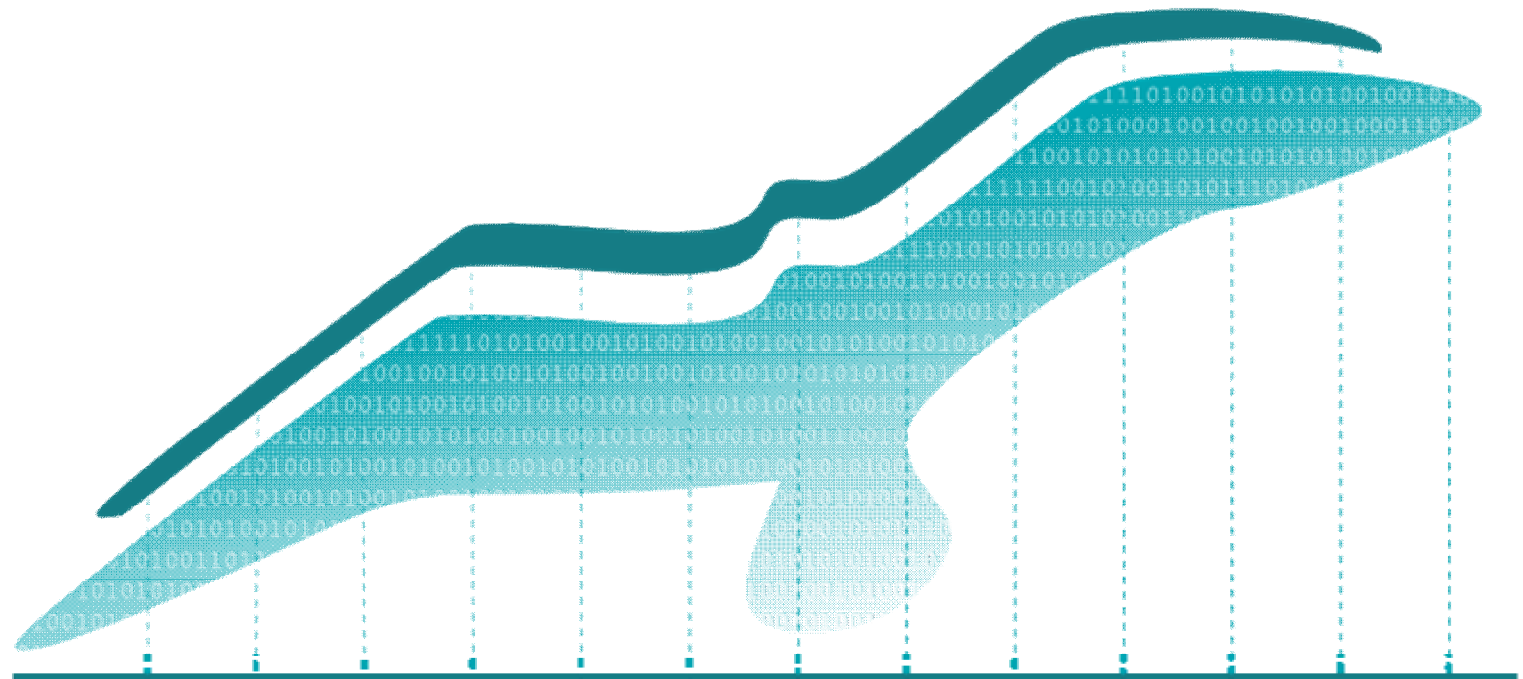


Higiena danych adresowych

Struktura bazy danych adresowych





Struktura bazy danych adresowych

Dokument opisuje strukturę zwracanej przez DataWise wynikowej tabeli z adresami po procesie czyszczenia adresów.

Struktura ta może też być także wykorzystana przy projektowaniu własnej bazy danych adresowych.

Tabela danych wynikowych

Standardowo zwracamy dane w jednym z niżej określonych formatów (format uzgadniamy ze zleceniodawcą czyszczenia adresów):

- zrzut tabeli do postaci poleceń SQL kompatybilnych z serwerem MySQL. Plik zawiera polecenia SQL INSERT INTO
- delimitowany plik tekstowy
- plik w formacie Microsoft Access

W przypadku pliku tekstowego dane są przygotowywane w jednej z trzech stron kodowych

- iso-8859-2 (latin2)
- windows-1250 (cp1250)
- utf-8

Niezależnie od formatu zwracanego pliku, jego zawartość odpowiada niżej podanej strukturze. Opisujemy strukturę dla formatu danych serwera MySQL.

Struktura tabeli wynikowej zawiera pola będące wynikiem procesu czyszczenia adresów oraz zawiera także pola wejściowe (inputowe), które przechowują wartości adresów do standaryzacji.



