



POLITECHNIKA RZESZOWSKA

im. Ignacego Łukasiewicza

WYDZIAŁ MATEMATYKI I FIZYKI STOSOWANEJ

Szymon Czebański, Krzysztof Kopiec, Andrii Kotovych,

172932, 173158, 173163

Inżynieria i Analiza Danych Rok 3

**Projekt Spotify**

APLIKACJE BAZODANOWE

Rzeszów 2025



## Spis treści

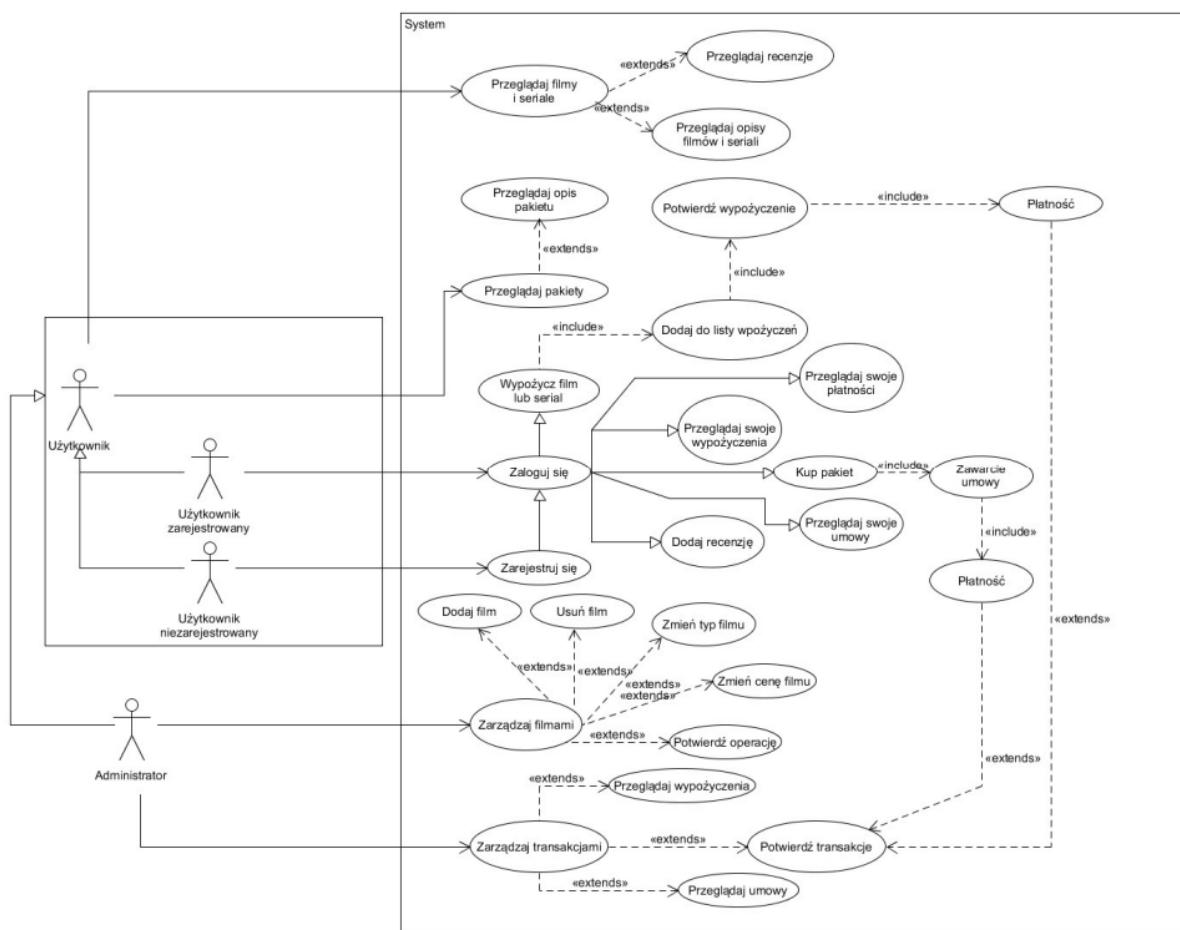
1.	Wstęp.....	5
2.	Relacje między procedurami i funkcjami w pakiecie .....	6
	HISTORIAODTWORZEN .....	8
	PODSUMOWANIEODTWARZANIA .....	9
	LOG_PRZYPOMNIENIA.....	9
	RAPORTODTWORZEN .....	10
3.	Funkcje .....	11
	Obsluga_liczby .....	11
	Obsluga_litery .....	12
	Obsluga_symbole.....	13
	CzyUzytkownikPremium .....	14
	LiczbaOdtworzenUtwor .....	15
	SredniCzasOdtworzenUtwor .....	15
	NajczesciejOdtwarzanyUtworDnia .....	16
	LacznyCzasOdtwarzaniaUzytkownika.....	17
	ZnajdzNajczesciejOdtwarzanyUtworUzytkownika .....	19
	SprawdzStatusSubskrypcji .....	22
	PobierzCzasTrwaniaSubskrypcji.....	23
	ObliczKwoteZaOkres .....	24
	CzyPlatnoscZrealizowana.....	25
	IleDniDoKonca.....	26
4.	Procedury.....	27
	DodajUzytkownika .....	27
	UsunUzytkownika .....	30
	ZaktualizujDaneUzytkownika.....	32
	ZmienHaslo .....	34
	GenerujRaportOdtworzen .....	37
	ZapiszNajczesciejOdtwarzanyUtwor .....	39
	PodsumowanieOdtwarzaniaUzytkownika .....	41
	UsunStareDaneOdtworzen .....	44
	ZarejestrujPlatnosc .....	45
	ZmienPlanSubskrypcji .....	46
	AnulujSubskrypcje.....	47
	OplacSubskrypcje .....	48
	PrzypomnijOZakonczeniuSubskrypcji .....	49

5.	Pakiety .....	51
	OBSLUGAUZYTKOWNIK .....	51
	STATYSTYKIODTWORZEN.....	58
	ZARZADZANIEPLATNOSCiami.....	66
6.	APEX.....	71
1.	Import bazy danych i pakietów z procedurami i funkcjami.....	71
2.	Tworzenie aplikacji bazodanowej.....	72
3.	Grupa stron dotyczących zarządzania użytkownikami .....	75
4.	Grupa stron dotyczących statystyk utworów i użytkowników .....	86
5.	Grupa stron dotyczących płatności i subskrypcji .....	101
6.	Strony dodatkowe.....	113
7.	Export aplikacji.....	120
	Podsumowanie .....	123

## 1. Wstęp

Do realizacji projektu wykorzystano schemat bazy danych stworzony podczas zajęć z przedmiotu **Bazy Danych**, oparty na modelu Spotify. W trakcie pracy zastosowane zostaną odpowiednie składnie oraz struktury obejmujące funkcje, procedury, pakiety, kolekcje i inne kluczowe elementy języka **PL/SQL**. Dodatkowo, do wizualizacji oraz obsługi aplikacji bazodanowej wykorzystana zostanie platforma **Oracle APEX**, umożliwiająca efektywne zarządzanie danymi i interfejsem użytkownika.

Use Cases bazy z przedmiotu bazy danych



## 2. Relacje między procedurami i funkcjami w pakiecie

Diagram Pakietu ObslugaUzytkownika

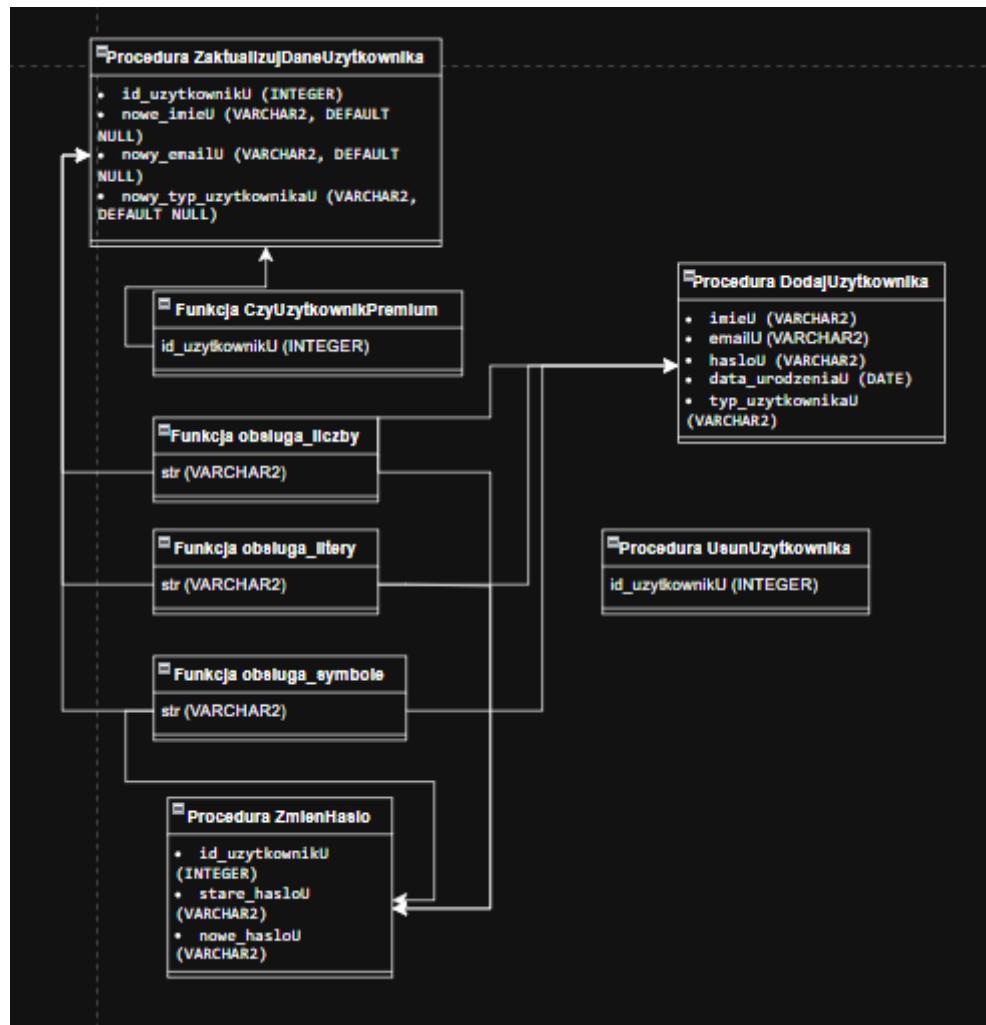


Diagram Pakietu StatystykiOdtworzen

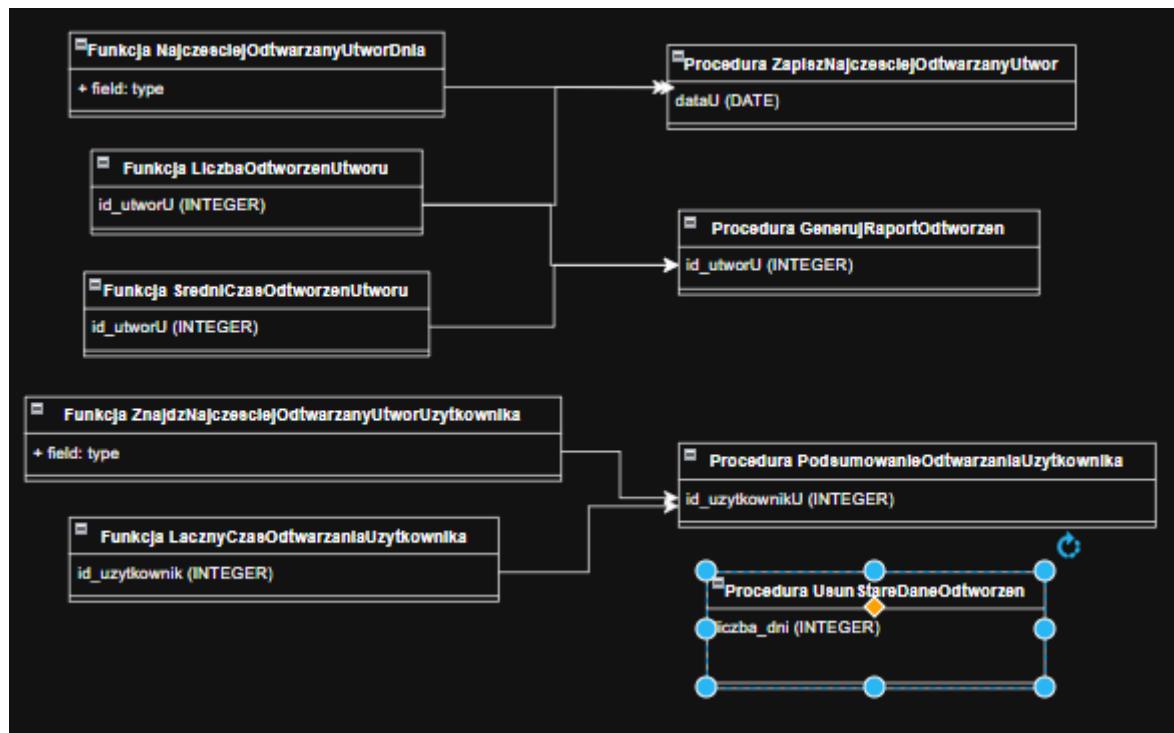
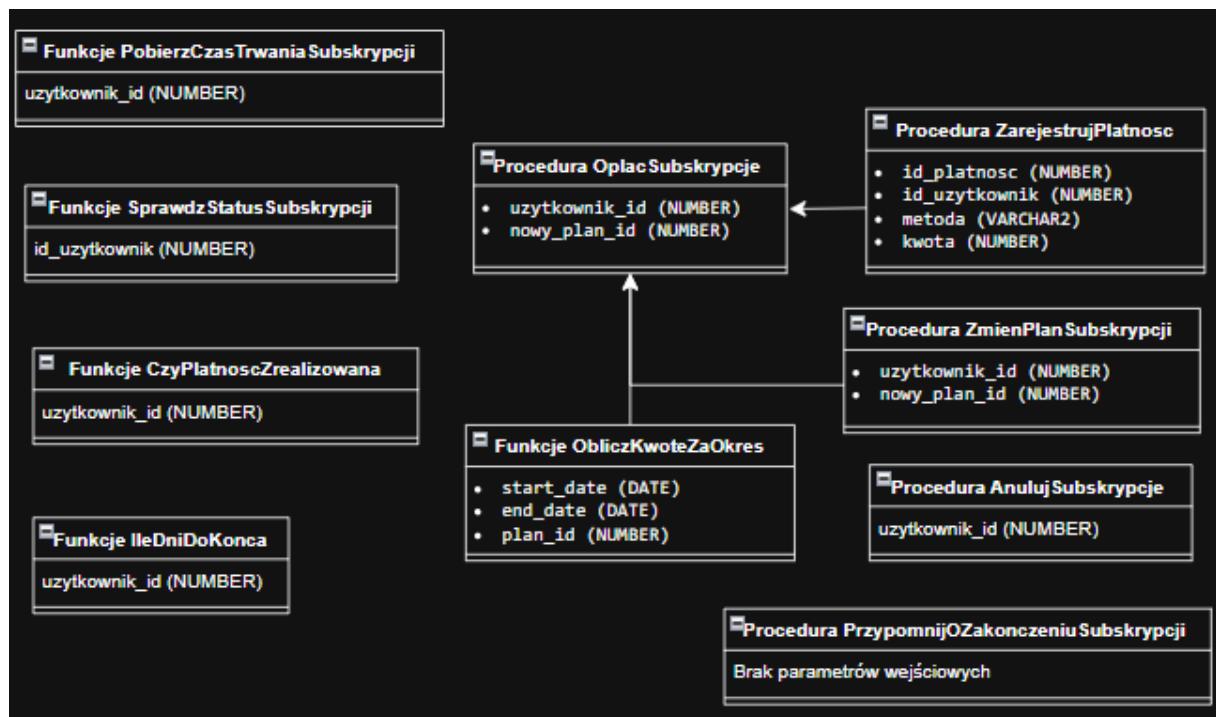


Diagram Pakietu ZarzadzaniePlatnosciami



Dodatkowo dla lepszego funkcjonowania funkcji i procedur do bazy danych zostały dodane kilka nowych tabel:

## HISTORIAODTWORZEN

Tabela HISTORIAODTWORZEN przechowuje historię odtworzeń utworów przez użytkowników, zawierając następujące kolumny:

ID\_ODTWRZENIA: Unikalny identyfikator odtworzenia (klucz główny).

ID\_UTWORU: Klucz obcy do tabeli UTWORY, identyfikujący odtwarzany utwór.

ID\_UZYTKOWNIK: Klucz obcy do tabeli UZYTKOWNIK, identyfikujący użytkownika.

CZAS\_ODTWRZENIA: Czas trwania odtworzenia (w sekundach).

DATA\_ODTWRZENIA: Data odtworzenia.

```
CREATE TABLE "HISTORIAODTWORZEN"
```

```
(  "ID_ODTWRZENIA" NUMBER(*,0),  
  "ID_UTWORU" NUMBER(*,0) NOT NULL ENABLE,  
  "ID_UZYTKOWNIK" NUMBER(*,0) NOT NULL ENABLE,  
  "CZAS_ODTWRZENIA" NUMBER,  
  "DATA_ODTWRZENIA" DATE,  
  PRIMARY KEY ("ID_ODTWRZENIA")  
  USING INDEX ENABLE  
);
```

```
ALTER TABLE "HISTORIAODTWORZEN" ADD FOREIGN KEY ("ID_UTWORU")
```

```
  REFERENCES "UTWORY" ("ID_UTWORU") ENABLE;
```

```
ALTER TABLE "HISTORIAODTWORZEN" ADD FOREIGN KEY ("ID_UZYTKOWNIK")
```

```
  REFERENCES "UZYTKOWNIK" ("ID_UZYTKOWNIK") ENABLE;
```

## **PODSUMOWANIEODTWARZANIA**

Tabela PODSUMOWANIEODTWARZANIA przechowuje podsumowanie danych dotyczących odtwarzania utworów przez użytkowników:

ID\_UZYTKOWNIKA: Unikalny identyfikator użytkownika (klucz główny).

IMIE\_UZYTKOWNIKA: Imię użytkownika.

LACZNY\_CZAS\_ODTWORZEN: Łączny czas odtworzeń użytkownika (w sekundach).

NAJCZESIEJ\_ODTWARZANY\_UTWOR: Nazwa utworu najczęściej odtwarzanego przez użytkownika.

```
CREATE TABLE "PODSUMOWANIEODTWARZANIA"
```

```
(    "ID_UZYTKOWNIKA" NUMBER,  
    "IMIE_UZYTKOWNIKA" VARCHAR2(100),  
    "LACZNY_CZAS_ODTWORZEN" NUMBER,  
    "NAJCZESIEJ_ODTWARZANY_UTWOR" VARCHAR2(100),  
    PRIMARY KEY ("ID_UZYTKOWNIKA")
```

```
USING INDEX ENABLE
```

```
);
```

## **LOG\_PRZYPOMNIENIA**

Tabela LOG\_PRZYPOMNIENIA przechowuje informacje o wysyłanych przypomnieniach do użytkowników:

DATA\_LOG: Data i czas zapisu logu (znacznik czasu).

EMAIL: Adres e-mail użytkownika, do którego wysłano przypomnienie.

KOMUNIKAT: Szczegóły komunikatu przypomnienia.

```
CREATE TABLE "LOG_PRZYPOMNIENIA"
```

```
(    "DATA_LOG" TIMESTAMP (6),  
    "EMAIL" VARCHAR2(50),  
    "KOMUNIKAT" VARCHAR2(4000)  
);
```

## RAPORTODTWRZEN

Tabela RAPORTODTWRZEN przechowuje szczegółowe informacje dotyczące odtworzeń utworów, które są generowane przez odpowiednie procedury raportujące:

ID\_UTWORU: Identyfikator unikalny dla utworu.

NAZWA\_UTWORU: Nazwa utworu.

NAZWA\_ARTYSTY: Imię/nazwa artysty, który stworzył utwór.

LICZBA\_ODTWRZEN: Całkowita liczba odtworzeń danego utworu.

SREDNI\_CZAS: Średni czas trwania odtworzeń utworu (w sekundach).

```
CREATE TABLE "RAPORTODTWRZEN"
```

```
(    "ID_UTWORU" NUMBER,  
    "NAZWA_UTWORU" VARCHAR2(100),  
    "NAZWA_ARTYSTY" VARCHAR2(100),  
    "LICZBA_ODTWRZEN" NUMBER,  
    "SREDNI_CZAS" NUMBER  
);
```

### 3. Funkcje

#### Obsluga\_liczby

Funkcja sprawdza, czy przekazany łańcuch znaków (str) zawiera jakikolwiek cyfrę.

Wejście: łańcuch znaków typu VARCHAR2.

Wyjście: Wartość BOOLEAN:

TRUE – jeśli w łańcuchu znajduje się przynajmniej jedna cyfra.

FALSE – jeśli brak cyfr lub ciąg jest NULL.

```
FUNCTION obsluga_liczby(str VARCHAR2) RETURN BOOLEAN IS
BEGIN
    IF str IS NULL THEN
        RETURN FALSE; -- Obsługa przypadku, gdy str jest NULL
    END IF;

    FOR i IN 1..LENGTH(str) LOOP
        IF ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('0') AND ASCII('9') THEN
            RETURN TRUE;
        END IF;
    END LOOP;

    RETURN FALSE;
END obsluga_liczby;
```

The screenshot shows a SQL query window with the following content:

```
1 DECLARE
2     v_result BOOLEAN;
3 BEGIN
4     -- Testowanie funkcji z przykładowymi danymi
5     v_result := obsluga_uzitkownik.obsluga_liczby('123abc');
```

Below the code, there is a navigation bar with tabs: Results, Explain, Describe, Saved SQL, and History. The 'Results' tab is selected. The result of the query is displayed as:

Czy zawiera liczby: TRUE

## Obsługa\_litery

Funkcja sprawdza, czy przekazany łańcuch znaków (str) zawiera co najmniej jedną literę.

Wejście: łańcuch znaków typu VARCHAR2.

Wyjście: Wartość BOOLEAN:

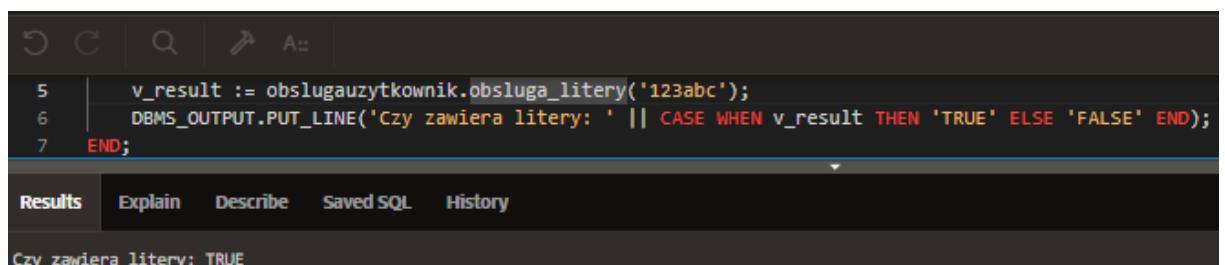
TRUE – jeśli w łańcuchu znajduje się co najmniej jedna litera.

FALSE – jeśli brak liter lub ciąg jest NULL.

```
FUNCTION obsluga_litery(str VARCHAR2) RETURN BOOLEAN IS
BEGIN
    IF str IS NULL THEN
        RETURN FALSE; -- Obsługa przypadku, gdy str jest NULL
    END IF;

    FOR i IN 1..LENGTH(str) LOOP
        IF (ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('A') AND ASCII('Z'))
            OR (ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('a') AND ASCII('z')) THEN
            RETURN TRUE;
        END IF;
    END LOOP;

    RETURN FALSE;
END obsluga_litery;
```



```
5 |     v_result := obslugauzytkownik.obsluga_litery('123abc');
6 |     DBMS_OUTPUT.PUT_LINE('Czy zawiera litery: ' || CASE WHEN v_result THEN 'TRUE' ELSE 'FALSE' END);
7 | END;
```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar icons: Undo, Redo, Search, Paste, Auto-Size.
- Code area:
  - Line 5: `v_result := obslugauzytkownik.obsluga_litery('123abc');`
  - Line 6: `DBMS_OUTPUT.PUT_LINE('Czy zawiera litery: ' || CASE WHEN v_result THEN 'TRUE' ELSE 'FALSE' END);`
  - Line 7: `END;`
- Tab bar: Results (selected), Explain, Describe, Saved SQL, History.
- Output pane:
  - Results tab.
  - Output: `Czy zawiera litery: TRUE`

## **Obsluga\_symbole**

Funkcja sprawdza, czy w podanym ciągu znaków (str) znajduje się:

Konkretna wartość znaku specjalnego, np. @, jeśli zostanie podany parametr expected\_symbol.

Jakikolwiek znak specjalny, jeśli parametr expected\_symbol pozostanie pusty.

```
FUNCTION obsluga_symbole(str VARCHAR2, expected_symbol CHAR DEFAULT NULL)
RETURN BOOLEAN IS
BEGIN
    IF str IS NULL THEN
        RETURN FALSE; -- Obsługa przypadku, gdy str jest NULL
    END IF;

    FOR i IN 1..LENGTH(str) LOOP
        -- Jeśli `expected_symbol` jest ustawiony, sprawdzamy tylko ten znak
        IF expected_symbol IS NOT NULL THEN
            IF SUBSTR(str, i, 1) = expected_symbol THEN
                RETURN TRUE;
            END IF;
        ELSE
            -- Sprawdzenie dowolnego znaku specjalnego
            IF NOT ((ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('A') AND ASCII('Z')) OR
                    (ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('a') AND ASCII('z')) OR
                    (ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('0') AND ASCII('9'))) THEN
                RETURN TRUE;
            END IF;
        END IF;
    END LOOP;

    RETURN FALSE;
END obsluga_symbole;
```

```
4  -- Testowanie funkcji z przykładowymi danymi
5  v_result := obslugauzytkownik.obsluga_symbole('hello!world', '@');
6  DBMS_OUTPUT.PUT_LINE('Czy zawiera symbol specjalny (@): ' || CASE WHEN v_result THEN 'TRUE' ELSE 'FALSE' END);

```

Results Explain Describe Saved SQL History

Czy zawiera symbol specjalny (@): FALSE

## CzyUzytkownikPremium

Funkcja sprawdza, czy użytkownik o podanym identyfikatorze (id\_uzytkownikU) posiada typ subskrypcji "PREMIUM".

Działanie:

Pobiera z tabeli Uzytkownik wartość kolumny typ\_uzytkownika dla użytkownika o podanym ID.

Porównuje pobraną wartość z typem subskrypcji "PREMIUM".

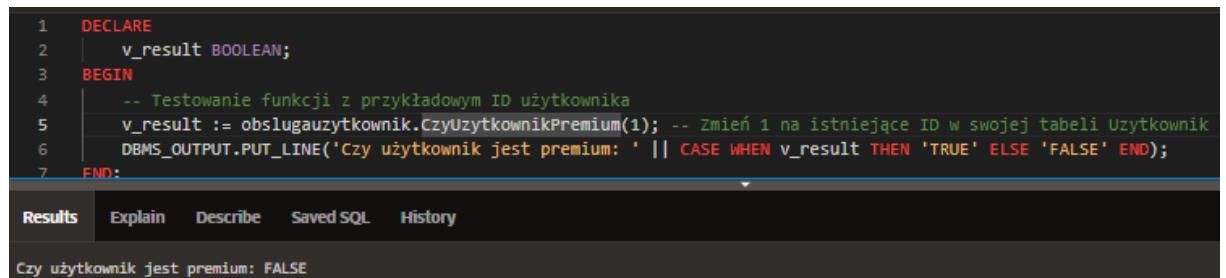
Zwraca:

TRUE, jeśli użytkownik ma typ "PREMIUM".

FALSE, jeśli użytkownik ma inny typ subskrypcji lub jeśli użytkownik o podanym ID nie istnieje.

```
FUNCTION CzyUzytkownikPremium (
    id_uzytkownikU IN INTEGER
) RETURN BOOLEAN IS
    typ_uzytkownikaU VARCHAR2(20);
BEGIN
    -- Pobranie typu subskrypcji użytkownika
    SELECT typ_uzytkownika
    INTO typ_uzytkownikaU
    FROM Uzytkownik
    WHERE id_uzytkownik = id_uzytkownikU;

    -- Sprawdzenie, czy subskrypcja to PREMIUM
    IF typ_uzytkownikaU = 'PREMIUM' THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN FALSE; -- Jeśli użytkownik nie istnieje, zwracamy FALSE
END CzyUzytkownikPremium;
```



```
1  DECLARE
2      v_result BOOLEAN;
3  BEGIN
4      -- Testowanie funkcji z przykładowym ID użytkownika
5      v_result := obslugauzytkownik.CzyUzytkownikPremium(1); -- Zmień 1 na istniejące ID w swojej tabeli Uzytkownik
6      DBMS_OUTPUT.PUT_LINE('Czy użytkownik jest premium: ' || CASE WHEN v_result THEN 'TRUE' ELSE 'FALSE' END);
7  END;
```

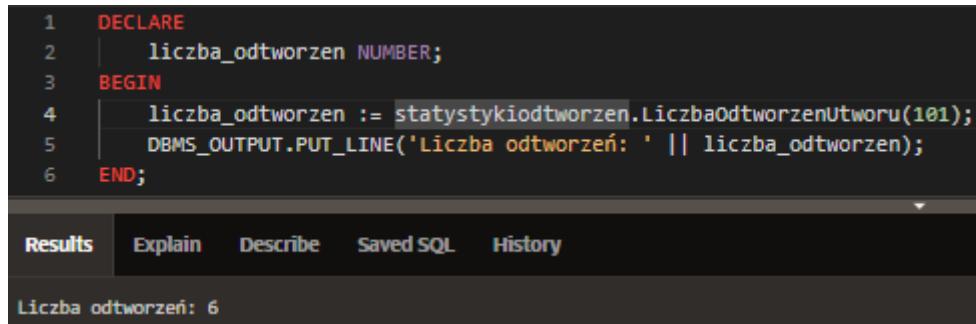
Results Explain Describe Saved SQL History

Czy użytkownik jest premium: FALSE

## LiczbaOdtworzenUtworu

Funkcja LiczbaOdtworzenUtworu oblicza i zwraca liczbę odtworzeń danego utworu na podstawie jego ID (id\_utworU). Wykorzystuje tabelę HistoriaOdtworzen do zliczenia wszystkich rekordów związanych z danym utworem. Jeśli utwór nie istnieje w tabeli, zwróci wartość 0.

```
FUNCTION LiczbaOdtworzenUtworu(id_utworU INT) RETURN NUMBER IS
    liczba NUMBER;
BEGIN
    SELECT COUNT(*) INTO liczba
    FROM HistoriaOdtworzen
    WHERE HistoriaOdtworzen.id_utworu = LiczbaOdtworzenUtworu.id_utworU;
    RETURN liczba;
END LiczbaOdtworzenUtworu;
```



```
1  DECLARE
2      liczba_odtworzen NUMBER;
3  BEGIN
4      liczba_odtworzen := statystykiodtworzen.LiczbaOdtworzenUtworu(101);
5      DBMS_OUTPUT.PUT_LINE('Liczba odtworzeń: ' || liczba_odtworzen);
6  END;
```

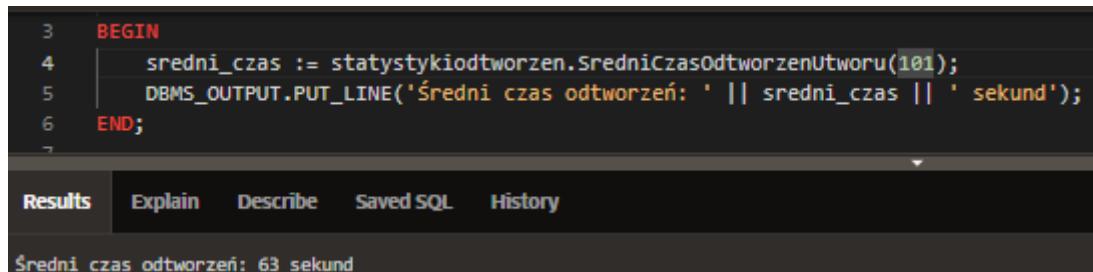
Results Explain Describe Saved SQL History

Liczba odtworzeń: 6

## SredniCzasOdtworzenUtworu

Funkcja SredniCzasOdtworzenUtworu oblicza i zwraca średni czas odtworzeń dla danego utworu na podstawie jego ID (id\_utworU). Korzysta z tabeli HistoriaOdtworzen, gdzie obliczana jest średnia wartość pola czas\_odtworzenia dla wskazanego utworu. Jeśli brak jest danych dla danego utworu, funkcja zwraca NULL.

```
FUNCTION SredniCzasOdtworzenUtworu(id_utworU INT) RETURN NUMBER IS
    sredni_czas NUMBER;
BEGIN
    SELECT AVG(czas_odtworzenia) INTO sredni_czas
    FROM HistoriaOdtworzen
    WHERE HistoriaOdtworzen.id_utworu = SredniCzasOdtworzenUtworu.id_utworU;
    RETURN sredni_czas;
END SredniCzasOdtworzenUtworu;
```



```
3  BEGIN
4      sredni_czas := statystykiodtworzen.SredniCzasOdtworzenUtworu(101);
5      DBMS_OUTPUT.PUT_LINE('Średni czas odtworzeń: ' || sredni_czas || ' sekund');
6  END;
```

Results Explain Describe Saved SQL History

Średni czas odtworzeń: 63 sekund

## NajczesciejOdtwarzanyUtworDnia

Funkcja NajczesciejOdtwarzanyUtworDnia zwraca ID utworu, który był najczęściej odtwarzany w podanym dniu (data). Wyszukuje dane w tabeli HistoriaOdtworzen, grupując je po ID utworu i sortując według liczby odtworzeń w porządku malejącym. Zwracany jest tylko pierwszy rekord (utwór z największą liczbą odtworzeń). Jeśli brak danych dla danego dnia, funkcja zwróci błąd NO\_DATA\_FOUND.

```
FUNCTION NajczesciejOdtwarzanyUtworDnia(data DATE) RETURN NUMBER IS
    id_utwor NUMBER;
BEGIN
    SELECT id_utwor
    INTO id_utwor
    FROM (
        SELECT id_utwor, COUNT(*) AS liczba_odtworzen
        FROM HistoriaOdtworzen
        WHERE TRUNC(data_odtworzenia) = data
        GROUP BY id_utwor
        ORDER BY liczba_odtworzen DESC
    )
    WHERE ROWNUM = 1;
    RETURN id_utwor;
END NajczesciejOdtwarzanyUtworDnia;
```

```
1  DECLARE
2      id_utworu NUMBER;
3  BEGIN
4      id_utworu := statystykiodtworzen.NajczesciejOdtwarzanyUtworDnia(TO_DATE('04-07-2024', 'MM-DD-YYYY'));
5      DBMS_OUTPUT.PUT_LINE('ID najczęściej odtwarzanego utworu: ' || id_utworu);
6  END;
7
```

Results Explain Describe Saved SQL History

ID najczęściej odtwarzanego utworu: 76

## LacznyCzasOdtwarzaniaUzytkownika

Funkcja oblicza łączny czas odtworzeń dla użytkownika na podstawie danych zawartych w tabeli historiaodtworzen.

Wejście:

`id_uzytkownik` (INTEGER): Identyfikator użytkownika, dla którego ma być obliczony łączny czas odtworzeń.

Wyjście:

Liczba (NUMBER): łączny czas odtworzeń w sekundach (lub innej jednostce, zgodnej z kolumną `czas_odtworzenia` w tabeli).

Działanie funkcji (krok po kroku):

Inicjalizacja: Ustawienie zmiennej `calkowity_czas` na 0.

Debugowanie: Wyświetlenie w logu ID użytkownika, dla którego będzie obliczany czas.

Walidacja danych:

Zliczenie rekordów w tabeli historiaodtworzen dla danego użytkownika.

Wyświetlenie liczby znalezionych rekordów w logu.

Obliczenie łącznego czasu odtworzeń:

Sumowanie wartości `czas_odtworzenia` z tabeli historiaodtworzen, gdzie `id_uzytkownik` odpowiada podanemu ID.

Wynik jest zapisywany w zmiennej `calkowity_czas`.

Debugowanie: Wyświetlenie w logu obliczonego łącznego czasu odtworzeń.

Zwrócenie wyniku: Zwrócenie wartości `calkowity_czas`.

Obsługa wyjątków:

`NO_DATA_FOUND`: Zwraca 0, jeśli brak danych w tabeli dla podanego użytkownika.

