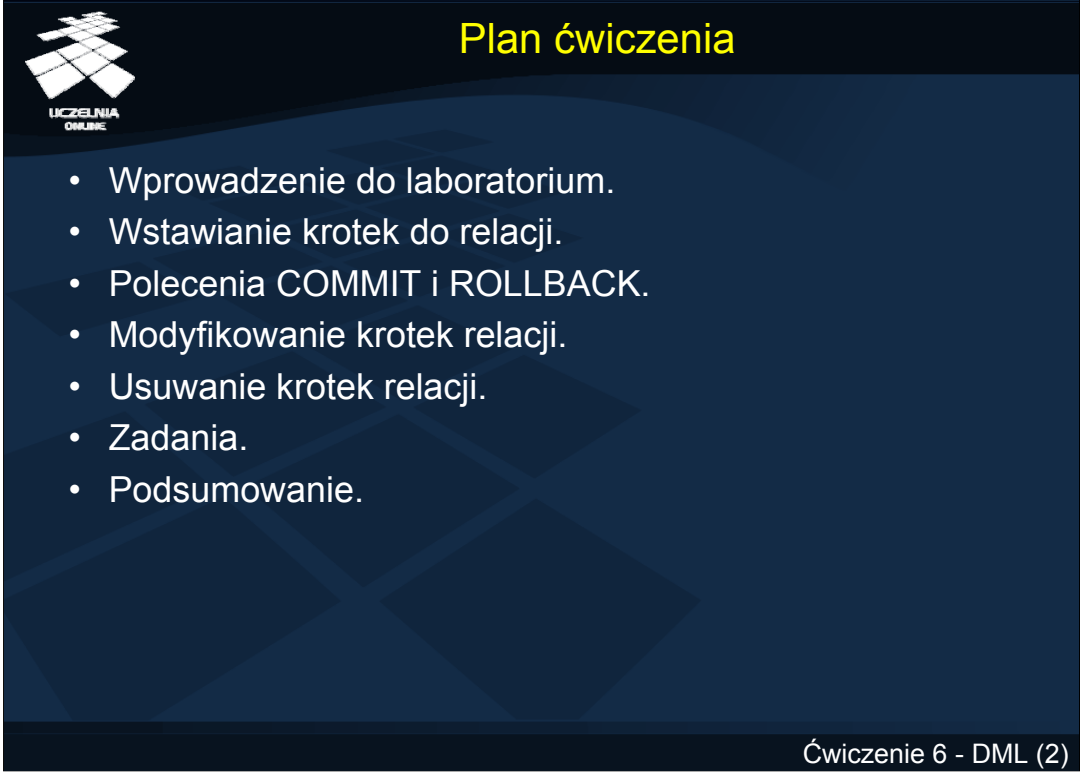
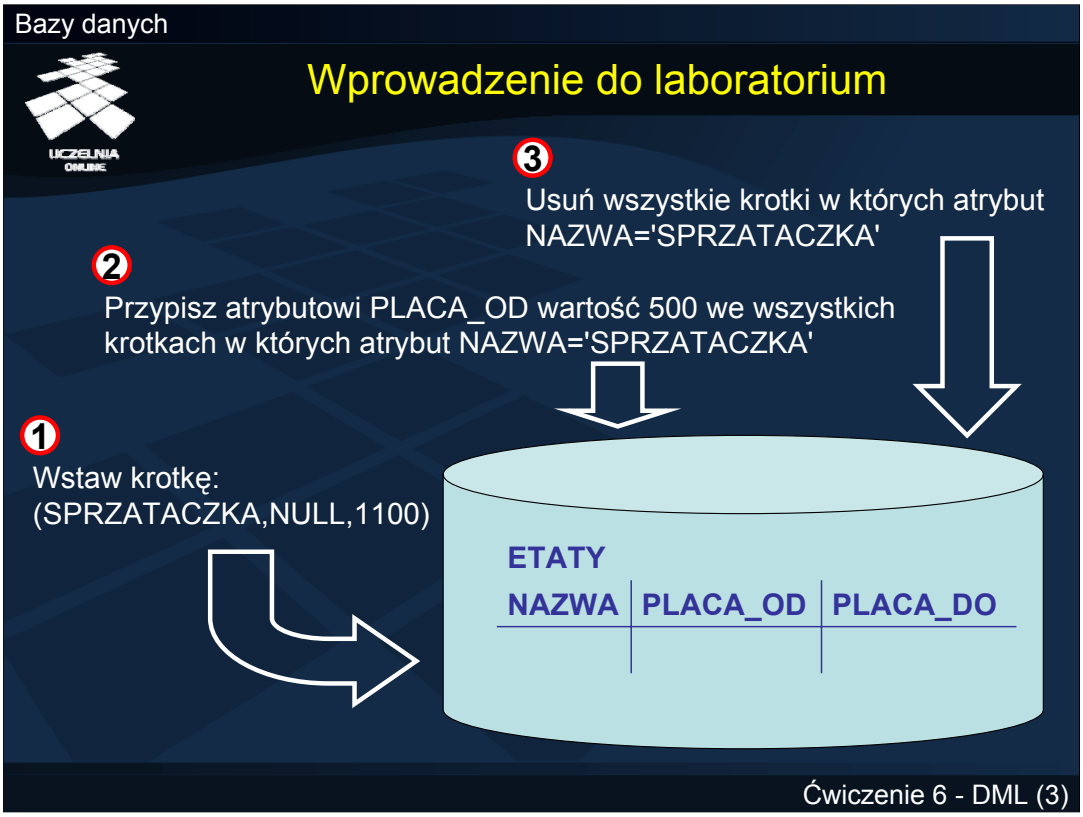
Na dotychczasowych zajęciach zapoznaliście się Państwo z poleceniem SELECT pozwalającym na wykonywanie zapytań do bazy danych i odczytywanie danych zawartych w relacjach. Celem tego ćwiczenia jest zapoznanie państwa z poleceniami, które można wykorzystać do: wstawiania, modyfikacji i usuwania krotek z relacji.

