|  |  |  |
| --- | --- | --- |
|  | **Dalhousie University**  **Faculty of Computer Science** | Image result for dalhousie logo |

# CSCI 2141 Intro to Database Systems Fall-2016

**Assignment 2 [50 marks]**

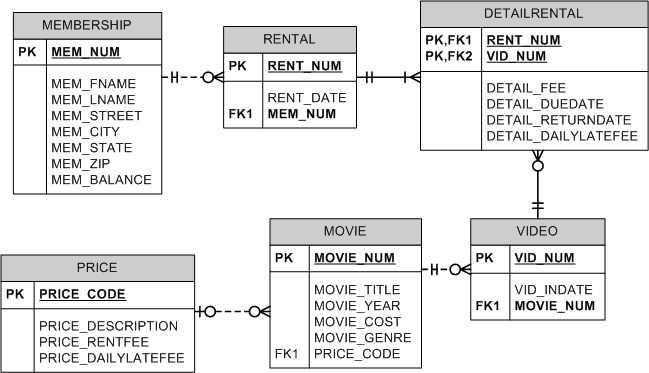
**Submission Deadline: TUESDAY, 11th OCTOBER at 12:00 noon**

**PROBLEM STATEMENT:**

Carefully study the Data Model given below. You are required to implement this database using MySQL, populate the tables with the data provided, and use that database to answer the Assignment Tasks that follow.

***DVDVidRental Database:***

DVDVidRental is a new company that provides DVD Video Rentals. DVDVidRental can own several copies (VIDEO) of each movie (MOVIE). For example, the store may have 10 copies of the movie “The Shawshank Redemption”. “The Shawshank Redemption” would be one MOVIE and each copy would be a VIDEO. A rental transaction (RENTAL) involves one or more videos being rented to a member (MEMBERSHIP). A video can be rented many times over its lifetime, therefore, there is a M:N relationship between RENTAL and VIDEO. DETAILRENTAL is the bridge table to resolve this relationship. The complete ERD is provided in Figure below:



**SUBMISSION INSTRUCTIONS:**

* Use this document as an answer template.
* Complete and submit this document with your answers filled in (scripts and screenshots of the results of each task) through Brightspace, before the submission deadline.
* Remember to rename the document with your ID (e.g. ***B00123456.doc*** *or* ***docx***)
* *Doing with this assignment on your own will immensely improve your knowledge of SQL (The more you struggle with it, the more you learn. Do ask for help if you get “really” stuck)*

**ASSIGNMENT TASKS:**

***Part 1: CREATING THE TABLES [10 marks]***

1. Write the SQL script to create the table structures for the entities shown in the above Figure. The structures should contain the attributes specified in the ERD. Use data types that would be appropriate for the data that will need to be stored in each attribute. Enforce primary key and foreign key constraints as indicated by the ERD. Use this script to create this database in MySQL.

