

# rusty\_ems

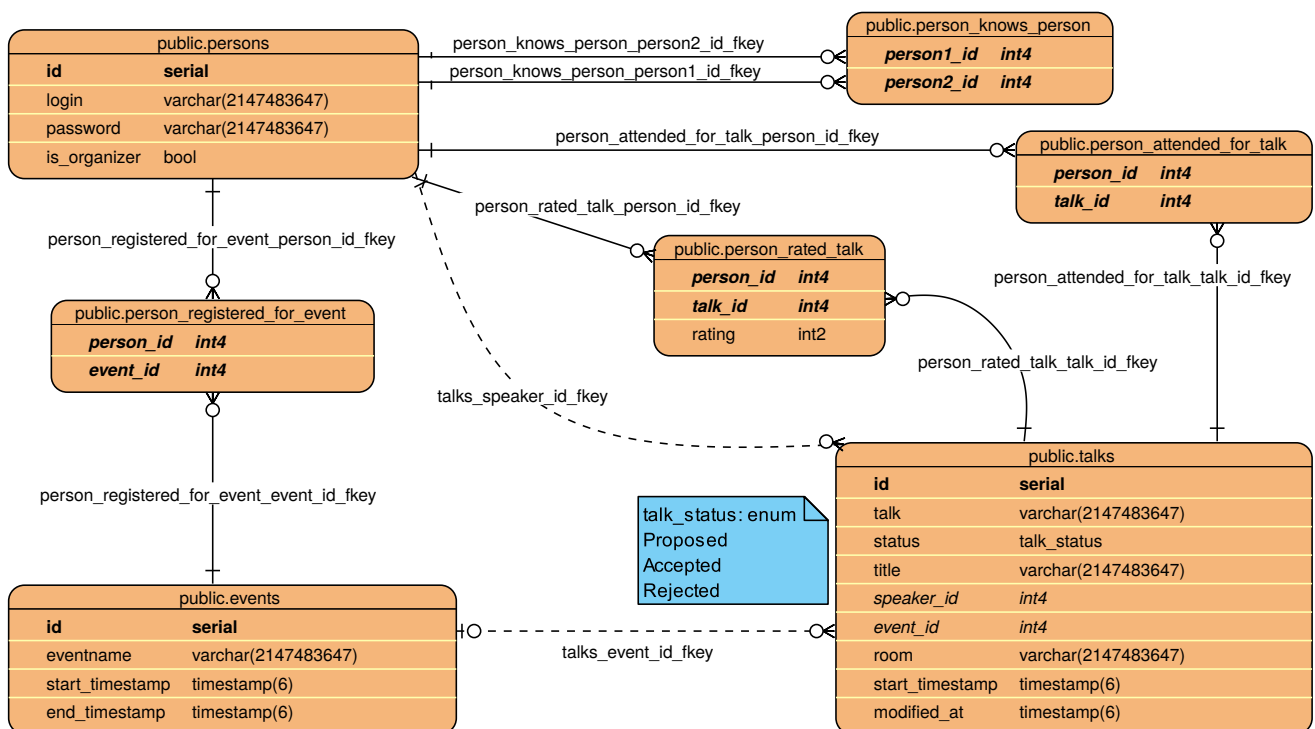
Projekt na przedmiot Bazy danych @ II UWr 2017

Całość napisana i przetestowana przy użyciu języka **Rust 1.18** z gałęzi **stable**.

## Artefakty

- **Model konceptualny** - Diagram E-R w postaci obrazu poniżej lub w [docs/erd.svg](#)
- **Model fizyczny** - [up.sql](#) w katalogu [migrations/20170529191530\\_base](#)
- **Dokumentacja**
- **Program** - [rusty\\_ems](#) w katalogu [target/release](#), zaś źródła w [src](#)

## Model konceptualny



## Dostosowanie środowiska

Rust posiada oficjalny instalator środowiska dostępny na stronie [rustup.rs](https://rustup.rs).  
Sprowadza się do wykonania w terminalu poniższej komendy i podążaniu za instrukcjami na ekranie:

```
curl https://sh.rustup.rs -sSf | sh
```

## Kompilacja projektu

Wykonanie poniższej komendy przeprowadzi kompilację programu w wersji finalnej:

```
cargo build --release
```

Program wykonywalny **rusty\_ems** pojawi się w katalogu **target/release**.

## Specyfikacja

Program przestrzega specyfikacji podanej przez wykładowcę na stronie SKOS.  
Każde zapytanie musi być podane w pełnej jednej linii wejścia w formacie JSON.  
Po każdym zapytaniu program zwraca rezultat w formacie JSON.

### Zaimplementowane zapytania

- (\*) **open** <baza> <login> <password>

Zwraca ERROR również jeśli nie uda się wgrać schemy do podanej bazy (np. z powodu braku uprawnień).

- (\*) **organizer** <secret> <newlogin> <newpassword>
- (\*O) **event** <login> <password> <eventname> <start\_timestamp> <end\_timestamp>

- (\*O) user <login> <password> <newlogin> <newpassword>
- (\*O) talk <login> <password>
- (\*U) register\_user\_for\_event <login> <password> <eventname>
- (\*U) attendance <login> <password> <talk>
- (\*U) evaluation <login> <password> <talk> <rating>
- (O) reject <login> <password> <talk>
- (U) proposal <login> <password> <talk> <title> <start\_timestamp>
- (U) friends <login1> <password> <login2>
- (\*N) user\_plan <login> <limit>
- (\*N) day\_plan <timestamp>
- (\*N) best\_talks <start\_timestamp> <end\_timestamp> <limit> <all>
- (\*N) most\_popular\_talks <start\_timestamp> <end\_timestamp> <limit>
- (\*U) attended\_talks <login> <password>
- (\*O) abandoned\_talks <login> <password> <limit>
- (N) recently\_added\_talks <limit>
- (U/O) rejected\_talks <login> <password>
- (O) proposals <login> <password>
- (U) friends\_talks <login> <password> <start\_timestamp>  
<end\_timestamp> <limit>
- (U) friends\_events <login> <password> <eventname>

Niezaimplementowane zapytania

- (U) recommended\_talks <login> <password> <start\_timestamp>  
<end\_timestamp> <limit>

## Testowanie

Dołożyłem wszelkich starań, by program został dobrze przetestowany. W tym celu napisałem prostą testerkę dostępną pod adresem <https://github.com/kamarkiewicz/tster>. Środowiskiem testowym był Ubuntu 16.04.2 LTS (live cd).