

# DOKUMENTACJA POWYKONAWCZA

## 8

### Struktura Bazy Danych

ZSSRWM      Zintegrowany System Sterowania Ruchem w Małopolsce

SCP      System czasu przejazdu

## SPIS TREŚCI

1. Wstęp .....	4
2. Schematy .....	6
3. Opis baz danych .....	19
4. Opis tabeli .....	19
4.1. DEVICE .....	19
4.2. LOCATION .....	20
4.3. ROUTER .....	20
4.4. METEO .....	20
4.5. METEOSENSOR .....	21
4.6. METEOSENSORDATA .....	21
4.7. RADAR .....	22
4.8. RADARLANE .....	22
4.9. RADARLANEDATA .....	23
4.10. LENGTHCLASSIFICATION .....	23
4.11. RADARLANEDATALENGTH .....	24
4.12. ANPR .....	24
4.13. ANPRLANE .....	25
4.14. PLATEDETECTION .....	25
4.15. ROUTE .....	26
4.16. SECTION .....	27
4.17. ANPRLANESECTION .....	28
4.18. SECTIONROUTE .....	28
4.19. ROUTEINTERVAL .....	28
4.20. SECTIONINTERVAL .....	29
4.21. ROUTEDATA .....	29
4.22. SECTIONDATA .....	30
4.23. MATCHING .....	30
4.24. SYSTEMUSER .....	31
4.25. SYSTEMUSERSTATUS .....	31
4.26. SYSTEMUSERACCESS .....	32
4.27. CAMERA .....	32

4.28.	IMAGELOG .....	33
4.29.	VMS .....	33
4.30.	VMSGGRAPHICFOLDER .....	33
4.31.	VMSGGRAPHIC.....	34
4.32.	VMSMACRO.....	34
4.33.	VMSMACROMTO.....	35
4.34.	MODULO .....	35
4.35.	MODULOFEATURE .....	35
4.36.	MODULEFEATUREUSERRIGHT .....	36
4.37.	VEHICLE .....	36
4.38.	TABLELE Z KODAMI IDENTYFIKUJĄCYMI POCHODZENIE POLSKICH TABLIC REJESTRACYJNYCH.....	37
4.38.1.	VOIVODESHIP .....	49
4.38.2.	POWIAT .....	49
4.38.3.	CITY.....	49
4.38.4.	DISTRICT .....	50
4.38.5.	SPECIALPOLISHPLATE .....	50
4.39.	WHITELISTVEHICLE .....	51
4.40.	WHITELISTDETECTION .....	51
4.41.	BLACKLISTDETECTION.....	52
4.42.	ALARMSTATUS.....	53
4.43.	ALARMCONFIGURATION .....	53
4.44.	ALARMUSERREAD.....	54
4.45.	ALARMUSERWRITE .....	54
4.46.	ALARMUSERSMS .....	54
4.47.	ALARMUSERMAIL .....	54
4.48.	SEQUENCES .....	54
4.49.	CAMERASEQUENCES .....	55
4.50.	REPORTFILTERS.....	55
4.51.	ANPRLANEREPORTFILTERS .....	56
4.52.	RADARLANEREPORTFILTERS.....	56
4.53.	VOIVODESHIPSREPORTFILTERS .....	56
4.54.	POWIATSREPORTFILTERS .....	57
4.55.	CITIESREPORTFILTERS.....	57

4.56.	DISTRICTSREPORTFILTERS .....	57
4.57.	SECTIONREPORTFILTERS.....	57
4.58.	REPORT.....	58
4.59.	MACRO .....	58
4.60.	CONTACT .....	59
4.61.	VEHICLECHRONOS .....	59
4.62.	SCHEDULE.....	60
4.63.	CALENDAR .....	60
4.64.	CALENDARDATA .....	60
4.65.	ANPRCALENDARVEHICLEUSER.....	61
4.66.	CONTACTANPRCALENDARVEHICLEUSER .....	61
4.67.	EXTERNALLAYERS .....	62
4.68.	EVENT .....	62
5.	SQL .....	63

## 1. Wstęp

Systemem zarządzania bazą danych wykorzystywany do przechowywania danych systemu SCP jest PostgreSQL w wersji 9.4 lub wyższej, jeżeli taka jest dostępna w momencie implementacji systemu. PostgreSQL to system zarządzania obiektowo- relacyjnymi bazami danych, udostępniany na zasadach licencji BSD z wolno dostępnym kodem źródłowym. Jest to najbardziej wydajny na rynku otwartoźródłowy system zarządzania bazami danych, a jego ostatnie wersje niczym nie ustępują systemom komercyjnym.

PostgreSQL używa modelu klient/ serwer oraz, aby zagwarantować stabilność systemu zamiast z *wielowątkowości* korzysta z *wieloprocusowości*. Błąd któregośkolwiek z procesów nie wpływa na pozostałe procesy oraz na działanie systemu.

### Charakterystyka

Dzięki swoim parametrom PostgreSQL jest jedną z najbardziej wydajnych i solidnych baz danych na rynku. Jej rozwój rozpoczął się ponad 16 lat temu i przez ten czas stabilność, wydajność, solidność oraz łatwość administrowania oraz implementacji standardów stały się cechami, na które stawiano największy nacisk. PostgreSQL świetnie radzi sobie z dużą ilością danych oraz wieloma użytkownikami jednocześnie korzystającymi z systemu.

### Ogólne

- Baza danych w 100% ACID (Atomicity, Consistency, Isolation and Durability)
- Integralność referencyjna
- Przestrzenie tabel (Tablespaces)
- Transakcje zagnieżdżone (punkty zapisu)
- Replikacja asynchroniczna/ synchroniczna/ Streaming replication- Hot Standby
- Dwufazowe potwierdzenie
- Odtworzenie bazy danych do zadanego punktu w czasie (PITR)
- Kopie zapasowe na gorąco (Online/ Hot backups)
- Unicode
- Międzynarodowe zestawy znaków
- Sortowanie po kolumnie
- Mechanizm MVCC (Multi-Version Concurrency Control)
- Wiele metod uwierzytelniania
- Dostęp szyfrowany poprzez SSL
- Zintegrowana aktualizacja in- situ (pg\_upgrade)
- Podwyższony poziom zabezpieczeń (SE-postgres)
- Kompletna dokumentacja
- Licencja BSD
- Dostępne dla Linux, UNIX we wszystkich odmianach (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) oraz Windows 32/64bit.

## Programowanie/ Rozwój

- Funkcje/ procedury składowane (stored procedures) w wielu językach programowania, między innymi PL/pgSQL (podobny do PL/SQL w bazie Oracle), PL/Perl, PL/Python y PL/Tcl
- Anonimowe bloki kodu procedur (polecenie DO)
- Wiele typów danych oraz możliwość definiowania nowych typów. Poza typami standardowymi dla wszystkich baz danych są dostępne również między innymi: typy geometryczne, typy adresów sieciowych, typy binarne, UUID, XML, macierze, etc.
- Przechowywanie dużych obiektów binarnych (grafika, video, pliki dźwiękowe,...)
- API do programowania w językach: C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt i wielu innych.

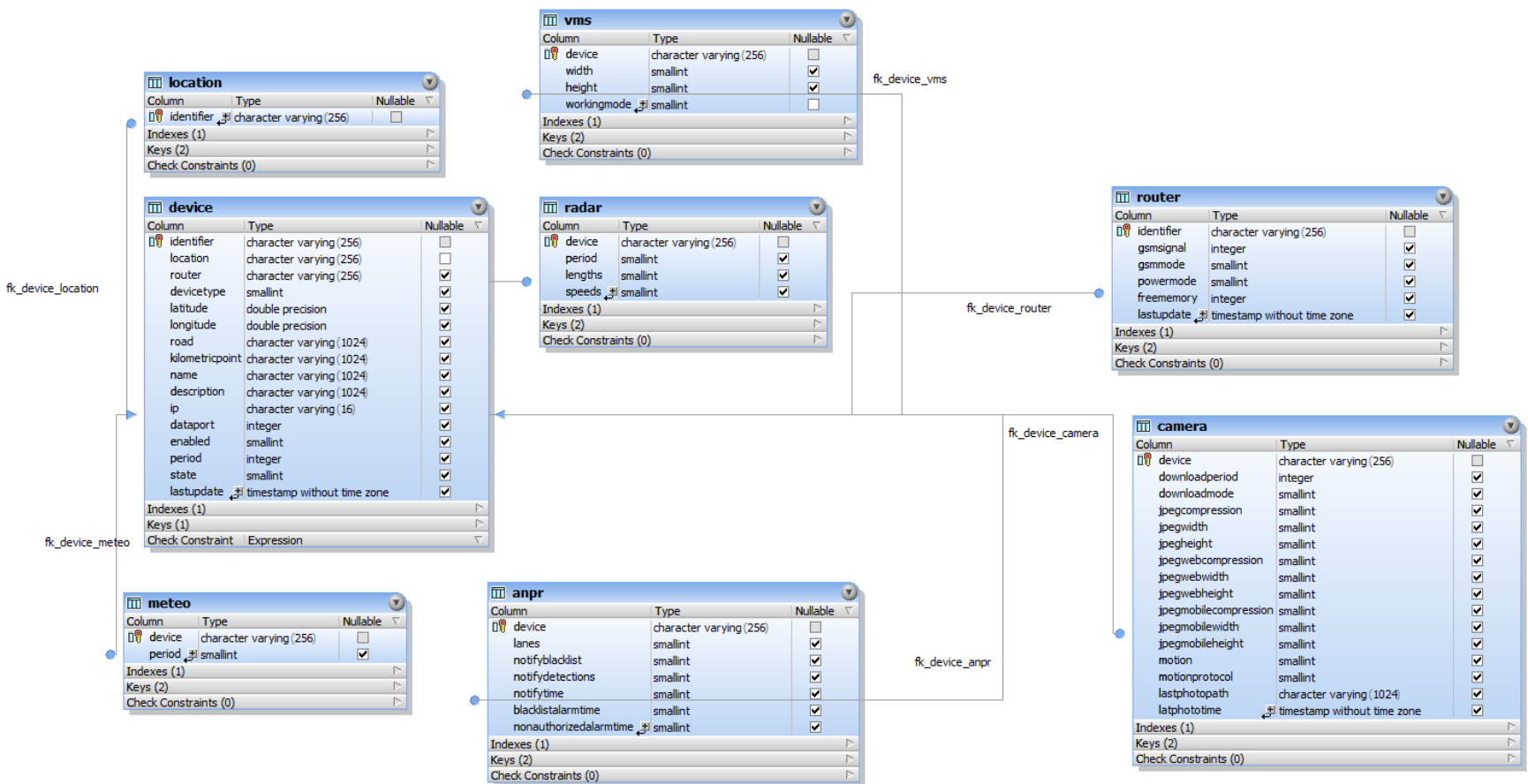
## SQL

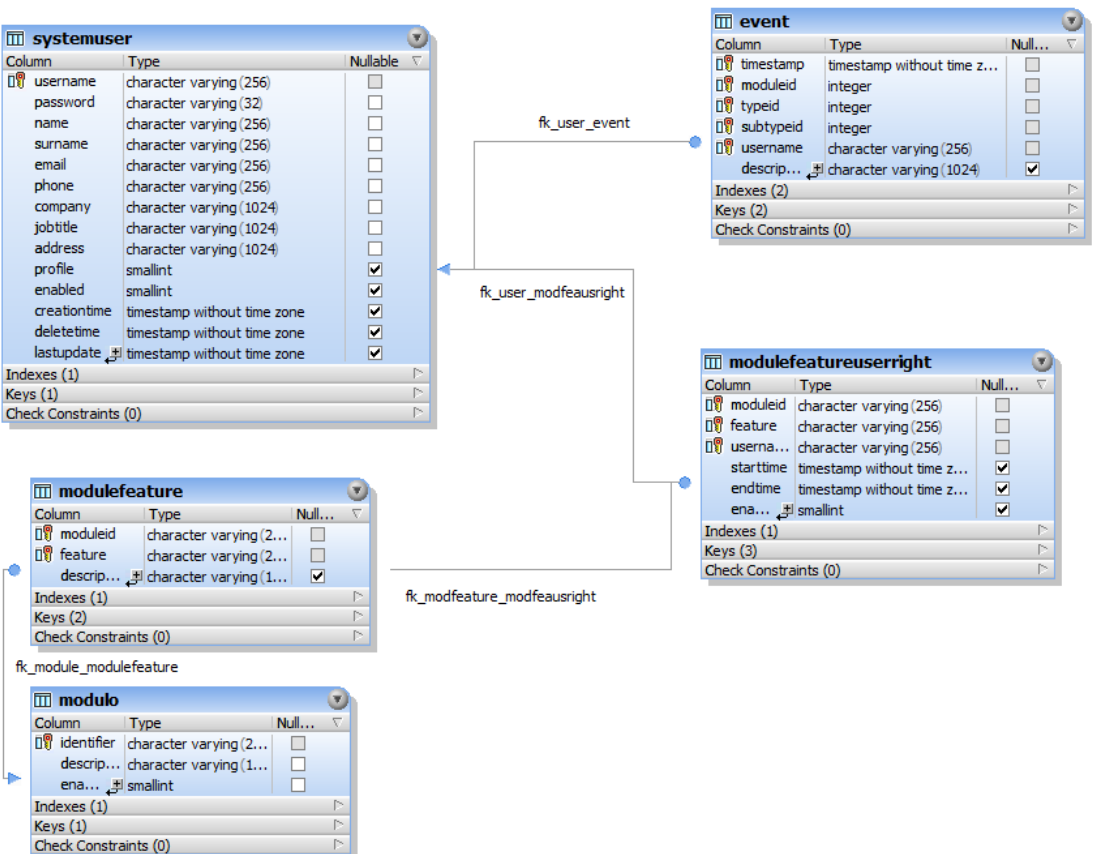
- SQL92,SQL99,SQL2003,SQL2008
- Klucze podstawowe (primary keys) i obce (foreign keys)
- Ograniczenia Check, Unique i Not null
- Ograniczenia możliwe do odroczenia (deferrable constraints)
- Kolumny automatycznego przyrostu
- Indeksy kompozytowe, unikatowe, częściowe i funkcjonalne we wszystkich dostępnych typach: B-tree, R-tree, Hash lub GiST
- Subselect
- Zapytania rekurencyjne
- Funkcje 'Windows'
- Join
- Widoki (views)
- Wyzwalacze (triggers) ogólne, warunkowe lub działające na poziomie kolumny
- Reguły (Rules)
- Dziedziczenie tabel (Inheritance)
- Powiadomienia o zajściu zdarzenia (LISTEN/NOTIFY).

## Ograniczenia

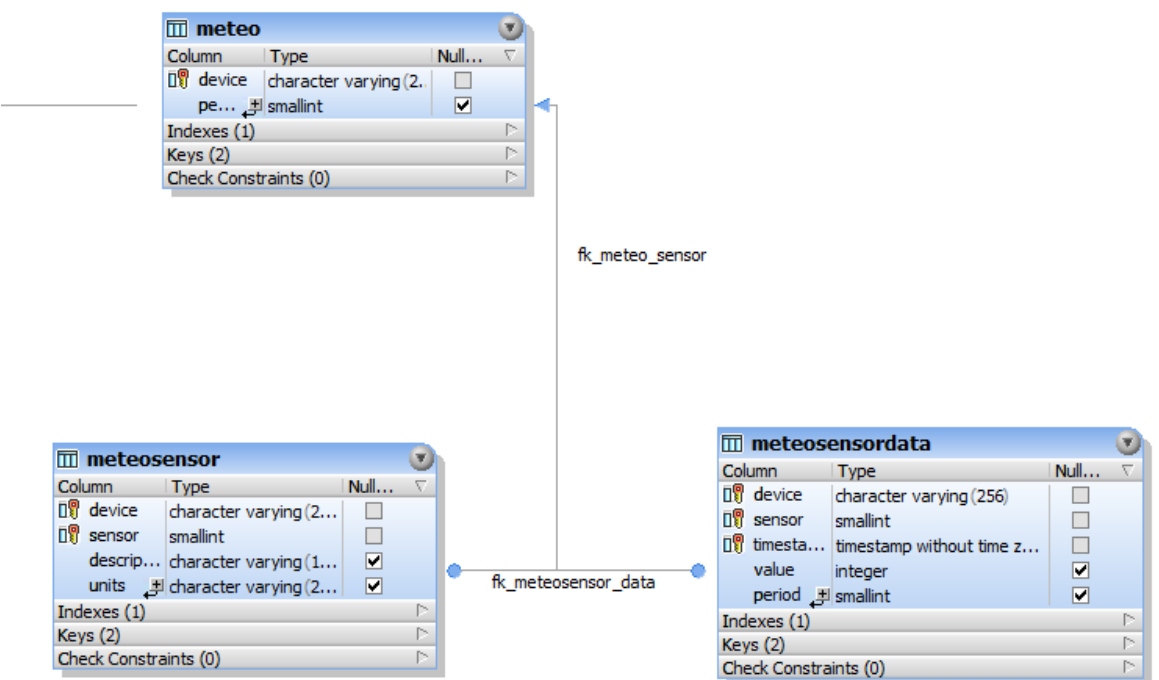
Nazwa ograniczenia	Wartość
Maksymalny rozmiar bazy	Bez ograniczeń
Maksymalny rozmiar tabeli	32 TB
Maksymalny rozmiar wiersza	1,6 TB
Maksymalny rozmiar pola w wierszu	1 GB
Maksymalny liczba wierszy w tabeli	Bez ograniczeń
Maksymalny liczba kolumn w tabeli	250 - 1600 (zależy od rodzaju)
Maksymalna liczba indeksów dla tabeli	Bez ograniczeń

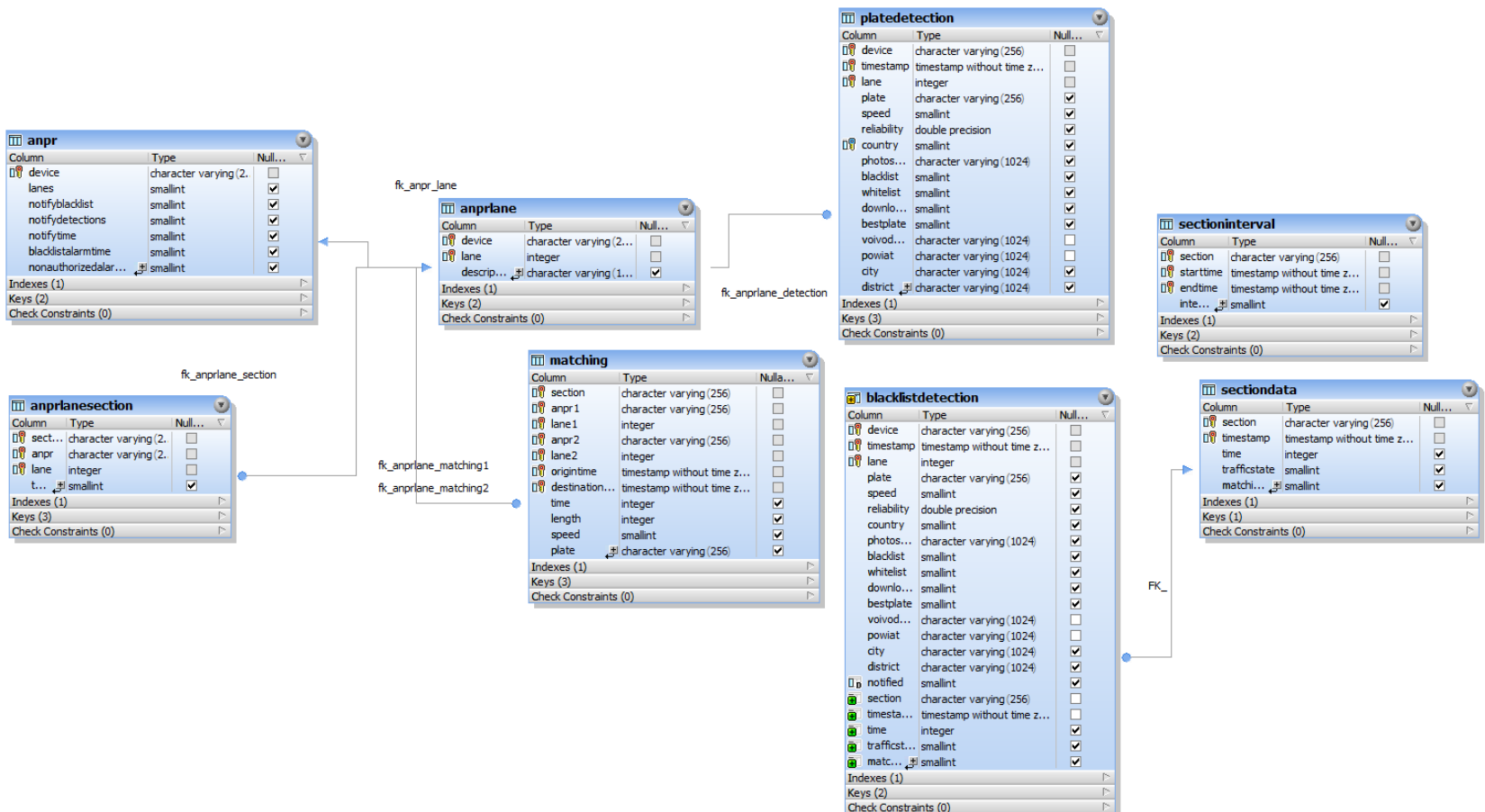
## 2. Schematy











Column	Type	Null...
plate	character varying (2...	<input type="checkbox"/>
organism	integer	<input checked="" type="checkbox"/>
mark	character varying (2...	<input checked="" type="checkbox"/>
model	character varying (2...	<input checked="" type="checkbox"/>
colour	character varying (2...	<input checked="" type="checkbox"/>
country	character varying (2...	<input checked="" type="checkbox"/>
owner	character varying (2...	<input checked="" type="checkbox"/>
blac...	character varying (1...	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
folder	character varying (1...	<input type="checkbox"/>
name	character varying (1...	<input type="checkbox"/>
objectdata	bytea	<input checked="" type="checkbox"/>
descrip...	character varying (1...	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (1)		
Check Constraints (0)		

