

ฐานข้อมูลร้านชาบูบุฟเฟต์

นายธีรภัทร คงเกต 116510901001-7

นายสุทธิวัฒน์ สุวรรณมณี 116510901004-1

นางสาวอัญชลีพร บุญปลูก 116510901006-6

เสนอ

ดร.รัฐพรหม พรหมคำ

ผู้ช่วยศาสตราจารย์ ดร.วงศ์วิศรุต เขื่องสตุง

สาขาวิชาคณิตศาสตร์ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

November 4, 2024

สารบัญ

1 ฐานข้อมูลร้านชาบูบุฟเฟต์

1.1 ปัญหา 4

1.2 ER Diagram 4

2 SQL

2.1 SQL คืออะไร และตัวอย่าง SQL 5-7

2.2 SQL ที่ใช้ในการแก้ปัญหา 8-10

3 Data On Website

3.1 Codeที่ใช้ในการขึ้นเว็บไซต์..... 11

3.1.1 server.js 11-13

3.1.2 HTML 14-24

3.1.3 script.js 25-28

4 Data On GIT

บทที่ 1

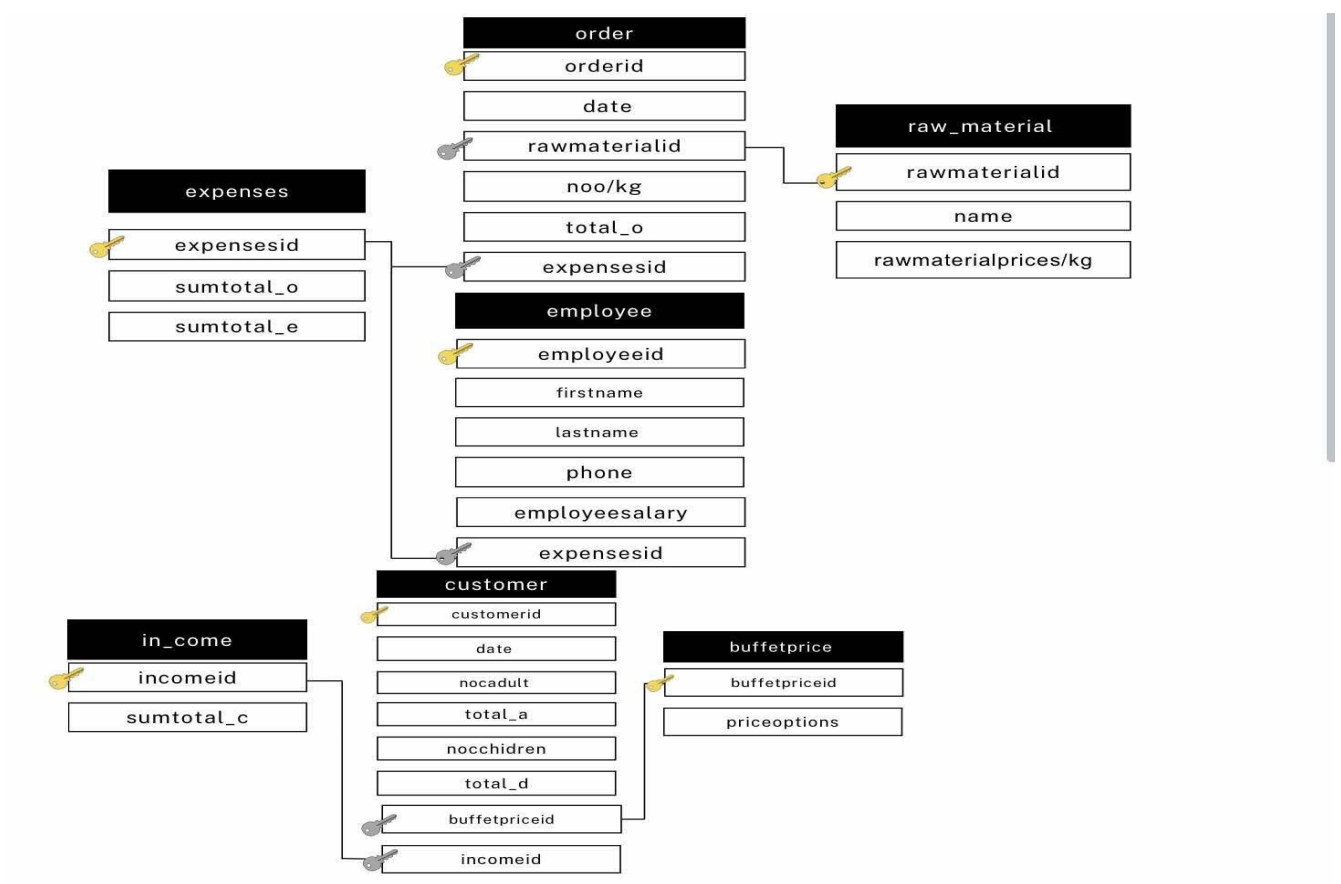
ฐานข้อมูลร้านชาบูบุฟเฟต์

การพัฒนาฐานข้อมูลสำหรับร้านชาบูบุฟเฟต์มีวัตถุประสงค์เพื่อจัดการข้อมูลสินค้าภายในร้านชาบู, ลูกค้า, และการสั่งซื้อวัตถุดิบ, การสั่งออเดอร์ ระบบฐานข้อมูลนี้ถูกออกแบบมาเพื่อตอบสนองความต้องการในการเก็บข้อมูลที่มีความหลากหลายของการคิดกำไรรายวัน, ราคาวัตถุดิบ, และจำนวนลูกค้า นอกจากนี้ยังมีการจัดเก็บข้อมูลลูกค้าเพื่อช่วยในการวิเคราะห์พฤติกรรมการซื้อและสร้างความสัมพันธ์ที่ดีกับลูกค้า การใช้ฐานข้อมูลเชิงสัมพันธ์ทำให้สามารถดึงข้อมูลที่ต้องการได้อย่างรวดเร็วและแม่นยำการวิจัยนี้ยังได้สำรวจเทคโนโลยีต่าง ๆ ที่สามารถนำมาใช้ในการพัฒนาระบบ เช่น SQL และโปรแกรมจัดการฐานข้อมูล เพื่อให้สามารถจัดการข้อมูลได้อย่างมีประสิทธิภาพ ผลลัพธ์ที่ได้จากการพัฒนาฐานข้อมูลนี้จะช่วยเพิ่มประสิทธิภาพในการดำเนินงานของร้านชาบูบุฟเฟต์ และยกระดับประสบการณ์ของเจ้าของกิจการร้านชาบูบุฟเฟต์

