Proiect final la SGDB
Baza de date a unei burse de valori

Mihail Feraru
mihail.feraru@s.unibuc.ro

Prezentare generala

Tema alease este un sistem de gestiune al unei burse de valori, deoarece sunt interesat de sistemele financiare si functionarea lor. Bursele de valori joaca un rol cheie in econimia moderna, iar dependenta lor fata de tehnologie este evidenta luand in considerare fluxul exorbitant de tranzactii ce se petrec in fiecare moment. Am descis astfel sa implementez o baza de date ce permite administrarea unui sistem de acest tip, avand in vedere urmatoarele functionalitati:

1. stocarea diferitelor instrumente financiare
2. licitarea si tranzactionarea de actiuni
3. transferul de captial
4. plata taxelor catre facilitatorul bursei
5. inregistrarea amanuntita a operatiunilor

In centrul unei burse de valori se afla instrumentele financiare, acestea reprezinta orice activ tranzactionabil in unitati distincte. Instrumentele financiare vor fi stocate in tabela **SECURITY**, fiecare fiind caracterizat de un identificator unic numit *ticker*. Deoarece acestea pot fi de mai multe tipuri am simulat o relatie de mostenire folosind tehnica *Class Table Inheritance (Table Per Type Inheritance)*. Cateva exemple fiind relatia intre **SECURITY** si **STOCK_SECURITY** (actiuni comune), **FUND_SECURITY** (fonduri de investitii), **BOND_SECURITY** (obligatiuni).

Cum instrumentele financiare nu sunt utile daca nu au detintatori, baza de date utilizeaza tabela de relatie **OWN** care face legatura intre **SECURITY** si **ACCOUNT** pentru a retine cantitatea de parti de instrument detinute de fiecare utilizator al sistemului. Detalii suplimentare despre utilizatori se pot gasi in tabela **HOLDER**.

Pentru oferte si tranzactii, se pun la dispozitie tabele **QUOTATION** si **TRADE** care sunt menite sa pastreze istoricul operatiunilor cu actiuni si capital.

Folosindu-ma de functionalitatile PL/SQL am implementat sub forma de proceduri si functii operatiunile de baza ale bursei, mentionate mai sus. Codul sursa este separat in trei pachete[1]:

1. **SE_TYPES** - contine tipuri de date care se regasez in mai multe tabele, definite pentru a imbunatati lizibilitatea logicii procedurilor
2. **SE_UTILS** - utilitare generice precum functii matematice sau tratare de exceptii
3. **SE_CORE** - functionalitatea de baza, contine produceduri pentru operatiuni precum licitarea si tranzactionarea actiunilor, calcularea taxelor, etc.

In pagina urmatoare avem diagrama conceptuala si cea entitate-relatie a bazei de date pentru a ne face o mai buna imagine asupra functionarii acesteia.

Codul sursa complet se poate gasi la: `https://github.com/JustBeYou/SGDB/tree/master/proiect`

[1]prefixul **SE** din numele pachetelor este prescurtarea pentru *Stock Exchange*

Diagrama entitate-relatie

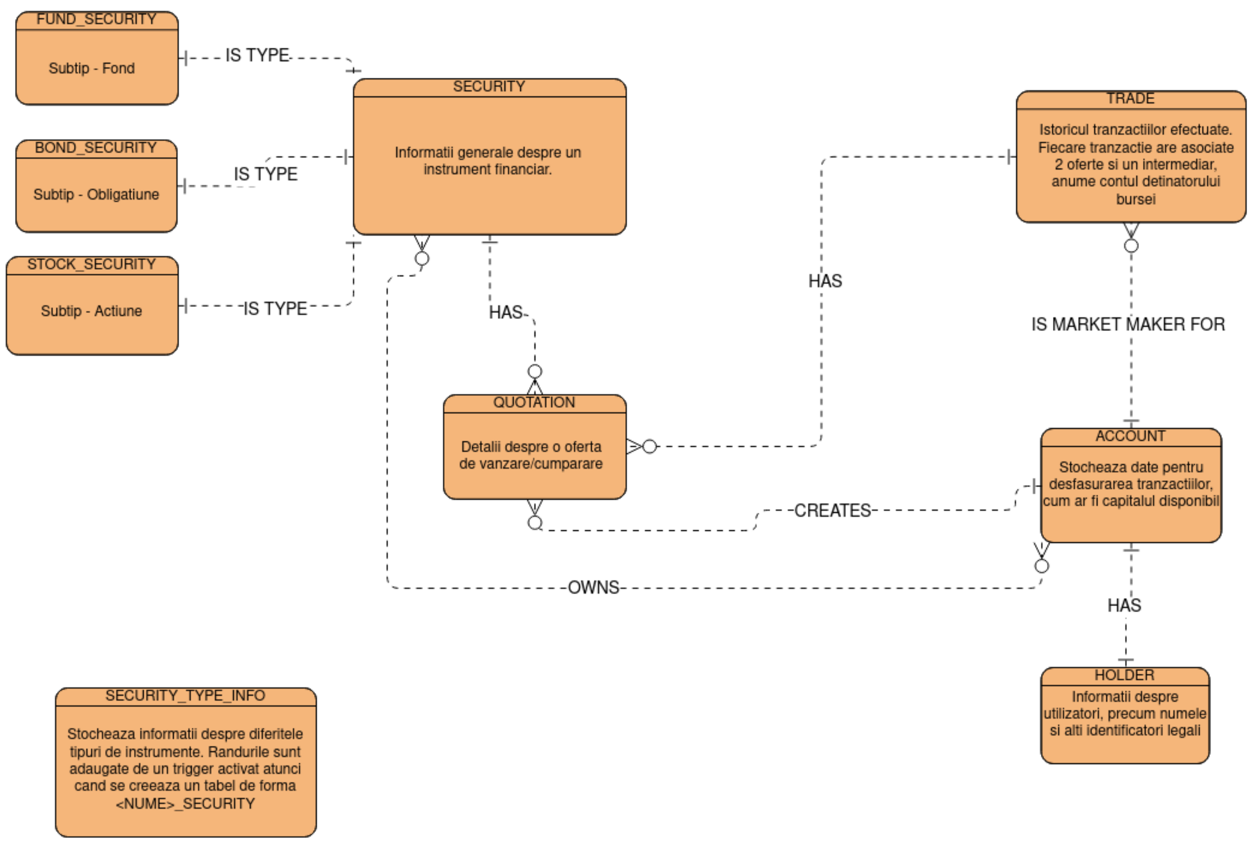
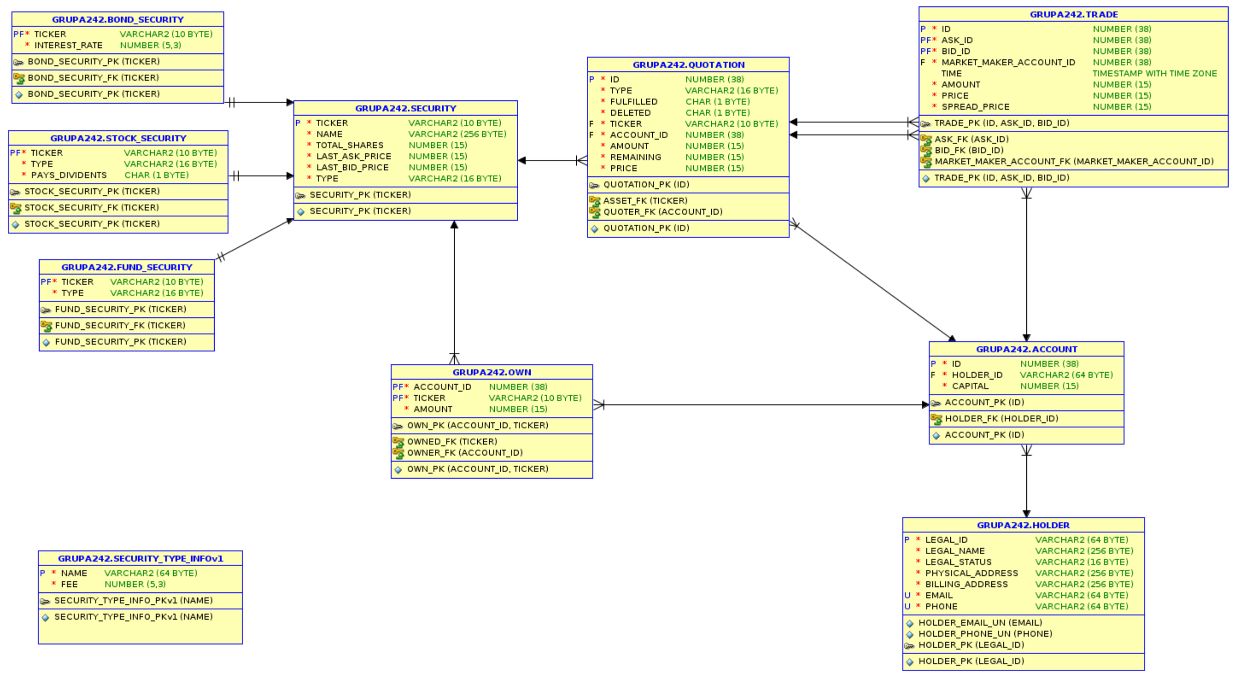

