**Dokumentacja użytkowa bazy danych systemu rezerwacji kortów tenisowych**

**Spis treści**

1. Opis struktury bazy danych
2. Widoki w bazie danych
3. Procedury
4. Triggery
5. Przykłady użycia

**1. Opis struktury bazy danych**

Baza danych systemu rezerwacji kortów tenisowych składa się z następujących tabel:

**Tabele główne:**

* **uzytkownicy**: Przechowuje informacje o użytkownikach systemu.
  + Klucz główny: id\_uzytkownika
  + Pola: imie, nazwisko, email, numer\_telefonu, data\_rejestracji
* **korty**: Zawiera dane o kortach sportowych.
  + Klucz główny: id\_kortu
  + Pola: nazwa\_kortu, lokalizacja, typ\_nawierzchni, status, godzina\_otwarcia, godzina\_zamkniecia
* **rezerwacje**: Przechowuje informacje o rezerwacjach kortów.
  + Klucz główny: id\_rezerwacji
  + Pola: id\_uzytkownika, id\_kortu, data\_rezerwacji, godzina\_start, godzina\_koniec, status
* **platnosci**: Rejestruje płatności związane z rezerwacjami.
  + Klucz główny: id\_platnosci
  + Pola: id\_rezerwacji, kwota, status, data\_platnosci
* **cennik:** Zawiera informacje o kosztach rezerwacji w zależności od typu kortu i czasu wynajmu.
  + - **Klucz główny:** id\_cennika
* **Pola:** id\_kortu ,cena\_za\_godzine.

**2. Widoki w bazie danych**

Widoki zostały utworzone w celu ułatwienia dostępu do najczęściej wykorzystywanych danych:

**2.1. szczegoly\_rezerwacji**

Wyświetla szczegóły dotyczące każdej rezerwacji, łącząc informacje z tabel uzytkownicy, korty, oraz rezerwacje.

* Pola: id\_rezerwacji, imie, nazwisko, email, nazwa\_kortu, data\_rezerwacji, godzina\_rezerwacji, status
* Przykład użycia: **select \* from szczegoly\_rezerwacji;**

**2.2. statystyki\_wykorzystania\_obiektow**

Zapewnia informacje statystyczne na temat liczby rezerwacji na poszczególnych kortach.

* Pola: id\_kortu, nazwa\_kortu, lokalizacja, liczba\_rezerwacji
* Przykład użycia: **select \* from statystyki\_wykorzystania\_obiektow;**

**2.3. rezerwacje\_z\_platnościami**

Łączy dane z tabel rezerwacje i platnosci, prezentując informacje o rezerwacjach, które zostały opłacone.

* Pola: id\_rezerwacji, id\_uzytkownika, id\_kortu, data\_rezerwacji, status\_rezerwacji, kwota, status\_platnosci
* Przykład użycia: **select \* from rezerwacje\_z\_płatnościami;**

**3. Procedury składowane**

Procedury zostały zaprojektowane do obsługi najważniejszych operacji w systemie:

**3.1. dodaj\_rezerwacje**

Dodaje nową rezerwację dla użytkownika.

* Parametry: id\_uzytkownika, id\_kortu, data\_rezerwacji, godzina\_rezerwacji
* Przykład użycia:

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

**CALL dodaj\_rezerwacje(**

**1, -- ID użytkownika**

**1, -- ID kortu**

**'10:00:00', -- Godzina rozpoczęcia**

**'12:00:00' , -- Godzina zakończenia**

**'2024-12-26' -- Data rezerwacji (dzień)**

**);**

**3.2. sprawdz\_dostepnosc\_kortow**

Sprawdza dostępność kortów w podanym terminie.

* Parametry: data\_rezerwacji, godzina\_rezerwacji
* Przykład użycia:

**CALL sprawdz\_dostepnosc\_kortow('2024-12-25', '10:00' , '12:00');**

**3.3. usun\_rezerwacje**

Usuwa rezerwację o podanym identyfikatorze.

* Parametry: id\_rezerwacji
* Przykład użycia: **CALL usun\_rezerwacje(10);**

**3.4. oplac\_platnosc**

Aktualizuje status płatności na "opłacona" oraz zmienia status rezerwacji na "zarezerwowana".

**Parametry:**

* **platnosc\_id** *(INT)*: Identyfikator płatności, którą chcemy oznaczyć jako opłaconą.
* **Przykład użycia: CALL oplac\_platnosc(1);**

**Szczegóły działania:**

1. **Pobranie danych płatności**: Procedura odczytuje bieżący status płatności i przypisaną do niej rezerwację.
2. **Sprawdzenie statusu płatności**: Jeśli status płatności to **"oczekująca"**, procedura przechodzi do aktualizacji; w przeciwnym razie zgłasza błąd.
3. **Aktualizacja płatności**: Zmienia status płatności na **"opłacona"** i ustawia bieżącą datę jako datę płatności.
4. **Aktualizacja rezerwacji**: Zmienia status rezerwacji powiązanej z płatnością na **"zarezerwowana"**.
5. **Obsługa błędów**: Jeśli płatność ma już status **"opłacona"** lub **"anulowana"**, procedura zgłasza błąd i nie wykonuje żadnych zmian.