Wymagania:

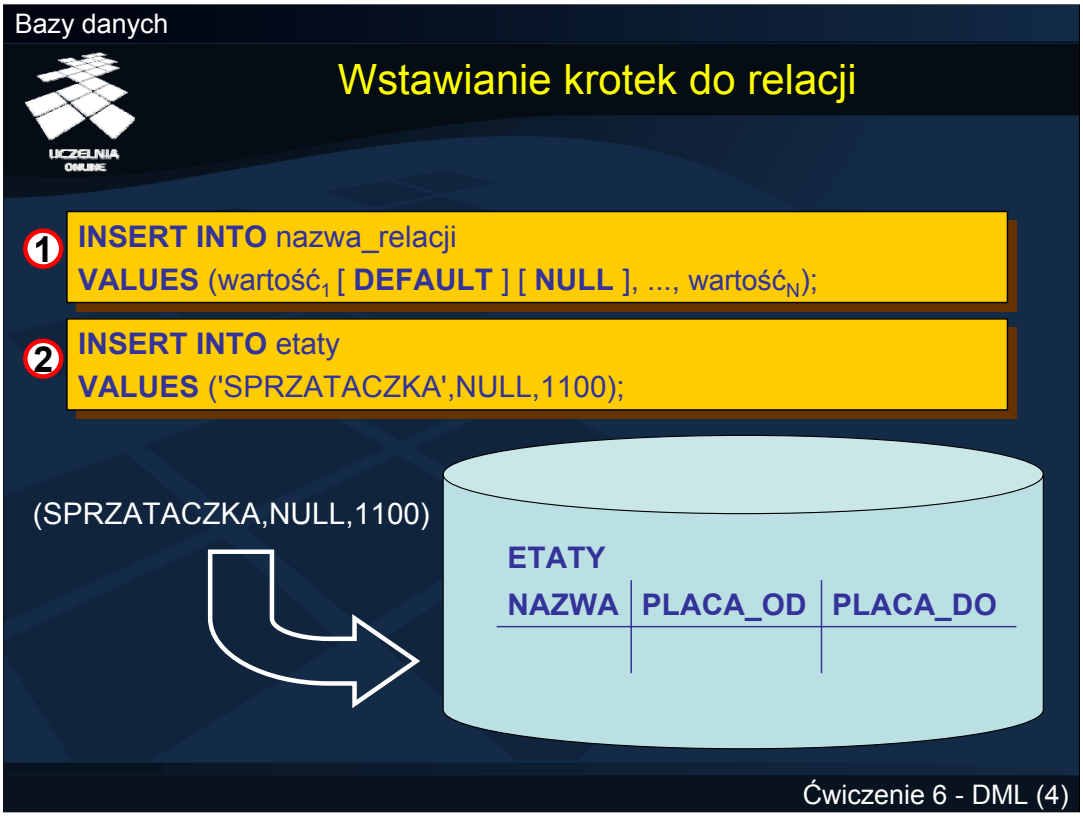
Znajomość składni polecenia SELECT omawianego na poprzednich ćwiczeniach z baz danych i umiejętność konstrukcji skomplikowanych zapytań z użyciem połączeń i podzapytań.



Ćwiczenie rozpocznie się od wprowadzenia do laboratorium, po którym omówimy składnię polecenia INSERT pozwalającego na wstawianie krotek do relacji. Następnie, wstępnie omówimy polecenia COMMIT i ROLLBACK dzięki którym możliwe będzie zapamiętanie bądź usunięcie wykonanych wcześniej modyfikacji danych. W dalszej części ćwiczenia, zapoznacie się państwo z poleceniami pozwalającymi na modyfikowanie wartości atrybutów krotek zapisanych w relacji i usuwanie krotek z relacji. Każdy z wyżej wymienionych tematów zakończony jest zadaniami do samodzielnego wykonania. Na końcu ćwiczenia przedstawimy państwu kilka dodatkowych zadań, które powinniście państwo wykonać w celu nabrania wprawy w posługiwaniu się poleceniami przedstawionymi na ćwiczeniu. Ćwiczenie zakończymy slajdem podsumowującym omówioną tematykę.



Na poprzednich ćwiczeniach z systemów baz danych poznawaliście państwo polecenie SELECT dzięki któremu można odczytywać dane z relacji, oraz wykonywać na nich proste operacje i obliczenia. Polecenie to nie przydaje się jednak na wiele, jeżeli nie można wstawiać do relacji własnych danych. Na ćwiczeniu piątym z baz danych zapoznacie się Państwo z poleceniami: INSERT pozwalającym na wstawianie nowych krotek do relacji (1), UPDATE pozwalającym na modyfikację wartości atrybutów krotek zapisanych w relacji (2) i, ostatecznie, poleceniu DELETE pozwalającym na usuwanie danych z relacji (3). Omówimy również polecenia COMMIT i ROLLBACK dzięki którym możliwe będzie zapamiętanie bądź usunięcie przeprowadzonych wcześniej modyfikacji danych



Do wstawiania krotek do relacji służy polecenie INSERT. Najprostszą wersję tego polecenia przedstawiono na przykładzie (1). Polecenie zaczyna się od słów kluczowych INSERT INTO po którym podaje się nazwę relacji, słowo kluczowe VALUES i w nawiasie listę wartości oddzielonych przecinkami. Wartości na tej liście odpowiadają kolejnym atrybutom relacji (w kolejności w jakiej atrybuty te zostały zdefiniowane). Wpisami na liście wartości mogą być: słowa kluczowe NULL i DEFAULT oraz konkretne wartości atrybutów i ewentualnie podzapytania zwracające jedną wartość. Jeżeli zamiast konkretnej wartości poda się NULL, wówczas nowa krotka będzie miała pustą wartość atrybutu odpowiadającego pozycji na której wpisano NULL. Podanie DEFAULT spowoduje, że w odpowiednim atrybucie zostanie zapisana jego wartość domyślna. W przykładowych relacjach, na których wykonywaliście państwo ćwiczenia, atrybuty zdefiniowano w następującej kolejności:

- relacja PRACOWNICY: ID_PRAC, NAZWISKO, IMIE, ETAT, ID_SZEFA, ZATRUDNIONY, PLACA_POD, PLACA_DOD i ID_ZESP,
- relacja ETATY: NAZWA, PLACA_OD, PLACA_DO,
- relacja ZESPOLY: ID_ZESP, NAZWA, ADRES.

