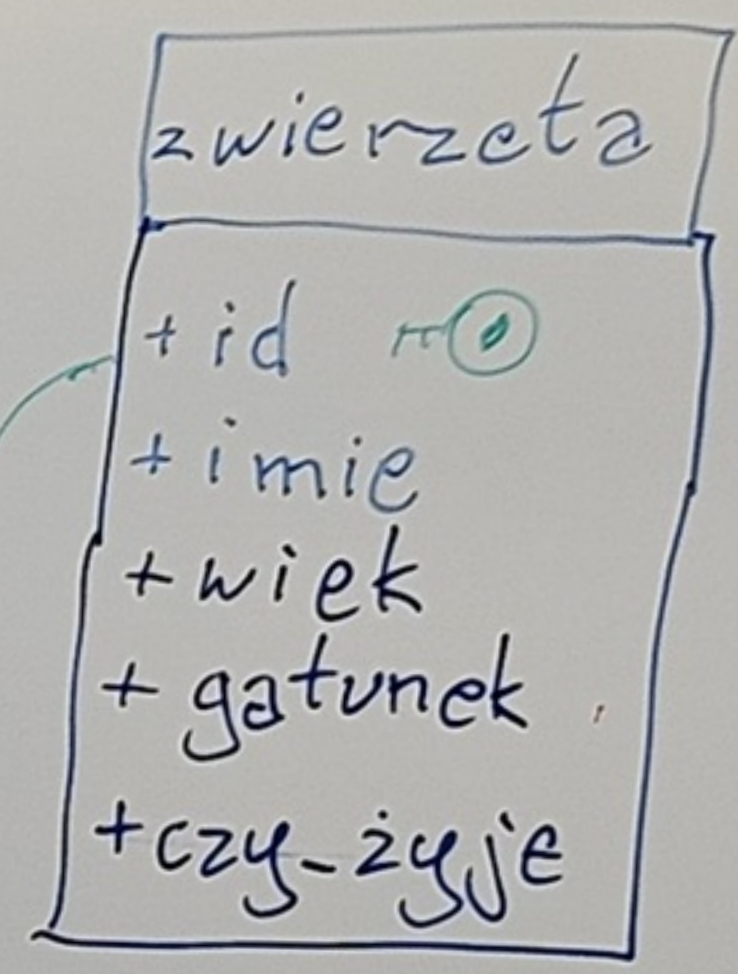
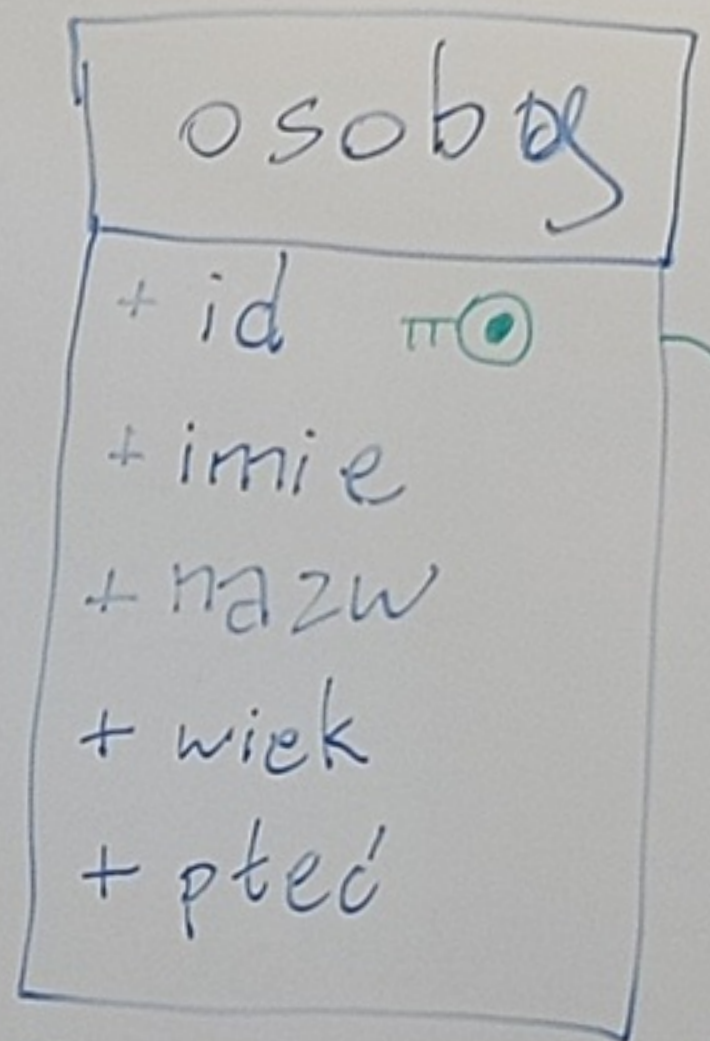
