# **Notatki**

|  |  |
| --- | --- |
| Nazwa | Opis |
| SELECT | Wybór danych z tabeli |
| FROM | Określenie z której tabeli mają być wybierane dane |
| WHERE | Filtruje rekordy |
| AS | Alias, nazwa kolumny w kwerendzie |
| % | Wiele lub brak znaków |
| \_ | Dokładnie jeden znak |
| \* | Wszystkie kolumny tabeli |
| DISTINCT | Wyświetla unikatowe wartości, nie będą się powtarzać |
| COUNT | Wyświetla ilość wierszy (samo **COUNT** – ilość wierszy w tabeli) |
| GROUP BY | Grupowanie wartości w kolumnie o takich samych wartościach, rekordy nie będą się powtarzać  **GROUP** **BY** 2 **→** grupowanie po drugiej kolumnie w kwerendzie |
| HAVING | Działa tak jak WHERE ale dotyczy tylko **GROUP** **BY**;ogranicza **GROUP BY** |
| LIKE | Sprawdza nazwę, wyświetla je |
| NOT LIKE | Sprawdza nazwę, nie wyświetla ich |
| IN | Wyświetla takie jak z góry podane, znajduje się w **WHERE** |
| NOT IN | Nie wyświetla takie jak z góry podane, znajduje się w **WHERE** |
| CONCAT | Łączenie ze sobą dwa lub więcej **STRING** |
| ORDER BY | Sortowanie rekordów, domyślnie rosnąco (**ASC**) lub malejąco (**DESC**) **ORDER** **BY** 2 **→** sortowanie po drugiej kolumnie w kwerendzie |
| REPLACE | Zastępowanie danych w **STRING** nowym **STRING** |
| ROUND | Zaokrąglanie do określonej liczby miejsc po przecinku |
| LENGTH | Zwraca długość danego **STRING** w bajtach (spacje też) |
| LEFT | Zwraca ilość znaków z **STRING** licząc od lewej strony |
| MID | Zwraca ilość znaków z **STRING** licząc od start |
| SUBSTRING | Zwraca ilość znaków z **STRING** licząc od start |
| IS NULL | Służy do sprawdzania wartości pustych (**NULL**) |
| IS NOT NULL | Służy do sprawdzania wartości nie pustych (**NOT** **NULL**) |
| BETWEEN | Pozwala na sprawdzanie przedziału |
| LIMIT | Maksymalna ilość zwracanych rekordów |
| CASE | Warunki, jeśli dany warunek jest spełniony wyświetla określone informacje, jeśli nie wyświetla informacje zawarte po **ELSE** |

## **Kolejność:**

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. HAVING
6. ORDER BY
7. LIMIT

**Typy baz danych:**

* Relacyjne
* Nierelacyjne (obiektowe)
* Grafowe

**Rodzaje relacji:**

* 1 : 1
* 1 : ∞
* ∞ : ∞

**W jakich plikach, w usystematyzowany sposób przechowuje się bazy danych:**

* **.csv**, comma-separated values,wartości odzdzielone „ , ” lub „ ; ”, spacją lub dowolnym innym znakiem
* **.json**, JavaScript Object Notation, java script objcet, słownikowe, klucz i wartość ???
* **.xml**, Extensible Markup Language ,struktura podobna do html

**Typy danych, indeksy:**

* **Klastrowane** – tylko jeden indeks w tabeli, odwołanie do pamięci fizycznej, sortuje rekordy
* **Nieklastrowane** – odwołanie do pamięci fizycznej

## **Podzapytanie (subquery)**

Podzapytanie to zapytanie SQL, które umieszczone jest wewnątrz innego zapytania. Podzapytanie zawsze otoczone jest parą nawiasów. Może wystąpić w SELECT, FROM, WHERE. Używane kiedy jedno zapytanie ma bazować na wyniku drugiego zapytania. Nazwy kolumn w podzapytaniach należy poprzedzać nazwami lub aliasami nazw tabel. Jeśli kolumna o podanej nazwie nie istnieje w tabeli odczytywanej przez wewnętrzne zapytanie, SQL Server sprawdza, czy kolumna o tej nazwie istnieje w tabeli odczytywanej przez zapytanie zewnętrzne. Jeżeli tak, zostanie ona użyta do sprawdzenia warunku WHERE.

**Używane są do:**

* Porównywania wyrażeń do rezultatu kwerendy
* Określania czy wyrażenie zawiera się w rezultacie kwerendy
* Sprawdzenia czy kwerenda wybierze jakiś rekord

**Syntax**

|  |
| --- |
| **SELECT** select\_list  **FROM** table  **WHERE** expr\_operator  (**SELECT** select\_list  **FROM** table) |

## **CASE**

Dla każdego wiersza zwraca wynik (**THEN**) zależny od warunku (**WHEN**), jeśli żaden z warunków nie zostanie spełniony zwraca to co znajduje się po **ELSE**.

