

## CO226 LAB-05

E/18/128

### Task1

```
Select MySQL 8.0 Command Line Client
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.5.5-10.4.22-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use E18128Lab05
Database changed
mysql> SELECT * from MOVIE
->
-> ;
+-----+-----+-----+-----+
| Movie_ID | Title                | Year | Director    |
+-----+-----+-----+-----+
| 101      | Gone with the wind   | 1939 | Victor Fleming |
+-----+-----+-----+-----+
1 row in set (0.10 sec)
```

```
Select MySQL 8.0 Command Line Client
mysql> INSERT INTO MOVIE VALUES(102,'Star Wars',1972,'George Lucas'),(103,'The Sound of Music',1965,'Robert Wise'),(104,'E.T',1982,'Steven Spielberg'),(105,'Titanic',1997,'James Cameron'),(106,'Snow White',1937,NULL),(107,'Avatar',2009,'James Cameron'),(108,'Raiders of the Lost Arc',1981,'Steven Spielberg');
Query OK, 7 rows affected (0.01 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql> SELECT * from MOVIE
-> ;
+-----+-----+-----+-----+
| Movie_ID | Title                | Year | Director    |
+-----+-----+-----+-----+
| 101      | Gone with the wind   | 1939 | Victor Fleming |
| 102      | Star Wars            | 1972 | George Lucas  |
| 103      | The Sound of Music   | 1965 | Robert Wise   |
| 104      | E.T                  | 1982 | Steven Spielberg |
| 105      | Titanic              | 1997 | James Cameron |
| 106      | Snow White           | 1937 | NULL          |
| 107      | Avatar               | 2009 | James Cameron |
| 108      | Raiders of the Lost Arc | 1981 | Steven Spielberg |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

```
mysql> CREATE TABLE RATING(Reviewer_ID int, Movie_ID int, Star int, RatingDate date,FOREIGN KEY(Movie_ID) REFERENCES MOVIE (Movie_ID),FOREIGN KEY(Reviewer_ID) REFERENCE
S REVIEWER (Reviewer_ID);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '' at line 1

mysql> CREATE TABLE REVIEWER(Reviewer_ID int, ReviewerName VARCHAR(100),PRIMARY KEY(Reviewer_ID));
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE RATING(Reviewer_ID int, Movie_ID int, Star int, RatingDate date,FOREIGN KEY(Movie_ID) REFERENCES MOVIE (Movie_ID),FOREIGN KEY(Reviewer_ID) REFERENCE
S REVIEWER (Reviewer_ID);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '' at line 1

mysql> CREATE TABLE RATING(Reviewer_ID int, Movie_ID int, Star int, RatingDate date,FOREIGN KEY(Movie_ID) REFERENCES MOVIE (Movie_ID),FOREIGN KEY(Reviewer_ID) REFERENCE
S REVIEWER (Reviewer_ID));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> INSERT INTO REVIEWER VALUES(201,'Sarah Martinez'),(202,'Daniel Lewis'),(203,'Brittany Harris'),(204,'Mike Anderson'),(205,'Chris Jackson'),(206,'Elizabeth Thomas
'),(207,'James Cameron'),(208,'Ashley White');
Query OK, 8 rows affected (0.00 sec)
Records: 8 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO RATING VALUES(201,101,2,'2011-01-22'),(201,101,4,'2011-01-27'),(202,106,4,NULL),(203,103,2,'2011-01-20'),(203,108,4,'2011-01-12'),(203,108,2,'2011-01
-30'),(204,101,3,'2011-01-09'),(205,103,3,'2011-01-27'),(205,104,2,'2011-01-22'),(205,108,4,NULL),(206,107,3,'2011-01-15'),(206,106,5,'2011-01-19'),(207,107,5,'2011-01-
20'),(208,104,3,'2011-01-02');
Query OK, 14 rows affected (0.10 sec)
Records: 14 Duplicates: 0 Warnings: 0
```

## Task2

1) Find all the details about the movies presented in the populated MOVIE table.

