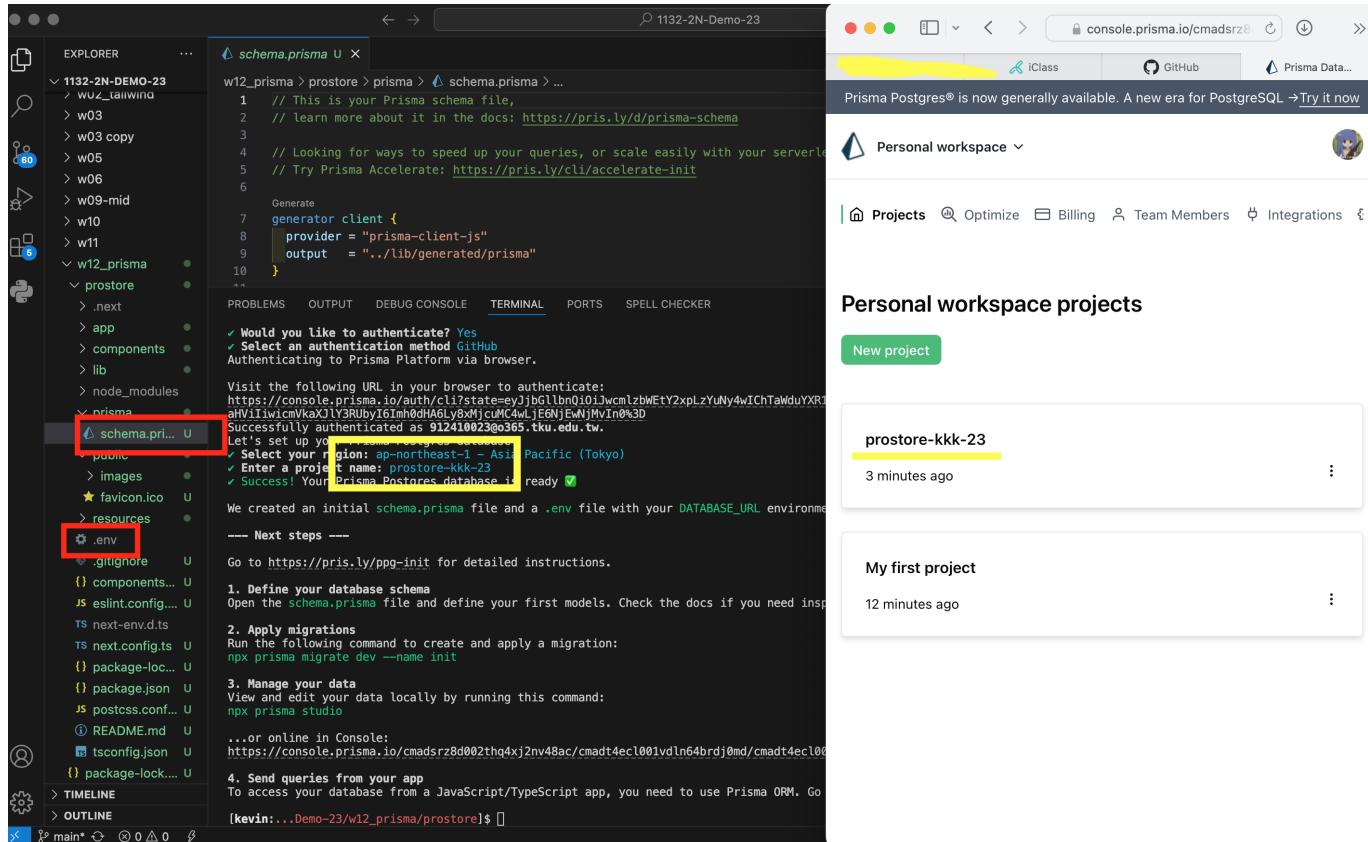


url

vercel

url

W12-p1: npx prisma init --db



W12-p1-1: migration

The screenshot shows the Prisma Data Platform Studio interface. On the left, the Explorer sidebar displays the project structure, including files like `w12.md`, `schema.prisma`, and `migration.sql`. The main area shows the `schema.prisma` file with a yellow box highlighting the `model Product_test` definition:

```

12 datasource db {
13   provider = "postgresql"
14   url      = env("DATABASE_URL")
15 }
16
17 model Product_test {
18   id      String @id @default(dbgenerated("gen_random_uuid"))
19   name    String
20 }