**Syntax**

|  |
| --- |
| **CASE**  **WHEN** warunek **THEN** …  **ELSE** …  END AS alias |

# **Przykłady i syntax**

**AS** Alias, nazwa kolumny w kwerendzie

|  |
| --- |
| **SELECT** column\_name **AS** alias\_name **FROM** table\_name |

**CONCAT** Łączenie ze sobą dwóch lub więcej STRING

|  |
| --- |
| **SELECT** CustomerName, **CONCAT**(Address, " ", PostalCode, " ", City) **AS** Address  **FROM** Customers |
| **CONCAT**(expression1, expression2, expression3,...) |

**ORDER BY** Sortowanie rekordów, domyślnie rosnąco (**ASC**) lub malejąco (**DESC**)

|  |
| --- |
| **SELECT** \*  **FROM** Customers  **ORDER** **BY** Country |
| **SELECT** column1, column2, ... **FROM** table\_name **ORDER** **BY** column1, column2, ... **ASC|DESC** |

**LENGTH** Zwraca długość danego **STRING** w bajtach (spacje też)

|  |
| --- |
| **SELECT** CustomerName, **LENGTH**(CustomerName) **AS** LengthOfName  **FROM** Customers |
| **LENGTH**(string) |

**IS NULL** Służy do sprawdzania wartości pustych (**NULL**)

|  |
| --- |
| **SELECT** CustomerName, ContactName, Address  **FROM** Customers  **WHER**E Address **IS NULL** |
| **SELECT** column\_names **FROM** table\_name **WHERE** column\_name **IS** **NULL** |

**IS NOT NULL** Służy do sprawdzania wartości nie pustych (**NOT** **NULL**)

|  |
| --- |
| **SELECT** CustomerName, ContactName, Address **FROM** Customers **WHERE** Address **IS NOT NULL** |
| **SELECT**column\_names **FROM**table\_name **WHERE**column\_name**IS NOT NULL** |

**REPLACE** zastępowanie danych w **STRING**

|  |
| --- |
| **SELECT** **REPLACE**('SQL Tutorial', 'T', 'M') |
| **REPLACE**(string, old\_string, new\_string) |

**ROUND** Zaokrąglanie do określonej liczby miejsc po przecinku

|  |
| --- |
| **SELECT ROUND**(235.415, 2)**AS**RoundValue |
| **ROUND**(number,decimals,operation) |

**LEFT** Zwraca ilość znaków z **STRING** licząc od lewej strony

|  |
| --- |
| **SELECT** **LEFT**('SQL Tutorial', 3) AS ExtractString |
| **LEFT**(string, number\_of\_chars) |

**MID** Zwraca ilość znaków z **STRING** licząc od start

|  |
| --- |
| **SELECT** **MID**("SQL Tutorial", 5, 3) **AS** ExtractString |
| **MID**(string, start, length) |

**SUBSTRING** Zwraca ilość znaków z **STRING** licząc od start

|  |
| --- |
| **SELECT** **SUBSTRING**(CustomerName, 2, 5) **AS** ExtractString  **FROM** Customers; |
| **SUBSTRING**(string, start, length) |

**BETWEEN** Pozwala na sprawdzanie przedziału

|  |
| --- |
| **SELECT** \*  **FROM** Products **WHERE** Price **BETWEEN** 10 **AND** 20; |
| **SELECT** column\_name(s) **FROM** table\_name **WHERE** column\_name **BETWEEN** value1 **AND** value2; |

**LIMIT** Maksymalna ilość zwracanych rekordów

|  |
| --- |
| **SELECT** \*  **FROM** Customers  **LIMIT** 10; |
| **SELECT** column\_name(s) **FROM** table\_name **WHERE** condition **LIMIT** number; |

# **Kwerendy z lekcji**

## **Zadanie 1**

1. Wyświetl imię, nazwisko i inicjały (w jednej kolumnie) dla wszystkich klientów, których inicjały to te same litery. Wyniki posortuj malejąco po imieniu, i rosnąco po nazwisku.

|  |
| --- |
| **SELECT** first\_name, last\_name, **CONCAT**(**LEFT**(first\_name, 1), **LEFT**(last\_name, 1)) **AS** Inicjaly  **FROM** customer  **WHERE** **LEFT**(first\_name,1) = **LEFT**(last\_name,1)  **ORDER** **BY** 1 **DESC**, 2 |

1. Wyświetl sumę długości filmów, ich średnią długość, minimalną długość oraz maksymalną długość na każdy rating w tabeli. Ograniczając wyniki do filmów które w swojej nazwę posiadają ciąg znaków "the". Wyniki posortuj rosnąco po sumie długości.

|  |
| --- |
| **SELECT** rating, **SUM**(length) **AS** Suma, **AVG**(length) **AS** Srednia, **MIN**(length) **AS** Miniumum, **MAX**(length) **AS** Maksimum  **FROM** film  **WHERE** title **LIKE** '%the%'  **GROUP** **BY** rating  **ORDER** **BY** 2 |

1. Wyświetl na każdy rating sumę długości filmu, dla filmów których długość jest równa lub mniejsza od średniej długości wszystkich filmów w tabeli. Wyniki posortuj malejąco po sumie długości filmów.

|  |
| --- |
| **SELECT** rating, **SUM**(length) **AS** Suma  **FROM** film  **WHERE** length <= (**SELECT** **AVG**(length) **FROM** film)  **GROUP** **BY** rating  **ORDER** **BY** 2 **DESC** |

1. Dla każdego filmu który posiada w swoim tytule trzy litery "a", wyświetl tytuł filmu oraz ilość aktorów grających w filmie. Wyniki posortuj po liczbie aktorów grających w filmie.

|  |
| --- |
| **SELECT** title, (**SELECT** **COUNT**(\*) **FROM** film\_actor **WHERE** film.film\_id = film\_actor.film\_id) **AS** Liczba\_aktorow  **FROM** film  **WHERE** title **LIKE** '%a%a%a%'  **ORDER** **BY** 2 |

