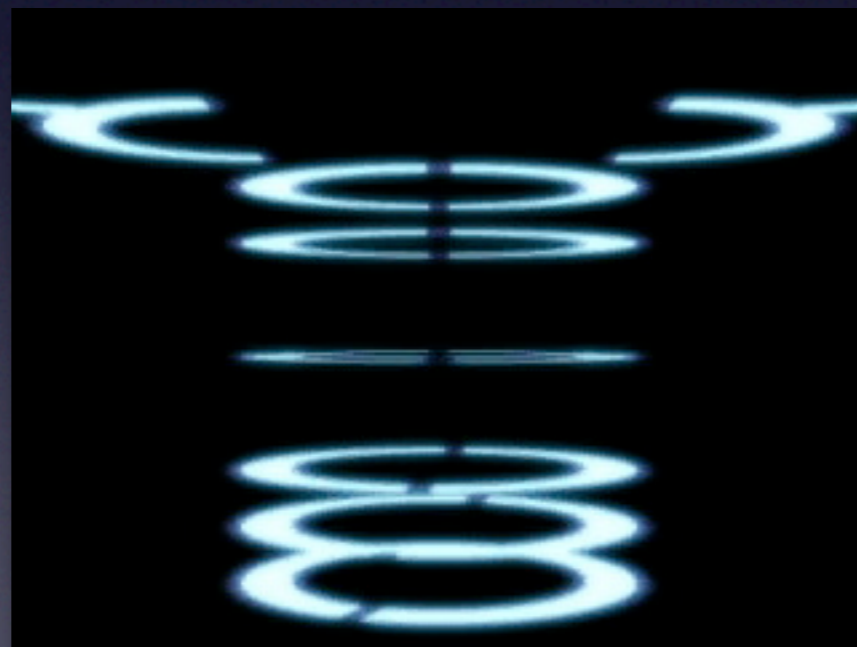




# JPS

ćwiczenia 4.

## Złożone zapytania



Operator group as

Operator group as

$q$  group as  $n$

group as  $n$

|  
 $q$

# Operator group as

**q group as n**

group as n  
|  
q

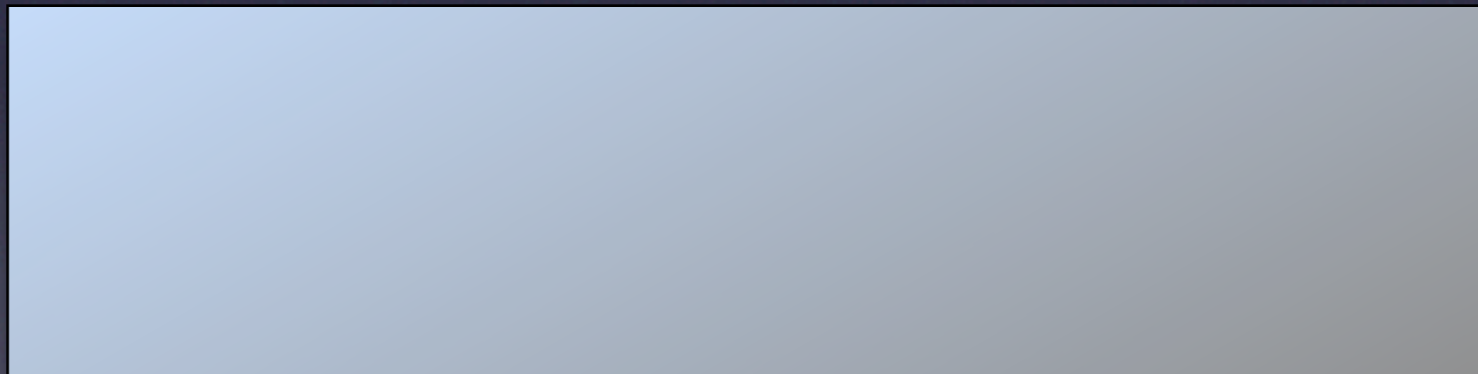
- Wykonać **eval(q)** i zrobić **qres = QRES.POP()**
- Dla całego qres utworzyć binder **<n, qres>** (inny zapis: **n(qres)**) i wrzucić go na QRES

