## Projektowanie baz danych za pomocą narzędzi CASE

Metody tworzenia systemów informatycznych w tym, także rozbudowanych baz danych są komputerowo wspomagane przez narzędzia CASE (*ang. Computer Aided Software Engineering*). Dzięki narzędziom CASE, proces tworzenia systemu staje się o wiele krótszy pociągając za sobą mniejsze koszty. Ponadto narzędzia CASE zapewniają wyższą jakość tworzonego systemu. Narzędzia CASE zawierają w sobie inżynierię dwustronną: *inżynieria do przodu* oraz *inżynieria wstecz*. Inżynieria do przodu jest niczym innym jak generacją kodu na podstawie modelu. Inżynieria wstecz to proces analizy oprogramowania, którego celem jest odtworzenie projektu i specyfikacji. W tym procesie program nie ulega zmianie. Kod źródłowy oprogramowania jest zwykle dostępny i stanowi daną wejściową dla procesu inżynierii wstecz. Celem inżynierii wstecz jest określenie projektu i specyfikacji systemu na podstawie kodu źródłowego.

Przykładem tego typu narzędzia jest program **DBDesigner4**. Jest to system wizualnego projektowania, modelowania i tworzenia baz danych. Program jest w całości darmowy i można go pobrać ze strony internetowej: <u>http://fabforce.net/index.php</u>. DBDesigner4 jest w wersji zarówno dla systemu Windows jak i Linux.

Pierwsza część ćwiczenia będzie pokazywała zastosowanie inżynierii do przodu, gdzie na podstawie diagramu ERD zostanie zaprojektowana baza danych a następnie wygenerowany zostanie kod źródłowy tej bazy w postaci skryptu SQL-a. Druga część będzie polegała na zastosowaniu inżynierii wstecz.

## 1. <u>Projektowanie bazy danych za pomocą DBDesigner4.</u>

Do modelowania danych wykorzystuje się tzw. diagramy związków encji (*ang. Entity Relationship Diagrams, ERD*). Podstawowymi pojęciami tego modelu są: encja i związek. Encja jest reprezentacją obiektu świata rzeczywistego identyfikowana za pomocą unikalnej nazwy i posiadająca pewne dla siebie charakterystyczne cechy. Przykładem encji może być np. osoba (wraz z cechami: imię, nazwisko, adres, numer PESEL). Relacje pomiędzy encjami ilustruje związek. Do najważniejszych cech związku należą unikalna nazwa oraz liczebność (1:1, 1:N, N:M). Celem tej części ćwiczenia będzie stworzenie diagramu ERD według schematu pokazanego na rysunku 1.



Rysunek 1. Diagram ERD przykładowej bazy danych.



Rysunek 2. Główne okno aplikacji DBDesigner4.

Główne okno programu **DBDesigner4** składa się z 5 obszarów: 1 - pasek narzędzi, 2 – obszar roboczy, 3 – okno nawigacji i informacji, 4 - okno typów danych, 5 – okno bieżącego modelu bazy danych.

Pracę z DBDesigner4 zaczynamy od utworzenia nowego projekty opcją *File*  $\rightarrow$  *New*. W celu dodania nowej tabeli do projektu wybieramy z paska narzędzi ikonę i klikamy na obszar roboczy. Kolejnym krokiem będzie edycja nowo utworzonej tabeli, w tym celu podwójnie klikamy na tabelę w obszarze roboczym. Efektem tego będzie pojawienie się okna edytora tabeli (rysunek 3). W edytorze wprowadzamy nazwę tabeli *tzawodnik* i kolejno kolumny tej tabeli zgodnie z tym, co jest pokazane na rysunku 4.

