

## DATABASHANTERING (RADERA)

### STEG 1

Skapa filen **remove.php** i huvudmappen. I denna fil ska det grafiska gränssnittet för radering utvecklas i form av ett formulär. Samtidigt behövs funktionalitet för att radera information. Skapa därför filen **delete.php** i mappen **assets/functions/**.

Utöver detta behöver vi en funktion för att hämta information om en enskild användare. Detta gör vi sedan tidigare med filen **select-id.php**.

Nya filer och mappar är **fetmarkerade**. I **notifications.index.php** ska vi samla all feedback som skrivs ut på **index.php**.

- assets/
  - config/
    - db.php
  - css/
    - album.css
    - all.min.css
    - bootstrap.min.css
  - functions/
    - **delete.php**
    - insert.php
    - select-id.php
    - select.php
    - update.php
  - images/
  - **includes/**
    - **notifications.index.php**
  - webfonts/
- add.php
- edit.php
- index.php
- **remove.php**

## STEG 2

Högerklicka på filen **edit.php** och välj **Clone**.

Ändra namnet på den nya filen till **remove.php** och öppna sedan filen genom att dubbelklicka på den.

**Överst på remove.php** ska det inkluderas funktionalitet för radering av användare och för att hämta information om en enskild användare.

Ange därför följande:

```
<?php
// Opens database connection
require_once 'assets/config/db.php';
// Delete information from database
require_once 'assets/functions/delete.php';
// Get specific information about user
require_once 'assets/functions/select-id.php';
?>
```

## STEG 3

Processen för att radera är enligt följande:

1. Hämta information om enskild användare (**select-id.php**).
2. Presentera denna för användaren (**remove.php**).
3. Erbjud möjligheten att radera informationen (**delete.php**).

Funktionalitet för att hämta information om en enskild användare har vi sedan tidigare i filen **select-id.php**.

## STEG 4

Nu är det dags att presentera denna i ett formulär. Öppna filen **remove.php** och börja med att **ändra <form> elementet** så att den pekar mot **remove.php** enligt följande:

- `<form action="remove.php" method="post">`

Ange därefter den hämtade informationen (**\$row**) inuti formuläret enligt följande:

```
<div class="form-group">
  <p>Är du säker på att du vill radera följande konto?</p>
  <p><?php echo ucwords($row['firstname']) . ' ' . $row['lastname']; ?></p>
</div>
```

## STEG 5

Skapa en knapp som heter **Radera** för att skicka formuläret enligt följande:

```
<button type="submit" class="btn btn-danger" name="delete">
  <i class="fas fa-trash-alt"></i> Radera
</button>
```

Skapa också en knapp som leder tillbaka till **index.php** om man ångrar sig. Knappen ser ut enligt följande:

```
<a href="index.php" class="btn btn-secondary">
  <i class="fas fa-chevron-circle-left"></i> Tillbaka
</a>
```

## STEG 6

När vi skickar detta formuläret så måste vi säga till databasen specifikt vem vi ska radera. Ett sätt är att skicka id-numret för användaren **gömt** i ett formulärfält.

Detta görs genom att lägga till följande direkt innan **</form>**:

- `<input type="hidden" name="id" value="<?php echo $row['id']; ?>">`

Testa nu att gå till `index.php` och klicka på en radera-länk. Informationen för en användare ska nu visas i formuläret.

Högerklicka på sidan `remove.php`, välj visa källkod och bekräfta att id-numret visas i slutet av formuläret.

## STEG 7

Öppna filen **delete.php** för att ange funktionalitet för att radera information från databasen.

```
<?php
// Checks whether delete button is pressed
if (isset($_POST['delete'])) {
    // Creates a query
    $sql = '
        DELETE FROM users
        WHERE id = :id
    ';
    // Prepares a query
    $stmt = $dbh->prepare($sql);
    // Connects form fields with db containers
    $stmt->bindValue(':id', $_POST['id']);
    // Sends query to database
    if ($stmt->execute()) {
        header('Location: ../../index.php?action=deleted');
        exit();
    }
}
?>
```

## STEG 8

Avsluta genom att skriva ut en **alert-ruta** på **index.php** som anger feedback för radering. Lägga till följande i **switch-satsen** från tidigare genomgång:

```
case 'deleted':
    echo '
        <div class="alert alert-danger">
            Posten har raderats från databasen!
        </div>
    ';
    break;
```

## STEG 9

Eftersom antalet notifikationer har blivit många så lyfter vi ut hela följande kod från **index.php** till **assets/includes/notifications.index.php**.

```
<?php
// Checks if an action is set
if (isset($_GET['action'])) {
    // Checks which action is set
    switch ($_GET['action']) {
        case 'inserted':
            echo '
                <div class="alert alert-success">
                    Posten har lagts till i databasen!
                </div>
            ';
            break;
        case 'updated':
            echo '
                <div class="alert alert-success">
                    Posten har uppdaterats i databasen!
                </div>
            ';
            break;
        case 'deleted':
            echo '
                <div class="alert alert-danger">
                    Posten har raderats från databasen!
                </div>
            ';
            break;
    }
}
?>
```

## STEG 10

Inkludera nu filen **notifications.index.php** i **index.php** på följande sätt:

```
<?php
    // Gets notifications for different actions
    require_once 'assets/includes/notifications.index.php';
?>
```

Testa funktionen så att den fungerar i sin helhet.