



Oprogramowanie i narzędzia internetowe

Bazy danych MySQL

dr inż. Jacek Czerniak

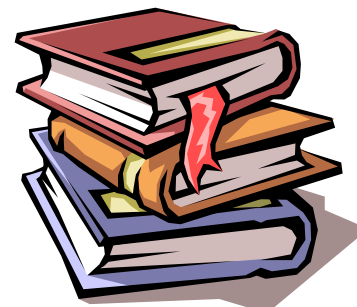
SQL



klient



motor
bazy
danych



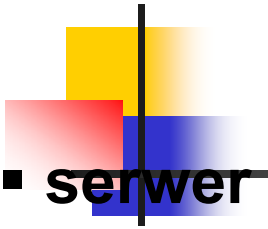
baza danych

Zalety systemu MySQL



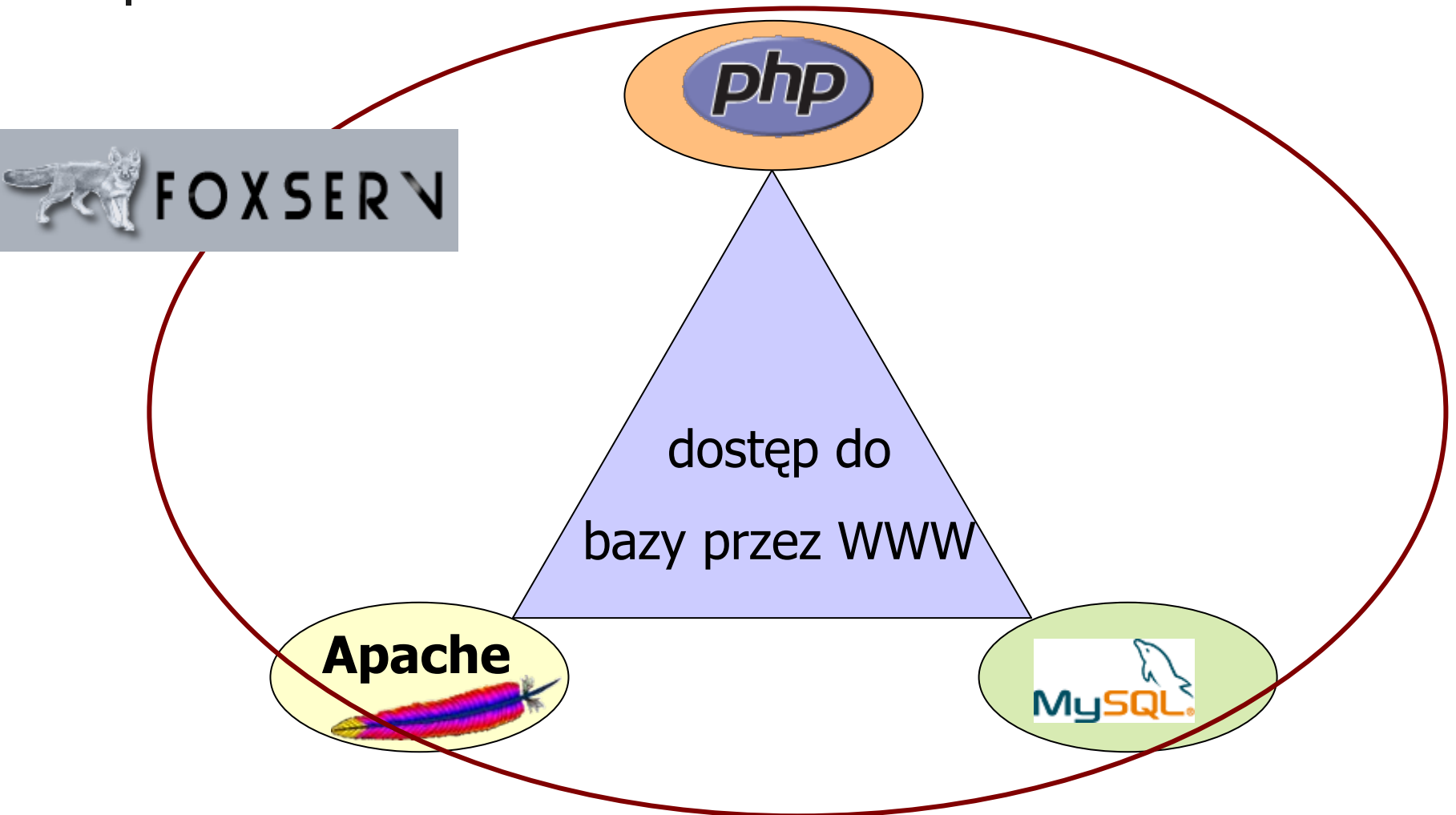
- **szybkość** – MySQL to najszybszy, dostępny SZBD
- **łatwość użycia** - względnie prosty nieskomplikowana instalacja
- **koszt** – bezpłatna dystrybucja systemu via Internet
- **obsługa języka zapytań** – do komunikacji pomiędzy klientem a serwerem wykorzystuje SQL dostęp do baz również poprzez ODBC (*Open Database Connectivity*)
- **możliwości** – równoczesne łączenie się wielu klientów z serwerem możliwość równoczesnej pracy z wieloma bazami danych
- **wykorzystanie sieci i bezpieczeństwo** – w pełni sieciowy produkt zapewniający dostęp do baz danych z każdego miejsca w sieci WWW; wbudowane mechanizmy kontroli dostępu
- **przenośność** – działa na różnych platformach systemowych i sprzętowych

