Assignment 2

Selection of Case study for the creation of the database

There was a choice of two cases which were could choose from so that we could create four linked tables.

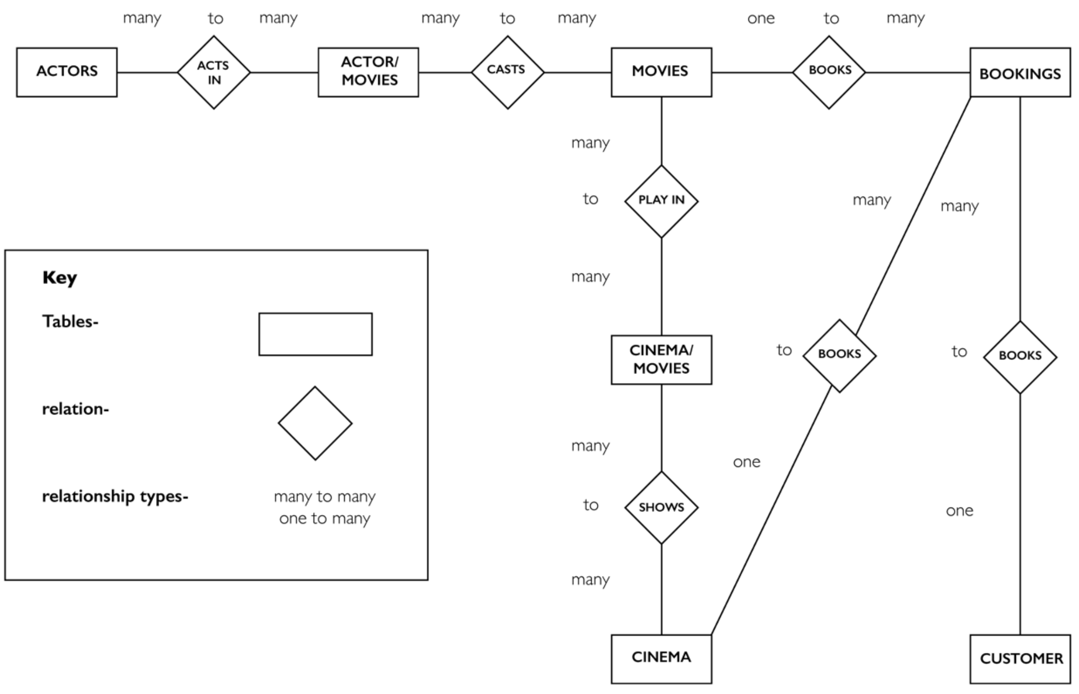
The option that we have selected is MovieHouse.

Design of the tables

MovieHouse
Movies – FilmID, FilmName, Screen,
Actormovies – FilmID*, ActorID*
Actors - ActorID, FirstName, Surname, Salary
Cinema – CinemaID, CinemaName,
Bookings – BookingID, FilmID*, CinemaID* CustomerID*
Customer – CustomerID, FirstName, Surname, Address
Cinemashows – CinemaID*, FilmID*

Key – (PK – underlined FK – *)

ER Diagram



Primary and Foreign Keys

Actors Table

Primary Key: Primary Key Within the actors table will be the ActorID. This is because this is the unique number which will identify each of the actors within the table.

Movies Table

Primary Key: Primary Key will be the FilmID. This is because this is the unique identifier used for each of the movies which are currently being shown within the cinema.

Bookings Table

Primary Key: The Primary Key within this table will be the booking ID because this is the unique number that each booking is given so that they would be located within the database easily. This also means that each booking is unique and can be located easily.

Foreign Key: There are three foreign keys in this table which are CinemaID, FilmID and CustomerID. These are the three foreign keys because this table must draw information from the Cinema, Movies and Customer table to ensure that the correct bookings can be taken so each customer is able to attend the correct film that they have booked at the correct Cinema of their choice.

Customer Table

Primary Key: The primary key within the customer table within the database will be the CustomerID. This is because this is the unique field which will be used to identify the customers who have place a booking to see a film which is being shown. This means that each unique customers booking could be located easily.

Foreign Key: This table will not contain any foreign keys because this table is passing information into the booking table and is not taking information from another table within the database. So this means that it is not necessary for a foreign key within the table.

Cinema Table

Primary Key: The primary key within this table. One is the cinema ID. Which is the unique identifier used for each of the Cinemas which exist within the MovieHouse Family. This means that the cinema can be identified easily.

Foreign Key: There is no foreign key needed within this table as it is passing information across into another table and is not taking information into this table and is linking to another table.

Cinemashows

Uses a composite key consisting of CinemaID and FilmID: This means a cinema can show multiple movies and movies can be shown in multiple cinemas.

Foreign Key: CinemaID and FilmID, this connects the Cinemas and Movies tables enabling users to select what cinema and what film they want to book.

Actormovies

Uses a composite key consisting of FilmID and ActorID: This means many actors cans star in many movies and a movie can have many actors.

