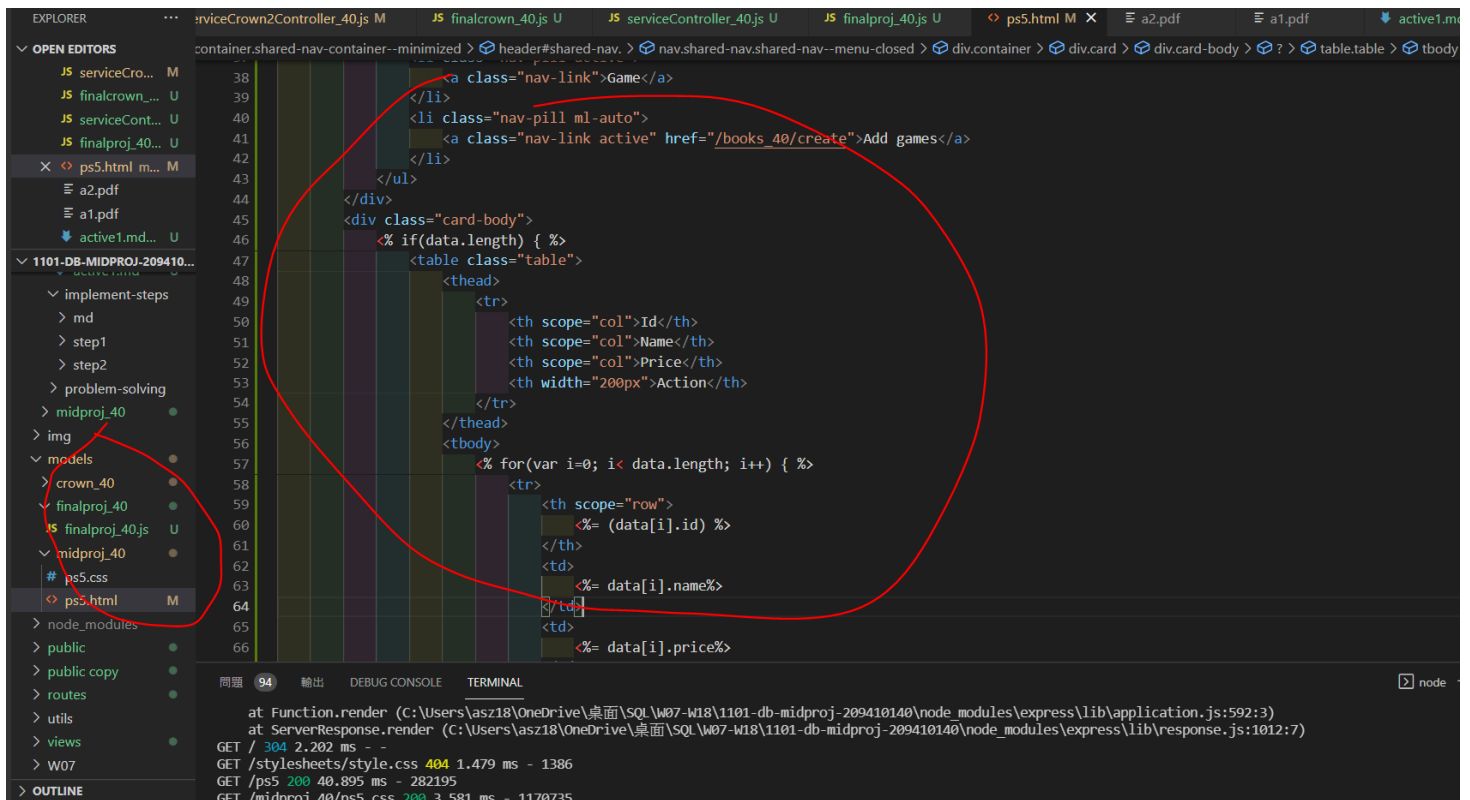


Active 1

嘗試加入**CRUD**的html介面，但弄了很久卻還是以失敗告終。

我是參考老師的上課教材來新增的，但成果不盡人意。



The screenshot shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'models', 'views', and 'controllers'. The code editor shows a file named 'ps5.html' with the following HTML code:

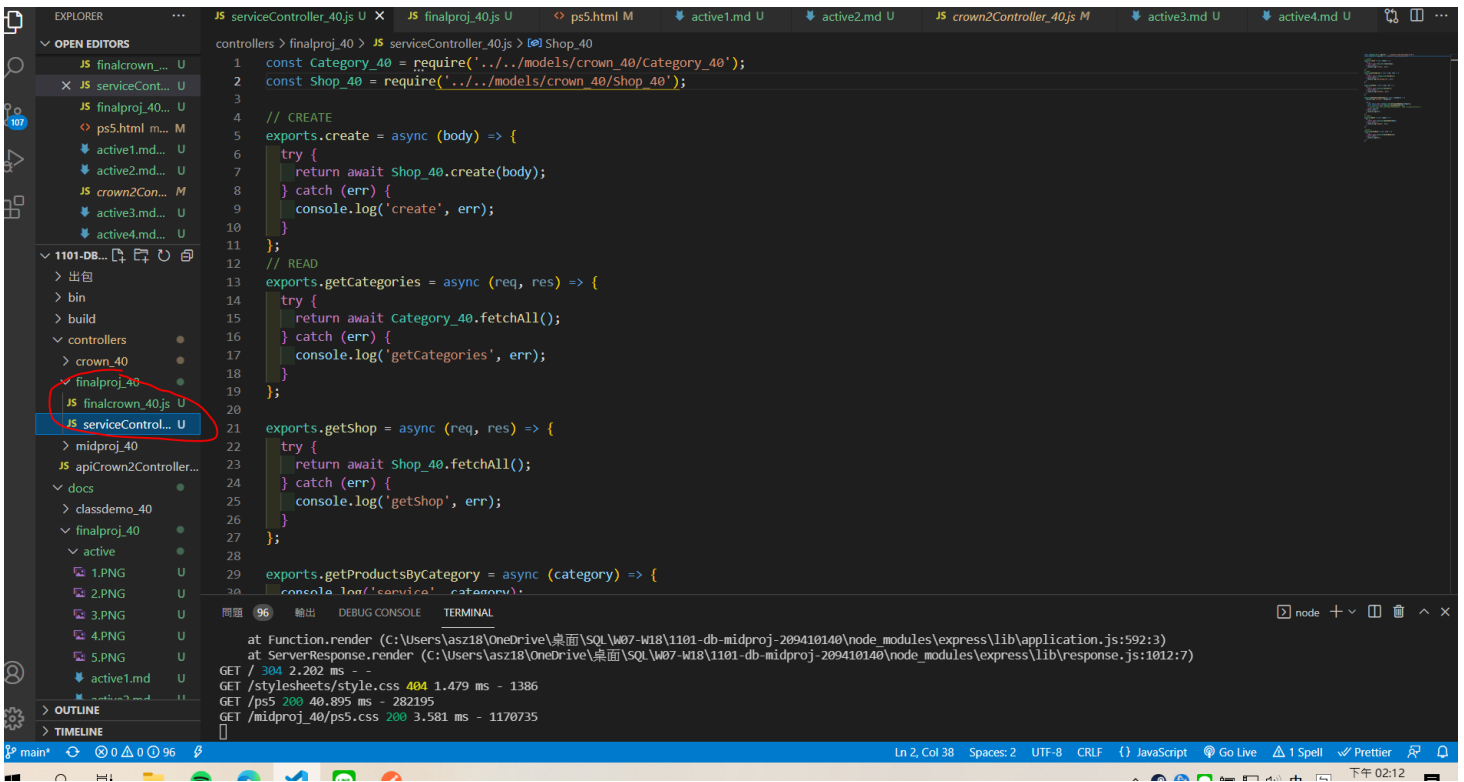
```
38 <a class="nav-link">Game</a>
39 </li>
40 <li class="nav-pill ml-auto">
41 <a class="nav-link active" href="/books_40/create">Add games</a>
42 </li>
43 </ul>
44 </div>
45 <div class="card-body">
46 <% if(data.length) { %>
47 <table class="table">
48 <thead>
49 <tr>
50 <th scope="col">Id</th>
51 <th scope="col">Name</th>
52 <th scope="col">Price</th>
53 <th width="200px">Action</th>
54 </tr>
55 </thead>
56 <tbody>
57 <% for(var i=0; i< data.length; i++) { %>
58 <tr>
59 <th scope="row">
60 <%= (data[i].id) %>
61 </th>
62 <td>
63 <%= data[i].name%>
64 </td>
65 <td>
66 <%= data[i].price%>
```

