

Limbajul SQL— aplicatii

- Proceduri si functii stocate
- Tranzactionare
- Utilizarea declansatoarelor (trigger)
- Aplicatie: o aplicatie bancara simplificata

Aplicatie: aplicatie bancara simplificata

DDL: Trigger

DML: Update, Insert

Proceduri si functii stocate

Sintaxa pentru comenzile utilizate

Trigger

```
CREATE [DEFINER = { user | CURRENT_USER }]
      TRIGGER trigger_name trigger_time trigger_event
      ON tbl_name
      FOR EACH ROW trigger_stmt
```

```
DROP TRIGGER [IF EXISTS]
[schema_name.] trigger_name
```

Update

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name
      SET col_name1={expr1|DEFAULT} [, col_name2={expr2|DEFAULT}] ...
      [WHERE where_condition]
      [ORDER BY ...]
      [LIMIT row_count]
```

SQL

Insert

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name [(col_name, ...)]
      {VALUES | VALUE} ({expr | DEFAULT}, ...), (...), ...
      [ ON DUPLICATE KEY UPDATE
        col_name=expr [, col_name=expr] ... ]
```

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name
      SET col_name={expr | DEFAULT}, ...
      [ ON DUPLICATE KEY UPDATE
        col_name=expr [, col_name=expr] ... ]
```

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name [(col_name, ...)]
      SELECT ...
      [ ON DUPLICATE KEY UPDATE
        col_name=expr [, col_name=expr] ... ]
```

Proceduri si functii stocate

```
CREATE [DEFINER = { user | CURRENT_USER }]
PROCEDURE sp_name ([proc_parameter[, ...]])
[characteristic ...] routine_body
```

```
CREATE [DEFINER = { user | CURRENT_USER }]
FUNCTION sp_name ([func_parameter[, ...]])
RETURNS type [characteristic ...] routine_body
```

proc_parameter:

[IN | OUT | INOUT] *param_name type*

func_parameter:

param_name type

type:

Any valid MySQL data type

characteristic:

```
LANGUAGE SQL | [NOT] DETERMINISTIC | { CONTAINS
SQL |
NO SQL | READS SQL DATA | MODIFIES SQL DATA } |
SQL SECURITY { DEFINER | INVOKER } | COMMENT
'string'
```

Aplicatia bancara simplificata

Descriere

Aplicatia gestioneaza conturile clientilor unei banci.

Operatiunile permise sunt:

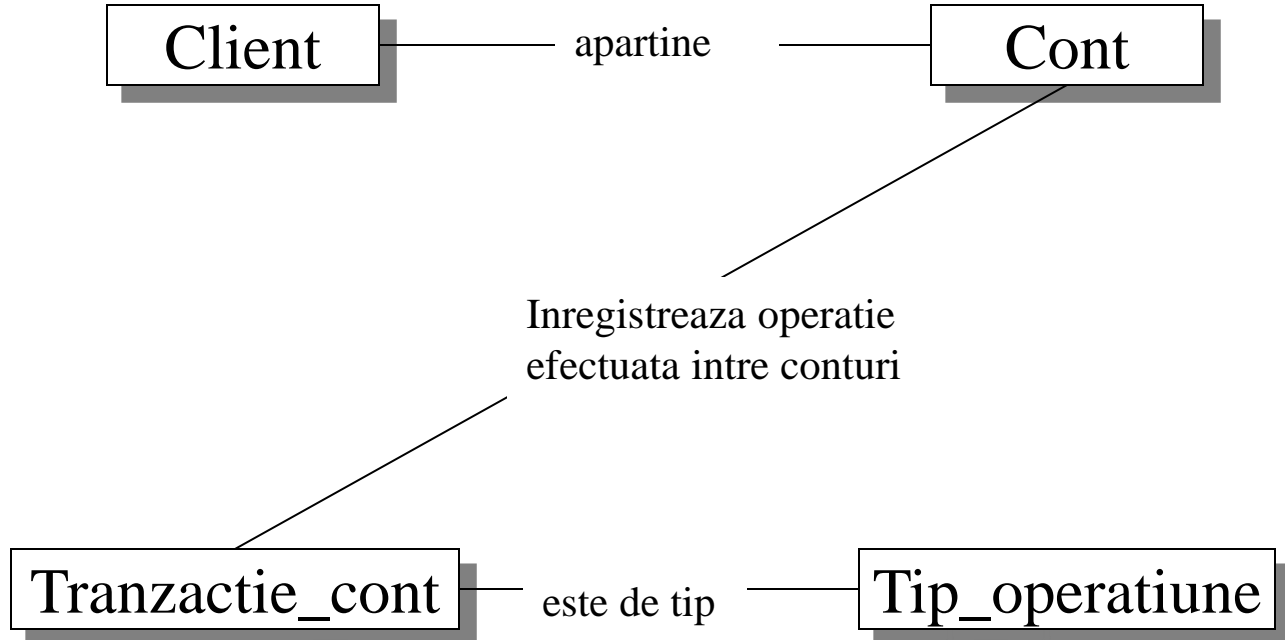
- Deschidere cont,
- Inchidere cont,
- Transfer intre conturi,
- Depunere ghiseu,
- Depunere ATM,
- Extragere ghiseu,
- Extragere ATM,
- Balanta cont,
- Istoric tranzactii;

Erorarea de tranzactionare:

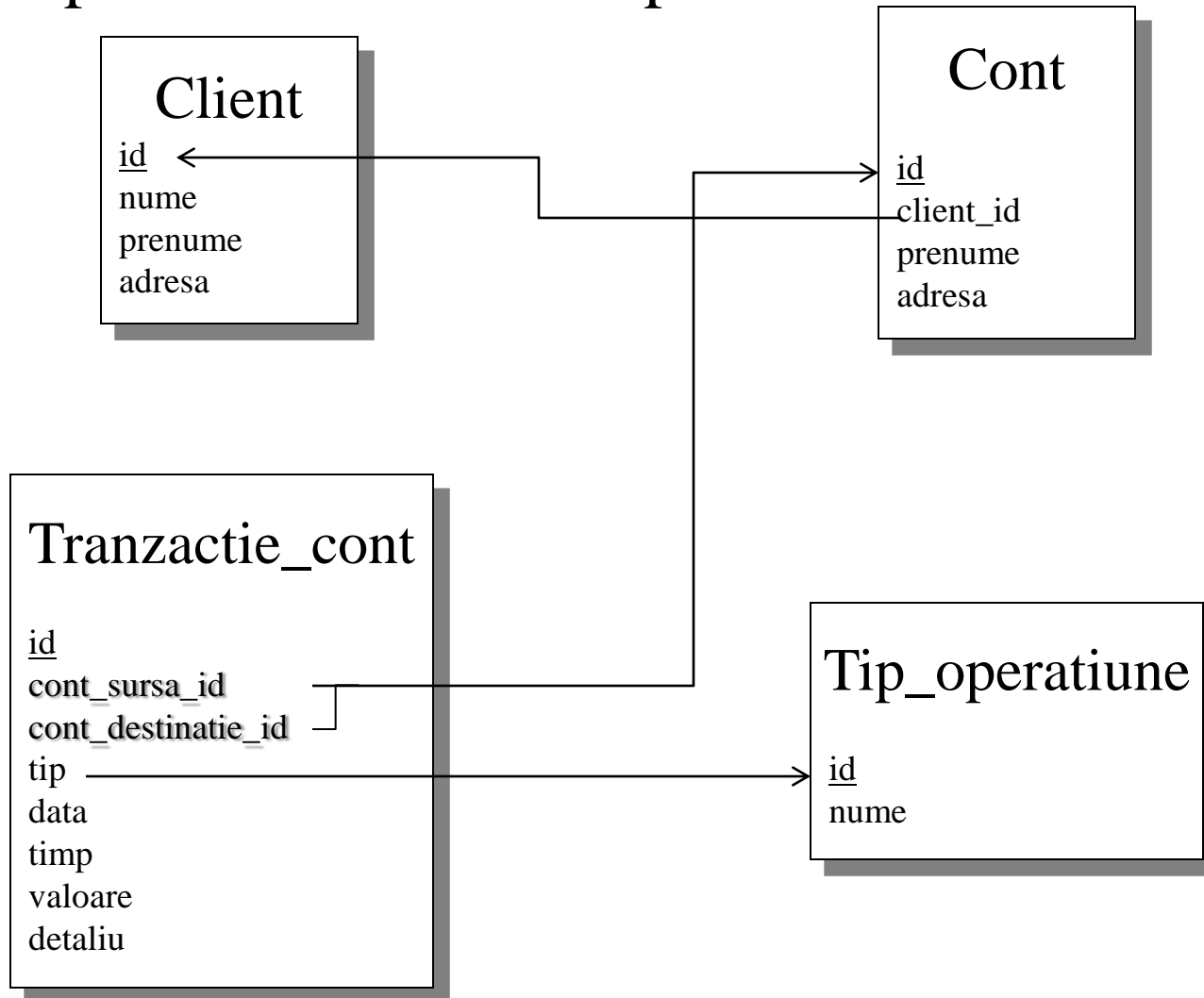
- TR_ERR: cont creditor insuficient;

Aplicatia bancara simplificata

Modelul entitate-relatie



Aplicatia bancara simplificata



Aplicatia bancara simplificata

Implementare

- ❑ Informatiile despre conturile clientilor sunt stocate in tabela **client**; informatiile despre conturile clientilor sunt stocate in tabela **cont**;
- ❑ Tipurile de operatiuni sunt stocate in tabela (nomenclator) **tip_operatiune**; tabela **tranzactie_cont** stocheaza informatiile despre operatiunile executate, inclusiv despre operatiuni de interogare precum 'Istoric tranzactii';
- ❑ Operatiunile care implica transferuri intre conturi sunt executate in interiorul unor tranzactii.
- ❑ Logica operatiunilor este implementata folosind:
 - Triggere (pentru creare cont la crearea unui client);
 - Proceduri stocate (pentru diferite operatiuni);
- ❑ Se utilizeaza triggere pentru unele operatiuni care trebuie inregistrate automat (vezi exemple mai jos).

Aplicatie bancara simplificata

Crearea tabelor bazei de date

```
##-----  
##  
## creare tabele client, cont, tip_operatiune, tranzactie_cont  
##  
##-----  
create DATABASE IF NOT EXISTS d;  
USE d;  
DROP TABLE IF EXISTS cont;  
DROP TABLE IF EXISTS client;  
CREATE TABLE IF NOT EXISTS client  
(id int unique auto_increment primary key,  
nume char(20),  
prenume char(20),  
adresa char(50));  
  
CREATE TABLE IF NOT EXISTS cont  
(id int unique auto_increment primary key,  
client_id int,  
cod int,  
valoare float,  
FOREIGN KEY (client_id) REFERENCES client(id));  
  
CREATE TABLE IF NOT EXISTS tip_operatiune  
(id int unique auto_increment primary key,  
nume char(40));  
  
CREATE TABLE IF NOT EXISTS tranzactie_cont  
(id int unique auto_increment primary key,  
cont_sursa_id int,  
cont_destinatie_id int,  
tip int,  
data date,  
timp time,  
valoare float,  
detaliu char(20),  
FOREIGN KEY (cont_sursa_id) REFERENCES cont(id),  
FOREIGN KEY (cont_destinatie_id) REFERENCES cont(id),  
FOREIGN KEY (tip) REFERENCES tip_operatiune(id));
```

