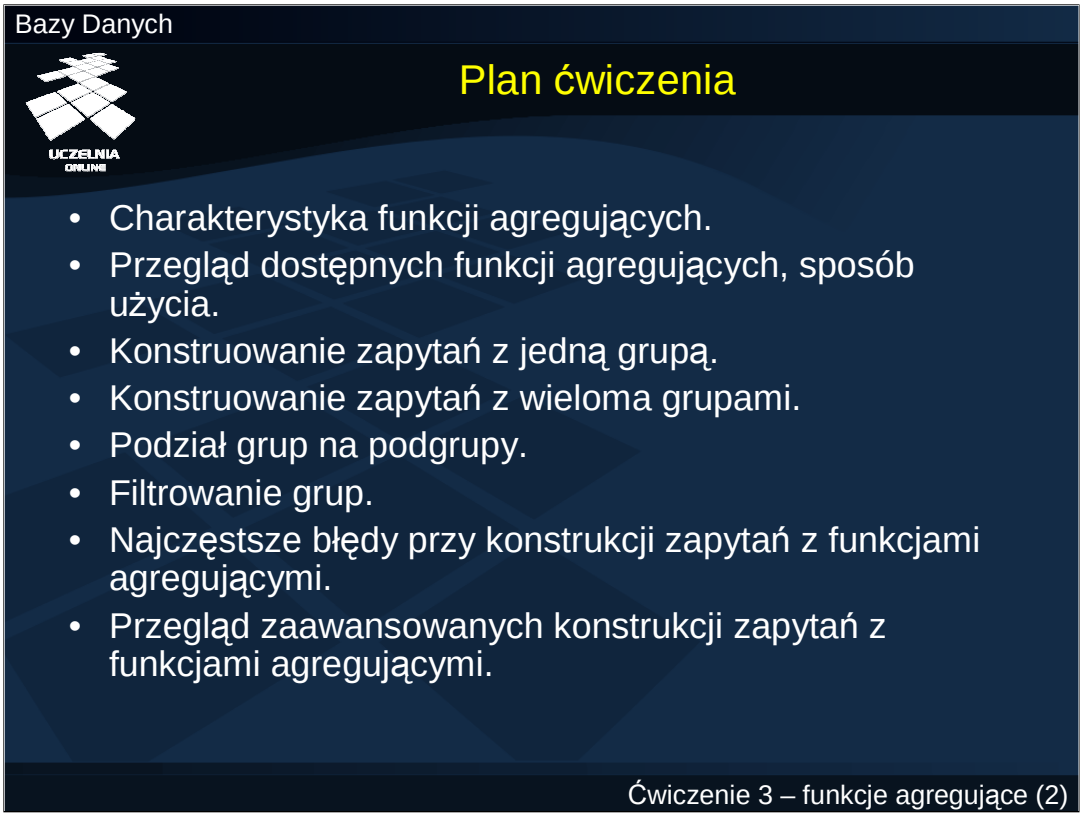
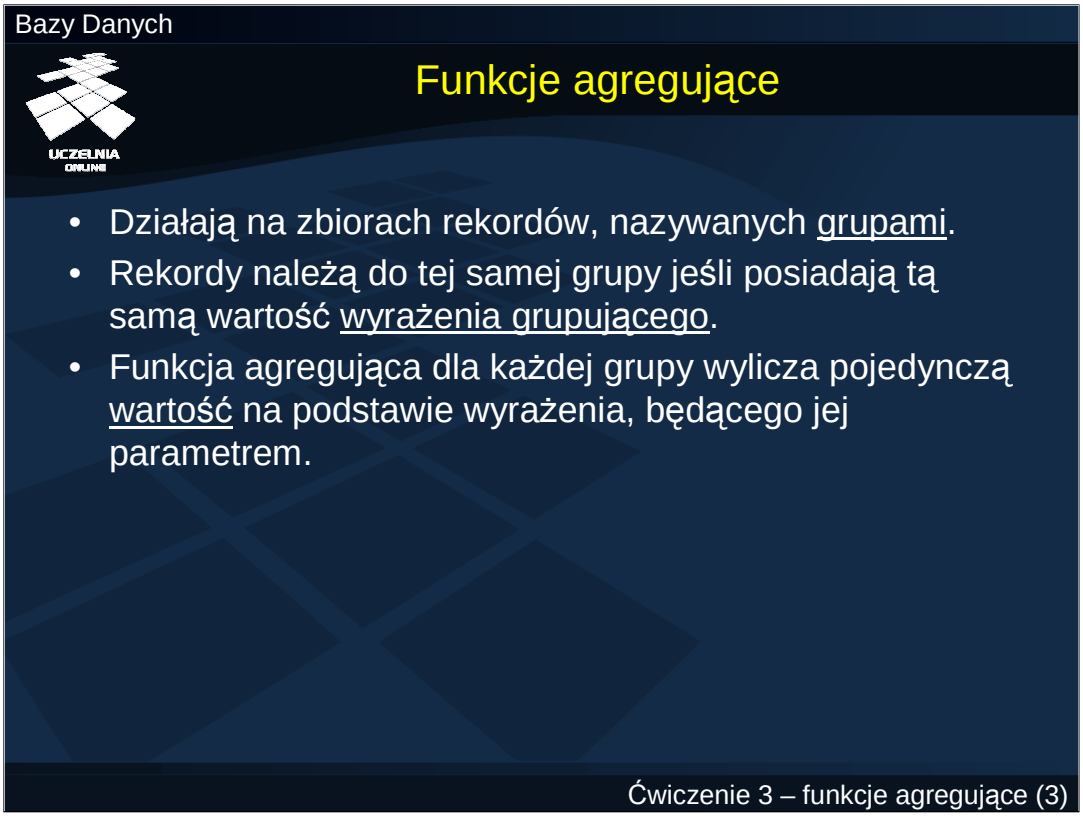

Celem ćwiczenia jest zaprezentowanie zagadnień dotyczących stosowania w zapytaniach języka SQL predefiniowanych funkcji agregujących.

Wymagania:

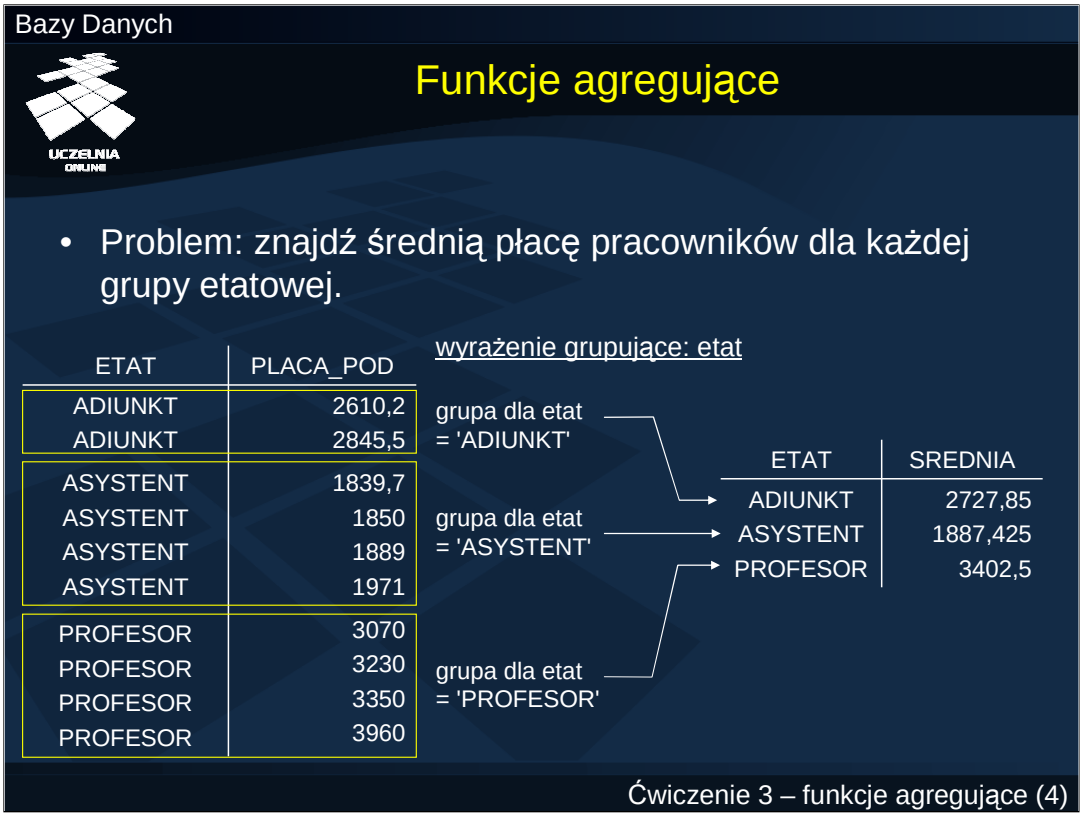
Umiejętność konstrukcji prostych zapytań w języku SQL, użycie funkcji wierszowych w zapytaniach.