`OTHERS`: Obsługuje inne błędy, podając komunikat o błędzie i zgłasza błąd aplikacyjny z kodem -20001.

```
FUNCTION LacznyCzasOdtwarzaniaUzytkownika(id_uzytkownik INT)
RETURN NUMBER IS
    calkowity_czas NUMBER := 0;
BEGIN
    -- Debugging
    DBMS_OUTPUT.PUT_LINE('Processing total playback time for user ID: ' || 
id_uzytkownik);
```

```
DECLARE
    record_count NUMBER := 0;
BEGIN
    SELECT COUNT(*)
    INTO record_count
    FROM historiaodtworzen
    WHERE id_uzytkownik = LacznyCzasOdtwarzaniaUzytkownika.id_uzytkownik;
```

```

-- Debugging
DBMS_OUTPUT.PUT_LINE('Number of records found for user ID ' || id_uzytownik
|| ':' || record_count);
END;

SELECT NVL(SUM(czas_odtworzenia), 0)
INTO calkowity_czas
FROM historiaodtworzen
WHERE id_uzytownik = LacznyCzasOdtwarzaniaUzytkownika.id_uzytownik;

-- Debugging
DBMS_OUTPUT.PUT_LINE('Total playback time for user ID ' || id_uzytownik || ':' ||
calkowity_czas);

RETURN calkowity_czas;

EXCEPTION
WHEN NO_DATA_FOUND THEN
-- Debugging
DBMS_OUTPUT.PUT_LINE('Nie znaleziono danych dla użytkownika o ID: ' ||
id_uzytownik);
RETURN 0;
WHEN OTHERS THEN
-- Debugging
DBMS_OUTPUT.PUT_LINE('An unexpected error occurred for user ID: ' ||
id_uzytownik);
RAISE_APPLICATION_ERROR(-20001, 'An error occurred while calculating the
playback time.');
END LacznyCzasOdtwarzaniaUzytkownika;

```

```

1  DECLARE
2    laczny_czas NUMBER;
3  BEGIN
4    laczny_czas := statystykiodtworzen.LacznyCzasOdtwarzaniaUzytkownika(13);
5    DBMS_OUTPUT.PUT_LINE('Łączny czas odtworzeń użytkownika: ' || laczny_czas || ' sekund');
6  END;
7

```

**Results**   Explain   Describe   Saved SQL   History

```

Processing total playback time for user ID: 13
Number of records found for user ID 13: 44
Total playback time for user ID 13: 4331

```

## **ZnajdzNajczesciejOdtwarzanyUtworUzytkownika**

Funkcja wyszukuje tytuł utworu, który był najczęściej odtwarzany przez konkretnego użytkownika.

Wejście:

**id\_uzytkownik** (NUMBER): Identyfikator użytkownika, dla którego należy znaleźć najczęściej odtwarzany utwór.

Wyjście:

**VARCHAR2**: Nazwa najczęściej odtwarzanego utworu.

Działanie funkcji (krok po kroku):

Krok 1:

Pobierz ID\_UTWORU z tabeli HISTORIAODTWRZEN, gdzie ID użytkownika (ID\_UZYTKOWNIK) jest zgodne z parametrem funkcji.

Rekordy są grupowane według ID\_UTWORU.

Policz liczbę odtworzeń (COUNT(\*)).

Posortuj je malejąco według liczby odtworzeń.

Wybierz tylko pierwszy wynik (ROWNUM = 1), co oznacza utwór z największą liczbą odtworzeń.

Wynik zostanie zapisany do zmiennej utwor\_id.

Krok 2:

Na podstawie utwor\_id, pobierz nazwę utworu (Nazwa) z tabeli UTWORY i zapisz do zmiennej tytul\_utworu.

Krok 3:

Zwróć wartość zmiennej tytul\_utworu jako wynik funkcji.

Obsługa wyjątków:

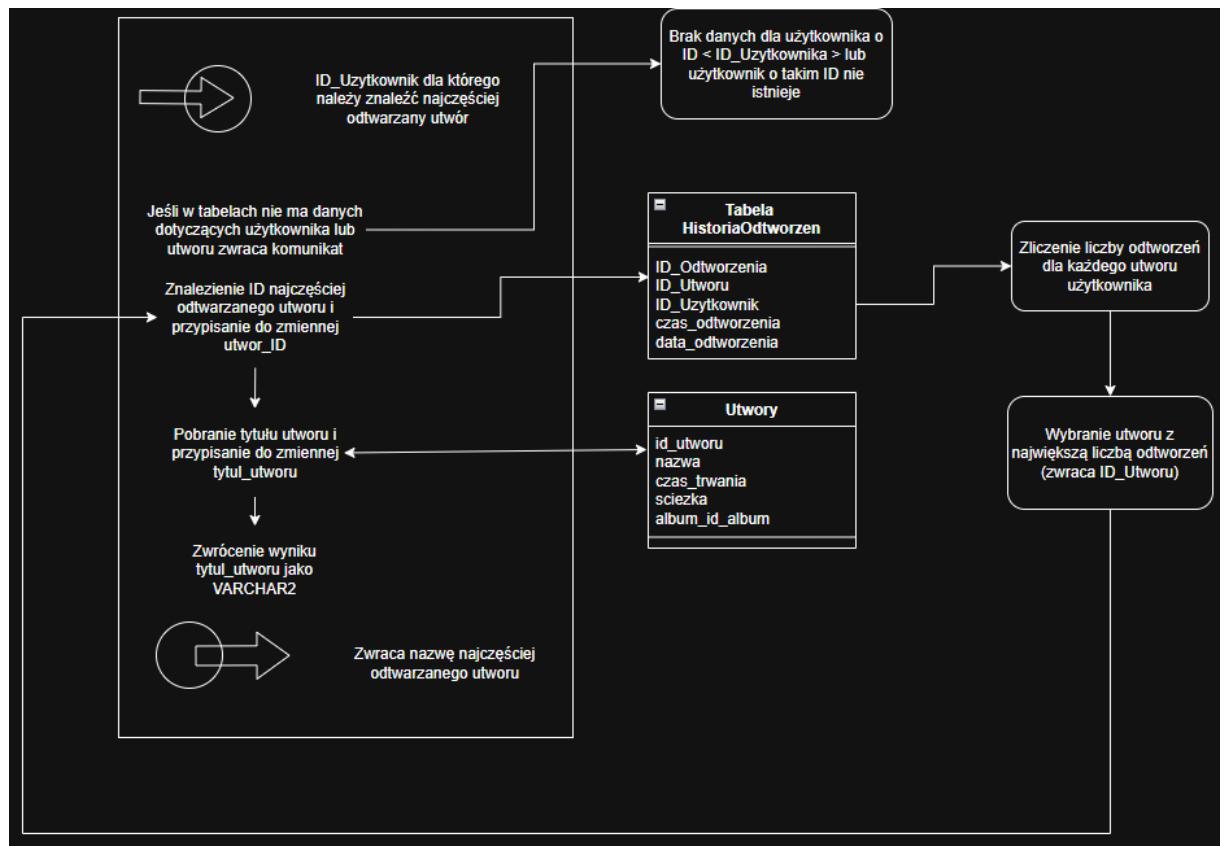
**NO\_DATA\_FOUND**:

Jeśli nie znaleziono danych w tabeli HISTORIAODTWRZEN lub UTWORY, funkcja zwraca

Brak danych.

**OTHERS**:

Obsługuje inne nieoczekiwane błędy i zwraca komunikat Błąd.



```

FUNCTION ZnajdzNajczesciejOdtwarzanyUtworUzytkownika (
    id_uzytkownik IN NUMBER
) RETURN VARCHAR2 IS
    utwor_id NUMBER;
    tytul_utworu VARCHAR2(100);
BEGIN
    -- Pobierz ID utworu, który był najczęściej odtwarzany przez użytkownika
    SELECT ID_UTWORU
    INTO utwor_id
    FROM (
        SELECT ID_UTWORU, COUNT(*) AS LICZBA_ODTWORZEN
        FROM HISTORIAODTWORZEN
        WHERE ID_UZYTKOWNIK =
            znajdznajczesciejodtwarzanyutworuzytkownika.id_uzytkownik
        GROUP BY ID_UTWORU
        ORDER BY LICZBA_ODTWORZEN DESC
    )
    WHERE ROWNUM = 1;

    -- Pobierz tytuł utworu na podstawie ID
    SELECT Nazwa
    INTO tytul_utworu
    FROM Utwory
    WHERE ID_UTWORU = utwor_id;

```

```

-- Zwróć tytuł utworu
RETURN tytul_utworu;

EXCEPTION
WHEN NO_DATA_FOUND THEN
    -- Jeśli brak danych, zwróć domyślną wartość
    RETURN 'Brak danych';
WHEN OTHERS THEN
    -- Jeśli wystąpił inny błąd, zwróć odpowiedni komunikat
    RETURN 'Błąd';
END NajdzNajczesciejOdtwarzanyUtworUzytkownika;

```

The screenshot shows a PL/SQL block in the SQL worksheet:

```

1  DECLARE
2      najczesciej_utwor VARCHAR2(100);
3  BEGIN
4      najczesciej_utwor := statystykiotworzen.ZnajdzNajczesciejOdtwarzanyUtworUzytkownika(13);
5      DBMS_OUTPUT.PUT_LINE('Najczęściej odtwarzany utwór użytkownika: ' || najczesciej_utwor);
6  END;
7

```

The results tab is selected, displaying the output of the DBMS\_OUTPUT.PUT\_LINE statement:

Najczęściej odtwarzany utwór użytkownika: odkąd nauczyłem się przeklinać

## SprawdzStatusSubskrypcji

Deklaracja zmiennej v\_status:

Typ VARCHAR2(20) przechowuje status subskrypcji.

Będzie użyta do zapisania wyniku zapytania SQL.

Blok BEGIN:

Zapytanie SQL wyszukuje status subskrypcji użytkownika w tabeli

UZYTKOWNIK\_PLAN\_SUBSKRYPCYJNY, pod warunkiem, że:

UZYTKOWNIK\_ID\_UZYTKOWNIK odpowiada podanemu identyfikatorowi użytkownika.

DATA\_KONIEC subskrypcji jest późniejsza niż bieżąca data (SYSDATE).

Wynik zapytania (kolumna STATUS) jest przypisywany do zmiennej v\_status.

Zwracanie statusu:

Jeśli zapytanie SQL zwróci wynik, funkcja zwraca zawartość v\_status, czyli status subskrypcji (np. "AKTYWNA", "ANULOWANA").

Blok EXCEPTION:

WHEN NO\_DATA\_FOUND:

Obsługuje sytuację, gdy użytkownik nie ma aktywnej subskrypcji. W takim przypadku funkcja zwraca wartość "BRAK SUBSKRYPCJI".

Domyślona obsługa błędów (WHEN OTHERS):

Można dodać dodatkową obsługę błędów, np. logowanie błędu.

```
FUNCTION SprawdzStatusSubskrypcji(id_uzytkownik NUMBER) RETURN VARCHAR2 IS
    v_status VARCHAR2(20);
BEGIN
    SELECT STATUS
    INTO v_status
    FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = id_uzytkownik
        AND DATA_KONIEC > SYSDATE;
    RETURN v_status;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 'BRAK SUBSKRYPCJI';
END SprawdzStatusSubskrypcji;
```

The screenshot shows the execution of the function in Oracle SQL Developer. The code is as follows:

```
1  SELECT zarzadzanieplatnosciami.sprawdzStatusSubskrypcji(11) AS statusSubskrypcji
2  FROM DUAL;
```

The results pane shows the output:

statusSubskrypcji
BRAK SUBSKRYPCJI

## PobierzCzasTrwaniaSubskrypcji

Funkcja ma na celu obliczenie i zwrócenie liczby dni trwania aktywnej subskrypcji użytkownika na podstawie jego identyfikatora. Jeśli użytkownik nie posiada aktywnej subskrypcji, funkcja zwraca 0.

Deklaracja zmiennej v\_czas\_trwania:

Typ NUMBER przechowuje wynik zapytania SQL, czyli różnicę między datą końca (DATA\_KONIEC) a datą startu (DATA\_START) subskrypcji.

Blok BEGIN:

Zapytanie SQL pobiera liczbę dni trwania subskrypcji (DATA\_KONIEC - DATA\_START) z tabeli UZYTKOWNIK\_PLAN\_SUBSKRYPCYJNY, jeśli:

UZYTKOWNIK\_ID\_UZYTKOWNIK odpowiada podanemu identyfikatorowi użytkownika. Subskrypcja jest aktywna, tzn. jej DATA\_KONIEC jest późniejsza niż bieżąca data (SYSDATE).

Wynik zapytania jest przypisywany do zmiennej v\_czas\_trwania.

Zwracanie liczby dni:

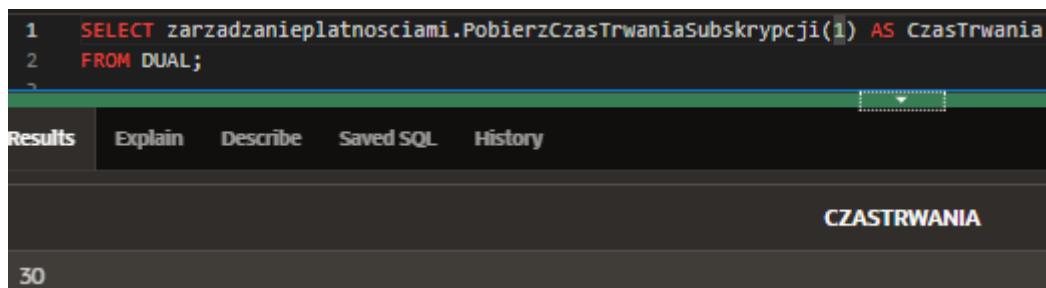
Jeśli zapytanie SQL zwróci wynik, funkcja zwraca liczbę dni (v\_czas\_trwania).

Blok EXCEPTION:

WHEN NO\_DATA\_FOUND:

Obsługuje sytuację, gdy użytkownik nie posiada aktywnej subskrypcji. Funkcja zwraca 0 jako wartość domyślną.

```
FUNCTION PobierzCzasTrwaniaSubskrypcji(uzytkownik_id NUMBER) RETURN NUMBER IS
    v_czas_trwania NUMBER;
BEGIN
    SELECT (DATA_KONIEC - DATA_START)
    INTO v_czas_trwania
    FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id
        AND DATA_KONIEC > SYSDATE;
    RETURN v_czas_trwania;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 0;
END PobierzCzasTrwaniaSubskrypcji;
```



The screenshot shows the Oracle SQL Developer interface. A query window is open with the following code:

```
1  SELECT zarzadzanieplatnosciami.PobierzCzasTrwaniaSubskrypcji(1) AS CzasTrwania
2  FROM DUAL;
```

The results tab is selected, showing the output:

CzasTrwania
30

## ObliczKwoteZaOkres

Funkcja ma na celu obliczenie kwoty, jaką użytkownik musi zapłacić za subskrypcję w określonym przedziale czasowym. Kwota jest proporcjonalna do liczby dni w podanym okresie.

Deklaracja zmiennych lokalnych:

v\_plan\_wartosc (typ NUMBER): Przechowuje miesięczny koszt planu subskrypcyjnego.  
v\_czas\_trwania (typ NUMBER): Przechowuje liczbę dni między datą początkową (start\_date) a końcową (end\_date).

Blok BEGIN:

Pobranie wartości planu subskrypcyjnego:

Funkcja wykonuje zapytanie SQL, aby pobrać koszt miesięczny planu subskrypcyjnego o identyfikatorze plan\_id z tabeli PLAN\_SUBSKRYPCYJNY.

Obliczenie liczby dni:

Funkcja oblicza liczbę dni w podanym okresie jako różnicę między datą końcową (end\_date) a datą początkową (start\_date).

Obliczenie kwoty:

Funkcja oblicza proporcjonalny koszt subskrypcji za liczbę dni w podanym okresie.

Zakłada, że każdy miesiąc ma 30 dni (przelicznik miesięczny).

Zwracanie wyniku:

Funkcja zwraca obliczoną kwotę, która jest iloczynem kosztu planu miesięcznego, liczby dni i proporcji miesięcznej (/30).

```
FUNCTION ObliczKwoteZaOkres(start_date DATE, end_date DATE, plan_id NUMBER)
RETURN NUMBER IS
    v_plan_wartosc NUMBER;
    v_czas_trwania NUMBER;
BEGIN
    SELECT WARTOSC
    INTO v_plan_wartosc
    FROM PLAN_SUBSKRYPCYJNY
    WHERE ID_PLAN_SUBSKRYPCYJNY = plan_id;
    v_czas_trwania := end_date - start_date;
    RETURN v_plan_wartosc * v_czas_trwania / 30; -- Przelicznik miesięczny
END ObliczKwoteZaOkres;
```

```
1  SELECT zarzadzanieplatnosciami.ObliczKwoteZaOkres(
2      TO_DATE('01-15-2025', 'MM-DD-YYYY'),
3      TO_DATE('02-14-2025', 'MM-DD-YYYY'),
4      1
5  ) AS Kwota
6  FROM DUAL;
7
```

Results	Explain	Describe	Saved SQL	History
5				

## CzyPlatnoscZrealizowana

Funkcja sprawdza, czy użytkownik dokonał płatności w dniu bieżącym (dzień, miesiąc, rok muszą być identyczne).

Parametry wejściowe:

uzytkownik\_id (NUMBER): ID użytkownika, dla którego chcemy sprawdzić, czy zrealizowano płatność.

Zwracana wartość:

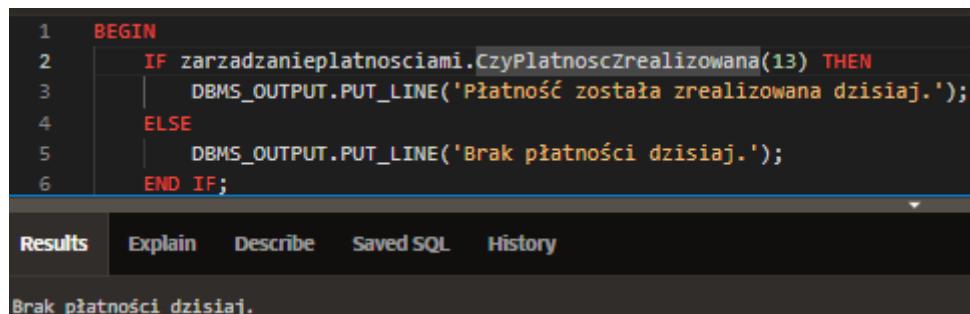
BOOLEAN:

TRUE: Jeśli użytkownik dokonał przynajmniej jednej płatności w dniu bieżącym.

FALSE: Jeśli nie znaleziono płatności dla użytkownika w dniu bieżącym.

```
FUNCTION CzyPlatnoscZrealizowana(uzytkownik_id NUMBER) RETURN BOOLEAN IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM PLATNOSC
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id
        AND TRUNC(DATA_PLATNOSC) = TRUNC(SYSDATE);

    RETURN v_count > 0;
END CzyPlatnoscZrealizowana;
```



```
1 BEGIN
2     IF zarzadzanieplatnosciami.CzyPlatnoscZrealizowana(13) THEN
3         DBMS_OUTPUT.PUT_LINE('Płatność została zrealizowana dzisiaj.');
4     ELSE
5         DBMS_OUTPUT.PUT_LINE('Brak płatności dzisiaj.');
6     END IF;
```

Results Explain Describe Saved SQL History

Brak płatności dzisiaj.

## IleDniDoKonca

Funkcja oblicza i zwraca liczbę dni pozostałych do końca subskrypcji użytkownika.

Parametry wejściowe:

uzytkownik\_id (NUMBER): ID użytkownika, dla którego sprawdzana jest liczba dni do końca subskrypcji.

Zwracana wartość:

NUMBER: Liczba dni pozostałych do końca subskrypcji użytkownika.

NULL: Jeśli nie znaleziono aktywnej subskrypcji dla użytkownika.

```
FUNCTION IleDniDoKonca(uzytkownik_id NUMBER) RETURN NUMBER IS
    v_dni NUMBER;
BEGIN
    SELECT DATA_KONIEC - TRUNC(SYSDATE)
    INTO v_dni
    FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id
        AND DATA_KONIEC > SYSDATE;
    RETURN v_dni;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL;
END IleDniDoKonca;
```

```
1  DECLARE
2      liczba_dni NUMBER;
3  BEGIN
4      liczba_dni := zarzadzanieplatnosciami.IleDniDoKonca(1); -- ID użytkownika
5      IF liczba_dni IS NOT NULL THEN
6          DBMS_OUTPUT.PUT_LINE('Liczba dni do końca subskrypcji: ' || liczba_dni);
7      ELSE
8          DBMS_OUTPUT.PUT_LINE('Brak aktywnej subskrypcji.');
9      END IF;
10 END;
```

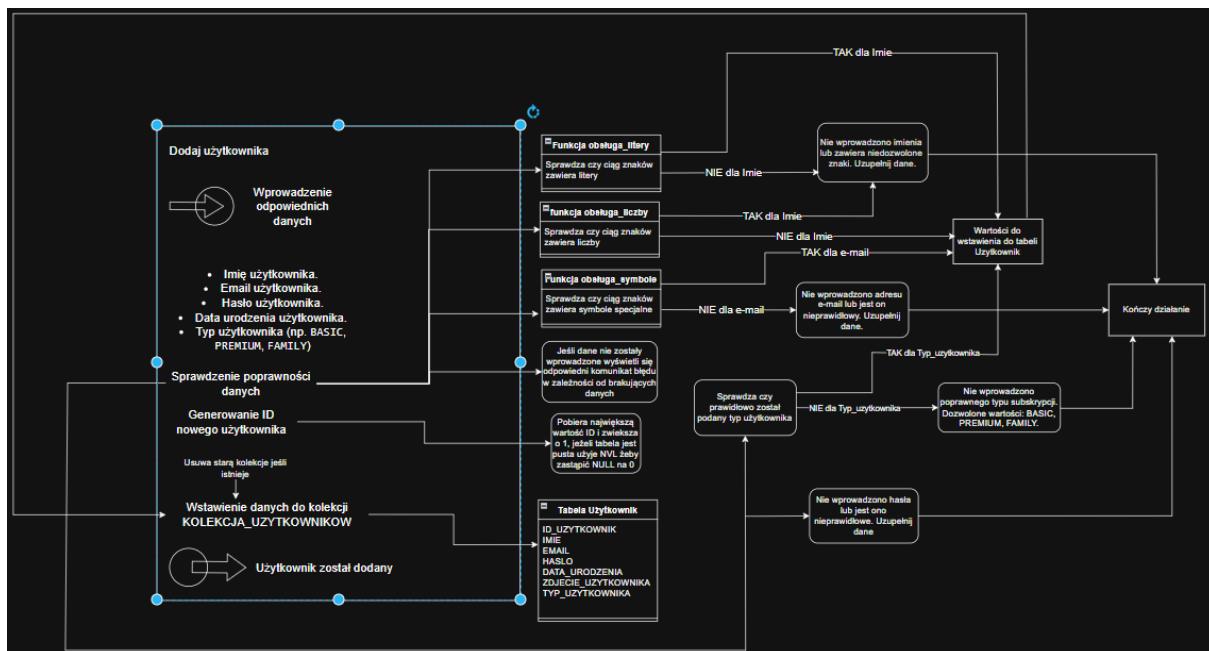
Results Explain Describe Saved SQL History

Liczba dni do końca subskrypcji: 8

## 4. Procedury

### DodajUzytkownika

Procedura DodajUzytkownika umożliwia dodanie nowego użytkownika do systemu. Dane użytkownika są najpierw walidowane, a następnie zapisywane w kolekcji APEX o nazwie KOLEKCJA\_UZYTKOWNIKOW.



### PROCEDURE DodajUzytkownika(

```

imieU VARCHAR2,
emailU VARCHAR2,
hasloU VARCHAR2,
data_urodzeniaU DATE,
typ_uzytownikaU VARCHAR2
)
```

) IS

```

brak_imienia EXCEPTION;
brak_emailu EXCEPTION;
brak_hasla EXCEPTION;
brak_typu_uzytownika EXCEPTION;
max_id_uzytownika NUMBER;
```

BEGIN

-- Tworzenie nowej kolekcji, jeśli jeszcze nie istnieje

```

IF NOT APEX_COLLECTION.COLLECTION_EXISTS('KOLEKCJA_UZYTKOWNIKOW') THEN
    APEX_COLLECTION.CREATE_COLLECTION('KOLEKCJA_UZYTKOWNIKOW');
```

```
END IF;
```

```
-- Walidacja danych wejściowych
```

```
IF imieU IS NULL OR obsluga_liczby(imieU) THEN
```

```
    RAISE brak_imienia;
```

```
ELSIF emailU IS NULL OR NOT obsluga_symbole(emailU) THEN
```

```
    RAISE brak_emailu;
```

```
ELSIF hasloU IS NULL THEN
```

```
    RAISE brak_hasla;
```

```
ELSIF typ_uzytkownikaU NOT IN ('REGULAR', 'PREMIUM', 'FAMILY') THEN
```

```
    RAISE brak_typu_uzytkownika;
```

```
ELSE
```

```
-- Generowanie nowego ID użytkownika
```

```
SELECT NVL(MAX(id_uzytkownik), 0) + 1 INTO max_id_uzytkownika
```

```
FROM Uzytkownik;
```

```
-- Dodanie danych użytkownika do kolekcji APEX
```

```
APEX_COLLECTION.ADD_MEMBER(
```

```
    p_collection_name => 'KOLEKCJA_UZYTKOWNIKOW',
```

```
    p_c001 => max_id_uzytkownika,
```

```
    p_c002 => imieU,
```

```
    p_c003 => emailU,
```

```
    p_c004 => hasloU,
```

```
    p_c005 => TO_CHAR(data_urodzeniaU, 'YYYY-MM-DD'),
```

```
    p_c006 => typ_uzytkownikaU
```

```
);
```

```
-- Komunikat do logów
```

```
DBMS_OUTPUT.PUT_LINE('Dane użytkownika zapisano do kolekcji  
KOLEKCJA_UZYTKOWNIKOW.');
```

```
END IF;
```

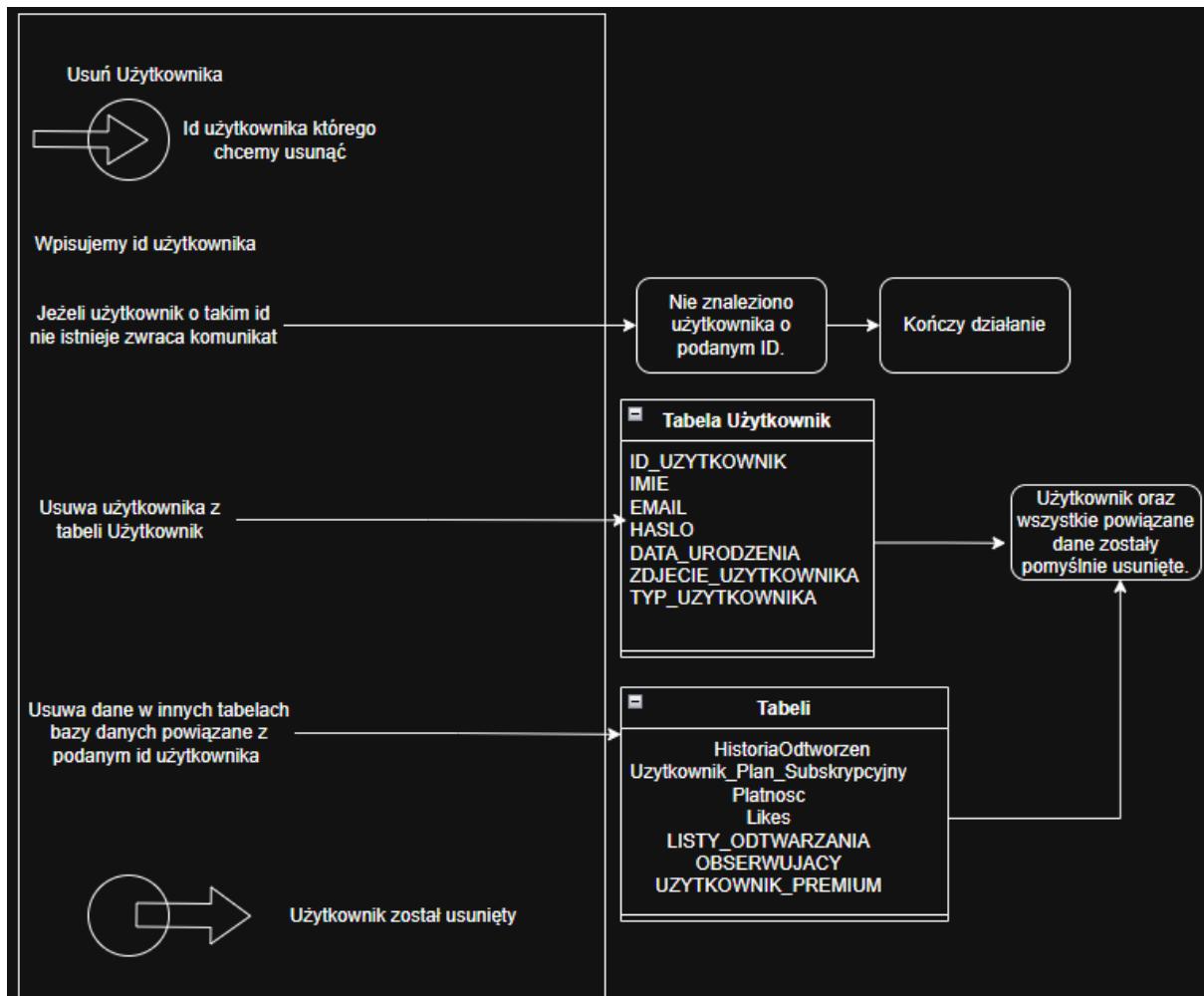
## EXCEPTION

```
WHEN brak_imienia THEN
    DBMS_OUTPUT.PUT_LINE('Nie wprowadzono imienia lub zawiera niedozwolone znaki. Uzupełnij dane.');
WHEN brak_emailu THEN
    DBMS_OUTPUT.PUT_LINE('Nie wprowadzono adresu e-mail lub jest on nieprawidłowy. Uzupełnij dane.');
WHEN brak_hasla THEN
    DBMS_OUTPUT.PUT_LINE('Nie wprowadzono hasła lub jest ono nieprawidłowe. Uzupełnij dane.');
WHEN brak_typy_uzytownika THEN
    DBMS_OUTPUT.PUT_LINE('Nie wprowadzono poprawnego typu subskrypcji. Dozwolone wartości: REGULAR, PREMIUM, FAMILY.');
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Wystąpił nieoczekiwany błąd podczas dodawania użytkownika.');
END DodajUzytkownika;
```

```
Language SQL | Rows 10 | Clear Command Find Tables
1 BEGIN
2     dodajUzytkownika;
3     :ime => 'Anna',
4     :email => 'anna.kowalska@example.com',
5     :haslo => '#BezpieczenHaslo@03',
6     :data_urodzenia => TO_DATE('1985-06-25', 'YYYY-MM-DD'),
7     :typ_uzytownika => 'PREMIUM'
8 );
9 END;
10 /
11 -- Sprawdzenie zawartości kolekcji w APEX
12 SELECT c001 AS ID_Uzytkownika,
13        c002 AS Imie,
14        c003 AS Email,
15        c004 AS Haslo,
16        c005 AS Data_Urodzenia,
17        c006 AS Typ_Uzytkownika
18   FROM APEX_COLLECTIONS
19 WHERE Collection_name = 'KOLEKCJA_UZYTKOWNIKOW';
Results Explain Describe Saved SQL History
ID_UZYTKOWNIKA | IMIE | EMAIL | HASLO | DATA_URODZENIA | TYP_UZYTKOWNIKA
30 | Anna | anna.kowalska@example.com | BezpieczenHaslo@03 | 1985-06-25 | PREMIUM
```

## UsunUzytkownika

Procedura UsunUzytkownika służy do całkowitego usunięcia użytkownika z systemu, łącznie z jego danymi powiązanymi w innych tabelach.



PROCEDURE UsunUzytkownika(

```

    id_uzytkowniku IN INTEGER
) IS
BEGIN
    -- Usunięcie powiązanych danych z tabel HistoriaOdtworzen
    DELETE FROM HistoriaOdtworzen
    WHERE id_uzytkownik = id_uzytkowniku;

    -- Usunięcie powiązanych danych z tabel Subskrypcje
    DELETE FROM Uzytkownik_Plan_Subskrypcyjny
    WHERE UZYTKOWNIK_id_uzytkownik = id_uzytkowniku;

```

```

-- Usunięcie powiązanych danych z tabel Platnosci

DELETE FROM Platnosc

WHERE UZYTKOWNIK_id_uzytkownik = id_uzytkownikU;

DELETE FROM Likes

where uzytkownik_id_uzytkownik = id_uzytkownikU;

DELETE FROM LISTY_ODTWARZANIA

WHERE UZYTKOWNIK_ID_UZYTKOWNIK = ID_uzytkownikU;

DELETE FROM OBSERWUJACY

WHERE UZYTKOWNIK_ID_UZYTKOWNIK =ID_UZYTKOWNIKU;

DELETE FROM UZYTKOWNIK_PREMIUM

WHERE UZYTKOWNIK_ID_UZYTKOWNIK = ID_UZYTKOWNIKU;

-- Usunięcie rekordu użytkownika z tabeli Uzytkownik

DELETE FROM Uzytkownik

WHERE id_uzytkownik = id_uzytkownikU;

-- Informacja o powodzeniu operacji

DBMS_OUTPUT.PUT_LINE('Użytkownik oraz wszystkie powiązane dane zostały pomyślnie
usunięte.');

EXCEPTION

WHEN NO_DATA_FOUND THEN

-- Obsługa sytuacji, gdy użytkownik nie istnieje

DBMS_OUTPUT.PUT_LINE('Nie znaleziono użytkownika o podanym ID.');

WHEN OTHERS THEN

-- Obsługa wszelkich innych błędów

DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas usuwania użytkownika.');

END UsunUzytkownika;

```

```

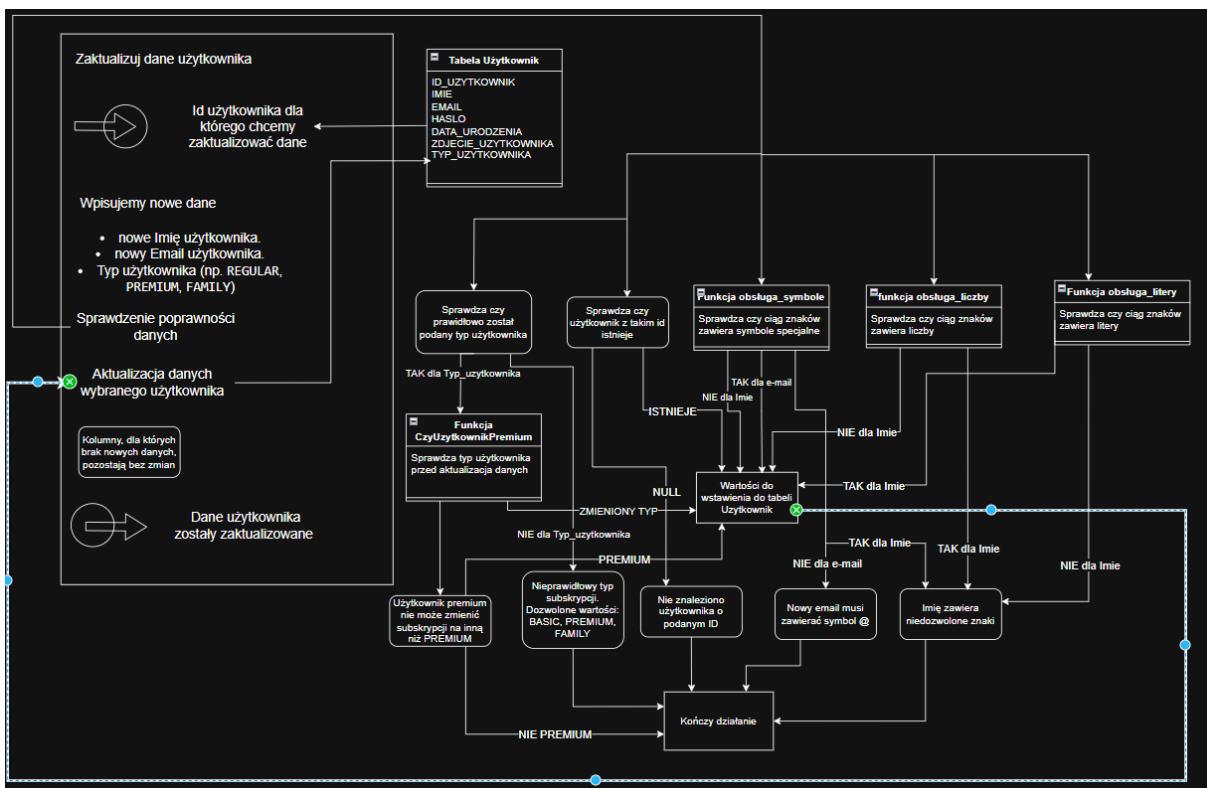
1  BEGIN
2      obslugaUzytkownika.UsunUzytkownika(
3          id_uzytkownika => 22 -- Podaj ID użytkownika do usunięcia
4      );
5  END;
6  /
7  -- Sprawdzenie, czy użytkownik został usunięty
8  SELECT * FROM Uzytkownik WHERE id_uzytkownik = 22;
9
10
11
12

```

Results Explain Describe Saved SQL History  
no data found

### ZaktualizujDaneUzytkownika

Procedura ZaktualizujDaneUzytkownika służy do aktualizacji danych użytkownika w tabeli Uzytkownik. Pozwala na modyfikację imienia, adresu e-mail oraz typu subskrypcji. Dodatkowo weryfikuje poprawność wprowadzonych danych, aby zapobiec wprowadzeniu nieprawidłowych wartości.



