

maj 2017 zad 3.1

Po wykonaniu podanego zapytania SQL do pewnej bazy danych wyniki będą zawsze uporządkowane niemalejąco według pola *nazwa*.

1.	SELECT nazwa, wartosc FROM dane ORDER BY wartosc, nazwa	P	F
2.	SELECT nazwa, wartosc FROM dane ORDER BY nazwa	P	F
3.	SELECT nazwa, sum(wartosc) FROM dane GROUP BY nazwa	P	F
4.	SELECT nazwa, sum(wartosc) FROM dane GROUP BY nazwa ORDER BY nazwa	P	F

maj 2017 zad 3.2

Rozważ następujące zapytanie SQL do pewnej bazy danych:

```
SELECT pesel, COUNT(*)
```

```
FROM samochody
```

```
WHERE pesel NOT IN (SELECT pesel FROM dokumenty_zastrzezone)
```

```
GROUP BY pesel HAVING COUNT(*) > 1
```

Po wykonaniu tego zapytania w odpowiedzi

1.	ten sam numer PESEL może pojawić więcej niż jeden raz.
2.	nie pojawi się żaden numer PESEL, który jest zapisany w tabeli dokumenty_zastrzezone.
3.	otrzymasz tabelę o 2 kolumnach.
4.	przy odpowiednich danych może pojawić się wiersz „82122302134, 1”.

Wskaż zdania prawdziwe dla języka SQL.

1.	W wynikach zapytania postaci <code>SELECT (...) ORDER BY (...)</code> zawsze dostajemy rekordy uporządkowane ściśle rosnąco według wskazanego pola.
2.	Zapytanie <code>UPDATE</code> może zmienić wartości pól w bazie danych.
3.	Zapytanie postaci <code>SELECT * FROM tabela1 WHERE pole LIKE (...)</code> może w pewnych warunkach dać wszystkie rekordy z tabeli <i>tabela1</i> .
4.	Wynik zapytania <code>SELECT * FROM tabela1 JOIN tabela2 ON tabela1.pole = tabela2.pole</code> może być pusty przy niepustych tabelach <i>tabela1</i> oraz <i>tabela2</i> .



W tabeli *T* zapisano wiele rekordów danych zawierających informacje o zawodnikach. Pola rekordu to: *id*, *nazwisko*, *imie*, *plec*, *wzrost*, *numer_startowy*, *punkty*, *id_klubu*.

Polecenie SQL obliczające sumę punktów zawodników z klubu o *id_klubu* równym liczbie 100, może mieć postać:

1	<code>select sum(punkty) as suma from T where id_klubu=100;</code>
2	<code>select avg(punkty) from T where id=100;</code>
3	<code>select punkty as suma from T where id_klubu=100;</code>
4	<code>select sum(punkty) from T where id_klubu=100;</code>

maj
2019



Dana jest tabela PRACOWNICY.

Nr_P	Nazwisko	Imię	Stanowisko	Nr_działu
736	Smitko	Alan	urzędnik	20
7499	Nowak	Kazimierz	sprzedawca	30
7521	Więcek	Mariusz	sprzedawca	30
7566	Jonas	Kamil	kierownik	20
7654	Martin	Leon	sprzedawca	30
7698	Bracki	Bartosz	kierownik	30
7782	Celerek	Agnieszka	kierownik	10
7788	Skotnik	Natalia	analityk	20
7839	King	Mirosława	prezes	10

1.	Wynikiem zapytania SELECT COUNT(Stanowisko) FROM PRACOWNICY; jest Stanowisko 5
2.	Wynikiem zapytania SELECT COUNT(Stanowisko) FROM PRACOWNICY WHERE Stanowisko <> "kierownik"; jest 6
3.	Wynikiem zapytania SELECT Stanowisko, COUNT(*) FROM PRACOWNICY GROUP BY Stanowisko; jest urzędnik 1 sprzedawca 3 kierownik 3 analityk 1 prezes 1
4.	Wynikiem zapytania SELECT COUNT(Stanowisko) FROM PRACOWNICY WHERE Stanowisko LIKE "*nik"; jest 2

