Transactions – seminar draft

Demo Database

The demo database *demotest* has been created on the slon.felk.cvut.cz with the following statements. Discuss them, especially:

- find the keys,
- find the artificial keys,
- find the foreign keys,
- find the table representing a week entity type;
- check appropriate DEFAULT values.

```
CREATE SEQUENCE seq_accountowner;

CREATE TABLE accountowner (
     id_accountowner int DEFAULT nextval('seq_accountowner'),
     name varchar(128) NOT NULL,
     address varchar (256) NOT NULL,
     datebirth date NOT NULL,
     createdby varchar(64) NOT NULL DEFAULT current_user,
     created timestamp NOT NULL DEFAULT now(),
     UNIQUE (name, address, datebirth),
     PRIMARY KEY (id_accountowner));

CREATE TABLE bankaccount (
     id_accountowner int NOT NULL REFERENCES accountowner,
     accountnumber numeric(10) NOT NULL,
     ammount numeric (18,2) NOT NULL DEFAULT 0 CHECK (ammount>0),
     ammount2 float NOT NULL DEFAULT 0 CHECK (ammount2>0),
     PRIMARY KEY (id_accountowner, accountnumber));

CREATE TABLE banktransfer (
     id_accountowner int,
     accountnumber numeric(10),
     id_accountowner_source int,
     accountnumber_source numeric(10),
     created timestamp NOT NULL DEFAULT now(),
     finished timestamp,
     ammount numeric(18,2) NOT NULL CHECK (ammount>0),
     FOREIGN KEY (id_accountowner,accountnumber)
          REFERENCES bankaccount (id_accountowner, accountnumber),
     FOREIGN KEY (id_accountowner_source,accountnumber_source)
          REFERENCES bankaccount (id_accountowner, accountnumber),
     PRIMARY KEY (id_accountowner, accountnumber,
          id_accountowner_source, accountnumber_source, created));
```

Tasks

1. Login to the *demotest* database on *slon* server (use your role). Be sure your client is not in the “autocomit mode”
2. Insert yourself as a new bank account owner
3. create your own bank account and set the initial ammount
4. with your colleagues, try to provide the bank transfer from/to your account including
    - the creation of a bank transfer
    - the changes of particular account total ammount
    - the finalization of the bank transfer (update the *finalized* column).
5. repeat the step 4 in a way corrupting the total ammount constraint – manually try to return the state into a previous consistent state
6. repeat the step 5 using the transaction and rollback
    - discuss, what’s happend
7. simulate/discuss the situations, when
    - it is not trivially possible return to previous consistent state
    - there are the conflicts (not only on the failures)
8. with the appropriate isolation level, try to provide the step 6 in the correct way.
    - discuss, what’s happened

Solution

Insert yourself as a new bank account owner:

```
INSERT INTO accountowner (name, address,datebirth)
     VALUES ('Mxx R','Piskova 80, Pribram', '1950-04-01')
```

Create your own bank account and set the initial ammount:

```
INSERT INTO bankaccount (id_accountowner,accountnumber,ammount)
     VALUES (1,589,1000);
```

With your colleagues, try to provide the bank transfer from/to your account

```
INSERT INTO banktransfer (id_accountowner, accountnumber,
id_accountowner_source, accountnumber_source,ammount)
     VALUES(1,567,1,765,100);

UPDATE bankaccount
     SET ammount=ammount+100
     WHERE id_accountowner=1 AND  accountnumber = 567;
UPDATE bankaccount
     SET ammount=ammount-100
     WHERE id_accountowner=1 AND  accountnumber = 765;

UPDATE banktransfer
     SET finished = now()
     WHERE …..
```

repeat the step 4 in a way corrupting the total ammount constraint – manually try to return the state into a previous consistent state

repeat the step 5 using the transaction and rollback

BEGIN
....
COMMIT/ROLLBACK

simulate/discuss the situations, when
(follow the scenarios from the lecture slides)