The code editor also shows a terminal window with the following output:

```
問題 94 輸出 DEBUG CONSOLE TERMINAL
at Function.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\application.js:592:3)
at ServerResponse.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\response.js:1012:7)
GET / 304 2.202 ms - -
GET /stylesheets/style.css 404 1.479 ms - 1386
GET /ps5 200 40.895 ms - 282195
GET /midproj_40/ps5.css 200 3.581 ms - 1170735
```

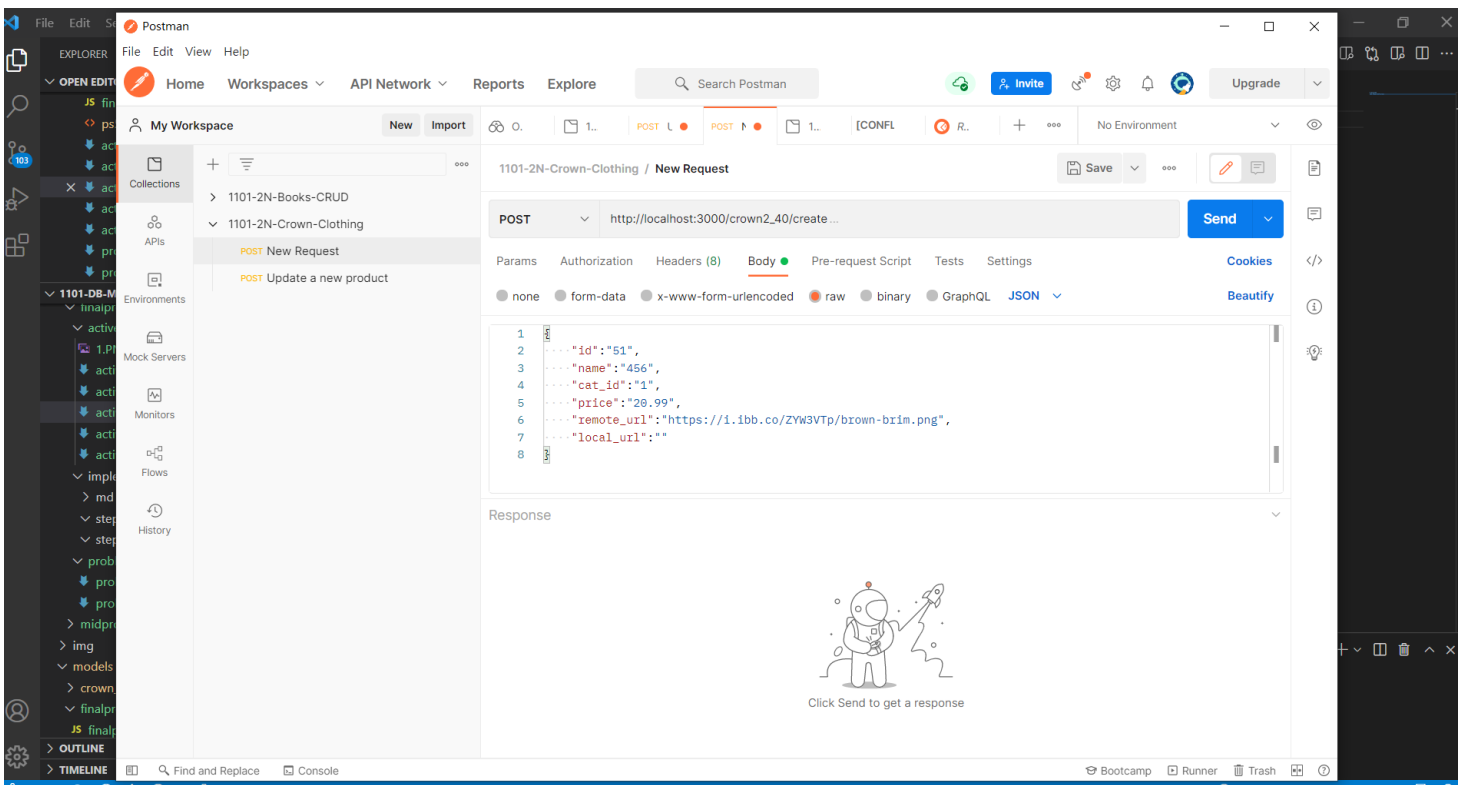
Active2

把**controllers**等**const**設定好，雖然沒弄好，但起碼有試過也跑過流程，加深印象。



Active3

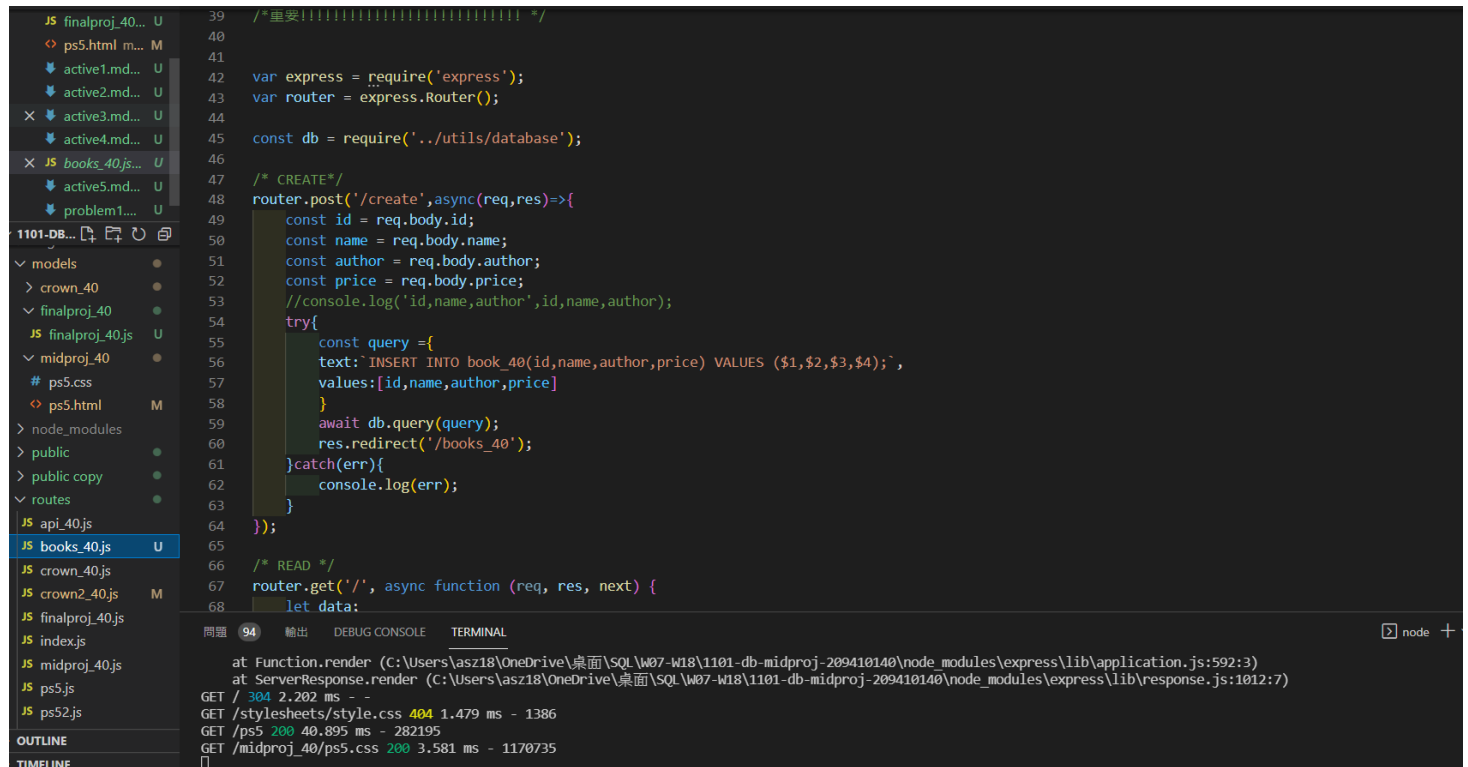
重新研究Postman，想知道它的運作邏輯，但也不確定自己有沒有看懂。



Active4

重新看過老師的上課影片，加深印象以對付考試。

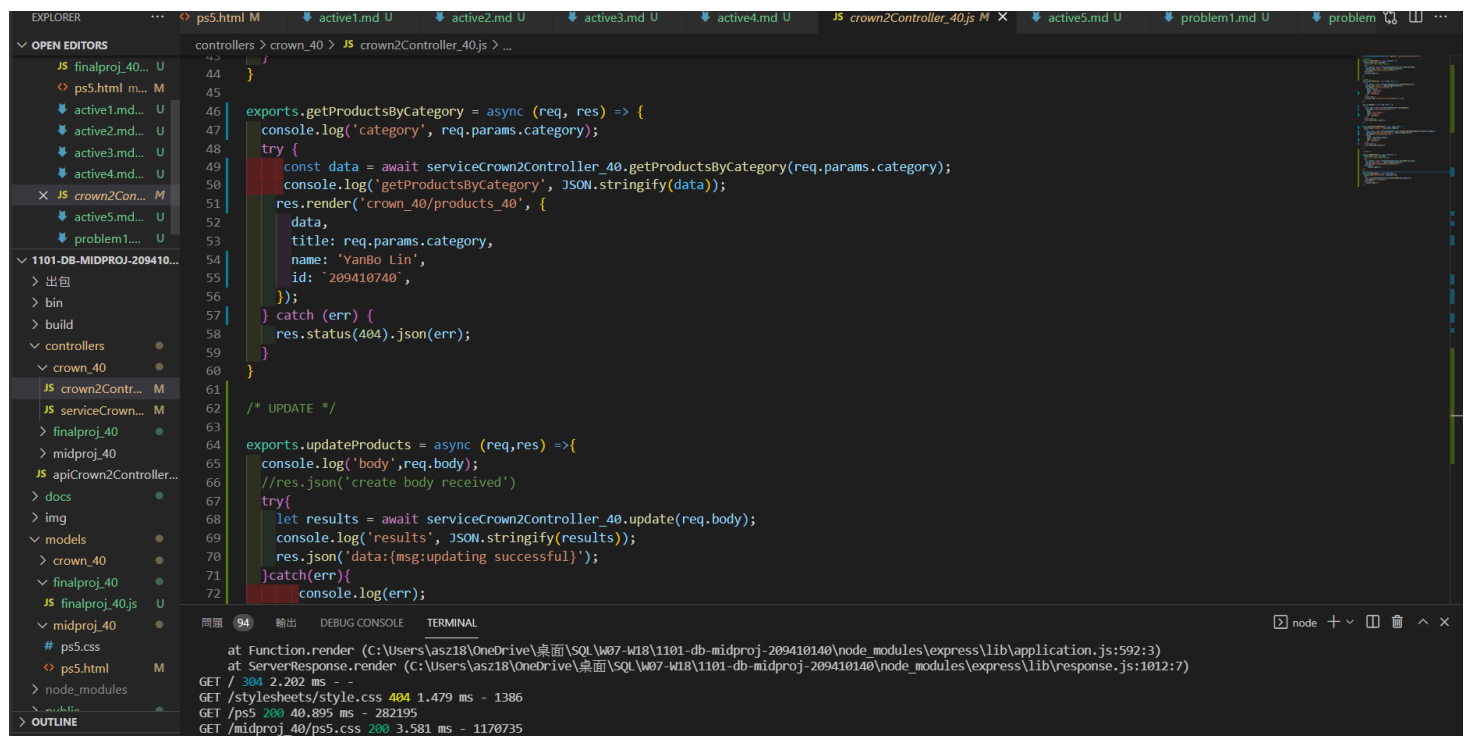
這一區花很多時間，但還是沒有完全搞懂，感覺不妙。



```
JS finalproj_40... U
ps5.html m... M
active1.md... U
active2.md... U
X active3.md... U
active4.md... U
X JS books_40.js... U
active5.md... U
problem1... U
1101-DB...
models
  crown_40
  finalproj_40
JS finalproj_40.js U
midproj_40
  ps5.css
  ps5.html M
node_modules
public
public copy
routes
JS api_40.js
JS books_40.js U
JS crown_40.js
JS crown2_40.js M
JS finalproj_40.js
index.js
midproj_40.js
ps5.js
ps52.js
OUTLINE
TIMELINE
```

```
39 /*重要!!!!!!!!!!!!!!!!!!!!!! */
40
41
42 var express = require('express');
43 var router = express.Router();
44
45 const db = require('../utils/database');
46
47 /* CREATE*/
48 router.post('/create', async (req, res) => {
49   const id = req.body.id;