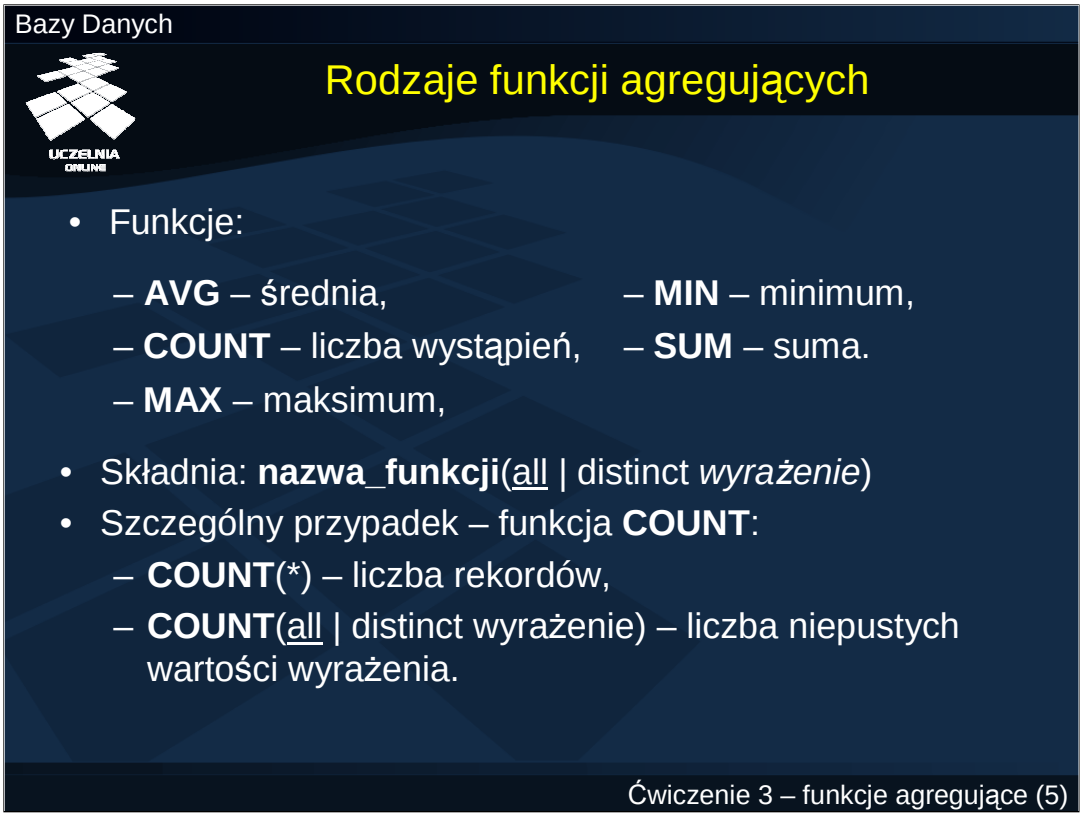

Na początku ćwiczenia zostanie omówiona charakterystyka funkcji agregujących, przegląd dostępnych funkcji agregujących i sposób ich użycia w zapytaniach. Następnie przyjrzymy się konstrukcji zapytań z jedną grupą, wykorzystujących funkcje agregujące. Kolejne zagadnienie to zapytania z wieloma grupami i podział grup na podgrupy. Dalej omówimy filtrowanie grup w zapytaniach oraz scharakteryzujemy najczęściej popełniane błędy. Ćwiczenie zakończymy przeglądem zaawansowanych konstrukcji zapytań z funkcjami agregującymi.




Funkcje agregujące działają na zbiorach rekordów, nazywanych grupami (w przeciwieństwie do funkcji wierszowych, które zawsze działają na jednym rekordzie). Przed zastosowaniem funkcji agregującej konieczne jest podzielenie rekordów na grupy, tzw. grupowanie. Do jednej grupy należą te rekordy relacji, dla których tzw. wyrażenie grupujące zwraca tą samą wartość. Wyrażeniem grupującym jest najczęściej pojedynczy atrybut relacji. Po podziale rekordów na grupy w każdej z grup zostaje zastosowana funkcja agregująca, która wylicza pojedynczą wartość dla grupy. Stąd w wyniku zapytania otrzymujemy tyle rekordów, ile grup zostało utworzonych w wyniku operacji grupowania.




Na bieżącym slajdzie zaprezentowano przykładowe zapytanie z grupowaniem i funkcją agregującą: „znajdź średnią płacę pracowników dla każdej grupy etatowej”. Wyrażeniem grupującym, które dostarcza wartości dzielące zbiór rekordów relacji PRACOWNICY na grupy, jest atrybut ETAT. Przykładowy zbiór rekordów został podzielony na trzy grupy: pierwszą dla wartości ADIUNKT, znajdują się w niej dwa rekordy, następną dla wartość ASYSTENT, należą do niej cztery rekordy, wreszcie ostatnią dla wartości PROFESOR, również z czterema rekordami. Następnie w każdej z grup wartości atrybutu PLACA_POD zostają poddane dzianiu funkcji agregującej, wyliczającej średnią. W wyniku zapytania otrzymujemy po jednym rekordzie dla każdej grupy etatowej: rekord zawiera wartość wyrażenia grupującego, a więc atrybutu ETAT i wyliczoną średnią płacą pracowników w tej grupie.

