 UTeM اونیورسیتی تکنیکل ملیسیا ملاک UNIVERSITI TEKNIKAL MALAYSIA MELAKA			FAKULTI TEKNOLOGI KEJURUTERAAN ELEKTRIK DAN ELEKTRONIK		
DATABASE MANAGEMENT SYSTEM					
BEEC 1343		SEMESTER 2		SESI 2020/2021	
Design and Development of DVD Rental System					
NAME OF GROUP MEMBERS & MATRIX NUMBER	1. RAHMAN KAZI ASHIKUR			B081910450	
	2. HAMSAPRIYAVATY A/P MANICKAM			B081910237	
	3. NIVEKHA A/P RAMADAS			B081910470	
COURSE		2BEEC			
DATE		25/6/2021			
NAME OF INSTRUCTOR		AHMAD FAIRUZ BIN MUHAMMAD AMIN			
EXAMINER'S COMMENT				VERIFICATION STAMP	

Table of Contents

1. OBJECTIVES .	3
2. EQUIPMENTS.....	3
3. INTRODUCTION	3
3.1 THE BENEFIT OF DATABASE	4
3.2 CONCEPTUAL DESIGN	4
3.3 LOGICAL DESIGN	5
3.4 NORMALIZATION	6
3.5 PHYSICAL DESIGN.....	8
4. DISCUSSIO	9
4.1 Entity-Relationship Diagram of our assignment system.....	9
4.2 Another example of ER diagram	11
4.3 A simple ER Diagram:	11
5 Explain about the SQL.	12
5.1 Create actors table.....	12
5.2 Dumping data for table `actors`	12
5.3 Indexes for table `actors`	12
5.4 Table structure for table `movies`	12
5.5 Dumping data for table `movies`	13
5.6 Indexes for table `movies`	13
5,7 Table structure for table `moviesactors`	13
5,8 Dumping data for table `moviesactors`	13
5.9 ALTER TABLE `moviesactors`	14
ALTER TABLE `moviesactors`	14
6. SELECT or read.	14
6.1 Movies Name Select Input Field	15
6.2 Update or input.....	15
6.3 movies Name Input Field	16
6.4 Movies to actors make relation field.	17
6.5 Delete movie or actor name.	18
7. RESULT	19
CONCLUSION.....	28
8. References	29

1. OBJECTIVES .

- To design and develop using MySQL, Php and Apache Web server (XAMPP)
- To construct a database with tables and Php for web interface using appropriate script and code.

2. EQUIPMENTS.

1. Apache Web Server (XAMPP)
2. Microsoft Visual Studio.
3. Web Browser (Chrome, Edge, Firefox)
4. Personal Computer or Laptop.

3. INTRODUCTION










A database Management System (DBMS) is a software system that allows users to define, build, manage, and regulate database access. End users may add, read, edit, and remove data in a database using DBMS. Database Management System offers more advantages in this sector than File Based Data Management System. The DBMS acts as a bridge between the user and the database. Because the database structure is stored as a collection of files, we may access the data in those files using the DBMS. All application requests are received by the DBMS, which then converts them into the complicated processes necessary to satisfy those demands. The database management system (DBMS) conceals most of the database's underlying complexity from application applications and users.

A database management system can help to decrease data by preventing numerous copies of the same file from being created, which leads to data redundancy. As a result, there is no possibility of meeting duplicate data. Following that, database users can exchange the data among themselves. There are several degrees of access to the data dependent on the right permission processes being followed. Others, the database privacy rule indicates that only authorized users can access the database in accordance with its privacy limitations. Finally, the Database Management System handles backup and recovery automatically. The user does not need to backup data on a regular basis because the DBMS handles this for them. Furthermore,

it restores the database to its former state following a crash or system failure. (Craig s.Mullins, 2005-20021)

3.1 THE BENEFIT OF DATABASE

Using database technology to gather, store and process information about your customers, suppliers and even competitors can give your business a distinct advantage.

-  A DBMS schedules concurrent access to data such that only one user may view the same data at the same time.
-  Data administration techniques that are consistent
-  DBMS provides a number of ways for storing and retrieving data.
-  Time spent developing applications has been reduced.
-  Application programmers are never exposed to data representation and storage specifics.
-  Provide data integrity and security
-  To achieve a high level of security against unauthorized data access, the DBMS employs integrity restrictions.
-  DBMS acts as an efficient handler, balancing the needs of various applications that use the same data.
-  A database management system (DBMS) employs a variety of sophisticated capabilities to effectively store and retrieve data. (Ireland, 2018)

3.2 CONCEPTUAL DESIGN

In this design process, the concept of constructing a data model is used in an enterprise, which is independent of any physical factors. The conceptual database design process begins with the development of an enterprise-wide conceptual data model that is independent of implementation specifics. For example, the target database management system, application program use, programming languages utilized, hardware platform, performance concerns, or any other physical factors.

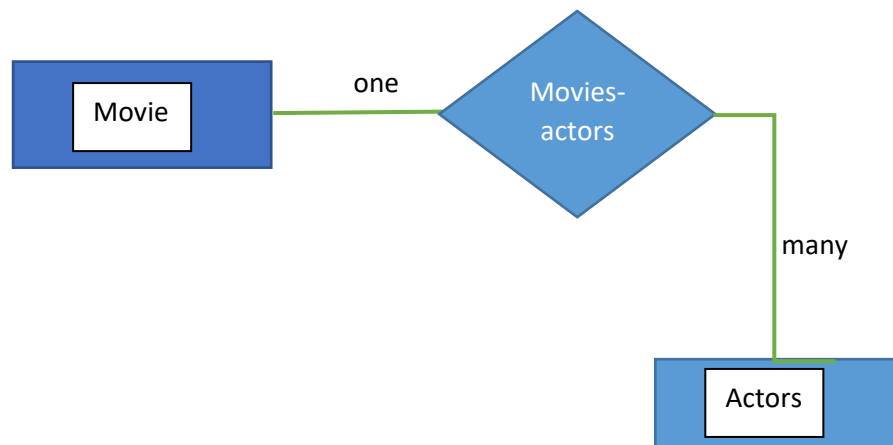


Figure: - Conceptual Design (CD)

3.3 LOGICAL DESIGN

A logical design is a conceptual, abstract design. ... The process of logical design involves arranging data into a series of logical relationships called entities and attributes. An entity represents a chunk of information. In relational databases, an entity often maps to a table.

