Bazy danych - Projekt
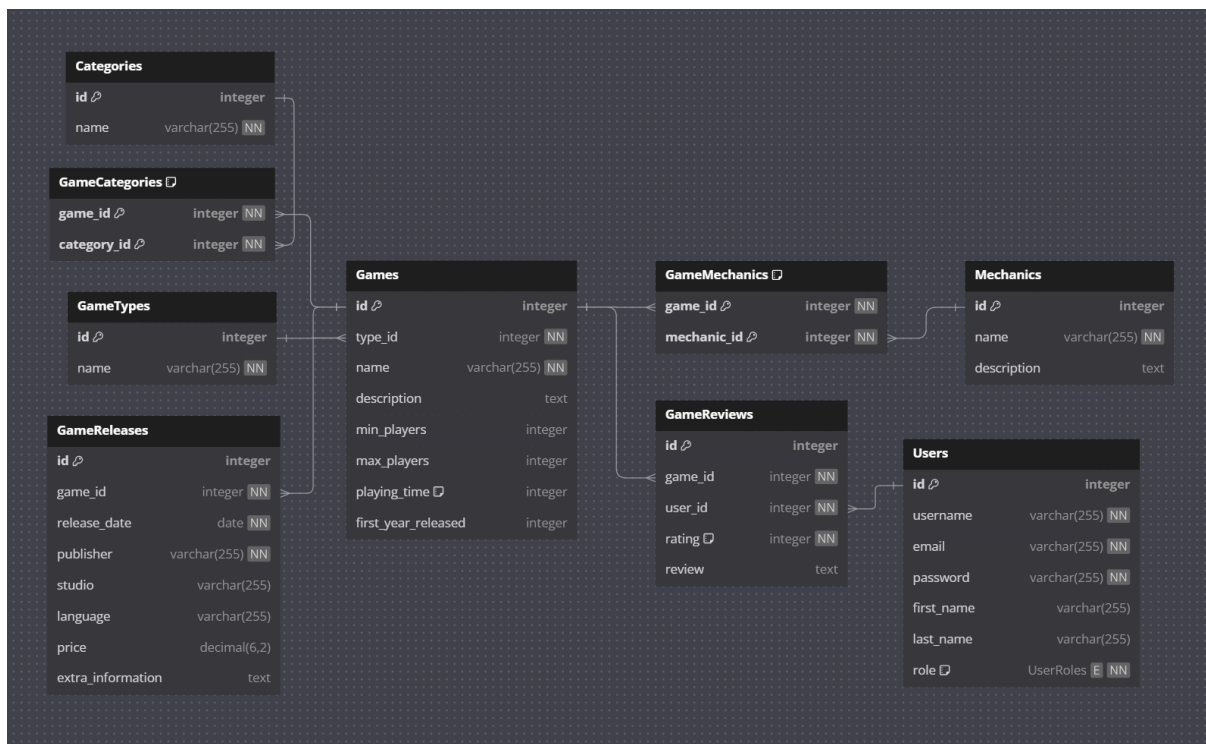
System zarządzania kolekcją gier planszowych

Część 2. Diagram DB i SQL

Prezentacja diagramu ERD

Diagram ERD przedstawia strukturę bazy danych, która będzie wykorzystywana w tworzonym projekcie. Zawiera on informacje o encjach, atrybutach oraz relacjach między nimi.

Diagram stworzyłem przy pomocy programu online „dbdiagram.io". Wybór tego narzędzia wynika z mojego wcześniejszego doświadczenia z jego użyciem oraz z użycia języka DBML do tworzenia diagramów. Link do diagramu na dbdiagram.io.

Rysunek 1: Diagram ERD bazy danych projektu.

Opis tabel i ich funkcji

Tabela `Games`

Tabela `Games` jest główną tabelą w bazie danych. Zawiera informacje o grach planszowych, takie jak nazwa, opis, minimalna i maksymalna liczba graczy, średni czas gry oraz kategorie. Kluczem głównym tej tabeli jest `id`, podobnie jak w pozostałych tabelach.

