**Movie Rental**

By Jeremy Swartwood

CS 360 (Databases) Spring 2011

Professor Scott, Kirk.

******

***NOTE:*** *the table of contents is dynamic and linked. To jump to a section CTRL+Click the Chapter.*

**Table of Contents**

[1. Project Overview 4](#_Toc291479135)

[1.1 Problem Domain 4](#_Toc291479136)

[1.2 Purpose 4](#_Toc291479137)

[2. E-R Diagram 5](#_Toc291479138)

[3. Database and Table Schema 6](#_Toc291479139)

[3.1 Database Creation 6](#_Toc291479140)

[3.2 User Creation 6](#_Toc291479141)

[3.3 Table Creation 6](#_Toc291479142)

[3.3.1 tblAccount 6](#_Toc291479143)

[3.3.2 tblAccountCustomer 6](#_Toc291479144)

[3.3.3 tblAccountPlan 7](#_Toc291479145)

[3.3.4 tblAccountRelationshipType 7](#_Toc291479146)

[3.3.5 tblCopyInformation 7](#_Toc291479147)

[3.3.6 tblCustomer 7](#_Toc291479148)

[3.3.7 tblGenre 8](#_Toc291479149)

[3.3.8 tblMediaType 8](#_Toc291479150)

[3.3.9 tblMovie 8](#_Toc291479151)

[3.3.10 tblRentalHistory 8](#_Toc291479152)

[4. Data Dump from Tables (Test Data) 9](#_Toc291479153)

[4.1 tblAccount 9](#_Toc291479154)

[4.2 tblAccountCustomer 9](#_Toc291479155)

[4.3 tblAccountPlan 9](#_Toc291479156)

[4.4 tblAccountRelationshipType 9](#_Toc291479157)

[4.5 tblCopyInformation 9](#_Toc291479158)

[4.6 tblCustomer 10](#_Toc291479159)

[4.7 tblGenre 10](#_Toc291479160)

[4.8 tblMediaType 11](#_Toc291479161)

[4.9 tblMovie 11](#_Toc291479162)

[4.10 tblRentalHistory 16](#_Toc291479163)

[5. Data Dictionary 17](#_Toc291479164)

[5.1 Domain Documentation 17](#_Toc291479165)

[5.2 Field Documentation 18](#_Toc291479166)

[6. Basic Queries 19](#_Toc291479167)

[6.1 Basic Select Queries 19](#_Toc291479168)

[6.2 Basic Order by and Group By Queries 20](#_Toc291479169)

[7. Advanced Queries 22](#_Toc291479170)

[7.1 Join Queries 22](#_Toc291479171)

[7.2 Subqueries, IN/EXISTS Queries 24](#_Toc291479172)

[8. Indexes 26](#_Toc291479173)

[8.1 Faster Customer Lookup by Last Name 26](#_Toc291479174)

[8.2 Faster Customer Lookup by First Name 26](#_Toc291479175)

[8.3 Faster Movie Lookup by Running Length 26](#_Toc291479176)

[8.4 Faster Movie Lookup by Title 26](#_Toc291479177)

[9. Showing Query Results in a Web page. 27](#_Toc291479178)

[9.1 Screenshot 27](#_Toc291479179)

[9.2 Code 27](#_Toc291479180)

[10. Adding records to and deleting records from a database. 31](#_Toc291479181)

[10.1 Sceenshot 31](#_Toc291479182)

[10.2 Code 31](#_Toc291479183)

[11. Retrieving a Record using a form. 35](#_Toc291479184)

[11.1 Screenshot 35](#_Toc291479185)

[11.2 Code 35](#_Toc291479186)

[12. Updating a Record using a form. 40](#_Toc291479187)

[12.1 Screenshot 40](#_Toc291479188)

[12.2 Code 40](#_Toc291479189)

# 1. Project Overview

## 1.1 Problem Domain

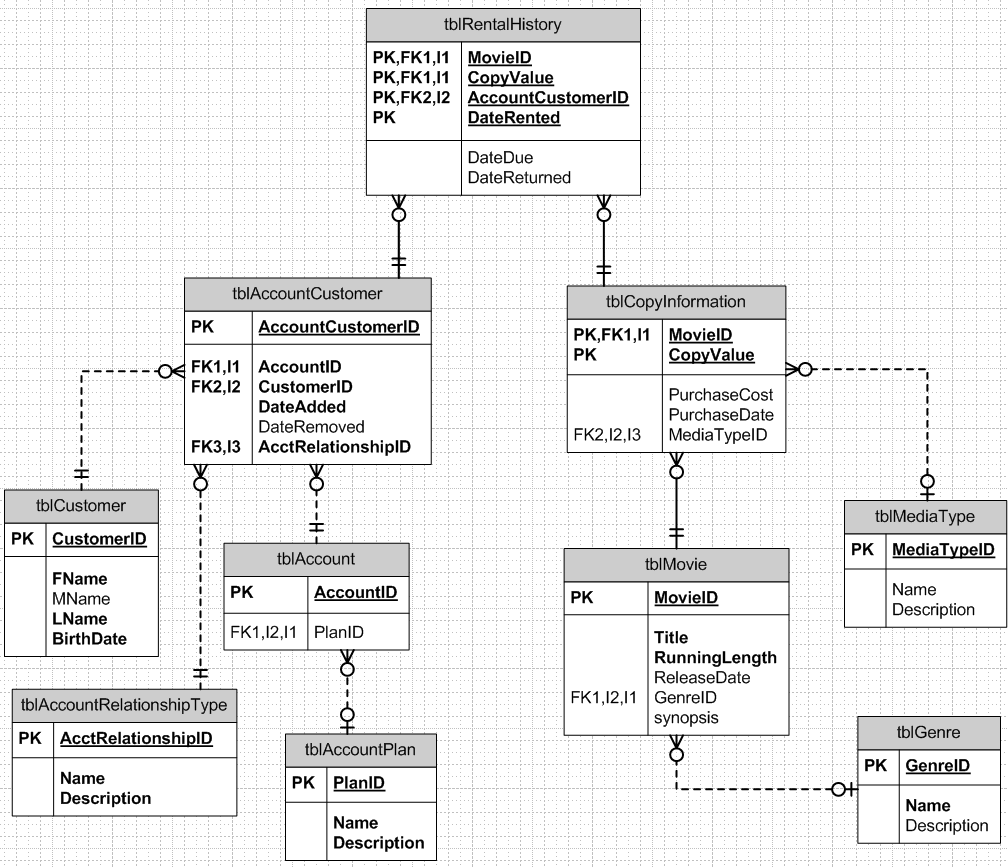
Every movie rental business needs a way to store movie information, keep track of customers, and manage the rental of those movies to the customers.

## 1.2 Purpose

There are many ways to do this, however the best way is by using a database. Not just any database, but a web based database.

This document covers the Movie Rental database that is used to track the rental of movies to customers, while allowing for restricted rentals to minors, and limited liability through tracked late fees.

# 2. E-R Diagram



# 3. Database and Table Schema

## 3.1 Database Creation

The following statement creates a database in MySQL that uses the UTF8 text character format.

CREATE DATABASE MovieRental CHARACTER SET = 'utf8';

## 3.2 User Creation

Creating a MySQL user account specifically for allowing online connections to the MySQL database helps to restrict the access to the Linux computer (web server).