PROCEDURE ZaktualizujDaneUzytkownika(

```

        id_uzytkownikaU IN INTEGER,
        nowe_imieU IN VARCHAR2 DEFAULT NULL,
        nowy_emailU IN VARCHAR2 DEFAULT NULL,
        nowy_typ_uzytkownikaU IN VARCHAR2 DEFAULT NULL
    
```

```

) IS

brak_liter EXCEPTION;
brak_symboli EXCEPTION;
brak_liczb EXCEPTION;
niedozwolony_typ_uzytkownika EXCEPTION;
czy_premium BOOLEAN;

BEGIN

czy_premium := CzyUzytkownikPremium(id_uzytkownikU);

IF nowe_imieU IS NOT NULL AND (obsluga_liczby(nowe_imieU) OR
obsluga_symbole(nowe_imieU)) THEN

    RAISE brak_liter;

ELSIF nowy_emailU IS NOT NULL AND NOT obsluga_symbole(nowy_emailU) THEN

    RAISE brak_symboli;

ELSIF nowy_typ_uzytkownikaU IS NOT NULL AND nowy_typ_uzytkownikaU NOT IN ('BASIC',
'PREMIUM', 'FAMILY') THEN

    RAISE niedozwolony_typ_uzytkownika;

END IF;

IF czy_premium AND nowy_typ_uzytkownikaU IS NOT NULL AND nowy_typ_uzytkownikaU != 'PREMIUM' THEN

    DBMS_OUTPUT.PUT_LINE('Uzytkownik premium nie moze zmienic subskrypcji na inną niż PREMIUM.');

    RETURN;

END IF;

UPDATE Uzytkownik

SET imie = NVL(nowe_imieU, imie),
email = NVL(nowy_emailU, email),
typ_uzytkownika = NVL(nowy_typ_uzytkownikaU, typ_uzytkownika)

WHERE id_uzytkownik = id_uzytkownikU;

DBMS_OUTPUT.PUT_LINE('Dane uzytkownika zostaly pomyslnie zaktualizowane.');

```

## EXCEPTION

```

WHEN brak_liter THEN
    DBMS_OUTPUT.PUT_LINE('Imię zawiera niedozwolone znaki.');

WHEN brak_symboli THEN
    DBMS_OUTPUT.PUT_LINE('Nowy email musi zawierać co najmniej jeden symbol (@ lub podobny).');

WHEN niedozwolony_typ_uzytkownika THEN
    DBMS_OUTPUT.PUT_LINE('Nieprawidłowy typ subskrypcji. Dozwolone wartości: BASIC, PREMIUM, FAMILY.');

WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nie znaleziono użytkownika o podanym ID.');

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas aktualizacji danych użytkownika.');

```

END ZaktualizujDaneUzytkownika;

```

1 BEGIN
2     obslugaUzytkownik.ZaktualizujDaneUzytkownika(
3         id_uzytkownika => 22, -- Podaj ID istniejącego użytkownika
4         nowe_imie => 'Piotr',
5         nowy_email => 'piotr.nowak@example.com',
6         nowy_typ_uzytkownika => 'PREMIUM'
7     );
8 END;
9 /
10 -- Sprawdzenie, czy dane użytkownika zostały zaktualizowane
11 SELECT * FROM Uzytkownik WHERE id_uzytkownika = 22;
12

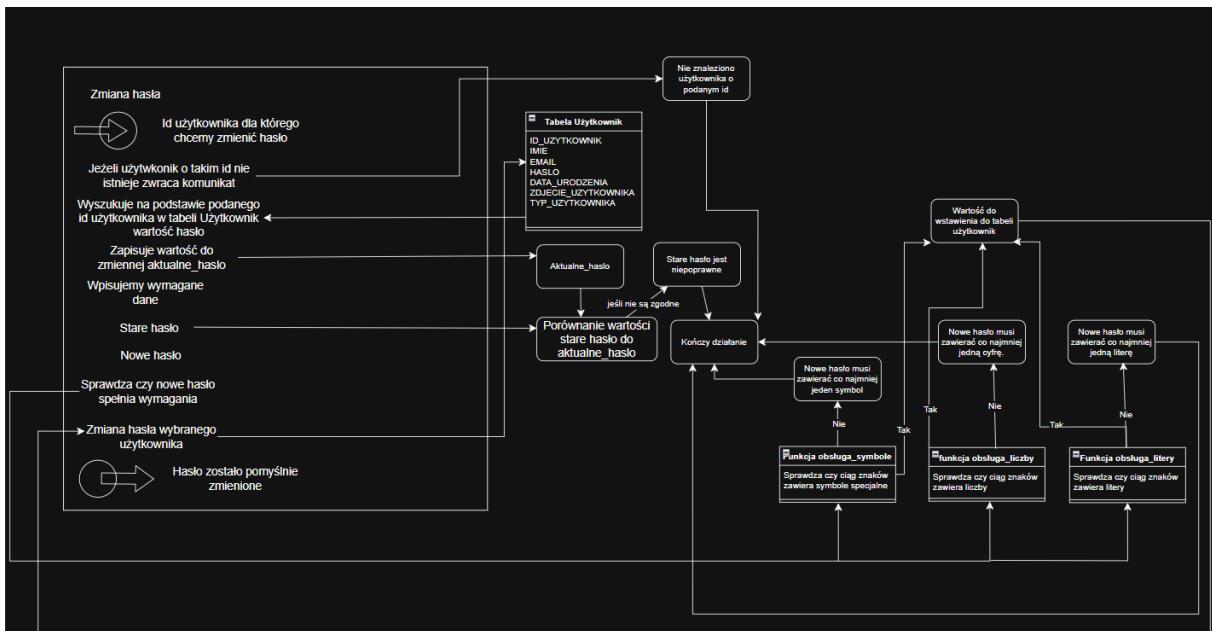
```

Results

ID_UZYTKOWNIK	IMIE	EMAIL	HASLO	DATA_URODZENIA	ZDJECIE_UZYTKOWNIKA
22	Piotr	piotr.nowak@example.com	Andrii123@	12/28/2002	PREMIUM

## ZmienHaslo

Procedura ZmienHaslo służy do zmiany hasła użytkownika w tabeli Uzytkownik. Zawiera mechanizmy walidacji poprawności starego hasła oraz nowego hasła, zapewniając jego zgodność z wymaganiami bezpieczeństwa.



```

PROCEDURE ZmienHaslo (
    id_uzyskownikU IN INTEGER,
    stare_hasloU IN VARCHAR2,
    nowe_hasloU IN VARCHAR2
) IS
    aktualne_hasloU VARCHAR2(100);
    brak_liter EXCEPTION;
    brak_liczb EXCEPTION;
    brak_symboli EXCEPTION;

BEGIN
    -- Pobranie obecnego hasła użytkownika
    SELECT haslo INTO aktualne_hasloU
    FROM Uzytkownik
    WHERE id_uzyskownik = id_uzyskownikU;

    -- Sprawdzenie poprawności starego hasła
    IF aktualne_hasloU != stare_hasloU THEN
        DBMS_OUTPUT.PUT_LINE('Podane stare hasło jest niepoprawne.');
        RETURN;
    END IF;

    -- Sprawdzenie, czy nowe hasło zawiera litery, liczby i symbole
    IF NOT obsluga_liter(nowe_hasloU) THEN
        RAISE brak_liter;
    ELSIF NOT obsluga_liczby(nowe_hasloU) THEN
        RAISE brak_liczb;
    ELSIF NOT obsluga_symbole(nowe_hasloU) THEN
        RAISE brak_symboli;
    END IF;

    -- Aktualizacja hasła

```

```

UPDATE Uzytkownik
SET haslo = nowe_hasloU
WHERE id_uzytkownik = id_uzytkownikU;

DBMS_OUTPUT.PUT_LINE('Hasło zostało pomyślnie zmienione.');

EXCEPTION
  WHEN brak_liter THEN
    DBMS_OUTPUT.PUT_LINE('Nowe hasło musi zawierać co najmniej jedną literę.');
  WHEN brak_liczb THEN
    DBMS_OUTPUT.PUT_LINE('Nowe hasło musi zawierać co najmniej jedną cyfrę.');
  WHEN brak_symboli THEN
    DBMS_OUTPUT.PUT_LINE('Nowe hasło musi zawierać co najmniej jeden symbol.');
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nie znaleziono użytkownika o podanym ID.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas zmiany hasła.');

END ZmienHaslo;

```

```

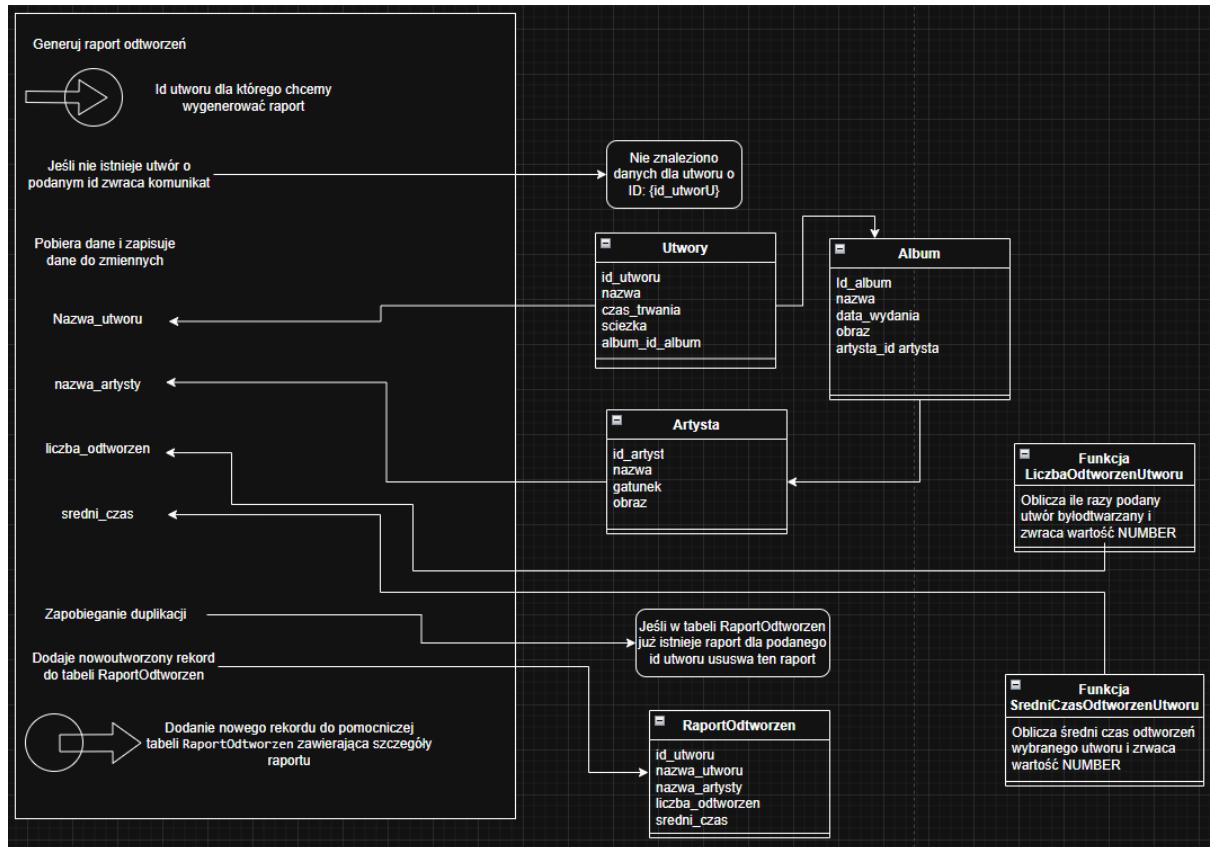
1  BEGIN
2    obslugauzytkownik.zmienHaslo(
3      id_uzytkowniku => 22, -- Podaj ID istniejącego użytkownika
4      stare_hasloU => 'Andrii123@',
5      nowe_hasloU => 'SlabeHaslo2023'
6    );
7  END;
8 /

```

Results	Explain	Describe	Saved SQL	History
<p>Nowe hasło musi zawierać co najmniej jeden symbol.</p> <p>Statement processed.</p>				

## GenerujRaportOdtworzen

Procedura GenerujRaportOdtworzen generuje raport dotyczący odtworzeń danego utworu. Raport zawiera informacje o liczbie odtworzeń, średnim czasie odtworzeń, nazwie utworu oraz nazwie artysty. Wyniki są zapisywane w tabeli RaportOdtworzen.



PROCEDURE GenerujRaportOdtworzen (

    id\_utworU IN INTEGER

) IS

    liczba\_odegrawek NUMBER;

    sredni\_czas NUMBER;

    nazwa\_utworu VARCHAR2(100);

    nazwa\_artysty VARCHAR2(100);

BEGIN

    -- Usuń istniejący raport dla tego samego ID utworu

    DELETE FROM RaportOdtworzen

    WHERE id\_utworu = GenerujRaportOdtworzen.id\_utworU;

    -- Pobierz nazwę utworu i artysty

    SELECT

```
        u.nazwa,  
        a.nazwa  
  
INTO  
  
        nazwa_utworu,  
        nazwa_artysty  
  
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.id_utworu = GenerujRaportOdtworzen.id_utworU;  
  
-- Pobierz liczbę odtworzeń i średni czas  
  
liczba_odtworzen := LiczbaOdtworzenUtworu(id_utworU);  
sredni_czas := SredniCzasOdtworzenUtworu(id_utworU);  
  
-- Wstaw dane do tabeli  
  
INSERT INTO RaportOdtworzen (   
        id_utworu,  
        nazwa_utworu,  
        nazwa_artysty,  
        liczba_odtworzen,  
        sredni_czas  
    ) VALUES (   
        id_utworU,  
        nazwa_utworu,  
        nazwa_artysty,  
        liczba_odtworzen,  
        sredni_czas
```

);

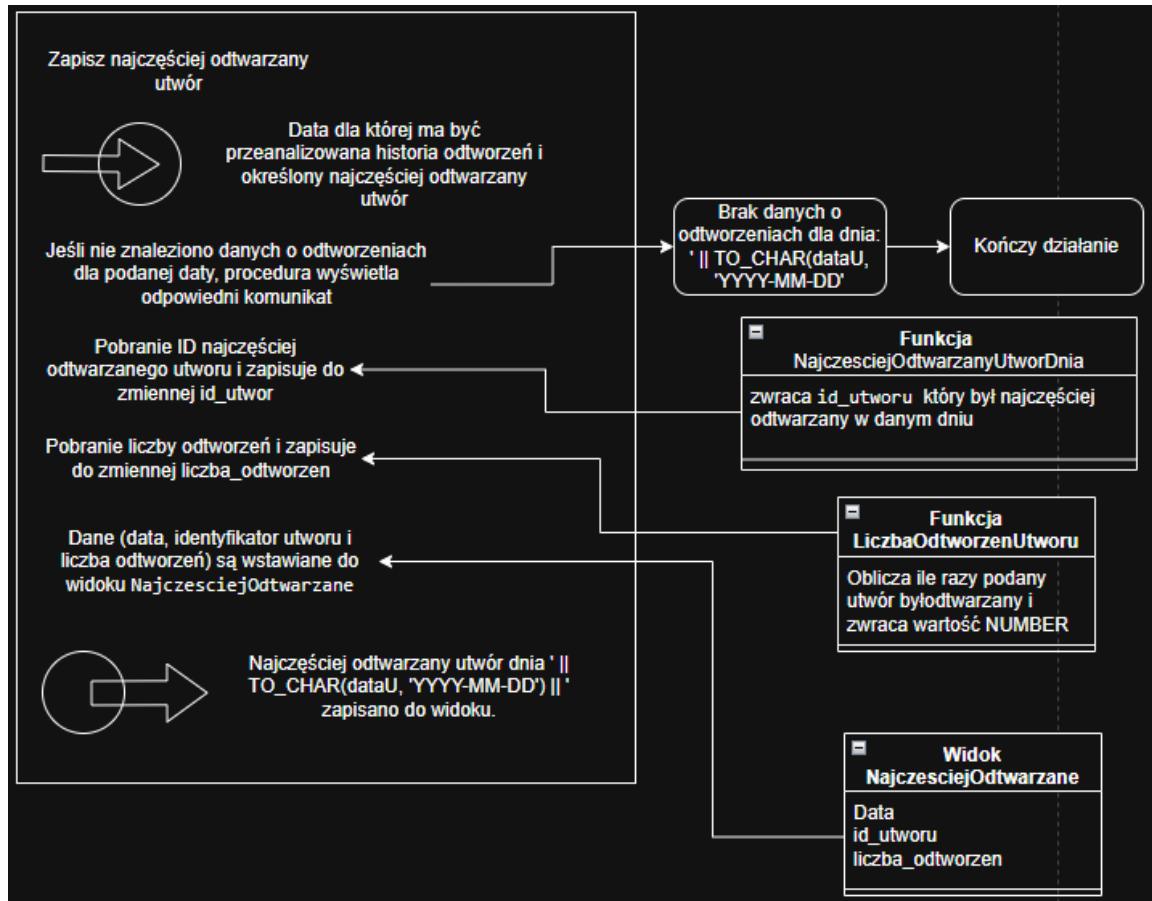
## EXCEPTION

```
WHEN NO_DATA_FOUND THEN  
    DBMS_OUTPUT.PUT_LINE('Nie znaleziono danych dla utworu o ID: ' || id_utworU);  
  
WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE('Wystąpił nieoczekiwany błąd podczas generowania raportu.');//  
    DBMS_OUTPUT.PUT_LINE('Kod błędu: ' || SQLCODE);  
    DBMS_OUTPUT.PUT_LINE('Komunikat błędu: ' || SQLERRM);  
  
END GenerujRaportOdtworzen;
```

```
1 BEGIN
2     statystykiOdtworzen.GenerujReportOdtworzen(1); -- Zamień 1 na ID utworzonego raportu
```

## Zapisz Najczęściej Odtwarzany Utwor

Procedura ZapiszNajczesciejOdtwarzanyUtwor identyfikuje najczęściej odtwarzany utwór dla podanego dnia, oblicza liczbę jego odtworzeń i zapisuje dane do tabeli NajczesciejOdtwarzane.



```

PROCEDURE ZapiszNajczesciejOdtwarzanyUtwor (
    dataU IN DATE
) IS
    id_utwor NUMBER;
    liczba_odtworzen NUMBER;
BEGIN
    -- Pobierz najczęściej odtwarzany utwór dnia
    id_utwor := NajczesciejOdtwarzanyUtworDnia(dataU);

    -- Sprawdź liczbę odtworzeń tego utworu
    liczba_odtworzen := LiczbaOdtworzenUtworu(id_utwor);

    -- Wstaw dane do tabeli NajczesciejOdtwarzane
    INSERT INTO NajczesciejOdtwarzane (
        data,
        id_utworu,
        liczba_odtworzen
    ) VALUES (
        dataU,
        id_utwor,
        liczba_odtworzen
    );

    -- Wyświetl komunikat
    DBMS_OUTPUT.PUT_LINE('Najczęściej odtwarzany utwór dnia ' || TO_CHAR(dataU, 'YYYY-MM-DD') || ' zapisano do widoku.');
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Brak danych o odtworzeniach dla dnia: ' || TO_CHAR(dataU, 'YYYY-MM-DD'));
    WHEN OTHERS THEN

```

```

DBMS_OUTPUT.PUT_LINE('Wystąpił nieoczekiwany błąd podczas zapisywania danych.');

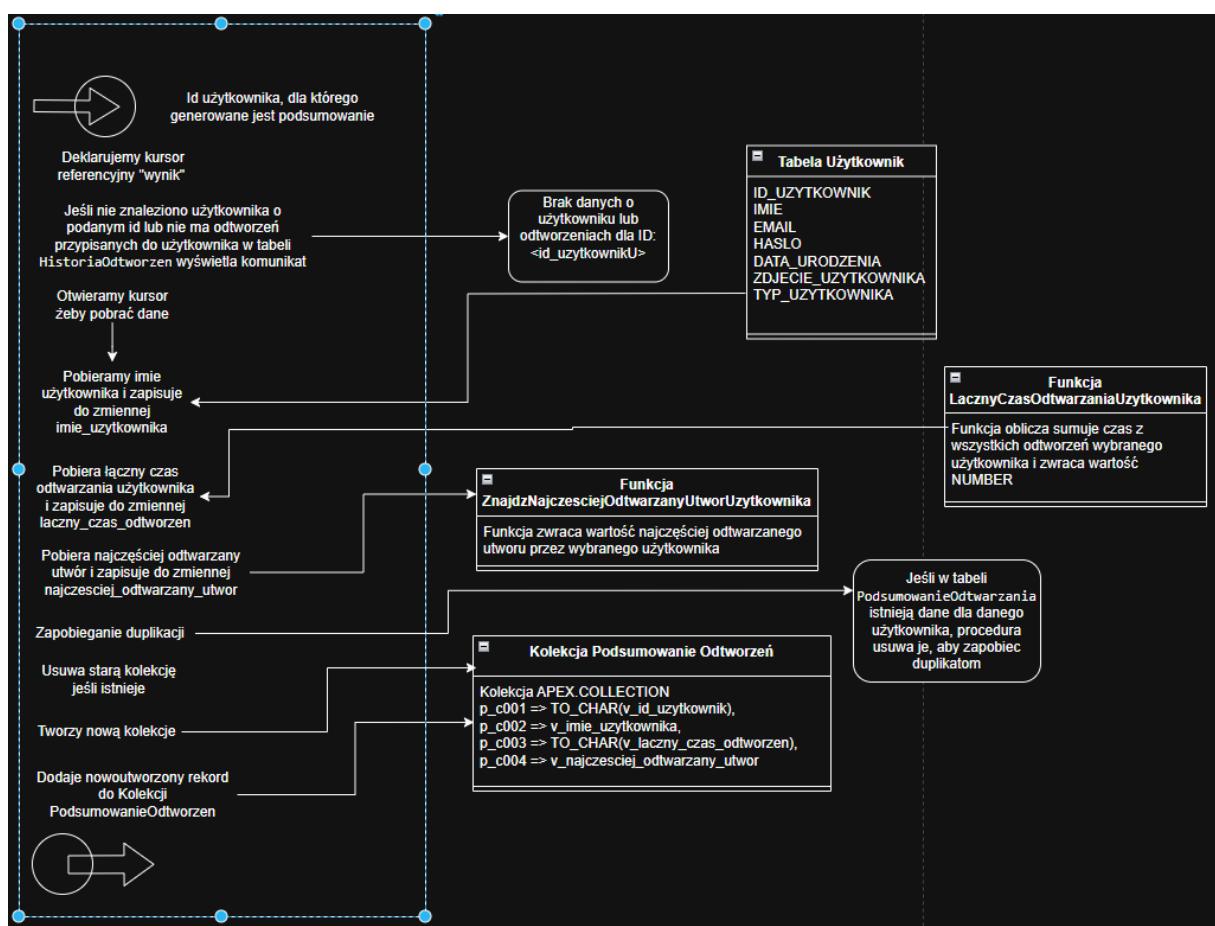
END ZapiszNajczesciejOdtwarzanyUtwor;

```

ZapiszNajczesciejOdtwarzanyUtwor		
Results	Explain	Describe
DATA 1/23/2025	ID_UTWORU 1	LICZBA_ODTWARZEN 1

## PodsumowanieOdtwarzaniaUzytkownika

Procedura PodsumowanieOdtwarzaniaUzytkownika generuje podsumowanie odtwarzania dla danego użytkownika (na podstawie jego ID). Dane są zbierane w kursorze referencyjnym i zapisywane w kolekcji APEX o nazwie PODSUMOWANIE\_ODTWARZANIA.



PROCEDURE PodsumowanieOdtwarzaniaUzytkownika(

```

    id_uzytkownikU IN INTEGER
)
```

-- Deklaracja kursora referencyjnego

wynik SYS\_REFCURSOR;

-- Zmienne pomocnicze do iteracji

```

v_id_uzytkownik NUMBER;
v_imie_uzytkownika VARCHAR2(100);
v_laczny_czas_odtworzen NUMBER;
v_najczesciej_odtwarzany_utwor VARCHAR2(100);

BEGIN
    -- Otwórz kursor referencyjny, aby pobrać dane
    OPEN wynik FOR
        SELECT
            u.id_uzytkownik AS ID_Uzytkownika,
            u.imie AS Imie_Uzytkownika,
            StatystykiOdtworzen.LacznyCzasOdtwarzaniaUzytkownika(u.id_uzytkownik) AS
            Laczny_Czas_Odtworzen,
            StatystykiOdtworzen.ZnajdzNajczesciejOdtwarzanyUtworUzytkownika(u.id_uzytkownik) AS
            Najczesciej_Odtwarzany_Utwor
        FROM
            Uzytkownik u
        WHERE
            u.id_uzytkownik = id_uzytkownikU;

    -- Usuń istniejącą kolekcję (jeśli istnieje)
    IF APEX_COLLECTION.COLLECTION_EXISTS('PODSUMOWANIE_ODTWARZANIA') THEN
        APEX_COLLECTION.DELETE_COLLECTION('PODSUMOWANIE_ODTWARZANIA');
    END IF;

    -- Utwórz nową kolekcję
    APEX_COLLECTION.CREATE_COLLECTION('PODSUMOWANIE_ODTWARZANIA');

    -- Iteracja po kursorze i dodanie danych do kolekcji
    LOOP
        FETCH wynik INTO v_id_uzytkownik, v_imie_uzytkownika, v_laczny_czas_odtworzen,
        v_najczesciej_odtwarzany_utwor;
        EXIT WHEN wynik%NOTFOUND;

```

```

APEX_COLLECTION.ADD_MEMBER(
    p_collection_name => 'PODSUMOWANIE_ODTWARZANIA',
    p_c001 => TO_CHAR(v_id_uzytkownika), -- Kolumna 1: ID użytkownika
    p_c002 => v_imie_uzytkownika, -- Kolumna 2: Imię użytkownika
    p_c003 => TO_CHAR(v_laczny_czas_odtworzen), -- Kolumna 3: Łączny czas odtworzeń
    p_c004 => v_najczesciej_odtworzany_utwor -- Kolumna 4: Najczęściej odtwarzany utwór
);
END LOOP;

-- Zamknięcie kursora
CLOSE wynik;

-- Komunikat do logów
DBMS_OUTPUT.PUT_LINE('Dane dla użytkownika o ID ' || id_uzytkowniku || ' zostały zapisane do kolekcji.');

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Brak danych o użytkowniku lub odtworzeniach dla ID: ' || id_uzytkowniku);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas generowania podsumowania dla użytkownika o ID: ' || id_uzytkowniku);
END PodsumowanieOdtwarzaniaUzytkownika;

```

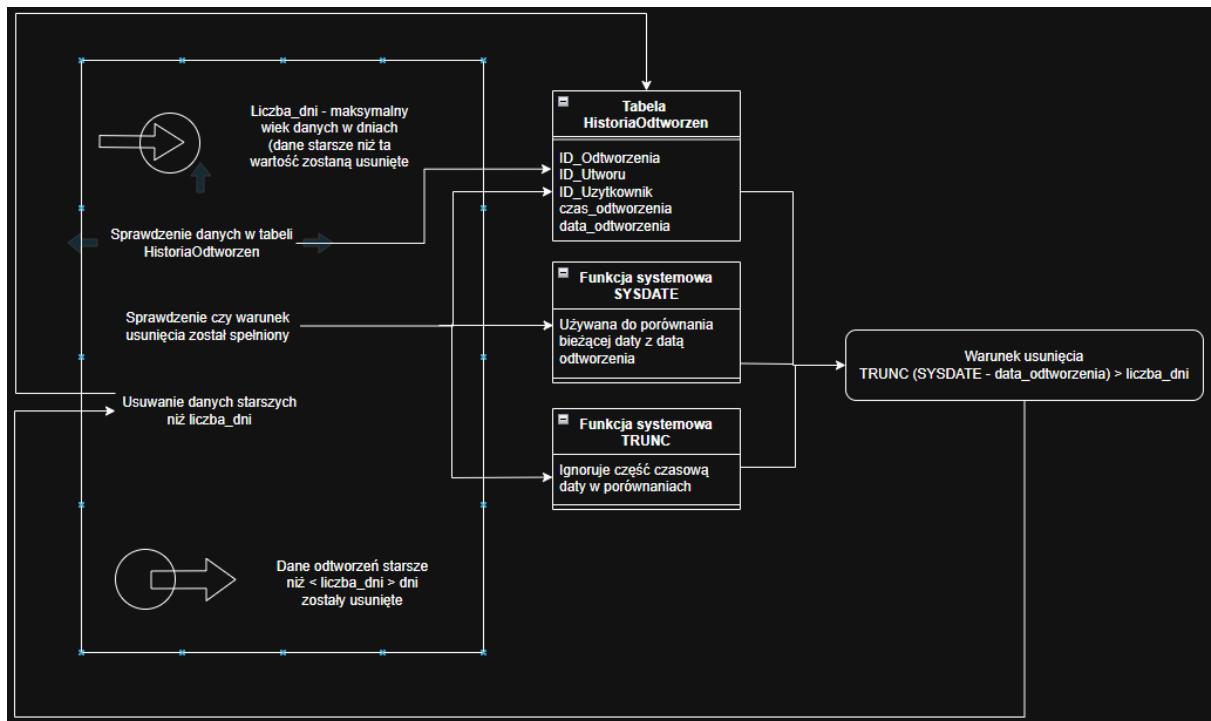
1 -- Wywołanie procedury dla konkretnego ID użytkownika  
 2 BEGIN  
 3 | statystykiodtworzen.PodsumowanieOdtwarzaniaUzytkownika(1); -- Zmień 1 na  
 4 END;  
 5 -- Weryfikacja wyników w kolekcji `PODSUMOWANIE\_ODTWARZANIA` (jeśli dostępna)  
 6 SELECT \*  
 7 FROM APEX\_COLLECTIONS  
 8 WHERE COLLECTION\_NAME = 'PODSUMOWANIE\_ODTWARZANIA';

**Results** Explain Describe Saved SQL History

COLLECTION_NAME	SEQ_ID	C001	C002	C003	C004	C005	C006
PODSUMOWANIE_ODTWARZANIA	1	1	John Doe	5104	miałem kolegę bartka	-	-

## UsunStareDaneOdtworzen

Procedura UsunStareDaneOdtworzen usuwa dane z tabeli HistoriaOdtworzen, które są starsze niż określona liczba dni podana jako parametr wejściowy liczba\_dni.



```

PROCEDURE UsunStareDaneOdtworzen (
    liczba_dni IN INTEGER
) IS
BEGIN
    DELETE FROM HistoriaOdtworzen
    WHERE TRUNC(SYSDATE - data_odtworzenia) > liczba_dni;

    DBMS_OUTPUT.PUT_LINE('Dane odtworzeń starsze niż ' || liczba_dni || ' dni zostały usunięte.');
END UsunStareDaneOdtworzen;

```

```

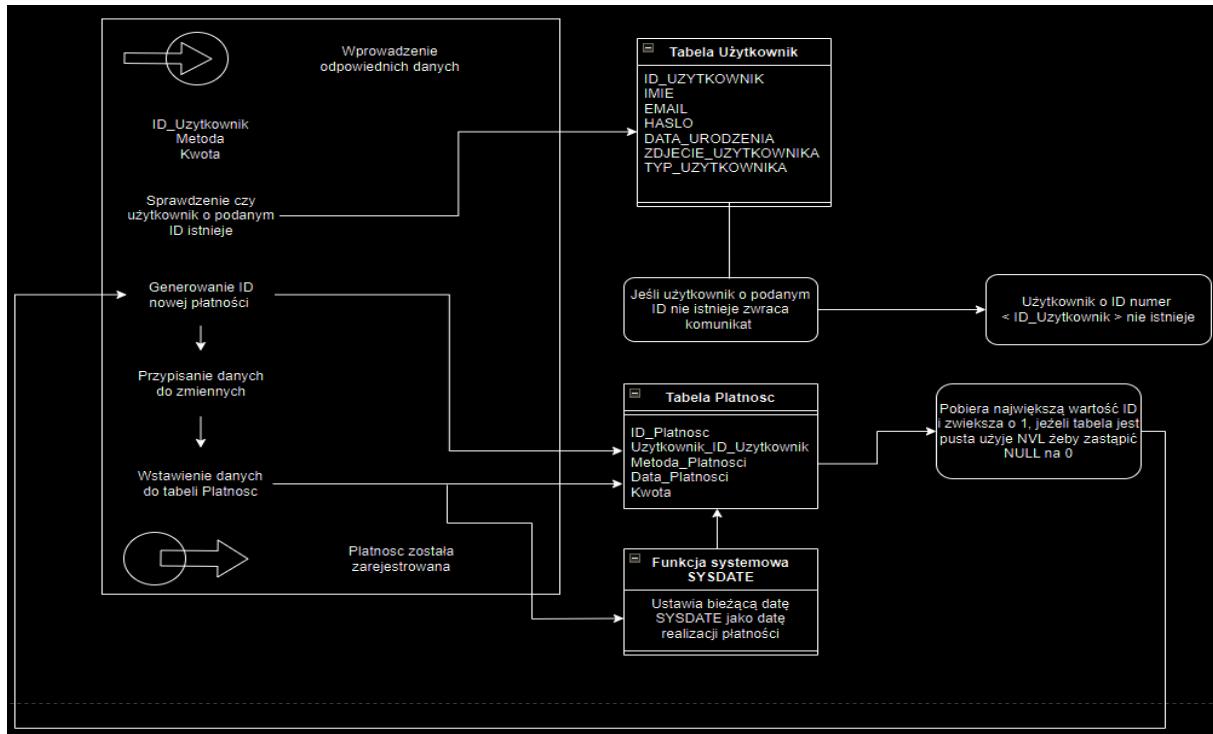
1 BEGIN
2     UsunStareDaneOdtworzen(370);
3 END;
4 SELECT *
5 FROM HistoriaOdtworzen
6 WHERE TRUNC(SYSDATE - data_odtworzenia) > 370;

```

Results	Explain	Describe	Saved SQL	History
no data found				

## ZarejestrujPlatnosc

Procedura ZarejestrujPlatnosc służy do rejestrowania nowych płatności użytkownika w tabeli PLATNOSC. Zapisuje informacje, takie jak identyfikator płatności, identyfikator użytkownika, metoda płatności i kwota. Automatycznie rejestruje także datę płatności jako bieżącą datę systemową (SYSDATE).



PROCEDURE ZarejestrujPlatnosc(

    id\_platnosc NUMBER,

    id\_uzytkownik NUMBER,

    metoda VARCHAR2,

    kwota NUMBER

) IS

BEGIN

    INSERT INTO PLATNOSC (ID\_PLATNOSC, UZYTKOWNIK\_ID\_UZYTKOWNIK, METODA\_PLATNOSCI, DATA\_PLATNOSC, KWOTA)

        VALUES (id\_platnosc, id\_uzytkownik, metoda, SYSDATE, kwota);

END ZarejestrujPlatnosc;

```

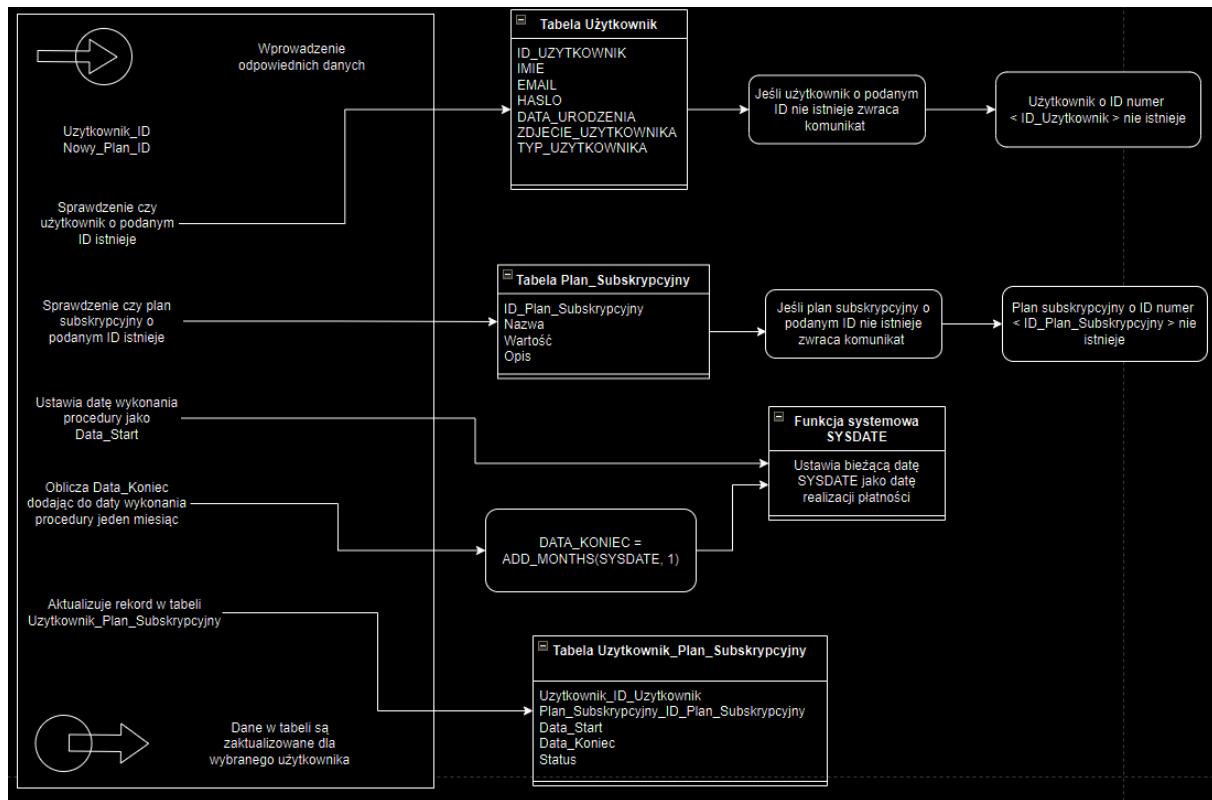
1  BEGIN
2      zarzadzanieplatnosciami.zarejestrujPlatnosc(
3          id_platnosc => 1001,
4          id_uzytkownik => 1,
5          metoda_platnosc => 'ONLINE',
6          kwota => 10
7      );
8  END;
9  SELECT *
10  FROM PLATNOSC
11  ORDER BY DATA_PLATNOSC DESC
12  FETCH FIRST 1 ROWS ONLY;
13

```

Results	Explain	Describe	Saved SQL	History
ID_PLATNOSC	METODA_PLATNOSCI	DATA_PLATNOSC	KWOTA	UZYTKOWNIK_ID_UZYTKOWNIK
1001	ONLINE	1/24/2025	10	1

## ZmienPlanSubskrypcji

Procedura aktualizuje plan subskrypcyjny użytkownika w tabeli UZYTKOWNIK\_PLAN\_SUBSKRYPCYJNY.