Dystrybucja systemu MySQL

- 
- **serwer SQL**
 - **programy klienta** - udostępniają zasoby serwera
 - **biblioteka klienta** - umożliwia pisanie własnych programów
 - **isamchk, myisamchk** - tabele, klucz, indeksy
 - **myisampack, pack_isam** - skompresowane tabele read-only
 - **mysql** - wysyłanie zapytań do serwera MySQL
 - **mysqlaccess** - testowanie uprawnień dostępu
 - **mysqladmin** - przeprowadzanie czynności administracyjnych
 - **mysqlbug** - generowanie raportów o błędach
 - **mysqld** - serwer MySQL
 - **mysqldump** - składowanie zawartości tabel bazy danych
 - **mysqlimport** - masowe ładowanie danych do tabel
 - **mysqlshow** - informacje o bazach danych i tabelach
 - **safe_mysqld** - uruchamianie i monitorowanie serwera MySQL



Omne Trinum Perfectum ;-)



Podstawy instalacji MySQL

- **setup.exe** – po tymczasowym umieszczeniu plików
- **C:\mysql** – domyślna kartoteka
- **D:\prog\mysql\bin\mysqld – basedir D:\prog\mysql**
– system przeniesiony do: D:\prog\mysql
- **mysqld** – pełne info dla debuggera automatyczna alokacja pamięci
- **mysqld-opt** – optymalizacja dla Pentium
- **Win 95/98/Me** – w tle:
 - start: C:\mysql\bin\mysqld-opt
 - stop: C:\mysql\bin\mysqladmin –u root shutdown
- **Win NT/2000/XP** – usługa:
 - instalacja: C:\mysql\bin\mysqld-nt –install
 - start: NET START mysql
 - stop: NET STOP mysql



SQL

- **SELECT** – wybór informacji z bazy danych do przetwarzania
- **UPDATE** – poprawienie informacji w poszczególnych wierszach
- **DELETE** – usunięcie informacji z bazy danych (*brak kryteriów – usunięcie wszystkich wierszy*)
- **INSERT** – wstawienie informacji do tabeli



SELECT

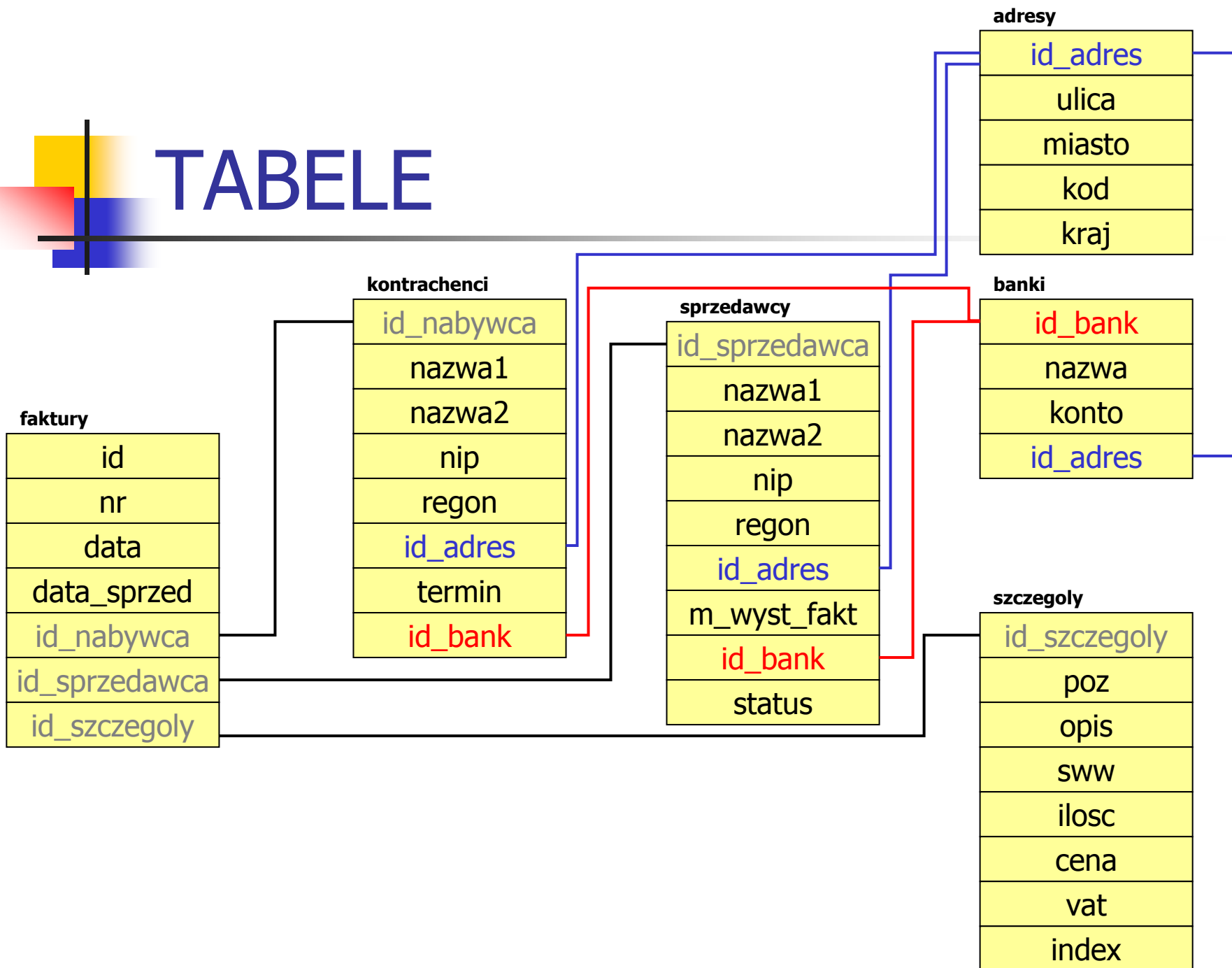
- SELECT /informacja/ FROM /tabela/ WHERE /warunki/