50   const name = req.body.name;
51   const author = req.body.author;
52   const price = req.body.price;
53   //console.log('id,name,author', id, name, author);
54   try {
55     const query = {
56       text: 'INSERT INTO book_40(id,name,author,price) VALUES ($1,$2,$3,$4);',
57       values: [id, name, author, price]
58     }
59     await db.query(query);
60     res.redirect('/books_40');
61   } catch (err) {
62     console.log(err);
63   }
64 });
65
66 /* READ */
67 router.get('/', async function (req, res, next) {
68   let data;
```

```
at Function.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\application.js:592:3)
at ServerResponse.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\response.js:1012:7)
GET / 304 2.202 ms - -
GET /stylesheets/style.css 404 1.479 ms - 1386
GET /ps5 200 40.895 ms - 282195
GET /midproj_40/ps5.css 200 3.581 ms - 1170735
```



```
EXPLORER
ps5.html M
active1.md U
active2.md U
active3.md U
active4.md U
JS crown2Controller_40.js M
active5.md U
problem1.md U
problem2.md U
OPEN EDITORS
controllers > crown_40 > JS crown2Controller_40.js > ...
JS finalproj_40... U
ps5.html m... M
active1.md... U
active2.md... U
active3.md... U
active4.md... U
X JS crown2Can... M
active5.md... U
problem1... U
1101-DB-MIDPROJ-209410...
  出包
  bin
  build
  controllers
    crown_40
    JS crown2Contr... M
    JS serviceCrown... M
  finalproj_40
  midproj_40
    JS apiCrown2Controller...
