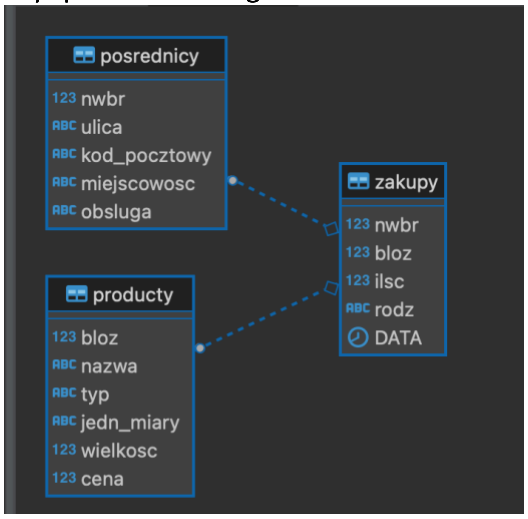
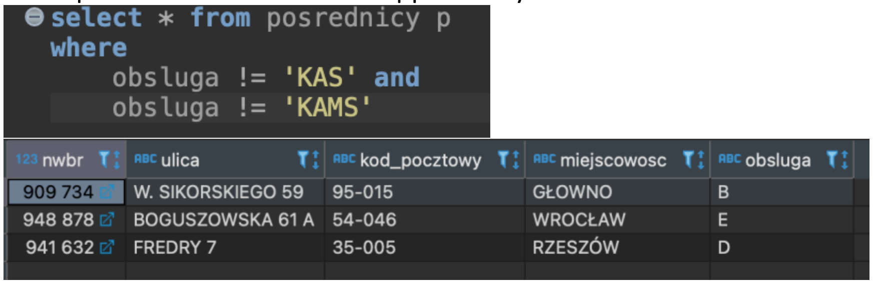
1. **Write SQL queries:**

1 View all pharmacies that are not supported by KAS and KAMS.
2 How many pharmacies made purchases on 03/30/2019?
3 Provide the location data of the pharmacy that did not make a purchase in March 2019?
4 Provide the value of paracetamol sales in Wroclaw for 2019.
5 How many tablets were sold in 2018? Provide the sum for each product, and a summary in one query.
6 Change the BLOZ data type in the PRODUCTS table to numeric (integers). [optional]

My queries in PostgreSQL:



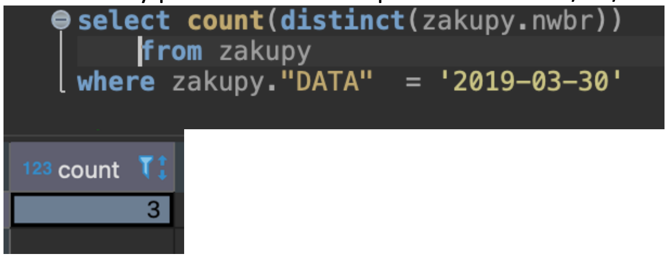
1. View all pharmacies that are not supported by KAS and KAMS



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

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

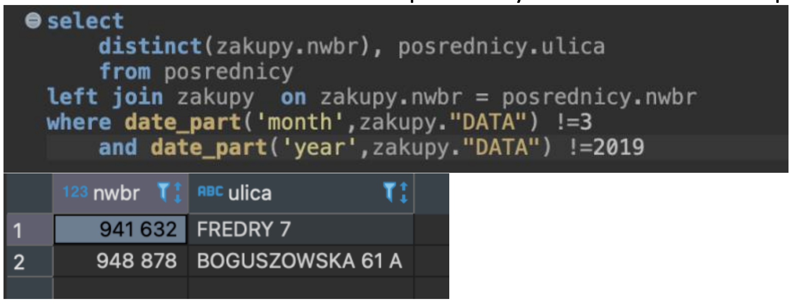
2. How many pharmacies made purchases on 03/30/2019?



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

count
3

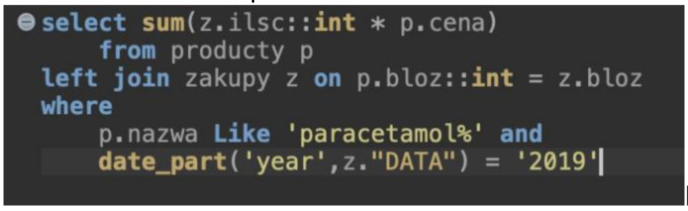
3. Provide the location data of the pharmacy that did not make a purchase in March 2019?



```
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
```

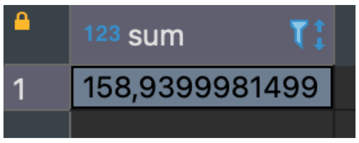
	nwbr	ulica
1	941 632	FREDRY 7
2	948 878	BOGUSZOWSKA 61 A

