DBS301 Assignment 1

Group Members:

Abdul Rehman Seneca ID: 149019176

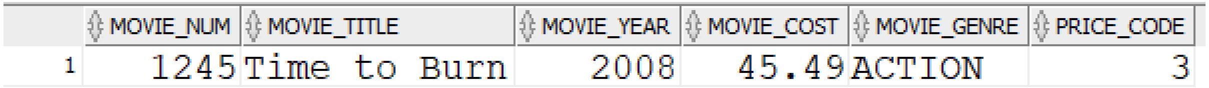
Imran Qureshi Seneca ID:023966005

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;

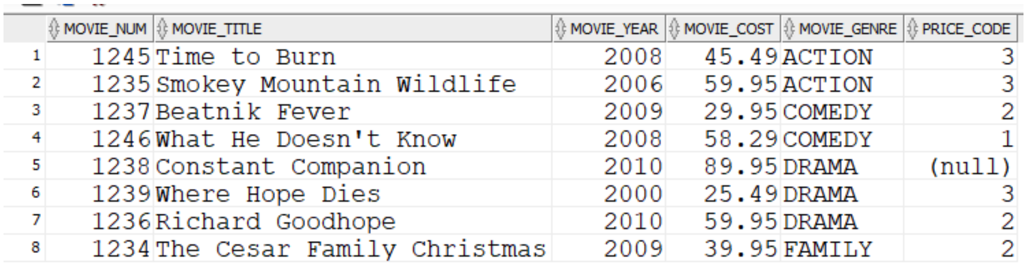


	MOVIE_NUM	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST	MOVIE_GENRE	PRICE_CODE
1	1245	Time to Burn	2008	45.49	ACTION	3

2. 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';

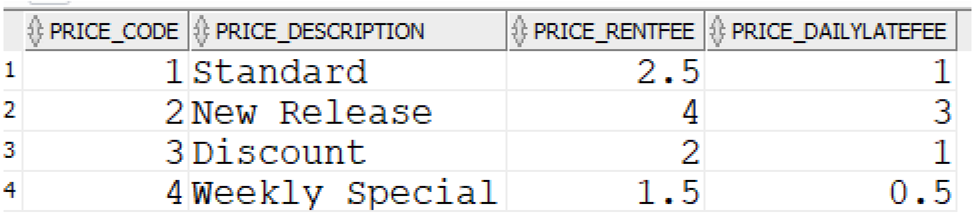


	MOVIE_NUM	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST	MOVIE_GENRE	PRICE_CODE
1	1245	Time to Burn	2008	45.49	ACTION	3
2	1235	Smokey Mountain Wildlife	2006	59.95	ACTION	3
3	1237	Beatnik Fever	2009	29.95	COMEDY	2
4	1246	What He Doesn't Know	2008	58.29	COMEDY	1
5	1238	Constant Companion	2010	89.95	DRAMA	(null)
6	1239	Where Hope Dies	2000	25.49	DRAMA	3
7	1236	Richard Goodhope	2010	59.95	DRAMA	2
8	1234	The Cesar Family Christmas	2009	39.95	FAMILY	2

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

update price

set price_rentfee = price_rentfee + 0.5;

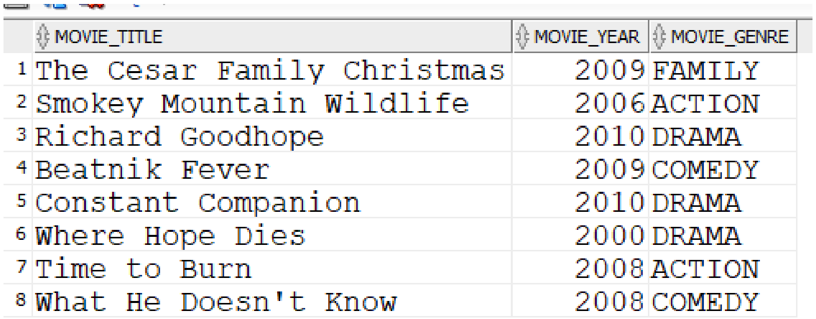


	PRICE_CODE	PRICE_DESCRIPTION	PRICE_RENTFEE	PRICE_DAILYLATEFEE
1	1	Standard	2.5	1
2	2	New Release	4	3
3	3	Discount	2	1
4	4	Weekly Special	1.5	0.5

4. 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;



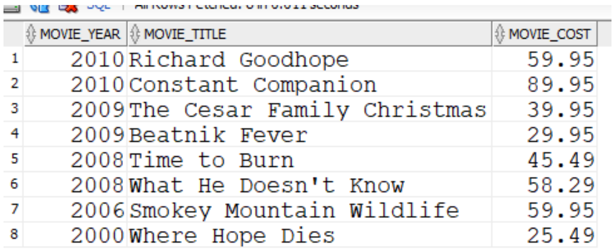
	MOVIE_TITLE	MOVIE_YEAR	MOVIE_GENRE
1	The Cesar Family Christmas	2009	FAMILY
2	Smokey Mountain Wildlife	2006	ACTION
3	Richard Goodhope	2010	DRAMA
4	Beatnik Fever	2009	COMEDY
5	Constant Companion	2010	DRAMA
6	Where Hope Dies	2000	DRAMA
7	Time to Burn	2008	ACTION
8	What He Doesn't Know	2008	COMEDY

5. **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;



	MOVIE_YEAR	MOVIE_TITLE	MOVIE_COST
1	2010	Richard Goodhope	59.95
2	2010	Constant Companion	89.95
3	2009	The Cesar Family Christmas	39.95
4	2009	Beatnik Fever	29.95
5	2008	Time to Burn	45.49
6	2008	What He Doesn't Know	58.29
7	2006	Smokey Mountain Wildlife	59.95
8	2000	Where Hope Dies	25.49