FUND_SECURITY
Subtip - Fond
IS TYPE
BOND_SECURITY
Subtip - Obligatiune
IS TYPE
STOCK_SECURITY
Subtip - Actiune
IS TYPE
SECURITY
Informatii generale despre un instrument financiar.
HAS
QUOTATION
Detalii despre o oferta de vanzare/cumparare
HAS
TRADE
Istoricul tranzactiilor efectuate. Fiecare tranzactie are asociate 2 oferte si un intermediar, anume contul detinatorului bursei
IS MARKET MAKER FOR
ACCOUNT
Stocheaza date pentru desfasurarea tranzactiilor, cum ar fi capitalul disponibil
CREATES
OWNS
HAS
HOLDER
Informatii despre utilizatori, precum numele si alti identificatori legali
SECURITY_TYPE_INFO
Stocheaza informatii despre diferitele tipuri de instrumente. Randurile sunt adaugate de un trigger activat atunci cand se creeaza un tabel de forma <NUME>_SECURITY

Diagrama conceptuala

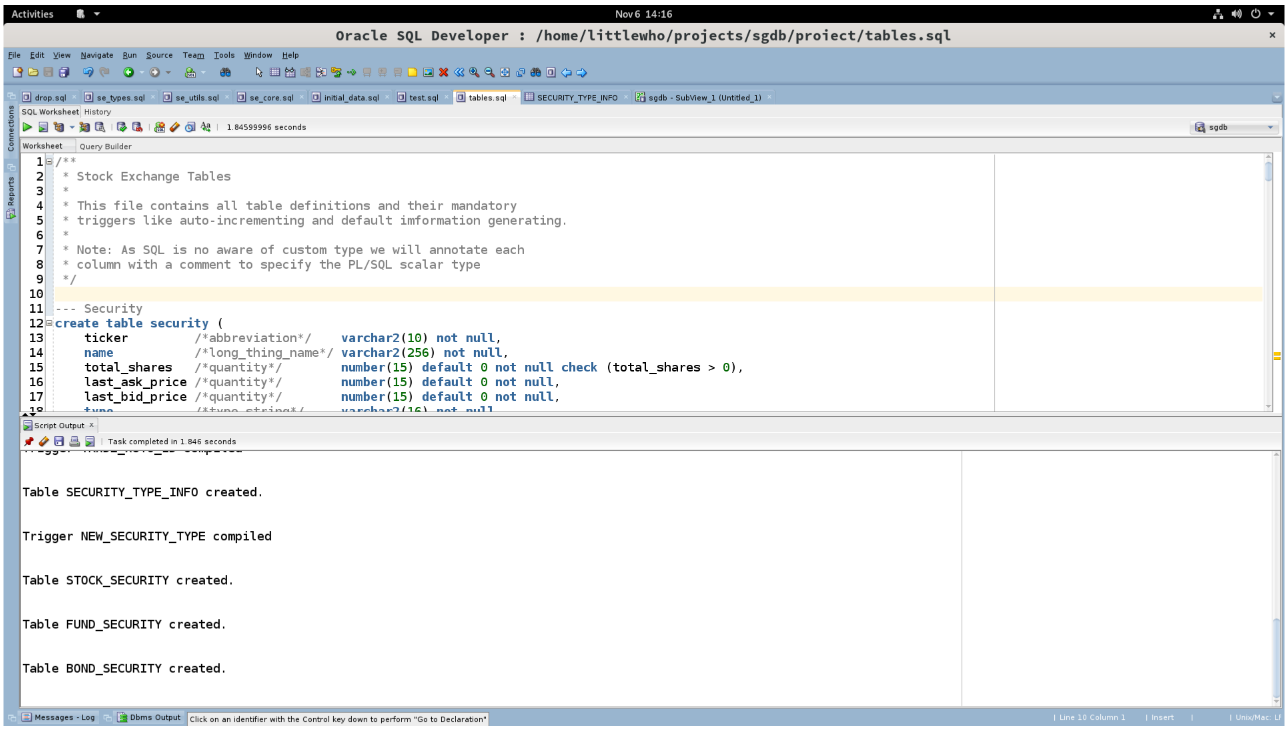

GRUPA242.BOND_SECURITY
PF* TICKER VARCHAR2 (10 BYTE)
* INTEREST_RATE NUMBER (5,3)
BOND_SECURITY_PK (TICKER)
BOND_SECURITY_FK (TICKER)
BOND_SECURITY_PK (TICKER)
GRUPA242.STOCK_SECURITY
PF* TICKER VARCHAR2 (10 BYTE)
* TYPE VARCHAR2 (16 BYTE)
* PAYS_DIVIDENTS CHAR (1 BYTE)
STOCK_SECURITY_PK (TICKER)
STOCK_SECURITY_FK (TICKER)
STOCK_SECURITY_PK (TICKER)
GRUPA242.FUND_SECURITY
PF* TICKER VARCHAR2 (10 BYTE)
* TYPE VARCHAR2 (16 BYTE)
FUND_SECURITY_PK (TICKER)
FUND_SECURITY_FK (TICKER)
FUND_SECURITY_PK (TICKER)
GRUPA242.SECURITY
P * TICKER VARCHAR2 (10 BYTE)
* NAME VARCHAR2 (256 BYTE)
* TOTAL_SHARES NUMBER (15)
* LAST_ASK_PRICE NUMBER (15)
* LAST_BID_PRICE NUMBER (15)
* TYPE VARCHAR2 (16 BYTE)
SECURITY_PK (TICKER)
SECURITY_PK (TICKER)
GRUPA242.QUOTATION
P * ID NUMBER (38)
* TYPE VARCHAR2 (16 BYTE)
* FULFILLED CHAR (1 BYTE)
* DELETED CHAR (1 BYTE)
F * TICKER VARCHAR2 (10 BYTE)
F * ACCOUNT_ID NUMBER (38)
* AMOUNT NUMBER (15)
* REMAINING NUMBER (15)
* PRICE NUMBER (15)
QUOTATION_PK (ID)
ASSET_FK (TICKER)
QUOTER_FK (ACCOUNT_ID)
QUOTATION_PK (ID)
GRUPA242.TRADE
P * ID NUMBER (38)
PF* ASK_ID NUMBER (38)
PF* BID_ID NUMBER (38)
F * MARKET_MAKER_ACCOUNT_ID NUMBER (38)
TIME TIMESTAMP WITH TIME ZONE
* AMOUNT NUMBER (15)
* PRICE NUMBER (15)
* SPREAD_PRICE NUMBER (15)
TRADE_PK (ID, ASK_ID, BID_ID)
ASK_FK (ASK_ID)
BID_FK (BID_ID)
MARKET_MAKER_ACCOUNT_FK (MARKET_MAKER_ACCOUNT_ID)
TRADE_PK (ID, ASK_ID, BID_ID)
GRUPA242.OWN
PF* ACCOUNT_ID NUMBER (38)
PF* TICKER VARCHAR2 (10 BYTE)
* AMOUNT NUMBER (15)
OWN_PK (ACCOUNT_ID, TICKER)
OWNED_FK (TICKER)
OWNER_FK (ACCOUNT_ID)
OWN_PK (ACCOUNT_ID, TICKER)
GRUPA242.ACCOUNT
P * ID NUMBER (38)
F * HOLDER_ID VARCHAR2 (64 BYTE)
* CAPITAL NUMBER (15)
ACCOUNT_PK (ID)
HOLDER_FK (HOLDER_ID)
ACCOUNT_PK (ID)
GRUPA242.HOLDER
P * LEGAL_ID VARCHAR2 (64 BYTE)
* LEGAL_NAME VARCHAR2 (256 BYTE)
* LEGAL_STATUS VARCHAR2 (16 BYTE)
* PHYSICAL_ADDRESS VARCHAR2 (256 BYTE)
* BILLING_ADDRESS VARCHAR2 (256 BYTE)
U * EMAIL VARCHAR2 (64 BYTE)
U * PHONE VARCHAR2 (64 BYTE)
HOLDER_EMAIL_UN (EMAIL)
HOLDER_PHONE_UN (PHONE)
HOLDER_PK (LEGAL_ID)
HOLDER_PK (LEGAL_ID)
GRUPA242.SECURITY_TYPE_INFOv1
P * NAME VARCHAR2 (64 BYTE)
* FEE NUMBER (5,3)
SECURITY_TYPE_INFO_PKv1 (NAME)
SECURITY_TYPE_INFO_PKv1 (NAME)

