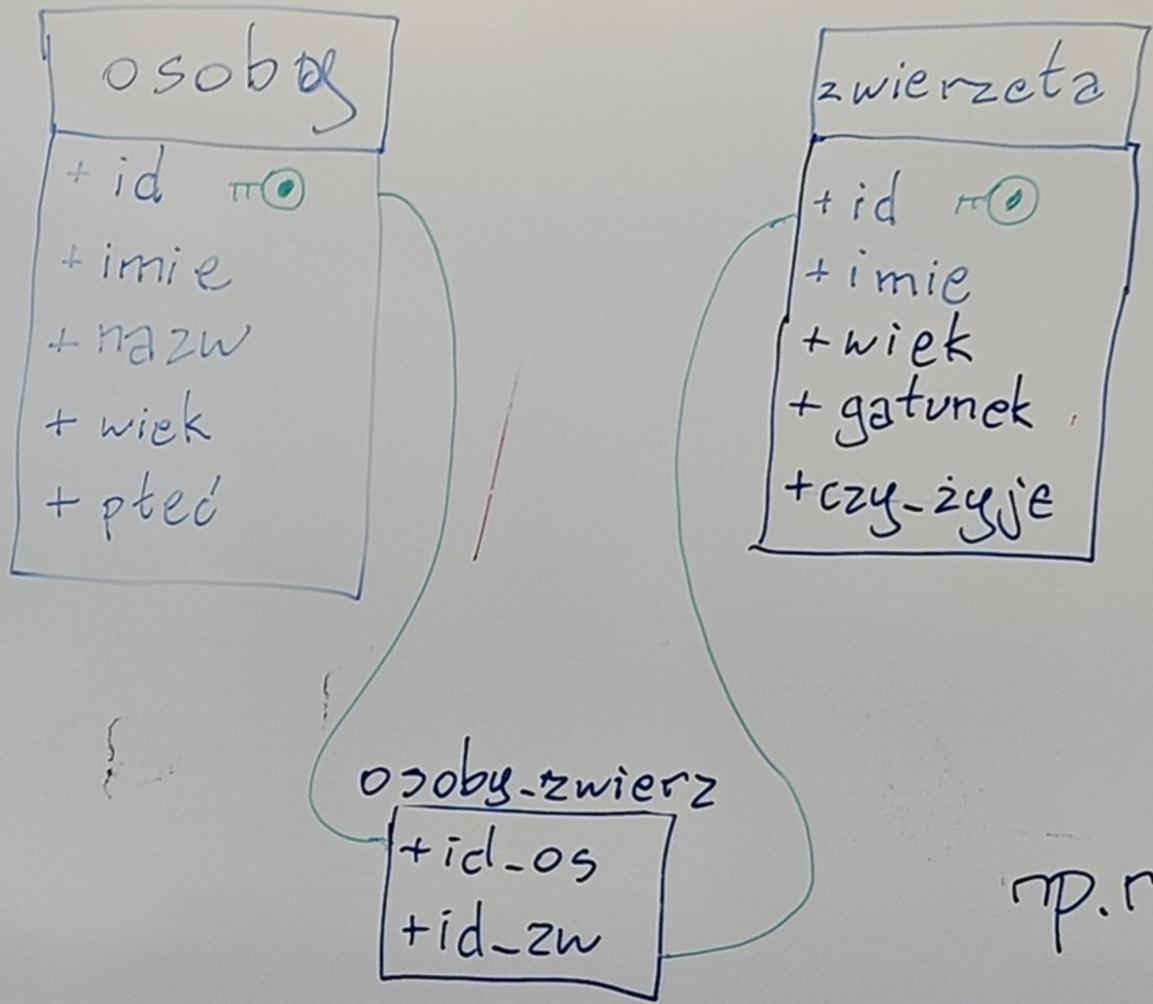
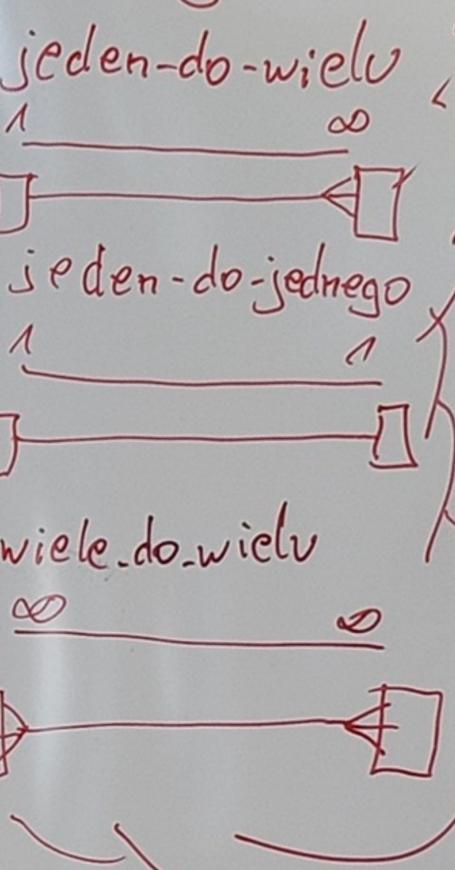
