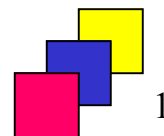


Język SQL. Rozdział 6.

Podzapytania

**Podzapytania proste i skorelowane,
podzapytania w klauzuli SELECT i FROM,
operatory ANY, ALL i EXISTS.**



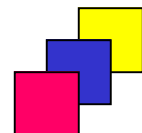
Podzapytania

- **Podzapytanie** jest poleceniem **SELECT** zagnieżdżonym w innym poleceniu **SELECT**. Podzapytanie może wystąpić wszędzie tam, gdzie system spodziewa się zbioru wartości, czyli w klauzulach **SELECT**, **FROM**, **WHERE**, **HAVING**.
- Ogólny format zagnieżdżania zapytań:
- Operatorem może być:
 - = <> < > <= >=
 - IN
 - ANY, ALL

```
SELECT atrybut1, atrybut2, ...  
FROM relacja  
WHERE atrybutn operator  
      (SELECT atrybuti, atrybutj  
       FROM relacja  
       WHERE warunek);
```

UWAGA!

W podzapytaniu nie może wystąpić klauzula **ORDER BY**
(wyjątek – podzapytanie w klauzuli **FROM**)



Podzapytania wyznaczające jedną krotkę

- Wyznacz pracownika zarabiającego najmniej w instytucie

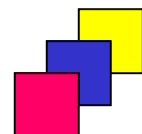
```
SELECT nazwisko, etat, placa_pod  
FROM pracownicy  
WHERE placa_pod =  
    (SELECT MIN(placa_pod)  
     FROM pracownicy);
```

208

- Wyznacz najgorzej zarabiającego asystenta

```
SELECT nazwisko, etat, placa_pod  
FROM pracownicy  
WHERE etat = 'ASYSTENT'  
AND placa_pod =  
    (SELECT MIN(placa_pod)  
     FROM pracownicy  
     WHERE etat='ASYSTENT');
```


371



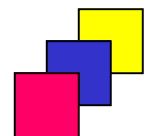
Podzapytania wyznaczające wiele krotek (1)

- Wyświetl nazwiska i płace pracowników, zatrudnionych w zespołach o nazwach "ADMINISTRACJA" lub "ALGORYTMY".

```
SELECT nazwisko, placa_pod, placa_dod
FROM pracownicy
WHERE id_zesp IN
      (SELECT id_zesp
       FROM zespoly
       WHERE nazwa in ('ALGORYTMY','ADMINISTRACJA'));
```



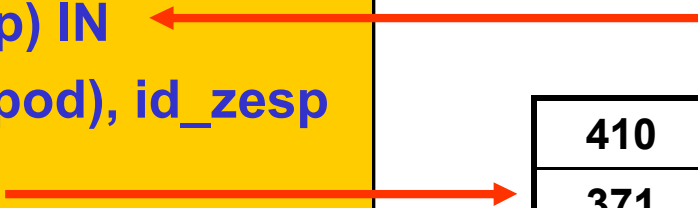
10
40



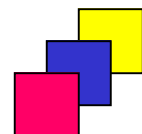
Podzapytania wyznaczające wiele krotek (2)

- Wyświetl nazwiska najgorzej zarabiających pracowników w każdym zespole

```
SELECT nazwisko, etat, placa_pod  
FROM pracownicy  
WHERE (placa_pod, id_zesp) IN  
      (SELECT MIN(placa_pod), id_zesp  
       FROM pracownicy  
       GROUP BY id_zesp);
```



410	10
371	20
208	30
1350	40



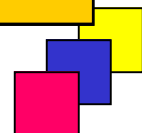
Podzapytania wyznaczające wiele krotek (3)

- Operator ANY
 - stosowany z operatorami logicznymi, warunek jest prawdziwy jeśli jest spełniony dla jakiegokolwiek wartości zwróconej przez podzapytanie.

```
SELECT nazwisko, placa_pod, etat, id_zesp FROM pracownicy  
WHERE placa_pod > ANY (SELECT DISTINCT placa_pod  
FROM pracownicy WHERE id_zesp = 30);
```

- Operator ALL
 - stosowany z operatorami logicznymi, warunek jest prawdziwy jeśli jest spełniony dla wszystkich wartości zwróconych przez podzapytanie.

```
SELECT nazwisko, placa_pod, etat, id_zesp FROM pracownicy  
WHERE placa_pod > ALL (SELECT DISTINCT placa_pod  
FROM pracownicy WHERE id_zesp = 30);
```



