationship

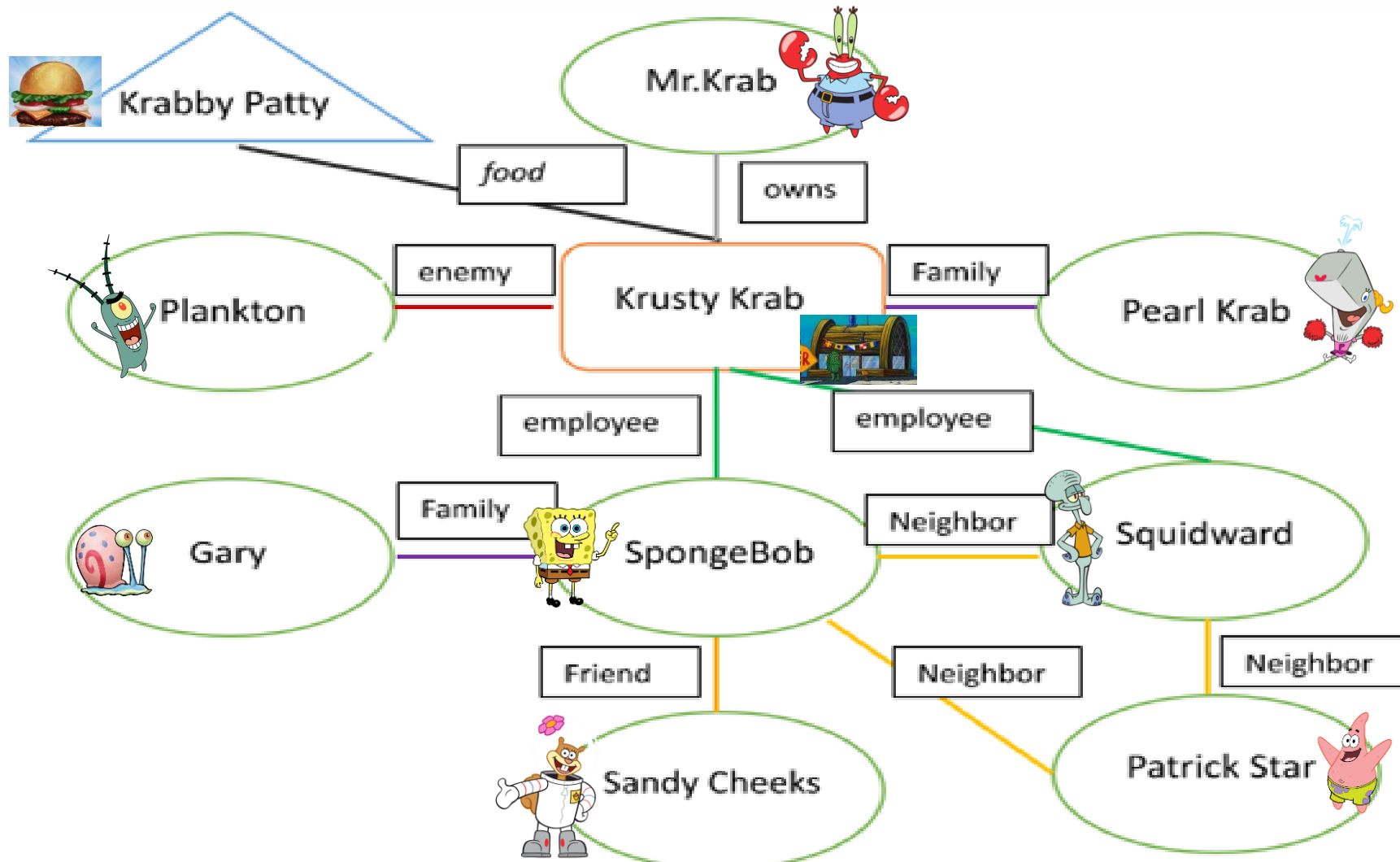
- KM about **Krusty Krab – Krabby Patty – Residents of Bikini City** relationship
 - **Krusty Krab(Restaurant)**
 - Who owns Krusty Krab
 - Who works at Krusty Krab
 - Who is going to be hired at Krusty Krab
 - **Krabby Patty**
 - Who likes or dislikes it
 - **Residents of Bikini City**
 - Neighbor / Friend & Enemy / Family
 - How much is Resident1's salary
 - What is Resident2's personality like
 - Can Resident3 eat Krabby-Patty every day