**First we create the user account with the following SQL Statement:**

CREATE USER 'remoteuser'@'localhost' IDENTIFIED BY 'RemoteUser1';

**Give the user permissions to select, insert, update, and delete ONLY to the MovieRental database.**

GRANT SELECT, INSERT, UPDATE, DELETE on MovieRental.\* TO 'remoteuser'@'localhost';

## 3.3 Table Creation

This database has 10 tables. Each table has its own sub section.

### 3.3.1 tblAccount

This is to store the unique account numbers and the plan that the account has.

CREATE TABLE tblAccount (

AccountID INT(10) NOT NULL AUTO\_INCREMENT,

PlanID INT(3) NOT NULL,

CONSTRAINT pk\_tblAccount PRIMARY KEY (AccountID),

CONSTRAINT fk\_tblAccount FOREIGN KEY (PlanID) REFERENCES tblAccountPlan(PlanID));

### 3.3.2 tblAccountCustomer

Stores the which customers are on what accounts. Including the type of relationship they have on that account.

CREATE TABLE tblAccountCustomer (

AccountCustomerID int(10) NOT NULL AUTO\_INCREMENT,

AccountID INTEGER NOT NULL,

CustomerID INTEGER NOT NULL,

DateAdded DATE NOT NULL,

DateRemoved DATE,

AcctRelationshipID INTEGER NOT NULL,

CONSTRAINT pk\_tblAcctCust PRIMARY KEY (AccountCustomerID),

CONSTRAINT fk\_tblAcctCust1 FOREIGN KEY (AccountID) REFERENCES tblAccount(AccountID),

CONSTRAINT fk\_tblAcctCust2 FOREIGN KEY (CustomerID) REFERENCES tblCustomer(CustomerID),

CONSTRAINT fk\_tblAcctCust3 FOREIGN KEY (AcctRelationshipID) REFERENCES tblAccountRelationshipType (AcctRelationshipID));

### 3.3.3 tblAccountPlan

Each plan has a specific way it will charge the user. Monthly, per DVD, late fees, etc.

CREATE TABLE tblAccountPlan(

PlanID INT(3) NOT NULL AUTO\_INCREMENT,

Name VARCHAR(35) NOT NULL,

Description VARCHAR(255) NOT NULL,

CONSTRAINT pk\_tblAccountPlan PRIMARY KEY (PlanID));

### 3.3.4 tblAccountRelationshipType

Another lookup table. This one stores information about the different types of account relationships customers can have.

CREATE TABLE tblAccountRelationshipType (

AcctRelationshipID int(10) NOT NULL AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Description VARCHAR(255) NOT NULL,

CONSTRAINT pk\_tblAcctRelateType PRIMARY KEY (AcctRelationshipID));

### 3.3.5 tblCopyInformation

Stores the details on each copy of a movie. Media Type, Purchase cost, Puchase date, etc.

CREATE TABLE tblCopyInformation (

MovieID INTEGER NOT NULL,

CopyValue VARCHAR(3) NOT NULL,

PurchaseCost DECIMAL(10,2),

PurchaseDate DATE,

MediaTypeID INT(3) NOT NULL,

CONSTRAINT pk\_tblCopyInfo PRIMARY KEY (MovieID, CopyValue),

CONSTRAINT fk\_tblCopyInfo FOREIGN KEY (MovieID) REFERENCES tblMovie(MovieID),

CONSTRAINT fk\_tblCopyInfo2 FOREIGN KEY (MediaTypeID) REFERENCES tblMediaType(MediaTypeID));

### 3.3.6 tblCustomer

List of all the customers

CREATE TABLE tblCustomer (

CustomerID int(10) NOT NULL AUTO\_INCREMENT,

FName VARCHAR(25) NOT NULL,

MName VARCHAR(25),

LName VARCHAR(25) NOT NULL,

BirthDate DATE NOT NULL,

CONSTRAINT pk\_tblCustomer PRIMARY KEY (CustomerID));

### 3.3.7 tblGenre

This table holds the different Styles of movies. For instance, Action, Comedy, Scary, etc.

CREATE TABLE tblGenre(

GenreID int(3) NOT NULL AUTO\_INCREMENT,

Name varchar(30) NOT NULL,

Description varchar(255),

CONSTRAINT pk\_tblGenre PRIMARY KEY (GenreID));

### 3.3.8 tblMediaType

The different possible formats the movie can be rented in. VHS, VCD, DVD, BD.

CREATE TABLE tblMediaType(

MediaTypeID INT(3) NOT NULL AUTO\_INCREMENT,

Name VARCHAR(30),

Description VARCHAR(255),

CONSTRAINT pk\_tblMediaType PRIMARY KEY (MediaTypeID));

### 3.3.9 tblMovie

Each plan has a specific way it will charge the user. Monthly, per dvd, late fees, etc.

CREATE TABLE tblMovie(

MovieID Int(10) NOT NULL AUTO\_INCREMENT,

Title VARCHAR(255) NOT NULL,

RunningLength INTEGER NOT NULL,

ReleaseDate DATE,

GenreID INT(3) NOT NULL,

Synopsis blob,

CONSTRAINT pk\_tblMovie PRIMARY KEY (MovieID),

CONSTRAINT fk\_tblMovieGenre FOREIGN KEY (GenreID) REFERENCES tblGenre(GenreID));

### 3.3.10 tblRentalHistory

Information on each copy of a movie rented, which customer rented it, details about when it was rented and when it is due.

CREATE TABLE tblRentalHistory (

MovieID INTEGER NOT NULL,

CopyValue VARCHAR(3) NOT NULL,

AccountCustomerID INTEGER NOT NULL,

DateRented DATE NOT NULL,

DateDue DATE,

DateReturned DATE,

CONSTRAINT pk\_tblRentalHistory PRIMARY KEY (MovieID, CopyValue, AccountCustomerID, DateRented),

CONSTRAINT fk\_tblRentalHistory1 FOREIGN KEY (MovieID, CopyValue) REFERENCES tblCopyInformation (MovieID, CopyValue),

CONSTRAINT fk\_tblRentalHistory2 FOREIGN KEY (AccountCustomerID) REFERENCES tblAccountCustomer (AccountCustomerID));

# 4. Data Dump from Tables (Test Data)

## 4.1 tblAccount

|  |  |
| --- | --- |
| **AccountID** | **PlanID** |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 3 |

## 4.2 tblAccountCustomer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **AccountCustomerID** | **AccountID** | **CustomerID** | **DateAdded** | **DateRemoved** | **AcctRelationshipID** |
| 1 | 1 | 1 | 2011-04-21 | NULL | 1 |
| 2 | 1 | 5 | 2011-04-21 | NULL | 2 |
| 3 | 2 | 11 | 2011-04-21 | NULL | 1 |
| 4 | 2 | 13 | 2011-04-21 | NULL | 2 |
| 5 | 2 | 12 | 2011-04-21 | NULL | 2 |
| 6 | 3 | 6 | 2011-04-21 | NULL | 1 |
| 7 | 3 | 2 | 2011-04-21 | NULL | 2 |
| 8 | 4 | 8 | 2011-04-21 | NULL | 1 |
| 9 | 4 | 3 | 2011-04-21 | NULL | 2 |
| 10 | 5 | 4 | 2011-04-21 | NULL | 1 |
| 11 | 5 | 9 | 2011-04-21 | NULL | 1 |
| 12 | 6 | 7 | 2011-04-21 | NULL | 1 |
| 13 | 6 | 10 | 2011-04-21 | NULL | 1 |