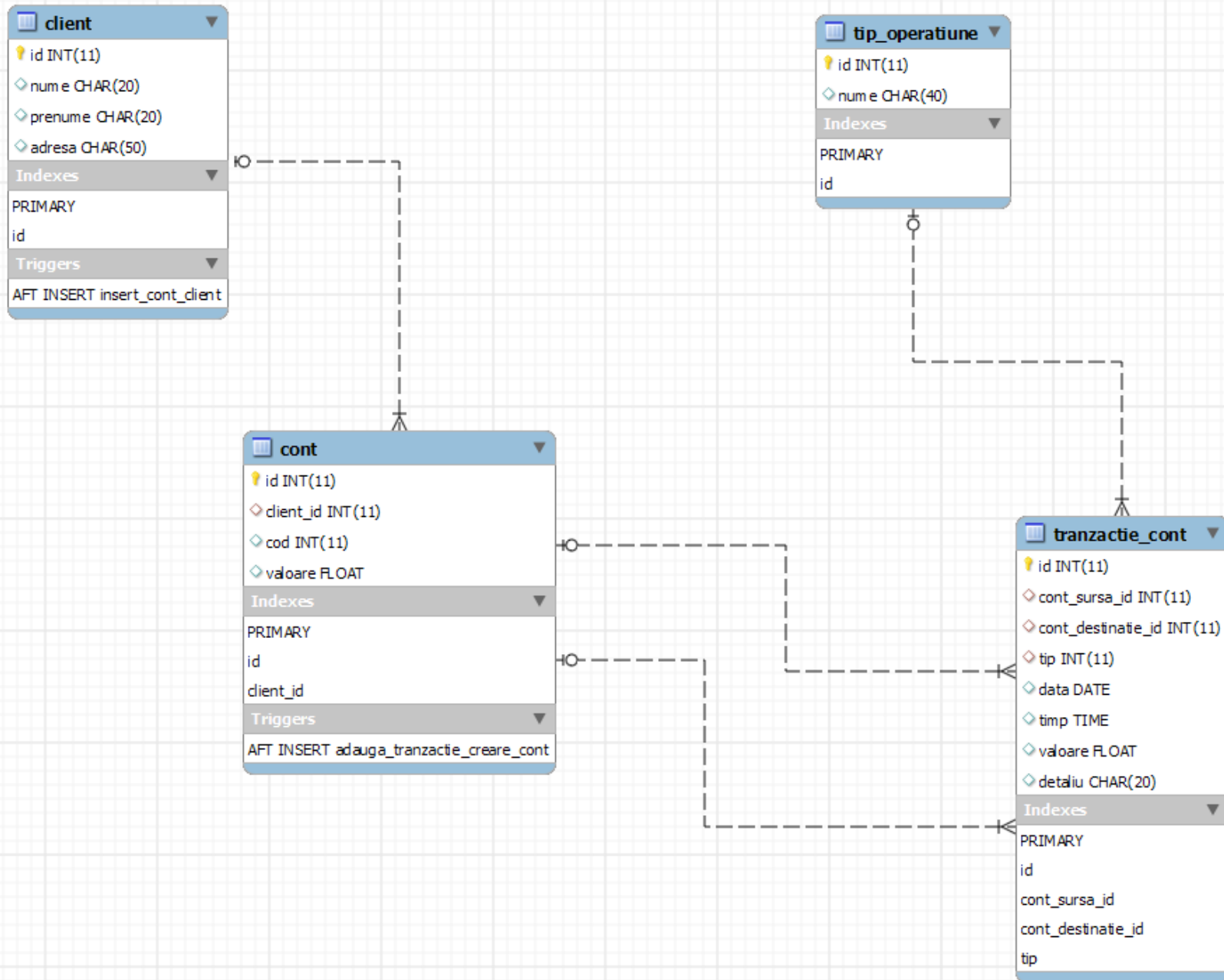
SQL

Triggere-le (declansatorii) sunt folosite pentru doua scopuri in aceasta aplicatie:

- (1) La crearea unui client, i se creeaza automat si un cont. Clientii sunt creati prin adaugarea unei noi inregistrari in tabela de client; trigger-ul este 'insert_cont_client', declanseaza dupa INSERT in tabela client si are ca efect inserarea unei noi inregistrari in tabela cont.
- (2) La adaugarea unei inregistrari in tabela cont (la crearea unui cont), se adauga o inregistrare de tip 'Deschidere cont' in tabela 'tranzactie_cont', tabela care inregistreaza toate tranzactiile/operatiunile care au loc.

Am preferat ca alte tipuri de modificari ale 'cont' si 'tranzactie_cont' sa nu utilizeze trigger pentru modificari corespunzatoare in tabela complementara, incluzandu-le in cadrul unor tranzactii, pentru un control mai strans.

```
##-----  
##  
## creare trigger  
##  
##-----  
  
## - Trigger AFTER INSERT tabela client  
## Este folosit pentru a adauga automat un cont pentru noul client  
DROP TRIGGER IF EXISTS insert_cont_client;  
  
delimiter //  
  
CREATE TRIGGER insert_cont_client AFTER INSERT ON client  
FOR EACH ROW BEGIN  
    INSERT INTO cont (client_id, cod, valoare) VALUES(NEW.id, NEW.id * 100,  
    END; //  
  
delimiter ;  
  
## - Trigger AFTER INSERT tabela cont  
## Este folosit pentru a adauga o inregistrare pentru crearea contului,  
## in tranzactie_cont  
DROP TRIGGER IF EXISTS adauga_tranzactie_creare_cont;  
  
delimiter //  
  
CREATE TRIGGER adauga_tranzactie_creare_cont AFTER INSERT ON cont  
FOR EACH ROW BEGIN  
    INSERT INTO tranzactie_cont  
    (cont_sursa_id, cont_destinatie_id,tip, data, timp, valoare) VALUES  
    (NEW.id, NEW.id, |  
    (SELECT id from tip_operatiune WHERE nume = 'Deschidere cont'),  
    CURRENT_DATE(), NOW(),0);  
    END; //  
  
delimiter ;
```



SQL

```
##-----  
##  
## inserare data in tabellele de client, tip_operatiune  
##  
##-----  
insert into tip_operatiune  
(nume)  
VALUES  
( 'Deschidere cont' ),  
( 'Inchidere cont' ),  
( 'Transfer intre conturi' ),  
( 'Depunere ghiseu' ),  
( 'Depunere ATM' ),  
( 'Extragere ghiseu' ),  
( 'Extragere ATM' ),  
( 'Balanta cont' ),  
( 'Istoric tranzactii' ),  
( 'TR_ERR: cont creditor insuficient' );  
  
INSERT INTO client  
(nume, prenume, adresa)  
VALUES  
( 'Popa', 'Ion', 'Str. Mare, Dabuleni, Ilfov' ),  
( 'Popescu', 'Maria', 'Blvd. Primaverii, Bucuresti' ),  
( 'Andreescu', 'Victor', 'Blvd. Iuliu Maniu, Bucuresti' ),  
( 'Predescu', 'Andreea', 'Str. Virtutii, Baia Mare' ),  
( 'Ionescu', 'Mihaela', 'Str. Visinilor, Bucuresti' ),  
( 'Dinu', 'Cornel', 'Calea Victoriei, Bucuresti' ),  
( 'Vodafone', 'S.A', 'Bucuresti' ),  
( 'Romtelecon', 'S.A', 'Bucuresti' );
```

```
mysql> select * from tranzactie_cont where tip = 1;
```

