Politechnika Wrocławska

# BAZY DANYCH – PROJEKT

DOKUMENTACJA PROJEKTOWA

Jacek Kozicki 248868

Agnieszka Jurijków 248888

Karolina Kowalczyk 250992

# SPIS TREŚCI

# WSTĘP TEORETYCZNY

## Podstawy relacyjnych baz danych

Relacyjne bazy danych to zbiór tabel powiązanych ze sobą za pomocą atrybutów kluczowych. Istniejące powiązania występujące pomiędzy poszczególnymi wierszami (krotkami) nazywane są relacjami. Na bazę danych składają się encje. To pojęcie oznacza każdy przedmiot, zjawisko, stan, obiekt, który potrafimy odróżnić od innych obiektów. W przypadku relacyjnych baz danych encja jest zazwyczaj utożsamiana z tabelą i zawiera atrybuty obiektu, który tworzy. Atrybuty charakteryzują cechy abstrakcyjnych obiektów umieszczonych w poszczególnych wierszach relacji. Inaczej jest to opis kolumn relacji. Zakres wartości jakie mogą przyjmować atrybuty (kolumny) nazywa się ich typem, domeną lub dziedziną. Na atrybuty mogą też być nałożone więzy, które określają dodatkowe ograniczenia nad nimi.

Istotnym pojęciem w temacie relacyjnych baz danych są atrybuty kluczowe. Wyróżniamy klucze proste i złożone. Klucz prosty to klucz, którego zbiór identyfikujący jest jednoelementowy, natomiast dla klucza złożonego zbiór identyfikujący jest kilkuelementowy. Klucz jest używany do jednoznacznej identyfikacji danych w dowolnie zadanym wierszu. Oznacza to, że klucz musi mieć wartości unikalne. Każda tabela powinna mieć swój klucz podstawowy – taki zabieg pozwala na uporządkowanie relacji między tabelami, co jest nazywane integralnością referencyjną. Istnieje także pojęcie klucza obcego, który jest atrybutem z obcej tabeli, gdzie ten atrybut jest podstawowym kluczem.

Relacje między tabelami dzieli się na:

- jeden do jednego (1 : 1),
- jeden do wielu (1 : n),
- wiele do wielu (m : n).

# CZĘŚĆ PRAKTYCZNA PROJEKTU

## Przedstawienie problemu

Celem projektu było zaimplementowanie jak najbardziej realistycznej struktury bazy danych banku, tak aby w przyszłości mogła być ona wykorzystana w aplikacji. Należało tak zaprojektować encje, krotki i atrybuty, aby możliwie jak najwierniej odzwierciedlały dane przechowywane w bazie danych banku.

Przy realizacji jednymi z najistotniejszych encji były rachunek, który jest podstawową jednostką funkcjonującą w instytucji bankowej i właściciel, do którego należy ten rachunek (jeden właściciel może posiadać kilka rachunków). Operacje pomiędzy rachunkami przedstawione są w transakcjach.

Na początkowym etapie projektowania bazy trzeba było przeprowadzić normalizację, dzięki czemu zmniejszyliśmy liczbę relacji w bazie danych i zoptymalizowaliśmy jej strukturę, poprzez usunięcie redundancji danych.

W encjach trzeba było odpowiednio dobrać typy danych i ograniczenia nałożone na atrybuty. Baza danych, szczególnie bankowa potrzebuje odpowiednich zabezpieczeń, dlatego przy jej tworzeniu, należało zwrócić szczególną uwagę na ten element.

## Wymagania systemu

Zrealizowany projekt powinien spełniać poniższe założenia:

- Użytkownicy będą mogli sprawdzać swój stan konta i historię operacji
- Użytkownicy będą mogli wykonywać operacje
- Po każdej operacji podsumowanie stanu konta
- Zaimplementowany będzie osobny widok dla:
  - Użytkownik: podgląd stanu konta, dokonywanie operacji
  - Administratora: możliwość modyfikacji/podglądu kont klientów

Zaimplementowana baza danych powinna być poprawna to znaczy spełniająca poniższe wymagania:

- ścisły związek z faktami świata rzeczywistego, tzn. łatwy sposób ich tworzenia i rozumienia,
- kompletność informacji,
- podatność na zmiany, a więc ewolucyjność schematu,
- stabilność, czyli projektowany schemat powinien uwzględniać przewidywalne zmiany,

- ❖ możliwość tworzenia różnych obrazów danych, czyli różnych logicznych modeli baz danych.

