

Soshogle CRM - Database Schema Documentation



Database Technology

ORM: Prisma

Database: PostgreSQL

Schema File: prisma/schema.prisma



Core Models

User

Primary user account model for authentication and ownership.

```

model User {
    id          String      @id @default(cuid())
    email       String      @unique
    password    String      // bcrypt hashed
    name        String?
    role        Role        @default(USER)

    // Profile
    phone       String?
    company    String?
    avatar     String?
    timezone   String      @default("America/New_York")

    // Onboarding
    onboardingComplete Boolean   @default(false)
    onboardingStep     Int      @default(0)

    // Company Profile (from onboarding)
    businessName  String?
    industry      String?
    website       String?
    companyAddress String?
    businessLanguage String?
    businessDescription String?

    // Timestamps
    createdAt    DateTime   @default(now())
    updatedAt    DateTime   @updatedAt
    emailVerified DateTime?

    // Relations
    accounts     Account[]
    sessions     Session[]
    contacts    Contact[]
    leads        Lead[]
    deals        Deal[]
    voiceAgents  VoiceAgent[]
    campaigns    Campaign[]
    workflows    Workflow[]
    appointments Appointment[]
    messages     Message[]
    payments     Payment[]
    calendarConnections CalendarConnection[]

    @@index([email])
}

enum Role {
    USER
    ADMIN
    AGENCY
    SUB_ACCOUNT
}

```

Contact

Customer/client contact information.

```

model Contact {
    id          String      @id @default(cuid())
    userId      String
    user        User        @relation(fields: [userId], references: [id], onDelete: Cascade)
    scade)

    // Basic Info
    firstName   String
    lastName    String?
    email       String?
    phone       String?
    company     String?

    // Segmentation
    tags         String[]   // ["VIP", "Customer", "Prospect"]
    source       String?    // "manual", "import", "lead_conversion", "website"

    // Custom Data
    customFields Json?     // Flexible key-value storage

    // Timestamps
    createdAt   DateTime   @default(now())
    updatedAt   DateTime   @updatedAt

    // Relations
    messages    Message[]
    appointments Appointment[]
    deals       DealContact[]
    notes       Note[]

    @@index([userId])
    @@index([email])
    @@index([phone])
    @@index([tags])
}

```

Lead

Business leads scraped from Google Places or manually added.

```

model Lead {
    id          String      @id @default(cuid())
    userId      String
    user        User        @relation(fields: [userId], references: [id], onDelete: Cascade)
    state       String
    zipCode     String
    country     String      @default("US")
    phone       String?
    website     String?
    email       String?

    // Business Info
    businessName String
    address      String
    city         String
    state        String
    zipCode      String
    country      String      @default("US")
    phone        String?
    website      String?
    email        String?

    // Metadata
    rating       Float?
    reviewCount Int?
    category     String      // "dentist", "restaurant", "salon", etc.

    // Google Places
    googlePlaceId String?  @unique

    // Status
    status        LeadStatus @default(NEW)
    source        String      // "google_places", "manual", "import"

    // Timestamps
    createdAt    DateTime   @default(now())
    updatedAt    DateTime   @updatedAt

    // Relations
    notes        Note[]

    @@index([userId])
    @@index([status])
    @@index([category])
    @@index([city])
}

enum LeadStatus {
    NEW
    CONTACTED
    QUALIFIED
    CONVERTED
    REJECTED
}

```

Deal

Sales pipeline opportunities.

```

model Deal {
    id          String      @id @default(cuid())
    userId      String
    user        User        @relation(fields: [userId], references: [id], onDelete: Cascade)
    value       Float       @default(0)
    currency    String      @default("USD")

    // Deal Info
    title       String
    description String?
    value       Float       @default(0)
    currency    String      @default("USD")

    // Pipeline
    stage       DealStage   @default(LEAD)
    probability Int        @default(0) // 0-100%

    // Dates
    expectedCloseDate DateTime?
    actualCloseDate   DateTime?

    // Timestamps
    createdAt    DateTime   @default(now())
    updatedAt    DateTime   @updatedAt

    // Relations
    contacts    DealContact[]
    notes       Note[]

    @@index([userId])
    @@index([stage])
}

enum DealStage {
    LEAD
    QUALIFIED
    PROPOSAL
    NEGOTIATION
    CLOSED_WON
    CLOSED_LOST
}

model DealContact {
    dealId     String
    deal       Deal        @relation(fields: [dealId], references: [id], onDelete: Cascade)
    contactId String
    contact    Contact   @relation(fields: [contactId], references: [id], onDelete: Cascade)

    isPrimary Boolean   @default(false)

    @@id([dealId, contactId])
}

