

Dokumentacja bazy danych



Przygotowane przez:

Karolina Romanowska

Marianna Gromadzka

Ksawery Chodyniecki

Opis rozwiązania

Baza danych została stworzona na potrzeby aplikacji elektronicznej książki kucharskiej. Aplikacja umożliwia użytkownikom m.in. dodawanie przepisów i opinii, tworzenie grup i list zakupów oraz obserwowanie innych użytkowników, lub wyróżnianie ich przepisów. W bazie przechowywane są wszystkie dane potrzebne do działania aplikacji, co zostanie opisane dokładniej poniżej.

Tabele

Projekt bazy danych składa się z czternastu tabel:

1. **User** – przechowuje informacje o użytkownikach: nazwę, hasło i preferowany system jednostek.
2. **Recipe** – przechowuje informacje potrzebne do sporządzenia dania (opis przygotowania, potrzebny czas etc.), oraz nazwę autora i datę dodania.
3. **Ingredient** – przechowuje informacje o składnikach – nazwę, gęstość oraz podstawową jednostkę w której jest wyrażana.
4. **Ingredient_list** – przechowuje informacje o składnikach wykorzystywanych w konkretnych przepisach jak również o składnikach dodanych przez użytkownika: nazwę składnika z tabeli Ingredient, oraz jego ilość i jednostkę w jakiej jest podana (będącą obiektem z tabeli Unit) i id przepisu w którym występuje (wartość może być równa NULL).
5. **Unit** – zawiera informacje o jednostkach dostępnych dla użytkowników. Przechowuje nazwę jednostki oraz jej przeliczenia na inne dostępne jednostki i informację o tym do jakiego systemu jednostek należy.
6. **Unit system** – przechowuje nazwę systemu jednostek. Użytkownik może wybrać preferowany system jednostek.
7. **Favorite** – przechowuje informację o ulubionych przepisach użytkowników: nazwę użytkownika i id przepisu.
8. **Opinion** – przechowuje opinie użytkowników na temat przepisów: ocenę, komentarz, datę dodania, autora i id przepisu którego dotyczy.
9. **Reported** – przechowuje informację o zgłoszonych przez użytkowników opiniach (ze względu na wulgaryzm itp.), id opinii oraz nazwę zgłoszonego użytkownika. Gdy wystarczająca ilość użytkowników zgłosi opinię jest ona usuwana.
10. **Followed** – przechowuje informacje o użytkownikach obserwowanych przez danego użytkownika poprzez id obserwowanego i obserwującego.
11. **Shopping_list** – składa się składniki znajdujące się w liście zakupów użytkowników aplikacji
12. **Group** – przechowuje id grupy oraz id listy zakupów która odpowiada tej grupie
13. **Belong** – przechowuje informacje o użytkownikach należących do danej grupy id grupy oraz nazwę użytkownika
14. **Publicity** – przechowuje dostępność przepisów dla różnych użytkowników. Grupa 0 stanowi grupę publiczną.

Funkcje

1. **calc_rating** – oblicza średnią ocenę przepisu
2. **convert_unit** – przyjmuje jednostkę obecną, docelową i ilość składnika (w obecnej jednostce). Zwraca ilość składnika w docelowej jednostce, na podstawie informacji o stosunku jednostki do litra z tabeli UNIT

Procedury

1. **add_grup** – przyjmuje nazwę założyciela grupy oraz nazwę grup, wywoływana przy tworzeniu

nowej grupy dodaje grupę do tabeli GROUP i informację o przynależności założyciela do grupy (w tabeli BELONG)

2. **delete_account** – usuwa użytkownika z tabeli USER oraz wszystkie rekordy, które zostały dodane przez niego z innych tabel
3. **add_ingredient_to_ingredient_list** – funkcja wspomagająca działanie procedur **add_new_ingredient_to_shopping_list**, **add_ingredient_to_recipe** sprawdzająca istnienie składnika w tabeli INGREDIENT i go ewentualne dodawanie jeśli nie istnieje, oraz sprawdzająca jego istnienie w INGREDIENT_LIST – jeśli nie istnieje, tworzony jest nowy rekord. Procedura zwraca id z INGREDIENT_LIST, która została dodana lub istniała.
4. **add_new_ingredient_to_shopping_list** – przyjmuje nazwę składnika, jego ilość i jednostkę oraz nazwę właściciela. Dodaje składnik do tabeli INGREDIENTS_LIST i tabeli INGREDIENT przez wywołanie procedury **add_ingredient_to_ingredient_list**, po czym jeśli w taki sam składnik nie został wcześniej dodany do tabeli SHOPPING_LIST, dodaje go.
5. **share_shopping_list** – przyjmuje nazwę użytkownika oraz id grupy. Dla każdego składnika z tabeli SHOPPING_LIST, którego właścicielem jest podany użytkownik i który nie został jeszcze przypisany do żadnej grupy (wartość GROUP_ID = null) sprawdza czy składnik o takim INGREDIENT_LIST_ID jest już w liście zakupów tej grupy, jeśli nie to dodaje go, zmieniając GROUP_ID na id danej grupy, jeśli tak, rekord jest usuwany.
6. **add_recipe** – przyjmuje wszystkie wartości potrzebne do dodania elementu do tabeli RECIPE oraz id grupy dla której powinien być ten przepis udostępniony, dodaje przepis do RECIPE i informację o przynależności do tabeli PUBLICITY. Procedura zwraca id dodanego przepisu.