# Model danych ERD

## IDENTYFIKACJA ZBIORU ENCJI WRAZ Z ICH ATRYBUTAMI KLUCZOWYMI

W zrealizowanym systemie wyróżniono encje i atrybuty identyfikujące je w jednoznaczny sposób.

| Encja | Atrybut |
|---|---|
| Adres | Id_adresu = Id_adresu_oddzialu |
| Właściciel | Id_wlasciciela |
| Rachunek | Id_rachunku = Id_rachunku_nadawcy = Id_rachunku_odbiorcy |
| Oddziały | Id_oddzialu |
| Transakcje | Id_transakcji |
| Czas | Id_czasu |
| Rodzaj_operacji | Id_rodzaju_operacji |

**Tabela 1. Encje i identyfikatory**

## IDENTYFIKACJA BEZPOŚREDNICH ZALEŻNOŚCI MIĘDZY ENCJAMI

| | Adres | Właściciel | Rachunek | Oddziały | Transakcje | Czas | Rodzaj_operacji |
|---|---|---|---|---|---|---|---|
| **Adres** | | X | | X | | | |
| **Właściciel** | X | | X | | | | |
| **Rachunek** | | X | | | X | | |
| **Oddziały** | X | | | | X | | |
| **Transakcje** | | | X | X | | X | X |
| **Czas** | | | | | X | | |
| **Rodzaj_operacji** | | | | | X | | |

**Tabela 2. Tabela krzyżowa, zależności bezpośrednie pomiędzy encjami**

| Atrybut | Opis |
|---|---|
| Id_adresu = Id_adresu_oddzialu | Identyfikator adresu |
| Kod_pocztowy | Kod pocztowy |
| Ulica | Nazwa ulicy |
| Nr_lokalu | Numer lokalu/domu |
| Nr_mieszkania | Numer mieszkania |
| Miejscowosc | Nazwa miejscowości |
| Id_wlasciciela | Identyfikator właściciela rachunku |
| Imie_wlasciciela | Imię właściciela rachunku |
| Nazwisko_wlasciciela | Nazwisko właściciela rachunku |
| Plec_wlasciciela | Płeć właściciela rachunku |
| PESEL_wlasciciela | PESEL właściciela rachunku |
| Id_rachunku = Id_rachunku_nadawcy = Id_rachunku_odbiorcy | Identyfikator rachunku |
| Data_zalozenia | Data założenia rachunku |
| Saldo | Saldo rachunku |
| Nr_rachunku | Numer rachunku |
| Id_oddzialu | Identyfikator oddziału banku |
| Nazwa_oddzialu | Nazwa oddziału banku |
| Id_transakcji | Identyfikator transakcji bankowej |
| Id_rodzaju_operacji | Identyfikator rodzaju operacji bankowej |
| Id_czasu | Identyfikator czasu |
| Kwota | Kwota transakcji bankowej |
| Rok | Rok wykonania transakcji |
| Miesiac | Miesiąc wykonania transakcji |
| Dzien | Dzień wykonania transakcji |
| Godzina | Godzina wykonania transakcji |
| Minuty | Minuty wykonania transakcji |
| Rodzaj_operacji | Rodzaj operacji bankowej |
| Oplata | Opłata za operację bankową |

**Tabela 3. Opis atrybutów**

| Encja | Atrybut |
|---|---|
| Adres | Id_adresu, Kod_pocztowy, Ulica, Nr_lokalu, Nr_mieszkania, Miejscowosc |
| Właściciel | Id_wlasciciela, Id_adresu, Imie_wlasciciela, Nazwisko_wlasciciela, Plec_wlasciciela, PESEL_wlasciciela |
| Rachunek | Id_rachunku, Id_wlasciciela, Data_zalozenia, Saldo, Nr_rachunku |

| | |
|---|---|
| Oddziały | Id_oddzialu, Nazwa_oddzialu, Id_adresu_oddziału |
| Transakcje | Id_transakcji, Id_rachunku_nadawcy, Id_oddzialu, Id_rodzaju_operacji, Id_czasu, Id_rachunku_odbiorcy, Kwota |
| Czas | Id_czasu, Rok, Miesiac, Dzien, Godzina, Minuty |
| Rodzaj_operacji | Id_ rodzaju_operacji, Rodzaj_operacji, Oplata |

