

1. Napisz polecenie SQL:

- 1 Wyświetlić wszystkie apteki które nie są w obsłudze KAS i KAMS.
- 2 Ile aptek dokonało zakupu dnia 30.03.2019?
- 3 Podać dane lokalizacji apteki która nie dokonała zakupu w marcu 2019?
- 4 Podać wartość sprzedaży paracetamolu w lokalizacji Wrocław za 2019 rok.
- 5 Ile sztuk tabletek zostało sprzedanych w 2018 roku?
- 6 Podać sumę dla każdego produktu, oraz podsumowanie w jednym zapytaniu.
- 6 Zmienić typ danych BLOZ w tabeli PRODUKTY na liczbowy (liczby całkowite). [nieobowiązkowe]

Moja realizacja:



1. Wyświetlić wszystkie apteki które nie są w obsłudze KAS i KAMS.

```
select * from posrednicy p
where
    obsluga != 'KAS' and
    obsluga != 'KAMS'
```

123 nwbr	ABC ulica	ABC kod_pocztowy	ABC miejscowosc	ABC obsluga
909 734	W. SIKORSKIEGO 59	95-015	GŁÓWNO	B
948 878	BOGUSZOWSKA 61 A	54-046	WROCŁAW	E
941 632	FREDRY 7	35-005	RZESZÓW	D

2. Ile aptek dokonało zakupu dnia 30.03.2019

```
select count(distinct(zakupy.nwbr))
from zakupy
where zakupy."DATA" = '2019-03-30'
```

123 count
3

3. Podać dane lokalizacji apteki która nie dokonała zakupu w marcu 2019?

```
select
    distinct(zakupy.nwbr), posrednicy.ulica
from posrednicy
left join zakupy on zakupy.nwbr = posrednicy.nwbr
where date_part('month',zakupy."DATA") !=3
and date_part('year',zakupy."DATA") !=2019
```

	123 nwbr	ABC ulica
1	941 632	FREDRY 7
2	948 878	BOGUSZOWSKA 61 A

4. Podać wartość sprzedaży paracetamolu w lokalizacji Wrocław za 2019 ro

```
select sum(z.ilsc::int * p.cena)
  from produkty p
 left join zakupy z on p.bloz::int = z.bloz
 where
  p.nazwa like 'paracetamol%' and
  date_part('year',z."DATA") = '2019'
```

k.

123 sum
1 158,9399981499

5. Ile sztuk tabletek zostało sprzedanych w 2018 roku?

Podać sumę dla każdego produktu, oraz podsumowanie w jednym zapytaniu.

```
select produkty.nazwa, zakupy.ilsc
  from produkty,zakupy
 where
  produkty.bloz::int = zakupy.bloz and
  produkty.typ like '%abletki' and
  date_part('year',zakupy."DATA") = '2018'
```

ABC nazwa	123 ilsc
1 paracetamol 20 tabl.	2

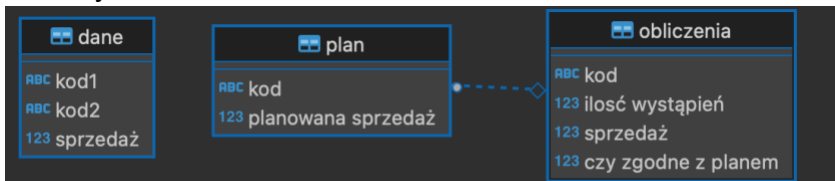
6. Zmienić typ danych BLOZ w tabeli PRODUKTY na liczbowy (liczby całkowite). [nieobowiązkowe]

```
alter table produkty alter column bloz type integer using (produkty.bloz::integer)
select * from produkty.bloz p
```

Name	Value
Updated Rows	0
Query	alter table produkty alter column bloz type integer using (produkty.bloz::integer)
Finish time	Fri Sep 23 20:53:08 CEST 2022

7. Podczas wykonywania zadań proszę nie zmieniać układu tabel i układu wierszy.
- 7.1. W tabeli 'Obliczenia' proszę wyliczyć ilość wystąpień każdego z kodów (kolumna "kod") i sumę sprzedaży dla każdego z kodów(kod1 i kod2) z tabeli dane (plan.kod = concat(dane.kod1, dane.kod2) .
- 7.2. W tabeli 'Obliczenia' w kolumnie "czy zgodne z planem" proszę wprowadzić '1' jeśli sprzedaż jest zgodna z planem lub '0' jeśli nie zgodna.

Moja realizacja:



```
/* create cte where concatenate rows kod1, kod2 in one row; also aggregate by summ
 * dane sprzedaż and the quantity of occurrences in the table of each concatenated code value */
with cte_concat_kod as(
    select
        concat(dane.kod1, dane.kod2) as kod,
        count(dane.sprzedaż) as qty,
        sum(dane.sprzedaż) as spr
    from dane
group by kod
),
/* joined table plan with plans values for each code*/
cte_2 as(
    select
        cte_concat_kod.kod,
        qty,
        spr,
        plan."planowana sprzedaż" as plan_spr
    from cte_concat_kod
left join plan
    on cte_concat_kod.kod = plan.kod
)
/* update table with calculated values from cte_2; also calculated the execution of the plan */
update obliczenia
set
    "ilość wystąpień" = cte_2.qty,
    "sprzedaż" = cte_2.spr,
    "czy zgodne z planem" = case
        when cte_2.spr >= cte_2.plan_spr then 1
        else 0
    end
from cte_2
where obliczenia.kod = cte_2.kod

select * from obliczenia
```

	ABC kod	123 ilość wystąpień	123 sprzedaż	123 czy zgodne z planem
1	DR824	1	5 639	1
2	DR401	1	295	1
3	DR880	3	3 588	1
4	ER632	1	457	1
5	ER351	1	3 582	1
6	DR205	1	3 606	1
7	CP569	3	2 945	1
8	ER753	2	1 447	1
9	CP869	1	1 521	1
10	ER171	2	6 353	1
11	DR869	1	1 646	1
12	CP791	1	2 179	1
13	ER698	1	5 556	1
14	CP891	1	647	0
15	DR751	1	3 825	1
16	CP118	1	3 035	1
17	ER914	1	2 756	1
18	DR405	3	2 949	1
19	ER177	1	1 631	0
20	DR605	1	3 887	1
21	DR584	1	3 240	1
22	CP548	1	2 852	1
23	ER945	2	6 332	1
24	DR104	3	6 317	0
25	DR768	3	1 899	1
26	CP254	1	2 592	1
27	DR151	1	3 401	1