informacja – kolumny tabeli (np.: nazwisko, imię, * - wszystkie)

tabela – nazwa tabeli bazy danych (np.: person), *jako dodatkowy parametr może wystąpić alias pod jakim znana będzie ta tabela w zapytaniu np.: person p*

warunki – ograniczenia zapytania (np.: nazwisko='Kowalski', imię='Jan'), *może wystąpić również znak maski '%' zastępujący dowolny ciąg znaków z formułą LIKE np.: nazwisko LIKE 'Kowa%'*

TABELE





SELECT

SELECT f.nr_f, f.data_f, f.data_sprzed, k.nazwa AS knazwa,
f.id_szczegoly, f.id_nabywca

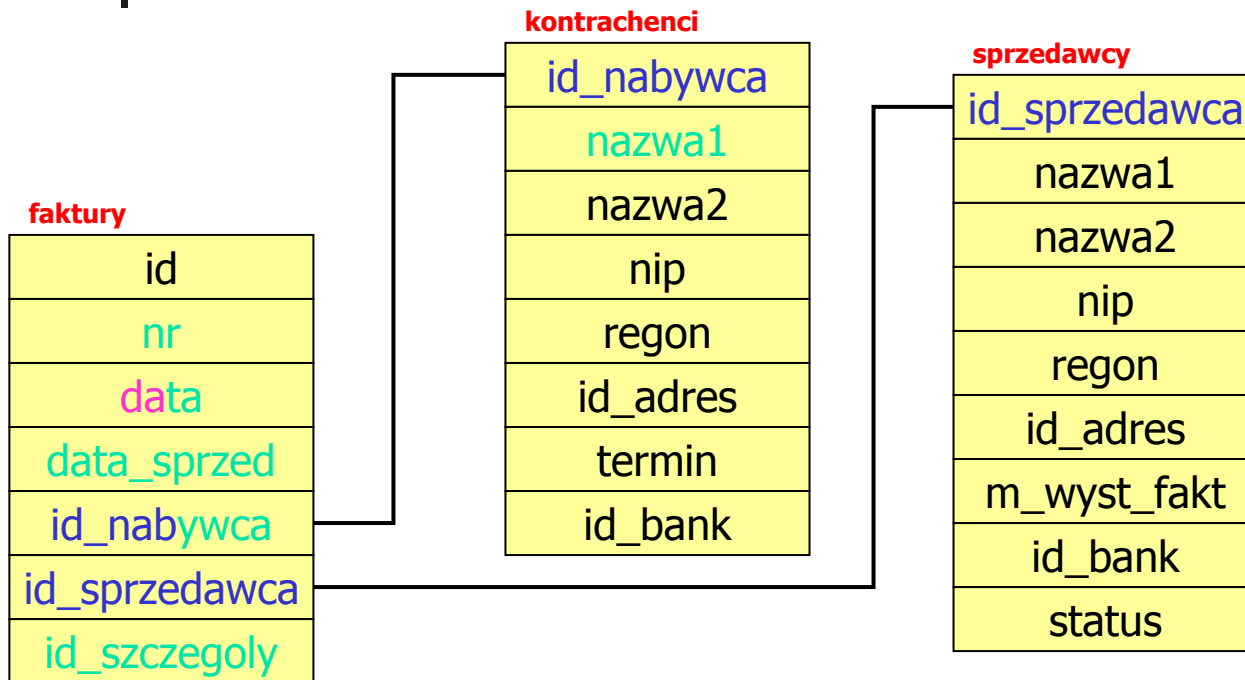
FROM faktury f

LEFT JOIN kontrahenci k **ON** k.id_nabywca=f.id_nabywca

LEFT JOIN sprzedawcy s **ON** s.id_sprzedawca=f.id_sprzedawca

ORDER BY f.data_f **DESC**

TABELE





UPDATE

- UPDATE /tabela/ SET /informacja+wartość/ WHERE /warunki/

tabela – nazwa tabeli bazy danych

informacja+wartość – zmieniane dane (np.: nazwisko='Kowalski')

warunki – określenie warunku zmiany (np.: nazwisko ='kowalski')



DELETE

- DELETE FROM /tabela/ WHERE /warunek/

tabela – nazwa tabeli bazy danych

warunek – określenie zakresu usuwania wierszy (np.:
nazwisko='nowak')



INSERT

- `INSERT INTO /tabela/ (/kolumny/,...) VALUES (/wartości/,...)`

tabela – nazwa tabeli bazy danych

kolumny – nazwy kolumn dane tabeli (np.: nazwisko, imie, adres)

wartości – wartości do wpisania zgodnie z kolejnością kolumn (np.:
'Kowalski','Jan','ul. Dąbrowskiego 10/2')



Połączenie

PostgreSQL

```
$conn_string = "host=sheep port=5432 dbname=test user=lamb password=bar";  
$dbconn4 = pg_connect ($conn_string);
```

MySQL

```
$link = mysql_connect('sheep', 'lamb', 'bar') or die('Nie można się połączyć: ' . mysql_error());
```

Otwiera połączenie z bazą "test" na komputerze "sheep" z użyciem nazwy użytkownika i hasła



Zapytanie

PostgreSQL

```
$query = " SELECT * FROM tabela WHERE 1";  
$result = pg_query ($query);
```

MySQL

```
$query = " SELECT * FROM tabela WHERE 1";  
$result = mysql_query($query);
```

Wykonuje zapytanie \$query i zwraca wynik w \$result



Liczba wierszy

PostgreSQL

```
$ile = pg_num_rows ($result);
```

MySQL

```
$ile = mysql_num_rows($result);
```

Informacja o ilości wierszy w zapytaniu



Przekształcenie w tablicę

PostgreSQL

```
$x = pg_fetch_array ($result, $wiersz);
```

MySQL

```
$x = mysql_fetch_array($result);
```

*Przekształcenie wyniku zapytania \$result w tablicę kolumn \$x (np.: \$x[„nazwisko”])
aktualnego wiersza*



Zamknięcie połączenia

PostgreSQL

```
$OK = pg_close ($conn_string);
```

MySQL

```
$OK = mysql_close($link);
```

Zamyka połączenie z bazą danych \$link, \$conn_string zwracając wynik zamknięcia w \$OK