**Tabela 4. Wykaz encji i powiązanych z nimi atrybutów**

| **Związek** | **Opis** |
|---|---|
| Charakteryzuje | Właściciel – Adres 1:1 |
| Wskazuje na | Rachunek – Właściciel N:1 |
| Charakteryzuje | Oddziały – Adres 1:1 |
| Zaopatruje | Transakcje – Rachunek N:1 |
| Dotyczy | Transakcje – Oddziały N:1 |
| Charakteryzuje | Transakcje – Czas 1:1 |
| Dotyczy | Transakcje – Rodzaj_operacji N:1 |

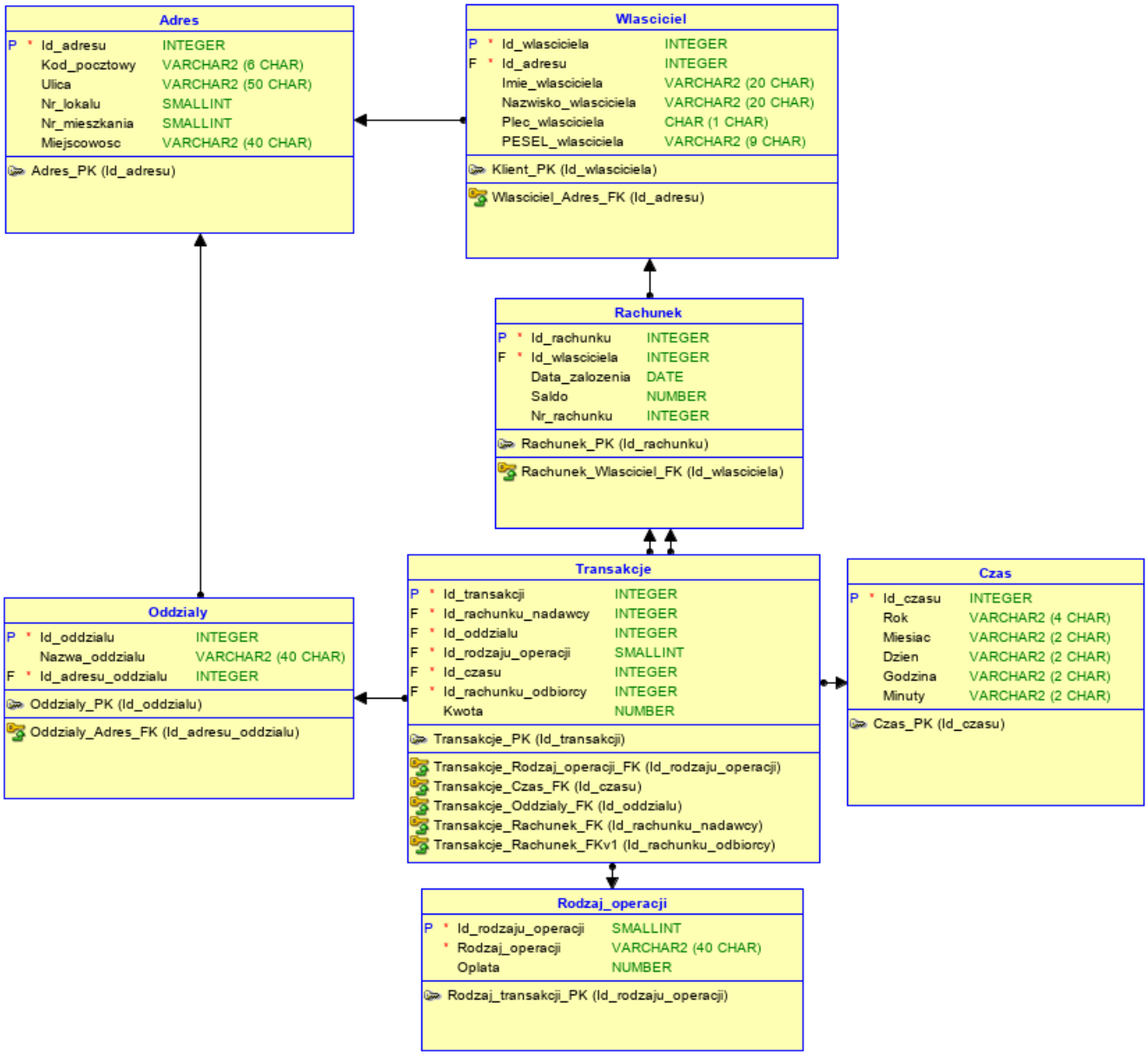
**Tabela 5. Opis związków pomiędzy encjami**

