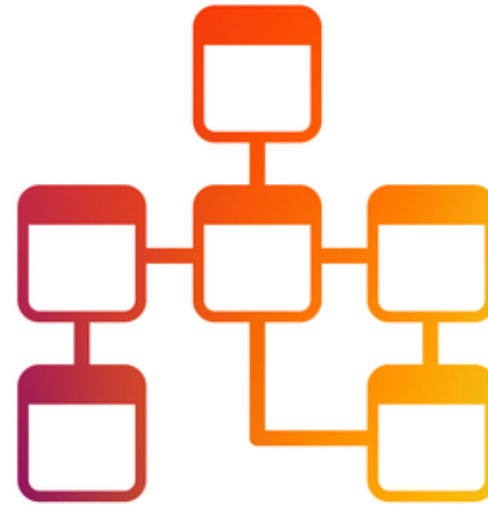


# Full-Stack Web Development



**FS1030 – Database Design and Principles**

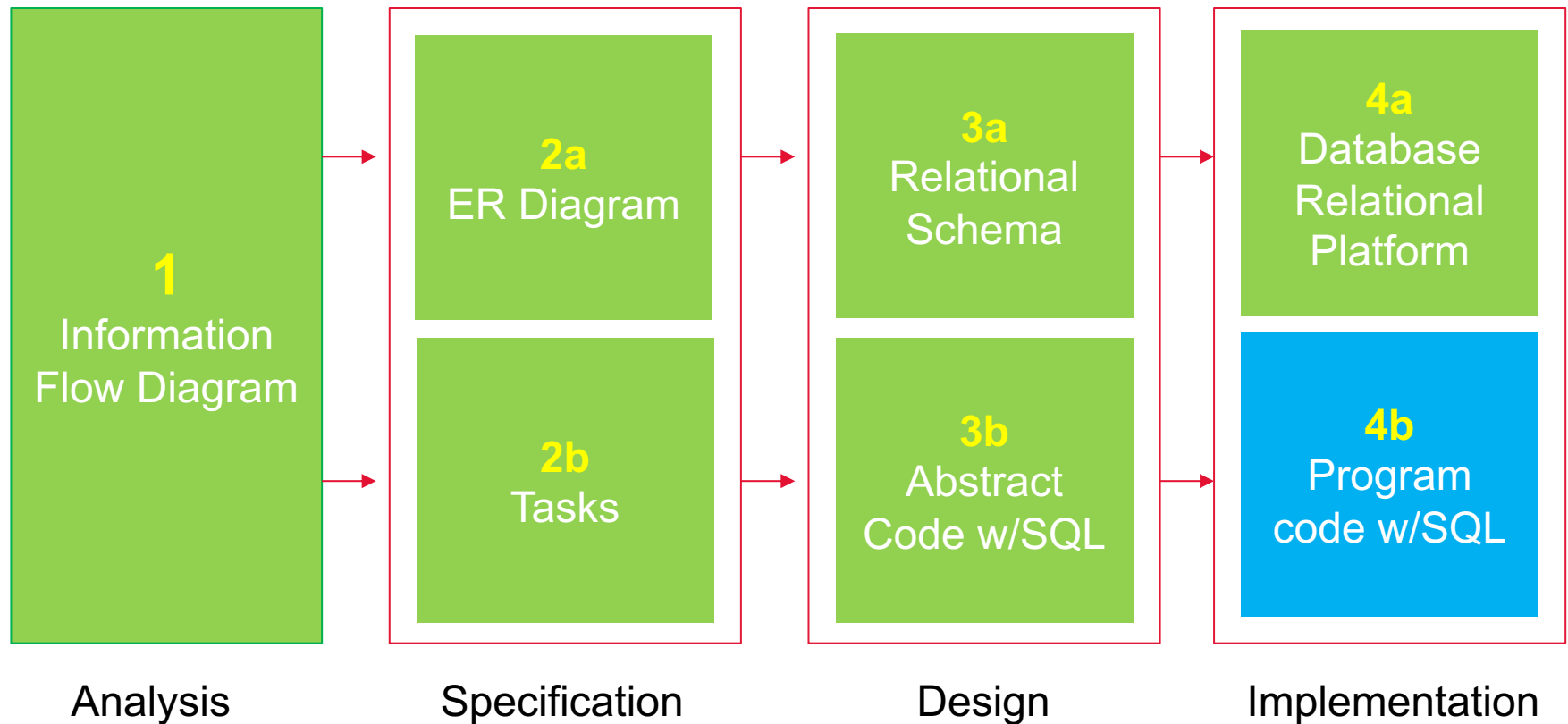
# What we learned so far

---

- Fundamentals of database
- Use cases around databases
- Why and when do we need various database types
- Data modeling and architecture
- Methodology
- Information flow diagrams
- Entity Relationship Diagrams
- Task decomposition
- Abstract Code
- Normalization Techniques
- Relational Algebra
- SQL Queries