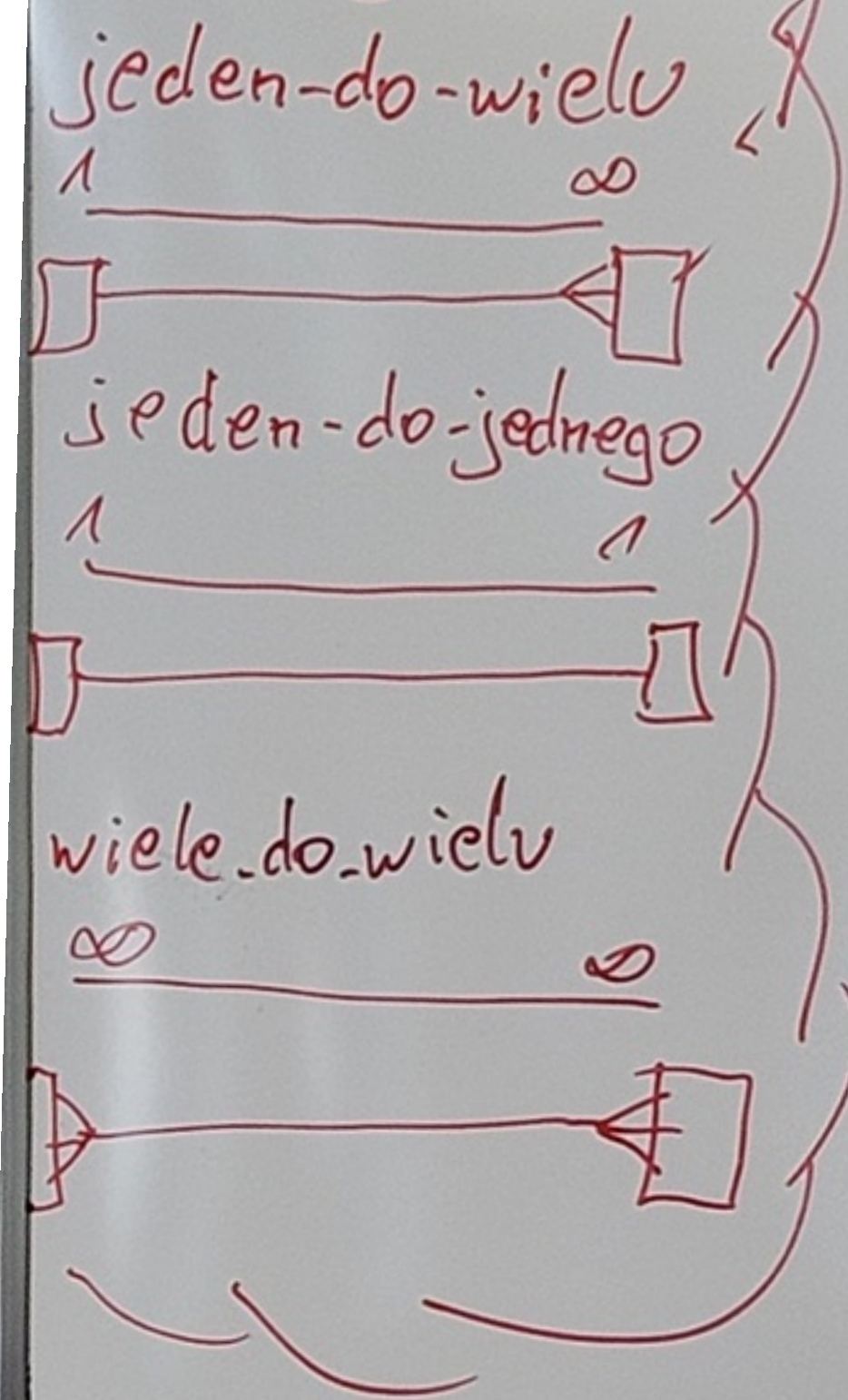