*(Hint: You may need to consider the sequence in which to create the tables)*

*CREATE TABLE MEMBERSHIP(*

*MEM\_NUM INTEGER NOT NULL UNIQUE,*

*MEM\_FNAME VARCHAR(10) NOT NULL,*

*MEM\_LNAME VARCHAR(10) NOT NULL,*

*MEM\_STREET VARCHAR(10) NOT NULL,*

*MEM\_CITY VARCHAR(10) NOT NULL,*

*MEM\_STATE VARCHAR(10) NOT NULL,*

*MEM\_ZIP VARCHAR(10) NOT NULL,*

*MEM\_BALANCE VARCHAR(10) NOT NULL,*

*PRIMARY KEY (MEM\_NUM));*

*CREATE TABLE RENTAL(*

*RENT\_NUM INTEGER NOT NULL UNIQUE,*

*RENT\_DATE date NOT NULL,*

*MEM\_NUM INTEGER,*

*FOREIGN KEY (MEM\_NUM)REFERENCES MEMBERSHIP(MEM\_NUM));*

*CREATE TABLE PRICE(*

*PRICE\_CODE INTEGER NOT NULL UNIQUE,*

*PRICE\_DESCRIPTION VARCHAR(10) NOT NULL,*

*PRICE\_RENTFEE VARCHAR(10) NOT NULL,*

*PRICE\_DAILYLATEFEE VARCHAR(10) NOT NULL,*

*PRIMARY KEY (PRICE\_CODE));*

*CREATE TABLE MOVIE(*

*MOVIE\_NUM INTEGER NOT NULL UNIQUE,*

*MOVIE\_TITLE VARCHAR(20) NOT NULL,*

*MOVIE\_YEAR date NOT NULL,*

*MOVIE\_COST VARCHAR(10) NOT NULL,*

*MOVIE\_GENRE VARCHAR(10) NOT NULL,*

*PRICE\_CODE INTEGER,*

*PRIMARY KEY(MOVIE\_NUM),*

*FOREIGN KEY(PRICE\_CODE)REFERENCES PRICE(PRICE\_CODE));*

*CREATE TABLE VIDEO(*

*VID\_NUM INTEGER NOT NULL UNIQUE,*

*VID\_INDATE date NOT NULL,*

*MOVIE\_NUM INTEGER,*

*PRIMARY KEY(VID\_NUM),*

*FOREIGN KEY(MOVIE\_NUM)REFERENCES MOVIE(MOVIE\_NUM));*

*CREATE TABLE DETAILRENTAL(*

*RENT\_NUM INTEGER NOT NULL UNIQUE,*

*VID\_NUM INTEGER NOT NULL UNIQUE,*

*DETAIL\_FEE VARCHAR(10) NOT NULL,*

*DETAIL\_DUEDATE DATE NOT NULL,*

*DETAIL\_RETURNDATE DATE NOT NULL,*

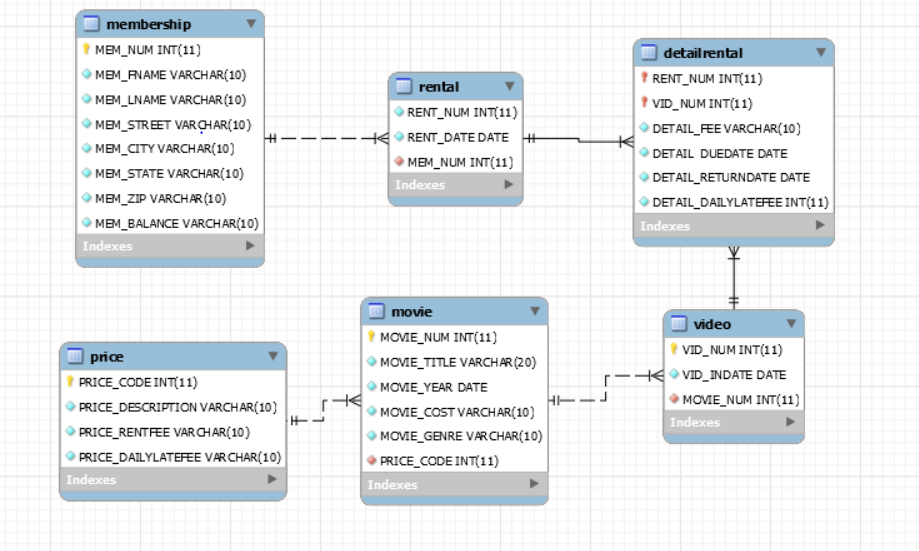
*DETAIL\_DAILYLATEFEE INTEGER NOT NULL,*

*PRIMARY KEY(RENT\_NUM, VID\_NUM),*

*FOREIGN KEY(RENT\_NUM)REFERENCES RENTAL(RENT\_NUM),*

*FOREIGN KEY(VID\_NUM)REFERENCES VIDEO(VID\_NUM));*

1. Generate the schema diagram for this database in MySQL and paste it in your assignment solution.



***Part 2: POPULATING THE DATABASE TABLES [10 marks]***

See the attached file “*DVDVidRental\_Data.xls*” that provides a sample of the data that will be kept in the database. This data needs to be inserted into the database for testing purposes. Write the INSERT commands necessary to place the data provided in the text file (*DVDVidRental\_Data.txt*) in the tables that were created in Task 1. Use these commands to insert this data into the MySQL database that you created in Task 1. Remember to save your newly inserted data into the database using COMMIT;

INSERT INTO membership VALUES (101,'Jesse','Carter','6050 University Avenue','Halifax','NS','12345','0'),