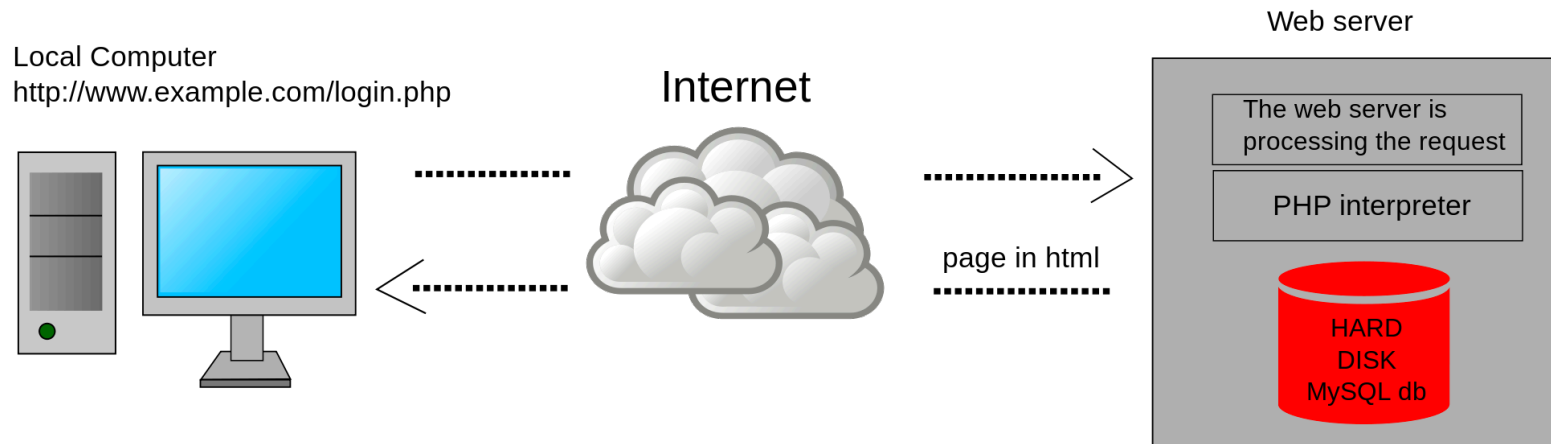
# What we are going to learn

---



# How server side languages work

---



# NodeJS + ExpressJS + MySQL

---

> Git clone <https://github.com/tarun27in/FS1030>

# NodeJS + ExpressJS + MySQL

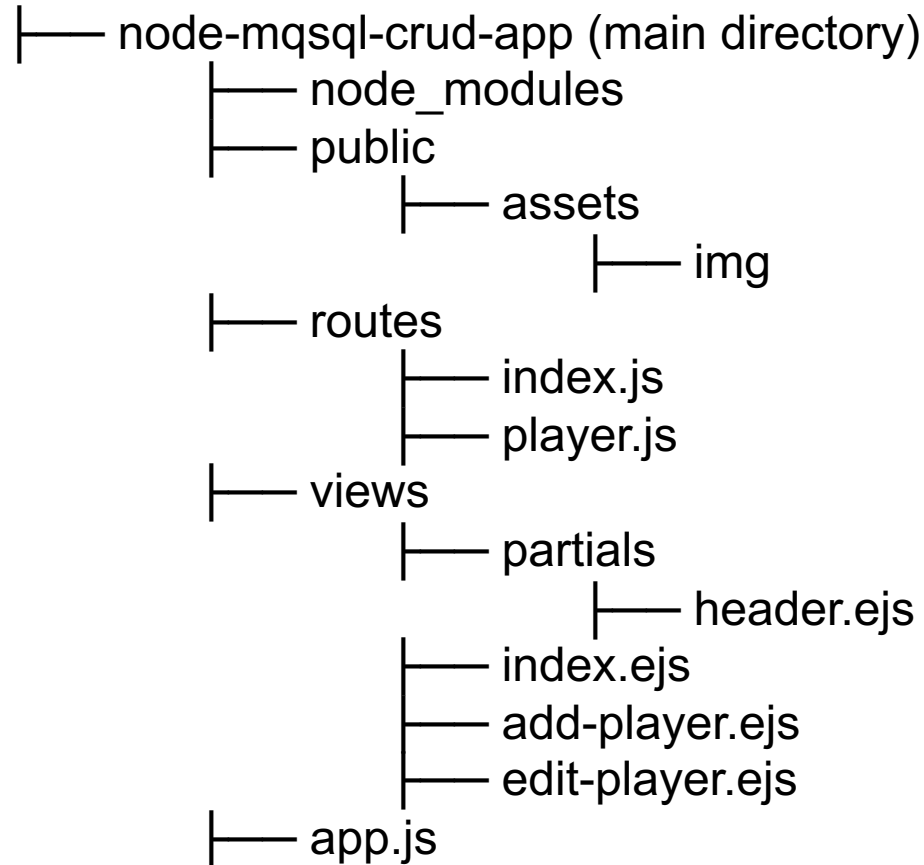
---

## Requirements

1. Node JS
2. XAMPP – Will be used for PHP as well
3. Understanding of EJS as templating engine
4. Visual Studio Code

# File Structure

---



# Modules Used

---

**express**: used to create handle routing and process requests from the client.

**express-fileupload**: Simple express file upload middleware that wraps around busboy.

**body-parser**: used to parse incoming request from the client.

**mysql**: Node JS driver for MySQL.

**ejs**: templating engine to render html pages for the app.

**req-flash**: used to send flash messages to the view

**nodemon**: Installed globally. It is used to watch for changes to files and automatically restart the server.



# Lets go through the files

---

header.ejs

The common header file for the application.

index.ejs

Homepage which contains the list of the players

add-player.ejs

Form to add a new player

edit-player.ejs

Form to edit a player

# Lets go through the files

---

## App.js

**mysql.createConnection** function takes in an object which contains the configuration of the database being connected to

## Routes

## index.js

The **db.query** function queries the database. It takes in the query and string and a callback which takes in two parameters, if the query is successful, the result is passed to the view in the **res.render** function.

## player.js

The **player.js** file is going to contain all the routes for the players page such as adding a player, updating a player's details and deleting a player.

# Exercises

---

- Extend the application to include two more fields in the form called "Ranking" and \$ value of the player
- Ranking will be select box (dropdown) with value from 1-10, 10-20, 20-50, 50-100, 100 and more
- \$ Value is a decimal value
- Do CRUD operation for those fields

---

## **Next Section: PHP & MySQL**

# PHP Basics

---

You need a webserver like Apache to run PHP.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
echo "Hello World";
```

```
?>
```

```
</body>
```

```
</html>
```

# PHP Basics - Variables

---

Variables hold the respective data values and cast.

```
<?php  
$txt = "Hello world!";  
$x = 5;  
$y = 10.5;  
?>
```

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
- Variable names are case-sensitive (\$age and \$AGE are two different variables)

# PHP Basics - Functions

---

Variables hold the respective data values and cast.

