

Задача №1

Существует таблица котировок финансовых инструментов

```
create table Rate(  
id int not null,  
code varchar(50) not null,  
date datetime not null,  
value float not null  
)
```

- Дата в поле date хранится без времени.

- Записи в таблицу добавляются в день смены курса (не каждый день). Курс действует с даты установки по дату следующей установки (не включительно), см. ниже.

1. Необходимо сгенерировать календарь по рабочим дням, поместить в таблицу Calendar за период с 05.06.2009 по 18.06.2009.

2. Вывести значение курса по каждой валюте за каждый день периода с 05.06.2009 по 18.06.2009 по каждой валюте.

Рабочие дни перечислены в таблице ниже.

Схема действия курса:

```
insert into rate (id, code, date, value) values (1,'EUR','20090605', 1.149) insert into rate (id, code,  
date, value) values (2,'EUR','20090615', 1.161) insert into rate (id, code, date, value) values  
(3,'EUR','20090617', 1.177) insert into rate (id, code, date, value) values (4,'USD','20090605',  
1.625) insert into rate (id, code, date, value) values (5,'USD','20090615', 1.639) insert into rate (id,  
code, date, value) values (6,'USD','20090617', 1.644)
```

Рабочие дни:

```
create table Calendar ( dDay datetime not null, ) Например: code dDay value weekday -----  
----- ----- EUR 2009-06-05 00:00:00.000 1.149 Friday EUR 2009-06-08  
00:00:00.000 1.149 Monday EUR 2009-06-09 00:00:00.000 1.149 Tuesday EUR 2009-06-10  
00:00:00.000 1.149 Wednesday EUR 2009-06-11 00:00:00.000 1.149 Thursday EUR 2009-06-15  
00:00:00.000 1.161 Monday EUR 2009-06-16 00:00:00.000 1.161 Tuesday EUR 2009-06-17  
00:00:00.000 1.177 Wednesday EUR 2009-06-18 00:00:00.000 1.177 Thursday EUR 2009-06-19  
00:00:00.000 1.177 Friday USD 2009-06-03 00:00:00.000 1.625 Wednesday USD 2009-06-04  
00:00:00.000 1.625 Thursday USD 2009-06-05 00:00:00.000 1.625 Friday USD 2009-06-08  
00:00:00.000 1.625 Monday USD 2009-06-09 00:00:00.000 1.625 Tuesday USD 2009-06-10  
00:00:00.000 1.625 Wednesday USD 2009-06-11 00:00:00.000 1.639 Thursday USD 2009-06-15  
00:00:00.000 1.639 Monday USD 2009-06-16 00:00:00.000 1.639 Tuesday USD 2009-06-17  
00:00:00.000 1.644 Wednesday USD 2009-06-18 00:00:00.000 1.644 Thursday USD 2009-06-19  
00:00:00.000 1.644 Friday
```

Решение

--создание таблицы rate

```
CREATE TABLE Rate(  
    id int not null,  
    code varchar(50) not null,  
    date date not null,  
    value float not null  
);
```

--наполнение таблицы rate

```
INSERT INTO rate (id, code, date, value) values (1,'EUR','20090605', 1.149);  
INSERT INTO rate (id, code, date, value) values (2,'EUR','20090615', 1.161);  
INSERT INTO rate (id, code, date, value) values (3,'EUR','20090617', 1.177);  
INSERT INTO rate (id, code, date, value) values (4,'USD','20090605', 1.625);  
INSERT INTO rate (id, code, date, value) values (5,'USD','20090615', 1.639);  
INSERT INTO rate (id, code, date, value) values (6,'USD','20090617', 1.644);
```

--создание таблицы calendar

```
CREATE TABLE Calendar ( dDay date not null);
```

--генерация значений в таблицу calendar за период с 05.06.2009-18.06.2009

```
INSERT INTO calendar (dday)  
VALUES (generate_series('2009-06-05'::date, '2009-06-18 12:00', '1 day'));
```

