

Bazy danych
Laboratorium 4
Info

Złączenia

NR_AKT	NAZWISKO	STANOWISKO	KIEROWNIK	DATA_ZATR	DATA_ZWOL	PLACA	DOD_FUNKCYJNY	PROWIZJA	ID_DZIALU	ID_DZIALU_1	NAZWA	SIEDZIBA
1	8904 SKALSKI	GŁÓWNY INFORMATYK	8901	89/08/18	(null)	3500	2500	(null)	10	10	ZARZAD	CZESTOCHOWA
2	8913 KOWALSKA	CZŁONEK ZARZADU	8901	89/11/15	(null)	5000	2000	(null)	10	10	ZARZAD	CZESTOCHOWA
3	8901 KROL	PREZES	(null)	89/07/01	(null)	7000	4000	(null)	10	10	ZARZAD	CZESTOCHOWA
4	9501 OCIEPA	GŁÓWNY INFORMATYK	8901	99/08/01	06/05/31	3500	2500	(null)	10	10	ZARZAD	CZESTOCHOWA
5	9411 LISIECKI	LABORANT	8910	94/09/10	(null)	1300	(null)	(null)	20	20	PRODUKCJA	HONG KONG
6	9647 PIETURA	LABORANT	8910	93/03/15	94/12/01	1200	(null)	(null)	20	20	PRODUKCJA	HONG KONG

Dane z tabeli Pracownicy

Dane z tabeli Dzialy

Rys. 1 Wynik uzyskany dla:

```
select * from pracownicy p JOIN dzialy d ON (p.id_dzialu=d.id_dzialu);
```

ID_DZIALU	NR_AKT	NAZWISKO	STANOWISKO	KIEROWNIK	DATA_ZATR	DATA_ZWOL	PLACA	DOD_FUNKCYJNY	PROWIZJA	NAZWA	SIEDZIBA
40	70	9028 LESZCZYNSKI	SPRZEDAWCA	9235	90/08/10	(null)	1200	(null)	6000	SPRZEDAZ	CZESTOCHOWA
41	70	9102 KOWALCZYK	AKWIZYTOR	9235	91/01/10	(null)	1200	(null)	12000	SPRZEDAZ	CZESTOCHOWA
42	70	9902 PROTASIEWICZ	DYREKTOR	8901	04/01/01	10/03/31	3500	1500	(null)	SPRZEDAZ	CZESTOCHOWA
43	70	9235 FIKUS	DYREKTOR	8901	94/09/16	(null)	5000	1500	(null)	SPRZEDAZ	CZESTOCHOWA
44	70	9641 MALYSZ	MANAGER GRUPY	9235	96/12/05	(null)	3000	500	10000	SPRZEDAZ	CZESTOCHOWA

Rys. 2 Wynik uzyskany dla:

```
select * from pracownicy NATURAL JOIN dzialy;
select * from pracownicy JOIN dzialy USING (id_dzialu);
```

ID_DZIALU	NR_AKT	NAZWISKO	STANOWISKO	KIEROWNIK	DATA_ZATR	DATA_ZWOL	PLACA	DOD_FUNKCYJNY	PROWIZJA	NAZWA	SIEDZIBA
41	70	9641 MALYSZ	MANAGER GRUPY	9235	96/12/05	(null)	3000	500	10000	SPRZEDAZ	CZESTOCHOWA
42	70	9102 KOWALCZYK	AKWIZYTOR	9235	91/01/10	(null)	1200	(null)	12000	SPRZEDAZ	CZESTOCHOWA
43	70	9028 LESZCZYNSKI	SPRZEDAWCA	9235	90/08/10	(null)	1200	(null)	6000	SPRZEDAZ	CZESTOCHOWA
44	70	9235 FIKUS	DYREKTOR	8901	94/09/16	(null)	5000	1500	(null)	SPRZEDAZ	CZESTOCHOWA
45	(null)	9780 SAMOSINSKI	PRAKTYKANT	9332	06/04/01	(null)	1150	(null)	(null)	(null)	(null)

Rys. 3 Wynik uzyskany dla:

```
select * from pracownicy LEFT JOIN dzialy USING (id_dzialu);
```