Programowanie stron WWW

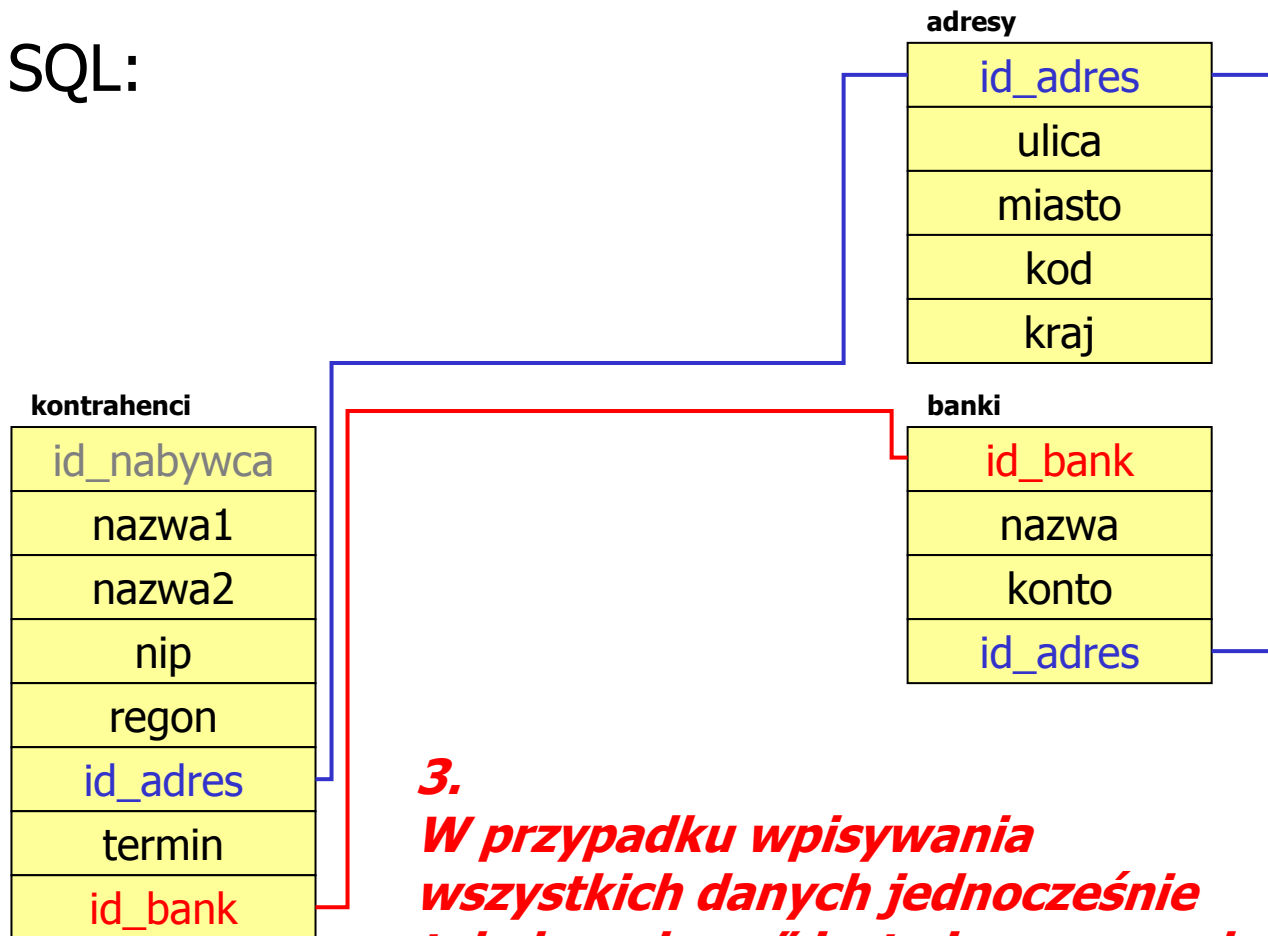
Wypełnianie tabel

SQL

Tabele

Dostępne polecenia SQL:

- SELECT
- INSERT
- DELETE
- UPDATE



3.
***W przypadku wpisywania
wszystkich danych jednocześnie
tabela „adresy” jest pierwszorzędną***

SQL

■ Kolejność

- Wpis adresu **banku** do tabeli **adresy**
- Wpis adresu **kontrahenta** do tabeli **adresy**
- Wpis danych o **banku** z parametrem **id_adres** do tabeli **banki**
- Wpis danych o **kontrahencie** z parametrem **id_adres** (*kontrahenta*) **id_bank** do tabeli **kontrahenci**