PROCEDURE ZmienPlanSubskrypcji(

  uzytkownik\_id NUMBER,

  nowy\_plan\_id NUMBER

) IS

BEGIN

  UPDATE UZYTKOWNIK\_PLAN\_SUBSKRYPCYJNY

    SET PLAN\_SUBSKRYPCYJNY\_ID\_PLAN\_SUBSKRYPCYJNY = nowy\_plan\_id,

    DATA\_START = SYSDATE,

    DATA\_KONIEC = ADD\_MONTHS(SYSDATE, 1),

    STATUS = 'AKTYWNA'

  WHERE UZYTKOWNIK\_ID\_UZYTKOWNIK = uzytkownik\_id;

END ZmienPlanSubskrypcji;

```

1 BEGIN
2   zarzadzaniemplatnosciami.AnulujSubskrypcje(
3     uzytkownik_id => 11,
4     nowy_plan_id => 2
5   );
6 END;
7 SELECT *
8   FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
9 WHERE UZYTKOWNIK_ID_UZYTKOWNIK = 11;
10

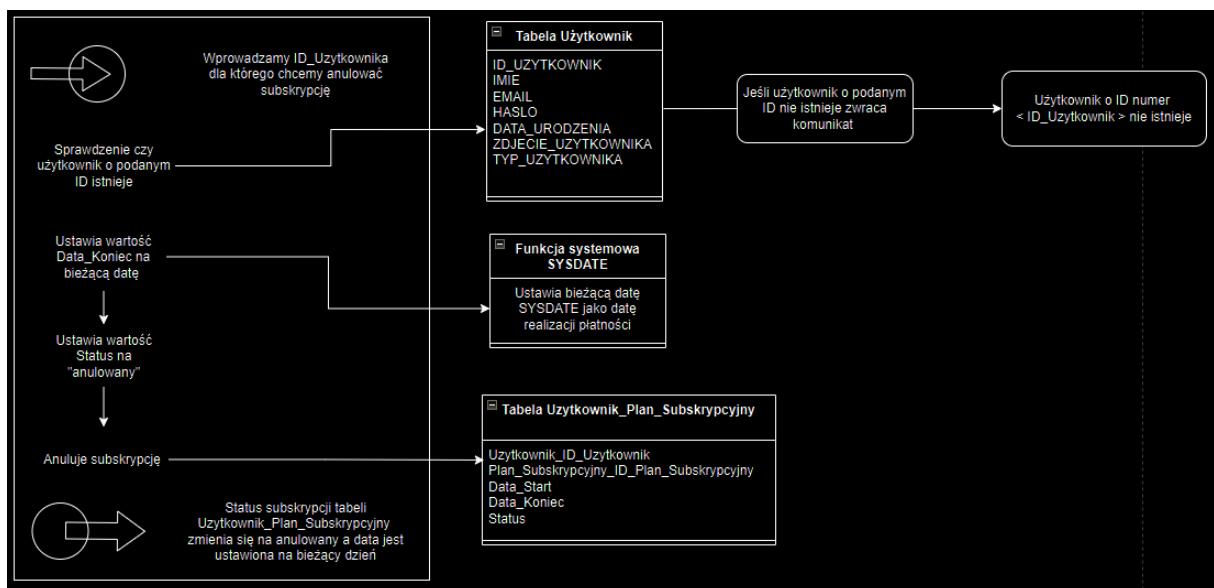
```

Results Explain Describe Saved SQL History

UZYTKOWNIK_ID_UZYTKOWNIK	PLAN_SUBSKRYPCYJNY_ID_PLAN_SUBSKRYPCYJNY	DATA_START	DATA_KONIEC	STATUS
11	2	1/24/2025	1/24/2025	AKTYWNA

## AnulujSubskrypcje

Procedura anuluje aktywną subskrypcję użytkownika, aktualizując jej status i ustawiając datę zakończenia na bieżącą datę.



```
PROCEDURE AnulujSubskrypcje(uzytkownik_id NUMBER) IS
```

```
BEGIN
```

```
  UPDATE UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
```

```
    SET STATUS = 'ANULOWANA',
```

```
    DATA_KONIEC = SYSDATE
```

```
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id;
```

```
END AnulujSubskrypcje;
```

```

1 BEGIN
2   zarzadzaniemplatnosciami.AnulujSubskrypcje(
3     uzytkownik_id => 11,
4     );
5 END;
6 SELECT *
7   FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
8 WHERE UZYTKOWNIK_ID_UZYTKOWNIK = 11;
9

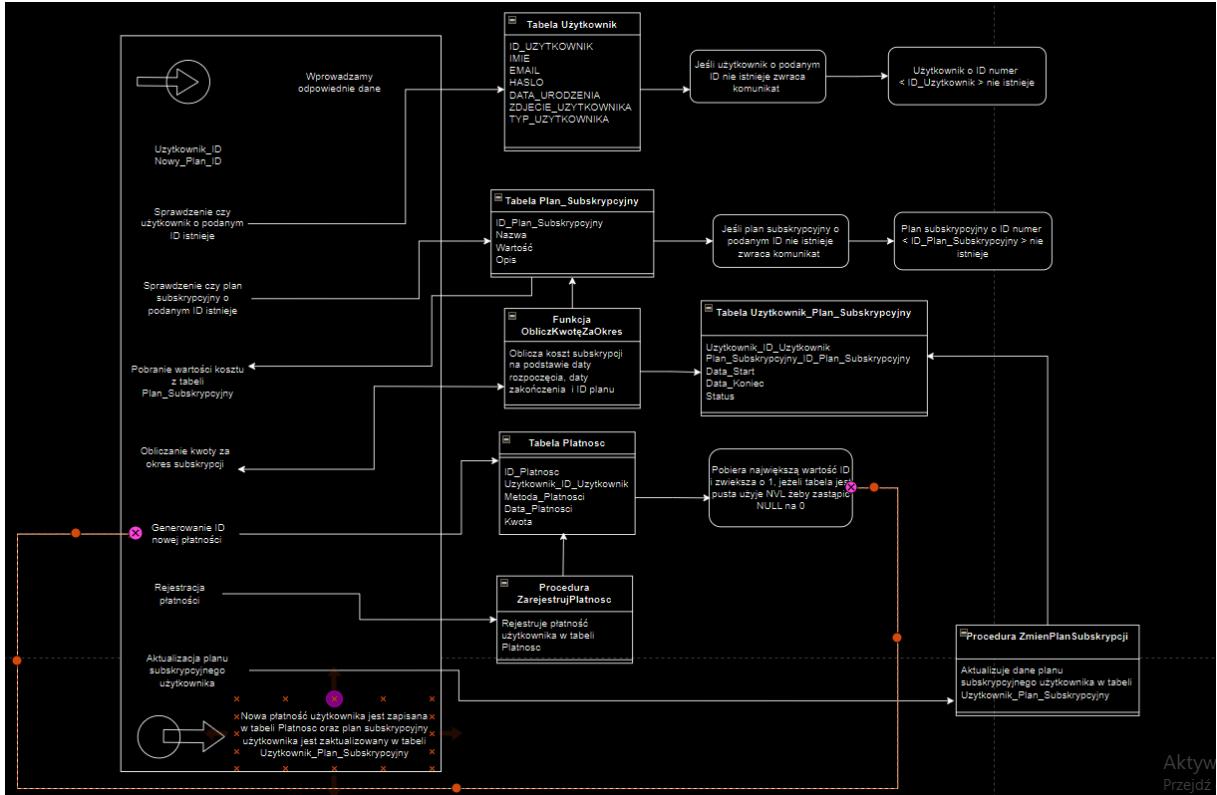
```

Results Explain Describe Saved SQL History

UZYTKOWNIK_ID_UZYTKOWNIK	PLAN_SUBSKRYPCYJNY_ID_PLAN_SUBSKRYPCYJNY	DATA_START	DATA_KONIEC	STATUS
11	2	1/24/2025	1/24/2025	ANULOWANA

## OplacSubskrypcje

Procedura realizuje opłacenie subskrypcji dla danego użytkownika, rejestrując płatność i zmieniając plan subskrypcji na nowy.



PROCEDURE OplacSubskrypcje(uzytkownik\_id NUMBER, nowy\_plan\_id NUMBER) IS

```

v_kwota NUMBER;
v_plan_wartosc NUMBER;
v_new_platnosc_id NUMBER;

BEGIN
    SELECT WARTOSC
    INTO v_plan_wartosc
    FROM PLAN_SUBSKRYPCEJNY
    WHERE ID_PLAN_SUBSKRYPCEJNY = nowy_plan_id;

    v_kwota := ObliczKwoteZaOkres(SYSDATE, ADD_MONTHS(SYSDATE, 1), nowy_plan_id);

```

```
v_kwota := ObliczKwoteZaOkres(SYSDATE, ADD_MONTHS(SYSDATE, 1), nowy_plan_id);
```

```
SELECT NVL(MAX(ID_PLATNOSC), 0) + 1
```

```
INTO v_new_platnosc_id
FROM PLATNOSC;
```

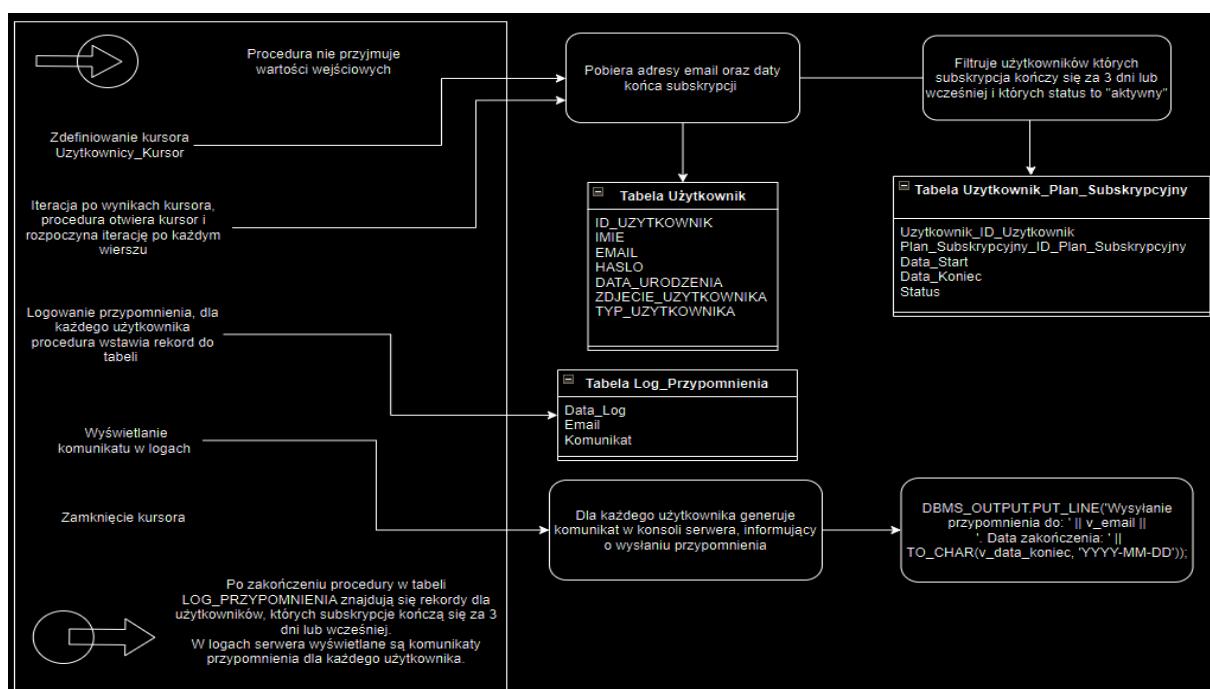
```
ZarejestrujPlatnosc(  
    id_platnosc => v_new_platnosc_id,  
    id_uzytkownika => uzytkownik_id,  
    metoda => 'ONLINE',  
    kwota => v_kwota  
);
```

ZmienPlanSubskrypcji(uzytkownik\_id, nowy\_plan\_id)

END OplacSubskrypcje;

**Przypomnij O Zakonczeniu Subskrypcji**

Procedura wysyła przypomnienia o zbliżającym się końcu subskrypcji użytkowników, których subskrypcje kończą się za 3 dni lub wcześniej.



```

PROCEDURE PrzypomnijOZakonczeniuSubskrypcji IS
    CURSOR uzytkownicy_cursor IS
        SELECT U.EMAIL, UPS.DATA_KONIEC
        FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY UPS
        JOIN UZYTKOWNIK U ON U.ID_UZYTKOWNIK = UPS.UZYTKOWNIK_ID_UZYTKOWNIK
        WHERE UPS.DATA_KONIEC - TRUNC(SYSDATE) <= 3
        AND UPS.STATUS = 'AKTYWNA';

    v_email VARCHAR2(50);
    v_data_koniec DATE;

    BEGIN
        OPEN uzytkownicy_cursor;
        LOOP
            FETCH uzytkownicy_cursor INTO v_email, v_data_koniec;
            EXIT WHEN uzytkownicy_cursor%NOTFOUND;

            -- Logowanie do tabeli
            INSERT INTO LOG_PRZYPOMNIENIA (data_log, email, komunikat)
            VALUES (
                SYSTIMESTAMP,
                v_email,
                'Przypomnienie wysłane. Data zakończenia: ' || TO_CHAR(v_data_koniec, 'YYYY-MM-DD')
            );

            DBMS_OUTPUT.PUT_LINE('Wysyłanie przypomnienia do: ' || v_email ||
                '. Data zakończenia: ' || TO_CHAR(v_data_koniec, 'YYYY-MM-DD'));

        END LOOP;
        CLOSE uzytkownicy_cursor;
    END PrzypomnijOZakonczeniuSubskrypcji;

```

```

1 BEGIN
2 | zarzadzanieplatnosciami.PrzypomnijOZakonczeniuSubskrypcji;
3 END;
4

```

Results Explain Describe Saved SQL History

Wysyłanie przypomnienia do: johndoe@example.com. Data zakończenia: 2025-01-26

## 5. Pakiety

### OBSLUGAUZYTKOWNIK

Zawarte funkcje:

- Obsluga\_liczby
- Obsluga\_litery
- Obsluga\_symbole
- CzyUzytkownikPremium

Zawarte procedury:

- DodajUzytkownika
- UsunUzytkownika
- ZaktualizujDaneUzytkownika
- ZmienHaslo

create or replace PACKAGE ObslugaUzytkownik IS

```

PROCEDURE DodajUzytkownika(
    imieU VARCHAR2,
    emailU VARCHAR2,
    hasloU VARCHAR2,
    data_urodzeniaU DATE,
    typ_uzytkownikaU VARCHAR2
);

PROCEDURE UsunUzytkownika(
    id_uzytkownikU IN INTEGER
);

PROCEDURE ZaktualizujDaneUzytkownika(
    id_uzytkownikU IN INTEGER,
    nowe_imieU IN VARCHAR2 DEFAULT NULL,
    nowy_emailU IN VARCHAR2 DEFAULT NULL,
    nowy_typ_uzytkownikaU IN VARCHAR2 DEFAULT NULL
);

PROCEDURE ZmienHaslo (
    id_uzytkownikU IN INTEGER,
    stare_hasloU IN VARCHAR2,
    nowe_hasloU IN VARCHAR2
);

```

```

);

FUNCTION CzyUzytkownikPremium (
    id_uzytkownikU IN INTEGER
) RETURN BOOLEAN;

FUNCTION obsluga_liczby(str VARCHAR2) RETURN BOOLEAN;

FUNCTION obsluga_litery(str VARCHAR2) RETURN BOOLEAN;

FUNCTION obsluga_symbole(str VARCHAR2) RETURN BOOLEAN;

END ObslugaUzytkownik;
/
create or replace PACKAGE BODY ObslugaUzytkownik IS

PROCEDURE DodajUzytkownika(
    imieU VARCHAR2,
    emailU VARCHAR2,
    hasloU VARCHAR2,
    data_urodzeniaU DATE,
    typ_uzytkownikaU VARCHAR2
) IS
    brak_imienia EXCEPTION;
    brak_emailu EXCEPTION;
    brak_hasla EXCEPTION;
    brak_typy_uzytkownika EXCEPTION;
    max_id_uzytkownika NUMBER;
BEGIN
    -- Tworzenie nowej kolekcji, jeśli jeszcze nie istnieje
    IF NOT APEX_COLLECTION.COLLECTION_EXISTS('KOLEKCJA_UZYTKOWNIKOW')
    THEN
        APEX_COLLECTION.CREATE_COLLECTION('KOLEKCJA_UZYTKOWNIKOW');
    END IF;

    -- Walidacja danych wejściowych
    IF imieU IS NULL OR obsluga_liczby(imieU) THEN
        RAISE brak_imienia;
    ELSIF emailU IS NULL OR NOT obsluga_symbole(emailU) THEN
        RAISE brak_emailu;
    ELSIF hasloU IS NULL THEN
        RAISE brak_hasla;
    ELSIF typ_uzytkownikaU NOT IN ('REGULAR', 'PREMIUM', 'FAMILY') THEN
        RAISE brak_typy_uzytkownika;
    ELSE

```

```

-- Generowanie nowego ID użytkownika
SELECT NVL(MAX(id_uzyskownik), 0) + 1 INTO max_id_uzyskownik
FROM Uzytkownik;

-- Dodanie danych użytkownika do kolekcji APEX
APEX_COLLECTION.ADD_MEMBER(
    p_collection_name => 'KOLEKCJA_UZYTKOWNIKOW',
    p_c001 => max_id_uzyskownika,
    p_c002 => imieU,
    p_c003 => emailU,
    p_c004 => hasloU,
    p_c005 => TO_CHAR(data_urodzeniaU, 'YYYY-MM-DD'),
    p_c006 => typ_uzyskownikaU
);

-- Komunikat do logów
DBMS_OUTPUT.PUT_LINE('Dane użytkownika zapisano do kolekcji
KOLEKCJA_UZYTKOWNIKOW.');
END IF;

EXCEPTION
    WHEN brak_imienia THEN
        DBMS_OUTPUT.PUT_LINE('Nie wprowadzono imienia lub zawiera niedozwolone
znaki. Uzupełnij dane.');
    WHEN brak_emailu THEN
        DBMS_OUTPUT.PUT_LINE('Nie wprowadzono adresu e-mail lub jest on
nieprawidłowy. Uzupełnij dane.');
    WHEN brak_hasla THEN
        DBMS_OUTPUT.PUT_LINE('Nie wprowadzono hasła lub jest ono nieprawidłowe.
Uzupełnij dane.');
    WHEN brak_typu_uzyskownika THEN
        DBMS_OUTPUT.PUT_LINE('Nie wprowadzono poprawnego typu subskrypcji.
Dozwolone wartości: REGULAR, PREMIUM, FAMILY.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Wystąpił nieoczekiwany błąd podczas dodawania
użytkownika.');
END DodajUzytkownika;

```

```

PROCEDURE UsunUzytkownika(
    id_uzyskownikU IN INTEGER
) IS
BEGIN
    -- Usunięcie powiązanych danych z tabel HistoriaOdtworzen
    DELETE FROM HistoriaOdtworzen
    WHERE id_uzyskownik = id_uzyskownikU;

```

```

-- Usunięcie powiązanych danych z tabel Subskrypcje
DELETE FROM Uzytkownik_Plan_Subskrypcyjny
WHERE UZYTKOWNIK_id_uzytkownik = id_uzytkowniku;

-- Usunięcie powiązanych danych z tabel Platnosci
DELETE FROM Platnosc
WHERE UZYTKOWNIK_id_uzytkownik = id_uzytkowniku;

DELETE FROM Likes
where uzytkownik_id_uzytkownik = id_uzytkowniku;

DELETE FROM LISTY_ODTWARZANIA
WHERE UZYTKOWNIK_ID_UZYTKOWNIK = ID_uzytkowniku;

DELETE FROM OBSERWUJACY
WHERE UZYTKOWNIK_ID_UZYTKOWNIK =ID_UZYTKOWNIKU;

DELETE FROM UZYTKOWNIK_PREMIUM
WHERE UZYTKOWNIK_ID_UZYTKOWNIK = ID_UZYTKOWNIKU;
-- Usunięcie rekordu użytkownika z tabeli Uzytkownik
DELETE FROM Uzytkownik
WHERE id_uzytkownik = id_uzytkowniku;

-- Informacja o powodzeniu operacji
DBMS_OUTPUT.PUT_LINE('Uzytkownik oraz wszystkie powiązane dane zostały
pomyślnie usunięte.');
EXCEPTION
WHEN NO_DATA_FOUND THEN
    -- Obsługa sytuacji, gdy użytkownik nie istnieje
    DBMS_OUTPUT.PUT_LINE('Nie znaleziono użytkownika o podanym ID.');
WHEN OTHERS THEN
    -- Obsługa wszelkich innych błędów
    DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas usuwania użytkownika.');
END UsunUzytkownika;

```

```

PROCEDURE ZaktualizujDaneUzytkownika(
    id_uzytkownikU IN INTEGER,
    nowe_imieU IN VARCHAR2 DEFAULT NULL,
    nowy_emailU IN VARCHAR2 DEFAULT NULL,
    nowy_typ_uzytkownikaU IN VARCHAR2 DEFAULT NULL
) IS
    brak_liter EXCEPTION;
    brak_symboli EXCEPTION;
    brak_liczb EXCEPTION;

```

```

niedozwolony_typ_uzytownika EXCEPTION;
czy_premium BOOLEAN;
BEGIN
    czy_premium := CzyUzytkownikPremium(id_uzytownikU);

    IF nowe_imieU IS NOT NULL AND (obsluga_liczby(nowe_imieU) OR
    obsluga_symbole(nowe_imieU)) THEN
        RAISE brak_liter;
    ELSIF nowy_emailU IS NOT NULL AND NOT obsluga_symbole(nowy_emailU) THEN
        RAISE brak_symboli;
    ELSIF nowy_typ_uzytownikaU IS NOT NULL AND nowy_typ_uzytownikaU NOT
    IN ('BASIC', 'PREMIUM', 'FAMILY') THEN
        RAISE niedozwolony_typ_uzytownika;
    END IF;

    IF czy_premium AND nowy_typ_uzytownikaU IS NOT NULL AND
    nowy_typ_uzytownikaU != 'PREMIUM' THEN
        DBMS_OUTPUT.PUT_LINE('Uzytkownik premium nie może zmienić subskrypcji
na inną niż PREMIUM.');
        RETURN;
    END IF;

    UPDATE Uzytkownik
    SET imie = NVL(nowe_imieU, imie),
        email = NVL(nowy_emailU, email),
        typ_uzytownika = NVL(nowy_typ_uzytownikaU, typ_uzytownika)
    WHERE id_uzytownik = id_uzytownikU;

    DBMS_OUTPUT.PUT_LINE('Dane uzytkownika zostały pomyślnie
zaktualizowane.');

EXCEPTION
    WHEN brak_liter THEN
        DBMS_OUTPUT.PUT_LINE('Imię zawiera niedozwolone znaki.');
    WHEN brak_symboli THEN
        DBMS_OUTPUT.PUT_LINE('Nowy email musi zawierać co najmniej jeden
symbol (@ lub podobny.).');
    WHEN niedozwolony_typ_uzytownika THEN
        DBMS_OUTPUT.PUT_LINE('Nieprawidłowy typ subskrypcji. Dozwolone
wartości: BASIC, PREMIUM, FAMILY.');
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nie znaleziono uzytkownika o podanym ID.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas aktualizacji danych
uzytkownika.');
END ZaktualizujDaneUzytkownika;

```

```

PROCEDURE ZmienHaslo (
    id_uzytkowniku IN INTEGER,
    stare_hasloU IN VARCHAR2,
    nowe_hasloU IN VARCHAR2
) IS
    aktualne_hasloU VARCHAR2(100);
    brak_liter EXCEPTION;
    brak_liczb EXCEPTION;
    brak_symboli EXCEPTION;
BEGIN
    -- Pobranie obecnego hasła użytkownika
    SELECT haslo INTO aktualne_hasloU
    FROM Uzytkownik
    WHERE id_uzytkownik = id_uzytkowniku;

    -- Sprawdzenie poprawności starego hasła
    IF aktualne_hasloU != stare_hasloU THEN
        DBMS_OUTPUT.PUT_LINE('Stare hasło jest niepoprawne.');
        RETURN;
    END IF;

    -- Sprawdzenie, czy nowe hasło zawiera litery, liczby i symbole
    IF NOT obsluga_liter(nowe_hasloU) THEN
        RAISE brak_liter;
    ELSIF NOT obsluga_liczby(nowe_hasloU) THEN
        RAISE brak_liczb;
    ELSIF NOT obsluga_symbole(nowe_hasloU) THEN
        RAISE brak_symboli;
    END IF;

    -- Aktualizacja hasła
    UPDATE Uzytkownik
    SET haslo = nowe_hasloU
    WHERE id_uzytkownik = id_uzytkowniku;

    DBMS_OUTPUT.PUT_LINE('Hasło zostało pomyślnie zmienione.');

EXCEPTION
    WHEN brak_liter THEN
        DBMS_OUTPUT.PUT_LINE('Nowe hasło musi zawierać co najmniej jedną literę.');
    WHEN brak_liczb THEN
        DBMS_OUTPUT.PUT_LINE('Nowe hasło musi zawierać co najmniej jedną cyfrę.');
    WHEN brak_symboli THEN

```

```

        DBMS_OUTPUT.PUT_LINE('Nowe hasło musi zawierać co najmniej jeden
symbol.');
WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nie znaleziono użytkownika o podanym ID.');
WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas zmiany hasła.');
END ZmienHaslo;

FUNCTION CzyUzytkownikPremium (
    id_uzytkownikU IN INTEGER
) RETURN BOOLEAN IS
    typ_uzytkownikaU VARCHAR2(20);
BEGIN
    -- Pobranie typu subskrypcji użytkownika
    SELECT typ_uzytkownika
    INTO typ_uzytkownikaU
    FROM Uzytkownik
    WHERE id_uzytkownik = id_uzytkownikU;

    -- Sprawdzenie, czy subskrypcja to PREMIUM
    IF typ_uzytkownikaU = 'PREMIUM' THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN FALSE; -- Jeśli użytkownik nie istnieje, zwracamy FALSE
END CzyUzytkownikPremium;

FUNCTION obsluga_liczby(str VARCHAR2) RETURN BOOLEAN IS
BEGIN
    IF str IS NULL THEN
        RETURN FALSE; -- Obsługa przypadku, gdy str jest NULL
    END IF;

    FOR i IN 1..LENGTH(str) LOOP
        IF ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('0') AND ASCII('9') THEN
            RETURN TRUE;
        END IF;
    END LOOP;

    RETURN FALSE;
END obsluga_liczby;

FUNCTION obsluga_litery(str VARCHAR2) RETURN BOOLEAN IS

```

```

BEGIN
  IF str IS NULL THEN
    RETURN FALSE; -- Obsługa przypadku, gdy str jest NULL
  END IF;

  FOR i IN 1..LENGTH(str) LOOP
    IF (ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('A') AND ASCII('Z'))
      OR (ASCII(SUBSTR(str, i, 1)) BETWEEN ASCII('a') AND ASCII('z')) THEN
      RETURN TRUE;
    END IF;
  END LOOP;

  RETURN FALSE;
END obsluga_litery;

FUNCTION obsluga_symbole(str VARCHAR2) RETURN BOOLEAN IS
BEGIN
  IF str IS NULL THEN
    RETURN FALSE; -- Obsługa przypadku, gdy str jest NULL
  END IF;

  FOR i IN 1..LENGTH(str) LOOP
    IF SUBSTR(str, i, 1) = '@' THEN
      RETURN TRUE;
    END IF;
  END LOOP;

  RETURN FALSE;
END obsluga_symbole;
END ObslugaUzytkownika;
/

```

## STATYSTYKI ODTWORZEŃ

Zawarte funkcje:

- LiczbaOdtworzenUtwor
- SredniCzasOdtworzenUtwor
- NajczesciejOdtwarzanyUtworDnia
- LacznyCzasOdtwarzaniaUzytkownika
- ZnajdzNajczesciejOdtwarzanyUtworUzytkownika

Zawarte procedury:

- GenerujRaportOdtworzen
- ZapiszNajczesciejOdtwarzanyUtwor
- PodsumowanieOdtwarzaniaUzytkownika
- UsunStareDaneOdtworzen

```

create or replace PACKAGE StatystykiOdtworzen AS
    FUNCTION LiczbaOdtworzenUtworu(id_utworu INT) RETURN NUMBER;
    FUNCTION SredniCzasOdtworzenUtworu(id_utworu INT) RETURN NUMBER;
    FUNCTION NajczesciejOdtwarzanyUtworDnia(data DATE) RETURN NUMBER;
    FUNCTION LacznyCzasOdtwarzaniaUzytkownika(id_uzytkownik INT) RETURN NUMBER;
    FUNCTION ZnajdzNajczesciejOdtwarzanyUtworUzytkownika(
        id_uzytkownik IN NUMBER
    ) RETURN VARCHAR2;

    PROCEDURE GenerujRaportOdtworzen(
        id_utworu IN INTEGER
    );

    PROCEDURE ZapiszNajczesciejOdtwarzanyUtwor(
        dataU IN DATE
    );

    PROCEDURE PodsumowanieOdtwarzaniaUzytkownika(
        id_uzytkownikU IN INTEGER
    );

    PROCEDURE UsunStareDaneOdtworzen(
        liczba_dni IN INTEGER
    );
END StatystykiOdtworzen;
/

```

```

create or replace PACKAGE BODY StatystykiOdtworzen AS
    FUNCTION LiczbaOdtworzenUtworu(id_utworu INT) RETURN NUMBER IS
        liczba NUMBER;
    BEGIN
        SELECT COUNT(*) INTO liczba
        FROM HistoriaOdtworzen
        WHERE HistoriaOdtworzen.id_utworu = LiczbaOdtworzenUtworu.id_utworu;
        RETURN liczba;
    END LiczbaOdtworzenUtworu;

    FUNCTION SredniCzasOdtworzenUtworu(id_utworu INT) RETURN NUMBER IS
        sredni_czas NUMBER;
    BEGIN
        SELECT AVG(czas_odtworzenia) INTO sredni_czas
        FROM HistoriaOdtworzen
        WHERE HistoriaOdtworzen.id_utworu = SredniCzasOdtworzenUtworu.id_utworu;
        RETURN sredni_czas;
    END;

```

```

END SredniCzasOdtworzenUtworu;

FUNCTION NajczesciejOdtwarzanyUtworDnia(data DATE) RETURN NUMBER IS
    id_utwor NUMBER;
BEGIN
    SELECT id_utwor
    INTO id_utwor
    FROM (
        SELECT id_utwor, COUNT(*) AS liczba_odtworzen
        FROM HistoriaOdtworzen
        WHERE TRUNC(data_odtworzenia) = data
        GROUP BY id_utwor
        ORDER BY liczba_odtworzen DESC
    )
    WHERE ROWNUM = 1;
    RETURN id_utwor;
END NajczesciejOdtwarzanyUtworDnia;

FUNCTION LacznyCzasOdtwarzaniaUzytkownika(id_uzytkownik INT)
RETURN NUMBER IS
    calkowity_czas NUMBER := 0;
BEGIN
    -- Debugging: Display the user ID being processed
    DBMS_OUTPUT.PUT_LINE('Processing total playback time for user ID: ' || id_uzytkownik);

    -- Additional query for debugging or validation
    DECLARE
        record_count NUMBER := 0;
    BEGIN
        SELECT COUNT(*)
        INTO record_count
        FROM historiaodtworzen
        WHERE id_uzytkownik = LacznyCzasOdtwarzaniaUzytkownika.id_uzytkownik;

        -- Debugging: Display number of records found
        DBMS_OUTPUT.PUT_LINE('Number of records found for user ID ' || id_uzytkownik || ':' || record_count);
    END;

    -- Calculate the total playback time for the given user
    SELECT NVL(SUM(czas_odtworzenia), 0)
    INTO calkowity_czas
    FROM historiaodtworzen
    WHERE id_uzytkownik = LacznyCzasOdtwarzaniaUzytkownika.id_uzytkownik; -- Ensure only matching rows are summed

    -- Debugging: Display the calculated total time

```

```

DBMS_OUTPUT.PUT_LINE('Total playback time for user ID ' || id_uzytkownik || ':' || 
calkowity_czas);

-- Return the calculated time
RETURN calkowity_czas;

EXCEPTION
WHEN NO_DATA_FOUND THEN
    -- Debugging: No data found for the user
    DBMS_OUTPUT.PUT_LINE('No data found for user ID: ' || id_uzytkownik);
    RETURN 0;
WHEN OTHERS THEN
    -- Debugging: Unexpected error occurred
    DBMS_OUTPUT.PUT_LINE('An unexpected error occurred for user ID: ' || 
id_uzytkownik);
    RAISE_APPLICATION_ERROR(-20001, 'An error occurred while calculating the
playback time.');
END LacznyCzasOdtwarzaniaUzytkownika;

```

```

PROCEDURE GenerujRaportOdtworzen (
    id_utworu IN INTEGER
) IS
    liczba_odtworzen NUMBER;
    sredni_czas NUMBER;
    nazwa_utworu VARCHAR2(100);
    nazwa_artysty VARCHAR2(100);
BEGIN
    -- Usuń istniejący raport dla tego samego ID utworu
    DELETE FROM RaportOdtworzen
    WHERE id_utworu = GenerujRaportOdtworzen.id_utworu;

    -- Pobierz nazwę utworu i artysty
    SELECT
        u.nazwa,
        a.nazwa
    INTO
        nazwa_utworu,
        nazwa_artysty
    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.id_utworu = GenerujRaportOdtworzen.id_utworU;

-- Pobierz liczbę odtworzeń i średni czas
liczba_odtworzen := LiczbaOdtworzenUtworu(id_utworU);
sredni_czas := SredniCzasOdtworzenUtworu(id_utworU);

-- Wstaw dane do tabeli
INSERT INTO RaportOdtworzen (
    id_utworu,
    nazwa_utworu,
    nazwa_artysty,
    liczba_odtworzen,
    sredni_czas
) VALUES (
    id_utworU,
    nazwa_utworu,
    nazwa_artysty,
    liczba_odtworzen,
    sredni_czas
);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nie znaleziono danych dla utworu o ID: ' || id_utworU);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Wystąpił nieoczekiwany błąd podczas generowania
raportu.');
        DBMS_OUTPUT.PUT_LINE('Kod błędu: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('Komunikat błędu: ' || SQLERRM);
END GenerujRaportOdtworzen;

PROCEDURE ZapiszNajczesciejOdtwarzanyUtwor (
    dataU IN DATE
) IS
    id_utwor NUMBER;
    liczba_odtworzen NUMBER;
BEGIN
    -- Pobierz najczęściej odtwarzany utwór dnia
    id_utwor := NajczesciejOdtwarzanyUtworDnia(dataU);

    -- Sprawdź liczbę odtworzeń tego utworu
    liczba_odtworzen := LiczbaOdtworzenUtworu(id_utwor);

    -- Wstaw dane do tabeli NajczesciejOdtwarzane

```

```

INSERT INTO NajczesciejOdtwarzane (
    data,
    id_utworu,
    liczba_odtworzen
) VALUES (
    dataU,
    id_utwor,
    liczba_odtworzen
);

-- Wyświetl komunikat
DBMS_OUTPUT.PUT_LINE('Najczęściej odtwarzany utwór dnia ' || TO_CHAR(dataU,
'YYYY-MM-DD') || ' zapisano do widoku.');

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Brak danych o odtworzeniach dla dnia: ' ||
TO_CHAR(dataU, 'YYYY-MM-DD'));
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Wystąpił nieoczekiwany błąd podczas zapisywania
danych.');
END ZapiszNajczesciejOdtwarzanyUtwor;

PROCEDURE PodsumowanieOdtwarzaniaUzytkownika(
    id_uzytkowniku IN INTEGER
) IS
    -- Deklaracja kursora referencyjnego
    wynik SYS_REFCURSOR;

    -- Zmienne pomocnicze do iteracji
    v_id_uzytkownik NUMBER;
    v_imie_uzytkownika VARCHAR2(100);
    v_laczny_czas_odtworzen NUMBER;
    v_najczesciej_odtworzany_utwor VARCHAR2(100);
BEGIN
    -- Otwórz kursor referencyjny, aby pobrać dane
    OPEN wynik FOR
        SELECT
            u.id_uzytkownik AS ID_Uzytkownika,
            u.imie AS Imie_Uzytkownika,
            StatystykiOdtworzen.LacznyCzasOdtwarzaniaUzytkownika(u.id_uzytkownik) AS
Laczny_Czas_Odtworzen,

            StatystykiOdtworzen.ZnajdzNajczesciejOdtwarzanyUtworUzytkownika(u.id_uzytkownik)
AS Najczesciej_Odtwarzany_Utwor
        FROM
            Uzytkownik u

