

Autorzy:

Michał Radziwiłko

Paweł Szczęch

Bazy danych 2 – projekt Zakład bukmacherski

Etap 1

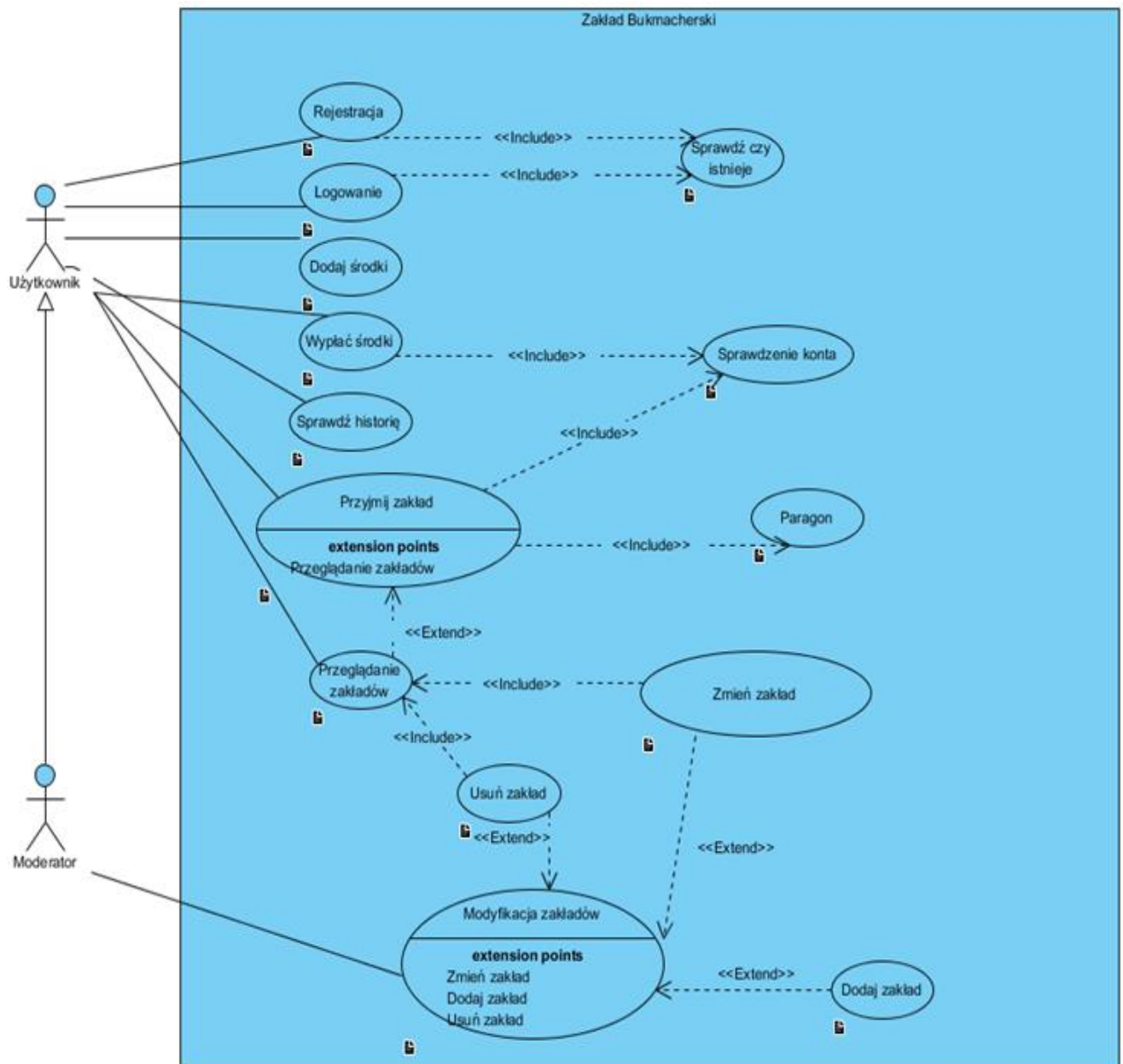
Wymagania funkcjonalne

1. Aplikacja umożliwia rejestrację użytkownika oraz logowanie do systemu.
2. Aplikacja umożliwia obstawianie wyników różnych wydarzeń sportowych osobom zarejestrowanym, o ile wydarzenia te są dostępne w aplikacji.
3. Dostępne do obstawiania wydarzenia sportowe są dodawane przez moderatorów aplikacji. Minione wydarzenia znikają 2 godziny po ich zakończeniu (np. po zakończeniu meczu).
4. Użytkownik ma możliwość zasilenia swojego konta w systemie za pomocą realnej waluty. Środki na koncie może wykorzystać jedynie do gry bukmacherskiej.
5. Obstawianie wyników wydarzeń jest możliwe od momentu dodania wydarzenia do aplikacji. Obstawianie wyników zostaje automatycznie zablokowane 10 minut przed rozpoczęciem się wydarzenia w realnym świecie.
6. Użytkownik może wykonać jednorazowo 1 serię zakładów zawierającą jednorazowo maksymalnie 5 pojedynczych zakładów oraz tylko wtedy, gdy posiada wystarczające fundusze.
7. Zakład bukmacherski może łatwo utworzyć, zmienić i usunąć informacje o wydarzeniach sportowych, zakładach oraz zlecających zakłady.
8. Użytkownik otrzymuje potwierdzenie obstawianego zakładu z informacjami o zakładzie, kursie oraz postawionej kwocie.
9. Użytkownik ma dostęp do historii zakładów z ostatnich 30 dni.

Wymagania niefunkcjonalne

1. System powinien pracować w systemach obsługujących technologię java oraz oferować nowoczesny graficzny interfejs.
2. System nie powinien udostępniać moderatorom informacji o stanie konta użytkowników.
3. Minimalna kwota postawienia na 1 zakład to 2 zł.

Diagram przypadków użycia



Scenariusze przypadków użycia

Nazwa PU: Rejestracja.

Cel: Utworzenie nowego użytkownika w bazie danych.

Warunki początkowe: Uruchomienie aplikacji, wybranie opcji Rejestracja.

Warunki końcowe: Dodanie nowego użytkownika do bazy, jeśli dane są wolne.

