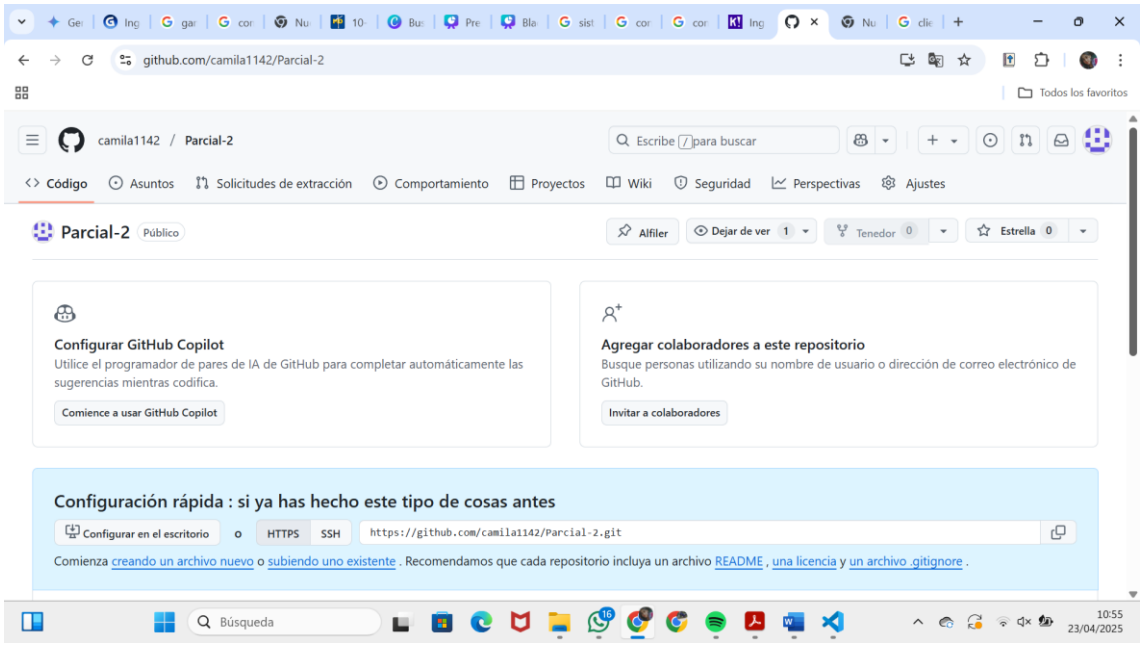
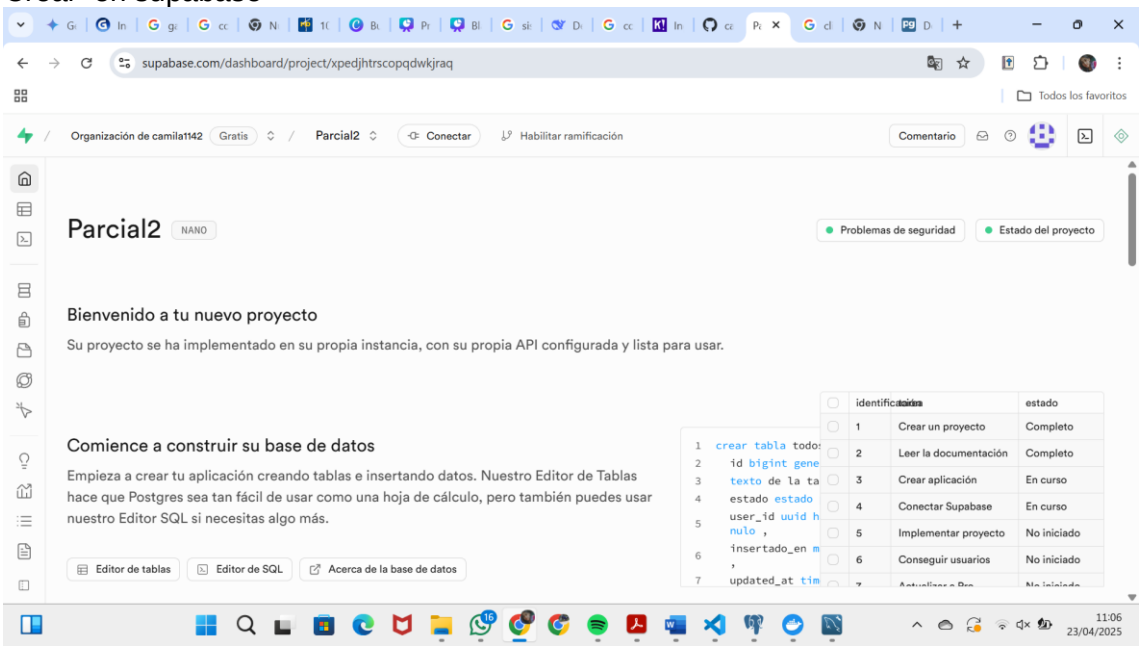
Parcial corte 2