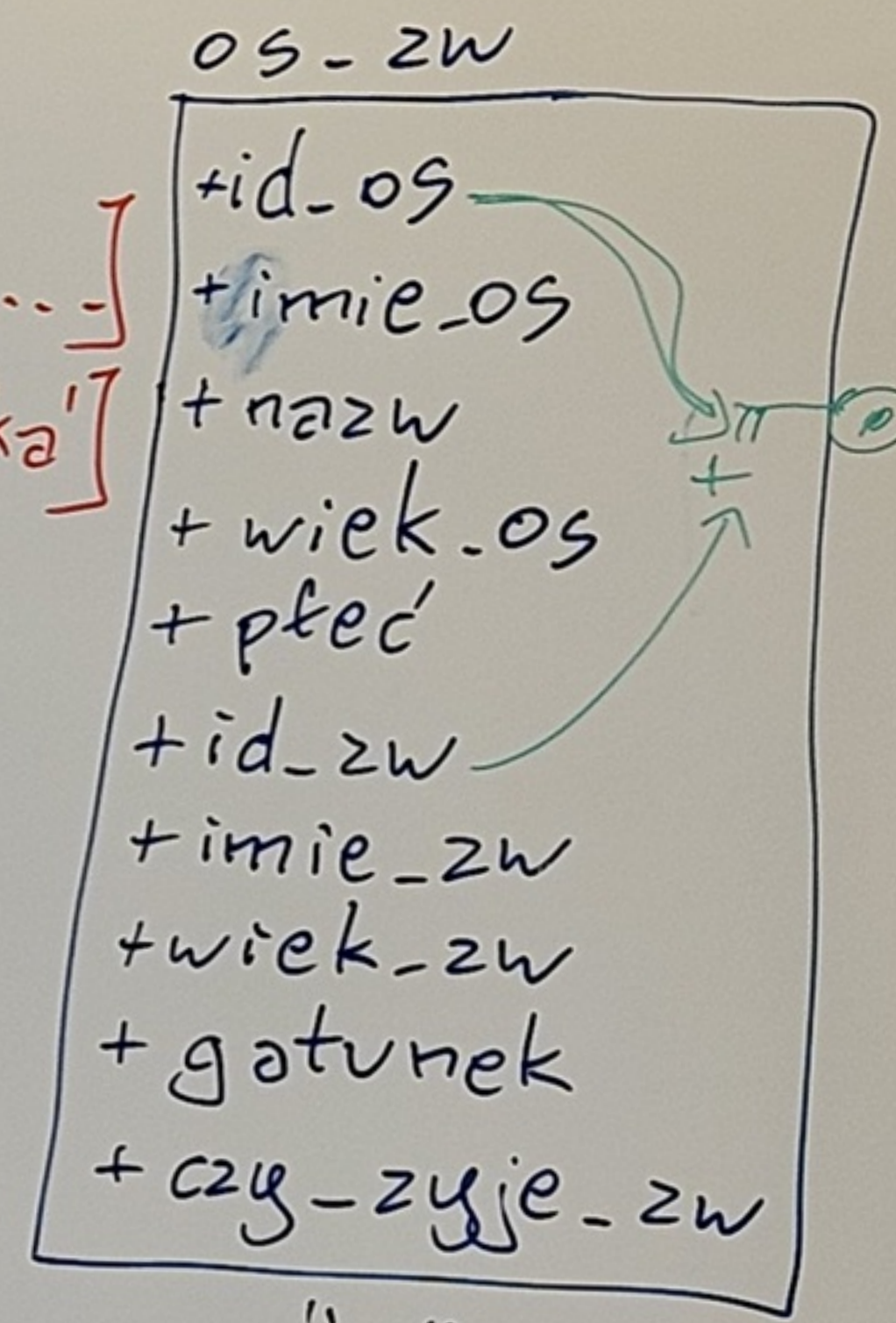
relacje:



```

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

