

Podzapytania

Podzapytania

Podzapytania to zapytania w języku SQL umieszczane wewnątrz innego zapytania, tzw. zapytania zewnętrzne. Składają się one z typowej konstrukcji umieszczane są jednak w nawiasach i występują najczęściej po wyrażeniach WHERE lub HAVING.

Podzapytania wykonują się w pierwszej kolejności dostarczając niezbędnej informacji do przeprowadzenia selekcji zapytania zewnętrznego.

Wyróżnia się podzapytania wierszowe i tablicowe.

Podzapytania

Podzapytania wierszowe – stosuje się w ich przypadku wszystkie wyrażenia logiczne (porównujące), zapytania te zwracają jeden rekord posiadający ewentualnie kilka wartości.

Przykład: Znajdź nazwisko pracownika otrzymującego najniższą płacę podstawową:

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

Podzapytania

Zadanie: Podaj nazwisko PROFESORA otrzymującego najniższą płacę podstawową wśród profesorów.

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

W podzapytaniu do wartości wynikowej został dodany ciąg znaków „PROFESOR” co zostanie wykorzystane do porównania z zapytaniem zewnętrznym. Zamiast ciągu znaków „PROFESOR” **nie wolno** użyć nazwy atrybutu „etat” (nie jest to wyrażenie agregujące), gdyż wykorzystuje się tutaj klauzule WHERE

Podzapytania

Podzapytania tablicowe – jako wynik zwracają jedną lub wiele wartości, dlatego też można stosować operatory IN (zawieranie w zbiorze), ANY (warunek spełniony dla przynajmniej jednej wartości zwróconej przez zapytanie) lub ALL (warunek spełniony dla wszystkich wartości).

Przykład: Podaj nazwiska pracowników otrzymujących najwyższe płace podstawowe w swoich grupach etatowych:

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

Podzapytania

Zadanie: Podaj nazwiska pracowników, których płaca podstawowa jest większa od płacy podstawowej dowolnego pracownika zespołu 30.

```
SELECT nazwisko FROM pracownicy  
WHERE placa_pod ANY  
(SELECT placa_pod FROM pracownicy WHERE  
id_zesp=30);
```

Zadanie: Podaj nazwiska pracowników, których płaca podstawowa jest większa od płac podstawowych wszystkich pracowników zespołu 30.

```
SELECT nazwisko FROM pracownicy WHERE placa_pod ALL  
(SELECT placa_pod FROM pracownicy WHERE  
id_zesp=30);
```

Podzapytania

Podzapytania w klauzuli HAVING są konstruowane na podobnej zasadzie co podzapytania w klauzuli WHERE.

Przykład: Podaj nazwy i średnie płace podstawowe, w zespołach w których średnia płaca przekracza średnią płacę wśród wszystkich pracowników.

```
SELECT z.nazwa, AVG(p.placa_pod)  
FROM pracownicy p NATURAL JOIN zespoły z  
GROUP BY z.nazwa  
HAVING AVG(p.placa_pod) <  
        (SELECT AVG(placa_pod)  
         FROM pracownicy);
```

Połączenia relacji

Zadanie: Wyświetl nazwiska i etaty pracowników pracujących w tym samym zespole co pracownik o nazwisku Nowak (załóż, że w zbiorze pracowników istnieje tylko jeden Nowak).

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

Zadanie: Wyświetl dane zespołów, które nie zatrudniają żadnych pracowników.

```
SELECT * FROM zespoly WHERE id_zesp NOT IN  
(SELECT id_zesp FROM pracownicy WHERE id_zesp IS  
NOT NULL);
```


Połączenia relacji

Zadanie: Wyświetl nazwiska tych profesorów, którzy wśród swoich podwładnych nie mają żadnych stażystów.

```
SELECT nazwisko FROM pracownicy  
WHERE etat = 'PROFESOR' AND id_prac NOT IN  
(SELECT id_szefa FROM pracownicy WHERE etat =  
    'STAŻYSTA');
```

Zadanie: Wyświetl numer zespołu wypłacającego miesięcznie swoim pracownikom najwięcej pieniędzy

```
SELECT id_zesp, SUM(placa_pod) FROM pracownicy  
GROUP BY id_zesp HAVING SUM(placa_pod) =  
(SELECT MAX(SUM(placa_pod)) FROM pracownicy  
GROUP BY id_zesp);
```

Połączenia relacji

Zadanie: Podaj nazwę zespołu zatrudniającego najwięcej pracowników

```
SELECT z.nazwa, z.id_zesp, MAX(COUNT(p.id_prac))  
FROM zespoly z NATURAL JOIN pracownicy p  
GROUP BY z.id_zesp HAVING (COUNT(p.id_prac)) =  
(SELECT MAX(COUNT(*)) FROM pracownicy GROUP  
BY id_zesp);
```

Połączenia relacji

Specyficzną grupę zapytań stanowią zapytania skorelowane. Wykonują się one wielokrotnie to znaczy tyle razy ile rekordów znajduje się w relacji zapytania zewnętrznego. Zapytanie to pobiera pierwszy rekord z relacji i sprawdza dla niego warunek, jeśli nie jest spełniony odrzuca te wartości i przechodzi do kolejnego rekordu. Jeśli rekord spełnia warunki jest umieszczony w zbiorze wynikowym.

Podzapytanie wykonywane w warunku zapytania zewnętrznego jest ściśle związane z zawartością aktualnie pobranego rekordu przez zapytanie zewnętrzne. Wartość ta oczywiście może ulegać zmianie, przez co zmieniać się może warunek w podzapytaniu, dlatego też konieczne jest wielokrotne przechodzenie i wykonywanie zapytania w bazie danych.

Połączenia relacji

Przykład zapytania skorelowanego: Podaj nazwiska pracowników zarabiających więcej niż średnia płaca w ich grupie etatowej.

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

pracownicy p

Janicki	PROFESOR	2500
Nowicki	PROFESOR	3000
Opolski	ASYSTENT	1000
Kotarski	ASYSTENT	1500

pracownicy

Janicki	PROFESOR	2500
Nowicki	PROFESOR	3000
Opolski	ASYSTENT	1000
Kotarski	ASYSTENT	1500

Wartość p.etat to wartość atrybutu etat z tabeli pracownicy p, dla aktualnie pobranego rekordu.