```

```

WHERE
    u.id_uzytkownik = id_uzytkownikU;

-- Usuń istniejącą kolekcję (jeśli istnieje)
IF APEX_COLLECTION.COLLECTION_EXISTS('PODSUMOWANIE_ODTWARZANIA') THEN
    APEX_COLLECTION.DELETE_COLLECTION('PODSUMOWANIE_ODTWARZANIA');
END IF;

-- Utwórz nową kolekcję
APEX_COLLECTION.CREATE_COLLECTION('PODSUMOWANIE_ODTWARZANIA');

-- Iteracja po kursorze i dodanie danych do kolekcji
LOOP
    FETCH wynik INTO v_id_uzytkownik, v_imie_uzytkownika,
    v_laczny_czas_odtworzen, v_najczesciej_odtwarzany_utwor;
    EXIT WHEN wynik%NOTFOUND;

    APEX_COLLECTION.ADD_MEMBER(
        p_collection_name => 'PODSUMOWANIE_ODTWARZANIA',
        p_c001 => TO_CHAR(v_id_uzytkownik), -- Kolumna 1: ID użytkownika
        p_c002 => v_imie_uzytkownika, -- Kolumna 2: Imię użytkownika
        p_c003 => TO_CHAR(v_laczny_czas_odtworzen), -- Kolumna 3: Łączny czas
        odtworzeń
        p_c004 => v_najczesciej_odtwarzany_utwor -- Kolumna 4: Najczęściej
        odtwarzany utwór
    );
END LOOP;

-- Zamknięcie kursora
CLOSE wynik;

-- Komunikat do logów
DBMS_OUTPUT.PUT_LINE('Dane dla użytkownika o ID ' || id_uzytkownikU || ' zostały
zapisane do kolekcji.');
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Brak danych o użytkowniku lub odtworzeniach dla ID: ' || id_uzytkownikU);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas generowania podsumowania dla
użytkownika o ID: ' || id_uzytkownikU);
END PodsumowanieOdtwarzaniaUzytkownika;

```

```

PROCEDURE UsunStareDaneOdtworzen (
    liczba_dni IN INTEGER
) IS
BEGIN

```

```

        DELETE FROM HistoriaOdtworzen
        WHERE TRUNC(SYSDATE - data_odtworzenia) > liczba_dni;

        DBMS_OUTPUT.PUT_LINE('Dane odtworzeń starsze niż ' || liczba_dni || ' dni zostały
usunięte.');
        END UsunStareDaneOdtworzen;

FUNCTION ZnajdzNajczesciejOdtwarzanyUtworUzytkownika (
    id_uzytkownik IN NUMBER
) RETURN VARCHAR2 IS
    utwor_id NUMBER;
    tytul_utworu VARCHAR2(100);
BEGIN
    -- Pobierz ID utworu, który był najczęściej odtwarzany przez użytkownika
    SELECT ID_UTWORU
    INTO utwor_id
    FROM (
        SELECT ID_UTWORU, COUNT(*) AS LICZBA_ODTWORZEN
        FROM HISTORIAODTWORZEN
        WHERE ID_UZYTKOWNIK =
            znajdznajczesciejodtwarzanyutworuzytkownika.id_uzytkownik
        GROUP BY ID_UTWORU
        ORDER BY LICZBA_ODTWORZEN DESC
    )
    WHERE ROWNUM = 1;

    -- Pobierz tytuł utworu na podstawie ID
    SELECT Nazwa
    INTO tytul_utworu
    FROM UTWORY
    WHERE ID_UTWORU = utwor_id;

    -- Zwróć tytuł utworu
    RETURN tytul_utworu;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        -- Jeśli brak danych, zwróć domyślną wartość
        RETURN 'Brak danych';
    WHEN OTHERS THEN
        -- Jeśli wystąpił inny błąd, zwróć odpowiedni komunikat
        RETURN 'Błąd';
END ZnajdzNajczesciejOdtwarzanyUtworUzytkownika;

END StatystykiOdtworzen;
/

```

## ZARZADZANIEPLATNOSCIAMI

Zawarte funkcje:

- SprawdzStatusSubskrypcji
- PobierzCzasTrwaniaSubskrypcji
- ObliczKwoteZaOkres
- CzyPlatnoscZrealizowana
- IleDniDoKonca

Zawarte procedury:

- ZarejestrujPlatnosc
- ZmienPlanSubskrypcji
- AnulujSubskrypcje
- OplacSubskrypcje
- PrzypomnijOZakonczeniuSubskrypcji

```
create or replace PACKAGE ZarzadzaniePlatnosciами AS
-- Funkcje
FUNCTION SprawdzStatusSubskrypcji(id_uzytownik NUMBER) RETURN VARCHAR2;
FUNCTION PobierzCzasTrwaniaSubskrypcji(uzytownik_id NUMBER) RETURN NUMBER;
FUNCTION ObliczKwoteZaOkres(start_date DATE, end_date DATE, plan_id NUMBER)
RETURN NUMBER;
FUNCTION CzyPlatnoscZrealizowana(uzytownik_id NUMBER) RETURN BOOLEAN;
FUNCTION IleDniDoKonca(uzytownik_id NUMBER) RETURN NUMBER;

-- Procedury
PROCEDURE ZarejestrujPlatnosc(
    id_platnosc NUMBER,
    id_uzytownik NUMBER,
    metoda VARCHAR2,
    kwota NUMBER
);
PROCEDURE ZmienPlanSubskrypcji(
    uzytownik_id NUMBER,
    nowy_plan_id NUMBER
);
PROCEDURE AnulujSubskrypcje(uzytownik_id NUMBER);
PROCEDURE OplacSubskrypcje(uzytownik_id NUMBER, nowy_plan_id NUMBER);
PROCEDURE PrzypomnijOZakonczeniuSubskrypcji;
END ZarzadzaniePlatnosciами;
/
```

```
create or replace PACKAGE BODY ZarzadzaniePlatnosciами AS
-- Funkcja: Sprawdzanie statusu subskrypcji
FUNCTION SprawdzStatusSubskrypcji(id_uzytownik NUMBER) RETURN VARCHAR2 IS
    v_status VARCHAR2(20);
BEGIN
    SELECT STATUS
```

```

INTO v_status
FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
WHERE UZYTKOWNIK_ID_UZYTKOWNIK = id_uzytownik
    AND DATA_KONIEC > SYSDATE;
RETURN v_status;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 'BRAK SUBSKRYPCJI';
END SprawdzStatusSubskrypcji;

-- Funkcja: Pobranie czasu trwania subskrypcji w dniach
FUNCTION PobierzCzasTrwaniaSubskrypcji(uzytownik_id NUMBER) RETURN NUMBER
IS
    v_czas_trwania NUMBER;
BEGIN
    SELECT (DATA_KONIEC - DATA_START)
    INTO v_czas_trwania
    FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id
        AND DATA_KONIEC > SYSDATE;
    RETURN v_czas_trwania;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 0;
END PobierzCzasTrwaniaSubskrypcji;

-- Funkcja: Obliczanie kwoty za dany okres
FUNCTION ObliczKwoteZaOkres(start_date DATE, end_date DATE, plan_id NUMBER)
RETURN NUMBER IS
    v_plan_wartosc NUMBER;
    v_czas_trwania NUMBER;
BEGIN
    SELECT WARTOSC
    INTO v_plan_wartosc
    FROM PLAN_SUBSKRYPCYJNY
    WHERE ID_PLAN_SUBSKRYPCYJNY = plan_id;

    v_czas_trwania := end_date - start_date;
    RETURN v_plan_wartosc * v_czas_trwania / 30; -- Przelicznik miesięczny
END ObliczKwoteZaOkres;

-- Funkcja: Czy płatność została zrealizowana w dzisiejszym dniu
FUNCTION CzyPlatnoscZrealizowana(uzytownik_id NUMBER) RETURN BOOLEAN IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM PLATNOSC

```

```

WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id
    AND TRUNC(DATA_PLATNOSC) = TRUNC(SYSDATE);

    RETURN v_count > 0;
END CzyPlatnoscZrealizowana;

-- Funkcja: Ile dni do końca subskrypcji
FUNCTION IleDniDoKonca(uzytkownik_id NUMBER) RETURN NUMBER IS
    v_dni NUMBER;
BEGIN
    SELECT DATA_KONIEC - TRUNC(SYSDATE)
    INTO v_dni
    FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id
        AND DATA_KONIEC > SYSDATE;
    RETURN v_dni;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL;
END IleDniDoKonca;

-- Procedura: Rejestracja płatności
PROCEDURE ZarejestrujPlatnosc(
    id_platnosc NUMBER,
    id_uzytkownik NUMBER,
    metoda VARCHAR2,
    kwota NUMBER
) IS
BEGIN
    INSERT INTO PLATNOSC (ID_PLATNOSC, UZYTKOWNIK_ID_UZYTKOWNIK,
    METODA_PLATNOSCI, DATA_PLATNOSC, KWOTA)
    VALUES (id_platnosc, id_uzytkownik, metoda, SYSDATE, kwota);
END ZarejestrujPlatnosc;

-- Procedura: Zmiana planu subskrypcji
PROCEDURE ZmienPlanSubskrypcji(
    uzytkownik_id NUMBER,
    nowy_plan_id NUMBER
) IS
BEGIN
    UPDATE UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
    SET PLAN_SUBSKRYPCYJNY_ID_PLAN_SUBSKRYPCYJNY = nowy_plan_id,
        DATA_START = SYSDATE,
        DATA_KONIEC = ADD_MONTHS(SYSDATE, 1),
        STATUS = 'AKTYWNA'
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id;
END ZmienPlanSubskrypcji;

```

```

-- Procedura: Anulowanie subskrypcji
PROCEDURE AnulujSubskrypcje(uzytkownik_id NUMBER) IS
BEGIN
    UPDATE UZYTKOWNIK_PLAN_SUBSKRYPYCJNY
    SET STATUS = 'ANULOWANA',
        DATA_KONIEC = SYSDATE
    WHERE UZYTKOWNIK_ID_UZYTKOWNIK = uzytkownik_id;
END AnulujSubskrypcje;

-- Złożona procedura: Opłacenie subskrypcji
PROCEDURE OplacSubskrypcje(uzytkownik_id NUMBER, nowy_plan_id NUMBER) IS
    v_kwota NUMBER;
    v_plan_wartosc NUMBER;
    v_new_platnosc_id NUMBER;
BEGIN
    SELECT WARTOSC
    INTO v_plan_wartosc
    FROM PLAN_SUBSKRYPYCJNY
    WHERE ID_PLAN_SUBSKRYPYCJNY = nowy_plan_id;

    v_kwota := ObliczKwoteZaOkres(SYSDATE, ADD_MONTHS(SYSDATE, 1), nowy_plan_id);

    SELECT NVL(MAX(ID_PLATNOSC), 0) + 1
    INTO v_new_platnosc_id
    FROM PLATNOSC;

    ZarejestrujPlatnosc(
        id_platnosc => v_new_platnosc_id,
        id_uzytkownik => uzytkownik_id,
        metoda => 'ONLINE',
        kwota => v_kwota
    );

    ZmienPlanSubskrypcji(uzytkownik_id, nowy_plan_id);
END OplacSubskrypcje;

PROCEDURE PrzypomnijOZakonczeniuSubskrypcji IS
CURSOR uzytkownicy_cursor IS
    SELECT U.EMAIL, UPS.DATA_KONIEC
    FROM UZYTKOWNIK_PLAN_SUBSKRYPYCJNY UPS
    JOIN UZYTKOWNIK U ON U.ID_UZYTKOWNIK = UPS.UZYTKOWNIK_ID_UZYTKOWNIK
    WHERE UPS.DATA_KONIEC - TRUNC(SYSDATE) <= 3
        AND UPS.STATUS = 'AKTYWNA';
    v_email VARCHAR2(50);
    v_data_koniec DATE;
BEGIN
    OPEN uzytkownicy_cursor;
    LOOP

```

```

FETCH uzytkownicy_cursor INTO v_email, v_data_koniec;
EXIT WHEN uzytkownicy_cursor%NOTFOUND;

-- Logowanie do tabeli
INSERT INTO LOG_PRZYPOMNIENIA (data_log, email, komunikat)
VALUES (
    SYSTIMESTAMP,
    v_email,
    'Przypomnienie wysłane. Data zakończenia: ' || TO_CHAR(v_data_koniec, 'YYYY-
MM-DD')
);

DBMS_OUTPUT.PUT_LINE('Wysyłanie przypomnienia do: ' || v_email || 
    '. Data zakończenia: ' || TO_CHAR(v_data_koniec, 'YYYY-MM-DD'));
END LOOP;
CLOSE uzytkownicy_cursor;
END PrzypomnijOZakonczeniuSubskrypcji;

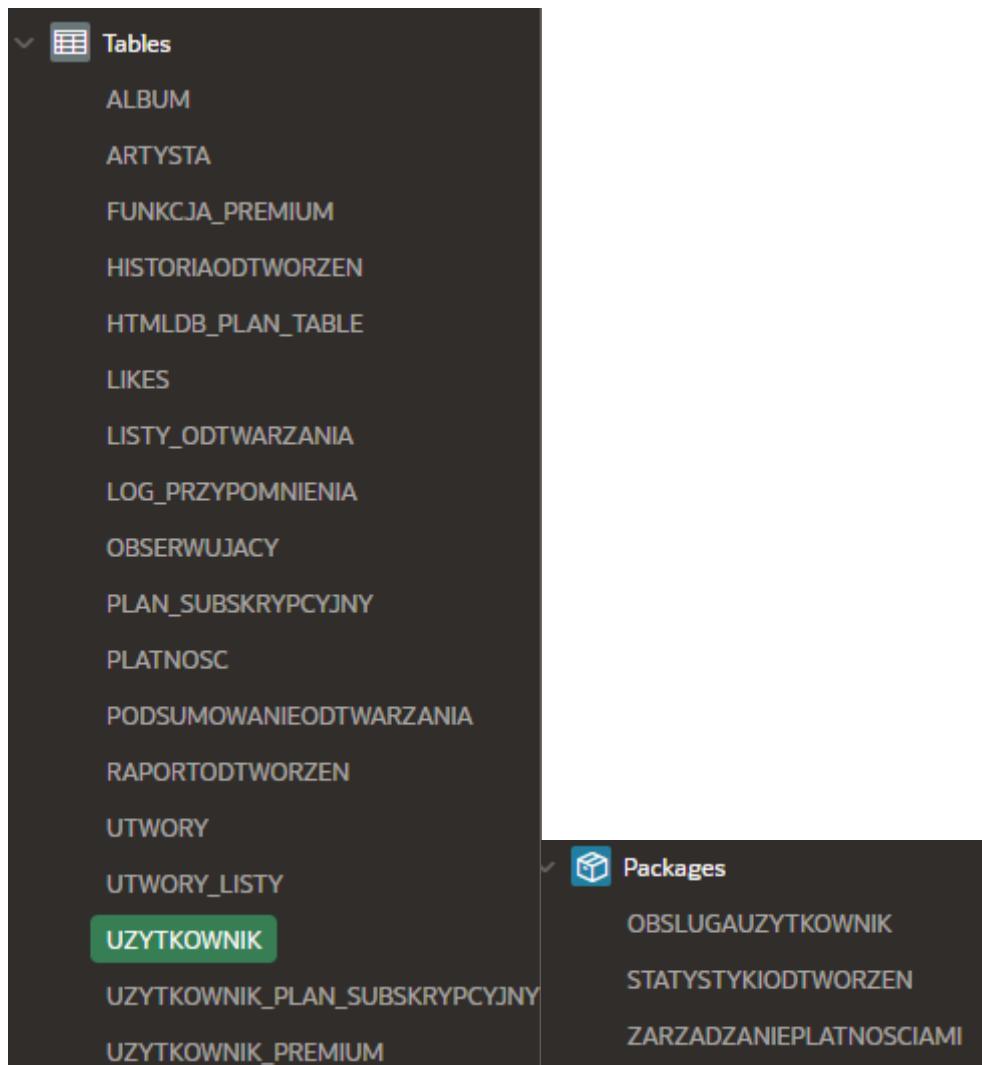
END ZarzadzaniePlatnosciami;
/

```

## 6. APEX

W tej części skupimy się na praktycznym wykorzystaniu narzędzia Oracle Application Express (APEX). W tym celu zostanie zainportowana baza danych stworzona w ramach projektu realizowanego na zajęciach z Baz Danych, a także włączone do niej wszystkie wcześniej opracowane funkcje, procedury i pakiety. Następnie ich działanie zostanie zademonstrowane w nowo utworzonej aplikacji bazodanowej.

### 1. Import bazy danych i pakietów z procedurami i funkcjami



## 2. Tworzenie aplikacji bazodanowej

Po zakończeniu importu wszystkich niezbędnych danych przystąpiliśmy do tworzenia aplikacji bazodanowej. W tym celu wybrano tryb tworzenia aplikacji przy użyciu „App Wizard”.

Następnie do strony głównej aplikacji dodano 5 przycisków i jedno okienko, z których każdy przekierowuje do innej strony. Większość z nich prezentuje kluczowe tabele zawarte w bazie danych.

The screenshot displays three main pages of the Spotify Administration application:

- Dashboard:** A blue header bar with a search icon and a circular progress bar. Below it is a white area labeled "Dashboard".
- Użytkownik search:** A table listing 20 user records. The columns are: Id, Imię, Email, Hasło, Data Urodzenia, and Typ Uzytkownika. The data includes various names like Jane Smith, Alice Johnson, Bob Brown, etc., with their respective emails, passwords, birthdates, and account types (BASIC, PREMIUM, FAMILY).
- Lista Utworów:** A table listing 122 song records. The columns are: Nazwa, Czas Trwania, Szczegóły, Album Id Album, and Średni Czas Odtwarzania. The data includes songs like "anyzowe żelki", "narcyz", "koleżanko mojej bylej", etc., with their durations, albums (Młody Say10), and average play times.

## Rejestracja płatności

Rejestracja płatności					Szczegóły Subskrypcji	Oblizanie Kosztów	Anuluj subskrypcję	Przyomienie
New								
Nowy Plan								
Q	Go	Actions						
Imię Użytkownika	Nazwa Planu Subskrypcyjnego	Data Start Subskrypcji	Data Koniec Subskrypcji	Status Subskrypcji				
Alexander Lee	Student	8/9/2022	9/9/2022	AKTYWNA				
Alice Johnson	Regular	6/19/2022	7/19/2022	AKTYWNA				
Ava Thompson	Regular	7/24/2022	8/24/2022	AKTYWNA				
Benjamin Scott	Premium	9/1/2022	10/1/2022	AKTYWNA				
Bob Brown	Regular	7/9/2022	8/9/2022	AKTYWNA				
Charlotte Green	Premium	9/3/2022	10/3/2022	AKTYWNA				
Emma Clark	Regular	9/24/2022	10/24/2022	AKTYWNA				
Eva Williams	Rodzinny	7/14/2022	8/14/2022	AKTYWNA				
Grace Perez	Premium	11/20/2022	12/20/2022	AKTYWNA				
Isabella Martinez	Student	9/1/2022	10/1/2022	AKTYWNA				
James White	Regular	8/11/2022	9/11/2022	AKTYWNA				
Jane Smith	Regular	2/9/2023	1/6/2025	AKTYWNA				
John Doe	Student	1/1/2025	1/26/2025	AKTYWNA				
Liam Rodriguez	Student	10/15/2022	11/15/2022	AKTYWNA				
Lucas Martinez	Premium	9/8/2022	10/8/2022	AKTYWNA				
Mia Hall	Student			AKTYWNA				

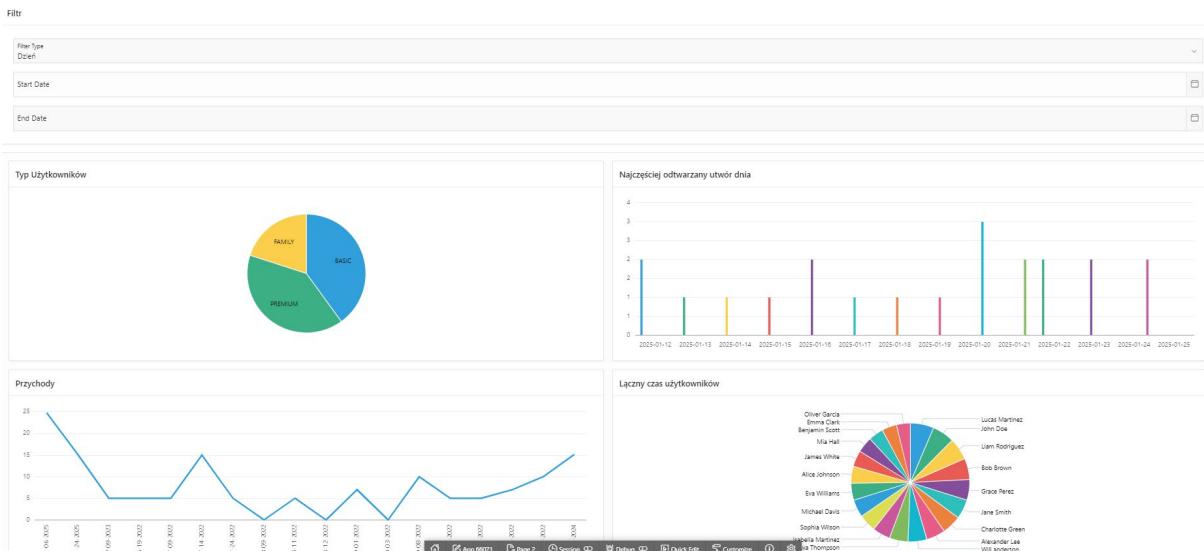
## Zarządzanie historią odtworzeń

Zarządzanie historią odtworzeń					Liczba Dni
Q	Go	Actions			
Id Odtworzenia	Id Utworu		Id Użytkownik	Data Odtworzenia	
1387	55		2	1/24/2025	
1354	20		11	1/24/2025	
1399	16		13	1/24/2025	
1266	97		18	1/24/2025	
1070	77		5	1/24/2025	
1265	52		19	1/24/2025	
1484	46		3	1/24/2025	
1091	10		6	1/24/2025	
1368	58		3	1/24/2025	
1259	53		11	1/24/2025	
1372	5		17	1/24/2025	
1465	68		11	1/24/2025	
1450	97		11	1/24/2025	
1112	76		11	1/23/2025	
1149	13		1	1/23/2025	
1215	24		18	1/23/2025	
1088	10		7	1/23/2025	
1321					1/23/2025

## Administration

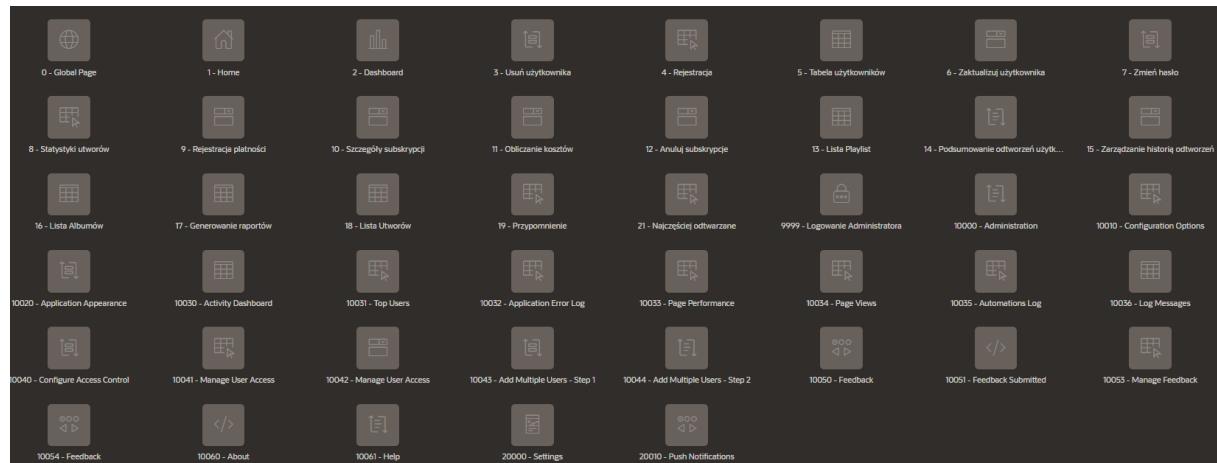
Configuration	Access Control
Configuration Options Enable or disable application features	Add User Only users defined in the application access control list may access this application
User Interface	
Theme Style Selection Set the default application look and feel	Administrator 1 Contributor 0 Reader 0
Activity Reports	Users Set level of access for authenticated users of this application
Dashboard View application activity metrics	Access Control Change access control settings and disable access control
Top Users Report of page views aggregated by user	Feedback
Application Error Log Report of errors logged by this application	Acknowledged 0 Closed 0 No Action 0 Open 0
Page Performance Report of activity and performance by application page	User Feedback Report of all feedback submitted by application users
Page Views Report of each page view by user including date of access and elapsed time	

## Dashboard



## Struktura wewnętrzna:

Strony 9999-20010 zostały utworzone podczas tworzenia aplikacji



### 3. Grupa stron dotyczących zarządzania użytkownikami

Użytkownik search

	Imię	Email	Hodo	Data Urodzenia	Typ Użytkownika
2	Jane Smith	janesmith@example.com	janey79	12/20/1988	BASIC
3	Alice Johnson	alice@example.com	alicepass	2/10/1995	BASIC
4	Bob Brown	bobbrown@example.com	bobpass	7/5/1992	BASIC
5	Eva Williams	eva@example.com	evapass	8/30/1999	PREMIUM
6	Michael Davis	michael@example.com	michaelpass	4/18/1997	PREMIUM
7	Sophia White	sophie@example.com	sophiapass	11/23/2001	FAMILY
8	Oliver Garcia	olive@example.com	oliverpass	8/8/2003	FAMILY
9	Isabella Martinez	isabella@example.com	isabellapass	3/14/2009	FAMILY
10	Liam Rodriguez	liam@example.com	liampass	6/27/2004	FAMILY
11	Will Anderson	will@example.com	willpass	10/25/1993	BASIC
12	Ava Thompson	ava@example.com	avapass	12/12/1996	BASIC
13	James White	james@example.com	jamespass	8/30/1998	BASIC
14	Emma Clark	emma@example.com	emmapass	2/28/1990	BASIC
15	Alexander Lee	alex@example.com	alexpass	5/8/2000	PREMIUM
16	Mia Hall	mia@example.com	miapass	8/15/2002	PREMIUM
17	Benjamin Scott	ben@example.com	benpass	11/5/2001	PREMIUM
18	Charlotte Green	charlotte@example.com	charlotepass	4/20/1999	PREMIUM
19	Lucas Martinez	lucas@example.com	lucaspass	3/3/2004	PREMIUM
20	Grace Perez	grace@example.com	gracepass	7/11/2003	PREMIUM
1	John Doe	john@example.com	password123	5/15/1990	BASIC

Report jest stworzony na podstawie tabeli Użytkownik

Na stronie znajdują się 4 przyciski, każdy przycisk po kliknięciu przekieruje do odpowiedniej strony

## 1. Strona Rejestracja/ Przycisk Add

### Rejestracja

Rejestracja

Imię

Email

Hasono

Data Urodzenia

Zdjęcie Użytkownika

Typ Użytkownika

Go Actions ▾

Cancel

Strona „Rejestracja” zawiera formularz stworzony na podstawie tabeli Uzytkownik. Po wpisaniu danych do formularza i wcisnięciu przycisku „Zapisz” wykonuje się procedura „DodajUzytkownika” z pakietu OBSLUGAUZYTKOWNIK”.

Validations

Processing

Processes

Process form Rejestracja

After Processing

Branches

Go To Page 4

Ajax Callback

Rejestracja

BANNER

AFTER LOGO BEFORE NAVIGATION BAR AFTER NAVIGATION BAR

TOP NAVIGATION

BREADCRUMB BAR

Breadcrumb

UP REGION BODY REGION CONTENT SUB REGIONS PREVIOUS CLOSE DELETE HELP CHANGE EDIT COPY CREATE NEXT SEARCH FIELD AND SMART FILTERS

FULL WIDTH CONTENT

BODY

Rejestracja

COPY EDIT PREVIOUS

Identification

Name: Process form Rejestracja

Type: Execute Code

Execution Chain: None

Editable Region: - Select -

Source

Location: Local Database

Language: PL/SQL

PL/SQL Code:

```
obslugauzytownik.DodajUzytkownika(
    imie => :P4_IMIE,
    email => :P4_EMAIL,
    haslo => :P4_HASLO,
    data_urodzin =>
    TO_DATE(:P4_DATA_URODZENIA, 'MM-DD-YYYY'),
    typ_uzytkownika =>
    :P4_TYP_UZYTKOWNIKA
);
end;
```

Execution

Sequence: 100

Point: Processing

Run Process: Once Per Page Visit (def)

Success Message

Success Message: Użytkownik został dodany do KOLEKCJI

Na stronie została dodana walidacja na podstawie funkcji z pakietu „OBSLUGAUZYTKOWNIK” takich jak „obsługa\_liczby”, „obsługa\_symbole”, „obsługa\_litery”

The screenshot shows the Oracle APEX validation configuration and the registration form setup.

**Validation Configuration:**

- Validations:**
  - obsługa\_liczby\_imie
  - obsługa\_liczby\_haslo
  - obsługa\_liczby\_typ\_uzytkownika
  - obsługa\_symbole\_email** (highlighted)
  - obsługa\_symbole\_imie
  - obsługa\_symbole\_haslo
  - obsługa\_litter\_email
  - obsługa\_litter\_haslo
  - obsługa\_symbole\_typ\_uzytkownika
- Processing:**
  - Process form Rejestracja** (highlighted)
- After Processing:**
  - Go To Page 4**
- Ajax Callback:**

**Registration Form Setup:**

- BANNER:** AFTER LOGO, BEFORE NAVIGATION BAR, AFTER NAVIGATION BAR
- TOP NAVIGATION:**
- BREADCRUMB BAR:**
  - Breadcrumb:** UP, REGION BODY, REGION CONTENT
  - SUB REGIONS:** PREVIOUS, CLOSE, DELETE, HELP, CHANGE, EDIT, COPY, CREATE, NEXT
- FULL WIDTH CONTENT:** BODY

**Validation Details:**

- Identification:** Name: obsługa\_symbole\_email
- Execution:** Sequence: 50
- Validation:**
  - Editable Region: - Select -
  - Type: Function Body (i)
  - Language: PL/SQL
  - PL/SQL Function Body:

```
RETURN
obsługa_uzytkownika.obsługa_symbole(:P4_EMAIL,
'@')
```

  - Always Execute: Enabled
- Error:**
  - Error Message: EMAIL musi zawierać symbol @
  - Display Location: Inline with Field
  - Associated Item: P4\_EMAIL

Po sprawdzeniu danych wpisanych do formularza wynik formularza jest zapisywany do kolekcji i wyświetlany na stronie:

The screenshot shows the registration page with a success message and a table of users.

**Registration Page:**

- Form Fields:** Imię, Email, Hasło, Data Urodzenia, Zdjęcie Użytkownika (Choose File), Typ Użytkownika (dropdown).
- Table:** Shows a list of users with columns: Id Użytkownika, Imię, Email, Hasło, Data Urodzenia, Typ Użytkownika.

Id Użytkownika	Imię	Email	Hasło	Data Urodzenia	Typ Użytkownika
21	Andrii	Andrii@gmail.com	Andrii123!	2002-12-28	PREMIUM

**Success Message:** Użytkownik został dodany do KOLEKCJI

Po wciśnięciu przycisku „Zapisz do tabeli” zostanie użyta akcja dynamiczna i rekord z kolekcji zostanie zapisany do tabeli Uzytkownik

The screenshot shows the Oracle Apex interface. On the left, there's a tree view with a node labeled "Zapisz do tabeli". Under it, "Dynamic Actions" is expanded, showing an "Insert" action. This action has a "True" condition, which contains an "Insert to Table" step pointing to the "Uzytkownik" table, and an "Alert" step displaying the message "Uzytkownik został dodany do tabeli". Below this, there's a "False" condition with another "Insert to Table" step. On the right, the application's code editor displays the following PL/SQL code:

```

1 BEGIN
2     FOR rec IN (
3         SELECT c002 AS Imie,
4             c003 AS Email,
5             c004 AS Haslo,
6             TO_DATE(c005, 'YYYY-MM-DD') AS Data_Urodzenia,
7             c006 AS Typ_Uzytkownika
8         FROM APEX_COLLECTIONS
9         WHERE COLLECTION_NAME = 'KOLEKCJA_UZYTKOWNIKOW'
10    ) LOOP
11        -- Wstawianie danych do tabeli Uzytkownik
12        INSERT INTO Uzytkownik (
13            id_uzytkownik,
14            imie,
15            email,
16            haslo,
17            data_urodzenia,
18            typ_uzytkownika
19        ) VALUES (
20            (SELECT NVL(MAX(id_uzytkownik), 0) + 1 FROM Uzytkownik),
21            rec.Imie,
22            rec.Email,
23            rec.Haslo,
24            rec.Data_Urodzenia,
25            rec.Typ_Uzytkownika
26        );
27    END LOOP;
28
29    -- Opróżnij kolekcję po zapisaniu danych
30    APEX_COLLECTION.DELETE_COLLECTION('KOLEKCJA_UZYTKOWNIKOW');
31 END;
32

```

	Imie	Email	Haslo	Data Urodzenia	Typ Uzytkownika
2	Jane Smith	janesmith@example.com	qwerty789	12/20/1988	BASIC
3	Alice Johnson	alice@example.com	alicepass	2/10/1995	BASIC
21	Andrii	Andrii@gmail.com	Andrii123!	12/28/2002	PREMIUM
4	Bob Brown	bobbrown@example.com	bobbypass	7/5/1992	BASIC
5	Eva Williams	eva@example.com	evapass	9/30/1999	PREMIUM

## 2. Strona Usuń użytkownika/Przycisk Delete

The screenshot shows a user interface for deleting a user. At the top is a header bar with the text "Usuń użytkownika". Below this is a form area. On the left is a dropdown menu labeled "New". On the right are two buttons: "Cancel" and "Delete".

Strona „Usuń użytkownika” zawiera formularz stworzony na podstawie tabeli „Uzytkownik”. Po wybraniu użytkownika z wypadającej listy typu select list i wcisnięciu przycisku „Delete” wykonuje się procedura „UsunUzytkownika” z pakietu „OBSLUGAUZYTKOWNIK”.

The screenshot shows the APEX developer interface for configuring a process named "Process DELETE". The configuration includes:

- Identification:** Name: Process DELETE, Type: Execute Code, Execution Chain: None, Editable Region: - Select -.
- Source:** Location: Local Database, Language: PL/SQL.
- PL/SQL Code:**

```

DECLARE
    v_checksum NUMBER;
BEGIN
    -- Pobierz aktualną wartość checksum z bazy
    SELECT DBMS_UTILITY.GET_HASH_VALUE(ID_UZYTKOWNIK,
                                         1_00000000) AS CHECKSUM
    INTO v_checksum
    FROM UZYTKOWNIK
    WHERE ID_UZYTKOWNIK =
        :P3_ID_UZYTKOWNIK;

    -- Pordwaj wartość checksum
    IF v_checksum != :P3_CHECKSUM THEN
        RAISE_APPLICATION_ERROR(-20001,
                                'Checksum mismatch: record has been modified by another user.');
    END IF;

    -- Usuń użytkownika
    DELETE FROM UZYTKOWNIK
    WHERE ID_UZYTKOWNIK = :P3_ID_UZYTKOWNIK;

```
- Execution:** Sequence: 20, Point: Processing, Run Process: Once Per Page Visit (def).
- Success Message:** Success Message: użytkownik został usunięty z bazy danych.

Select list P3\_ID\_UZYTKOWNIK jest zdefiniowany przez SQL Query

The screenshot shows the configuration of a select list "P3\_ID\_UZYTKOWNIK" in the "Body" section. The configuration includes:

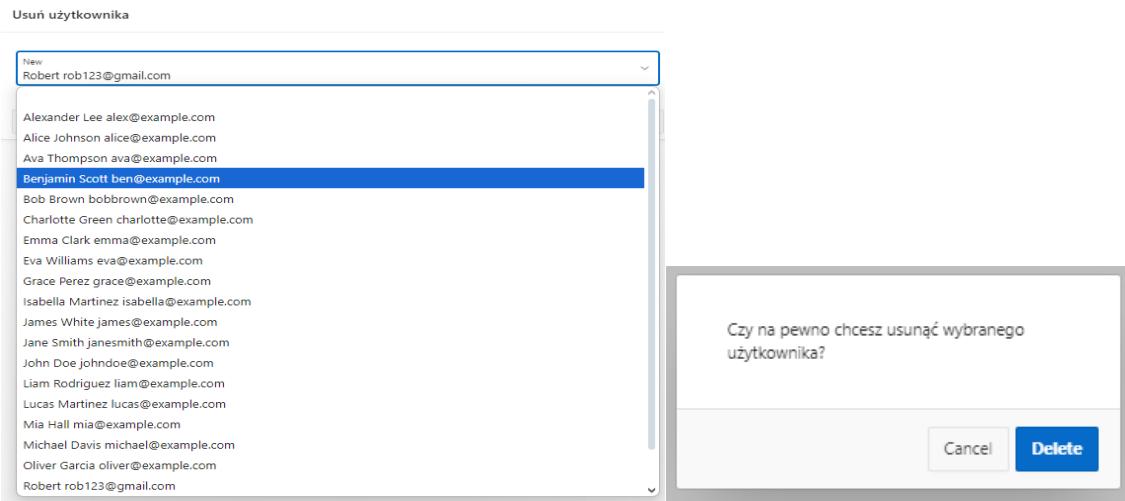
- Value Required:** Off.
- List of Values:**
  - Type:** SQL Query.
  - SQL Query:**

```

SELECT IMIE || ' ' || EMAIL AS DISPLAY_VALUE, ID_UZYTKOWNIK AS RETURN_VALUE
FROM UZYTKOWNIK
ORDER BY IMIE;

```
  - Display Extra Values:** On.
  - Display Null Value:** On.
  - Null Display Value:** (empty field).
  - Null Return Value:** (empty field).