# Schemat diagramu ERD

## OPIS APLIKACJI W KTÓREJ MODELOWANO SCHEMAT

Schemat modelowano w SQL Data Modeler. Jest to darmowe narzędzie pozwalające na tworzenie, przeglądanie i edycję tabel i relacji między nimi. Cechuje się bardzo przejrzystym interfejsem graficznym.

PREZENTACJA SCHEMATU ERD BAZY DANYCH




**Rysunek 1. Schemat ERD**

# Rozwiązanie problemu

## SYSTEM BAZODANOWY

### *Utworzenie bazy danych*

Do stworzenia bazy danych został wykorzystany Oracle SQL Developer. Pracowaliśmy na serwerze Politechniki Wrocławskiej, udostępnionym nam w ramach zajęć laboratoryjnych. Obok tabel już stworzonych, dodane zostały potrzebne tabele o określonych na diagramie ERD atrybutach.

*Implementacja obiektów*

Przy implementacji zapytań są używane formularze dostępne w Oracle SQL Developerze. Poniżej został umieszony listing skryptów tworzących tabele.

```
CREATE TABLE Adres (
Id_adresu INT,
Kod_pocztowy VARCHAR2 (6),
Ulica VARCHAR2 (50),
Nr_lokalu SMALLINT,
Nr_mieszkania SMALLINT,
Miejscowosc VARCHAR2 (40),

PRIMARY KEY (Id_adresu)
);


CREATE TABLE Wlasciciel (
Id_wlasciciela INT,
Id_adresu INT,
Imie_wlasciciela VARCHAR2(20),
Nazwisko_wlasciciela VARCHAR2(20),
Plec_wlasciciela CHAR,
PESEL_wlasciciela VARCHAR2(9) NOT NULL UNIQUE,

PRIMARY KEY (Id_wlasciciela),
CONSTRAINT Id_adresu FOREIGN KEY (Id_adresu)
REFERENCES Adres(Id_adresu)
);


CREATE TABLE Rachunek (
Id_rachunku INT,
Id_wlasciciela INT,
Data_zalozenia DATE,
Saldo NUMBER DEFAULT 0,
Nr_rachunku VARCHAR2 (40) UNIQUE,

PRIMARY KEY (Id_rachunku),
CONSTRAINT Id_wlasciciela FOREIGN KEY (Id_wlasciciela)
REFERENCES Wlasciciel(Id_wlasciciela
);


CREATE TABLE Oddzialy (
Id_oddzialu INT,
Nazwa_oddzialu VARCHAR2(40),
Id_adresu_oddzialu INT,

PRIMARY KEY (Id_oddzialu),
CONSTRAINT Id_adresu_oddzialu FOREIGN KEY (Id_adresu_oddzialu)
REFERENCES Adres(Id_adresu)
);


CREATE TABLE Czas (
Id_czasu INT,
Rok VARCHAR2(4),
Miesiac VARCHAR2(2),
Dzien   VARCHAR2(2),
```

```
Godzina VARCHAR2(2),
Minuty  VARCHAR2(2),
PRIMARY KEY (Id_czasu)
);

CREATE TABLE Rodzaj_operacji(
Id_rodzaju_operacji SMALLINT,
Rodzaj_operacji VARCHAR2(40) DEFAULT 'Zwykla',
Oplata NUMBER DEFAULT 0,

PRIMARY KEY (Id_rodzaju_operacji)
);


CREATE TABLE Transakcje (
Id_transakcji INT,
Id_rachunku_nadawcy INT,
Id_oddzialu INT,
Id_rodzaju_operacji SMALLINT,
Id_czasu INT,
Id_rachunku_odbiorcy INT,
Kwota NUMBER NOT NULL,

PRIMARY KEY (Id_transakcji),

CONSTRAINT Id_rachunku_nadawcy FOREIGN KEY (Id_rachunku_nadawcy)
REFERENCES Rachunek(Id_rachunku),

CONSTRAINT Id_oddzialu FOREIGN KEY (Id_oddzialu)
REFERENCES Oddzialy(Id_oddzialu),

CONSTRAINT Id_rodzaju_operacji FOREIGN KEY (Id_rodzaju_operacji)
REFERENCES Rodzaj_operacji(Id_rodzaju_operacji),

CONSTRAINT Id_czasu FOREIGN KEY (Id_czasu)
REFERENCES Czas(Id_czasu),

CONSTRAINT Id_rachunku_odbiorcy FOREIGN KEY (Id_rachunku_odbiorcy)
REFERENCES Rachunek(Id_rachunku)
);
```

