

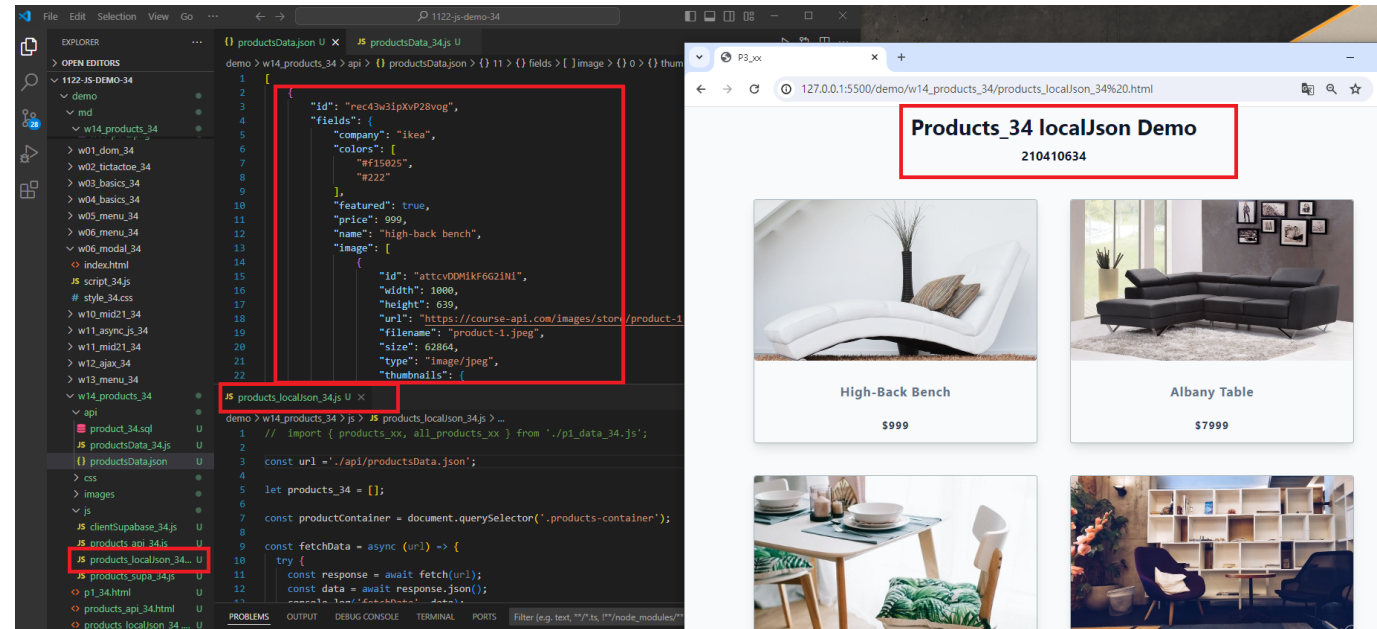
W14-P1: Do products_api_xx.js to get data from an api url

The screenshot shows a web browser displaying a product catalog titled "Products_34 Demo" with ID "210410634". The catalog lists six products: High-Back Bench (\$999), Albany Table (\$7999), Accent Chair (\$2599), Wooden Table (\$4599), Dining Table (\$6999), and Sofa Set (\$6999). Each product has a corresponding image. To the right, the Chrome DevTools console is open, showing the response from the API endpoint `products_api_34.js`. The response is a JSON array of product objects. A red box highlights the first product object in the array, which matches the "High-Back Bench" product shown in the catalog. The product object includes fields like `colors`, `company`, `featured`, `image`, `filename`, `height`, `id`, `size`, `thumbnails`, `width`, `length`, `name`, and `price`.