1	2	3	4	5	6	
ID_RYBY	ID_RYBY	NAZWA	ID_RYBY	NAZWA	ID_RYBY	NAZWA
1	1	KARP	1	KARP	1	KARP
2	2	LIN	2	LIN	2	LIN
3	3	LESZCZ	3	LESZCZ	3	LESZCZ
4	4	AMUR	4	AMUR	4	AMUR
6	5	WEGORZ	6	BRZANA	5	WEGORZ
7	6	BRZANA	7	SWINKA	6	BRZANA
9	7	SWINKA	9	SZCZUPAK	7	SWINKA
10	8	JAZ	10	SANDACZ	8	JAZ
11	9	SZCZUPAK	11	SUM	9	SZCZUPAK
15	10	SANDACZ	15	OKON	10	SANDACZ
17	11	SUM	17	PLOC	11	SUM
20	12	KLEN	20	BOLEN	12	KLEN
(null)	13	PSTRAG POTOKOWY	(null)	(null)	13	PSTRAG POTOKOWY
	14	PSTRAG ZRODLANY			14	PSTRAG ZRODLANY
	15	OKON			15	OKON
	16	LIPIEN			16	LIPIEN
	17	PLOC			17	PLOC
	18	MIETUS			18	MIETUS
	19	INNE			19	INNE
	20	BOLEN			20	BOLEN
	21	KARAS			21	KARAS
	22	JELEC			22	JELEC
	23	SIEJA			23	SIEJA
	(null)	(null)			(null)	(null)

Rys. 4 Kolor żółty/niebieski – gatunki ryb, których odnotowano/nie odnotowano zdarzenia połowu

1. SELECT distinct(id_ryby) FROM rejestr order by 1;
2. SELECT id_ryby, nazwa FROM ryba;
3. SELECT distinct(id_ryby), nazwa FROM rejestr r JOIN ryba b USING (id_ryby) order by 1;
4. SELECT distinct(id_ryby), nazwa FROM rejestr r LEFT JOIN ryba b USING (id_ryby) order by 1;
5. SELECT distinct(id_ryby), nazwa FROM rejestr r RIGHT JOIN ryba b USING (id_ryby) order by 1;
6. SELECT distinct(id_ryby), nazwa FROM rejestr r FULL JOIN ryba b USING (id_ryby) order by 1;

Przykłady

1. Z tabeli rejestr wyświetl uporządkowaną informację dotyczącą statystyk poszczególnych łowisk w zakresie liczby odnotowanych na nich połowów wędkarskich (z podziałem na wszystkie/udane) oraz łącznej wagi złowionych na nich ryb.

ID_LOWISKA	Liczba polowow	Liczba udanych polowow	Laczna waga ryb
1 C11	18	11	15,75
2 1	6	5	15,4
3 C40	11	8	10,95

Rys. 5 Fragment wyniku:

```
SELECT id_lowiska, count(*) as "Liczba polowow" , count(ALL(id_ryby)) as "Liczba udanych polowow",
sum(NVL(waga,0)) as "Laczna waga ryb"
FROM rejestr GROUP BY id_lowiska ORDER BY 4 DESC;
```

2. Rozszerzenie zadania 1. Z tabeli rejestr wyświetl uporządkowaną informację dotyczącą statystyk połowu poszczególnych gatunków ryb na poszczególnych łowiskach w zakresie liczby oraz łącznej wagi złowionych ryb.

Wyświetl informacje o gatunkach, których łączna waga złowionych ryb w danym łowisku znajduje się w przedziale [2, 5] kg a liczba złowionych ryb (sztuk) jest większa niż 1.

ID_LOWISKA	ID_RYBY	Liczba złowionych ryb	Laczna waga ryb
1 C11	1	3	4,95
2 C11	10	2	4
3 C36	1	3	3,95
4 C02	4	2	3,5
5 029	9	2	3,25
6 C02	2	2	2

Rys. 6 Wynik:

```
SELECT id_lowiska, id_ryby, count(*) as "Liczba złowionych ryb", sum(NVL(waga,0)) as "Laczna waga ryb"
FROM rejestr WHERE id_ryby is not NULL
GROUP BY id_lowiska, id_ryby HAVING sum(NVL(waga,0)) between 2 and 5 and count(id_ryby)>1 ORDER BY 4
DESC; -- zastosowanie id_ryby is not NULL powoduje, że nie uwzględnia się połowów „bez ryby” więc można count(*) i bez NVL
```

3. Rozszerzenie zadania 3 o podanie nazw łowisk i gatunków ryb oraz dodatkowe statystyki (dołączenie tabel ryba, lowisko i wedkarz).

