from django.db import models

from django.contrib.auth.models import User

from django.core.validators import MinValueValidator, MaxValueValidator

from django.utils import timezone

class InformeAccidente(models.Model):

# Predefined choices

RESPONSIBLE\_AGENTS = [

('WILSON SAIZ', 'Wilson Saiz'),

('ANDREA GUAVITA', 'Andrea Guavita'),

('SANDRA LOPEZ', 'Sandra Lopez'),

('ANGEL FORERO', 'Angel Forero'),

('MAURICIO MORALES', 'Mauricio Morales'),

('DIEGO PEREZ', 'Diego Perez'),

('CARLOS VARON', 'Carlos Varon'),

('NESTOR VALBUENA', 'Nestor Valbuena'),

('MIGUEL HERNANDEZ', 'Miguel Hernandez'),

('LEONARDO RAMIREZ', 'Leonardo Ramirez'),

('EDER PERILLA', 'Eder Perilla'),

('EDIXON MANCERA', 'Edixon Mancera'),

('RICARDO SANCHEZ', 'Ricardo Sanchez'),

('MARCELA MARTINEZ', 'Marcela Martinez'),

('VIANY NOVA', 'Viany Nova'),

('JAVIER TIMOTE', 'Javier Timote'),

('YAMPIER RUIZ', 'Yampier Ruiz'),

('POLCA', 'Polca'),

]

VIA\_CHOICES = [

('CALLE', 'Calle'),

('CARRERA', 'Carrera'),

('DIAGONAL', 'Diagonal'),

('CASA', 'Casa'),

('LOTE', 'Lote'),

('MANZANA', 'Manzana'),

('AVENIDA', 'Avenida'),

('VEREDA', 'Vereda'),

('KILOMETRO', 'Kilometro'),

('TRANSVERSAL', 'Transversal'),

('VÍA', 'Vía'),

]

AREA\_CHOICES = [

('URBANA', 'Urbana'),

('RURAL', 'Rural'),

]

ZAT\_CHOICES = [

('ZAT1', 'ZAT 1'),

('ZAT2', 'ZAT 2'),

('ZAT3', 'ZAT 3'),

('ZAT4', 'ZAT 4'),

('ZAT5', 'ZAT 5'),

('ZAT6', 'ZAT 6'),

('ZAT7', 'ZAT 7'),

('ZAT8', 'ZAT 8'),

('ZAT9', 'ZAT 9'),

('ZAT10', 'ZAT 10'),

('ZAT11', 'ZAT 11'),

('ZAT12', 'ZAT 12'),

]

COMPLEMENT\_2\_CHOICES = [

('-', '-'),

('CARRERA', 'Carrera'),

('DIAGONAL', 'Diagonal'),

('CALLE', 'Calle'),

('TRANSVERSAL', 'Transversal'),

('ENTRE', 'Entre'),

]