Tabela posiada relacje z tabelami `GameTypes`, `Categories`, `Mechanics`, `GameReleases` oraz `GameReviews`:
- relacja z tabelą `GameTypes` jest typu jeden do wielu, ponieważ jeden typ może być przypisany do wielu gier, ale jedna gra może mieć przypisaną tylko jednego typu. Relacja jest zrealizowana za pomocą klucza obcego `type_id` w tabeli `Games`.
- relacja z tabelą `GameMechanics` jest typu wiele do wielu, ponieważ jedna gra może mieć wiele mechanik, a jedna mechanika może być przypisana do wielu gier. Relacja ta jest zrealizowana za pomocą tabeli pośredniczącej `GameMechanics`.

- relacja z tabelą `Categories` jest typu wiele do wielu, ponieważ jedna gra może być przypisana do wielu kategorii, a jedna kategoria może być przypisana do wielu gier. Relacja ta jest zrealizowana za pomocą tabeli pośredniczącej `GameCategories`.
- relacja z tabelą `GameReleases` jest typu jeden do wielu, ponieważ jedna gra może mieć wiele wydań, ale jedno wydanie może dotyczyć tylko jednej gry. Relacja jest zrealizowana za pomocą klucza obcego `game_id` w tabeli `GameReleases`.
- relacja z tabelą `GameReviews` jest typu jeden do wielu, ponieważ jedna gra może mieć wiele recenzji, ale jedna recenzja dotyczy tylko jednej gry. Relacja jest zrealizowana za pomocą klucza obcego `game_id` w tabeli `GameReviews`.

Table `GameTypes`

Tabela `GameTypes` przechowuje informacje o typach gier planszowych. Zawiera jedynie pole `name`, które jest unikalne i służy do identyfikacji typu.

Tabela ta posiada relację z tabelą `Games` opisaną wcześniej.

Tabela `Categories`

Tabela `Categories` przechowuje informacje o kategoriach gier planszowych. Zawiera jedynie pole `name`, które jest unikalne i służy do identyfikacji kategorii.

Tabela ta posiada relację z tabelą `Games` opisaną wcześniej.

Tabela `Mechanics`

Tabela `Mechanics` przechowuje informacje o mechanikach gier planszowych. Zawiera pola `name` oraz `description`, z których pierwsze jest unikalne i służy do identyfikacji mechaniki.

Tabela ta posiada relację z tabelą `Games` opisaną wcześniej.

Tabela asocjacyjna `GameMechanics`

Tabela `GameMechanics` jest tabelą pośredniczącą między tabelami `Games` i `Mechanics`. Zawiera pola `game_id` oraz `mechanic_id`, które są kluczami obcymi do odpowiednich tabel. Te pola tworzą jeden klucz główny tej tabeli, który zachowuje unikalność relacji między grami a mechanikami.

Tabela `GameReleases`

Tabela `GameReleases` przechowuje informacje o wydaniach gier planszowych. Zawiera pola `game_id`, `release_date`, `publisher`, `studio`, `language`, `price` oraz `extra_information`.

Tabela ta posiada relację z tabelą `Games` opisaną wcześniej.

Tabela `GameReviews`

Tabela `GameReviews` przechowuje informacje o recenzjach gier planszowych, pisanych przez użytkowników. Zawiera pola `game_id`, `user_id`, `rating` oraz `review`.

Tabela ta posiada relację z tabelą `Games` opisaną wcześniej oraz relację z tabelą `Users`. Relacja z tabelą `Users` jest typu jeden do wielu, ponieważ jedna recenzja może być napisana przez jednego użytkownika, ale jeden użytkownik może napisać wiele recenzji. Relacja ta jest zrealizowana za pomocą klucza obcego `user_id` w tabeli `GameReviews`.