Dane są tabele Uczniowie i Oceny. Przeanalizuj i oceń poniższe zapytanie w języku SQL.

```
SELECT Uczniowie.imie, Uczniowie.nazwisko, AVG(Oceny.ocena)
FROM Uczniowie INNER JOIN Oceny ON Uczniowie.id_ucznia = Oceny.id_ucznia
GROUP BY Uczniowie.id_ucznia, Uczniowie.imie, Uczniowie.nazwisko
HAVING AVG(Oceny.ocena) >= 4
ORDER BY AVG(Oceny.ocena), Uczniowie.nazwisko;
```

1.	W wyniku zapytania, przy odpowiednich danych, mogą pojawić się następujące po sobie wiersze: Jan Abacki 4.08 Jan Kowalski 4.85
2.	W wyniku zapytania to samo imię i nazwisko może pojawić się tylko raz, nawet jeśli dwóch uczniów ma takie samo imię i nazwisko.
3.	W wyniku zapytania otrzymamy trzy kolumny z danymi.
4.	Jedynym kryterium określającym kolejność wierszy w odpowiedzi jest średnia ocena.

czerwiec
2019



Zadanie 3.2. (0–1)

	A	B	C	D	E	F	G	H	I	J	K
1		0	1	2	3	4	5	6	7	8	9
2	10	100	121	144	169	196	225	256	289	324	361
3	20	400	441	484	529	576	625	676	729	784	841
4	30	900	961	1024	1089	1156	1225	1296	1369	1444	1521
5	40	1600	1681	1764	1849	1936	2025	2116	2209	2304	2401
6	50	2500	2601	2704	2809	2916	3025	3136	3249	3364	3481
7	60	3600	3721	3844	3969	4096	4225	4356	4489	4624	4761
8	70	4900	5041	5184	5329	5476	5625	5776	5929	6084	6241
9	80	6400	6561	6724	6889	7056	7225	7396	7569	7744	7921
10	90	8100	8281	8464	8649	8836	9025	9216	9409	9604	9801

Powyższą tablicę kwadratów w arkuszu kalkulacyjnym można otrzymać, jeżeli skopiuje się tylko jedną formułę z komórki B2 do pozostałych komórek z zakresu B2:K10.

W tym celu do komórki B2 należy wpisać

1.	$=(\$A2+B\$1)*(\$A2+B\$1)$
2.	$=(A2+B1)*(A2+B1)$
3.	$=(\$A2+B\$1)^2$
4.	$=(\$A\$2+\$B\$1)^2$

W tabeli T zapisano wiele rekordów danych zawierających informacje o zawodnikach. Pola rekordu to: *id*, *nazwisko*, *imie*, *plec*, *wzrost*, *numer_startowy*, *punkty*, *id_klubu*.

Polecenie SQL obliczające średnią punktów zawodników z klubu o *id_klubu* równym liczbie 100, może mieć postać:

1.	<code>select count(punkty) as srednia from T where id_klubu=100;</code>
2.	<code>select avg(punkty) as srednia from T where id=100;</code>
3.	<code>select sum(punkty) from T where id_klubu=100;</code>
4.	<code>select avg(punkty) from T where id_klubu=100;</code>

**czerwiec
2020**



Dana jest tabela pracownicy

nr_p	nazwisko	imie	staz	pensja
736	Smitko	Alan	10	2000
7499	Nowak	Kazimierz	15	3000
7521	Więcek	Mariusz	11	3500
7566	Jonas	Kamil	12	2500
7654	Martin	Leon	20	2300
7698	Bracki	Bartosz	15	1530
7782	Celerek	Agnieszka	12	1680
7788	Skotnik	Natalia	21	2000
7839	King	Mirosław	22	1500

oraz zapytanie SQL:

```
SELECT *  
FROM pracownicy  
WHERE pensja < 2000  
ORDER BY staz DESC;
```

