**1. Вывести все директории в виде:**

**ID, Название, Путь до корня**

WITH RECURSIVE directory\_tree AS (

SELECT

id,

name,

parent\_id,

name::varchar(2000) AS path\_to\_root

FROM

public.file\_system

WHERE

type = 'DIR' AND parent\_id IS NULL

UNION ALL

SELECT

fs.id,

fs.name,

fs.parent\_id,

CONCAT(dt.path\_to\_root, '/', fs.name)::varchar(2000) AS path\_to\_root

FROM

public.file\_system AS fs

INNER JOIN

directory\_tree AS dt

ON

fs.parent\_id = dt.id

WHERE

fs.type = 'DIR'

)

SELECT

id,

name,

path\_to\_root

FROM

directory\_tree

ORDER BY

path\_to\_root;



**3. Для каждой директории посчитать объем занимаемого места на диске (с учетом всех вложенных папок)**

**ID, Название, Путь до корня, total\_size**

WITH RECURSIVE directory\_tree AS (

SELECT

id,

name,

parent\_id,

name::varchar(2000) AS path\_to\_root,

file\_size,

CASE

WHEN type = 'DIR' THEN id

ELSE NULL

END AS directory\_id

FROM

public.file\_system

WHERE

parent\_id IS NULL

UNION ALL

SELECT

fs.id,

fs.name,

fs.parent\_id,

CONCAT(dt.path\_to\_root, '/', fs.name)::varchar(2000) AS path\_to\_root,

fs.file\_size,

CASE

WHEN fs.type = 'DIR' THEN fs.id

ELSE dt.directory\_id

END AS directory\_id

FROM

public.file\_system AS fs

INNER JOIN

directory\_tree AS dt

ON

fs.parent\_id = dt.id

)

SELECT

directory\_id AS id,

MAX(name) AS name,

MAX(path\_to\_root) AS path\_to\_root,

SUM(file\_size) AS total\_size

FROM

directory\_tree

WHERE

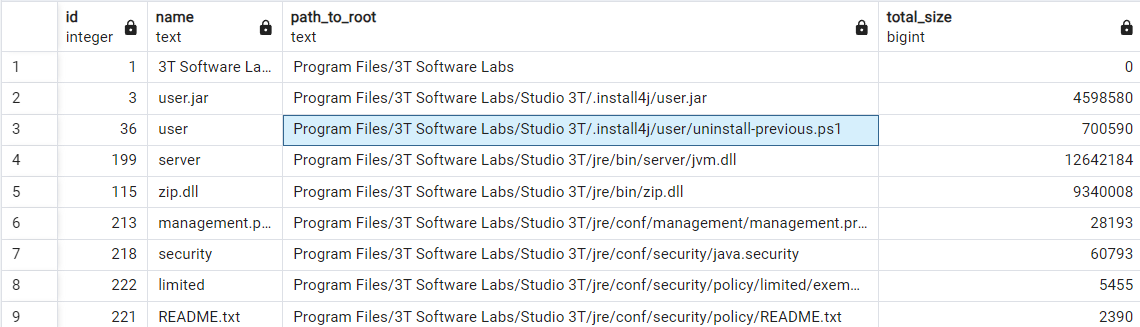
directory\_id IS NOT NULL

GROUP BY

directory\_id

ORDER BY

path\_to\_root;



**4. Добавить в запрос: сколько процентов директория занимает места относительно всех среди своих соседей (siblings)**

**ID, Название, Путь до корня, total\_size, ratio**

WITH RECURSIVE directory\_tree AS (

SELECT

id,

name,

parent\_id,

name::varchar(2000) AS path\_to\_root,

file\_size,

CASE

WHEN type = 'DIR' THEN id

ELSE NULL

END AS directory\_id

FROM

public.file\_system

WHERE

parent\_id IS NULL

UNION ALL

SELECT

fs.id,

fs.name,

fs.parent\_id,

CONCAT(dt.path\_to\_root, '/', fs.name)::varchar(2000) AS path\_to\_root,

fs.file\_size,

CASE

WHEN fs.type = 'DIR' THEN fs.id

ELSE dt.directory\_id

END AS directory\_id

FROM

public.file\_system AS fs

INNER JOIN

directory\_tree AS dt

ON

fs.parent\_id = dt.id

)

, directory\_sizes AS (

SELECT

directory\_id AS id,

MAX(name) AS name,

MAX(path\_to\_root) AS path\_to\_root,

parent\_id,

SUM(file\_size) AS total\_size

FROM

directory\_tree

WHERE

directory\_id IS NOT NULL

GROUP BY

directory\_id, parent\_id

)

SELECT

ds.id,

ds.name,

ds.path\_to\_root,

ds.total\_size,

ROUND((ds.total\_size \* 100.0) / NULLIF(SUM(ds.total\_size) OVER (PARTITION BY ds.parent\_id), 0), 2) AS ratio

FROM

directory\_sizes AS ds

ORDER BY

ds.path\_to\_root;

