Heaven's Light is Our Guide Rajshahi University of Engineering and Technology



Course Code ECE 2216

Course Title

Database Systems Sessional

Experiment Date: November 21, 2023, Submission Date: November 26, 2023

Lab Report 6: Creating a database with primary and foreign keys and doing operations on it with MySQL using JOIN.

Submitted to
Md. Robiul Islam
Assistant Professor
Dept of ECE, RUET

Submitted by Md. Tajim An Noor Roll: 2010025

Contents

1	Too	ls Use	\mathbf{d}	3
2	Pro	cess		3
	2.1	Previo	ous lab's problems using join	3
		2.1.1	Find the customer name who are given highest commission	3
		2.1.2	Find the amount of total commission given by salesman_id =	
			"5001"	3
		2.1.3	Find the salesman name whose customer's grade is lowest	4
		2.1.4	Find the total order number, ner from 5-9-2016 to 17-10-216	
			from the given table	4
	2.2	New o	lata	5
		2.2.1	Find the domestic & international sales for each movie	6
		2.2.2	Show the sales number for each movie that did better inter-	
			nationally rather than domestically	6
		2.2.3	List all the movies by their ratings in descending order	6
		2.2.4	Find the movie ID, name of the movies that the highest length	
			movie of each director	7
		2.2.5	Find the local domestic sales of movies by John Lasseter so far.	7
3	Out	put		8

1 Tools Used

- MySQL
- VS Code as an IDE to use SQL
- MacTeX -LATEX compiler
- VS Code with LaTeX workshop extension as a text editor

2 Process

SQL Codes:

- 2.1 Previous lab's problems using join.
- 2.1.1 Find the customer name who are given highest commission.

Code:

```
c.customer_name as "Customer Name",
s.commission as Commission

FROM
Relations.Customer c
JOIN Relations.salesman s ON c.salesman_id = s.salesman_id
ORDER BY
s.commission DESC
LIMIT
10 1;
```

2.1.2 Find the amount of total commission given by salesman_id = "5001".

```
SELECT
s.name as "Salesman Name",
SUM(o.purchase_amount * s.commission) as "Total Commission"
```

```
FROM
Relations.Order o
JOIN Relations.salesman s ON o.salesman_id = s.salesman_id
WHERE
o.salesman_id = 5001;
```

2.1.3 Find the salesman name whose customer's grade is lowest.

Code:

```
SELECT
       s.name as "Salesman Name"
   FROM
       Relations.salesman s
       JOIN Relations.Customer c ON s.salesman_id = c.salesman_id
5
   WHERE
       c.grade = (
           SELECT
                MIN(grade)
           FROM
10
                Relations.Customer
11
       );
12
```

2.1.4 Find the total order number,ner from 5-9-2016 to 17-10-216 from the given table.

```
SELECT
COUNT(o.order_no) as "Total Order"
FROM
Relations.Order o
JOIN Relations.Customer c ON o.customer_id = c.customer_id
WHERE
o.order_date BETWEEN '2016-09-05'
AND '2016-10-17';
```

