

Rozwiązania zadań do kursu stacjonarnego "Podstawy SQL" na platformie LNU.
Copyright 2022 Soft Power sp. z o.o.

5.1 Kolumna rating w tabeli movies zawiera oceny filmów. Wyświetl najwyższą ocenę występującą w tabeli.

```
SELECT MAX(rating) FROM movies;
```

5.2 Kolumna year w tabeli movies zawiera lata produkcji filmów. Wyświetl rok produkcji najstarszego filmu.

```
SELECT MIN(year) FROM movies;
```

5.3 Kolumna price w tabeli cell_phones zawiera ceny telefonów, zaś kolumna brand nazwy producentów.

Wyświetl najwyższą cenę, spośród telefonów marek: Samsung, Sony i Nokia.

```
SELECT MAX(price) FROM cell_phones WHERE brand IN ('Samsung', 'Sony', 'Nokia');
```

5.4 Kolumna cast w tabeli movies zawiera nazwiska aktorów, odtwarzających główne role. Wyświetl rok produkcji najstarszego filmu, w którym wystąpił Tom Cruise.

```
SELECT MIN(year) FROM movies WHERE [cast] LIKE '%Tom Cruise%';
```

5.5 Wyświetl najwyższą ocenę filmu z tabeli movies, w którym nie wystąpił ani Tom Cruise, ani Clint Eastwood.

```
SELECT MAX(rating) FROM movies WHERE [cast] NOT LIKE '%Tom Cruise%' AND [cast] NOT LIKE '%Clint Eastwood%';
```

5.6 Wyświetl liczbę określającą ile filmów z gatunku dramat (drama) zawiera tabela movies. Jako argumentu funkcji COUNT użyj kolumny id.

```
SELECT COUNT(id) FROM movies WHERE type = 'drama';
```

5.7 Z tabeli movies wyświetl liczbę filmów, który zostały wyprodukowane od roku 1980 do roku 2000. Jako argumentu funkcji COUNT użyj kolumny id.

```
SELECT COUNT(id) FROM movies WHERE year BETWEEN 1980 AND 2000;
```

5.8 Reżyserem ilu filmów z tabeli movies był Ridley Scott lub jego brat Tony Scott? Pobierz tę liczbę, jako argumentu funkcji COUNT używając pola id.

```
SELECT count(id) FROM movies WHERE director IN ("Tony Scott", "Ridley Scott");
```

5.9 Z tabeli cell_phones wyświetl liczbę telefonów, których nazwa producenta (brand) zaczyna się na literę 'S'. Pobierz tę liczbę, jako argumentu funkcji COUNT używając pola id.

```
SELECT count(id) FROM cell_phones WHERE brand LIKE 'S%';
```

5.10 Ilu różnych producentów zawiera kolumna brand w tabeli cell_phones? Wyświetl tę liczbę. Wynikowej kolumnie nadaj nazwę producenci.

```
SELECT COUNT(DISTINCT brand) AS producenci FROM cell_phones;  
SELECT COUNT(*) AS producenci FROM (SELECT DISTINCT brand FROM cell_phones);
```

5.11 Oblicz średnią ocen (rating) wszystkich filmów z tabeli movies.

```
SELECT AVG(rating) FROM movies;
```

5.12 Z tabeli movies oblicz średnią ocen z filmów, w których zagrał Brad Pitt.

```
SELECT AVG(rating) FROM movies WHERE [cast] LIKE "%Brad Pitt%";
```

5.13 Z tabeli cell_phones oblicz średnią cenę telefonów marki Apple.

```
SELECT AVG(price) FROM cell_phones WHERE brand = 'Apple';
```

5.14 Z tabeli cell_phones oblicz średnią cenę z pięciu telefonów wyposażonych w największe ekrany.

```
SELECT AVG(price) FROM (SELECT * FROM cell_phones ORDER BY screen DESC LIMIT 5);
```

5.15 Z tabeli movies wyświetl średnią ocenę filmów wyprodukowanych od roku 1980 do roku 1990.

```
SELECT AVG(rating) FROM movies WHERE year BETWEEN 1980 AND 1990;
```