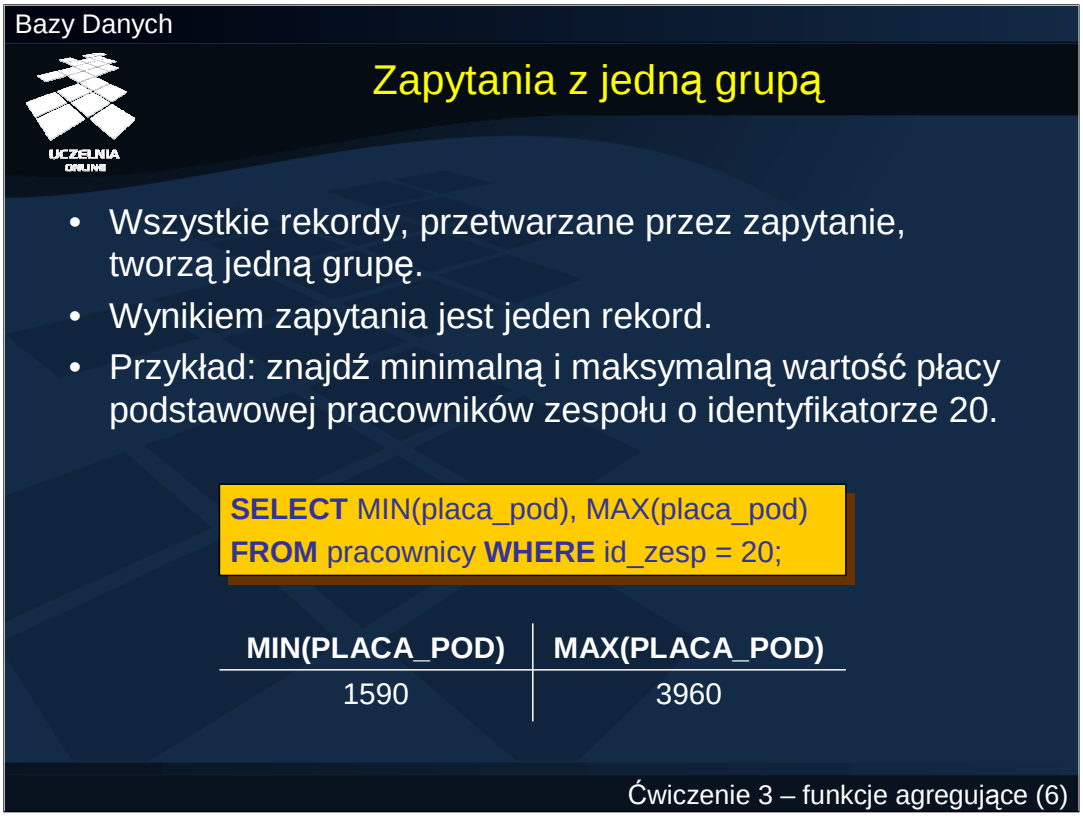



Omówimy teraz poszczególne funkcje agregujące. Do wyliczenia średniej służy funkcja AVG. Funkcja COUNT pozwala na znalezienie liczby wystąpień. Funkcje MAX i MIN umożliwiają znalezienie, odpowiednio, wielkości maksymalnej i minimalnej w zbiorze. Funkcja SUM umożliwia wyliczenie sumy elementów.

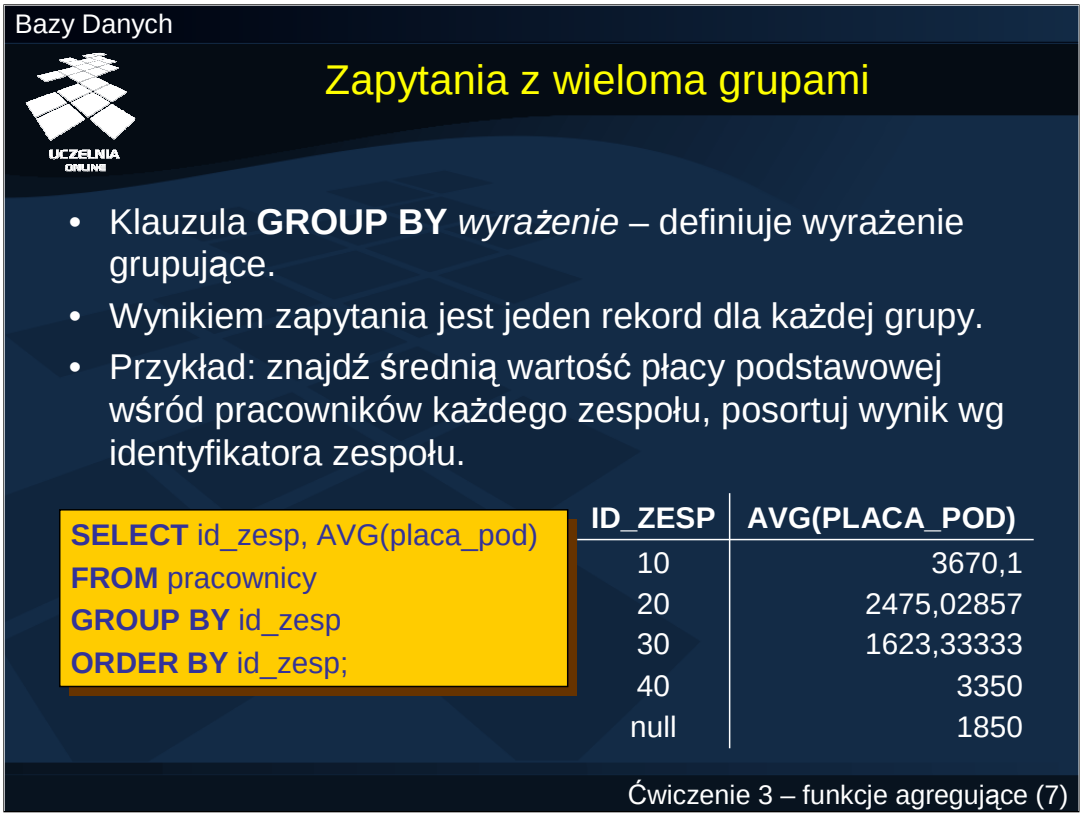
Każda z funkcji posiada jeden parametr, będący wyrażeniem dostarczającym wartości do obliczeń. W przypadku funkcji AVG i SUM wyrażenie musi dostarczać wartości liczbowe, pozostałe funkcje agregujące przyjmują wartości dowolnego typu podstawowego. Przed wyrażeniem można umieścić słowo kluczowe DISTINCT, wówczas do obliczeń zostaną wzięte wartości wyrażenia po eliminacji powtórzeń. Umieszczenie w wywołaniu funkcji samego wyrażenia lub poprzedzenie wyrażenia słowem ALL powoduje, że do obliczeń będą brane wszystkie wartości wyrażenia.

Dodatkowego komentarza wymaga użycie funkcji COUNT. Funkcja zwróci liczbę niepustych wartości wyrażenia w grupie (wartości różnych od null). Dodanie słowa DISTINCT spowoduje policzenie różnych niepustych wystąpień wartości wyrażenia w grupie. Z kolei jeśli w wywołaniu funkcji wyrażenie zastąpimy gwiazdką (symbol *), wówczas zostanie policzona liczba rekordów należących do grupy.