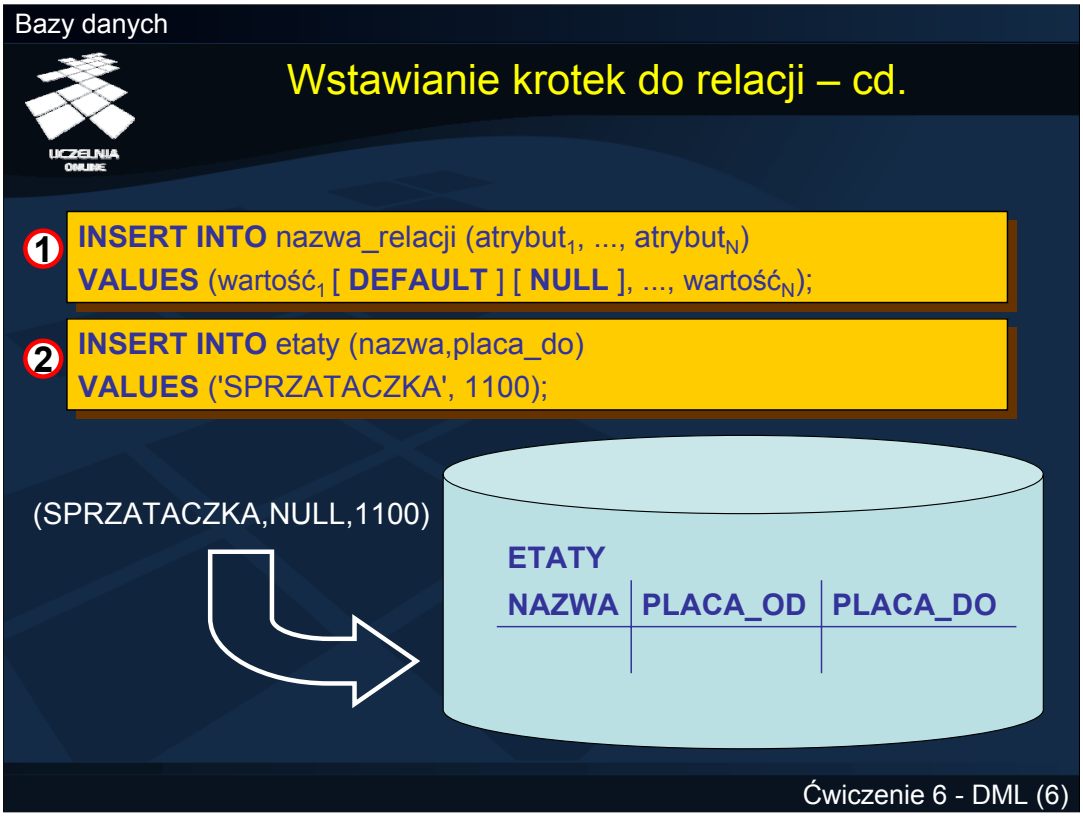
Na przykładzie (2) pokazano polecenie wstawiające do tabeli ETATY krotkę reprezentującą etat 'SPRZATACZKA', której praca minimalna jest niezdefiniowana, a płaca maksymalna wynosi 1100 zł. Działanie przykładu demonstruje rysunek na slajdzie.

Jak wspomniano wcześniej, w poleceniu INSERT można zagnieździć podzapytanie zwracające konkretną wartość, która powinna zostać zapisana we wstawianej krotce.

Przykładowe polecenie INSERT z podzapytaniem wygląda następująco:

```
INSERT INTO etaty
VALUES ('SPRZATACZKA',NULL,
 (SELECT min(placa_pod)*0.9
  FROM pracownicy)
 );
```

Polecenie to powoduje wstawienie do relacji ETATY krotki reprezentującej etat SPRZATACZKA, której maksymalna placa (atrybut PLACA_DO) powinna wynosić 90% najmniejszej płacy pracownika z relacji PRACOWNICY.

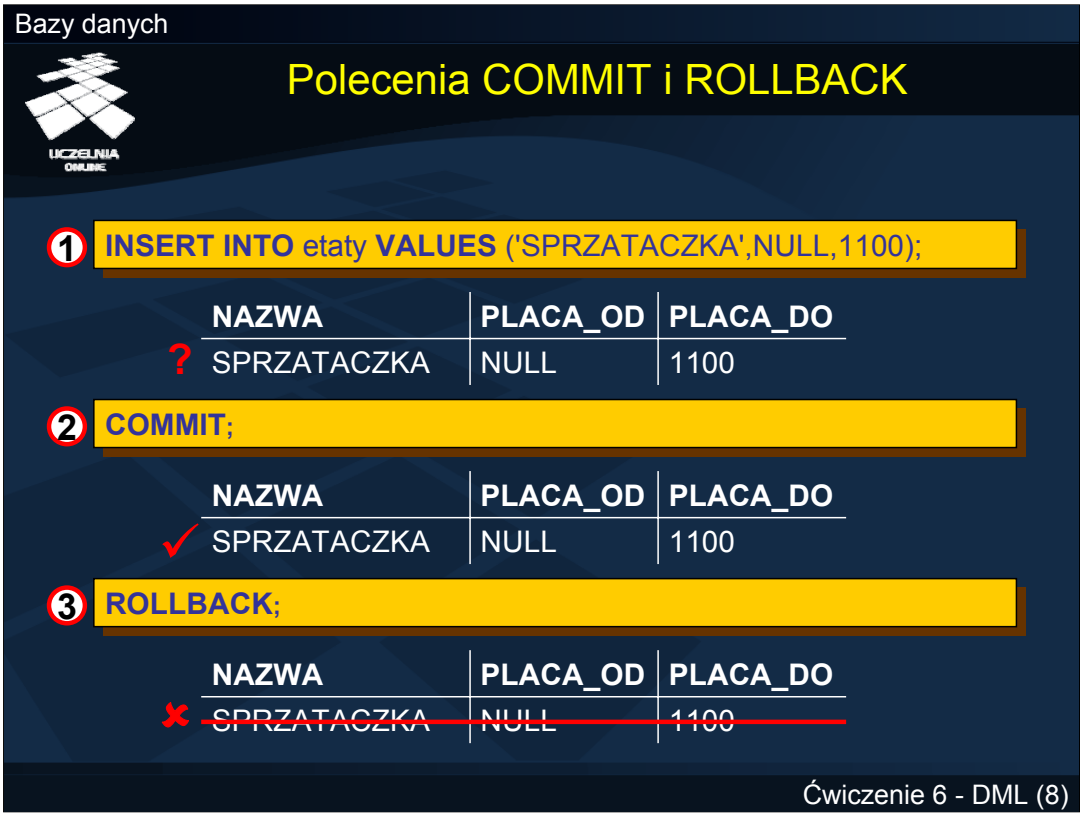


Podstawową wadą wersji polecenia INSERT pokazanej na poprzednim slajdzie jest to, iż należy znać kolejność w jakiej zostały zdefiniowane w relacji atrybuty. Dodatkowo, trzeba podawać na liście wartości, wartości dla wszystkich atrybutów i jawnie specyfikować NULL, jeżeli dana wartość w momencie wstawiania krotki jest nieznana, bądź nieistotna. Aby poradzić sobie z tym problemem, można zdefiniować których atrybutów dotyczą kolejne wartości na liście wartości podanej w poleceniu INSERT. Robi się to podając za nazwą relacji w poleceniu INSERT listę atrybutów, do których będą kolejno wstawiane wartości z listy wartości. Przykład (1) pokazuje rozszerzoną składnię polecenia INSERT. Przykład (2) pokazuje polecenie wstawiające do relacji ETATY krotkę opisujące etat SPRZATACZKA, dla której definiowana jest tylko placa maksymalna (atrybut PLACA_DO). Do atrybutów, dla których nie podano wartości w poleceniu, wstawiana jest wartość NULL, bądź wartość domyślna, jeżeli taka została zdefiniowana (sposób definiowania wartości domyślnych zostanie omówiony na następnych zajęciach). Rysunek na slajdzie demonstruje sposób działania polecenia na przykładzie (2).

