

PRZYKŁADY

Przedstawione przykłady dotyczą następującej bazy danych:

D(D# , NAZWISKO, STATUS, D_MIASTO)
-dostawcy

C(C# , NAZWA, KOLOR, MASA, C_MIASTO) -
części

DC(D#, C#, ILOŚĆ) - dostawy

PRZYKŁAD 1

Podać nazwiska dostawców dostarczających wszystkie części.

Sformułujmy to zapytanie inaczej:

Podać nazwiska dostawców, dla których nie istnieje część, której by oni nie dostarczali

PRZYKŁAD 1

SELECT NAZWISKO

FROM D

WHERE NOT EXISTS

(SELECT *

FROM C

WHERE NOT EXISTS

(SELECT *

FROM DC

WHERE D# = D.D# AND C# = C.C#))

PRZYKŁAD 1

Najbardziej wewnętrzny blok określa dostawę identyfikowaną przez atrybuty pochodzące z bloków zewnętrznych. W poszczególnych blokach wykorzystywano różne relacje. Zastosowanie synonimów nie było więc potrzebne.

PRZYKŁAD 2

Podać numery dostawców dostarczających wszystkie części.

Możemy zastosować konstrukcję podobną do rozwiązania z przykładu 1.

PRZYKŁAD 2

```
SELECT D#  
FROM D  
WHERE NOT EXISTS  
  ( SELECT *  
    FROM C  
    WHERE NOT EXISTS  
      (SELECT *  
        FROM DC  
        WHERE D# = D.D# AND C# = C.C#))
```

PRZYKŁAD 2 – inne rozwiązanie

Można zauważyć, że do uzyskania odpowiedzi relacja D nie jest potrzebna. Atrybut D# możemy uzyskać z relacji DC.

PRZYKŁAD 2 – inne rozwiązanie

```
SELECT DISTINCT D#
```

```
FROM DC X
```

```
WHERE NOT EXISTS
```

```
  ( SELECT *
```

```
    FROM C
```

```
    WHERE NOT EXISTS
```

```
      (SELECT *
```

```
        FROM DC
```

```
        WHERE D# = X.D# AND C# = C.C#));
```


PRZYKŁAD 2 – inne rozwiązanie

Najbardziej wewnętrzny blok określa dostawę identyfikowaną przez dostawcę występującego w dostawie z bloku najbardziej zewnętrznego oraz przez część z bloku pośredniego. DISTINCT jest potrzebne, ponieważ dostawca może występować wielokrotnie w relacji DC.

W przykładach 3 i 4 zastosowano podobną metodologię.

PRZYKŁAD 3

Podać numery dostawców wszystkich części czerwonych.

```
SELECT DISTINCT D#  
FROM DC X  
WHERE NOT EXISTS  
  ( SELECT * FROM C  
    WHERE KOLOR = 'Red' AND NOT EXISTS  
      (SELECT * FROM DC  
       WHERE D# = X.D# AND C# = C.C#));
```

PRZYKŁAD 4

Podać numery dostawców wszystkich części dostarczanych przez D2.

```
SELECT DISTINCT D# FROM DC X
WHERE NOT EXISTS
  ( SELECT * FROM C WHERE C# IN
    (SELECT C# FROM DC
     WHERE D# = 'D2') AND NOT EXISTS
      (SELECT * FROM DC
       WHERE D# = X.D# AND C# = C.C#));
```

PRZYKŁAD 5

Podać numery części dostarczanych przez wszystkich dostawców z Londynu

```
SELECT C# FROM C
WHERE NOT EXISTS
  (SELECT * FROM D
   WHERE MIASTO = 'LONDYN'
   AND NOT EXISTS
     (SELECT * FROM DC
      WHERE D# = D.D# AND C# = C.C#));
```

ASERCJE

ASERCJE

Ogólne ograniczenia integralnościowe definiuje się za pomocą asercji. Ograniczenia takie dotyczą najczęściej więcej niż jednej relacji.