### *Wprowadzenie danych*

Dane wygenerowane zostały z wykorzystaniem strony: https://www.mockaroo.com/. Nie wszystkie wygenerowane dane są zgodne z polską konwencją (np. kod pocztowy, numer konta bankowego, PESEL). Poniżej umieszczone zostały listingi obrazujące przykładowe wprowadzenie danych do każdej z tabel.

```
insert into Adres (Id_adresu, Kod_pocztowy, Ulica, Nr_Lokalu,
Nr_Mieszkania, Miejscowosc) values (1, '80600', 'Summit', 104, 20, 'Johor
Bahru');

insert into Wlasciciel (Id_wlasciciela, Id_adresu, Imie_wlasciciela,
Nazwisko_wlasciciela, Plec_wlasciciela, PESEL_wlasciciela) values (1, 92,
'Ingmar', 'Bing', 'M', 919716311);

insert into Oddzialy (Id_oddzialu, Nazwa_oddzialu, Id_adresu_oddzialu)
values (1, 'neque aenean', 191);
```

```
insert into Rachunek (Id_rachunku, Id_wlasciciela, Data_zalozenia, Saldo,
Nr_rachunku) values (1, 1, DATE '2014-09-15', 43213.36, 'MD68 UKX6 J6EE
7DID YVKN 4URN');

insert into Czas (Id_czasu, Rok, Miesiac, Dzien, Godzina, Minuty) values
(1, 2000, 6, 20, 1, 30);

insert into Rodzaj_operacji (Id_rodzaju_operacji, Rodzaj_operacji, Oplata)
values (2, 'Elixir', 5);

insert into Transakcje (Id_transakcji, Id_rachunku_nadawcy, Id_oddzialu,
Id_rodzaju_operacji, Id_czasu, Id_rachunku_odbiorcy, Kwota) values (1, 115,
48, 3, 245, 39, 568.13);
```

*Zdefiniowanie w języku SQL poleceń dla realizacji typowych operacji (wstawianie, usuwanie, modyfikacja, selekcja/prezentacja)*

