

Zadania

1. Wyszukać wszystkie pracowników firmy.
2. Wyszukać wszystkie osoby, których imię rozpoczyna się na literę "M"
3. Wyszukać wszystkie osoby, których ostatnia nazwiska to "i", a trzecia od końca "s"
4. Dodać osobę, która nie jest nigdzie zatrudniona. Następnie wyszukać wszystkie osoby, które nigdy nie pracowały.
5. Wyszukać wszystkie osoby z województwa "MAZ". Wyświetlić także pełną nazwę województwa.
6. Wyszukać wszystkie osoby, a następnie wykonać union i union all z tabelą osoby . Zanotować wnioski.
7. Wyszukać wszystkich pracowników, którzy pracują od '1994-01-01' i nie skończyła się jeszcze im umowa. Wyniki uporządkować według pensji, w kolejności rosnącej.
8. Wyświetlić pracowników, którzy zarabiają powyżej średniej.

Uwagi:

1. Proszę o tworzenie aliasów tabel.
2. Proszę zwrócić uwagę na łączenie (left / right / join).
3. Tabele powinny być łączone od najmniejszej (pod względem ilości rekordów). Jest to dobra praktyka wpływająca na szybkość wykonywania zapytań.
4. Określenie "pracownik" odnosi się do relacji osoba - etat.

Robienie zadań

1. Wyszukiwanie wszystkich pracowników firmy

```
SELECT imię_i_nazwisko FROM osoby AS o  
JOIN etaty AS e ON o.id_osoby = e.id_osoby;
```

2. Wyszukiwanie wszystkich osób, których imię rozpoczyna się na literę "M".

```
SELECT * FROM dbo.osoby  
WHERE imię LIKE 'M%';
```

3. Wyszukiwanie wszystkich osób, których ostatnia nazwiska to "i", a trzecia od końca "s".

```
SELECT * FROM osoby AS o WHERE o.nazwisko LIKE '%s_i';
```

4. Dodawanie osoby, która nie jest nigdzie zatrudniona. Wyszukiwanie wszystkich osób, które nigdy nie pracowały.

Dodawanie osoby

```
INSERT INTO osoby (id_miasta, imię, nazwisko) VALUES ('1', 'Bartosz', 'Wojcik');
```

```
SELECT imię_i_nazwisko FROM osoby AS o
LEFT JOIN etaty AS e ON e.id_osoby NOT IN (select id_osoby from etaty);
```

5. Wyszukiwanie wszystkich osób z województwa "MAZ". Wyświetlanie także pełną nazwę województwa.

```
SELECT imię_i_nazwisko, nazwa FROM osoby AS o
JOIN woj AS w ON w.kod_woj = 'MAZ';
```

6. Wyszukiwanie wszystkich osób, wykonywanie union i union all z tabelą osoby .

```
SELECT * FROM osoby
UNION
SELECT * FROM osoby
```

Tutaj 7 recordów.

```
SELECT * FROM osoby
UNION ALL
SELECT * FROM osoby
```

Wniosek: różnią się ilość recordów, bo dla UNION łączą różne(tylko unikalne), a dla UNION ALL łączą wszystkie(nie unikalne).

7. Wyszukiwanie wszystkich pracowników, którzy pracują od '1994-01-01' i nie skończyła się jeszcze im umowa. Wyniki uporządkowane według pensji, w kolejności rosnącej.

```
SELECT imię_i_nazwisko FROM osoby AS o
JOIN etaty AS e ON o.id_osoby = e.id_osoby
WHERE e.od > '1994.01.01' AND e.do IS NULL
ORDER BY e.pensja;
```

8. Wyświetlić pracowników, którzy zarabiają powyżej średniej.

```
SELECT imię_i_nazwisko FROM osoby AS o
INNER JOIN etaty AS e ON o.id_osoby = e.id_osoby
WHERE e.pensja > (SELECT AVG(pensja) FROM etaty);
```

CALY KOD:

```
SELECT imię_i_nazwisko FROM osoby as o  
JOIN etaty as e ON o.id_osoby = e.id_osoby;
```

```
SELECT * FROM dbo.osoby  
WHERE imię LIKE 'M%';
```

```
SELECT * FROM osoby as o WHERE o.nazwisko LIKE '%s_i';
```

```
INSERT INTO osoby (id_miasta, imię, nazwisko) VALUES ('1', 'Ivanek', 'Prokop');
```

```
SELECT imię_i_nazwisko FROM osoby as o  
LEFT JOIN etaty as e ON e.id_osoby not in (select id_osoby from etaty);
```

```
SELECT * FROM firmy;
```

```
SELECT imię_i_nazwisko, nazwa FROM osoby as o  
JOIN woj as w on w.kod_woj = 'MAZ';
```

```
SELECT * FROM osoby  
UNION  
SELECT * FROM osoby
```

```
SELECT * FROM osoby  
UNION ALL  
SELECT * FROM osoby
```

```
SELECT imię_i_nazwisko FROM osoby as o  
JOIN etaty as e ON o.id_osoby = e.id_osoby  
WHERE e.od > '1994.01.01' AND e.do IS NULL  
ORDER BY e.pensja;
```

```
SELECT imię_i_nazwisko FROM osoby AS o  
INNER JOIN etaty AS e ON o.id_osoby = e.id_osoby  
WHERE e.pensja > (SELECT AVG(pensja) FROM etaty);
```