

[Github](#)

[Vercel](#)

W5-P1: Create task_xx table in Supabase with 4 data using SQL

The screenshot shows the Supabase Table Editor interface. At the top, the URL is supabase.com/dashboard/project/szbwskgtksqzkhaoKhez/editor/29569. The schema is set to 'public'. The table being edited is 'task_31'. The table has three columns: 'id' (int4), 'title' (varchar), and 'is_done' (bool). The data consists of four rows:

	id	title	is_done
1	1	Do W1 Assigment	FALSE
2	2	Do W2 Assigment	FALSE
3	3	Do W3 Assigment	TRUE
4	4	Do W4 Assigment	FALSE

6a783ea 陳亮瑜 Wed Mar 20 18:39:36 2024 +0800 git log --pretty=format:%h%x09%an%x09%ad%

W5-P2: Use React Query to get tasks data from Supabas

The screenshot shows a code editor with several files open. On the left, the file structure includes a .env file, clientSupabase_31.jsx, .gitignore, and a db folder containing clientSupabase_31.js. The clientSupabase_31.jsx file contains code to create a Supabase client using the supabase-js library. The .env file contains environment variables for the Supabase URL and key.

```
import { createClient } from "@supabase/supabase-js";
export const supabase = createClient(
  import.meta.env.VITE_SUPABASE_URL,
  import.meta.env.VITE_SUPABASE_ANON_KEY
)
```

The terminal at the bottom shows the command `npm i @supabase/supabase-js` being run, and the output indicates 15 packages were added and audited.

Code editor showing two files: App_31.jsx and Items_31.jsx.

App_31.jsx

```

1 import { ToastContainer } from 'react-toastify';
2 import { nanoid } from 'nanoid';
3 import Form_31 from './Form_31';
4 import Items_31 from './Items_31';
5 import { useState } from 'react';
6
7 const defaultItems = [
8   { id: nanoid(), title: 'walk the dog', isDone: false },
9   { id: nanoid(), title: 'wash dishes', isDone: false },
10  { id: nanoid(), title: 'drink coffee', isDone: true },
11  { id: nanoid(), title: 'take a nap', isDone: false },
12];
13
14 const App_31 = () => {
15   const [items, setItems] = useState(defaultItems);
16   return (
17     <section className='section-center'>
18       <ToastContainer position='top-center' />
19       <Form_31 />
20       <Items_31 items={items}/>
21     </section>
22   );
23 }
24
25 export default App_31;

```

Items_31.jsx

```

1 import SingleItem_31 from './SingleItem_31';
2
3 import { useQuery } from '@tanstack/react-query';
4
5 import customFetch_31 from './utils_31';
6
7 const Items_31 = ({ items }) => {
8
9   return (
10     <div className='items'>
11       {items.map((item) => {
12         return <SingleItem_31 key={item.id} item={item} />;
13       })}
14     </div>
15   );
16 }
17
18 export default Items_31;

```

Output in browser: Task Bud -- Htchung

Add Task

- Walk The Dog Delete
- Wash Dishes Delete
- Drink Coffee Delete
- Take A Nap Delete

Code editor showing Items_31.jsx and App_31.jsx.

Items_31.jsx

```

1 import SingleItem_31 from './SingleItem_31';
2
3 import { useQuery } from '@tanstack/react-query';
4
5 import { supabase } from './clientSupabase_31';
6
7 const Items_31 = () => {
8   const [data, isLoading, isError, error] = useQuery([
9     { queryFn: async () => {
10       try {
11         let { data, error } = await supabase.from('task_31').select();
12         console.log('data', data);
13         return data;
14       } catch (error) {
15         console.log(error);
16       }
17     },
18     select: [
19       'id',
20       'title',
21       'isDone'
22     ]
23   ]);
24
25   if (isLoading) {
26     return <p> Loading... </p>;
27   }
28
29   if (error) {
30     return <p> {error.message} </p>;
31   }
32
33   return (
34     <div className='items'>
35       {data.map((item) => {
36         return <SingleItem_31 key={item.id} item={item} />;
37       })}
38     </div>
39   );
40 }
41
42 export default Items_31;

