# DBS301 Assignment 1

Group Members:

Abdul Rehman Seneca ID: 149019176

Imran Qureshi Seneca ID:023966005

1. **Write the SQL command to change the movie year for movie number 1245 to 2008.**

update movie

set movie\_year = 2008

where movie\_num = 1245;

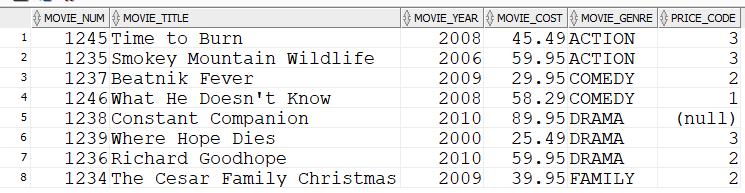


1. **Write the SQL command to change the price code for all Action movies to price code 3.**

**update movie**

set price\_code = 3

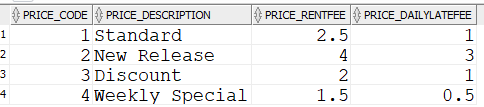
where movie\_genre ='ACTION';



1. **Write a single SQL command to increase all price rental fee values by $0.50.**

update price

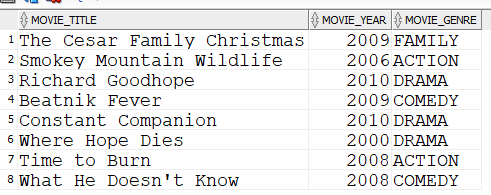
set price\_rentfee = price\_rentfee + 0.5;



1. **Write a query to display the movie title, movie year, and movie genre for all movies.**

select movie\_title,movie\_year,movie\_genre

from movie;

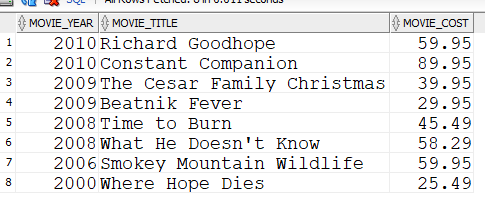


1. **Write a query to display the movie year, movie title, and movie cost sorted by movie year in descending order.**

select movie\_year,movie\_title,movie\_cost

from movie

order by 1 desc;



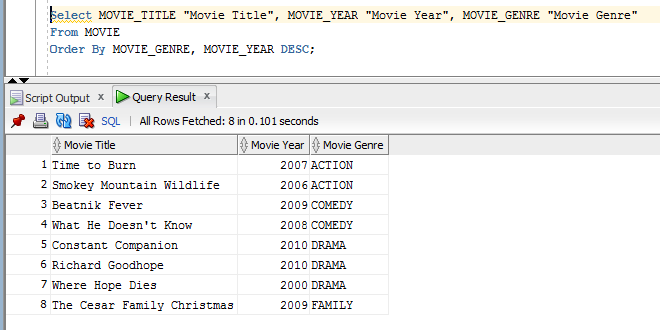
**6. Write a query to display the movie title, movie year, and movie genre for all movies sorted by movie genre in ascending order, then sorted by movie year in descending order within genre**

Ans:6

Select MOVIE\_TITLE "Movie Title", MOVIE\_YEAR "Movie Year", MOVIE\_GENRE "Movie Genre"

From MOVIE

Order By MOVIE\_GENRE, MOVIE\_YEAR DESC;

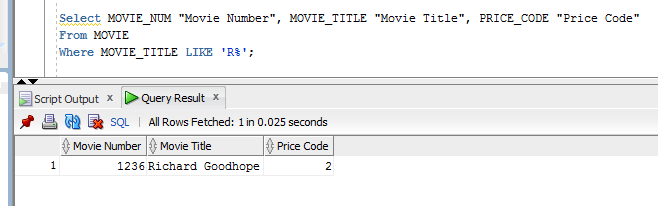


**7. Write a query to display the movie number, movie title, and price code for all movies with a title that starts with the letter “R” (result shown in Figure P7.75).**

Select MOVIE\_NUM "Movie Number", MOVIE\_TITLE "Movie Title", PRICE\_CODE "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 ascending order by title (result shown in figure P7.76).** 

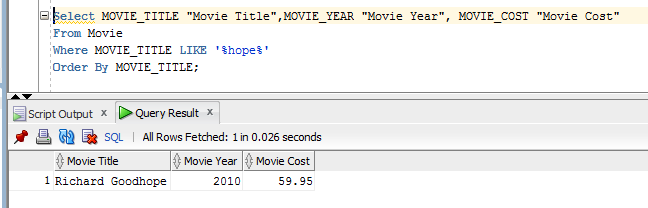
Select MOVIE\_TITLE "Movie Title”, MOVIE\_YEAR "Movie Year", MOVIE\_COST "Movie Cost"

