

WYDZIAŁ
MATEMATYKI
I FIZYKI STOSOWANEJ
POLITECHNIKI RZESZOWSKIEJ

Aplikacje bazodanowe

Projekt

Baza danych internetowej platformy filmowej

Autorzy:

Krystian Pupiec,

Konrad Olszewski

gr. L4

Rzeszów, 10.01.2024r.

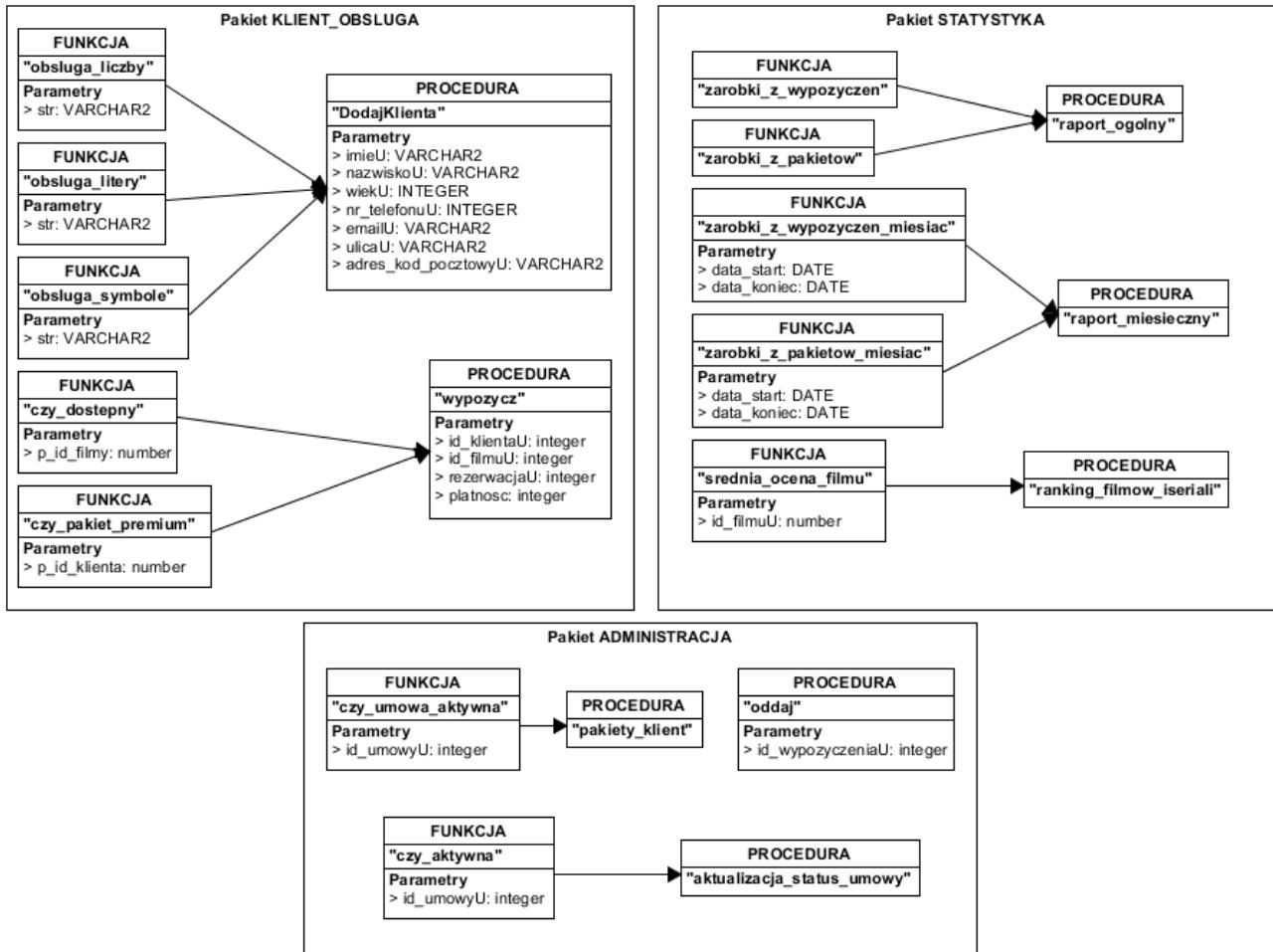
1.	Wstęp	3
2.	Use Cases – Pakiety, procedury, funkcje.....	3
	3
	4
3.	Funkcje.....	4
3.1.	Zarobki_z_wypozyzczem.....	4
3.2.	Zarobki_z_pakietow	5
3.3.	Zarobki_z_wypozyzczem_miesiac.....	5
3.4.	Zarobki_z_pakietow_miesiac.....	6
3.5.	Obsluga_liczby	7
3.6.	Obsluga_litery	7
3.7.	Obsluga_symbole	8
3.8.	Srednia_ocena_filmu	8
3.9.	Czy_aktywna	9
3.10.	Czy_umowa_aktywna.....	9
3.11.	Czy_dostepny.....	10
3.12.	Czy_pakiet_premium.....	10
4.	Procedury	11
4.1.	Raport_ogolny	11
4.2.	Raport_miesieczny	12
4.3.	DodajKlienta.....	14
4.4.	Ranking_filmow_i_seriali	17
4.5.	Aktualizacja_status_umowy	18
4.6.	Pakiety_klient	20
4.7.	Wypożycz	21
4.8.	Oddaj	27
5.	Pakiety.....	32
5.1.	Statystyka.....	32
5.2.	Klient_obsługa.....	37
5.3.	Administracja.....	46
6.	APEX	54
6.1.	Import bazy danych i pakietów z procedurami i funkcjami	54
6.2.	Tworzenie aplikacji bazodanowej – strona główna, menu.....	55
6.3.	Klient – interfejs. Wykorzystanie procedur i funkcji.	60

6.4.	Administrator – interfejs. Wykorzystanie procedur i funkcji.....	72
6.5.	Export aplikacji bazodanowej	91
7.	Wnioski i podsumowanie.....	95

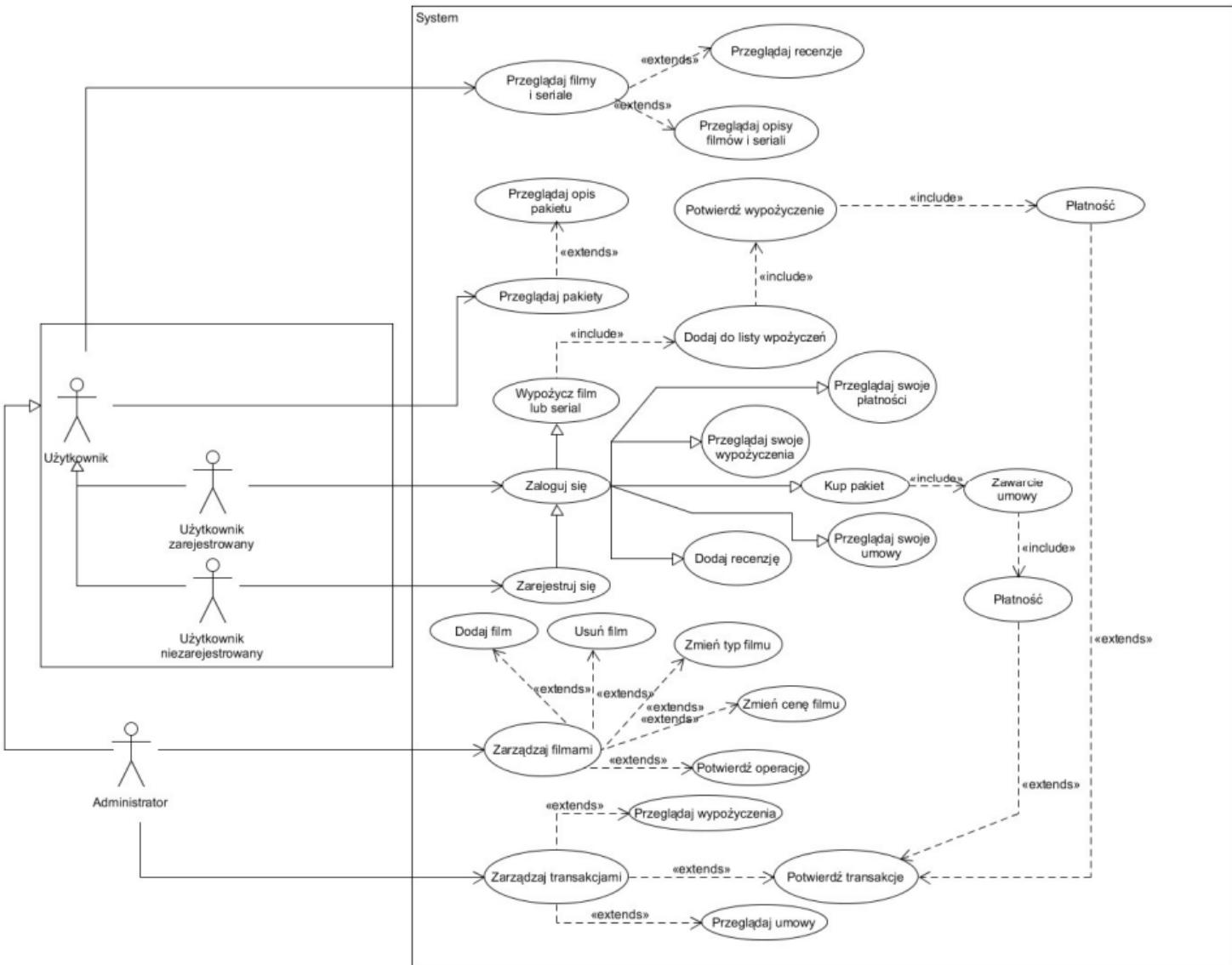
1. Wstęp

W celu wykonania projektu użyty został schemat bazy wykonanej podczas realizowania projektu z zajęć Bazy Danych – Baza danych internetowej platformy filmowej. W ramach wykonywanego pojęcia wykorzystywane będą odpowiednie skladnie, struktury funkcji, procedur, pakietów, kolekcji oraz innych ważnych poleceń w języku PL/SQL. Oprócz tego w ramach wizualizacji aplikacji bazodanowej użyta zostanie w tym celu platforma APEX.

2. Use Cases – Pakiety, procedury, funkcje



Rysunek 1 Use Cases - Pakiety, procedury, funkcje



Rysunek 2 Use Cases - baza danych

3. Funkcje

3.1. Zarobki_z_wypożyczen

Poniższa funkcja zlicza zarobki wygenerowane z wypożyczeń filmów oraz seriali. Wchodzi ona w skład procedury zarobki_z_pakietow.

CREATE OR REPLACE FUNCTION zarobki_z_wypożyczen RETURN INTEGER IS

zarobki INTEGER;

BEGIN

SELECT SUM(kwota_koncowa)

INTO zarobki

FROM wypożyczenia;

RETURN zarobki;

END zarobki_z_wypożyczen;

```
529 | SELECT zarobki_z_wypozczen() FROM DUAL;
```

Function created.

ZAROBKI_Z_WYPOZCZEN()
230

Rysunek 3 Wywołanie funkcji zarobki_z_wypozczen oraz wynik

3.2. Zarobki_z_pakietow

Poniższa funkcja zlicza zarobki wygenerowane z umów zawartych w wypożyczalni online. Wchodzi ona w skład procedury zarobki_z_pakietow.

```
CREATE OR REPLACE FUNCTION zarobki_z_pakietow RETURN INTEGER IS
zarobki INTEGER;
BEGIN
SELECT SUM(kwota_koncowa)
INTO zarobki
FROM umowy;
RETURN zarobki;
END zarobki_z_pakietow;
```

```
540 | SELECT zarobki_z_pakietow() FROM DUAL;
```

Function created.

ZAROBKI_Z_PAKIETOW()
380

Rysunek 4 Wywołanie funkcji zarobki_z_pakietow oraz wynik

3.3. Zarobki_z_wypozczen_miesiac

Funkcja mająca na celu obliczenie zarobków wygenerowanych przez wypożyczenia klientów w wirtualnej wypożyczalni filmów i seriali. Wchodzi ona w skład procedury (raport_miesieczny) pozwalającej na ukazanie zarobków miesięcznych dla wybranego roku. Poniższa funkcja może zwracać wartości również dla wybranego okresu.

```

CREATE OR REPLACE FUNCTION zarobki_z_wypozyzen_miesiac(data_start
DATE, data_koniec DATE) RETURN INTEGER IS
    zarobki INTEGER;
BEGIN
    SELECT SUM(kwota_koncowa)
    INTO zarobki
    FROM wypozyzenia
    WHERE data_wypozyzenia BETWEEN data_start AND data_koniec;
    RETURN COALESCE(zarobki, 0);
END zarobki_z_wypozyzen_miesiac;

```

571 `SELECT zarobki_z_wypozyzen_miesiac('01-DEC-22', '31-DEC-22') FROM DUAL;`

Function created.

ZAROBKI_Z_WPOZYCZEN_MIESIAC('01-DEC-22', '31-DEC-22')
50

[Download CSV](#)

Rysunek 5 Przykład wykonania funkcji dla okresu 01.12.2022-31.12.2022

3.4. Zarobki_z_pakietow_miesiac

Funkcja obliczająca zarobki wygenerowane z umów zawieranych przez klientów w wirtualnej wypożyczalni filmów i seriali. Wchodzi ona w skład procedury (raport_miesieczny) pozwalającej na ukazanie zarobków miesięcznych dla wybranego roku. Poniższa funkcja może zwracać wartości również dla wybranego okresu.

```

CREATE OR REPLACE FUNCTION zarobki_z_pakietow_miesiac(data_start DATE,
data_koniec DATE) RETURN INTEGER IS

```

zarobki INTEGER;

BEGIN

SELECT SUM(kwota_koncowa)

INTO zarobki

FROM umowy

WHERE data_zawarcia_umowy BETWEEN data_start AND data_koniec;

RETURN COALESCE(zarobki, 0);

```
END zarobki_z_pakietow_miesiac;
```

```
582  SELECT zarobki_z_pakietow_miesiac('01-JUN-22', '31-DEC-22') FROM DUAL;
```

Function created.

ZAROBKI_Z_PAKIETOW_MIESIAC('01-JUN-22', '31-DEC-22')
140

140

[Download CSV](#)

Rysunek 6 Przykład wykonania funkcji dla okresu 01.06.2022-31.12.2022

3.5. Obsluga_liczby

Jest to funkcja utworzona w celu obsługi wyjątków w procedurze DodajKlienta. Sprawdza czy ciąg znaków zawiera cyfry.

```
CREATE OR REPLACE FUNCTION obsluga_liczby(str VARCHAR2) RETURN
BOOLEAN IS
BEGIN
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;
```

3.6. Obsluga_litery

Jest to funkcja utworzona w celu obsługi wyjątków w procedurze DodajKlienta. Sprawdza czy ciąg znaków zawiera litery.

```
CREATE OR REPLACE FUNCTION obsluga_litery(str VARCHAR2) RETURN
BOOLEAN IS
BEGIN
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;

```

3.7. Obsluga_symbole

Jest to funkcja utworzona w celu obsługi wyjątków w procedurze DodajKlienta. Sprawdza czy w ciągu znaków jest zawarty znak specjalny '@'.

```

CREATE OR REPLACE FUNCTION obsluga_symbole(str VARCHAR2) RETURN
BOOLEAN IS

BEGIN

FOR i IN 1..LENGTH(str) LOOP

IF SUBSTR(str, i, 1) = '@' THEN

    RETURN TRUE;

END IF;

END LOOP;

RETURN FALSE;

```

3.8. Srednia_ocena_filmu

Funkcja wyliczająca średnią ocenę danego filmu lub serialu. Jako parametr przyjmuje ona numer id przypisany do produktu. Zawarta jest w procedurze ranking_filmow_i_seriali.

```

CREATE OR REPLACE FUNCTION srednia_ocena_filmu(id_filmuU NUMBER)
RETURN NUMBER IS

srednia_ocena NUMBER;

BEGIN

SELECT AVG(ocena)
INTO srednia_ocena
FROM recenzja_filmu_i_serialu
WHERE filmy_i_seriale_id_f_s = id_filmuU;
RETURN srednia_ocena;

```

```
END srednia_ocena_filmu;
```

Rysunek 7 Wykonanie funkcji dla filmu lub serialu o id 1

3.9. Czy_aktywna

Funkcja po podaniu odpowiedniego numeru id umowy sprawdza czy jest jeszcze aktywna, jeżeli różnica między aktualną datą a zawarciem umowy wynosi więcej lub równo 30 to zwraca wartość TRUE inaczej wartość FALSE. Znajduje się w procedurze aktualizacja_status_umowy.

```
CREATE OR REPLACE FUNCTION czy_aktywna(id_umowyU IN INTEGER)  
RETURN BOOLEAN IS
```

```

data_zawarcia_umowy.umowy.data_zawarcia_umowy%TYPE;
aktualna_data DATE;
roznica NUMBER;

BEGIN

SELECT data_zawarcia_umowy INTO data_zawarcia_umowy
FROM umowy
WHERE id_umowy = id_umowyU;
aktualna_data := SYSDATE;
roznica := aktualna_data - data_zawarcia_umowy;
IF roznica >= 30 THEN
    RETURN TRUE;
ELSE
    RETURN FALSE;
END IF;

END;

```

3.10. Czy umowa aktywna

Funkcja ta działa w podobny sposób jak funkcja `czy_aktywna`. Zawiera się w procedurze pakietu `klient`.

```

CREATE OR REPLACE FUNCTION czy_umowa_aktywna(id_umowyU) IN
INTEGER) RETURN BOOLEAN IS

    data_zakonczenia DATE;

BEGIN

    SELECT data_zakoczenia_umowy INTO data_zakonczenia
    FROM umowy
    WHERE id_umowy = id_umowyU;

    IF data_zakonczenia IS NULL OR data_zakonczenia > SYSDATE THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END;

```

3.11. Czy_dostepny

Funkcja sprawdzająca czy film, który klient chce wypożyczyć jest dostępny w bazie filmów (jest odpowiednia ilość na stanie).

```

CREATE OR REPLACE FUNCTION czy_dostepny(p_id_filmu NUMBER) RETURN
BOOLEAN IS

```

```

    v_ilosc NUMBER;

BEGIN

    SELECT ilosc INTO v_ilosc
    FROM filmy_i_seriale
    WHERE id_f_s = p_id_filmu;

    RETURN v_ilosc > 0;
END czy_dostepny;

```

3.12. Czy_pakiet_premium

Funkcja sprawdzająca, czy dany użytkownik posiada pakiet premium, zawiera się w procedurze wypożycz.

```

CREATE OR REPLACE FUNCTION czy_pakiet_premium(p_id_klienta NUMBER)
RETURN BOOLEAN IS

```

```

    pakietU VARCHAR2(40);

```

```

BEGIN
    SELECT pakiet INTO pakietU
    FROM klient
    WHERE id_klienta = p_id_klienta;

    RETURN pakietU = 'Premium';
END czy_pakiet_premium;

```

4. Procedury

4.1. Raport_ogolny

Procedura ta składa się z dwóch funkcji – zarobki_z_wypożyczen oraz zarobki_z_pakietow. Generuje raport przedstawiający aktualne zarobki wypożyczalni filmów i seriali wygenerowane przez wypożyczenia klientów jak i umowy przez nich zawartych w bazie danych a także sumuje je.

```

CREATE OR REPLACE PROCEDURE Raport_ogolny IS
    zarobki_wypożyczenia INTEGER;
    zarobki_pakiety INTEGER;

BEGIN
    zarobki_wypożyczenia := zarobki_z_wypożyczen();
    zarobki_pakiety := zarobki_z_pakietow();

    DBMS_OUTPUT.PUT_LINE('Raport o zarobkach:');
    DBMS_OUTPUT.PUT_LINE('Zarobki z wypożyczen: ' || zarobki_wypożyczenia);
    DBMS_OUTPUT.PUT_LINE('Zarobki z pakietow: ' || zarobki_pakiety);
    DBMS_OUTPUT.PUT_LINE('Calkowite zarobki: ' || (zarobki_wypożyczenia +
    zarobki_pakiety));

END Raport_ogolny;

```

```

556  BEGIN
557      Raport_ogolny();
558  END;
559 /

```

Procedure created.

Statement processed.
Raport o zarobkach:
Zarobki z wypożyczen: 230
Zarobki z pakietow: 380
Calkowite zarobki: 610

Rysunek 8 Wywołanie procedury Raport_ogolny

4.2. Raport_miesieczny

Poniżej przedstawiona procedura tworzy raport miesięczny z zarobków wygenerowanych w danym okresu (tutaj roku) dla wypożyczalni filmów i seriali. Odpowiednie wartości są wyświetlane następnie w czytelny dla użytkownika sposób. Procedura ta używa dwóch funkcji: zarobki_z_pakietow_miesiac i zarobki_z_wypożyczen_miesiac,

CREATE OR REPLACE PROCEDURE rapport_miesieczny IS

poczatkowa_data DATE;

koncowa_data DATE;

miesiac_poczatek DATE;

miesiac_koniec DATE;

zar_wyp INTEGER;

zar_pak INTEGER;

BEGIN

poczatkowa_data := TO_DATE('2022-01-01', 'YYYY-MM-DD');

koncowa_data := TO_DATE('2022-12-31', 'YYYY-MM-DD');

DBMS_OUTPUT.PUT_LINE('RAPORT O ZAROBKACH MIESIECZNYCH:');

DBMS_OUTPUT.PUT_LINE('xxxxxxxxxxxxxxxxxxxx');

FOR i IN 1..12 LOOP

miesiac_poczatek := ADD_MONTHS(poczatkowa_data, i - 1);

miesiac_koniec := LAST_DAY(miesiac_poczatek);

```

zar_wyp := zarobki_z_wypozyzen_miesiac(miesiac_poczatek, miesiac_koniec);
zar_pak := zarobki_z_pakietow_miesiac(miesiac_poczatek, miesiac_koniec);

DBMS_OUTPUT.PUT_LINE('Miesiac: ' // TO_CHAR(miesiac_poczatek,
'MM/YYYY'));
DBMS_OUTPUT.PUT_LINE('Zarobki z wypozyzen: ' // zar_wyp);
DBMS_OUTPUT.PUT_LINE('Zarobki z pakietow: ' // zar_pak);
DBMS_OUTPUT.PUT_LINE('Calkowite zarobki: ' // (zar_wyp + zar_pak));
DBMS_OUTPUT.PUT_LINE('xxxxxxxxxxxxxxxxxxxx');

END LOOP;

END rapport_miesieczny;

```

612	BEGIN
613	<i>raport_miesieczny();</i>
614	END;
615	/

Procedure created.

Statement processed.
 RAPORT O ZAROBKACH MIESIECZNYCH:
 XXXXXXXXXXXXXXXXXXXX
 Miesiac: 01/2022
 Zarobki z wypozyzen: 60
 Zarobki z pakietow: 120
 Calkowite zarobki: 180
 XXXXXXXXXXXXXXXXXXXX
 Miesiac: 02/2022
 Zarobki z wypozyzen: 20
 Zarobki z pakietow: 20
 Calkowite zarobki: 40
 XXXXXXXXXXXXXXXXXXXX
 Miesiac: 03/2022
 Zarobki z wypozyzen: 0
 Zarobki z pakietow: 0
 Calkowite zarobki: 0
 XXXXXXXXXXXXXXXXXXXX
 Miesiac: 04/2022
 Zarobki z wypozyzen: 0

Rysunek 9 Wywołanie procedury rapport_miesieczny

4.3. DodajKlienta

Poniższa procedura dodaje nowego klienta do bazy danych wypożyczalni. W kodzie została zawarta również obsługa wyjątków, która nie pozwala na niepoprawne wpisywanie danych przy rejestracji użytkownika. W celu obsługi błędów zostały dodane trzy funkcje. Oprócz tego procedura sama dodaje następne kolejne id użytkownikowi.

```
CREATE OR REPLACE PROCEDURE DodajKlienta(  
    imieU VARCHAR2,  
    nazwiskoU VARCHAR2,  
    wiekU INTEGER,  
    nr_telefonuU INTEGER,  
    emailU VARCHAR2,  
    ulicaU VARCHAR2,  
    adres_kod_pocztowyU VARCHAR2 ) IS  
    brak_imienia EXCEPTION;  
    brak_nazwiska EXCEPTION;  
    brak_wieku EXCEPTION;  
    brak_numeru EXCEPTION;  
    brak_emailu EXCEPTION;  
    brak_ulicy EXCEPTION;  
  
    adres_id VARCHAR2(50);  
    max_klient_id NUMBER;  
  
BEGIN  
    IF (imieU IS NULL) OR obsluga_liczby(imieU) THEN  
        RAISE brak_imienia;  
    ELSIF nazwiskoU IS NULL OR obsluga_liczby(nazwiskoU) THEN  
        RAISE brak_nazwiska;  
    ELSIF wiekU IS NULL OR obsluga_litery(wiekU) OR obsluga_symbole(wiekU)  
    THEN  
        RAISE brak_wieku;  
    ELSIF nr_telefonuU IS NULL OR obsluga_litery(nr_telefonuU) OR  
    obsluga_symbole(nr_telefonuU) THEN  
        RAISE brak_emailu;
```

```

RAISE brak_numeru;

ELSIF emailU IS NULL OR NOT obsluga_symbole(emailU) THEN
RAISE brak_emailu;

ELSIF ulicaU IS NULL THEN
RAISE brak_ulicy;

ELSE
SELECT kod_pocztowy INTO adres_id FROM adres WHERE kod_pocztowy =
adres_kod_pocztowyU;

SELECT MAX(id_klienta) INTO max_klient_id FROM klient;

INSERT INTO klient (id_klienta, imie, nazwisko, wiek, nr_telefonu, email, ulica,
adres_kod_pocztowy)

VALUES (max_klient_id + 1, imieU, nazwiskoU, wiekU, nr_telefonuU, emailU,
ulicaU, adres_kod_pocztowyU);

DBMS_OUTPUT.PUT_LINE('Klient dodany pomyślnie.');

END IF;

```

EXCEPTION

```

WHEN brak_imienia THEN
DBMS_OUTPUT.PUT_LINE('Nie wprowadzono imienia lub wprowadzono
niedozwolone znaki. Uzupełnij dane.');

WHEN brak_nazwiska THEN
DBMS_OUTPUT.PUT_LINE('Nie wprowadzono nazwiska lub wprowadzono
niedozwolone znaki. Uzupełnij dane.');

WHEN brak_wieku THEN
DBMS_OUTPUT.PUT_LINE('Nie wprowadzono wieku lub wprowadzono
niedozwolone znaki. Uzupełnij dane.');

WHEN brak_numeru THEN
DBMS_OUTPUT.PUT_LINE('Nie wprowadzono numeru telefonu lub
wprowadzono niedozwolone znaki. Uzupełnij dane.');

WHEN brak_emailu THEN
DBMS_OUTPUT.PUT_LINE('Nie wprowadzono adresu e-mail lub jest
nieprawidłowy. Uzupełnij dane.');

WHEN brak_ulicy THEN

```

```

    DBMS_OUTPUT.PUT_LINE('Nie wprowadzono ulicy w adresie lub
    wprowadzono niedozwolone znaki. Uzupełnij dane.');
WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Podany kod pocztowy nie istnieje w tabeli adres.');
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas dodawania klienta');
END DodajKlienta;
/

```

```

711 BEGIN
712 DodajKlienta('Tomasz', 'Kowlaski', 32, '123411555', 'kowal_sd.pl', 'Mila', '33-230');
713 END;
714 /

```

Statement processed.
Nie wprowadzono adresu e-mail lub jest nieprawidłowy. Uzupełnij dane.

