2016

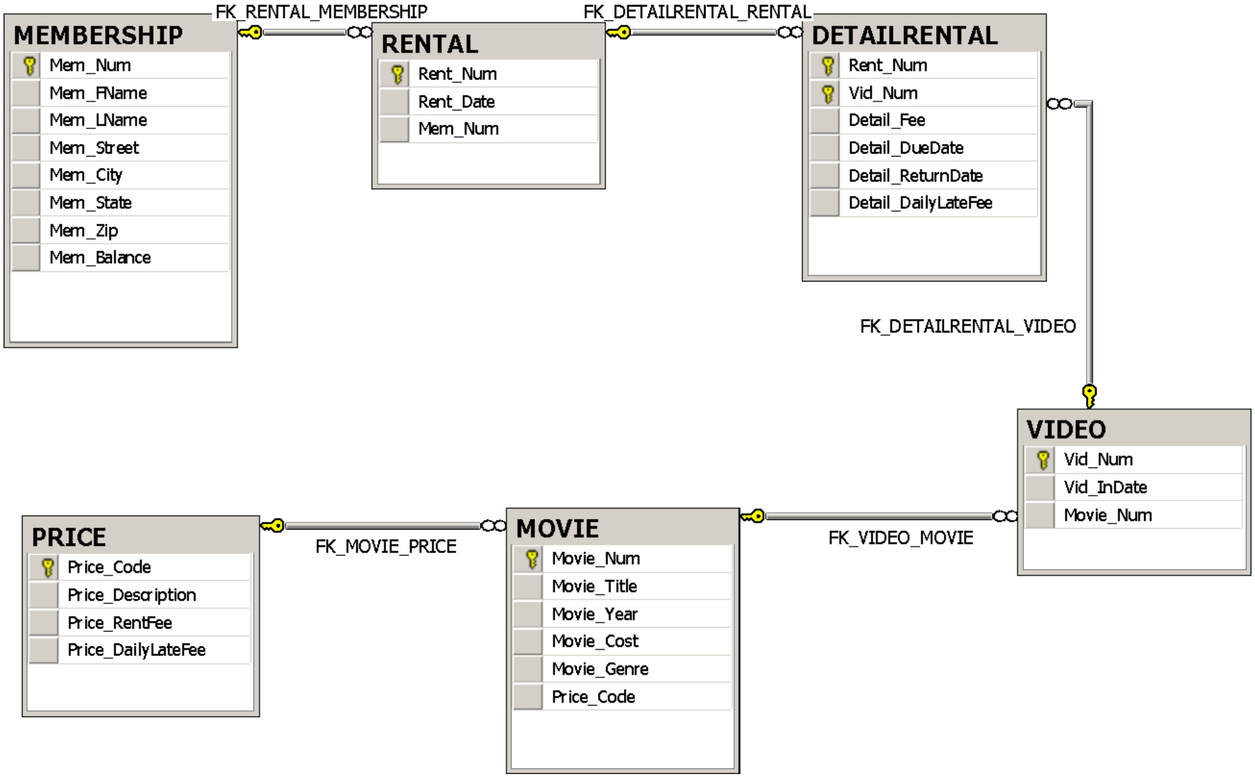
Assignment 8

CIS 310

JONES,CHRISTOPHER MORICHIKA

Christopher Jones CIS310 A8 3/7/16

Part 1:



Part 2:

72. Write a query to display movie title, movie year, and movie genre for all movies.

```
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_GENRE
FROM MOVIE
```

```
73.WRITE A QUERY TO DISPLAY THE MOVIE YEAR, MOVIE TITLE, AND MOVIE COST SORTED BY MOVIE
YEAR IN DESCENDING ORDER


SELECT MOVIE_YEAR, MOVIE_TITLE, MOVIE_COST
FROM MOVIE
ORDER BY MOVIE_YEAR DESC


--74. 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 ASC, MOVIE_YEAR DESC
```

```sql
--75. 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.MOVIE_NUM, MOVIE.MOVIE_TITLE, PRICE.PRICE_CODE
FROM MOVIE INNER JOIN PRICE ON MOVIE.PRICE_CODE = PRICE.PRICE_CODE
WHERE MOVIE_TITLE LIKE 'R%'


--76. Write a query to display the movie title, movie year, and movie cost for all movies
that contain the word hope in the title.
-- Sort the results in ascending order by title.

SELECT MOVIE.MOVIE_TITLE, MOVIE.MOVIE_YEAR, PRICE_CODE
from MOVIE
WHERE MOVIE_TITLE LIKE '%hope%'
ORDER BY MOVIE_TITLE ASC

--77. Write a query to display the movie title, movie year, and movie genre for all
action movies



SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_GENRE
FROM MOVIE
WHERE MOVIE_GENRE = 'Action'

---78. Write a query to display the movie number, movie title, and movie cost for all
movies that cost more than $40.
SELECT MOVIE_NUM, MOVIE_TITLE, MOVIE_COST
FROM MOVIE
WHERE MOVIE_COST > 40

--79. Write a query to display the movie number, movie title, movie cost, and movie genre
for all action or comedy movies that cost less than $50.

Select MOVIE_NUM, MOVIE_TITLE, MOVIE_COST, MOVIE_GENRE
FROM MOVIE
WHERE MOVIE_GENRE = 'Comedy' AND MOVIE_COST < 50 OR MOVIE_GENRE = 'Action' and MOVIE_COST
< 50
ORDER BY MOVIE_GENRE ASC

--80. Write a query to display the movie number and movie description for all movies,
--where the movie description is a combination of the movie title, movie year, and movie
genre, with the movie year enclosed in parentheses


SELECT MOVIE_NUM, CONCAT(MOVIE_TITLE,' (' , MOVIE_YEAR , ') ' ,MOVIE_GENRE) AS "Movie
Description"
FROM MOVIE


--81. Write a query to display the movie genre and the number of movies in each genre.

SELECT MOVIE_GENRE, COUNT(*) AS "Number of Movies"
 FROM MOVIE
 GROUP BY MOVIE_GENRE

--82. Write a query to display the average cost of all the movies
```

```sql
SELECT AVG(MOVIE_COST) AS "Average movie cost"
FROM MOVIE


--83. Write a query to display the movie the movie genre and average cost of movies in
each genre
SELECT DISTINCT Movie_Genre, AVG(MOVIE_COST) AS "Average Cost"
FROM MOVIE
GROUP BY MOVIE_GENRE

--84. 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 inner join PRICE ON MOVIE.PRICE_CODE = PRICE.PRICE_CODE


--85. Write a query to display the movie genre and average rental fee for movies in each
genre that have a price

SELECT MOVIE_GENRE, AVG(PRICE_RENTFEE) AS "AVERAGE RENTAL FEE"
FROM MOVIE INNER JOIN PRICE ON MOVIE.PRICE_CODE = PRICE.PRICE_CODE
GROUP BY MOVIE_GENRE

--86. Write a query to display the movie title, movie year, and breakeven amount for each
movie that has a price.
--The breakeven amount is the movie cost divided by the price rental fee for each movie
that has a price; it determines
--the number of rentals needed to break even on the purchase of the movie.


SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_COST/PRICE_RENTFEE AS "BREAKEVEN AMOUNT"
FROM MOVIE X INNER JOIN PRICE Y ON X.PRICE_CODE = Y.PRICE_CODE


--87. Write a query to display the movie title and movie year for all movies that have a
price code

SELECT MOVIE_TITLE, MOVIE_YEAR
FROM MOVIE

--88. wRITE A QUERY TO DISPLAY THE MOVIE TITLE, MOVIE YEAR, AND MOVIE COST FOR ALL MOVIES
THAT COST BETWEEN $44.99 AND $49.99

SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_COST
FROM MOVIE
WHERE MOVIE_COST BETWEEN '44.99' AND '49.99'

--89. Write a query to display the movie title, movie year, price description, and price
rental fee for all movies that are in the genres of family, comedy, or drama.

SELECT MOVIE_TITLE, MOVIE_YEAR, PRICE_DESCRIPTION, PRICE_RENTFEE, MOVIE_GENRE