# Operator group as

**q group as n**

group as n  
|  
q

- Wykonać `eval(q)` i zrobić `qres = QRES.POP()`
- Dla całego `qres` utworzyć binder `<n, qres>` (inny zapis: `n(qres)`) i wrzucić go na QRES



# Operator group as

**q group as n**

group as n  
|  
q

- Wykonać `eval(q)` i zrobić `qres = QRES.POP()`
- Dla całego `qres` utworzyć binder `<n, qres>` (inny zapis: `n(qres)`) i wrzucić go na QRES

Przykłady:

# Operator group as

**q group as n**

group as n  
|  
q

- Wykonać `eval(q)` i zrobić `qres = QRES.POP()`
- Dla całego `qres` utworzyć binder `<n, qres>` (inny zapis: `n(qres)`) i wrzucić go na QRES

Przykłady:

| group as nazwa = nazwa(l)



# Operator group as

**q group as n**

group as n  
|  
q

- Wykonać `eval(q)` i zrobić `qres = QRES.POP()`
- Dla całego `qres` utworzyć binder `<n, qres>` (inny zapis: `n(qres)`) i wrzucić go na QRES

Przykłady:

`l group as nazwa = nazwa(l)`

`bag(1, 2, 3) group as nazwa = nazwa(bag(1, 2, 3))`

# Operator group as

**q group as n**

group as n  
|  
q

- Wykonać `eval(q)` i zrobić `qres = QRES.POP()`
- Dla całego `qres` utworzyć binder `<n, qres>` (inny zapis: `n(qres)`) i wrzucić go na QRES

Przykłady:

```
l group as nazwa = nazwa(l)
bag(1, 2, 3) group as nazwa = nazwa(bag(1, 2, 3))
emp group as nazwa = nazwa(bag(i1, i2, i3))
```

Operator as

Operator as

q as n

as n  
|  
q

# Operator as

**q as n**

as n  
|  
q

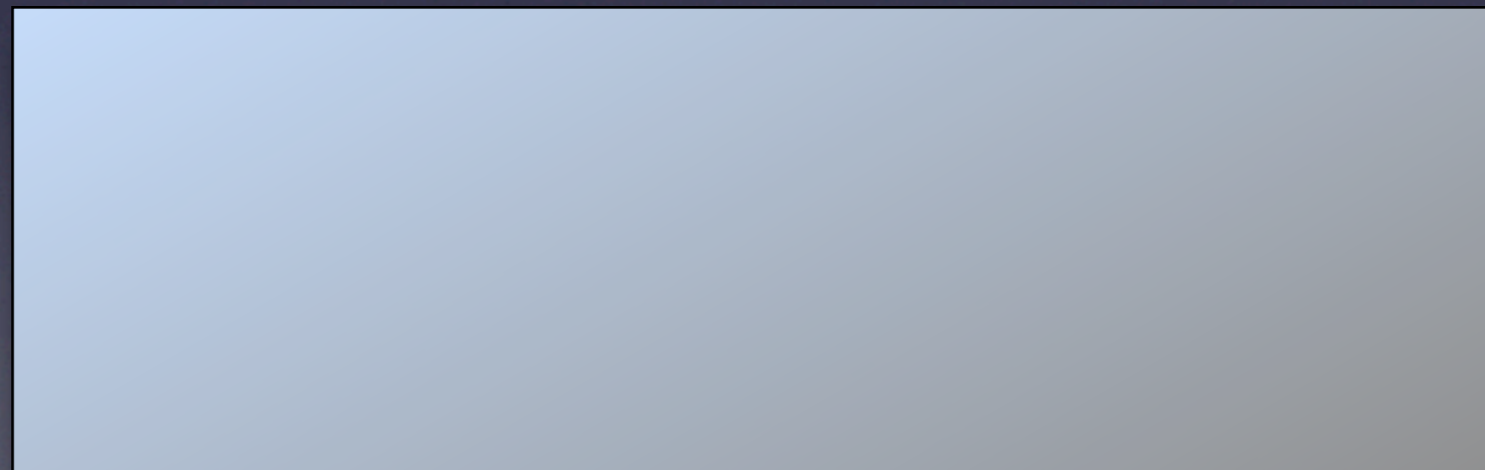
- Wykonać **eval**(q) i zrobić  $qres = QRES.POP()$
- Dla każdego  $x \in qres$  utworzyć binder  $\langle n, x \rangle$
- Bindery wynikowe trafiają do бага, którego umieszczamy na QRES

