

## Zadania

1. Pracownicy z największej firmy (obliczone na podstawie funkcji) mają zostać przeniesieni do innego miasta (wskazanego przez studenta).
2. Następnie mają oni (i tylko oni) dostać podwyżkę o 1000 zł.
3. Następnie przeprowadzana jest redukcja pracowników. Mają zostać zwolnieni Ci, którzy zarabiają 10% powyżej średniej (i więcej).
4. Wprowadzono w firmie parytet, więc ma być zatrudniona taka sama liczba kobiet jak i mężczyzn. Obliczyć osobno jedną i drugą ilość, a następnie wykonać stosowną ilość insertów.

Zadania odnoszące się do wszystkich firm:

5. Policzyc średnie, minimalne, maksymalne zarobki dla każdego stanowiska w zależności od miasta. Proszę użyć group by.

## Robienie zadań

1. Przenoszenie pracowników z największej firmy do innego miasta.

```
SELECT TOP 1 id_firmy
      FROM osoby AS o
JOIN etaty AS e
      ON o.id_osoby = e.id_osoby
JOIN firmy AS f
      ON f.nazwa_skr = e.id_firmy
GROUP BY e.id_firmy ORDER BY COUNT(e.id_firmy) DESC
UPDATE firmy SET id_miasta = 2
WHERE nazwa_skr = (SELECT TOP 1 id_firmy FROM etaty
GROUP BY id_firmy ORDER BY COUNT(id_firmy) DESC)
```

2. Przenoszone pracownicy z największej firmy dostaną podwyżkę o 1000 zł.

```
UPDATE etaty SET pensja = pensja + 1000
WHERE id_firmy = (SELECT TOP 1 id_firmy FROM etaty
GROUP BY id_firmy order by COUNT(id_firmy) DESC)
```

3. Redukcja pracowników. Zostaną zwolnieni Ci, którzy zarabiają 10% powyżej średniej (i więcej).

```
DELETE FROM etaty WHERE pensja >= (SELECT AVG(pensja) FROM etaty)*1.1
```

4. Obliczanie wszystkich mężczyzn

```
SELECT count(*) FROM osoby AS o
WHERE o.nazwisko LIKE '%i'
      OR o.nazwisko LIKE '%y'
      OR o.nazwisko LIKE '%n'
```

```
OR o.nazwisko LIKE '%p';
```

(No column name)

7

### Obliczanie wszystkich kobiet

```
SELECT COUNT(*) FROM osoby AS o
WHERE o.nazwisko LIKE '%a' AND o.imię LIKE '%a';
```

(No column name)

0

Zrobienie 7 insertów ( ma być ostatnia litera nazwiska 'a' to będzie oznaczać, że jest kobieta )

```
INSERT INTO osoby
VALUES (1, 'Tania', 'Stowoska');
INSERT INTO osoby
VALUES (2, 'Vsnia', 'Krowoska');
INSERT INTO osoby
VALUES (1, 'Annia', 'Worowoska');
INSERT INTO osoby
VALUES (3, 'Karinaa', 'Gotwoska');
INSERT INTO osoby
VALUES (1, 'Zuzia', 'Kujtawoska');
INSERT INTO osoby
VALUES (2, 'Inna', 'Bioesska');
INSERT INTO osoby
VALUES (1, 'Zina', 'Jowskska');
```

5. Liczenie średnie, minimalne, maksymalne zarobki dla każdego stanowiska w zależności od miasta, używając GROUP BY.

```
SELECT m.nazwa, e.stanowisko,
AVG(e.pensja) AS srednia,
MIN(e.pensja) AS minimum,
MAX(e.pensja) AS maksimum
FROM dbo.miasta AS m
INNER JOIN etaty AS e
INNER JOIN osoby AS o
ON e.id_osoby = o.id_osoby
ON m.id_miasta = o.id_miasta
GROUP BY m.nazwa, e.stanowisko;
```

6. Tworzenie nowej firmy i zatrudnienie w niej osoby o sumie długości imienia i nazwiska większej niż średnia.

```
INSERT INTO firmy
VALUES ( 'AU', 1, 'Autl Uw', 22-041, 'Wejska');
```

```
SELECT o.imię, o.nazwisko,
SUM(imię) AS suma SUM(nazwisko) AS sumaa > AVG(imię) AS srednia AND AVG(nazwisko) AS
sredniaa
FROM dbo.osoby AS o
INSERT INTO osoby AS o
VALUES (1, 'Krutoj', 'Perec');
```

## CALY KOD:

```
SELECT TOP 1 id_firmy
FROM osoby AS o
JOIN etaty AS e
    ON o.id_osoby = e.id_osoby
JOIN firmy AS f
    ON f.nazwa_skr = e.id_firmy
GROUP BY e.id_firmy ORDER BY COUNT(e.id_firmy) DESC
UPDATE firmy SET id_miasta = 2
WHERE nazwa_skr = (SELECT TOP 1 id_firmy FROM etaty
GROUP BY id_firmy ORDER BY COUNT(id_firmy) DESC)

UPDATE etaty SET pensja = pensja + 1000
WHERE id_firmy = (SELECT TOP 1 id_firmy FROM etaty
GROUP BY id_firmy order by COUNT(id_firmy) DESC)

DELETE FROM etaty WHERE pensja > (SELECT AVG(pensja) FROM etaty)*1.1;

SELECT count(*) FROM osoby AS o
WHERE o.imię LIKE '%a' AND o.imię LIKE '%a';

SELECT count(*) FROM osoby AS o
WHERE o.nazwisko LIKE '%i'
    OR o.nazwisko LIKE '%y'
    OR o.nazwisko LIKE '%n'
    OR o.nazwisko LIKE '%p';

INSERT INTO osoby
VALUES (1, 'Tania', 'Stowoska');
INSERT INTO osoby
VALUES (2, 'Vsnia', 'Krowoska');
INSERT INTO osoby
VALUES (1, 'Annia', 'Worowoska');
INSERT INTO osoby
VALUES (3, 'Karinaa', 'Gotwoska');
INSERT INTO osoby
VALUES (1, 'Zuzia', 'Kujtawoska');
INSERT INTO osoby
VALUES (2, 'Inna', 'Bioesska');
INSERT INTO osoby
VALUES (1, 'Zina', 'Jowskska');

SELECT m.nazwa

SELECT m.nazwa, e.stanowisko,
AVG(e.pensja) AS srednia,
MIN(e.pensja) AS minimum,
MAX(e.pensja) AS maksimum
FROM dbo.miasta AS m
    INNER JOIN etaty AS e
        INNER JOIN osoby AS o
            ON e.id_osoby = o.id_osoby
        ON m.id_miasta = o.id_miasta
GROUP BY m.nazwa, e.stanowisko;
```