**1**Zaprojektuj widok, który zwraca wartości z kolumn `customerNumber`, `customerName`, `creditLimit` tabeli `customers`. Zdefiniuj widok tak, aby w wynikach pojawiły dane klientów o limicie kredytowym creditLimit > 100000 oraz numer klienta wyświetlał się w kolumnie o nazwie `id`.

Definicja widoku powinna się poprawnie wykonać bez względu na to, czy widok o takiej samej nazwie już istnieje lub jeszcze nie istnieje w bazie danych.

Jako odpowiedź wstaw opracowaną definicję widoku.

CREATE OR REPLACE VIEW wysoki\_limit\_kredytowy AS SELECT customerNumber AS id, customerName, creditLimit FROM customers WHERE creditLimit > 100000

**2**Stwórz definicję widoku o nazwie `luckynumbers`, który zwraca trzy wartości liczbowe 7, 17, 48 w kolumnie o nazwie `number`.

CREATE VIEW luckynumbers AS SELECT 7 AS number UNION ALL SELECT 17 AS number UNION ALL SELECT 48 AS number UNION ALL

UNION ALL -> UNION

**3**Przeanalizuj warunki WHERE dla trzech typów zapytań zilustrowanych poniższymi przykładami:

SELECT \* FROM `employees` WHERE firstName='Yoshimi' and lastName='Kato'

SELECT \* FROM `employees` WHERE lastName='Kato' and firstName='Yoshimi'

SELECT \* FROM `employees` WHERE firstName='Yoshimi'

Zaproponuj definicję dodania indeksu do tabeli `employees`, która usprawni wyszukiwanie w każdym z tych przypadków.

CREATE INDEX employees\_name ON employees(lastName, firstName)

CREATE INDEX employees\_name -> ALTER TABLE employees

**4**Zmodyfikuj tabelę `offices`, tak aby niemożliwe było dodanie kolejnego biura w tym samym regionie (state) danego kraju (country).

Jako odpowiedź wpisz zapytanie modyfikujące tabelę.

ALTER TABLE offices ADD CONSTRAINT unikalny\_region\_na\_kraj UNIQUE(state, country)