

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

9.1. W pojedynczej kolumnie o nazwie apple avg price wyświetl średnią cenę telefonów marki Apple.

```
SELECT AVG(price) AS [apple avg price] FROM cell_phones WHERE brand = 'Apple';
```

9.2. W pojedynczej kolumnie o nazwie avg price wyświetl średnią cenę telefonów z systemem Android i pamięcią RAM minimum 64 GB. Wynik zaokrąglij do pełnych centów.

```
SELECT ROUND( AVG(price), 2 ) AS [avg price] FROM cell_phones WHERE system = 'Android' AND RAM >= 64;
```

9.3. Kolumna ward w tabeli crimes zawiera nazwy okręgów i miejscowości, których dotyczą dane, kolumna population - liczbę ich mieszkańców, a kolumna district - nazwy dystryktów, do których należą. Kolumny od y_12 do y_16 zawierają liczby przestępstw popełnionych w latach 2012 - 2016. Dla każdego z okręgów wyświetl średnią roczną liczbę przestępstw popełnionych w okresie od 2012 do 2016 roku. Wynikowa tabela powinna zawierać dwie kolumny:

1. ward - z nazwą okręgu,
2. avg - ze średnią roczną liczbą przestępstw.

Posortuj tabelę malejąco po liczbie przestępstw.

```
SELECT ward, ( y_12 + y_13 + y_14 + y_15 + y_16 ) / 5.0 AS avg FROM crimes ORDER BY avg DESC;
```

9.4. Wyświetl wynik działania $3 / 4$. Wynikową kolumnę nazwij floating point arithmetic

```
SELECT 3.0/4 AS [floating point arithmetic];
```

9.5. Oblicz wartość wyrażenia: $3 / 2 * 4 + 5 * 4 / 12$. Wynikowej kolumnie nadaj nazwę 7 UP

```
SELECT 3/2.0*4+5*4/12.0 AS [7 UP];
```

9.6. Kolumna ward w tabeli crimes zawiera nazwy okręgów i miejscowości, których dotyczą dane, kolumna population - liczbę ich mieszkańców, a kolumna district - nazwy dystryktów, do których należą. Kolumny od y_12 do y_16 zawierają liczby przestępstw popełnionych w latach 2012 - 2016. Dla każdego z okręgów wyświetl liczbę przestępstw popełnionych w 2016 roku w przeliczeniu na jednego mieszkańca. Wynikowa tabela powinna zawierać dwie kolumny:

1. ward - z nazwą okręgu,
2. per capita 2016 - z liczbą przestępstw przypadających na jednego mieszkańca w danym okręgu w roku 2016.

Posortuj tabelę malejąco po liczbie przestępstw.

```
SELECT ward, y_16 / CAST( population AS FLOAT ) AS [per capita 2016] FROM crimes ORDER BY [per capita 2016] DESC;
```

9.7. tabela GDP (ang. gross domestic product) zawiera dane dotyczące Produktu Krajowego Brutto oraz ludności w niektórych państwach. Oblicz wartość PKB przypadającą na jednego mieszkańca dla każdego kraju.

Wynikowa tabela powinna zawierać dwie kolumny:

1. country
2. GDP per capita - zawierającą obliczoną wartość

Tabelę posortuj malejąco po wartościach z ostatniej kolumny.

```
SELECT country, CAST ( GDP_2016 AS FLOAT ) / population AS [GDP per capita] FROM GDP ORDER BY [GDP per capita] DESC;
```

9.8. Kolumna ward w tabeli crimes zawiera nazwy okręgów i miejscowości, których dotyczą dane, kolumna population - liczbę ich mieszkańców, a kolumna district - nazwy dystryktów, do których należą. Kolumny od y_12 do y_16 zawierają liczby przestępstw popełnionych w latach 2012 - 2016. Dla każdego z okręgów wyświetl liczbę przestępstw popełnionych w okresie od 2012 do 2016 w przeliczeniu na jednego mieszkańca. Wynikowa tabela powinna zawierać dwie kolumny:

1. ward- z nazwą okręgu,

2. crimes per capita - z liczbą przestępstw przypadających na jednego mieszkańca w danym okręgu. Posortuj tabelę malejąco po liczbie przestępstw.

```
SELECT ward, ( y_12 + y_13 + y_14 + y_15 + y_16 ) / CAST( population AS FLOAT )  
AS [crimes per capita] FROM crimes ORDER BY [crimes per capita] DESC;
```

