Bissenova Anara / 20BD / Lab7

TASK1. How can we store large-object types?

Large objects are stored as a large object:

- blob: binary large object - object is a large collection of uninterpreted binary
- clob: character large object - object is a large collection of character data

When a query returns a large object, a pointer is returned rather than the large object itself.

TASK2. What is the difference between privilege, role and user?

- Privileges control the ability to run SQL statements.
- A role is a group of privileges.
- Granting a role to a user gives them the privileges contained in the role.

```
○ create accountant, administrator, support roles and grant appropriate privileges
○ create some users and assign them roles
○ give to some of them permission to grant roles to other users
○ revoke some privilege from particular user

create role accountant;
create role administrator;
create role support;
grant all privileges on accounts, customers, transactions to administrator;
grant select on accounts, transactions, customers to support with grant option;
grant insert, update, select on transactions to accountant;

create user Mary;
create user Richard;
create user Ann;
grant accountant to Mary;
grant administrator to Ann;
grant support to Richard;

revoke grant option for select on accounts from Richard;
revoke update on transactions from accountant;
```

TASK3. Add appropriate constraints

- check if transaction has same currency for source and destination accounts (use assertion) - ?
- add not null constraints

```
create table accounts(
    account_id varchar(40) primary key ,
    customer_id integer references customers(id),
    currency varchar(3) not null,
    balance float,
    "limit" float
);
create table transactions (
    id serial primary key ,
    date timestamp,
    src_account varchar(40) references accounts(account_id),
    dst_account varchar(40) references accounts(account_id),
    amount float not null,
    status varchar(20) not null
);
```

TASK4. Change currency column type to user-defined in accounts table - ?

TASK5. Create indexes:

- index so that each customer can only have one account of one currency
- index for searching transactions by currency and balance

```
create unique index accounts_with_currency on accounts(customer_id, currency);
create index transactions_by_c_and_b on accounts(currency, balance);
```

TASK6. Write a SQL transaction that illustrates money transaction from one account to another:

- create transaction with “init” status
- increase balance for destination account and decrease for source account
- if in source account balance becomes below limit, then make rollback
- update transaction with appropriate status(commit or rollback)

```sql
begin transaction;
update accounts
    set balance = balance + (select amount from transactions_with_check where status = 'init')
        where account_id = (select dst_account from transactions_with_check where status = 'init');
update accounts
    set balance = balance - (select amount from transactions_with_check where status = 'init')
        where account_id = (select src_account from transactions_with_check where status = 'init');

if exists (select account_id, balance from transactions_with_check inner join accounts on
transactions_with_check.src_account = account_id where status = 'init' and balance < accounts.limit) then
    begin;
    rollback;
    update transactions_with_check
    set status = 'rollback' where id = (select id from transactions_with_check where status = 'init');
    end;
else
    begin;
    update transactions_with_check
    set status = 'committed' where id = (select id from transactions_with_check where status = 'init');
    commit;
    end;
end transaction ;
```