# Задача №3

### Контекст

В PostgreSQL существуют такие вещи как *партиции таблиц*. Ознакомиться с тем что это такое и как это употреблять можно [здесь](https://gitlab.atp-fivt.org/courses-public/db2023-supplementary/global/-/tree/main/practice/seminars/11-olap#11-partitioning-%D0%BF%D0%B0%D1%80%D1%82%D0%B8%D1%86%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D1%81%D0%B5%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). Достаточно посмотреть [пункт 1.1.1](https://gitlab.atp-fivt.org/courses-public/db2023-supplementary/global/-/tree/main/practice/seminars/11-olap#111-%D0%BE%D0%B1%D1%89%D0%B8%D0%B5-%D1%81%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F), но можно и больше :)

### Постановка

Создать представление v\_first\_level\_partition\_info(parent\_schema, parent\_table, child\_schema, child\_table), которая агрегирует информацию о партициях первого уровня (непосредственные партиции). Скрипт создания должен быть нечувствителен к уже объявленным представлениям.

Для решения вам потребуются системные таблицы: pg\_inherits, pg\_class, pg\_namespace.

**Пример**

Имея таблицу:

CREATE TABLE people\_partitioned (  
 person\_id SERIAL,  
 first\_name VARCHAR(128) NOT NULL,  
 last\_name VARCHAR(128) NOT NULL,  
 birthday DATE NOT NULL  
) PARTITION BY RANGE (birthday);  
  
  
CREATE TABLE people\_partitioned\_birthdays\_1800\_to\_1850  
 PARTITION OF people\_partitioned  
 FOR VALUES FROM ('1800-01-01') TO ('1849-12-31');  
  
CREATE TABLE people\_partitioned\_birthdays\_1850\_to\_1900  
 PARTITION OF people\_partitioned  
 FOR VALUES FROM ('1850-01-01') TO ('1899-12-31');  
  
CREATE TABLE people\_partitioned\_birthdays\_1900\_to\_1950  
 PARTITION OF people\_partitioned  
 FOR VALUES FROM ('1900-01-01') TO ('1949-12-31');  
  
CREATE TABLE people\_partitioned\_birthdays\_1950\_to\_2000  
 PARTITION OF people\_partitioned  
 FOR VALUES FROM ('1950-01-01') TO ('1999-12-31');  
  
CREATE TABLE people\_partitioned\_birthdays\_2000\_to\_2050  
 PARTITION OF people\_partitioned  
 FOR VALUES FROM ('2000-01-01') TO ('2049-12-31');

Пример запроса к нашему представлению:

SELECT  
 child\_schema,  
 child\_table  
FROM  
 v\_first\_level\_partition\_info  
WHERE 1=1  
 AND parent\_schema = 'public'  
 AND parent\_table = 'people\_partitioned'  
ORDER BY  
 child\_schema,  
 child\_table;  
---  
 ----------------------------------------------------------  
| child\_schema | child\_table |  
|--------------|-------------------------------------------|  
| public | people\_partitioned\_birthdays\_1800\_to\_1850 |  
| public | people\_partitioned\_birthdays\_1850\_to\_1900 |  
| public | people\_partitioned\_birthdays\_1900\_to\_1950 |  
| public | people\_partitioned\_birthdays\_1950\_to\_2000 |  
| public | people\_partitioned\_birthdays\_2000\_to\_2050 |  
 ----------------------------------------------------------

### Ожидаемый формат ответа

Выводить ничего не надо. Скрипт с решением должен содержать **только** объявление представления.