

Pomost pomiędzy obiektowymi bazami danych ODRA i Versant

Jakub Siwek

Plan prezentacji:

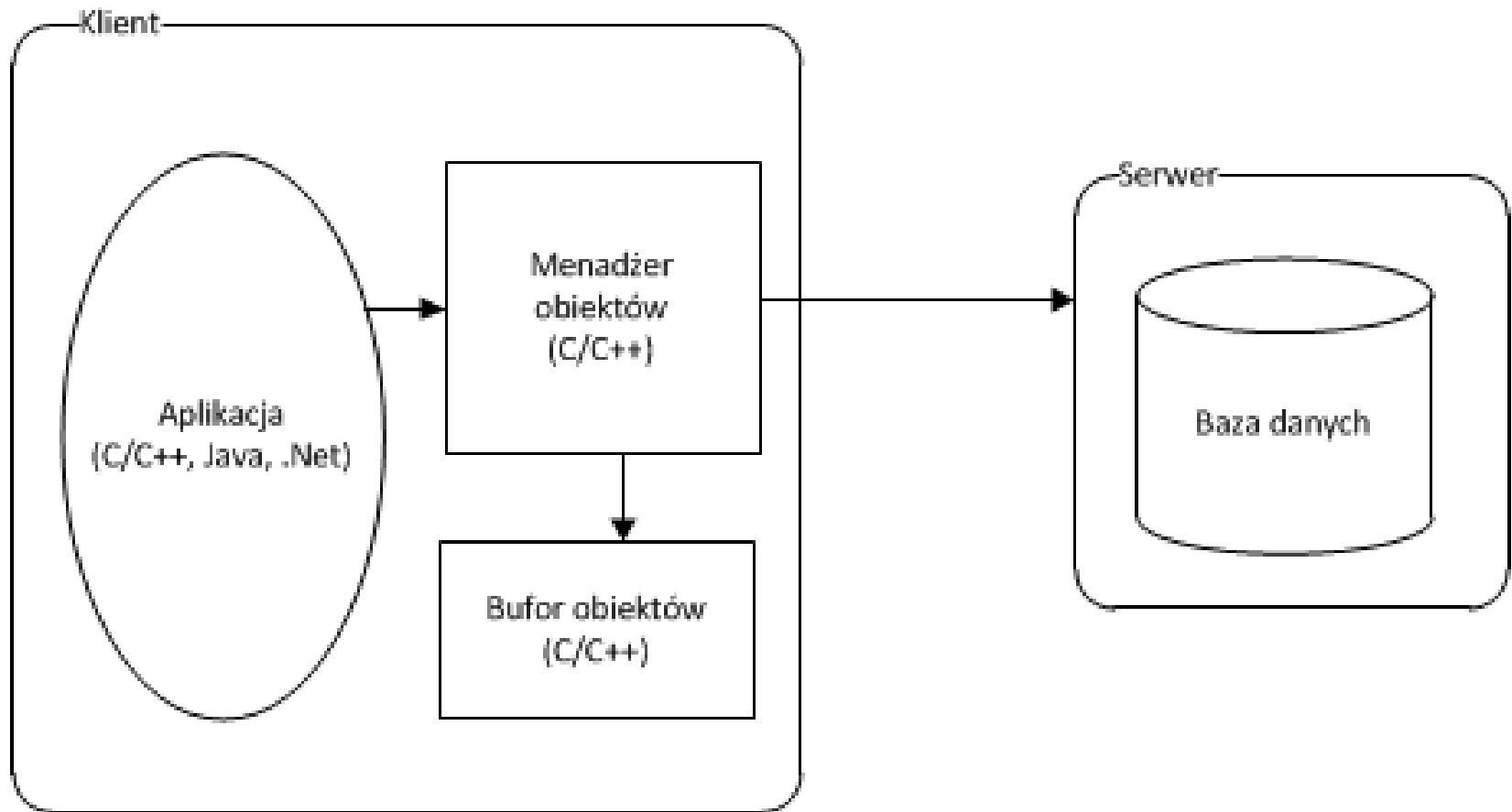
- ▶ Obiektowa baza danych Versant
- ▶ Pobieranie opisu danych i jego import do ODRA
- ▶ Przetwarzanie zapytań SBQL



