

C-nema

Projekt na Inżynierię danych

Filip Jasionowicz
Jakub Oskwarek
Przemysław Pasiński

19 marca 2023

Zarys projektu

Jako temat bazy danych wybraliśmy system rezerwacji biletów kinowych o nazwie C-nema. Chcielibyśmy zrealizować następujące funkcjonalności dostępne dla użytkownika:

- możliwość przeglądania seansów i sal
- możliwość dodawania seansów w różnych salach
- możliwość dodawania filmów, gatunków, aktorów oraz producentów
- dostęp do informacji o szczegółach filmów (reżyser, producent, główni aktorzy, rok wydania itp.)
- możliwość rejestrowania się
- możliwość rezerwowania biletów na wybrane miejsca
- możliwość odwoływania rezerwacji

Struktura bazy

W ogólnym ujęciu, baza dzieli się na trzy główne części, odpowiadające kolejno za:

- (I) przechowywanie informacji o filmach
- (II) przechowywanie informacji o seansach
- (III) przechowywanie informacji o biletach i rezerwacjach

W dalszej części opisu nie będziemy przy każdym polu tabeli powtarzać, że jest obowiązkowe. Dodamy komentarz jeśli będzie inaczej.

Filmy

W skład tej części wchodzi następujące tabele:

- **languages** — reprezentuje informacje o językach:
 - id — klucz główny
 - nazwa języka
- **movies** — przechowuje ona informacje o konkretnych filmach:
 - id — klucz główny
 - tytuł
 - czas trwania

- ograniczenie wiekowe — minimalny wiek (w latach, co najwyżej 19) widza, dla którego przeznaczony jest ten film (przyjeliśmy, że jest to pole obowiązkowe, aby uniknąć ewentualnych problemów z prawem i wymusić ustalenie, dla jakiej widowni jest to film odpowiedni)
- data międzynarodowej premiery (nie jest obowiązkowa, ponieważ dla niszowych filmów może w ogóle nie istnieć, a dla starych filmów może nie być znana)
- język oryginału (nieobowiązkowy, ponieważ film może być niemy) — klucz obcy wskazujący na id w tabeli `languages`
- **genres** — przechowuje listę gatunków filmowych:
 - id — klucz główny
 - pełna nazwa (unikalna)
 - skrócona nazwa (np. sci-fi zamiast science-fiction, nieobowiązkowa)
- **people** — przechowuje listę osób z branży filmowej, które chcielibyśmy wyświetlać przy prezentowaniu filmu, czyli reżyserów, aktorów itd. Rozważamy dodanie większej liczby osób, takich jak np. kompozytorzy muzyki lub operatorzy kamery.
 - imię
 - nazwisko
 - opcjonalny pseudonim, np. „The Rock” dla Dwayne’a Johnsona
- **producers** — przechowuje listę producentów (rozumianych tu jako firmy, spółki, wytwórnie, itp., czyli np. jeśli producentem filmu jest jedna osoba, to w tej tabeli wystąpi nazwa, pod którą zarejestrowane jest jej przedsiębiorstwo produkcyjne — w ten sposób przechowujemy również informacje o tym, od jakiego podmiotu nabyliśmy prawa do pokazów tego filmu)
 - id — klucz główny
 - nazwa
- **journals** — przechowuje dane o czasopismach w których publikowane są wszelakie recenzje.
 - id — klucz główny
 - pełna nazwa (unikalna)
 - skrócona nazwa
- **reviews** — przechowuje dane o recenzjach publikowanych w czasopismach. Wartości są unikalne na trójce (journal id, tytuł, data publikacji).
 - id — klucz główny
 - journal id — klucz obcy do tabeli `journals`
 - tytuł (unikalna)
 - opis — treść recenzji
 - streszczenie treści recenzji
 - data publikacji
 - ocena w gwiazdkach od 0 do 10
 - skrócona nazwa

Pozostałe tabele, tzn. `movies_genres`, `movies_producers`, `movies_directors`, `movies_actors`, `movies_reviews` itp, służą jako pomocnicze tabele umożliwiające reprezentowanie relacji wiele-do-wielu, zgodnie z nazwami. Posiadają odpowiednie obowiązkowe klucze obce. Każda para w tych tabelach ma być unikalna.

