

### Question 1

#### Referring to the code from line 19 to line 21.

Explain in detail what this code does.

[8 Marks]

- Line 19 creates a local variable called `$SQL1` and assign to it an `INSERT INTO` query to create a new record in the orders table.
- The `INSERT INTO` SQL query inserts 2 data items in that table: the user id currently in the session and the system date and time.
- Line 21 executes the SQL query for our database connection which results in the actual insertion in the orders table provided no errors were made.

### Question 2

#### Referring to the code from line 19 to line 21.

Explain what `$_SESSION['c_userid']` is (line 20), where it gets its value from and how this value is used here.

[8 Marks]

- `$_SESSION['c_userid']` refers to the id of the user as stored in the session variable.
- This was created on the `login_process.php` when the user successfully logged into the system.
- The user ID was stored in this session variable so that it could be retrieved here on `checkout.php` to assign the right order to the right user.
- This is possible because `userid` is a Foreign Key (FK) in the orders child table that references the Primary Key (PK) `userid` of the Users parent table.
- Indeed, the tables orders and users are interconnected with a relationship with a cardinality of one-to-many. One user can place many orders, but an order is for one and only one user.
- Consequently, orders is the child table and carries `userId` as a FK that references the PK `userId` of its parent table Users.

### Question 3

#### Referring to the code from line 19 to line 21.

Explain what `$currentdatetime` is (line 20), where it gets its value from and how this value is used here.

[8 Marks]

- `$currentdatetime` is a local variable of type date.
- It stores the system date and time as currently captured by the PHP `date` function.
- It uses the format ('Y-m-d H:i:s') so that it can be accepted by the MySQL Database and inserted into the orders table.

### Question 4

#### Referring to the code from line 23 to line 31.

Explain in detail what this code does.

[8 Marks]

- Line 23 checks whether the SQL query error detector returns the value of zero. If it does, this means that the new order specified on line 19 and 20 was successfully added on line 21.
- Line 25 creates a local variable called `$SQL2` and assign to it an `SELECT` query to RETRIEVE the highest order number (max) i.e. the latest order number for the user currently logged in this session (as the orders table has an `auto_increment` on `orderNo`).
- The value of the `orderNo` for the new order just placed by the user currently logged in needs to be retrieved so that it can be used to allocate the order lines to this specific order just generated.

- This is because `orderNo` is a FK in the `order_line` that references `orderNo` as a PK in the `Orders` table.
- Indeed, the tables `orders` and `order_line` are interconnected with a relationship with a cardinality of one-to-many. One order has many order lines, but an order line is for one and only one order.
- Consequently, `order_line` is the child table and carries `orderId` as a FK that references the PK `orderId` of its parent table `orders`.
- On line 25 `Max (orderNo)` is aliased as `maxno` so that we can refer to it in the array of records `results2` created on line 28.

### Question 5

#### Referring to the code from line 23 to line 31.

Explain in detail what `$exeSQL2` and `$results2` are (lines 27 and 28) and how they are used here.

[8 Marks]

- `$exeSQL2` is an execution variable (a Boolean) that contains the outcome of the execution of the SQL query.
- `$results2` is an array of records that is populated using the `mysql_fetch_array` function.
- It contains one value (the highest order no for the logged in user) since only one attribute `max (orderNo)` and one row are retrieved by the SQL query. Only one attribute is listed in the SQL query and the query uses the MAX aggregating SQL function, which by definition always retrieves only one row.
- The value is assigned to a local variable `$latestorderno` on line 29 which is displayed as an order reference number on line 31.

### Question 6

#### Referring to the code from line 23 to line 31.

`maxno` appears on line 29 inside the square brackets. Should `orderNo` be used instead? Explain.

[8 Marks]

- `maxno` is an alias used in the query on line 25 to rename the attribute retrieved by the aggregating function `max (orderNo)`.
- It is necessary to use an alias when referencing the value in the square bracket when reading from the array of records.
- Line 29 refers to this alias in the statement `$results2['maxno']`.
- This is what will get the value of the highest order no and assign it to the local variable `$latestorderno` so that it can be used as a FK value in the INSERT INTO query on line 53.

### Question 7

#### Referring to the code from line 41 and from line 51 to line 55.

Briefly explain the code on line 41.

[8 Marks]

- The foreach loop on line 41 splits the session array between the key (referred to here as `$indexprodid`) and the value (referred to here as `$quantityinbasket`).
- This allows to read the values stored in the session array as key-value pair.
- For each iteration:
  - the id of the selected product gets retrieved and stored in `$indexprodid`.
  - the required number of items for the selected product gets retrieved and stored in `$quantityinbasket`.

### Question 8

#### Referring to the code from line 41 and from line 51 to line 55.

Explain what `$SQL4` is (line 51) and what it is used for.

[8 Marks]

- Line 51 creates a local variable called `$SQL4` and assign to it an INSERT INTO query to create a new record in the `order_line` table.

- For each key-value pair in the basket session array it inserts a new record in the order\_line table: the order no of the current order, the product id of the selected product, the number of items specified for the selected product and the sub total.

### Question 9

**Referring to the code from line 41 and from line 51 to line 55.**

Explain what `$latestorderNo`, `$indexprodid`, `$quantityinbasket` and `$subtotal` are (line 53 and 54), where they get their values from and how these values are used.