1.1 ปัญหา

- 1.กำไรจากการเปิดร้านภายในสองสัปดาห์เป็นอย่างไร
- 2.วัตถุดิบในร้านที่ราคาต่อกิโลแพ่งที่สุดคืออะไร
- 3.วัตถุดิบในร้านที่ราคาต่อกิโลถูกที่สุดคืออะไร
- 4.เงินเดือนพนักงานคนไหนสูงที่สุด
- 5.จำนวนลูกค้าที่มาใช้บริการเยอะที่สุดและได้รับยอดเงินเป็นจำนวนเท่าไร

1.2 ER Diagram



บทที่ 2

SQL

2.1 SQL คืออะไร และตัวอย่าง SQL

SQL คือ Structured Query Language (SQL) เป็นภาษาโปรแกรมสำหรับจัดเก็บและประมวลผลข้อมูลในฐานข้อมูลแบบเชิงสัมพันธ์ ฐานข้อมูลแบบเชิงสัมพันธ์เก็บข้อมูลในรูปแบบตารางที่มีแถวและคอลัมน์ที่เป็นตัวแทนของหมวดข้อมูลที่แตกต่างกันและความสัมพันธ์ต่างๆ ระหว่างค่าข้อมูล สามารถใช้คำสั่ง SQL ในการจัดเก็บ ปรับปรุง ลบ ค้นหา และดึงข้อมูลจากฐานข้อมูล นอกจากนี้ยังสามารถใช้ SQL ในการรักษาและเพิ่มประสิทธิภาพการทำงานของฐานข้อมูล

ตาราง SQL เป็นองค์ประกอบพื้นฐานของฐานข้อมูลแบบเชิงสัมพันธ์ ตารางฐานข้อมูล SQL ประกอบด้วยแถวและคอลัมน์ วิศวกรฐานข้อมูลสร้างความสัมพันธ์ระหว่างตารางฐานข้อมูลหลายตารางเพื่อเพิ่มประสิทธิภาพพื้นที่เก็บข้อมูล

ตัวอย่างในการสร้าง SQL ที่ใช้ในการสร้างฐานข้อมูลร้านชาบูบุฟเฟต์ ในที่นี้ใช้เป็นตารางของวัตถุดิบมายกตัวอย่าง

```
CREATE TABLE raw_material (  
  
    ( rawmaterialid INT PRIMARY KEY,  
  
    name VARCHAR(100) NOT NULL,  
  
    rawmaterialprices_kg DECIMAL(10, 2) NOT NULL  
  
);
```