## **Zadanie 2**

1. Wyświetl tytuł oraz ilość słów w opisie filmu. Wyniki ogranicz do top 25 filmów z najdłuższym opisem. Wyniki posortuj od A do Z

|  |
| --- |
| **SELECT** title, **LENGTH**(description)-length(**REPLACE**(description, ' ',''))+1 **AS** Ilosc\_slow  **FROM** film  **ORDER** **BY** 2, 1  **LIMIT** 25 |

1. Wyświetl dla każdego ratingu długość filmu, rental duration oraz ilość filmów których rental duration jest parzysta, a film posiada przynajmniej dwa special features. Wyniki posortuj malejąco po rental duration.

|  |
| --- |
| **SELECT** rating, rental\_duration, **COUNT**(\*), **SUM**(length)  **FROM** film  **WHERE** rental\_duration % 2 = 0 **AND** special\_features **LIKE** '%,%'  **GROUP** **BY** rating  **ORDER** **BY** 2 **DESC** |

1. Wyświetl dla wszystkich unikatowych adresów z Californi ile osób z danego adresu jest użytkownikami. Wyniki posortuj po adresie i liczbie wypożyczeń rosnąco.

|  |
| --- |
| **SELECT** **DISTINCT** address.address, (**SELECT** **COUNT**(\*) **FROM** customer **WHERE** customer.address\_id = address.address\_id) **AS** Ilosc\_klientow  **FROM** address  **WHERE** address.district = 'California'  **ORDER** **BY** 1, 2 |

1. Wyświetl kraj oraz ilość miast z danego kraju. Wyniki posortuj malejąco po liczbie miast.

|  |
| --- |
| **SELECT** country, (**SELECT** **COUNT**(\*) **FROM** city **WHERE** country.country\_id = city.country\_id) **AS** 'Ilosc miast'  **FROM** country  **ORDER** **BY** 2 **DESC** |

## **Zadanie 3**

1. Dla każdego ratingu wyświetl sumę, wartość maksymalną, minimalną rental\_duration oraz ile razy dany rating występuje w tabeli. Wyniki posortuj po liczbie wierszy na rating i rosnąco po sumie rental\_duration

|  |
| --- |
| **SELECT** rating, **SUM**(rental\_duration), **MAX**(rental\_duration), **MIN**(rental\_duration), **COUNT**(rating)  **FROM** film  **GROUP** **BY** rating  **ORDER** **BY** 5 **DESC**, 2 **ASC** |

1. Wyniki ogranicz jedynie do filmów posiadających literę „a” w swoim tytule

|  |
| --- |
| **SELECT** rating, **SUM**(rental\_duration), **MAX**(rental\_duration), **MIN**(rental\_duration), **COUNT**(rating)  **FROM** film  **WHERE** title **LIKE** '%a%'  **GROUP** **BY** rating  **ORDER** **BY** 5 **DESC**, 2 **ASC** |

1. Wyniki ogranicz do filmów których długość jest większa od średniej długości wszystkich filmów

|  |
| --- |
| **SELECT** rating, **SUM**(rental\_duration), **MAX**(rental\_duration), **MIN**(rental\_duration), **COUNT**(rating)  **FROM** film  **WHERE** title **LIKE** '%a%' **AND** length > (**SELECT** **AVG**(length) **FROM** film)  **GROUP** **BY** rating  **ORDER** **BY** 5 **DESC**, 2 **ASC** |

1. Wyniki ogranicz do tych których minimum długości filmów jest większe od 85 minut

|  |
| --- |
| **SELECT** rating, **SUM**(rental\_duration), **MAX**(rental\_duration), **MIN**(rental\_duration), **COUNT**(rating)  **FROM** film  **WHERE** title **LIKE** '%a%' **AND** length > (**SELECT** **AVG**(length) **FROM** film)  **GROUP** **BY** rating  **HAVING** 85 < (**SELECT** **SUM**(length) **FROM** film)  **ORDER** **BY** 5 **DESC**, 2 **ASC** |

1. Dla każdego rental\_rate wyświetl rental\_rate, sumę, średnią długości filmów oraz ilość filmów. Wyniki posortuj rosnąco po rental\_rate i malejąco po rental\_rate

|  |
| --- |
| **SELECT** rental\_rate, **SUM**(length), **AVG**(length), **COUNT**(\*)  **FROM** film  **GROUP** **BY** rental\_rate  **ORDER** **BY** 1 **DESC**, 1 **ASC** |

1. Zapytanie ogranicz do filmów których druga litera to „t”

|  |
| --- |
| **SELECT** rental\_rate, **SUM**(length), **AVG**(length), **COUNT**(\*)  **FROM** film  **WHERE** title **LIKE** '\_t%'  **GROUP** **BY** rental\_rate  **ORDER** **BY** 1 **DESC**, 1 **ASC** |

1. Wyświetl również dla każdego rental\_rate sumę długości wszystkich filmów, dla których rental\_rate jest różne od obecnego rental\_rate

|  |
| --- |
| **SELECT** tab1.rental\_rate, **SUM**(tab1.length) **AS** Suma, **AVG**(tab1.length) **AS** Srednia, **COUNT**(\*) **AS** Ile, (**SELECT** **SUM**(tab2.length) **FROM** film tab2 **WHERE** tab2.rental\_rate <> tab1.rental\_rate) **AS** INNE  **FROM** film tab1  **WHERE** tab1.title **LIKE** '\_t%'  **GROUP** **BY** 1  **ORDER** **BY** 1 |