  docs
  img
  models
    crown_40
    finalproj_40
    JS finalproj_40.js U
  midproj_40
    ps5.css
    ps5.html M
  node_modules
  OUTLINE
  TIMELINE
```

```
44 }
45
46 exports.getProductsByCategory = async (req, res) => {
47   console.log('category', req.params.category);
48   try {
49     const data = await serviceCrown2Controller_40.getProductsByCategory(req.params.category);
50     console.log('getProductsByCategory', JSON.stringify(data));
51     res.render('crown_40/products_40', {
52       data,
53       title: req.params.category,
54       name: 'YanBo Lin',
55       id: '209410740',
56     });
57   } catch (err) {
58     res.status(404).json(err);
59   }
60 }
61
62 /* UPDATE */
63
64 exports.updateProducts = async (req, res) => {
65   console.log('body', req.body);
66   //res.json('create body received')
67   try {
68     let results = await serviceCrown2Controller_40.update(req.body);
69     console.log('results', JSON.stringify(results));
70     res.json('data:[msg:updating successful]');
71   } catch (err) {
72     console.log(err);
73   }
74 }
```

```
at Function.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\application.js:592:3)
at ServerResponse.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\response.js:1012:7)
GET / 304 2.202 ms - -
GET /stylesheets/style.css 404 1.479 ms - 1386
GET /ps5 200 40.895 ms - 282195
GET /midproj_40/ps5.css 200 3.581 ms - 1170735
```

