

SYSTEMY BAZ DANYCH

SQL



TEMAT 21-05: Tabele jako zbiory.

Autor dokumentu: Wojciech Galiński

piątek, 18 listopada 2016 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY: <http://dev.mysql.com/doc/>, <http://www.bitbybit.dk/carsten/blog/?p=71>.

Zagadnienia obowiązkowe

1. **Tabele z przykładowymi danymi** – będą nam potrzebne do pokazania różnych rodzajów złączeń. Oto one:

Tabela: OSOBA				Tabela: KLASA		
ID	Nazwisko	Imię	IDKL	ID	Nazwa	Opis
1	Kos	Jan	1	1	1 m „a”	matematyczna
2	Sęk	Anna	4	2	1 h „b”	humanistyczna
4	Bąk	Adam	2	3	2 g „c”	geograficzna

2. **Operacje na zbiorach** – to operacje łączące wyniki wielu zapytań SQL w jedno. Wymagane jest, aby obydwa zapytania SQL zwracały taką samą liczbę kolumn.

Wyróżniamy następujące operacje na zbiorach:

- **suma tabel („UNION”)**;
- **różnica tabel („MINUS”)**;
- **przecięcie (iloczyn) tabel („INTERSECT”)**.

ZAPAMIĘTAJ! Kolejność wykonywania kilku operacji na zbiorach jest istotna.

3. **Suma tabel** – to wszystkie rekordy, które są wynikiem pierwszego lub drugiego zapytania (rekordy zwracane przez obydwa zapytania występują w sumie bez powtórzeń). Do tworzenia sumy tabel używamy operatora „UNION”. Oto wzorzec:

```
SELECT ...  
UNION [ALL]  
SELECT ... ;
```

PRZYKŁAD:

```
SELECT id, nazwa, opis FROM klasa  
UNION  
SELECT 100+id, nazwisko, imie FROM osoba;
```

SELECT ID, Nazwisko, Imię FROM osoba UNION SELECT ID, Nazwa, Opis FROM klasa;		
ID	Nazwisko	Imię
1	Kos	Jan
2	Sęk	Anna
4	Bąk	Adam
1	1 m „a”	matematyczna
2	1 h „b”	humanistyczna
3	2 g „c”	geograficzna

4. **Różnica tabel** – to wszystkie rekordy, które są wynikiem pierwszego, ale NIE drugiego zapytania. W języku SQL do tworzenia sumy tabel używamy operatora „**MINUS**”. Oto wzorzec:

```
SELECT ...  
MINUS  
SELECT ... ;
```

PRZYKŁAD:

```
SELECT id,nazwa,opis FROM klasa MINUS SELECT id,nazwa,opis FROM klasa WHERE id>3;
```

Operator „**MINUS**” NIE JEST dostępny w MySQL – zastępujemy go m. in. w następujący sposób:

```
SELECT DISTINCT tabela1.kolumna1, ...  
FROM tabela1  
WHERE (tabela1.kolumna1, ...) NOT IN  
(SELECT tabela2.kolumna1, ... FROM tabela2);
```

PRZYKŁAD:

```
SELECT DISTINCT imie FROM uczen WHERE imie NOT IN (SELECT DISTINCT imie FROM rodzic);
```

SELECT ID, Nazwa, Opis FROM klasa; MINUS SELECT ID, Nazwa, Opis FROM klasa WHERE Nazwa LIKE '1%';		
ID	Nazwa	Opis
3	2 g „c”	geograficzna

5. **Przecięcie tabel** – służy do tego operator „**INTERSECT**”.

```
SELECT ...  
INTERSECT  
SELECT ... ;
```

Operator „**INTERSECT**” NIE JEST dostępny w MySQL, ale można go zastąpić m. in. w następujący sposób:

```
SELECT DISTINCT tabela1.kolumna1, ...  
FROM tabela1  
WHERE (tabela1.kolumna1, ...) IN  
(SELECT tabela2.kolumna1, ... FROM tabela2);
```

PRZYKŁAD:

```
SELECT imie FROM uczen WHERE id_klasa=1 AND imie IN (SELECT imie FROM uczen WHERE id_klasa=3);
```

SELECT ID, Nazwa, Opis FROM klasa; INTERSECT SELECT ID, Nazwa, Opis FROM klasa WHERE Nazwa LIKE '1%';		
ID	Nazwa	Opis
1	1 m „a”	matematyczna
2	1 h „b”	humanistyczna

Pytania kontrolne

1. Wymień i opisz operatory do operacji na tabelach jako zbiorach.
2. Jakie wymaganie muszą spełniać zapytania, do których stosujemy operacje na zbiorach?
3. Które operatory operacji na zbiorach NIE SĄ dostępne w MySQL i jak ominąć to ograniczenie?

Zadania

Do poniższych zadań wykorzystaj bazę danych „**bde_oceny**”:

1. Wyświetl listę nazwisk uczniów oraz listę nazw klas używając jednego zapytania SQL.
2. Wyświetl imiona uczniów w klasach pierwszych, które nie występują u uczniów w klasach drugich.
3. Wyświetl na jednym wydruku imiona i nazwiska uczniów i rodziców.
4. Wyświetl wspólne imiona uczniów w klasach pierwszych i drugich.
5. Wyświetl nazwiska uczniów, które nie występują u rodziców. Wyświetl nazwiska rodziców, które nie występują u uczniów.
6. Wyświetl imiona, które występują i u uczniów i u rodziców.