

2015



# WTYCZKA FARA-TCM

Dane techniczne dla twórców  
zewnętrznych aplikacji do obsługi map  
cmentarza

Dokumentacja techniczna dostępu do podstawowych danych cmentarnych w programie FARA.

wersja 1.0

aktualizacja: 15 września 2015



## Spis treści

Zasada.....	2
Włączanie i wyłączanie wtyczki w programie FARA.....	2
Błędy podczas synchronizacji danych między bazą główną a pomocniczą.....	2
Ponowne przeładowanie bazy pomocniczej.....	2
Wymagania.....	2
Lokalizacja.....	2
Format.....	3
Struktura.....	3
Specyfikacja.....	7

## Zasada

Program FARA nie udostępnia zawartości bazy danych innym aplikacjom – dane te są zaszyfrowane i możliwe do odczytu tylko przez program FARA. Z tego względu utworzona została wtyczka o nazwie FARA-TCM, której zasada polega na prowadzeniu przez program FARA równolegle drugiej, niezaszyfrowanej bazy danych zawierającej:

- ✓ lokalizację grobów,
- ✓ datę rozpoczęcia dzierżawy,
- ✓ okres dzierżawy (w latach),
- ✓ Informacje o osobach pochowanych obejmujące:
  - imię,
  - nazwisko,
  - datę urodzenia,
  - datę śmierci,
- ✓ lokalizacje i nazwy plików z załączonymi do księgi grobów zdjęciami

Baza ta jest aktualizowana przez program FARA synchronicznie w trakcie pracy programu – każda zmiana danych w głównej bazie programu powoduje natychmiastową aktualizację nieszyfrowanej bazy zewnętrznej.

## Włączanie i wyłączanie wtyczki w programie FARA

W programie FARA należy przejść do części **CMENTARZ** i kliknąć w ikonę **Komunikator dla zewnętrznych programów**. W wyświetlonym oknie należy zaznaczyć lub odznaczyć znacznik wtyczki i kliknąć w ikonę **Zastosuj**.

## Błędy podczas synchronizacji danych między bazą główną a pomocniczą

W przypadku wystąpienia błędu podczas synchronizacji (błąd odczytu dysku lub inna możliwość) na głównym oknie programu, w prawym panelu wyświetla się duże, żółte pole o postaci notatki z informacją o wystąpieniu błędu. Umieszczony tam przycisk „**napraw teraz**” pozwala na przeładowanie całej bazy pomocniczej i wypełnienie jej od nowa.

## Ponowne przeładowanie bazy pomocniczej

Jeśli z jakichś względów wystąpi potrzeba przeładowania całej bazy pomocniczej należy w programie FARA uruchomić funkcję **Komunikator dla zewnętrznych programów** (por. w rozdziale „Włączanie i wyłączanie wtyczki w programie FARA”) i wyłączyć wtyczkę. Spowoduje to usunięcie pomocniczej bazy danych. Następnie należy ponownie włączyć wtyczkę – spowoduje to ponowne wygenerowanie i wypełnienie danymi plików bazy pomocniczej.

## Wymagania

Program FARA w wersji 5.0.59 lub wyższej (pełny numer wersji wyświetlany jest w dolnym pasku głównego okna programu)

## Lokalizacja

W wersji 1-stanowiskowej programu FARA:

`%folder%\dla_innych\cmentarz`

gdzie **%folder%** oznacza folder, w którym zainstalowany jest program. Przykład: jeśli program zainstalowany jest w folderze **c:\fara** to baza danych do odczytu przez zewnętrzną aplikację umieszczona jest w folderze **c:\fara\dla\_innych\cmentarz**

W wersji sieciowej programu FARA:

**%serwer%\dla\_innych\cmentarz**

gdzie **%serwer%** oznacza folder, w którym znajduje się serwer programu i bazy danych. Przykład: jeśli baza danych programu zainstalowana jest w folderze **z:\fara\_serwer** to baza danych do odczytu przez zewnętrzną aplikację umieszczona jest w folderze **z:\fara\_serwer\dla\_innych\cmentarz**