Active5

最後就是複習SQL的語法，透過助教的上課影片和W3School。

複習過了就看腦子有沒有記住了。

W3School

The screenshot shows the W3School website's SQL section. The navigation menu on the left lists various SQL topics. The main content area displays the syntax for creating a unique index, a note about database-specific syntax, and an example of creating an index on the 'LastName' column of the 'Persons' table. The sidebar on the right features a 'Get certified' badge and a 'CODE GAME' section with a fox illustration.

SQL Like
SQL Wildcards
SQL In
SQL Between
SQL Aliases
SQL Joins
SQL Inner Join
SQL Left Join
SQL Right Join
SQL Full Join
SQL Self Join
SQL Union
SQL Group By
SQL Having
SQL Exists
SQL Any, All
SQL Select Into
SQL Insert Into Select
SQL Case
SQL Null Functions
SQL Stored Procedures
SQL Comments
SQL Operators

SQL Database
SQL Create DB
SQL Drop DB

```
CREATE INDEX index_name  
ON table_name (column1, column2, ...);
```

CREATE UNIQUE INDEX Syntax

Creates a unique index on a table. Duplicate values are not allowed:

```
CREATE UNIQUE INDEX index_name  
ON table_name (column1, column2, ...);
```

Note: The syntax for creating indexes varies among different databases. Therefore: Check the syntax for creating indexes in your database.

CREATE INDEX Example

The SQL statement below creates an index named "idx_lastname" on the "LastName" column in the "Persons" table:

```
CREATE INDEX idx_lastname  
ON Persons (LastName);
```

Get certified by completing a course today!

W3SCHOOL CERTIFIED

Get started

CODE GAME

Illustration of a fox wearing a hat.