```
mysql> SELECT * FROM MOVIE;
+-----+-----+-----+-----+
| Movie_ID | Title          | Year | Director      |
+-----+-----+-----+-----+
| 101      | Gone with the wind | 1939 | Victor Fleming |
| 102      | Star Wars        | 1972 | George Lucas  |
| 103      | The Sound of Music | 1965 | Robert Wise   |
| 104      | E.T.             | 1982 | Steven Spielberg |
| 105      | Titanic          | 1997 | James Cameron |
| 106      | Snow White       | 1937 | NULL          |
| 107      | Avatar           | 2009 | James Cameron |
| 108      | Raiders of the Lost Arc | 1981 | Steven Spielberg |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

2) Find all the details about the movies directed by 'James Cameron'.

```
mysql> SELECT * FROM MOVIE where Director="James Cameron";
+-----+-----+-----+-----+
| Movie_ID | Title          | Year | Director      |
+-----+-----+-----+-----+
| 105      | Titanic        | 1997 | James Cameron |
| 107      | Avatar         | 2009 | James Cameron |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

3) Find all the details about the movies directed by 'James Cameron', on or after year 2000.

```
mysql> SELECT * FROM MOVIE where Director="James Cameron" AND Year>2000;
+-----+-----+-----+-----+
| Movie_ID | Title          | Year | Director      |
+-----+-----+-----+-----+
| 107      | Avatar         | 2009 | James Cameron |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

4) Find all the stars presented in the rating table.

```
mysql> SELECT Star FROM Rating;
+-----+
| Star |
+-----+
| 2     |
| 4     |
| 4     |
| 2     |
| 4     |
| 2     |
| 3     |
| 3     |
| 2     |
| 4     |
| 3     |
| 5     |
| 5     |
| 3     |
+-----+
14 rows in set (0.00 sec)
```

5) Find the distinct stars presented in the table.

```
mysql> SELECT Distinct Star FROM Rating;
```

Star
2
4
3
5

```
4 rows in set (0.00 sec)
```

6) Find movie id s and each movie's director.

```
mysql> SELECT Movie_ID, Director from MOVIE;
```

Movie_ID	Director
101	Victor Fleming
102	George Lucas
103	Robert Wise
104	Steven Spielberg
105	James Cameron
106	NULL
107	James Cameron
108	Steven Spielberg

```
8 rows in set (0.00 sec)
```

7) Find movie id s, title s, year s of the movies directed by ' Steven Spielberg '.

```
mysql> SELECT Movie_ID, Title, Year from MOVIE where Director="Steven Spielberg";
```

Movie_ID	Title	Year
104	E.T	1982
108	Raiders of the Lost Arc	1981

```
2 rows in set (0.00 sec)
```

8)Obtain the Cartesian product of the details presented in two tables MOVIE and RATING.

```
mysql> SELECT * FROM MOVIE,RATING where MOVIE.Movie_ID=RATING.Movie_ID;
```

Movie_ID	Title	Year	Director	Reviewer_ID	Movie_ID	Star	RatingDate
101	Gone with the wind	1939	Victor Fleming	201	101	2	2011-01-22
101	Gone with the wind	1939	Victor Fleming	201	101	4	2011-01-27
101	Gone with the wind	1939	Victor Fleming	204	101	3	2011-01-09
103	The Sound of Music	1965	Robert Wise	203	103	2	2011-01-20
103	The Sound of Music	1965	Robert Wise	205	103	3	2011-01-27
104	E.T	1982	Steven Spielberg	205	104	2	2011-01-22
104	E.T	1982	Steven Spielberg	208	104	3	2011-01-02
106	Snow White	1937	NULL	202	106	4	NULL
106	Snow White	1937	NULL	206	106	5	2011-01-19
107	Avatar	2009	James Cameron	206	107	3	2011-01-15
107	Avatar	2009	James Cameron	207	107	5	2011-01-20
108	Raiders of the Lost Arc	1981	Steven Spielberg	203	108	4	2011-01-12
108	Raiders of the Lost Arc	1981	Steven Spielberg	203	108	2	2011-01-30
108	Raiders of the Lost Arc	1981	Steven Spielberg	205	108	4	NULL

```
14 rows in set (0.00 sec)
```

9) Obtain the Cartesian product of the movie id and title from MOVIE table with movie id ,reviewer id and stars from RATING table.

```
mysql> SELECT MOVIE.Movie_ID, MOVIE.Title, RATING.Movie_ID, RATING.Reviewer_ID, RATING.Star FROM RATING,MOVIE where RATING.Movie_ID=MOVIE.Movie_ID;
```

