Bazy Danych

Wykład p.t.

Podzapytania.

Zagnieżdżanie zapytań. Podzapytania nieskorelowane i skorelowane

Antoni Ligęza

ligeza@agh.edu.pl
http://galaxy.uci.agh.edu.pl/~ligeza

Wykorzystano materiały:
http:
//www.postgresql.org/docs/8.3/interactive/index.html

Komendy SQL: SELECT

```
Command:     SELECT
Description: retrieve rows from a table or view
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
    * | expression [ AS output_name ] [, ...]
    [ FROM from_item [, ...] ]
    [ WHERE condition ]
    [ GROUP BY expression [, ...] ]
    [ HAVING condition [, ...] ]
    [ { UNION | INTERSECT | EXCEPT }
[ ALL ] select ]
    [ ORDER BY expression
[ ASC | DESC | USING operator ] [, ...] ]
    [ LIMIT { count | ALL } ]
    [ OFFSET start ]
    [ FOR { UPDATE | SHARE }
[ OF table_name [, ...] ] [ NOWAIT ] [...] ]

where from_item can be one of:

    [ ONLY ] table_name [ * ]
[ [ AS ] alias [ ( column_alias [, ...] ) ] ]
    ( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
    function_name ( [ argument [, ...] ] )
[ AS ] alias [ ( column_alias [, ...]
| column_definition [, ...] ) ]
    function_name ( [ argument [, ...] ] )
AS ( column_definition [, ...] )
    from_item [ NATURAL ] join_type from_item
[ ON join_condition | USING ( join_column [, ...] ) ]
```

Podzapytania

Zapytanie SQL może zawierać wewnątrz inne zapytanie SQL. Pamiętamy, że każde zapytanie (`SELECT ...`) może być interpretowane jako żądanie wyliczenia pewnych wartości.

Taka konstrukcja nosi nazwę *zapytania z podzapytaniem*. Mówi się także o *zagnieżdżaniu zapytań*.

Zapytanie główne to zapytanie *nadrzędne* lub *zewnętrzne*.

Zapytanie pomocnicze to zapytanie *podrzędne* lub *wewnętrzne*.

Zapytanie podrzędne zazwyczaj zwraca pojedynczą wartość lub listę wartości wykorzystywanych w zapytaniu zewnętrznym.

Przykład:

Podaj listę pracowników zatrudnionych w lokalizacji 'Mysieoczko'.

Kolejność operacji:

- w tablicy `dzial` odnaleźć dział (lub wszystkie działy) zlokalizowane w podanej miejscowości; należy zwrócić `id_dzial` (lub listę),
- w tablicy `tpra` wyszukać wszystkich pracowników pracujących w podanym dziale (podanych działach).

Zazwyczaj zapytanie podrzędne można wykonać niezależnie, jeden raz dla zadanych parametrów.

Wyniki zapytania wewnętrznego są przekazywane do zapytania nadrzędnego.

Konstrukcja zapytania z podzapytaniem pozwala na sprawne jego wykonanie w przypadku zmiany danych w tablicach.

Zapytanie z podzapytaniem realizowane jest *jako jedna całość*.

System zarządzania bazą danych analizuje i optymalizuje wykonanie zapytania z podzapytaniem.

Przykład zapytania z podzapytaniem – realizacja

Podaj listę pracowników zatrudnionych w lokalizacji ’Mysieoczko’.

```
pracownicy=> SELECT * FROM tpra;
  id   |  nazwisko  |   imie   | dzial | stanowisko | pobory
-------+------------+----------+-------+------------+---------
 110   | Kowalik    | Artur    | PD303 | kierownik  | 1500.00
 101   | Kowalczyk  | Amadeusz | PK101 | kierownik  | 1000.00
 1101  | Kowalski   | Antoni   | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy   | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy   | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy    | PK101 | analityk   | 6000.00
 1100  | Kowal      |          | PD303 | robotnik   | 1500.00
 1110  | Kowalewski |          | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |          | PR202 | kierownik  | 2500.00
(9 rows)


pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)
```

Realizacja dwustopniowa oraz w postaci zapytania z podzapytaniem

```
pracownicy=> SELECT id_dzial FROM dzial
                   WHERE lokalizacja = 'Mysieoczko';
 id_dzial
----------
 PR202
(1 row)
```