Najczęściej spotykane błędy

- Lista atrybutów w klauzuli SELECT podzapytania jest niezgodna z listą atrybutów w warunku:

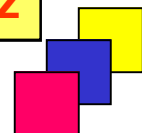
```
SELECT nazwisko, etat, placa_pod FROM pracownicy  
WHERE id_zesp =  
      (SELECT nazwisko, id_zesp FROM pracownicy  
       WHERE nazwisko='SLOWINSKI');
```

ORA-00913: za duża liczba wartości

- Podzapytanie zwraca więcej niż jeden wiersz a w warunku użyto operatora przewidzianego do porównywania wartości skalarnych:

```
SELECT nazwisko, etat, placa_pod FROM pracownicy  
WHERE placa_pod =  
      (SELECT MAX(placa_pod) FROM pracownicy  
       GROUP BY id_zesp );
```

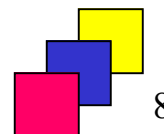
ORA-01427: jednowierszowe podzapytanie zwraca więcej niż jeden wiersz



Podzapytania w klauzuli HAVING

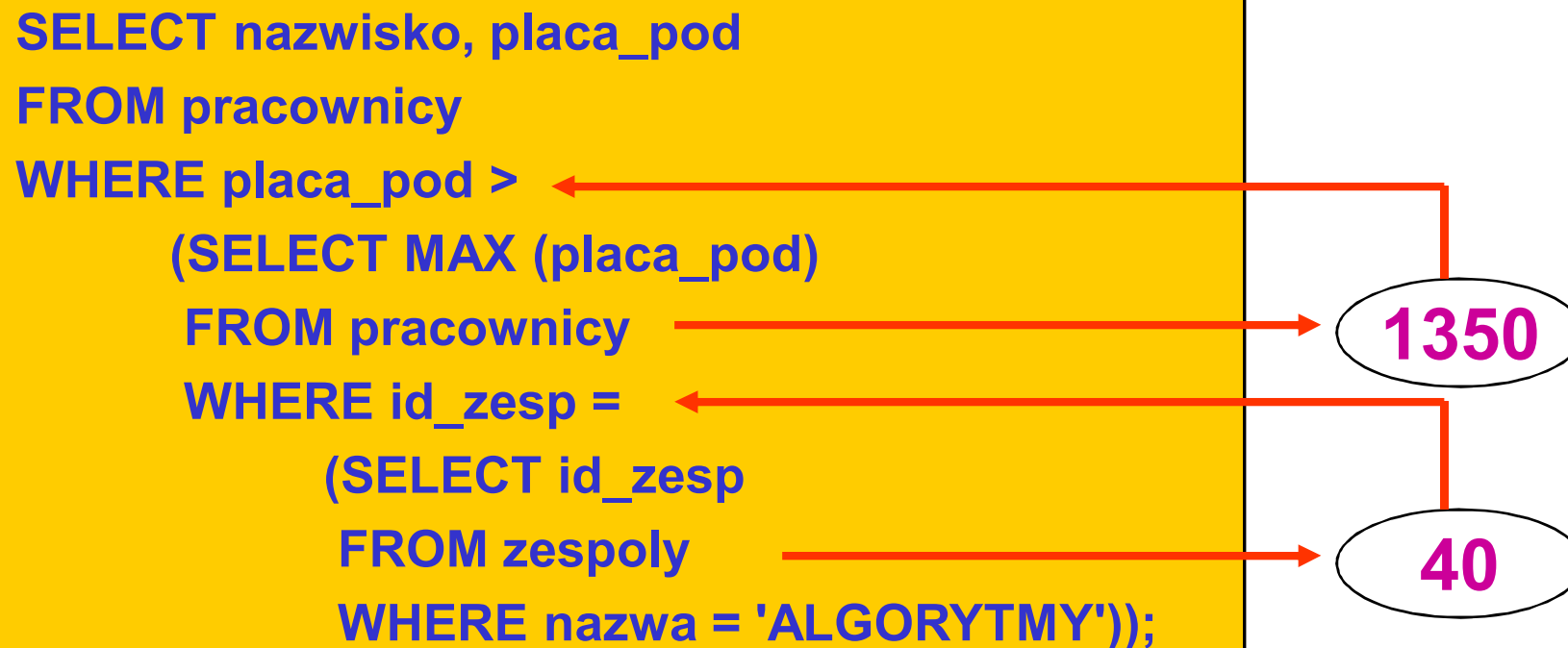
- Wyświetl te zespoły, w których średnia płaca podstawowa jest większa niż średnia płaca w całym instytucie.

```
SELECT z.nazwa, AVG(p.placa_pod) AS srednia
FROM pracownicy p, zespoły z
WHERE p.id_zesp = z.id_zesp
GROUP BY z.nazwa
HAVING AVG(p.placa_pod) >
      (SELECT AVG(placa_pod)
       FROM pracownicy);
```



