SQL_PROJECT_MOVIE_RENTAL

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

Sol)

UPDATE movie

SET movie_year = 2008

WHERE movie_num = 1245;

SELECT * from movie;

Output Figure 1

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

2. 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

sol)

SELECT movie_title , movie_year , movie_genre

FROM movie

ORDER BY movie_genre asc , movie_year desc;

output FIGURE 2

	MOVIE_TITLE	MOVIE_YEAR	MOVIE_GENRE
1	Time to Burn	2008	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

3. 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 F3).;

sol)

SELECT movie_title,movie_year,movie_cost

FROM movie

WHERE movie_title like '%hope%'

OR movie_title **like** '%Hope%';

RESULT FIGURE 3:

	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST
1	Richard Goodhope	2010	59.95
2	Where Hope Dies	2000	25.49

4. 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 or that have a cost that is less than $50. Sort the results in ascending order by genre. (Result shown in Figure F4.);

sol)

SELECT movie_num,movie_title,

movie_cost,movie_genre

FROM movie

WHERE movie_genre in ('ACTION' , 'COMEDY')

OR movie_cost< 50

ORDER BY movie_genre asc;

RESULT FIGURE 4:

	MOVIE_NUM	MOVIE_TITLE	MOVIE_COST	MOVIE_GENRE
1	1245	Time to Burn	45.49	ACTION
2	1235	Smokey Mountain Wildlife	59.95	ACTION
3	1246	What He Doesn't Know	58.29	COMEDY
4	1237	Beatnik Fever	29.95	COMEDY
5	1239	Where Hope Dies	25.49	DRAMA
6	1234	The Cesar Family Christmas	39.95	FAMILY

5. 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 F5).;

sol)

```
SELECT movie_num,
movie_title ||' '||'('||movie_year||')'|| ' '|| movie_genre
AS "Movie_Description"
from movie;
```

RESULT FIGURE 5:

	MOVIE_NUM	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 (2008) ACTION
8	1246	What He Doesn't Know (2008) COMEDY

6. Write a query to display the movie genre and the number of movies in each genre (result shown in Figure F6);

sol)

```
SELECT movie_genre,
COUNT(movie_genre) as "Number of Movies"
FROM movie
GROUP BY movie_genre
ORDER BY movie_genre ;
```

RESULT FIGURE 6:

	MOVIE_GENRE	Number of Movies
1	ACTION	2
2	COMEDY	2
3	DRAMA	3
4	FAMILY	1

7. Write a query to display the movie genre and average cost of movies in each genre (result shown in Figure F7).;

sol)

SELECT movie_genre,

ROUND(avg(movie_cost),2) as "Average Cost"

FROM movie

GROUP BY movie_genre

ORDER BY movie_genre;

RESULT FIGURE 7:

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

8. Write a query to display the movie title, movie genre, price description, and price rental fee for all movies with a price code (result shown in Figure F8).;

sol)

SELECT m.movie_title,m.movie_genre,

p.price_description,p.price_rentfee

from movie m

join price p

on p.price_code = m.price_code

ORDER BY price_description desc;

RESULT FIGURE 8:

Query Result

All Rows Fetched: 7 in 0.013 seconds

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

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

that have a price (result shown in Figure F9). ;

sol)

SELECT m.movie_genre,

AVG(p.price_rentfee) as "Average Rental Fee"

FROM movie m

INNER JOIN price p

ON m.price_code = p.price_code

GROUP BY movie_genre

ORDER BY movie_genre;

RESULT FIGURE 9:

Query Result

All Rows Fetched: 4 in 0.024 seconds

	MOVIE_GENRE	Average Rental Fee
1	ACTION	2.5
2	COMEDY	3.25
3	DRAMA	3
4	FAMILY	4

10. Write a query to display the minimum balance, maximum balance, and average balance for memberships that have a rental (result shown in Figure F10).;

sol)

SELECT min(mem_balance) as "Minimum Balance",

MAX(mem_balance) as "Maximum Balance",

ROUND(avg(mem_balance),2) as "Average Balance"

FROM membership;

RESULT FIGURE 10:

Query Result

All Rows Fetched: 1 in 0.006 seconds

	Minimum Balance	Maximum Balance	Average Balance
1	0	15	4.83

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

(result shown in Figure F11).;

sol)

SELECT movie_title,movie_year

FROM movie

WHERE price_code **IS NOT NULL**;

RESULT FIGURE 11:

Query Result

All Rows Fetched: 7 in 0.004 seconds

	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

12. 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 (result shown in Figure F12);

sol)

SELECT movie_title,movie_year,movie_cost

FROM movie

WHERE movie_cost >44.99 and movie_cost<49.99;

RESULT FIGURE 12:

Query Result

All Rows Fetched: 1 in 0.004 seconds

	MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST
1	Time to Burn	2008	45.49

13. 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 F13).

