

# SQL MICRO-PROJECT

NAME:DHANALAKSHMI B

EMAIL ID:10DROSSDL@GMAIL.COM

Table created and data filled by running the script given.

Table PRICE created.	1 row inserted.
Table MOVIE created.	1 row inserted.
Table VIDEO created.	1 row inserted.
Table MEMBERSHIP created.	1 row inserted.
Table RENTAL created.	1 row inserted.
Table DETAILRENTAL created.	1 row inserted.

Q1.

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

QUERY:

UPDATE MOVIE

SET MOVIE\_YEAR=2008

WHERE MOVIE\_NUM=1245;

SCREENSHOT:

6	1239 Where Hope Dies	2000
7	1245 Time to Burn	2008

1 row updated.

Q2.

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.

QUERY:

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

ORDER BY MOVIE\_GENRE , MOVIE\_YEAR DESC;

SCREENSHOT:

	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

Q3.

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.

QUERY:

```
SELECT MOVIE_TITLE, MOVIE_YEAR, MOVIE_COST
FROM MOVIE
WHERE UPPER(MOVIE_TITLE) LIKE '%HOPE%'
ORDER BY MOVIE_TITLE;
```

SCREENSHOT:

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

Q4.

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.

QUERY:

```
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;
```

SCREENSHOT:

	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

Q5.

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.

QUERY:

```
SELECT MOVIE_NUM, CONCAT(MOVIE_TITLE,  
CONCAT(' (' ,  
CONCAT(MOVIE_YEAR,  
CONCAT(') ',MOVIE_GENRE)))) AS MOVIE_DESCRIPTION  
FROM MOVIE;
```

Or

```
SELECT MOVIE_NUM,  
(MOVIE_TITLE||'('||MOVIE_YEAR||')'||MOVIE_GENRE)AS MOVIE_DESCRIPTION  
FROM MOVIE;
```

SCREENSHOT:

MOVIE_NUM	MOVIE_DESCRIPTION
1234	The Cesar Family Christmas (2009) FAMILY
1235	Smokey Mountain Wildlife (2006) ACTION
1236	Richard Goodhope (2010) DRAMA
1237	Beatnik Fever (2009) COMEDY
1238	Constant Companion (2010) DRAMA
1239	Where Hope Dies (2000) DRAMA
1245	Time to Burn (2008) ACTION
1246	What He Doesn't Know (2008) COMEDY

Q6.

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

QUERY:

```
SELECT MOVIE_GENRE,COUNT(MOVIE_GENRE) AS "NUMBER OF MOVIES" FROM MOVIE
GROUP BY MOVIE_GENRE
ORDER BY MOVIE_GENRE;
```

SCREEN SHOT:

	MOVIE_GENRE	NUMBER OF MOVIES
1	ACTION	2
2	COMEDY	2
3	DRAMA	3
4	FAMILY	1

Q7.

Write a query to display the movie genre and average cost of movies in each genre.

QUERY:

```
SELECT MOVIE_GENRE,ROUND(AVG(MOVIE_COST),2) AS "AVERAGE COST"
FROM MOVIE
GROUP BY MOVIE_GENRE
ORDER BY MOVIE_GENRE;
```

SCREENSHOT:

MOVIE_GENRE	AVERAGE COST
ACTION	52.72
COMEDY	44.12
DRAMA	58.46
FAMILY	39.95

Q8.

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

QUERY:

```
SELECT M.MOVIE_TITLE, M.MOVIE_GENRE,P.PRICE_DESCRIPTION, P.PRICE_RENTFEE
FROM PRICE P
INNER JOIN MOVIE M ON P.PRICE_CODE=M.PRICE_CODE
ORDER BY PRICE_DESCRIPTION DESC;
```

or

```
SELECT MOVIE_TITLE, MOVIE_GENRE, PRICE_DESCRIPTION, PRICE_RENTFEE
FROM PRICE P, MOVIE M
WHERE P.PRICE_CODE=M.PRICE_CODE
ORDER BY PRICE_DESCRIPTION DESC;
```

SCREENSHOT:

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

Q9.

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

QUERY:

```
SELECT MOVIE_GENRE, ROUND(AVG(PRICE_RENTFEE), 2) 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;
```

SCREENSHOT:

MOVIE_GENRE	AVERAGE RENTAL FEE
1 ACTION	2.5
2 COMEDY	3.25
3 DRAMA	3
4 FAMILY	4

Q10.

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

QUERY:

```

SELECT MIN(MEM_BALANCE)AS MINIMUM_BALANCE ,MAX(MEM_BALANCE)AS
MAXIMUM_BALANCE, ROUND(AVG(MEM_BALANCE),2)AS AVERAGE_BALANCE FROM

(SELECT M.MEM_NUM,COUNT(M.MEM_NUM),M.MEM_BALANCE

FROM MEMBERSHIP M,RENTAL R

WHERE M.MEM_NUM=R.MEM_NUM

GROUP BY M.MEM_NUM,M.MEM_BALANCE HAVING COUNT(M.MEM_NUM)>=1);

```

SCREENSHOT:

MINIMUM_BALANCE	MAXIMUM_BALANCE	AVERAGE_BALANCE
0	15	6.43

Q11.

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

QUERY:

```

SELECT MOVIE_TITLE, MOVIE_YEAR

FROM PRICE P

INNER JOIN MOVIE M

ON P.PRICE_CODE=M.PRICE_CODE;

```

SCREENSHOT:

	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

Q12.

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.

QUERY:

```

SELECT MOVIE_TITLE,MOVIE_YEAR,MOVIE_COST

FROM MOVIE

WHERE MOVIE_COST BETWEEN 44.99 AND 49.99;

```

SCREENSHOT:

MOVIE_TITLE	MOVIE_YEAR	MOVIE_COST
Time to Burn	2008	45.49

Q13.