The screenshot shows a code editor with two files open: `products_api_34.js` and `products_localJson_34.js`. The `products_api_34.js` file contains the following code:

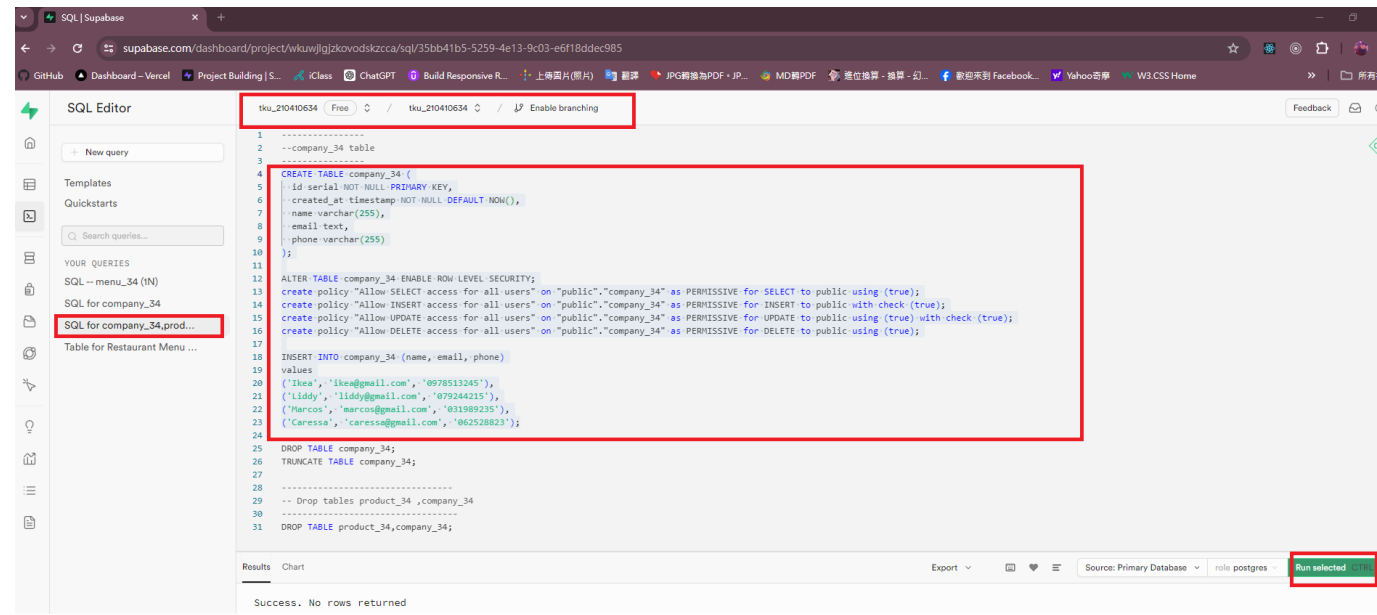
```
demo > w14_products_34 > js > products_api_34.js > ...
1 // Import ( products_xx, all_products_xx ) from './p1_data_34.js';
2
3 const url = 'https://www.course-api.com/javascript-store-products';
4
5 let products_34 = [];
6
7 const productContainer = document.querySelector('.products-container');
8
9 const fetchData = async (url) => {
10   try {
11     const response = await fetch(url);
12     const data = await response.json();
13     console.log('fetchData', data);
14     return data;
15   } catch (error) {
16     console.log(error);
17   }
18 };
19
20 console.log('products_34', products_34);
21
22 const displayProducts = (products) => {
23   let productsContent = products
24   .map((product) => {
25     const {name, price, image} = product.fields;
26     return `
27     <div class="single-product">
28       
32       <div>
33         <h3 class="name">${name}</h3>
34         <span class="price">${price}</span>
35       </div>
36     </div>
37   `;
38   });
39   .join('');
40   productContainer.innerHTML = productsContent;
41
42   document.addEventListener('DOMContentLoaded', async () => {
43     products_34 = await fetchData(url);
44     displayProducts(products_34);
45   });
46
47 }
```