Rezolvarea cerintelor obligatorii

Cerinta 4

```
/**
 * Stock Exchange Tables
 *
 * This file contains all table definitions and their mandatory
 * triggers like auto-incrementing and default information generating.
 *
 * Note: As SQL is no aware of custom type we will annotate each
 * column with a comment to specify the PL/SQL scalar type
 */

--- Security
create table security (
    ticker          /*abbreviation*/    varchar2(10) not null,
    name            /*long_thing_name*/ varchar2(256) not null,
    total_shares    /*quantity*/        number(15) default 0 not null check (total_shares > 0),
    last_ask_price  /*quantity*/        number(15) default 0 not null,
    last_bid_price  /*quantity*/        number(15) default 0 not null,
    type            /*type_string*/     varchar2(16) not null,

    constraint security_pk primary key (ticker)
);

--- Holder
create table holder (
    -- social security number/registration number
    legal_id         /*thing_name*/      varchar2(64) not null,
    legal_name       /*long_thing_name*/ varchar2(256) not null,
    legal_status     /*type_string*/     varchar2(16) not null
        check (legal_status in ('PERSON', 'COMPANY')),
    physical_address /*long_thing_name*/ varchar2(256) not null,
    billing_address  /*long_thing_name*/ varchar2(256) not null,
    email            /*thing_name*/      varchar2(64) unique not null,
    phone            /*thing_name*/      varchar2(64) unique not null,

    constraint holder_pk primary key (legal_id)
);

--- Account
create table account (
    id        /*id*/         integer not null,
    holder_id /*thing_name*/ varchar2(64) not null,
    capital   /*quantity*/   number(15) default 0 not null,

    constraint account_pk primary key (id),
    constraint holder_fk
        foreign key (holder_id)
        references holder(legal_id)
);

create sequence account_ids start with 1;
create or replace trigger account_auto_id
before insert on account
for each row
```

```
begin
  select account_ids.nextval
  into   :new.id
  from   dual;
end;
/

--- account Owns security
create table own (
    account_id /*id*/           integer not null,
    ticker     /*abbreviation*/ varchar2(10) not null,
    amount     /*quantity*/     number(15) not null,

    constraint own_pk primary key (account_id, ticker),

    constraint owner_fk
        foreign key (account_id)
        references account(id),
    constraint owned_fk
        foreign key (ticker)
        references security(ticker)
);

--- Quotation
create table quotation (
    id /*id*/ integer not null,

    type      /*type_string*/ varchar2(16) not null
        check (type in ('ASK', 'BID')),
    fulfilled /*status*/      char default 0 not null
        check (fulfilled in (0, 1)),
    deleted   /*status*/      char default 0 not null
        check (deleted in (0, 1)),

    ticker     /*abbreviation*/ varchar2(10) not null,
    account_id /*id*/           integer not null,

    amount    /*quantity*/ number(15) not null,
    remaining /*quantity*/ number(15) not null,
    price     /*quantity*/ number(15) not null,

    constraint quotation_pk primary key (id),

    constraint quoter_fk
        foreign key (account_id)
        references account(id),
    constraint asset_fk
        foreign key (ticker)
        references security(ticker)
);

create sequence quotation_ids start with 1;
create or replace trigger quotation_auto_id
before insert on quotation
for each row
begin
  select quotation_ids.nextval
  into   :new.id
  from   dual;
end;
/

--- Trade
create table trade (
    id     /*id*/ integer not null,
    ask_id /*id*/ integer not null,
    bid_id /*id*/ integer not null,

    market_maker_account_id /*id*/ integer not null,
    time timestamp with time zone,

    amount       /*quantity*/     number(15) not null,
    price        /*quantity*/     number(15) not null,
    spread_price /*quantity*/     number(15) not null,

    constraint trade_pk primary key (id, ask_id, bid_id),

    constraint ask_fk
        foreign key (ask_id)
        references quotation(id),
    constraint bid_fk
        foreign key (bid_id)
        references quotation(id),

    constraint market_maker_account_fk
        foreign key (market_maker_account_id)
        references account(id)
);

create sequence trade_ids start with 1;
create or replace trigger trade_auto_id
before insert on trade
for each row
begin
  select trade_ids.nextval
  into   :new.id
  from   dual;
end;
/

--- Security Types

create table security_type_info (
    name /*thing_name*/ varchar2(64) not null,
    fee  /*percentage*/ number(5, 3) default 0.02 not null,

    constraint security_type_info_pk primary key (name)
);

CREATE TRIGGER new_security_type AFTER CREATE
    ON SCHEMA
    DECLARE
    BEGIN
        FOR ctable IN (
            SELECT a.table_name
            FROM all_tables a
            WHERE a.table_name LIKE '%_SECURITY'
             AND a.table_name NOT IN (
                    SELECT b.name as table_name from
                    security_type_info b
                )
        ) LOOP
            INSERT INTO security_type_info (name)
            VALUES (ctable.table_name);
        END LOOP;
    END;
/

create table stock_security (
    ticker /*abbreviation*/ varchar2(10) not null,
    constraint stock_security_pk primary key (ticker),
    constraint stock_security_fk
        foreign key (ticker)
        references security(ticker),

    type           /*type_string*/ varchar2(16) not null
        check (type in ('COMMON', 'PREFERRED')),
    pays_dividents /*status*/      char default 0 not null
        check (pays_dividents in (0, 1))
);

create table fund_security (
    ticker /*abbreviation*/ varchar2(10) not null,
    constraint fund_security_pk primary key (ticker),
    constraint fund_security_fk
        foreign key (ticker)
        references security(ticker),

    type /*type_string*/ varchar2(16) not null
        check (type in ('HEDGE', 'MUTUAL', 'BOND'))
);

create table bond_security (
    ticker /*abbreviation*/ varchar2(10) not null,
    constraint bond_security_pk primary key (ticker),
    constraint bond_security_fk
        foreign key (ticker)
        references security(ticker),

    interest_rate /*percentage*/ number(5, 3) not null
        check (interest_rate >= 0.01)
);
```


Activities
Nov 6 14:16
Oracle SQL Developer : /home/littlewho/projects/sgdb/proiect/tables.sql
File Edit View Navigate Run Source Team Tools Window Help
drop.sql
se_types.sql
se_utils.sql
se_core.sql
initial_data.sql
test.sql
tables.sql
SECURITY_TYPE_INFO
sgdb - SubView_1 (Untitled_1)
SQL Worksheet
History
1.84599996 seconds
sgdb
Worksheet
Query Builder
Script Output
Task completed in 1.846 seconds
Messages - Log
Dbms Output
Click on an identifier with the Control key down to perform "Go to Declaration"
Line 10 Column 1
Insert
Unix/Mac: LF

```
/**
 * Stock Exchange Tables
 *
 * This file contains all table definitions and their mandatory
 * triggers like auto-incrementing and default imformation generating.
 *
 * Note: As SQL is no aware of custom type we will annotate each
 * column with a comment to specify the PL/SQL scalar type
 */

--- Security
create table security (
    ticker         /*abbreviation*/    varchar2(10) not null,
    name           /*long_thing_name*/ varchar2(256) not null,
    total_shares   /*quantity*/        number(15) default 0 not null check (total_shares > 0),
    last_ask_price /*quantity*/        number(15) default 0 not null,
    last_bid_price /*quantity*/        number(15) default 0 not null,
```

```
Table SECURITY_TYPE_INFO created.


Trigger NEW_SECURITY_TYPE compiled


Table STOCK_SECURITY created.


Table FUND_SECURITY created.


Table BOND_SECURITY created.
```

