# Задача №8

## Контекст

В схеме `public` имеется таблица с историей криптовалюты.

Postgresql:

CREATE TABLE IF NOT EXISTS coins(

**dt** VARCHAR(16),

**avg\_price** NUMERIC,

**tx\_cnt** NUMERIC,

**tx\_vol** NUMERIC,

**active\_addr\_cnt** NUMERIC,

**symbol** VARCHAR(8),

**full\_nm** VARCHAR(128),

**open\_price** NUMERIC,

**high\_price** NUMERIC,

**low\_price** NUMERIC,

**close\_price** NUMERIC,

**vol** NUMERIC,

**market** NUMERIC

)

Описание значений, хранящихся в колонках:

\* `dt` — дата измерений (гггг-мм-дд)

\* `avg\_price` — средняя цена монеты за торговый день в USD

\* `tx\_cnt` — количество транзакций в сети данной монеты

\* `tx\_vol` — объем монет переведенных между адресами в сети данной монеты

\* `active\_addr\_cnt` — количество адресов совершавших а данный день транзакции в сети данной монеты

\* `symbol` — сокращенное название монеты

\* `full\_nm` — полное название монеты

\* `open\_price` — цена монеты в начале торгов данного дня в USD

\* `high\_price` — самая высокая цена данной монеты в течение данного торгового дня в USD

\* `low\_price` — самая низкая цена данной монеты в течение данного торгового дня в USD

\* `close\_price` — цена монеты в конце торгов данного дня в USD

\* `vol` — объем торгов данной монетой на биржах в данный день

\* `market` — капитализация данной монеты в данный день в USD

## Постановка

Вывести для каждой криптомонеты день, когда значение цены (`high\_price`) было самым высоким. Упорядочить результат по цене в порядке убывания, а в случае равенства цен упорядочить по имени монеты. Если имеется несколько дней с таким значением, то выбираем самый ранний. Имя криптовалюты ожидается в верхнем регистре.

## Ожидаемый формат ответа

Ваш запрос должен возвращать таблицу формата:

| full\_name | dt | price |

|-----------|------------|---------|

| BITCOIN | 2018-12-31 | 60000.0 |