kontrahenci

id_nabywca
nazwa1
nazwa2
nip
regon
id_adres
termin
id_bank

adresy

id_adres
ulica
miasto
kod
kraj

banki

id_bank
nazwa
konto
id_adres

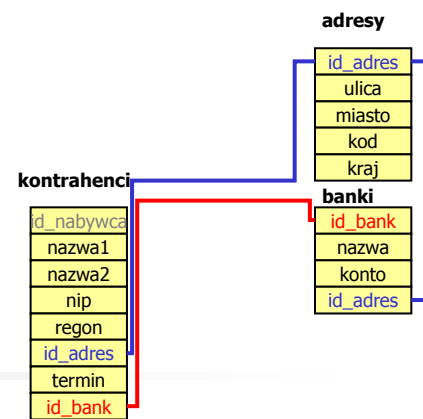


SQL

- Klucze bazy i identyfikatory nie powinny być tworzone na podstawie wprowadzanych danych.
- W czasie wpisywania danych należy wykonać dwie operacje aby uzyskać niezbędne informacje do wpisania w kolejnej tabeli



SQL adresy(kontrahent)



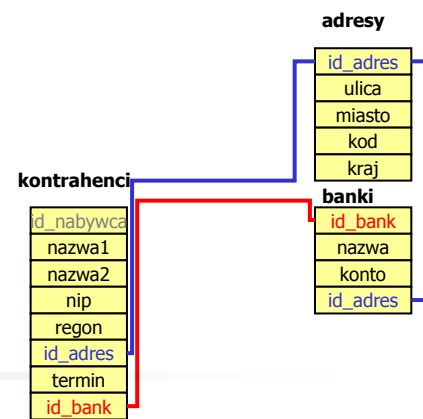
Zakładamy adres kontrahenta w tabeli adresy i sprawdzamy przydzielony identyfikator