Obiektowa baza Versant

Metody dostępu i VQL

Architektura Versant



Metody dostępy z języka Java

Interfejsy specyficzne dla VOD:

- ▶ Fundamental JVI
- ▶ Transparent JVI

Wspierane standardy:

- ▶ Java Data Objects (JDO)
- ▶ ODMG 2.0 Java binding specification



Fundamental JVI

- ▶ Wszystko jest uchwyttem (ang. Handle)

```
Handle handleFromString = session.newHandle  
(loidAsString);
```

```
AttrString desc = session.newAttrString ("description");  
handle.put (desc, "has a lean and hungry look");
```

```
handle.deleteObject ();
```



Fundamental JVI - Zapytania

```
FundQuery query = new FundQuery(session_,
    "select selfoid from test.model.Order where date_ > $date");

query.bind("date", new java.sql.Date(105, 3, 10));

FundQueryResult result = query.execute();

Handle hnd;
int count = 0;
while ((hnd = result.next()) != null) {
    count++;
}

query.close();
```



Transparent JVI

```
class Person
{
    String name;
    int age;
}
class Student extends Person
{
    double gpa;
    Course[ ] courses;
    char[ ] grades;
}
```

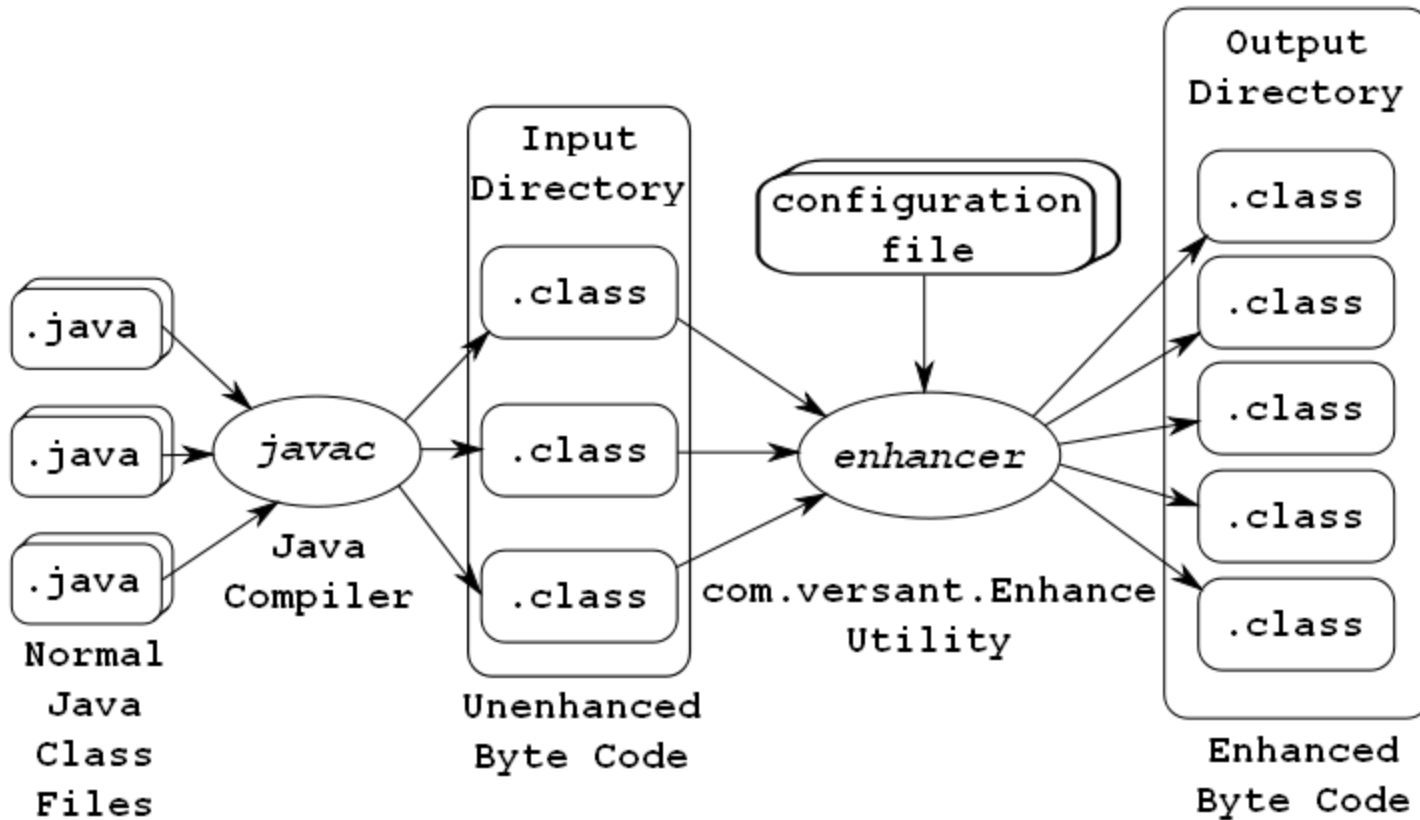


Transparent JVI

```
Student student = new Student ();  
student.name = "John Q. Student";  
student.age = 22;  
student.gpa = 3.4;  
student.courses = new Course [5];  
student.grades = new char [5];  
session.makePersistent (student);
```



Transparent JVITransparent JVI



Transparent JVI

===== CLASS 'Person' =====

superclasses:

'com.versant.trans.CapableWithHash'

attributes:

_vj_hashCode : o_4b

age : o_4b

name : char[]

===== CLASS 'Student, =====

superclasses:

'Person'

attributes:

_vj_hashCode : o_4b

age : o_4b

name : char[]

grades : o_2b[]

courses -> (NULL_DOMAIN)

gpa : o_double



Transparent JVI - Zapytania

```
Query query = new Query(session_,  
„select selfoid from test.model.Order where date > $date");
```

```
query.bind("date", new java.sql.Date(105, 3, 10));
```

```
QueryResult result = query.execute());
```

```
Object obj;
```

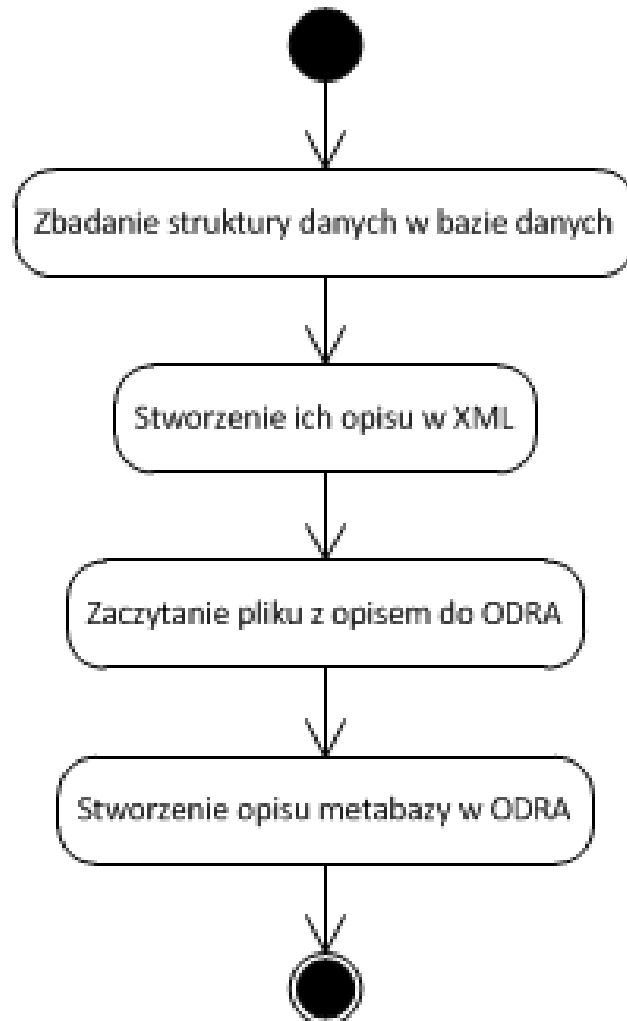
```
while ((obj = result.next()) != null) {  
    System.out.println("Result : " + obj);  
}  
query.close();
```