Julieth Camila Castro Parada

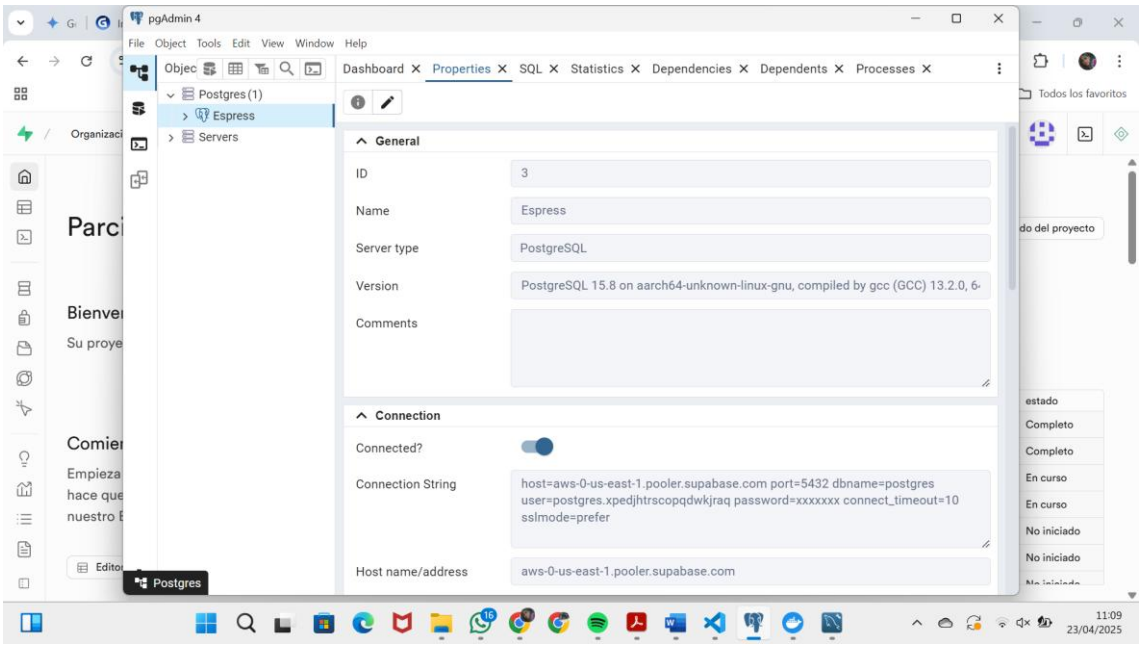
1. Crear repositorio



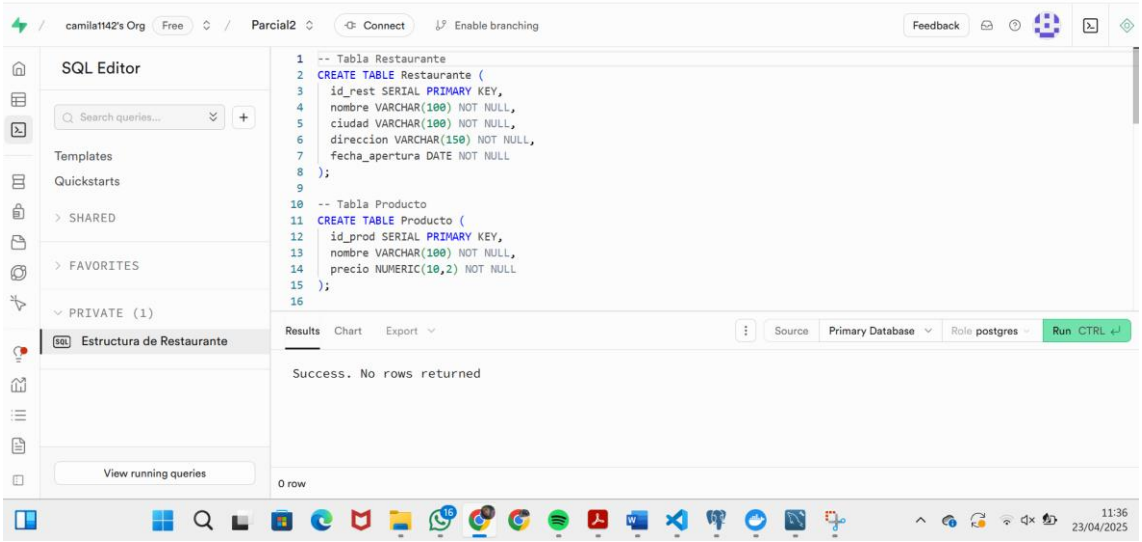
2. Crear en supabase

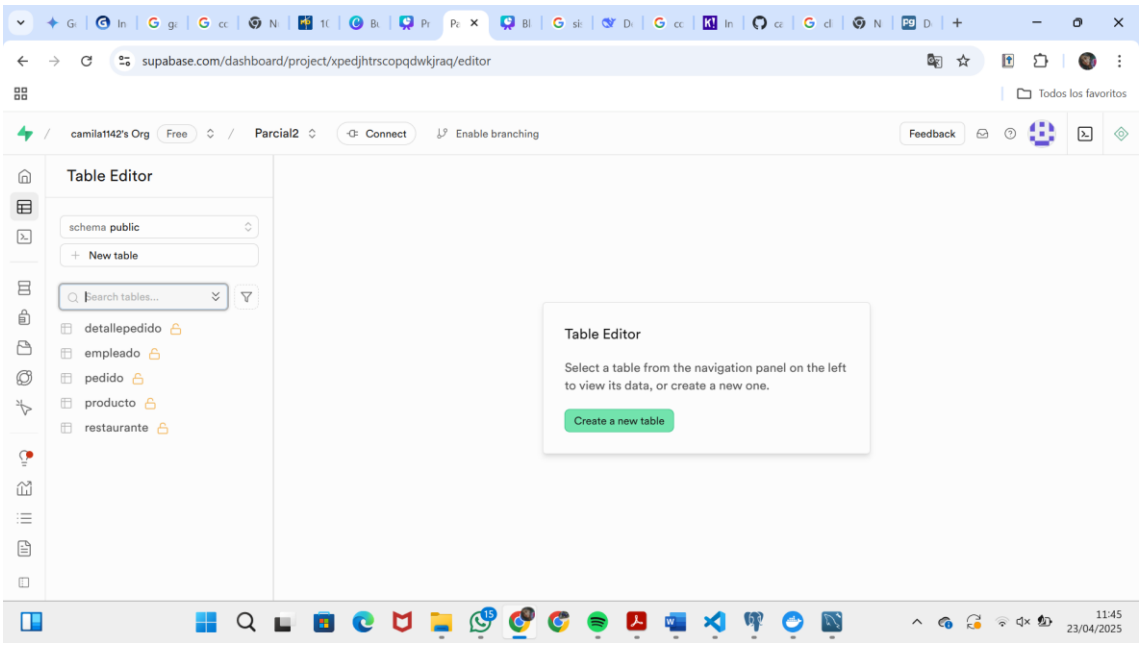


3. Hacer conexion con supabase

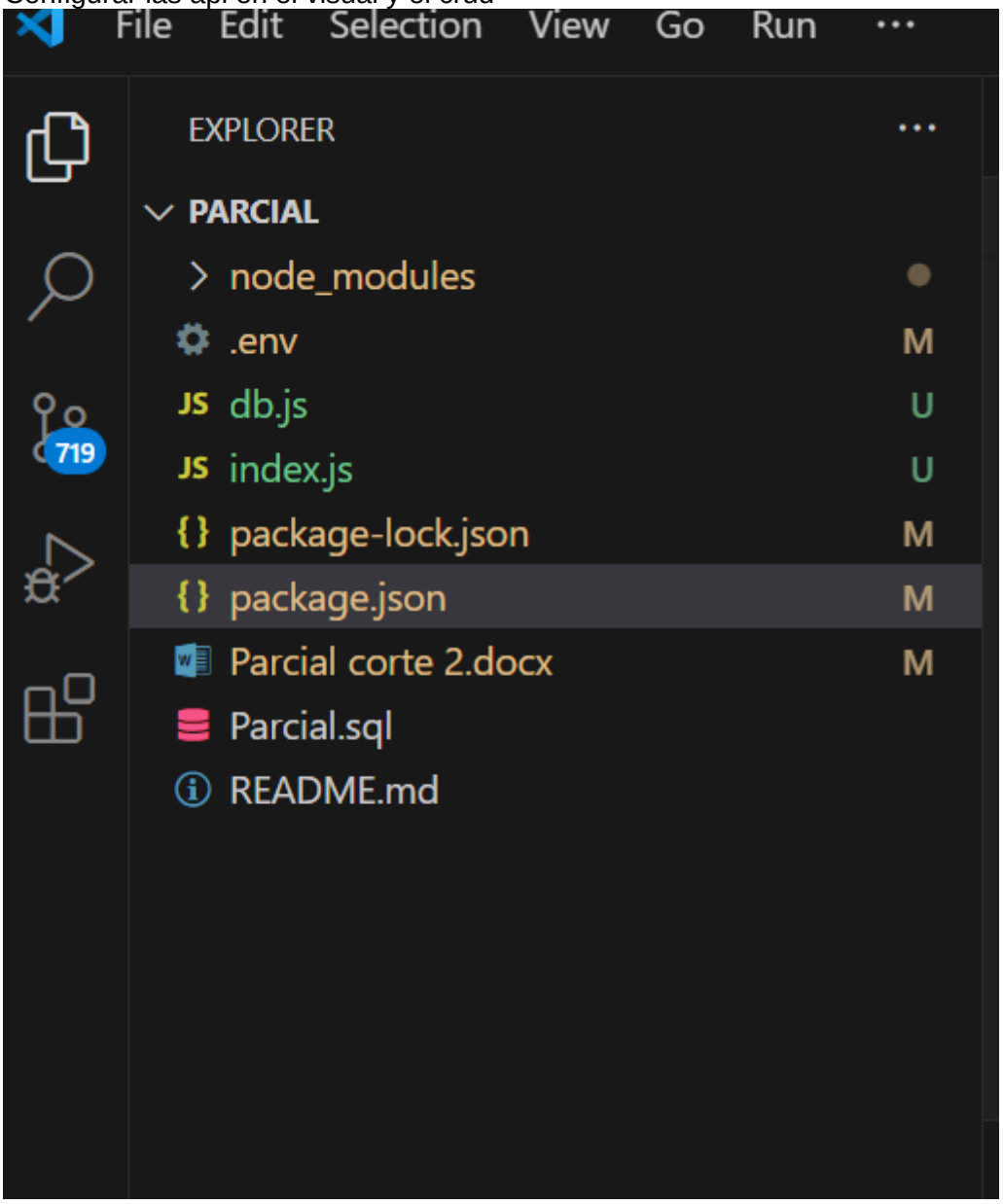


Crear tablas en supabase

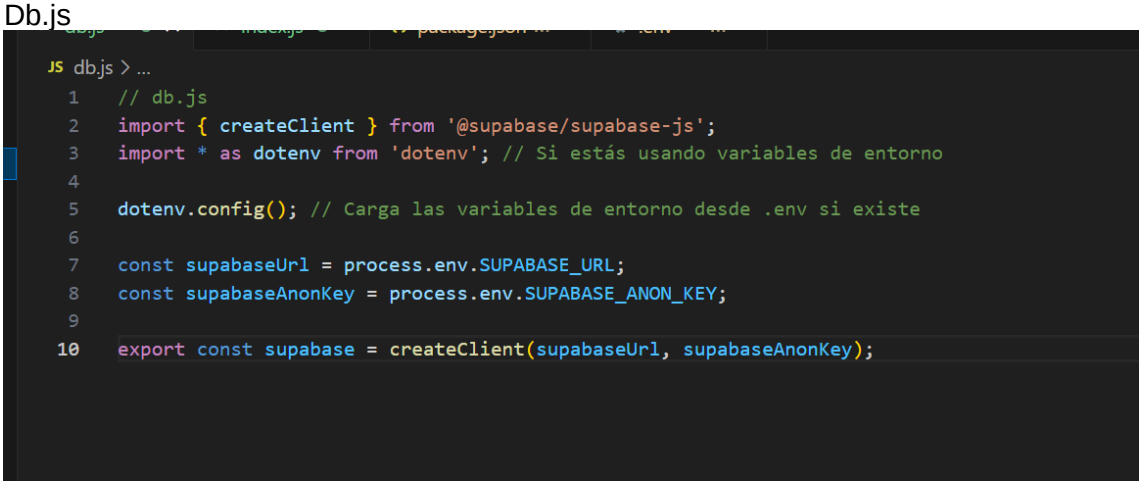



supabase.com/dashboard/project/xpedjhtrscopqdwkjraq/editor
camila1142's Org
Free
Parcial2
Connect
Enable branching
Feedback
Table Editor
schema public
New table
Search tables...
detallepedido
empleado
pedido
producto
restaurante
Table Editor
Select a table from the navigation panel on the left to view its data, or create a new one.
Create a new table
11:45
23/04/2025