```

Note

Notes attached to contacts, deals, or leads.

```

model Note {}
  id      String    @id @default(cuid())
  // Content
  content String

  // Relationships (polymorphic)
  contactId String?
  contact   Contact? @relation(fields: [contactId], references: [id], onDelete: Cascade)
  dealId    String?
  deal      Deal?    @relation(fields: [dealId], references: [id], onDelete: Cascade)
  leadId    String?
  lead      Lead?    @relation(fields: [leadId], references: [id], onDelete: Cascade)

  // Timestamps
  createdAt DateTime @default(now())
  updatedAt DateTime @updatedAt

  @@index([contactId])
  @@index([dealId])
  @@index([leadId])
}

```

Tag

Centralized tag management (optional - currently using String[]).

```

model Tag {}
  id      String    @id @default(cuid())
  userId String
  name   String
  color   String?   // Hex color code

  createdAt DateTime @default(now())

  @@unique([userId, name])
  @@index([userId])
}

```

Communication Models

VoiceAgent

ElevenLabs AI voice agents with Twilio phone numbers.

```

model VoiceAgent {}  

  id          String    @id @default(cuid())  

  userId      String  

  user        User      @relation(fields: [userId], references: [id], onDelete  

e: Cascade)  

  

  // Agent Info  

  name        String  

  

  // ElevenLabs Integration  

  elevenLabsAgentId String? @unique  

  phoneNumberId   String? // ElevenLabs phone number ID  

  

  // Twilio Integration  

  twilioPhoneNumber String? @unique  

  

  // Configuration  

  firstMessage   String?  

  systemPrompt   String?  

  language       String    @default("en")  

  

  // Status  

  status         AgentStatus @default(ACTIVE)  

  

  // Timestamps  

  createdAt      DateTime @default(now())  

  updatedAt      DateTime @updatedAt  

  

  // Relations  

  calls          Call[]  

  

  @@index([userId])  

  @@index([elevenLabsAgentId])  

}  

  

enum AgentStatus {}  

  ACTIVE  

  INACTIVE  

  ERROR
}

```

Call

Call logs (inbound and outbound).

```

model Call {
    id           String      @id @default(cuid())
    voiceAgentId String?
    voiceAgent   VoiceAgent? @relation(fields: [voiceAgentId], references: [id])

    // Call Details
    direction    CallDirection
    from         String      // Phone number
    to          String      // Phone number

    // Twilio
    callSid      String?    @unique
    duration     Int?       // Seconds
    recordingUrl String?

    // Status
    status       CallStatus @default(QUEUED)

    // Timestamps
    startedAt    DateTime?
    endedAt      DateTime?
    createdAt    DateTime  @default(now())

    @@index([voiceAgentId])
    @@index([callSid])
}

enum CallDirection {
    INBOUND
    OUTBOUND
}

enum CallStatus {
    QUEUED
    RINGING
    IN_PROGRESS
    COMPLETED
    FAILED
    BUSY
    NO_ANSWER
}

```

Message

Unified messaging (SMS, Email, WhatsApp).

```

model Message {
    id          String      @id @default(cuid())
    userId      String
    user        User        @relation(fields: [userId], references: [id], onDelete: Cascade)
    contactId   String?
    contact     Contact?   @relation(fields: [contactId], references: [id], onDelete: Cascade)

    // Message Content
    content      String
    subject     String?    // For emails

    // Channel
    channel      MessageChannel @default(SMS)
    direction    MessageDirection

    // Identifiers
    from         String      // Phone/email
    to          String      // Phone/email

    // Status
    status       MessageStatus @default(SENT)

    // External IDs
    twilioSid    String?
    sendgridId   String?

    // Timestamps
    createdAt    DateTime   @default(now())
    updatedAt    DateTime   @updatedAt

    // Relations
    conversationId String?
    conversation   Conversation? @relation(fields: [conversationId], references: [id])
    @index([userId])
    @index([contactId])
    @index([conversationId])
    @index([channel])
}

enum MessageChannel {
    SMS
    EMAIL
    WHATSAPP
    FACEBOOK
    INSTAGRAM
}

enum MessageDirection {
    INBOUND
    OUTBOUND
}

enum MessageStatus {
    QUEUED
    SENT
    DELIVERED
    READ
    FAILED
}

