

SELECT

- SELECT pobiera dane z tabeli. Każda tabela zawiera wiersze i kolumny - Można wybrać kilka kolumn i ignorować pozostałe kolumny.
- Pozostałe nazwy kolumn na wybranej linii kontroli, które kolumny można uzyskać FROM kontroli, która tabela dostęp klauzuli

Tabela olimpiady zawiera dane w jakim roku jakie miasto gościło Olimpiadę..

Olimpiady		
yr	city	y1
2000	Sydney	
2004	Athens	2000
2008	Beijing	2004
2012	London	2008

Ćwiczenie 1.

Spraw by klauzula SELECT z tabeli Olimpiady zwróciła W jakim mieście w którym roku gościła Olimpiada.

```
SELECT yr, city FROM Olimpiady
```

SELECT .. WHERE

The table games shows the year and the city hosting the Olympic Games.

```
Olimpiady
yr    city
2000 Sydney
2004 Athens
2008 Beijing
2012 London
```

1.The SELECT statement returns results from a *table*. With a WHERE clause only some rows are returned. This example shows the year that Athens hosted the Olympic games.

```
SELECT yr, city
```

```
FROM Olimpiady
```

WHERE yr = 2004

SELECT .. GROUP BY

Host cities and continents for the Olympics Games are stored in the table `games`. Notice that Europe appears in the table twice:

```
      games
  yr  city  continent
2000 Sydney Australasia
2004 Athens Europe
2008 Beijing Asia
2012 London Europe
1.
```

In a GROUP BY statement only *distinct* values are shown for the column in the GROUP BY. This example shows the continents hosting the Olympics with the count of the number of games held.

```
SELECT continent, COUNT(yr) FROM games
```

```
GROUP BY continent
```

```
*****
```

The SELECT .. JOIN statement

Sometimes you need to access two or more tables to get the data required.

games		city	
yr	city	name	country
1896	Athens	Sydney	Australia
1948	London	Athens	Greece
2004	Athens	Beijing	China
2008	Beijing	London	UK
2012	London		

1.

You can use a JOIN to get results from two or more related tables. In this example each row of the table `games` is related to a row of the table `city`. If you want to find the country where the games took place you must JOIN the games table to the city table on the common field that is `games.city` and `city.name`

```
SELECT games.yr, city.country
```

```
FROM games JOIN city
```

```
ON (games.city = city.name)
```

.....

INSERT .. VALUES

```
CREATE TABLE USERS (
```

```
  UserId INT NOT NULL AUTO_INCREMENT,  
  FName VARCHAR (255),  
  Mname VARCHAR (255),  
  LName VARCHAR (255),  
  PRIMARY KEY (UserId)  
)
```

```
INSERT INTO USERS (Fname,Mname,Lname) VALUES ('Saad','A','Mian');
```

INSERT .. SELECT

INSERT .. SELECT

The table `games` shows the year and the city hosting the Olympic Games.

games

yr	city
2000	Sydney
2004	Athens
2008	Beijing

1.

The INSERT SELECT statement adds a new row to the table based on a SELECT statement: In this example you run the next three Olympic games in the same three venues:

```
INSERT INTO games(yr,city)
```

```
  SELECT yr+12, city FROM games;
```

```
SELECT * FROM games;
```

UPDATE

UPDATE

The table `games` shows the year and the city hosting the Olympic Games.

games	
yr	city
2000	Sydney
2004	Athens
2008	Beijing
2012	London

The table `number` shows the year and the city hosting the Olympic Games.

number	
yr	number
2000	1
2004	2
2008	3
2012	4

1.

The UPDATE statement can be used to change a values in rows that already exists. In this example we move the 2012 games from London to Paris.

```
UPDATE games SET city='Paris' WHERE yr = 2012;
```

```
SELECT * FROM games;
```

LEFT JOIN

The SELECT .. LEFT JOIN statement

The LEFT JOIN will include rows from the left table even when the linking value is null.

games

yr	city	city	name	country
2004	Athens			
2008	Beijing	Sydney		Australia
2012	London	Athens		Greece
2032		Beijing		China
		London		UK

1.