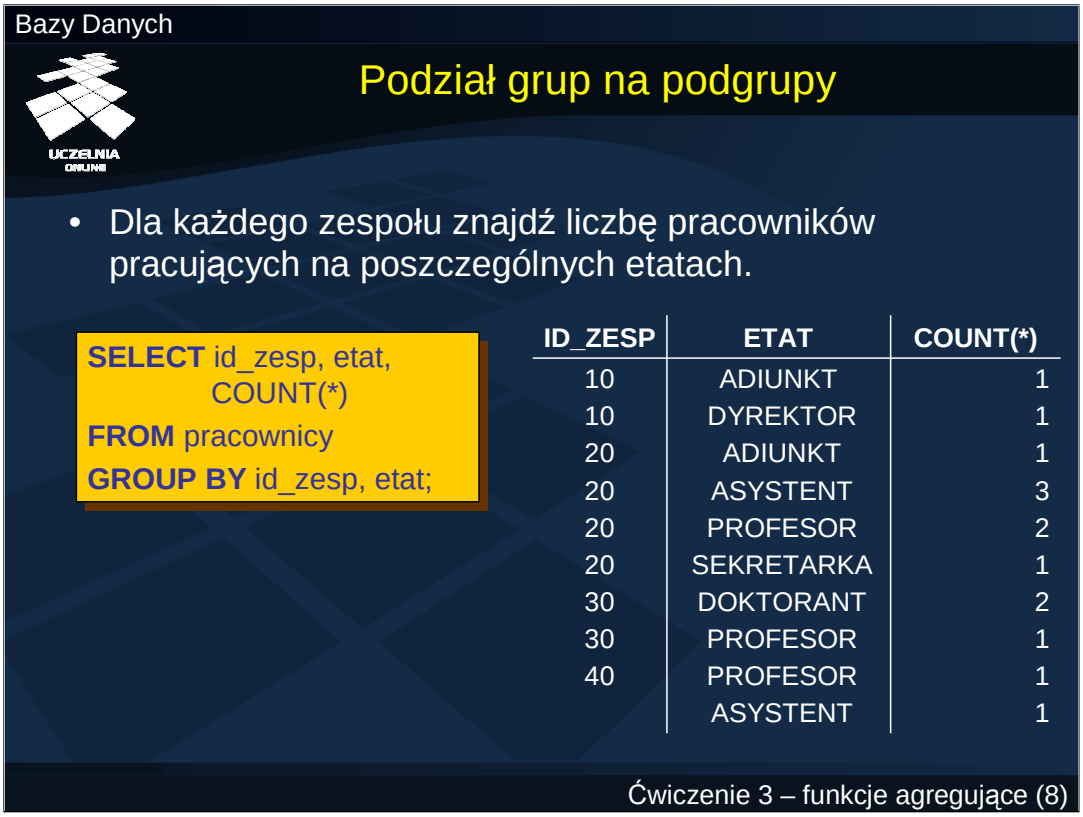
Pozostałe funkcje agregujące również pomijają przy obliczeniach wartości puste, stąd nie ma konieczności stosowania mechanizmów eliminujących wartości puste.



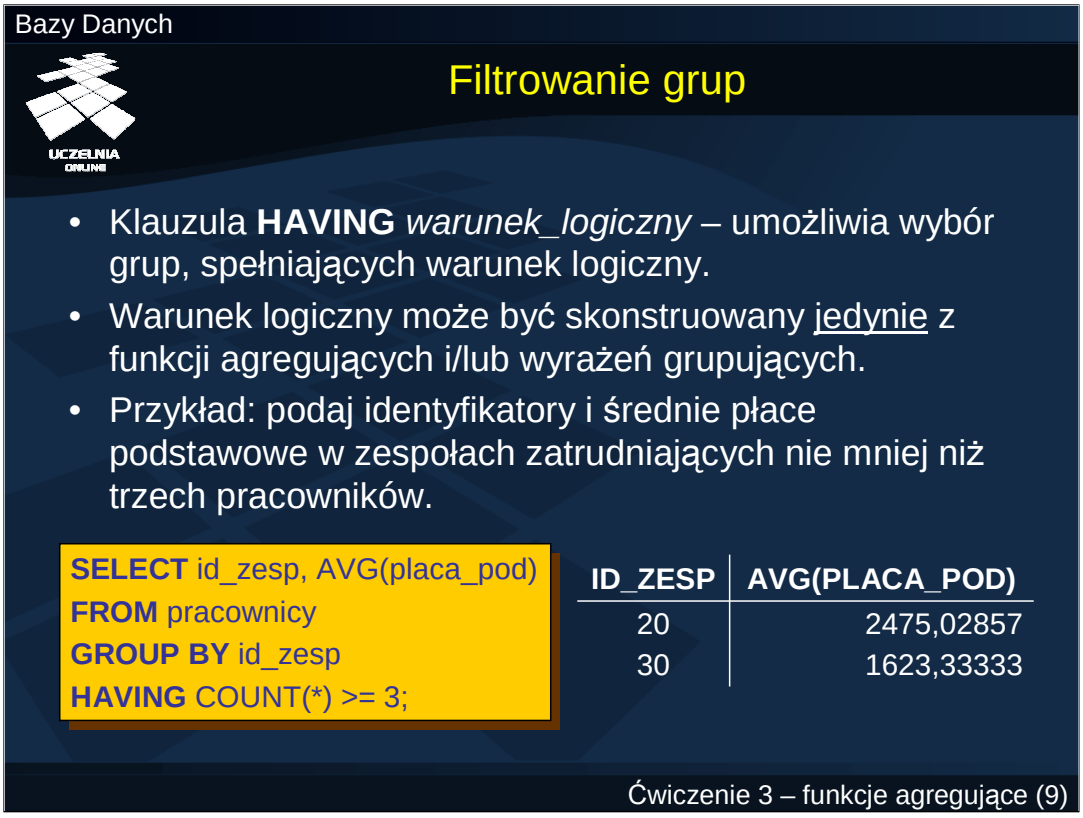

Przejdziemy teraz do omawiania zasad konstruowania zapytań z funkcjami agregującymi. Rozpoczniemy od zapytań z jedną grupą. Są to zapytania bez wyrażenia grupującego, stąd wszystkie rekordy, odczytane przez zapytanie, trafiają do tej samej jednej grupy, a wynikiem zapytania będzie zawsze jeden rekord z wartościami wyliczonymi przez umieszczone w klauzuli SELECT zapytania funkcje agregujące. Przykładowe zapytanie wylicza dwie wartości: minimalną pensję podstawową i maksymalną pensję podstawową wśród pracowników zespołu o numerze 20. Wynikiem zapytania jest jeden rekord z dwiema wartościami.