Seanse

- **rooms** — przechowuje informacje o salach kinowych
 - id
 - nazwa sali (obowiązkowa i unikalna)
- **seats** — przechowuje informacje o miejscach
 - id — klucz główny
 - sala, w której znajduje się to miejsce — klucz obcy wskazujący na id w tabeli **rooms**
 - numer rzędu
 - numer miejsca w rzędzie

Trójka (sala, rząd, miejsce) jest unikatowa. Można zatem przeorganizować układ sali przy zachowaniu tych samych id siedzeń, ale — zgodnie z intuicją — w danym miejscu może znajdować się tylko jedno siedzenie.

- **regionalizations** — przechowuje informacje o kombinacji językowej w zregionalizowanej wersji filmu:
 - id — klucz główny
 - język audio (klucz obcy wskazujący na id w tabeli **languages**; nieobowiązkowy — NULL oznacza oryginalną ścieżkę)
 - język lektora (klucz obcy wskazujący na id w tabeli **languages**; nieobowiązkowy — NULL oznacza brak lektora)
 - język napisów (klucz obcy wskazujący na id w tabeli **languages**; nieobowiązkowy — NULL oznacza brak napisów)

Trójka (audio, lektor, napisy) jest unikatowa.

- **abstract_screening** — przechowuje informacje o abstrakcyjnych instancjach seansów (jest to swego rodzaju harmonogram służący do zmniejszenia rozmiaru tabeli **screenings**)
 - id — klucz główny
 - godzina rozpoczęcia seansu
 - sala — klucz obcy wskazujący na id w tabeli **rooms**
 - bazowa cena biletu (tzn. bez uwzględniania żadnych zniżek)

Zdecydowaliśmy się zostawić bazową cenę biletu w tej tabeli ponieważ, seanse w różnych godzinach cieszą się różną popularnością oraz różnicę w salach w których realizowany jest seans mogą wpływać na cenę (normalna sala w porównaniu do sali 5D).

- **screenings** — przechowuje informacje o konkretnej instancji abstrakcyjnego seansu:
 - id — klucz główny
 - data — dzień w który odbędzie się dana instancja seansu **movies**
 - abstract screening id — klucz obcy wskazujący na id abstrakcyjnego seansu w tabeli **abstract_screening**
- **movie_realisations** — przechowuje informacje o realizacji (regionalizacji) filmu:
 - id — klucz główny
 - film — klucz obcy wskazujący na id w tabeli **movies**
 - regionalizacja — klucz obcy wskazujący na id w tabeli **regionalizations** (Obowiązkowy)
- **movies_screenings** — łączy **abstract screening** z **movie_realisation** za pomocą id

Bilety i rezerwacje

- **ticket_types** — przechowuje informacje o rodzajach (kategoriach) biletów, np. normalny, ulgowy, dla studentów, dla seniorów itp.
 - id — klucz główny
 - nazwa
 - zniżka — ułamek reprezentujący, o jaką część obniżamy bazową cenę
- **customers** — przechowuje informacje o klientach
 - id — klucz główny
 - nazwa użytkownika
 - email — unikalne pole
- **reservations** — przechowuje informacje o rezerwacjach. Jedna rezerwacja może składać się z wielu biletów:
 - id — klucz główny
 - klient — klucz obcy wskazujący na id w tabeli **customers**
 - data złożenia
- **tickets** — przechowuje informacje o sprzedanych biletach
 - id — klucz główny
 - seans — klucz obcy wskazujący na id w tabeli **screenings**
 - rodzaj biletu — klucz obcy wskazujący na id w tabeli **ticket_types**
 - rezerwacja — klucz obcy wskazujący na id w tabeli **reservations**
 - data anulowania — nieobowiązkowa, domyślnie NULL — każdy bilet może zostać anulowany co najwyżej raz. Jeśli ten sam użytkownik będzie chciał jeszcze raz kupić bilet na ten sam seans, otrzyma już bilet o innym id. Dzięki temu możemy łatwo przechowywać historię wydanych i anulowanych biletów.