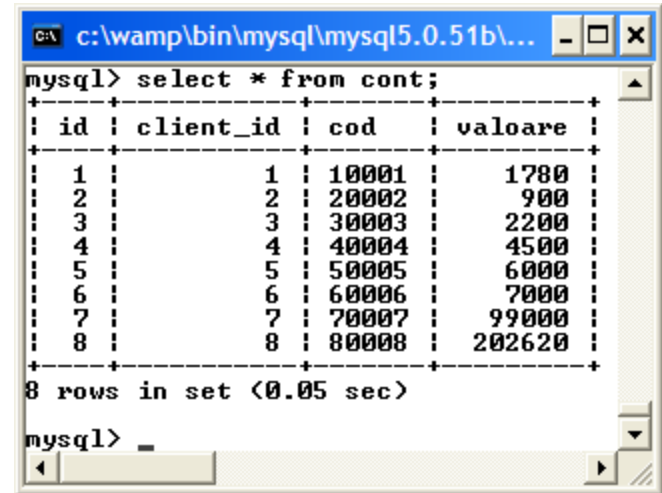
id	cont_sursa_id	cont_destinatie_id	tip	data	timp	valoare	detaliu
1	1	1	1	2008-12-14	12:07:47	0	NULL
2	2	2	1	2008-12-14	12:07:47	0	NULL
3	3	3	1	2008-12-14	12:07:47	0	NULL
4	4	4	1	2008-12-14	12:07:47	0	NULL
5	5	5	1	2008-12-14	12:07:47	0	NULL
6	6	6	1	2008-12-14	12:07:47	0	NULL
7	7	7	1	2008-12-14	12:07:47	0	NULL
8	8	8	1	2008-12-14	12:07:47	0	NULL

Populare tabelle

Nu se populeaza tabellele 'cont' si 'tranzactie_cont':

In tabela 'cont' se creeaza automat, utilizand primul trigger, inregistrările corespunzand clientilor creati in tabela 'client';

In tabela 'tranzactie_cont' se creeaza automat, utilizand al doilea trigger, inregistrările corespunzatoare creării conturilor.



```
c:\wamp\bin\mysql\mysql5.0.51b\... - _ X  
mysql> select * from cont;  
+----+-----+-----+-----+  
| id | client_id | cod | valoare |  
+----+-----+-----+-----+  
| 1 | 1 | 10001 | 1780 |  
| 2 | 2 | 20002 | 900 |  
| 3 | 3 | 30003 | 2200 |  
| 4 | 4 | 40004 | 4500 |  
| 5 | 5 | 50005 | 6000 |  
| 6 | 6 | 60006 | 7000 |  
| 7 | 7 | 70007 | 99000 |  
| 8 | 8 | 80008 | 202620 |  
+----+-----+-----+-----+  
8 rows in set (0.05 sec)  
mysql> _
```

Procedura alimentare cont

```
DELIMITER //

CREATE PROCEDURE ALIMENTARE_CONT(cont_destinatie_cod int, valoare_depunere float)

BEGIN
    START TRANSACTION;
    SELECT @ID_D := id FROM cont WHERE cod = cont_destinatie_cod;
    SELECT @VAL := valoare FROM cont WHERE id = @ID_D;
    SELECT @ID_TIP := id FROM tip_operatiune WHERE nume = 'Depunere ghiseu';

    IF(@ID_D IS NOT NULL) THEN
        BEGIN
            INSERT INTO tranzactie_cont
                (cont_sursa_id, cont_destinatie_id, valoare, tip, data, timp,detaliu)
            VALUES
                (NULL, @ID_D, valoare_depunere, @ID_TIP, CURRENT_DATE(), NOW(), 'Depunere ghiseu');

            UPDATE cont SET valoare = valoare + valoare_depunere WHERE id = @ID_D;

            COMMIT;
        END;
    ELSE ROLLBACK;
    END IF;
END //
DELIMITER ;
```

Procedura primeste ca parametrii codul contului destinatie si valoarea depunerii; tipul operatiunii este 'Depunere ghiseu'; tranzactia consta in 2 operatiuni complementare:

- inregistrarea unei tranzactii in tabela 'tranzactie_cont' si
- cresterea soldului contului accesat cu aceeasi valoare (tabela 'cont').
- Tranzactia este finalizata numai daca se opereaza cu un cont destinatie existent; altfel, se face rollback (tranzactia este anulata)

SQL

Procedura alimentare cont

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

mysql> CALL ALIMENTARE_CONT(10001,2000);
+-----+
| @ID_D := id |
+-----+
|          1 |
+-----+
1 row in set (0.06 sec)

+-----+
| @VAL := valoare |
+-----+
|          1780 |
+-----+
1 row in set (0.06 sec)

+-----+
| @ID_TIP := id |
+-----+
|          4 |
+-----+
1 row in set (0.13 sec)

Query OK, 0 rows affected (0.16 sec)

mysql> CALL ISTORIC_TRANZACTII(10001, CURRENT_DATE(), CURRENT_DATE(),1);
+-----+-----+-----+-----+-----+-----+-----+
| Sursa | Destinatie | Valoare | Operatiune | Data | Timp | Detalii |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 0 | Deschidere cont | 2008-12-14 | 12:07:47 | NULL |
| NULL | 1 | 2000 | Depunere ghiseu | 2008-12-14 | 12:07:59 | Depunere ghiseu |
| 1 | 8 | 120 | Transfer intre conturi | 2008-12-14 | 12:08:07 | Plata utilitati |
| 1 | 2 | 100 | Transfer intre conturi | 2008-12-14 | 12:08:07 | Ramburs imprumut |
| 7 | 1 | 1000 | Transfer intre conturi | 2008-12-14 | 12:08:07 | Plata salariu |
| 1 | NULL | 500 | Extragere ATM | 2008-12-14 | 12:08:08 | Extragere ATM reusit |
| 1 | NULL | 500 | Extragere ATM | 2008-12-14 | 12:08:08 | Extragere ATM reusit |
| 1 | NULL | 0 | TR_ERR: cont creditor insuficient | 2008-12-14 | 12:08:08 | Extragere ATM esuata |
| NULL | 1 | 2000 | Depunere ghiseu | 2008-12-14 | 19:56:05 | Depunere ghiseu |
+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.06 sec)
```

Procedura se aplica contului 10001 pentru a se alimenta contul cu 2000 lei; este un cont asupra caruia s-au efectuat deja mai multe operatiuni (operatiunea curenta este ultima listata). In tabela de tranzactii apare inregistrata depunerea numerar la 19:56. 14 Pentru listarea tranzactiilor, se utilizeaza procedura istoric_tranzactii (vezi mai jos).

Procedura transfer intre conturi

