

Projekt – Bazy danych

System zarządzania kolekcją gier planszowych

Krystian Kościelecki (169427)

## 1.Określenie tematyki i zakresu projektu

- Temat: System zarządzania kolekcją gier planszowych
- Zakres:
  1. Katalogowanie gier według mechanik (deck-building, worker placement, dice-rolling, trading)
  2. Opis mechanik gry
  3. Filtrowanie gier według wybranych mechanik
  4. Śledzenie historii wydań
  5. Zbieranie recenzji użytkowników

## 2.Przedstawienie, zagadnień związanych z tematem

- Struktura danych
  1. **Encje:**
    - a. Gry
    - b. Mechaniki
    - c. Historia wydań
    - d. Recenzje
  2. **Relacje:**
    - a. Gry-mechaniki
    - b. Gry – Wydania
    - c. Gry – Recenzje
  3. **Funkcje:**
    - a. Filtrowanie gier
    - b. Śledzenie historii wydań
    - c. Zbieranie recenzji użytkowników

## 3. Określenie funkcji bazy danych i ich priorytetu

1. Katalogowanie gier, mechanik i historii wydań:

Opis:

System powinien umożliwiać katalogowanie gier, mechanik i ich historii wydań, aby użytkownicy mogli w łatwy sposób znaleźć i poznać szczegółowe informacje o interesujących ich grach.

Priorytet:

Wysoki. Katalogowanie stanowi podstawową funkcję systemu, zapewniając dostęp do wszystkich informacji o grach.

## 2. Zbieranie i wyświetlanie recenzji:

Opis:

Funkcja ta umożliwia użytkownikom dodawanie recenzji gier oraz przeglądanie opinii innych graczy.

Priorytet:

Wysoki. Recenzje mają znaczący wpływ na ocenę i wybór gry, dlatego powinny być dostępne dla każdej gry w systemie.

## 3. Podstawowe operacje CRUD (Create, Read, Update, Delete) dla zarządzania danymi gier:

Opis:

System powinien umożliwiać tworzenie, odczytywanie, aktualizowanie i usuwanie danych dotyczących gier, mechanik, historii wydań i recenzji. Te operacje są podstawą zarządzania i utrzymywania aktualnej bazy danych.

Priorytet:

Wysoki. Operacje CRUD są niezbędne do zarządzania wszystkimi elementami bazy danych.

## 4. Filtrowanie gier

Opis:

Funkcja umożliwiająca łatwiejsze poruszanie się po spisie gier.

Priorytet:

Niski. Funkcja pomagająca użytkownikom poruszanie się po systemie.

## 4. wybór technologii i rodzaju bazy danych wybór narzędzi

- **Rodzaj Baz Danych:**

Relacyjna baza danych (RDBMS) ponieważ zawiera uporządkowane dane.

- **Technologia:**

MySQL jako baza danych, a MySQL Workbench do projektowania, zarządzania i implementacji.

- **Język programowania(opcjonalnie):**

JavaScript poprzez wykorzystanie frameworku Node.js

## 5. prezentacja przygotowanego repozytorium z opisem.

[https://github.com/ermir2001/bazy\\_danych\\_projekt/tree/main](https://github.com/ermir2001/bazy_danych_projekt/tree/main)

## 6. prezentacja diagramu bazy danych

## 7. opis tabel bazy danych i ich funkcji

### Tabela games

- **Funkcja:** Tabela games przechowuje informacje o grach, w tym ich nazwy i opisy.
- **Kolumny:**
  - id\_games: Unikalny identyfikator gry (klucz główny).
  - name: Nazwa gry.
  - description: Opcjonalny opis gry.

### Tabela mechanics

- **Funkcja:** Zawiera informacje na temat różnych mechanik, które mogą być wykorzystywane w grach, takich jak handel, umieszczanie kafelków itp.
- **Kolumny:**
  - id\_mechanics: Unikalny identyfikator mechaniki (klucz główny).
  - name: Nazwa mechaniki.
  - description: Opcjonalny opis mechaniki.