Polecenie INSERT można wykorzystać do wstawiania do relacji wyników zapytania. Aby to zrobić, należy zastąpić zapytaniem (polecenie SELECT) słowo kluczowe VALUES i listę wartości. Taka modyfikacja jest możliwa dla obu przedstawionych wcześniej wersji polecenia INSERT (z podaniem listy atrybutów i bez). Zapytanie użyte w poleceniu INSERT musi zwracać w relacji wynikowej tyle samo atrybutów ile jest atrybutów w relacji do której wstawiane są nowe krotki, bądź tyle, ile podano na liście atrybutów za nazwą relacji do której wstawiane są krotki. Zmodyfikowaną składnię obu wersji polecenia INSERT przedstawiono na przykładach (1) i (2). Przykład (3) pokazuje polecenie wstawiające do relacji ETATY krotki będące wynikiem zapytania:

SELECT 'NAD'||nazwa,placa_od,placa_do+1000 FROM etaty;

Ponieważ za nazwą relacji ETATY w poleceniu INSERT podano listę atrybutów (NAZWA, PLACA_OD, PLACA_DO), to kolejne atrybuty relacji wynikowej zapytania będą odpowiadać kolejnym atrybutom z tej listy. Do atrybutu NAZWA zostaną zapisane wyniki wyrażenia 'NAD'||nazwa (np. NADPROFESOR, NADADIUNKT), do atrybutu PLACA_OD zostaną przepisane wartości atrybutu PLACA_OD z wyniku zapytania, a do atrybutu PLACA_DO zostaną zapisane wyniki wyrażenia placa_do+1000. Kilka przykładowych krotek, które mogą zostać wstawione do relacji ETATY w wyniku wykonania polecenia (3) przedstawiono na slajdzie.



Do wykonywania zadań z poleceń DML przydadzą się państwu jeszcze dwa polecenia: COMMIT i ROLLBACK. Dokładne wytłumaczenie funkcjonalności tych poleceń zostanie przedstawione na późniejszych ćwiczeniach. Na tym ćwiczeniu opiszemy jedynie praktyczne znaczenie wykonania poleceń COMMIT i ROLLBACK. Rozważmy sytuację, w której wstawiono do relacji ETATY krotkę (przykład (1)). Taka krotka jeszcze nie jest zapisana do tej relacji na stałe. Aby zatwierdzić wszystkie dokonane w relacjach zmiany należy wykonać polecenie COMMIT (2). Jeżeli jednak zdecydujemy, że krotka nie powinna być zapisana do relacji, to możliwe jest jej usunięcie za pomocą polecenia ROLLBACK (3). Polecenie ROLLBACK wycofuje wszystkie zmiany w relacjach aż do ostatniego polecenia COMMIT.

Uwaga ! O ile nie zostanie w zadaniu zaznaczone inaczej, wycofuj (ROLLBACK) wyniki każdego wykonanego zadania, aby dane w relacjach wykorzystywane do kolejnych zadań były nie zmienione. Jeżeli jednak zatwierdzisz modyfikacje możesz przywrócić początkowy stan relacji PRACOWNICY, ETATY i ZESPOLY wykonując jeszcze raz skrypt pracownicy.sql załączony do kursu.

Zadanie (1)

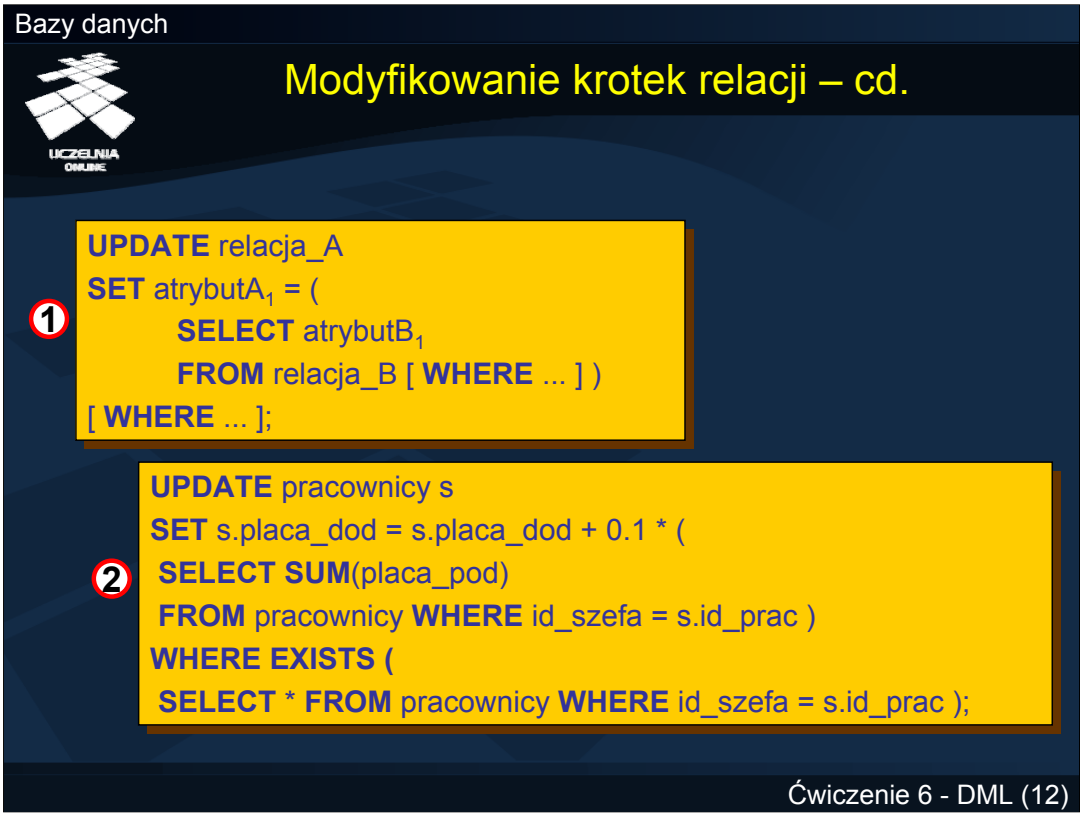
- Znajdź największą wartość atrybutu ID_ZESP (ID_ZESP jest kluczem głównym) w tabeli ZESPOLY (za pomocą polecenia SELECT). Wstaw do tabeli ZESPOLY nowy zespół o nazwie „SYSTEMY WBUDOWANE” i wartości ID_ZESP większej od odczytanej poprzednio wartości. Adresu nie podawaj. Sprawdź, czy krotka rzeczywiście została zapisana do tabeli.

Slajd pokazuje rozwiązanie zadania (1), którego treść przytoczono poniżej.