## 4.3 tblAccountPlan

|  |  |  |
| --- | --- | --- |
| **PlanID** | **Name** | **Description** |
| 1 | Individual | Each movie is a set cost, late fees. |
| 2 | Monthly - Basic | No per movie cost, 1 movie out at a time. |
| 3 | Monthly - Advanced | No per movie cost, 3 movie out at a time. |
| 4 | Unlimited - Fees | No limit on movies rented at a time, but late fees |
| 5 | Unlimited . no fee | No limit on movies rented at a time, no late fees |

## 4.4 tblAccountRelationshipType

|  |  |  |
| --- | --- | --- |
| **AcctRelationshipID** | **Name** | **Description** |
| 1 | Primary | The owner of the account. |
| 2 | Joint | Any other person on the account. |

## 4.5 tblCopyInformation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MovieID** | **CopyValue** | **PurchaseCost** | **PurchaseDate** | **MediaTypeID** |
| 1 | A | NULL | NULL | 3 |
| 2 | A | 12.99 | 2010-05-08 | 3 |
| 3 | A | 9.99 | 2009-07-01 | 3 |
| 4 | A | 7.99 | 2011-04-01 | 3 |
| 5 | A | NULL | 2010-03-20 | 3 |
| 6 | A | 5.99 | 2011-02-07 | 3 |
| 7 | A | 17.95 | 2011-04-20 | 3 |
| 8 | A | 15.65 | 2010-02-14 | 3 |
| 9 | A | 19.9 | 2008-02-11 | 3 |
| 10 | A | 13 | 2005-07-19 | 3 |
| 11 | A | 22.25 | 2009-01-02 | 3 |
| 12 | A | 19.95 | 2007-03-01 | 3 |
| 13 | A | 16.57 | 2004-08-01 | 3 |
| 7 | B | 17.95 | 2011-04-20 | 3 |
| 8 | B | 15.65 | 2010-02-14 | 3 |
| 1 | B | 19.75 | 2011-04-20 | 4 |
| 2 | B | 19.45 | 2011-04-20 | 4 |
| 3 | B | 18.85 | 2011-04-20 | 4 |
| 7 | C | 16.45 | 2011-04-20 | 4 |
| 8 | C | 13.75 | 2011-04-20 | 4 |
| 9 | B | 15.15 | 2011-04-20 | 4 |
| 12 | B | 17.75 | 2011-04-20 | 4 |
| 14 | B | 12.85 | 2011-04-20 | 4 |
| 2 | C | 19.75 | 2005-04-20 | 2 |

## 4.6 tblCustomer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CustomerID** | **FName** | **MName** | **LName** | **BirthDate** |
| 1 | Woody | NULL | Woodpecker | 1940-05-13 |
| 2 | Donald | Fauntleroy | Duck | 1934-06-09 |
| 3 | Mighty | NULL | Mouse | 1942-08-28 |
| 4 | Yogi | NULL | Bear | 1958-05-16 |
| 5 | Pink | NULL | Panther | 1964-01-08 |
| 6 | Daffy | NULL | Duck | 1937-04-17 |
| 7 | Wile | E | Coyote | 1949-09-16 |
| 8 | Mickey | NULL | Mouse | 1928-11-18 |
| 9 | Bugs | NULL | Bunny | 1940-07-27 |
| 10 | Scrappy | Conelius | Doo | 1979-01-21 |
| 11 | Jeremy | Joseph | Swartwood | 1981-08-28 |
| 12 | Kaltinrl | NULL | Peacefoot | 2006-01-01 |
| 13 | Valtari | NULL | Vas-Lassi | 1999-12-11 |

## 4.7 tblGenre

|  |  |  |
| --- | --- | --- |
| **GenreID** | **Name** | **Description** |
| 1 | Action | Lots of movement or fighting |
| 2 | Adult | Raunchy or of high sexual content |
| 3 | Comedy | Funny, or comical. |
| 4 | Drama | Emotional or chick flick |
| 5 | Family | Movie for the whole family, kids, adults, etc |
| 6 | Life Story | Documentary or history of a person. |
| 7 | Romantic | Love story |
| 8 | Scary | Horror or thriller |
| 9 | Sci-Fi | Science Fiction, fantasy |
| 10 | Suspence | Riveting story that keeps you on the edge of your seat |

## 4.8 tblMediaType

|  |  |  |
| --- | --- | --- |
| **MediaTypeID** | **Name** | **Description** |
| 1 | VHS | Virtual Home System |
| 2 | VCD | Video Compact Disk |
| 3 | DVD | Digital Video Disk |
| 4 | BD | Blu-ray Disk |

## 4.9 tblMovie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MovieID** | **Title** | **RunningLength** | **ReleaseDate** | **GenreID** | **Synopsis** |
| 1 | Bewafaa | 150 | 2005-02-25 | 7 | Anjali, living in Canada with her parents, hides her relationship to the musician Raja from her parents. When her sister dies during the birth of twins, she abandons her love, marries the widower Aditja and moves to New Delhi to become the new mother of her sisters children. Three years later, when Raja visits Delhi on his concert tour their love inflames again. |
| 2 | Veer-Zaara | 150 | 2004-11-12 | 7 | The story of the love between Veer Pratap Singh, an Indian, and Zaara Hayaat Khan, a Pakistani...a love so great it knows no boundaries... |
| 3 | Mujhse Shaadi Karogi | 156 | 2004-01-08 | 3 | Sameer, fast at losing his temper is re-located to Goa where he falls in love with Rani. But Sameers new roommate Sunny, has some plans of his own. |
| 4 | Kabhi Alvida Naa Kehna | 192 | 2006-08-11 | 4 | There are three kinds of married people in the world. First, whose marriages are arranged, Ive never quite understood that, but I.m sure they know what theyre doing. Then there are those who fall in love and marry their soul mates. These few, I believe, are the most fortunate people in the world. And lastly, there are those couples that marry for their parents, for money or play it safe and marry a friend. These are the most unfortunate ones in the world. and they don.t even know it. Until one day, riding the fast train of life they run into their soulmate, and are faced with the hardest question of all. What do you do when you meet the love of your life and youre married to someone else? What do you do? What do you do? |
| 5 | Insan | 120 | 2005-01-14 | 1 | Amjad is a auto-rickshaw driver in Bombay, and lives with his mother in a small tenement in Bombays slums. His mother had originally given birth to another son, Munna, who has been missing for several years. She is over-joyed when Munna returns home, and both she and Amjad welcome him, and celebrate. They do not know that Munna is now known as Azhar Khan - a member of a Jihaadi terrorist group, who is now using Amjad and his mother as a cover, in order to carry out a deadly and daring terrorist attack. |
| 6 | Pyaar Mein Twist | 140 | 2005-09-02 | 5 | With their respective childrens marriages about to take place, two middle-aged single parents fall in love. |
| 7 | Prince | 138 | 2010-04-09 | 1 | One of the savviest thieves in the world commits the biggest heist of his life. He wakes up next morning to realize he has a gunshot wound on his arm that he doesnt remember getting. In his quest to find answers he discovers his name is Prince, he used to work for a man named Sarang and his girlfriends name is Maya. He is being hunted by the secret service of India- I Grip, the CBI and the biggest white collared criminals in the world. He is the most wanted man in the country because only he knows the whereabouts of the heist, which contains a secret that is linked not only to his loss of memory but threatens the future of the Human Race. Prince can rely only on his razor sharp instincts to salvage himself. The web of deception spins and Prince gets even more entangled. Nothing and no one can be trusted. Every day he meets a new girl claiming to be Maya... Every cop and criminal in the country is out for his blood... He doesnt remember where he has hidden the Heist... He doesnt know which side of the law he is on... His life is at stake... The future of mankind is at stake... Prince has no one but himself to trust ... Failure is not an option... He has just 5 Days of his life... Time is running out will he survive |
| 8 | Dostana | 145 | 2008-11-14 | 3 | Kunal and Sameer, two straight guys who pretend to be a gay couple to secure an posh Miami apartment, but both of them fall for their gorgeous room-mate Neha, hilarity ensures as they strive to convince one and all they are gay! While secretly trying to win Nehas heart. |
| 9 | Jab We Met | 138 | 2007-10-26 | 7 | A depressed wealthy businessman finds his life changing after he meets a spunky and care-free young woman. |
| 10 | Krishna Cottage | 130 | 2004-05-13 | 8 | During the year 1990 Professor-Author Siddharth Das releases his first book Kahi Unkahi Baatein which contain a total of 10 short stories albeit with the last story Krishna Cottage without any ending. A total of fifty thousand copies are printed and distributed to five different cities in India. Three trucks carrying these copies meet with mysterious accidents, two toppling off cliffs, and one catching fire; and one ship sinks losing its entire consignment. One copy of this book is found and kept under lock and key in the store-room of J.C. College. Then in 2002 collegians Akshay, Kabir, Nupur, Pulit, Shanti, and Manav, induct new-comer, Disha, to their group, and also invite her to the engagement ceremony of Shanti and Manav. The group will soon find out that her induction will set loose a chain of events that will throw their lives in jeopardy as she may be a ferocious shape-changing entity that is out to claim her lover, Amar Khanna, from a past life, and also prevent anyone from reading the last chapter of Professor Das book. |
| 11 | Rab Ne Bana Di Jodi | 167 | 2008-12-12 | 4 | A middle-aged man who lost his love for life rediscovers it by assuming a new identity in order to rekindle the romantic spark in his marriage. |
| 12 | Dhoom:2 | 152 | 2006-12-24 | 1 | ACP Jai Dixit and his best friend Ali takes on an international thief known only as Mr. A. Will they be able to catch him? |
| 13 | Vaada | 135 | 2005-01-07 | 10 | A blind husband. An obsessed lover. And the missing body of a murdered wife...? |
| 14 | Koi... Mil Gaya | 171 | 2003-08-08 | 9 | A developmentally disabled young man tries to continue the work his father did in communicating with extra-terrestrials from outer space, which leads to something miraculous and wonderful. |