Przekazujemy `PR202` do następnego zapytania:

```
pracownicy=> SELECT * FROM tpra WHERE dzial = 'PR202';
  id   |  nazwisko  | imie | dzial | stanowisko | pobory
-------+------------+------+-------+------------+---------
 1110  | Kowalewski |      | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |      | PR202 | kierownik  | 2500.00
(2 rows)
```

Zapytanie z podzapytaniem tworzą jedną całość:

```
pracownicy=> SELECT * FROM tpra
                    WHERE dzial =
                    (SELECT id_dzial
                         FROM dzial
                         WHERE lokalizacja = 'Mysieoczko'
                    );

  id   |  nazwisko  | imie | dzial | stanowisko | pobory
-------+------------+------+-------+------------+---------
 1110  | Kowalewski |      | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |      | PR202 | kierownik  | 2500.00
(2 rows)
```

Przykłady zapytań z podzapytaniem

Znajdź pracowników zatrudnionych w dziale produkcyjnym:

```
pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)

pracownicy=> \i podzapytanie1t.sql
  id   | nazwisko |  imie  | dzial | stanowisko | pobory
-------+----------+--------+-------+------------+---------
 110   | Kowalik  | Artur  | PD303 | kierownik  | 1500.00
 1101  | Kowalski | Antoni | PD303 | robotnik   | 4500.00
 1100  | Kowal    |        | PD303 | robotnik   | 1500.00
(3 rows)

podzapytanie1t.sql:
SELECT *
    FROM tpra
    WHERE dzial = (SELECT id_dzial
       FROM dzial
       WHERE nazwa = 'Produkcyjny'

   );
```

Przykłady zapytań z podzapytaniem

Znajdź pracowników zatrudnionych w dziale zlokalizowanym w miejscowości z literą ’c’ w nazwie:

```
pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)

pracownicy=> \i podzapytanie2t.sql
  id   |  nazwisko  |  imie  | dzial | stanowisko | pobory
-------+------------+--------+-------+------------+---------
 110   | Kowalik    | Artur  | PD303 | kierownik  | 1500.00
 1101  | Kowalski   | Antoni | PD303 | robotnik   | 4500.00
 1100  | Kowal      |        | PD303 | robotnik   | 1500.00
 1110  | Kowalewski |        | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |        | PR202 | kierownik  | 2500.00
(5 rows)

podzapytanie2t.sql:
SELECT *
     FROM tpra
     WHERE dzial IN (SELECT id_dzial
        FROM dzial
        WHERE lokalizacja LIKE ’%c%’
        );
```

Alternatywne lokalizacje

Podzapytanie nie musi zwracać pojedynczej wartości!!!

```
pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)
pracownicy=> \i podzapytanie3t.sql
  id   |  nazwisko  |   imie   | dzial | stanowisko | pobory
-------+------------+----------+-------+------------+---------
 110   | Kowalik    | Artur    | PD303 | kierownik  | 1500.00
 101   | Kowalczyk  | Amadeusz | PK101 | kierownik  | 1000.00
 1101  | Kowalski   | Antoni   | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy   | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy   | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy    | PK101 | analityk   | 6000.00
 1100  | Kowal      |          | PD303 | robotnik   | 1500.00
(7 rows)

podzapytanie3t.sql:
SELECT *
      FROM tpra
      WHERE dzial IN (SELECT id_dzial
        FROM dzial
        WHERE lokalizacja IN ('Mysiecko','Mysieko')
     );
```

Podzapytania: zagnieżdżanie wgłąb

W zapytaniu może wystąpić podzapytanie, w którym z kolei zagnieżdżone jest następne podzapytanie.

Taka struktura to zagnieżdżanie wielostopniowe (szeregowe).

Przykład:
Znajdź dane kierownika pewnego pracownika o `id = '1011'`.

```
pracownicy=> SELECT * FROM tpra;
  id   |  nazwisko  |   imie   | dzial | stanowisko | pobory
-------+------------+----------+-------+------------+---------
 110   | Kowalik    | Artur    | PD303 | kierownik  | 1500.00
 101   | Kowalczyk  | Amadeusz | PK101 | kierownik  | 1000.00
 1101  | Kowalski   | Antoni   | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy   | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy   | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy    | PK101 | analityk   | 6000.00
 1100  | Kowal      |          | PD303 | robotnik   | 1500.00
 1110  | Kowalewski |          | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |          | PR202 | kierownik  | 2500.00
(9 rows)

pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)
```

Rozwiązanie

```
pracownicy=> \i podzapytanie4t.sql
  id   | nazwisko  |   imie   | dzial | stanowisko | pobory
-------+-----------+----------+-------+------------+---------
 101   | Kowalczyk | Amadeusz | PK101 | kierownik  | 1000.00
(1 row)


podzapytanie4t.sql:

SELECT * FROM  tpra
      WHERE id= (SELECT kierownik
                    FROM dzial
                    WHERE id_dzial = (SELECT dzial
                                        FROM tpra
                                        WHERE id='1011'
                                      )
                );
```

Zagnieżdżanie równoległe

W zapytaniu mogą wystąpić podzapytania, które są niezależne od siebie – żadne z nich nie jest zagnieżdżone w drugim.