Cerinta 5

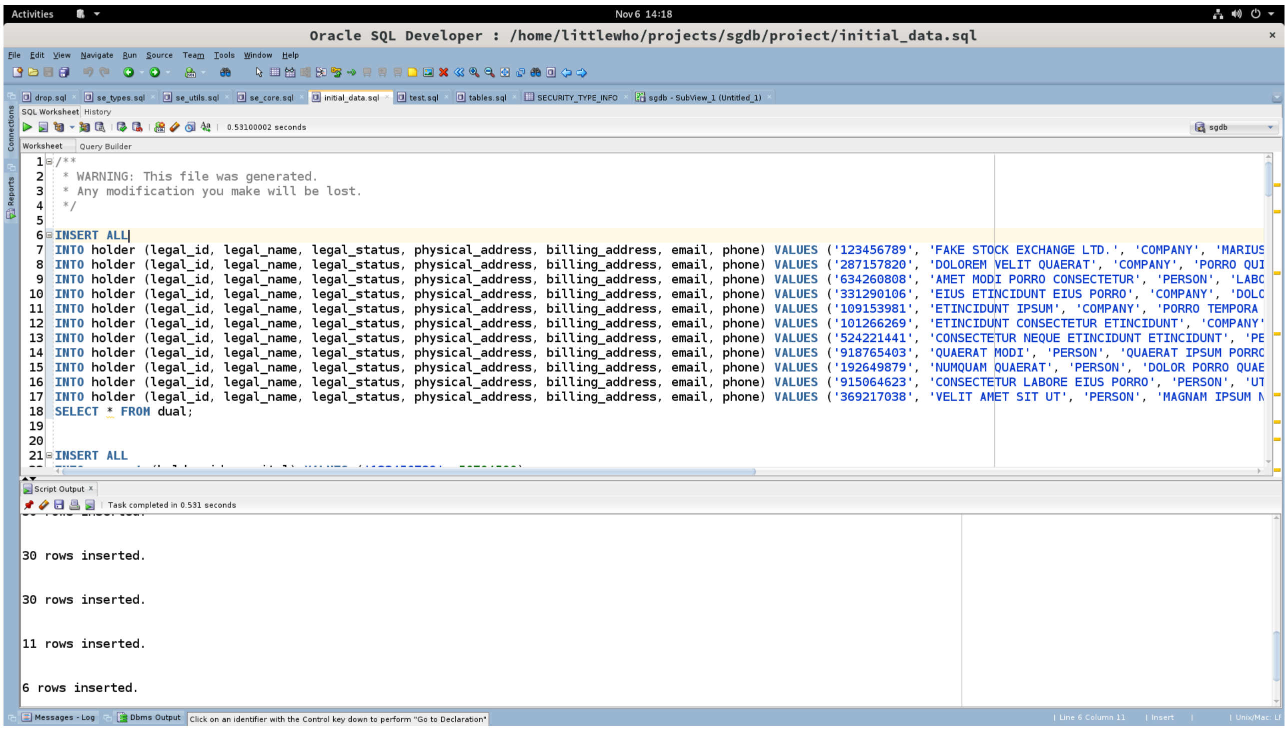
Datele inserate in tabele sunt generate automat. Sursa scriptului se poate gasi aici: `https://github.com/JustBeYou/SGDB/blob/master/proiect/gen.py`

```
/**
 * WARNING: This file was generated.
 * Any modification you make will be lost.
 */

INSERT ALL
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
VALUES ('123456789', 'FAKE STOCK EXCHANGE LTD.', 'COMPANY', 'MARIUS LACATUS NO. 7',
        'STEPHAN THE GREAT NO 1453', 'michaeljackson@obama.gov', '07222222')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('287157820', 'DOLOREM VELIT QUAERAT', 'COMPANY', 'PORRO QUISQUAM CONSECTETUR ADIPISCI LA
    'AMET LABORE SIT', 'company0@email.com', '+99-222-0')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('634260808', 'AMET MODI PORRO CONSECTETUR', 'PERSON',
    'LABORE LABORE DOLORE PORRO EST DOLOREM', 'AMET DOLORE EST', 'company1@email.com', '+99-222-1')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('331290106', 'EIUS ETINCIDUNT EIUS PORRO', 'COMPANY',
    'DOLOR VELIT CONSECTETUR', 'DOLOR ETINCIDUNT NON SIT NEQUE', 'company2@email.com', '+99-222-2')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('109153981', 'ETINCIDUNT IPSUM', 'COMPANY',
    'PORRO TEMPORA SED UT EIUS SED', 'LABORE TEMPORA', 'company3@email.com', '+99-222-3')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('101266269', 'ETINCIDUNT CONSECTETUR ETINCIDUNT', 'COMPANY',
    'SIT EST', 'EST ADIPISCI UT CONSECTETUR TEMPORA QUIQUIA', 'company4@email.com', '+99-222-4')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('524221441', 'CONSECTETUR NEQUE ETINCIDUNT ETINCIDUNT', 'PERSON',
    'NEQUE NON ADIPISCI CONSECTETUR', 'MODI ALIQUAM EIUS QUIQUIA DOLORE MAGNAM', 'company5@email.com
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('918765403', 'QUAERAT MODI', 'PERSON', 'QUAERAT IPSUM PORRO ADIPISCI NEQUE QUISQUAM',
    'MODI DOLORE NEQUE DOLORE SED UT', 'company6@email.com', '+99-222-6')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('192649879', 'NUMQUAM QUAERAT', 'PERSON', 'DOLOR PORRO QUAERAT UT VELIT ADIPISCI',
    'EST NUMQUAM SIT MAGNAM DOLOR MODI', 'company7@email.com', '+99-222-7')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('915064623', 'CONSECTETUR LABORE EIUS PORRO', 'PERSON',
    'UT MODI', 'VELIT QUISQUAM', 'company8@email.com', '+99-222-8')
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone)
    VALUES ('369217038', 'VELIT AMET SIT UT', 'PERSON', 'MAGNAM IPSUM NON SIT CONSECTETUR ETINCIDUNT
    'SIT PORRO AMET CONSECTETUR', 'company9@email.com', '+99-222-9')
SELECT * FROM dual;


INSERT ALL
INTO account (holder_id, capital) VALUES ('123456789', 56704500)
INTO account (holder_id, capital) VALUES ('287157820', 304217400)
INTO account (holder_id, capital) VALUES ('634260808', 945853200)
INTO account (holder_id, capital) VALUES ('331290106', 179966700)
INTO account (holder_id, capital) VALUES ('109153981', 173267900)
INTO account (holder_id, capital) VALUES ('101266269', 901900500)
INTO account (holder_id, capital) VALUES ('524221441', 287197900)
INTO account (holder_id, capital) VALUES ('918765403', 971694900)
INTO account (holder_id, capital) VALUES ('192649879', 715233300)
INTO account (holder_id, capital) VALUES ('915064623', 552150400)
INTO account (holder_id, capital) VALUES ('369217038', 152553400)
SELECT * FROM dual;


INSERT ALL
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('NC', 'NUMQUAM CONSECTETUR', 7941317, 1600, 4900, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('NV', 'NEQUE VELIT', 1856051, 1300, 18600, 'BOND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('QC', 'QUISQUAM CONSECTETUR', 2452166, 18000, 5400, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('DEIN', 'DOLOREM EST IPSUM NON', 5388521, 9800, 6200, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('EAP', 'EST AMET PORRO', 5423426, 5400, 19600, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('PMI', 'PORRO MAGNAM IPSUM', 3758807, 12700, 13600, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('TMPA', 'TEMPORA MAGNAM PORRO AMET', 1544585, 2200, 1000, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('DA', 'DOLOREM ADIPISCI', 6718479, 15900, 6100, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('LVIC', 'LABORE VELIT IPSUM CONSECTETUR', 3764784, 12300, 19500, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('DM', 'DOLOREM MAGNAM', 3755827, 15700, 16100, 'BOND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('NMA', 'NEQUE MODI ALIQUAM', 5240109, 16300, 14300, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('QENM', 'QUISQUAM EIUS NUMQUAM MAGNAM', 2226257, 12800, 2700, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('MQVME', 'MAGNAM QUIQUIA VOLUPTATEM MODI EIUS', 7381346, 6100, 11900, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('DQDEV', 'DOLORE QUAERAT DOLOR EST VELIT', 7239577, 15500, 10600, 'BOND_SECURITY')
```

```
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('QN', 'QUAERAT NEQUE', 4444127, 7400, 5800, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('SE', 'SIT ETINCIDUNT', 6813608, 9700, 13400, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('UD', 'UT DOLOREM', 7451037, 19000, 12300, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('IE', 'IPSUM ETINCIDUNT', 6139423, 12500, 3900, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('AAV', 'AMET ADIPISCI VELIT', 7495077, 14200, 9300, 'BOND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('TPAI', 'TEMPORA PORRO ADIPISCI IPSUM', 5198561, 7600, 16100, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('AE', 'ADIPISCI EIUS', 5637403, 600, 15400, 'BOND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('AQ', 'ALIQUAM QUIQUIA', 4719769, 9700, 3400, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('IU', 'IPSUM UT', 3578159, 15500, 12300, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('SPLSA', 'SED PORRO LABORE SED ADIPISCI', 9181037, 4300, 13700, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('VA', 'VELIT ALIQUAM', 1396933, 100, 8500, 'FUND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('ELICM', 'ETINCIDUNT LABORE IPSUM CONSECTETUR MODI', 9621084, 8800, 16500, 'STOCK_SECURIT
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('SUE', 'SED UT EST', 5205968, 11700, 11700, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('VDADD', 'VELIT DOLOR ALIQUAM DOLOR DOLOR', 7022473, 3700, 2900, 'STOCK_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('QSQ', 'QUAERAT SIT QUAERAT', 9046515, 6500, 13500, 'BOND_SECURITY')
INTO security (ticker, name, total_shares, last_ask_price, last_bid_price, type)
    VALUES ('IAQDS', 'IPSUM ALIQUAM QUISQUAM DOLOR SIT', 6761587, 15500, 14600, 'FUND_SECURITY')
SELECT * FROM dual;


INSERT ALL
INTO own (account_id, ticker, amount) VALUES (1, 'NC', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'NV', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'QC', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'DEIN', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'EAP', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'PMI', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'TMPA', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'DA', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'LVIC', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'DM', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'NMA', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'QENM', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'MQVME', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'DQDEV', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'QN', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'SE', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'UD', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'IE', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'AAV', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'TPAI', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'AE', 1)
```

```sql
INTO own (account_id, ticker, amount) VALUES (1, 'AQ', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'IU', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'SPLSA', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'VA', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'ELICM', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'SUE', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'VDADD', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'QSQ', 1)
INTO own (account_id, ticker, amount) VALUES (1, 'IAQDS', 1)
SELECT * FROM dual;


INSERT ALL
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'NC', 1, 7941316, 1600, 7941316)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'NV', 1, 1856050, 1300, 1856050)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'QC', 1, 2452165, 18000, 2452165)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'DEIN', 1, 5388520, 9800, 5388520)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'EAP', 1, 5423425, 5400, 5423425)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'PMI', 1, 3758806, 12700, 3758806)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'TMPA', 1, 1544584, 2200, 1544584)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'DA', 1, 6718478, 15900, 6718478)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'LVIC', 1, 3764783, 12300, 3764783)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'DM', 1, 3755826, 15700, 3755826)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'NMA', 1, 5240108, 16300, 5240108)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'QENM', 1, 2226256, 12800, 2226256)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'MQVME', 1, 7381345, 6100, 7381345)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'DQDEV', 1, 7239576, 15500, 7239576)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'QN', 1, 4444126, 7400, 4444126)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'SE', 1, 6813607, 9700, 6813607)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'UD', 1, 7451036, 19000, 7451036)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'IE', 1, 6139422, 12500, 6139422)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'AAV', 1, 7495076, 14200, 7495076)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'TPAI', 1, 5198560, 7600, 5198560)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'AE', 1, 5637402, 600, 5637402)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'AQ', 1, 4719768, 9700, 4719768)
```

```
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'IU', 1, 3578158, 15500, 3578158)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'SPLSA', 1, 9181036, 4300, 9181036)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'VA', 1, 1396932, 100, 1396932)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'ELICM', 1, 9621083, 8800, 9621083)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'SUE', 1, 5205967, 11700, 5205967)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'VDADD', 1, 7022472, 3700, 7022472)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'QSQ', 1, 9046514, 6500, 9046514)
INTO quotation (type, ticker, account_id, amount, price, remaining)
    VALUES ('ASK', 'IAQDS', 1, 6761586, 15500, 6761586)
SELECT * FROM dual;


INSERT ALL
INTO stock_security (ticker, type, pays_dividents) VALUES ('NC', 'COMMON', 1)
INTO stock_security (ticker, type, pays_dividents) VALUES ('EAP', 'COMMON', 1)
INTO stock_security (ticker, type, pays_dividents) VALUES ('PMI', 'COMMON', 0)
INTO stock_security (ticker, type, pays_dividents) VALUES ('NMA', 'COMMON', 1)
INTO stock_security (ticker, type, pays_dividents) VALUES ('MQVME', 'PREFERRED', 0)
INTO stock_security (ticker, type, pays_dividents) VALUES ('QN', 'COMMON', 1)
INTO stock_security (ticker, type, pays_dividents) VALUES ('UD', 'PREFERRED', 1)
INTO stock_security (ticker, type, pays_dividents) VALUES ('IE', 'COMMON', 1)
INTO stock_security (ticker, type, pays_dividents) VALUES ('ELICM', 'COMMON', 0)
INTO stock_security (ticker, type, pays_dividents) VALUES ('SUE', 'COMMON', 0)
INTO stock_security (ticker, type, pays_dividents) VALUES ('VDADD', 'PREFERRED', 0)
SELECT * FROM dual;


INSERT ALL
INTO bond_security (ticker, interest_rate) VALUES ('NV', 0.1)
INTO bond_security (ticker, interest_rate) VALUES ('DM', 0.09)
INTO bond_security (ticker, interest_rate) VALUES ('DQDEV', 0.04)
INTO bond_security (ticker, interest_rate) VALUES ('AAV', 0.05)
INTO bond_security (ticker, interest_rate) VALUES ('AE', 0.04)
INTO bond_security (ticker, interest_rate) VALUES ('QSQ', 0.09)
SELECT * FROM dual;


INSERT ALL
INTO fund_security (ticker, type) VALUES ('QC', 'HEDGE')
INTO fund_security (ticker, type) VALUES ('DEIN', 'HEDGE')
INTO fund_security (ticker, type) VALUES ('TMPA', 'MUTUAL')
INTO fund_security (ticker, type) VALUES ('DA', 'HEDGE')
INTO fund_security (ticker, type) VALUES ('LVIC', 'HEDGE')
INTO fund_security (ticker, type) VALUES ('QENM', 'HEDGE')
INTO fund_security (ticker, type) VALUES ('SE', 'MUTUAL')
INTO fund_security (ticker, type) VALUES ('TPAI', 'BOND')
INTO fund_security (ticker, type) VALUES ('AQ', 'MUTUAL')
INTO fund_security (ticker, type) VALUES ('IU', 'BOND')
INTO fund_security (ticker, type) VALUES ('SPLSA', 'HEDGE')
INTO fund_security (ticker, type) VALUES ('VA', 'HEDGE')
INTO fund_security (ticker, type) VALUES ('IAQDS', 'BOND')
SELECT * FROM dual;
```