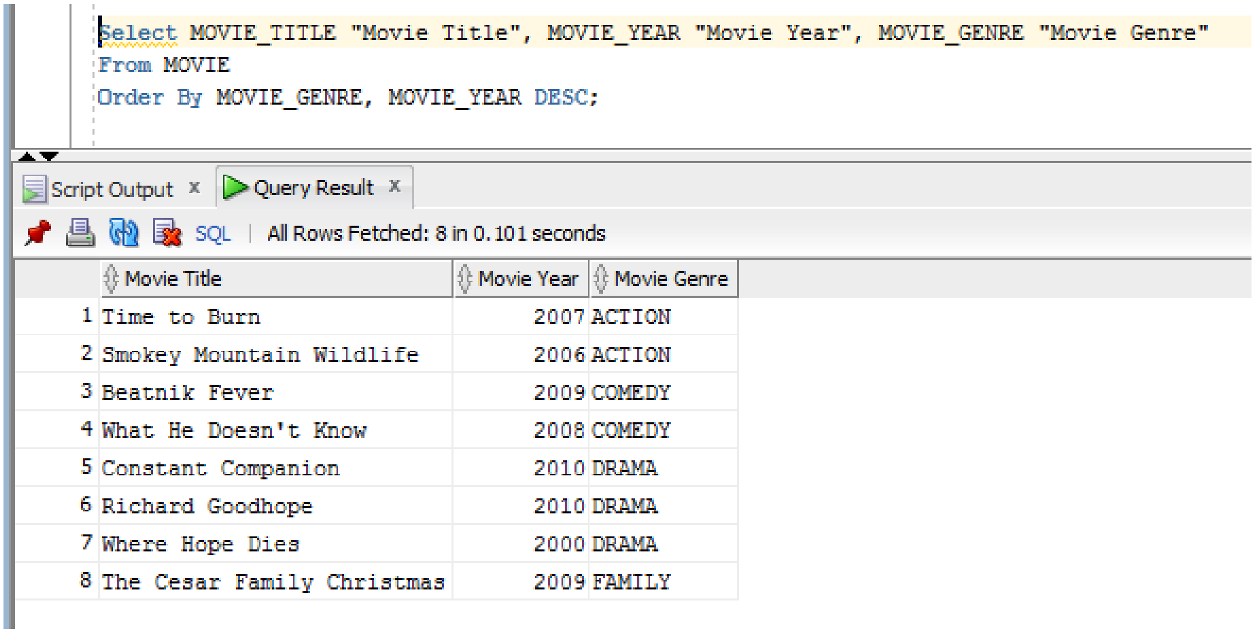
6. Write a query to display the movie title, movie year, and movie genre for all movies sorted by movie genre in ascending order, then sorted by movie year in descending order within genre

Ans:6

Select MOVIE_TITLE "Movie Title", MOVIE_YEAR "Movie Year", MOVIE_GENRE "Movie Genre"

From MOVIE

Order By MOVIE_GENRE, MOVIE_YEAR DESC;



```
Select MOVIE_TITLE "Movie Title", MOVIE_YEAR "Movie Year", MOVIE_GENRE "Movie Genre"
From MOVIE
Order By MOVIE_GENRE, MOVIE_YEAR DESC;
```

All Rows Fetched: 8 in 0.101 seconds

	Movie Title	Movie Year	Movie Genre
1	Time to Burn	2007	ACTION
2	Smokey Mountain Wildlife	2006	ACTION
3	Beatnik Fever	2009	COMEDY
4	What He Doesn't Know	2008	COMEDY
5	Constant Companion	2010	DRAMA
6	Richard Goodhope	2010	DRAMA
7	Where Hope Dies	2000	DRAMA
8	The Cesar Family Christmas	2009	FAMILY

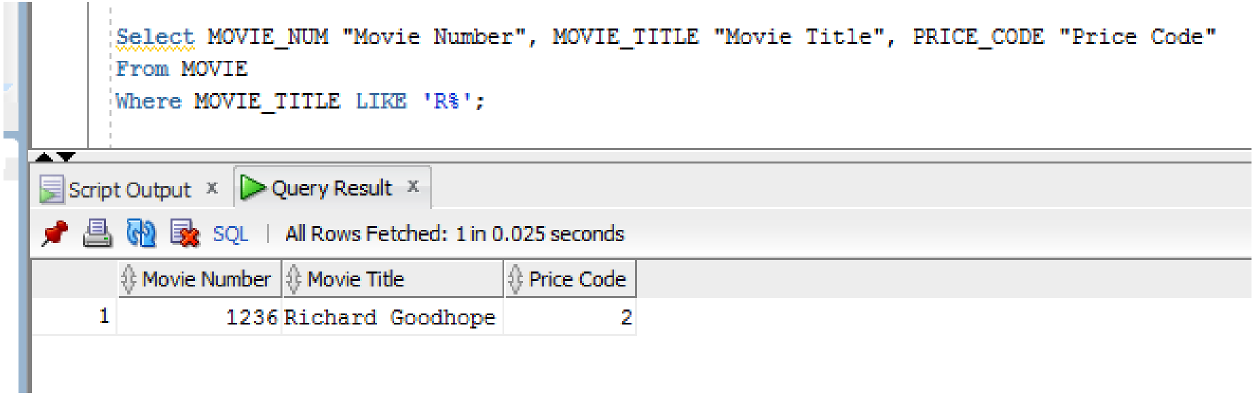
7. Write a query to display the movie number, movie title, and price code for all movies with a title that starts with the letter “R” (result shown in Figure P7.75).

Select MOVIE_NUM "Movie Number", MOVIE_TITLE "Movie Title", PRICE_CODE "Price Code"

From MOVIE

Where MOVIE_TITLE LIKE 'R%';

8. **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_NUM "Movie Number", MOVIE_TITLE "Movie Title", PRICE_CODE "Price Code"
From MOVIE
Where MOVIE_TITLE LIKE 'R%';
```

All Rows Fetched: 1 in 0.025 seconds

	Movie Number	Movie Title	Price Code
1	1236	Richard Goodhope	2

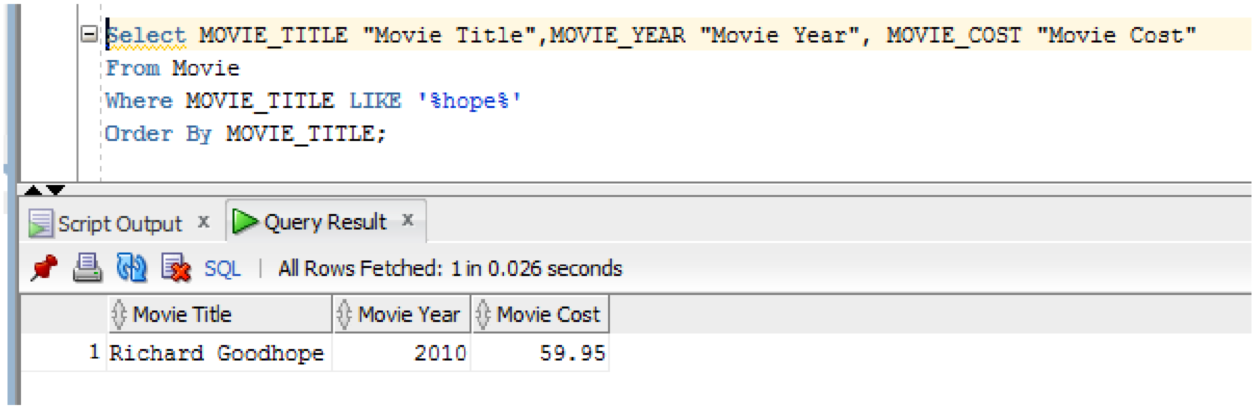
Select MOVIE_TITLE "Movie Title”, MOVIE_YEAR "Movie Year", MOVIE_COST "Movie Cost"

From Movie

Where MOVIE_TITLE LIKE '%hope%'

Order By MOVIE_TITLE;

Select