Column Name       Data Type       NN Al       Flags       Default Value       Comments         Image: Standard Inserts       Image: Standard	Table Editor Table Name Table_01	Table Prefix Default (no prefix)	Table Type MYISAM (Standard)		₩eak entity ♦ ☐ is n:m Table
Indices Indices Advanced Standard Inserts Comments Index Type: PRIMARY	Column Name	DataType NN	AI Flags	Default Value C	
	☐ Indices ☐ Table Options ☐ Advanced ☐ Standard Inserts ☐ Comments	Indices	Indexname Index Type: PRIMARY	Columns (use Drag'n	'Drop to add Columns)

Rysunek 3. Okno edytora tabeli.

able Name zawodnik	Table Prefix Default (no	prefix)	Table Type MYISAM (Standard)	• •	eak entity is n:m Table
Column Name	DataType	NN .	Al Flags D	efault Value Comments	
id_zawodnika	S INTEGER	-	🖊 🗹 UNSIGNED 🔛 ZEROFILL		
🔉 imie	送 VARCHAR(20)	~	BINARY		
🔉 nazwisko	🛃 VARCHAR(45)	1	BINARY		
data_urodz	<b>DATETIME</b>	~			
🔉 id_klub	🛃 INTEGER	-	UNSIGNED ZEROFILL		
🔉 id_kraj	🕵 INTEGER	~	🖌 UNSIGNED 🔄 ZEROFILL		
liczba_meczy	💽 INTEGER		UNSIGNED ZEROFILL		
liczba_goli	<b>INTEGER</b>		UNSIGNED ZEROFILL		
Indices Table Options	Indices		Indexname Colu	umps (use Dran'n'Drop to a	add Columns
Advanced Standard Inserts Comments	PRIMARY		PRIMARY id_	zawodnika	

Rysunek 4. Edytor tabeli - kolumny tabeli tzawodnik.

Program DBDesigner4 jest na tyle sprytny, że potrafi sam automatycznie wydedukować, że pierwsze pole tabeli powinno być kluczem głównym i mieć włączoną autoinkrementację. W podobny sposób postępujemy z pozostałymi tabelami: *tkraj, tklub, tfederacja*.

Po utworzeniu wszystkich niezbędnych tabel wchodzących w skład projektowanej bazy danych pozostaje nam powiązać je relacjami. DBDesigner4 obsługuje wszystkie spotykane w bazach danych relacje: jeden do jednego (1:1), jeden do wielu (1:N) oraz wiele do wielu (N:M). Każda z nich jest w pełni konfigurowalna: możemy wpłynąć na nazwy pól, etykiety oraz zachowanie się relacji w trakcie np. usuwania rekordów.

Przystąpmy do utworzenia relacji pomiędzy tabelami tzawodnik oraz tkraj. W celu dodania relacji 1:N z paska narzędzi wybieramy ikonę  $\mathbf{F}$  (zaraz pod ikoną *Nowa tabela*). Gdy narzędzie jest aktywne klikamy najpierw w tabelę *tkraj*, następnie w tabelę *tzawodnik*. W efekcie uzyskamy połączenie pomiędzy tabelami w postaci linii reprezentującej relacje typu 1:N. Ponadto w tabeli *tzawodnik* pojawiło się pole o nazwie *tzawodnik\_FKIndex1* będące kluczem obcym (*FK*). Teraz przystąpimy do edycji relacji, podwójne klikamy na utworzoną linię. Efektem tego będzie pojawienie się okna edytora relacji (rysunek 5).

Relation Editor	×					
Relation Name     Relation Kind     Visibil       zawodnik-kraj     1:n (Non-Identifying)     Visibil	ity ble 🗢					
Tables Source: tkraj Destination: tzawodnik						
Foreign Keys						
Source Column Dest. Name Comment id_kraj id_kraj						
Reference Definition						
On Delete: NO ACTION	<b>+</b>					
On Update: NO ACTION						
	< ≈					

Rysunek 5. Okno edytora relacji.

Nadajemy relacji nazwę "zawodnik-kraj" i zmieniamy nazwę "Dest. Name" na *id\_kraj*. Zatwierdzamy wprowadzone zmiany przyciskiem OK. Teraz pozostaje nam zdefiniować pozostałe relacje 1:N w podobny sposób:

- kraj klub (klikamy najpierw na tabelę *tkraj*, następnie *tklub*),
- zawodnik klub (klikamy najpierw na tabelę tklub, nastepnie tzawodnik),
- kraj federacja (klikamy najpierw na tabelę tfederacja, następnie tkraj).

W końcowym efekcie uzyskamy rezultat taki jak na rysunku 1. Zapisujemy nasz projekt do pliku opcją *File*  $\rightarrow$ *Save*. Tak zaprojektowaną bazę danych możemy za pomocą DBDesigner4 wyeksportować do skryptu SQL-a tworzącego wszystkie tabele. Z menu *File* wybieramy *Export*  $\rightarrow$  *SQL Create Script* pojawi się nam okno, w którym zaznaczamy opcje tak jak to jest pokazane na rysunku 6.