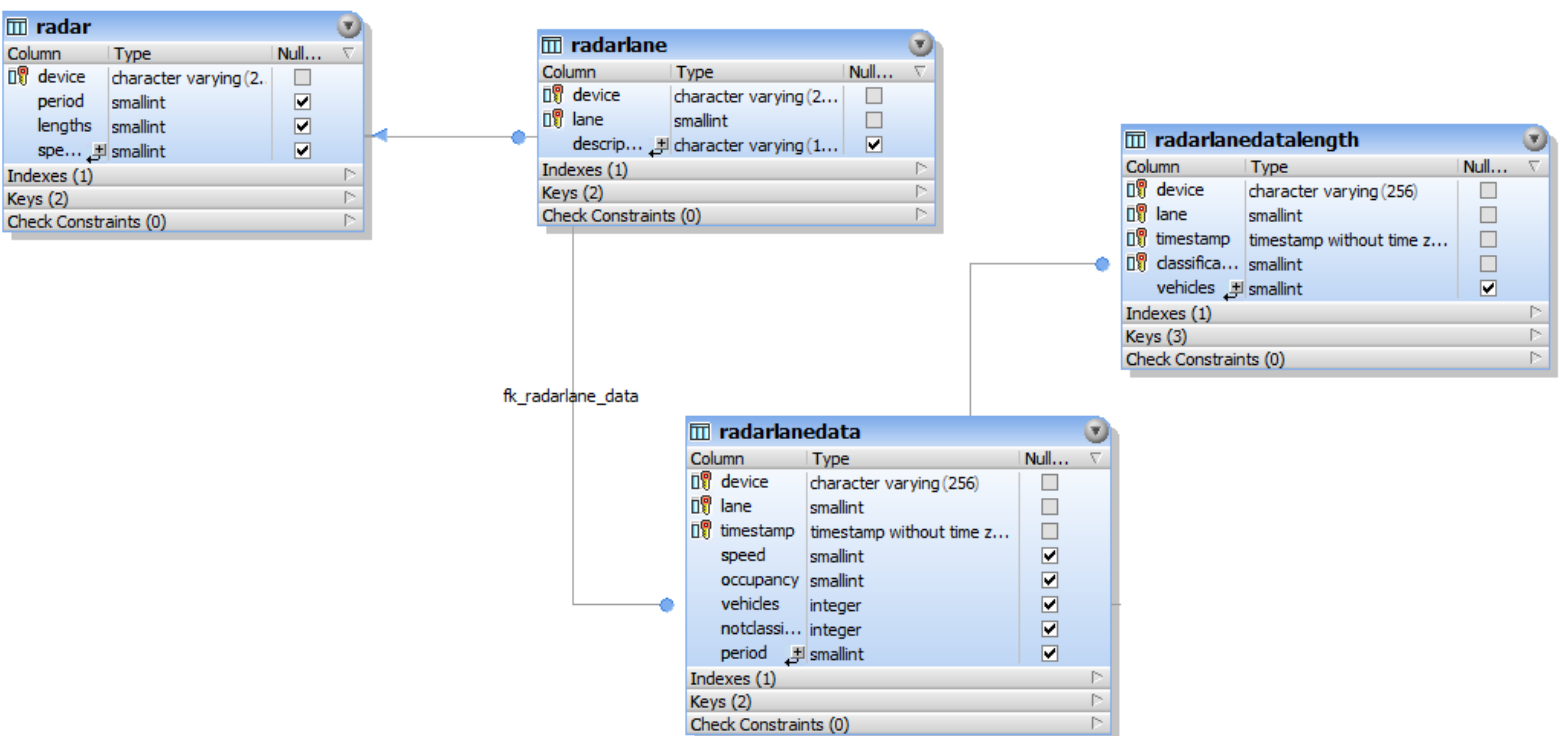
Column	Type	Null...
identifier	character varying (1...	<input type="checkbox"/>
device	character varying (2...	<input type="checkbox"/>
ordernu...	smallint	<input type="checkbox"/>
viewt...	smallint	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

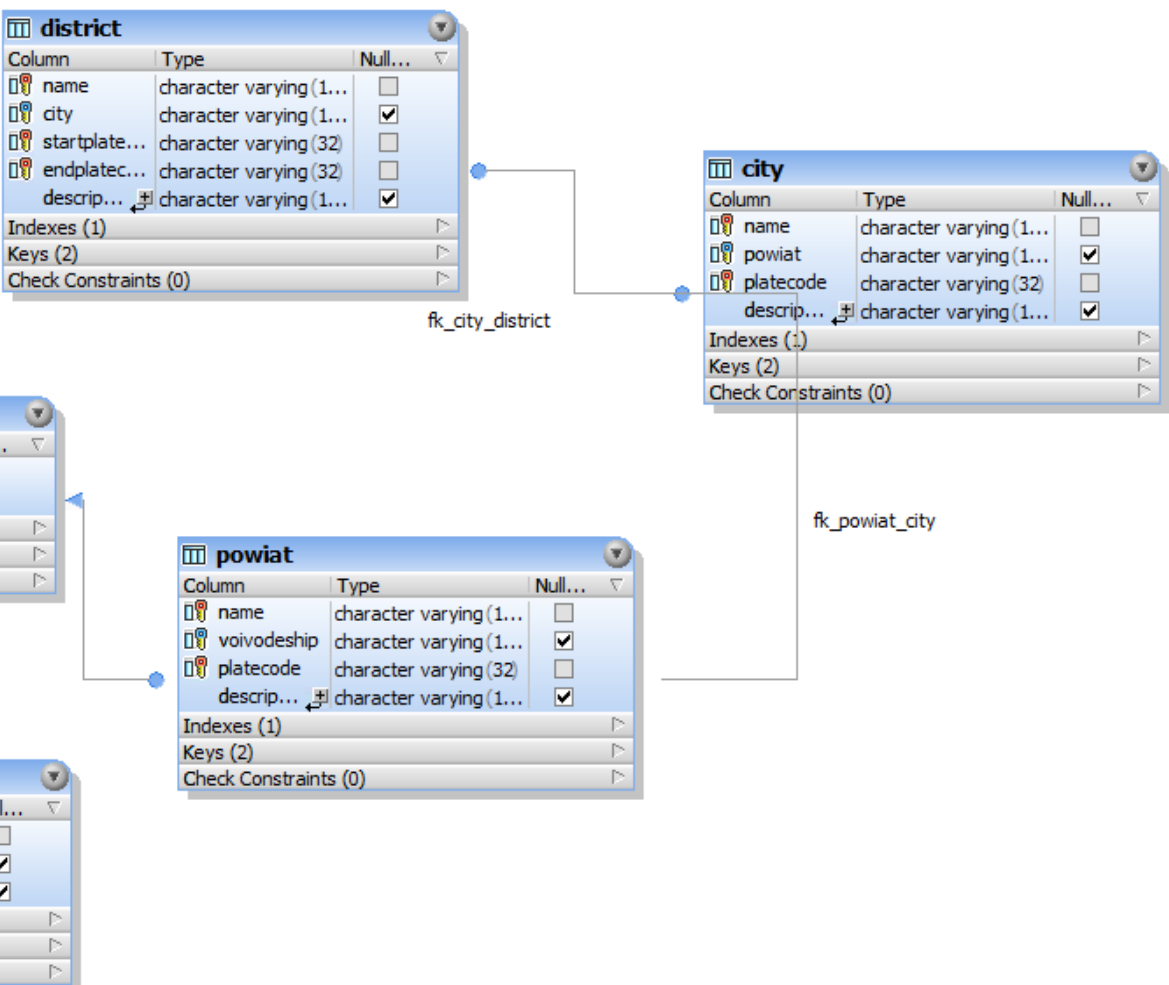
Column	Type	Null...
device	character varying (256)	<input type="checkbox"/>
timesta...	timestamp without time z...	<input type="checkbox"/>
imagep...	character varying (1024)	<input checked="" type="checkbox"/>
dele...	smallint	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (1)		
Check Constraints (0)		

Column	Type	Null...
name	character varying (2...	<input type="checkbox"/>
i...	bytea	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (1)		
Check Constraints (0)		

Column	Type	Null...
ident...	character varying (1...	<input type="checkbox"/>
n...	character varying (1...	<input type="checkbox"/>
Indexes (1)		
Keys (1)		
Check Constraints (0)		

fk\_sequences\_sequences



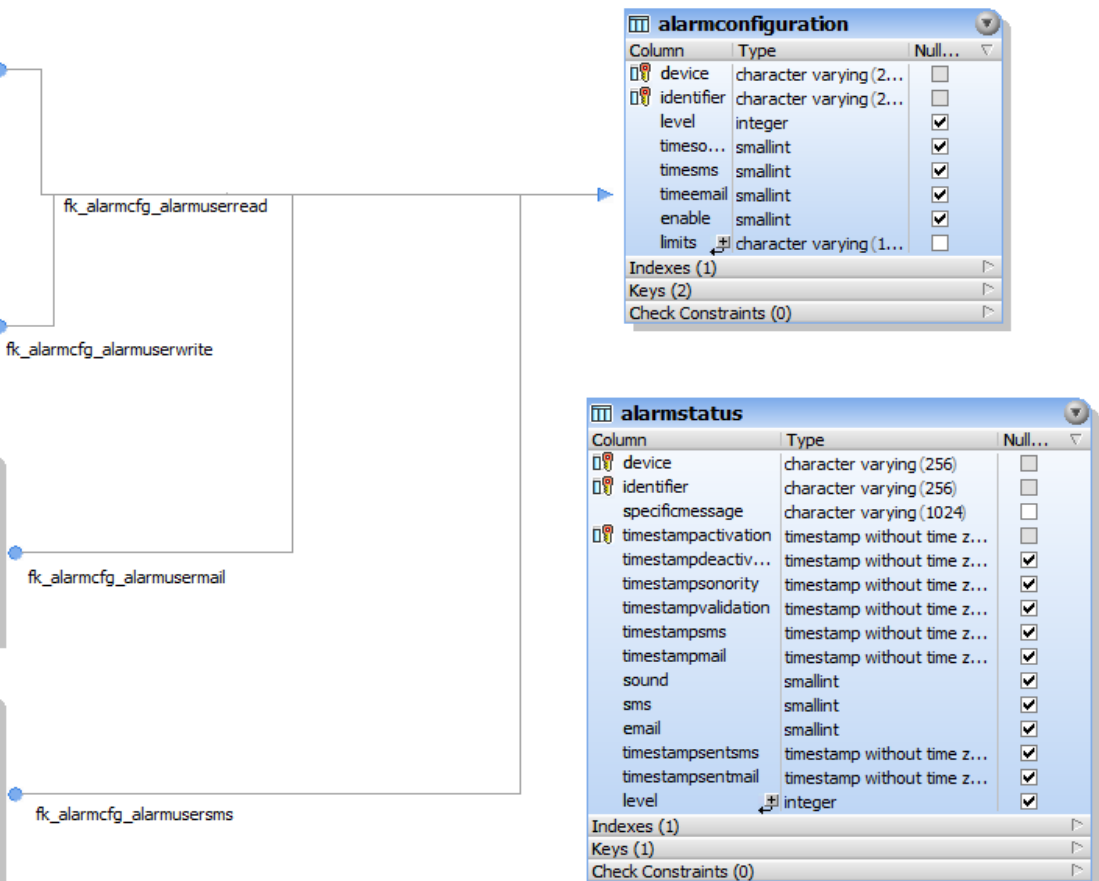


Column	Type	Null...
device	character varying (2..)	<input type="checkbox"/>
alarm	character varying (2..)	<input type="checkbox"/>
usern...	character varying (2..)	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
device	character varying (2..)	<input type="checkbox"/>
alarm	character varying (2..)	<input type="checkbox"/>
usern...	character varying (2..)	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
device	character varying (2..)	<input type="checkbox"/>
alarm	character varying (2..)	<input type="checkbox"/>
usern...	character varying (2..)	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
device	character varying (2..)	<input type="checkbox"/>
alarm	character varying (2..)	<input type="checkbox"/>
usern...	character varying (2..)	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		



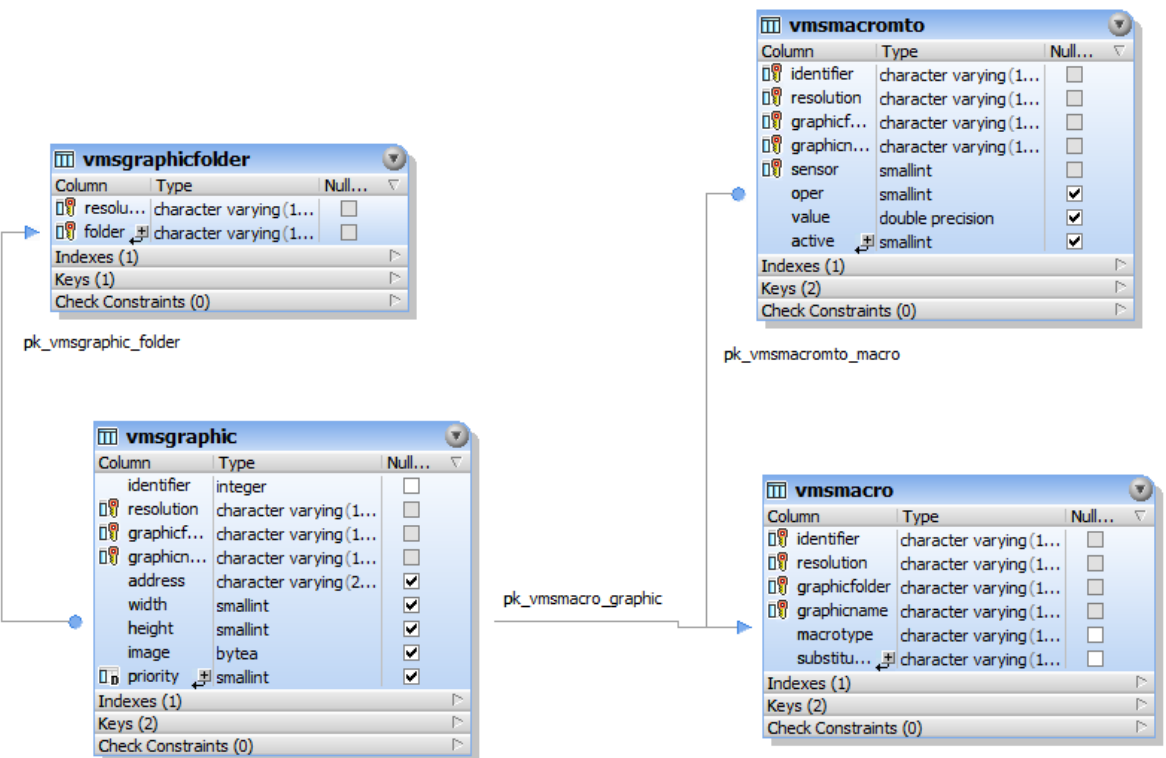
routedata		
Column	Type	Null...
route	character varying (256)	<input type="checkbox"/>
timestamp	timestamp without time z...	<input type="checkbox"/>
time	integer	<input checked="" type="checkbox"/>
traffict...	smallint	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

sectionroute		
Column	Type	Null...
route	character varying (2.	<input type="checkbox"/>
section	character varying (2.	<input type="checkbox"/>
ordernu...	smallint	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (3)		
Check Constraints (0)		

routeinterval		
Column	Type	Null...
route	character varying (256)	<input type="checkbox"/>
starttime	timestamp without time z...	<input type="checkbox"/>
endtime	timestamp without time z...	<input type="checkbox"/>
inte...	smallint	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

fk\_route\_data

route		
Column	Type	Null...
identifier	character varying (256)	<input type="checkbox"/>
creationlogin	character varying (256)	<input type="checkbox"/>
creationtime	timestamp without time z...	<input type="checkbox"/>
direction	character varying (256)	<input checked="" type="checkbox"/>
description	character varying (1024)	<input checked="" type="checkbox"/>
fixedinterval	smallint	<input checked="" type="checkbox"/>
length	integer	<input checked="" type="checkbox"/>
modeltime	integer	<input checked="" type="checkbox"/>
clientvisible	smallint	<input checked="" type="checkbox"/>
webvisible	smallint	<input checked="" type="checkbox"/>
mobilevisible	smallint	<input checked="" type="checkbox"/>
working...	smallint	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (1)		
Check Constraints (0)		





Column	Type	Null...
reportfilter	integer	<input type="checkbox"/>
voivodeship	character varying(2..)	<input type="checkbox"/>
ordernu...	smallint	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
reportfilter	integer	<input type="checkbox"/>
anpr	character varying(2..)	<input type="checkbox"/>
lane	integer	<input type="checkbox"/>
ordernu...	smallint	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
reportfilter	integer	<input type="checkbox"/>
powiat	character varying(2..)	<input type="checkbox"/>
ordernu...	smallint	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
reportfilter	integer	<input type="checkbox"/>
radar	character varying(2..)	<input type="checkbox"/>
lane	smallint	<input type="checkbox"/>
ordernu...	smallint	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

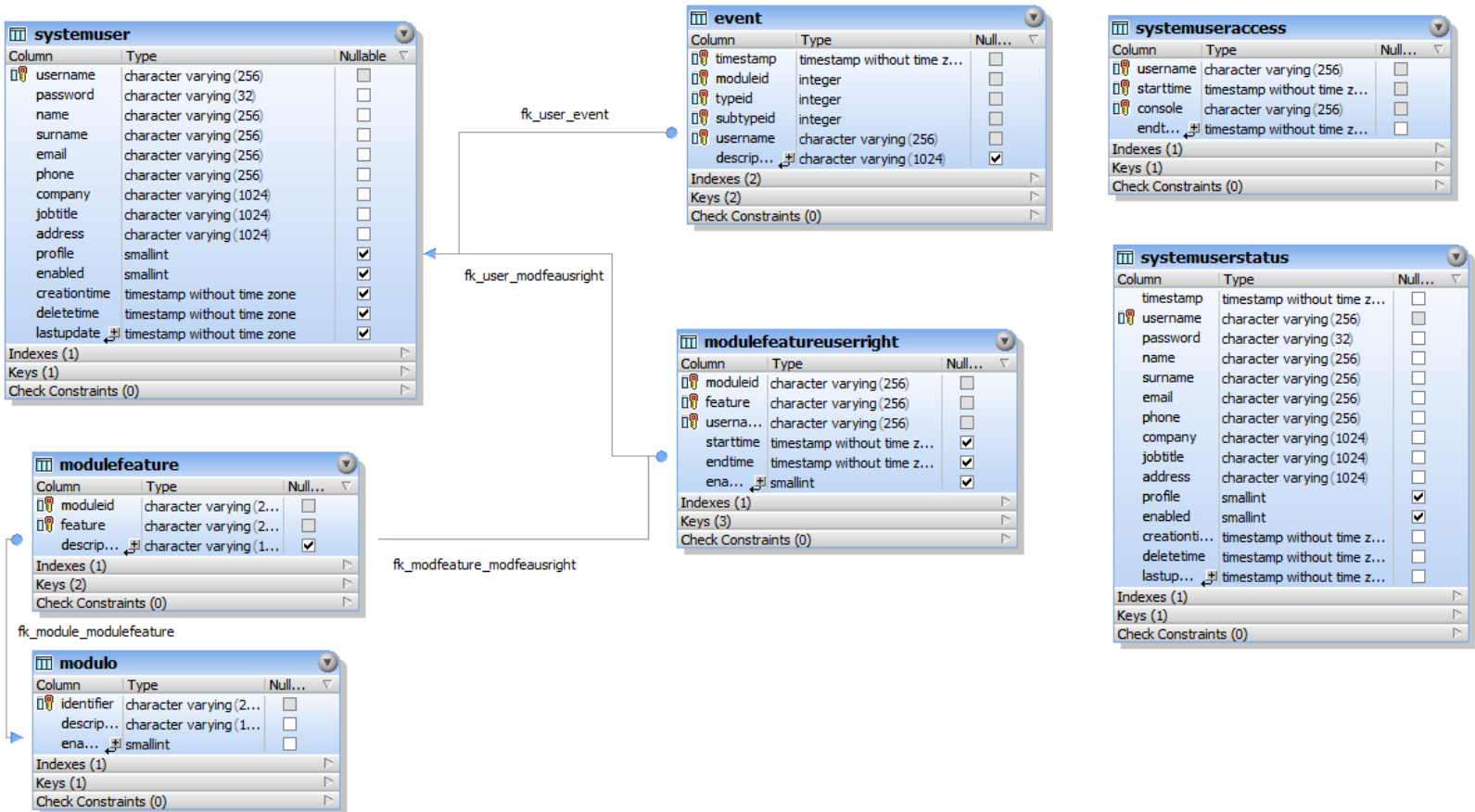
Column	Type	Null...
identifier	integer	<input type="checkbox"/>
reportn...	character varying(1024)	<input type="checkbox"/>
filtername	character varying(1024)	<input type="checkbox"/>
savepdf	smallint	<input checked="" type="checkbox"/>
openrep...	smallint	<input checked="" type="checkbox"/>
creation...	timestamp without time z...	<input type="checkbox"/>
createdby	character varying(256)	<input type="checkbox"/>
period	integer	<input checked="" type="checkbox"/>
units	integer	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (1)		
Check Constraints (0)		

