-- Создадим таблицу SALARY

**CREATE** **TABLE** SALARY ('EMP\_ID (PK)' **INT**,

'PERIOD (PK)' **DATE**,

amount **INT**) ;

-- Наполним ее исходными данными

**INSERT** **INTO** SALARY ('EMP\_ID (PK)','PERIOD (PK)',amount)

**VALUES** (1,201901,1000),(1,201902,1300), (1,201903,1200), (1,201904,1000),

(1,201905,700), (1,201906,1800), (1,201907,1100), (2,201901,700),

(2,201902,720), (2,201903,740), (2,201904,700), (2,201905,710),

(2,201906,715), (2,201907,759), (3,201901,600), (3,201902,620),

(3,201903,640), (3,201904,600), (3,201905,610), (3,201906,615),

(3,201907,630), (4,201901,740), (4,201902,730), (4,201903,720),

(4,201904,700), (4,201905,720), (4,201906,715), (4,201907,743),

(5,201901,400), (5,201902,420), (5,201903,440), (5,201904,400),

(5,201905,410), (5,201906,415), (5,201907,430), (6,201901,310),

(6,201902,320), (6,201903,330), (6,201904,315), (6,201905,310),

(6,201906,330), (6,201907,340), (7,201901,370), (7,201902,300),

(7,201903,310), (7,201904,345), (7,201905,330), (7,201906,310) ;

-- Поменяем названия колонок, т.к. со скобками неудобно работать

**ALTER** **TABLE** SALARY

**RENAME** **COLUMN** 'EMP\_ID (PK)' **TO** EMP\_ID\_PK;

**ALTER** **TABLE** SALARY

**RENAME** **COLUMN** 'PERIOD (PK)' **TO** PERIOD\_PK;

-- Дату приведем к привычному виду

**UPDATE** SALARY

**SET** PERIOD\_PK = **DATE**(**SUBSTRING**(PERIOD\_PK,0,5)||'-'||**SUBSTRING**(PERIOD\_PK,5)||'-01')

-- Создадим таблицу EMPLOYER

**CREATE** **TABLE** EMPLOYER ('EMP\_ID\_(PK)' **INT**,

'DEP\_ID\_(FK,\_NULLABLE)' **INT**,

'EMP\_NAME\_(Not\_null,\_unique)' **CHAR**,

'POSITION\_(Not\_null)' **CHAR**) ;

**INSERT** **INTO** EMPLOYER ('EMP\_ID\_(PK)','DEP\_ID\_(FK,\_NULLABLE)','EMP\_NAME\_(Not\_null,\_unique)','POSITION\_(Not\_null)')

**VALUES**

('1',**null**,'Добрынин Дмитрий Данилович','Директор'),

('2','1','Абрамова Анна Анатольевна','Главный бухгалтер'),

('3','1','Богданов Борис Бедросович','Бухгалтер'),

('4','2','Воробьева Вера Викторовна','Начальник отдела'),

('5','3','Гаврилова Галина Георгиевна','Главный специалист'),

('6','3','Ежов Евгений Егорович','Специалист'),

('7','3','Иванов Игорь Ильич','Специалист');

-- Поменяем названия колонок, т.к. со скобками неудобно работать

**ALTER** **TABLE** EMPLOYER

**RENAME** **COLUMN** 'EMP\_ID\_(PK)' **TO** EMP\_ID\_PK;

**ALTER** **TABLE** EMPLOYER

**RENAME** **COLUMN** 'DEP\_ID\_(FK,\_NULLABLE)' **TO** DEP\_ID\_FK;

**ALTER** **TABLE** EMPLOYER

**RENAME** **COLUMN** 'EMP\_NAME\_(Not\_null,\_unique)' **TO** EMP\_NAME;

**ALTER** **TABLE** EMPLOYER

**RENAME** **COLUMN** 'POSITION\_(Not\_null)' **TO** POSITION\_;

-- Создадим таблицу DEPARTMENT

**CREATE** **TABLE** DEPARTMENT ('DEP\_ID\_(PK)' **INT**,

'DEP\_NAME\_(Not\_null,\_unique)' **CHAR**) ;

**INSERT** **INTO** DEPARTMENT ('DEP\_ID\_(PK)','DEP\_NAME\_(Not\_null,\_unique)')

**VALUES**

('1','Бухгалтерия'),

('2','HR'),

('3','Продажи') ;