(102,'Tami','Dawson','2632 Takli Circle','Norene','TN','37136','11'),

(103,'Curt','Knight','4025 Cornell Court','Flatgap','KY','41219','6'),

(104,'Jamal','Melendez','788 East 145th Avenue','Quebeck','TN','38579','0'),

(105,'Iva','Mcclain','6045 Musket Ball Circle','Summit','KY','42783','15'),

(106,'Miranda','Parks','4469 Maxwell Place','Germantown','TN',38183,0),

(107,'Rosario','Elliott','7578 Danner Avenue','Columbia','TN',38402,5),

(108,'Mattie','Guy','4390 Evergreen Street','Li','KY',40740,0),

(109,'Clint','Ochoa','1711 Elm Street','Greeneville','TN',37745,10),

(110,'Lewis','Rosales','4524 Southwind Circle','Council','TN',38326,0),

(111,'Stacy','Mann','2789 East Cook Avenue','Murfreesboro','TN',37132,8),

(112,'Luis','Trujillo','7267 Melvin Avenue','Heiskell','TN',37754,3),

(113,'Minnie','Gonzales','6430 Vasili Drive','Williston','TN',38076,0);

INSERT INTO rental VALUES ('1001','01-Mar-16','103'),

('1002','01-Mar-16','105'),

('1003','02-Mar-16','102'),

('1004','02-Mar-16','110'),

('1005','02-Mar-16','111'),

('1006','02-Mar-16','107'),

('1007','02-Mar-16','104'),

('1008','03-Mar-16','105'),

('1009','03-Mar-16','111');

INSERT INTO price VALUES (1,'Standard',2,1),

(2,'New Release',3.5,3)

(3,'Discount',1.5,1)

(4,'Weekly Special',1,0.5);

INSERT INTO movie VALUES (1234,'The Cesar Family Christmas','2014','39.95','FAMILY','2'),

(1235,'Smokey Mountain Wildlife','2011','59.95','ACTION','1'),

(1236,'Richard Goodhope','2015','59.95','DRAMA',2),

(1237,'Beatnik Fever','2014','29.95','COMEDY',2)

(1238,'Constant Companion','2015','89.95','DRAMA'),

(1239,'Where Hope Dies','2005','25.49','DRAMA',3),

(1245,'Time to Burn','2012','45.49','ACTION',1),

(1246,'What He Doesn't Know','2013','58.29','COMEDY',1);

INSERT INTO video VALUES(54321,'18-Jun-15',1234),

(54324,'18-Jun-15',1234),

(54325,'18-Jun-15',1234),

(34341,'22-Jan-14',1235),

(34342,'22-Jan-14',1235),

(34366,'02-Mar-16',1236),

(34367,'02-Mar-16',1236),

(34368,'02-Mar-16',1236),

(34369,'02-Mar-16',1236),

(44392,'21-Oct-15',1237),

(44397,'21-Oct-15',1237),

(59237,'14-Feb-16',1237),

(61388,'25-Jan-14',1239),

(61353,'28-Jan-13',1245),

(61354,'28-Jan-13',1245),

(61367,'30-Jul-15',1246),

(61369,'30-Jul-15',1246);

INSERT INTO detailrental VALUES(1001,34342,2,'04-Mar-16','02-Mar-16',1),

(1001,61353,2,'04-Mar-16','03-Mar-16',1),

(1003,61369,2,'06-Mar-16','09-Mar-16',1),

(1003,61388,0,'06-Mar-16','09-Mar-16',1),

(1004,44392,3.5,'05-Mar-16','07-Mar-16',3),

(1004,34367,3.5,'05-Mar-16','07-Mar-16',3),

(1004,34341,2,'07-Mar-16','07-Mar-16',1),

(1005,34342,2,'07-Mar-16','05-Mar-16',1),

(1005,44397,3.5,'05-Mar-16','05-Mar-16',3),

(1006,34366,3.5,'05-Mar-16','04-Mar-16',3),

(1006,61367,2,'07-Mar-16',null,1),

(1007,34368,3.5,'05-Mar-16',null,3),

(1008,34369,3.5,'05-Mar-16','05-Mar-16',3),

(1009,54324,3.5,'05-Mar-16',null,3),

(1001,34366,3.5,'04-Mar-16','02-Mar-16',3);

\*The date values were altered manually in MySQL workbench as the ones provided in the excel sheet did not match the SQL Date format properly

***Part 3: [30 marks]***

Write the SQL commands for accomplishing the following tasks. For each task, ensure that your command is correct by executing it in your MySQL database.