Pobieranie opisu danych

Konwersja typów, opis obiektów w XML, import do ODRA

Pobranie opisu danych



XML opisujący typy

- ▶ Nazwa
 - ▶ Lista super typów
 - ▶ Lista sub typów
 - ▶ Lista atrybutów
 - ▶ Nazwa
 - ▶ Typ
 - ▶ Czy jest tablicą
- ```
<!DOCTYPE CLASSES [

 <!ELEMENT CLASSES (CLASS*)>
 <!ELEMENT CLASS
 (NAME,ATTRIBUTES*,SUPERCLASSES*,SUB
 CLASSES*)>
 <!ELEMENT NAME (#PCDATA)>
 <!ELEMENT ATTRIBUTES (ATTRIBUTE+)>
 <!ELEMENT ATTRIBUTE (NAME,TYPE)>
 <!ELEMENT TYPE (#PCDATA)>
 <!ELEMENT SUPERCLASSES (NAME+)>
 <!ELEMENT SUBCLASSES (NAME+)>

 <!ATTLIST ATTRIBUTE isarray
 (yes|no|unknown) #REQUIRED>

]>
```



# Konwersja typów podstawowych

| Versant  | ODRA    |
|----------|---------|
| char     | STRING  |
| o_date   | DATE    |
| o_bool   | BOOLEAN |
| o_float  | REAL    |
| o_double | REAL    |
| o_1b     | INTEGER |
| o_u1b    | INTEGER |
| o_2b     | INTEGER |
| o_u2b    | INTEGER |
| o_4b     | INTEGER |
| o_u4b    | INTEGER |
| o_8b     | INTEGER |
| o_u8b    | INTEGER |





# XML opisujący typy c.d.

---

```
<?xml version="1.0"?>
<classes>
 <class>
 <name>SimpleClassName</name>
 <attributes>
 <attribute isarray="no">
 <name>attr</name>
 <type>string</type>
 </attribute>
 </attributes>
 <superclasses>
 <name>SuperClass</name>
 </superclasses>
 <subclasses>
 <name>SubClass</name>
 </subclasses>
 </class>
</classes>
```



# Jakie informacje można znaleźć o typie?

---

- ▶ Nazwa
- ▶ Lista super typów
- ▶ Lista sub typów
- ▶ Atrybuty
  - ▶ Nazwa
  - ▶ Typ (jeżeli jest to typ prosty)



# Jakich informacji nie można znaleźć?

---

- ▶ Czy atrybut jest tablicą?
- ▶ Jeżeli atrybut jest typem złożonym, to jakim?

??? Tablica ??? Jaki typ złożony ???



# Załadowanie opisu do ODRA

---

```
add module versant as versantwrapper on
127.0.0.1:5019 /home/kewis/versantSchema.xml
```

```
compile versant
```

```
cm versant
```