Oracle SQL Developer : /home/littlewho/projects/sgdb/proiect/initial_data.sql

```
/**
 * WARNING: This file was generated.
 * Any modification you make will be lost.
 */

INSERT ALL
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('123456789', 'FAKE STOCK EXCHANGE LTD.', 'COMPANY', 'MARIUS
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('287157820', 'DOLOREM VELIT QUAERAT', 'COMPANY', 'PORRO QUI
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('634260808', 'AMET MODI PORRO CONSECTETUR', 'PERSON', 'LABO
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('331290106', 'EIUS ETINCIDUNT EIUS PORRO', 'COMPANY', 'DOLO
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('109153981', 'ETINCIDUNT IPSUM', 'COMPANY', 'PORRO TEMPORA
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('101266269', 'ETINCIDUNT CONSECTETUR ETINCIDUNT', 'COMPANY'
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('524221441', 'CONSECTETUR NEQUE ETINCIDUNT ETINCIDUNT', 'PE
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('918765403', 'QUAERAT MODI', 'PERSON', 'QUAERAT IPSUM PORRO
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('192649879', 'NUMQUAM QUAERAT', 'PERSON', 'DOLOR PORRO QUAE
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('915064623', 'CONSECTETUR LABORE EIUS PORRO', 'PERSON', 'UT
INTO holder (legal_id, legal_name, legal_status, physical_address, billing_address, email, phone) VALUES ('369217038', 'VELIT AMET SIT UT', 'PERSON', 'MAGNAM IPSUM N
SELECT * FROM dual;


INSERT ALL
```

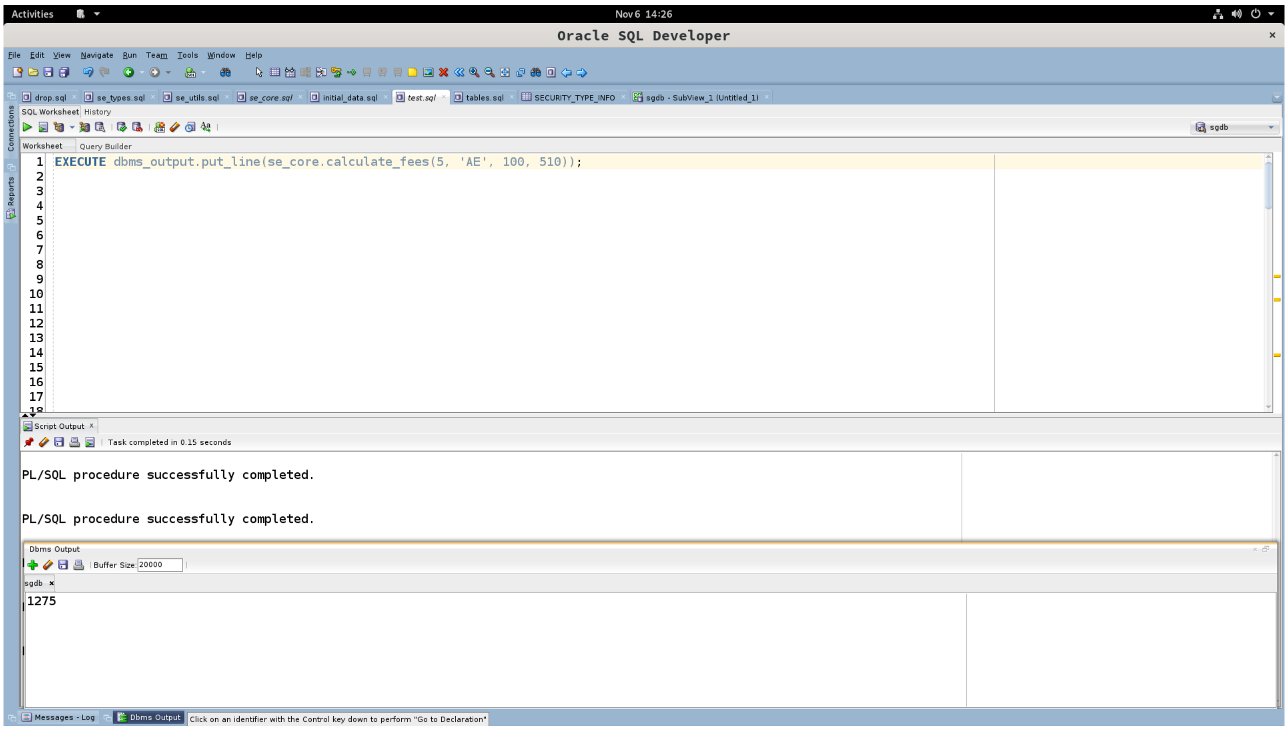
Script Output — Task completed in 0.531 seconds

```
30 rows inserted.


30 rows inserted.


11 rows inserted.


6 rows inserted.
```