--значение курса по каждой валюте за каждый день в таблице calendar

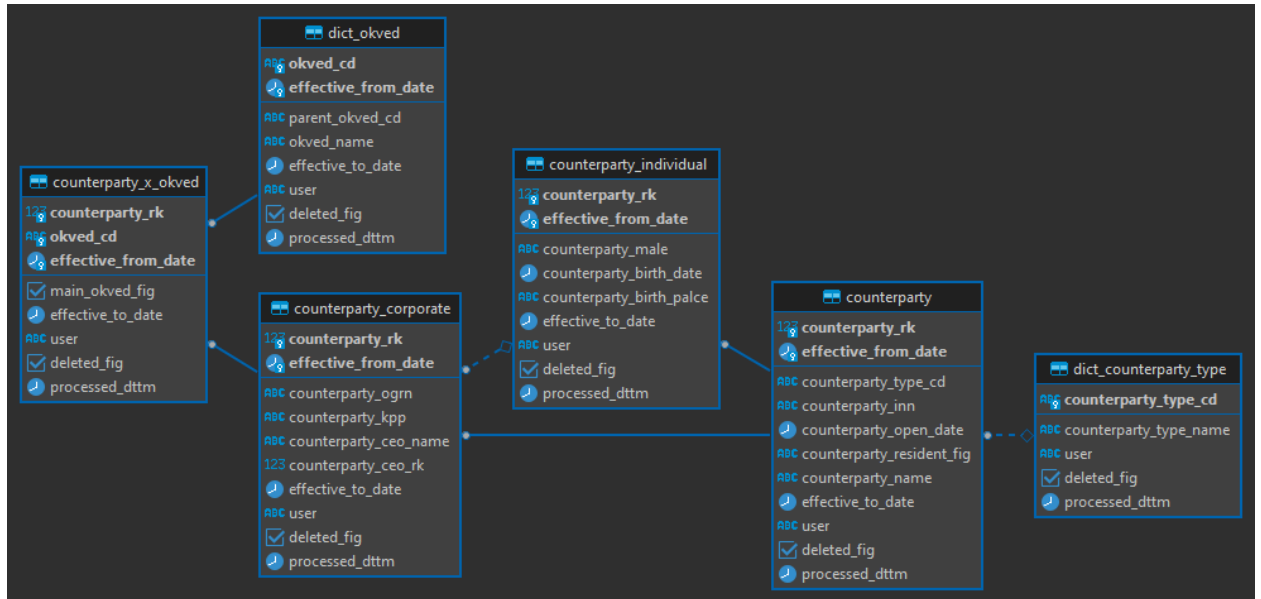
```
SELECT m_v.code AS currency, m_v.dday AS date, r.value  
FROM rate r, (  
    SELECT rr.code, c.dday, max(rr.date) as cur_date  
    FROM rate rr, calendar c  
    WHERE rr.date<=c.dday  
    GROUP BY rr.code, c.dday) AS m_v  
WHERE m_v.cur_date=r.date AND m_v.code=r.code  
ORDER BY date, currency;
```

Задание №2

Создание БД

Перечень полей для задания 2:	
Поле	Описание
Тип клиента	Клиентами банка могут быть физические лица, юридические лица, включая индивидуальных предпринимателей
Наименование клиента	Наименование клиента
ИНН	ссылка
Дата оформления клиента	Дата оформления клиента
Признак резидента	ссылка
ОГРН	ссылка
КПП	ссылка
ФИО генерального директора	ФИО генерального директора, при этом, если он является клиентом банка, то необходима ссылка на него
ОКВЭД	ссылка
Пол	Пол
Дата рождения	Дата рождения
Место рождения	Место рождения

Решение



```
create type resident as enum ('resident', 'not_resident');
```

```
create table counterpart (
    counterpart_rk int unique not null,
    counterparty_type_cd text,
    counterparty_inn text,
    counterparty_open_date date,
    counterparty_resident_fig resident,
    counterparty_name text,
    effective_from_date date not null,
    effective_to_date date not null,
    "user" text not null,
    deleted_fig bool not null,
    processed_dttm timestamp not null,
    primary key (counterparty_rk, effective_from_date)
);
```

```
create table dict_counterparty_type (
    counterparty_type_cd text unique not null,
    counterparty_type_name text not null,
```

```
    "user" text not null,  
    deleted_fig bool not null,  
    processed_dttm timestamp not null,  
    primary key (counterparty_type_cd)  
);
```

```
alter table counterparty  
add constraint counterparty_type_cd  
foreign key (counterparty_type_cd)  
references dict_counterparty_type (counterparty_type_cd)  
on delete cascade;
```

```
create table counterparty_corporate (  
    counterparty_rk int unique not null,  
    counterparty_ogrn text,  
    counterparty_kpp text,  
    counterparty_ceo_name text,  
    counterparty_ceo_rk int,  
    effective_from_date date not null,  
    effective_to_date date not null,  
    "user" text not null,  
    deleted_fig bool not null,  
    processed_dttm timestamp not null,  
    primary key (counterparty_rk, effective_from_date),  
    foreign key (counterparty_rk) references counterparty (counterparty_rk) on delete cascade  
);
```

```
create table counterparty_x_okved (  
    counterparty_rk int not null,  
    okved_cd text not null,  
    main_okved_fig bool not null,  
    effective_from_date date not null,
```

```
effective_to_date date not null,  
"user" text not null,  
deleted_fig bool not null,  
processed_dttm timestamp not null,  
primary key (counterparty_rk, okved_cd, effective_from_date),  
foreign key (counterparty_rk) references counterparty_corporate (counterparty_rk) on  
delete cascade  
);
```

```
create table dict_okved (  
    okved_cd text unique not null,  
    parent_okved_cd text,  
    okved_name text not null,  
    effective_from_date date not null,  
    effective_to_date date not null,  
    "user" text not null,  
    deleted_fig bool not null,  
    processed_dttm timestamp not null,  
    primary key (okved_cd, effective_from_date)  
);
```

```
alter table counterparty_x_okved  
add constraint okved_cd  
foreign key (okved_cd)  
references dict_okved (okved_cd)  
on delete cascade;
```

```
create type male as enum ('man', 'woman');
```

```
create table counterparty_individual (  
    counterparty_rk int unique not null,  
    counterparty_male male,
```

```
counterparty_birth_date date,  
counterparty_birth_palce text,  
effective_from_date date not null,  
effective_to_date date not null,  
"user" text not null,  
deleted_fig bool not null,  
processed_dttm timestamp not null,  
primary key (counterparty_rk, effective_from_date),  
foreign key (counterparty_rk) references counterparty (counterparty_rk) on delete cascade  
);
```

```
alter table counterparty_corporate  
add constraint counterparty_ceo_rk  
foreign key (counterparty_ceo_rk)  
references counterparty_individual (counterparty_rk)  
on delete cascade;
```