Poniższe listingi przedstawiają typowe operacje.

```
/* Wyswietla historie transakcji wybranej osoby*/
SELECT w.Imie_wlasciciela AS "Imie nadawcy", w.Nazwisko_wlasciciela AS
"Nazwisko nadawcy", r.Nr_rachunku AS "Nr rachunku nadawcy",
w2.Imie_wlasciciela AS "Imie odbiorcy", w2.Nazwisko_wlasciciela AS
"Nazwisko odbiorcy", r2.Nr_rachunku AS "Nr rachunku odbiorcy", t.kwota,
c.rok || '-' || c.miesiac || '-' || c.dzien || ' ' ||  c.godzina || ':' ||
c.minuty AS "Data transakcji"
FROM Transakcje t
JOIN Rachunek r
ON t.id_rachunku_nadawcy = r.id_rachunku
JOIN Wlasciciel w
ON r.id_wlasciciela = w.id_wlasciciela
JOIN Rachunek r2
ON t.id_rachunku_odbiorcy = r2.id_rachunku
JOIN Wlasciciel w2
ON r2.id_wlasciciela = w2.id_wlasciciela
JOIN Czas c
ON c.id_czasu = t.id_czasu
WHERE w.id_wlasciciela =  &id_wlasciciela;


/* Tworzy nowy adres a nastepnie nowego wlasciciela i przypisuje mu ten
adres*/
INSERT INTO Adres
(id_adresu, kod_pocztowy, ulica, nr_lokalu, nr_mieszkania, miejscowosc)
VALUES ((SELECT MAX(id_adresu) + 1 FROM Adres), '&kod_pocztowy', '&ulica',
&nr_lokalu, &nr_mieszkania, '&miejscowosc');

INSERT INTO Wlasciciel
(id_wlasciciela, id_adresu, imie_wlasciciela, nazwisko_wlasciciela,
plec_wlasciciela, pesel_wlasciciela)
VALUES ((SELECT MAX(id_wlasciciela) + 1 FROM Wlasciciel), (SELECT
MAX(id_adresu) FROM Adres), '&imie_wlasciciela', '&nazwisko_wlasciciela',
'&plec_wlasciciela', '&pesel_wlasciciela');

/*Usuwa nieużywane adresy*/
DELETE FROM Adres
WHERE id_adresu NOT IN (SELECT id_adresu_oddzialu FROM Oddzialy) AND
id_adresu NOT IN (SELECT id_adresu FROM Wlasciciel);
```

```sql
/*Aktualizacja danych osobowych wybranego wlasciciela*/
UPDATE Wlasciciel
SET imie_wlasciciela = '&imie_wlasciciela', nazwisko_wlasciciela =
'&nazwisko_wlasciciela', plec_wlasciciela = '&plec_wlasciciela',
pesel_wlasciciela = '&pesel_wlasciciela'
WHERE id_wlasciciela = &id_wlasciciela;

/*usuwa wybranego wlasciciela*/
DELETE FROM Wlasciciel
WHERE id_wlasciciela = &id_wlasciciela;

/*sprawdza stan konta*/
SELECT r.saldo AS "stan konta"
FROM Rachunek r
JOIN Wlasciciel w
ON r.id_wlasciciela = w.id_wlasciciela
WHERE r.id_wlasciciela = &id_wlasciciela;

/*Wyswietlanie transakcji wykonanych przez danego klienta i odebranych
przez niego. Zakladamy, że dwa razy nalezy podac to samo id_rachunku.*/
SELECT * FROM Transakcje WHERE Id_rachunku_nadawcy = &Id_rachunku
UNION
SELECT * FROM Transakcje WHERE Id_rachunku_odbiorcy = &Id_rachunku;

/* Wyswietlenie sald wszystkich klientów wraz z ich id.*/
SELECT r.id_wlasciciela, r.saldo FROM Rachunek r
ORDER BY r.id_wlasciciela;

/* Wyswietlenie salda konkretnego klienta wraz z jego id.*/
SELECT r.id_wlasciciela, r.saldo FROM Rachunek r
WHERE r.id_wlasciciela = &id_wlasciciela;

 /*Wyświetlenie wydatkow wszystkich klientów*/
SELECT r.id_wlasciciela, SUM(t.kwota) AS "WYDATKI"  FROM Transakcje t JOIN
Rachunek r ON t.Id_rachunku_nadawcy = r.Id_rachunku
GROUP BY r.id_wlasciciela ORDER BY r.id_wlasciciela;

/*Wyświetlenie wydatkow konkretnego klienta*/
SELECT r.id_wlasciciela, SUM(t.kwota) AS "WYDATKI" FROM Transakcje t JOIN
Rachunek r ON t.Id_rachunku_nadawcy = r.Id_rachunku
WHERE r.id_wlasciciela = &id_wlasciciela
GROUP BY r.id_wlasciciela ;

/*Wyświetlenie wydatkow wszystkich klientów*/
SELECT r.id_wlasciciela, SUM(t.kwota) AS "PRZYCHOD"
FROM Transakcje t
JOIN Rachunek r ON t.Id_rachunku_odbiorcy = r.Id_rachunku
GROUP BY r.id_wlasciciela ORDER BY r.id_wlasciciela;

/*Wyświetlenie wydatkow konkretnego klienta*/
SELECT r.id_wlasciciela, SUM(t.kwota) AS "PRZYCHOD"
FROM Transakcje t
JOIN Rachunek r ON t.Id_rachunku_odbiorcy = r.Id_rachunku
WHERE r.id_wlasciciela = &id_wlasciciela
GROUP BY r.id_wlasciciela ;

/* Wyswietlanie ilosci transakcji dla kazdego oddzialu.*/
SELECT Id_oddzialu, Count(*) AS "ILOSC_TRANSAKCJI"
FROM Transakcje t
JOIN Oddzialy o USING (Id_oddzialu)
GROUP BY Id_oddzialu ORDER BY Id_oddzialu;
```

```
/* Wyswietlanie ilosci transakcji dla każdego rodzaju transakcji.*/
SELECT r.rodzaj_operacji, COUNT(*) AS "ILOSC_OPERACJI" FROM Transakcje t
JOIN Rodzaj_operacji r USING (id_rodzaju_operacji)
GROUP BY r.rodzaj_operacji ORDER BY r.rodzaj_operacji;
```