# Operator as

**q as n**

as n  
|  
q

- Wykonać **eval(q)** i zrobić `qres = QRES.POP()`
- Dla każdego  $x \in \text{qres}$  utworzyć binder  $\langle n, x \rangle$
- Bindery wynikowe trafiają do бага, którego umieszczamy na QRES



# Operator as

**q as n**

as n  
|  
q

- Wykonać **eval(q)** i zrobić  $qres = QRES.POP()$
- Dla każdego  $x \in qres$  utworzyć binder  $\langle n, x \rangle$
- Bindery wynikowe trafiają do бага, którego umieszczamy na QRES

Przykłady:

# Operator as

**q as n**

as n  
|  
q

- Wykonać **eval(q)** i zrobić `qres = QRES.POP()`
- Dla każdego  $x \in qres$  utworzyć binder  $\langle n, x \rangle$
- Bindery wynikowe trafiają do `baga`, którego umieszczamy na QRES

Przykłady:

`l as nazwa = nazwa(l)`



# Operator as

**q as n**

as n  
|  
q

- Wykonać **eval(q)** i zrobić `qres = QRES.POP()`
- Dla każdego  $x \in qres$  utworzyć binder  $\langle n, x \rangle$
- Bindery wynikowe trafiają do `baga`, którego umieszczamy na QRES

Przykłady:

`l as nazwa = nazwa(l)`

`bag(1, 2, 3) as nazwa = bag(nazwa(1), nazwa(2), nazwa(3))`

# Operator as

**q as n**

as n  
|  
q

- Wykonać **eval(q)** i zrobić **qres = QRES.POP()**
- Dla każdego  $x \in \text{qres}$  utworzyć binder  $\langle n, x \rangle$
- Bindery wynikowe trafiają do **baga**, którego umieszczamy na QRES

Przykłady:

```
l as nazwa = nazwa(l)
bag(l, 2, 3) as nazwa = bag(nazwa(l), nazwa(2), nazwa(3))
((l as nazwa1) as nazwa2) as nazwa3 = nazwa3(nazwa2(nazwa1(l)))
```

# Operator as

**q as n**

as n  
|  
q

- Wykonać **eval(q)** i zrobić **qres = QRES.POP()**
- Dla każdego  $x \in \text{qres}$  utworzyć binder  $\langle n, x \rangle$
- Bindery wynikowe trafiają do **baga**, którego umieszczamy na QRES

Przykłady:

```
l as nazwa = nazwa(l)
bag(l, 2, 3) as nazwa = bag(nazwa(l), nazwa(2), nazwa(3))
((l as nazwa1) as nazwa2) as nazwa3 = nazwa3(nazwa2(nazwa1(l)))
struct(l, 2) as nazwa = nazwa(struct(l, 2))
```

# Operator as

**q as n**

as n  
|  
q

- Wykonać **eval(q)** i zrobić **qres = QRES.POP()**
- Dla każdego  $x \in \text{qres}$  utworzyć binder  $\langle n, x \rangle$
- Bindery wynikowe trafiają do **baga**, którego umieszczamy na QRES

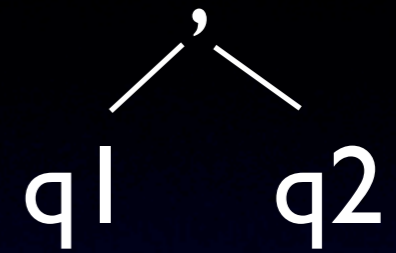
Przykłady:

```
l as nazwa = nazwa(l)
bag(l, 2, 3) as nazwa = bag(nazwa(l), nazwa(2), nazwa(3))
((l as nazwa1) as nazwa2) as nazwa3 = nazwa3(nazwa2(nazwa1(l)))
struct(l, 2) as nazwa = nazwa(struct(l, 2))
emp as nazwa = bag(nazwa(i1), nazwa(i2), nazwa(i3))
```