## 4.10 tblRentalHistory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MovieID** | **CopyValue** | **AccountCustomerID** | **DateRented** | **DateDue** | **DateReturned** |
| 1 | A | 1 | 2010-01-21 | 2010-01-22 | 2010-01-22 |
| 5 | A | 1 | 2010-04-20 | 2010-04-21 | NULL |
| 7 | B | 2 | 2011-04-20 | 2011-04-21 | NULL |
| 12 | A | 4 | 2010-08-11 | 2010-09-11 | 2010-08-25 |
| 10 | A | 3 | 2009-01-31 | 2009-02-01 | 2009-02-01 |
| 9 | B | 5 | 2011-04-21 | 2011-04-22 | 2011-04-22 |
| 14 | B | 6 | 2011-04-23 | 2011-04-24 | NULL |

# 5. Data Dictionary

The results from 5.2 can be obtained through the web interface in the reports in addition to this document.

## 5.1 Domain Documentation



## 5.2 Field Documentation



# 6. Basic Queries

## 6.1 Basic Select Queries

**1. Find all movies that are less than 2 hours 10 mins.**

SELECT \*

FROM tblMovie

WHERE RunningLength<130;



**2. Find all customers who do not have a middle initial or name.**

SELECT \*

FROM tblCustomer

WHERE MName IS NULL;



**3. Find all movies that were purchased for less than $10.**

SELECT \*

FROM tblCopyInformation

WHERE PurchaseCost<10;



**4. Get the list of all the different types of Genres**

SELECT \* FROM tblGenre;



## 6.2 Basic Order by and Group By Queries

**1. Get all Movies sorted by Movie Name A-Z.**

SELECT MovieID, Title, RunningLength

FROM tblMovie ORDER BY Title ASC;



**2. Find the MovieID for all movies that have 2 or more copies.**

SELECT MovieID, COUNT(\*) AS TOTAL\_COPIES

FROM tblCopyInformation

GROUP BY MovieID

HAVING COUNT(\*) > 1

ORDER BY COUNT(\*) DESC;

**3. All customers in last name, first name, middle name in A-Z order, that have FName and Lname starting with ‘D’.**

SELECT \*

FROM tblCustomer

WHERE LName like 'D%' AND FName like 'D%'

ORDER BY LName, FName, Mname;



**4. List all customers who are younger than 18 in descending order.**

SELECT C.CustomerID, C.Fname, C.Lname, ((date\_format(now(),'%Y') - date\_format(C.BirthDate,'%Y')) - (date\_format(now(),'00-%m-%d') < date\_format(C.BirthDate,'00-%m-%d'))) AS age

FROM tblCustomer C

WHERE ((date\_format(now(),'%Y') - date\_format(C.BirthDate,'%Y')) - (date\_format(now(),'00-%m-%d') < date\_format(C.BirthDate,'00-%m-%d'))) < 18

ORDER BY 2 desc;



# 7. Advanced Queries

## 7.1 Join Queries

**1. List all customers on an account, including the relationship type:**

SELECT C.FName, C.LName, C.BirthDate, TART.Name Acct\_Relation\_Type, TAC.DateAdded, TA.AccountID, TAP.Name Acct\_Plan

FROM tblCustomer C

INNER JOIN tblAccountCustomer TAC ON TAC.CustomerID = C.CustomerID

INNER JOIN tblAccountRelationshipType TART ON TART.AcctRelationshipID = TAC.AcctRelationshipID

INNER JOIN tblAccount TA ON TA.AccountID = TAC.AccountID

INNER JOIN tblAccountPlan TAP ON TAP.PlanID = TA.PlanID;



**2. Show all Movies overdue:**

SELECT M.Title, CI.CopyValue, RH.DateRented, RH.DateDue, DATEDIFF(sysdate(),RH.DateDue) 'Days Late', TC.FName, TC.LName, TART.Name RelationshipType, TAP.Name 'Account Plan', TAP.Description 'Account Plan Description'

FROM tblMovie M

JOIN tblCopyInformation CI ON CI.MovieID = M.MovieID

JOIN tblRentalHistory RH ON RH.MovieID = CI.MovieID AND RH.CopyValue = CI.CopyValue

JOIN tblAccountCustomer TAC ON TAC.AccountCustomerID = RH.AccountCustomerID

JOIN tblAccount TA ON TA.AccountID = TAC.AccountID

JOIN tblAccountPlan TAP ON TAP.PlanID = TA.PlanID

JOIN tblCustomer TC on TC.CustomerID = TAC.CustomerID

JOIN tblAccountRelationshipType TART ON TART.AcctRelationshipID = TAC.AcctRelationshipID

WHERE RH.DateReturned IS NULL;