Cerinta 6

Subprogramul calculeaza care este comisionul datorat detinatorului bursei la creearea unei oferte. (acesta face parte din **SE_CORE**)

```
FUNCTION calculate_fees(
    acc_id se_types.id,
    tic    se_types.abbreviation,
    amount se_types.quantity,
    price  se_types.quantity
) RETURN se_types.quantity IS
    thresholds se_types.thresholds_array;
    fee        se_types.percentage;
    fee_factor se_types.percentage;
    volume     se_types.quantity;
    acc        account%rowtype;
BEGIN
    -- fee to pay = standard security fee increased by threshold level
    thresholds := se_types.thresholds_array(
        threshold_fee_pair(se_utils.dollars_to_points(0), 0.25),
        threshold_fee_pair(se_utils.dollars_to_points(1000), 0.20),
        threshold_fee_pair(se_utils.dollars_to_points(5000), 0.15),
        threshold_fee_pair(se_utils.dollars_to_points(10000), 0.10),
        threshold_fee_pair(se_utils.dollars_to_points(50000), 0.5),
        threshold_fee_pair(se_utils.dollars_to_points(100000), 0.1)
    );

    SELECT * INTO acc
    FROM account a
    WHERE a.id = acc_id;
    SELECT COALESCE(SUM((price + spread_price) * amount), 0) INTO volume
    FROM (
        SELECT distinct t.id, t.price, t.amount, t.spread_price
        FROM account a
        INNER JOIN quotation q
        ON q.account_id = acc_id
        INNER JOIN trade t
        ON (t.ask_id = q.id OR t.bid_id = q.id)
    );

    FOR i IN 1 .. thresholds.count LOOP
        IF (thresholds(i).above(volume) = 1) THEN
            fee_factor := thresholds(i).fee_factor;
        END IF;
    END LOOP;

    SELECT sti.fee INTO fee
    FROM security_type_info sti, security s
    WHERE sti.name = s.type AND s.ticker = tic;

    RETURN CEIL(
        se_utils.increase_by_percent(
            fee * price * amount,
            fee_factor
        )
    );
EXCEPTION
    WHEN no_data_found THEN
        dbms_output.put_line('Ticker or account not found.');
    WHEN others THEN
        se_utils.exception_fallback;
END calculate_fees;
```



```
EXECUTE dbms_output.put_line(se_core.calculate_fees(5, 'AE', 100, 510));
```

Task completed in 0.15 seconds

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

Buffer Size: 20000

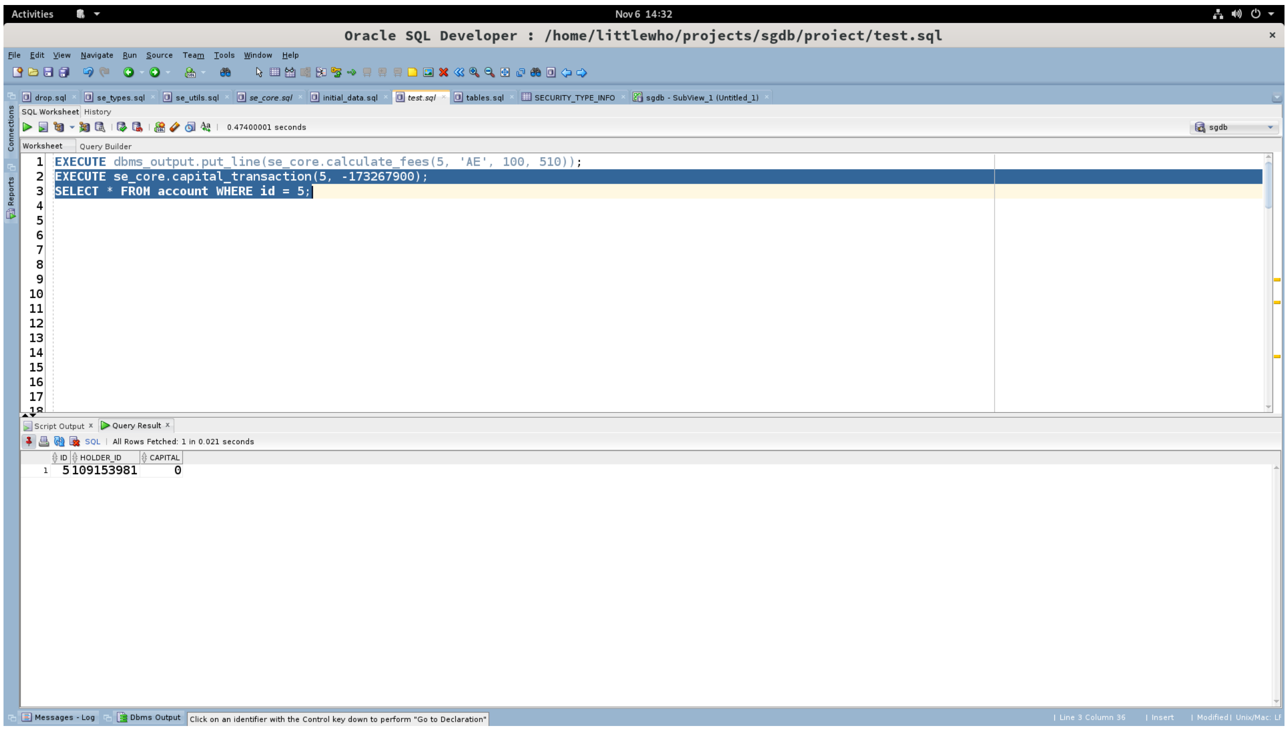
```
1275
```

Click on an identifier with the Control key down to perform "Go to Declaration"

Cerinta 7

Subprogramul efectueaza un transfer de capital si utilizeaza cursorii pentru a asigura atomicitatea tranzactiei.

```
PROCEDURE capital_transaction(
    acc_id se_types.id,
    amount se_types.quantity
) IS
    current_capital se_types.quantity;

    CURSOR c_account IS
        SELECT a.capital
        FROM account a
        WHERE a.id = acc_id and a.capital + amount >= 0
        FOR UPDATE OF a.capital;

    under_zero EXCEPTION;
BEGIN
    OPEN c_account;

    FETCH c_account INTO current_capital;
    IF c_account%notfound THEN
        RAISE under_zero;
    END IF;

    UPDATE account a
    SET a.capital = current_capital + amount
    WHERE a.id = acc_id;

    COMMIT;
    CLOSE c_account;
EXCEPTION
    WHEN under_zero THEN
        dbms_output.put_line('Insufficient funds.');
    WHEN others THEN
        se_utils.exception_fallback;
END capital_transaction;
```


Oracle SQL Developer : /home/littlewho/projects/sgdb/proiect/test.sql

```
EXECUTE dbms_output.put_line(se_core.calculate_fees(5, 'AE', 100, 510));
EXECUTE se_core.capital_transaction(5, -173267900);
SELECT * FROM account WHERE id = 5;
```

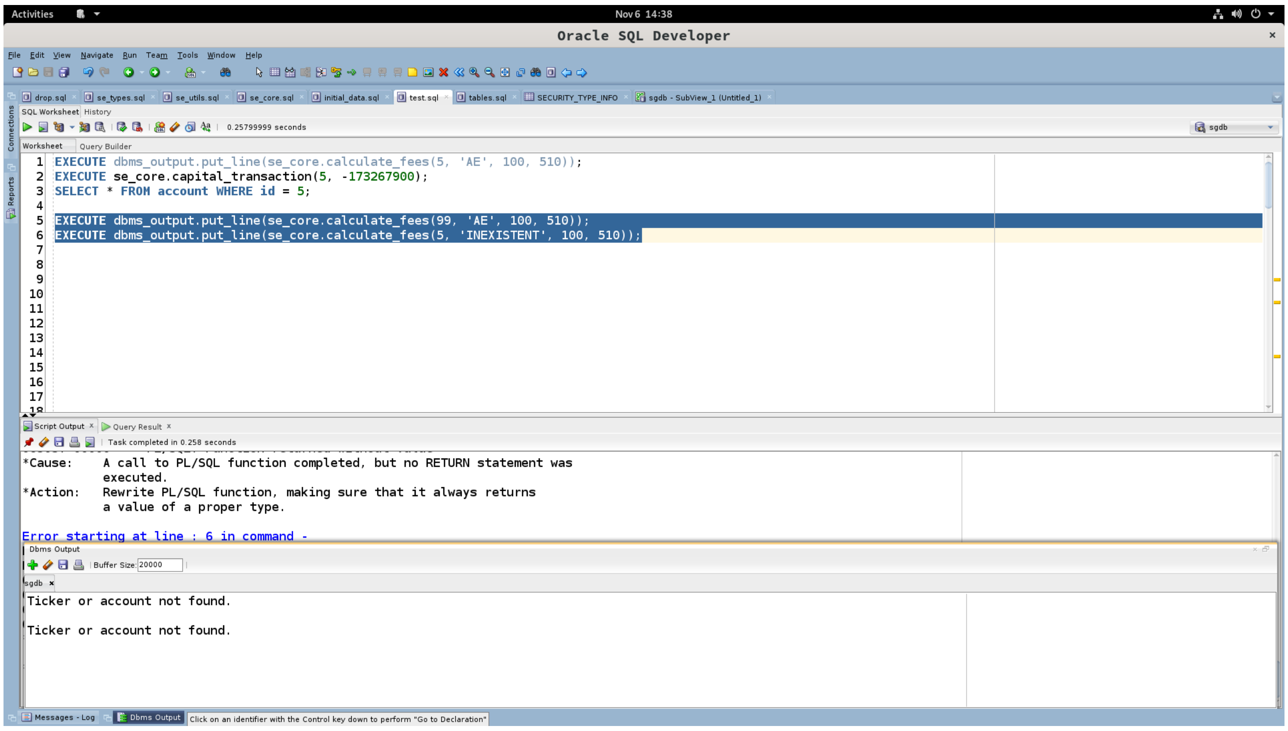
All Rows Fetched: 1 in 0.021 seconds

```
   ID HOLDER_ID  CAPITAL
1   5 109153981        0
```

Cerinta 8

Functia prezentata mai sus, anume **CALCULATE_FEES**, indeplineste cerinta. Aceasta poate arunca exceptia **NO_DATA_FOUND**, doar in cazul in care *acc_id* sau *ticker* sunt invalide.

```
EXECUTE dbms_output.put_line(se_core.calculate_fees(99, 'AE', 100, 510));
EXECUTE dbms_output.put_line(se_core.calculate_fees(5, 'INEXISTENT', 100, 510));
```