From Movie

Where MOVIE\_TITLE LIKE '%hope%'

Order By MOVIE\_TITLE;

Select

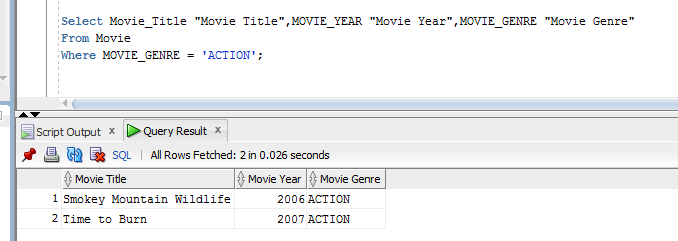


**9.Write a query to display the movie title, movie year, and movie genre for all action movies (result shown in Figure P7.77).**

Select Movie\_Title "Movie Title",MOVIE\_YEAR "Movie Year",MOVIE\_GENRE "Movie Genre"

From Movie

Where MOVIE\_GENRE = 'ACTION';

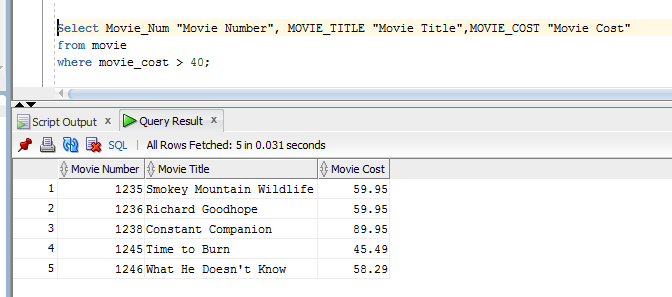


**10.Write a query to display the movie number, movie title, and movie cost for all movies with a cost greater than $40 (result shown in Figure P7.78).**

Select Movie\_Num "Movie Number", MOVIE\_TITLE "Movie Title",MOVIE\_COST "Movie Cost"

from movie

where movie\_cost > 40;



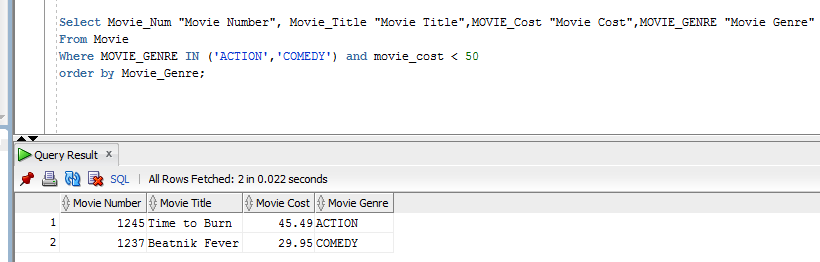
**11.Write a query to display the movie number, movie title, movie cost, and movie genre for all movies that are either action or comedy movies and that have a cost that is less than $50. Sort the results in ascending order by genre.**

Select Movie\_Num "Movie Number", Movie\_Title "Movie Title",MOVIE\_Cost "Movie Cost",MOVIE\_GENRE "Movie Genre"

From Movie

Where MOVIE\_GENRE IN ('ACTION','COMEDY') and movie\_cost < 50

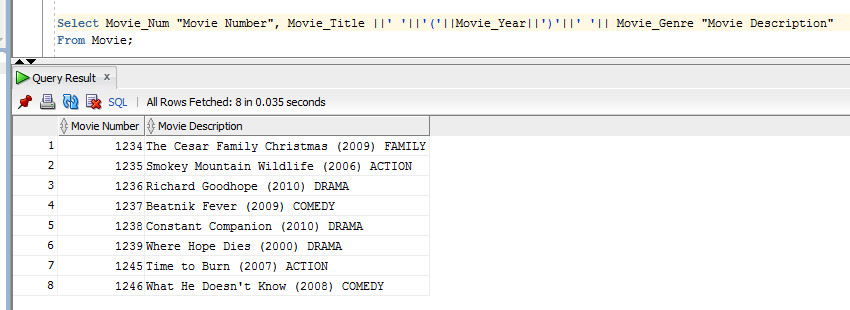
order by Movie\_Genre;



**12.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 (result shown in Figure P7.80).**

Select Movie\_Num "Movie Number", Movie\_Title ||' '||'('||Movie\_Year||')'||' '|| Movie\_Genre "Movie Description"