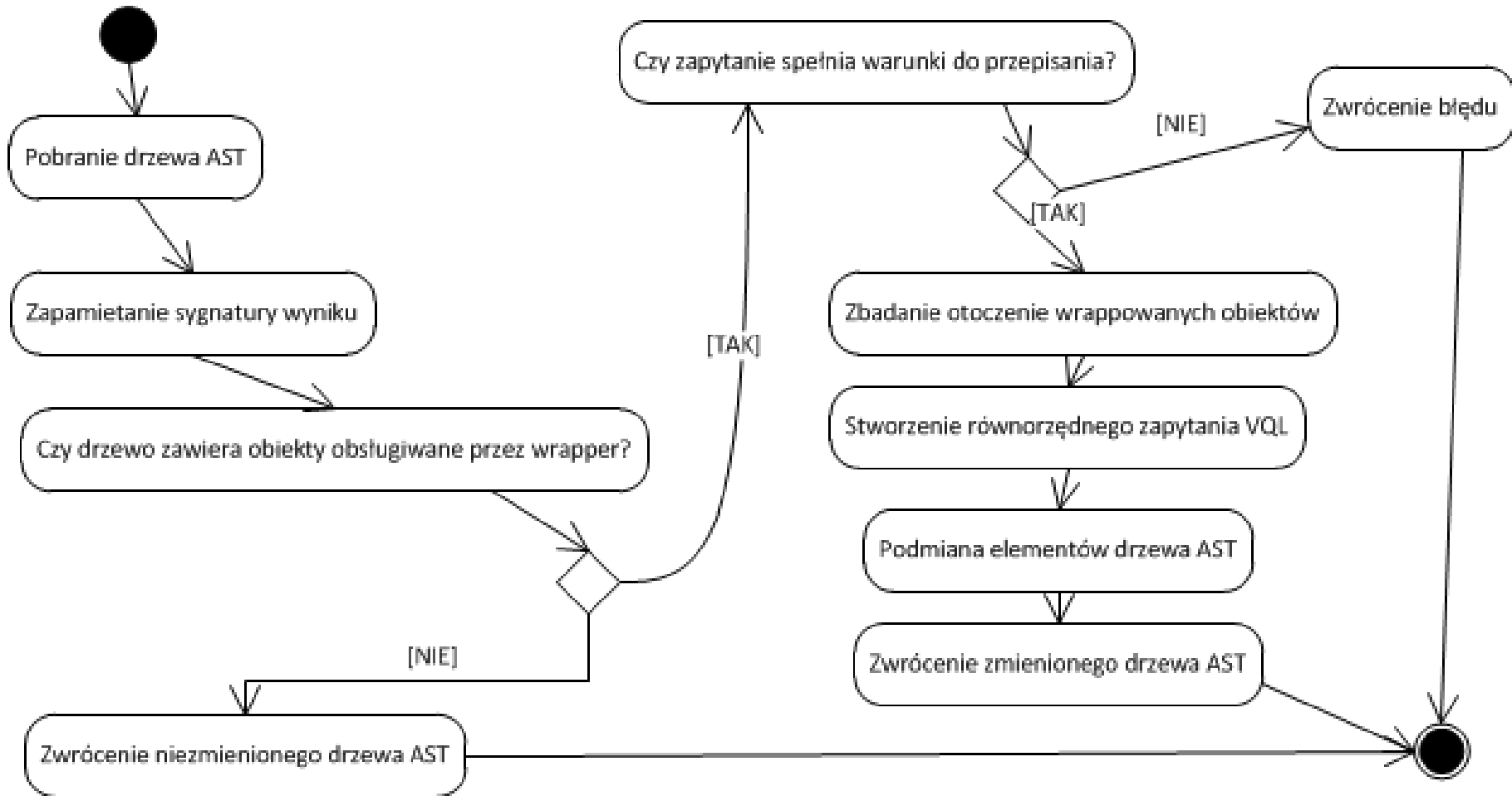
```
set optimization none | versantrewrite
```