```

The terminal at the bottom shows the command line output for running migrations:

```

[kevin:~/Downloads/1132-2N-Demo-23] cd w12_prisma
[kevin:.../1132-2N-Demo-23/w12_prisma]$ cd prostore
[kevin:.../1132-2N-Demo-23/w12_prisma/prostore]$ npx prisma migrate dev --name init
Environment variables loaded from .env
Prisma schema loaded from prisma/schema.prisma
Datasource "db": PostgreSQL database "postgres", schema "public" at "accelerate.prisma-data.net"
Applying migration `20250507110347_init`

The following migration(s) have been created and applied from new schema changes:
migrations/
└── 20250507110347_init/
    └── migration.sql

Your database is now in sync with your schema.

✓ Generated Prisma Client (v6.7.0, engine=none) to ./lib/generated/prisma

```

The screenshot shows the Prisma Data Platform Studio interface. The `migration.sql` file is open in the editor, with a yellow box highlighting the SQL code for creating the `Product_test` table:

```

1 -- CreateTable
2 CREATE TABLE "Product_test" (
3   "id" UUID NOT NULL DEFAULT gen_random_uuid(),
4   "name" TEXT NOT NULL,
5
6   CONSTRAINT "Product_test_pkey" PRIMARY KEY ("id")
7 );
8

```

The terminal at the bottom shows the command line output for running migrations:

```

[kevin:~/Downloads/1132-2N-Demo-23] cd w12_prisma
[kevin:.../1132-2N-Demo-23/w12_prisma]$ cd prostore
[kevin:.../1132-2N-Demo-23/w12_prisma/prostore]$ npx prisma migrate dev --name init
Environment variables loaded from .env
Prisma schema loaded from prisma/schema.prisma
Datasource "db": PostgreSQL database "postgres", schema "public" at "accelerate.prisma-data.net"
Applying migration `20250507110347_init`

The following migration(s) have been created and applied from new schema changes:
migrations/
└── 20250507110347_init/
    └── migration.sql

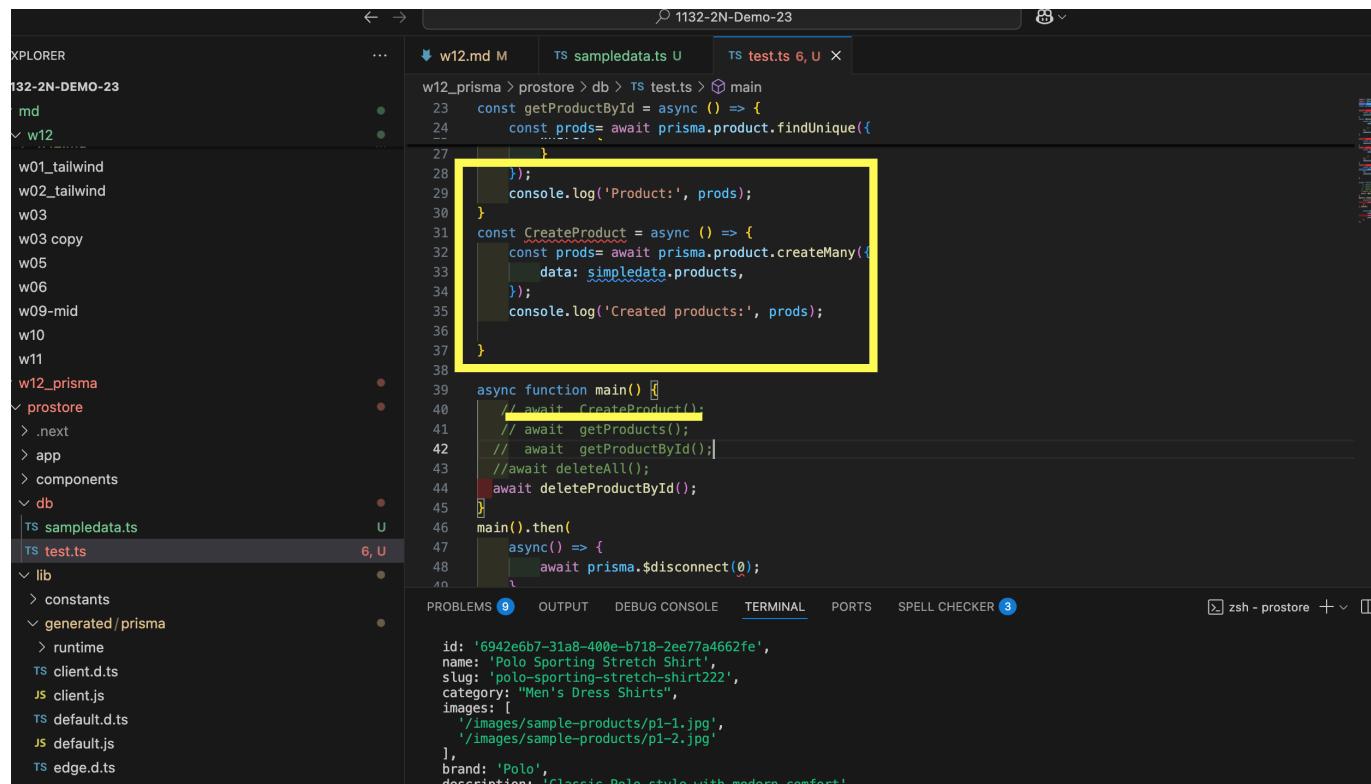
Your database is now in sync with your schema.

✓ Generated Prisma Client (v6.7.0, engine=none) to ./lib/generated/prisma in 76ms