2. Schema



3. 1 Implemented Instance - Simple



3. 1 Implemented Instance - Intermediate & Advanced



Personality

- positive
- clean
- curious
- friendly
- dumb

Appearance

- Outfit: Shirt, Squarepants, Shoes
- Yellow
- Big tooth
- Smile



Sandy Cheeks

Personality

- positive
- clean
- curious
- friendly

Appearance

- Outfit: Spacesuit, Shoes
- Brown
- tooth
- Smile



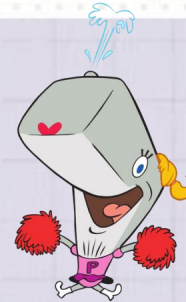
Patrick Star

Personality

- positive
- curious
- friendly
- dumb

Appearance

- Outfit: Flowerpants
- Pink
- Smile



Pearl

Personality

- positive
- clean
- curious
- friendly
- dumb

Appearance

- Outfit: TShirt, Skirt, Shoes
- Blue
- tooth
- Smile



Squidward

Personality

- clean

Appearance

- Outfit: Shirt, Blue

3.2 Part of KM code

```
KM> (*Spongebob has
(instance-of (Sponge))
(employee-of (*Krusty-Krab))
(friend (*Patrick *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl))
(friend-of (*Patrick *Sandy *Mr.Krab *Pearl *Gary))
(family (*Gary))
(enemy-of (*Squidward *Plankton))
(likes (*Krabby-Patty *Fish))
(neighbor (*Patrick *Squidward)))

(*Patrick has
(instance-of (Fish))
(friend (*Spongebob *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl))
(friend-of (*Spongebob *Sandy *Mr.Krab *Gary *Pearl))
(enemy-of (*Squidward *Plankton))
(likes (*Krabby-Patty))
(neighbor (*Spongebob *Squidward)))

(*Squidward has
(instance-of (Fish))
(friend (*Sandy))
(friend-of (*Sandy))
(enemy (*Spongebob *Patrick))
(dislikes (Fish *Spongebob *Krabby-Patty))
(neighbor (*Spongebob *Patrick)))

(*Sandy has
(instance-of (Animal))
(friend (*Spongebob *Squidward *Patrick *Gary *Mr.Krab))
(friend-of (*Spongebob *Squidward *Patrick *Gary *Mr.Krab))
(likes (*Krabby-Patty)))
```

```
(*Mr.Krab has
(instance-of (Fish))
(friend (*Spongebob *Patrick *Squidward *Gary *Sandy *Pearl))
(friend-of (*Spongebob *Patrick *Gary *Sandy *Pearl))
(enemy (*Plankton))
(enemy-of (*Plankton))
(owns (*Krusty-Krab))
(family (*Pearl))
(likes (*Krabby-Patty)))

(*Plankton has
(instance-of (Fish))
(friend-of (*Spongebob *Patrick))
(enemy (*Mr.Krab *Spongebob *Patrick))
(enemy-of (*Mr.Krab))
(likes (*Krabby-Patty)))

(*Gary has
(instance-of (Fish))
(friend (*Spongebob *Patrick *Sandy *Mr.Krab))
(friend-of (*Spongebob *Patrick *Sandy *Mr.Krab))
(family (*Spongebob)))

(*Pearl has
(instance-of (Fish))
(friend (*Spongebob *Patrick *Mr.Krab))
(friend-of (*Spongebob *Patrick *Mr.Krab))
(family (*Mr.Krab))
(likes (*Krabby-Patty)))

(*Krusty-Krab has
(instance-of (Restaurant))
(produces (*Krabby-Patty))
(employee (*Spongebob *Squidward)))
```


