

## Sprawdzian z SQL nr 1

21.04.2022

Dla każdego z poniższych zadań napisz odpowiednie polecenia SQL. Oczekujemy rozwiązania w postaci **treści** poleceń SQL, a nie znalezionej odpowiedzi. Nie będą sprawdzane jakiegokolwiek zapytania niepoprawne składniowo — sprawdź swoje rozwiązanie używając `\i plik.sql` ! Plik możesz wysyłać wielokrotnie, sprawdzana będzie wyłącznie najnowsza wersja.

Wczytaj do swojej bazy danych plik `tpch-keys.sql`. Rozwiązania wysyłaj przez formularz pod adresem `http://hera.stud.ii:2015`. Rób to jak najczęściej! Wszystkie dane na Waszych komputerach są kasowane po restarcie.

Format pierwszej linijki rozwiązania: `-- grupa-imie-nazwisko`, gdzie grupa to inicjały prowadzącego Twoją grupę (`pwi/jotop/plg/mabi/akr/rfe`), np. `pwi-Jan-Kowalski`. Wymagany format całego pliku z rozwiązaniem:

```
-- grupa-imie-nazwisko
-- Zadanie 1
<zapytanie>

-- Zadanie 2
<zapytanie>
...
```

**Zadanie 1** Wypisz numery zamówień (`orderkey`) wraz z datą złożenia zamówienia (`orderdate`) oraz jego priorytetem (`orderpriority`), dla których któraś pozycja (`lineitem`) była zrealizowana (`shipdate`) później niż 120 dni po złożeniu zamówienia. Uwzględnij wyłącznie zamówienia złożone w sierpniu dowolnego roku mające priorytet `'2-HIGH'` lub `'1-URGENT'`. Wyniki posortuj wg priorytetu (najpierw `'1-URGENT'`), w drugiej kolejności malejąco wg daty złożenia zamówienia, a w trzeciej rosnąco wg numeru zamówienia.

### Rozwiązanie

```
SELECT DISTINCT o_orderkey, o_orderdate, o_orderpriority
FROM orders
JOIN lineitem ON l_orderkey = o_orderkey
WHERE l_shipdate > o_orderdate + 120
AND o_orderpriority < '3'
AND EXTRACT (month FROM o_orderdate) = 8
ORDER BY o_orderpriority ASC, o_orderdate DESC, o_orderkey ASC;
```

*Zapytanie wzorcowe zwraca 139 krotek.*

**Zadanie 2** Wypisz numery zamówień (`orderkey`) składające się z więcej niż 1 pozycji (`lineitem`), których wszystkie pozycje były dostarczane w ten sam sposób (`shipmode`). Wyniki posortuj rosnąco.

## Rozwiązanie

```
(
SELECT DISTINCT l1.l_orderkey
  FROM lineitem l1
  JOIN lineitem l2 ON l1.l_orderkey = l2.l_orderkey
  WHERE l1.l_linenumber <> l2.l_linenumber
)
EXCEPT
(
SELECT l1.l_orderkey
  FROM lineitem l1
  JOIN lineitem l2 on l1.l_orderkey=l2.l_orderkey
  WHERE l1.l_shipmode <> l2.l_shipmode
)
ORDER BY 1;
```

Zapytanie wzorcowe zwraca 3623 numery, używa **EXCEPT** oraz dwóch prostych zapytań **SELECT**.

- Zadanie 3**
- Dodaj więz klucza obcego dla kolumny **regionkey** tabeli **nation** odwołujący się do kolumny **regionkey** w tabeli **region**, przy czym ustaw **ON DELETE** w taki sposób, aby usunięcie kontynentu powodowało usunięcie wszystkich jego krajów.
  - Do obu powyższych tabel dodaj po jednej krotce reprezentującej Australię odpowiednio jako kraj oraz jako kontynent.
  - Usuń nowo dodany kontynent z tabeli **region**. Sprawdź zapytaniem **SELECT**, czy zniknęła odpowiednia krotka z tabeli **nation**.

## Rozwiązanie

```
ALTER TABLE nation
  ADD CONSTRAINT fk_nation_region
  FOREIGN KEY (n_regionkey) REFERENCES region(r_regionkey)
  ON DELETE CASCADE;

INSERT INTO region VALUES (666, 'AUSTRALIA', NULL);
INSERT INTO nation VALUES (2137, 'AUSTRALIA', 666, NULL);

DELETE FROM region WHERE r_name = 'AUSTRALIA';
SELECT * FROM nation WHERE n_name = 'AUSTRALIA';
```

- Zadanie 4** Wypisz numery wszystkich klientów (**custkey**) z tabeli **customer**, którzy nie złożyli żadnego zamówienia (**orders**). Wyniki posortuj rosnąco.

## Rozwiązanie

```
SELECT c_custkey
  FROM customer
 LEFT JOIN orders ON c_custkey = o_custkey
 WHERE o_orderkey IS NULL
 ORDER BY 1;
```

*Zapytanie wzorcowe zwraca 5000 numerów.*

**Zadanie 5** Dopisz do salda kont (acctbal) wszystkich europejskich klientów z tabeli customer bonus w wysokości 10 procent obecnego salda.

## Rozwiązanie

```
UPDATE customer
  SET c_acctbal = c_acctbal * 1.1
  FROM nation
 JOIN region ON n_regionkey = r_regionkey
 WHERE c_nationkey = n_nationkey
 AND r_name = 'EUROPE';
```

*Zapytanie wzorcowe zmienia saldo 2968 krotkom.*