```

W12-P2: Do insert, get, delete operations on Product table

=> create data from sample-data.ts



```
w12_prisma > prostore > db > TS test.ts > main
  23  const getProductById = async () => {
  24    const prods= await prisma.product.findUnique({
  25      ...
  26    });
  27    console.log('Product:', prods);
  28  }
  29  const CreateProduct = async () => {
  30    const prods= await prisma.product.createMany({
  31      data: simpledata.products,
  32    });
  33    console.log('Created products:', prods);
  34  }
  35
  36
  37
  38
  39  async function main() {
  40    // await CreateProduct();
  41    // await getProducts();
  42    // await getProductById();
  43    //await deleteAll();
  44    await deleteProductById();
  45  }
  46
  47  main().then(
  48    async() => {
  49      await prisma.$disconnect();
  50    }
  51  )
}
id: '6942e6b7-31a8-400e-b718-2ee77a4662fe',
name: 'Polo Sporting Stretch Shirt',
slug: 'polo-sporting-stretch-shirt222',
category: "Men's Dress Shirts",
images: [
  '/images/sample-products/p1-1.jpg',
  '/images/sample-products/p1-2.jpg'
],
brand: 'Polo',
description: 'Classic Polo style with modern comfort',
```

=> get all products from Product table

The screenshot shows a dual-pane interface. On the left is a code editor in VS Code with a file named `w12.md`. The code contains TypeScript and Prisma queries. On the right is the Prisma Data Platform Studio, which displays a table named `Product` with columns `id`, `banner`, and `text`. An "Insert row" dialog is open, showing a single row being added.

```

const getProductId = async () => {
  const prods = await prisma.product.findUnique({
    where: {
      id: '6942e6b7-31a8-400e-b718-2ee77a4662fe'
    }
  });
  console.log('Product:', prods);
}

const CreateProduct = async () => {
  const prods = await prisma.product.createMany({
    data: simpledata.products,
  });
  console.log('Created products:', prods);
}

async function main() {
  // await CreateProduct();
  // await getProducts();
  // await getProductId();
  // await deleteAll();
  // await deleteProductById();
}

main().then(
  async() => {
    await prisma.$disconnect();
  }
).catch(
  async(e) => {
    console.error('Error in main function:', e);
    await prisma.$disconnect();
    process.exit(1);
  }
)

```

This screenshot is identical to the one above, showing the same code in VS Code and the same table in the Prisma Data Platform Studio. The "Insert row" dialog is still open, and the table data is identical to the previous screenshot.

	<code>id</code>	<code>banner</code>	<code>text</code>
<input type="checkbox"/>	6942e6b7-31a8-400e-b718-2ee77a4662fe	bCCCCCCCCCanner-1.jpg	
<input type="checkbox"/>	fd499f60-0854-4db7-b8a2...	banner-1.jpg	
<input type="checkbox"/>	7a84dfaf-6db9-4bd7-af47...	banner-2.jpg	
<input type="checkbox"/>	6ec76fab-b178-40e5-95e2...	NULL	
<input type="checkbox"/>	30503843-3e1c-426c-8b74...	NULL	
<input type="checkbox"/>	90dbe06f-caac-4945-996c...	NULL	
<input type="checkbox"/>	07c59cb4-6d29-41b1-a1c3...	NULL	