Przebieg:

1.

- a) Wprowadzenie danych przez użytkownika: login, imię, nazwisko, hasło, e-mail.
- b) Wywołanie PU Sprawdź czy istnieje.
- c) Jeśli PU Sprawdź czy istnieje zwraca wynik negatywny, tworzone jest nowe konto i umieszczane w bazie.
- d) Wyświetlenie komunikatu potwierdzającego rejestrację.

2.

- a) Wprowadzenie danych przez użytkownika: login, imię, nazwisko, hasło, e-mail.
- b) Wywołanie PU Sprawdź czy istnieje.
- c) Jeśli PU Sprawdź czy istnieje zwraca wynik pozytywny, rejestracja jest niemożliwa, system odmawia utworzenia nowego konta
- d) Wyświetlenie komunikatu o niepowodzeniu.

Nazwa PU: Logowanie

Cel: Zalogowanie się użytkownika.

Warunki początkowe: Uruchomienie aplikacji, wybranie opcji Logowanie.

Warunki końcowe: Jeśli podane dane są poprawne, zalogowanie się do systemu aplikacji.

Przebieg:

1.

- a) Podanie danych
- b) Wywołanie PU Sprawdź czy istnieje.
- c) Jeśli PU Sprawdź czy istnieje zwraca wynik pozytywny aplikacja loguje użytkownika
- d) Wyświetlenie komunikatu potwierdzającego oraz przeniesie do panelu użytkownika.

2.

- a) Podanie danych
- b) Wywołanie PU Sprawdź czy istnieje.
- c) Jeśli PU Sprawdź czy istnieje zwraca wynik negatywny aplikacja odmawia dostępu użytkownikowi
- d) Wyświetlenie komunikatu odmowy.

Nazwa PU: Sprawdź czy istnieje

Cel: Sprawdzenie, czy dany użytkownik istnieje w bazie.

Warunki początkowe: Wywoływany przez PU Logowanie lub PU Rejestracja.

Warunki końcowe: Zwrócenie wyniku przeszukiwania bazy danych.

Przebieg:

- 1. Szukanie podanych danych przez użytkownika w bazie danych, jeśli zostaną odnalezione zwracany jest wynik pozytywny.
- 2. Szukanie podanych danych przez użytkownika w bazie danych, jeśli nie zostaną odnalezione zwracany jest wynik negatywny.

Nazwa PU: Dodaj środki

Cel: Dodanie pieniędzy do konta użytkownika.

Warunki początkowe: Wymaga wcześniejszego zalogowania przez PU Logowanie

Warunki końcowe: Dodanie pieniędzy do konta użytkownika.