```

```
model Conversation {
    id      String    @id @default(cuid())
    contactId String
    channel  MessageChannel

    messages Message[]

    createdAt DateTime @default(now())
    updatedAt DateTime @updatedAt

    @@unique([contactId, channel])
    @@index([contactId])
}
```

Marketing & Automation

Campaign

Email and SMS campaigns.

```

model Campaign {
    id          String      @id @default(cuid())
    userId      String
    user        User        @relation(fields: [userId], references: [id], onDelete: Cascade)
    // Campaign Info
    name        String
    type        CampaignType
    // Targeting
    targetTags  String[]   // ["VIP", "Customer"]
    // Content
    subject     String?    // For emails
    message     String
    // Status
    status      CampaignStatus @default(DRAFT)
    // Scheduling
    scheduledAt DateTime?
    sentAt      DateTime?
    // Metrics
    sentCount   Int         @default(0)
    deliveredCount Int       @default(0)
    openCount   Int         @default(0)
    clickCount  Int         @default(0)
    // Timestamps
    createdAt   DateTime   @default(now())
    updatedAt   DateTime   @updatedAt
    @@index([userId])
    @@index([status])
}

enum CampaignType {
    EMAIL
    SMS
}

enum CampaignStatus {
    DRAFT
    SCHEDULED
    ACTIVE
    COMPLETED
    CANCELLED
}

```

Workflow

Automation workflows with triggers, conditions, and actions.

```

model Workflow {}  

  id          String    @id @default(cuid())  

  userId      String  

  user        User      @relation(fields: [userId], references: [id], onDelete: Cascade)  

  

  // Workflow Info  

  name        String  

  description String?  

  

  // Trigger  

  trigger     WorkflowTrigger  

  

  // Conditions & Actions (JSON)  

  conditions   Json?    // Array of condition objects  

  actions      Json      // Array of action objects  

  

  // Status  

  status       WorkflowStatus @default(ACTIVE)  

  

  // Metrics  

  executionCount Int      @default(0)  

  lastExecutedAt DateTime?  

  

  // Timestamps  

  createdAt    DateTime  @default(now())  

  updatedAt    DateTime  @updatedAt  

  

  // Relations  

  executions   WorkflowExecution[]  

  

  @@index([userId])  

  @@index([status])  

}  

  

enum WorkflowTrigger {}  

  CONTACT_CREATED  

  CONTACT_UPDATED  

  TAG_ADDED  

  TAG_REMOVED  

  DEAL_CREATED  

  DEAL_STAGE_CHANGED  

  APPOINTMENT_SCHEDULED  

  MESSAGE_RECEIVED  

  FORM_SUBMITTED  

  MANUAL  

}  

  

enum WorkflowStatus {}  

  ACTIVE  

  INACTIVE  

}  

  

model WorkflowExecution {}  

  id          String    @id @default(cuid())  

  workflowId String  

  workflow    Workflow  @relation(fields: [workflowId], references: [id], onDelete: Cascade)  

  

  // Execution Details  

  status      ExecutionStatus  

  error       String?

```

```
// Context
contextData Json      // Data that triggered the workflow

// Timestamps
startedAt  DateTime @default(now())
completedAt DateTime?

@@index([workflowId])
@@index([status])
}

enum ExecutionStatus {
  PENDING
  RUNNING
  COMPLETED
  FAILED
}
```

Calendar & Appointments

Appointment

Scheduled appointments with contacts.

```

model Appointment {
    id          String      @id @default(cuid())
    userId      String
    user        User        @relation(fields: [userId], references: [id], onDelete: Cascade)
    contactId   String?
    contact     Contact?   @relation(fields: [contactId], references: [id], onDelete: Cascade)

    // Appointment Info
    title       String
    description String?
    location    String?

    // Timing
    startTime   DateTime
    endTime     DateTime
    timezone   String      @default("America/New_York")

    // Status
    status      AppointmentStatus @default(SCHEDULED)

    // External Calendar
    externalEventId String?  // Google/Microsoft event ID
    calendarProvider String? // "google", "microsoft", "apple"

    // Reminders
    reminderSent Boolean    @default(false)

    // Timestamps
    createdAt   DateTime   @default(now())
    updatedAt   DateTime   @updatedAt

    @@index([userId])
    @@index([contactId])
    @@index([startTime])
}

enum AppointmentStatus {
    SCHEDULED
    CONFIRMED
    CANCELLED
    COMPLETED
    NO_SHOW
}