**Przykład zastosowania:**

* Rezerwacja o ID 1 ma przypisaną płatność o ID 1 ze statusem **"oczekująca"**.
* Po wywołaniu **CALL oplac\_platnosc(1);** :
  + Status płatności zmieni się na **"opłacona"**.
  + Status rezerwacji zmieni się na **"zarezerwowana"**.
* Jeśli płatność o ID 1 jest już opłacona, zostanie zgłoszony błąd:  
  **"Płatność już została opłacona lub anulowana."**

**4. Triggery**

**4.1 Trigger: before\_insert\_uzytkownicy**

* **Typ:** BEFORE INSERT
* **Cel:**  
  Automatyczne uzupełnianie pola data\_rejestracji w tabeli uzytkownicy, jeśli podczas wstawiania nowego rekordu wartość ta nie została podana.
* **Opis działania:**
  1. Gdy użytkownik wprowadza nowy rekord do tabeli uzytkownicy, trigger sprawdza, czy pole data\_rejestracji ma przypisaną wartość.
  2. Jeśli pole jest puste (NULL), trigger automatycznie ustawia bieżącą datę (CURDATE()) jako wartość tego pola.
* **Użycie:**
  1. Użytkownik nie musi ręcznie wypełniać pola data\_rejestracji, ponieważ trigger automatycznie wprowadzi odpowiednią wartość.
  2. W przypadku, gdy data została podana, trigger nie podejmuje żadnych działań.
* **Przykład:** **Zapytanie wstawiające rekord:**

**INSERT INTO uzytkownicy (imie, nazwisko,email,telefon) VALUES ('Fabian', 'Drwalczyk','fabian@gmail.com','123456789')**; 

**4.2 Trigger: sprawdz\_godziny\_otwarcia**

* **Typ:** BEFORE INSERT
* **Cel:** Weryfikacja, czy godziny rezerwacji kortu mieszczą się w jego godzinach otwarcia i zamknięcia, w celu zapobieżenia błędnym zapisom w tabeli rezerwacje.
* **Opis działania:**
  1. Przy każdej próbie wstawienia nowego rekordu do tabeli rezerwacje trigger pobiera godziny otwarcia (godzina\_otwarcia) i zamknięcia (godzina\_zamkniecia) kortu na podstawie jego identyfikatora (id\_kortu).
  2. Następnie sprawdza, czy godzina rozpoczęcia rezerwacji (godzina\_start) i godzina jej zakończenia (godzina\_koniec) mieszczą się w tych godzinach.
  3. Jeśli godziny rezerwacji wykraczają poza zakres, trigger zgłasza błąd i przerywa operację wstawienia.  
     **Komunikat błędu:**  
     *"Rezerwacja jest poza godzinami otwarcia dla tego kortu."*
* **Użycie:**
  1. Przed wstawieniem rekordu do tabeli rezerwacje użytkownik musi upewnić się, że godziny rezerwacji są zgodne z godzinami pracy kortu.
  2. Jeśli operacja zostanie przerwana, komunikat błędu wyjaśni przyczynę problemu.
* **Przykład błędnej rezerwacji:**  
  **Zapytanie:**

**INSERT INTO rezerwacje (id\_kortu, godzina\_start, godzina\_koniec)**

**VALUES (1, '18:00:00', '20:00:00');**

**Efekt:**  
Trigger przerwie operację, jeśli godziny otwarcia kortu to 08:00:00 - 16:00:00.  
**Komunikat błędu:**  
"Rezerwacja jest poza godzinami otwarcia dla tego kortu."

**5. Przykłady użycia**

**5.1. Tworzenie nowego użytkownika**

**INSERT INTO uzytkownicy (imie, nazwisko, email, telefon)**

**VALUES ('Wojciech', 'Barnas', 'ojwojtek@example.com', '123456789');**

**5.2. Rezerwacja kortu**

**CALL dodaj\_rezerwacje(1, 4, '2024-12-26', '14:00');**

**5.3. Sprawdzenie dostępności kortów**

**CALL sprawdz\_dostepnosc\_kortow('2024-12-26', '12:00' , '14:00' );**

**5.4. Usunięcie rezerwacji**

**CALL usun\_rezerwacje(3);**

**5.5. Opłacenie płatności**

**CALL oplac\_platnosc(3);**

1. **Diagram ERD bazy danych**

**Obraz zawierający tekst, zrzut ekranu, numer, Równolegle

Opis wygenerowany automatycznie**