Przebieg:

1.

- a) W nowym oknie użytkownik podaje kwotę oraz metodę płatności.
- b) Dokonanie płatności w wybranej opcji.
- c) Wyświetlenie potwierdzenia po poprawnie przeprowadzonej akcji.

2.

- a) W nowym oknie użytkownik podaje kwotę oraz metodę płatności.
- b) Anulowanie wykonania płatności.
- c) Wyświetlenie komunikatu potwierdzającego anulowanie.

Nazwa PU: Wypląć środki

Cel: Wypłacenie pieniędzy do konta użytkownika.

Warunki początkowe: Wymaga wcześniejszego zalogowania przez PU
Logowanie

Warunki końcowe: Wypłacenie pieniędzy do konta użytkownika.

Przebieg:

1.

- a) W nowym oknie użytkownik podaje kwotę oraz metodę.
- b) Wykonanie działania w wybranej opcji.
- c) Wywołanie PU Sprawdzenie konta, otrzymanie wyniku pozytywnego.
- d) Wyświetlenie potwierdzenia po poprawnie przeprowadzonej akcji.

2.

- a) W nowym oknie użytkownik podaje kwotę oraz metodę.
- b) Wykonanie działania w wybranej opcji.
- c) Wywołanie PU Sprawdzenie konta, otrzymanie wyniku negatywnego.
- d) Wyświetlenie komunikatu o odmowie.

3.

- a) W nowym oknie użytkownik podaje kwotę oraz metodę.
- b) Anulowanie wykonania.
- c) Wyświetlenie komunikatu potwierdzającego anulowanie.

Nazwa PU: Sprawdź historię

Cel: Sprawdzenie zakładów z ostatnich 30 dni.

Warunki początkowe: Wymaga wcześniejszego zalogowania przez PU Logowanie.

Warunki końcowe: Zwraca historię zakładów wraz z wynikami.

Przebieg:

1. Wyświetlenie informacji o wykonanych zakładach.

Nazwa PU: Przyjmij zakład

Cel: Dokonanie serii zakładów

Warunki początkowe: Wymaga wcześniejszego zalogowania przez PU Logowanie.

Warunki końcowe: Utworzenie nowych danych informujących o postawionych dyscyplinach, meczach, drużynach, ilościach zakładów, kwocie, datach, możliwej wygranej

Przebiegi:

1.
 - a) Wpisanie kodu zakładu oraz przeznaczonej kwoty.
 - b) Jeśli są spełnione wymagania: minimalna kwota 2 zł, maksymalnie 5 zakładów oraz wywołanie PU Sprawdzenie konta zwraca wynik pozytywny, system pozwala na kontynuację.
 - c) Użytkownik może wybrać czy chce dodawać kolejny zakład (przejdzie do kroku a lub zakończyć kupon.
 - d) System wykonuje operacje na danych oraz wywołuje PU Paragon.
2.
 - a) Użytkownik wybiera zakład z listy oraz przeznaczoną kwotę.
 - b) Jeśli któreś z wymagań nie jest spełnione: minimalna kwota 2 zł, maksymalnie 5 zakładów lub wywołanie PU Sprawdzenie konta zwraca wynik negatywny, system pozwala odmawia wykonania dalszego obstawiania.

- c) Jeśli zostały utworzone jakiekolwiek zakłady system wykonuje operacje na danych oraz wywołuje PU Paragon, w innym wypadku wyświetla komunikat o niepowodzeniu.

Nazwa PU: Przeglądanie zakładów

Cel: Wyświetlenie możliwości obstawiania dostępnych w bazie.

Warunki początkowe: Wymaga wcześniejszego zalogowania przez PU Logowanie lub wywoływane przez PU Zmień zakład.

Warunki końcowe: Wyświetlenie listy z zakładami.

Przebieg:

1. Wyświetlenie listy zakładów.

Nazwa PU: Sprawdzenie konta

Cel: Sprawdzenie czy na koncie użytkownika jest wystarczająca kwota.

Warunki początkowe: Wywołanie przez PU Wypląć środki lub Przyjmij zakład.

Warunki końcowe: Wynik porównania pozytywny lub negatywny.

Przebieg:

1. Zwrócenie wyników porównania kwot: pozytywnego, gdy na koncie są wystarczające środki oraz negatywnego w przeciwnym wypadku.