INSERT INTO adresy (ulica, miasto, kod, kraj) **VALUES** ('ul. Kochanowskiego 15/5', 'Szczecin', '70-000', 'Polska');

SELECT id_adres **FROM** adresy **WHERE** ulica='ul. Kochanowskiego 15/5' **AND** miasto='Szczecin' **AND** kod='70-000' **AND** kraj='Polska';

SELECT id_adresy **FROM** adresy **ORDER BY** id_adresy **DESC LIMIT** 1;

SQL adresy(bank)



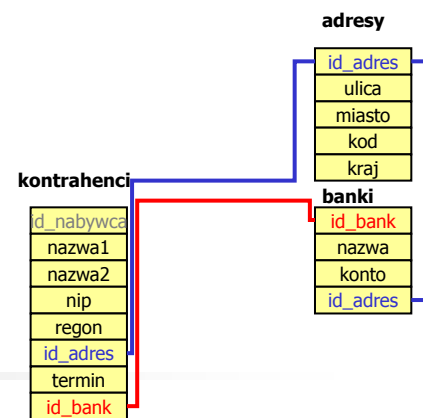
Zakładamy adres banku w tabeli adresy i sprawdzamy przydzielony identyfikator

```
INSERT INTO adresy (ulica, miasto, kod, kraj) VALUES ('ul. Mazowiecka 1/1', 'Szczecin', '70-001', 'Polska');
```

```
SELECT id_adres FROM adresy WHERE ulica='ul. Mazowiecka 1/1' AND miasto='Szczecin' AND kod='70-001' AND kraj='Polska';
```

```
SELECT id_adresy FROM adresy ORDER BY id_adresy DESC LIMIT 1;
```

SQL banki(kontrahent)



Zakładamy informacje o banku w tabeli banki z identyfikatorem adresu z tabeli adresy

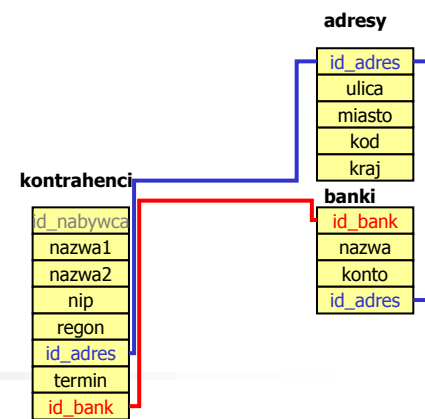
```
INSERT INTO banki (nazwa, konto, id_adres) VALUES ('PKO SA', '12345678-2020202020', $id_adres_banku);
```

```
SELECT id_bank FROM banki WHERE nazwa='PKO SA' AND  
konto='12345678-2020202020' AND id_adres=$id_adres_banku;
```

```
SELECT id_bank FROM banki ORDER BY id_bank DESC LIMIT 1;
```



SQL kontrahent(kontrahent)



Zakładamy informacje o kontrahencie w tabeli kontrahenci z identyfikatorem adresu z tabeli adresy i identyfikatorem banku z tabeli banki