```
Select MOVIE_TITLE "Movie Title",MOVIE_YEAR "Movie Year", MOVIE_COST "Movie Cost"
From Movie
Where MOVIE_TITLE LIKE '%hope%'
Order By MOVIE_TITLE;
```

All Rows Fetched: 1 in 0.026 seconds

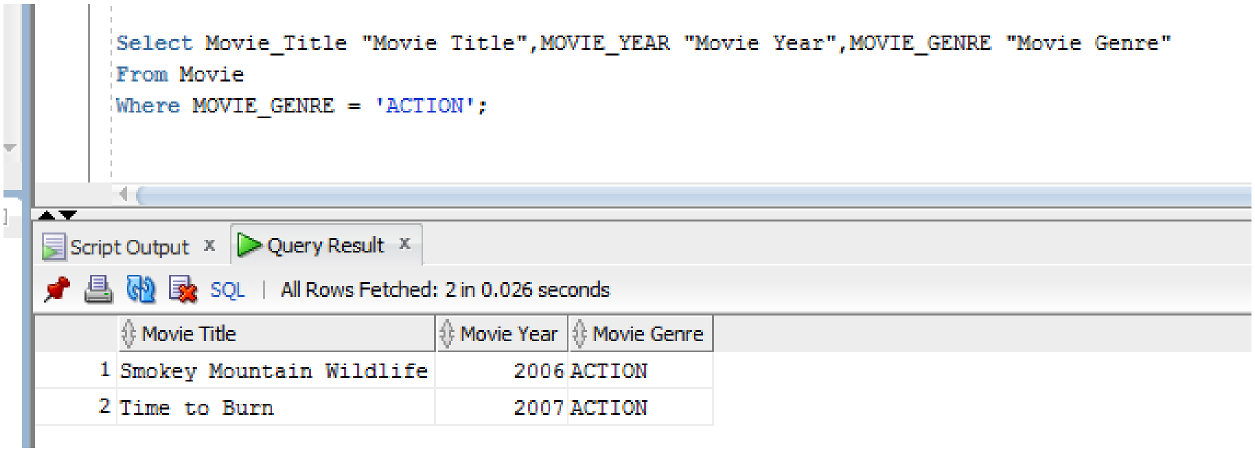
	Movie Title	Movie Year	Movie Cost
1	Richard Goodhope	2010	59.95

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';



```
Select Movie_Title "Movie Title",MOVIE_YEAR "Movie Year",MOVIE_GENRE "Movie Genre"
From Movie
Where MOVIE_GENRE = 'ACTION';
```

All Rows Fetched: 2 in 0.026 seconds

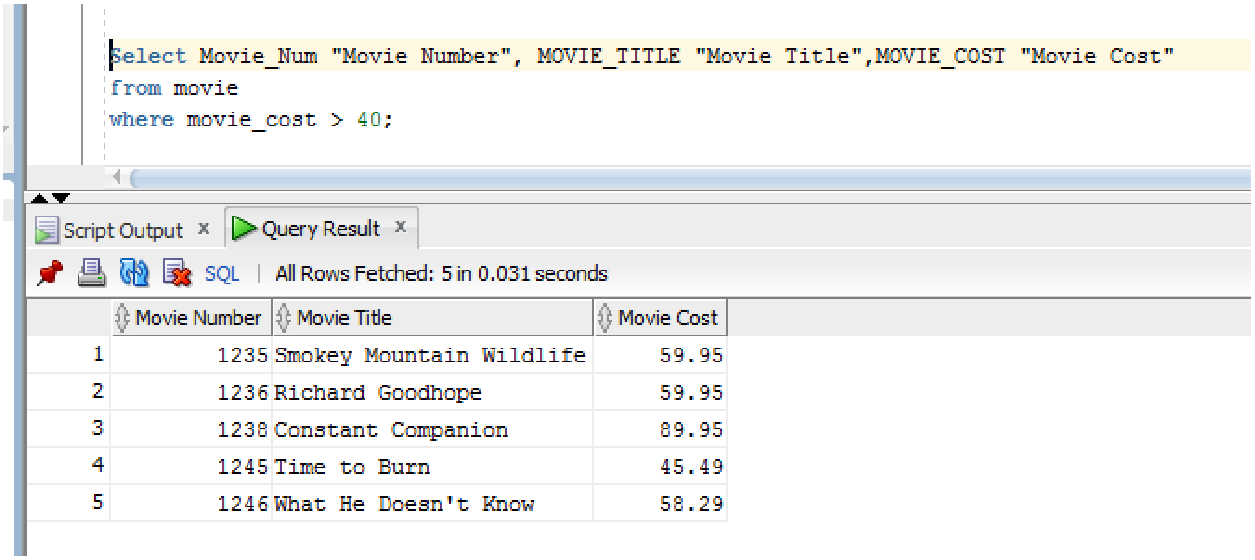
	Movie Title	Movie Year	Movie Genre
1	Smokey Mountain Wildlife	2006	ACTION
2	Time to Burn	2007	ACTION

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;