Export SQL Script	×
General Settings          Export selected Tables only         Order Tables by Foreign Keys	
SQL Creates Settings         Image: Define Primary Keys       Image: Output Table Options         Image: Define Primary Keys       Image: Output Standard Inserts         Image: Define Primary Keys       Image: Output Standard Inserts         Image: Define Primary Keys       Image: Output Standard Inserts         Image: Output Comments       Image: Output Comments	
Copy Script to Clipboard	
Create Indices Output Standard Inserts Define Foreign Key References when enabled in Relations' Editors Output Comments Copy Script to Clipboard Save Script to file SQL Creates saved to file bazz1.sql.	

Rysunek 6. Okno eksportu skryptu SQL.

Zapisujemy nasz skrypt naciskając "*Save Script to file*". Teraz możemy zobaczyć wygenerowany przez nas skrypt otwierając zapisany plik w notatniku (rysunek 7).

```
CREATE TABLE tfederacja (
   id_fedracja INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
   nazwa_federacji CHAR(10) NOT NULL,
PRIMARY KEY(id_fedracja)
);
CREATE TABLE tklub (
   id_klub INTEGER UNSIGNED NOT NULL AUTO_INCREMENT.
   tkraj_id_kraj INTEGER UNSIGNED NOT NULL,
nazwa_klubu VARCHAR(45) NOT NULL,
rok_powstania YEAR NOT NULL,
   id_kraj INTEGER UNSIGNED NOT NULL,
PRIMARY KEY(id_klub),
   INDEX tklub_FKIndex1(tkraj_id_kraj)
);
CREATE TABLE tkraj (
id_kraj INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
   id_fedracja INTEGER UNSIGNED NOT NULL,
   nazwa_kraju VARCHAR(45) NOT NULL,
   id_federacja INTEGER UNSIGNED NOT NULL,
PRIMARY KEY(id_kraj),
INDEX tkraj_FKIndex1(id_fedracja)
);
CREATE TABLE tzawodnik (
   id_zawodnika INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
   id_kraj INTEGER UNSIGNED NOT NULL,
imie VARCHAR(20) NOT NULL,
nazwisko VARCHAR(45) NOT NULL,
   data_urodz DATETIME NOT NULL,
   id_klub INTEGER UNSIGNED NOT NULL,
   liczba_meczy INTEGER UNSIGNED NULL,
liczba_goli INTEGER UNSIGNED NULL,
   PRIMARY KEY(id_zawodnika),
INDEX tzawodnik_FKIndex1(id_kraj),
INDEX tzawodnik_FKIndex2(id_klub)
);
```

Rysunek 7. Skrypt SQL.

## 2. Inżynieria wstecz z wykorzystaniem DBDesigner4.

Uruchamiamy MySQL i logujemy się do niego, następnie kolejno wykonujemy polecenia:

- o create database <nazwa\_bazy>;
- o use <nazwa\_bazy>;
- o source <nazwa\_skryptu>;

gdzie *<nazwa\_skryptu>* jest to nazwa naszego zapisanego skryptu z pierwszej części ćwiczenia wraz z pełną ścieżką dostępu do niego. W ten sposób załadowaliśmy nasz projekt bazy ze wszystkimi tabelami do MySQL-a.

🛤 MySQL Command Line Client	- 🗆	×
mysql> create database baza_DBDesigner4; Query OK, 1 row affected (0.00 sec)		
mysql> use baza_DBDesigner4; Database changed mysql> source D:baza2.sql; Query OK, Ø rows affected (0.02 sec)		
Query OK, Ø rows affected (0.03 sec)		
Query OK, Ø rows affected (0.02 sec)		
Query OK, Ø rows affected (0.02 sec)		
mysql> show tables;		
Tables_in_baza_dbdesigner4		
tfederacja tklub tkraj tzawodnik		
4 rows in set (0.00 sec)		
mysql>		
		•

Rysunek 8. Przykład poprawnego wykonania stworzonego skryptu.