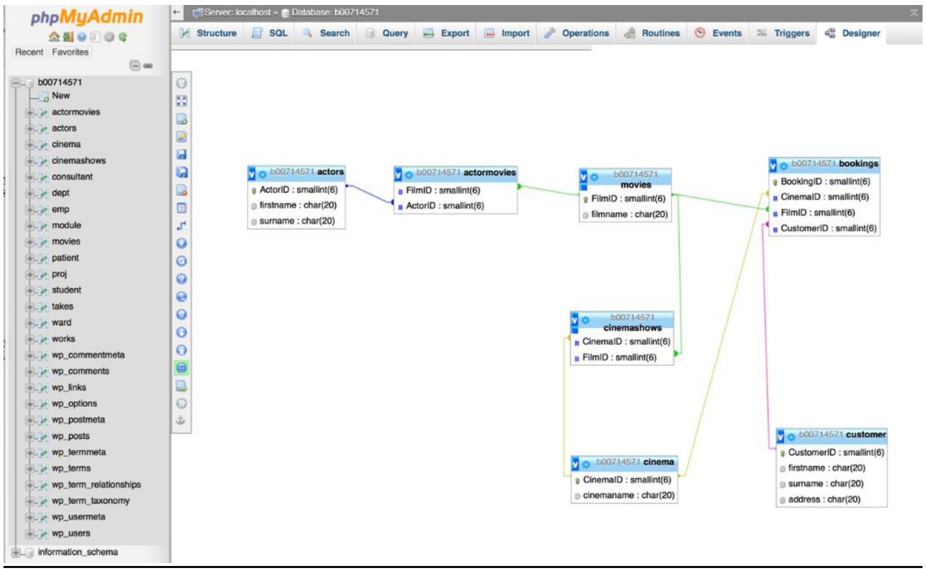
Foreign Key: ActorID and FilmID, this connects the Actors and Movies tables enabling users to see which actors star in which movies.

Create Tables

Customer Table	**Booking Table**	**Actors Table**	**Movies Table**
CREATE TABLE`customer`(	CREATE TABLE`bookings`(	CREATE TABLE `actors`(	CREATE TABLE `movies`(
`CustomerID` SMALLINT(6) NOT NULL,	`BookingID` SMALLINT(6) NOT NULL,	`ActorID` SMALLINT(6) NOT NULL,	`FilmID` SMALLINT(6) NOT NULL,
`firstname` char(20) NOT NULL,	`CinemaID` SMALLINT(6) NOT NULL,	`firstname` char(20) NOT NULL,	`filmname` char(20) NOT NULL,
`surname` char(20) NOT NULL,	`FilmID` SMALLINT(6) NOT NULL,	`surname` char(20) NOT NULL	PRIMARY KEY (`FilmID`)
`address` char(20) NOT NULL,		PRIMARY KEY (`ActorID`)	)
PRIMARY KEY (`CustomerID`)		)	
);			

Cinema Table	**Cinemashows table**	**Actormovies table**
CREATE TABLE`cinema`(	CREATE TABLE`cinemashows`(	CREATE TABLE`actormovies`(
`CinemaID` SMALLINT(6) NOT NULL,	`CinemaID` SMALLINT(6) NOT NULL,	`FilmID` SMALLINT(6) NOT NULL,
`cinemaname` char(20) NOT NULL,	`FilmID` SMALLINT(6) NOT NULL,	`ActorID` SMALLINT(6) NOT NULL,
PRIMARY KEY (`CinemaID`)	FOREIGN KEY (`FilmID`) REFERENCES movies(`FilmID`),	FOREIGN KEY (`FilmID`) REFERENCES movies(`FilmID`),
)	FOREIGN KEY (`CinemaID`) REFERENCES cinema(`CinemaID`)	FOREIGN KEY (`ActorID`) REFERENCES actors(`ActorID`)
	)	)

Screenshot of the relational model


phpMyAdmin
Server: localhost » Database: b00714571
Structure
SQL
Search
Query
Export
Import
Operations
Routines
Events
Triggers
Designer
Recent
Favorites
b00714571
New
actormovies
actors
cinema
cinemashows
consultant
dept
emp
module
movies
patient
proj
student
takes
ward
works
wp_commentmeta
wp_comments
wp_links
wp_options
wp_postmeta
wp_posts
wp_termmeta
wp_terms
wp_term_relationships
wp_term_taxonomy
wp_usermeta
wp_users
information_schema
b00714571 actors
ActorID : smallint(6)
firstname : char(20)
surname : char(20)
b00714571 actormovies
FilmID : smallint(6)
ActorID : smallint(6)
b00714571 movies
FilmID : smallint(6)
filmname : char(20)
b00714571 bookings
BookingID : smallint(6)
CinemaID : smallint(6)
FilmID : smallint(6)
CustomerID : smallint(6)
b00714571 cinemashows
CinemaID : smallint(6)
FilmID : smallint(6)
b00714571 cinema
CinemaID : smallint(6)
cinemaname : char(20)
b00714571 customer
CustomerID : smallint(6)
firstname : char(20)
surname : char(20)
address : char(20)