**3. Show all movies detailing the Genre, Media type, and Copy:**

Select M.Title, TG.Name Genre, TMT.Name Media\_Type, TCI.CopyValue

FROM tblMovie M

Inner join tblCopyInformation TCI on TCI.MovieID = M.MovieID

Inner join tblMediaType TMT on TMT.MediaTypeID = TCI.MediaTypeID

Inner join tblGenre TG on TG.GenreID = M.GenreID

ORDER BY M.Title, TCI.CopyValue;



**4. Find Breakdown of number of movies per Genre.**

SELECT TG.Name Genre, COUNT(M.MovieID) Num\_of\_Genre

FROM tblGenre TG

LEFT JOIN tblMovie M

ON TG.GenreID = M.GenreID

GROUP BY TG.Name

ORDER BY 2 DESC, TG.Name ASC;

## 7.2 Subqueries, IN/EXISTS Queries

**1. Find the number of movies rented per each Customer.**

SELECT TC.FName,

TC.LName,

Coalesce(TAC.MoviesRented,0) Movies\_Rented

FROM tblCustomer TC

LEFT JOIN (SELECT CustomerID, count(\*) MoviesRented

FROM tblRentalHistory T1

INNER JOIN tblAccountCustomer T2

ON T2.AccountCustomerID = T1.AccountCustomerID

GROUP BY CustomerID) TAC

ON TAC.CustomerID = TC.CustomerID

GROUP BY TC.FName, TC.Lname

ORDER BY 3 desc;

**2. Find all the movies where there is not a Blue Ray Disk Copy**

SELECT Title

FROM tblMovie M

WHERE NOT EXISTS (SELECT 1

FROM tblCopyInformation TCI

WHERE TCI.MovieID = M.MovieID

AND TCI.MediaTypeID = 4);

**3. Find all the movies where the Genre is not in a specific set of genres**

SELECT MovieID, Title

FROM tblMovie M

WHERE M.GenreID NOT IN (1,5,7,4)

ORDER BY Title DESC;

**4. Find all Action, Drama, or Suspense movies, where the Title has “NA” in it somewhere.**

SELECT M.Title, TG.Name Genre, M.ReleaseDate, M.RunningLength

FROM tblMovie M

INNER JOIN tblGenre TG

ON TG.GenreID = M.GenreID

WHERE lower(M.Title) like '%na%' and M.GenreID in (1,4,10);



# 8. Indexes

## 8.1 Faster Customer Lookup by Last Name

**Index created for Query #3 in Section 6B. Allows for faster customer lookup by Name (first).**

CREATE INDEX index\_empfname ON tblCustomer(FName);

## 8.2 Faster Customer Lookup by First Name

**Index created for Query #3 in Section 6B. Allows for faster customer lookup by Name (last).**

CREATE INDEX index\_emplname ON tblCustomer(LName);

## 8.3 Faster Movie Lookup by Running Length

**Index created for Query #1 in Section 6A. Allows for faster movie lookup by Running Length.**

CREATE INDEX index\_runlength ON tblMovie(RunningLength);

## 8.4 Faster Movie Lookup by Title

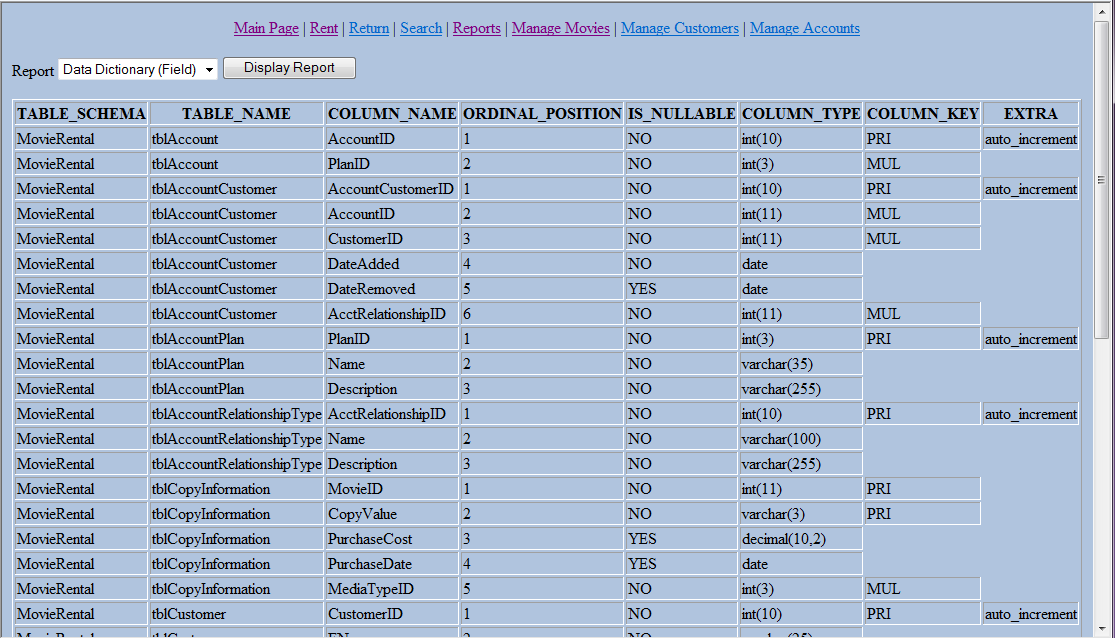
**Index created for Query #4 in Section 7B. Allows for faster movie lookup by searching for the Title.**

CREATE INDEX index\_title ON tblMovie(Title);

# 9. Showing Query Results in a Web page.

This is a simple export of data into a grid view. 5 pre-made queries are available for use.

