

## РЫНОЧНЫЙ РИСК

### ИНСТРУКЦИЯ К ЗАДАНИЮ 1

Среда разработки PostgreSQL, **все пункты делаются последовательно!**

1. Разбить **xlsx** файл на две части (рыночная информация – вкладки **base1** и **base2** и характеристики облигаций – вкладка **instrs**) и сохранить в формате CSV.

2. Создать таблицу для рыночной информации и загрузить соответствующий файл CSV (ссылка на CSV файл

[https://drive.google.com/open?id=1sMMTIA\\_B9EkYVGXZKTaCloM0\\_UeNVQg](https://drive.google.com/open?id=1sMMTIA_B9EkYVGXZKTaCloM0_UeNVQg))

```
create table base1 (ID integer, TIME date, ACCRUEDINT float,
ASK float,
ASK_SIZE float,
ASK_SIZE_TOTAL float,
AVGE_PRICE float,
BID float,
BID_SIZE float,
BID_SIZE_TOTAL float,
BOARDID varchar(255),
BOARDNAME varchar(255),
BUYBACKDATE float,
BUYBACKPRICE float,
CBR_LOMBARD float,
CBR_PLEDGE float,
CLOSE float,
CPN float,
CPN_DATE date,
CPN_PERIOD float,
DEAL_ACC float,
FACEVALUE varchar(255),
ISIN varchar(255),
ISSUER varchar(255),
ISSUESIZE float,
MAT_DATE date,
MPRICE float,
MPRICE2 float,
SPREAD float,
VOL_ACC float, Y2O_ASK float, Y2O_BID float, YIELD_ASK float,
YIELD_BID float);

COPY base1 FROM '/ВАШ_ПУТЬ/base_prices2.csv' DELIMITER ';' CSV HEADER;
select * from base1 limit 5;
```

3. Создать таблицу для характеристик облигаций и загрузить соответствующий файл CSV (ссылка на CSV файл [https://drive.google.com/open?id=1\\_Ls9\\_Ctom13Z6yI9liuXatlBKnj1T68](https://drive.google.com/open?id=1_Ls9_Ctom13Z6yI9liuXatlBKnj1T68))

```
create table instrs1 (
ID int,
EfirCode varchar(250),
ShortNameRus varchar(250),
FullNameRus varchar(250),
ISIN varchar(250),
EfirCFI varchar(250),
CFIName varchar(250),
Exchange varchar(250),
ExchTicker varchar(250),
ExchSymbol varchar(250),
EmitentCode varchar(250),
EmitentName varchar(250),
MarketSector varchar(250),
LotSize float,
ExpDate date,
Currency varchar(250),
Visible varchar(250),
RegNum float);

COPY instrs1 FROM '/ВАШ_ПУТЬ/instrs.csv' DELIMITER ';' CSV HEADER;
select * from instrs1 limit 5;
```

