



POLITECHNIKA RZESZOWSKA

im. Ignacego Łukasiewicza

WYDZIAŁ MATEMATYKI I FIZYKI STOSOWANEJ

Szymon Czeakański, Krzysztof Kopiec, Andrii Kotovych,

172932, 173158, 173163

Inżynieria i Analiza Danych Rok 2

Baza danych Spotify

BAZY DANYCH

Rzeszów 2024

Spis treści

1.	Projekt konceptualny	3
1.	Formułowanie zadania projektowego	3
2.	Analiza stanu wyjściowego	3
3.	Analiza wstępnych wymagań użytkownika:	3
4.	Określenie scenariuszy użycia	4
5.	Diagram UML	5
6.	Identyfikacja funkcji	6
7.	Diagram ERD	7
2.	Projekt logiczny	8
1.	Przejdźcie z modelu ERD na model relacyjny	8
2.	Diagram w notacji Crow's Foot	9
3.	Normalizacja	10
4.	Diagram po normalizacji	12
5.	Proponowane operacje na danych	14
3.	Projekt implementacyjny	15
1.	Implementacja kwerend i ich wyników	15
2.	Skrypty bazy danych	26
4.	Określenie kierunków rozwoju aplikacji	42
5.	Opracowanie doświadczeń wynikających z realizacji projektu	42
6.	Wykaz literatury	42

1. Projekt konceptualny

1. Formułowanie zadania projektowego

Projekt ma na celu zaprojektowanie i implementację bazy danych dla platformy muzycznej wzorowanej na Spotify. Głównym celem jest stworzenie spójnej struktury przechowującej informacje o użytkownikach, utworach muzycznych, artystach, listach odtwarzania.

2. Analiza stanu wyjściowego

Spotify jako platforma muzyczna: Analiza funkcjonalności i interfejsu użytkownika platformy Spotify. Zrozumienie sposobu, w jaki użytkownicy korzystają z aplikacji, zarządzają playlistami, przeglądają utwory oraz tworzą spersonalizowane listy.

Struktura danych w Spotify: Zbieranie informacji o strukturze danych przechowywanych przez Spotify, takich jak informacje o użytkownikach, artystach, albumach, utworach, listach odtwarzania itp.

3. Analiza wstępnych wymagań użytkownika:

Rejestracja i logowanie: System rejestracji użytkowników, zarządzanie kontem, logowanie, zapomniane hasło.

Zarządzanie playlistami: Tworzenie, edycja i usuwanie list odtwarzania.

Zarządzanie utworami i artystami: Możliwość wyszukiwania, filtrowania i przeglądania utworów oraz artystów, pobieranie informacji o albumach i twórcach.

Projekt będzie opierał się na analizie tych aspektów, aby zidentyfikować kluczowe wymagania użytkownika oraz stworzyć efektywną strukturę bazy danych dla platformy muzycznej podobnej do Spotify.

4. Określenie scenariuszy użycia

1. Rejestracja nowego użytkownika:

Użytkownik otwiera aplikację i wybiera opcję rejestracji.

Wypełnia formularz rejestracyjny, podając swoje dane osobowe.

System weryfikuje unikalność informacji i tworzy nowe konto użytkownika.

2. Logowanie do konta:

Użytkownik wprowadza dane logowania (nazwa użytkownika/e-mail i hasło).

System weryfikuje poprawność danych i autoryzuje użytkownika do dostępu.

3. Zarządzanie listami odtwarzania:

Użytkownik ma możliwość tworzenia/usuwania nowych playlist, dodawania/usuwania utworów.

Może przeglądać playlisty innych użytkowników.

4. Przeglądanie i odtwarzanie muzyki:

Użytkownik może przeglądać utwory, artystów, albumy.

Może odtwarzać utwory i je polubić.

5. Zarządzanie profilem użytkownika:

Użytkownik może edytować swoje dane osobowe, zmieniać hasło.

6. Wybór subskrypcji:

Użytkownik loguje się na konto i przechodzi do sekcji subskrypcji.

Wybiera odpowiedni plan subskrypcji (np. premium) oraz metodę płatności.

7. Proces płatności:

Użytkownik wprowadza dane płatności (dane karty kredytowej, bon podarunkowy, BLIK itp.) i potwierdza zakup subskrypcji.

System przetwarza płatność i aktualizuje status konta użytkownika na "premium".

8. Korzyści z subskrypcji premium:

Użytkownik zyskuje dostęp do dodatkowych funkcji, takich jak brak reklam, możliwość pobierania utworów offline, pomijanie utworów bez limitu, dźwięk wysokiej jakości, odtwarzanie dowolnego utworu.

Może korzystać z funkcji premium przez miesiąc.

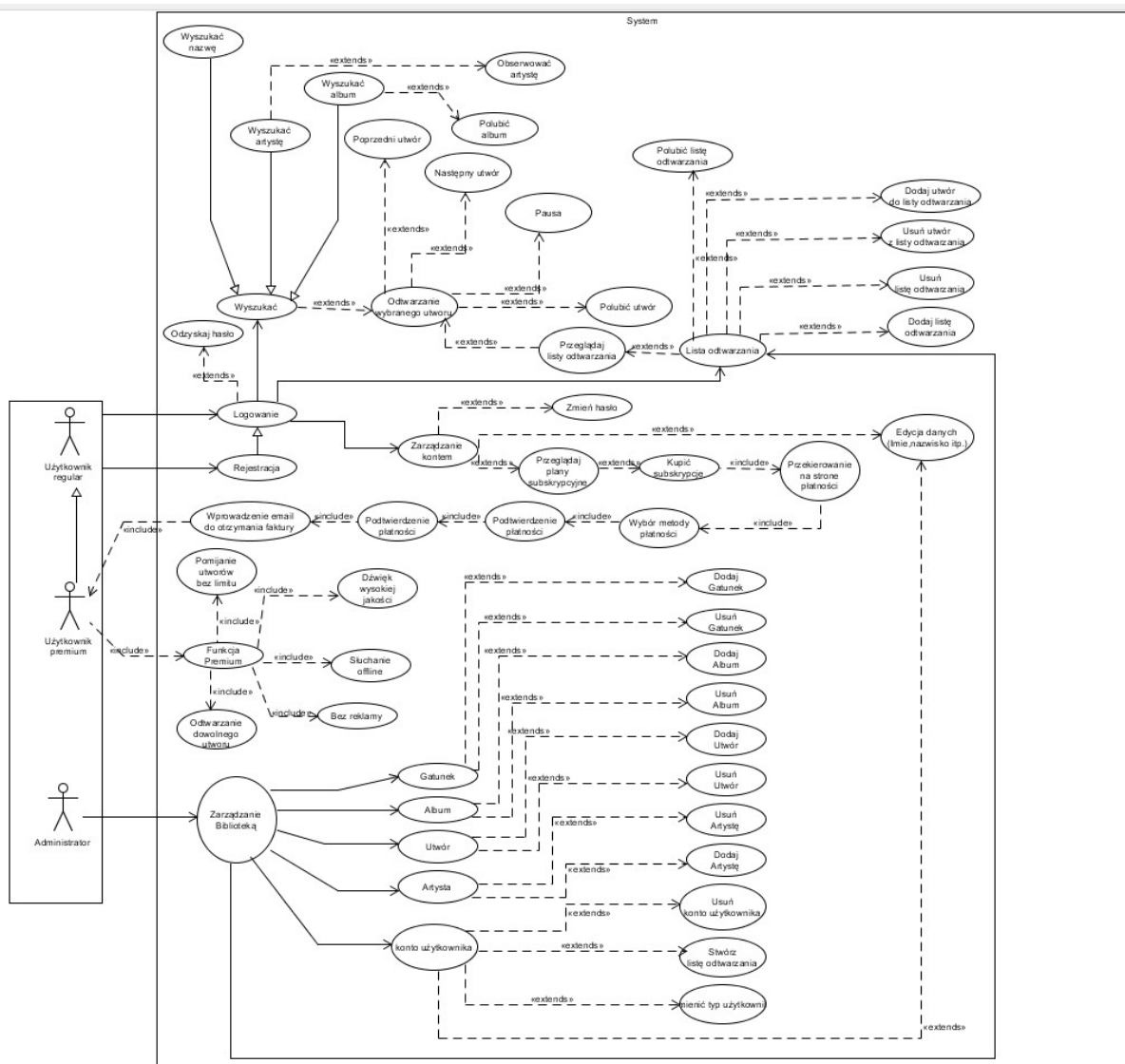
9. Zarządzanie użytkownikami:

Administrator ma możliwość przeglądania listy użytkowników, edycji ich danych osobowych, usuwania kont użytkowników

10. Zarządzanie zawartością:

Administrator może dodawać/usuwać utwory, albumy, informacje o artystach.

5. Diagram UML



6. Identyfikacja funkcji

1. Użytkownik regular:

Rejestracja – rejestracja nowego użytkownika.

Logowanie – logowanie do konta użytkownika.

Zarządzanie kontem – zarządzanie danymi osobistymi użytkownika(imie nazwisko hasło itp.)i subskrypcją.

Wyszukać – wyszukać utwór , album, artystę.

Listy odtwarzania – zarządzanie listami odtwarzania (tworzenie/usuwanie list, polubić listę dodać/usunąć utwory z listy) .

Odtwarzanie wybranego utworu – odtwarzanie przewijanie zatrzymanie utworu.

2. Użytkownik premium

Powtarza funkcje użytkownika regular

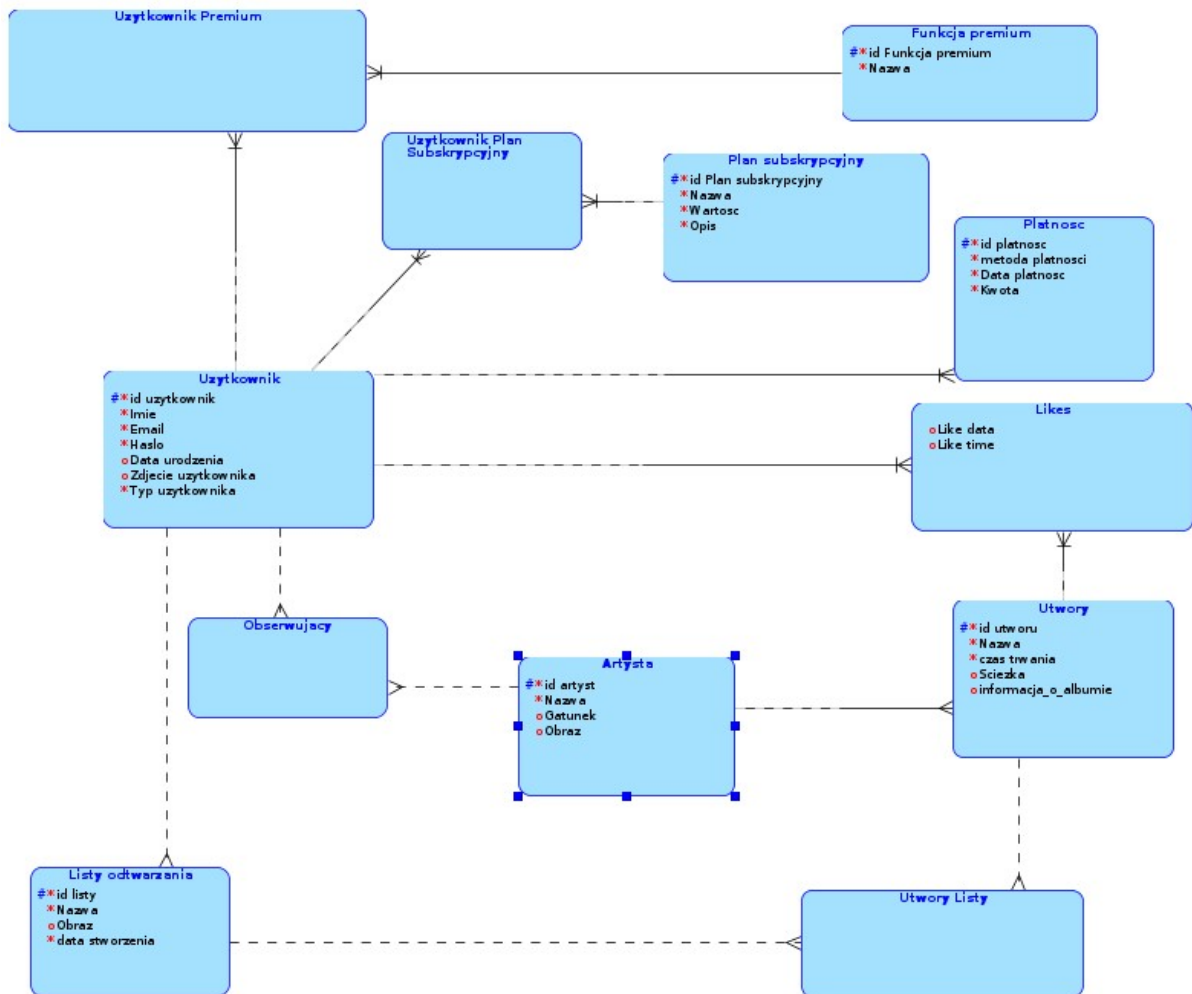
Funkcja premium – Użytkownik zyskuje dostęp do dodatkowych funkcji, takich jak brak reklam, możliwość pobierania utworów offline, pomijanie utworów bez limitu, Dźwięk wysokiej jakości, odtwarzanie dowolnego utworu.

