Pobieranie danych

SELECT

(Podzapytania lub zapytania zagnieżdżone)

Podzapytania lub zapytania zagnieżdżone

to instrukcje SELECT umieszczone wewnątrz innych instrukcji SELECT.

Podzapytanie może być użyte w dowolnej klauzuli
— w klauzuli FROM będziemy je wykorzystywać jako źródła danych,

— w klauzuli SELECT jako odpowiedniki zmiennych lub funkcji,

— w klauzuli WHERE do wybierania danych itd.

Podobnie jak wywołania funkcji w innych funkcjach, zapytania możemy zagnieżdżać w innych zapytaniach.

Podzapytanie musi zostać zapisane w nawiasie.

KOLEJNOŚĆ WYKONYWANIA ZAPYTAŃ

Serwer baz danych wykonuje podzapytania, **zaczynając od najbardziej wewnętrznej instrukcji SELECT**, po to aby wynik tej instrukcji wykorzystać do wykonania zapytań zewnętrznych.

W zależności od typu zwracanych przez wewnętrzne zapytania wartości, podzapytania dzieli się na:

-podzapytania zwracające pojedynczą wartość skalarną, np. nazwisko sprzedawcy, który sprzedał najwięcej towarów;

-podzapytania zwracające listę wartości, np. identyfikatory sprzedanych w danym miesiącu towarów;

-podzapytania zwracające dane tabelaryczne, np. dane sprzedawców uzupełnione o liczbę i wartość zrealizowanych przez nich zamówień.

Niezależnie od typu zwracanych wartości, podzapytania mogą być powiązane lub niepowiązane:

1. W podzapytaniach niepowiązanych wewnętrzne zapytanie jest wykonywane tylko raz, a więc zwraca jeden wynik.

2. W podzapytaniach powiązanych wewnętrzne zapytanie jest wykonywane dla każdego wiersza zwróconego przez zewnętrzne zapytanie, a więc zwraca tyle wyników, ile wierszy liczy wynik zewnętrznego zapytania.

Podzapytania niepowiązane

Podzapytania niepowiązane wykonywane są następująco:

1. **Wykonana zostaje wewnętrzna instrukcja SELECT.**
2. **Jej wyniki są przekazywane do zapytania zewnętrznego.**
3. **Otrzymane dane pozwalają wykonać zapytanie zewnętrzne.**

Podzapytania jako źródła danych

Wynik podzapytania może być zbiorem danych źródłowych dla innego zapytania. W takim przypadku podzapytanie znajduje się w klauzuli FROM zapytania nadrzędnego.

Pozwala to:

1. Uprościć zapytanie i poprawić jego czytelność — ponieważ zapytanie wewnętrzne wykonywane jest jako pierwsze, zdefiniowane w nim aliasy kolumn mogą być używane w każdej klauzuli zapytania zewnętrznego;

2. Dynamicznie filtrować wiersze i wyliczać dane bazowe dla zapytań zewnętrznych.

PODZAPYTANIE w sekcji FROM

Podzapytanie jako źródło danych, czyli w roli dynamicznego widoku

```
SELECT * FROM (SELECT title, fname, lnami FROM customer) AS t
    WHERE title ='Mr';
```

title	fname	lname
Mr	Andrew	Stones
Mr	Adrian	Matthew
Mr	Simon	Cozens
Mr	Neil	Matthew
Mr	Richard	Stones
Mr	Mike	Howard
Mr	Dave	Jones
Mr	Richard	Neill
Mr	Bill	Neill
Mr	David	Hudson

MySQL, wykonując tę instrukcję, najpierw wykonał wewnętrzne zapytanie i **nazwał wynik t**, a następnie odczytał z tymczasowej tabeli **t** te wiersze, dla których spełniony był warunek **title ='Mr'**.

Uwaga: podzapytanie wykonuje się tylko raz!!!

Uwaga: Należy zwrócić uwagę na konieczność umieszczania Aliasu po zaptaniu uzytym w sekcji FROM!!!

PODZAPYTANIE w sekcji SELECT

Podzapytania jako zmienne

Podzapytania mogą być potraktowane jako zmienne.

Podzapytania zwracające pojedynczą wartość są odpowiednikami zmiennych typów prostych (przechowują jedną pojedynczą wartość);

Podzapytania zwracające listę wartości są odpowiednikiem zmiennych tabelarycznych.

Najwyraźniej widać to podobieństwo na przykładzie podzapytań niepowiązanych. Wynik podzapytania może zostać potraktowany jak zmienna z ustawioną w wyniku wykonania podzapytania wartością.