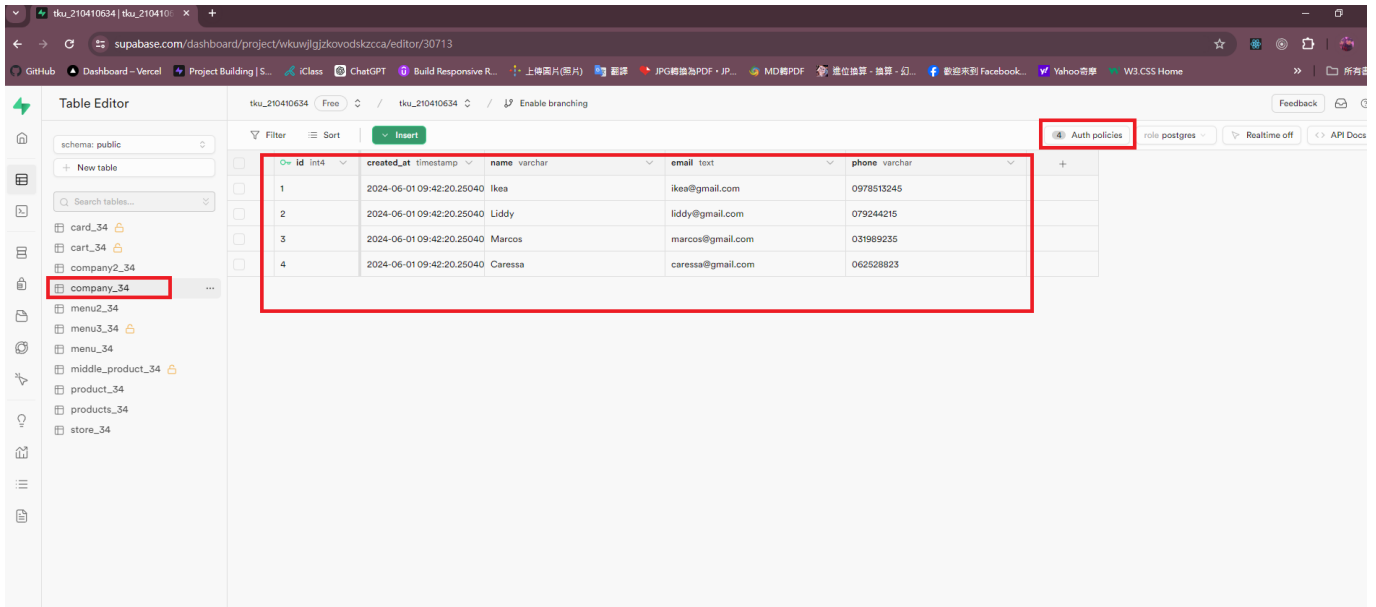
W14-P2: Do products_localJson_xx.js to get local json data (/api/productsData.json)