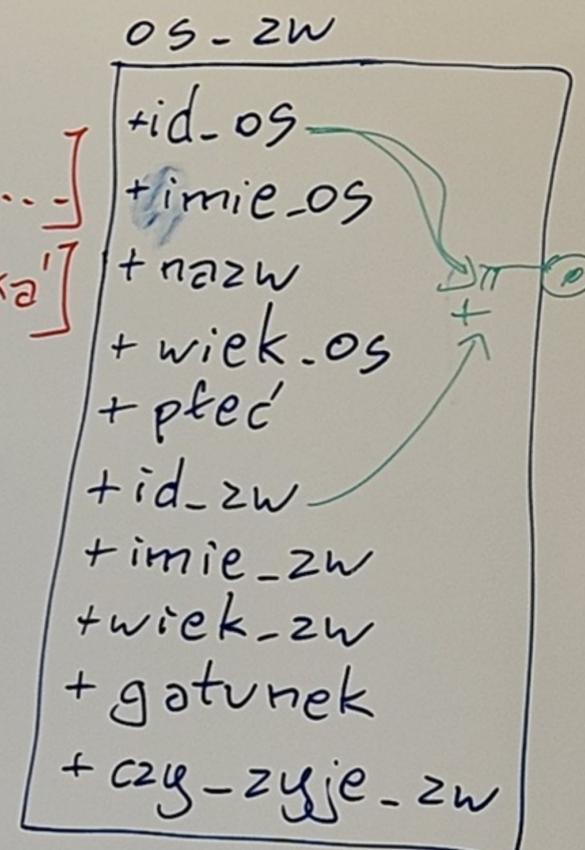