Oracle SQL Developer

```
EXECUTE dbms_output.put_line(se_core.calculate_fees(5, 'AE', 100, 510));
EXECUTE se_core.capital_transaction(5, -173267900);
SELECT * FROM account WHERE id = 5;

EXECUTE dbms_output.put_line(se_core.calculate_fees(99, 'AE', 100, 510));
EXECUTE dbms_output.put_line(se_core.calculate_fees(5, 'INEXISTENT', 100, 510));
```

Task completed in 0.258 seconds

```
*Cause:    A call to PL/SQL function completed, but no RETURN statement was
           executed.
*Action:   Rewrite PL/SQL function, making sure that it always returns
           a value of a proper type.

Error starting at line : 6 in command -
```

Dbms Output — Buffer Size: 20000

```
Ticker or account not found.

Ticker or account not found.
```

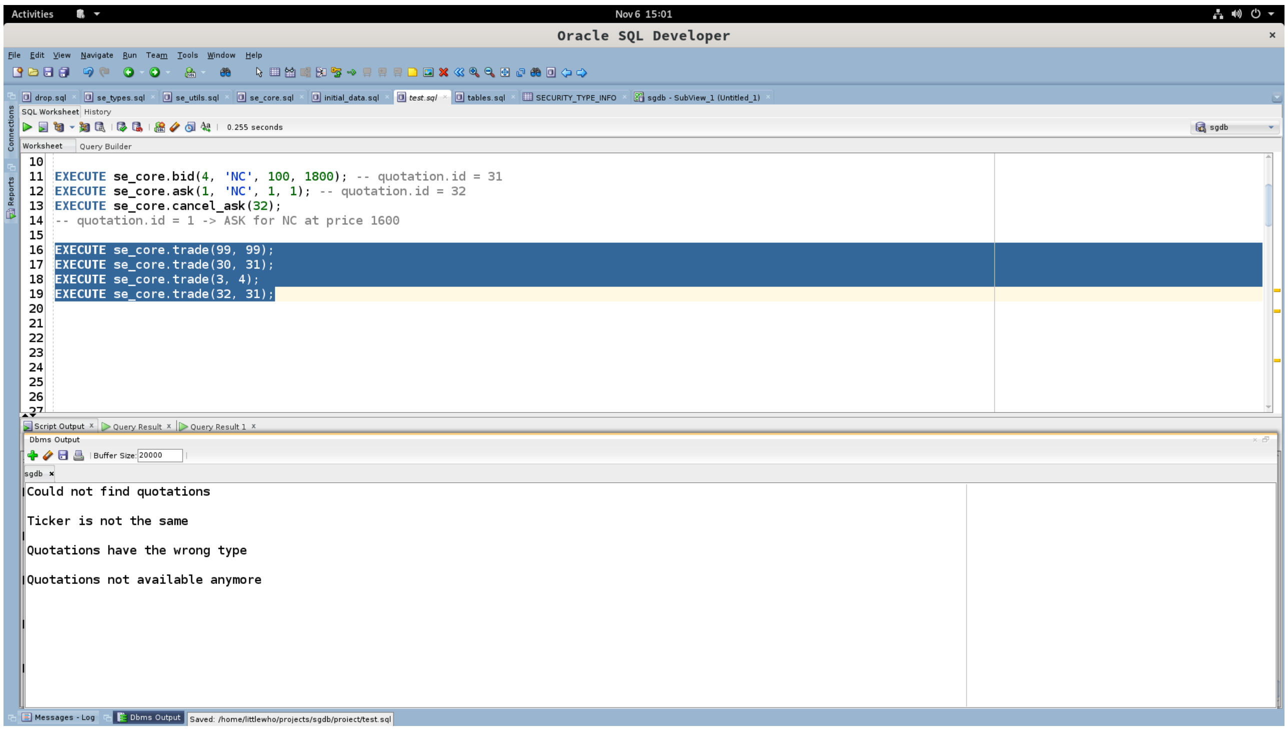
Cerinta 9

Subprogramul **TRADE** asociaza doua oferte pentru realizarea unei tranzactii.

```
PROCEDURE trade(
    ask_id se_types.id,
    bid_id se_types.id
) IS
    ask                 quotation%rowtype;
    bid                 quotation%rowtype;
    ask_would_fulfill  se_types.status := 0;
    bid_would_fulfill  se_types.status := 0;
    spread              se_types.quantity;
    shares_to_transfer se_types.quantity;
    trade_price         se_types.quantity;


    CURSOR c_ask IS
        SELECT *
        FROM quotation q
        WHERE q.id = ask_id
        FOR UPDATE OF fulfilled, remaining;

    CURSOR c_bid IS
        SELECT *
        FROM quotation q
        WHERE q.id = bid_id
        FOR UPDATE OF fulfilled, remaining;

    not_found EXCEPTION;
    wrong_ticker EXCEPTION;
    invalid_type EXCEPTION;
    not_available EXCEPTION;
    cant_match EXCEPTION;
BEGIN
    OPEN c_ask;
    OPEN c_bid;

    FETCH c_ask INTO ask;
    FETCH c_bid INTO bid;
    IF (c_ask%notfound OR c_bid%notfound) THEN
        RAISE not_found;
    END IF;

    IF (ask.type != 'ASK' or bid.type != 'BID') THEN
        RAISE invalid_type;
    END IF;

    IF (ask.ticker != bid.ticker) THEN
        RAISE wrong_ticker;
    END IF;

    IF (ask.price > bid.price) THEN
        RAISE cant_match;
    END IF;

    IF (ask.fulfilled = 1 OR
        bid.fulfilled = 1 OR
        ask.deleted = 1 OR
        bid.deleted = 1) THEN
        RAISE not_available;
```

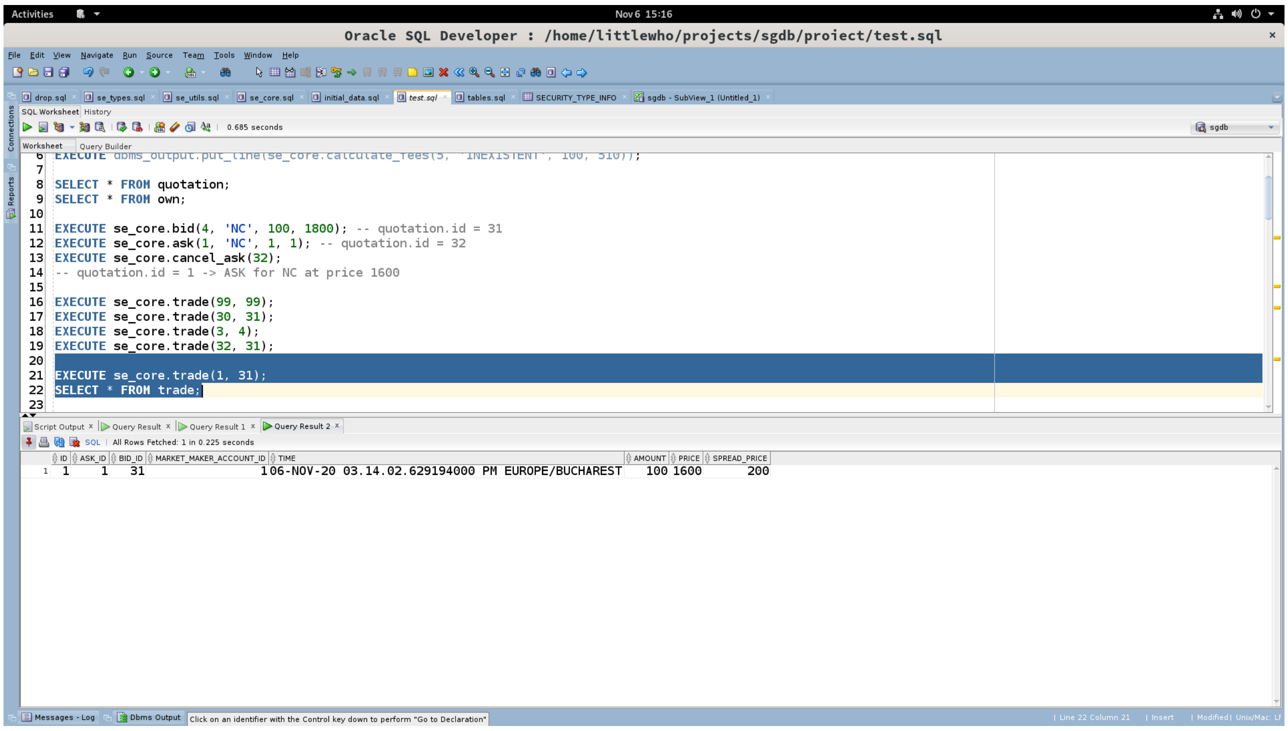
```
END IF;

spread := bid.price - ask.price;
trade_price := ask.price;
shares_to_transfer := LEAST(ask.remaining, bid.remaining);

dbms_output.put_line('Trade ' || shares_to_transfer ||
    ' ' || ask.ticker || ' at price ' || trade_price
    || ' from ' || ask.account_id || ' to ' || bid.account_id ||
    ' with spread of ' || spread);

IF (shares_to_transfer = ask.remaining) THEN
    ask_would_fulfill := 1;
END IF;

IF (shares_to_transfer = bid.remaining) THEN
    bid_would_fulfill := 1;
END IF;

UPDATE quotation q
SET q.fulfilled = ask_would_fulfill,
    q.remaining = ask.remaining - shares_to_transfer
WHERE q.id = ask.id;

UPDATE quotation q
SET q.fulfilled = bid_would_fulfill,
    q.remaining = bid.remaining - shares_to_transfer
WHERE q.id = bid.id;

capital_transaction(ask.account_id, trade_price * shares_to_transfer);
shares_transaction(bid.account_id, bid.ticker, shares_to_transfer);

capital_transaction(default_market_acc_id, spread * shares_to_transfer);

INSERT INTO trade (
    ask_id,
    bid_id,
    market_maker_account_id,
    time,
    amount,
    price,
    spread_price
) VALUES (
    ask.id,
    bid.id,
    default_market_acc_id,
    CURRENT_TIMESTAMP,
    shares_to_transfer,
    trade_price,
    spread
);

UPDATE security s
SET s.last_ask_price = ask.price,
    s.last_bid_price = bid.price
WHERE s.ticker = bid.ticker;
```

```
    COMMIT;
    CLOSE c_ask;
    CLOSE c_bid;
EXCEPTION
    WHEN not_found THEN
        dbms_output.put_line('Could not find quotations');
    WHEN wrong_ticker THEN
        dbms_output.put_line('Ticker is not the same');
    WHEN not_available THEN
        dbms_output.put_line('Quotations not available anymore');
    WHEN invalid_type THEN
        dbms_output.put_line('Quotations have the wrong type');
    WHEN cant_match THEN
        dbms_output.put_line('ASK > BID');
    WHEN others THEN
        se_utils.exception_fallback;
END;
```