1. Wyniki ogranicz do rekordów gdzie suma długości filmów na rental\_rate jest podzielna przez 5

|  |
| --- |
| **SELECT** tab1.rental\_rate, **SUM**(tab1.length) **AS** Suma, **AVG**(tab1.length) **AS** Srednia, **COUNT**(\*) **AS** Ile, (**SELECT** **SUM**(tab2.length) **FROM** film tab2 **WHERE** tab2.rental\_rate <> tab1.rental\_rate) **AS** INNE  **FROM** film tab1  **WHERE** tab1.title **LIKE** '\_t%'  **GROUP** **BY** 1  **HAVING** Suma % 5 = 0  **ORDER** **BY** 1 |

## **Zadanie 4**

1. Wyświetl dla wszystkich filmów których długość jest mniejsza od liczby 120 tytuł filmu, jego długość oraz w trzeciej kolumnie jeżeli rating nie jest "R" lub "PG" słowa "Nie dla dzieci". w przeciwnym razie napisz "Dla dzieci". Wyniki posortuj malejąco po długości filmu

|  |
| --- |
| **SELECT** length, title, rating, **CASE**  **WHEN** rating = 'R' **THEN** 'Nie dla dzieci'  **WHEN** rating = 'PG' **THEN** 'Nie dla dzieci'  **ELSE** 'Dla dzieci'  **END** **AS** nazwa  **FROM** film  **WHERE** length < 120  **ORDER** **BY** 1 **DESC** |

## **Zadanie 5**

1. Wyświetl dla każdego miasta którego city\_id jest podzielne przez 5 nazwę miasta oraz jeżeli długość nazwy miasta jest mniejsza bądź równa 10 znaków – wyświetl państwo z którego jest dane miasto, w przeciwnym razie wyświetl district.  
   Tabele city, address, country

|  |
| --- |
| **SELECT** city.city **AS** Miasto, city.city\_id, (**CASE**  **WHEN** **LENGTH**(city.city) <= 10 **THEN** (**SELECT** country.country **FROM** country **WHERE** country.country\_id = city.country\_id)  **ELSE** (**SELECT** address.district **FROM** address **WHERE** address.city\_id = city.city\_id) **END**) **AS** Kolumna  **FROM** city  **WHERE** city.city\_id % 5 = 0 |

1. Wyświetl dla wszystkich filmów posiadających przynajmniej dwa special\_feauters jego kategorię wtedy gdy długość filmu przekracza średnią długość wszystkich filmów w ratingu danego filmu, w przeciwnym wypadku wyświetl ile sztuk danego filmu jest w magazynie. Wyniki posortuj od Z do A po nazwie filmu.

|  |
| --- |
| **SELECT** Nazwa1.title, Nazwa1.rating, (**CASE**  **WHEN** Nazwa1.length > (**SELECT** **AVG**(film.length) **FROM** film **WHERE** film.rating = Nazwa1.rating) **THEN** (**SELECT** category.name **FROM** category **WHERE** category.category\_id = (**SELECT** film\_category.category\_id **FROM** film\_category **WHERE** film\_category.film\_id = Nazwa1.film\_id))  **ELSE** (**SELECT** **COUNT**(\*) **FROM** inventory **WHERE** inventory.film\_id = Nazwa1.film\_id)  **END**) **AS** Kolumna  **FROM** film Nazwa1  **WHERE** Nazwa1.special\_features **LIKE** '%,%'  **ORDER** **BY** title **DESC** |

## **Zadanie 6**

1. Wyświetl drugą literę z tytułu każdego filmu, sumę długości filmu, średnią wartość z kolumny ‘rental\_duration’ zaokrągloną do jednego miejsca po przecinku i jako ostatnią kolumnę warunek gdy ilość klientów których pierwsza litera imienia to ta druga litera tytułu filmu jest większa od średniej wartości ‘rental\_duration’, gdy warunek będzie spełniony napisz „Warunek spełniony”, w przeciwnym razie napisz „Nie spełniono warunku”. Wyniki wyświetl dla drugiej litery. Całkowite wyniki ogranicz do ilości filmów których druga litera to ta wybrana jest większa od liczby 45.

|  |
| --- |
| **SELECT** **SUBSTRING**(tabf.title, 2, 1) **AS** Druga\_Litera,  **SUM**(tabf.length) **AS** Suma,  **ROUND**(**AVG**(tabf.rental\_duration), 1) **AS** Srednia,  (**CASE**  **WHEN** (**SELECT** **COUNT**(\*) **FROM** customer **WHERE** **SUBSTRING**(customer.first\_name, 1, 1) = Druga\_Litera) > **ROUND**(**AVG**(tabf.rental\_duration), 1) **THEN** 'Warunek spełniono'  **ELSE** 'Nie spełniono warunku'  **END**) **AS** Kolumna  **FROM** film tabf  **GROUP** **BY** 1  **HAVING** **COUNT**(\*) > 45 |

1. Wyświetl w ilu filmach grają aktorzy których imię zaczyna się na „a”.

|  |
| --- |
| **SELECT** **COUNT**(**DISTINCT** film\_actor.film\_id)  **FROM** film\_actor  **WHERE** film\_actor.actor\_id **IN** (**SELECT** actor.actor\_id **FROM** actor **WHERE** **SUBSTRING**(actor.first\_name, 1, 1) = 'A') |

1. Dla filmów których tytuł jest dłuższy niż średnia ilość znaków w tytule ogólnie wyświetl „Tak” w przeciwnym wypadku wyświetl „Nie”

|  |
| --- |
| **SELECT** inTab.SredDlugoscTytulu, **COUNT**(\*) **FROM**  (**SELECT** film.title, (**CASE**  **WHEN** **LENGTH**(film.title) > (**SELECT** **AVG**(**LENGTH**(film.title)) **FROM** film) **THEN** 'Tak'  **ELSE** 'Nie'  **END**) **AS** SredDlugoscTytulu  **FROM** film) **AS** inTab  **GROUP** **BY** 1 |