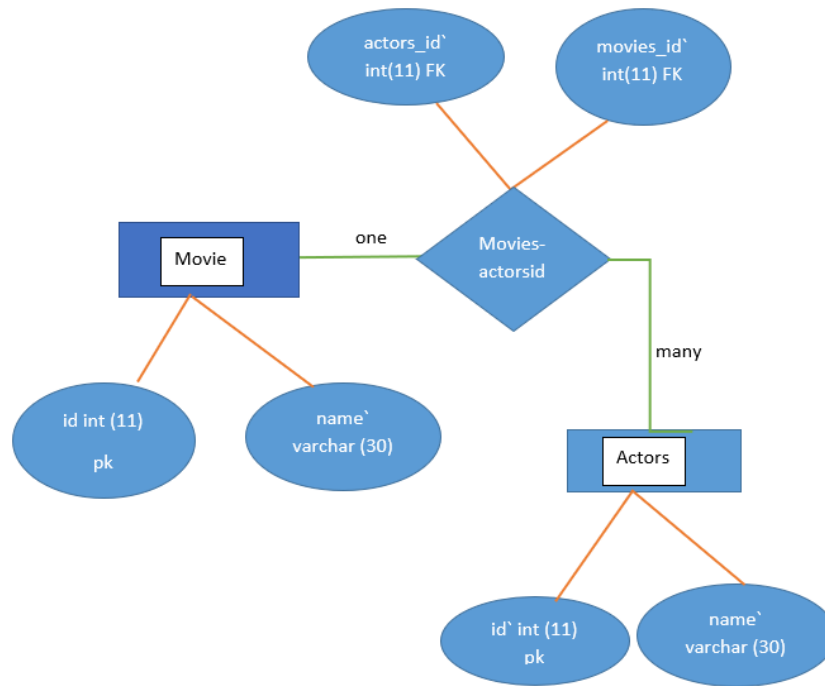


figure: logical design

The aim of the logical database design approach is to transform a conceptual data model into a logical data model, then authorize it to determine if it is architecturally sound and capable of enabling the needed transactions. The primary aim of this stage of the database development life cycle is to transform the conceptual data model produced in the previous chapter's conceptual approach into a logical data model of the enterprise's data requirements. The actions listed below can assist you in achieving this goal:

- ✚ Test the logical data model with a user.
- ✚ Integrate logical data models into the overall model (This step is an optional one)
- ✚ Obtain the logical data model's relations;
- ✚ Authorize those relations through normalization; and
- ✚ Validate those relations against user transactions.
- ✚ Examine the potential for future growth and development.
- ✚ Examine integrity control and its limitations.

3.4 NORMALIZATION

Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion

Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables. (warehousing, 2015)

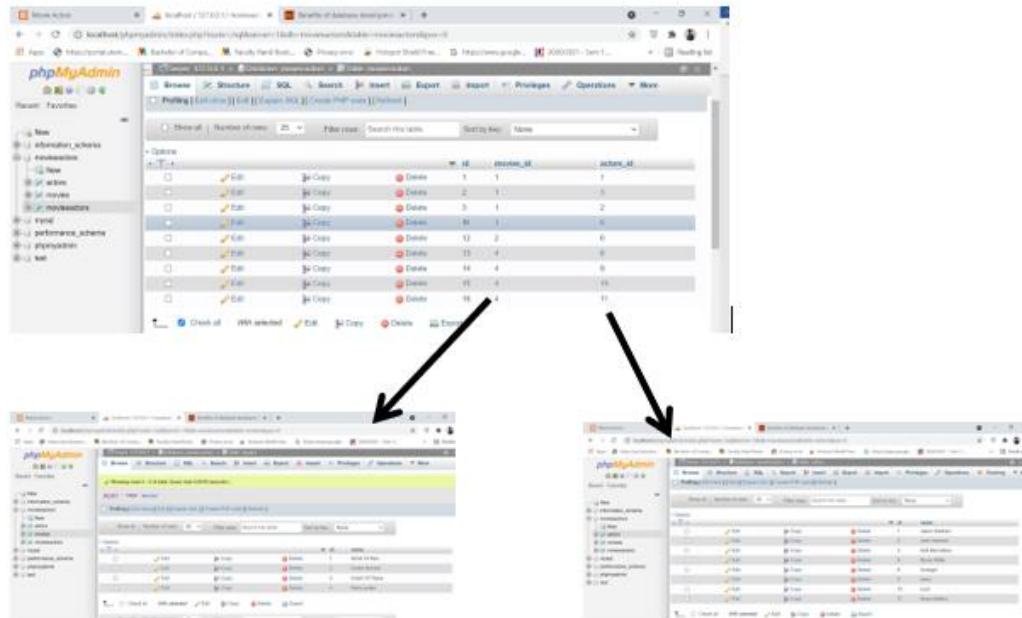


Figure: Normalization in DBMS: 1NF, 2NF.

When developing the schema of a relational database, one of the most important aspects to be taken into account is to ensure that the duplication is minimized. This is done for 2 purposes:

- Reducing the amount of storage needed to store the data.
- Avoiding unnecessary data conflicts that may creep in because of multiple copies of the same data getting stored.

Database design approach that avoids unwanted features such as Insertion, Update, and Deletion Anomalies and decreases data redundancy. Database normalization is a database schema design method that involves modifying an existing schema to reduce data redundancy and dependence. Normalization divides a huge table into smaller tables and defines relationships between them to improve data organization clarity. A database's normalization is accomplished by following a set of rules known as 'forms' while constructing the database.

- The terms normalization and normal form apply to the structure of a database.
- Normalization was created in the 1970s by IBM researcher E.F.Codd.

- ✚ Normalization improves the clarity of data organization in databases.

3.5 PHYSICAL DESIGN

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output. ... It is concerned with user interface design, process design, and data design. (warehousing, 2015)

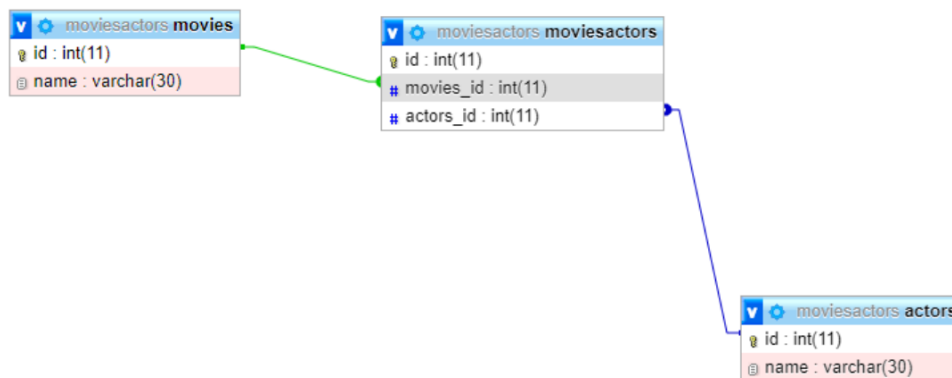


Figure: - Physical Database design

It is the process of establishing the database's secondary storage execution, including the basic relations, file organization, and indexes required to guarantee efficient data access, as well as any related integrity restrictions and security measures.