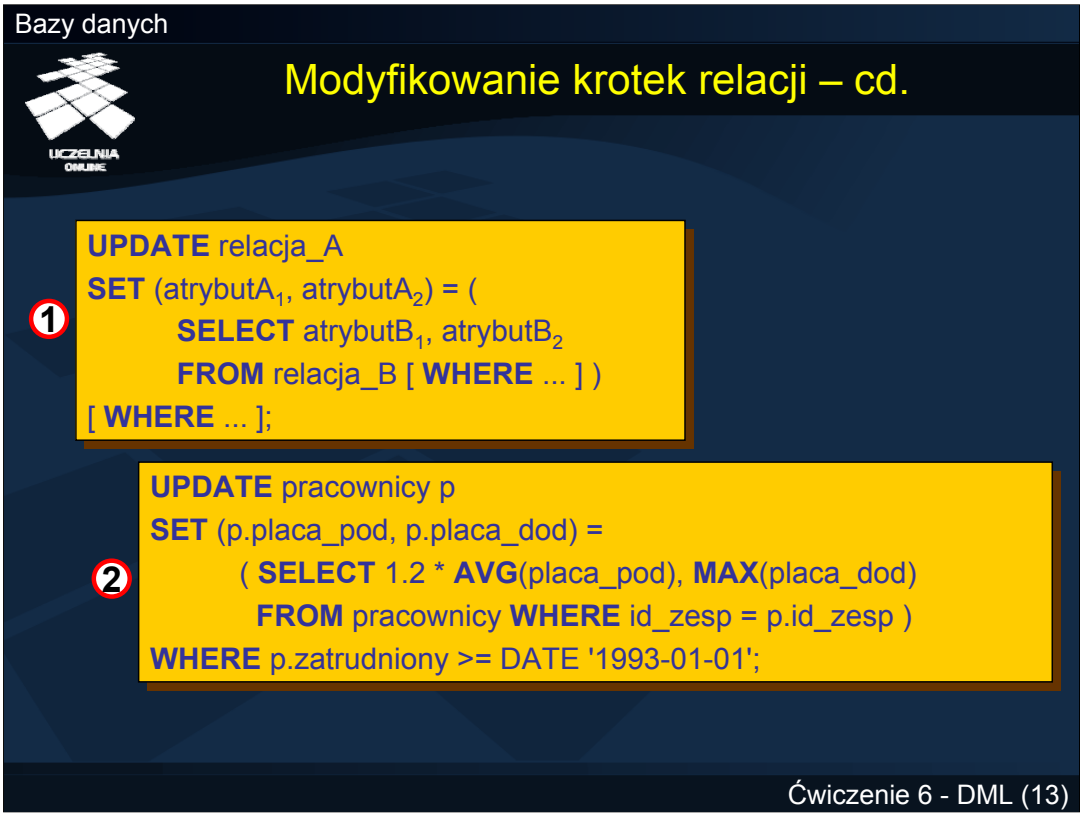
Znajdź największą wartość atrybutu ID_ZESP (ID_ZESP jest kluczem głównym) w tabeli ZESPOLY (za pomocą polecenia SELECT). Wstaw do tabeli ZESPOLY nowy zespół o nazwie „SYSTEMY WBUDOWANE” i wartości ID_ZESP większej od odczytanej poprzednio wartości. Adresu nie podawaj. Sprawdź, czy krotka rzeczywiście została zapisana do tabeli.

Aby zmodyfikować wartości atrybutów w krotkach, które znajdują się w relacji, należy użyć polecenia UPDATE. Podstawową składnię polecenia UPDATE przedstawiono na przykładzie (1). Polecenie rozpoczyna się od słowa kluczowego UPDATE, po którym podaje się nazwę modyfikowanej relacji, słowo kluczowe SET i listę operacji przypisania atrybut=wartość, gdzie atrybutem może być dowolny atrybut relacji, a wartością dowolne wyrażenie, m. in: stała, wyrażenie matematyczne na wartościach atrybutów modyfikowanej krotki (starych) albo podzapytanie. Całe polecenie można opcjonalnie zakończyć znaną z polecenia SELECT klauzulą WHERE. W wyniku wykonania polecenia UPDATE modyfikacji ulegają wszystkie krotki, które spełniają warunek w klauzuli WHERE. Sposób obliczenia nowych wartości atrybutów definiują operacje przypisania wymienione po przecinku za słowem kluczowym SET. Jeżeli warunek WHERE zostanie pominięty, modyfikacji ulegają wszystkie krotki w relacji.

Przykładowe polecenie UPDATE (2) wykonuje następujące operacje: znajduje wszystkie krotki w relacji PRACOWNICY, dla których atrybut NAZWISKO ma wartość „Grzybowska”, zmienia im wartość atrybutu ETAT na PROFESOR oraz zwiększa dwu i półkrotnie wartość atrybutu PLACA_POD.



Jak wspomniano na poprzednim slajdzie, możliwe jest stosowanie podzapytań w poleceniu UPDATE. Podzapytania można wykorzystać tutaj zarówno do zaawansowanego obliczania nowych wartości atrybutów, jak i w klauzuli WHERE do zaawansowanego wyszukiwania krotek, które mają zostać zmodyfikowane. Należy pamiętać jedynie o tym, aby podzapytanie obliczające nową wartość atrybutu zwracało relację wynikową złożoną z jednego atrybutu i jednej krotki. Podzapytania w klauzuli WHERE można definiować w sposób analogiczny do sposobu definiowania podzapytań w poleceniu SELECT. Składnię polecenia UPDATE z wykorzystaniem podzapytania przedstawiono na przykładzie (1). Przykład (2) zawiera zaawansowane polecenie UPDATE wykorzystujące podzapytania. Rozpocznijmy analizę tego polecenia od klauzuli WHERE. Według warunku zdefiniowanego w tej klauzuli, modyfikacji ulegną jedynie te krotki relacji PRACOWNICY, dla których istnieje przynajmniej jedna krotka w wyniku podzapytania. Jest to podzapytanie skorelowane odszukujące wszystkich podwładnych rozważanego w poleceniu UPDATE pracownika. W celu odróżnienia atrybutów relacji PRACOWNICY przeglądanych przez podzapytanie od atrybutów modyfikowanej relacji, na początku polecenia UPDATE nadano modyfikowanej relacji pracownicy alias „s”. Podsumowując, warunek WHERE przykładowego polecenia SELECT mówi nam, iż zmodyfikowani mają zostać jedynie ci pracownicy, którzy posiadają jakichś podwładnych. Przejdźmy obecnie do operacji przypisania tego polecenia. W operacji tej, przypisujemy atrybutowi PLACA_DOD starą wartość tego atrybutu powiększoną o 10% wartości zwróconej przez podzapytanie. Podzapytanie jest również skorelowane i zwraca sumę wartości atrybutu PLACA_POD krotek opisujących podwładnych rozważanego przez polecenie UPDATE pracownika. Podsumowując, można powiedzieć, że polecenie UPDATE zwiększa płacę dodatkową wszystkim pracownikom posiadającym podwładnych o 10% sumarycznej płacy podstawowej wszystkich jego podwładnych.