6.1 Kolumna fine w tabeli tickets zawiera wartości wystawionych mandatów. Oblicz średnią wysokość mandatów wystawionych w Chicago. Wynikową kolumnę nazwij Chicago

```
SELECT AVG(fine) AS Chicago FROM tickets WHERE location = 'Chicago';
```

6.2 Z tabeli movies wyświetl dane tych filmów, których ocena jest większa, niż średnia z ocen wszystkich filmów z tabeli.

```
SELECT * FROM movies WHERE rating > (SELECT AVG(rating) FROM movies);
```

6.3 Kolumna fine w tabeli tickets zawiera wysokości wystawionych mandatów. Wyświetl sumę, na jaką zostały wystawione wszystkie mandaty w Chicago.

```
SELECT SUM(fine) FROM tickets WHERE location = 'Chicago';
```

6.4 Kolumna id w tabeli drivers oraz kolumna driver_id w tabeli tickets zawierają numery identyfikacyjne kierowców. Z tabeli tickets oblicz sumę mandatów wystawionych na kierowcę o imieniu (first_name) Fletcher i nazwisku (family_name) Reede. Wynikową kolumnę nazwij Fletcher Reede.

```
SELECT SUM(fine) AS [Fletcher Reede] FROM tickets WHERE driver_id = (SELECT id FROM drivers WHERE first_name = 'Fletcher' AND family_name = 'Reede');
```

6.5 Oblicz średnią ocenę tych filmów z tabeli movies, których numer id znajduje się także w tabeli my_movies. Wynikową kolumnę nazwij My movies avg

```
SELECT AVG(rating) AS [My movies avg] FROM movies WHERE id IN (SELECT id FROM my_movies);
```

6.6 Kolumna type w tabeli movies zawiera dane dotyczące kategorii filmów. Wyświetl liczbę filmów należących do każdej z kategorii. Wynikowa tabela powinna zawierać tylko dwie kolumny: kolumnę type z nazwami kategorii oraz kolumnę o nazwie amount, która będzie zawierać liczbę filmów z danej kategorii.

```
SELECT type, COUNT(1) AS amount FROM movies GROUP BY type;
```

6.7 Kolumna brand w tabeli cell_phones zawiera nazwy producentów telefonów. Wyświetl liczbę czarnych (black) telefonów należących do każdego producenta. Wynikowa tabela powinna zawierać tylko dwie kolumny: kolumnę brand z nazwą producenta oraz kolumnę o nazwie amount, która będzie zawierać liczbę telefonów.

```
SELECT brand, COUNT(1) AS amount FROM cell_phones WHERE color = 'black' GROUP BY brand;
```

6.8 Kolumna brand w tabeli cell_phones zawiera nazwy producentów telefonów. Wyświetl średnią cenę telefonów należących do każdego producenta. Wynikowa tabela powinna zawierać tylko dwie kolumny: kolumnę brand z nazwą producenta oraz kolumnę o nazwie avg price, która będzie zawierać średnią cenę telefonów.

```
SELECT brand, AVG(price) AS [avg price] FROM cell_phones GROUP BY brand;
```

6.9 Kolumna color w tabeli cell_phones zawiera kolory telefonów. Wyświetl liczbę telefonów w każdym z kolorów. Wynikowa tabela powinna zawierać tylko dwie kolumny: kolumnę color oraz kolumnę o nazwie amount, która będzie zawierać liczbę telefonów w danym kolorze. Posortuj wyniki malejąco.

```
SELECT color, COUNT(1) AS amount FROM cell_phones GROUP BY color ORDER BY amount DESC;
```

6.10 Kolumna fine w tabeli tickets zawiera wysokości mandatów. Wyświetl sumy na jakie zostały wystawione mandaty w poszczególnych miastach. Wynikowa tabela powinna zawierać tylko dwie kolumny: kolumnę location oraz kolumnę o nazwie sum, która będzie zawierać sumaryczną wartość mandatów wystawionych w danym mieście. Wyświetl dane jedynie dla trzech miast, w których wystawiono mandaty na najwyższą sumę.

```
SELECT location, SUM(fine) AS sum FROM tickets GROUP BY location ORDER BY sum DESC LIMIT 3;
```