- ✚ Transform the logical data model for the target DBMS
- ✚ Create base relations
- ✚ Create derived data representation
- ✚ General design limitations
- ✚ Create file organization and indexes
- ✚ Examine transactions
- ✚ Select file classifications
- ✚ Select indexes
- ✚ Calculate storage space needs

- ✚ Create user views
- ✚ Create security measures
- ✚ Consider using controlled redundancy
- ✚ Maintain and fine-tune the operational system

Common Characteristics of a Physical Data Model

- ✚ It usually depicts the data requirements for a specific project or application. Sometimes it's even a component of an application.
- ✚ Through the use of a repository of common entities, it is possible to combine it into other physical data models.
- ✚ It generally has 10-1000 tables however this number varies greatly depending on the extent of the data model.
- ✚ It includes table relationships that handle the cardinality and null ability (optionality) of the connections.
- ✚ Designed and built to be dependent on a certain version of a DBMS, data storage location, or technology.
- ✚ Data types with precise precisions and lengths will be allocated to database columns. Null ability (optional) will be applied to columns.
- ✚ Definitions for tables and columns will be provided.

4. DISCUSSIO

4.1 Entity-Relationship Diagram of our assignment system

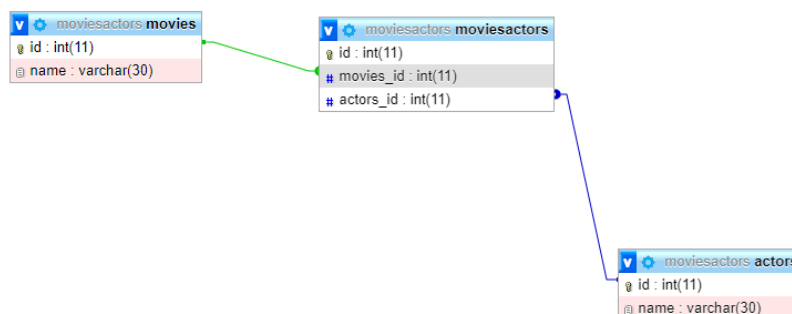


Figure: one to many relations ER diagram

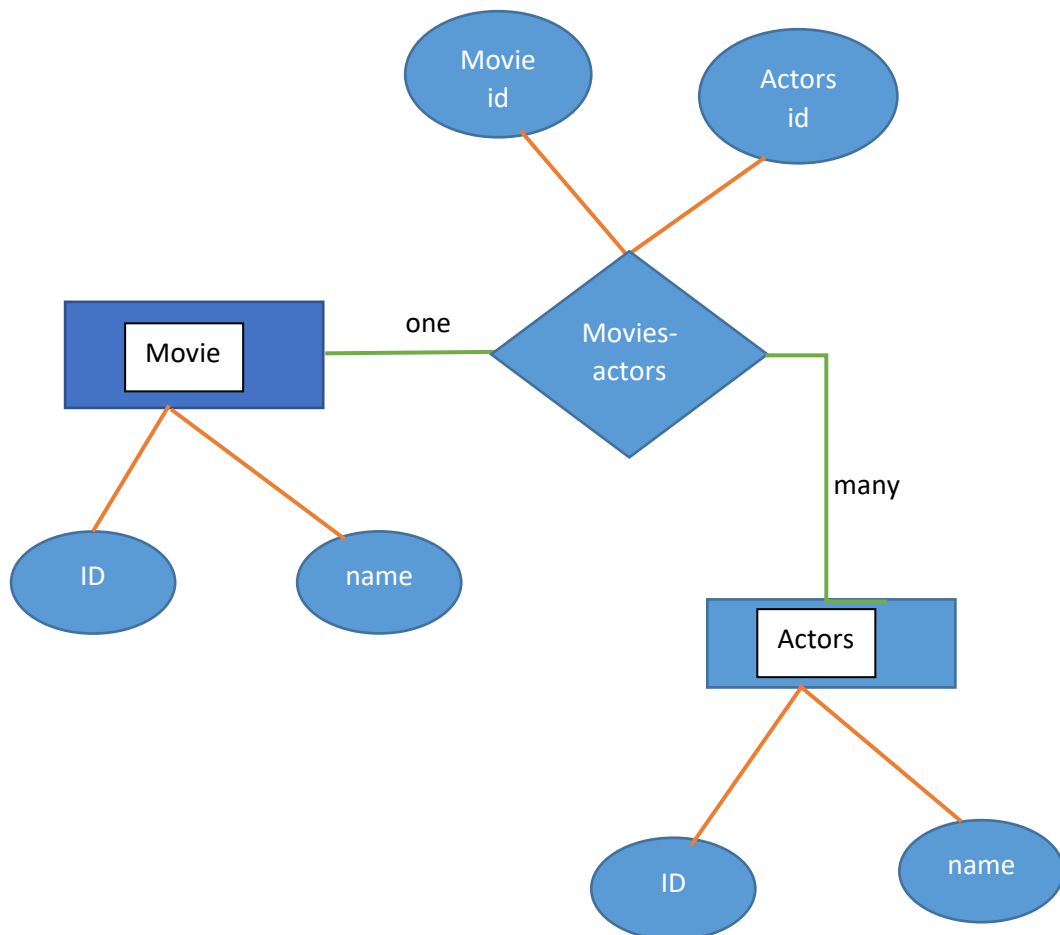


Figure: Entity-Relationship Diagram

There are 3 tables from database movie which is movies, actors and movisactors.

Table movisactors

have both primary key and foreign key which is actorsID and MoviesID. This table make a relationship between two tables which is movies and actors. So we can

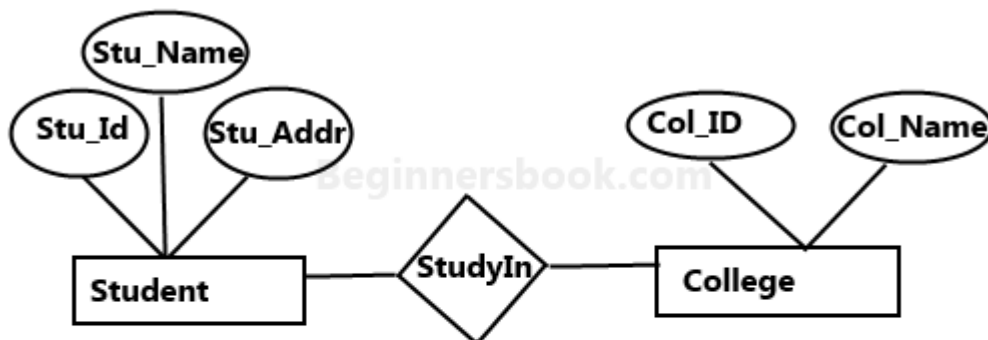
access data from table movies and actors by joining the tables through moviesactors.

4.2 Another example of ER diagram

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are entity set and relationship set.

What is an Entity Relationship Diagram (ER Diagram)?

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities, and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.