### Tabela game\_mechanics

- **Funkcja:** Jest to tabela łącząca, która tworzy relację wiele-do-wielu między grami a mechanikami. Pozwala to na przypisanie wielu mechanik do jednej gry i na odwrót.
- **Kolumny:**
  - id\_game: Identyfikator gry (klucz obcy).
  - id\_mechanic: Identyfikator mechaniki (klucz obcy).

- **Relacje:**
  - Klucze obce odnoszą się odpowiednio do tabel games i mechanics.

#### **Tabela user**

- **Funkcja:** Przechowuje informacje o użytkownikach, którzy mogą recenzować gry.
- **Kolumny:**
  - id\_user: Unikalny identyfikator użytkownika (klucz główny).
  - username: Nazwa użytkownika.
  - email: Adres email użytkownika.
  - password: Hasło użytkownika.
  - first\_name: Imię użytkownika.
  - last\_name: Nazwisko użytkownika.

#### **Tabela game\_reviews**

- **Funkcja:** Przechowuje recenzje gier dodane przez użytkowników, wraz z ocenami.
- **Kolumny:**
  - id\_game\_reviews: Unikalny identyfikator recenzji (klucz główny).
  - id\_game: Identyfikator gry, której dotyczy recenzja (klucz obcy).
  - id\_user: Identyfikator użytkownika, który dodał recenzję (klucz obcy).
  - rating: Ocena gry.
  - review: Tekst recenzji.
- **Relacje:**
  - Klucze obce odnoszą się odpowiednio do tabel games i user.

#### **Tabela publisher**

- **Funkcja:** Zawiera informacje o wydawcach gier.
- **Kolumny:**
  - id\_publisher: Unikalny identyfikator wydawcy (klucz główny).
  - name: Nazwa wydawcy.

#### **Tabela versions**

- **Funkcja:** Zawiera informacje o różnych wersjach gier, w tym o numerach wersji, datach wydania i wydawcach.
- **Kolumny:**

- id\_versions: Unikalny identyfikator wersji (klucz główny).
  - id\_game: Identyfikator gry, której dotyczy wersja (klucz obcy).
  - version\_number: Numer wersji.
  - release\_date: Data wydania wersji.
  - id\_publisher: Identyfikator wydawcy, który wydał wersję (klucz obcy).
- **Relacje:**
    - Klucze obce odnoszą się odpowiednio do tabel games i publisher.

## 8. prezentacja wykonania bazy danych (SQL, diagram)

### 9. proste zapytania SQL (dodanie danych, selekcja)

```
INSERT INTO `boardgames`.`games`
(`id_games`, `name`, `description`)
VALUES
  (1, 'Catan', 'Strategiczna gra planszowa, w której gracze zdobywają zasoby i budują osady'),
  (2, 'Carcassonne', 'Gra polegająca na układaniu kafelków i kontrolowaniu terenu'),
  (3, 'Pandemic', 'Kooperacyjna gra, w której gracze walczą z rozprzestrzeniającymi się chorobami');

INSERT INTO `boardgames`.`mechanics`
(`id_mechanics`, `name`, `description`)
```

VALUES

```
(1, 'Trading', 'Gracze mogą wymieniać  
się zasobami między sobą'),  
(2, 'Tile Placement', 'Gracze układają  
kafelki na planszy, tworząc mapę'),  
(3, 'Cooperative Play', 'Gracze muszą  
współpracować, aby osiągnąć wspólny cel');
```

```
INSERT INTO `boardgames`.`game_mechanics`  
(`id_game`, `id_mechanic`)
```

VALUES

```
(1, 1),  
(2, 2),  
(3, 3);
```

```
INSERT INTO `boardgames`.`user`  
(`id_user`, `username`, `email`,  
`password`, `first_name`, `last_name`)
```