Wielopoziomowe zagnieżdżanie zapytań

- Wyświetlić nazwiska i płace pracowników, zarabiających więcej niż wynosi maksymalna płaca w zespole o nazwie ALGORYTMY



Reguły zagnieżdżania podzapytań

- W podzapytaniu nie używamy klauzuli ORDER BY, klauzula ORDER BY może wystąpić wyłącznie jako ostatnia klauzula najbardziej zewnętrznego zapytania.
- Liczba oraz typy atrybutów występujących w klauzuli SELECT podzapytania musi być zgodna z liczbą i typem atrybutów użytych w warunku zapytania zewnętrznego.
- Podzapytania są zawsze wykonywane w kolejności od najgłębiej zagnieżdżonego do najbardziej zewnętrznego.
- Podzapytania mogą się znaleźć w dowolnym miejscu w klauzuli WHERE.

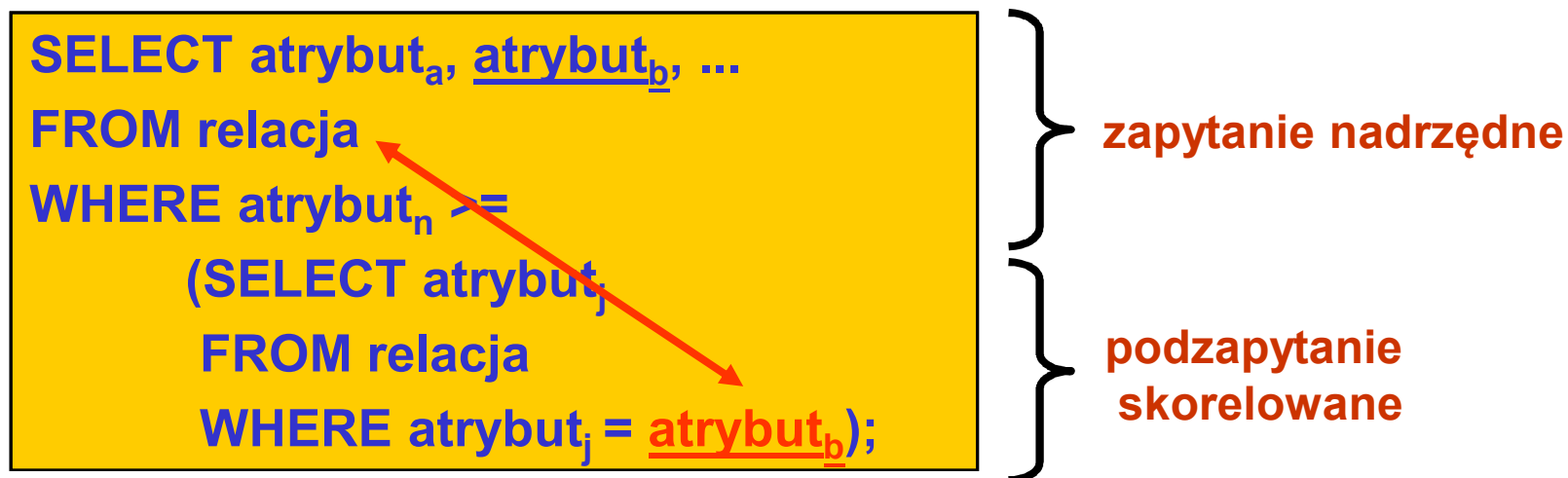
```
SELECT * FROM pracownicy  
WHERE (SELECT MIN(placa_pod) FROM pracownicy) = placa_pod;
```

```
SELECT * FROM pracownicy  
WHERE (SELECT MAX(placa_pod) FROM pracownicy  
        WHERE etat = 'PROFESOR' ) * 0.5 <= placa_pod;
```