Sample E-R Diagram

4.3 A simple ER Diagram:

In the following diagram we have two entities Student and College and their relationship. The relationship between Student and College is many to one as a college can have many students however a student cannot study in multiple colleges at the same time. Student entity has attributes such as Stu_Id, Stu_Name & Stu_Addr and College entity has attributes such as Col_ID & Col_Name. E-R Diagram Here are the geometric shapes and their meaning in an E-R Diagram. We will discuss these terms in detail in the next section (Components of a ER Diagram) of this guide so don't worry too much about these terms now, just go through them once. (Guru99, 2017)

5 Explain about the SQL.

5.1 Create actors table.

```
CREATE TABLE `actors` (  
  
    `id` int(11) NOT NULL,  
  
    `name` varchar(30) NOT NULL  
  
)
```

5.2 Dumping data for table `actors`

```
INSERT INTO `actors` (`id`, `name`) VALUES(1, 'Jason Statham');  
  
INSERT INTO `actors` (`id`, `name`) VALUES(2, 'Josh Hartnett');  
  
INSERT INTO `actors` (`id`, `name`) VALUES(3, 'Holt McCallany');  
  
INSERT INTO `actors` (`id`, `name`) VALUES(6, 'Bruce Willis');
```

5.3 Indexes for table `actors`

```
ALTER TABLE `actors`  
  
    ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for dumped tables

5.4 Table structure for table `movies`

```
CREATE TABLE `movies` (  
  
    `id` int(11) NOT NULL,
```

```
`name` varchar(30) NOT NULL
)
```

5.5 Dumping data for table `movies`

```
INSERT INTO `movies` (`id`, `name`) VALUES(1, 'Wratt Of Man');

INSERT INTO `movies` (`id`, `name`) VALUES(2, 'Green Honnet');

INSERT INTO `movies` (`id`, `name`) VALUES(3, 'Clash Of Titans');
```

5.6 Indexes for table `movies`

```
ALTER TABLE `movies`
```

```
ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for table `actors`

5,7 Table structure for table `moviesactors`

```
CREATE TABLE `moviesactors` (

  `id` int(11) NOT NULL,

  `movies_id` int(11) NOT NULL,

  `actors_id` int(11) NOT NULL

)
```

5,8 Dumping data for table `moviesactors`

```
INSERT INTO `moviesactors` (`id`, `movies_id`, `actors_id`) VALUES(1, 1, 1);

INSERT INTO `moviesactors` (`id`, `movies_id`, `actors_id`) VALUES(2, 1, 3);

INSERT INTO `moviesactors` (`id`, `movies_id`, `actors_id`) VALUES(3, 1, 2);

INSERT INTO `moviesactors` (`id`, `movies_id`, `actors_id`) VALUES(11, 1, 6);

INSERT INTO `moviesactors` (`id`, `movies_id`, `actors_id`) VALUES(12, 2, 6);
```

5.9 ALTER TABLE `moviesactors`

```
ADD PRIMARY KEY (`id`),
```

```
ADD KEY `Constraint_FK_Movies_ID` (`movies_id`),
```

```
ADD KEY `Constraint_FK_actors_ID` (`actors_id`);
```

AUTO_INCREMENT for table `movies`

```
ALTER TABLE `moviesactors`
```

```
ADD CONSTRAINT `Constraint_FK_Movies_ID` FOREIGN KEY (`movies_id`)
REFERENCES `movies` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `Constraint_FK_actors_ID` FOREIGN KEY (`actors_id`)
REFERENCES `actors` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

The first we need to CREATE table, which is also known as add or insert. The SQL insert statement will be used to insert a new record in this transaction. To add new records to a table, SQL utilises the INSERT INTO command. (W3school, 1999-2021)

6. SELECT or read.

Actors Name select Input Field.

```
<?php
|
$link = mysqli_connect("127.0.0.1", "root", "", "moviesactors");

// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Escape user inputs for security
$actor_name = mysqli_real_escape_string($link, $_REQUEST['name']);

// Attempt insert query execution
$sql = "INSERT INTO actors (name) VALUES ('$actor_name')";
if(mysqli_query($link, $sql)){
    echo "Records added successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// Close connection
mysqli_close($link);
```

6.1 Movies Name Select Input Field

```
<?php

$link = mysqli_connect("127.0.0.1", "root", "", "moviesactors");

if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Escape user inputs for security
$movie_name = mysqli_real_escape_string($link, $_REQUEST['name']);

// Attempt insert query execution
$sql = "INSERT INTO movies (name) VALUES ('$movie_name')";
if(mysqli_query($link, $sql)){
    echo "Records added successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// Close connection
mysqli_close($link);
?>
```

SELECT (data retrieval). The word 'read' refers to retrieving data or a record set from a table that has been listed (s). The data is retrieved using SQL's SELECT statement. we can use SQL Server Management Studio, SQL Server Data Tools, or sqlcmd to run queries, depending on your preferences.

6.2 Update or input

Actors Name input Field

```
<!DOCTYPE html>
<html>
    <head>
        <title>Actor</title>
    </head>
    <body>
        <!-- HTML Form -->
        <form action="insertActorss.php" method="post">
            <fieldset>|
                <legend>
                    Actor information:
                </legend>
                Actor Name:
                <!-- Name Input Field -->
                <input type="text" name="name" required><br><br>
                <!-- Submit button field -->
                <input type="submit" value="submit">
            </fieldset>

        </form>
    </body>
</html>
<br><br>
<a href="index.php">Home</a>
```

6.3 movies Name Input Field

```
<!DOCTYPE html>
<html>
  <head>
    <title>Movie</title>
  </head>
  <body>
    <!-- HTML Form -->
    <form action="insertMovie.php" method = "post">
      <fieldset>
        <legend>
          Movie information:
        </legend>
        Movie Name:
        <!-- Name Input Field -->
        <input type="text" name="name" required><br><br>
        <!-- Submit button field -->
        <input type="submit" value="submit">
      </fieldset>
    </form>
  </body>
</html>
<br><br>
<a href="index.php">Home</a>
```


6.4 Movies to actors make relation field.

```
<?php
    $db_conn = mysqli_connect("127.0.0.1","root","","moviesactors");

    // Check database connection status
    if (mysqli_connect_errno()){
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
?>
<!DOCTYPE html>|
<html>
    <head>
        <title>Actor</title>
    </head>
    <body>
        <!-- HTML Form -->
        <form action="movie_actor.php" method = "post">
            <fieldset>
                <legend>
                    Actor/Movie information:
                </legend>
                Actor Name:
                <!-- Name select Input Field -->
                <?php
                    $query = "SELECT * FROM `actors` WHERE 1";
                    $res = mysqli_query($db_conn, $query);
                    echo "<select name = 'nameA'>";
                    while (($row = mysqli_fetch_array($res)) != null)
                    {
                        echo "<option value = '{ $row['id']}'>";
                        echo ">{ $row['name']}</option>";
                    }
                    echo "</select><br><br>";
                ?>

                Movie Name:
                <!-- Actors select Input Field -->

                <?php
                    $query = "SELECT * FROM `movies` WHERE 1";
                    $res = mysqli_query($db_conn, $query);
                    echo "<select name = 'nameM'>";
                    while (($row = mysqli_fetch_array($res)) != null)
                    {