Movie_ID	Title	Movie_ID	Reviewer_ID	Star
101	Gone with the wind	101	201	2
101	Gone with the wind	101	201	4
101	Gone with the wind	101	204	3
103	The Sound of Music	103	203	2
103	The Sound of Music	103	205	3
104	E.T	104	205	2
104	E.T	104	208	3
106	Snow White	106	202	4
106	Snow White	106	206	5
107	Avatar	107	206	3
107	Avatar	107	207	5
108	Raiders of the Lost Arc	108	203	4
108	Raiders of the Lost Arc	108	203	2
108	Raiders of the Lost Arc	108	205	4

14 rows in set (0.00 sec)

10) Select movie id s of each movie with its title , reviewer id and stars received.

```
mysql> SELECT MOVIE.Movie_ID,MOVIE.Title, RATING.Reviewer_ID,RATING.Star from MOVIE,RATING where RATING.Movie_ID=MOVIE.Movie_ID;
```

Movie_ID	Title	Reviewer_ID	Star
101	Gone with the wind	201	2
101	Gone with the wind	201	4
101	Gone with the wind	204	3
103	The Sound of Music	203	2
103	The Sound of Music	205	3
104	E.T	205	2
104	E.T	208	3
106	Snow White	202	4
106	Snow White	206	5
107	Avatar	206	3
107	Avatar	207	5
108	Raiders of the Lost Arc	203	4
108	Raiders of the Lost Arc	203	2
108	Raiders of the Lost Arc	205	4

14 rows in set (0.00 sec)

11) Select movie id s of each movie with its title , reviewer id and stars received, where number of stars are less than or equal to three.

```
mysql> SELECT MOVIE.Movie_ID,MOVIE.Title, RATING.Reviewer_ID,RATING.Star from MOVIE,RATING where RATING.Movie_ID=MOVIE.Movie_ID;
```

Movie_ID	Title	Reviewer_ID	Star
101	Gone with the wind	201	2
101	Gone with the wind	201	4
101	Gone with the wind	204	3
103	The Sound of Music	203	2
103	The Sound of Music	205	3
104	E.T	205	2
104	E.T	208	3
106	Snow White	202	4
106	Snow White	206	5
107	Avatar	206	3
107	Avatar	207	5
108	Raiders of the Lost Arc	203	4
108	Raiders of the Lost Arc	203	2
108	Raiders of the Lost Arc	205	4

14 rows in set (0.00 sec)

12) Select movie id s of each movie with its title , reviewer id and stars received, where the number of stars is between two and four (two and four inclusive)

```
mysql> SELECT MOVIE.Movie_ID,MOVIE.Title, RATING.Reviewer_ID,RATING.Star from MOVIE,RATING where RATING.Movie_ID=MOVIE.Movie_ID AND Star<=4 AND Star>=2 ;
```

Movie_ID	Title	Reviewer_ID	Star
101	Gone with the wind	201	2
101	Gone with the wind	201	4
101	Gone with the wind	204	3
103	The Sound of Music	203	2
103	The Sound of Music	205	3
104	E.T	205	2
104	E.T	208	3
106	Snow White	202	4
107	Avatar	206	3
108	Raiders of the Lost Arc	203	4
108	Raiders of the Lost Arc	203	2
108	Raiders of the Lost Arc	205	4

12 rows in set (0.00 sec)

13) Select reviewer id s with the corresponding movie id reviewed by each reviewer

```
mysql> SELECT Reviewer_ID, Movie_ID from RATING;
+-----+-----+
| Reviewer_ID | Movie_ID |
+-----+-----+
| 201         | 101      |
| 201         | 101      |
| 202         | 106      |
| 203         | 103      |
| 203         | 108      |
| 203         | 108      |
| 204         | 101      |
| 205         | 103      |
| 205         | 104      |
| 205         | 108      |
| 206         | 107      |
| 206         | 106      |
| 207         | 107      |
| 208         | 104      |
+-----+-----+
14 rows in set (0.00 sec)
```

14) Select distinct tuples from the results produced by the execution of the above query (query number 14).

```
mysql> SELECT DISTINCT Reviewer_ID, Movie_ID from RATING;
+-----+-----+
| Reviewer_ID | Movie_ID |
+-----+-----+
| 201         | 101      |
| 202         | 106      |
| 203         | 103      |
| 203         | 108      |
| 204         | 101      |
| 205         | 103      |
| 205         | 104      |
| 205         | 108      |
| 206         | 107      |
| 206         | 106      |
| 207         | 107      |
| 208         | 104      |
+-----+-----+
12 rows in set (0.00 sec)
```