9.9. Tabela jai_alai zawiera statystyki zawodników jai_alai. Kolumna W (win) zawiera liczbę wygranych meczów zawodnika, kolumna GP (games played) liczbę rozgrywek w których wziął udział. Twoim zadaniem jest obliczenie dla każdego zawodnika jaki procent rozgrywek, w których brał udział zakończył zwycięstwem. Wynikowa kolumna powinna zawierać trzy kolumny:

1. # - z numerem zawodnika
2. name - z przydomkiem zawodnika
3. W% - z wyrażonym w procentach udziałem zwycięstw we wszystkich meczach zawodnika.

Tabelę posortuj malejąco po wartościach kolumny W%

```
SELECT [#], name, CAST ( W AS FLOAT ) / GP * 100 AS [W%] FROM jai_alai ORDER BY  
[W%] DESC;
```

9.10. Z tabeli movies wyświetl tytuł filmu, reżysera, obsadę, oraz stosunek oceny (kolumna metacore) do czasu trwania filmu (kolumna runtime_minutes). Tę ostatnią kolumnę nazwij score/time, a wartości w niej zaokrąglaj do dwóch miejsc po przecinku.

```
SELECT title, director, [cast], ROUND ( CAST ( metacore AS FLOAT ) /  
runtime_minutes, 2 ) AS [score/time] FROM movies;
```

9.11. Tabela sausages zawiera listę różnych rodzajów kiełbas z całego świata. W kolumnie items podano liczbę kiełbas w opakowaniu, zaś w kolumnie price cenę w dolarach. Wyświetl koszt pojedynczej kiełbasy dla każdego rodzaju kiełbasy w tabeli. Wynikowa tabela powinna zawierać trzy kolumny:

1. name
2. country
3. price per item - z ceną pojedynczej kiełbasy.

```
SELECT name, country, ROUND ( price / items, 2 ) AS [price per item] FROM  
sausages;
```

9.12. Z tabeli cell_phones wyświetl trzy kolumny:

1. brand
2. model
3. price - zawierającą ceny telefonów zaokrągloną do pełnych dolarów.

```
SELECT brand, model, ROUND ( price ) AS price FROM cell_phones;
```

9.13. Tabela cell_phones w kolumnie price zawiera ceny telefonów podane w dolarach. Wyświetl średnie ceny telefonów posiadających taki sam system. Zaokrąglaj ją do pełnych centów.

Wynikowa tabela powinna zawierać dwie kolumny:

1. system
2. avg price - ze średnią ceną telefonów dla danego systemu

```
SELECT system, ROUND( AVG(price), 2 ) AS [avg price] FROM cell_phones GROUP BY  
system;
```

9.14. tabela famous_cats w kolumnie tail_length zawiera długości ogonów kotów podane w cm. W celu dostosowania tabeli dla klienta z Nowego Jorku, musisz przeliczyć centymetry na cale. Uwaga: Twoim zadaniem jest zmodyfikowanie oryginalnej tabeli (trwała zmiana), nie zaś jedynie wyświetlenie wynikowej tabeli z przeliczonymi wartościami. Przypomnij sobie, w jaki sposób zmienia się wartości w polach tabeli. Długości ogonów w calach mają być wyrażone z dokładnością do dwóch miejsc po przecinku.

Przyjmujemy, że 1 cal = 2.54 cm

```
UPDATE famous_cats SET tail_length = ROUND ( tail_length / 2.54, 2 );
```

9.15. Tabela population zawiera liczbę ludności poszczególnych krajów afrykańskich w latach od 2010 do 2015. Dla każdego kraju wyświetl średnią roczną zmianę populacji w okresie od 2010 do 2015 roku wyrażoną liczbowo. Wartość zaokrąglaj do dwóch miejsc po przecinku.

Wynikowa tabela powinna zawierać cztery kolumny:

1. country
2. MAX - z największą wartością populacji poszczególnych krajów w podanym okresie czasu,
3. MIN - z najmniejszą wartością populacji poszczególnych krajów w podanym okresie czasu,
4. change - zawierającą wyrażoną liczbowo średnią roczną zmianę populacji w podanym okresie czasu.