Activities
Nov 6 15:01
Oracle SQL Developer
File Edit View Navigate Run Team Tools Window Help
drop.sql se_types.sql se_utils.sql se_core.sql initial_data.sql test.sql tables.sql SECURITY_TYPE_INFO sgdb - SubView_1 (Untitled_1)
SQL Worksheet History
0.255 seconds
Worksheet Query Builder
Script Output Query Result Query Result 1
Dbms Output
Buffer Size 20000
sgdb
Messages - Log Dbms Output
Saved: /home/littlewho/projects/sgdb/project/test.sql

```

EXECUTE se_core.bid(4, 'NC', 100, 1800); -- quotation.id = 31
EXECUTE se_core.ask(1, 'NC', 1, 1); -- quotation.id = 32
EXECUTE se_core.cancel_ask(32);
-- quotation.id = 1 -> ASK for NC at price 1600

EXECUTE se_core.trade(99, 99);
EXECUTE se_core.trade(30, 31);
EXECUTE se_core.trade(3, 4);
EXECUTE se_core.trade(32, 31);
```

```
Could not find quotations

Ticker is not the same

Quotations have the wrong type

Quotations not available anymore
```


Oracle SQL Developer : /home/littlewho/projects/sgdb/proiect/test.sql

```
EXECUTE dbms_output.put_line(se_core.calculate_fees(5, 'INEXISTENT', 100, 510));

SELECT * FROM quotation;
SELECT * FROM own;

EXECUTE se_core.bid(4, 'NC', 100, 1800); -- quotation.id = 31
EXECUTE se_core.ask(1, 'NC', 1, 1); -- quotation.id = 32
EXECUTE se_core.cancel_ask(32);
-- quotation.id = 1 -> ASK for NC at price 1600

EXECUTE se_core.trade(99, 99);
EXECUTE se_core.trade(30, 31);
EXECUTE se_core.trade(3, 4);
EXECUTE se_core.trade(32, 31);

EXECUTE se_core.trade(1, 31);
SELECT * FROM trade;
```

All Rows Fetched: 1 in 0.225 seconds

	ID	ASK_ID	BID_ID	MARKET_MAKER_ACCOUNT_ID	TIME	AMOUNT	PRICE	SPREAD_PRICE
1	1	1	31	1	06-NOV-20 03.14.02.629194000 PM EUROPE/BUCHAREST	100	1600	200

Cerinta 10

...

Cerinta 11

Trigger-ul **QUOTATION_AUTO_ID** este utilizat pentru a genera automat ID-urile noilor intrari. Declansare lui este vizibila in **Cerinta 5**.

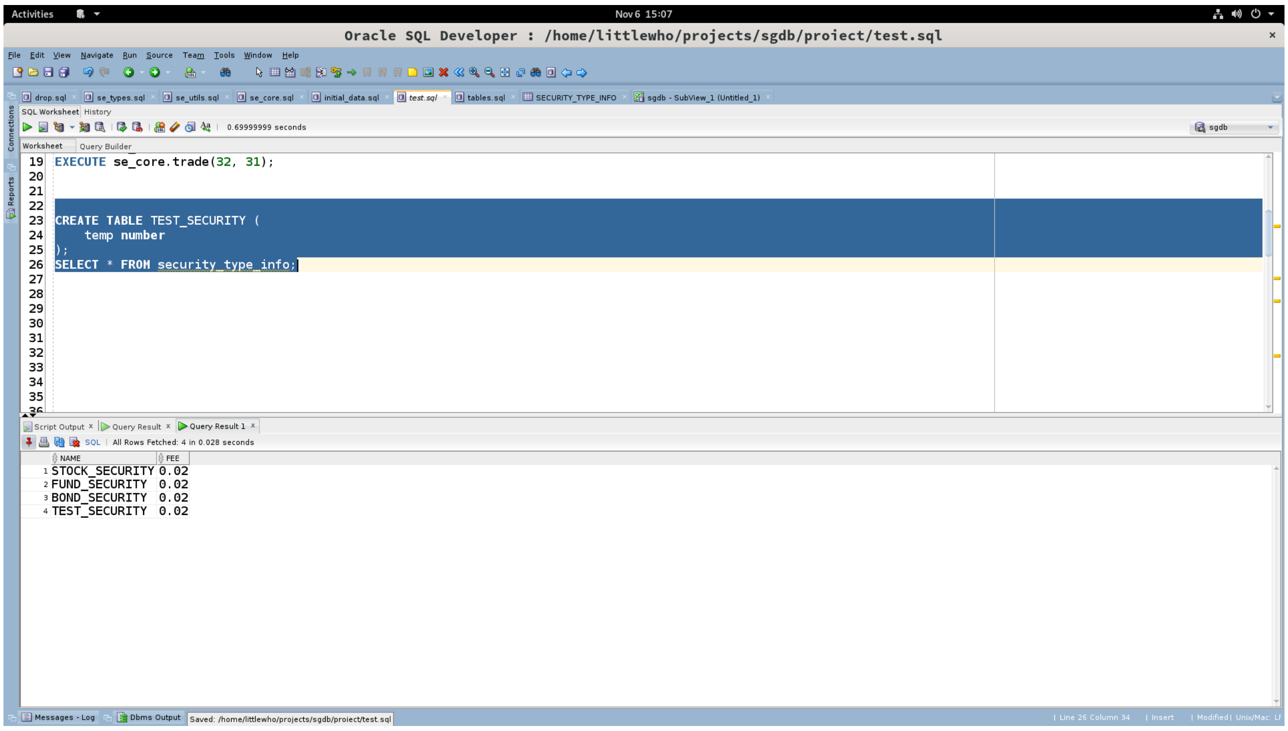
```
create sequence quotation_ids start with 1;
create or replace trigger quotation_auto_id
before insert on quotation
for each row
begin
  select quotation_ids.nextval
  into   :new.id
  from   dual;
end;
/
```

Cerinta 12

Atunci cand se doreste adaugarea unui nou tip de instrument financiar, se va creea un tabel cu numele **<TIP_INSTRUMENT>_SECURITY**. Dupa creare, se va declansa un trigger care va adauga noul tip de instrument in tabela cu informatii. Acesta este definit astfel:

```
CREATE TRIGGER new_security_type AFTER CREATE
    ON SCHEMA
    DECLARE
    BEGIN
        FOR ctable IN (
            SELECT a.table_name
            FROM all_tables a
            WHERE a.table_name LIKE '%_SECURITY'
```

```
                AND a.table_name NOT IN (
                       SELECT b.name as table_name from
                       security_type_info b
                    )
        ) LOOP
            INSERT INTO security_type_info (name)
            VALUES (ctable.table_name);
        END LOOP;
    END;
/
```



```
EXECUTE se_core.trade(32, 31);



CREATE TABLE TEST_SECURITY (
    temp number
);
SELECT * FROM security_type_info;
```

All Rows Fetched: 4 in 0.028 seconds

```
   NAME           FEE
 1 STOCK_SECURITY 0.02
 2 FUND_SECURITY  0.02
 3 BOND_SECURITY  0.02
 4 TEST_SECURITY  0.02
```

Rezolvarea cerintelor suplimentare

Cerinta 13

Proiectul este impartit in 3 pachete descrise la inceputul documentului. Acestea se pot gasi in fisierele **se_types.sql**, **se_utils.sql** si **se_core.sql**.

Cerinta 14

Consider ca cerintele anterioare cat si codul sursa ilustreaza suficient de bine operatiile si structurile de date definite, in proiect regasindu-se toate conceptele importante de PL/SQL: proceduri, functii, cursori, tranzactii, colectii, obiecte, exceptii, tipuri de date definite de utilizator, etc.

In cocluzie, cerinta este indeplinita de la sine.