nazwa pola	format pola	zawartość pola	przykład	uwagi
dat_guid	int(8)	unikalny identyfikator rekordu w tabeli na serwerze DataWise	2	jeżeli dane inputowe zawierają unikalne ID rekordu w postaci liczby całkowitej, pole dat_guid jest wypełniane tym ID, w innych przypadkach DataWise nadaje wartość dat_guid automatycznie narastającą wartością liczbową
dat_input_fields	varchar(255)	pola z danymi wejściowymi adresu		
dat_input_zip_locality	varchar(255)	kod pocztowy i nazwa miejscowości w jednym polu	03-343 warszawa	
dat_input_street_all	varchar(255)	pozostała część adresu (prefix + nazwa + numer)	halera 2 lok 45 (parter)	
dat_locality_result	varchar(75)	nazwa miejscowości wg standardu DataWise	Warszawa	jeżeli nazwy nie udało się wystandaryzować, w tym polu podawana jest wartość inputowa (wejściowa)
dat_district_result	varchar(75)	nazwa dzielnicy lub części miejscowości	Praga-Północ	dla miast: Łódź, Kraków, Poznań, Warszawa, Wrocław wypełniana jest nazwa dzielnicy. W pozostałych przypadkach pojawia się nazwa tylko wtedy, gdy w inputie podana była nazwa części miejscowości
dat_locality_gus_symbol	char(7)	symbol nazwy miejscowości wg rejestru TERYT GUS	0918123	jeżeli nazwy nie udało się wystandaryzować, w tym polu podawana jest wartość 0000000
dat_district_gus_symbol	char(7)	symbol dzielnicy lub nazwy części miejscowości wg rejestru TERYT GUS	0919298	podawana jest tylko wtedy, gdy pole dat_district_result jest niepuste
dat_locality_result_in_dict	tinyint(1)	czy nazwa miejscowości jest odnajdywana w słowniku nazw miejscowości, pole przyjmuje wartości 0=nie, 1=tak	1	nazwa miejscowości może być odnajdywana w słowniku nazw miejscowości, ale może nie mieć przypisanego (pole dat_locality_gus_symbol). Dotyczy to sytuacji, kiedy poszczególne części adresu mają odnajdywane w słownikach wartości, ale adres nie jest logicznie spójny.
dat_post_result	varchar(75)	nazwa poczty wg DataWise		poczta zwracana jest tylko w sytuacji, kiedy nazwa



				poczty była podana w danych inputowych ORAZ nazwa poczty jest różna od nazwy miejscowości (pole dat_locality_result)
dat_locality_gus_symbol	char(7)	symbol nazwy miejscowości wg rejestru TERYT GUS		jeżeli pole dat_post_result jest niepuste, ale nie możnabyło nadać symbolu miejscowości dla poczty, pole przyjmuje wartość 0000000, jeżeli pole dat_post_result jest puste, to wtedy pole dat_locality_gus_symbol jest także puste
dat_locality_result_in_dict	tinyint(1)	czy nazwa poczty jest odnajdywana w słowniku nazw miejscowości, pole przyjmuje wartości 0=nie, 1=tak	0	patrz uwagi do pola dat_locality_result_in_dict
dat_street_prefix	varchar(5)	rodzaj typologiczny dla nazwy, przyjmuje wartości: ul., al., os., pl. oraz pusty string	UI	DataWise wyodrębnia tylko cztery podane rodzaje typologiczne nazw, pozostałe rodzaje typologiczne (np. most czy zaulek) wchodzi do nazwy ulicy
dat_street_result	varchar(75)	nazwa ulicy wg standardu DataWise	Hallera	
dat_street_building_number	varchar(50)	numer budynku	2	
dat_street_flat_number	varchar(50)	numer lokalu (mieszkania)	45	
dat_street_notes	varchar(75)	dodatkowe uwagi w adresie	PARTER	
dat_street_gus_name	varchar(75)	nazwa ulicy wg zapisu w rejestrze TERYT GUS	gen Józefa Hallera	
dat_street_gus_symbol	char(5)	symbol nazwy ulicy w rejestrze TERYT GUS	06484	
dat_geocode_mwg7	char(7)	symbol gminy wg rejestru TERYT GUS	1465011	pole przyjmuje wartość 0000000, jeżeli nie udało się nadać kodu gminy
dat_geocode_mwg7_district	char(7)	symbol dzielnicy wg rejestru TERYT GUS	1465088	pole wypełniane tylko dla miast : Łódź, Kraków, Poznań, Warszawa, Wrocław, w pozostałych przypadkach pusty string
dat_geocode_voivodship	varchar(75)	województwo	mazowieckie	pole przyjmuje wartość pustego stringu, jeżeli nie udało się nadać kodu gminy



dat_geocode_powiat	varchar(75)	powiat	Warszawa	pole przyjmuje wartość pustego stringu, jeżeli nie udało się nadać kodu gminy
dat_geocode_municipality	varchar(75)	gmina	Warszawa	pole przyjmuje wartość pustego stringu, jeżeli nie udało się nadać kodu gminy
dat_geocode_municipality_type	varchar(75)	typ gminy	miejska	pole przyjmuje wartość pustego stringu, jeżeli nie udało się nadać kodu gminy
dat_coord_x	float(8,6)	współrzędna X (długość geograficzna) adresu	20.954060	Pole przyjmuje wartość 0.000000 jeżeli nie było żądania geokodowania adresu lub jeżeli nie można było nadać współrzędnych adresu. Separatorem części dziesiętnej jest kropka.
dat_coord_y	float(8,6)	współrzędna Y (szerokość geograficzna) adresu	52.314880	Pole przyjmuje wartość 0.000000 jeżeli nie było żądania geokodowania adresu lub jeżeli nie można było nadać współrzędnych adresu. Separatorem części dziesiętnej jest kropka.
dat_coord_accuracy	smallint(4)	dokładność, z jaką nadano współrzędne XY	7	
dat_address_quality_score	smallint(4)	wskaźnik jakości adresu	9	wskaźnik jakości adresu przyjmuje wartości od 0 (najniższa jakość adresu) do 10 (najwyższa jakość adresu)
dat_hd_version	varchar(4)	numer wersji oprogramowania do czyszczenia adresów	22	

Uwagi:

1. pola typu varchar oraz char są polami tekstowymi. Liczba w nawiasie określa liczbę znaków, które pole może pomieścić
2. pola typu int, smallint, tinyint są polami liczbowymi, przeznaczonymi do przechowywania liczb całkowitych.
3. pola float są polami liczbowymi, przeznaczonymi do przechowywania liczb z częściami dziesiętnymi (po przecinku)