

# PHP & MySQL

## Bazy danych MySQL

### Wprowadzenie

# PHP & MySQL

## Baza danych

Tabela „osoby”					
id	imie	nazwisko	wiek	mail	kurs
1	Iwona	Andrzejewska	25	ia@przyklad.pl	HTML
2	Andrzej	Nowak	22	an@test.pl	PHP & MySQL
3	Jerzy	Pionek	32	jp@przyklad.pl	PHP & MySQL
4	Alina	Pasikonik	27	ap@adres.pl	Java Script
5	Tomasz	Kowal	24	tk@test.pl	HTML
6	Andrzej	Nowak	29	an@przyklad.pl	Java Script
7	Barbara	Zalewska	21	bz@adres.pl	PHP & MySQL
8					

- Baza danych składa się z jednej lub wielu tabel, każdy wiersz tabeli zawiera jeden rekord, czyli zestaw danych.
- Rekord składa się z pól, które mogą zawierać dane o wcześniej zdefiniowanym typie.
- Odczytywanie danych z bazy odbywa się wierszami.
- Generalną zasadą jest unikanie powtórzeń, ponieważ mogą one prowadzić do wystąpienia błędów.

# PHP & MySQL

## Relacyjna baza danych

id	imie	nazwisko	wiek	mail	kurs_id
1	Iwona	Andrzejewska	25	ia@przyklad.pl	2
2	Andrzej	Nowak	22	an@test.pl	1
3	Jerzy	Pionek	32	jp@przyklad.pl	1
4	Alina	Pasikonik	27	ap@adres.pl	3
5	Tomasz	Kowal	24	tk@test.pl	2
6	Andrzej	Nowak	29	an@przyklad.pl	3
7	Barbara	Zalewska	21	bz@adres.pl	1
8					

id	kurs
1	PHP & MySQL
2	HTML
3	Java Script
4	CSS
5	

- Tabela powinna zawierać *klucz podstawowy*, który **jednoznacznie** identyfikuje każdy rekord.
- Klucz podstawowy z jednej tabeli może być użyty w innej tabeli, gdzie jest *kluczem obcym*.
- Tworzenie relacji między tabelami pozwala na unikanie powtórzeń w tabelach.
- Umieszczenie danych w wielu tabelach pozwala na łatwiejsze zarządzanie nimi.
- Istnieją 3 typy relacji: 1-1 (jeden do jednego), 1-N (jeden do wielu) i N-M (wiele do wielu).

# PHP & MySQL

## Klucz podstawowy

Tabela „osoby”				
id	imie	nazwisko	wiek	mail
1	Iwona	Andrzejewska	25	ia@przyklad.pl
2	Andrzej	Nowak	22	an@test.pl
3	Jerzy	Pionek	32	jp@przyklad.pl
4	Alina	Pasikonik	27	ap@adres.pl
5	Tomasz	Kowal	24	tk@test.pl
6	Andrzej	Nowak	29	an@przyklad.pl
7	Barbara	Zalewska	21	bz@adres.pl
8				

- Klucz podstawowy (*primary key*) jednoznacznie identyfikuje każdy rekord w tabeli.
- W kolumnie zdefiniowanej jako zawierająca klucz podstawowy wartości muszą być unikalne. Sprawdzanie powtórzeń wykonywane jest przez bazę MySQL.
- Dane do kolumny z kluczem mogą być wprowadzane automatycznie przez serwer bazy danych (*auto increment*). Dzięki temu raz użyta wartość nigdy nie zostanie powtórzona.
- Klucz podstawowy może składać się z większej ilości kolumn.
- W tabeli może istnieć tylko jeden klucz podstawowy.

# PHP & MySQL

## Relacja 1-1

Tabela „osoby”				
id	imie	nazwisko	wiek	mail
1	Iwona	Andrzejewska	25	ia@przyklad.pl
2	Andrzej	Nowak	22	an@test.pl
3	Jerzy	Pionek	32	jp@przyklad.pl
4	Alina	Pasikonik	27	ap@adres.pl
5	Tomasz	Kowal	24	tk@test.pl
6	Andrzej	Nowak	29	an@przyklad.pl
7	Barbara	Zalewska	21	bz@adres.pl
8				



Tabela „hasla”	
id	haslo
1	iwonka5
2	34rtano3
3	szachmat
4	romek
5	5kobalt
6	rakieta
7	alamakota
8	