Nazwa PU: Paragon

Cel: Wyświetlenie listy wykonanych zakładów

Warunki początkowe: Wywołane przez PU Przyjmij zakład

Warunki końcowe: Wyświetlenie listy wykonanych zakładów.

Przebieg:

1.

- a) Wyświetlenie wykonanych zakładów: dyscyplin, drużyn, postawionych kwot oraz możliwej wygranej.
- b) Zapisanie Paragonu.

Nazwa PU: Modyfikacja zakładów

Cel: Wyświetlenie opcji operacji na zakładach.

Warunki początkowe: Wymaga wcześniejszego zalogowania przez PU Logowanie na konto z uprawnieniami moderatora.

Warunki końcowe: Wyświetlenie okna z opcjami modyfikacji.

Przebieg:

1.

- a) Dla użytkowników z uprawnieniami moderatora system umożliwia wybranej dodatkowej opcji modyfikowania zakładów.
- b) Wywołanie odpowiednio wybranej opcji: PU Zmień zakład, PU Dodaj Zakład, PU Usuń zakład

Nazwa PU: Zmień zakład

Cel: Modyfikacja istniejącego już zakładu.

Warunki początkowe: Wywołane przez PU Modyfikacja zakładów.

Warunki końcowe: Zmiana danych o podanym zakładzie w bazie.

Przebieg:

1.

- a) Wywołanie PU Przeglądanie zakładów.
- b) Wybranie zakładu przez moderatora.
- c) Zmiana wybranych informacji.
- d) Wyświetlenie informacji potwierdzającej zmianę

2.

- a) Wywołanie PU Przeglądanie zakładów.
- b) Wybranie zakładu przez moderatora.
- c) Anulowanie.
- d) Wyświetlenie komunikatu o anulowaniu.

Nazwa PU: Dodaj zakład

Cel: Dodanie nowego zakładu.

Warunki początkowe: Wywołane przez PU Modyfikacja zakładów.

Warunki końcowe: Dodanie nowych danych do bazy.

Przebieg:

1.

- a) Wprowadzenie danych przez moderatora.
- b) Potwierdzenie przez moderatora nowego zakładu.
- c) Dodanie nowych danych do bazy oraz wyświetlenie komunikatu potwierdzającego

2.

- a) Wprowadzenie danych przez moderatora.
- b) Anulowanie przez moderatora nowego zakładu.
- c) Wyświetlenie komunikatu o anulacji.

Nazwa PU: Usuń zakład

Cel: Usunięcie wybranego istniejącego zakładu.

Warunki początkowe: Wywołane przez PU Modyfikacja zakładów.

Warunki końcowe: Usunięcie danych z bazy.

Przebieg:

1.

- a) Wywołanie PU Przeglądanie zakładów.
- b) Wybranie zakładu przez moderatora i akceptacja.
- c) Wyświetlenie informacji potwierdzającej usunięcie.

2.

- a) Wywołanie PU Przeglądanie zakładów.
- b) Wybranie zakładu przez moderatora.
- c) Anulowanie.
- d) Wyświetlenie komunikatu o anulowaniu.