# ... (continue with all the rural localities from the list)

RURAL\_LOCALITIES = [

('PATIO BONITO', 'Patio Bonito'),

('MONTEBELLO', 'Montebello'),

('LA PRIMAVERA', 'La Primavera'),

('SAN ISIDRO DE CHICHIMENE', 'San Isidro de Chichimene'),

('SANTA ROSA', 'Santarosa'),

('LOMA DE TIGRE', 'Loma de Tigre'),

('EL TRIUNFO', 'El Triunfo'),

('LA ESMERALDA', 'La Esmeralda'),

('LA UNIÓN', 'La Unión'),

('EL CENTRO', 'El centro'),

('SAN NICOLAS', 'San Nicolas'),

('DINAMARCA', 'Dinamarca'),

('QUEBRADITAS', 'Quebraditas'),

('LA LOMA', 'La Loma'),

('SAN JOSE DE LAS PALOMAS', 'San jose de las Palomas'),

('SAN CAYETANO', 'San cayetano'),

('LAS MARGARITAS', 'Las Margaritas'),

('CAÑO HONDO', 'Caño Hondo'),

('MOTELIBANO', 'Montelibano'),

('SANTA TERERSITA', 'Santa Teresita'),

('MOTELIBANO BAJO', 'Montelibano Bajo'),

('EL RESGUARDO', 'El Resguardo'),

('EL ROSARIO', 'El Rosario'),

('SARDINATA', 'Sardinata'),

('LA CECILITA', 'La Cecilita'),

('RANCHO GRANDE', 'Rancho Grande'),

('CRUCE DE SAN JOSE', 'Cruce de San Jose'),

('EL PLAYON', 'El Playón'),

('SAN JUANITO', 'San Juanito'),

('FRESCO VALLE', 'Fresco Valle'),

('EL RECREO', 'El recreo'),

('ALTO ACACIITAS', 'Alto Acaciitas'),

('ALTO ACACIAS', 'Alto Acacías'),

('SAN CRISTOBAL', 'San Cristobal'),

('LAS BLANCAS', 'Las Blancas'),

('EL DIAMANTE', 'El Diamante'),

('LOMA DEL PAÑUELO', 'Loma del Pañuelo'),

('LOMA DE SAN JUAN', 'Loma de San Juan'),

('LOMA DE SAN PABLO', 'Loma de San Pablo'),

('BRISAS DEL GUAYURIBA', 'Brisas del Guayuriba'),

('LA PRADERA', 'La Pradera'),

('VISTA HERMOSA', 'Vista Hermosa'),

('LABERINTO', 'Laberinto'),

('LIBANO', 'Libano'),

('LOS PINOS ', 'Los Pinos'),

('PORTACHUELO', 'Portachuelo'),

('MANZANARES', 'Manzanares'),

('VENECIA', 'Venecia'),

]

# ZAT Neighborhoods Mapping

ZAT1\_NEIGHBORHOODS = [

'ASOCIACION DE AMIGOS', 'VILLA MAQUENCI', 'LA ESTRELLA', 'VILLA CASTILLA',

'VILLA MANUELA', 'VILLA AURORA 1', 'VILLA AURORA 2', 'LOS CEDROS',

'PRADOS DE CODEM', 'BELLA SUIZA', 'ALTOS DE COVICOM', 'NUEVO HORIZONTE',

'GAVIOTAS', 'VILLA DEL LLANO'

]

ZAT2\_NEIGHBORHOODS = [

' VIOLETAS', 'VILLA LUCIA', 'DON BOSCO', 'LOS OLIVOS',

'VILLA MAGALY', 'GRUPO LOS 18', 'ATAHUALPA', 'CIUDADELA LOS ANGELES',

'GUARATARA 2', 'LA TIZA', 'EL TREBOL', 'EL PALMAR'

]

ZAT3\_NEIGHBORHOODS = [

' BAMBU', 'LA ALBORADA', 'PORTALES DE SAN CARLOS', 'MORICHAL 1',

'MORICHAL 2', 'VILLAS DE SAN CARLOS', 'ARAGUANEY ', 'LA ESPERANZA',

'LAS VILLAS ', 'SANTA ANA', 'CONDADO NUEVO MILENIO', 'CIUDADELA EL CONSTRUCTOR'

]

ZAT4\_NEIGHBORHOODS = [

' PABLO VI', 'LA PALMA', 'LA CORALINA', 'MANCERA'

]

ZAT5\_NEIGHBORHOODS = [

' VILLA TERESA', 'LAS VEGAS', 'PALERMO', 'POPULAR'

]

ZAT6\_NEIGHBORHOODS = [

' ALTOS DE LA FLORIDA', 'LA FLORIDA', 'LAS COLINAS', 'CIMARRON',

'COMCAJA', 'BALCONES DE SAN DIEGO', 'ALTOS DE LUCIA'

]

ZAT7\_NEIGHBORHOODS = [

'LA INDEPENDENCIA', 'NUEVA VICTORIA', 'BACHUE', 'ASOVIVIENDA',

'NUTIBARA', 'RINCON DE BACHUE', 'ASOMENDA', 'EL PARAISO',

'LA PRADERA', 'LAS ACACIAS', 'EL BOSQUE'

]

ZAT8\_NEIGHBORHOODS = [

' BRISAS DEL PLAYON', 'LA PRIMAVERA ETAPA 1', 'LA PRIMAVERA ETAPA 2', 'LOS LAURELES',

'DIVINO NIÑO', 'EL TREBOL', 'OASIS', 'PINOS',

'EL JORDAN ', 'CEDRITOS', 'SAMAN', 'PANORAMA', 'ALCARAVAN'

]

ZAT9\_NEIGHBORHOODS = [

'SAN JOSE', 'DORADO ALTO', 'DORADO BAJO', 'LOS NARANJOS',

'COOPERATIVO'

]

ZAT10\_NEIGHBORHOODS = [

'ARRAYANES', 'BALCONES DE SANTA ISABEL', 'BRISAS', 'LLANOMAR',

'COOPERATIVO'

]