3. Administrator

Zarządzanie biblioteką – zarządzanie informacją i danymi o gatunkach, artystach, utworach, listach odtwarzania i użytkownikach.

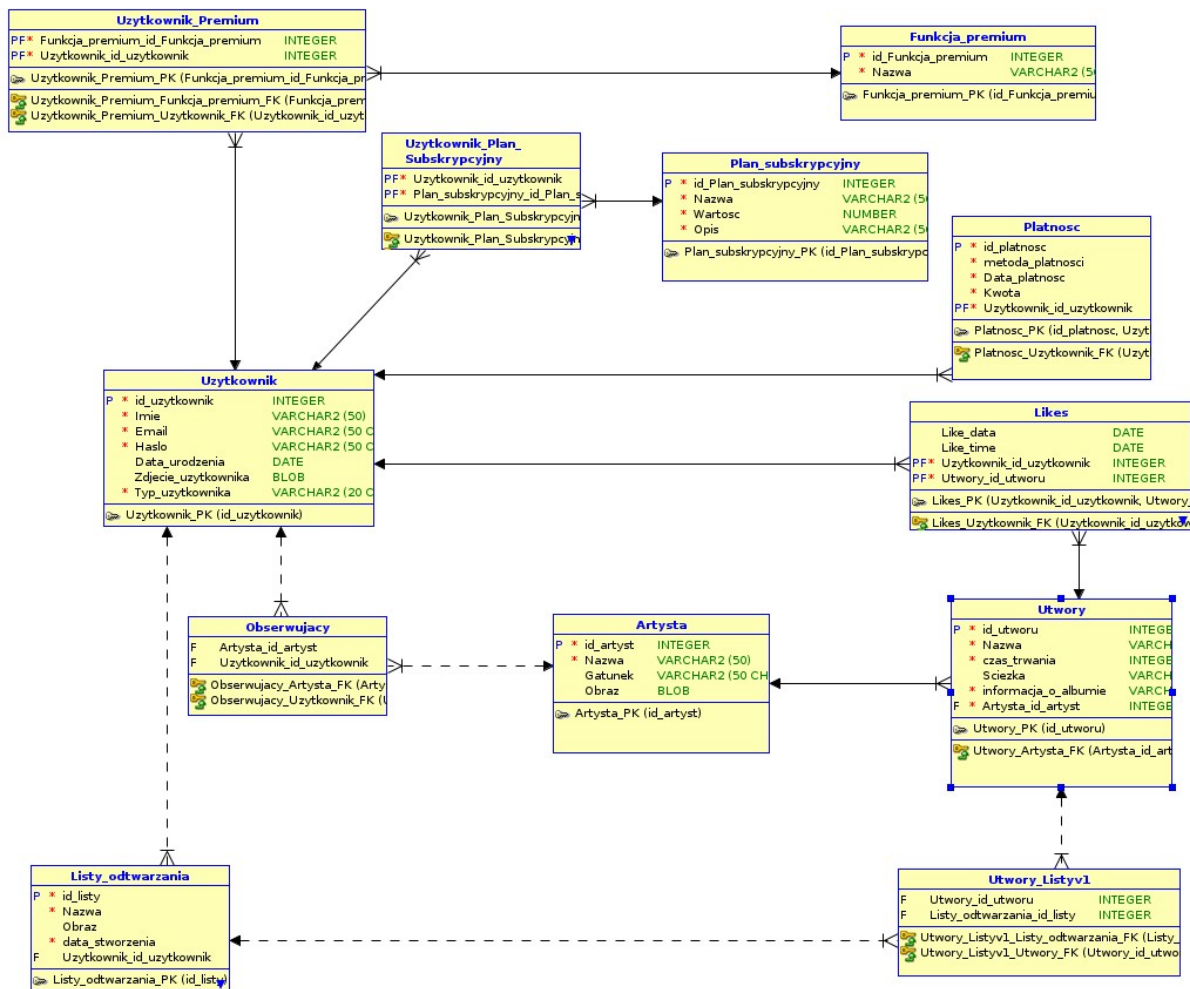
7. Diagram ERD

Propozycja encji i ich atrybutów, powiązań przedstawiona za pomocą diagramu ERD

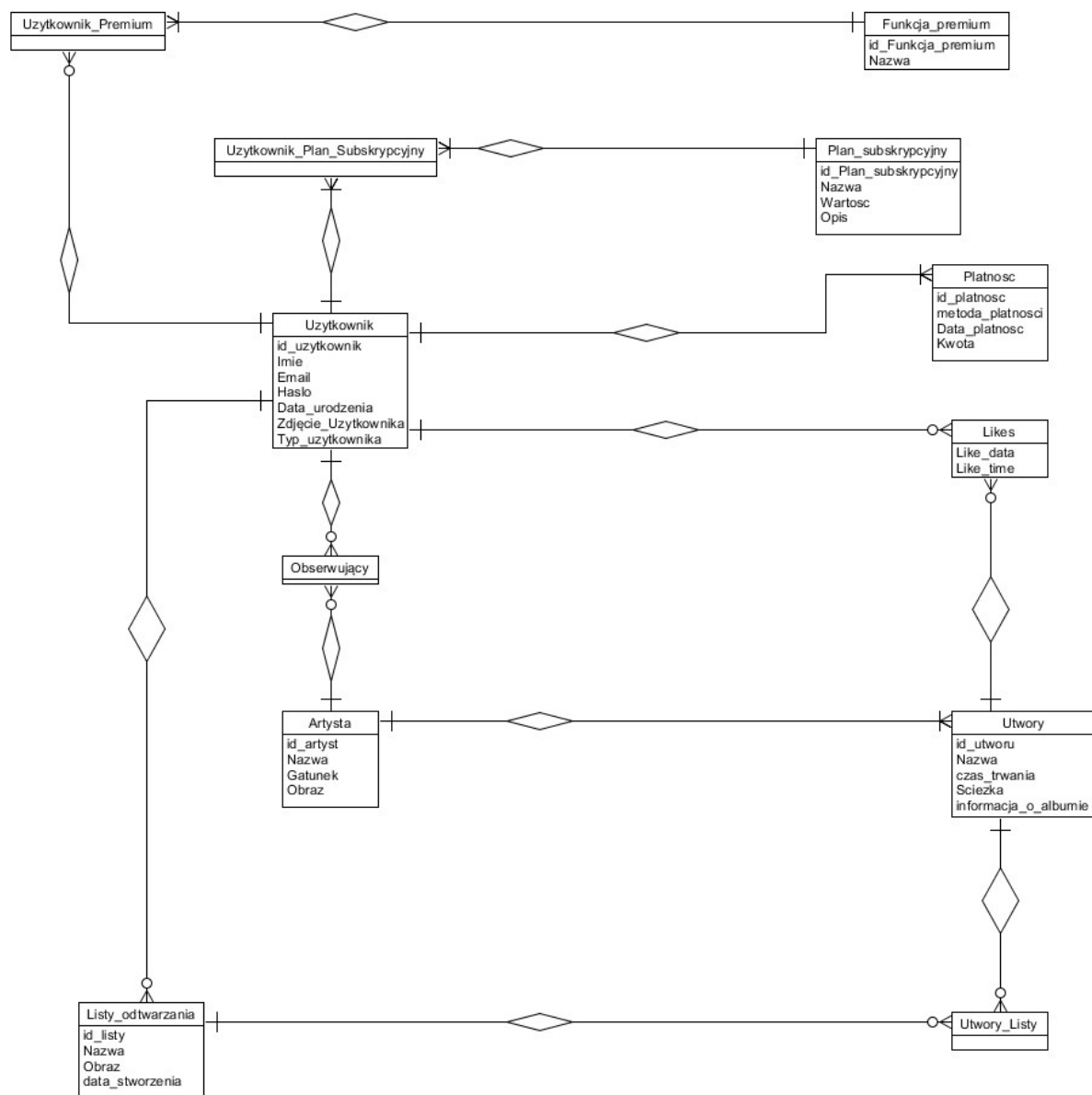


2. Projekt logiczny

1. Przejście z modelu ERD na model relacyjny



2. Digram w notacji Crow's Foot



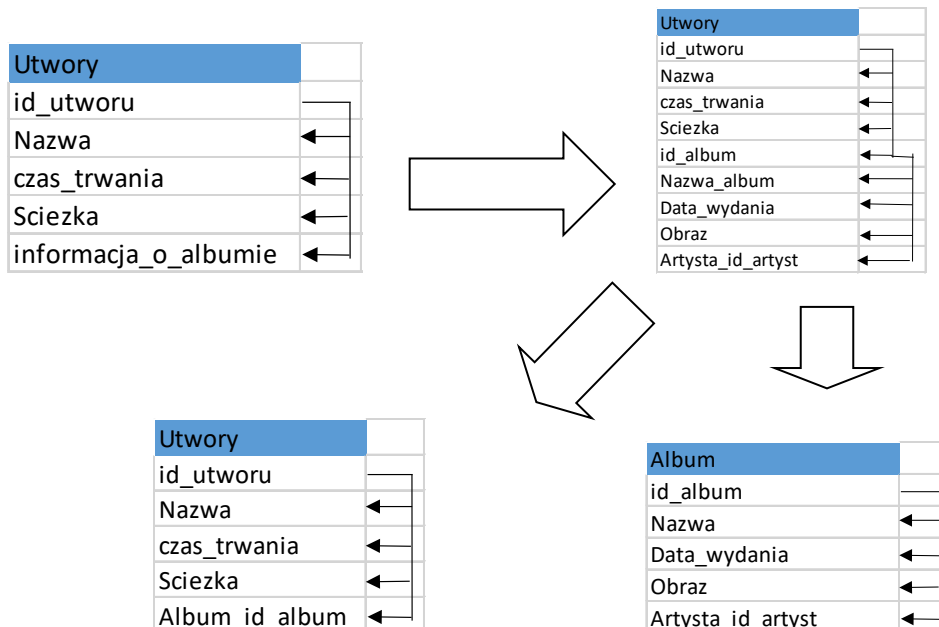
3. Normalizacja

Uzytkownik	
id_uzytkownika	
Imie	←
Email	←
Haslo	←
Data_urodzenia	←
Zdjecie_uzytkownika	←
Typ_uzytkownika	←

Spełnia warunki 1NF 2NF 3NF

Artysta	
id_artysta	
Nazwa	←
Gatunek	←
Obraz	←

Spełnia warunki 1NF 2NF 3NF



1NF	Tabela „Utwory” nie spełnia warunki 1NF, kolumna informacja_o_albumie zawiera wartości nie atomowe (nazwa,data wydania,obraz), rozbijamy kolumnę na kilka
2NF	Tabela „Utwory’ nie spełnia warunki 2NF, kolumny nazwa_album, data_wydania, obraz, artysta_id_artyst nie są zależne od klucza głównego id_utworu, tworzymy relację o nazwie „Album” z kolumnami:id_album(PK), nazwa_album, data_wydania, obraz, artysta_id_artyst(FK), w tabeli „Utwory” zostawiamy klucz obcy Album_id_album
3NF	Spełnia warunki 3NF

Listy_odtworzenia	
id_listy	
Nazwa	←
Obraz	←
data_stworzenia	←
Uzytkownik_id_uzytkownik	←

Spełnia warunki 1NF 2NF 3NF

Plan_subskrypcyjny	
id_Plan_subskrypcyjny	
Nazwa	←
Wartosc	←
Opis	←

Spełnia warunki 1NF 2NF 3NF

Funkcja_premium	
id_Funkcja_premium	
Nazwa	←

Spełnia warunki 1NF 2NF 3NF

Platnosc	
id_platnosc	
metoda_platnosci	←
data_platnosci	←
Kwota	←
Uzytkownik_id_uzytkownika	←