# Podsumowanie

## OKREŚLENIE, KTÓRE Z ZAŁOŻEŃ PRZYJĘTYCH DO PROJEKTU BAZY DANYCH ZOSTAŁY SPEŁNIONE, KTÓRE NIE I Z JAKIEGO POWODU.

Zostało zaimplementowane sprawdzanie stanu konta oraz historii transakcji dla określonego id właściciela – jest to podstawa do późniejszego rozszerzenia o sprawdzanie swojego stanu konta przez konkretnego użytkownika.

Nie udało się zaimplementować wykonywania operacji, ponieważ pojawiły się problemy związane z używaniem procedur, które były niezbędne w realizacji zaplanowanej struktury. W związku z tym nie wdrożono także podsumowania stanu konta po każdej operacji, jednakże, tak jak wspomniano powyżej, jest możliwość wyświetlania salda w każdej chwili.

Nie zostały wdrożone osobne widoki dla administratora i użytkownika ze względu na brak aplikacji zewnętrznej, jednak zostały wdrożone przykładowe operacje dostępne dla administratora: usuwanie, dodawanie, modyfikacja i wyświetlanie. Natomiast dla użytkownika zostały zaimplementowane zapytania pozwalające sprawdzić podstawowe informacje (np. przychód, wydatki).

## WNIOSKI

Podczas realizacji projektu największą trudność sprawiło stworzenie nowej bazy danych przy użyciu Oracle SQL Developera i braku własnego serwera. Dobrym rozwiązaniem okazało się skorzystanie z dostępnego na laboratorium serwera.

Dzięki dobrze dopracowanemu diagramowi ERD implementacja obiektów nie przysporzyła dużych problemów. Przy pisaniu zapytań należało się zastanowić nad informacjami jakie mogłyby interesować potencjalnych użytkowników tej bazy danych.

Wybrany generator danych wydał się najlepszym spośród darmowych, ale niestety sprostał oczekiwaniom. Na przykład pomimo zerowej szansy na wartość NULL z dużą częstotliwością była ona otrzymywana. Brakowało również możliwości ustawienia formatu

danych zgodnych z polskim standardem, przez co powstała potrzeba zmiany ograniczeń dla niektórych atrybutów.

Mimo, że przed przystąpieniem do projektu wydawało się, że stworzenie bazy danych i operacje na niej są stosunkowo nieskomplikowane i mało czasochłonne, okazało się, że jest zupełnie odwrotnie i pozornie najprostsze rzeczy sprawiły najwięcej problemów.

# LITERATURA


[1] Chałon, M. (2001). *Systemy Baz Danych - Wprowadzenie.* Oficyna Wydawnicza Politechniki Wrocławskiej.

[2] Kukuczka, J. (2000). *Relacyjne Bazy Danych.* Wydawnictwo Wyższej Szkoły Infromatyki i Zarządzania Bielsko-Biała.