***For each task, paste a screenshot of your results with your answers.***: ***(See the example shown below for task ‘0’)***

***Example Task 0: Write a query to display the movie title, movie year, and movie genre for all movies.***

***Answer)*** SELECT MOVIE\_TITLE, MOVIE\_YEAR, MOVIE\_GENRE

FROM MOVIE;



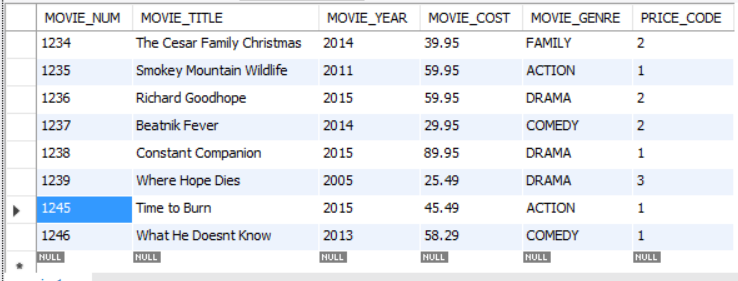
***Tasks:***

1. Write the SQL command to change the movie year for movie number 1245 to 2015.

*UPDATE movie*

*SET MOVIE\_YEAR='2015'*

*WHERE MOVIE\_NUM=1245;*

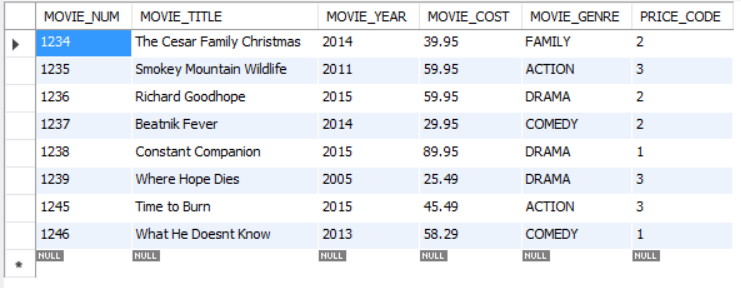


1. Write the SQL command to change the price code for all Action movies to price code 3.

UPDATE movie

SET PRICE\_CODE=3

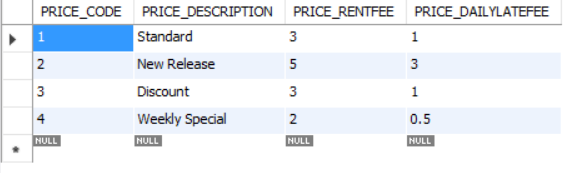
WHERE MOVIE\_GENRE='Action'



1. Write a single SQL command to increase all price rental fee values in the PRICE table by $1.00

UPDATE price

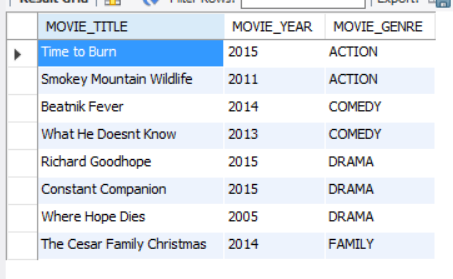
SET PRICE\_RENTFEE=PRICE\_RENTFEE+1



1. Write a query to display the movie title, movie year, and movie genre for all movies sorted by movie genre in ascending order, then sorted by movie year in descending order within genre.

SELECT MOVIE\_TITLE, MOVIE\_YEAR, MOVIE\_GENRE FROM movie

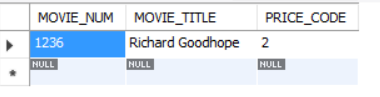
ORDER BY MOVIE\_GENRE, MOVIE\_YEAR DESC;



1. Write a query to display the movie number, movie title, and price code for all movies with a title that starts with the letter “R”.

SELECT MOVIE\_NUM, MOVIE\_TITLE, PRICE\_CODE FROM movie

WHERE MOVIE\_TITLE LIKE 'R%';

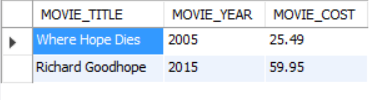


1. Write a query to display the movie title, movie year, and movie cost for all movies that contain the word “hope” anywhere in the title. Sort the results in descending order by title.