```
Select Movie_Num "Movie Number", MOVIE_TITLE "Movie Title",MOVIE_COST "Movie Cost"
from movie
where movie_cost > 40;
```

All Rows Fetched: 5 in 0.031 seconds

	Movie Number	Movie Title	Movie Cost
1	1235	Smokey Mountain Wildlife	59.95
2	1236	Richard Goodhope	59.95
3	1238	Constant Companion	89.95
4	1245	Time to Burn	45.49
5	1246	What He Doesn't Know	58.29

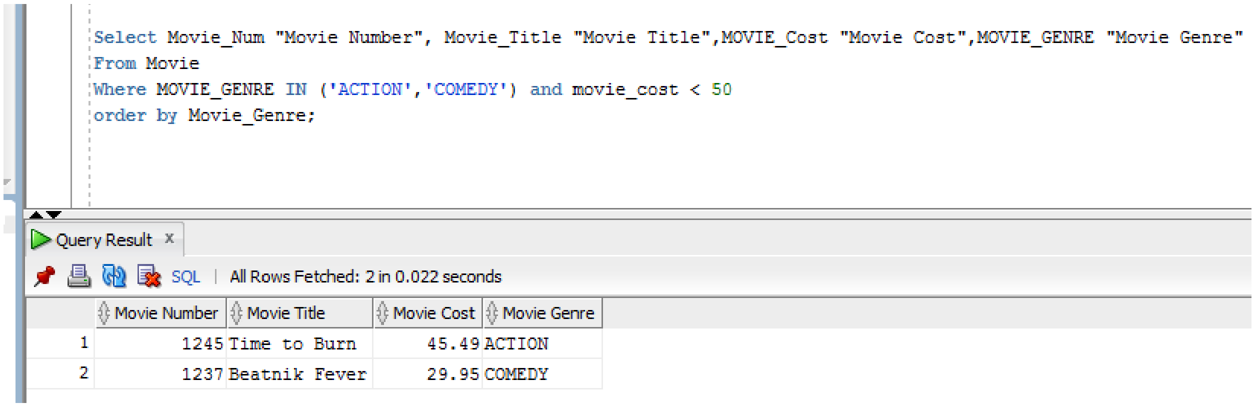
11.

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;



```
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;
```

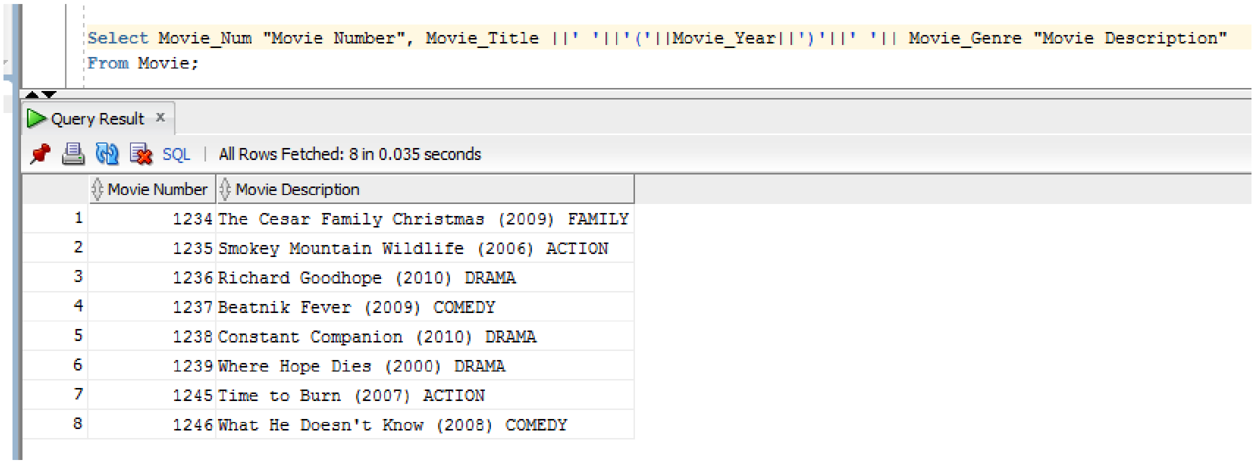
All Rows Fetched: 2 in 0.022 seconds

	Movie Number	Movie Title	Movie Cost	Movie Genre
1	1245	Time to Burn	45.49	ACTION
2	1237	Beatnik Fever	29.95	COMEDY

12.

Select Movie_Num "Movie Number", Movie_Title ||' '||'('||Movie_Year||')'||' '|| Movie_Genre "Movie Description"

From Movie;



```
Select Movie_Num "Movie Number", Movie_Title ||' '||'('||Movie_Year||')'||' '|| Movie_Genre "Movie Description"
From Movie;
```

All Rows Fetched: 8 in 0.035 seconds

	Movie Number	Movie Description
1	1234	The Cesar Family Christmas (2009) FAMILY
2	1235	Smokey Mountain Wildlife (2006) ACTION
3	1236	Richard Goodhope (2010) DRAMA
4	1237	Beatnik Fever (2009) COMEDY
5	1238	Constant Companion (2010) DRAMA
6	1239	Where Hope Dies (2000) DRAMA
7	1245	Time to Burn (2007) ACTION
8	1246	What He Doesn't Know (2008) COMEDY

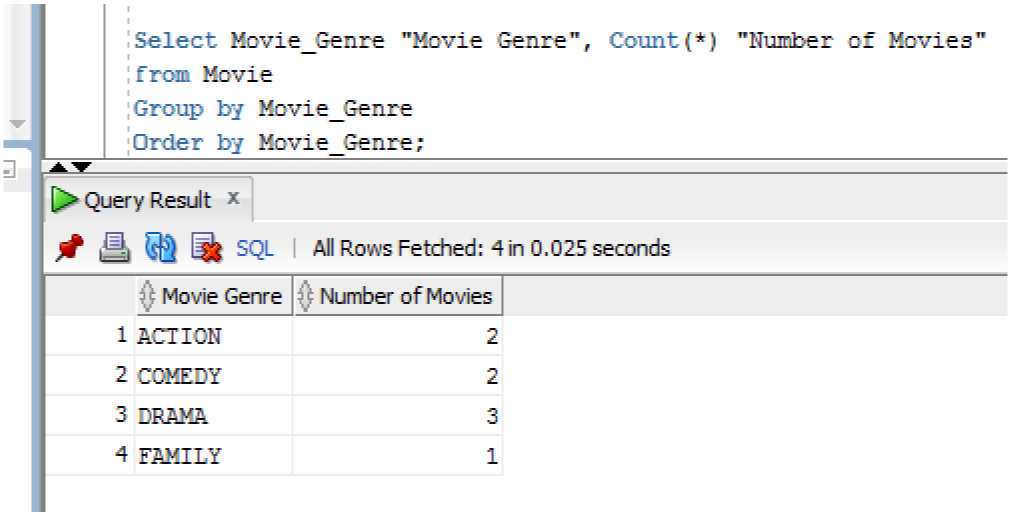
13.

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

from Movie

Group by Movie_Genre

Order by Movie_Genre;



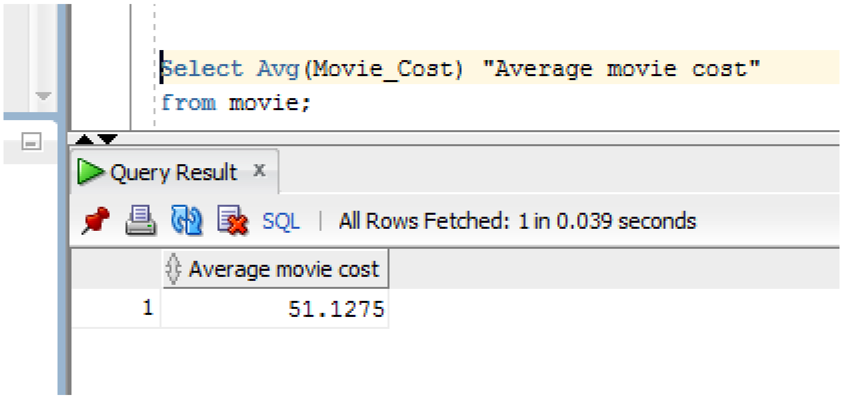
```
Select Movie_Genre "Movie Genre", Count(*) "Number of Movies"
from Movie
Group by Movie_Genre
Order by Movie_Genre;
```