Po wybraniu użytkownika z listy wypadającej i wcisnięciu przycisku musimy potwierdzić usunięcie użytkownika, po potwierdzeniu wykonuje się procedura i przekierowuje do strony Tabela Użytkownika



**Użytkownik search**

**Użytkownik został usunięty z bazy danych**

Id	Imię	Email	Hasło	Data Urodzenia	Typ Użytkownika
2	Jane Smith	janesmith@example.com	qwerty789	12/20/1988	BASIC
3	Alice Johnson	alice@example.com	alicepass	2/10/1995	BASIC
4	Bob Brown	bobbrown@example.com	bobbypass	7/5/1992	BASIC
5	Eva Williams	eva@example.com	evapass	9/30/1999	PREMIUM
6	Michael Davis	michael@example.com	michaelpass	4/18/1987	PREMIUM
7	Sophia Wilson	sophia@example.com	sophiapass	11/22/2001	FAMILY
8	Oliver Garcia	oliver@example.com	oliverpass	8/8/2003	FAMILY
9	Isabella Martinez	isabella@example.com	isabellapass	3/14/2005	FAMILY
10	Liam Rodriguez	liam@example.com	liampass	6/27/2004	FAMILY
11	Will Anderson	will@example.com	willpass	10/25/1993	BASIC
12	Ava Thompson	ava@example.com	avapass	12/12/1996	BASIC
13	James White	james@example.com	jamespass	8/30/1986	BASIC
14	Emma Clark	emma@example.com	emmapass	2/28/1990	BASIC
15	Alexander Lee	alex@example.com	alexpass	5/8/2000	PREMIUM
16	Mia Hall	mia@example.com	miapass	9/15/2002	PREMIUM
17	Benjamin Scott	ben@example.com	benpass	11/5/2001	PREMIUM
18	Charlotte Green	charlotte@example.com	charlottepass	4/20/1999	PREMIUM
19	Lucas Martinez	lucas@example.com	lucaspass	3/3/2004	PREMIUM
20	Grace Perez	grace@example.com	gracepass	7/11/2003	PREMIUM
1	John Doe	johndoe@example.com	password123	5/15/1990	BASIC

1 rows selected      Total 20

### 3. Strona Zaktualizuj Użytkownika/Przycisk Edit

Zaktualizuj użytkownika

ID	
Imię	
Email	
Typ Użytkownika	

Strona „Zaktualizuj Użytkownika” zawiera formularz stworzony na podstawie tabeli „Użytkownik”. Po wybraniu użytkownika z wypadającej listy typu select list i wcisnięciu przycisku „Aktualizacja” wykonuje się procedura „ZaktualizujDaneUzytkownika” z pakietu **OBSLUGA\_UZYTKOWNIK**”

The screenshot shows the Oracle APEX application builder interface. On the left, the 'Validations' section lists several validation rules for fields like 'obsugaliczb\_imie', 'obsugasymbole\_imie', 'obsugasymbole\_email', and 'obsugalitery\_email'. Below it, the 'Processes' section highlights the 'Process form EDIT Uzytkownik' process. The main area displays the 'Zaktualizuj uzytkownika' page structure with regions for 'BANNER', 'TOP NAVIGATION', 'REGION BODY', and 'SUB REGIONS'. The 'REGION BODY' contains a 'Breadcrumb' with 'UP' and 'REGION CONTENT' buttons. The 'SUB REGIONS' section includes buttons for 'PREVIOUS', 'CLOSE', 'DELETE', 'HELP', 'CHANGE', 'EDIT', 'COPY', and 'CREATE'. To the right, the 'Identification' panel shows the process details: Name is 'Process form EDIT Uzytkownika', Type is 'Form - Automati...', Execution Chain is 'None', and Form Region is 'EDIT Uzytkownika'. The 'Settings' panel shows the target type as 'PL/SQL Code' and the PL/SQL code for the process:

```

BEGIN
  ObslugaUzytkownik.ZaktualizujDaneUzytkownika(
    id_uzytkowniku => :P6_ID_UZYTKOWNIK,
    nowe_imieU => :P6_IMIE,
    nowy_emailU => :P6_EMAIL,
    nowy_typ_uzytkownikaU =>
    :P6_TYP_UZYTKOWNIKA
  );
END;
  
```

Select list P6\_ID\_UZYTKOWNIK jest zdefiniowany przez SQL Query

The screenshot shows the configuration of the 'P6\_ID\_UZYTKOWNIK' select list. In the 'Body' section, under 'Region Body', the list is defined with items for 'P6\_ID\_UZYTKOWNIK', 'P6\_IMIE', 'P6\_EMAIL', 'P6\_HASLO', 'P6\_DATA\_URODZENIA', 'P6\_ZDIECIE\_UZYTKOWNIKA', and 'P6\_TYP\_UZYTKOWNIKA'. The 'List of Values' configuration on the right specifies a SQL query to generate the options:

```

SELECT IMIE || ' ' || EMAIL AS DISPLAY_VALUE, ID_UZYTKOWNIK AS RETURN_VALUE
FROM UZYTKOWNIK
ORDER BY IMIE;
  
```

Other settings include 'Format Mask' (empty), 'Height' (1 line), 'Validation' (disabled), and 'Display Extra Values' and 'Display Null Value' (both enabled).

Select list P6\_TYP\_UZYTKOWNIKA jest zdefiniowany przez SQL Query

P6\_ZDJECIE\_UZYTKOWNIKA  
P6\_TYP\_UZYTKOWNIKA

Close CANCEL  
Create Aktualizuj  
Alering

REGION CONTENT  
SUB REGIONS  
CLOSE HELP DELETE  
CREATE Aktualizuj  
FOOTER  
DIALOGS DRAWERS AND MODULES

Type SQL Query  
SQL Query  
SELECT 'BASIC' AS DISPLAY\_VALUE, 'BASIC' AS RETURN\_VALUE FROM DUAL  
UNION ALL  
SELECT 'PREMIUM', 'PREMIUM' FROM DUAL  
UNION ALL  
SELECT 'FAMILY', 'FAMILY' FROM DUAL;  
Display Extra Values  
Display Null Value

Na stronie została dodana walidacja na podstawie funkcji z pakietu „OBSLUGAUZYTKOWNIK” takich jak „obsługa\_liczby”, „obsługa\_symbole”, „obsługa\_litery”

Validations  
obsługalicz\_imie  
obsługasymbole\_imie  
obsługasymbole\_email  
obsługalitery\_email

Processing  
Processes  
Process form EDIT Uzytkownik  
After Processing  
Branches  
Go To Page 1

Ajax Callback

Zaktualizuj uzytkownika

BANNER  
AFTER LOGO BEFORE NAVIGATION BAR AFTER NAVIGATION BAR  
TOP NAVIGATION  
BREADCRUMB BAR  
Breadcrumb UP REGION BODY REGION CONTENT SUB REGIONS PREVIOUS CLOSE DELETE HELP CHANGE EDIT COPY CREATE NEXT SEARCH FIELD AND SMART FILTERS FULL WIDTH CONTENT

Name obsługasymbole\_imie  
Execution Sequence 20  
Validation  
Editable Region - Select -  
Type Function Body (r)  
Language PL/SQL  
PL/SQL Function Body  
RETURN NOT obslugauzytkownik.obsługa\_symbole(:P6\_IMIE);  
Always Execute  
Error  
Error Message Imię nie może zawierać symbole  
Display Location Inline with Field

Zaktualizuj uzytkownika

ID  
Andrii Andrii@gmail.com

Imie  
Robert123  
Imię nie może zawierać cyfry

Email  
robert@gmail.com  
Email musi zawierać symbol @

Typ Uzytkownika  
PREMIUM

Cancel Zaktualizuj

Po sprawdzeniu danych wpisanych do formularza i wciśnięciu przycisku wynik formularza jest zapisywany do tabeli i przekierowuje na stronę Tabela Użytkowników:

The screenshot shows a user management interface. At the top, a green banner displays a checkmark icon and the text "Dane użytkownika zostały zmienione". Below the banner, the title "Użytkownik search" is visible, along with buttons for "Add", "Delete", "Edit", and "Edit password". A search bar with placeholder text "Search: All Text Columns" and a "Go" button are also present. To the right of the search bar is a "Reset" button. The main area contains a table with the following data:

	Imie	Email	Hasło	Data Urodzenia	Typ Uzytkownika
2	Jane Smith	janesmith@example.com	qwerty789	12/20/1988	BASIC
3	Alice Johnson	alice@example.com	alicepass	2/10/1995	BASIC
21	Robert	rob123@gmail.com	Andrii123!	12/28/2002	PREMIUM
4	Bob Brown	bobbrown@example.com	bobbypass	7/5/1992	BASIC
5	Eva Williams	eva@example.com	evapass	9/30/1999	PREMIUM
6	Michael Davis	michael@example.com	michaelpass	4/18/1987	PREMIUM

#### 4. Strona Zmień hasło/ Przycisk Edit Password

Zmień hasło

Uzytkownik

Hasło

Nowe Hasło

Cancel

Zmien

Strona „Zmień hasło” zawiera formularz stworzony na podstawie tabeli „Uzytkownik”. Po wybraniu użytkownika z wypadającej listy typu select list, wpisaniu aktualnego hasła użytkownika, nowego hasła spełniającego wymogi walidacji i wcisnięciu przycisku „Zmień” wykonuje się procedura "ZmienHaslo" z pakietu „OBSLUGAUZYTKOWNIK”.

```

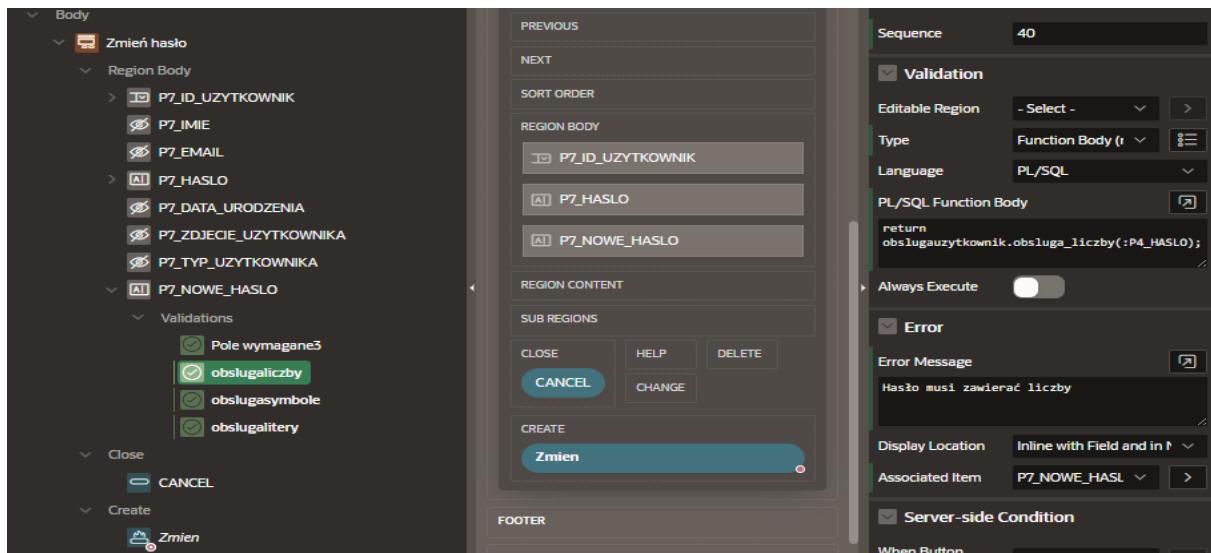
BEGIN
    ObslugaUzytkownik.ZmienHaslo(
        id_uzytkowniku => :P7_ID_UZYTKOWNIK,
        stare_hasloU => :P7_HASLO,
        nowe_hasloU => :P7_NOWE_HASLO
    );
END;
  
```

Select list P7\_ID\_UZYTKOWNIK jest zdefiniowany przez SQL Query

```

SELECT imie || ' ' || email AS
display_value, id_uzytkownik AS return_value
FROM Uzytkownik
ORDER BY imie;
  
```

Na stronie została dodana walidacja na podstawie funkcji z pakietu „OBSLUGAUZYTKOWNIK” takich jak „obsługa\_liczby”, „obsługa\_symbole”, „obsługa\_litery”



## Change Pass

Zmień hasło

Użytkownik: Robert rob123@gmail.com

Hasło: Andrii123

Nowe Hasło: (Field highlighted with a red border)

**Hasło musi zawierać litery**

Cancel Zmien

Po sprawdzeniu danych wpisanych do formularza i wciśnięciu przycisku następuje aktualizacja wartości hasła dla wybranego użytkownika i przekierowuje na stronę Tabela Użytkowników:

Użytkownik search					
Actions		Hasło zostało zmienione			
Id	Imię	Email	Hasło	Data Urodzenia	Typ Użytkownika
2	Jane Smith	janesmith@example.com	qwerty789	12/20/1968	BASIC
3	Alice Johnson	alice@example.com	alicepass	2/10/1995	BASIC
21	Robert	rob123@gmail.com	Robert123\$	12/28/2002	PREMIUM

## 4. Grupa stron dotyczących statystyk utworów i użytkowników

### Lista utworów

#### Lista Utworów

	Nazwa	Czas Trwania	Sciezka	Album Id Album	Sredni Czas Odtworzenia
<input checked="" type="checkbox"/>	anyżowe żelki	133	Młody Say10		120
<input type="checkbox"/>	narcyz	143	Młody Say10		130
<input type="checkbox"/>	koleżanki mojej byłej	152	Młody Say10		138
<input type="checkbox"/>	drzewko	167	Młody Say10		150
<input type="checkbox"/>	to ostatni raz gdy jadę nad bartyk	155	Młody Say10		140
<input type="checkbox"/>	miałem kolegę bartka	166	Młody Say10		152
<input type="checkbox"/>	kupiłem, co paż czu toroski?	147	Młody Say10		135
<input type="checkbox"/>	jestes najlepszy/najlepsza	144	Młody Say10		130
<input type="checkbox"/>	mam chybą za drogie auto	177	Młody Say10		160
<input type="checkbox"/>	narcyz w klubie	126	Młody Say10		115
<input type="checkbox"/>	to dopiero początek	97	nauczyłem się przeklinać		85
<input type="checkbox"/>	odkąd nauczyłem się przeklinać	149	nauczyłem się przeklinać		135
<input type="checkbox"/>	generacja hip hop	172	nauczyłem się przeklinać		155
<input type="checkbox"/>	xo	164	nauczyłem się przeklinać		148
<input type="checkbox"/>	palenie zabija	145	nauczyłem się przeklinać		130
<input type="checkbox"/>	death metalowy t-shirt	157	nauczyłem się przeklinać		140
<input type="checkbox"/>	mam na twarzy krew i tym razem nie jest szluczna	162	nauczyłem się przeklinać		145
<input type="checkbox"/>	jakbym miał w kiesu klamkę	167	nauczyłem się przeklinać		150
<input type="checkbox"/>	comme des gars	160	nauczyłem się przeklinać		145
<input type="checkbox"/>	młody say10	154	nauczyłem się przeklinać		140
<input type="checkbox"/>	charger	160	nauczyłem się przeklinać		145
<input type="checkbox"/>	hellcat redeye	147	nauczyłem się przeklinać		130
<input type="checkbox"/>	zadzi dla mnie	144	nauczyłem się przeklinać		130
<input type="checkbox"/>	to nie koniec	132	nauczyłem się przeklinać		120
<input type="checkbox"/>	najlepsi pod stołem	164	nauczyłem się przeklinać		170

1 rows selected

App 66071 Page 18 Session Debug Quick Edit Customize

Total 122

Report jest stworzony na podstawie tabeli Utwory

Na stronie znajdują się 6 przycisków, każdy przycisk po kliknięciu przekieruje do odpowiedniej strony

## 1. Strona Lista Playlist/Przycisk Playlisty

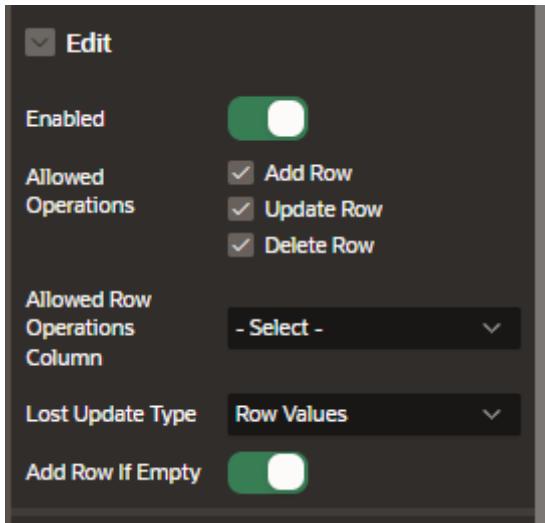
		Id Listy	Nazwa	Data Stworzenia	Użytkownik Id Użytkownik	Publiczna
<input checked="" type="checkbox"/>		1	Ulubione utwory	1/10/2023	1	tak
<input type="checkbox"/>		2	Playlista na weekend	2/15/2023	2	nie
<input type="checkbox"/>		3	Melodie relaksacyjne	4/20/2023	3	tak
<input type="checkbox"/>		4	Energetyczne rytmы	12/5/2022	4	nie
<input type="checkbox"/>		5	Mix hitów	11/30/2022	5	tak
<input type="checkbox"/>		6	Chillout na wieczór	8/10/2022	6	nie
<input type="checkbox"/>		7	Lista do biegania	3/25/2023	7	tak
<input type="checkbox"/>		8	Utwory na podróz	9/10/2022	8	nie
<input type="checkbox"/>		9	Najlepsze przeboje	6/15/2023	9	tak
<input type="checkbox"/>		10	Playlista na trening	7/20/2022	10	nie
<input type="checkbox"/>		11	Ulubione melodyki	8/2/2023	11	tak
<input type="checkbox"/>		12	Relaksujące dźwięki	5/10/2022	12	nie
<input type="checkbox"/>		13	Mix hip-hopowy	4/3/2022	13	tak
<input type="checkbox"/>		14	Lista imprezowa	1/20/2023	14	nie
<input type="checkbox"/>		15	Muzyka do pracy	12/30/2022	15	tak
<input type="checkbox"/>		16	Popowe hity	11/5/2023	16	nie
<input type="checkbox"/>		17	Rockowe kiszyki	10/17/2022	17	tak
<input type="checkbox"/>		18	Lista na zimne wieczory	9/22/2023	18	nie
<input type="checkbox"/>		19	Meandry melodi	6/7/2022	19	tak
<input type="checkbox"/>		20	Dobre rytmы	4/10/2023	20	nie

1 rows selected

Total 20

Report jest stworzony na podstawie tabeli Listy\_ odtworzen

Przyciski na stronie są zdefiniowane przy użyciu wbudowanych narzędzi Apex



## 2. Strona Lista Albumów/Przycisk Albumy

Lista Albumów

		Nazwa	Data Wydania	Artysta Id Artyst
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Młody Say10	9/22/2022	Chivas
<input type="checkbox"/>	<input type="checkbox"/>	nauczyłem się przeklinać	5/21/2021	Chivas
<input type="checkbox"/>	<input type="checkbox"/>	Uzumaki forma ostateczna	7/27/2023	Szpeku
<input type="checkbox"/>	<input type="checkbox"/>	Atypowy	10/20/2018	Szpeku
<input type="checkbox"/>	<input type="checkbox"/>	One last time	9/28/2022	ZYZZ
<input type="checkbox"/>	<input type="checkbox"/>	Say it right	4/20/2022	ZYZZ
<input type="checkbox"/>	<input type="checkbox"/>	Safe and sound	9/10/2022	BAKI
<input type="checkbox"/>	<input type="checkbox"/>	Runaway	9/11/2022	BAKI
<input type="checkbox"/>	<input type="checkbox"/>	Goodbye and Good Riddance	12/10/2018	Juice WRLD
<input type="checkbox"/>	<input type="checkbox"/>	Playdead	9/27/2023	SIM
<input type="checkbox"/>	<input type="checkbox"/>	*	3/3/2017	Ed Sheeran
<input type="checkbox"/>	<input type="checkbox"/>	Serymentainist	3/29/2023	Golubenko
<input type="checkbox"/>	<input type="checkbox"/>	The evil in which we thrive	2/2/2023	Smora
<input type="checkbox"/>	<input type="checkbox"/>	Oppai and Thicc Thights	2/19/2023	ovgi

1 rows selected

Total 14

Report jest stworzony na podstawie tabeli Album

The screenshot shows the Oracle APEX report builder interface. On the left, there's a tree view with nodes like 'Breadcrumb Bar', 'Body' (selected), and 'Post-Rendering'. Under 'Body', 'Lista Albumów' is selected, revealing its configuration options: 'Columns', 'Column Groups', and 'Saved Reports'. To the right, there are three main sections: 'BANNER' (with tabs for 'AFTER LOGO', 'BEFORE NAVIGATION BAR', and 'AFTER NAVIGATION BAR'), 'TOP NAVIGATION' (with a 'BREADCRUMB BAR' section containing 'UP', 'REGION BODY', 'REGION CONTENT', and 'SUB REGIONS'), and 'Name' (containing fields for 'Title', 'Type' set to 'Interactive Grid', and a 'Source' section with 'Location' set to 'Local Database', 'Type' set to 'Table / View', 'Table Owner' set to 'Parsing Schema', 'Table Name' set to 'ALBUM', and a 'Where Clause' field).

Przyciski na stronie są zdefiniowane przy użyciu wbudowanych narzędzi Apex

The screenshot shows the Oracle APEX edit configuration dialog for the 'Edit' button. It includes sections for 'Enabled' (switched on), 'Allowed Operations' (checkboxes for 'Add Row', 'Update Row', and 'Delete Row' all checked), 'Allowed Row Operations Column' (dropdown set to '- Select -'), 'Lost Update Type' (dropdown set to 'Row Values'), 'Add Row If Empty' (switched on), and 'Edit Authorization' (sections for 'Add', 'Update', and 'Delete' each with a dropdown set to '- Select -').

### 3. Strona Najczęściej odtwarzane/Przycisk Popularne Najczęściej odtwarzane

Date  
1/25/2025

Generuj

Actions ▾

Data	Utwor	Artysta	Liczba Odtworzen
1/24/2025	death metalowy t-shirt	Chivas	1
1/24/2025	Intro	Golubenko	2
1/24/2025	narcyz w klubie	Chivas	1
1/24/2025	Lucid Dreams	Juice WRLD	1
1/24/2025	Runaway	BAKI	1
1/24/2025	DO THE DANCE	SiM	1
1/24/2025	Intro	Juice WRLD	1
1/24/2025	PLAYDEAD	SiM	1
1/24/2025	młody say10	Chivas	1
1/24/2025	Betrayal	Juice WRLD	1
1/24/2025	to ostatni raz gdy jadę nad bałtyk	Chivas	1
1/24/2025	Oddajemy krew Wampirom	Szpaku	1
1/23/2025	BBT	SiM	1
1/23/2025	TOO LATE	SiM	1
1/23/2025	Karma	Juice WRLD	2
1/23/2025	Oddajemy krew Wampirom	Szpaku	1
1/23/2025	narcyz w klubie	Chivas	1
1/23/2025	najlepsi pod słońcem	Chivas	1
1/23/2025	Intro	Golubenko	1
1/23/2025	HIDE and SEEK	SiM	1
1/23/2025	mam chyba za drogie auto	Chivas	1

Strona „Najczęściej odtwarzane” zawiera formularz z reportem. Po wybraniu daty z kalendarza i wcisnięciu przycisku „Generuj” wykonuje się procedura „ZapiszNajczesciejOdtwarzanyUtwor” z pakietu „StatystykiOdtworzen”.

Report jest stworzony na podstawie SQL Query

The screenshot shows the Oracle APEX Report Builder interface. On the left, the navigation tree includes 'Breadcrumb Bar' and 'Body' sections. Under 'Body', the 'Najczęściej odtwarzane' report is selected. The report's properties are displayed on the right, including:

- Name:** Najczęściej odtwarzane
- Title:** [redacted]
- Type:** Interactive Report
- Source:**
  - Location:** Local Database
  - Type:** SQL Query
  - SQL Query:**

```
SELECT
    n.data AS Data,
    u.nazwa AS Utwor,
    a.nazwa AS Artysta,
    n.liczba_odezwan AS Liczba_Odezwan
FROM
    NajczesciejOdtwarzane n
JOIN
    Utwory u ON n.id_utworu = u.id_utworu
JOIN
    Album al ON u.album_id_album = al.id_album
JOIN
    Artysta a ON al.artysta_id_artyst = a.id_artyst
ORDER BY
    n.data DESC;
```

W polu P12\_date podajemy date dla której chcemy znaleźć najczęściej odtwarzany utwór w podanym dniu i zapisać go do NajczesciejOdtwarzane

Po wciśnięciu przycisku Generuj zadziała akcja dynamiczna

The screenshot shows the Oracle APEX developer interface. On the left, under 'Body' > 'Region Body' > 'Generuj' > 'Dynamic Actions' > 'New', there is a list of options: 'Execute Server-side Code' (highlighted in green), 'Refresh', and 'Alert'. The 'Execute Server-side Code' option is selected.

In the center, the 'FULL WIDTH CONTENT' editor shows a 'BODY' section containing a 'Najczęściej odtwarzane' report item. Below it is a 'REGION BODY' section with a 'P21\_DATE' report item and a 'Generuj' button.

On the right, the 'Settings' panel is open, showing:

- Language:** PL/SQL
- PL/SQL Code:**

```
BEGIN
StatystykiOdtworzen.ZapiszNajczesciejOdtwarzanyUtwor(:P21_DATE);
END;
```
- Items to Submit:** P21\_DATE
- Items to Return:** (empty)
- Execution:**
  - Sequence:** 10
  - Event:** New
  - Fire When Event Result Is:** True
  - Fire on Initialization:** (checkbox is off)

The screenshot shows the Oracle APEX application running. At the top, there is a search bar with 'Date' and '1/24/2025' selected, and a 'Generuj' button.

The main area displays a table with columns: Data, Utwor, Artysta, and Liczba Odtworzen. The data includes:

Data	Utwor	Artysta	Liczba Odtworzen
1/24/2025	death metalowy t-shirt	Chivas	1
1/24/2025	Intro	Golubenko	2
1/24/2025	narcyz w klubie	Chivas	1
1/24/2025	Lucid Dreams	Juice WRLD	1
1/24/2025	Runaway		1
1/24/2025	DO THE DANCE		1
1/24/2025	Intro		1
1/24/2025	PLAYDEAD		1
1/24/2025	młody say10	Chivas	1

A modal dialog box is displayed in the center, containing the message 'Utwór został dodany' (Song was added) and an 'OK' button.

#### **4. Strona Generowanie Raportów/Przycisk Raport Utworu**

Strona „Generowanie Raportów” zawiera formularz z reportem. Po wybraniu Utworu z wypadającej listy typu select list i wcisnięciu przycisku „Generate” wykonuje się procedura „GenerujRaportOdtworzen” z pakietu „StatystykiOdtworzen”.

Formularz jest stworzony na podstawie tabeli HistoriaOdtworzen

The screenshot shows the Power BI Report Builder interface. On the left, under the 'Body' section, there is a report item titled 'Generowanie raportów'. This item is expanded, revealing a list of measures:

- PI7\_ID\_ODTWORZENIA
- PI7\_ID\_UTWORU
- PI7\_ID\_UZYTKOWNIK
- PI7\_CZAS\_ODTWORZENIA
- PI7\_DATA\_ODTWORZENIA
- PI7\_LICZBA\_ODTWORZEN
- PI7\_SREDNI\_CZAS

Below this list is a 'Breadcrumb Bar' component. The breadcrumb path consists of the following items:

- UP
- REGION BODY
- REGION CONTENT
- SUB REGIONS

On the right side of the interface, there is a 'Type' ribbon tab selected. Under the 'Source' tab, the following properties are visible:

Location	Local Database
Type	Table / View
Table Owner	Parsing Schema
Table Name	HISTORIAODTWORZ
Include ROWID Column	<input checked="" type="checkbox"/>
Where Clause	<input type="button" value="..."/>

W formularzu wybieramy z listy wypadającej zdefiniowanej przez SQL QUERY utwór dla którego chcemy wygenerować raport

```
SELECT
    u.nazwa || ' (' || a.nazwa || ')' AS display_value,
    u.id_utworu AS return_value
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
ORDER BY
    u.nazwa;
```

Raport pod formularzem jest stworzony na podstawie tabeli RaportOdtworzen

```
Location: Local Database
Type: Table / View
Table Owner: Parsing Schema
Table Name: RAPORTODTWORZE
Include ROWID Column: Off
Where Clause:
```

Po wybraniu utworu i kliknięciu po przycisku „GENERATE” wykonuje się procedura i rekord zostaje zapisany do tabeli i wyświetlany na stronie

The screenshot shows the Oracle APEX application interface. On the left, there's a sidebar with navigation items like 'Processes' (highlighted), 'After Processing', 'Branches', and 'Ajax Callback'. A green button labeled 'GenerujReport' is visible. The main area has tabs for 'COPY', 'EDIT', 'PREVIOUS', 'NEXT', 'SORT ORDER', 'REGION BODY', and 'REGION CONTENT'. In the 'REGION BODY' tab, three regions are listed: 'P17\_ID\_UTWORU', 'P17\_LICZBA\_ODTWRZEN', and 'P17\_SREDNI\_CZAS'. The 'P17\_ID\_UTWORU' region is highlighted. On the right, a configuration panel for a process step named 'GenerujReport' is shown. It includes fields for 'Name' (GenerujReport), 'Type' (Execute Code), 'Execution Chain' (None), 'Editable Region' (- Select -), and a 'Source' section. The source code is written in PL/SQL:

```

BEGIN
    -- Wykonaj procedurę generującą raport
    StatystykiOdtworzen.GenerujReportOdtworzen(:P17_ID_UTWORU);
END;

```

Below the configuration panel, a green banner displays the message 'Raport został utworzony' (Report was generated) with a checkmark icon.

## Generowanie raportów

The main content area is titled 'Generowanie raportów'. It contains a dropdown menu with 'New' and 'Wybierz' options, a 'Cancel' button, and a 'Generate' button. Below these, a table displays the generated report data:

<b>Id Utworu</b>	<b>Nazwa Utworu</b>	<b>Nazwa Artysty</b>	<b>Liczba Odtworzeń</b>	<b>Sredni Czas</b>
1	anyżowe żelki	Chivas	13	83.07692307692307692307692307692307692307692307692307692308
39	W Krainie Czarów	Szpaku	11	78.27
107	anyżowe żelki	Chivas	8	62.75
117	anyżowe żelki	Chivas	8	94.5
118	Bilingual	ovgl	7	94.28571428571428571428571428571428571428571429

1 - 5

## 5. Strona Podsumowanie odtworzeń użytkownika/Przycisk Raport Użytkownika

# Podsumowanie odtworzeń użytkownika

Podsumowanie odtworzeń użytkownika

Id Użytkownik Alexander Lee (alex@example.com)	<input type="text"/> Go	Actions ▾																												
<input type="text"/>																														
Podsumowanie																														
Cancel	<b>Generuj</b>	Zapisz do Tabeli																												
Release 1.0																														
<table border="1"><thead><tr><th>id_uzytkownika</th><th>imie_uzytkownika</th><th>laczny_czas_odtworzen</th><th>najczesciej_odtwarzany_utwor</th></tr></thead><tbody><tr><td>13</td><td>James White</td><td>4331</td><td>odkąd nauczyłem się przeklinać</td></tr><tr><td>1</td><td>John Doe</td><td>5038</td><td>miałem kolegę bartka</td></tr><tr><td>9</td><td>Isabella Martinez</td><td>4336</td><td>Betrayal</td></tr><tr><td>7</td><td>Sophia Wilson</td><td>4278</td><td>Hinata</td></tr><tr><td>8</td><td>Oliver Garcia</td><td>3816</td><td>Black and White</td></tr><tr><td>17</td><td>Benjamin Scott</td><td>4240</td><td>drzewko</td></tr></tbody></table>			id_uzytkownika	imie_uzytkownika	laczny_czas_odtworzen	najczesciej_odtwarzany_utwor	13	James White	4331	odkąd nauczyłem się przeklinać	1	John Doe	5038	miałem kolegę bartka	9	Isabella Martinez	4336	Betrayal	7	Sophia Wilson	4278	Hinata	8	Oliver Garcia	3816	Black and White	17	Benjamin Scott	4240	drzewko
id_uzytkownika	imie_uzytkownika	laczny_czas_odtworzen	najczesciej_odtwarzany_utwor																											
13	James White	4331	odkąd nauczyłem się przeklinać																											
1	John Doe	5038	miałem kolegę bartka																											
9	Isabella Martinez	4336	Betrayal																											
7	Sophia Wilson	4278	Hinata																											
8	Oliver Garcia	3816	Black and White																											
17	Benjamin Scott	4240	drzewko																											

Strona „Podsumowanie odtworzeń użytkownika” zawiera formularz z reportem. Po wybraniu Użytkownika z wypadającej listy typu select list i wcisnięciu przycisku „Generuj” wykonuje się procedura „PodsumowanieOdtwarzaniaUzytkownika” z pakietu „StatystykiOdtworzen”.

Formularz jest stworzony na podstawie tabeli HistoriaOdtworzen

The screenshot shows the Oracle Forms Designer interface. On the left, the 'Body' region is expanded, showing various fields: P14\_ID\_ODTWRZENIA, P14\_ID\_UTWORU, P14\_ID\_UZYTKOWNIK, P14\_CZAS\_ODTWRZENIA, P14\_DATA\_ODTWRZENIA, P14\_LACZNY\_CZAS\_ODTWRZEN, P14\_NEW, and P14\_NAJCZESCIEJ\_ODTWARZANY. The 'P14\_ID\_UZYTKOWNIK' field is highlighted with a green border. In the center, there are navigation buttons: 'PREVIOUS', 'NEXT', 'SORT ORDER', and 'REGION BODY'. The 'REGION BODY' section contains the same list of fields. On the right, the 'Source' panel is open, showing the configuration for the 'P14\_ID\_UZYTKOWNIK' field. The 'Type' is set to 'Form', 'Location' to 'Local Database', 'Type' to 'Table / View', 'Table Owner' to 'Parsing Schema', and 'Table Name' to 'HISTORIAODTWRZI'. The 'Include ROWID Column' switch is off. The 'Where Clause' section is empty.

W formularzu wybieramy z listy wypadającej zdefiniowanej przez SQL QUERY użytkownika dla którego chcemy wygenerować raport

This screenshot shows the 'Body' region with the 'Report' item selected. The 'P14\_ID\_UZYTKOWNIK' field is highlighted with a green border. In the center, the 'List of Values' section is visible, with the 'Type' set to 'SQL Query'. The SQL query is:

```
SELECT imie || ' (' || email || ')' AS display_value, id_uzytkownik AS return_value
FROM Uzytkownik
ORDER BY imie;
```

The 'Display Extra Values' and 'Display Null Value' options are turned on. On the right, the 'Source' panel is shown, with the 'Type' set to 'Function Body' and 'Language' to 'PL/SQL'. The PL/SQL Function Body is defined as:

```
BEGIN
    :P14_NAJCZESCIEJ_ODTWARZANY := StatystykiOdtworzen.ZnajdzNajczesciejOdtwarzanyUtworUzytkownika(:P14_ID_UZYTKOWNIK);
    DBMS_OUTPUT.PUT_LINE('Najczęściej odtwarzany utwór: ' || :P14_NAJCZESCIEJ_ODTWARZANY);
END;
```

Pole P14\_najczesciej\_odtwarzany jest zdefiniowane przez funkcję „ZnajdzNajczesciejOdtwarzanyUtworUzytkownika” z pakietu „StatystykiOdtworzen”

This screenshot shows the 'Body' region with the 'Report' item selected. The 'P14\_NAJCZESCIEJ\_ODTWARZANY' field is highlighted with a green border. In the center, the 'Report' section is visible, showing 'SORT ORDER' and 'REGION BODY' buttons. On the right, the 'Source' panel is shown, with the 'Type' set to 'Function Body' and 'Language' to 'PL/SQL'. The PL/SQL Function Body is defined as:

```
BEGIN
    :P14_NAJCZESCIEJ_ODTWARZANY := StatystykiOdtworzen.ZnajdzNajczesciejOdtwarzanyUtworUzytkownika(:P14_ID_UZYTKOWNIK);
    DBMS_OUTPUT.PUT_LINE('Najczęściej odtwarzany utwór: ' || :P14_NAJCZESCIEJ_ODTWARZANY);
END;
```

The 'Used' dropdown is set to 'Always, replacing any existing value'.

Pod formularzem znajduje się report kolekcji zdefiniowany przez SQL Query gdzie będzie wyświetlany wygenerowany rekord

The screenshot shows the Oracle APEX application builder interface. On the left, a tree view lists items under 'P14\_ID\_UZYTKOWNIK'. Under 'Rapor' (Report), there are two columns: 'IMIE\_UZYTKOWNIKA' and 'NAJCZESCIEJ\_ODTWARZANY\_UTWOR'. In the center, a report region titled 'Szczegóły odtworzeń' is displayed with a search bar and a 'NEXT' button. On the right, the 'SQL Query' section contains the following code:

```

SELECT
    c001 AS ID_Uzytkownika,
    c002 AS Imie_Uzytkownika,
    c003 AS Laczny_Czas_Odtworzen,
    c004 AS Najczesciej_Odtwarzany_Utwor
FROM
    APEX_COLLECTIONS
WHERE
    COLLECTION_NAME =
        'PODSUMOWANIE_ODTWARZANIA';

```

Below the query are sections for 'Page Items to Submit' and 'Optimizer Hint'.

Dodatkowy report „Szczegóły odtworzeń” wyświetla wszystkie odtworzenia wybranego użytkownika

The screenshot shows the configuration of a report named 'Szczegóły odtworzeń'. The 'Source' tab is selected, showing the SQL query:

```

SELECT
    H.DATA_ODTWRZENIA AS Data,
    U.Nazwa AS Utwór,
    H.CZAS_ODTWRZENIA AS Czas_Odtwarzania
FROM
    HISTORIAODTWRZEN_H
JOIN UTWORY U ON H.ID_UTWORU = U.ID_UTWORU
WHERE
    H.ID_UZYTKOWNIK = :P14_ID_UZYTKOWNIK
ORDER BY H.DATA_ODTWRZENIA DESC;

```

On the left, the report structure is shown with 'Saved Reports' and 'Sub Regions' expanded. The 'Region Body' section contains the report itself. On the right, there are buttons for 'CLOSE', 'HELP', 'DELETE', 'CHANGE', and 'CREATE'.