4. Provide the value of paracetamol sales in Wroclaw for 2019



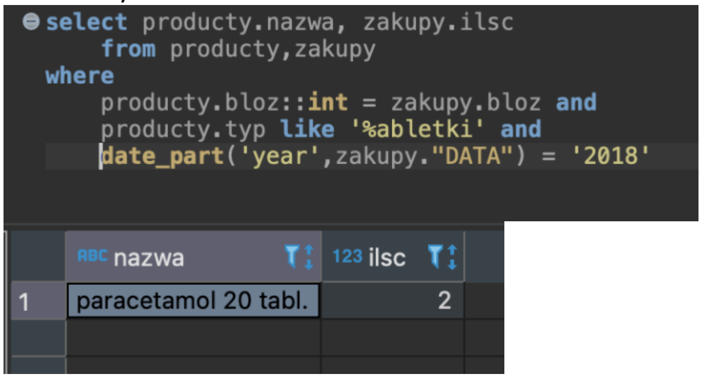
```
select sum(z.ilsc::int * p.cena)
    from producty p
left join zakupy z on p.bloz::int = z.bloz
where
    p.nazwa Like 'paracetamol%' and
    date_part('year',z."DATA") = '2019'
```

k.



```
   123 sum
1  158,9399981499
```

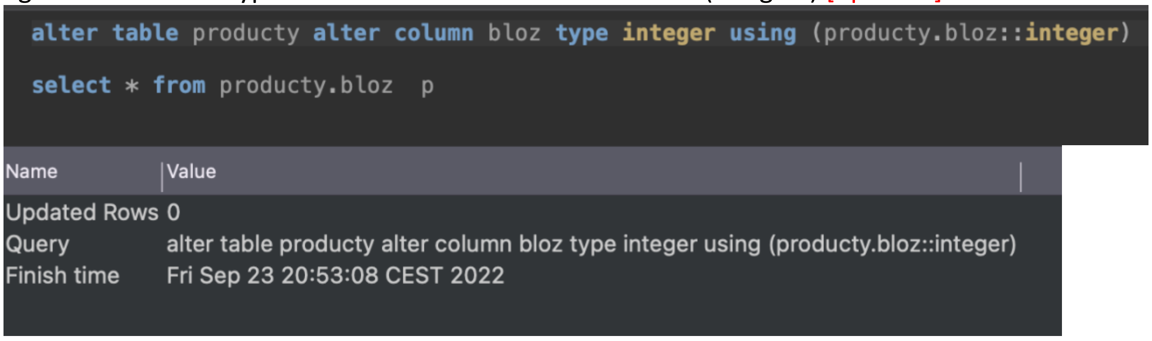
5. How many tablets were sold in 2018? Provide the sum for each product, and a summary in one query



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

```
   nazwa                 ilsc
1  paracetamol 20 tabl.     2
```

Change the BLOZ data type in the PRODUCTS table to numeric (integers) [optional]



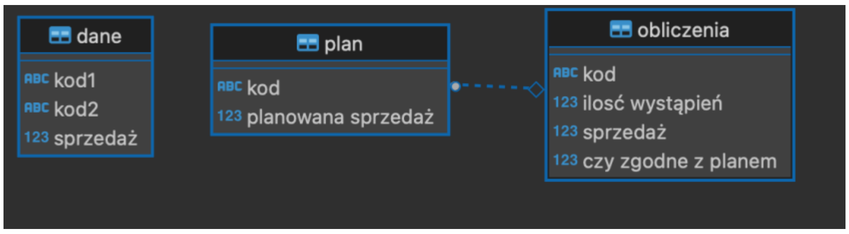
```
alter table producty alter column bloz type integer using (producty.bloz::integer)

select * from producty.bloz  p
```