และตัวอย่างในการใส่ข้อมูลลงในตาราง จากตัวอย่างด้านบน

INSERT INTO raw_material VALUES

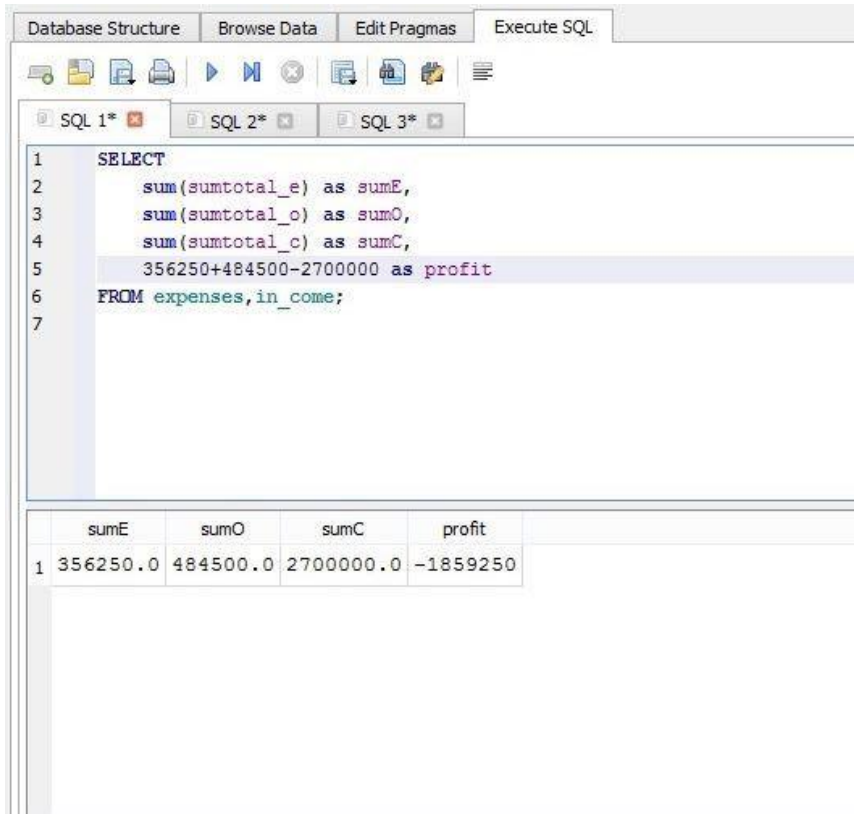
(1, 'Sirloin', 250),
(2, 'Gyu harami', 300),
(3, 'Tendered beef', 270),
(4, 'Pork' , 180),
(5, 'Bacon' , 150),
(6, 'Pork liver' , 140),
(7, 'Tendered pork' , 160),
(8, 'Pork black pepper sauce' , 120),
(9, 'Pork mala sauce' , 110),
(10, 'Buta corm' , 145),
(11, 'Fresh shrimp' , 220),
(12, 'Squid' , 180),
(13, 'Clam' , 130),
(14, 'Salmon head' , 70),
(15, 'Pangasius hypophthalmus' , 90),
(16, 'Jellyfish' , 60),
(17, 'Pork ball' , 120),
(18, 'Fish ball' , 70),
(19, 'Kani tsume ball' , 80),

(20, 'Seaweed roll' , 60),
(21, 'Cheese tofu' , 100),
(22, 'Fish tofu' , 65),
(23, 'Egg tofu' , 75),
(24, 'Egg' , 120),
(25, 'Vermicelli' , 40),
(26, 'Spinach' , 120),
(27, 'Morning glory' , 70),
(28, 'Pak choi' , 60),
(28, 'Celery' , 80),
(29, 'Chinese cabbage' , 65),
(30, 'Cabbage' , 55),
(31, 'Straw mushroom' , 120),
(32, 'Enoki mushroom' , 80),
(33, 'Sweet corn' , 50),
(34, 'Pumpkin' , 30),
(35, 'Carrot' , 40),
(36, 'Watermelon' , 45),
(37, 'Pineapple' , 50);

SELECT * FROM raw_material;

2.1 SQL ที่ใช้ในการแก้ปัญหา

1.กำไรจากการเปิดร้านภายในสองสัปดาห์เป็นอย่างไร



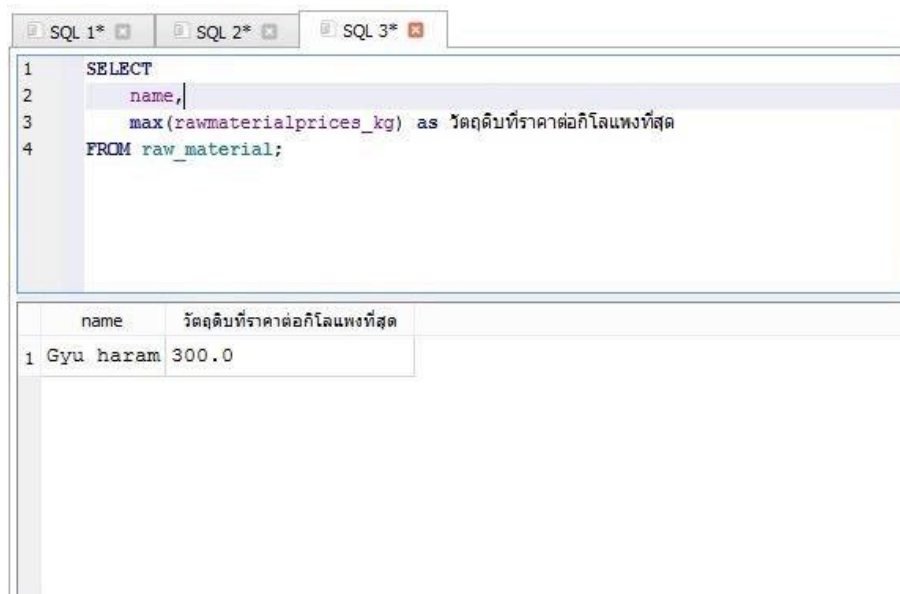
The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL code:

```
1 SELECT
2     sum(sumtotal_e) as sumE,
3     sum(sumtotal_o) as sumO,
4     sum(sumtotal_c) as sumC,
5     356250+484500-2700000 as profit
6 FROM expenses,in_come;
7
```