From Movie;



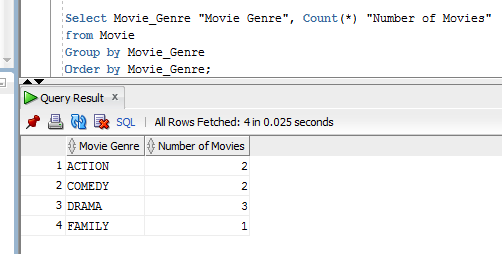
**13.Write a query to display the movie genre and the number of movies in each genre (result shown in Figure P7.81).**

Select Movie\_Genre "Movie Genre", Count(\*) "Number of Movies"

from Movie

Group by Movie\_Genre

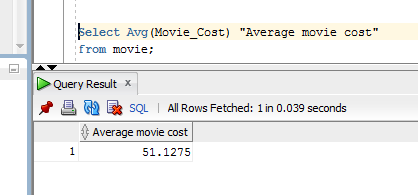
Order by Movie\_Genre;



**14.Write a query to display the average cost of all of the movies (result shown in Figure P7.82).**

Select Avg(Movie\_Cost) "Average movie cost"

from movie;

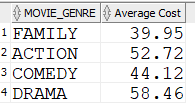


**15.Write a query to display the movie genre and average cost of movies in each genre (result shown in Figure P7.83).**

select movie\_genre, trunc(avg(movie\_cost),2) "Average Cost"

from movie

group by movie\_genre;

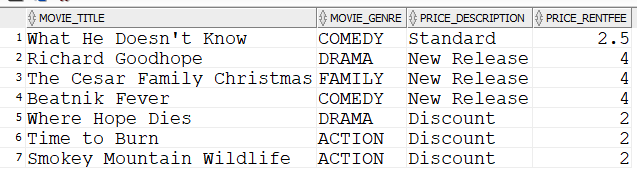


**16.Write a query to display the movie title, movie genre, price description, and price rental fee for all movies with a price code**

select m.movie\_title, m.movie\_genre, p.price\_description, p.price\_rentfee

from movie m, price p

where m.price\_code = p.price\_code;



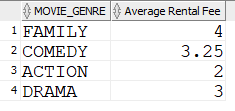
**17.Write a query to display the movie genre and average price rental fee for movies in each genre that have a price**

select movie\_genre, avg(price\_rentfee) "Average Rental Fee"

from movie m, price p

where m.price\_code = p.price\_code

group by movie\_genre;

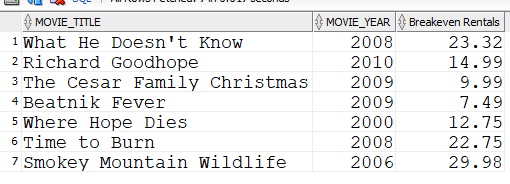


**18. Write a query to display the movie title, movie year, and the movie cost divided by the price rental fee for each movie that has a price to determine the number of rentals it will take to break even on the purchase of the movie (result shown in Figure P7.86).**

select movie\_title,movie\_year, round(movie\_cost/price\_rentfee,2) "Breakeven Rentals"

from movie m, price p

where m.price\_code = p.price\_code;

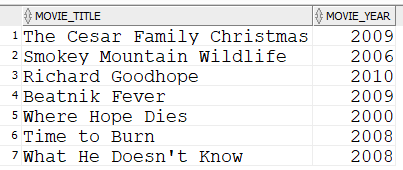


**19. Write a query to display the movie title and movie year for all movies that have a price code.**

select movie\_title, movie\_year

from movie

where price\_code is NOT NULL;



**20. Write a query to display the movie title, movie year, and movie cost for all movies that have a 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;



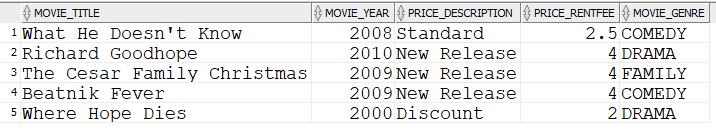
**21. Write a query to display the movie title, movie year, price description, and price rental fee for all movies that are in the genres Family, Comedy, or Drama (result shown in Figure P7.89).**

select movie\_title, movie\_year, price\_description, p.price\_rentfee, movie\_genre

from movie JOIN price p

using(price\_code)

where movie\_genre IN ('FAMILY','COMEDY','DRAMA');



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

select movie\_num, m.movie\_title, m.movie\_year

from movie m LEFT JOIN video v

using (movie\_num)

where vid\_num IS NULL;