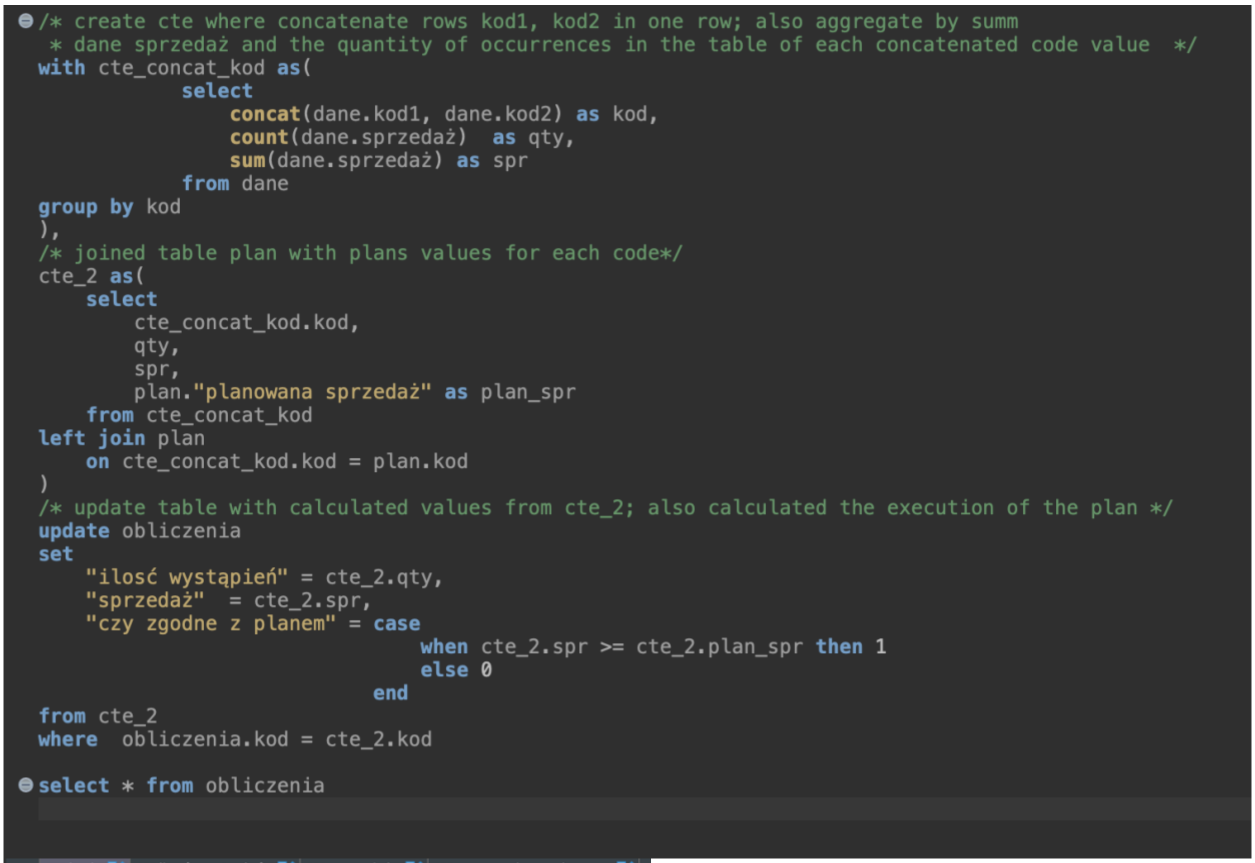
```
Name          Value
Updated Rows  0
Query         alter table producty alter column bloz type integer using (producty.bloz::integer)
Finish time   Fri Sep 23 20:53:08 CEST 2022
```

6. During the execution of tasks, please do not change the table layout

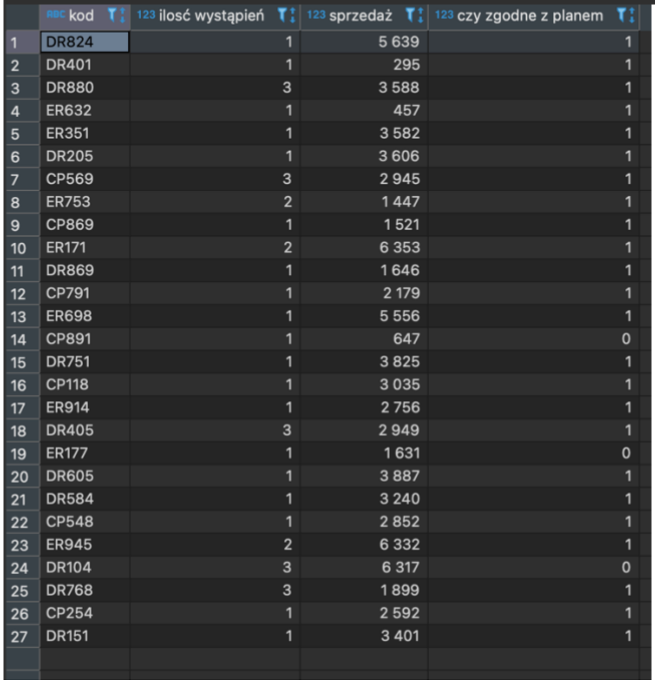
6.1. In the *Obliczenia* (*Calculations*) table, please calculate the number of occurrences of each code (“kod”) and the sum of sales (“sprzedaż” for each of the codes (column H). “kod”= as concatenate “kod1” + “kod2”, for example “kod” = ‘TP368’ and “kod1” = ‘TP’ and “kod2”= ‘368’

6.2. In the table *Obliczenia* (*Calculations*) in column “Czy zgodne z planem” (“plan completed?”), please enter ‘1’ if the sale is consistent with the plan or ‘0’ if not.





```sql
/* 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
    "ilosć 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
```



	kod	ilość wystąpień	sprzedaż	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