**BD1 – Opis projektu – zespół nr 88**

Członkowie zespołu: Jakub Proboszcz, Kamil Michalak

Temat projektu: Obsługa banku (wpłat, wypłat, rejestracji klientów, itp.).

Nasz projekt reprezentuje bazę danych banku. Przechowywane są dane walut, klientów, ich kont, sald ich kont w różnych walutach, kredytów (loans), lokat (investments) oraz historia przelewów wewnętrznych i zewnętrznych, zgodnie ze schematami w plikach   
„Schemat E-R.png” oraz „Schemat relacyjny.png”. Jedyna relacja wiele do wielu – pomiędzy klientami a kontami (założyliśmy, że może być wielu właścicieli konta) – została odwzorowana do tabeli pośredniej CLIENTS\_ACCOUNTS.

Więzy integralności bazy danych są sprawdzane ograniczeniami na poziomie kolumn (np. nieujemność kwoty lokaty), ograniczeniami na poziomie tabeli (np. kolejność dat utworzenia i zamknięcia konta) oraz wyzwalaczami (np. istnienie konta w czasie wzięcia kredytu).

Skrypty tworzące tabele są zawarte w pliku „ddl.sql”, a skrypty wstawiające dane do tabel są zawarte w pliku „data\_creation.sql”.

Zdefiniowaliśmy następujące nazwane bloki PL/SQL:

* funkcja monthly\_card\_payment: oblicza miesięczną opłatę za karty płatnicze dla danego klienta
* funkcja calculate\_total\_balance: oblicza ogólne saldo w PLN wszystkich kont klienta, wliczając pożyczki i lokaty
* procedura take\_loan: rejestruje wzięcie pożyczki przez podane konto i dodaje pieniądze do konta
* procedura make\_investment: rejestruje założenie lokaty przez podane konto i odejmuje pieniądze z konta
* procedura make\_inside\_transaction: rejestruje przelew wewnętrzny i dodaje oraz odejmuje pieniądze zgodnie z obecnym kursem walut
* procedura make\_outside\_transaction: rejestruje przelew zewnętrzny (wychodzący do lub przychodzący z innego banku) i dodaje lub odejmuje odpowiednią ilość pieniędzy
* procedura block\_unpaid\_accounts: blokuje wszystkie konta, które przez rok nie spłaciły 10% wziętej pożyczki; wykorzystuje kursor
* wyzwalacze inside\_transaction\_date\_trig, outside\_transaction\_date\_trig, loan\_date\_trig, investment\_date\_trig: weryfikują, czy konta uczestniczące w przelewie lub biorące lokatę/kredyt istniały w danym momencie

Wszystkie te funkcje, procedury i wyzwalacze są zdefiniowane w pliku „plsql\_named\_blocks.sql”. Skrypty demonstrujące ich działanie (i wykonujące nietrywialne zapytania SELECT do bazy danych) są zawarte w pliku „testing.sql”.

Kod źródłowy naszej aplikacji w JAVIE jest w repozytorium pod adresem <https://gitlab-stud.elka.pw.edu.pl/jprobosz/22Z_BD1_z88_Obsluga_banku>.

Program może wykonać 10 różnych funkcjonalności odczytujących dane z bazy danych lub zapisujących dane do bazy danych poprzez polecenie UPDATE albo zdefiniowane wcześniej procedury. Niektóre z nich pobierają parametry od użytkownika i przekazują je do bazy danych.

Komunikacja z bazą danych jest zrealizowana przez JDBC.

**Analiza rozwiązania**

Nasza baza danych skutecznie reprezentuje najbardziej kluczowe części obsługi banku – klientów, ich konta oraz ich transakcje. Wystarczyłoby to do realizacji obsługi banku, jeżeli wszystkie bardziej skomplikowane aspekty byłyby zrealizowane w aplikacji. Wykonywana jest weryfikacja poprawności danych w bazie praktycznie w całości – brakuje jedynie sprawdzania niektórych bardziej skomplikowanych zasad, jak np. poprawność PESELu lub numerów kont.

Pozostaje jednak wiele ważnych szczegółów, które mogłyby też zostać zareprezentowane w bazie danych banku. Można by było rozszerzyć naszą bazę danych np. w kierunku przechowywania większej ilości danych o klientach, między innymi w celu oceny zdolności kredytowej lub zapobiegania praniu pieniędzy (co jest obowiązkiem banków). Obecnie przechowywane dane zdecydowanie nie wystarczałyby do takich analiz. Można by też było przechowywać więcej informacji o innych bankach, do komunikacji z nimi oraz do weryfikacji poprawności przelewów zewnętrznych.

Ogólnie, nasza baza danych jest spójnym i kompletnym rozwiązaniem problemu obsługi danych banku – znacznie uproszczonym w stosunku do świata rzeczywistego, ale jest to spodziewane po krótkim projekcie, który nie będzie użyty na potrzeby prawdziwego banku.