

UAS Basis Data

Aldi Maulana Iqbal – 2021080122

Jelaskan Perbedaan Antara OLAP, ROLAP, MOLAP, Dan DSS.

OLAP (Online Analytical Processing) adalah teknologi yang digunakan untuk mengekstrak informasi dari basis data multidimensional. ROLAP (Relational OLAP) menggunakan basis data relasional sebagai dasar untuk data OLAP, sementara MOLAP (Multidimensional OLAP) menggunakan struktur data multidimensional yang terpisah. DSS (Decision Support System) adalah sistem yang digunakan untuk membantu dalam pengambilan keputusan, sering kali dengan menggunakan teknologi OLAP.

Secara singkat, perbedaan antara OLAP, ROLAP, MOLAP, dan DSS adalah :

1. OLAP adalah teknologi yang digunakan untuk mengekstrak informasi dari basis data multidimensional.
2. ROLAP menggunakan basis data relasional sebagai dasar untuk data OLAP.
3. MOLAP menggunakan struktur data multidimensional yang terpisah.
4. DSS adalah sistem yang digunakan untuk membantu dalam pengambilan keputusan, sering kali dengan menggunakan teknologi OLAP.

Jelaskan Karakteristik Data Warehouse.

1. Data integration: Data warehouse merupakan kumpulan data yang diintegrasikan dari berbagai sumber, seperti sistem transaksi, sistem pemasaran, sistem produksi, dll.
2. Subject-oriented: Data warehouse diorientasikan pada subjek tertentu, seperti penjualan, inventori, atau pelanggan. Hal ini membuat data lebih mudah dipahami dan dianalisis.
3. Time-variant: Data warehouse menyimpan data dari periode waktu yang berbeda, sehingga memungkinkan analisis perkembangan data dari waktu ke waktu.
4. Non-volatile: Data di dalam data warehouse tidak berubah setelah ditambahkan atau diubah. Hal ini membuat data lebih stabil dan dapat diandalkan untuk analisis.
5. Read-only: Data warehouse hanya digunakan untuk membaca data, tidak untuk menulis atau mengubah data. Hal ini membuat data lebih aman dan terlindung dari risiko kesalahan atau kehilangan data.
6. Metadata: Data warehouse menyimpan informasi tentang data yang disimpan, seperti sumber data, tanggal pemuatan, dan definisi kolom. Ini membuat data lebih mudah dipahami dan digunakan.
7. Scalability: Data warehouse dapat ditingkatkan dengan mudah untuk menangani jumlah data yang semakin banyak.
8. Data quality: Data warehouse menyimpan data yang diperiksa dan disempurnakan untuk menjamin kualitas data yang tinggi untuk analisis yang akurat.

Secara Definisi, Data Warehouse Adalah Subject Oriented, Integrated, Time Variant, Nonvolatile, Dan Accessible. Jelaskan Definisi Dari Masing-Masing Point Tersebut.

1. Subject oriented: Data warehouse berfokus pada suatu subjek tertentu, seperti penjualan atau pelanggan, daripada menyimpan data secara umum.
2. Integrated: Data warehouse menyatukan data dari berbagai sumber yang berbeda dan menyimpannya dalam satu tempat yang terkoordinasi.
3. Time variant: Data warehouse menyimpan data dalam periode waktu tertentu, seperti bulanan atau tahunan, sehingga memungkinkan analisis perkembangan data dari waktu ke waktu.
4. Nonvolatile: Data yang disimpan dalam data warehouse tidak dapat diubah atau dihapus setelah ditambahkan, sehingga memungkinkan analisis data yang akurat dan konsisten.
5. Accessible: Data warehouse diatur sedemikian rupa sehingga mudah diakses oleh pengguna yang berbeda, seperti analis data atau manajer, untuk tujuan analisis dan pengambilan keputusan.

Apa Jenis Index Yang Dibuat Untuk Warehouse? Ilustrasikan Penggunaan Masing-Masing Index Dengan Sebuah Contoh.