15) Select each movie id with its corresponding title , reviewer id , reviewer name and stars received.

```
mysql> SELECT RATING.Movie_ID, MOVIE.Title, RATING.Reviewer_ID, ReviewerName, Star from MOVIE, REVIEWER, RATING where MOVIE.Movie_ID=RATING.Movie_ID AND RATING.Reviewer_ID=REVIEWER.Reviewer_ID;
+-----+-----+-----+-----+-----+
| Movie_ID | Title           | Reviewer_ID | ReviewerName | Star |
+-----+-----+-----+-----+-----+
| 101      | Gone with the wind | 201         | Sarah Martinez | 2    |
| 101      | Gone with the wind | 201         | Sarah Martinez | 4    |
| 101      | Gone with the wind | 204         | Mike Anderson  | 3    |
| 103      | The Sound of Music | 203         | Brittany Harris | 2    |
| 103      | The Sound of Music | 205         | Chris Jackson  | 3    |
| 104      | E.T              | 205         | Chris Jackson  | 2    |
| 104      | E.T              | 208         | Ashley White   | 3    |
| 106      | Snow White       | 202         | Daniel Lewis   | 4    |
| 106      | Snow White       | 206         | Elizabeth Thomas | 5    |
| 107      | Avatar           | 206         | Elizabeth Thomas | 3    |
| 107      | Avatar           | 207         | James Cameron  | 5    |
| 108      | Raiders of the Lost Arc | 203         | Brittany Harris | 4    |
| 108      | Raiders of the Lost Arc | 203         | Brittany Harris | 2    |
| 108      | Raiders of the Lost Arc | 205         | Chris Jackson  | 4    |
+-----+-----+-----+-----+-----+
14 rows in set (0.00 sec)
```

16) Select each movie id with its corresponding title , reviewer id , reviewer name and stars received, where the number of stars received is equal to five.

```
mysql> SELECT RATING.Movie_ID,MOVIE.Title, RATING.Reviewer_ID, ReviewerName, Star from MOVIE,REVIEWER, RATING where MOVIE.Movie_ID=RATING.Movie_ID AND RATING.Reviewer_ID=REVIEWER.Reviewer_ID AND Star=5;
```

Movie_ID	Title	Reviewer_ID	ReviewerName	Star
106	Snow White	206	Elizabeth Thomas	5
107	Avatar	207	James Cameron	5

2 rows in set (0.00 sec)

17) Select movie title with its corresponding reviewer name and stars , where movie's rating date is missing.

```
mysql> SELECT MOVIE.Title, ReviewerName, Star from MOVIE,REVIEWER, RATING where MOVIE.Movie_ID=RATING.Movie_ID AND RATING.Reviewer_ID=REVIEWER.Reviewer_ID AND RATING.RatingDate IS NULL;
```

Title	ReviewerName	Star
Snow White	Daniel Lewis	4
Raiders of the Lost Arc	Chris Jackson	4

2 rows in set (0.00 sec)

18) Select all the movie director names and reviewer name s into one column. Do not include null values.

```
mysql> SELECT Director, ReviewerName FROM MOVIE,REVIEWER,RATING where MOVIE.Movie_ID=RATING.Movie_ID AND REVIEWER.Reviewer_ID=RATING.Reviewer_ID AND ReviewerName IS NOT NULL;
```

Director	ReviewerName
Victor Fleming	Sarah Martinez
Victor Fleming	Sarah Martinez
Victor Fleming	Mike Anderson
Robert Wise	Brittany Harris
Robert Wise	Chris Jackson
Steven Spielberg	Chris Jackson
Steven Spielberg	Ashley White
NULL	Daniel Lewis
NULL	Elizabeth Thomas
James Cameron	Elizabeth Thomas
James Cameron	James Cameron
Steven Spielberg	Brittany Harris
Steven Spielberg	Brittany Harris
Steven Spielberg	Chris Jackson

14 rows in set (0.00 sec)

19) Select the details about the reviewers who have a last name called ' Martinez'.