Określenie tabel

Użytkownik - ID_użytkownika, imię, nazwisko, e-mail, login, hasło

Konto - ID_konta, UżytkownikID_użytkownika, UprawnieniaID_uprawnienia, środki

Uprawnienia - ID_uprawnienia, nazwa, poziom_dostępu

Dyscypliny - ID_dyscypliny, nazwa, rodzaj

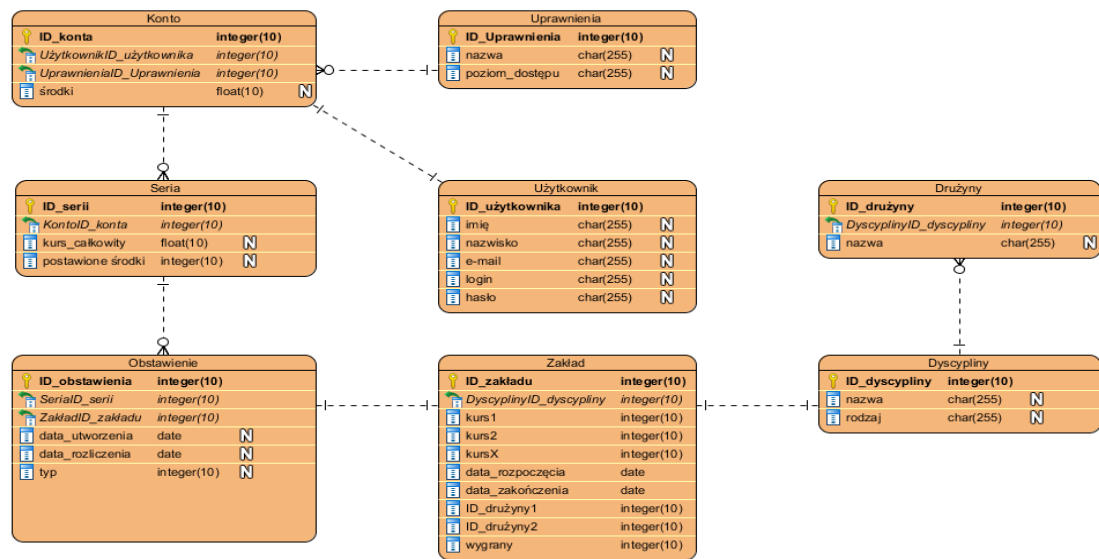
Drużyny - ID_drużyny, DyscyplinyID_dyscypliny, nazwa

Zakład - ID_zakładu, ID_drużyny1, ID_drużyny2, DyscyplinyID_dyscypliny, kurs1, kurs2, kursX, data_rozpoczęcia, data_zakończenia, wygrany

Seria - ID_serii, KontoID_konta, kurs_całkowity, postawione środki

Obstawienie - ID_obstawienia, SerialID_serii, ZakładID_zakładu, data_utworzenia, data_rozliczenia, typ

Diagram związków



ETAP 2

Analiza liczby instancji każdej encji

Tabela	Maksymalna liczba wierszy	Średnia liczba wierszy	Rozmiar kolumn	Maksymalny rozmiar tabeli	Średni rozmiar tabeli
Użytkownik	2 000	1 000	6	12 000	6 000
Konto	2 000	1 000	4	8 000	4 000
Uprawnienia	2 000	1 000	3	6 000	3 000
Obstawianie	80 000	40 000	6	480 000	240 000
Seria	20 000	10 000	4	80 000	40 000
Zakład	10 000	5 000	10	100 000	50 000
Drużyny	2 000	1 000	3	6 000	3 000
Dyscypliny	50	25	3	150	75

Analiza użycia identyfikująca podstawowe rodzaje transakcji

Tabela	Rodzaje transakcji
Użytkownik	dodawanie, modyfikowanie, wyszukiwanie
Konto	modyfikowanie, wyszukiwanie
Uprawnienia	modyfikowanie, wyszukiwanie
Obstawianie	wstawianie, wyszukiwanie
Seria	wyszukiwanie, wstawianie
Zakład	wyszukiwanie, wstawianie, modyfikowanie, usuwanie
Drużyny	wyszukiwanie, wstawianie, usuwanie, modyfikowanie
Dyscypliny	wyszukiwanie, wstawianie, usuwanie, modyfikowanie

Sformułowanie wymagań dotyczących dostępu – określenie częstości wykonania operacji na danych

Opisy przebiegów operacji wymienionych w tabeli częstotliwości ich wykonania zamieszczonej poniżej:

Ad 1. Raport historii zakładów to operacja wywoływana na żądanie użytkownika, polega ona na uzyskaniu zapamiętanych informacji o seriach zakładów wykonanych przez użytkownika w ciągu ostatnich 30 dni – potrzebne są tabele: *Użytkownik, Obstawienie, Seria*.

Ad 2. Rejestracja użytkownika to operacja dodająca rekord do tabel *Użytkownik, Konto* oraz zdefiniowaniem uprawnień nowego użytkownika znajdujących się w tabeli *Uprawnienia*.

Ad 3. Utworzenie paragonu jest operacją wykorzystującą dane z tabel *Obstawienie, Seria, Użytkownik*. Jest ona wywoływana jednorazowo oraz automatycznie w chwili wykonania serii zakładów.

Ad 4. Przeglądanie zakładów to operacja wyświetlająca wszystkie rodzaje zakładów, znajdujące się w bazie danych. Jest ona wywoływana na żądanie użytkownika lub moderatora – potrzebne są tabele: *Zakład, Dyscypliny, Drużyny*.

Ad 5. Wykonanie serii zakładów jest operacją dodającą rekord do tabeli *Seria*, oraz *Obstawienie*, na żądanie użytkownika – potrzebne są tabele: *Użytkownik, Seria, Obstawienie, Zakład*.