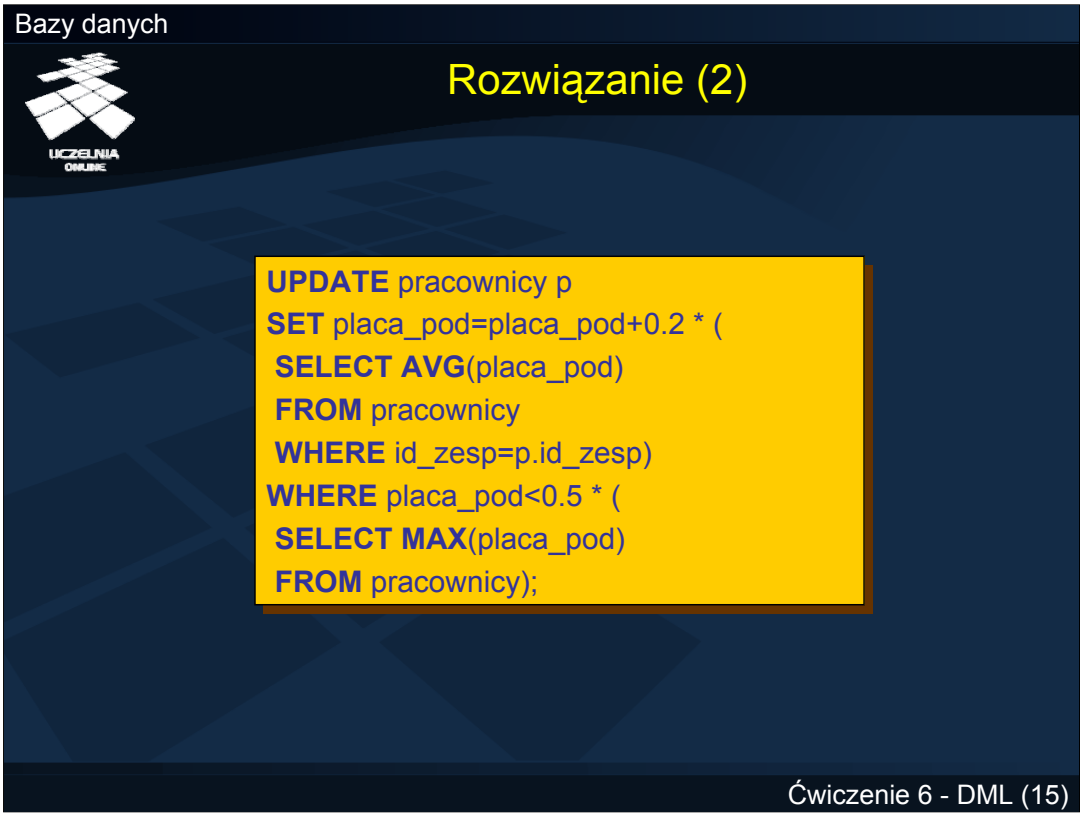


Na poprzednim slajdzie powiedziano, że podzapytanie obliczające nową wartość atrybutu musi zwracać dokładnie jedną wartość (jedna krotka z jednym atrybutem). W szczególnych przypadkach można to wymaganie częściowo ominąć stosując składnię przedstawioną na przykładzie (1). W składni tej, po lewej stronie przypisania podano listę atrybutów, a po prawej stronie podzapytanie zwracające jedną krotkę, o takiej liczbie atrybutów jaką podano na liście po lewej stronie przypisania. Wartości kolejnych atrybutów relacji wynikowej podzapytania są przypisywane do kolejnych atrybutów na liście.

Rozważmy polecenie UPDATE na przykładzie (2). Według klauzuli WHERE modyfikacji będą podlegać jedynie pracownicy zatrudnieni po 1 stycznia 1993. Rozważmy teraz podzapytanie. Jest to podzapytanie skorelowane, które oblicza 120% średniej płacy podstawowej i maksymalną płacę dodatkową wszystkich pracowników w zespole pracownika, który jest rozważany przez polecenie UPDATE. Wartości te są następnie przypisywane odpowiednio do atrybutów PLACA_POD i PLACA_DOD modyfikowanej krotki. Podsumowując, polecenie ustala, wszystkim pracownikom zatrudnionym po 1 stycznia 1993, płacę podstawową na 120% średniej płacy w ich zespole, a płacę dodatkową na największą płacę dodatkową w ich zespole.

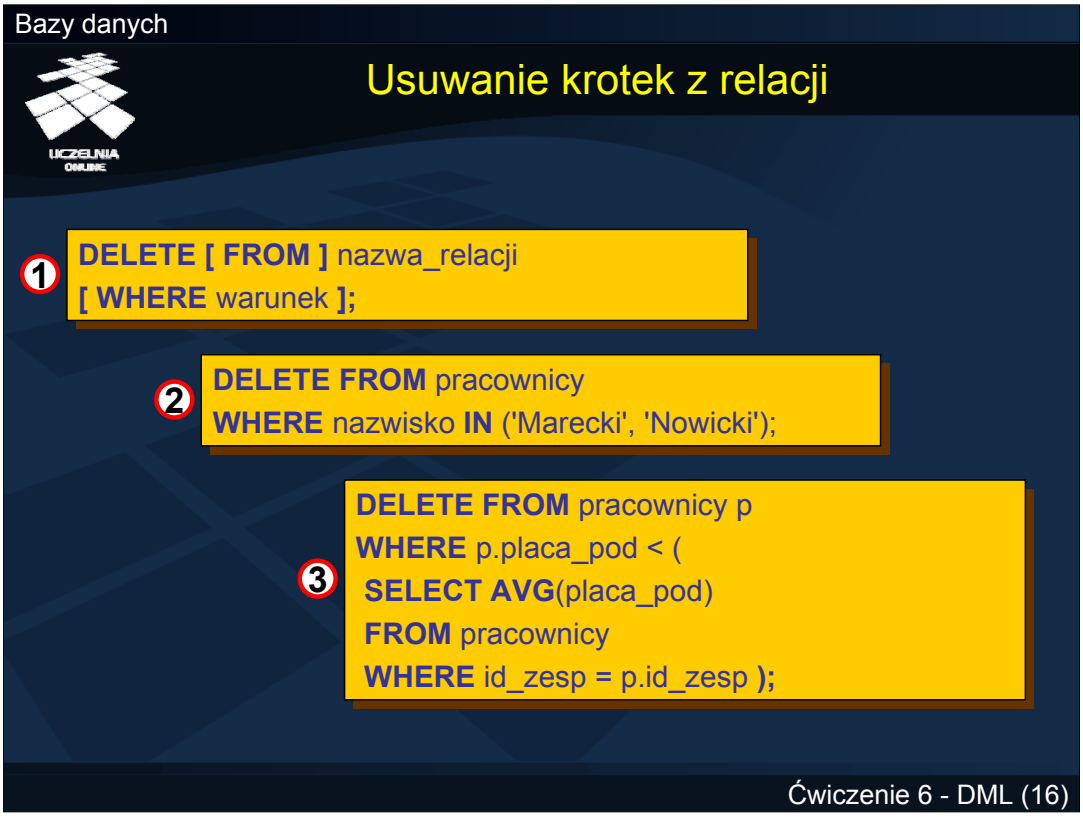
Zadanie (2)

- Wszystkim pracownikom zarabiającym mniej niż 50% najwyższej płacy podstawowej podnieś pensję o 20% średniej płacy w ich zespole.