# Przetwarzanie zapytań SBQL

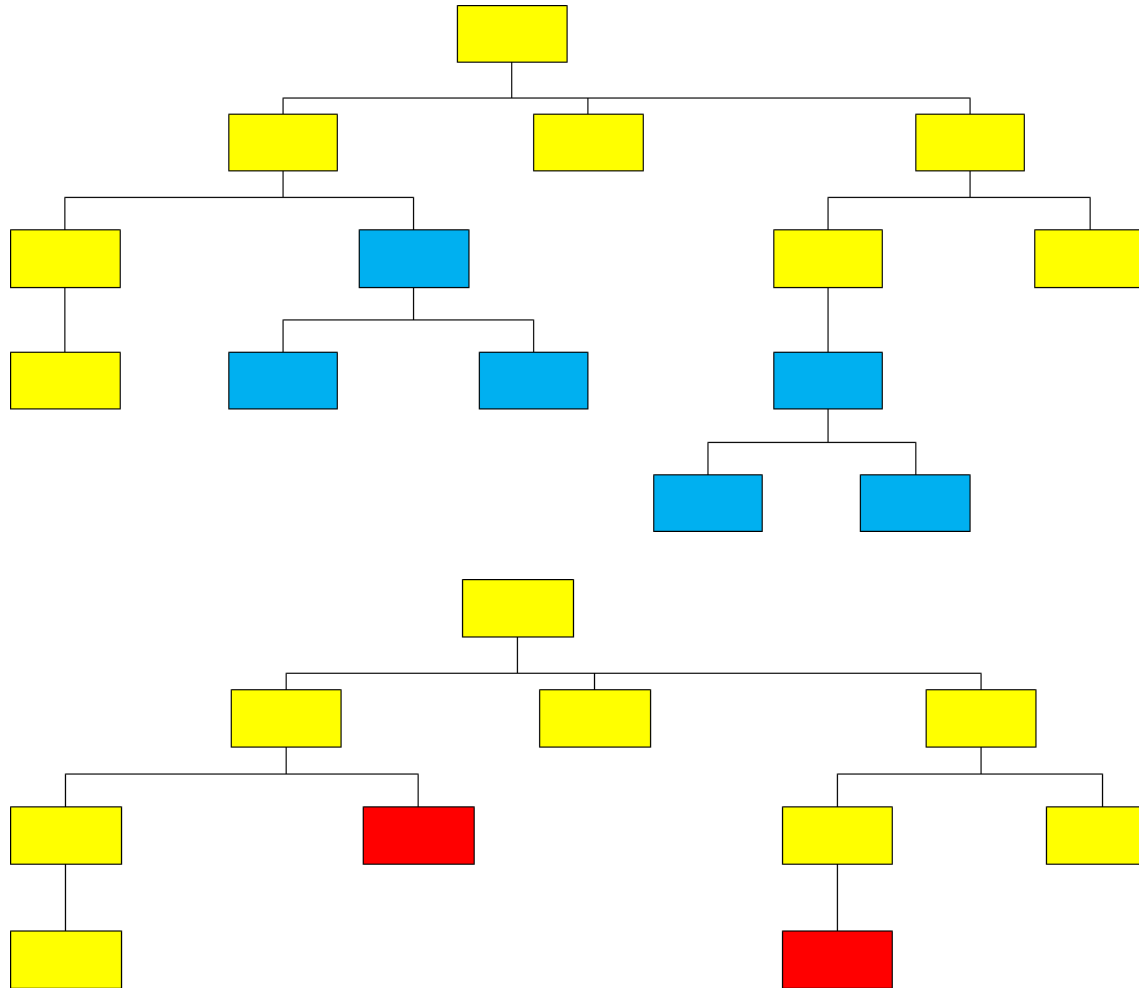
„Optymalizacja” zapytań przez wrappery, konwersja drzewa AST do VQL