Podzapytanie skorelowane (1)

Cechy

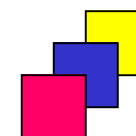
- Podzapytanie skorelowane jest wykonywane dla każdej krotki przeglądanej przez zapytanie nadrzędne
- Podzapytanie skorelowane operuje na wartościach atrybutów przekazanych przez zapytanie nadrzędne
- Podzapytanie skorelowane zawsze posiada odwołanie do atrybutu zapytania nadrzędnego



Podzapytanie skorelowane (2)

- Polecenie **SELECT** z podzapytaniem skorelowanym wykonywane jest następująco:
 1. pobranie krotki K_n przez zapytanie nadrzędne
 2. wykonanie podzapytania na podstawie wartości z krotki K_n
 3. zaakceptowanie bądź odrzucenie krotki K_n
 4. pobranie kolejnej krotki K_{n+1} przez zapytanie nadrzędne i powtórzenie kroków 2-4
- **Przykład:** Wyświetl nazwiska pracowników zarabiających powyżej średniej dla swojego zespołu.

```
SELECT nazwisko, placa_pod, id_zesp
FROM pracownicy p
WHERE placa_pod >
      (SELECT AVG(placa_pod) FROM pracownicy
       WHERE id_zesp = p.id_zesp);
```



Podzapytanie skorelowane (3)

P

NAZWISKO	PLACA_POD	ID_ZESP
LECH	300	1
CZECH	400	1
RUS	400	2
PIAST	500	2

Pracownicy

NAZWISKO	PLACA_POD	ID_ZESP
LECH	300	1
CZECH	400	1
RUS	400	2
PIAST	500	2

```
SELECT nazwisko, placa_pod, id_zesp
FROM pracownicy p
WHERE placa_pod >
  (SELECT AVG(placa_pod) FROM pracownicy
   WHERE id_zesp = p.id_zesp);
```

Operator EXISTS

- Operator zwraca wartość TRUE jeżeli podzapytanie zwraca jakąkolwiek wartość. Podzapytanie nie musi zwracać wartości z bazy danych, równie dobrze może zwracać dowolny literał.

```
SELECT id_prac, nazwisko, etat, id_zesp  
FROM pracownicy p  
WHERE EXISTS (SELECT id_prac FROM pracownicy  
              WHERE id_szefa = p.id_prac);
```

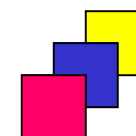
```
SELECT nazwisko, etat, id_zesp  
FROM pracownicy p  
WHERE NOT EXISTS (SELECT 1 FROM zespoly  
                 WHERE id_zesp = p.id_zesp);
```

Podzapytania w klauzuli SELECT

- Zapytanie, które zwraca dokładnie jedną wartość, jest poprawnym wyrażeniem i może być wykorzystane wszędzie tam, gdzie SQL oczekuje wyrażenia, np. w klauzuli SELECT

```
SELECT nazwa,  
       (SELECT MAX(placa_pod)  
        FROM pracownicy  
        WHERE id_zesp = z.id_zesp ) AS max_placa  
FROM zespoly z;
```

```
SELECT p.nazwisko,  
       (SELECT nazwisko  
        FROM pracownicy  
        WHERE id_prac = p.id_szefa ) AS szef  
FROM pracownicy p  
ORDER BY nazwisko;
```



Podzapytania w klauzuli FROM

- Wynik podzapytania może być wykorzystany jako wejściowy zbiór danych dla innego zapytania. Stąd w klauzuli FROM zamiast nazwy relacji można umieścić podzapytanie. W takim podzapytaniu dopuszczalne jest użycie klauzuli ORDER BY.
- Przykład: wyświetl nazwiska pracowników, którzy zarabiają więcej niż średnia płaca w zespole, w którym pracują

```
SELECT nazwisko, placa_pod, srednia_placa_w_zespole
FROM
    (SELECT id_zesp, AVG(placa_pod) as
        srednia_placa_w_zespole
    FROM pracownicy
    GROUP by id_zesp) z
JOIN pracownicy p ON z.id_zesp = p.id_zesp
WHERE placa_pod > srednia_placa_w_zespole;
```