Bazy danych 2023: lista zadań nr 7

27 luty 2023

Zadania z tej listy polegają na stworzeniu (np. w HTML i PHP z MySQLi) fragmentów interfejsu użytkownika dla bazy danych z pracowni. Do zadań stosują się te same uwagi, zastrzeżenia i wskazówki, co do poprzedniej.

Tworząc formularze pamiętaj o zasadzie, że użytkownik nie powinien musieć zapamiętywać i wprowadzać „gołych" numerów ID – najlepiej, jakby nigdy nie były widoczne na stronach (ale oczywiście będą się pojawiać w ich źródłach oraz w adresach stron).

Niektóre zadania mogą być nieco bardziej złożone niż dotychczasowe. Jeśli masz problemy z przedstawieniem rozwiązania w terminie – **z wyprzedzeniem** skontaktuj się z prowadzącym grupę i przedstaw swoje postępy.

Zad. 1. Stwórz skrypt, który wypisze (np. w formie tabeli) wszystkie dni (w których były składane zamówienia) oraz sumaryczną *wartość* zamówień złożonych w każdy z nich.

Zad. 2. Stwórz formularz, w którym będzie można wprowadzić nazwę produktu, jego cenę oraz stan magazynowy, oraz skrypt PHP, który pobierze dane wysłane z tego formularza i doda produkt o takich własnościach do bazy.

Skrypt powinien wyświetlać stosowny komunikat w zależności od powodzenia operacji, powinien też sprawdzać poprawność danych (zwłaszcza liczbowych, w tym ceny, która nie musi być liczbą całkowitą) przed wysłaniem zapytania SQL do bazy.

Zad. 3. Stwórz analogiczny interfejs do tworzenia nowych zamówień. Wybór klienta, którego dotyczy nowe zamówienie, powinien być realizowany np. za pomocą elementu `<select>`.

Zad. 4. Stwórz skrypt, który wywołany z *query string* `klient=`***nn*** (czyli z takim parametrem w adresie strony, przesłanym metodą GET) wygeneruje stronę z formularzem umożliwiającym modyfikację klienta o ID ***nn***.

Formularz powinien być wygodny w użyciu, np. użytkownik nie powinien musieć wypełniać wszystkich pól, aby zmienić tylko jeden atrybut. Czy możliwość modyfikacji ID klienta jest konieczna? wskazana?

Strona powinna też umożliwiać usunięcie użytkownika – przez kliknięcie linku, osobny formularz, lub *checkbox* w głównym formularzu. Stwórz skrypt, który pobierze dane wysłane z tego formularza i je obsłuży.

Zad. 5. Stwórz skrypt, który wywołany z *query string* `zamowienie=`***nn*** wygeneruje stronę umożliwiającą modyfikację detali zamówienia o ID ***nn***, tj. obejmowanych przez nie produktów i ich liczb sztuk (ale nie jego daty ani tym bardziej ID).

Strona powinna umożliwiać zmianę liczby sztuk (opcjonalnie także produktu) w już istniejących detalach zamówienia oraz ich usuwanie, a także dodawanie nowych, z podaną przez użytkownika liczbą sztuk oraz wybranym produktem. Przemyśl, ile formularzy będzie na stronie (stosunkowo najłatwiej zrobić ich $k + 1$, gdzie k to liczba detali danego zamówienia, ale można też ograniczyć się do dwóch) oraz ile zapytań `SELECT` musisz wykonać, by wygenerować stronę (im mniej, tym lepiej – powinno Ci się udać z dwoma, można zejść do jednego, ale to trochę nienaturalne).

Stwórz skrypt, który pobierze dane z tych formularzy i je obsłuży.

Zad. 6. Stwórz formularz, w którym będzie można wybrać datę, oraz skrypt PHP, który ją pobierze i wyświetli informacje o zamówieniach złożonych danego dnia. Jeśli danego dnia nie złożono żadnych zamówień, skrypt powinien wyświetlać stosowny komunikat. Przemyśl strukturę dokumentu, jaka będzie właściwa do prezentacji wyniku.

Do „informacji o zamówieniach" należą tu również nazwy klientów, którzy je złożyli, oraz nazwy i liczby sztuk produktów należących do danego zamówienia – a ponieważ skrypt powinien wykonywać tylko jedno zapytanie SQL, konieczne będzie użycie złączeń. Najbardziej elegancko byłoby użyć także grupowania po zamówieniach, a co za tym idzie również agregacji nazw oraz liczb sztuk produktów (np. do osobnych tablic JSON), ale można tego nie robić i zamiast tego iterować w bardziej skomplikowany sposób po wyniku zapytania po stronie aplikacji.

Zad. 7. Zmodyfikuj skrypt z zad. 1 tak, by wypisywane daty były linkami prowadzącymi do skryptu z zad. 6 z odpowiednimi argumentami w adresie. Jeśli w zad. 6 używałaś/-eś metody POST[1], zmodyfikuj jego rozwiązanie tak, by korzystało z GET, a linki działały zgodnie z intencją.

[1]Metoda POST poza tym, że sprawia, że skrypt nie jest „kompatybilny" z bieżącym zadaniem, jest też „niewłaściwa": nie zamierzamy *zapisywać* wysyłanych danych na serwerze, tylko *pobrać* informacje z nimi związane.