Slajd pokazuje rozwiązanie zadania (2), którego treść przytoczono poniżej.

Wszystkim pracownikom zarabiającym mniej niż 50% najwyższej płacy podstawowej podnieś pensję o 20% średniej płacy w ich zespole.



Aby usunąć krotki z relacji należy użyć polecenia DELETE. Składnia polecenia DELETE została przedstawiona na przykładzie (1). Polecenie rozpoczyna się od słowa kluczowego DELETE, po którym można opcjonalnie podać słowo kluczowe FROM, a następnie nazwę relacji i opcjonalną klauzulę WHERE. Polecenie DELETE powoduje usunięcie wszystkich krotek z relacji, której nazwę podano w poleceniu, i które spełniają warunek w klauzuli WHERE. Jeżeli klauzula WHERE zostanie pominięta, to usuwane są wszystkie krotki z relacji. W klauzuli WHERE można stosować dowolnie skomplikowane podzapytania, dzięki którym można określić, które krotki mają zostać usunięte. Opcjonalne słowo kluczowe FROM w poleceniu nie zmienia jego działania, ale jego użycie zwiększa czytelność polecenia.

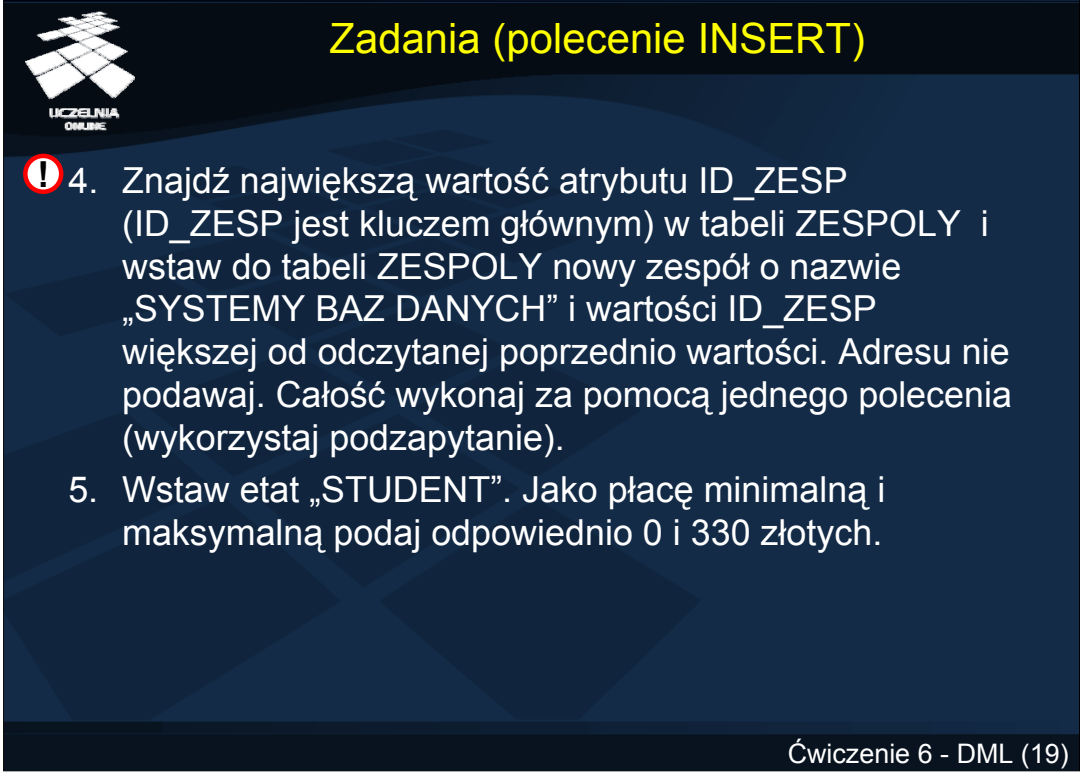
Przykład (2) pokazuje polecenie DELETE usuwające wszystkich pracowników, których nazwiska znajdują się na podanej liście (tutaj usuwani są Marecki i Nowicki). Przykład (3) pokazuje sposób użycia polecenia DELETE z podzapytaniem w klauzuli WHERE. Jest to podzapytanie skorelowane, które zwraca średnią płacę pracowników pracujących w tym samym zespole, co pracownik rozważany przez polecenie DELETE. Można zatem powiedzieć, że polecenie to usuwa wszystkich pracowników, którzy zarabiają mniej od średniej płacy w ich zespole.

Zadanie (3)

- Usuń podwładnych pracownika o nazwisku Nowicki, którzy zarabiają więcej niż 1000 zł.

Slajd pokazuje rozwiązanie zadania (3), którego treść przytoczono poniżej.

Usuń podwładnych pracownika o nazwisku Nowicki, którzy zarabiają więcej niż 1000 zł.



(!) Wszystkie zadania na wykorzystanie polecenia INSERT są od siebie zależne i muszą być wykonane po kolei, bez wycofywania pośrednich wyników. Rezultat tych zadań wycofaj po zakończeniu ostatniego z nich.

Zadania (polecenie INSERT) – cd.

6. Zatrudnij się na etacie „STUDENT” w zespole „SYSTEMY BAZ DANYCH” podając jako datę zatrudnienia aktualną datę systemową, płacę podstawową ustaw na 330 złotych, a dodatkową na 5% średniej płacy w zespole ADMINISTRACJA. Jako ID_PRAC podaj największą wartość ID_PRAC w tabeli PRACOWNICY zwiększoną o 10. Całość wykonaj za pomocą jednego polecenia (wykorzystaj podzapytania).

Zadania (polecenie UPDATE)

7. Podnieś do średniej pracowniczej pensję najgorzej zarabiającym pracownikom.
8. Uaktualnij płace dodatkowe pracowników zespołu 20. Nowe płace dodatkowe mają być równe średniej płacy pracowników, których przełożonym jest Marecki.
9. Pracownikom zespołu o nazwie 'SYSTEMY ROZPROSZONE' daj 25% podwyżkę.
10. Zmniejsz dolną granicę widełek (PLACA_OD) o 10%, a górną granicę widełek (PLACA_DO) zwiększ o 20% dla etatów, na których sumaryczna płaca jest największa pośród wszystkich etatów.