4. В таблице BASE1 определить доли полей, заполненных менее чем на 50%. Значения выбираются вручную из таблицы, где будут выведены доли пустых полей, код таблицы:

```
select CAST(sum(case when isin is null then 1 else 0 end) as FLOAT) / /*в числителе количество полей с нулевым значением*/  
      (CAST(count(isin) as FLOAT) + CAST(sum(case when isin is null then 1 else 0 end) as FLOAT))  
as ISIN_null_share, /*в знаменателе сумма полей с нулевым и ненулевым значением*/  
      CAST(sum(case when id is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(id) as FLOAT) + CAST(sum(case when id is null then 1 else 0 end) as FLOAT))  
as ID_null_share,  
      CAST(sum(case when TIME is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(TIME) as FLOAT) + CAST(sum(case when TIME is null then 1 else 0 end) as FLOAT))  
as TIME_null_share,  
      CAST(sum(case when accruedint is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(accruedint) as FLOAT) + CAST(sum(case when accruedint is null then 1 else 0 end) as FLOAT))  
as accruedint_null_share,  
      CAST(sum(case when ASK is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(ASK) as FLOAT) + CAST(sum(case when ASK is null then 1 else 0 end) as FLOAT))  
as ASK_null_share,  
      CAST(sum(case when ask_size is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(ask_size) as FLOAT) + CAST(sum(case when ask_size is null then 1 else 0 end) as FLOAT))  
as ask_size_null_share,  
      CAST(sum(case when ask_size_total is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(ask_size_total) as FLOAT) + CAST(sum(case when ask_size_total is null then 1 else 0 end) as FLOAT))  
as ask_size_total_null_share,  
      CAST(sum(case when avge_prce is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(avge_prce) as FLOAT) + CAST(sum(case when avge_prce is null then 1 else 0 end) as FLOAT))  
as avge_prce_null_share,  
      CAST(sum(case when BID is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(BID) as FLOAT) + CAST(sum(case when BID is null then 1 else 0 end) as FLOAT))  
as BID_null_share,  
      CAST(sum(case when BID_SIZE is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(BID_SIZE) as FLOAT) + CAST(sum(case when BID_SIZE is null then 1 else 0 end) as FLOAT))  
as BID_SIZE_null_share,  
      CAST(sum(case when BID_SIZE_TOTAL is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(BID_SIZE_TOTAL) as FLOAT) + CAST(sum(case when BID_SIZE_TOTAL is null then 1 else 0 end) as FLOAT))  
as BID_SIZE_TOTAL_null_share,  
      CAST(sum(case when BOARDID is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(BOARDID) as FLOAT) + CAST(sum(case when BOARDID is null then 1 else 0 end) as FLOAT))  
as BOARDID_null_share,  
      CAST(sum(case when BOARDNAME is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(BOARDNAME) as FLOAT) + CAST(sum(case when BOARDNAME is null then 1 else 0 end) as FLOAT))  
as BOARDNAME_null_share,  
      CAST(sum(case when BUYBACKDATE is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(BUYBACKDATE) as FLOAT) + CAST(sum(case when BUYBACKDATE is null then 1 else 0 end) as FLOAT))  
as BUYBACKDATE_null_share,  
      CAST(sum(case when BUYBACKPRICE is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(BUYBACKPRICE) as FLOAT) + CAST(sum(case when BUYBACKPRICE is null then 1 else 0 end) as FLOAT))  
as BUYBACKPRICE_null_share,  
      CAST(sum(case when CBR_LOMBARD is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(CBR_LOMBARD) as FLOAT) + CAST(sum(case when CBR_LOMBARD is null then 1 else 0 end) as FLOAT))  
as CBR_LOMBARD_null_share,  
      CAST(sum(case when CBR_PLEDGE is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(CBR_PLEDGE) as FLOAT) + CAST(sum(case when CBR_PLEDGE is null then 1 else 0 end) as FLOAT))  
as CBR_PLEDGE_null_share,  
      CAST(sum(case when CLOSE is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(CLOSE) as FLOAT) + CAST(sum(case when CLOSE is null then 1 else 0 end) as FLOAT))  
as CLOSE_null_share,  
      CAST(sum(case when CPN is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(CPN) as FLOAT) + CAST(sum(case when CPN is null then 1 else 0 end) as FLOAT))  
as CPN_null_share,  
      CAST(sum(case when CPN_DATE is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(CPN_DATE) as FLOAT) + CAST(sum(case when CPN_DATE is null then 1 else 0 end) as FLOAT))  
as CPN_DATE_null_share,  
      CAST(sum(case when CPN_PERIOD is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(CPN_PERIOD) as FLOAT) + CAST(sum(case when CPN_PERIOD is null then 1 else 0 end) as FLOAT))  
as CPN_PERIOD_null_share,  
      CAST(sum(case when DEAL_ACC is null then 1 else 0 end) as FLOAT) /  
      (CAST(count(DEAL_ACC) as FLOAT) + CAST(sum(case when DEAL_ACC is null then 1 else 0 end) as FLOAT))  
as DEAL_ACC_null_share,  
      CAST(sum(case when ISSUER is null then 1 else 0 end) as FLOAT) /
```

```

(CAST(count(ISSUER) as FLOAT) + CAST(sum(case when ISSUER is null then 1 else 0 end) as FLOAT))
as ISSUER_null_share,
CAST(sum(case when ISSUESIZE is null then 1 else 0 end) as FLOAT) /
(CAST(count(ISSUESIZE) as FLOAT) + CAST(sum(case when ISSUESIZE is null then 1 else 0 end) as FLOAT))
as ISSUESIZE_null_share,
CAST(sum(case when MAT_DATE is null then 1 else 0 end) as FLOAT) /
(CAST(count(MAT_DATE) as FLOAT) + CAST(sum(case when MAT_DATE is null then 1 else 0 end) as FLOAT))
as MAT_DATE_null_share,
CAST(sum(case when MPRICE is null then 1 else 0 end) as FLOAT) /
(CAST(count(MPRICE) as FLOAT) + CAST(sum(case when MPRICE is null then 1 else 0 end) as FLOAT))
as MPRICE_null_share,
CAST(sum(case when MPRICE2 is null then 1 else 0 end) as FLOAT) /
(CAST(count(MPRICE2) as FLOAT) + CAST(sum(case when MPRICE2 is null then 1 else 0 end) as FLOAT))
as MPRICE2_null_share,
CAST(sum(case when SPREAD is null then 1 else 0 end) as FLOAT) /
(CAST(count(SPREAD) as FLOAT) + CAST(sum(case when SPREAD is null then 1 else 0 end) as FLOAT))
as SPREAD_null_share,
CAST(sum(case when VOL_ACC is null then 1 else 0 end) as FLOAT) /
(CAST(count(VOL_ACC) as FLOAT) + CAST(sum(case when VOL_ACC is null then 1 else 0 end) as FLOAT))
as VOL_ACC_null_share,
CAST(sum(case when Y2O_ASK is null then 1 else 0 end) as FLOAT) /
(CAST(count(Y2O_ASK) as FLOAT) + CAST(sum(case when Y2O_ASK is null then 1 else 0 end) as FLOAT))
as Y2O_ASK_null_share,
CAST(sum(case when Y2O_BID is null then 1 else 0 end) as FLOAT) /
(CAST(count(Y2O_BID) as FLOAT) + CAST(sum(case when Y2O_BID is null then 1 else 0 end) as FLOAT))
as Y2O_BID_null_share,
CAST(sum(case when YIELD_ASK is null then 1 else 0 end) as FLOAT) /
(CAST(count(YIELD_ASK) as FLOAT) + CAST(sum(case when YIELD_ASK is null then 1 else 0 end) as FLOAT))
as YIELD_ASK_null_share,
CAST(sum(case when YIELD_BID is null then 1 else 0 end) as FLOAT) /
(CAST(count(YIELD_BID) as FLOAT) + CAST(sum(case when YIELD_BID is null then 1 else 0 end) as FLOAT))
as YIELD_BID_null_share