# **Kartkówki**

## **Kartkówka nr 1**

POLECENIE, całość na 5

|  |
| --- |
| **SELECT** address.district,  **SUM**(**LEFT**(address.address,1)) **AS** Suma\_1\_znak,  **AVG** (address.postal\_code) **AS** Sred\_pc,  (**SELECT** **COUNT**(\*)  **FROM** country  **WHERE** country.country **LIKE** '%a%' **AND** country.country\_id **IN**  (**SELECT** city.country\_id  **FROM** city  **WHERE** city.city\_id = address.city\_id)) **AS** Ile\_krajow  **FROM** address  **WHERE** length(address.address) > 4  **GROUP** **BY** address.district  **HAVING** Sred\_pc > 1000  **ORDER** **BY** 2 **DESC** |

# **Kwerendy z sqlzoo**

## **1. SELECT basics**

1. The example uses a WHERE clause to show the population of 'France'. Note that strings (pieces of text that are data) should be in 'single quotes'. **Modify it to show the population of Germany**

|  |
| --- |
| **SELECT** population  **FROM** world  **WHERE** name = 'Germany' |

1. Checking a list The word IN allows us to check if an item is in a list. The example shows the name and population for the countries 'Brazil', 'Russia', 'India' and 'China'. **Show the name and the population for 'Sweden', 'Norway' and 'Denmark'.**

|  |
| --- |
| **SELECT** name, population  **FROM** world  **WHERE** name **IN** ('Sweden', 'Norway', 'Denmark') |

1. Which countries are not too small and not too big? BETWEEN allows range checking (range specified is inclusive of boundary values). The example below shows countries with an area of 250,000-300,000 sq. km. **Modify it to show the country and the area for countries with an area between 200,000 and 250,000.**

|  |
| --- |
| **SELECT** name, area  **FROM** world  **WHERE** area **BETWEEN** 200000 **AND** 250000 |

## **2. SELECT name**

1. You can use WHERE name LIKE 'B%' to find the countries that start with "B". The % is a wild-card it can match any characters. **Find the country that start with Y**

|  |
| --- |
| **SELECT** name  **FROM** world **WHERE** name **LIKE** 'Y%' |

1. **Find the countries that end with y**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '%Y' |

1. Luxembourg has an **x** - so does one other country. List them both. **Find the countries that contain the letter x**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '%x%' |

1. Iceland, Switzerland end with **land** - but are there others? **Find the countries that end with land**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '%land' |

1. Columbia starts with a **C** and ends with **ia** - there are two more like this. **Find the countries that start with C and end with ia**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** 'C%ia' |

1. Greece has a double **e** - who has a double **o**? **Find the country that has oo in the name**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '%oo%' |

1. Bahamas has three **a** - who else? **Find the countries that have three or more a in the name**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '%a%a%a%' |

1. India and Angola have an **n** as the second character. You can use the underscore as a single character wildcard. **Find the countries that have "t" as the second character.**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '\_t%'  **ORDER** **BY** name |

1. Lesotho and Moldova both have two "o" characters separated by two other characters. **Find the countries that have two "o" characters separated by two others.**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '%o\_\_o%' |

1. Cuba and Togo have four characters names. **Find the countries that have exactly four characters.**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '\_\_\_\_' |

1. The capital of Luxembourg is Luxembourg. Show all the countries where the capital is the same as the name of the country. **Find the country where the name is the capital city.**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** capital |

1. The capital of **Mexico** is **Mexico City**. Show all the countries where the capital has the country together with the word "City". **Find the country where the capital is the country plus "City".** The CONCAT function - The function CONCAT is short for concatenate - you can use it to combine two or more strings.

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** capital = **CONCAT**(name, ' City') |

1. **Find the capital and the name where the capital includes the name of the country.**

|  |
| --- |
| **SELECT** capital, name  **FROM** world  **WHERE** capital **LIKE** **CONCAT**('%', name, '%') |

1. **Find the capital and the name where the capital is an extension of name of the country.** You should include **Mexico City** as it is longer than **Mexico**. You should not include **Luxembourg**as the capital is the same as the country.

|  |
| --- |
| **SELECT** capital, name  **FROM** world  **WHERE** capital LIKE concat(name, '\_%') |

1. For **Monaco-Ville** the name is **Monaco** and the extension is **-Ville**. **Show the name and the extension where the capital is an extension of name of the country.** You can use the SQL function REPLACE.

|  |
| --- |
| **SELECT** name, **MID**(capital, length(name)+1) **EXT** **FROM** world  **WHERE** capital **LIKE** **CONCAT**(name,'\_%') |

## **3. SELECT FROM world**

1. Read the notes about this table. Observe the result of running this SQL command to show the name, continent and population of all countries.

|  |
| --- |
| **SELECT** name, continent, population  **FROM** world |

1. How to use WHERE to filter records. Show the name for the countries that have a population of at least 200 million. 200 million is 200000000, there are eight zeros.

