



# Podstawy baz danych

## Baza danych Access - zadania

Opracował: Andrzej Nowak

### Zadanie\_1\_Noworodki

Pliki **noworodki.txt** oraz **mamy.txt** zawierają dane o dzieciach i ich matkach.

W pliku **noworodki.txt** każdy wiersz zawiera następujące informacje o jednym dziecku, rozdzielone znakami odstępu:

**identyfikator, płeć (c - córka, s - syn), imię, data urodzenia, waga [g], wzrost [cm] oraz identyfikator matki.**

#### Przykład:

```
1 c Agnieszka 20-lis-1999 2450 48 33
```

W pliku **mamy.txt** każdy wiersz zawiera informacje o jednej kobiecie, rozdzielone znakami odstępu:

**identyfikator matki, imię, wiek.**

#### Przykład:

```
1 Agata 25
```

Identyfikator matki z pliku **noworodki.txt** odpowiada identyfikatorowi w pliku **mamy.txt**.

Wykorzystując dane zawarte w plikach **mamy.txt** i **noworodki.txt** oraz dostępne narzędzia informatyczne, wykonaj poniższe polecenia.

- Podaj imię i wzrost najwyższego chłopca oraz imię i wzrost najwyższej dziewczynki.  
Uwaga: Jest tylko jeden taki chłopiec i tylko jedna taka dziewczynka.
- W którym dniu urodziło się najwięcej dzieci? Podaj datę i liczbę dzieci.  
Uwaga: Jest tylko jeden taki dzień.
- Podaj imiona kobiet w wieku poniżej 25 lat, które urodziły dzieci o wadze powyżej 4000 g.
- Podaj imiona i daty urodzenia dziewczynek, które odziedziczyły imię po matce.
- W pliku **noworodki.txt** zapisane są informacje o narodzinach bliźniąt. Bliźnięta można rozpoznać po tej samej dacie urodzenia i tym samym identyfikatorze matki. Pamiętaj, że przykładowo Jacek i Agatka oraz Agatka i Jacek to ta sama para. Możesz założyć, że w danych nie ma żadnych trojaczków, czworaczków, itd. Podaj daty, w których urodziły się bliźnięta.

Odpowiedzi zapisz w pliku tekstowym o nazwie **odpowiedz.txt**



# Rozwiązanie

Z opisu wynika, że w bazie danych będą dwie tabele o następującej strukturze kolumn:

noworodki						
id_dziecka	plec	imie_dziecka	data_urodzenia	waga	wzrost	id_mamy
INT	CHAR(1)	VARCHAR(20)	DATE	INT	INT	INT

matki		
id_matki	imie_matki	wiek
INT	VARCHAR(20)	INT

Tabele są połączone relacją: `noworodki.id_mamy` → `matki.id_matki`

Logujemy się na serwer mysql i tworzymy odpowiednie tabele:

```
CREATE TABLE noworodki
(
  id_dziecka INT PRIMARY KEY,
  plec CHAR(1) NOT NULL,
  imie_dziecka VARCHAR(20) NOT NULL,
  data_urodzenia DATE NOT NULL,
  waga INT NOT NULL,
  wzrost INT NOT NULL,
  id_mamy INT NOT NULL
);
```

```
CREATE TABLE matki
(
  id_matki INT PRIMARY KEY,
  imie_matki VARCHAR(20) NOT NULL,
  wiek INT NOT NULL
);
```

Pobieramy pliki **noworodki.txt** do katalogu **/tmp**.

Plik wczytujemy do tabeli w bazie danych

```
LOAD DATA INFILE '/tmp/noworodki.txt' INTO TABLE noworodki FIELDS TERMINATED BY ' '
```

Sprawdź, czy dane zostały poprawnie odczytane:

```
SELECT * FROM noworodki;
```

Rekordów ma być 180.

Pobieramy do katalogu /tmp plik **mamy.txt** i wczytujemy go do tabeli:

```
LOAD DATA INFILE '/tmp/mamy.txt' INTO TABLE matki FIELDS TERMINATED BY ',';
SELECT * FROM matki;
```

Tabela powinna zawierać 174 wiersze.

