

Задания по SQL

От простого, к сложному:

1. Может ли в Oracle функция стоять в выражении FROM?

ОТВЕТ:

Да, если написанная на PL/SQL функция возвращает коллекцию. В таком случае функцию можно вызывать, обращаясь к её результату как к реляционному набору данных:

```
SELECT * FROM TABLE(function_name(...))
```

2. Исходные данные:

DEPARTMENT

DEP_ID (PK)	DEP_NAME (Not null, unique)
1	Бухгалтерия
2	HR
3	Продажи

EMPLOYER

EMP_ID (PK)	DEP_ID (FK, NULLABLE)	EMP_NAME (Not null, unique)	POSITION (Not null)
1		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

SALARY

EMP_ID (PK)	PERIOD (PK)	AMOUNT
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
7	201907	310

- А. Написать запрос, который формирует ведомость за май 2019 (ФИО сотрудника, отдел, должность, сумма выплат за май).

ОТВЕТ:

```
select
    emp_name,
    coalesce(dep_name, '-') as dep_name,
    position,
    amount
from employer e
left join department d on e.dep_id = d.dep_id
left join salary s on e.emp_id = s.emp_id
where period = 201905
order by emp_name;
```

employer(+) 1

select emp_name, coalesce(dep_name, '-') as c | Введите SQL выражение чтобы отфильтровать

	ABC emp_name	ABC dep_name	ABC position	123 amount
1	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	710
2	Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	610
3	Воробьева Вера Викторовна	HR	Начальник отдела	720
4	Гаврилова Галина Георгиевна	Продажи	Главный специалист	410
5	Добрынин Дмитрий Данилович	-	Директор	700
6	Ежов Евгений Егорович	Продажи	Специалист	310
7	Иванов Игорь Ильич	Продажи	Специалист	330

- В. 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 = e.dep_id) 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 = e.dep_id;
```

ОТВЕТ:

Результаты второго запроса не будут включать данные по тем сотрудникам, для которых не определено подразделение (dep_id). Чтобы второй запрос выдавал полные данные, исправить его можно так:

```
select e.emp_name, d.dep_name, e.position
from employer e
left join department d
on d.dep_id = e.dep_id;
```

- С. Написать запрос, который выводит список сотрудников, заработавших за 1-ый квартал более 2000.

ОТВЕТ:

Вариант 1:

```
with salary_1q as (
    select emp_id, sum(amount) as amount
    from salary
    where period in (201901, 201902, 201903)
    group by emp_id
)
select
    emp_name,
    coalesce(dep_name, '-') as dep_name,
    position,
    amount
from employer e
left join department d on e.dep_id = d.dep_id
left join salary_1q s on e.emp_id = s.emp_id
where amount > 2000
order by emp_name;
```

employer 1

with salary_1q as (select emp_id, sum(amount) Введите SQL выражение чтобы отфильтров

	emp_name	dep_name	position	amount
1	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	2 160
2	Воробьева Вера Викторовна	HR	Начальник отдела	2 190
3	Добрынин Дмитрий Данилович	-	Директор	3 500

Вариант 2:

```
with salary_1q as (
    select emp_id, sum(amount) as amount
    from salary
    where period >= 201901 and period <= 201903
    group by emp_id
    having sum(amount) > 2000
)
select
    emp_name,
    coalesce(dep_name, '-') as dep_name,
    position,
    amount
from employer e
left join department d on e.dep_id = d.dep_id
join salary_1q s on e.emp_id = s.emp_id
order by amount desc;
```

employer 1

with salary_1q as (select emp_id, sum(amount) Введите SQL выражение чтобы отфильтров

	emp_name	dep_name	position	amount
1	Добрынин Дмитрий Данилович	-	Директор	3 500
2	Воробьева Вера Викторовна	HR	Начальник отдела	2 190
3	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	2 160

- D. Написать запрос, который будет возвращать доходы сотрудников за год нарастающим итогом в следующем виде:

EMP_NAME	DEP_NAME	POSITION	PERIOD	RUNNING_SUM
Добрынин Дмитрий Данилович		Директор	201901	1000
Добрынин Дмитрий Данилович		Директор	201902	2300
Добрынин Дмитрий Данилович		Директор	201903	3500
Добрынин Дмитрий Данилович		Директор	201904	4500
Добрынин Дмитрий Данилович		Директор	201905	5200
Добрынин Дмитрий Данилович		Директор	201906	7000
Добрынин Дмитрий Данилович		Директор	201907	8100
Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201901	700
Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201902	1420
Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201903	2160
Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201904	2860
Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201905	3570
Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201906	4285
Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201907	5044
Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	201901	600
Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	201902	1220
Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	201903	1860
Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	201904	2460
Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	201905	3070
Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	201906	3685
Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	201907	4315
Воробьева Вера Викторовна	HR	Начальник отдела	201901	740
Воробьева Вера Викторовна	HR	Начальник отдела	201902	1470
Воробьева Вера Викторовна	HR	Начальник отдела	201903	2190
Воробьева Вера Викторовна	HR	Начальник отдела	201904	2890
Воробьева Вера Викторовна	HR	Начальник отдела	201905	3610
Воробьева Вера Викторовна	HR	Начальник отдела	201906	4325
Воробьева Вера Викторовна	HR	Начальник отдела	201907	5068
Гаврилова Галина Георгиевна	Продажи	Главный специалист	201901	400

Гаврилова Галина Георгиевна	Продажи	Главный специалист	201902	820
Гаврилова Галина Георгиевна	Продажи	Главный специалист	201903	1260
Гаврилова Галина Георгиевна	Продажи	Главный специалист	201904	1660
Гаврилова Галина Георгиевна	Продажи	Главный специалист	201905	2070
Гаврилова Галина Георгиевна	Продажи	Главный специалист	201906	2485
Гаврилова Галина Георгиевна	Продажи	Главный специалист	201907	2915
Ежов Евгений Егорович	Продажи	Специалист	201901	310
Ежов Евгений Егорович	Продажи	Специалист	201902	630
Ежов Евгений Егорович	Продажи	Специалист	201903	960
Ежов Евгений Егорович	Продажи	Специалист	201904	1275
Ежов Евгений Егорович	Продажи	Специалист	201905	1585
Ежов Евгений Егорович	Продажи	Специалист	201906	1915
Ежов Евгений Егорович	Продажи	Специалист	201907	2255
Иванов Игорь Ильич	Продажи	Специалист	201901	370
Иванов Игорь Ильич	Продажи	Специалист	201902	670
Иванов Игорь Ильич	Продажи	Специалист	201903	980
Иванов Игорь Ильич	Продажи	Специалист	201904	1325
Иванов Игорь Ильич	Продажи	Специалист	201905	1655
Иванов Игорь Ильич	Продажи	Специалист	201906	1965
Иванов Игорь Ильич	Продажи	Специалист	201907	2275

ОТВЕТ:

```
select
  emp_name,
  coalesce(dep_name, '') as dep_name,
  position,
  period,
  sum(amount) over (partition by e.emp_id order by period) as running_sum
from employer e
left join department d on e.dep_id = d.dep_id
left join salary s on e.emp_id = s.emp_id
order by e.emp_id, period;
```

employer(+) 1

select emp_name, coalesce(dep_name, "") as de Введите SQL выражение чтобы отфильтровать результаты

	abc emp_name	abc dep_name	abc position	123 period	123 running_sum
1	Добрынин Дмитрий Данилович		Директор	201901	1000
2	Добрынин Дмитрий Данилович		Директор	201902	2300
3	Добрынин Дмитрий Данилович		Директор	201903	3500
4	Добрынин Дмитрий Данилович		Директор	201904	4500
5	Добрынин Дмитрий Данилович		Директор	201905	5200
6	Добрынин Дмитрий Данилович		Директор	201906	7000
7	Добрынин Дмитрий Данилович		Директор	201907	8100
8	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201901	700
9	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201902	1420
10	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201903	2160
11	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201904	2860
12	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201905	3570
13	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201906	4285
14	Абрамова Анна Анатольевна	Бухгалтерия	Главный бухгалтер	201907	5044
15	Богданов Борис Бедросович	Бухгалтерия	Бухгалтер	201901	600

Сложное задание. Исходные данные.

Дана таблица с ребрами простого неориентированного графа, где для каждого ребра графа указан вес. С помощью одного SQL запроса вывести путь между двумя вершинами графа с минимальным весом. Если для двух путей совпадает вес, то вывести с минимальной длиной пути. Если совпадает и путь, и длина вывести первый попавшийся.

Таблица GRAPH. Уникальный индекс UI_GRAPH_1 on GRAPH (LEAST(p1,p2), GREATEST(p1,p2));

P1	P2	WEIGHT
1	2	1
2	3	1
3	4	5
4	5	1
5	6	1
7	6	1
1	8	1
7	11	1
3	7	10
3	9	2
7	9	2

ОТВЕТ:

Привожу решение для PostgreSQL с применением рекурсивных запросов.

В качестве примера здесь вычислен минимальный путь из вершины 2 в вершину 11.

```
with recursive make_path(last_vertix, path_steps, iterations, path_length) as (  
  -- начинаем движение из вершины S  
  values (2, array[2::smallint], 1, 0) --<< ЗДЕСЬ ЗАДАТЬ НАЧАЛЬНУЮ ВЕРШИНУ  
  union all  
  select  
    g.p2 as last_vertix,  
    p.path_steps || array[g.p2], -- добавляем вершину в путь  
    p.iterations + 1, -- счётчик итераций рекурсии  
    p.path_length + g.weight -- считаем длину пути  
  from  
    (select p1, p2, weight from graph  
     union  
     select p2, p1, weight from graph) as g,  
    -- здесь учитываем, что граф неориентированный  
    make_path p  
  where g.p1 = p.last_vertix  
         and p.iterations <=   
           (select count(*) from (  
             select p1 from graph union select p2 from graph  
           ) all_vertices)  
         -- чтобы не уходить в бесконечную рекурсию: ограничим число итераций, не должно превышать число вершин в графе  
         and not g.p2 = any(p.path_steps)  
         -- убираем маршруты с повторением вершин (защипыванием)  
)  
,  
my_paths as (  
  select path_steps, path_length, iterations  
  from make_path  
  -- выбираем только маршруты, приводящие в вершину T  
  where last_vertix = 11 --<< ЗДЕСЬ ЗАДАТЬ КОНЕЧНУЮ ВЕРШИНУ  
  -- выбираем кратчайший путь  
  order by path_length, iterations  
)  
select *  
from my_paths  
limit 1
```

ультат 1		
th recursive make_path(last_vertix, path_s		
path_steps	123	path_length
{2,3,9,7,11}	6	5