Taka struktura to zagnieżdżanie równoległe.

Przykład:
Znajdź dane pracowników będących kierownikami działów z lokalizacji `Mysieoczko` *lub (oraz) zarabiających więcej niż średnia.*

```
pracownicy=> SELECT * FROM tpra;
  id   |  nazwisko  |   imie   | dzial | stanowisko | pobory
-------+------------+----------+-------+------------+---------
 110   | Kowalik    | Artur    | PD303 | kierownik  | 1500.00
 101   | Kowalczyk  | Amadeusz | PK101 | kierownik  | 1000.00
 1101  | Kowalski   | Antoni   | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy   | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy   | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy    | PK101 | analityk   | 6000.00
 1100  | Kowal      |          | PD303 | robotnik   | 1500.00
 1110  | Kowalewski |          | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |          | PR202 | kierownik  | 2500.00
(9 rows)
pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)
```

Rozwiązanie

```
pracownicy=> \i podzapytanie5t.sql
  id   |  nazwisko  |  imie  | dzial | stanowisko | pobory
-------+------------+--------+-------+------------+---------
 1101  | Kowalski   | Antoni | PD303 | robotnik   | 4500.00
 102   | Janiak     | Jerzy  | PK101 | analityk   | 6000.00
 1110  | Kowalewski |        | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |        | PR202 | kierownik  | 2500.00
(4 rows)
```

podzapytanie5t.sql:

```
SELECT * FROM tpra
         WHERE id=      (SELECT kierownik
                            FROM dzial
                            WHERE lokalizacja=’Mysieoczko’
                        )
               OR
               pobory > (SELECT AVG(pobory)
                            FROM tpra
                        );
```

Warianty działania podzapytania

Podzapytanie może zwracać:

- zawsze pojedynczą wartość (możliwe użycie np. =),(**kiedy jest pewność zwrotu pojedynczej wartości?**)
- zero, jedną lub listę wartości (IN, ANY, ALL),
- być zastosowane jako test istnienia rozwiązania (EXISTS).

Możliwe operatory:

- `<exp> IN (<list>)` — wyrażenie `<exp>` należy do listy zwracanej podzapytaniem,
- `<exp> NOT IN (<list>)` — wyrażenie `<exp>` nie należy do listy zwracanej podzapytaniem,
- `<exp> <rel> ANY (<list>)` — wyrażenie `<exp>` spełnia relację `<rel>` w odniesieniu do pewnych (przynajmniej jednego) elementów listy zwracanej podzapytaniem,
- `<exp> <rel> ALL (<list>)` — wyrażenie `<exp>` spełnia relację `<rel>` w odniesieniu do wszystkich elementów listy zwracanej podzapytaniem,
- `EXISTS` — podzapytanie zwraca jakąś wartość,
- `NOT EXISTS` — podzapytanie nie zwraca żadnej wartości.

`<rel>` — to zwykle operator porównania: =, >, <. >=, <=, <>.

Examples from PostgreSQL docs

```
SELECT ... FROM fdt WHERE c1 > 5;


SELECT ... FROM fdt WHERE c1 IN (1, 2, 3);


SELECT ... FROM fdt WHERE c1 IN
                     (SELECT c1 FROM t2;)


SELECT ... FROM fdt WHERE c1 IN
                     (SELECT c3
                          FROM t2
                          WHERE c2 = fdt.c1 + 10);


SELECT ... FROM fdt WHERE c1 BETWEEN
                     (SELECT c3
                          FROM t2
                          WHERE c2 = fdt.c1 + 10
                     ) AND 100;


SELECT ... FROM fdt WHERE EXISTS
                     (SELECT c1
                          FROM t2
                          WHERE c2 > fdt.c1);
```

Przykładowe zamienniki

- `ANY` — `SOME`,
- `IN` — `=ANY`,
- `>= ALL` — `>= SELECT MAX`,
- `<= ALL` — `<= SELECT MIN`,
- `>= ANY` — `>= SELECT MIN`,
- `<= ANY` — `<= SELECT MAX`,

Uwaga:

`<exp> <> ANY (<list>)` to nie to samo co `<exp> NOT IN (<list>)`.