relacje:



{  
 'kot': ['Mruczek' ...]  
 'pies': ['Aron', 'Kropka']  
 'papuga':  
 }



unikatowy

# Schemat bazy danych

## Modelowanie danych

id-os	id-zw
3	8
16	1

np. random.choice(["ma zwierze", "nie ma zwierz"],  
 p=[0.8, 0.2])

wygodniej:

```

dane_osob = gen_osoby(...)
dane_zwierz = gen_zwierz(...)
...
kursor.execute("INSERT INTO osoby VALUES
  dane_osob")
...
kursor.execute("INSERT INTO zwierzeta
  dane_zwierz")
  
```

obsługuje  
złączenie  
tabel

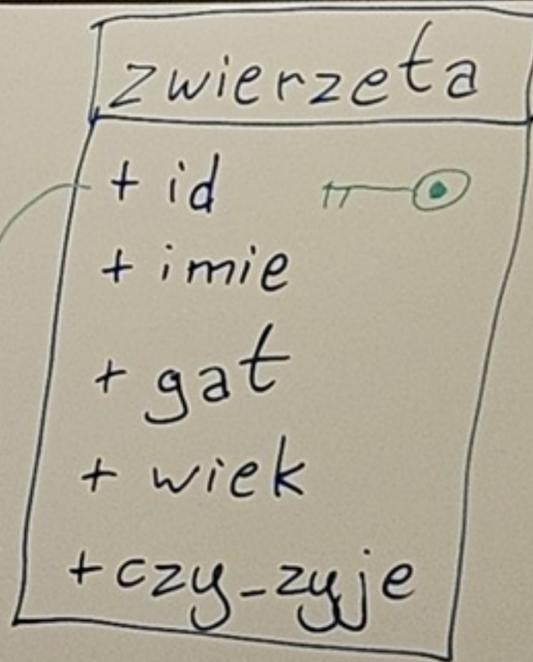
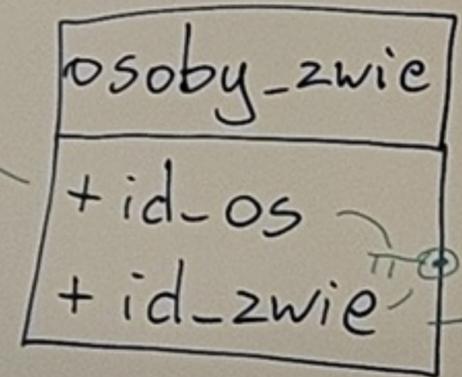
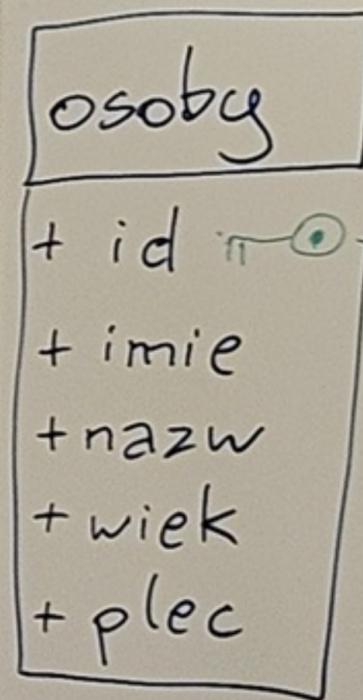
```
SELECT zwierzeta.imie, zwierzeta.gat  
FROM osoby, osoby_zwie, zwierzeta
```

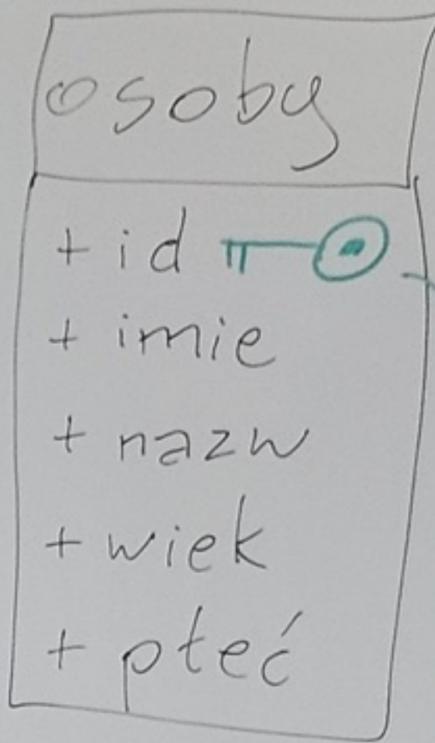
```
WHERE osoby.id = osoby_zwie.id_os AND  
zwierzeta.id = osoby_zwie.id_zwie AND
```