Spełnia warunki 1NF 2NF 3NF

Utwory_Listy	
listy_odtworzenia_id_listy	
Utwory_id_utworu	

Spełnia warunki 1NF 2NF 3NF

Obserwujacy	
Uzytkownik_id_uzytkownik	
Artysta_id_artyst	

Spełnia warunki 1NF 2NF 3NF

Likes	
Like_data	←
Like_time	←
Uzytkownik_id_uzytkownik	
Utwory_id_utworu	

Spełnia warunki 1NF 2NF 3NF

Uzytkownik_Premium	
Uzytkownik_id_uzytownik	
Funkcja_premium_id_Funkcja_premium	

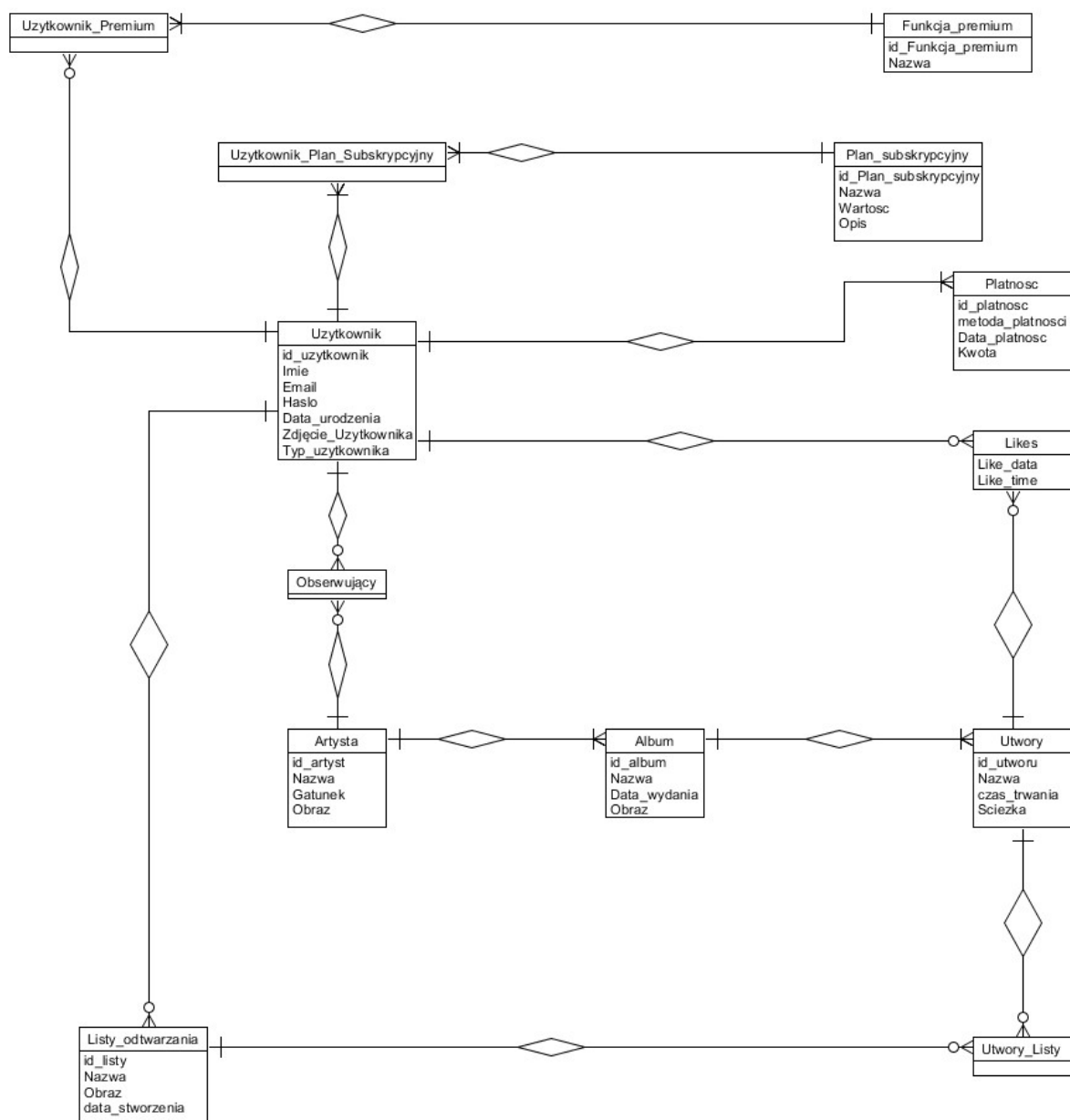
Spełnia warunki 1NF 2NF 3NF

Uzytkownik_Plan_Subskrypcyjny	
Plan_subskrypcyjny_id_Plan_subskrypcyjny	
Uzytkownik_id_uzytownik	

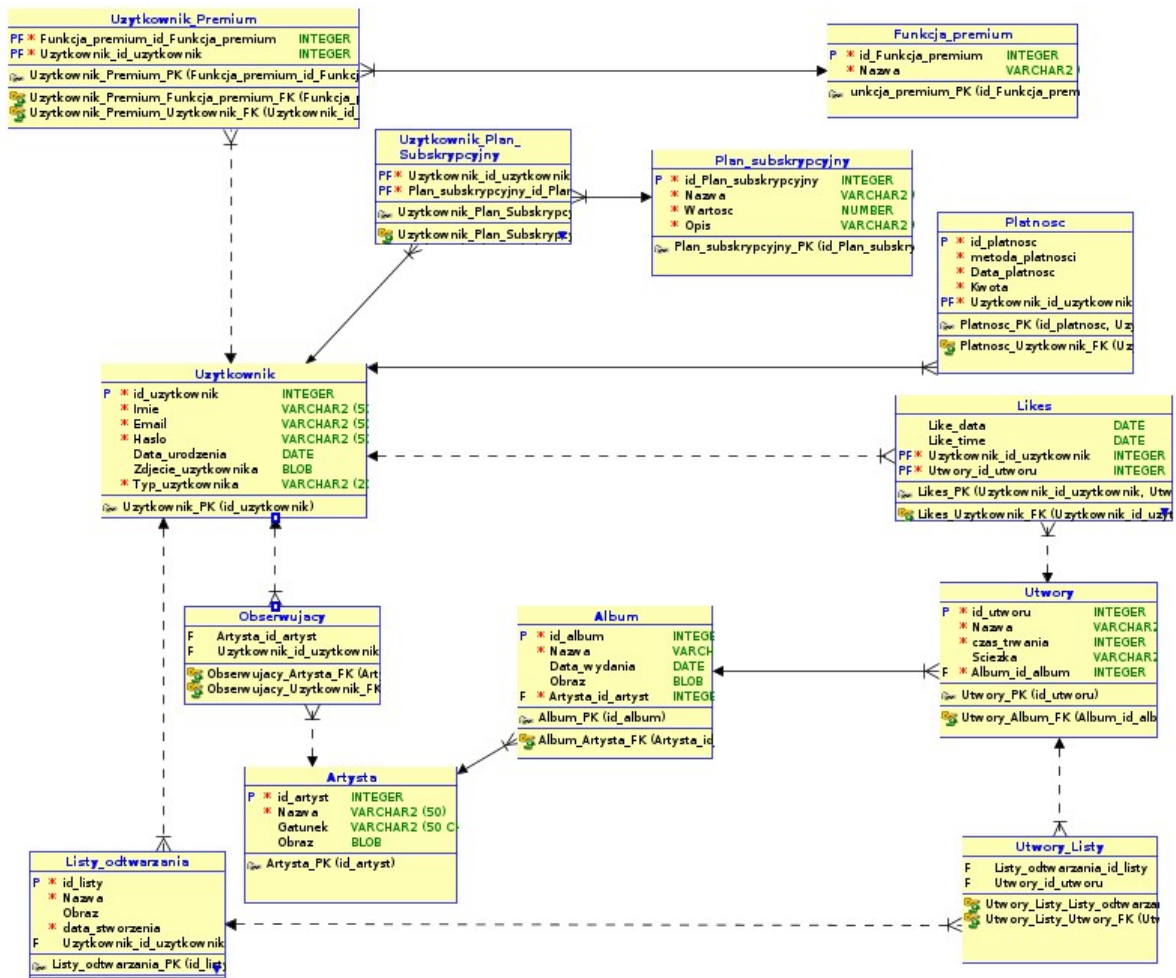
Spełnia warunki 1NF 2NF 3NF

4. Diagram po normalizacji

1. Crow's Foot



2. Diagram klas



5. Propozycje operacji na danych

1. Pobierz nazwę artysty i nazwy wszystkich utworów w albumie
2. Pobierz nazwę użytkownika, nazwę listy odtwarzania i wszystkie utwory na liście odtwarzania
3. Pobierz nazwę użytkownika i wszystkich artystów, których obserwuje
4. Pobierz nazwę użytkownika i wszystkie utwory, które polubił
5. Pobierz nazwę artysty i nazwę wszystkich utworów w albumie, ale tylko w przypadku albumów wydanych od 2022 roku
6. Pobierz nazwę artysty, albumu i czas trwania (suma) utworów z tego albumu
7. Pobierz nazwę użytkownika, nazwę listy odtwarzania i wszystkie utwory na liście odtwarzania, ale tylko w przypadku list odtwarzania utworzonych w ciągu ostatniego roku
8. Znajdź użytkowników, którzy obserwują najwięcej artystów
9. Wybierz wszystkie informacje o użytkownikach, którzy posiadają zdjęcie profilowe
10. Znajdź nazwy wszystkich albumów wydanych w roku 2023
11. Znajdź nazwę artysty i nazwę wszystkich utworów, które mają czas trwania powyżej 5 minut
12. Znajdź informacje o użytkownikach, którzy są typem "premium"
13. Znajdź nazwę użytkownika i nazwę listy odtwarzania dla użytkowników, dokonali płatności bonem lub blikiem
14. Znajdź wszystkie informacje o użytkownikach którzy mają subskrypcję student
15. Znajdź imię użytkownika i nazwy polubionych utworów dla użytkowników kto dokonał płatności od września 2022 do stycznia 2023 włącznie

3. Projekt implementacyjny

1. Implementacja kwerend i ich wyników

1. Pobierz nazwę artysty i nazwy wszystkich utworów w albumie

```
SELECT a.nazwa AS nazwa_artysty, u.nazwa AS nazwa_utworu
FROM album al
JOIN artysta a ON al.artysta_id_artyst = a.id_artyst
JOIN utwory u ON al.id_album = u.album_id_album
```

	NAZWA_ARTYSTY	NAZWA_UTWORU
45	Szpaku	Nieważne
46	Szpaku	Oddajemy krew Wampirom
47	Szpaku	Ja YETI
48	Szpaku	MOJO JOJO
49	ZYZZ	One Last Time
50	ZYZZ	Say it right
51	BAKI	Safe and Sound
52	BAKI	Runaway
53	Juice WRLD	Intro
54	Juice WRLD	All Girls Are the Same

- Pobierz nazwę użytkownika, nazwę listy odtwarzania i wszystkie utwory na liście odtwarzania

Worksheet

Query Builder

```

SELECT u.imie AS nazwa_uzytkownika, l.nazwa AS nazwa_listy, ut.nazwa AS nazwa_utworu
FROM uzytkownik u
JOIN listy_odtwarzania l ON u.id_uzytkownik = l.uzytkownik_id_uzytkownik
JOIN utwory_listy ul ON l.id_listy = ul.listy_odtwarzania_id_listy
JOIN utwory ut ON ul.utwory_id_utworu = ut.id_utworu

```

Query Result x

SQL | Fetched 50 rows in 0.277 seconds

	NAZWA_UZYTKOWNIKA	NAZWA_LISTY	NAZWA_UTWORU
10	John Doe	Ulubione utwory	NA ZAWSZE
11	Jane Smith	Playlista na weekend	Betrayal
12	Jane Smith	Playlista na weekend	Ty znajesh
13	Jane Smith	Playlista na weekend	Die Alone
14	Jane Smith	Playlista na weekend	odkąd nauczyłem się przeklinać
15	Jane Smith	Playlista na weekend	Intro
16	Jane Smith	Playlista na weekend	SCOOBY DOO
17	Jane Smith	Playlista na weekend	Blue Balls
18	Jane Smith	Playlista na weekend	xo
19	Jane Smith	Playlista na weekend	Galway Girl
20	Jane Smith	Playlista na weekend	All Girls Are the Same
21	Jane Smith	Playlista na weekend	Hoe! Hoe! Hoe!
22	Alice Johnson	Melodie relaksacyjne	I`m Still
23	Alice Johnson	Melodie relaksacyjne	GOD MODE
24	Alice Johnson	Melodie relaksacyjne	Not So Weak
25	Alice Johnson	Melodie relaksacyjne	TOTORO
26	Alice Johnson	Melodie relaksacyjne	How Would You Feel