Również w tym przypadku **wewnętrzne zapytanie-wyrażenie jest wykonywane tylko raz podczas wykonywania całej instrukcji SELECT**.

Podzapytanie jako wyrażenie. W tym przypadku podzapytanie posłużyło nam do wyliczenia średniej ceny towarów

```
SELECT description, sell_price, (SELECT AVG(sell_price) FROM item)
    FROM item;
```

description	sell_price	(SELECT AVG(sell_price) FROM item)
Wood Puzzle	21.95	10.435455
Rubik Cube	11.49	10.435455
Linux CD	2.49	10.435455
Tissues	3.99	10.435455
Picture Frame	9.95	10.435455
Fan Small	15.75	10.435455
Fan Large	19.95	10.435455
Toothbrush	1.45	10.435455
Roman Coin	2.45	10.435455
Carrier Bag	0.00	10.435455
Speakers	25.32	10.435455
SQL Server 2005	NULL	10.435455

Zapytanie zwracające różnice pomiędzy ceną danego produktu a średnią ceną wszystkich towarów.

```
SELECT description, sell_price, sell_price – (SELECT AVG(sell_price) FROM item)
    FROM item;
```

description	sell_price	(SELECT AVG(sell_price)
Wood Puzzle	21.95	10.435455
Rubik Cube	11.49	10.435455
Linux CD	2.49	10.435455
Tissues	3.99	10.435455
Picture Frame	9.95	10.435455
Fan Small	15.75	10.435455
Fan Large	19.95	10.435455
Toothbrush	1.45	10.435455
Roman Coin	2.45	10.435455
Carrier Bag	0.00	10.435455
Speakers	25.32	10.435455
SQL Server 2005	NULL	10.435455

Jak widać pierwsze dwie kolumny były to dane uzyskane z zewnętrznego zapytania SELECT, natomiast ostatnia kolumna to wynik działania:

Cena towaru (z zewnętrznego zapytania)

- Średnia cena towaru (z zapytania zagnieżdżonego)

==========

PODZAPYTANIE w sekcji WHERE

Zapytanie zwracające nazwę najdroższego towaru.

```
SELECT description, sell_price FROM item
    WHERE sell_price = (SELECT MAX(sell_price) FROM item);
```

description	sell_price
Speakers	25.32

Przeanalizujmy wykonanie instrukcji przez MySQL:

1.Najpierw zostaje wykonana wewnętrzna instrukcja SELECT max(sell_price) FROM item. W jej wyniku otrzymujemy najwyższą cenę sprzedaży (25,32).

2.Następnie wykonywana jest instrukcja SELECT description, sell_price FROM item WHERE sell_price=25,32;, po której zrealizowaniu otrzymamy nazwę najdroższego towaru.

Ćwiczenie: Wypisać imie i wiek kotów starszych niż 5 lat – listę kotów należy pobrać sobie jako tymczasową tabelę w sekcji FROM (jako podzapytanie).

imie	wiek
fruzia	12
gotfryd	7
leopard	9
szemrak	12
macius	23
zdzisio	9
mruczus	6
bury	7
franka	7

SELECT * FROM (SELECT imie, wiek FROM koty) as K WHERE wiek > 5

imie	wiek
fruzia	12
gotfryd	7
leopard	9
szemrak	12
macius	23
zdzisio	9
mruczus	6
bury	7
franka	7

Ćwiczenie: Wypisac nazwe i wartosc zdobyczy, których wartosc jest z przedzialu (10,30) – liste zdobyczy pobierać jako wewnętrzne zapytanie SELECT w sekcji FROM.

nazwa	wartosc
wedzona makrela	12
medalik	22
ser	13
zapas mleka	23
obrazek	11

```
SELECT nazwa, wartosc FROM (SELECT * FROM zdobycze) as Z

    WHERE wartosc >10 and wartosc<30
```

nazwa	wartosc
wedzona makrela	12
medalik	22
ser	13
zapas mleka	23
obrazek	11

Ćwiczenie: Wyświetl nazwy, wartosci oraz różnicę pomiędzy wartością średnią a wartością zdobyczy (podzapytanie uzyte w sekcji SELECT).

nazwa	wartosc	ROZNICA
puszka szprotek	6	-26.5333
wedzona makrela	12	-20.5333
medalik	22	-10.5333
sztuczna mysz	34	1.4667
ser	13	-19.5333
bawik	34	1.4667
zapas mleka	23	-9.5333
paletko	54	21.4667
stare skarpety	1	-31.5333
pilka	65	32.4667
ryby	45	12.4667
kapsel	1	-31.5333
buda	78	45.4667
obrazek	11	-21.5333
szyneczka	89	56.4667