Tabela `Users`

Tabela `Users` przechowuje informacje o użytkownikach kolekcji. Zawiera pola `username`, `email`, `password`, `first_name`, `last_name` oraz `role`, która określa rolę użytkownika w systemie.

Rola może być jedną z dwóch wartości: `USER` lub `ADMIN`. Ten wybór jest zrealizowany za pomocą typu `ENUM`.

Tabela ta posiada relację z tabelą `GameReviews` opisaną wcześniej.

Zapytania SQL

Zapytania tworzące tabele

Poniżej znajdują się zapytania SQL tworzące tabele w bazie danych projektu. Zapytania te zawierają definicje tabel, kluczy głównych, kluczy obcych oraz ograniczeń.

```SQL
1   CREATE TYPE "UserRoles" AS ENUM ( 'ADMIN', 'USER' );
2
3   CREATE TABLE "Users"
4   (
5       "id"         serial PRIMARY KEY,
6       "username"   varchar(255) NOT NULL,
7       "email"      varchar(255) NOT NULL,
8       "password"   varchar(255) NOT NULL,
9       "first_name" varchar(255),
10      "last_name"  varchar(255),
11      "role"       "UserRoles"  NOT NULL
12  );
13
14  CREATE TABLE "GameTypes"
15  (
16      "id"   serial PRIMARY KEY,
17      "name" varchar(255) UNIQUE NOT NULL
18  );
19
20  CREATE TABLE "Games"
21  (
22      "id"                serial PRIMARY KEY,
23      "type_id"           integer     NOT NULL REFERENCES "GameTypes" ("id"),
24      "name"              varchar(255) NOT NULL,
25      "description"       text,
26      "min_players"       integer,
27      "max_players"       integer,
28      "playing_time"      integer,
29      "first_year_released" integer
30  );
31
32  CREATE TABLE "Categories"
```

```sql
33  (
34      "id"   serial PRIMARY KEY,
35      "name" varchar(255) UNIQUE NOT NULL
36  );
37
38  CREATE TABLE "GameCategories"
39  (
40      "game_id"     integer NOT NULL REFERENCES "Games" ("id"),
41      "category_id" integer NOT NULL REFERENCES "Categories" ("id"),
42      CONSTRAINT "pk_game_categories" PRIMARY KEY ("game_id", "category_id")
43  );
44
45  COMMENT ON COLUMN "Games"."playing_time" IS 'Approximate playing time in minutes';
46
47  CREATE TABLE "Mechanics"
48  (
49      "id"          serial PRIMARY KEY,
50      "name"        varchar(255) UNIQUE NOT NULL,
51      "description" text
52  );
53
54  CREATE TABLE "GameMechanics"
55  (
56      "game_id"     integer NOT NULL REFERENCES "Games" ("id"),
57      "mechanic_id" integer NOT NULL REFERENCES "Mechanics" ("id"),
58      CONSTRAINT "pk_game_mechanics" PRIMARY KEY ("game_id", "mechanic_id")
59  );
60
61  CREATE TABLE "GameReviews"
62  (
63      "id"      serial PRIMARY KEY,
64      "game_id" integer NOT NULL REFERENCES "Games" ("id"),
65      "user_id" integer NOT NULL REFERENCES "Users" ("id"),
66      "rating"  integer NOT NULL,
67      "review"  text
68  );
69
70  COMMENT ON COLUMN "GameReviews"."rating" IS 'Rating from 1 to 10';
71
72  CREATE TABLE "GameReleases"
73  (
74      "id"           serial PRIMARY KEY,
75      "game_id"      integer     NOT NULL REFERENCES "Games" ("id"),
76      "release_date" date        NOT NULL,
77      "publisher"    varchar(255) NOT NULL,
78      "studio"       varchar(255),
```

```sql
79      "language"              varchar(255),
80      "price"                 decimal(6, 2),
81      "extra_information" text
82 );
```

Zapytania wprowadzające dane

Poniżej znajdują się zapytania SQL wprowadzające przykładowe dane do tabel.