The results pane displays a single row of data:

	sumE	sumO	sumC	profit
1	356250.0	484500.0	2700000.0	-1859250

2.วัตถุดิบในร้านที่ราคาต่อกิโลแพงที่สุดคืออะไร



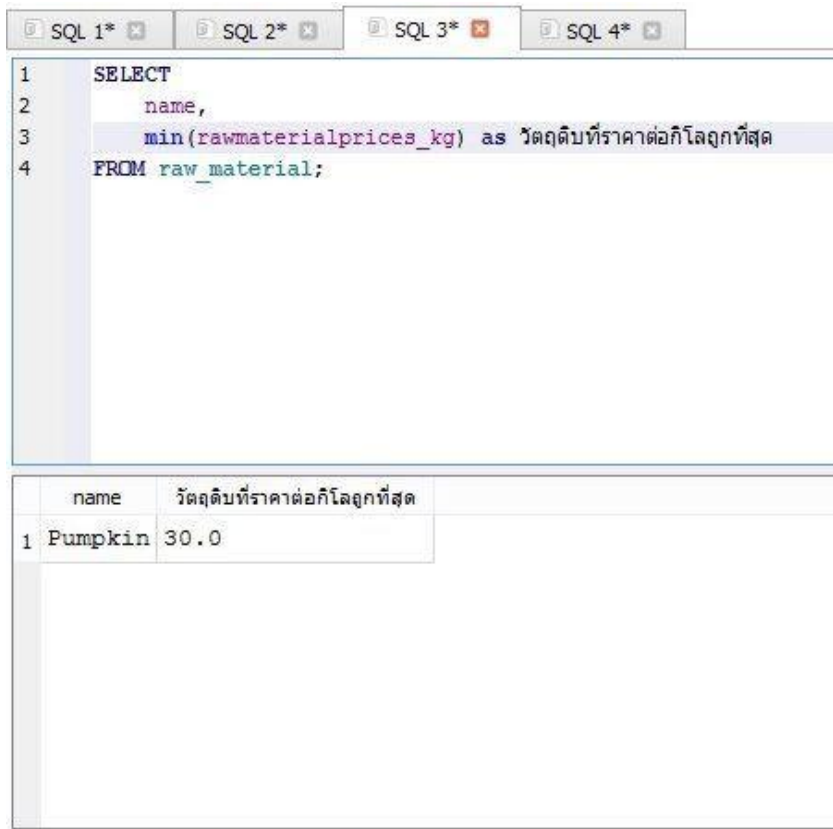
The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL code:

```
1 SELECT
2     name,
3     max(rawmaterialprices_kg) as วัตถุดิบที่ราคาต่อกิโลแพงที่สุด
4 FROM raw_material;
```

The results pane displays a single row of data:

	name	วัตถุดิบที่ราคาต่อกิโลแพงที่สุด
1	Gyu haram	300.0

3. วัตถุดิบในร้านที่ราคาต่อกิโลถูกที่สุดคืออะไร



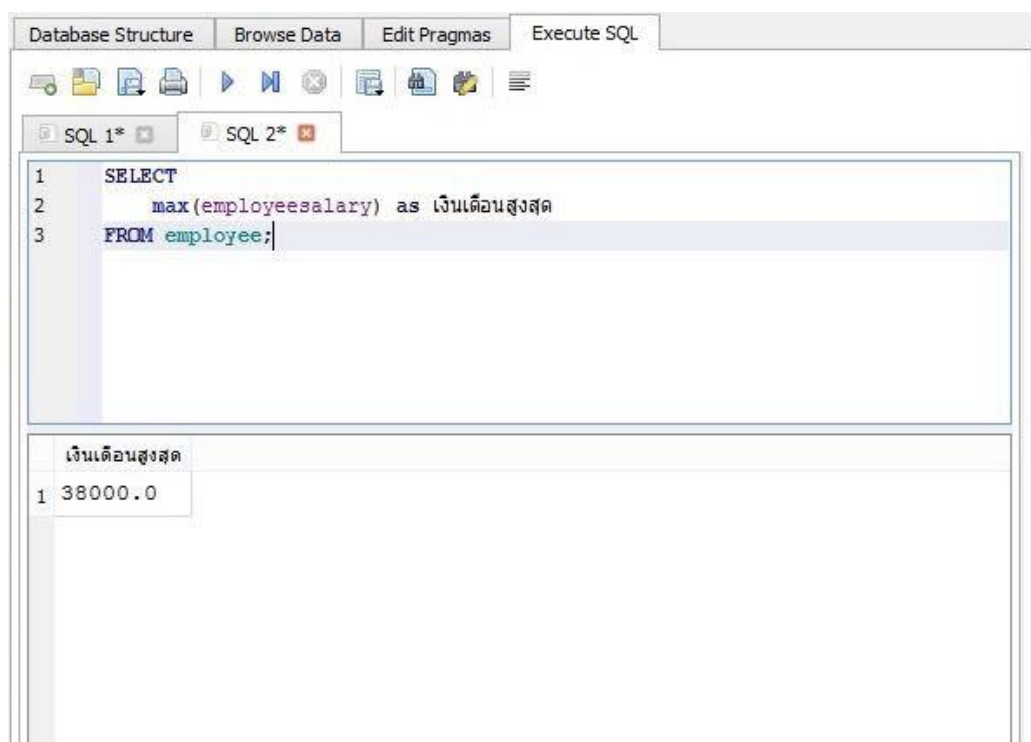
The screenshot shows a SQL Developer window with four tabs: SQL 1*, SQL 2*, SQL 3*, and SQL 4*. The active tab is SQL 1*, which contains the following SQL query:

```
1 SELECT
2     name,
3     min(rawmaterialprices_kg) as วัตถุดิบที่ราคาต่อกิโลถูกที่สุด
4 FROM raw_material;
```

