

Supabase Connection String - Quick Setup Guide

Get Your Connection String in 2 Minutes

Step 1: Open Supabase Dashboard

1. Go to app.supabase.com (<https://app.supabase.com>)
2. Sign in with your account
3. Select your project: `rwdnixmjqpigpmelvfa` (or your project name)

Step 2: Navigate to Database Settings

1. Click **Settings** () icon in the left sidebar
2. Click **Database** in the settings menu
3. Scroll down to the “**Connection string**” section

Step 3: Copy Connection String

You'll see two tabs:

Option A: Transaction Mode (Recommended for Development)

1. Select the “**URI**” tab (not “Connection pooling”)
2. You'll see:
`postgresql://postgres:[YOUR-PASSWORD]@db.rwdnixmjqpigpmelvfa.supabase.co:5432/postgres`
3. Replace `[YOUR-PASSWORD]` with: `artya123`
4. Add at the end: `?sslmode=require`

Final connection string:

```
DATABASE_URL="postgresql://postgres:artya123@db.rwdnixmjqpigpmelvfa.supabase.co:5432/postgres?sslmode=require"
```

Option B: Session Mode (Connection Pooling for Production)

1. Click “**Connection pooling**” toggle
2. Select “**Session**” mode
3. You'll see:
`postgresql://postgres.rwdnixmjqpigpmelvfa:[YOUR-PASSWORD]@aws-0-us-east-1.pooler.supabase.com:5432/postgres`
4. Replace `[YOUR-PASSWORD]` with: `artya123`
5. Add at the end: `?pgbouncer=true&connection_limit=1`

Final connection string:

```
DATABASE_URL="postgresql://postgres.rwdnixmjqpigpmelvfa:artya123@aws-0-us-east-1.pooler.supabase.com:5432/postgres?pgbouncer=true&connection_limit=1"
```

Which One Should I Use?

Use Transaction Mode (Option A) if:

-  You're developing locally
-  You're using Prisma migrations
-  You need full PostgreSQL feature support
-  You're running scripts/seed data

Use Session Mode (Option B) if:

-  You're deploying to Vercel/Netlify
-  You need connection pooling
-  You expect high traffic
-  You want to avoid connection limits

For ArchitectPro: Start with **Option A** (Transaction Mode)



Update .env File

1. Open `/home/ubuntu/architect_pro/nextjs_space/.env`
2. Replace the `DATABASE_URL` line with the connection string from above
3. Save the file

Example .env:

```
# Supabase Database Configuration
DATABASE_URL="postgresql://postgres:artya123@db.rwdnixmjqpigpmelvfa.supabase.co:5432/postgres?sslmode=require"

# AI API Keys
OPENAI_API_KEY="sk-proj-...."
DEEPSEEK_API_KEY="sk-...."
GROQ_API_KEY="gsk_...."
GEMINI_API_KEY="AIza...."
```



Test the Connection

Step 1: Generate Prisma Client

```
cd /home/ubuntu/architect_pro/nextjs_space
npx prisma generate
```

Expected output:

```
✓ Generated Prisma Client (v6.7.0)
```

Step 2: Push Database Schema

```
npx prisma db push
```

Expected output:

```
Datasource "db": PostgreSQL database "postgres"
🎉 Your database is now in sync with your Prisma schema.
✓ Generated Prisma Client
```

If successful, you'll see:

- No connection errors
- Tables created in Supabase
- Schema synced

Step 3: Verify in Supabase Dashboard

1. Go back to Supabase dashboard
2. Click **Table Editor** in left sidebar
3. You should see new tables:
 - Template
 - Design
 - Configuration
 - UserPreference

Step 4: Seed Database (Optional)

```
npx tsx --require dotenv/config scripts/seed_new.ts
```

Expected output:

```
✓ Seeded 2 templates
✓ Database seeded successfully!
```

Troubleshooting

Error: “Can’t reach database server”

Cause: Incorrect host or connection blocked

Solutions:

1. Verify the host exactly matches your Supabase project
2. Check if Supabase project is paused (free tier pauses after 7 days inactivity)
3. Go to Supabase dashboard → Settings → Database → Click “**Resume project**”
4. Check if you’re behind a firewall/VPN that blocks Supabase

Error: “Authentication failed”

Cause: Wrong password

Solutions:

1. Verify password is exactly: artya123 (case-sensitive)
2. No extra spaces before/after password
3. If forgotten, reset password:
 - Go to Supabase → Settings → Database
 - Scroll to “**Database password**”
 - Click “**Generate new password**”
 - Copy and update `.env`

Error: “Tenant or user not found”

Cause: Wrong project reference in connection string

Solutions:

1. Get the exact connection string from Supabase dashboard
2. Verify project reference matches your project
3. Don’t manually construct the URL - copy from dashboard

Error: “SSL connection required”

Cause: Missing SSL mode

Solution:

```
# Add this at the end of connection string:  
?sslmode=require
```

**Screenshot Reference**

Based on your Supabase dashboard screenshots:

From “Connect to your project” page:**Direct Connection String:**

```
Host: db.rwdnixmjqpigpmelvfa.supabase.co
Port: 5432
Database: postgres
User: postgres
Password: [your-password]
```

Formatted for Prisma:

```
postgresql://postgres:artya123@db.rwdnixmjqpigpmelvfa.supabase.co:5432/postgres?sslmode=require
```

Security Note

IMPORTANT:

- .env is in .gitignore - not pushed to GitHub
- Never share .env file publicly
- Use different passwords for dev/production
- Rotate passwords regularly

Next Steps After Connection Works

1. Run npx prisma db push to create tables
2. Run seed script to populate templates
3. Start dev server: yarn dev
4. Test floor plan designer at localhost:3000/designer
5. Verify templates load correctly

Still Having Issues?

If connection still fails after trying all solutions:

1. **Screenshot your error:** Capture the exact error message
2. **Check Supabase status:** status.supabase.com (<https://status.supabase.com>)
3. **Verify project details:**
 - Go to Settings → General
 - Confirm project reference
 - Check if project is active
4. **Use Supabase support:** supabase.com/support (<https://supabase.com/support>)

Quick Copy-Paste Commands

Once you have the correct DATABASE_URL in .env :

```
# Navigate to project
cd /home/ubuntu/architect_pro/nextjs_space

# Generate Prisma Client
npx prisma generate

# Create database tables
npx prisma db push

# Seed with templates
npx tsx --require dotenv/config scripts/seed_new.ts

# Start development server
yarn dev
```

🎉 Once connected, you're ready to use ArchitectPro with Supabase!