(!) W przeciwieństwie do zadań na polecenie INSERT, w zadaniach na polecenia UPDATE i DELETE wycofuj wyniki każdego zadania po jego wykonaniu.

Zadania (Polecenie DELETE)

11. Usuń zespół na którym nie ma zatrudnionych żadnych pracowników.
12. Usuń pracowników, którym nie płaci się pensji dodatkowej, i pracują na etacie 'ASYSTENT'.
13. Usuń wszystkich pracowników, którzy nie są szefami.
14. Usuń pracowników, których pensja mieści się w widełkach dwóch różnych etatów i pracują w zespole 'SYSTEMY ROZPROSZONE'.

Slajd pokazuje rozwiązania zadań (4) i (5), których treść przytoczono poniżej.

(4) Znajdź największą wartość atrybutu ID_ZESP (ID_ZESP jest kluczem głównym) w tabeli ZESPOLY i wstaw do tabeli ZESPOLY nowy zespół o nazwie „SYSTEMY BAZ DANYCH” i wartości ID_ZESP większej od odczytanej poprzednio wartości. Adresu nie podawaj. Całość wykonaj za pomocą jednego polecenia (wykorzystaj podzapytanie).

(5) Wstaw etat „STUDENT”. Jako płacę minimalną i maksymalną podaj odpowiednio 0 i 330 złotych.

Slajd pokazuje rozwiązanie zadania (6), których treść przytoczono poniżej.

(6) Zatrudnij się na etacie „STUDENT” w zespole „SYSTEMY BAZ DANYCH” podając jako datę zatrudnienia aktualną datę systemową, płacę podstawową ustaw na 330 złotych, a dodatkową na 5% średniej płacy w zespole ADMINISTRACJA. Jako ID_PRAC podaj największą wartość ID_PRAC w tabeli PRACOWNICY zwiększoną o 10. Całość wykonaj za pomocą jednego polecenia (wykorzystaj podzapytania).

Slajd pokazuje rozwiązania zadań (7) i (8), których treść przytoczono poniżej.

(7) Podnieś do średniej pracowniczej pensję najgorzej zarabiającym pracownikom.

(8) Uaktualnij płace dodatkowe pracowników zespołu 20. Nowe płace dodatkowe mają być równe średniej płacy pracowników, których przełożonym jest Marecki.

Slajd pokazuje rozwiązania zadań (9) i (10), których treść przytoczono poniżej.

(9) Pracownikom zespołu o nazwie 'SYSTEMY ROZPROSZONE' daj 25% podwyżkę.

(10) Zmniejsz dolną granicę widełek (PLACA_OD) o 10%, a górną granicę widełek (PLACA_DO) zwiększ o 20% dla etatów, na których sumaryczna płaca jest największa pośród wszystkich etatów.

Slajd pokazuje rozwiązania zadań (11),(12) i (13), których treść przytoczono poniżej.

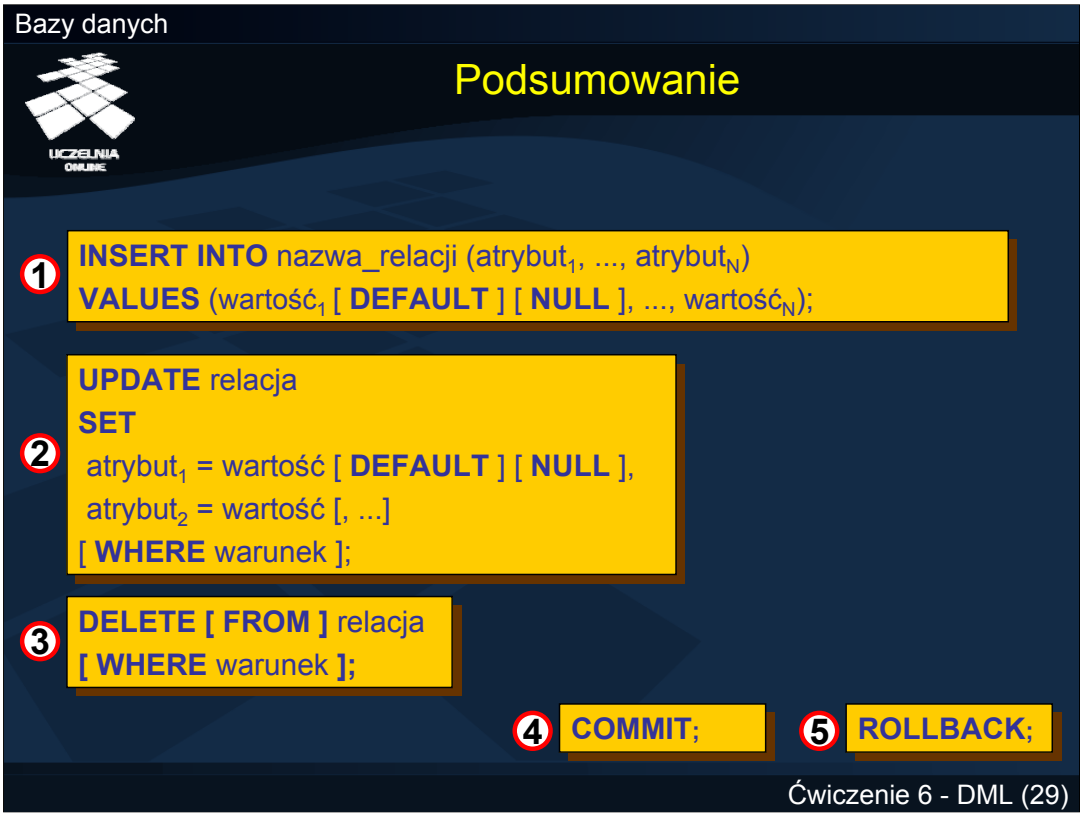
(11) Usuń zespół na którym nie ma zatrudnionych żadnych pracowników.

(12) Usuń pracowników, którym nie płaci się pensji dodatkowej, i pracują na etacie 'ASYSTENT'.

(13) Usuń wszystkich pracowników, którzy nie są szefami.

Slajd pokazuje rozwiązanie zadania (14), którego treść przytoczono poniżej.

(14) Usuń pracowników, których pensja mieści się w widełkach dwóch różnych etatów i pracują w zespole 'SYSTEMY ROZPROSZONE'.



Na ćwiczeniu poznaliście państwo polecenia INSERT (1), UPDATE(2) i DELETE (3), które służą odpowiednio do wstawiania, modyfikacji i usuwania krotek z relacji. Poznaliście również polecenia COMMIT (4) i ROLLBACK (5), które pozwalają na zatwierdzenie, bądź wycofanie zmian z relacji. Działanie tych poleceń przećwiczyliście państwo na zadaniach do samodzielnego wykonania.