3. Pobierz nazwę użytkownika i wszystkich artystów, których obserwuje

```
SELECT u.imie AS nazwa_uzytkownika, a.nazwa AS nazwa_artysty  
FROM uzytkownik u  
JOIN obserwujacy o ON u.id_uzytkownik = o.uzytkownik_id_uzytkownik  
JOIN artysta a ON o.artysta_id_artyst = a.id_artyst;
```

Query Result x	
SQL Fetched 50 rows in 0.1 seconds	
NAZWA_UZYTEKOWNIKA	NAZWA_ARTYSTY
16 Bob Brown	Ed Sheeran
17 Bob Brown	Golubenko
18 Bob Brown	Sxmptra
19 Bob Brown	ovg!
20 Eva Williams	Chivas
21 Eva Williams	Szpaku
22 Michael Davis	ZYZZ
23 Michael Davis	Szpaku
24 Michael Davis	Ed Sheeran
25 Michael Davis	SiM
26 Michael Davis	Juice WRLD
27 Michael Davis	BAKI
28 Sophia Wilson	Chivas
29 Sophia Wilson	Juice WRLD
30 Sophia Wilson	Sxmptra
31 Sophia Wilson	ovg!
32 Oliver Garcia	Juice WRLD

4. Pobierz nazwę użytkownika i wszystkie utwory, które polubił

```
SELECT u.imie AS nazwa_uzytkownika, ut.nazwa AS nazwa_utworu
FROM uzytkownik u
JOIN likes l ON u.id_uzytkownik = l.uzytkownik_id_uzytkownik
JOIN utwory ut ON l.utwory_id_utworu = ut.id_utworu;
```

Query Result x	
SQL All Rows Fetched: 60 in 0.095 seconds	
NAZWA_UZYTEKOWNIKA	NAZWA_UTWORU
1 John Doe	All Girls Are the Same
2 John Doe	zabij dla mnie
3 John Doe	LOCKJAW
4 Jane Smith	Betrayal
5 Jane Smith	Ty znajesh
6 Jane Smith	Die Alone
7 Alice Johnson	I`m Still
8 Alice Johnson	GOD MODE
9 Alice Johnson	Not So Weak
10 Bob Brown	zabij dla mnie
11 Bob Brown	Ja YETI
12 Bob Brown	Castle on the Hill
13 Eva Williams	All Girls Are the Same
14 Eva Williams	KRAINA LODU
15 Eva Williams	Ty znajesh
16 Michael Davis	MOJO JOJO
17 Michael Davis	Sinnin` and Sinnin`

5. Pobierz nazwę artysty i nazwę wszystkich utworów w albumie, ale tylko w przypadku albumów wydanych od 2022 roku

Query Builder	
<pre>SELECT a.nazwa AS nazwa_artysty, u.nazwa AS nazwa_utworu FROM album al JOIN artysta a ON al.artysta_id_artyst = a.id_artyst JOIN utwory u ON al.id_album = u.album_id_album WHERE al.data_wydania >= TO_DATE('2022-01-01', 'YYYY-MM-DD');</pre>	
Query Result x	
SQL All Rows Fetched: 65 in 0.007 seconds	
NAZWA_ARTYSTY	NAZWA_UTWORU
1 Chivas	anyżowe żelki
2 Chivas	narcyz
3 Chivas	koleżanko mojej byłej
4 Chivas	drzewko
5 Chivas	to ostatni raz gdy jadę nad bałtyk
6 Chivas	miałem kolegę bartka
7 Chivas	kupić jej gaz czy torebkę?
8 Chivas	jesteś najlepszy/najlepsza
9 Chivas	mam chyba za drogie auto
10 Chivas	narcyz w klubie
11 Szpaku	UZUMAKI FORMA OSTATECZNA
12 Szpaku	KRAINA LODU
13 Szpaku	CRAZY FROG
14 Szpaku	CHAINSAW MAN
15 Szpaku	MAŁY RUI
16 Szpaku	FULL METAL SKIT
17 Szpaku	TOTORO

- Pobierz nazwę artysty, albumu i czas trwania (suma) utworów z tego albumu

QL Developer

Team Tools Window Help

Welcome Page x 173163 x 12345 x

Worksheet Query Builder

```

SELECT
  a.nazwa AS nazwa_artysty,
  al.nazwa AS nazwa_albumu,
  SUM(u.czas_trwania) AS suma_czasu_trwania_w_sekundach
FROM
  album al
JOIN
  artysta a ON al.artysta_id_artyst = a.id_artyst
JOIN
  utwory u ON al.id_album = u.album_id_album
GROUP BY
  a.nazwa, al.nazwa;

```

Query Result x

SQL | All Rows Fetched: 14 in 0.015 seconds

	NAZWA_ARTYSTY	NAZWA_ALBUMU	SUMA_CZASU_TRWANIA_W_SEKUNDACH
1	Chivas	Młody Say10	1510
2	Chivas	nauczyłem się przeklinać	2294
3	Szpaku	Uzumaki forma ostateczna	2119
4	Szpaku	Atypowy	2177
5	ZYZZ	One last time	156
6	ZYZZ	Say it right	150
7	BAKI	Safe and sound	143
8	BAKI	Runaway	199
9	Juice WRLD	Goodbye and Good Riddance	2423
10	SiM	Playdead	2606
11	Ed Sheeran	÷	3567
12	Golubenko	Sentymentalnist	622
13	Sxmpra	The evil in which we thrive	700
14	ovg!	Oppai and Thicc Thighs	1875

- Pobierz nazwę użytkownika, nazwę listy odtwarzania i wszystkie utwory na liście odtwarzania, ale tylko w przypadku list odtwarzania utworzonych w ciągu ostatniego roku

```
SELECT u.imie AS nazwa_uzytkownika, l.nazwa AS nazwa_listy, u2.nazwa AS nazwa_utworu
FROM uzytkownik u
JOIN listy_odtwarzania l ON u.id_uzytkownik = l.uzytkownik_id_uzytkownik
JOIN utwory_listy ul ON l.id_listy = ul.listy_odtwarzania_id_listy
JOIN utwory u2 ON ul.utwory_id_utworu = u2.id_utworu
WHERE EXTRACT(YEAR FROM l.data_stworzenia) = 2023;
```

Query Result x

SQL | All Rows Fetched: 138 in 0.154 seconds

	NAZWA_UZYTKOWNIKA	NAZWA_LISTY	NAZWA_UTWORU
13	Jane Smith	Playlista na weekend	Die Alone
14	Jane Smith	Playlista na weekend	odkąd nauczyłem się przeklinać
15	Jane Smith	Playlista na weekend	Intro
16	Jane Smith	Playlista na weekend	SCOOBY DOO
17	Jane Smith	Playlista na weekend	Blue Balls
18	Jane Smith	Playlista na weekend	xo
19	Jane Smith	Playlista na weekend	Galway Girl
20	Jane Smith	Playlista na weekend	All Girls Are the Same
21	Jane Smith	Playlista na weekend	Hoe! Hoe! Hoe!
22	Alice Johnson	Melodie relaksacyjne	I`m Still
23	Alice Johnson	Melodie relaksacyjne	GOD MODE
24	Alice Johnson	Melodie relaksacyjne	Not So Weak
25	Alice Johnson	Melodie relaksacyjne	TOTORO
26	Alice Johnson	Melodie relaksacyjne	How Would You Feel
27	Alice Johnson	Melodie relaksacyjne	to dopiero początek
28	Alice Johnson	Melodie relaksacyjne	Sad Song

8. Znajdź użytkowników, którzy obserwują najwięcej artystów

Worksheet Query Builder

```
SELECT u.imie AS nazwa_uzytkownika, COUNT(o.artysta_id artyst) AS liczba_obserwowanych_artystow
FROM uzytkownik u
LEFT JOIN obserwujacy o ON u.id_uzytkownik = o.uzytkownik_id_uzytkownik
GROUP BY u.imie
ORDER BY liczba_obserwowanych_artystow DESC;
```

Query Result x

SQL | All Rows Fetched: 20 in 0.005 seconds

	NAZWA_UZYTKOWNIKA	LICZBA_OBSERWOWANYCH_ARTYSTOW
1	Jane Smith	8
2	Michael Davis	6
3	Lucas Martinez	5
4	James White	5
5	Isabella Martinez	5
6	John Doe	4
7	Sophia Wilson	4
8	Bob Brown	4
9	Emma Clark	4
10	Ava Thompson	4
11	Liam Rodriguez	4
12	Benjamin Scott	3
13	William Anderson	3
14	Oliver Garcia	3
15	Alice Johnson	3
16	Alexander Lee	3
17	Mia Hall	2
18	Eva Williams	2
19	Grace Perez	2
20	Charlotte Green	1

9. Wybierz wszystkie informacje o użytkownikach, którzy urodzili się po 2000 roku

```
SELECT *
FROM uzytkownik
WHERE EXTRACT(YEAR FROM data_urodzenia) > 2000;
```

Query Result x

SQL | All Rows Fetched: 8 in 0.104 seconds

ID_UZYTEKOWNIK	IMIE	EMAIL	HASLO	DATA_URODZENIA	ZDJECIE_UZYTEKOWNIKA	TYP_UZYTEKOWNIKA
1	7 Sophia Wilson	sophia@example.com	sophiapass	22-NOV-01	(BLOB)	premium
2	8 Oliver Garcia	oliver@example.com	oliverpass	08-AUG-03	(BLOB)	premium
3	9 Isabella Martinez	isabella@example.com	isabellapass	14-MAR-05	(BLOB)	premium
4	10 Liam Rodriguez	liam@example.com	liampass	27-JUN-04	(BLOB)	premium
5	16 Mia Hall	mia@example.com	miapass	15-SEP-02	(BLOB)	premium
6	17 Benjamin Scott	ben@example.com	benpass	05-NOV-01	(BLOB)	premium
7	19 Lucas Martinez	lucas@example.com	lucaspas	03-MAR-04	(BLOB)	premium
8	20 Grace Perez	grace@example.com	gracepass	11-JUL-03	(BLOB)	premium

10. Znajdź nazwy wszystkich albumów wydanych w roku 2023

```
SELECT nazwa
FROM album
WHERE EXTRACT(YEAR FROM data_wydania) = 2023;
```

Query Result x | Query Result 1 x | Query Result 2 x

SQL | All Rows Fetched: 5 in 0.05 seconds

NAZWA
1 Uzumaki forma ostateczna
2 Playdead
3 Sentymenalist
4 The evil in which we thrive
5 Oppai and Thicc Thighs

11. Znajdź nazwę artysty i nazwę wszystkich utworów, które mają czas trwania powyżej 4 minut

```
SELECT a.nazwa AS nazwa_artysty, u.nazwa AS nazwa_utworu
FROM utwory u
JOIN album al ON u.album_id_album = al.id_album
JOIN artysta a ON al.artysta_id_artyst = a.id_artyst
WHERE u.czas_trwania > 4* 60; -- 4 minut przeliczone na sekundy (4 * 60)
```

Query Result x

SQL | All Rows Fetched: 8 in 0.003 seconds

	NAZWA_ARTYSTY	NAZWA_UTWORU
1	Juice WRLD	I'll Be Fine
2	SiM	UNDER THE TREE
3	SiM	The Rumbling
4	Ed Sheeran	Castle on the Hill
5	Ed Sheeran	Perfect
6	Ed Sheeran	Hearts Don't Break Around Here
7	Ed Sheeran	How Would You Feel
8	Ed Sheeran	Save Myself

12. Znajdź informacje o użytkownikach, którzy są typem "premium"

```
SELECT *
FROM uzytkownik
WHERE typ_uzytkownika = 'premium';
```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x

SQL | All Rows Fetched: 12 in 0.006 seconds

	ID_UZYTEKOWNIK	IMIE	EMAIL	HASLO	DATA_URODZENIA	ZDJECIE_UZYTEKOWNIKA	TYP_UZYTEKOWNIKA
1	5	Eva Williams	eva@example.com	evapass	30-SEP-99	(BLOB)	premium
2	6	Michael Davis	michael@example.com	michaelpass	18-APR-87	(BLOB)	premium
3	7	Sophia Wilson	sophia@example.com	sophiapass	22-NOV-01	(BLOB)	premium
4	8	Oliver Garcia	oliver@example.com	oliverpass	08-AUG-03	(BLOB)	premium
5	9	Isabella Martinez	isabella@example.com	isabellapass	14-MAR-05	(BLOB)	premium
6	10	Liam Rodriguez	liam@example.com	liampass	27-JUN-04	(BLOB)	premium
7	15	Alexander Lee	alex@example.com	alexpase	08-MAY-00	(BLOB)	premium
8	16	Mia Hall	mia@example.com	miapass	15-SEP-02	(BLOB)	premium
9	17	Benjamin Scott	ben@example.com	benpass	05-NOV-01	(BLOB)	premium
10	18	Charlotte Green	charlotte@example.com	charlottepass	20-APR-99	(BLOB)	premium
11	19	Lucas Martinez	lucas@example.com	lucaspase	03-MAR-04	(BLOB)	premium
12	20	Grace Perez	grace@example.com	gracepass	11-JUL-03	(BLOB)	premium

