

# Razvoj SUPB z integracijo v programski jezik Python

Janez Sedeljšak

Mentor: doc. dr. Boštjan Slivnik

Somentor: asist. dr. Marko Požnenel

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

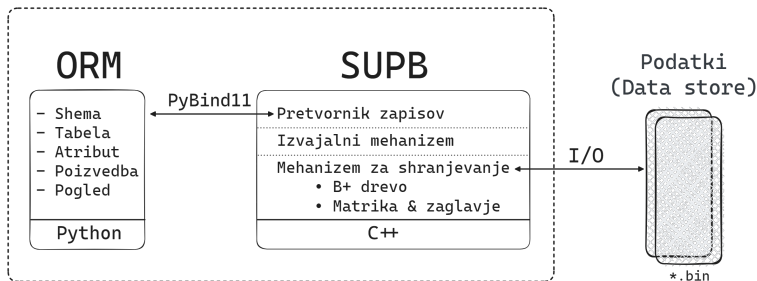
*js0578@student.uni-lj.si*

September 7, 2023

# Motivacija

- Spoznati delovanje podatkovnih baz
  - Izogib uporabi anti-vzorcev
  - Boljša implementacija podatkovnega sloja
- Razvoj minimalističnega SUPB za programski jezik Python:
  - SUPB na nivoju programskega jezika C++
  - Indeksiranje z uporabo B+ dreves
  - Intuitiven način komunikacije s podatkovno bazo
  - Izhodišče sta MySQL in SQLite

# Arhitektura rešitve Graphenix



# Struktura shranjenih podatkov

## Zaglavje

Prosto mesto: 2.	
ID	VRSTICA V MATRIKI
0	-1 (pobrisano)
1	0
2	1
3	-1 (pobrisano)
4	3

## Matrika podatkov

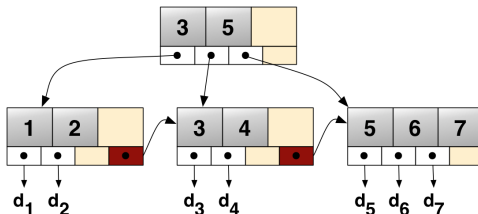
ODMIK	VRSTICA V MATRIKI
0	zapis (ID = 1)
1	zapis (ID = 2)
2	prazno (izbrisan zapis)
3	zapis (ID = 4)
...	

Zapisi

Atributi

# Indeksiranje z uporabo B+ drevesa

- Implementacija s pomočjo programskega jezika C++
- Shranjevanje strukture v binarni datoteki
- Uproaba “generikov” za različne podatkovne tipe (nizi, cela števila, realna števila, povezave)



## Definiranje sheme

```
class User(gx.Model):  
    name = gx.Field.String(size=100)  
    tasks = gx.Field.VirtualLink("user")  
    sent = gx.Field.VirtualLink("sender")  
    recieved = gx.Field.VirtualLink("reciever")
```

```
class Task(gx.Model):  
    content = gx.Field.String(size=100)  
    user = gx.Field.Link()
```

```
class Message(gx.Model):  
    content = gx.Field.String(size=50)  
    date = gx.Field.DateTime()  
    sender = gx.Field.Link().as_index()  
    reciever = gx.Field.Link().as_index()
```

# Poizvedovanje

```
# uporabniki urejeno po imenu - padajoče  
_, view = User.order(User.name.desc()).all()  
  
# uporabniki in njihove naloge  
_, view = User.link(tasks=Task).all()  
  
# število sporočil in datum zadnjega sporočila po uporabnikih  
counts = Message.agg(by=Message.sender,  
    count=gx.AGG.count(), latest=gx.AGG.max(Message.date))
```

## Vgnezdne poizvedbe

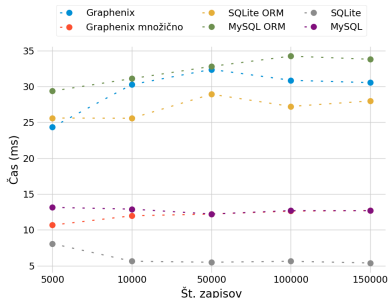
Nabor uporabnikov, njihovih nalog in prejetih sporočil, kjer na sporočila vežemo še pošiljatelja

```
_, view = User.link(  
  tasks=Task.limit(3),  
  recieved=Message.link(sender=User)  
) .filter(User.name.iregex('john.*')).all()
```

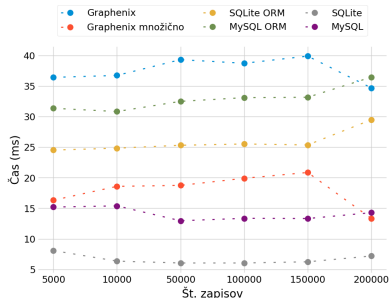
```
{  
  "name": "John Doe",  
  "tasks": [  
    {"content": "Finish the diploma"},  
  ],  
  "recieved": [  
    {"content": "Hello", "sender": {}},  
  ]  
}
```



# Vstavljanje zapisov

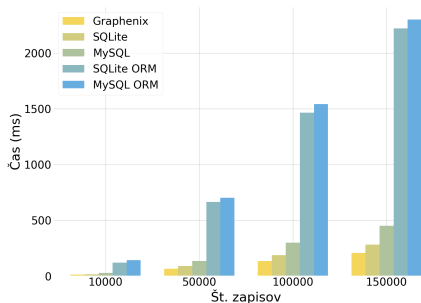


Množično vstavljanje podatkov.

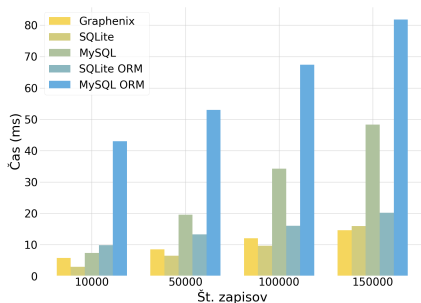


Vstavljanje z dodatnim indeksiranim poljem.

# Poizvedovanje

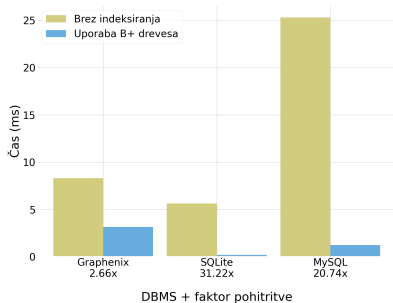


Branje brez dodatnih parametrov  
znotraj poizvedbe.

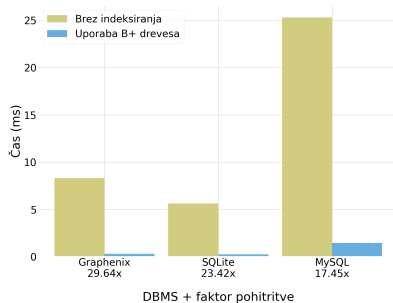


Branje z omejitvami - prvih 500  
uporabnikov + filtriranje +  
urejanje.

## Indeksiranje pred in po optimizaciji ( $10^5$ zapisov)



Pohitritve s pomočjo indeksiranja.



Pohitritve s pomočjo indeksiranja po optimizaciji.

## Sklepne ugotovitve

- Kje je rešitev uporabna?
- Kaj rešitvi manjka za uporabo v produkcijskem okolju?
- Ali je bil razvoj uspešen?

# Hvala za pozornost