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

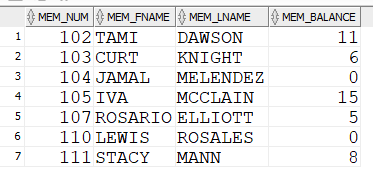
select distinct mem\_num, m.mem\_fname, m.mem\_lname, m.mem\_balance

from membership m left JOIN rental r

using(mem\_num)

where r.rent\_num is NOT NULL

order by 1;



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

select min(mem\_balance) "Minimum Balance" , max(mem\_balance) "Maximum Balance", round(avg(mem\_balance),2) "Average Balance"

from membership m join rental r

using(mem\_num)

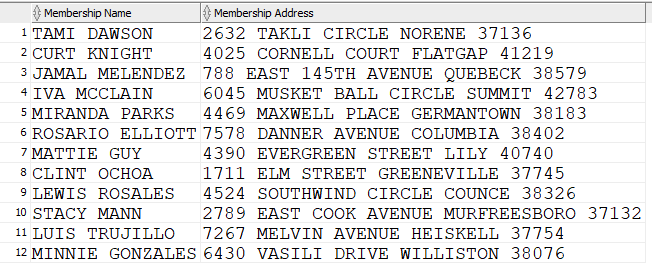
where rent\_num is NOT NULL;



**25. Write a query to display the membership name (concatenate the first name and last name with a space between them into a single column), membership address (concatenate the street, city, state, and zip codes into a single column with spaces (result shown in Figure P7.93).**

select trim(mem\_fname) ||' ' || trim(mem\_lname) "Membership Name", trim(mem\_street) || ' ' || trim(mem\_city) || ' ' || trim(mem\_zip) "Membership Address"

from membership;



**26.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 (result shown in Figure P7.94).**

SELECT R.RENT\_NUM, R.RENT\_DATE, D.VID\_NUM, M.MOVIE\_TITLE, D.DETAIL\_DUEDATE, D.DETAIL\_RETURNDATE

From rental r,detailrental d,video v,movie m

where (r.rent\_num = d.rent\_num) and (d.vid\_num = v.vid\_num) and (v.movie\_num = m.movie\_num)

and D.Detail\_ReturnDate > D.DETAIL\_DUEDATE

order by r.rent\_num,m.movie\_title;

OR

SELECT R.RENT\_NUM, R.RENT\_DATE, D.VID\_NUM, M.MOVIE\_TITLE, D.DETAIL\_DUEDATE, D.DETAIL\_RETURNDATE

FROM RENTAL R JOIN DETAILRENTAL D

ON R.RENT\_NUM = D.RENT\_NUM

JOIN VIDEO V

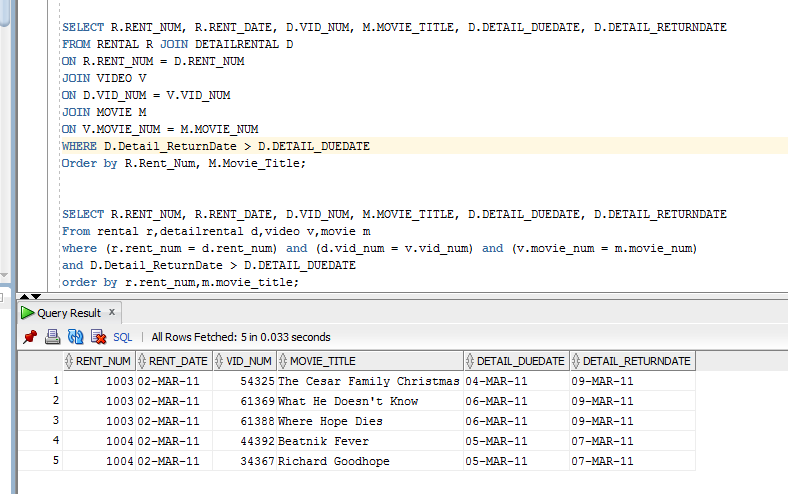
ON D.VID\_NUM = V.VID\_NUM

JOIN MOVIE M

ON V.MOVIE\_NUM = M.MOVIE\_NUM

WHERE D.Detail\_ReturnDate > D.DETAIL\_DUEDATE

Order by R.Rent\_Num, M.Movie\_Title;



**27.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 the due date that the video was returned for each video that was returned after the due date. Sort the results by rental number and movie title. (Result shown in Figure P7.95.)**