Rysunek 10 Działanie wyjątków

Poprawne wykonanie procedury:

ID_Klienta	IMIE	NAZWISKO	WIEK	NR_TELEFONU	EMAIL	ULICA	ADRES_KOD_POCZTOWY	PAKET
1	Jan	Kowlaski	19	517928516	jkowalski@onet	Urocza	33-230	-
2	Piotr	Kawka	19	516753851	pkawka@onet	Solna	33-232	-
3	Jadwiga	Nowak	19	587626881	jnowak@onet	Wielkopolska	33-231	-
4	Zuzanna	Kijek	33	501693279	zkijek@onet	Partyzantow	33-235	-
5	Klaudia	Hula	25	719465578	khula@onet	Podkarpacka	33-233	-
6	Waldemar	Sroka	42	798587436	wsroka@onet	Ptasia	33-231	-
7	Pawel	Nicpon	28	587469875	pnicpona@onet	Mila	33-232	-
8	Aleksandra	Mila	17	953678731	amila@onet	Wesola	33-230	-
9	Marcin	Ziolko	31	467987656	mzilko@onet	Weteranow	33-234	-
10	Anna	Wesolowska	39	678876876	awesolowska@onet	Sadowa	33-230	-

Download CSV

Rysunek 11 Tabela klient przed dodaniem użytkownika

```

711 BEGIN
712 DodajKlienta('Tomasz', 'Kowlaski', 32, '123411555', 'kowal@sd.pl', 'Mila', '33-230');
713 END;
714 /

```

Statement processed.
Klient dodany pomyślnie.

Rysunek 12 Dodanie klienta

```
715 select * from klient
```

ID_Klienta	IMIE	NAZWISKO	WIEK	NR_TELEFONU	EMAIL	ULICA	ADRES_KOD_POCZTOWY	PAKIET
11	Tomasz	Kowlaski	32	123411555	kowal@sd.pl	Mila	33-230	-
1	Jan	Kowlaski	19	517928516	jkowalski@onet	Uroczna	33-230	-
2	Piotr	Kawka	19	516753851	pkawka@onet	Solna	33-232	-

Rysunek 13 Efekt wykonania procedury

4.4. Ranking_filmow_i_seriali

Procedura tworząca ranking zrecenzowanych filmów przez użytkowników internetowej wypożyczalni według uśrednionej oceny. Ta procedura używa funkcji srednia_ocena_filmu.

```

CREATE OR REPLACE PROCEDURE ranking_filmow_i_seriali IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Ranking filmów:');
    DBMS_OUTPUT.PUT_LINE('xxxxxxxxxxxxxxxxxxxx');
    FOR film IN (SELECT f.tytul,
    COALESCE(TO_CHAR(srednia_ocena_filmu(f.id_f_s)), 'Film nie został jeszcze
    oceniony') AS srednia_ocena
    FROM filmy_i_seriali f
    ORDER BY TO_NUMBER(srednia_ocena_filmu(f.id_f_s)) DESC) LOOP
        DBMS_OUTPUT.PUT_LINE(film.tytul || ' - Średnia ocena: ' ||
        film.srednia_ocena);
    END LOOP;
END ranking_filmow_i_seriali;
/

```

```
547 BEGIN  
548     ranking_filmow_i_seriali;  
549 END;  
550 /
```

Procedure created.

Statement processed.

Ranking filmów:

xxxxxxxxxxxxxxxxxxxx

Uśmiechnij się - Średnia ocena: Film nie został jeszcze oceniony

Avatar Istota wody - Średnia ocena: 4.5

Nietykalni - Średnia ocena: 1

Rysunek 14 Wywołanie procedury ranking_filmow_i_seriali

4.5. Aktualizacja status umowy

Procedura aktualizująca status umowy w tabeli umowy. Korzysta z funkcji czy_aktywna.

```
CREATE OR REPLACE PROCEDURE aktualizacja_status_umowy IS
```

CURSOR umowy_cursor IS

SELECT *

FROM umowy;

umowa umowy%ROWTYPE;

BEGIN

FOR umowa rec IN umowy cursor LOOP

umowa := umowa rec;

JE CZY AKTYWNA(UMOWA.ID_UMOWY) THEN

UPDATE umowy

SET status = 'NIEAKTYWNA'

WHERE id_umowy = umowa.id_umowy;

DBMS_OUTPUT.PUT_LINE('Status umowy o ID ' || umowa.id_umowy || '
został zaktualizowany na NIEAKTYWNA.');

ELSE

```

        DBMS_OUTPUT.PUT_LINE('Nie minęło jeszcze 30 dni od zawarcia umowy
o ID ' || umowa.id_umowy);

    END IF;

    END LOOP;

END;
/

```

ID_UMOWY	KLIENT_ID_Klienta	ADMINISTRATOR_ID_A	DATA_ZAWARCIA_UMOWY	DATA_ZAKOCZENIA_UMOWY	KWOTA_KONCOWA	PAKIETY_ID_PAKIETU	RODZAJ_PLATNOSCI_ID_P	STATUS
1	1	1	28-DEC-22	28-JAN-23	60	1	1	-
2	5	2	12-JAN-22	12-FEB-22	60	1	1	-
3	8	1	19-FEB-22	19-MAR-22	20	3	2	-
4	9	2	28-JAN-22	28-FEB-22	60	1	3	-
5	4	2	14-DEC-22	14-JAN-23	60	1	1	-
6	10	2	10-JUL-22	10-AUG-22	20	3	2	-
7	7	1	28-MAR-20	28-APR-20	40	2	3	-
8	6	1	01-APR-20	01-MAY-20	60	1	2	-

Rysunek 15 Tabela umowy przed wywołaniem procedury

```

596 v begin
597 aktualizacja_status_umowy;
598 end;
599 /

```

Statement processed.
Status umowy o ID 1 został zaktualizowany na NIEAKTYWNA.
Status umowy o ID 2 został zaktualizowany na NIEAKTYWNA.
Status umowy o ID 3 został zaktualizowany na NIEAKTYWNA.
Status umowy o ID 4 został zaktualizowany na NIEAKTYWNA.
Status umowy o ID 5 został zaktualizowany na NIEAKTYWNA.
Status umowy o ID 6 został zaktualizowany na NIEAKTYWNA.
Status umowy o ID 7 został zaktualizowany na NIEAKTYWNA.
Status umowy o ID 8 został zaktualizowany na NIEAKTYWNA.

Rysunek 16 Wywołanie procedury

ID_UMOWY	KLIENT_ID_Klienta	ADMINISTRATOR_ID_A	DATA_ZAWARCIA_UMOWY	DATA_ZAKOCZENIA_UMOWY	KWOTA_KONCOWA	PAKIETY_ID_PAKIETU	RODZAJ_PLATNOSCI_ID_P	STATUS
1	1	1	28-DEC-22	28-JAN-23	60	1	1	NIEAKTYWNA
2	5	2	12-JAN-22	12-FEB-22	60	1	1	NIEAKTYWNA
3	8	1	19-FEB-22	19-MAR-22	20	3	2	NIEAKTYWNA
4	9	2	28-JAN-22	28-FEB-22	60	1	3	NIEAKTYWNA
5	4	2	14-DEC-22	14-JAN-23	60	1	1	NIEAKTYWNA
6	10	2	10-JUL-22	10-AUG-22	20	3	2	NIEAKTYWNA
7	7	1	28-MAR-20	28-APR-20	40	2	3	NIEAKTYWNA
8	6	1	01-APR-20	01-MAY-20	60	1	2	NIEAKTYWNA

Rysunek 17 Tabela umowy po wywołaniu procedury

4.6. Pakiety_klient

Procedura ta sprawdza czy dany klient ma jakąś aktywną umowę, w zależności od typu umowy przypisuje mu odpowiedni pakiet w tabeli klient.

```
CREATE OR REPLACE PROCEDURE pakiet_klient AS
```

```
    id_umowy umowy.id_umowy%TYPE;
    id_klientaU umowy.klient_id_klienta%TYPE;
    rodzaj_umowy umowy.rodzaj_platnosci_id_p%TYPE;

    BEGIN
        FOR umowa IN (SELECT * FROM umowy) LOOP
            id_umowy := umowa.id_umowy;
            id_klientaU := umowa.klient_id_klienta;
            rodzaj_umowy := umowa.rodzaj_platnosci_id_p;
            IF czy_umowa_aktywna(id_umowy) THEN
                UPDATE klient
                SET pakiet = CASE
                    WHEN rodzaj_umowy = 1 THEN 'Podstawowy'
                    WHEN rodzaj_umowy = 2 THEN 'Standardowy'
                    WHEN rodzaj_umowy = 3 THEN 'Premium'
                    ELSE 'Brak'
                END;
                WHERE id_klienta = id_klientaU;
            ELSE
                DBMS_OUTPUT.PUT_LINE('BRAK AKTYWNEJ UMOWY DLA
Klienta o ID: ' || id_klientaU);
            END IF;
        END LOOP;
    END pakiet_klient;
```

```
/
```

```

642 | BEGIN
643 |     pakiet_klient;
644 | END;
645 |

```

```

Statement processed.
BRAK AKTYWNEJ UMOWY DLA Klienta o ID: 1
BRAK AKTYWNEJ UMOWY DLA Klienta o ID: 5
BRAK AKTYWNEJ UMOWY DLA Klienta o ID: 8
BRAK AKTYWNEJ UMOWY DLA Klienta o ID: 9
BRAK AKTYWNEJ UMOWY DLA Klienta o ID: 4
BRAK AKTYWNEJ UMOWY DLA Klienta o ID: 10
BRAK AKTYWNEJ UMOWY DLA Klienta o ID: 7
BRAK AKTYWNEJ UMOWY DLA Klienta o ID: 6

```

Rysunek 18 Wywołanie procedury pakiet_klient

4.7. Wypożycz

Procedura tworząca nowe wypożyczenie dla klienta, korzysta z funkcji czy_dostepny i czy_pakiet_premium. Procedura sprawdza ilość danego filmu na stanie wypożyczalni. Jeżeli aktualnie film jest niedostępny w zależności, czy użytkownik zadeklarował chęć rezerwacji tworzy ją dla niego oraz dodaje do tabeli rezerwacje. Oprócz głównego zastosowania kod został również zaopatrzony w obsługę błędów - wyjątki.

```

CREATE OR REPLACE PROCEDURE wypożycz(
    id_klientaU IN INTEGER,
    id_filmuU IN INTEGER,
    rezerwacjaU IN INTEGER DEFAULT 0,
    platnosc in integer
) IS
    ilosc_dniU NUMBER := 7;
    cenaU NUMBER;
    data_wypU DATE;
    data_oddU DATE;
    data_zakU DATE;
    data_rezU DATE;
    pakietU VARCHAR2(40);
    stan_filmuU INTEGER;
    data_koncowaU DATE;
    wypożyczenia_idU INTEGER;
    zla_platnosc exception;

```

```

rez_blad exception;
zly_klient exception;
zly_film exception;

BEGIN

IF platnosc NOT IN (1, 2, 3) THEN
    RAISE zla_platnosc;

ELSIF rezerwacjaU NOT IN (0,1) THEN
    RAISE rez_blad;

ELSIF id_klientaU is NULL or id_klientaU = 0 then
    RAISE zly_klient;

ELSIF id_filmuU is NULL or id_filmuU = 0 then
    RAISE zly_film;

ELSE

    IF czy_dostepny(id_filmuU) THEN
        -- Pobranie aktualnego stanu filmu
        SELECT ilosc INTO stan_filmuU
        FROM filmy_i_seriale
        WHERE id_f_s = id_filmuU;

        IF czy_pakiet_premium(id_klientaU) THEN
            SELECT CASE
                WHEN ilosc_dniU > (SELECT
MONTHS_BETWEEN(u.data_zakoczenia_umowy, SYSDATE) + 1
                FROM umowy u
                WHERE u.klient_id_klienta = id_klientaU
                AND SYSDATE BETWEEN u.data_zawarcia_umowy AND
u.data_zakoczenia_umowy)
                THEN (SELECT MONTHS_BETWEEN(u.data_zakoczenia_umowy,
SYSDATE) + 1
                FROM umowy u
                WHERE u.klient_id_klienta = id_klientaU

```

```

        AND SYSDATE BETWEEN u.data_zawarcia_umowy AND
u.data_zakoczenia_umowy)

        ELSE ilosc_dniU

        END

        INTO ilosc_dniU

        FROM dual;

cenaU := 0;

ELSE

        SELECT cena INTO cenaU

        FROM filmy_i_seriale

        WHERE id_f_s = id_filmuU;

END IF;

data_wypU := SYSDATE;

data_oddU := SYSDATE + ilosc_dniU;

UPDATE filmy_i_seriale

SET ilosc = stan_filmuU - 1

WHERE id_f_s = id_filmuU;

SELECT MAX(id_wypozyzzenia) + 1 INTO wypozyzzenia_idU

FROM wypozyzzenia;

INSERT INTO wypozyzzenia (id_wypozyzzenia, klient_id_klienta,
administrator_id_ad, filmy_i_seriale_id_f_s, data_wypozyzzenia, data_oddania,
kwota_koncowa, rodzaj_platnosci_id_p, status)

VALUES (wypozyzzenia_idU, id_klientaU, 1, id_filmuU, data_wypU, data_oddU,
cenaU, platnosc, 'W TRAKCIE');

DBMS_OUTPUT.PUT_LINE('Film wypożyczony! Data oddania: ' ||
TO_CHAR(data_oddU, 'DD.MM.YYYY'));

ELSE

IF rezerwacjaU = 0 THEN

DBMS_OUTPUT.PUT_LINE('Wpadnij kiedy indziej!');


```

```

SELECT MAX(data_koncowa) INTO data_zakU
FROM rezerwacje
WHERE filmy_i_seriale_id_f_s = id_filmuU
AND data_koncowa IS NOT NULL;

SELECT MAX(data_oddania) INTO data_oddU
FROM wypozyzzenia
WHERE filmy_i_seriale_id_f_s = id_filmuU;
IF data_zakU IS NOT NULL AND data_zakU > data_oddU THEN
    DBMS_OUTPUT.PUT_LINE('Ostatnia rezerwacja zakończona dnia: ' ||
TO_CHAR(data_zakU, 'DD.MM.YYYY'));
ELSE
    DBMS_OUTPUT.PUT_LINE('Ostatnie wypożyczenie zakończone dnia: ' ||
TO_CHAR(data_oddU, 'DD.MM.YYYY'));
END IF;
ELSIF rezerwacjaU = 1 THEN

SELECT MAX(data_koncowa) INTO data_zakU
FROM rezerwacje
WHERE filmy_i_seriale_id_f_s = id_filmuU
AND data_koncowa IS NOT NULL;

SELECT MAX(data_oddania) INTO data_oddU
FROM wypozyzzenia
WHERE filmy_i_seriale_id_f_s = id_filmuU;

IF data_zakU IS NOT NULL AND data_zakU > data_oddU THEN
    data_rezU := data_zakU + 1;
ELSIF data_oddU IS NOT NULL THEN
    data_rezU := data_oddU + 1;
ELSE

```

```

    data_rezU := SYSDATE + 1;
END IF;

data_koncowaU := data_rezU + 7;

SELECT MAX(id_rezerwacji) + 1 INTO wypozyzenia_idU
FROM rezerwacje;

INSERT INTO rezerwacje (id_rezerwacji, klient_id_klienta,
filmy_i_seriale_id_f_s, data_początkowa, data_koncowa)
VALUES (rezerw.NEXTVAL, id_klientaU, id_filmuU, data_rezU,
data_koncowaU);

DBMS_OUTPUT.PUT_LINE('Film zarezerwowany do ' ||
TO_CHAR(data_koncowaU, 'DD.MM.YYYY'));

END IF;

END IF;

END IF;

EXCEPTION

WHEN zla_platnosc THEN

DBMS_OUTPUT.PUT_LINE('zla platnosc');

RETURN;

WHEN rez_blad THEN

DBMS_OUTPUT.PUT_LINE('Zle okreslenie rezerwacji! Jezeli chcesz
zarezerwowac przy wypozyzeniu wpisz 1, jezeli nie wpisz 0!');

RETURN;

when zly_klient then

DBMS_OUTPUT.PUT_LINE('Niepoprawne id uzytkownika');

RETURN;

when zly_film then

DBMS_OUTPUT.PUT_LINE('Niepoprawne id filmu');

RETURN;

when OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Cos poszlo nie tak! Sprawdz wpisane dane!');


```

```

    RETURN;
END wypożycz;

```

```

856 BEGIN
857   wypożycz(1, 3, 1,2);
858 END;
859 /

```

Statement processed.
Film wypożyczony! Data oddania: 25.12.2023

Rysunek 19 Wywołanie procedury w przypadku wypożyczenia

868 select * from wypożyczenia									
20	1	1	2	18-DEC-23	25-DEC-23	20	2		W TRAKCIE
22	6	1	2	01-APR-20	17-DEC-23	20	2		ODDANE
31	1	1	3	18-DEC-23	25-DEC-23	30	2		W TRAKCIE

Rysunek 20 Dodanie wypożyczenia

```

856 BEGIN
857   wypożycz(1, 2, 1,2);
858 END;

```

Statement processed.
Film zarezerwowany do 18.01.2024

Rysunek 21 Wywołanie procedury w przypadku rezerwacji

866 select * from rezerwacje				
ID_REZERWACJI	KLIENT_ID_Klienta	FILMY_I_SERIALE_ID_F_S	DATA_POCZĄTKOWA	DATA_KONCOWA
2	1	1	26-DEC-23	02-JAN-24
3	1	1	03-JAN-24	10-JAN-24
4	1	1	11-JAN-24	18-JAN-24
10	1	2	11-JAN-24	18-JAN-24
9	1	2	03-JAN-24	10-JAN-24

Rysunek 22 Dodana rezerwacja

4.8. Oddaj

Procedura sprawdzająca, czy dane wypożyczenie skończyło się – minął termin. W przypadku, gdy minie okres wypożyczenia filmu lub serialu sprawdzany jest pod kątem rezerwacji. W momencie, gdy istnieją rezerwacje na dany tytuł najwcześniej utworzona rezerwacja zostaje dodana jako nowe wypożyczenie do tabeli wypożyczenia, natomiast z tabeli rezerwacje zostaje dodana do kolekcji tablicy i usuwana z głównej tabeli rezerwacje. Utworzona akcja utworzona została w celu tworzenia historii na temat rezerwacji. W momencie, gdy nie istnieje żadna rezerwacja na dany film lub serial stan filmu zostaje powiększony o ilość oddanych sztuk. Procedura została zaopatrzona w obsługę błędów – wyjątki.

```
CREATE OR REPLACE PROCEDURE oddaj (
    id_wypożyczeniaU IN INTEGER
) IS
    statusU      VARCHAR2(40);
    data_wypożyczeniaU DATE;
    data_dzisiejszaU DATE := SYSDATE;
    cenaU        INTEGER;
    stan_filmuU   INTEGER;
    rezerwacja_idU INTEGER;
    stan_premiumU VARCHAR2(10);
    ilosc_dniU     INTEGER;
    data_oddaniaU DATE;
    id_klientaU    INTEGER;
    nowe_id_wypożyczeniaU INTEGER;
    rekord_rezerwacje%ROWTYPE;
    i NUMBER:=0;
    max_id_wypożyczenia INTEGER;

    TYPE rezerwacje_historia IS TABLE OF rezerwacje%ROWTYPE;
    rezerwacja rezerwacje_historia:=rezerwacje_historia();
    zle_id_wypożyczenia exception;

BEGIN
    SELECT MAX(id_wypożyczenia) INTO max_id_wypożyczenia FROM wypożyczenia;
```

```

IF id_wypożyczeniaU > max_id_wypożyczenia THEN
RAISE zle_id_wypożyczenia;
ELSE
    SELECT status, data_wypożyczenia, klient_id_klienta
    INTO statusU, data_wypożyczeniaU, id_klientaU
    FROM wypożyczenia
    WHERE id_wypożyczenia = id_wypożyczeniaU;

IF statusU = 'W TRAKCIE' THEN
    IF data_dzisiejszaU - data_wypożyczeniaU > 7 THEN
        UPDATE wypożyczenia
        SET status = 'ODDANE'
        WHERE id_wypożyczenia = id_wypożyczeniaU;

SELECT MIN(id_rezerwacji)
INTO rezerwacja_idU
FROM rezerwacje
WHERE filmy_i_seriale_id_f_s = (
    SELECT filmy_i_seriale_id_f_s
    FROM wypożyczenia
    WHERE id_wypożyczenia = id_wypożyczeniaU
);

IF rezerwacja_idU IS NOT NULL THEN
    SELECT klient_id_klienta, filmy_i_seriale_id_f_s
    INTO id_klientaU, stan_filmuU
    FROM rezerwacje
    WHERE id_rezerwacji = rezerwacja_idU;
    FOR rekord IN (SELECT * FROM rezerwacje WHERE id_rezerwacji =
rezerwacja_idU) LOOP
i:=i+1;

```

```

rezerwacja.EXTEND;
rezerwacja(i):=rekord;
END LOOP;
DELETE FROM rezerwacje WHERE id_rezerwacji = rezerwacja_idU;
FOR i IN 1..rezerwacja.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE('ID Rezerwacji: ' ||
rezerwacja(i).id_rezerwacji);
    DBMS_OUTPUT.PUT_LINE('ID Klienta: ' ||
rezerwacja(i).klient_id_klienta);
    DBMS_OUTPUT.PUT_LINE('ID filmu lub serialu: ' ||
rezerwacja(i).FILMY_I_SERIALE_ID_F_S );
    DBMS_OUTPUT.PUT_LINE('Data początkowa: ' ||
rezerwacja(i).DATA_POCZATKOWA );
    DBMS_OUTPUT.PUT_LINE('Data koncowa: ' ||
rezerwacja(i).DATA_KONCOWA );
END LOOP;

SELECT cena
INTO cenaU
FROM filmy_i_serial
WHERE id_f_s = stan_filmuU;

SELECT pakiet
INTO stan_premiumU
FROM klient
WHERE id_klienta = id_klientaU;

IF stan_premiumU = 'PREMIUM' THEN
    cenaU := 0;
    SELECT u.DATA_ZAKOCZENIA_UMOWY - SYSDATE
    INTO ilosc_dniU