ZAT11\_NEIGHBORHOODS = [

'URBANIZACION VILLA DEL PRADO', 'URBANIZACION EVEREST', 'PABLO EMILIO RIVEROS', 'EL RETORNO',

'SANTA ISABEL', 'CIUDAD JARDIN', 'LAS FERIAS'

]

ZAT12\_NEIGHBORHOODS = [

'BALMORAL', 'JUAN MELLAO', 'LA UNIÓN', 'EL CENTRO'

]

# Similar lists for ZAT2, ZAT3, etc. can be added following the same pattern

# Auto-incrementing Accident Number

numero\_accidente = models.AutoField(primary\_key=True)

# IPAT Number (15-character integer)

numero\_ipat = models.CharField(max\_length=15, unique=True)

# Responsible Agent

agente\_responsable = models.CharField(

max\_length=50,

choices=RESPONSIBLE\_AGENTS

)

# Accident Date

fecha\_accidente = models.DateField()

# Accident Month (automatically derived from fecha\_accidente)

mes\_accidente = models.CharField(

max\_length=20,

choices=[

('ENERO', 'Enero'), ('FEBRERO', 'Febrero'),

('MARZO', 'Marzo'), ('ABRIL', 'Abril'),

('MAYO', 'Mayo'), ('JUNIO', 'Junio'),

('JULIO', 'Julio'), ('AGOSTO', 'Agosto'),

('SEPTIEMBRE', 'Septiembre'), ('OCTUBRE', 'Octubre'),

('NOVIEMBRE', 'Noviembre'), ('DICIEMBRE', 'Diciembre')

]

)

# Fixed Days for IPAT Delivery

dias\_establecidos\_entrega = models.IntegerField(default=1)

# Established Date for IPAT Delivery (accident date + 1 day)

fecha\_establecida\_entrega = models.DateField()

# Real IPAT Delivery Date

fecha\_real\_entrega = models.DateField(null=True, blank=True)

# Real Days of IPAT Delivery

dias\_reales\_entrega = models.IntegerField(null=True, blank=True)

# Delivery Status (Green if within terms)

entregado\_dentro\_terminos = models.BooleanField(default=False)

# Accident Time (24-hour format, Colombian time zone)

hora\_accidente = models.TimeField()

# Via (Street Type)

via = models.CharField(

max\_length=20,

choices=VIA\_CHOICES

)

# Street Number (3 characters)

numero = models.CharField(max\_length=3)

# Additional Address Complement 1

complemento\_1 = models.CharField(max\_length=100, blank=True, null=True)

# Additional Address Complement 2

complemento\_2 = models.CharField(

max\_length=20,

choices=COMPLEMENT\_2\_CHOICES,

blank=True,

null=True

)

# Additional Address Information

otra\_informacion\_direccion = models.CharField(max\_length=200, blank=True, null=True)

# Concatenated Address

direccion\_concatenada = models.CharField(max\_length=500)

# Area Type

area = models.CharField(

max\_length=10,

choices=AREA\_CHOICES

)

# ZAT (Territorial Administrative Zone)

zat = models.CharField(

max\_length=10,

choices=ZAT\_CHOICES

)

# Rural Locality (only if Area is Rural)

centro\_poblado\_vereda = models.CharField(

max\_length=100,

choices=RURAL\_LOCALITIES,

blank=True,

null=True

)

def save(self, \*args, \*\*kwargs):

# Automatically set month based on accident date

self.mes\_accidente = self.fecha\_accidente.strftime('%B').upper()

# Automatically set established delivery date (accident date + 1 day)

self.fecha\_establecida\_entrega = self.fecha\_accidente + timezone.timedelta(days=1)

# Calculate real delivery days if both dates are available

if self.fecha\_real\_entrega and self.fecha\_establecida\_entrega:

self.dias\_reales\_entrega = (self.fecha\_real\_entrega - self.fecha\_establecida\_entrega).days

# Check if delivery was within terms

self.entregado\_dentro\_terminos = self.dias\_reales\_entrega <= self.dias\_establecidos\_entrega

# Concatenate address

address\_parts = [

self.via,

self.numero,

self.complemento\_1 or '',

self.complemento\_2 or '',

self.otra\_informacion\_direccion or ''

]

self.direccion\_concatenada = ' '.join(filter(bool, address\_parts))

super().save(\*args, \*\*kwargs)

def \_\_str\_\_(self):

return f"Accident {self.numero\_accidente} - {self.fecha\_accidente}"

# Choices for various fields

TOTAL\_VEHICLES\_CHOICES = [(i, str(i)) for i in range(1, 11)]