Posortuj rekordy zaczynając od krajów, w których średnia roczna zmiana populacji była największa, a kończąc na tych, w których była najmniejsza.

```
SELECT country, MAX( Y2010, Y2011, Y2012, Y2013, Y2014, Y2015 ) AS MAX,  
MIN( Y2010, Y2011, Y2012, Y2013, Y2014, Y2015 ) AS MIN, ( ( Y2011 - Y2010) +  
( Y2012 - Y2011) + ( Y2013 - Y2012) + ( Y2014 - Y2013) + ( Y2015 - Y2014) ) /  
5.0 AS change FROM population ORDER BY change DESC;
```

9.16. Wyświetl wartość bezwzględną liczby -5 w kolumnie o nazwie abs value

```
SELECT ABS ( -5 ) AS [abs value];
```

9.17. W wielopoziomowym, zautomatyzowanym parkingu, samochody transportowane są na odpowiednie poziomy za pomocą windy. Winda zaczyna i kończy operację zawsze na poziomie ulicy (poziom 0). Parking posiada część naziemną (piętra od 0 do 8) oraz część podziemną (piętra od -1 do -5). Wysokość jednego piętra to 2.5 metra. Tabela elevator zawiera rekordy z danymi dotyczącymi operacji wykonanych przez windę: w kolumnie # zapisane są numery kolejnych operacji, zaś w kolumnie floor numery pięter, na które dostarczone lub z którego odebrane zostały samochody. Oblicz sumaryczną drogę pokonaną przez windę w metrach. Wynikowa tabela powinna zawierać tylko jedną kolumnę distance z obliczoną wartością.

```
SELECT SUM ( ABS ( floor ) * 2 ) * 2.5 AS distance FROM elevator;
```

9.18. Tabela account zawiera dane transakcji z konta bankowego, w tym:

1. id_trans - numer transakcji
2. id_contr - id kontrahenta
3. amount - kwota transakcji w złotych; wartości dodatnie to wpływy na konto, wartości ujemne to wydatki z konta.

Oblicz średnią wartość wydatków. Wyświetl ją jako wartość dodatnią, zaokrągloną do pełnych groszy, w pojedynczej kolumnie o nazwie avg_exp

```
SELECT ROUND(ABS(AVG(amount)), 2) AS avg_exp FROM account WHERE amount < 0;
```

9.19. Tabela pressure zawiera dane pomiarów ciśnienia w komorze, dokonywanych dwa razy dziennie przez 365 kolejnych dni. Wyświetl w porządku malejącym dziesięć rekordów, w których zmiana ciśnienia pomiędzy pomiarem A i B była największa, niezależnie od tego czy był to spadek, czy wzrost ciśnienia.

Wynikowa tabela powinna zawierać cztery kolumny:

1. #
2. A
3. B
4. diff - zawierającą wartość bezwzględną różnicy ciśnień pomiędzy pomiarami A i B.

```
SELECT [#], A, B, ABS( A - B ) AS diff FROM pressure ORDER BY diff DESC limit  
10;
```

9.20. Z tabeli movies dla każdego z gatunków filmów wyświetl procentowy udział gatunku we wszystkich filmach. Zaokrąglaj wynik do pełnych promili. Wynikowa kolumna powinna zawierać dwie kolumny:

1. type - z nazwą gatunku
2. % - z procentowym udziałem gatunku w całej tabeli.

```
SELECT type, ROUND ( COUNT(*) / CAST ( ( SELECT COUNT(*) FROM movies ) AS  
FLOAT ) * 100, 1 ) AS [%] FROM movies GROUP BY type;
```

9.21. Wygeneruj losową liczbę.

```
SELECT RANDOM();
```

9.22. Wygeneruj losową liczbę z zakresu od 0 do 4.

```
SELECT ABS( RANDOM() ) % 5;
```

9.23. Wygeneruj losową liczbę z zakresu od 100 do 102.

```
SELECT ABS( RANDOM() ) % 3 + 100;
```

9.24. Tabela temperature zawiera odczyty czujnika temperatury dokonywane co godzinę przez 365 roku. Zakładając, że rok rozpoczął się od poniedziałku, oblicz średnią temperaturę mierzoną w samo południe (kolumna t12) ze wszystkich piątków w ciągu roku. Wynikowa tabela powinna zawierać tylko jedną kolumnę o nazwie avg12

```
SELECT AVG(t12) AS avg12 FROM temperature WHERE day % 7 = 5;
```