```

```
FROM umowy u
JOIN klient k ON u.klient_id_klienta = k.id_klienta
WHERE k.id_klienta = id_klientaU;
```

```
IF ilosc_dniU < 7 THEN
    data_oddaniaU := SYSDATE + ilosc_dniU;
ELSE
    data_oddaniaU := SYSDATE + 7;
END IF;
ELSE
    data_oddaniaU := SYSDATE + 7;
END IF;
```

```
SELECT MAX(id_wypozyczenia) + 1
INTO nowe_id_wypozyczeniaU
FROM wypozyczenia;
```

```
INSERT INTO wypozyczenia (id_wypozyczenia, klient_id_klienta,
filmy_i_seriale_id_f_s, administrator_id_ad, data_wypozyczenia, data_oddania,
kwota_koncowa, rodzaj_platnosci_id_p, status)
VALUES (nowe_id_wypozyczeniaU, id_klientaU, stan_filmuU, 1, SYSDATE,
data_oddaniaU, cenaU, 2, 'W TRAKCIE');
ELSE
    UPDATE filmy_i_seriale
    SET ilosc = ilosc + 1
    WHERE id_f_s = (
        SELECT filmy_i_seriale_id_f_s
        FROM wypozyczenia
        WHERE id_wypozyczenia = id_wypozyczeniaU
    );
END IF;
END IF;
```

```

ELSE
    DBMS_OUTPUT.PUT_LINE('Film nie jest wypożyczony.');
END IF;
END IF;
EXCEPTION
WHEN zle_id_wypozyczenia THEN
    DBMS_OUTPUT.PUT_LINE('Zle id wypozyczenia');
RETURN;
END oddaj;
/

```

select * from wypozyczenia								
40	6	1	2	01-APR-20	17-DEC-23	20	2	W TRAKCIE

Rysunek 23 Wypożyczenie, które zostanie oddane

W tym przypadku istnieją również rezerwacje dla filmu lub serialu o id równym 2.

```

861 v BEGIN
862      oddaj(40);
863 END;
864 v /
oocr

Statement processed.
ID Rezerwacji: 9
ID Klienta: 1
ID filmu lub serialu: 2
Data początkowa: 03-JAN-24
Data koncowa: 10-JAN-24

```

Rysunek 24 Wywołanie procedury oddaj oraz wyświetlenie się zawartości kolekcji (wyświetlenie informacji o nowym wypożyczeniu z rezerwacji)

```
866 | select * from rezerwacje
```

ID_REZERWACJI	KLIENT_ID_Klienta	FILMY_I_SERIALE_ID_F_S	DATA_POCZATKOWA	DATA_KONCOWA
2	1	1	26-DEC-23	02-JAN-24
3	1	1	03-JAN-24	10-JAN-24
4	1	1	11-JAN-24	18-JAN-24
10	1	2	11-JAN-24	18-JAN-24

Rysunek 25 Potwierdzenie usunięcia rezerwacji

868 select * from wypozyzenia								
18	1	1	2	18-DEC-23	25-DEC-23	20	2	W TRAKCIE
11	1	1	1	18-DEC-23	25-DEC-23	20	2	W TRAKCIE
41	1	1	2	18-DEC-23	25-DEC-23	20	2	W TRAKCIE

Rysunek 26 Dodanie wypożyczenia z rezerwacji

5. Pakiety

5.1. Statystyka

Zawarte funkcje:

- Zarobki_z_wypożyczeń
- Zarobki_z_pakietów
- Zarobki_z_wypożyczeń_miesiąc
- Zarobki_z_pakietów_miesiąc
- Średnia_ocena_filmu

Zawarte procedury:

- Raport_ogólny
- Raport_miesieczny
- Raking_filmów_i_seriali

Budowa pakietu:

```
CREATE OR REPLACE PACKAGE STATYSTYKA IS
FUNCTION zarobki_z_wypożyczeń RETURN INTEGER;
FUNCTION zarobki_z_pakietów RETURN INTEGER;
FUNCTION zarobki_z_wypożyczeń_miesiąc(data_start DATE, data_koniec DATE)
RETURN INTEGER;
FUNCTION zarobki_z_pakietów_miesiąc(data_start DATE, data_koniec DATE)
RETURN INTEGER;
```

```

FUNCTION srednia_ocena_filmu(id_filmuU NUMBER) RETURN NUMBER;
PROCEDURE raport_ogolny;
PROCEDURE raport_miesieczny;
PROCEDURE ranking_filmow_i_seriali;
END;
/
CREATE OR REPLACE PACKAGE BODY STATYSTYKA AS
/* zarobki_z_wypozyzen */
FUNCTION zarobki_z_wypozyzen RETURN INTEGER IS
    zarobki INTEGER;
BEGIN
    SELECT SUM(kwota_koncowa)
    INTO zarobki
    FROM wypozyzenia;
    RETURN zarobki;
END zarobki_z_wypozyzen;
/* zarobki_z_pakietow */
FUNCTION zarobki_z_pakietow RETURN INTEGER IS
    zarobki INTEGER;
BEGIN
    SELECT SUM(kwota_koncowa)
    INTO zarobki
    FROM umowy;
    RETURN zarobki;
END zarobki_z_pakietow;
/* zarobki_z_wypozyzen_miesiac */
FUNCTION zarobki_z_wypozyzen_miesiac(data_start DATE, data_koniec DATE)
RETURN INTEGER IS
    zarobki INTEGER;
BEGIN
    SELECT SUM(kwota_koncowa)

```

```

INTO zarobki

FROM wypozyzzenia

WHERE data_wypozyzzenia BETWEEN data_start AND data_koniec;

RETURN COALESCE(zarobki, 0);

END zarobki_z_wypozyzzen_miesiac;

/* zarobki_z_pakietow_miesiac */

FUNCTION zarobki_z_pakietow_miesiac(data_start DATE, data_koniec DATE)
RETURN INTEGER IS

    zarobki INTEGER;

BEGIN

    SELECT SUM(kwota_koncowa)

    INTO zarobki

    FROM umowy

    WHERE data_zawarcia_umowy BETWEEN data_start AND data_koniec;

    RETURN COALESCE(zarobki, 0);

END zarobki_z_pakietow_miesiac;

/* srednia_ocena_filmu */

FUNCTION srednia_ocena_filmu(id_filmuU NUMBER) RETURN NUMBER IS

    srednia_ocena NUMBER;

BEGIN

    SELECT AVG(ocena)

    INTO srednia_ocena

    FROM recenzja_filmu_i_serialu

    WHERE filmy_i_seriale_id_f_s = id_filmuU;

    RETURN srednia_ocena;

END srednia_ocena_filmu;

/* raport_ogolny */

PROCEDURE raport_ogolny IS

    zarobki_wypozyzzenia INTEGER;

    zarobki_pakiety INTEGER;

BEGIN

```

```


zarobki_wypozyczenia := zarobki_z_wypozyczen();
zarobki_pakiety := zarobki_z_pakietow();

DBMS_OUTPUT.PUT_LINE('Raport o zarobkach:');
DBMS_OUTPUT.PUT_LINE('Zarobki z wypozyczen: ' || zarobki_wypozyczenia);
DBMS_OUTPUT.PUT_LINE('Zarobki z pakietow: ' || zarobki_pakiety);
DBMS_OUTPUT.PUT_LINE('Calkowite zarobki: ' || (zarobki_wypozyczenia +
zarobki_pakiety));

END Raport_ogolny;
/* raport_miesieczny */

PROCEDURE raport_miesieczny IS
    poczatkowa_data DATE;
    koncowa_data DATE;
    miesiac_poczatek DATE;
    miesiac_koniec DATE;
    zar_wyp INTEGER;
    zar_pak INTEGER;

BEGIN
    poczatkowa_data := TO_DATE('2022-01-01', 'YYYY-MM-DD');
    koncowa_data := TO_DATE('2022-12-31', 'YYYY-MM-DD');

    DBMS_OUTPUT.PUT_LINE('RAPORT O ZAROBKACH MIESIECZNYCH:');
    DBMS_OUTPUT.PUT_LINE('xxxxxxxxxxxxxxxxxxxxxx');

    FOR i IN 1..12 LOOP
        miesiac_poczatek := ADD_MONTHS(poczatkowa_data, i - 1);
        miesiac_koniec := LAST_DAY(miesiac_poczatek);

        zar_wyp := zarobki_z_wypozyczen_miesiac(miesiac_poczatek, miesiac_koniec);
        zar_pak := zarobki_z_pakietow_miesiac(miesiac_poczatek, miesiac_koniec);
    

```

```

DBMS_OUTPUT.PUT_LINE('Miesiąc: ' // TO_CHAR(miesiąc_początek,
'MM/YYYY'));

DBMS_OUTPUT.PUT_LINE('Zarobki z wypożyczeń: ' // zar_wyp);
DBMS_OUTPUT.PUT_LINE('Zarobki z pakietów: ' // zar_pak);
DBMS_OUTPUT.PUT_LINE('Całkowite zarobki: ' // (zar_wyp + zar_pak));
DBMS_OUTPUT.PUT_LINE('xxxxxxxxxxxxxxxxxxxxxx');

END LOOP;

END raport_miesieczny;

/* ranking_filmow_i_seriali */

PROCEDURE ranking_filmow_i_seriali IS

BEGIN

DBMS_OUTPUT.PUT_LINE('Ranking filmów:');
DBMS_OUTPUT.PUT_LINE('xxxxxxxxxxxxxxxxxxxxx');

FOR film IN (SELECT f.tytul,
COALESCE(TO_CHAR(srednia_ocena_filmu(f.id_f_s)), 'Film nie został jeszcze
oceniony') AS srednia_ocena
FROM filmy_i_seriali f
ORDER BY TO_NUMBER(srednia_ocena_filmu(f.id_f_s)) DESC) LOOP

DBMS_OUTPUT.PUT_LINE(film.tytul // ' - Średnia ocena: ' ||
film.srednia_ocena);

END LOOP;

END ranking_filmow_i_seriali;

END;
/

```

5.2. Klient_obsługa

Zawarte funkcje:

- Obsluga_liczby
- Obsluga_litery
- Obsluga_symbole
- Czy_dostepny
- Czy_pakiet_premium

Zawarte procedury:

- DodajKlienta
- Wypożycz

Budowa pakietu:

```
CREATE OR REPLACE PACKAGE KLIENT_OBSLUGA IS  
FUNCTION obsluga_liczby(str VARCHAR2) RETURN BOOLEAN;  
FUNCTION obsluga_litery(str VARCHAR2) RETURN BOOLEAN;  
FUNCTION obsluga_symbole(str VARCHAR2) RETURN BOOLEAN;  
FUNCTION czy_dostepny(p_id_filmu NUMBER) RETURN BOOLEAN;  
FUNCTION czy_pakiet_premium(p_id_klienta NUMBER) RETURN BOOLEAN;  
PROCEDURE DodajKlienta(  
    imieU VARCHAR2,  
    nazwiskoU VARCHAR2,  
    wiekU INTEGER,  
    nr_telefonuU INTEGER,  
    emailU VARCHAR2,  
    ulicaU VARCHAR2,  
    adres_kod_pocztowyU VARCHAR2);  
PROCEDURE wypożycz(  
    id_klientaU IN INTEGER,  
    id_filmuU IN INTEGER,  
    rezerwacjaU IN INTEGER DEFAULT 0,  
    platnosc in integer);  
END;  
/
```

```

CREATE OR REPLACE PACKAGE BODY KLIENT_OBSLUGA AS

/* OBSLUGA_LICZBY */

FUNCTION obsluga_liczby(str VARCHAR2) RETURN BOOLEAN IS
BEGIN
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;

/* OBSLUGA_LTERY */

FUNCTION obsluga_litery(str VARCHAR2) RETURN BOOLEAN IS
BEGIN
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;

/* OBSLUGA_SYMBOLE */

FUNCTION obsluga_symbole(str VARCHAR2) RETURN BOOLEAN IS
BEGIN
FOR i IN 1..LENGTH(str) LOOP
    IF SUBSTR(str, i, 1) = '@' THEN
        RETURN TRUE;
    END IF;
END LOOP;

```

```

    RETURN FALSE;
END obsluga_symbole;
/* CZY_DOSTEPNY */
FUNCTION czy_dostepny(p_id_filmu NUMBER) RETURN BOOLEAN IS
    v_ilosc NUMBER;
BEGIN
    SELECT ilosc INTO v_ilosc
    FROM filmy_i_seriale
    WHERE id_f_s = p_id_filmu;

    RETURN v_ilosc > 0;
END czy_dostepny;
/* CZY_PAKIET_PREMIUM */
FUNCTION czy_pakiet_premium(p_id_klienta NUMBER) RETURN BOOLEAN IS
    pakietU VARCHAR2(40);
BEGIN
    SELECT pakiet INTO pakietU
    FROM klient
    WHERE id_klienta = p_id_klienta;

    RETURN pakietU = 'Premium';
END czy_pakiet_premium;
/* DODAJKlienta */
PROCEDURE DodajKlienta(
    imieU VARCHAR2,
    nazwiskoU VARCHAR2,
    wiekU INTEGER,
    nr_telefonuU INTEGER,
    emailU VARCHAR2,
    ulicaU VARCHAR2,

```

```

adres_kod_pocztowyU VARCHAR2 ) IS
brak_imienia EXCEPTION;
brak_nazwiska EXCEPTION;
brak_wieku EXCEPTION;
brak_numeru EXCEPTION;
brak_emailu EXCEPTION;
brak_ulicy EXCEPTION;

adres_id VARCHAR2(50);
max_klient_id NUMBER;
BEGIN
IF (imieU IS NULL) OR obsluga_liczby(imieU) THEN
RAISE brak_imienia;
ELSIF nazwiskoU IS NULL OR obsluga_liczby(nazwiskoU) THEN
RAISE brak_nazwiska;
ELSIF wiekU IS NULL OR obsluga_litery(wiekU) OR obsluga_symbole(wiekU)
THEN
RAISE brak_wieku;
ELSIF nr_telefonuU IS NULL OR obsluga_litery(nr_telefonuU) OR
obsluga_symbole(nr_telefonuU) THEN
RAISE brak_numeru;
ELSIF emailU IS NULL OR NOT obsluga_symbole(emailU) THEN
RAISE brak_emailu;
ELSIF ulicaU IS NULL THEN
RAISE brak_ulicy;
ELSE
SELECT kod_pocztowy INTO adres_id FROM adres WHERE kod_pocztowy =
adres_kod_pocztowyU;
SELECT MAX(id_klienta) INTO max_klient_id FROM klient;
INSERT INTO klient (id_klienta, imie, nazwisko, wiek, nr_telefonu, email, ulica,
adres_kod_pocztowy)

```

```

VALUES (max_klient_id + 1, imieU, nazwiskoU, wiekU, nr_telefonuU, emailU,
ulicaU, adres_kod_poczтовyU);

DBMS_OUTPUT.PUT_LINE('Klient dodany pomyślnie. ');

END IF;

EXCEPTION

WHEN brak_imienia THEN

DBMS_OUTPUT.PUT_LINE('Nie wprowadzono imienia lub wprowadzono
niedozwolone znaki. Uzupełnij dane.');

WHEN brak_nazwiska THEN

DBMS_OUTPUT.PUT_LINE('Nie wprowadzono nazwiska lub wprowadzono
niedozwolone znaki. Uzupełnij dane.');

WHEN brak_wieku THEN

DBMS_OUTPUT.PUT_LINE('Nie wprowadzono wieku lub wprowadzono
niedozwolone znaki. Uzupełnij dane.');

WHEN brak_numeru THEN

DBMS_OUTPUT.PUT_LINE('Nie wprowadzono numeru telefonu lub
wprowadzono niedozwolone znaki. Uzupełnij dane.');

WHEN brak_emailu THEN

DBMS_OUTPUT.PUT_LINE('Nie wprowadzono adresu e-mail lub jest
nieprawidłowy. Uzupełnij dane.');

WHEN brak_ulicy THEN

DBMS_OUTPUT.PUT_LINE('Nie wprowadzono ulicy w adresie lub
wprowadzono niedozwolone znaki. Uzupełnij dane.');

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE('Podany kod pocztowy nie istnieje w tabeli adres. ');

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Wystąpił błąd podczas dodawania klienta');

END DodajKlienta;

/* WYPOZYCZ */

PROCEDURE wypożycz(
    id_klientaU IN INTEGER,
    id_filmuU IN INTEGER,

```

```

rezerwacjaU IN INTEGER DEFAULT 0,
platnosc in integer
) IS
ilosc_dniU NUMBER := 7;
cenaU NUMBER;
data_wypU DATE;
data_oddU DATE;
data_zakU DATE;
data_rezU DATE;
pakietU VARCHAR2(40);
stan_filmuU INTEGER;
data_koncowaU DATE;
wypozyzenia_idU INTEGER;
zla_platnosc exception;
rez_blad exception;
zly_klient exception;
zly_film exception;
BEGIN
IF platnosc NOT IN (1, 2, 3) THEN
RAISE zla_platnosc;
ELSIF rezerwacjaU NOT IN (0,1) THEN
RAISE rez_blad;
ELSIF id_klientaU is NULL or id_klientaU = 0 then
RAISE zly_klient;
ELSIF id_filmuU is NULL or id_filmuU = 0 then
RAISE zly_film;
ELSE
IF czy_dostepny(id_filmuU) THEN
SELECT ilosc INTO stan_filmuU
FROM filmy_i_seriale

```

WHERE id_f_s = id_filmU;

IF czy_pakiet_premium(id_klientaU) THEN

SELECT CASE

WHEN ilosc_dniU > (SELECT

MONTHS_BETWEEN(u.data_zakoczenia_umowy, SYSDATE) + 1

FROM umowy u

WHERE u.klient_id_klienta = id_klientaU

AND SYSDATE BETWEEN u.data_zawarcia_umowy AND
u.data_zakoczenia_umowy)

THEN (SELECT MONTHS_BETWEEN(u.data_zakoczenia_umowy,
SYSDATE) + 1

FROM umowy u

WHERE u.klient_id_klienta = id_klientaU

AND SYSDATE BETWEEN u.data_zawarcia_umowy AND
u.data_zakoczenia_umowy)

ELSE ilosc_dniU

END

INTO ilosc_dniU

FROM dual;

cenaU := 0;

ELSE

SELECT cena INTO cenaU

FROM filmy_i_seriale

WHERE id_f_s = id_filmU;

END IF;

data_wypU := SYSDATE;

data_oddU := SYSDATE + ilosc_dniU;

UPDATE filmy_i_seriale

SET ilosc = stan_filmU - 1

WHERE id_f_s = id_filmU;

SELECT MAX(id_wypożyczenia) + 1 INTO wypożyczenia_idU

FROM wypożyczenia;

*INSERT INTO wypożyczenia (id_wypożyczenia, klient_id_klienta,
administrator_id_ad, filmy_i_seriale_id_f_s, data_wypożyczenia, data_oddania,
kwota_koncowa, rodzaj_platnosci_id_p, status)*

*VALUES (wypożyczenia_idU, id_klientaU, 1, id_filmU, data_wypU, data_oddU,
cenaU, platnosc, 'W TRAKCIE');*

*DBMS_OUTPUT.PUT_LINE('Film wypożyczony! Data oddania: ' ||
TO_CHAR(data_oddU, 'DD.MM.YYYY'));*

ELSE

IF rezerwacjaU = 0 THEN

DBMS_OUTPUT.PUT_LINE('Wpadnij kiedy indziej!');

SELECT MAX(data_koncowa) INTO data_zakU

FROM rezerwacje

WHERE filmy_i_seriale_id_f_s = id_filmU

AND data_koncowa IS NOT NULL;

SELECT MAX(data_oddania) INTO data_oddU

FROM wypożyczenia

WHERE filmy_i_seriale_id_f_s = id_filmU;

IF data_zakU IS NOT NULL AND data_zakU > data_oddU THEN

*DBMS_OUTPUT.PUT_LINE('Ostatnia rezerwacja zakończona dnia: ' ||
TO_CHAR(data_zakU, 'DD.MM.YYYY'));*

ELSE

*DBMS_OUTPUT.PUT_LINE('Ostatnie wypożyczenie zakończone dnia: ' ||
TO_CHAR(data_oddU, 'DD.MM.YYYY'));*

END IF;

ELSIF rezerwacjaU = 1 THEN

```

SELECT MAX(data_koncowa) INTO data_zakU
FROM rezerwacje
WHERE filmy_i_seriale_id_f_s = id_filmuU
AND data_koncowa IS NOT NULL;

SELECT MAX(data_oddania) INTO data_oddU
FROM wypozyzzenia
WHERE filmy_i_seriale_id_f_s = id_filmuU;

IF data_zakU IS NOT NULL AND data_zakU > data_oddU THEN
    data_rezU := data_zakU + 1;
ELSIF data_oddU IS NOT NULL THEN
    data_rezU := data_oddU + 1;
ELSE
    data_rezU := SYSDATE + 1;
END IF;
data_koncowaU := data_rezU + 7;

SELECT MAX(id_rezerwacji) + 1 INTO wypozyzienia_idU
FROM rezerwacje;

INSERT INTO rezerwacje (id_rezerwacji, klient_id_klienta,
filmy_i_seriale_id_f_s, data_poczatkowa, data_koncowa)
VALUES (rezerw.NEXTVAL, id_klientaU, id_filmuU, data_rezU,
data_koncowaU);

DBMS_OUTPUT.PUT_LINE('Film zarezerwowany do ' ||
TO_CHAR(data_koncowaU, 'DD.MM.YYYY'));

END IF;
END IF;
END IF;

EXCEPTION
WHEN zla_platnosc THEN

```

```

DBMS_OUTPUT.PUT_LINE('zla platnosc');

RETURN;

WHEN rez_blad THEN

DBMS_OUTPUT.PUT_LINE('Zle okreslenie rezerwacji! Jezeli chcesz
zarezerwowac przy wypozyczeniu wpisz 1, jezeli nie wpisz 0!');

RETURN;

when zly_klient then

DBMS_OUTPUT.PUT_LINE('Niepoprawne id uzytkownika');

RETURN;

when zly_film then

DBMS_OUTPUT.PUT_LINE('Niepoprawne id filmu');

RETURN;

when OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Cos poszlo nie tak! Sprawdz wpisane dane!');

RETURN;

END wypożycz;

END;

/

```

5.3. Administracja

Zawarte funkcje:

- Czy_umowa_aktywna
- Czy_aktywna

Zawarte procedury:

- Aktualizacja_status_umowy
- Pakiety_klient
- Oddaj

Budowa pakietu:

```

CREATE OR REPLACE PACKAGE ADMINISTRACJA IS

FUNCTION czy_umowa_aktywna(id_umowyU IN INTEGER) RETURN BOOLEAN;
FUNCTION czy_aktywna(id_umowyU IN INTEGER) RETURN BOOLEAN;
PROCEDURE aktualizacja_status_umowy;
PROCEDURE pakiety_klient;

```

```

PROCEDURE oddaj (id_wypozyczeniaU IN INTEGER);
END;
/
CREATE OR REPLACE PACKAGE BODY ADMINISTRACJA AS
/* CZY_UMOWA_AKTYWNA */
FUNCTION czy_umowa_aktywna(id_umowyU IN INTEGER) RETURN BOOLEAN IS
    data_zakonczenia DATE;
BEGIN
    SELECT data_zakoczenia_umowy INTO data_zakonczenia
    FROM umowy
    WHERE id_umowy = id_umowyU;
    IF data_zakonczenia IS NULL OR data_zakonczenia > SYSDATE THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END;
/* CZY_AKTYWNA */
FUNCTION czy_aktywna(id_umowyU IN INTEGER) RETURN BOOLEAN IS
    data_zawarcia_umowy umowy.data_zawarcia_umowy%TYPE;
    aktualna_data DATE;
    roznica NUMBER;
BEGIN
    SELECT data_zawarcia_umowy INTO data_zawarcia_umowy
    FROM umowy
    WHERE id_umowy = id_umowyU;
    aktualna_data := SYSDATE;
    roznica := aktualna_data - data_zawarcia_umowy;
    IF roznica >= 30 THEN
        RETURN TRUE;
    END IF;
END;

```

```

ELSE

RETURN FALSE;

END IF;

END;

/* AKTUALIZACJA_STATUS_UMOWY */

PROCEDURE aktualizacja_status_umowy IS

CURSOR umowy_cursor IS

SELECT *

FROM umowy;

umowa umowy%ROWTYPE;

BEGIN

FOR umowa_rec IN umowy_cursor LOOP

umowa := umowa_rec;

IF czy_aktywna(umowa.id_umowy) THEN

UPDATE umowy

SET status = 'NIEAKTYWNA'

WHERE id_umowy = umowa.id_umowy;

DBMS_OUTPUT.PUT_LINE('Status umowy o ID ' || umowa.id_umowy || '
został zaktualizowany na NIEAKTYWNA.');

ELSE

DBMS_OUTPUT.PUT_LINE('Nie minęło jeszcze 30 dni od zawarcia umowy o
ID ' || umowa.id_umowy);

END IF;

END LOOP;

END;

/* PAKIET_KLIENT */

PROCEDURE pakiet_klient AS

id_umowy umowy.id_umowy%TYPE;
id_klientaU umowy.klient_id_klienta%TYPE;