1. Index B-Tree: digunakan untuk mengindeks data yang diurutkan secara alfabetis atau numerik. Contohnya, dalam sebuah warehouse, index B-Tree dapat digunakan untuk mengindeks nomor barang atau kode produk.
2. Index Hash: digunakan untuk mengindeks data yang tidak diurutkan secara alfabetis atau numerik. Contohnya, dalam sebuah warehouse, index Hash dapat digunakan untuk mengindeks lokasi gudang atau nomor seri barang.
3. Index Bitmap: digunakan untuk mengindeks data yang memiliki nilai Boolean (ya atau tidak). Contohnya, dalam sebuah warehouse, index Bitmap dapat digunakan untuk mengindeks barang yang tersedia atau tidak tersedia.
4. Index Spatial: digunakan untuk mengindeks data yang berbasis lokasi. Contohnya, dalam sebuah warehouse, index Spatial dapat digunakan untuk mengindeks lokasi barang dalam gudang.
5. Index Full-Text: digunakan untuk mengindeks data yang berbasis teks. Contohnya, dalam sebuah warehouse, index Full-Text dapat digunakan untuk mengindeks deskripsi produk atau nama barang.

Jelaskan Konsep Recovery System Database. Sebutkan Dan Jelaskan Kegagalan-Kegagalan Dalam Sebuah Sistem.

Recovery system database adalah sistem yang digunakan untuk mengembalikan kondisi database ke kondisi sebelum terjadi kegagalan. Kegagalan dalam sebuah sistem dapat terjadi karena berbagai alasan, seperti:

1. Hardware failure: Kegagalan pada perangkat keras seperti hard drive, memory, atau power supply.
2. Software failure: Kegagalan pada perangkat lunak seperti sistem operasi atau aplikasi yang digunakan.
3. Human error: Kegagalan yang disebabkan oleh kesalahan manusia, seperti salah mengoperasikan perangkat keras atau lunak.
4. Natural disaster: Kegagalan yang disebabkan oleh bencana alam seperti gempa bumi, tsunami, atau banjir.
5. Cyber attack: Kegagalan yang disebabkan oleh serangan cyber seperti hacking atau virus.

Recovery system database dapat digunakan untuk mengembalikan data yang hilang atau rusak sebagai akibat dari kegagalan-kegagalan tersebut. Beberapa metode yang digunakan dalam recovery system database diantaranya adalah backup dan restore, replication, dan failover.

Jelaskan Metode-Metode Keamanan Database.

Ada beberapa metode keamanan yang dapat digunakan untuk melindungi database dari ancaman keamanan yang berbeda, diantaranya:

1. Encryption: Proses mengubah data yang dapat dibaca menjadi data yang tidak dapat dibaca tanpa kunci dekripsi yang sesuai. Ini digunakan untuk melindungi data sensitif dari pengintaian atau pencurian.
2. Authentication: Proses untuk menverifikasi identitas pengguna atau aplikasi yang mengakses database. Ini digunakan untuk mencegah akses yang tidak sah ke data.
3. Access Control: Proses untuk mengatur hak akses pengguna atau aplikasi terhadap data yang tersimpan dalam database. Ini digunakan untuk mencegah akses yang tidak sah atau tidak diinginkan ke data.
4. Backup dan Restore: Proses untuk menyimpan salinan data yang dapat digunakan untuk mengembalikan data jika terjadi kegagalan atau kerusakan. Ini digunakan untuk menjaga integritas data.
5. Auditing: Proses untuk mencatat aktivitas yang terjadi dalam database. Ini digunakan untuk memantau aktivitas yang tidak sah atau mencurigakan dan mengidentifikasi sumber masalah keamanan.
6. Firewall: Proses untuk mengatur akses jaringan ke dan dari database. Ini digunakan untuk mencegah serangan jaringan yang tidak diinginkan.

7. Penetration Testing : Proses untuk menguji keamanan database dengan mencoba untuk meretas sistem secara tidak sah. Ini digunakan untuk menemukan celah keamanan dan meningkatkan keselamatan.

Semua metode ini harus digunakan secara bersama-sama untuk menciptakan lingkungan keamanan yang kuat dan memastikan bahwa data yang tersimpan dalam database tetap aman.

Jelaskan Perbedaan Privasi Dan Security/Keamanan. Jelaskan Security Dalam Konteks Sistem Basis Data.