9.25. Kolumna ward w tabeli crimes zawiera nazwy okręgów i miejscowości, których dotyczą dane, kolumna population - liczbę ich mieszkańców, a kolumna district - nazwy dystryktów, do których należą. Kolumny od y_12 do y_16 zawierają liczby przestępstw popełnionych w latach 2012 - 2016. Dla każdego z dystryktów wyświetl liczbę przestępstw popełnionych w okresie od 2012 do 2016 w przeliczeniu na jednego mieszkańca. Wynikowa tabela powinna zawierać dwie kolumny:

1. district - z nazwą dystryktu,

2. crimes per capita - z liczbą przestępstw przypadających na jednego mieszkańca w danym dystrykcie. Wartości w tej kolumnie zaokrąglaj do dwóch miejsc po przecinku.

Posortuj tabelę malejąco po liczbie przestępstw.

```
SELECT district, ROUND( SUM( y_12 + y_13 + y_14 + y_15 + y_16 ) / CAST  
( SUM(population) AS FLOAT ), 2 ) AS [crimes per capita] FROM crimes GROUP BY  
district ORDER BY [crimes per capita] DESC;
```

10.1. Z tabeli movies wyszukaj rekordy dotyczące filmów, które w tytule zawierają apostrof '.

```
SELECT * FROM movies WHERE Title LIKE '%\'';
```

10.2. Z tabeli pass pobierz rekordy, które w polu password zawierają znak _.

```
SELECT * FROM pass WHERE password LIKE '%!_%' ESCAPE '!';
```

10.3. Skorzystaj z funkcji LIKE(), aby wyszukać w tabeli pass rekordy, zawierające w polu password przynajmniej jeden z dwóch znaków: % lub _.

```
SELECT * FROM pass WHERE LIKE ('!%', password, '!') OR LIKE ('!_%', password,  
'!');
```

10.4. Z tabeli tests pobierz rekordy, które na końcu numeru seryjnego (kolumna SN) mają małą literę. Na końcu numeru seryjnego występuje zawsze tylko jedna z trzech liter: a, b lub c.

```
SELECT * FROM tests WHERE SN GLOB '*[a-c]';
```

10.5. Z tabeli tests pobierz rekordy dotyczące silników, których numer seryjny (SN) zaczyna się od cyfr 1, 2, 3, 4 lub 5.

```
SELECT * FROM tests WHERE SN GLOB '[1-5]*';
```

10.6. Z tabeli tests pobierz rekordy dotyczące silników, które w numerze seryjnym, na czwartym miejscu w numerze seryjnym mają jedną z liter: x, e, b, s, d, f.

```
SELECT * FROM tests WHERE SN GLOB '???[xebdsf]*';
```

10.7. Z tabeli Artists pobierz rekordy dotyczące artystów urodzony w krajach, których nazwa zaczyna się na jedną z liter od A do H. Nazwę kraju urodzenia znajdziesz w kolumnie place_of_birth po nazwie miejscowości, przecinku i spacji.

```
SELECT * FROM Artists WHERE place_of_birth GLOB ', [A-H]*';
```

10.8. Z tabeli tests pobierz rekordy, które w polu SN na drugim miejscu nie zawierają żadnej z cyfr 2, 5, 8 zaś kończą się na literę a lub b.

```
SELECT * FROM tests WHERE SN GLOB '?[^258]*[ab]';
```

10.9. Utwórz tabelę o nazwie valves, zawierającą dwie kolumny:

1. id typu INTEGER będącą kluczem głównym tabeli
2. SN typu TEXT, która nie może zawierać wartości NULL, oraz do której można wprowadzić tylko łańcuchy spełniające następujące warunki: muszą się składać z pięciu znaków, pierwszym znakiem musi być cyfra, drugim znakiem musi być litera spośród standardowych liter kodu ASCII (obojętnie: mała lub duża).

```
CREATE TABLE valves (id INTEGER PRIMARY KEY, SN TEXT NOT NULL CHECK( SN GLOB '[0-9][a-zA-Z]???' ) );
```