```
# Transfer din cont in cont
DROP PROCEDURE IF EXISTS TRANSFER_INTRE_CONTURI;

DELIMITER //

CREATE PROCEDURE
    TRANSFER_INTRE_CONTURI(cont_sursa_cod int, cont_destinatie_cod int,
        valoare_transfer float, detaliu_transfer char(20))

BEGIN
    START TRANSACTION;
    SELECT @ID_S := id FROM cont WHERE cod = cont_sursa_cod;
    SELECT @ID_D := id FROM cont WHERE cod = cont_destinatie_cod;
    SELECT @VAL := valoare FROM cont WHERE id = @ID_S;
    SELECT @ID_TIP1 := id FROM tip_operatiune WHERE nume = 'Transfer intre conturi';
    SELECT @ID_TIP2 := id FROM tip_operatiune WHERE nume = 'TR_ERR: cont creditor insuficient';

    IF (@VAL < valoare_transfer) THEN
        BEGIN
            INSERT INTO tranzactie_cont
                (cont_sursa_id, cont_destinatie_id, valoare, tip, data, timp,detaliu)
            VALUES
                (@ID_S, @ID_D, 0, @ID_TIP2, CURRENT_DATE(), NOW(),detaliu_transfer);
            COMMIT;
        END;
    ELSEIF (@ID_S IS NOT NULL AND @ID_D IS NOT NULL) THEN
        BEGIN
            INSERT INTO tranzactie_cont
                (cont_sursa_id, cont_destinatie_id, valoare, tip, data, timp,detaliu)
            VALUES
                (@ID_S, @ID_D, valoare_transfer, @ID_TIP1, CURRENT_DATE(), NOW(),detaliu_transfer);

            UPDATE cont SET valoare = valoare - valoare_transfer WHERE id = @ID_S;

            UPDATE cont SET valoare = valoare + valoare_transfer WHERE id = @ID_D;

            COMMIT;
        END;
    ELSE ROLLBACK;
    END IF;

END //
DELIMITER ;
```

SQL

Procedura transfer intre conturi (tranzactie finalizata, transfer reusit)

```
mysql> CALL TRANSFER_INTRE_CONTURI(10001,40004,450,'Cumparaturi diverse');
```

```
+-----+
| @ID_S := id |
+-----+
|           1 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @ID_D := id |
+-----+
|           4 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @VAL := valoare |
+-----+
|           3780 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @ID_TIP1 := id |
+-----+
|             3 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @ID_TIP2 := id |
+-----+
|            10 |
+-----+
1 row in set (0.00 sec)
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> CALL ISTORIC_TRANZACTII(10001, CURRENT_DATE(), CURRENT_DATE(),1);
```

Sursa	Destinatie	Valoare	Operatiune	Data	Timp	Detaliu
1	1	0	Deschidere cont	2008-12-14	12:07:47	NULL
NULL	1	2000	Depunere ghiseu	2008-12-14	12:07:59	Depunere ghiseu
1	8	120	Transfer intre conturi	2008-12-14	12:08:07	Plata utilitati
1	2	100	Transfer intre conturi	2008-12-14	12:08:07	Ramburs imprumut
7	1	1000	Transfer intre conturi	2008-12-14	12:08:07	Plata salariu
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	12:08:08	Extragere ATM esuata
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
NULL	1	2000	Depunere ghiseu	2008-12-14	19:56:05	Depunere ghiseu
1	4	450	Transfer intre conturi	2008-12-14	20:30:32	Cumparaturi diverse

SQL

Procedura transfer intre conturi (tranzactie finalizata, transfer esuat)

```
mysql> CALL TRANSFER_INTRE_CONTURI(10001,40004,4500,'Plata imprumut');
```

```
+-----+
| @ID_S := id |
+-----+
|           1 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @ID_D := id |
+-----+
|           4 |
+-----+
1 row in set (0.01 sec)
```

```
+-----+
| @VAL := valoare |
+-----+
|           3330 |
+-----+
1 row in set (0.01 sec)
```

```
+-----+
| @ID_TIP1 := id |
+-----+
|             3 |
+-----+
1 row in set (0.01 sec)
```

```
+-----+
| @ID_TIP2 := id |
+-----+
|             10 |
+-----+
1 row in set (0.01 sec)
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> CALL ISTORIC_TRANZACTII(10001, CURRENT_DATE(), CURRENT_DATE(),1);
```

Sursa	Destinatie	Valoare	Operatiune	Data	Timp	Detaliu
1	1	0	Deschidere cont	2008-12-14	12:07:47	NULL
NULL	1	2000	Depunere ghiseu	2008-12-14	12:07:59	Depunere ghiseu
1	8	120	Transfer intre conturi	2008-12-14	12:08:07	Plata utilitati
1	2	100	Transfer intre conturi	2008-12-14	12:08:07	Ramburs imprumut
7	1	1000	Transfer intre conturi	2008-12-14	12:08:07	Plata salariu
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	12:08:08	Extragere ATM esuata
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
NULL	1	2000	Depunere ghiseu	2008-12-14	19:56:05	Depunere ghiseu
1	4	450	Transfer intre conturi	2008-12-14	20:30:32	Cumparaturi diverse
1	4	0	TR_ERR: cont creditor insuficient	2008-12-14	20:34:12	Plata imprumut

Procedura transfer intre conturi (tranzactie anulata – cont inexistent, transfer esuat)

```

c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> CALL TRANSFER_INTRE_CONTURI(100001,40004,450,'Plata imprumut');
Empty set (0.00 sec)

+-----+
| @ID_D := id |
+-----+
|          4 |
+-----+
1 row in set (0.00 sec)

Empty set (0.02 sec)

+-----+
| @ID_TIP1 := id |
+-----+
|          3 |
+-----+
1 row in set (0.02 sec)