osoby.id

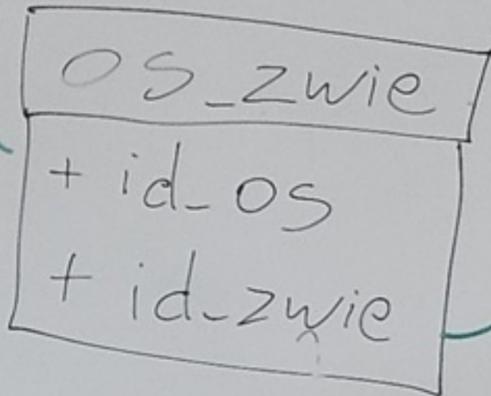
```
{ osoby.imie = 'Piotr' AND  
osoby.nazw = 'Prokopiak'
```

ZADANIE - WYZWANIE: usun' z bazy martwe zwierzeta

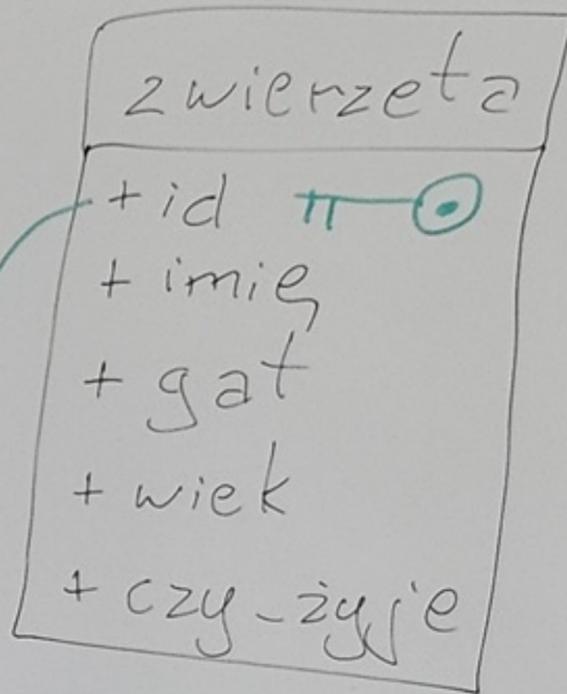




= 30



= 20



= 50

Program ma być elastyczny!

Zastanów się:

- jak z 50 zwierząt wylosować 20 bez powtórek?

RIGHT OUTER

LEFT

SELECT osoby.imie, osoby.nazw, zwierzeta.imie,  
zwierzeta.gat

FROM osoby, os\_zwie, zwierzeta

WHERE osoby.id = os\_zwie.id\_os AND

os\_zwie.id\_zwie = zwierzeta.id

SELECT o.imie, o.nazw, z.imie, z.gat

FROM osoby AS o, os AS os\_zwie, zwierzeta AS z

WHERE o.id = os.id\_os AND os.id\_zwie = z.id AND

o.imie = "Piotr"

W domu:

jakie są preferencje ptci  
w zakresie gatunków zw.