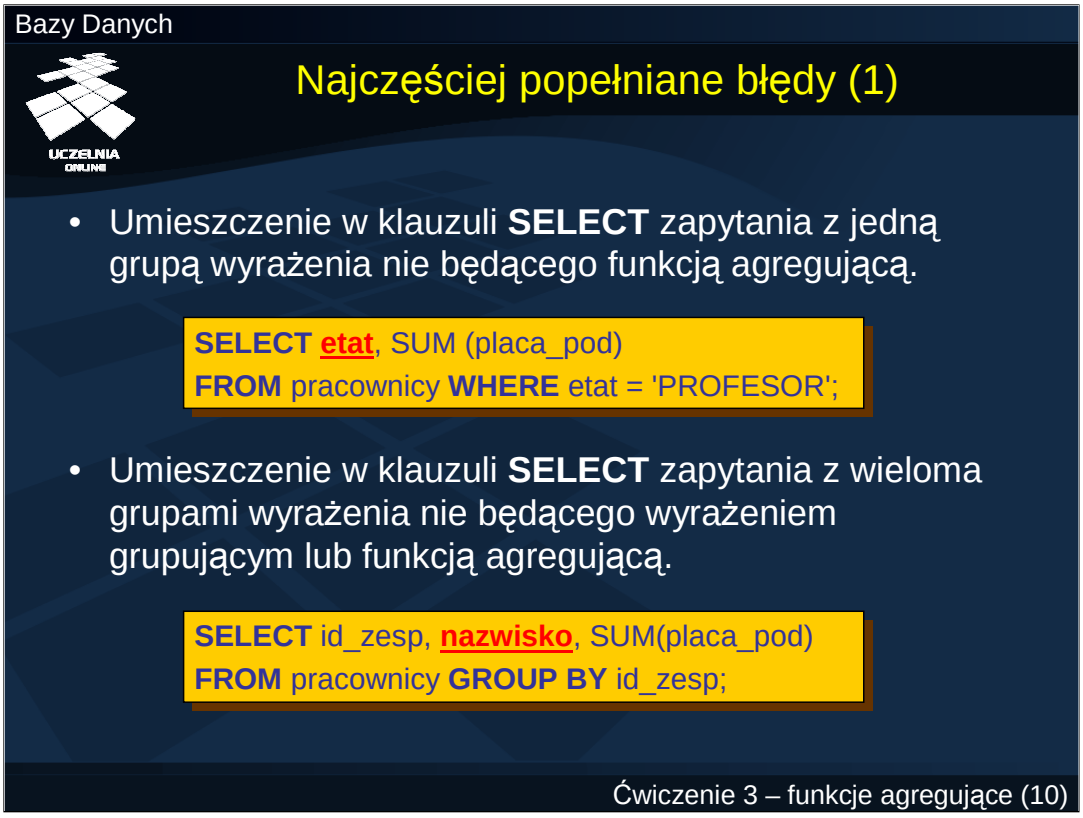

Konstruując zapytania z wieloma grupami konieczne jest zdefiniowanie wyrażenia grupujące. Wyrażenie grupujące umieszcza się w klauzuli GROUP BY. W przykładzie zbiór rekordów relacji PRACOWNICY zostaje podzielony na grupy ze względu na wartość atrybutu ID_ZESP, następnie w każdej z grup zostaje wyliczona wartość średniej płacy pracowników, zatrudnionych w danym zespole. W wyniku zapytania otrzymujemy tyle rekordów, w ilu zespołach pracują pracownicy (rekord z pustą wartością ID_ZESP jest tworzony przez rekordy pracowników, którzy nie należą do żadnego zespołu). Wynik zapytania zostaje posortowany ze względu na wartość wyrażenia grupującego, a więc atrybutu ID_ZESP.




Istnieje możliwość podziału grup, odczytywanych przez zapytanie, na podgrupy. Realizuje się to umieszczając w klauzuli GROUP BY kilka wyrażeń grupujących. W przykładzie zbiór rekordów relacji PRACOWNICY zostaje podzielony na grupy ze względu na wartość atrybutu ID_ZESP (pierwsze wyrażenie grupujące), następnie rekordy w każdej z grup zostają podzielona na podgrupy ze względu na wartość atrybutu ETAT (drugie wyrażenie grupujące). Funkcja agregująca zostaje wykonana w każdej z podgrup. Interpretacja wyniku przykładowego zapytania jest następująca: otrzymujemy dla każdego zespołu informację o liczbie pracowników tego zespołu zatrudnionych na poszczególnych etatach.



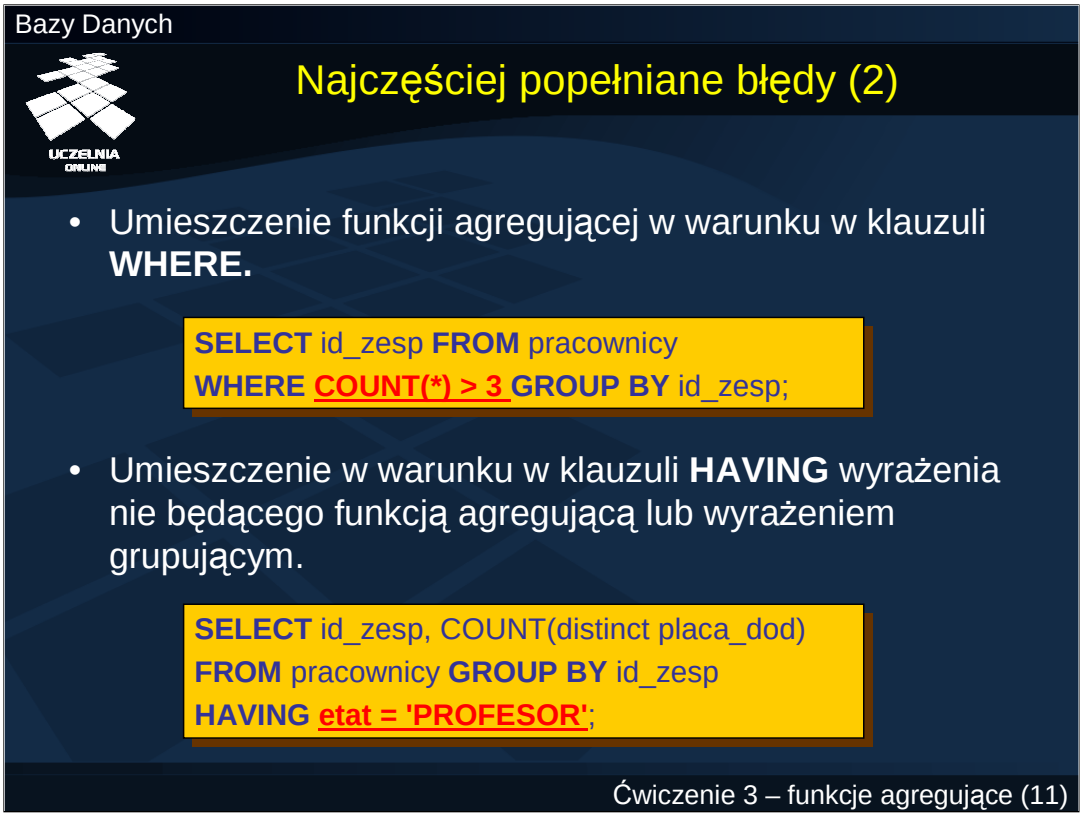

Zapytanie z grupowaniem można dodatkowo wyposażyć w mechanizm filtrowania grup. Realizuje się to umieszczając warunek logiczny w dodatkowej klauzuli HAVING. Należy pamiętać, że w warunku tym można użyć jedynie wyrażeń grupujących i/lub funkcji agregujących. Wartościowanie warunku następuje po utworzeniu grup. Grupy, dla których warunek nie jest spełniony, zostają odrzucone (nie pojawiają się w wyniku zapytania). W przykładzie zbiór rekordów relacji PRACOWNICY zostaje pogrupowany ze względu na wartość atrybutu ID_ZESP, jednak odrzucone zostają te grupy, w których jest mniej niż trzy rekordy. W każdej z pozostałych grup zostaje wyliczona średnia płaca pracowników, należących do zespołów.




Omówione teraz zostaną najczęściej popełniane błędy przy konstrukcji zapytań z funkcjami agregującymi.