+-----+
| @ID_TIP2 := id |
+-----+
|         10 |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

mysql> CALL ISTORIC_TRANZACTII(10001, CURRENT_DATE(), CURRENT_DATE(),1);
+-----+-----+-----+-----+-----+-----+-----+
| Sursa | Destinatie | Valoare | Operatiune | Data | Timp | Detaliu |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 0 | Deschidere cont | 2008-12-14 | 12:07:47 | NULL |
| NULL | 1 | 2000 | Depunere ghiseu | 2008-12-14 | 12:07:59 | Depunere ghiseu |
| 1 | 2 | 100 | Transfer intre conturi | 2008-12-14 | 12:08:07 | Ramburs imprumut |
| 7 | 1 | 1000 | Transfer intre conturi | 2008-12-14 | 12:08:07 | Plata salariu |
| 1 | 8 | 120 | Transfer intre conturi | 2008-12-14 | 12:08:07 | Plata utilitati |
| 1 | NULL | 500 | Extragere ATM | 2008-12-14 | 12:08:08 | Extragere ATM reusit |
| 1 | NULL | 500 | Extragere ATM | 2008-12-14 | 12:08:08 | Extragere ATM reusit |
| 1 | NULL | 0 | TR_ERR: cont creditor insuficient | 2008-12-14 | 12:08:08 | Extragere ATM esuata |
| NULL | 1 | 2000 | Depunere ghiseu | 2008-12-14 | 19:56:05 | Depunere ghiseu |
| 1 | 4 | 450 | Transfer intre conturi | 2008-12-14 | 20:30:32 | Cumparaturi diverse |
| 1 | 4 | 0 | TR_ERR: cont creditor insuficient | 2008-12-14 | 20:34:12 | Plata imprumut |
+-----+-----+-----+-----+-----+-----+-----+

```

Procedura extragere ATM

```
# Extragere de la ATM
DROP PROCEDURE IF EXISTS EXTRAGERE_ATM;

DELIMITER //

CREATE PROCEDURE EXTRAGERE_ATM(cont_sursa_cod int, valoare_extragere float)

BEGIN
  START TRANSACTION;
  SET @ID_S = NULL, @VAL = NULL, @ID_TIP1 = NULL, @ID_TIP2 = NULL;
  SELECT @ID_S := id FROM cont WHERE cod = cont_sursa_cod;
  SELECT @VAL := valoare FROM cont WHERE id = @ID_S;
  SELECT @ID_TIP1 := id FROM tip_operatiune WHERE nume = 'Extragere ATM';
  SELECT @ID_TIP2 := id FROM tip_operatiune WHERE nume = 'TR_ERR: cont creditor insuficient';

  IF (@ID_S IS NOT NULL AND @VAL < valoare_extragere) THEN
    BEGIN
      INSERT INTO tranzactie_cont
        (cont_sursa_id, cont_destinatie_id, valoare, tip, data, timp,detaliu)
      VALUES
        (@ID_S, NULL, 0, @ID_TIP2, CURRENT_DATE(), NOW(), 'Extragere ATM esuata');
      COMMIT;
    END;
  ELSEIF (@ID_S IS NOT NULL) THEN
    BEGIN
      INSERT INTO tranzactie_cont
        (cont_sursa_id, cont_destinatie_id, valoare, tip, data, timp,detaliu)
      VALUES
        (@ID_S, NULL, valoare_extragere, @ID_TIP1, CURRENT_DATE(), NOW(), 'Extragere ATM reusita');

      UPDATE cont SET valoare = valoare - valoare_extragere WHERE id = @ID_S;

      COMMIT;
    END;
  ELSE ROLLBACK;
  END IF;

END //
DELIMITER ;
```

Procedura primeste ca parametru codul contului debitat si valoarea extragerii; tranzactia este fie finalizata, cu transferul efectuat (contul exista si suma exista in cont), fie tranzactia este finalizata iar transferul esuat (cont existent dar suma insuficienta), fie tranzactia nu este finalizata (cont inexistent, se face rollback).

SQL

Procedura extragere ATM (tranzactie finalizata, transfer reusit)

```
mysql> CALL EXTRAGERE_ATM(10001, 500);
```

```
+-----+
| @ID_S := id |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @VAL := valoare |
+-----+
|          2880 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @ID_TIP1 := id |
+-----+
|           7 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @ID_TIP2 := id |
+-----+
|          10 |
+-----+
1 row in set (0.02 sec)
```

Query OK, 0 rows affected, 1 warning (0.02 sec)

```
mysql> CALL ISTORIC_TRANZACTII(10001, CURRENT_DATE(), CURRENT_DATE(),0);
```

Sursa	Destinatie	Valoare	Operatiune	Data	Timp	Detaliiu
1	1	0	Deschidere cont	2008-12-14	12:07:47	NULL
NULL	1	2000	Depunere ghiseu	2008-12-14	12:07:59	Depunere ghiseu
7	1	1000	Transfer intre conturi	2008-12-14	12:08:07	Plata salariu
1	8	120	Transfer intre conturi	2008-12-14	12:08:07	Plata utilitati
1	2	100	Transfer intre conturi	2008-12-14	12:08:07	Ramburs imprumut
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	12:08:08	Extragere ATM esuata
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
NULL	1	2000	Depunere ghiseu	2008-12-14	19:56:05	Depunere ghiseu
1	4	450	Transfer intre conturi	2008-12-14	20:30:32	Cumparaturi diverse
1	4	0	TR_ERR: cont creditor insuficient	2008-12-14	20:34:12	Plata imprumut
1	NULL	500	Extragere ATM	2008-12-14	21:01:37	Extragere ATM reusit

SQL

Procedura extragere ATM (tranzactie finalizata, transfer esuat – suma solicitata nu este disponibila)