```
All Rows Fetched: 4 in 0.025 seconds

    Movie Genre   Number of Movies
  1 ACTION                       2
  2 COMEDY                       2
  3 DRAMA                        3
  4 FAMILY                       1
```

14.

Select Avg(Movie_Cost) "Average movie cost"

from movie;



```
Select Avg(Movie_Cost) "Average movie cost"
from movie;
```

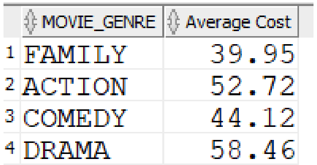
```
All Rows Fetched: 1 in 0.039 seconds

    Average movie cost
  1            51.1275
```

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;

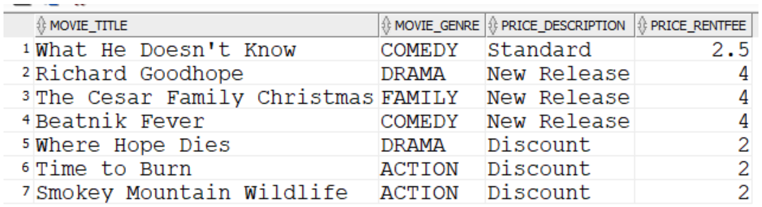


```
  MOVIE_GENRE  Average Cost
1 FAMILY              39.95
2 ACTION              52.72
3 COMEDY              44.12
4 DRAMA               58.46
```

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;



	MOVIE_TITLE	MOVIE_GENRE	PRICE_DESCRIPTION	PRICE_RENTFEE
1	What He Doesn't Know	COMEDY	Standard	2.5
2	Richard Goodhope	DRAMA	New Release	4
3	The Cesar Family Christmas	FAMILY	New Release	4
4	Beatnik Fever	COMEDY	New Release	4
5	Where Hope Dies	DRAMA	Discount	2
6	Time to Burn	ACTION	Discount	2
7	Smokey Mountain Wildlife	ACTION	Discount	2

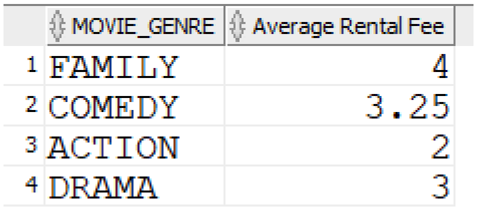
17.Write a query to display the movie genre and average price rental fee for movies in each genre that have a price

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;



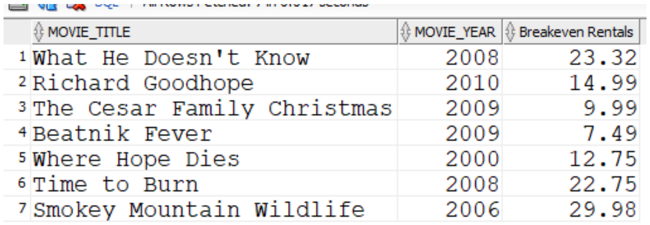
	MOVIE_GENRE	Average Rental Fee
1	FAMILY	4
2	COMEDY	3.25
3	ACTION	2
4	DRAMA	3

18. Write a query to display the movie title, movie year, and the movie cost divided <u>by the price rental fee for each movie that has a price</u> 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;



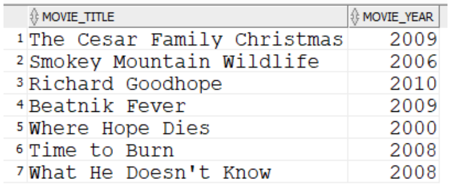
	MOVIE_TITLE	MOVIE_YEAR	Breakeven Rentals
1	What He Doesn't Know	2008	23.32
2	Richard Goodhope	2010	14.99
3	The Cesar Family Christmas	2009	9.99
4	Beatnik Fever	2009	7.49
5	Where Hope Dies	2000	12.75
6	Time to Burn	2008	22.75
7	Smokey Mountain Wildlife	2006	29.98

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;



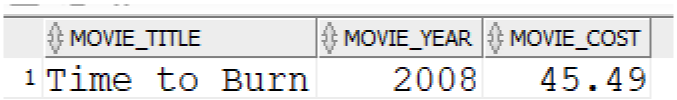
	MOVIE_TITLE	MOVIE_YEAR
1	The Cesar Family Christmas	2009
2	Smokey Mountain Wildlife	2006
3	Richard Goodhope	2010
4	Beatnik Fever	2009
5	Where Hope Dies	2000
6	Time to Burn	2008
7	What He Doesn't Know	2008

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;



	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST
1	Time to Burn	2008	45.49

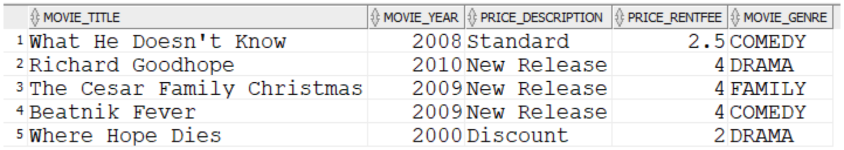
21. Write a query to display the movie title, movie year, price description, and price rental fee for all movies that are in the genres Family, Comedy, or Drama (result shown in Figure P7.89).

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');



	MOVIE_TITLE	MOVIE_YEAR	PRICE_DESCRIPTION	PRICE_RENTFEE	MOVIE_GENRE
1	What He Doesn't Know	2008	Standard	2.5	COMEDY
2	Richard Goodhope	2010	New Release	4	DRAMA
3	The Cesar Family Christmas	2009	New Release	4	FAMILY
4	Beatnik Fever	2009	New Release	4	COMEDY
5	Where Hope Dies	2000	Discount	2	DRAMA

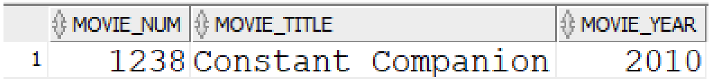
22. Write a query to display the movie number, movie title, and movie year for all movies that do not have a video.

select movie_num, m.movie_title, m.movie_year

from movie m LEFT JOIN video v

using (movie_num)

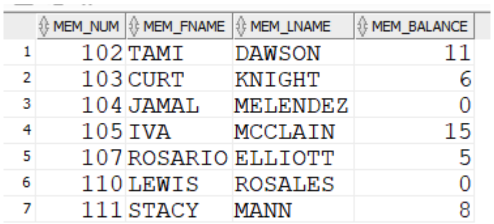
where vid_num IS NULL;



	MOVIE_NUM	MOVIE_TITLE	MOVIE_YEAR
1	1238	Constant Companion	2010

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

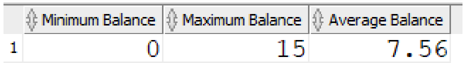
```
select distinct mem_num, m.mem_fname, m.mem_lname, m.mem_balance