VALUES

```
(1, 'user1', 'user1@example.com',  
'password123', 'John', 'Doe'),  
(2, 'user2', 'user2@example.com',  
'password234', 'Jane', 'Smith'),  
(3, 'user3', 'user3@example.com',  
'password345', 'Alice', 'Johnson');
```

```
INSERT INTO `boardgames`.`game_reviews`  
(`id_game_reviews`, `id_game`, `id_user`,  
`rating`, `review`)
```

```
VALUES
```

```
    (1, 1, 1, 5, 'Świetna gra dla całej  
rodziny!'),
```

```
    (2, 2, 2, 4, 'Dobra zabawa, ale  
czasami zbyt losowa.'),
```

```
    (3, 3, 3, 5, 'Doskonała gra  
kooperacyjna, pełna wyzwań.');
```

```
INSERT INTO `boardgames`.`publisher`  
(`id_publisher`, `name`)
```

```
VALUES
```

```
    (1, 'Asmodee'),
```

```
    (2, 'Z-Man Games'),
```

```
    (3, 'Kosmos');
```

```
INSERT INTO `boardgames`.`versions`  
(`id_versions`, `id_game`,  
`version_number`, `release_date`,  
`id_publisher`)
```



VALUES

```
(1, 1, 1, '2015-05-15', 3),  
(2, 2, 1, '2016-07-20', 1),  
(3, 3, 2, '2017-09-10', 2);
```

-- Zapytanie zwraca nazwy gier, które  
używają mechaniki "Trading"

```
SELECT g.name  
FROM boardgames.games g  
INNER JOIN boardgames.game_mechanics gm ON  
g.id_games = gm.id_game  
INNER JOIN boardgames.mechanics m ON  
gm.id_mechanic = m.id_mechanics  
WHERE m.name = 'Trading';
```

-- Zapytanie zwraca nazwy gier oraz  
średnią ocen i liczbę ocen, gdzie średnia  
ocena jest równa lub wyższa niż 4

```
SELECT g.name, AVG(gr.rating) AS
average_rating, COUNT(gr.rating) AS
number_of_ratings
FROM boardgames.games g
LEFT JOIN boardgames.game_reviews gr ON
g.id_games = gr.id_game
GROUP BY g.id_games
HAVING AVG(gr.rating) >= 4;
```

-- Zapytanie zwraca nazwy użytkowników i  
liczbę napisanych przez nich recenzji,  
gdzie liczba recenzji przekracza 5

```
SELECT u.username,
COUNT(gr.id_game_reviews) AS review_count
FROM boardgames.user u
LEFT JOIN boardgames.game_reviews gr ON
u.id_user = gr.id_user
GROUP BY u.id_user
HAVING COUNT(gr.id_game_reviews) > 5;
```

-- Zapytanie zwraca nazwy gier, które  
zostały wydane przez wydawcę o nazwie  
"Asmodee"

```
SELECT g.name
FROM boardgames.games g
JOIN boardgames.versions v ON g.id_games =
v.id_game
JOIN boardgames.publisher p ON
v.id_publisher = p.id_publisher
WHERE p.name = 'Asmodee';
```

-- Zapytanie zwraca nazwy gier, których  
najnowsza wersja została wydana po 1  
stycznia 2010 roku

```
SELECT g.name, MAX(v.release_date) AS
latest_release
FROM boardgames.games g
JOIN boardgames.versions v ON g.id_games =
v.id_game
GROUP BY g.id_games
HAVING latest_release > '2010-01-01';
```

-- Zapytanie zwraca nazwy użytkowników,  
którzy napisali przynajmniej jedną  
recenzję

```
SELECT u.username,  
COUNT(gr.id_game_reviews) AS review_count  
FROM boardgames.user u  
LEFT JOIN boardgames.game_reviews gr ON  
u.id_user = gr.id_user  
WHERE gr.id_game_reviews IS NOT NULL  
GROUP BY u.id_user;
```