SELECT nazwa, wartosc, (wartosc - (SELECT AVG(wartosc) FROM zdobycze)) as ROZNICA FROM zdobycze

nazwa	wartosc	ROZNICA
puszka szprotek	6	-26.5333
wedzona makrela	12	-20.5333
medalik	22	-10.5333
sztuczna mysz	34	1.4667
ser	13	-19.5333
bawik	34	1.4667
zapas mleka	23	-9.5333
paletko	54	21.4667
stare skarpety	1	-31.5333
pilka	65	32.4667
ryby	45	12.4667
kapsel	1	-31.5333
buda	78	45.4667
obrazek	11	-21.5333
szyneczka	89	56.4667

Ćwiczenie: Wyświetl imie, wiek, oraz średni wiek kotów (podzapytanie w sekcji SELECT).

imie	wiek	SREDNI_WIEK
gapcio	3	6.1500
fruzia	12	6.1500
smykos	4	6.1500
lady gada	5	6.1500
myszka	2	6.1500
gotfryd	7	6.1500
marmelada	2	6.1500
leopard	9	6.1500
szemrak	12	6.1500
macius	23	6.1500
zdzisio	9	6.1500
mruczus	6	6.1500
czarnulka	4	6.1500
zbyszko	2	6.1500
bury	7	6.1500
zadziora	2	6.1500
franka	7	6.1500
malutka	2	6.1500
milka	3	6.1500
stasiu	2	6.1500

SELECT imie, wiek, (SELECT AVG(wiek) FROM koty) as SREDNI_WIEK FROM koty

imie	wiek	SREDNI_WIEK
gapcio	3	6.1500
fruzia	12	6.1500
smykos	4	6.1500
lady gada	5	6.1500
myszka	2	6.1500
gotfryd	7	6.1500
marmelada	2	6.1500
leopard	9	6.1500
szemrak	12	6.1500
macius	23	6.1500
zdzisio	9	6.1500
mruczus	6	6.1500
czarnulka	4	6.1500
zbyszko	2	6.1500
bury	7	6.1500
zadziora	2	6.1500
franka	7	6.1500
malutka	2	6.1500
milka	3	6.1500
stasiu	2	6.1500

Ćwiczenie: Wypisz imie i wiek kotów, których wiek jest wyższy niż średni wiek kotów (podzapytanie w sekcji WHERE).

imie	wiek
fruzia	12
gotfryd	7
leopard	9
szemrak	12
macius	23
zdzisio	9
bury	7
franka	7

SELECT imie, wiek FROM koty WHERE wiek > (SELECT AVG(wiek) FROM koty)

imie	wiek
fruzia	12
gotfryd	7
leopard	9
szemrak	12
macius	23
zdzisio	9
bury	7
franka	7

Ćwiczenie: Wypisz imie i wiek kotów, których wiek jest wyższy niż średni wiek kotów z bandy nr 2 (podzapytanie w sekcji WHERE).

imie	wiek
gapcio	3
fruzia	12
smykos	4
lady gada	5
gotfryd	7
leopard	9
szemrak	12
macius	23
zdzisio	9
mruczus	6
czarnulka	4
bury	7
franka	7
milka	3

```
SELECT imie, wiek FROM koty

    WHERE wiek > (SELECT AVG(wiek) FROM koty WHERE id_bandy=2)
```

imie	wiek
gapcio	3
fruzia	12
smykos	4
lady gada	5
gotfryd	7
leopard	9
szemrak	12
macius	23
zdzisio	9
mruczus	6
czarnulka	4
bury	7
franka	7
milka	3

Podzapytania powiązane (SKORELOWANE)

Podzapytania powiązane (skorelowane)

Wykonanie poprzednio opisanych podzapytań sprowadza się do wykonania wewnętrznej instrukcji SELECT i zwrócenia obliczonego wyniku do zapytania zewnętrznego.

Podzapytania drugiego typu, **podzapytania powiązane**, wykonywane są według innego schematu. W tym wypadku **podzapytanie wykonywane jest dla każdego wiersza wyniku zapytania zewnętrznego** i może być z nim porównywane. Podzapytanie powiązane jest przykładem dynamicznego złączenia wyniku zapytania z każdym kolejnym wierszem wyniku zapytania zewnętrznego.

Uwaga: Podzapytanie powiązane jako przykład dynamicznego złączenia można łatwo rozpoznać po tym, że kolumna (kolumny) wyniku podzapytania jest porównywana z kolumną (kolumnami) wyniku zapytania zewnętrznego. Niezależne wykonanie wewnętrznego zapytania jest w tym przypadku niemożliwe.