13. Znajdź nazwę użytkownika i nazwę listy odtwarzania dla użytkowników, dokonali płatosc bonem lub blikiem

```
SELECT u.imie AS nazwa_uzytkownika, l.nazwa AS nazwa_listy
FROM platnosc p
JOIN uzytkownik u ON p.uzytkownik_id = u.id_uzytkownik
JOIN listy_odtwarzania l ON u.id_uzytkownik = l.uzytkownik_id_uzytkownik
WHERE p.metoda_platnosci IN ('Bon podarunkowy', 'BLIK');
```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x

SQL | All Rows Fetched: 8 in 0.003 seconds

NAZWA_UZYTEKOWNIKA	NAZWA_LISTY
1 Isabella Martinez	Najlepsze przeboje
2 William Anderson	Ulubione melodyjki
3 James White	Mix hip-hopowy
4 Emma Clark	Lista imprezowa
5 Alexander Lee	Muzyka do pracy
6 Mia Hall	Popowe hity
7 Benjamin Scott	Rockowe klasyki
8 Charlotte Green	Lista na zimne wieczory

14. Znajdź wszystkie informacje o użytkownikach którzy mają subskrypcje student

```
SELECT u.id_uzytkownik, u.imie, u.email, u.haslo, u.data_urodzenia, u.zdjecie_uzytkownika, u.typ_uzytkownika
FROM uzytkownik u
Left JOIN uzytkownik_plan_subskrypcyjny ups ON u.id_uzytkownik = ups.uzytkownik_id_uzytkownik
join plan_subskrypcyjny on ups.plan_subskrypcyjny_id_plan_subskrypcyjny = plan_subskrypcyjny.id_plan_subskrypcyjny
WHERE plan_subskrypcyjny.nazwa = 'Student';
```

Query Result x

SQL | All Rows Fetched: 4 in 0.003 seconds

ID_UZYTEKOWNIKA	IMIE	EMAIL	HASLO	DATA_URODZENIA	ZDJECIE_UZYTEKOWNIKA	TYP_UZYTEKOWNIKA
1	9 Isabella Martinez	isabella@example.com	isabellapass	14-MAR-05	(BLOB)	premium
2	10 Liam Rodriguez	liam@example.com	liampass	27-JUN-04	(BLOB)	premium
3	15 Alexander Lee	alex@example.com	alexpass	08-MAY-00	(BLOB)	premium
4	16 Mia Hall	mia@example.com	miapass	15-SEP-02	(BLOB)	premium

15. Znajdź imie użytkownika i nazwy polubionych utworów dla użytkowników kto dokonał płatność od września 2022 do stycznia 2023 włącznie

```
SELECT u.imie AS imie_uzytkownika, ut.nazwa AS nazwa_utworu
FROM uzytkownik u
JOIN platnosc p ON u.id_uzytkownik = p.uzytkownik_id_uzytkownik
JOIN likes l ON u.id_uzytkownik = l.uzytkownik_id_uzytkownik
JOIN utwory ut ON l.utwory_id_utworu = ut.id_utworu
WHERE p.data_platnosc BETWEEN TO_DATE('2022-09-01', 'YYYY-MM-DD') AND TO_DATE('2023-01-31', 'YYYY-MM-DD');
```

Query Result x

SQL | All Rows Fetched: 27 in 0.006 seconds

IMIE_UZYTEKOWNIKA	NAZWA_UTWORU
1 John Doe	All Girls Are the Same
2 John Doe	zabij dla mnie
3 John Doe	LOCKJAW
4 Isabella Martinez	zabij dla mnie
5 Isabella Martinez	New Man
6 Isabella Martinez	I'm Still
7 Liam Rodriguez	Runaway
8 Liam Rodriguez	GOD MODE
9 Liam Rodriguez	UNDER THE TREE
10 William Anderson	Alicja
11 William Anderson	Die Alone
12 William Anderson	JDM DRIFT EDIT VERY TRENDY HOUSE PHONK
13 Emma Clark	All Girls Are the Same
14 Emma Clark	TOTORO
15 Emma Clark	New Man
16 Benjamin Scott	NA ZAWSZE

2. Skrypty bazy danych

--Tworzenie użytkownika i nadanie uprawnień

```
create user c##123456 identified by 123456;
```

```
ALTER USER c##123456 DEFAULT TABLESPACE USERS QUOTA UNLIMITED ON USERS;
```

```
ALTER USER c##123456 TEMPORARY TABLESPACE TEMP;
```

```
GRANT CREATE SESSION, CREATE VIEW, ALTER SESSION, CREATE SEQUENCE TO c##123456;
```

```
GRANT CREATE SYNONYM, CREATE DATABASE LINK, RESOURCE , UNLIMITED TABLESPACE  
TO c##123456;
```

```
GRANT CREATE TRIGGER to c##123456;
```

```
ALTER USER c##123456 ACCOUNT UNLOCK;
```

```
GRANT CREATE SESSION TO c##123456;
```

```
GRANT CREATE TABLE TO c##123456;
```

```
GRANT CREATE ANY TABLE TO c##123456;
```

```
GRANT DROP ANY TABLE TO c##123456;
```

```
GRANT INSERT ANY TABLE TO c##123456;
```

```
GRANT SELECT ANY TABLE TO c##123456;
```

```
CREATE TABLE album (
```

```
    id_album    INTEGER NOT NULL,
```

```
    nazwa       VARCHAR2(50) NOT NULL,
```

```
    data_wydania DATE,
```

```
    obraz       BLOB,
```

```
    artysta_id_artyst INTEGER NOT NULL
```

```
);
```

```
ALTER TABLE album ADD CONSTRAINT album_pk PRIMARY KEY ( id_album );
```

```
CREATE TABLE artysta (
```

```
    id_artyst INTEGER NOT NULL,
```

```
    nazwa     VARCHAR2(50) NOT NULL,
```

```
    gatunek   VARCHAR2(50 CHAR),
```

```
    obraz     BLOB
```

```
);
```

```
ALTER TABLE artysta ADD CONSTRAINT artysta_pk PRIMARY KEY ( id_artyst );
```

```
CREATE TABLE funkcja_premium (
```

```
    id_funkcja_premium INTEGER NOT NULL,
```

```

    nazwa          VARCHAR2(50 CHAR) NOT NULL
);

ALTER TABLE funkcja_premium ADD CONSTRAINT unkcja_premium_pk PRIMARY KEY (
id_funkcja_premium );

CREATE TABLE likes (

    like_data      DATE,

    like_time      DATE,

    uzytkownik_id_uzytkownik INTEGER NOT NULL,

    utwory_id_utworu    INTEGER NOT NULL

);

ALTER TABLE likes ADD CONSTRAINT likes_pk PRIMARY KEY ( uzytkownik_id_uzytkownik,

                utwory_id_utworu );

CREATE TABLE listy_odtwarzania (

    id_listy        INTEGER NOT NULL,

    nazwa           VARCHAR2(50 CHAR) NOT NULL,

    obraz           BLOB,

    data_stworzenia    DATE NOT NULL,

    uzytkownik_id_uzytkownik INTEGER

);

ALTER TABLE listy_odtwarzania ADD CONSTRAINT listy_odtwarzania_pk PRIMARY KEY ( id_listy );

CREATE TABLE obserwujacy (

    artysta_id_artyst    INTEGER,

    uzytkownik_id_uzytkownik INTEGER

);

CREATE TABLE plan_subskrypcyjny (

    id_plan_subskrypcyjny INTEGER NOT NULL,

    nazwa                 VARCHAR2(50 CHAR) NOT NULL,

    wartosc               NUMBER NOT NULL,

    opis                  VARCHAR2(500 CHAR) NOT NULL

);

ALTER TABLE plan_subskrypcyjny ADD CONSTRAINT plan_subskrypcyjny_pk PRIMARY KEY (
id_plan_subskrypcyjny );

CREATE TABLE platnosc (

    id_platnosc          INTEGER NOT NULL,

```

```

    metoda_platnosci    VARCHAR2(50 CHAR) NOT NULL,
    data_platnosc       DATE NOT NULL,
    kwota                NUMBER NOT NULL,
    uzytkownik_id_uzytkownik INTEGER NOT NULL
);
ALTER TABLE platnosc ADD CONSTRAINT platnosc_pk PRIMARY KEY ( id_platnosc,
                                                                uzytkownik_id_uzytkownik );

CREATE TABLE utwory (
    id_utworu    INTEGER NOT NULL,
    nazwa        VARCHAR2(50 CHAR) NOT NULL,
    czas_trwania INTEGER NOT NULL,
    sciezka      VARCHAR2(500 CHAR),
    album_id_album INTEGER NOT NULL
);
ALTER TABLE utwory ADD CONSTRAINT utwory_pk PRIMARY KEY ( id_utworu );

CREATE TABLE utwory_listy (
    listy_odtwarzania_id_listy INTEGER,
    utwory_id_utworu          INTEGER
);

CREATE TABLE uzytkownik (
    id_uzytkownik    INTEGER NOT NULL,
    imie             VARCHAR2(50) NOT NULL,
    email            VARCHAR2(50 CHAR) NOT NULL,
    haslo            VARCHAR2(50 CHAR) NOT NULL,
    data_urodzenia   DATE,
    zdjecie_uzytkownika BLOB,
    typ_uzytkownika  VARCHAR2(20 CHAR) NOT NULL
);
ALTER TABLE uzytkownik ADD CONSTRAINT uzytkownik_pk PRIMARY KEY ( id_uzytkownik );

CREATE TABLE uzytkownik_plan_subskrypcyjny (
    uzytkownik_id_uzytkownik          INTEGER NOT NULL,
    -- ERROR: Column name length exceeds maximum allowed length(30)
    plan_subskrypcyjny_id_plan_subskrypcyjny INTEGER NOT NULL);
-- ERROR: PK name length exceeds maximum allowed length(30)