Figure F13 Movies with specific genres;

sol)

```
SELECT m.movie_title,m.movie_year,
p.price_description,p.price_rentfee,m.movie_genre
FROM movie m
INNER JOIN price p
ON m.price_code  = p.price_code
WHERE m.movie_genre in ('FAMILY','DRAMA','COMEDY');
```

RESULT FIGURE 13:

Query Result

All Rows Fetched: 5 in 0.004 seconds

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

14. Write a query to display the movie number, movie title, and movie year for all movies that do not have a video (result shown in Figure F14).;

sol)

```
SELECT movie_num,movie_title, movie_year
FROM movie
WHERE movie_num not in (SELECT movie_num
                        FROM video) ;
```

RESULT FIGURE 14:

Query Result

All Rows Fetched: 1 in 0.005 seconds

	MOVIE_NUM	MOVIE_TITLE	MOVIE_YEAR
1	1238	Constant Companion	2010

15. Write a query to display the membership number, first name, last name, and balance of the

memberships that have a rental (result shown in Figure F15).

Figure F15 Balances of memberships with rentals;

sol)

SELECT DISTINCT(m.mem_num),m.mem_fname,

m.mem_lname,m.mem_balance

FROM membership m

INNER JOIN rental r

ON m.mem_num = r.mem_num

ORDER BY mem_num;

RESULT FIGURE 15:

Query Result

SQL | All Rows Fetched: 7 in 0.005 seconds

	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

16. 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 F16).;

sol)

SELECT r.rent_num,r.rent_date,v.vid_num, m.movie_title,

d.detail_duedate,d.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 m.movie_num = v.movie_num

WHERE d.DETAIL_RETURNDATE>d.DETAIL_DUEDATE

ORDER BY rent_num asc , movie_title;

RESULT FIGURE 16:

Query Result

SQL | All Rows Fetched: 5 in 0.006 seconds

	RENT_NUM	RENT_DATE	VID_NUM	MOVIE_TITLE	DETAIL_DUEDATE	DETAIL_RETURNDATE
1	1003	02-03-11	54325	The Cesar Family Christmas	04-03-11	09-03-11
2	1003	02-03-11	61369	What He Doesn't Know	06-03-11	09-03-11
3	1003	02-03-11	61388	Where Hope Dies	06-03-11	09-03-11
4	1004	02-03-11	44392	Beatnik Fever	05-03-11	07-03-11
5	1004	02-03-11	34367	Richard Goodhope	05-03-11	07-03-11

17. 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 F17)

sol)

SELECT CONCAT(mem_fname,concat(' ',mem_lname)) as "Membership name",

mem_street||','||mem_city||','||mem_state||' '||Mem_Zip as "Membership address"

FROM membership;

RESULT FIGURE 17:

Query Result

SQL | All Rows Fetched: 12 in 0.004 seconds

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

18. 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 F18.);

sol)

```
SELECT r.rent_num,r.rent_date,v.vid_num, m.movie_title,
d.detail_duedate,d.detail_returndate,
d.detail_fee,(d.detail_returndate-d.detail_duedate)  AS "days past due date"
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 m.movie_num = v.movie_num
WHERE (d.DETAIL_RETURNDATE>d.DETAIL_DUEDATE)
order by rent_num asc , movie_title;
```

RESULT FIGURE 18:

Query Result

All Rows Fetched: 5 in 0.003 seconds

	RENT_NUM	RENT_DATE	VID_NUM	MOVIE_TITLE	DETAIL_DUEDATE	DETAIL_RETURNDATE	DETAIL_FEE	days past due date
1	1003	02-03-11	54325	The Cesar Family Christmas	04-03-11	09-03-11	3.5	5
2	1003	02-03-11	61369	What He Doesn't Know	06-03-11	09-03-11	2	3
3	1003	02-03-11	61388	Where Hope Dies	06-03-11	09-03-11	0	3
4	1004	02-03-11	44392	Beatnik Fever	05-03-11	07-03-11	3.5	2
5	1004	02-03-11	34367	Richard Goodhope	05-03-11	07-03-11	3.5	2

19. 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 F19).;

sol)

```
SELECT r.rent_num,r.rent_date,
m.movie_title,d.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 m.movie_num = v.movie_num

WHERE d.DETAIL_RETURNDATE<=d.DETAIL_DUEDATE;

RESULT FIGURE 19:

Query Result

All Rows Fetched: 9 in 0.007 seconds

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

20. Write a query to display the membership number, last name, and total rental fees earned from that membership (result shown in Figure F20). 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. ;

sol)

SELECT m.mem_num,m.mem_lname,m.mem_fname,

SUM(d.detail_fee) as "rental fee revenue"

FROM membership m

INNER JOIN rental r

ON r.mem_num=m.mem_num

INNER JOIN detailrental d

ON d.rent_num=r.rent_num

GROUP BY m.mem_num, m.mem_lname, m.mem_fname

ORDER BY mem_num;

RESULT FIGURE 20:

Query Result

SQL | All Rows Fetched: 7 in 0.007 seconds

	MEM_NUM	MEM_LNAME	MEM_FNAME	rental fee revenue
1	102	DAWSON	TAMI	5.5
2	103	KNIGHT	CURT	7.5
3	104	MELENDEZ	JAMAL	3.5
4	105	MCCLAIN	IVA	7
5	107	ELLIOTT	ROSARIO	5.5
6	110	ROSALES	LEWIS	9
7	111	MANN	STACY	9