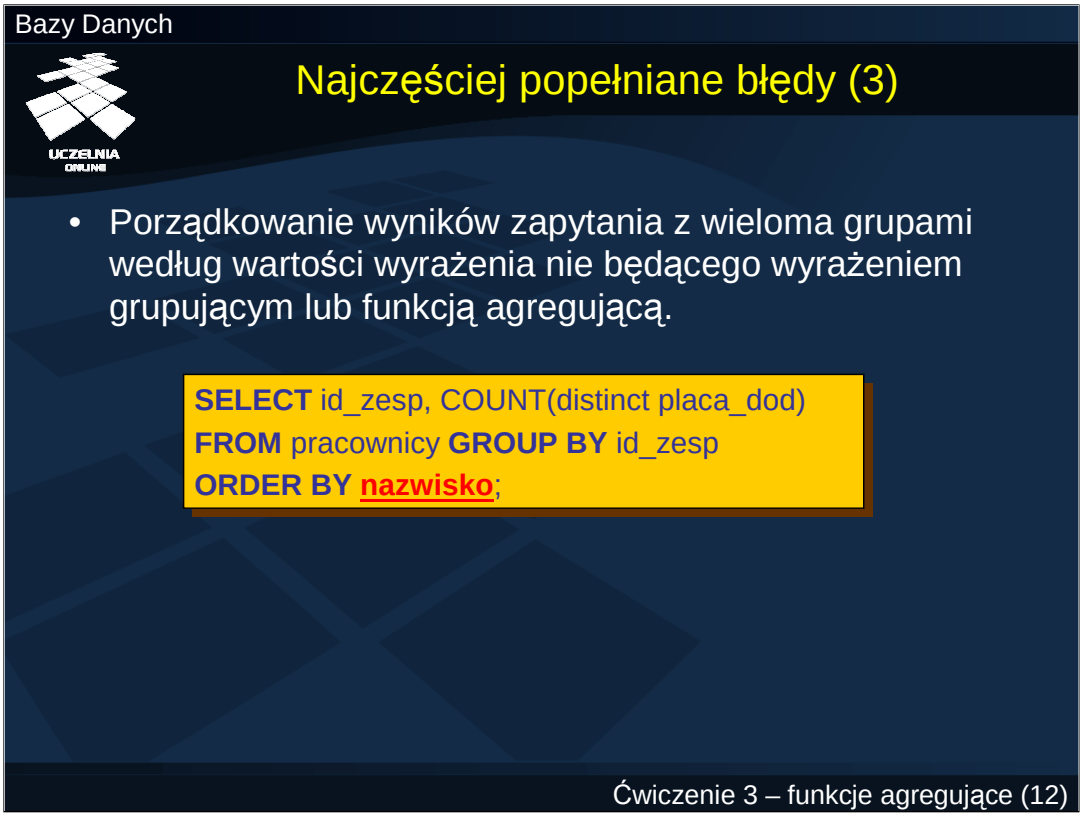
Pierwszym błędem, często pojawiającym się przy zapytaniach z jedną grupą, jest umieszczenie w klauzuli SELECT wyrażenia nie będącego funkcją agregującą (w przykładzie umieszczono atrybut ETAT). W zapytaniach z jedną grupą w klauzuli SELECT mogą być umieszczone jedynie funkcje agregujące.

Kolejny błąd dotyczy zapytań z wieloma grupami i polega na umieszczeniu w klauzuli SELECT wyrażenia, nie będącego wyrażeniem grupującym (a więc nie występującym w klauzuli GROUP BY) lub funkcją agregującą. W przykładzie w klauzuli SELECT umieszczono atrybut NAZWISKO, tymczasem wyrażeniem grupującym jest atrybut ID_ZESP.

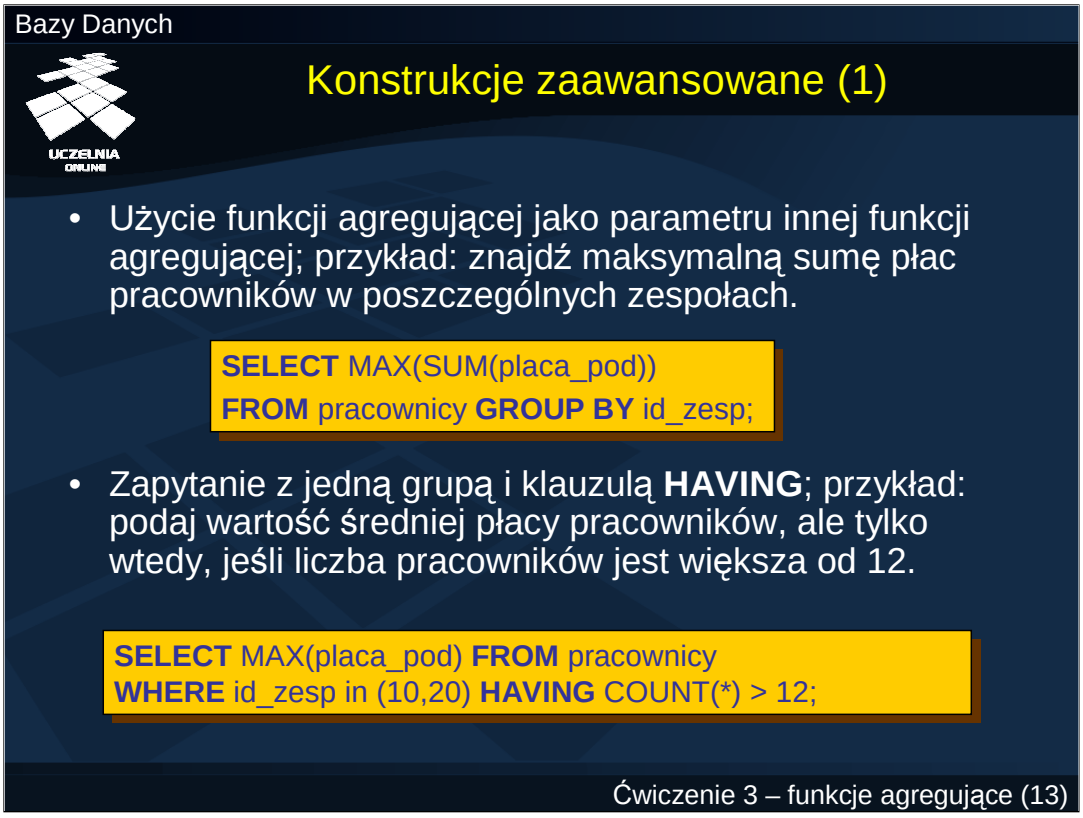



Następnym częstym błędem jest umieszczenie warunku, wykorzystującego funkcję agregującą, w klauzuli WHERE. Taki warunek zawsze powinien być umieszczony w klauzuli HAVING.

Kolejny błąd polega na konstrukcji warunku w klauzuli HAVING z wyrażeniem innym niż wyrażenie grupujące lub funkcja agregująca. Taki warunek powinien zostać umieszczony w klauzuli WHERE.