Column	Type	Null...
identifier	integer	<input type="checkbox"/>
reportn...	character varying(1024)	<input type="checkbox"/>
filtername	character varying(1024)	<input type="checkbox"/>
creation...	timestamp without time z...	<input type="checkbox"/>
createdby	character varying(256)	<input type="checkbox"/>
link	character varying(1024)	<input checked="" type="checkbox"/>
path	character varying(1024)	<input checked="" type="checkbox"/>
period	integer	<input checked="" type="checkbox"/>
tsinit1	timestamp without time z...	<input checked="" type="checkbox"/>
tsend1	timestamp without time z...	<input checked="" type="checkbox"/>
tsinit2	timestamp without time z...	<input checked="" type="checkbox"/>
tsend2	timestamp without time z...	<input checked="" type="checkbox"/>
Indexes (1)		
Keys (1)		
Check Constraints (0)		

Column	Type	Null...
reportfilter	integer	<input type="checkbox"/>
city	character varying(2..)	<input type="checkbox"/>
ordernu...	smallint	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
reportfilter	integer	<input type="checkbox"/>
sectionid	character varying(2..)	<input type="checkbox"/>
ordernu...	smallint	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		

Column	Type	Null...
reportfilter	integer	<input type="checkbox"/>
district	character varying(2..)	<input type="checkbox"/>
ordernu...	smallint	<input type="checkbox"/>
Indexes (1)		
Keys (2)		
Check Constraints (0)		



### 3. Opis baz danych

W systemie znajdują się trzy bazy danych, które korzystają z relacyjnego systemu zarządzania bazami danych PostgreSQL w celu uzyskania maksymalnej dyspozycyjności, bezpieczeństwa i wydajności systemu.

Dwie pierwsze znajdują się na serwerach 11 i 12 i funkcjonują, jako główne bazy danych systemu jako: baza nadrzędna (Master) i podrzędna (Slave). Obie bazy danych są ze sobą synchronizowane, dzięki czemu zawsze zawierają te same informacje i zapewniona jest integralność danych.

Jeśli system wykryje awarię serwera Master, serwer Slave w sposób automatyczny rozpoczyna zarządzanie zadaniami serwera Master. Aplikacje klienckie w sposób automatyczny wykrywają awarię serwera nadrzędnego i łączą się automatycznie z serwerem podrzędnym.

Trzecia baza danych znajduje się na serwerze 13 i jest wykorzystywana wyłącznie do zarządzania i przechowywania informacji detekcji pojazdów znajdujących się na białej liście. W ten sposób zapewnione zostaje maksymalne bezpieczeństwo i niezależność.

### 4. Opis tabeli

Baza danych systemu SCP składa się z następujących tabeli:

#### 4.1. DEVICE

W tej tabeli przechowywane są informacje dotyczące różnych urządzeń systemu (kamera, ANPR, VMS, Radar, METEO).

Kolumna	Opis
identifier	Identyfikator urządzenia.
location	Identyfikator lokalizacji, do której należy urządzenie.
router	Identyfikator routera, którego używa urządzenie do komunikacji.
deviceType	Typ urządzenia { 0 : ANPR, 1 : VMS, 2 : Kamera, 3 : Radar, 4 : METEO, 5: Odcinek, 6: Trasa }
latitude	Pozycja GPS urządzenia- szerokość geograficzna.
longitude	Pozycja GPS urządzenia- długość geograficzna.
road	Droga, na której znajduje się urządzenie.
kilometricPoint	Kilometraż.
name	Nazwa urządzenia.
description	Opis urządzenia.
ip	Adres IP urządzenia.
dataport	Port danych urządzenia.
enabled	Stan urządzenia { 0 : Wyłączone, 1 : Włączone }.

<b>period</b>	Okres sprawdzania przez system alarmów i stanów pracy urządzenia (w sekundach).
<b>state</b>	Stan pracy urządzenia { 0 : Stan normalny, 1 : Stan anormalny, 2 : Awaria. 3 : Brak komunikacji, 4 : Nieznany }.
<b>lastUpdate</b>	Data ostatniej modyfikacji informacji dotyczącej urządzenia.

#### 4.2. LOCATION

W tej tabeli przechowywane są informacje dotyczące różnych lokalizacji urządzeń systemu. Lokalizacją nazywa się punkt, w którym znajdują zainstalowane są różne urządzenia (kamera, ANPR, VMS, Radar, METEO).

Kolumna	Opis
<b>identifier</b>	Identyfikator lokalizacji.

#### 4.3. ROUTER

W tej tabeli przechowywane są informacje dotyczące różnych routerów systemu.

Kolumna	Opis
<b>identifier</b>	Identyfikator routera.
<b>gsmSignal</b>	Poziom sygnału GSM (0 – 100%)
<b>gsmMode</b>	Tryb GSM { 0 : Nieznany, 1 : GPRS, 2 : EDGE, 3 : 3G- UMTS, 4 : HSPA, 5 : HSUPA, 6 : 4G LTE }.
<b>powerMode</b>	Tryb zasilania { 0 : Stabilne, 1 : Akumulatorowe, 2 : Inne }.
<b>freeMemory</b>	Procent wolnej pamięci { 0..100 % }.
<b>lastUpdate</b>	Data ostatniej modyfikacji informacji dotyczącej routera.

#### 4.4. METEO

W tej tabeli przechowywane są informacje dotyczące stacji meteorologicznych (METEO) znajdujących się w systemie.

Kolumna	Opis
<b>device</b>	Identyfikator stacji METEO.
<b>period</b>	Okres zbierania danych ze stacji METEO (w sekundach). W każdym okresie generuje się nowy rekord dla każdego czujnika w tabeli METEOSENSORDATA. Rekord ten zawiera informacje zebrane przez dany czujnik w tym okresie czasu.

#### 4.5. METEOSENSOR

W tej tabeli przechowywane są dane dotyczące czujników w stacjach METEO znajdujących się w systemie. Czujniki są urządzeniami podłączonymi do stacji meteorologicznych, które dokonują konkretnych pomiarów pogodowych (temperatura, wilgotność, opad atmosferyczny, stan drogi, etc.).

Kolumna	Opis
device	Identyfikator stacji METEO.
sensor	Identyfikator czujnika.
description	Opis czujnika.
units	Jednostka miary czujnika, np. °C, %, mm/h, etc.

#### 4.6. METEOSENSORDATA

W tej tabeli przechowywane są dane historyczne z czujników w stacjach METEO systemu. Dla każdego okresu integracji (zdefiniowanego przez pole METEO.period) w tabeli generowany jest jeden rekord dla danego czujnika.

Kolumna	Opis
device	Identyfikator stacji METEO.
sensor	Identyfikator czujnika { 0 : opad atmosferyczny, 1 : Widzialność pozioma, 2 : Prędkość wiatru, 3 : Kierunek wiatru, 4 : Temperatura powietrza, 5 : Wilgotność powietrza, 6 : Temperatura punktu rosy, 7 : Temperatura jezdni, 8 : Stężenie soli, 9 : Temperatura zamarzania , 10 : Grubość warstwy wody, 11 : Stan nawierzchni }.
timestamp	Data zebrania danych czujnika.
value	Wartość czujnika Stan nawierzchni { 0 : Sucha, 1 : Zasolona, 2: Wilgotna, 3 : Mokra, 4: Zamarzająca, 5 : Szron, 6 : Suchy śnieg, 7: Suchy lód, 8 : Mokry śnieg, 9 : Lód }.
period	Okres integracji danych (w sekundach).

#### 4.7. RADAR

W tej tabeli przechowywane są informacje dotyczące urządzeń radarowych systemu. Urządzenie radarowe obejmuje swoim zasięgiem jeden lub dwa pasy ruchu, z których zbierane są dane dotyczące ruchu (liczba pojazdów poruszających się po danym pasie ruchu, średnia prędkość pojazdów, średnia zajętość pasa, klasyfikacja ze względu na długość lub prędkość wykrytych pojazdów, etc.).

Kolumna	Opis
device	Identyfikator urządzenia radarowego.
period	Okres zbierania danych z urządzenia RADAR (w sekundach). W każdym okresie generuje się: dla każdego pasa ruchu nowy rekord w tabeli RADARLANEDATA, rekord dla każdej zdefiniowanej klasy długości w tabeli RADARLANEDATALENGTH oraz rekord dla każdej zdefiniowanej klasy prędkości w tabeli RADARLANEDATASPEED. Rekordy te zawierają dane zebrane w danym okresie czasu dotyczące każdego pasa ruchu.
lengths	Liczba zdefiniowanych klas długości.
speeds	Liczba zdefiniowanych klas prędkości.

#### 4.8. RADARLANE

Jeden radar może kontrolować kilka pasów ruchu, w tej tabeli przechowywane są informacje dotyczące pasów ruchu obsługiwanych przez różne urządzenia radarowe systemu.

Kolumna	Opis
device	Identyfikator urządzenia radarowego.
lane	Identyfikator pasa ruchu.
description	Opis pasa ruchu.

#### 4.9. RADARLANEDATA

W tej tabeli przechowywane są dane historyczne dotyczące ruchu pojazdów na pasach obsługiwanych przez urządzenia radarowe systemu. Dla każdego okresu integracji (zdefiniowanego przez pole RADAR.period) generowany jest w tej tabeli jeden rekord dla jednego pasa ruchu.

Kolumna	Opis
device	Identyfikator urządzenia radarowego.
lane	Identyfikator pasa ruchu.
timestamp	Data zebrania danych z pasa ruchu.
speed	Średnia prędkość pojazdów wykrytych w okresie integracji (w km/h).
occupancy	Średnia zajętość pasa ruchu (w %) przez pojazdy wykryte w okresie integracji.
vehicles	Ilość pojazdów wykrytych w okresie integracji. Zawiera zarówno pojazdy, które mogły być zaklasyfikowane w którejkolwiek z klas długości, jak również te, których nie udało się sklasyfikować.
notClassified	Ilość pojazdów wykrytych o okresie integracji, których nie udało się sklasyfikować w żadnej ze zdefiniowanych klas długości.
period	Okres integracji danych (w sekundach).

#### 4.10. LENGTHCLASSIFICATION

W tej tabeli przechowywane są klasy długości pojazdów zdefiniowane dla każdego urządzenia radarowego. Każda klasa określona jest przez dolną i górną granicę, np. urządzenie radarowe z identyfikatorem 1111 posiada zdefiniowane 2 klasy długości: pierwsza od 0 dm do 20 dm oraz druga od 20 dm do 999 dm. Po wykryciu pojazdu przez urządzenie radarowe oraz obliczeniu jego długości, klasyfikuje się go do jednej z dwóch zdefiniowanych klas.

Kolumna	Opis
device	Identyfikator urządzenia radarowego.
identifier	Identyfikator klasy.
description	Opis klasyfikacji.
lowerLimit	Dolna granica długości danej klasy (w dm).
upperLimit	Górna granica długości danej klasy (w dm).

#### 4.11. RADARLANEDATALENGTH

W tej tabeli przechowywane są, dla każdego pasa ruchu obsługiwane przez urządzenie radarowe, dla każdego okresu integracji oraz dla każdej klasy długości, informacje dotyczące liczby pojazdów wykrytych na danym pasie, w danym okresie i przynależących do danej klasy. Dla każdego rekordu tabeli RADARLANEDATA istnieją rekordy w tabeli RADAR.lengths, po jednym dla każdej zdefiniowanej klasy długości.

Kolumna	Opis
device	Identyfikator urządzenia radarowego.
lane	Identyfikator pasa ruchu.
timestamp	Data zebrania danych z pasa ruchu.
classification	Identyfikator klasy długości.
vehicles	Liczba pojazdów poruszających się po pasie ruchu w okresie integracji, które przynależą do danej klasy długości.

#### 4.12. ANPR

W tej tabeli przechowywane są informacje dotyczące różnych urządzeń ANPR należących do systemu.

Kolumna	Opis
device	Identyfikator urządzenia ANPR.
lanes	Liczba pasów ruchu obsługiwanych przez urządzenie ANPR.
notifyBlackList	Powiadamianie o pojazdach wykrytych przez urządzenie ANPR, które znajdują się na jego czarnej liście { 0 : Nie, 1 : Tak}.
notifyDetections	Powiadamianie o wszystkich pojazdach wykrytych przez urządzenie ANPR { 0 : Nie, 1 : Tak}.
notifyTime	Czas (w sekundach), po którym nie powiadamia się już o wykryciu, jeśli data wykrycia pojazdu jest wcześniejsza niż data aktualna minus ten czas
blackListAlarmTime	Czas (w sekundach), w którym będzie aktywny alarm wykrycia pojazdu znajdującego się na czarnej liście.
nonAuthorizedAlarmTime	Czas (w sekundach), w którym będzie aktywny alarm wykrycia pojazdu nieuprawnionego.



#### 4.13. ANPRLANE

Urządzenie ANPR może wykrywać tablice rejestracyjne pojazdów poruszających się po więcej niż jednym pasie ruchu. W tej tabeli przechowywane są informacje dotyczące pasów ruchu obsługiwanych przez różne urządzenia ANPR znajdujące się w systemie.

Kolumna	Opis
device	Identyfikator urządzenia ANPR.
lane	Identyfikator pasa ruchu obsługiwanego przez urządzenie ANPR.
description	Opis pasa ruchu.

#### 4.14. PLATEDETECTION

W tej tabeli przechowywane są dane historyczne dotyczące detekcji pojazdów wykonanych przez poszczególne urządzenia ANPR zainstalowane w systemie. W tej tabeli przechowywane są informacje dotyczące detekcji pojazdów znajdujących się na białej liście. Detekcje pojazdów z białej listy przechowywane są w tabeli *Whitelistdetection*, która znajduje się na osobnym serwerze.

Kolumna	Opis
device	Identyfikator urządzenia ANPR.
timestamp	Data wykrycia pojazdu.
lane	Identyfikator pasa ruchu, na którym został wykryty pojazd.
plate	Tablica rejestracyjna pojazdu.
speed	Prędkość pojazdu.
reliability	Wiarygodność odczytu tablicy (w %).
country	Identyfikator kraju, do którego przynależy tablica.
photosPath	Katalog, w którym przechowywane są zdjęcia z detekcji pojazdu.
BlackList	Wskazuje, czy detekcja dotyczy pojazdu z czarnej listy {0 = nie, 1 = tak}.
Downloaded	Wskazuje, czy obraz został pobrany.
Bestplate	Wskazuje indeks obrazu z najlepszą wykorzystaną do odczytu tablicy jakością możliwą dla danego urządzenia.
Voivodeship	Województwo, do którego przynależy tablica.
Powiat	Powiat, do którego przynależy tablica.
City	Miasto, do którego przynależy tablica.
District	Dzielnica, do której przynależy tablica.

#### 4.15. ROUTE

W tej tabeli przechowywane są informacje dotyczące tras i odcinków zdefiniowanych w systemie. Trasą nazywamy, znajdujący się między dwoma określonymi punktami, zbiór następujących po sobie odcinków drogi. Odcinek drogi to przestrzeń między dwoma kolejnymi urządzeniami ANPR.

Kolumna	Opis
identifier	Identyfikator trasy.
creationLogin	Użytkownik, który utworzył trasę.
creationTime	Data utworzenia trasy.
direction	Kierunek trasy.
description	Opis trasy.
fixedInterval	Interwał czasu (w sekundach), który system wykorzystuje do gromadzenia i analizowania danych oraz przedstawiania wyników, kiedy trasa jest w trybie interwały stałego lub, kiedy znajduje się w trybie interwału okresowego i nie istnieje okres, który zawierałby aktualną datę i godzinę.
length	Długość trasy w metrach.
modelTime	Czas (w sekundach), który potrzebny jest na przebycie trasy w normalnych warunkach, to znaczy, przy płynnym ruchu i z maksymalną dozwoloną prędkością.
clientVisible	Wskazuje, czy trasa jest widoczna dla aplikacji klienckich { 0 : niewidoczna, 1 : widoczna }
webVisible	Wskazuje, czy trasa jest widoczna dla aplikacji internetowych { 0 : niewidoczna, 1 : widoczna }
mobileVisible	Wskazuje, czy trasa jest widoczna dla aplikacji mobilnych { 0 : niewidoczna, 1 : widoczna }
workingMode	Tryb obliczania czasu przejazdu trasy { 0 : Stały, 1 : Okresowy } W trybie interwału stałego, algorytm obliczania czasu przejazdu zawsze wykorzystuje interwał stały, np. 15 minutowy. W trybie interwału okresowego, algorytm obliczania czasu przejazdu może wykorzystywać różne czasu obliczeń zdefiniowane między znacznikami czasu. Znaczniki czasu przechowywane są w tabeli ROUTEINTERVAL.

#### 4.16. SECTION

W tej tabeli przechowywane są zdefiniowane w systemie odcinki dróg. Odcinek drogi to przestrzeń między dwoma kolejnymi urządzeniami ANPR.

Kolumna	Opis
identifier	Identyfikator odcinka.
direction	Kierunek odcinka.
description	Opis odcinka.
fixedInterval	Interwał czasu (w sekundach), który system wykorzystuje do gromadzenia i analizowania danych oraz przedstawiania wyników, kiedy odcinek jest w trybie interwału stałego lub, kiedy znajduje się w trybie interwału okresowego i nie istnieje okres, który zawierałby aktualną datę i godzinę.
length	Długość odcinka w metrach.
modelTime	Czas (w sekundach), który potrzebny jest na przebycie odcinka w normalnych warunkach, to znaczy, przy płynnym ruchu i z maksymalną dozwoloną prędkością.
minIgnoreTime	Czas przejazdu (w sekundach), poniżej którego dane dotyczące czasu przejazdu nie będą uwzględniane przy obliczaniu czasu przejazdu dla danego odcinka.
maxIgnoreTime	Czas przejazdu (w sekundach), powyżej którego dane dotyczące czasu przejazdu nie będą uwzględniane przy obliczaniu czasu przejazdu dla danego odcinka.
maxSpeed	Maksymalna dopuszczalna prędkość na odcinku. Jeśli na danym odcinku występują różne ograniczenia prędkości, dopuszczalna maksymalna prędkość będzie średnią ważoną odległością.
workingMode	Tryb obliczania czasu przejazdu odcinka { 0 : Stały, 1 : Okresowy } W trybie interwału stałego, algorytm obliczania czasu przejazdu zawsze wykorzystuje interwał stały, np. 15 minutowy. W trybie interwału okresowego, algorytm obliczania czasu przejazdu może wykorzystywać różne czasy obliczeń zdefiniowane między znacznikami czasu. Znaczniki czasu przechowywane są w tabeli SECTIONINTERVAL.
matchings	Minimalna liczba prawidłowych dopasowań na odcinku w okresie określonym w algorytmie obliczania czasu przejazdu, potrzebna do obliczenia czasu, który potrzebny jest na przebycie danego odcinka.