```



unikatowy

Schemat bazy danych

Modelowanie danych

id-os	id-zw
3	8
16	1

wygodniej:

```

dane_osob = gen_osoby(...)
dane_zwierz = gen_zwierz(...)
:
kursor.executemany("INSERT INTO osoby VALUES
                    dane_osob)
                    INTO zwierzeta...
                    dane_zwierzat

```



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: usunąć z bazy martwe zwierzęta

osoby

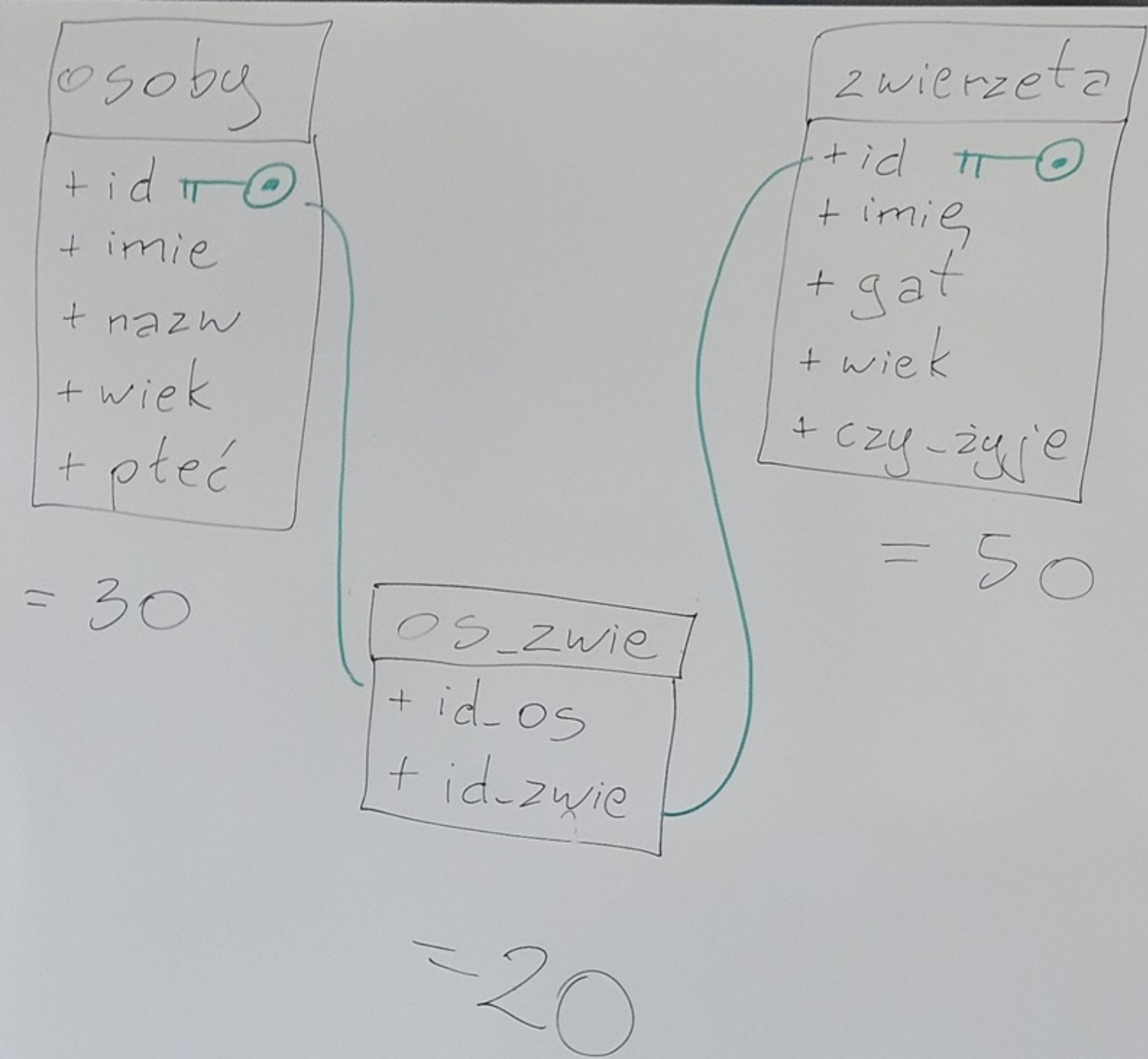
+ id
+ imie
+ nazw
+ wiek
+ plec

osoby_zwie

+ id_os
+ id_zwie

zwierzeta

+ id
+ imie
+ gat
+ wiek
+ czy-zyje



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

FROM osoby INNER JOIN os_zwie

ON osoby.id = os_zwie.id_os
LEFT
INNER JOIN zwierzeta

ON os_zwie.id_zwie = zwierzeta.id

WHERE osoby.imie = "Piotr"