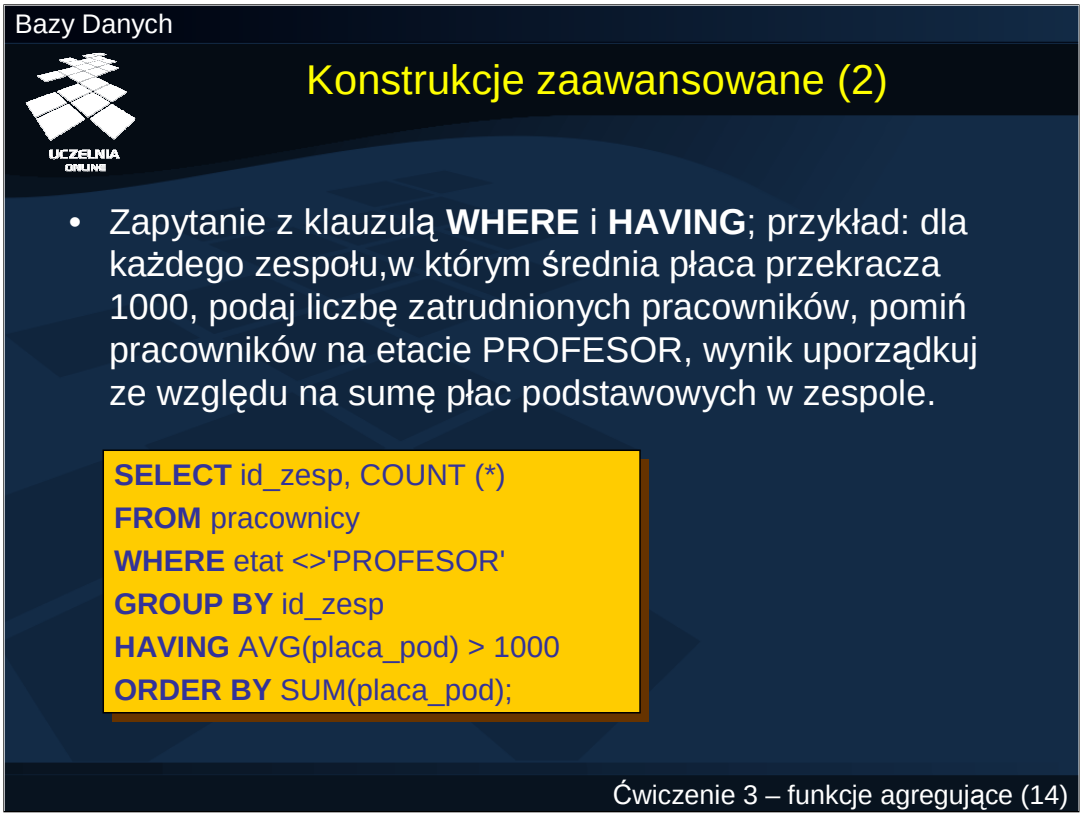

Wreszcie ostatni błąd, polegający na użyciu do sortowania wyniku zapytania z grupowaniem wyrażenia nie będącego wyrażeniem grupującym bądź funkcją agregującą. W zaprezentowanym przykładzie wynik zapytania może zostać posortowany jedynie ze względu na wartość atrybutu ID_ZESP (wyrażenie grupujące) lub wartość wyliczaną przez dowolną funkcję agregującą.



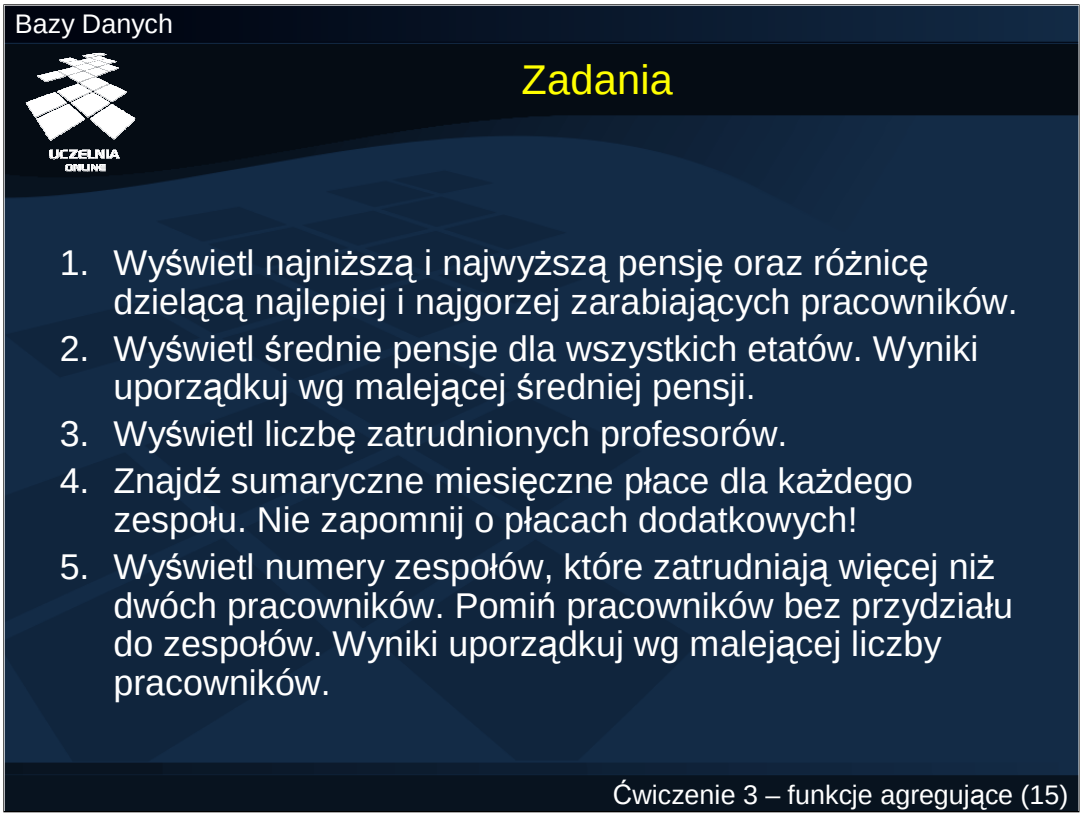

Dokonamy teraz przeglądu zaawansowanych konstrukcji wykorzystujących funkcje agregujące.

Pierwsza konstrukcja to użycie funkcji agregującej jako parametru innej funkcji agregującej. Przykładowe zapytanie należy wykonywać dwuetapowo. Pierwszy etap to wykonanie zapytania w postaci „SELECT SUM(placa_pod) FROM pracownicy GROUP BY id_zesp”. Wynik tego zapytania to zbiór sum płac podstawowych pracowników w poszczególnych zespołach. Drugi etap to wybór spośród wyliczony wartości wielkości maksymalnej. W wyniku otrzymujemy jeden rekord (zapytanie wykonywane w drugim etapie jest w istocie zapytaniem działającym na jednej grupie).

Kolejny przykład prezentuje zapytanie z jedną grupą (a więc bez zdefiniowanego wyrażenia grupującego), w którym zastosowano klauzulę HAVING. W takim przypadku warunek filtrujący zostaje zastosowany do jedynej grupy zapytania, jeśli warunek nie jest spełniony, zapytanie zwraca wynik pusty. W przykładzie otrzymamy wynik, maksymalną płacę podstawową pracowników z zespołów 10 i 20, pod warunkiem, że w obu zespołach zatrudniono w sumie ponad 12 osób.




Wreszcie zapytanie, w którym użyto wszystkich zaprezentowanych dotąd klauzul. Należy pamiętać o kolejności wykonywania klauzul. Jako pierwszy zostaje przetworzony warunek w klauzuli WHERE, dokonujący filtrowania rekordów relacji PRACOWNICY ze względu na wartość atrybutu ETAT. Do dalszego przetwarzania zostaną wzięte tylko te rekordy, gdzie ETAT różni się od ciągu znaków „PROFESOR”. Następnie realizowane jest grupowanie, wyrażeniem grupującym, umieszczonym w klauzuli GROUP BY, jest atrybut ID_ZESP. Powstałe grupy są filtrowane ze względu na warunek logiczny w klauzuli HAVING. Wreszcie wyliczana jest wartość funkcji agregującej, umieszczonej w klauzuli SELECT a wynik zostaje posortowany ze względu na wyrażenie umieszczone w klauzuli ORDER BY.




Bieżący slajd rozpoczyna zbiór zadań, których celem jest utrwalenie wiadomości o konstrukcji zapytań wykorzystujących funkcje agregujące.


## Zadania

6. Wyświetl średnie pensje wypłacane w ramach poszczególnych etatów i liczbę pracowników zatrudnionych na danym etacie. Pomiń pracowników zatrudnionych po 1990 roku.
7. Dla każdego pracownika wyświetl pensję najgorzej zarabiającego podwładnego. Wyniki uporządkuj wg malejącej pensji.
8. Sprawdź, czy identyfikatory pracowników są unikalne.


Bieżący slajd przedstawia rozwiązania zadań (1), (2), (3), (4) i (5), których treść zacytowano poniżej.

