Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут”
Кафедра АСОІУ

Лабораторна робота № 3
з дисципліни
“Бази даних - 2. Програмні додатки з використанням баз даних”
Тема: Ієрархічні запити

Прийняв:
Клименко О. М.

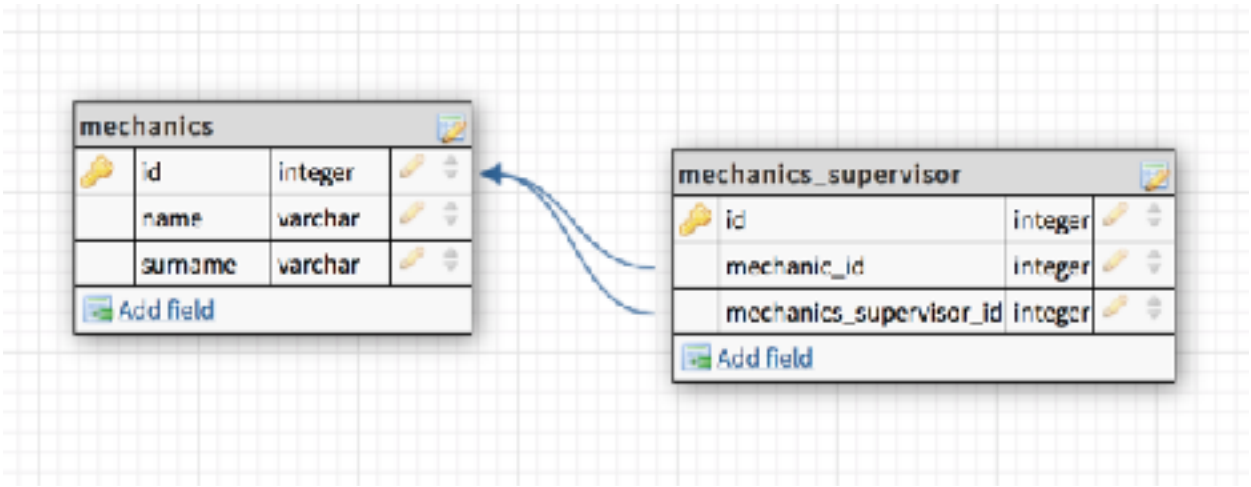
Виконав:
студент 3-го курсу
гр. ІП-52 ФІОТ
Набоков Едуард
Максимович

Київ 2017

Постановка задачі:

1) Вивести список всіх « наща дків » вказан ого « предка » .
2) Вивести список всіх « пре дків » вказан ого « нащадка » .
3) Вивести список, другий полем якого є « рівень » (аналог псевдо стовпчика level в connect by).
4) (2 запити) Змінити дані в доданій таблиці так, щоб утворився цикл. Написати запит, що видає помилку при зациклюванні. Змінити цей запит так, що б помилки не було.
5) Для всіх « нащадків » (це перше поле : Іванов) вивести список « пре дків » через « / » , де останнім в ланцюгу є цей « нащадок » (це друге поле: Іванен ко /Іванченко/Іванчук /Іванов)

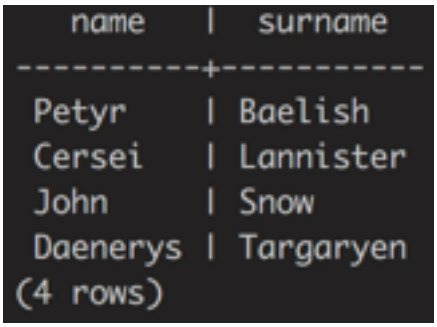
Схема бази данних:



Запити:

```
CREATE TABLE mechanics (
  id SERIAL PRIMARY KEY,
  name VARCHAR(10),
  surname VARCHAR(10)
);

CREATE TABLE mechanics_supervisor (
  id SERIAL PRIMARY KEY,
  mechanic_id INT REFERENCES mechanics(id),
  mechanics_supervisor_id INT REFERENCES mechanics(id)
);


INSERT INTO mechanics (name, surname) VALUES ('Daenerys',
'Targaryen');
INSERT INTO mechanics (name, surname) VALUES ('John', 'Snow');
INSERT INTO mechanics (name, surname) VALUES ('Cersei', 'Lannister');
INSERT INTO mechanics (name, surname) VALUES ('Petyr', 'Baelish');
INSERT INTO mechanics (name, surname) VALUES ('Tyrion', 'Lannister');
INSERT INTO mechanics (name, surname) VALUES ('Sandor', 'Clegane');
INSERT INTO mechanics (name, surname) VALUES ('Eddard', 'Stark');
INSERT INTO mechanics (name, surname) VALUES ('Sansa', 'Stark');
INSERT INTO mechanics (name, surname) VALUES ('Joffrey', 'Baratheon');
INSERT INTO mechanics (name, surname) VALUES ('Lord', 'Varys');
INSERT INTO mechanics (name, surname) VALUES ('Grey', 'Worm');
INSERT INTO mechanics (name, surname) VALUES ('Samwell', 'Tarly');

INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (1, 2);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (2, 3);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (3, 4);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (3, 11);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (4, 11);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (4, 12);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (5, 11);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (5, 6);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (6, 7);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (7, 8);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (6, 9);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (6, 10);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (5, 11);
INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (5, 11);
```

```sql
-- TASK 1

WITH RECURSIVE sub(mechanic_id) AS
  (SELECT mechanic_id FROM mechanics_supervisor WHERE
mechanics_supervisor_id = (SELECT id FROM mechanics WHERE
name='Samwell')
  UNION ALL
  SELECT mechanics_supervisor.mechanic_id FROM sub
  INNER JOIN mechanics_supervisor ON
mechanics_supervisor.mechanics_supervisor_id = sub.mechanic_id
  )

  SELECT mechanics.name, mechanics.surname FROM sub
  JOIN mechanics ON mechanics.id = sub.mechanic_id;
```



```
   name    |  surname
-----------+-----------
 Petyr     | Baelish
 Cersei    | Lannister
 John      | Snow
 Daenerys  | Targaryen
(4 rows)
```

```
-- TASK 2

WITH RECURSIVE sub(mechanics_supervisor_id) AS
    (SELECT mechanics_supervisor_id FROM mechanics_supervisor
WHERE mechanic_id = (SELECT id FROM mechanics WHERE
name='Daenerys')
   UNION ALL
   SELECT mechanics_supervisor.mechanics_supervisor_id FROM sub
   INNER JOIN mechanics_supervisor ON
sub.mechanics_supervisor_id = mechanics_supervisor.mechanic_id
    )

    SELECT mechanics.name, mechanics.surname FROM sub
    JOIN mechanics ON mechanics.id =
sub.mechanics_supervisor_id;
```

```
  name   |  surname
---------+-----------
 John    | Snow
 Cersei  | Lannister
 Petyr   | Baelish
 Grey    | Worm
 Grey    | Worm
 Samwell | Tarly
(6 rows)
```

```
-- TASK 3

WITH RECURSIVE sub(mechanics_supervisor_id, level) AS
     (SELECT mechanics_supervisor_id, 1 FROM mechanics_supervisor
WHERE mechanic_id = (SELECT id FROM mechanics WHERE
name='Daenerys')
      UNION ALL
      SELECT mechanics_supervisor.mechanics_supervisor_id, level
+ 1 FROM sub
      INNER JOIN mechanics_supervisor ON
sub.mechanics_supervisor_id = mechanics_supervisor.mechanic_id
     )

     SELECT mechanics.name, mechanics.surname, level FROM sub
     JOIN mechanics ON mechanics.id = sub.mechanics_supervisor_id
     ORDER BY level;
```

```
  name   |  surname   | level
---------+------------+-------
 John    | Snow       |     1
 Cersei  | Lannister  |     2
 Petyr   | Baelish    |     3
 Grey    | Worm       |     3
 Grey    | Worm       |     4
 Samwell | Tarly      |     4
(6 rows)
```

```
-- TASK 4

INSERT INTO mechanics_supervisor (mechanic_id,
mechanics_supervisor_id) VALUES (12, 1);

WITH RECURSIVE sub(mechanics_supervisor_id, level, path, cycle)
AS
    (SELECT mechanics_supervisor_id, 1,
ARRAY[mechanics_supervisor_id], false FROM mechanics_supervisor
WHERE mechanic_id = (SELECT id FROM mechanics WHERE
name='Daenerys')
     UNION ALL
     SELECT mechanics_supervisor.mechanics_supervisor_id, level
+ 1, path || sub.mechanics_supervisor_id,
sub.mechanics_supervisor_id = ANY(path) FROM sub
     INNER JOIN mechanics_supervisor ON
sub.mechanics_supervisor_id = mechanics_supervisor.mechanic_id
     WHERE NOT cycle
    )

    SELECT mechanics.name, mechanics.surname, level FROM sub
    JOIN mechanics ON mechanics.id = sub.mechanics_supervisor_id
    ORDER BY level;
```

```
  name   |  surname   | level
---------+------------+-------
 John    | Snow       |     1
 Cersei  | Lannister  |     2
(2 rows)
```

```
-- TASK 5

WITH RECURSIVE sub(mechanics_supervisor_id, route, level, path,
cycle) AS
     (SELECT mechanics_supervisor_id, '/' || (SELECT name || ' '
|| surname FROM mechanics where mechanic_id = mechanics.id) ||
'/' || (SELECT name || ' ' || surname FROM mechanics where
mechanics_supervisor_id = mechanics.id), 1,
ARRAY[mechanics_supervisor_id], false FROM mechanics_supervisor
WHERE mechanic_id = (SELECT id FROM mechanics WHERE
name='Sandor')
      UNION ALL
      SELECT mechanics_supervisor.mechanics_supervisor_id,
sub.route || '/' || (SELECT name || ' ' || surname FROM
mechanics limit 1), level + 1, path ||
sub.mechanics_supervisor_id, sub.mechanics_supervisor_id =
ANY(path) FROM sub
      INNER JOIN mechanics_supervisor ON
sub.mechanics_supervisor_id = mechanics_supervisor.mechanic_id
      WHERE NOT cycle
     )

     SELECT mechanics.name || ' ' || mechanics.surname AS NAME,
route FROM sub
     JOIN mechanics ON mechanics.id = sub.mechanics_supervisor_id
     ORDER BY level;
```

```
       name        |                        route
-------------------+-----------------------------------------------
 Eddard Stark      | /Sandor Clegane/Eddard Stark
 Joffrey Baratheon | /Sandor Clegane/Joffrey Baratheon
 Lord Varys        | /Sandor Clegane/Lord Varys
 Sansa Stark       | /Sandor Clegane/Eddard Stark/Daenerys Targaryen
(4 rows)
```