Teraz zamykamy MySQL-a, natomiast otwieramy DBDesigner4. Z menu "*Database*" wybieramy opcje "*Connect to Database*", co spowoduje pojawienie się okna wyboru połączenia.

Select Database Connection							
Network Hosts	All Database Connectio	ns					
All Connections  All Connections  MySQL  Syl Localhost  Syl Localh	Connection	Туре	Host	Database	Description		
🖻 🧰 ODBC	Connection: DBD Username: root	esignerMys	sql Passwo	ord: xxxxxxxx		nnect	

Rysunek 9. Okno wyboru połączenia z bazą danych.

Naciskamy "New Database Connection" powinno się pojawić nowe okno, w którym wpisujemy nazwę połączenia, ustalamy parametry tego połączenia, określamy nazwę bazy

danych, z którą chcemy się połączyć oraz podajemy nazwę użytkownika i hasło. Rysunek 10 przestawia przykładowe ustawienia dla bazy danych o nazwie *"baza\_DBDesigner4"*. Ustawienia zatwierdzamy przyciskiem OK. W ten sposób zdefiniowaliśmy połączenie DBDesigner4 z MySQL-em. Zamykamy pozostałe okna.

Data	abase Connection Editor	×
	General Advanced	
	Connection Name: DBDesigner2MySQL Driver: MySQL \$	
	Connection Settings	
	Hostname: Localhost  Port: 3306	
	Database Name: baza_DBDesigner4	
	Username: root Password: *******	
	Description:	
	🔀 Abort 🗸 OK	

Rysunek 10. Okno edytora ustawienia połączeń.

Z menu "*DataBase*" wybieramy opcje "*Reverse Engineering*" spowoduje to pojawienie się okienka (rysunek 11), mamy w nim już zdefiniowane jedno połączenie, które wybieramy i naciskamy "*Connect*". Wtedy DBdesigner4 powinien się połączyć z MySQL i z naszą bazą danych "*baza\_DBDesigner4*". Jeśli udało się połączyć z naszą bazą danych pojawi się okienko "Reverse Engineering" (rysunek 12), w którym zaznaczamy opcję "*Build Relations*". W sytuacji nie połączenia się z bazą należy wykonać następujące polecenie w MySQL-u :

SET PASSWORD for <użytkownik@localhost> = OLD\_PASSWORD('haslo'); I ponownie wybrać opcje "*Reverse Engineering*".

Sele	ect Database Connection							×
	Network Hosts	All Database Connectio	II Database Connections					
	🗁 All Connections	Connection	Туре	Host	Database	Description		_
	🖨 🗁 MySQL	DBDesigner2My	MySQL	Localhost	baza_DBDesign			
	🕀 🚮 Localhost							
	🖻 💑 Network Hosts							
	🕀 🥞 Localhost							
	🖻 🛅 SQLite							
	🖻 🛅 Oracle							
	🖻 🧰 MSSQL							
	🖻 🛅 ODBC	J						
								5
		Connection: DBD	esigner2M	ySQL			V Connect	J
	🗟 New Database Connection 🥒	Username: root		Passwo	ord: *******		🗙 Abort	

Rysunek 11. Okno wyboru połączenia.

Reverse Engineering	×
Database Connection	
DBDesigner2MySQL	5
Tables ♥ tfederacja ♥ tklub ♥ tkraj ♥ tzawodnik	Schema All Important When connecting to Oracle or MS SQL databases please make sure to have the appropriate rights to access the system tables. In doubt connect as DBA Users: SYSTEM (Oracle) or sa (MS SQL).
General Options Use MySQL specific functions 10 Tables placed in a Row Collapse Tables to save space on Canvas Import Schema as Table Prefixes Put Default Values in Quotes	Use Datatype substitution MySQL Standard int=INTEGER dec=DECIMAL
<ul> <li>Build Relations</li> <li>Build Relations based on Primary Keys</li> <li>Based on Tablenames and ID-Fieldnames</li> <li>Use Native Selects to retrive Relations</li> </ul>	Create Standard Inserts from table data
Connected to Database root@baza_D8Designer4	鞭 Execute 🛛 💥 Close

Rysunek 12. Okno Reverese Engineering.

Aby uzyskać spodziewany efekt należy nacisnąć przycisk "*Execute*", czego skutkiem będzie pojawienie się modelu naszej bazy danych.