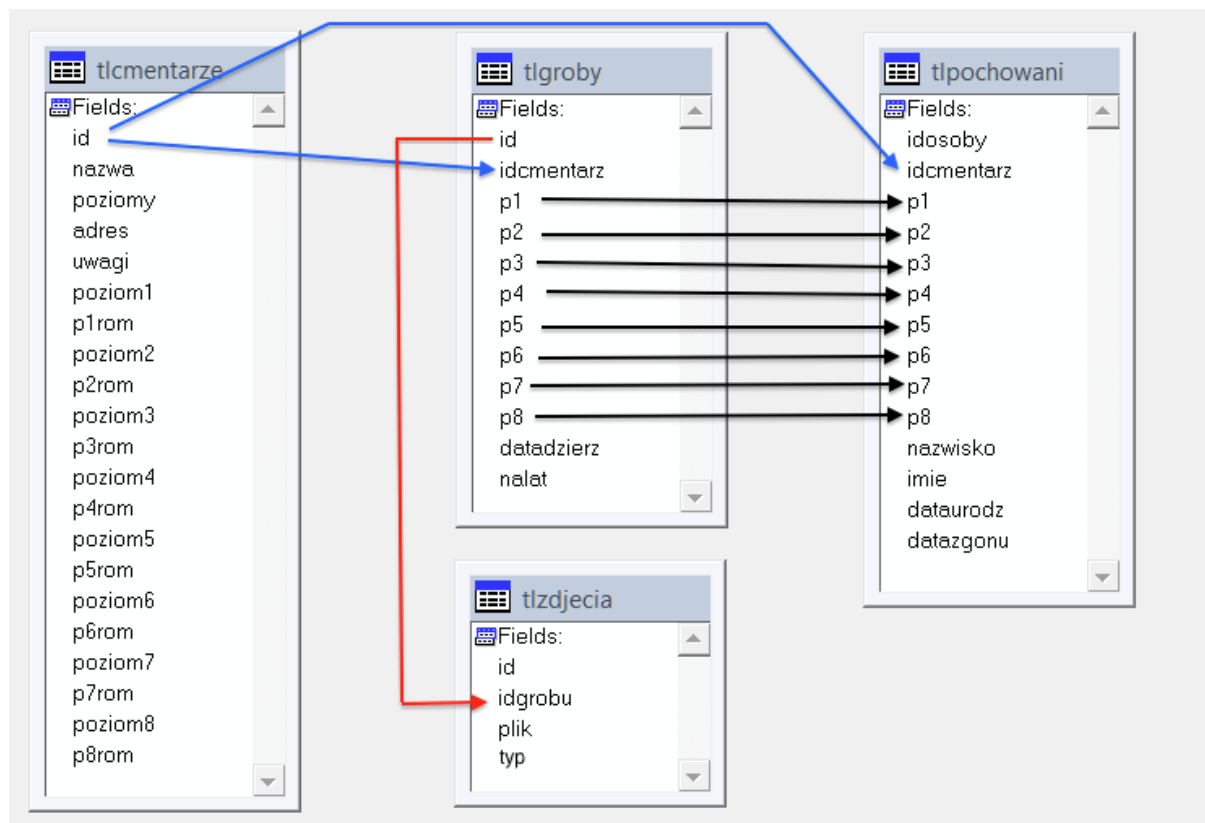
## Format

Komunikator pozwala na utrzymywanie bazy w dwóch, alternatywnych formatach:

- **SQLite:**
  - plik tlCmentarz.db z 4 indeksowanymi tablicami
  - kodowanie UTF-8
- **DBF:**
  - 4 nieindeksowane pliki DBF
  - Kodowanie: Windows 1250

## Struktura

Schemat bazy zamieszczono na rysunku 1.



Rysunek 1 - schemat nieszyfrowanej bazy danych z podstawowymi danymi z ksiąg cmentarnych

Tablica tlCmentarze:

Kolumna	Uwagi
id	Unikalny identyfikator cmentarza w programie FARA
nazwa	Nazwa cmentarza
adres	Adres cmentarza
uwagi	
poziomy	Liczba poziomów w strukturze cmentarza, np. jeśli cmentarz podzielony jest na sektory, te na rzędy, a te na groby to liczba poziomów wynosi <b>3</b> . Maksymalna liczba poziomów w programie FARA wynosi 8 – ale zazwyczaj stosowanych jest 2, 3 lub 4.
poziom1	Nazwa pierwszego poziomu w strukturze cmentarza – dla powyższego przykładu ma ona wartość „ <b>sektor</b> ”
p1rom	Wartość logiczna TRUE jeśli w dane na tym poziomie należy interpretować, jako liczby rzymskie (patrz: przykładowa tabela poniżej)
poziom2	Nazwa drugiego poziomu w strukturze cmentarza – dla powyższego przykładu ma ona wartość „ <b>rząd</b> ”
p2rom	Wartość logiczna TRUE jeśli w dane na tym poziomie należy interpretować, jako liczby rzymskie (patrz: przykładowa tabela poniżej)
.....	
poziom8	
p8rom	