Privasi adalah hak seseorang untuk mengontrol informasi tentang diri mereka sendiri, termasuk bagaimana informasi tersebut digunakan, dibagikan, dan dilindungi. Security atau keamanan, di sisi lain, adalah upaya untuk melindungi informasi dari akses yang tidak sah atau penyalahgunaan.

Dalam konteks sistem basis data, security berfokus pada upaya untuk melindungi data dari akses yang tidak sah atau penyalahgunaan. Hal ini dapat dilakukan dengan menggunakan teknologi dan prosedur seperti autentikasi, enkripsi, dan kontrol akses. Ini bertujuan untuk mencegah akses yang tidak sah ke data, seperti akses oleh pengguna yang tidak berwenang, atau penyalahgunaan data oleh pengguna yang berwenang.

Secara umum, privasi dan security/keamanan memiliki tujuan yang sama, yaitu untuk melindungi informasi dari akses yang tidak sah atau penyalahgunaan, namun privasi lebih mengarah pada aspek kontrol informasi oleh individu, sementara security/keamanan lebih mengarah pada aspek perlindungan informasi dari akses yang tidak sah.

Jelaskan Bagaimana Security Dicapai Pada Level Sistem Database?

Security pada level sistem database dicapai melalui beberapa cara, diantaranya:

1. Authentication: proses verifikasi identitas user yang akan mengakses sistem database. User harus menyediakan username dan password yang valid untuk dapat mengakses sistem.
2. Authorization: proses pemberian izin akses kepada user terhadap objek-objek dalam sistem database. Hal ini dilakukan untuk mencegah user yang tidak berwenang untuk mengakses data yang sensitif.
3. Encryption: proses pengubahan data menjadi kode yang tidak dapat dibaca oleh orang yang tidak berwenang. Ini dilakukan untuk mencegah data dari dicuri atau dibaca oleh orang yang tidak berwenang.
4. Audits: proses pemantauan aktivitas user dalam sistem database. Ini dilakukan untuk mengetahui siapa saja yang mengakses sistem dan apa yang dilakukan.

5. Backup and recovery: proses penyimpanan cadangan data untuk dapat digunakan kembali jika terjadi kerusakan atau kehilangan data. Ini dilakukan untuk mencegah kerugian data akibat kegagalan sistem.
6. Firewall: proses pemblokiran akses yang tidak diinginkan dari luar jaringan. Ini dilakukan untuk mencegah serangan dari luar jaringan dan menjaga keamanan sistem.

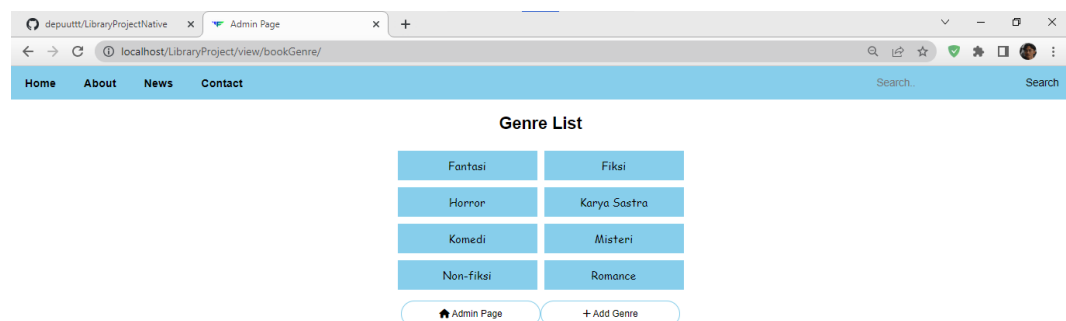
Secara keseluruhan, security pada level sistem database dilakukan untuk mencegah akses yang tidak diinginkan dari user yang tidak berwenang, menjaga integritas data, dan mencegah kerugian data.

Studi Kasus

Contoh studi kasus yang saya buat adalah sistem CRUD untuk genre buku, sistem ini merupakan bagian dari project personal saya dalam membuat sistem perpustakaan online.