                        echo "<option value = '{ $row['id']}'>";
                        echo ">{ $row['name']}</option>";
                    }
                    echo "</select><br><br>";
                ?>

                <!-- Submit button field -->
                <input type="submit" value="submit">
            </fieldset>
            <br><br>
            <a href="index.php">Home</a>

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

we will need to identify the target table and the columns that need to be changed, as well as the related values, and you may also need to know which rows need to be modified when executing an update. To minimise lock escalation and concurrent problems, we should keep the number of rows to a minimum. The set keyword is followed by the column name and the value to be set, which can be an expression, default, keyword, or null value. If you want to indicate which rows are changed by a search criterion, use the following syntax: Everything is the same as in the previous example, but this time you will see an expression after the "where" clause. (overflow, 2021)

6.5 Delete movie or actor name.

SQL Queries

Deleting Actor

```
DELETE FROM actors WHERE id = id;
DELETE FROM moviesactors WHERE actors_id = id;
```

Deleting Movie

```
DELETE FROM movies WHERE id = id;
DELETE FROM moviesactors WHERE movies_id = id;
```

Deleting Actors in Movies

```
DELETE FROM moviesactors WHERE actors_id = id;
```

If we want to delete by php code.

```
if (isset($_GET['deleteItem'])) {
    // Store data into the variables
    $item = $_GET['id_item'];

    // Insert query
    $sql = "DELETE FROM `items` WHERE id = ' . $item . '";

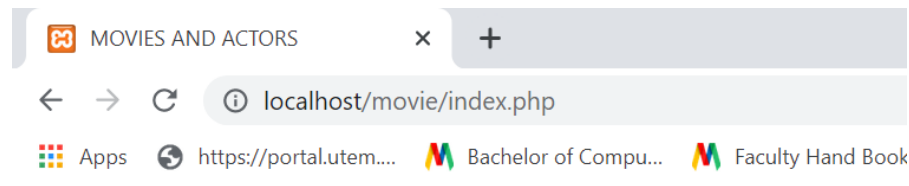
    // Inserting data into items table
    $delete_data = mysqli_query($db_conn, $sql);

    // Check if data deleted
    if($delete_data){
        echo "Data deleted successfully.";
    }
    // else data not deleted
    else{
        echo "Something error occurred";
    }
}
```

The DELETE command in SQL is used to delete the record(s) from the table. When crafting a DELETE statement, you'll provide the target table as well as the rows you want to delete. In

its most basic form, the syntax is the DELETE keyword followed by the table name. In rare cases, running a query without a WHERE clause will remove ALL existing entries from the table. Use the WHERE clause followed by the phrase to add a condition clause to the SQL DELETE statements (overflow, 2021)

7. RESULT



MOVIES ACTORS

[Add Movie](#)

[Add Actor](#)

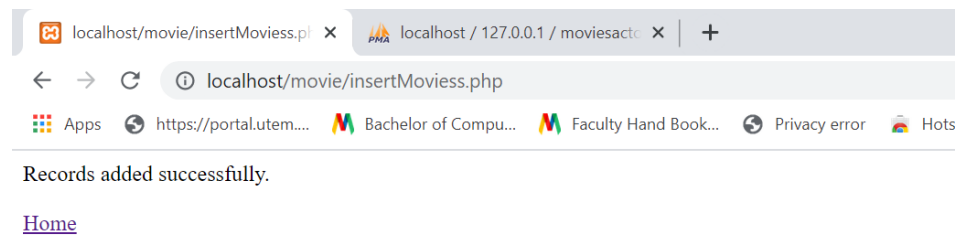
[Add Actor To A Movie](#)

[View Movie Actors](#)

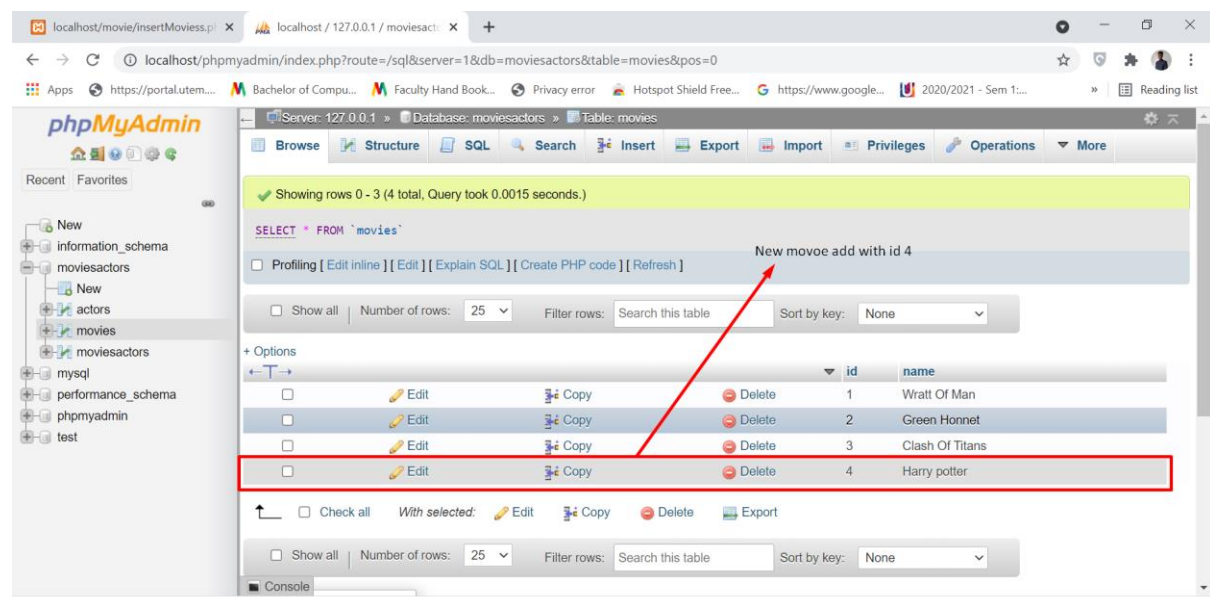
Add movie name.