Przykład rekordu tablicy opisującej cmentarz o 3 poziomej strukturze z nazewnictwem sektor-rząd-grób, gdzie rzędy są numerowane liczbami rzymskimi:

Kolumna	Wartość
id	<b>1</b>
nazwa	<b>Cmentarz parafialny</b>
adres	<b>ul. Klonowa 10, Przykładowice</b>
uwagi	
poziomy	<b>3</b>
poziom1	<b>Sektor</b>
p1rom	<b>FALSE</b>
poziom2	<b>rząd</b>
p2rom	<b>TRUE</b>
poziom3	<b>Grób</b>
P4rom	<b>FALSE</b>
poziom4	
p4rom	
poziom5	
p5rom	
poziom6	
p6rom	
poziom7	
p7rom	

poziom8	
p8rom	

#### Tablica tlGroby:

Kolumna	Uwagi
id	Unikalny identyfikator grobu w programie FARA. Identyfikator jest unikalny w całej bazie, nie tylko w ramach jednego cmentarza. Jeśli w bazie danych są 2 cmentarze to identyfikatory grobów w dalszym ciągu będą unikalne w skali całego zasobu danych.
idCmentarz	Identyfikator cmentarza → tlCmentarze.id
dataDzierz	Data rozpoczęcia dzierżawy w postaci tekstowej w formacie DD/MM/RRRR, gdzie dowolny znak może być zastąpiony znakiem '-' (minus), np. „--/--/2009”
naLat	Na ile lat jest dzierżawa
P1	Wartość pierwszego poziomu numeracji
P2	Wartość drugiego poziomu numeracji
.....	
P8	Wartość ósmego poziomu numeracji

Przykład rekordu tablicy opisującej grób A-VII-4a dzierżawiony od 03/05/2007 roku na 20 lat:

Kolumna	Wartość
id	<b>178</b>
idCmentarz	<b>1</b>
dataDzierz	<b>03/05/2007</b>
naLat	<b>20</b>
P1	<b>A</b>
P2	<b>VII</b>
P3	<b>4a</b>
P4	
P5	
P6	
P7	
P8	

#### Tablica tlPochowani:

Kolumna	Uwagi
idOsoby	Unikalny identyfikator osoby w programie FARA
idCmentarz	Identyfikator cmentarza → tlCmentarze.id
imie	Imię
nazwisko	Nazwisko
dataUrodz	Data urodzenia w postaci tekstowej w formacie DD/MM/RRRR, gdzie dowolny znak może być zastąpiony znakiem '-' (minus), np. „--/--/1942”
dataZgonu	Data śmierci w postaci tekstowej w formacie DD/MM/RRRR, gdzie dowolny znak może być zastąpiony znakiem '-' (minus), np. „--/--/1942”

P1	Wartość pierwszego poziomu numeracji
P2	Wartość drugiego poziomu numeracji
.....	
P8	Wartość ósmego poziomu numeracji

Przykład rekordu tablicy opisującej osoby pochowaną w grobie A-VII-4a:

Kolumna	Wartość
idOsoby	2630
idCmentarz	1
imie	Jan
nazwisko	Kowalski
dataUrodz	--/--/1941
dataZgonu	12/05/2007
P1	A
P2	VII
P3	4a
P4	
P5	
P6	
P7	
P8	

W przypadku większej liczby osób pochowanych w jednym grobie każda osoba opisana jest w osobnym rekordzie.

Tablica tlPochowani nie zawiera technicznego identyfikatora grobu. Relacja między tablicami tlGroby a tlPochowni opiera się na zgodności lokalizacji, czyli pól P1, P2 do P8.

Tablica tlZdjecia:

Kolumna	Uwagi
Id	Unikalny identyfikator zdjęcia w programie FARA
idGrobu	Identyfikator cmentarza → tlGroby.id
typ	Typ nazwy pliku w kolumnie <b>plik</b> : <b>0</b> – kolumna zawiera nazwę pliku na dysku komputera <b>1</b> – kolumna zawiera adres URL pliku
plik	Dla <b>typ=0</b> : Pełna nazwa pliku ze zdjęciem. Nazwa pliku może (nie musi) zawierać w sobie zmienną „%server%”, np.: „%server%\foto-groby\dcim003.jpg”. Zmienna %server% oznacza, iż w wersji sieciowej programu jest to folder serwera, zaś w wersji 1-stanowiskowej folder, w którym zainstalowany jest program. Dla <b>typ=1</b> : Pełny adres URL pliku