```
<?php  
$txt = "Hello world!";  
$x = 5;  
$y = 10.5;  
?>
```

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
- Variable names are case-sensitive (\$age and \$AGE are two different variables)

# PHP Basics – Global Variable

---

```
<?php
$x = 5; // global scope

function myTest() {
    // using x inside this function will generate an error
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();

echo "<p>Variable x outside function is: $x</p>";
?>
```



# PHP Basics - Print

---

```
<?php
$txt1 = "Learn PHP";
$txt2 = "FS1030";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```

# PHP Basics – Data Types

---

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

# PHP Basics – Data Types

---

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

# PHP Basics – Data Types

---

## Strings

```
<?php  
$x = "Hello world!";  
$y = 'Hello world!';
```

```
echo $x;  
echo "<br>";  
echo $y;  
?>
```

# PHP Basics – Data Types

---

## Integer

```
<?php  
$x = 5985;  
var_dump($x);  
?>
```

var\_dump returns the data type and value

## Float

```
<?php  
$x = 10.365;  
var_dump($x);  
?>
```

# PHP Basics – Data Types

---

## Array

```
<?php  
$cars = array("Volvo","BMW","Toyota");  
var_dump($cars);  
?>
```

# PHP Basics – Data Types

---

## Class & Objects

```
<?php
class Car {
    function Car() {
        $this->model = "VW";
    }
}
```

```
// create an object
$carobj = new Car();
```

```
// show object properties
echo $carobj ->model;
?>
```

# PHP Basics – Data Types

---

## Class & Objects

```
<?php
class Car {
    function Car() {
        $this->model = "VW";
    }
}
```

```
// create an object
$carobj = new Car();
```

```
// show object properties
echo $carobj ->model;
?>
```



# PHP Basics – If...else...

---

```
if (condition) {  
    code to be executed if this condition is true;  
} elseif (condition) {  
    code to be executed if this condition is true;  
} else {  
    code to be executed if all conditions are false;  
}
```

## Example

```
<?php  
$t = date("H");  
  
if ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>
```

# PHP Basics – Switch

---

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

# PHP Basics – Loops

---

## While Loop

```
<?php
$x = 1;

while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
} ?>
```

## Do While Loop

```
<?php
$x = 1;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
```

# PHP Basics – Loops

---

## For Loop

```
for (init counter; test counter; increment counter) {  
    code to be executed;  
}
```

## Example

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

```
<?php  
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as $value) {  
    echo "$value <br>";  
} ?>
```

# PHP Basics – Loops

---

## Foreach Loop

```
foreach ($array as $value) {  
    code to be executed;  
}
```

## Example

```
<?php  
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as $value) {  
    echo "$value <br>";  
} ?>
```

# PHP Basics – Functions

---

```
<?php
function familyName($fname) {
    echo "Welcome $fname.<br>";
}
```

```
familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
?>
```

```
<?php
function familyName($fname, $year) {
    echo "$fname Refsnes. Born in $year <br>";
}
```

```
familyName("Hege", "1975");
familyName("Stale", "1978");
familyName("Kai Jim", "1983"); ?>
```

# PHP Basics – Arrays

---

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
echo count($cars);
$arlength = count($cars);

for($x = 0; $x < $arlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```

# PHP Basics – Arrays

---

## Associative Arrays

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
echo count($cars);
$arrlength = count($cars);

for($x = 0; $x < $arrlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```



# PHP Basics – Forms

---

forms.php

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

welcome.php

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

# PHP Basics – Sessions

---

index.html

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

welcome.php

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

# PHP Basics – Forms

---

index.html

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

welcome.php

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

# PHP Basics – Sessions

---

## Set Session

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
```

```
</body>
</html>
```

# PHP Basics – Sessions

---

## Get Session

```
<?php
session_start();
?>

<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . "<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

# PHP + MySQL

---

> Git clone <https://github.com/tarun27in/FS1030php>

# AJAX

---

AJAX = Asynchronous JavaScript and XML.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Run <http://localhost/fs1030php/ajax/example.php> in your browser

# AJAX

---

## XMLHttpRequest Object Properties

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the <a href="#">Http Messages Reference</a>
statusText	Returns the status-text (e.g. "OK" or "Not Found")