3.2 Part of KM code

```
(*Spongebob has
(instance-of ((a Sponge)))
(salary (1500))
(job ((a Job with
      (make ((the produces of (the restaurant of (the likes of (the friend of Sp
      (sex (male))
      (lives-in (vikini-city))
      (personality ((a Personality with
                     (yes (Positive Clean Friendly))
                     (no (Notcurious Notdumb))))))
      (appearance ((a Appearance with
                     (outfit ((a Outfit with
                               (top (Shirt))
                               (bottom (Squarepants))
                               (shoes (Yes))))))
                     (color (Yellow))
                     (tooth (Big))
                     (smile (Yes))))))))))
```

```
(*Spongebob has
  (friend ((a Friend with
            (name (Larry))
            (instance-of ((a Fish)))
            (lives-in (vikini-city))
            (sex (male))
            (favorite ((a Favorite with
                        (name (Krabby-Patty))
                        (price (3.99))
                        (category (burger))
                        (restaurant (Krusty-Krab))))))))))
```

```
(*Sandy has
(instance-of ((a Animal)))
(salary (1500))
(salary-want (3500))
(apply (applicant))
(sex (female))
(lives-in (vikini-city))
(personality ((a Personality with
                (yes (Positive Clean Friendly Notdumb))
                (no (Notcurious))))))
      (appearance ((a Appearance with
                    (outfit ((a Outfit with
                              (top (Spacesuit))
                              (bottom (Spacesuit))
                              (shoes (Yes))))))
                    (color (Brown))
                    (tooth (Yes))
                    (smile (Yes))))))
```

```
(*Pearl has
(instance-of ((a Fish)))
(salary (2000))
(salary-want (3000))
(apply (applicant))
(sex (female))
(lives-in (vikini-city))
(personality ((a Personality with
                (yes (Notcurious Friendly Notdumb))
                (no (Positive Clean))))))
      (appearance ((a Appearance with
                    (outfit ((a Outfit with
                              (top (Tshirt))
                              (bottom (Skirt))
                              (shoes (Yes))))))
                    (color (Blue))
                    (tooth (Yes))
                    (smile (Yes))))))
```

4-1. Simple Queries - 1, 2

```
KM> (the neighbor of *Spongebob)
1 -> (the neighbor of *Spongebob)
1 (Retrieving answer computed and cached earlier:
1 (the neighbor of *Spongebob) = (*Patrick *Squidward)))
1 <- (*Patrick *Squidward) [(the neighbor of *Spongebob)]
(*Patrick *Squidward)
(1 inferences and 2 KB accesses in 0.0 sec [31 lips, 62 kaps]))
```

Find the neighbor of SpongeBob

```
KM> (the likes-of of *Krabby-Patty)
1 -> (the likes-of of *Krabby-Patty)
1 (Retrieving answer computed and cached earlier:
1 (the likes-of of *Krabby-Patty) = (*Spongebob *Patrick *Sandy *Mr.Krab *Plankton *Pearl)))
1 <- (*Spongebob *Patrick *Sandy *Mr.Krab *Plankton *Pearl) [(the
(*Spongebob *Patrick *Sandy *Mr.Krab *Plankton *Pearl)
```

Who likes Krabby Patty?