# „Optymalizacja” zapytań przez wrapper



# „Optymalizacja” zapytań przez wrapper

---



# „Optymalizacja” zapytań przez wrapper

---

(Student where year > 3).surname

((execVqlExpression(„select selfoid from Student”)) where year > 3).surname

(execVqlExpression(„select selfoid from Student where year > 3”)).surname

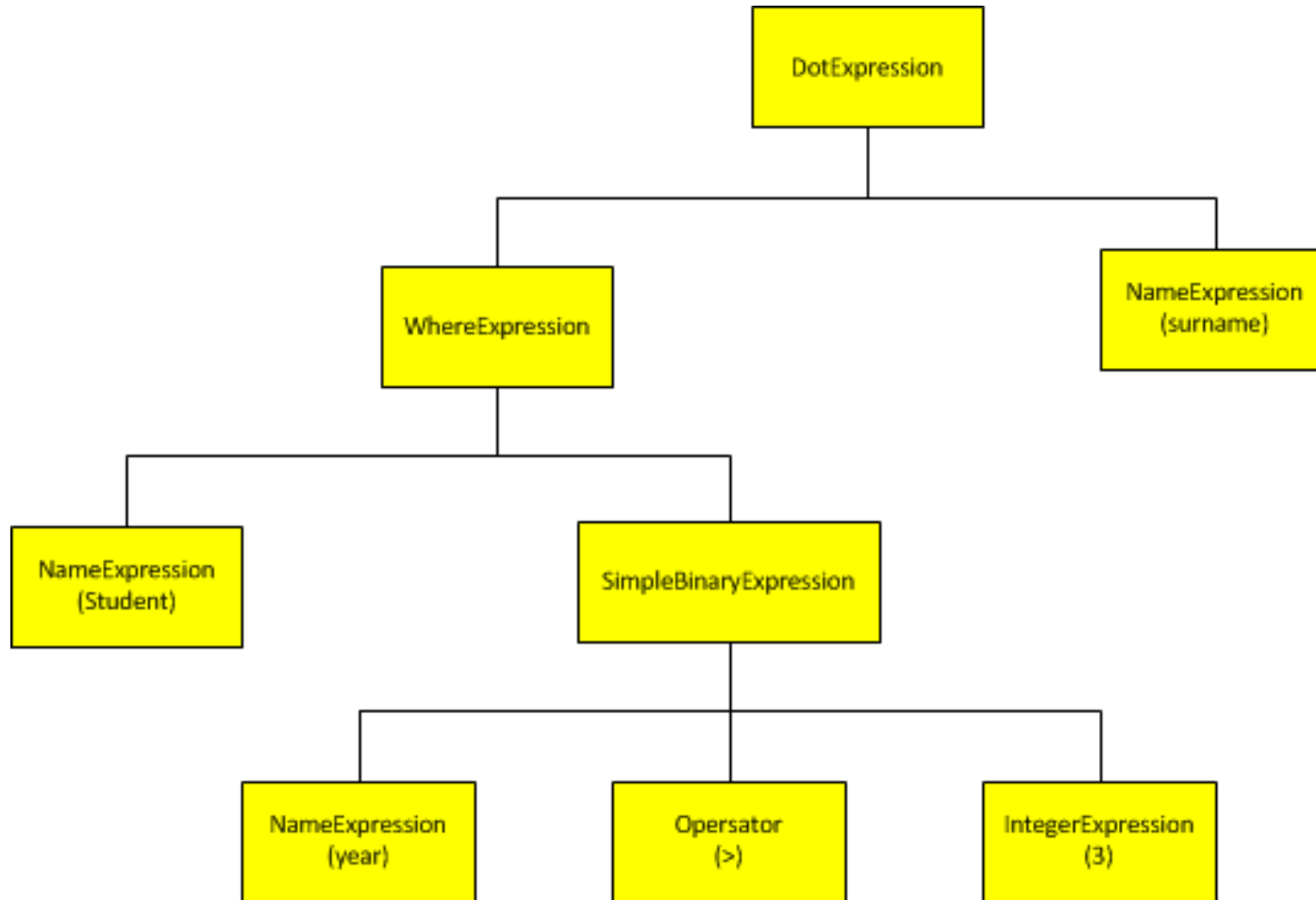
---



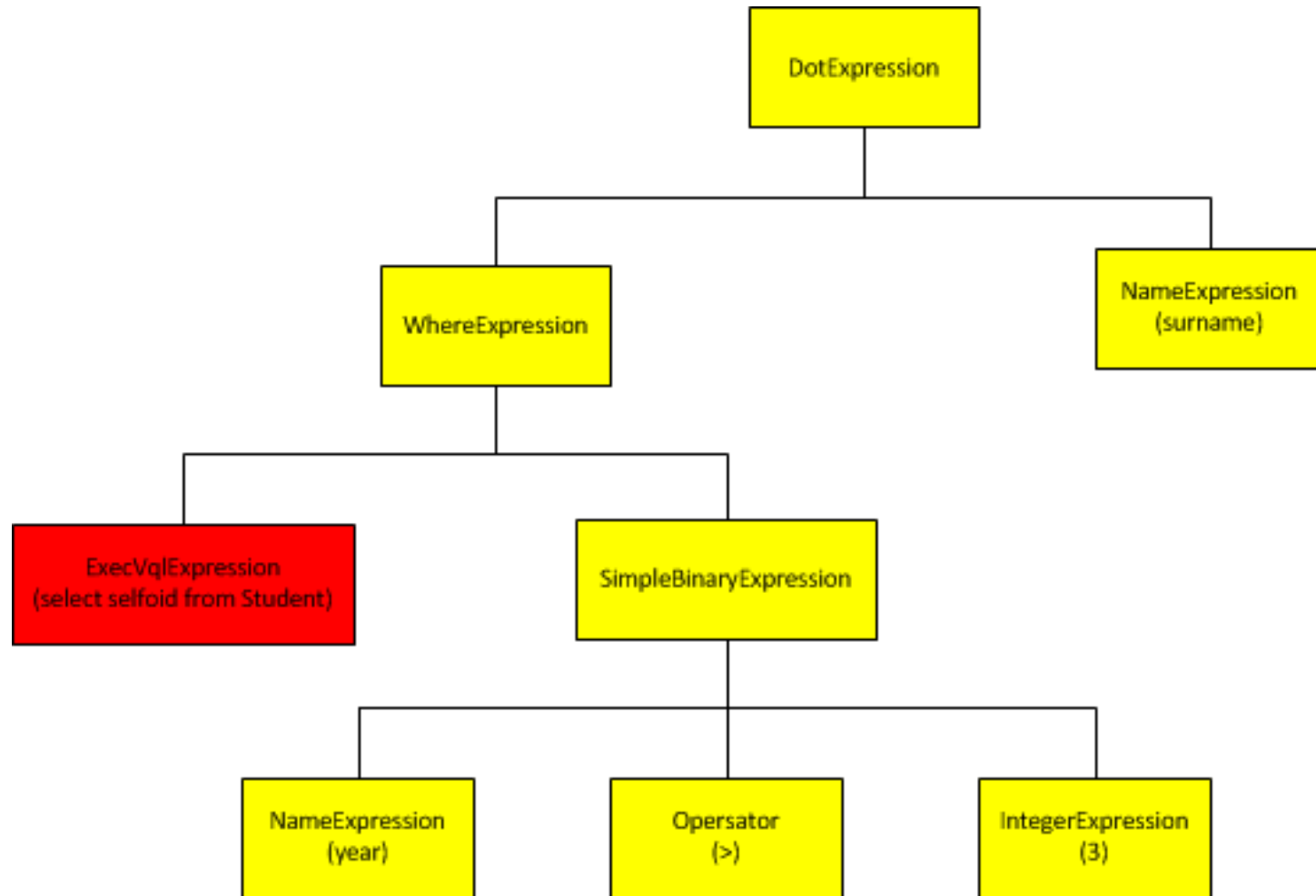


# „Optymalizacja” zapytań przez wrapper

---

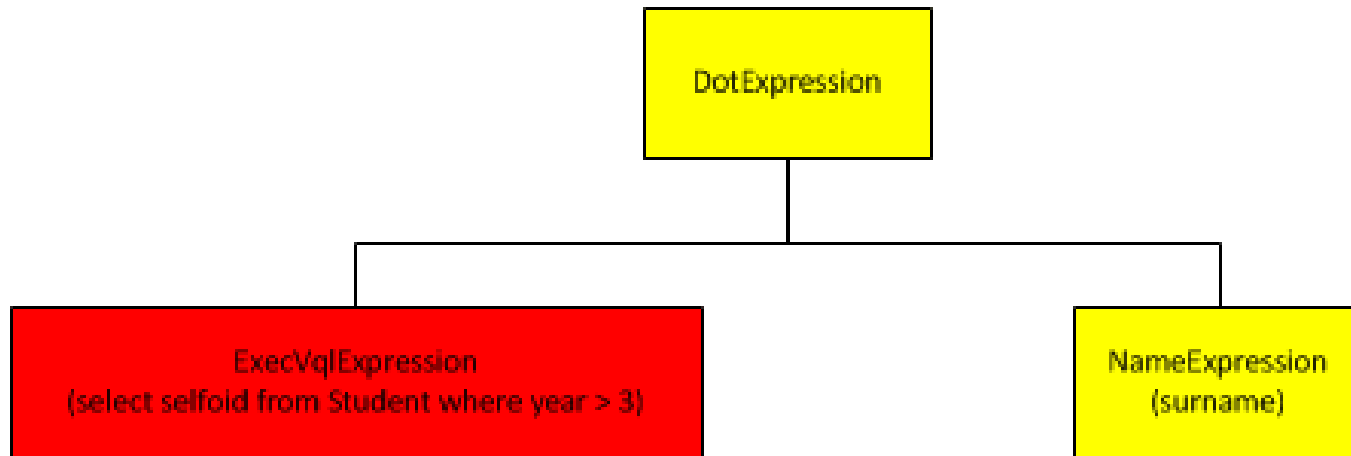


# „Optymalizacja” zapytań przez wrapper



# „Optymalizacja” zapytań przez wrapper

---





Pytania?



Dziękuję