- 1 za pytaniem

FROM osoby INNER JOIN os\_zwie

ON osoby.id = os\_zwie.id\_os

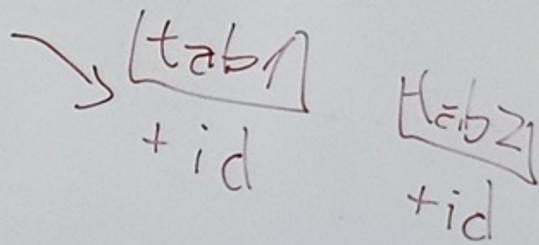
INNER JOIN zwierzeta

ON os\_zwie.id\_zwie = zwierzeta.id

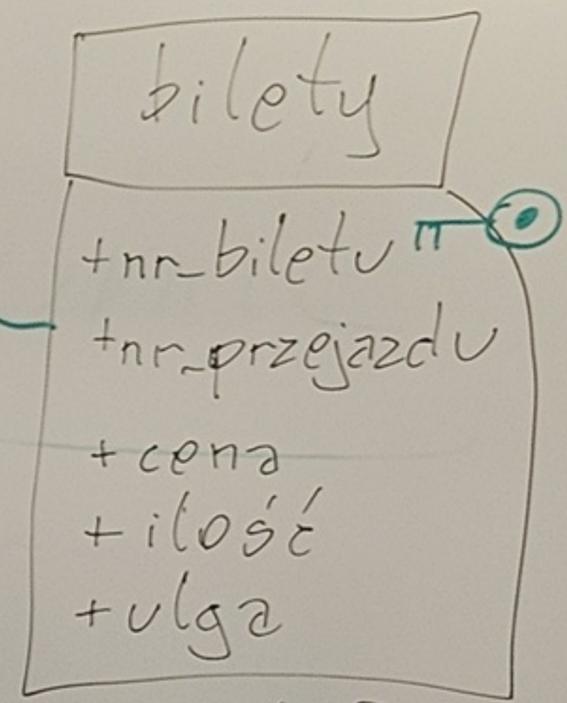
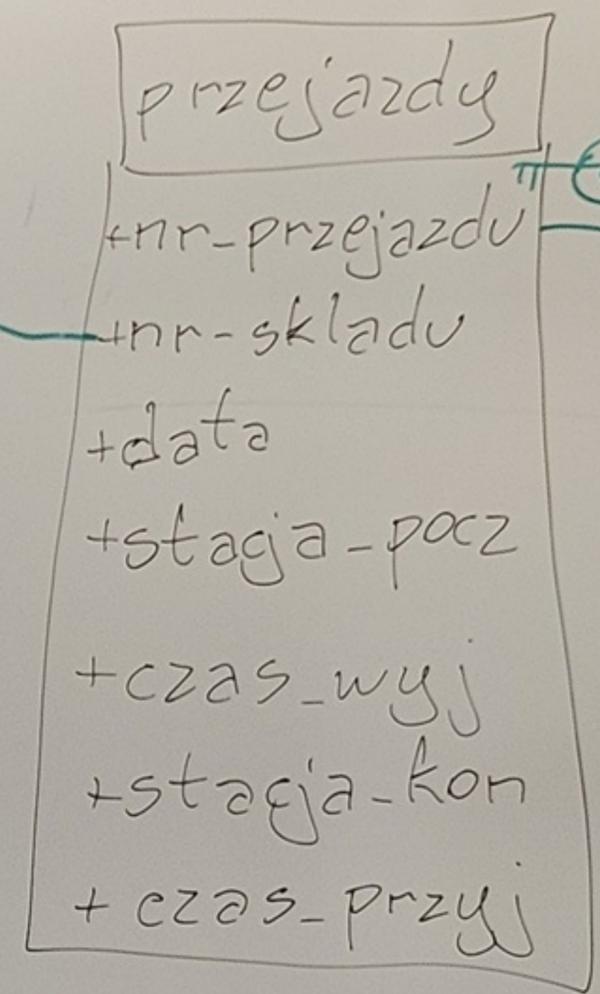
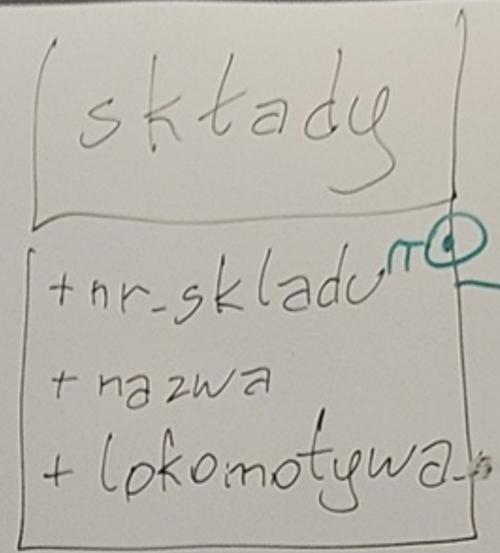
WHERE osoby.imie = "Piotr"