=> delete all products from Product table

```

w12_prisma > pristore > db > TS test.ts > [o] getProductsById
6 const getProductsById = async () => {
7   console.log('Products:', prods);
8
9   const prods = await prisma.product.findUnique({
10     where: {
11       id: 'c61fff00-af0f-4c00-94b5-70345c7de0c6'
12     }
13   });
14
15   console.log('Product:', prods);
16
17 }
18 const CreateProduct = async () => {
19   const prods = await prisma.product.createMany({
20     data: simpledata.products,
21   });
22   console.log('Created products:', prods);
23
24 }
25
26 async function main() {
27   // await CreateProduct();
28
29   const product = await prisma.product.create({
30     data: {
31       id: 'c61fff00-af0f-4c00-94b5-70345c7de0c6',
32       name: 'Polo Sporting Stretch Shirt',
33       category: "Men's Dress Shirts",
34       images: [
35         '/images/sample-products/p1-1.jpg',
36         '/images/sample-products/p1-2.jpg'
37       ],
38       brand: 'Polo',
39       description: 'Classic Polo style with modern comfort',
40       stock: 5,
41       price: 59.99,
42       rating: 4.5,
43       numReviews: 10,
44       isFeatured: true,
45       banner: 'banner-1.jpg',
46       createdAt: 2025-05-07T12:01:53.791Z
47     }
48   });
49
50   await getProductsById();
51
52   main().then(
53     async () => {
54       await prisma.$disconnect();
55     }
56   ).catch(
57     error => {
58       console.error(error);
59     }
60   );
61
62 }
63
64 main().catch((error) => {
65   console.error(error);
66 });

```

Studio

Edit data in your Prisma Postgres® instance without leaving the console.

[Feedback](#)

Tables

	id	banner
<input type="checkbox"/>	b61fff00-af0f-4c00-94b5-70345c7de0c6	banner-1.jpg
<input type="checkbox"/>	7f325736-10b9-4b22-952b...	NULL
<input type="checkbox"/>	0a2b5c9d-3c47-4277-ba54...	NULL
<input type="checkbox"/>	e27ed01b-b5ce-4c6a-8f47...	NULL

```

w12_prisma > pristore > db > TS test.ts > [o] getProductsById
10 const getProductsById = async () => {
11
12   const CreateProduct = async () => {
13     const prods = await prisma.product.createMany({
14       data: simpledata.products,
15     });
16     console.log('Created products:', prods);
17
18   }
19
20   sync function main() {
21     // await CreateProduct();
22
23     await getProductsById();
24
25   main().then(
26     async () => {
27       await prisma.$disconnect();
28     }
29   ).catch(
30     error => {
31       console.error(error);
32     }
33   );
34
35 }
36
37 main().catch((error) => {
38   console.error(error);
39 });

```

Personal workspace / pristore-kkk-23 / Development

Dashboard Studio Accelerate Database API Keys Settings

Studio

Edit data in your Prisma Postgres® instance without leaving the console.

[Feedback](#)

Tables

	id	banner
<input type="checkbox"/>	b61fff00-af0f-4c00-94b5-70345c7de0c6	banner-1.jpg
<input type="checkbox"/>	7f325736-10b9-4b22-952b...	NULL
<input type="checkbox"/>	0a2b5c9d-3c47-4277-ba54...	NULL
<input type="checkbox"/>	e27ed01b-b5ce-4c6a-8f47...	NULL

=> create data from sample-data.ts again

The screenshot shows the Prisma Data Platform Studio interface. On the left, the Explorer sidebar displays the project structure for "w12_prisma". The "test.ts" file is open in the code editor, showing TypeScript code for a Prisma client. The code includes imports for `prisma` and `sampledata.ts`, and defines an `async` function `main` that performs various database operations like creating products and deleting them.

The main workspace shows the "Studio" tab active, with the heading "Edit data in your Prisma Postgres® instance without leaving the console." Below this is a "Feedback" section. To the right, there's a "Tables" section with a table for "Product" containing several rows of product data, and an "Insert row" button.

	id	banner_text
<input type="checkbox"/>	fd499f60-0854-4db7-b8a2...	banner-1.jpg
<input type="checkbox"/>	7a84dfa...-4bd7-af47...	banner-2.jpg
<input type="checkbox"/>	6ec76fab-b178-40e5-95e2...	NULL
<input type="checkbox"/>	30503843-3e1c-426c-8b74...	NULL
<input type="checkbox"/>	90dbe06f-caac-4945-996c...	NULL
<input type="checkbox"/>	07c59cb4-6d29-41b1-a1c3...	NULL