```
mysql> CALL EXTRAGERE_ATM(10001, 30000);
```

```
+-----+
| @ID_S := id |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @VAL := valoare |
+-----+
|          2380 |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| @ID_TIP1 := id |
+-----+
|           7 |
+-----+
1 row in set (0.01 sec)
```

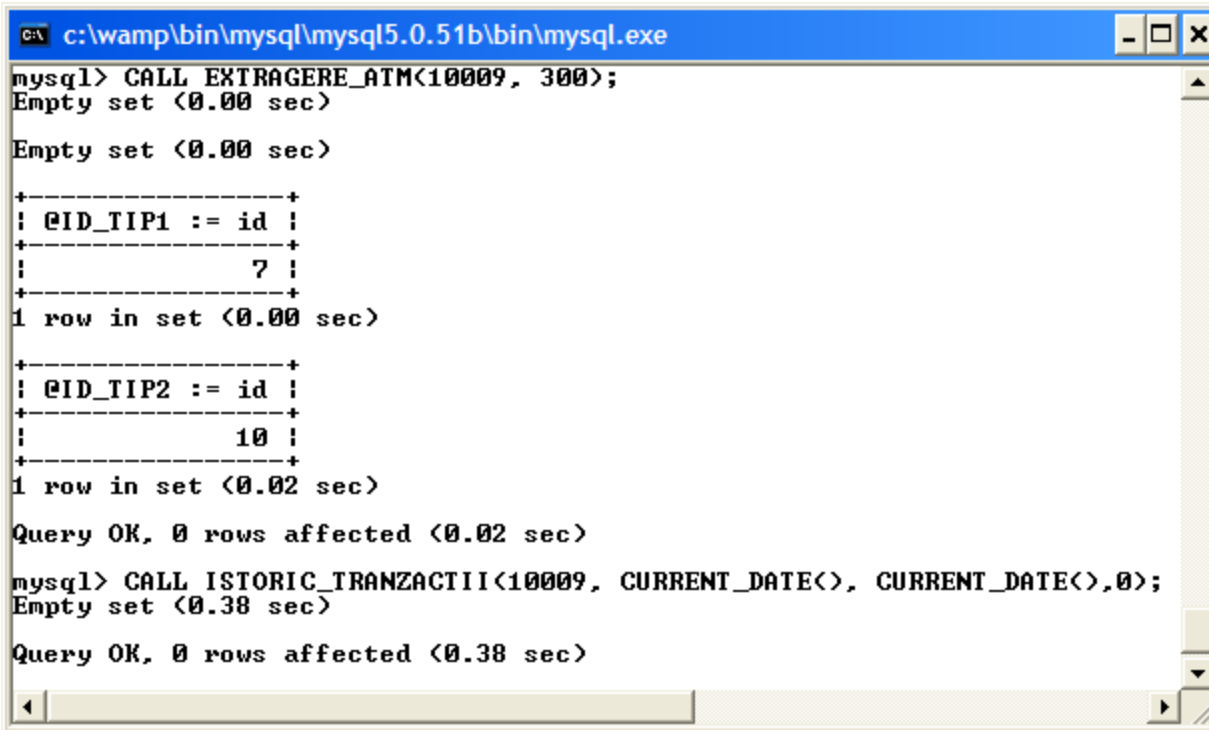
```
+-----+
| @ID_TIP2 := id |
+-----+
|          10 |
+-----+
1 row in set (0.01 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CALL ISTORIC_TRANZACTII(10001, CURRENT_DATE(), CURRENT_DATE(),0);
```

Sursa	Destinatie	Valoare	Operatiune	Data	Timp	Detaliu
1	1	0	Deschidere cont	2008-12-14	12:07:47	NULL
NULL	1	2000	Depunere ghiseu	2008-12-14	12:07:59	Depunere ghiseu
7	1	1000	Transfer intre conturi	2008-12-14	12:08:07	Plata salariu
1	8	120	Transfer intre conturi	2008-12-14	12:08:07	Plata utilitati
1	2	100	Transfer intre conturi	2008-12-14	12:08:07	Ramburs imprumut
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	12:08:08	Extragere ATM esuata
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
NULL	1	2000	Depunere ghiseu	2008-12-14	19:56:05	Depunere ghiseu
1	4	450	Transfer intre conturi	2008-12-14	20:30:32	Cumparaturi diverse
1	4	0	TR_ERR: cont creditor insuficient	2008-12-14	20:34:12	Plata imprumut
1	NULL	500	Extragere ATM	2008-12-14	21:01:37	Extragere ATM reusit
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	21:04:32	Extragere ATM esuata

Procedura extragere ATM (tranzactie nefinalizata – cont inexistent)



```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> CALL EXTRAGERE_ATM<10009, 300>;
Empty set <0.00 sec>

Empty set <0.00 sec>

+-----+
| @ID_TIP1 := id |
+-----+
|                7 |
+-----+
1 row in set <0.00 sec>

+-----+
| @ID_TIP2 := id |
+-----+
|                10 |
+-----+
1 row in set <0.02 sec>

Query OK, 0 rows affected <0.02 sec>

mysql> CALL ISTORIC_TRANZACTII<10009, CURRENT_DATE(), CURRENT_DATE(),0>;
Empty set <0.38 sec>

Query OK, 0 rows affected <0.38 sec>
```

Procedura istoric tranzactii

```
# Procedure istoric tranzactii
DROP PROCEDURE IF EXISTS ISTORIC_TRANZACTII;

DELIMITER //

CREATE PROCEDURE ISTORIC_TRANZACTII(cod_cont int, data_initiala date, data_finala date, stocheaza int)

BEGIN
    IF(stocheaza = 1) THEN
        BEGIN
            START TRANSACTION;
            INSERT INTO tranzactie_cont
            (tip, data, timp,detaliu)
            VALUES
            ((SELECT id from tip_operatiune WHERE nume = 'Istoric tranzactii'),
            CURRENT_DATE(), NOW(), 'Consultare tranzactii');
            COMMIT;
        END;
    END IF;

    SELECT cont_sursa_id AS Sursa, cont_destinatie_id AS Destinatie,
    tranzactie_cont.valoare AS Valoare, tip_operatiune.nume AS Operatiune, data AS Data, timp AS Timp,
    detaliu as Detaliu
    FROM tranzactie_cont, tip_operatiune, cont, client
    WHERE tranzactie_cont.tip = tip_operatiune.id
    AND ( cont.id = cont_sursa_id OR cont.id = cont_destinatie_id)
    AND client.id = cont.id
    AND cont.cod = cod_cont
    AND data BETWEEN data_initiala AND data_finala
    ORDER BY Data, Timp;

END //
DELIMITER ;
```

Procedura primește ca parametrii codul contului debitat, data initiala, data finala și opțiunea pentru a stoca operațiunea curentă (vizualizare istoric tranzactii) în tabela de tranzactii; opțiunea este necesară pentru că e posibil ca această operațiune să fie taxată în anumite condiții și atunci trebuie să fie înregistrată

SQL

Procedura istoric tranzactii – listare completa tranzactii cont 10001 in ziua curenta si cu inregistrarea in tabela de tranzactii a tranzactiei curente