FROM tab1 JOIN tab2

ON id



model  
lokon



101; Bachus; EP-07

1.001; 501; 180; 1; 0,34

501; 101; 15.07.2022; Wrocław Gł.; 08:12:00; Przemysł; 14:55

które składy nie wyjechały (nie używane w żadnym przejeździe)

(SELECT nr\_skladu FROM przejazdy)

Ile modeli lokomotyw znajduje się w składach?

SELECT COUNT(\*) DISTINCT  
FROM (SELECT lokomotywa FROM składy)

Numery przejazdów na które nikt nie kupił biletów.

---

Posortowane malejąco w/g ilości numery biletów na przejazd 404.

---

00 z której stacji wyjechało najwięcej pasażerów?

W domu

SELECT . . . LEFT  
FROM składy (INNER) JOIN  
przejazdy  
ON nr-skladu

SELECT \* FROM przejazdy

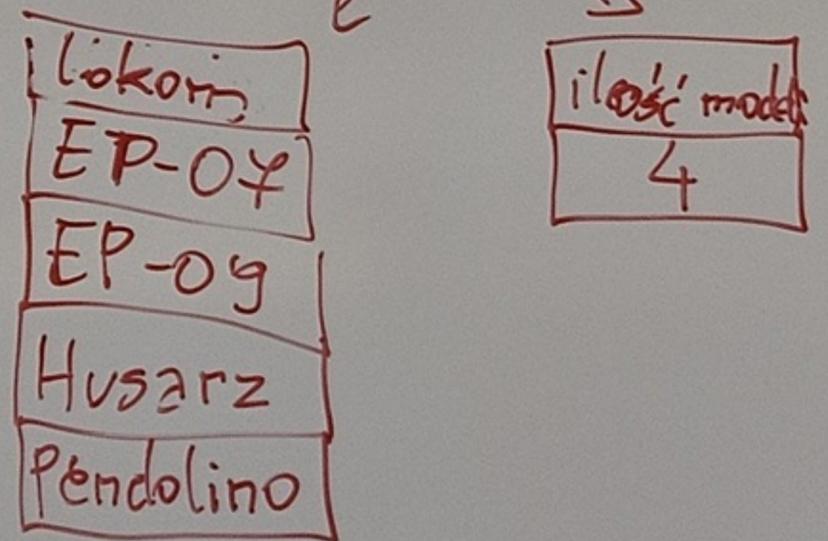
SELECT \* FROM przejazdy JOIN bilety

ON przejazdy.nr-przejazdu = bilety.nr-przejazdu

Zad. 1. Numery biletów na przejazd 504,  
posortowane w/g ilości

! Przy importowaniu danych kontroluj typy! "504"