Configurar las api en el visual y el crud



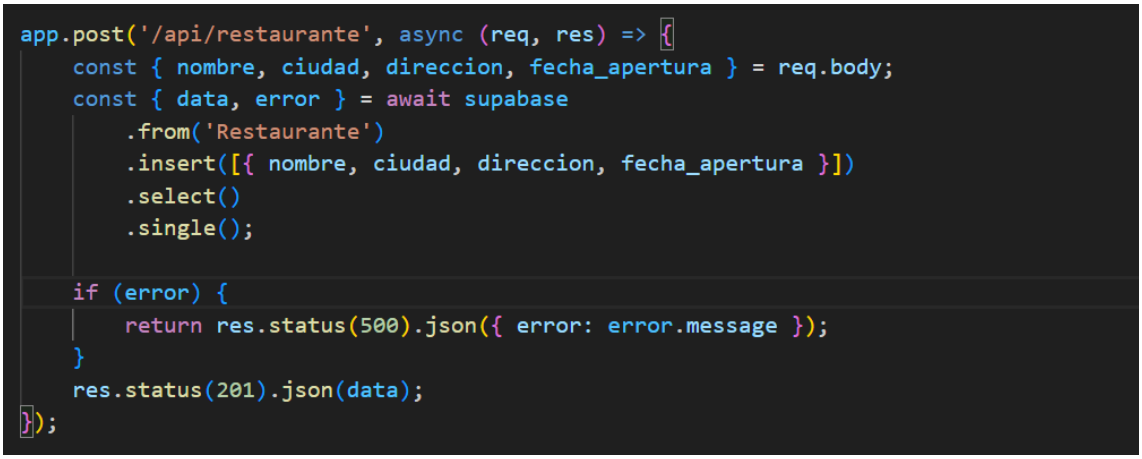
Db.js



Api get restaurante

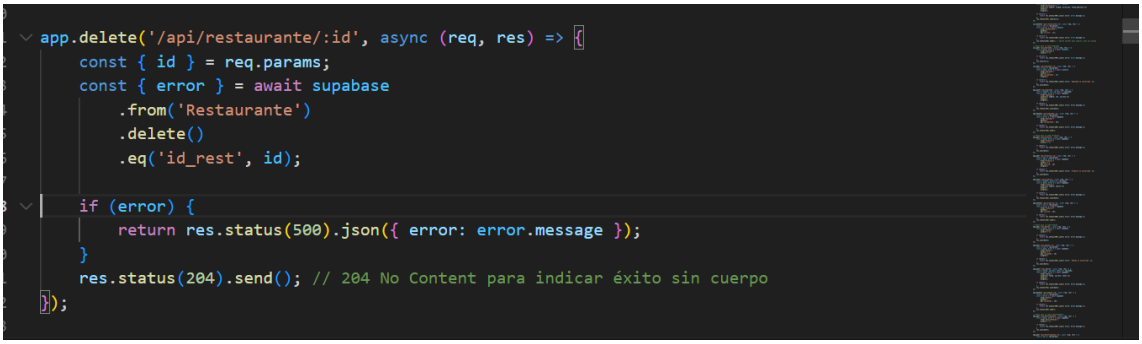
```
// Rutas para la tabla Restaurante
app.get('/api/restaurante', async (req, res) => {
    const { data, error } = await supabase
        .from('Restaurante')
        .select('*');

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.json(data);
});
```

Post



```
app.post('/api/restaurante', async (req, res) => {
    const { nombre, ciudad, direccion, fecha_apertura } = req.body;
    const { data, error } = await supabase
        .from('Restaurante')
        .insert([{ nombre, ciudad, direccion, fecha_apertura }])
        .select()
        .single();

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.status(201).json(data);
});
```

Delete



```
app.delete('/api/restaurante/:id', async (req, res) => {
    const { id } = req.params;
    const { error } = await supabase
        .from('Restaurante')
        .delete()
        .eq('id_rest', id);

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.status(204).send(); // 204 No Content para indicar éxito sin cuerpo
});
```

Get empleado

```
// Rutas para la tabla Empleado
app.get('/api/empleado', async (req, res) => {
    const { data, error } = await supabase
        .from('Empleado')
        .select('*');

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.json(data);
});

app.get('/api/empleado/:id', async (req, res) => {
    const { id } = req.params;
    const { data, error } = await supabase
        .from('Empleado')
        .select('*')
        .eq('id_empleado', id)
        .single();

    if (error) {
        return res.status(404).json({ error: 'Empleado no encontrado' });
    }
    res.json(data);
});
```

Post y delete empleado

```
89
90  app.post('/api/empleado', async (req, res) => {
91      const { nombre, rol, id_rest } = req.body;
92      const { data, error } = await supabase
93          .from('Empleado')
94          .insert([{ nombre, rol, id_rest }])
95          .select()
96          .single();
97
98      if (error) {
99          return res.status(500).json({ error: error.message });
100     }
101     res.status(201).json(data);
102 });
103
104 app.delete('/api/empleado/:id', async (req, res) => {
105     const { id } = req.params;
106     const { error } = await supabase
107         .from('Empleado')
108         .delete()
109         .eq('id_empleado', id);
110
111     if (error) {
112         return res.status(500).json({ error: error.message });
113     }
114     res.status(204).send();
115 });
```

Get Producto

```
// Rutas para la tabla Producto
app.get('/api/producto', async (req, res) => {
    const { data, error } = await supabase
        .from('Producto')
        .select('*');

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.json(data);
});

app.get('/api/producto/:id', async (req, res) => {
    const { id } = req.params;
    const { data, error } = await supabase
        .from('Producto')
        .select('*')
        .eq('id_prod', id)
        .single();

    if (error) {
        return res.status(404).json({ error: 'Producto no encontrado' });
    }
    res.json(data);
});
```

Post producto

```
app.post('/api/producto', async (req, res) => {
    const { nombre, precio } = req.body;
    const { data, error } = await supabase
        .from('Producto')
        .insert([{ nombre, precio }])
        .select()
        .single();

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.status(201).json(data);
});
```

Delete Producto

```
app.delete('/api/producto/:id', async (req, res) => {
    const { id } = req.params;
    const { error } = await supabase
        .from('Producto')
        .delete()
        .eq('id_prod', id);

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.status(204).send();
});
```

Get pedido

```
// Rutas para la tabla Pedido
app.get('/api/pedido', async (req, res) => {
    const { data, error } = await supabase
        .from('Pedido')
        .select('*');

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.json(data);
});

app.get('/api/pedido/:id', async (req, res) => {
    const { id } = req.params;
    const { data, error } = await supabase
        .from('Pedido')
        .select('*')
        .eq('id_pedido', id)
        .single();

    if (error) {
        return res.status(404).json({ error: 'Pedido no encontrado' });
    }
    res.json(data);
});

app.post('/api/pedido', async (req, res) => {
    const { fecha, id_rest, total } = req.body;
    const { data, error } = await supabase
```

Post pedido

```
app.post('/api/pedido', async (req, res) => {
    const { fecha, id_rest, total } = req.body;
    const { data, error } = await supabase
        .from('Pedido')
        .insert([{ fecha, id_rest, total }])
        .select()
        .single();

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.status(201).json(data);
});
```

Delete pedido

```
app.delete('/api/pedido/:id', async (req, res) => {
    const { id } = req.params;
    const { error } = await supabase
        .from('Pedido')
        .delete()
        .eq('id_pedido', id);

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.status(204).send();
});
```

Get de detalle de pedido

```
app.get('/api/detallepedido', async (req, res) => {
    const { data, error } = await supabase
        .from('DetallePedido')
        .select('*');

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.json(data);
});

app.get('/api/detallepedido/:id', async (req, res) => {
    const { id } = req.params;
    const { data, error } = await supabase
        .from('DetallePedido')
        .select('*')
        .eq('id_detalle', id)
        .single();

    if (error) {
        return res.status(404).json({ error: 'Detalle de Pedido no encontrado' });
    }
    res.json(data);
});
```

