

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,
- dane klientów,
- dane kont klientów,
- salda kont klientów w różnych walutach,
- karty płatnicze,
- wzięte kredyty (loans),
- założone lokaty (investments),
- historia przelewów wewnętrznych,
- historia przelewów zewnętrznych,

zgodnie ze schematami w plikach „*Schemat E-R.png*” oraz „*Schemat relacyjny.png*”.

Jedyna relacja „wiele do wielu” występuje pomiędzy klientami, a kontami (założyliśmy, że może być wielu właścicieli konta) – została ona odwzorowana do tabeli pośredniej CLIENTS_ACCOUNTS.

Nowe numery kont są generowane z sekwencji account_numbers_seq.

Więzy integralności bazy danych są sprawdzane ograniczeniami:

- na poziomie kolumn (np. nieujemność kwoty lokaty),
- na poziomie tabeli (np. kolejność dat utworzenia i zamknięcia konta),
- wyzwalaczami (np. istnienie konta w czasie wzięcia kredytu).

Skrypty tworzące tabele (i sekwencję) 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 lub przychodzący z innego banku) i dodaje lub odejmuje odpowiednią ilość pieniędzy,
- procedura **block_unpaid_accounts**: blokuje wszystkie konta, które przez rok nie spłaciły więcej niż 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 gitowym pod adresem:

https://gitlab-stud.elka.pw.edu.pl/jprobosz/22Z_BD1_z88_Obsluga_banku

Program może wykonać 10 różnych funkcjonalności odczytujących 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**.

W razie potrzeby można bazę danych usunąć uruchamiając skrypt „*drops.sql*”.

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 na poziomie 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 zaimplementowane 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 przez nas 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.