FROM MOVIE M INNER JOIN PRICE P ON M.PRICE_CODE = P.PRICE_CODE
where MOVIE_GENRE = 'Family' OR MOVIE_GENRE = 'Comedy' OR MOVIE_GENRE = 'DRAMA'

--90. Write a query to display the movie number, movie title, and movie year for all
movies that do not have a video
```

```sql
SELECT M.MOVIE_NUM, MOVIE_TITLE, MOVIE_YEAR
FROM MOVIE M FULL JOIN VIDEO V ON M.MOVIE_NUM = V.MOVIE_NUM
WHERE V.VID_NUM IS NULL

--91. Write a query to display the membership number, first name, last name, and balance
of the memberships that have a rental

SELECT M.MEM_NUM, MEM_FNAME, MEM_LNAME, MEM_BALANCE
FROM MEMBERSHIP M FULL JOIN RENTAL R ON M.MEM_NUM = R.MEM_NUM
WHERE R.RENT_NUM IS NOT NULL

--92. Write a query to display the minimum balance, maximum balance, and average balance
for memberships that have a rental

SELECT MIN(MEM_BALANCE) AS 'Minimum Balance', Max(Mem_Balance) AS 'Maximum Balance',
AVG(Mem_Balance) AS 'Average Balance'
from Membership M FULL JOIN RENTAL R ON M.MEM_NUM = R.MEM_NUM
WHERE R.RENT_NUM IS NOT NULL

--93. Write a query to display the membership name and membership address. The membership
name is a concatenation of the
--first name and last name with a space added between them in a single column. The
membership address is a concatenation of the
--street, city, state, and zip code into a single column with spaces

SELECT CONCAT(MEM_FNAME,' ',MEM_LNAME) AS 'Membership Name', CONCAT(Mem_Street,'
',Mem_City,' ',Mem_State,' ',Mem_Zip) AS 'Membership Address'
FROM MEMBERSHIP

--94. 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 # and movie title

SELECT R.RENT_NUM, R.RENT_DATE, D.VID_NUM, MOVIE_TITLE, DETAIL_DUEDATE, DETAIL_RETURNDATE
FROM RENTAL R INNER JOIN DETAILRENTAL D ON R.RENT_NUM = D.RENT_NUM
INNER JOIN VIDEO V ON D.VID_NUM = V.VID_NUM
INNER JOIN MOVIE M ON V.MOVIE_NUM = M.MOVIE_NUM
WHERE D.Detail_ReturnDate > D.DETAIL_DUEDATE
--????Why doesn't my sort work :GROUP BY R.RENT_NUM, MOVIE_TITLE

--95. Write a query to display the rental number, rental date, video number, movie title,
due date, return date, detail fee,
--and number of days past due for each video that was returned after the due date. Sort
the results by rental number and movie title.

SELECT R.RENT_NUM, R.RENT_DATE, D.VID_NUM, MOVIE_TITLE, DETAIL_DUEDATE,
DETAIL_RETURNDATE, DETAIL_FEE, DIFFERENCE(D.DETAIL_RETURNDATE,D.DETAIL_DUEDATE) AS
'Number of days past due'
FROM RENTAL R INNER JOIN DETAILRENTAL D ON R.RENT_NUM = D.RENT_NUM
INNER JOIN VIDEO V ON D.VID_NUM = V.VID_NUM
INNER JOIN MOVIE M ON V.MOVIE_NUM = M.MOVIE_NUM
WHERE D.Detail_ReturnDate > D.DETAIL_DUEDATE
--??Why doesn't my sort work: Group By R.RENT_NUM, M.MOVIE_TITLE

--96. Write a query to display the rental number, rental date, movie title, and detail
fee for each movie that was returned on or
--before the due date
```

```sql
SELECT D.RENT_NUM, R.RENT_DATE, MOVIE_TITLE, DETAIL_FEE
FROM RENTAL R INNER JOIN DETAILRENTAL D ON R.RENT_NUM = D.RENT_NUM
INNER JOIN VIDEO V ON D.VID_NUM = V.VID_NUM
INNER JOIN MOVIE M ON V.MOVIE_NUM = M.MOVIE_NUM
WHERE D.Detail_ReturnDate <= D.DETAIL_DUEDATE
--Why Doesn't my sort work?: GROUP BY D.RENT_NUM

--97. Write a query to display the membership number, last name, first name, and total
rental fees earned from that membership

SELECT M.MEM_NUM, M.MEM_LNAME, M.MEM_FNAME, SUM(D.DETAIL_FEE) AS 'TOTAL RENTAL FEES'
FROM MEMBERSHIP M INNER JOIN RENTAL R ON M.MEM_NUM = R.MEM_NUM
INNER JOIN DETAILRENTAL D ON R.RENT_NUM = D.RENT_NUM
WHERE D.Detail_ReturnDate > D.DETAIL_DUEDATE
GROUP BY M.MEM_NUM, M.MEM_LNAME, M.MEM_FNAME

--98. Write a query to display the movie number, movie genre, average cost of movies in
that genre, cost of the individual movie,
--and the percentage difference between the average movie cost and the individual movie
cost.

SELECT MOVIE_NUM, MOVIE_GENRE, AVG(MOVIE_COST) AS 'AVERAGE MOVIE COST', MOVIE_COST,
DIFFERENCE(MOVIE_COST,AVG(MOVIE_COST))/AVG(MOVIE_COST)*100 AS 'PERECENT DIFFERENCE'
FROM MOVIE
GROUP BY MOVIE_NUM, MOVIE_GENRE, MOVIE_COST
```