```

CalendarConnection

External calendar sync settings.

```

model CalendarConnection {
    id          String    @id @default(cuid())
    userId      String
    user        User      @relation(fields: [userId], references: [id], onDelete: Cascade)
    provider    String    // "google", "microsoft", "apple"

    // OAuth
    accessToken  String?
    refreshToken String?
    tokenExpiry DateTime?

    // Settings
    syncEnabled Boolean   @default(true)
    lastSyncedAt DateTime?

    // Timestamps
    createdAt    DateTime @default(now())
    updatedAt    DateTime @updatedAt

    @@index([userId])
}

```

Payments & Billing

Payment

Payment transactions.

```

model Payment {
    id          String      @id @default(cuid())
    userId      String
    user        User        @relation(fields: [userId], references: [id], onDelete: Cascade)
    amount      Float
    currency    String      @default("USD")
    description String?

    // Provider
    provider    PaymentProvider

    // External IDs
    stripePaymentIntentId String?
    paypalOrderId   String?
    squarePaymentId String?

    // Status
    status       PaymentStatus @default(PENDING)

    // Timestamps
    createdAt    DateTime   @default(now())
    updatedAt    DateTime   @updatedAt
    paidAt      DateTime?

    @@index([userId])
    @@index([status])
}

enum PaymentProvider {
    STRIPE
    PAYPAL
    SQUARE
}

enum PaymentStatus {
    PENDING
    PROCESSING
    SUCCEEDED
    FAILED
    REFUNDED
}

```

Authentication (NextAuth)

Account

OAuth account connections.

```

model Account {
  id           String @id @default(cuid())
  userId       String
  user         User   @relation(fields: [userId], references: [id], onDelete: Cascade)
  type         String
  provider     String
  providerAccountId String
  refresh_token String?
  access_token  String?
  expires_at    Int?
  token_type    String?
  scope         String?
  id_token      String?
  session_state String?

  @@unique([provider, providerAccountId])
  @@index([userId])
}

```

Session

Active user sessions.

```

model Session {
  id           String @id @default(cuid())
  sessionToken String @unique
  userId       String
  user         User   @relation(fields: [userId], references: [id], onDelete: Cascade)
  expires      DateTime

  @@index([userId])
}

```

VerificationToken

Email verification tokens.

```

model VerificationToken {
  identifier String
  token      String @unique
  expires    DateTime

  @@unique([identifier, token])
}

```



Indexes & Performance

Key Indexes

- **User:** email (unique)
 - **Contact:** userId, email, phone, tags
 - **Lead:** userId, status, category, city, googlePlaceId (unique)
 - **Deal:** userId, stage
 - **VoiceAgent:** userId, elevenLabsAgentId (unique), twilioPhoneNumber (unique)
 - **Call:** voiceAgentId, callSid (unique)
 - **Message:** userId, contactId, conversationId, channel
 - **Campaign:** userId, status
 - **Workflow:** userId, status
 - **Appointment:** userId, contactId, startTime
-



Migrations

Running Migrations

```
# Development
yarn prisma migrate dev --name migration_name

# Production
yarn prisma migrate deploy
```

Generate Prisma Client

```
yarn prisma generate
```

View Database (Prisma Studio)

```
yarn prisma studio
```

Relationships Summary

User (1) —— (M) Contact
 |
 —— (M) Lead
 —— (M) Deal
 —— (M) VoiceAgent
 —— (M) Campaign
 —— (M) Workflow
 —— (M) Appointment
 —— (M) Message
 —— (M) Payment

VoiceAgent (1) —— (M) Call

Contact (1) —— (M) Message
 |
 —— (M) Appointment
 —— (M) Note
 —— (M) DealContact

Deal (1) —— (M) Note
 |
 —— (M) DealContact

Lead (1) —— (M) Note

Workflow (1) —— (M) WorkflowExecution

Conversation (1) —— (M) Message

Last Updated: November 2025

Schema Version: 2.0