LOWISKO	GATUNEK	Liczba złowionych ryb	Laczna waga ryb	Najlżejsza ryba	Najcięższa ryba	Srednia dlugosc
1 Poraj	KARP	3	4,95	1,1	2,6	45,3
2 Poraj	SANDACZ	2	4	1,7	2,3	60,5
3 Ostrowy	KARP	3	3,95	1,15	1,55	44
4 Zielona	AMUR	2	3,5	1,2	2,3	46,5
5 Wisla	SZCZUPAK	2	3,25	1,1	2,15	56,5
6 Zielona	LIN	2	2	0,7	1,3	37,5

Rys. 7 Wynik:

```
SELECT l.nazwa Lowisko, b.nazwa Gatunek, count(id_ryby) as "Liczba złowionych ryb", sum(NVL(waga,0)) as
"Laczna waga ryb", min(waga) as "Najlżejsza ryba", max(waga) as "Najcięższa ryba", Round(avg(dlugosc),1) as "Srednia
dlugosc"
FROM rejestr r LEFT JOIN ryba b USING(id_ryby) JOIN lowisko l USING(id_lowiska)
WHERE id_ryby is not NULL GROUP BY l.nazwa, b.nazwa HAVING sum(NVL(waga,0)) between 2 and 5 and
count(id_ryby)>1 ORDER BY 4 DESC; -- zamiast LEFT JOIN sam JOIN i wówczas odrzucimy „połowcy bez ryby”
```

4. Kontynuacja zagadnienia z poprzednich zadań. Wyświetl gatunki ryb (bez informacji o łowiskach oraz dodatkowych statystyk zdefiniowanych w ramach zadania 3), których łączna waga złowionych ryb jest większa od średniej liczonej z sum wag ryb poszczególnych gatunków (np. waga wszystkich karpia + waga wszystkich leszczy + ... + waga wszystkich szczupaków podzielona przez liczbę gatunków).

GATUNEK	Liczba złowionych ryb	Laczna waga ryb
1 SZCZUPAK	13	25,05
2 KARP	12	23
3 SANDACZ	8	17,5
4 LESZCZ	10	14
5 SUM	1	9,15

Rys. 8 Wynik:

```
SELECT b.nazwa Gatunek, count(id_ryby) as "Liczba złowionych ryb", sum(waga) as "Laczna waga ryb"
FROM rejestr r LEFT JOIN ryba b USING(id_ryby)
WHERE id_ryby is not NULL
GROUP BY b.nazwa HAVING sum(waga) > (SELECT AVG(sum(waga)) FROM rejestr WHERE id_ryby is not NULL
GROUP BY id_ryby) ORDER BY 3 DESC;
```

5. Kolejny krok wtajemniczenia ;) Załóżmy, że chcemy wyświetlić informacje o połowach ryb na danym łowisku, których waga jest większa od średniej wagi ryb tego gatunku łowionych na określonym łowisku (np. jeżeli w łowisku Poraj złowiono 5 sztuk karpia o wagach kolejno 1, 2, 3, 4 i 5 kg to średnia waga wynosi 3 kg co oznacza, że zostaną wyświetlone informacje o połowach na łowisku Poraj karpia o masach 4 i 5 kg – ponieważ ich wagi są większe niż średnia).

Ponadto w zadaniu zakładamy, że w statystyce uwzględnione zostaną tylko pary gatunek-łowisko, które w tabeli rejestr występują przynajmniej 3 razy (w powyższym przykładzie na łowisku Poraj odnotowano połów 5 ryb z gatunku karp).

Proponowane korki postępowania

Krok 1. Zdefiniowanie zapytania, które zwróci średnie wagi łowionych ryb danego gatunku na danym łowisku pod warunkiem, że liczba złowionych ryb wynosiła co najmniej 3 sztuki (zwróć uwagę na nazwy kolumn).

ID_LOWISKA	ID_RYBY	SREDNIA	LICZBA_RYB
1 C40	9	1,91	4
2 C20	15	0,32	3
3 C11	1	1,65	3
4 13	10	2,53	3
5 C20	9	1,85	3
6 C11	3	1,1	5
7 C36	1	1,32	3

Rys. 9 Wynik: **SELECT id_lowiska, id_ryby, Round(avg(waga),2) Srednia, count(*) Liczba_ryb FROM rejestr WHERE id_ryby is not NULL GROUP BY id_lowiska, id_ryby HAVING count(*)>=3;**

Krok 2 Każda para łowisko-gatunek (spełniającej warunek w zakresie liczebności) ma swoją wartość średniej. W zadaniu musimy odszukać zdarzenia połowu, w których to waga ryby danego gatunku na danym łowisku była większa od średniej. W praktyce oznacza to, że nie uda nam się zastosować operatora IN (jak w zadaniu 6) ponieważ mamy sytuację, gdzie dwa elementy muszą być sobie równe (id_ryby, id_lowiska) ale waga musi być większa od średniej. W tej sytuacji możemy zastosować złączenie wyniku uzyskanego w kroku 1 z tabelą rejestr (wynik porównaj z wynikiem z Rys. 5).