Dopasowaniem jest skojarzenie tablicy rejestracyjnej pojazdu przez dwa urządzenia ANPR, które tworzą odcinek, w którym został wykryty pojazd. Detekcja musi nastąpić najpierw przez urządzenie ANPR punktu początkowego i następnie przez urządzenie punktu końcowego odcinka.

#### 4.17. ANPRLANESECTION

Odcinek drogi jest przestrzenią pomiędzy dwoma kolejnymi urządzeniami ANPR (początkowym i końcowym), jednak każde urządzenie ANPR rejestruje jedynie określone pasy ruchu- te, które posiadają ten sam kierunek, co dany odcinek. W tej tabeli przechowywane są informacje dotyczące relacji między pasami ruchu przypisanymi do danego urządzenia ANPR oraz odcinkami, do których przynależą, przy wskazaniu dla każdego pasa czy dane urządzenie ANPR jest punktem początkowym czy końcowym określonego odcinka.

Kolumna	Opis
section	Identyfikator odcinka.
anpr	Identyfikator urządzenia ANPR.
lane	Identyfikator pasa ruchu rejestrowanego przez ANPR.
type	{ 0 : początek odcinka, 1 : koniec odcinka }

#### 4.18. SECTIONROUTE

Trasa jest to zbiór następujących po sobie odcinków drogi, z których każdy ma swoje określone położenie. W tej tabeli przechowywane są informacje dotyczące relacji między odcinkami i trasami, do których te przynależą oraz informacje dotyczące położenia tych odcinków na każdej trasie.

Kolumna	Opis
route	Identyfikator trasy.
section	Identyfikator odcinka.
orderNumber	Numer porządkowy (odpowiadający położeniu) danego odcinka na trasie.

#### 4.19. ROUTEINTERVAL

Dla każdej ze zdefiniowanych tras możliwe jest określenie, czy powinny one pracować z interwałem stałym czy z interwałami okresowymi. Interwał to przedział czasowy, który wykorzystuje system do gromadzenia i analizowania danych oraz do przedstawiania wyników.

W trybie interwału stałego, algorytm czasu przejazdu wykorzystuje zawsze interwał stały, na przykład 15 minutowy.

Kolumna	Opis
route	Identyfikator trasy.
startTime	Data i godzina rozpoczęcia okresu.
endTime	Data i godzina zakończenia okresu.
interval	Interwał czasowy w sekundach wykorzystywany przez algorytm obliczania czasu przejazdu, kiedy trasa jest skonfigurowana w trybie okresowym.

#### 4.20. SECTIONINTERVAL

Dla każdego ze zdefiniowanych odcinków możliwe jest określenie, czy powinny one pracować z interwałem stałym czy z interwałami okresowymi. Interwał to przedział czasowy, który wykorzystuje system do gromadzenia i analizowania danych oraz do przedstawiania wyników. Tabela ta odpowiada tabeli ROUTEINTERVAL, jednak zamiast informacji dotyczących tras zawiera informacje dotyczące odcinków.

Kolumna	Opis
section	Identyfikator odcinka.
startTime	Data i godzina rozpoczęcia okresu.
endTime	Data i godzina zakończenia okresu.
interval	Interwał czasowy w sekundach wykorzystywany przez algorytm obliczania czasu przejazdu, kiedy odcinek jest skonfigurowany w trybie okresowym.

#### 4.21. ROUTEDATA

W tej tabeli przechowywane są dane historyczne dotyczące czasów przejazdu obliczonych dla poszczególnych tras zdefiniowanych w systemie.

Kolumna	Opis
route	Identyfikator trasy.
timestamp	Data wykonania obliczenia czasu przejazdu.
time	Czas przejazdu w sekundach obliczony dla danej trasy w danej dacie.
trafficState	Relacja między obliczonym czasem przejazdu oraz rzeczywistym czasem potrzebnym do przebycia trasy w warunkach normalnych (ROUTE.modelTime). { 0 : Płynny, 1 : Dosyć płynny, 2 : Między 25%-50% powyżej normalnego czasu, 3 : Między 50%-75%, 4 : Między 75%-100%, 5 : Między 100%-125%, 6 : Między 125%-150%, 7 : Między 150%-175%, 8 : Między 175%-200%, 9 : Powyżej 200%, 10: Droga zamknięta}.

#### 4.22. SECTIONDATA

W tej tabeli przechowywane są dane historyczne dotyczące czasów przejazdu obliczonych dla poszczególnych odcinków zdefiniowanych w systemie.

Kolumna	Opis
section	Identyfikator odcinka.
timestamp	Data wykonania obliczenia czasu przejazdu.
time	Czas przejazdu w sekundach obliczony dla danego odcinka w danej dacie.
trafficState	Relacja między obliczonym czasem przejazdu oraz rzeczywistym czasem potrzebnym do przebycia odcinka w warunkach normalnych (SECTION.modelTime). { 0 : Płynny, 1 : Dosyć płynny, 2 : Między 25%-50% powyżej normalnego czasu, 3 : Między 50%-75%, 4 : Między 75%-100%, 5 : Między 100%-125%, 6 : Między 125%-150%, 7 : Między 150%-175%, 8 : Między 175%-200%, 9 : Powyżej 200%, 10: Droga zamknięta}
matchings	Liczba dopasowań wykonanych i wykorzystanych do obliczenia czasu przejazdu danego odcinka.

#### 4.23. MATCHING

W tej tabeli przechowywane są informacje dotyczące dopasowania tablic rejestracyjnych pojazdów, które wykryte zostały na poszczególnych odcinkach. Dopasowaniem jest skojarzenie tablicy rejestracyjnej pojazdu przez dwa urządzenia ANPR, które tworzą ten sam odcinek, w którym został wykryty pojazd. Detekcja musi nastąpić najpierw przez urządzenie ANPR punktu początkowego i następnie przez urządzenie punktu końcowego odcinka.

Kolumna	Opis
section	Identyfikator odcinka.
Anpr1	Identyfikator urządzenia ANPR w punkcie początkowym odcinka, gdzie wykryty został pojazd.
Lane1	Identyfikator pasa ruchu obsługiwanego przez urządzenie ANPR znajdujące się w punkcie początkowym odcinka gdzie wykryty został pojazd.
anpr2	Identyfikator urządzenia ANPR w punkcie końcowym odcinka gdzie wykryty został pojazd.
lane2	Identyfikator pasa ruchu obsługiwanego przez urządzenie ANPR znajdujące się w punkcie końcowym odcinka gdzie wykryty został pojazd.
originTime	Data wykrycia pojazdu przez urządzenie ANPR punktu początkowego odcinka.
destinationTime	Data wykrycia pojazdu przez urządzenie ANPR punktu końcowego odcinka.
time	Czas w sekundach, który zajęło pojazdowi przejechanie odcinka.
length	Długość odcinka gdzie wykryty został pojazd (w metrach).
speed	Średnia prędkość pojazdu.

#### 4.24. SYSTEMUSER

W tej tabeli przechowywane są informacje dotyczące użytkowników systemu. Użytkownik systemu, korzystając ze swojej nazwy użytkownika i hasła (username i password) będzie mógł uzyskać dostęp do systemu pod warunkiem, że nie zostanie wyłączony.

Kolumna	Opis
username	Identyfikator użytkownika, który będzie wykorzystywany do uzyskania dostępu do systemu.
password	Hasło użytkownika.
name	Imię użytkownika.
username	Nazwisko użytkownika.
email	E-mail użytkownika.
phone	Numer telefonu użytkownika.
company	Firma, do której przynależy użytkownik.
jobtitle	Stanowisko użytkownika.
address	Adres użytkownika.
profile	Profil użytkownika { 0 : operator, 1 : administrator }.
enabled	Status użytkownika { 0 : Wyłączony (nie może uzyskać dostępu do systemu), 1 : Włączony }.
creationtime	Data utworzenia użytkownika.
deletetime	Data usunięcia użytkownika.
lastupdate	Data ostatniej modyfikacji informacji dotyczącej użytkownika.

#### 4.25. SYSTEMUSERSTATUS

W tej tabeli przechowywane są informacje dotyczące statusu użytkowników systemu.

Kolumna	Opis
timestamp	Data zmiany statusu użytkownika systemu.
username	Identyfikator użytkownika, który będzie wykorzystywany do uzyskania dostępu do systemu.
password	Hasło użytkownika.
name	Imię użytkownika.
username	Nazwisko użytkownika.
email	E-mail użytkownika.
phone	Numer telefonu użytkownika.
company	Firma, do której przynależy użytkownik.
jobtitle	Stanowisko użytkownika.
address	Adres użytkownika.
profile	Profil użytkownika { 0 : operator, 1 : administrator }.
enabled	Status użytkownika { 0 : Wyłączony (nie może uzyskać dostępu do systemu), 1 : Włączony }.
creationtime	Data utworzenia użytkownika.

<b>deletetime</b>	Data usunięcia użytkownika.
<b>lastupdate</b>	Data ostatniej modyfikacji informacji dotyczącej użytkownika.

#### 4.26. SYSTEMUSERACCESS

W tej tabeli przechowywane są informacje dotyczące dostępów do systemu poszczególnych użytkowników.

Kolumna	Opis
<b>username</b>	Identyfikator użytkownika, który zalogował się do systemu.
<b>startTime</b>	Data zalogowania się do systemu.
<b>console</b>	Nazwa urządzenia (stacja robocza, laptop, itd.), z której użytkownik zalogował się do systemu.
<b>endTime</b>	Data wylogowania się z systemu, początkowo pole to nie będzie miało żadnej wartości.

#### 4.27. CAMERA

W tej tabeli przechowywane są informacje dotyczące kamer znajdujących się w systemie.

Kolumna	Opis
<b>device</b>	Identyfikator kamery.
<b>downloadPeriod</b>	Okres (w sekundach) pobierania obrazów z kamery, pod warunkiem, że tryb pobierania (downloadMode) będzie miał wartość 0 (automatyczny).
<b>downloadMode</b>	Tryb pobierania { 0 : Automatyczny, 1 : Manualny na żądanie operatora }.
<b>jpegCompression</b>	Procent kompresji pobieranego obrazu w formacie JPEG.
<b>jpegWidth</b>	Szerokość pobranego obrazu w pikselach.
<b>jpegHeight</b>	Wysokość pobranego obrazu w pikselach.
<b>jpegWebCompression</b>	Procent kompresji obrazu JPEG, który ma być wysłany na stronę www.
<b>jpegWebWidth</b>	Szerokość obrazu JPEG, który ma być wysłany na stronę www (w pikselach).
<b>jpegWebHeight</b>	Wysokość obrazu JPEG, który ma być wysłany na stronę www (w pikselach).
<b>jpegMobileCompression</b>	Procent kompresji obrazu JPEG, który ma być wysłany do aplikacji mobilnych.
<b>jpegMobileWidth</b>	Szerokość obrazu JPEG, który ma być wysłany do aplikacji mobilnych (w pikselach).
<b>jpegMobileHeight</b>	Wysokość obrazu JPEG, który ma być wysłany do aplikacji mobilnych (w pikselach).
<b>motion</b>	Typ kamery { 0 : obrotowa, 1 : stała }.
<b>motionProtocol</b>	Protokół sterujący kamery (PelcoP, PelcoD, VCL, Bosch, itd.)
<b>lastPhotoPath</b>	Katalog, w którym przechowywany jest ostatni pobrany obraz.
<b>lastPhotoTime</b>	Data pobrania ostatniego obrazu.



#### 4.28. IMAGELOG

W tej tabeli przechowywane są dane historyczne dotyczące obrazów pobranych z kamer zainstalowanych w systemie.

Kolumna	Opis
device	Identyfikator kamery.
timestamp	Data pobrania obrazu.
imagePath	Katalog, w którym przechowywany jest pobrany obraz.
deleted	Data usunięcia kopii obrazu z serwera, data ta nie wskazuje czy obraz został usunięty z serwera przechowywującego dane.

#### 4.29. VMS

W tej tabeli przechowywane są informacje dotyczące tablic i znaków VMS znajdujących się w systemie.

Kolumna	Opis
device	Identyfikator tablicy VMS.
width	Szerokość tablicy VMS w pikselach.
height	Wysokość tablicy VMS w pikselach.
workingMode	Tryb pracy { 0 : Automatyczny, 1 : Manualny }

#### 4.30. VMSGGRAPHICFOLDER

W tej tabeli przechowywane są dane dotyczące nazw folderów i różnych rozdzielczości obrazów, które wykorzystywane są do organizowania grafik VMS.

Kolumna	Opis
resolution	Wskazuje szerokość i wysokość grafiki w pikselach, które wykorzystywane są do organizowania folderów.
folder	Nazwa folderu, w którym przechowywana jest grafika. Pozwala to na grupowanie grafik w folderach (zagrożenia, nakazy, informacje, czas przejazdu, itd.), aby ułatwić do nich dostęp z aplikacji.

#### 4.31. VMSGGRAPHIC

W tej tabeli przechowywane są informacje dotyczące grafik, które mogą być wyświetlane na tablicach VMS.

Kolumna	Opis
identifier	Identyfikator grafiki.
resolution	Rozdzielczość grafiki.
graphicFolder	Nazwa folderu, w którym przechowywane są grafiki. Pozwala to na grupowanie grafik w folderach (zagrożenia, nakazy, informacje, czas przejazdu, itd.), aby ułatwić do nich dostęp z aplikacji.
graphicName	Nazwa grafiki.
address	Adres pamięci tablicy VMS, w której przechowywana jest grafika.
width	Szerokość grafiki w pikselach.
height	Wysokość grafiki w pikselach.
image	Informacja dotycząca kolorów pikseli, które tworzą komunikat. Piksele są zwracane uporządkowane w linie i kolumny, z góry na dół i od lewej do prawej. Zwracane są bajty height x width (wysokość x szerokość), gdzie każdy bajt reprezentuje kolor piksela { 0 : Biały, 1 : Żółty, 2 : Czerwony, 3 : Zielony, 4 : Niebieski, 5 : Czarny}.
priority	Wskazuje priorytet makra { 1 do 5, gdzie 1 wskazuje makro o najwyższym priorytecie}.

#### 4.32. VMSMACRO

W tej tabeli przechowywane są makra wyświetlane na tablicy VMS. Do dyspozycji są trzy typy makr: CZAS PRZEJAZDU - makro to aktywuje się kiedy przekroczony zostaje wskazany czas przejazdu i włącza tablicę pokazując czas przejazdu określonej trasy w pozycji tagu %n, WARUNKI METEOROLOGICZNE - gdzie możliwe jest wybranie tablicy, która się włączy w zależności od konkretnych warunków meteorologicznych oraz typ WARTOSCI METEO - gdzie wskazuje się włączenie przez system tablicy w zależności od konkretnych wartości meteorologicznych skontrastowanych z tymi otrzymanymi z czujników w stacjach meteorologicznych i w zależności od wybranego typu operatora.

Kolumna	Opis
identifier	Identyfikator makro, którym jest tag dodawany podczas tworzenia tablicy typu %n.
resolution	Rozdzielczość makra.
graphicFolder	Wskazuje, do którego folderu przynależy makro.
graphicName	Wskazuje nazwę grafiki, do której przynależy makro.
macrotype	Wskazuje typ makra {1 : CZAS PRZEJAZDU, 2 : WARUNKI METEOROLOGICZNE, 3 : WARTOSCI METEO} , gdzie
substitution	Wskazuje, do którego urządzenia odnosi się to makro.

#### 4.33. VMSMACROMTO

W tej tabeli przechowywane są informacje o czujnikach wybranych dla każdego makra VMS, wartości, które powinno się brać pod uwagę, aby móc zdecydować czy aktywować dane makro czy nie, jak również położenie makra VMS.

Kolumna	Opis
identifier	Identyfikator makro, którym jest tag dodawany podczas tworzenia tablicy, np. typ %n.
resolution	Rozdzielczość makra.
graphicFolder	Wskazuje, do którego folderu przynależy makro.
graphicName	Wskazuje nazwę grafiki, do której przynależy makro.
sensor	Identyfikator czujnika stacji meteo, z którego korzysta makro.
oper	Operator logiczny służący do sprawdzenia wartości (większy niż >, mniejszy niż < lub równy).
value	Wskazuje wartość określoną jako próg, po przekroczeniu którego makro się aktywuje.
active	Wskazuje, czy czujnik jest aktywny dla aktualnego makra {1 aktywny, 0 nieaktywny}. Jeśli czujnik jest dezaktywowany oznacza to, że nie jest brany pod uwagę.

#### 4.34. MODULO

W tej tabeli przechowywane są informacje o poszczególnych modułach systemu wraz z informacją, czy są one aktywne czy nie.

Kolumna	Opis
identifier	Identyfikator modułu.
description	Opis modułu.
enabled	Wskazuje, czy moduł jest włączony czy wyłączony {1 włączony, 0 wyłączony}.

#### 4.35. MODULOFEATURE

W tej tabeli przechowywane są informacje o poszczególnych funkcjonalnościach modułów.

Kolumna	Opis
moduleId	Identyfikator modułu.
feature	Identyfikator funkcjonalności.
description	Opis funkcjonalności.

#### 4.36. MODULEFEATUREUSERRIGHT

W tej tabeli przechowywane są informacje o uprawnieniach, które przypisane są użytkownikom w poszczególnych modułach wraz z informacją, czy dane uprawnienie jest włączone czy wyłączone. Możliwe jest również wskazanie daty aktywowania konkretnego uprawnienia.

Kolumna	Opis
moduleId	Identyfikator modułu.
Feature	Opis funkcjonalności do jakiej odnosi się uprawnienie.
username	Użytkownik, do którego przypisuje się uprawnienie do korzystania z danej funkcjonalności.
startTime	Data początkowa, od której użytkownik ma aktywne uprawnienie.
endTime	Data wygaśnięcia uprawnienia.
enabled	Wskazuje, czy użytkownik jest włączony czy wyłączony {1 włączony, 0 wyłączony}.

#### 4.37. VEHICLE

W tej tabeli przechowywane są informacje o pojazdach PSP.

Kolumna	Opis
plate	Identyfikator i tablica rejestracyjna pojazdu.
organism	Identyfikator organizacji lub instytucji.
mark	Marka pojazdu.
model	Model pojazdu.
colour	Kolor pojazdu.
country	Kraj, z którego pochodzi pojazd.
owner	Właściciel pojazdu.

#### 4.38. TABLELE Z KODAMI IDENTYFIKUJĄCYMI POCHODZENIE POLSKICH TABLIC REJESTRACYJNYCH

Od 1 maja 2000 r. wydawane są w Polsce tablice rejestracyjne posiadające czarne znaki na białym tle. Dla każdego miasta/ powiatu przypisane są dwie lub trzy wyróżniające go litery. Istnieją różne sposoby tworzenia numerów rejestracyjnych na tablicach (# oznacza literę, \* oznacza cyfrę):

- Dwie litery, istnieją trzy możliwe kombinacje: ## \*\*\*\*\* , ## \*\*\*\*\*# lub ## \*\*\*\*\*##
- Trzy litery, istnieje sześć możliwych kombinacji: ### \*\*\*\*\* , ### \*\*\*\*\*# , ### \*\*\*\*\*#\* , ### \*\*\*\*\*#\* , ### \*\*\*\*\*#\* lub ### \*\*\*\*\*#\* .
- Motocykle i ciągniki posiadają kombinację ## \*\*\*\*\* lub ## \*\*\*\*\*# dla miast/ powiatów z wyróżnikiem dwuliterowym, a w przypadku kombinacji tablic z miast/ powiatów z wyróżnikiem trzyliterowym możliwe są takie same kombinacje jak w przypadku samochodów (białe tablice z czarnymi znakami).
- Samochody zabytkowe posiadają tablice z konfiguracją ## \*\*\*\*\* (wyznacznik dwuliterowy) lub ### \*\*\*\*\*# (wyznacznik trzyliterowy), z żółtym tłem i czarnymi literami.