<https://github.com/depuuttt/LibraryProjectNative>

Tampilan Home:



```
<?php

require '../model/bookGenre/function.php';

$genreDatas = takeAllData("SELECT * FROM book_genre ORDER BY
genre ASC");

?>
<!DOCTYPE html>
<html>
<head>
<title>Admin Page</title>
```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" type="text/css"
href="../../asset/css/bookGenre/style.css">
    <link rel="stylesheet" type="text/css"
href="../../asset/fontawesome/css/all.css">
    <link rel="icon" type="image/png"
href="../../asset/img/logo.png">
    <script src="https://kit.fontawesome.com/8460c0c2c8.js"
crossorigin="anonymous"></script>
</head>
<body>
    <header>
        <div id="nav" class="flex">
            <div class="btn-box flex">
                <a href="#">Home</a>
                <a href="#">About</a>
                <a href="#">News</a>
                <a href="#">Contact</a>
            </div>
            <div class="src-box flex">
                <input type="text" name="src" placeholder="Search..">
                <button type="button" name="button">Search</button>
            </div>
        </div>
    </header>
    <article>
        <div id="title">
            <h1>Genre List</h1>
        </div>
        <!-- <div id="memberTable">
            <table border="1">
                <tr>
                    <th>No.</th>
                    <th>Nama</th>
                    <th>Action</th>
                </tr>
                <?php $no = 1; ?>
                <?php foreach ($genreDatas as $genreData): ?>
                    <tr>
                        <td><?= $no++ ?></td>
                        <td><?= $genreData["nama_genre"] ?></td>
                        <td>
                            <a href="delete.php?id=<?= $genreData["id"]
?>">Delete</a>
                            <a href="update.php?id=<?= $genreData["id"]
?>">Update</a>

```

```

        </td>
    </tr>
    <?php endforeach; ?>
</table>
</div> -->
<div id="genreButton">
    <?php foreach ($genreDatas as $genreData): ?>
        <a href="detail.php?id=<?= $genreData['id'] ?>"><?=
$genreData["genre"] ?></a>
    <?php endforeach ?>
</div>
<div id="otherAct">
    <a href="../adminPage/">
        <button><i class="fa-solid fa-house"></i> Admin
Page</button>
    </a>
    <a href="create.php">
        <button><i class="fa-solid fa-plus"></i> Add
Genre</button>
    </a>
</div>
</article>
</body>
</html>

```

Model untuk menyambungkan ke database:

```

<?php

$host = "localhost";
$user = "root";
$pass = "";
$db = "db_05_xirpl_si_perpustakaan_aldi";

$conn = mysqli_connect($host, $user, $pass, $db);

function takeAllData($sql) {
    global $conn;
    $result = mysqli_query($conn, $sql);
    while ( $row = mysqli_fetch_assoc($result) ) {
        $allRow[] = $row;
    }
    return $allRow;
}

function takeOneData($sql) {
    global $conn;

```



```

$result = mysqli_query($conn,$sql);
$data = mysqli_fetch_assoc($result);
return $data;
}

function addData($data) {
    global $conn;
    $genreID = "gnr-".substr(uniqid(), 9);
    $genreName = $data["genreName"];
    $genreInfo = $data["genreInfo"];
    $sql = "INSERT INTO book_genre
            VALUES ('$genreID','$genreName','$genreInfo')";
    $result = mysqli_query($conn,$sql);
    return mysqli_affected_rows($conn);
}

function checkID($id) {
    global $conn;
    $sql = "SELECT * FROM book_genre WHERE id = '$id' ";
    $result = mysqli_query($conn, $sql);
    $totalData = mysqli_num_rows($result);
    if ($totalData > 0) {
        return true;
    } else {
        return false;
    }
}

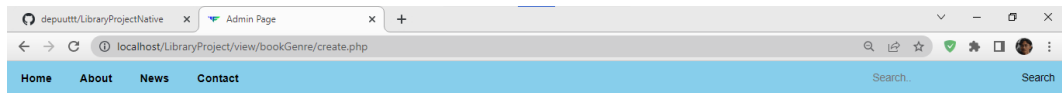
function deleteData($sql) {
    global $conn;
    $result = mysqli_query($conn, $sql);
    return mysqli_affected_rows($conn);
}

function updateData($data) {
    global $conn;
    $genreID = $data["genreID"];
    $genreName = $data["genreName"];
    $genreInfo = $data["genreInfo"];
    $sql = "UPDATE book_genre
            SET genre = '$genreName',
                description = '$genreInfo'
            WHERE id = '$genreID'";
    $result = mysqli_query($conn,$sql);
    return mysqli_affected_rows($conn);
}

```

?>

Tampilan Tambah Data Genre:



Add Genre Data

Genre Name
Genre Information
Submit Query