## Specyfikacja

Specyfikacja tablic podana została na przykładzie bazy SQLite. Nazwy i typy kolumn w tabelach DBF są identyczne.

```
CREATE TABLE tlcmentarze (  
    id          INT          NOT NULL,  
    nazwa      TEXT( 254 )  NOT NULL,  
    poziomy    INT          NOT NULL,  
    adres      TEXT( 254 )  NOT NULL,  
    uwagi     VARCHAR      NOT NULL,  
    poziom1   TEXT( 30 )   NOT NULL,  
    p1rom     BOOLEAN      NOT NULL,  
    poziom2   TEXT( 30 )   NOT NULL,  
    p2rom     BOOLEAN      NOT NULL,  
    poziom3   TEXT( 30 )   NOT NULL,  
    p3rom     BOOLEAN      NOT NULL,  
    poziom4   TEXT( 30 )   NOT NULL,  
    p4rom     BOOLEAN      NOT NULL,  
    poziom5   TEXT( 30 )   NOT NULL,  
    p5rom     BOOLEAN      NOT NULL,  
    poziom6   TEXT( 30 )   NOT NULL,  
    p6rom     BOOLEAN      NOT NULL,  
    poziom7   TEXT( 30 )   NOT NULL,  
    p7rom     BOOLEAN      NOT NULL,  
    poziom8   TEXT( 30 )   NOT NULL,  
    p8rom     BOOLEAN      NOT NULL  
);  
  
CREATE INDEX idx_tlcmentarze_id ON tlcmentarze (id);
```

---

```
CREATE TABLE tlGroby (  
    id          INTEGER     NOT NULL,  
    idCmentarz INT          NOT NULL,  
    P1         TEXT( 10 )   NOT NULL,  
    P2         TEXT( 10 )   NOT NULL,  
    P3         TEXT( 10 )   NOT NULL,  
    P4         TEXT( 10 )   NOT NULL,  
    P5         TEXT( 10 )   NOT NULL,  
    P6         TEXT( 10 )   NOT NULL,  
    P7         TEXT( 10 )   NOT NULL,  
    P8         TEXT( 10 )   NOT NULL,  
    dataDzierz TEXT( 10 )   NOT NULL,  
    naLat      INT          NOT NULL  
);  
  
CREATE INDEX idx_tlGroby_id ON tlGroby (id);  
CREATE INDEX idx_tlGroby_idCmentarz ON tlGroby (idCmentarz);  
CREATE INDEX idx_tlGroby_P1 ON tlGroby (P1);  
CREATE INDEX idx_tlGroby_P2 ON tlGroby (P2);  
CREATE INDEX idx_tlGroby_P3 ON tlGroby (P3);
```



```
CREATE INDEX idx_tlGroby_P4 ON tlGroby (P4);
```

---

```
CREATE TABLE tlPochowani (  
    idOsoby    INTEGER        NOT NULL,  
    idCmentarz INT           NOT NULL,  
    P1        TEXT( 10 )     NOT NULL,  
    P2        TEXT( 10 )     NOT NULL,  
    P3        TEXT( 10 )     NOT NULL,  
    P4        TEXT( 10 )     NOT NULL,  
    P5        TEXT( 10 )     NOT NULL,  
    P6        TEXT( 10 )     NOT NULL,  
    P7        TEXT( 10 )     NOT NULL,  
    P8        TEXT( 10 )     NOT NULL,  
    nazwisko  VARCHAR( 75 )  NOT NULL,  
    imie      VARCHAR( 75 )  NOT NULL,  
    dataUrodz TEXT( 10 )     NOT NULL,  
    dataZgonu TEXT( 10 )     NOT NULL  
);
```

```
CREATE INDEX idx_tlPochowani_idOsoby ON tlPochowani (idOsoby);  
CREATE INDEX idx_tlPochowani_idCmentarz ON tlPochowani (idCmentarz);  
CREATE INDEX idx_tlPochowani_P1 ON tlPochowani (P1);  
CREATE INDEX idx_tlPochowani_P2 ON tlPochowani (P2);  
CREATE INDEX idx_tlPochowani_P3 ON tlPochowani (P3);  
CREATE INDEX idx_tlPochowani_P4 ON tlPochowani (P4);
```

---

```
CREATE TABLE tlZdjecia (  
    id        INTEGER NOT NULL,  
    idGrobu  INTEGER NOT NULL,  
    plik     VARCHAR NOT NULL,  
    typ      INTEGER NOT NULL  
);
```

```
CREATE INDEX idx_tlZdjecia_id ON tlZdjecia (id);  
CREATE INDEX idx_tlZdjecia_idGrobu ON tlZdjecia (idGrobu);
```