ID_LOWISKA	ID_RYBY	SREDNIA	LICZBA_RYB	DATAZAS	ID_WEDKARZA	DLUGOSC	WAGA
1 13	10	2,53	3	15/07/21 22:35:00,000000000	10004	62	2,4
2 13	10	2,53	3	15/08/14 05:10:00,000000000	10010	59	1,8
3 13	10	2,53	3	15/07/21 23:55:00,000000000	10004	68	3,4
4 C11	1	1,65	3	15/04/24 15:10:00,000000000	10001	45	1,25
5 C11	1	1,65	3	16/06/29 18:40:00,000000000	10010	40	1,1
6 C11	1	1,65	3	15/06/14 05:10:00,000000000	10001	51	2,6
7 C11	3	1,1	5	15/07/24 17:35:00,000000000	10003	43	0,95
8 C11	3	1,1	5	14/07/14 08:55:00,000000000	10002	38	0,7
9 C11	3	1,1	5	15/04/24 17:05:00,000000000	10001	40	0,85
10 C11	3	1,1	5	15/04/24 15:45:00,000000000	10001	56	1,8
11 C11	3	1,1	5	16/06/29 12:10:00,000000000	10010	46	1,2
12 C20	9	1,85	3	15/05/01 15:10:00,000000000	10002	51	1,05
13 C20	9	1,85	3	14/04/24 15:10:00,000000000	10001	61	2,15
14 C20	9	1,85	3	15/08/24 15:10:00,000000000	10001	63	2,35

Rys. 10 Fragment wyniku (kolumna Liczba_ryb może dezorientować):

```
SELECT * FROM
( SELECT id_lowiska, id_ryby, Round(avg(waga),2) Srednia, count(*) Liczba_ryb FROM rejestr
WHERE id_ryby is not NULL GROUP BY id_lowiska, id_ryby HAVING count(*)>=3)
JOIN rejestr USING (id_lowiska, id_ryby) ORDER BY 1, 2;
```

Krok 3. Dokonanie selekcji poprzez zdefiniowanie warunku, w którym to zostaną odszukane połowy o wadze złowionych ryb większej od średniej (obliczonej dla danego gatunku na danym łowisku).

ID_LOWISKA	ID_RYBY	SREDNIA	LICZBA_RYB	DATA CZAS	ID_WEDKARZA	DLUGOSC	WAGA
1	13	10	2,53	315/07/21 23:55:00,000000000	10004	68	3,4
2	C11	1	1,65	315/06/14 05:10:00,000000000	10001	51	2,6
3	C11	3	1,1	515/04/24 15:45:00,000000000	10001	56	1,8
4	C11	3	1,1	516/06/29 12:10:00,000000000	10010	46	1,2
5	C20	9	1,85	314/04/24 15:10:00,000000000	10001	61	2,15
6	C20	9	1,85	315/08/24 15:10:00,000000000	10001	63	2,35
7	C20	15	0,32	316/02/24 14:10:00,000000000	10005	29	0,4
8	C36	1	1,32	315/06/26 12:00:00,000000000	10008	46	1,55
9	C40	9	1,91	414/04/28 12:15:00,000000000	10003	71	3,85

Rys. 11 Wynik:

SELECT * FROM

(*SELECT id_lowiska, id_ryby, Round(avg(waga),2) Srednia, count(*) Liczba_ryb FROM rejestr WHERE id_ryby is not NULL GROUP BY id_lowiska, id_ryby HAVING count(*)>=3*)

JOIN rejestr USING (id_lowiska, id_ryby)

WHERE waga>srednia

ORDER BY 1, 2;

Krok 4. Dołączenie tabel ryba, lowisko i wedkarz w celu wyświetlenia stosownych szczegółów połowów.

LOWISKO	GATUNEK	WAGA	SREDNIA	NAZWISKO	Dzien polowu
1	Ostrowy KARP	1,55	1,32	Adamczyk	15/06/27
2	Pilica SZCZUPAK	3,85	1,91	Polak	14/04/29
3	Poraj KARP	2,6	1,65	Kowalski	15/06/14
4	Poraj LESZCZ	1,2	1,1	Zachorski	16/06/30
5	Poraj LESZCZ	1,8	1,1	Kowalski	15/04/25
6	Turawa SANDACZ	3,4	2,53	Drozd	15/07/22
7	Warta OKON	0,4	0,32	Pawlak	16/02/25
8	Warta SZCZUPAK	2,35	1,85	Kowalski	15/08/25
9	Warta SZCZUPAK	2,15	1,85	Kowalski	14/04/25

Rys. 12 Wynik:

SELECT l.nazwa Lowisko, b.nazwa Gatunek, waga, srednia, nazwisko, Round(dataczas) as "Dzien polowu"
FROM

(*SELECT id_lowiska, id_ryby, Round(avg(waga),2) Srednia, count(*) Liczba_ryb FROM rejestr WHERE id_ryby is not NULL GROUP BY id_lowiska, id_ryby HAVING count(*)>=3*) JOIN rejestr USING (id_lowiska, id_ryby) JOIN ryba b USING(id_ryby) JOIN lowisko l USING(id_lowiska) JOIN wedkarz w USING(id_wedkarza)

WHERE waga>srednia ORDER BY 1, 2;

6. Na podstawie danych znajdujących się w tabelach rejestr, ryba, wedkarz i lowisko wyświetl informacje dotyczące najcięższej ryby złowionej przez danego wędkarza w danym roku kalendarzowym.