2.2 New data.

Creating the new tables and inserting data.

```
('Finding Nemo', 'Andrew

→ Stanton', 2003, 107),

   CREATE DATABASE IF NOT
                                                  ('The Incredibles',
      EXISTS movie;
                                                   → 'Brad Bird', 2004,
                                                       116),
   CREATE TABLE IF NOT EXISTS
                                                  ('Cars', 'John
       movie.movies(
                                                   → Lasseter', 2006,
       id INT AUTO_INCREMENT
                                                   \rightarrow 117),
        → PRIMARY KEY,
                                                  ('Ratatouille', 'Brad
       title VARCHAR(50),
                                                   → Bird', 2007, 115),
       director VARCHAR(50),
                                                  ('WALL-E', 'Andrew
                                          30
       year INT,
                                                   → Stanton', 2008, 104),
       length_minutes INT
                                                  ('Up', 'Pete Docter',
                                          31
   );
                                                   \rightarrow 2009, 101),
10
                                                  ('Toy Story 3', 'Lee
   CREATE TABLE
11
                                                   → Unkrich', 2010, 103),
       movie.box_office(
                                                  ('Cars 2', 'John
                                          33
       movie_id INT,
12

    Lasseter', 2011,

       rating DECIMAL(3, 1),
13
                                                     120),
       domestic_sales BIGINT,
14
                                                  ('Brave', 'Brenda
                                          34
       international_sales
15
                                                   → Chapman', 2012, 102),
        → BIGINT.
                                                  ('Monsters University',
       CONSTRAINT box2movies
16
                                                       'Dan Scanlon', 2013,
          FOREIGN KEY
                                                       110);
           (movie_id)
                                          36
           REFERENCES
                                              INSERT INTO
                                          37
            movie.movies(id)
                                                  movie.box_office (
                                          38
   );
17
                                                       movie_id,
                                          39
18
                                                       rating,
   INSERT INTO
19
                                                       domestic_sales,
       movie.movies (title,
20
                                                       international_sales
                                          42
            director, year,
                                                  )
                                          43
            length_minutes)
                                              VALUES
                                          44
   VALUES
21
                                                  (5, 8.2, 380843261,
                                          45
        ('Toy Story', 'John
22

→ 555900000),

        → Lasseter', 1995, 81),
                                                  (14, 7.4, 268492764,
        ('A Bugs Life', 'John
23

→ 475066843),

        → Lasseter', 1998, 95),
                                                  (8, 8, 206445654,
        ('Toy Story 2', 'John
24

→ 417277164),

        → Lasseter', 1999, 93),
                                                  (12, 6.4, 191452396,
                                          48
        ('Monsters, Inc.', 'Pete
25
                                                   \rightarrow 368400000),
        → Docter', 2001, 92),
```

```
(3, 7.9, 245852179,
                                                 (10, 8.3, 293004164,
49

→ 239163000),

→ 438338580),

       (6, 8, 261441092,
                                                 (4, 8.1, 289916256,
50

→ 272900000),

→ 370001000),

       (9, 8.5, 223808164,
                                                 (2, 7.2, 162798565,
51
                                                 → 200600000),

→ 297503696),

       (11, 8.4, 415004880,
                                                 (13, 7.2, 237283207,
52
                                                 → 301700000);

→ 648167031),

       (7, 8.3, 191796233,
        → 170162503),
```

2.2.1 Find the domestic & international sales for each movie.

Code:

```
title as Title,
fORMAT(domestic_sales, 'N') as "Domestic Sales",
fORMAT(international_sales, 'N') as "International Sales"
fROM
movie.movies m
JOIN movie.box_office b ON b.movie_id = m.id;
```

2.2.2 Show the sales number for each movie that did better internationally rather than domestically.

Code:

```
title as Title,
fORMAT(domestic_sales, 'N') as "Domestic Sales",
fORMAT(international_sales, 'N') as "International Sales"
fROM
movie.movies m
JOIN movie.box_office b ON b.movie_id = m.id
WHERE
international_sales > domestic_sales;
```

2.2.3 List all the movies by their ratings in descending order.

```
SELECT title as Title,
```

```
rating as Rating
FROM
movie.movies m
JOIN movie.box_office b ON b.movie_id = m.id
ORDER BY
rating DESC;
```

2.2.4 Find the movie ID, name of the movies that the highest length movie of each director.

Code:

```
SELECT
       m.id AS "Movie ID",
       m.title as Title,
       m.director as Director,
       m.length_minutes as "Length(Minutes)"
   FROM
       movie.movies m
   WHERE
       (m.director, m.length_minutes) IN (
9
            SELECT
10
                director,
11
                MAX(length_minutes) AS max_length
           FROM
                movie.movies
14
           GROUP BY
15
                director
16
       );
17
```

2.2.5 Find the local domestic sales of movies by John Lasseter so far.

```
1 SELECT
2     director as Director,
3     FORMAT(SUM(domestic_sales), 'N') as "Total Domestic Sales"
4 FROM
5     movie.movies m
6     JOIN movie.box_office b ON m.id = b.movie_id
7 WHERE
8     director = 'John Lasseter';
```

3 Output



The complete Customer table.