Below the query editor, the results are displayed in a table with two columns: 'name' and 'วัตถุดิบที่ราคาต่อกิโลถูกที่สุด'. The table contains one row with the value 'Pumpkin' and '30.0'.

	name	วัตถุดิบที่ราคาต่อกิโลถูกที่สุด
1	Pumpkin	30.0

4. เงินเดือนพนักงานคนไหนสูงที่สุด



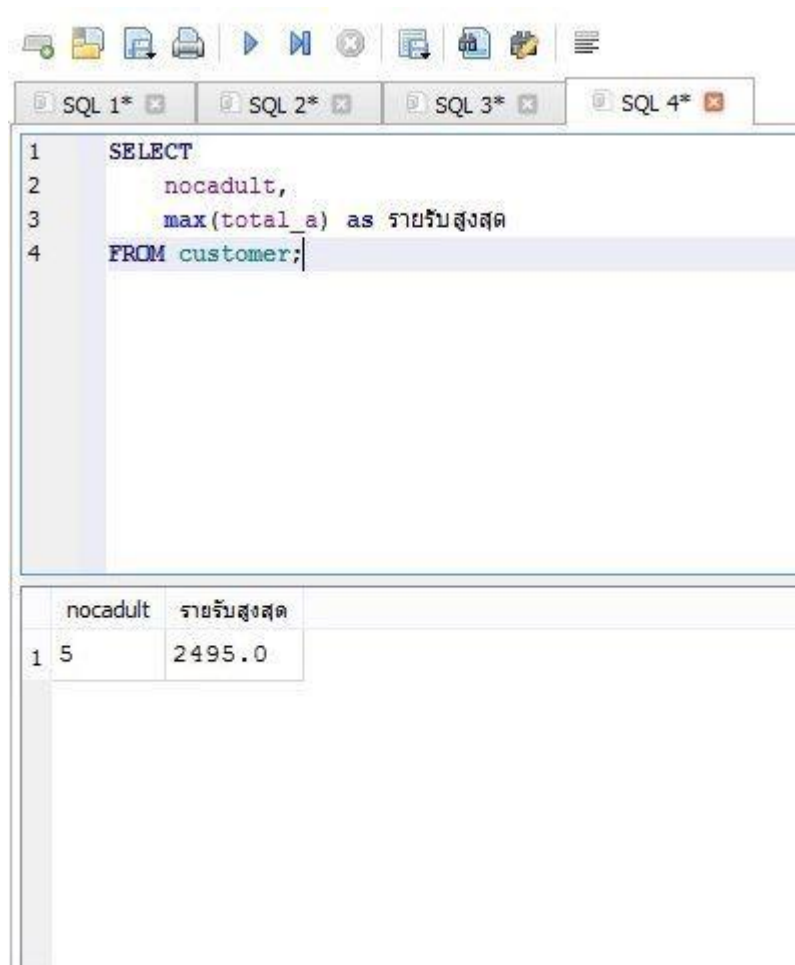
The screenshot shows a SQL Developer window with four tabs: Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The active tab is Execute SQL, which contains the following SQL query:

```
1 SELECT
2     max(employeeesalary) as เงินเดือนสูงสุด
3 FROM employee;
```

Below the query editor, the results are displayed in a table with one column: 'เงินเดือนสูงสุด'. The table contains one row with the value '38000.0'.

	เงินเดือนสูงสุด
1	38000.0

5.จำนวนลูกค้าที่มาใช้บริการเยอะที่สุดและได้รับยอดเงินเป็นจำนวนเท่าไร



```
1 SELECT
2     nocadult,
3     max(total_a) as รายรับสูงสุด
4 FROM customer;
```

	nocadult	รายรับสูงสุด
1	5	2495.0

บทที่ 3

Data On Website

3.1 Code ในการนำขึ้นเว็บไซต์

3.1.1 server.js

ส่วนแรกคือ server.js ที่เป็นการนำตารางขึ้นบนเว็บไซต์ ส่วนของการสร้างและแสดงตารางตัวอย่าง

```
const express = require('express');  
  
const sqlite3 = require('sqlite3').verbose();  
  
const path = require('path');  
  
  
const app = express();  
  
const PORT = 3000;  
// เชื่อมต่อฐานข้อมูล SQLite  
const db = new sqlite3.Database('./neww.db', (err) => {  
  if (err) {  
    console.error('Error opening database', err);  
  } else {  
    console.log('Connected to the SQLite database.');  }  
});
```

```

// เปิดใช้งาน middleware สำหรับการประมวลผล JSON และ static files
app.use(express.json());
app.use(express.static(path.join(__dirname, 'public'))); // สมมติว่าไฟล์
HTML อยู่ในโฟลเดอร์ public
// ตัวอย่างการเรียกข้อมูลจากตาราง customer
app.get('/api/customers', (req, res) => {
    const sql = 'SELECT * FROM customer';
    db.all(sql, [], (err, rows) => {
        if (err) {
            res.status(400).json({ error: err.message });
            return;
        }
        res.json({ data: rows });
    });
});
// Endpoint สำหรับค้นหาข้อมูลในตาราง customer
app.get('/api/search/customers', (req, res) => {
    const searchQuery = req.query.q; // รับคำค้นหาจาก URL parameter
    const sql = `SELECT * FROM customer WHERE firstName LIKE ? OR
lastName LIKE ?`; // ตัวอย่างการค้นหาในตาราง customer
    const params = [`%${searchQuery}%`, `%${searchQuery}%`];