[Home](#)

When I click submit button the movie will add with give message Records added successfully

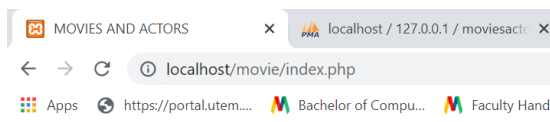
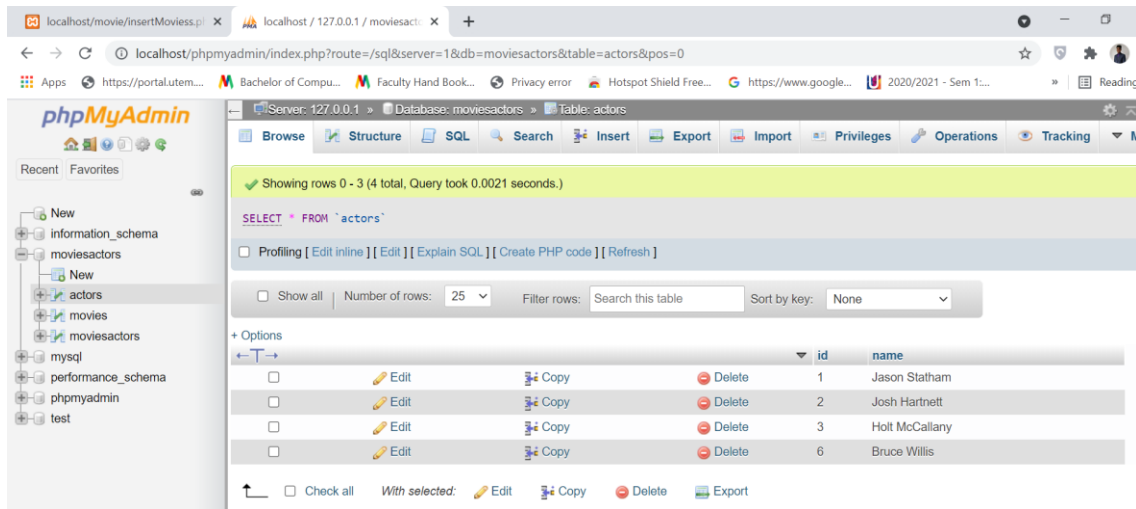


On my local server the movie name added with id number



Now I want to add actor's names. So, we know 1 move can act many people so we may add many names.

My local server Before adding harry potter movie's actor name. we can see here 4 data.



MOVIES ACTORS

[Add Movie](#)

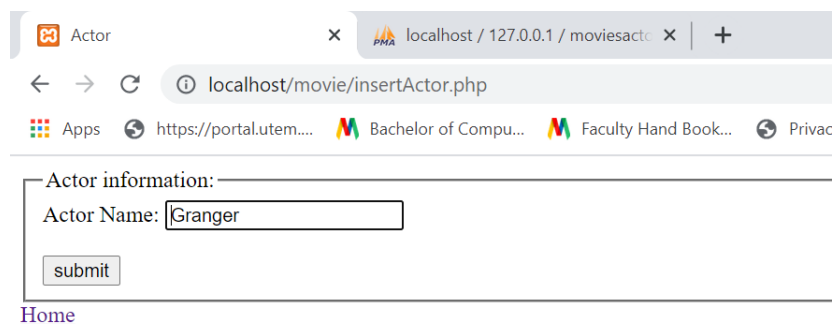
[Add Actor](#)

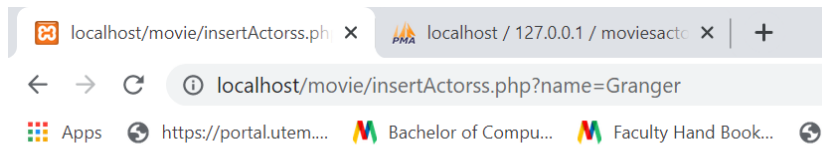
[Add Actor To A Movie](#)

[View Movie Actors](#)

I will add 4 actors name which is

1. Granger
2. harry
3. Lord
4. Draco Malfo

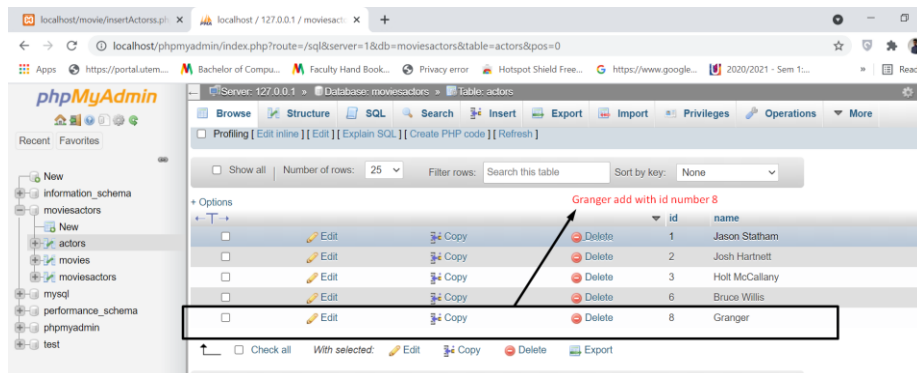




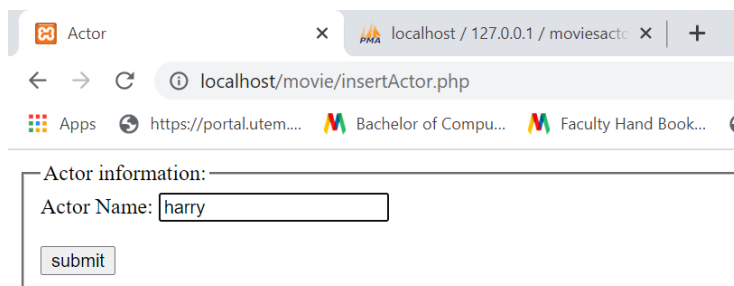
Records added successfully.

[Home](#)

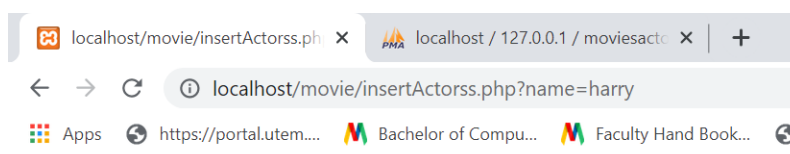
Actor name Granger add on my local server with id number 8.



Second name add.



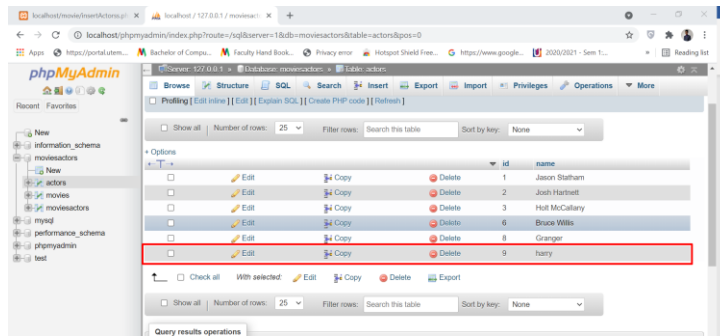
[Home](#)



Records added successfully.