Na dole strony znajduje się tabela HTML zdefiniowana przez poniższy kod

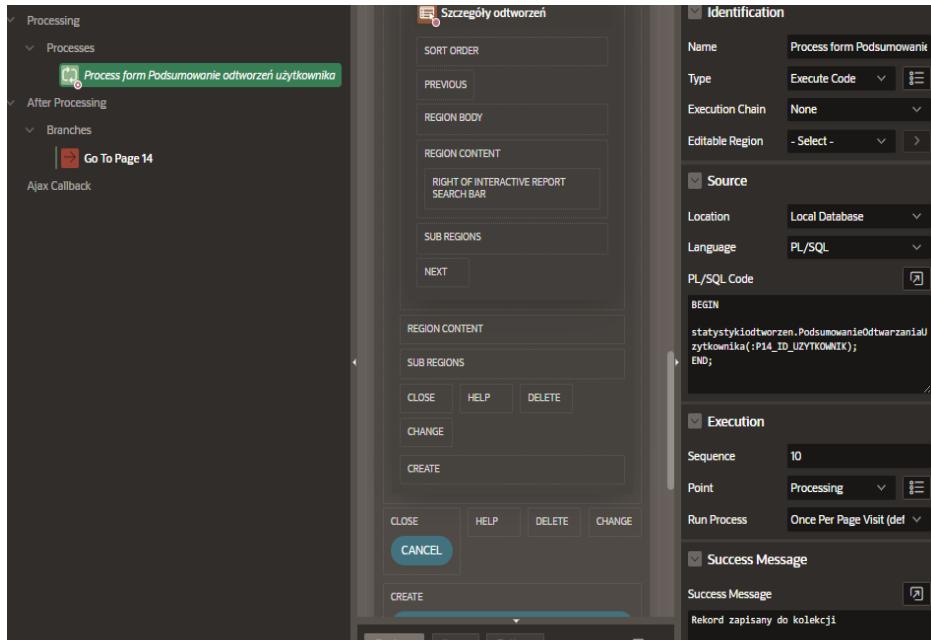
Code Editor - PL/SQL Function Body returning a CLOB

```

1  http.p('<style>
2  table, th, td {
3  border: 1px solid black;
4  border-collapse: collapse;
5  }
6
7  </style>');
8
9  http.p('<table style="width:100%">
10 <tr>
11 <th>id_uzytkownika</th>
12 <th>imie_uzytkownika</th>
13 <th>laczny_czas_odtworzen</th>
14 <th>najczesciej_odtwarzany_utwore</th>
15 </tr>');
16
17 for x in (select id_uzytkownika, imie_uzytkownika, laczny_czas_odtworzen, najczesciej_odtwarzany_utwore from podsumowanieodtwarzania)
18 loop
19 http.p('<tr>
20 <td>'||x.id_uzytkownika||'</td>
21 <td>'||x.imie_uzytkownika||'</td>
22 <td>'||x.laczny_czas_odtworzen||'</td>
23 <td>'||x.najczesciej_odtwarzany_utwore||'</td>
24 </tr>');
25 end loop;
26 http.p('</table>');

```

Po wybraniu Użytkownika dla którego chcemy wygenerować raport i wciśnięciu przycisku „Generuj” wykonuje się procedura a rekord zapisuje się do kolekcji i wyświetla się na stronie



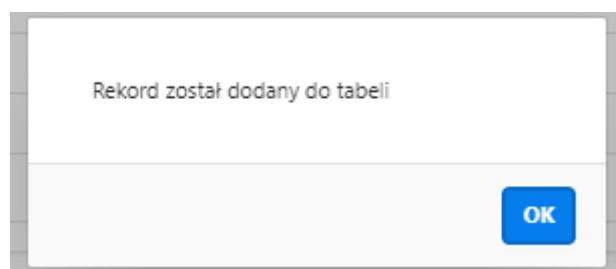
### Podsumowanie odtworzeń użytkownika

Id Użytkownik Alexander Lee (alex@example.com)			
Najczęściej odtwarzany utwór użytkownika			
Alexander Lee	FULL METAL SKIT	15	7118
1 - 1			
<b>Szczegóły odtworzeń</b>			
1/22/2025	zabij dla mnie		118
1/22/2025	xo		200
1/22/2025	DO THE DANCE		142
1/17/2025	Nevahomist		78
1/16/2025	Nieważne		134
1/16/2025	anyżowe żelki		187
1/13/2025	Nancy Mulligan		198
1/10/2025	FULL METAL SKIT		148
1/9/2025	UZUMAKI FORMA OSTATECZNA		154
1/9/2025	Used To		199
1/8/2025	comme des garçons		100

Po wciśnięciu przycisku „Zapisz do tabeli” uruchamia się akcja dynamiczna i zapisuje rekord z kolekcji do tabeli PODSUMOWANIEODTWORZEN

Code Editor - PL/SQL Code

```
1 BEGIN
2     FOR rec IN (
3         SELECT c001 AS ID_Uzytkownika,
4                c002 AS Imie_Uzytkownika,
5                c003 AS Laczny_Czas_Odtworzen,
6                c004 AS Najczesciej_Odtwarzany_Utwor
7         FROM APEX_COLLECTIONS
8        WHERE COLLECTION_NAME = 'PODSUMOWANIE_ODTWARZANIA'
9     ) LOOP
10        BEGIN
11            INSERT INTO PODSUMOWANIEODTWARZANIA (
12                ID_UZYTKOWNIKA,
13                IMIE_UZYTKOWNIKA,
14                LACZNY_CZAS_ODTWRZEN,
15                NAJCSZCIEJ_ODTWARZANY_UTWOR
16            ) VALUES (
17                TO_NUMBER(rec.ID_Uzytkownika),
18                rec.Imie_Uzytkownika,
19                TO_NUMBER(rec.Laczny_Czas_Odtworzen),
20                rec.Najczesciej_Odtwarzany_Utwor
21            );
22        EXCEPTION
23            WHEN DUP_VAL_ON_INDEX THEN
24                UPDATE PODSUMOWANIEODTWARZANIA
25                SET IMIE_UZYTKOWNIKA = rec.Imie_Uzytkownika,
26                    LACZNY_CZAS_ODTWRZEN = TO_NUMBER(rec.Laczny_Czas_Odtworzen),
27                    NAJCSZCIEJ_ODTWARZANY_UTWOR = rec.Najczesciej_Odtwarzany_Utwor
28                WHERE ID_UZYTKOWNIKA = TO_NUMBER(rec.ID_Uzytkownika);
29        END;
30    END LOOP;
31 END;
```



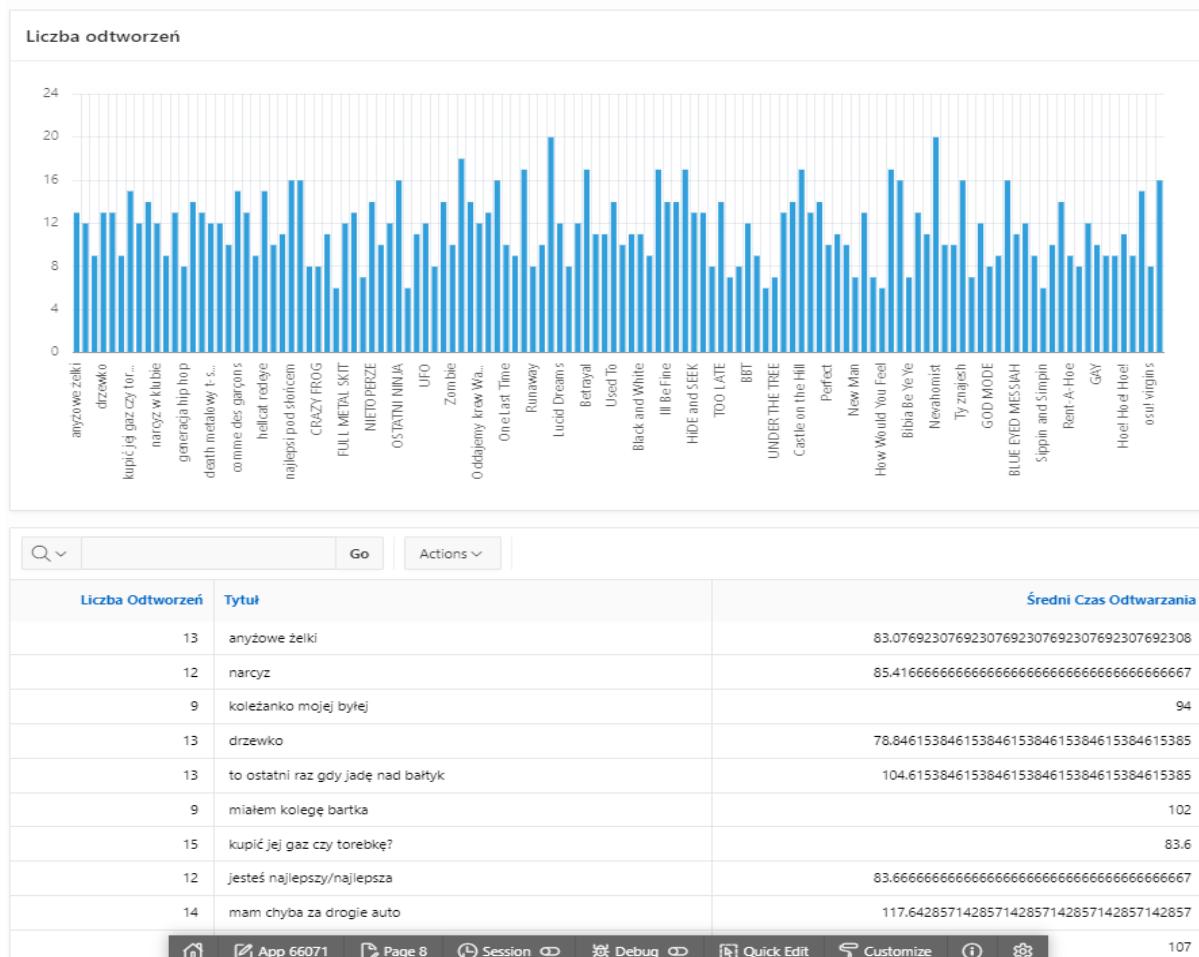
## Podsumowanie

Release 1.0

id_uzytkownika	imie_uzytkownika	lacyny_czas_odtworzen	najczesciej_odtwarzany_utwor
13	James White	4331	odkąd nauczyłem się przeklinać
1	John Doe	5038	miałem kolegę bartka
15	Alexander Lee	7118	FULL METAL SKIT
9	Isabella Martinez	4336	Betrayal
5	Eva Williams	6370	Perfect
7	Sophia Wilson	4278	Hinata
8	Oliver Garcia	3816	Black and White
17	Benjamin Scott	4240	drzewko

## **6. Strona Statystyki odtworzeń/Przycisk Statystyki Utworów**

### **Statystyki utworów**



Strona „Statystyki utworów” zawiera wykres z reportem. Wyświetla na wykresie ilość odtworzeń każdego utowaru. Raport wyświetla ilość odtworzeń tytułu i średni czas odtworzeń.

Wykres jest generowany za pomocą SQL Query, została użyta funkcja z pakietu „StatystykiOdtworzen”: „LiczbaOdtworzenUtworu”

The screenshot shows the SAP Fiori Launchpad interface. On the left, there's a navigation tree with items like 'Column Groups', 'Saved Reports', 'Region Body', and a selected item 'Liczba odtworzeń' (Number of Reproductions). Below it are 'Series' and 'Axes' sections. The 'Axes' section has two icons: 'x' and 'y'. In the center, there's a large modal window titled 'REGION CONTENT'. It contains a 'RIGHT OF INTERACTIVE REPORT SEARCH BAR' with fields for 'Label' (TYTUŁ) and 'Value' (LICZBA\_ODTWORZEŃ). At the bottom of this window are tabs for 'Regions', 'Items', and 'Buttons'. To the right of the modal, there's a vertical toolbar with buttons for 'COPY', 'EDIT', 'PREVIOUS', 'NEXT', 'SORT ORDER', 'REGION BODY', 'REGION CONTENT', 'SUB REGIONS', and 'CLOSE', 'HELP', 'DELETE' buttons. Below these are 'CREATE' and 'CHANGE' buttons. At the very bottom, there are icons for search, filter, sort, and other report functions. On the far right, there's a sidebar with sections for 'Location' (Local Database), 'Type' (SQL Query), 'SQL Query' containing the code: 'SELECT UTWORY.Nazwa AS Tytuł, StatystykIdOtworzony.LiczbaOdtworzenUtworzone(UTWORY.ID\_UTWORU) AS Liczba\_Odtworzeń FROM UTWORY;', 'Page Items to Submit', 'Optimizer Hint', 'Order By' (with a checked checkbox), 'Type' (None), and 'Column Mapping' (with a checked checkbox). The 'Column Mapping' section lists 'Series Name' as 'Select', 'Label' as 'TYTUŁ', 'Value' as 'LICZBA\_ODTWORZEŃ', and 'Custom Tooltip' and 'Legend Tooltip' both as 'Select'.

Raport jest generowany za pomocą SQL Query, zostały użyte funkcje z pakietu „StatystykiOdtworzen”:  
„LiczbaOdtworzenUtworów”, „SredniczasOdtworzenUtworów”.

The screenshot shows the BIRT Report Designer interface with three main panels:

- Breadcrumb:** Shows the report structure: Body > Statystyki utworów > Liczba odtworzeń.
- Report Structure:** A tree view of the report regions:
  - Body:**
    - Statystyki utworów:**
      - Columns
      - Column Groups
      - Saved Reports
      - Liczba odtworzeń:**
        - Liczba odtworzeń utworów:** Selected item.
        - Series
        - Axes
          - X
          - Y
  - Region Body:** A panel for the selected "Liczba odtworzeń" region, containing buttons for COPY, EDIT, PREVIOUS, NEXT, SORT ORDER, and REGION BODY.
  - SQL Editor:** A panel on the right showing the SQL query definition:

```
SELECT
    UTWORY.ID_UTWORU,
    UTWORY.Nazwa AS Tytuł,
    StatystykiOdtworzen.LiczbaOdtworzenUtworu(UTWORY.ID_UTWORU) AS Liczba_Odtworzeń,
    StatystykiOdtworzen.SredniczasOdtworzenUtworu(UTWORY.ID_UTWORU) AS Średni_Czas_Odtwarzania
FROM
```

## 5. Grupa stron dotyczących płatności i subskrypcji

### 1. Strona Rejestracja płatności

#### Rejestracja płatności

Szczegółowy Subskrypcji   Obliczanie Kosztów   Anuluj subskrypcję   Przy pominięciu

Rejestracja płatności				
<input type="text"/> Użytkownik <input type="text"/> Nowy Plan				
<input type="button" value="Go"/> Actions ▾				
Imię Użytkownika	Nazwa Planu Subskrypcyjnego	Data Start Subskrypcji	Data Koniec Subskrypcji	Status Subskrypcji
Alexander Lee	Student	8/9/2022	9/9/2022	AKTYWNA
Alice Johnson	Regular	6/19/2022	7/19/2022	AKTYWNA
Ava Thompson	Regular	7/24/2022	8/24/2022	AKTYWNA
Benjamin Scott	Premium	9/1/2022	10/1/2022	AKTYWNA
Bob Brown	Regular	7/9/2022	8/9/2022	AKTYWNA
Charlotte Green	Premium	9/3/2022	10/3/2022	AKTYWNA
Emma Clark	Regular	9/24/2022	10/24/2022	AKTYWNA
Eva Williams	Rodzinny	7/14/2022	8/14/2022	AKTYWNA
Grace Perez	Premium	11/20/2022	12/20/2022	AKTYWNA
Isabella Martinez	Student	9/1/2022	10/1/2022	AKTYWNA
James White	Regular	8/11/2022	9/11/2022	AKTYWNA
Jane Smith	Regular	2/9/2023	1/8/2025	AKTYWNA
John Doe	Student	1/1/2025	1/26/2025	AKTYWNA
Liam Rodriguez	Student	10/15/2022	11/15/2022	AKTYWNA
Lucas Martinez	Premium	9/8/2022	10/8/2022	AKTYWNA
Mia Hall	Student	8/12/2022	9/12/2022	AKTYWNA
Michael Davis	Rodzinny	7/14/2022	8/14/2022	AKTYWNA
Oliver Garcia	Rodzinny	11/18/2024	12/18/2024	AKTYWNA
Sophia Wilson	Rodzinny	7/14/2022	8/14/2022	AKTYWNA
Will Anderson	Regular	1/24/2025	2/24/2025	AKTYWNA

1 - 20

Cancel      App 66071      Page 9      Session      Debug      Quick Edit      Customize     

Strona „Rejestracja płatności” zawiera formularz z reportem. Po wybraniu Użytkownika z wypadającej listy typu select list, wybraniu planu subskrypcji z wypadającej listy typu select list i wcisnięciu przycisku „Opłać subskrypcje” wykonuje się procedura „OplacSubskrypcje” z pakietu „ZarzadzaniePlatnosciami”.

The screenshot displays the Oracle APEX application interface. At the top, there is a navigation bar with links for 'Szczegółowy Subskrypcji', 'Obliczanie Kosztów', 'Anuluj subskrypcję', and 'Przy pominięciu'. Below the navigation bar is a search bar with dropdowns for 'Użytkownik' and 'Nowy Plan'. Underneath the search bar is a table listing users with their subscription details. The table has columns: Imię Użytkownika, Nazwa Planu Subskrypcyjnego, Data Start Subskrypcji, Data Koniec Subskrypcji, and Status Subskrypcji. The table contains 20 rows of data. At the bottom of the table is a footer with the text '1 - 20'. Below the table is a toolbar with standard APEX buttons: 'Cancel', 'Home', 'App 66071', 'Page 9', 'Session', 'Debug', 'Quick Edit', 'Customize', 'Help', 'Star', 'List', 'Print', and 'Oplać subskrypcje' (Pay Subscription).

On the left side, there is a sidebar with a tree view showing the structure of the application. The 'Rejestracja płatności' node is expanded, showing its children: 'Region Body' and 'P9\_ID\_PLATNOSC', 'P9\_UZYTKOWNIK\_ID\_UZYTKOWNIK', 'P9\_METODA\_PLATNOSCI', 'P9\_DATA\_PLATNOSC', and 'P9\_KWOTA'. The 'P9\_UZYTKOWNIK\_ID\_UZYTKOWNIK' field is highlighted with a green border.

The main content area shows a 'Subskrypcje' report. It includes sections for 'SORT ORDER' (with 'PREVIOUS' and 'NEXT' buttons), 'REGION BODY', 'REGION CONTENT' (containing the text 'RIGHT OF INTERACTIVE REPORT SEARCH BAR'), and 'SUB REGIONS'. To the right of the report, there is a 'List of Values' configuration panel. It shows the 'Type' as 'SQL Query' and the SQL query:

```
SELECT imie || ' (' || email || ')' AS display_value, id_uzytkownik AS return_value
FROM Uzytkownik
ORDER BY imie;
```

Below this, another 'List of Values' configuration panel is shown, also with 'Type' set to 'SQL Query' and the SQL query:

```
SELECT nazwa AS display_value,
id_plan_subskrypcyjny AS return_value
FROM Plan_Subskrypcyjny
ORDER BY nazwa;
```

A toggle switch at the bottom of this panel is labeled 'Display Extra Values'.

Report jest generowany za pomocą SQL Query

The screenshot shows the Oracle APEX report builder interface. On the left, there's a tree view of the report structure under 'Body'. In the center, there's a preview window titled 'Subskrypcje' showing a grid of data with columns like 'REGION BODY', 'REGION CONTENT', and 'SUB REGIONS'. On the right, there's a 'SQL Query' editor window with the following code:

```
SELECT
    u.imie AS Imie_Uzytkownika,
    p.nazwa AS Nazwa_Planu_Subskrypcjnego,
    ups.data_start AS Data_Start_Subskrypcji,
    ups.data_koniec AS Data_Koniec_Subskrypcji,
    ups.status AS Status_Subskrypcji
FROM
    Uzytkownik u
```

Below the SQL editor are sections for 'Page Items to Submit' and 'Optimizer Hint'.

Na stronie znajdują się 4 przyciski, każdy przycisk po kliknięciu przekieruje do odpowiedniej strony.

The screenshot shows the Oracle APEX page builder interface. On the left, there's a tree view of the page structure under 'Breadcrumb Bar' and 'Body'. In the center, there's a preview window showing a page with a breadcrumb bar and a main body section. The body section contains a form with fields like 'P9\_UZYTKOWNIK\_ID\_UZYTKOWNIK', 'P9\_KWOTA', and 'P9\_NOWY\_PLAN'. On the right, there's a detailed configuration panel for a button:

Button Name	Szczegóły subskrypcji
Label	Szczegóły Subskrypcji
Layout	
Sequence	10
Region	Breadcrumb
Slot	Next
Appearance	
Button Template	Text with Icon
Hot	(checkbox)
Template Options	Use Template Defaults, Left
CSS Classes	(button)
Icon	(button)
Behavior	
Action	Redirect to Page
Target	Page 10
Warn on Unsaved Changes	Do Not Check
Database Action	- Select -
Requires Confirmation	(checkbox)

Po wybraniu utworu , planu subskrypcyjnego i kliknięciu po przycisku „Opłać subskrypcję” wykonuje się procedura i rekord zostaje zapisany do tabeli i wyświetlanym na stronie

The screenshot displays two main components of an Oracle APEX application:

- Process Configuration:** On the left, a tree view shows "Processes" expanded, with "Process form Rejestracja płatności" selected. This panel includes sections for "BANNER", "TOP NAVIGATION", "REGION BODY", "SUB REGIONS", and "BODY". The "BODY" section contains buttons for "Szczegóły subskrypcji", "Oblicz", "Anuluj", and "Przypomnienie".
- Code Editor:** On the right, the "Form Region" configuration is shown. It includes fields for "Name" (Process form Rejestracja płatności), "Type" (Form - Automat), "Execution Chain" (None), and "Form Region" (Rejestracja płatn). The "PL/SQL Code to Insert/Update/Delete" section contains the following PL/SQL code:
 

```
BEGIN
ZarzadzaniePlatnosciami.OpłacSubskrypcje(
    :p9_uzytkownik_id => :P9_UZYTKOWNIK,
    nowy_plan_id => :P9_NOWY_PLAN
);
END;
```
- Success Message:** Below the code editor, a success message "Subskrypcja została opłacona" is displayed.
- Registration Page:** At the bottom, the application's registration page is shown. The title is "Rejestracja płatności". The "BODY" section contains input fields for "Użytkownik" (containing "Nowy Plan Premium") and a table listing users with their subscription details. The table has columns: Imię Użytkownika, Nazwa Planu Subskrypcyjnego, Data Start Subskrypcji, Data Koniec Subskrypcji, and Status Subskrypcji. A row for "Sophia Wilson" is highlighted with a yellow background.
- Page Footer:** The footer includes links for "Cancel", "Home", "App 66071", "Page 9", "Session", "Debug", "Quick Edit", "Customize", and a blue button labeled "Opłać subskrypcję".

## 2. Strona Szczegóły subskrypcji/Przycisk Szczegóły subskrypcji

Szczegóły subskrypcji

New

Status Subskrypcji  
BRAK SUBSKRYPCJI

Czas Trwania  
0

Ile Dni Do Końca

Czy Płatność realizowana dzisiaj

Strona „Szczegóły subskrypcji” zawiera formularz stworzony na podstawie tabeli Użytkownik\_plan\_sybskrypcyjny. Na stronie zostały dodane dodatkowe objekty. Po wybraniu użytkownika z listy wypadającej select list wyświetla Status Subskrypcji, Czas Trwania subskrypcji, Ile dni do końca, Czy Płatność została zrealizowana dzisiaj.

Body

Szczegóły subskrypcji

Region Body

- P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK
- P10\_PLAN\_SUBSKRYPCYJNY\_ID\_PLAN\_SUBSKI
- P10\_DATA\_START
- P10\_DATA\_KONIEC
- P10\_STATUS
- P10\_STATUS\_SUBSKRYPCJI
- P10\_CZAS\_TRWANIA
- P10\_ILE\_DNI\_DO\_KOŃCA
- P10\_RESULT

AFTER LOGO      BEFORE NAVIGATION BAR      AFTER NAVIGATION BAR

TOP NAVIGATION

BREADCRUMB BAR

Breadcrumb

UP

REGION BODY

REGION CONTENT

SUB REGIONS

PREVIOUS    CLOSE    DELETE    HELP    CHANGE

EDIT    COPY    CREATE

Type: Form

Source: Local Database

Type: Table / View

Table Owner: Parsing Schema

Table Name: UZYTKOWNIK\_PLAN

Include ROWID Column: Off

Where Clause:

SQL Query:

```
SELECT imie || ' (' || email || ')' AS display_value, id_uzytkownik AS return_value
FROM Uzytkownik
ORDER BY imie;
```

Display Extra Values: On

Display Null Value: On

Null Display Value:

Body

Szczegóły subskrypcji

Region Body

- P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK
- P10\_PLAN\_SUBSKRYPCYJNY\_ID\_PLAN\_SUBSKI
- P10\_DATA\_START
- P10\_DATA\_KONIEC
- P10\_STATUS
- P10\_STATUS\_SUBSKRYPCJI
- P10\_CZAS\_TRWANIA
- P10\_ILE\_DNI\_DO\_KOŃCA
- P10\_RESULT

COPY

EDIT

PREVIOUS

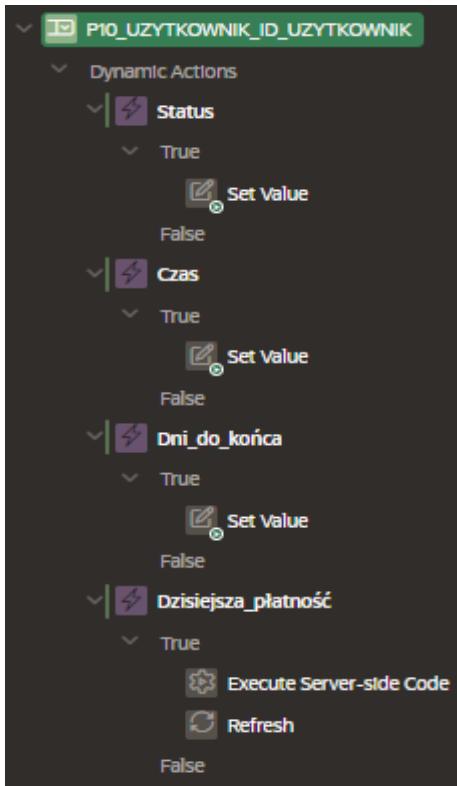
NEXT

SORT ORDER

REGION BODY

- P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK
- P10\_STATUS\_SUBSKRYPCJI
- P10\_CZAS\_TRWANIA

Zostały zdefiniowane funkcje dynamiczne:



Akcja Status: korzysta z funkcji „SprawdzStatusSubskrypcji” z pakietu ZarzadzaniePlatnosciami, zwraca wartość w polu P10\_status\_subskrypcji.

Szczegóły subskrypcji

- Region Body
- P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK
  - Dynamic Actions
    - Status
      - True
        - Set Value
      - False
    - Czas
      - True
        - Set Value
      - False

Szczegóły subskrypcji

Settings

Set Type: PL/SQL Function Body

```

BEGIN
  RETURN
  ZarzadzaniePlatnosciami.SprawdzStatusSubskrypcji(:P10_Uzytkownik_ID_Uzytkownika);
END;
  
```

Items to Submit: P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK

Escape Special Characters:

Suppress Change Event:

Akcja Czas: korzysta z funkcji „PobierzCzasTrwaniaSubskrypcji” z pakietu ZarzadzaniePlatnosciami, zwraca wartość w polu P10\_czas\_trwania.

Region Body

- P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK
  - Dynamic Actions
    - Status
    - Czas
      - True
        - Set Value
      - False

Settings

Set Type: PL/SQL Function Body

PL/SQL Function Body

```

BEGIN
  RETURN
  ZarzadzaniePlatnosciami.PobierzCzasTrwaniaSubskrypcji(:P10_Uzytkownik_ID_Uzytkownika);
END;
  
```

Items to Submit: P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK

Escape Special Characters:

Suppress Change Event:

Akcja Dni\_do\_Konca: korzysta z funkcji „IleDniDoKonca” z pakietu ZarzadzaniePlatnosciami zwraca wartość w polu P10\_ilе\_dni\_do\_konca.

The screenshot shows the Oracle APEX application builder interface. On the left, a tree view lists items under 'PL0\_UZYTKOWNIK\_ID\_UZYTKOWNIK'. Under 'Dynamic Actions', 'Status' and 'Czas' are listed. Under 'Dni do końca', 'True' has a 'Set Value' button. Under 'Dzisiajsza płatność', 'True' is selected. The main panel shows 'EDIT' mode with buttons for 'PREVIOUS', 'NEXT', and 'SORT ORDER'. The 'REGION BODY' contains three items: 'PL0\_UZYTKOWNIK\_ID\_UZYTKOWNIK' (with a dropdown icon), 'PL0\_STATUS\_SUBSKRYPACJI' (with a list icon), and 'PL0\_CZAS\_TRWANIA' (with a list icon). On the right, the 'Set Type' is 'PL/SQL Function Body' and the function body code is:

```
BEGIN
  RETURN
  ZarzadzaniePlatnosciami.IleDniDoKonca(:PL0_UZYTKOWNIK_ID_UZYTKOWNIK);
END;
```

On the far right, there are buttons for 'Items to Submit' (labeled 'PL0\_UZYTKOWNIK\_ID\_UZYTKOWNIK'), 'Escape Special Characters' (green switch), and 'Suppress Change Event' (grey switch).

Akcja Dzisiejsza\_płatność: korzysta z funkcji „CzyPlatnoscZrealizowana” z pakietu ZarzadzaniePlatnosciami zwraca wartość w polu P10\_Result.

P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK

Dynamic Actions

- Status
- Czas
- Dni do końca
- Dzisiejsza płatność
  - True

Execute Server-side Code

Refresh

False

P10\_PLAN\_SUBSKRYPCYJNY\_ID\_PLAN\_SUBSKI

COPY

EDIT

PREVIOUS

NEXT

SORT ORDER

REGION BODY

P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK

P10\_STATUS\_SUBSKRYPCJI

P10\_CZAS\_TRWANIA

Language PL/SQL

PL/SQL Code

```
BEGIN
  IF
    zarzadzanieplatnosciami.CzyPlatnoscZrealizowana(:P10_uzytkownik_ID_UZYTKOWNIK) THEN
      :P10_result := 'TAK';
    ELSE
      :P10_RESULT := 'NIE';
    END IF;
END;
```

Items to Submit P10\_UZYTKOWNIK\_ID\_UZYTKOWNIK

Items to Return P10\_RESULT

Suppress Change Event

Po wybraniu użytkownika wyświetla Status Subskrypcji, Czas Trwania subskrypcji, Ile dni do końca, Czy Płatność została zrealizowana dzisiaj.

### 3. Strona Obliczanie kosztów/Przycisk Oblicz

#### Obliczanie kosztów

Obliczanie kosztów

Plan subskrypcyjny

Kwota

Start Date

End Date

Cancel

Strona obliczanie kosztów zawiera formularz stworzony na podstawie tabeli Plan\_subskrypcji. Po wybraniu planu subskrypcyjnego z listy wypadającej select list, daty rozpoczęcia i daty końca subskrypcji generuje koszt subskrypcji.

Obliczanie kosztów

Region Body

P11\_ID\_PLAN\_SUBSKRYPCYJNY

P11\_NAZWA

P11\_WARTOSC

P11\_OPIS

P11\_KWOTA

P11\_START\_DATE

P11\_END\_DATE

Source

Location: Local Database

Type: Table / View

Table Owner: Parsing Schema

Table Name: PLAN\_SUBSKRYPCYJNY

Include ROWID Column: Off

Where Clause:

List of Values

Type: SQL Query

SQL Query:

```
SELECT nazwa AS display_value,
       id_plan_subskrypcyjny AS return_value
  FROM Plan_Subskrypcyjny
 ORDER BY id_plan_subskrypcyjny;
```

Display Extra Values: On

Display Null Value: On

Została zdefiniowana akcja dynamiczna „Koszt”, korzysta z funkcji „obliczkwotezaokres” z pakietu ZarzadzaniePlatnosciami, pobiera dane wprowadzone w polach P11\_START\_DATE, P11\_END\_DATE, P11\_ID\_PLAN\_SUBSKRYPCYJNY.

Region Body

P11\_ID\_PLAN\_SUBSKRYPCYJNY

P11\_NAZWA

P11\_WARTOSC

P11\_OPIS

P11\_KWOTA

Dynamic Actions

Koszt

True

Set Value

False

PL/SQL Function Body

```
BEGIN
  RETURN;
  ZarzadzaniePlatnosciami.ObliczKwoteZaOkres(
    :P11_START_DATE,
    :P11_END_DATE,
    :P11_ID_PLAN_SUBSKRYPCYJNY
  ) || ' PLN';
END;
```

Items to Submit: P11\_ID\_PLAN\_SUBSKRYPCYJNY

Escape Special Characters: On

Po wybraniu planu subskrypcyjnego z listy wypadającej select list, daty rozpoczęcia i daty końca subskrypcji generuje koszt subskrypcji w polu Kwota.

Plan subskrypcyjny  
Rodzinny

Kwota  
30.5 PLN

Start Date  
1/29/2025

End Date  
3/31/2025

Cancel

#### 4. Strona Anuluj subskrypcje/Przycisk Anuluj Subskrypcje

#### Anuluj subskrypcje

Anuluj subskrypcje

Użytkownik

Cancel

Anuluj

Strona Anuluj subskrypcje zawiera formularz stworzony na podstawie tabeli Uzytkownik\_plan\_subskrypcyjny. Po wybraniu Użytkownika z listy wypadającej select list i wcisnięciu przycisku Anuluj wykonuje się procedura „AnulujSubskrypcje” z pakietu ZarzadzniePlatnosciami.

Body

Anuluj subskrypcje

Region Body

- P12\_UZYTKOWNIK\_ID\_UZYTKOWNIK
- P12\_PLAN\_SUBSKRYPCYJNY\_ID\_PLAN\_SUBSKR
- P12\_DATA\_START
- P12\_DATA\_KONIEC
- P12\_STATUS

Close

CANCEL

Delete

FULL WIDTH CONTENT

BODY

Anuluj subskrypcje

COPY

EDIT

PREVIOUS

NEXT

SORT ORDER

Type: Form

Source: Local Database

Type: Table / View

Table Owner: Parsing Schema

Table Name: UZYTKOWNIK\_PLAN

Include ROWID Column: Off

Where Clause:

**List of Values**

Type: SQL Query

```
SELECT IMIE || ' ' || EMAIL AS DISPLAY_VALUE, ID_UZYTKOWNIK AS RETURN_VALUE
FROM UZYTKOWNIK
ORDER BY IMIE;
```

Display Extra Values: On

Display Null Value: On

Po wybraniu Użytkownika z listy wypadającej select list i wcisnięciu przycisku Anuluj zmienia status subskrypcji użytkownika na „Anulowana” i ustawia jako datę końca bieżącą datę. Przekierowuje na stronę Rejestracja płatności żeby sprawdzić w tabeli czy naprawdę status jest zmieniony.

Imię Użytkownika	Nazwa Planu Subskrypcyjnego	Data Start Subskrypcji	Data Koniec Subskrypcji	Status Subskrypcji
Alexander Lee	Student	8/9/2022	9/9/2022	AKTYWNA
Alice Johnson	Regular	6/19/2022	7/19/2022	AKTYWNA
Ava Thompson	Regular	7/24/2022	8/24/2022	AKTYWNA
Benjamin Scott	Premium	9/1/2022	10/1/2022	AKTYWNA
Bob Brown	Regular	7/9/2022	8/9/2022	AKTYWNA
Charlotte Green	Premium	9/3/2022	10/3/2022	AKTYWNA
Emma Clark	Regular	9/24/2022	10/24/2022	AKTYWNA
Eva Williams	Rodzinny	7/14/2022	8/14/2022	AKTYWNA
Grace Perez	Premium	11/20/2022	12/20/2022	AKTYWNA
Isabella Martinez	Student	9/1/2022	10/1/2022	AKTYWNA
James White	Regular	8/11/2022	9/11/2022	AKTYWNA
Jane Smith	Regular	2/9/2023	1/8/2025	AKTYWNA
John Doe	Student	1/1/2025	1/26/2025	AKTYWNA
Liam Rodriguez	Student	10/15/2022	11/15/2022	AKTYWNA
Lucas Martinez	Premium	9/8/2022	10/8/2022	AKTYWNA
Mia Hall	Student	8/12/2022	9/12/2022	AKTYWNA
Michael Davis	Rodzinny	7/14/2022	8/14/2022	AKTYWNA
Oliver Garcia	Rodzinny	11/18/2024	12/18/2024	AKTYWNA
Sophia Wilson	Premium	1/29/2025	1/29/2025	ANULOWANA
Will Anderson	Regular	1/24/2025	2/24/2025	AKTYWNA

## 5. Strona Przypomnienie/Przycisk Przypomnienie

### Przypomnienie

Wyslij przypomnienie

Liczba Użytkownikow  
13

Actions					
Id Użytkownika	Imię Użytkownika	Plan Subskrypcyjny	Data Rozpoczecia	Data Zakończenia	Status
3	Alice Johnson	Regular	6/19/2022	2/19/2022	AKTYWNA
12	Ava Thompson	Regular	7/24/2022	8/24/2022	AKTYWNA
15	Alexander Lee	Student	8/9/2022	9/9/2022	AKTYWNA
13	James White	Regular	8/11/2022	9/11/2022	AKTYWNA
16	Mia Hall	Student	8/12/2022	9/12/2022	AKTYWNA
9	Isabella Martinez	Student	9/1/2022	10/1/2022	AKTYWNA
17	Benjamin Scott	Premium	9/1/2022	10/1/2022	AKTYWNA
18	Charlotte Green	Premium	9/3/2022	10/3/2022	AKTYWNA
19	Lucas Martinez	Premium	9/8/2022	10/8/2022	AKTYWNA
14	Emma Clark	Regular	9/24/2022	10/24/2022	AKTYWNA
10	Liam Rodriguez	Student	10/15/2022	11/15/2022	AKTYWNA
20	Grace Perez	Premium	11/20/2022	12/20/2022	AKTYWNA
8	Oliver Garcia	Rodzinny	11/18/2024	12/18/2024	AKTYWNA

1 - 13

Actions		
Data Log	Email	Komunikat
1/24/2025	johndoe@example.com	Przypomnienie wyslane. Data zakończenia: 2025-01-26

Strona Przypomnienie zawiera 2 reporty i dodatkowy item „Liczba Użytkowników”.