Tabela GameCategories

```sql
1 INSERT INTO "GameTypes" (name) VALUES
2      ('Abstract'), ('Area Control'), ('Cooperative'), ('Deck Building'),
  ('Economic'), ('Family'), ('Party'), ('Thematic'), ('War Games'), ('Strategy');
```

Tabela Games

```sql
1  INSERT INTO "Games" (type_id, name, description, min_players, max_players,
   playing_time, first_year_released) VALUES
2      ((SELECT id FROM "GameTypes" WHERE name = 'Abstract'), 'Chess', 'A classic
   two-player strategy game of capturing the opponent''s king.', 2, 2, 30, 1475),
3      ((SELECT id FROM "GameTypes" WHERE name = 'Abstract'), 'Go', 'An abstract
   strategy game for two players involving surrounding and capturing your opponent''s
   stones.', 2, 2, 60, -2200),
4      ((SELECT id FROM "GameTypes" WHERE name = 'Abstract'), 'Hive', 'An abstract
   strategy game where players compete to surround the opponent''s queen bee.', 2,
   4, 30, 2001),
5      ((SELECT id FROM "GameTypes" WHERE name = 'Area Control'), 'Risk', 'A classic
   board game of world domination through dice rolling and strategic troop placement.',
   2, 6, 120, 1957),
6      ((SELECT id FROM "GameTypes" WHERE name = 'Area Control'), 'Carcassonne',
   'A tile-laying game where players compete to claim land and build features on a
   growing map.', 2, 5, 45, 2000),
7      ((SELECT id FROM "GameTypes" WHERE name = 'Area Control'), 'Twilight Imperium',
   'A grand strategy game of galactic conquest and diplomacy for experienced players.',
   3, 8, 360, 2000),
8      ((SELECT id FROM "GameTypes" WHERE name = 'Cooperative'), 'Pandemic', 'A
   cooperative game where players work together to stop the spread of diseases around
   the world.', 2, 4, 45, 2008),
9      ((SELECT id FROM "GameTypes" WHERE name = 'Cooperative'), 'Forbidden Island',
   'A cooperative game where players work together to recover treasures from a sinking
   island.', 2, 4, 30, 2010),
10     ((SELECT id FROM "GameTypes" WHERE name = 'Cooperative'), 'Spirit Island',
   'A cooperative game where players take on the roles of powerful spirits defending
   their island from colonizers.', 1, 4, 90, 2017),
11     ((SELECT id FROM "GameTypes" WHERE name = 'Deck Building'), 'Dominion', 'The
   original deck-building game where players compete to build the most efficient deck
   of cards.', 2, 4, 30, 2008),
```

```sql
12      ((SELECT id FROM "GameTypes" WHERE name = 'Deck Building'), 'Star Realms',
        'A fast-paced deck-building game set in space where players battle for control of
        the galaxy.', 2, 6, 20, 2014),
13      ((SELECT id FROM "GameTypes" WHERE name = 'Deck Building'), 'Clank!', 'A deck-
        building adventure game where players explore a dungeon, collect treasure, and
        avoid the dragon.', 2, 4, 60, 2016),
14      ((SELECT id FROM "GameTypes" WHERE name = 'Economic'), 'Acquire', 'A classic
        economic game where players invest in and merge companies to earn the most money.',
        2, 6, 90, 1964),
15      ((SELECT id FROM "GameTypes" WHERE name = 'Economic'), 'Power Grid', 'A
        strategic economic game where players compete to power the most cities and earn
        the most money.', 2, 6, 120, 2004),
16      ((SELECT id FROM "GameTypes" WHERE name = 'Economic'), 'Food Chain Magnate',
        'A cutthroat economic game where players run competing fast food chains and try
        to outmaneuver their rivals.', 2, 5, 180, 2015),
17      ((SELECT id FROM "GameTypes" WHERE name = 'Family'), 'Ticket to Ride', 'A
        family-friendly game of building train routes across North America.', 2, 5, 45,
        2004),
18      ((SELECT id FROM "GameTypes" WHERE name = 'Family'), 'Dixit', 'A creative
        storytelling game where players interpret abstract illustrations and try to match
        each other''s descriptions.', 3, 6, 30, 2008),
19      ((SELECT id FROM "GameTypes" WHERE name = 'Family'), 'Everdell', 'A charming
        game of building a woodland critter village and attracting adorable critters to
        live there.', 1, 4, 80, 2018),
20      ((SELECT id FROM "GameTypes" WHERE name = 'Family'), 'Scrabble', 'A classic
        word game where players compete to create words and earn points based on letter
        values.', 2, 4, 90, 1938),
21      ((SELECT id FROM "GameTypes" WHERE name = 'Party'), 'Codenames', 'A word
        association game where players give one-word clues to help their teammates guess
        the correct words.', 4, 8, 15, 2015),
22      ((SELECT id FROM "GameTypes" WHERE name = 'Party'), 'Telestrations', 'A drawing
        and guessing game where players pass around sketchbooks and try to interpret each
        other''s drawings.', 4, 12, 30, 2009),
23      ((SELECT id FROM "GameTypes" WHERE name = 'Party'), 'Just One', 'A cooperative
        party game where players try to guess a word based on one-word clues, but duplicate
        clues are eliminated.', 3, 7, 20, 2018),
24      ((SELECT id FROM "GameTypes" WHERE name = 'Thematic'), 'Arkham Horror', 'A
        cooperative game of cosmic horror where players investigate mysteries and battle
        otherworldly monsters.', 1, 8, 240, 1987),
25      ((SELECT id FROM "GameTypes" WHERE name = 'Thematic'), 'Gloomhaven', 'A
        cooperative campaign game of tactical combat and exploration in a dark fantasy
        world.', 1, 4, 120, 2017),
26      ((SELECT id FROM "GameTypes" WHERE name = 'Thematic'), 'Malhya: Lands of
        Legends', 'A narrative-driven adventure game set in a rich fantasy world with
        branching storylines and epic quests.', 1, 5, 180, 2024),
```

```sql
27    ((SELECT id FROM "GameTypes" WHERE name = 'War Games'), 'Axis & Allies', 'A
      classic World War II strategy game of global conflict and military strategy.', 2,
      5, 180, 1981),
28    ((SELECT id FROM "GameTypes" WHERE name = 'War Games'), 'Twilight Struggle',
      'A two-player game of Cold War politics and strategy where players compete for
      global influence.', 2, 2, 180, 2005),
29    ((SELECT id FROM "GameTypes" WHERE name = 'War Games'), 'Root', 'A strategic
      war game of woodland creatures vying for control of the forest and its resources.',
      2, 4, 90, 2018),
30    ((SELECT id FROM "GameTypes" WHERE name = 'Strategy'), 'Terraforming Mars', 'A
      strategic game of colonizing and terraforming Mars to make it habitable for human
      life.', 1, 5, 120, 2016),
31    ((SELECT id FROM "GameTypes" WHERE name = 'Strategy'), 'Scythe', 'A strategic
      game of farming, combat, and resource management set in an alternate history 1920s
      Europe.', 1, 7, 115, 2016),
32    ((SELECT id FROM "GameTypes" WHERE name = 'Strategy'), 'The Castles of Burgundy:
      Special Edition', 'A strategic game of building and developing a medieval estate
      through tile placement and resource management.', 1, 4, 90, 2023),
33    ((SELECT id FROM "GameTypes" WHERE name = 'Strategy'), 'Catan', 'A classic
      board game of trading and building settlements on the island of Catan.', 3, 4,
      90, 1995);
```

