



1. Podaj autorów spoza Polski. Wyniki posortuj według nazwisk.

```

select nazwisko,kraj from Autorzy
where kraj!='Polska'
order by nazwisko
  
```

2. Podaj książki zawierające ciąg XML w tytule. Wyniki posortuj po tytule.

```

select tytul from Ksiazki
where tytul like '%XML%'
order by tytul
  
```

3. Znajdź książki droższe od *Fuzzy Logic*. Wyniki posortuj po tytule.

Użyj złączenia tabel.

```

select b.tytul from Ksiazki a
cross join Ksiazki b
where a.tytul='Fuzzy Logic' and b.cena>a.cena
order by b.tytul
  
```

4. Znajdź książki droższe od *Fuzzy Logic*. Wyniki posortuj po tytule.

Użyj podzapytania.

```

select tytul from Ksiazki
where cena>(select cena from Ksiazki where tytul='Fuzzy Logic')
order by tytul
  
```

5. Podaj autorów książek z informatyki. Wyniki posortuj po nazwisku.

Użyj złączenia tabel.

```
select distinct(nazwisko) from Autorzy  
join Ksiazki on id_autor=autor  
where dzial='informatyka'  
order by nazwisko
```

6. Podaj autorów książek z informatyki. Wyniki posortuj po nazwisku.

Użyj podzapytania.

```
select nazwisko from Autorzy  
where id_autor in (select autor from Ksiazki where dzial='informatyka')  
order by nazwisko
```

7. Podaj liczbę książek w każdym z działów. Wyniki posortuj po nazwie działu.

Użyj funkcji agregującej.

```
select dzial,count(*) as "ile" from Ksiazki  
group by dzial  
order by dzial
```

8. Podaj średnią cenę książek Sapkowskiego.

Użyj funkcji agregującej.

```
select avg(cena) as "srednia" from Ksiazki  
group by autor having autor=(select id_autor from Autorzy where  
nazwisko='Sapkowski')
```

9. Podaj tytuł najtańszej książki z informatyki.

Użyj podzapytania.

```
select tytuł from Ksiazki  
where cena=(select min(cena) from Ksiazki where dzial='informatyka')
```

10. Podaj tytuł najtańszej książki dla każdego z działów. Wyniki posortuj po nazwie działu.

Użyj podzapytania.

select dzial, tytul from Ksiazki

where cena in (select min(cena) from Ksiazki group by dzial)

order by dzial

11. Podaj autorów, którzy napisali przynajmniej dwie książki po roku 1996. Wyniki posortuj po nazwisku.

Użyj grupowania.

select a.nazwisko,k.ile from Autorzy a

**join (select autor,count(tytul) as "ile" from Ksiazki where rok_wydania>1996
group by autor having count(tytul)>1) as k on a.id_autor=k.autor**

order by a.nazwisko

12. Znajdź działy, do których pisze więcej niż jeden autor. Wyniki posortuj po nazwie.

Użyj grupowania.

select dzial,count(distinct(autor)) as "licz_autorow" from Ksiazki

group by dzial having count(distinct(autor))>1

order by dzial

13. Znajdź dział o największej liczbie książek. Wyniki posortuj po nazwie działu.

Nie używaj złączenia tabel.

select dzial, count(tytul) as "ile książek" from Ksiazki

**group by dzial having count(tytul)=(select max(k.ile) from (select count(tytul) as
"ile" from Ksiazki group by dzial) as k)**

order by dzial

14. Podaj autora, którego książek nie ma w bazie. Wyniki posortuj po nazwisku.

Użyj złączenia tabel oraz IS NULL.

Nie używaj podzapytania.

select a.nazwisko from Autorzy a left join Ksiazki k on a.id_autor=k.autor

where k.autor is null

15. Podaj autora, którego książek nie ma w bazie. Wyniki posortuj po nazwisku.

Użyj podzapytania.

Nie używaj złączenia tabel i IS NULL.

```
select nazwisko from Autorzy  
  
where id_autor not in (select autor from Ksiazki)  
  
order by nazwisko
```

16. Podaj autora, którego książek nie ma w bazie. Wyniki posortuj po nazwisku.

Użyj EXISTS.

Nie używaj złączenia tabel i IS NULL.

```
select a.nazwisko from Autorzy a  
  
where not EXISTS (select 1 from Ksiazki k where k.autor=a.id_autor)
```

17. Podaj tytuły książek droższych niż wynosi średnia cena książki w ich dziale. Wyniki posortuj po tytule.

Użyj funkcji agregującej i podzapytania.

```
select a.tytul from Ksiazki a  
  
where a.cena>(select avg(b.cena) from Ksiazki b where a.dzial=b.dzial)  
  
order by a.tytul
```