### a) imię i wzrost najwyższego chłopca oraz imię i wzrost najwyższej dziewczynki.

Rozwiązanie uzyskamy za pomocą dwóch zapytań, które wyświetlą z tabeli noworodki zawartości dwóch kolumn: **imie\_dziecka** oraz **wzrost**. W pierwszym zapytaniu zażądamy, aby pola **plec** zawierały **s** (syn), a w drugim **c** (córka). Wyniki będą posortowane malejąco względem kolumny **wzrost**. Wystarczy zatem odczytać pierwszy wiersz tabeli wynikowej:

```
SELECT imie_dziecka AS Imię, wzrost AS Wzrost
FROM noworodki
WHERE plec='s'
ORDER BY wzrost DESC;
```

```
SELECT imie_dziecka AS Imię, wzrost AS Wzrost
FROM noworodki
WHERE plec='c'
ORDER BY wzrost DESC;
```

Gdyby chodziło o wyświetlenie tylko tych dzieci, bez pozostałej reszty, to musimy wykorzystać grupowanie i funkcję agregującą **MAX**. Grupowanie powoduje wykonanie funkcji agregującej na wszystkich rekordach, które w danej kolumnie posiadają taką samą wartość. Zatem jeśli zgrupujemy tablicę noworodki wg kolumny **plec**, to powstaną dwie grupy: chłopców o polu **plec** równym **s** oraz dziewcząt o polu **plec** równym **c**.

Wydajemy polecenie:

```
SELECT plec, MAX(wzrost)
FROM noworodki
GROUP BY plec;
```

Znając maksymalny wzrost w grupie dziewcząt (**c, 61**) i chłopców (**s,62**), możemy wydać polecenia:

```
SELECT imie_dziecka AS Imię, wzrost AS Wzrost
FROM noworodki
WHERE plec='s' AND wzrost=62;
```

oraz:

```
SELECT imie_dziecka AS Imię, wzrost AS Wzrost
FROM noworodki
WHERE plec='c' AND wzrost=61;
```

W zapytaniu można również wykorzystywać wynik zwrócony przez inne zapytanie. Identyczny efekt otrzymamy, każąc bazie danych wyliczyć odpowiednie wzrosty maksymalne w grupie chłopców i dziewcząt:

```
SELECT imie_dziecka AS Imię, wzrost AS Wzrost
FROM noworodki
WHERE plec='s' AND wzrost=(SELECT MAX(wzrost) FROM noworodki WHERE plec = 's');
```

```
SELECT imie_dziecka AS Imię, wzrost AS Wzrost
FROM noworodki
WHERE plec='c' AND wzrost=(SELECT MAX(wzrost) FROM noworodki WHERE plec = 'c');
```

## b) w którym dniu urodziło się najwięcej dzieci? Podaj datę i liczbę dzieci.

Tutaj musimy zastosować grupowanie wg daty i policzyć liczbę rekordów w każdej z grup. Wykonamy to za pomocą funkcji agregującej **COUNT(\*)**:

```
SELECT data_urodzenia, COUNT(*) AS 'Liczba urodzin'
FROM noworodki
GROUP BY data_urodzenia
ORDER BY COUNT(*);
```

Wynik odczytujemy z ostatniego wiersza.

## c) imiona kobiet w wieku poniżej 25 lat, które urodziły dzieci o wadze powyżej 4000 g

W zapytaniu musimy połączyć ze sobą obie tabele:

```
SELECT imie_matki AS Matka, wiek AS Wiek, waga AS 'Waga dziecka'
FROM matki,noworodki
WHERE id_matki=id_mamy AND wiek < 25 AND waga > 4000
ORDER BY Matka;
```

## d) imiona i daty urodzenia dziewczynek, które odziedziczyły imię po matce

Znów łączymy tabele w zapytaniu:

```
SELECT imie_dziecka AS 'Imię dziecka', data_urodzenia AS 'Data urodzenia'
FROM matki, noworodki
WHERE id_mamy=id_matki AND imie_dziecka=imie_matki
ORDER BY imie_dziecka;
```

## e) daty, w których urodziły się bliźnięta

Warunek: ta sama data urodzenia oraz ta sama matka. Musimy zatem zgrupować wyniki zapytania względem kolumn **data\_urodzenia** i **id\_mamy**. Następnie należy policzyć ilość rekordów w każdej grupie i wyświetlić ze zgrupowania tylko te, które mają wartość **COUNT(\*)** większą od 1:

```
SELECT data_urodzenia AS 'Data narodzin bliźniąt'  
FROM noworodki  
GROUP BY data_urodzenia,id_mamy  
HAVING COUNT(*)>1;
```

Podsumowując, w pliku odpowiedzi umieszczamy:

**a) Oskar 62, Maja 61**

**b) 1999-11-20 23**

**c) Barbara, Maria, Marzena, Eliza, Janina, Paulina**

**d) Anna 199-11-21, Wiktoria 1999-11-20**

**e) 1999-11-19, 1999-11-21, 1999-11-22, 1999-11-23, 1999-12-09, 1999-12-12**