```
mysql> CALL ISTORIC_TRANZACTII(10001, CURRENT_DATE(), CURRENT_DATE(),1);
```

Sursa	Destinatie	Valoare	Operatiune	Data	Timp	Detaliu
1	1	0	Deschidere cont	2008-12-14	12:07:47	NULL
NULL	1	2000	Depunere ghiseu	2008-12-14	12:07:59	Depunere ghiseu
7	1	1000	Transfer intre conturi	2008-12-14	12:08:07	Plata salariu
1	8	120	Transfer intre conturi	2008-12-14	12:08:07	Plata utilitati
1	2	100	Transfer intre conturi	2008-12-14	12:08:07	Ramburs imprumut
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	12:08:08	Extragere ATM esuata
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
NULL	1	2000	Depunere ghiseu	2008-12-14	19:56:05	Depunere ghiseu
1	4	450	Transfer intre conturi	2008-12-14	20:30:32	Cumparaturi diverse
1	4	0	TR_ERR: cont creditor insuficient	2008-12-14	20:34:12	Plata imprumut
1	NULL	500	Extragere ATM	2008-12-14	21:01:37	Extragere ATM reusit
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	21:04:32	Extragere ATM esuata
1	NULL	300	Extragere ATM	2008-12-14	21:07:52	Extragere ATM reusit

```
14 rows in set (0.01 sec)
```

```
Query OK, 0 rows affected, 1 warning (0.06 sec)
```

... si rezultatul complet este ...

```
mysql> select * from tranzactie_cont;
```

id	cont_sursa_id	cont_destinatie_id	tip	data	timp	valoare	detaliu
1	1	1	1	2008-12-14	12:07:47	0	NULL
2	2	2	1	2008-12-14	12:07:47	0	NULL
3	3	3	1	2008-12-14	12:07:47	0	NULL
4	4	4	1	2008-12-14	12:07:47	0	NULL
5	5	5	1	2008-12-14	12:07:47	0	NULL
6	6	6	1	2008-12-14	12:07:47	0	NULL
7	7	7	1	2008-12-14	12:07:47	0	NULL
42	NULL	NULL	9	2008-12-14	20:55:56	NULL	Consultare tranzacti
43	1	NULL	7	2008-12-14	21:01:37	500	Extragere ATM reusit
44	1	NULL	10	2008-12-14	21:04:32	0	Extragere ATM esuata
45	1	NULL	7	2008-12-14	21:07:52	300	Extragere ATM reusit
46	NULL	NULL	9	2008-12-14	21:20:49	NULL	Consultare tranzacti

```
45 rows in set (0.00 sec)
```


Procedura balanta cont

```
# Balanta cont (la o data anume)
DROP PROCEDURE IF EXISTS BALANTIA_CONT;

DELIMITER //

CREATE PROCEDURE BALANTIA_CONT(cont_cod int, data_balanta date, ora_balanta time)

BEGIN
    SET @ID = NULL, @ID_TIP = NULL;
    SELECT @ID := id FROM cont WHERE cod = cont_cod;
    SELECT @ID_TIP := id FROM tip_operatiune WHERE nume = 'Balanta cont';

    IF(@ID IS NOT NULL) THEN
        BEGIN
            SELECT @Credit:= SUM(valoare)
            FROM tranzactie_cont
            WHERE (data < data_balanta OR (data = data_balanta AND timp <= ora_balanta))
            GROUP BY cont_destinatie_id
            HAVING tranzactie_cont.cont_destinatie_id = @ID;

            SELECT @Debit:= SUM(valoare)
            FROM tranzactie_cont
            WHERE (data < data_balanta OR (data = data_balanta AND timp <= ora_balanta))
            GROUP BY cont_sursa_id
            HAVING tranzactie_cont.cont_sursa_id = @ID;

            SELECT cont_cod AS Cont, @Credit - @Debit AS Balanta,
            data_balanta AS Data, ora_balanta As Ora;

        END;
    END IF;
END //
DELIMITER ;
```

SQL

Procedura balanta cont (data, ora curenta)

```

c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> CALL BALANTA_CONT(10001, CURRENT_DATE(), NOW());
+-----+
| @ID := id |
+-----+
|          1 |
+-----+
1 row in set (0.02 sec)

+-----+
| @ID_TIP := id |
+-----+
|              8 |
+-----+
1 row in set (0.02 sec)

+-----+
| @Credit:= SUM(valoare) |
+-----+
|                    5000 |
+-----+
1 row in set (0.02 sec)

+-----+
| @Debit:= SUM(valoare) |
+-----+
|                    2470 |
+-----+
1 row in set (0.02 sec)

+-----+
| Cont | Balanta | Data | Ora |
+-----+
| 10001 | 2530 | 2008-12-14 | 21:52:12 |
+-----+
1 row in set (0.02 sec)
  
```

(data curenta, ora anterioara)

```

c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> CALL BALANTA_CONT(10001, CURRENT_DATE(), '12:09');
+-----+
| @ID := id |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)

+-----+
| @ID_TIP := id |
+-----+
|              8 |
+-----+
1 row in set (0.00 sec)

+-----+
| @Credit:= SUM(valoare) |
+-----+
|                    3000 |
+-----+
1 row in set (0.00 sec)

+-----+
| @Debit:= SUM(valoare) |
+-----+
|                    1220 |
+-----+
1 row in set (0.00 sec)

+-----+
| Cont | Balanta | Data | Ora |
+-----+
| 10001 | 1780 | 2008-12-14 | 12:09:00 |
+-----+
1 row in set (0.00 sec)
  
```

Sursa	Destinatie	Valoare	Operatiune	Data	Timp	Detaliu
1	1	0	Deschidere cont	2008-12-14	12:07:47	NULL
NULL	1	2000	Depunere ghiseu	2008-12-14	12:07:59	Depunere ghiseu
7	1	1000	Transfer intre conturi	2008-12-14	12:08:07	Plata salariu
1	8	120	Transfer intre conturi	2008-12-14	12:08:07	Plata utilitati
1	2	100	Transfer intre conturi	2008-12-14	12:08:07	Ramburs imprumut
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	12:08:08	Extragere ATM esuata
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
1	NULL	500	Extragere ATM	2008-12-14	12:08:08	Extragere ATM reusit
NULL	1	2000	Depunere ghiseu	2008-12-14	19:56:05	Depunere ghiseu
1	4	450	Transfer intre conturi	2008-12-14	20:30:32	Cumparaturi diverse
1	4	0	TR_ERR: cont creditor insuficient	2008-12-14	20:34:12	Plata imprumut
1	NULL	500	Extragere ATM	2008-12-14	21:01:37	Extragere ATM reusit
1	NULL	0	TR_ERR: cont creditor insuficient	2008-12-14	21:04:32	Extragere ATM esuata
1	NULL	300	Extragere ATM	2008-12-14	21:07:52	Extragere ATM reusit