Aplikacja

Interfejs dla klienta pozwala na

- przeglądanie zaplanowanych seansów
- wyświetlanie bardziej szczegółowych informacji o filmach
- kupowanie biletów na seans
 - przeglądanie już kupionych biletów na ten seans
 - wybieranie miejsc
 - wybieranie rodzaju biletu
- przeglądanie historii
- anulowanie biletów (nie później niż godzinę przed seansem)

Interfejs dla administratora pozwala na

- przeglądanie i dodawanie języków wspieranych w kinie (jako audio, lektor lub napisy)
- zarządzanie filmami:
 - dodawanie nowych filmów
 - przeglądanie, dodawanie i modyfikację szczegółów filmu
 - wyszukiwanie filmów po tytule

- zarządzanie recenzjami:
 - dodawanie recenzji
 - przeglądanie szczegółów recenzji
 - podpinanie i odpinanie autorów i recenzowanych filmów
 - usuwanie recenzji, które nie są do niczego podpięte
- zarządzanie osobami:
 - dodawanie, modyfikację i usuwanie (niepodpiętych) osób
 - wyszukiwanie po fragmencie danych osobowych
- zarządzanie gatunkami:
 - dodawanie gatunków
 - usuwanie nieużywanych gatunków
- zarządzanie rodzajami biletów:
 - dodawanie rodzajów biletów z odpowiednimi zniżkami
 - usuwanie nieużywanych rodzajów biletów
- zarządzanie seansami:
 - dodawanie seansów
 - usuwanie seansów, na które nikt nie kupił biletów
 - wyświetlanie informacji o zarezerwowanych biletach
- przeglądanie statystyk:
 - najchętniej wybierane miejsca
 - stali klienci
 - przychody kina w wybrany okresie

GUI i użyteczne funkcje, widoki, triggerzy

Dodatkowo dodaliśmy różne funkcje i widoki, by ułatwić korzystanie z GUI oraz inne, które również są użyteczne.

- `get_or_make_user_id` — funkcja która przyjmuje `username` i `mail`, która zwraca id customera, jeżeli customer nie istnieje oraz mail nie jest zajęty tworzy nowego użytkownika i zwraca jego id. Gdy mail jest zajęty wyrzuca `Exception`. Jest to jedyna metoda służąca rejestrowaniu customerów.
- `buy_ticket` — jedyna funkcja z której korzystamy by dodawać bilety. Przyjmuje `screening_id`, `seat_id`, `ticket_type id`, `reservation_id` oraz `customer_id`. By kupić bilet potrzebujemy wszystkie te wartości, z wyjątkiem dwóch ostatnich, gdyż potrzebujemy tylko jedną z nich. Jeżeli `reservation_id` jest nullem tworzymy dla klienta rezerwację, w przeciwnym przypadku dodajemy do istniejącej już rezerwacji bilet.
- `add_to_schedule` — funkcja służąca do dodawania seansów w kinie. Przyjmuje ona wszystkie informacje o planowanym seansie, sprawdza, czy film i sala w której ma się odbywać projekcja istnieją w bazie. Jeżeli nie, to nie pozwala dodać seansu. Oprócz tego funkcja sprawdza, czy istnieje już w bazie realizacja filmu z odpowiednią regionalizacją, abstrakcyjny seans odpowiadający realizacji filmu, podanej godzinie, sali i cenie podstawowej. Jeżeli tak, to tworzy nowy seans dodając pojedynczą krotkę w tabeli `screenings` podpinając ją do odpowiedniej krotki w tabeli `abstract_screenings`. Jeżeli zaś nie ma abstrakcyjnego seansu/realizacji filmu/regionalizacji odpowiadającej naszemu seansowi, to tworzy wymagane krotki w tabelach `abstract_screenings/movies_screenings/movie_realizations/regionalizations` a następnie podcina do nich nową krotkę w tabeli `screenings` reprezentującą nasz seans. Funkcja nie pozwala również dodać seansu odbywającego się w przeszłości lub kolidującego z już istniejącymi seansami. Seanse można również dodawać poprzez insert do widoku `schedule` efektywnie wywołujący funkcję `add_to_schedule` z odpowiednimi argumentami.