W14-P3: Use SQL to create schemas and data for company_xx and products_xx

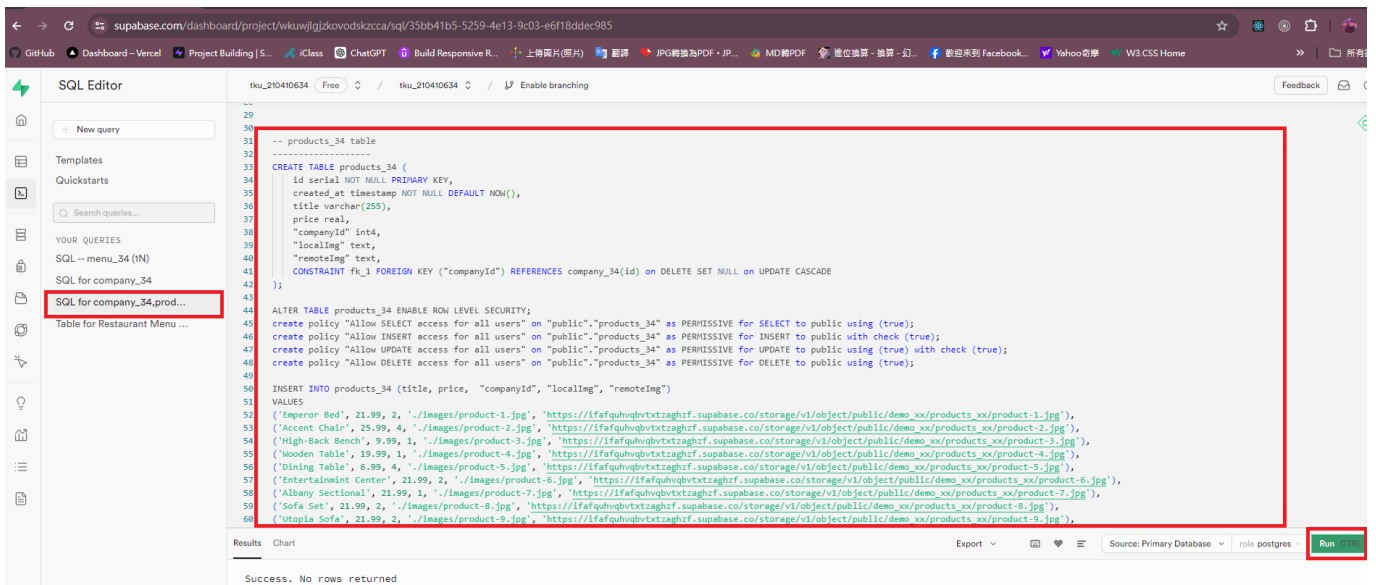
=> company_xx schema and data





id	created_at	name	email	phone
1	2024-06-01 09:42:20.25040	Ikea	ikea@gmail.com	0978513245
2	2024-06-01 09:42:20.25040	Liddy	liddy@gmail.com	079244215
3	2024-06-01 09:42:20.25040	Marcos	marcos@gmail.com	031989235
4	2024-06-01 09:42:20.25040	Caressa	caressa@gmail.com	062528823