6.11 Kolumna driver_id w tabeli tickets zawiera numer identyfikacyjny kierowcy. Wyświetl kolumnę driver_id oraz kolumnę o nazwie amount zawierającą liczbę

mandatów, które otrzymał dany kierowca. Ogranicz liczbę wyświetlanych rekordów do czterech kierowców, którzy otrzymali największą liczbę mandatów.

```
SELECT driver_id, COUNT(1) AS amount FROM tickets GROUP BY driver_id ORDER BY amount DESC LIMIT 4;
```

6.12 Kolumna price w tabeli cell_phones zawiera ceny telefonów, zaś kolumna brand nazwy producentów.

Wyświetl dane najtańszego modelu telefonu dla każdego producenta. Tabela powinna zawierać trzy kolumny: brand, model oraz kolumnę o nazwie price zawierającą cenę najtańszego modelu danego producenta.

Posortuj wyniki od modelu najtańszego do najdroższego.

```
SELECT brand, model, MIN(price) AS price FROM cell_phones GROUP BY brand ORDER BY price;
```

6.13 Kolumna brand w tabeli cell_phones zawiera nazwy producentów telefonów.

Wyświetl nazwę producenta, którego modele telefonów są najliczniejsze w tabeli. Wynikowa tabela powinna zawierać tylko kolumnę brand.

```
SELECT brand FROM cell_phones GROUP BY brand ORDER BY COUNT(1) DESC LIMIT 1;
```

6.14 Z tabeli cell_phones wyświetl średnie ceny telefonów poszczególnych producentów, ale tylko te, które są nie mniejsze niż 500\$. Tabela powinna zawierać kolumny: brand oraz avg price

```
SELECT brand, AVG(price) AS [avg price] FROM cell_phones GROUP BY brand HAVING [avg price] >= 500;
```

6.15 Kolumna driver_id w tabeli tickets zawiera numer identyfikacyjny kierowcy, zaś kolumna fine wysokość mandatu. Wyświetl sumę wysokości mandatów dla poszczególnych kierowców, ale tylko dla tych, którzy otrzymali przynajmniej trzy mandaty. Wynikowa tabela powinna składać się z dwóch kolumn: driver_id oraz kolumny o nazwie sum zawierającej sumę mandatów dla danego kierowcy. Wyniki posortuj od największej do najmniejszej sumy mandatów.

```
SELECT driver_id, SUM(fine) AS sum FROM tickets GROUP BY driver_id HAVING COUNT(1) >= 3 ORDER BY sum DESC;
```

7.1 Z tabeli tickets wyświetl w porządku malejącym sumy wartości mandatów dla tych miast, w których wystawiono ich co najmniej trzy. Weź pod uwagę jedynie mandaty wyższe niż 30\$. Wynikowa tabela powinna zawierać dwie kolumny: location oraz kolumnę o nazwie sum zawierającą sumy na jakie zostały wystawione mandaty w danych miastach.

```
SELECT location, SUM(fine) AS sum FROM tickets WHERE fine > 30 GROUP BY location HAVING COUNT(1) >= 3 ORDER BY sum DESC;
```

7.2 Z tabeli movies wyświetl reżysera (director), tytuł (title), rok produkcji (year), oraz ocenę (nazwij tę kolumnę rating) najwyższej ocenianego filmu dla każdego reżysera, biorąc pod uwagę jedynie filmy wyprodukowane po roku 1989. Uszereguj wyniki malejąco po kolumnie rating.

```
SELECT director, title, year, MAX(rating) AS rating FROM movies WHERE year >= 1990 GROUP BY director ORDER BY rating DESC;
```

7.3 Kolumna price w tabeli cell_phones zawiera ceny netto telefonów. Wyświetl kolumny: model, price oraz kolumnę o nazwie plus tax, która będzie zawierać ceny brutto modeli. Podatek wynosi 23%.