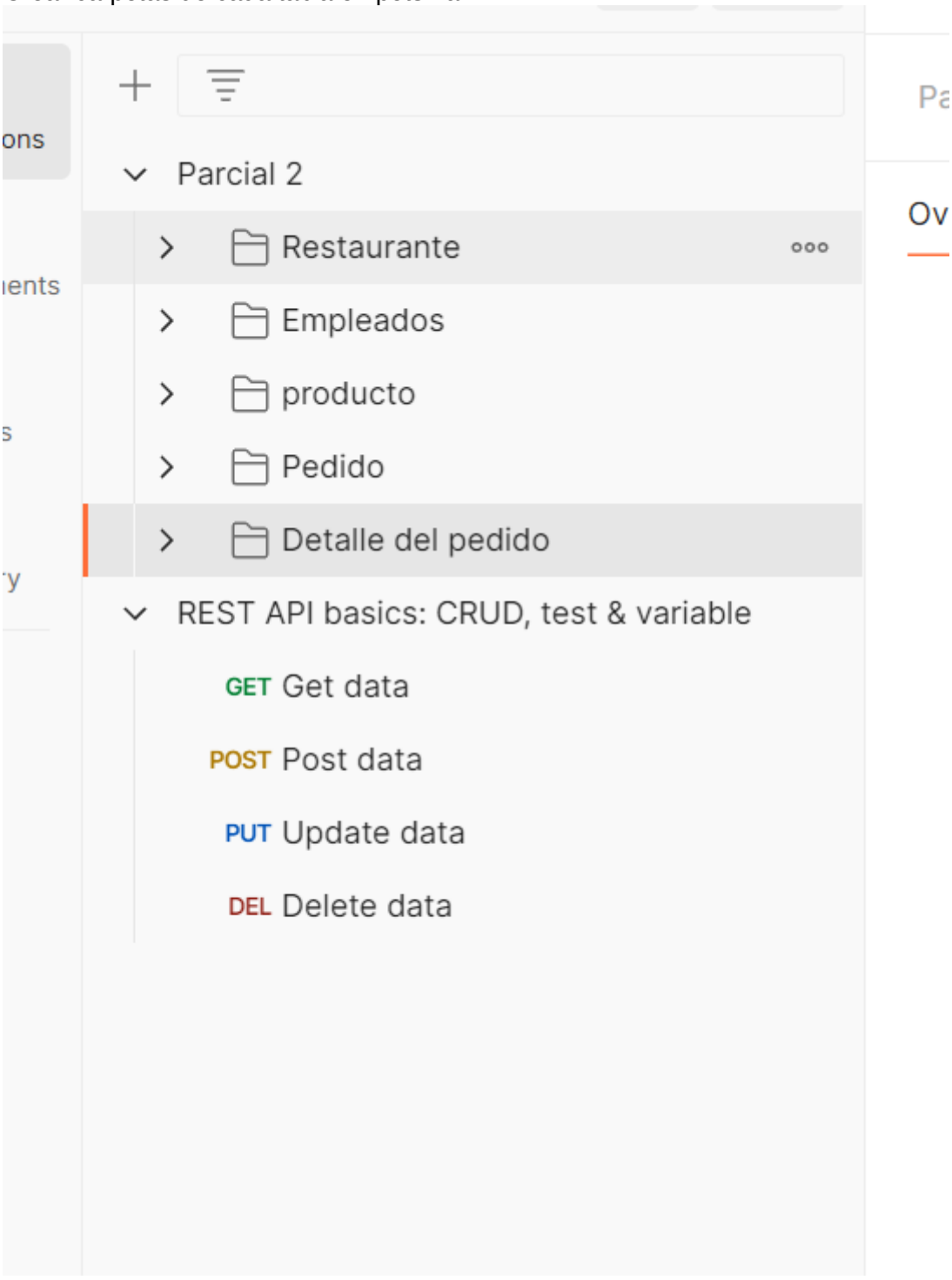
Post detalle de pedido

```
app.post('/api/detallepedido', async (req, res) => {
    const { id_pedido, id_prod, cantidad, subtotal } = req.body;
    const { data, error } = await supabase
        .from('DetallePedido')
        .insert([{ id_pedido, id_prod, cantidad, subtotal }])
        .select()
        .single();

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.status(201).json(data);
```

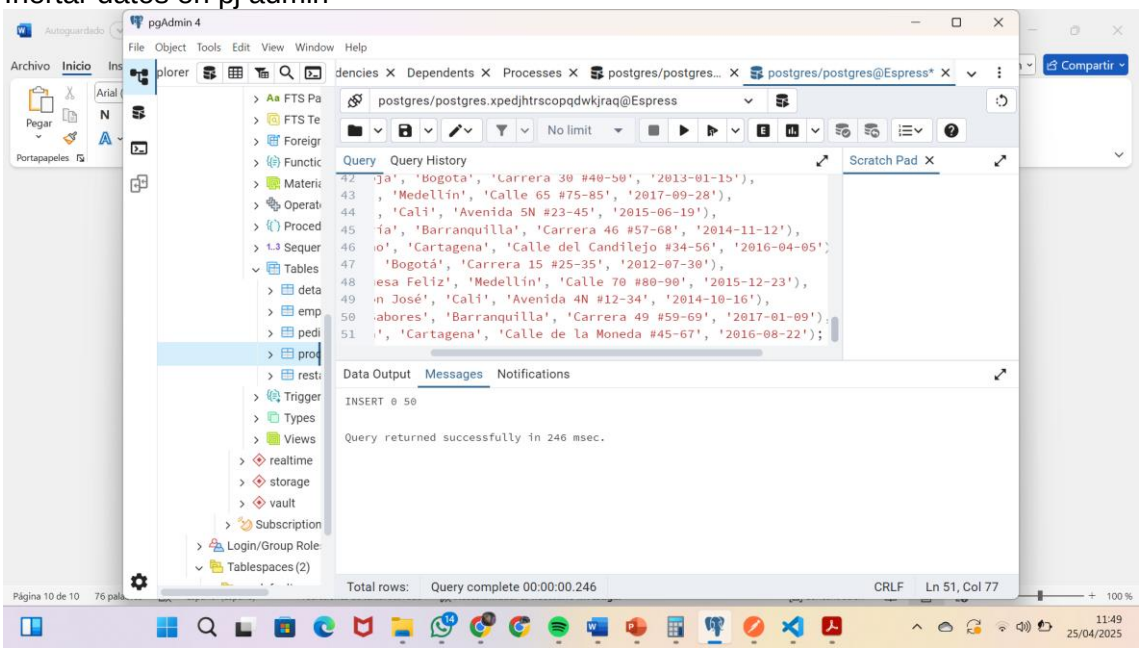
Delete detalle de pedido

```
app.delete('/api/detallepedido/:id', async (req, res) => {
    const { id } = req.params;
    const { error } = await supabase
        .from('DetallePedido')
        .delete()
        .eq('id_detalle', id);

    if (error) {
        return res.status(500).json({ error: error.message });
    }
    res.status(204).send();
});

app.listen(port, () => {
    console.log(`Servidor escuchando en el puerto ${port}`);
});
```