10.10. W tabeli movies niektóre tytuły filmów zostały niepotrzebnie ujęte w cudzysłowy. Wyświetl całą tabelę w taki sposób, aby przy tytułach filmów nie było cudzysłowów. Zadbaj o to, aby w wynikowej tabeli nazwy kolumn pozostały takie same jak w tabeli źródłowej.

```
SELECT id, TRIM(title, '"') AS title, year, type, director, [cast], rating FROM movies;
```

10.11. Niektóre hasła użytkowników zawarte w kolumnie password w tabeli pass, zaczynają lub kończą się spacjami. System nie akceptuje takich haseł. Zmień dane w źródłowej tabeli w taki sposób, aby hasła nie zawierały spacji.

```
UPDATE pass SET password = TRIM(password);
```

10.12. Tabela colors zawiera listę barw i odpowiadających im kodów RGB w systemie szesnastkowym i dziesiętnym. Wyświetl całą tabelę w taki sposób, aby wartości w kolumnie Hex nie zawierały na początku znaku #, zaś wartości w kolumnie Dec były pozbawione nawiasów (). Nazwy kolumn powinny pozostać bez zmian.

```
SELECT Color, LTRIM(Hex, '#') AS Hex, LTRIM( RTRIM( Dec, ')'), '(') AS Dec FROM colors;
```

10.13. Wyświetl liczbę znaków, z których składają się kolejne hasła w kolumnie password w tabeli pass. Wynikowa tabela powinna zawierać trzy kolumny:

1. user_id
 2. password
 3. length - z liczbą znaków, z których składa się dane hasło.
- Posortuj tabelę malejąco po wartościach ostatniej kolumny.

```
SELECT user_id, password, length(password) AS length FROM pass ORDER BY length DESC;
```

10.14. Tabela tickets w polu driver_id zawiera PESEL identyfikujący kierowcę. Przedostatnia cyfra w PESELu służy do określenia płci: cyfra parzysta oznacza kobietę, cyfra nieparzysta mężczyznę. Pobierz z tabeli wszystkie rekordy dotyczące mandatów wystawionych kobietom.

```
SELECT * FROM tickets WHERE SUBSTR(driver_id, -2, 1) % 2 = 0;
```

10.15. Tabela tickets zawiera dane o wystawionych mandatach. W kolumnie date podana jest data wystawienia w formacie YYYY-MM-DD (czyli rok, miesiąc, dzień). Rozmieść dane o roku, miesiącu i dniu wystawienia mandatów w trzech osobnych kolumnach.

Wynikowa tabela powinna zawierać pięć kolumn:

1. ticket_nr
2. year - z rokiem wystawienia mandatu
3. month - z miesiącem wystawienia mandatu
4. day - z dniem miesiąca wystawienia mandatu.
5. driver_id

Uwaga: liczby określające miesiąc i dzień nie mogą zaczynać się od zera!

Np. dla pierwszego mandatu w tabeli źródłowej wiersz w tabeli wynikowej powinien wyglądać tak:

```
ticket_nr year month day driver_id  
213 1975 3 12 47073065432
```

```
SELECT ticket_nr, SUBSTR(date, 1, 4) AS year, LTRIM( SUBSTR(date, 6, 2), '0') AS month, LTRIM( SUBSTR(date, -2, 2), '0') AS day, driver_id FROM tickets;
```

10.16. W tabeli movies kolumna Genre zawiera opis gatunków, do jakich należą poszczególne filmy. Wyświetl kolumny: Title, Genre, Director i Actors, zamieniając w kolumnie Genre oznaczenie Sci-Fi na SF. Nazwy kolumn powinny zostać takie, jak w tabeli źródłowej.

```
SELECT Title, REPLACE(Genre, 'Sci-Fi', 'SF') AS Genre, Director, Actors FROM movies;
```

10.17. Zmodyfikuj tabelę źródłową tests w taki sposób, aby nieprawidłowo wpisany symbol As w polu SN zamienić na symbol Qs.

```
UPDATE tests SET SN = REPLACE(SN, 'As', 'Qs');
```

10.18. W tabeli vehicles postanowiono ujednoczyć pisownię marek samochodów i pisać je dużymi literami. Zmień dane w tabeli źródłowej vehicles w kolumnie model, tak aby wszystkie nazwy zawierały same duże litery.

```
UPDATE vehicles SET model = UPPER(model);
```

10.19. tabela tests zawiera rekordy z wynikami testów silników o numerach seryjnych zapisanych w kolumnie SN. Dla każdego z silników wyświetl liczbę wykonanych na nim testów. Nie bierz przy tym uwagę rozróżnienia na rodzaj użytych zaworów, które w numerze seryjnym zostały oznaczone małą lub dużą literą na ostatnim miejscu. Tzn. numery seryjne składające się z takich samych ciągów znaków – w tym obojętnie: dużych czy małych liter – traktuj jako odnoszące się do tego samego silnika.