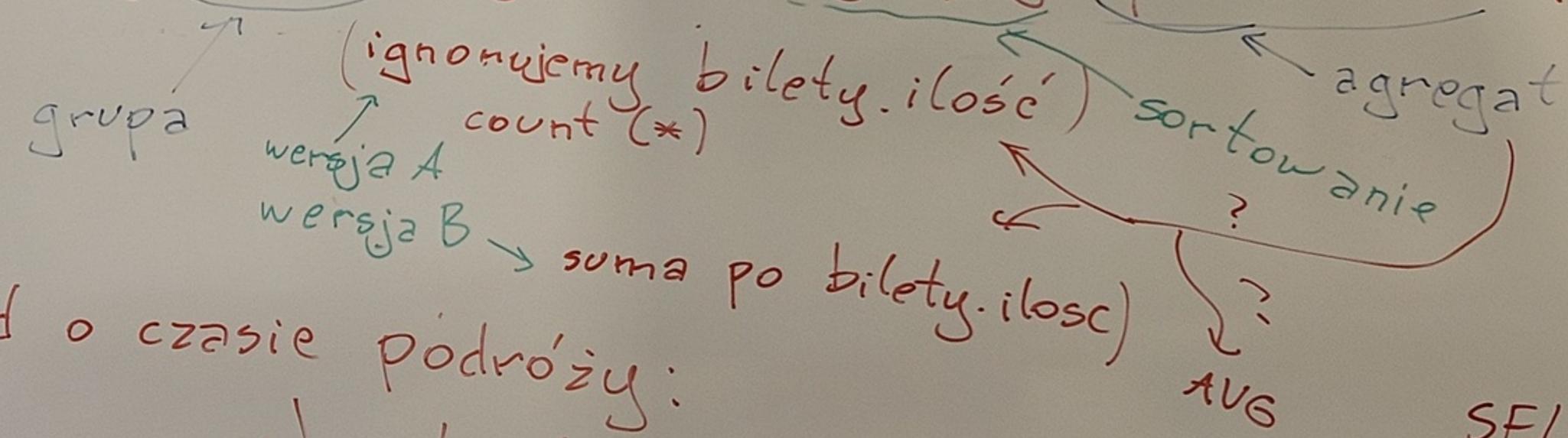
Zad. 2. Ile modeli lokomotyw znajduje się w składach?



Zad. 3. Które składy pociągów nie były używane w żadnym przejeździe (nie wyjechały)?

Zad. 4. Numery przejazdów, na które nikt nie kupił biletu?

Zad. 5. Z której stacji wyjechało najwięcej pasażerów?



Zad. 6. Przejazd o czasie podróży:  
a) najkrótszym  
b) najdłuższym

```
TIMEDIFF(  
SELECT "03:00:00",  
- "01:01:00")
```

Zad. 7. Który model lokomotywy jest najczęściej wykorzystywany?

# DATA i CZAS

'14/03/2026'

substr('14/03/2026', 7, 4) ||

'-' ||

substr('14/03/2026', 4, 2) ||



2026-03-14

CAST

SELECT (strftime('%Y', '2026-03-14') AS INTEGER)

SELECT DATE('now'), TIME('now')

SELECT '2026-03-14' < '2026-04-02'

SELECT julianday( ) - julianday( )  
j - dzień roku  
Y - dni juliańskich

SELECT CURRENT\_DATE,  
- '' - TIME,  
'' - TIMESTAMP

SELECT DATE('2026-03-14'),

TIME('12:45:00'),

DATETIME('2026-03-14 12:45:00'),

SELECT DATE('2026-03-14', '-4 days', '+1 months')

M

5  
4  
3  
2  
1

- Y - rok
- m - miesiąc
- d - dzień miesiąca
- H - godzina
- M - minuta
- S - sekunda
- s - sekunda "linuksowa"
- w - dzień tygodnia
- W - tydzień roku

liczba aktorów

COUNT (\*)  
COUNT(actor\_id)

SELECT COUNT(DISTINCT first\_name) AS licznosc

Lista wszystkich nazwisk (bez powtórzeń)   
 z DISTINCT   
 bez DISTINCT   
 ↪ sortowanie

nazwisko | licznosc

sort malejaco

tylko te, które występują  
więcej niż jednokrotnie

ORDER BY COUNT(\*)

w jakich filmach grał dany aktor  
↑ o actor\_id

licznosc

2

LIMIT 5

w ilu filmach grał każdy aktor?

imię | nazw | ile filmów

country city

w domu : państwa  
wyświetl miasta, które mają  
id identyczne id.

państwo	miasto

SELECT country.country AS państwo, city.city AS miasto  
FROM country JOIN city USING (country\_id)

zamiast (ON country.country\_id = city.country\_id)

FROM country JOIN city USING country\_id

vs

FROM country JOIN city ON country.country\_id = city.country\_id