Perspektywy w języku SQL

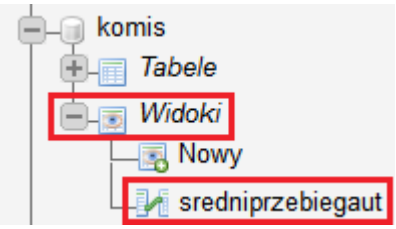
Dotychczas definiowaliśmy tabele za pomocą instrukcji **CREATE TABLE**. Takie tabele fizycznie istnieją w bazie danych i zajmują obszar w fizycznej pamięci systemu. Są konkretnymi plikami na dysku i trwałymi obiektami bazy danych. Istnieją też innego typu obiekty – wirtualne, które fizycznie nie zajmują żadnej przestrzeni na dysku twardym. Powstają jako rezultat zapytań i działań na obiektach trwałych, lecz same nie zawierają danych. Chociaż przyjmują formę wyrażeń, można z nimi pracować jak z trwałymi tabelami i są dostępne od momentu zdefiniowania aż do ich usunięcia.

Mowa o **perspektywach** (ang. perspective), inaczej **widokach wirtualnych** lub krótko **widokach** (ang. views). Perspektywa może zawierać wybrane lub wszystkie wiersze tabeli. Może być utworzona z jednej lub wielu tabel. Wszystko to zależy od zapytania SQL tworzącego widok wirtualny. Z uwagi na swój charakter perspektywy mogą znacząco spowalniać działanie systemu, dlatego przy ich tworzeniu należy kierować się rozwagą. Perspektywy umożliwiają:
- strukturyzowanie i prezentację danych w sposób wygodny dla użytkownika,
- ograniczanie dostępu do danych w taki sposób , by użytkownicy widzieli i operowali wyłącznie na tych danych, których potrzebują,
- podsumowanie danych z różnych tabel do generowania raportów.

Pomimo, że zachowują się w zapytaniach jak tabele, perspektywy nie pozwalają na modyfikację danych. Do ich konstrukcji służy instrukcja **CREATE VIEW**.

Ćwiczenie 1
Definicja perspektywy (baza komis)

```
CREATE VIEW SredniPrzebiegAut AS
SELECT rocznik, AVG(przebieg) sredni_p
FROM Samochody GROUP BY rocznik;
```



Rys.1 PHPMyAdmin – baza danych komis z widokiem: sredniprzebiegaut

W wyniku tej instrukcji zostanie utworzony wirtualny widok prezentujący dane w dwóch kolumnach. Do utworzonej perspektywy można zgłaszać zapytania, posługując się jej nazwą, np.:

```
SELECT rocznik FROM SredniPrzebiegAut WHERE sredni_p < 100000;
```

```
+---------+
| rocznik |
+---------+
|    2010 |
|    2011 |
+---------+
```

Znane polecenie z SHOW TABLES bez dodatkowych parametrów nie rozróżnia tabel od widoków:

```
SHOW TABLES;
+------------------+
| Tables_in_komis  |
+------------------+
| handlowcy        |
| klienci          |
| modele           |
| operacje         |
| pracownicy       |
| producenci       |
| rejestry         |
| rezerwacje       |
| samochody        |
| sredniprzebiegaut |
+------------------+
```

Użycie polecenia:

```
SHOW FULL TABLES WHERE Table_type='BASE TABLE';
```

wyświetla tylko listę tabel w bieżącej bazie danych:

```
+-----------------+------------+
| Tables_in_komis | Table_type |
+-----------------+------------+
| handlowcy       | BASE TABLE |
| klienci         | BASE TABLE |
| modele          | BASE TABLE |
| operacje        | BASE TABLE |
| pracownicy      | BASE TABLE |
| producenci      | BASE TABLE |
| rejestry        | BASE TABLE |
| rezerwacje      | BASE TABLE |
| samochody       | BASE TABLE |
+-----------------+------------+
```

Natomiast użycie polecenia:

```
SHOW FULL TABLES WHERE Table_type='VIEW';
```

wyświetla tylko listę widoków w bieżącej bazie danych:

```
+------------------+------------+
| Tables_in_komis  | Table_type |
+------------------+------------+
| sredniprzebiegaut | VIEW       |
+------------------+------------+
```