## 9.1 Screenshot



## 9.2 Code

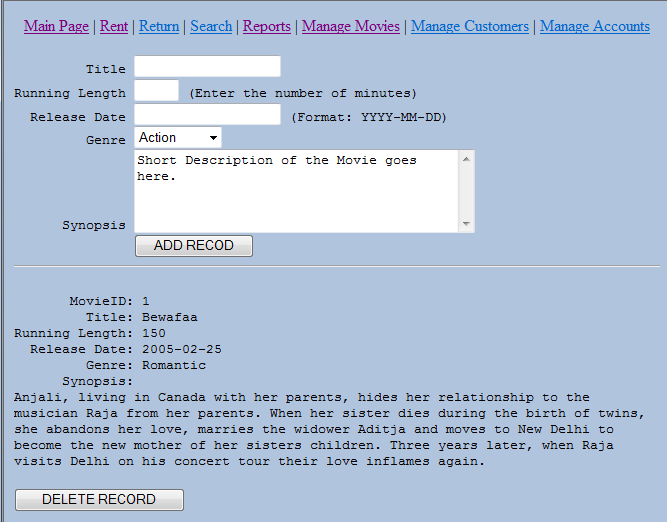
|  |
| --- |
| <?php  //Include the Database Login Information  require\_once 'login.php';  if (isset($\_POST['reportselector']))  {  //Store the returned value from the Report Selector  //into rsrv = Report Selector Return value  $rsrv = $\_POST['reportselector'];  //Open the connection using the information from the login.php file  $db\_server = mysql\_connect($db\_hostname,$db\_username, $db\_password);  //attempt to connect to the MySQL server  if (!$db\_server) die("Unable to Connect to MySQL:" . mysql\_error());  //Attempt to open the database specified in login.php  mysql\_select\_db($db\_database) or die("Unable to select database: " . mysql\_error());  //In-Line query select statement  if ($rsrv=="allmovies")  $query = "SELECT \* FROM tblMovie";  elseif ($rsrv=="allcustomers")  $query = "SELECT \* FROM tblCustomer";  elseif ($rsrv=="moviesoverdue")  {  $query = "SELECT M.Title, CI.CopyValue, RH.DateRented, RH.DateDue, DATEDIFF(sysdate(),RH.DateDue) 'Days Late', ";  $query .= "TC.FName, TC.LName, TART.Name RelationshipType, TAP.Name 'Account Plan', TAP.Description 'Account Plan Description' ";  $query .= "FROM tblMovie M ";  $query .= "INNER JOIN tblCopyInformation CI ON CI.MovieID = M.MovieID ";  $query .= "INNER JOIN tblRentalHistory RH ON RH.MovieID = CI.MovieID AND RH.CopyValue = CI.CopyValue ";  $query .= "INNER JOIN tblAccountCustomer TAC ON TAC.AccountCustomerID = RH.AccountCustomerID ";  $query .= "INNER JOIN tblAccount TA ON TA.AccountID = TAC.AccountID ";  $query .= "INNER JOIN tblAccountPlan TAP ON TAP.PlanID = TA.PlanID ";  $query .= "INNER JOIN tblCustomer TC on TC.CustomerID = TAC.CustomerID ";  $query .= "INNER JOIN tblAccountRelationshipType TART ON TART.AcctRelationshipID = TAC.AcctRelationshipID ";  $query .= "WHERE RH.DateReturned IS NULL";  }  elseif ($rsrv=="allaccounts")  $query = "SELECT \* FROM tblAccount";  elseif ($rsrv=="FieldDoc")  {  $query = "SELECT TABLE\_SCHEMA, TABLE\_NAME, COLUMN\_NAME, ORDINAL\_POSITION, IS\_NULLABLE, COLUMN\_TYPE, COLUMN\_KEY, EXTRA ";  $query .= "FROM information\_schema.COLUMNS C ";  $query .= "WHERE TABLE\_SCHEMA = 'MovieRental'";  }  else  $query = "SELECT '' FROM tblAccount";    $result = mysql\_query($query);  if (!$result) die("Database access failed: " . mysql\_error());  //Display results on screen  $rows = mysql\_num\_rows($result);  $cols = mysql\_num\_fields($result);  //Create Header Row  $reportoutput = "<table border='1'><tr>";    for ($i = 0; $i < $cols; ++$i)  {  $reportoutput .= "<th>" . mysql\_field\_name($result,$i) . "</th>";  }    $reportoutput .= "</tr>";    //Add rows/records  for ($j = 0; $j < $rows ; ++$j)  {  $reportoutput .= "<tr>";  $row = mysql\_fetch\_row($result);  for ($i = 0;$i < $cols; ++$i)  {  $reportoutput .= "<td>" . $row[$i] . "</td>";  }  $reportoutput .= "</tr>";  }  $reportoutput .= "</table>";  //$reportoutput = $\_POST['reportselector'];  }  else $reportoutput = "";  echo <<<\_END  <html>  <head>  <link rel="stylesheet" type="text/css" href="generic.css" />  <title>Movie Rental - Reports</title>  </head>  <body>  <center>  <a href='mainpage.php'>Main Page</a> |  <a href='rent.php'>Rent</a> |  <a href='return.php'>Return</a> |  <a href='search.php'>Search</a> |  <a href='reports.php'>Reports</a> |  <a href='managemovies.php'>Manage Movies</a> |  <a href='managecustomers.php'>Manage Customers</a> |  <a href='manageaccounts.php'>Manage Accounts</a><br />  </center>  <form method="post" action="reports.php">  Report <select name="reportselector" size="1">  <option value="allmovies">All Movies</option>  <option value="allcustomers">All Customers</option>  <option value="allaccounts">All Accounts</option>  <option value="moviesoverdue">Movies Over Due</option>  <option value="FieldDoc">Data Dictionary (Field)</option>  </select>  <input type="submit" value="Display Report" />  </form>  <p>$reportoutput</p>  </body>  </html>  \_END;  ?> |

# 10. Adding records to and deleting records from a database.

Top section allows adding new movies.

This Demonstrates INSERTING, DELETING, SELECTING from within a form, in addition to a TEXTAREA and SELECT list.