# PHP & MySQL

## Relacja 1-N

Tabela „osoby”					
id	imie	nazwisko	wiek	mail	kurs_id
1	Iwona	Andrzejewska	25	ia@przyklad.pl	2
2	Andrzej	Nowak	22	an@test.pl	1
3	Jerzy	Pionek	32	jp@przyklad.pl	1
4	Alina	Pasikonik	27	ap@adres.pl	3
5	Tomasz	Kowal	24	tk@test.pl	2
6	Andrzej	Nowak	29	an@przyklad.pl	3
7	Barbara	Zalewska	21	bz@adres.pl	1
8					

Tabela „kursy”	
id	kurs
1	PHP & MySQL
2	HTML
3	Java Script
4	CSS
5	

- Najczęstszy typ relacji między tabelami to *jeden do wielu*.
- Klucz podstawowy z jednej tabeli jest użyty w innej tabeli jako klucz obcy.

# PHP & MySQL

## Relacja N-M

Tabela „osoby”				
id	imie	nazwisko	wiek	mail
1	Iwona	Andrzejewska	25	ia@przyklad.pl
2	Andrzej	Nowak	22	an@test.pl
3	Jerzy	Pionek	32	jp@przyklad.pl
4	Alina	Pasikonik	27	ap@adres.pl
5	Tomasz	Kowal	24	tk@test.pl
6	Andrzej	Nowak	29	an@przyklad.pl
7	Barbara	Zalewska	21	bz@adres.pl
8				

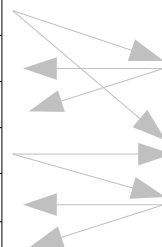
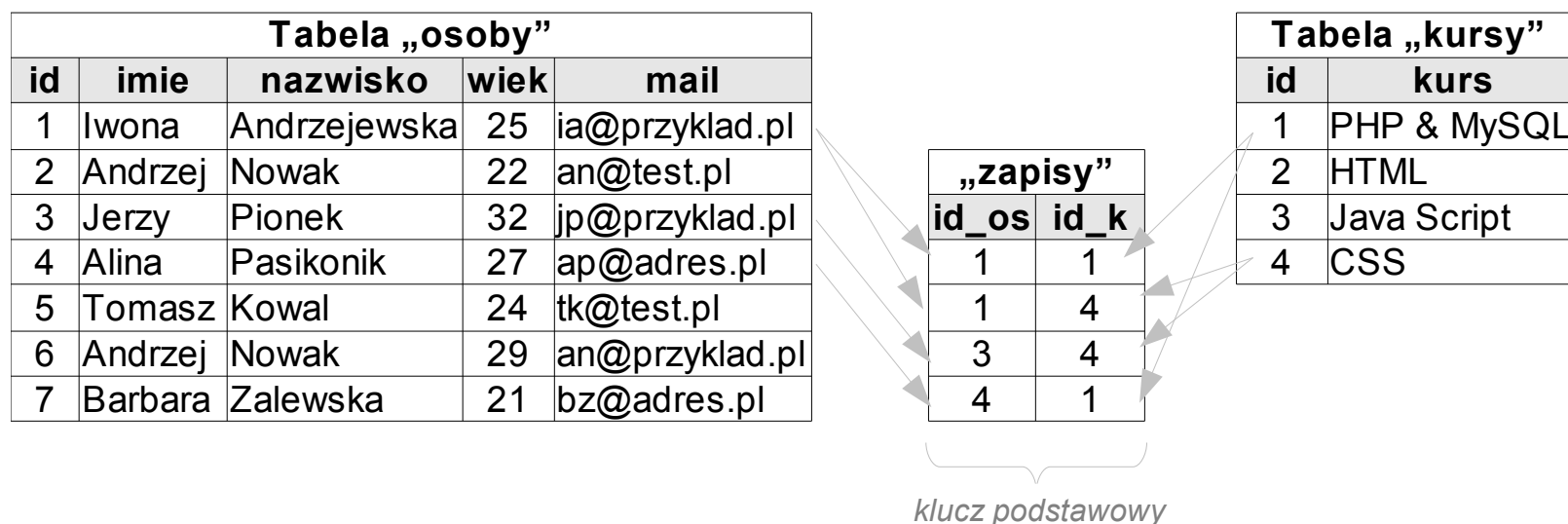


Tabela „kursy”	
id	kurs
1	PHP & MySQL
2	HTML
3	Java Script
4	CSS
5	

# PHP & MySQL

## Relacja N-M



- Relacja *wiele do wielu* realizowana jest za pośrednictwem dodatkowej tabeli.
- Oba klucze obce mogą składać się na klucz podstawowy tabeli „zapisy”, dzięki czemu można wykluczyć powtórzenie się tej samej kombinacji kluczy obcych.