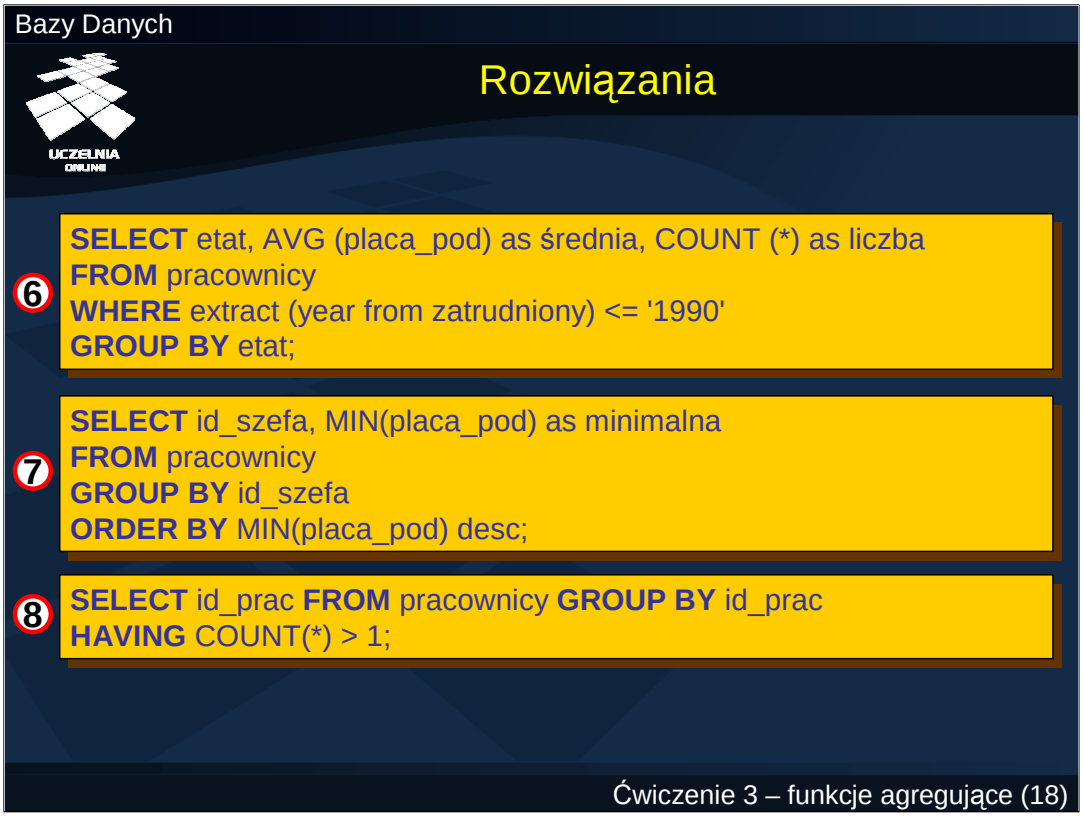
(1) Wyświetl najniższą i najwyższą pensję oraz różnicę dzielącą najlepiej i najgorzej zarabiających pracowników.

(2) Wyświetl średnie pensje dla wszystkich etatów. Wyniki uporządkuj wg malejącej średniej pensji.

(3) Wyświetl liczbę zatrudnionych profesorów.

(4) Znajdź sumaryczne miesięczne płace dla każdego zespołu. Nie zapomnij o płacach dodatkowych!

(5) Wyświetl numery zespołów, które zatrudniają więcej niż dwóch pracowników. Pomiń pracowników bez przydziału do zespołów. Wyniki uporządkuj wg malejącej liczby pracowników.

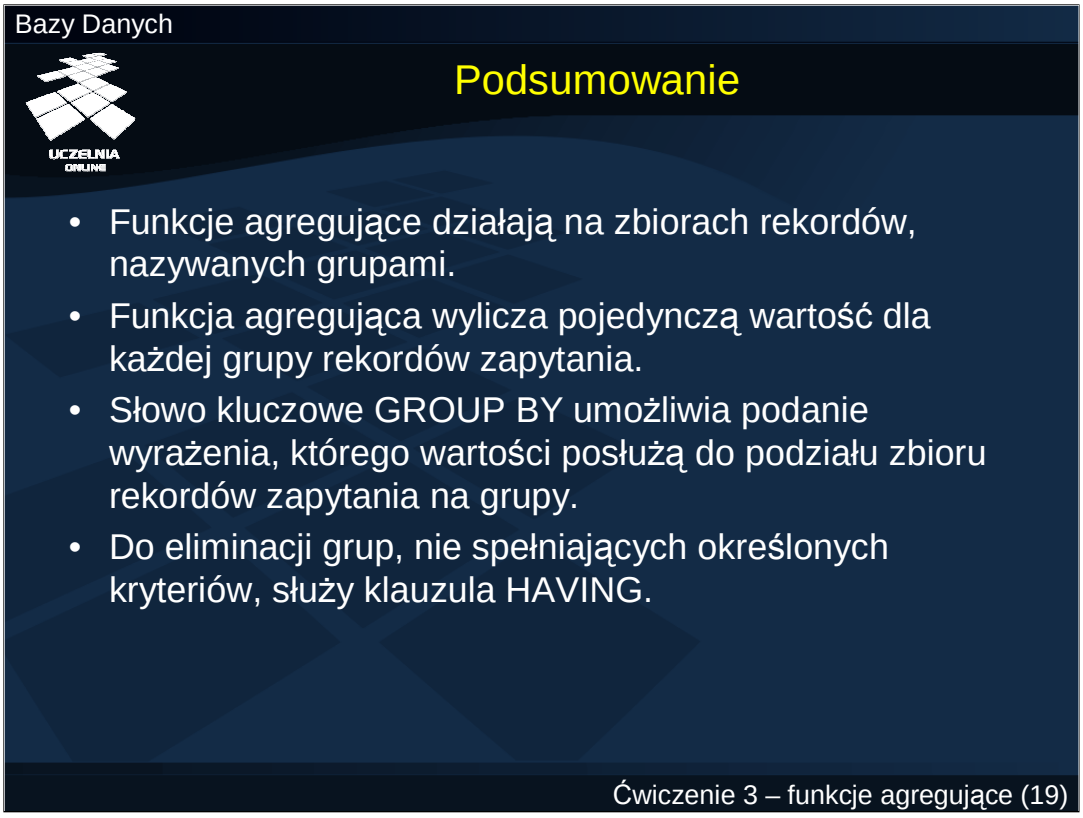



Bieżący slajd przedstawia rozwiązania zadań (6), (7) i (8), których treść zacytowano poniżej.

(6) Wyświetl średnie pensje wypłacane w ramach poszczególnych etatów i liczbę pracowników zatrudnionych na danym etacie. Pomiń pracowników zatrudnionych po 1990 roku.

(7) Dla każdego pracownika wyświetl pensję najgorzej zarabiającego podwładnego. Wyniki uporządkuj wg malejącej pensji.

(8) Sprawdź, czy identyfikatory pracowników są unikalne.




W zakończonym ćwiczeniu zaprezentowano funkcje agregujące. Są to funkcje, działające na grupach rekordów, wyliczające dla każdej z grup dokładnie jedną wartość. Do podziału zbioru rekordów na grupy służy klauzula GROUP BY zawierająca wyrażenie, wokół którego wartości tworzone są grupy rekordów. Do eliminacji grup z wyniku zapytania służy klauzula HAVING, w której podaje się warunek filtrujący.

Każde z omówionych zagadnień zostało utrwalone przez serię zadań.