from membership m left JOIN rental r

using(mem_num)

where r.rent_num is NOT NULL

order by 1;
```



	MEM_NUM	MEM_FNAME	MEM_LNAME	MEM_BALANCE
1	102	TAMI	DAWSON	11
2	103	CURT	KNIGHT	6
3	104	JAMAL	MELENDEZ	0
4	105	IVA	MCCLAIN	15
5	107	ROSARIO	ELLIOTT	5
6	110	LEWIS	ROSALES	0
7	111	STACY	MANN	8

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

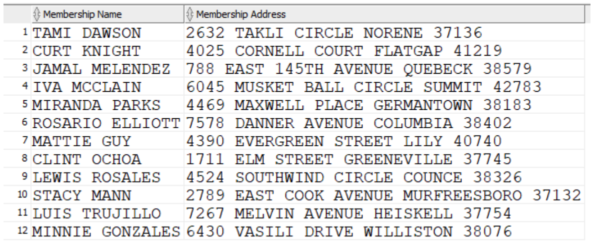
```
select min(mem_balance) "Minimum Balance" , max(mem_balance) "Maximum Balance",
round(avg(mem_balance),2) "Average Balance"

from membership m join rental r

using(mem_num)

where rent_num is NOT NULL;
```



	Minimum Balance	Maximum Balance	Average Balance
1	0	15	7.56

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;
```



	Membership Name	Membership Address
1	TAMI DAWSON	2632 TAKLI CIRCLE NORENE 37136
2	CURT KNIGHT	4025 CORNELL COURT FLATGAP 41219
3	JAMAL MELENDEZ	788 EAST 145TH AVENUE QUEBECK 38579
4	IVA MCCLAIN	6045 MUSKET BALL CIRCLE SUMMIT 42783
5	MIRANDA PARKS	4469 MAXWELL PLACE GERMANTOWN 38183
6	ROSARIO ELLIOTT	7578 DANNER AVENUE COLUMBIA 38402
7	MATTIE GUY	4390 EVERGREEN STREET LILY 40740
8	CLINT OCHOA	1711 ELM STREET GREENEVILLE 37745
9	LEWIS ROSALES	4524 SOUTHWIND CIRCLE COUNCE 38326
10	STACY MANN	2789 EAST COOK AVENUE MURFREESBORO 37132
11	LUIS TRUJILLO	7267 MELVIN AVENUE HEISKELL 37754
12	MINNIE GONZALES	6430 VASILI DRIVE WILLISTON 38076

```
26.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

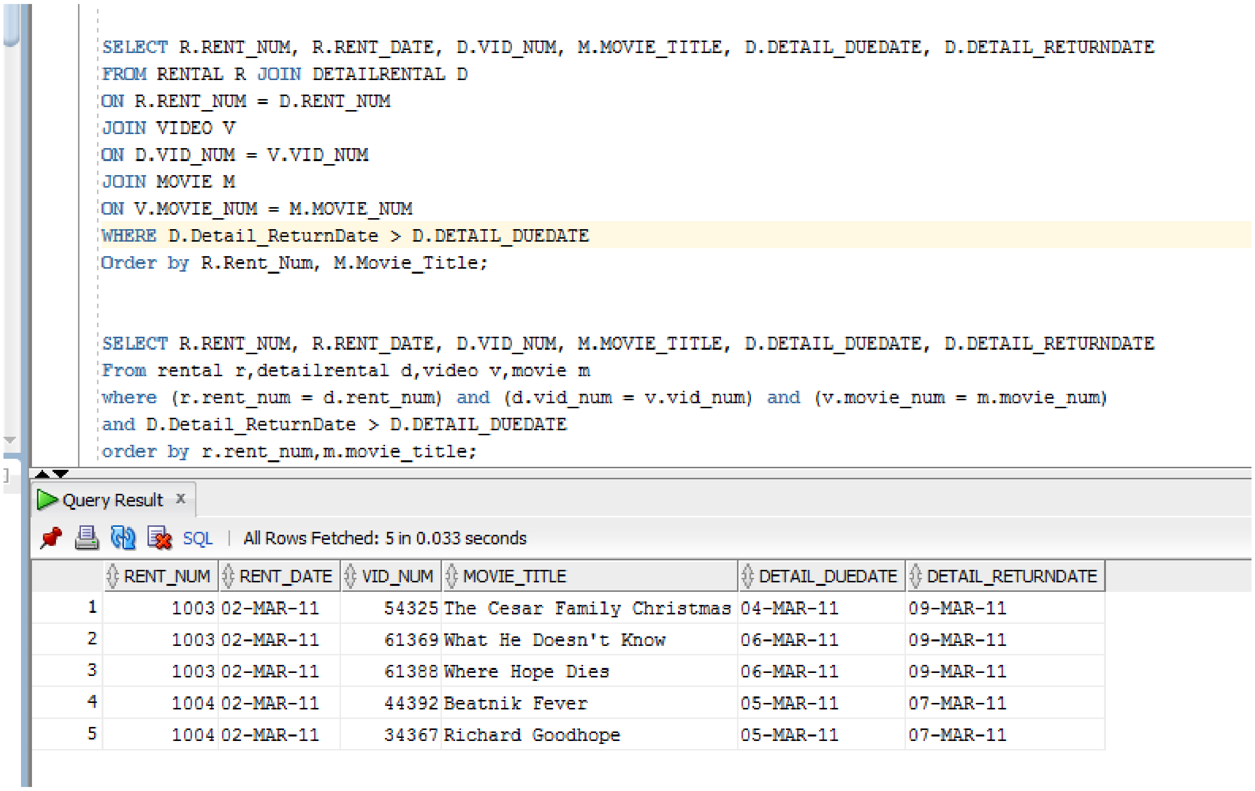
ON D.VID_NUM = V.VID_NUM

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;