Pierwszy report jest wygenerowany przy użyciu SQL Query, wyświetla użytkowników których subskrypcja ma mniej niż 3 dni do końca lub już się skończyła i użytkownik ma status subskrypcji „Aktywna”.

```

SELECT
    u.id_uzytkownik,
    u.imie AS imie_uzytkownika,
    ps.nazwa AS plan_subskrypcyjny,
    ups.data_start AS data_rozpoczecia,
    ups.data_koniec AS data_zakonczenia,
    ups.status
FROM
    uzytkownik_plan_subskrypcyjny ups
JOIN
    uzytkownik u ON u.id_uzytkownik = ups.uzytkownik_id_uzytkownik
JOIN
    plan_subskrypcyjny ps ON
    ps.id_plan_subskrypcyjny =
    ups.plan_subskrypcyjny_id_plan_subskrypcyjny
WHERE
    ups.data_koniec - TRUNC(SYSDATE) <= 3
    AND ups.status = 'AKTYWNA'
ORDER BY
    ups.data_koniec;

```

Została zdefiniowana akcja dynamiczna „Wysłać”, korzysta z procedury „PrzypomnijOZakonczeniuSubskrypcji” z pakietu ZarzadzaniePlatnosciami.

The screenshot shows the Oracle APEX report builder interface. In the left sidebar under 'Body', there is a 'Dynamic Actions' item named 'Wyślij'. The 'Action' panel on the right is configured for 'Execute Server-side Code' in PL/SQL. The code is:

```

BEGIN
  ZarzadzaniePlatnosciami.PrzypomnijOZakonczeniuSubskrypcji;
END;

```

The 'Execution' section shows 'Sequence' set to 10 and 'Event' set to 'Wysłać'.

Drugi report jest wygenerowany na podstawie tabeli LogPrzypomnienia.

The screenshot shows the Oracle APEX report builder interface. In the left sidebar under 'Body', there is a 'Sub Regions' item named 'Wiadomość'. The 'Action' panel on the right is configured for 'Table / View' with 'Table Name' set to 'LOG\_PRZYPOMNIENI'. The 'Where Clause' field is empty.

Dodatkowy item jest zdefiniowany przy użyciu PL/SQL functional body, zlicza ile użytkowników mają subskrypcję która ma mniej niż 3 dni do końca lub już się skończyła i użytkownik ma status subskrypcji „Aktywna”.

The screenshot shows the Oracle APEX report builder interface. In the left sidebar under 'Body', there is a 'Sub Regions' item named 'LICZBA\_UZYTKOWNIKOW'. The 'Action' panel on the right shows the PL/SQL Function Body:

```

DECLARE
  liczba_uzytkownikow NUMBER;
BEGIN
  SELECT COUNT(*)
  INTO liczba_uzytkownikow
  FROM UZYTKOWNIK_PLAN_SUBSKRYPCYJNY
  WHERE DATA_KONIEC = TRUNC(SYSDATE) + 3
    AND STATUS = 'AKTYWNA';

APEX_UTIL.SET_SESSION_STATE('LICZBA_UZYTKOWNIKOW', liczba_uzytkownikow);
end;

```

Po kliknięciu przycisku „Wyślij” wykonuje się akcja dynamiczna i do tabeli Log\_przypomnienia zapisuje email użytkownika, wiadomość kiedy się kończy subskrypcja i data wysłania.

Przypomnienie zostało wysłane

OK

## Przypomnienie

Wyślij przypomnienie

Liczba Użytkowników  
1

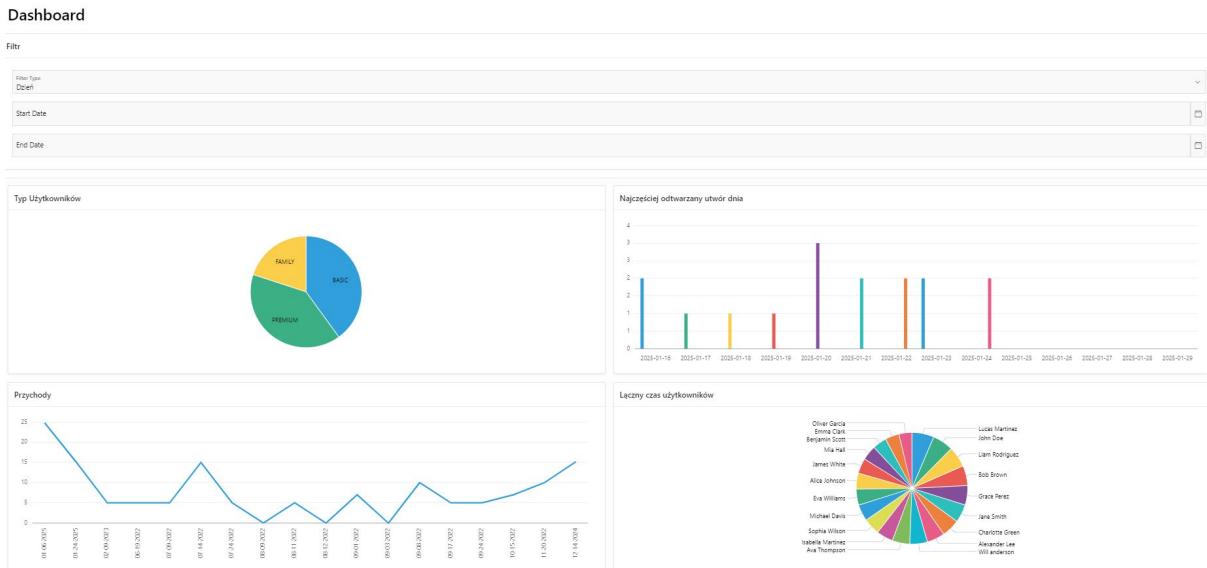
Q ▾		Go	Actions ▾			
Id Użytkownik	Imię Użytkownika	Plan Subskrypcyjny	Data Rozpoczecia	Data Zakonczenia	Status	
20	Grace Perez	Premium	11/20/2022	1/31/2025	AKTYWNA	1 - 1

Q ▾		Go	Actions ▾	X
<input type="checkbox"/>	<input checked="" type="checkbox"/>  Data Log >= 1/29/2025			
Data Log	Email	Komunikat		

1/29/2025 grace@example.com Przypomnienie wysłane. Data zakończenia: 2025-01-31

## 6. Strony dodatkowe

### 1. Dashboard



Strona dashboard została wygenerowana automatycznie podczas tworzenia aplikacji

### 1. Chart Typ Użytkowników

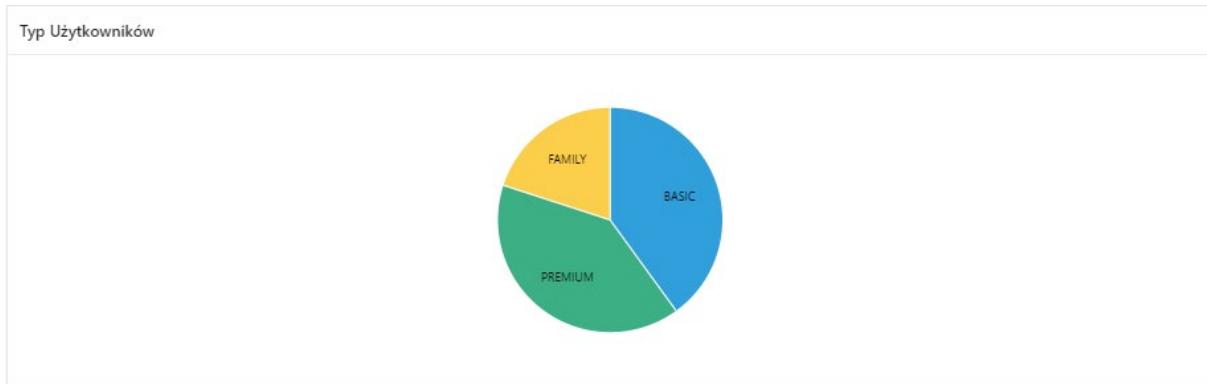
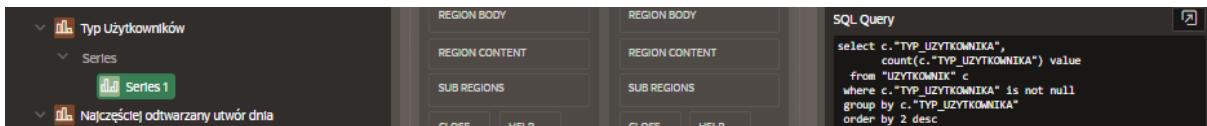


Chart Typ Użytkowników został wygenerowany przy użyciu SQL Query i wyświetla ile użytkowników każdego typu jest w bazie danych.



## 2. Chart Najczęściej odtwarzany utwór dnia

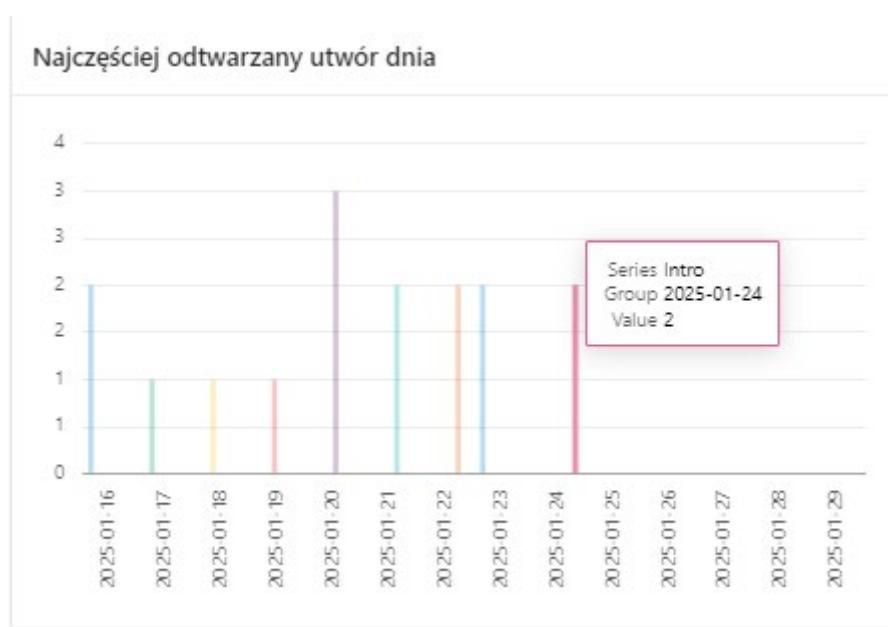


Chart Najczęściej odtwarzany utwór dnia został wygenerowany przy użyciu SQL Query, została użyta funkcja „NajczesciejOdtwarzanyUtwor\_dnia” z pakietu StatystykiOdtworzen, wyświetla najczęściej odtwarzany utwory, nazwę , ilość odtworzeń w ostatnich dwóch tygodniach.

### 3. Chart Przychody

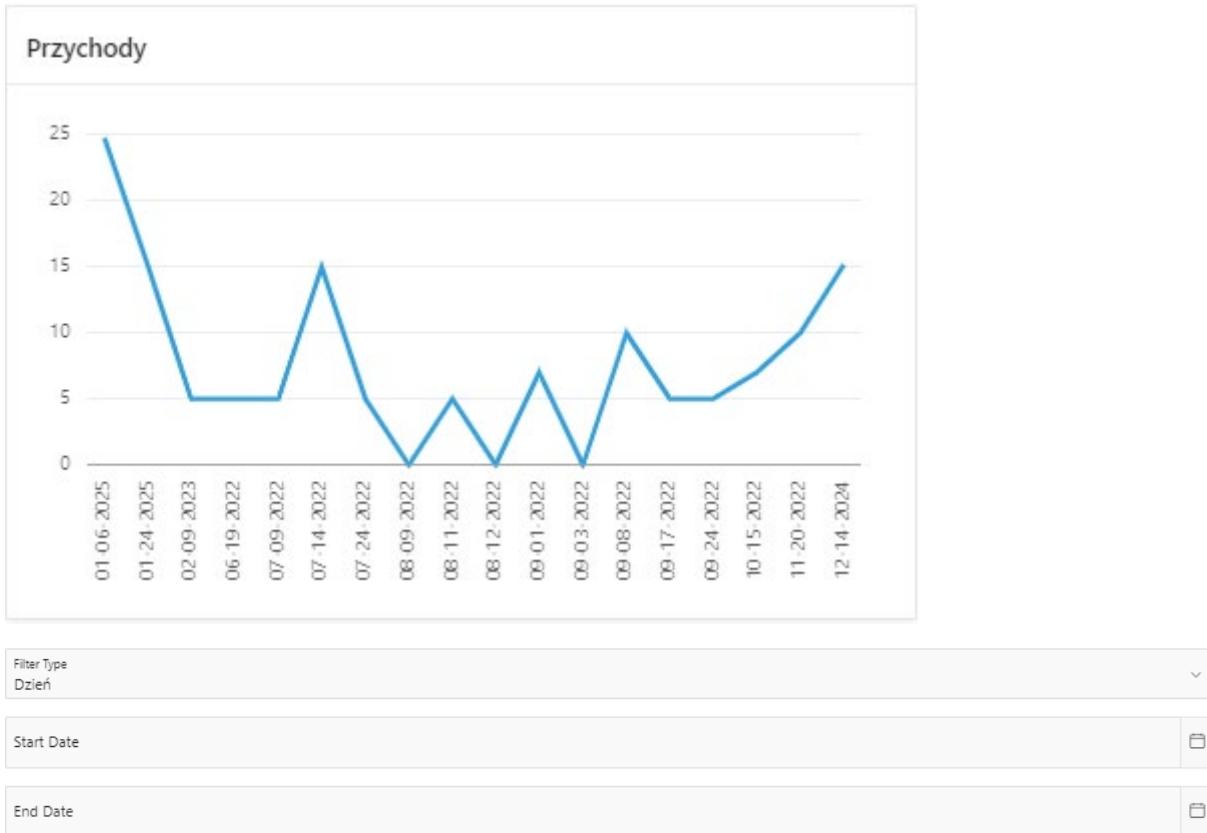


Chart Przychody został wygenerowany przy użyciu SQL Query, dodatkowo został dodany filtr dla podanego wykresu, możliwość zmiany jednostki czasowej i przedziału czasowego, wyświetla na jaką kwotę zostały zrealizowane płatności.

The screenshot shows the Oracle APEX application's page structure and the generated SQL query. The page structure includes a sidebar with components like "Body", "Typ Użytkowników", "Najczęściej odwiedzany utwór dnia", "Przychody", and "Chart 4". The "Przychody" section is currently active. The SQL query is as follows:

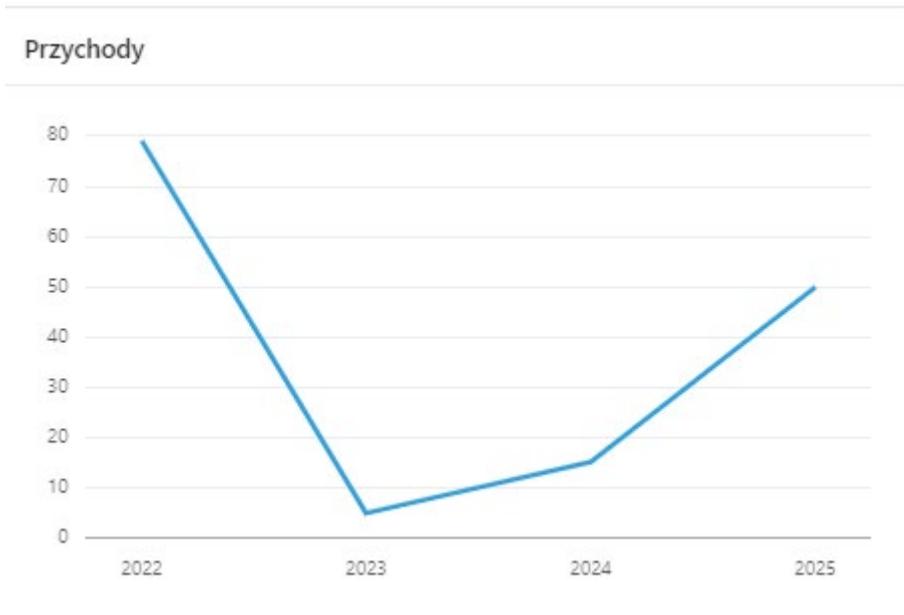
```

SQL Query
SELECT
    TO_CHAR(DATA_PLATNOSC,
        CASE :P2_FILTER_TYPE
            WHEN 'DAY' THEN 'MM-DD-YYYY'
            WHEN 'MONTH' THEN 'MM-YYYY'
            WHEN 'YEAR' THEN 'YYYY'
        END
    ) AS LABEL,
    SUM(KWOTA) AS VALUE
FROM PLATNOSC
WHERE DATA_PLATNOSC BETWEEN
    TO_DATE(NVL(:P2_START_DATE, '01-01-2000'), 'MM-DD-YYYY')
    AND
    TO_DATE(NVL(:P2_END_DATE, SYSDATE), 'MM-DD-YYYY')
GROUP BY TO_CHAR(DATA_PLATNOSC,
    CASE :P2_FILTER_TYPE
        WHEN 'DAY' THEN 'MM-DD-YYYY'
        WHEN 'MONTH' THEN 'MM-YYYY'
        WHEN 'YEAR' THEN 'YYYY'
    END
)
ORDER BY LABEL;

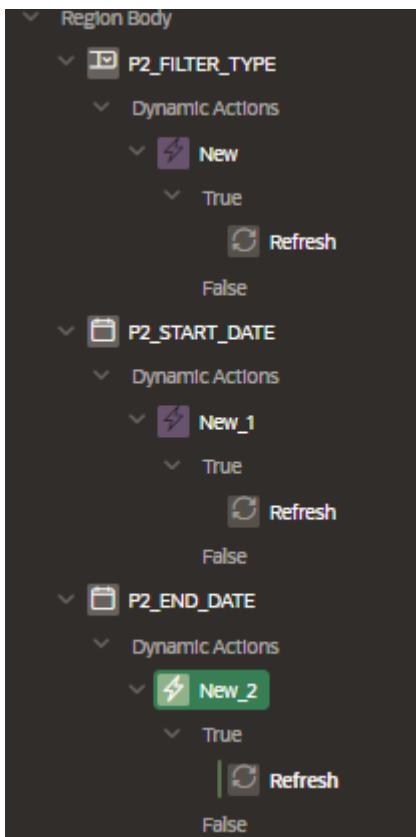
```

The screenshot shows the Oracle APEX application's components and static values configuration. The components pane lists "Breadcrumb Bar", "Filtr", "Region Body", "Dynamic Actions", and "P2\_START\_DATE". The "Filtr" component is expanded, showing "P2\_FILTER\_TYPE" selected. The static values configuration for "P2\_FILTER\_TYPE" is shown in the "Static Values" dialog, mapping "Dzień" to "DAY", "Miesiąc" to "MONTH", and "Rok" to "YEAR".

Filter Type	Rok	▼
Start Date	1/01/2022	<input type="button" value=""/>
End Date	4/30/2025	<input type="button" value=""/>



Na stronie zostały zdefiniowane kilka akcji dynamicznych do dynamicznego odświeżania wykresu



#### 4. Chart Łączny czas użytkownika

Łączny czas użytkowników

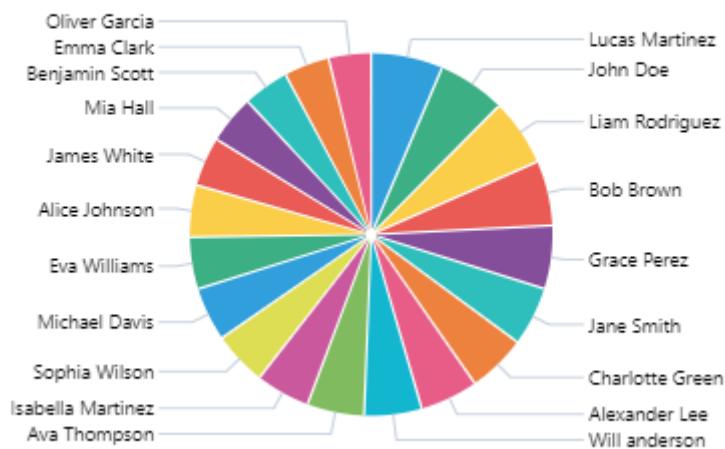


Chart Łączny czas użytkownika został wygenerowany przy użyciu SQL Query, korzysta z funkcji „LacznyCzasOdtwarzaniaUzytkownika” z pakietu StatystykiOdtworzen. Wyświetla ogólny czas odtwarzania każdego użytkownika.

The screenshot shows a software interface with three main panels. The left panel displays the chart with a legend entry for "Series 1". The middle panel shows two cards: "Przychody" and "Łączny czas użytkownika", both with standard UI controls like COPY, EDIT, PREVIOUS, NEXT, and SORT ORDER. The right panel is a SQL Query editor window containing the following code:

```
SQL Query
SELECT
    UZYTKONIK.IMIE AS IMIE_UZYTKONIKA,
    StatystykiOdtworzen.LacznyCzasOdtwarzaniaUzytkownika(UZYTKONIK.ID_UZYTKONIK) AS LACZNY_CZAS_ODTWRZEN
FROM
    UZYTKONIK
ORDER BY
    LACZNY_CZAS_ODTWRZEN DESC;
```

## 2. Strona Zarządzanie historią odtworzeń/Przycisk Zarządzanie historią odtworzeń

### Zarządzanie historią odtworzeń

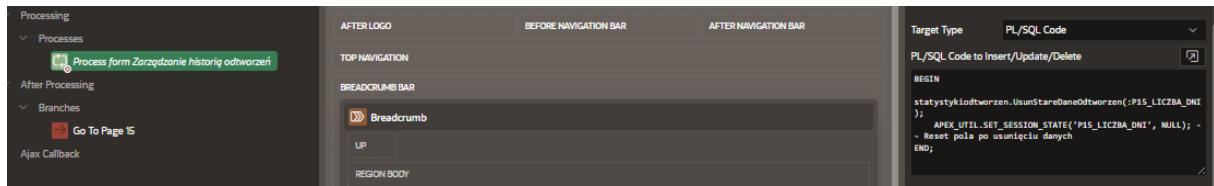
Zarządzanie historią odtworzeń				
Liczba Dni				
<input type="button" value="Usun"/> <span style="margin-left: 10px;"><input type="button" value="Go"/></span> <span style="margin-left: 10px;">Actions ▾</span>				
Id Odtworzenia	Id Utworu	Id Użytkownika	Data Odtworzenia	
421	9	9	1/20/2024	
324	18	19	1/20/2024	
841	50	17	1/21/2024	
582	18	18	1/21/2024	
952	24	1	1/22/2024	
394	34	10	1/22/2024	
253	45	1	1/23/2024	
457	92	16	1/23/2024	
343	25	19	1/24/2024	
615	25	20	1/24/2024	
897	47	20	1/25/2024	

Zarządzanie historią odtworzeń zawiera report wygenerowany przy użyciu SQL Query, wyświetla Id odtworzenia, id utworu, id użytkownika i datę odtworzenia. Został dodany dodatkowy item P15\_Liczba\_dni typu numeric field.

```

SELECT ID_ODTWRZENIA, ID_UTWORU, ID_UZYTKOWNIK, DATA_ODTWRZENIA
FROM HistoriaOdtworzen
ORDER BY DATA_ODTWRZENIA DESC;
    
```

Po wpisaniu Liczby dni i wciśnięciu przycisku usuń wykonuje się procedura „UsunStareDaneOdtworzen” z pakietu StatystykiOdtworzen i usuwa rekordy starsze niż podana liczba dni.



Po wpisaniu 369:

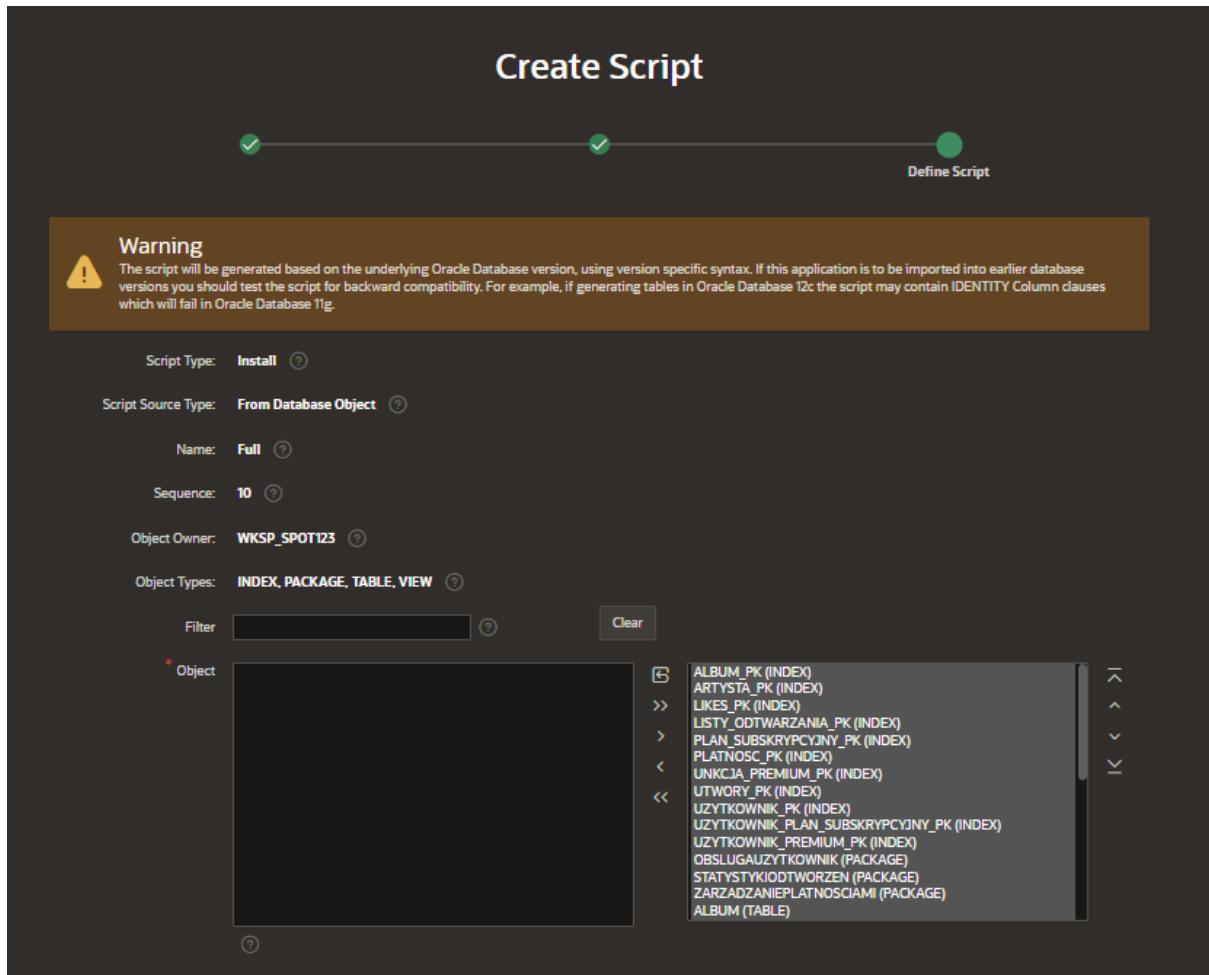
Id Odtworzenia	Id Utworu	Id Użytkownik	Data Odtworzenia
659	51	3	1/26/2024
629	114	10	1/26/2024
503	10	12	1/26/2024
51	11	15	1/26/2024
396	13	6	1/26/2024
140	69	12	1/26/2024
706	4	17	1/26/2024
866	70	10	1/26/2024
576	49	11	1/26/2024
512	1	5	1/27/2024
788	42	7	1/27/2024
298	45	13	1/27/2024

## 7. Export aplikacji

W tym punkcie utworzona aplikacja bazodanowa zostanie wyeksportowana a następnie ponownie zaimportowana przy pomocy pliku z rozszerzeniem .sql do środowiska APEX do nowo utworzonej przestrzeni roboczej (workspace).

Przed wykonaniem exportu do aplikacji przypisane zostaną skrypty SQL by w trakcie importu wszystkie tabele, procedury, funkcje i pakiety zostały zaimportowane razem z aplikacją.

Tworzenie skryptu



# Create Script

Script Type: **Install** [?](#)

Script Source Type: **Data Package** [?](#)

\* Name: **Data** [?](#)

\* Sequence: **20** [?](#)

Object Owner: **WKSP\_SPOT123** [?](#)

Filter:  [?](#)

Tables:

ALBUM
ARTYSTA
>> FUNKCJA_PREMIUM
HISTORIA_ODTWARZEN
> HTMLDB_PLAN_TABLE
LIKES
< LISTY_ODTWARZANIA
<< LOG_PRZYPOMNIENIA
OBSERWUJACY

## Export Application

Application	Name	Pages	Updated By	Updated	Owner
06071	Administracja Spotify	45	ANDRIY28122002@GMAIL.COM	3 minutes ago	WKSP_SPOT123

Readable Format  [?](#)

Build Status Override: Run and Build Application [?](#)

Supporting Object Definitions: Yes and Install on Import Automatically [?](#)

Split into Multiple Files  [?](#)

Public Reports  [?](#)

Private Reports  [?](#)

Report Subscriptions  [?](#)

Developer Comments  [?](#)

Translations  [?](#)

Debugging  [?](#)

Original IDs  [?](#)

Owner Override:  [?](#)

Audit Information: **None** [Names and Dates](#) [Dates Only](#) [?](#)

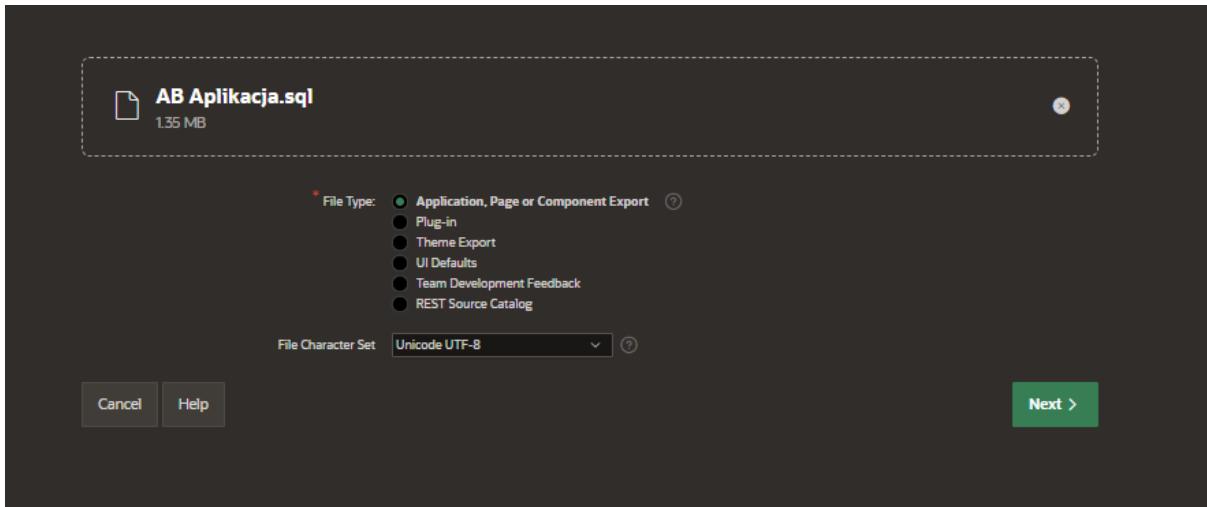
As of:  minutes ago [?](#)

File Character Set: **Unicode UTF-8** [?](#)

Actions: Messages Prerequisites Substitutions Build Options Validations **Install** Upgrade Deinstall Export

Lock	Name	Sequence	Script
✓	full	10	CREATE TABLE "ARTYSTA" ("ID_ARTYST" NUMBER(*,0) NOT NULL ENABLE, "NAZWA" VARCHAR2(50) NOT NULL ENABLE, "GATUNEK" VARCHAR2(50 CHAR), CONSTRAINT "ARTYSTA_PK" PRIMARY KEY ("ID_ARTYST")) ...
✓	data	20	begin --ARTYSTA: 10/10000 rows exported. APEX\$DATA\$PKG/ARTYSTA\$57745 apex_data_install.load_supporting_object_data:p_table_name => 'ARTYSTA'; p_delete_after_install => true } ...

Importujemy aplikacje na nowym workspace'sie



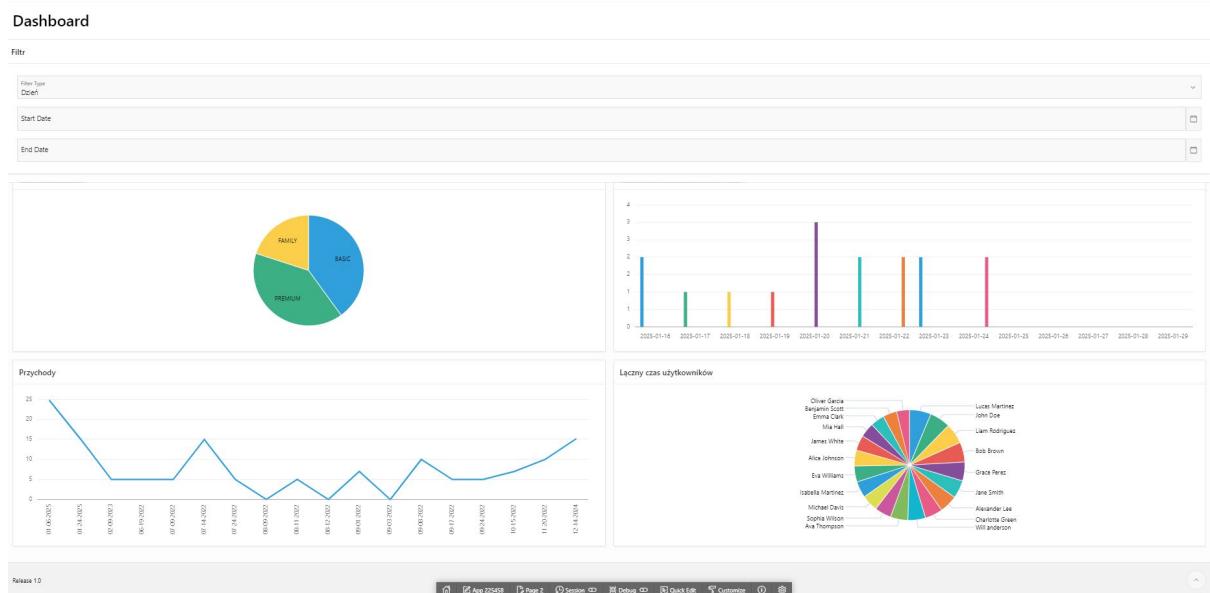
Pod czas importu były problemy z plikiem insertu danych, w komunikacie została wyświetlona informacja że import przebiegł pomyślnie jednak nie zostały dodane dane do tabel, po kilku krotnym odinstalowaniu i ponownym importem aplikacji problem został usunięty.(Możliwie jakiś bug APEX)

Sprawdzamy czy zostały dodane wszystkie elementy

The screenshot shows the Oracle Database SQL Developer interface. On the left, there is a tree view of tables, including 'ARTYSTA' which is selected. The main panel displays the 'ARTYSTA' table data:

ID_ARTYST	NAZWA	GATUNEK
1	Chivas	Rap
2	Szpała	Rap
3	ZYZZ	Gym Hardstyle
4	Baki	Gym Hardstyle
5	Juice WRLD	Rap
6	SM	Rock
7	Ed Sheeran	Pop
8	Golubenko	Pop
9	Simpba	Gym Hardstyle
10	avg'	Phunk

## Sprawdzamy czy działa aplikacja



Operacja exportu oraz importu jak wynika z załączonych zrzutów ekranów obrazujących ten proces przebiegła pomyślnie.

## Podsumowanie

Oracle APEX to wydajne narzędzie umożliwiające szybkie tworzenie aplikacji bazodanowych w ekosystemie Oracle. Dzięki intuicyjnemu projektowaniu interfejsu oraz łatwej integracji z bazą danych, APEX stanowi solidną platformę do budowy aplikacji webowych. Kluczową rolę odgrywają tutaj PL/SQL, procedury, funkcje, pakiety i kolekcje, które wspierają modułowe i efektywne zarządzanie aplikacją. W naszym projekcie wykorzystujemy Oracle APEX, ceniąc jego prostotę implementacji, elastyczność oraz skuteczność w tworzeniu nowoczesnych systemów bazodanowych. Równocześnie kładziemy nacisk na właściwe zarządzanie kodem źródłowym, co ułatwia długoterminowe utrzymanie i rozwój aplikacji.