Ad 6. Dodanie zakładu to operacja dodająca rekord do tabeli *Zakład*, która jest wykonywana na żądanie moderatora – potrzebne są tabele: *Użytkownik, Uprawnienia, Zakład, Drużyny*.

Ad 7. Dodanie drużyny to operacja taka sama jak dodanie zakładu, z tą różnicą, że rekord dodawany jest do tabeli *Drużyny* – potrzebne są tabele: *Użytkownik, Uprawnienia, Drużyny*.

Ad 8. Dodanie dyscypliny to operacja taka sama jak dodanie zakładu, z tą różnicą, że rekord dodawany jest do tabeli *Dyscypliny* – potrzebne są tabele: *Użytkownik, Uprawnienia, Dyscypliny*.

Lp.	Operacja	Maksymalna częstotliwość wykonywania	Maksymalny czas wykonania zapytania
1	Raport historii zakładów – na żądanie	Do 5000 razy dziennie	3 s
2	Rejestracja użytkownika	Do 200 razy dziennie	3 s
3	Utworzenie paragonu	Do 2000 razy dziennie	3 s
4	Przeglądanie zakładów	Do 8000 razy dziennie	3 s
5	Wykonanie serii zakładów	Do 2000 razy dziennie	3 s
6	Dodanie zakładu	Do 200 razy dziennie	3 s
7	Dodanie drużyny	Do 50 razy dziennie	3 s
8	Dodanie dyscypliny	Do 50 razy dziennie	3 s

Analiza i obsługa więzów integralności

Integralność łączy w sobie formalną poprawność bazy danych i procesów przetwarzania, poprawność fizycznej organizacji danych, zgodność ze schematem bazy danych, zgodność z ograniczeniami integralności oraz z regułami dostępu.

Wynikiem klasycznej analizy danych jest logiczny projekt danych zawierający specyfikację odpowiednich plików i zbiór wewnętrznych więzów integralności, czyli warunków, które powinny być spełnione przez określony podzbiór danych z bazy. Spełnienie tych warunków świadczy, że baza danych jest w stanie spójnym.

Integralność encji

Integralność encji jest zachowana, jeśli każda tabela posiada klucz główny, a jego wartości w ramach tabeli są unikatowe i różne od wartości NULL (to zapewni nam, że krotki w tabeli nie będą się powtarzać). Integralność encji jest charakterystyczna dla modelu relacyjnego baz danych. W naszym projekcie jest to zachowane poprzez umieszczenie identyfikatorów w każdej tabeli.

Integralność krotki

Każda krotka opisuje jeden element świata rzeczywistego. Na wartości krotki można nałożyć niezależne więzy (ograniczenia), które muszą być spełnione przez wszystkie krotki niezależnie. Wiązami tymi są:

- zawężenie dziedziny atrybutu poprzez podanie przedziału wartości, listy możliwych zawartości
- podanie formatu wartości [ID-konta – Integer(10)]
- zadeklarowanie konieczności występowania jakiejś wartości (NOT NULL) – każde konto musi zawierać login, który służy do logowania
- zdefiniowanie niepowtarzalnych wartości atrybutu (UNIQUE) - nie można użyć tego samego adresu e-mail wielokrotnie

Więzy wewnętrzne relacji: Sprawdzane są wartości występujących w krotkach w ramach tej samej relacji.

Integralność odwołań

Odnosi się do związku pomiędzy rzeczywistymi obiektami podczas modelowania. W relacyjnych bazach danych tworzona jest na kluczu pierwotnym oraz odpowiadającym mu kluczem obcym. Łącząc kolumny tych samych typów danych z różnych tabel, a więc tworząc klucz obcy definiujemy związek między tymi dwoma kluczami. W relacyjnej bazie danych mówimy o sytuacjach, gdy klucz obcy jednej tabeli jest również kluczem pierwotnym innej tabeli. Z warunku integralności wynika, że dla każdego wiersza pierwszej tabeli musi istnieć taki wiersz w tabeli drugiej, by ich wartości były jednakowe.

Naruszenie integralności może nastąpić w sytuacji, gdy klucz główny, który jest kluczem obcym w innej zostanie usunięty. By rozwiązać ten problem i uniknąć odwołań do nieistniejących elementów można zastosować różne metody. W naszej aplikacji operacje usuwania będą bardzo występowały rzadko. Potrzeba ich wywołania będzie raczej występowała w szczególnych przypadkach, więc stosując odpowiednie rozwiązania, min. wpisując wartości NULL zamiast stosowania odwołań do konkretnego wiersza w innej tabeli szansa wystąpienia problemów jest stosunkowo niska.