Przykład 1. Wyświetl pracowników, którzy zarabiają mniej, niż wynosi średnia płaca w ich działach.

Na początek lista wszystkich pracowników:

ID	IMIE	WIEK	email	zarobki	dzial
1	Jacek	43	*NULL*	2500	1
2	Marek	45	e1@op.pl	2100	2
3	Marta	34	*NULL*	3400	3
4	Jola	23	e2@op.pl	5500	2
5	Jasio	34	*NULL*	2300	0
6	Radek	54	e3@op.pl	1800	2
7	Marek	45	*NULL*	1500	1
8	Jan	33	e4@op.pl	2000	1
9	Olek	55	e5@op.pl	2500	2
10	Maria	22	*NULL*	4000	3
11	Franek	44	*NULL*	3000	0
12	Maja	44	*NULL*	2000	0

```
SELECT id, imie, zarobki, dzial
    FROM users as ZEW
        WHERE zarobki < (SELECT AVG(zarobki) FROM users as WEW
            WHERE WEW.dzial=ZEW.dzial);
```

id	imie	zarobki	dzial
2	Marek	2100	2
3	Marta	3400	3
5	Jasio	2300	0
6	Radek	1800	2
7	Marek	1500	1
9	Olek	2500	2
12	Maja	2000	0

To co widać to uzaleznienie wyniku wyszukiwania zapytania wewnętrznego od wyników zapytania zewnętrznego...

Analiza przykładu…

SELECT id, imie, zarobki, dzial FROM users as ZEW WHERE zarobki < (SELECT AVG(zarobki) FROM users as WEW WHERE WEW.dzial=ZEW.dzial);

Najpierw BD przeprowadzi selekcję, które dane z wiersza nas będą interesować:

SELECT id, imie, zarobki, dzial

We „FROM users as ZEW” wskazujemy, że te atrybuty weźmiemy z tabeli users.

Analiza przykładu…

SELECT id, imie, zarobki, dzial FROM users as ZEW WHERE zarobki < (SELECT AVG(zarobki) FROM users as WEW WHERE WEW.dzial=ZEW.dzial);

W podzapytaniu skorelowanym (wewnętrznym) musimy skorzystać z aliasu ZEW, czyli formy wskazania, że chodzi nam dokładnie o tabelę z pytania ZEWnętrznego.

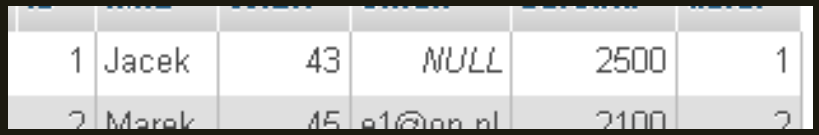
Alias WEW dla tabeli z podzapytania jest zbędny, ale znacząco poprawia czytelność (przejrzystość).

Analiza przykładu…

SELECT id, imie, zarobki, dzial FROM users as ZEW WHERE zarobki < (SELECT AVG(zarobki) FROM users as WEW WHERE WEW.dzial=ZEW.dzial);

Ten alias ZEW w zapytanie WEWnętrznym w klauzuli WHERE wskaże, ze wartość WEW.dzial podzapytanie ma sobie pobrać z wiersza, który akurat wzięło pod lupę zapytanie ZEWnętrzne.

Czyli bierzemy pierwszą osobę z listy



1	Jacek	43	*NULL*	2500	1
2	Marek	45	a1@pp.pl	2100	2

i sprawdzamy czy jego zarobki są mniejsze od ŚREDNICH ZAROBKÓW w jego dziale (dział nr 1). Zadaniem podzapytania będzie zatem wyliczyć średnią zarobków dla tego działu: SELECT AVG(zarobki) FROM users as WEW WHERE WEW.dzial=ZEW.dzial

zatem w miejsce ZEW.dzial podstawimy 1 i obliczymy wartość wyrażenia:

SELECT AVG(zarobki) FROM users as WEW WHERE WEW.dzial=1

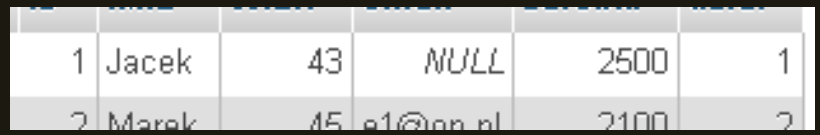
Analiza przykładu…

SELECT id, imie, zarobki, dzial FROM users as ZEW WHERE zarobki < (SELECT AVG(zarobki) FROM users as WEW WHERE WEW.dzial=ZEW.dzial);

Wynik podzapytania (SELECT AVG(zarobki) FROM users as WEW WHERE WEW.dzial=1) będzie podstawiony w miejsce oprównania w zapytanie ZEWNĘTRZNYM.