Poniżej przedstawione są tabele zawierające kody tablic oznaczające miasta, powiaty i dzielnice, pogrupowane według województw:

<b>Dolnośląskie</b>		<b>Kujawsko-Pomorskie</b>	
<b>Kod</b>	<b>Powiat / Miasto /Dzielnica</b>	<b>Kod</b>	<b>Powiat / Miasto /Dzielnica</b>
DB	Wałbrzych (miasto)	CAL	Aleksandrów Kujawski
DBA	Wałbrzych (powiat)	CB	Bydgoszcz (miasto)
DBL	Bolesławiec	CBR	Brodnica
DDZ	Dzierżoniów	CBY	Bydgoszcz (powiat)
DGL	Głogów	CCH	Chełmno
DGR	Góra	CG	Grudziądz (miasto)
DJ	Jelenia Góra (miasto)	CGD	Golub-Dobrzyn
DJA	Jawor	CGR	Grudziądz (powiat)
DJE	Jelenia Góra (powiat)	CIN	Inowrocław
DKA	Kamienna Góra	CLI	Lipno
DKL	Kłodzko	CMG	Mogilno
DL	Legnica (miasto)	CNA	Nakło
DLB	Lubań	CRA	Radziejów
DLE	Legnica (powiat)	CRY	Rypin
DLU	Lubin	CSE	Sępólno Krajeńskie
DLW	Lwówek Śląski	CSW	Świecie
DMI	Milicz	CT	Toruń (miasto)
DOA	Oława	CTR	Toruń (powiat)
DOL	Oleśnica	CTU	Tuchola
DPL	Polkowice	CW	Włocławek (miasto)
DSR	Środa Śląska	CWA	Wąbrzeźno
DST	Strzelin	CWL	Włocławek (powiat)
DSW	Świdnica	CZN	Żnin
DTR	Trzebnica		
DW	Wrocław (miasto)		
DWL	Wrocław		
DWR	Wrocław (powiat)		
DZA	Ząbkowice Śląskie		
DZG	Zgorzelec		
DZL	Złotoryja		

Łódzkie		Lubelskie	
Kod	Powiat / Miasto / Dzielnica	Kod	Powiat / Miasto / Dzielnica
EBE	Bełchatów	LB	Biała Podlaska (miasto)
EBR	Brzeziny	LBI	Biała Podlaska (powiat)
EKU	Kutno	LBL	Biłgoraj
EL	Łódź (miasto)	LC	Chełm (miasto)
ELA	Łask	LCH	Chełm (powiat)
ELC	Łowicz	LHR	Hrubieszów
ELE	Łęczyca	LJA	Janów Lubelski
ELW	Łódź Wschód (powiat Łódź Wschód)	LKR	Kraśnik
EOP	Opoczno	LKS	Krasnystaw
EP	Piotrków Trybunalski (miasto)	LLB	Lubartów
EPA	Pabianice	LLE	Łęczna
EPD	Poddębice	LLU	Łuków
EPI	Piotrków Trybunalski (powiat)	LOP	Opole Lubelskie
EPJ	Pajęczno	LPA	Parczew
ERA	Radomsko	LPU	Puławny
ERW	Rawa Mazowiecka	LRA	Radzyń Podlaski
ES	Skierniewice (miasto)	LRY	Ryki
ESI	Sieradz	LSW	Świdnik
ESK	Skierniewice (powiat)	LTM	Tomaszów Lubelski
ETM	Tomaszów Mazowiecki	LU	Lublin (miasto)
EWE	Wieruszów	LUB	Lublin (powiat)
EWI	Wieluń	LWL	Włodawa
EZD	Zduńska Wola	LZ	Zamość (miasto)
EZG	Zgierz	LZA	Zamość (powiat)

Lubuskie		Małopolskie	
Kod	Powiat / Miasto / Dzielnica	Kod	Powiat / Miasto / Dzielnica
FG	Gorzów Wielkopolski (miasto)	KBC	Bochnia
FGW	Gorzów Wielkopolski (powiat)	KBR	Brzesko
FKR	Krosno Odrzańskie	KCH	Chrzanów
FMI	Międzyrzecz	KDA	Dąbrowa Tarnowska
FNW	Nowa Sól	KGR	Gorlice
FSD	Strzelce Krajeńskie-Drezdenko	KLI	Limanowa
FSL	Słubice	KMI	Miechów
FSU	Sulęcín	KMY	Myślenice
FSW	Świebódzin	KN	Nowy Sącz (miasto)
FWS	Wschowa	KNS	Nowy Sącz (powiat)
FZ	Zielona Góra (miasto)	KNT	Nowy Targ
FZA	Żary	KOL	Olkusz
FZG	Żagań	KOS	Oświęcim
FZI	Zielona Góra (powiat)	KPR	Proszowice
		KR	Cracovia (miasto)
		KRA	Cracovia (powiat)
		KSU	Sucha Beskidzka
		KT	Tarnów (miasto)
		KTA	Tarnów (powiat)
		KTT	Tatrzański (Zakopane)
		KWA	Wadowice
		KWI	Wieliczka



Mazowieckie		Opolskie	
Kod	Powiat / Miasto /Dzielnica	Kod	Powiat / Miasto /Dzielnica
WA	Warszawa (Białołęka)	OB	Brzeg
WB	Warszawa (Bemowo)	OGL	Głubczyce
WBR	Białobrzegi	OK	Kędzierzyn-Koźle
WCI	Ciechanów	OKL	Kluczbork
WD	Warszawa (Bielany)	OKR	Krapkowice
WE	Warszawa (Mokotów)	ONA	Namysłów
WF	Warszawa (Praga Południe)	ONY	Nysa
WG	Garwolin	OOL	Olesno
WGM	Grodzisk Mazowiecki	OP	Opole (miasto)
WGR	Grójec	OPO	Opole (powiat)
WGS	Gostynin	OPR	Prudnik
WH	Warszawa (Praga Północ)	OST	Strzelce Opolskie
WI	Warszawa (Śródmieście)		
WJ	Warszawa (Targówek)		
WK	Warszawa (Ursus)		
WKZ	Kozienice		
WL	Legionowo		
WLI	Lipsko		
WLS	Łosice		
WM	Mińsk Mazowiecki		
WMA	Maków Mazowiecki		
WML	Mława		
WN	Warszawa (Ursynów)		
WND	Nowy Dwór Mazowiecki		
WO	Ostrołęka (miasto)		
WOR	Ostrów Mazowiecka		
WOS	Ostrołęka (powiat)		
WOT	Otwock		
WP	Płock (miasto)		
WPI	Piaseczno		
WPL	Płock (powiat)		
WPN	Płońsk		
WPR	Pruszków		
WPU	Pułtusk		
WPY	Przysucha		
WPZ	Przasnysz		
WR	Radom (miasto)		
WRA	Radom (powiat)		
WS	Siedlce (miasto)		
WSC	Sochaczew		
WSE	Sierpc		
WSI	Siedlce (powiat)		
WSK	Sokołów Podlaski		

WSZ	Szydłowiec		
WT	Warszawa (Wawer)		
WU	Warszawa (Ochota)		
WV	Wołomin		
WW	Wyróżnik na końcu K, L, M, N, V - Warszawa (Włochy)		
WW	Wyróżnik na końcu F, G, H, J, W - Warszawa (Wilanów)		
WW	Wyróżnik na końcu A, C, E, X, Y - Warszawa (Rembertów)		
WWE	Węgrów		
WWL	Wołomin		
WWY	Wyszków		
WX	Warszawa (Żoliborz)		
WX	***Y# - Warszawa (Wesoła)		
WY	Warszawa (Wola)		
WZ	Warszawski Zachodni(Powiat Warszawski Zachodni)		
WZU	Żuromin		
WZW	Zwoleń		
WZY	Żyrardów		

Podkarpackie		Podlaskie	
Kod	Powiat / Miasto /Dzielnica	Kod	Powiat / Miasto /Dzielnica
RBI	Bieszczadzki (Ustrzyki Dolne)	BAU	Augustów
RBR	Brzozów	BBI	Bielsk Podlaski
RDE	Dębica	BGR	Grajewo
RJA	Jarosław	BHA	Hajnówka
RJS	Jasło	BI	Białystok (miasto)
RK	Krosno (miasto)	BIA	Białystok (powiat)
RKL	Kolbuszowa	BKL	Kolno
RKR	Krosno (powiat)	BL	Łomża (miasto)
RLA	Łańcut	BLM	Łomża (powiat)
RLE	Leżajsk	BMN	Mońki
RLS	Lesko	BS	Suwałki (miasto)
RLU	Lubaczów	BSE	Sejny
RMI	Mielec	BSI	Siemiatycze
RNI	Nisko	BSK	Sokółka
RP	Przemyśl (miasto)	BSU	Suwałki (powiat)
RPR	Przemyśl (powiat)	BWM	Wysokie Mazowieckie
RPZ	Przeworsk	BZA	Zambrów
RRS	Ropczyce Sędziszów		
RSA	Sanok		
RSR	Strzyżów		
RST	Stalowa Wola		
RT	Tarnobrzeg (miasto)		
RTA	Tarnobrzeg (powiat)		
RZ	Rzeszów (miasto)		
RZE	Rzeszów (powiat)		

Pomorskie		Śląskie	
Kod	Powiat / Miasto /Dzielnica	Kod	Powiat / Miasto /Dzielnica
GA	Gdynia	SB	Bielsko-Biała (miasto)
GBY	Bytów	SBE	Będzin
GCH	Chojnice	SBI	Bielsko-Biała(powiat)
GCZ	Człuchów	SBL	Bieruń-Lędziny
GD	Gdańsk (miasto)	SC	Częstochowa (miasto)
GDA	Pruszcz Gdański (powiat)	SCI	Cieszyn
GKA	Kartuzy	SCZ	Częstochowa (powiat)
GKS	Kościerzyna	SD	Dąbrowa Górnicza
GKW	Kwidzyń	SG	Gliwice (miasto)
GLE	Lębork	SGL	Gliwice (powiat)
GMB	Malbork	SH	Chorzów
GND	Nowy Dwór Gdański	SI	Siemianowice Śląskie
GPU	Puck	SJ	Jaworzno
GS	Słupsk (miasto)	SJZ	Jastrzębie-Zdrój
GSL	Słupsk (powiat)	SK	Katowice
GSP	Sopot	SKL	Kłobuck
GST	Starogard Gdański	SL	Ruda Śląska
GSZ	Sztum	SLU	Lubliniec
GTC	Tczew	SM	Mysłowice
GWE	Wejherowo	SMI	Mikołów
		SMY	Myszków
		SO	Sosnowiec
		SPI	Piekary Śląskie
		SPS	Pszczyna
		SR	Rybnik (miasto)
		SRB	Rybnik (powiat)
		SRC	Racibórz
		SRS	Ruda Śląska aktualnie SL
		ST	Tychy (miasto)
		STY	Tychy (powiat) aktualnie SBL
		STA	Tarnowskie Góry
		SW	Świętochłowice
		SWD	Wodzisław Śląski
		SY	Bytom
		SZ	Zabrze
		SZA	Zawiercie
		SZO	Żory
		SZY	Żywiec

Świętokrzyskie		Warmińsko-Mazurskie	
Kod	Powiat / Miasto / Dzielnica	Kod	Powiat / Miasto / Dzielnica
TBU	Busko-Zdrój	NBA	Bartoszyce
TJE	Jędrzejów	NBR	Braniewo
TK	Kielce (miasto)	NDZ	Działdowo
TKA	Kazimierza Wielka	NE	Elbląg (miasto)
TKI	Kielce (powiat)	NEB	Elbląg (powiat)
TKN	Końskie	NEL	Ełk
TLW	Włoszczowa	NGI	Giżycko
TOP	Opatów	NGO	Gołdap
TOS	Ostrowiec-Świętokrzyski	NIL	Iława
TPI	Pińczów	NKE	Kętrzyn
TSA	Sandomierz	NLI	Lidzbark Warmiński
TSK	Skarżysko-Kamienna	NMR	Mrągowo
TST	Starachowice	NNI	Nidzica
TSZ	Staszów	NNM	Nowe Miasto Lubawskie
		NO	Olsztyn (miasto)
		NOE	Olecko
		NOG	Olecko-Gołdap aktualnie NOE lub NGO
		NOL	Olsztyn (powiat)
		NOS	Ostróda
		NPI	Pisz
		NSZ	Szczytno
		NWE	Węgorzewo

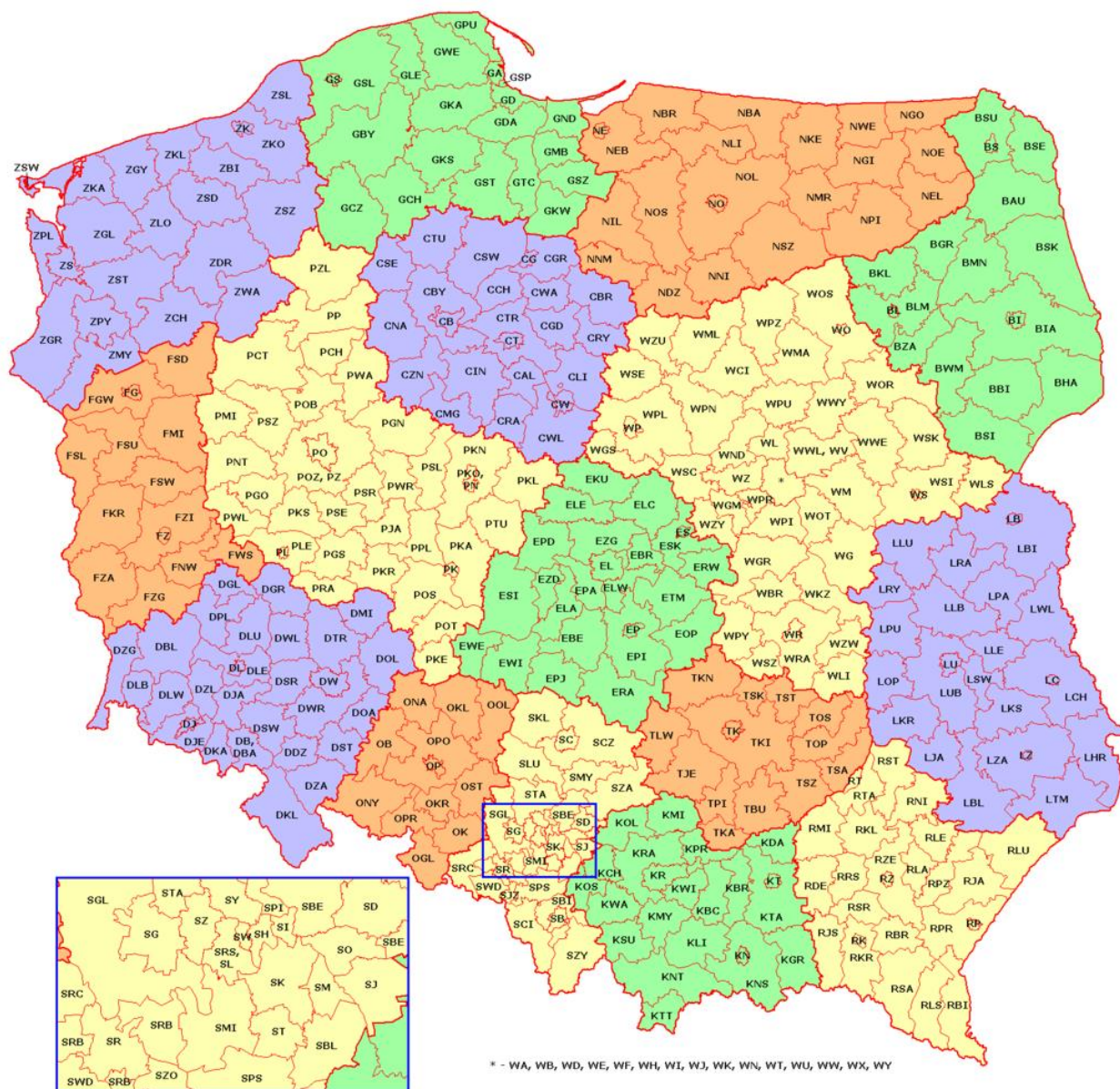
Wielkopolskie		Zachodniopomorskie	
Kod	Powiat / Miasto /Dzielnica	Kod	Powiat / Miasto /Dzielnica
PCH	Chodzież	ZBI	Białogard
PCT	Czarnków - Trzcianka	ZCH	Choszczno
PGN	Gniezno	ZDR	Drawsko Pomorskie
PGO	Grodzisk Wielkopolski	ZGL	Goleniów
PGS	Gostyń	ZGR	Gryfino
PJA	Jarocin	ZGY	Gryfice
PK	Kalisz (miasto)	ZK	Koszalin (miasto)
PKA	Kalisz (powiat)	ZKA	Kamień Pomorski
PKE	Kępno	ZKL	Kołobrzeg
PKL	Koło	ZKO	Koszalin (powiat)
PKN	Konin (powiat)	ZLO	Łobez
PKO	Konin (miasto) aktualnie PN	ZMY	Myślibórz
PKR	Krotoszyn	ZPL	Police (powiat)
PKS	Kościan	ZPY	Pyrzyce
PL	Leszno (miasto)	ZS	Szczecin
PLE	Leszno (powiat)	ZSD	Świdwin
PMI	Międzychód	ZSL	Sławno
PN	Konin (miasto)	ZST	Stargard Szczeciński
PNT	Nowy Tomyśl	ZSW	Świnoujście
PO	Poznań (miasto)	ZSZ	Szczecinek
POB	Oborniki	ZWA	Wałcz
POS	Ostrów Wielkopolski		
POT	Ostrzeszów		
POZ	Poznań (powiat) aktualnie PZ		
PP	Piła		
PPL	Pleszew		
PRA	Rawicz		
PSE	Śrem		
PSL	Słupca		
PSR	Środa Wlkp.		
PSZ	Szamotuły		
PTU	Turek		
PWA	Wągrowiec		
PWL	Wolsztyn		
PWR	Września		
PZ	Poznań (powiat)		
PZL	Złotów		

Poniżej przedstawiona jest tabela, w której przechowywane są kody tablic specjalnych, które nie odnoszą się do żadnego miasta, powiatu ani dzielnicy:

Kod	Opis
UA	samochody osobowe, osobowo-terenowe oraz specjalne na podwoziu osobowym (osobowo-terenowym)
UB	pojazdy bojowe
UC	samochody osobowo-ciężarowe (dostawcze)
UD	autobusy
UE	samochody ciężarowe i ciężarowo-terenowe o przeznaczeniu transportowym
UG	pojazdy specjalne na podwoziu ciężarowym (ciężarowo-terenowym)
UI	przyczepy transportowe
UJ	przyczepy specjalne
UK	motocykle
UL	pojazdy przyjmowane z gospodarki narodowej podczas mobilizacji (pojazdy posiadające obecnie "cywilne" numery rejestracji)
HA	Centralne Biuro Antykorupcyjne
HB	Biuro Ochrony Rządu
HC	Służba Celna
HK	Agencja Bezpieczeństwa Wewnętrznego i Agencja Wywiadu
HM	Służba Kontrwywiadu Wojskowego i Służba Wywiadu Wojskowego
HP	Policja
HS	Kontrola Skarbowa
HW	Straż Graniczna



Mapa powiatów, miast i dzielnic, z odpowiadającymi im oznaczeniami tablic rejestracyjnych:





#### 4.38.1. VOIVODESHIP

W tej tabeli przechowywane są informacje dotyczące województw, na które podzielona jest Polska.

Kolumna	Opis
name	Nazwa województwa.
description	Opis województwa.

#### 4.38.2. POWIAT

W tej tabeli przechowywane są informacje dotyczące powiatów, na które podzielona jest Polska oraz odpowiadające im oznaczenia tablic rejestracyjnych.

Kolumna	Opis
name	Nazwa powiatu.
voivodeship	Województwo, do którego przynależy powiat.
plateCode	Wyróżniki literowe tablicy rejestracyjnej odpowiadające danemu powiatowi.
description	Opis powiatu.

#### 4.38.3. CITY

W tej tabeli przechowywane są informacje dotyczące polskich miast oraz miast na prawach powiatu, dla których istnieją odpowiadające im oznaczenia tablic rejestracyjnych.

Kolumna	Opis
name	Nazwa miasta.
powiat	Powiat, do którego przynależy miasto.
plateCode	Wyróżniki literowe tablicy rejestracyjnej odpowiadające danemu miastu.
description	Opis miasta.

#### 4.38.4. DISTRICT

W tej tabeli przechowywane są informacje dotyczące dzielnic, które posiadają odpowiadające im oznaczenia tablic rejestracyjnych.

Kolumna	Opis
name	Nazwa dzielnicy.
city	Miasto, do którego przynależy dzielnica.
startPlateCode	Wyróżniki literowe na początku tablicy rejestracyjnej odpowiadające danemu powiatowi.
endPlateCode	Wyróżniki literowe na końcu tablicy rejestracyjnej odpowiadające danemu powiatowi. Jeśli pole to nie zawiera wartości oznacza to, że dana dzielnica nie posiada konkretnych wyróżników, które muszą znajdować się na końcu tablicy.
description	Opis dzielnicy.

#### 4.38.5. SPECIALPOLISHPLATE

W tej tablicy przechowywane są informacje dotyczące kodów tablic, które nie odpowiadają żadnemu powiatowi, miastu ani dzielnicy.