There is no data on where the 2032 games will be held. The LEFT JOIN will include a row for 2032 even though it has no corresponding city.

```
SELECT games.yr, city.country
```

```
FROM games LEFT JOIN city
```

```
ON (games.city = city.name)
```

DELETE

The table games shows the year and the city hosting the Olympic Games.

games

yr	city
2000	Sydney
2004	Athens
2008	Beijing
2012	London

1.

The SELECT statement returns results from a *table*. The DELETE statement can be used to remove rows from a table. In this example we remove the 2000 games from the table:

```
DELETE FROM games WHERE yr=2000;
```

```
SELECT * FROM games;
```

.....

CREATE VIEW

The table `games` shows the year and the city hosting the Olympic Games.

```
games
yr  city
2004 Athens
2008 Beijing
1.
```

The `CREATE VIEW` names a `SELECT` query. That query may be used as if it were a table in many contexts. In this example the `VIEW old_games` shows those games before 2006.

```
CREATE VIEW og AS
SELECT yr,city FROM games
WHERE yr<2006;
SELECT * FROM og;
```

UNION

Make union between different tables to build one single view or request?

1.

List a number of `SELECT` statements separated by the `UNION` key word. Be sure that you have the same number of columns in each of the `SELECT` statements.

```
SELECT name FROM bbc WHERE name LIKE 'Z%'
UNION
```

```
SELECT name FROM actor WHERE name LIKE 'Z%'
```

name	continent	area	population	gdp
Afghanistan	Asia	652230	25500100	20343000000
Albania	Europe	28748	2831741	12960000000
Algeria	Africa	2381741	37100000	188681000000
Andorra	Europe	468	78115	3712000000
Angola	Africa	1246700	20609294	100990000000

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 = 'France'
```

```
SELECT population FROM world
WHERE name = 'Germany'
```

2. The query shows the name and population density

for each country where the area is over 5,000,000 km². Population density is not a column in the World table, but we

can calculate it as `population/area`.

Modify it to show the name and per capita gdp: `gdp/population` for each country where the area is over 5,000,000 km²

```
SELECT name, population/area FROM world
```

```
WHERE area > 5000000
```

```
SELECT name, gdp/population FROM world
```

```
WHERE area > 5000000
```

3. 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 'Luxembourg', 'Mauritius' and 'Samoa'. Show the **name** and the **population** for 'Ireland', 'Iceland' and 'Denmark'.

```
SELECT name, population FROM world
```

```
WHERE name IN ('Luxembourg', 'Mauritius', 'Samoa');
```

```
SELECT name, population FROM world
```

```
WHERE name IN ('Iceland', 'Denmark');
```

4.

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
```

nobel		
yr	subject	winner
1960	Chemistry	Willard F. Libby
1960	Literature	Saint-John Perse
1960	Medicine	Sir Frank Macfarlane Burnet
1960	Medicine	Peter Madawar
1950	Chemistry	Kurt Alder
1950	Chemistry	Otto Diels
1950	Literature	Bertrand Russell
1950	Medicine	Philip S. Hench
1950	Medicine	Edward C. Kendall
1950	Medicine	Tadeus Reichstein
1950	Peace	Ralph Bunche
1950	Physics	Cecil Powell

1.

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

```
SELECT yr, subject, winner
FROM nobel
WHERE yr = 1950
```

2.

Show who won the 1962 prize for Literature.

```
SELECT winner
FROM nobel
WHERE yr = 1962
AND subject = 'Literature'
```

3.

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

```
SELECT yr, subject
FROM nobel
WHERE winner = 'Albert Einstein'
```

4.

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

```
SELECT winner  
FROM nobel  
WHERE yr >= 2000  
AND subject = 'Peace'
```

5.

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

```
SELECT * FROM nobel  
WHERE yr BETWEEN 1980 AND 1989  
AND subject IN ('Literature')
```