Operator ,

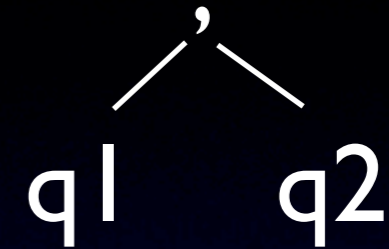
Operator ,

**q1, q2**



# Operator ,

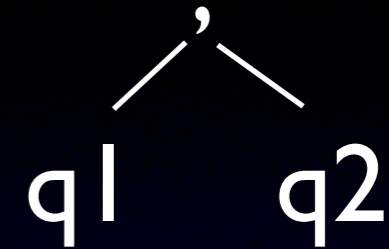
**q1, q2**



- Wykonać **eval(q1)** i **eval(q2)**
- Zrobić  $q2res = QRES.POP()$ ,  
 $q1res = QRES.POP()$
- Wykonać  $q1res \times q2res$
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą baga, którego umieszczamy na QRES

# Operator ,

**q1, q2**



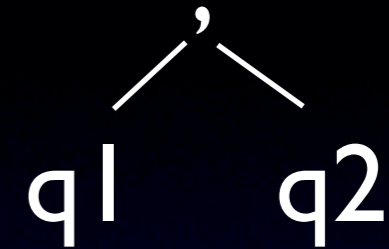
- Wykonać **eval(q1)** i **eval(q2)**
- Zrobić  $q2res = QRES.POP()$ ,  
 $q1res = QRES.POP()$
- Wykonać  $q1res \times q2res$
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES





# Operator ,

**q1, q2**



- Wykonać **eval(q1)** i **eval(q2)**
- Zrobić **q2res = QRES.POP()**,  
**q1res = QRES.POP()**
- Wykonać **q1res x q2res**
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

Przykłady:

# Operator ,

**q1, q2**

- Wykonać `eval(q1)` i `eval(q2)`
- Zrobić `q2res = QRES.POP()`,  
`q1res = QRES.POP()`
- Wykonać `q1res x q2res`
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

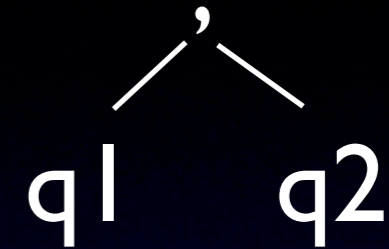


Przykłady:

`1, 2 = bag(struct(1, 2))`

# Operator ,

**q1, q2**



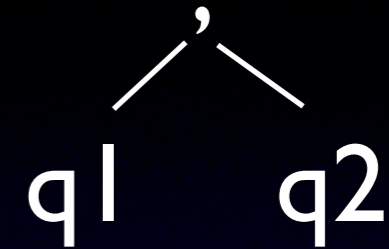
- Wykonać `eval(q1)` i `eval(q2)`
- Zrobić `q2res = QRES.POP()`,  
`q1res = QRES.POP()`
- Wykonać `q1res x q2res`
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

Przykłady:

```
1, 2 = bag(struct(1, 2))  
bag(1, 2), 3 = bag(struct(1, 3),  
                    struct(2, 3))
```

# Operator ,

**q1, q2**



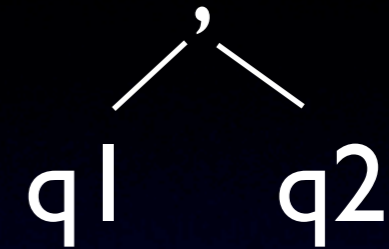
- Wykonać `eval(q1)` i `eval(q2)`
- Zrobić `q2res = QRES.POP()`,  
`q1res = QRES.POP()`
- Wykonać `q1res x q2res`
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

Przykłady:

```
1, 2 = bag(struct(1, 2))
bag(1, 2), 3 = bag(struct(1, 3),
                  struct(2, 3))
bag(1, 2), bag(3, 4) = bag(struct(1, 3),
                          struct(1, 4),
                          struct(2, 3),
                          struct(2, 4))
```