W12-P3:

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows the project structure. The current file is `test.ts`. Other files like `w12_prisma`, `sampledata.ts`, and `test.ts` are also listed.
- Code Editor:** Displays the following TypeScript code:

```
1 import {PrismaClient} from '@lib/generated/prisma';
2 import sampledata from './sampledata';
3 const prisma = new PrismaClient().$extends(withAccelerate());
4
5 const getProductsByPlainSQL = () => {
6   const prods = await prisma.$queryRaw`SELECT * FROM "Product"`;
7   console.log('Products:', prods);
8 }
9 }
```

A yellow box highlights the `$queryRaw`SELECT * FROM "Product"`` line.
- Terminal:** Shows the output of running `npx tsx ./db/test.ts`. The output is:

```
[kevin..._Demo-23/w12_prisma/prostore]$ npx tsx ./db/test.ts
Products: [
  {
    id: 'fd499f60-0854-4db7-b8a2-9ebba21cad6c',
    name: 'Polo Sporting Stretch Shirt',
    slug: 'polo-sporting-stretch-shirt',
    category: "Men's Dress Shirts",
    images: [
      '/images/sample-products/p1-1.jpg',
      '/images/sample-products/p1-2.jpg'
    ],
    brand: 'Polo',
    description: 'Classic Polo style with modern comfort',
    stock: 5,
    price: 59.99,
    rating: 4.5,
    numReviews: 10,
    isFeatured: true,
    banner: 'banner-1.jpg',
    createdAt: 2025-05-07T12:21:19.409Z
  },
  {
    id: '7a84dfaf-6db9-4bd7-af47-5c69e99329a9',
    name: 'Brooks Brothers Long Sleeved Shirt',
    slug: 'brooks-brothers-long-sleeved-shirt',
    category: "Men's Dress Shirts",
    images: [
      '/images/sample-products/p2-1.jpg',
      '/images/sample-products/p2-2.jpg'
    ],
    brand: 'Brooks Brothers',
    description: 'Timeless style and premium comfort',
    stock: 10,
    price: 85.9,
    rating: 4.2,
    numReviews: 8,
    isFeatured: true,
    banner: 'banner-2.jpg',
  }
]
```
- Bottom Status Bar:** Shows the current file is `Ln 23, Col 1`, with `Spaces: 4`, `UTF-8`, `LF`, `TypeScript`, `Go Live`, and `Prettier` icons.

```

1132-2N-DEMO-23
  md ...
    > w01_tailwind
    > w02_tailwind
    > w03
    > w03 copy
    > w05
    > w06
    > w09-mid
    > w10
    > w11
    > w12_prisma ...
      > .next
      > app
      > components
      > db ...
        TS sampledata... U
        TS test.ts 9, U
          lib ...
            > constants
            > generated...
              > runtime
              TS client.dts
              JS client.js
              TS default.d.ts
              JS default.js
              TS edge.dts
              JS edge.js M
              JS index-bro... M
              TS index.dts M
              JS index.js M
              {} package.j... M
        > TIMELINE
        > OUTLINE
main* ⌂ 0 down 1 up ⌂ 9 ⌂ 0 ⌂ 3 ⌂
  
```

```

1132-2N-DEMO-23
  md ...
    > w01_tailwind
    > w02_tailwind
    > w03
    > w03 copy
    > w05
    > w06
    > w09-mid
    > w10
    > w11
    > w12_prisma ...
      > .next
      > app
      > components
      > db ...
        TS sampledata... U
        TS test.ts 9, U
          lib ...
            > constants
            > generated...
              > runtime
              TS client.dts
              JS client.js
              TS default.d.ts
              JS default.js
              TS edge.dts
              JS edge.js M
              JS index-bro... M
              TS index.dts M
              JS index.js M
              {} package.j... M
        > TIMELINE
        > OUTLINE
main* ⌂ 0 down 1 up ⌂ 9 ⌂ 0 ⌂ 3 ⌂
  
```

w12_prisma > pristore > db > TS test.ts > main

```

9 }
10
11 const getProductsByPlainSQL2 = async (brand:any) => {
12   const prods = await prisma.$queryRaw`SELECT * FROM "Product"
13
14   console.log('Products:', prods);
15 }
  