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** population > 200000000 |

1. Give the name and the **per capita GDP** for those countries with a population of at least 200 million. How to calculate per capita GDP - per capita GDP is the GDP divided by the population GDP/population

|  |
| --- |
| **SELECT** name, gdp/population  **FROM** world  **WHERE** population > 200000000 |

1. Show the name and population in millions for the countries of the continent 'South America'. Divide the population by 1000000 to get population in millions.

|  |
| --- |
| **SELECT** name, population/1000000  **FROM** world  **WHERE** continent = 'South America' |

1. Show the name and population for France, Germany, Italy

|  |
| --- |
| **SELECT** name, population  **FROM** world  **WHERE** name **IN** ('France','Germany','Italy') |

1. Show the countries which have a name that includes the word 'United'

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '%United%' |

1. Two ways to be big: A country is **big** if it has an area of more than 3 million sq km or it has a population of more than 250 million. **Show the countries that are big by area or big by population. Show name, population and area.**

|  |
| --- |
| **SELECT** name, population, area  **FROM** world  **WHERE** area>3000000 **OR** population>250000000 |

1. **Exclusive OR (XOR). Show the countries that are big by area or big by population but not both. Show name, population and area.**Australia has a big area but a small population, it should be **included**.  
   Indonesia has a big population but a small area, it should be **included**.  
   China has a big population **and** big area, it should be **excluded**.  
   United Kingdom has a small population and a small area, it should be **excluded**.

|  |
| --- |
| **SELECT** name, population, area **FROM** world  **WHERE** (population>250000000 **OR** area>3000000) **AND** **NOT**(population>250000000 **AND** area>3000000) |

1. Show the name and population in millions and the GDP in billions for the countries of the continent 'South America'. Use the ROUND function to show the values to two decimal places. **For South America show population in millions and GDP in billions both to 2 decimal places.** Millions and billions - Divide by 1000000 (6 zeros) for millions. Divide by 1000000000 (9 zeros) for billions.

|  |
| --- |
| **SELECT** name, **ROUND**(population/1000000,2), **ROUND**(gdp/1000000000,2)  **FROM** world  **WHERE** continent = 'South America' |

1. Show the name and per-capita GDP for those countries with a GDP of at least one trillion (1000000000000; that is 12 zeros). Round this value to the nearest 1000. **Show per-capita GDP for the trillion dollar countries to the nearest $1000.**

|  |
| --- |
| **SELECT** name, **ROUND**(gdp/population,-3)  **FROM** world  **WHERE** gdp>1000000000000 |

1. Greece has capital Athens. Each of the strings 'Greece', and 'Athens' has 6 characters. **Show the name and capital where the name and the capital have the same number of characters.** You can use the LENGTH function to find the number of characters in a string

|  |
| --- |
| **SELECT** name, capital  **FROM** world  **WHERE** **LENGTH**(name)=**LENGTH**(capital) |

1. The capital of Sweden is Stockholm. Both words start with the letter 'S'. **Show the name and the capital where the first letters of each match. Don't include countries where the name and the capital are the same word.** You can use the function LEFT to isolate the first character. You can use <> as the **NOT EQUALS** operator.

|  |
| --- |
| **SELECT** name, capital **FROM** world  **WHERE** **LEFT**(name,1) = **LEFT**(capital,1) **AND** name<>capital |

1. **Equatorial Guinea** and **Dominican Republic** have all of the vowels (a e i o u) in the name. They don't count because they have more than one word in the name. **Find the country that has all the vowels and no spaces in its name.**You can use the phrase name NOT LIKE '%a%' to exclude characters from your results. The query shown misses countries like Bahamas and Belarus because they contain at least one 'a'

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** name **LIKE** '%a%'  **AND** name **LIKE** '%e%'  **AND** name **LIKE** '%i%'  **AND** name **LIKE** '%o%'  **AND** name **LIKE** '%u%'  **AND** name **NOT** **LIKE** '% %' |

### **3.2 WHERE filters**

1. The population of 'France'. **Strings should be in 'single quotes'.**

|  |
| --- |
| **SELECT** population  **FROM** bbc  **WHERE** name = 'France' |

1. The names and population densities for the very large countries. **We can use mathematical and string expressions as well as field names and constants.**

|  |
| --- |
| **SELECT** name, population/area  **FROM** bbc  **WHERE** area > 5000000 |

1. Where to find some very small, very rich countries. **We use AND to ensure that two or more conditions hold true.**

|  |
| --- |
| **SELECT** name, region  **FROM** bbc  **WHERE** area < 2000 **AND** gdp > 5000000000 |

## **4. SELECT from Nobel Tutorial**

1. Change the query shown so that it displays Nobel prizes for 1950.

|  |
| --- |
| **SELECT** yr, subject, winner  **FROM** nobel  **WHERE** yr = 1950 |

1. Show who won the 1962 prize for Literature.

|  |
| --- |
| **SELECT** winner  **FROM** nobel  **WHERE** yr = 1962 **AND** subject = 'Literature' |

1. Show the year and subject that won 'Albert Einstein' his prize.

|  |
| --- |
| **SELECT** yr, subject  **FROM** nobel  **WHERE** winner = 'Albert Einstein' |

1. Give the name of the 'Peace' winners since the year 2000, including 2000.

|  |
| --- |
| **SELECT** winner  **FROM** nobel  **WHERE** subject = 'Peace' **AND** yr >= 2000 |

1. Show all details (**yr**, **subject**, **winner**) of the Literature prize winners for 1980 to 1989 inclusive.

|  |
| --- |
| **SELECT** yr, subject, winner  **FROM** nobel  **WHERE** subject = 'Literature' **AND** yr **BETWEEN** 1980 AND 1989 |

1. Show all details of the presidential winners: Theodore Roosevelt, Woodrow Wilson, Jimmy Carter,Barack Obama

|  |
| --- |
| **SELECT** \*  **FROM** nobel  **WHERE** winner **IN** ('Theodore Roosevelt', 'Woodrow Wilson', 'Jimmy Carter', 'Barack Obama') |