# PHP & MySQL

## Najważniejsze typy danych

### Łańcuchy znaków

<b>CHAR (n)</b>	łańcuch znaków o stałej długości n (0-255)
<b>VARCHAR (n)</b>	łańcuch znaków o zmiennej długości n (0-255) od MySQL 5.0.3 o zmiennej długości n (0-65535)
<b>TINYTEXT</b>	do 255 znaków
<b>TEXT</b>	do 65535 znaków
<b>MEDIUMTEXT</b>	do 16777215 znaków
<b>LONGTEXT</b>	do 4294967295 znaków

Typ **BLOB** zapamiętuje ciągi binarne (np. obrazy, muzyka itd.)

<b>TINYBLOB</b>	do 255 znaków
<b>BLOB</b>	do 65535 znaków
<b>MEDIUMBLOB</b>	do 16777215 znaków
<b>LOBLOB</b>	do 4294967295 znaków

# PHP & MySQL

## Najważniejsze typy danych

### Liczby całkowite

Typ	Minimalna wartość (ze znakiem/bez znaku)	Maksymalna wartość (ze znakiem/bez znaku)
<b>TINYINT</b>	-128	127
	0	255
<b>SMALLINT</b>	-32768	32767
	0	65535
<b>MEDIUMINT</b>	-8388608	8388607
	0	16777215
<b>INT</b>	-2147483648	2147483647
	0	4294967295
<b>BIGINT</b>	-9223372036854775808	9223372036854775807
	0	18446744073709551615

- Do zdefiniowania liczby bez znaku używa się słowa kluczowego UNSIGNED
- Typ **BOOLEAN** (prawda/fałsz) jest realizowany jako liczba typu TINYINT:  
1 = prawda, 0 = fałsz

# PHP & MySQL

## Najważniejsze typy danych

### Liczby zmiennoprzecinkowe

Typy zmiennoprzecinkowe używane są do przechowywania liczb ułamkowych.

#### **FLOAT**

liczba zmiennoprzecinkowa o pojedynczej precyzji, zakres:

$\pm 1.175\text{E}-38$

$\pm 3.403\text{E}+38$

#### **DOUBLE**

liczba zmiennoprzecinkowa o podwójnej precyzji, zakres

$\pm 2.225\text{E}-308$

$\pm 1.798\text{E}+308$

#### **DECIMAL**

liczba zmiennoprzecinkowa przechowywana jako tekst (CHAR), używa się ją np. do przechowywania cen.

Użycie: DECIMAL(4,2) pozwoli na przechowywanie wartości posiadających 4 znaki z lewej strony przecinka i dwa po przecinku, np. 1240.55; zakres w tym przypadku: -9999.99 do 9999.99

# PHP & MySQL

## Najważniejsze typy danych

Data i czas

**DATETIME**

data w formacie YYYY-MM-DD HH:MM:SS

**DATE**

data w formacie YYYY-MM-DD

**TIME**

godzina w formacie HH:MM:SS

# PHP & MySQL

## Baza danych KALORIE

data	ilosc	produkt	kcal
2008-04-01	120	Chleb	180
2008-04-01	50	Jabłko	50
2008-04-01	120	Chleb	180
2008-04-02	250	Pepsi Cola	90
2008-04-02	60	Ser żółty	220

### PROBLEMY

- brak jednoznacznej identyfikacji rekordów
- baza w tej formie nie jest zoptymalizowana, ponieważ występują powtórzenia

# PHP & MySQL

## Baza danych KALORIE

Tabela „spozycie”			
id	data	ilosc	id_produktu
1	2008-04-01	120	2
2	2008-04-01	50	1
3	2008-04-01	120	2
4	2008-04-02	250	6
5	2008-04-02	60	3

Tabela „produkty”		
id	nazwa	kcal
1	Jabłko	45
2	Chleb	200
3	Ser żółty	315
4	Pomidor	15
5	Bigos	120
6	Pepsi Cola	45



- każda tabela ma własny klucz podstawowy
- relacja 1-N między tabelą „produkty” a tabelą „spozycie”

# PHP & MySQL

## Najważniejsze komendy SQL

Wybieranie danych z tabeli

*SELECT kolumny FROM tabela;*

SELECT \* FROM produkty;

SELECT nazwa, kcal FROM produkty;

SELECT id, data, ilosc FROM spozycie;

SELECT 15\*5;

# PHP & MySQL

## Najważniejsze komendy SQL

Wybieranie danych z tabeli

*SELECT kolumny FROM tabela WHERE warunek;*

```
SELECT * FROM produkty WHERE id = 10;
```

```
SELECT nazwa, kcal FROM produkty WHERE nazwa = 'Pomidor';
```

```
SELECT id, data, ilosc FROM spozycie WHERE data < '2008-04-01' AND ilosc > 50;
```

```
SELECT * FROM spozycie WHERE data != '2008-04-01';
```