* zaokrąglaj wynik z podatkiem do dwóch miejsc po przecinku

```
SELECT model, price, price*1.23 AS [plus tax] FROM cell_phones;  
SELECT model, price, round( price*1.23, 2) AS [plus tax] FROM cell_phones;
```

7.4 Od dzisiaj do odwołania obowiązuje promocja na telefony - wszystkie są tańsze o 20\$! Z tabeli cell_phones wyświetl dane telefonów z promocyjną ceną. Tabela ma zawierać kolumny: brand, model, price oraz kolumnę o nazwie sale! zawierającą promocyjną cenę telefonu.

```
SELECT brand, model, price, price - 20 AS [sale!] FROM cell_phones;
```

7.5 Z tabeli movies wyświetl rekordy w których wartość pola rating jest co najmniej o 10% większa niż średnia dla tego pola z całej tabeli.

```
SELECT * FROM movies WHERE rating >= 1.1 * ( SELECT AVG(rating) FROM movies );
```

7.6 Korzystając z danych z tabeli samples, wyświetl kolumny: id, kolumnę o nazwie sum zawierającą sumę wartości pól test_1, test_2 oraz test_3 oraz kolumnę group.

* Zaokrąglij sumy do dwóch miejsc po przecinku

```
SELECT id, test_1 + test_2 + test_3 AS sum, [group] FROM samples;  
SELECT id, round(test_1 + test_2 + test_3, 2) AS sum, [group] FROM samples;
```

7.7 Rekordy z tabeli samples w polu group zawierają jedną z trzech wartości: 'A', 'B' lub 'C'. Dla każdej z tych trzech grup rekordów wyświetl najmniejszą wartość z iloczynu sumy pól test_1, test_2, test_3 i wartości pola factor. Wynikowa tabela powinna zawierać kolumny: group, id oraz kolumnę o nazwie value będącą iloczynem sumy pól test_1, test_2, test_3 i wartości pola factor.

* Zaokrąglij sumy do czterech miejsc po przecinku

```
SELECT [group], id, MIN( (test_1 + test_2 + test_3 ) * factor ) FROM samples  
GROUP BY [group];  
SELECT [group], id, round(MIN( (test_1 + test_2 + test_3 ) * factor ), 4) AS  
value FROM samples GROUP BY [group];
```

7.8 Pogrupuj rekordy z tabeli samples według wartości z kolumny group i dla każdej z grup oblicz liczbę rekordów o nieparzystej wartości pola id. Wyświetl dane tylko dla grup, w których liczba rekordów spełniająca to kryterium jest większa od 1. Wynikowa tabela powinna zawierać kolumny: group oraz kolumnę o nazwie amount z liczbą rekordów dla danej grupy spełniającej podane kryteria.

```
SELECT [group], COUNT(1) AS amount FROM samples WHERE id % 2 <> 0 GROUP BY  
[group] HAVING amount > 1;
```

7.9 Korzystając z danych zawartych w tabeli vehicles wyświetl w jednej kolumnie o nazwie vehicle info następujące dane o samochodach: numer rejestracyjny (licence_plate), imię (o_first_name) i nazwisko (o_family_name) właściciela, model. Zastosuj format (spacje i przecinki) jak w poniższym przykładzie:

RBH99L3, Rust Cohle, Toyota

```
SELECT licence_plate || ', ' || o_first_name || ' ' || o_family_name || ', ' ||  
model AS [vehicle info] FROM vehicles;
```

7.10 Dla każdego rekordu z tabeli vehicles wyświetl liczbę, która określa sumę znaków w polach o_first_name, o_family_name oraz dodatkowego znaku spacji. Wynikowa kolumna powinna nazywać się number of chars

```
SELECT length(o_first_name) + length(o_family_name) + 1 AS [number of chars]  
FROM vehicles;  
SELECT length( o_first_name || o_family_name ) + 1 AS [number of chars] FROM  
vehicles;
```