order_no	purchase_amo	order_date	customer_id	salesman_id
₫c Filter	alc Filter	a ⊡ c Filter	abc Filter	abc Filter
70001	150.50	2012-10-05	3005	5002
70002	65.26	2016-10-05	3002	5001
70003	2480.40	2012-10-10	3009	5003
70004	110.50	2012-08-17	3009	5003
70005	2400.60	2012-07-27	3007	5001
70007	948.50	2012-09-10	3005	5002
70008	5760.00	2012-09-10	3002	5001
70009	270.65	2012-09-10	3001	5005
70010	1983.43	2012-10-10	3004	5006
70011	75.29	2012-08-17	3003	5007
70012	250.45	2012-06-27	3008	5002
70013	3045.60	2012-04-25	3002	5001

The complete Order table.

salesman_id	name	city	commission
abc Filter	abc Filter	а <mark>Б</mark> с Filter	а <mark>Б</mark> с Filter
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5003	Lauson Hen	NULL	0.12
5005	Pit Alex	London	0.11
5006	Mc Lyon	New York	0.14
5007	Pal Adam	Rome	0.13

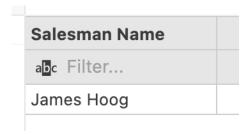
The complete Salesman table.



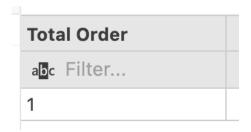
Customer who are given highest commission.



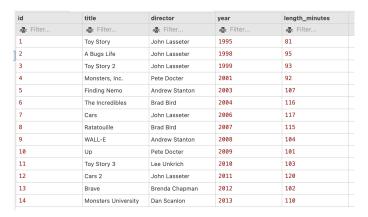
Amount of total commission given by salesman_id = "5001"



Salesman name whose customer's grade is lowest.



Total order number from 5-9-2016 to 17-10-2016



The complete Movies table.

movie_id	rating	domestic_sales	international_s	
a b c Filter	abc Filter	а <mark>Б</mark> с Filter	а <mark>Б</mark> с Filter	
5	8.2	380843261	555900000	
14	7.4	268492764	475066843	
8	8.0	206445654	417277164	
12	6.4	191452396	368400000	
3	7.9	245852179	239163000	
6	8.0	261441092	370001000	
9	8.5	223808164	297503696	
11	8.4	415004880	648167031	
7	8.3	191796233	170162503	
10	8.3	293004164	438338580	
4	8.1	289916256	272900000	
2	7.2	162798565	200600000	
13	7.2	237283207	301700000	

The complete Box Office table.

Title	Domestic Sales	International S
alc Filter	alc Filter	abc Filter
Finding Nemo	380,843,261	555,900,000
Monsters University	268,492,764	475,066,843
Ratatouille	206,445,654	417,277,164
Cars 2	191,452,396	368,400,000
Toy Story 2	245,852,179	239,163,000
The Incredibles	261,441,092	370,001,000
WALL-E	223,808,164	297,503,696
Toy Story 3	415,004,880	648,167,031
Cars	191,796,233	170,162,503
Up	293,004,164	438,338,580
Monsters, Inc.	289,916,256	272,900,000
A Bugs Life	162,798,565	200,600,000
Brave	237,283,207	301,700,000

Domestic & international sales for each movie.



Sales for each movie that did better internationally rather than domestically.

Title	Rating
abc Filter	abc Filter
WALL-E	8.5
Toy Story 3	8.4
Cars	8.3
Up	8.3
Finding Nemo	8.2
Monsters, Inc.	8.1
Ratatouille	8.0
The Incredibles	8.0
Toy Story 2	7.9
Monsters University	7.4
A Bugs Life	7.2
Brave	7.2
Cars 2	6.4

Movie ID	Title	Director	Length(Minutes
a @ c Filter	a ⊡ c Filter	a @ c Filter	a ⊡ c Filter
5	Finding Nemo	Andrew Stanton	107
6	The Incredibles	Brad Bird	116
10	Up	Pete Docter	101
11	Toy Story 3	Lee Unkrich	103
12	Cars 2	John Lasseter	120
13	Brave	Brenda Chapman	102
14	Monsters University	Dan Scanlon	110

List of all the movies by their ratings in descending order.

Movie ID, Name of the movies that the highest length movie of each director.

Director	Total Domestic
abc Filter	abc Filter
John Lasseter	791,899,373

Local domestic sales of movies by John Lasseter so far.