## 10.1 Sceenshot



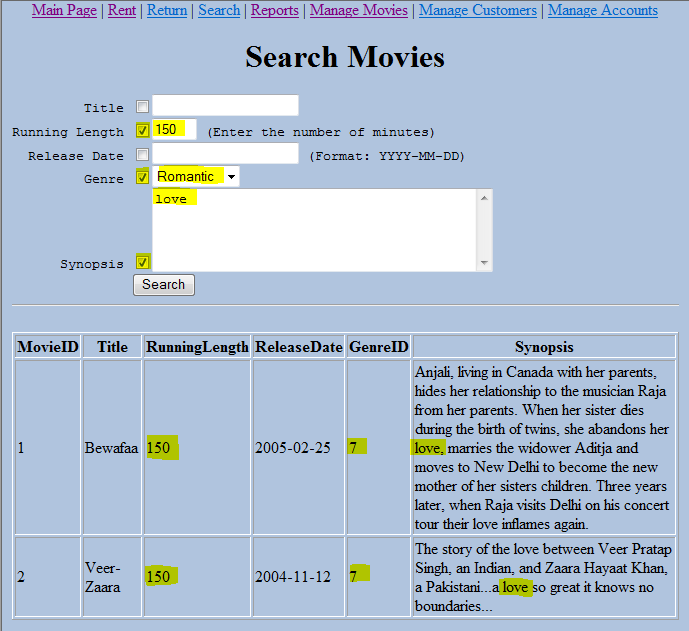
## 10.2 Code

|  |
| --- |
| <?php //managemovies.php  require\_once 'login.php';  $db\_server = mysql\_connect($db\_hostname, $db\_username, $db\_password);    if (!$db\_server) die("Unable to connect to MySQL: " .mysql\_error());    mysql\_select\_db($db\_database, $db\_server)  or die("Unable to select database: " . mysql\_error());    //Get the Genre into the drop down list  $query = "SELECT GenreID, Name FROM tblGenre";  $genreDDL = create\_ddl('GenreID',$query);    //Delete a record if they clicked delete  if (isset($\_POST['delete']) && isset($\_POST['MovieID']))  {  $MovieID = get\_post('MovieID');  $query = "DELETE FROM tblMovie WHERE MovieID = $MovieID";    if (!mysql\_query($query, $db\_server))  echo "DELETE failed: $query<br />" . mysql\_error() . "<br /><br />";  }    //Add a record if they clicked add  if (isset($\_POST['Title']) &&  isset($\_POST['RunLength'])  )  {  $Title = get\_post('Title');  $RunLength = get\_post('RunLength');  $ReleaseDate = get\_post('ReleaseDate');  $GenreID = get\_post('GenreID');  $Synopsis = get\_post('Synopsis');    $query = "INSERT INTO tblMovie VALUES" .  "(null,'$Title',$RunLength,'$ReleaseDate',$GenreID, '$Synopsis')";    if (!mysql\_query($query, $db\_server))  echo "INSERT failed: $query<br />" . mysql\_error() . "<br /><br />";  }    //HTML General  echo <<<\_END  <html>  <head>  <link rel="stylesheet" type="text/css" href="generic.css" />  <title>Movie Rental - Movie Management</title>  </head>  <body>  <center>  <a href='mainpage.php'>Main Page</a> |  <a href='rent.php'>Rent</a> |  <a href='return.php'>Return</a> |  <a href='search.php'>Search</a> |  <a href='reports.php'>Reports</a> |  <a href='managemovies.php'>Manage Movies</a> |  <a href='managecustomers.php'>Manage Customers</a> |  <a href='manageaccounts.php'>Manage Accounts</a><br />  </center>  \_END;  //HTML Form  echo <<<\_END  <form action="managemovies.php" method="post">  <pre>  Title <input type="text" name="Title" />  Running Length <input type="text" size="3" name="RunLength" /> (Enter the number of minutes)  Release Date <input type ="text" name="ReleaseDate" /> (Format: YYYY-MM-DD)  Genre $genreDDL  Synopsis <textarea rows="5" cols="40" name="Synopsis" />Short Description of the Movie goes here.</textarea>  <input type="submit" value="ADD RECOD" />  <hr />  </pre></form>  \_END;  $query = "SELECT M.MovieID, M.Title, M.RunningLength, M.ReleaseDate, G.Name, M.Synopsis FROM tblMovie M ";  $query .= "INNER JOIN tblGenre G ON G.GenreID = M.GenreID";  $result = mysql\_query($query);  if (!$result) die ("Database access failed: " . mysql\_error());  $rows = mysql\_num\_rows($result);  for($j = 0 ; $j < $rows ; ++$j)  {  $row = mysql\_fetch\_row($result);  echo <<<\_END    <pre>  MovieID: $row[0]  Title: $row[1]  Running Length: $row[2]  Release Date: $row[3]  Genre: $row[4]  Synopsis: <div class="wraptext"><p class="wraptext">$row[5]</p></div>  </pre>  <form action="managemovies.php" method="post">  <input type="hidden" name="delete" value="yes" />  <input type="hidden" name="MovieID" value="$row[0]" />  <input type="submit" value="DELETE RECORD" />  </form>  \_END;  }  echo <<<\_END  </body>  </html>  \_END;  mysql\_close($db\_server);  function get\_post($var)  {  return sanitizeMySQL($\_POST[$var]);  }  //Function: create\_ddl  //Purpose: Create an HTML Drop Down List  //Requires: Input of a Select Statement with ID column, Name Column  // Current Connection to MySQL database  //Example: $query = "SELECT ID, Name FROM lookup\_table";  //Usage: $HTML = create\_ddl($query);  function create\_ddl($ddl\_name, $var)  {  $result = mysql\_query($var);  if (!$result) die ("Database access failed: " . mysql\_error());  $rows = mysql\_num\_rows($result);  $html\_ddl = '<select name="' . $ddl\_name . '" size="1">';  for($j = 0 ; $j < $rows ; ++$j)  {  $row = mysql\_fetch\_row($result);  $html\_ddl .= '<option value='.$row[0].'>'.$row[1].'</option>';  }  $html\_ddl .= '</select>';  return $html\_ddl;  }  function sanitizeString($var)  {  $var = stripslashes($var);  $var = htmlentities($var);  $var = strip\_tags($var);  return $var;  }  function sanitizeMySQL($var)  {  $var = mysql\_real\_escape\_string($var);  $var = sanitizeString($var);  return $var;  }  ?> |

# 11. Retrieving a Record using a form.

The code allows for multiple search criteria on the movie. Demonstrates use of Checkbox, Text Area, and Select list. Any of the 5 items can be check marked, or they all can. Search by any or all of the fields.

## 11.1 Screenshot



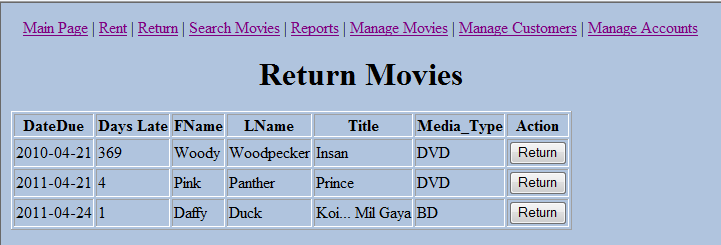
## 11.2 Code

|  |
| --- |
| <?php //managemovies.php  require\_once 'login.php';  $db\_server = mysql\_connect($db\_hostname, $db\_username, $db\_password);    if (!$db\_server) die("Unable to connect to MySQL: " .mysql\_error());    mysql\_select\_db($db\_database, $db\_server)  or die("Unable to select database: " . mysql\_error());    //Get the Genre into the drop down list  $query = "SELECT GenreID, Name FROM tblGenre";  $genreDDL = create\_ddl('GenreID',$query);    //Search for a record if they clicked search  if (isset($\_POST['chk\_search']) )  {  $chk\_in = $\_POST['chk\_search'];  $Title = get\_post('Title');  $RunLength = (int)get\_post('RunLength');  $ReleaseDate = get\_post('ReleaseDate');  $GenreID = get\_post('GenreID');  $Synopsis = get\_post('Synopsis');    $query = "SELECT \* " .  "FROM tblMovie M " .  "WHERE 1=1 ";    foreach ($chk\_in as $i => $value)  {  if ($chk\_in[$i] == 'Title' && isset($\_POST['Title']))  $query .= " AND M.Title like '%$Title%' ";  elseif ($chk\_in[$i] == 'RunLength' && isset($\_POST['RunLength']))  $query .= " AND M.RunningLength = $RunLength ";  elseif ($chk\_in[$i] == 'ReleaseDate' && isset($\_POST['ReleaseDate']))  $query .= " AND M.ReleaseDate = '$ReleaseDate' ";  elseif ($chk\_in[$i] == 'Genre' && isset($\_POST['GenreID']))  $query .= " AND M.GenreID = $GenreID ";  elseif ($chk\_in[$i] == 'Synopsis' && isset($\_POST['Synopsis']))  $query .= " AND M.Synopsis like '%$Synopsis%' ";  }    $htmloutput = table\_query\_output($query);  }  else  $htmloutput = "";    //HTML General  echo <<<\_END  <html>  <head>  <link rel="stylesheet" type="text/css" href="generic.css" />  <title>Movie Rental - Search Movies</title>  </head>  <body>  <center>  <a href='mainpage.php'>Main Page</a> |  <a href='rent.php'>Rent</a> |  <a href='return.php'>Return</a> |  <a href='search.php'>Search</a> |  <a href='reports.php'>Reports</a> |  <a href='managemovies.php'>Manage Movies</a> |  <a href='managecustomers.php'>Manage Customers</a> |  <a href='manageaccounts.php'>Manage Accounts</a><br />  <h1> Search Movies </h1>  </center>  \_END;  //HTML Form  echo <<<\_END  <form action="searchmovies.php" method="post">  <pre>  Title <input type="checkbox" name="chk\_search[]" value="Title" /><input type="text" name="Title" />  Running Length <input type="checkbox" name="chk\_search[]" value="RunLength" /><input type="text" size="3" name="RunLength" /> (Enter the number of minutes)  Release Date <input type="checkbox" name="chk\_search[]" value="ReleaseDate" /><input type ="text" name="ReleaseDate" /> (Format: YYYY-MM-DD)  Genre <input type="checkbox" name="chk\_search[]" value="Genre" />$genreDDL  Synopsis <input type="checkbox" name="chk\_search[]" value="Synopsis" /><textarea rows="5" cols="40" name="Synopsis" />Short Description of the Movie goes here.</textarea>  <input type="submit" value="Search" />  <hr />  </pre></form>  <p>$htmloutput</p>  </body>  </html>  \_END;  mysql\_close($db\_server);  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // FUNCTIONS  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  function get\_post($var)  {  return sanitizeMySQL($\_POST[$var]);  }  function table\_query\_output($var)  {  $result = mysql\_query($var);    if (!$result) die("Database access failed: " . mysql\_error());    //Display results on screen  $rows = mysql\_num\_rows($result);  $cols = mysql\_num\_fields($result);    if ($rows >0)  {  //Create Header Row  $reportoutput = "<table border='1'><tr>";    for ($i = 0; $i < $cols; ++$i)  {  $reportoutput .= "<th>" . mysql\_field\_name($result,$i) . "</th>";  }    $reportoutput .= "</tr>";    //Add rows/records  for ($j = 0; $j < $rows ; ++$j)  {  $reportoutput .= "<tr>";  $row = mysql\_fetch\_row($result);  for ($i = 0;$i < $cols; ++$i)  {  $reportoutput .= "<td>" . $row[$i] . "</td>";  }  $reportoutput .= "</tr>";  }  $reportoutput .= "</table>";  }  else  $reportoutput = "No Results, try using less information in your search.";  return $reportoutput;  }  //Function: create\_ddl  //Purpose: Create an HTML Drop Down List  //Requires: Input of a Select Statement with ID column, Name Column  // Current Connection to MySQL database  //Example: $query = "SELECT ID, Name FROM lookup\_table";  //Usage: $HTML = create\_ddl($query);  function create\_ddl($ddl\_name, $var)  {  $result = mysql\_query($var);  if (!$result) die ("Database access failed: " . mysql\_error());  $rows = mysql\_num\_rows($result);  $html\_ddl = '<select name="' . $ddl\_name . '" size="1">';  for($j = 0 ; $j < $rows ; ++$j)  {  $row = mysql\_fetch\_row($result);  $html\_ddl .= '<option value='.$row[0].'>'.$row[1].'</option>';  }  $html\_ddl .= '</select>';  return $html\_ddl;  }  function sanitizeString($var)  {  $var = stripslashes($var);  $var = htmlentities($var);  $var = strip\_tags($var);  return $var;  }  function sanitizeMySQL($var)  {  $var = mysql\_real\_escape\_string($var);  $var = sanitizeString($var);  return $var;  }  ?> |