-- Поменяем названия колонок, т.к. со скобками неудобно работать

**ALTER** **TABLE** DEPARTMENT

**RENAME** **COLUMN** 'DEP\_ID\_(PK)' **TO** DEP\_ID\_PK;

**ALTER** **TABLE** DEPARTMENT

**RENAME** **COLUMN** 'DEP\_NAME\_(Not\_null,\_unique)' **TO** DEP\_NAME;

**Примечание:**

Не совсем понятно под какую именно СУБД были написаны запросы для подготовки таблиц, поэтому при подготовке БД переформатировал всё под PostgreSQL. В некоторых запросах ниже синтаксис может (но теоретически не должен) отличаться от ожидаемого.

Запросы тестировались на последней официальной версии докер-образа postgres

-- a) Написать запрос, который формирует ведомость за май 2019

-- (ФИО сотрудника, отдел, должность, сумма выплат за май).

**SELECT e.EMP\_NAME AS "ФИО сотрудника",**

**d.DEP\_NAME AS "Отдел",**

**e.POSITION\_ AS "Должность",**

**s.amount AS "Сумма выплат за май"**

**FROM employer e**

**LEFT JOIN salary s ON e.EMP\_ID\_PK = s.EMP\_ID\_PK**

**LEFT JOIN department d ON e.DEP\_ID\_FK = d.DEP\_ID\_PK**

**WHERE s.PERIOD\_PK = TO\_DATE(201905::TEXT, 'YYYYMM');**

b) Будут ли отличаться результаты этих двух запросов?

-- Если будут, то как исправить второй запрос, чтобы результат был такой же, как у первого?

select e.emp\_name,

(select dep\_name from department d where d.DEP\_ID\_PK = e.DEP\_ID\_FK) dep\_name,

e.POSITION\_

from EMPLOYER e;

select e.emp\_name, d.dep\_name, e.POSITION\_

from employer e, department d

where d.DEP\_ID\_PK = e.DEP\_ID\_FK ;

Второй запрос не выведет сотрудника с ID 1 из-за значения null в DEP\_ID\_FK. Null-значение ничему никогда не равно.

Можно заменить условие на left join. За исключением сбившейся сортировки выводы будут одинаковы. (Если сортировка важна – можно восстановить добавлением в конец ORDER BY e.EMP\_ID\_PK;)

**select e.emp\_name, d.dep\_name, e.POSITION\_**

**from employer e**

**LEFT JOIN department d on d.DEP\_ID\_PK=e.DEP\_ID\_FK;**

-- с) Написать запрос, который выводит список сотрудников,

-- заработавших за 1-ый квартал более 2000.

**SELECT e.EMP\_NAME**

**FROM EMPLOYER e**

**JOIN SALARY s ON e.EMP\_ID\_PK=s.EMP\_ID\_PK**

**WHERE s.PERIOD\_PK BETWEEN TO\_DATE('201901', 'YYYYMM') AND TO\_DATE('201903', 'YYYYMM')**

**GROUP BY e.EMP\_NAME**

**HAVING SUM(s.amount)>2000;**

-- d) Написать запрос, который будет возвращать доходы сотрудников за год

-- нарастающим итогом

-- 1) Через оконные функции

**SELECT e.EMP\_NAME,**

**s.PERIOD\_PK,**

**s.amount,**

**SUM(s.amount) OVER (PARTITION BY e.EMP\_ID\_PK ORDER BY s.PERIOD\_PK) AS total**

**FROM EMPLOYER e**

**JOIN SALARY s ON s.EMP\_ID\_PK=e.EMP\_ID\_PK;**

-- 2) Через подзапрос

-- После выражения FROM пишется таблица из БД или подзапрос, в котором уже

-- могут быть использованы какие-либо функции

**SELECT e.EMP\_NAME,**

**s.PERIOD\_PK,**

**s.amount,**

**(SELECT SUM(s2.amount) FROM SALARY s2 WHERE s2.EMP\_ID\_PK = e.EMP\_ID\_PK AND s2.PERIOD\_PK <= s.PERIOD\_PK) AS total**

**FROM EMPLOYER e**

**JOIN SALARY s ON s.EMP\_ID\_PK = e.EMP\_ID\_PK;**