**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;

1. Wyszukiwanie wszystkich osób, których imię rozpoczyna się na literę “M”.

SELECT \* FROM dbo.osoby

WHERE imię LIKE 'M%';

1. Wyszukiwanie wszystkich osób, których ostatnia nazwiska to “i”, a trzecia od końca “s”.

SELECT \* FROM osoby AS o WHERE o.nazwisko LIKE '%s\_i';

1. 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);

1. 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';

1. 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 łączę wszystkie( nie unikalne).*

1. 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;

1. 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);