```
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;


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;
```

All Rows Fetched: 5 in 0.033 seconds

	RENT_NUM	RENT_DATE	VID_NUM	MOVIE_TITLE	DETAIL_DUEDATE	DETAIL_RETURNDATE
1	1003	02-MAR-11	54325	The Cesar Family Christmas	04-MAR-11	09-MAR-11
2	1003	02-MAR-11	61369	What He Doesn't Know	06-MAR-11	09-MAR-11
3	1003	02-MAR-11	61388	Where Hope Dies	06-MAR-11	09-MAR-11
4	1004	02-MAR-11	44392	Beatnik Fever	05-MAR-11	07-MAR-11
5	1004	02-MAR-11	34367	Richard Goodhope	05-MAR-11	07-MAR-11

27.

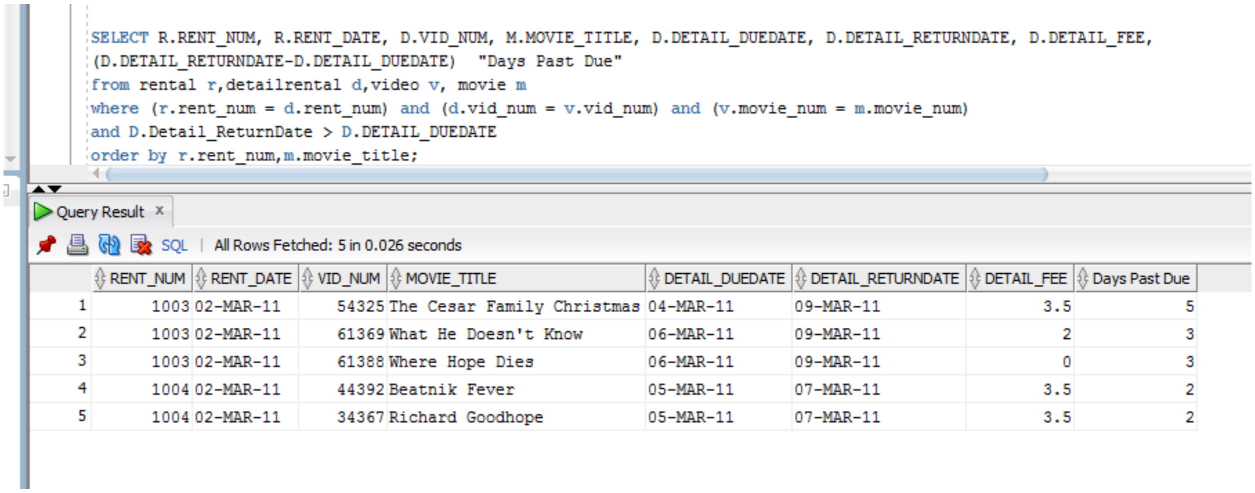
SELECT R.RENT_NUM, R.RENT_DATE, D.VID_NUM, M.MOVIE_TITLE, D.DETAIL_DUEDATE, D.DETAIL_RETURNDATE, D.DETAIL_FEE, (D.DETAIL_RETURNDATE-D.DETAIL_DUEDATE) "Days Past Due"

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;



```
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;
```

All Rows Fetched: 5 in 0.026 seconds