1.	Wynik powyższego zapytania to 3.																				
2.	Wynikiem powyższego zapytania jest zestawienie: <table><tr><th>nr_p</th><th>nazwisko</th><th>imie</th><th>staz</th><th>pensja</th></tr><tr><td>7839</td><td>King</td><td>Mirosław</td><td>22</td><td>1500</td></tr><tr><td>7698</td><td>Bracki</td><td>Bartosz</td><td>15</td><td>1530</td></tr><tr><td>7782</td><td>Celerek</td><td>Agnieszka</td><td>12</td><td>1680</td></tr></table>	nr_p	nazwisko	imie	staz	pensja	7839	King	Mirosław	22	1500	7698	Bracki	Bartosz	15	1530	7782	Celerek	Agnieszka	12	1680
nr_p	nazwisko	imie	staz	pensja																	
7839	King	Mirosław	22	1500																	
7698	Bracki	Bartosz	15	1530																	
7782	Celerek	Agnieszka	12	1680																	
3.	Wynikiem zapytania będą wiersze z tabeli pracownicy, zawierające wszystkie dane z tej tabeli dotyczące pracowników, dla których wartość z kolumny pensja jest mniejsza niż 2000, posortowane nierosnąco według parametru staz.																				
4.	Wynikiem powyższego zapytania jest zestawienie: <table><tr><th>nr_p</th><th>staz</th><th>pensja</th></tr><tr><td>7782</td><td>12</td><td>1680</td></tr><tr><td>7698</td><td>15</td><td>1530</td></tr><tr><td>7839</td><td>22</td><td>1500</td></tr></table>	nr_p	staz	pensja	7782	12	1680	7698	15	1530	7839	22	1500								
nr_p	staz	pensja																			
7782	12	1680																			
7698	15	1530																			
7839	22	1500																			

**marzec
2021**



W bazie danych istnieje tabela *oceny*(*id_oceny*, *id_ucznia*, *przedmiot*, *ocena*), zawierająca następujące dane:

id_oceny	id_ucznia	przedmiot	ocena
1	1	matematyka	3
2	1	informatyka	4
3	1	fizyka	2
4	2	matematyka	6
5	2	fizyka	3
6	2	informatyka	5
7	3	matematyka	4
8	3	fizyka	2
9	3	informatyka	3

1.	Wynikiem zapytania SELECT COUNT(id_ucznia) FROM oceny; jest 3
2.	Wynikiem zapytania SELECT COUNT (id_ucznia) FROM oceny WHERE przedmiot="fizyka"; jest 3
3.	Wynikiem zapytania SELECT COUNT(przedmiot) FROM oceny; jest 9
4.	Wynikiem zapytania SELECT COUNT(przedmiot) FROM oceny WHERE ocena > 3; jest 4

Piątek godzina 15.30 to nienajlepszy czas na myślenie. O tej porze można gadać i robić głupoty straszne :)

Bycie inteligentnym jest zobowiązujące.
Wszyscy oczekują, że palniesz
coś głupiego.



Zadanie 3.3. (0–1)

Różnica $1011101_2 - 10111_2$ dwóch liczb zapisanych w systemie binarnym jest:

1.	mniejsza niż 100111_2	P	F
2.	równa 1000110_2	P	F
3.	większa niż 10111_2	P	F
4.	równa 1001000_2	P	F

$$\begin{array}{r} 1011101 \\ - 10111 \\ \hline 1000110 \end{array} \quad \begin{array}{r} 93 \\ - 23 \\ \hline 70 \end{array}$$



Zadanie 3.3. (0–1)

N bazie danych istnieje tabela *mandaty*(numer, id_osoby, punkty) zawierająca następujące dane:

numer	id_osoby	punkty
1	1	5
2	1	14
3	2	20
4	3	21
5	2	1
6	1	2

1.	Wynikiem zapytania: <pre>SELECT id_osoby, sum(punkty) FROM mandaty GROUP BY id_osoby HAVING sum(punkty) > 5</pre> jest zestawienie: <pre>1 14 2 20 3 21</pre>	P	F
2.	Wynikiem zapytania: <pre>SELECT id_osoby, sum(punkty) FROM mandaty GROUP BY id_osoby</pre> jest zestawienie: <pre>1 21 2 21 3 21</pre>	P	F
3.	Wynikiem zapytania: <pre>SELECT numer + punkty FROM mandaty</pre> jest <pre>86</pre>	P	F
4.	Wynikiem zapytania: <pre>SELECT count(punkty) FROM mandaty WHERE punkty = 21</pre> jest	P	F

maj 2022