# 12. Updating a Record using a form.

Clicking the Return button will return the individual movie currently rented out.

## 12.1 Screenshot



## 12.2 Code

|  |
| --- |
| <?php //return.php  require\_once 'login.php';  $db\_server = mysql\_connect($db\_hostname, $db\_username, $db\_password);    if (!$db\_server) die("Unable to connect to MySQL: " .mysql\_error());    mysql\_select\_db($db\_database, $db\_server)  or die("Unable to select database: " . mysql\_error());  //Update Rental History  if (isset($\_POST['MovieID']) &&  isset($\_POST['CopyValue']) &&  isset($\_POST['AccountCustomerID']) &&  isset($\_POST['DateRented'])  )  {  $MovieID = get\_post('MovieID');  $CopyValue = get\_post('CopyValue');  $AccountCustomerID = get\_post('AccountCustomerID');  $DateRented = get\_post('DateRented');    $query = "UPDATE tblRentalHistory " .  "SET DateReturned = sysdate() " .  "WHERE MovieID = $MovieID " .  "AND CopyValue = '$CopyValue' " .  "AND AccountCustomerID = $AccountCustomerID " .  "AND DateRented = '$DateRented' ";    if (!mysql\_query($query, $db\_server))  echo "UPDATE failed: $query<br />" . mysql\_error() . "<br /><br />";  }    //HTML General  echo <<<\_END  <html>  <head>  <link rel="stylesheet" type="text/css" href="generic.css" />  <title>Movie Rental - Return a movie</title>  </head>  <body>  <center>  <a href='mainpage.php'>Main Page</a> |  <a href='rent.php'>Rent</a> |  <a href='return.php'>Return</a> |  <a href='searchmovies.php'>Search Movies</a> |  <a href='reports.php'>Reports</a> |  <a href='managemovies.php'>Manage Movies</a> |  <a href='managecustomers.php'>Manage Customers</a> |  <a href='manageaccounts.php'>Manage Accounts</a><br />  <h1> Return Movies</h1>  </center>    \_END;  //HTML Form  $query = "SELECT TRH.MovieID, TRH.CopyValue, TRH.AccountCustomerID, TRH.DateRented, TRH.DateDue, " .  "DATEDIFF(sysdate(),TRH.DateDue) 'Days Late', C.FName, C.LName, M.Title, TMT.Name Media\_Type " .  "FROM tblRentalHistory TRH " .  "INNER JOIN tblCopyInformation TCI ON TCI.MovieID = TRH.MovieID AND TCI.CopyValue = TRH.CopyValue " .  "INNER JOIN tblMovie M ON M.MovieID = TCI.MovieID " .  "INNER JOIN tblMediaType TMT ON TMT.MediaTypeID = TCI.MediaTypeID " .  "INNER JOIN tblAccountCustomer TAC ON TAC.AccountCustomerID = TRH.AccountCustomerID " .  "INNER JOIN tblCustomer C ON C.CustomerID = TAC.CustomerID " .  "WHERE TRH.DateReturned IS NULL AND DATEDIFF(sysdate(),TRH.DateDue)>=0 ";    $htmlOutput = table\_query\_output($query,4);    echo <<<\_END  $htmlOutput  </body>  </html>  \_END;  mysql\_close($db\_server);  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // FUNCTIONS  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  function get\_post($var)  {  return sanitizeMySQL($\_POST[$var]);  }  function sanitizeString($var)  {  $var = stripslashes($var);  $var = htmlentities($var);  $var = strip\_tags($var);  return $var;  }  function sanitizeMySQL($var)  {  $var = mysql\_real\_escape\_string($var);  $var = sanitizeString($var);  return $var;  }  function table\_query\_output($var, $startCol)  {  $result = mysql\_query($var);    if (!$result) die("Database access failed: " . mysql\_error());    //Display results on screen  $rows = mysql\_num\_rows($result);  $cols = mysql\_num\_fields($result);    if ($rows >0)  {  //Create Header Row  $reportoutput = "<table border='1'><tr>";    for ($i = $startCol; $i < $cols; ++$i)  {  $reportoutput .= "<th>" . mysql\_field\_name($result,$i) . "</th>";  }    $reportoutput .= "<th>Action</th>";    $reportoutput .= "</tr>";    //Add rows/records  for ($j = 0; $j < $rows ; ++$j)  {  $reportoutput .= '<tr><form action="return.php" method="post">'; //Add a form to each row so we can add a Return button to each row.  $row = mysql\_fetch\_row($result);  for ($i = $startCol;$i < $cols; ++$i)  {  $reportoutput .= "<td>" . $row[$i] . "</td>";  }  //Store the primary key information  $reportoutput .= '<td><input type="hidden" name="MovieID" value="' . $row[0] . '" />' .  '<input type="hidden" name="CopyValue" value="' . $row[1] . '" />' .  '<input type="hidden" name="AccountCustomerID" value="' . $row[2] . '" />' .  '<input type="hidden" name="DateRented" value="' . $row[3] . '" />' .  '<input type="submit" value="Return" /></td>';    $reportoutput .= "</form></tr>";  }  $reportoutput .= "</table>";  }  else  $reportoutput = "There are no movies currently due or late";  return $reportoutput;  }  ?> |