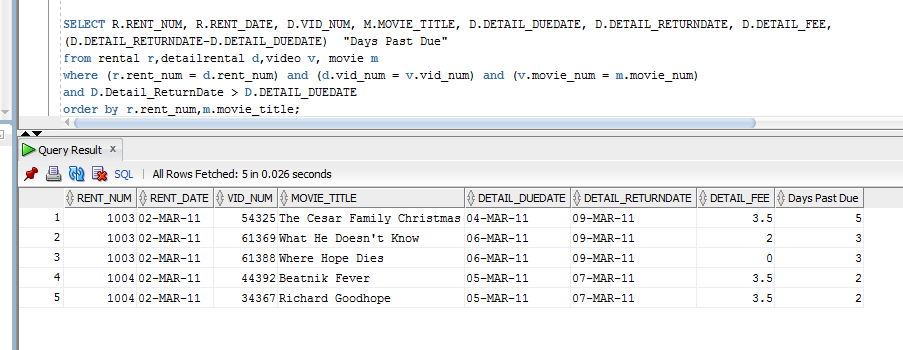
SELECT R.RENT\_NUM, R.RENT\_DATE, D.VID\_NUM, M.MOVIE\_TITLE, D.DETAIL\_DUEDATE, D.DETAIL\_RETURNDATE, D.DETAIL\_FEE, (D.DETAIL\_RETURNDATE-D.DETAIL\_DUEDATE) "Days Past Due"

from rental r,detailrental d,video v, movie m

where (r.rent\_num = d.rent\_num) and (d.vid\_num = v.vid\_num) and (v.movie\_num = m.movie\_num)

and D.Detail\_ReturnDate > D.DETAIL\_DUEDATE

order by r.rent\_num,m.movie\_title;



**28.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 (result shown in Figure P7.96).**

Select r.RENT\_NUM,r.RENT\_DATE,m.MOVIE\_TITLE,d.DETAIL\_FEE

FROM RENTAL R JOIN DETAILRENTAL D

ON R.RENT\_NUM = D.RENT\_NUM

JOIN VIDEO V

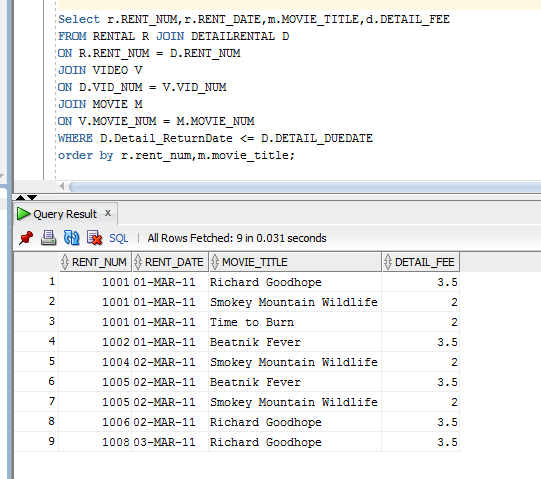
ON D.VID\_NUM = V.VID\_NUM

JOIN MOVIE M

ON V.MOVIE\_NUM = M.MOVIE\_NUM

WHERE D.Detail\_ReturnDate <= D.DETAIL\_DUEDATE

order by r.rent\_num,m.movie\_title;



**29.Write a query to display the membership number, last name, and total rental fees earned from that membership (result shown in Figure P7.97). The total rental fee is the sum of all of the detail fees (without the late fees) from all movies that the membership has rented.**

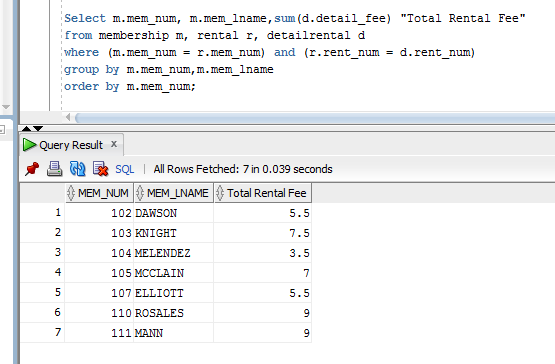
Select m.mem\_num, m.mem\_lname,sum(d.detail\_fee) "Total Rental Fee"

from membership m, rental r, detailrental d

where (m.mem\_num = r.mem\_num) and (r.rent\_num = d.rent\_num)

group by m.mem\_num,m.mem\_lname

order by m.mem\_num;



30.**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 (result shown in Figure P7.98).**

select m.movie\_num, m.movie\_genre, avg(m.movie\_cost), ((g.movie\_cost - avg(g.move\_cost)/g.movie\_cost))\*100 "diff"

from movie m self join movie g

using (movie\_num)

group by m.MOVIE\_NUM, m.MOVIE\_GENRE, m.MOVIE\_COST;