Dostrajanie bazy danych pod względem wydajności

Asercje

Asercje służą do kontroli wpisywanych do tabel wartości. Są stosowane, by sprawdzać lub ograniczać czy dane wartości pól spełniają wymagania. W projekcie aplikacji zakładu bukmacherskiego tego typu sprawdzanie można stosować w wielu przypadkach, np.:

- W tabeli Użytkownik e-mail musi posiadać znak '@' oraz po nim kropkę pomiędzy znakami
- W tabeli Użytkownik login musi spełniać wymaganą długość
- W tabeli Użytkownik hasło musi spełniać wymagania znaków (przynajmniej jedna litera różniąca się wielkością lub przynajmniej jedna cyfra)
- W tabeli Seria postawione środki nie mogą być ujemne
- W tabeli Obstawienie typ może mieć tylko wartości [1,2,3]
- W tabelach Obstawienie i Zakład daty rozpoczęcia muszą mieć wcześniejszą datę od daty zakończenia

Indeksowanie

Wcześniejsza analiza przedstawiła nam, które operacje będą najczęściej wykonywane i na jej podstawie postanowiliśmy, że wprowadzimy indeksy przyspieszające wyszukiwanie zakładów. Zakładamy utworzenie indeksów wykorzystując pola daty rozpoczęcia oraz dyscypliny.

Uważamy, że w naszym projekcie nie ma potrzeby używania innych metod.

Implementacja bazy danych

ETAP 3

1. Fizyczny projekt bazy danych

Baza danych została zaimplementowana w języku SQL przy pomocy środowiska deweloperskiego Oracle Database. Sposób tworzenia tabel został przedstawiony na przykładzie tabeli *Zaklad*. Kod tworzący tę tabelę znajduje się poniżej.

```
CREATE TABLE "HAZARD1"."ZAKLAD"
(
    "ID_ZAKLADU" NUMBER(10,0) NOT NULL ENABLE,
    "ID_DYSCYPLINY" NUMBER(10,0) NOT NULL ENABLE,
    "KURS1" FLOAT(2) NOT NULL ENABLE,
    "KURS2" FLOAT(2) NOT NULL ENABLE,
    "KURS3" FLOAT(2) NOT NULL ENABLE,
    "DATA_ROZPOCZECIA" DATE NOT NULL ENABLE,
    "DATA_ZAKONCZENIA" DATE NOT NULL ENABLE,
    "ID_DRUZYNY1" NUMBER(10,0) NOT NULL ENABLE,
    "ID_DRUZYNY2" NUMBER(10,0) NOT NULL ENABLE,
    "WYGRANY" NUMBER(1,0),
    CONSTRAINT "ZAKLAD_PK" PRIMARY KEY
("ID_ZAKLADU") ENABLE,
    CONSTRAINT "ZAKLAD_FK_DYSCYPLINY" FOREIGN KEY
("ID_DYSCYPLINY")
REFERENCES "HAZARD1"."DYSCYPLINY"
("ID_DYSCYPLINY") ENABLE,
    CONSTRAINT "ZAKLAD_FK2_DRUZYNY1" FOREIGN KEY
("ID_DRUZYNY1")
REFERENCES "HAZARD1"."DRUZYNY" ("ID_DRUZYNY")
    CONSTRAINT "ZAKLAD_FK3_DRUZYNY2" FOREIGN KEY
("ID_DRUZYNY2")
REFERENCES "HAZARD1"."DRUZYNY" ("ID_DRUZYNY")
ENABLE;
```

```
--indeks
CREATE INDEX IDX_ZAKLAD ON zaklad (DATA_ROZPOCZECIA,
DATA_ZAKONCZENIA, ID_DYSCYPLINY);
```

2. Zbiór zapytań zoptymalizowanych

Po testach częstych zapytań do bazy danych stwierdziliśmy, że indeksowanie tabel nie ma wpływu na czas wykonywania zapytań. Z tego powodu nie ma potrzeby jego stosowania.

Testy były przeprowadzane na losowo wygenerowanych danych. Największa ilość danych w tabeli, to 800 krotek.