Wynikowa tabela powinna zawierać dwie kolumny:

1. SN - z małymi literami zamienionymi na duże,
2. tests - z liczbą testów danego silnika.

Tabelę posortuj malejącą po kolumnie tests.

```
SELECT UPPER(SN) AS SN, COUNT(1) AS [tests] FROM tests GROUP BY UPPER(SN) ORDER BY [tests] DESC;
```

10.20. Minister transportu postanowił umożliwić posiadaczom samochodów zdalny dostęp do bazy z danymi na tematów pojazdów. Login każdego właściciela auta generowany jest automatycznie i składa się z połączonych ze sobą w jeden wyraz trzech pierwszych liter imienia oraz trzech pierwszych liter nazwiska. Cały login zawiera jedynie małe litery.

Na podstawie tabeli vehicles, zawierającej imiona i nazwiska właścicieli samochodów, wyświetl tabelę zawierającą trzy kolumny:

1. o_first_name - z imieniem właściciela pojazdu
2. o_family_name - z nazwiskiem właściciela pojazdu
3. login - z loginem do systemu dla danego właściciela.

Uwaga: rekordy w wynikowej tabeli muszą być unikalne, tzn. każda osoba może pojawić się w niej tylko raz.

```
SELECT DISTINCT o_first_name, o_family_name, LOWER( SUBSTR( o_first_name, 1, 3 ) || SUBSTR( o_family_name, 1, 3 ) ) AS login FROM vehicles ORDER BY login;
```

10.21. W tabeli morse_code zawiera serię komunikatów nadanych alfabetem morse'a. Dla każdego komunikatu wyświetl miejsce wystąpienia słowa over, które w kodzie morse'a ma zapis: --- ...- . .-. .

Wynikowa tabela powinna zawierać dwie kolumny:

1. signal
2. over - z pozycją pierwszego znaku słowa over.

```
SELECT signal, INSTR(signal, '--- ...- . .-.') AS over FROM morse_code;
```

22. Wyświetl tytuły wszystkich filmów z tabeli movies i inicjały ich reżyserów. Nazwy kolumn i wartości w pierwszym wierszu powinny wyglądać tak:

```
title          initials
Blade runner   RS
```

```
SELECT title, SUBSTR(director, 1, 1) || SUBSTR(director, INSTR(director, ' ') + 1, 1) AS initials FROM movies;
```

23. Tabela cell_phones w polu price zawiera cenę telefonów w dolarach. Wyświetl informacje o producencie, modelu telefonu oraz cenie podanej w euro, zaokrąglonej do pełnych eurocentów. Dla jasności w jakiej walucie wyrażona jest cena, za kwotą wstaw spację oraz symbol euro (w notacji dziesiętnej numer tego

znaku w tabeli UNICODE to 8364).

Wynikowa tabela powinna zawierać trzy kolumny:

1. brand
2. model
3. price - z ceną wyrażoną w euro.

Przyjmujemy, że 1 euro to 1.15 dolara.

```
SELECT brand, model, ROUND( price / 1.15, 2 ) || ' ' || CHAR(8364) AS price FROM cell_phones;
```

24. Na podstawie danych z tabeli tickets, oblicz średnią wysokość mandatów wystawionych w miesiącach sierpień i wrzesień. Wynik wyświetl w pojedynczej kolumnie o nazwie avg 8-9

```
SELECT AVG(fine) AS [avg 8-9] FROM tickets WHERE SUBSTR(date, 6, 2) = '08' OR SUBSTR(date, 6, 2) = '09';
```