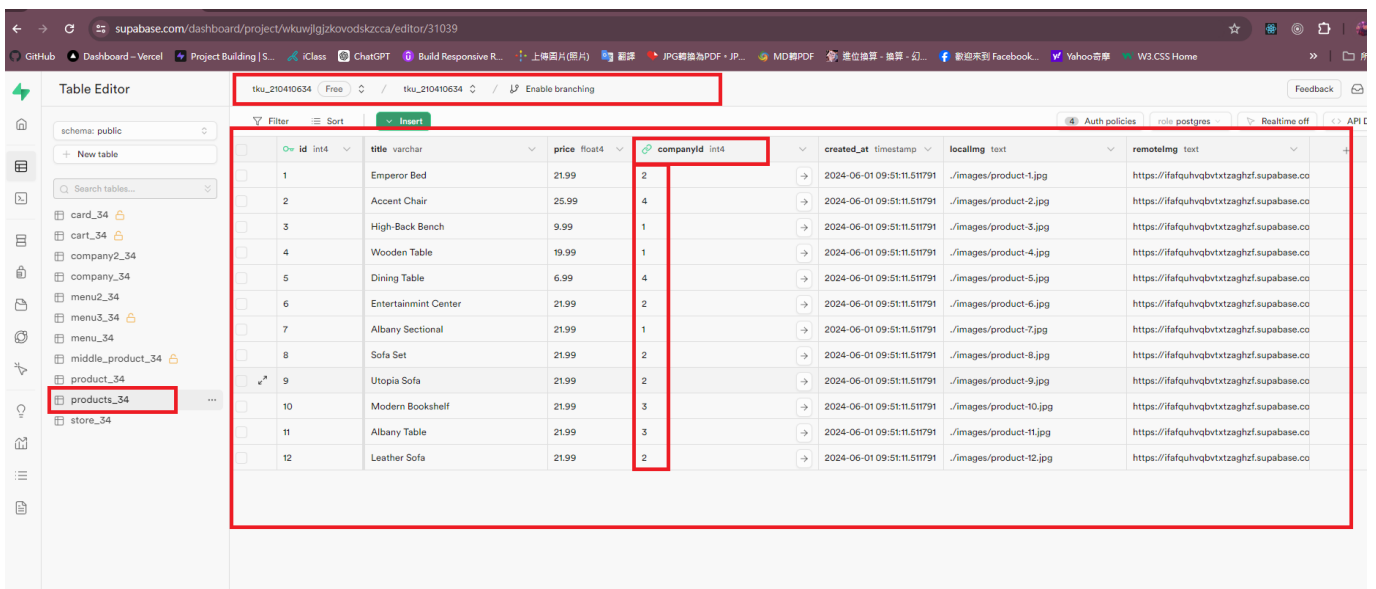
=> products_xx schema and data



```
-- products_34 table
CREATE TABLE products_34 (
  id serial NOT NULL PRIMARY KEY,
  created_at timestamp NOT NULL DEFAULT NOW(),
  title varchar(255),
  price real,
  "companyId" int4,
  "localing" text,
  "remotelm" text,
  CONSTRAINT fk_1 FOREIGN KEY ("companyId") REFERENCES company_34(id) ON DELETE SET NULL ON UPDATE CASCADE
);

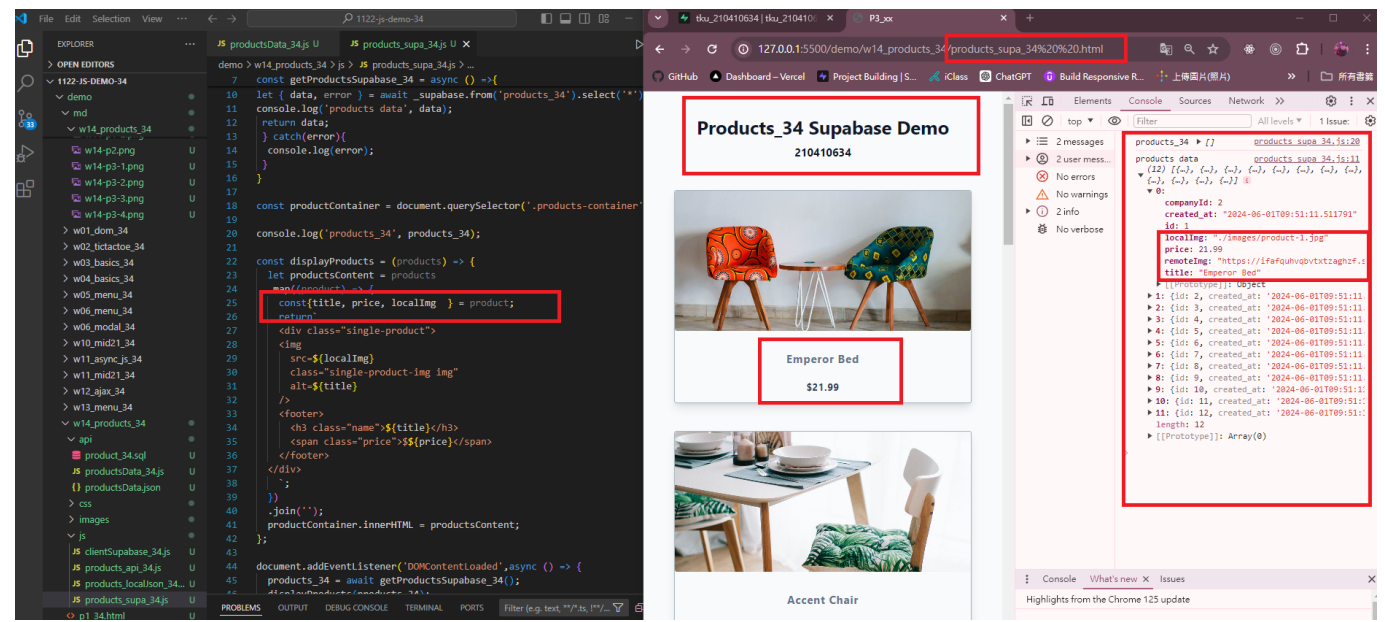
ALTER TABLE products_34 ENABLE ROW LEVEL SECURITY;
create policy "Allow SELECT access for all users" on "public"."products_34" as PERMISSIVE for SELECT to public using (true);
create policy "Allow INSERT access for all users" on "public"."products_34" as PERMISSIVE for INSERT to public with check (true);
create policy "Allow UPDATE access for all users" on "public"."products_34" as PERMISSIVE for UPDATE to public with check (true);
create policy "Allow DELETE access for all users" on "public"."products_34" as PERMISSIVE for DELETE to public using (true);

INSERT INTO products_34 (title, price, "companyId", "localing", "remotelm")
VALUES
('Emperor Bed', 21.99, 2, './images/product-1.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-1.jpg'),
('Accent Chair', 25.99, 4, './images/product-2.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-2.jpg'),
('High-Back Bench', 9.99, 1, './images/product-3.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-3.jpg'),
('Wooden Table', 19.99, 1, './images/product-4.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-4.jpg'),
('Dining Table', 6.99, 4, './images/product-5.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-5.jpg'),
('Entertainment Center', 21.99, 2, './images/product-6.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-6.jpg'),
('Albany Sectional', 21.99, 1, './images/product-7.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-7.jpg'),
('Sofa Set', 21.99, 2, './images/product-8.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-8.jpg'),
('Utopia Sofa', 21.99, 2, './images/product-9.jpg', 'https://ifafquhvbtvtzaghf.supabase.co/storage/v1/object/public/demo_xx/products_xx/product-9.jpg');
```