NAZWISKO	ROK	Waga ryby	GATUNEK	Złowiona w lowisku	Dzien polowu
1	Adamczyk	2014	0,7 LIN	Zielona	14/12/12
2	Adamczyk	2015	2,35 LESZCZ	Pilica	15/07/27
3	Drozd	2014	2,3 AMUR	Zielona	14/11/15
4	Drozd	2015	9,15 SUM	Odra	15/09/04
5	Drozd	2016	0,4 PLOC	Pilica	16/01/25
6	Kowalski	2014	2,15 SZCZUPAK	Warta	14/04/25
7	Kowalski	2015	2,6 KARP	Poraj	15/06/14

Rys. 13 Fragment wyniku:

SELECT nazwisko, EXTRACT(YEAR FROM dataczas) rok, waga as "Waga ryby", b.nazwa Gatunek, l.nazwa as "Złowiona w lowisku", Round(dataczas) as "Dzien polowu"

FROM rejestr r JOIN wedkarz w USING (id_wedkarza) JOIN ryba b USING(id_ryby) JOIN lowisko l USING(id_lowiska)

WHERE (EXTRACT(YEAR FROM dataczas), id_wedkarza, waga) IN (*SELECT EXTRACT(YEAR FROM dataczas), id_wedkarza, max(waga) FROM rejestr JOIN lowisko USING(id_lowiska) WHERE id_ryby is not NULL and id_okregu like 'PZW%' GROUP BY EXTRACT(YEAR FROM dataczas), id_wedkarza*)

ORDER BY 1,2;

Rozszerzenia klauzuli GROUP BY

ROLLUP - dla n kryteriów grupowania daje $n + 1$ dodatkowych zestawień; dodatkowe podsumowania uzyskuje poprzez związanie ostatnich elementów z listy.

Polecenie ROLLUP grupuje dane według wskazanych kolumn w sposób hierarchiczny. Użycie ROLLUP(a, b, c) spowoduje wygenerowanie następujących grupowań: (a, b, c), (a, b), (a) oraz rekordu agregującego cały zbiór. Inaczej mówiąc, użycie w zapytaniu konstrukcji ROLLUP(a, b, c) jest równoważne z wykonaniem czterech zapytań połączonych klauzulą UNION ALL, gdzie każde zapytanie posiada odpowiednią klauzulę GROUP BY ((a, b, c), (a, b), (a), ()). [1]

Przykład

Założenie: nazwiska wędkarzy nie powtarzają się.

NAZWISKO	LOWISKO	GATUNEK	Liczba polowow	Laczna waga	Liczba sztuk
17 Drozd	Mala Panew	SZCZUPAK	1	1	1
18 Drozd	Mala Panew	(null)	1	0	0
19 Drozd	Mala Panew	(null)	2	1	1
20 Drozd	Odra	SUM	1	9,15	1
21 Drozd	Odra	(null)	1	9,15	1
22 Drozd	Ostrowy	KARP	1	1,15	1
23 Drozd	Ostrowy	(null)	1	1,15	1
24 Drozd	Pilica	OKON	2	0,55	2
25 Drozd	Pilica	PLOC	1	0,4	1
26 Drozd	Pilica	SZCZUPAK	1	1,25	1
27 Drozd	Pilica	(null)	4	2,2	4
28 Drozd	Poraj	(null)	1	0	0
29 Drozd	Poraj	(null)	1	0	0
30 Drozd	Turawa	SANDACZ	2	5,8	2
31 Drozd	Turawa	(null)	2	5,8	2
32 Drozd	Zielona	AMUR	1	2,3	1
33 Drozd	Zielona	(null)	1	2,3	1
34 Drozd	(null)	(null)	12	21,6	10
134 Zachorski	Turawa	SANDACZ	1	1,8	1
135 Zachorski	Turawa	(null)	1	1,8	1
136 Zachorski	(null)	(null)	16	21,05	12
137 (null)	(null)	(null)	86	100,95	58

Rys. 14 Fragment wyniku:

```
SELECT nazwisko, l.nazwa Lowisko, b.nazwa Gatunek, count(*) as "Liczba polowow", sum(NVL(waga,0))
as
"Laczna waga", count(id_ryby) as "Liczba sztuk"
FROM rejestr LEFT JOIN ryba b USING(id_ryby) JOIN wedkarz USING(id_wedkarza) JOIN lowisko l
USING(id_lowiska)
GROUP BY ROLLUP(nazwisko, l.nazwa, b.nazwa)
ORDER BY 1, 2, 3, 4;
```