```

```

rodzaj_umowy umowy.rodzaj_platnosci_id_p%TYPE;
BEGIN
FOR umowa IN (SELECT * FROM umowy) LOOP
    id_umowy := umowa.id_umowy;
    id_klientaU := umowa.klient_id_klienta;
    rodzaj_umowy := umowa.rodzaj_platnosci_id_p;
    IF czy_umowa_aktywna(id_umowy) THEN
        UPDATE klient
        SET pakiet = CASE
            WHEN rodzaj_umowy = 1 THEN 'Podstawowy'
            WHEN rodzaj_umowy = 2 THEN 'Standardowy'
            WHEN rodzaj_umowy = 3 THEN 'Premium'
            ELSE 'Brak'
        END
        WHERE id_klienta = id_klientaU;
    ELSE
        DBMS_OUTPUT.PUT_LINE('BRAK AKTYWNEJ UMOWY DLA Klienta o
ID: ' || id_klientaU);
    END IF;
END LOOP;
END pakiet_klient;
/* ODDAJ */
PROCEDURE oddaj (
    id_wypozyczeniaU IN INTEGER
) IS
    statusU      VARCHAR2(40);
    data_wypozyczeniaU DATE;
    data_dzisiejszaU DATE := SYSDATE;
    cenaU      INTEGER;
    stan_filmuU   INTEGER;
    rezerwacja_idU  INTEGER;

```

```

stan_premiumU    VARCHAR2(10);
ilosc_dniU      INTEGER;
data_oddaniaU   DATE;
id_klientaU     INTEGER;
nowe_id_wypozyczeniaU INTEGER;
rekord_rezerwacje%ROWTYPE;
i NUMBER:=0;
max_id_wypozyczenia INTEGER;

TYPE rezerwacje_historia IS TABLE OF rezerwacje%ROWTYPE;
rezerwacja rezerwacje_historia:=rezerwacje_historia();
zle_id_wypozyczenia exception;

BEGIN
SELECT MAX(id_wypozyczenia) INTO max_id_wypozyczenia FROM wypozyczenia;
IF id_wypozyczeniaU > max_id_wypozyczenia THEN
RAISE zle_id_wypozyczenia;
ELSE
SELECT status, data_wypozyczenia, klient_id_klienta
INTO statusU, data_wypozyczeniaU, id_klientaU
FROM wypozyczenia
WHERE id_wypozyczenia = id_wypozyczeniaU;

IF statusU = 'W TRAKCIE' THEN
IF data_dzisiejszaU - data_wypozyczeniaU > 7 THEN
UPDATE wypozyczenia
SET status = 'ODDANE'
WHERE id_wypozyczenia = id_wypozyczeniaU;

SELECT MIN(id_rezerwacji)
INTO rezerwacja_idU

```

```

FROM rezerwacje
WHERE filmy_i_seriale_id_f_s = (
    SELECT filmy_i_seriale_id_f_s
FROM wypozyzenia
WHERE id_wypozyzenia = id_wypozyzeniaU
);

IF rezerwacja_idU IS NOT NULL THEN
    SELECT klient_id_klienta, filmy_i_seriale_id_f_s
    INTO id_klientaU, stan_filmuU
FROM rezerwacje
WHERE id_rezerwacji = rezerwacja_idU;
FOR rekord IN (SELECT * FROM rezerwacje WHERE id_rezerwacji =
rezerwacja_idU) LOOP
    i:=i+1;
    rezerwacja.EXTEND;
    rezerwacja(i):=rekord;
END LOOP;

DELETE FROM rezerwacje WHERE id_rezerwacji = rezerwacja_idU;

FOR i IN 1..rezerwacja.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE('ID Rezerwacji: ' //
rezerwacja(i).id_rezerwacji);
    DBMS_OUTPUT.PUT_LINE('ID Klienta: ' //
rezerwacja(i).klient_id_klienta);
    DBMS_OUTPUT.PUT_LINE('ID filmu lub serialu: ' //
rezerwacja(i).FILMY_I_SERIALE_ID_F_S);
    DBMS_OUTPUT.PUT_LINE('Data początkowa: ' //
rezerwacja(i).DATA_POCZATKOWA);
    DBMS_OUTPUT.PUT_LINE('Data koncowa: ' //
rezerwacja(i).DATA_KONCOWA);
END LOOP;

```

```

SELECT cena
INTO cenaU
FROM filmy_i_serial
WHERE id_f_s = stan_filmuU;

SELECT pakiet
INTO stan_premiumU
FROM klient
WHERE id_klienta = id_klientaU;

IF stan_premiumU = 'PREMIUM' THEN
    cenaU := 0;

SELECT u.DATA_ZAKOCZENIA_UMOWY - SYSDATE
INTO ilosc_dniU
FROM umowy u
JOIN klient k ON u.klient_id_klienta = k.id_klienta
WHERE k.id_klienta = id_klientaU;

IF ilosc_dniU < 7 THEN
    data_oddaniaU := SYSDATE + ilosc_dniU;
ELSE
    data_oddaniaU := SYSDATE + 7;
END IF;

ELSE
    data_oddaniaU := SYSDATE + 7;
END IF;

SELECT MAX(id_wypozyczenia) + 1
INTO nowe_id_wypozyczeniaU

```

```

    FROM wypożyczania;

    INSERT INTO wypożyczania (id_wypożyczania, klient_id_klienta,
filmy_i_seriale_id_f_s, administrator_id_ad, data_wypożyczania, data_oddania,
kwota_koncowa, rodzaj_platnosci_id_p, status)

        VALUES (nowe_id_wypożyczaniaU, id_klientaU, stan_filmuU, 1, SYSDATE,
data_oddaniaU, cenaU, 2, 'W TRAKCIE');

    ELSE

        UPDATE filmy_i_seriale

        SET ilosc = ilosc + 1

        WHERE id_f_s = (
            SELECT filmy_i_seriale_id_f_s

            FROM wypożyczania

            WHERE id_wypożyczania = id_wypożyczaniaU

        );

    END IF;

    END IF;

    ELSE

        DBMS_OUTPUT.PUT_LINE('Film nie jest wypożyczony.');

    END IF;

    END IF;

    EXCEPTION

    WHEN zle_id_wypożyczania THEN

        DBMS_OUTPUT.PUT_LINE('Zle id wypożyczania');

        RETURN;

    END oddaj;

    END;

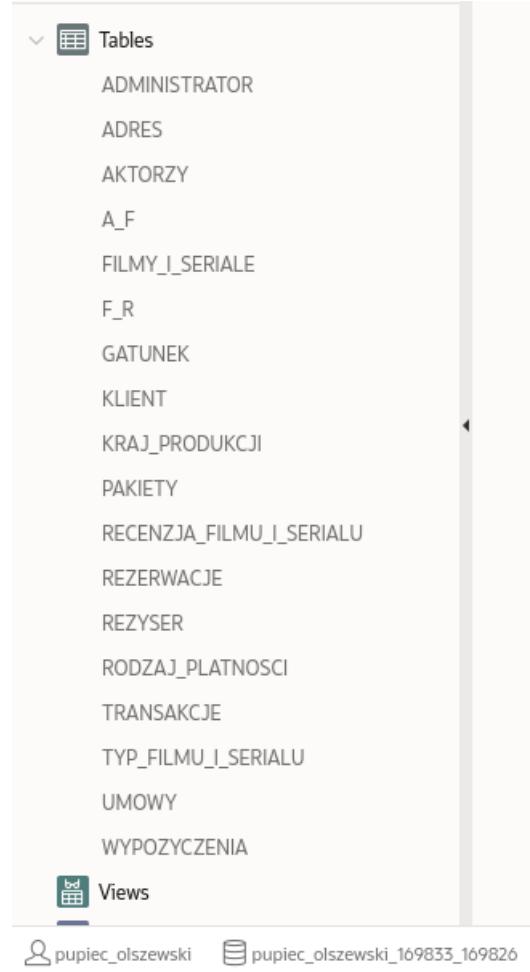
/

```

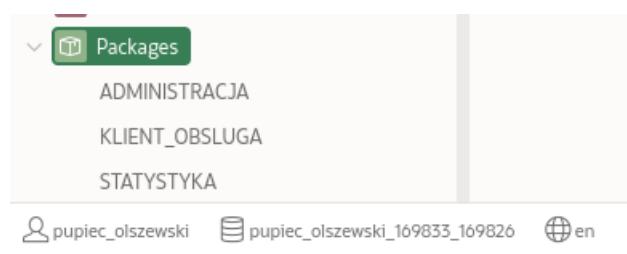
6. APEX

W tym punkcie skupimy się na obsłudze narzędzia, które oferuje nam Oracle czyli Oracle Application Express (APEX). W tym celu zaimportowana zostanie baza danych utworzona podczas realizacji projektu z przedmiotu Bazy Danych a następnie wszystkie poprzednio utworzone funkcje, procedury i pakiety, których działanie zostanie ukazane na utworzonej aplikacji bazodanowej.

6.1. Import bazy danych i pakietów z procedurami i funkcjami



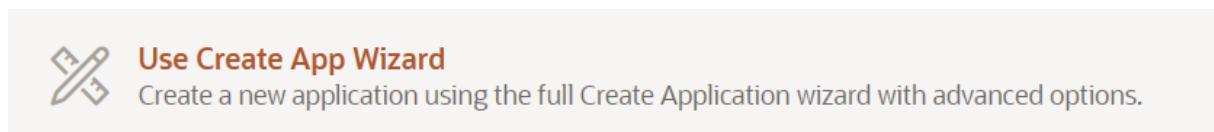
Rysunek 27 Zainportowane tabele



Rysunek 28 Zainportowane pakiety z procedurami i funkcjami

6.2. Tworzenie aplikacji bazodanowej – strona główna, menu.

Po zimportowaniu wszystkich potrzebnych danych przeszliśmy do tworzenia aplikacji bazodanowej. W tym celu wybrany został sposób tworzenia aplikacji na App Wizard:



Rysunek 29 Wybrana opcja tworzenia aplikacji

Następnie do strony głównej aplikacji dodane zostało dziesięć okienek, które po naciśnięciu przekierowują na inną podstronę. W większości przedstawiają najważniejsze tabele zawierające się w bazie danych. Oprócz tego dodane również zostały dwie opcje, które mają przedstawiać funkcjonalności strony z perspektywy klienta jak i administratora.

Rysunek 30 Strona główna

Rysunek 31 Filmy i seriale

Pupiec_Olszewski

Klient Id Klienta ↑	Administrator Id Ad	Filmy I Seriale Id F S	Data Wypożyczenia	Data Oddania	Kwota Koncowa	Rodzaj Płatnosci Id P	Status
Aleksandra	Piotr	Nietkali	2/19/2022	12/28/2022	20	karta	ODDANE
Anna	Joanna	Usmiechnij sie	7/10/2022	7/15/2022	30	karta	ODDANE
Jan	Piotr	Skazanie na Shawshank	12/28/2022	12/30/2022	20	blik	ODDANE
Klaudia	Joanna	Avatar Istota wody	1/12/2022	1/28/2022	30	blik	ODDANE
Marcin	Joanna	Avatar Istota wody	1/28/2022	12/20/2022	30	przelew	ODDANE
Pawel	Piotr	Usmiechnij sie	3/28/2020	12/30/2020	30	przelew	ODDANE
Piotr	Piotr	Skazanie na Shawshank	1/8/2024	1/15/2024	20	karta	W TRAKCIE
Piotr	Piotr	Nietkali	1/8/2024	1/15/2024	20	karta	W TRAKCIE
Waldemar	Piotr	Skazanie na Shawshank	4/1/2020	4/5/2020	20	karta	ODDANE
Waldemar	Piotr	Skazanie na Shawshank	4/1/2020	4/28/2020	20	karta	ODDANE
Zuzanna	Joanna	Usmiechnij sie	12/14/2022	12/15/2022	30	blik	ODDANE

1 - 11

Release 1.0

Go Actions ▾ Reset

Rysunek 32 Wypożyczenia

Pupiec_Olszewski

Klient Id Klienta ↑	Administrator Id A	Data Zawarcia Umowy	Data Zakoczenia Umowy	Kwota Koncowa	Pakiety Id Pakietu	Rodzaj Płatnosci Id P	Status
Aleksandra	Piotr	2/19/2022	3/19/2022	20	Standard	karta	NIEAKTYWNA
Anna	Joanna	7/10/2022	9/10/2022	20	Standard	karta	NIEAKTYWNA
Jan	Piotr	12/20/2023	1/20/2024	60	Premium	blik	NIEAKTYWNA
Klaudia	Joanna	1/12/2022	2/12/2022	60	Premium	blik	NIEAKTYWNA
Marcin	Joanna	1/28/2022	2/28/2022	60	Premium	przelew	NIEAKTYWNA
Pawel	Piotr	3/28/2020	4/28/2020	40	Gold	przelew	NIEAKTYWNA
Waldemar	Piotr	4/1/2020	5/1/2020	60	Premium	karta	NIEAKTYWNA
Zuzanna	Joanna	12/14/2022	1/14/2023	60	Premium	blik	NIEAKTYWNA

1 - 8

Release 1.0

Go Actions ▾ Reset

Rysunek 33 Umowy

Pupiec_Olszewski

Imie ↑	Nazwisko	Wiek	Nr Telefonu	Email	Ulica	Adres Kod Pocztowy	Pakiet
Aleksandra	Mila	17	953,678,731	amila@onet	Wesola	Rzeszow	
Aleksandra	Mila	17	953,678,731	amila@onet	Wesola	Rzeszow	
Ana	Wesolowska	39	678,876,876	awesolowska@onet	Sadowa	Rzeszow	
Ana	Wesolowska	39	678,876,876	awesolowska@onet	Sadowa	Rzeszow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jan	Kowalski	19	517,928,516	jkowalski@onet	Uroczna	Rzeszow	Podstawowy
Jan	Kowalski	19	517,928,516	jkowalski@onet	Uroczna	Rzeszow	
Klaudia	Hula	25	719,465,576	khula@onet	Podkarpacka	Gdansk	
Klaudia	Hula	25	719,465,576	khula@onet	Podkarpacka	Gdansk	
Konrad	Testowy	12,312,312,312	2,131	mail@asd.pl	Karowa	Rzeszow	
Konrad	O	15	1,231,244	mail@mail.pl	Połitechniczna	Gdansk	
Konrad	Testowy	23	2,131	mail@asd.pl	Karowa	Rzeszow	
Marcin	Zielko	21	457,092,656	marcin@onet	Wietrznowo	Katowice	

Release 1.0

Go Actions ▾ Reset

Rysunek 34 Klienci

Rysunek 35 Rezerwacje

Film	Średnia ocena
Avatar Istota wody	4.8
Skazanie na Shawshank	4.2
Nietyklini	1.2

Rysunek 36 Ranking według recenzji

Nazwa	Cena
Gold	40
Premium	60
Standard	20

Rysunek 37 Pakiety

Opis	Klient Id Klienta	Filmy i Seriale Id F S	Ocena	Data Recenzji
Bardzo fakny film	Klaudia	Avatar Istota wody	5.0	12/26/2022
Okropne	Anna	Nietyklini	1.0	11/13/2022
Polecam	Piotr	Skazanie na Shawshank	4.0	12/28/2022
Super!!	Piotr	Avatar Istota wody	4.0	12/12/2022
Świętne	Jadwiga	Skazanie na Shawshank	5.0	12/17/2022
Tragedia	Jan	Skazanie na Shawshank	2.0	11/10/2022

Rysunek 38 Recenzje

Rysunek 39 Klient - interfejs

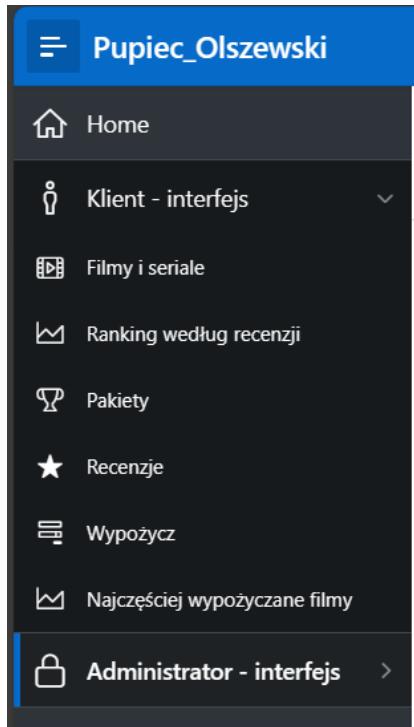
Rysunek 40 Administrator – interfejs

Struktura wewnętrzna:

Rysunek 41 Utworzone elementy - APEX

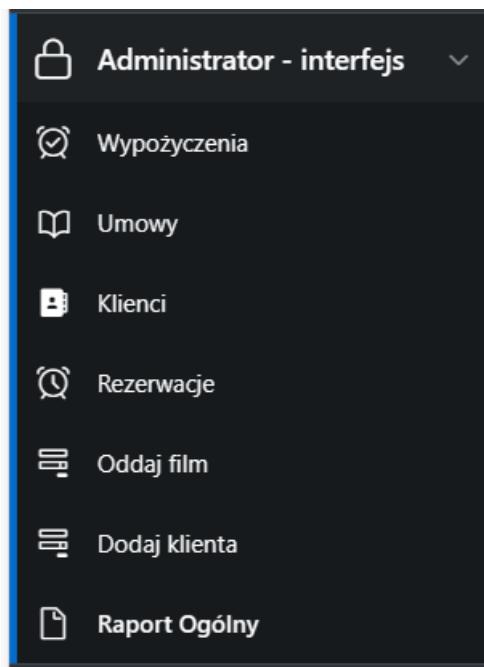
Oprócz strony głównej również zostało utworzone menu z paska rozwijane, które ma podział na dwie sekcje:

- Klient – interfejs



Rysunek 42 Klient - interfejs

- Administrator – interfejs



Rysunek 43 Administrator – interfejs

Do każdej sekcji zostały dodane odpowiednie tabele oraz operacje, które w założeniu miałyby wykonywać dany użytkownik w zależności od roli. W poniższym etapie zostanie przedstawiony proces tworzenia podstron umożliwiających wykonywanie procedur jak i przycisków, które również mają swoje odpowiednie zastosowanie.

6.3. Klient – interfejs. Wykorzystanie procedur i funkcji.

W tym punkcie skupiona zostanie uwaga na wykorzystaniu procedur oraz funkcji w odpowiednich obiektach. Strona Klient – interfejs zawiera 5 przycisków:

- a) Filmy i seriale
- b) Pakiety
- c) Recenzje
- d) Najczęściej wypożyczane filmy
- e) Ranking według recenzji
- f) Wypożycz film

The screenshot shows a user interface for a client application. At the top, there is a blue header bar with the user's name "Pupiec_Olszewski" and an email address "169826@stud.prz.edu.pl". Below the header, the main title "Klient - interfejs" is displayed. The interface is divided into sections:

- Przydatne informacje**: A sidebar containing several buttons:
 - Filmy i seriale
 - Pakiety
 - Recenzje

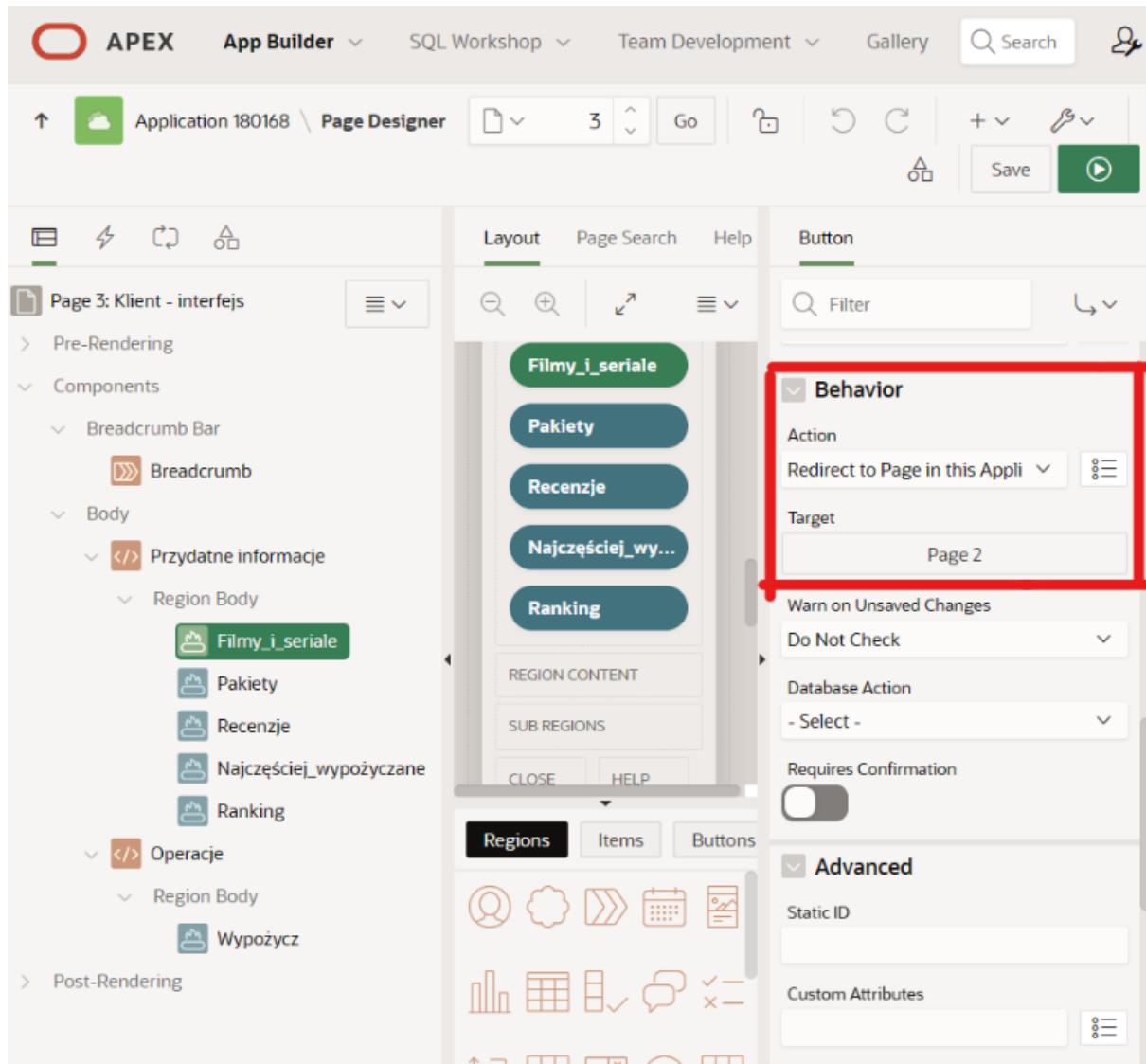
Najczęściej wypożyczane filmy

Ranking według recenzji
- Operacje**: A sidebar containing one button:

Wypożycz film

Rysunek 44 Klient – interfejs

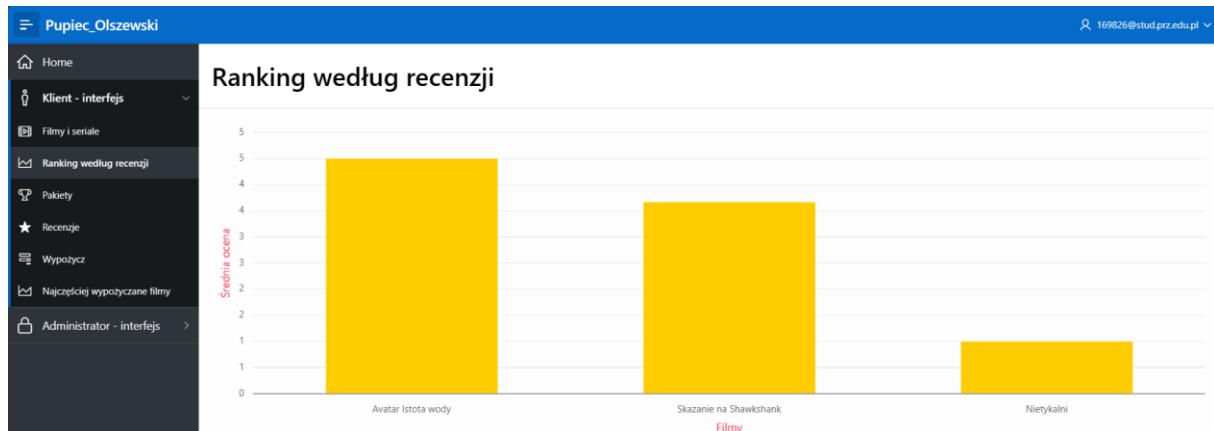
Każdy z powyższych przycisków przekierowuje użytkownika do kolejnej podstrony z danymi. W tym celu by opisana operacja była możliwa przyciskom ustawione zostały odpowiednie wartości funkcji:



Rysunek 45 Opcja pozwalająca na przekierowanie na kolejną stronę

- Ranking według recenzji

Jest to wykres przedstawiający średnią ocenę danego filmu według recenzji.