# PHP & MySQL

## Najważniejsze komendy SQL

Wybieranie danych z tabeli

### Operatory dla WHERE

**Podstawowe operatory:** =, <, >, <=, >=, <> (lub !=), IS NULL, IS NOT NULL

SELECT \* FROM produkty WHERE nazwa IS NULL;

**BETWEEN**

SELECT \* FROM produkty WHERE id BETWEEN 10 AND 15;

**IN** oraz **NOT IN**

SELECT \* FROM produkty WHERE id IN (10, 20, 30);

**LIKE** oraz **NOT LIKE**

SELECT \* FROM produkty WHERE nazwa LIKE '%dżem%';

Znak % zastępuje dowolną liczbę dowolnych znaków.  
Znak \_ zastępuje jeden znak.

# PHP & MySQL

## Najważniejsze komendy SQL

Manipulacja wynikami wyszukiwania

Sortowanie:

**ORDER BY**

```
SELECT * FROM produkty WHERE id BETWEEN 1 AND 15  
ORDER BY nazwa ASC;
```

```
SELECT * FROM spozycie ORDER BY data DESC;
```

Pomijanie duplikatów:

**DISTINCT**

```
SELECT DISTINCT data FROM spozycie;
```

Ograniczanie ilości wybieranych rekordów:

**LIMIT**

```
SELECT * FROM spozycie ORDER BY data DESC LIMIT 1;
```

# PHP & MySQL

## Najważniejsze komendy SQL

Wyszukiwanie danych w wielu tabelach

Tabela „spozycie”			
id	data	ilosc	id_produktu
1	2008-04-01	120	2
2	2008-04-01	50	1
3	2008-04-01	80	2
4	2008-04-02	250	6
5	2008-04-02	60	2

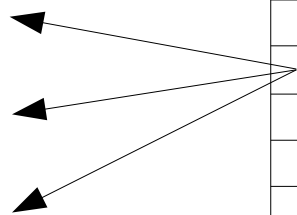


Tabela „produkty”		
id	nazwa	kcal
1	Jabłko	45
2	Chleb	200
3	Ser żółty	315
4	Pomidor	15
5	Bigos	120
6	Pepsi Cola	45

```
SELECT data, ilosc, nazwa, kcal FROM spozycie, produkty  
WHERE spozycie.id_produktu = produkty.id;
```

Jest to połączenie zwane „equi-join”.

# PHP & MySQL

## Najważniejsze komendy SQL

Wyszukiwanie danych w wielu tabelach

```
SELECT data, ilosc, nazwa, kcal FROM spozycie, produkty  
WHERE spozycie.id_produktu = produkty.id;
```

wynik:

data	ilosc	nazwa	kcal
2008-04-01	120	Chleb	200
2008-04-01	50	Jabłko	45
2008-04-01	80	Chleb	200
2008-04-02	250	Pepsi Cola	45
2008-04-02	60	Chleb	200

# PHP & MySQL

## Najważniejsze komendy SQL

Wstawianie danych do tabeli

*INSERT INTO tabela (kolumna1, kolumna2,...)  
VALUES (wartość1, wartość2,...);*

```
INSERT INTO produkty VALUES ('','Kapusta',20);
```

```
INSERT INTO produkty (nazwa, kcal) VALUES ('Kapusta',20);
```

```
INSERT INTO produkty SET nazwa = 'Kapusta', kcal = 20;
```

# PHP & MySQL

## Najważniejsze komendy SQL

Zmiana danych w tabeli

*UPDATE tabela SET kolumna1 = wartość1,...  
WHERE warunek ;*

UPDATE produkty SET nazwa = 'Kapusta biała' WHERE id = 202;

UPDATE telewizory SET cena = cena \*1.2;

# PHP & MySQL

## Najważniejsze komendy SQL

Usuwanie danych z tabeli

`DELETE FROM tabela WHERE warunek;`

`DELETE FROM produkty;`

`DELETE FROM produkty WHERE id = 202;`

**UWAGA:** Dane usuwane są nieodwracalnie!

# PHP & MySQL

## Najważniejsze komendy SQL

Funkcje agregujące

**COUNT, SUM, AVG, MIN, MAX**

```
SELECT MAX(kcal) FROM produkty;
```

```
SELECT COUNT(id) AS ile FROM produkty WHERE nazwa LIKE '%dżem%';
```