```

```

db.all(sql, params, (err, rows) => {
  if (err) {
    res.status(400).json({ error: err.message });
    return;
  }
  res.json({ data: rows });
});

});
// ปิดการเชื่อมต่อฐานข้อมูลเมื่อหยุดใช้งานแอป
app.on('close', () => {
  db.close();
});
// เริ่มเซิร์ฟเวอร์
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
  app.get('/', (req, res) => {
    res.send('Hell Express!');
  })
});

```

3.1.2 HTML

ส่วนที่2คือ HTML หรือส่วนที่แสดงบนเว็บไซต์ตัวอย่าง

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Shabu buffrt</title>
```

```
<link href="22.css" rel="stylesheet" type="text/css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```

```

```
<h1>Shabu buffrt</h1>
```

```
<nav>
```

```
<a href="home.html">Home</a>
```

```
<a href="about.html">Order</a>
```

Buffetprice

Contact

</nav>

</div>

</div>

<h1>Buffetprice</h1>

<form id="buffetpriceForm">

<input type="number" id="bufsfetpriceid" placeholder="buffetpriceid">

<input type="number" id="priceoptions" placeholder="priceoptions">

<button type="submit">Find items</button>

</form>

<table id="buffetpriceList">

<tr>

<th>buffetpriceid</th>

<th>priceoptions</th>

</tr>

</table>

<h1>customer</h1>

<form id="customerForm">

<input type="number" id="customerid" placeholder="customerid">

<input type="text" id="date" placeholder="date">

<input type="number" id="nocadult" placeholder="nocadult">

<input type="number" id="total_a" placeholder="total_a">

<input type="number" id="nocchildren" placeholder="nocchildren">

<input type="number" id="total_d" placeholder="total_d">

<input type="number" id="buffetpriceid" placeholder="buffetpriceid">

<input type="number" id="" placeholder="incomeid">

<button type="submit">Find customer</button>

</form>

<table id="customerList">

<tr>

<th>nocadult</th>

<th>total_a</th>

<th>nocchildren</th>

<th>total_d</th>

<th>buffetpriceid</th>

<th>incomeid</th>

</tr>

</table>

<h1>employee</h1>

<form id="employeeForm">

<input type="number" id="employeeid" placeholder="employeeid">

<input type="text" id="FirstName" placeholder="firstName">

<input type="text" id="LastName" placeholder="lastName">

<input type="phone" id="phone" placeholder="phone">

<input type="employeesalay" id="employeesalay"

placeholder="employeesalay">

<input type="expensesid" id="expensesid" placeholder="expensesid">

<button type="submit">Find employee</button>

</form>

<table id="employeeList">

<tr>

<th>employeeid</th>

<th>firstName</th>

<th>lastName</th>

<th>phone</th>

<th>employeesalary</th>

<th>expensesid</th>

</tr>

</table>

<h1>expenses</h1>

<form id="expensesForm">

<input type="number" id="expenses" placeholder="expenses">

<input type="number" id="sumtotal_o" placeholder="sumtotal_o">

<input type="number" id="sumtotal_e" placeholder="sumtotal_e">

<button type="submit">Find expenses</button>

</form>

<table id="expensesList">

<tr>

<th>expenses</th>

<th>sumtotal_o</th>

<th>sumtotal_e</th>

</tr>

</table>

<h1>in_come</h1>

<form id="in_comeForm">

<input type="number" id="in_comeid" placeholder="in_comeid">

<input type="number" id="sumtotal_c" placeholder="sumtotal_c">

<button type="submit">Find</button>

</form>

```
<table id="in_comeList">
```

```
<tr>
```

```
<th>in_comeid</th>
```

```
<th>sumtotal_c</th>
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<h1>order</h1>
```

```
<form id="orderForm">
```

```
<input type="number" id="orderid" placeholder="orderid">
```

```
<input type="text" id="date" placeholder="date">
```

```
<input type="number" id="rawmaterialid" placeholder="rawmaterialid">
```

```
<input type="number" id="noo_kg" placeholder="noo_kg">
```

```
<input type="number" id="total_o" placeholder="total_o">
```

```
<input type="number" id="expensesid" placeholder="expensesid">
```

```
<button type="submit">Find order</button>
```

```
</form>
```

```
<table id="orderList">
```

```
<tr>
```

```
<th>orderid</th>
```

```
<th>date</th>
```

```
<th>rawmaterialid</th>
```

```
<th>noo_kg</th>
```

```
<th>total_o</th>
```

```
<th>expensesid</th>
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<h1>raw_material</h1>
```

```
<form id="raw_materialForm">
```

```
<input type="number" id="raw_materialid" placeholder="raw_materialid">
```

```
<input type="text" id="name" placeholder="name">
```

```
<input type="number" id="raw_materialprices_kg"
```

```
placeholder="raw_materialprices_kg">
```

<button type="submit">Find</button>

</form>

<table id="raw_materialList">

<tr>

<th>Traw_materialid</th>

<th>name</th>

<th>raw_materialprices_kg</th>

</tr>

</table>

<!--เพิ่มการกรอก-->

<body>

<h1>ค้นหาข้อมูลลูกค้า</h1>

<form id="searchForm">

<input type="text" id="searchQuery" placeholder="กรอกคำค้นหา...">

<button type="submit">ค้นหา</button>

</form>