Tabele `Categories` i `GameCategories`

```sql
1    INSERT INTO "Categories" (name) VALUES
2        ('Word Games'), ('Spies/Secret Agents'), ('Humor'), ('Fantasy'), ('Science
     Fiction'), ('Adventure'), ('Novel-based'), ('Horror'), ('Territory Building'),
     ('Medieval');
3
4    INSERT INTO "GameCategories" (game_id, category_id) VALUES
5        ((SELECT id FROM "Games" WHERE name = 'Codenames'), (SELECT id FROM "Categories"
     WHERE name = 'Word Games')),
6        ((SELECT id FROM "Games" WHERE name = 'Codenames'), (SELECT id FROM "Categories"
     WHERE name = 'Spies/Secret Agents')),
7        ((SELECT id FROM "Games" WHERE name = 'Telestrations'), (SELECT id FROM
     "Categories" WHERE name = 'Humor')),
8        ((SELECT id FROM "Games" WHERE name = 'Gloomhaven'), (SELECT id FROM "Categories"
     WHERE name = 'Fantasy')),
9        ((SELECT id FROM "Games" WHERE name = 'Gloomhaven'), (SELECT id FROM "Categories"
     WHERE name = 'Adventure')),
10       ((SELECT id FROM "Games" WHERE name = 'Spirit Island'), (SELECT id FROM
     "Categories" WHERE name = 'Fantasy')),
11       ((SELECT id FROM "Games" WHERE name = 'Clank!'), (SELECT id FROM "Categories"
     WHERE name = 'Fantasy')),
12       ((SELECT id FROM "Games" WHERE name = 'Arkham Horror'), (SELECT id FROM
     "Categories" WHERE name = 'Horror')),
```

