schema.prisma – Bot Cripto Telegram

Banco: PostgreSQL (Supabase). Cliente: prisma-client-js.

# Conteúdo

generator client {  
 provider = "prisma-client-js"  
}  
  
datasource db {  
 provider = "postgresql"  
 url = env("DATABASE\_URL")  
}  
  
enum AlertDirection { ACIMA ABAIXO }  
enum TradeSide { BUY SELL }  
enum LogLevel { INFO WARN ERROR }  
enum ApiService { BINANCE TELEGRAM RSS OPENAI TAAPI REDIS SUPABASE }  
  
model User {  
 id String @id @default(uuid()) @db.Uuid  
 chatId BigInt @unique  
 username String?  
 language String?  
 isActive Boolean @default(true)  
 createdAt DateTime @default(now())  
  
 alerts Alert[]  
 trades Trade[]  
 logs BotLog[]  
}  
  
model Alert {  
 id BigInt @id @default(autoincrement())  
 userId String @db.Uuid  
 symbol String  
 direction AlertDirection  
 targetBrl Decimal @db.Decimal(18, 2)  
 active Boolean @default(true)  
 triggeredAt DateTime?  
 createdAt DateTime @default(now())  
 user User @relation(fields: [userId], references: [id], onDelete: Cascade)  
 @@index([userId, active, symbol])  
}  
  
model Trade {  
 id BigInt @id @default(autoincrement())  
 userId String @db.Uuid  
 symbol String  
 side TradeSide  
 qty Decimal @db.Decimal(28, 12)  
 priceBrl Decimal @db.Decimal(18, 2)  
 feeBrl Decimal @db.Decimal(18, 2) @default(0)  
 ts DateTime @default(now())  
 user User @relation(fields: [userId], references: [id], onDelete: Cascade)  
 @@index([userId, symbol, ts])  
}  
  
model BotLog {  
 id BigInt @id @default(autoincrement())  
 userId String? @db.Uuid  
 workflow String  
 action String?  
 level LogLevel @default(INFO)  
 message String?  
 details Json?  
 createdAt DateTime @default(now())  
 user User? @relation(fields: [userId], references: [id])  
 @@index([createdAt])  
 @@index([workflow, level])  
}  
  
model ApiUsage {  
 id BigInt @id @default(autoincrement())  
 service ApiService  
 endpoint String  
 count Int @default(0)  
 periodStart DateTime  
 periodEnd DateTime  
 meta Json?  
 @@unique([service, endpoint, periodStart, periodEnd])  
}  
  
model ErrorState {  
 id BigInt @id @default(autoincrement())  
 workflow String  
 node String?  
 message String  
 payload Json?  
 createdAt DateTime @default(now())  
 @@index([createdAt])  
}