```

```
ALTER TABLE uzytkownik_plan_subskrypcyjny ADD CONSTRAINT uzytkownik_plan_subskrypcyjny_pk  
PRIMARY KEY ( uzytkownik_id_uzytkownik, plan_subskrypcyjny_id_plan_subskrypcyjny);
```

```
CREATE TABLE uzytkownik_premium (
```

```
-- ERROR: Column name length exceeds maximum allowed length(30)
```

```
    funkcja_premium_id_funkcja_premium INTEGER NOT NULL,
```

```
    uzytkownik_id_uzytkownik          INTEGER NOT NULL
```

```
);
```

```
ALTER TABLE uzytkownik_premium ADD CONSTRAINT uzytkownik_premium_pk PRIMARY KEY ( funkcja_premium_id_funkcja_premium, uzytkownik_id_uzytkownik );
```

```
ALTER TABLE album
```

```
    ADD CONSTRAINT album_artysta_fk FOREIGN KEY ( artysta_id_artyst )
```

```
    REFERENCES artysta ( id_artyst );
```

```
ALTER TABLE likes
```

```
    ADD CONSTRAINT likes_utwory_fk FOREIGN KEY ( utwory_id_utworu )
```

```
    REFERENCES utwory ( id_utworu );
```

```
ALTER TABLE likes
```

```
    ADD CONSTRAINT likes_uzytkownik_fk FOREIGN KEY ( uzytkownik_id_uzytkownik )
```

```
    REFERENCES uzytkownik ( id_uzytkownik );
```

```
-- ERROR: FK name length exceeds maximum allowed length(30)
```

```
ALTER TABLE listy_odtwarzania
```

```
    ADD CONSTRAINT listy_odtwarzania_uzytkownik_fk FOREIGN KEY ( uzytkownik_id_uzytkownik )
```

```
    REFERENCES uzytkownik ( id_uzytkownik );
```

```
ALTER TABLE obserwujacy
```

```
    ADD CONSTRAINT obserwujacy_artysta_fk FOREIGN KEY ( artysta_id_artyst )
```

```
    REFERENCES artysta ( id_artyst );
```

```
ALTER TABLE obserwujacy
```

```
    ADD CONSTRAINT obserwujacy_uzytkownik_fk FOREIGN KEY ( uzytkownik_id_uzytkownik )
```

```
    REFERENCES uzytkownik ( id_uzytkownik );
```

```
ALTER TABLE platnosc
```

```
    ADD CONSTRAINT platnosc_uzytkownik_fk FOREIGN KEY ( uzytkownik_id_uzytkownik )
```

```
    REFERENCES uzytkownik ( id_uzytkownik );
```

```
ALTER TABLE utwory
```

```
    ADD CONSTRAINT utwory_album_fk FOREIGN KEY ( album_id_album )
```

```
    REFERENCES album ( id_album );
```

```
-- ERROR: FK name length exceeds maximum allowed length(30)
```

```

ALTER TABLE utwory_listy

ADD CONSTRAINT utwory_listy_listy_odtwarzania_fk FOREIGN KEY ( listy_odtwarzania_id_listy )

REFERENCES listy_odtwarzania ( id_listy );

ALTER TABLE utwory_listy

ADD CONSTRAINT utwory_listy_utwory_fk FOREIGN KEY ( utwory_id_utworu )

REFERENCES utwory ( id_utworu );

-- ERROR: FK name length exceeds maximum allowed length(30)

ALTER TABLE uzytkownik_plan_subskrypcyjny

ADD CONSTRAINT uzytkownik_plan_subskrypcyjny_plan_subskrypcyjny_fk FOREIGN KEY (
plan_subskrypcyjny_id_plan_subskrypcyjny )

REFERENCES plan_subskrypcyjny ( id_plan_subskrypcyjny );

-- ERROR: FK name length exceeds maximum allowed length(30)

ALTER TABLE uzytkownik_plan_subskrypcyjny

ADD CONSTRAINT uzytkownik_plan_subskrypcyjny_uzytkownik_fk FOREIGN KEY (
uzytkownik_id_uzytkownik )

REFERENCES uzytkownik ( id_uzytkownik );

-- ERROR: FK name length exceeds maximum allowed length(30)

ALTER TABLE uzytkownik_premium

ADD CONSTRAINT uzytkownik_premium_funkcja_premium_fk FOREIGN KEY (
funkcja_premium_id_funkcja_premium )

REFERENCES funkcja_premium ( id_funkcja_premium );

-- ERROR: FK name length exceeds maximum allowed length(30)

ALTER TABLE uzytkownik_premium

ADD CONSTRAINT uzytkownik_premium_uzytkownik_fk FOREIGN KEY ( uzytkownik_id_uzytkownik )

REFERENCES uzytkownik ( id_uzytkownik );

INSERT INTO uzytkownik (id_uzytkownik, imie, email, haslo, data_urodzenia, zdjecie_uzytkownika,
typ_uzytkownika)

VALUES

(1, 'John Doe', 'johndoe@example.com', 'password123', TO_DATE('1990-05-15', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'regular'),

(2, 'Jane Smith', 'janesmith@example.com', 'qwerty789', TO_DATE('1988-12-20', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'regular'),

(3, 'Alice Johnson', 'alice@example.com', 'alicepass', TO_DATE('1995-02-10', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'regular'),

(4, 'Bob Brown', 'bobbrown@example.com', 'bobbypass', TO_DATE('1992-07-05', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'regular'),

(5, 'Eva Williams', 'eva@example.com', 'evapass', TO_DATE('1999-09-30', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

```

(6, 'Michael Davis', 'michael@example.com', 'michaelpass', TO_DATE('1987-04-18', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(7, 'Sophia Wilson', 'sophia@example.com', 'sophiapass', TO_DATE('2001-11-22', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(8, 'Oliver Garcia', 'oliver@example.com', 'oliverpass', TO_DATE('2003-08-08', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(9, 'Isabella Martinez', 'isabella@example.com', 'isabellapass', TO_DATE('2005-03-14', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(10, 'Liam Rodriguez', 'liam@example.com', 'liampass', TO_DATE('2004-06-27', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(11, 'William Anderson', 'will@example.com', 'willpass', TO_DATE('1993-10-25', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'regular'),

(12, 'Ava Thompson', 'ava@example.com', 'avapass', TO_DATE('1996-12-12', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'regular'),

(13, 'James White', 'james@example.com', 'jamespass', TO_DATE('1986-08-30', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'regular'),

(14, 'Emma Clark', 'emma@example.com', 'emmapass', TO_DATE('1990-02-28', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'regular'),

(15, 'Alexander Lee', 'alex@example.com', 'alexpasse', TO_DATE('2000-05-08', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(16, 'Mia Hall', 'mia@example.com', 'miapass', TO_DATE('2002-09-15', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(17, 'Benjamin Scott', 'ben@example.com', 'benpass', TO_DATE('2001-11-05', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(18, 'Charlotte Green', 'charlotte@example.com', 'charlottepass', TO_DATE('1999-04-20', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(19, 'Lucas Martinez', 'lucas@example.com', 'lucaspasse', TO_DATE('2004-03-03', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium'),

(20, 'Grace Perez', 'grace@example.com', 'gracepass', TO_DATE('2003-07-11', 'YYYY-MM-DD'),
EMPTY_BLOB(), 'premium');

INSERT INTO artysta (id_artyst, nazwa, gatunek, obraz)

VALUES

(1, 'Chivas', 'Rap', empty_blob()),

(2, 'Szpaku', 'Rap', empty_blob()),

(3, 'ZYZZ', 'Gym Hardstyle', empty_blob()),

(4, 'BAKI', 'Gym Gardstyle', empty_blob()),

(5, 'Juice WRLD', 'Rap', empty_blob()),

(6, 'SiM', 'Rock', empty_blob()),

(7, 'Ed Sheeran', 'Pop', empty_blob()),

(8, 'Golubenko', 'Pop', empty_blob()),

```

(9, 'Sxmpira', 'Gym Hardstyle', empty_blob()),
(10, 'ovg!', 'Phonk', empty_blob())
INSERT INTO album (id_album, nazwa, data_wydania, obraz, artysta_id_artyst)
VALUES
(1, 'Młody Say10', TO_DATE('2022-09-22', 'YYYY-MM-DD'), empty_blob(), 1),
(2, 'nauczyłem się przeklinać', TO_DATE('2021-05-21', 'YYYY-MM-DD'), empty_blob(), 1),
(3, 'Uzumaki forma ostateczna', TO_DATE('2023-07-27', 'YYYY-MM-DD'), empty_blob(), 2),
(4, 'Atypowy', TO_DATE('2018-10-20', 'YYYY-MM-DD'), empty_blob(), 2),
(5, 'One last time', TO_DATE('2022-09-28', 'YYYY-MM-DD'), empty_blob(), 3),
(6, 'Say it right', TO_DATE('2022-04-20', 'YYYY-MM-DD'), empty_blob(), 3),
(7, 'Safe and sound', TO_DATE('2022-09-10', 'YYYY-MM-DD'), empty_blob(), 4),
(8, 'Runaway', TO_DATE('2022-09-11', 'YYYY-MM-DD'), empty_blob(), 4),
(9, 'Goodbye and Good Riddance', TO_DATE('2018-12-10', 'YYYY-MM-DD'), empty_blob(), 5),
(10, 'Playdead', TO_DATE('2023-09-27', 'YYYY-MM-DD'), empty_blob(), 6),
(11, '-', TO_DATE('2017-03-03', 'YYYY-MM-DD'), empty_blob(), 7),
(12, 'Sentymentalnist', TO_DATE('2023-03-29', 'YYYY-MM-DD'), empty_blob(), 8),
(13, 'The evil in which we thrive', TO_DATE('2023-02-03', 'YYYY-MM-DD'), empty_blob(), 9),
(14, 'Oppai and Thicc Thighs', TO_DATE('2023-02-19', 'YYYY-MM-DD'), empty_blob(), 10)
INSERT INTO plan_subskrypcyjny (id_plan_subskrypcyjny, nazwa, wartosc, opis)
VALUES
(1, 'Regular', 5.00, 'Plan regularny pozwala korzystać z podstawowych funkcji'),
(2, 'Premium', 10.00, 'Plan premium pozwala korzystać z funkcji premium'),
(3, 'Rodzinny', 15.00, 'Plan rodzinny 4 osobom korzystają z jednej subskrypcji.'),
(4, 'Student', 7.00, 'Plan dla studentów');
INSERT INTO funkcja_premium (id_funkcja_premium, nazwa)
VALUES
(1, 'Odtwarzanie dowolnego utworu'),
(2, 'Pomijanie utworów bez limitu'),
(3, 'Bez reklamy'),
(4, 'Słuchanie offline'),
(5, 'Dźwięk wysokiej jakości');

INSERT INTO utwory (id_utworu, nazwa, czas_trwania, sciezka, album_id_album)

```


VALUES

- (1, 'anyżowe żelki', 133, null, 1),
- (2, 'narcyz', 143, null, 1),
- (3, 'koleżanko mojej byłej', 152, null, 1),
- (4, 'drzewko', 167, null, 1),
- (5, 'to ostatni raz gdy jadę nad bałtyk', 155, null, 1),
- (6, 'miałem kolegę bartka', 166, null, 1),
- (7, 'kupić jej gaz czy torebkę?', 147, null, 1),
- (8, 'jesteś najlepszy/najlepsza', 144, null, 1),
- (9, 'mam chyba za drogie auto', 177, null, 1),
- (10, 'narcyz w klubie', 126, null, 1),
- (11, 'to dopiero początek', 97, null, 2),
- (12, 'odkąd nauczyłem się przeklinać', 149, null, 2),
- (13, 'generacja hip hop', 172, null, 2),
- (14, 'xo', 164, null, 2),
- (15, 'palenie zabija', 145, null, 2),
- (16, 'death metalowy t-shirt', 157, null, 2),
- (17, 'mam na twarzy krew i tym razem nie jest sztuczna', 162, null, 2),
- (18, 'jakbym miał w ręku kłamkę', 167, null, 2),
- (19, 'comme des garçons', 160, null, 2),
- (20, 'młody say10', 154, null, 2),
- (21, 'charger', 160, null, 2),
- (22, 'hellcat redeye', 147, null, 2),
- (23, 'zabij dla mnie', 144, null, 2),
- (24, 'to nie koniec', 132, null, 2),
- (25, 'najlepsi pod słońcem', 184, null, 2),
- (26, 'UZUMAKI FORMA OSTATECZNA', 198, null, 3),
- (27, 'KRAINA LODU', 213, null, 3),
- (28, 'CRAZY FROG', 237, null, 3),
- (29, 'CHAINSAW MAN', 182, null, 3),
- (30, 'MAŁY RUI', 202, null, 3),
- (31, 'FULL METAL SKIT', 121, null, 3),
- (32, 'TOTORO', 176, null, 3),
- (33, 'NA ZAWSZE', 135, null, 3),
- (34, 'NIETOPERZE', 185, null, 3),

(35, 'BYŁEM MŁODY', 196, null, 3),
(36, 'SCOOBY DOO', 128, null, 3),
(37, 'OSTATNI NINJA', 146, null, 3),
(38, 'Alicja', 158, null, 4),
(39, 'W Krainie Czarów', 185, null, 4),
(40, 'UFO', 215, null, 4),
(41, 'Szlam', 234, null, 4),
(42, 'Lavender Town', 233, null, 4),
(43, 'Zombie', 221, null, 4),
(44, 'Hinata', 189, null, 4),
(45, 'Nieważne', 186, null, 4),
(46, 'Oddajemy krew Wampirom', 214, null, 4),
(47, 'Ja YETI', 192, null, 4),
(48, 'MOJO JOJO', 150, null, 4),
(49, 'One Last Time', 156, null, 5),
(50, 'Say it right', 150, null, 6),
(51, 'Safe and Sound', 143, null, 7),
(52, 'Runaway', 199, null, 8),
(53, 'Intro', 75, null, 9),
(54, 'All Girls Are the Same', 166, null, 9),
(55, 'Lucid Dreams', 240, null, 9),
(56, 'Lean Wit Me', 176, null, 9),
(57, 'I'm Still', 193, null, 9),
(58, 'Betrayal', 64, null, 9),
(59, 'Candles', 184, null, 9),
(60, 'Scared of Love', 171, null, 9),
(61, 'Used To', 176, null, 9),
(62, 'Karma', 122, null, 9),
(63, 'Hurt Me', 75, null, 9),
(64, 'Black and White', 187, null, 9),
(65, 'Long Gone', 187, null, 9),
(66, 'End of the Road', 163, null, 9),
(67, 'I'll Be Fine', 244, null, 9),
(68, 'PLAYDEAD', 120, null, 10),
(69, 'RED', 234, null, 10),

