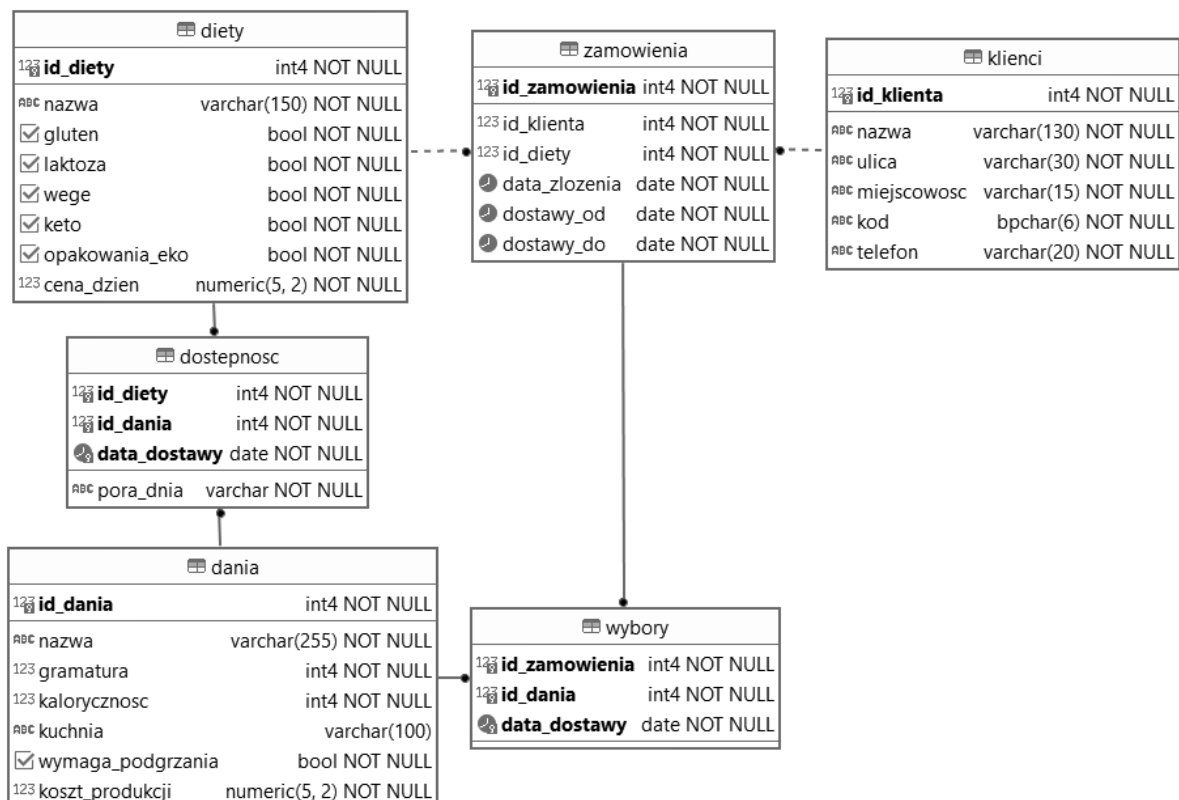


Kolokwium II grupa A

W odpowiedziach do zadań nie trzeba używać nazwy schematu - przyjmuje się, że *search_path* jest prawidłowo ustawiony. Nie ma wymogu sortowania wyników.