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"

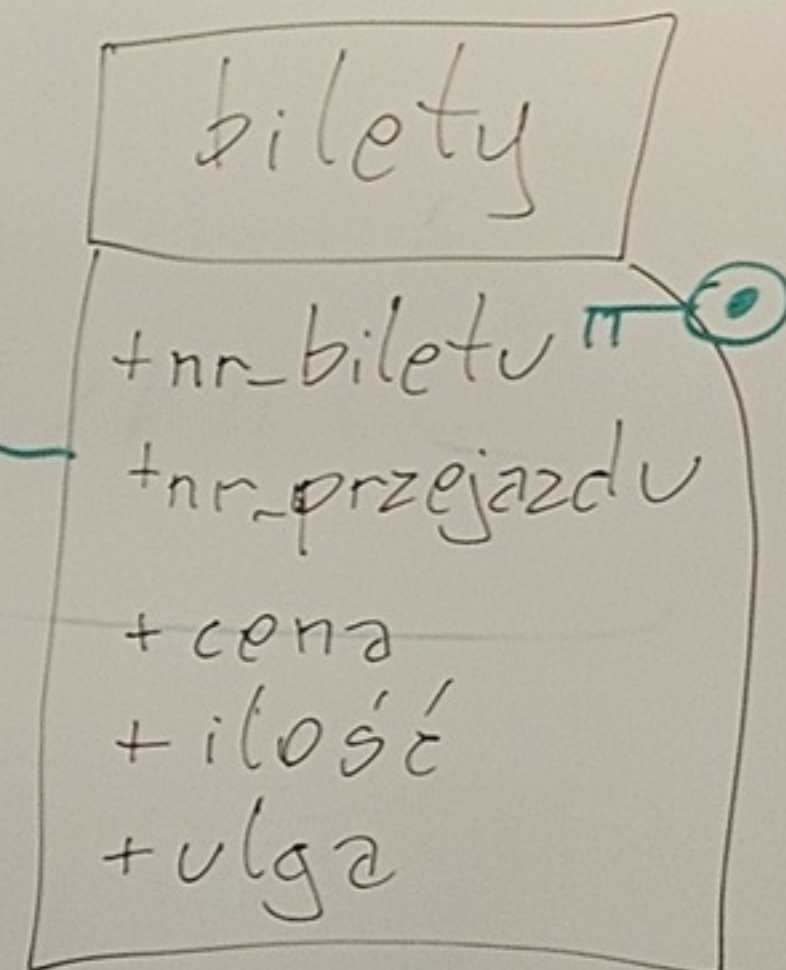
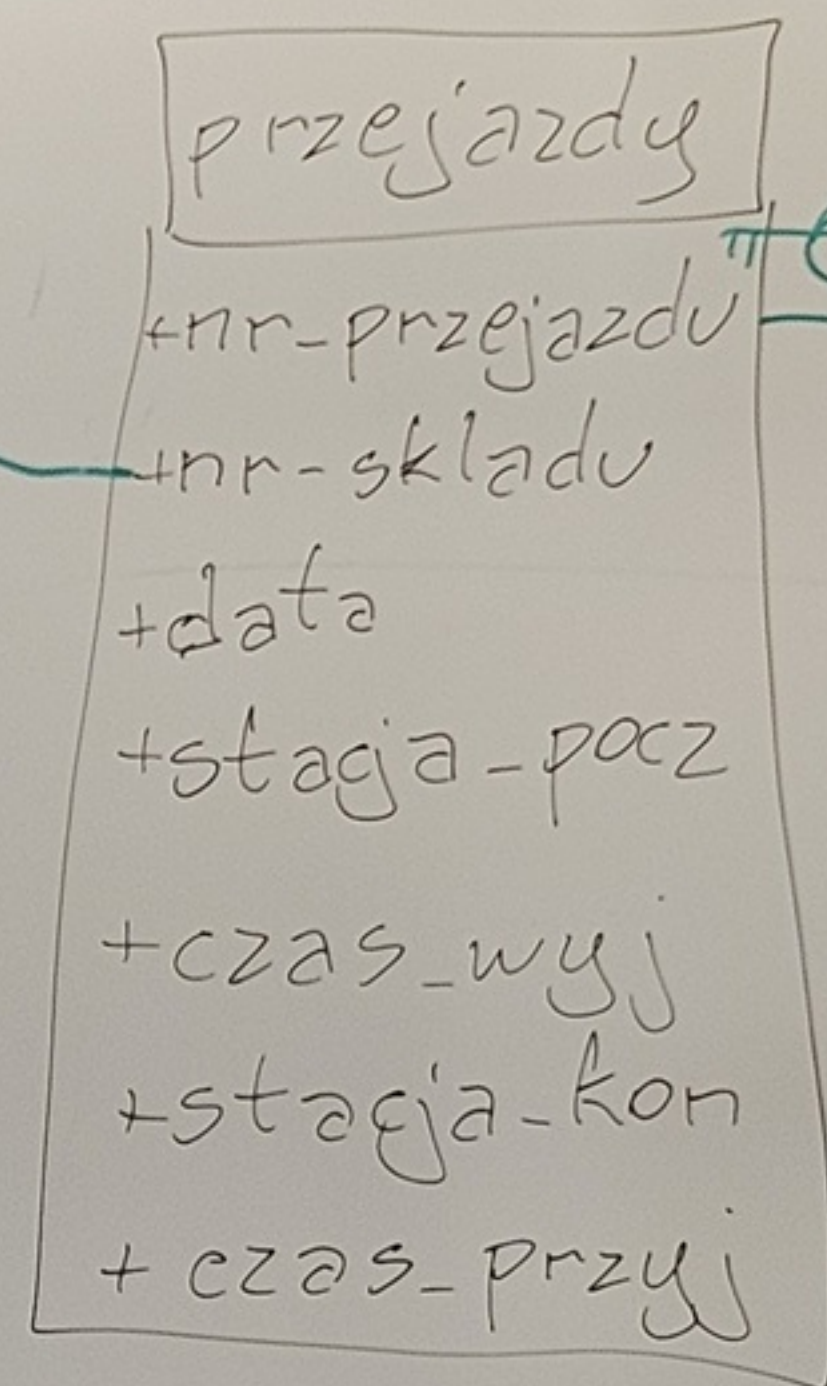
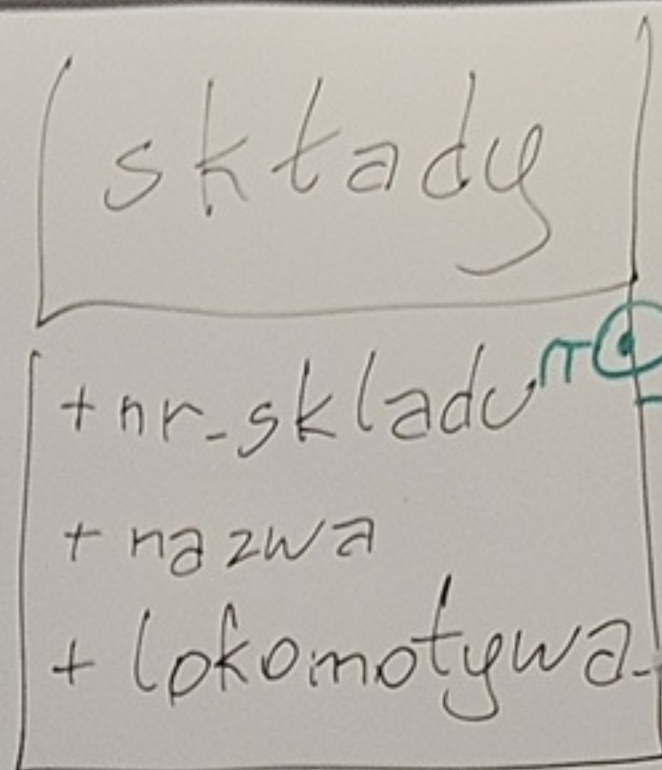
FROM tab1 JOIN tab2
ON id

→ [tab1]
+ id [tab2]
+ id

W domu:

jakie są preferencje ptci
w zakresie gatunków zw.
- 1 za pytaniem

model
lokom



101; Bachus; EP-07

1.001; 501; 180; 1; 0,34

501; 101; 15.07.2022; Wrocław Gł.; 08:12:00; Prąmyśl; 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?