id	title	price	companyId	created_at	localing	remotelm
1	Emperor Bed	21.99	2	2024-06-01 09:51:11.51791	./images/product-1.jpg	https://ifafquhvbtvtzaghf.supabase.co
2	Accent Chair	25.99	4	2024-06-01 09:51:11.51791	./images/product-2.jpg	https://ifafquhvbtvtzaghf.supabase.co
3	High-Back Bench	9.99	1	2024-06-01 09:51:11.51791	./images/product-3.jpg	https://ifafquhvbtvtzaghf.supabase.co
4	Wooden Table	19.99	1	2024-06-01 09:51:11.51791	./images/product-4.jpg	https://ifafquhvbtvtzaghf.supabase.co
5	Dining Table	6.99	4	2024-06-01 09:51:11.51791	./images/product-5.jpg	https://ifafquhvbtvtzaghf.supabase.co
6	Entertainment Center	21.99	2	2024-06-01 09:51:11.51791	./images/product-6.jpg	https://ifafquhvbtvtzaghf.supabase.co
7	Albany Sectional	21.99	1	2024-06-01 09:51:11.51791	./images/product-7.jpg	https://ifafquhvbtvtzaghf.supabase.co
8	Sofa Set	21.99	2	2024-06-01 09:51:11.51791	./images/product-8.jpg	https://ifafquhvbtvtzaghf.supabase.co
9	Utopia Sofa	21.99	2	2024-06-01 09:51:11.51791	./images/product-9.jpg	https://ifafquhvbtvtzaghf.supabase.co
10	Modern Bookshelf	21.99	3	2024-06-01 09:51:11.51791	./images/product-10.jpg	https://ifafquhvbtvtzaghf.supabase.co
11	Albany Table	21.99	3	2024-06-01 09:51:11.51791	./images/product-11.jpg	https://ifafquhvbtvtzaghf.supabase.co
12	Leather Sofa	21.99	2	2024-06-01 09:51:11.51791	./images/product-12.jpg	https://ifafquhvbtvtzaghf.supabase.co

W14-P4: Do products_supa_xx.js to get data from Supabase



ab9ea0c A-iong Sat Jun 1 23:26:26 2024 +0800 W14-P4: Do products_supa_xx.js to get data from Supabase