13	`((SELECT id FROM "Games" WHERE name = 'Arkham Horror'), (SELECT id FROM "Categories" WHERE name = 'Adventure')),`
14	`((SELECT id FROM "Games" WHERE name = 'Arkham Horror'), (SELECT id FROM "Categories" WHERE name = 'Novel-based')),`
15	`((SELECT id FROM "Games" WHERE name = 'Scythe'), (SELECT id FROM "Categories" WHERE name = 'Science Fiction')),`
16	`((SELECT id FROM "Games" WHERE name = 'The Castles of Burgundy: Special Edition'), (SELECT id FROM "Categories" WHERE name = 'Medieval')),`
17	`((SELECT id FROM "Games" WHERE name = 'The Castles of Burgundy: Special Edition'), (SELECT id FROM "Categories" WHERE name = 'Territory Building'));`

Tabele `Mechanics` i `GameMechanics`

```SQL
1  INSERT INTO "Mechanics" (name) VALUES
2      ('Tile Placement'), ('Hand Management'), ('Dice Rolling'), ('Team-Based Game'), ('Trading'), ('Memory'), ('Auction/Bidding'), ('Map Addition'), ('Player Elimination'), ('Deck Building'), ('Income'), ('End Game Bonuses');
3
4  INSERT INTO "GameMechanics" (game_id, mechanic_id) VALUES
5      ((SELECT id FROM "Games" WHERE name = 'Carcassonne'), (SELECT id FROM "Mechanics" WHERE name = 'Tile Placement')),
6      ((SELECT id FROM "Games" WHERE name = 'Carcassonne'), (SELECT id FROM "Mechanics" WHERE name = 'Map Addition')),
7      ((SELECT id FROM "Games" WHERE name = 'Dominion'), (SELECT id FROM "Mechanics" WHERE name = 'Hand Management')),
8      ((SELECT id FROM "Games" WHERE name = 'Dominion'), (SELECT id FROM "Mechanics" WHERE name = 'Deck Building')),
9      ((SELECT id FROM "Games" WHERE name = 'Power Grid'), (SELECT id FROM "Mechanics" WHERE name = 'Auction/Bidding')),
10     ((SELECT id FROM "Games" WHERE name = 'Root'), (SELECT id FROM "Mechanics" WHERE name = 'Hand Management')),
11     ((SELECT id FROM "Games" WHERE name = 'Root'), (SELECT id FROM "Mechanics" WHERE name = 'Dice Rolling')),
12     ((SELECT id FROM "Games" WHERE name = 'Everdell'), (SELECT id FROM "Mechanics" WHERE name = 'Hand Management')),
13     ((SELECT id FROM "Games" WHERE name = 'Everdell'), (SELECT id FROM "Mechanics" WHERE name = 'Income')),
14     ((SELECT id FROM "Games" WHERE name = 'Everdell'), (SELECT id FROM "Mechanics" WHERE name = 'End Game Bonuses')),
15     ((SELECT id FROM "Games" WHERE name = 'Scrabble'), (SELECT id FROM "Mechanics" WHERE name = 'Tile Placement')),
16     ((SELECT id FROM "Games" WHERE name = 'Scrabble'), (SELECT id FROM "Mechanics" WHERE name = 'Hand Management')),
17     ((SELECT id FROM "Games" WHERE name = 'Scrabble'), (SELECT id FROM "Mechanics" WHERE name = 'End Game Bonuses'));
```

