Autor: Kamil Markiewicz, numer indeksu 264407

rusty_ems

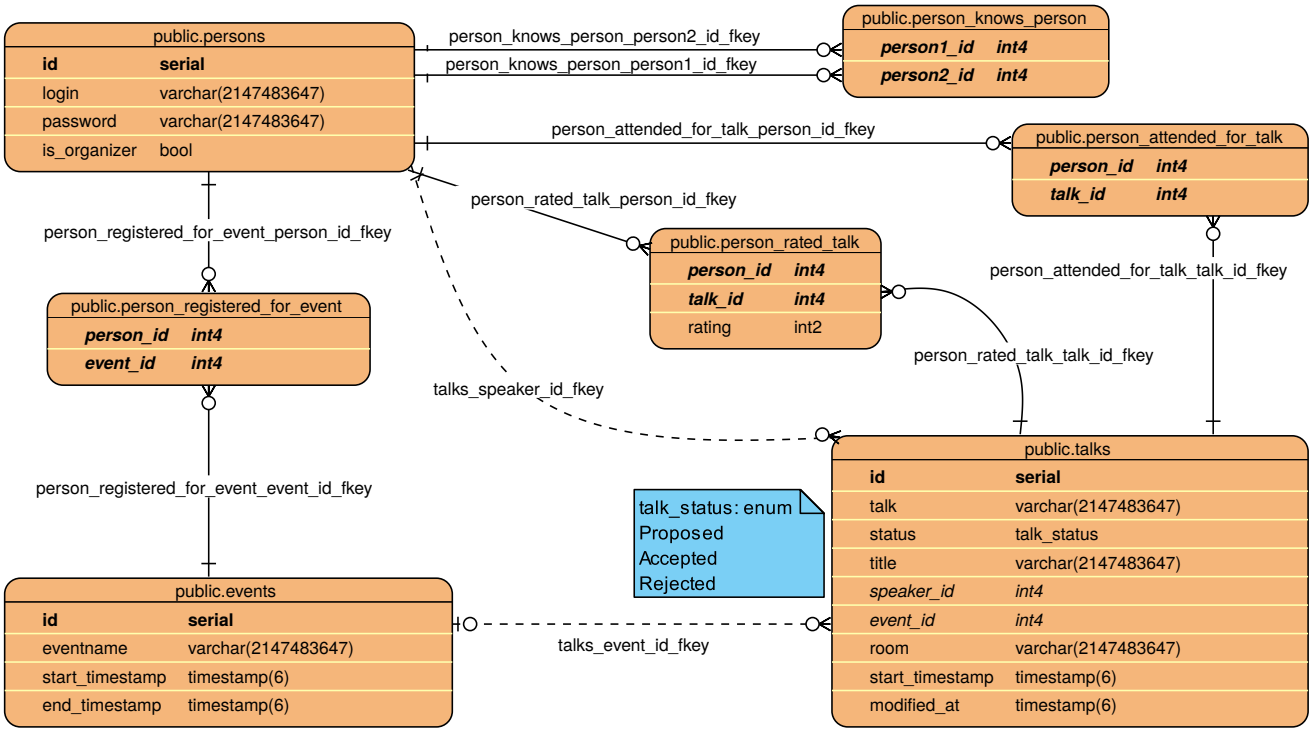
Projekt na przedmiot Bazy danych @ II UWr 2017

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. 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.

Specyfilacja

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.

Przy zapytaniach, gdzie mogły pojawić się niejednoznaczności w zwracanej kolejności, przyjąłem dodatkowe sortowanie po kolejności wstawienia interesującego względem zapytania obiektu, np. user_plan jest sortowany w pierwszej kolejności po czasie rozpoczęcia referatu, zaś w drugiej po kolejności wstawiania referatów do bazy.

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).