4.1 Simple Queries - 3

```
KM> (the family of (the instances of Fish))
1 -> (the family of (the instances of Fish))
2 -> (the instances of Fish)
2 <- (*Patrick *Squidward *Mr.Krab *Plankton *Gary *Pearl) [(the
...
2 -> (the family of (:set *Patrick *Squidward *Mr.Krab *Plankton *Gary *Pearl))
3 -> (the family of *Patrick)
3 (Retrieving answer computed and cached earlier:
3 (the family of *Patrick) = NIL))
3 <- NIL [(the family of *Patrick)]
3 -> (the family of *Squidward)
3 (Retrieving answer computed and cached earlier:
3 (the family of *Squidward) = NIL))
3 <- NIL [(the family of *Squidward)]
3 -> (the family of *Mr.Krab)
3 (Retrieving answer computed and cached earlier:
3 (the family of *Mr.Krab) = (*Pearl)))
3 <- (*Pearl) [(the family of *Mr.Krab)]
3 -> (the family of *Plankton)
3 (Retrieving answer computed and cached earlier:
3 (the family of *Plankton) = NIL))
3 <- NIL [(the family of *Plankton)]
3 -> (the family of *Gary)
3 (Retrieving answer computed and cached earlier:
3 (the family of *Gary) = (*Spongebob)))
3 <- (*Spongebob) [(the family of *Gary)]
3 -> (the family of *Pearl)
3 (Retrieving answer computed and cached earlier:
3 (the family of *Pearl) = (*Mr.Krab)))
3 <- (*Mr.Krab) [(the family of *Pearl)]
2 <- (*Pearl *Spongebob *Mr.Krab) [(the
...
1 <- (*Pearl *Spongebob *Mr.Krab) [(the
...
(*Pearl *Spongebob *Mr.Krab)
```

Who has family?

• 4.1 Simple queries - 4

```
KM> (the employee-of of (the friend of *Patrick))
1 -> (the employee-of of (the friend of *Patrick))
2 -> (the friend of *Patrick)
2 (Retrieving answer computed and cached earlier:
2 (the friend of *Patrick) = (*Spongebob *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl)))
2 <- (*Spongebob *Sandy *Mr.Krab *Squidward *Plankton *Gary
*Pearl) [(the fri...
2 -> (the employee-of of (:set *Spongebob *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl))
3 -> (the employee-of of *Spongebob)
3 (Retrieving answer computed and cached earlier:
3 (the employee-of of *Spongebob) = (*Krusty-Krab)))
3 <- (*Krusty-Krab) [(the
3 -> (the employee-of of *Sandy))
3 (Retrieving answer computed and cached earlier:
3 (the employee-of of *Sandy) = NIL))
3 <- NIL [(the employee-of of *Sandy)]
3 -> (the employee-of of *Mr.Krab)
3 (Retrieving answer computed and cached earlier:
3 (the employee-of of *Mr.Krab) = NIL))
3 <- NIL [(the employee-of of *Mr.Krab)]
3 -> (the employee-of of *Squidward)
3 (Retrieving answer computed and cached earlier:
3 (the employee-of of *Squidward) = (*Krusty-Krab)))
3 <- (*Krusty-Krab) [(the
3 -> (the employee-of of *Plankton))
3 (Retrieving answer computed and cached earlier:
3 (the employee-of of *Plankton) = NIL))
3 <- NIL [(the
3 -> (the employee-of of *Gary))
3 (Retrieving answer computed and cached earlier:
3 (the employee-of of *Gary) = NIL))
3 <- NIL [(the employee-of of *Gary)]
3 -> (the employee-of of *Pearl)
3 (Retrieving answer computed and cached earlier:
3 (the employee-of of *Pearl) = NIL))
3 <- NIL [(the employee-of of *Pearl)]
2 <- (*Krusty-Krab) [(the
1 <- (*Krusty-Krab) ... [(the
(*Krusty-Krab) ...
```

Who employs the friend of Patrick?