Usunięcie widoku: `DROP VIEW SredniPrzebiegAut;`

Można również używać w zapytaniu SQL perspektyw wraz z tabelami trwałymi, np.:

```
SELECT nazwa FROM Modele m, SredniPrzebiegAut s
WHERE m.rocznik_konc = s.rocznik;

SELECT nazwa FROM Modele m INNER JOIN SredniPrzebiegAut s
ON m.rocznik_konc = s.rocznik
```

Perspektywy pełniąc funkcję nowych, często wstępnie uporządkowanych tabel, mogą zawierać dane różniące się od tych, które przechowują oryginalne tabele. Aby uniknąć nieporozumień między użytkownikami systemu, nadajemy poszczególnym argumentom nowe, funkcjonalne nazwy.
Można to uzyskać, definiując perspektywę następująco:

Ćwiczenie 2

```
CREATE VIEW SrdeniPrzebiegAut(grupa, srednia) AS
SELECT rocznik, AVG(przebieg)
FROM Samochody GROUP BY rocznik;
```

Widok wirtualny pozostanie ten sam, ale zmianie ulegną nagłówki kolumn tabeli. W powyższej definicji obie nazwy kolumn podaliśmy w nawiasach po nazwie perspektywy, dlatego nie było już potrzeby powtarzania nazwy sredni_p dla kolumny wynikowej operacji agregacji w klauzuli SELECT.

Perspektywy można usuwać, choć nie istnieją fizycznie. Usuwanie polega na wydaniu polecenia:

```
DROP VIEW nazwa_perspektywy
```

Usunięcie perspektywy nie skutkuje usunięciem rekordów z tabeli bazowej. Znika jedynie widok wirtualny, który wcześniej utworzyliśmy. Tak naprawdę jest to usunięcie samej definicji perspektywy. Wydane polecenia usuwającego tabelę, na podstawie której utworzono perspektywę, spowoduje, że perspektywa ta staje się niedostępna.

Ćwiczenie 3
Wykonaj kolejno w bazie danych **Komis** następujące czynności:

a) utwórz widok ModeleNazwy, w którym wyświetlisz dla każdego modelu jego nazwę (w kolumnie o nazwie model) i nazwę producenta pobraną z tabeli Producenci (w kolumnie o nazwie producent) oraz rocznik początkowy i rocznik końcowy danego modelu,
b) za pomocą widoku policz, ile jest modeli Toyoty i wyniki wyświetl w kolumnie o nazwie Toyota,
c) przy użyciu perspektywy ModeleNazwy wyświetl nazwy wszystkich modeli produkowanych przez Toyotę

UWAGA: W celu lepszego zauważenia działania utworzonej perspektywy ModeleNazwy zanim ją utworzysz dodaj do bazy kolejne modele Toyoty np.: Aygo, Yaris, Auris, Avensis

Ćwiczenie 4

Wykonaj kolejno w bazie danych **Projekty** następujące czynności:

a) utwórz widok PracownicyInfo, w którym wyświetlisz następujące kolumny Pracownik (Nazwisko Imię), Wykształcenie, Staż pracy, Adres, Kod pocztowy, Miejscowość, Województwo
b) za pomocą zapytania SELECT do widoku PracownicyInfo uporządkuj listę pracowników rosnąco wg pola Pracownik,
c) za pomocą zapytania do widoku PracownicyInfo policz wszystkich pracowników z wyższym wykształceniem i wynik wyświetl w kolumnie o nazwie Wyższe,

Ćwiczenie 5

Wykonaj kolejno w bazie danych **Projekty** następujące czynności:

a) utwórz widok PracownicyPrzydzialZadaniaProjekty, w którym wyświetlisz kolumny Pracownik(Nazwisko Imię), Wynagrodzenie(LiczbaGodzin * Stawka), Nazwa projektu, Dziedzina projektu,
b) za pomocą zapytania do widoku PracownicyPrzydzialZadaniaProjekty wyświetl wszystkie nazwy projektów, w których uczestniczył Abacki Jan,
c) za pomocą zapytania do widoku PracownicyPrzydzialZadaniaProjekty wyświetl pogrupowany łączny Koszt wszystkich projektów
d) za pomocą zapytania do widoku PracownicyPrzydzialZadaniaProjekty policz ilu pracowników, brało udział w projekcie "Konserwacja i modernizacja systemu informacyjnego"