Pierwsze tłumaczy się jako dysjunkcja:

$$< exp >\neq l_1 \vee < exp >\neq l_2 \vee \ldots \vee < exp >\neq l_k,$$

a drugie jako koniunkcję:

$$< exp >\neq l_1 \wedge < exp >\neq l_2 \wedge \ldots \wedge < exp >\neq l_k,$$

Przykład z IN

Znajdź pracowników zatrudnionych w działach (lista) spełniających określone warunki:

```
pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)

pracownicy=> SELECT * FROM tpra;
  id   |  nazwisko  |   imie   | dzial | stanowisko | pobory
-------+------------+----------+-------+------------+---------
 110   | Kowalik    | Artur    | PD303 | kierownik  | 1500.00
 101   | Kowalczyk  | Amadeusz | PK101 | kierownik  | 1000.00
 1101  | Kowalski   | Antoni   | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy   | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy   | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy    | PK101 | analityk   | 6000.00
 1100  | Kowal      |          | PD303 | robotnik   | 1500.00
 1110  | Kowalewski |          | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |          | PR202 | kierownik  | 2500.00
(9 rows)
```

Rozwiązanie z IN

```
pracownicy=> \i podzapytanie3.sql
  id   |  nazwisko  |   imie   | dzial | stanowisko | pobory
-------+------------+----------+-------+------------+---------
 110   | Kowalik    | Artur    | PD303 | kierownik  | 1500.00
 101   | Kowalczyk  | Amadeusz | PK101 | kierownik  | 1000.00
 1101  | Kowalski   | Antoni   | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy   | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy   | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy    | PK101 | analityk   | 6000.00
 1100  | Kowal      |          | PD303 | robotnik   | 1500.00
 1110  | Kowalewski |          | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |          | PR202 | kierownik  | 2500.00
(9 rows)

podzapytanie3.sql:
SELECT *
     FROM tpra
     WHERE dzial IN
               (SELECT id_dzial
          FROM dzial
  WHERE lokalizacja IN (’Mysiecko’,’Mysieko’)
    OR  nazwa IN (’Promocji’,’Zbytu’)
      );
```

Podzapytania skorelowane – średnia

Ile wynosi średnia płaca:

```
pracownicy=> SELECT * FROM tpra;
  id   |  nazwisko  |   imie   | dzial | stanowisko | pobory
-------+------------+----------+-------+------------+---------
 110   | Kowalik    | Artur    | PD303 | kierownik  | 1500.00
 101   | Kowalczyk  | Amadeusz | PK101 | kierownik  | 1000.00
 1101  | Kowalski   | Antoni   | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy   | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy   | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy    | PK101 | analityk   | 6000.00
 1100  | Kowal      |          | PD303 | robotnik   | 1500.00
 1110  | Kowalewski |          | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |          | PR202 | kierownik  | 2500.00
(9 rows)
pracownicy=> SELECT round(AVG(pobory),2) FROM tpra;
  round
---------
 2833.33
(1 rows)
```

Ile wynosi średnia w działach:

```
pracownicy=> SELECT round(AVG(pobory),2)
                FROM tpra GROUP BY dzial;
  round
---------
 3000.00
 3000.00
 2500.00
(3 rows)
```

Podzapytania skorelowane – średnia

Kto zarabia więcej niż średnia:

```
pracownicy=> SELECT * FROM tpra
                 WHERE pobory >
                 (SELECT round(AVG(pobory),2) FROM tpra);

  id   |  nazwisko  |  imie  | dzial | stanowisko | pobory
-------+------------+--------+-------+------------+---------
 1101  | Kowalski   | Antoni | PD303 | robotnik   | 4500.00
 102   | Janiak     | Jerzy  | PK101 | analityk   | 6000.00
 1110  | Kowalewski |        | PR202 | robotnik   | 3500.00
(3 rows)
```

Kto zarabia więcej niż średnia w **jego** *dziale:*

```
pracownicy=> \i wiecej-niz-srednia-w-jego-dziale.sql
  id   |  nazwisko  |  imie  | dzial | stanowisko | pobory
-------+------------+--------+-------+------------+---------
 1101  | Kowalski   | Antoni | PD303 | robotnik   | 4500.00
 102   | Janiak     | Jerzy  | PK101 | analityk   | 6000.00
 1110  | Kowalewski |        | PR202 | robotnik   | 3500.00
(3 rows)

 wiecej-niz-srednia-w-jego-dziale.sql:
SELECT *
     FROM tpra z
     WHERE z.pobory >
        (SELECT round(AVG(pobory),2)
            FROM tpra w
            WHERE z.dzial = w.dzial);
```

Uwaga: konieczne użycie aliasów!