Kolumna	Opis
name	Nazwa grupy specjalnej tablic.
plateCode	Wyróżniki literowe, które muszą znajdować się na początku tablicy rejestracyjnej, aby przynależać do danej grupy specjalnej.
description	Opis pojazdów, które oznaczone są danym kodem tablic (wozy bojowe, pojazdy Straży Granicznej, itd.)

#### 4.39. WHITELISTVEHICLE

W tej tabeli przechowywane są informacje dotyczące pojazdów znajdujących się na białej liście. Tabela ta znajduje się na osobnym serwerze.

Kolumna	Opis
plate	Tablica rejestracyjna pojazdu.
groupName	Grupa, do której przynależy pojazd.
mark	Marka pojazdu.
model	Model pojazdu.
vehicleColour	Kolor pojazdu.
activationTimeStart	Czas aktywacji potrzebny do określenia, czy wykryty został pojazd z białej listy.
activationTimeEnd	Czas, w którym dezaktywuje się sprawdzanie, czy wykryty został pojazd z białej listy.
country	Kraj, do którego przynależy biała lista.
owner	Właściciel pojazdu znajdującego się na białej liście.

#### 4.40. WHITELISTDETECTION

W tej tabeli przechowywane są informacje dotyczące detekcji pojazdów znajdujących się na białej liście. Tabela ta znajduje się na osobnym serwerze.

Kolumna	Opis
device	Identyfikator urządzenia ANPR.
timestamp	Data wykrycia pojazdu.
lane	Identyfikator pasa ruchu obsługiwanego przez urządzenia ANPR, gdzie wykryty został pojazd.
plate	Tablica rejestracyjna pojazdu.
speed	Prędkość poruszania się pojazdu.
reliability	Wiarygodność w % odczytu tablicy .
country	Identyfikator kraju, do którego przynależy tablica.
photosPath	Katalog, w którym przechowywane są zdjęcia z detekcji pojazdu.
bestplate	Wskazuje indeks obrazu z najlepszą wykorzystaną do odczytu tablicy jakością możliwą dla danego urządzenia.
voivodeship	Województwo, do którego przynależy tablica.
powiat	Powiat, do którego przynależy tablica.
city	Miasto, do którego przynależy tablica.
district	Dzielnica, do której przynależy tablica.

#### 4.41. BLACKLISTDETECTION

W tej tabeli przechowywane są dane historyczne dotyczące pojazdów znajdujących się na czarnej liście.

Kolumna	Opis
device	Identyfikator urządzenia ANPR.
timestamp	Data wykrycia pojazdu.
lane	Identyfikator pasa ruchu obsługiwanego przez urządzenia ANPR, gdzie wykryty został pojazd.
plate	Tablica rejestracyjna pojazdu.
speed	Prędkość poruszania się pojazdu.
reliability	Wiarygodność w % odczytu tablicy .
country	Identyfikator kraju, do którego przynależy tablica.
photosPath	Katalog, w którym przechowywane są zdjęcia z detekcji pojazdu.
bestplate	Wskazuje indeks obrazu z najlepszą wykorzystaną do odczytu tablicy jakością możliwą dla danego urządzenia.
voivodeship	Województwo, do którego przynależy tablica.
powiat	Powiat, do którego przynależy tablica.
city	Miasto, do którego przynależy tablica.
district	Dzielnica, do którego przynależy tablica.
notified	Wskazuje, czy zostało wysłane powiadomienie o detekcji {1 tak, 0 nie}.

#### 4.42. ALARMSTATUS

W tej tabeli przechowywane są informacje o alarmach, które system otrzymuje z urządzeń. System wskazuje, o której godzinie aktywowany został alarm, czy dalej jest aktywny, w jaki sposób i kto został powiadomiony o alarmie.

Kolumna	Opis
device	Identyfikator urządzenia, do którego przynależy alarm.
identifier	Identyfikator alarmu.
specificmessage	Konkretny komunikat alarmu.
timestampactivation	Data aktywacji alarmu.
timestampdeactivation	Data dezaktywacji alarmu.
timestampsonority	Data aktywacji alarmu dźwiękowego.
timestampvalidation	Data sprawdzenia alarmu.
timestampsms	Data wysłania powiadomienia o alarmie w formie wiadomości SMS.
timestampmail	Data wysłania powiadomienia o alarmie w formie wiadomości e-mail.
sound	Wskazuje, czy jest to alarm zaprogramowany jako alarm dźwiękowy {1: dźwiękowy, 0: brak dźwięku}.
Sms	Wskazuje, czy alarm jest zaprogramowany do wysyłania powiadomień w formie wiadomości SMS {1: wysyła SMS, 0: nie wysyła SMSa}.
email	Wskazuje, czy alarm jest zaprogramowany do wysyłania powiadomień w formie wiadomości e-mail {1: wysyła e-mail, 0: nie wysyła e-maila}.
level	Poziom ważności alarmu {-1: nieznany, 0: zerowy, 1: łagodny, 2: średni, 3: poważny, 4: bardzo poważny, 5: krytyczny}.
timestampsentsms	Data wysłania wiadomości SMS.
timestampsentmail	Data wysłania wiadomości e-mail.

#### 4.43. ALARMCONFIGURATION

W tej tabeli przechowywane są informacje o konfiguracji alarmów urządzeń w systemie.

Kolumna	Opis
device	Identyfikator urządzenia, do którego przynależy alarm.
identifier	Identyfikator alarmu.
level	Poziom ważności alarmu {-1: nieznany, 0: zerowy, 1: łagodny, 2: średni, 3: poważny, 4: bardzo poważny, 5: krytyczny}.
timeSound	Czas w sekundach, po którym włącza się dźwięk alarmu.
timeSMS	Czas, po którym wysyłana jest wiadomość SMS.
timeEmail	Czas, po którym wysyłana jest wiadomość e-mail.
enable	Wskazuje, czy jest aktywna czy nie konfiguracja aktualnego alarmu {1: konfiguracja alarmu aktywna, 0: konfiguracja alarmu nieaktywna}.
limits	Limit wartości uznawany za alarm. Należy podać zakres wartości koniecznych do uznania alarmu, np. <10, [10,30], >40.

#### 4.44. ALARMUSERREAD

W tej tabeli definiuje się, którzy użytkownicy mogą widzieć alarmy.

Kolumna	Opis
device	Identyfikator urządzenia, do którego przynależy alarm.
alarm	Identyfikator alarmu.
username	Identyfikator użytkownika.

#### 4.45. ALARMUSERWRITE

W tej tabeli definiuje się, którzy użytkownicy mogą edytować alarmy.

Kolumna	Opis
device	Identyfikator urządzenia, do którego przynależy alarm.
alarm	Identyfikator alarmu.
username	Identyfikator użytkownika.

#### 4.46. ALARMUSERSMS

W tej tabeli przechowywane są informacje o użytkownikach, którzy otrzymają powiadomienie w formie wiadomości SMS po aktywowaniu się konkretnego alarmu.

Kolumna	Opis
device	Identyfikator urządzenia, do którego przynależy alarm.
alarm	Identyfikator alarmu.
username	Identyfikator użytkownika.

#### 4.47. ALARMUSERMAIL

W tej tabeli przechowywane są informacje o użytkownikach, którzy otrzymają powiadomienie w formie wiadomości e-mail po aktywowaniu się konkretnego alarmu.

Kolumna	Opis
device	Identyfikator urządzenia, do którego przynależy alarm.
alarm	Identyfikator alarmu.
username	Identyfikator użytkownika.

#### 4.48. SEQUENCES

W tej tabeli przechowywane są informacje dotyczące poszczególnych sekwencji kamer utworzonych w systemie.

Kolumna	Opis
identifier	Identyfikator sekwencji.
name	Nazwa sekwencji.

#### 4.49. CAMERASEQUENCES

W tej tabeli definiuje się kamery wchodzące w skład każdej sekwencji.

Kolumna	Opis
identifier	Identyfikator sekwencji.
device	Identyfikator kamery, która przynależy do wskazanej sekwencji.
orderNumber	Numer porządkowy służący uporządkowaniu kolejności wyświetlania obrazu z poszczególnych kamer wchodzących w skład sekwencji.
viewtime	Czas w sekundach wyświetlania obrazu z aktualnej kamery we wskazanej sekwencji.

#### 4.50. REPORTFILTERS

W tej tabeli przechowywane są filtry utworzone przez użytkownika, na które składają się zestawy parametrów i wartości, listy z urządzeń ANPR, urządzeń radarowych lub obszarów wybranych przez użytkownika, w celu ich późniejszego użycia przy tworzeniu raportów. Pozwala to użytkownikowi na zapisanie i odzyskanie konfiguracji parametrów zapytań w celu generowania przez system różnego rodzaju raportów dla urządzeń ANPR, radarów lub związanych z czasem przejazdu.

Kolumna	Opis
identifier	Identyfikator zapisanego filtra.
reportName	Identyfikator i nazwa rodzaju raportu.
filterName	Nazwa filtra. Jest to nazwa unikalna dla typu raportu, do którego przynależy.
savePDF	Wskazuje serwerowi, czy raport ma zostać wygenerowany jako plik PDF {1: generuje plik PDF, 0: nie generuje pliku PDF}.
openReport	Wskazuje serwerowi, czy powinien automatycznie otworzyć raport po wygenerowaniu do pliku Excel {: otwiera raport, 0: nie otwiera raportu}. Raport zapisywanie jest automatycznie niezależnie od tych wartości.
creationTime	Data utworzenia raportu.
createdBy	Użytkownik, który zażądał utworzenia raportu.
period	Typ jednostki, w którym generowany jest raport {0: Minuty, 1: Godzina, 2: Dzień, 3: Tydzień, 4: Miesiąc}.
units	Wskazuje wartość okresu. Np. Jednostka-Okres 1- Minuta.

#### 4.51. ANPRLANEREPORTFILTERS

W tej tabeli przechowywane są informacje o pasach ruchu urządzenia ANPR zapisanych w filtrach, które wykorzystywane są do tworzenia raportów.

Kolumna	Opis
reportfilter	Identyfikator filtra.
anpr	Identyfikator urządzenia ANPR.
lane	Identyfikator pasa ruchu.
orderNumber	Indeks listy pasów ruchu urządzenia ANPR { 0: Należy do listy pasów ruchu urządzenia ANPR dla zapytań podstawowych, 1: należy do listy pasów ruchu urządzenia ANPR dla zapytań porównawczych}.

#### 4.52. RADARLANEREPORTFILTERS

W tej tabeli przechowywane są informacje o pasach ruchu urządzenia radarowego zapisanych w filtrach, które wykorzystywane są do tworzenia raportów.

Kolumna	Opis
reportfilter	Identyfikator filtra.
radar	Identyfikator urządzenia radarowego.
lane	Identyfikator pasa ruchu.
orderNumber	Indeks listy pasów ruchu urządzenia radarowego { 0: Należy do listy pasów ruchu urządzenia radarowego dla zapytań podstawowych, 1: należy do listy pasów ruchu urządzenia radarowego dla zapytań porównawczych}.

#### 4.53. VOIVODESHIPSREPORTFILTERS

W tej tabeli przechowywane są informacje dotyczące województw zapisanych w filtrach, które wykorzystywane są do tworzenia raportów.

Kolumna	Opis
reportfilter	Identyfikator filtra.
voivodeship	Identyfikator województwa.
orderNumber	Indeks listy województw { 0: Należy do listy województw dla zapytań podstawowych, 1: należy do listy województw dla zapytań porównawczych}.



#### 4.54. POWIATSREPORTFILTERS

W tej tabeli przechowywane są informacje dotyczące powiatów zapisanych w filtrach, które wykorzystywane są do tworzenia raportów.

Kolumna	Opis
reportfilter	Identyfikator filtra.
powiat	Identyfikator powiatu.
orderNumber	Indeks listy powiatów { 0: Należy do listy powiatów dla zapytań podstawowych, 1: należy do listy powiatów dla zapytań porównawczych}.

#### 4.55. CITIESREPORTFILTERS

W tej tabeli przechowywane są informacje dotyczące miast zapisanych w filtrach, które wykorzystywane są do tworzenia raportów.

Kolumna	Opis
reportfilter	Identyfikator filtra.
city	Identyfikator miasta.
orderNumber	Indeks listy miast { 0: Należy do listy miast dla zapytań podstawowych, 1: należy do listy miast dla zapytań porównawczych}.

#### 4.56. DISTRICTSREPORTFILTERS

W tej tabeli przechowywane są informacje dotyczące dzielnic zapisanych w filtrach, które wykorzystywane są do tworzenia raportów.

Kolumna	Opis
reportfilter	Identyfikator filtra.
district	Identyfikator dzielnicy.
orderNumber	Indeks listy dzielnic { 0: Należy do listy dzielnic dla zapytań podstawowych, 1: należy do listy dzielnic dla zapytań porównawczych}.

#### 4.57. SECTIONREPORTFILTERS

W tej tabeli przechowywane są informacje dotyczące odcinków zapisanych w filtrach, które wykorzystywane są do tworzenia raportów.

Kolumna	Opis
reportfilter	Identyfikator filtra.
sectionId	Identyfikator odcinka.
orderNumber	Indeks listy odcinków { 0: Należy do listy odcinków dla zapytań podstawowych, 1: należy do listy odcinków dla zapytań porównawczych}.

#### 4.58. REPORT

W tej tabeli przechowywane są informacje i linki informujące użytkownika o utworzonych raportach i umożliwiające ich otwarcie

Kolumna	Opis
identifier	Identyfikator wygenerowanego raportu.
reportName	Identyfikator i nazwa rodzaju raportu.
filterName	Identyfikator filtra użytego do wygenerowania aktualnego filtra.
creationTime	Data utworzenia raportu.
createdBy	Użytkownik, który zażądał utworzenia raportu.
link	Link do pobrania raportu, który jest tym samym linkiem, który zawiera kod QR.
path	Ścieżka, pod którą znajduje się raport na serwerze.
period	Okres, w którym grupowane są dane raportu.
tsInit1	Data początkowa dla zapytania o dane wykorzystane do utworzenia raportu.
tsEnd1	Data końcowa dla zapytania o dane wykorzystane do utworzenia raportu.
tsInit2	Data początkowa dla zapytania o dane wykorzystane do porównania.
tsEnd2	Data końcowa dla zapytania o dane wykorzystane do porównania.

#### 4.59. MACRO

W tej tabeli przechowywane są makra stworzone w systemie w celu wysyłania poleceń do urządzeń znajdujących się w sieci. W każdym makro użytkownik zapisuje liczbę żądanych poleceń. Po stworzeniu makra użytkownik może uruchomić serię poleceń i sprawdzić stan ich wysłania w czasie rzeczywistym dowolną ilość razy. Możliwe jest również późniejsze edytowanie makr, dodawania i usuwanie poleceń zapisanych w makro.

Kolumna	Opis
folder	Folder, w którym zapisane jest makro.
name	Identyfikator i nazwa makra.
orderNumber	Identyfikator polecenia zapisanego w makro.
objectData	Informacja, której potrzebuje, aby mogło być wykonane.
description	Szczegółowy opis czynności wykonanych przez polecenie.

#### 4.60. CONTACT

W tej tabeli przechowywane są informacje o kontaktach, których system używa do konfiguracji powiadomień o wykryciach tablic pojazdów znajdujących się na czarnej liście. Każdy użytkownik systemu posiada swoją własną grupę kontaktów zabezpieczoną poprzez identyfikację nazwą użytkownika i hasłem.

Kolumna	Opis
name	Indeks i nazwa kontaktu.
description	Szczegółowy opis kontaktu.
email	E-mail, na który wysłane zostanie powiadomienie o detekcji pojazdu z czarnej listy.
phone	Numer telefonu, na który wysłane zostanie powiadomienie w formie wiadomości sms o detekcji pojazdu z czarnej listy.
username	Indeks i nazwa użytkownika - właściciela kontaktu.

#### 4.61. VEHICLECHRONOS

W tej tabeli przechowywane są informacje o pojazdach w celu późniejszej konfiguracji czarnej listy. Tablica rejestracyjna może być tablicą kompletną lub zakresem wyszukiwania znaków tablicy, np. WE\*, które powiadomi o wszystkich odnalezionych tablicach dla danego kalendarza i urządzenia ANPR, które zaczynają się na WE.

Kolumna	Opis
plate	Tablica lub wzór wyszukiwania.
groupName	Grupa, do której przynależy pojazd.
mark	Marka pojazdu.
model	Model pojazdu.
vehicleColour	Kolor pojazdu.
activationTimeStart	Data, od której wyszukiwanie jest aktywne dla danego pojazdu lub grupy pojazdów.
activationTimeEnd	Data wygaśnięcia wyszukiwania danego pojazdu lub grupy pojazdów.
country	Kraj, do którego przynależy pojazd.
owner	Właściciel pojazdu.
blacklist	Powiadomienia o detekcjach danego pojazdu {1: Powiadomienie w formie wiadomości SMS, e-mail i na ekranie, 0: powiadomienie w formie wiadomości SMS, e-mail i bez powiadomienia na ekranie}
username	Użytkownik, który utworzył pojazd w systemie. Użytkownik może widzieć jedynie te pojazdy, które sam stworzył, ponieważ są one zabezpieczone poprzez identyfikację nazwą użytkownika i hasłem.

#### 4.62. SCHEDULE

W tej tabeli przechowywane są informacje o harmonogramach utworzonych przez użytkownika, które tylko on może widzieć (są zabezpieczone poprzez identyfikację nazwy użytkownika i hasła). Tworzenie harmonogramu pozwala na wskazanie pory dnia (w minutach), kiedy aktywne jest wyszukiwanie detekcji pojazdów znajdujących się na czarnej liście. System prosi nas o wybranie koloru pierwszoplanowego (foreground) i koloru tła (background), w celu wykorzystania tych kolorów do późniejszego wyboru harmonogramu z kalendarza dla wybranego dnia. Foreground to kolor tekstu dnia, natomiast background to kolor tła wybranego dnia.

Kolumna	Opis
name	Identyfikator i nazwa utworzonego harmonogramu.
foreground	Identyfikator koloru wybranego przez użytkownika jako foreground harmonogramu.
background	Identyfikator koloru wybranego przez użytkownika jako background harmonogramu.
values	Wartość binarna odpowiadająca minutom dnia, kiedy aktywne jest wyszukiwanie.
username	Użytkownik, który utworzył harmonogram w systemie. Użytkownik może widzieć jedynie te harmonogramy, które sam stworzył, ponieważ są one zabezpieczone poprzez identyfikację nazwą użytkownika i hasłem.

#### 4.63. CALENDAR

W tej tabeli przechowywane są relacje dotyczące kalendarzy przypisanych do użytkowników.

Kolumna	Opis
name	Identyfikator i nazwa kalendarza.
username	Użytkownik, który utworzył kalendarz w systemie. Użytkownik może widzieć jedynie te kalendarze, które sam stworzył, ponieważ są one zabezpieczone poprzez identyfikację nazwą użytkownika i hasłem.

#### 4.64. CALENDARDATA

W tej tabeli przechowywane są informacje dotyczące dni kalendarza. Dla każdego dnia możliwy jest wybór innego harmonogramu z tych stworzonych przez użytkownika w systemie.

Kolumna	Opis
calendar	Identyfikator i nazwa kalendarza.
timestamp	Data odpowiadająca dniu kalendarza.
schedule	Identyfikator harmonogramu wybranego dla danego dnia i kalendarza.
username	Użytkownik, który utworzył kalendarz w systemie. Użytkownik może widzieć jedynie te kalendarze, które sam stworzył, ponieważ są one zabezpieczone poprzez identyfikację nazwą użytkownika i hasłem.

#### 4.65. ANPRCALENDARVEHICLEUSER

W tej tabeli definiowana jest czarna lista oraz wskazuje się związek pomiędzy urządzeniami ANPR, kalendarzem i pojazdami stworzonymi przez użytkownika. Dla każdego urządzenia ANPR wskazuje się pojazdy, które mają być wyszukiwane w ramach jego detekcji, w określonym okresie (daty i godziny) wskazanym w kalendarzu, aby możliwa była aktywacja powiadomień o wykrytych tablicach.

Kolumna	Opis
anpr	Identyfikator urządzenia ANPR.
calendar	Identyfikator kalendarza.
plate	Tablica lub wzór wyszukiwania.
username	Użytkownik, który utworzył czarną listę, w celu detekcji i powiadomienia o wykryciu znajdujących się na niej pojazdów. Użytkownik może widzieć jedynie te konfiguracje, które sam stworzył, ponieważ są one zabezpieczone poprzez identyfikację nazwą użytkownika i hasłem.
groupName	Grupa, do której przynależy pojazd.
blacklist	Powiadomienia o detekcjach danego pojazdu {1: Powiadomienie w formie wiadomości SMS, e-mail i na ekranie, 0: powiadomienie w formie wiadomości SMS, e-mail i bez powiadomienia na ekranie}

#### 4.66. CONTACTANPRCALENDARVEHICLEUSER

W tej tabeli definiuje się kontakty i sposób, w jaki mają one być powiadamiane o wykryciu pojazdu znajdującego się na czarnej liście.