```

App_31.jsx

```

1 import { ToastContainer } from 'react-toastify';
2 import { nanoid } from 'nanoid';
3 import Form_31 from './Form_31';
4 import Items_31 from './Items_31';
5 import { useState } from 'react';
6
7 const defaultItems = [
8   { id: nanoid(), title: 'Do W1 Assigment', isDone: false },
9   { id: nanoid(), title: 'Do W2 Assigment', isDone: false },
10  { id: nanoid(), title: 'Do W3 Assigment', isDone: true },
11  { id: nanoid(), title: 'Do W4 Assigment', isDone: false },
12];
13
14 const App_31 = () => {
15   const [items, setItems] = useState(defaultItems);
16   return (
17     <section className='section-center'>
18       <ToastContainer position='top-center' />
19       <Form_31 />
20       <Items_31 items={items}/>
21     </section>
22   );
23 }
24
25 export default App_31;

```

Output in browser: Task Bud -- Htchung

Add Task

- Do W1 Assigment Delete
- Do W2 Assigment Delete
- Do W3 Assigment Delete
- Do W4 Assigment Delete

Console output:

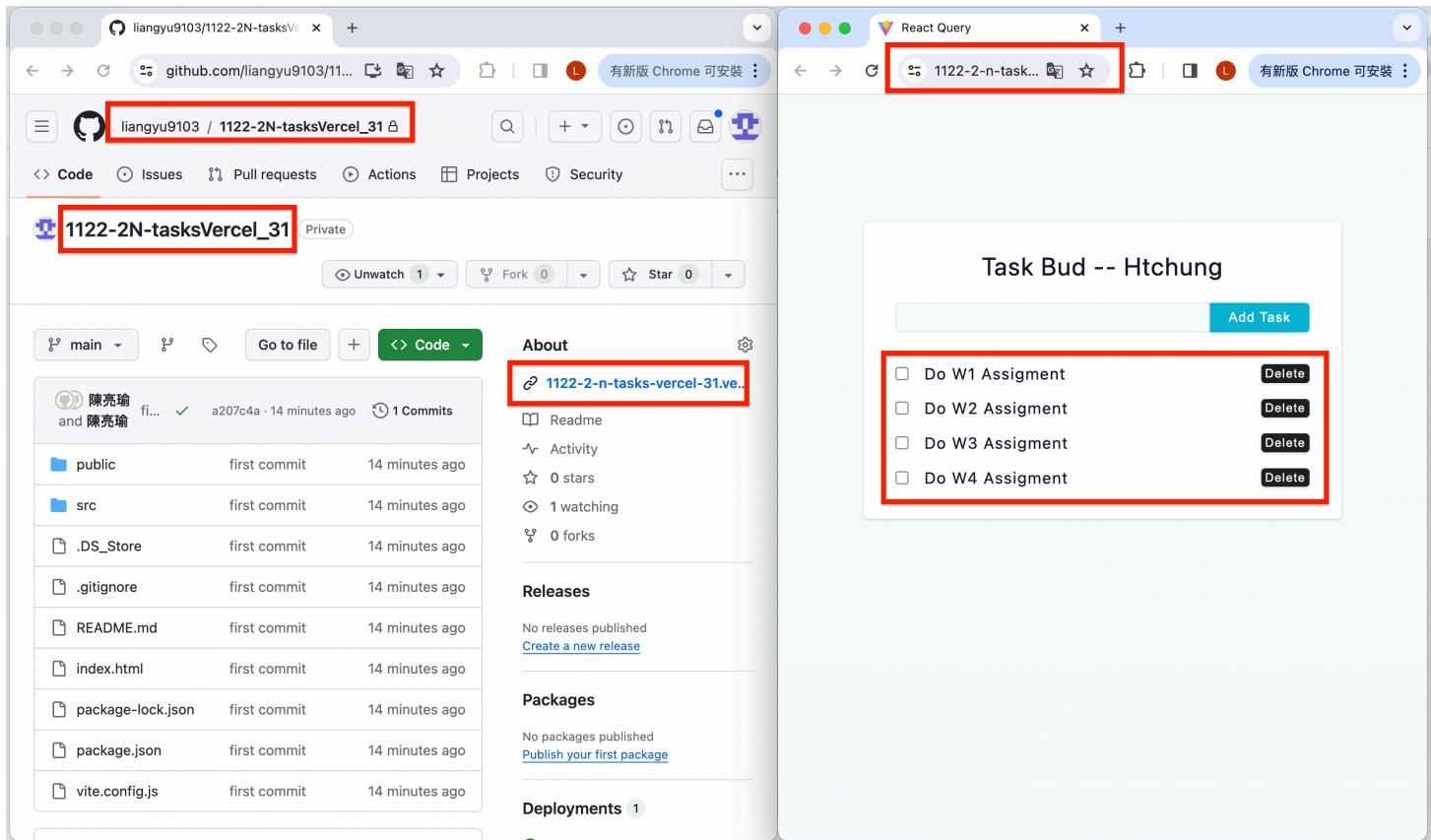
```

data: [
  { id: 0, title: 'Do W1 Assigment', isDone: false },
  { id: 1, title: 'Do W2 Assigment', isDone: false },
  { id: 2, title: 'Do W3 Assigment', isDone: true },
  { id: 3, title: 'Do W4 Assigment', isDone: false }
]
tasks: [
  { id: 0, title: 'Do W1 Assigment', isDone: false },
  { id: 1, title: 'Do W2 Assigment', isDone: false },
  { id: 2, title: 'Do W3 Assigment', isDone: true },
  { id: 3, title: 'Do W4 Assigment', isDone: false }
]