# Operator ,

**q1, q2**



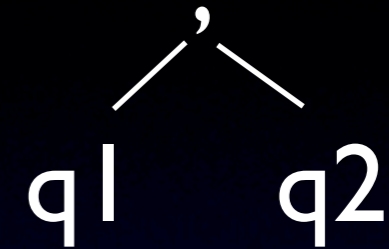
- Wykonać `eval(q1)` i `eval(q2)`
- Zrobić `q2res = QRES.POP()`,  
`q1res = QRES.POP()`
- Wykonać `q1res x q2res`
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

Przykłady:

```
1, 2 = bag(struct(1, 2))
bag(1, 2), 3 = bag(struct(1, 3),
                    struct(2, 3))
bag(1, 2), bag(3, 4) = bag(struct(1, 3),
                           struct(1, 4),
                           struct(2, 3),
                           struct(2, 4))
struct(1, 2), 3 = struct(1, 2, 3)
```

# Operator ,

**q1, q2**



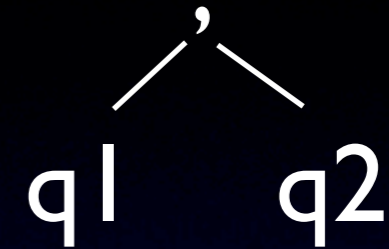
- Wykonać `eval(q1)` i `eval(q2)`
- Zrobić `q2res = QRES.POP()`,  
`q1res = QRES.POP()`
- Wykonać `q1res x q2res`
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

Przykłady:

```
1, 2 = bag(struct(1, 2))
bag(1, 2), 3 = bag(struct(1, 3),
                    struct(2, 3))
bag(1, 2), bag(3, 4) = bag(struct(1, 3),
                           struct(1, 4),
                           struct(2, 3),
                           struct(2, 4))
struct(1, 2), 3 = struct(1, 2, 3)
struct(1, 2), bag(3, 4) = bag(struct(1, 2, 3),
                              struct(1, 2, 4))
```

# Operator ,

**q1, q2**



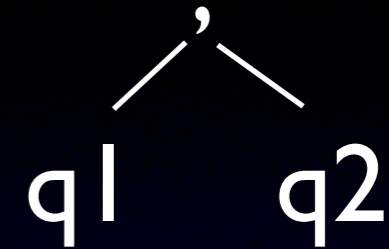
- Wykonać `eval(q1)` i `eval(q2)`
- Zrobić `q2res = QRES.POP()`,  
`q1res = QRES.POP()`
- Wykonać `q1res x q2res`
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

Przykłady:

```
1, 2 = bag(struct(1, 2))
bag(1, 2), 3 = bag(struct(1, 3),
                  struct(2, 3))
bag(1, 2), bag(3, 4) = bag(struct(1, 3),
                          struct(1, 4),
                          struct(2, 3),
                          struct(2, 4))
struct(1, 2), 3 = struct(1, 2, 3)
struct(1, 2), bag(3, 4) = bag(struct(1, 2, 3),
                              struct(1, 2, 4))
1, 2, 3, 4 = bag(struct(1, 2, 3, 4))
```

# Operator ,

**q1, q2**



- Wykonać `eval(q1)` i `eval(q2)`
- Zrobić `q2res = QRES.POP()`,  
`q1res = QRES.POP()`
- Wykonać `q1res x q2res`
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

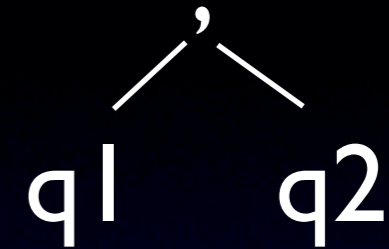
Przykłady:

```
1, 2 = bag(struct(1, 2))
bag(1, 2), 3 = bag(struct(1, 3),
                    struct(2, 3))
bag(1, 2), bag(3, 4) = bag(struct(1, 3),
                           struct(1, 4),
                           struct(2, 3),
                           struct(2, 4))
struct(1, 2), 3 = struct(1, 2, 3)
struct(1, 2), bag(3, 4) = bag(struct(1, 2, 3),
                              struct(1, 2, 4))
1, 2, 3, 4 = bag(struct(1, 2, 3, 4))
bag(1, 2), 3, 4 = bag(struct(1, 3, 4),
                     struct(2, 3, 4))
```