Rysunek 46 Średnia ocena filmu – wykres

W celu utworzenia tego typu wykresu został wykorzystany specjalny szablon tworzenia strony pozwalający na statystyczne wykorzystanie danych w celu wizualizacji ich na wykresie. Jest to wykres słupkowy. Poniżej została przedstawiona struktura wraz z kodem, który umożliwia pobranie danych do utworzenia wykresu:

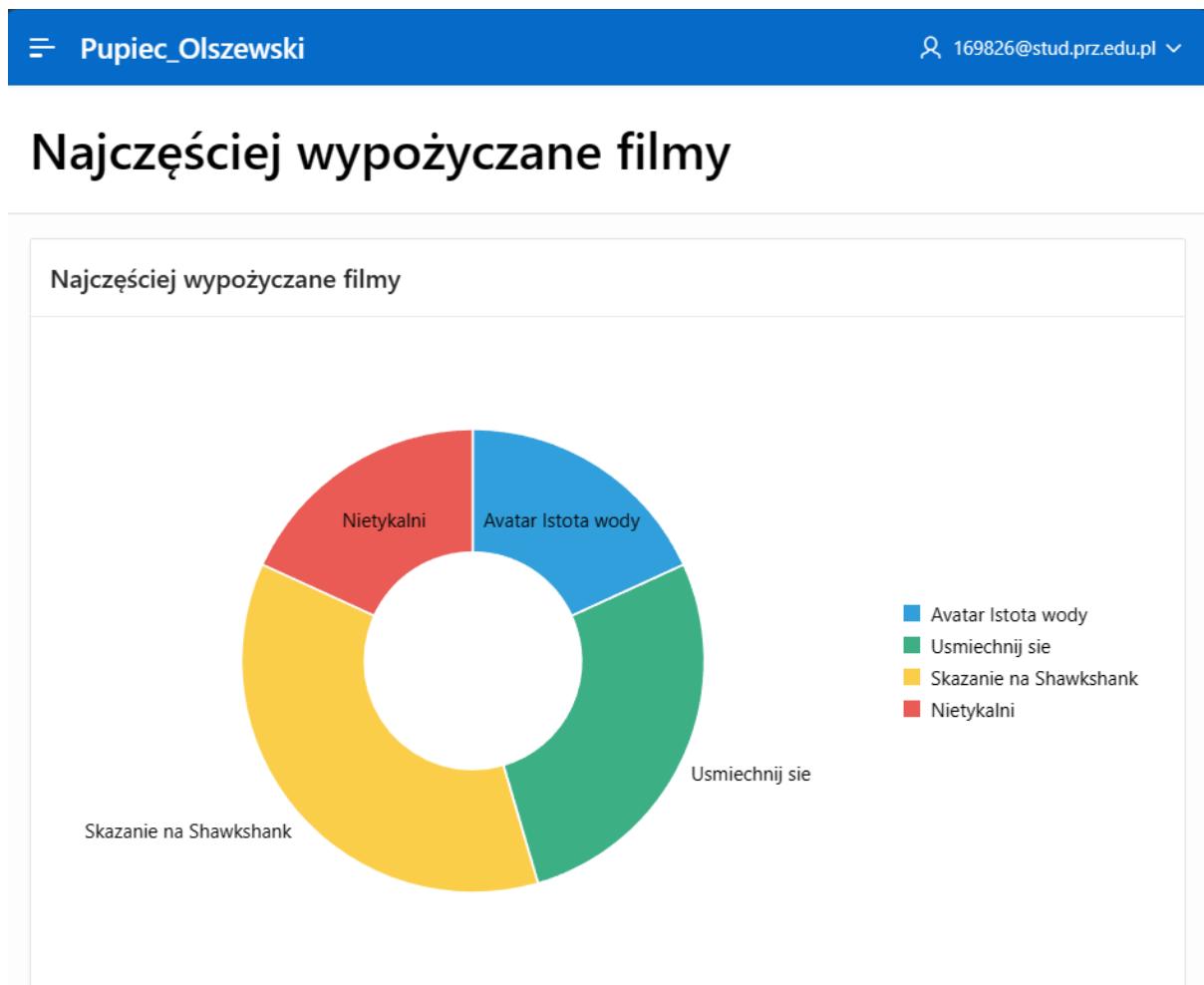
```

SELECT FIS.TYTUL AS NAZWA_FILMU,
       AVG(RFS.OCENA) AS SREDNIA_OCENA
  FROM RECENZJA_FILMU_I_SERIALU RFS
  JOIN FILMY_I_SERIALE FIS ON
    RFS.FILMY_I_SERIALE_ID_F_S =
      FIS.ID_F_S
 WHERE RFS.OCENA IS NOT NULL
 GROUP BY FIS.TYTUL
 ORDER BY SREDNIA_OCENA DESC;
  
```

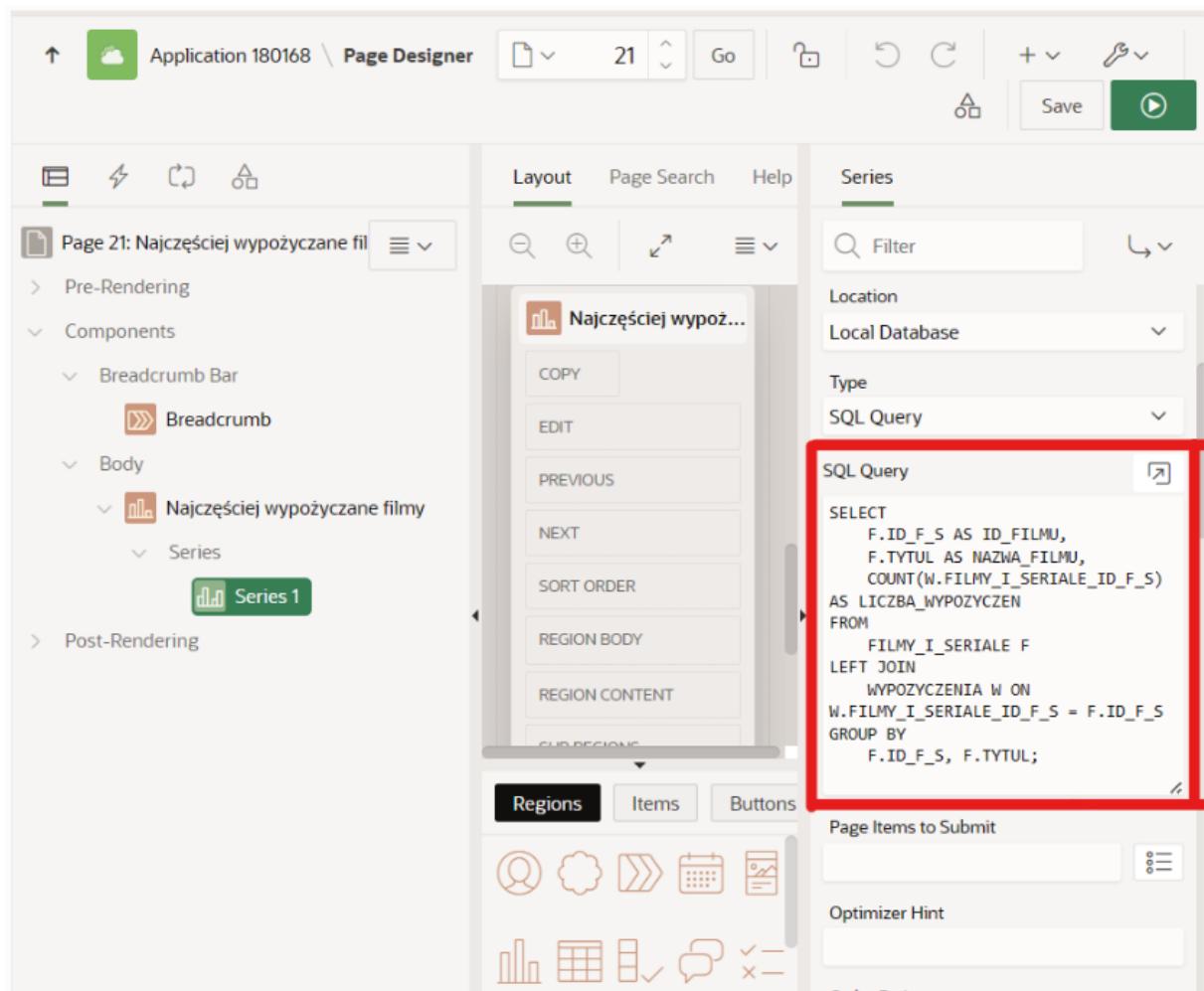
Rysunek 47 Struktura oraz kod wybierający dane - Ranking według recenzji

- Najczęściej wypożyczane filmy

Jest to podstrona przedstawiająca wykres kołowy zawierający informacje o ilości wypożyczeń danego filmu. W prosty sposób wizualizuje on jaki film był wypożyczany najczęściej. Poniżej przedstawiona zostaje struktura oraz kod pobierający dane do wykresu.



Rysunek 48 Najczęściej wypożyczane filmy - wykres



Rysunek 49 Struktura oraz kod wybierający dane - Najczęściej wypożyczane filmy

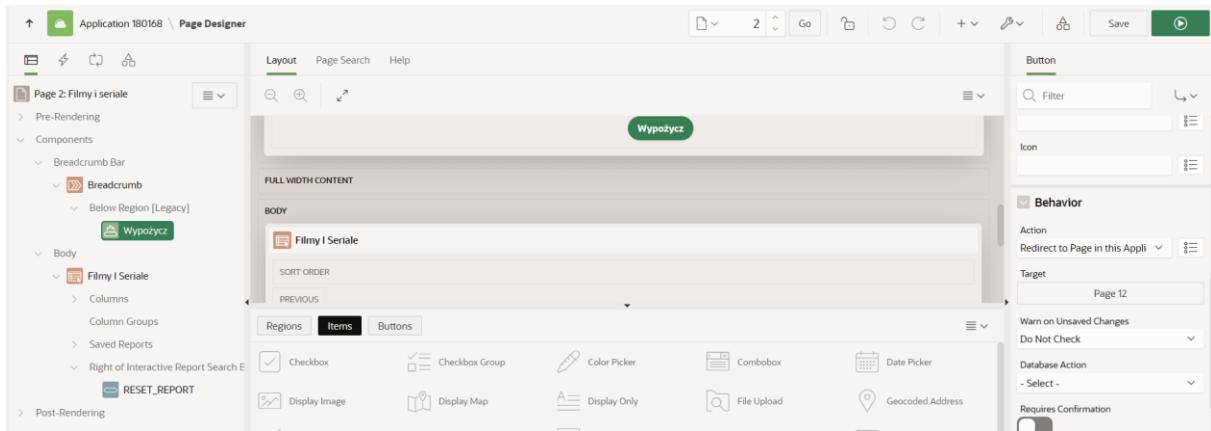
- Filmy i seriale, Pakiety, Recenzje

Każda z tych stron przedstawia zawartość tabeli wybraną z bazy danych. Dzięki temu użytkownik w prosty sposób może przeczytać recenzje na temat danego filmu, sprawdzić bibliotekę wypożyczalni a także pakiety jakie ma w swojej ofercie. Poniżej zostaje przedstawiona wizualna część tych stron a także ich struktura:

The screenshot shows the 'Filmy i seriale' page. The sidebar menu includes 'Home', 'Klient - interfejs', 'Filmy i seriale' (selected), 'Ranking według recenzji', 'Pakiety', 'Recenzje', 'Wypożycz', 'Najczęściej wypożyczane filmy', and 'Administrator - interfejs'. The main content area displays a table with columns: Tytuł, Opis, Cena, Gatunek Id Gatunku, Typ Filmu I Serialu Id T, Kraj Produkcji Id Kraju, Administrator Id A, Pakiety Id Pakietu, and Ilosc. The table data is as follows:

Tytuł	Opis	Cena	Gatunek Id Gatunku	Typ Filmu I Serialu Id T	Kraj Produkcji Id Kraju	Administrator Id A	Pakiety Id Pakietu	Ilosc
Avatar Istota wody	Pandorę znów napada wroga korporacja w poszukiwaniu cennych mineralów. Jack i Neytiri wraz z rodziną zmuszeni są opuścić wioskę i szukać pomocy u innych plemion zamieszkujących planetę.	30	Fantasy	Premiera	USA	Piotr	Premium	1.0
Nietynalni	Sparalizowany milioner zatrudnia do opieki młodego chłopaka z przedmieścia, który właśnie wyszedł z więzienia.	20	Dramat	Klasyk	Francja	Joanna	Gold	1.0
Skazanie na Shawshank	Adaptacja opowiadania Stephena Kinga. Niesłusznie skazany na dożywocie bankier, stara się przetrwać w brutalnym, więziennym świecie.	20	Dramat	Klasyk	USA	Joanna	Standard	2.0
Usmiechnij sie	Po tym, jak dr Rose Carter bierze udział w traumatycznym zdarzeniu z udziałem pacjentki, wokół niej zaczynają działać się niewyobrażalne rzeczy	30	Horror	Premiera	USA	Piotr	Premium	4.0

Rysunek 50 Filmy i seriale



Rysunek 51 Page designer - Filmy i seriale

Na powyższej stronie – Filmy i seriale – znajduje się również dodatkowy przycisk, który umożliwia przejście do strony umożliwiającej wypożyczenie filmu przez użytkownika.

Nazwa	Cena
Gold	40
Premium	60
Standard	20

Rysunek 52 Pakiety

Rysunek 53 Page designer - Pakiety

Strona ta przedstawia pakiety, które są dostępne w wypożyczalni.

Opis	Klient Id Klienta	Filmy I Seriali Id F S	Ocena	Data Recenzji
Bardzo fakny film	Klaudia	Avatar Istota wody	5.0	12/26/2022
Okropne	Anna	Nietykalni	1.0	11/13/2022
Polecam	Piotr	Skazanie na Shawshank	4.0	12/28/2022
Super!!	Piotr	Avatar Istota wody	4.0	12/12/2022
Świetnie	Jadwiga	Skazanie na Shawshank	5.0	12/17/2022
Tragedia	Jan	Skazanie na Shawshank	2.0	11/10/2022

Rysunek 54 Recenzje

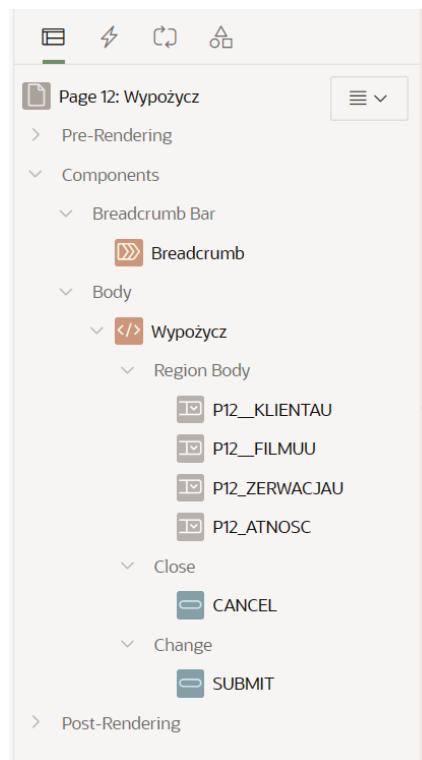
Rysunek 55 Page designer - Recenzje

Powyższa strona reprezentuje recenzje dodane przez użytkowników do danego filmu.

- Wypożycz

Przedstawiona poniżej strona została utworzona na podstawie procedury opartej na podanych parametrach. Daje ona możliwość klientowi wypożyczenia filmu czy też jego rezerwacji. Wszystkie pola utworzone zostały jako lista rozwijana co pomaga w uniknięciu błędów, które mogą wystąpić przy samodzielny wpisywaniu wartości. Oprócz tego zawarta została również obsługa wyjątków co zostanie przedstawione dla wartości odpowiadających id klienta, gdzie jako jedynej wartości można wpisać 0 co zostało wykonane specjalnie w celu ukazania poprawności działania obsługi wyjątków. Reszta wartości można wybrać z ograniczonych list rozwijanych do rzeczywistych wartości.

Rysunek 56 Wypożycz



Rysunek 57 Widok renderingu - Wypożycz

Przykład listy rozwijanej:

Wypożycz

Wypożycz

ID Klienta

ID Filmu

- Avatar Istota wody
- Avatar Istota wody
- Nietykalni
- Skazanie na Shawshank
- Usmiechnij się**
- Rodzaj płatności
- blik

Cancel

Submit

Rysunek 58 Przedstawienie listy rozwijanej

The screenshot shows the Oracle APEX Page Designer interface. On the left, the page structure is defined with regions like 'Breadcrumb Bar' and 'Wypożycz'. The main area displays a form with fields for 'PI2_FILMUU', 'PI2_ZERWACJAU', and 'PI2_ATNOSC'. Below these is a 'REGION CONTENT' section containing buttons for 'CLOSE', 'HELP', 'DELETE', 'CHANGE', 'CANCEL', and 'SUBMIT'. To the right, the 'Page Item' settings for 'PI2_FILMUU' are being configured. A red box highlights the 'List of Values' section, which is set to 'Shared Component' with the identifier 'FILMY_U_SERIALE.TYTUL'. Other settings shown include 'Display Extra Values' (on) and 'Display Null Value' (off). At the bottom, toolbars for 'Regions', 'Items', and 'Buttons' are visible, along with various input field components.

Rysunek 59 Sposób wykonania listy rozwijanej

Oprócz ustawień listy wartości typ obiektu został ustawiony jako Select List.

The screenshot shows the Oracle Application Express Page Designer interface. The left sidebar contains navigation links: After Submit, Validating, Processing (selected), Processes (with a 'Run Stored Procedure' button), After Processing, Branches, and Ajax Callback. The main area displays a process configuration page for a step named 'Wypożycz'. The configuration includes sections for COPY, EDIT, PREVIOUS, NEXT, SORT ORDER, and REGION BODY. Below this, there are tabs for Regions, Items (selected), and Buttons. Under the Items tab, various input types are listed: Checkbox, Checkbox Group, Color Picker, Combobox, Date Picker, Display Image, Display Map, Display Only, File Upload, Geocoded Address, Hidden, Image Upload, List Manager, Markdown Editor, and Number Field. On the right side, there are panels for Process (with a Filter button), Source (Location: Local Database, Language: PL/SQL), and PL/SQL Code (containing a block of PL/SQL code). The bottom right corner shows an Execution panel.

Rysunek 60 Page designer - przedstawienie wyjątków

Code Editor - PL/SQL Code

```

1 declare
2   zla_platnosc exception;
3   rez_blad exception;
4   zly_klient exception;
5   zly_film exception;
6 begin
7   "#OWNER#"."KLIENT_OBSLUGA"."WYPOZYCZ"(
8     "ID_KLIENTAU" => :P12_KLIENTAU,
9     "ID_FILMUU" => :P12_FILMUU,
10    "REZERWACJAU" => :P12_ZERWACJAU,
11    "PLATNOSC" => :P12_ATNOSC);
12
13  IF :P12_ATNOSC NOT IN (1, 2, 3) THEN
14    | RAISE zla_platnosc;
15  ELSIF :P12_ZERWACJAU NOT IN (0,1) THEN
16    | RAISE rez_blad;
17  ELSIF :P12_KLIENTAU is NULL or :P12_KLIENTAU = 0 then
18    | RAISE zly_klient;
19  ELSIF :P12_FILMUU is NULL or :P12_FILMUU = 0 then
20    | RAISE zly_film;
21  end if;
22  EXCEPTION
23  WHEN zla_platnosc THEN
24    RAISE_APPLICATION_ERROR(-20001,'zla platnosc');

```

Rysunek 61 Kod pozwalający na wywoływanie własnych wyjątków

Efekt obsługi błędów:

Wypożycz

ID Klienta

ID Filmu
Nietykalni

Czy rezerwacja?
Tak

Rodzaj płatności
przelew

Cancel Submit

1 error has occurred
ORA-20003: Niepoprawne id użytkownika ⓘ

Rysunek 62 Obsługa błędów

Po przedstawieniu struktury wewnętrznej strony zostanie przedstawiony efekt wykonania procedury:

Wypożycz

Wypożycz

ID Klienta
8

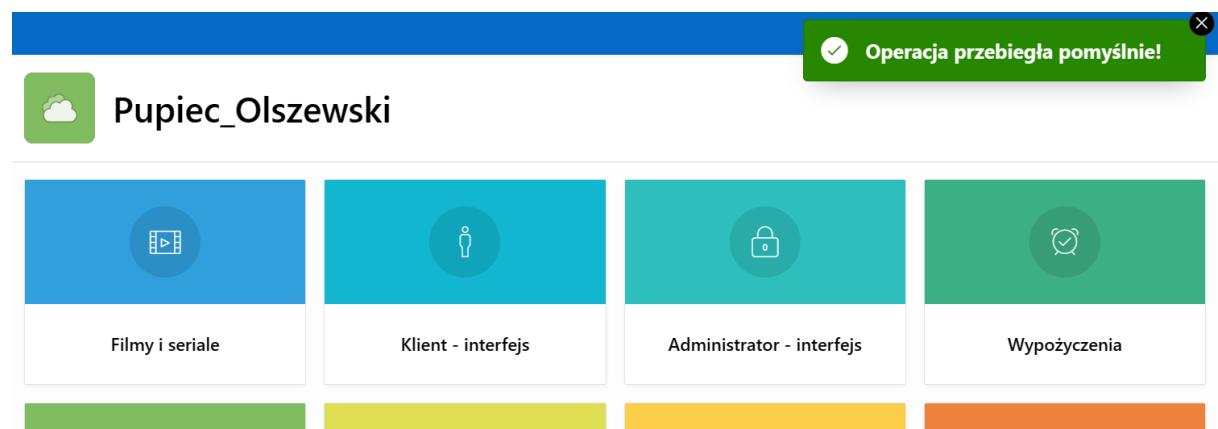
ID Filmu
Nietykalni

Czy rezerwacja?
Tak

Rodzaj płatności
przelew

Cancel Submit

Rysunek 63 Wprowadzone dane



Rysunek 64 Powiadomienie o udanej operacji

Wypożyczenia

Klient Id Klienta	Administrator Id Ad	Filmy I Seriale Id F S	Data Wypożyczenia	Data Oddania	Kwota Koncowa	Rodzaj Płatnosc Id P	Status
Aleksandra	Piotr	Nietykalni	1/9/2024	1/16/2024	20	przelew	W TRAKCIE
Aleksandra	Piotr	Nietykalni	2/19/2022	12/28/2022	20	karta	ODDANE
Anna	Joanna	Usmiechnij sie	7/10/2022	7/15/2022	30	karta	ODDANE
Jan	Piotr	Skazanie na Shawshank	12/28/2022	12/30/2022	20	blik	ODDANE
Klaudia	Joanna	Avatar Istota wody	1/12/2022	1/28/2022	30	blik	ODDANE
Marcin	Joanna	Avatar Istota wody	1/28/2022	12/20/2022	30	przelew	ODDANE
Pawel	Piotr	Usmiechnij sie	3/28/2020	12/30/2020	30	przelew	ODDANE

Rysunek 65 Wynik operacji

Następnie w celu sprawdzenia czy w wypadku braku odpowiedniej ilości filmów na stanie doda rezerwacje przeprowadzone zostało kolejne wypożyczenie tylko na innego użytkownika.

Rezerwacje

Rezerwacje			
Klient Id Klienta ↑↓	Filmy I Seriale Id F S	Data Początkowa	Data Koncowa
Anna	Nietypalni	1/17/2024	1/24/2024
1 - 1			

Rysunek 66 Poprawne dodanie rezerwacji

6.4. Administrator – interfejs. Wykorzystanie procedur i funkcji.

W tym punkcie skupiona zostanie uwaga na wykorzystaniu procedur oraz funkcji w odpowiednich obiektach. Strona Administrator– interfejs zawiera 5 przycisków:

- a) Wypożyczenia
- b) Umowy
- c) Klienci
- d) Rezerwacje
- e) Oddaj film
- f) Dodaj klienta
- g) Raport ogólny



Tabele

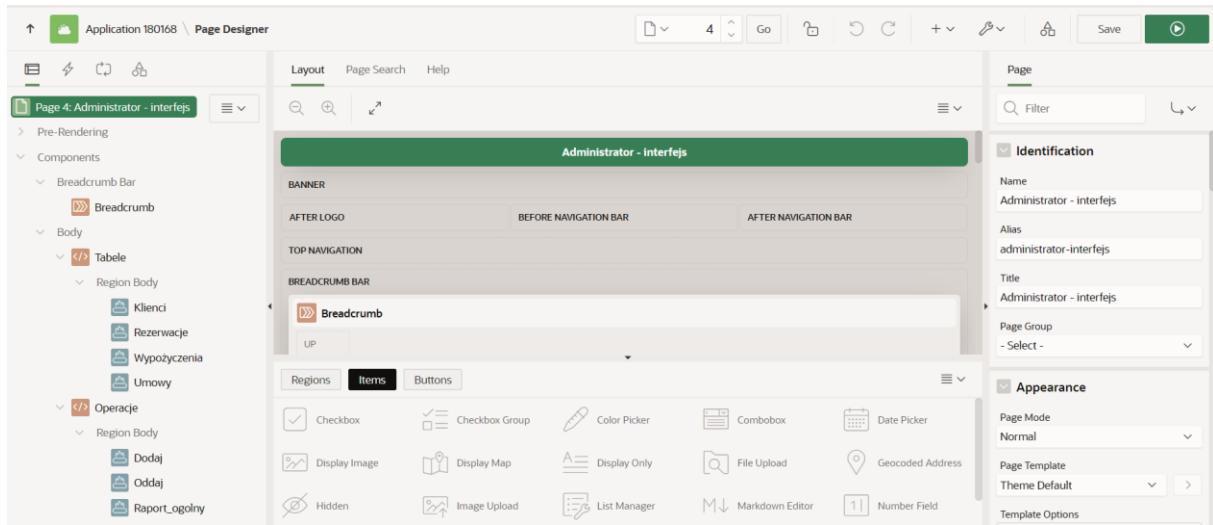


Operacje



Rysunek 67 Strona Administrator – interfejs

Powyzsza strona zawiera 7 przycisków. Każdy z nich wykonuje przekierowanie użytkownika na odpowiednią stronę, w celu wykonania tych operacji została użyta ta sama metoda co poprzednio w Klient - interfejs.