25. W tabeli Artists w polu name przed przecinkiem umieszczono nazwisko artysty, zaś po przecinku jego imię. Wyświetl w pojedynczej kolumnie o nazwie name same nazwiska artystów w kolejności alfabetycznej. Weź pod uwagę tylko te rekordy, które w polu name zawierają przecinek.

```
SELECT SUBSTR(name, INSTR(name, ','), - ( INSTR(name, ',')-1 )) AS name FROM Artists WHERE name LIKE '%,%' ORDER BY name;
```

11.1. W bazie danych znajdziesz tabele: famous_cats - z danymi sławnych kotów oraz cartoons - z danymi różnych bajek. W wierszu poleceń wpisz zapytanie:

```
SELECT * FROM famous_cats, cartoons;
```

11.2. Tabela chessboard_rows w kolumnie row zawiera symbole rzędów szachownicy (cyfry od 1 do 8), zaś tabela chessboard_columns w kolumnie col zawiera symbole kolumn szachownicy (litery od A do H). Wyświetl tabelę składającą się z dwóch kolumn: row i col, w której każdy wiersz zawiera współrzędne jednego pola szachownicy.

```
SELECT * FROM chessboard_rows, chessboard_columns;
```

11.3. Przygotowujesz dla dzieci grę polegającą na połączeniu linią imion sławnych kotów z bajkami z jakich pochodzą. W tym celu chcesz wydrukować tabelę zawierającą wszystkie kombinacje imion kotów (kolumna name) z tabeli famous_cats i tytułów bajek (kolumna title) z tabeli cartoons.

```
SELECT name, title FROM famous_cats, cartoons;
```

11.4. Tabela books zawiera dane książek, zaś tabela authors dane pisarzy.

Wyświetl tytuły wszystkich książek z tabeli books (kolumna title) oraz imiona i nazwiska ich autorów (kolumny first_name i last_name).

```
SELECT title, first_name, last_name FROM books, authors WHERE author_id = id;
```

11.5. Dla wszystkich książek z tabeli books, wyświetl ich tytuły oraz nazwy wydawców. Nazwy wydawców zawiera kolumna name w tabeli publishers.

```
SELECT title, name FROM books, publishers WHERE id = publisher_id;
```

11.6. Dla każdej książki z tabeli books wyświetl w następujących kolumnach dane:

1. title - tytuł książki
2. author - imię i nazwisko autora, oddzielone spacją
3. publisher - nazwę wydawcy

Tabelę posortuj alfabetycznie po tytułach książek.

```
SELECT title, first_name || " " || last_name AS author, name AS publisher FROM books, authors, publishers WHERE author_id = authors.id AND publisher_id = publishers.id ORDER BY title;
```

11.7. Korzystając z tabel w bazie danych, wyświetl w wynikowej tabeli kolumny:

1. first_name - imię kierowcy
2. last_name - nazwisko kierowcy

3. sum - sumę na jaką zostały wystawione wszystkie mandaty dla danego kierowcy. Tabelę posortuj poczynając od kierowców, na których nałożono największą łączną karę.

```
SELECT first_name, last_name, SUM(fine) AS sum FROM tickets JOIN drivers ON driver_id = id GROUP BY driver_id ORDER BY sum DESC;
```

11.8. Korzystając z tabel w bazie danych, dla wszystkich mandatów z tabeli tickets wyświetl w wynikowej tabeli kolumny:

1. ticket_nr - numer mandatu
2. fine - wysokość mandatu
3. o_first_name - imię właściciela pojazdu
4. o_last_name - nazwisko właściciela pojazdu
5. licence_plate - numer rejestracyjny pojazdu

Użyj NATURAL JOIN

```
SELECT ticket_nr, fine, o_first_name, o_last_name, licence_plate FROM tickets NATURAL JOIN vehicles;
```

11.9. Tabela protagonists zawiera dane bohaterów bajek, zaś tabela cartoons dane samych bajek. Wyświetl imię bohatera, tytuł bajki, z której pochodzi oraz datę premiery bajki (release_year). Tabelę posortuj od bajek najnowszych do najstarszych. Użyj NATURAL JOIN

```
SELECT name, title, release_year FROM protagonists NATURAL JOIN cartoons ORDER BY release_year DESC;
```

11.10. Obydwie tabele w bazie danych zawierają kolumnę id z numerami identyfikacyjnymi filmów. Wyświetl kolumny:

1. id - z numerami id filmów
2. title - z tytułami filmów
3. diff - z różnicą pomiędzy oceną filmów z tabeli my_movies (kolumna rating), a oceną filmów z tabeli movies (kolumna rating). Różnicę zaokrąglij do jednego miejsca po przecinku.