ACCIDENT\_TYPE\_CHOICES = [

('HERIDOS', 'Heridos'),

('MUERTOS', 'Muertos'),

('DAÑOS MATERIALES', 'Daños Materiales')

]

ACCIDENT\_CLASS\_CHOICES = [

('ATROPELLO', 'Atropello'),

('CAIDA', 'Caída'),

('COLISIÓN', 'Colisión'),

('CHOQUE', 'Choque'),

('VOLCAMIENTO', 'Volcamiento'),

('OTRO', 'Otro')

]

ROAD\_TYPE\_CHOICES = [

('RURAL', 'Rural'),

('URBANA', 'Urbana'),

('NACIONAL', 'Nacional'),

('DEPARTAMENTAL', 'Departamental'),

('MUNICIPAL', 'Municipal')

]

COLLISION\_TYPE\_CHOICES = [

('VEHICULO', 'Vehículo'),

('SEMOVIENTE', 'Semoviente'),

('OBJETO FIJO', 'Objeto Fijo')

]

FIXED\_OBJECT\_CHOICES = [

('MURO', 'Muro'),

('POSTE', 'Poste'),

('ARBOL', 'Árbol'),

('BARANDA', 'Baranda'),

('SEMAFORO', 'Semáforo'),

('INMUEBLE', 'Inmueble'),

('HIDRANTE', 'Hidrante'),

('VALLA', 'Valla'),

('SEÑAL', 'Señal'),

('TARIMA', 'Tarima'),

('CASETA', 'Caseta'),

('VEHICULO ESTACIONADO', 'Vehículo Estacionado'),

('OTRO', 'Otro')

]

SERVICE\_TYPE\_CHOICES = [

('PARTICULAR', 'Particular'),

('PUBLICO', 'Público'),

('OFICIAL', 'Oficial'),

('DIPLOMATICO', 'Diplomático')

]

VEHICLE\_CLASS\_CHOICES = [

('AUTOMOVIL', 'Automóvil'),

('BUS', 'Bus'),

('BUSETA', 'Buseta'),

('BICICLETA', 'Bicicleta'),

('CAMION', 'Camión'),

('CAMIONETA', 'Camioneta'),

('CAMPERO', 'Campero'),

('TRACTOCAMION', 'Tractocamión - Camión Tractor'),

('CUATRIMOTO', 'Cuatrimoto'),

('MICROBUS', 'Microbus'),

('MOTOCARRO', 'Motocarro'),

('MOTOCICLETA', 'Motocicleta'),

('MOTOTRICICLO', 'Mototriciclo'),

('VOLQUETA', 'Volqueta'),

('VEHICULO TRACCION ANIMAL', 'Vehículo de Tracción Animal'),

('FUGA DE VEHICULO', 'Fuga de Vehículo'),

('VEHICULO FANTASMA', 'Vehículo Fantasma')

]

GENDER\_CHOICES = [

('MASCULINO', 'Masculino'),

('FEMENINO', 'Femenino')

]

AGE\_RANGE\_CHOICES = [

('PRIMERA INFANCIA', 'Primera Infancia (0-5 años)'),

('INFANCIA', 'Infancia (6-11 años)'),

('ADOLESCENCIA', 'Adolescencia (12-18 años)'),

('JUVENTUD', 'Juventud (14-26 años)'),

('ADULTEZ', 'Adultez (27-59 años)'),

('PERSONA MAYOR', 'Persona Mayor (60 años o más)')

]

INJURED\_TYPE\_CHOICES = [

('PEATON', 'Peatón'),

('CONDUCTOR', 'Conductor'),

('ACOMPAÑANTE', 'Acompañante'),

('PASAJERO', 'Pasajero'),

('NO APLICA', 'No Aplica')

]

BOOLEAN\_CHOICES = [

('SI', 'Sí'),

('NO', 'No'),

('NO APLICA', 'No Aplica')

]

INTOXICATION\_LEVEL\_CHOICES = [

('GRADO 1', 'Grado 1'),

('GRADO 2', 'Grado 2'),

('GRADO 3', 'Grado 3')

]

# Main Accident Details

total\_vehiculos\_involucrados = models.IntegerField(

choices=TOTAL\_VEHICLES\_CHOICES,

validators=[MinValueValidator(1), MaxValueValidator(10)]

)

accidente\_con = models.CharField(

max\_length=50,

choices=ACCIDENT\_TYPE\_CHOICES,

blank=True,

null=True

)

clase\_accidente = models.CharField(

max\_length=20,

choices=ACCIDENT\_CLASS\_CHOICES

)

clase\_accidente\_otro = models.CharField(

max\_length=100,

blank=True,

null=True,

help\_text="Solo llenar si se seleccionó 'OTRO' en clase de accidente"