```

from base1;

Результат выполнения скрипта – доля ПУСТЫХ ячеек в столбце, если больше 0,5 столбец выбрасывается:

buybackdate_null_share double precision	buybackprice_null_share double precision	cbr_lombard_null_share double precision	cbr_pledge_null_share double precision
0.750742443392564	0.573857112174006	0.524254871832999	0.774083459311211

5. После данной процедуры из столбцов, пустых более чем на 50% создается новая таблица

```

create table base1_null_fields AS select
id,
avge_prce,
boardid,
buybackdate,
buybackprice,
cbr_lombard,
cbr_pledge,
y2o_ask,
y2o_bid,
yield_ask,
yield_bid from base1;
select * from base1_null_fields limit 5;

```

6. Пустые столбцы удаляются из основной таблицы с рыночной информацией BASE1

```

ALTER TABLE base1 DROP COLUMN avge_prce,
DROP COLUMN boardid,
DROP COLUMN buybackdate,
DROP COLUMN buybackprice,
DROP COLUMN cbr_lombard,

```

```

DROP COLUMN cbr_pledge,
DROP COLUMN y2o_ask,
DROP COLUMN y2o_bid,
DROP COLUMN yield_ask,
DROP COLUMN yield_bid;
select * from base1 limit 5;

```

7. Следующим запросом выполняется поиск трех облигаций у которых доля не торговых дней (по ASK) менее 10% (в нашем случае менее 30%):

```

select distinct id, isin from base1 where (select CAST(sum(case when ASK = 0 then 1 else 0 end) as FLOAT) /
(CAST(count(ASK) as FLOAT))
as ASK_null_share
from base1) < 0.3 limit 3 ;

```

Результат скрипта – отобранные облигации:

	id integer	isin character varying (255)
1	14515	RU0001737595
2	14547	RU0007489621
3	32874	RU0008958863