4.2 Intermediate queries - 1

```

KM> (allof (the instances of Fish) where ((the number of (the friend of It)) >= (0.7 * (the number of (the friend of *Spongebob)))))
1 -> (allof (the instances of Fish) where ((the number of (the friend of It)) >= (0.7 * (the number of (the friend of *Spongebob)))))
2 -> (forall (the instances of Fish) where ((the number of (the friend of It)) >= (0.7 * (the number of (the friend of *Spongebob))))) It)
3 -> (the instances of Fish)
3 <- (*Patrick *Squidward *Mr.Krab *Plankton *Gary *Pearl _Fish2 _Fish3
_Fish5...
3 -> ((the number of (the friend of *Patrick)) >= (0.7 * (the number of (the friend of *Spongebob)))))
4 -> (the number of (the friend of *Patrick))
5 -> (the friend of *Patrick)
5 (Retrieving answer computed and cached earlier:
5 (the friend of *Patrick) = (*Spongebob *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl)))
5 <- (*Spongebob *Sandy *Mr.Krab *Squidward *Plankton *Gary
*Pearl) [(the fri...
5 -> (the number of (:set *Spongebob *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl))
5 <- (7) [(the
...
4 <- (7) [(the
...
4 -> (0.7 * (the number of (the friend of *Spongebob)))
5 -> ((the number of (the friend of *Spongebob)))
6 -> (the number of (the friend of *Spongebob))
7 -> (the friend of *Spongebob)
7 (Retrieving answer computed and cached earlier:
7 (the friend of *Spongebob) = (*Patrick *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl _Friend6)))
7 <- (*Patrick *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl
_Friend6) [(...
7 -> (the number of (:set *Patrick *Sandy *Mr.Krab *Squidward *Plankton *Gary *Pearl _Friend6))
7 <- (8) [(the
...
6 <- (8) [(the
...
5 <- (8) [((the
...
4 <- (5.6) [(0.7
...
3 <- (t) [((the
...
2 <- (*Patrick *Mr.Krab) [(forall
...
1 <- (*Patrick *Mr.Krab) [(allof
...
(*Patrick *Mr.Krab)

```

Which fish thinks of itself as a famous fish in VIKINI city?
(Find the fish which has friends 0.7 times or more the number of SpongeBob's friends)

4.2 Intermediate queries - 2

Which character has negative personality?

```
KM> (allof (the lives-in-of of (vikini-city))  
      where ((the number of (allof ?x in (the no of (the personality of It))  
                where ((?x) = (Positive)))) = 1))  
(*Squidward *Pearl *Mr.Krab *Plankton)  
(103 inferences and 1542 KB accesses in 0.8 sec [134 lips, 2015 kaps]))
```

4.3 Advanced queries - 1(1)

```
KM> (every Fish has
      (fail ((if ((the salary-want of Self) > 3000)
                  then (*Yes)
                  else (*No)))))
(Fish)
(1 inferences and 26 KB accesses in 0.0 sec [31 lips, 812 kaps]))

KM> (every Animal has
      (fail ((if ((the salary-want of Self) > 3000)
                  then (*Yes)
                  else (*No)))))
(Animal)
(1 inferences and 17 KB accesses in 0.0 sec)

KM> (the fail of *Sandy)
(*Yes)
(5 inferences and 119 KB accesses in 0.0 sec [106 lips, 2531 kaps]))

KM> (the fail of *Patrick)
(*No)
(5 inferences and 107 KB accesses in 0.0 sec [156 lips, 3343 kaps]))

KM> (the fail of *Pearl)
(*No)
(5 inferences and 103 KB accesses in 0.0 sec [161 lips, 3322 kaps]))
```

Mr.Krab is trying to hire an employee.
First, he considers firing if the desired wage of the
applicant exceeds \$3000.