Przykład - podzapytanie skorelowane

Kto zarabia więcej niż jego *kierownik:*

```
pracownicy=> SELECT * FROM tpra;
  id   |  nazwisko  |   imie   | dzial | stanowisko | pobory
-------+------------+----------+-------+------------+---------
 110   | Kowalik    | Artur    | PD303 | kierownik  | 1500.00
 101   | Kowalczyk  | Amadeusz | PK101 | kierownik  | 1000.00
 1101  | Kowalski   | Antoni   | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy   | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy   | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy    | PK101 | analityk   | 6000.00
 1100  | Kowal      |          | PD303 | robotnik   | 1500.00
 1110  | Kowalewski |          | PR202 | robotnik   | 3500.00
 111   | Kowalczuk  |          | PR202 | kierownik  | 2500.00
(9 rows)
pracownicy=> \i wiecej-niz-jego-kierownik.sql
  id   |  nazwisko  |  imie  | dzial | stanowisko | pobory
-------+------------+--------+-------+------------+---------
 1101  | Kowalski   | Antoni | PD303 | robotnik   | 4500.00
 1011  | Kowalowski | Alojzy | PK101 | robotnik   | 2500.00
 1010  | Kawula     | Alojzy | PK101 | robotnik   | 2500.00
 102   | Janiak     | Jerzy  | PK101 | analityk   | 6000.00
 1110  | Kowalewski |        | PR202 | robotnik   | 3500.00
(5 rows)
wiecej-niz-jego-kierownik.sql:
SELECT *
   FROM tpra z WHERE z.pobory >
                (SELECT  pobory FROM  tpra w
                  WHERE w.stanowisko='kierownik'
                  AND w.dzial=z.dzial);
```

Przykład – podzapytania a złączenia

wiecej-niz-jego-kierownik-2.sql:

```
SELECT * FROM tpra z
   WHERE z.pobory >
             (SELECT pobory
                 FROM tpra w
                 WHERE w.id =
                       (SELECT kierownik
                           FROM dzial d
                           WHERE z.dzial=d.id_dzial)
                    );
```

wiecej-niz-jego-kierownik-z.sql:

```
SELECT p.id, p.nazwisko, p.imie,
       p.dzial, p.stanowisko, p.pobory
   FROM tpra p, dzial d, tpra k
   WHERE p.dzial=d.id_dzial AND
         d.kierownik=k.id
         AND
         p.pobory > k.pobory;
```

Przykład – EXISTS

Podaj działy które zatrudniają jakichś pracowników:

```
pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)

pracownicy=> \i exists.sql
 id_dzial |    nazwa    | lokalizacja | kierownik
----------+-------------+-------------+-----------
 PD303    | Produkcyjny | Mysiecko    | 110
 PK101    | Projektowy  | Mysieko     | 101
 PR202    | Promocji    | Mysieoczko  | 111
(3 rows)

exists.sql:
SELECT *
    FROM dzial d
    WHERE EXISTS
          (SELECT id
              FROM tpra p
              WHERE p.dzial = d.id_dzial);
```

Przykład – NOT EXISTS

Podaj działy które nie zatrudniają żadnych pracowników:

```
pracownicy=> SELECT * FROM dzial;
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PD303    | Produkcyjny  | Mysiecko    | 110
 PK101    | Projektowy   | Mysieko     | 101
 PR202    | Promocji     | Mysieoczko  | 111
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(5 rows)

pracownicy=> \i not-exists.sql
 id_dzial |    nazwa     | lokalizacja | kierownik
----------+--------------+-------------+-----------
 PZ404    | Zaopatrzenia | Myszków     |
 PZ505    | Zbytu        | Maszków     |
(2 rows)
```

not-exists.sql:

```
SELECT *
    FROM dzial d
    WHERE NOT EXISTS
          (SELECT id
              FROM tpra p
              WHERE p.dzial = d.id_dzial);
```