20) Select the details about the ratings which have been rated before the 10th day of the month. Use substring comparison.

```
mysql> SELECT * FROM RATING WHERE SUBSTR(RatingDate,9,2)<10 AND RatingDate IS NOT NULL;
ERROR 1054 (42S22): Unknown column 'RatingDate9' in 'where clause'
mysql> SELECT * FROM RATING WHERE SUBSTR(RatingDate,9,2)<10 AND RatingDate IS NOT NULL;
```

Reviewer_ID	Movie_ID	Star	RatingDate
204	101	3	2011-01-09
208	104	3	2011-01-02

2 rows in set (0.01 sec)

21. Write the above query (query number 20) without using substring comparison.

```
mysql> SELECT * FROM RATING WHERE MOD(RatingDate,9,2)<10 AND RatingDate IS NOT NULL;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '2)<10 AND R
atingDate IS NOT NULL' at line 1
mysql> SELECT * FROM RATING WHERE MOD(RatingDate,50)<10 AND RatingDate IS NOT NULL;
```

Reviewer_ID	Movie_ID	Star	RatingDate
204	101	3	2011-01-09
208	104	3	2011-01-02

2 rows in set (0.00 sec)

22. Show the effect of giving one more star to the movies reviewed by 'Brittany Harris'. Here select relevant details from the RATING table.

```
mysql> SELECT *, Star, Star+1 FROM REVIEWER, RATING where REVIEWER.Reviewer_ID=RATING.Reviewer_ID AND ReviewerName="Brittany Harris";
```

Reviewer_ID	ReviewerName	Reviewer_ID	Movie_ID	Star	RatingDate	Star	Star+1
203	Brittany Harris	203	103	2	2011-01-20	2	3
203	Brittany Harris	203	108	4	2011-01-12	4	5
203	Brittany Harris	203	108	2	2011-01-30	2	3

3 rows in set (0.01 sec)

23) Select movie titles with its reviewer name and stars received. Order the result by movie title in the alphabetical order.

```
mysql> SELECT Title, ReviewerName, Star FROM MOVIE, REVIEWER, RATING where MOVIE.Movie_ID= RATING.Movie_ID AND REVIEWER.Reviewer_ID=RATING.Reviewer_ID ORDER BY Title;
```

Title	ReviewerName	Star
Avatar	Elizabeth Thomas	3
Avatar	James Cameron	5
E.T	Chris Jackson	2
E.T	Ashley White	3
Gone with the wind	Sarah Martinez	2
Gone with the wind	Sarah Martinez	4
Gone with the wind	Mike Anderson	3
Raiders of the Lost Arc	Brittany Harris	4
Raiders of the Lost Arc	Brittany Harris	2
Raiders of the Lost Arc	Chris Jackson	4
Snow White	Daniel Lewis	4
Snow White	Elizabeth Thomas	5
The Sound of Music	Brittany Harris	2
The Sound of Music	Chris Jackson	3

14 rows in set (0.01 sec)

24) Select movie titles with its stars received and rating date. Order the result by movie title in the alphabetical order, then by stars and rating date both in descending order.

```
mysql> SELECT Title, Star, RatingDate FROM MOVIE LEFT JOIN RATING ON MOVIE.Movie_ID=RATING.Movie_ID ORDER BY Title, Star DESC, RatingDate DESC;
```

Title	Star	RatingDate
Avatar	5	2011-01-20
Avatar	3	2011-01-15
E.T	3	2011-01-02
E.T	2	2011-01-22
Gone with the wind	4	2011-01-27
Gone with the wind	3	2011-01-09
Gone with the wind	2	2011-01-22
Raiders of the Lost Arc	4	2011-01-12
Raiders of the Lost Arc	4	NULL
Raiders of the Lost Arc	2	2011-01-30
Snow White	5	2011-01-19
Snow White	4	NULL
Star Wars	NULL	NULL
The Sound of Music	3	2011-01-27
The Sound of Music	2	2011-01-20
Titanic	NULL	NULL

16 rows in set (0.00 sec)

25) Write a nested query to retrieve the details of the movies directed by a director who is also a reviewer.

```
mysql> SELECT * FROM MOVIE WHERE DIRECTOR IN (SELECT ReviewerName FROM REVIEWER);
```

Movie_ID	Title	Year	Director
105	Titanic	1997	James Cameron
107	Avatar	2009	James Cameron

2 rows in set (0.01 sec)