WHERE zarobki < (wynik podzapytania)

Na tej podstawie będzie podjęta decyzja czy JACEK ma zostać wyświetlony w wynikach wyszukiwania…



1	Jacek	43	*NULL*	2500	1
2	Marek	45	e1@op.pl	2100	2

Analiza przykładu…

Cała procedura będzie powtarzana dla każdej osoby z listy…

ID	IMIE	WIEK	email	zarobki	dzial
1	Jacek	43	*NULL*	2500	1
2	Marek	45	e1@op.pl	2100	2
3	Marta	34	*NULL*	3400	3
4	Jola	23	e2@op.pl	5500	2
5	Jasio	34	*NULL*	2300	0
6	Radek	54	e3@op.pl	1800	2
7	Marek	45	*NULL*	1500	1
8	Jan	33	e4@op.pl	2000	1
9	Olek	55	e5@op.pl	2500	2
10	Maria	22	*NULL*	4000	3
11	Franek	44	*NULL*	3000	0
12	Maja	44	*NULL*	2000	0

…a więc podzapytanie wyliczy średnią dla działu nr 2 (Marek), nr 3 (Marta), … itd.

WNIOSEK?

Podzapytania skorelowane, są bardzo użyteczne, dają ogromne możliwości, ale są dość obciążające dla systemu BD.

Przykład 2. Wyświetl informacje o 5 najlepiej zarabiających pracownikach.

```
SELECT * FROM users as ZEW
    WHERE 5 > (SELECT COUNT(*) FROM users as WEW
                WHERE WEW.zarobki>ZEW.zarobki)
```

ID	IMIE	WIEK	email	zarobki	dzial
1	Jacek	43	*NULL*	2500	1
3	Marta	34	*NULL*	3400	3
4	Jola	23	e2@op.pl	5500	2
9	Olek	55	e5@op.pl	2500	2
10	Maria	22	*NULL*	4000	3
11	Franek	44	*NULL*	3000	0

Analiza przykładu…

SELECT * FROM users as ZEW WHERE 5 > (SELECT COUNT(*) FROM users as WEW WHERE WEW.zarobki>ZEW.zarobki)

COUNT(*) w podzapytaniu WEWnętrznym zlicza, ile jest osób, które mają pensję większą od tej w danym wierszu pobranym z zapytania ZEWnętrznego.

Jeżeli zliczeń jest mniej niż 5, wyświetla wiersz pobrany w zapytaniu ZEWnętrznym.

…i znów kolejno dla każdej osoby z listy będzie trzeba dokonać zliczeń z zapytania WEWnętrznego…

Ćwiczenie: Wyświetl koty, których wiek jest niższy od średniego wieku kotów z ich bandy.

id_k	imie	wiek	id_bandy
1	gapcio	3	1
5	myszka	2	2
6	gotfryd	7	1
7	marmelada	2	3
12	mruczus	6	1
13	czarnulka	4	3
14	zbyszko	2	1
15	bury	7	1
16	zadziora	2	2
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

```
SELECT * FROM koty as ZEW
    WHERE ZEW.wiek < (SELECT AVG(WEW.wiek) FROM koty as WEW
                      WHERE WEW.id_bandy=ZEW.id_bandy)
```

id_k	imie	wiek	id_bandy
1	gapcio	3	1
5	myszka	2	2
6	gotfryd	7	1
7	marmelada	2	3
12	mruczus	6	1
13	czarnulka	4	3
14	zbyszko	2	1
15	bury	7	1
16	zadziora	2	2
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

Ćwiczenie:Wypisz imie, wiek oraz bandę do której należą koty będące szefami.

imie	wiek	BADNA_DO_KTOREJ_NALEZY
zbyszko	2	dachowce
smykos	4	rakiety
milka	3	szalone samice

Podpowiedź (przypomnienie): kot może być szefem bandy, do której nienależy, oraz kot może być szefem więcej niż 1 bandy.

```
SELECT ZEW.imie, ZEW.wiek, nazwa as BADNA_DO_KTOREJ_NALEZY
FROM koty as ZEW, bandy
WHERE ZEW.id_bandy = id_b and ((SELECT COUNT(*)
                                FROM bandy as WEW
                                WHERE WEW.id_szefa = ZEW.id_k) >=1)
```

imie	wiek	BADNA_DO_KTOREJ_NALEZY
zbyszko	2	dachowce
smykos	4	rakiety
milka	3	szalone samice

Więcej przykładów i informacji w fachowej literaturze oraz w sieci.