Rysunek 68 Page designer - Administrator – interfejs

W sekcji ‘tabele’ ponownie mamy przedstawione utworzone na potrzeby aplikacji tabele zawierające odpowiednie wartości z zainportowanej bazy danych.

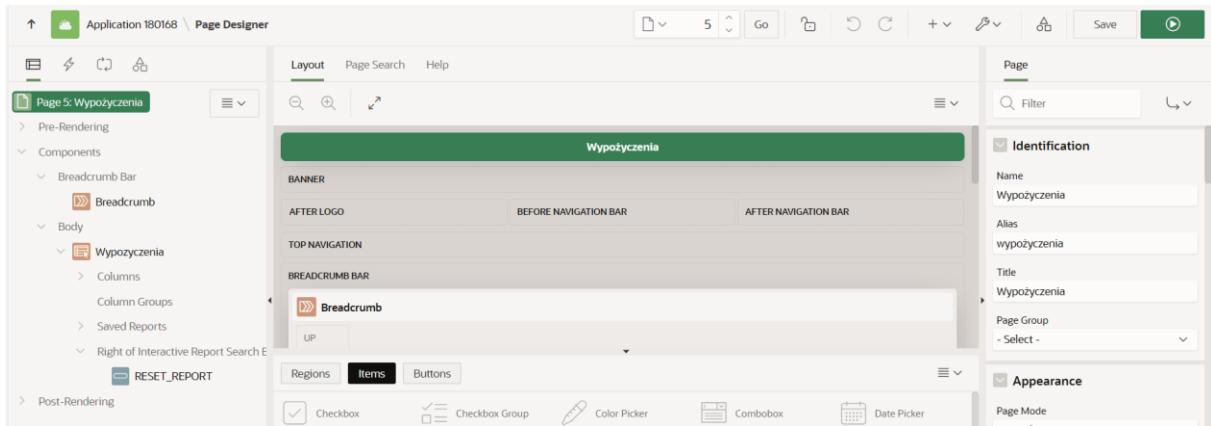
- Wypożyczenia

W tym punkcie przedstawione zostaną strony utworzone na podstawie tabeli oraz procedury, które wykonują na nich dane operacje.

Wypożyczenia

Actions		Go	Q				
Klient Id Klienta ↑=	Administrator Id Ad	Filmy I Seriale Id F S	Data Wypożyczenia	Data Oddania	Kwota Koncowa	Rodzaj Platnosci Id P	Status
Aleksandra	Piotr	Nietynalni	1/9/2024	1/16/2024	20	przelew	W TRAKCIE
Aleksandra	Piotr	Nietynalni	2/19/2022	12/28/2022	20	karta	ODDANE
Anna	Joanna	Usmiechnij sie	7/10/2022	7/15/2022	30	karta	ODDANE
Jan	Piotr	Skazanie na Shawshank	12/28/2022	12/30/2022	20	blik	ODDANE
Klaudia	Joanna	Avatar Istota wody	1/12/2022	1/28/2022	30	blik	ODDANE
Marcin	Joanna	Avatar Istota wody	1/28/2022	12/20/2022	30	przelew	ODDANE
Pawel	Piotr	Usmiechnij sie	3/28/2020	12/30/2020	30	przelew	ODDANE
Piotr	Piotr	Skazanie na Shawshank	1/8/2024	1/15/2024	20	karta	W TRAKCIE
Piotr	Piotr	Nietynalni	1/8/2024	1/15/2024	20	karta	W TRAKCIE
Waldemar	Piotr	Skazanie na Shawshank	4/1/2020	4/5/2020	20	karta	ODDANE
Waldemar	Piotr	Skazanie na Shawshank	4/1/2020	4/28/2020	20	karta	ODDANE
Zuzanna	Joanna	Usmiechnij sie	12/14/2022	12/15/2022	30	blik	ODDANE

Rysunek 69 Wypożyczenia



Rysunek 70 Page designer - wypożyczenia

Powyższa strona przedstawia wypożyczenia klientów, datę oddania, status, datę wypożyczenia, nazwę filmu, rodzaj płatności oraz kwotę do zapłaty.

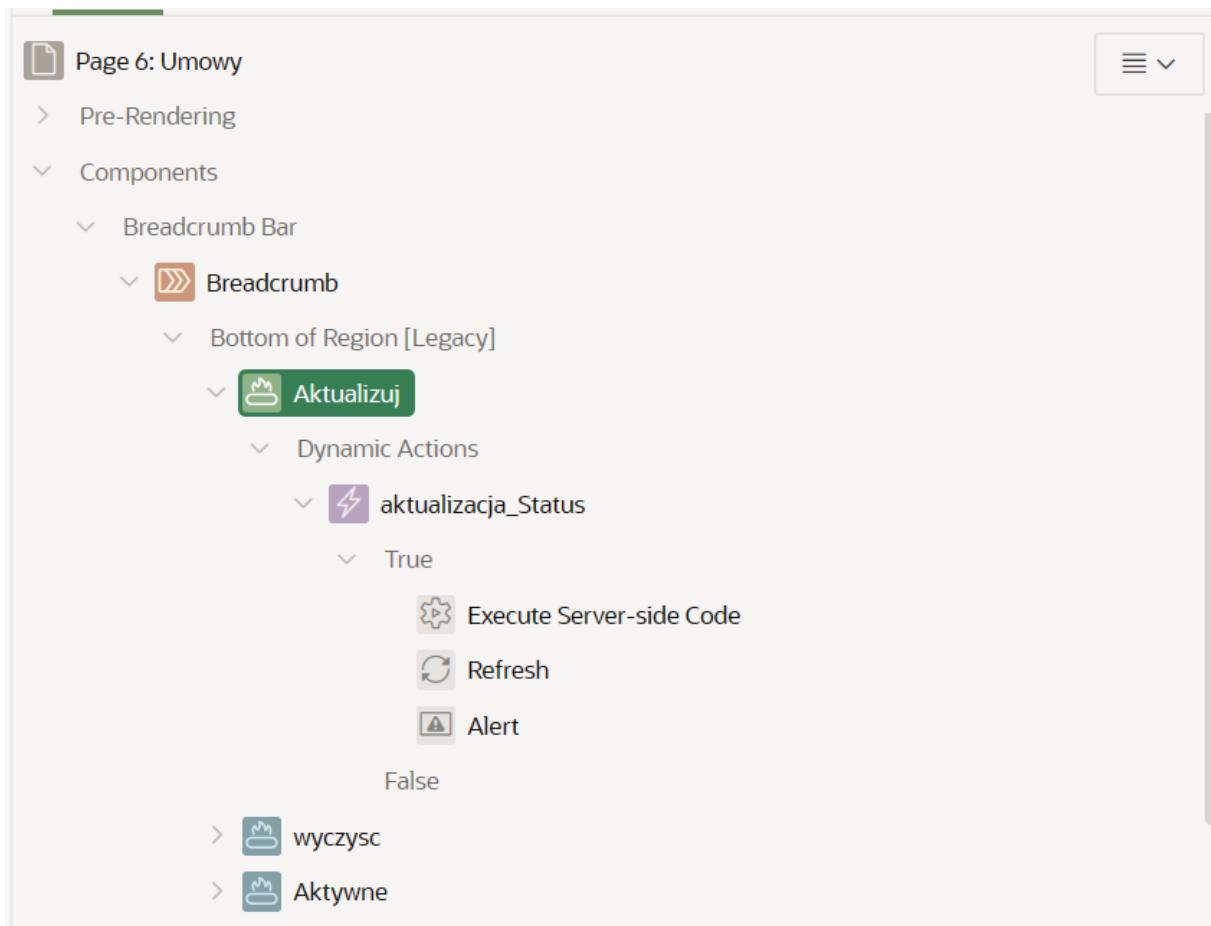
- Umowy

Umowy

Umowy									
							Aktualizuj status	Wyczść	Ustaw aktywne
<input type="text"/> Q		<input type="button" value="Go"/>	<input type="button" value="Actions"/>		<input type="button" value="Reset"/>				
Klient Id Klienta	Administrator Id A	Data Zawarcia Umowy	Data Zakoczenia Umowy	Kwota Koncowa	Pakiety Id Pakietu	Rodzaj Platnosci Id P	Status		
Aleksandra	Piotr	2/19/2022	3/19/2022	20	Standard	karta			
Anna	Joanna	7/10/2022	9/10/2022	20	Standard	karta			
Jan	Piotr	12/20/2023	1/20/2024	60	Premium	blik			
Klaudia	Joanna	1/12/2022	2/12/2022	60	Premium	blik			
Marcin	Joanna	1/28/2022	2/28/2022	60	Premium	przelew			
Pawel	Piotr	3/28/2020	4/28/2020	40	Gold	przelew			
Waldemar	Piotr	4/1/2020	5/1/2020	60	Premium	karta			
Zuzanna	Joanna	12/14/2022	1/14/2023	60	Premium	blik			

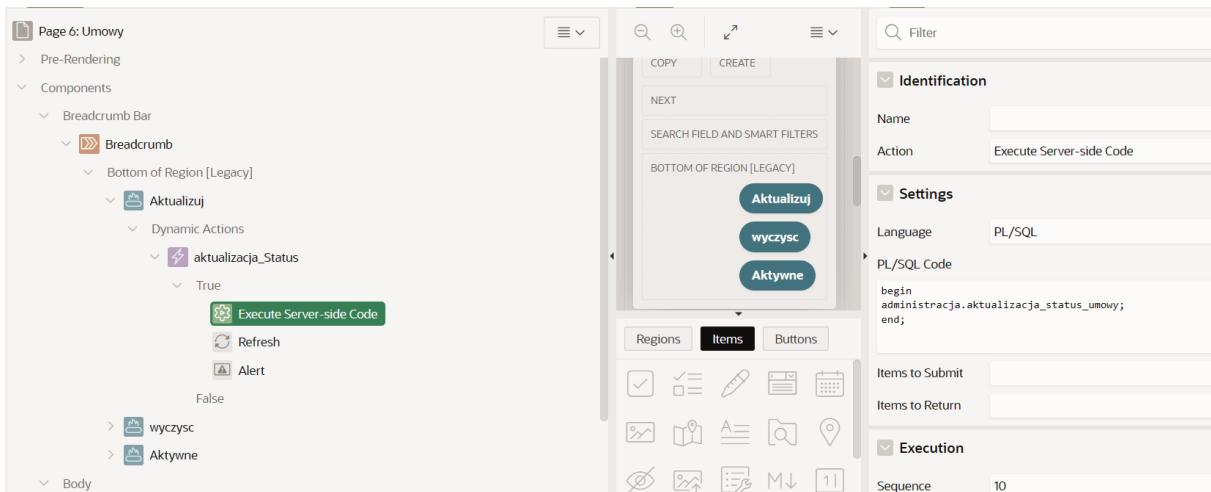
Rysunek 71 Umowy

Kolejnym elementem jest tabela umowy. W tym przypadku oprócz samych wartości znajdują się również 3 przyciski. Jeden z nich wykonuje procedurę a dwa kolejne mają charakter pomocniczy, które pozwalają na przedstawienie działania procedury, która ustawia status umowy na odpowiedni w zależności czy jest ona nadal aktywna. Poniżej przedstawiony zostanie sposób implementacji tych czynności.

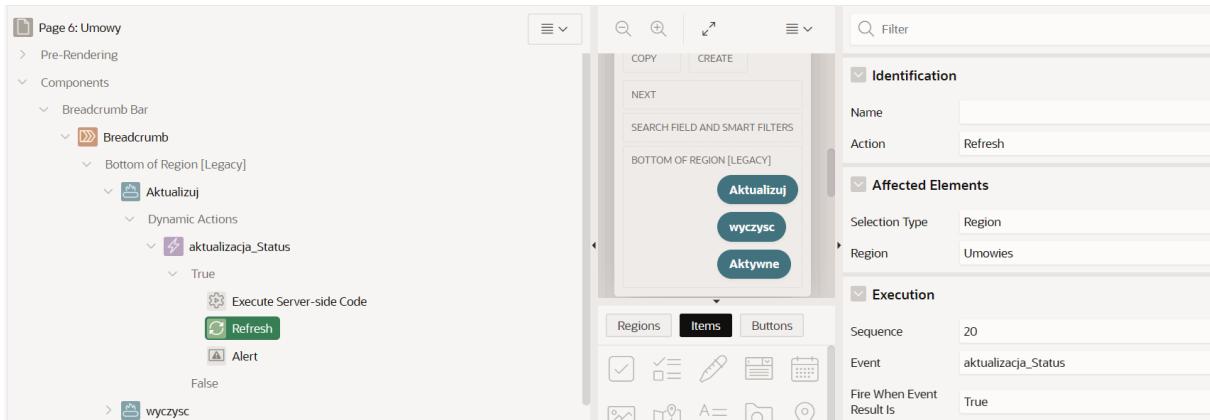


Rysunek 72 Przycisk - Aktualizuj status

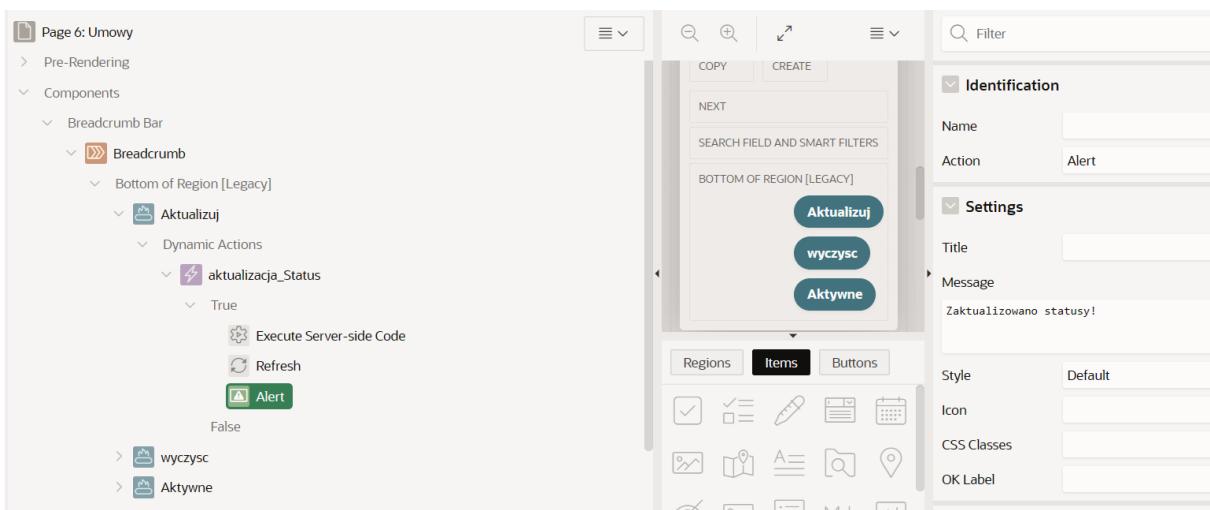
Do przycisku została dodana dynamiczna akcja, w której zostaje wywołana procedura z pakietu ADMINISTRACJA. Oprócz tego dodany został również proces odświeżania tabeli po wykonaniu operacji a także wyświetlająca się informacja o powodzeniu wykonanych czynności.



Rysunek 73 Dynamiczna akcja - wykonanie procedury



Rysunek 74 Akcja odświeżania



Rysunek 75 Pojawiający się alert

Wykonanie:

Umowy

Aktualizuj status Wyczść Ustaw aktywne

Klient Id Klienta	Administrator Id A	Data Zawarcia Umowy	Data Zakoczenia Umowy	Kwota Koncowa	Pakiety Id Pakietu	Rodzaj Platnosci Id P	Status
Aleksandra	Piotr	2/19/2022	3/19/2022	20	Standard	karta	
Anna	Joanna	7/10/2022	9/10/2022	20	Standard	karta	
Jan	Piotr	12/20/2023	1/20/2024	60	Premium	blik	
Klaudia	Joanna	1/12/2022	2/12/2022	60	Premium	blik	
Marcin	Joanna	1/28/2022	2/28/2022	60	Premium	przelew	
Pawel	Piotr	3/28/2020	4/28/2020	40	Gold	przelew	
Waldemar	Piotr	4/1/2020	5/1/2020	60	Premium	karta	
Zuzanna	Joanna	12/14/2022	1/14/2023	60	Premium	blik	

Rysunek 76 Tabela przed zmianą

Na zrzucie ekranu powyżej widzimy jedną aktywną umowę, której po wykonaniu procedury status zmieni się na poprawny.

Umowy							
				Aktualizuj status Wyczysć Ustaw aktywne			
<input type="text"/> Actions ▾		Go		Reset			
Klient Id Klienta ↑	Administrator Id A	Data Zawarcia Umowy	Data Zakoczenia Umowy	Kwota Koncowa	Pakiety Id Pakietu	Rodzaj Platnosci Id P	Status
Aleksandra	Piotr			20	Standard	karta	NIEAKTYWNA
Anna	Joanna			20	Standard	karta	NIEAKTYWNA
Jan	Piotr			60	Premium	blik	
Klaudia	Joanna			60	Premium	blik	NIEAKTYWNA
Marcin	Joanna	1/28/2022	2/28/2022	60	Premium	przelew	NIEAKTYWNA
Pawel	Piotr	3/28/2020	4/28/2020	40	Gold	przelew	NIEAKTYWNA
Waldemar	Piotr	4/1/2020	5/1/2020	60	Premium	karta	NIEAKTYWNA
Zuzanna	Joanna	12/14/2022	1/14/2023	60	Premium	blik	NIEAKTYWNA

Rysunek 77 Powiadomienie o wykonaniu operacji

Umowy							
				Aktualizuj status Wyczysć Ustaw aktywne			
<input type="text"/> Actions ▾		Go		Reset			
Klient Id Klienta ↑	Administrator Id A	Data Zawarcia Umowy	Data Zakoczenia Umowy	Kwota Koncowa	Pakiety Id Pakietu	Rodzaj Platnosci Id P	Status
Aleksandra	Piotr	2/19/2022	3/19/2022	20	Standard	karta	NIEAKTYWNA
Anna	Joanna	7/10/2022	9/10/2022	20	Standard	karta	NIEAKTYWNA
Jan	Piotr	12/20/2023	1/20/2024	60	Premium	blik	
Klaudia	Joanna	1/12/2022	2/12/2022	60	Premium	blik	NIEAKTYWNA
Marcin	Joanna	1/28/2022	2/28/2022	60	Premium	przelew	NIEAKTYWNA
Pawel	Piotr	3/28/2020	4/28/2020	40	Gold	przelew	NIEAKTYWNA
Waldemar	Piotr	4/1/2020	5/1/2020	60	Premium	karta	NIEAKTYWNA
Zuzanna	Joanna	12/14/2022	1/14/2023	60	Premium	blik	NIEAKTYWNA

Rysunek 78 Efekt procedury

- Klienci

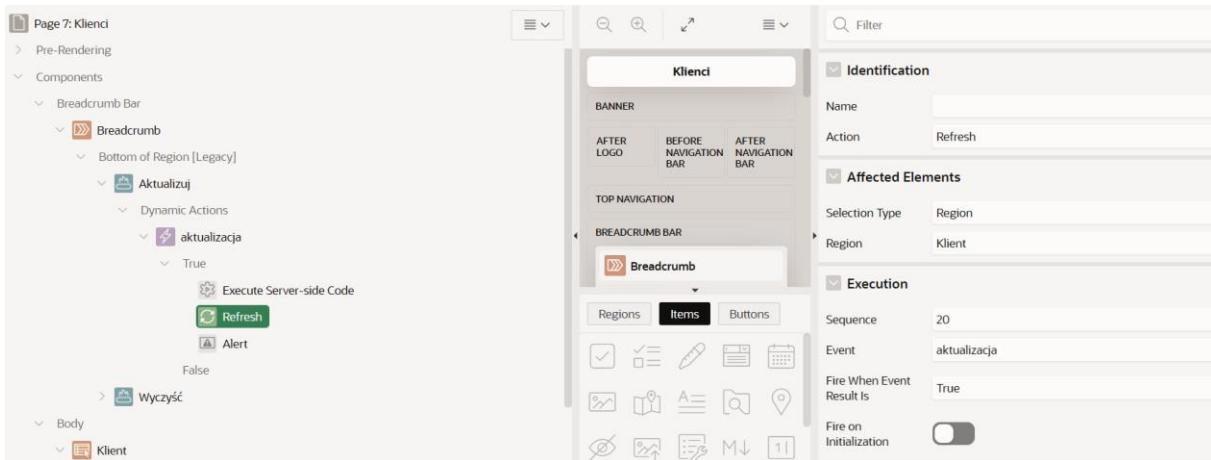
Klienci

Imię ↑↓	Nazwisko	Wiek	Nr Telefonu	Email	Ulica	Adres Kod Pocztowy	Pakiet
Aleksandra	Mila	17	953,678,731	amila@onet	Wesola	Rzeszow	
Aleksandra	Mila	17	953,678,731	amila@onet	Wesola	Rzeszow	
Anna	Wesolowska	39	678,876,876	awesolowska@onet	Sadowa	Rzeszow	
Anna	Wesolowska	39	678,876,876	awesolowska@onet	Sadowa	Rzeszow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jan	Kowlaski	19	517,928,516	jkowalski@onet	Urocza	Rzeszow	Podstawowy
Jan	Kowlaski	19	517,928,516	jkowalski@onet	Urocza	Rzeszow	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdansk	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdansk	
Konrad	Testowy	12,312,312,312	2,131	mail@asd.pl	Karowa	Rzeszow	

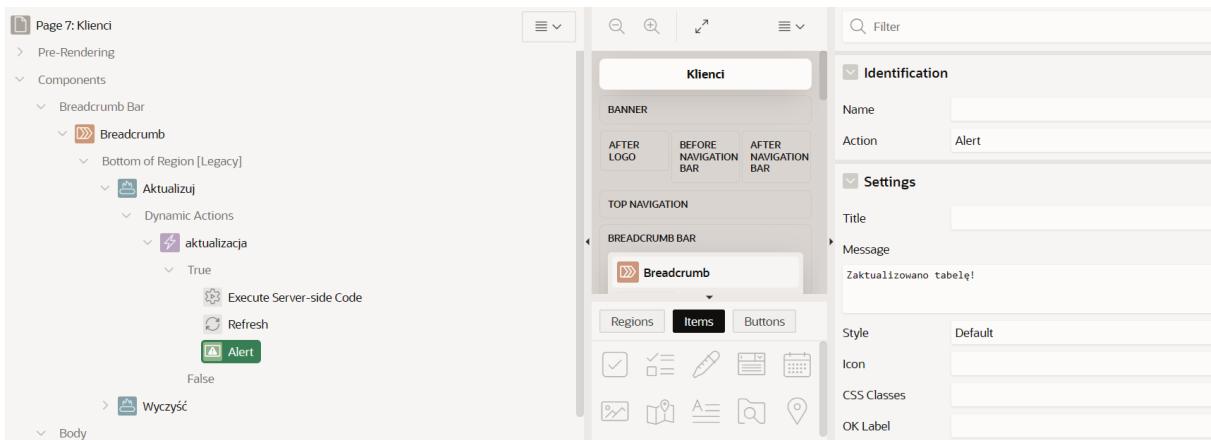
Rysunek 79 Klienci

Przechodząc dalej do strony klienci ponownie widnieje oprócz tabeli z danymi przycisk wykonujący procedurę przypisującą odpowiedni pakiet w zależności od aktywnej umowy – Aktualizuj pakiet. Dodatkowo ponownie został utworzony przycisk ułatwiający pokazanie efektów wykonania procedury. Tak jak w poprzednim przykładzie do guzika została przypisana dynamiczna akcja.

Rysunek 80 Wywołanie procedury w dynamicznej akcji



Rysunek 81 Odświeżanie tabeli po wykonaniu operacji



Rysunek 82 Powiadomienie o poprawności wykonania operacji

Wynik:

Klienci

[Aktualizuj pakiet](#) [Wyczść](#)

Imie	Nazwisko	Wiek	Nr Telefonu	Email	Ulica	Adres Kod Pocztowy	Pakiet
Aleksandra	Mila	17	953,678,731	amila@onet	Wesola	Rzeszow	
Aleksandra	Mila	17	953,678,731	amila@onet	Wesola	Rzeszow	
Anna	Wesolowska	39	678,876,876	awesolowska@onet	Sadowa	Rzeszow	
Anna	Wesolowska	39	678,876,876	awesolowska@onet	Sadowa	Rzeszow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jan	Kowlaski	19	517,928,516	jkowalski@onet	Urocza	Rzeszow	
Jan	Kowlaski	19	517,928,516	jkowalski@onet	Urocza	Rzeszow	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdansk	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdansk	
Konrad	Testowy	12,312,312	2,131	mail@asd.pl	Karowa	Rzeszow	

Rysunek 83 Tabela przed zmianą

Klienci

[Aktualizuj pakiety](#) [Wyczysć](#)

Imie ↑	Nazwisko	Wiek	Nr Telefonu	Email	Ulica	Adres Kod Pocztowy	Pakiet
Aleksandra	Mila				Wesola	Rzeszow	
Aleksandra	Mila				Wesola	Rzeszow	
Anna	Wesolowska				Sadowa	Rzeszow	
Anna	Wesolowska				Sadowa	Rzeszow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jan	Kowlaski	19	517,928,516	jkowalski@onet	Uroczna	Rzeszow	Podstawowy
Jan	Kowlaski	19	517,928,516	jkowalski@onet	Uroczna	Rzeszow	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdansk	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdansk	
Konrad	Testowy	12,312,312,312	2,131	mail@asd.pl	Karowa	Rzeszow	
Konrad		15	1,231,244	mail@mail.pl	Politechniczna	Gdansk	

Zaktualizowano tabelę!