Użyj USING()

```
SELECT id, title, ROUND((my_movies.rating - movies.rating), 1) AS diff FROM movies JOIN my_movies USING (id);
```

11.11. Wyświetl nazwy wszystkich wydawców z tabeli publishers oraz tytuły wydanych przez nich książek (o ile takie istnieją w bazie danych).

```
SELECT name, title FROM publishers LEFT JOIN books ON id = publisher_id;
```

11.12. Dla wszystkich autorów z tabeli authors, których książki znajdują się w bazie danych, wyświetl tytuły tych książek oraz nazwy ich wydawnictw.

Wynikowa tabela powinna zawierać trzy kolumny:

1. author - z nazwiskiem i imieniem autora (oddzielonymi spacją)
2. title - z tytułem książki
3. name - z nazwą wydawnictwa

Tabelę posortuj alfabetycznie po autorach.

```
SELECT last_name || " " || first_name AS author, title, name FROM authors JOIN books ON author_id = authors.id JOIN publishers ON publisher_id = publishers.id ORDER BY author;
```

11.13. Dla wszystkich kierowców z tabeli drivers wyświetl ich imiona, nazwiska oraz średnią wysokość otrzymanych przez nich mandatów. Jeżeli dany kierowca nie otrzymał żadnych mandatów, pole to powinno zawierać wartość null.

Wynikowa tabela powinna składać się z trzech kolumn:

1. first_name
2. last_name
3. avg - ze średnią wartością mandatów wystawionych na danego kierowcę, zaokrągloną do pełnych centów.

Tabelę posortuj malejąco po kolumnie avg.

```
SELECT first_name, last_name, ROUND(AVG(fine), 2) AS avg FROM drivers LEFT JOIN tickets ON id = driver_id GROUP BY id ORDER BY avg DESC;
```

11.14. Z tabeli movies dla każdego filmu, wyświetl w kolejnych wierszach tytuły filmów, które mają od niego wyższą ocenę oraz wartość wskazującą o ile jest to

wyższa ocena (zaokrągloną do jednego miejsca po przecinku).

Wynikowa tabela powinna składać się z trzech kolumn:

1. title1 - z tytułem pierwotnego filmu
2. title2 - z tytułem filmu, który ma wyższą ocenę niż film pierwotny,
3. diff - z różnicą między oceną filmu z wyższym ratingiem, a oceną filmu pierwotnego, zaokrągloną do jednego miejsca po przecinku.

Uwzględnij tylko te filmy, dla których istnieją filmy z wyższą oceną.

Wynikową tabelę posortuj alfabetycznie, w pierwszej kolejności po tytułach filmów pierwotnych, a w drugiej po tytułach filmów z wyższą oceną.

```
SELECT m1.title title1, m2.title title2, ROUND(m2.rating - m1.rating, 1) diff
FROM movies m1 JOIN movies m2 ON m1.rating < m2.rating ORDER BY m1.title,
m2.title;
```

11.15. Powtórz polecenie z poprzedniego zadania, z jedną różnicą: tym razem wyświetlając tytuły filmów pierwotnych, uwzględnij także najwyżej notowane filmy (czyli te, dla których nie ma filmów z wyższą oceną).

```
SELECT m1.title title1, m2.title title2, ROUND(m2.rating - m1.rating, 1) diff
FROM movies m1 LEFT JOIN movies m2 ON m1.rating < m2.rating ORDER BY m1.title,
m2.title;
```

11.16. Z tabeli employees wyświetl w kolumnie employee imię i nazwisko każdego pracownika (oddzielone spacją), a w kolumnie supervisor imię i nazwisko jego przełożonego (również oddzielone spacją). Jeżeli pracownik nie ma przełożonego, w kolumnie supervisor powinna znaleźć się wartość null.

Posortuj tabelę alfabetycznie po kolumnie employee.

```
SELECT t1.first_name || " " || t1.last_name employee, t2.first_name || " " ||
t2.last_name supervisor FROM employees t1 LEFT JOIN employees t2 ON
t1.supervisor_id = t2.id ORDER BY employee;
```