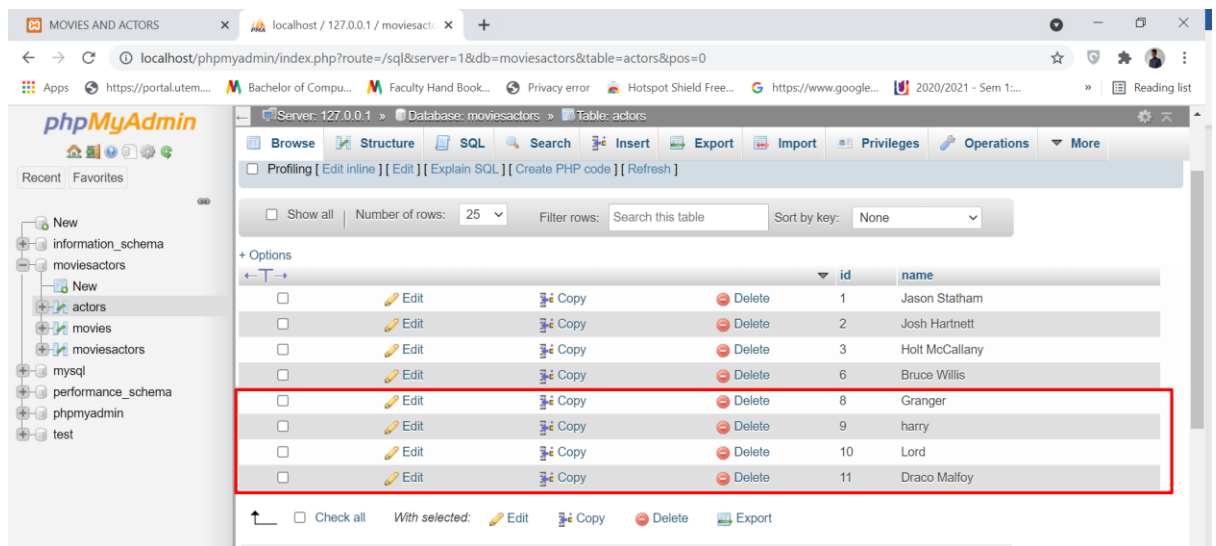
[Home](#)

Actor name harry add on my local server with id number 9.

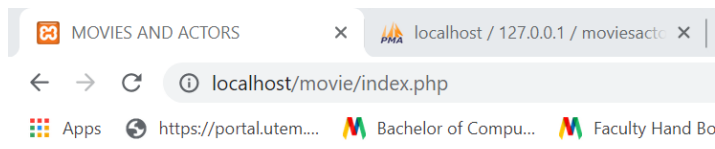


Same way we will add 4 actor's names.

On my local host added 4 actor's names.



Now we will make relation between movies and actors. 1 movie many actors



MOVIES ACTORS

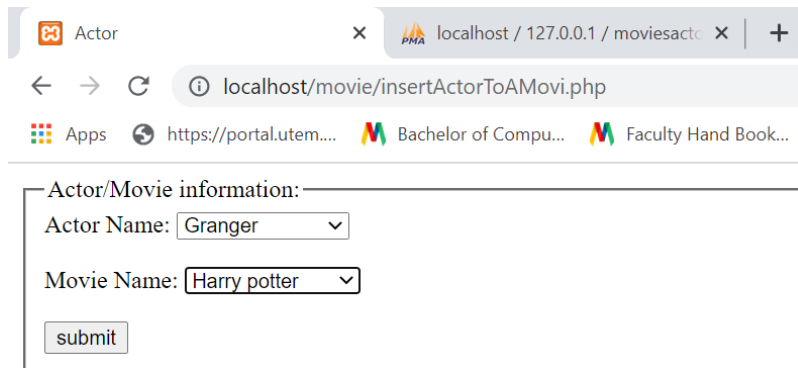
[Add Movie](#)

[Add Actor](#)

[Add Actor To A Movie](#)

[View Movie Actors](#)

ASSIGNMENT – BEEC 1343



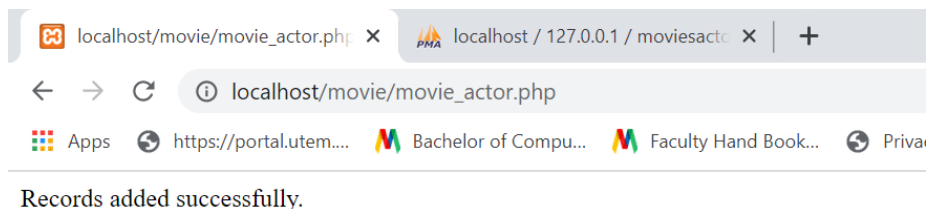
Actor/Movie information:

Actor Name:

Movie Name:

[Home](#)

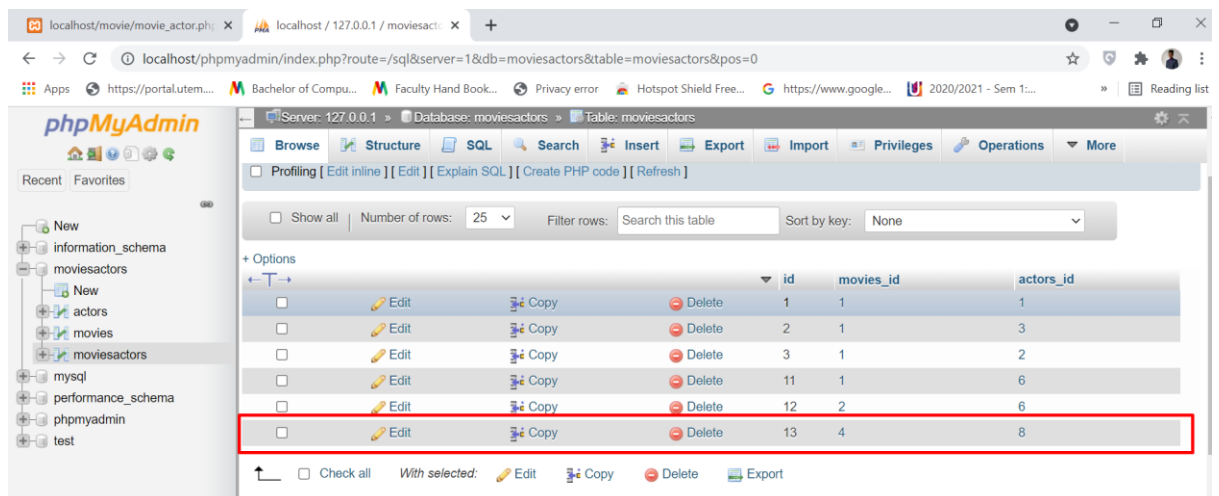
Related first actor with movie.



Records added successfully.

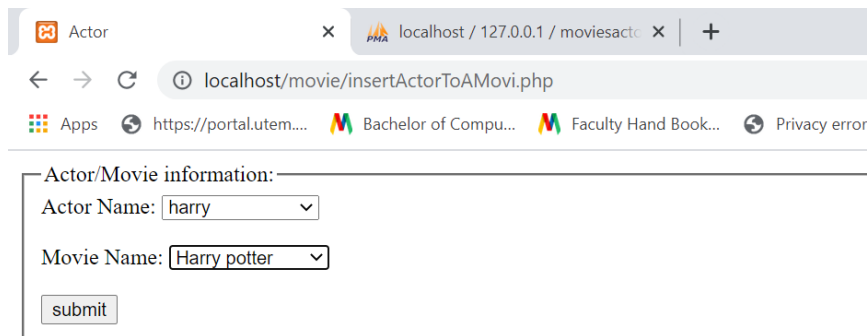
[Home](#)

Movie id 4 and actors id 8 this two are foreign key. So new id 13 makes a relation between movie name with actor name.