CUBE - dla n kryteriów grupowania daje 2^n zestawień;

Polecenie CUBE działa w sposób podobny do polecenia ROLLUP, z tą różnicą, iż przy grupowaniu danych uwzględniane są wszystkie kombinacje wskazanych kolumn. Użycie CUBE(a, b, c) spowoduje więc wygenerowanie następujących grupowań: (a, b, c), (a, b), (a, c), (b, c), (a), (b), (c) oraz rekordu agregującego cały zbiór. [1]

Przykład. Zmodyfikowany przykład z Rys. 10

NAZWISKO	LOWISKO	GATUNEK	Liczba polowow	Laczna waga	Liczba sztuk	
22	Drozd	Mala Panew	SZCZUPAK	1	1	1
23	Drozd	Mala Panew	(null)	1	0	0
24	Drozd	Mala Panew	(null)	2	1	1
25	Drozd	Odra	SUM	1	9,15	1
26	Drozd	Odra	(null)	1	9,15	1
27	Drozd	Ostrowy	KARP	1	1,15	1
28	Drozd	Ostrowy	(null)	1	1,15	1
29	Drozd	Pilica	OKON	2	0,55	2
30	Drozd	Pilica	PLOC	1	0,4	1
31	Drozd	Pilica	SZCZUPAK	1	1,25	1
32	Drozd	Pilica	(null)	4	2,2	4
33	Drozd	Poraj	(null)	1	0	0
34	Drozd	Poraj	(null)	1	0	0
35	Drozd	Turawa	SANDACZ	2	5,8	2
36	Drozd	Turawa	(null)	2	5,8	2
37	Drozd	Zielona	AMUR	1	2,3	1
38	Drozd	Zielona	(null)	1	2,3	1
39	Drozd	(null)	AMUR	1	2,3	1
40	Drozd	(null)	KARP	1	1,15	1
41	Drozd	(null)	OKON	2	0,55	2
42	Drozd	(null)	PLOC	1	0,4	1
43	Drozd	(null)	SANDACZ	2	5,8	2
44	Drozd	(null)	SUM	1	9,15	1
45	Drozd	(null)	SZCZUPAK	2	2,25	2
46	Drozd	(null)	(null)	2	0	0
47	Drozd	(null)	(null)	12	21,6	10
209	(null)	Pilica	LESZCZ	1	2,35	1
210	(null)	Pilica	OKON	2	0,55	2
211	(null)	Pilica	PLOC	1	0,4	1
212	(null)	Pilica	SZCZUPAK	4	7,65	4
213	(null)	Pilica	(null)	3	0	0
214	(null)	Pilica	(null)	11	10,95	8
256	(null)	(null)	SANDACZ	8	17,5	8
257	(null)	(null)	SUM	1	9,15	1
258	(null)	(null)	SWINKA	1	0,35	1
259	(null)	(null)	SZCZUPAK	13	25,05	13
260	(null)	(null)	(null)	28	0	0
261	(null)	(null)	(null)	86	100,95	58

Rys. 15 Fragment wyniku:

```

SELECT nazwisko, l.nazwa Lowisko, b.nazwa Gatunek, count(*) as "Liczba polowow", sum(NVL(waga,0))
as
    "Laczna waga", count(id_ryby) as "Liczba sztuk"
FROM rejestr LEFT JOIN ryba b USING(id_ryby) JOIN wedkarz USING(id_wedkarza) JOIN lowisko l
    USING(id_lowiska)
GROUP BY CUBE(nazwisko, l.nazwa, b.nazwa)
ORDER BY 1, 2, 3, 4;

```

GROUPING SETS - pozwala na jednoznaczne określenie, które z dodatkowych zestawień chcemy uzyskać.

Polecenie **GROUPING SETS** działa analogicznie do poleceń **ROLLUP** i **CUBE**, ale daje nam możliwość jawnego określenia wszystkich grupowań jakie mają być uwzględnione w wynikowym zbiorze. Poszczególne grupy wymieniane są po przecinku, przy czym jeżeli w ich skład wchodzi kilka kolumn muszą być ujęte w nawiasy. Puste nawiasy () oznaczają rekord ogólny dla całego zbioru. [1]