Kolumna	Opis
contact	Identyfikator kontaktu, który ma otrzymać powiadomienie.
anpr	Identyfikator urządzenia ANPR.
calendar	Identyfikator kalendarza.
plate	Tablica lub wzór wyszukiwania.
username	Użytkownik, który zdefiniował kontakty, które otrzymywać będą powiadomienia. Użytkownik może widzieć jedynie te konfiguracje powiadomień, które sam stworzył, ponieważ są one zabezpieczone poprzez identyfikację nazwą użytkownika i hasłem.

#### 4.67. EXTERNALLAYERS

W tej tabeli tworzone są nowe warstwy zewnętrzne WMS. Znajdują się tu informacje, które będą załadowane na żądanie użytkownika na mapie w systemie.

Kolumna	Opis
name	Identyfikator i nazwa warstwy zewnętrznej.
info	Dane binarne warstwy zewnętrznej, gdzie definiowane są: wykorzystywana wersja WMS, style i URL połączenia.

#### 4.68. EVENT

W tej tabeli przechowywane są dane historyczne dotyczące możliwych zdarzeń, które mogą wystąpić w systemie. Na przykład, wysłanie przez użytkownika polecenia wyłączenia konkretnego urządzenia, modyfikacji okresu pobierania danych z kamery, zmiany trybu interwału obliczania czasu przejazdu trasy lub odcinka, modyfikacji jakiegokolwiek informacji (IP, port, itd.) dotyczącej urządzenia, routera lub lokalizacji, utworzenia użytkownika, usunięcia użytkownika, modyfikacji informacji o użytkowniku, zmiana trybu pracy systemu SCP, itd.

Kolumna	Opis
timestamp	Data zdarzenia.
moduleid	Identyfikator modułu, w którym wystąpiło zdarzenie.
typeid	Typ zdarzenia.
subtypeid	Podtyp zdarzenia.
username	Użytkownik, który spowodował zdarzenie.
description	Szczegółowy opis zdarzenia.

## 5. SQL

```

CREATE TABLE DEVICE
(
    identifier          VARCHAR(256) NOT NULL,
    location            VARCHAR(256) NOT NULL,
    router              VARCHAR(256),
    deviceType          SMALLINT,
    latitude            FLOAT,
    longitude           FLOAT,
    road                VARCHAR(1024),
    kilometricPoint     VARCHAR(1024),
    name                VARCHAR(1024),
    description         VARCHAR(1024),
    ip                  VARCHAR(16),
    dataPort            INTEGER,
    enabled             SMALLINT,
    period              INTEGER,
    state               SMALLINT,
    lastUpdate          TIMESTAMP(6),
    CONSTRAINT pk_device PRIMARY KEY (identifier)
);

CREATE TABLE LOCATION
(
    identifier          VARCHAR(256) NOT NULL,
    CONSTRAINT pk_location PRIMARY KEY (identifier),
    CONSTRAINT fk_device_location FOREIGN KEY (identifier) REFERENCES DEVICE
(identifier) ON DELETE CASCADE
);

CREATE TABLE ROUTER
(
    identifier          VARCHAR(256) NOT NULL,
    gsmSignal           INTEGER,
    gsmMode             SMALLINT,
    powerMode           SMALLINT,
    freeMemory          INTEGER,
    lastUpdate          TIMESTAMP(6),
    CONSTRAINT pk_router PRIMARY KEY (identifier),
    CONSTRAINT fk_device_router FOREIGN KEY (identifier) REFERENCES DEVICE
(identifier) ON DELETE CASCADE
);

CREATE TABLE METEO
(
    device              VARCHAR(256) NOT NULL,
    period              SMALLINT,
    CONSTRAINT pk_meteo PRIMARY KEY (device),
    CONSTRAINT fk_device_meteo FOREIGN KEY (device) REFERENCES DEVICE
(identifier) ON DELETE CASCADE
);

CREATE TABLE METEOSENSOR
(

```

```

device          VARCHAR(256) NOT NULL,
sensor          SMALLINT,
description     VARCHAR(1024),
units          VARCHAR(256),
CONSTRAINT pk_meteosensor PRIMARY KEY (device, sensor),
CONSTRAINT fk_meteo_sensor FOREIGN KEY (device) REFERENCES METEO (device)
ON DELETE CASCADE
);

CREATE TABLE METEOSENSORDATA
(
  device          VARCHAR(256) NOT NULL,
  sensor          SMALLINT,
  timestamp       TIMESTAMP(6) NOT NULL,
  value           INTEGER, /* *100 */
  period         SMALLINT,
  CONSTRAINT pk_meteosensordata PRIMARY KEY (device, sensor, timestamp),
  CONSTRAINT fk_meteosensor_data FOREIGN KEY (device, sensor) REFERENCES
METEOSENSOR (device, sensor) ON DELETE CASCADE
);

CREATE TABLE RADAR
(
  device          VARCHAR(256) NOT NULL,
  period         SMALLINT,
  lengths        SMALLINT,
  speeds         SMALLINT,
  CONSTRAINT pk_radar PRIMARY KEY (device),
  CONSTRAINT fk_device_radar FOREIGN KEY (device) REFERENCES DEVICE
(identifier) ON DELETE CASCADE
);

CREATE TABLE RADARLANE
(
  device          VARCHAR(256) NOT NULL,
  lane           SMALLINT,
  description     VARCHAR(1024),
  CONSTRAINT pk_radarlane PRIMARY KEY (device, lane),
  CONSTRAINT fk_radar_lane FOREIGN KEY (device) REFERENCES RADAR (device) ON
DELETE CASCADE
);

CREATE TABLE RADARLANEDATA
(
  device          VARCHAR(256) NOT NULL,
  lane           SMALLINT,
  timestamp       TIMESTAMP(6) NOT NULL,
  speed          SMALLINT,
  occupancy      SMALLINT,
  vehicles       INTEGER,
  notClassified  INTEGER,
  period         SMALLINT,
  CONSTRAINT pk_radarlanedata PRIMARY KEY (device, lane, timestamp),
  CONSTRAINT fk_radarlane_data FOREIGN KEY (device, lane) REFERENCES
RADARLANE (device, lane) ON DELETE CASCADE
);

CREATE TABLE LENGTHCLASSIFICATION
(
  device          VARCHAR(256) NOT NULL,

```



```

        identifier          SMALLINT,
        description         VARCHAR(256),
        lowerLimit          SMALLINT,
        upperLimit          SMALLINT,
        CONSTRAINT pk_lengthclas PRIMARY KEY (device, identifier),
        CONSTRAINT fk_device_lengthclas FOREIGN KEY (device) REFERENCES DEVICE
(identifier) ON DELETE CASCADE
    );

CREATE TABLE RADARLANEDATALength
(
    device          VARCHAR(256) NOT NULL,
    lane            SMALLINT,
    timestamp       TIMESTAMP(6) NOT NULL,
    classification  SMALLINT,
    vehicles        SMALLINT,
    CONSTRAINT pk_radarlanedatalength PRIMARY KEY (device, lane, timestamp,
classification),
    CONSTRAINT fk_radarlanedata_length FOREIGN KEY (device, lane, timestamp)
REFERENCES RADARLANEDATA (device, lane, timestamp) ON DELETE CASCADE,
    CONSTRAINT fk_lengthclas_length FOREIGN KEY (device, classification)
REFERENCES LENGTHCLASSIFICATION (device, identifier) ON DELETE CASCADE
);

CREATE TABLE ANPR
(
    device          VARCHAR(256) NOT NULL,
    lanes           SMALLINT,
    notifyBlackList SMALLINT,
    notifyDetections SMALLINT,
    notifyTime      SMALLINT,
    blacklistAlarmTime SMALLINT,
    nonAuthorizedAlarmTime SMALLINT,
    CONSTRAINT pk_anpr PRIMARY KEY (device),
    CONSTRAINT fk_device_anpr FOREIGN KEY (device) REFERENCES DEVICE
(identifier) ON DELETE CASCADE
);

CREATE TABLE ANPRLANE
(
    device          VARCHAR(256) NOT NULL,
    lane            INTEGER,
    description     VARCHAR(1024),
    CONSTRAINT pk_anprlane PRIMARY KEY (device, lane),
    CONSTRAINT fk_anpr_lane FOREIGN KEY (device) REFERENCES ANPR (device) ON
DELETE CASCADE
);

CREATE TABLE PLATEDETECTION
(
    device          VARCHAR(256) NOT NULL,
    timestamp       TIMESTAMP(6) NOT NULL,
    lane            INTEGER,
    plate           VARCHAR(256),
    speed           SMALLINT,
    reliability      FLOAT,
    country         SMALLINT,
    photosPath      VARCHAR(1024),
    blacklist       SMALLINT,
    whitelist       SMALLINT,

```

```

downloaded      SMALLINT,
bestplate       SMALLINT,
voivodeship     VARCHAR(1024) NOT NULL,
powiat          VARCHAR(1024) NOT NULL,
city            VARCHAR(1024),
district        VARCHAR(1024),
CONSTRAINT pk_anprdetetection PRIMARY KEY (device, timestamp, lane),
CONSTRAINT fk_anprlane_detection FOREIGN KEY (device, lane) REFERENCES
anprlane (device, lane) ON UPDATE NO ACTION ON DELETE CASCADE
)
    
```

```

CREATE TABLE ROUTE
(
    identifier          VARCHAR(256) NOT NULL,
    creationLogin       VARCHAR(256) NOT NULL,
    creationTime        TIMESTAMP(6) NOT NULL,
    direction           VARCHAR(256),
    description         VARCHAR(1024),
    fixedInterval       SMALLINT,
    length              INTEGER,
    modelTime           INTEGER,
    clientVisible       SMALLINT,
    webVisible          SMALLINT,
    mobileVisible       SMALLINT,
    workingMode         SMALLINT,
    CONSTRAINT pk_route PRIMARY KEY (identifier)
);
    
```

```

CREATE TABLE SECTION
(
    identifier          VARCHAR(256) NOT NULL,
    direction           VARCHAR(256),
    description         VARCHAR(1024),
    fixedInterval       INTEGER,
    length              INTEGER,
    modelTime           INTEGER,
    minIgnoreTime       INTEGER,
    maxIgnoreTime       INTEGER,
    maxSpeed            SMALLINT,
    workingMode         SMALLINT,
    matchings           SMALLINT,
    CONSTRAINT pk_section PRIMARY KEY (identifier)
);
    
```

```

CREATE TABLE ANPRLANESECTION
(
    section            VARCHAR(256) NOT NULL,
    anpr               VARCHAR(256) NOT NULL,
    lane               INTEGER,
    type               SMALLINT,
    CONSTRAINT pk_anprlanesection PRIMARY KEY (section, anpr, lane),
    CONSTRAINT fk_anprlane_section FOREIGN KEY (anpr, lane) REFERENCES ANPRLANE
(device, lane) ON DELETE CASCADE,
    CONSTRAINT fk_section_section FOREIGN KEY (section) REFERENCES SECTION
(identifier) ON DELETE CASCADE
);
    
```

```

CREATE TABLE SECTIONROUTE
(
    route          VARCHAR(256) NOT NULL,
    section        VARCHAR(256) NOT NULL,
    orderNumber    SMALLINT,
    CONSTRAINT pk_sectionroute PRIMARY KEY (route, section),
    CONSTRAINT fk_route_secroute FOREIGN KEY (route) REFERENCES ROUTE
(identifier) ON DELETE CASCADE,
    CONSTRAINT fk_section_secroute FOREIGN KEY (section) REFERENCES SECTION
(identifier) ON DELETE CASCADE
);

CREATE TABLE ROUTEINTERVAL
(
    route          VARCHAR(256) NOT NULL,
    startTime      TIMESTAMP(6) NOT NULL,
    endTime        TIMESTAMP(6) NOT NULL,
    interval        SMALLINT,
    CONSTRAINT pk_routeInterval PRIMARY KEY (route, startTime, endTime),
    CONSTRAINT fk_route_rinterval FOREIGN KEY (route) REFERENCES ROUTE
(identifier) ON DELETE CASCADE
);

CREATE TABLE SECTIONINTERVAL
(
    section        VARCHAR(256) NOT NULL,
    startTime      TIMESTAMP(6) NOT NULL,
    endTime        TIMESTAMP(6) NOT NULL,
    interval        SMALLINT,
    CONSTRAINT pk_sectionInterval PRIMARY KEY (section, startTime, endTime),
    CONSTRAINT fk_section_sinterval FOREIGN KEY (section) REFERENCES SECTION
(identifier) ON DELETE CASCADE
);

CREATE TABLE ROUTEDATA
(
    route          VARCHAR(256) NOT NULL,
    timestamp      TIMESTAMP(6) NOT NULL,
    time           INTEGER,
    trafficState    SMALLINT,
    CONSTRAINT pk_routedata PRIMARY KEY (route, timestamp),
    CONSTRAINT fk_route_data FOREIGN KEY (route) REFERENCES ROUTE (identifier)
ON DELETE CASCADE
);

CREATE TABLE SECTIONDATA
(
    section        VARCHAR(256) NOT NULL,
    timestamp      TIMESTAMP(6) NOT NULL,
    time           INTEGER,
    trafficState    SMALLINT,
    matchings      SMALLINT,
    CONSTRAINT pk_sectiondata PRIMARY KEY (section, timestamp),
    CONSTRAINT fk_section_data FOREIGN KEY (section) REFERENCES SECTION
(identifier) ON DELETE CASCADE
    
```

```
);
```

```

CREATE TABLE MATCHING (
    section VARCHAR(256) NOT NULL,
    anpr1 VARCHAR(256) NOT NULL,
    lane1 INTEGER,
    anpr2 VARCHAR(256) NOT NULL,
    lane2 INTEGER,
    originTime TIMESTAMP(6) NOT NULL,
    destinationTime TIMESTAMP(6) NOT NULL,
    time INTEGER,
    length INTEGER,
    speed SMALLINT,
    CONSTRAINT pk_sectionmatching PRIMARY KEY (section , anpr1 , lane1 ,
anpr2 , lane2 , originTime , destinationTime),
    CONSTRAINT fk_section_matching FOREIGN KEY (section)
        REFERENCES SECTION (identifier)
        ON DELETE CASCADE,
    CONSTRAINT fk_anprlane_matching1 FOREIGN KEY (anpr1 , lane1)
        REFERENCES ANPRLANE (device , lane)
        ON DELETE CASCADE,
    CONSTRAINT fk_anprlane_matching2 FOREIGN KEY (anpr2 , lane2)
        REFERENCES ANPRLANE (device , lane)
        ON DELETE CASCADE
);
    
```

```
);
```

```

CREATE TABLE SYSTEMUSER
(
    username      VARCHAR(256) NOT NULL,
    password      VARCHAR(32)  NOT NULL,
    name          VARCHAR(256) NOT NULL,
    surname       VARCHAR(256) NOT NULL,
    email         VARCHAR(256) NOT NULL,
    phone         VARCHAR(256) NOT NULL,
    company       VARCHAR(1024) NOT NULL,
    jobtitle      VARCHAR(1024) NOT NULL,
    address       VARCHAR(1024) NOT NULL,
    profile       SMALLINT,
    enabled       SMALLINT,
    creationTime  TIMESTAMP(6) ,
    deleteTime   TIMESTAMP(6) ,
    lastUpdate    TIMESTAMP(6) ,
    CONSTRAINT pk_user PRIMARY KEY (username)
);
    
```

```

CREATE TABLE SYSTEMUSERSTATUS
(
    timestamp     TIMESTAMP(6) NOT NULL,
    username      VARCHAR(256) NOT NULL,
    password      VARCHAR(32)  NOT NULL,
    name          VARCHAR(256) NOT NULL,
    surname       VARCHAR(256) NOT NULL,
    email         VARCHAR(256) NOT NULL,
    phone         VARCHAR(256) NOT NULL,
    company       VARCHAR(1024) NOT NULL,
    jobtitle      VARCHAR(1024) NOT NULL,
    address       VARCHAR(1024) NOT NULL,
    profile       SMALLINT,
    
```

```

    enabled                SMALLINT,
    creationTime            TIMESTAMP(6) NOT NULL,
    deleteTime             TIMESTAMP(6) NOT NULL,
    lastUpdate              TIMESTAMP(6) NOT NULL,
    CONSTRAINT pk_userstatus PRIMARY KEY (username)
);

CREATE TABLE SYSTEMUSERACCESS
(
    username                VARCHAR(256) NOT NULL,
    startTime               TIMESTAMP(6) NOT NULL,
    console                 VARCHAR(256) NOT NULL,
    endTime                 TIMESTAMP(6) NOT NULL,
    CONSTRAINT pk_useraccess PRIMARY KEY (username, startTime, console),
    CONSTRAINT fk_user FOREIGN KEY (username) REFERENCES SYSTEMUSER (username)
ON DELETE CASCADE
);

CREATE TABLE CAMERA
(
    device                  VARCHAR(256) NOT NULL,
    downloadPeriod          INTEGER,
    downloadMode            SMALLINT,
    jpegCompression         SMALLINT,
    jpegWidth               SMALLINT,
    jpegHeight              SMALLINT,
    jpegWebCompression      SMALLINT,
    jpegWebWidth            SMALLINT,
    jpegWebHeight           SMALLINT,
    jpegMobileCompression   SMALLINT,
    jpegMobileWidth         SMALLINT,
    jpegMobileHeight        SMALLINT,
    motion                  SMALLINT,
    motionProtocol          SMALLINT,
    lastPhotoPath           VARCHAR(1024),
    latPhotoTime            TIMESTAMP(6),
    CONSTRAINT pk_camera PRIMARY KEY (device),
    CONSTRAINT fk_device_camera FOREIGN KEY (device) REFERENCES DEVICE
(identifier) ON DELETE CASCADE
);

CREATE TABLE IMAGELOG
(
    device                  VARCHAR(256) NOT NULL,
    timestamp               TIMESTAMP(6) NOT NULL,
    imagePath               VARCHAR(1024),
    deleted                 SMALLINT DEFAULT 0,
    CONSTRAINT pk_imagelog PRIMARY KEY (device, timestamp),
    CONSTRAINT fk_camera_imagelog FOREIGN KEY (device) REFERENCES CAMERA
(device) ON DELETE CASCADE
);

CREATE TABLE VMS
(
    device                  VARCHAR(256) NOT NULL,
    width                   SMALLINT,
    height                  SMALLINT,
    workingMode             SMALLINT NOT NULL,
    CONSTRAINT pk_vms PRIMARY KEY (device),

```

```

    CONSTRAINT fk_device_vms FOREIGN KEY (device) REFERENCES DEVICE
(identifier) ON DELETE CASCADE
);

CREATE TABLE VMSEGRAPHICFOLDER
(
  resolution    VARCHAR(1024) NOT NULL,
  folder        VARCHAR(1024) NOT NULL,
  CONSTRAINT pk_vmsgraphicfolder PRIMARY KEY (resolution, folder)
);

CREATE TABLE VMSEGRAPHIC
(
  identifier      INTEGER NOT NULL,
  resolution      VARCHAR(1024) NOT NULL,
  graphicFolder   VARCHAR(1024) NOT NULL,
  graphicName     VARCHAR(1024) NOT NULL,
  address         VARCHAR(256),
  width          SMALLINT,
  height         SMALLINT,
  image          BYTEA,
  priority        SMALLINT DEFAULT 5,
  CONSTRAINT pk_vmsgraphic PRIMARY KEY (resolution, graphicFolder,
graphicname),
  CONSTRAINT pk_vmsgraphic_folder FOREIGN KEY (resolution, graphicFolder)
REFERENCES VMSEGRAPHICFOLDER(resolution, folder) ON DELETE CASCADE ON UPDATE
CASCADE
);

CREATE TABLE VMSEMACRO
(
  identifier      VARCHAR(1024) NOT NULL,
  resolution      VARCHAR(1024) NOT NULL,
  graphicFolder   VARCHAR(1024) NOT NULL,
  graphicName     VARCHAR(1024) NOT NULL,
  macrotype       VARCHAR(1024) NOT NULL,
  substitution    VARCHAR(1024) NOT NULL,
  CONSTRAINT pk_vmsmacro PRIMARY KEY (identifier, resolution, graphicFolder,
graphicname),
  CONSTRAINT pk_vmsmacro_graphic FOREIGN KEY (resolution, graphicFolder,
graphicname) REFERENCES VMSEGRAPHIC(resolution, graphicFolder, graphicname) ON
DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE VMSEMACROMTO
(
  identifier      VARCHAR(1024) NOT NULL,
  resolution      VARCHAR(1024) NOT NULL,
  graphicFolder   VARCHAR(1024) NOT NULL,
  graphicName     VARCHAR(1024) NOT NULL,
  sensor         SMALLINT NOT NULL,
  oper           SMALLINT,
  value          FLOAT,
  active         SMALLINT,
  CONSTRAINT pk_vmsmacromto PRIMARY KEY (identifier, resolution,
graphicFolder, graphicname, sensor),
  CONSTRAINT pk_vmsmacromto_macro FOREIGN KEY (identifier, resolution,
graphicFolder, graphicname) REFERENCES VMSEMACRO(identifier, resolution,
graphicFolder, graphicname) ON DELETE CASCADE ON UPDATE CASCADE
);