SELECT MOVIE\_TITLE, MOVIE\_YEAR, MOVIE\_COST FROM movie

WHERE MOVIE\_TITLE LIKE '%hope%'

ORDER BY MOVIE\_TITLE DESC;

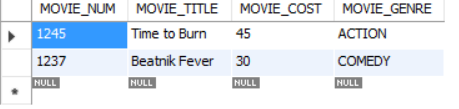


1. Write a query to display the movie number, movie title, movie cost, and movie genre for all movies that are either action or comedy movies and that have a cost that is less than $50. Sort the results in ascending order by genre.

SELECT MOVIE\_NUM, MOVIE\_TITLE, MOVIE\_COST, MOVIE\_GENRE FROM movie

WHERE (MOVIE\_GENRE ='ACTION' OR MOVIE\_GENRE='COMEDY') AND MOVIE\_COST<50.0

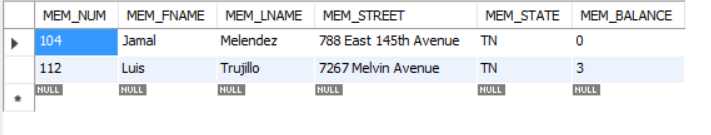
ORDER BY MOVIE\_GENRE;



1. Write a query to display the membership number, name, street, state, and balance for all members in Tennessee (TN), with a balance less than $5, and whose street name ends in “Avenue.”

SELECT MEM\_NUM, MEM\_FNAME, MEM\_LNAME, MEM\_STREET, MEM\_STATE, MEM\_BALANCE FROM membership

WHERE MEM\_STATE LIKE 'TN' AND MEM\_BALANCE<5 AND MEM\_STREET LIKE '%AVENUE';



1. Write a query to display the movie genre, the number of movies in each genre and the average cost of movies in each genre.
2. Write a query to display the movie title, movie genre, price description, and price rental fee for all movies with a price code.

SELECT MOVIE\_TITLE, MOVIE\_GENRE, PRICE\_DESCRIPTION, PRICE\_RENTFEE FROM movie, price

WHERE movie.PRICE\_CODE=price.PRICE\_CODE



1. Write a query to display the movie title and the movie cost divided by the price rental fee for each movie that has a price to determine the number of rentals it will take to break even on the purchase of the movie.

SELECT MOVIE\_TITLE, (MOVIE\_COST/PRICE\_RENTFEE) AS 'BREAK EVEN' FROM movie, price

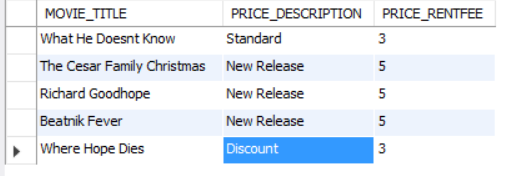
WHERE movie.PRICE\_CODE=price.PRICE\_CODE;



1. Write a query to display the movie title, price description, and price rental fee for all movies that are in the genres Family, Comedy, or Drama.

SELECT MOVIE\_TITLE, PRICE\_DESCRIPTION, PRICE\_RENTFEE FROM movie, price

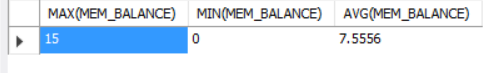
WHERE movie.PRICE\_CODE=price.PRICE\_CODE AND (MOVIE\_GENRE LIKE 'Family' OR MOVIE\_GENRE LIKE 'Comedy' OR MOVIE\_GENRE LIKE 'Drama');



1. Write a query to display the minimum balance, maximum balance, and average balance for memberships that have a rental.

SELECT MAX(MEM\_BALANCE), MIN(MEM\_BALANCE), AVG(MEM\_BALANCE) FROM membership, rental

WHERE membership.MEM\_NUM=rental.MEM\_NUM;



1. Write a query to display the rental number, rental date, video number, movie title, due date, and return date for all videos that were returned after the due date. Sort the results by rental number and movie title.
2. Write a query to display the movie number, movie genre, average movie cost of movies in that genre, movie cost of that individual movie, and the percentage difference between the average movie cost and the individual movie cost.   
   Note: the percentage difference is calculated as the cost of the individual movie minus the average cost of movies in that genre, divided by the average cost of movies in that genre multiplied by 100. For example, if the average cost of movies in the “Family” genre is $25, if a given Family movie cost $26, then the calculation would be ((26 – 25) / 25 \* 100), which would work out to be 4.00%. This indicates that this movie costs 4% more than the average Family movie.