[OK](#)

[Reset](#)

[Home](#) [App 180168](#) [Page 7](#) [Session](#) [Debug](#) [Quick Edit](#) [Customize](#) [Info](#) [Settings](#)

Rysunek 84 Powiadomienie o wykonaniu operacji

Klienci

[Aktualizuj pakiety](#) [Wyczysć](#)

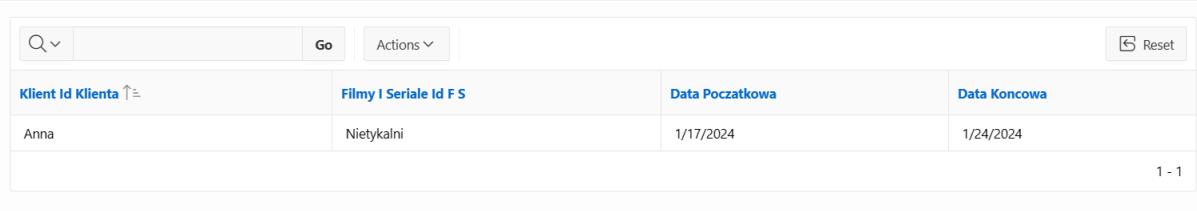
Imie ↑	Nazwisko	Wiek	Nr Telefonu	Email	Ulica	Adres Kod Pocztowy	Pakiet
Aleksandra	Mila	17	953,678,731	amila@onet	Wesola	Rzeszow	
Aleksandra	Mila	17	953,678,731	amila@onet	Wesola	Rzeszow	
Anna	Wesolowska	39	678,876,876	awesolowska@onet	Sadowa	Rzeszow	
Anna	Wesolowska	39	678,876,876	awesolowska@onet	Sadowa	Rzeszow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jadwiga	Nowak	19	587,626,881	jnowak@onet	Wielkopolska	Krakow	
Jan	Kowlaski	19	517,928,516	jkowalski@onet	Uroczna	Rzeszow	Podstawowy
Jan	Kowlaski	19	517,928,516	jkowalski@onet	Uroczna	Rzeszow	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdansk	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdansk	
Konrad	Testowy	12,312,312,312	2,131	mail@asd.pl	Karowa	Rzeszow	

Rysunek 85 Efekt operacji

- Rezerwacje

Jest to strona, która ukazuje rezerwacje utworzone dla danych użytkowników na odpowiedni film. Zawiera ona jedynie tabelę, która podczas wykonywania procedury wypożyczenia czy oddania filmu na bieżąco jest aktualizowana.

Rezerwacje

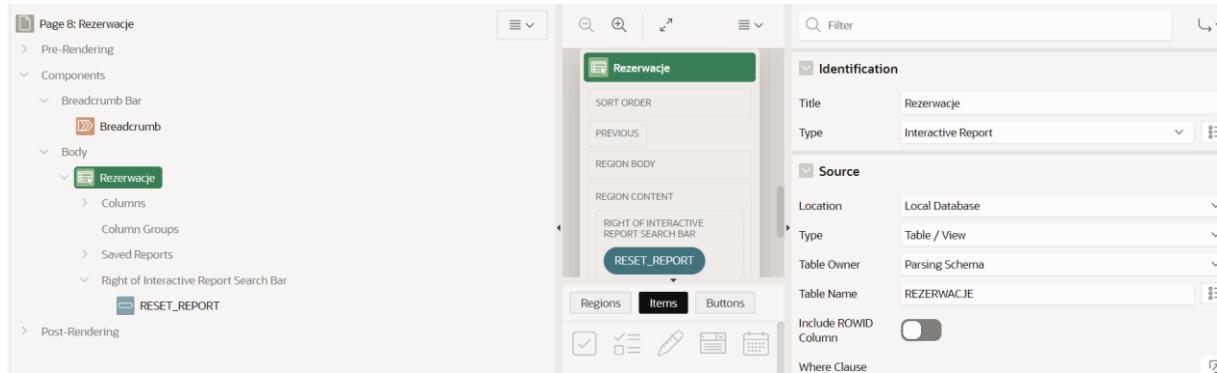


The screenshot shows a report titled 'Rezerwacje'. At the top, there are search and filter fields, followed by a 'Reset' button. Below is a table with four columns: 'Klient Id Klienta', 'Filmy I Seriale Id F S', 'Data Początkowa', and 'Data Koncowa'. The table contains one row with the values: Anna, Nietypalni, 1/17/2024, and 1/24/2024. A page number '1 - 1' is at the bottom right.

Klient Id Klienta ↑	Filmy I Seriale Id F S	Data Początkowa	Data Koncowa
Anna	Nietypalni	1/17/2024	1/24/2024

1 - 1

Rysunek 86 Rezerwacje



The screenshot shows the Oracle Application Express Page Designer interface for 'Page 8: Rezerwacje'. On the left, the tree view shows sections like 'Pre-Rendering', 'Components' (including 'Breadcrumb Bar' and 'Body' which contains 'Rezerwacje'), and 'Post-Rendering'. The main area shows the report structure with a 'Regions' tab selected. On the right, the 'Identification' panel shows the title 'Rezerwacje' and type 'Interactive Report'. The 'Source' panel shows the location as 'Local Database', type as 'Table / View', and table name as 'REZERWACJE'. A 'RESET_REPORT' button is visible in the report structure.

Rysunek 87 Page designer – Rezerwacje

- Dodaj klienta

Strona ta daje możliwość dodania nowego klienta. Tak jak w przypadku strony służącej do tworzenia wypożyczeń została zaopatrzona w obsługę błędów by przeciwdziałać błędному wpisywaniu wartości. Dla pola Kod Pocztowy została utworzona lista rozwijana, która zawiera odpowiednie wartości, które znajdują się odgórnie w bazie ze względu na jej budowę. Poniżej zostaje przedstawiony schemat strony, obsługa wyjątków – błędów oraz test działania.

Dodaj klienta

Dodaj klienta

Imię
Nazwisko
Wiek
Telefon
Email
Ulica
Kod Pocztowy 33-230

App 180168 Page 16 Session Debug Quick Edit Customize

Rysunek 88 Dodaj klienta

Application 180168 \ Page Designer

Rendering Dynamic Actions Processing Page Shared Components

Pre-Rendering Components Body Dodaj klienta Region Body P16_IEU P16_ZWISKOU P16_EKU P16__TELEFONUU P16_AILU P16_JCAU P16_RES_KOD_POCZTOWYU Close CANCEL Change SUBMIT Post-Rendering

Layout Page Search Help

BANNER AFTER LOGO BEFORE NAVIGATION BAR AFTER NAVIGATION BAR TOP NAVIGATION BREADCRUMB BAR Breadcrumb Regions Items Buttons

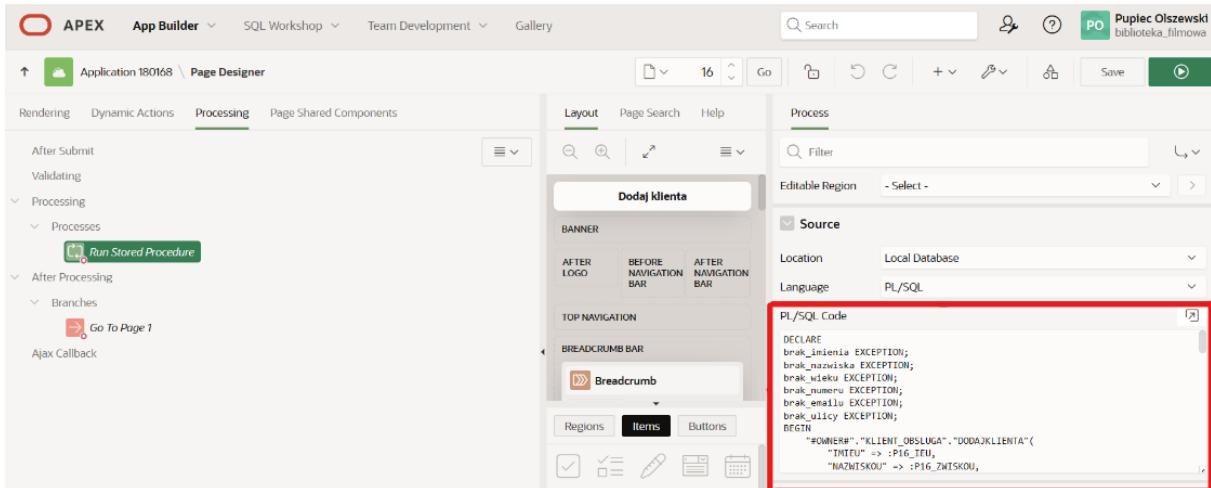
Page Identification Name Dodaj klienta Alias dodaj-klienta Title Dodaj klienta Page Group - Select -

Appearance Page Mode Normal Page Template Theme Default Template Options Use Template Defaults CSS Classes Media Type

Navigation Menu Override User Interface Level

Rysunek 89 Page designer – Dodaj klienta

Do wpisywania odpowiednich parametrów strona posiada 8 specjalnych pól. Tak jak dla wypożyczenia filmu przechodząc do zakładki Processing przedstawiony zostanie sposób wykonania obsługi błędów – utworzenia działających wyjątków.



Rysunek 90 Zakładka Processing - miejsce kodu

The screenshot shows the Oracle APEX Code Editor with the title 'Code Editor - PL/SQL Code'. The code editor window contains the same PL/SQL code as the previous screenshot, with line numbers 1 through 35. The code handles various input fields (IMEU, NAZWISKOU, WIEKU, NR_TELEFONUU, EMAILU, ULICAU, ADRES_KOD_POCZTOWYU) and checks for their nullity or specific character requirements. It uses RAISE statements to trigger exceptions like 'brak_imienia', 'brak_nazwiska', 'brak_wieku', etc., if the conditions are not met. The code ends with an exception block for 'brak_imienia' that raises an application error with code -20001 and a descriptive message.

Rysunek 91 Część kodu z obsługą błędów – wyjątkami

Efekt obsługi błędów:

The screenshot shows a web-based application interface for managing clients. The left sidebar has a dark theme with white icons and text, listing various menu items: Home, Klient - interfejs, Administrator - interfejs (with dropdown), Wypożyczenia, Umowy, Klienci, Rezerwacje, Oddaj film, Dodaj klienta (selected and highlighted in blue), and Raport Ogólny. The main content area has a light background. A red header bar at the top displays the title "Pupiec_Olszewski" and the page title "Dodaj klienta". Below the title is a form with several input fields: "Imię" (First Name), "Nazwisko" (Last Name), "Wiek" (Age), "Telefon" (Phone), "Email" (Email), and "Ulica" (Street). The "Imię" field is empty. A red error message box at the top right says "1 error has occurred" with a warning icon. It lists one error: "ORA-20001: Nie wprowadzono imienia lub wprowadzono niedozwolone znaki. Uzupełnij dane." (ORA-20001: First name was not entered or contains invalid characters. Please enter the data.). At the bottom of the form are "Cancel" and "Submit" buttons.

Rysunek 92 Brak wpisanej wartości dla pola Imię

This screenshot shows the same application interface as the previous one, but with different data entered in the form. The "Imię" field now contains "Konrad", "Nazwisko" contains "Janusz", "Wiek" contains "54", "Telefon" contains "123123123", and "Ulica" is empty. The "Email" field contains "email.com", which is clearly a typo. The red error message box at the top right reappears, stating "ORA-20005: Nie wprowadzono adresu e-mail lub jest nieprawidłowy. Uzupełnij dane." (ORA-20005: Email address was not entered or is invalid. Please enter the data.). The rest of the interface is identical to the first screenshot.

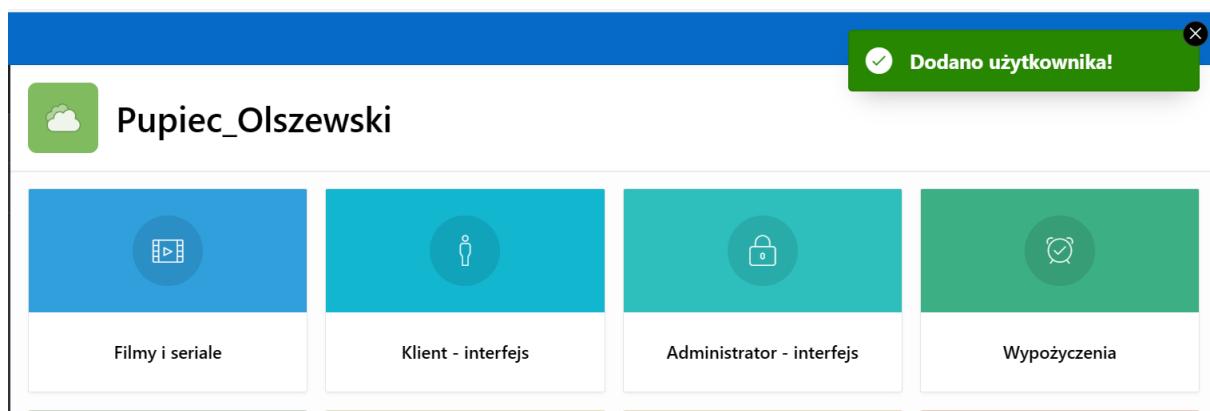
Rysunek 93 Błędnie podany email

Wykonanie:

Dodaj klienta

Imie Konrad	
Nazwisko Janusz	
Wiek 54	
Telefon 123123123	
Email email@kk.com	
Ulica Jaśminowa	
Kod Pocztowy 33-230	
<input type="button" value="Cancel"/>	<input type="button" value="Submit"/>

Rysunek 94 Wpisywane dane



Rysunek 95 Powiadomienie o udanej operacji

Pupiec_Olszewski

Klienci								
Aktualizuj pakiet Wyczść								
Imię ↑	Nazwisko	Wiek	Nr Telefonu	Email	Ulica	Adres Kod Pocztowy	Pakiet	
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdańsk		
Klaudia	Hula	25	719,465,578	khula@onet	Podkarpacka	Gdańsk		
Konrad	Janusz	54	123,123,123	email@kk.com	Jaśminowa	Rzeszów		
Konrad	Testowy	23	2,131	mail@asd.pl	Karowa	Rzeszów		
Konrad	Testowy	12,312,312,312	2,131	mail@asd.pl	Karowa	Rzeszów		
Konrad	O	15	1,231,244	mail@mail.pl	Politechniczna	Gdańsk		
Marcin	Ziółko	31	467,987,656	mziłko@onet	Weteranow	Katowice		
Marcin	Ziółko	31	467,987,656	mziłko@onet	Weteranow	Katowice		

Rysunek 96 Efekt operacji

- Raport ogólny

Strona ta posiada tabelę utworzoną przy pomocy kodu html zawartego w PL/SQL, która powstała przy użyciu metody htp.p. Do wypełnienia tabeli danymi wykorzystane zostały schemat procedury STATYSTYKA.raport_ogolny i funkcje w niej zawarte.

The screenshot shows a web application interface titled 'Raport Ogólny'. On the left, there is a sidebar with various menu items: Home, Klient - interfejs, Administrator - interfejs (which is currently selected), Wypożyczania, Umowy, Klienci, Rezerwacje, Oddaj film, Dodaj klienta, and Raport Ogólny. The main content area displays a table titled 'RAPORT OGÓLNY' with the following data:

ZAROBKI Z WYPOŻYCZEŃ	340
ZAROBKI Z PAKIETÓW	380
CAŁKOWITE ZAROBKI	720

Rysunek 97 Raport ogólny

The screenshot shows the Oracle Application Express Page Designer interface for 'Page 18: Raport Ogólny'. The left sidebar shows the page structure with components like Pre-Rendering, Components (Breadcrumb Bar, Report ogólny), Body, and Post-Rendering. The main area shows the page layout with regions for PREVIOUS, NEXT, SORT ORDER, REGION BODY, and REGION CONTENT. The right panel shows the 'Source' section where the PL/SQL code for generating the report is displayed:

```

DECLARE
    zarobki_wypozyczenia NUMBER;
    zarobki_pakietu NUMBER;
BEGIN
    zarobki_wypozyczenia := STATYSTYKA.zarobki_z_wypozyczen();
    zarobki_pakietu := STATYSTYKA.zarobki_z_pakietow();
END;
/

```

The code also includes HTML and CSS for styling the report table.

Rysunek 98 Page designer - Raport ogólny

Raport ten automatycznie aktualizuje się odczytując na bieżąco dane z odpowiednich tabel. Poniżej przedstawiony zostaje kod PL/SQL, który pozwala na utworzenie tego obiektu.

```

Code Editor - PL/SQL Function Body returning a CLOB
X
Save and Run Page
Cancel OK

1 DECLARE
2     zarobki_wypozerzenia NUMBER;
3     zarobki_pakiety NUMBER;
4 BEGIN
5     zarobki_wypozerzenia := STATYSTYKA.zarobki_z_wypozeren();
6     zarobki_pakiety := STATYSTYKA.zarobki_z_pakietow();
7
8     http.p('<table border="1" style="width: 60%;>');
9     http.p('<tr><td colspan="2" style="text-align: center; font-weight: bold; font-size: 18px; background-color: #e6e6e6;">RAPORT OGÓLNY</td></tr>');
10    http.p('<tr><td style="width: 40%; text-align: center; font-weight: bold; background-color: #e6e6e6;">ZAROBKI Z WYPOŻYCZEŃ</td><td style="text-align: center;">' ||
11        || zarobki_wypozerzenia || '</td></tr>');
12    http.p('<tr><td style="text-align: center; font-weight: bold; background-color: #e6e6e6;">ZAROBKI Z PAKIETÓW</td><td style="text-align: center;">' ||
13        || zarobki_pakiety || '</td></tr>');
14    http.p('<tr><td style="text-align: center; font-weight: bold; background-color: #e6e6e6;">CATKOWITE ZAROBKI</td><td style="text-align: center;">' ||
15        || (zarobki_wypozerzenia + zarobki_pakiety) || '</td></tr>');
16    http.p('</table>');
17 END;
18

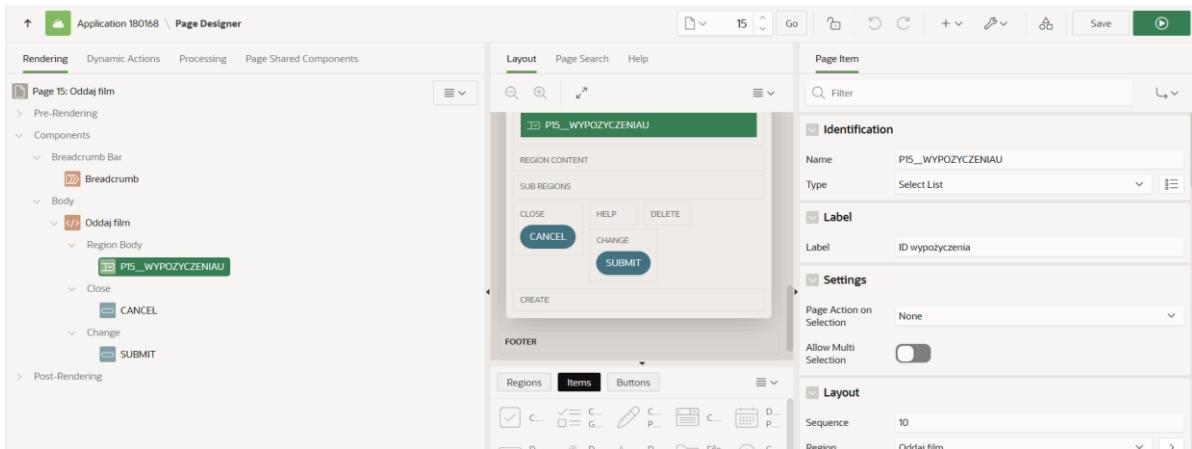
```

Rysunek 99 Kod tworzący tabelę - Raport ogólny

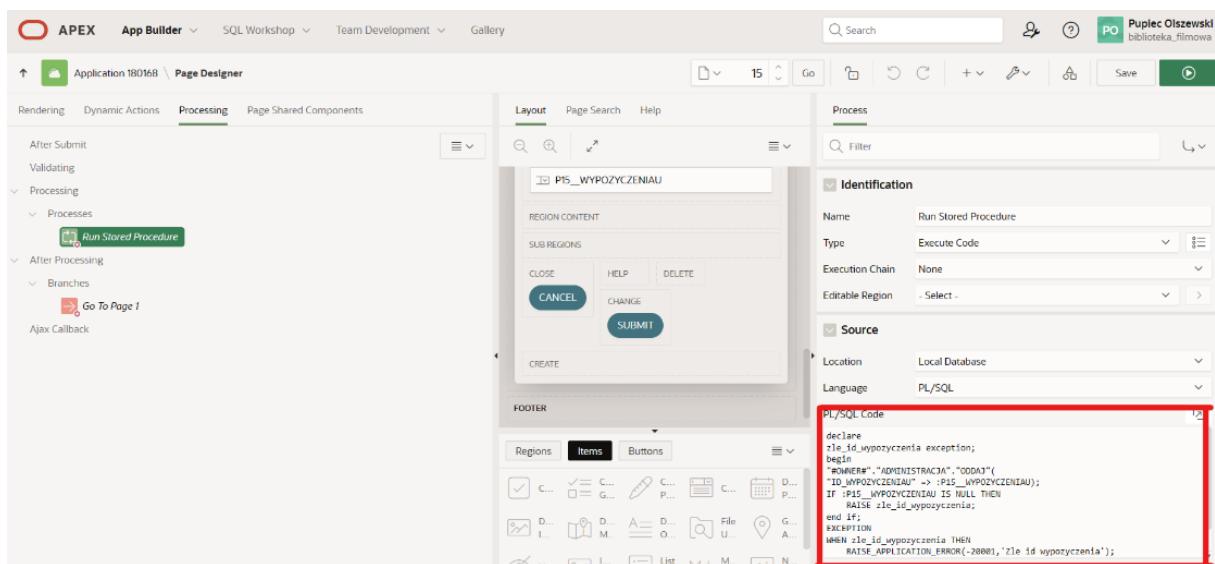
- Oddaj film

Strona ta daje możliwość oddania filmu, który aktualnie jest w trakcie wypożyczenia. Do wyboru z listy wybieranej widnieją tylko wartości id, którego status wypożyczenia nadal jest aktywny. Oprócz tego by uniknąć błędów ustwiono, że wartość ta nie może być pusta. Oprócz tego przy oddawaniu filmu dodawane jest nowe wypożyczenie w zależności czy istnieje rezerwacja na dany film oraz tworzona jest tablica, która przyjmuje funkcję historii rezerwacji.

Rysunek 100 Oddaj film



Rysunek 101 Page designer – Oddaj film



Rysunek 102 Kod znajdujący się w zakładce Processing

The screenshot shows a PL/SQL code editor interface. At the top, there are standard file operations like Save and Run Page and a settings gear icon. A green banner at the top indicates "Validation successful". Below the banner is the PL/SQL code:

```

1 declare
2 zle_id_wypozyzenia exception;
3 begin
4 "#OWNER#"."ADMINISTRACJA"."ODDAJ"(
5 "ID_WPOZYCZENIAU" => :P15__WPOZYCZENIAU);
6 IF :P15__WPOZYCZENIAU IS NULL THEN
7 | RAISE zle_id_wypozyzenia;
8 end if;
9 EXCEPTION
10 WHEN zle_id_wypozyzenia THEN
11 | RAISE_APPLICATION_ERROR(-20001,'zle id wypozyzenia');
12 |
13 RETURN;
14 end;

```

The code contains several syntax errors, such as missing colons after declarations and assignments, and a missing closing parenthesis for the IF block. The code editor highlights these errors in red.

Rysunek 103 Skrypt z obsługą błędów – wyjątki

Wynik:

Wypożyczenia

The screenshot shows a table of loan records. The columns are: Klient Id Klienta, Administrator Id Ad, Filmy I Seriale Id F S, Data Wypożyczenia, Data Oddania, Kwota Koncowa, Rodzaj Płatnosci Id P, and Status. The table contains 12 rows of data. The last row, which corresponds to the highlighted row in the previous screenshot, is highlighted with a red border.