(70, 'HiDE and SEEK', 217, null,10),
 (71, 'SWEET DREAMS', 210, null,10),
 (72, 'Die Alone', 137, null,10),
 (73, 'TOO LATE', 161, null,10),
 (74, 'KiSS OF DEATH', 232, null,10),
 (75, 'Sad Song', 203, null,10),
 (76, 'BBT', 210, null,10),
 (77, 'DO THE DANCE', 196, null,10),
 (78, 'Not So Weak', 161, null,10),
 (79, 'UNDER THE TREE', 271, null,10),
 (80, 'The Rumbling', 254, null,10),
 (81, 'Eraser', 227,null,11),
 (82, 'Castle on the Hill', 261,null,11),
 (83, 'Dive', 238,null,11),
 (84, 'Shape of You', 233,null,11),
 (85, 'Perfect', 263,null,11),
 (86, 'Galway Girl', 170,null,11),
 (87, 'Happier', 207,null,11),
 (88, 'New Man', 189,null,11),
 (89, 'Hearts Don` t Break Around Here', 248,null,11),
 (90, 'What Do I Know', 237,null,11),
 (91, 'How Would You Feel', 280,null,11),
 (92, 'Supermarket Flowers', 221,null,11),
 (93, 'Barcelona', 191,null,11),
 (94, 'Bibia Be Ye Ye', 176,null,11),
 (95, 'Nancy Mulligan', 179,null,11),
 (96, 'Save Myself', 247,null,11),
 (97, 'Intro', 13,null, 12),
 (98, 'Nevahomist', 109,null, 12),
 (99, 'Vylikuj', 115,null, 12),
 (100, 'Sentymentalnist', 134,null, 12),
 (101, 'Ty znajesh', 120,null, 12),
 (102, 'Marionetka', 104,null, 12),
 (103, 'Outro', 27,null, 12),
 (104, 'GOD MODE', 163, NULL,13),

```

(105, 'JDM DRIFT EDIT VERY TRENDY HOUSE PHONK', 137, NULL,13),
(106, 'LOCKJAW', 134, NULL,13),
(107, 'BLUE EYED MESSIAH', 127, NULL,13),
(108, 'NIGHTMARE', 139, NULL,13),
(109, 'Blue Balls', 159, null, 14),
(110, 'Sippin` and Simpín`, 107, null, 14),
(111, 'Virginity Syndrome', 144, null, 14),
(112, 'Hentai Circulation', 148, null, 14),
(113, 'Rent-A-Hoe', 121, null, 14),
(114, 'Overflow', 141, null, 14),
(115, 'Loli Pop', 153, null, 14),
(116, 'GAY', 111, null, 14),
(117, 'Anime Thighs', 158, null, 14),
(118, 'Bilingual', 155, null, 14),
(119, 'Hoe! Hoe! Hoe!', 143, null, 14),
(120, 'Health Project', 71, null, 14),
(121, 'Robo-Peen', 123, null, 14),
(122, 'osu! virgins', 141, null, 14);
INSERT INTO uzytkownik_premium(uzytkownik_id_uzytkownik, funkcja_premium_id_funkcja_premium)
values
(5,1),(5,2),(5,3),(5,4),(5,5),
(6,1),(6,2),(6,3),(6,4),(6,5),
(7,1),(7,2),(7,3),(7,4),(7,5),
(8,1),(8,2),(8,3),(8,4),(8,5),
(9,1),(9,2),(9,3),(9,4),(9,5),
(10,1),(10,2),(10,3),(10,4),(10,5),
(15,1),(15,2),(15,3),(15,4),(15,5),
(16,1),(16,2),(16,3),(16,4),(16,5),
(17,1),(17,2),(17,3),(17,4),(17,5),
(18,1),(18,2),(18,3),(18,4),(18,5),
(19,1),(19,2),(19,3),(19,4),(19,5),
(20,1),(20,2),(20,3),(20,4),(20,5);

```

```

INSERT INTO obserwujacy (uzytkownik_id_uzytkownik, artysta_id_artyst)

```

values

(1,1),(1,2),(1,4),(1,10),
(2,1),(2,2),(2,9),(2,4),(2,3),(2,5),(2,7),(2,10),
(3,4),(3,6),(3,9),
(4,7),(4,8),(4,9),(4,10),
(5,1),(5,2),
(6,3),(6,2),(6,7),(6,6),(6,5),(6,4),
(7,1),(7,5),(7,9),(7,10),
(8,5),(8,3),(8,7),
(9,10),(9,9),(9,1),(9,6),(9,5),
(10,4),(10,1),(10,2),(10,5),
(11,8),(11,1),(11,10),
(12,3),(12,4),(12,8),(12,9),
(13,1),(13,5),(13,2),(13,7),(13,6),
(14,4),(14,8),(14,9),(14,10),
(15,1),(15,6),(15,7),
(16,2),(16,3),
(17,4),(17,5),(17,7),
(18,2),
(19,4),(19,7),(19,10),(19,5),(19,3),
(20,1),(20,3);

INSERT INTO

uzytkownik_plan_subskrypcyjny(uzytkownik_id_uzytkownik,plan_subskrypcyjny_id_plan_subskrypcyjny)

values

(1,1),(2,1),(3,1),(4,1),
(11,1),(12,1),(13,1),(14,1),
(5,3),(6,3),(7,3),(8,3),
(9,4),(10,4),(15,4),(16,4),
(17,2),(18,2),(19,2),(20,2);

INSERT INTO listy_odtwarzania (id_listy, nazwa, obraz, uzytkownik_id_uzytkownik, data_stworzenia)

VALUES

(1, 'Ulubione utwory', EMPTY_BLOB(), 1, TO_DATE('2023-01-10', 'YYYY-MM-DD')), -- Data po 2023
(2, 'Playlista na weekend', EMPTY_BLOB(), 2, TO_DATE('2023-02-15', 'YYYY-MM-DD')), -- Data po 2023
(3, 'Melodie relaksacyjne', EMPTY_BLOB(), 3, TO_DATE('2023-04-20', 'YYYY-MM-DD')), -- Data po 2023

(4, 'Energiczne rytmy', EMPTY_BLOB(), 4, TO_DATE('2022-12-05', 'YYYY-MM-DD')), -- Data przed 2023

(5, 'Mix hitów', EMPTY_BLOB(), 5, TO_DATE('2022-11-30', 'YYYY-MM-DD')), -- Data przed 2023

(6, 'Chillout na wieczór', EMPTY_BLOB(), 6, TO_DATE('2022-08-18', 'YYYY-MM-DD')), -- Data przed 2023

(7, 'Lista do biegania', EMPTY_BLOB(), 7, TO_DATE('2023-03-25', 'YYYY-MM-DD')), -- Data po 2023

(8, 'Utwory na podróż', EMPTY_BLOB(), 8, TO_DATE('2022-09-10', 'YYYY-MM-DD')), -- Data przed 2023

(9, 'Najlepsze przeboje', EMPTY_BLOB(), 9, TO_DATE('2023-06-15', 'YYYY-MM-DD')), -- Data po 2023

(10, 'Playlist na trening', EMPTY_BLOB(), 10, TO_DATE('2022-07-20', 'YYYY-MM-DD')), -- Data przed 2023

(11, 'Ulubione melodyjki', EMPTY_BLOB(), 11, TO_DATE('2023-08-02', 'YYYY-MM-DD')), -- Data po 2023

(12, 'Relaksujące dźwięki', EMPTY_BLOB(), 12, TO_DATE('2022-05-12', 'YYYY-MM-DD')), -- Data przed 2023

(13, 'Mix hip-hopowy', EMPTY_BLOB(), 13, TO_DATE('2022-04-03', 'YYYY-MM-DD')), -- Data przed 2023

(14, 'Lista imprezowa', EMPTY_BLOB(), 14, TO_DATE('2023-01-28', 'YYYY-MM-DD')), -- Data po 2023

(15, 'Muzyka do pracy', EMPTY_BLOB(), 15, TO_DATE('2022-12-30', 'YYYY-MM-DD')), -- Data przed 2023

(16, 'Popowe hity', EMPTY_BLOB(), 16, TO_DATE('2023-11-05', 'YYYY-MM-DD')), -- Data po 2023

(17, 'Rockowe klasyki', EMPTY_BLOB(), 17, TO_DATE('2022-10-17', 'YYYY-MM-DD')), -- Data przed 2023

(18, 'Lista na zimne wieczory', EMPTY_BLOB(), 18, TO_DATE('2023-09-22', 'YYYY-MM-DD')), -- Data po 2023

(19, 'Melancholijne nuty', EMPTY_BLOB(), 19, TO_DATE('2022-06-07', 'YYYY-MM-DD')), -- Data przed 2023

(20, 'Dobre rytmy', EMPTY_BLOB(), 20, TO_DATE('2023-04-12', 'YYYY-MM-DD')); -- Data po 2023

INSERT INTO utwory_listy (listy_odtwarzania_id_listy, utwory_id_utworu)

VALUES

(1, 54),(1, 23),(1, 106),(1, 78),(1, 92),(1, 41),(1, 115),(1, 7),(1, 88),(1, 33),

(2, 58),(2, 101),(2, 72),(2, 12),(2, 97),(2, 36),(2, 109),(2, 14),(2, 86),(2, 54),(2, 119),

(3, 57),(3, 104),(3, 78),(3, 32),(3, 91),(3, 11),(3, 75),(3, 23),(3, 118),(3, 40),(3, 86),(3, 102),

(4, 23),(4, 47),(4, 82),(4, 101),(4, 35),(4, 76),(4, 11),(4, 91),(4, 58),(4, 99),(4, 22),(4, 67),(4, 117),

(5, 54),(5, 27),(5, 101),(5, 78),(5, 92),(5, 37),(5, 110),(5, 65),(5, 85),(5, 19),(5, 12),(5, 73),(5, 44),(5, 106),

(6, 48),(6, 110),(6, 76),(6, 17),(6, 92),(6, 3),(6, 41),(6, 89),(6, 29),(6, 60),(6, 105),(6, 7),(6, 82),(6, 50),(6, 121),

(7, 27),(7, 81),(7, 5),(7, 104),(7, 36),(7, 95),(7, 73),(7, 19),(7, 62),(7, 110),(7, 47),(7, 89),(7, 12),(7, 58),(7, 91),(7, 33),

(8, 17),(8, 45),(8, 92),(8, 33),(8, 104),(8, 21),(8, 77),(8, 56),(8, 88),(8, 12),(8, 67),(8, 41),(8, 109),(8, 25),(8, 98),(8, 7),(8, 72),

(9, 23),(9, 88),(9, 57),(9, 110),(9, 14),(9, 45),(9, 76),(9, 101),(9, 33),(9, 94),(9, 4),(9, 78),(9, 39),(9, 67),(9, 81),(9, 25),(9, 106),(9, 11),

(10, 52),(10, 104),(10, 79),(10, 15),(10, 91),(10, 37),(10, 63),(10, 110),(10, 22),(10, 76),(10, 29),(10, 45),(10, 8),(10, 98),(10, 27),(10, 116),(10, 61),(10, 94),(10, 40),

(11, 38),(11, 72),(11, 105),(11, 20),(11, 89),(11, 14),(11, 116),(11, 53),(11, 97),(11, 42),(11, 79),(11, 61),(11, 32),(11, 25),(11, 111),(11, 93),(11, 13),(11, 70),(11, 56),

(12, 36),(12, 85),(12, 19),(12, 104),(12, 72),(12, 53),(12, 27),(12, 91),(12, 7),(12, 65),(12, 14),(12, 49),(12, 108),(12, 82),(12, 33),(12, 101),(12, 12),(12, 68),

(13, 50),(13, 89),(13, 3),(13, 27),(13, 112),(13, 45),(13, 71),(13, 16),(13, 98),(13, 22),(13, 74),(13, 5),(13, 120),(13, 34),(13, 105),(13, 90),(13, 61),

(14, 54),(14, 32),(14, 88),(14, 101),(14, 19),(14, 73),(14, 12),(14, 107),(14, 65),(14, 40),(14, 90),(14, 8),(14, 116),(14, 27),(14, 51),(14, 76),

(15, 7),(15, 92),(15, 31),(15, 106),(15, 18),(15, 85),(15, 54),(15, 23),(15, 111),(15, 40),(15, 75),(15, 99),(15, 12),(15, 63),(15, 110),

(16, 23),(16, 56),(16, 89),(16, 34),(16, 101),(16, 72),(16, 45),(16, 11),(16, 77),(16, 99),(16, 21),(16, 67),(16, 112),(16, 38),

(17, 33),(17, 102),(17, 77),(17, 12),(17, 56),(17, 88),(17, 44),(17, 109),(17, 65),(17, 23),(17, 98),(17, 31),(17, 75),

(18, 29),(18, 57),(18, 81),(18, 14),(18, 105),(18, 43),(18, 72),(18, 93),(18, 38),(18, 66),(18, 99),(18, 20),

(19, 45),(19, 87),(19, 12),(19, 34),(19, 109),(19, 76),(19, 23),(19, 98),(19, 54),(19, 21),(19, 112),

(20, 67),(20, 42),(20, 105),(20, 18),(20, 81),(20, 35),(20, 93),(20, 12),(20, 58),(20, 77);

INSERT INTO likes (like_data, like_time, uzytkownik_id_uzytkownik, utwory_id_utworu)