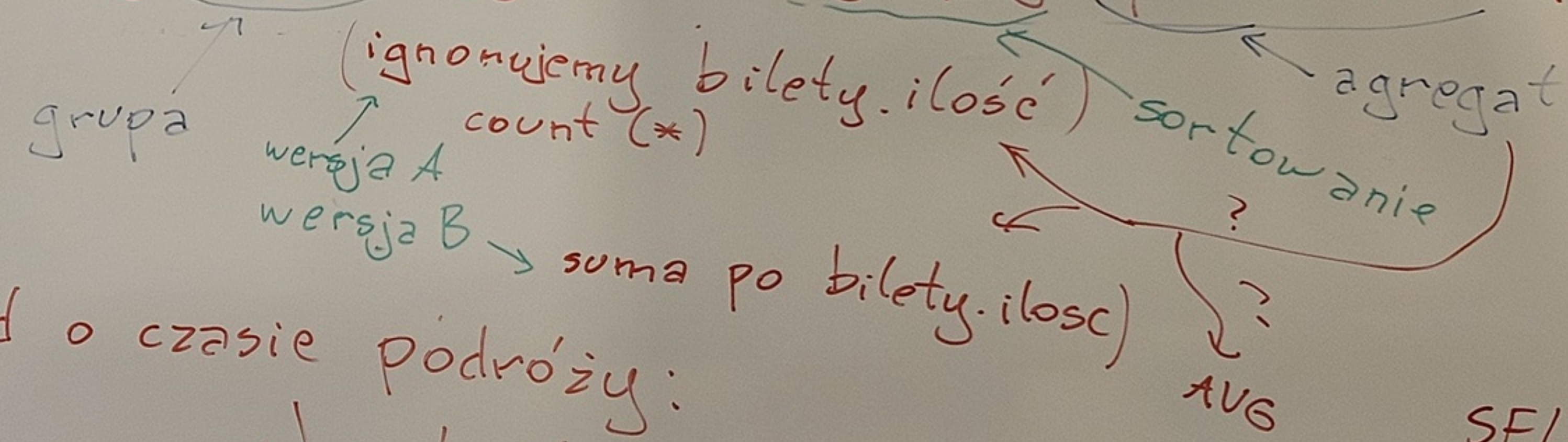
lokom
EP-07
EP-09
Husarz
Pendolino

ilość modeli
4

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?



^{na później}
Zad. 6. Przejazd o czasie podróży:
a) najkrótszym
b) najdłuższym

Zad. 7. Który model lokomotywy jest najczęściej wykorzystywany?

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

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()

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

ms
s

j - dzień roku
Y - dni juliańskich

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

<u>nazwisko</u>	<u>licznosc</u>
-----------------	-----------------

sort malejaco

tylko te, które występują
więcej niż jednokrotnie

ORDER BY COUNT(*)

w jakich filmach grał dany aktor
↑
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

SELECT instr('Polowanie na kurdle', 'nie')

substr(— 11 — ', 2, 3)

100 100
mm

|||||

'Polowanie na kurdle' LIKE "P_L%" 0

length('Bzówowie')

rtrim, ltrim, trim('!Sprawdzian z SQL!', '!')

replace('Polowanie na kurdle', 'Kurdle', 'Pokemony')

lower('Bogdan')

upper(— 11 —)

--

coalesce('Józef', 'ktoś')