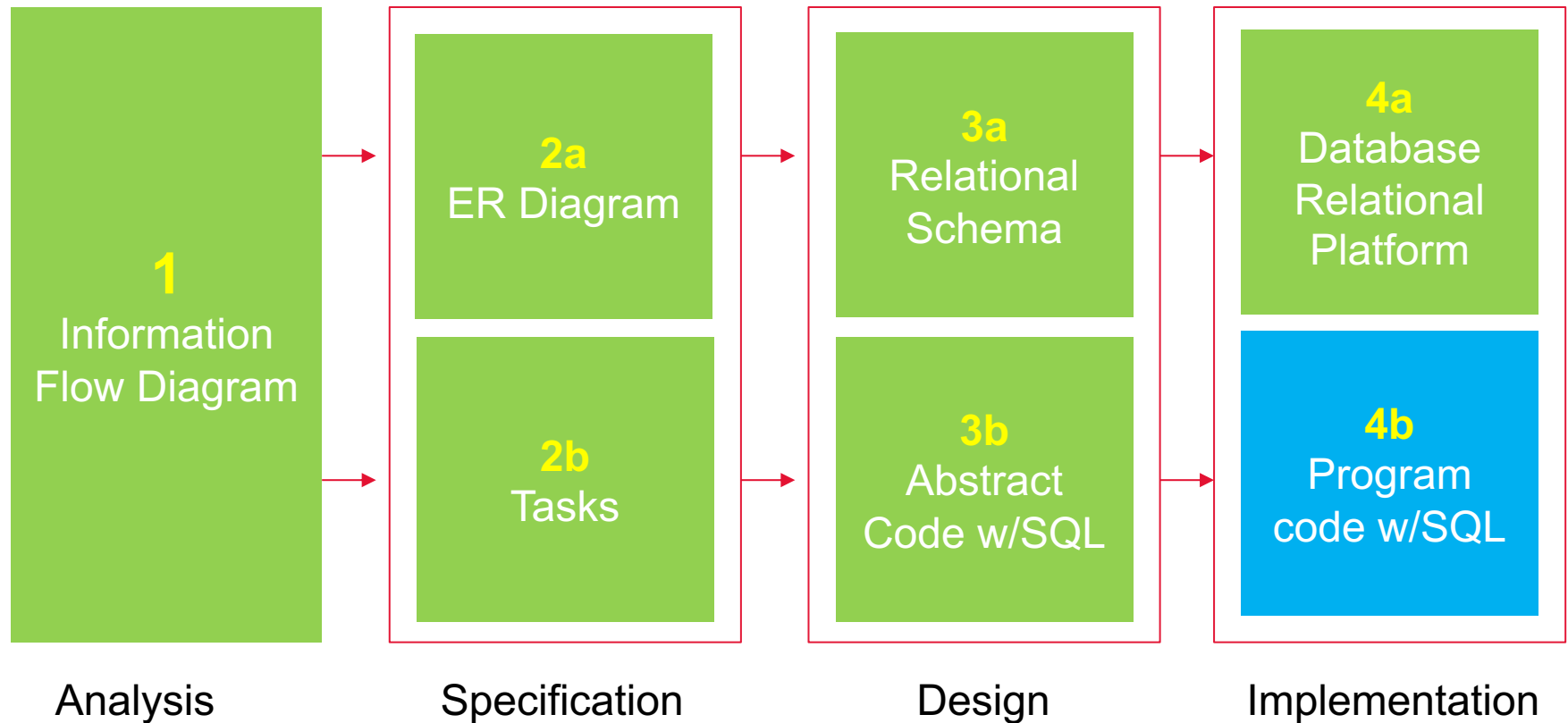
# Exercises

---

- Add 2 more fields in the employee form : department, position
- Department is a dropdown – add the departments you think are necessary
- Position is free text box
- Do CRUD operations

# What we learned today

---



# Thank you!

---

Submissions due before next class:

1. Group Project Phase #2 – Due before 17<sup>th</sup> June, 2019
2. Take Home Assignment #2 – Due before next class
3. Group Project Final Presentation and Code – Due on Sunday June 23