INSERT INTO kontrahenci

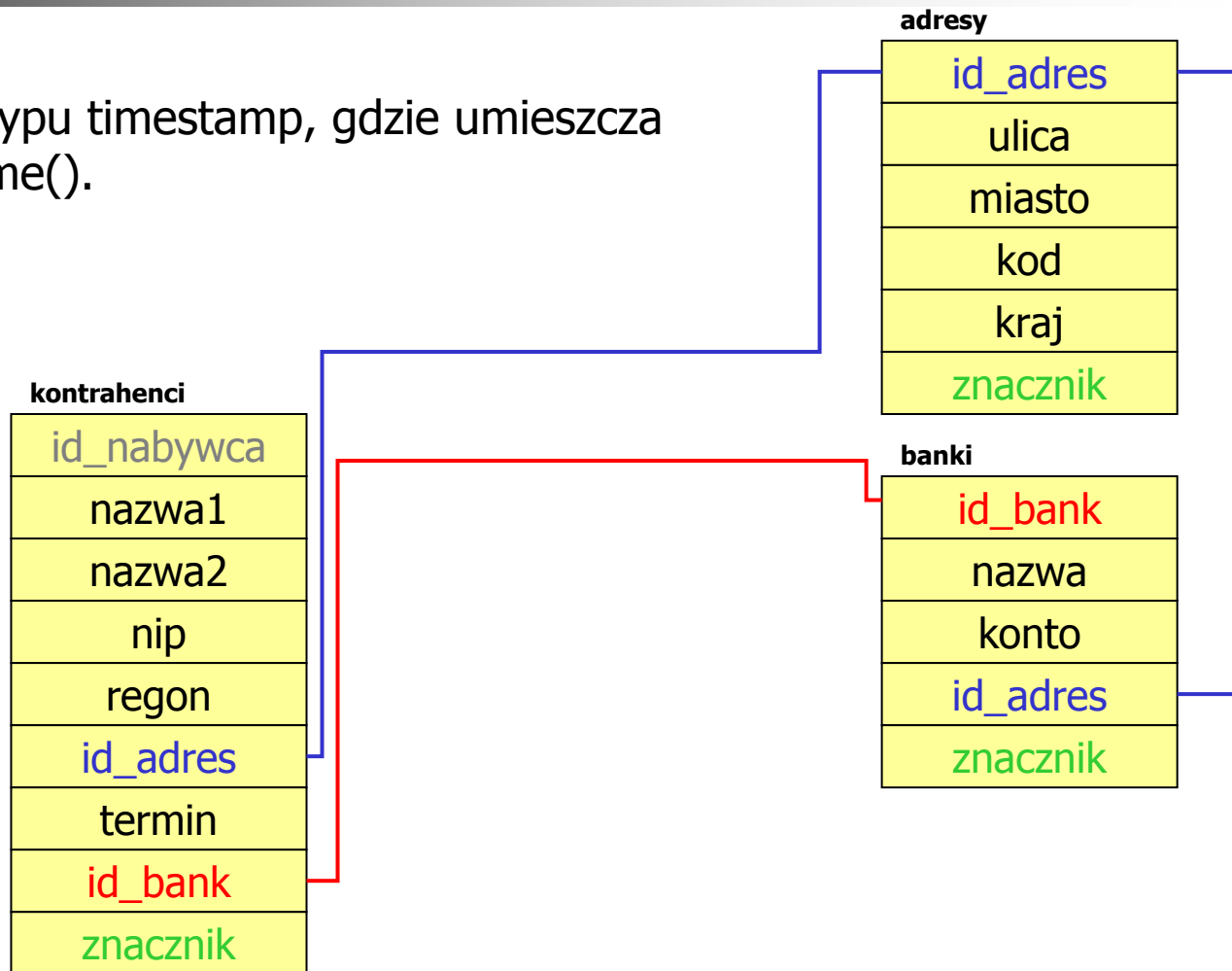
(nazwa1, nazwa2, nip, regon, id_adres, termin, id_bank)

VALUES

('Firma Budowlana', 'Kowalski Jan & Syn', '999-999-99-99', '',
\$id_adres_kontrahenta, '14 dni', \$id_bank_kontrahenta);

Metoda znacznika

Znacznik – najlepiej typu timestamp, gdzie umieszcza się wartość funkcji time().





SQL adresy(kontrahent, znacznik)

Zakładamy adres kontrahenta w tabeli adresy i sprawdzamy przydzielony identyfikator

```
$znacznik = time();
```

```
INSERT INTO adresy (ulica, miasto, kod, kraj, znacznik) VALUES ('ul.  
Kochanowskiego 15/5', 'Szczecin', '70-000', 'Polska', $znacznik);
```

```
SELECT id_adres FROM adresy WHERE znacznik=$znacznik;
```



SQL adresy(bank)

Zakładamy adres banku w tabeli adresy i sprawdzamy przydzielony identyfikator

```
INSERT INTO adresy (ulica, miasto, kod, kraj, znacznik) VALUES ('ul.  
Mazowiecka 1/1', 'Szczecin', '70-001', 'Polska', $znacznik);
```

```
SELECT id_adres FROM adresy WHERE znacznik=$znacznik;
```



SQL banki(kontrahent)

Zakładamy informacje o banku w tabeli banki z identyfikatorem adresu z tabeli adresy

```
INSERT INTO banki (nazwa, konto, id_adres, znacznik) VALUES ('PKO SA',  
'12345678-2020202020', $id_adres_banku, $znacznik);
```

```
SELECT id_bank FROM banki WHERE znacznik=$znacznik;
```



SQL kontrahent(kontrahent)

Zakładamy informacje o kontrahencie w tabeli kontrahenci z identyfikatorem adresu z tabeli adresy i identyfikatorem banku z tabeli banki

INSERT INTO kontrahenci

(nazwa1, nazwa2, nip, regon, id_adres, termin, id_bank, znacznik)

VALUES

('Firma Budowlana', 'Kowalski Jan & Syn', '999-999-99-99', '',
\$id_adres_kontrahenta, '14 dni', \$id_bank_kontrahenta, \$znacznik);

Pierwsze kroki z MySQL (1)