```
<?php

require '../model/bookGenre/function.php';

if (isset($_POST["submit"])) {
    $added = addData($_POST);
    if ($added > 0) {
        echo "
        <script>
            alert('Genre Data Added');
            document.location.href='../bookGenre/';
        </script>";
    } else {
        echo "
        <script>
            alert('Error While Adding Genre Data');
            document.location.href='../bookGenre/';
        </script>";
    }
}

?>

<!DOCTYPE html>
<html>
<head>
    <title>Admin Page</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

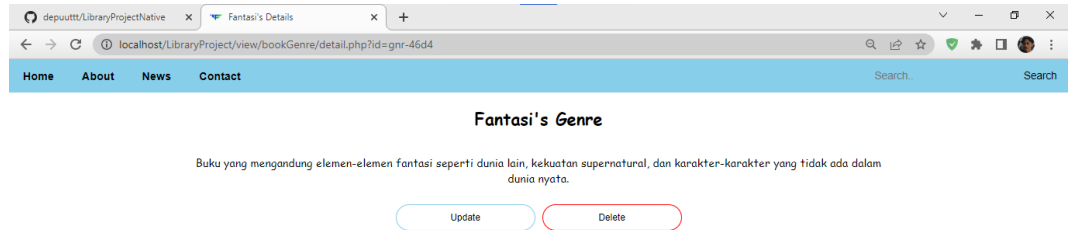
```

    <link rel="stylesheet" type="text/css"
href="../../asset/css/bookGenre/create.css">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.2.0/css/all.css"
integrity="sha384-
hWvjflwFxF6sNzntih27bfXkr27PmbbK/iSvJ+a4+0owXq79v+lsFkW54bOGbiD
Q" crossorigin="anonymous">
    <link rel="icon" type="image/png"
href="../../asset/img/logo.png">
</head>
<body>
    <header>
        <div id="nav" class="flex">
            <div class="btn-box flex">
                <a href="#">Home</a>
                <a href="#">About</a>
                <a href="#">News</a>
                <a href="#">Contact</a>
            </div>
            <div class="src-box flex">
                <input type="text" name="src" placeholder="Search..">
                <button type="button" name="button">Search</button>
            </div>
        </div>
    </header>
    <article>
        <div id="title">
            <h1>Add Genre Data</h1>
        </div>
        <div id="form">
            <form method="post">
                <div class="genreName">
                    <input type="text" name="genreName"
placeholder="Genre Name">
                </div>
                <div class="genreInfo">
                    <textarea name="genreInfo">Genre
Information</textarea>
                </div>
                <div class="memberSubmit">
                    <button type="submit" name="submit">Submit
Query</button>
                </div>
            </form>
        </div>
    </article>
</body>

```

</html>

Tampilan Detail Genre Buku:



```
<?php
```

```
require '../..../model/bookGenre/function.php';

if (checkID($_GET["id"])) {
    $id = $_GET["id"];
} else {
    echo "
    <script>
        alert('Unknown Genre Data');
        document.location.href='../bookGenre/';
    </script>";
}

$genreData = takeOneData("SELECT * FROM book_genre WHERE
id='$id'");

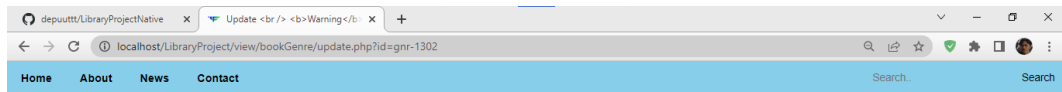
?>
<!DOCTYPE html>
<html>
<head>
    <title><?= $genreData["genre"] ?>'s Details</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```

    <link rel="stylesheet" type="text/css"
href="../../../asset/css/bookGenre/detail.css">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.2.0/css/all.css"
integrity="sha384-
hWvjflwFxFxL6sNzntih27bfxkr27PmbbK/iSvJ+a4+0owXq79v+lsFkW54bOGbiD
Q" crossorigin="anonymous">
    <link rel="icon" type="image/png"
href="../../../asset/img/logo.png">
</head>
<body>
    <header>
        <div id="nav" class="flex">
            <div class="btn-box flex">
                <a href="#">Home</a>
                <a href="#">About</a>
                <a href="#">News</a>
                <a href="#">Contact</a>
            </div>
            <div class="src-box flex">
                <input type="text" name="src" placeholder="Search..">
                <button type="button" name="button">Search</button>
            </div>
        </div>
    </header>
    <article>
        <div id="title">
            <h1><?= $genreData["genre"] ?>'s Genre</h1>
        </div>
        <div id="genreInfo">
            <p><?= $genreData["description"] ?></p>
        </div>
        <div id="otherAct">
            <a href="update.php?id=<?= $genreData['id']
?>"><button>Update</button></a>
            <a href="delete.php?id=<?= $genreData['id'] ?>"><button
class="warning">Delete</button></a>
        </div>
    </article>
</body>
</html>