Tabele `GameReviews` i `Users`

```sql
INSERT INTO "Users" (username, email, password, role) VALUES
                            ('admin',              '174725@stud.prz.edu.pl',
'$2y$10$xjw.5TWX7uzvlWEsyaf6BuRhIXv2M2zWC9NcsIyGVe50DefmQYOEq', 'ADMIN'),
                            ('testuser',           'kpomykala2002@gmail.com',
'$2y$10$H.5U1jmjzdLY44BecfDp5Ojc2UVI74FaCqaHFMcFix8fez3dhPfsC', 'USER');

INSERT INTO "GameReviews" (game_id, user_id, rating, review) VALUES
    ((SELECT id FROM "Games" WHERE name = 'Carcassonne'), (SELECT id FROM "Users"
WHERE username = 'testuser'), 4, 'Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Etiam id consequat lacus. Cras ultricies, nunc molestie placerat tincidunt,
enim sapien imperdiet nulla, sit amet tincidunt felis lorem pretium erat.'),
    ((SELECT id FROM "Games" WHERE name = 'Dominion'), (SELECT id FROM "Users" WHERE
username = 'testuser'), 5, 'Integer eget ligula nec nisi accumsan dignissim nec
eget libero. Class aptent taciti sociosqu ad litora torquent per conubia nostra,
per inceptos himenaeos. Ut suscipit volutpat odio in tempus. Sed gravida a lectus
sed vulputate. Maecenas eget diam a ante consequat consequat et sit amet diam. '),
    ((SELECT id FROM "Games" WHERE name = 'Power Grid'), (SELECT id FROM "Users"
WHERE username = 'admin'), 2, 'Morbi porta libero ut nunc fermentum ullamcorper.
Nunc nulla ex, iaculis ut sapien vitae, mollis accumsan ex. Etiam varius nisi ut
consectetur placerat. Aliquam aliquam vel purus quis ultricies. Phasellus eget
metus sit amet erat mollis dapibus nec ac mi. Quisque porta tincidunt sapien eu
maximus. Nullam ut dictum ipsum. '),
    ((SELECT id FROM "Games" WHERE name = 'Root'), (SELECT id FROM "Users" WHERE
username = 'admin'), 4, null),
    ((SELECT id FROM "Games" WHERE name = 'Root'), (SELECT id FROM "Users" WHERE
username = 'testuser'), 5, null),
    ((SELECT id FROM "Games" WHERE name = 'Everdell'), (SELECT id FROM "Users"
WHERE username = 'testuser'), 3, null),
    ((SELECT id FROM "Games" WHERE name = 'Scrabble'), (SELECT id FROM "Users"
WHERE username = 'admin'), 1, null);
```