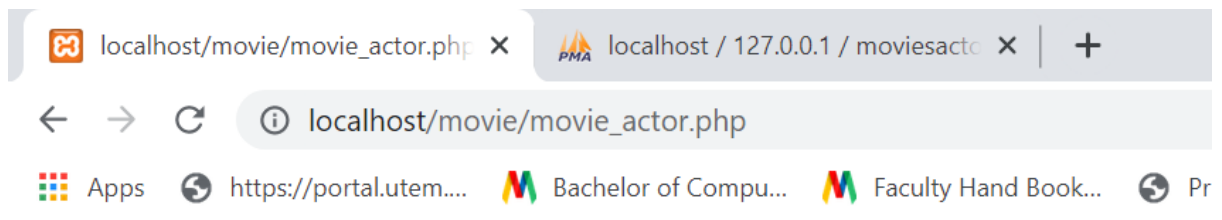
	id	movies_id	actors_id
<input type="checkbox"/>	1	1	1
<input type="checkbox"/>	2	1	3
<input type="checkbox"/>	3	1	2
<input type="checkbox"/>	11	1	6
<input type="checkbox"/>	12	2	6
<input type="checkbox"/>	13	4	8

Again we will make relation between movies with second actors because we know with 1 movie can relation with many actors so we can make many relation actors name with 1 movie



[Home](#)

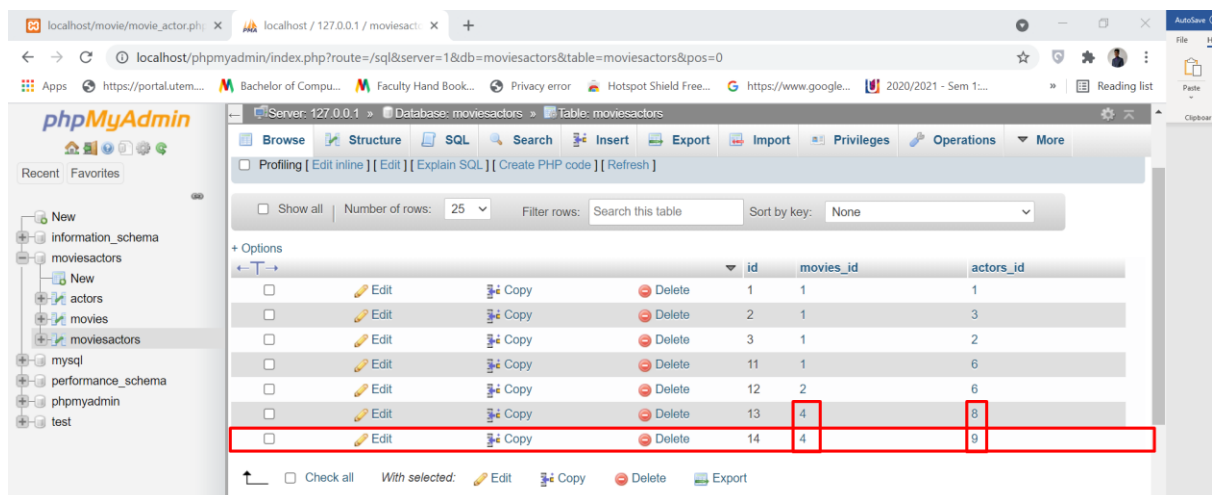
Second Record added successfully.



Records added successfully.

[Home](#)

On my local server we can see Movie id 4 and now actors id is 9 this two are foreign key. So new id 14 makes a relation between movie name with actor name. here we can see movie id not changing but actor id changing because 1 movie many actor.



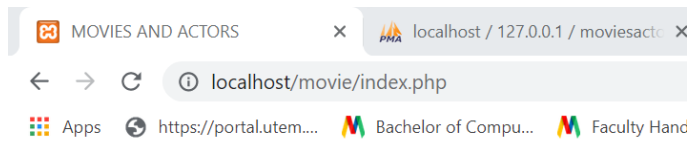
	id	movies_id	actors_id
<input type="checkbox"/>	1	1	1
<input type="checkbox"/>	2	1	3
<input type="checkbox"/>	3	1	2
<input type="checkbox"/>	11	1	6
<input type="checkbox"/>	12	2	6
<input type="checkbox"/>	13	4	8
<input type="checkbox"/>	14	4	9

So, like this way we will make relation another 2-actor name with movie name.

On my local server we can see that 4 actors make relation with 1 movie

	id	movies_id	actors_id
	1	1	1
	2	1	3
	3	1	2
	11	1	6
	12	2	6
	13	4	8
	14	4	9
	15	4	10
	16	4	11

Now if we want to see from website that harry Potter movie how many actor make



MOVIES ACTORS

[Add Movie](#)

[Add Actor](#)

[Add Actor To A Movie](#)

[View Movie Actors](#)

Movie Actors information:

Movie Name: Wratt Of Man

Show Movie A

[Home](#)

Here we can see 4 actor make harry Potter movie

Movie Actors information:-

Movie Name:

- Jason Statham
- Holt McCallany
- Josh Hartnett
- Bruce Willis

[Home](#)

CONCLUSION

Finally, we used equipment such as XAMPP Server Software, a Web Browser, and a Personal Computer or Laptop. In addition, we created a MySQL database table called DVD Titleship. We learned how to add information on four of our favourite movies as a result of this. Then we must use files such as php or html that must be saved in this place. We'll make a more secure password for ourselves. Finally, we have completed the design and development of the DVD Rental System.

8. References

- Craig s.Mullins, M. C. (2 5, 2005-20021). *TechTarget*. Retrieved from searchsqlserver.Techtarget.com:
<https://searchsqlserver.techtarget.com/definition/database-management-system>
- Guru99. (21 11, 2017). *Guru99*. Retrieved from guru99.com: <https://www.guru99.com/er-diagram-tutorial-dbms.html#:~:text=ER%20Diagram%20stands%20for%20Entity,the%20logical%20structure%20of%20databases.>
- Ireland, I. N. (1 4, 2018). *NIBUSINESS INFO*. Retrieved from <https://www.nibusinessinfo.co.uk/>:
<https://www.nibusinessinfo.co.uk/content/benefits-database-development>
- overflow, s. (2 5, 2021). *stackoverflow*. Retrieved from stackoverflow.com:
<https://stackoverflow.com/questions/27345377/try-catch-in-mysqli>
- W3school. (21 2, 1999-2021). *Weschool sqltutrial*. Retrieved from <https://www.w3schools.com/>: <https://www.w3schools.com/sql/default.Asp>
- warehousing, O. d. (2 9, 2015). *Oracle@*. Retrieved from <https://docs.oracle.com/>:
https://docs.oracle.com/cd/E18283_01/server.112/e16579/physical.htm