)

tipo\_via = models.CharField(

max\_length=20,

choices=ROAD\_TYPE\_CHOICES

)

choque\_con = models.CharField(

max\_length=20,

choices=COLLISION\_TYPE\_CHOICES

)

objeto\_fijo = models.CharField(

max\_length=30,

choices=FIXED\_OBJECT\_CHOICES,

blank=True,

null=True

)

objeto\_fijo\_otro = models.CharField(

max\_length=100,

blank=True,

null=True,

help\_text="Solo llenar si se seleccionó 'OTRO' en objeto fijo"

)

# Vehicle 1 Details

tipo\_servicio\_vehiculo\_1 = models.CharField(

max\_length=20,

choices=SERVICE\_TYPE\_CHOICES

)

clase\_vehiculo\_1 = models.CharField(

max\_length=30,

choices=VEHICLE\_CLASS\_CHOICES

)

genero\_involucrado\_vehiculo\_1 = models.CharField(

max\_length=10,

choices=GENDER\_CHOICES

)

rango\_edad\_vehiculo\_1 = models.CharField(

max\_length=20,

choices=AGE\_RANGE\_CHOICES

)

heridos\_vehiculo\_1 = models.CharField(

max\_length=20,

choices=INJURED\_TYPE\_CHOICES

)

fallecidos\_vehiculo\_1 = models.CharField(

max\_length=20,

choices=INJURED\_TYPE\_CHOICES

)

embriaguez\_conductor\_1 = models.CharField(

max\_length=20,

choices=BOOLEAN\_CHOICES

)

grado\_embriaguez\_1 = models.CharField(

max\_length=10,

choices=INTOXICATION\_LEVEL\_CHOICES,

blank=True,

null=True

)

numero\_heridos\_vehiculo\_1 = models.IntegerField(

choices=TOTAL\_VEHICLES\_CHOICES,

validators=[MinValueValidator(0), MaxValueValidator(10)]

)

numero\_fallecidos\_vehiculo\_1 = models.IntegerField(

choices=TOTAL\_VEHICLES\_CHOICES,

validators=[MinValueValidator(0), MaxValueValidator(10)]

)

herido\_1\_fallece\_30\_dias\_1 = models.CharField(

max\_length=20,

choices=BOOLEAN\_CHOICES

)

herido\_2\_fallece\_30\_dias\_1 = models.CharField(

max\_length=20,

choices=BOOLEAN\_CHOICES,

blank=True,

null=True

)

herido\_3\_fallece\_30\_dias\_1 = models.CharField(

max\_length=20,

choices=BOOLEAN\_CHOICES,

blank=True,

null=True

)

# Fallecidos Vehicle 1 Details

nombre\_fallecido\_1\_vehiculo\_1 = models.CharField(

max\_length=200,

blank=True,

null=True

)

direccion\_fallecido\_1\_vehiculo\_1 = models.CharField(

max\_length=300,

blank=True,

null=True

)

nombre\_fallecido\_2\_vehiculo\_1 = models.CharField(

max\_length=200,

blank=True,

null=True

)

direccion\_fallecido\_2\_vehiculo\_1 = models.CharField(

max\_length=300,

blank=True,

null=True

)

nombre\_fallecido\_3\_vehiculo\_1 = models.CharField(

max\_length=200,

blank=True,

null=True

)

direccion\_fallecido\_3\_vehiculo\_1 = models.CharField(

max\_length=300,

blank=True,

null=True

)

# Similar fields for Vehicle 2, 3, and 4 would follow the same pattern

# (I'll omit them for brevity, but they would be structured identically)

# Calculated Total Fields

total\_heridos = models.IntegerField(default=0)

total\_fallecidos = models.IntegerField(default=0)

def save(self, \*args, \*\*kwargs):

# Calculate total injured and deceased

self.total\_heridos = (

self.numero\_heridos\_vehiculo\_1 +

# Add other vehicle injury counts here when implemented

0 # Placeholder for now

)

self.total\_fallecidos = (

self.numero\_fallecidos\_vehiculo\_1 +

# Add other vehicle fatality counts here when implemented

0 # Placeholder for now

)

super().save(\*args, \*\*kwargs)

def \_\_str\_\_(self):

return f"Accident Report - {self.total\_vehiculos\_involucrados} vehicles"

class Meta:

verbose\_name = "Accident Report"

verbose\_name\_plural = "Accident Reports"