Tabela GameReleases

```sql
INSERT  INTO  "GameReleases" (game_id, release_date, publisher, studio,
language, price, extra_information) VALUES
    ((SELECT id FROM "Games" WHERE name = 'Carcassonne'), '2020-01-01', 'Bard
Centrum Gier', 'Bard Centrum Gier', 'Polish', 101.00, null),
    ((SELECT id FROM "Games" WHERE name = 'Catan'), '2023-06-01', 'NeoTroy Games',
null, 'Turkish', 150.00, 'Limited edition'),
    ((SELECT id FROM "Games" WHERE name = 'Catan'), '2021-09-10', 'Albi, KOSMOS',
'KOSMOS', 'Czech', 98.21, 'Alternative name: "Catan: Základní hra". Used'),
    ((SELECT id FROM "Games" WHERE name = 'Catan'), '2022-06-30', 'KOSMOS', 'KOSMOS',
'German', 129.99, 'Alternative name: "Catan: Das Spiel"'),
    ((SELECT id FROM "Games" WHERE name = 'Catan'), '2021-01-12', 'Swan Panasia
Co., Ltd.', 'KOSMOS', 'Chinese', 150.00, 'Alternative name: "卡坦島". Imported'),
    ((SELECT id FROM "Games" WHERE name = 'Dominion'), '2013-12-12', 'Ystari Games',
null, 'French', 80.00, 'Third printing');
```

Przykładowe zapytania selekcyjne

Zapytanie o gry z kategorii „Fantasy"

```sql
1  SELECT "Games"."name" FROM "Games"
2  INNER JOIN "GameCategories" ON "Games"."id" = "GameCategories"."game_id"
3  INNER JOIN "Categories" ON "GameCategories"."category_id" = "Categories"."id"
4  WHERE "Categories"."name" = 'Fantasy';
```

Zapytanie o gry z mechaniką „Hand Management"

```sql
1  SELECT "Games"."name" FROM "Games"
2  INNER JOIN "GameMechanics" ON "Games"."id" = "GameMechanics"."game_id"
3  INNER JOIN "Mechanics" ON "GameMechanics"."mechanic_id" = "Mechanics"."id"
4  WHERE "Mechanics"."name" = 'Hand Management';
```

Zapytanie o gry, które posiadają średnią recenzji powyżej 4

```sql
1  SELECT "Games"."name", AVG("GameReviews"."rating") AS "average_rating",
   COUNT("GameReviews"."rating") AS "number_of_ratings" FROM "Games"
2  RIGHT JOIN "GameReviews" ON "Games"."id" = "GameReviews"."game_id"
3  GROUP BY "Games"."id"
4  HAVING AVG("GameReviews"."rating") >= 4;
```

Zapytanie o kategorie, które posiadają więcej niż 10, ale mniej niż 20 gier

```sql
1  SELECT "Categories"."name", COUNT("Games"."id") AS "game_count" FROM
   "Categories"
2  INNER JOIN "GameCategories" ON "Categories"."id" = "GameCategories"."category_id"
3  INNER JOIN "Games" ON "GameCategories"."game_id" = "Games"."id"
4  GROUP BY "Categories"."id"
5  HAVING COUNT("Games"."id") > 10 AND COUNT("Games"."id") < 20;
```

Zapytanie o użytkowników z rolą „USER" i ilością napisanych recenzji

```sql
1  SELECT "Users"."username", COUNT("GameReviews"."id") AS "review_count" FROM
   "Users"
2  LEFT JOIN "GameReviews" ON "Users"."id" = "GameReviews"."user_id"
3  WHERE "Users"."role" = 'USER'
4  GROUP BY "Users"."id";
```

Zapytanie o gry wydane po 2010 roku

```sql
1  SELECT "Games"."name", "Games"."first_year_released" FROM "Games"
2  WHERE "Games"."first_year_released" > 2010;
```

Zapytanie o gry, których zsumowana cena wydań przekracza 250 zł

```SQL
SELECT    "Games"."name",    SUM("GameReleases"."price")    AS    "price_sum",
COUNT("GameReleases"."price") AS "number_of_releases" FROM "Games"
INNER JOIN "GameReleases" ON "Games"."id" = "GameReleases"."game_id"
GROUP BY "Games"."id"
HAVING SUM("GameReleases"."price") > 250;
```