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.

QUERY:

```
SELECT MOVIE_TITLE, MOVIE_YEAR, PRICE_DESCRIPTION, PRICE_RENTFEE, MOVIE_GENRE
FROM MOVIE M
JOIN PRICE P ON P.PRICE_CODE=M.PRICE_CODE
WHERE MOVIE_GENRE IN ('FAMILY','COMEDY','DRAMA');
```

SCREENSHOT:

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

Q14.

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

QUERY:

```
SELECT MOVIE_NUM,MOVIE_TITLE,MOVIE_YEAR
FROM MOVIE
WHERE MOVIE_NUM NOT IN (SELECT MOVIE_NUM FROM VIDEO);
```

SCREENSHOT:

MOVIE_NUM	MOVIE_TITLE	MOVIE_YEAR
1	1238 Constant Companion	2010

Q15.

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

**QUERY:**

```
SELECT UNIQUE(M.MEM_NUM),MEM_FNAME,MEM_LNAME,MEM_BALANCE
FROM MEMBERSHIP M
JOIN RENTAL R
ON M.MEM_NUM=R.MEM_NUM
ORDER BY MEM_NUM;
```

**SCREENSHOT:**

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

**Q16.**

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.

**QUERY:**

```
SELECT R.RENT_NUM, TO_CHAR(R.RENT_DATE,'DD-MON-YY')AS RENT_DATE
,V.VID_NUM,M.MOVIE_TITLE,TO_CHAR(D.DETAIL_DUEDATE,'DD-MON-YY')AS
DETAIL_DUEDATE,TO_CHAR(D.DETAIL_RETURNDATE,'DD-MON-YY')AS DETAIL_RETURNDATE
FROM MOVIE M
JOIN VIDEO V ON M.MOVIE_NUM=V.MOVIE_NUM
JOIN DETAILRENTAL D ON V.VID_NUM=D.VID_NUM
JOIN RENTAL R ON R.RENT_NUM=D.RENT_NUM
WHERE DETAIL_RETURNDATE>DETAIL_DUEDATE
ORDER BY RENT_NUM,MOVIE_TITLE;
```



SCREENSHOT:

	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

Q17.

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).

QUERY:

```
SELECT MEM_FNAME||' '||MEM_LNAME AS MEMBERSHIP_NAME,  
MEM_STREET||' '||MEM_CITY||', '||MEM_STATE||', '||MEM_ZIP AS MEMBERSHIP_ADDRESS  
FROM MEMBERSHIP;
```

SCREENSHOT:

	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

Q18.

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.

QUERY:

```
SELECT R.RENT_NUM,TO_CHAR(R.RENT_DATE,'DD-MON-YY')AS  
RENT_DATE,V.VID_NUM,M.MOVIE_TITLE,TO_CHAR(D.DETAIL_DUEDATE,'DD-MON-YY')AS  
DETAIL_DUEDATE,
```

```

TO_CHAR(D.DETAIL_RETURNDATE,'DD-MON-YY')AS
DETAIL_RETURNDATE,D.DETAIL_FEE,D.DETAIL_RETURNDATE-D.DETAIL_DUEDATE AS "DAYS PASSED
DUE"

FROM MOVIE M

JOIN VIDEO V ON M.MOVIE_NUM=V.MOVIE_NUM

JOIN DETAILRENTAL D ON V.VID_NUM=D.VID_NUM

JOIN RENTAL R ON R.RENT_NUM=D.RENT_NUM

WHERE DETAIL_RETURNDATE>DETAIL_DUEDATE

ORDER BY RENT_NUM,MOVIE_TITLE;

```

**SCREENSHOT:**

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

**Q19.**

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.

**QUERY:**

```

SELECT R.RENT_NUM,TO_CHAR(R.RENT_DATE,'DD-MON-YY')AS
RENT_DATE,M.MOVIE_TITLE,D.DETAIL_FEE FROM MOVIE M

JOIN VIDEO V ON M.MOVIE_NUM=V.MOVIE_NUM

JOIN DETAILRENTAL D ON V.VID_NUM=D.VID_NUM

JOIN RENTAL R ON R.RENT_NUM=D.RENT_NUM

WHERE DETAIL_RETURNDATE<=DETAIL_DUEDATE

ORDER BY RENT_NUM;

```

**SCREENSHOT:**

RENT_NUM	RENT_DATE	MOVIE_TITLE	DETAIL_FEE
1001	01-MAR-11	Smokey Mountain Wildlife	2
1001	01-MAR-11	Richard Goodhope	3.5
1001	01-MAR-11	Time to Burn	2
1002	01-MAR-11	Beatnik Fever	3.5
1004	02-MAR-11	Smokey Mountain Wildlife	2
1005	02-MAR-11	Beatnik Fever	3.5
1005	02-MAR-11	Smokey Mountain Wildlife	2
1006	02-MAR-11	Richard Goodhope	3.5
1008	03-MAR-11	Richard Goodhope	3.5

**Q20.**

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

**QUERY:**

```
SELECT M.MEM_NUM, M.MEM_LNAME, M.MEM_FNAME, SUM(D.DETAIL_FEE) AS "RENTAL FEE  
REVENUE"
```

```
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, M.MEM_FNAME
```

```
ORDER BY MEM_NUM;
```

**SCREENSHOT:**

MEM_NUM	MEM_LNAME	MEM_FNAME	RENTAL FEE REVENUE
102	DAWSON	TAMI	5.5
103	KNIGHT	CURT	7.5
104	MELENDEZ	JAMAL	3.5
105	MCCLAIN	IVA	7
107	ELLIOTT	ROSARIO	5.5
110	ROSALES	LEWIS	9
111	MANN	STACY	9