■ logowanie

```
mysql -h komp -u user -p
```

- p – wymaganie hasła, serwer o nie zapyta

■ tworzenie bazy

```
mysql> create database nazwa_bazy;
```

- każde polecenie musi kończyć się średnikiem

■ wskazanie bazy aktywnej

```
mysql> use nazwa_bazy
```

- po instalacji MySQL dostępne są 2 bazy: mysql – systemowa i test - testowa

■ przywileje

```
GRANT przywileje [kolumny]  
ON obiekt  
TO ident-user [IDENTIFIED BY 'hasło']  
[WITH GRANT OPTION]
```

```
REVOKE przywileje  
ON obiekt  
TO ident-user
```

Pierwsze kroki z MySQL (2)

■ nowy użytkownik

```
use mysql;  
insert into user (host, user, password) values  
( 'localhost', 'ident', 'hasło' );  
grant all  
on *.*  
to ident@... identified by hasło  
with grant option;  
flush privileges;
```

Usunąć:
reload, shutdown,
process, file

■ tworzenie tabeli danej bazy danych

```
create table nazwa_tabeli (kolumny);  
create table książki  
(isbn char(13), autor char(30), tytuł char(60), cena float(4.2));
```

■ prezentacja konstrukcji

```
mysql> show tables;  
mysql> show databases;  
mysql> describe nazwa_tablicy;
```

Pierwsze kroki z MySQL (3)

■ zapisywanie danych do bazy

```
insert into nazwa_tabeli (kl1, kl2,...) values (wr1, wr2,...);
```

```
insert into klienci values  
(NULL, 'Adam Małysz', 'Skoczków 2A', 'Wisła');
```

```
insert into klienci (nazwisko, adres, miasto) values  
( 'Adam Małysz', 'Skoczków 2A', 'Wisła');
```

```
insert into klienci  
set nazwisko = 'Adam Małysz',  
    adres = 'Skoczków 2A',  
    miasto = 'Wisła';
```

```
select pozycje  
from nazwy_tabel  
[where warunek]  
[group by rodzaj_grupowania]  
[having wartość_funkcji]  
[order by porządek_sortowania]  
[limit granica];
```

■ wyszukiwanie danych w bazie

Pierwsze kroki z MySQL (4)

■ wyszukiwanie danych w bazie - przykłady

```
select nazwisko, miasto  
from klienci;
```

```
select *  
from klienci;
```

```
select *  
from klienci  
where label=5;
```

```
select nazwisko  
from klienci  
limit 2, 3;
```

```
select avg(wartość)  
from zamówienia;
```

```
select klientid, avg(wartość)  
from zamówienia  
group by klientid;
```

```
select nazwisko, miasto  
from klienci  
having avg(wartość)>50;
```

```
select nazwisko, miasto  
from klienci  
group by klientid;
```

```
select *  
from klienci  
order by imie asc;
```

```
select *  
from klienci  
order by imie desc;
```

Pierwsze kroki z MySQL (5)

■ usuwanie rekordów z bazy

```
delete from nazwa_tabeli  
[where warunek]  
[limit ilość];
```

■ usuwanie całej tabeli, całej bazy danych

```
drop table nazwa_tabeli;  
drop database nazwa_bazy;
```

■ zmiana zawartości rekordu

```
update nazwa_tabeli  
set kol1=wyr1, kol2=wyr2,...  
[where warunek]  
[limit ilość];
```

```
update książki  
set cena=cena*1.2;
```

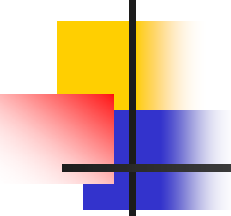
■ zmiana struktury tabeli

```
alter table klienci  
modify nazwisko char(45);
```

```
alter table zamówienia  
drop podatek;
```

```
alter table zamówienia  
add podatek float(6,2) after wartość;
```

Literatura

- 
-
- Beynon-Davies P. Systemy baz danych WNT Warszawa 1998
 - Date C.J., Darwen H., SQL. Omówienie standardu języka, WNT, Warszawa, 2000
 - DuBois P., MySQL, Wyd. MIKOM, Warszawa, 2000
 - Grochala M., Java, aplikacje bazodanowe, Wyd. Helion, 2001
 - Hilton C., Wilis J., PHP3. Internetowe aplikacje bazodanowe, Wyd. Helion, 2000
 - Meloni J., C., PHP. Podręcznik tworzenia stron WWW, Wyd. MIKOM, 2000
 - Welling L., Thomson L., PHP i MySQL. Tworzenie stron WWW, Wyd. Helion, 2002
 - M.Nowakowski, PHP-Nuke. Ćwiczenia, 2002, Helion



Warunki zaliczenia

- **Data: 12.06**
- **Godz. 14:30**
- Forma: test wyboru
- Warunek: wpis tylko po „*za!*” z laboratorium
- Bonus: „+1” do oceny dla osób z 5.0 z laboratoriów
- www.internet.z.pl



Dziękuję za uwagę!