```

8f0c529 陳亮瑜 Wed Mar 20 19:17:25 2024 +0800 ### W5-P2: 設定 supabase client，使使用 .e
44b2c3e 陳亮瑜 Wed Mar 20 20:08:23 2024 +0800 ### W5-P2: Use React Query to get tasks

W5-P3: Deploy W5-P2 to Vercel



6b123a3 陳亮瑜 Wed Mar 20 20:33:39 2024 +0800 ### W5-P3: Deploy W5-P2 to Vercel

W5-P4: Create new Task, deploy W5-P3 to Vercel

=> local

The screenshot displays a development environment with two code editors and a browser window.

Code Editors:

- Left Editor:** Shows the file `Form_31.jsx`. A red box highlights the mutation logic:

```
mutationFn: async (title) => {
  try {
    const { data, error } = await
      .from('task_31')
      .insert([{ title, is_done: false }]);
  } catch (error) {
    console.log(error);
  }
},
```
- Right Editor:** Shows the same file `Form_31.jsx`. A red box highlights the handleSubmit function:

```
const handleSubmit = (e) => {
  e.preventDefault();
  if(newItemName === '') {
    toast.warning('Please Provide title');
  } else {
    createTasks(newItemName)
  }
};
```

Browser:

- Title Bar:** Task Bud -- 208410331
- Form:** An input field and a "Add Task" button.
- List:** A list of tasks with checkboxes and delete buttons. One task is selected, highlighted with a red box:
 - Do W1 Assigment
 - Do W2 Assigment
 - Do W3 Assigment
 - Do W4 Assigment
 - Liang Yu, 208410331
- Console:** Shows the tasks array and its contents. A red box highlights the last item:

```
tasks: [
  { id: 1, title: 'Do W1 Assigment', is_done: false },
  { id: 2, title: 'Do W2 Assigment', is_done: false },
  { id: 3, title: 'Do W3 Assigment', is_done: true },
  { id: 4, title: 'Do W4 Assigment', is_done: false },
  { id: 8, title: 'Liang Yu, 208410331', is_done: false }
]
```

=> vercel

1122-2-n-tasks-vercel-31.vercel.app

Tasks Added

Task Bud -- 208410331

Add Task

- Do W1 Assigment Delete
- Do W2 Assigment Delete
- Do W3 Assigment Delete
- Do W4 Assigment Delete
- Liang Yu, 208410331 Delete
- Add New Tasks,208410331 Delete

bbc3eda 陳亮瑜 Wed Mar 20 21:31:22 2024 +0800 ### W5-P4: Create new Task, deploy W5-P3

W5: Git log of W5

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-03-18"
```

```
bbc3eda 陳亮瑜 Wed Mar 20 21:31:22 2024 +0800 ### W5-P4: Create new Task, deploy W5-P3
6b123a3 陳亮瑜 Wed Mar 20 20:33:39 2024 +0800 ### W5-P3: Deploy W5-P2 to Vercel
44b2c3e 陳亮瑜 Wed Mar 20 20:08:23 2024 +0800 ### W5-P2: Use React Query to get tasks
8f0c529 陳亮瑜 Wed Mar 20 19:17:25 2024 +0800 ### W5-P2: 設定 supabase client, 使使用 .env
6a783ea 陳亮瑜 Wed Mar 20 18:39:36 2024 +0800 git log --pretty=format:%h%x09%an%x09%ad%
```