Zmień Dątę	Q v	Go	Actions v	Reset			
Klient Id Klienta	Administrator Id Ad	Filmy I Seriale Id F S	Data Wypożyczenia	Data Oddania	Kwota Koncowa	Rodzaj Płatnosci Id P	Status
Aleksandra	Piotr	Nietypalni	1/9/2024	1/16/2024	20	przelew	W TRAKCIE
Aleksandra	Piotr	Nietypalni	2/19/2022	12/28/2022	20	karta	ODDANE
Anna	Joanna	Usmiechnij się	7/10/2022	7/15/2022	30	karta	ODDANE
Jan	Piotr	Skazanie na Shawshank	12/28/2022	12/30/2022	20	blik	ODDANE
Klaudia	Joanna	Avatar Istota wody	1/12/2022	1/28/2022	30	blik	ODDANE
Marcin	Joanna	Avatar Istota wody	1/28/2022	12/20/2022	30	przelew	ODDANE
Pawel	Piotr	Usmiechnij się	3/28/2020	12/30/2020	30	przelew	ODDANE
Piotr	Piotr	Skazanie na Shawshank	1/8/2024	1/15/2024	20	karta	W TRAKCIE
Piotr	Piotr	Nietypalni	1/8/2024	1/9/2022	20	karta	W TRAKCIE
Waldemar	Piotr	Skazanie na Shawshank	4/1/2020	4/5/2020	20	karta	ODDANE
Waldemar	Piotr	Skazanie na Shawshank	4/1/2020	4/28/2020	20	karta	ODDANE
Zuzanna	Joanna	Usmiechnij się	12/14/2022	12/15/2022	30	blik	ODDANE

1 - 12

Rysunek 104 Przed oddaniem. ID Wypożyczenia 11.

Rezerwacje

Actions		Reset	
Klient Id Klienta ↑	Filmy I Seriale Id F S	Data Poczatkowa	Data Koncowa
Anna	Nietykalni	1/17/2024	1/24/2024
1 - 1			

Rysunek 105 Rezerwacje przed oddaniem

Oddaj film

Oddaj film
ID wypożyczenia 11
<input type="button" value="Cancel"/> <input type="button" value="Submit"/>

Rysunek 106 Wprowadzona wartość

Pupiec_Olszewski

Operacja została wykonana! Jeżeli minął termin film został oddany!

- Filmy i seriale
- Klient - interfejs
- Administrator - interfejs
- Wypożyczenia

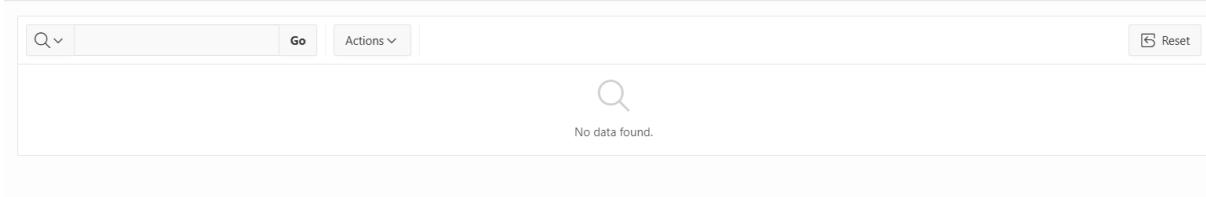
Rysunek 107 Komunikat po wykonaniu operacji

Wypożyczenia

Zmień Dátę							
Klient Id Klienta ↑	Administrator Id Ad	Filmy I Seriale Id F S	Data Wypożyczenia	Data Oddania	Kwota Koncowa	Rodzaj Płatnosci Id P	Status
Aleksandra	Piotr	Nietykalni	2/19/2022	12/28/2022	20	karta	ODDANE
Aleksandra	Piotr	Nietykalni	1/9/2024	1/16/2024	20	przelew	W TRAKCIE
Anna	Joanna	Usmiechnij się	7/10/2022	7/15/2022	30	karta	ODDANE
Anna	Piotr	Nietykalni	1/9/2024	1/16/2024	20	karta	W TRAKCIE
Jan	Piotr	Skazanie na Shawshank	12/28/2022	12/30/2022	20	blik	ODDANE
Klaudia	Joanna	Avatar Istota wody	1/12/2022	1/28/2022	30	blik	ODDANE
Marcin	Joanna	Avatar Istota wody	1/28/2022	12/20/2022	30	przelew	ODDANE
Pawel	Piotr	Usmiechnij się	3/28/2020	12/30/2020	30	przelew	ODDANE
Piotr	Piotr	Skazanie na Shawshank	1/8/2024	1/15/2024	20	karta	W TRAKCIE
Piotr	Piotr	Nietykalni	1/1/2023	1/9/2024	20	karta	ODDANE
Waldemar	Piotr	Skazanie na Shawshank	4/1/2020	4/28/2020	20	karta	ODDANE
Waldemar	Piotr	Skazanie na Shawshank	4/1/2020	4/5/2020	20	karta	ODDANE
Zuzanna	Joanna	Usmiechnij się	12/14/2022	12/15/2022	30	blik	ODDANE

Rysunek 108 Wynik - tabela wypożyczenia

Rezerwacje



Rysunek 109 Rezerwacje po wykonaniu operacji

Jak można zauważyc rezerwacja zmieniła swój stan i przeszła do tabeli wypożyczenia a także dostała status ‘W TRAKCIE’. Daty zostały przypisane automatycznie.

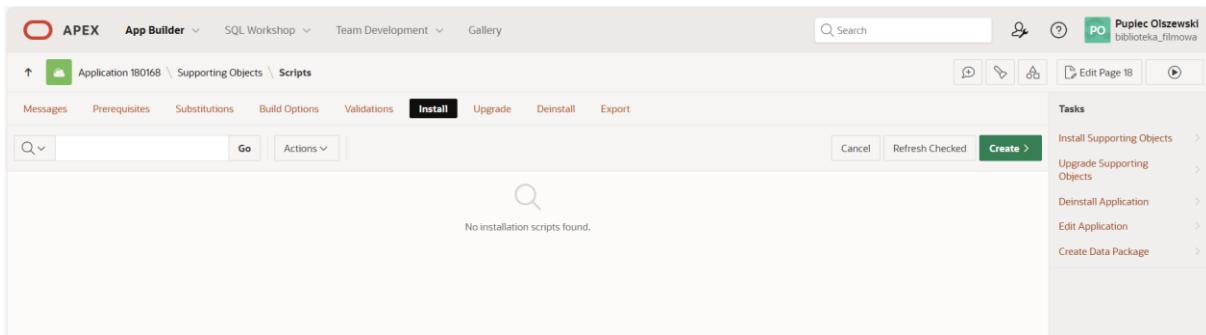
6.5. Export aplikacji bazodanowej

W tym punkcie utworzona aplikacja bazodanowa zostanie wyeksportowana a następnie ponownie zainportowana 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 zainportowane razem z aplikacją.

- Dodanie skryptów

W celu dodania skryptów przechodzimy do zakładki SUPPORTING OBJECTS a następnie w sekcji Installation do zakładki Installation Scripts, gdzie następnie dodajemy utworzone skrypty tworzące bazę danych oraz pakiety z procedurami i funkcjami.



Rysunek 110 Miejsce dodania skryptów SQL

Tworzenie skryptów:

Pierwszym krokiem było utworzenie skryptu posiadającego wszystkie tabele wraz z zainportowanymi do utworzenia projektu danymi. Wybierając odpowiednią opcję przenieśliśmy do skryptu całą oczekiwana zawartość.

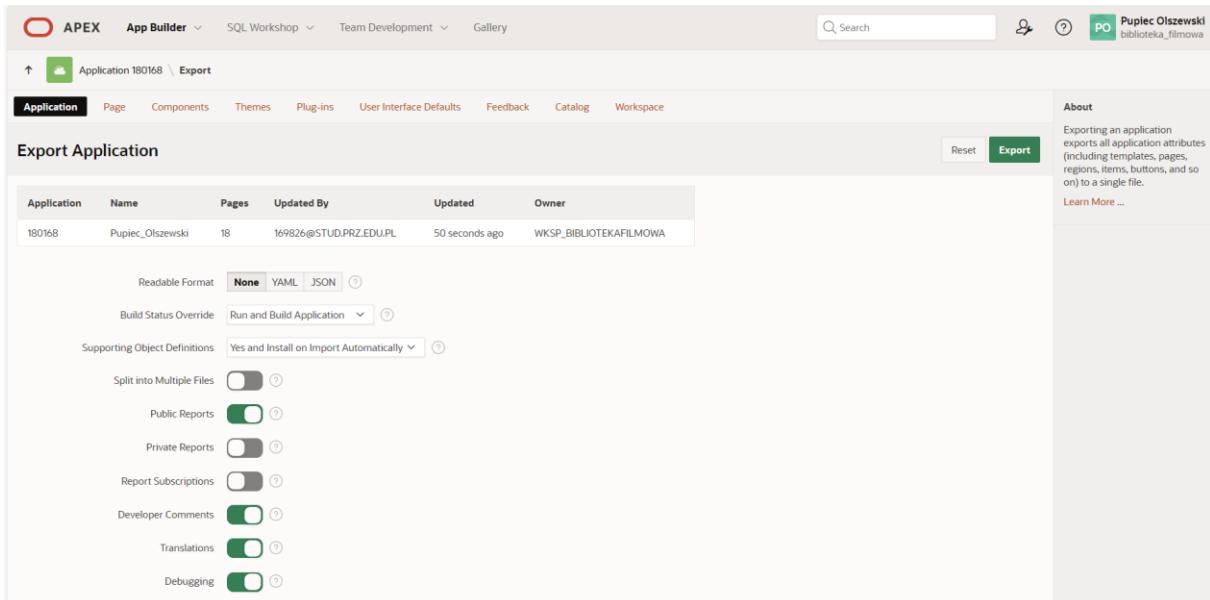


The screenshot shows the Oracle APEX App Builder interface. In the center, there is a table titled 'Script' with one row. The row contains the following information: Lock (calosc), Name (10), Script (CREATE TABLE a_f (aktorzy_id_aktora INTEGER NOT NULL, filmy_i_serialne_id_f_s INTEGER NOT NULL); ALTER TABLE a_f ADD CONSTRAINT a_f_pk PRIMARY KEY (aktorzy_id_aktora, ...)), Status (12 minutes ago), Updated (169826@stud.prz.edu.pl), Updated By (Manual), and Source (checkbox). At the top of the table, there are buttons for 'Install', 'Upgrade', 'Deinstall', and 'Export'. To the right of the table, there is a 'Tasks' sidebar with options like 'Install Supporting Objects', 'Upgrade Supporting Objects', and 'Deinstall Application'. The status bar at the bottom indicates '1 - 1 Script'.

Rysunek 111 Utworzony skrypt

- Export

Przy eksportie włączliśmy funkcje by dodane skrypty zostały automatycznie zainstalowane w nowej aplikacji.



The screenshot shows the Oracle APEX App Builder interface with the 'Export' tab selected. On the left, there is a table titled 'Export Application' with one row. The row contains columns: Application (180168), Name (Pupiec_Olszewski), Pages (18), Updated By (169826@STUD.PRZ.EDU.PL), Updated (50 seconds ago), and Owner (WKSP_BIBLIOTEKA_FILMOWA). On the right, there is a 'About' section with the following text: 'Exporting an application exports all application attributes (including templates, pages, regions, items, buttons, and so on) to a single file.' Below this, there is a 'Learn More ...' link. Underneath the table, there are several configuration options: 'Readable Format' (None, YAML, JSON), 'Build Status Override' (Run and Build Application), 'Supporting Object Definitions' (Yes and Install on Import Automatically), 'Split into Multiple Files' (disabled), 'Public Reports' (enabled), 'Private Reports' (disabled), 'Report Subscriptions' (disabled), 'Developer Comments' (disabled), 'Translations' (disabled), and 'Debugging' (disabled).

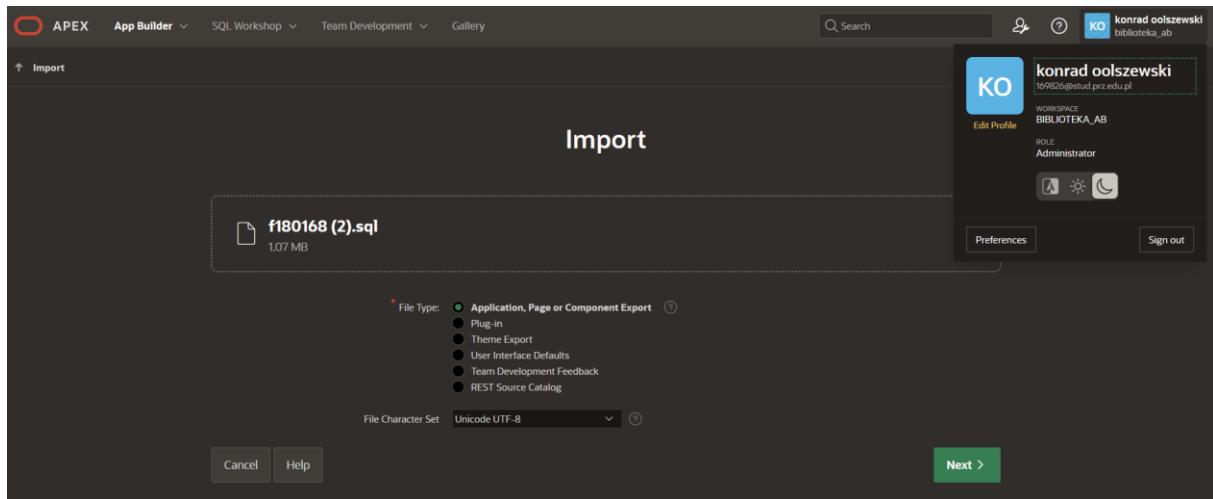
Rysunek 112 Export



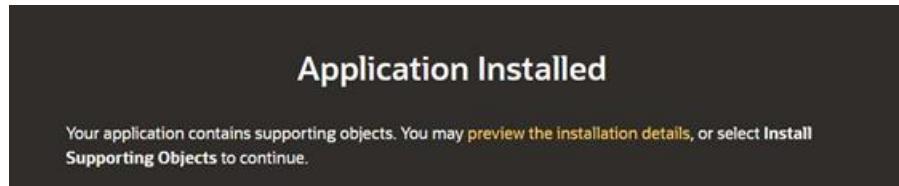
Rysunek 113 Utworzony plik z rozszerzeniem SQL

- Import

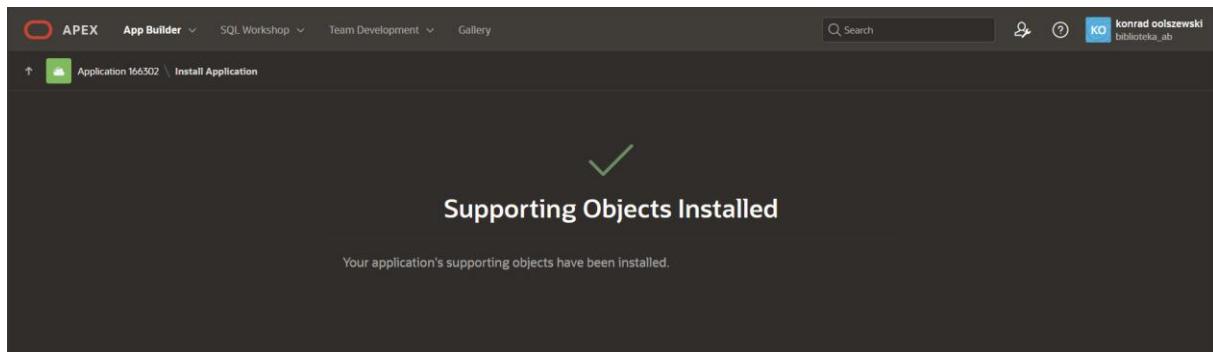
Operacja importu wykonana zostanie na nowo utworzonej przestrzeni roboczej:



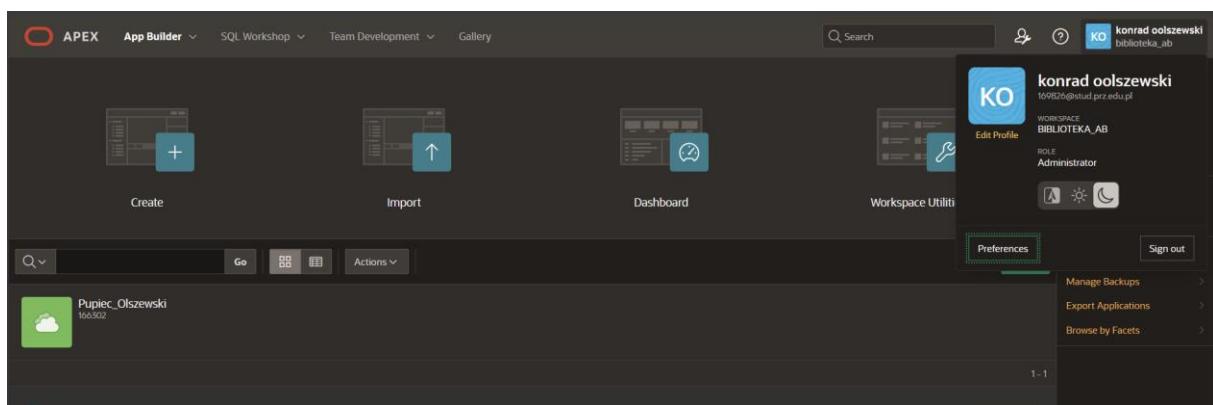
Rysunek 114 Import aplikacji bazodanowej na nowym Workspace



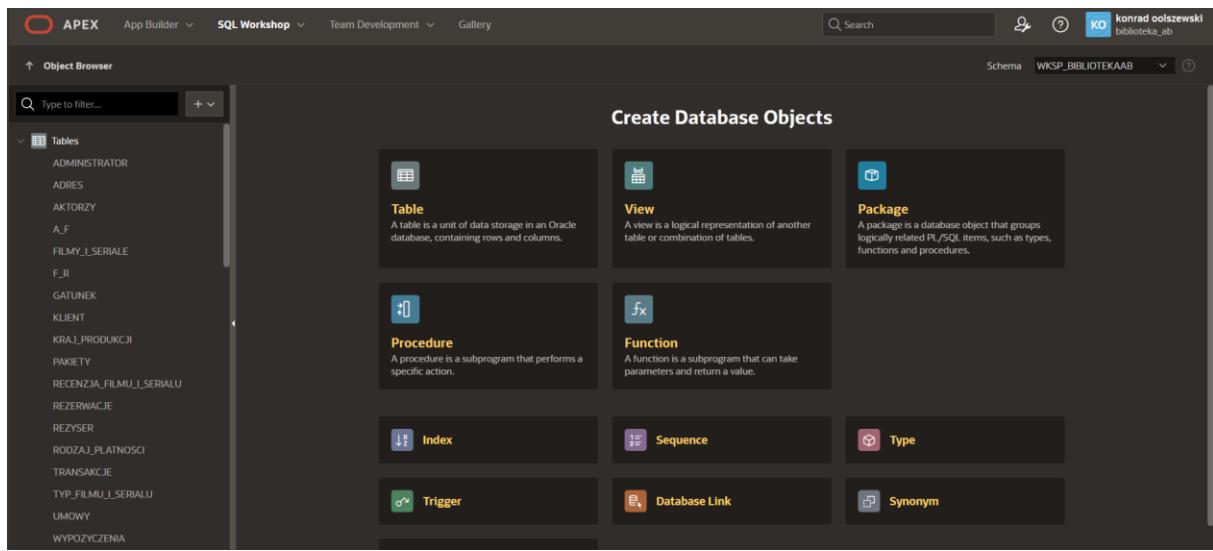
Rysunek 115 Zainstalowanie aplikacji



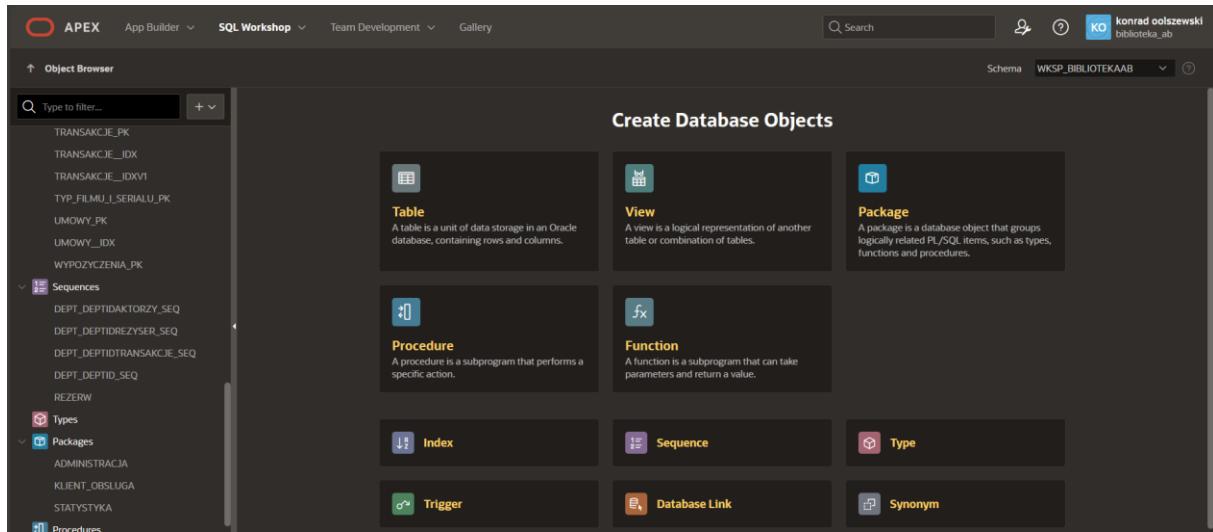
Rysunek 116 Poprawna instalacja Supporting Objects



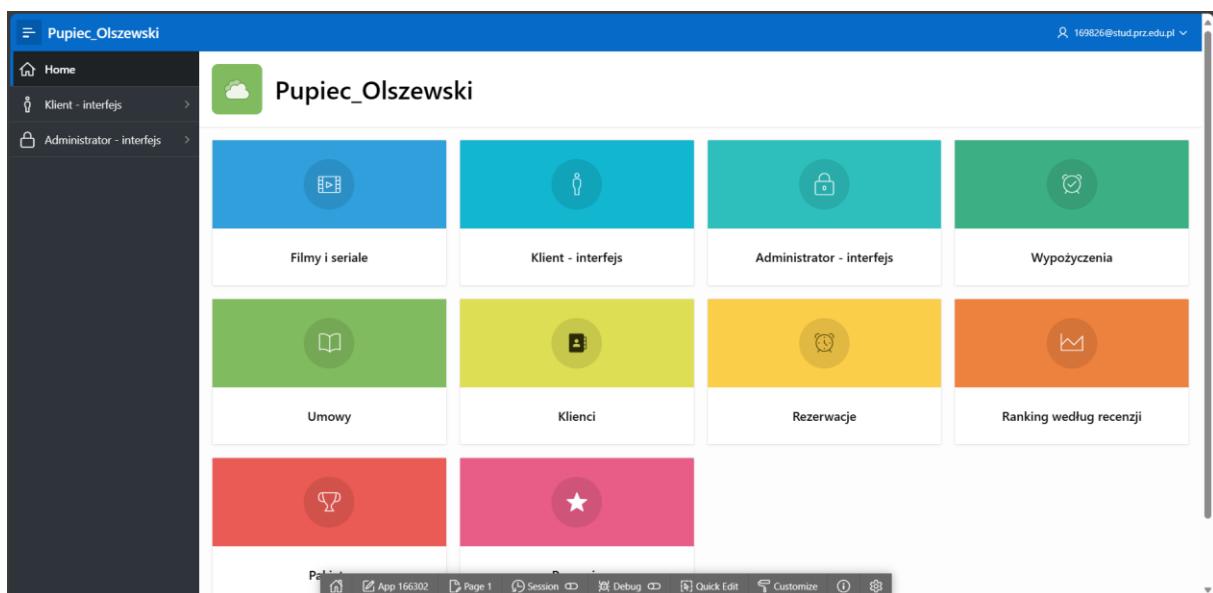
Rysunek 117 Zainstalowana aplikacja



Rysunek 118 Zainstalowane Supporting Objects - Tabele



Rysunek 119 Zainstalowane Supporting Objects - Indeksy, Sekwencje, Pakiety



Rysunek 120 Poprawne uruchomienie zimportowanej aplikacji

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

7. Wnioski i podsumowanie

Oracle APEX to efektywne narzędzie umożliwiające szybkie tworzenie aplikacji bazodanowych w środowisku Oracle. Dzięki prostocie w projektowaniu interfejsu użytkownika oraz łatwej integracji z bazą danych, APEX jest potężnym narzędziem do budowy aplikacji webowych. Kluczowymi elementami tego rozwiązania są PL/SQL, procedury, funkcje, pakiety i kolekcje, które pozwalają na rozwijanie i zarządzanie aplikacjami w sposób modułowy i efektywny. W projekcie z wykorzystaniem Oracle APEX zwracamy uwagę na korzyści płynące z prostoty implementacji, elastyczności oraz skuteczności w tworzeniu nowoczesnych aplikacji bazodanowych. Jednocześnie, dbamy o odpowiednie zarządzanie kodem źródłowym, co przyczynia się do łatwiejszego utrzymania i rozbudowy projektu w dłuższej perspektywie czasowej.