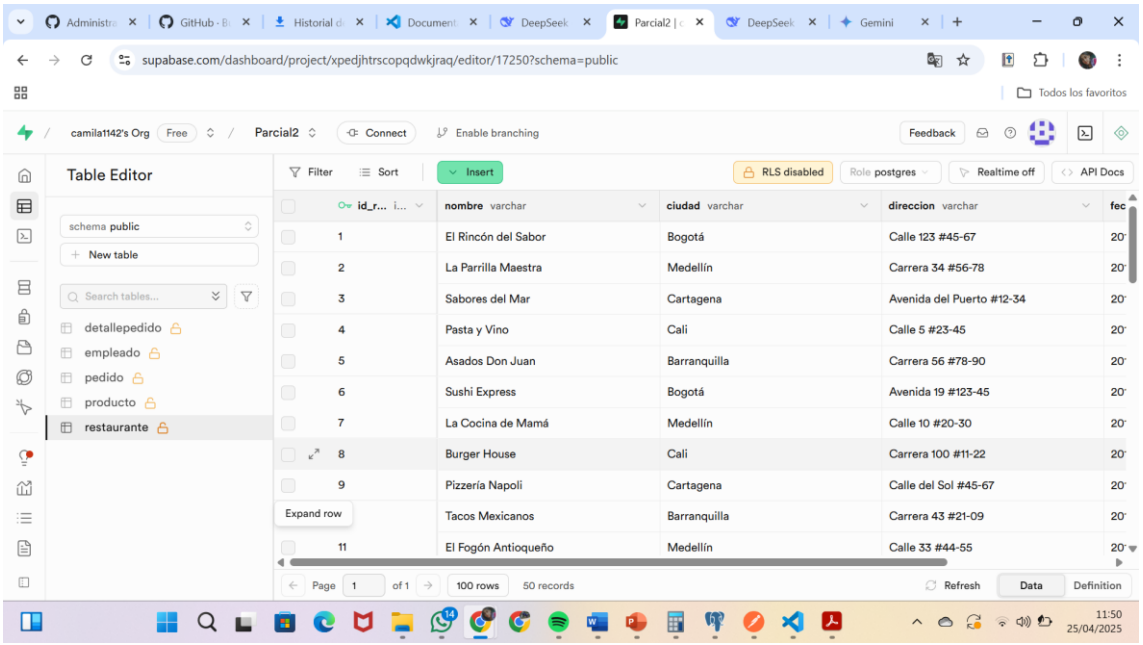
Crear carpetas de cada tabla en potsman



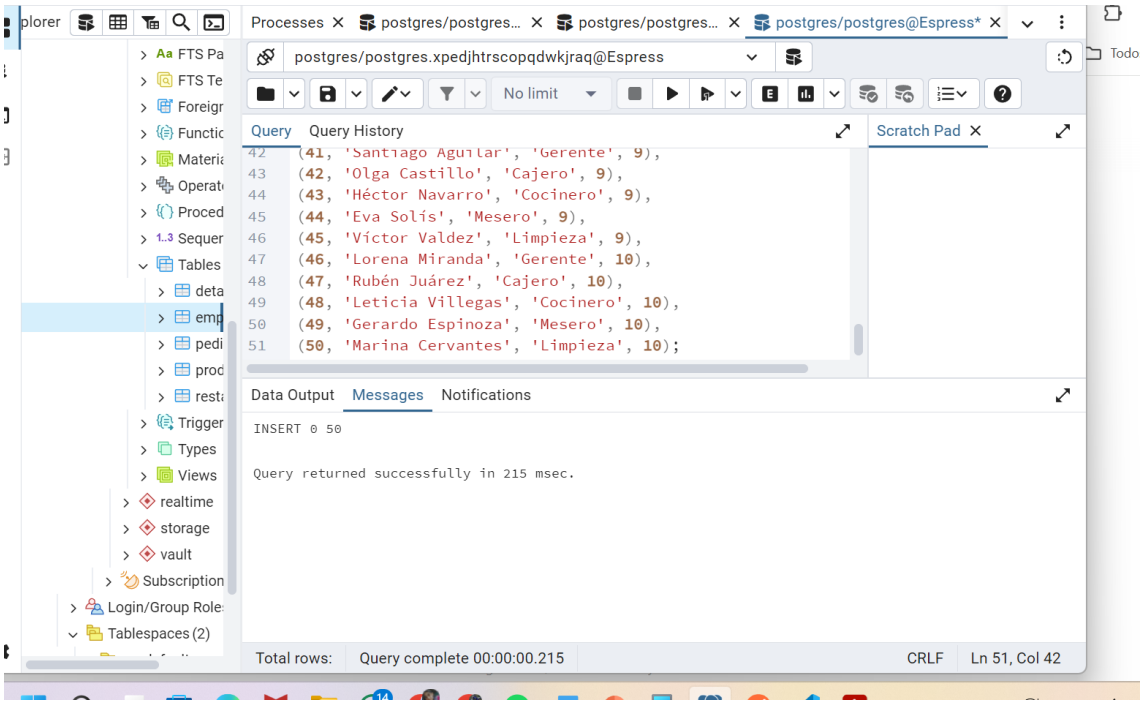
Inertar datos en pj admin



Verifica que aperezcan en supabase