1. Show the winners with first name John

|  |
| --- |
| **SELECT** winner  **FROM** nobel  **WHERE** winner **LIKE** 'John %' |

1. Show the year, subject, and name of Physics winners for 1980 together with the Chemistry winners for 1984.

|  |
| --- |
| **SELECT** \*  **FROM** nobel  **WHERE** (subject='physics' **AND** yr=1980) **OR** (subject='chemistry' **AND** yr=1984) |

1. Show the year, subject, and name of winners for 1980 excluding Chemistry and Medicine

|  |
| --- |
| **SELECT** \*  **FROM** nobel  **WHERE** yr=1980 **AND** subject **NOT** **IN** ('Chemistry', 'Medicine') |

1. Show year, subject, and name of people who won a 'Medicine' prize in an early year (before 1910, not including 1910) together with winners of a 'Literature' prize in a later year (after 2004, including 2004)

|  |
| --- |
| **SELECT** \*  **FROM** nobel  **WHERE** (subject = 'Medicine' **AND** yr <1910) **OR** (subject = 'Literature' **AND** yr>=2004) |

1. Find all details of the prize won by PETER GRÜNBERG. Non-ASCII characters - The u in his name has an **umlaut**.

|  |
| --- |
| **SELECT** \*  **FROM** nobel  **WHERE** winner **IN** ('Peter Grünberg') |

1. Find all details of the prize won by EUGENE O'NEILL. Escaping single quotes - You can't put a single quote in a quote string directly. You can use two single quotes within a quoted string.

|  |
| --- |
| **SELECT** \*  **FROM** nobel  **WHERE** winner **IN** ('Eugene O''Neill') |

1. List the winners, year and subject where the winner starts with Sir. Show the most recent first, then by name order.

|  |
| --- |
| **SELECT** winner, yr, subject  **FROM** nobel  **WHERE** winner **LIKE** 'Sir%'  **ORDER** **BY** yr **DESC**, winner |

1. The expression **subject IN ('Chemistry','Physics')** can be used as a value - it will be 0 or 1. Show the 1984 winners and subject ordered by subject and winner name; but list Chemistry and Physics last.

|  |
| --- |
| **SELECT** winner, subject  **FROM** nobel  **WHERE** yr=1984  **ORDER** **BY** subject **IN** ('Physics', 'Chemistry'), subject, winner |

## **5. SELECT within SELECT Tutorial**

1. **List each country name where the population is larger than that of 'Russia'.** world (name, continent, area, population, gdp)

|  |
| --- |
| **SELECT** name **FROM** world  **WHERE** population > (**SELECT** population **FROM** world **WHERE** name='Russia') |

1. **Show the names of countries in Europe with a per capita GDP greater than 'United Kingdom'.** Per Capita GDP - The per capita GDP is the gdp/population

|  |
| --- |
| **SELECT** name **FROM** world  **WHERE** continent='Europe' **AND** gdp/population > (**SELECT** gdp/population **FROM** world **WHERE** name='United Kingdom') |

1. **List the name and continent of countries in the continents containing either Argentina or Australia. Order by name of the country.**

|  |
| --- |
| **SELECT** name, continent  **FROM** world  **WHERE** continent **IN** (**SELECT** continent **FROM** world **WHERE** name **IN** ('Australia', 'Argentina'))  **ORDER** BY name |

1. **Which country has a population that is more than Canada but less than Poland? Show the name and the population.**

|  |
| --- |
| **SELECT** name, population  **FROM** world  **WHERE** population **BETWEEN**  (**SELECT** population+1 **FROM** world **WHERE** name='Canada') **AND** (**SELECT** population-1 **FROM** world **WHERE** name='Poland') |

1. Germany (population 80 million) has the largest population of the countries in Europe. Austria (population 8.5 million) has 11% of the population of Germany. **Show the name and the population of each country in Europe. Show the population as a percentage of the population of Germany.** Decimal places - You can use the function ROUND to remove the decimal places. Percent symbol % - You can use the function CONCAT to add the percentage symbol.

|  |
| --- |
| **SELECT** name, **CONCAT**(**ROUND**(100\*population/(**SELECT** population **FROM** world **WHERE** name='Germany')),'%')  **FROM** world  **WHERE** continent='Europe' |

1. **Which countries have a GDP greater than every country in Europe? [Give the name only.] (Some countries may have NULL gdp values)**

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** gdp > **ALL** (**SELECT** gdp **FROM** world **WHERE** continent = 'Europe' **AND** gdp **IS** **NOT** **NULL**) |

1. **Find the largest country (by area) in each continent, show the continent, the name and the area**

|  |
| --- |
| **SELECT** continent, name, area  **FROM** world x  **WHERE** area >= **ALL** (**SELECT** area **FROM** world y **WHERE** y.continent=x.continent and area > 0 ) |

1. **List each continent and the name of the country that comes first alphabetically.**

|  |
| --- |
| **SELECT** continent, name  **FROM** world x  **WHERE** x.name <= **ALL** (**SELECT** name **FROM** world y **WHERE** x.continent=y.continent) |

1. **Find the continents where all countries have a population <= 25000000. Then find the names of the countries associated with these continents. Show name, continent and population.**

|  |
| --- |
| **SELECT** name, continent, population  **FROM** world x  **WHERE** 25000000 >= **ALL** (**SELECT** population **FROM** world y **WHERE** x.continent=y.continent **AND** y.population>0) |