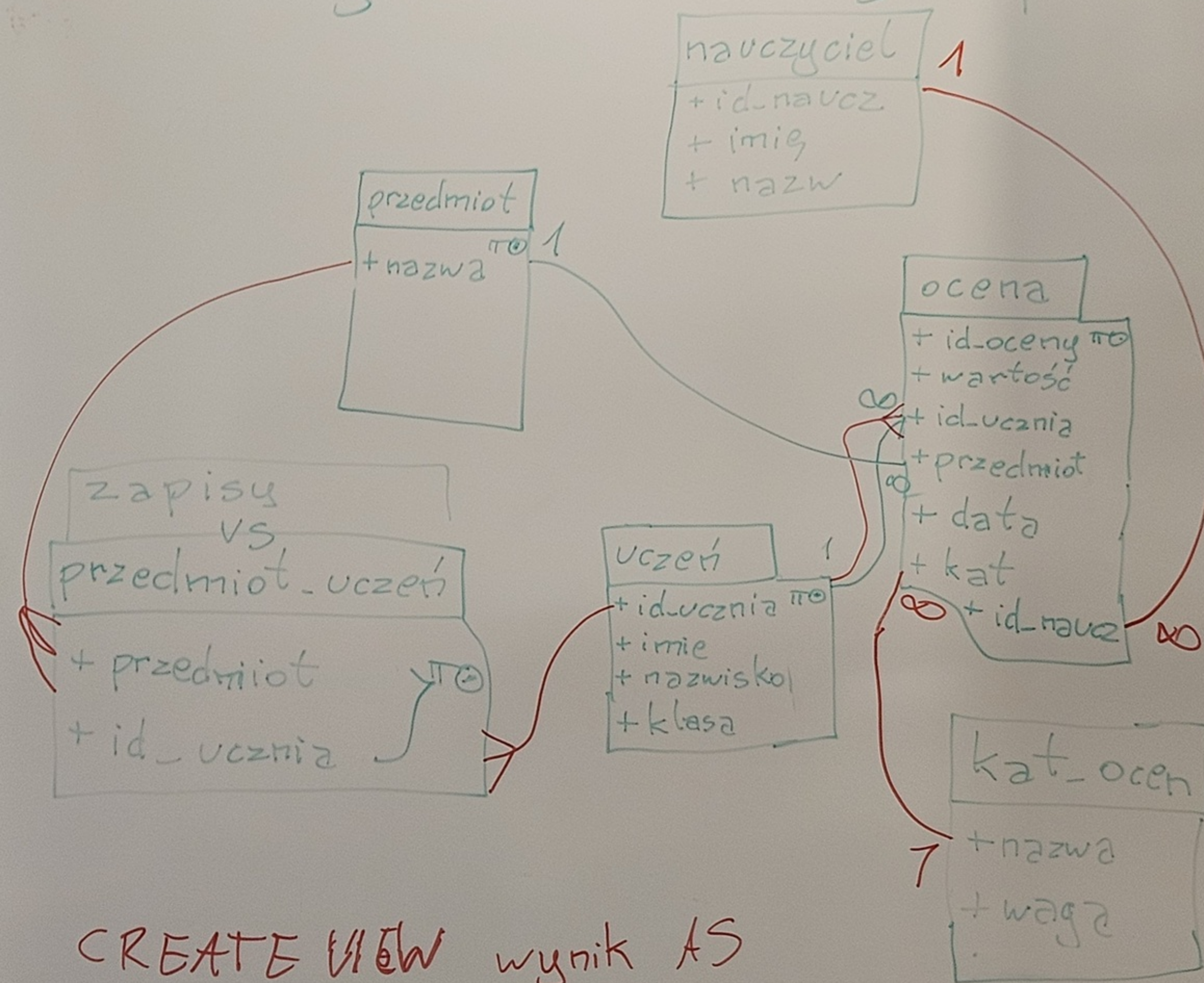
abs(random()) % 100 + 1

Losowa próbka 5-ci
rekordów tabeli.

ORDER BY



Oceny ^{częstkowe} uczniów z różnych przedmiotów



CREATE VIEW wynik AS

```

SELECT first_name, COUNT(*) AS ile
FROM actor
GROUP BY first_name
ORDER BY ile
    
```

```

SELECT * FROM wynik
WHERE ile > 3
    
```

```

SELECT *
FROM ...
WHERE id IN
    
```

```

(SELECT *
    
```

WITH wynik AS

```

(SELECT first_name, last_name
FROM actor)
    
```

```

SELECT wynik.last_name
FROM wynik JOIN film_actor
    
```