Inserta datos en la tabla empleados



Verifica en subabase los datos insertados

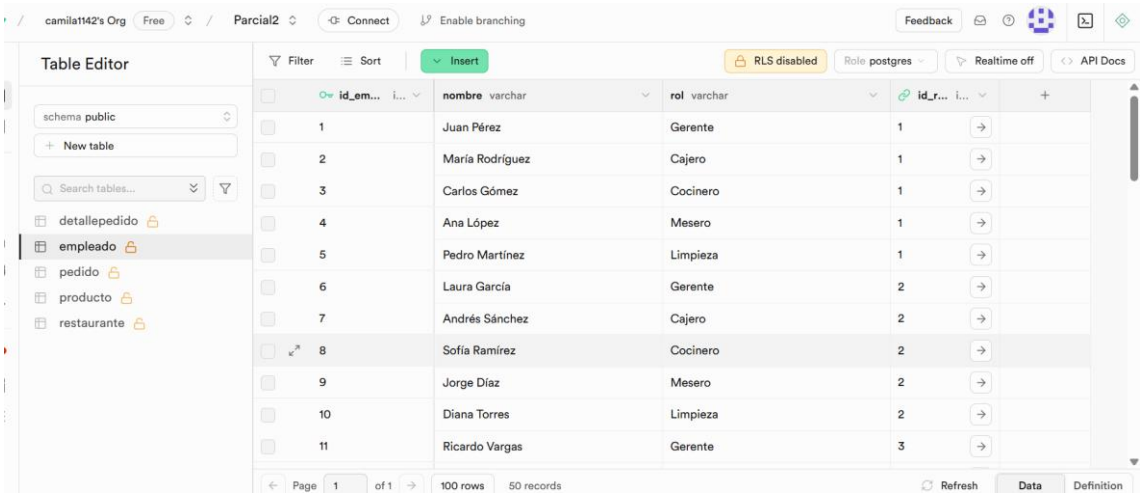
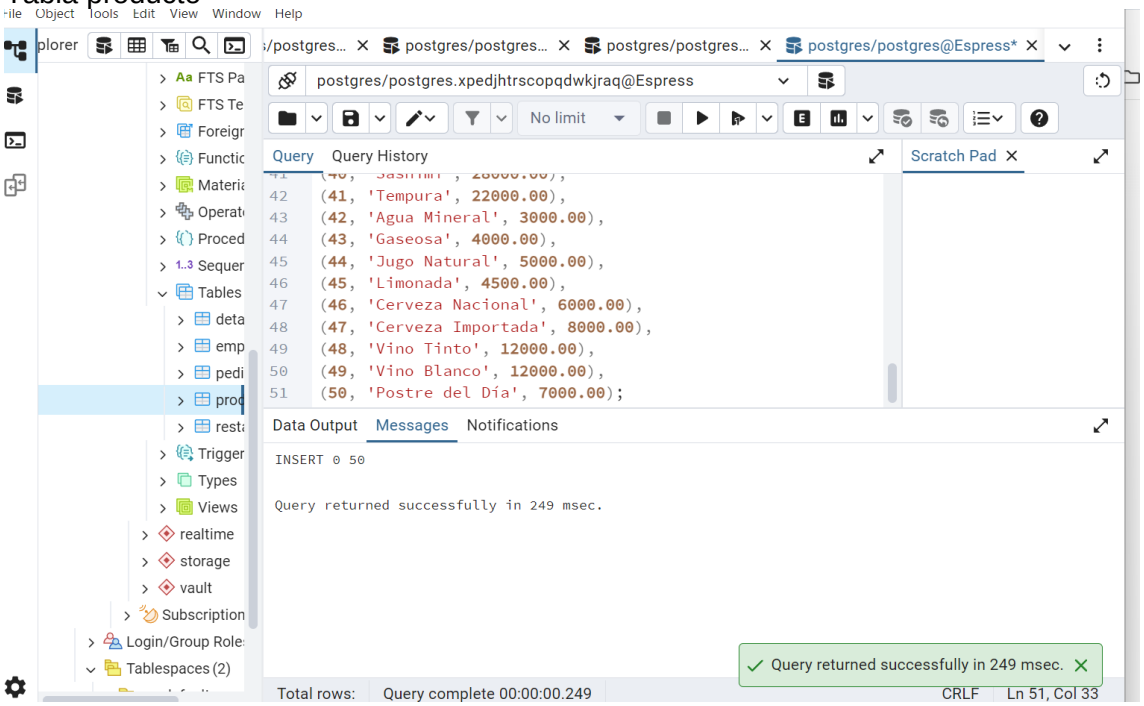
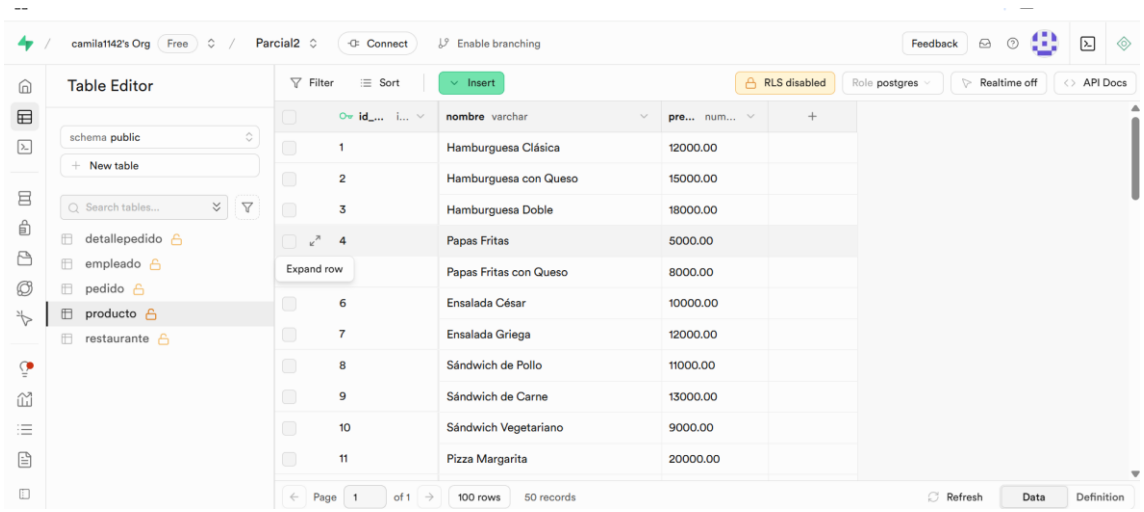