[8 Marks]

- `orderNo` gets the value of the number of the order retrieved from the database `$latestorderno` (see line 29). Therefore, `$latestorderno` is a local variable of type integer that contains the latest order no retrieved from the database on line 29.
- `prodId` gets its value from the key of the basket session array for this present iteration `$indeprodid`. Therefore, `$indexprodid` is a local variable of type integer that contains the key of the session array.
- The quantity ordered gets its value from the value stored inside basket the session array for this present iteration `quantityinbasket`. Therefore, `$quantityinbasket` is a local variable of type integer that contains the value of the session array.
- The subtotal gets the value from the calculation operated on line 49. Therefore, `$subtotal` is a local variable of type decimal that contains the result of the calculation  $\text{price} \times \text{required number of items}$ .

### Question 10

**Referring to the code from line 101 to line 121.**

Explain in detail what this code does and clearly describe the web page it generates.

[10 Marks]

- Line 107 creates a local variable of type string called `$SQL` and assigns to it a SQL query that retrieves all the details from album table ordered by album code.
- Line 108 executes this query using the `mysqli_query` function for our database connection and the output of this execution is stored in the local variable `$exeSQL`.
- This query multiple records i.e. as many as there are albums.
- Line 110 creates an array of records called `$results`, populates it with the result of the execution of the SQL query using the `mysqli_fetch_array` function and iterates through it.
- For every iteration through the array of records `$results` it will display the following:
  - The album title as an anchor with a query string `u_id` attached to it that passes through the album code using the GET method.
  - The artist as plain text.
  - The year of release as plain text.

These 3 elements are separated by `||` for formatting purposes.

This is a quick illustration of the output:

```
The Wall?u_id=354 || Pink Floyd || 1979
Wu-Tang Forever?u_id=770 || Wu-Tang Clan || 1997
Abbey Road?u_id=912 || The Beatles || 1969
```

### Question 11

**Referring to the code on line 112.**

A reference to **followingpage.php** (`href=followingpage.php`) appears on line 112 of the **initialpage.php**.

Write the PHP code for the **followingpage.php** file according to the specification below:

When clicking on a link located on `initialpage.php`, the user should be able to access `followingpage.php` which should read from the database table `Album_Copy` and display a list of items related to the one item selected by the user on `initialpage.php`.

[10 Marks]

### 1<sup>st</sup> Answer: Two separate queries

```
$albumCode = $_GET['u_id'];

$SQL1 = " SELECT albumTitle, artist, yearOfRelease
        FROM Album
        WHERE albumCode = ".$albumCode;

$exeSQL1 = mysqli_query($conn,$SQL1) or die (mysqli_error($conn));

$arraya = mysqli_fetch_array($exeSQL1);

echo "<p>Album Title: ".$arraya['albumTitle']. "</p>";
echo "<p>Artist: ".$arraya['artist']. "</p>";
echo "<p>Year Of Release: ".$arraya['yearOfRelease']. "</p>";

$SQL2 = " SELECT  copyCode, purchaseDate, condition
        FROM Album_Copy
        WHERE albumCode = ".$albumCode;

$exeSQL2 = mysqli_query($conn,$SQL2) or die (mysqli_error($conn));

while ($arrayc = mysqli_fetch_array($exeSQL2))
{
    echo "<p>Copy code: ".$arrayc['copyCode']. "</p>";
    echo "<p>Purchase Date: ".$arrayc['purchaseDate']. "</p>";
    echo "<p>Condition: ".$arrayc['condition']. "</p>";
}
```

### 2<sup>nd</sup> Answer: One JOIN query

```
$albumCode = $_GET['u_id'];

$SQL = "SELECT a.albumTitle AS albumTitle, a.artist AS artist,
        a.yearOfRelease AS yor, ac.copyCode AS copyCode, ac.purchaseDate AS
        purchaseDate, ac.condition AS condition
        FROM Album a JOIN Album_Copy ac
        ON a.albumCode = ac.albumCode
        AND a.albumCode = ".$albumCode;

$exeSQL = mysqli_query($conn,$SQL) or die (mysqli_error($conn));

while ($array = mysqli_fetch_array($exeSQL))
{
    echo "<p>Album Title: ".$array['albumTitle']. "</p>";
    echo "<p>Artist: ".$array['artist']. "</p>";
    echo "<p>Year Of Release: ".$array['yor']. "</p>";

    echo "<p>Copy code: ".$array['copyCode']. "</p>";
    echo "<p>Purchase Date: ".$array['purchaseDate']. "</p>";
    echo "<p>Condition: ".$array['condition']. "</p>";
}
```

## Question 12

### General question.

Explain how JavaScript, traditionally a client-side scripting language, can be used for back-end Web development as part of the Node.js runtime environment.

[8 Marks]

- JavaScript was initially a browser client-scripting language. This means that it was hosted by the browser and could be used to:
  - Capture of user's input.
  - Make of HTTP requests.
  - Process instructions to output HTML and CSS.
- The runtime Node.js environment was introduced to allow JavaScript to
  - Be used as utility scripts to configure system and to read, write and manipulate files.
  - Send and receive data over the network.
  - Implement efficient web servers and attach application code to it.
  - Make and serve HTTP requests.
- This allowed JavaScript:
  - to be uncoupled from browser and to run on the server, including on cloud-based servers.
  - to use the same language both on the server and client.
  - to be used as part of several front-end frameworks (e.g., React, Vue.js, Angular) and back-end frameworks (e.g., Express.js, Next.js, Sails.js)
  - to be used as part of several stacks e.g., MEAN stack (MongoDB, Express, Angular, Node) and MERN stack (MongoDB, Express, React, Node).