Przykładowe zapytanie do bazy danych wykorzystane do testów, to zapytanie wyświetlające historię wszystkich obstawień użytkownika. Wygląda ono następująco:

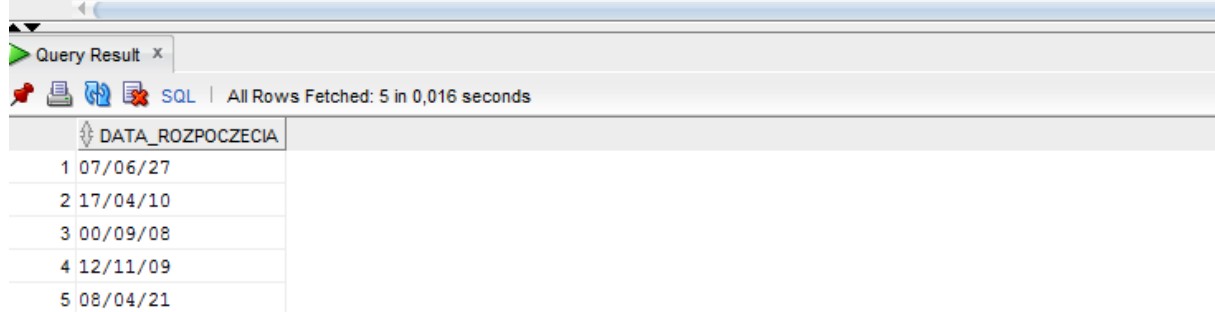
```
SELECT
    obstawienie.DATA_UTWORZENIA, obstawienie.DATA_ROZLICZENIA,
    obstawienie.TYP, seria.KURS_CALKOWITY, seria.POSTAWIONE_SRODKI, zaklad.WYGRANY
FROM uzytkownik
join konto using(id_uzytkownika)
join seria using(id_konta)
join obstawienie using(id_serii)
join zaklad using (id_zakladu)
WHERE uzytkownik.imie = 'jwjyfxpd';
```

	DATA_UTWORZENIA	DATA_ROZLICZENIA	TYP	KURS_CALKOWITY	POSTAWIONE_SRODKI	WYGRANY
1	06/03/16	00/03/11	2	9	20	1
2	10/06/14	03/05/14	3	20	80	1
3	14/04/22	12/12/06	2	9	20	2
4	07/05/16	07/11/19	3	20	80	1
5	04/03/21	13/03/02	1	20	80	2

Wszystkie zapytania, które stworzyliśmy z uwagą na to, jakich będziemy używać w aplikacji wykonywały się bardzo szybko. Przykład poniżej:

```
SELECT zaklad.DATA_ROZPOCZECIA
FROM uzytkownik
join konto using(id_uzytkownika)
join seria using(id_konta)
join obstawienie using(id_serii)
join zaklad using (id_zakladu)
WHERE uzytkownik.imie = 'twdiclma' and zaklad.DATA_ZAKONCZENIA BETWEEN '12/01/01' AND '15/12/31';

SELECT *
```



The screenshot shows a database query result window. The query is a complex JOIN statement filtering for a specific user and a date range. The result set, titled 'Query Result', shows 5 rows of data for the column 'DATA_ROZPOCZECIA'. The status bar indicates 'All Rows Fetched: 5 in 0,016 seconds'.

	DATA_ROZPOCZECIA
1	07/06/27
2	17/04/10
3	00/09/08
4	12/11/09
5	08/04/21

3. Polityka bezpieczeństwa

W naszej aplikacji zastosowane zostaną następujące metody ochrony procesu logowania:

- Czas oczekiwania pomiędzy nieudanymi próbami logowania - zapobiegnie atakom typu brute force lub podobnym i złamaniem hasła
- Sposób odzyskiwania hasła, za pomocą maila użytkownika, lub pytania pomocniczego.
- Przy tworzeniu użytkownika w aplikacji hasło będzie haszowane, co uniemożliwi jego użycie w razie wycieku lub kradzieży danych.
- Okresowe wymuszanie zmiany hasła poprawi bezpieczeństwo konta użytkownika.
- Wszystkie hasła muszą się składać, z co najmniej 8 znaków, w tym co najmniej 1 numeru, 1 litery dużej oraz małej i 1 znaku specjalnego.

- Jest utworzona tabela *Uprawnienia*, która służy do weryfikacji dostępu do danych, oraz weryfikacji możliwości ich edycji. Dzięki niej jedynie moderator ma możliwość dodawania, edycji oraz usuwania zakładów.