```

slug: 'polo-sporting-stretch-shirt',
category: 'Men's Dress Shirts',
images: [
 '/images/sample-products/p1-1.jpg',
 '/images/sample-products/p1-2.jpg'
],
brand: 'Polo',
description: 'Classic Polo style with modern comfort',
stock: 5,
price: 59.99,
rating: 4.5,
numReviews: 10,
isFeatured: true,
banner: 'banner-1.jpg',
createdAt: 2025-05-07T12:21:19.409Z },
,
 id: '00dbe06f-caac-4945-996c-45d8e5be74b9',
 name: 'Polo Ralph Lauren Oxford Shirt',
 slug: 'polo-ralph-lauren-oxford-shirt',
 category: 'Men's Dress Shirts',
 images: [
 '/images/sample-products/p5-1.jpg',
 '/images/sample-products/p5-2.jpg'
],
 brand: 'Polo',
 description: 'Iconic Polo design with refined oxford fabric',
 stock: 6,
 price: 79.99,
 rating: 4.7,
 numReviews: 18,
 isFeatured: false,
 banner: null,
 createdAt: 2025-05-07T12:21:19.409Z },
,
 id: '07c59cb4-6d29-41b1-a1c3-cd807759e015',
 name: 'Polo Classic Pink Hoodie',
 slug: 'polo-classic-pink-hoodie',
 category: 'Men's Sweatshirts',
 images: [
 '/images/sample-products/p6-1.jpg',

```

1132-2N-DEMO-23
  md ...
    > w01_tailwind
    > w02_tailwind
    > w03
    > w03 copy
    > w05
    > w06
    > w09-mid
    > w10
    > w11
    > w12_prisma ...
      > .next
      > app
      > components
      > db ...
        TS sampledata... U
        TS test.ts 9+, U
          lib ...
            > constants
            > generated...
              > runtime
              TS client.dts
              JS client.js
              TS default.d.ts
              JS default.js
              TS edge.dts
              JS edge.js M
              JS index-bro... M
              TS index.dts M
              JS index.js M
              {} package.j... M
        > TIMELINE
        > OUTLINE
main* ⌂ 0 down 1 up ⌂ 9 ⌂ 0 ⌂ 3 ⌂
  
```

```

1132-2N-DEMO-23
  md ...
    > w01_tailwind
    > w02_tailwind
    > w03
    > w03 copy
    > w05
    > w06
    > w09-mid
    > w10
    > w11
    > w12_prisma ...
      > .next
      > app
      > components
      > db ...
        TS sampledata... U
        TS test.ts 9+, U
          lib ...
            > constants
            > generated...
              > runtime
              TS client.dts
              JS client.js
              TS default.d.ts
              JS default.js
              TS edge.dts
              JS edge.js M
              JS index-bro... M
              TS index.dts M
              JS index.js M
              {} package.j... M
        > TIMELINE
        > OUTLINE
main* ⌂ 0 down 1 up ⌂ 9 ⌂ 0 ⌂ 3 ⌂
  
```

w12_prisma > pristore > db > TS test.ts > main

```

6   const getProductsByPlainSQL1 = async () => {
7     console.log('Products:', prods);
8   }
9 }
10
11 const getProductsByPlainSQL2 = async (brand:any) => {
12   const prods = await prisma.$queryRaw`SELECT * FROM "Product"
13
14   console.log('Products:', prods);
15 }
16
17 const getProductsByPlainSQL3 = async (brand:any) => {
18   let query=Prisma.sql`SELECT * FROM "Product" WHERE "brand" = ${brand}`;
19
20   const prods = await prisma.$queryRaw(query);
21
22   console.log('Products:', prods);
23 }
24
25 const deleteAll = async () => {
  
```

'/images/sample-products/p1-1.jpg',
'/images/sample-products/p1-2.jpg'
,
 id: '00dbe06f-caac-4945-996c-45d8e5be74b9',
 name: 'Polo Ralph Lauren Oxford Shirt',
 slug: 'polo-ralph-lauren-oxford-shirt',
 category: 'Men's Dress Shirts',
 images: [
 '/images/sample-products/p5-1.jpg',
 '/images/sample-products/p5-2.jpg'
],
 brand: 'Polo',
 description: 'Iconic Polo design with refined oxford fabric',
 stock: 6,
 price: 79.99,
 rating: 4.7,
 numReviews: 18,

git log

97fa24b 05235	Wed May 7 21:00:11 2025 +0800	week12 p3
0b4a4f2 05235	Wed May 7 19:09:41 2025 +0800	week12 p1-p2