4.3 Advanced queries - 1(2)

```
KM> (allof (the apply-of of (applicant)) where
      ((the number of (the yes of (the personality of It)))
       >= 4))
1 -> (allof (the apply-of of (applicant)) where ((the number of (the yes of (the personality of It))) >= 4))
2 -> (forall (the apply-of of (applicant)) where ((the number of (the yes of (the personality of It))) >= 4) It)
3 -> (the apply-of of (applicant))
4 -> (applicant)
4 <- (applicant)
4 -> (the apply-of of applicant)
4 (1) Local value(s): (:set *Patrick *Sandy *Pearl)
5 -> (:set *Patrick *Sandy *Pearl) [for (the apply-of of applicant)]
5 <- (*Patrick *Sandy *Pearl)
4 <- (*Patrick *Sandy *Pearl)
3 <- (*Patrick *Sandy *Pearl)
3 -> ((the number of (the yes of (the personality of *Patrick))) >= 4)
4 -> (the number of (the yes of (the personality of *Patrick)))
3 -> ((the number of (the yes of (the personality of *Sandy))) >= 4)
4 -> (the number of (the yes of (the personality of *Sandy)))
5 -> (the yes of (the personality of *Sandy))
6 -> (the personality of *Sandy)
6 (1) Local value(s): (a Personality with (yes (Positive Clean Friendly Notdumb)) (no (Notcurious)))
7 -> (a Personality with (yes (Positive Clean Friendly Notdumb)) (no (Notcurious))) [for (the personality of *Sandy)]
7 <- (_Personality9)
6 <- (_Personality9)
6 -> (the yes of _Personality9)
6 (1) Local value(s): (:set Positive Clean Friendly Notdumb)
7 -> (:set Positive Clean Friendly Notdumb) [for (the yes of _Personality9)]
7 <- (Positive Clean Friendly Notdumb)
6 <- (Positive Clean Friendly Notdumb)
5 <- (Positive Clean Friendly Notdumb)
5 -> (the number of (:set Positive Clean Friendly Notdumb))
5 <- (4)
4 <- (4)
3 <- (t)
```

Sandy applied again after contracting wage negotiation with Mr.Krab.

Mr. Krab wants to hire positive, neat, not curious, friendly, and not dumb applicant.

If more than four conditions are met, it will be hired.

4.3 Advanced queries - 1(2)

```
3   -> ((the number of (the yes of (the personality of *Pearl))) >= 4)
4   -> (the number of (the yes of (the personality of *Pearl)))
5   -> (the yes of (the personality of *Pearl))
6   -> (the personality of *Pearl)
6   (1) Local value(s): (a Personality with (yes (Notcurious Friendly Notdumb)) (no (Positive Clean)))
7   -> (a Personality with (yes (Notcurious Friendly Notdumb)) (no (Positive Clean))) [for (the personality of *Pearl)]
7   <- (_Personality10) [a
...
6   <- (_Personality10) [(the personality of *Pearl)]
6   -> (the yes of _Personality10)
6   (1) Local value(s): (:set Notcurious Friendly Notdumb)
7   -> (:set Notcurious Friendly Notdumb) [for (the yes of _Personality10)]
7   <- (Notcurious Friendly Notdumb) [(:set
...
6   <- (Notcurious Friendly Notdumb) [(the yes of _Personality10)]
5   <- (Notcurious Friendly Notdumb) [(the
...
5   -> (the number of (:set Notcurious Friendly Notdumb))
5   <- (3) [(the
...
4   <- (3) [(the
...
3   <- FAIL! [((the
...
2   <- (*Sandy) [(forall
...
1   <- (*Sandy) [(allof
...
(*Sandy)
```

4.3 Advanced queries - 2

Can Plankton and Patrick buy one krabby patty per a day?

```
KM> (every Fish has
      (everyday-Krabby-Patty(
        (if ((the salary of Self) >= (30 * (the price of (the favorite of (the friend of *Spongebob))))))
        then (*Yes)
        else (*No))))))
(Fish)
(1 inferences and 26 KB accesses in 0.0 sec [62 lips, 1625 kaps]))

KM> (the everyday-Krabby-Patty of *Patrick)
(*No)
(33 inferences and 1563 KB accesses in 0.9 sec [37 lips, 1786 kaps]))

KM> (the everyday-Krabby-Patty of *Plankton)
(*Yes)
(20 inferences and 128 KB accesses in 0.1 sec [212 lips, 1361 kaps]))
```

5. Extensibility

- We only covered a small number of characters of Spongebob SquarePants and dealt with a very small part of the relationship.
- We can do some similar queries using other characteristic of characters (ex. Appearance)
- New characters and restaurants can easily be adopted.

Thank you for listening