```

Tampilan Update Data Buku:



```
<?php

require '../..../model/bookGenre/function.php';

if (checkID($_GET["id"])) {
    $id = $_GET["id"];
    $genreData = takeOneData("SELECT * FROM book_genre WHERE
id='$id'");
} else {
    echo "
    <script>
        alert('Unknown Genre ID');
        document.location.href='../bookGenre/';
    </script>";
}

if (isset($_POST["submit"])) {
    $updated = updateData($_POST);
    if ($updated > 0) {
        echo "
        <script>
            alert('Genre Data Updated');
            document.location.href='../bookGenre/';
        </script>";
    } else {
        echo "
        <script>
            alert('Error While Updating Genre Data');
            document.location.href='../bookGenre/';
        </script>";
    }
}
```

```

        </script>";
    }
}

?>
<!DOCTYPE html>
<html>
<head>
    <title>Update <?= $genreData["nama_genre"] ?>'s Genre</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" type="text/css"
href="../../asset/css/bookGenre/update.css">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.2.0/css/all.css"
integrity="sha384-
hWvjflwFxL6sNzntih27bfxkr27PmbbK/iSvJ+a4+0owXq79v+lsFkW54bOGbiD
Q" crossorigin="anonymous">
    <link rel="icon" type="image/png"
href="../../asset/img/logo.png">
</head>
<body>
    <header>
        <div id="nav" class="flex">
            <div class="btn-box flex">
                <a href="#">Home</a>
                <a href="#">About</a>
                <a href="#">News</a>
                <a href="#">Contact</a>
            </div>
            <div class="src-box flex">
                <input type="text" name="src" placeholder="Search..">
                <button type="button" name="button">Search</button>
            </div>
        </div>
    </header>
    <article>
        <div id="title">
            <h1>Update Genre Data</h1>
        </div>
        <div id="form">
            <form method="post">
                <div class="genreID">
                    <input type="hidden" name="genreID" value="<?=
$genreData['id'] ?>" readonly>
                </div>
                <div class="genreName">

```

```

        <input type="text" name="genreName" value="<?=$genreData['genre'] ?>">
    </div>
    <div class="genreInfo">
        <textarea name="genreInfo"><?=$genreData["description"] ?></textarea>
    </div>
    <div class="memberSubmit">
        <button type="submit" name="submit">Submit
Query</button>
    </div>
</form>
</div>
</article>
</body>
</html>

```

Kode untuk menghapus buku:

```

<?php

require '../..../model/bookGenre/function.php';

if (checkID($_GET["id"])) {
    $id = $_GET["id"];
} else {
    echo "
    <script>
        alert('Unknown Member ID');
        document.location.href= '../bookGenre';
    </script>
    ";
}

$deleted = deleteData("DELETE FROM book_genre WHERE
id='$id'");
if ($deleted > 0) {
    echo "
    <script>
        alert('Delete Succesful');
        document.location.href= '../bookGenre';
    </script>
    ";
} else {
    echo "
    <script>
        alert('Delete Unsuccesful');
        document.location.href= '../bookGenre';
    </script>
    ";
}

```



```
</script>  
";  
}  
?>
```