7.11 Usuń tabelę movies.

```
DROP TABLE movies;
```

7.12 Ktoś narobił strasznego bajzlu w tej bazie danych. Usuń wszystkie tabele.

```
DROP TABLE qwerty;  
DROP TABLE students;  
DROP TABLE games;
```

7.13 Utwórz tabelę o nazwie smoczki, zawierającą dwie kolumny: nazwa typu TEXT oraz cena typu FLOAT.

```
CREATE TABLE smoczki ( nazwa TEXT, cena FLOAT );
```

7.14 Utwórz tabelę o nazwie smoczki, zawierającą trzy kolumny: nazwa typu TEXT, cena typu FLOAT oraz zapas typu INTEGER. Kolumny nazwa oraz cena nie mogą zawierać wartości NULL, a kolumna zapas powinna mieć wartość domyślną 0.

```
CREATE TABLE smoczki ( nazwa TEXT NOT NULL, cena FLOAT NOT NULL, zapas INTEGER  
DEFAULT 0 );
```

7.15 Utwórz tabelę o nazwie smoczki, zawierającą kolumny:

- nazwa typu TEXT, która nie może zawierać wartości NULL
- cena typu FLOAT, która nie może zawierać wartości NULL
- zapas typu INTEGER, która powinna mieć wartość domyślną 0 oraz nie można do nie wprowadzać wartości ujemnych.
- wzor typu TEXT, w której wartości nie mogą się powtarzać.

```
CREATE TABLE smoczki (nazwa TEXT NOT NULL, cena FLOAT NOT NULL, zapas INTEGER
DEFAULT 0, wzor TEXT UNIQUE, CHECK ( zapas >= 0 ) );
```

8.1 Utwórz tabelę o nazwie sliniaczki zawierającą kolumny:

- numer typu INTEGER będącą kluczem głównym tabeli z atrybutem AUTOINCREMENT,
- rozmiar typu INTEGER, która nie może zawierać wartości NULL,
- cena typu FLOAT, która domyślnie ma wartość 0 i nie może zawierać wartości ujemnych.

```
CREATE TABLE sliniaczki (numer INTEGER PRIMARY KEY AUTOINCREMENT, rozmiar
INTEGER NOT NULL, cena FLOAT DEFAULT 0 CHECK ( cena >= 0 ) );
```

8.2 W edytorze znajdziesz utworzoną wcześniej tabelę barbie. Zmień jej nazwę na my_dolls.

```
ALTER TABLE barbie RENAME TO my_dolls;
```

8.3 Do tabeli best_albums dodaj kolumnę songs_nr typu INTEGER. Uwzględnij ograniczenie, aby do tej kolumny dało się wprowadzić jedynie wartości większe od 0.

```
ALTER TABLE best_albums ADD COLUMN songs_nr INTEGER CHECK ( songs_nr > 0 );
```

8.4 Usuń wszystkie rekordy z tabeli best_movies_ever.

```
DELETE FROM best_movies_ever;
```

8.5 Z tabeli movies usuń wszystkie filmy, w których nie wystąpił Clint Eastwood.

```
DELETE FROM movies WHERE [cast] NOT LIKE '%Clint Eastwood%';
```

8.6 Dodaj do tabeli movies nowy rekord, dotyczący filmu "Incepcja". Dane filmu:

Tytuł oryginału: Inception

Rok produkcji: 2010

Gatunek: SF

Reżyser: Christopher Nolan

Obsada: Leonardo DiCaprio, Joseph Gordon-Levitt, Ellen Page

Ocena: 8.7

Nie zapomnij również przypisać właściwej wartości polu id - o jeden większej niż wartość tego pola id w ostatnim rekordzie. Wszystkie pola typu TEXT wpisz dokładnie w takiej formie, w jakiej zostały podane wyżej, uwzględniając duże i małe litery oraz znaki interpunkcyjne.

```
INSERT INTO movies VALUES (31, 'Inception', 2010, 'SF', 'Christopher Nolan',
'Leonardo DiCaprio, Joseph Gordon-Levitt, Ellen Page', '8.7');
```