Przykład 1. Poniższe zapytanie zwróci wynik dokładnie taki sam jaki zwróciło zapytanie z zastosowaniem polecenia **ROLLUP** (patrz Rys. 10).

```
SELECT nazwisko, l.nazwa Lowisko, b.nazwa Gatunek, count(*) as "Liczba polowow", sum(NVL(waga,0))
as
    "Laczna waga", count(id_ryby) as "Liczba sztuk"
FROM rejestr LEFT JOIN ryba b USING(id_ryby) JOIN wedkarz USING(id_wedkarza) JOIN lowisko l
    USING(id_lowiska)
GROUP BY GROUPING SETS((nazwisko, l.nazwa, b.nazwa), (nazwisko, l.nazwa), (nazwisko), ())
ORDER BY 1, 2, 3, 4;
```

Przykład 2

ID_OKREGU	NAZWISKO	Liczba polowow	
1	PZW Opole	Drozd	4
2	PZW Opole	Polak	1
3	PZW Opole	Zachorski	6
4	PZW Opole	(null)	11
5	PZW Katowice	Wilk	3
6	PZW Katowice	Polak	3
7	PZW Katowice	Pawlak	2
8	PZW Katowice	(null)	8
9	PZW Czestochowa	Drozd	2
10	PZW Czestochowa	Nowak	2
11	PZW Czestochowa	Polak	1
12	PZW Czestochowa	Pawlak	3
13	PZW Czestochowa	Wolski	3
14	PZW Czestochowa	Adamczyk	5
15	PZW Czestochowa	Kowalski	5
16	PZW Czestochowa	Kownacki	5
17	PZW Czestochowa	(null)	26
18	(null)	(null)	45

efekt dołączenia klauzuli
HAVING count(*)>3

ID_OKREGU	NAZWISKO	Liczba polowow	
1	PZW Opole	Drozd	4
2	PZW Opole	Zachorski	6
3	PZW Opole	(null)	11
4	PZW Katowice	(null)	8
5	PZW Czestochowa	Adamczyk	5
6	PZW Czestochowa	Kowalski	5
7	PZW Czestochowa	Kownacki	5
8	PZW Czestochowa	(null)	26
9	(null)	(null)	45

Rys. 16 Wynik

```
SELECT id_okregu, nazwisko, count(*) as "Liczba polowow"
FROM rejestr LEFT JOIN ryba b USING(id_ryby) JOIN wedkarz USING(id_wedkarza) JOIN lowisko l
    USING(id_lowiska)
WHERE id_okregu like 'PZW%' and Extract(Year from dataczas)=2015
GROUP BY GROUPING SETS((id_okregu), (id_okregu, nazwisko), ()) HAVING count(*)>3;
```

GROUPING - funkcja zwracająca wartość 1 dla wyrażeń zwijanych dla dodatkowych zestawień.

Funkcja **GROUPING** bywa wykorzystywana podczas grupowania danych za pośrednictwem prezentowanych powyżej poleceń. Pozwala ona określić dla każdego wiersza wynikowego zbioru czy dana kolumna jest zagregowana (posiada wartość NULL) czy nie. W przypadku kolumny zagregowanej funkcja zwraca 1, w przeciwnym wypadku 0. [1]

Przykład

Zmodyfikowany przykład zaprezentowany podczas omawiania polecenia ROLLUP.

	NAZWISKO	LOWISKO	GATUNEK	Liczba polowow	Laczna waga	Liczba sztuk
16	Adamczyk	Podsumowanie	Razem	9	8,2	5
17	Drozd	Zielona	AMUR	1	2,3	1
18	Drozd	Zielona	Razem	1	2,3	1
19	Drozd	Turawa	SANDACZ	2	5,8	2
20	Drozd	Turawa	Razem	2	5,8	2
21	Drozd	Poraj	Brak polowu	1	0	0
22	Drozd	Poraj	Razem	1	0	0
23	Drozd	Pilica	SZCZUPAK	1	1,25	1
24	Drozd	Pilica	PLOC	1	0,4	1
25	Drozd	Pilica	OKON	2	0,55	2
26	Drozd	Pilica	Razem	4	2,2	4
27	Drozd	Ostrowy	KARP	1	1,15	1
28	Drozd	Ostrowy	Razem	1	1,15	1
29	Drozd	Odra	SUM	1	9,15	1
30	Drozd	Odra	Razem	1	9,15	1
31	Drozd	Mala Panew	SZCZUPAK	1	1	1
32	Drozd	Mala Panew	Brak polowu	1	0	0
33	Drozd	Mala Panew	Razem	2	1	1
34	Drozd	Podsumowanie	Razem	12	21,6	10
35	Kowalski	Warta	SZCZUPAK	2	4,5	2