# Operator ,

**q1, q2**



- Wykonać `eval(q1)` i `eval(q2)`
- Zrobić `q2res = QRES.POP()`,  
`q1res = QRES.POP()`
- Wykonać `q1res x q2res`
- Każda para iloczynu kartezjańskiego tworzy nową strukturę.  
Jeśli  $x \in q1res$  lub  $y \in q1res$  jest strukturą, to tylko rozszerzamy tę strukturę
- Wszystkie struktury tworzą бага, którego umieszczamy na QRES

Przykłady:

```
1, 2 = bag(struct(1, 2))
bag(1, 2), 3 = bag(struct(1, 3),
                    struct(2, 3))
bag(1, 2), bag(3, 4) = bag(struct(1, 3),
                           struct(1, 4),
                           struct(2, 3),
                           struct(2, 4))
struct(1, 2), 3 = struct(1, 2, 3)
struct(1, 2), bag(3, 4) = bag(struct(1, 2, 3),
                              struct(1, 2, 4))
1, 2, 3, 4 = bag(struct(1, 2, 3, 4))
bag(1, 2), 3, 4 = bag(struct(1, 3, 4),
                     struct(2, 3, 4))
```

Przykładowe zastosowanie: podaj nazwiska pracowników razem z ich pensjami: `emp.(name, salary)`

# Operator join

# Operator join

**q1 join q2**



# Operator join

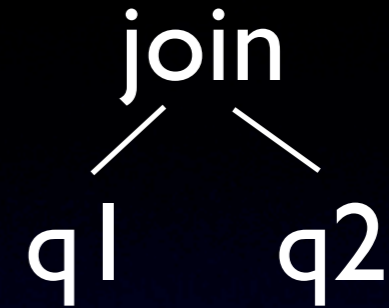
## q1 join q2



- Zainicjalizować `joinres = bag()`
- Wykonać `eval(q1)` i zrobić `q1res = QRES.POP()`
- Dla każdego elementu  $x \in q1res$  wykonać:
  - Utworzyć nową sekcję na ENVS
  - Wykonać `nested(x)`. Wynik wprowadzić do sekcji utworzonej w poprzednim kroku
  - Wykonać `eval(q2)` i zrealizować `q2res = QRES.POP()`
  - Wykonać operację  $x, q2res$ , a wynik dodać do `joinres`
  - Zamknąć sekcję ENVS
- Wykonać `QRES.PUSH(joinres)`

# Operator join

## q1 join q2



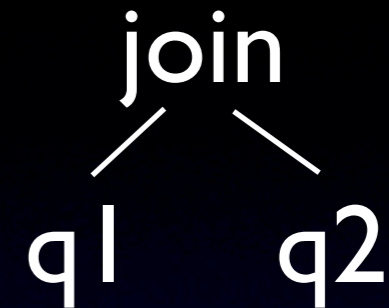
- Zainicjalizować `joinres = bag()`
- Wykonać `eval(q1)` i zrobić `q1res = QRES.POP()`
- Dla każdego elementu  $x \in q1res$  wykonać:
  - Utworzyć nową sekcję na ENVS
  - Wykonać `nested(x)`. Wynik wprowadzić do sekcji utworzonej w poprzednim kroku
  - Wykonać `eval(q2)` i zrealizować `q2res = QRES.POP()`
  - Wykonać operację  $x, q2res$ , a wynik dodać do `joinres`
  - Zamknąć sekcję ENVS
- Wykonać `QRES.PUSH(joinres)`



# Operator join

## q1 join q2

- Zainicjalizować `joinres = bag()`
- Wykonać `eval(q1)` i zrobić `q1res = QRES.POP()`
- Dla każdego elementu  $x \in q1res$  wykonać:
  - Utworzyć nową sekcję na ENVS
  - Wykonać `nested(x)`. Wynik wprowadzić do sekcji utworzonej w poprzednim kroku
  - Wykonać `eval(q2)` i zrealizować `q2res = QRES.POP()`
  - Wykonać operację  $x, q2res$ , a wynik dodać do `joinres`
  - Zamknąć sekcję ENVS
- Wykonać `QRES.PUSH(joinres)`