8.7 nazywasz się Boner. Steve Boner. Jesteś policjantem drogówki w Chicago. Jest piękny, wiosenny poranek 15 maja 2017 roku. Właśnie zauważyłeś samochód o numerach rejestracyjnych HCB77E6, zaparkowany nieprawidłowo na miejscu dla niepełnosprawnych. Za takie wykroczenie obowiązuje mandat w wysokości 120\$. Zatrzymujesz się obok pojazdu. Z pobliskiego sklepu wychodzi staruszek i zmierza w Twoim kierunku. Staruszek widząc w Twoim ręku mandat marszczy czoło i przyśpiesza kroku. Wkłada rękę do kieszeni płaszczka. W bezwarunkowym odruchu, świadczącym o setkach godzin spędzonych na policyjnej strzelnicy, kładziesz dłoń na rękojeści rewolweru. Staruszek wyciąga z kieszeni kluczyki i autopilotem otwiera zamki swojego Mercedesa. Próbuje przepchnąć się obok Ciebie i zająć fotel kierowcy. Oczywiście uniemożliwiasz to. Prosisz uprzejmie o prawo jazdy. Staruszek to Richard Kimble, lat 75, id 42071355512. Notowany. Kilkanaście nieopłaconych, zaległych mandatów. Cóż, to nie Twoja sprawa, niech się tym martwi komornik. Musisz tylko wypełnić druczek mandatu o numerze 34662 i wręczyć go kierowcy. Robisz to wysłuchując niewzruszenie biadolenia staruszka, niestworzonych historii o jego rzekomym, niesłusznym posądzeniu o morderstwo żony, złorzeczeń na bezustanne prześladowania ze strony policji. Z służbowym

uśmiechem wręczasz kwit i wsiadasz do swojego wozu. Czas ruszyć w patrol. Ach, polityka firmy: trzeba jeszcze wprowadzić mandat do bazy danych...
`INSERT INTO tickets VALUES(34662, '2017-05-15', 'Chicago', '120.0', 'HCB77E6', 42071355512);`

8.8 Dodaj do tabeli A1709281523 pięć rekordów z wynikami pomiarów (każdy wiersz to jeden rekord):

```
1.12, 1.11,  
1.21, 1.43,  
1.17, null,  
1.54, 1.67,  
1.87, 1.93,  
INSERT INTO A1709281523 VALUES (1.12, 1.11), (1.21, 1.43), (1.17, null), (1.54, 1.67), (1.87, 1.93);
```

8.9 W tabeli movies ocena filmu 'Gran Torino' jest za niska. Zamiast 7.7 powinna wynosić 7.8. Zmień tę wartość.

```
UPDATE movies SET rating = '7.8' WHERE title = 'Gran Torino';
```

8.10 Tabela tickets zawiera dane mandatów wystawionych dla kierowców. Dane kierowców możemy odszukać w tabeli drivers. Pole id jest kluczem głównym tabeli drivers i zawiera numer identyfikacyjny kierowcy. W tabeli tickets numer identyfikacyjny kierowcy zawiera pole driver_id.

W wynikowej tabeli chcemy wyświetlić pola ticket_nr oraz fine z tabeli tickets, uzupełnione o imię i nazwisko kierowcy ukaranego mandatem (pola first_name i family_name w tabeli drivers).

```
SELECT tickets.ticket_nr, tickets.fine, drivers.first_name, drivers.family_name  
FROM tickets INNER JOIN drivers ON tickets.driver_id = drivers.id;
```

8.11 Z tabeli tickets wyświetl pola ticket_nr, fine dla wszystkich mandatów. W przypadku mandatów wystawionych dla kierowców, których jesteśmy w stanie odszukać w tabeli drivers, wyświetl także imię i nazwisko kierowcy ukaranego mandatem (pola first_name i family_name w tabeli drivers).