Rys. 17 Fragment wyniku

```
SELECT nazwisko,  
       Decode(GROUPING(l.nazwa), 1, ' Podsumowanie', l.nazwa) Lowisko,  
       Decode(GROUPING(b.nazwa), 1, ' Razem ', NVL(b.nazwa,'Brak polowu')) Gatunek,  
       count(*) as "Liczba polowow", sum(NVL(waga,0)) as "Laczna waga", count(id_ryby) as "Liczba  
sztuk"  
FROM rejestr LEFT JOIN ryba b USING(id_ryby) JOIN wedkarz USING(id_wedkarza) JOIN lowisko l  
       USING(id_lowiska)  
GROUP BY ROLLUP(nazwisko, l.nazwa, b.nazwa)  
ORDER BY 1, 2 DESC, 3 DESC, 4;
```

GROUPING_ID

Zastępuje wielokrotne użycie funkcji GROUPING (dla każdej kolumny). Dla wskazanego zestawu kolumn generowana jest postać binarna na podstawie poszczególnych wyników funkcji GROUPING, następnie wynik zwracany jest w postaci wartości typu int.

Przykład

Założmy, że uzyskaliśmy następujące wyniki w danym wierszu:

- GROUPING(a) = 1,
- GROUPING(b) = 1,
- GROUPING(c) = 0,
- GROUPING(d) = 0.

Wywołanie funkcji GROUPING_ID(a, b, c, d) na podstawie wartości binarnej **1100** da wynik **12**. [1]

źródło

[1] <https://mndevnotes.wordpress.com>

Z serii sprawdź jak to działa

```
select * from pracownicy natural join dzialy;
select * from pracownicy join dzialy using (id_dzialu);
select * from pracownicy left join dzialy using (id_dzialu);
select * from pracownicy join dzialy using (id_dzialu) join stanowiska using(stanowisko);
select * from pracownicy natural join stanowiska;
select * from pracownicy right join stanowiska using (stanowisko);
select * from pracownicy join stanowiska using (stanowisko) where placa not between placa_min and
placa_max;
```

```
select (select count(*) from pracownicy) as "Liczba Pracownikow",
(select count(*) from dzialy) as "Liczba dzialow",
(select count(*) from stanowiska) as "Liczba stanowisk",
(select count(*) from pracownicy cross join dzialy) as "Cross join P-Dz",
(select count(*) from pracownicy cross join dzialy cross join stanowiska) as "Cross Join P-Dz-S",
(select count(*) from pracownicy join dzialy using (id_dzialu)) as "Join P-Dz",
(select count(*) from pracownicy left join dzialy using (id_dzialu)) as "Left Join P-Dz",
(select count(*) from pracownicy join stanowiska using(stanowisko)) as "Join P-S",
(select count(*) from pracownicy left join stanowiska using(stanowisko)) as "Left Join P-S",
(select count(*) from pracownicy right join stanowiska using(stanowisko)) as "Right Join P-S",
(select count(*) from pracownicy full join stanowiska using(stanowisko)) as "Full P-S",
(select count(*) from pracownicy join dzialy using(id_dzialu) join stanowiska using (stanowisko)) as "Join P-
Dz-S",
(select count(*) from pracownicy left join dzialy using(id_dzialu) right join stanowiska using (stanowisko)) as
"P Left Dz Right S"
from dual;
```

```
select * from kierowcy join pojazdy on (id_kier=wlasciciel);
```

```
select nr_akt, stanowisko, nazwisko, p.id_dzialu as "id z Prac ", d.id_dzialu as "id z Dzialy"
from pracownicy p join dzialy d on(p.id_dzialu=d.id_dzialu);
```

Z zakresu rozszerzenia klauzuli group by

```
select tryb, stopien, kierunek, rok, count(*) as "liczba studentow" from studenci
group by rollup(tryb, stopien, kierunek, rok);
```

```
select tryb, stopien, kierunek, rok, count(*) as "liczba studentow" from studenci
group by cube(tryb, stopien, kierunek, rok) having count(*)>=50;
```

```
select tryb, stopien, kierunek, rok, count(*) as "liczba studentow" from studenci
group by grouping sets( (tryb, stopien, kierunek, rok), (tryb, stopien, rok), (tryb), (stopien), ());
```

```
select decode(grouping(tryb), 1, 'Podsumowanie', tryb) as tryb, decode(stopien, null, ' ', stopien) as stopien,
decode(kierunek, null, ' ', kierunek) as kierunek, decode(rok, null, ' ', rok) as rok,
count(*) as "liczba studentow" from studenci
group by rollup(tryb, stopien, kierunek, rok);
```

```
select decode(grouping_id(tryb , stopien, kierunek, rok), 0, 'Na danym roku w TSK', 1, 'Na danym kierunku w
TS', 3, 'W ramach danego stopnia w T', 7, 'W danym trybie', 15, 'Ogolem studiuje', 'cos :') as komentarz,
tryb, stopien, kierunek, rok, count(*)as "Liczba studentow" from studenci
group by rollup(tryb, stopien, kierunek, rok);
```