Przykład:

# Operator join

## q1 join q2

- Zainicjalizować `joinres = bag()`
- Wykonać `eval(q1)` i zrobić `q1res = QRES.POP()`
- Dla każdego elementu  $x \in q1res$  wykonać:
  - Utworzyć nową sekcję na ENVS
  - Wykonać `nested(x)`. Wynik wprowadzić do sekcji utworzonej w poprzednim kroku
  - Wykonać `eval(q2)` i zrealizować `q2res = QRES.POP()`
  - Wykonać operację  $x, q2res$ , a wynik dodać do `joinres`
  - Zamknąć sekcję ENVS
- Wykonać `QRES.PUSH(joinres)`



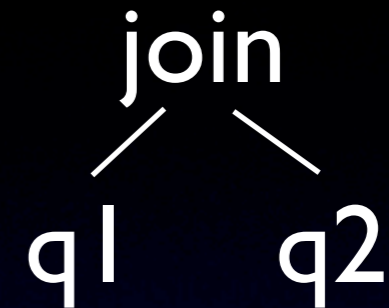
Przykład:

`1 join 2 = struct(1, 2)`

# Operator join

## q1 join q2

- Zainicjalizować `joinres = bag()`
- Wykonać `eval(q1)` i zrobić `q1res = QRES.POP()`
- Dla każdego elementu  $x \in q1res$  wykonać:
  - Utworzyć nową sekcję na ENVS
  - Wykonać `nested(x)`. Wynik wprowadzić do sekcji utworzonej w poprzednim kroku
  - Wykonać `eval(q2)` i zrealizować `q2res = QRES.POP()`
  - Wykonać operację  $x, q2res$ , a wynik dodać do `joinres`
  - Zamknąć sekcję ENVS
- Wykonać `QRES.PUSH(joinres)`



Przykład:

`1 join 2 = struct(1, 2)`

`1 join struct(2, 3) = struct(1, 2, 3)`



# Operator join

## q1 join q2

- Zainicjalizować `joinres = bag()`
- Wykonać `eval(q1)` i zrobić `q1res = QRES.POP()`
- Dla każdego elementu  $x \in q1res$  wykonać:
  - Utworzyć nową sekcję na ENVS
  - Wykonać `nested(x)`. Wynik wprowadzić do sekcji utworzonej w poprzednim kroku
  - Wykonać `eval(q2)` i zrealizować `q2res = QRES.POP()`
  - Wykonać operację  $x, q2res$ , a wynik dodać do `joinres`
  - Zamknąć sekcję ENVS
- Wykonać `QRES.PUSH(joinres)`



Przykład:

`1 join 2 = struct(1, 2)`

`1 join struct(2, 3) = struct(1, 2, 3)`

Przykładowe zastosowanie: podaj dane pracowników, razem z działami w których pracują: `emp join works_in.dept`.

# Operator join

## q1 join q2



- Zainicjalizować `joinres = bag()`
- Wykonać `eval(q1)` i zrobić `q1res = QRES.POP()`
- Dla każdego elementu  $x \in q1res$  wykonać:
  - Utworzyć nową sekcję na ENVS
  - Wykonać `nested(x)`. Wynik wprowadzić do sekcji utworzonej w poprzednim kroku
  - Wykonać `eval(q2)` i zrealizować `q2res = QRES.POP()`
  - Wykonać operację  $x, q2res$ , a wynik dodać do `joinres`
  - Zamknąć sekcję ENVS
- Wykonać `QRES.PUSH(joinres)`

Przykład:

```
l join 2 = struct(1, 2)
```

```
l join struct(2, 3) = struct(1, 2, 3)
```

Przykładowe zastosowanie: podaj dane pracowników, razem z działami w których pracują: `emp join works_in.dept`.  
Ten problem można też rozwiązać za pomocą `as`: `(emp as x).(x, x.works_in.dept)`



Ćwiczenia