Pole id jest kluczem głównym tabeli drivers i zawiera numer identyfikacyjny kierowcy. W tabeli tickets numer identyfikacyjny kierowcy zawiera pole driver_id.

```
SELECT tickets.ticket_nr, tickets.fine, drivers.first_name, drivers.family_name  
FROM tickets LEFT JOIN drivers ON tickets.driver_id = drivers.id;
```

8.12 Z tabeli tickets wyświetl pole ticket_nr uzupełnione o nazwę modelu i rok produkcji samochodu, na który wystawiony został mandat. Dane te zawiera tabela vehicles (pola model oraz year). Interesują nas tylko te mandaty, dla których znajdziemy dane samochodu. Pole licence_plate jest kluczem głównym tabeli vehicles i zawiera numer rejestracyjny samochodu. W tabeli tickets numer rejestracyjny zawiera pole o takiej samej nazwie: licence_plate.

```
SELECT tickets.ticket_nr, vehicles.model, vehicles.year FROM tickets INNER JOIN  
vehicles ON tickets.licence_plate = vehicles.licence_plate;
```

8.13 W bazie danych znajdują się dwie tabele. Pole id jest kluczem głównym tabeli movies, natomiast tabela my_movies w kolumnie id zawiera wartości tego pola odpowiadające tym filmom, którym użytkownik wystawił swoją ocenę, zawartą w polu my_rating. Wyświetl tytuły filmów (title) oraz oceny użytkownika (my_rating), dla wszystkich filmów, dla których oceny użytkownik wystawił.

```
SELECT movies.title, my_movies.rating FROM movies INNER JOIN my_movies ON  
movies.id = my_movies.id;  
SELECT movies.title, my_movies.my_rating FROM my_movies LEFT JOIN movies ON  
my_movies.id = movies.id;
```

8.14 Wyświetl tytuły wszystkich książek z tabeli ksiazki oraz nazwy ich wydawnictw. Kluczowi głównemu id tabeli wydawnictwa odpowiada kolumna id_wydawnictwa w tabeli ksiazki.

```
SELECT ksiazki.tytul, wydawnictwa.nazwa FROM ksiazki LEFT JOIN wydawnictwa ON  
wydawnictwa.id = ksiazki.id_wydawnictwa;
```

8.15 Wyświetl nazwy wydawnictw oraz tytuły książek, które wydały. Weź pod uwagę tylko te wydawnictwa, których książki znajdują się w tabeli książki. Wyniki posortuj alfabetycznie po nazwach wydawnictw, a w ramach danego wydawnictwa po tytułach książek.

```
SELECT wydawnictwa.nazwa, książki.tytuł FROM wydawnictwa INNER JOIN książki ON wydawnictwa.id = książki.id_wydawnictwa ORDER BY wydawnictwa.nazwa, książki.tytuł;
```

8.16 Wyświetl numery mandatów z tabeli tickets (ticket_nr), numery rejestracyjne samochodów na które zostały wystawione (licence_plate), nazwy modeli tych samochodów (model) oraz w jednej kolumnie o nazwie driver imię i nazwisko kierowcy, który został ukarany (first_name, family_name). Imię od nazwiska powinno być oddzielone spacją. Interesują nas jedynie te mandaty, dla których możemy znaleźć komplet danych.

```
SELECT tickets.ticket_nr, vehicles.licence_plate, vehicles.model, drivers.first_name || " " || drivers.family_name AS driver FROM tickets INNER JOIN vehicles ON tickets.licence_plate = vehicles.licence_plate INNER JOIN drivers ON tickets.driver_id = drivers.id;
```

8.17 Z tabeli tickets wyświetl zsumowane wysokości mandatów, które zostały wystawione na poszczególne samochody (nazwij tę kolumnę total), numery rejestracyjne tych samochodów (licence_plate), oraz nazwiska ich właścicieli (o_family_name). Interesują nas jedynie te samochody, dla których potrafimy zidentyfikować właściciela.

```
SELECT SUM(fine) AS total, tickets.licence_plate, vehicles.o_family_name FROM tickets INNER JOIN vehicles ON tickets.licence_plate = vehicles.licence_plate GROUP BY tickets.licence_plate;
```