- `last_res_id` — funkcja która zwraca id ostatniej rezerwacji danego customera. Nie jest ona używana w GUI, natomiast wraz z funkcją `buy_ticket` ułatwia ona dodawanie kilku biletów do danej rejestracji.
- `get_free_seats` — funkcja, która dostaje `screening_id` i zwraca tabelę wszystkich wolnych miejsc w sali w której ten seans się odbywa.
- `delete_reservation_if_empty` — trigger włączający się po akcji DELETE. Sprawdza on czy rezerwacja z której usunęliśmy bilet jest pusta. W takim przypadku usuwamy reservation. Jest to jedyny sposób w który usuwamy rezerwacje.
- `user_history` — funkcja przyjmująca `customer_id` zwracająca tabelę z informacjami o biletach customera.
- `cancel_ticket` — funkcja przyjmująca `ticket_id` która zmienia status biletu na anulowany.
- `all_tickets` — widok wyświetlający informacje o wszystkich biletach
- `movie_info` — widok wyświetlający podsumowanie wszystkich informacji o filmie
- dodatkowo mamy w GUI możliwość obliczenia przychodów kina z danego okresu
- `schedule` — widok prezentujący wszystkie nadchodzące seanse (takie, które jeszcze się nie rozpoczęły). W GUI służy do prezentacji planu kina dla admina oraz zwykłego użytkownika.
- na poziomie bazy posiadamy też funkcję `activity_count_as_parameterized` — przyjmuje ona id osoby oraz sufix nazwy tabeli i na tej podstawie zlicza liczbę wystąpień tej osoby w odpowiedniej tabeli, wykonując dynamicznie tworzone zapytanie (wynik nazywamy *aktywnością*, jest ona mniej-więcej proporcjonalna do „medialności” danej osoby w branży filmowej). Za pomocą tej funkcji można bardzo łatwo tworzyć konkretne funkcje `activity_count_as_actor`, `activity_count_as_director` itd. Poza tym, jeśli w przyszłości użytkownik bazy chciałby dodać kolejny rodzaj połączenia osób z filmami, np. rolę scenografa czy projektanta kostiumów, to funkcja od razu będzie działać ze zliczaniem. Jedynym warunkiem jest przestrzeganie konwencji z nazwą tabeli łączącej według wzorca `movies_...`.

Proste funkcje i widoki formatujące i wyciągające dane w przystępnej formie

- funkcja `short_format_review` zwraca dla podanej recenzji tekst pokazujący podsumowanie oraz skrót nazwy czasopisma, z którego pochodzi fragment.
- widok `people_info` pokazuje wszystkie informacje o osobach, dodatkowo wyświetlając ich dane osobowe, sformatowane jako `imie`, `pseudonim`, `nazwisko`, oraz licznik sumarycznej aktywności. Używa w tym celu funkcji `format_person`.
- funkcja `format_author` oprócz informacji o osobie wyświetla jeszcze rolę
- funkcje `format_movie...` oraz `short_format_movie...` zwracają oddzielone przecinkami informacje o osobach w danym filmie
- widok `ticket_types_with_statistics` wyświetla informacje o liczbie rezerwacji poszczególnych typów biletów

Sposób uruchomienia aplikacji

1. Proszę zainstalować Node.js w wersji przynajmniej 16.15.1. (mirror odpowiedniego skryptu do tego celu znajduje się pod adresem https://deb.nodesource.com/setup_16.x, następnie wystarczy go uruchomić przez `sudo -E bash` i zainstalować `sudo apt install -y nodejs`).
2. Jeśli przy poprzednim kroku nie zainstalował się npm, to proszę również doinstalować.
3. Proszę udać się do katalogu `cnema/app` i wykonać `npm install`, aby pobrać i dołączyć odpowiednie zależności.
4. Proszę za pomocą `create.sql` utworzyć w PostgreSQL na porcie 5432 bazę o nazwie `cnema`, której właścicielem jest użytkownik `cnemaadmin` o hasle `bazunia` (jeśli to jest niewygodne, credentials do połączenia można zmienić w linii 12 w pliku `cnema/app/app.ts`).

5. Proszę w katalogu `cnema/app` wykonać `npm run dev`, co uruchomi serwer i będzie reagować na zmiany w pliku `app.ts`. Połączenie z bazą powinno nastąpić automatycznie. Aplikacja będzie domyślnie dostępna na `localhost:8000/`.
6. Aby zalogować się na zwykłego użytkownika, należy podać nazwę użytkownika (dowolną) i poprawny adres email (adresy nie mogą się powtarzać dla różnych użytkowników, ale nazwy już tak)
7. Aby zalogować się na konto admina, podajemy username `admin` i email `admin`