VALUES

(TO_DATE('2023-01-15','YYYY-MM-DD'), TO_TIMESTAMP('12:30','HH24:MI'), 1, 54),
(TO_DATE('2023-01-15','YYYY-MM-DD'), TO_TIMESTAMP('13:45','HH24:MI'), 1, 23),
(TO_DATE('2023-01-16','YYYY-MM-DD'), TO_TIMESTAMP('09:20','HH24:MI'), 1, 106),
(TO_DATE('2023-02-10','YYYY-MM-DD'), TO_TIMESTAMP('10:15','HH24:MI'), 2, 58),
(TO_DATE('2023-02-10','YYYY-MM-DD'), TO_TIMESTAMP('11:30','HH24:MI'), 2, 101),
(TO_DATE('2023-02-11','YYYY-MM-DD'), TO_TIMESTAMP('09:45','HH24:MI'), 2, 72),
(TO_DATE('2022-06-20','YYYY-MM-DD'), TO_TIMESTAMP('08:30','HH24:MI'), 3, 57),
(TO_DATE('2022-06-21','YYYY-MM-DD'), TO_TIMESTAMP('12:45','HH24:MI'), 3, 104),
(TO_DATE('2022-06-22','YYYY-MM-DD'), TO_TIMESTAMP('17:20','HH24:MI'), 3, 78),
(TO_DATE('2022-07-10','YYYY-MM-DD'), TO_TIMESTAMP('09:15','HH24:MI'), 4, 23),
(TO_DATE('2022-07-11','YYYY-MM-DD'), TO_TIMESTAMP('14:00','HH24:MI'), 4, 47),
(TO_DATE('2022-07-12','YYYY-MM-DD'), TO_TIMESTAMP('19:30','HH24:MI'), 4, 82),
(TO_DATE('2022-08-05','YYYY-MM-DD'), TO_TIMESTAMP('10:30','HH24:MI'), 5, 54),
(TO_DATE('2022-08-06','YYYY-MM-DD'), TO_TIMESTAMP('15:45','HH24:MI'), 5, 27),
(TO_DATE('2022-08-07','YYYY-MM-DD'), TO_TIMESTAMP('20:00','HH24:MI'), 5, 101),

(TO_DATE('2022-09-10','YYYY-MM-DD'), TO_TIMESTAMP('12:15','HH24:MI'), 6, 48),
 (TO_DATE('2022-09-11','YYYY-MM-DD'), TO_TIMESTAMP('16:30','HH24:MI'), 6, 110),
 (TO_DATE('2022-09-12','YYYY-MM-DD'), TO_TIMESTAMP('18:45','HH24:MI'), 6, 76),
 (TO_DATE('2022-07-15','YYYY-MM-DD'), TO_TIMESTAMP('09:30','HH24:MI'), 7, 27),
 (TO_DATE('2022-07-15','YYYY-MM-DD'), TO_TIMESTAMP('14:45','HH24:MI'), 7, 81),
 (TO_DATE('2022-07-15','YYYY-MM-DD'), TO_TIMESTAMP('18:20','HH24:MI'), 7, 5),
 (TO_DATE('2022-08-01','YYYY-MM-DD'), TO_TIMESTAMP('10:15','HH24:MI'), 8, 17),
 (TO_DATE('2022-08-01','YYYY-MM-DD'), TO_TIMESTAMP('14:30','HH24:MI'), 8, 45),
 (TO_DATE('2022-08-01','YYYY-MM-DD'), TO_TIMESTAMP('20:00','HH24:MI'), 8, 92),
 (TO_DATE('2022-09-01','YYYY-MM-DD'), TO_TIMESTAMP('09:30','HH24:MI'), 9, 23),
 (TO_DATE('2022-09-01','YYYY-MM-DD'), TO_TIMESTAMP('12:45','HH24:MI'), 9, 88),
 (TO_DATE('2022-09-01','YYYY-MM-DD'), TO_TIMESTAMP('18:20','HH24:MI'), 9, 57),
 (TO_DATE('2022-10-15','YYYY-MM-DD'), TO_TIMESTAMP('08:00','HH24:MI'), 10, 52),
 (TO_DATE('2022-10-15','YYYY-MM-DD'), TO_TIMESTAMP('13:30','HH24:MI'), 10, 104),
 (TO_DATE('2022-10-15','YYYY-MM-DD'), TO_TIMESTAMP('17:45','HH24:MI'), 10, 79),
 (TO_DATE('2022-09-18','YYYY-MM-DD'), TO_TIMESTAMP('10:20','HH24:MI'), 11, 38),
 (TO_DATE('2022-09-18','YYYY-MM-DD'), TO_TIMESTAMP('15:45','HH24:MI'), 11, 72),
 (TO_DATE('2022-09-18','YYYY-MM-DD'), TO_TIMESTAMP('19:30','HH24:MI'), 11, 105),
 (TO_DATE('2022-07-25','YYYY-MM-DD'), TO_TIMESTAMP('08:10','HH24:MI'), 12, 36),
 (TO_DATE('2022-07-25','YYYY-MM-DD'), TO_TIMESTAMP('13:20','HH24:MI'), 12, 85),
 (TO_DATE('2022-07-25','YYYY-MM-DD'), TO_TIMESTAMP('17:45','HH24:MI'), 12, 19),
 (TO_DATE('2022-08-12','YYYY-MM-DD'), TO_TIMESTAMP('09:30','HH24:MI'), 13, 50),
 (TO_DATE('2022-08-12','YYYY-MM-DD'), TO_TIMESTAMP('15:45','HH24:MI'), 13, 89),
 (TO_DATE('2022-08-12','YYYY-MM-DD'), TO_TIMESTAMP('19:20','HH24:MI'), 13, 3),
 (TO_DATE('2022-09-25','YYYY-MM-DD'), TO_TIMESTAMP('10:15','HH24:MI'), 14, 54),
 (TO_DATE('2022-09-25','YYYY-MM-DD'), TO_TIMESTAMP('14:20','HH24:MI'), 14, 32),
 (TO_DATE('2022-09-25','YYYY-MM-DD'), TO_TIMESTAMP('18:45','HH24:MI'), 14, 88),
 (TO_DATE('2022-08-10','YYYY-MM-DD'), TO_TIMESTAMP('08:00','HH24:MI'), 15, 7),
 (TO_DATE('2022-08-11','YYYY-MM-DD'), TO_TIMESTAMP('12:30','HH24:MI'), 15, 92),
 (TO_DATE('2022-08-12','YYYY-MM-DD'), TO_TIMESTAMP('16:45','HH24:MI'), 15, 31),
 (TO_DATE('2022-08-13','YYYY-MM-DD'), TO_TIMESTAMP('09:15','HH24:MI'), 16, 23),
 (TO_DATE('2022-08-14','YYYY-MM-DD'), TO_TIMESTAMP('14:20','HH24:MI'), 16, 56),
 (TO_DATE('2022-08-15','YYYY-MM-DD'), TO_TIMESTAMP('18:00','HH24:MI'), 16, 89),
 (TO_DATE('2022-09-01','YYYY-MM-DD'), TO_TIMESTAMP('12:30','HH24:MI'), 17, 33),
 (TO_DATE('2022-09-02','YYYY-MM-DD'), TO_TIMESTAMP('15:45','HH24:MI'), 17, 102),


```
(TO_DATE('2022-09-03','YYYY-MM-DD'), TO_TIMESTAMP( '18:20','HH24:MI'), 17, 77),
(TO_DATE('2022-09-04','YYYY-MM-DD'), TO_TIMESTAMP( '09:15','HH24:MI'), 18, 29),
(TO_DATE('2022-09-05','YYYY-MM-DD'), TO_TIMESTAMP( '14:20','HH24:MI'), 18, 57),
(TO_DATE('2022-09-06','YYYY-MM-DD'), TO_TIMESTAMP( '17:50','HH24:MI'), 18, 81),
(TO_DATE('2022-09-07','YYYY-MM-DD'), TO_TIMESTAMP( '10:30','HH24:MI'), 19, 45),
(TO_DATE('2022-09-08','YYYY-MM-DD'), TO_TIMESTAMP( '13:45','HH24:MI'), 19, 87),
(TO_DATE('2022-09-09','YYYY-MM-DD'), TO_TIMESTAMP( '16:00','HH24:MI'), 19, 12),
(TO_DATE('2022-11-20','YYYY-MM-DD'), TO_TIMESTAMP( '08:15','HH24:MI'), 20, 67),
(TO_DATE('2022-11-21','YYYY-MM-DD'), TO_TIMESTAMP( '11:30','HH24:MI'), 20, 42),
(TO_DATE('2022-11-22','YYYY-MM-DD'), TO_TIMESTAMP( '14:45','HH24:MI'), 20, 105);
```

```
ALTER TABLE likes ADD time_only VARCHAR(5);
```

```
UPDATE likes SET time_only = TO_CHAR(like_time, 'HH24:MI');
```

```
ALTER TABLE likes DROP COLUMN like_time;
```

```
insert into platnosc (id_platnosc, metoda_platnosci, data_platnosc, kwota, uzytkownik_id_uzytkownik)
VALUES
```

```
(1, 'GOOGLE PAY', TO_DATE('2023-01-14','YYYY-MM-DD'), 5.00, 1),
(2, 'APPLE PAY', TO_DATE('2023-02-09','YYYY-MM-DD'), 5.00, 2),
(3, 'Karta płatnicza', TO_DATE('2022-06-19','YYYY-MM-DD'), 5.00, 3),
(4, 'Karta płatnicza', TO_DATE('2022-07-9','YYYY-MM-DD'), 5.00, 4),
(5, 'Karta płatnicza', TO_DATE('2022-07-14','YYYY-MM-DD'), 3.75, 5),
(6, 'Karta płatnicza', TO_DATE('2022-07-14','YYYY-MM-DD'), 3.75, 6),
(7, 'Karta płatnicza', TO_DATE('2022-07-14','YYYY-MM-DD'), 3.75, 7),
(8, 'Karta płatnicza', TO_DATE('2022-07-14','YYYY-MM-DD'), 3.75, 8),
(9, 'BLIK', TO_DATE('2022-09-01','YYYY-MM-DD'), 7.00, 9),
(10, 'GOOGLE PAY', TO_DATE('2022-10-15','YYYY-MM-DD'), 7.00, 10),
(11, 'BLIK', TO_DATE('2022-09-17','YYYY-MM-DD'), 5.00, 11),
(12, 'APPLE PAY', TO_DATE('2022-07-24','YYYY-MM-DD'), 5.00, 12),
(13, 'BLIK', TO_DATE('2022-08-11','YYYY-MM-DD'), 5.00, 13),
(14, 'BLIK', TO_DATE('2022-09-24','YYYY-MM-DD'), 5.00, 14),
(15, 'Bon podarunkowy', TO_DATE('2022-08-09','YYYY-MM-DD'), 0.00, 15),
(16, 'Bon podarunkowy', TO_DATE('2022-08-12','YYYY-MM-DD'), 0.00, 16),
(17, 'Bon podarunkowy', TO_DATE('2022-09-01','YYYY-MM-DD'), 0.00, 17),
(18, 'Bon podarunkowy', TO_DATE('2022-09-03','YYYY-MM-DD'), 0.00, 18),
(19, 'Karta płatnicza', TO_DATE('2022-09-08','YYYY-MM-DD'), 10.00, 19),
(20, 'Karta płatnicza', TO_DATE('2022-11-20','YYYY-MM-DD'), 10.00, 20)
```

4. Określenie kierunków rozwoju aplikacji

1. Wprowadzenie rekomendacji utworów, artystów, albumów, gatunków(Similarity)
2. Możliwość stworzenia wspólnych list odtwarzania(grupa użytkowników tworzy wspólną listę i każdy może nią zarządzać)
3. Możliwość dla artystów stworzyć tymczasowe zespoły pod innym imieniem
4. Możliwość tworzenia personalizowanej statystyki dla każdego użytkownika pod koniec roku

5. Opracowanie doświadczeń wynikających z realizacji projektu

Podczas realizacji projektu największą trudność sprawiał dobór odpowiedniego typu danych do kolumn, zmiana diagramu pod czas propozycji kwerend(edycja tabel, połączeń) .

6. Wykaz literatury

https://mfiles.pl/pl/index.php/Model_związków_encji

<https://www.w3schools.com/sql/default.asp>

<https://www.sqlpedia.pl/projektowanie-i-normalizacja-bazy-danych/>

<https://docs.oracle.com/cloud/help/pl/analytics-cloud/ACSMD/GUID-4EE8C58D-391B-46A9-B0B3-9FA91B8159D1.htm#BILUG664>

A. Silberschatz, H. F. Korth, S. Sudarshan Database System Concepts, 17 edition McGraw-Hill Higher Education.