```
   RENT_NUM RENT_DATE VID_NUM MOVIE_TITLE               DETAIL_DUEDATE DETAIL_RETURNDATE DETAIL_FEE Days Past Due
1      1003 02-MAR-11   54325 The Cesar Family Christmas 04-MAR-11      09-MAR-11                3.5             5
2      1003 02-MAR-11   61369 What He Doesn't Know      06-MAR-11      09-MAR-11                  2             3
3      1003 02-MAR-11   61388 Where Hope Dies           06-MAR-11      09-MAR-11                  0             3
4      1004 02-MAR-11   44392 Beatnik Fever             05-MAR-11      07-MAR-11                3.5             2
5      1004 02-MAR-11   34367 Richard Goodhope          05-MAR-11      07-MAR-11                3.5             2
```

28.

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

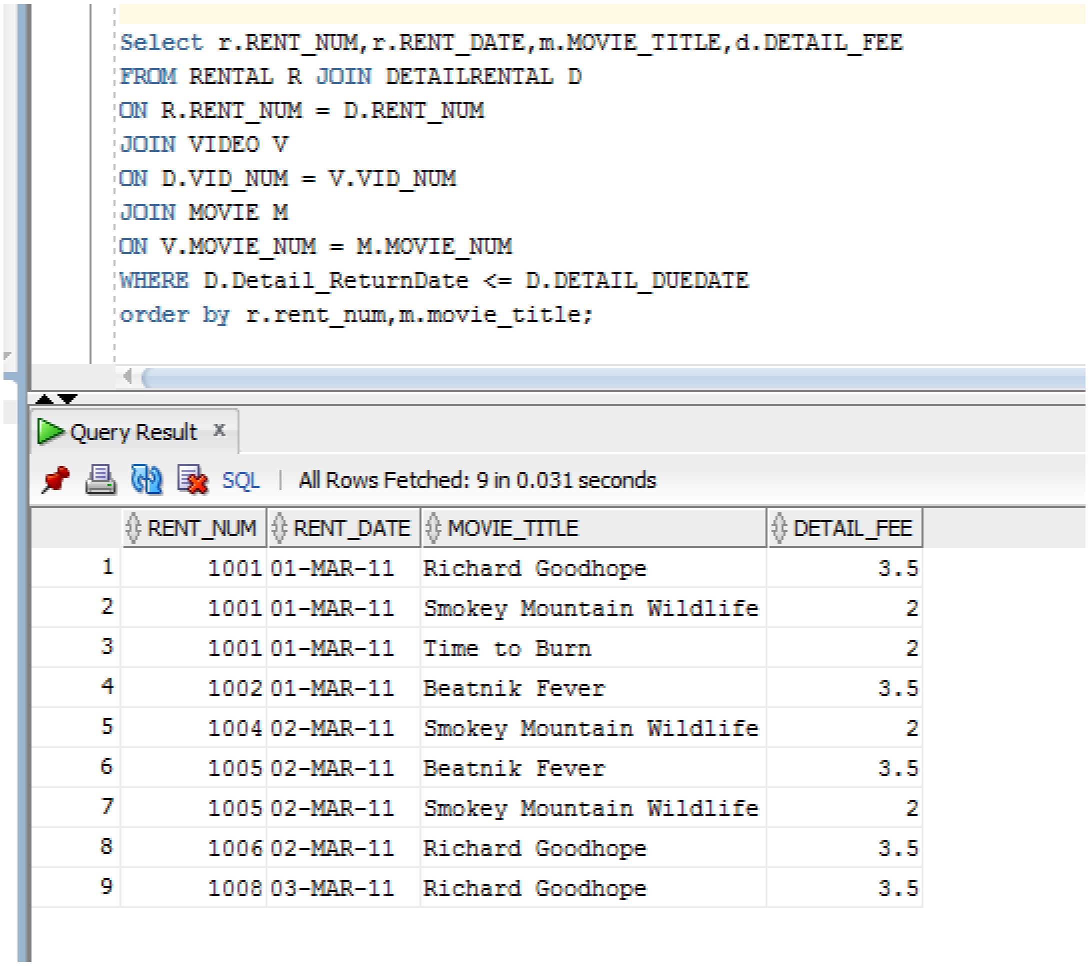
ON D.VID_NUM = V.VID_NUM

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;



```
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;
```

All Rows Fetched: 9 in 0.031 seconds

	RENT_NUM	RENT_DATE	MOVIE_TITLE	DETAIL_FEE
1	1001	01-MAR-11	Richard Goodhope	3.5
2	1001	01-MAR-11	Smokey Mountain Wildlife	2
3	1001	01-MAR-11	Time to Burn	2
4	1002	01-MAR-11	Beatnik Fever	3.5
5	1004	02-MAR-11	Smokey Mountain Wildlife	2
6	1005	02-MAR-11	Beatnik Fever	3.5
7	1005	02-MAR-11	Smokey Mountain Wildlife	2
8	1006	02-MAR-11	Richard Goodhope	3.5
9	1008	03-MAR-11	Richard Goodhope	3.5

29.

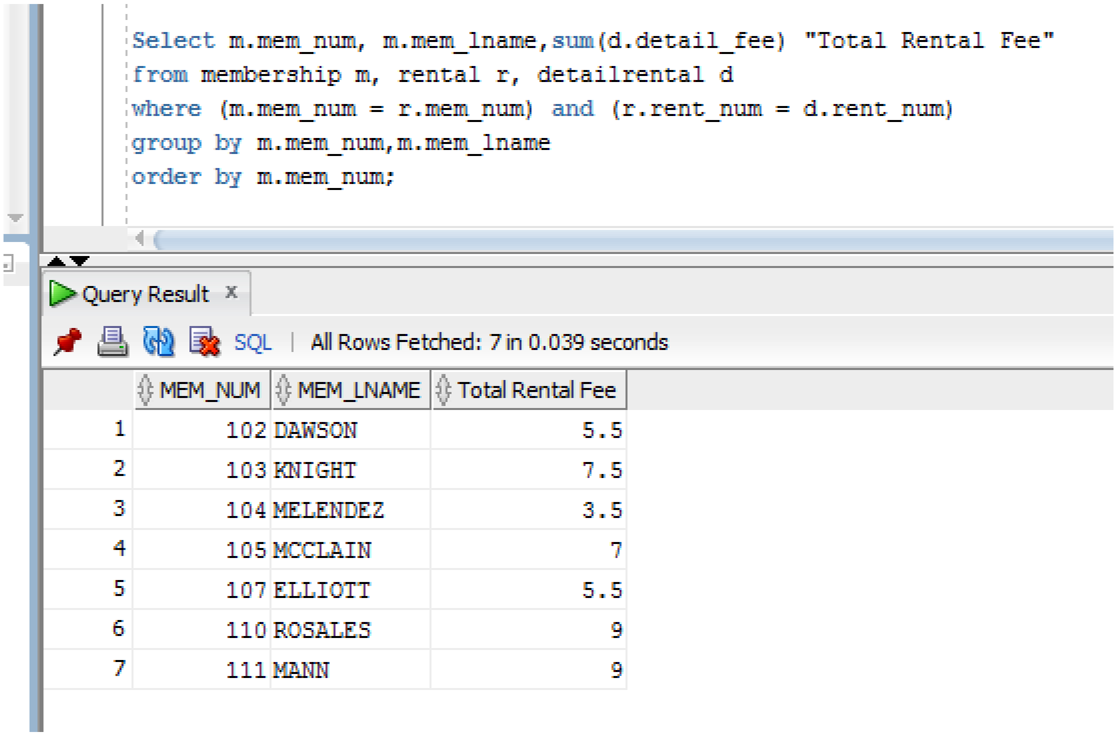
Select m.mem_num, m.mem_lname,sum(d.detail_fee) "Total Rental Fee"

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;



```
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;
```

All Rows Fetched: 7 in 0.039 seconds

	MEM_NUM	MEM_LNAME	Total Rental Fee
1	102	DAWSON	5.5
2	103	KNIGHT	7.5
3	104	MELENDEZ	3.5
4	105	MCCLAIN	7
5	107	ELLIOTT	5.5
6	110	ROSALES	9
7	111	MANN	9

30.

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).**

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.

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;