- (5p.) Sieć sklepów projektuje nowy system do inwentaryzacji. Narysuj diagram związków encji według poniższego opisu. Pogrubione słowa oznaczają encje, które powinny znaleźć się na diagramie.
 - Każdy **sklep** posiada co najmniej 10 **towarów**.
 - Każdy **towar** może mieć (lub nie mieć) zgłoszone **zapotrzebowanie**.
 - Na każdy towar może być dokonana dowolna liczba **rezerwacji**.
 - W każdym sklepie pracuje co najmniej 5 **pracowników**. Hierarchia pracowników może mieć strukturę drzewa (np. jeden pracownik może podlegać innemu pracownikowi).
- (5p.) Dana jest relacja R o schemacie $H = \{A, B, C, D\}$ oraz zbiór zależności funkcyjnych $F = \{ \{C, D\} \rightarrow A, A \rightarrow B, B \rightarrow C \}$.
Sprawdź czy dekompozycja:
 $H_1 = \{A, C, D\}, H_2 = \{A, B, C\}$ zachowuje zależności funkcyjne. Uzasadnij odpowiedź.
- (5p.) Napisz zapytania SQL tworzące w bazie tabele *diety* i *dostepnosc* (patrz: załączony schemat). Zadbaj o utworzenie właściwych kluczy głównych i kluczy obcych (mogą być częścią zapytań CREATE lub stanowić odrębne zapytania typu ALTER). Dodatkowo nałóż ograniczenia na kolumny, aby nazwa diety miała co najmniej 5 znaków oraz pora dnia mogła przyjmować tylko jedną z wartości: 'śniadanie', 'drugie śniadanie', 'obiad', 'podwieczorek', 'kolacja'.
- (5p.) Korzystając z operatorów *any* oraz *all* (obu) napisz zapytanie SQL pobierające z bazy ID wszystkich dań, które wybrano co najmniej raz w roku 2023, a które jednocześnie nigdy nie były dostępne na kolację. Nie używaj złączeń JOIN.
- (5p.) Napisz funkcję o nazwie *wybierz_najciezszy_posilek*, która wybierze posiłek o najwyższej gramaturze (doda wiersz do tabeli *wybory*) dla wybranego zamówienia, daty dostawy oraz pory dnia. Funkcja przyjmuje 3 argumenty: *id_zamowienia* (int), *data_dostawy* (date), *pora_dnia* (varchar).
Możesz przyjąć, że istnieje tylko jeden posiłek spełniający kryteria. Jeżeli wybrano już wcześniej posiłek to funkcja aktualizuje id dania w istniejącym rekordzie. Funkcja zwraca id wybranego posiłku (int).



Kolokwium II grupa B

W odpowiedziach do zadań nie trzeba używać nazwy schematu - przyjmuje się, że *search_path* jest prawidłowo ustawiony. Nie ma wymogu sortowania wyników.

- (5p.) Sieć sklepów projektuje nowy system do inwentaryzacji. Narysuj diagram związków encji według poniższego opisu. Pogrubione słowa oznaczają encje, które powinny znaleźć się na diagramie.
 - Każdy **sklep** posiada co najmniej 100 **towarów**.
 - Każdy **towar** może mieć (lub nie mieć) tabelę **wartości odżywczych**.
 - Na każdy towar może być dokonana dowolna liczba **rezerwacji**.
 - Każdy towar należy do **kategorii**. Hierarchia kategorii może mieć strukturę drzewa (np. jedna kategoria może być podkategorią innej).
- (5p.) Dana jest relacja R o schemacie $H = \{A, B, C, D\}$ oraz zbiór zależności funkcyjnych $F = \{ \{C, D\} \rightarrow A, A \rightarrow B, B \rightarrow C \}$
Sprawdź czy dekompozycja:
 $H_1 = \{A, C, D\}, H_2 = \{B, C\}$ zachowuje zależności funkcyjne. Uzasadnij odpowiedź.
- (5p.) Napisz zapytania SQL tworzące w bazie tabele *dania* i *dostepnosc* (patrz: załączony schemat). Zadbaj o utworzenie właściwych kluczy głównych i kluczy obcych (mogą być częścią zapytań CREATE lub stanowić odrębne zapytania typu ALTER). Dodatkowo nałóż ograniczenia na kolumny, aby nazwa *dania* miała co najmniej 5 znaków oraz pora dnia mogła przyjmować tylko jedną z wartości: 'śniadanie', 'drugie śniadanie', 'obiad', 'podwieczorek', 'kolacja'.
- (5p.) Korzystając z operatorów *any* oraz *all* (obu) napisz zapytanie SQL pobierające z bazy ID wszystkich dań, które co najmniej raz były dostępne na kolację, a które jednocześnie ani razu nie zostały wybrane w roku 2023. Nie używaj złączeń JOIN.
- (5p.) Napisz funkcję o nazwie *wybierz_najciezszy_posilek*, która wybierze posiłek o najwyższej gramaturze (doda wiersz do tabeli *wybory*) dla wybranego zamówienia, daty dostawy oraz pory dnia. Funkcja przyjmuje 3 argumenty: *id_zamowienia* (int), *data_dostawy* (date), *pora_dnia* (varchar).
Możesz przyjąć, że istnieje tylko jeden posiłek spełniający kryteria. Jeżeli wybrano już wcześniej posiłek to funkcja aktualizuje id dania w istniejącym rekordzie. Funkcja zwraca id wybranego posiłku (int).