Wyzwalacze

1. **tg_add_to_public** - po każdym dodaniu użytkownika dodaje do tablicy BELONG informację o tym że przynależy do grupy publicznej
2. **add_favourites_tg** – za każdym razem gdy użytkownik obserwuje drugiego użytkownika, przepisy obserwowanego są dodawane do ulubionych przepisów obserwującego

Analiza rozwiązania

Znane ograniczenia

- Przepis może być udostępniony wyłącznie jednej grupie.
- Grupa nie posiada właściciela, przez co w aplikacji wszyscy mają dostęp do wszystkich opcji takich jak zapraszanie czy usuwanie (nie musi być to złe, ale lepiej dać użytkownikom wybór).
- Motyw kolorystyczny aplikacji jest zapisywany lokalnie, a nie w bazie danych, przez co przy logowaniu na nowym urządzeniu trzeba ponownie go ustawiać.
- Hasła są zachowywane jako zwykły tekst, nie są ani trochę szyfrowane.
- Wszystkie ceny przepisów są zapisywane w złotych.
- Przy usuwaniu użytkownika nie ma zapytania, czy chce usunąć swoje przepisy. W ten sposób tracimy publiczne przepisy które użytkownik mógłby chcieć zostawić dla potomnych.

Możliwości dalszego rozwoju

- Kategorie jedzenia - dodanie dwóch tabel, jedna z samymi kategoriami, a druga przypisująca kategorię poszczególnym przepisom (podobnie jak składnik); przyspieszyło by to proces wyszukiwania jedzenia przez użytkowników.
- Dodanie przechowywania zdjęć do przepisów (choćby jednego na pogląd), dzięki czemu użytkownik mógłby szybciej określić czy dany przepis jest warty jego uwagi czy nie.
- Notatka własna użytkownika przypisywana do przepisu, widziana wyłącznie przez niego.
- Możliwość zapisania listy zakupów jako oddzielny plik .txt w komputerze.

Testy zaimplementowanych funkcji

Testy są umieszczone w pliku *tests.sql* i opisane po kolei, zostaną umieszczone zrzuty DBMS output.

TEST PROCEDURE share_shopping_list

If some ingredient already in group shopping list, then dont add them to group shopping list Before: 2, after in user shopping list 0, after in group shopping list 1	
--	--

Przed udostępnieniem użytkownik ma 2 składniki na liście, po udostępnieniu do grupy jeden ze składników zapisuje się, a jeden zostaje usunięty, ponieważ składnik o tym ingredient_list_id już jest na liście.

TEST PROCEDURE add_new_ingredient_to_shopping_list

Before: 110, after: 111, new id in ingredient_list: 130	
---	--

Składnik zostaje dodany i zostaje mu nadane id.

TEST PROCEDURE add_new_ingredient_to_shopping_list

Before in ingredient list: 110, after in ingredient list: 111 Before in ingredient: 67, after in ingredient: 68 New ingredient in shopping list: 131	
--	--

Po dodaniu składnika który nie istniał wcześniej w bazie danych zwiększa się ilość składników w INGREDIENTS_LIST oraz SHOPPING_LIST

TEST PROCEDURE add_recipe, PROCEDURE add_ingredient_to_recipe

New recipe id: 24 Recipe id: 24 -- > Ingredient name: mÄ...ka Unit: gram Amount: 100 Recipe id: 24 -- > Ingredient name: sÄll Unit: teaspoon Amount: 1	
--	--

Został dodany nowy przepis, wygenerowało się id. Dodano dwa składniki, recipe id, nazwa, jednostka i zapisują się poprawnie.

TEST FUNCTION calc_rating

5.33	
------	--

Średnia jest liczona poprawnie (wstawione oceny to 9, 6 i 1).

TEST PROCEDURE add_group

BurnedOliveTree	
-----------------	--

Zostaje dodana grupa testowa, wyświetla się poprawna nazwa właściciela.

TEST TRIGGER tg_add_to_public

Before: 18, after: 21	
-----------------------	--

Po dodaniu 3 użytkowników, liczba osób w publicznej grupie zwiększyła się poprawnie.

TEST PROCEDURE delete_account

Before: 1, after: 0 Before: 1, after: 0	
--	--

Po usunięciu konta użytkownik o danej nazwie nie występuje w tabelach USER i BELONG.

TEST PROCEDURE delete_group

Before: 1, after: 0 Before: 1, after: 0	
--	--

Po usunięciu grupy grupa o danej nazwie nie występuje w tabelach GROUP i BELONG.

TEST FUNCTION convert_unit

Before: 1 mililiter, after: .00001 gallon Before: 10 gallon, after: 758000 drop	
--	--

Przetestowano zamianę najmniejszej jednostki w bazie na największą i odwrotnie. W obu przypadkach wynik jest poprawny.

TEST TRIGGER add_favourites_tg

Before: 0 favourite recipes, after following othe user who has 6 recipes, following has 6 favourite recipes	
---	--

Po zaobserwowaniu użytkownika jego przepisy dodają się do ulubionych obserwującego.