```

**CREATE TABLE** MODULO

```
(
  identifier      VARCHAR(256) NOT NULL,
  description     VARCHAR(1024) NOT NULL,
  enabled        SMALLINT NOT NULL,
  CONSTRAINT pk_module PRIMARY KEY (identifier)
);
```

**CREATE TABLE** MODULEFEATURE

```
(
  moduleId       VARCHAR(256) NOT NULL,
  feature        VARCHAR(256) NOT NULL,
  description     VARCHAR(1024),
  CONSTRAINT pk_modulefeature PRIMARY KEY (moduleId, feature),
  CONSTRAINT fk_module_modulefeature FOREIGN KEY (moduleId) REFERENCES MODULO
(identifier) ON DELETE CASCADE ON UPDATE CASCADE
);
```

**CREATE TABLE** MODULEFEATUREUSERRIGHT

```
(
  moduleId       VARCHAR(256) NOT NULL,
  feature        VARCHAR(256) NOT NULL,
  username       VARCHAR(256) NOT NULL,
  startTime      TIMESTAMP(6),
  endTime       TIMESTAMP(6),
  enabled        SMALLINT,
  CONSTRAINT pk_modfeatureuserright PRIMARY KEY (moduleId, feature,
username),
  CONSTRAINT fk_modfeature_modfeausright FOREIGN KEY (moduleId, feature)
REFERENCES MODULEFEATURE (moduleId, feature) ON DELETE CASCADE,
  CONSTRAINT fk_user_modfeausright FOREIGN KEY (username) REFERENCES
SYSTEMUSER (username) ON DELETE CASCADE
);
```

**CREATE TABLE** VEHICLE

```
(
  plate          VARCHAR(256) NOT NULL,
  organism       INTEGER,
  mark           VARCHAR(256),
  model          VARCHAR(256),
  colour         VARCHAR(256),
  country        VARCHAR(256),
  owner          VARCHAR(256),
  CONSTRAINT pk_vehicle PRIMARY KEY (plate)
);
```

**CREATE TABLE** VOIVODESHIP

```
(
  name           VARCHAR(1024),
  description     VARCHAR(1024),
  CONSTRAINT pk_voivodeship PRIMARY KEY (name)
);
```

**CREATE TABLE** POWIAT

```
(
  name           VARCHAR(1024),
  voivodeship    VARCHAR(1024),
  plateCode      VARCHAR(32),
```

```

description          VARCHAR(1024),
CONSTRAINT pk_powiat PRIMARY KEY (name, platecode),
CONSTRAINT fk_voivodeship_powiat FOREIGN KEY (voivodeship) REFERENCES
VOIVODESHIP (name) ON DELETE CASCADE
);

CREATE TABLE CITY
(
  name                VARCHAR(1024),
  powiat              VARCHAR(1024),
  plateCode           VARCHAR(32),
  description          VARCHAR(1024),
  CONSTRAINT pk_city PRIMARY KEY (name, platecode),
  CONSTRAINT fk_powiat_city FOREIGN KEY (powiat, platecode) REFERENCES POWIAT
(name, platecode) ON DELETE CASCADE
);

CREATE TABLE DISTRICT
(
  name                VARCHAR(1024),
  city                VARCHAR(1024),
  startPlateCode       VARCHAR(32),
  endPlateCode         VARCHAR(32),
  description          VARCHAR(1024),
  CONSTRAINT pk_district PRIMARY KEY (name, startplatecode, endplatecode),
  CONSTRAINT fk_city_district FOREIGN KEY (city, startplatecode) REFERENCES
CITY (name, platecode) ON DELETE CASCADE
);

CREATE TABLE SPECIALPOLISHPLATE
(
  name                VARCHAR(1024),
  plateCode           VARCHAR(32),
  description          VARCHAR(1024),
  CONSTRAINT pk_special PRIMARY KEY (name)
);

CREATE TABLE WHITELISTVEHICLE
(
  plate                VARCHAR(256) NOT NULL,
  groupName            VARCHAR(256) NOT NULL,
  mark                 VARCHAR(256),
  model                VARCHAR(256),
  vehicleColour        VARCHAR(256),
  activationTimeStart  TIMESTAMP(6),
  activationTimeEnd    TIMESTAMP(6),
  country              VARCHAR(256),
  owner                VARCHAR(256),
  CONSTRAINT pk_whitevehicle PRIMARY KEY (plate)
);

CREATE TABLE WHITELISTDETECTION
(
  device               VARCHAR(256) NOT NULL,
  timestamp            TIMESTAMP(6) NOT NULL,
  lane                 INTEGER,
  plate                VARCHAR(256),
  speed                SMALLINT,
  reliability           FLOAT,
  country              SMALLINT,

```



```

photosPath      VARCHAR(1024),
blacklist        SMALLINT,
whitelist        SMALLINT,
downloaded       SMALLINT,
bestplate        SMALLINT,
voivodeship      VARCHAR(1024) NOT NULL,
powiat           VARCHAR(1024) NOT NULL,
city             VARCHAR(1024),
district         VARCHAR(1024),
CONSTRAINT pk_whitelistdetection PRIMARY KEY (device, timestamp, lane),
CONSTRAINT fk_anprlane_white FOREIGN KEY (device, lane) REFERENCES ANPRLANE
(device, lane) ON DELETE CASCADE,
CONSTRAINT fk_country_white FOREIGN KEY (country) REFERENCES COUNTRY
(country) ON DELETE CASCADE
);
    
```

**CREATE TABLE** BLACKLISTDETECTION

```

(
    device        VARCHAR(256) NOT NULL,
    timestamp      TIMESTAMP(6) NOT NULL,
    lane           INTEGER,
    plate          VARCHAR(256),
    speed          SMALLINT,
    reliability     FLOAT,
    country        SMALLINT,
    photosPath     VARCHAR(1024),
    blacklist       SMALLINT,
    whitelist       SMALLINT,
    downloaded      SMALLINT,
    bestplate       SMALLINT,
    voivodeship     VARCHAR(1024) NOT NULL,
    powiat          VARCHAR(1024) NOT NULL,
    city           VARCHAR(1024),
    district        VARCHAR(1024),
    notified        SMALLINT DEFAULT 0,
    CONSTRAINT pk_blacklistdetection PRIMARY KEY (device, timestamp, lane),
    CONSTRAINT fk_anprlane_black FOREIGN KEY (device, lane) REFERENCES ANPRLANE
(device, lane) ON DELETE CASCADE,
    CONSTRAINT fk_country_black FOREIGN KEY (country) REFERENCES COUNTRY
(country) ON DELETE CASCADE
);
    
```

**CREATE TABLE** ALARMSTATUS

```

(
    device character varying(256) NOT NULL,
    identifier character varying(256) NOT NULL,
    specificmessage character varying(1024) NOT NULL,
    timestampactivation timestamp(6) NOT NULL,
    timestampdeactivation timestamp(6) ,
    timestampsonority timestamp(6) ,
    timestampvalidation timestamp(6) ,
    timestampsms timestamp(6) ,
    timestampmail timestamp(6) ,
    sound smallint,
    sms smallint,
    email smallint,
    timestampsentsms timestamp(6) ,
    timestampsentmail timestamp(6) ,
    level integer,
    )
    
```

```
CONSTRAINT pk_alarmstatus PRIMARY KEY (device, identifier,  
timestampactivation),  
CONSTRAINT fk_device_alarmstatus FOREIGN KEY (device) REFERENCES device  
(identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE ALARMCONFIGURATION  
(  
device character varying(256) NOT NULL,  
identifier character varying(256) NOT NULL,  
level integer,  
timeSound smallint,  
timeSMS smallint,  
timeEmail smallint,  
enable smallint,  
limits character varying(1024) NOT NULL,  
CONSTRAINT pk_alarmconfig PRIMARY KEY (device, identifier),  
CONSTRAINT fk_device_alarmconfig FOREIGN KEY (device) REFERENCES device  
(identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE ALARMUSERREAD  
(  
device character varying(256) NOT NULL,  
alarm character varying(256) NOT NULL,  
username character varying(256) NOT NULL,  
CONSTRAINT pk_alarmuserread PRIMARY KEY (device, alarm, username),  
CONSTRAINT fk_alarmcfg_alarmuserread FOREIGN KEY (device, alarm) REFERENCES  
ALARMCONFIGURATION (device, identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE ALARMUSERWRITE  
(  
device character varying(256) NOT NULL,  
alarm character varying(256) NOT NULL,  
username character varying(256) NOT NULL,  
CONSTRAINT pk_alarmuserwrite PRIMARY KEY (device, alarm, username),  
CONSTRAINT fk_alarmcfg_alarmuserwrite FOREIGN KEY (device, alarm)  
REFERENCES ALARMCONFIGURATION (device, identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE ALARMUSERSMS  
(  
device character varying(256) NOT NULL,  
alarm character varying(256) NOT NULL,  
username character varying(256) NOT NULL,  
CONSTRAINT pk_alarmusersms PRIMARY KEY (device, alarm, username),  
CONSTRAINT fk_alarmcfg_alarmusersms FOREIGN KEY (device, alarm) REFERENCES  
ALARMCONFIGURATION (device, identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE ALARMUSERMAIL  
(  
device character varying(256) NOT NULL,  
alarm character varying(256) NOT NULL,
```

```
username character varying(256) NOT NULL,
CONSTRAINT pk_alarmusermail PRIMARY KEY (device, alarm, username),
CONSTRAINT fk_alarmcfg_alarmusermail FOREIGN KEY (device, alarm) REFERENCES
ALARMCONFIGURATION (device, identifier) ON DELETE CASCADE
);
```

**CREATE TABLE SEQUENCES**

```
(
    identifier          VARCHAR(1024) NOT NULL,
    name                VARCHAR(1024) NOT NULL,
    CONSTRAINT pk_sequences PRIMARY KEY (identifier)
);
```

**CREATE TABLE CAMERASEQUENCES**

```
(
    identifier          VARCHAR(1024) NOT NULL,
    device              VARCHAR(256) NOT NULL,
    orderNumber         SMALLINT,
    viewtime            SMALLINT,
    CONSTRAINT pk_camerasequences PRIMARY KEY (identifier, device,
orderNumber),
    CONSTRAINT fk_camera_sequences FOREIGN KEY (device) REFERENCES CAMERA
(device) ON DELETE CASCADE,
    CONSTRAINT fk_sequences_sequences FOREIGN KEY (identifier) REFERENCES
SEQUENCES (identifier) ON DELETE CASCADE
);
```

**CREATE TABLE REPORTFILTERS**

```
(
    identifier          INTEGER NOT NULL,
    reportName          VARCHAR(1024) NOT NULL,
    filterName          VARCHAR(1024) NOT NULL,
    savePDF             SMALLINT,
    openReport          SMALLINT,
    creationTime         TIMESTAMP(6) NOT NULL,
    createdBy           VARCHAR(256) NOT NULL,
    period              INTEGER,
    units               INTEGER,
    CONSTRAINT pk_reportfilters PRIMARY KEY (identifier)
);
```

**CREATE TABLE ANPRLANEREPORTFILTERS**

```
(
    reportfilter        INTEGER NOT NULL,
    anpr                VARCHAR(256) NOT NULL,
    lane                INTEGER,
    orderNumber         SMALLINT,
    CONSTRAINT pk_anprreportfilter PRIMARY KEY (reportfilter, anpr, lane,
orderNumber),
    CONSTRAINT fk_reportfilters_report FOREIGN KEY (reportfilter) REFERENCES
REPORTFILTERS (identifier) ON DELETE CASCADE
);
```

**CREATE TABLE RADARLANEREPORTFILTERS**

```
(
    reportfilter        INTEGER NOT NULL,
    radar              VARCHAR(256) NOT NULL,
    lane                SMALLINT,
    orderNumber         SMALLINT,
```

```
CONSTRAINT pk_radarlanereportfilter PRIMARY KEY (reportfilter, radar, lane,  
orderNumber),  
CONSTRAINT fk_reportfilters_report FOREIGN KEY (reportfilter) REFERENCES  
REPORTFILTERS (identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE VOIVODESHIPSREPORTFILTERS  
(  
    reportfilter          INTEGER NOT NULL,  
    voivodeship           VARCHAR(256) NOT NULL,  
    orderNumber           SMALLINT,  
    CONSTRAINT pk_voivodeshipreportfilter PRIMARY KEY (reportfilter,  
voivodeship, orderNumber),  
    CONSTRAINT fk_reportfilters_voivo FOREIGN KEY (reportfilter) REFERENCES  
REPORTFILTERS (identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE POWIATSREPORTFILTERS  
(  
    reportfilter          INTEGER NOT NULL,  
    powiat               VARCHAR(256) NOT NULL,  
    orderNumber           SMALLINT,  
    CONSTRAINT pk_powiatreportfilter PRIMARY KEY (reportfilter, powiat,  
orderNumber),  
    CONSTRAINT fk_reportfilters_powiat FOREIGN KEY (reportfilter) REFERENCES  
REPORTFILTERS (identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE CITIESREPORTFILTERS  
(  
    reportfilter          INTEGER NOT NULL,  
    city                 VARCHAR(256) NOT NULL,  
    orderNumber           SMALLINT,  
    CONSTRAINT pk_cityreportfilter PRIMARY KEY (reportfilter, city,  
orderNumber),  
    CONSTRAINT fk_reportfilters_city FOREIGN KEY (reportfilter) REFERENCES  
REPORTFILTERS (identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE DISTRICTSREPORTFILTERS  
(  
    reportfilter          INTEGER NOT NULL,  
    district             VARCHAR(256) NOT NULL,  
    orderNumber           SMALLINT,  
    CONSTRAINT pk_districtreportfilter PRIMARY KEY (reportfilter, district,  
orderNumber),  
    CONSTRAINT fk_reportfilters_district FOREIGN KEY (reportfilter) REFERENCES  
REPORTFILTERS (identifier) ON DELETE CASCADE  
);
```

```
CREATE TABLE SECTIONREPORTFILTERS  
(  
    reportfilter          INTEGER NOT NULL,  
    sectionId            VARCHAR(256) NOT NULL,  
    orderNumber           SMALLINT,  
    CONSTRAINT pk_sectionreportfilter PRIMARY KEY (reportfilter, sectionId,  
orderNumber),
```

```

    CONSTRAINT fk_sectionreportfilters_report FOREIGN KEY (reportfilter)
REFERENCES REPORTFILTERS (identifier) ON DELETE CASCADE
);

```

```

CREATE TABLE REPORT
(
  identifier          INTEGER NOT NULL,
  reportName         VARCHAR(1024) NOT NULL,
  filterName         VARCHAR(1024) NOT NULL,
  creationTime       TIMESTAMP(6) NOT NULL,
  createdBy          VARCHAR(256) NOT NULL,
  link               VARCHAR(1024),
  path               VARCHAR(1024),
  period             INTEGER,
  tsInit1            TIMESTAMP(6),
  tsEnd1             TIMESTAMP(6),
  tsInit2            TIMESTAMP(6),
  tsEnd2             TIMESTAMP(6),
  CONSTRAINT pk_report PRIMARY KEY (identifier)
);

```

```

CREATE TABLE MACRO
(
  folder             VARCHAR(1024) NOT NULL,
  name               VARCHAR(1024) NOT NULL,
  orderNumber        INTEGER NOT NULL,
  objectData         BYTEA,
  description         VARCHAR(1024),
  CONSTRAINT pk_macro PRIMARY KEY (folder, name)
);

```

```

CREATE TABLE CONTACT
(
  name               VARCHAR(1024) NOT NULL,
  description         VARCHAR(1024),
  email              VARCHAR(1024),
  phone              VARCHAR(256),
  company            VARCHAR(1024),
  username           VARCHAR(256) NOT NULL,
  CONSTRAINT pk_contact PRIMARY KEY (name, username)
);

```

```

CREATE TABLE VEHICLECHRONOS
(
  plate              VARCHAR(256) NOT NULL,
  groupName          VARCHAR(256) NOT NULL,
  mark               VARCHAR(256),
  model              VARCHAR(256),
  vehicleColour      VARCHAR(256),
  activationTimeStart TIMESTAMP(6),
  activationTimeEnd   TIMESTAMP(6),
  country            VARCHAR(256),
  owner              VARCHAR(256),
  blacklist           SMALLINT,
  whitelist           SMALLINT,
  username           VARCHAR(256) NOT NULL,
  CONSTRAINT pk_vehicle_chronos PRIMARY KEY (plate, username)
);

```

```

CREATE TABLE SCHEDULE
(
    name          VARCHAR(256) NOT NULL,
    foreground     BIGINT,
    background     BIGINT,
    values         BYTEA,
    username       VARCHAR(256) NOT NULL,
    CONSTRAINT pk_schedule PRIMARY KEY (name, username)
);

CREATE TABLE CALENDAR
(
    name          VARCHAR(256) NOT NULL,
    username       VARCHAR(256) NOT NULL,
    CONSTRAINT pk_calendar PRIMARY KEY (name, username)
);

CREATE TABLE CALENDARDATA
(
    calendar       VARCHAR(256) NOT NULL,
    timestamp      TIMESTAMP(6) NOT NULL,
    schedule       VARCHAR(256) NOT NULL,
    username       VARCHAR(256) NOT NULL,
    CONSTRAINT pk_calendardata PRIMARY KEY (calendar, timestamp, username),
    CONSTRAINT fk_calendar_caldata FOREIGN KEY (calendar, username) REFERENCES
CALENDAR (name, username) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_schedule_caldata FOREIGN KEY (schedule, username) REFERENCES
SCHEDULE (name, username) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE ANPRCALENDARVEHICLEUSER
(
    anpr           VARCHAR(256) NOT NULL,
    calendar       VARCHAR(256) NOT NULL,
    plate          VARCHAR(256) NOT NULL,
    username       VARCHAR(256) NOT NULL,
    groupName      VARCHAR(256) NOT NULL,
    blacklist      SMALLINT,
    whitelist      SMALLINT,
    CONSTRAINT pk_anprcalendarvehuser PRIMARY KEY (anpr, calendar, plate,
username),
    CONSTRAINT fk_calendar_anprcal FOREIGN KEY (calendar, username) REFERENCES
CALENDAR (name, username) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_vehicle_caldata FOREIGN KEY (plate, username) REFERENCES
VEHICLECHRONOS (plate, username) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE CONTACTANPRCALENDARVEHICLEUSER
(
    contact        VARCHAR(1024) NOT NULL,
    anpr           VARCHAR(256) NOT NULL,
    calendar       VARCHAR(256) NOT NULL,
    plate          VARCHAR(256) NOT NULL,
    username       VARCHAR(256) NOT NULL,
    CONSTRAINT pk_contactanprcalendarvehuser PRIMARY KEY (contact, anpr,
calendar, plate, username),

```

```
CONSTRAINT fk_anprcalvehuser_canprcal FOREIGN KEY (anpr, calendar, plate,
username) REFERENCES ANPRCALENDARVEHICLEUSER (anpr, calendar, plate,
username) ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT fk_contact_canprcal FOREIGN KEY (contact, username) REFERENCES
CONTACT (name, username) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE EXTERNALLAYERS
(
    name          VARCHAR(256) NOT NULL,
    info          BYTEA,
    CONSTRAINT pk_externallayers PRIMARY KEY (name)
);

CREATE TABLE EVENT
(
    timestamp      TIMESTAMP(6) NOT NULL,
    moduleId       INTEGER NOT NULL,
    typeId         INTEGER NOT NULL,
    subtypeId      INTEGER NOT NULL,
    username       VARCHAR(256) NOT NULL,
    description     VARCHAR(1024),
    CONSTRAINT pk_event PRIMARY KEY (timestamp, moduleId, typeId, subtypeId,
username),
    CONSTRAINT fk_user_event FOREIGN KEY (username) REFERENCES SYSTEMUSER
(username) ON UPDATE CASCADE ON DELETE CASCADE
);
```