Asercje definiuje się za pomocą instrukcji

`CREATE ASSERTION nazwa CHECK (P),`

gdzie P oznacza wyrażenie logiczne opisujące warunek asercji. Warunek ten musi być spełniony. Instrukcje, które go naruszają, nie są wykonywane.

PRZYKŁAD 1

Każda część waży przynajmniej 1 kg

```
CREATE ASSERTION A1  
CHECK ((SELECT MIN(MASA)  
        FROM C) >1)
```

PRZYKŁAD 2

Status dostawcy jest liczbą dodatnią

Zastosujemy predykat NOT EXISTS.

Predykat ten jest dość często stosowany przy definiowaniu asercji.

Zgodnie z poleceniem nie może istnieć dostawca, którego status nie jest większy niż 0.

Takiemu sformułowaniu warunku odpowiada następująca definicja:

PRZYKŁAD 2

Status dostawcy jest liczbą dodatnią

```
CREATE ASSERTION A2  
CHECK (NOT EXISTS  
      (SELECT *  
       FROM D  
       WHERE NOT (STATUS>0)))
```

PRZYKŁAD 3

Całkowita dostawa żadnej części nie może przekraczać 100 egzemplarzy

Zgodnie z poleceniem nie może istnieć część, dla której suma dostaw jest większa niż 100.

PRZYKŁAD 3

```
CREATE ASSERTION suma_dostaw
CHECK (NOT EXISTS
      (SELECT *
        FROM C X
         WHERE (SELECT SUM(ILOSC)
                 FROM DC
                 WHERE C# = X.C# ) > 100));
```

PRZYKŁAD 4

Wszyscy dostawcy ze statusem ≥ 100 mieszkają w Londynie

(Nie istnieje dostawca ze statusem ≥ 100 , który nie mieszka w Londynie)

```
CREATE ASSERTION A4
CHECK (NOT EXISTS
      (SELECT *
        FROM D
        WHERE STATUS  $\geq$  100 AND
          MIASTO  $\neq$  'LONDYN'));
```

PRZYKŁAD 5

Masa dostawy nie może przekraczać 10 000

Masa dostawy jest równa iloczynowi masy jednostkowej części (relacja C – atrybut MASA) oraz wielkości dostawy (relacja DC – atrybut ILOSC).

Należy połączyć obydwie relacje i wyeliminować krotki, w których $MASA * ILOSC > 10\ 000$.

Dostawa z takimi wartościami atrybutów nie może istnieć.

PRZYKŁAD 5

```
CREATE ASSERTION A5
CHECK (NOT EXISTS
      (SELECT *
       FROM C, DC
       WHERE DC.C# = C.C# AND
            (MASA * ILOSC) > 10 000));
```

PRZYKŁAD 6

Dostawcy z Londynu muszą dostarczać części w ilości większej niż 100

Mogą istnieć dostawcy z Londynu, którzy nie dostarczają żadnej części. Jeżeli jednak coś dostarczają, to wielkość dostawy musi przekraczać 100.

Zatem nie istnieje dostawca z Londynu, który dostarcza coś w ilości mniejszej niż 100.

PRZYKŁAD 6

Dostawcy z Londynu muszą dostarczać części w ilości większej niż 100

```
CREATE ASSERTION A6
```

```
CHECK (NOT EXISTS
```

```
    (SELECT *
```

```
      FROM D, DC
```

```
      WHERE MIASTO = 'LONDYN'
```

```
      AND DC.D# = D.D# AND
```

```
      ILOSC <= 100));
```


PRZYKŁAD 7

Ten przykład dotyczy następującej bazy:

LEKI(L#, PRODUCENT#, OPIS_L) - Leki

FARM(P#, MIASTO, OPIS_P) – producenci leków

Warunek integralności:

Nie akceptujemy leków z Londynu

(nie może być w relacji LEKI leków produkowanych przez producentów z Londynu)

PRZYKŁAD 7

CREATE ASSERTION A7

CHECK (NOT EXISTS

(SELECT *

FROM LEKI, FARM

WHERE PRODUCENT# = P# AND

MIASTO = 'LONDYN'));