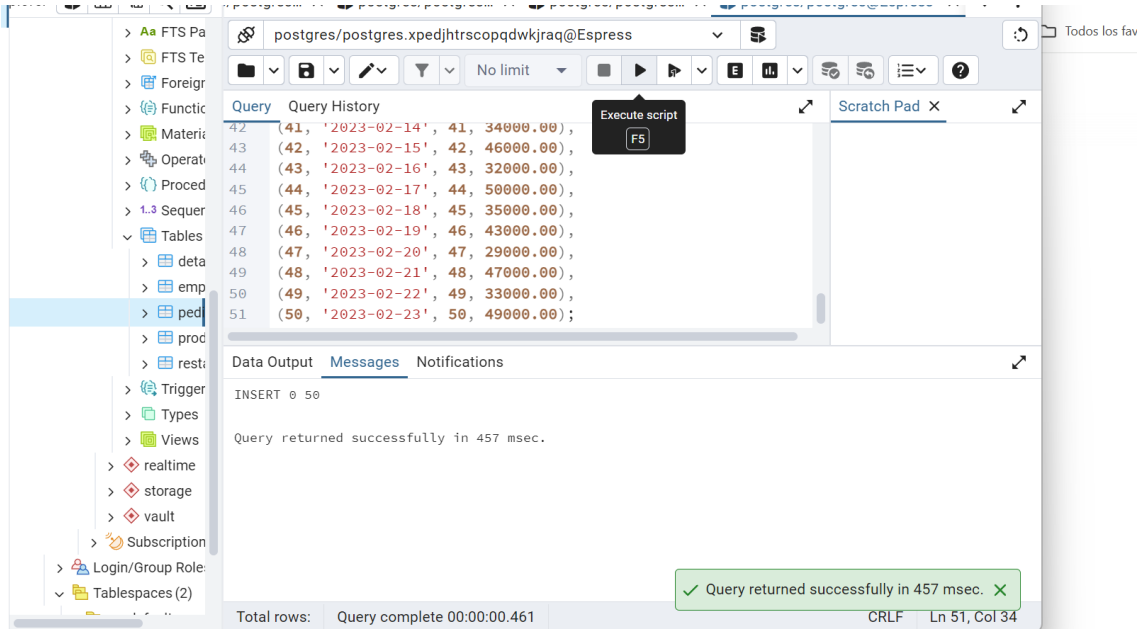
Tabla producto



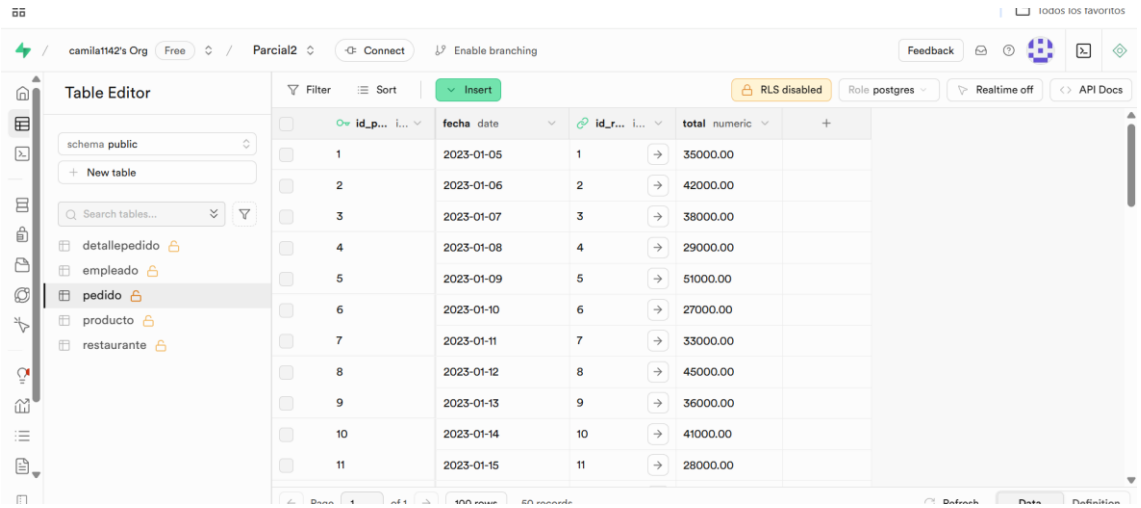
Verifica los datos en supabase



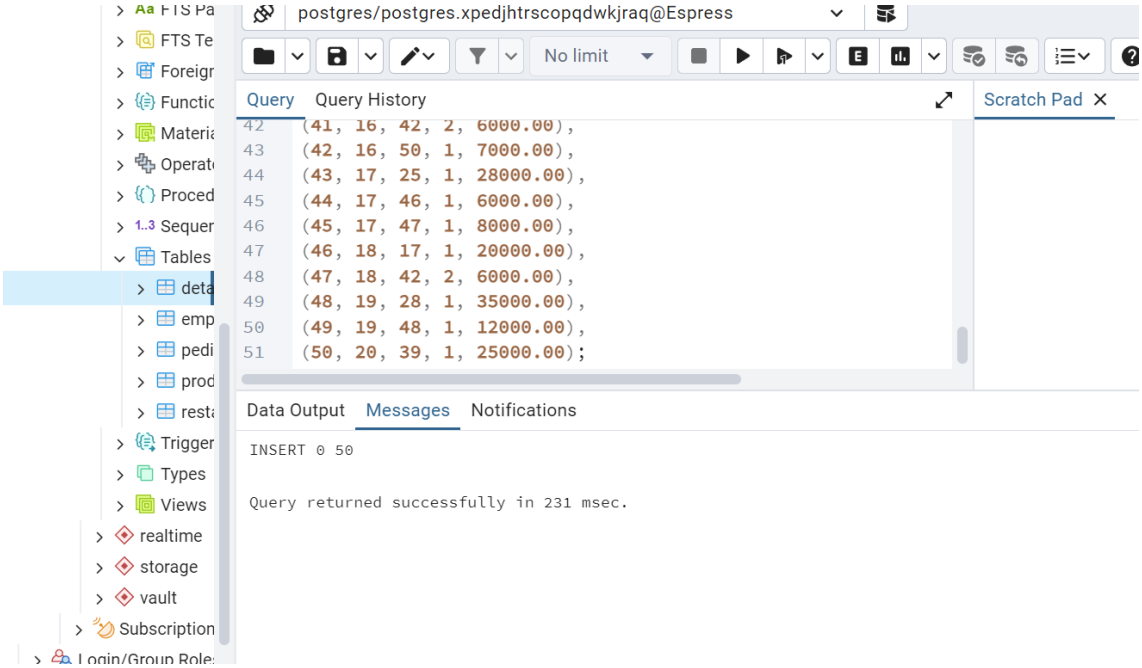
Inserta en la tabla pedidos



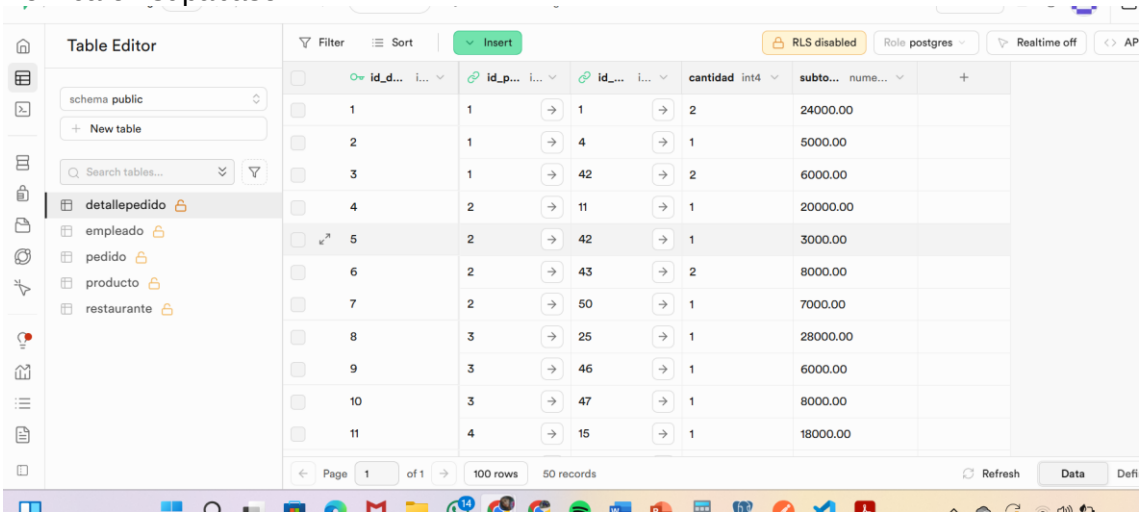
Verifica en supabase



Inserta en la tabla detalle del pedido



Verifica en supabase



Esquema en supabase