```
<h2>ผลลัพธ์การค้นหา</h2>
```

```
<table id="searchResults">
```

```
<tr>
```

```
<th>ชื่อ</th>
```

```
<th>นามสกุล</th>
```

```
<th>ข้อมูลอื่นๆ</th>
```

```
</tr>
```

```
</table>
```

```
<script>
```

```
document.addEventListener('DOMContentLoaded', () => {
```

```
  fetch('/api/customers')
```

```
    .then(response => response.json())
```

```
    .then(data => {
```

```
      const customerTable = document.getElementById('customerList');
```

```
      data.data.forEach(customer => {
```

```
        const row = document.createElement('tr');
```

```
        row.innerHTML = `
```

```

        <td>${customer.nocadult}</td>

        <td>${customer.total_a}</td>

        <td>${customer.nocchildren}</td>

        <td>${customer.total_d}</td>

        <td>${customer.buffetpriceid}</td>

        <td>${customer.incomeid}</td>

        `;

        customerTable.appendChild(row);

    });

})

.catch(error => console.error('Error fetching customer data:', error));

});

</script>

</body>

</html>

```


3.1.3 script.js

ส่วนที่3คือ script.js ตัวอย่าง

```
async function fetchAllData() {  
  
  try {  
  
    const response = await fetch('/all-data');  
  
    const data = await response.json();  
  
    console.log('Data from API:', data); // เพิ่มการแสดงผลข้อมูลใน console  
  
    const resultContainer = document.getElementById('resultContainer');  
  
    resultContainer.innerHTML = ""; // เคลียร์เนื้อหาเก่าทุกครั้งที่ค้นหาใหม่  
  
    if (data.error) {  
  
      resultContainer.innerHTML = `<p>Error: ${data.error}</p>`;   
  
      return;  
  
    }  
  
    const tableNames = Object.keys(data);  
  
    if (tableNames.length === 0) {  
  
      resultContainer.innerHTML = `<p>No results found.</p>`;   
  
      return;  
    }  
  }  
}
```

```

}

// สร้างตารางสำหรับแต่ละตารางในฐานข้อมูล

tableNames.forEach(tableName => {

    const tableTitle = document.createElement('h2');

    tableTitle.textContent = tableName;

    resultContainer.appendChild(tableTitle);

    const table = document.createElement('table');

    table.border = '1';

    resultContainer.appendChild(table);

    if (data[tableName].length > 0) {

        const headerRow = document.createElement('tr');

        Object.keys(data[tableName][0]).forEach(key => {

            const th = document.createElement('th');

            th.textContent = key;

            headerRow.appendChild(th);

        });

        table.appendChild(headerRow);
    }
}

```

```

data[tableName].forEach(row => {

    const tr = document.createElement('tr');

    Object.values(row).forEach(value => {

        const td = document.createElement('td');

        td.textContent = value;

        tr.appendChild(td);

    });

    table.appendChild(tr);

});

} else {

    const noDataRow = document.createElement('tr');

    const noDataCell = document.createElement('td');

    noDataCell.colSpan = Object.keys(data[tableName][0] || {}).length
|| 1;

    noDataCell.textContent = 'No data available';

    noDataRow.appendChild(noDataCell);

    table.appendChild(noDataRow);

```

```
    }

    });

    } catch (error) {

        console.error('Error:', error);

        document.getElementById('resultContainer').innerHTML = '<p>Error
loading data</p>';