1. **Some countries have populations more than three times that of any of their neighbours (in the same continent). Give the countries and continents.**

|  |
| --- |
| **SELECT** name, continent  **FROM** world x  **WHERE** population > **ALL**(**SELECT** population\*3 **FROM** world y **WHERE** y.continent = x.continent **AND** y.name != x.name) |

### **5.1 Using nested SELECT**

1. List each country in the same continent as 'Brazil'.

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** continent = (**SELECT** continent **FROM** world **WHERE** name = 'Brazil') |

1. List each country and its continent in the same continent as 'Brazil' or 'Mexico'.

|  |
| --- |
| **SELECT** name, continent  **FROM** world  **WHERE** continent **IN** (**SELECT** continent **FROM** world **WHERE** name='Brazil' **OR** name='Mexico') |

1. Show the population of China as a multiple of the population of the United Kingdom.

|  |
| --- |
| **SELECT** population/(**SELECT** population **FROM** world **WHERE** name='United Kingdom')  **FROM** world  **WHERE** name = 'China' |

1. **Show each country that has a population greater than the population of ALL countries in Europe.** Note that we mean greater than every single country in Europe; not the combined population of Europe.

|  |
| --- |
| **SELECT** name  **FROM** world  **WHERE** population > **ALL** (**SELECT** population **FROM** world **WHERE** continent='Europe') |

### **5.2 SELECT .. SELECT**

1. You may use a SELECT statement in the FROM line. In this case the derived table has columns name and gdp\_per\_capita. The calculated values in the inner SELECT can be used in the outer SELECT.

|  |
| --- |
| **SELECT** name, **ROUND**(gdp\_per\_capita)  **FROM** (**SELECT** name, gdp/population **AS** gdp\_per\_capita **FROM** bbc) X  **WHERE** gdp\_per\_capita>20000 |

1. **Find the countries in the same region as Bhutan.** You may use a SELECT statement in the WHERE line - this returns a list of regions.

|  |
| --- |
| **SELECT** name  **FROM** bbc  **WHERE** region **IN** (**SELECT** region **FROM** bbc **WHERE** name='Bhutan') |

1. If a value from the outer query appears in the inner query this is "correlated subquery". **Show the countries where the population is greater than 5 times the average for its region.**

|  |
| --- |
| SELECT name  FROM bbc b1  WHERE population > 5\*(SELECT AVG(population) FROM bbc WHERE region=b1.region) |

## **6. SUM and COUNT**

1. Show the total **population** of the world. world(name, continent, area, population, gdp)

|  |
| --- |
| **SELECT** **SUM**(population)  **FROM** world |

1. List all the continents - just once each.

|  |
| --- |
| **SELECT** **DISTINCT**(continent)  **FROM** world |

1. Give the total GDP of Africa

|  |
| --- |
| **SELECT** **SUM**(gdp)  **FROM** world  **WHERE** continent = 'Africa' |

1. How many countries have an **area** of at least 1000000

|  |
| --- |
| **SELECT** **COUNT**(name)  **FROM** world  **WHERE** area >= 1000000 |

1. What is the total **population** of ('Estonia', 'Latvia', 'Lithuania')

|  |
| --- |
| **SELECT** **SUM**(population)  **FROM** world  **WHERE** name **IN** ('Estonia', 'Latvia', 'Lithuania') |

1. For each **continent** show the **continent** and number of countries.

|  |
| --- |
| **SELECT** continent, **COUNT**(name)  **FROM** world  **GROUP** **BY**(continent) |

1. For each **continent** show the **continent** and number of countries with populations of at least 10 million.

|  |
| --- |
| **SELECT** continent, **COUNT**(name)  **FROM** world  **WHERE** population >= 10000000  **GROUP** **BY**(continent) |

1. List the continents that **have** a total population of at least 100 million.

|  |
| --- |
| **SELECT** continent  **FROM** world  **GROUP** **BY** continent  **HAVING** **SUM**(population)>= 100000000 |

### **6.1 Using SUM, COUNT, MAX, DISTINCT, ORDER BY**

1. The total population and GDP of Europe.

|  |
| --- |
| **SELECT** **SUM**(population), SUM(gdp)  **FROM** bbc  **WHERE** region = 'Europe' |

1. What are the regions?

|  |
| --- |
| **SELECT** **DISTINCT** region  **FROM** bbc |

1. Show the name and population for each country with a population of more than 100000000. **Show countries in descending order of population.**

|  |
| --- |
| **SELECT** name, population  **FROM** bbc  **WHERE** population > 100000000  **ORDER** BY population DESC |

### **5.6 Using GROUP BY and HAVING**

1. For each continent show the number of countries: world(name, continent, area, population, gdp)

|  |
| --- |
| **SELECT** continent, **COUNT**(name)  **FROM** world  **GROUP** **BY** continent |

1. For each continent show the total population: world(name, continent, area, population, gdp)

|  |
| --- |
| **SELECT** continent, **SUM**(population)  **FROM** world  **GROUP** **BY** continent |

1. WHERE and GROUP BY. The WHERE filter takes place before the aggregating function. For each relevant continent show the number of countries that has a population of at least 200000000.

|  |
| --- |
| **SELECT** continent, **COUNT**(name)  **FROM** world  **WHERE** population>200000000  **GROUP** **BY** continent |

1. GROUP BY and HAVING. The HAVING clause is tested after the GROUP BY. You can test the aggregated values with a HAVING clause. Show the total population of those continents with a total population of at least half a billion.

|  |
| --- |
| **SELECT** continent, **SUM**(population)  **FROM** world  **GROUP** **BY** continent  **HAVING** **SUM**(population)>500000000 |