    }

}

fetchAllData();
```

บทที่ 4

Data On GIT

The screenshot shows the GitHub interface for the repository 'minnie0244/project_shabu'. The repository is public and has 1 branch (main) and 5 commits. The file list includes:

File Name	Commit Message	Time Ago
0011.webp	initial commit	2 hours ago
0022.jpg	initial commit	2 hours ago
0033.png	initial commit	2 hours ago
0044.jpg	initial commit	2 hours ago
19.css	initial commit	2 hours ago
22.css	initial commit	2 hours ago
Bacon.jpg	initial commit	2 hours ago
Buta-corn.jpg	initial commit	2 hours ago
Cabbage.jpg	initial commit	2 hours ago
Carrot.jpg	initial commit	2 hours ago

The screenshot shows the same GitHub repository page, but with a different set of files and commit history. The file list includes:

File Name	Commit Message	Time Ago
Pork-black-pepper-sauce.webp	initial commit	2 hours ago
Pork-liver.jpg	initial commit	2 hours ago
Pork-mala-sauce.webp	initial commit	2 hours ago
Pumpkin.jpg	initial commit	2 hours ago
README.md	Create README.md	1 hour ago
Salmon-head.webp	initial commit	2 hours ago
Seaweed-roll.webp	initial commit	2 hours ago
Shabu.pdf.pdf	Add files via upload	1 hour ago
Spinach.jpg	initial commit	2 hours ago
Straw-mushroom.jpg	initial commit	2 hours ago
Sweet-corn.jpg	initial commit	2 hours ago
Tendered-beef.jpg	initial commit	2 hours ago
Tendered-pork.jpg	initial commit	2 hours ago
Untitled.ipynb	initial commit	2 hours ago
Vermicelli.jpg	initial commit	2 hours ago
Watermelon.jpg	initial commit	2 hours ago
app.js	initial commit	2 hours ago
gyu-haram.jpg	initial commit	2 hours ago

GitHub - minnie0244/project_shabu

File Name	Commit Type	Time Ago
Celery.jpg	initial commit	2 hours ago
Cheese-tofu.jpg	initial commit	2 hours ago
Chinese-cabbage.jpg	initial commit	2 hours ago
Clam.jpg	initial commit	2 hours ago
Egg-tofu.jpg	initial commit	2 hours ago
Egg.jpg	initial commit	2 hours ago
Enoki-mushroom.jpg	initial commit	2 hours ago
Fish-ball.jpg	initial commit	2 hours ago
Fish-tofu.jpg	initial commit	2 hours ago
Fresh-shrimp.jpg	initial commit	2 hours ago
Jellyfish.webp	initial commit	2 hours ago
Kani-tsume-ball.jpg	initial commit	2 hours ago
Morning-glory.jpg	initial commit	2 hours ago
Pak-choi.jpg	initial commit	2 hours ago
Pangasius-hypophthalmus.webp	initial commit	2 hours ago
Pineapple.webp	initial commit	2 hours ago
Pork-ball.png	initial commit	2 hours ago
Pork-black-pepper-sauce.webp	initial commit	2 hours ago

File Size Distribution:

- Jupyter Notebook 43.5%
- HTML 35.2%
- CSS 15.9%
- JavaScript 5.4%

style02.css Add files via upload 1 hour ago

style03.css Add files via upload 1 hour ago

table.html initial commit 2 hours ago

README

1.ชื่อโครงการ ฐานข้อมูลร้านชาบูบุฟเฟ่ต์ 2.ที่มาและความสำคัญ โครงการฉบับนี้จัดทำขึ้นเพื่อประโยชน์ในการศึกษาและเรียนรู้ ในการสร้างระบบฐานข้อมูลที่มีประสิทธิภาพสำหรับร้านชาบูบุฟเฟ่ต์ มีวัตถุประสงค์เพื่อจัดการข้อมูลสินค้าภายในร้านขาย, ลูกค้า, และการสั่งซื้อวัตถุดิบ, การส่งออเดอร์ ระบบฐานข้อมูลนี้ถูกออกแบบมาเพื่อตอบสนองความต้องการในการเก็บข้อมูลที่มีความหลากหลาย ของการคิดกำไรรายสัปดาห์, ราคาวัตถุดิบ, และจำนวนลูกค้า นอกจากนี้ยังมีการจัดเก็บข้อมูลลูกค้าเพื่อใช้ในการวิเคราะห์ พฤติกรรมการซื้อและสร้างความสัมพันธ์ที่ดีกับลูกค้า 3.จุดประสงค์ที่ทำไมโครงการ -เพื่อหาว่ากำไรในการเปิดร้านใน2สัปดาห์เป็น อย่างไร -เพื่อหาแนวทางในการเปิดร้านอย่างไรให้ไม่ขาดทุน -เพื่อสำรวจราคาวัตถุดิบในท้องถิ่นที่เปิดร้าน 4.กระบวนการ แก้ปัญหา จากฐานข้อมูลร้านชาบูบุฟเฟ่ต์ ได้กำหนดปัญหาขึ้นมาดังนี้ 1.ถ้าจากการเปิดร้านภายในสองสัปดาห์เป็นอย่างไร 2.วัตถุดิบในร้านที่ราคาต่อกิโลแพ่งที่สุดคืออะไร 3.วัตถุดิบในร้านที่ราคาต่อกิโลถูกที่สุดคืออะไร 4.เงินเดือนพนักงานคนไหนสูงที่สุด 5.จำนวนลูกค้าที่มาใช้บริการเยอะที่สุดและได้รับยอดเงินเป็นจำนวนเท่าไร 5.ผลลัพธ์ที่ได้จากโครงการ -จากการที่ได้ใช้ระบบ ฐานข้อมูลในการช่วยแก้ปัญหา ทำให้รู้ว่าการใช้ระบบฐานข้อมูลในการช่วยคิดกำไรได้สะดวกต่อการใช้งานเป็นอย่างมาก

- การวิเคราะห์ข้อมูลสามารถเห็นได้ชัดและผลลัพธ์ออกมาอย่างรวดเร็วเมื่อใช้งานและคำสั่งอย่างถูกต้อง 6.สรุปผลและข้อเสนอแนะ การดำเนินการไปเรื่องงานขึ้นนี้ยังคงค่อนข้างที่จะมีข้อผิดพลาดที่จะเยอะมาก เราได้ช่วยกันหาทางออกอย่างดีที่สุดแล้วแต่ก็ไม่ ประสบความสำเร็จ แต่ทางกลุ่มก็ได้ทำโปรเจกต์ขึ้นนี้กันอย่างตั้งใจและเต็มที่กับมันอย่างสุดความสามารถ เพราะมันคือครั้งแรกทาง หล่มพวกเรายังขอยกยไว ณ ที่นี้ด้วยนะ เต้นที่ที่สุดได้แค่นี้จริงๆ

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information